



# DARwIn-OP Education

## Dossier Technique





## Table des matières

<b>Table des matières</b> .....	<b>2</b>
<b>1. Définition du produit réel</b> .....	<b>5</b>
1.1. <i>Présentation générale du produit réel</i> .....	5
1.1.1. Définition et présentation.....	5
1.1.2. Historique et évolution de DARwIn-OP.....	7
1.1.3. Robots, littérature et éthique ( <i>source Wikipédia</i> ).....	9
1.1.4. Présentation technique, commerciale, environnementale.....	9
1.2. <i>Expression fonctionnelle du produit réel</i> .....	14
1.2.1. Expression fonctionnelle du besoin.....	14
1.2.1.1. Analyse du besoin.....	14
1.2.1.2. Expression du besoin.....	15
1.2.1.3. Problématique.....	16
1.2.1.4. Validation du besoin.....	16
1.2.2. Expression du cahier des charges.....	17
1.2.2.1. Diagramme des cas d'utilisation.....	17
1.2.2.2. Diagramme des exigences.....	19
1.3. <i>Description fonctionnelle</i> .....	20
1.3.1. Décomposition de DARwIn-OP.....	20
1.3.2. Schéma blocs.....	22
1.3.3. Schéma cinématique, terminologie.....	22
1.4. <i>Description de la structure mécanique</i> .....	24
1.4.1. Vues d'ensemble de DARwIn-OP avec et sans habillage.....	24
1.4.2. Nomenclatures mécaniques.....	25
1.4.2.1. Squelette de DARwIn-OP.....	25
1.4.2.2. Habillage de DARwIn-OP.....	27
1.4.2.3. Agencement des servomoteurs.....	28
1.4.3. Bloc Membres inférieurs.....	29
1.4.3.1. Fonction.....	29
1.4.3.2. Présentation et terminologie.....	29
1.4.3.3. Vue en perspective et éclaté de la hanche et de la cuisse droites.....	31
1.4.3.4. Vue en perspective et éclaté de la jambe et du genou droits.....	32
1.4.3.5. Vue en perspective et éclaté de la cheville et du pied droits.....	33
1.4.3.6. Caractéristiques dimensionnelles.....	34
1.4.4. Bloc Membres supérieurs.....	34
1.4.4.1. Fonction.....	34
1.4.4.2. Présentation, vue en perspective et éclaté.....	35
1.4.4.3. Caractéristiques dimensionnelles.....	36
1.4.5. Bloc Tête.....	37
1.4.6. Bloc Servomoteurs Dynamixel MX-28.....	38
1.4.6.1. Fonction.....	38
1.4.6.2. Description.....	38
1.4.6.3. Caractéristiques des servomoteurs MX-28.....	38
1.4.6.4. Caractéristiques techniques des moteurs.....	40
1.4.6.5. Capteur de position (codeur).....	41
1.4.6.6. Plans d'ensemble et schéma.....	42
1.4.6.7. Paramètres de pilotage des servomoteurs.....	47
1.4.6.8. Connexions des servomoteurs.....	47





<b>1.5.</b>	<b>Description de la structure de la commande.....</b>	<b>49</b>
1.5.1.	Bloc Contrôleur principal FitPC (carte mère, ou main board, ou main controller) .....	49
1.5.2.	Bloc HDMI .....	52
1.5.3.	Bloc USB .....	52
1.5.4.	Bloc Wi-Fi .....	53
1.5.5.	Bloc Ethernet .....	53
1.5.6.	Bloc « Contrôleur secondaire CM-730 » (Sub controller) .....	54
1.5.7.	Bloc gyromètre.....	56
1.5.8.	Bloc Accéléromètre .....	59
1.5.9.	Bloc E/S boutons.....	61
1.5.10.	Bloc LEDs .....	62
1.5.11.	Bloc Batterie Li-Po .....	63
1.5.12.	Bloc Caméra : l'œil de DARwIn-OP .....	63
1.5.13.	Bloc Microphone.....	64
1.5.14.	Bloc Haut-parleur.....	65
1.5.15.	Description des flux échangés entre les blocs internes .....	66
1.5.15.1.	DC.....	67
1.5.15.2.	Hp.....	67
1.5.15.3.	MIC.....	68
1.5.15.4.	LEDs tête .....	68
1.5.15.5.	Port HDMI.....	68
1.5.15.6.	Port Contrôle, Wi-Fi, Port USB.....	68
1.5.15.7.	Carte-arr .....	68
1.5.15.8.	Bus USB.....	69
1.5.15.9.	Bus Dynamixel.....	70
<b>2.</b>	<b>Définition du produit didactique.....</b>	<b>75</b>
2.1.	Identification du produit .....	75
2.2.	Présentation générale du produit didactique .....	75
2.3.	Déclaration de conformité CE .....	75
2.4.	Notice d'instruction du produit didactique .....	75
2.4.1.	Mise en service de l'équipement .....	75
2.4.2.	Contenu du colis .....	75
2.4.3.	Localisation des capteurs et des raccords .....	76
2.4.4.	Manutention .....	77
2.4.5.	Assemblage et raccordement .....	77
2.4.6.	Première mise en service .....	77
2.4.6.1.	Mise en charge des batteries.....	77
2.4.6.2.	Statut de charge .....	78
2.4.6.3.	Changer de batterie sans éteindre DARwIn-OP.....	79
2.4.7.	Notice d'utilisation .....	80
2.4.7.1.	Préparation de DARwIn-OP et mise sous tension .....	80
2.4.7.2.	Exécution des programmes de démonstration.....	84
2.4.7.3.	Calibration de la couleur et de la balance des blancs .....	88
2.4.7.4.	Arrêter DARwIn-OP .....	92
2.4.7.5.	Récapitulatif : diagramme de mode d'utilisation de DARwIn-OP .....	93
2.4.8.	Entretien de l'équipement .....	94
2.4.8.1.	Nettoyage.....	94
2.4.8.2.	Maintenance préventive .....	94
2.4.9.	Sécurité.....	95



2.4.9.1.	Consignes de sécurité.....	95
2.4.9.2.	Procédure de consignation .....	95
2.4.10.	Maintenance corrective .....	95
2.4.10.1.	Remplacement du fusible .....	95
2.4.10.2.	Remplacement d'un servomoteur.....	95
2.4.10.3.	Liste des opérations pour la maintenance .....	96
2.5.	<i>Documents techniques</i> .....	104
2.5.1.	Schémas électriques / électroniques .....	104
2.5.1.1.	Buzzer .....	104
2.5.1.2.	ZigBee et Bluetooth.....	104
2.5.1.3.	Réseau RS-485.....	105
2.5.1.4.	Connecteur d'extensions pour capteurs optionnels .....	105
2.5.1.5.	Convertisseur USB / Série.....	106
2.5.1.6.	Unité de contrôle sans fil de la puissance (ZigBee).....	106
2.5.1.7.	Volt ADC .....	107
2.5.1.8.	Dynamixel Power Control .....	107
2.5.2.	Les programmes.....	108
2.5.2.1.	Différents points de vue .....	108
2.5.2.2.	Module « Walking », ou module de « Marche » .....	109
2.5.2.3.	Module « Marche simple ».....	120
2.5.2.4.	Module « Action Script » .....	122
2.5.2.5.	Module « Ball Traking », ou « Suivi de balle ».....	123



## 1. Définition du produit réel

### 1.1. Présentation générale du produit réel

#### 1.1.1. Définition et présentation

Le robot humanoïde DARwIn-OP (Dynamic Anthropomorphic Robot with Intelligence – Open Platform) est le dernier né des robots humanoïdes. Avec son apparence futuriste (Figure 1.1-1), il intègre toutes les dernières technologies et des fonctionnalités très avancées.

DARwIn-OP est capable de marcher, se relever après une chute en avant ou en arrière, suivre une balle et jouer au football, parler et reconnaître des documents.



Figure 1.1-1 : DARwIn-OP



### ■ Qu'est-ce qu'un robot humanoïde ?

Un **robot humanoïde** est un robot dont l'apparence générale rappelle celle d'un corps humain. Généralement, les robots humanoïdes ont un torse avec une tête, deux bras et deux jambes, bien que certains modèles ne représentent qu'une partie du corps, par exemple à partir de la taille.

Certains robots humanoïdes peuvent avoir un « visage », avec des « yeux » et une « bouche ».

Un **androïde** est un robot humanoïde dont l'apparence est très proche de l'humain.

### ■ Performances et caractéristiques (Figure 1.1-2)

- ✓ Hauteur : 45,45 cm
- ✓ Poids : 2,9 kg
- ✓ Processeur principal Intel Atom Z530 à 1,6 GHz avec 4GO mémoire flash SSD
- ✓ Carte contrôleur CM-730 avec ARM Cortex M3 à 72MHz
- ✓ Vitesse de marche par défaut :  $24\text{cm}\cdot\text{s}^{-1}$ , modifiable par l'utilisateur
- ✓ Temps mis par le robot pour se relever quand il est sur le dos : 2,8 s
- ✓ Temps mis par le robot pour se relever quand il est sur le ventre: 3,9 s

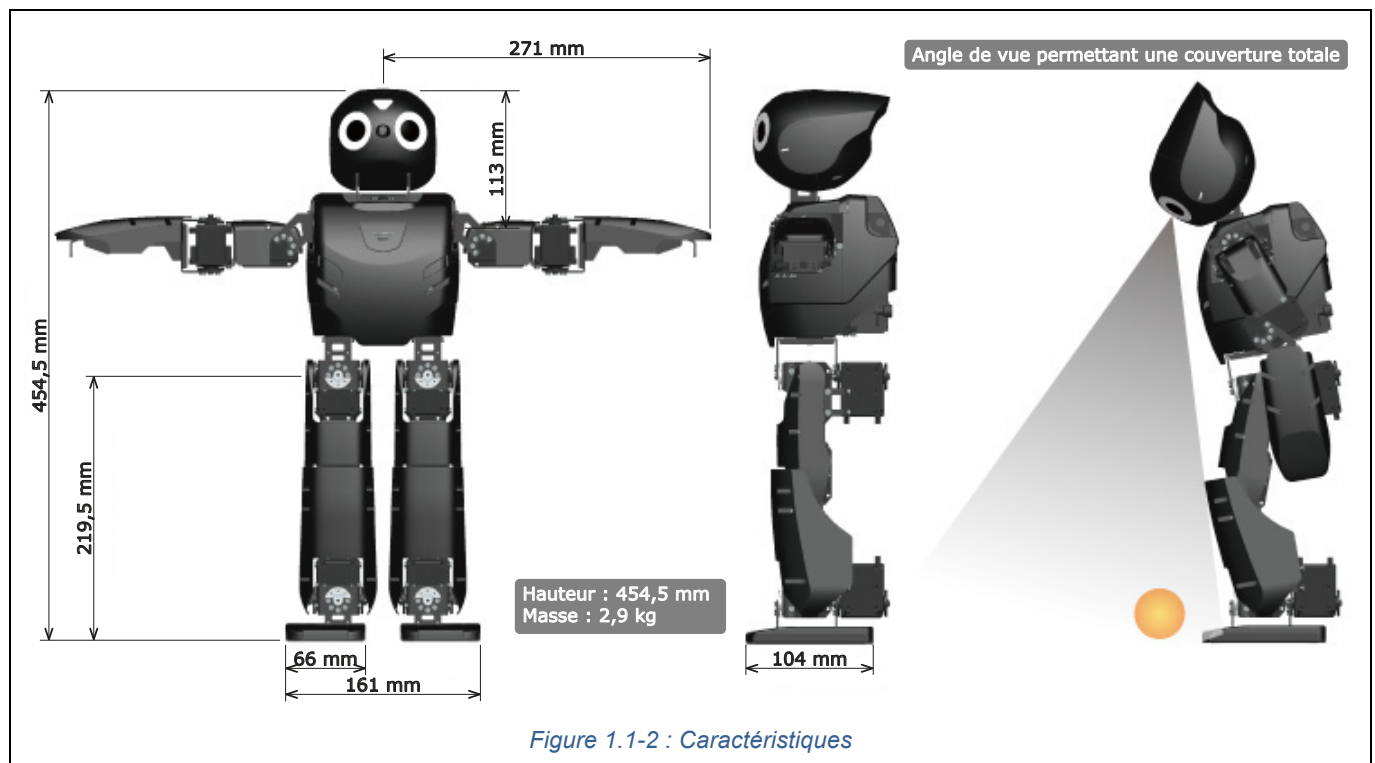


Figure 1.1-2 : Caractéristiques

### ■ Connectique disponible

La connectique disponible sur le robot humanoïde DARwIn-OP est la suivante :

- ✓ Un port HDMI
- ✓ Deux ports USB
- ✓ Un port Ethernet
- ✓ Un câble batterie



## ■ Capteurs

Le robot DARwIn-Op est doté des capteurs suivants :

- ✓ Un gyromètre 3 axes
- ✓ Un accéléromètre 3 axes
- ✓ Deux microphones
- ✓ Une caméra USB, 2 MPixels
- ✓ Des yeux avec LED

## ■ Motorisation

- ✓ La motorisation du robot humanoïde est composée de 20 servomoteurs Dynamixel MX-28
- ✓ 6 degrés de liberté pour chaque jambe
- ✓ 3 degrés de liberté pour chaque bras
- ✓ 2 degrés de liberté pour le cou

Les moteurs Dynamixel MX-28 sont parmi les meilleurs servos du marché.

## ■ Alimentation

Le robot DARwIn-OP est doté d'une batterie Li-Po 11.1V 1000mAh. Un mode stand-by est disponible afin de limiter la consommation électrique. La batterie a une autonomie de 30 minutes.

## ■ Open source et plateforme complètement ouverte

La désignation "**open source**" s'applique aux logiciels dont la licence respecte des critères précisément établis par l'*Open Source Initiative*, c'est-à-dire la possibilité de **libre redistribution, d'accès au code source et aux travaux dérivés**.

La plateforme DARwIn-OP est complètement ouverte grâce à :

- ✓ sa base d'ordinateur PC fonctionnant avec le système d'exploitation Linux-Ubuntu ;
- ✓ des spécifications matérielles disponibles (carte électronique, modèle 3D) ;
- ✓ des spécifications logicielles disponibles (codes sources et kit de composant logiciel) ;
- ✓ des extensions possibles avec 13 entrées / sorties et un port analogique / digital ;
- ✓ une structure modulaire pour une maintenance aisée.

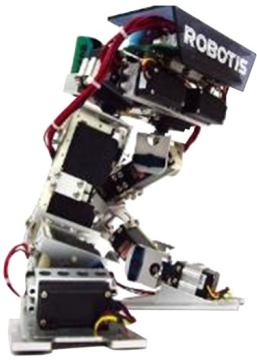



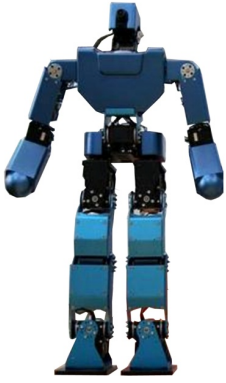







### 1.1.2. Historique et évolution de DARwIn-OP

Le robot DARwIn-OP, c'est plus de 10 années de rêve et d'innovation, de recherche et développement, au sein de la société ROBOTIS. Les partenaires du projet :





Le tableau suivant montre les différentes étapes de l'évolution du robot DARwIn.

			
Avant 2000	2001 - 2002	2003	2004
			
2005	2006	2007	
			
2008	2009	2010	2011





### 1.1.3. Robots, littérature et éthique (source Wikipédia)

De nombreux écrivains ont écrit des œuvres sur les robots, dont **Isaac Asimov** (1920-1992) qui a publié de très nombreux romans et nouvelles dans lesquels sont, entre autres soulevés des problèmes d'éthique.

L'ensemble de son œuvre forme une seule grande histoire, le cycle des Robots, qui s'étale sur plusieurs millénaires. Toutes les nouvelles de robotique publiées par l'auteur ont été regroupées dans deux grands recueils nommés Le Grand Livre des robots.

Il renouvelle complètement ce thème en inventant des « robots positroniques » gouvernés par trois lois protégeant les êtres humains et, *a priori*, parfaites et inviolables. Le jeu d'Asimov consiste à imaginer des situations révélant des failles de ces lois (exemple : un robot peut-il, restant passif, laisser un humain *fumer une cigarette* ?) et des bizarreries de comportement de robots qui *semblent* les enfreindre, puis à faire découvrir au lecteur comment cela est possible, à la manière d'une enquête policière.

Les trois lois sont :

**Première Loi** : « Un robot ne peut porter atteinte à un être humain ni, restant passif, laisser cet être humain exposé au danger. » ;

**Deuxième Loi** : « Un robot doit obéir aux ordres donnés par les êtres humains, sauf si de tels ordres sont en contradiction avec la Première Loi. » ;

**Troisième Loi** : « Un robot doit protéger son existence dans la mesure où cette protection n'entre pas en contradiction avec la Première ou la Deuxième Loi. »

Deux robots exceptionnels, R. Daneel Olivaw et R. Giskard Reventlov, en viennent à ajouter une **Loi Zéro**, qui stipule qu'un robot ne peut porter atteinte à l'humanité dans son ensemble, même pour protéger un être humain :

**Loi Zéro** : « Un robot ne peut ni nuire à l'humanité ni, restant passif, permettre que l'humanité souffre d'un mal. »

Cette loi est apparue dans *Les Robots et l'empire* (chapitre LXIII).

Le thème des robots, tel que traité par Asimov, constitue aussi un plaidoyer antiraciste *discret, mais sûr* : les robots, de plus en plus perfectionnés et dotés d'aspects de plus en plus humains, deviennent méprisés, voire haïs, par bien des êtres humains — d'autant que les trois lois les mettent à l'abri de défauts qu'on pourrait leur reprocher.

### 1.1.4. Présentation technique, commerciale, environnementale

Les quatre pages suivantes reproduisent la notice commerciale publiée par Robotis.



**Open Platform Humanoid Project**

# DARwin-OP

DARwin-OP (Dynamic Anthropomorphic Robot with Intelligence-Open Platform) is an affordable, miniature-humanoid-robot platform with advanced computational power, sophisticated sensors, high payload capacity, and dynamic motion ability to enable many exciting research and education activities.



R&D Collaborators • 



# DARwin-OP

## 1. Standard PC-based Robot with Convenient Interfaces



## 2. High Performance and Advanced Features

- Default walking speed: 24.0 cm/sec (9.5 in/sec) 0.25 sec/step - user modifiable gait
- Default standing up time from ground: 2.8 sec (from facing down) and 3.9 sec (from facing up) - user modifiable speed
- Built-in PC: 1.6 GHz Intel Atom Z530 on-board 4GB flash SSD  
(specs as of Dec. 2010. Check <http://sourceforge.net/projects/darwinop>)
- Management controller (CM-730): ARM CortexM3 STM32F103RE 72MHz
- 20 actuator modules (6 DOF leg x2+ 3 DOF arm x2 + 2 DOF neck)
- 1 spare actuator (for maintenance and expansion)
- Actuators with durable metallic gears (Dynamixel RX-28M)
- Self-maintenance kit (easy to follow steps and instructions)
- Standby mode for low power consumption
- 4.5Mbps high-speed Dynamixel bus for joint control
- Battery (30 minutes of operations), charger, and external power adapter  
(Battery can be removed from robot without shutting down by plugging in external power before removal)
- Versatile functionality (can accept legacy, current, and future peripherals)
- 3-axis gyro, 3-axis accelerometer, button x3, detection microphone x2 (optional)



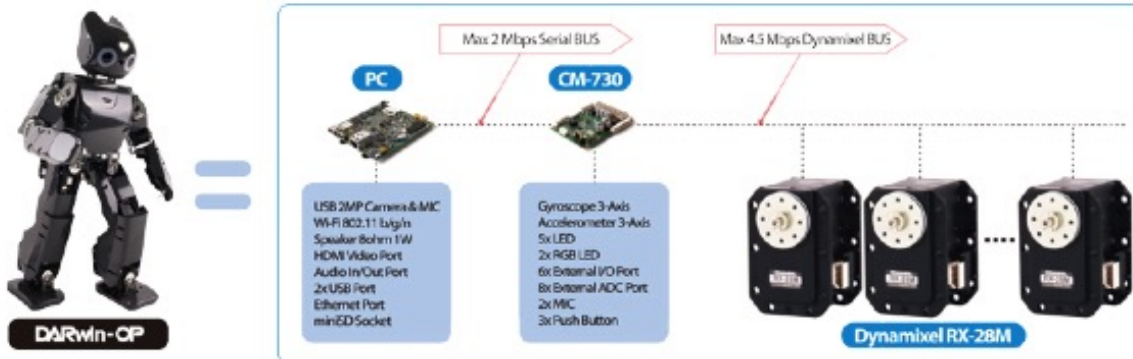


**Open Platform Humanoid Project**

**3. Efficient and Versatile Modular Configuration**



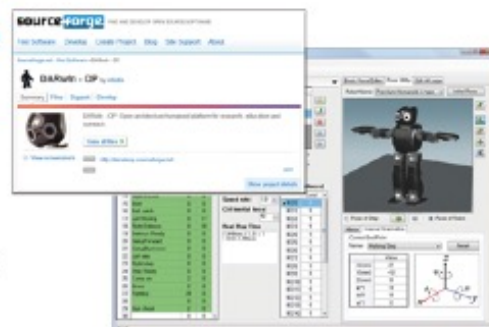
**4. Simple and Clean Architecture**



**5. Open Platform (Hardware and Software)**

<http://sourceforge.net/projects/darwinop>

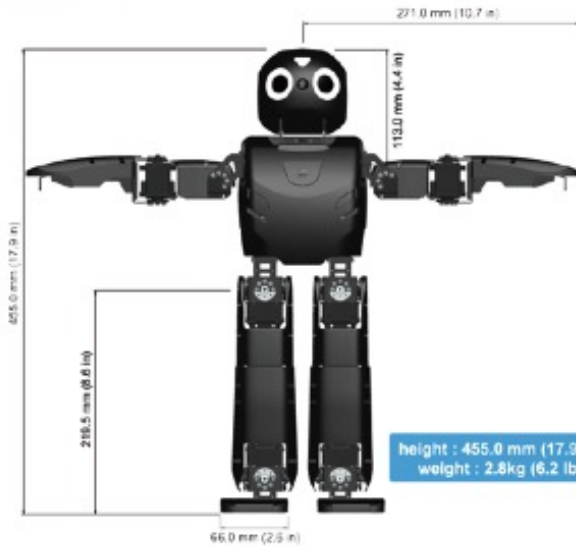
- Mechanics Information (Dimensions, Kinematics, Dynamics, CAD data)
- Electronics Information (Controllers, Sub B/Ds, Schematics, Part Information)
- Software Information (Development Environment, Framework, Source Code)
- Management Information (Detailed Assembly Diagrams, User Maintenance Guide)
- Community Resources (User-developed Code, Various Application Examples)



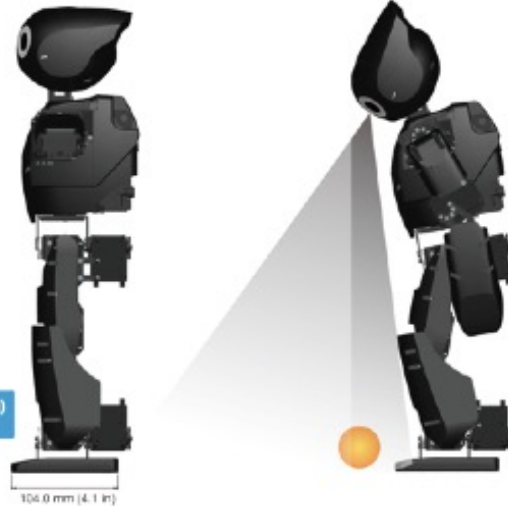
www.robotis.com



**6. Dimensions and Weight**



Wide-angle view allows plentiful coverage



**DYNAMIXEL**

- Specialized robot all-in-one actuator for robotic applications
- High speed bus and simple wiring
- Various models, mechanical links, and accessories

Dynamixel applications



Dynamixel RX-28M for DARwIn-OP

- Maxon RE-Max customized DC motor
- 193:1 reduction ratio
- Up to 4.5 Mbps Buffered TTL interface
- User programmable PID Gain
- Auto-adaptable feed forward Control
- Dynamixel 1.0 and 2.0 protocol
- Up to 4096 resolution non-contact feedback

**ROBOTIS**

[USA]

16 Technology Dr, Suite 106, Irvine, CA, 92618  
contactus2@robotis.com

[KOREA]

#1506 Ace High End No. 3, 371-50 Gasandong  
Geumcheongu, Seoul, Korea 153-783  
Tel : +82-70-8671-2800 Fax : +82-70-8230-1336  
contactus@robotis.com



## 1.2. Expression fonctionnelle du produit réel

Parmi les nombreuses applications de robotique dans lesquelles les robots humanoïdes peuvent être impliqués (surveillance, intervention en milieu hostile, jeu, compagnie, recherche, exhibition, etc.), nous choisissons ici la “**robotique d'assistance à la personne**”.

### 1.2.1. Expression fonctionnelle du besoin

#### 1.2.1.1. Analyse du besoin

##### ■ *Pourquoi le besoin existe-t-il ?*

Avec le vieillissement annoncé des populations et l'augmentation du nombre de personnes dépendantes, il est de plus en plus nécessaire d'assister la personne humaine dans ses mouvements, ses besoins ou de lui tenir compagnie.

La formation et l'embauche de personnels spécialisés sont une réponse à ce besoin mais la nécessité d'avoir une présence permanente, de réaliser des tâches souvent répétitives et le coût que tout cela implique sont autant d'obstacles à surmonter.

##### ■ *Comment ce besoin pourrait-il disparaître ?*

Ce besoin pourrait disparaître en utilisant des moyens techniques reproduisant au mieux le comportement des humains, par exemple :

- ✓ Apporter une aide physique dans la perte d'autonomie fonctionnelle.
- ✓ Apporter un soutien émotionnel et une compagnie à des personnes atteintes de handicaps.
- ✓ Évoluer dans un environnement humain afin d'interagir avec les objets du quotidien, communiquer avec leurs propriétaires ou leur porter secours.

##### ■ *Comment pourrait-il évoluer ?*

Ce besoin pourrait évoluer en créant des robots humanoïdes ayant un comportement proche de l'humain, tant dans l'allure physique que dans l'interaction et la communication avec la personne assistée.





**1.2.1.2. Expression du besoin**

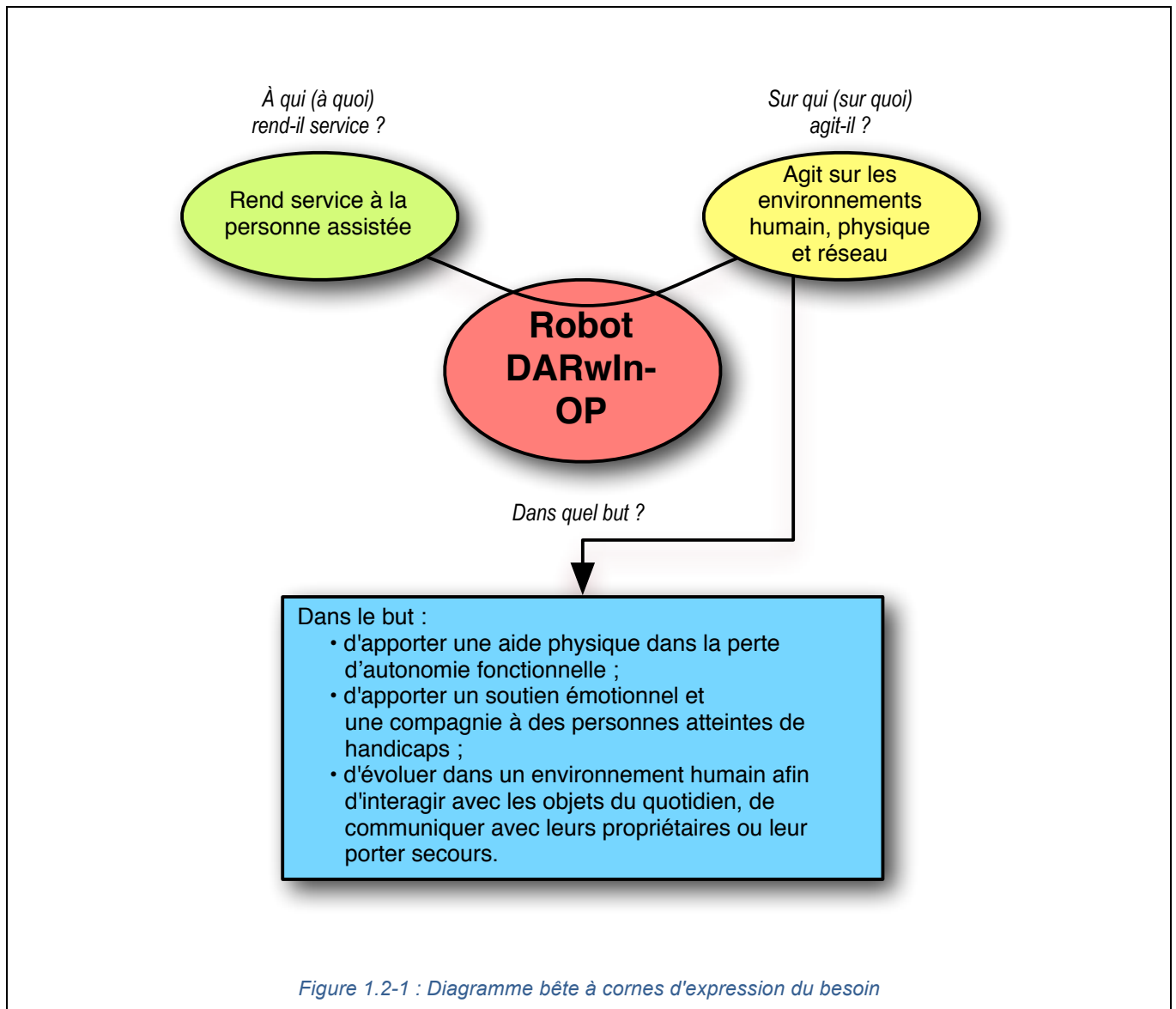


Figure 1.2-1 : Diagramme bête à cornes d'expression du besoin

Afin de réduire la complexité de cette étude fonctionnelle, nous limiterons notre besoin à l'expression suivante : « **Évoluer dans un environnement humain afin d'interagir avec les objets du quotidien, communiquer avec leurs propriétaires ou leur porter secours** ».

Il est évident que le robot DARwIn-OP, vu sa taille, ne pourra pas satisfaire à toutes les fonctions que l'on attend d'un robot d'assistance à la personne. Nous allons donc nous placer dans le contexte où DARwIn-OP est utilisé dans une **plateforme de recherche et développement**.



### 1.2.1.3. Problématique

Dans ce contexte, la problématique est : « Réaliser des plateformes de recherche et développement reproduisant, à échelle réduite, les mouvements, leur pilotage et leur coordination dans la conception et la réalisation de robots humanoïdes ».

- **Acteurs et éléments environnants dans le contexte d'un laboratoire de recherche ou de Sciences Industrielles pour l'Ingénieur**

Le besoin est maintenant : « Reproduire, de manière autonome, à échelle réduite, les mouvements et leur pilotage d'un robot humanoïde évoluant dans un environnement humain ».

La Figure 1.2-2 définit les contextes dans le cas de l'utilisation du robot dans une plateforme de recherche et de développement.

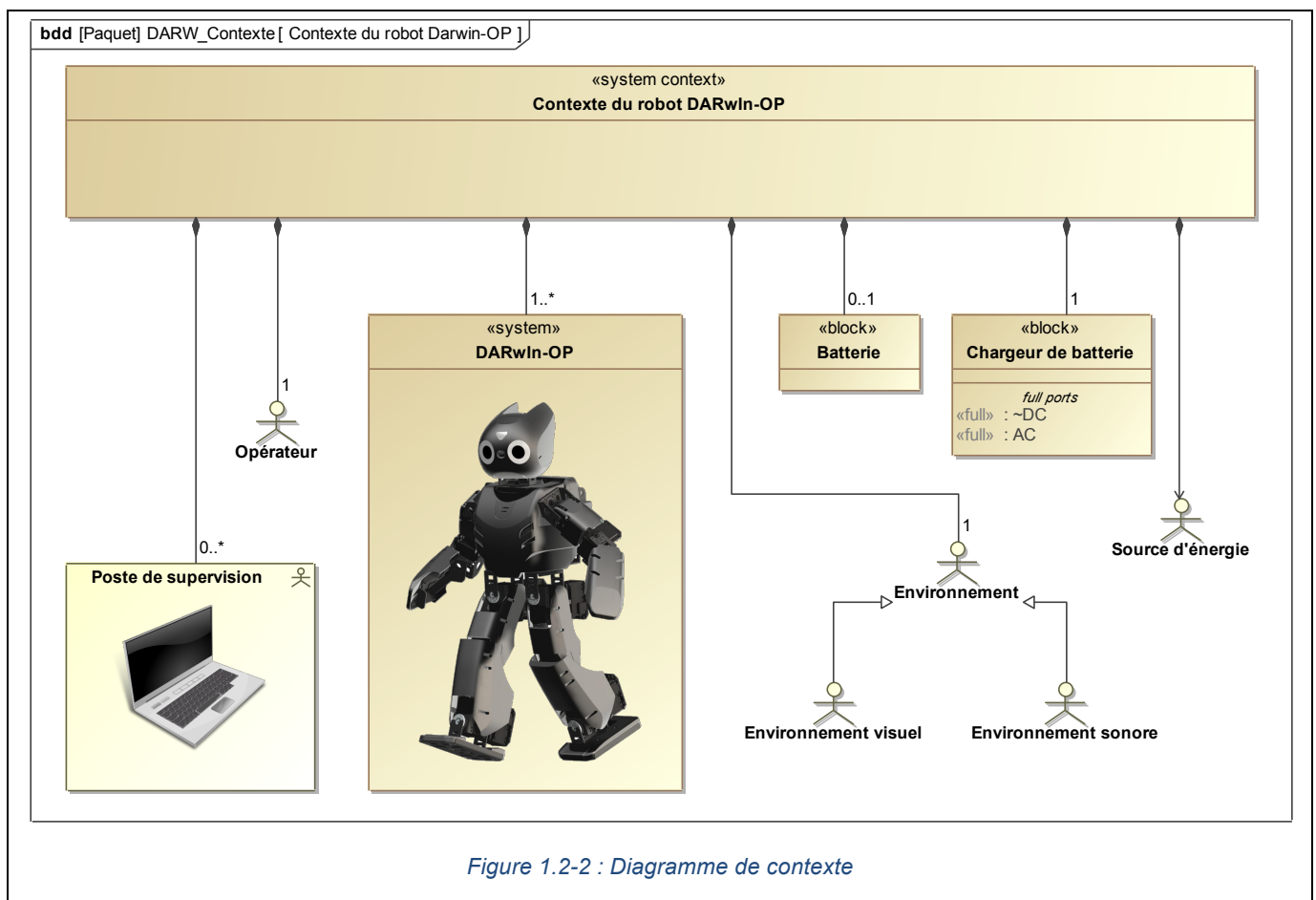


Figure 1.2-2 : Diagramme de contexte

### 1.2.1.4. Validation du besoin

Le besoin est validé par la création de plateformes d'études qui permettent la conception et la réalisation de robots humanoïdes interactifs et communicants.



## 1.2.2. Expression du cahier des charges

Pour cette description, nous entendons par “système” le **robot DARwin-OP** et son **contexte** lié au **laboratoire de recherche et de développement**, ou encore le **laboratoire de Sciences Industrielles pour l’Ingénieur**.

### 1.2.2.1. Diagramme des cas d’utilisation

Le package de la figure Figure 1.2-3 définit globalement les deux cas d’utilisation du système, qui peuvent être définis comme suit :

- ✓ simuler un être humain ;
- ✓ superviser un robot DARwin.

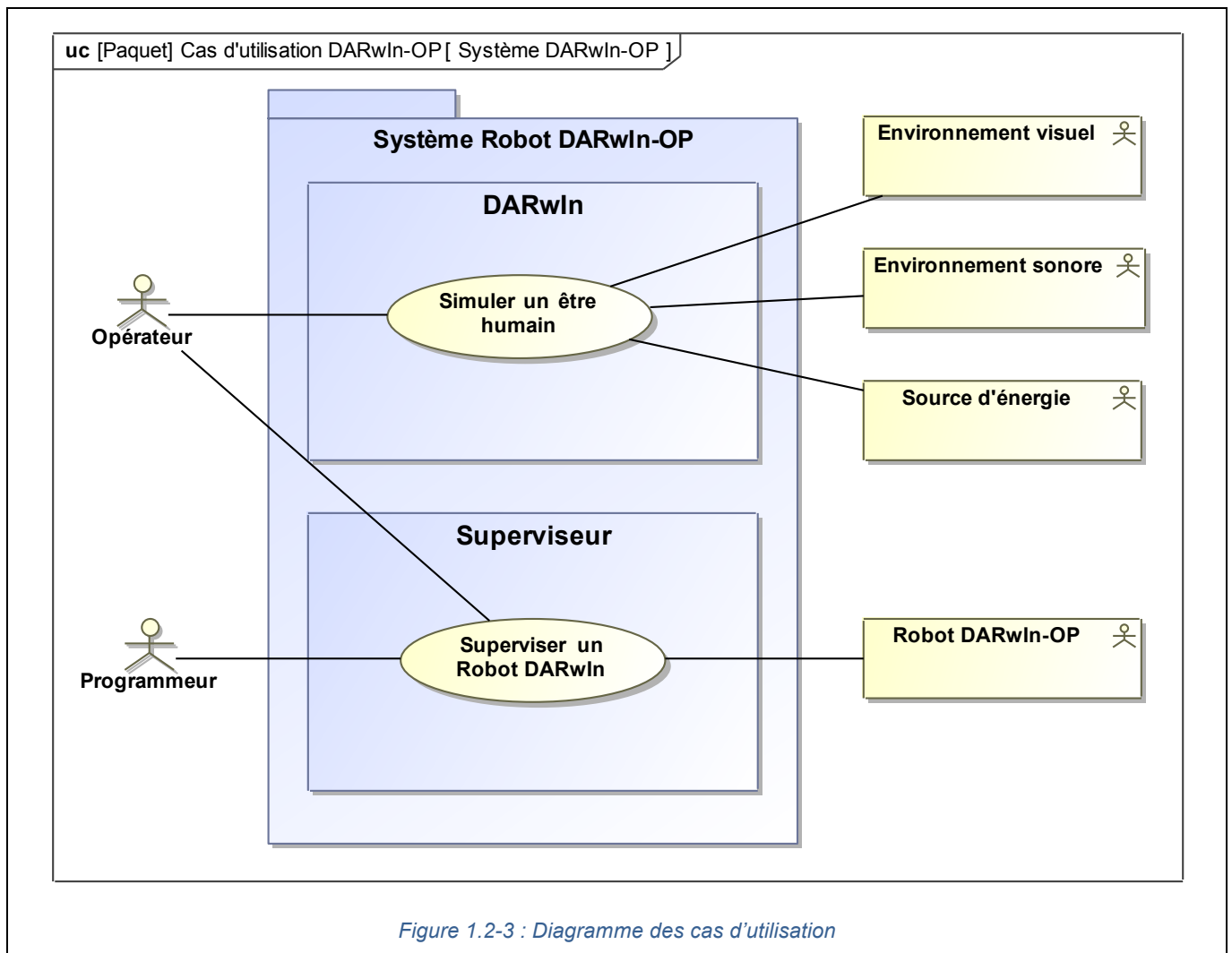


Figure 1.2-3 : Diagramme des cas d'utilisation

Les deux figures suivantes détaillent chacun de ces cas d’utilisation.



uc [Paquet] Cas d'utilisation DARwIn-OP [ Simuler un être humain détaillé ]

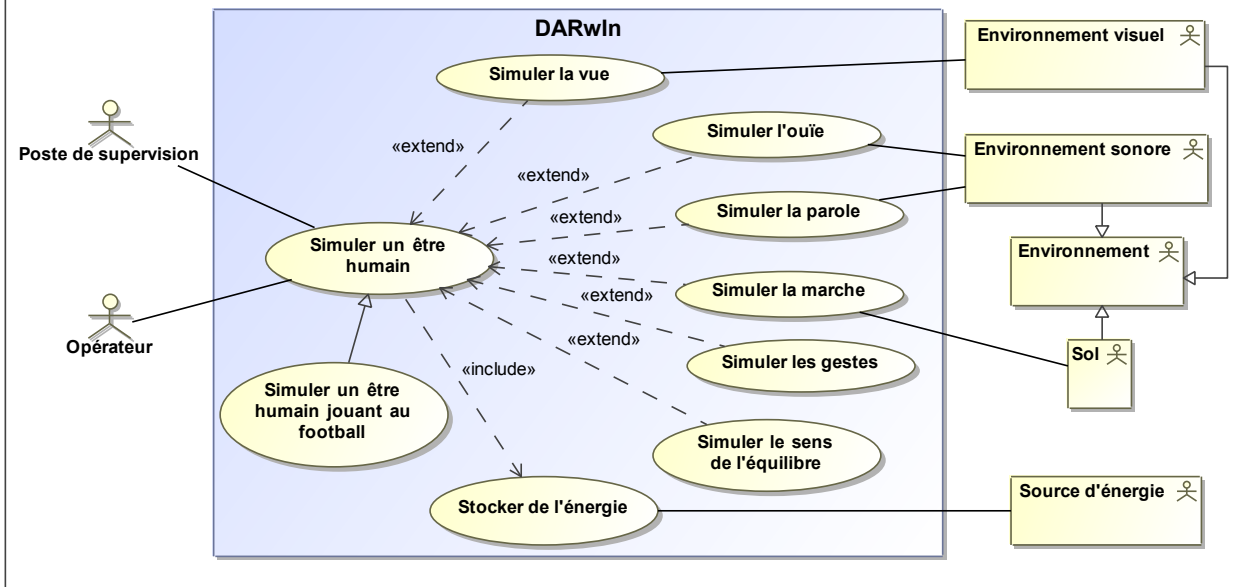


Figure 1.2-4 : Diagramme du cas d'utilisation « Simuler un être humain »

uc [Paquet] Cas d'utilisation DARwIn-OP [ Superviser un Robot DARwIn ]

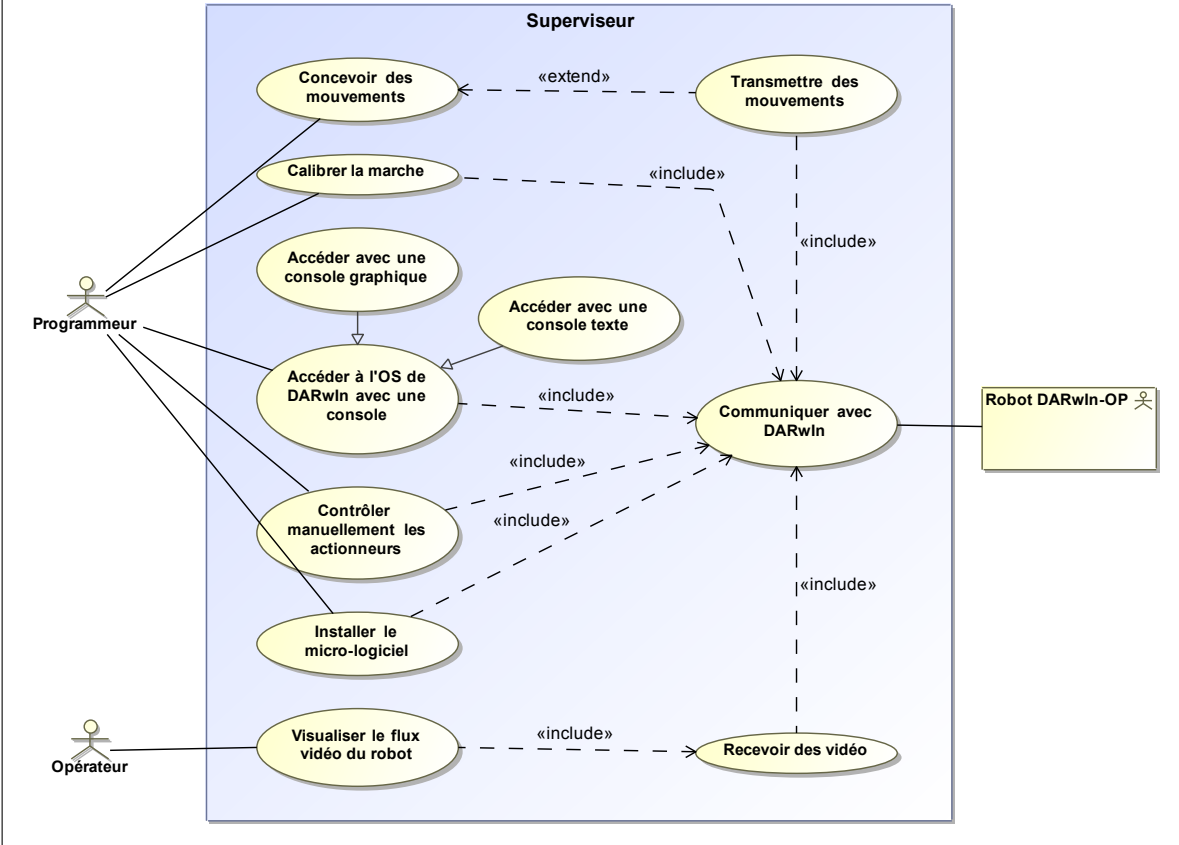


Figure 1.2-5 : Diagramme du cas d'utilisation « Superviser un robot DARwIn-OP »



**1.2.2.2. Diagramme des exigences**

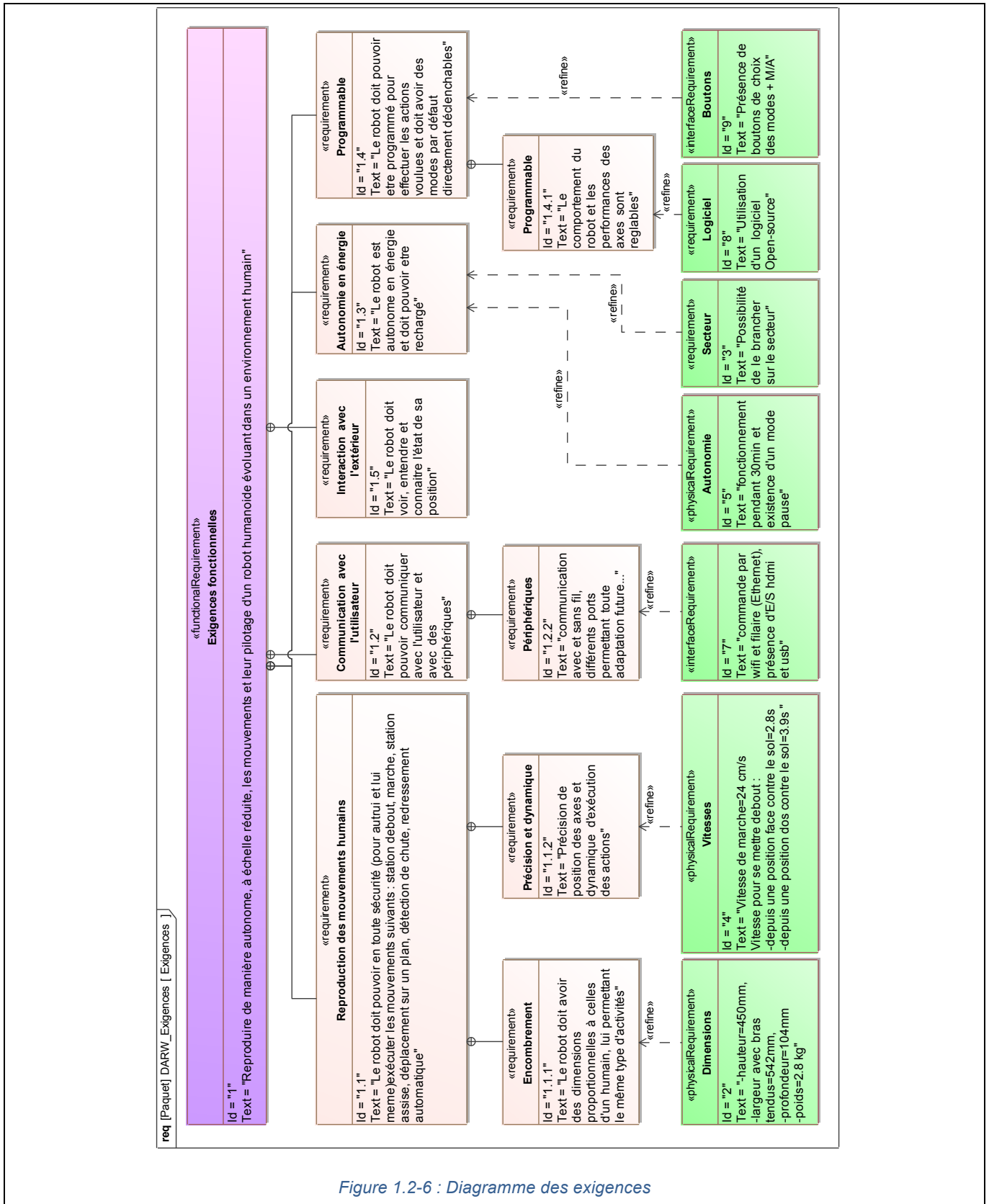


Figure 1.2-6 : Diagramme des exigences



## 1.3. Description fonctionnelle

### 1.3.1. Décomposition de DARwIn-OP

Le robot DARwIn-OP est équipé des dernières technologies, basé sur une plateforme PC embarqué SBC FTIPC2i de la société COMPULAB (sous linux), assisté d'un contrôleur CM-730 de Robotis pour le contrôle/commande des capteurs et des actionneurs spécifiques à la robotique (gyromètre, accéléromètre, bus Dynamixel, etc.). La Figure 1.3-1 recense l'ensemble des capteurs et actionneurs, ainsi que la connectique et les interfaces accessibles sur le robot.



Figure 1.3-1 : Les capteurs, les actionneurs et la connectique de DARwIn-OP

Afin de comprendre la structure interne de DARwIn-OP, il est utile de proposer une décomposition du bloc « DARwIn-OP» sous la forme d'un diagramme de définition de blocs de structure (Figure 1.3-2).







### 1.3.2. Schéma blocs

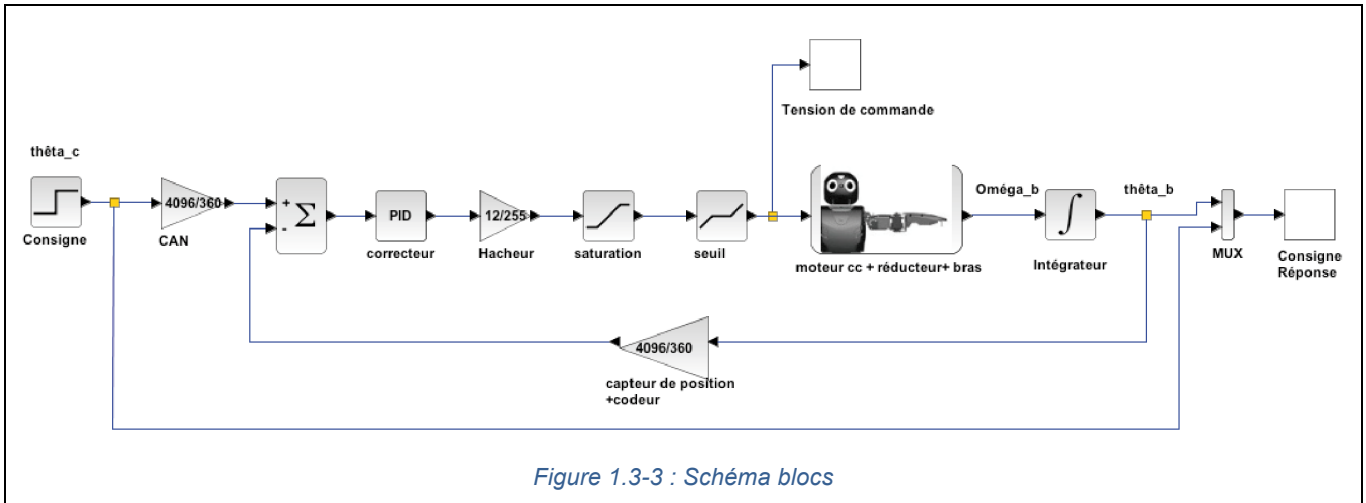


Figure 1.3-3 : Schéma blocs

### 1.3.3. Schéma cinématique, terminologie

Le schéma cinématique Figure 1.3-4 précise :

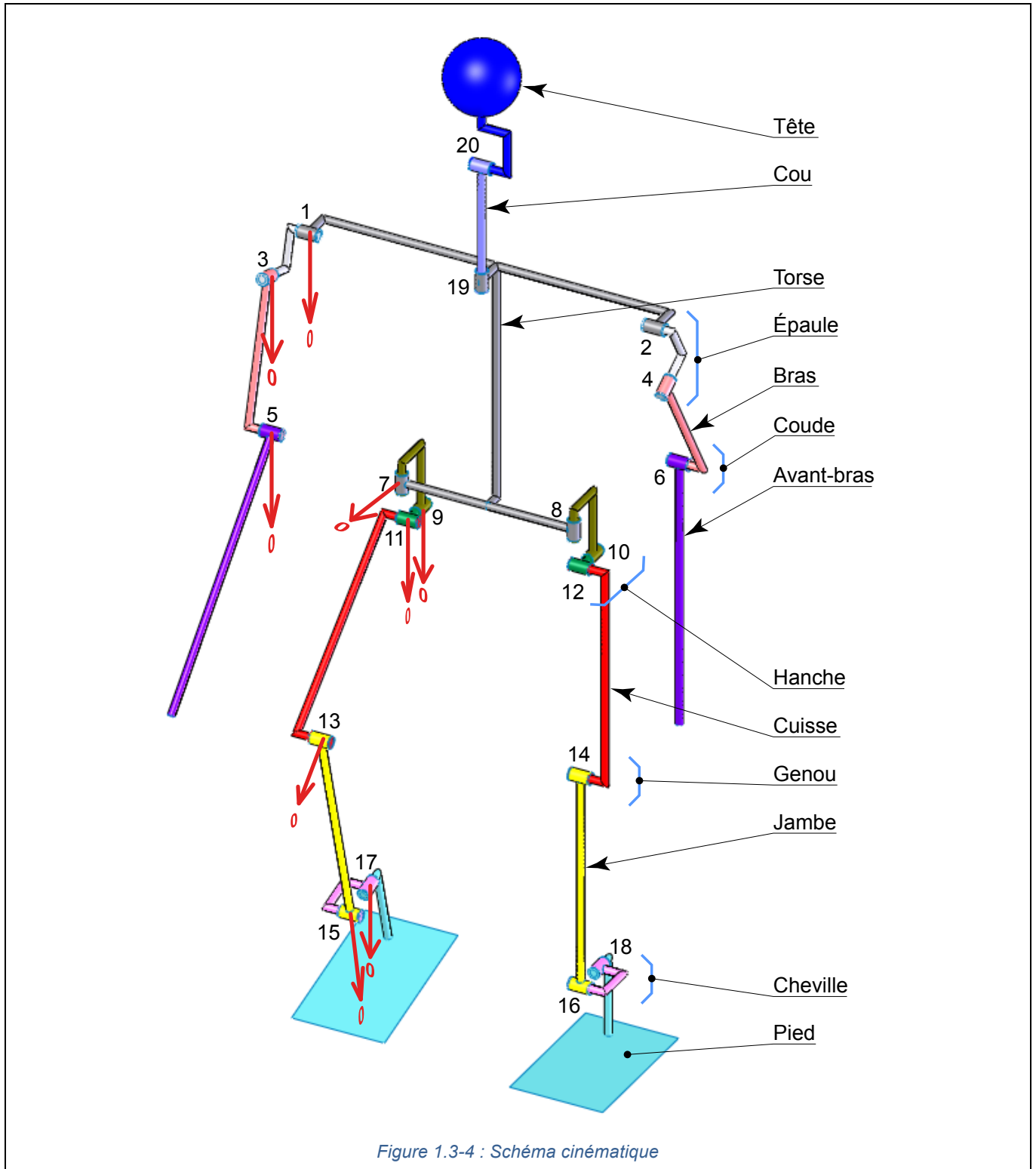
- ✓ le nom des différentes parties du robot, en accord avec la maquette numérique élaborée sous SolidWorks ;
- ✓ les numéros des différentes articulations, en accord avec les numéros des ID (identifiants) des servomoteurs ;
- ✓ les positions « zéro » des différentes articulations du côté droit (idem pour le côté gauche) qui sont matérialisées par les flèches rouges. Les sens de rotation sont donnés à la Figure 1.4-23.

Dans l'ensemble des programmes sources du kit de développement DARwin-OP, le nom des articulations associé à leur numéro est le suivant :

Numéro de l'articulation droite	Nom dans les programmes sources	Numéro de l'articulation gauche	Nom dans les programmes sources
1	ID_R_SHOULDER_PITCH	2	ID_L_SHOULDER_PITCH
3	ID_R_SHOULDER_ROLL	4	ID_L_SHOULDER_ROLL
5	ID_R_ELBOW	6	ID_L_ELBOW
7	ID_R_HIP_YAW	8	ID_L_HIP_YAW
9	ID_R_HIP_ROLL	10	ID_L_HIP_ROLL
11	ID_R_HIP_PITCH	12	ID_L_HIP_PITCH
13	ID_R_KNEE	14	ID_L_KNEE
15	ID_R_ANKLE_PITCH	16	ID_L_ANKLE_PITCH
17	ID_R_ANKLE_ROLL	18	ID_L_ANKLE_ROLL



Numéro de l'articulation	Nom dans les programmes sources
19	ID_HEAD_PAN
20	ID_HEAD_TILT





## 1.4. Description de la structure mécanique

### 1.4.1. Vues d'ensemble de DARwIn-OP avec et sans habillage

DARwIn-OP est constitué d'un squelette de tôles d'acier découpées, pliées et assemblées par éléments de visserie. L'ensemble est ensuite habillé à l'aide de capots en plastique moulés et fixés par vis. De ce fait, il est facilement démontable et observable.

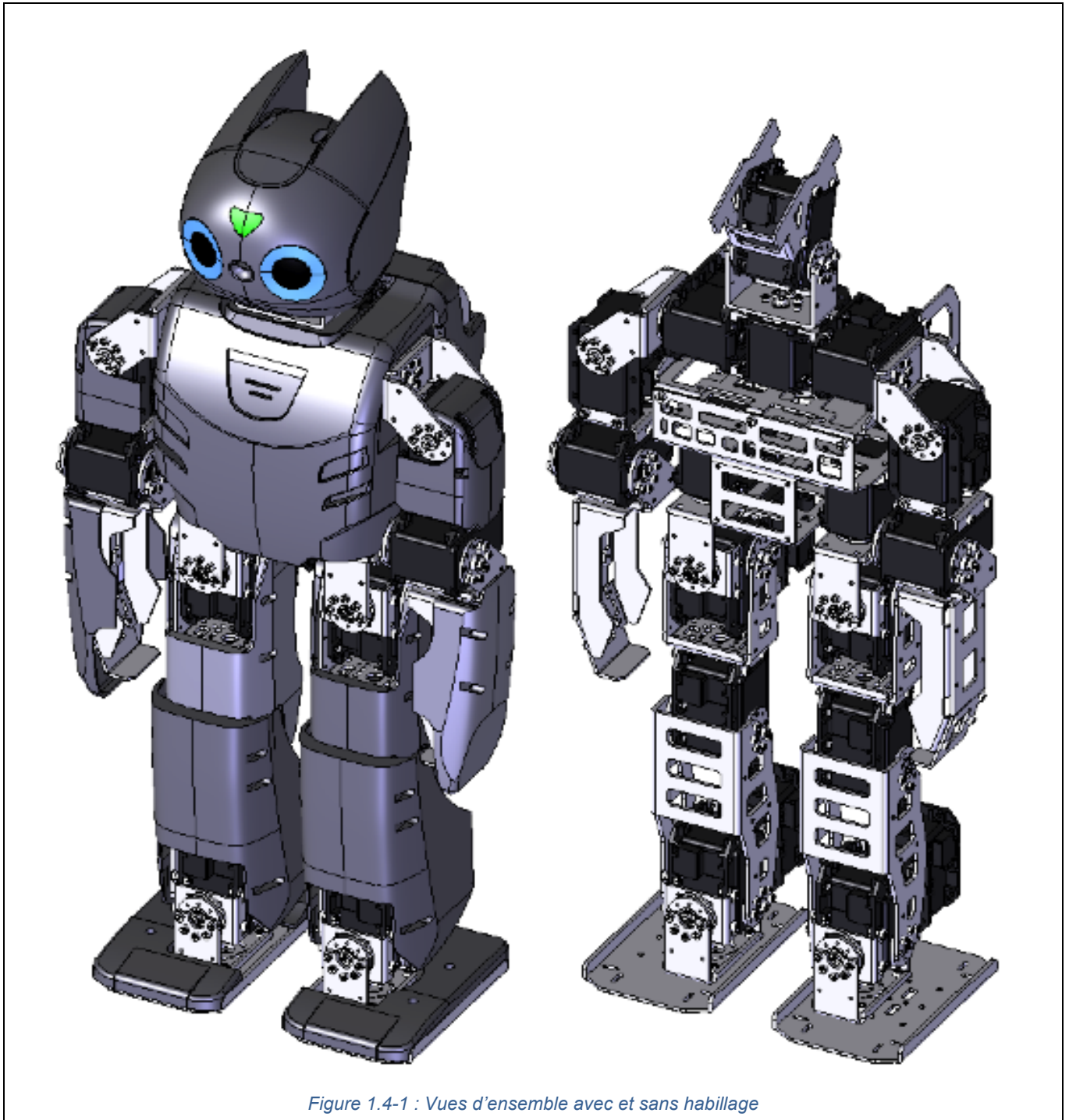


Figure 1.4-1 : Vues d'ensemble avec et sans habillage



## 1.4.2. Nomenclatures mécaniques

### 1.4.2.1. Squelette de DARwIn-OP

Seul le squelette de tôles est représenté dans la figure ci-dessous.

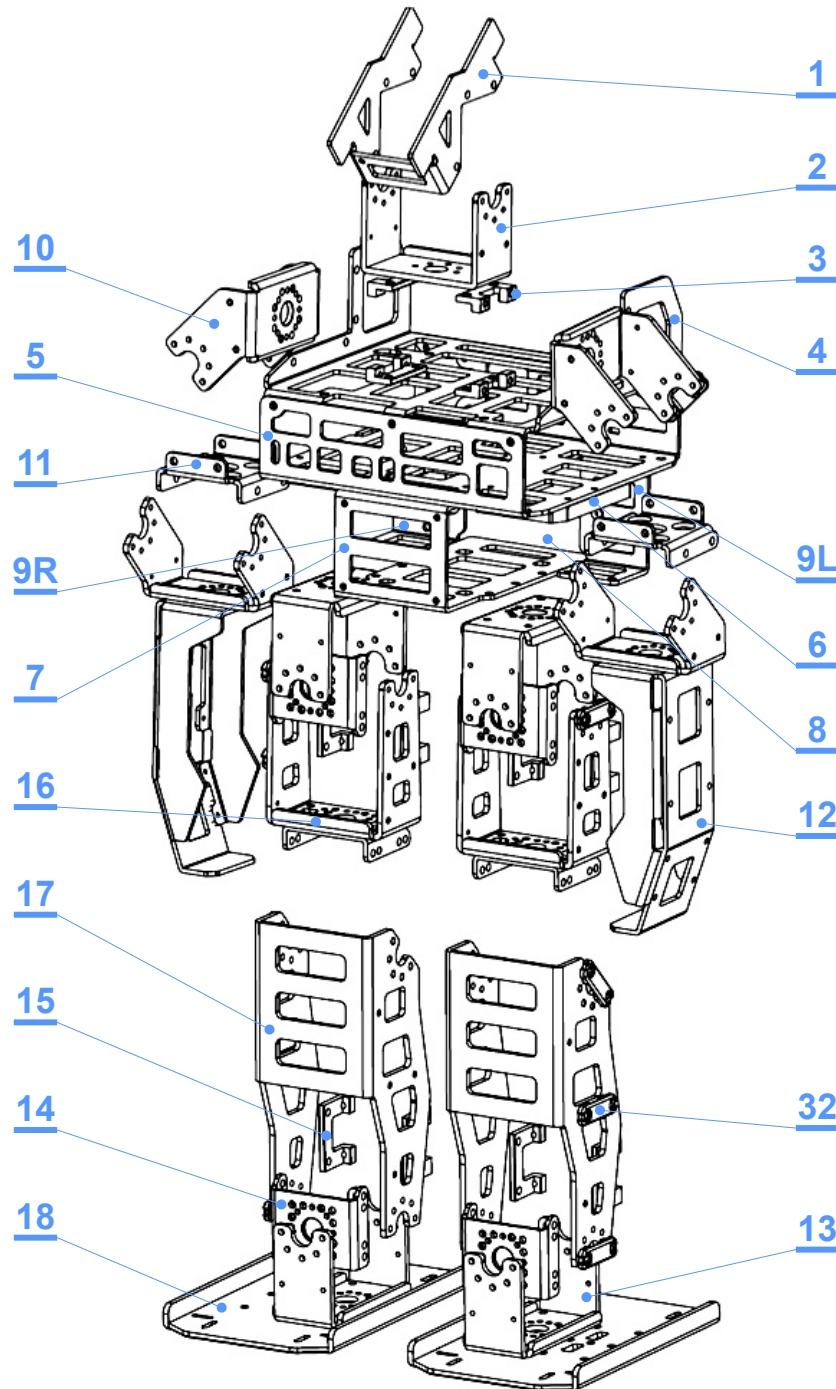


Figure 1.4-2 : Squelette seul





Repère	Nombre	Désignation	Repère dans SolidWorks
1	1	Support de tête	PR13_B01_BRKT_CAM
2	1	Support de moteur de cou	FR07_H104
3	4	Support de moteur	FR07_X101
4	1	Haut de torse	PR13_B01_FRM_PCB_MAIN_T
5	1	Bas de torse	PR13_B01_FRM_PCB_MAIN_B
6	2	Platine d'assemblage bassin	PR13_B_SPACER_PELV
7	1	Compartiment de la batterie	FR07_E130
8	1	Couvercle du compartiment de la batterie	PR13_B_FRM_CVR_BAT_ASM
9L	1	Support arrière gauche bassin	PR13_B_BRKT_PELV_L
9R	1	Support arrière droit bassin	PR13_B_BRKT_PELV_R
10	4	Support incliné de moteur	FR07_H102
11	2	Platine de moteurs double U	FR07_SC101
12	2	Avant-bras	FR07_E180
13	4	Support en U de moteur	FR07_H120
14	6	Platine de moteurs simple U	FR07_S101
15	8	Connecteur de moteur	FR07_X102
16	2	Tôle de cuisse	FR07_H132
17	2	Tôle de jambe	FR07_H133
18	2	Tôle de pied	FR07_E160
32	10	Support de câble	DC04_HOLDER_CABLE_PRM





### 1.4.2.2. Habillage de DARwIn-OP

Seuls les habillages sont représentés dans la figure ci-dessous.

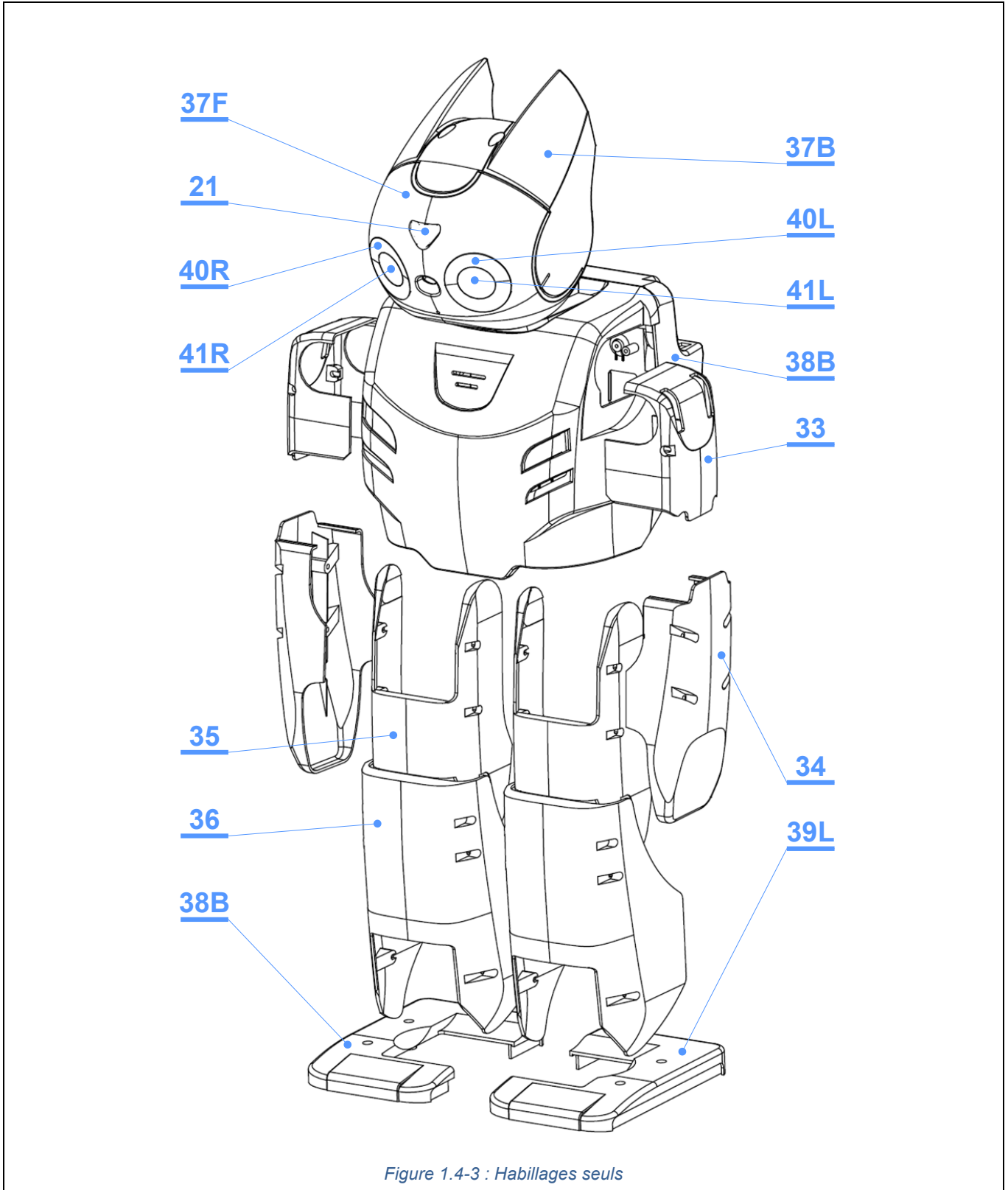


Figure 1.4-3 : Habillages seuls



Repère	Nombre	Désignation	Repère dans SolidWorks
21	1	Habillage LED de tête	PR13_B_WIN_LED
33	2	Habillage bras	33_PR13_B_CVR_ARM_U
34	2	Habillage avant-bras	34_PR13_B_CVR_ARM_L1
35	2	Habillage cuisse	35_PR13_B_CVR_THIGH
36	2	Habillage jambe	PR13_B_CVR_TIBIA1
37B	1	Habillage arrière tête	37B_PR13_B_CVR_HEAD_B
37F	1	Habillage avant tête	PR13_B_CVR_HEAD_F
38B	1	Habillage arrière torse	38B_PR13_B_CVR_BODY_B
38F	1	Habillage avant torse	38F_PR13_B_CVR_BODY_F
39L	1	Habillage pied gauche	PR13_B_CVR_FOOT_L1
39R	1	Habillage pied droit	PR13_B_CVR_FOOT_R1
40L	1	Habillage œil gauche	PR13_B_CVR_EYE_L
40R	1	Habillage œil droit	PR13_B_CVR_EYE_R
41L	1	Pupille gauche	PR13_B_CVR_PUPIL_L
41R	1	Pupille droite	PR13_B_CVR_PUPIL_R

#### 1.4.2.3. Agencement des servomoteurs

Seuls les servomoteurs et les pièces essentielles de leurs assemblages sont représentés dans la Figure 1.4-4.

On remarquera l'orientation relative des servomoteurs ainsi que leurs assemblages (liaisons encastremets) qui se font par leur carter :

- ✓ M1 et M2 sont fixés sur M19 ;
- ✓ M5 sur M3, de même que M6 sur M4 ;
- ✓ M9 sur M11, de même que M10 sur M12 ;
- ✓ M17 sur M15, de même que M18 sur M16.

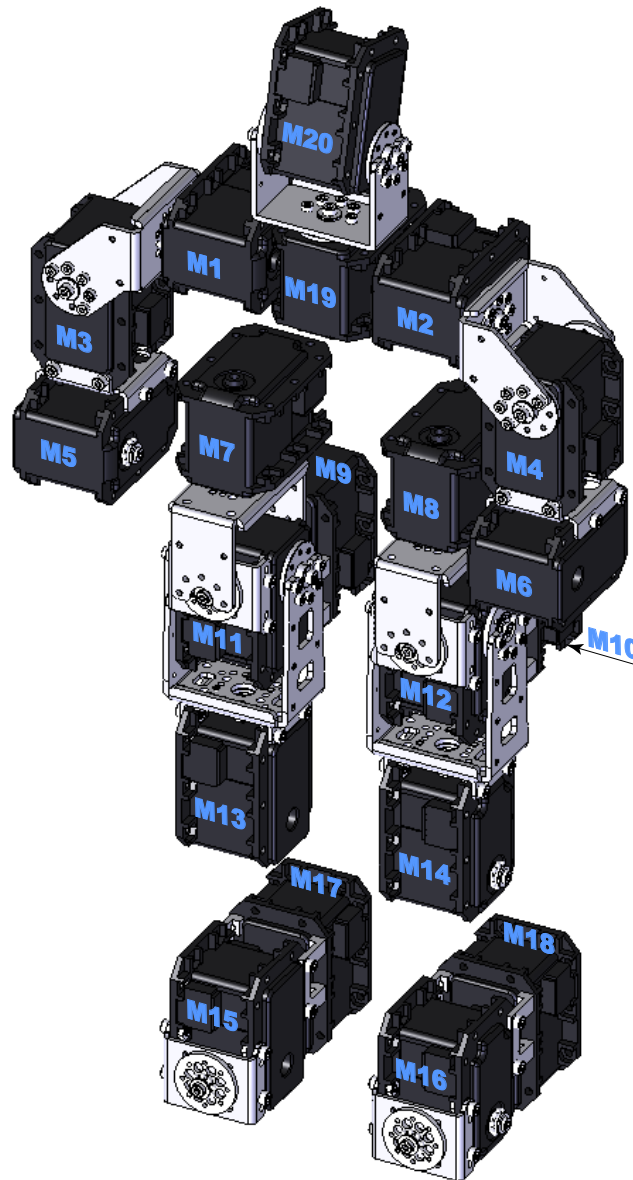


Figure 1.4-4 : Agencement des servomoteurs

### 1.4.3. Bloc Membres inférieurs

#### 1.4.3.1. Fonction

**Donner une apparence humanoïde et permettre au robot de marcher.**

#### 1.4.3.2. Présentation et terminologie

DARwIn-OP est équipé de deux membres inférieurs. Afin de visualiser plus facilement les éléments qui composent un membre inférieur, un sous-découpage a été effectué pour détailler, comme le montre la figure ci-dessous :

- ✓ la hanche et la cuisse ;
- ✓ le genou et la jambe ;
- ✓ la cheville et le pied.



**Membre inférieur droit**

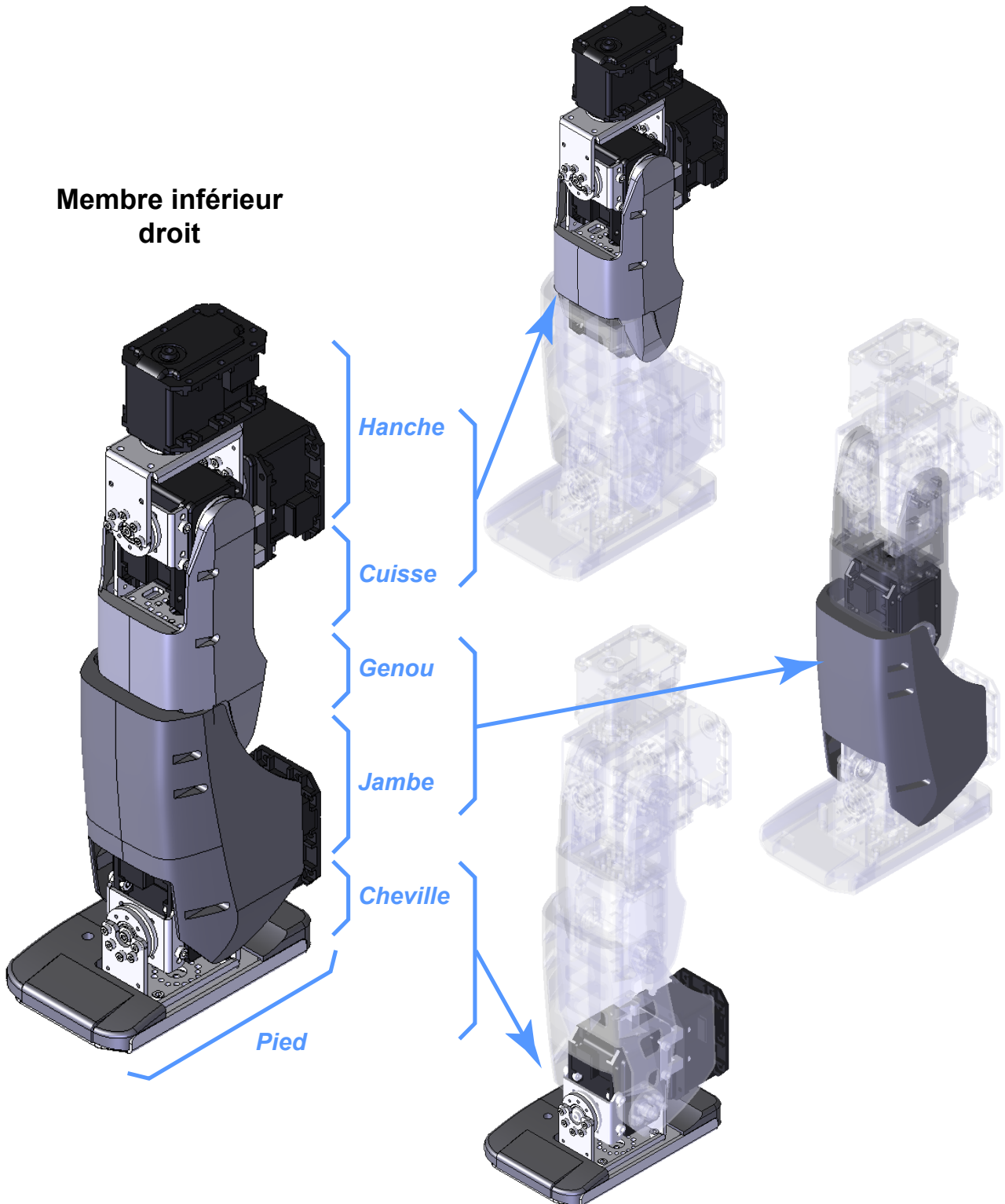


Figure 1.4-5 : Structure d'un membre inférieur

Chacun des deux membres inférieurs est mû par six servomoteurs MX28 : trois pour la hanche, un pour le genou et deux pour la cheville.



**1.4.3.3. Vue en perspective et éclaté de la hanche et de la cuisse droites**

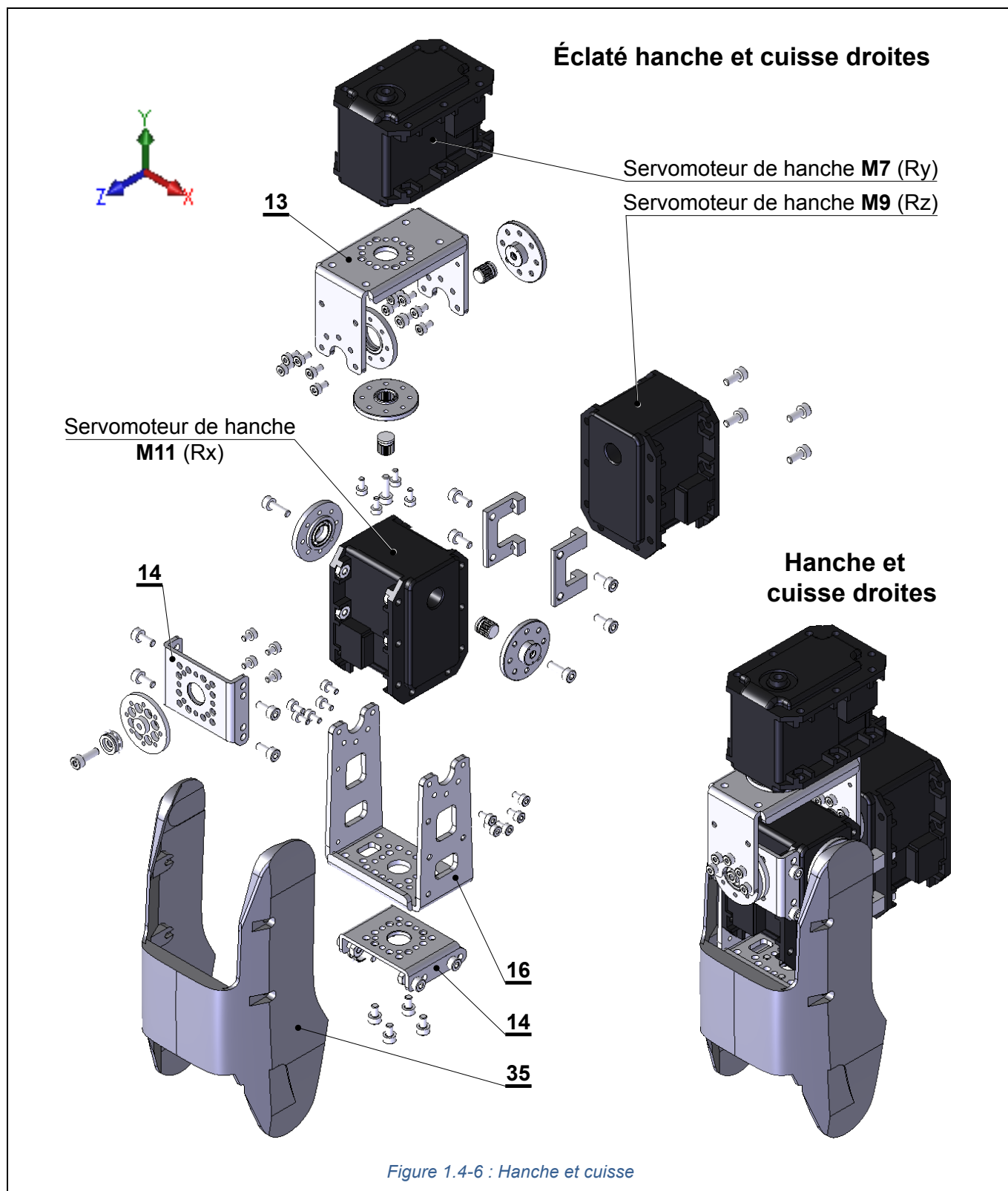


Figure 1.4-6 : Hanche et cuisse

**REMARQUE** – Le carter du servomoteur de hanche M7 (rotation Ry) est lié à la platine d'assemblage du bassin 6.



**1.4.3.4. Vue en perspective et éclaté de la jambe et du genou droits**

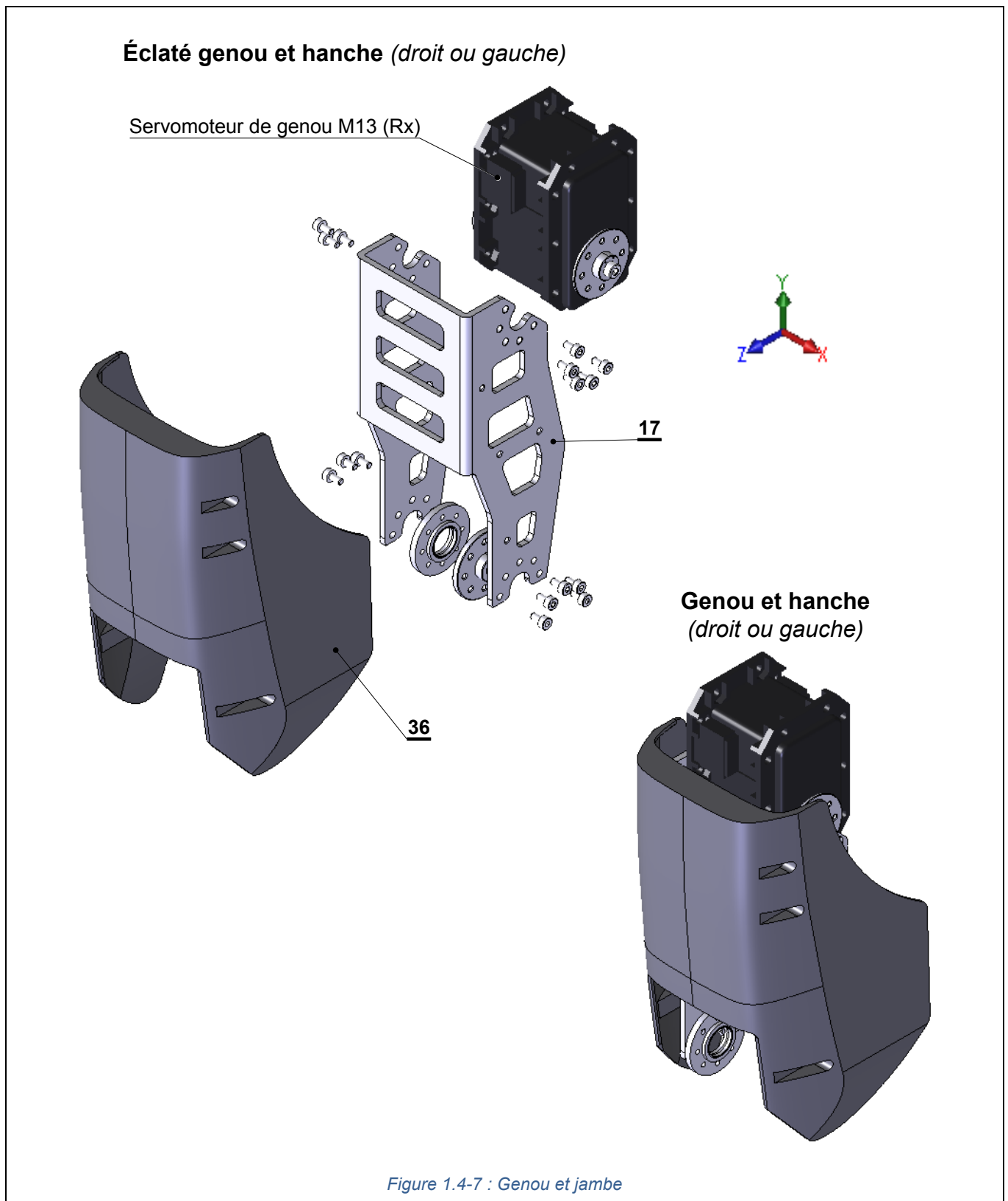
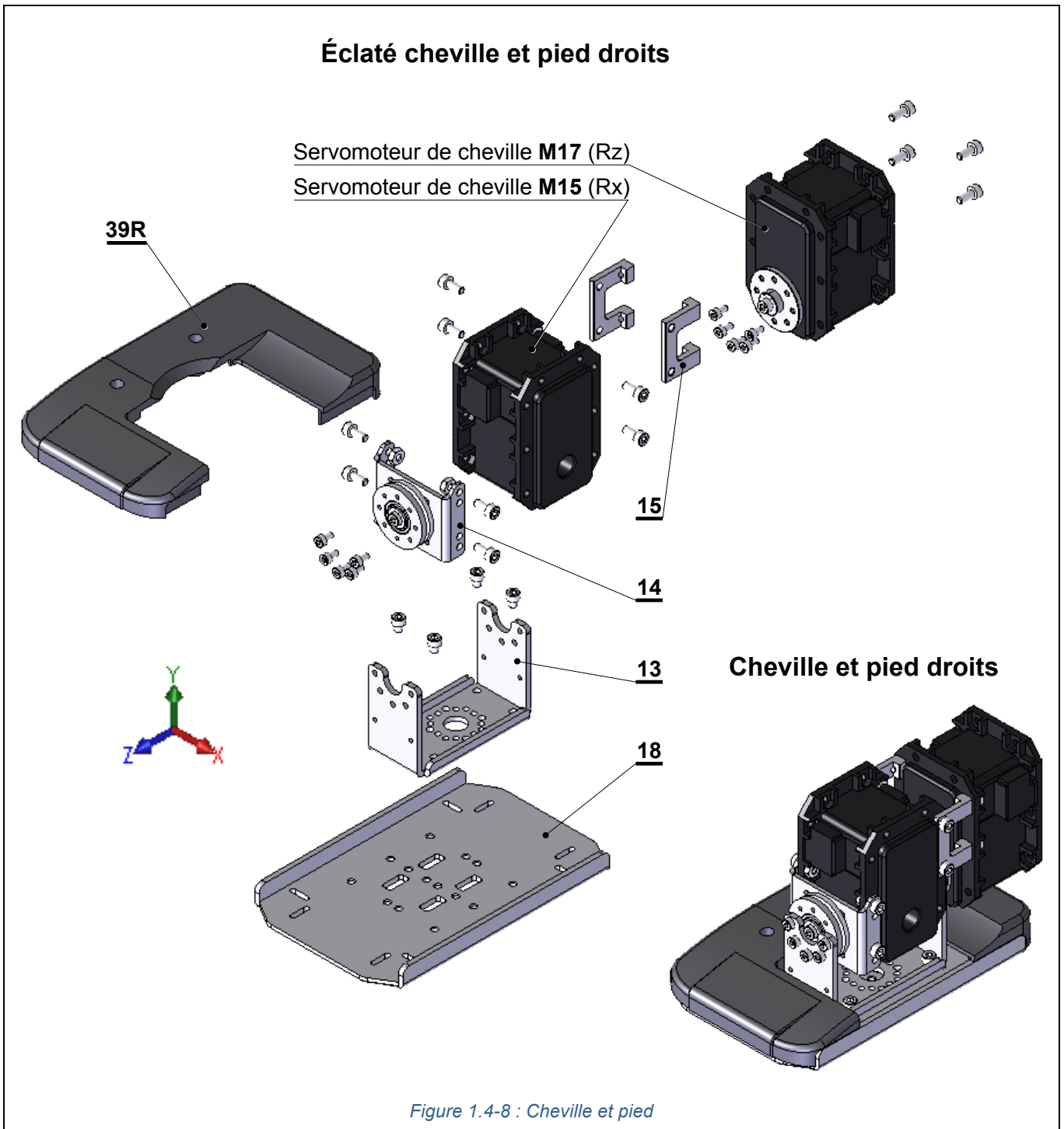


Figure 1.4-7 : Genou et jambe



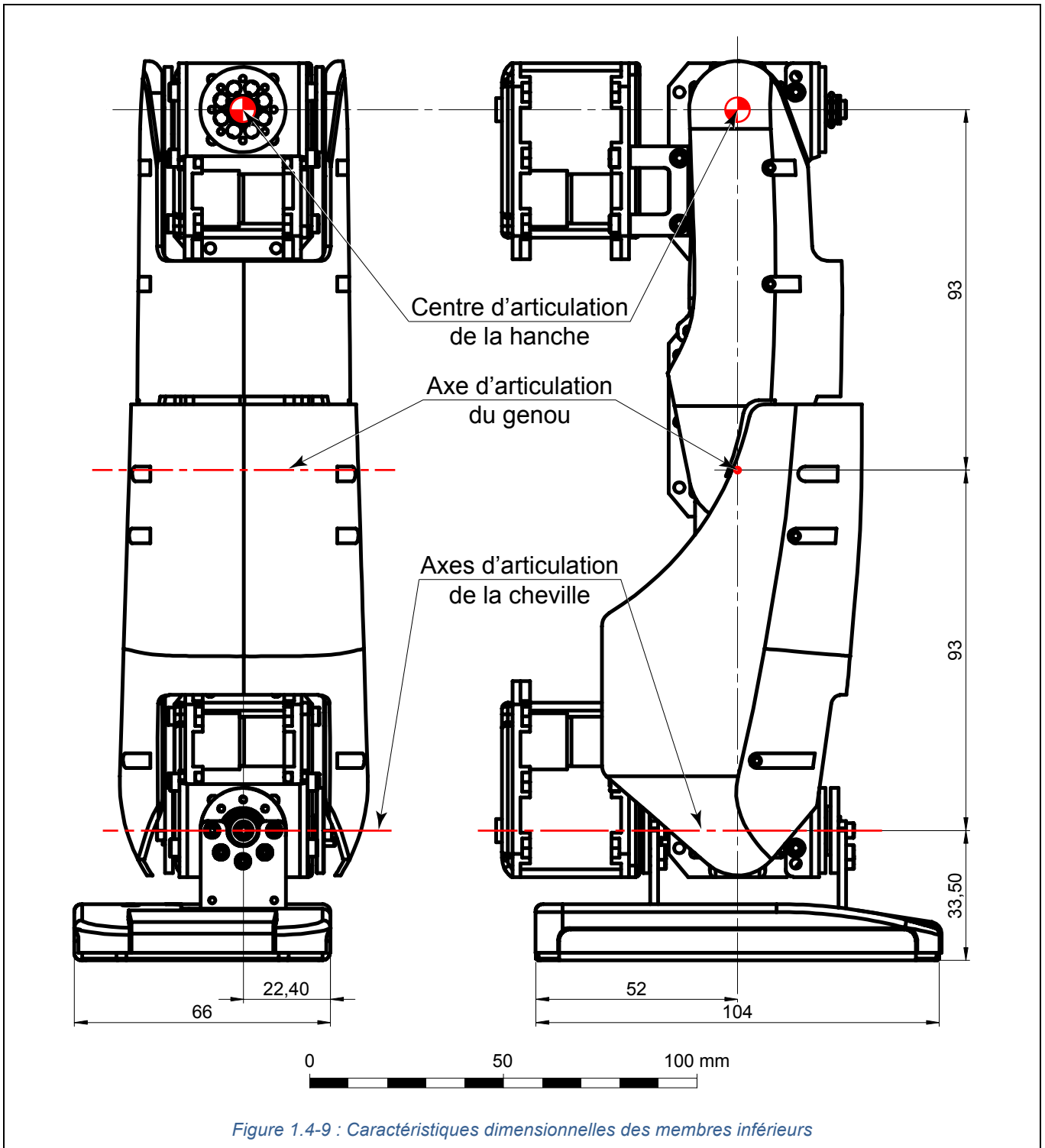


1.4.3.5. Vue en perspective et éclaté de la cheville et du pied droits





### 1.4.3.6. Caractéristiques dimensionnelles



## 1.4.4. Bloc Membres supérieurs

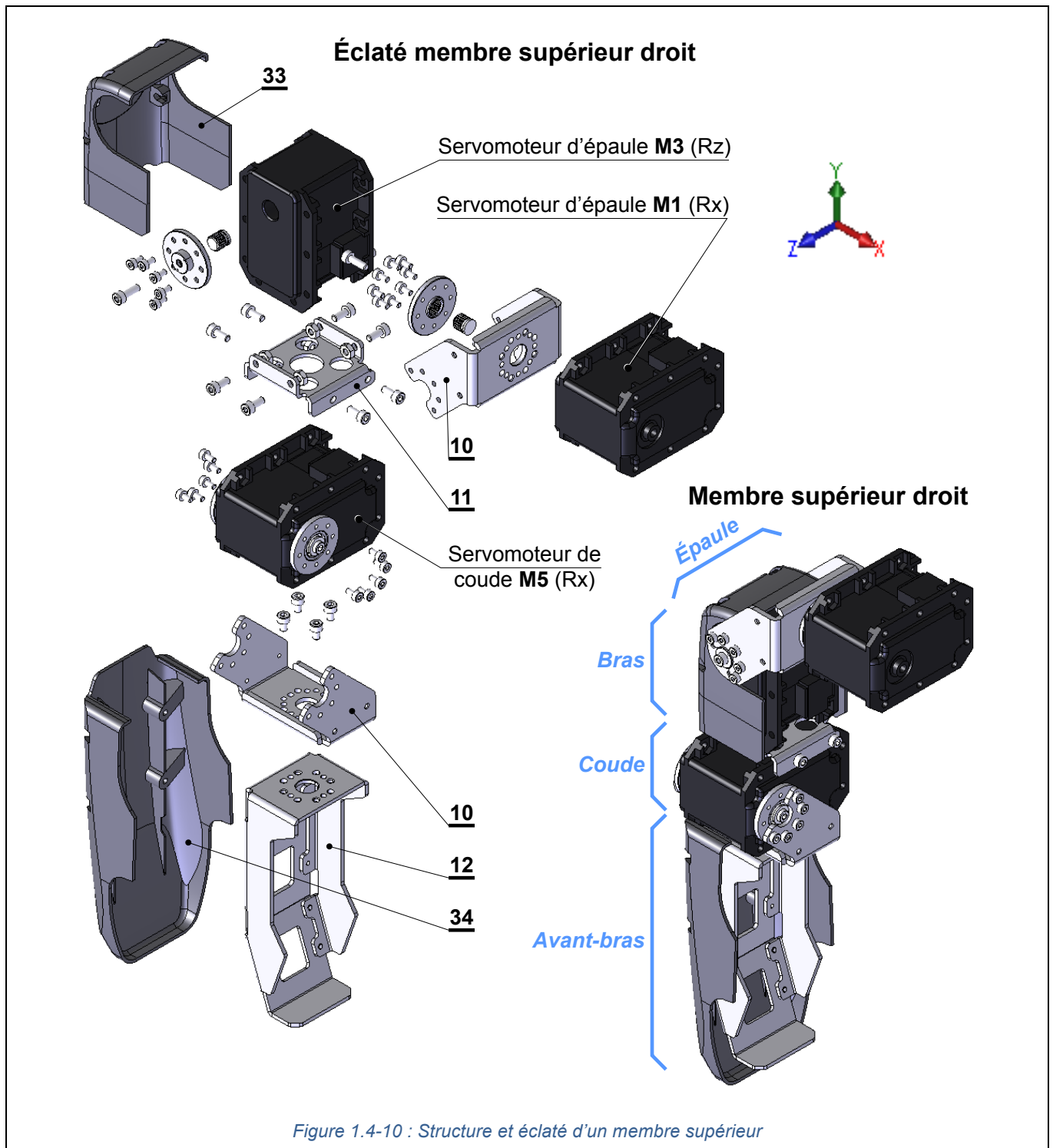
### 1.4.4.1. Fonction

**Donner une apparence humanoïde, porter les doigts optionnels et saisir des objets.**



**1.4.4.2. Présentation, vue en perspective et éclaté**

DARwIn-OP est équipé de deux membres supérieurs. Chacun d'eux est mû par trois servomoteurs MX28 : deux pour l'épaule (M1 et M3 du côté droit, M2 et M4 du côté gauche), un pour le coude (M5 du côté droit, M6 du côté gauche).



**REMARQUE** – Les carters des servomoteurs des épaules M1 et M2 (rotations Rx) sont liés au haut de torse 4 par l'intermédiaire de deux supports de moteur 3.



1.4.4.3. Caractéristiques dimensionnelles

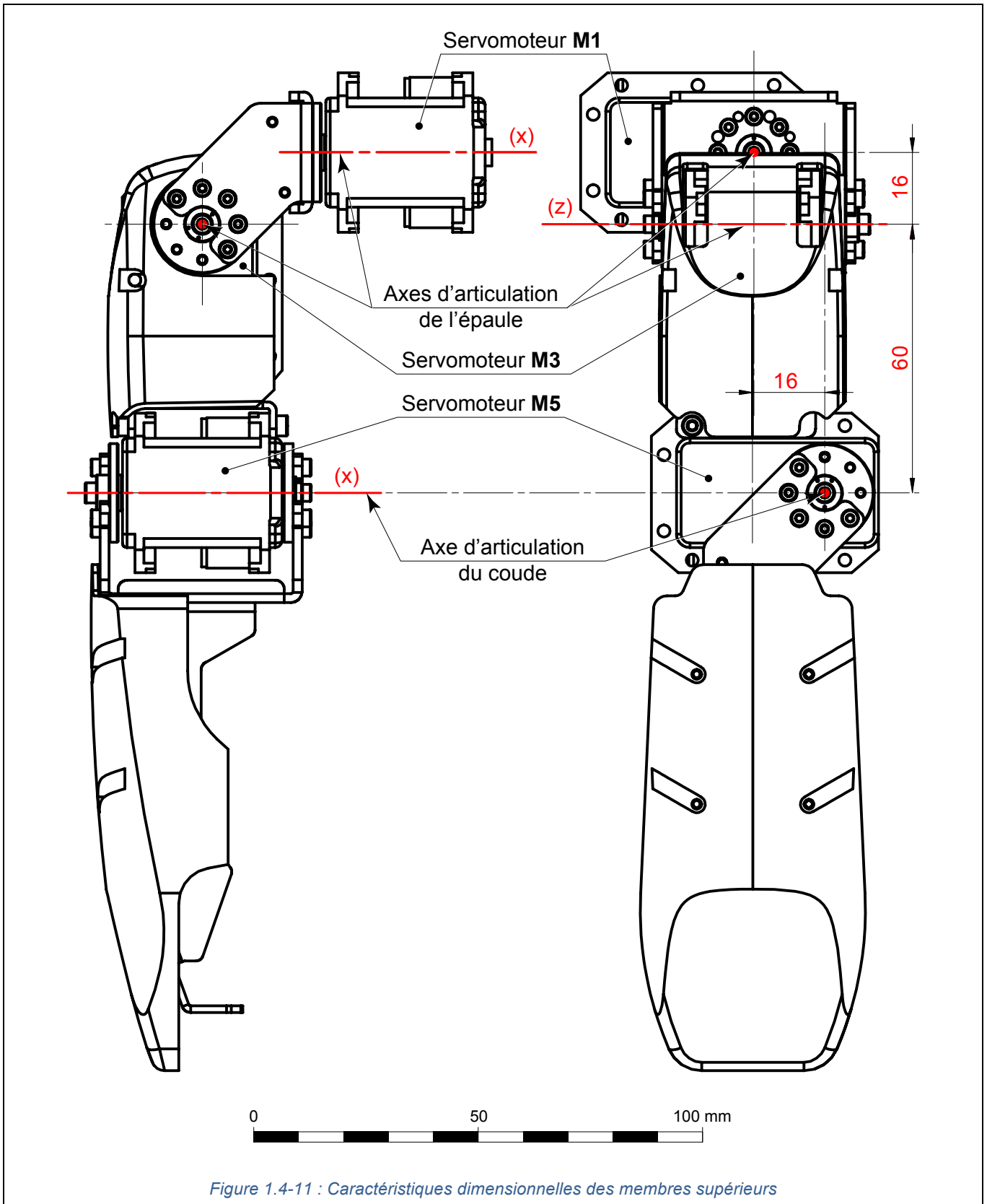


Figure 1.4-11 : Caractéristiques dimensionnelles des membres supérieurs

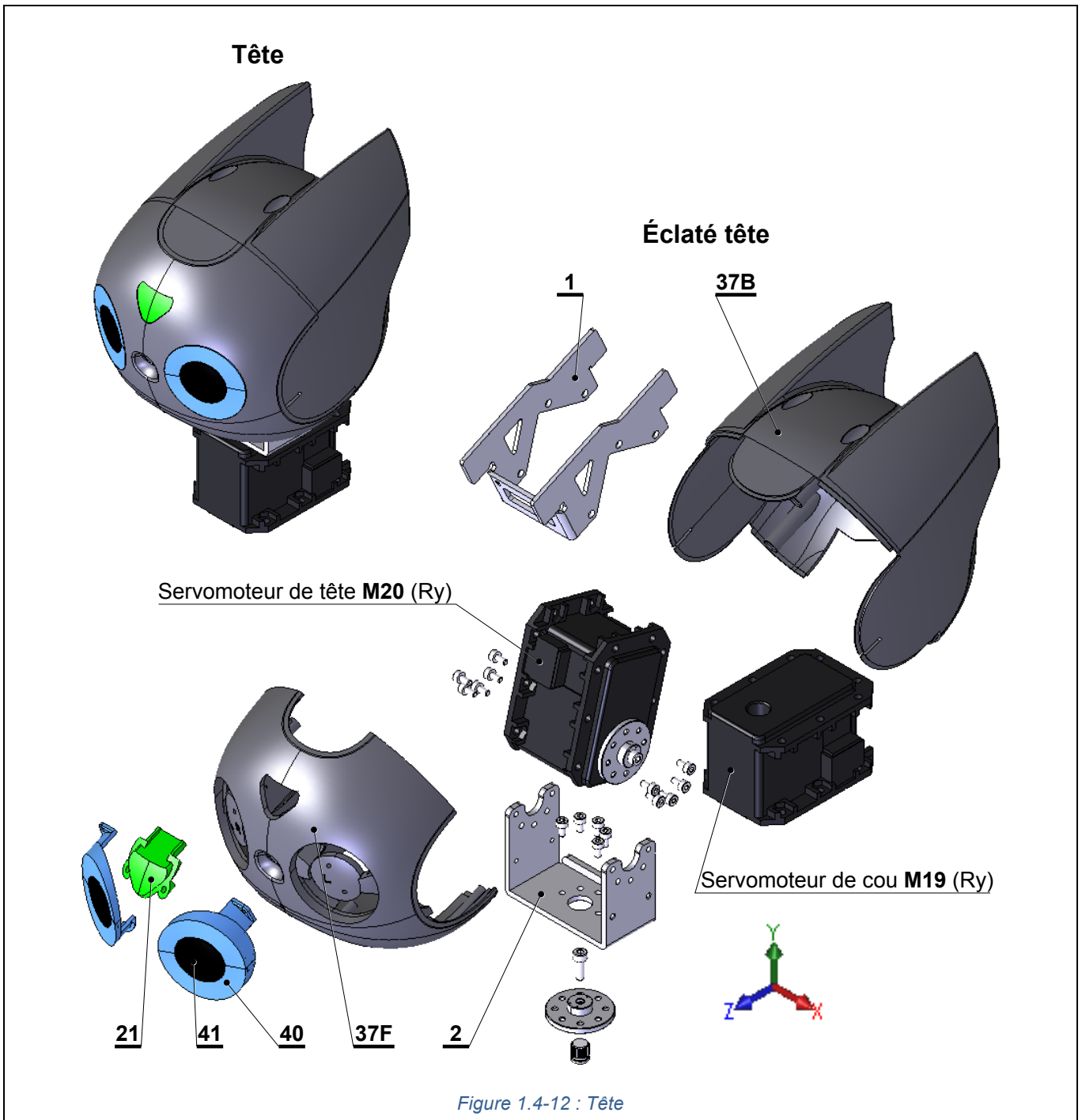


### 1.4.5. Bloc Tête

#### 1.4.5.1. Fonction

*Donner une apparence humanoïde et contenir les LEDs de tête et la caméra.*

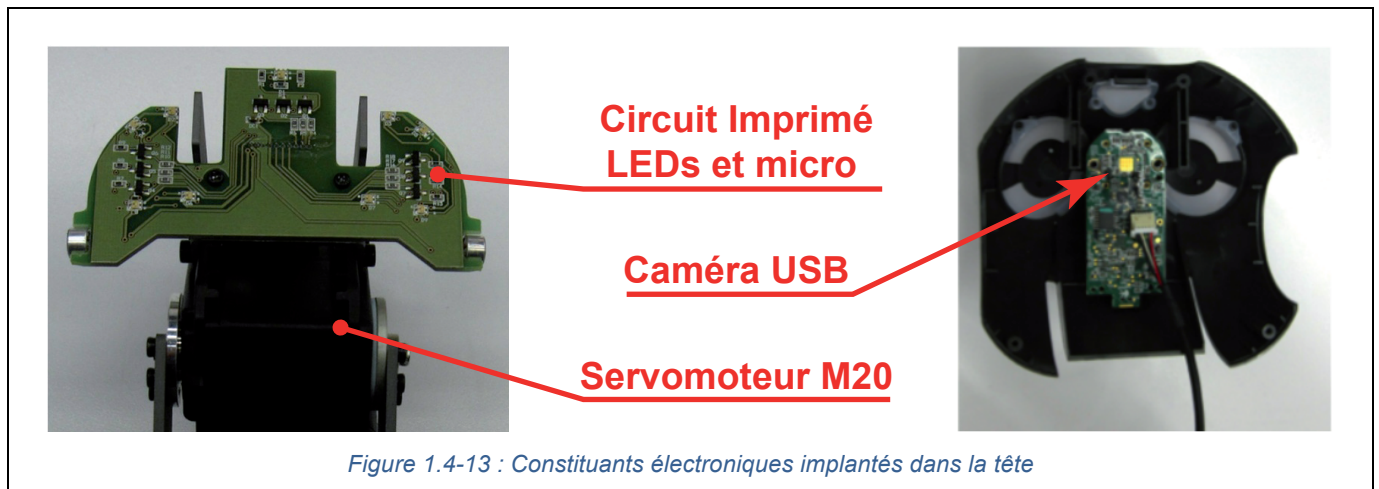
#### 1.4.5.2. Présentation



**REMARQUE** – L'arbre de sortie du servomoteur de cou M19 (rotation Ry) est lié au haut de torse 4.  
Le bloc tête sert de support à la caméra USB du robot. Les LEDs tricolores et les microphones sont



disposés sur un circuit imprimé solidaire du servomoteur M20, les signaux électriques sont issus de la carte CM-730.



## 1.4.6. Bloc Servomoteurs Dynamixel MX-28

### 1.4.6.1. Fonction

**Transformer l'énergie électrique en énergie mécanique en contrôlant la vitesse et la position de la sortie.**

### 1.4.6.2. Description

Les servomoteurs MX-28 Dynamixel sont les derniers nés de la firme ROBOTIS. Ces actuateurs à haute performance incluent toutes les fonctionnalités dont un roboticien pourrait avoir besoin pour contrôler des articulations. Le microcontrôleur intégré (*CORTEX-M3* de 32 bits) permet à la Série MX d'un nouveau concept de disposer des fonctions les plus avancées.

### 1.4.6.3. Caractéristiques des servomoteurs MX-28

- ✓ MCU : ST CORTEX-M3 ( ST32F103C8 @ 72MHZ,32BIT)
- ✓ Capteur de position (codeur) : Encodeur absolu sans contact (12BIT, 360 degrés) (*voir description au §0*)
- ✓ Couple : 2.3N.m (à 11.1V, 1.3A), 2.5N.m (à 12V, 1.4A), 3.1N.m (à 14.8V, 1.7A)
- ✓ Vitesse sans charge : 54 rpm (12V)
- ✓ Protocole : « propriétaire » transmission par paquets de données (Maître/esclave)
- ✓ Transmission : Asynchrone Série (8bit, 1stop, pas de Parité)...
- ✓ Lien (physique) : TTL ou RS485 (suivant version MX 28T ou MX 28R)
- ✓ Vitesse de communication : de 8000bps à 3Mbps
- ✓ Tension : 10 ~ 14.8V (tension 12V recommandé)
- ✓ Ratio du réducteur : 192,6:1
- ✓ ID : 254 Identificateurs (adresses) (0~253)
- ✓ Feedback : Position, Température, Charge, Input Voltage, etc.
- ✓ Consommation en veille : 100 mA

**Particularités des servomoteurs :** Ils intègrent un régulateur PID et quelques autres nouveautés décrites ci-après.

- ✓ Régulateur PID (Figure 1.4-14) :



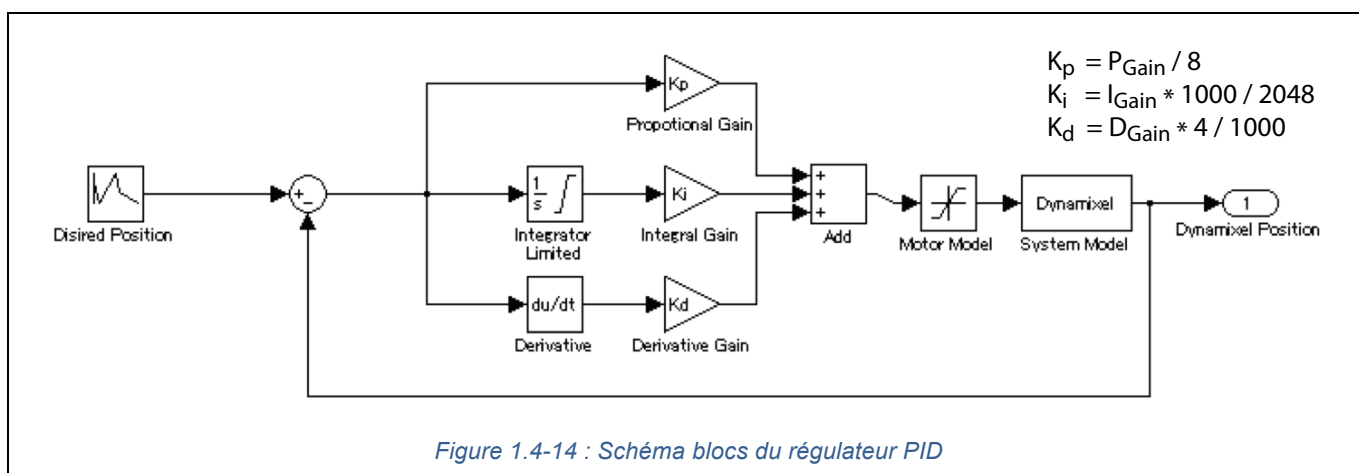


Figure 1.4-14 : Schéma blocs du régulateur PID

- ✓ Une grande **précision du contrôle de la position** grâce à son capteur de position angulaire (codeur absolu sur 12 bits) (voir description au §0).
- ✓ De nombreuses **fonctionnalités de feedback** sur les différentes variables physiques du servomoteur (position, vitesse, tension, température interne...). Une **alarme** en fonction de la température interne, du couple ou de la tension d'alimentation est déclenchée lorsque ces valeurs dépassent des limites définies. Le servomoteur peut automatiquement corriger la situation.
- ✓ Un **câblage facile** en Bus (daisy-chain), les servomoteurs Dynamixel ont un identifiant unique et communiquent par paquets de données numériques suivant un protocole propriétaire « très simple ». Les servomoteurs MX-28 sont déclinés en deux versions MX-28R « bus RS485 » et MX-28T « bus TTL ».

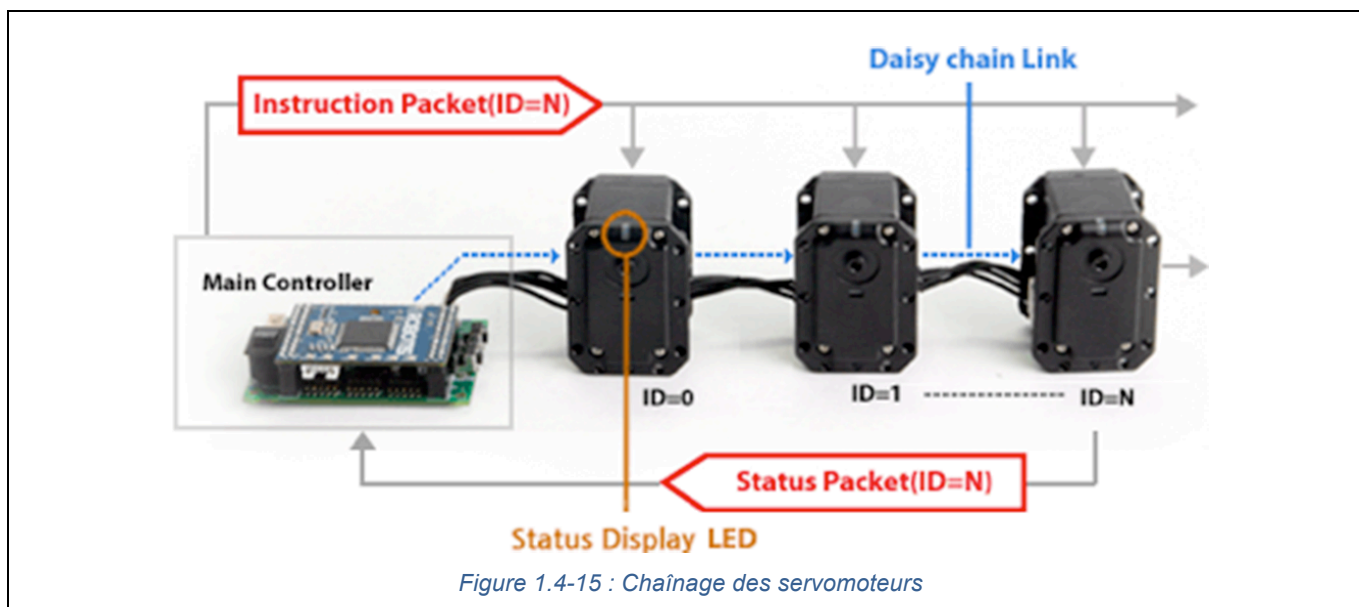



Figure 1.4-15 : Chaînage des servomoteurs

- ✓ Une **alarme** en fonction de la température interne, du couple ou de la tension d'alimentation est déclenchée lorsque ces valeurs dépassent des limites définies. Le servomoteur peut automatiquement corriger la situation.
- ✓ Établissement du **couple** : Le couple peut être établi sur 1023 pas allant du couple maximal à l'état de rotation libre.
- ✓ Une **LED d'état** sert d'affichage visuel sur l'état du Dynamixel.



#### 1.4.6.4. Caractéristiques techniques des moteurs

##### ■ Valeurs à la tension nominale

	Tension d'alimentation	12 V
	Vitesse à vide	10600 tr.mn <sup>-1</sup>
	Courant à vide	9,2 mA
	Vitesse nominale	8110 tr.mn <sup>-1</sup>
Couple nominal max permanent		3,67 mNm
Courant nominal max permanent		0,35 A
Couple de décrochage		15,5 mNm
Courant de démarrage		1,45 A
Rendement maximum		85 %

##### ■ Caractéristiques détaillées

Résistance aux bornes	8,3 Ohm
Inductance	0,205 mH
Constante de couple	10,7 mNm.A <sup>-1</sup>
Constante de vitesse	889 tr.mn <sup>-1</sup> .V <sup>-1</sup>
Pente vitesse/couple	687 tr.mn <sup>-1</sup> .(mNm) <sup>-1</sup>
Constante de temps électromécanique	6,25 ms
Inertie du rotor	0,868 gcm <sup>2</sup>

##### ■ Données thermiques

Température ambiante mini de fonctionnement	-30 °C
Température ambiante maxi de fonctionnement	+65 °C
Température max. rotor	+85 °C



■ **Données mécaniques**

Paliers	Paliers lisses
Vitesse limite	19000 tr.mn-1
Jeu axial minimum	0,05 mm
Jeu axial maximum	0,15 mm
Jeu radial	0,012 mm
Charge axiale maximum (dynamique)	0,8 N
Force de chassage maximum (statique)	35 N
Charge radiale maximum	1,4 N
à une distance de la face de :	5 mm

**1.4.6.5. Capteur de position (codeur)**

Le capteur de position intégré au servomoteur est un **codeur absolu, rotatif, magnétique à effet Hall et sans contact**. Il est constitué d'un circuit AS5045 (d'Austria Micro Systems) soudé sur la carte électronique et d'un aimant rotatif fixé sur l'arbre de sortie, comme le montre la figure ci-dessous. Son codage sur 12 bits (résolution de 4096 pas par tour) lui confère une précision de 0,0879°. La Figure 1.4-16 ci-dessous montre le principe du capteur de position et l'allure du signal généré avant d'être codé.

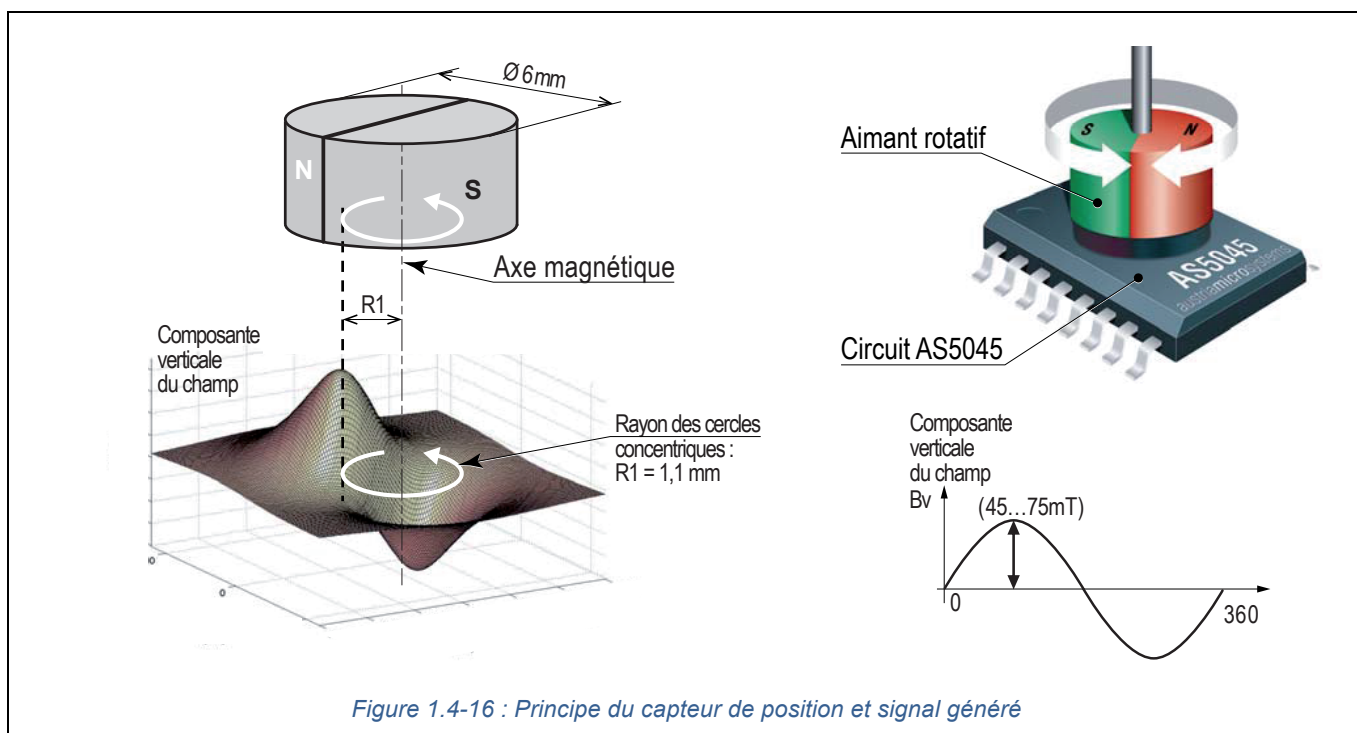


Figure 1.4-16 : Principe du capteur de position et signal généré



**1.4.6.6. Plans d'ensemble et schéma**

**■ Dessin d'ensemble**

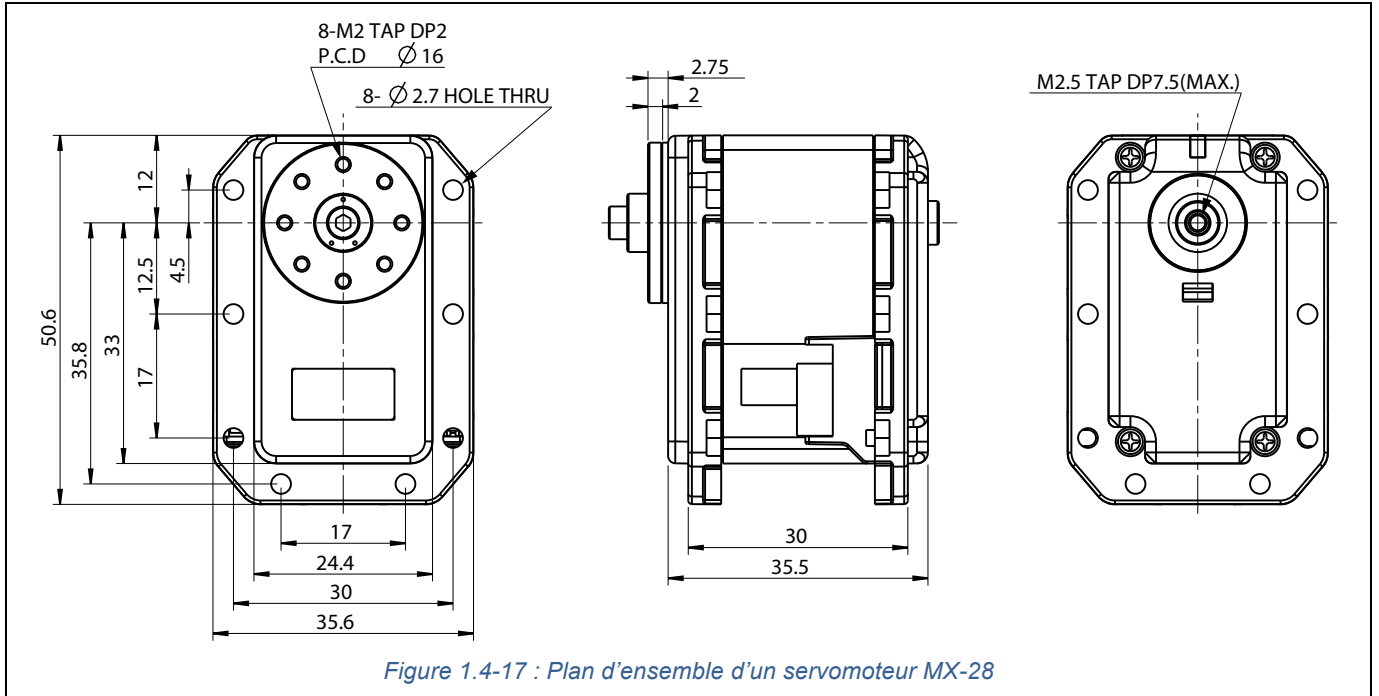


Figure 1.4-17 : Plan d'ensemble d'un servomoteur MX-28

**■ Vue éclatée**

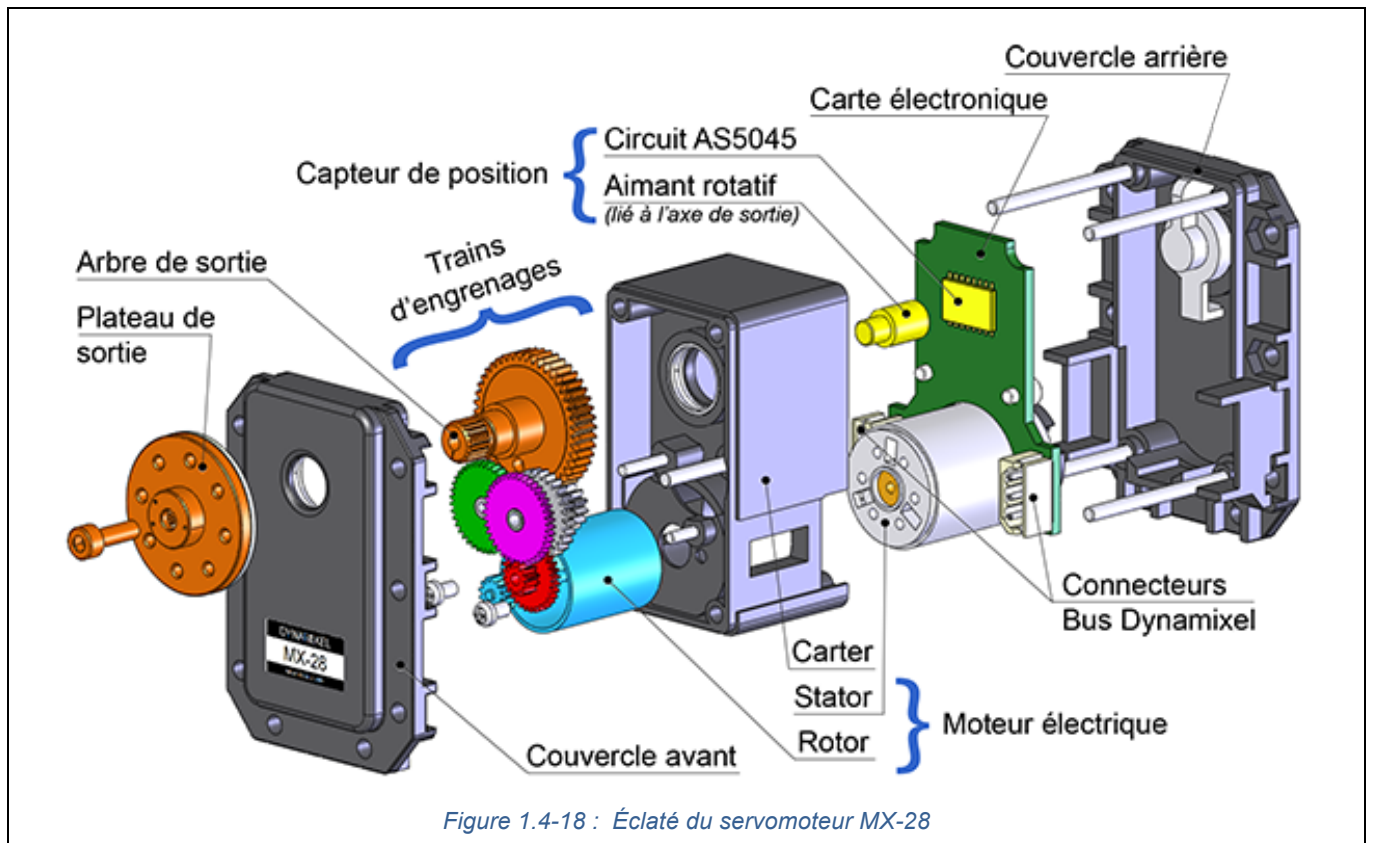


Figure 1.4-18 : Éclaté du servomoteur MX-28



■ **Diagramme de définition de blocs de structure (BDD)**

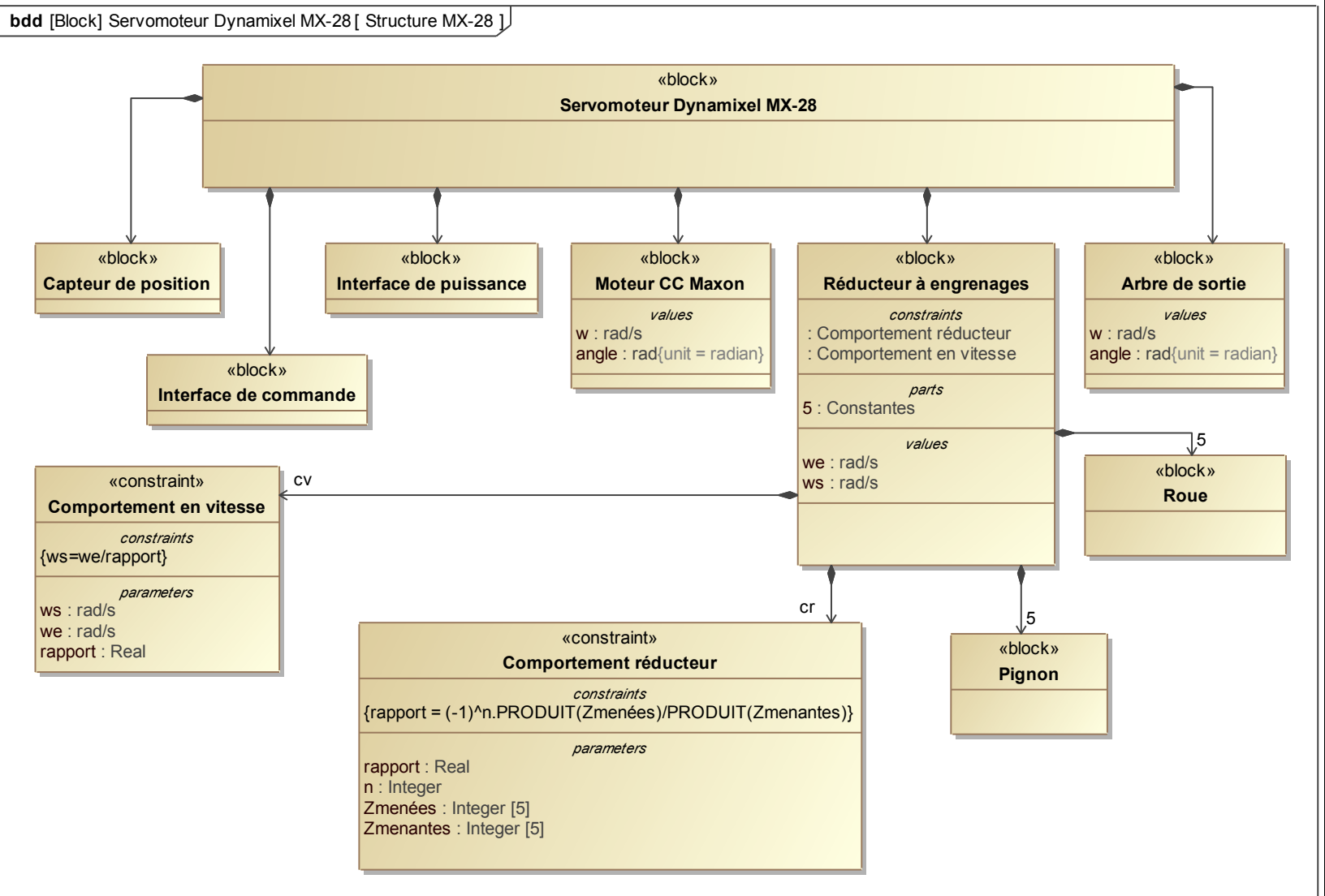


Figure 1.4-19 : Bloc et ports du servomoteur MX-28





■ *Vue en perspective des trains d'engrenages seuls*

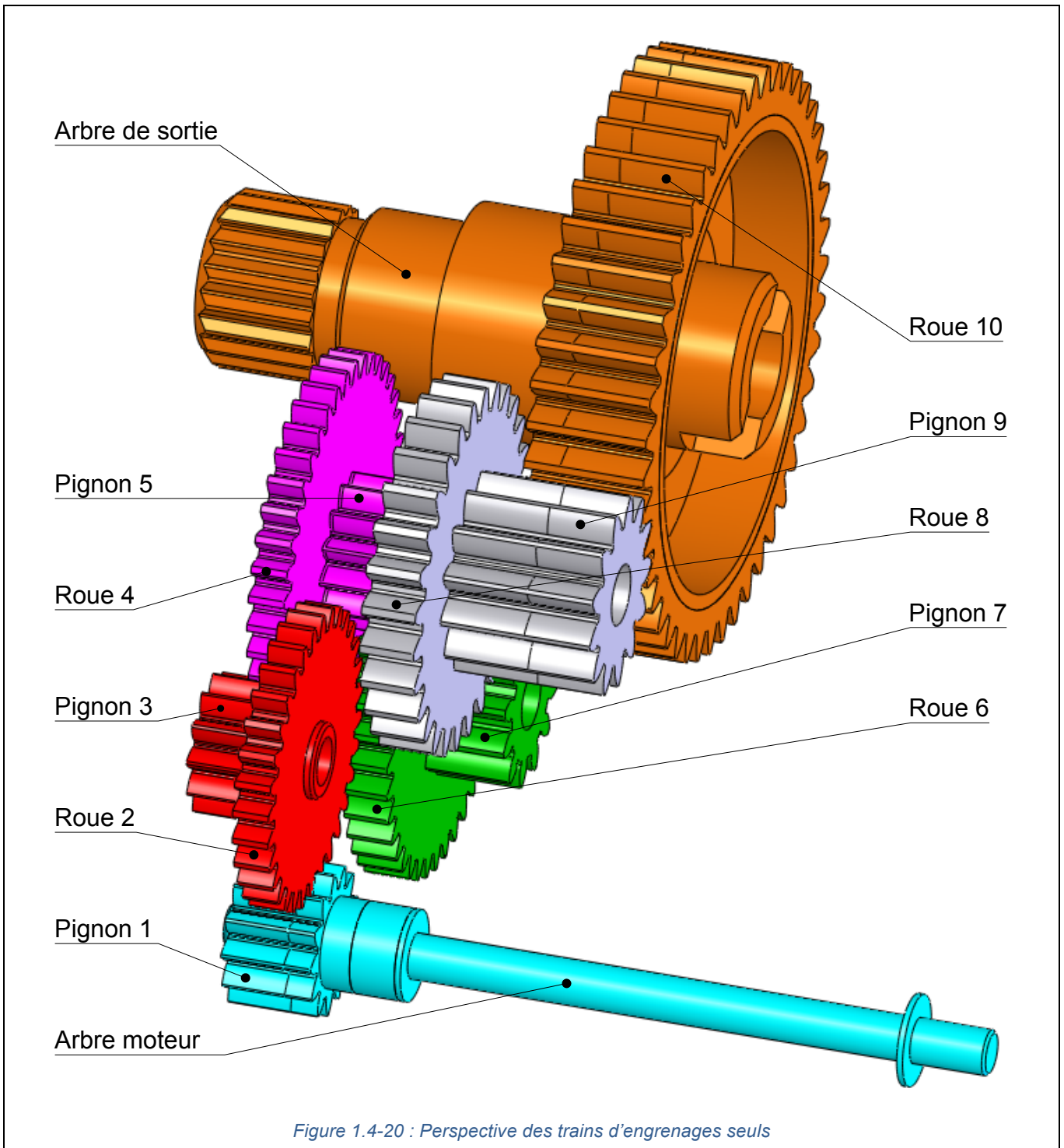


Figure 1.4-20 : Perspective des trains d'engrenages seuls

**REMARQUES**

- ✓ Les pignons et les roues de la même couleur sont solidaires.
- ✓ Le pignon ( $2 \cdot i - 1$ ) et la roue ( $2 \cdot i$ ), pour  $i = 1$  à 5, engrènent ensemble et constituent le train numéro ( $i$ ). Par exemple, le train n°3 est constitué du pignon 5 et de la roue 6.





■ **Schéma plan développé des trains d'engrenages**

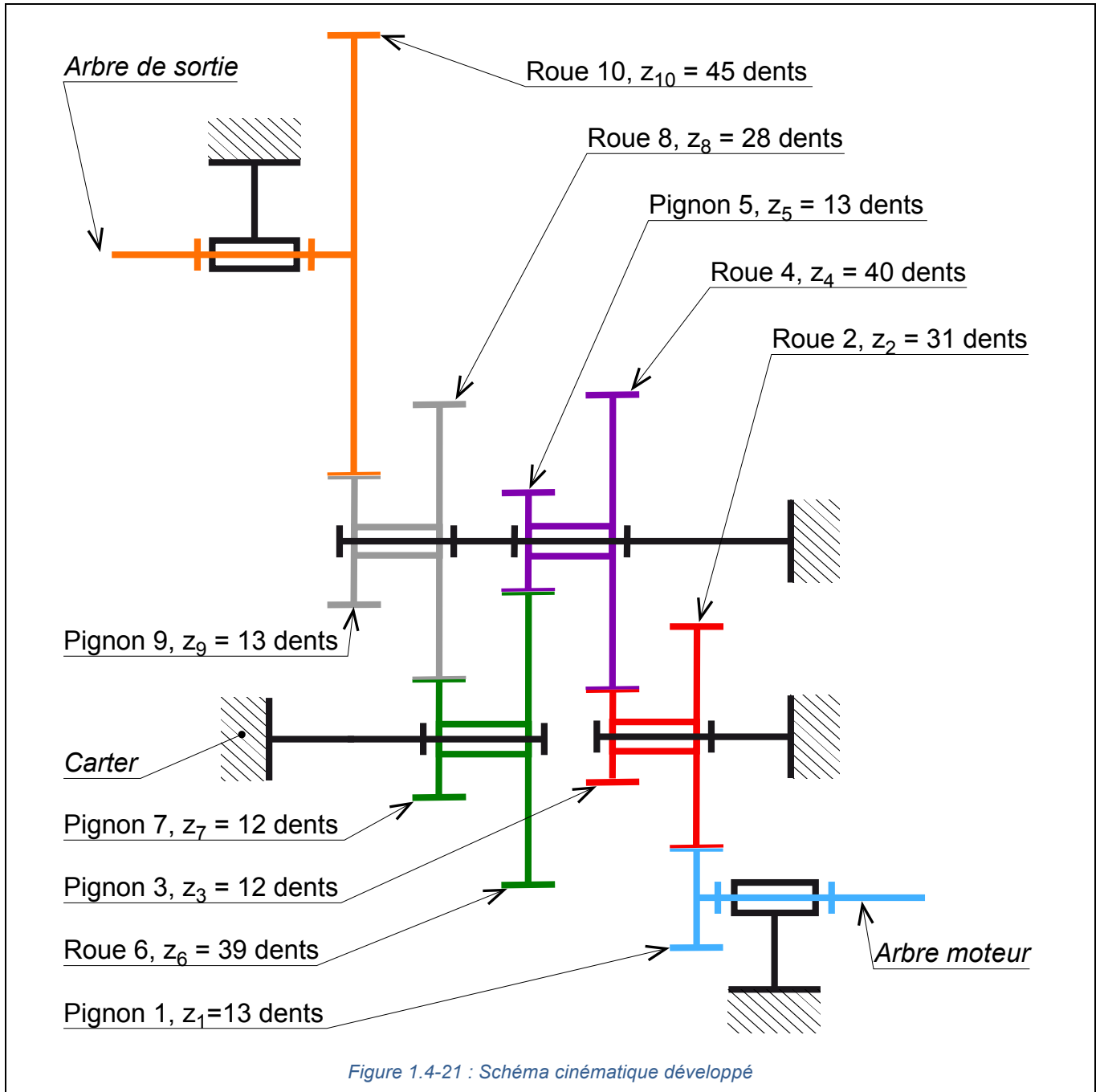


Figure 1.4-21 : Schéma cinématique développé

■ **Rapport de transmission (norme NF E 23-001)**

$$i = \frac{\omega_1}{\omega_{10}} = (-1)^n \frac{\prod z_{menées}}{\prod z_{menantes}} = (-1)^n \frac{z_2 \cdot z_4 \cdot z_6 \cdot z_8 \cdot z_{10}}{z_1 \cdot z_3 \cdot z_5 \cdot z_7 \cdot z_9}$$

$$i = (-1)^5 \frac{31 \times 40 \times 39 \times 28 \times 45}{13 \times 12 \times 13 \times 12 \times 13} = -192,6$$

L'arbre de sortie tourne moins vite que l'arbre moteur : c'est donc un ensemble **réducteur de vitesse**.



■ **Diagramme paramétrique du servomoteur MX-28**

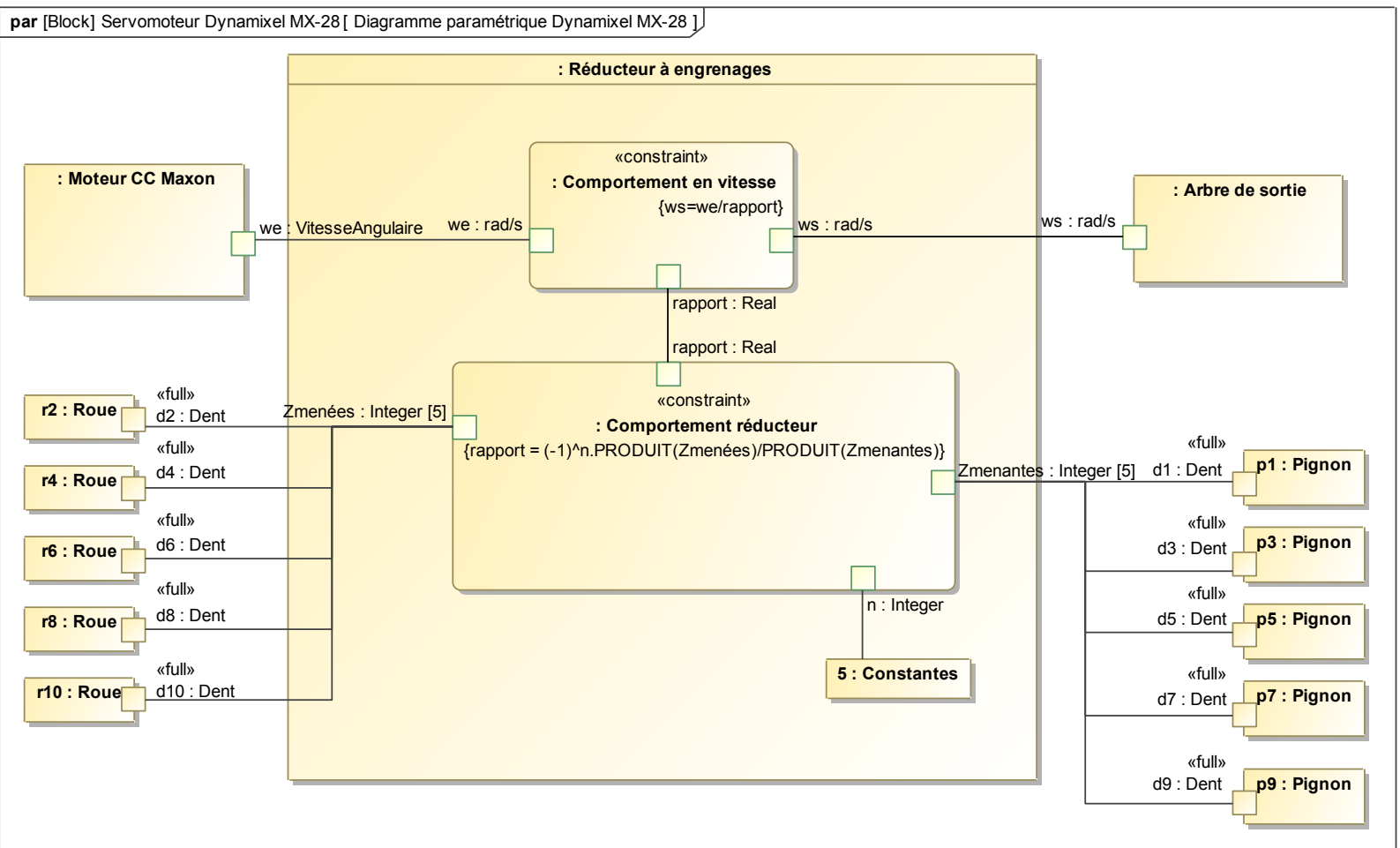


Figure 1.4-22 : Diagramme paramétrique du servomoteur MX-28



### 1.4.6.7. Paramètres de pilotage des servomoteurs

Pour pouvoir piloter individuellement un servomoteur ou pour en prélever des informations, il est nécessaire de connaître son numéro d'ID (IDentifiant).

De même, pour envoyer une consigne de position de l'arbre de sortie, il est nécessaire de connaître la position « zéro » (Figure 1.3-4) et la variation des angles en fonction du sens de rotation, ce que montre la Figure ci-dessous.

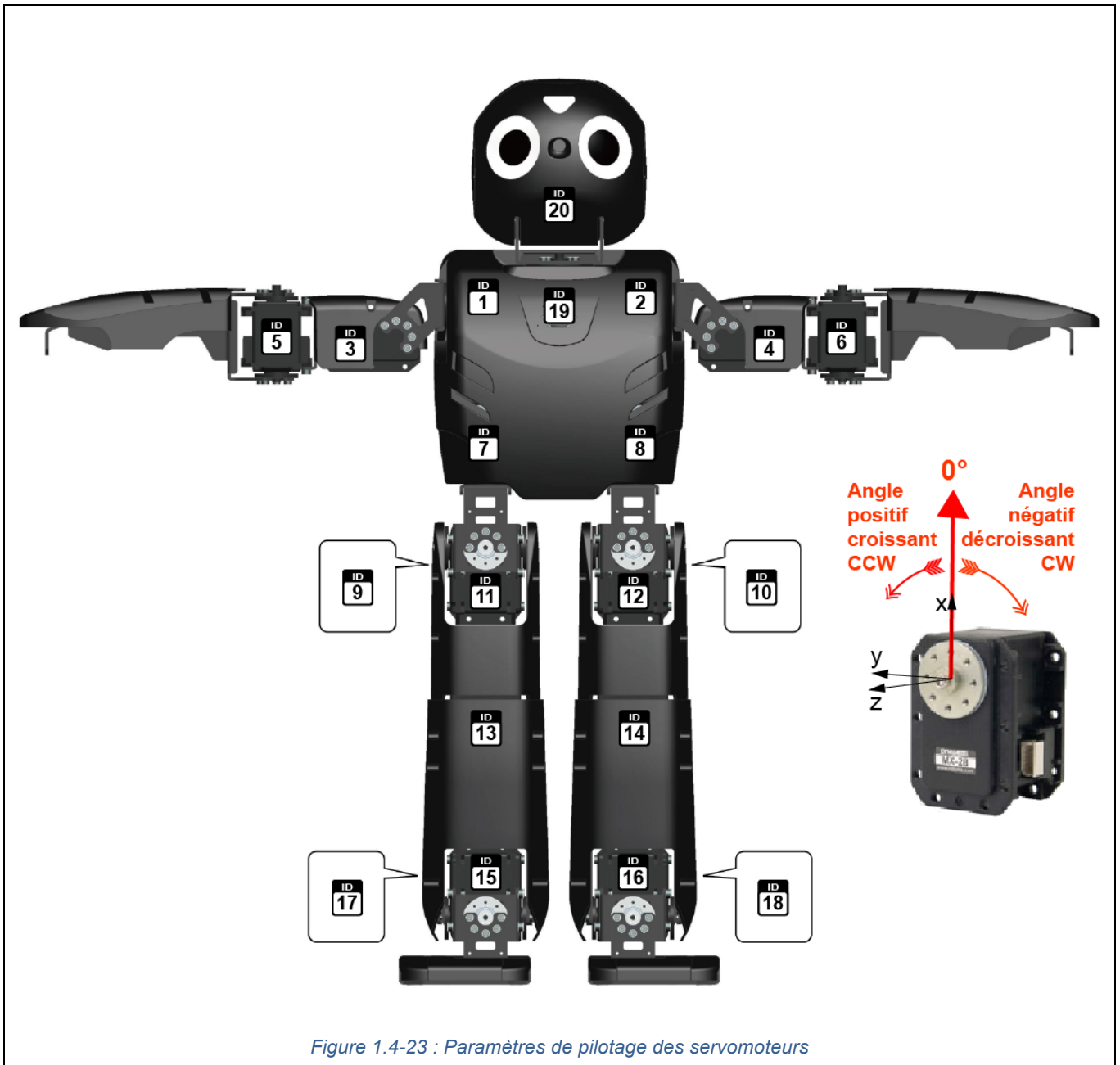


Figure 1.4-23 : Paramètres de pilotage des servomoteurs

### 1.4.6.8. Connexions des servomoteurs

Étant donné le nombre important de servomoteurs à contrôler et pour des commodités de câblage, les liaisons électriques sont réparties sur 5 connecteurs d'interface comme le montre la Figure 1.4-24.



- ✓ Liaison tête : M20 → M19
- ✓ Liaison bras droit : M5 → M3 → M1
- ✓ Liaison bras gauche : M6 → M4 → M2
- ✓ Liaison jambe droite : M17 → M15 → M13 → M11 → M9 → M7
- ✓ Liaison jambe gauche : M18 → M16 → M14 → M12 → M10 → M8

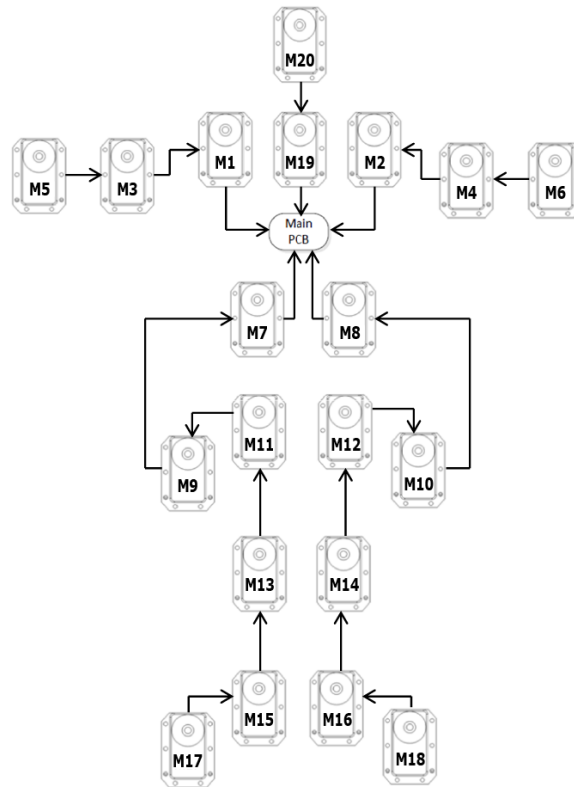


Figure 1.4-24 : Connexions des servomoteurs

La Figure 1.4-25 suivante montre les connecteurs situés sur la carte secondaire CM-730 ainsi que le schéma de l'interface.

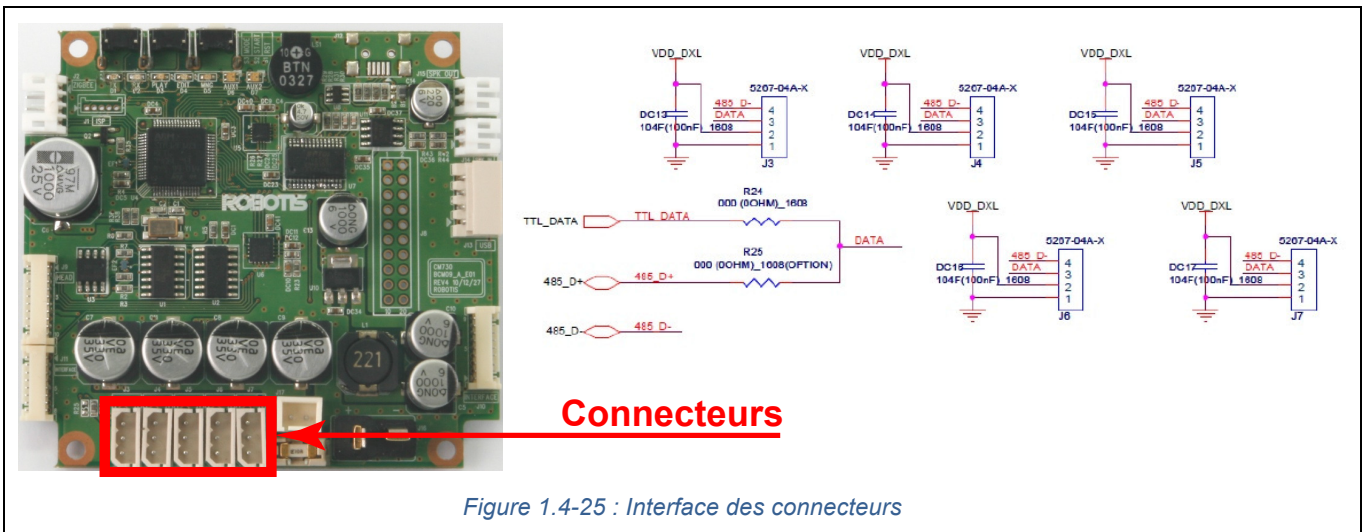


Figure 1.4-25 : Interface des connecteurs



## 1.5. Description de la structure de la commande

### 1.5.1. Bloc Contrôleur principal FitPC (carte mère, ou main board, ou main controller)

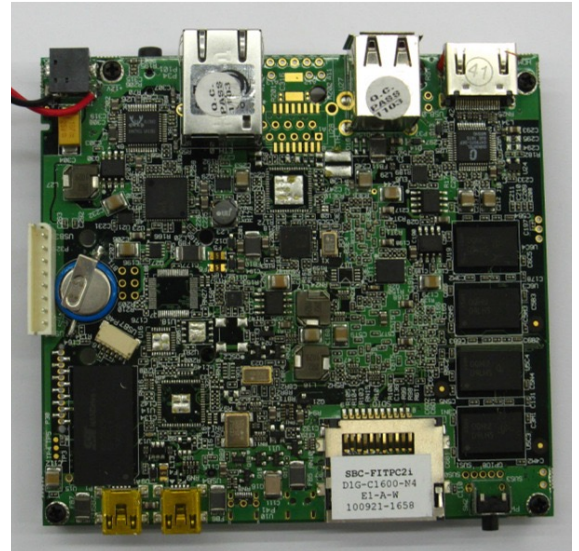
#### ■ Fonction

**Gérer le fonctionnement global du robot.**

#### ■ Présentation

Le « SBC-FITPC2i » est un ordinateur **mono carte** spécialement conçu pour les systèmes embarqués. Ces systèmes exigent des fonctionnalités très différentes de nos PC standards, parmi les plus importantes :

- ✓ des dimensions et un poids réduits ;
- ✓ une consommation très faible pour permettre un fonctionnement sur batterie ;
- ✓ l'absence de pièces mécaniques en mouvement, (disque dur, lecteur de CD) ;
- ✓ il doit fonctionner sous un système d'exploitation (Linux, Windows XP et Windows 7) et doit pouvoir embarquer des logiciels « standard » du marché (traitement d'image, serveur Web...). elle intègre également les interfaces (Ethernet, Wi-Fi, USB) pour communiquer avec le monde extérieur ;
- ✓ Le choix de la carte SBC-FITPC2i n'est pas innocent, spécialement étudié pour optimiser les ressources sur les systèmes embarqués (poids, consommation, coût, etc.). Elle répond aux besoins particuliers du robot, elle supporte un système d'exploitation Linux (open source), permettant d'intégrer des logiciels standard du marché pour la vision de Darwin, un serveur Web pour communiquer avec le monde extérieur. Elle permet par ce choix judicieux d'éviter l'étude et la fabrication d'une carte par la société et de réduire le coût de développement et par voie de conséquence celui du produit fini.



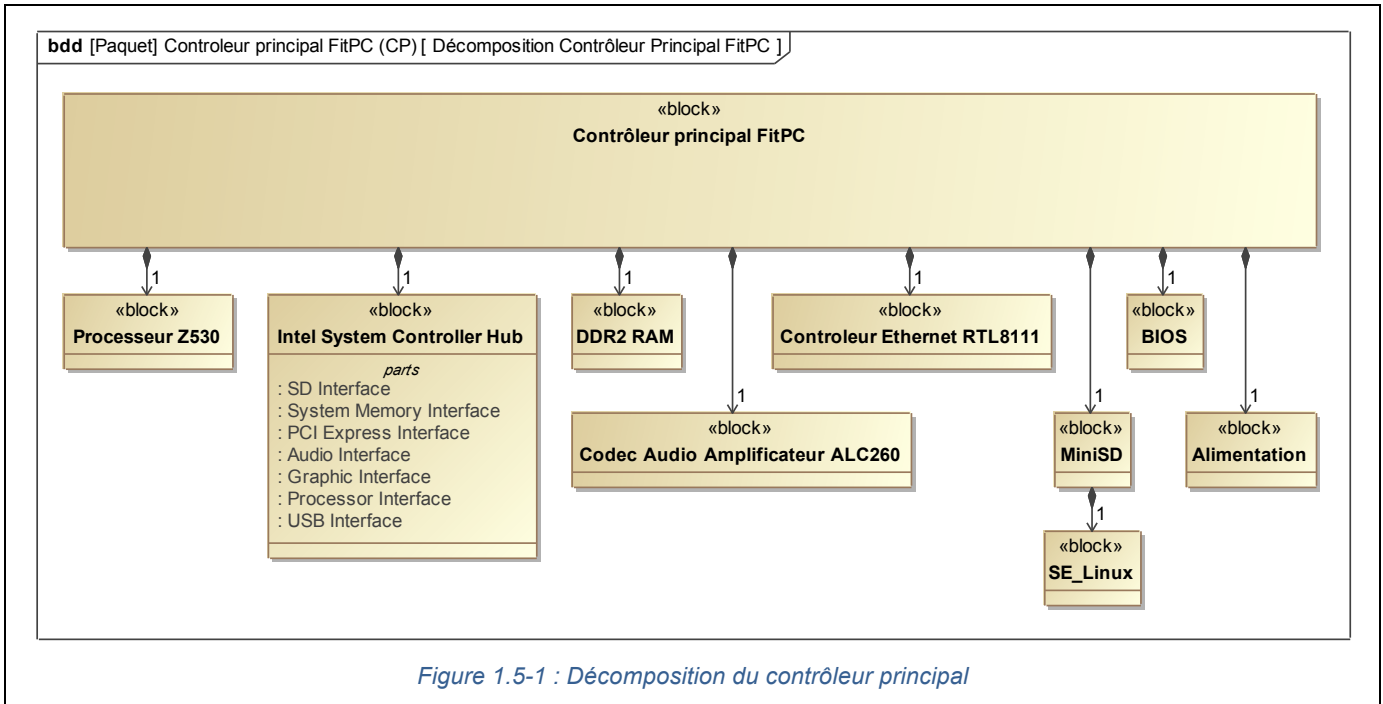
#### ■ Caractéristiques

Processeur (+ chipset)	Processeur Intel Atom Z530 à 1,6 GHz (Intel US15W)
Mémoire RAM	RAM DDR2 1GB - 533/400 MHz, 64-bit
Mémoire disque dur	Disque "bootable" à l'état solide "mémoire Flash" 4GB
Ethernet	Contrôleur Ethernet Gigabit PCI-Express (IEEE 802.3ab)
Wi-Fi	Normes 802.11 b/g/n ; 54 Mbps ; bande 2.4 GHz.
USB	2 ports USB 2.0
Port Vidéo Externe	Connecteur HDMI
Port Audio	Audio haute definition : Line Out, Line In, Microphone In.
Stockage Externe	Carte mini SD
BIOS	Phoenix
Système d'exploitation	Linux Ubuntu (pré-installé)





■ **BDD du contrôleur principal**





■ **IBD du contrôleur principal**

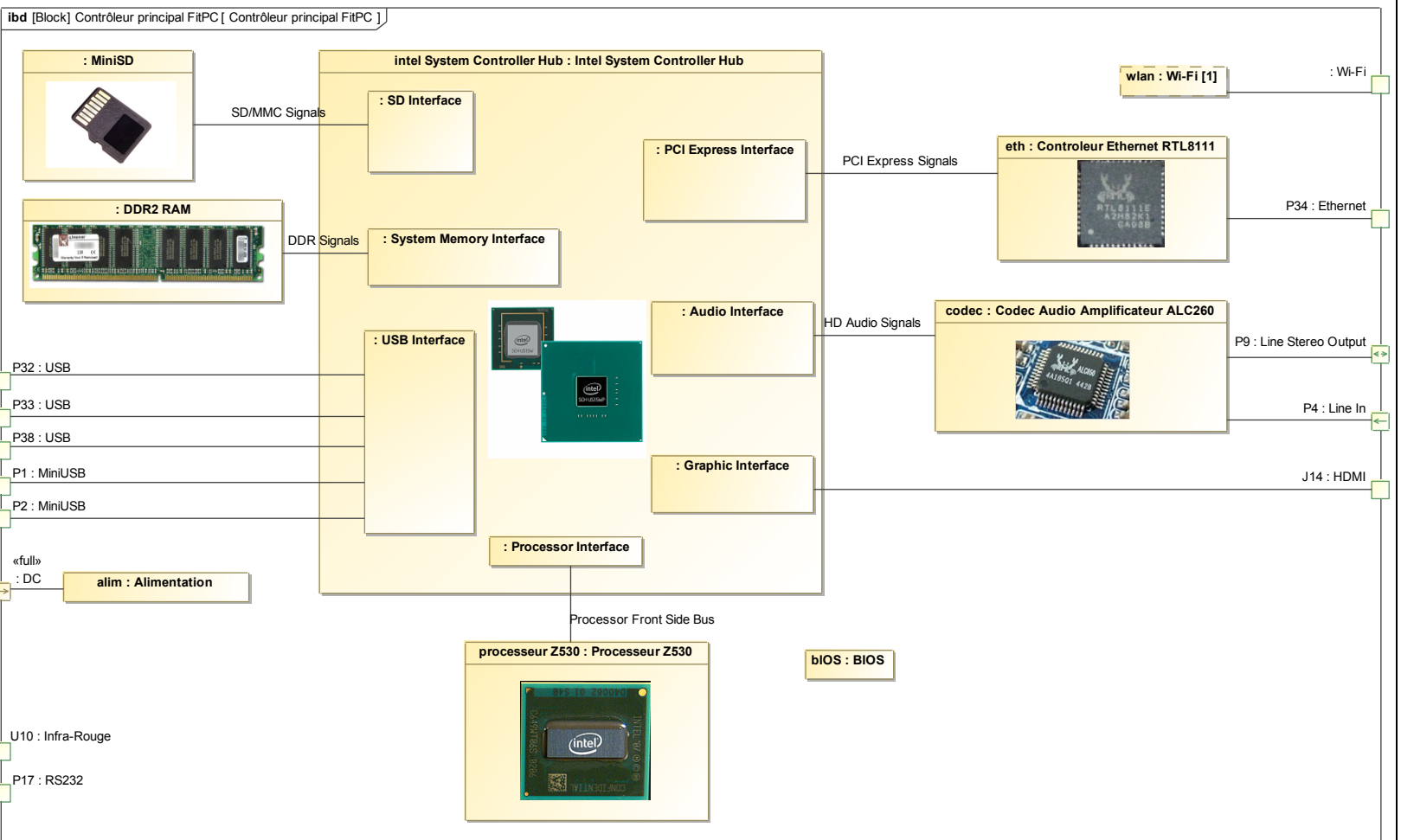


Figure 1-5-2 : IBD du contrôleur principal

Société DMS

Aeroparc St MARTIN – 12 rue de Caulet – 31300 TOULOUSE – ☎ : + 33 (0)5 62 88 72 72 📠 : + 33 (0)5 62 88 72 79

Site internet : [www.dmseducation.com](http://www.dmseducation.com) Email : [info@dmseducation.com](mailto:info@dmseducation.com)

51/124

Ce document et les logiciels fournis sont protégés par les droits de la propriété intellectuelle et ne peuvent pas être copiés sans accord préalable écrit de DMS.

Copyright DMS 2013



## 1.5.2. Bloc HDMI

### ■ Fonction

**Piloter un écran graphique.**

### ■ Présentation

L'interface graphique (DVI >> DisplayPort) est une interface numérique pour écran mise en place par le consortium « [Video Electronics Standards Association \(VESA\)](http://www.vesa.com). Il définit une nouvelle interconnexion numérique audio/vidéo, sans droit ni licence. Celle-ci est conçue pour relier un ordinateur à ses moniteurs, ou un ordinateur et un système de home cinéma.

### ■ Caractéristiques

- ✓ Interface DVI (Digital Visual Interface),
- ✓ vitesse de transmission → 165M pixels/second.
- ✓ Supporte des résolutions de 1600x1200 et 1920x1200 pixels.
- ✓ Connecteur de type standard HDMI-A. (J14)



## 1.5.3. Bloc USB

### ■ Fonction

**Connecter des périphériques USB.**

### ■ Présentation

La carte dispose au total de 7 ports USB au standard 1.1 et 2.0. Les débits atteignent respectivement 12 Mbits /sec pour le standard USB 1.1 et 480 Mbits/sec pour le USB 2.0.



### ■ Caractéristiques

- ✓ Les connecteurs P1 et P2 « Mini USB type AB » (non accessibles à l'extérieur).
- ✓ Les connecteurs internes P32, P38 et P40 sont exclusivement réservés pour les périphériques USB 2.0 fonctionnant à 480 Mbits/sec (disque USB, caméra...).
- ✓ Connecteur U27 (x2) type A (externe, sur le dos de Darwin) permet la connexion de périphériques USB 1.1 et USB 2.0.



### 1.5.4. Bloc Wi-Fi

#### ■ Fonction

**Communiquer sans fil avec son environnement.**

#### ■ Présentation

La carte contrôleur dispose d'une interface Wi-Fi pour communiquer sans fil avec son environnement. Elle est dotée d'un module radio additionnel et d'une antenne fonctionnant dans la bande 2.4 Ghz.

#### ■ Caractéristiques

Les normes 802.11b et 802.11g, proposent des modes de fonctionnement, permettant d'obtenir différents débits en fonction de la portée.



Standard	Bande de fréquence	Débit	Portée
Wi-Fi B (802.11b)	2.4 GHz	11 Mbit/s	100 m
Wi-Fi G (802.11g)	2.4 GHz	54 Mbit/s	100 m

### 1.5.5. Bloc Ethernet

#### ■ Fonction

**Communiquer avec son environnement.**

#### ■ Présentation

La carte contrôleur dispose d'une interface Ethernet pour communiquer avec un PC (point à point) ou en réseau. Equipé d'un connecteur RJ 45 ou deux suivant option (Dans notre cas 1 seul connecteur est installé dans le dos de Darwin).

#### ■ Caractéristiques

Le contrôleur Ethernet RTL8111 intégré à la carte supporte la norme **1000BASE-T**, (IEEE 802.3ab) aussi appelée Gigabit Ethernet. Elle autorise des débits de 1 000 Mbit/s sur 4 paires de fils de cuivre Cat5e (avec utilisation de connecteurs RJ45), sur une longueur maximale de 100 m.

1000BASE-T permet l'utilisation des 4 paires torsadées en mode *full duplex*. Ce standard est compatible avec 100BASE-TX et 10BASE-T, (10 et 100 Mbits/sec) il assure la détection automatique des vitesses de transfert en émission et réception.





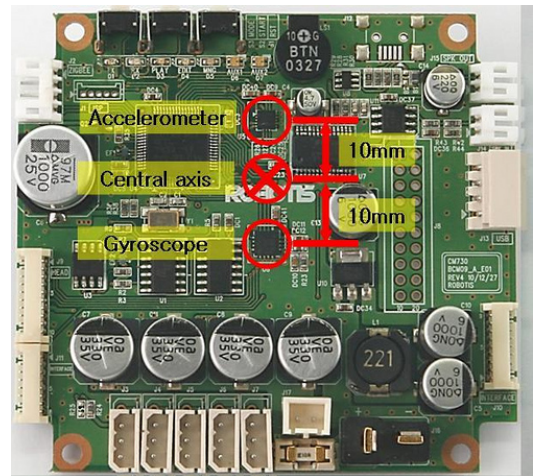
### 1.5.6. Bloc « Contrôleur secondaire CM-730 » (Sub controller)

#### ■ Fonction

**Contrôler les capteurs et actionneurs du robot.**

#### ■ Présentation

Cette carte embarque une unité de traitement équipé d'un microcontrôleur ARM Cortex M3 à architecture « RISC \* » et l'électronique « spécifique » du robot. Elle intègre le gyromètre, l'accéléromètre, les entrées/sorties standards et les interfaces : Leds, haut-parleur, etc... Sans oublier la gestion des 20 servomoteurs MX28... En étroite collaboration avec la carte PC embarquée par l'intermédiaire liaison à 2 Mbps « Méga bits par seconde », elle contrôle les capteurs et actionneurs du robot.



*\*RISC Acronyme de (Reduced Instruction Set Computer), l'architecture RISC désigne un processeur à "jeu d'instruction réduit" (une cinquantaine), offrant peu d'instructions mais un temps d'exécution défini et constant. Les RISC sont principalement représentés par la famille processeurs ARM. Ils sont très répandus dans les appareils embarquée (téléphones, mp3, robots, ...). Ils permettent une programmation utilisant des instructions préprogrammées simples et rapides. Ces processeurs ont en revanche une faible consommation de ressources.*

*CISC : Acronyme de (Complex Instruction Set Computer), l'architecture CISC désigne un processeur à "jeu d'instruction étendu" (environ 200), et couplé à des modes d'adressage complet. Les processeurs CISC sont principalement représentés par les x86 d'Intel. Ce sont les processeurs équipant la quasi-totalité des PCs. Ils permettent une programmation utilisant des instructions préprogrammées complexes. En contrepartie, les processeurs CISC sont consommateurs de ressources.*

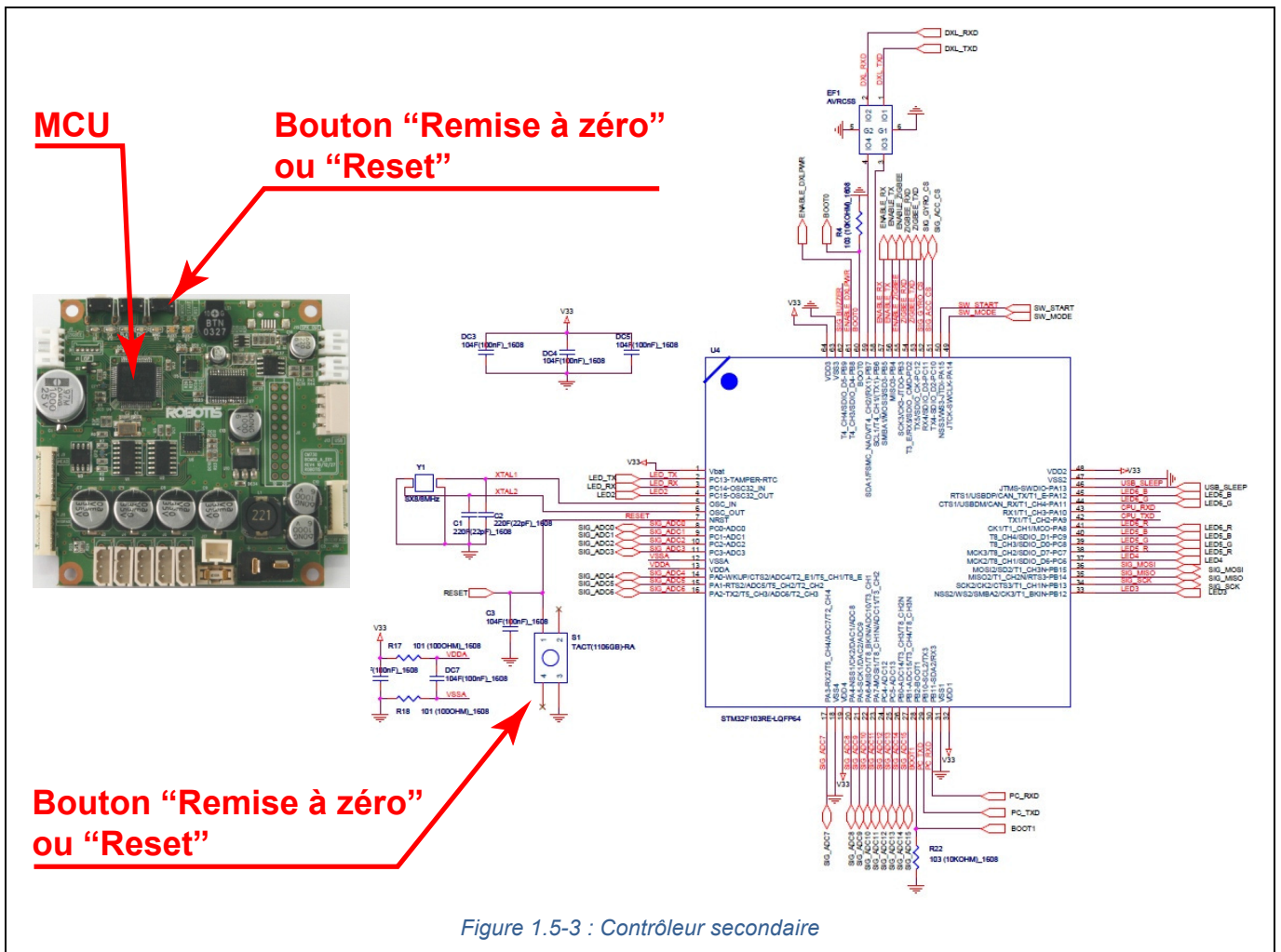
#### ✓ Mais pourquoi 2 contrôleurs ; Une démarche industrielle ?

L'un destiné au traitement des interfaces avec le monde extérieur, intègre des fonctions de communication de haut niveau (Ethernet, Wi-Fi, Usb...), serveur Web, traitement de la vision ... Ces applications exigeant de grandes puissances de calcul sont représentées sur le marché par un grand nombre de cartes « standards » destinées aux applications embarquées, aux PCs industriels (donc équipés de microprocesseurs CISC)... Il suffit au concepteur de faire son marché et de sélectionner la carte disposant des caractéristiques répondant à son besoin.

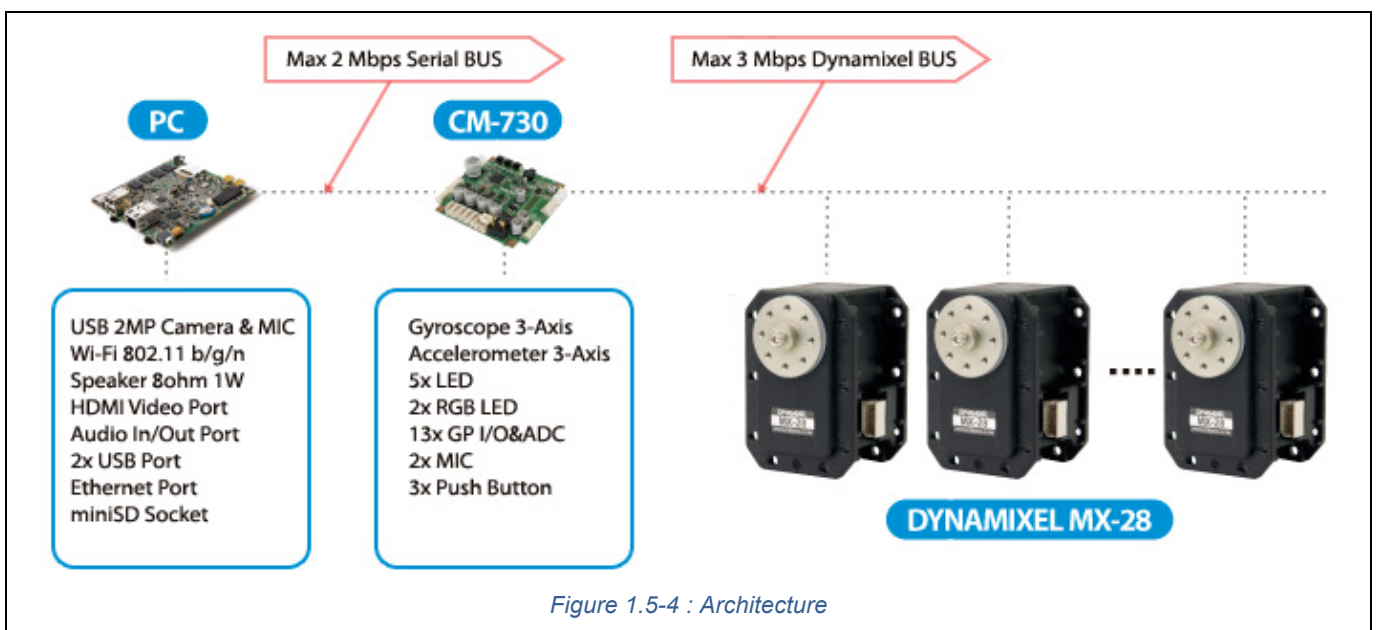
L'autre destiné à la gestion des capteurs absents des contrôleurs « standard » : gyromètre, bus propriétaire Dynamixel, etc ... Les capteurs et actionneurs « spécifiques à la robotique » sont souvent peu gourmands en terme de calcul, mais exigent des traitements simples et rapides comme les entrées/sorties tout ou rien (acquisition fin de course) ou le traitement des requêtes sur le bus Dynamixel. Dans ce cas pas de doute, il faut concevoir une carte adaptée aux caractéristiques des capteurs et actionneurs utilisés. Il est alors incontournable d'envisager une architecture autour d'un microcontrôleur RISC.

Le schéma structurel du processeur ARM est donné Figure 1.5-3 :





**■ Caractéristiques**







Microcontrôleur	ARM Cortex M3-32 bits, fréquence d'horloge 72 MHz (512KB Flash, 64KB SRAM, RAM statique)
Périphériques Internes	Timers, PWM (modulateur largeur d'impulsion), convertisseurs A/D et D/A
Ports Externes	13 ports E/S TOR ou Analogique/Digitale
Capteurs	3-axes gyromètre, 3-axes accéléromètre,
Audio & Micro	Amplificateur audio et micro
Communication	USB, Port Série, 5x TTL Ports (Dynamixel), RS485
Interface	5 x LED, 2 x RGB LED, 3 x Button, 1 x Buzzer

### 1.5.7. Bloc gyromètre

#### ■ Fonction

**Détecter et mesurer les variations de vitesse angulaire autour de 3 axes perpendiculaires (rotations).**

#### ■ Présentation

Le bloc gyromètre est essentiellement formé d'un circuit intégré électronique de dimensions 4 x 7.5 x 1,1mm. Il fait partie de la famille des MEMS, (*Micro Electro Mechanical Systems*). Ce sont des microsystèmes électromécaniques dont la taille varie de quelques dizaines de nanomètres à quelques [microns](#). Ces circuits sont de plus en plus présents au quotidien dans nos Airbags, nos Smartphones, jeux vidéos, etc. Ces micromachines souvent réalisées en poly-silicium permettent de s'intégrer sur le même circuit que les micro-capteurs et l'électronique de traitement associée.

Les technologies de fabrication des microsystèmes dérivent de celles de la microélectronique. Les principales spécificités des technologies de microsystèmes, comparées à la microélectronique, sont liées à la réalisation de parties mobiles, donc relativement détachées du support. Comme en microélectronique, les [Wafers](#) (Figure 1.5-5) (tranche) de silicium sont généralement utilisés comme substrat, et les microsystèmes sont produits de la même manière par une succession d'étapes d'[épitaxie](#), de [photolithographie](#). (*Micromoteur de la société MNX*, Figure 1.5-6).

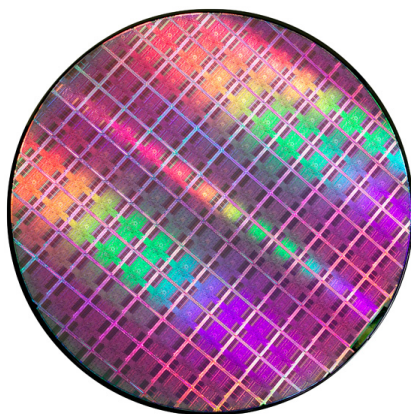


Figure 1.5-5 : Wafer

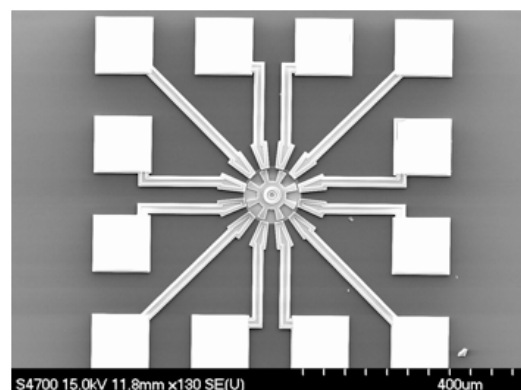


Figure 1.5-6 : Micromoteur de la société MNX



**Gyromètres vibrants :** 
$$\vec{F}_C = -2m\vec{\Omega}(t) \wedge \vec{v}$$

Le gyromètre est un [capteur](#) de mouvement. Il fournit une information de [vitesse angulaire](#) par rapport à un [référentiel inertielle](#). On distingue le gyromètre et le gyroscope qui est un capteur de position angulaire. La distinction est parfois subtile car un même appareil peut fonctionner suivant les deux méthodes.

Ces gyromètres sont basés sur des corps vibrant selon un certain [mode de vibration](#) (ou direction) et dont la rotation va permettre d'exciter un autre mode (parasite) en raison de l'apparition d'un couplage lié à la [force de Coriolis](#). La mesure de l'amplitude de vibration de ce mode parasite permet de remonter à la vitesse angulaire. Ce principe est particulièrement utilisé pour les [microsystèmes](#) réalisés sur silicium.

Le gyromètre de STMicroelectronics (LYPR540AH) (Figure 1.5-7 et Figure 1.5-10) va mesurer les trois degrés de rotation. Ce n'est pas le cas de tous les modèles présents sur le marché qui se limitent parfois à deux degrés de liberté, suffisant pour orienter l'écran d'un Smartphone, mais inadaptes à l'évolution dans l'espace en 3 dimensions de DARwIn-OP.



Figure 1.5-7 : Gyromètre de STMicroelectronics (LYPR540AH)

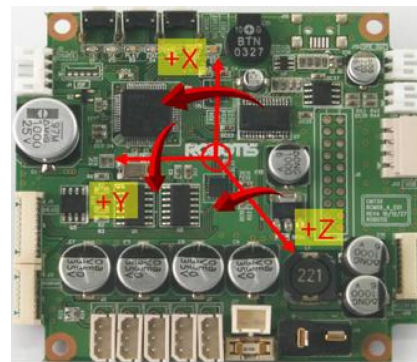


Figure 1.5-8 : Axes du gyromètre dans le contrôleur secondaire

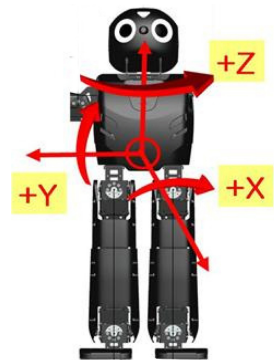


Figure 1.5-9 : Axes du gyromètre dans DARwIn-OP

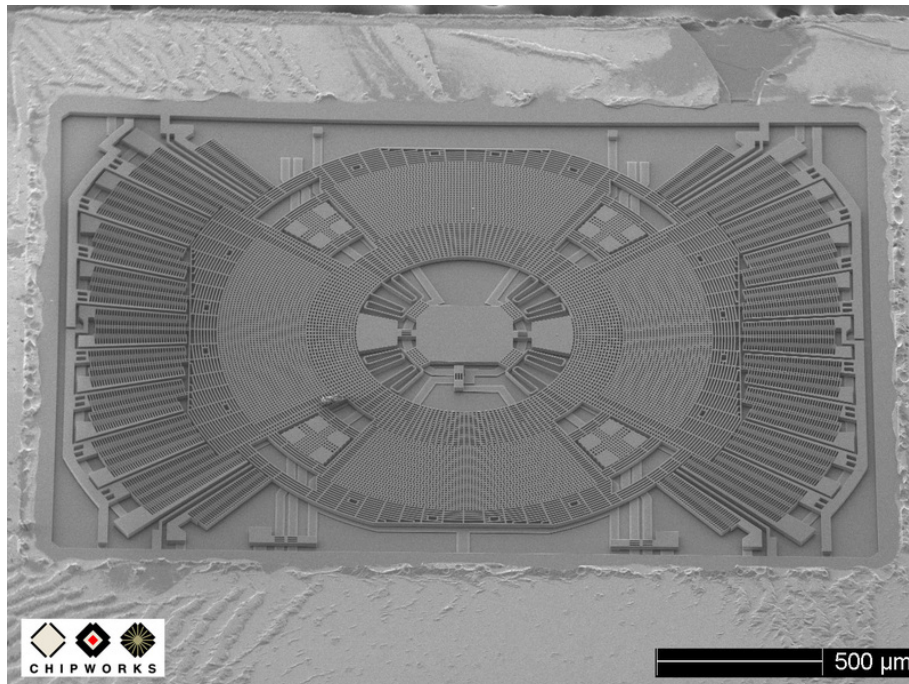
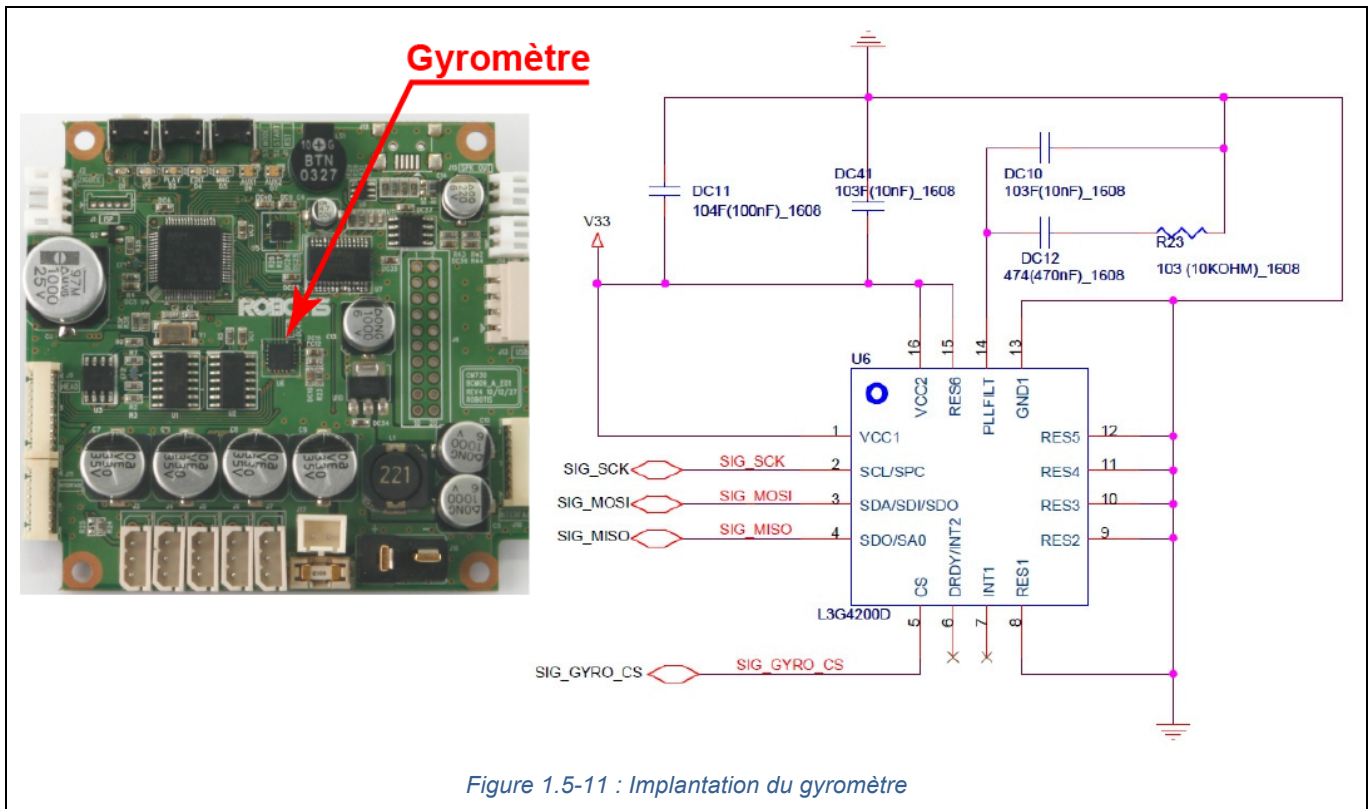


Figure 1.5-10 : Photo de la micromachine du circuit STMicroelectronics (LYPR540AH) prise avec un microscope à balayage électronique



■ **Caractéristiques**

Tension d'alimentation (Vdd)	2,7 V à 3,6 V (Typiquement 3 V)	
Sorties mesure (Analogique)	1 par axe(X, Y, Z) non-amplifiée ; 1 par axe(X, Y, Z) amplifiée x 4	
Gamme de mesure	Sortie non-amplifiée (X, Y, Z)	+/- 1600 dps (degré par seconde)
Gamme de mesure	Sortie amplifiée (X, Y, Z) (x 4)	+/- 400 dps (degré par seconde)
Sensibilité	Sortie non-amplifiée (X, Y, Z)	0,8 mV/dps
Sensibilité	Sortie amplifiée (X, Y, Z) (x 4)	3,2 mV/dps
Niveau Zéro (Voff)	(Tension de sortie sans mouvement) 1,5 V (pour Vdd = 3 V)	







### 1.5.8. Bloc Accéléromètre

#### ■ Fonction

**Détecter et mesurer les accélérations suivant 3 directions perpendiculaires (translations).**

#### ■ Présentation

Le bloc accéléromètre est un circuit intégré électronique de dimensions 4 x 4 x 1,45 mm disposés comme le montre la Figure 1.5-12. Il fait partie de la famille des MEMS (voir paragraphe précédent et Figure 1.5-13).

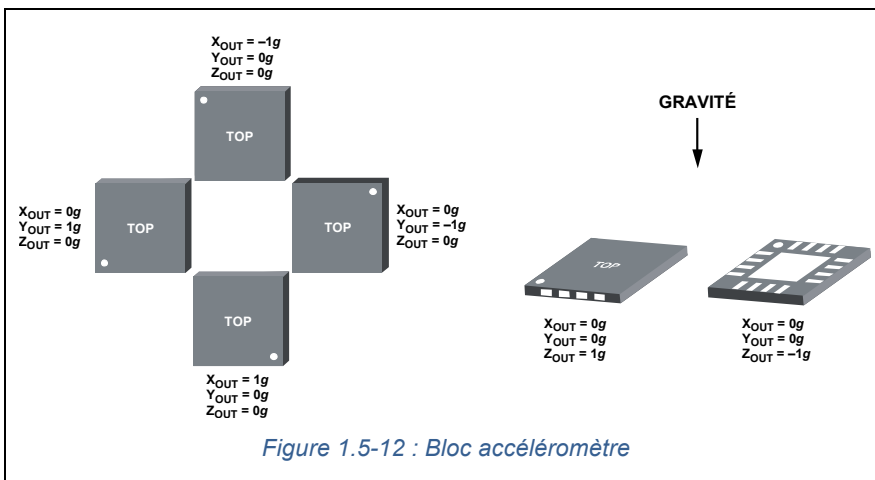


Figure 1.5-12 : Bloc accéléromètre

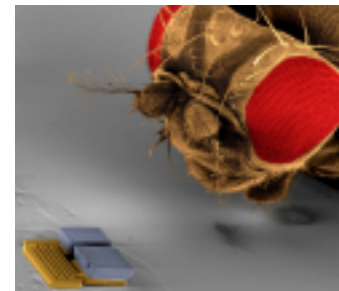
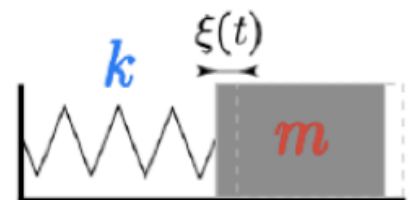


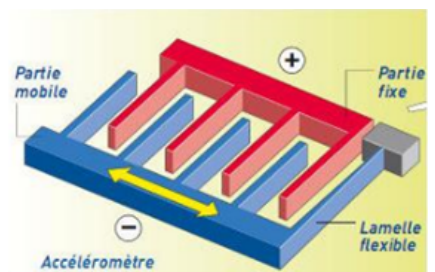
Figure 1.5-13 : Un Mems, mesurant 300 micromètres de longueur, mis au point par l'équipe suisse ETH Zurich qui a participé à la RoboCup 2009. c ETH Zurich

Un accéléromètre est un capteur qui, fixé à un mobile, permet de mesurer l'accélération de ce dernier.

La plupart des accéléromètres sont basés sur le principe fondamental de la dynamique  $F = m \times A$ , (où  $F$  somme des forces appliquées à une masse  $m$  appelée masse (sismique) mobile,  $A$  est l'accélération à laquelle est soumis la masse  $m$ ). On va mesurer le déplacement de  $m$  dû à une accélération en reliant  $m$  à une partie fixe. En bref, un accéléromètre peut être modélisé par un système masse-ressort.



Plusieurs techniques de détection existent (piézo-électrique, piézo-résistive, capacitives,...) pour récupérer les grandeurs physiques produites par les micromachines et la transformer en signal électrique, la détection capacitive est la méthode la plus utilisée à ce jour pour les accéléromètres « mems », et celle employée par notre circuit (ADXL335). Une partie fixe solidaire du substrat et placée en quinconce avec un peigne mobile « masse sismique », le tout retenu par une lamelle flexible « ressort ». Les mouvements réduisent ou augmentent la distance entre les lames modifiant en conséquence l'épaisseur du diélectrique, donc la valeur de la capacité. Cette valeur pouvant être aisément mesurée par une électronique adaptée, nous obtenons une différence de potentiel proportionnelle à l'accélération...





Ce capteur permet de mesurer :

- ✓ l'accélération (en  $m.s^{-2}$  ou en g) dans les axes x (axe avant/arrière), y (axe droite/gauche) et z (haut/bas).  $1 g = 9.8 m.s^{-2}$  ;
- ✓ l'inclinaison du capteur en combinant la mesure des différents axes ;
- ✓ les mouvements, les accélérations et décélérations, les chocs ;
- ✓ la gravité « **niveau zéro-g** » : C'est l'accélération mesurée lorsque l'accéléromètre n'est soumis à aucun mouvement. En général, on mesure 0g sur les axes X et Y, et 1g sur l'axe Z. Mais tout dépend de la position du circuit (voir Figure 1.5-14 et Figure 1.5-15).

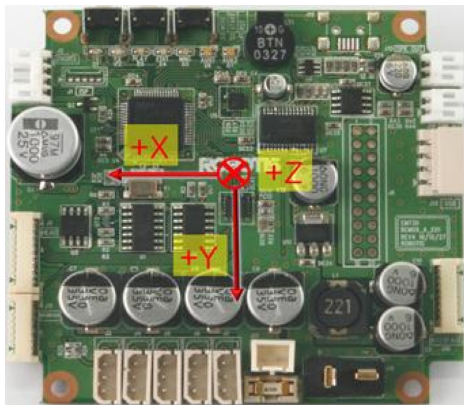


Figure 1.5-14 : Axes de l'accéléromètre dans le contrôleur secondaire

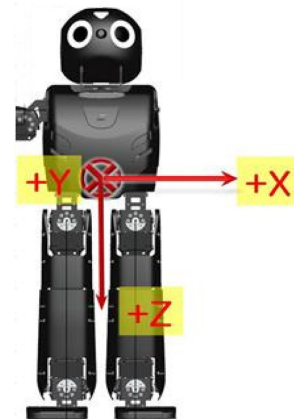
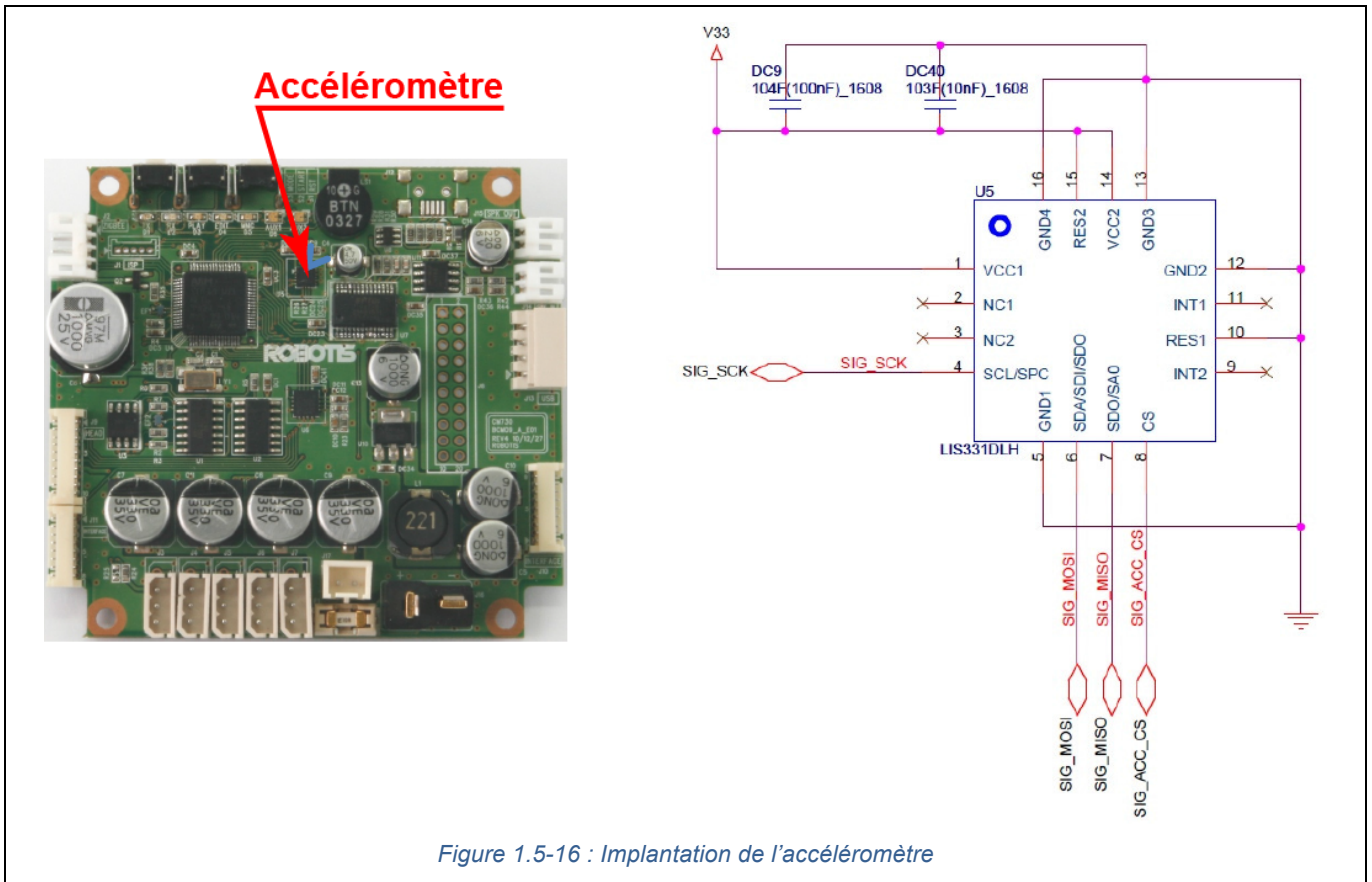


Figure 1.5-15 : Axes de l'accéléromètre dans DARwIn-OP

### ■ Caractéristiques

Caractéristiques à 25° et Vs=3V	Min	Typ	Max	observations
Tension d'alimentation	1,8V	3V	3,6V	Typiquement 3 V
Sorties mesure (Analogique)	1 par axe(X,Y,Z) impédance de sortie des amplis (~ 32KΩ)			
Gamme de mesure	+/- 3g	+/- 3,6g		
Sensibilité axe (X, Y ,Z)	270 mV/g	300 mV/g	330 mV/g	Proportionnelle à Vs (~ Vs/10)
Niveau Zéro (0g) axe (X,Y)	1,35V	1,5V	1,65V	Proportionnelle à Vs (~ Vs/2)
Niveau Zéro (0g) axe (Z)	1,2V	1,5V	1,8V	Proportionnelle à Vs (~ Vs/2)
Tension de sortie des amplis	0,1V		2,8V	Sans charge



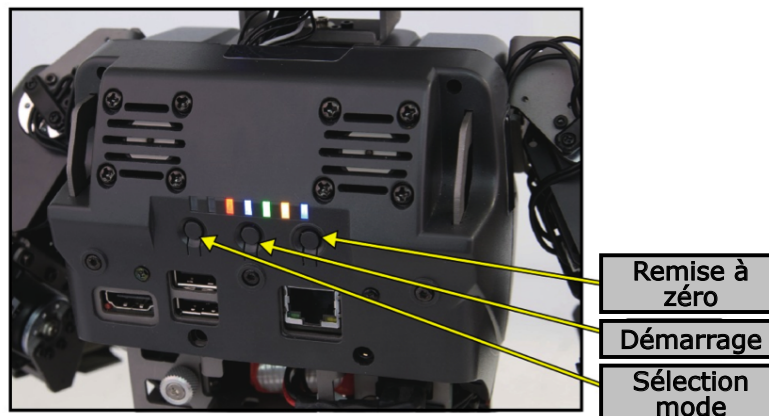
### 1.5.9. Bloc E/S boutons

■ **Fonction**

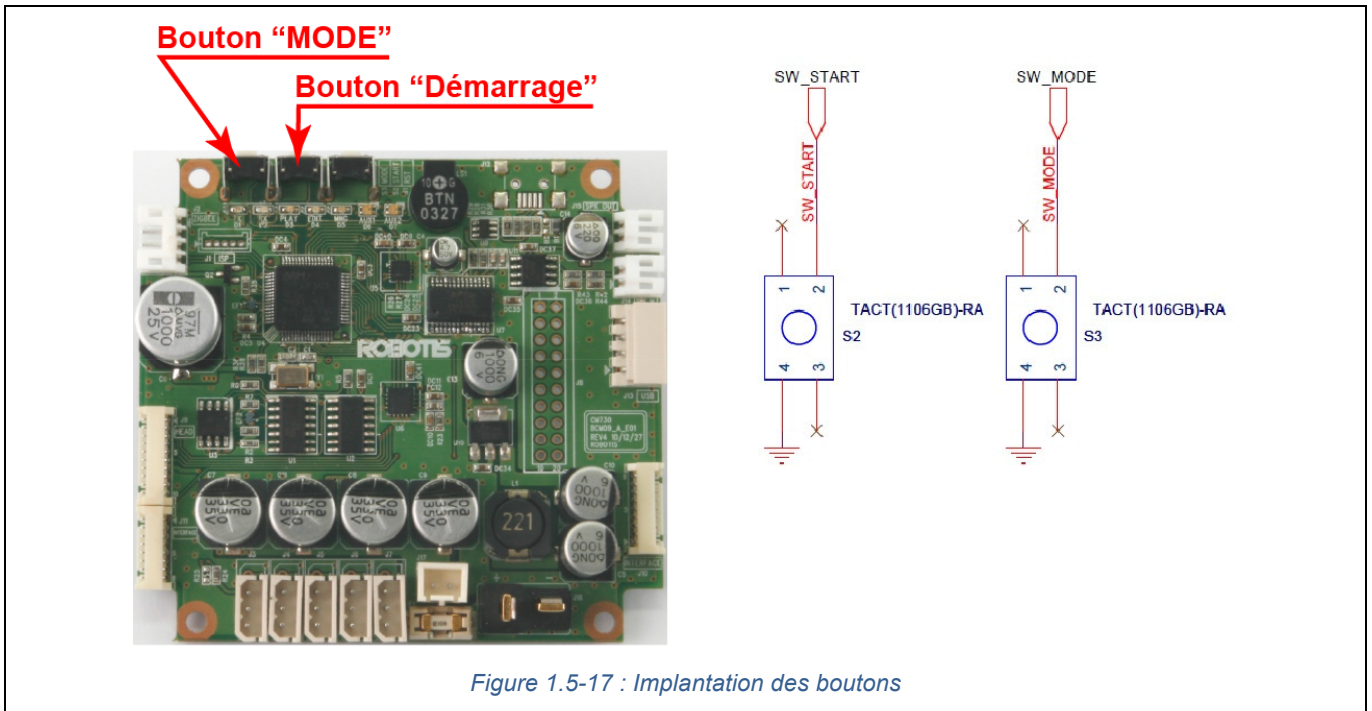
Activer les modes de fonctionnement du robot.

■ **Présentation**

Les boutons disposés sur le dos du robot permettent d'activer les modes de fonctionnement préprogrammés (démonstration, marche, vision, etc.) (Voir chapitre 2.4.7.2.)







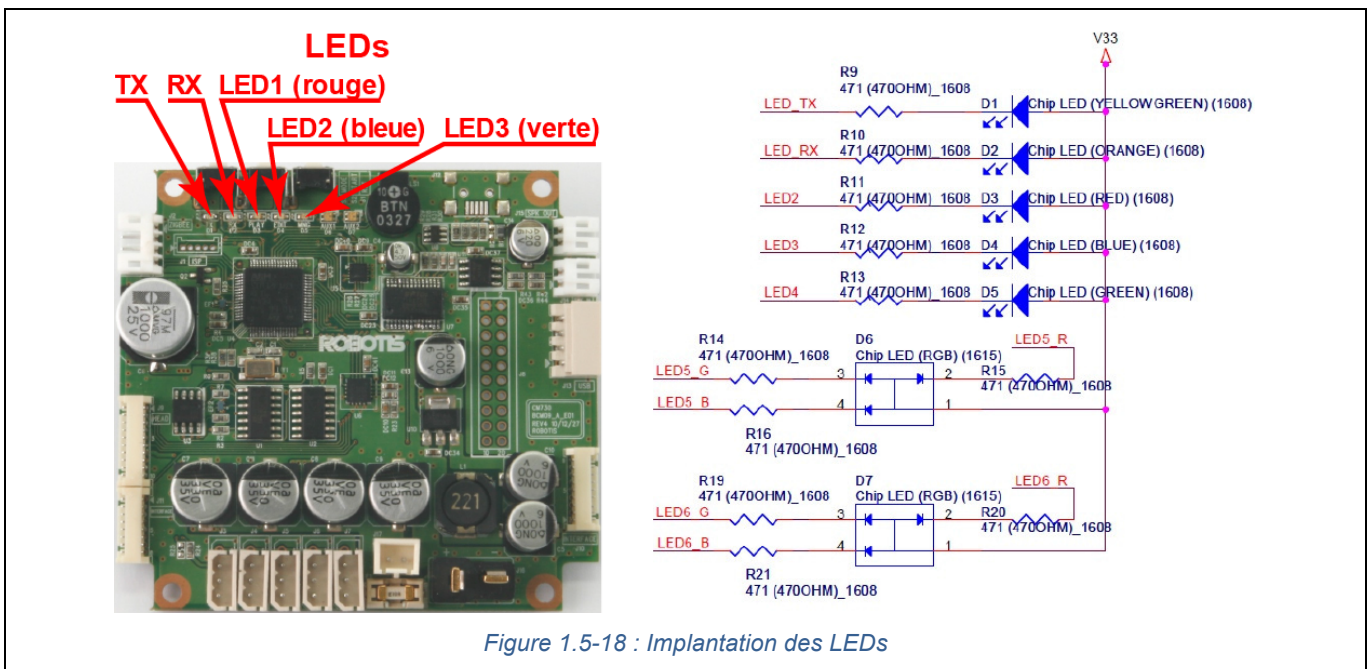
### 1.5.10. Bloc LEDs

■ **Fonction**

**Visualiser les modes de fonctionnement du robot.**

■ **Présentation**

Implantées sur la carte (avec les boutons) dans le dos, elles indiquent l'état et les modes de fonctionnement préprogrammés (démonstration, marche, vision, etc.). (Voir chapitre 2.4.7.2.)





### 1.5.11. Bloc Batterie Li-Po

#### ■ Fonction

**Alimenter le robot en énergie électrique de façon à le rendre autonome.**

#### ■ Présentation

Batterie embarquée sur DARwIn-OP, de forme parallélépipédique et comportant les connecteurs pour la relier d'une part sur le chargeur, et d'autre part sur DARwIn-OP.



#### ■ Caractéristiques

La batterie fournit l'énergie nécessaire pour faire fonctionner le circuit de puissance (servomoteurs) et le circuit de commande (contrôleurs et communications).

Type	Lithium polymère
Nombre de cellules (tension)	3 cellules de 3.7V (soit 11,1V)
Capacité	1000mAh
Capacité de décharge	10C = 10A (peut fournir instantanément 10 fois sa capacité soit 10 x 1000mAh)
Protection	contre toute charge ou décharge excessive, et les courts-circuits.

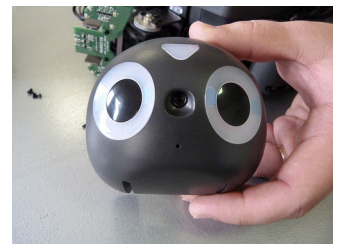
### 1.5.12. Bloc Caméra : l'œil de DARwIn-OP

#### ■ Fonction

**Voir l'environnement dans l'axe des yeux.**

#### ■ Présentation

Pour des images de grande qualité la lentille est conçue en collaboration avec l'un des pionniers de l'industrie optique, **Carl Zeiss**. La mise au point est automatique, avec autofocus intégré. Elle est équipée d'un capteur CCD de 2 mégapixels. Avec un éclairage faible ou à contre-jour, la caméra s'adapte de manière intelligente afin de produire la meilleure image possible.



#### ■ Caractéristiques

Optique	Carl Zeiss ® avec mise au point automatique.
Capteur CCD	Capteur natif haute définition « HD » de 2 Méga pixels.
Microphone	Microphone intégré avec réduction de bruit.
Balayage	Jusqu'à 30 images par seconde.
Vitesse transmission	Vitesse de la liaison à 2 Mbits/sec « USB 2.0 ».

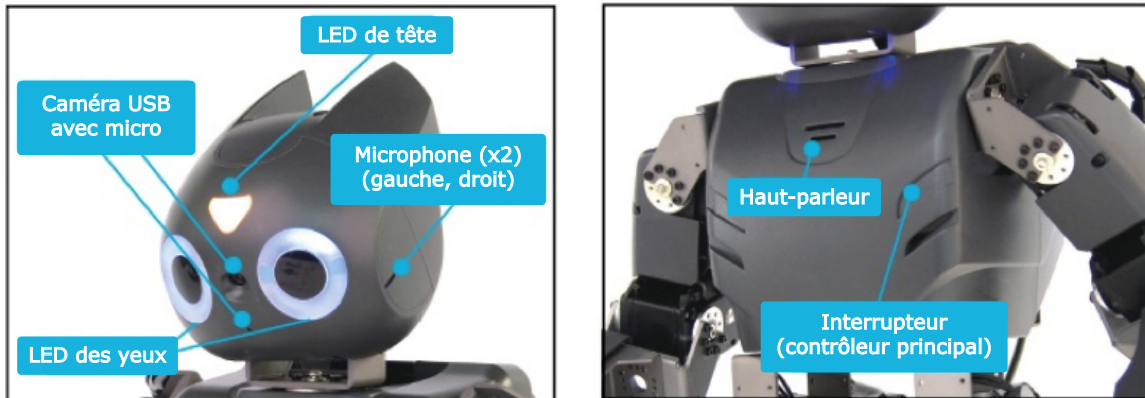


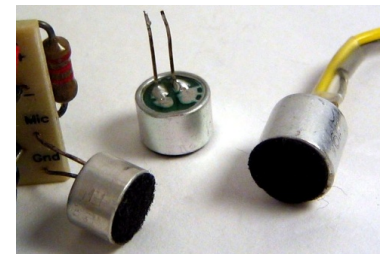
Figure 1.5-19 : Situation de la camera, microphone, haut-parleur et LEDs de tête

### 1.5.13. Bloc Microphone

#### ■ Fonction

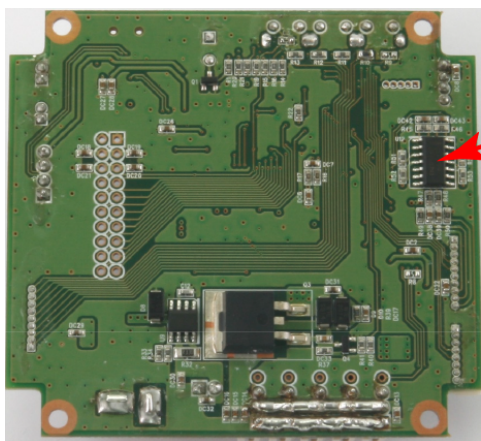
**Capter les sons environnants et les transformer en informations «électriques».**

#### ■ Présentation



Les signaux des 2 microphones sont acheminés sur le sous-contrôleur et ensuite amplifiés par un circuit spécialisé avant d'être envoyés sur les entrées des convertisseurs A/N et traités par le microcontrôleur du CM-730.

#### ■ Caractéristiques



#### Amplificateur de microphone

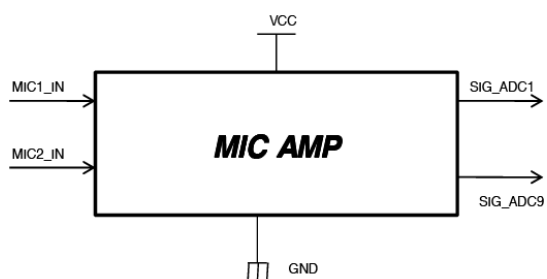


Figure 1.5-20 : Implantation de l'amplificateur de microphone



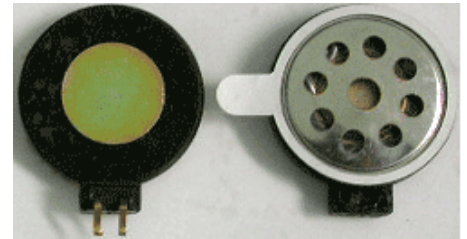
### 1.5.14. Bloc Haut-parleur

■ **Fonction**

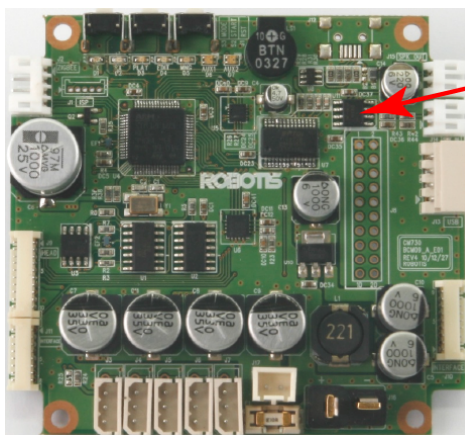
Émettre les sons ou les paroles générés par le robot.

■ **Présentation**

Pour le haut-parleur la source audio est fournie par le contrôleur principal, elle est ensuite amplifiée par un circuit amplificateur sur la carte du sous-contrôleur.



■ **Caractéristiques**



**Amplificateur de haut-parleur**

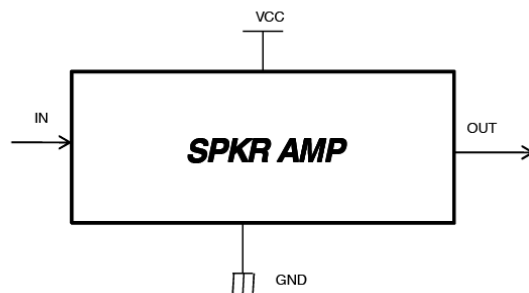


Figure 1.5-21 : Implantation de l'amplificateur de haut-parleur





**1.5.15. Description des flux échangés entre les blocs internes**

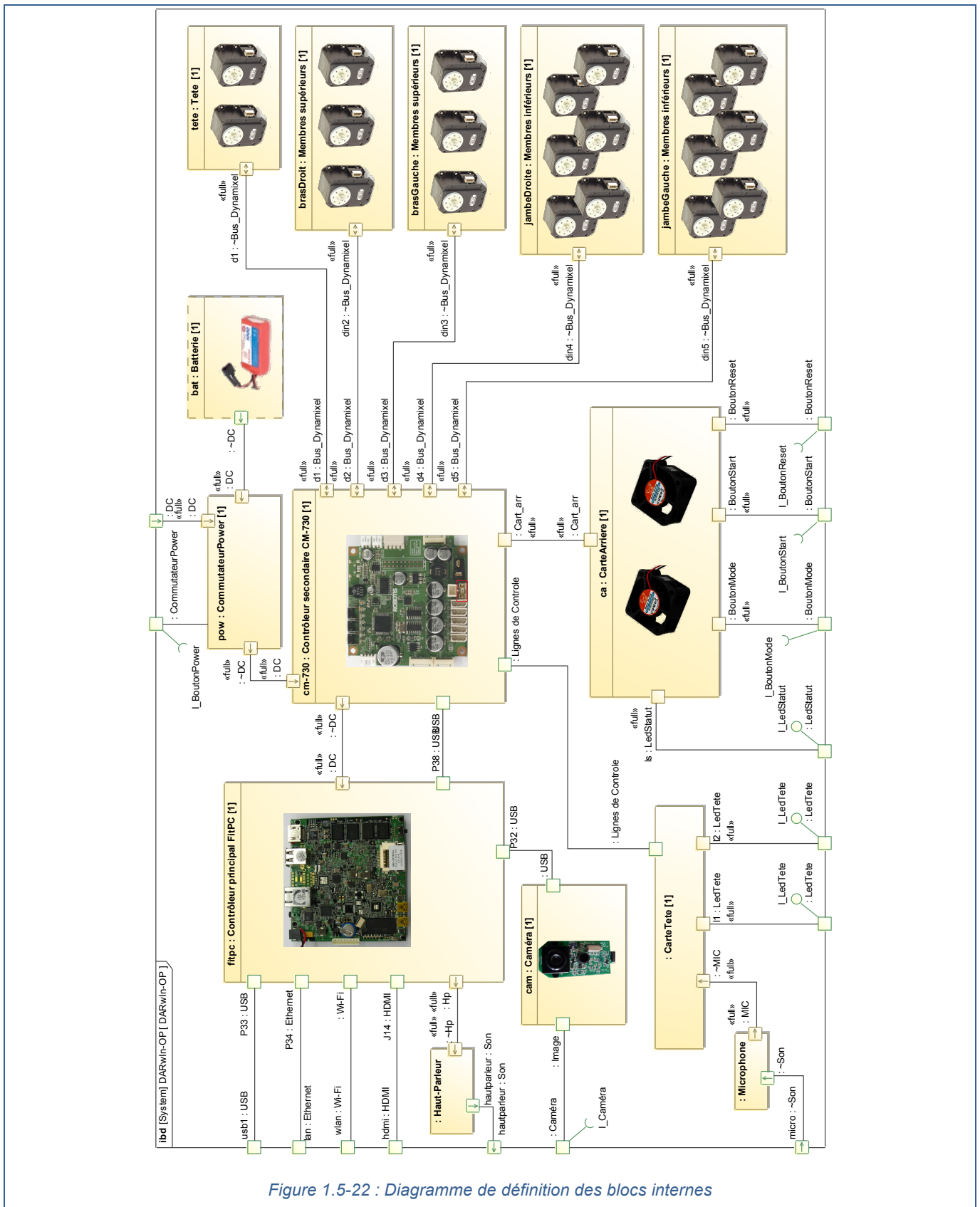


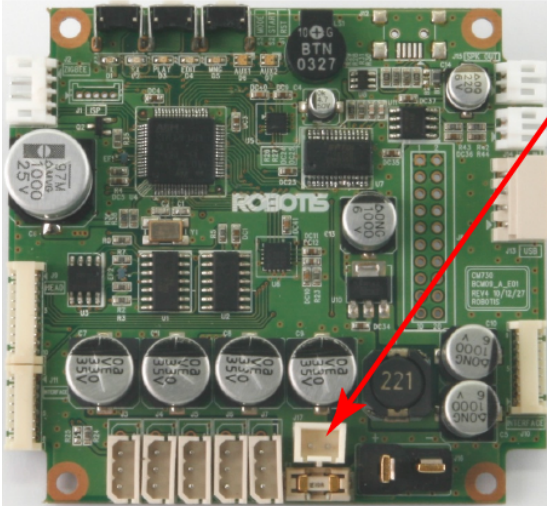
Figure 1.5-22 : Diagramme de définition des blocs internes



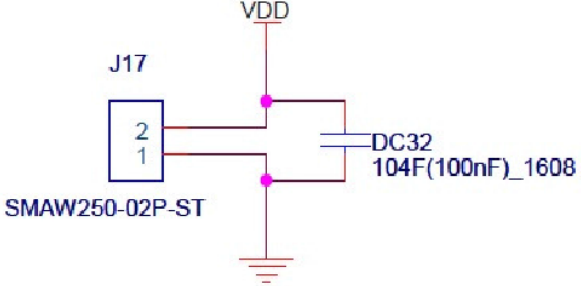


**1.5.15.1. DC**

Courant continu de la batterie (11,1V) permettant d'alimenter les sous-ensembles électriques et électronique du robot DARwIn-OP.



Alimentation du contrôleur principal

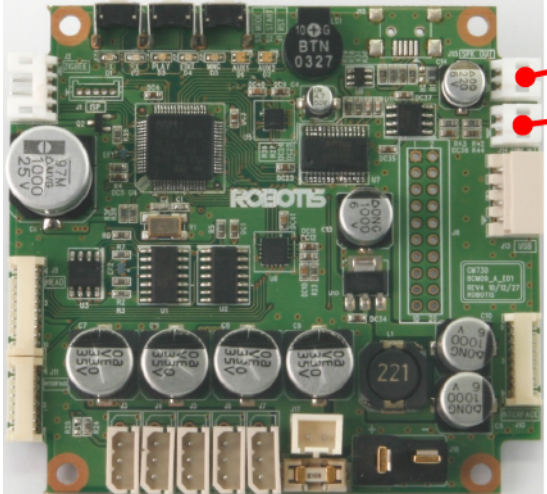


*Figure 1.5-23 : Implantation du branchement de l'alimentation du contrôleur principal*

**1.5.15.2. Hp**

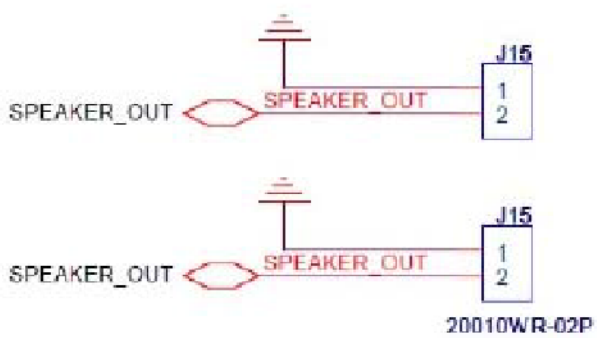
Ligne analogique véhiculant les signaux audiofréquences destinés au haut-parleur Miniature (situé dans le tronc de DARwIn-OP). Pour le haut-parleur la source audio est fournie par le contrôleur principal, il est ensuite amplifié par un circuit amplificateur sur la carte CM-730.





Sortie (out)

Entrée (in)

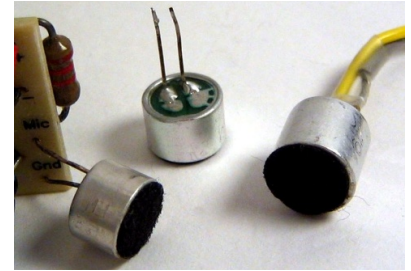
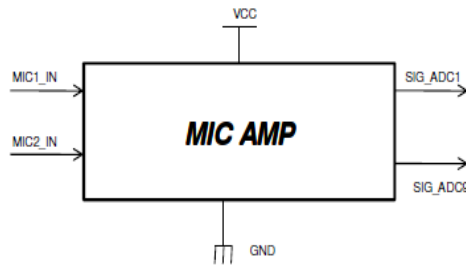


*Figure 1.5-24 : Implantation des connecteurs de haut-parleur*



### 1.5.15.3. MIC

Les signaux à bas niveau des 2 microphones sont acheminés sur le sous-contrôleur est ensuite amplifiés par un circuit spécialisé avant d'être envoyés sur les entrées des convertisseurs A/N et traités par le microcontrôleur du CM-730.



### 1.5.15.4. LEDs tête

Connecteurs et liaisons permettant d'alimenter les LEDs (LEDs tête) et de récupérer l'état des boutons (Bout\_mod) disposés sur la carte arrière.

**Connecteur de LED (tête)**

*Figure 1.5-25 : Implantation du connecteur de LED de tête*

### 1.5.15.5. Port HDMI

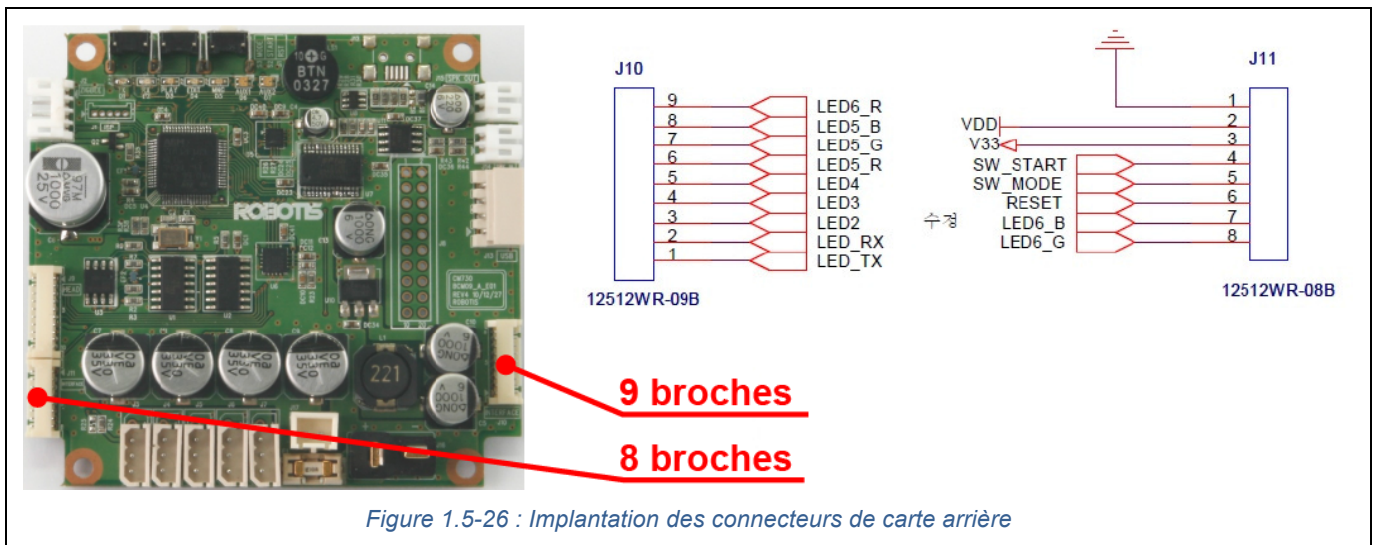
Signaux audio et vidéo au format numérique avec connectique au standard HDMI permettant la transmission en HD (Haute définition).

### 1.5.15.6. Port Contrôle, Wi-Fi, Port USB

Interface de communication du contrôleur principal permettant de récupérer la vidéo de la caméra et/ou de contrôler DARwIn-OP. Soit par un réseau filaire (Ethernet) soit par un réseau sans fil (Wi-Fi).

### 1.5.15.7. Carte-arr

Connecteurs et liaisons permettant d'alimenter les LEDs (LEDs mode) et de récupérer l'état des boutons (Bout-mod) disposés sur la carte arrière (Figure 1.5-26).



### 1.5.15.8. Bus USB

Liaison série haut débit permettant les échanges de données entre le contrôleur principal et secondaire, et bien sûr le flux vidéo de la caméra USB de DARwIn-OP.

Le bus USB est un standard sur tous les PC, et utilisé pour brancher les imprimantes, scanners, modems et de nombreux appareils stockant des données (disque dur, clé USB, appareils photo, etc.).

« **USB 2.0 Full Speed** » : le débit peut atteindre 12 Mbits/s (1,5 Moctets/s).

« **USB 2.0 High Speed** » : le débit peut atteindre 480 Mbits/s (60 Moctets/s).

Les connexions se font **point à point**. Il est possible de connecter jusqu'à 127 périphériques simultanément. Les ports USB supportent le **Hot Plug & Play**, c'est à dire qu'un périphérique peut être connecté sous tension et reconnu, sans redémarrage de l'ordinateur. Ils peuvent être connectés les uns à la suite des autres (en bus) ou reliés à un Hub (en étoile).



Le câble se compose de 4 fils. Le câble comporte un connecteur mâle de type A à une extrémité (connexion vers l'hôte) et un connecteur mâle de type B à l'autre extrémité (connexion vers l'appareil). Un blindage est fortement recommandé pour une utilisation à 12 Mbits/s ou plus. La longueur maximale est de 5 mètres. Il existe une autre série de connecteurs appelée USB "mini A" et "mini B". Ce sont les connecteurs que l'on retrouve sur les appareils photo et les lecteurs MP3.

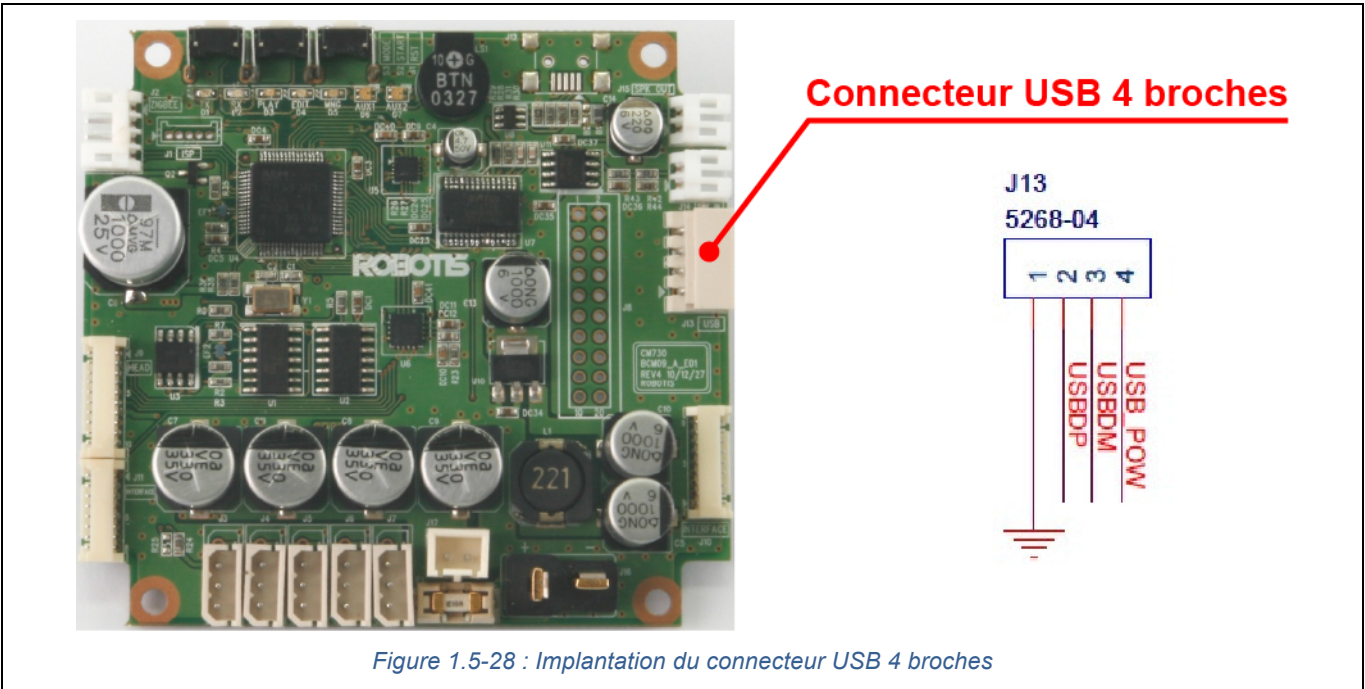
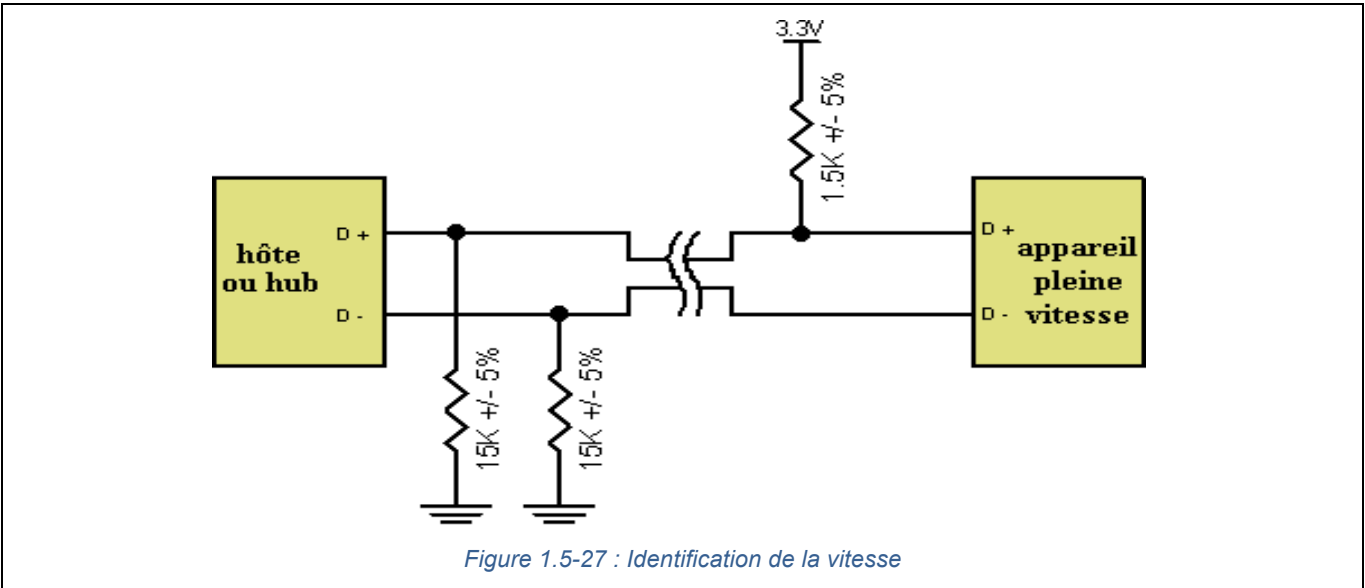
- ✓ L'alimentation, +5V (VBUS) (permet d'alimenter éventuellement les périphériques) et la masse GND.
- ✓ Les fils DATA+ (D+) et DATA- (D-) forment une **paire torsadée** qui transfèrent les signaux de données « différentiels ». La transmission différentielle améliore l'immunité aux bruits parasites.

N° de la broche	Couleur du fil	Fonction
n°1	rouge	alimentation Vbus (5v)
n°2	blanc	D-
n°3	vert	D+
n°4	noir	masse

Les **connecteurs USB type mini** disposent de **5** connexions. Celles du bus USB, plus une connexion supplémentaire pour la configuration du périphérique. L'OTG (On The Go) est une évolution de la norme USB, qui permet la connexion entre deux appareils **sans ordinateur** (un appareil photo et une imprimante). L'USB ayant besoin d'un hôte (host) unique, la fonction de l'OTG est de donner ce rôle d'hôte à un des deux périphériques. Le rôle host/device est fixé par le sens du câble, en positionnant cette broche supplémentaire à +Vbus ou à la masse (GND) suivant les extrémités.

Identification de la vitesse : Un appareil USB doit indiquer sa vitesse en mettant D+ ou D- à la tension 3,3V. Un appareil **haute vitesse** utilise une résistance de rappel rattachée à D+. Un appareil **basse vitesse** utilise une résistance de rappel rattachée à D-.





### 1.5.15.9. Bus Dynamixel

Pour contrôler un servomoteur Dynamixel, la communication doit être établie selon le protocole « propriétaire » de ROBOTIS et suivant une transmission asynchrone (8 bit de données, 1 bit de stop et pas de parité...), en mode « half-duplex ».

#### ■ Niveaux électriques

Pour commander les actionneurs Dynamixel MX 28T, le contrôleur CM-730 a besoin d'adapter les signaux de l'UART (Unité Asynchrone de Réception et Transmission) en signaux de type TTL. Le schéma est montré Figure 1.5-29.

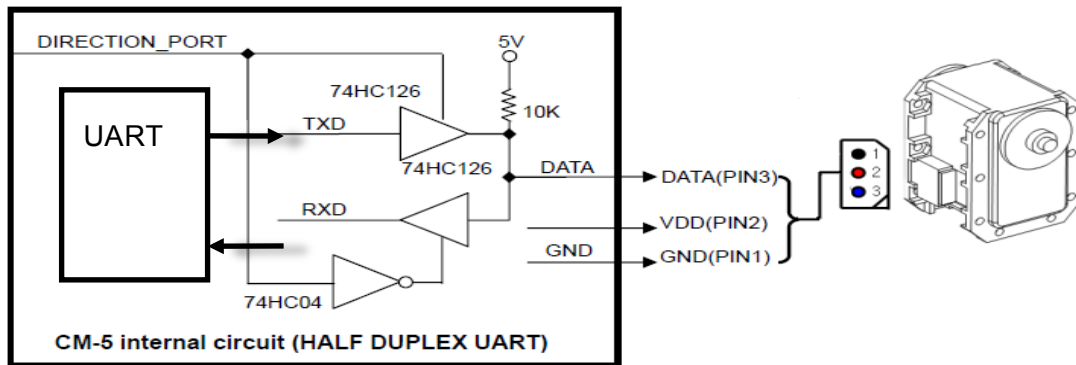
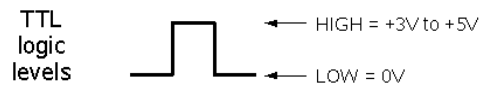


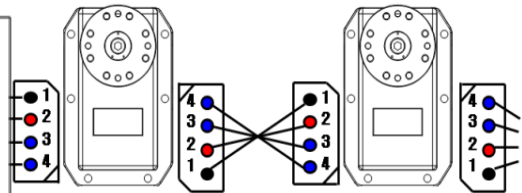
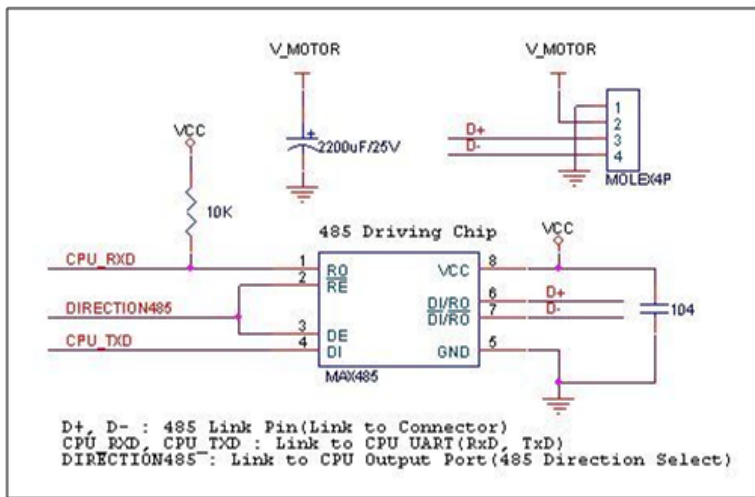
Figure 1.5-29 : Schéma de l'adaptateur

Les niveaux électriques du type **TTL** (Transistor/Transistor/Logique), n'ont rien d'un standard de transmission industriel. Mais sont tout simplement les niveaux électriques issus directement des circuits du contrôleur.



Le signal de données TTL (DATA pin 3) du bus Dynamixel est issu du circuit 74HC126 et maintenu au repos à un niveau proche de +5V par une résistance de rappel (Pull-up).

Les servomoteurs sont alimentés par les broches VDD (+) pin 2 et GND (masse=0v) pin 1.



Les servomoteurs sont alimentés par Pin1 (masse), Pin2 (+). La direction du signal de données en émission ou en réception ( TxD et RxD) est déterminée en fonction du niveau du signal «direction 485 ».

Niveau haut: TxD -> (D+ /D-)

Niveau bas: RxD -> (D+ /D-)

Figure 1.5-30 : Connexions des servomoteurs

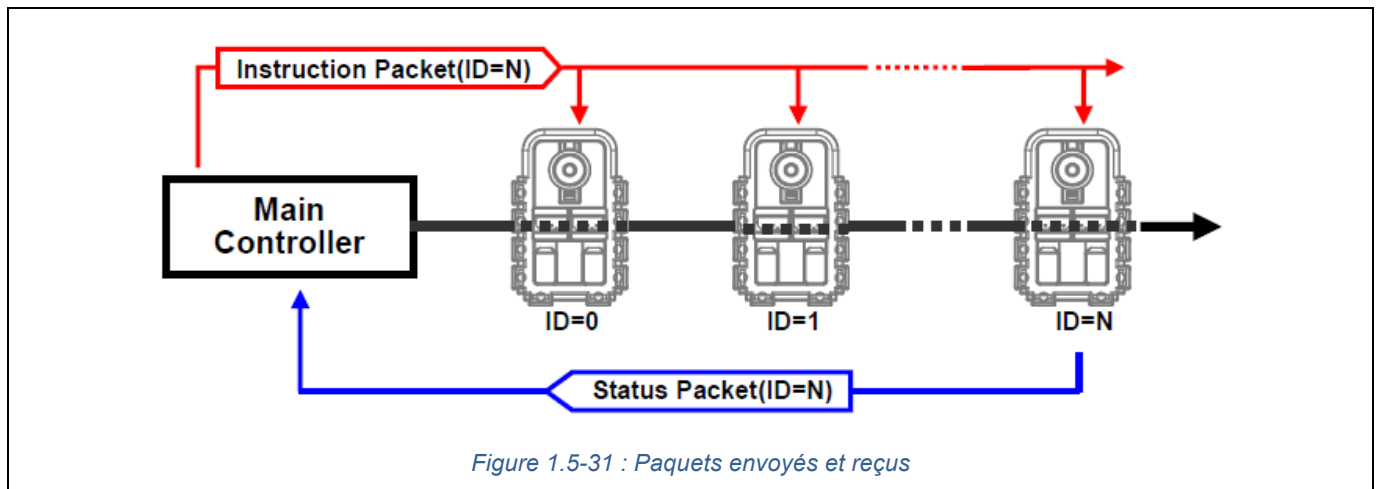
Pour contrôler la série MX 28R avec un contrôleur CM-730, le signal de l'UART doit être converti en signal de type RS485.

La communication ne peut être établie que dans un seul sens à la fois. Tous les périphériques (servomoteurs) doivent être en mode « **réception** » à l'exception du contrôleur qui lui est en « **émission** ». Le contrôleur définit la direction de la communication en émission (demande) ou réception (réponse).

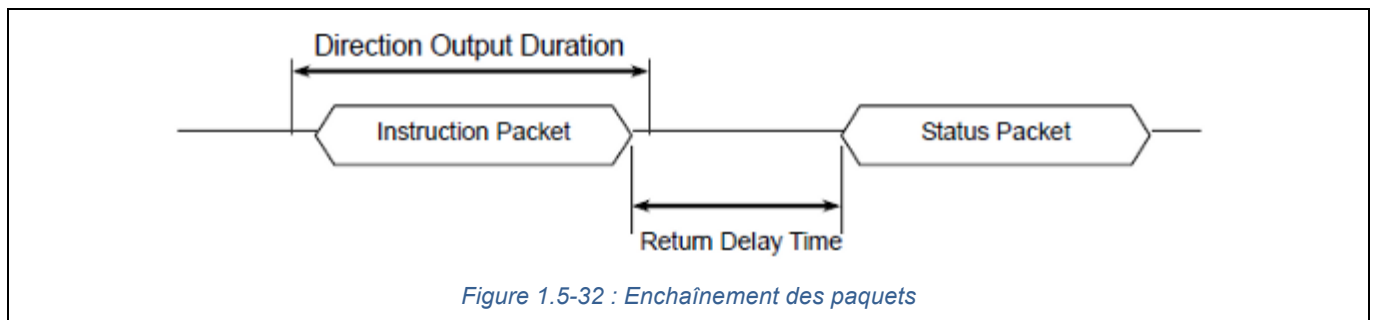




Le contrôleur CM-730 communique avec les servomoteurs en envoyant et recevant des données par « paquets ». Il existe deux sortes de paquets :



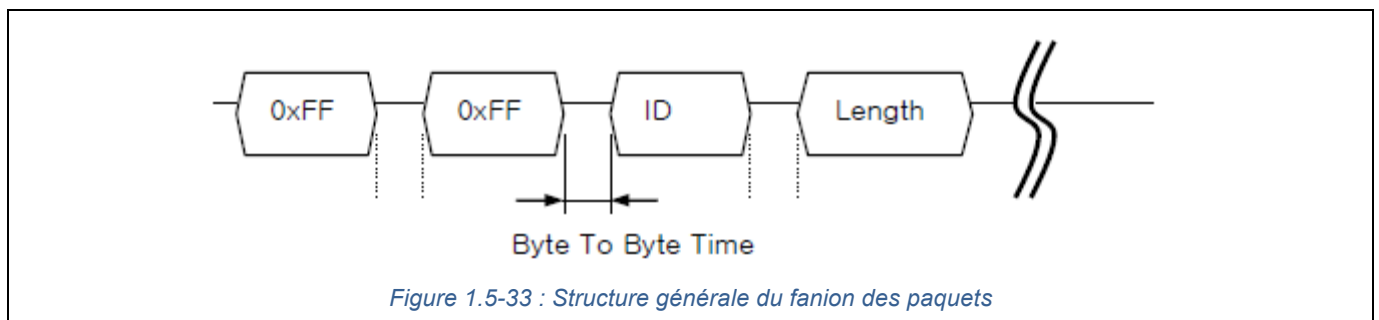
- Les Paquets **Instruction**, émis par le contrôleur vers les servomoteurs.
- Les Paquets **Statut**, renvoyés par les servomoteurs vers le contrôleur.



■ **Structure des paquets (ou trames)**

Dans les réseaux informatiques, une **trame** est un paquet d'information véhiculé au travers d'un support physique (cuivre, fibre optique, etc.). La caractéristique d'une trame est qu'il est possible d'en reconnaître le début et la fin, grâce à une série particulière de bits dénommée fanion, préambule ou en-tête (*flag en anglais*).

La structure générale du fanion des paquets (instruction ou statut) est donnée ci-dessous :





- 0xFF--- 0xFF : Ces deux premiers octets informent les récepteurs du début d'un paquet. Dans le cas de notre protocole, il s'agit de 2 octets dont les bits sont positionnés à « 1 » en hexadécimal « FF » et en binaire « 1111 1111 »...
- ID : Il s'agit de l'identifiant (adresse) du servomoteur a qui est destiné (ou qui renvoi) un paquet. Chaque servomoteur a une ID unique (de la même manière que l'adresse IP sur un réseau informatique). On peut utiliser 254 ID : de 0 à 253 (0X00 ~ 0xFD).
- ID de diffusion : adresse de diffusion (broadcast en anglais) ID = 254 (0xFE).
- Si l'ID de diffusion est utilisée, tous les servomoteurs sont destinataires du paquet d'instruction et aucun paquet statut n'est retourné.
- Length : C'est la longueur du paquet. La longueur est calculée comme étant « le nombre de paramètres (N) + 2 ».

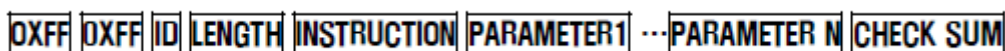
Chaque octet est séparé par un temps de retard (byte to byte time). Si le délai est supérieur à 100 ms, alors le récepteur reconnaît un problème de communication et attend le prochain en-tête (0xff 0xff).

### ■ Structure des paquets d'instruction

Le paquet d'instruction est une suite de données que le contrôleur envoie aux servomoteurs. Cette suite de données se compose d'un octet **instruction** (commande) ; suivi de paramètres si l'instruction nécessite des données auxiliaires (vitesse, position, etc.) et se termine par une somme de contrôle (Chek Sum). Elle est utilisée pour vérifier si le paquet est endommagé pendant la communication. La somme de contrôle est calculée selon la formule suivante :

$$\text{Check Sum} = \sim (\text{ID} + \text{Longueur} + \text{Instruction} ++ \text{Parameter1} \dots \text{Paramètre N})$$

Le signe " ~ " est l'opérateur binaire NON.



- INSTRUCTION : Cet octet donne une instruction au servomoteur (liste des instructions dans le tableau ci-après).

Valeur	Nom	Fonction	Paramètres
0x01	PING	Aucune exécution. Il est utilisé lorsque le contrôleur est prêt à recevoir des paquets statut (état).	0
0x02	READ_DATA	Cette instruction lit les données du servomoteur.	2
0x03	WRITE_DATA	Cette instruction écrit des données dans le servomoteur.	2 ou plus
0x04	REG WRITE	Il est semblable à WRITE_DATA, mais il reste à l'état de veille (dans un registre ou case mémoire) sans être exécutée tant que la commande ACTION n'est pas transmise.	2 ou plus



0x05	ACTION	Cette instruction lance la commande enregistrée dans le registre REG WRITE.	0
0x06	RAZ	Cette commande restaure l'état du servomoteur au réglage usine par défaut.	0
0x83	SYNC WRITE	Cette commande est utilisée pour contrôler simultanément plusieurs servomoteurs à la fois.	4 ou plus

### ■ Structure des paquets statuts (d'états)

Le servomoteur exécute l'instruction reçue du contrôleur et renvoie le résultat au contrôleur. Les données renvoyées sont appelés paquets statuts (d'état). Cette suite de données se compose d'un octet **statut** suivi de paramètres si la réponse nécessite des données auxiliaires (sauf en cas d'erreur) et se termine par une somme de contrôle (Chek Sum). La somme de contrôle est calculée selon la formule suivante.

$$\text{Check Sum} = \sim (\text{ID} + \text{Longueur} + \text{Erreur} + \text{Parameter1} + \dots + \text{Paramètre N})$$

Le signe " ~ " est l'opérateur binaire NON.

**0xFF 0xFF ID LENGTH ERROR PARAMETER1 PARAMETER2... PARAMETER N CHECK SUM**

La signification de chaque bit **d'erreur** est décrite dans le tableau ci-dessous : les bits sont mis à « 1 » en cas d'erreur.

Bit	Nom	Contenu
Bit 7	0	-
Bit 6	Erreur d'instruction	En cas d'envoi d'une instruction non définie ou la réception d'une instruction « action » sans instruction « reg-write ».
Bit 5	Erreur surcharge	Lorsque le courant de charge ne peut pas contrôler le couple.
Bit 4	Checksum	Lorsque la somme de contrôle du paquet instruction transmis est incorrecte.
Bit 3	Range Error	Quand une instruction est hors de la plage d'utilisation.
Bit 2	Erreur surchauffe	Lorsque la température interne du servomoteur est hors de la plage de température de fonctionnement définie.
Bit 1	Erreur limite d'angle	Lorsque la position est hors de la plage de limite d'angle à droite ou à gauche .
Bit 0	Erreur de tension d'entrée	Lorsque la tension appliquée est hors de la plage de tension de fonctionnement définie.



## 2. Définition du produit didactique

### 2.1. Identification du produit

### 2.2. Présentation générale du produit didactique

### 2.3. Déclaration de conformité CE

### 2.4. Notice d'instruction du produit didactique

#### 2.4.1. Mise en service de l'équipement

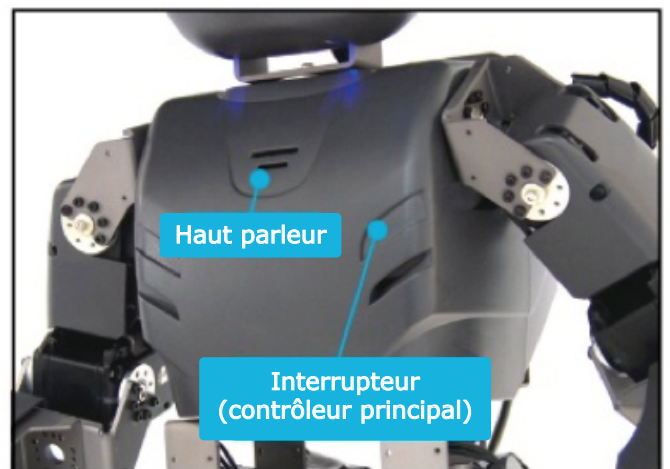
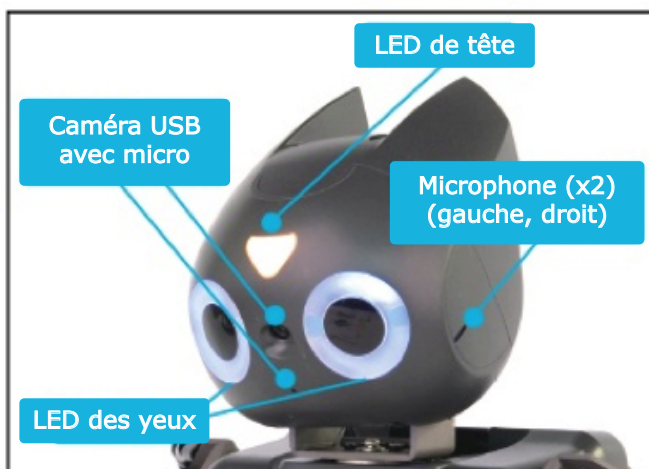
#### 2.4.2. Contenu du colis



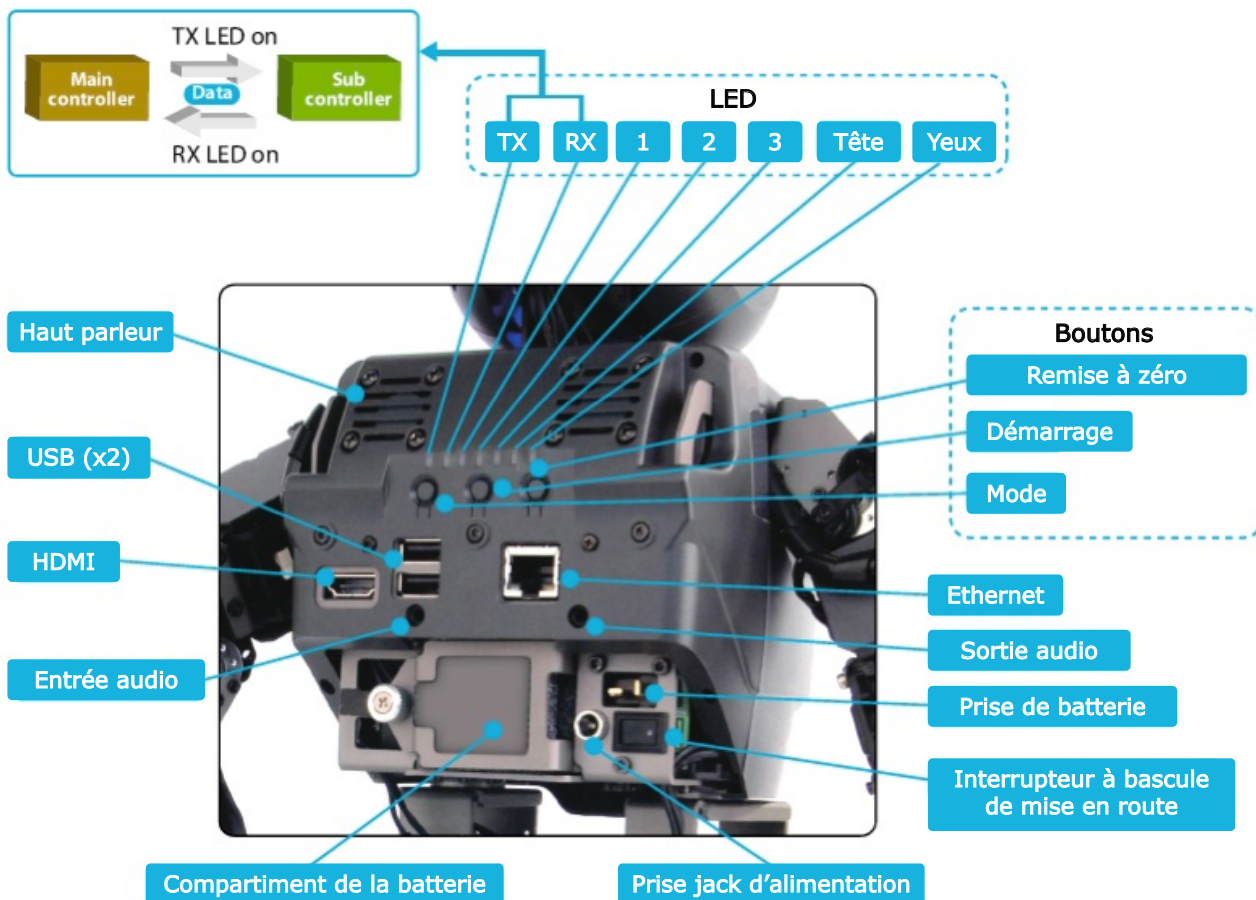


Repère	Nombre	Désignation
1	1	Robot DARwIn-OP
2	3	Batterie rechargeable
3	1	Chargeur
4	1	Alimentation 220V alternatif (AC) en 12V continu (DC)
5	1	Câble d'alimentation
6	1	Câble Ethernet
7	2	Tournevis + clef à lène
8	1	Ensemble de câbles TTL
9	1	Ensemble de vis
10	1	Balle rouge
11	1	Jeu de cartes couleurs (x7)
12	1	Clef USB de réinstallation
13	1	Guide de prise en main
14	1	CD avec le logiciel RoboPlus
15	2	Fusible

### 2.4.3. Localisation des capteurs et des raccords







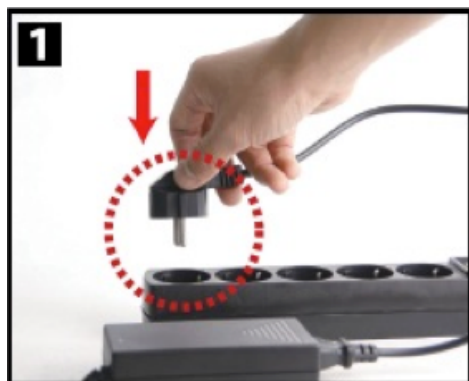
#### 2.4.4. Manutention

#### 2.4.5. Assemblage et raccordement

#### 2.4.6. Première mise en service

##### 2.4.6.1. Mise en charge des batteries

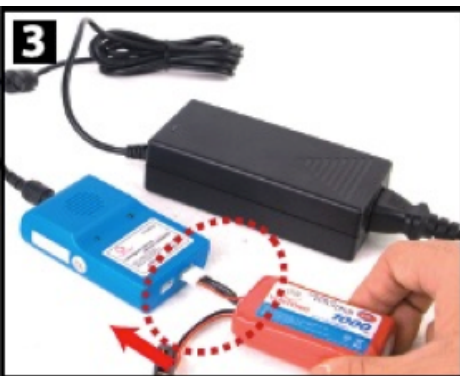
Pour charger la batterie procéder comme suit :



Brancher la prise du transformateur 220V alternatif.





Connecter la fiche jack du transformateur sur le chargeur.



Connecter la batterie sur le chargeur en sélectionnant la prise qui convient.

**※ Si la connexion n'est pas stable, débrancher l'ensemble complet et le rebrancher de nouveau.**

### 2.4.6.2. Statut de charge

	
<p style="text-align: center;"><b>LED = Rouge</b> Batterie en charge</p>	<p style="text-align: center;"><b>LED = Vert</b> Batterie chargée</p>

**※ Utiliser uniquement le chargeur fourni.**

**※ L'utilisation de tout autre chargeur peut causer de graves dégâts.**



### 2.4.6.3. Changer de batterie sans éteindre DARwin-OP

Une particularité de DARwin-OP est sa capacité de changer de batterie sans l'éteindre.

Pour changer de batterie, procéder comme suit :

<p>Connecter la fiche jack du transformateur sur la prise jack d'alimentation.</p>	
<p>Déconnecter le câble de la batterie.</p>	
<p>Dévisser la vis et enlever le capot du compartiment de la batterie.</p>	
<p>Enlever la batterie.</p>	



<p>Insérer une batterie chargée puis refermer le compartiment de la batterie.</p>	
<p>Coinnecter le câble de la batterie.</p>	
<p>Déconnecter la fiche jack du transformateur.</p>	

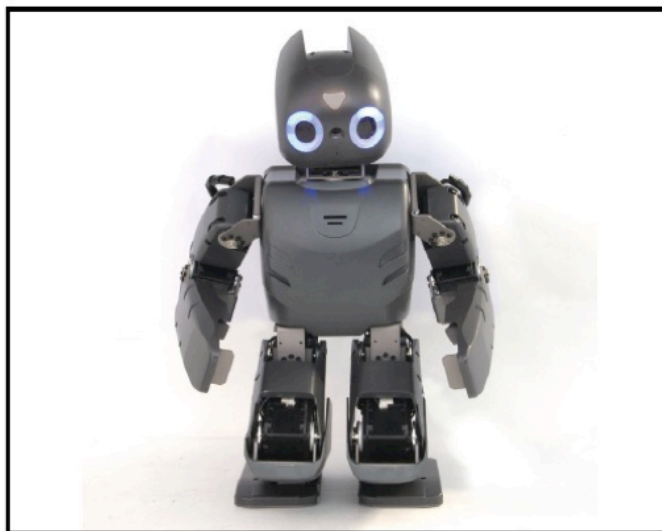
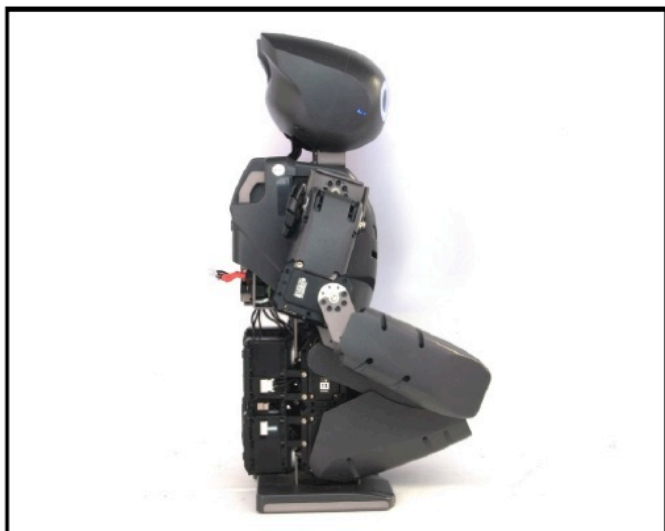
## 2.4.7. Notice d'utilisation

### 2.4.7.1. Préparation de DARwIn-OP et mise sous tension

La procédure de mise en route suivante doit être appliquée.

- **ÉTAPE 1 : Mettre DARwIn-OP dans la position agenouillée comme indiqué ci-dessous**

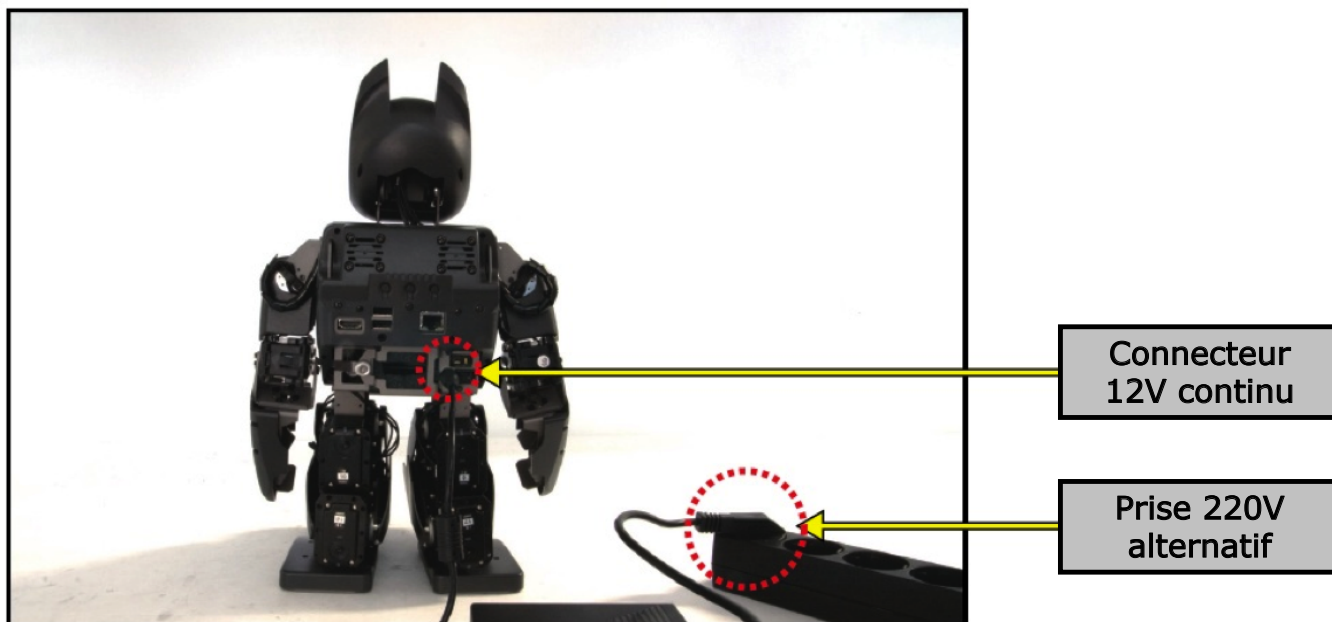




✘ **Disposer DARwIn-OP dans n'importe quelle autre position peut causer des dégats.**

■ **ÉTAPE 2-1 : Brancher l'alimentation en courant continu (12V)**

Brancher la prise de courant 220V alternatif (AC) sur le secteur puis le connecteur 12V continu sur la prise d'alimentation correspondante de DARwIn-OP. Cette prise se trouve à l'arrière de DARwIn-OP, en bas et à droite.



■ **ÉTAPE 2-2 : Utiliser la batterie (11,1V) à la place de l'alimentation 12V**

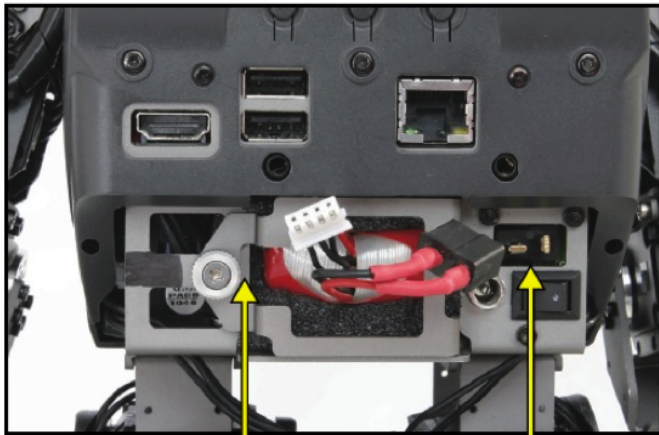
Vous pouvez utiliser aussi bien la batterie que l'alimentation 12V. Si vous voulez utiliser la batterie, vous devez :

1. vérifier que la batterie est chargée ;
2. ouvrir le compartiment de la batterie (dévisser la vis et enlever le couvercle), insérer la batterie puis remettre le couvercle et revisser la vis ;



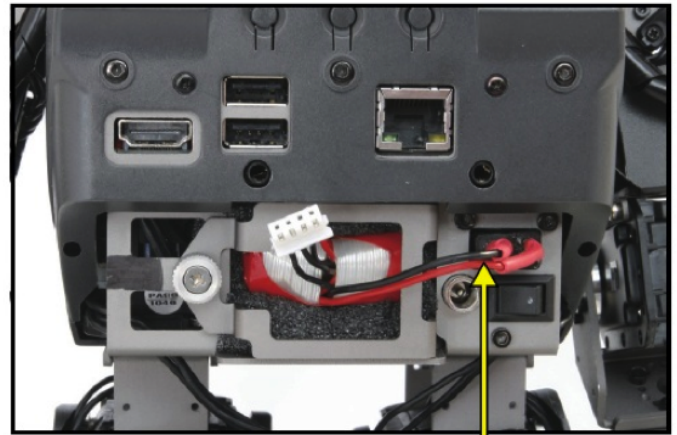


3. connecter le câble de la batterie sur la prise de batterie correspondante ;
4. débrancher la fiche de l'alimentation 12V.



Couvercle du compartiment de la batterie

Prise de batterie

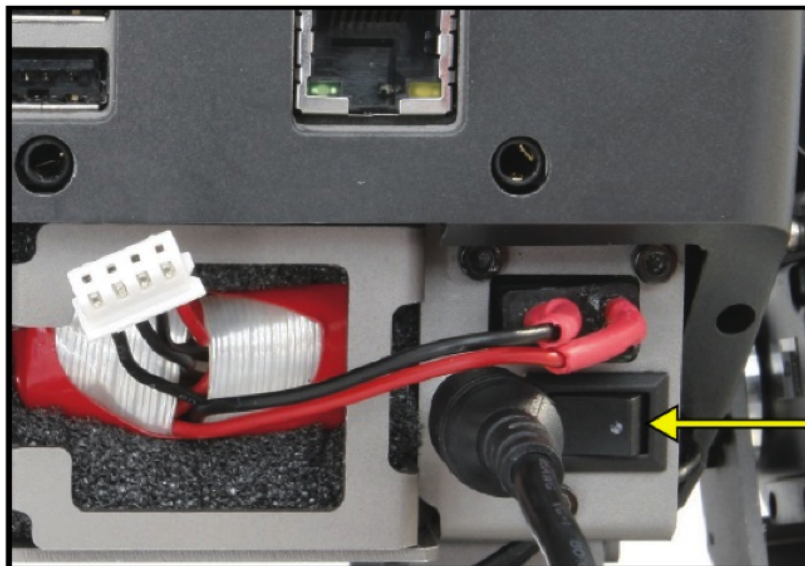


Câble de la batterie

※ **Pour empêcher tout arrêt inattendu, veuillez vous assurer que DARwIn-OP est toujours bien branché sur une alimentation active.**

■ **ÉTAPE 3 : Mise en route**





Appuyer sur l'interrupteur à bascule situé à l'arrière de DARwIn-OP, en bas et à droite.



Bouton M/A (interrupteur à bascule)



Lorsque la mise en route s'effectue, le robot passe séquentiellement par les états suivants :

États	Actions du robot
 Les LED des yeux s'éclairent	Le robot est sous tension.
 La LED de tête s'éclaire en vert	Le système d'exploitation du PC interne à DARwIn-OP démarre.
 La LED de tête passe du vert au jaune	Le programme de démonstration est chargé et est prêt à s'exécuter.
	Toutes les LED au dos de DARwIn-OP sont allumées, en attente du choix d'un programme de démonstration.



### 2.4.7.2. Exécution des programmes de démonstration

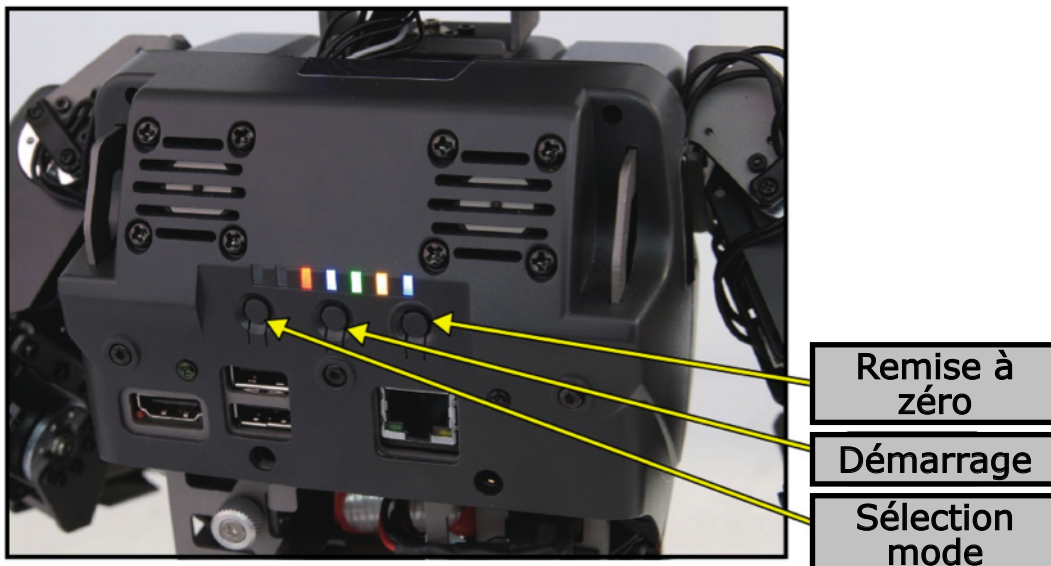
DARwIn-OP est pré-configuré selon 4 modes de fonctionnement :

- ✓ Mode “Démonstration”
- ✓ Mode “Football autonome”
- ✓ Mode “Mouvement interactif”
- ✓ Mode “Traitement vision”

À la mise en route, le mode par défaut est “Démonstration”. Ce mode n’est pas un mode à proprement parler car, en fait, il permet d’accéder à l’un des 3 autres modes.

Pour basculer d'un mode à l'autre, appuyer sur le bouton “Sélection mode”. À chaque pression, DARwIn-OP annonce le mode qui est aussi indiqué par l'allumage d'une LED.

Pour exécuter le mode choisi, appuyer sur le bouton “Démarrage”. DARwIn-OP se redresse alors et commence l'exécution.



#### ※ *Bouton de remise à zéro (RESET button)*

Le bouton de remise à zéro réinitialise les moteurs mais pas le robot entier. Il faut garder en mémoire que le programme de démonstration continue à tourner dans le contrôleur principal.

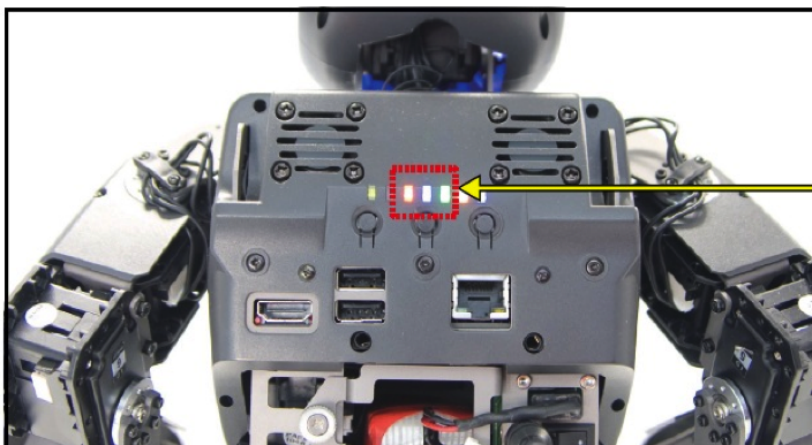
**Il est fortement recommandé que DARwIn-OP soit dans la position agenouillée ou tenu en l'air par la poignée de transport avant d'appuyer sur le bouton de remise à zéro. Appuyer sur ce bouton alors que DARwIn-OP est en mouvement peut causer des dégâts dans les moteurs ou sur le robot lui-même.**

#### ■ **Exécuter le mode “Démonstration”**

Le mode “Démonstration” est le mode par défaut. Les LED 1 (rouge), LED 2 (bleue) et LED 3 (verte) sont allumées ; la LED de tête passe du vert au jaune et DARwIn-OP annonce « *mode Démonstration* ». Il est prêt à démarrer.

DARwIn-OP reste en position agenouillée et ne bouge pas sous ce mode. C’est le mode le mieux adapté pour changer la source d’alimentation.





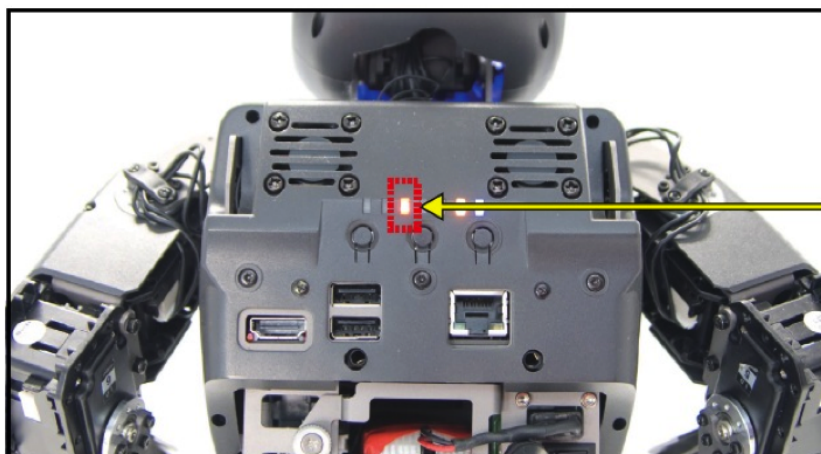
LED 1	<b>Rouge</b>
LED 2	<b>Bleu</b>
LED 3	<b>Vert</b>
Allumées	

### ■ Exécuter le mode “Football autonome”

Dans ce mode, DARwIn-OP poursuit et donne des coups de pied dans une balle rouge (la couleur de la balle peut être changée), en jouant au football tout seul.

Quand DARwIn-OP tombe, autant sur le dos que sur le ventre, il se lève, reprend la recherche de la balle et continue.

- ✓ Pour démarrer le mode “Football autonome”
  - Appuyer sur le bouton “Sélection mode” pour allumer la LED 1 (rouge). DARwIn-OP annonce « *mode Football autonome* ».



LED 1	<b>Rouge</b>
Allumée	

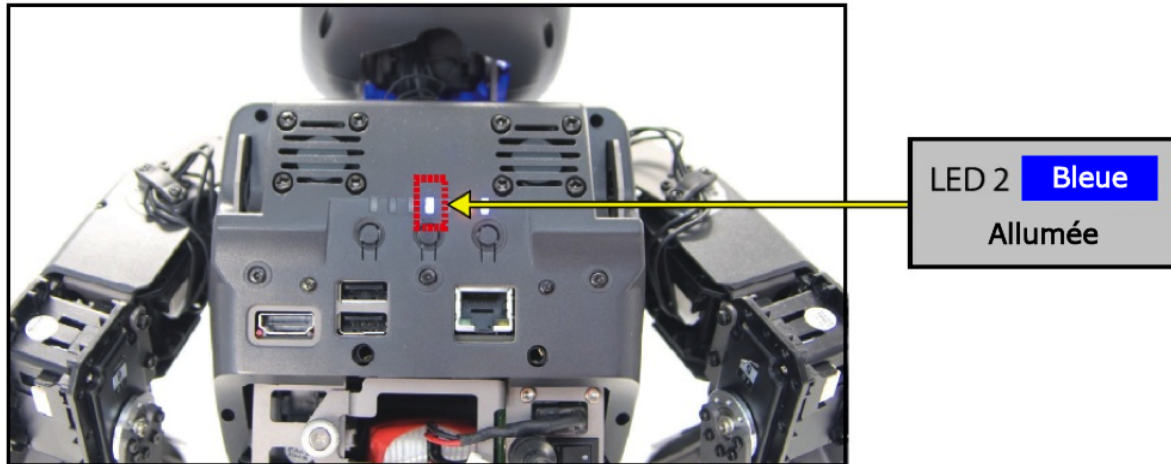
- Appuyer sur le bouton “Démarrage”. DARwIn-OP se relève et annonce « *démarrage football autonome* ».
- Juste après, si DARwIn-OP annonce le message « *calibration capteur effectuée* », c’est qu’il est prêt pour exécuter le mode “Football autonome”. S’il annonce le message « *calibration capteur échouée* », il restera debout tant que le capteur gyroscopique ne sera pas calibré correctement suivi du message « *calibration capteur effectuée* ». Si la calibration ne peut pas s’effectuer, lancer à nouveau le programme de démonstration.
- Quand DARwIn-OP voit la balle, il se déplace vers elle. Lorsque la balle est proche, il lui donne un coup soit avec le pied gauche, soit avec le pied droit. Si DARwIn-OP chute pendant la poursuite de la balle ou pendant le coup de pied, il se relève.
- ✓ Pour arrêter le mode “Football autonome”
  - Appuyer sur le bouton “Sélection mode”. DARwIn-OP retourne alors dans le mode « *Démonstration* ».











■ **Exécuter le mode “Mouvement interactif”**

Dans ce mode, DARwIn-OP exécute séquentiellement des mouvements pré-programmés pendant qu’il parle.

- ✓ Pour démarrer le mode “Mouvement interactif”
  - Appuyer sur le bouton “Sélection mode” pour allumer la LED 2 (bleue). DARwIn-OP annonce « *mode interactif* ».



- Appuyer sur le bouton “Démarrage”. DARwIn-OP se lève et annonce « démarrage du *mode interactif* ».
- DARwIn-OP effectue alors les actions séquentielles suivantes :

 1 - Merci	 2 - Introduction	 3 - Woua!	 4 - Je m’assois
 5 - Je me lève	 6 - On applaudit !	 7 – Oops !	 8 – Au revoir

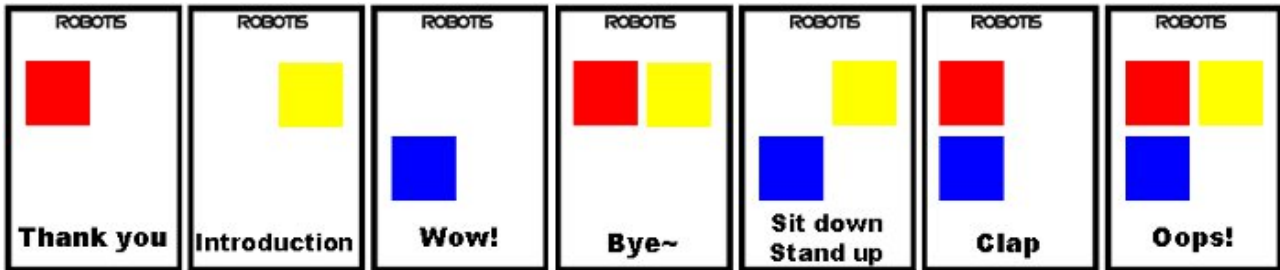
- ✓ Pour arrêter le mode “Mouvement interactif”
  - Appuyer sur le bouton “Sélection mode”. DARwIn-OP retourne alors dans le mode « *Démonstration* ».



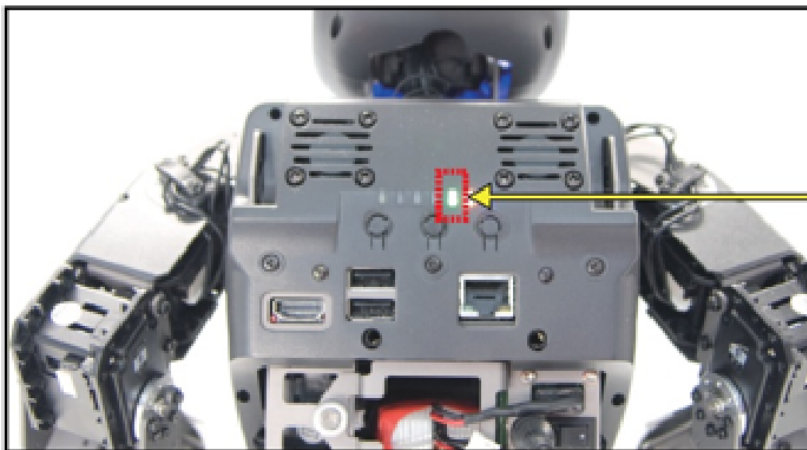


■ **Exécuter le mode “Traitement vision”**

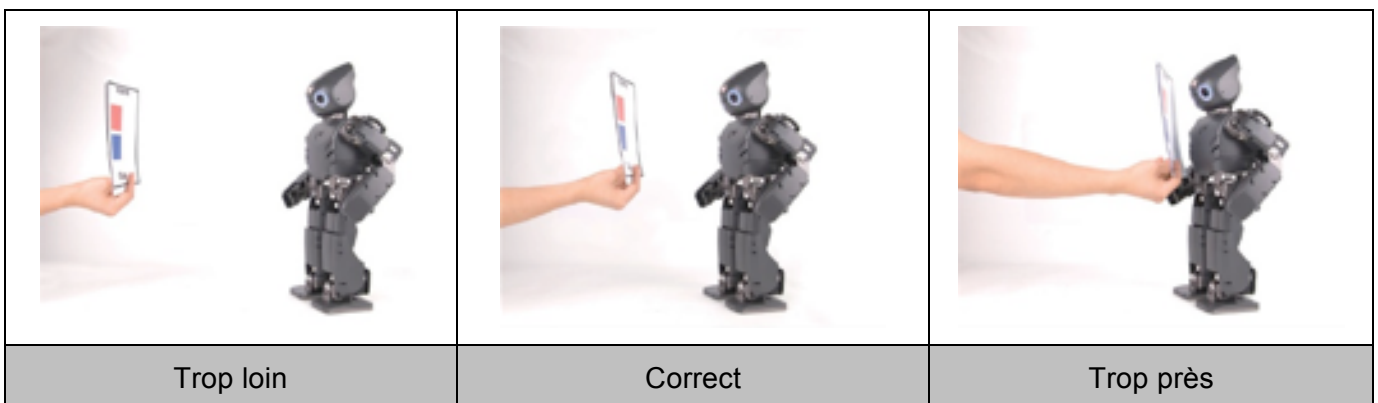
Dans ce mode, DARwIn-OP exécute les mêmes mouvements que dans le mode “Mouvement interactif” mais individuellement en fonction des cartes lues. Utiliser pour cela les cartes en couleurs fournies :



- ✓ Pour démarrer le mode “Traitement vision”
  - Appuyer sur le bouton “Sélection mode” pour allumer la LED 3 (verte). DARwIn-OP annonce « *mode traitement vision* ».



- Appuyer sur le bouton “Démarrage”. DARwIn-OP se lève et annonce « démarrage du *mode traitement vision* ».
- Sélectionner une carte en couleur et placez-la en face de DARwIn-OP à environ 15cm.



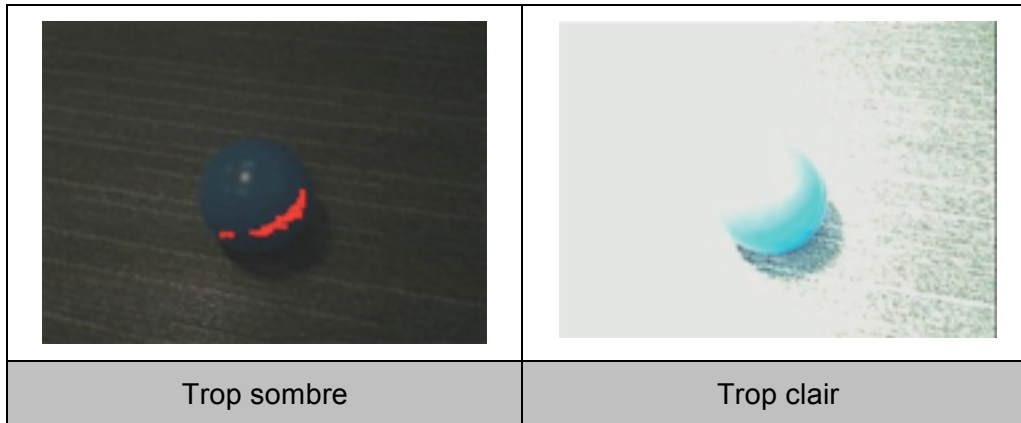
- ✓ Pour arrêter le mode “Traitement vision”
  - Appuyer sur le bouton “Sélection mode”. DARwIn-OP retourne alors dans le mode « *Démonstration* ».

Si DARwIn-OP rencontre des difficultés pour lire la carte, vous devez ajuster la couleur et la balance des blancs. Voir le paragraphe suivant pour plus de détail.



### 2.4.7.3. Calibration de la couleur et de la balance des blancs

Si l'éclairage autour de DARwIn-OP est trop clair ou trop sombre, il peut avoir des difficultés pour lire.

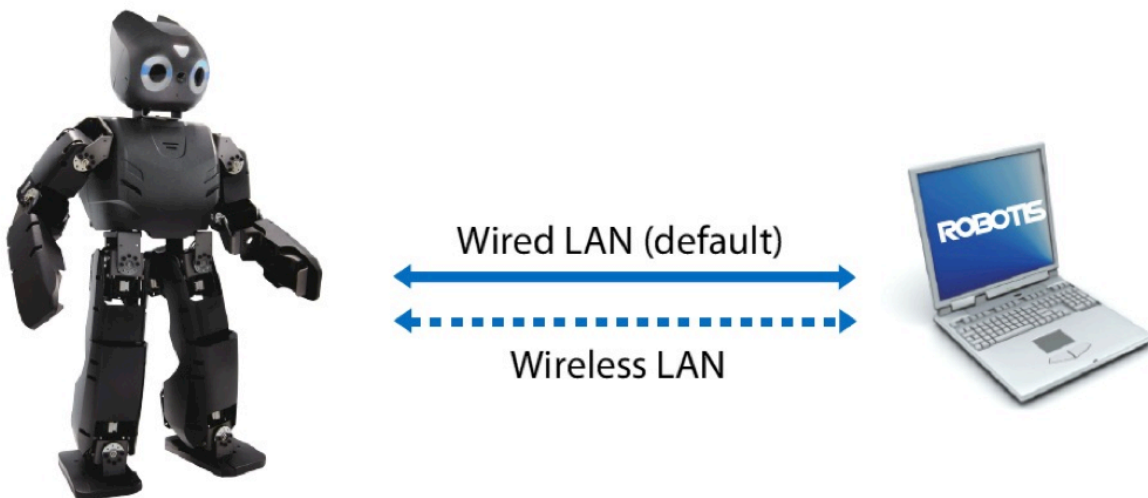


Vous devez changer les paramètres de la couleur et de la balance des blancs ?

Vous devez modifier les paramètres de couleur de la balle pour le mode "Football autonome" ?

■ **Pour réaliser ces opérations, vous devez :**

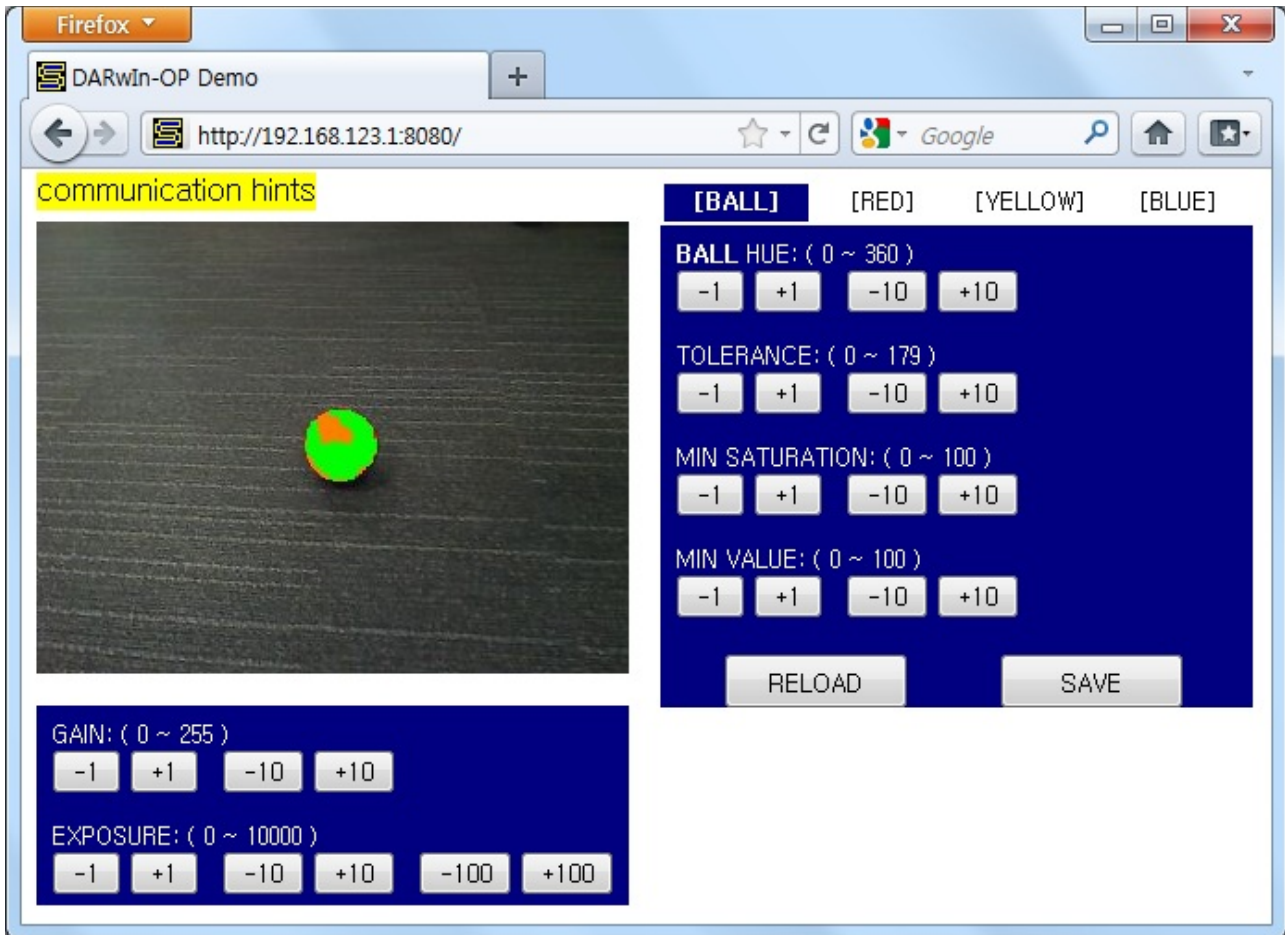
1. vous connecter à DARwIn-OP par l'intermédiaire d'un réseau local câblé ou sans fil ;



2. dans le menu des propriétés du protocole Internet (TCP/IP) 'Local area connection properties', sélectionner 'Obtain an IP address automatically' ;
3. avoir un navigateur Internet (Firefox recommandé, Chrome risque de créer un manque de mémoire).

**Pour réaliser une connexion par l'intermédiaire du réseau local et accéder au réglage des couleurs, suivre la procédure suivante :**

1. vérifier que la connexion avec DARwIn-OP est bien réalisée ;
2. lancer votre navigateur ;
3. taper <http://192.168.123.1:8080> dans la zone de saisie des adresses ;
4. dans la fenêtre, cliquer sur les boutons d'incrément/décément tant que vous n'avez pas obtenu les réglages désirés. Les changements sont visibles en direct.



Les valeurs suivantes sont les valeurs par défaut :

	Ball	Rouge	Jaune	Bleu
Hue (0~360)	356	0	60	225
Tolerance (0~180)	15	15	15	15
Minimum Saturation (0~100)	50	55	45	45
Minimum Value (0~100)	10	0	0	0
Gain (0~255)	255	255	255	255
Exposure (0~10000)	1000	1000	1000	1000

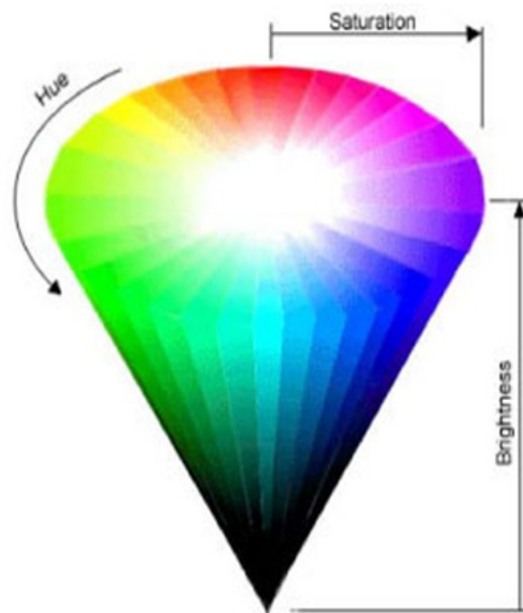
### ■ *Échelle de couleurs HSV*

Le modèle de couleurs **HVS** (**H**ue [teinte], **S**aturation, **V**aleur), connu aussi sous **HSB** (**H**ue, **S**aturation, **B**rightness [luminosité]), définit une échelle de couleurs en termes de trois composantes :



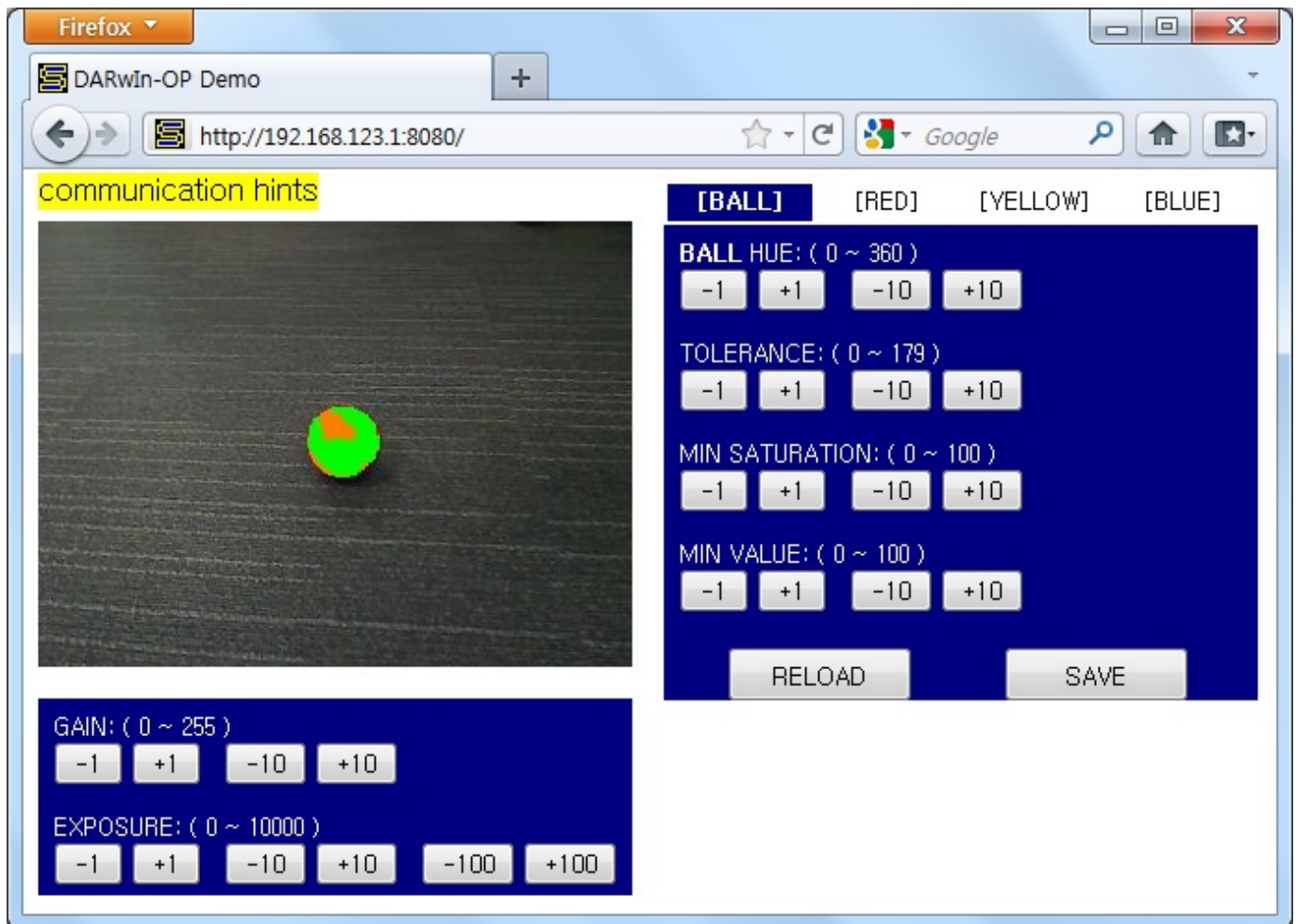
*Échelle des teintes*

- ✓ Hue [teinte]
  - Le type de couleur (rouge, bleu ou jaune)
  - Gamme de valeurs 0-360 (mais normalisée à 0-100% dans certaines applications)
- ✓ Saturation (chroma)
  - Le “dynamisme” de la couleur
  - Gamme de valeurs 0-100%
  - Appelée quelquefois “pureté” par analogie avec la pureté colorimétrique
  - Plus la saturation d’une couleur est faible, plus son apparence sera estompée ; ainsi, il est utile de définir la désaturation comme étant la qualité inverse de la saturation
- ✓ Valeur (intensité)
  - La luminosité de la couleur
  - Gamme de valeurs 0-100%





■ **Comment savoir si la calibration de la couleur de la balle est correctement ajustée à la couleur désirée ?**



L'illustration ci-dessus représente la balle couverte d'une tâche orange et verte en bas à droite.

- ✓ Orange signifie que la calibration de la couleur est correctement appariée à la couleur désirée de la balle. Dans le cas particulier de la figure, la couleur de la balle est bleue.

Cependant, il y a aussi une tâche verte dans la partie basse et droite de la balle.

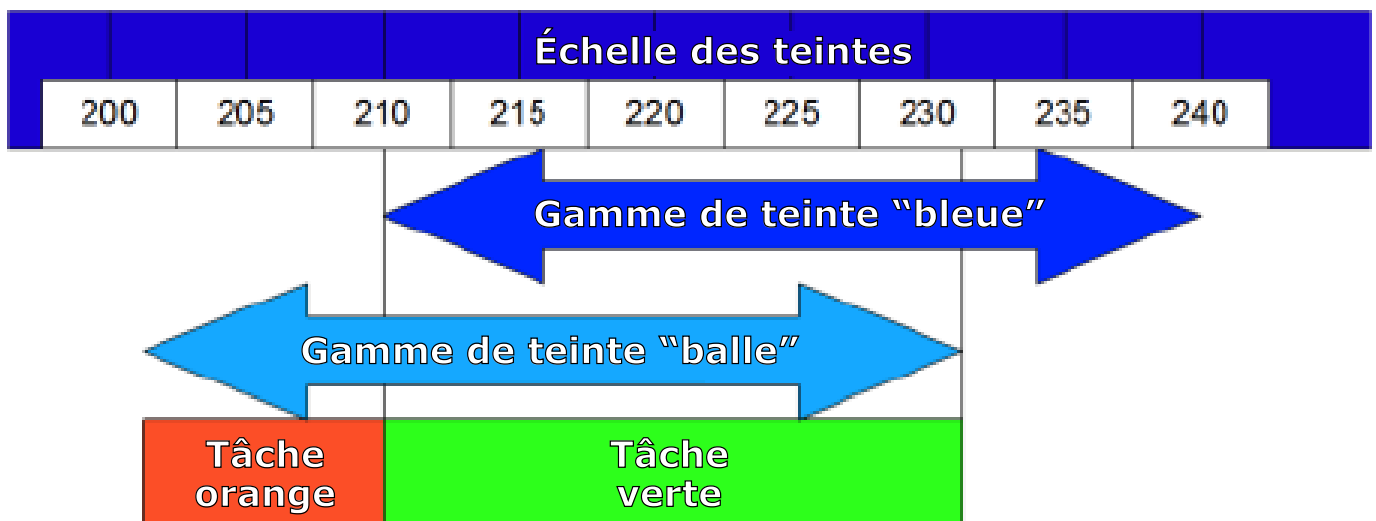
- ✓ Vert signifie qu'il y a chevauchement entre la plage de tolérance de teinte des paramètres de la balle et de l'un des autres paramètres de couleur (bleu dans ce cas).

Se souvenir que la valeur de la teinte bleue par défaut est 225 avec une tolérance de 15, ce qui donne une fourchette de valeurs de 210 à 240.

Dans cet exemple, la valeur de la teinte de la balle a été fixée à 216 (balle bleue) avec une tolérance de 15. Cela donne une fourchette de teinte de balle de 201 à 231.

Les valeurs entre 210 et 231 se chevauchent. Cela signifie que la valeur de la teinte dans la partie basse et droite de la balle est comprise entre 210 et 231.

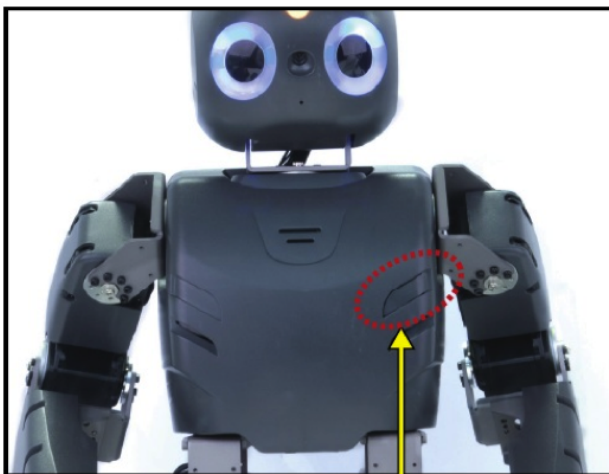




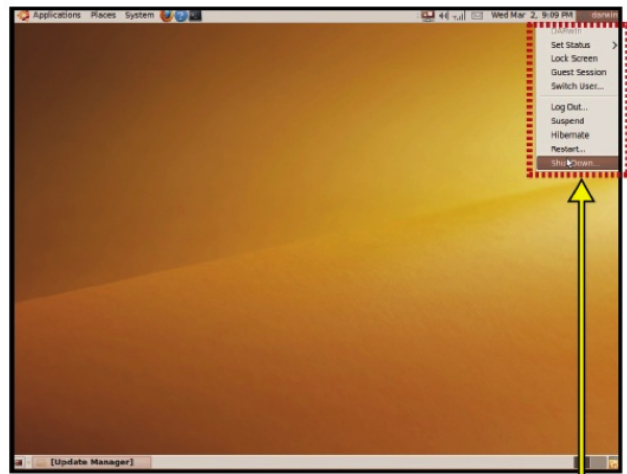
#### 2.4.7.4. Arrêter DARwIn-OP

Pour arrêter l'ordinateur de DARwIn-OP, effectuer l'une des opérations suivantes :

- ✓ Appuyer 3 fois sur le bouton creux (1<sup>er</sup> creux) situé sur le devant, côté gauche de la poitrine.
- ✓ Si vous pouvez connecter un moniteur et voir l'écran, cliquer sur "darwin" (situé en haut à droite de l'écran) puis sélectionner "Shut Down".



Bouton d'arrêt

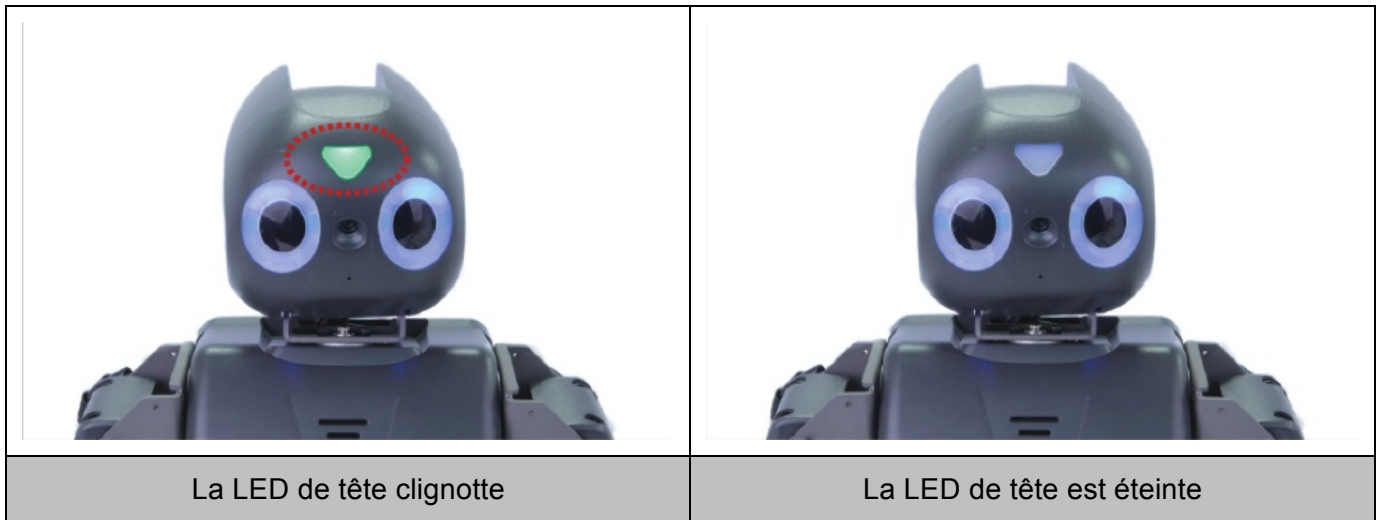


Shut Down

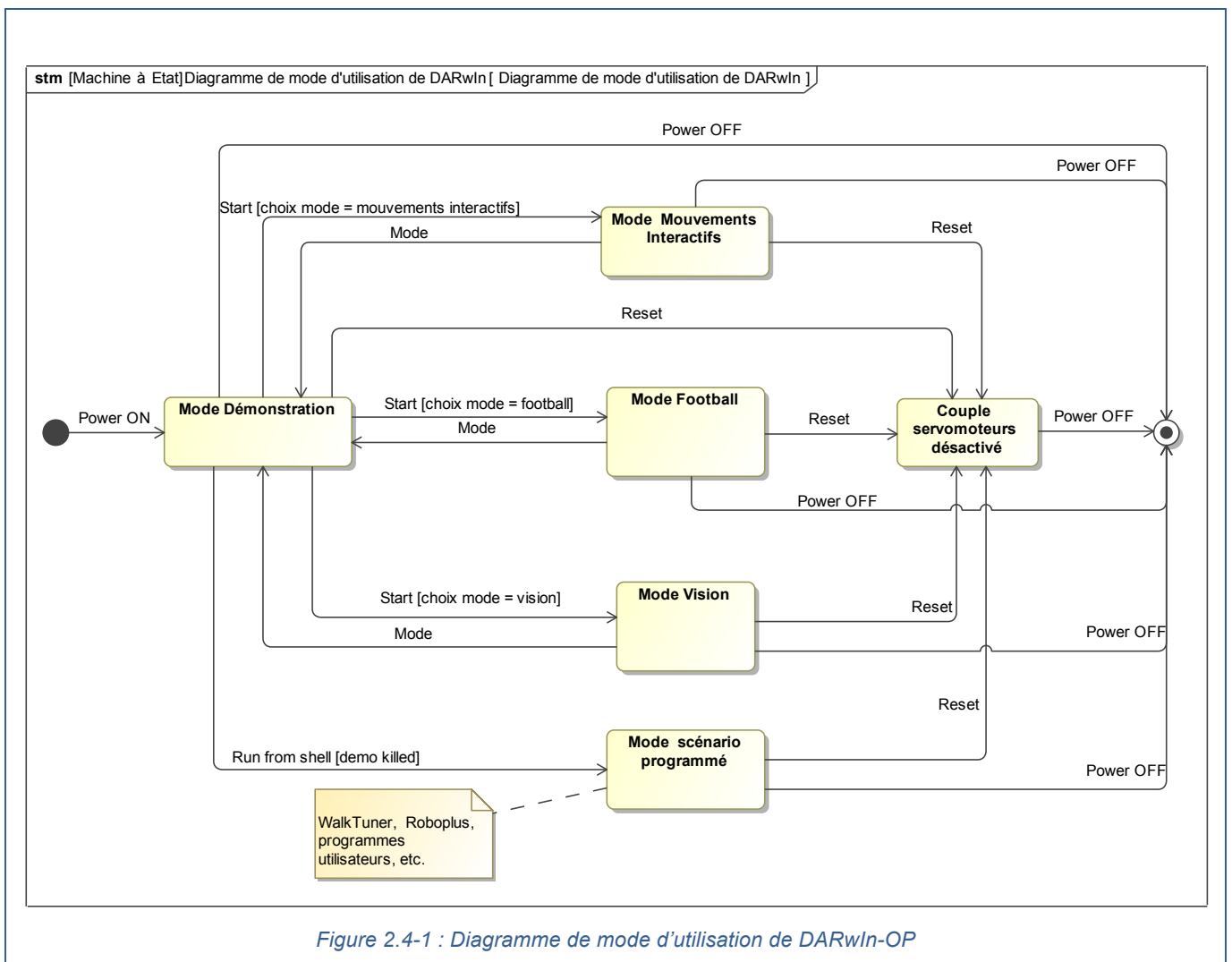
DARwIn-OP annonce « *bye-bye* » et démarre la procédure d'arrêt :

- ✓ La LED de tête de DARwIn-OP clignote puis s'éteint (OS arrêté). Lorsque la LED s'arrête de clignoter et reste éteinte, vous devez couper l'alimentation. Les LED des yeux restent éclairées. Ne pas en tenir compte et couper l'alimentation.

**⚠ Si vous coupez l'alimentation pendant que la LED de tête clignote, alors DARwIn-OP risque d'avoir des problèmes au prochain redémarrage.**



**2.4.7.5. Récapitulatif : diagramme de mode d'utilisation de DARwIn-OP**





## 2.4.8. Entretien de l'équipement

### 2.4.8.1. Nettoyage

Il est nécessaire de nettoyer le robot DARwIn-OP au moins une fois tous les mois.

En effet, bien que la poussière ne puisse pas nuire aux composants internes du contrôleur principal et du contrôleur secondaire, un environnement humide et l'électricité statique peuvent provoquer des dommages importants aux deux contrôleurs.

La poussière, la saleté peuvent être enlevées avec un chiffon sec, un nettoyant à l'air comprimé en bombe ou une brosse douce non métallique.

### 2.4.8.2. Maintenance préventive

ROBOTIS et DMS recommandent une maintenance préventive toutes les 20 heures d'utilisation environ.

#### ■ **Maintenance mécanique.**

Afin de garder les parties mécaniques en fonctionnement optimal, il est nécessaire de réaliser cette maintenance préventive. La négligence peut entraîner des dommages coûteux et inutiles.

Pour réaliser cette maintenance préventive, il est nécessaire d'avoir une table ou établi propre, sec et bien éclairé.

Il est donc recommandé que chaque vis et chaque écrou soit serré correctement. Un serrage excessif n'est pas nécessaire et il peut même provoquer des dommages sur les filetages des vis et des écrous, sur les têtes de vis et sur l'outillage.

#### ■ **Démontage de DARwIn-OP**

- ✓ Enlever la batterie.
- ✓ Enlever tous les carters du Robot DARwIn-OP.
- ✓ Si le robot DARwIn-OP est équipé des capteurs de pression sous les pieds, vous devez également enlever les protections sous les pieds.
- ✓ Démontez le robots DARwIn-OP en suivant strictement la procédure de dépannage ci-dessous.
  - a. Démontage du haut du corps ainsi que le cou (ID 1,2 et 19), du contrôleur secondaire et des 4 connexions inter-servomoteurs :
    - débrancher toute les connexions entre le contrôleur principal et le contrôleur secondaire ;
    - débrancher tous les câbles entre les bras et le haut du corps ;
    - retirer les bras de la partie haut du corps.
  - b. Démontez la tête et la caméra USB.
  - c. Démontez chaque bras (ID 3 et 5 pour la droite, ID 4 et 6 pour la gauche) ainsi que les pièces d'assemblages (main, coude / épaule et actionneur latéral).
  - d. Démontez le torse (ID 7, 8, 9, 10) ainsi que les pièces d'assemblage pour le contrôleur principal, les deux pare chocs, le connecteur de puissance sur la carte électronique, l'interrupteur de puissance, le bas du dos, les deux hanches et les 4 connexions inter-servomoteurs :
    - débrancher les câbles des jambes reliés au torse ;
    - retirer les jambes du torse ;
    - le contrôleur principal n'a pas besoin d'être détaché du torse.
  - e. Démontez chaque cuisse (ID 11 pour la cuisse droite, ID 12 pour la cuisse gauche) ainsi que les pièces d'assemblage des deux servomoteurs latéraux et des cuisses.
  - f. Démontez les pièces d'assemblage du genou et démontez la jambe (ID 13, 15 pour



- la droite, ID 14, 16 pour la gauche).
- g. Démontez les pieds (ID 17 pour la droite et ID 18 pour la gauche) ainsi que les pièces d'assemblage pied, cheville.

#### ■ **Maintenance électrique.**

- ✓ Assurez-vous que toutes les connexions sont correctement fixées.
- ✓ Assurez-vous que l'ensemble des câbles soit bien à plat sans boucle.
- ✓ Remplacer les câbles lorsque la couche isolante présente une coupure qui laisse le ou les fils à l'air.

**Les manuels de montage mécanique et de câblage électrique sont disponibles dans le dossier ressource.**

### 2.4.9. Sécurité

#### 2.4.9.1. Consignes de sécurité

- ✓ Avant d'utiliser DARwIn-OP, vérifiez que les câbles électriques de raccordement entre les servomoteurs ne sont pas coincés.
- ✓ Ne pas utiliser le robot DARwIn-OP dans un environnement trop humide ou sous la pluie.
- ✓ Ne pas utiliser le robot DARwIn-OP dans un environnement qui pourrait lui provoquer des grosses chutes (escalier, etc.)
- ✓ Ne pas utiliser le robot DARwIn-OP sans ses habillages.
- ✓ Ne pas utiliser d'autres fusibles que ceux préconisés dans la présente notice.

#### 2.4.9.2. Procédure de consignation

### 2.4.10. Maintenance corrective


#### 2.4.10.1. Remplacement du fusible

Il est possible que lorsque vous appuyez sur le bouton marche / arrêt au dos du robot DARwIn-OP, il ne se passe rien et ce indépendamment de la source d'alimentation (sur secteur via l'adaptateur ou sur batterie). Si c'est le cas, il s'agit probablement du fusible qui est défectueux ou grillé.

Le fusible est un 10 ampère DC sous 12V.

Pour remplacer le fusible, il est nécessaire de démonter le carter arrière du corps principal du robot DARwIn-OP. Le fusible se trouve sur le contrôleur secondaire CM-730 à côté des alimentations (fils rouge et noir). Attention, le démontage doit être réalisé avec le cordon d'alimentation externe enlevé et la batterie enlevée également.

Retirer avec précautions le fusible avec une pince à épiler et insérer délicatement le nouveau fusible.

 **Attention, un mauvais démontage ou remontage du fusible peut entraîner des dommages sur les autres composants du contrôleur secondaire CM-730.**

#### 2.4.10.2. Remplacement d'un servomoteur

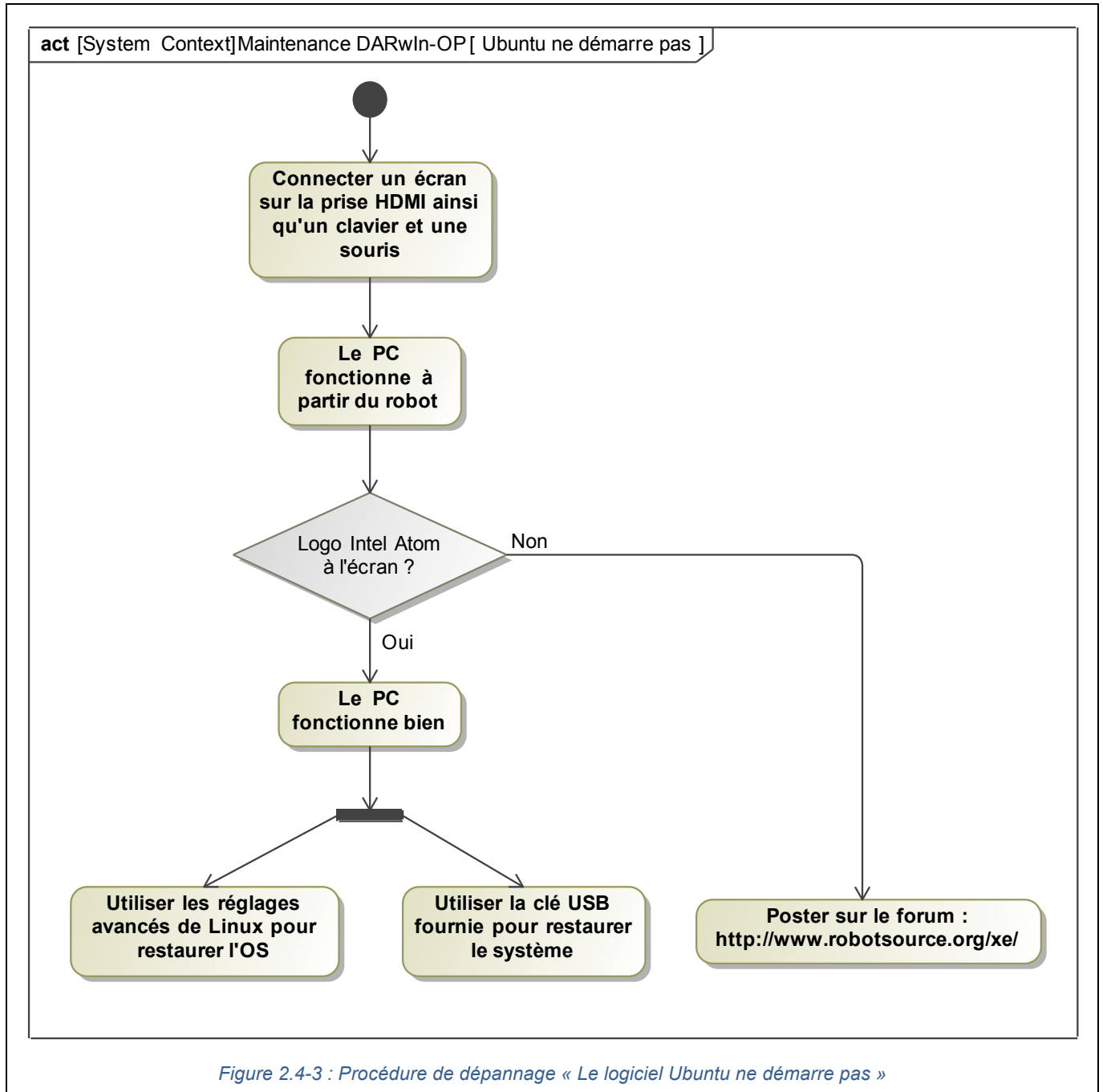






■ **Le logiciel Ubuntu ne démarre pas**

Si le logiciel Ubuntu ne démarre pas, suivre la procédure de dépannage ci-dessous.



■ **Le programme de démonstration ne fonctionne pas à la mise en service**

Si le programme de démonstration ne fonctionne pas à la mise en service, suivre la procédure de dépannage de la page suivante.

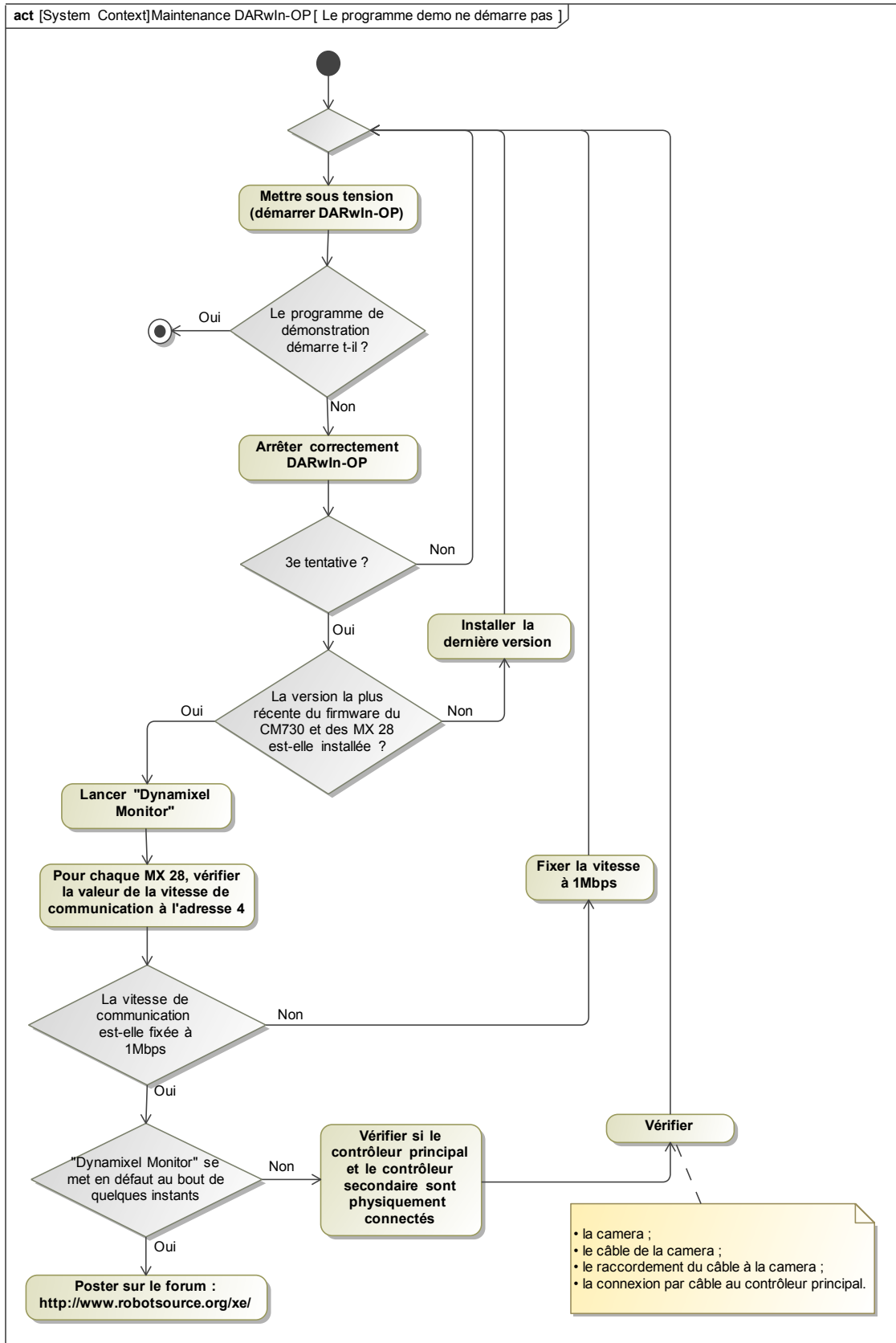


Figure 2.4-4 : Procédure de dépannage « Le programme de démonstration ne fonctionne pas à la mise en service »



Lors de la vérification des connexions, il peut se produire le cas où le contrôleur principal de DARwin-OP perd la connexion avec la caméra.

Veillez vous assurer que le câble USB de la caméra est correctement connecté à la caméra et au contrôleur principal.

Pour vérifier la connexion de la caméra avec Linux OS procédez comme suit :

- ✓ sur une fenêtre de terminal aller dans le répertoire '/dev' ;
- ✓ saisissez ls et vous verrez une liste des périphériques, puis rechercher « video0 ». C'est la caméra en tant que périphérique USB.

```

root@darwin:/dev# ls
adsp          loop0          ram0          sda2          tty16         tty36         tty56         usbmon4
agpgart       loop1          ram1          sda5          tty17         tty37         tty57         v4l
audio         loop2          ram10         sequencer     tty18         tty38         tty58         vcs
audio1        loop3          ram11         sequencer2    tty19         tty39         tty59         vcs1
block         loop4          ram12         serial        tty2          tty4          tty6          vcs2
bus           loop5          ram13         sg0           tty20         tty40         tty60         vcs3
char          loop6          ram14         shm           tty21         tty41         tty61         vcs4
console       loop7          ram15         snapshot      tty22         tty42         tty62         vcs5
core          mapper        ram2          snd           tty23         tty43         tty63         vcs6
cpu_dma_latency mem            ram3          sndstat       tty24         tty44         tty7          vcs7
disk          mixer          ram4          stderr        tty25         tty45         tty8          vcs8
dri           mixer1        ram5          stdin         tty26         tty46         tty9          vcsa
dsp           net           ram6          stdout        tty27         tty47         ttyS0         vcsa1
dsp1          network_latency ram7          tty           tty28         tty48         ttyS1         vcsa2
ecryptfs      network_throughput ram8          tty0          tty29         tty49         ttyS2         vcsa3
fd            null          ram9          tty1          tty3          tty5          ttyS3         vcsa4
full          oldmem        ram10         random        tty10         tty30         ttyUSB0       vcsa5
fuse          port          rfc11         tty11         tty31         tty51         urandom       vcsa6
hpet          ppp           rtc           tty12         tty32         tty52         usbmon0       vcsa7
input         psaux        rtc0          tty13         tty33         tty53         usbmon1       vcsa8
kmsg          ptmx         sda           tty14         tty34         tty54         usbmon2       video0
log           pts          sda1         tty15         tty35         tty55         usbmon3       zero
root@darwin:/dev#

```

The camera as an USB device

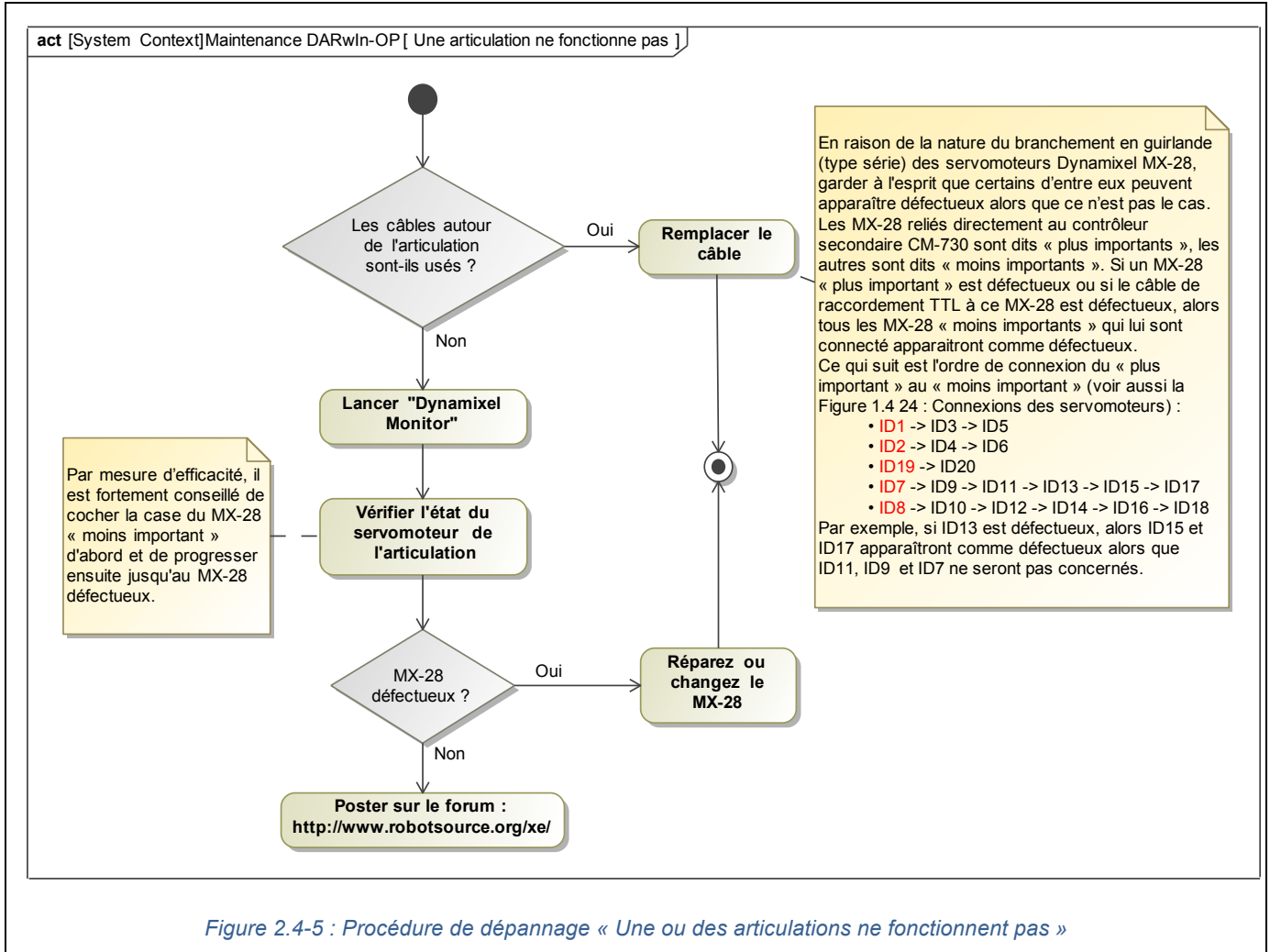
Si vous ne voyez pas le périphérique « video0 », alors vous devrez peut-être remplacer le câble de caméra et/ou la caméra. Dans ce cas, vous pouvez remplacer le câble USB, la caméra ou prendre contact avec ROBOTIS pour un arrangement.

Veillez noter que si vous perdez la connexion physique réelle (mécanique) entre le contrôleur principal et le contrôleur secondaire, vous perdez automatiquement la connexion de la caméra. C'est parce que le câble USB pour le partage entre CM-730 et caméra utilise le même connecteur Molex.



■ **Une ou des articulations ne fonctionnent pas**

Si une ou des articulations ne fonctionnent pas, suivre la procédure de dépannage ci-dessous.





■ **Le robot fait des mouvements anormaux et se comporte bizarrement**

Si le robot DARwIn-OP fait des mouvements anormaux et se comporte bizarrement, suivre la procédure de dépannage ci-dessous.

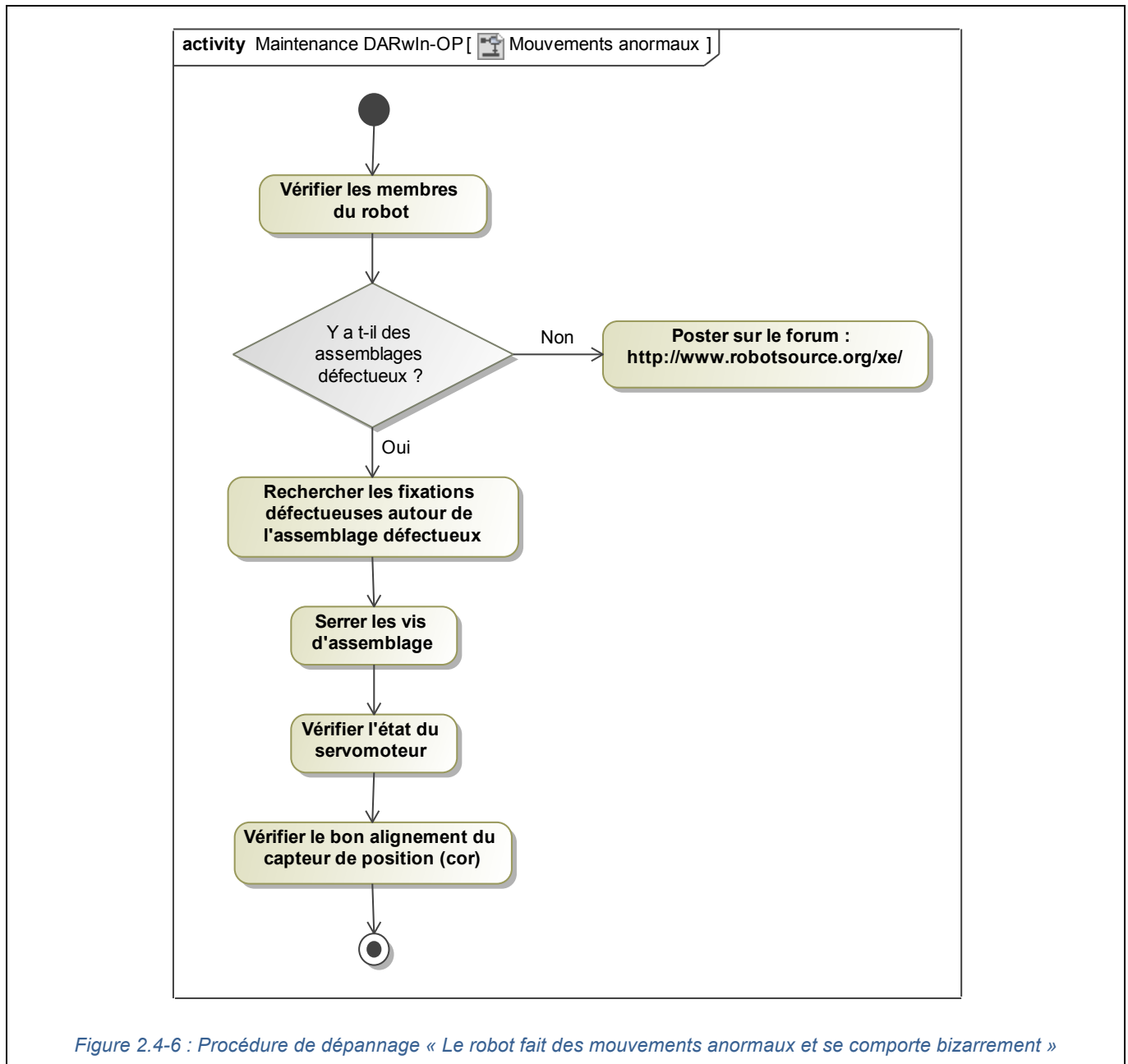


Figure 2.4-6 : Procédure de dépannage « Le robot fait des mouvements anormaux et se comporte bizarrement »

■ **Les mouvements sont lents ou instables**

Si les mouvements sont lents ou instables, trois éléments peuvent en être la cause : la batterie, le système d'exploitation Linux OS ou l'interfaçage avec DARwIn-OP.

- ✓ Batterie (si DARwIn-OP ne fonctionne que sur la batterie)

DARwIn-OP a été programmé pour émettre un bip si la batterie est faible. Toutefois, dans les environnements bruyants vous ne pouvez pas être en mesure d'entendre ces bips. Dans ce cas, surveillez les mouvements lents, étranges, ou saccadés .





✓ Linux OS

Le système d'exploitation du contrôleur principal fitPC a été optimisé de telle sorte que DARwIn-OP peut se déplacer doucement. Ceci est dû probablement au fait qu'il y a trop de programmes en cours d'exécution en arrière-plan.

Causes probables :

- a) des mises à jour automatiques s'effectuent alors que DARwIn-OP est en marche ;
- b) le bureau à distance est actif au cours du fonctionnement ;
- c) définition d'une résolution d'affichage trop haute (1080 p par exemple) ;
- d) tout programme / processus qui prend beaucoup trop de ressources fitPC ;
- e) toute combinaison de a) à d).

Veuillez garder à l'esprit qu'il s'agit d'une limitation de conception du système de l'Intel Atom et non de Linux OS, de la fitPC ou de ROBOTIS.

✓ Interfaçage avec DARwIn-OP

Nous recommandons fortement l'interfaçage avec DARwIn-OP via un terminal SSH plutôt que le bureau à distance. Toutefois, si vous ne pouvez pas vous passer de "Remote Desktop", fermez tous les programmes en tâche de fond d'Ubuntu et utilisez une résolution d'écran conventionnelle (1024x768 par exemple).

■ **Le robot ne peut pas "voir" ou "trouver" la balle**

Si le robot DARwIn-OP ne peut pas "voir" ou "trouver" la balle, il faut contrôler les réglages de balance des blancs et de couleur.

Ces réglages de balance des blancs et de couleur sont toujours un problème avec le traitement d'image : DARwIn-OP ne fait pas exception.

Vous devez toujours vérifier les conditions d'éclairage de votre environnement de travail. Dans cette optique, un programme de tests d'étalonnage de la balance des blancs et de couleur a été mis au point (voir le §2.4.7.4).

■ **Les yeux et la tête du robot ne s'allument pas, ni les LEDs à l'arrière du robot**

Si les yeux et la tête du robot DARwIn-OP ne s'allument pas, ni les LEDs à l'arrière du robot, se reporter à la section des LEDs du symptôme « Le robot ne s'allume pas ».



■ **Le robot n'a plus de son**

Si le robot DARwIn-OP n'a plus de son, suivre la procédure de dépannage ci-dessous.

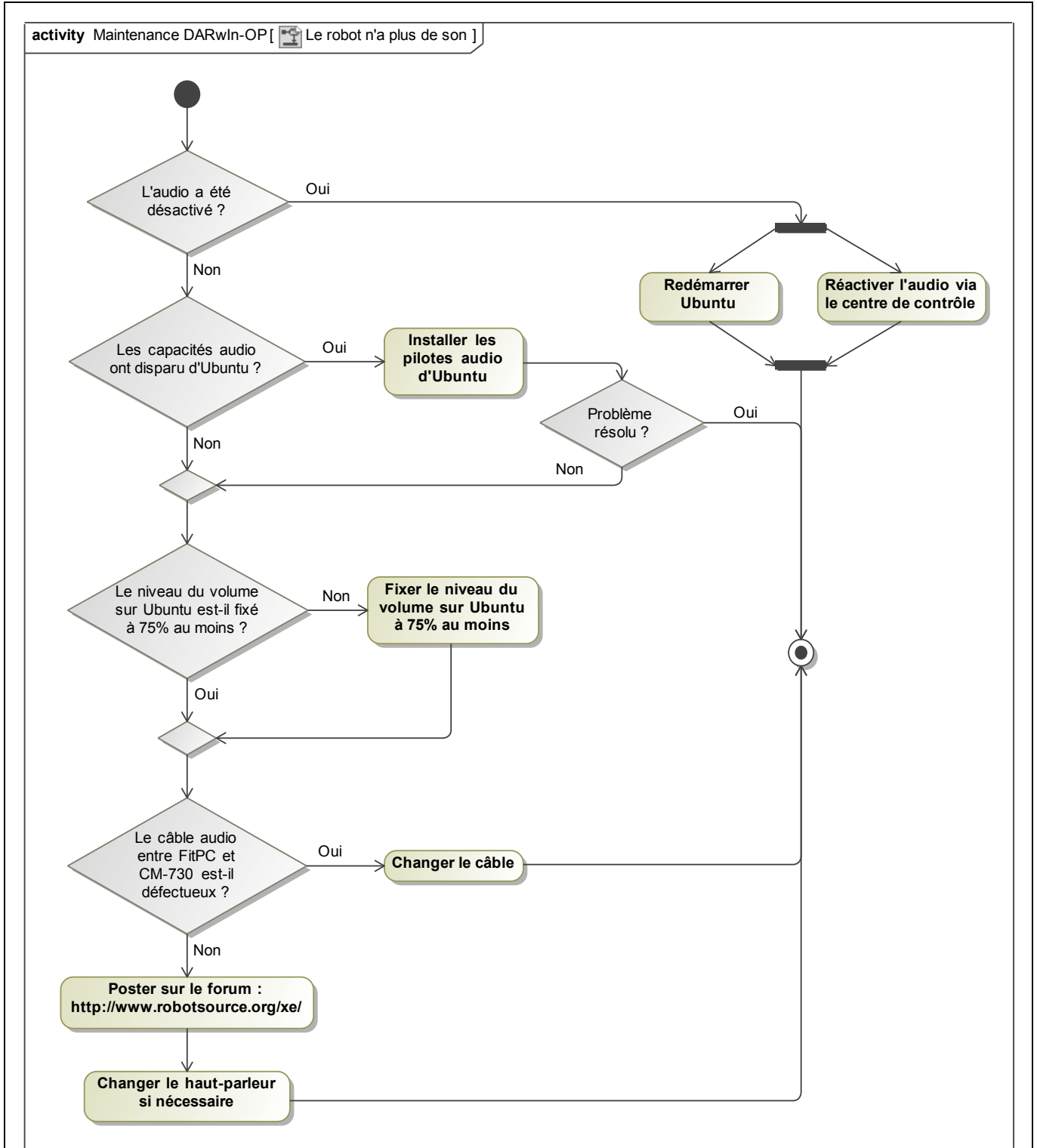


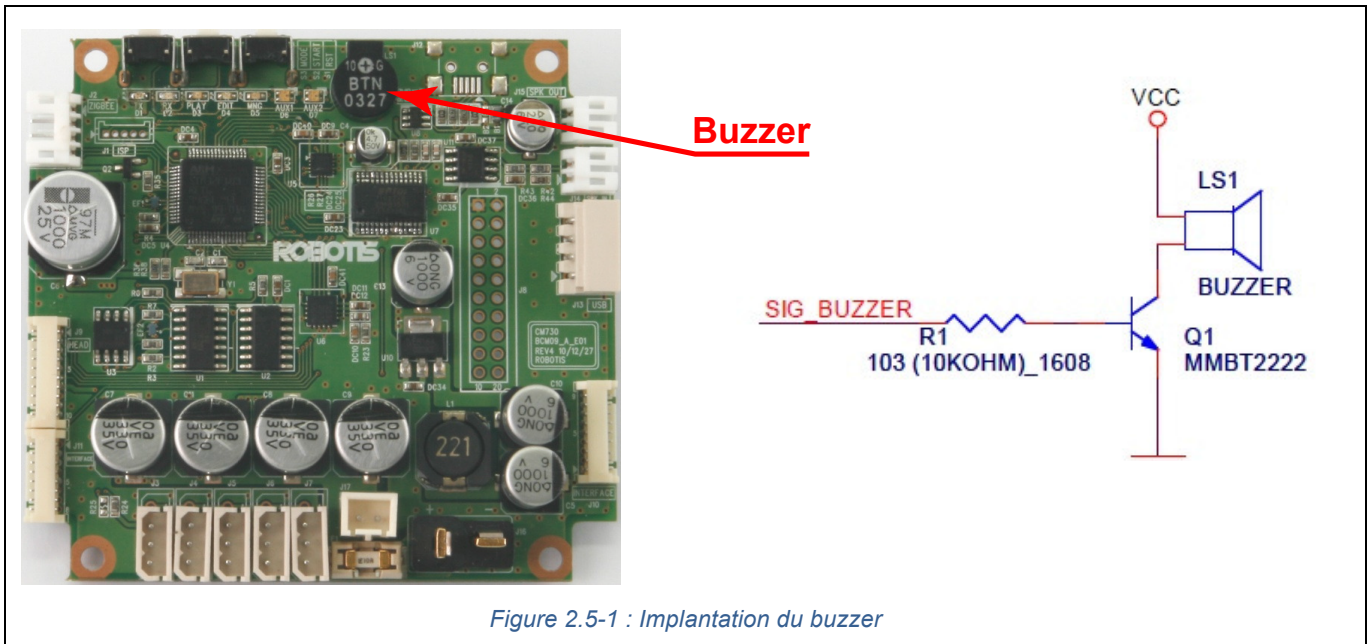
Figure 2.4-7 : Procédure de dépannage « Le robot n'a plus de son »



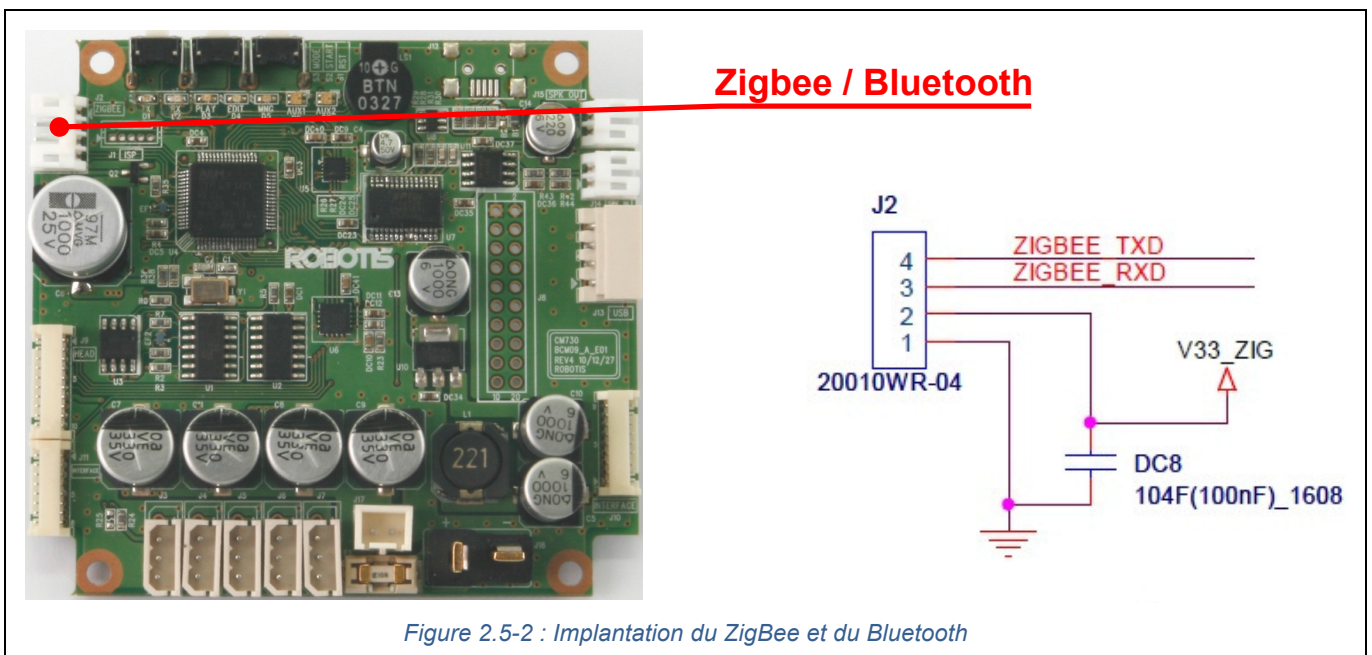
## 2.5. Documents techniques

### 2.5.1. Schémas électriques / électroniques

#### 2.5.1.1. Buzzer

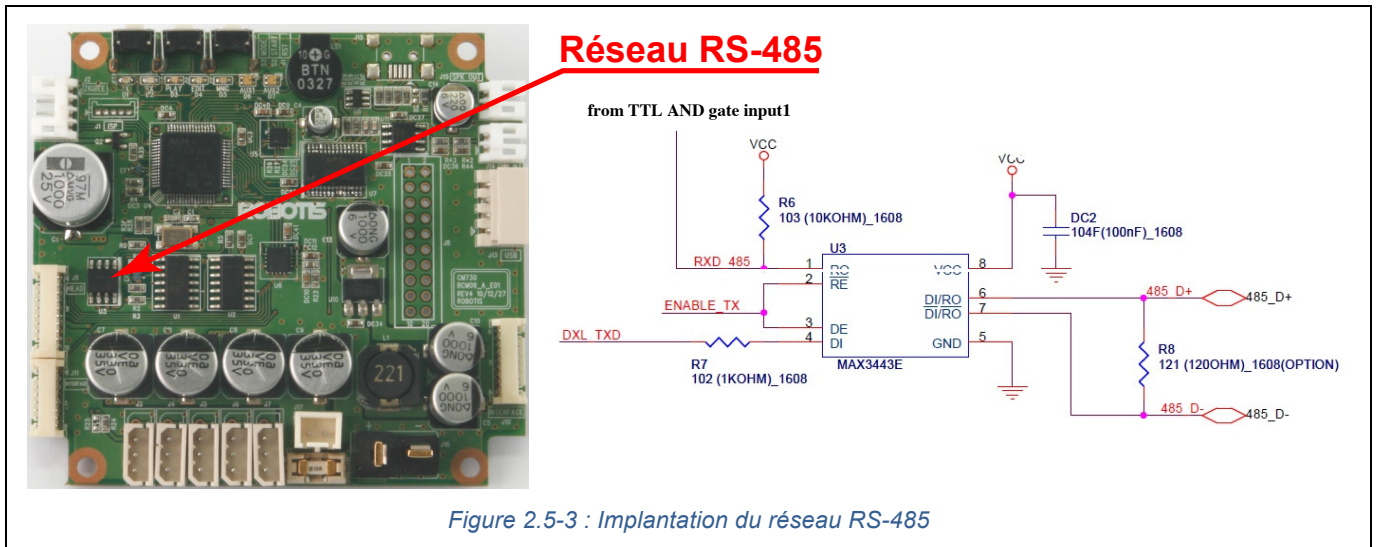


#### 2.5.1.2. ZigBee et Bluetooth

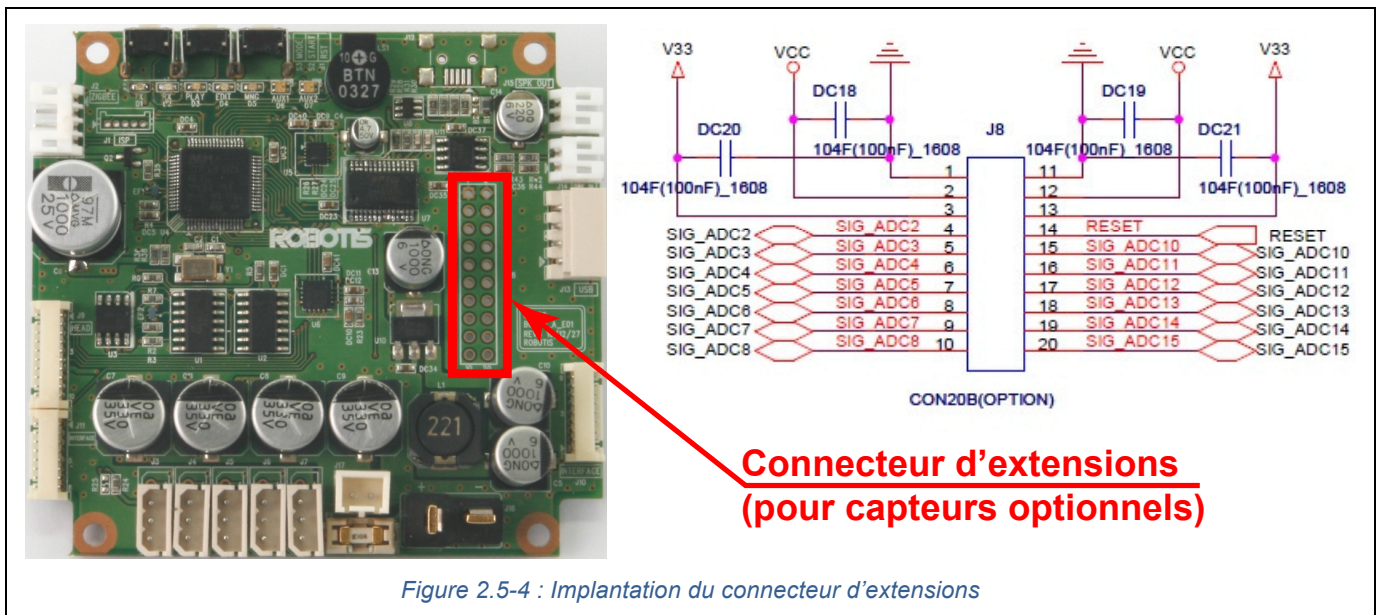




**2.5.1.3. Réseau RS-485**



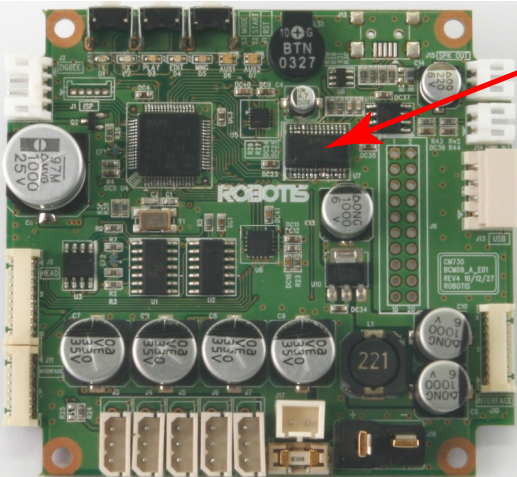
**2.5.1.4. Connecteur d'extensions pour capteurs optionnels**








### 2.5.1.5. Convertisseur USB / Série



**Convertisseur USB vers série**

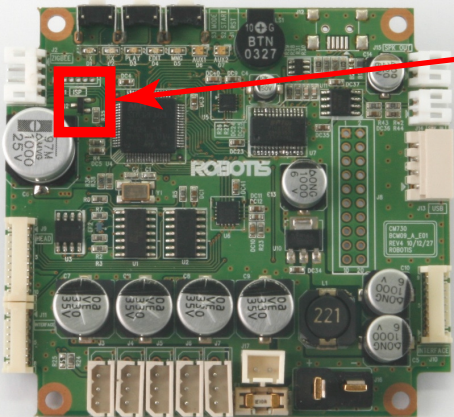


**USB2Serial**

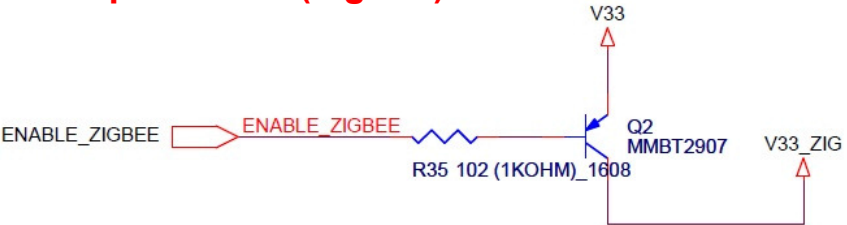
*Figure 2.5-5 : Implantation du convertisseur USB vers série*

### 2.5.1.6. Unité de contrôle sans fil de la puissance (ZigBee)

**ZigBee** est un protocole de haut niveau permettant la communication de petites radios, à consommation réduite, basée sur la norme [IEEE 802.15.4](#) pour les réseaux à dimension personnelle.



**Unité de contrôle sans fil de la puissance (ZigBee)**



*Figure 2.5-6*

*Figure 2.5-7 : Implantation de l'unité de contrôle sans fil de la puissance*





**2.5.1.7. Volt ADC**

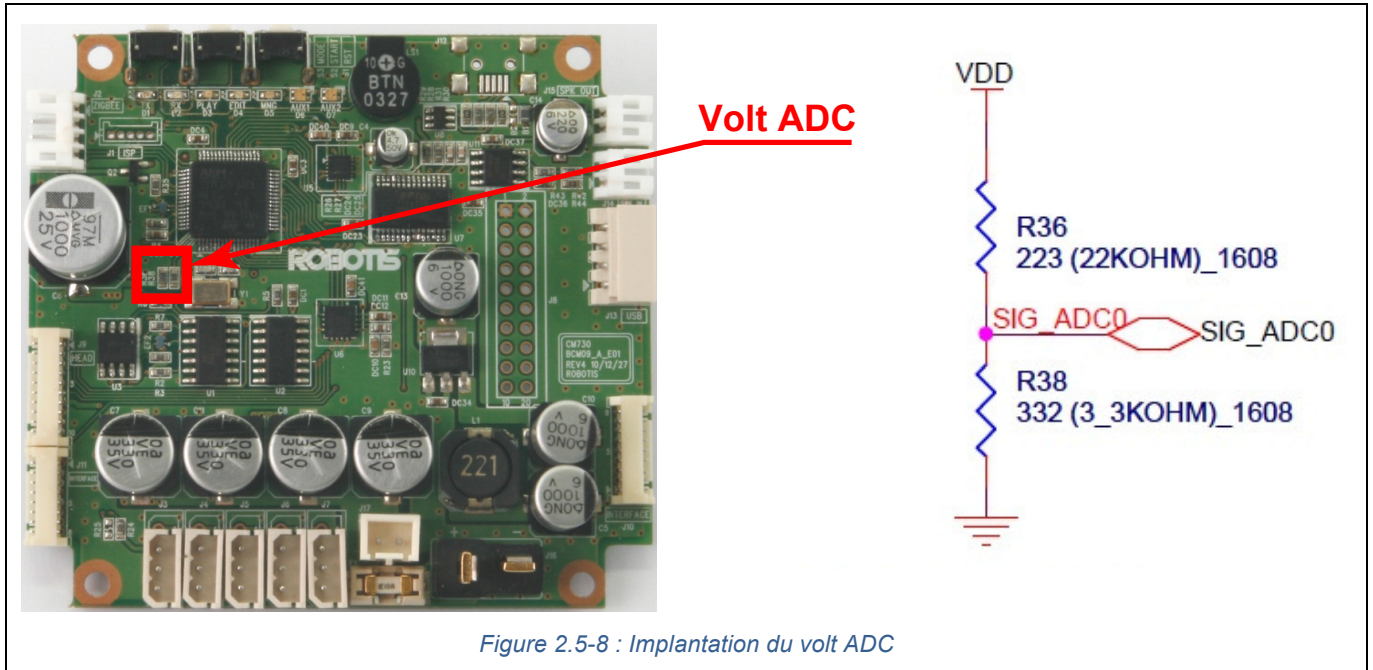


Figure 2.5-8 : Implantation du volt ADC

**2.5.1.8. Dynamixel Power Control**

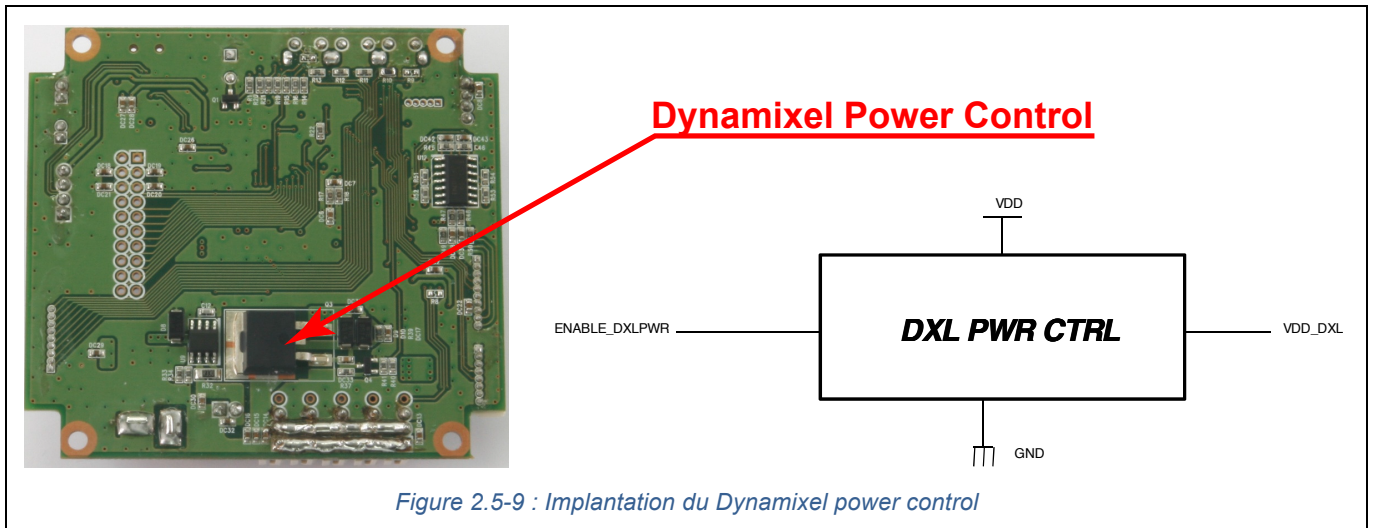


Figure 2.5-9 : Implantation du Dynamixel power control



## 2.5.2. Les programmes

### 2.5.2.1. Différents points de vue

#### ■ Point de vue « Supervision du robot »

Ce point de vue modélise ce qu'il est possible de faire avec l'ensemble de la suite logicielle qui permet :

- ✓ de concevoir des déplacements ;
- ✓ de contrôler les actionneurs ;
- ✓ de contrôler la vision ;
- ✓ etc.

Dans ce point de vue, **le robot est un acteur.**

Une décomposition de la fonction « Superviser un robot DARwin-OP » est représentée dans la figure ci-dessous.

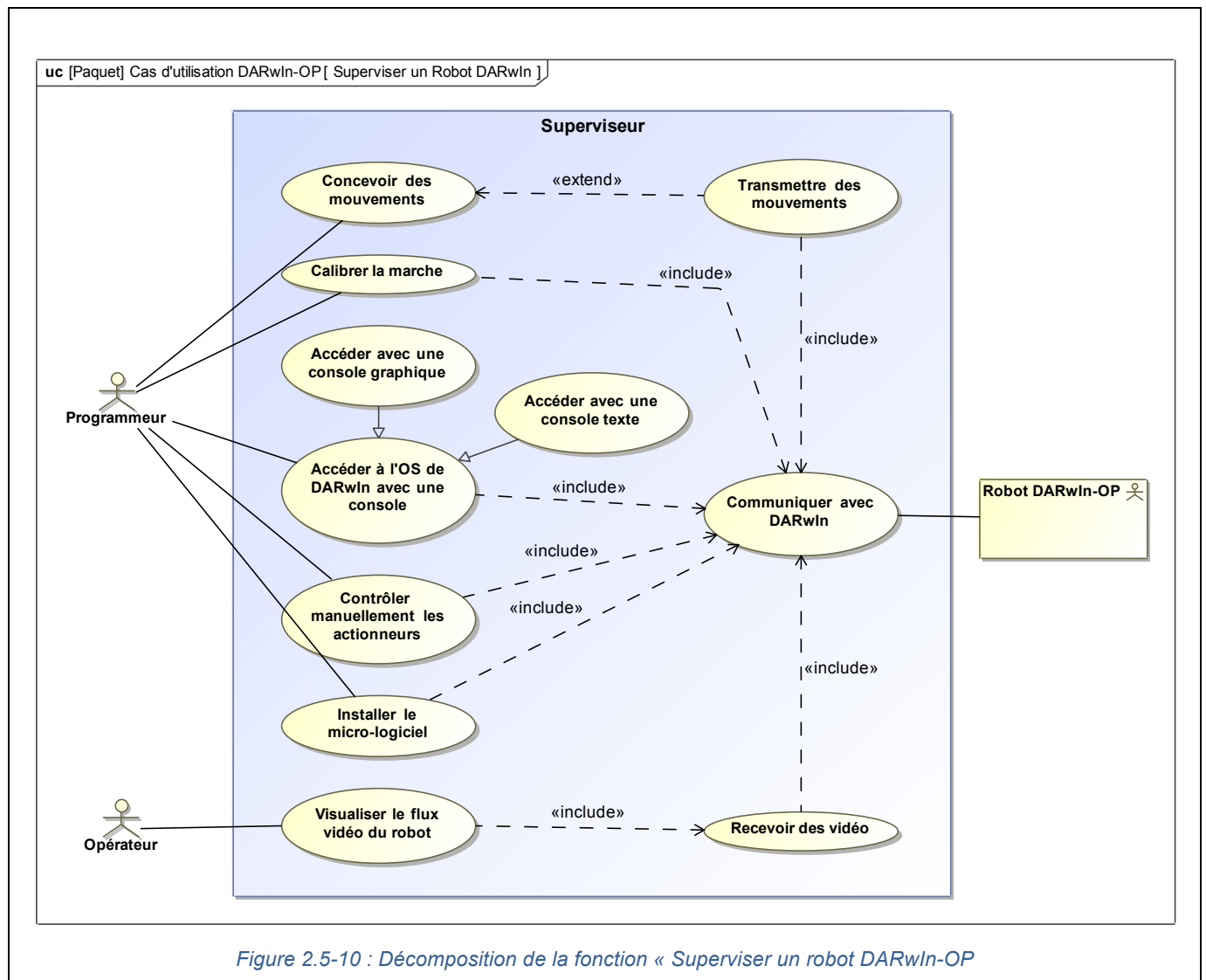


Figure 2.5-10 : Décomposition de la fonction « Superviser un robot DARwin-OP »



■ **Point de vue du « Robot »**

Dans ce point de vue du robot, la **supervision est un acteur**.

Le diagramme correspondant a été donné à la Figure 1.2-4 et il est rappelé ci-après (Figure 2.5-11).

Dans ce cas, on "rentre" à l'intérieur du robot pour mettre en évidence les cas d'utilisation internes. Cela permet ainsi d'isoler le cas d'utilisation « Simuler la marche » pour lequel le module « Walking » est détaillé dans le paragraphe suivant.

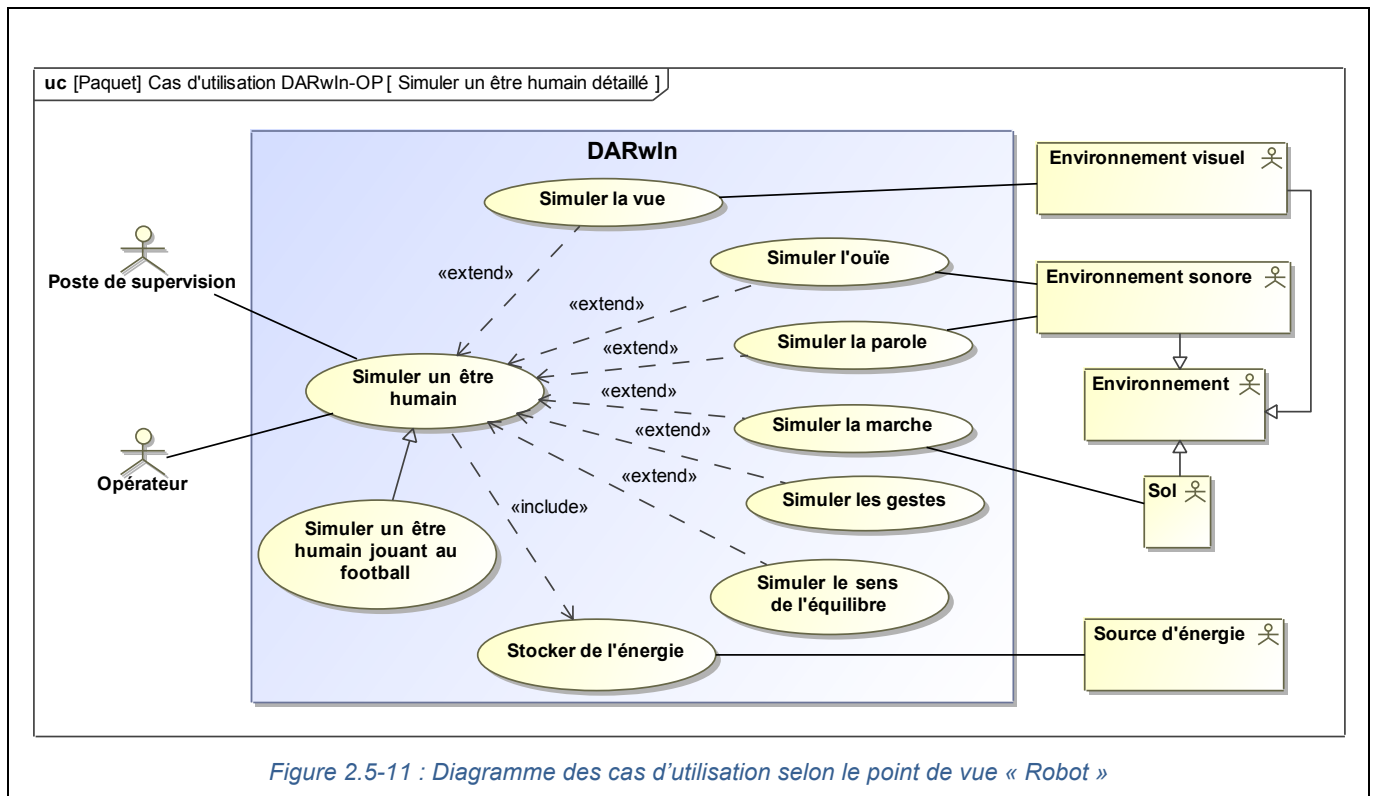


Figure 2.5-11 : Diagramme des cas d'utilisation selon le point de vue « Robot »

**2.5.2.2. Module « Walking », ou module de « Marche »**

■ **Diagramme de classe associé au module « Walking »**

Le diagramme (Figure 2.5-12) montre de manière simplifiée les modules logiciels participant au déplacement du robot :

- ✓ La classe principale (Main) chargée d'instancier les classes et d'indiquer la direction au robot ;
- ✓ La classe MotionManager chargée de la gestion des mouvements en règle générale ;
- ✓ La classe Walking chargée principalement de la marche (via la classe MotionModule et le patron de conception « stratégie ») ;
- ✓ Les classes assistantes :
  - CM370, MX28 : accès aux moteurs ;
  - LinuxMotionTimer : cadencement ;
  - JointData, Point2D, Point3D, Vector3D, Matrix3D, : nécessaires aux différentes opérations de calcul.



package DARW\_Vue\_logique [ Diagramme de classe - Module "Walking" ]

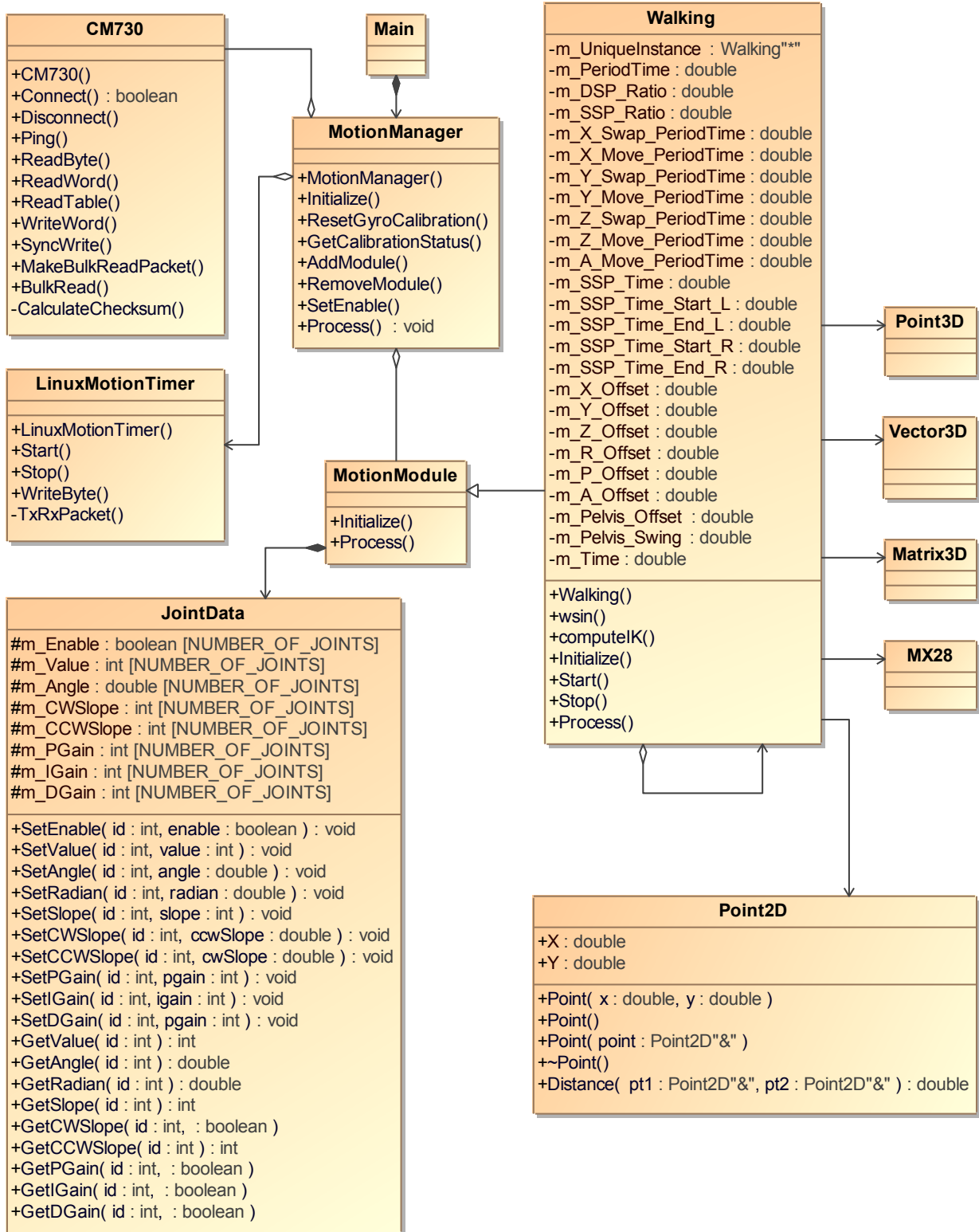


Figure 2.5-12 : Diagramme de classe associé au module « Walking »



■ Diagramme d'activité du module de marche

Le diagramme (Figure 2.5-13) montre les différentes activités (réalisées par différents modules logiciels et matériels) qui s'enchaînent pour réaliser un déplacement du robot.

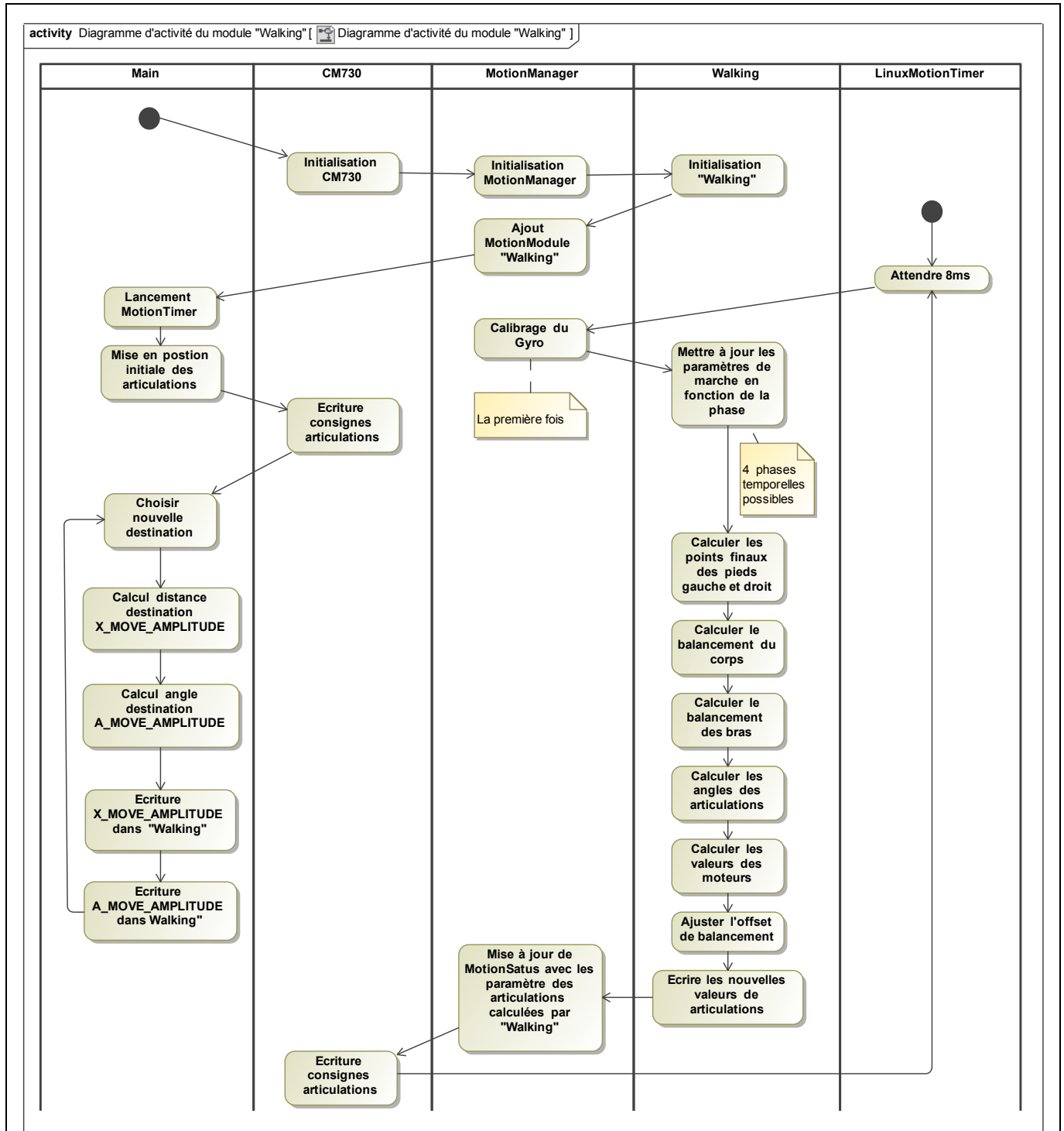


Figure 2.5-13 : Diagramme d'activité du module de marche





- ✓ Après une phase d'initialisation des différents objets (instances des classes CM-730, MotionManager et Walking), le programme principal (main) lance un timer (MotionTimer) qui va exécuter le processus de marche toutes les 8 ms.
- ✓ Le programme principal doit donner une distance à parcourir et un angle à prendre par rapport à son axe X (c'est le cas dans le programme de suivi de balle par exemple).
- ✓ L'opération « Process() » du module « Walking » est alors exécutée toutes les 8 ms sur un cycle de marche allant jusqu'à 600 ms et effectue les actions suivantes à chaque exécution :
  - calcul des coordonnées finales des pieds gauche et droit en fonction des phases de la marche (*voir plus loin*) ;
  - calcul du balancement du corps afin de respecter les critères de stabilité du robot ;
  - calcul du balancement des bras du robot ;
  - calcul des valeurs des angles des articulations (hanches, genoux, chevilles) ;
  - calcul des valeurs correspondantes pour les moteurs des articulations ;
  - écritures des consignes pour chaque articulation.

### ■ Les paramètres de la marche

Un cycle de marche se déroule comme le montre la Figure 2.5-14.

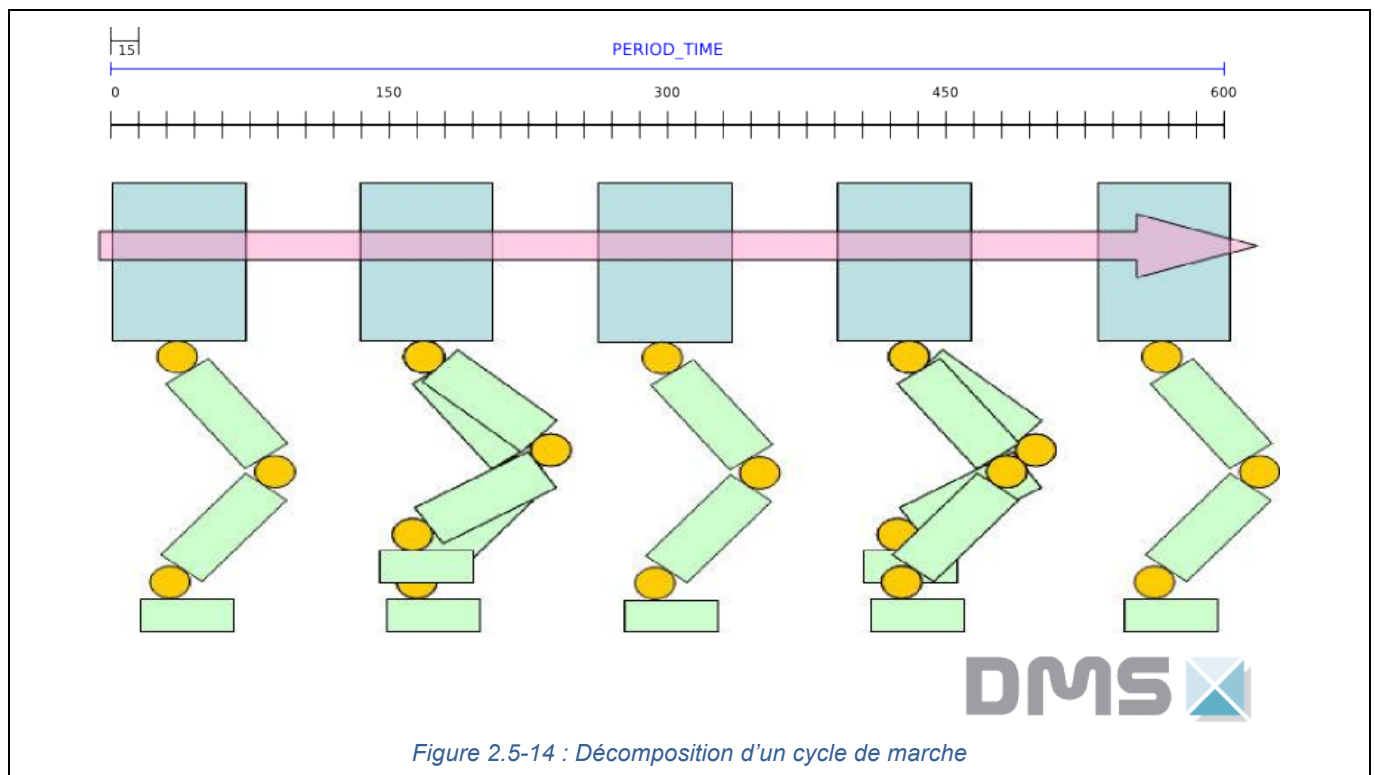
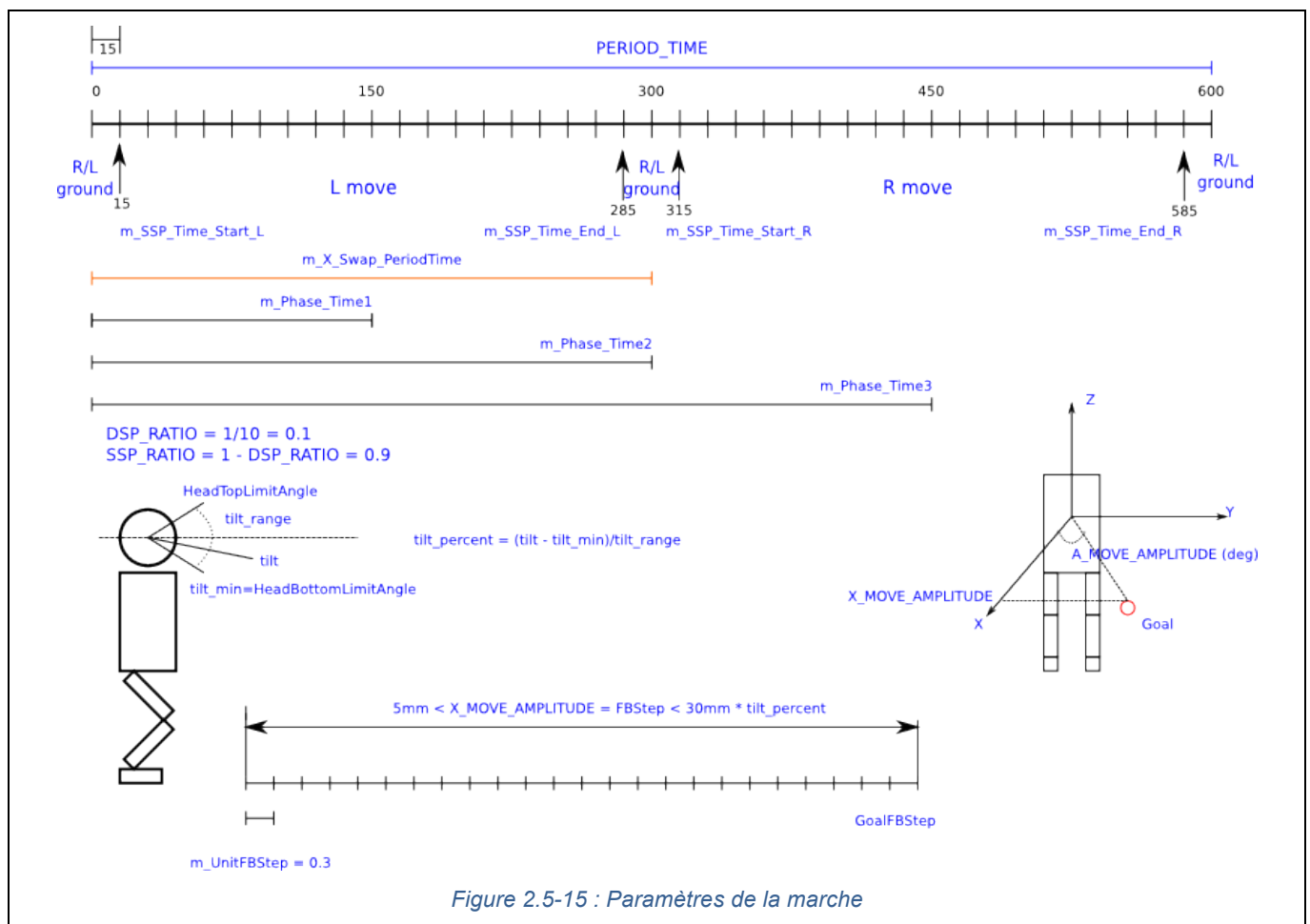


Figure 2.5-14 : Décomposition d'un cycle de marche

Le paramètre `PERIOD_TIME` que l'on retrouve dans le module « Walking.cpp » représente le temps pour faire deux pas complets (pied droit/pied gauche). `PERIOD_TIME` vaut 600 ms par défaut.

Les autres paramètres importants de la marche sont décrits dans la Figure 2.5-15.



Pour se déplacer jusqu'à sa cible, on doit fournir au minimum deux informations au module « Walking » :

- ✓ X\_MOVE\_AMPLITUDE qui représente le déplacement sur son axe X (par pas de 30 mm maximum) ;
- ✓ A\_MOVE\_AMPLITUDE qui représente l'angle que fait la cible par rapport à son axe X (en degré).

Dans le cas du programme de suivi de balle, ces informations sont calculées par le module de Vision grâce aux images prises par la caméra.

Les trajectoires des pieds gauche et droit sont ensuite calculées sur la période « PERIOD\_TIME » en fonction de l'avancement du temps :

- ✓  $m\_time < m\_SSP\_Time\_Start\_L$  : les deux pieds sont au sol ;
- ✓  $m\_SSP\_Time\_Start\_L < m\_time < m\_SSP\_Time\_End\_L$  : le pied gauche effectue sa trajectoire de marche (pied droit au sol) ;
- ✓  $m\_SSP\_Time\_End\_L < m\_time < m\_SSP\_Time\_Start\_R$  : les deux pieds sont au sol ;
- ✓  $m\_SSP\_Time\_Start\_R < m\_time < m\_SSP\_Time\_End\_R$  : le pied droit effectue sa trajectoire de marche (pied gauche au sol) ;
- ✓  $m\_SSP\_Time\_End\_R < m\_time < PERIOD\_TIME$  : les deux pieds sont au sol.



Le cycle recommence.

Détaillons l'opération Process() du module Walking.cpp exécutée toutes les 8 ms :

Une des premières opérations appelée régulièrement dans Process() est le calcul des jalons temporels :

```
void Walking::update_param_time()
{
    // temps requis pour réaliser deux pas entier (gauche/droit) : def 600 ms
    m_PeriodTime = PERIOD_TIME;
    // % du temps pendant lequel les deux pieds sont sur le sol : 0.10
    m_DSP_Ratio = DSP_RATIO;
    // % du temps pendant lequel 1 pied est sur le sol : 0.90
    m_SSP_Ratio = 1 - DSP_RATIO;

    m_X_Swap_PeriodTime = m_PeriodTime / 2;           // m_X_Swap_PeriodTime = 600/2 = 300
    m_X_Move_PeriodTime = m_PeriodTime * m_SSP_Ratio; // m_X_Move_PeriodTime = 600 * 0.9 = 540
    m_Y_Swap_PeriodTime = m_PeriodTime;              // m_Y_Swap_PeriodTime = 600
    m_Y_Move_PeriodTime = m_PeriodTime * m_SSP_Ratio; // m_Y_Move_PeriodTime = 540
    m_Z_Swap_PeriodTime = m_PeriodTime / 2;         // m_Z_Swap_PeriodTime = 300
    m_Z_Move_PeriodTime = m_PeriodTime * m_SSP_Ratio / 2; // m_Z_Move_PeriodTime = 600 * 0.9 / 2 = 270
    m_A_Move_PeriodTime = m_PeriodTime * m_SSP_Ratio; // m_A_Move_PeriodTime = 600 * 0.9 = 540
    m_SSP_Time = m_PeriodTime * m_SSP_Ratio;        // m_SSP_Time = 540
    m_SSP_Time_Start_L = (1 - m_SSP_Ratio) * m_PeriodTime / 4; //
    m_SSP_Time_Start_L = (1 - 0.9) * 600/4 = 15
    m_SSP_Time_End_L = (1 + m_SSP_Ratio) * m_PeriodTime / 4; // m_SSP_Time_End_L = (1 + 0.9) * 600/4 = 285
    m_SSP_Time_Start_R = (3 - m_SSP_Ratio) * m_PeriodTime / 4; //
    m_SSP_Time_Start_R = (3 - 0.9) * 600/4 = 315
    m_SSP_Time_End_R = (3 + m_SSP_Ratio) * m_PeriodTime / 4; // m_SSP_Time_End_R = (3 + 0.9) * 600/4 = 585

    m_Phase_Time1 = (m_SSP_Time_End_L + m_SSP_Time_Start_L) / 2; // m_Phase_Time1 = (285+15) / 2 = 150
    m_Phase_Time2 = (m_SSP_Time_Start_R + m_SSP_Time_End_L) / 2; // m_Phase_Time2 = (315+285) / 2 = 300
    m_Phase_Time3 = (m_SSP_Time_End_R + m_SSP_Time_Start_R) / 2; // m_Phase_Time3 = (585+315) / 2 = 450

    m_Pelvis_Offset = PELVIS_OFFSET*MX28::RATIO_ANGLE2VALUE; // m_Pelvis_Offset = 3 * 11.378 = 34; // 4096 / 360;
    m_Pelvis_Swing = m_Pelvis_Offset * 0.35; // m_Pelvis_Swing = 34 * 0.35 = 12
    m_Arm_Swing_Gain = ARM_SWING_GAIN;
}
}
```



La deuxième opération appelée régulièrement dans Process() est la mise à jour des paramètres généraux pour le déplacement qui serviront ensuite au calcul des trajectoires des pieds et du corps du robot :

```
void Walking::update_param_move()
{
    // Forward/Back
    m_X_Move_Amplitude = X_MOVE_AMPLITUDE;           // m_X_Move_Amplitude = 30 mm max
    m_X_Swap_Amplitude = X_MOVE_AMPLITUDE * STEP_FB_RATIO; // m_X_Swap_Amplitude = 30 * 0.28
    = 8.4

    // Right/Left
    m_Y_Move_Amplitude = Y_MOVE_AMPLITUDE / 2;       // = 0 dans le cas ballfollower
    if(m_Y_Move_Amplitude > 0)
        m_Y_Move_Amplitude_Shift = m_Y_Move_Amplitude;
    else
        m_Y_Move_Amplitude_Shift = -m_Y_Move_Amplitude; // m_Y_Move_Amplitude_Shift = 0
    m_Y_Swap_Amplitude = Y_SWAP_AMPLITUDE + m_Y_Move_Amplitude_Shift * 0.04; //
    m_Y_Swap_Amplitude = 20 + 0 * 0.04 = 20
    m_Z_Move_Amplitude = Z_MOVE_AMPLITUDE / 2;       // m_Z_Move_Amplitude = 40 / 2 = 20
    m_Z_Move_Amplitude_Shift = m_Z_Move_Amplitude / 2; // m_Z_Move_Amplitude_Shift = 20 /
    2 = 10
    m_Z_Swap_Amplitude = Z_SWAP_AMPLITUDE;           // m_Z_Swap_Amplitude = 5
    m_Z_Swap_Amplitude_Shift = m_Z_Swap_Amplitude;   // m_Z_Swap_Amplitude_Shift = 5

    // Direction
    if(A_MOVE_AIM_ON == false)
    {
        // ex A_MOVE_AMPLITUDE = 0
        m_A_Move_Amplitude = A_MOVE_AMPLITUDE * PI / 180.0 / 2; // m_A_Move_Amplitude = 0
        if(m_A_Move_Amplitude > 0)
            m_A_Move_Amplitude_Shift = m_A_Move_Amplitude;
        else
            m_A_Move_Amplitude_Shift = -m_A_Move_Amplitude; // m_A_Move_Amplitude_Shift = 0
    }
    else
    {
        m_A_Move_Amplitude = -A_MOVE_AMPLITUDE * PI / 180.0 / 2;
        if(m_A_Move_Amplitude > 0)
            m_A_Move_Amplitude_Shift = -m_A_Move_Amplitude;
        else
            m_A_Move_Amplitude_Shift = m_A_Move_Amplitude;
    }
}
```

Une fois ces calculs effectués, Process() détermine dans quelle phase de la marche il se trouve et calcule en conséquence :

- ✓ les coordonnées de destination finale x\_swap, y\_swap et z\_swap du robot pour cette période ;
- ✓ les coordonnées de destination finale x\_move\_l, y\_move\_l, z\_move\_l et c\_move\_l pour le pied gauche ;
- ✓ les coordonnées de destination finale x\_move\_r, y\_move\_r, z\_move\_r et c\_move\_r pour le pied droit.

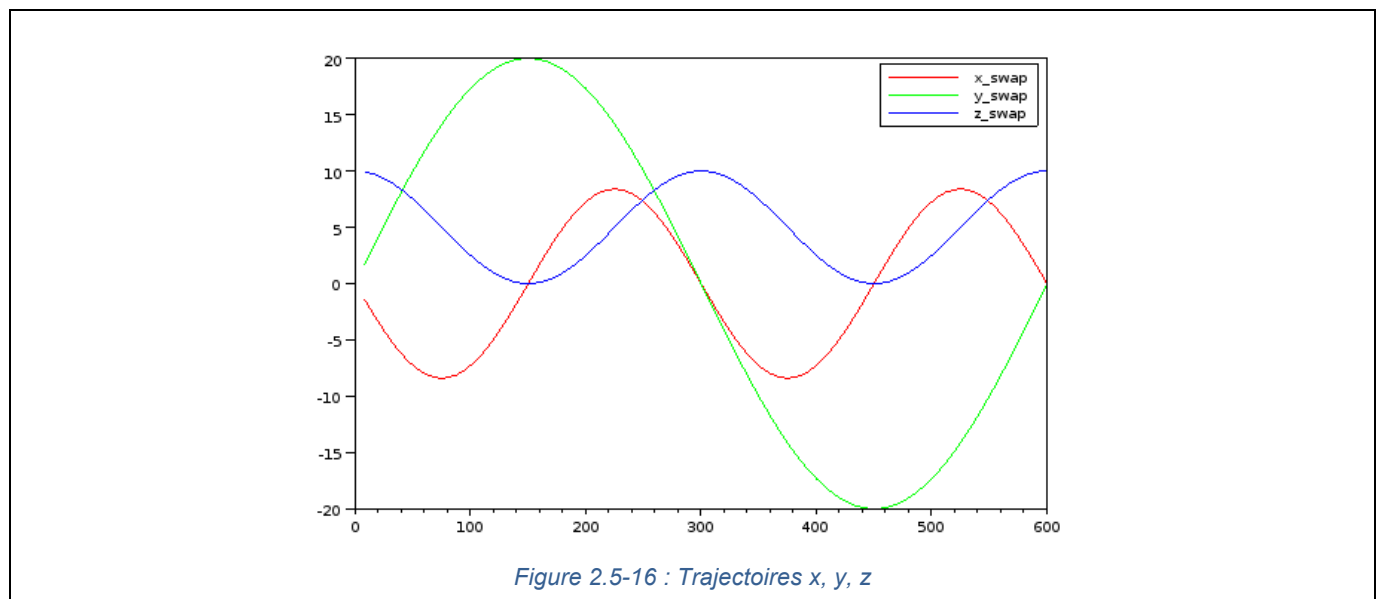


```
// Compute endpoints
x_swap = wsin(m_Time, m_X_Swap_PeriodTime, m_X_Swap_Phase_Shift, m_X_Swap_Amplitude,
m_X_Swap_Amplitude_Shift);
y_swap = wsin(m_Time, m_Y_Swap_PeriodTime, m_Y_Swap_Phase_Shift, m_Y_Swap_Amplitude,
m_Y_Swap_Amplitude_Shift);
z_swap = wsin(m_Time, m_Z_Swap_PeriodTime, m_Z_Swap_Phase_Shift, m_Z_Swap_Amplitude,
m_Z_Swap_Amplitude_Shift);
a_swap = 0;
b_swap = 0;
c_swap = 0;

if(m_Time <= m_SSP_Time_Start_L)
{
x_move_l = wsin(m_SSP_Time_Start_L, m_X_Move_PeriodTime, m_X_Move_Phase_Shift + 2 * PI /
m_X_Move_PeriodTime * m_SSP_Time_Start_L, m_X_Move_Amplitude, m_X_Move_Amplitude_Shift);
y_move_l = wsin(m_SSP_Time_Start_L, m_Y_Move_PeriodTime, m_Y_Move_Phase_Shift + 2 * PI /
m_Y_Move_PeriodTime * m_SSP_Time_Start_L, m_Y_Move_Amplitude, m_Y_Move_Amplitude_Shift);
z_move_l = wsin(m_SSP_Time_Start_L, m_Z_Move_PeriodTime, m_Z_Move_Phase_Shift + 2 * PI /
m_Z_Move_PeriodTime * m_SSP_Time_Start_L, m_Z_Move_Amplitude, m_Z_Move_Amplitude_Shift);
c_move_l = wsin(m_SSP_Time_Start_L, m_A_Move_PeriodTime, m_A_Move_Phase_Shift + 2 * PI /
m_A_Move_PeriodTime * m_SSP_Time_Start_L, m_A_Move_Amplitude, m_A_Move_Amplitude_Shift);
x_move_r = wsin(m_SSP_Time_Start_L, m_X_Move_PeriodTime, m_X_Move_Phase_Shift + 2 * PI /
m_X_Move_PeriodTime * m_SSP_Time_Start_L, -m_X_Move_Amplitude, -m_X_Move_Amplitude_Shift);
y_move_r = wsin(m_SSP_Time_Start_L, m_Y_Move_PeriodTime, m_Y_Move_Phase_Shift + 2 * PI /
m_Y_Move_PeriodTime * m_SSP_Time_Start_L, -m_Y_Move_Amplitude, -m_Y_Move_Amplitude_Shift);
z_move_r = wsin(m_SSP_Time_Start_R, m_Z_Move_PeriodTime, m_Z_Move_Phase_Shift + 2 * PI /
m_Z_Move_PeriodTime * m_SSP_Time_Start_R, m_Z_Move_Amplitude, m_Z_Move_Amplitude_Shift);
c_move_r = wsin(m_SSP_Time_Start_L, m_A_Move_PeriodTime, m_A_Move_Phase_Shift + 2 * PI /
m_A_Move_PeriodTime * m_SSP_Time_Start_L, -m_A_Move_Amplitude, -m_A_Move_Amplitude_Shift);
pelvis_offset_l = 0;
pelvis_offset_r = 0;
}
...
```

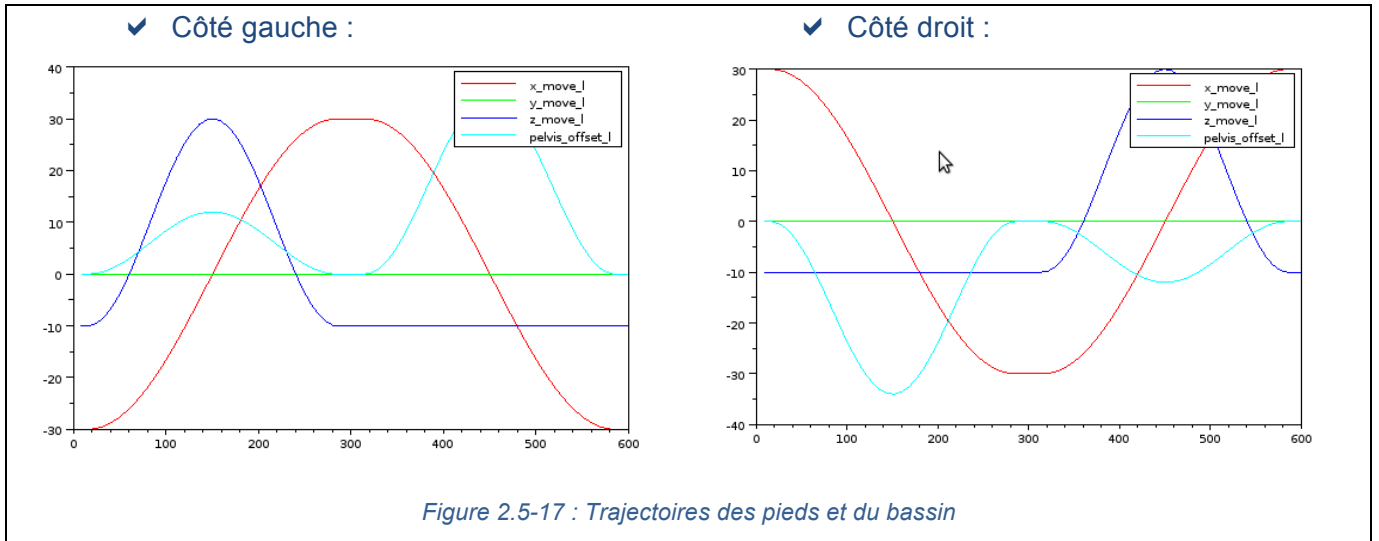
Les courbes suivantes montrent l'évolution des coordonnées pour les paramètres de marche fixés :

✓ Trajectoires x, y, z générales :



✓ Trajectoires des pieds et du bassin



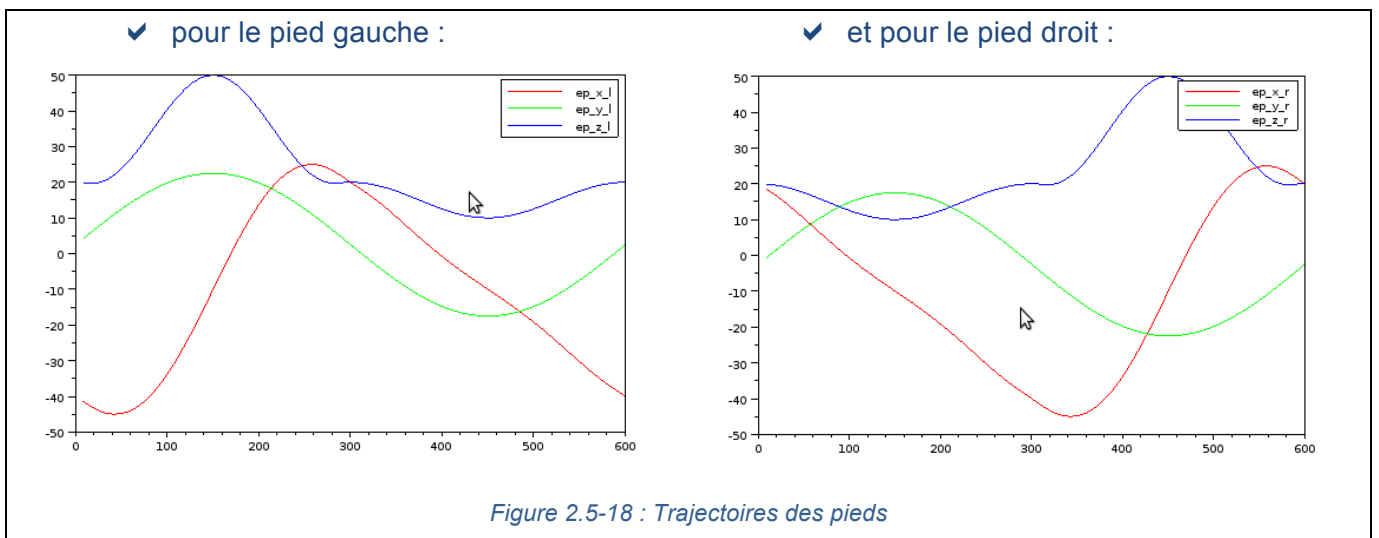


Les trajectoires sont ensuite combinées pour calculer les trajectoires des pieds gauche et droit :

```

ep[0] = x_swap + x_move_r + m_X_Offset;           // x_r
ep[1] = y_swap + y_move_r - m_Y_Offset / 2;      // y_r
ep[2] = z_swap + z_move_r + m_Z_Offset;          // z_r
ep[3] = a_swap + a_move_r - m_R_Offset / 2;      // roll_r (axe x)
ep[4] = b_swap + b_move_r + m_P_Offset;          // pitch_r (axe y)
ep[5] = c_swap + c_move_r - m_A_Offset / 2;      // yaw_r (axe z)
ep[6] = x_swap + x_move_l + m_X_Offset;           // x_l
ep[7] = y_swap + y_move_l + m_Y_Offset / 2;      // y_l
ep[8] = z_swap + z_move_l + m_Z_Offset;          // z_l
ep[9] = a_swap + a_move_l + m_R_Offset / 2;      // roll_l (axe x)
ep[10] = b_swap + b_move_l + m_P_Offset;          // pitch_r (axe y)
ep[11] = c_swap + c_move_l + m_A_Offset / 2;      // yaw_r (axe z)
    
```

Cela donne :



Les trajectoires des bras et du corps sont ensuite calculées :



```
// Compute body swing
if(m_Time <= m_SSP_Time_End_L)
{
    m_Body_Swing_Y = -ep[7];
    m_Body_Swing_Z = ep[8];
}
else
{
    m_Body_Swing_Y = -ep[1];
    m_Body_Swing_Z = ep[2];
}
m_Body_Swing_Z -= Kinematics::LEG_LENGTH;

// Compute arm swing
if(m_X_Move_Amplitude == 0)
{
    angle[12] = 0; // Right
    angle[13] = 0; // Left
}
else
{
    angle[12] = wsine(m_Time, m_PeriodTime, PI * 1.5, -m_X_Move_Amplitude * m_Arm_Swing_Gain, 0);
    angle[13] = wsine(m_Time, m_PeriodTime, PI * 1.5, m_X_Move_Amplitude * m_Arm_Swing_Gain, 0);
}
```

Les angles de chaque articulation sont ensuite déduits des coordonnées finales des pieds gauche et droit :

```
// Compute angles
if((computeIK(&angle[0], ep[0], ep[1], ep[2], ep[3], ep[4], ep[5]) == 1)
    && (computeIK(&angle[6], ep[6], ep[7], ep[8], ep[9], ep[10], ep[11]) == 1))
{
    for(int i=0; i<12; i++)
        angle[i] *= 180.0 / PI;
}
else
{
    return; // Do not use angle;
}
```

Les valeurs des moteurs sont déduites des calculs précédents :

```
// Compute motor value
for(int i=0; i<14; i++)
{
    offset = (double)dir[i] * angle[i] * MX28::RATIO_ANGLE2VALUE;
    if(i == 1) // R_HIP_ROLL
        offset += (double)dir[i] * pelvis_offset_r;
    else if(i == 7) // L_HIP_ROLL
        offset += (double)dir[i] * pelvis_offset_l;
    else if(i == 2 || i == 8) // R_HIP_PITCH or L_HIP_PITCH
        offset -= (double)dir[i] * HIP_PITCH_OFFSET * MX28::RATIO_ANGLE2VALUE;
    outValue[i] = MX28::Angle2Value(initAngle[i]) + (int)offset;
}
```



En fonction de la balance et de l'erreur du gyroscope, les valeurs sont placées dans les articulations afin d'être transmises aux moteurs :

```
// adjust balance offset
if(BALANCE_ENABLE == true)
{
    double rlGyroErr = MotionStatus::RL_GYRO;
    double fbGyroErr = MotionStatus::FB_GYRO;
    outValue[1] += (int)(dir[1] * rlGyroErr * BALANCE_HIP_ROLL_GAIN*4); // R_HIP_ROLL
    outValue[7] += (int)(dir[7] * rlGyroErr * BALANCE_HIP_ROLL_GAIN*4); // L_HIP_ROLL
    outValue[3] -= (int)(dir[3] * fbGyroErr * BALANCE_KNEE_GAIN*4); // R_KNEE
    outValue[9] -= (int)(dir[9] * fbGyroErr * BALANCE_KNEE_GAIN*4); // L_KNEE
    outValue[4] -= (int)(dir[4] * fbGyroErr * BALANCE_ANKLE_PITCH_GAIN*4); // R_ANKLE_PITCH
    outValue[10] -= (int)(dir[10] * fbGyroErr * BALANCE_ANKLE_PITCH_GAIN*4); // L_ANKLE_PITCH
    outValue[5] -= (int)(dir[5] * rlGyroErr * BALANCE_ANKLE_ROLL_GAIN*4); // R_ANKLE_ROLL
    outValue[11] -= (int)(dir[11] * rlGyroErr * BALANCE_ANKLE_ROLL_GAIN*4); // L_ANKLE_ROLL
}

m_Joint.SetValue(JointData::ID_R_HIP_YAW, outValue[0]);
m_Joint.SetValue(JointData::ID_R_HIP_ROLL, outValue[1]);
m_Joint.SetValue(JointData::ID_R_HIP_PITCH, outValue[2]);
m_Joint.SetValue(JointData::ID_R_KNEE, outValue[3]);
m_Joint.SetValue(JointData::ID_R_ANKLE_PITCH, outValue[4]);
m_Joint.SetValue(JointData::ID_R_ANKLE_ROLL, outValue[5]);
m_Joint.SetValue(JointData::ID_L_HIP_YAW, outValue[6]);
m_Joint.SetValue(JointData::ID_L_HIP_ROLL, outValue[7]);
m_Joint.SetValue(JointData::ID_L_HIP_PITCH, outValue[8]);
m_Joint.SetValue(JointData::ID_L_KNEE, outValue[9]);
m_Joint.SetValue(JointData::ID_L_ANKLE_PITCH, outValue[10]);
m_Joint.SetValue(JointData::ID_L_ANKLE_ROLL, outValue[11]);
m_Joint.SetValue(JointData::ID_R_SHOULDER_PITCH, outValue[12]);
m_Joint.SetValue(JointData::ID_L_SHOULDER_PITCH, outValue[13]);
m_Joint.SetAngle(JointData::ID_HEAD_PAN, A_MOVE_AMPLITUDE);

for(int id = JointData::ID_R_HIP_YAW; id <= JointData::ID_L_ANKLE_ROLL; id++)
{
    m_Joint.SetPGain(id, P_GAIN);
    m_Joint.SetIGain(id, I_GAIN);
    m_Joint.SetDGain(id, D_GAIN);
}
```



### 2.5.2.3. Module « Marche simple »

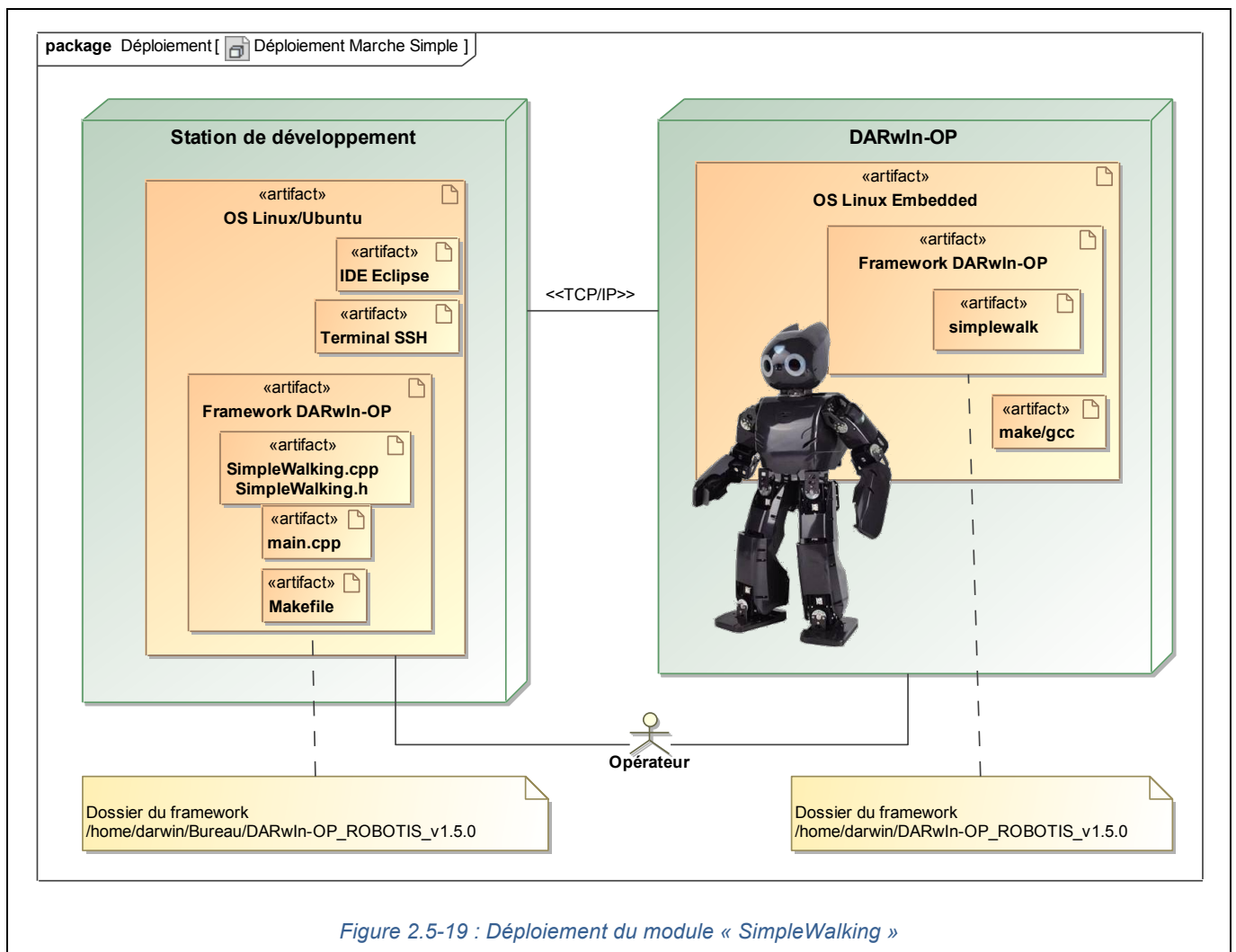
L'objectif de ce module est de permettre le pilotage des servomoteurs de DARwIn-OP afin de lui faire effectuer une marche simple en ligne droite.

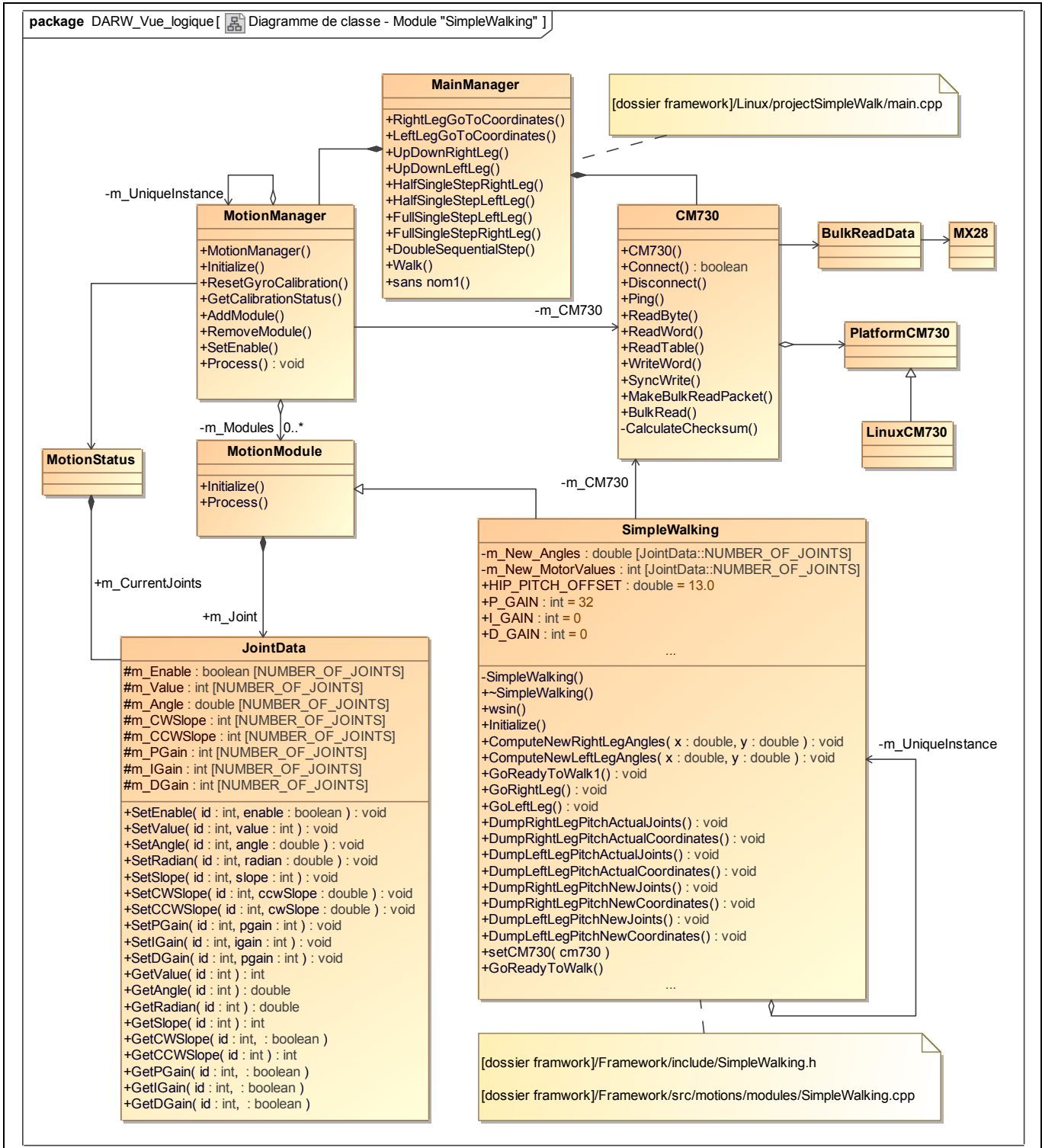
La Fig. 2.5-19 présente le diagramme de déploiement de l'application « Marche simple » en phase de développement.

La Fig. 2.5-20 présente le diagramme de classe l'application « Marche simple ».

Les principales classes sont :

- ✓ **MainManager** : c'est en fait le programme principal défini dans le fichier main.cpp. Le programme principal sera chargé d'initialiser le framework de DARwIn-OP et de lancer l'une ou l'autre des fonctions décrivant étape par étape la marche simple.
- ✓ **MotionManager** : c'est la classe chargée d'assurer les mouvements du robot.
- ✓ **SimpleWalking** : c'est la classe chargée de calculer les positions de servomoteurs en fonction de la phase de la marche.
- ✓ **CM730** (et les autres classes en relation) : classes permettant de lire et d'écrire les consignes sur les servomoteurs.



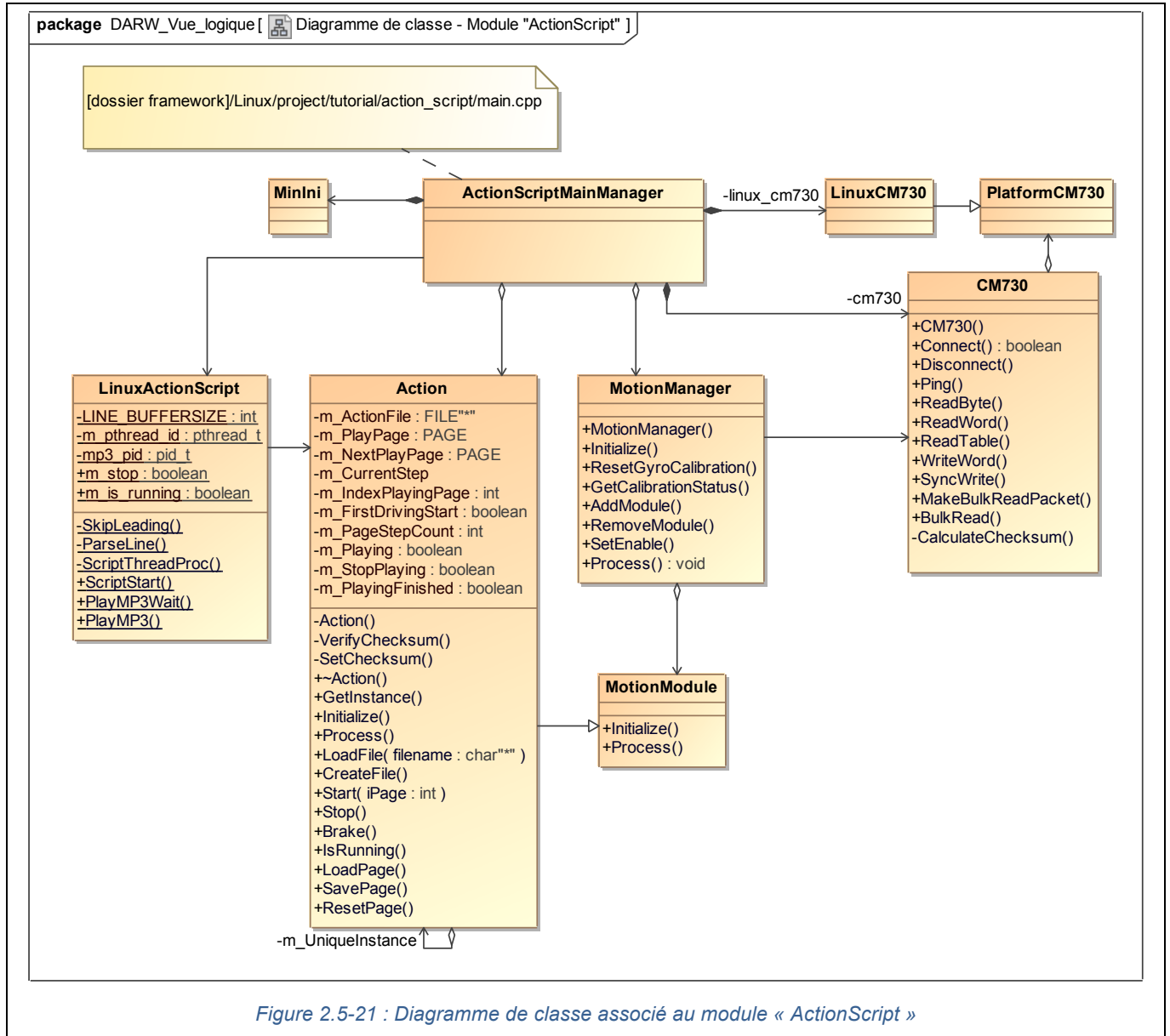






### 2.5.2.4. Module « Action Script »

L'objectif de ce module est de permettre l'exécution d'actions élémentaires par DARwIn. Ces actions sont consignées dans un simple fichier texte.





### 2.5.2.5. Module « Ball Traking », ou « Suivi de balle »

L'objectif de ce module est de réaliser une machine à état simple où le robot va se lever s'il détecte une balle dans son champ de vision et s'assoir si il ne voit plus la balle.

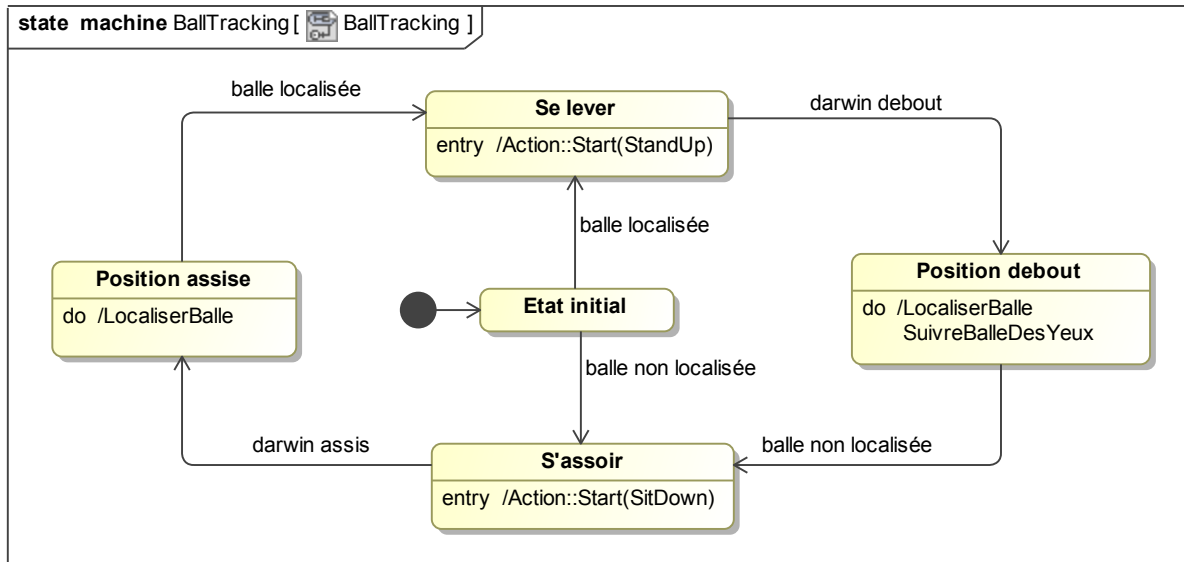


Figure 2.5-22 : Diagramme Machine à état du module « Ball Tracking »

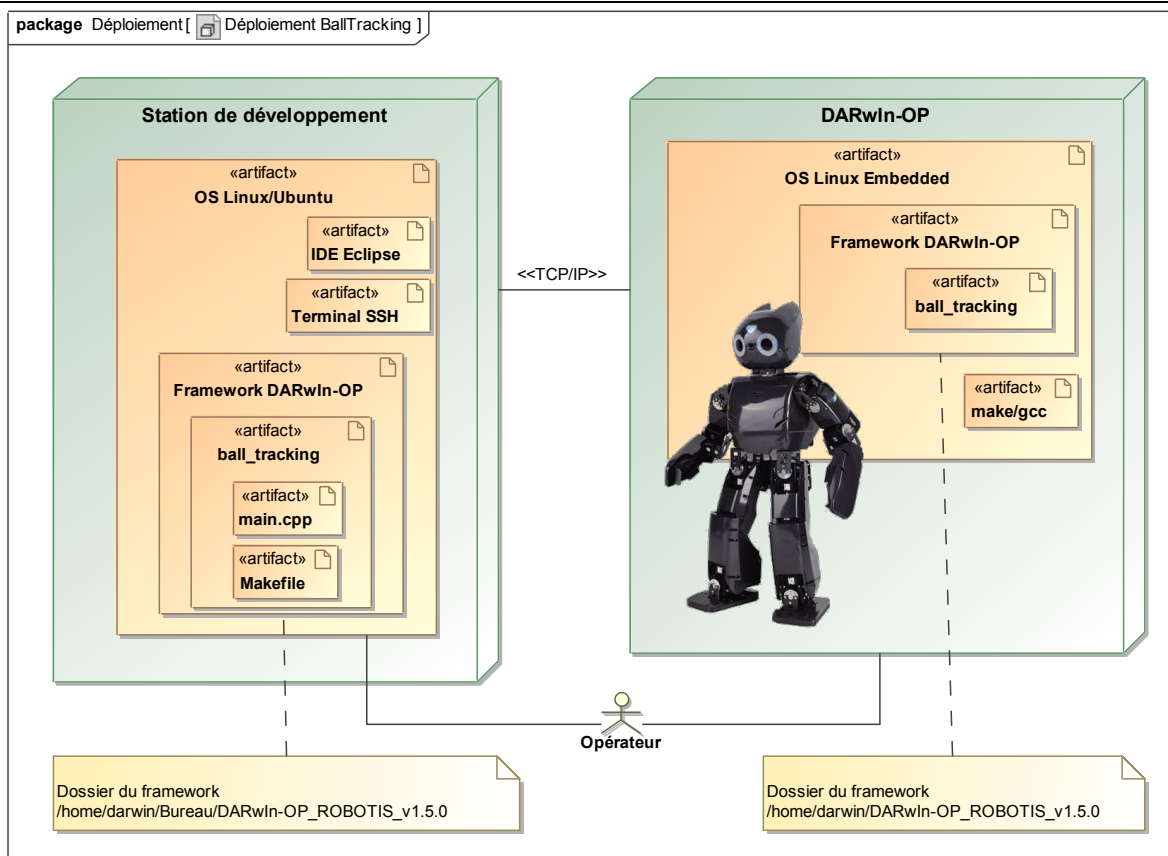


Figure 2.5-23 : Déploiement du module « Ball Tracking »



package DARW\_Vue\_logique [ Diagramme de classe - Module "BallTracking" ]

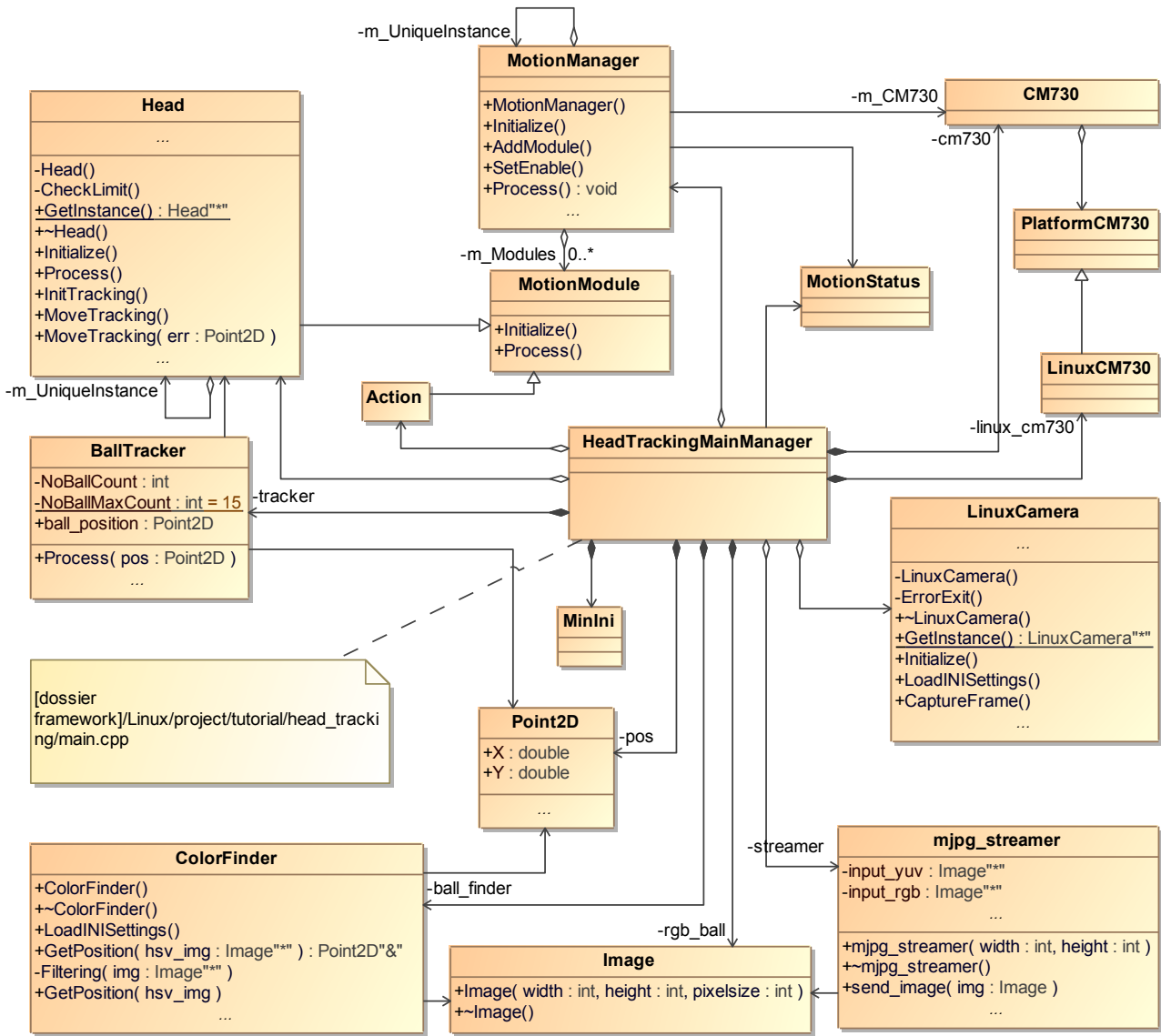


Figure 2.5-24 : Diagramme de classe associé au module « Ball Tracking »