



# Darwin-Mini

## Dossier Technique





## Table des matières

<b>Table des matières</b> .....	<b>2</b>
<b>1. Le produit réel : Darwin-Mini</b> .....	<b>3</b>
1.1. Robotique et robots.....	3
1.1.1. Qu'est-ce que la robotique ?.....	3
1.1.2. Les robots.....	3
1.1.3. Les robots humanoïdes.....	3
1.2. Analyse du besoin.....	3
1.3. Présentation de Darwin-Mini.....	4
1.3.1. Constitution.....	4
1.3.2. Mouvements et servomoteurs.....	5
1.3.3. Caractéristiques des servomoteurs.....	6
<b>2. Le produit didactique</b> .....	<b>8</b>
2.1. Présentation.....	8
2.2. Suite logicielle RobotPlus 2.0.....	8
2.2.1. Applications pour smartphone et tablette.....	8
2.2.2. R+Design et R+m.Design.....	8
2.2.2.1. Fonction.....	8
2.2.2.2. Icones.....	9
2.2.2.3. Écrans.....	9
2.2.3. R+Motion et R+m.Motion.....	10
2.2.3.1. Fonction.....	10
2.2.3.2. Icones.....	10
2.2.3.3. Quel est le lien entre un fichier de mouvement et un fichier des tâches ?.....	10
2.2.4. R+Task et R+m.Task.....	<b>Erreur ! Le signet n'est pas défini.</b>
2.2.4.1. Fonction.....	<b>Erreur ! Le signet n'est pas défini.</b>
2.2.4.2. Icones.....	<b>Erreur ! Le signet n'est pas défini.</b>
2.2.4.3. Quel est le lien entre un fichier de mouvement et un fichier des tâches ?... <b>Erreur ! Le signet n'est pas défini.</b>	
2.2.4.4. Utilisation de R+Task.....	<b>Erreur ! Le signet n'est pas défini.</b>
2.3. Customisation du robot.....	13
2.3.1. Dispositifs périphériques.....	15
2.3.1.1. Inventaire des dispositifs actuels.....	15
2.3.1.2. Connexion.....	16
2.3.1.3. Module LED (LED Module LM-10).....	16
2.3.1.4. Capteur infrarouge (IR Sensor IRSS-10).....	17
2.3.1.5. Capteur de distance (DMS Sensor DMS-80).....	18
2.3.1.6. Capteur à contact (Touch Sensor TS-10).....	19
2.3.1.7. Capteur de couleur (Color Sensor CS-10).....	19
2.3.1.8. Capteur magnétique (Magnetic Sensor MGSS-10).....	20
2.3.1.9. Capteur de température (Temperature Sensor TPS-10).....	20
2.3.1.10. Capteur gyroscopique (Gyro sensor GS-12).....	20
2.3.1.11. Détecteur d'objet (Passive InfraRed Sensor PIR-10).....	20
2.3.2. Apparence du robot.....	21



## 1. Le produit réel : Darwin-Mini

### 1.1. Robotique et robots

#### 1.1.1. Qu'est-ce que la robotique ?

La **robotique** est l'ensemble des techniques permettant la conception et la réalisation de machines automatiques ou de **robots**.

#### 1.1.2. Les robots

Un **robot** est un dispositif mécanique, électronique et informatique qui accomplit automatiquement soit des tâches qui sont généralement dangereuses, pénibles, répétitives ou impossibles pour les humains, soit des tâches plus simples mais en les réalisant mieux que ce que ferait un être humain.

On trouve de plus en plus de robots dans toutes les branches de l'industrie (construction automobile, aéronautique), dans les secteurs de pointe comme la chirurgie, le spatial et l'armée, ainsi que dans l'environnement domestique (robots ménagers).

#### 1.1.3. Les robots humanoïdes

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.

## 1.2. 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.

### ■ *Les robots humanoïdes pour quoi faire ?*

Le dessin ci-dessous donne un raccourci humoristique de quelques tâches que peuvent accomplir certains robots et les robots humanoïdes.

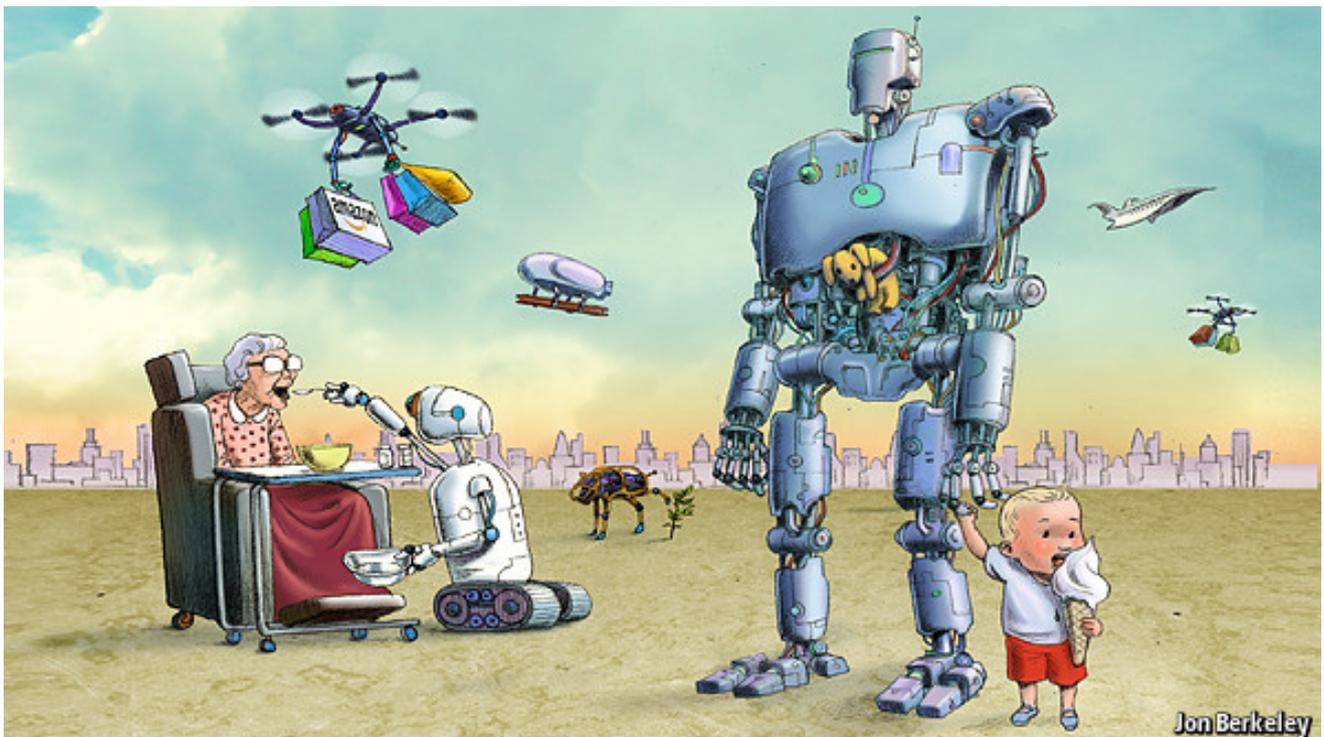


Figure 1.2-1 : Dessin tiré du magazine hebdomadaire "The Economist" du 29 mars au 4 avril 2014

## 1.3. Présentation de Darwin-Mini

### 1.3.1. Constitution

En tant que robot humanoïde, Darwin-Mini a une apparence semblable à celle d'un humain. La Figure 1.3-1 de la page suivante rappelle la terminologie des différentes parties du corps et les situe sur le robot.

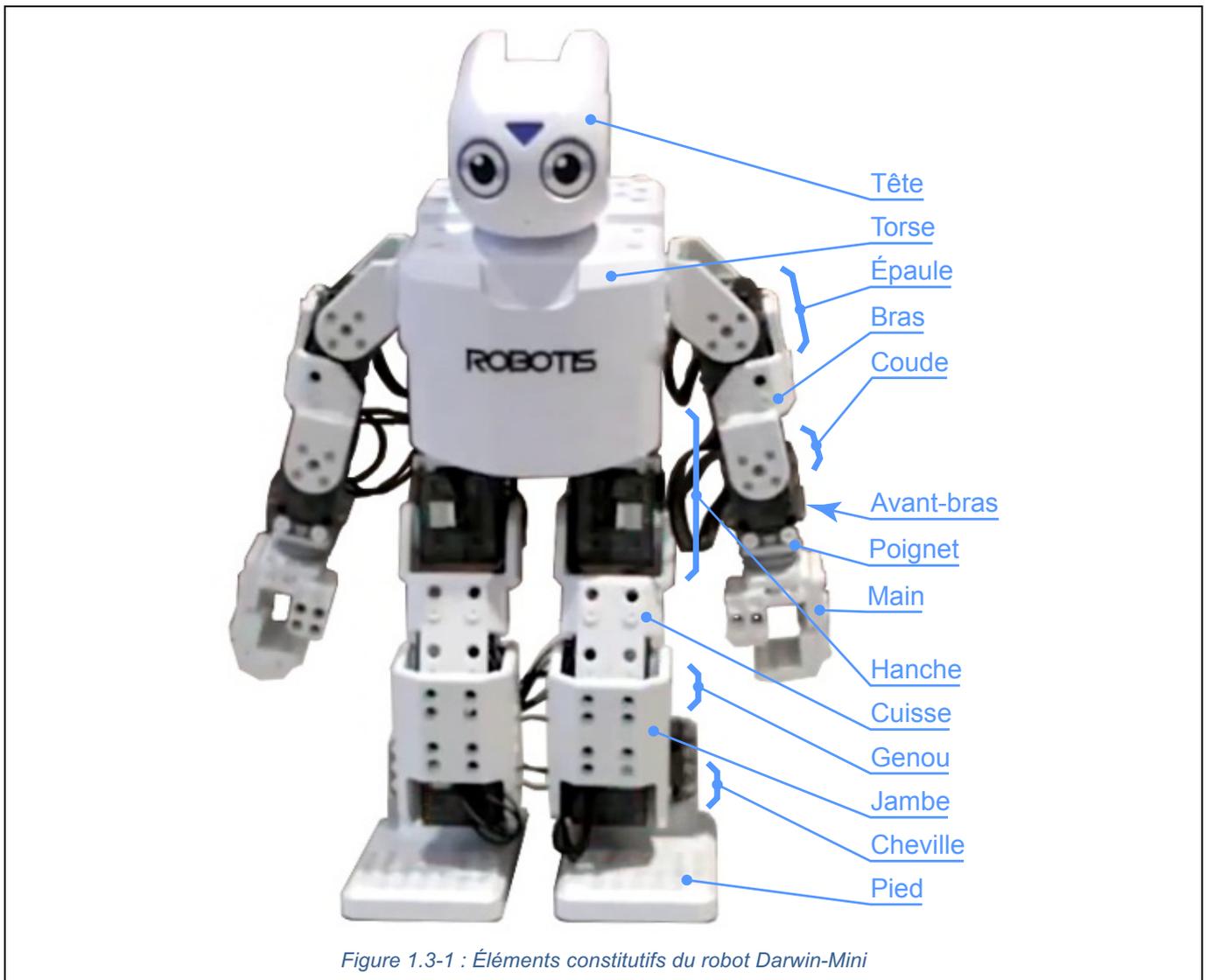


Figure 1.3-1 : Éléments constitutifs du robot Darwin-Mini

### 1.3.2. Mouvements et servomoteurs

En mécanique, le **mouvement** est le déplacement caractérisé par la **position** et la **vitesse** d'un solide par rapport à un autre solide pris comme référence.

Dans le robot, les seuls mouvements possibles sont des **mouvements de rotation autour d'axes**. Les liaisons correspondantes sont appelées des « **articulations** ». Par exemple, dans l'articulation entre le **bras** et l'**avant-bras**, le mouvement de l'**avant-bras** par rapport au **bras** (voir Figure 1.3-1) est une rotation caractérisée à chaque instant par :

- ✓ l'**angle** qu'ils forment entre eux ;
- ✓ la **vitesse de rotation**.

La mise en mouvement des différents membres du robot est assurée par des **servomoteurs** : 3 pour chaque bras et 5 pour chaque jambe, soit 16 au total. La Figure 1.3-2 donne la latéralisation du robot, l'implantation des servomoteurs ainsi que leur numéro d'identifiant (ID).

#### REMARQUES

- La tête est fixe par rapport au torse, et les poignets sont fixes par rapport aux avant-bras.
- Les numéros d'identifiants impairs correspondent au côté droit et les pairs au côté gauche.

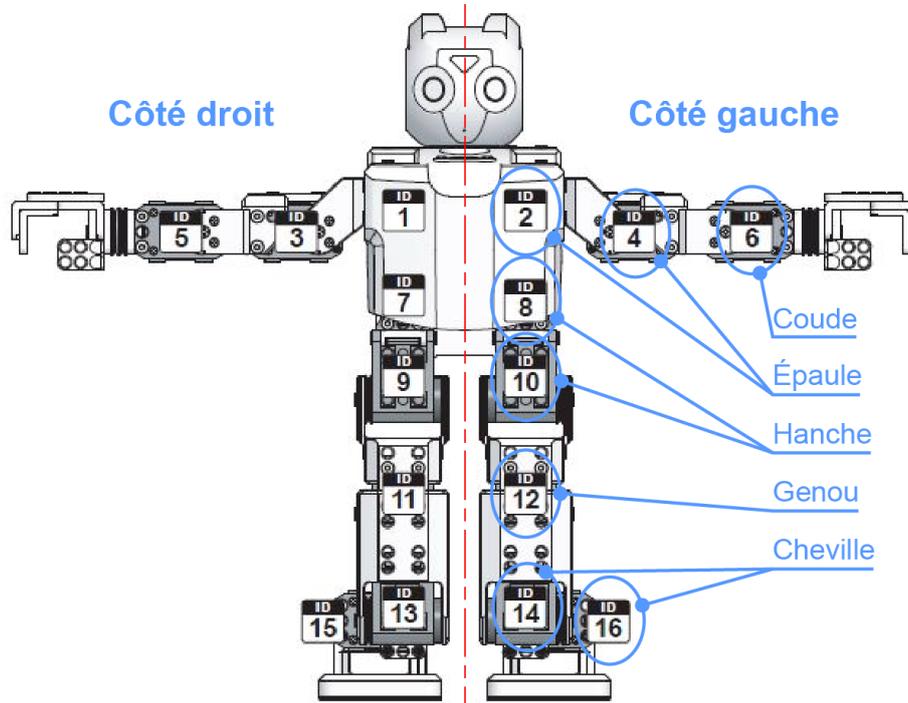


Figure 1.3-2 : Implantation des servomoteurs et latéralisation

Pour animer le robot, il suffira de créer les mouvements adaptés au niveau de chaque articulation, donc d'envoyer des ordres adéquats aux servomoteurs correspondants.

### 1.3.3. Caractéristiques des servomoteurs

Les servomoteurs sont des constituants motorisés capables d'atteindre des positions prédéterminées, puis de les maintenir. Ils sont constitués d'un petit moteur électrique et d'un réducteur de vitesse à engrenages afin de réduire la vitesse de rotation du moteur.

Les servomoteurs utilisés dans Darwin-Mini sont des « Dynamixel XL-320 » (Figure 1.3-3).

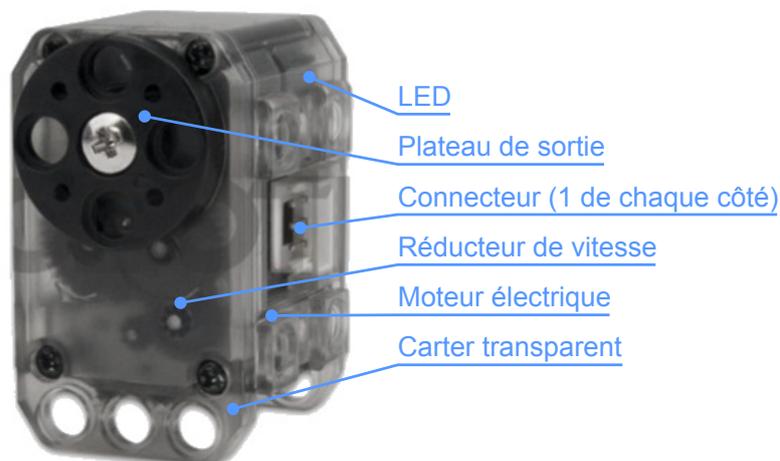


Figure 1.3-3 : Vue globale d'un servomoteur Dynamixel XL-320



Poids	16.7g
Dimensions	24 x 36 x 27 mm
Signal de commande	Paquet Numérique
Couple (kg.cm)	3.98
Couple (N.m)	0.39
Vitesse (tr/min)	114
Tension d'alimentation	6 ~ 8.4V
Tension maximale	8.4V
Tension recommandée	7.4V
Rapport de réduction	238 : 1
Rotation minimum	Environ 0.29 degrés x 1024
Tension	7.4 V
Température de fonctionnement	-5° ~ +70°
ID	254 ID (0~253)
Vitesse de transmission	7843bps ~ 1 Mbps
Retour d'informations	Position, de température, de charge, tension d'entrée, etc
Moteur	Cored

Le servomoteur est équipé d'une Diode Electro Luminescente (DEL ou LED) qui clignote une fois si la puissance est correctement fournie au servomoteur via le câblage. La couleur de clignotement varie suivant la nature de l'information qui est transmise.



## 2. Le produit didactique

### 2.1. Présentation

Darwin-Mini est livré démonté dans une petite mallette (Figure 2.1-1).

La première étape consiste donc à réaliser son assemblage. L'application **R+Design** est faite pour faciliter cette étape en proposant des vues 3D et des éclatés des assemblages.



Figure 2.1-1 : Darwin-Mini assemblé et sa mallette

Une fois assemblé, Darwin-Mini est prêt pour fonctionner et être piloté et programmé. Ceci sera réalisé avec les applications R+Motion et R+Task.

### 2.2. Suite logicielle RobotPlus 2.0

Darwin-Mini est livré avec une suite logicielle qui fonctionne sur un ordinateur PC mais aussi sur smartphone ou sur tablette. Les paragraphes qui suivent donnent les grandes lignes du contenu de ces applications sans rentrer dans le détail de leur utilisation qui fera l'objet de fiches guides.

#### 2.2.1. Applications pour smartphone et tablette

Ces applications permettent de piloter le robot comme avec une télécommande. Les trois applications qui suivent sont aussi exploitables sur ces appareils. Elles sont repérables par le « m. » qui figure dans le nom de l'application.

#### 2.2.2. R+Design et R+m.Design

##### 2.2.2.1. Fonctions

Applications pour **manipuler virtuellement** le « **robot 3D** » (maquette numérique 3D du robot) afin de :

- ✓ dérouler le film de l'assemblage initial du robot (Figure 2.2-2-a) ;
- ✓ visualiser la nomenclature complète des constituants (Figure 2.2-2-b);
- ✓ visualiser ses composants (Figure 2.2-2-c);



- ✓ créer des éclatés (Figure 2.2-2-d);
- ✓ concevoir et constituer d'autres variantes d'assemblages.

### 2.2.2.2. Icones

La figure ci-dessous montre de gauche à droite, les icones de l'application d'ordinateur, de l'application mobile et des fichiers générés par les applications.



Figure 2.2-1 : Icones de R+Task

### 2.2.2.3. Écrans

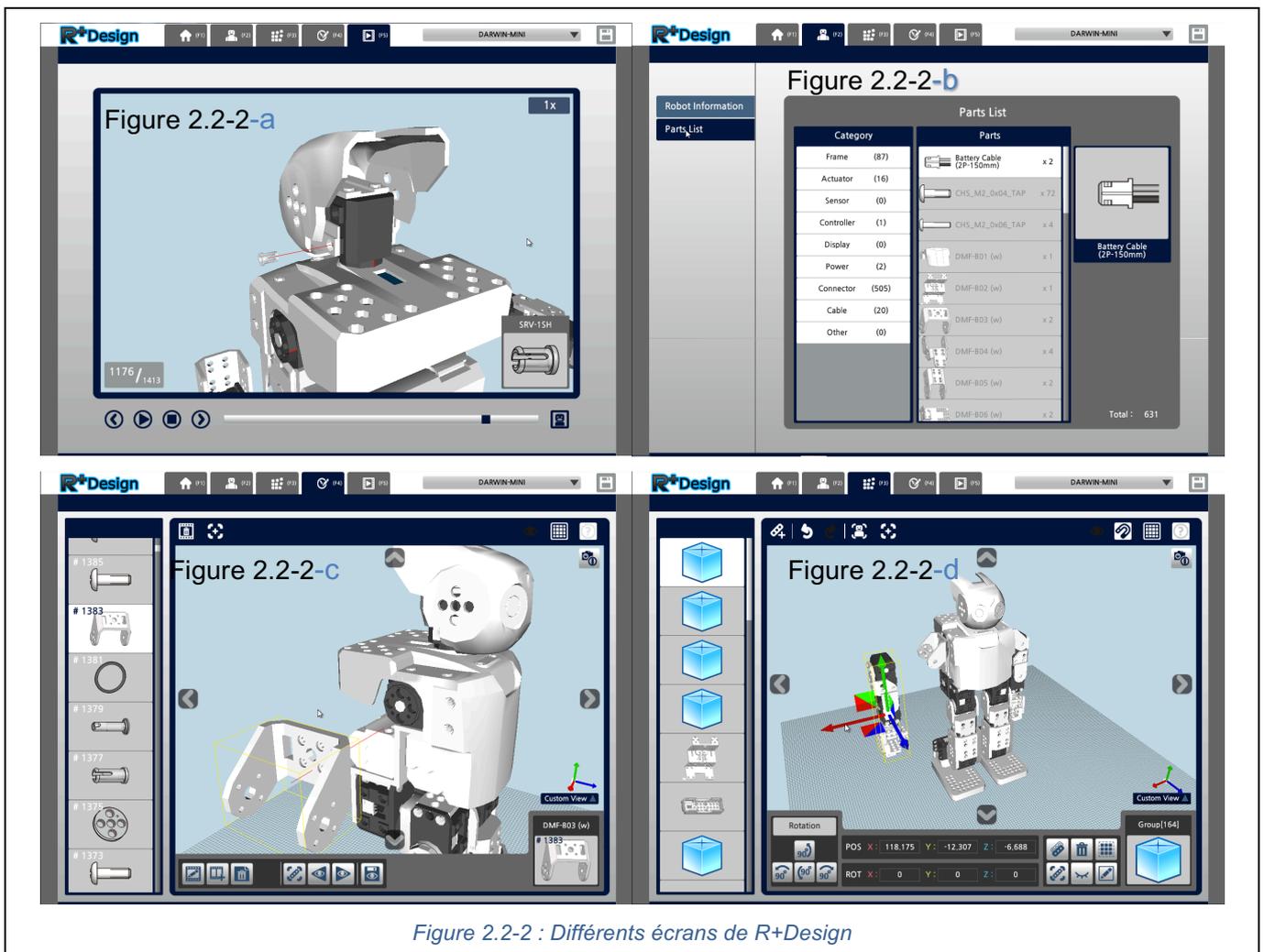


Figure 2.2-2 : Différents écrans de R+Design



## 2.2.3. R+Motion et R+m.Motion

### 2.2.3.1. Fonctions

Applications pour **gérer** la configuration et les mouvements du **robot réel** et de **robot 3D** en vue de :

- ✓ **créer, charger, modifier** ou **sauvegarder** des projets de mouvements et **gérer** les servomoteurs (Figure 2.2-2-a) ;
- ✓ **Piloter** en direct des mouvements sur un ou plusieurs servomoteurs (Figure 2.2-2-b) ;
- ✓ **créer, charger, modifier** ou **sauvegarder** des mouvements élémentaires (motion unit) et les **exécuter** (Figure 2.2-2-c) ;
- ✓ **créer, charger, modifier** ou **sauvegarder** des mouvements et des groupes de mouvements et les **exécuter** (Figure 2.2-2-d).

### 2.2.3.2. Icones

La figure ci-dessous montre de gauche à droite, les icones de l'application d'ordinateur, de l'application mobile et des fichiers générés par les applications.



### 2.2.3.3. Fichier de mouvement

Un **mouvement** est un ensemble de **données de positions et de vitesses** pour les moteurs. Ces données deviennent par la suite un mouvement exécutable pour le robot.

Un fichier de mouvement contient donc des mouvements élémentaires et un enchaînement de ceux-ci.

Si le robot est connecté à l'ordinateur en Bluetooth ou avec un câble USB, le fichier correspondant peut être téléchargé sur le robot et exécuté en direct.

Le fichier de mouvement a une extension « mtnx ».

### 2.2.3.4. Édition d'un fichier de mouvement

L'édition d'un fichier de mouvement est divisée en 4 phases (Figure 2.2-4) qui doivent être abordées dans un ordre séquentiel. On y accède par les onglets du menu en haut de l'écran.

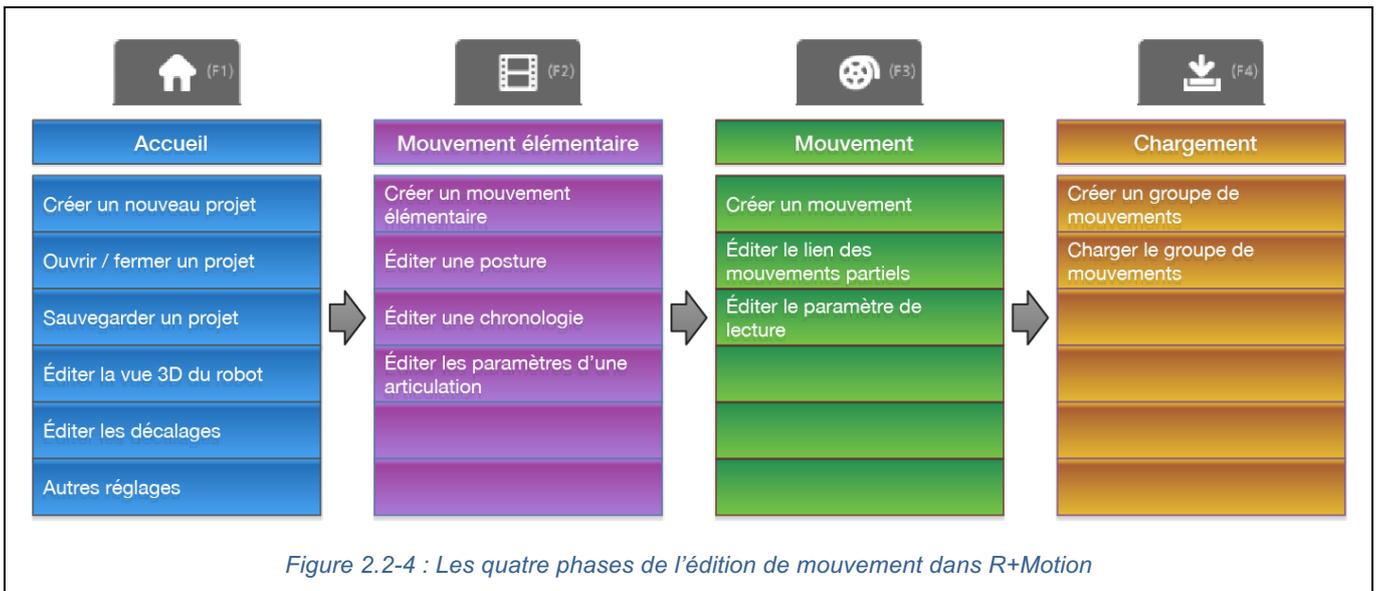


Figure 2.2-4 : Les quatre phases de l'édition de mouvement dans R+Motion

■ **Écrans**

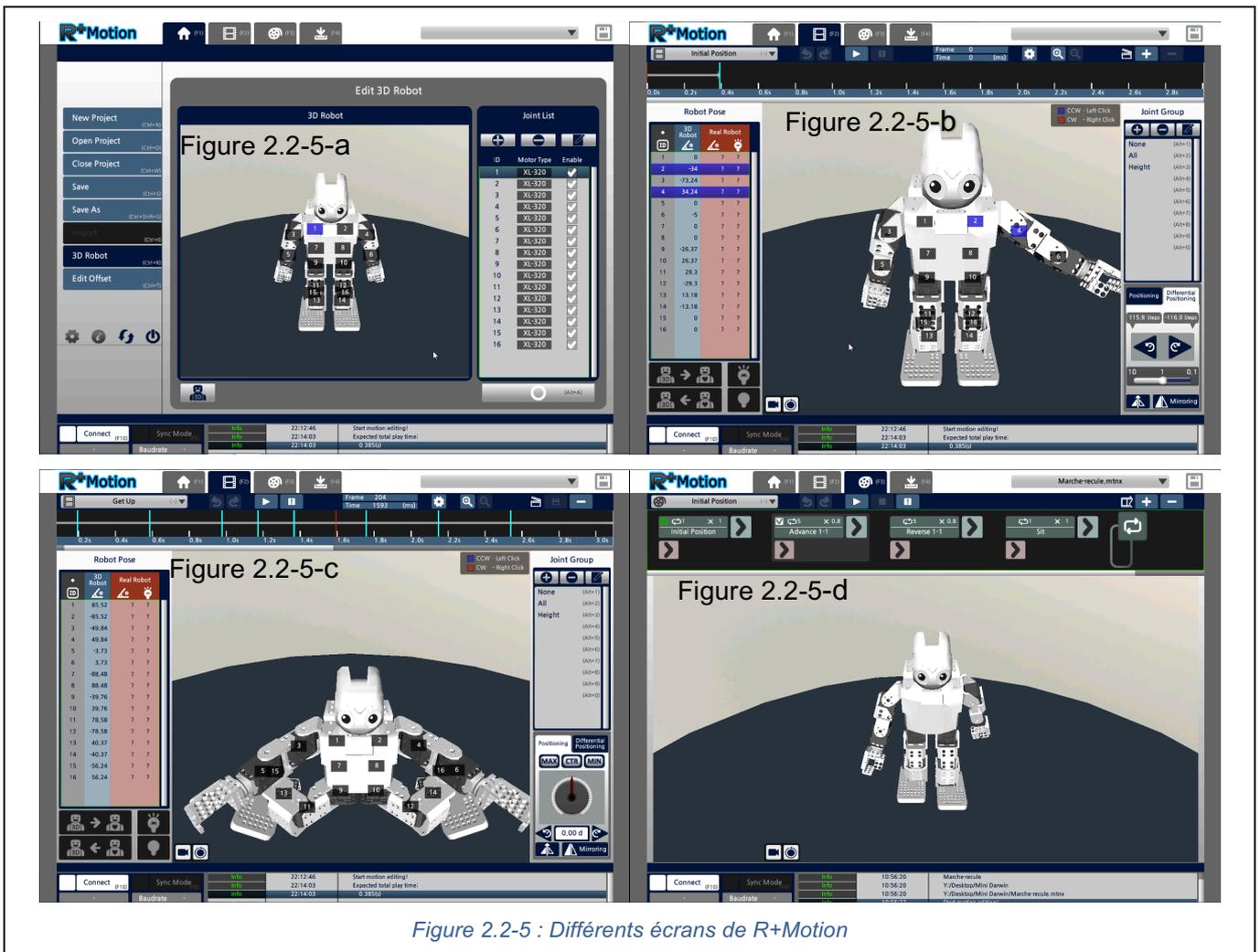


Figure 2.2-5 : Différents écrans de R+Motion



■ **Mouvements élémentaires prédéfinis**

Le tableau ci-dessous donne la liste de quelques mouvements élémentaires prédéfinis.

Numéro d'index dans Motion	Nom tel que dans R+Motion	Traduction	Action du robot
1	Initial Position	Position initiale	Se met en position initiale (debout)
2	Get Up	Se relever	Se relève depuis la position assise sur le sol
3	Sit	S'asseoir	S'assoit
4	Stand Up	Se lever	Se lève depuis la position accroupie
5	Greet 1	Saluer	Salue en se penchant en avant, les bras tendus le long du corps
6	Greet 2	Saluer	Salue en se penchant en avant le bras droit plié devant, le bras gauche plié dans le dos
7	Right Jab	Coup droit	Attaque en portant un coup avec le bras droit
8	Right Hook	Crochet droit	Attaque en faisant un crochet avec le bras droit
9	Right Oppercut	Uppercut droit	Attaque en portant un coup de poing droit remontant
10	Left Jab	Coup gauche	Attaque en portant un coup avec le bras droit
11	Left Hook	Crochet gauche	Attaque en faisant un crochet avec le bras droit
12	Left Oppercut	Uppercut gauche	Attaque en portant un coup de poing gauche remontant
13	Right Hand Wave	Onde de la main droite	
14	Left Hand Wave	Onde de la main gauche	
15	Right Turn	Rotation à droite	Fait quelques pas pour pivoter de 45° environ vers la droite
16	Left Turn	Rotation à gauche	Fait quelques pas pour pivoter de 45° environ vers la gauche
17	Sidestep Right	Déplacement latéral à droite	Fait quelques pas pour se déplacer latéralement à droite
18	Sidestep Left	Déplacement latéral à gauche	Fait quelques pas pour se déplacer latéralement à gauche
19	Advance	Avance	Fait quelques pas de marche en avant
20	Reverse	Sens inverse	Fait quelques pas de marche en arrière



21	Front Roll			
22	Sit Up			
23	Push Up			
24	Back Roll			
25	Left Attack 1			
26	Right Attack 1			
27	Left Attack 2			
28	Right Attack 2			
29	Front Attack			
30	Left Kick			
31	Right Kick			
32	Left Side Kick			
33	Right Side Kick			
34	Block 1			
35	Left Block			
36	Right Block			
37	Block 2			
38	Hand Stand			
39	Fast Advance			
40	Bboy1			
41	Bboy2			
42	Horse Dance			

## 2.3. Scratch pour Darwin-Mini

### 2.3.1. Fonction

Scratch pour Darwin-Mini est une application qui fait appel à la programmation graphique et dynamique par blocs. Elle permet de programmer très facilement le robot.

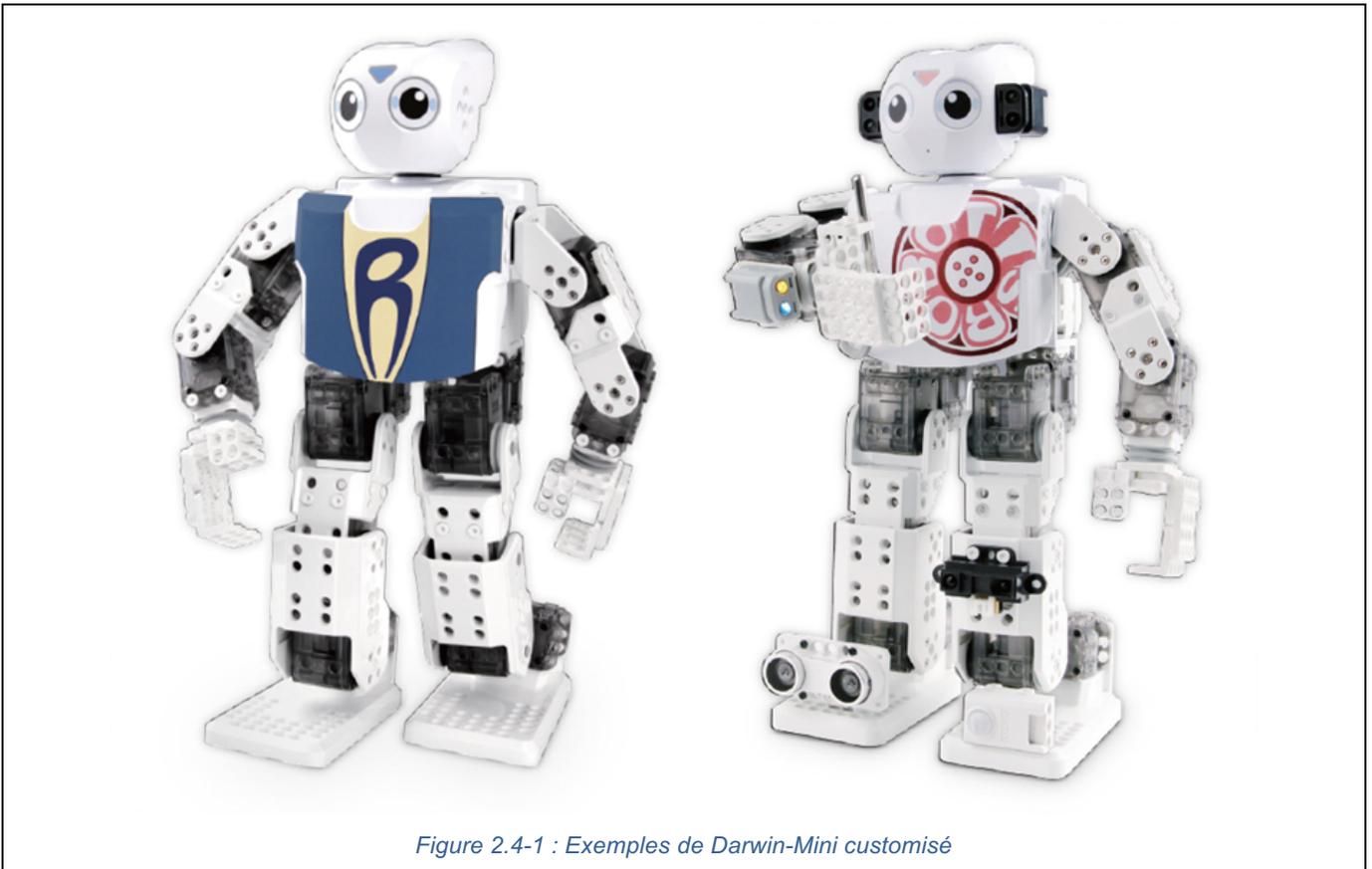


### 2.3.2. Installation

### 2.3.3. Présentation

## 2.4. Customisation du robot

Le robot livré initialement peut être adapté en fonction des besoins par le changement des autocollants ou par l'ajout de dispositifs périphériques tels que des capteurs (figure ci-dessous).





## 2.4.1. Dispositifs périphériques

### 2.4.1.1. Inventaire des dispositifs actuels

Le tableau suivant donne un aperçu de ces dispositifs en précisant leur nom, leur type (entrée ou sortie), leur image et le port sur lequel ils peuvent être connectés (voir le paragraphe suivant pour la connexion).

Nom du dispositif	Image	Type	Ports spécifiques de connexion
Module LED (LED Module LM-10)		Sortie	2 et 3
Capteur infrarouge (IR Sensor IRSS-10)		Entrée	1 et 4
Capteur de distance (DMS Sensor DMS-80)		Entrée	Tous
Capteur à contact (Touch Sensor TS-10)		Entrée	Tous
Capteur de couleur (Color Sensor CS-10)		Entrée	2 et 3
Capteur magnétique (Magnetic Sensor MGSS-10)		Entrée	Tous
Capteur de température (Temperature Sensor TPS-10)		Entrée	Tous
Capteur gyroscopique (Gyro sensor GS-12)		Entrée	Tous
Détecteur d'objet (Passive InfraRed Sensor PIR-10)		Entrée	Tous

Chacun de ces dispositifs est détaillé ci-après.



### 2.4.1.2. Connexion

Ces dispositifs peuvent être connectés sur la carte contrôleur « OpenCM9.04 » qui équipe le robot (figure ci-dessous). Quatre ports numérotés de (1) à (4) sont matérialisés par des connecteurs mâles à 5 broches qui reçoivent les connecteurs complémentaires femelles des capteurs.

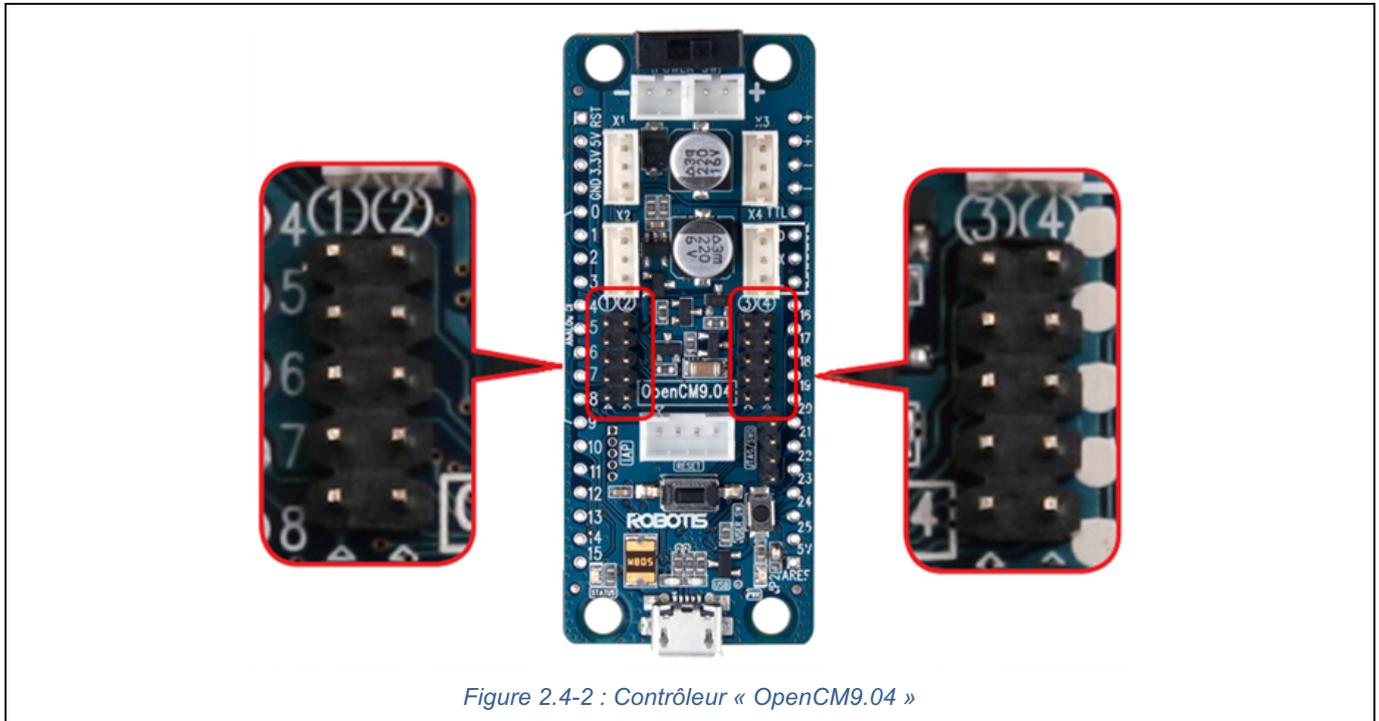
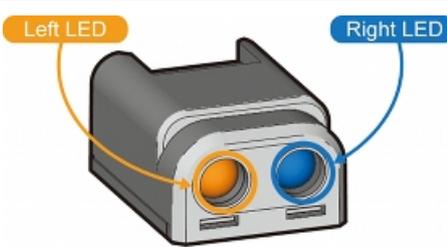


Figure 2.4-2 : Contrôleur « OpenCM9.04 »

### 2.4.1.3. Module LED (LED Module LM-10)

Image	Utilisation	Spécifications												
	Ce module est utilisé pour faire de la lumière avec deux types de LED : Orange et Bleu. Chaque LED peut être contrôlée séparément par l'intermédiaire du port de connexion.	Masse : 4 g Dimensions (mm) : 24 x 18 x 12												
		<b>Ports spécifiques de connexion</b>  2 et 3												
<b>Programmation dans R+Task</b>														
Chacune des deux LED peut être éteinte (OFF) ou allumée (ON). La LED orange est nommée "Left" et la bleue est nommé "Right". Si le module LED est connecté sur le port [2], en fonction de la combinaison d'éclairage que l'on veut réaliser, il suffira donc d'entrer l'une des quatre lignes ci-dessous :														
<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="padding: 2px;">🔔 PORT[2]:LED Module</td> <td style="padding: 2px;">=</td> <td style="padding: 2px;">[Left OFF, Right OFF] (0)</td> </tr> <tr> <td style="padding: 2px;">🔔 PORT[2]:LED Module</td> <td style="padding: 2px;">=</td> <td style="padding: 2px;">[Left OFF, Right ON] (1)</td> </tr> <tr> <td style="padding: 2px;">🔔 PORT[2]:LED Module</td> <td style="padding: 2px;">=</td> <td style="padding: 2px;">[Left ON, Right OFF] (2)</td> </tr> <tr> <td style="padding: 2px;">🔔 PORT[2]:LED Module</td> <td style="padding: 2px;">=</td> <td style="padding: 2px;">[Left ON, Right ON] (3)</td> </tr> </table>			🔔 PORT[2]:LED Module	=	[Left OFF, Right OFF] (0)	🔔 PORT[2]:LED Module	=	[Left OFF, Right ON] (1)	🔔 PORT[2]:LED Module	=	[Left ON, Right OFF] (2)	🔔 PORT[2]:LED Module	=	[Left ON, Right ON] (3)
🔔 PORT[2]:LED Module	=	[Left OFF, Right OFF] (0)												
🔔 PORT[2]:LED Module	=	[Left OFF, Right ON] (1)												
🔔 PORT[2]:LED Module	=	[Left ON, Right OFF] (2)												
🔔 PORT[2]:LED Module	=	[Left ON, Right ON] (3)												

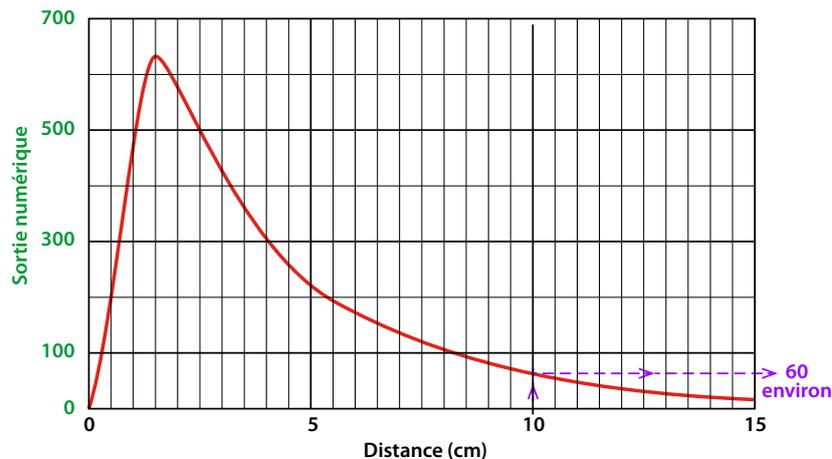


**2.4.1.4. Capteur infrarouge (IR Sensor IRSS-10)**

Image	Principe et utilisation	Spécifications
	<p>Ce type de capteur est utilisé pour détecter des parois ou des objets situés à une distance donnée.</p> <p>Il produit une lumière infrarouge et calcule la quantité réfléchi. La lumière infrarouge ne convient pas pour mesurer la distance précise entre les objets parce que même si deux objets sont à la même distance, la quantité de lumière réfléchi peut varier en fonction de la couleur et de la luminosité.</p>	<p>Masse : 4 g</p> <p>Dimensions (mm) : 24 x 18 x 12</p>
		<p><b>Ports spécifiques de connexion</b></p> <p>1 et 4</p>

**Relation entre distance et valeur de sortie du capteur**

Le capteur infrarouge génère une **valeur numérique** comprise entre 0 et 650 (environ) qui dépend de la distance de l'objet détecté, comme le montre la courbe réponse du capteur (sortie numérique) en fonction de la distance de l'objet (en cm). Par exemple, pour un objet se situant à 10 cm du capteur, la valeur numérique de sortie de celui-ci sera d'environ 60.



**Programmation dans R+Task**

Pour un capteur infrarouge connecté au port [1] et si l'on veut s'approcher jusqu'à 10 cm d'un objet, on pourra utiliser l'une des structures suivantes :

```
IF (  PORT[1]:IR Sensor > 60 )
{
  // Instructions à exécuter
}
```

**Si** la valeur numérique de sortie du **capteur infrarouge connecté au port [1]** est **supérieure** à la valeur **60** (10 cm environ), **alors** exécuter les instructions...

Autrement dit, si le capteur détecte un objet, alors...

```
WAIT WHILE (  PORT[1]:IR Sensor > 60 )
```

**Attendre tant que** la valeur numérique de sortie du **capteur infrarouge connecté au port [1]** est **supérieure** à la valeur **60** (10 cm environ).



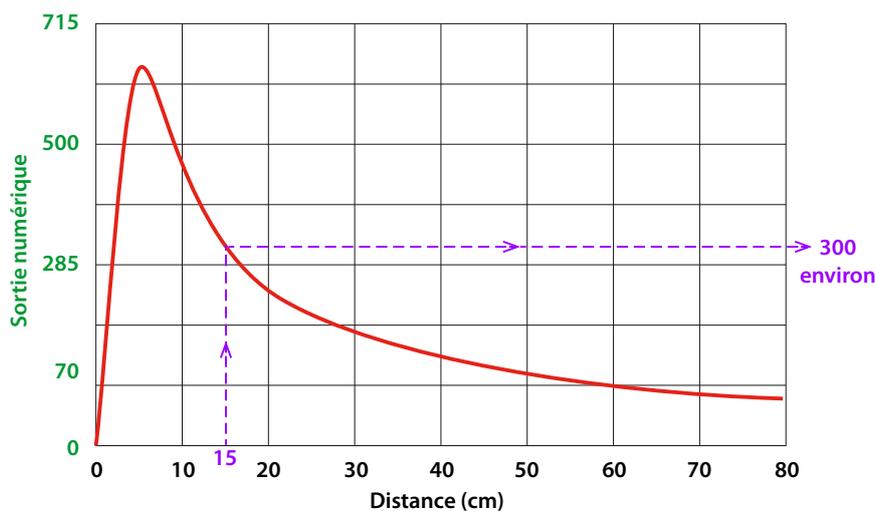
**2.4.1.5. Capteur de distance (DMS Sensor DMS-80)**

Images	Utilisation	Spécifications		
	<p>Ce type de capteur est utilisé pour détecter des objets ou des parois situés à une distance donnée.</p> <p>Contrairement aux capteurs infrarouges, il n'est pas été affecté par la couleur de l'objet.</p>	<p>Masse : 4,4 g Distance de détection : 10 à 80 cm Tension recommandée : 4,5 ~ 5,5 V</p> <table border="1" data-bbox="911 577 1477 712"> <thead> <tr> <th data-bbox="911 577 1477 645">Ports spécifiques de connexion</th> </tr> </thead> <tbody> <tr> <td data-bbox="911 645 1477 712">Tous</td> </tr> </tbody> </table>	Ports spécifiques de connexion	Tous
Ports spécifiques de connexion				
Tous				

**Relation entre distance et valeur de sortie du capteur**

Le diagramme ci-dessous donne la courbe de la réponse du capteur (sortie numérique) en fonction de la distance de l'objet (en cm).

Par exemple, pour un objet se situant à 15cm du capteur, la grandeur de sortie de celui-ci sera d'environ 300.



**Programmation dans R+Task**

Pour un capteur de distance connecté au port [1] et si l'on veut s'approcher jusqu'à 15 cm d'un objet , on pourra utiliser l'une des structures suivantes :

```
IF (  > 300 )
{
  // Instructions à exécuter
}
```

Si la valeur numérique de sortie du **capteur de distance connecté au port [1]** est **supérieure** à la valeur **300** (15 cm environ), **alors** exécuter les instructions...

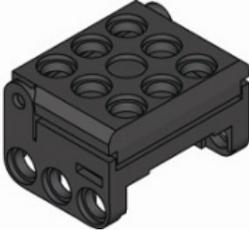
Autrement dit, si le capteur détecte un objet, alors...

```
WAIT WHILE (  > 300 )
```

**Attendre tant que** la valeur numérique de sortie du **capteur DMS connecté au port [1]** est **supérieure** à la valeur **300** (15 cm environ).



**2.4.1.6. Capteur à contact (Touch Sensor TS-10)**

Images	Utilisation	Spécifications	
	<p>Ce capteur est un simple interrupteur qui fonctionne par <b>contact direct</b> avec un objet.</p> <p>Par exemple, il peut être utilisé sur une main du robot pour détecter un impact.</p>	<p>Masse : 4 g</p> <p>Dimensions (mm) : 24 x 18 x 15</p>	
		<p><b>Ports spécifiques de connexion</b></p>	
		<p>Tous</p>	

**Programmation dans R+Task**

Le capteur à contact génère une valeur binaire VRAI (TRUE, ou 1) lorsqu'il détecte un objet ou FAUX (FALSE, ou 0) lorsqu'il ne détecte rien.

Pour un capteur à contact connecté au port [1], on pourra utiliser l'une des structures suivantes :

```
IF (  PORT[1]:Touch Sensor == TRUE (1) )
{
  // Instructions à exécuter
}
```

**Si** la valeur binaire de sortie du **capteur à contact connecté au port [1]** est (==) **VRAI**, alors exécuter les instructions...

Autrement dit, si le capteur entre en contact avec un objet, alors...

```
WAIT WHILE (  PORT[1]:Touch Sensor != FALSE (0) )
```

**Attendre tant que** la valeur binaire de sortie du **capteur à contact connecté au port [1]** est différente (!=) de **FAUX** (contact avec un objet).

**2.4.1.7. Capteur de couleur (Color Sensor CS-10)**

Images	Utilisation	Spécifications	
	<p>Ce capteur permet de détecter les couleurs.</p> <p>Il peut distinguer 6 couleurs différentes : <b>rouge</b>, <b>bleu</b>, <b>vert</b>, <b>jaune</b>, blanc et <b>noir</b>.</p> <p>Lorsque des objets sont trop proches (moins de 6 mm) ou trop loin (plus de 18 mm), le capteur de couleur ne peut pas interpréter correctement les valeurs des mesures.</p>	<p>Masse : 4 g</p> <p>Dimensions (mm) : 24 x 18 x 12</p>	
		<p><b>Ports spécifiques de connexion</b></p>	
		<p>2 et 3</p>	



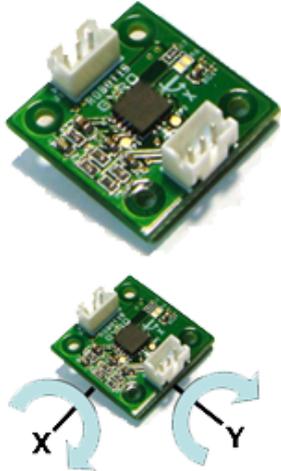
**2.4.1.8. Capteur magnétique (Magnetic Sensor MGSS-10)**

Images	Utilisation	Spécifications
	<p>Ce capteur détecte les champs magnétiques.</p> <p>Le capteur se comporte comme un capteur tactile et détecte un aimant de près ou de loin.</p> <p>Il faut placer l'aimant à environ 5 mm devant le capteur pour qu'il se comporte comme un capteur tactile.</p>	<p>Masse : 3 g</p> <p>Dimensions (mm) : 24 x 18 x 12</p> <p>Distance maximale entre les pôles N / S de l'aimant et la face avant du capteur : 1,5 cm</p> <p>L'aimant ne sera pas détecté si la distance est inférieure à 5 mm</p>
		<b>Ports spécifiques de connexion</b>
		Tous

**2.4.1.9. Capteur de température (Temperature Sensor TPS-10)**

Images	Utilisation	Spécifications
	<p>Ce capteur détecte les variations de température.</p>	<p>Masse : 18 g</p> <p>Câble : 1 m</p> <p>Plage de températures : -20°C ~ 120°C</p>
		<b>Ports spécifiques de connexion</b>
		Tous

**2.4.1.10. Capteur gyroscopique (Gyro sensor GS-12)**

Images	Utilisation	Spécifications
	<p>Capteur de vitesse angulaire suivant deux axes (X et Y).</p> <p>Il permet de détecter de quel côté le robot s'incline.</p> <p>Il est utilisé pour maintenir l'équilibre du robot et peut être utilisé pour d'autres applications de mouvement.</p>	<p>Masse : 2,8 g</p> <p>Dimensions (mm) : 23 x 23 x 10</p> <p>Températures de fonctionnement : -40°C ~ 85°C</p> <p>Plage de vitesse angulaire : -300°/s ~ 300°/s</p> <p>Bande passante : 140 Hz</p> <p>Sensibilité : 3,33 mV/dps</p> <p>Tension recommandée : 4,5 ~ 5,5 V</p>
		<b>Ports spécifiques de connexion</b>
		Tous

**2.4.1.11. Détecteur d'objet (Passive InfraRed Sensor PIR-10)**



Images	Utilisation	Spécifications
	<p>Capteur de détection de mouvements. Il est utilisé pour détecter le mouvement d'un objet.</p> <p>Le mouvement de l'objet détecté doit se produire dans les plages suivantes :</p> <p>Plage de mesure : 20 cm ~ 2 m</p> <p>Angle de mesure : +/- 45 degrés</p>	<p>Masse : 10 g</p> <p>Dimensions (mm) : 18 x 30 x 8</p>
		<p><b>Ports spécifiques de connexion</b></p> <p>Tous</p>
<p><b>Programmation dans R+Task</b></p>		
<p>Le détecteur d'objets génère une valeur binaire VRAI (TRUE, ou 1) lorsqu'il détecte un objet ou FAUX (FALSE, ou 0) lorsqu'il ne détecte rien.</p> <p>Pour un détecteur d'objets connecté au port [1], on pourra utiliser la structure suivante :</p> <div style="display: flex; justify-content: space-between;"> <div data-bbox="132 972 802 1144" style="border: 1px solid #ccc; padding: 5px; background-color: #f9f9f9;"> <pre>IF (  PORT[1]:Passive Infrared Sensor == FALSE (0) ) {     // Instructions à exécuter }</pre> </div> <div data-bbox="831 972 1485 1144"> <p><b>Si la valeur binaire de sortie du <b>détecteur d'objets connecté au port [1]</b> est (==) <b>FAUX</b>, alors exécuter les instructions...</b></p> <p>Autrement dit, si le capteur ne détecte aucun mouvement d'objet, alors...</p> </div> </div>		

### 2.4.2. Apparence du robot

Darwin-Mini est livré avec deux planches d'autocollants de façon à modifier son apparence en fonction des besoins. Ces planches sont téléchargeables sur le site de [Robotis](http://Robotis.com).