



## Règlement

*Version : 1.03  
version finale*

Avec le soutien du





## Table des matières

1.	AVANT-PROPOS .....	3
1.1.	Description .....	3
1.2.	Contacts .....	3
1.3.	Références .....	3
2.	PRESENTATION GENERALE .....	4
3.	LE QUARTIER ROBOTCITE .....	5
3.1.	Les différentes zones .....	5
3.2.	La voirie.....	5
3.3.	Les bornes de recharge .....	5
4.	LES ROBOTS.....	6
4.1.	Base Roulante.....	6
4.2.	Carte PC embarquée .....	6
4.3.	Dimensions du robot .....	7
4.4.	Restrictions/Contraintes sur la création.....	7
5.	STRUCTURE INFORMATIQUE .....	9
5.1.	Le serveur .....	10
5.2.	Logiciel Embarqué.....	10
5.3.	Aperçu de la programmation .....	11
5.4.	Développement informatique .....	12
5.5.	Interface PC embarqué et électronique personnelles des équipes .....	15
5.6.	Plus d'informations .....	17
6.	CONSIGNES DE SECURITE .....	18
6.1.	Tension à bord .....	18
6.2.	Air comprimé .....	18
6.3.	Laser .....	19
6.4.	Sources lumineuses de forte puissance.....	19
7.	DOCUMENTS A RENDRE ET ECHEANCES .....	20
7.1.	Le dossier d'avant-projet .....	20
7.2.	Le workshop :.....	20
7.3.	Le compte-rendu .....	20
7.4.	Le Challenge .....	21



## **1. AVANT-PROPOS**

### **1.1. Description**

Ce document est une description des possibilités offertes et des contraintes que devront respecter les robots participants à RobotCité.

### **1.2. Contacts**

#### **Théo PEROZ**

Chargé de mission RobotCité

Email : [theo.peroz@planete-sciences.org](mailto:theo.peroz@planete-sciences.org)

Adresse : Planète Sciences  
16 place Jacques Brel  
91130 Ris-Orangis

#### **Coordonnées RobotCité**

Site web : [www.robotcite.fr](http://www.robotcite.fr)

Email : [robotcite@planete-sciences.org](mailto:robotcite@planete-sciences.org)

### **1.3. Références**

Ce document fait référence aux documents suivants :

- Conditions d'inscription à la rencontre RobotCité.
- Description technique de la base roulante (document à paraître ultérieurement).
- Description technique du Quartier RobotCité (document à paraître ultérieurement).



---

## **2. PRESENTATION GENERALE**

Projet conjoint de la FING (Fondation Internet Nouvelle Génération) et de Planète Sciences, RobotCité préfigure la ville du futur : un environnement qui communique avec des robots qui déambulent de manière autonome, et surtout, la possibilité pour tout le monde d'interagir avec le quartier de demain.

RobotCité est aussi un défi ouvert aux étudiants, associés aux mondes universitaire et industriel, pour peupler cet environnement de petits robots autonomes.

Le challenge RobotCité est un défi amical de robotique amateur. Pendant un an, des équipes d'étudiants venant d'horizons différents travailleront sur les robots qui composeront notre environnement de la ville de demain.

Les participants doivent concevoir puis réaliser un robot, conforme au règlement et à l'esprit de cette rencontre.

RobotCité n'est pas une compétition mais un défi ouvert aux étudiants. Il s'agit également d'un lieu d'échanges de savoir et de savoir-faire entre les participants.

### **3. LE QUARTIER ROBOTCITE**

Le Quartier RobotCité est une maquette d'environ 40m<sup>2</sup> représentant les principaux lieux de vie d'une ville. Toutes les précisions à propos du quartier se trouvent dans le document « Description Technique du Quartier RobotCité », à paraître ultérieurement.

#### **3.1. Les différentes zones**

Il existe 3 zones distinctes sur la maquette:

- La zone urbaine : ensemble de bâtiments représentatifs d'une ville accessible par un ensemble de route.
- La maison : bâtiment dans lequel les robots pourront évoluer. Ce bâtiment peut avoir une échelle différente des autres structures.
- Le parc : surface dégagée symbolisant un espace naturel. Il peut être peuplé d'arbres et autres accessoires symbolisant la nature.

#### **3.2. La voirie**

Les robots peuvent se déplacer dans le quartier RobotCité. Il existe différents types de routes pour accéder aux bâtiments. Ces routes sont représentatives d'un réseau urbain contemporain. Ainsi la maquette peut être constituée de :

- routes à double sens
- routes à sens unique
- impasses
- etc...

Les robots voulant emprunter les routes devront respecter les principes du code de la route.

#### **3.3. Les bornes de recharge**

Afin que les robots puissent être autonomes durant la durée complète de la rencontre RobotCité, des bornes de recharge seront placées à des endroits stratégiques sur la maquette. Le nombre de bornes de recharge sera adapté au nombre de robots participant.

Il ne faut en aucun cas obstruer les contacts de charge de batterie placés sous le robot. Ceci empêcherait tout bon fonctionnement de la base roulante. Se référer à la documentation de la base roulante pour plus de détails sur le positionnement de ces contacts.



## **4. LES ROBOTS**

Les robots sont le produit d'une collaboration entre plusieurs savoir-faire : ingénierie, design, art, etc. Ci-dessous les contraintes essentiellement techniques pour la réalisation.

Chaque équipe met en jeu au maximum un seul robot.

Le robot est une machine totalement autonome. Il embarque sa propre source d'énergie, ses actionneurs et son système de contrôle.

Le budget pour la création du robot importe à l'équipe d'étudiants participant à RobotCité. Trouver les financements pour la réalisation de son projet fait parti du Challenge RobotCité.

### **4.1. Base Roulante**

Les organisateurs, (Planète Sciences et la FING), fournissent à chaque équipe inscrite une base roulante spécifique au concours RobotCité.

Cette base roulante est une aide pour la construction du robot. Elle embarque les tâches de localisation, d'asservissement des moteurs de propulsion, une caméra et une puissance de calcul embarquée. La base roulante fournie est immédiatement utilisable et fonctionnelle. L'équipe peut ainsi se concentrer sur l'usage du robot et développer sereinement son application. L'utilisation de la base roulante dans son état d'origine est obligatoire.

L'utilisation de la base roulante peut être adaptée par chaque équipe. Il n'est pas obligatoire de ce servir de l'intégralité des actionneurs et capteurs intégrés à la base roulante.

L'ensemble des caractéristiques techniques de la base roulante se trouve dans le document : "Description technique de la base roulante".

Les bases seront distribuées après la validation des projets de chaque équipe par le comité organisateur. Elles seront rendues au comité organisateur à la fin de l'exposition.

Pour plus de détails sur la base roulante, se référer au document cité en référence.

### **4.2. Carte PC embarquée**

La base roulante supporte une carte électronique comprenant :

- Un PC embarqué type PXA270 : la colibri de Toradex cadencée à 520MHz ([www.toradex.com](http://www.toradex.com)) ayant 64Mo de SDRAM
- Une caméra branchée directement sur le port de capture vidéo du processeur
- Une clé usb de stockage de 1Go supportant le système d'exploitation : Linux, environ 800Mo est disponible pour les équipes.
- La carte électronique en elle-même qui supporte les 2 composants précédents et la connectique vers différents périphériques (USB, série RS232, Ethernet RJ45, audio)

### **4.3. Dimensions du robot**

Les équipes peuvent concevoir leur robot avec des mécanismes se déployant. Si de tels mécanismes sont utilisés, le robot effectuera son déploiement en dehors des zones de circulation.

Lors de la présence sur la voie, le périmètre est limité au périmètre de la base roulante. Tous les éléments ajoutés par l'équipe doivent être compris dans la projection verticale de la base roulante.

La hauteur lors de déplacement est limitée à 400mm.

#### **Remarque importante :**

Il est fortement conseillé aux équipes de se ménager une certaine "marge de sécurité" vis à vis de ces contraintes, et de rester plusieurs millimètres en deçà des valeurs indiquées, afin d'éviter toute "mauvaise surprise" lors des phases d'homologation.

### **4.4. Restrictions/Contraintes sur la création**

Sur la base roulante l'équipe ajoute les éléments qu'elle désire : mécanique, électronique, logiciel pour réaliser son usage.

Chaque équipe laisse libre cours à sa créativité, en respectant les contraintes de construction et de sécurité.

L'équipe réalise une étude puis fabrique un robot pour le faire participer au challenge RobotCité.

Le robot ne doit être dangereux ni pour lui, ni pour les autres robots, ni pour le public, ni pour les utilisateurs, ni pour la maquette.

Le robot ne doit pas comporter de partie saillante ou pointue susceptible de provoquer des dégâts ou d'être dangereuse.

Tous les éléments du robots doivent être solidaires les uns des autres.

L'utilisation de produits liquides, corrosifs ou pyrotechniques et d'êtres vivants est interdite.

Tous ces points seront vérifiés à l'arrivée des équipes sur le lieu de la rencontre, avant que les robots ne soient autorisés à participer au challenge.

#### **4.4.1. Mobilité**

Par défaut, le robot possède un système de propulsion intégré dans la base roulante. Il s'agit du seul moyen de propulsion autorisé pour se déplacer.

Si l'équipe veut que son robot ait un rôle qui ne nécessite aucun déplacement, la partie propulsion pourra être désactivée mais pas supprimée. L'équipe devra informer le comité organisateur de son choix technique, afin qu'il soit prévu une alimentation électrique fixe sur le lieu de travail du robot.

#### 4.4.2. Support de cible

Il est demandé aux équipes d'installer un support sur leur robot afin d'accueillir une cible 2D. Cette cible permettra au serveur informatique de localiser le robot sur la maquette. Le poids de la cible sera négligeable.

Ce support devra à tout moment respecter les points suivants :

- avoir une surface plane et horizontale, dont la taille sera un carré de 200 mm de côté, positionnée sur la partie la plus haute du robot.
- il sera situé le plus au centre possible du robot, la distance le séparant d'un bord du robot non déployé ne devant pas faire moins de 50 % que celle le séparant d'un autre bord
- Une cible 2D dédiée à votre robot sera installée sur ce support le jour de la rencontre.
- Aucun objet ne devra se situer de manière permanente ou provisoire dans la projection verticale de la cible. La zone réservée à ce support devra être le plus possible centrée sur le robot.

#### 4.4.3. Aspect Esthétique

Le travail de design du robot, son intégration dans l'environnement humain et dans la ville devra être soigné.

#### 4.4.4. Arrêt d'urgence

Chaque robot devra comporter un arrêt d'urgence standard. Le bouton devra avoir un diamètre minimum de 200mm et être facilement accessible. Aucun obstacle ne doit gêner l'accès à ce dispositif de sécurité.

#### 4.4.5. Charge utile

Après différents tests, nous estimons la charge utile maximale à 8kg. Ne pas dépasser cette charge afin de ne pas endommager la base et d'assurer une manutention aisée.

#### 4.4.6. Visibilité

Deux espaces rectangulaires de 100x70 mm doivent être laissés libres sur 2 faces latérales (au choix) du robot. Les équipes recevront des autocollants imprimés par l'organisation (numéro d'équipe, sponsors de l'événement), qu'elles placeront sur ces espaces libres. Les autocollants pourront être placés sur deux faces différentes du robot.

#### 4.4.7. Eclairage

Durant le Challenge RobotCité, il pourra y avoir des éclairages de fortes puissances sur la maquette. Dans le cas de l'utilisation de caméras ou d'éléments sensibles à la lumière et/ou aux infrarouges, les équipes devront prendre garde à disposer d'un système de calibration de leur matériel optique de manière à pouvoir s'adapter aux conditions lumineuses.

## 5. STRUCTURE INFORMATIQUE

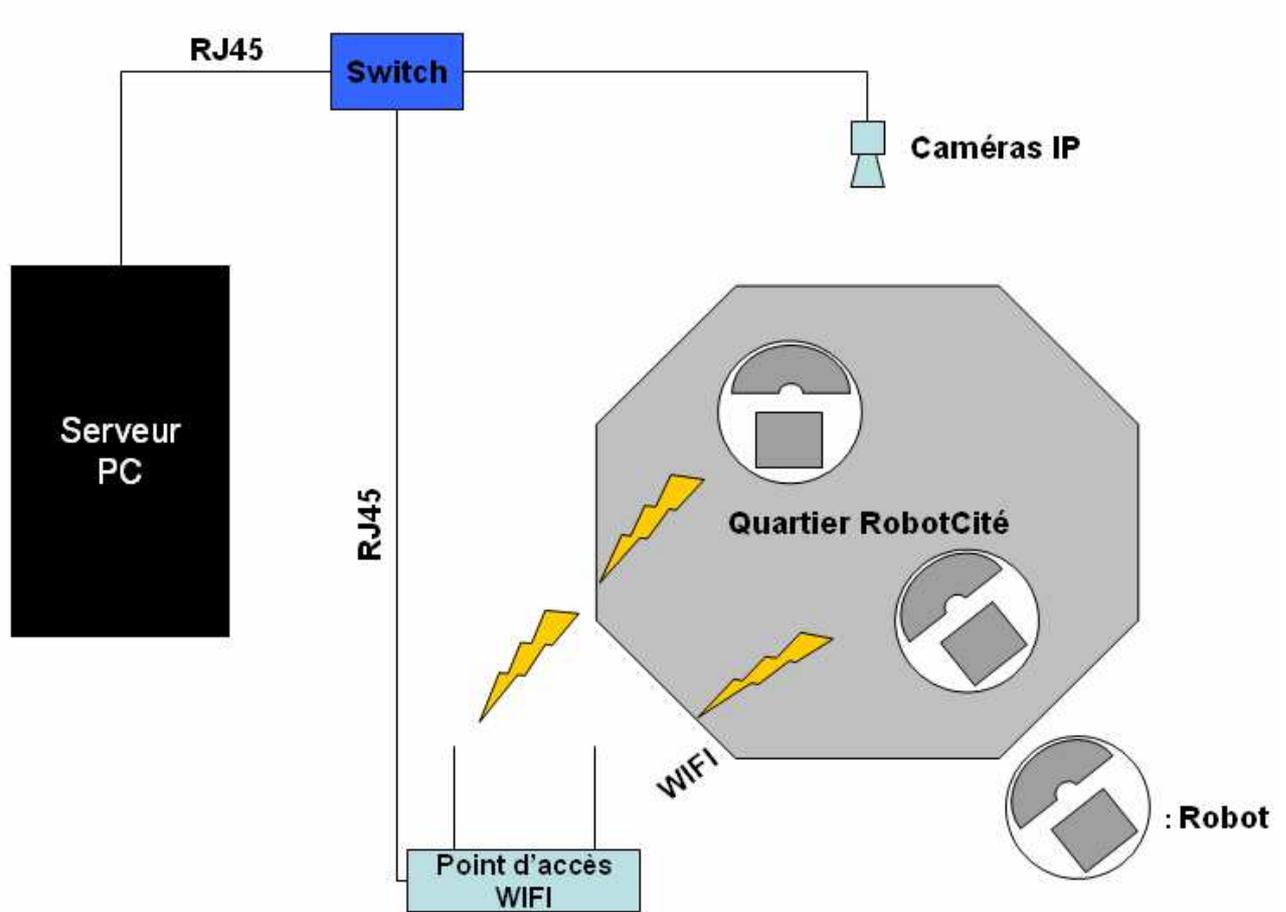


Figure 1 - Architecture globale de la maquette RobotCité

Tous les robots embarquent une carte réseau Wifi. Cette carte réseau permet de se connecter à un routeur Wifi et à un serveur. Cela permet de faciliter la communication entre tous les organes intelligents de RobotCité.



## **5.1. Le serveur**

Le serveur est un ordinateur central, relié au réseau Wifi et donc relié aux robots, qui assure différentes tâches. Ce serveur est entièrement géré par les organisateurs; les équipes n'auront pas accès à cet ordinateur.

Ce paragraphe décrit le fonctionnement de cette machine dans RobotCité pour que vous ayez une vue d'ensemble du système informatique et compreniez les différentes limites logicielles.

Afin de limiter les risques de dysfonctionnement, le serveur s'occupera de surveiller les robots et pilotera les différents éléments interactifs de la maquette.

Les itinéraires et positions des robots seront supervisés par le serveur.

Le programme développé par les participants n'aura qu'à envoyer une simple commande au serveur, pour recevoir un message contenant la localisation du robot dans la ville, ainsi que des trajets proposés pour un déplacement aisé des robots dans la ville.

Il sera possible de connaître les itinéraires et positions des autres robots en envoyant une requête au serveur.

Dans le cas de pilotage à distance par des humains, les requêtes passeront par le serveur qui transférera les informations aux robots concernés après en avoir vérifié la validité.

Si l'application développée par une équipe rencontre un problème d'exécution, le serveur a la capacité de reprendre le contrôle du robot défaillant et de le piloter à son tour pour le ramener à une base de rechargement ou faire de l'animation sur la maquette.

Cela assure un mouvement continu dans le quartier RobotCité, même si les programmes ne fonctionnent plus.

## **5.2. Logiciel Embarqué**

Les bases roulantes fournies par les organisateurs contiennent un PC embarqué.

Dans la mémoire de ce PC se trouvent un ensemble de fichiers informatiques installés. Ces fichiers permettent, au démarrage du PC embarqué, de gérer les signaux de la base roulante et la communication réseau avec l'extérieur.

Ces données informatiques ne doivent en aucun cas être supprimées, ni modifiées à votre initiative.

Le comité organisateur peut éventuellement fournir des mises à jour du programme Urbi en fournissant aux équipes un patch à appliquer.

Une description plus détaillée du logiciel et de son utilisation vous sera remise par la suite avec les bases roulantes.

Dès que vous recevez votre base roulante, telle que les organisateurs vous la fournissent, elle est déjà capable de beaucoup d'actions.



Toutes ces actions ont des commandes simplifiées permettant leur utilisation dans des scripts Urbi:

- Rouler, grâce à un asservissement (odométrie) pour un repérage local de précision ;
- Détecter des obstacles face à elle grâce au "bumper" à avant ;
- Via un script, se recharger automatiquement quand ses batteries sont faibles ;
- Envoyer à un site web/autre robot/PC extérieur la vidéo de sa caméra ;
- Faire de la musique : un buzzer interne permet de jouer des notes simples.

Lorsque votre robot sera sur le quartier RobotCité, il pourra aussi:

- Se localiser dans le quartier RobotCité : en demandant au serveur sa position, le serveur renvoi à la base roulante ses coordonnées dans la ville.
- Se déplacer sans encombre d'un point à l'autre du quartier RobotCité: une requête de déplacement envoyée au serveur, et votre robot se déplace simplement là où vous le désirez (éviter des obstacles et des autres robots).

### **5.3. Aperçu de la programmation**

Lors de l'allumage du robot, après quelques secondes de démarrage du système d'exploitation, la base roulante émet quelques sons pour signifier quelle est prête à l'usage. Ces bips sont générés par un programme, l'"engine", qui est la couche basse du robot.

L'engine est forcément démarré lorsque la carte mère du robot est prête. Dans le cas contraire, le robot ne fonctionnera pas.

Pour piloter le robot, lire ses capteurs et avoir accès au réseau et aux autres robots, il vous faudra interagir avec ce programme.

Pour cela vous avez plusieurs choix:

- Programmation en Urbi ;
- Programmation C/C++ ;
- Programmation en Java / Matlab → non recommandé car le support pour ces langages est limité.

Tous les programmes en rapport avec les bases roulantes ont une partie de code de communication réseau (TCP/IP). Ce code est invisible pour les utilisateurs car il est déjà programmé.

Tout programme fait en Urbi contient forcément une couche réseau, ce qui lui permet d'être exécuté à distance sur un autre PC que celui où se trouve le programme principal ; il est seulement nécessaire que ces 2 machines soient connectées en réseau.



Les organisateurs devront au préalable être avertis de l'utilisation d'un PC extérieur sur le réseau de manière à prévoir la structure adéquate.

La couche Urbi permet de gérer le multitâche nativement.

Si vous demandez au robot d'exécuter 2 tâches simultanément un simple '&' entre 2 commandes permet de faire le multitâche

**Exemple :**

```
Led.val = 1 & MoteurL.speed = 200;
```

**Vocabulaire :**

- Engine : le programme maître qui tourne toujours sur le robot. Il gère les couches informatiques basses du robot (pilotage moteur, capteurs, caméra, réseau,...).
- Object Remote : un programme créé par l'utilisateur pour le robot et exécuté à distance, sur un ordinateur par exemple.
- UObject Remote local : un programme créé par l'utilisateur pour le robot et exécuté sur le robot (autonome).
- Script Urbi: programme en format texte, non compilé et facile à prendre en main.

## **5.4. Développement informatique**

L'équipe devra développer une application numérique capable de contrôler le robot afin de réaliser les actions correspondantes à son projet.

Les équipes peuvent écrire leur programme dans le langage de leur choix (voir les paragraphes précédents).

Dans les premiers temps, nous vous conseillons de vous familiariser avec le robot grâce à la télécommande "Urbi Remote".

Cette télécommande est un programme avec une interface graphique simple et intuitive qui vous permet de piloter en quelques clics le robot.

De plus, cette télécommande vous permettra par la suite de tester simplement vos scripts Urbi (fichiers.u) et aussi vos UObjects compilés, qu'ils soient en local sur votre PC ou sur le robot.

L'ordinateur de développement peut être sous Windows, Linux, ou Mac.



L'environnement recommandé est Linux. Pour les utilisateurs de Windows et dans un souci de flexibilité, un environnement spécial vous sera fourni pour compiler et cross-compiler facilement vos programmes C/C++.

Linux et Mac contiennent déjà GCC. Leurs utilisateurs devront installer un simple package, qui vous sera fourni, pour développer pour RobotCité.

Dans le cas où une équipe n'utilise que des scripts Urbi (fichiers textes), cet environnement de développement est inutile, un simple éditeur de texte suffit!

Dans tous les cas, toute la suite logicielle vous sera accessible pour vous donner le maximum de souplesse.

Par expérience, nous savons que lors de l'utilisation d'Urbi sur des robots, les utilisateurs développent une grosse partie du traitement en C/C++ et peaufinent les appels de leurs programmes par des scripts Urbi. C'est un cas général, il est tout à fait possible de n'utiliser uniquement que les UObjects compilés, ou plus souvent, que les scripts Urbi.

Les logiciels qui vous seront fournis :

- Urbi Remote : la télécommande pour tester facilement le robot ;
- SDK de Cross-Compilation : pour les équipes voulant programmer en C/C++ sur la carte mère du robot ;
- Des petits scripts de démo de la base roulante ;
- WinScp : un logiciel de transfert des fichiers sur le robot (protocole sécurisé SSH).

La documentation associée :

- Documentation Urbi : script, programmation, compilation, etc...
- Documentation spécifique à la base roulante (logiciel Urbi et protocole de communication de la base roulante) ;
- Documentation de la carte PC embarquée.

### 5.4.1. Programmation Scripts Urbi

#### Avantages:

- C'est la méthode la plus simple ;
- Un simple éditeur de texte suffit pour programmer ;
- Pas besoin de compiler le programme ;
- On peut tester le programme à distance sans avoir à le télécharger sur le robot (mode télécommande) : développement très rapide.

#### Inconvénients:

- Les scripts peuvent être un peu plus long à s'exécuter que du code en C ;
- On ne peut pas utiliser de bibliothèques spécifiques si elles ne sont pas implantées : traitement vidéo avancé non possible en script par exemple.

Vous rédigez un fichier texte avec les commandes à envoyer au robot.

Souvent le fichier de script Urbi possède l'extension .u, comme Urbi.

Avec la télécommande sur votre PC (fournie avec le kit de développement RobotCité), vous envoyez très simplement le fichier au robot qui l'exécute immédiatement.

Une fois votre script validé, vous le téléchargez définitivement dans le robot en le copiant dans sa mémoire. Puis vous spécifiez à l'engine de charger votre script.

Enfin vous relancez l'engine (ou redémarrez le robot) et votre fichier se charge avec les commandes et s'exécute.

Les commandes peuvent être:

- des ordres simples : "robot.avance(500mm);" : commande demandant au robot d'avancer de 500mm.
- des ordres plus complexes comme la réaction à un événement : "at(sensor.distance < 50mm) robot.stop()]" lorsque le capteur mesure une distance de moins de 50mm; le robot s'arrête.
- des ordres évolués : combinaisons d'ordres complexes mêlant des événements, des commandes programmées et des commandes calculées : approche de l'intelligence artificielle.

Différents scripts peuvent être chargés en parallèle pour exécuter des tâches bien plus complexes.

Des exemples de scripts évolués peuvent être téléchargés sur <http://www.urbiforge.org> pour vous donner une idée des possibilités.



## 5.4.2. Programmation C/C++

### **Avantages:**

- Plus de rapidité dans l'exécution du code par rapport aux scripts
- Aucune limitation dans la programmation, utilisation de bibliothèques déjà existantes possible

### **Inconvénients:**

- Il faut compiler le programme ce qui n'est pas toujours simple (différence d'architecture entre le robot et votre PC)

Pour qu'il soit compatible Urbi, votre programme devra avoir quelques instructions spécifiques expliquées dans la documentation.

Une couche réseau est automatiquement rajoutée à votre programme lors de la compilation.

Ensuite pour lancer votre programme il est nécessaire de lui préciser où est le robot (adresse IP) et votre programme se connecte sur le robot et s'exécute.

De cette manière vous pouvez compiler votre programme pour votre PC et l'exécuter sur votre PC. Le programme se connecte à l'adresse IP du robot et le robot prend en compte ce nouveau programme : c'est un UObject Remote.

Vous pouvez compiler votre programme pour le robot, le télécharger sur le robot, et préciser à l'engine de le lancer au démarrage avec l'adresse IP locale: 127.0.0.1. Comme cela le programme est exécuté sur le robot sans avoir besoin d'un ordinateur externe.

Ces méthodes permettent de développer simplement sur une machine puissante (votre PC) des programmes complexes, et de les tester sur la machine cible (le robot).

Toutes les possibilités du C/C++ sont disponibles.

## **5.5. Interface PC embarqué et électronique personnelles des équipes**

Il est bien sûr possible d'ajouter de l'électronique à la base roulante pour contrôler vos propres actionneurs et capteurs.

Votre électronique peut être totalement autonome ou elle peut communiquer avec la carte PC.

L'interface qui est conseillée entre votre électronique et la carte PC embarqué : Port série RS232. Le protocole des données transitant sur la liaison RS232 utilisée entre la carte PC et votre électronique est libre.



Les organisateurs RobotCité vous fourniront un UObject spécial appelé "RS232" qui permet une communication complète coté carte PC.

En rendant accessible votre électronique personnelle à la carte mère et aux données au format Urbi, vous pouvez par exemple, la contrôler par des scripts, sans passer par des fichiers en C/C++ compilé.

Un exemple pour plus de clarté:

### **Préparation:**

- Vous avez fait une carte électronique que vous avez branchée à la carte PC embarquée par le port RS232.
- Vous avez intégré le UObject RS232 que nous vous fournissons dans votre engine.
- Vous avez défini un protocole de communication entre la carte PC et votre électronique, par exemple quand le PC embarqué envoie "\_\_A+XX\_\_",
- Cela demande à votre électronique de placer le moteur numéro A à la position XX (protocole totalement arbitraire)

### **Utilisation:**

- Dans un fichier .u stocké sur le robot, saisir :

```
rs232.send("__1+24__");
```

- Sur la télécommande, sur votre PC, après vous être connecté au robot, tapez:

```
rs232.send("__1+24__");
```

- Dans un fichier en C/C++ compilé, sur le robot:

```
URBI((rs232.send("__1+24__");));
```

Ces 3 méthodes auront le même effet : mettre le moteur 1 à la position 24.

Cet exemple permet de voir tous les niveaux de programmation offerts aux équipes. A vous de choisir votre stratégie de programmation.

Les organisateurs ne fournissent aucun protocole de communication entre votre électronique personnelle (que nous ne connaissons pas), et la carte PC embarquée.

Nous vous fournissons uniquement le matériel logiciel permettant de développer la communication entre votre électronique et la carte PC embarquée.



---

## 5.6. Plus d'informations

Des documentations générales sur le langage et au format pdf se trouvent ici:

- URBI 1.5 quickstart : <http://www.gostai.com/doc/URBI-quickstart-1.5.pdf> (présentation succincte de tout ce qu'il est possible de faire avec ce langage)
- URBI 1.5 tutorial : <http://www.gostai.com/doc/en/urbi-tutorial-1.5.pdf> (description du langage de script Urbi)
- URBI spécifications 1.5 : <http://www.gostai.com/doc/URBI-Specif-1.5.pdf> (documentation complète du langage Urbi)
- UObject documentation <http://www.gostai.com/doc/en/uobject.pdf> (si vous désirez programmer en C/C++)

Le forum technique Urbi où de nombreuses informations ont déjà été postées: <http://forum.gostai.com> et où vous pouvez trouver des exemples.

Et bien sûr le Forum RobotCité <http://planete-sciences.org/forums>, section RobotCité, pour poser vos questions spécifiques.

## **6. CONSIGNES DE SECURITE**

Vous trouverez ci-après une liste de règles de sécurité. Cette liste n'est pas exhaustive – les organisateurs décideront de la dangerosité des systèmes.

En règle générale, vous devez élaborer des systèmes qui répondent à des critères de fabrication qui ne mettent en danger ni votre équipe, ni le public, aussi bien sur les stands que pendant l'évolution sur le Quartier RobotCité.

C'est aussi pourquoi vous devez vous assurer que vos systèmes respectent les lois applicables aux dispositifs techniques.

De façon générale, tout système estimé dangereux pour l'assistance ne sera pas autorisé à fonctionner dans l'enceinte de l'exposition. Dans ce cas, l'équipe devra apporter les modifications nécessaires pour présenter son robot en homologation.

### **6.1. Tension à bord**

Tout robot devra correspondre aux standards légaux concernant "la basse tension". Donc, la tension interne des robots ne doit pas excéder 48 V.

Afin d'éviter tout risque de feu, il est demandé de porter une attention particulière au choix des fils conducteurs, en fonction de l'intensité des courants les traversant.

Des tensions de plus de 48 V sont autorisées uniquement si elles ne sont présentes qu'à l'intérieur de dispositifs commercialisés en l'état, tels que lasers, rétro éclairage d'afficheurs LCD,... Ces dispositifs doivent être clos et ne pas avoir été modifiés.

### **6.2. Air comprimé**

Tout système sous pression devra respecter la loi en vigueur selon le "Conseil Général des Mines" (Conseil français de sécurité et d'approbation).

Rappel du décret 63 du 18 janvier 1943 et Ordre Ministériel du 25 juillet 1943 (législation française) :

- Pression maximale : 4 Bars
- [Pression Maximale] x [Volume de réservoir] → (80 bars) x [nombre de litre maximum]

Renseignements complémentaires sur :

<http://www.industrie.gouv.fr/sdsi/dgap/textes/1498-2.html>

### 6.3. Laser

Seules les notions de classes de laser seront prises en considération, et les équipes utilisant des lasers devront fournir la notice de classification des équipements concernés, ou la fiche technique (datasheet) du composant laser. L'incapacité de fournir ces documents empêchera le robot de pouvoir être homologué en l'état.

L'usage de sources laser est autorisé à condition de pouvoir justifier son appartenance à la classe 1. Un laser de classe 2 est toléré si le spot laser ne peut jamais croiser le visage dans son fonctionnement normal. Les lasers de classe 3 et 4 sont totalement interdits.

ATTENTION : Certain pointeur laser "bon marché" génère des puissances assimilées au laser de classe 3. C'est pourquoi il est demandé un justificatif conforme indiquant l'appartenance à une classe.

Pour raison de sécurité, l'utilisation de composants lasers extraits de lecteurs ou enregistreurs de CD/DVD n'est pas autorisée. En effet, même si l'appareil est affecté en classe 1, cette classification n'est valide qu'en étant utilisé sans modification et notamment avec son boîtier fermé. Les lasers inclus peuvent être de classe 3 et dans le spectre non visible (infrarouge).

ATTENTION : Démontez ce genre d'appareil et le faire fonctionner ouvert peut être très dangereux (ce qui est précisé en général sur les autocollants apposés sur le boîtier).

Les robots utilisant des lasers de classe 2 doivent présenter, de manière visible, une indication d'avertissement conformément à la réglementation 2003 EN 60825-1 et similaire à l'exemple ci-après :

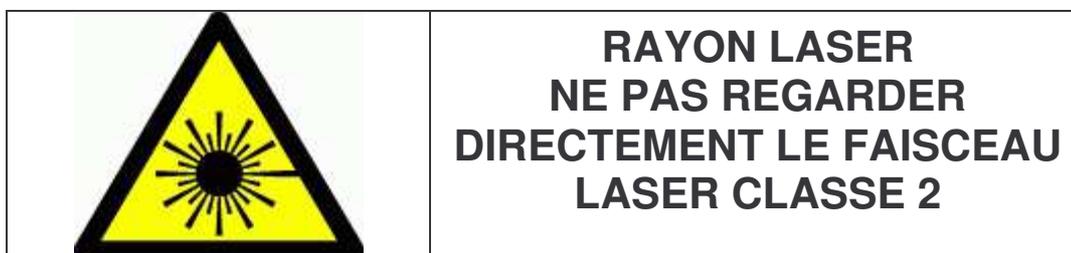


Figure 2 - Exemple d'autocollant indiquant la présence d'un système Laser

Les équipes ne se conformant pas à ces règles (par exemple en fournissant des documents ne correspondant pas aux dispositifs utilisés) seront tenues juridiquement responsables en cas de dommages causés par leur système.

### 6.4. Sources lumineuses de forte puissance

En cas d'utilisation d'une source lumineuse de forte intensité, l'intensité lumineuse ne doit pas être dangereuse pour l'œil humain en cas d'illumination directe. Notez que certains types de LED peuvent excéder cette limite.



## **7. DOCUMENTS A RENDRE ET ECHEANCES**

Afin de garantir un bon suivi de votre projet, nous vous demandons de nous fournir tout au long de cette période de travail, différents documents. Ceux-ci sont détaillés ci-dessous.

### **7.1. Le dossier d'avant-projet**

Au cours du mois de Février, les équipes devront remettre un avant-projet.

Il s'agit d'un questionnaire à remplir. Celui-ci vous sera transmis par e-mail et sera disponible sur le site [www.robotcite.fr](http://www.robotcite.fr).

### **7.2. Le workshop :**

Le workshop aura lieu le 7 Mars 2007 et permettra aux équipes de présenter leur projet face à un public.

L'équipe devra réaliser une présentation multimédia (diaporama ou animation) de qualité de 15 minutes. La présentation devra être compréhensible du grand public et présenter le principe du robot. Cette présentation contiendra un descriptif du projet et de l'étude.

### **7.3. Le compte-rendu**

Un compte-rendu devra être fourni au comité d'organisation par chaque équipe au courant du mois de Juin 2008. La date sera précisée ultérieurement aux équipes.

Ce document devra comprendre plusieurs chapitres abordant les points suivants :

- L'étude prospective de l'usage
- Le modèle économique, la cible et l'aspect commercial du produit
- La conception et les moyens mis en action pour sa création
- Le détail technique du robot
- Les pistes à travailler pour améliorer le produit

Ce document devra être fournis au format PDF.

## **7.4. Le Challenge**

### **7.4.1. Homologation**

Pour participer au Challenge RobotCité, un robot doit être obligatoirement homologué.

Le comité organisateur examine le robot et s'entretient avec l'équipe pour vérifier les points suivants:

- Le robot répond aux règles (le robot doit pouvoir faire état de toutes ses actions possibles et de ses configurations déployées pour aider à la vérification).
- L'équipe remet un dossier technique des systèmes utilisés par le robot (lasers,...)
- L'équipe fournit une affiche (voir 7.4.2ci-dessous)
- L'équipe comprend les règles et l'esprit de la compétition.
- Le bouton d'arrêt d'urgence fonctionne correctement et tous les actionneurs sont désactivés quand ce dernier est pressé.

### **7.4.2. L'affiche**

Chaque équipe est tenue de fournir une affiche technique de grande taille durant l'exposition. L'affiche est destinée à promouvoir l'échange et la communication entre les équipes et le public.

Un effort particulier devra être fait pour rendre l'affiche compréhensible par un auditoire non technique. Idéalement, l'affiche devra inclure des images et des diagrammes pour aider à expliquer les concepts.

Cette affiche présente les informations liées à la réalisation de votre projet : des illustrations, des renseignements techniques, des détails de conception, une description générale du projet,...

Elle doit être au minimum au format DIN A1 (594x841mm). Le format DIN A0 (841x1189mm) est recommandé. L'impression de ce poster est au frais des équipes participantes.

Le poster doit également faire apparaître:

- le nom de l'équipe
- le nom des membres de l'équipe

Ce poster sera affiché sur le stand de l'équipe sur les lieux du Challenge.

Le poster doit aussi être fourni au comité organisateur au format PDF. La résolution choisie doit garantir la lisibilité de tous les textes (72 dpi minimum – 300 dpi recommandé). Le fichier PDF résultant ne doit pas excéder 25MB.

De manière générale, le comité organisateur incite les équipes à communiquer autour de leur projet sur Internet, via les forums, au moyen de vidéos...



---

### 7.4.3. Les démonstrations publiques

Au cours de l'exposition chaque équipe devra présenter publiquement son projet.

Des démonstrations spécifiques à chaque robot auront lieu en parallèle sur la maquette.

Un effort particulier devra être fait pour rendre la présentation compréhensible par un auditoire non technique.

D'autres manifestations pourront potentiellement avoir lieu après l'exposition principale. La participation des équipes à ces autres manifestations est facultative.

Pour vos questions à propos de ce document,  
postez-les sur le forum :

<http://www.planete-sciences.org/forums>, section RobotCité



---

## Table des illustrations

Figure 1 - Architecture globale de la maquette RobotCité .....	9
Figure 2 - Exemple d'autocollant indiquant la présence d'un système Laser.....	19