

## I- Présentation

Vous disposez de deux robots Makeblock (robot pince, robot tank), de divers capteurs et d'une carte joystick.

### Le robot tank :

Il devra être autonome et suivre un trajet matérialisé au sol par une ligne noire. Il devra détecter automatiquement tout obstacle et s'arrêter tant que le chemin n'est pas dégagé.

### Le robot pince :

Il sera piloté par joystick, son rôle est de dégager les obstacles sur le chemin du robot tank



Bien sûr, la description du cahier des charges ci-dessus est largement évolutive et si vous avez des idées supplémentaires, libre à vous !

### Modules makeblock disponibles :

- Me Ligne Follower
- Me Sound Sensor
- Me 4 Button
- Me Ultrasonic Sensor
- Me Bluetooth

### Modules "maison" disponibles :

- Modules XBEE
- Carte joystick XBEE

## II- Description protocole XBEE

Toutes les 100 ms, la carte joystick XBEE transmet par onde la position du joystick (2 axes) et l'état des boutons poussoir. La trame de données est transmise dans l'ordre suivant (protocole) :

Grandeur	Rang	Format (en octet)
Octet début trame	1	0xAA → indique le début d'une trame
Position verticale du joystick	2	De 0 à 255 selon la position du joystick <ul style="list-style-type: none"> <li>• 0 → joystick en bas</li> <li>• 127 → joystick au centre</li> <li>• 255 → joystick en haut</li> </ul>
Position horizontale du joystick	3	De 0 à 255 selon la position du joystick <ul style="list-style-type: none"> <li>• 0 → joystick à gauche</li> <li>• 127 → joystick au centre</li> <li>• 255 → joystick à droite</li> </ul>
Etat du bouton poussoir BP1	4	0 → si BP1 relâché 255 → si BP1 appuyé
Etat du bouton poussoir BP2	5	0 → si BP2 relâché 255 → si BP2 appuyé

Pour recevoir les données XBEE avec la carte Orion, on pourra implémenter ce programme :

```
#include <SoftwareSerial.h>
```

```
SoftwareSerial softuart(13, 12); // Initialise la communication avec le module XBEE (port 3)
```

```
char Position_joystick_H, Position_joystick_V, BP1, BP2, Donnee_XBEE;
```

```
void setup()
```

```
{
  softuart.begin(9600);
}
```

```
// Programme principal //
```

```
void loop()
```

```
{
  if (softuart.available()) // Une donnée est-elle disponible sur le module XBEE?
  {
    reception_donnee_XBEE(); // Réception de toutes les données (position et boutons poussoir)
  }
}
```

```
void reception_donnee_XBEE()
{
  while ((Donnee_XBEE = softuart.read()) != 0xAA) {} // Synchronisation avec octet de début de trame 0xAA
  debut_trame = Donnee_XBEE;
  delay(1);
  // Réception des données images de la position du joystick
  Position_joystick_H = softuart.read();
  delay(1);
  Position_joystick_V = softuart.read();
  delay(1);
  // Réception des données images des états des boutons poussoir
  BP1 = softuart.read();
  delay(1);
  BP2 = softuart.read();
  delay(1);
}
```

### III- Installation de l'IDE Arduino version portable et de la librairie Makeblock

Vous allez installer la dernière version de l'IDE Arduino (Linux 32 bits) en version portable. Il vous sera donc possible de la stocker sur une clé USB pour l'exécuter depuis n'importe quel PC linux.

Remarque : On pourra désinstaller la version précédente d'Arduino

```
sudo apt remove arduino --purge
```

Il faut d'abord la télécharger à l'adresse :

<https://www.arduino.cc/en/software>

Décompresser les paquets puis ouvrez un terminal dans le répertoire **arduino-1.8.13**. Pour rendre cette version "portable", il suffit de créer un nouveau répertoire vide appelé **portable** puis de lancer l'IDE arduino :

```
cd arduino-1.8.13
```

```
mkdir portable
```

```
./arduino
```

Fermer l'IDE, vous pouvez vérifier que le dossier portable n'est plus vide...

### Downloads

Arduino IDE 1.8.13

The open-source Arduino Software (IDE) makes it easy to write code and upload it to the board. This software can be used with any Arduino board.

Refer to the [Getting Started](#) page for Installation instructions.

SOURCE CODE

Active development of the Arduino software is [hosted by GitHub](#). See the instructions for [building the code](#). Latest release source code archives are available [here](#). The archives are PGP-signed so they can be verified using [this](#) gpg key.

DOWNLOAD OPTIONS

- Windows Win 7 and newer
- Windows ZIP file
- Windows app Win 8.1 or 10 [Get](#)
- Linux 32 bits**
- Linux 64 bits
- Linux ARM 32 bits
- Linux ARM 64 bits
- Mac OS X 10.10 or newer

Release Notes

Checksums (sha512)

Une librairie « Makeblock » est disponible et permet de gérer la carte "orion" et tous les modules makeblock. Pour l'installer, ouvrir un terminal :

```
cd arduino-1.8.13/portable/sketchbook/libraries
```

```
git clone https://github.com/Makeblock-official/Makeblock-Libraries.git
```

Le dossier **Makeblock-Libraries-master** doit être présent dans le dossier **libraries**. Il suffit de redémarrer le logiciel Arduino pour la librairie Makeblock soit effective.

Dans le répertoire **Makeblock-Libraries-master**, un dossier **examples** contient (comme son nom l'indique) de nombreux exemples... Bien entendu, il est fortement conseillé de s'en inspirer !

N'oubliez pas non plus les exemples disponibles dans le répertoire **arduino-1.8.13/examples**.