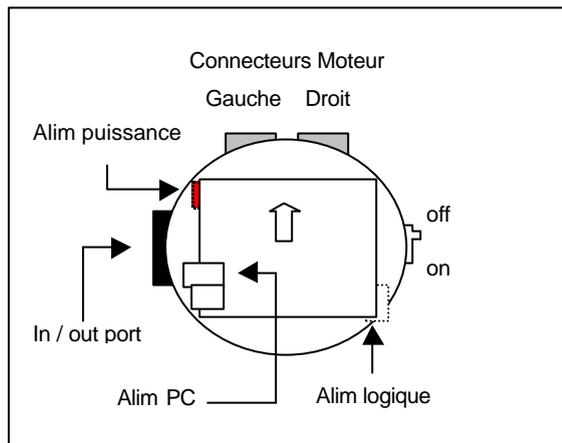


## Ce document contient :

- Alimentation
- Câblage
- Recharge des accus
- Interface logiciel
- Gestion des moteurs
- Gestion des infra-rouges
- I/O optionnelles
- Notes

## Alimentation



Connectique du robot

Bien que les connecteurs soient tous montés avec des détrompeurs, il est prudent de vérifier le sens de l'alimentation de puissance.

	Tension	Courant
Alim puissance	6 V	500 mA
Alim logique	7 – 16 V	1200 mA
Alim PC	N'utiliser que des alims AT	

Tableau des alimentations

### Fonctionnement autonome

Pour le fonctionnement sur accus : brancher les accus sur le robot, vérifier que l'interrupteur principal est bien sur OFF. ATTENTION à respecter la polarisation des accus. Mettre l'interrupteur sur ON, en maintenant le robot surélevé. Il est fréquent que les roues tournent pendant le boot.

### NE RIEN BRANCHER SUR ALIM PC

### Fonctionnement sur alimentation

Il est préférable d'utiliser des alimentations à courant limitées. Limiter le courant aux valeurs du tableau ci-dessus. Brancher le robot en

vérifiant que l'interrupteur est en position OFF. L'allumer en le maintenant surélevé.

### NE RIEN BRANCHER SUR ALIM PC

### Fonctionnement sur alim AT

Ce mode d'utilisation est le seul permettant l'utilisation du connecteur Alim PC. Mettre l'interrupteur sur off. Branchez l'alimentation AT sur le connecteur Alim PC. Brancher un accu ou une alimentation stabilisée sur Alim puissance. Allumer l'alimentation, en conservant l'interrupteur du robot sur off.

### NE RIEN BRANCHER SUR ALIM LOGIQUE

### Utilisation d'un lecteur de disquettes

Pour utiliser un lecteur de disquettes la connectique d'alimentation est la même que pour le fonctionnement sur Alim AT. Avant de mettre sous tension, brancher le lecteur de disquettes sur l'Alim, et connecter le câble en nappe sur le robot. Mettre sous tension.

### NE RIEN BRANCHER SUR ALIM LOGIQUE

En aucun cas, vous ne devez utiliser un autre câblage que ceux indiqués ci-dessus. Ne jamais débrancher de câbles, la Ram ou la Compact Flash sous tension. Seuls le clavier et l'écran peuvent être débranchés sous tension, mais il est préférable de limiter cette manipulation afin d'augmenter la durée de vie des câbles.

## Recharge des accus

Pour les accus, utiliser uniquement des accus NiMh aux caractéristiques suivantes :

	Tension	Charge
Accu puissance	6 V	1300 mAH
Accu logique	12 V	1300 mAH

Tableau des accus

Il y a deux façon de recharger les accus :

- En fixant le temps
- En fixant la charge

Il vaut mieux utiliser un chargeur d'accus plutôt qu'une alimentation stabilisée. Donc, il vaut mieux les charger en fixant la charge (celle du chargeur). Nous poserons les variables suivantes :

- Tc : Temps de charge (Heures)
- Ca : Charge des accus (mAH)
- Cc : Courant de charge (mA)

## En fixant le temps

$$C_c = \frac{1.4 \times C_a}{T}$$

Par exemple, nous voulons charger un accus 1300 mAH en 5 heures :

T = 5 heures

Ca = 1300 mA

On en déduit un courant de charge de

$$C_c = \frac{1.4 \times 1300}{5} = 364mA$$

## En fixant la charge

$$T = \frac{1.4 \times C_a}{C_c}$$

Par exemple, nous voulons charger un accus 1300 mAH à 180mA :

Cc = 180 mA

Ca = 1300 mA

On en déduit un temps de charge de

$$T = \frac{1.4 \times 1300}{180} \approx 10H$$

Pour améliorer la durée de vie des accus, l'idéal est de fixer le temps de charge des accus à 14 heures et de les charger à 1/10 de leur capacité. Il vaut mieux éviter de les charger plus rapidement, même si cela est possible occasionnellement.

## Tableau récapitulatif

Si vous utiliser un chargeur A2 Pro, les charges sont fixes. Donc, il faut recharger les accus en respectant le tableau suivant :

Charge	Temps	Remarque
50 mA	36 Heures	A éviter
60 mA	30 Heures	A éviter
80 mA	22 Heures	A éviter
140 mA	13 Heures	Ok
180 mA	10 Heures	Ok
500 mA	4 Heures	Occasionnellement

## Interface logiciel

L'ensemble des actionneurs et capteurs est connecté au bus ISA. La mise à jour des données se fait par deux valeurs :

- L'adresse
- Les données

Chaque capteur / actionneur possède une adresse. Pour activer un actionneur en C, on utilise l'instruction « outp »

outp (Adresse, Valeur) ;

Pour lire un capteur, on utilise l'instruction « inp »

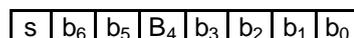
Valeur=inp (Adresse) ;

## Gestion des moteurs

Pour chaque moteur il faut distinguer la commande et l'information des codeurs.

### La commande

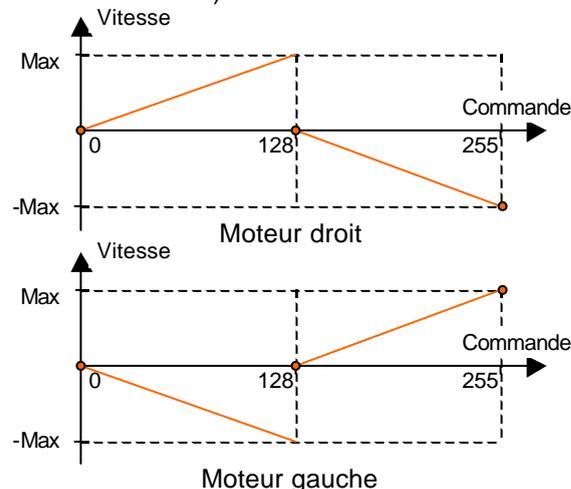
La commande se fait sur 8 bits, dont un de sens :



Adresse du moteur droit : 0x103

Adresse du moteur gauche : 0x102

A cause du bit de sens et de la symétrie du robot, les moteurs se commandent dans le sens inverse l'un de l'autre. Le moteur droit tourne en avant pour les valeurs de 0 à 127 (127 vitesse maximale) et en arrière de 128 à 255 (255 vitesse maximale). Le moteur gauche tourne en avant de 128 à 255 (255 vitesse maximale) et en arrière de 0 à 127 (127 vitesse maximale).

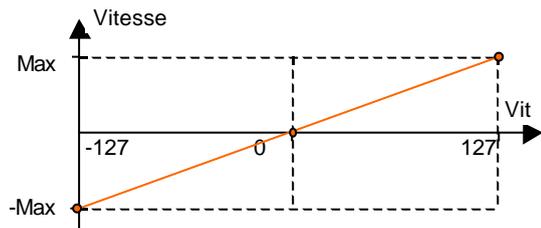


Voici deux routines qui permettent de commander les moteurs sans se soucier de ces problèmes d'inversion :

```
void Mdroit (int Vit)
{
    if (Vit>0)
        outp(0x102,Vit);
    else
        outp(0x102,128-Vit);
}

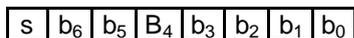
void Mgauche (int Vit)
{
    if (vitesse>0)
        outp(0x103,128+Vit);
    else
        outp(0x103,-Vit);
}
```

Dans ces deux routine, le paramètre vitesse est compris entre -127 et 127. La vitesse en fonction de la commande est représenté sur la courbe suivante :



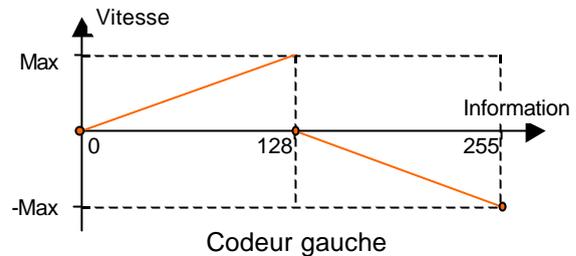
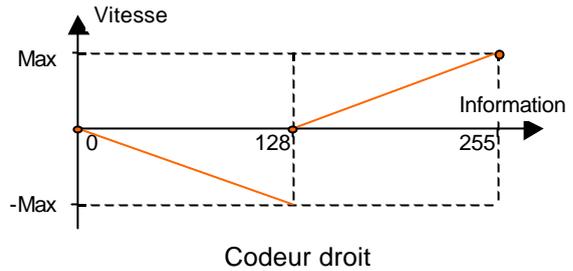
### Les codeurs

Les moteurs sont équipés de codeurs magnétiques 16 impulsions par tour avec 2 canaux. Les moteurs sont montés sur des réducteurs 1/21, ce qui donne 336 impulsions par tour de roue. Les deux canaux nous permettent de connaître le sens de rotation du moteur. L'information des codeurs est codée de la façon suivante :



Adresse du codeur droit : 0x104  
 Adresse du codeur gauche : 0x105

Comme pour la commande, les informations des codeurs droit et gauche sont inversées.



Les deux routines suivantes permettent de récupérer les données des codeurs sans tenir compte de l'inversion :

```
int Cgauche ()
{
    int val;
    val=inp (0x105);
    if (val>=128) return (-val+128);
    return (val);
}

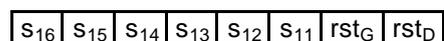
int Cdroit ()
{
    int val;
    val=inp (0x104);
    if (val>=128) return (val-128);
    return (-val);
}
```

### Le reset codeurs

Il est possible de remettre les codeurs à 0. Si l'information codeur venait à dépasser 127 ou -127, le compteur se remet automatiquement à 0. Malgré cela, pour éviter de perdre des informations, il est préférable de remettre les codeurs à 0 avant le dépassement.

Adresse du reset : 0x106

Le port 0x106 est utilisé pour le reset des codeurs et aussi pour les sorties optionnelles :



rst <sub>G</sub>	Codeur gauche	rst <sub>D</sub>	Codeur droit
1	0	1	0
0	Comptage autorisé	0	Comptage autorisé

La routine `rst_codeur ()` remet les deux codeurs à 0.

```
void rst_codeur ()
{
    outp (0x106,3);
    outp (0x106,0);
}
```

La routine `rst_codeur_droit ()` remet le codeur droit à zero.

```
void rst_codeur_droit ()
{
    outp (0x106,1);
    outp (0x106,0);
}
```

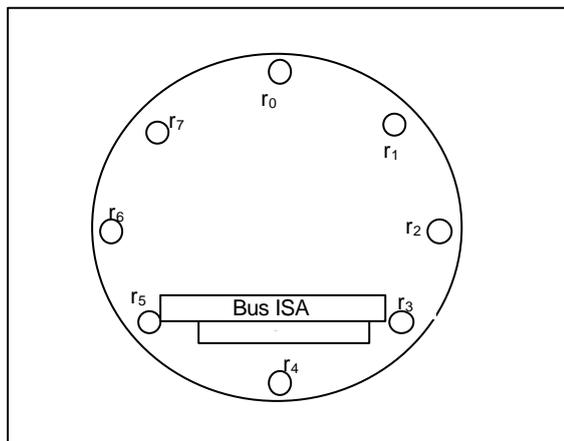
La routine `rst_codeur_gauche ()` remet le codeur gauche à 0.

```
void rst_codeur_gauche ()
{
    outp (0x106,2);
    outp (0x106,0);
}
```

### CES ROUTINES METTENT A ZERO LES SORTIES OPTIONNELLES

## Gestion des infra-rouge

Il faut ici distinguer l'émission et la réception des infra-rouges. Le robot est équipé de 16 émetteurs et de 8 récepteurs disposés de la façon suivante :



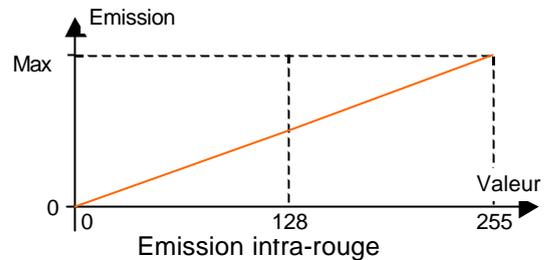
Disposition des récepteurs infra-rouge

Les 16 émetteurs s'allument en même temps, il n'est possible de les dissocier. Ils créent une ceinture infra-rouge autour du robot. En

revanche, il est possible de choisir la portée des infra-rouge, donc le diamètre de la ceinture.

### Emission infra-rouge

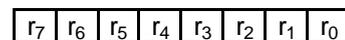
L'émission se fait par l'intermédiaire de l'adresse 0x101.



La longueur d'émission est proportionnelle à la valeur envoyée sur l'adresse 0x101. Si la valeur est égale à 0, il n'y a

### Réception infra-rouge

La réception se fait par l'intermédiaire de l'adresse 0x100. La réception se fait en tout ou rien, chaque bit représente un capteur.



La valeur 255 signifie qu'aucun des 8 capteurs n'a de réception, et la valeur 0, que tous les capteurs reçoivent un signal.

r <sub>x</sub>	Etat
0	Réception
1	Pas de réception

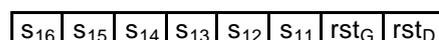
## I/O Optionnelles

Le robot est équipé d'un port entrées / sorties. C'est le connecteur HE10 situé au dessus de la roue gauche. Il comporte 6 sorties (0x106) et 8 entrées (0x107).

S <sub>15</sub>	S <sub>13</sub>	S <sub>11</sub>	e <sub>9</sub>	e <sub>7</sub>	e <sub>5</sub>	e <sub>3</sub>	Gnd
S <sub>16</sub>	S <sub>14</sub>	S <sub>12</sub>	e <sub>10</sub>	e <sub>8</sub>	e <sub>6</sub>	e <sub>4</sub>	VCC

Schéma du connecteur vue de face

Adresse 0x106 :



### LES DEUX PREMIERS BITS CORRESPONDENT AU RESET CODEURS

