

Hemisson



Hemgenio



VERSION 1.4
DÉCEMBRE 2003

Auteur de la documentation

Alexandre Colot, K-Team S.A.
Ch. de Vuasset, CP 111
1028 Préverenges
Suisse

email : info@hemisson.com

Url : www.hemisson.com

MARQUE DÉPOSÉE :
Hemisson : K-TEAM SA.
SysQuake LE : Calerga SA

INDICATIONS :

- Le contenu de ce manuel est susceptible de changer sans préavis.
- Tous les efforts ont été fait afin d'assurer l'exactitude de ce manuel.
Si toutefois vous trouvez une erreur, merci d'en informer K-Team S.A.
- Malgré les commentaires ci-dessus, K-Team S.A. ne pourra être tenu pour responsable des conséquences liées à une erreur dans ce manuel.



1	Introduction	4
1.1	Comment utiliser ce manuel	4
1.2	Précautions d'emploi	4
1.3	Recyclage	4
2	Connexion	5
2.1	Assemblage	5
2.2	Désassemblage	6
3	Le module HemGenIO	7
3.1	Vue d'ensemble	7
3.2	Adresse I2C	8
3.3	Adresses des registres	8
3.3.1	Registre Firmware Version	8
3.3.2	Registre entrée analogique A0	8
3.3.3	Registre entrée analogique A1	8
3.3.4	Registre entrée analogique A2	9
3.3.5	Registre entrée analogique A3	9
3.3.6	Registre entrée analogique A4	9
3.3.7	Registre entrées analogiques A0 à A4	9
3.3.8	Registre I/O digitale D0	9
3.3.9	Registre I/O digitale D1	9
3.3.10	Registre I/O digitale D2	9
3.3.11	Registre I/O digitale D3	11
3.3.12	Registre I/O digitale D4	11
3.3.13	Registre I/O digitale D5	11
3.3.14	Registre I/O digitale D6	11
3.3.15	Registre I/O digitale D7	11
3.3.16	Registre I/O digitale D8	11
3.3.17	Registre I/O digitale D9	12
3.3.18	Registre I/O digitale D10	12
3.3.19	Registre I/O digitale D11	12
3.3.20	Changement d'adresse I2C	12
3.4	Electronique	13

4	Utilisation	14
4.1	Programmation en C	14
4.1.1	char HemGenIO_Init(void)	14
4.1.2	char HemGenIO_Read_Version(void)	14
4.1.3	int1 HemGenIO_Read_Digital(char input)	14
4.1.4	void HemGenIO_Write_Digital(char input, int1 state)	15
4.1.5	char HemGenIO_Read_Analog(char input)	15
4.2	Contrôle par port série	15
A	Spécifications techniques	16
B	Connecteurs	17



Le module HemGenIO permet à l'utilisateur de développer ses propres modules d'extension en intégrant ses composants à l'aide des 12 entrées/sorties digitales ou encore des 5 entrées analogiques 8 bits. Une large zone est percée au pas 2,54 mm pour venir y souder ses propres composants.

1.1 Comment utiliser ce manuel

Ce manuel introduit le module HemGenIO destiné au robot Hemisson. Si ce manuel ne répond pas à l'un des problèmes auquel vous êtes confronté, consultez le site web Hemisson (www.hemisson.com) et plus particulièrement le Forum ou la section FAQs¹.

1.2 Précautions d'emploi

Voici quelques recommandations pour faire bon usage du module HemGenIO :

- **Eloignez-le des endroits humides.**
Un contact avec de l'eau pourrait provoquer un court-circuit.
- **Entreposez-le de façon stable.**
Cela évitera les risques de chutes qui pourraient le casser ou causer des dommages à un tiers.
- **Ne branchez pas le module pendant que le robot est allumé.**
Pour éviter tout dommage, effectuez toutes les connexions lorsque le robot est éteint.
- **Ne laissez jamais Hemisson allumé lorsqu'il est inutilisé.**
Lorsque vous avez fini de travailler avec Hemisson, éteignez le.
- **Vérifiez votre montage avant de le mettre sous tension**
Si votre montage comporte des erreurs comme par exemple des courts-circuits, vous risquez d'endommager vos composants, Hemisson et sa batterie.

1.3 Recyclage

Pensez à la fin de vie de ce matériel!

Certaines parties peuvent être recyclées, par exemple ne jetez pas votre pile à la poubelle mais portez-la plutôt dans des containers adaptés. En la recyclant, vous contribuez à offrir un environnement plus propre et plus sain pour les générations futures. Pour toutes ces raisons, faites attention à la fin de vie de ce matériel, en le retournant par exemple au fabricant ou à votre revendeur local.

¹Foire Aux Questions



L'assemblage et le désassemblage du module d'extension est une opération délicate. Veuillez lire avec attention les instructions ci-dessous afin d'éviter de faire subir des dommages à celui-ci. K-TEAM n'assumera aucune responsabilité pour les dommages causés par une mauvaise manipulation.

2.1 Assemblage

Commencez par vous assurer que le robot est éteint (position Off) puis insérez HemGenIO le plus verticalement possible en respectant l'orientation ci-dessous :

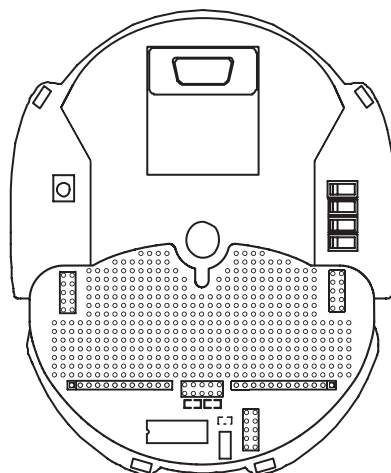


FIG. 2.1 – Module HemGenIO sur Hemisson vu de dessus

Ne tentez surtout pas d'insérer HemGenIO d'une autre façon ; cela pourrait causer de graves dommages à votre robot Hemisson ainsi qu'au module d'extension.

2.2 D sassemblage

Pour le d sassemblage, veuillez tout d'abord   ce que votre robot soit  teint (position Off), puis en le tenant d'une main, tirez le plus verticalement possible le module HemGenIO avec l'autre main en prenant bien garde de ne pas tordre ni casser les pins des connecteurs.



3.1 Vue d'ensemble

Le module HemGenIO se compose comme suit :

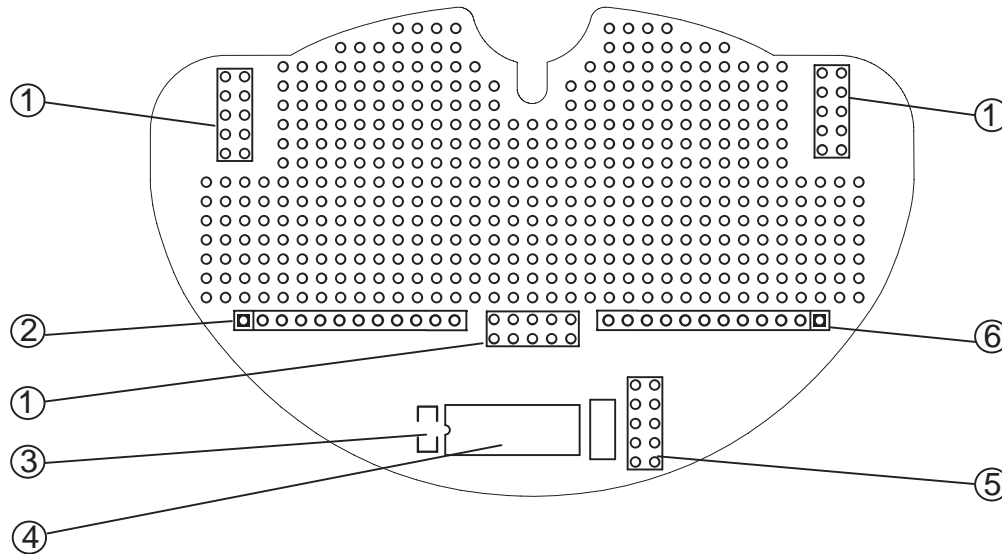


FIG. 3.1 – Détails du module HemGenIO : vue de dessus

1. Connecteurs d'extension Hemisson
2. Connecteur I/Os digitales
3. LED
4. Processeur
5. Connecteur de reprogrammation du module
6. Connecteur d'alimentations, d'entrées analogiques et d'I2C

Pour plus d'informations sur les détails des connecteurs, veuillez consulter l'Annexe à la fin de ce manuel.

3.2 Adresse I2C

Le robot Hemisson accède au module HemLinCam par le biais d'une communication I2C. L'adresse de ce module est définie sur 7 bits + 1 bit pour la sélection du mode "lecture" ou "écriture".



FIG. 3.2 – Adresse I2C du module

Ainsi, l'adresse en mode "lecture" est **0xD1** ; et **0xD0** en mode "écriture".

3.3 Adresses des registres

Pour obtenir les valeurs des entrées ou encore définir l'état d'une sortie, il s'agit d'accès "lecture" et "écriture" à des registres dont la liste se trouve ci-dessous.

Tout au long de la section suivante nous utiliserons la notation :

- *R* : pour un registre accessible uniquement en mode "Lecture".
- *W* : pour un registre accessible uniquement en mode "Ecriture".
- *R&W* : pour un registre accessible en mode "Lecture" et "Ecriture".

3.3.1 Registre Firmware Version

Description : permet de lire la version du firmware présent dans la mémoire du processeur.

Adresse : 0 (0x00)

Accès : R

3.3.2 Registre entrée analogique A0

Description : permet d'obtenir la valeur 8 Bits de l'entrée analogique A0.

Adresse : 16 (0x10)

Accès : R

3.3.3 Registre entrée analogique A1

Description : permet d'obtenir la valeur 8 Bits de l'entrée analogique A1.

Adresse : 17 (0x11)

Accès : R

3.3.4 Registre entrée analogique A2

Description : permet d'obtenir la valeur 8 Bits de l'entrée analogique A2.

Adresse : 18 (0x12)

Accès : R

3.3.5 Registre entrée analogique A3

Description : permet d'obtenir la valeur 8 Bits de l'entrée analogique A3.

Adresse : 19 (0x13)

Accès : R

3.3.6 Registre entrée analogique A4

Description : permet d'obtenir la valeur 8 Bits de l'entrée analogique A4.

Adresse : 20 (0x14)

Accès : R

3.3.7 Registre entrées analogiques A0 à A4

Description : permet d'obtenir les valeurs 8 Bits de toutes les entrées analogiques (dans l'ordre A0 à A4).

Adresse : 21 (0x15)

Accès : R

3.3.8 Registre I/O digitale D0

Description : lors d'un accès en écriture cette pin est définie en tant que sortie, tandis qu'un accès en lecture la configurera en entrée.

Adresse : 32 (0x20)

Accès : R&W

3.3.9 Registre I/O digitale D1

Description : lors d'un accès en écriture cette pin est définie en tant que sortie, tandis qu'un accès en lecture la configurera en entrée.

Adresse : 33 (0x21)

Accès : R&W

3.3.10 Registre I/O digitale D2

Description : lors d'un accès en écriture cette pin est définie en tant que sortie, tandis qu'un accès en lecture la configurera en entrée.

Adresse : 34 (0x22)

Accès : R&W

3.3.11 Registre I/O digitale D3

Description : lors d'un accès en écriture cette pin est définie en tant que sortie, tandis qu'un accès en lecture la configurera en entrée.

Adresse : 35 (0x23)

Accès : R&W

3.3.12 Registre I/O digitale D4

Description : lors d'un accès en écriture cette pin est définie en tant que sortie, tandis qu'un accès en lecture la configurera en entrée.

Adresse : 36 (0x24)

Accès : R&W

3.3.13 Registre I/O digitale D5

Description : lors d'un accès en écriture cette pin est définie en tant que sortie, tandis qu'un accès en lecture la configurera en entrée.

Adresse : 37 (0x25)

Accès : R&W

3.3.14 Registre I/O digitale D6

Description : lors d'un accès en écriture cette pin est définie en tant que sortie, tandis qu'un accès en lecture la configurera en entrée.

Adresse : 38 (0x26)

Accès : R&W

3.3.15 Registre I/O digitale D7

Description : lors d'un accès en écriture cette pin est définie en tant que sortie, tandis qu'un accès en lecture la configurera en entrée.

Adresse : 39 (0x27)

Accès : R&W

3.3.16 Registre I/O digitale D8

Description : lors d'un accès en écriture cette pin est définie en tant que sortie, tandis qu'un accès en lecture la configurera en entrée.

Adresse : 40 (0x28)

Accès : R&W

3.3.17 Registre I/O digitale D9

Description : lors d'un accès en écriture cette pin est définie en tant que sortie, tandis qu'un accès en lecture la configurera en entrée.

Adresse : 41 (0x29)

Accès : R&W

3.3.18 Registre I/O digitale D10

Description : lors d'un accès en écriture cette pin est définie en tant que sortie, tandis qu'un accès en lecture la configurera en entrée.

Adresse : 48 (0x30)

Accès : R&W

3.3.19 Registre I/O digitale D11

Description : lors d'un accès en écriture cette pin est définie en tant que sortie, tandis qu'un accès en lecture la configurera en entrée.

Adresse : 49 (0x31)

Accès : R&W

3.3.20 Changement d'adresse I2C

Description : ce registre permet de changer l'adresse I2C du module. Cette nouvelle adresse est stockée en EEPROM ainsi il n'est pas nécessaire de lui redonner la nouvelle adresse à chaque redémarrage. Il faut lui spécifier une nouvelle adresse en mode écriture, donc une adresse paire (par exemple 0xA0). Si vous entrez une adresse éronnée (par exemple 0x51), le module retournera automatiquement à son adresse de base, c'est à dire 0xD0.

Adresse : 96 (0x60)

Accès : W

3.4 Electronique

Il est très important de noter que sur toutes les entrées/sorties qu'elles soient analogiques ou numériques, il y a une pull-down connectée de 100 KOhms. Ceci afin de ne pas obtenir des valeurs éronnées lorsque l'on lit une entrée (analogique ou digitale) qui n'est reliée à rien du tout.

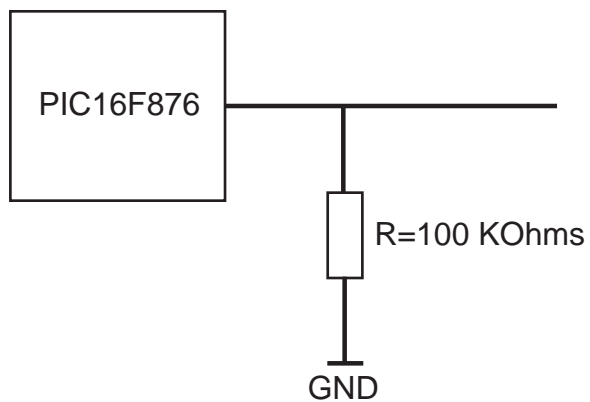


FIG. 3.3 – Pull-Down



Il y a deux manières d'utiliser HemLinCam avec HemiOs :

- Programmation en C
- Contrôle par port série (exemple de SysQuake LE)

4.1 Programmation en C

Pour programmer HemiOs en C, vous devez avoir auparavant fait l'acquisition du compilateur CCS (disponible sur le site web HemiOs dans la section logiciels). Nous vous conseillons tout d'abord de lire la dernière version du manuel de l'HemiOs (<http://www.hemios.com/French/assistance.html>), dans cette doc vous trouverez toutes les informations nécessaires sur l'HemiOs ainsi que l'installation de CCS.

- Téléchargez ensuite (<http://www.hemios.com/French/assistance.html>) la dernière version disponible de l'HemiOs, ainsi que la dernière version de HemGenIO Lib (contenant HemGenIO.h).
- Ouvrez le projet avec CCS

La librairie HemGenIO.h intègre les fonctions suivantes :

4.1.1 char HemGenIO_Init(void)

But : Initialisation du module HemGenIO, c'est la première fonction à appeler tout au début du code.

Exemple :

```
HemGenIO_Init();
```

4.1.2 char HemGenIO_Read_Version(void)

But : Lire la version du firmware.

Exemple :

```
char Version;  
Version = HemGenIO_Read_Version();
```

4.1.3 int1 HemGenIO_Read_Digital(char input)

But : Lire l'état (1 ou 0) d'une entrée digitale, le paramètre input permet de sélectionner l'entrée (0 à 11).

Exemple :

```
int1 StateIODig1;  
StateIODig1 = HemGenIO_Read_Digital(1);
```

4.1.4 void HemGenIO_Write_Digital(char input, int1 state)

But : Définir l'état d'une sortie. Le premier paramètre concerne la sélection de la sortie (0 à 11), tandis que le deuxième paramètre définit son état (0 ou 1)

Exemple :

```
HemGenIO_Write_Digital(9,0); // Mettre D9 à 0 (GND)
```

4.1.5 char HemGenIO_Read_Analog(char input)

Pour lire la valeur analogique 8 Bits (0 = 0V , 255 = 5V) d'une entrée analogique (0 à 4) newline Exemple :

```
char Analog2Value;  
Analog2Value = HemGenIO_Read_Analog(2);
```

Rappel : pour utiliser ces fonctions, vous devez avoir préalablement fait un include de la librairie au début de votre code : #include "HemGenIO.h" (il faut bien entendu que le fichier soit placé dans le même répertoire.)

4.2 Contrôle par port série

A partir de la version 1.30 du Firmware d'Hemisson disponible sur le site web (<http://www.hemisson.com/French/assistance.html>) vous pouvez accéder au modules I2C par le biais de commande RS232. Pour cela référez-vous à la dernière version du manuel utilisateur Hemisson.

Exemples :

- W,D0,26,01 > : sortie digitale D6 est mise à 1 (5V).
- R,D0,00,01 > : renvoi la version du firmware.

Cette propriété d'accès par port série est exploitée dans le cas de SysQuake LE qui est un logiciel freeware distribué par Calerga (<http://www.calerga.com>), il permet d'envoyer et recevoir des caractères ascii via le port série et dispose également d'une interface graphique.

Vous trouverez sur le site Hemisson (<http://www.hemisson.com/French/assistance.html>) une interface SysQuake (.sq) pour ce module HemGenIO. Elle vous permettra par exemple de tester votre module (pour cela le module doit être pluggé sur Hemisson et le robot doit être connecté au PC via un port série). Le code de cette interface étant en open-source (sous licence GPL), vous pouvez vous en servir d'exemple pour développer vos propres interfaces de contrôle pour votre robot.

A SPÉCIFICATIONS TECHNIQUES



- Poids [g] : NC
- Tension [V] : 5
- Freq. Max. I2C [KHz] : 400
- Nombre d'entrées analogiques : 5
- Convertisseur AD [Bits] : 8
- Nombre d'entrées/sorties digitales : 12
- Courant maximum d'une sortie digitale [mA] : 25

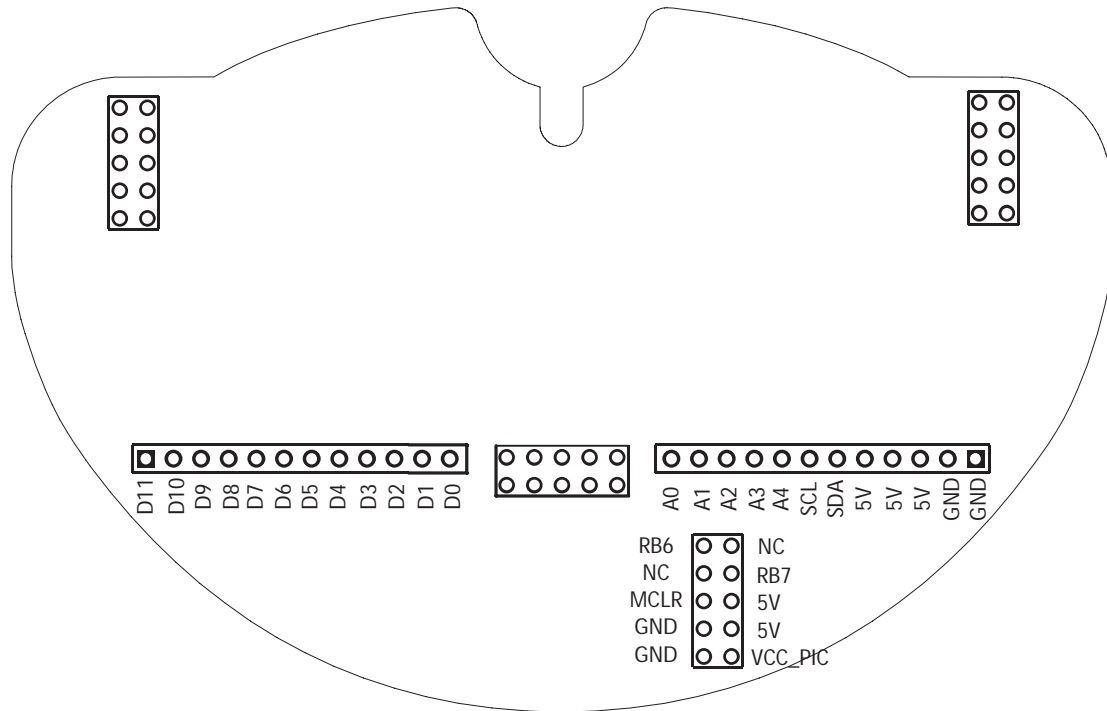


FIG. B.1 – Détails des connecteurs

- D0-D11 : entrées/sorties digitales
- A0-A4 : entrées analogiques
- SCL,SDA : signaux I2C
- 5V : alimentation 5V
- GND : masse
- RB6,RB7,MCLR,GND,VCC_PIC : signaux de reprogrammation du processeur (voir modules HemFlexExtProg et HemIcdAdaptor)



K-Team SA
1028 Préverenges
CH DE VUASSET, CP 111
SWITZERLAND
