

Author

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

email: info@hemisson.com

Url: www.hemisson.com

TRADEMARK:

Hemisson: K-TEAM SA.

SysQuake LE: Calerga SA

INDICATIONS:

- The contents of this manual are subject to change without notice.
- All efforts have been made to ensure the accuracy of this manual.
However, should any error be detected, please inform K-TEAM S.A.
- The above notwithstanding, K-TEAM S.A. can assume no responsibility for any error in this manual.

TABLE OF CONTENTS



1	Introduction	4
1.1	How to use this manual	4
1.2	Recommendations	4
1.3	Recycling	4
2	Connexion	5
2.1	Install	5
2.2	Uninstall	6
3	The HemGenIO module	7
3.1	Overview	7
3.2	I2C Address	8
3.3	Register Addresses	8
3.3.1	Firmware Version Register	8
3.3.2	A0 Analog Input Register	8
3.3.3	A1 Analog Input Register	8
3.3.4	A2 Analog Input Register	9
3.3.5	A3 Analog Input Register	9
3.3.6	A4 Analog Input Register	9
3.3.7	A0-A4 Analog Input Register	9
3.3.8	D0 Digital I/O Register	9
3.3.9	D1 Digital I/O Register	9
3.3.10	D2 Digital I/O Register	9
3.3.11	D3 Digital I/O Register	11
3.3.12	D4 Digital I/O Register	11
3.3.13	D5 Digital I/O Register	11
3.3.14	D6 Digital I/O Register	11
3.3.15	D7 Digital I/O Register	11
3.3.16	D8 Digital I/O Register	11
3.3.17	D9 Digital I/O Register	12
3.3.18	D10 Digital I/O Register	12
3.3.19	D11 Digital I/O Register	12
3.3.20	I2C Address Change	12
3.4	Electronics	13

4	Use	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)	15
4.1.4	void HemGenIO_Write_Digital(char input, int1 state)	15
4.1.5	char HemGenIO_Read_Analog(char input)	15
4.2	Serial Port Control	15
A	Technical Specifications	16
B	Connectors	17

1 INTRODUCTION



The HemGenIO module allows to develop your own extension by connecting external components to the 12 digital input/output or to the 5 8-bit analog inputs. On HemGenIO, a large area is prebored with a 0.1" spacing in order to solder or wrap your own component.

1.1 How to use this manual

This manual introduces the HemGenIO module dedicated to Hemisson robot. If the manual do not answer to a question, please consult Hemisson website (www.hemisson.com) and in particular the Forum and the FAQs¹.

1.2 Recommendations

Here are a few recommendations:

- **Keep it away from wet areas.**
Contact with water could cause malfunction and/or breakdown.
- **Store it in a stable position.**
This will avoid the risks of falls, which could break it or cause damage to a person.
- **Do not plug or unplug it while the robot is powered on.**
To avoid any damage, make all connections when the robot power is off.
- **Never leave Hemisson powered when it is unuse.**
When you are finished working with Hemisson, turn it off. It will save battery life.
- **Check your electronics assembly before the robot is powered on**
If your assembly is false or if there are short-cut, you may risk to damage your components, your robot and your battery.

1.3 Recycling

Think to your material end of life of your material !
Most parts can be recycled, so please bring used parts in appropriate containers or return it to the manufacturer or to your local dealer. By recycling you contribute to a cleaner and healthier environment for the future generations.

¹Foire Aux Questions



Installing and uninstalling the extension module is delicate. Please read carefully instructions below to avoid damages. K-TEAM will not take in charge damages caused by mismanipulation.

2.1 Install

Start by checking that robot is switched Off, then insert HemGenIO as much vertical as possible and according to the orientation below:

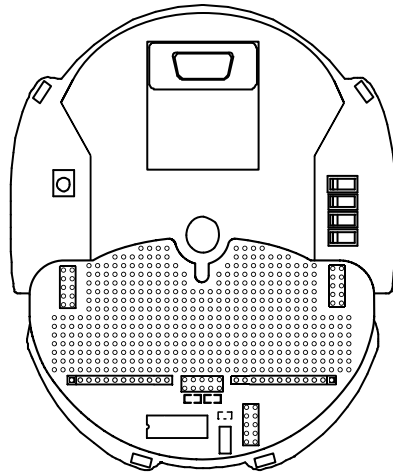


Figure 2.1: HemGenIO Module Top-View

Please do not attempt to insert HemGenIO in a different orientation or slightly biased, it could cause important damages to Hemisson or the extension module.

2.2 Uninstall

Check first that Robot is switched off, hold firmly Hemisson in your hand, pull as much vertical as possible the HemGenIO module with the other hand. Please take extra care in not bending connector pins.



3.1 Overview

Here follows a complete description of the HemGenIO module:

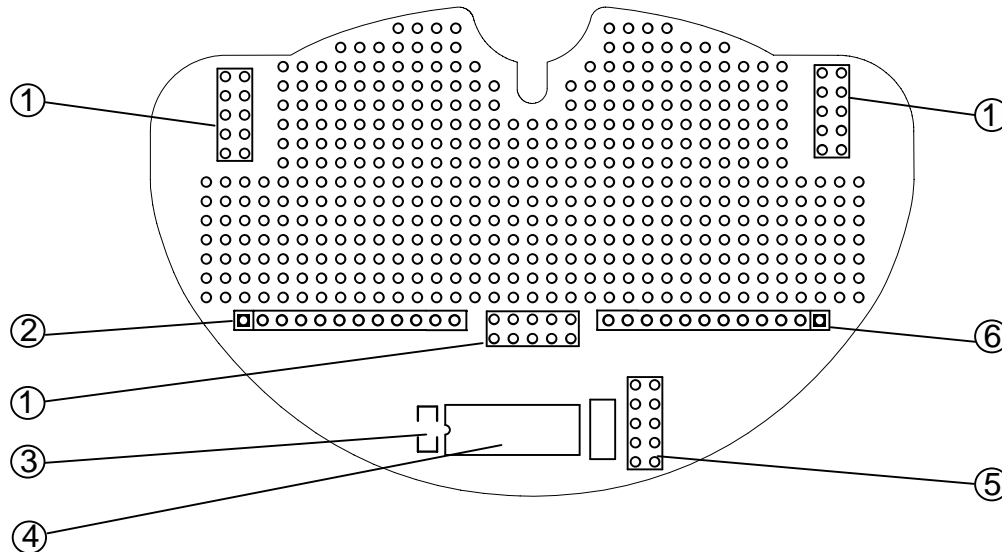


Figure 3.1: HemGenIO modules: topview

1. Hemisson Extension Connectors
2. Digital I/O Connector
3. LED
4. Processor
5. Module Reprogramming Connector
6. Connector for Power Supply, Analog Inputs and I2C Bus

For more information about connectors, please read the related annex.

3.2 I2C Address

The Hemisson robot access to HemLinCam module thanks to I2C bus. The module address is defined on 7 bits + 1 bit for the access mod ("read" or "write").

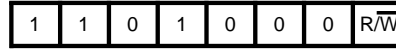


Figure 3.2: Module I2C Adress

Therefore, the HemGenIO address in "read" mode is **0xD1**; and in write mode **0xD0**.

3.3 Register Addresses

To get values of inputs or to define the state of an output, you need to access to the registers as follows below.

Along this section, we will use the following notation:

- *R* : for a "Read" only register.
- *W* : for a "Write" only register.
- *R&W* : for a "Read" and a "Write" register.

3.3.1 Firmware Version Register

Description : allow to read firmware version in processor memory.

Adresse : 0 (0x00)

Accès : R

3.3.2 A0 Analog Input Register

Description : allow to obtain the 8-bit value of the analog input A0.

Adresse : 16 (0x10)

Accès : R

3.3.3 A1 Analog Input Register

Description : allow to obtain the 8-bit value of the analog input A1.

Adresse : 17 (0x11)

Accès : R

3.3.4 A2 Analog Input Register

Description : allow to obtain the 8-bit value of the analog input A2.

Adresse : 18 (0x12)

Accès : R

3.3.5 A3 Analog Input Register

Description : allow to obtain the 8-bit value of the analog input A3.

Adresse : 19 (0x13)

Accès : R

3.3.6 A4 Analog Input Register

Description : allow to obtain the 8-bit value of the analog input A4.

Adresse : 20 (0x14)

Accès : R

3.3.7 A0-A4 Analog Input Register

Description : allow to obtain the 8-bit value of all the analog inputs in the order A0 to A4.

Adresse : 21 (0x15)

Accès : R

3.3.8 D0 Digital I/O Register

Description : in write mode, this pin acts as an output, while in read mode, this pin acts as an input.

Adresse : 32 (0x20)

Accès : R&W

3.3.9 D1 Digital I/O Register

Description : in write mode, this pin acts as an output, while in read mode, this pin acts as an input.

Adresse : 33 (0x21)

Accès : R&W

3.3.10 D2 Digital I/O Register

Description : in write mode, this pin acts as an output, while in read mode, this pin acts as an input.

Adresse : 34 (0x22)

Accès : R&W

3.3.11 D3 Digital I/O Register

Description : in write mode, this pin acts as an output, while in read mode, this pin acts as an input.

Adresse : 35 (0x23)

Accès : R&W

3.3.12 D4 Digital I/O Register

Description : in write mode, this pin acts as an output, while in read mode, this pin acts as an input.

Adresse : 36 (0x24)

Accès : R&W

3.3.13 D5 Digital I/O Register

Description : in write mode, this pin acts as an output, while in read mode, this pin acts as an input.

Adresse : 37 (0x25)

Accès : R&W

3.3.14 D6 Digital I/O Register

Description : in write mode, this pin acts as an output, while in read mode, this pin acts as an input.

Adresse : 38 (0x26)

Accès : R&W

3.3.15 D7 Digital I/O Register

Description : in write mode, this pin acts as an output, while in read mode, this pin acts as an input.

Adresse : 39 (0x27)

Accès : R&W

3.3.16 D8 Digital I/O Register

Description : in write mode, this pin acts as an output, while in read mode, this pin acts as an input.

Adresse : 40 (0x28)

Accès : R&W

3.3.17 D9 Digital I/O Register

Description : in write mode, this pin acts as an output, while in read mode, this pin acts as an input.

Adresse : 41 (0x29)

Accès : R&W

3.3.18 D10 Digital I/O Register

Description : in write mode, this pin acts as an output, while in read mode, this pin acts as an input.

Adresse : 48 (0x30)

Accès : R&W

3.3.19 D11 Digital I/O Register

Description : in write mode, this pin acts as an output, while in read mode, this pin acts as an input.

Adresse : 49 (0x31)

Accès : R&W

3.3.20 I2C Address Change

Description : this register allows to modify the I2C module address. The new address is stored in EEPROM, so that you do not need to modify on reboot. You need to specify a new even address in write mode registre, e.g., 0xA0. If you enter an odd address, e.g., 0x51, the module will default to its default base address: 0xD0.

Adresse : 96 (0x60)

Accès : W

3.4 Electronics

Please note that in order to avoid erroneous reading when reading an unconnected input, all I/Os, either digital or analog, are connected to a 100-kOhm pull-down resistor (see schematics below).

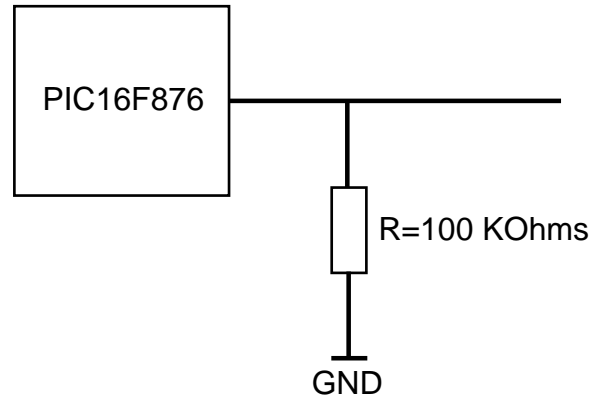


Figure 3.3: Pull-Down



There is two ways to use HemGenIO:

- Program it in C using CCS C Compiler (non provided).
- Control it via serial port either using your own software, or using for instance SysQuake LE.

4.1 Programmation en C

To program HemiOs in C, you must acquire CCS C compiler (link to CCS store in the software section of hemisson.com). We recommend that you first read the latest version of the HemiOs manual (<http://www.hemisson.com/English/assistance.html>). This manual provides all necessary information about installing and using CCS on HemiOs modules.

- Download the latest version of HemiOs, as well as the latest version of HemGenIO Lib including HemGenIO.h.
- Open the project using CCS C Compiler.

To use the following functions, you must first include HemGenIO.h at the beginning of your code:

```
\#include "HemGenIO.h"
```

HemGenIO.h must be in your projects's directory.

HemGenIO.h includes the following functions:

4.1.1 char HemGenIO_Init(void)

Goal: Initialize HemGenIO module. You must call this fonction first when using HemGenIO module.

Example :

```
HemGenIO_Init();
```

4.1.2 char HemGenIO_Read_Version(void)

Goal: Read the HemGenIO firmware version.

Exemple :

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

4.1.3 int1 HemGenIO_Read_Digital(char input)

Goal: Return state (0 or 1) of a digital input. The input parameter allows to select the input (0 to 11).

Example :

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

4.1.4 void HemGenIO_Write_Digital(char input, int1 state)

Goal: Define the state of a digital output. The first parameter selects the output (0 to 11), while the seconds defines its state (0 or 1)

Example :

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

4.1.5 char HemGenIO_Read_Analog(char input)

Goal: Returns a 8-bit analog value (0 = 0V , 255 = 5V) of an analog input (0 to 4)

Example :

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

4.2 Serial Port Control

Thanks to the latest Hemisson firmware available on the web site (<http://www.hemisson.com/English/support.html>) you can access to I2C modules via the RS232 command line. For that, please refer to the latest Hemisson user manual.

Examples :

- W,D0,26,01C: digital output D6 set to 1 (5V).
- R,D0,00,01C: returns firmware version.

You can use any development environment that can access serial port to interface to HemGenIO. For instance, we developped a graphical interface, thanks to SysQuake LE, an advanced technical programming environment, to test your HemGenIO.

You must first plug your HemGenIO to Hemisson, the plug your Hemisson to the serial port, and then connect the PC to the serial port. The SysQuake source code is open source: you can modify it to develop your own interfaces.

A TECHNICAL SPECIFICATIONS



- Weight [g]: NC
- Voltage [V]: 5
- I2C Max. Freq. [KHz]: 400
- Analog Input Number: 5
- AD Converter Resolution [Bits]: 8
- Digital Input/Output Number: 12
- Maximum Output Current of a Digital Output [mA]: 25

B CONNECTORS



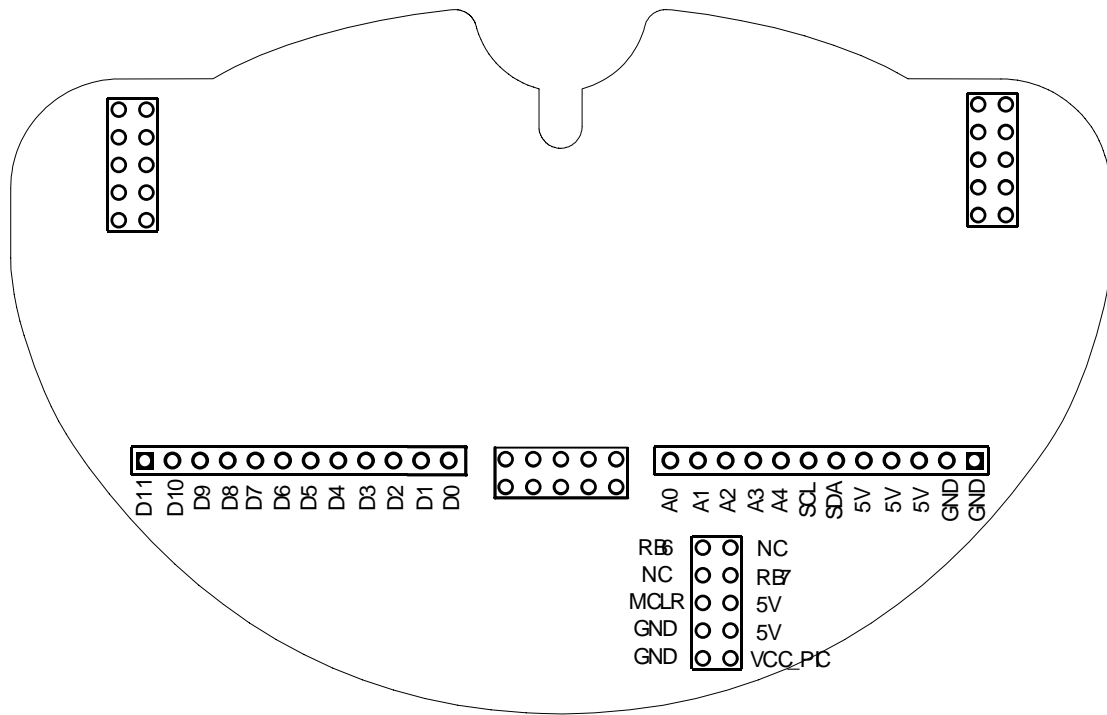


Figure B.1: Connector Details

- D0-D11: Digital input/output
- A0-A4: Analog input
- SCL,SDA: I2C bus signals
- 5V: 5V power supply
- GND: Ground
- RB6,RB7,MCLR,GND,VCC_PIC: Processor Reprogramming Signals (see HemFlexExtProg and HemIcdAdaptor)