

DOCUMENTATION AUTHOR

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

email: info@hemisson.com

Url: www.hemisson.com

CREDITS

Sandra Ide,
Oleg Gerovich.

TRADEMARK ACKNOWLEDGMENTS

Philips: Koninklijke Philips Electronics Corp.

Sysquake: Calerga SA.

MPLAB: Microchip SA.

LabVIEW: National Instruments Corp.

MATLAB: MathWorks Corp.

Webots-Hemisson: Cyberbotics S.A.

NOTICE

- 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.



1	Introduction	4
1.1	The Hemisson Story	4
1.2	How to use this handbook	4
1.3	Warnings	4
1.4	Recycling	5
2	Your First Steps With Hemisson	6
2.1	Package Contents	6
2.2	First Startup	8
3	Robot & Accessories	10
3.1	Global view	10
3.2	The Hemisson Robot	11
3.2.1	Microprocessor	11
3.2.2	LEDs	11
3.2.3	Buzzer	11
3.2.4	IR Remote Control	11
3.2.5	Switches	11
3.2.6	Infra-red sensors	12
3.2.7	Drive mechanism	14
3.3	Accessories	15
3.3.1	Serial Cable	15
3.3.2	9V Battery (Hemisson Pack)	15
3.3.3	Rechargeable Battery (Hemisson DeLuxe Pack)	15
3.3.4	Felt Pen	15
3.3.5	Webots-Hemisson (Hemisson DeLuxe Pack)	16
3.4	Extensions	16
3.4.1	HemLinCam	17
3.4.2	HemGenIO	17
3.4.3	HemTextToSpeech	18
3.4.4	HemUltraSonicSensor	18
4	Running Modes	19
4.1	Exec Running Modes	19
4.1.1	Remote Control	19
4.1.2	Obstacle Avoidance	19
4.1.3	Line Following	20
4.1.4	Dance	21

4.1.5	Bot-Studio Execution	21
4.1.6	Serial Port Control	21
4.1.7	Infra-red Remote Control	23
4.1.8	Software Tools for Hemisson	24
4.2	Reprogramming Hemisson	26
4.2.1	Hemisson Uploader	26
4.2.2	CCS C Compiler	27
4.2.3	Others	27
A	RS232 commands available	28
B	Extension connectors	32
C	Electronic diagram	33
D	Warranty	35



Thank you for buying Hemisson. Hemisson will initiate your exposure to the extraordinary world of mobile robotics. Thanks to its wealth of sensors and its software and hardware openness, you will be able to create complex behavior, making you an expert of this promising technology.

1.1 The Hemisson Story

Hemisson resulted from extra-curricular activities of students at the Swiss Institute of Technology (EPFL). Their goal was to propose a behavior-based robotics course aimed at 10 to 15 year old students. Students would build the robot during the course and then bring it home at the end of the course. The first version of Hemisson was already an affordable, attractive, robust and fully-featured mobile robot. Beyond the original idea, Hemisson proved to be a very attractive tool for teaching many science and technology courses: programming, control, electronics, mechanics, physics of sensors, mathematics...

To face the interest of students and teachers, improvements to the initial version of Hemisson came thanks to K-Team's competence and our long-standing experience with miniature mobile robotics. (www.k-team.com).

1.2 How to use this handbook

This handbook introduces the Hemisson robot and its various operating modes. For a quick start, jump to section 2.2.

If this handbook does not answer one of the problems you are confronted with, please consult the Hemisson Web site (www.hemisson.com) and, especially, the Forum or the FAQs¹.

1.3 Warnings

Here are some recommendations on how to correctly use Hemisson:

¹Frequently Asked Questions

- **Keep the robot away from wet areas.** Contact with water could cause malfunction and/or breakdown.
- **Store your robot in a stable position.** This will avoid the risks of falls, which could break it or cause damage to a person.
- **Check the conformity of your batteries and accumulators.** Only use a standard battery or a rechargeable battery with similar specification as the ones provided in your Hemisson Pack.
- **Do not attach any connectors 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 unused.** When you are finished working with Hemisson, turn it off. It will save battery life.

1.4 Recycling

Think about the end of life of your robot.

Most robot 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.



2.1 Package Contents

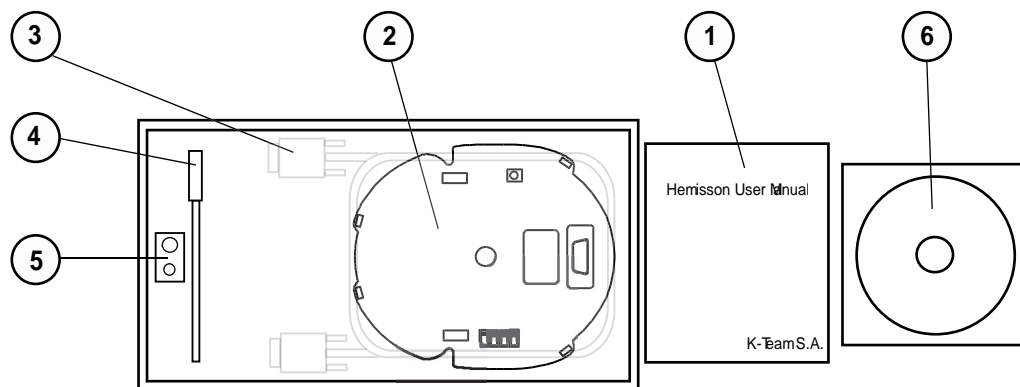


Figure 2.1: Contents of Pack

Your package should contain the following items:

1. Hemisson User Manual.
2. Hemisson Robot.
3. RS232 Cable.
4. Felt Pen.
5. Battery 9V (HemPack) or Ni-MH Rechargeable battery (HemPackDeLuxe).
6. CDROM(s):
 - Hemisson Support CD.
 - Webots-Hemisson CD (HemPackDeLuxe Only).

Here is a visual description of your package content:



Figure 2.2: Hemisson Pack



Figure 2.3: Manual and CD(s)



Figure 2.4: Felt-Pen



Figure 2.5: Battery



Figure 2.6: Hemisson Robot



Figure 2.7: Serial Cable

2.2 First Startup

Throughout this handbook we will use the following notation to indicate the switch positions.

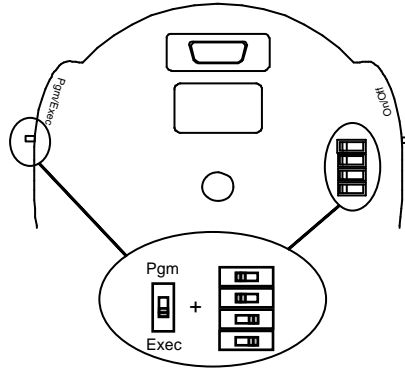


Figure 2.8: Notation of the switches

The test procedure below allows you to check the status of your Hemisson robot. You will find additional description on your Hemisson Support CD.

Please follow the following steps:

- Check that the robot is powered off (refer to section 3.1 to locate the on/off switch).
- Install the battery (Hemisson Pack) or the rechargeable battery (Hemisson DeLuxe Pack). For that, first connect the battery and then insert it down into the hole (see section 3.1).
- Configure the switches as indicated below:

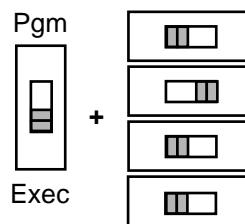


Figure 2.9: Switch settings

- Turn the robot on by putting the master switch in the "On" position (refer to section 3.1).

Hemisson should dance, i.e. drawing arcs of circle inside a fixed inner circle (see complete description of dance behavior in section 4.1).

Warning: when you turn Hemisson off, please wait 5 seconds before switching it back on.



3.1 Global view

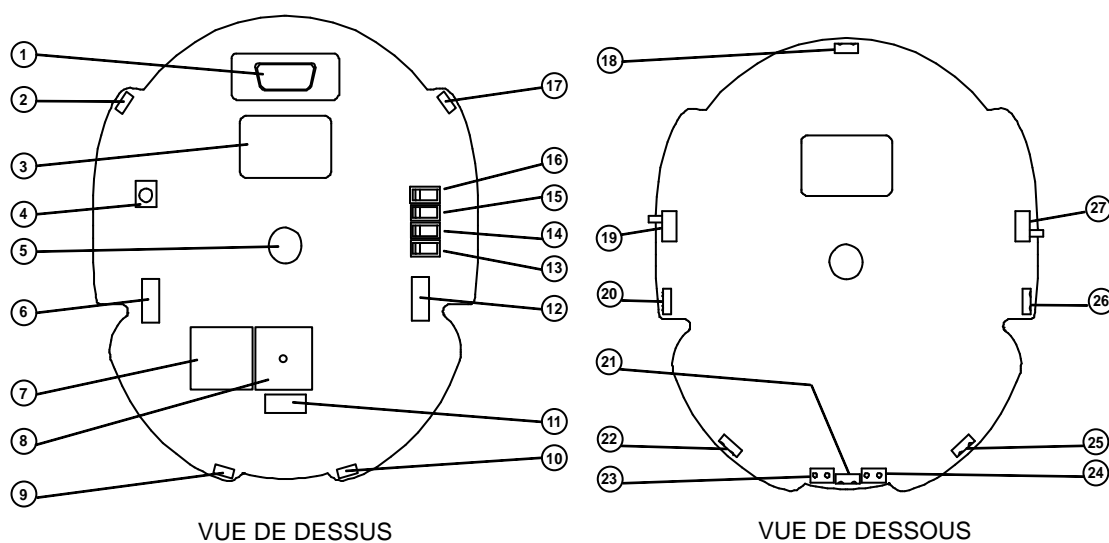


Figure 3.1: Views of the Hemiison Robot

1: RS232 connector	15: Switch 2
2: Pgm/Exec LED	16: Switch 1
3: Battery Location	17: On/Off LED
4: IR Remote receiver	18: Rear sensor
5: Felt-Pen Location	19: Switch On/Off
6: Extension Bus (A)	20: Left sensor
7: Microprocessor	21: Front sensor
8: Buzzer	22: Front-Left sensor
9 : Front Right LED	23: Left Ground Sensor
10: Front Left LED	24: Right Ground Sensor
11: Extension Bus (B)	25: Front-Right Sensor
12: Extension Bus (C)	26: Right Sensor
13: Switch 4	27: Prog/Exec Switch
14: Switch 3	

Warning, when switching off Hemiison, please wait 5 seconds before switching it back on.

3.2 The Hemisson Robot

3.2.1 Microprocessor

Hemisson's microcontroller is a Microchip PIC16F877 at 20MHz in SMD packaging. It is the most powerful 14 bits PIC of Microchip linecard. It relies on a simplified set of 35 instructions.

Inside the PIC16F877, there are three kind of memories:

- 8000 words of 14 Bits of Flash memory¹
- 368 words of 8 Bits of RAM²
- 256 words of 8 Bits of EEPROM³

For more information on PIC16F877 internal resources, please consult documentation on Microchip website (<http://www.microchip.com>).

3.2.2 LEDs

Hemisson has four SMD LEDs. While in Exec mode, i.e, the mode in which Hemisson executes programs in Flash (see Prog/Exec Switch in 3.1 and 4.1), the On/Off LED blinks and the Pgm/Exec LED is off. In Pgm mode, i.e. the mode in which Hemisson Flash Memory can be reprogrammed, the On/Off LED and the Pgm/Exec LED stay on.

3.2.3 Buzzer

Hemisson contains a buzzer circuit. Unlike a loudspeaker, a buzzer can emit sound at a unique frequency, its resonance frequency. The resonance frequency of this buzzer is 4KHz. As a consequence, you cannot produce a melody with Hemisson. For your information, the buzzer consumes 4mA under 5V and has an acoustic power of 85db at 10cm.

3.2.4 IR Remote Control

An IR remote receiver allows to control Hemisson wirelessly. This is a 36kHz demodulator. You can control or send data to your robot with a standard TV remote control. More information is provided in section 4.1.7.

3.2.5 Switches

There are six switches: four on the top and two on the side of Hemisson. The top 4 switches select internal behaviors described in section 4.1.

The two other switches have s predefined function:

¹The Flash memory is non volatile, i.e. any data in memory will remain after the robot is switched off.

²The RAM (Random Access Memory) is volatile, i.e. any data will be lost when the robot is switched off.

³The EEPROM (Electrically Erasable Programmable Read-Only Memory) is non volatile.

- The On/Off switch allows to start or stop Hemisson.
Note: On the printed circuit board, you will find the label "On/Off", which will help you remember its use.
- The Pgm/Exec switch allows to select the mode Pgm (Programming) or Exec (Execution). Pgm mode allows to update or change Hemisson's Firmware⁴.
Note: On the printed circuit board, you will find the label "Pgm/Exec", which will help you remember its use.

3.2.6 Infra-red sensors

There are 8 similar IR sensors: six on the side and two facing the ground. As a matter of fact, the sensor includes two components:

- an infra-red light emitter (LED).
- an infra-red light receiver (phototransistor).

We will describe below the two operating modes of these infra-red sensors.

Passive mode

In passive mode, only the phototransistor is used to measure ambient IR emission, also called luminosity measurement. For your information, there are many natural or artificial IR sources in your environment, e.g. sunlight, incandescent lamps or a candle flame.

Active mode

In active mode, IR light is first emitted by the LED and then a measurement of ambient IR light is made by the phototransistor. If there is an object near the sensor, it will reflect IR light. Moreover, the closer the obstacle, the more IR light will be reflected.

However, IR reflection from an object will vary very much depending on the material, color and surface finishing. Usually, the darker the object, the less IR it reflects, but, as you can see below, there are exceptions:

- White paper: 100%
- White PVC: 90%
- White Polystyrene: 120%
- Black on white paper made with drawing ink (Higgins, Pelikan, Rotring): 4-6%
- Black on white paper made with fiber-tip pen, black (Stabilo): 76%
- Plexiglass, 1mm thick: 10%
- Cast aluminium, matt: 45%
- Aluminium, bright: 110%

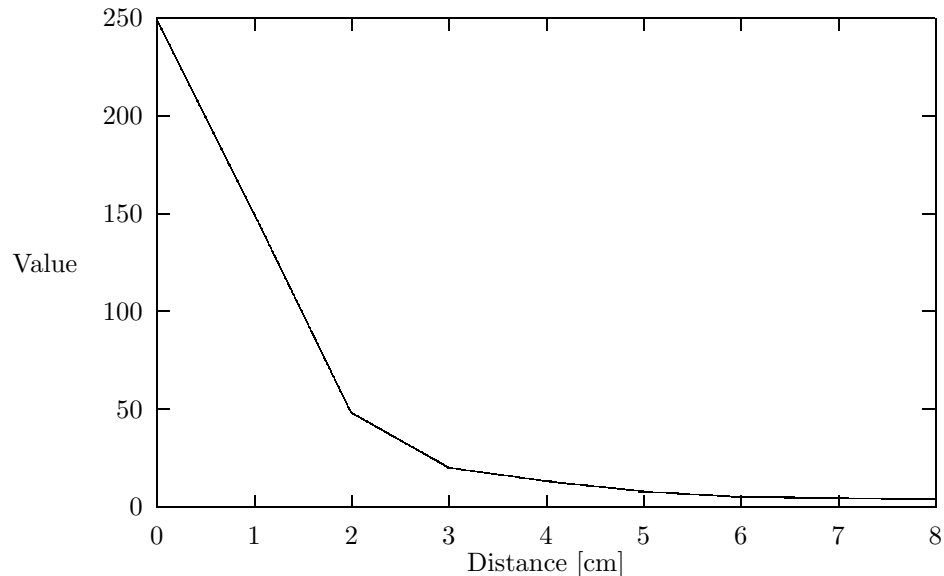
⁴Before use, please read carefully section 4.2.

- Gold plating, matt: 150%
- Brass, bright: 160%

Hence, except if your robot's environment is made of the same material everywhere, it cannot recover distance because according to the very nature of the obstacle, it will not measure the same distance for two different obstacles at the same distance. For ground sensors, the active mode will vary according to the material beneath. Hence, the robot can follow a line on the ground or detect table edges if materials are well chosen (see list above).

Last, in order to avoid perturbation by ambient light, the robot samples ambient light before and after emitting IR light. The difference of the two sampled values is returned. However, under certain lighting conditions, e.g. sunlight, infra-red sensors may saturate and, therefore, active mode will return incorrect information.

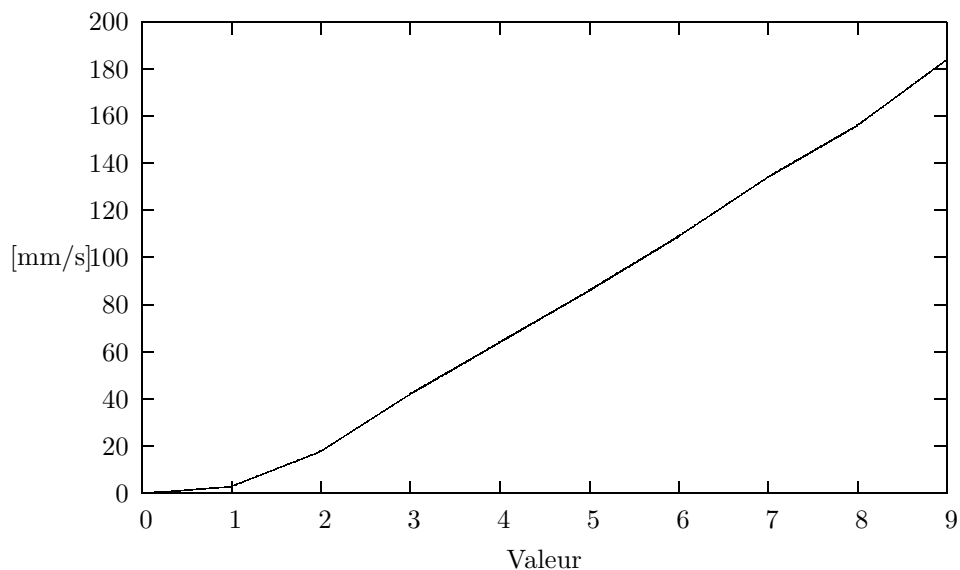
Typical response in active mode



The graph above displays the typical value read on Hemisson while changing the distance from a given obstacle. You can notice that the response is exponential, i.e. there is no linear correlation between the value read and the distance of the obstacle.

3.2.7 Drive mechanism

Hemisson uses two Direct Current (DC) motors to drive its two wheels. It is a differential drive system. The main advantages of this system are related to steering. By making the motors turn in opposite directions, the robot will spin around its own axis, which makes it much more maneuverable than a system that can only turn as it moves forwards. Differential drive also makes it very easy to make turns because it just depends on the relative speeds of the motors.



A Pulse Width Modulation (PWM) signal allows to vary speed linearly on both motors.

3.3 Accessories

We will detail hereafter accessories included in your Hemisson Pack.

3.3.1 Serial Cable

We provide a serial cable to connect Hemisson to your computer. For that, your computer must have a DB9 serial port connector, i.e. a 9-pole connector as shown in figure 3.2.

If your computer has no serial port, but only USB ports, you can purchase a USB

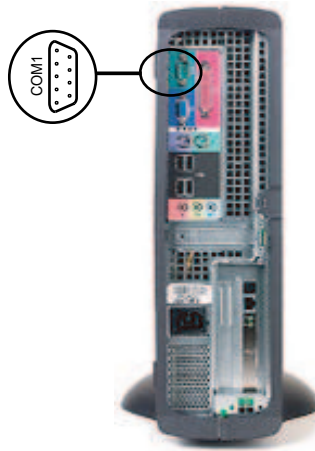


Figure 3.2: Where to connect Hemisson

serial adapter from your local computer store.

3.3.2 9V Battery (Hemisson Pack)

This battery allows you to use the robot. When the battery runs out, **DO NOT RECHARGE IT**. You can buy an equivalent alkaline battery from your local store (type 6LR61 or 9V E-Block). For cleaner environment, we recommend to use rechargeable batteries.

3.3.3 Rechargeable Battery (Hemisson DeLuxe Pack)

We provide a Ni-MH rechargeable battery, with which you will be able to use Hemisson while respecting the environment. You can acquire a similar rechargeable battery in your local store (type 6F22, 8.4V, Ni-MH, 150mAh).

3.3.4 Felt Pen

A felt pen is provided with Hemisson to draw Hemisson's trajectory. You must set it in the hole right in the middle (see section 3.1). First, switch off Hemisson. Then remove the cap and insert firmly the pen in its location while holding Hemisson



Figure 3.3: Felt-Pen in its location

in your hand. During this procedure **do not use any infra-red sensors for support** to avoid damaging them. Adjust the height of the pen, so that the wheels touch the ground. Finally, you should test it, for example, by setting Hemisson in dance running mode (refer to 4.1.4). Most of the time, you will need to adjust the position slightly. If the pen rubs the ground too much, push it up slightly while turning it. If it does not have enough contact, push it down slightly while turning. This is a very tight adjustment, and you will need a few trials before success, especially in the beginning.

3.3.5 Webots-Hemisson (Hemisson DeLuxe Pack)

Webots-Hemisson (the 3D simulator) and Bot-Studio (the graphical programming environment) are provided on a CD-ROM. You may access the CD-ROM with Microsoft Windows 95/98/ME/2000/XP and Linux.

To install Webots-Hemisson, please insert the CD-ROM and follow instructions (see section 3.3.5). If nothing happens upon CD-ROM insertion, please open the file `index.html` in CD-ROM root directory. Your license number is printed on the CDROM case.

3.4 Extensions

Hemisson has an open architecture: 3 extension connectors allow to add devices, either made by K-Team S.A. (cf. www.hemisson.com), or that you will build yourself.

Each of the three extension connectors is dedicated:

- Front connector provides I2C bus communication.
- Left side connector allows flash-memory programming.
- Right side connector provides serial port communication.

You will find all the information to build your own devices in connector pinout schematics in the Appendix B, and in Hemisson schematics in Appendix C.

3.4.1 HemLinCam

The HemLinCam module allow Hemisson to perceive its environment. The camera reads one line of 102 pixels in 256 level of gray. The optic block is a standard one (M12x0.5), so that you can change it to fit to your specific needs. As in all the intelligent Hemisson modules, there is an board processor (PIC16F876), dedicated here to visual processing. Like Hemisson, the source code of the visual processing in under LGPL license and, as a consequence, you can write your own visual routines. To download your own code on HemLinCam, you can use the same tools as for Hemisson (CCS C, In-Circuit-Debug Interface, Hemisson Uploader, External Programmer Interface).

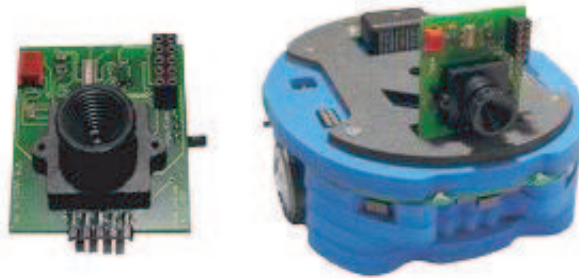


Figure 3.4: HemLinCam

3.4.2 HemGenIO

The HemGenIO allows to interface your own electronics. It is a perfect tool to implement your own module. A board area allows you to add components (2.54mm/.1" spacing). The documentation explains how to access your own peripherals from the central processor (12 digital I/O, 5 analog 8-bit inputs and I2C bus).

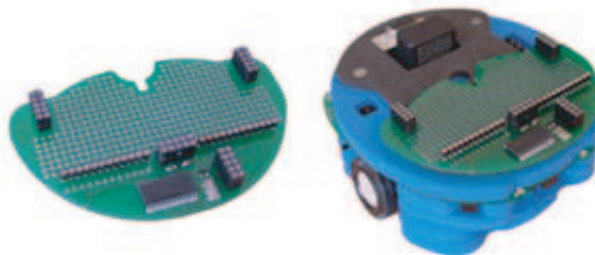


Figure 3.5: HemGenIO

3.4.3 HemTextToSpeech

This interface makes your Hemisson speaks fluent English. The on-board speaker will pronounce every word transfered in ASCII code on Hemisson I2C bus.



Figure 3.6: HemTextToSpeech

3.4.4 HemUltraSonicSensor

If you are looking for a higher range than the default Hemisson sensors, this ultrasound sensor enables Hemisson to measure distances of up to 6 m with a 1 cm precision.

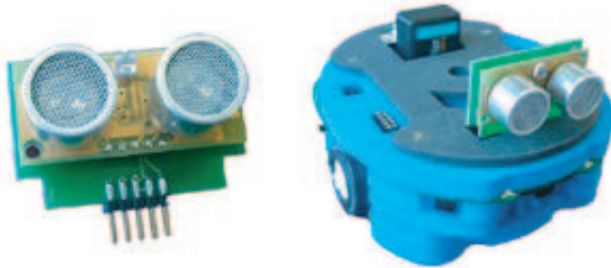


Figure 3.7: HemUltraSonicSensor

4 RUNNING MODES



Hemisson starts in the run mode selected by the position of the Pgm/Exec switch :

- Exec Position: execute program in flash memory.
- Pgm Position: use in flash programming mode. This is an advanced feature reserved for users wanting either to update HemiOS, or to program Hemisson in C (see Hemisson Uploader and HemiOS manuals).

Note: The Pgm/Exec switch is read at robot start-up, therefore, to take into account a switch change done while running, you must restart the robot.

4.1 Exec Running Modes

For all operations described in this section, the Pgm/Exec switch must remain in Exec position. Hemisson is provided with a remote control running mode and a set of basic behaviors for demonstration purpose.

4.1.1 Remote Control

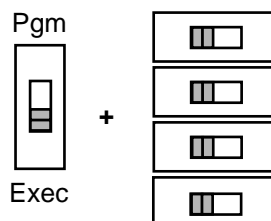


Figure 4.1: Settings for the Remote-Control Running Mode

This mode allows to initialize the robot in controlled mode. This configuration is valid to control via serial port (see section 4.1.6), via IR remote control (see section 4.1.7) and for use with Webots-Hemisson.

4.1.2 Obstacle Avoidance

In this running mode, Hemisson detects obstacles in front of it and avoids them. The left front LED indicates that an obstacle has been detected on the left, similarly the right front LED indicates that an obstacle has been detected on the right. When obstacles are found on the right and on the left, Hemisson emits a sound and

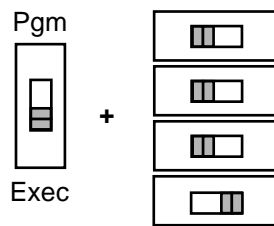


Figure 4.2: Settings for the Obstacle-Avoidance Running Mode

makes a half-turn.

Beware of any parasitic source of infra-red light, like incandescent lamps and sunlight, that can perturbate Hemisson's behavior.

4.1.3 Line Following

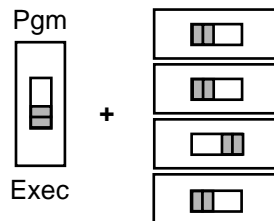


Figure 4.3: Settings for the Line-Following Running Mode

Hemisson can follow a line by reading its two front infra-red sensors that are directed toward the ground. The line must be a black line over a white ground and must be at least 20mm in width. You will find printable track sections on the CD-Rom in the "Extras" directory.

The left front LED shows that Hemisson is on the left handside of the line. Similarly, the right front LED shows that Hemisson is on the right handside of the line.

Beware, on start-up, the robot must be set on the black line.

Beware of any parasitic source of infra-red light, like incandescent lamps and sunlight, that can perturbate Hemisson's behavior.

4.1.4 Dance

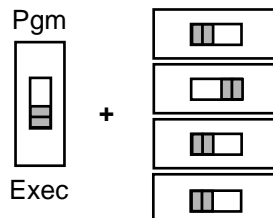


Figure 4.4: Settings for the Dance Running Mode

In this mode, Hemisson carries out a circular "dance". If you add a felt-pen in Hemisson's center, Hemisson will draw an unstable hypocycloid (more information on hypocycloid at: <http://mathworld.wolfram.com/Hypocycloid.html>).

4.1.5 Bot-Studio Execution

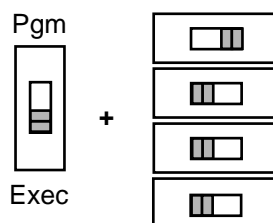


Figure 4.5: Settings for Bot-Studio Execution Mode

This running mode allows to execute a code previously compiled and stored in memory with Bot-Studio (please consult section 4.1.8 and Webots-Hemisson documentation available on the CD-Rom).

4.1.6 Serial Port Control

You can control Hemisson via your PC serial port. Many software allow to directly monitor serial port communication, like Hyperterminal or TeraTerm.

Here is the procedure to control Hemisson with Hyperterminal:

- Connect Hemisson to one of your computer's serial ports, using the RS232 serial port cable provided in the pack.
- Click Start → Programs → Accessoires → Communications → Hyperterminal
- Define a name for the session (like Hemisson_CCOM1_115200).

- Select the serial port on which HemiOS is connected.
- Set all the parameters as shown in figure 4.6.

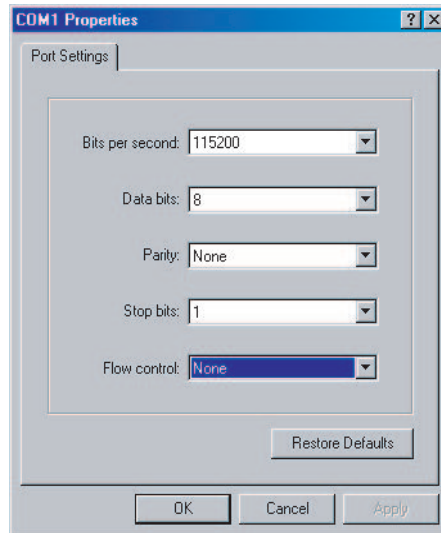


Figure 4.6: Hyperterminal Parameters

- Set switches to start in remote control mode (refer to section 4.1.1).
- Switch HemiOS on.

You should see the same start-up message as in figure 4.7. Here is a sequence of commands to test HemiOS:

- type: B ↵. HemiOS should display the version of the HemiOS software.
- by typing D,-5,5 ↵. HemiOS should turn on itself.
- by typing D,0,0 ↵. HemiOS should stop turning.

Notation :

↵ stands for the Enter or Return key on the keyboard.

All the available commands are described in appendix A.

For your information, you can use any third party software, like MATLAB, LabVIEW, (etc...) to control HemiOS through a serial port. You can also write your own software to control HemiOS via the serial link.

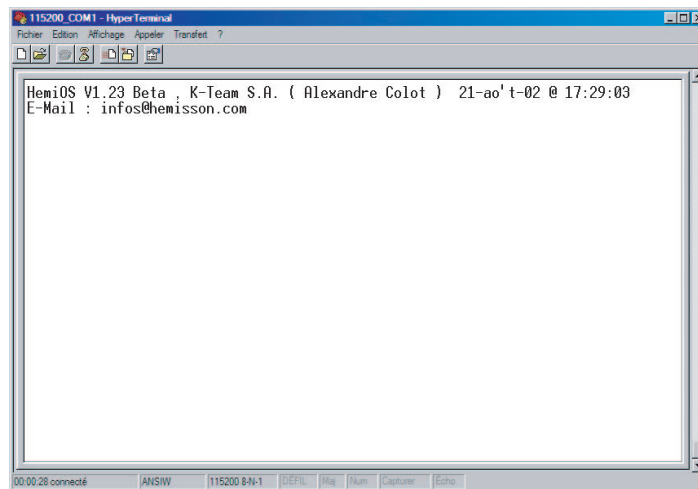


Figure 4.7: HemiOS Start-up Message on Serial Port

4.1.7 Infra-red Remote Control

In order to proceed, your infra-red remote control must follow the RC5 standard. Many TV sets are provided with RC5 remote control like Philips, and also Daewoo, Goldstar, Hitachi, Loewe, Mitsubishi, Samsung, and many others. Of course, if you have a programmable remote control, you can emulate the RC5 standard.

Note: *You must use a TV remote control and not a remote control for a VCR or any other auxilliary device. If you are using a universal remote control, you must first set your remote control in TV mode.*

As a matter of fact, signals differ for a TV keyboard and a VCR keyboard to avoid cross-communication.

Once the compatibility problem settled, you must set your HemiOS in remote control mode (see section 4.1.1). The available commands are (refer to figure 4.8):

- Channel 1: Moves leftward in a large circle.
- Channel 2: Moves forward.
- Channel 3: Moves rightward in a large circle.
- Channel 4: Moves leftward on itself.
- Channel 5: Stop.
- Channel 6: Moves rightward on itself.
- Channel 7: Same as 1 but moves backward.
- Channel 8: Moves backward.
- Channel 9: Same as 3 but moves backward.

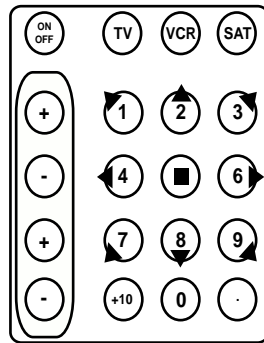


Figure 4.8: Remote Control Keyboard

4.1.8 Software Tools for Hemisson

Any third party software, like MATLAB, LabVIEW, (etc...) can control Hemisson through a serial port. You can also write your own software to control Hemisson via the serial link.

Webots-Hemisson

Webots-Hemisson is provided with your Hemisson DeLuxe Pack. This is a powerful software for programming and simulating Hemisson.

The Bot-Studio windows allows you to define Hemisson's behavior graphically. Bot-Studio programming relies on finite state machine (FSM). FSM is often used in famous first-person shooters (Quake, Doom, etc...). FSM belongs to functional programming, an alternative to procedural programming (Basic, C, ...). In procedural programming, you specify an explicit sequences of steps to follow to produce a result. In functional programming, you specify a set of function definitions and an expression whose value is output as the program's result. Said differently, a FSM defines a self-running engine and not a sequence of action. A FSM contains states and triggers between states. For instance, here is a FSM that makes Hemisson move forward till you find an obstacle:

- State go forward, where you set motor speed to, for example +5.
- State stop, where you set motor speed to 0.
- Transition obstacle detected from go forward to stop, triggered by, for example a value superior to 100 on the front infrared sensor.
- Transition obstacle removed from stop to go forward, triggered by, for example a value inferior or equal to 100 on the front infrared sensor.

Bot-Studio allows you to upload a FSM directly to Hemisson.

Moreover, Bot-Studio allows you to simulate a FSM inside inside a virtual 3D world that you can modify to match your own environment.

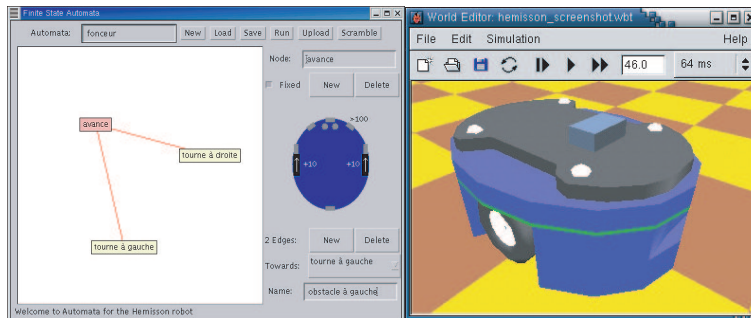


Figure 4.9: Bot-Studio and Webots-Hemisson Screenshots

To install Webots-Hemisson, please insert the CD-Rom and follow the instructions (see section 3.3.5). If nothing happens after CD-ROM insertion, please open the file index.html in the CD-ROM root directory.

Beware, you must register your license before having access to all Webots-Hemisson features.

SysQuake

SysQuake is a professional mathematical software developed by Calerga. SysQuake Light Edition (LE) is provided within Hemisson Pack and Hemisson DeLuxe Pack. SysQuake LE allows to control Hemisson from your PC serial port. Thanks to SysQuake, you will be able to visualize graphically all Hemisson sensors and set all Hemisson parameters. A sample code for Hemisson is provided under GPL license on CDROM. Thanks to this example, you will be able to develop your own control interface. For more information on SysQuake, please consult the SysQuake tutorial at <http://www.calerga.com/>.

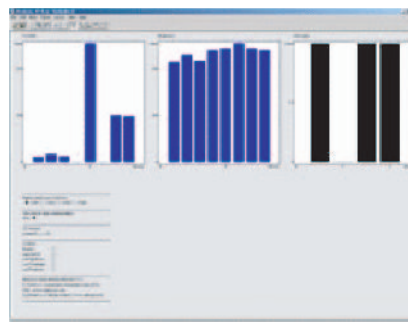


Figure 4.10: SysQuake LE

Others

Any development environment able to send ASCII character on the serial port of your computer can be used to control HemiSSon.

4.2 Reprogramming HemiSSon

The Pgm Running mode allows to program HemiSSon's flash memory. It allows to update your robot with HemiSSon Uploader, but also to fully reprogram it with CCS C compiler or your own set of PIC tools.

Beware, uploading your own code inside HemiSSon's flash memory means that you are responsible for damages caused to HemiSSon's memory.

As a matter of fact, the serial port upload process relies on a boot loader at the bottom of the memory. If your code overwrite this piece of code, you will need the external programmer interface (HemFlexExtProg) and an adequate PIC programmer to reprogram HemiSSon. You can also return HemiSSon to K-Team S.A., but the repair is not covered by the warranty.

4.2.1 HemiSSon Uploader

HemiSSon Uploader allows to upgrade your firmware or to load your own version of HemiOS. N.B.: Instead of HemiSSon's Uploader, you can also connect your favorite



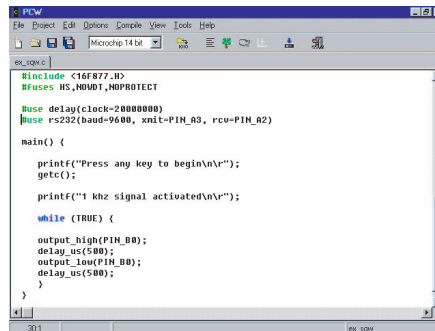
Figure 4.11: HemiSSon Uploader

PIC serial programmer via the left-handside HemiSSon extension connector (for pinouts refer to Appendix B).

4.2.2 CCS C Compiler

If you aim at accessing directly HemiOS's hardware, you may use the CCS C Compiler. The CCS C compiler is the development environment of HemiOS's firmware (HemiOS). HemiOS is under GPL, and as consequence, you will be able to modify it.

You will find the latest source code of HemiOS on the web site:
<http://www.hemisson.com/English/support.html>



```
ex_sig.c |
#include <16F877.h>
#uses RS,NOOBT,NOPROTECT

#use delay(clock=2000000)
#use rs232(baud=9600, xmit=PIN_A3, rcv=PIN_A2)

main() {
    printf("Press any key to begin\n");
    getch();
    printf("1 khz signal activated\n");
    while (TRUE) {
        output_high(PIN_B0);
        delay_us(500);
        output_low(PIN_B0);
        delay_us(500);
    }
}
```

Figure 4.12: CCS C Compiler (PCW version)

4.2.3 Others

You are also free to use any other PIC development tools to program HemiOS. We provide the necessary Bootloader code on our web site.

Beware, we recommend having a PIC programming experience for the operation.

A RS232 COMMANDS AVAILABLE



Notation:

␣ stands for carriage return, that is press the Enter or Return key on the keyboard.

\r stands for ASCII character 0xA (line feed).

\n stands for ASCII character 0xD (carriage return).

A Unused

B Display HemiOS Version

Format of the command: B ␣

Format of the response: b,version_HemiOs\r\n

Effect: Return the HemiOs version present in the flash memory.

C Unused

D Set Speed

Format of the command: D,speed_motor_left,speed_motor_right ␣

Format of the response: d\r\n

Effect: Sets the speed of both motors. 0 stops the engine. Maximum forward speed is 9 and maximum backward speed is -9.

Example: D,5,-5 ␣

Note: Keep the speed values between -9 and 9.

E Unused

F Unused

G Unused

H Beep

Format of the command: H,Buzzer_State >

Format of the response: h\r\n

Effect: Generates a continuous beep (0=Off, 1=On).

Example: H,1 >

I Read Switches

Format of the command: I >

Format of the response: i,val_int1,val_int2,val_int3,val_int4\r\n

Effect: Reads the status of the four top switches (0=robot's right handside, 1=robot's left handside). The first switch comes first starting from the front of the robot.

J Unused

K Unused

L Change LED state

Format of the command: L,Led_On/Off,Led_Pgm/Exec,Led_Front_Left,Led_Front_Right >

Format of the response: l\r\n

Effect: Turns on or off four LEDs (0=Off, 1=On) on Hemisson.

Example: L,1,0,1,0 >

Note: Led_On/Off is permanently blinking and this command cannot stop it.

M Unused

N Read proximity sensors

Format of the command: N >

Format of the response: n,Front,FrontRight,FrontLeft,Right,Left,Rear,GroundRight,GroundLeft\r\n

Effect: Reads the 8-bit (0 to 255) proximity values of each infrared sensor. The smaller the value, the further the object is from it. A value of 255 means that an obstacle is very (too) close.

O Read ambient light sensors

Format of the command: O >

Format of the response: o,Front,FrontRight,FrontLeft,Right,Left,Rear,GroundRight,GroundLeft\r\n

Effect: Reads the 8-bit (0 to 255) brightness values of each infrared sensor. The

smaller the value, the more infrared light is detected. A value of 255 means that there is no light detected in front of the sensors.

P Unused

Q Reserved for Webots-Hemisson

R Read I2C Extension Bus

Format of the command: R,Slave_Write_Address,Register_Address,Number_of_Register >

Format of the response: r,Register_Data\r\n

Effect: Reads a given number of registers of an I2C peripheral. Please provide register write address, i.e., odd address.

Example: R,C0,20,03 >

Reads three registers at address 0x20, 0x21 and 0x22 of peripheral 0xC1.

Note that all parameters must be two characters equivalent to the value in hexadecimal.

S Reserved for Webots-Hemisson

T Read TV remote buffer

Format of the command: T >

Format of the response: t,TV_Data\r\n

Effect: Reads the last byte received by the TV remote receiver.

U Reserved for Webots-Hemisson

V Unused

W Write I2C Extension Bus

Format of the command: W,Slave_Address,Register_Address,Number_of_Register >

Format of the response: w\r\n

Effect: Writes a register of an I2C peripheral.

Example: W,C0,00,FF >

Writes 0xFF at address 0x00 of peripheral 0xC0.

Note that all parameters must be two characters equivalent to the value in hexadecimal.

X Reserved for Webots-Hemisson

Y Unused

Z Processor Reset

Format of the command: Z ⤵ *Format of the response:* z\r\n

Effect: This command allows to reset robot as if it was cycled On/Off.

B EXTENSION CONNECTORS

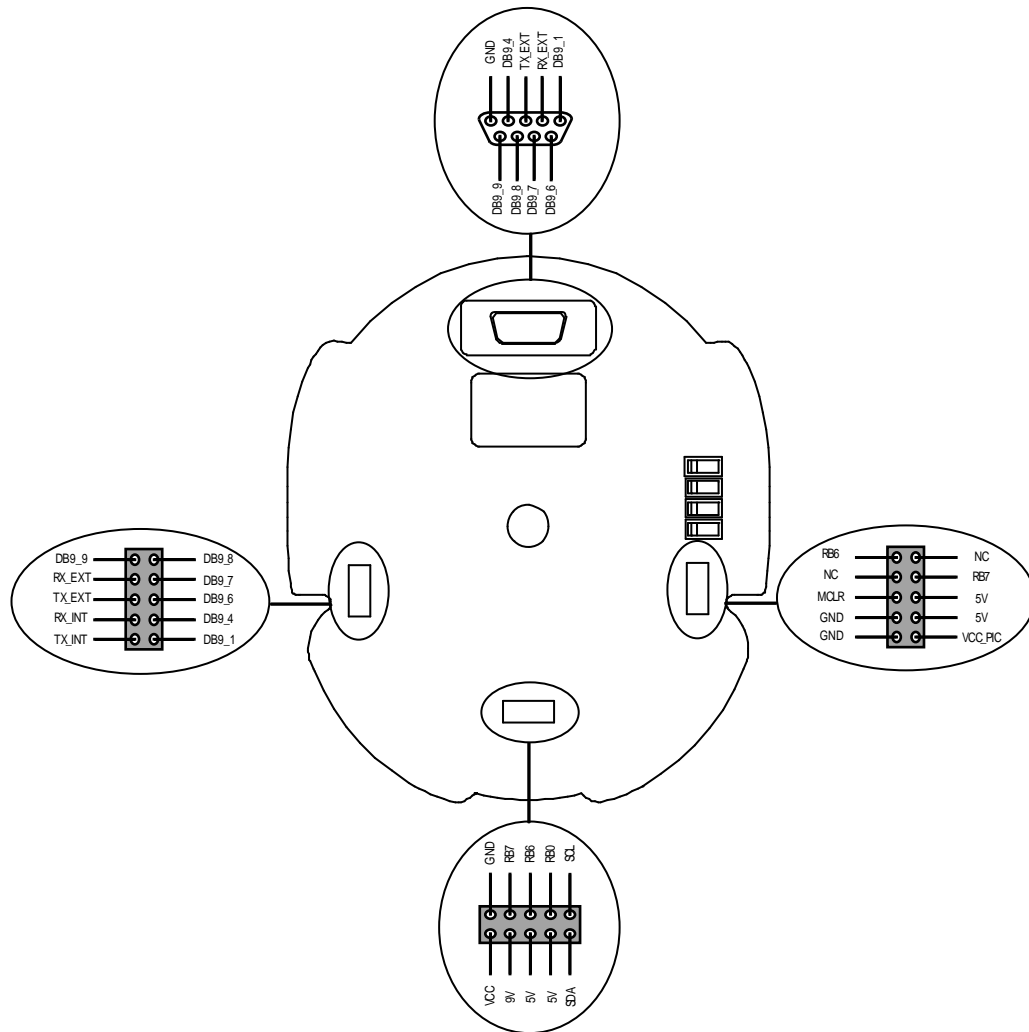


Figure B.1: Hemisson connectors

Note: You will find the meaning of the labels in appendix C.

C ELECTRONIC DIAGRAM



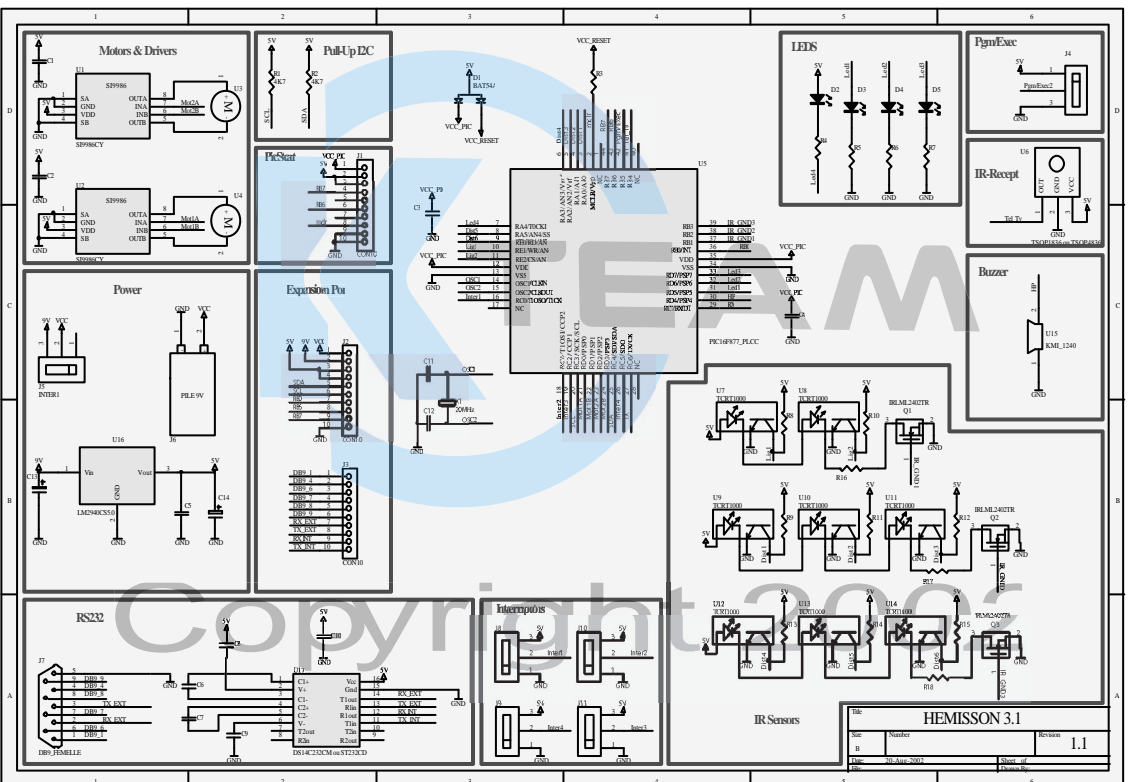


Figure C.1: Hemisson Schematics: Please use the electronic Version for zoomed-in view

D WARRANTY



K-TEAM warrants that Hemisson Pack, respectively Hemisson Pack DeLuxe, is free from defects in materials and workmanship and in conformity with the respective specifications of the product for the minimal legal duration, respectively two (2) years from the date of delivery.

Upon discovery of a defect in materials, workmanship or failure to meet the specifications in the Product during the afore mentioned period, Customer must request help on Hemisson Internet forum on <http://www.hemisson.com> by detailing:

- The type of Hemisson used (package, version).
- The expansion modules.
- The programming environment of the robot (standard, version, OS).
- The standard use of Product before the appearance of the problem.
- The description of the problem.

If no answers have been received within two working days, Customer can contact K-TEAM support by phone or by electronic mail with the full reference of its order and Hemisson serial number.

K-TEAM shall then, at K-TEAM's sole discretion, either repair such Product or replace it with the equivalent product without charging any technical labor fee and repair parts cost to Customer, on the condition that Customer brings such Product to K-TEAM within the afore mentioned period. In case of repair or replacement, K-TEAM may own all the parts removed from the defective Product. K-TEAM may use new and/or reconditioned parts made by various manufacturers in performing warranty repairs and replacement of the Product. Even if K-TEAM repairs or replaces the Product, its original afore mentioned warranty term is not extended.

This limited warranty is invalid if the factory-applied serial number has been altered or removed from the Product.

This limited warranty covers only the hardware and software components contained in the Product. It does not cover technical assistance for hardware or software usage and it does not cover any software products contained in the Product; K-TEAM excludes all warranties expressed or implied in respect of any additionnal software provided with Product and any such software is provided "AS IS" unless expressly provided for in any enclosed software limited warranty. Please refer to

the End User License Agreements included with the Product for your rights with regard to the licensor or supplier of the software parts of the Product and the parties' respective obligations with respect to the software.

This limited warranty is non-transferable.

It is likely that the contents of Customer's flash memory will be lost or reformatted in the course of the service and K-TEAM will not be responsible for any damage to or loss of any programs, data or other information stored on any media or any part of the Product serviced hereunder or damage or loss arising from the Product not being available for use before, during or after the period of service provided or any indirect or consequential damages resulting therefrom.

IF DURING THE REPAIR OF THE PRODUCT THE CONTENTS OF THE FLASH MEMORY ARE ALTERED, DELETED, OR IN ANY WAY MODIFIED, K-TEAM IS NOT RESPONSIBLE WHATSOEVER. CUSTOMER'S PRODUCT WILL BE RETURNED TO CUSTOMER CONFIGURED AS ORIGINALLY PURCHASED (SUBJECT TO AVAILABILITY OF SOFTWARE).

Be sure to remove all third parties' hardware, software, features, parts, options, alterations, and attachments not warranted by K-TEAM prior to Product service. K-TEAM is not responsible for any loss or damage to these items.

This warranty is limited as set out herein and does not cover, inter alia, any consumable items (such as batteries) supplied with the Product; any accessory products which is not contained in the Product; cosmetic damages; damage or loss to any software programs, data, or removable storage media; or damage due to (1) acts of God, accident, misuse, abuse, negligence, commercial use or modifications of the Product; (2) improper operation or maintenance of the Product; (3) connection to improper voltage supply; or (4) attempted repair by any party other than a K-TEAM authorized robot service facility.

This limited warranty does not apply when the malfunction results from the use of the Product in conjunction with any accessories, products or ancillary or peripheral equipment, or where it is determined by K-Team that there is no fault with the Product itself.

K-TEAM EXPRESSLY DISCLAIMS ALL OTHER WARRANTIES THAN STATED HEREINABOVE, EXPRESSED OR IMPLIED, INCLUDING WITHOUT LIMITATION IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE TO THE FULLEST EXTENT PERMITTED BY LAW.

Limitation of Liability: IN NO EVENT SHALL EITHER PARTY BE LIABLE TO THE OTHER FOR ANY INDIRECT, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES RESULTING FROM PERFORMANCE OR FAILURE TO PERFORM UNDER THE CONTRACT, OR FROM THE FURNISHING, PERFORMANCE OR USE OF ANY GOODS OR SERVICE SOLD OR PROVIDED PURSUANT HERETO, WHETHER DUE TO A BREACH OF CONTRACT, BREACH OF WARRANTY, NEGLIGENCE, OR OTHERWISE. SAVE THAT NOTHING HEREIN SHALL LIMIT EITHER PARTY'S LIABILITY FOR DEATH OR PERSONAL INJURY ARISING FROM ITS NEGLIGENCE, NEITHER PARTY SHALL HAVE ANY LIABILITY TO THE OTHER FOR INDIRECT OR PUNITIVE DAMAGES OR FOR ANY CLAIM BY ANY THIRD PARTY EXCEPT AS EXPRESSLY PROVIDED HEREIN.