

# HemiSSon

HemLinCam

**Documentation drawn up by:**

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

E-mail: <mailto:info@hemisson.com>  
Website: <http://www.hemisson.com/>

**Registered Trade Mark:**

Hemisson: K-TEAM SA.

**Please note:**

- The contents of this manual may be changed without prior notification.
- Every effort has been made to ensure the accuracy of this manual.  
However, if you do find an error, we would be grateful if you would inform K-Team S.A.
- Notwithstanding the above, K-Team S.A. cannot be held liable for any consequences arising from an error in this manual.

# Table of Contents

---

<b>1</b>	<b>Introduction</b>	<b>3</b>
1.1	How to Use this Manual.....	3
1.2	Precautionary Measures.....	3
1.3	Recycling.....	3
<b>2</b>	<b>Connecting Up</b>	<b>4</b>
2.1	Fitting.....	4
2.2	Removal.....	5
<b>3</b>	<b>The HemLinCam Module</b>	<b>6</b>
3.1	Overall View.....	6
3.2	I2C Address. . . . .	7
3.3	Register Addresses.....	7
3.3.1	Firmware Register.....	7
3.3.2	ReadPixels Register.....	7
3.3.3	Read Thresholded Pixels Register.....	7
3.3.4	Threshold Value Register.....	8
3.3.5	Exposure Time Register.....	8
3.3.6	LED Register.....	8
<b>4</b>	<b>Utilisation</b>	<b>9</b>
4.1	C Programming.....	9
4.1.1	Void HemLinCam Init(void).....	9
4.1.2	Char HemLinCam Read Version(void).....	9
4.1.3	Void HemLinCam Set Threshold(unsigned char).....	9
4.1.4	Unsigned Char HemLinCam Read Threshold(void).....	10
4.1.5	Void HemLinCam Set Exposure Time(unsigned char).....	10
4.1.6	Unsigned Char HemLinCam Read Exposure Time(void).....	10
4.1.7	Void HemLinCam Read Pixels(void).....	10
4.1.8	Void HemLinCam Read Pixels Thresholded(void).....	10
4.1.9	Void HemLinCam Set LED State(char).....	11
4.2	Serial Port Control.....	11
<b>A</b>	<b>Technical Specifications</b>	<b>12</b>
<b>B</b>	<b>Connectors</b>	<b>13</b>

# 1 Introduction

---

The HemLinCam module allows you to add the dimension of vision to Hemisson. In actual fact, with this module you will be able to build information into your algorithms that includes a lot more than just proximity alone.

## 1.1 How to Use this Manual

This manual is an introduction to the HemLinCam module designed for the Hemisson robot. If you cannot find any information in this manual relating to a given problem that you encounter, please go to the Hemisson website (<http://www.hemisson.com/>), and in particular the Forum or the FAQs<sup>1</sup> section.

## 1.2 Precautionary Measures

Here are a few recommendations for proper use of the HemLinCam module:

- **Keep it away from damp areas.**  
Any contact with water might cause it to short circuit.
- **Always make sure it is placed in a stable position.**  
This will avoid any risk of it falling that might lead to its breaking or causing injury to another.
- **Do not plug in the module whilst the robot is switched on.**  
So as to avoid all possibility of damage, all connections should be made should be done when the robot is switched off.
- **Never leave Hemisson switched on when not in use.**  
Once you have finished working with Hemisson, switch it off.

## 1.3 Recycling

You need to consider what to do with the equipment once it reaches the end of its useful life. Some parts can be recycled, such as the battery, which should not be thrown in the bin but put in the proper container. If you recycle it you will be contributing to making the environment cleaner and safer for future generations. For all these reasons, please ensure you deal appropriately with the equipment once it reaches the end of its useful life, for example either by returning it to the manufacturer or your local reseller.

<sup>1</sup>Question Forum

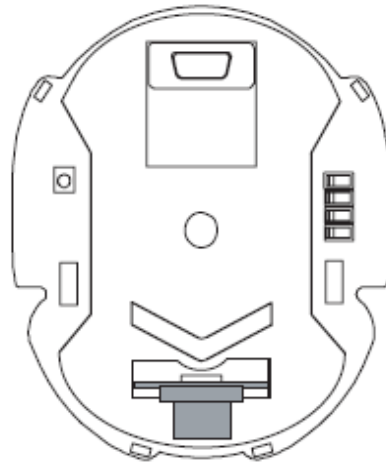
## 2 Connecting Up

---

*Fitting and removal of the extension module is a delicate operation. Please read the instructions below to prevent damage to it. K-TEAM cannot be held liable for any damage caused by improper handling.*

### 2.1 Fitting

To begin with, make sure the robot is switched off (position O); then insert the HemLinCam module (with its lens facing forward of the robot) without forcing it, but keeping it as vertical as possible, in the front connector on Hemisson, as follows (the module is in grey):



VIEW FROM ABOVE

Fig. 2.1 – Positioning the HemLinCam Module on Hemisson

*Under no circumstances try to insert HemLinCam in any of the other extension connectors on Hemisson, or with it facing the rear of the robot, as this might cause serious damage to your Hemisson robot as well as the extension module.*

## 2.2 Removal

To remove, first of all make sure that your robot is switched off (position O), then withdraw the HemLinCam module vertically with one hand, whilst holding the robot with the other.

## 3 The HemLinCam Module

---

### 3.1 Overall View

The HemLinCam module is built up as follows:

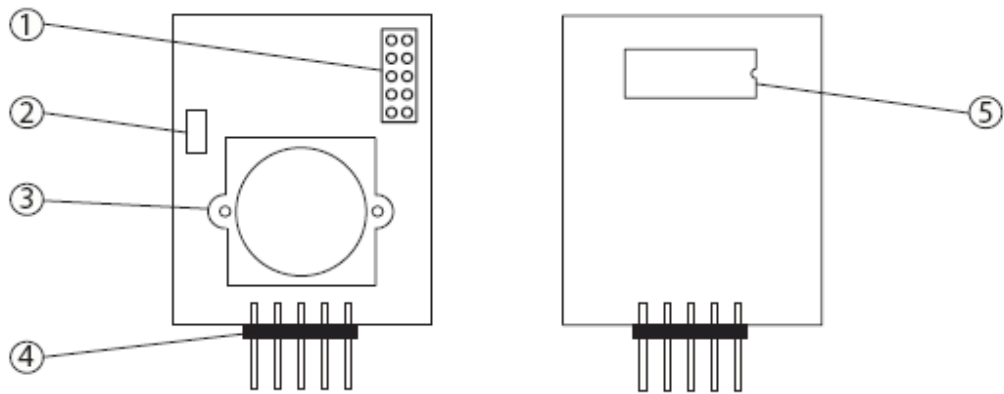


Fig. 3.1 – Detail of the HemLinCam Module - front and rear view

1. Reprogramming connector
2. LED
3. M12x0.5 lens
4. Main connector
5. Microprocessor

## 3.2 I2C Address

The Hemisson robot accesses the HemLinCam module via I2C communication. The module has a 7-bit address, plus 1 bit for selecting “read” or “write” mode.

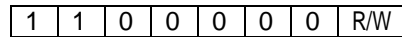


Fig. 3.2 – Module I2C Address

This therefore gives a “read” mode address of **0xC1**, and a “write” mode address of **0xC0**.

## 3.3 Register Addresses

To specify the camera parameters or to read the pixel values, you need to get “read” and “write” access to the registers listed below

*Throughout the following section we shall use notation as follows:*

- *R*: For a register that can only be accessed in “Read” mode.
- *W*: For a register that can only be accessed in “Write” mode.
- *R&W*: For a register that can be accessed in both “Read” and “Write” mode.

### 3.3.1 Firmware Register

**Description:** Enables reading of the firmware version of the HemLinCam module.

**Address:** 0 (0x00)

**Access:** R

### 3.3.2 ReadPixels Register

**Description:** Enables reading of the 102 pixel values. Each pixel has an unsigned 8-bit grey level value (0-255 or 0x00-0xFF).

**Address:** 16 (0x10)

**Access:** R

### 3.3.3 Read Thresholded Pixels Register

**Description:** Enables reading of the value of 102 pixels with thresholding. Each pixel has a binary value 0 (0x00) if it is below the threshold, or 255 (0xFF) if it is above.

**Address:** 17 (0x11)

**Access:** R



### **3.3.4 Threshold Value Register**

**Description:** Enables reading or writing of the threshold value that is to be applied in the thresholding function. This is an unsigned 8-bit value (0-255 or 0x00-0xFF).

**Address:** 32 (0x20)

**Access:** R&W

### **3.3.5 Exposure Time Register**

**Description:** Enables reading or writing of the exposure time value. This is an unsigned 8-bit value (0-255 or 0x00-0xFF). In practice we recommend use of values falling between 1 (0x01) and 10 (0x0A).

**Address:** 33 (0x21)

**Access:** R&W

### **3.3.6 LED Register**

**Description:** Enables specification of the LED state (on or off). A zero value (0 or 0x00) switches the LED off; any other value switches it on.

**Address:** 48 (0x30)

**Access:** W

## 4 Utilisation

---

There are two ways of using HemLinCam with Hemisson:

- C Programming
- Serial port control

### 4.1 C Programming

To program Hemisson in C, you first need to acquire the CCS compiler (available from the software section of the Hemisson website). We first recommend you read the latest version of the HemiOs manual (<http://www.hemisson.com/French/assistance.html>); in this document you will find all the information you need on the HemiOs as well as how to install CCS.

- Then download (<http://www.hemisson.com/French/assistance.html>) the latest available version of HemiOs.
- Open the project in CCS

The hemlincam.h (V1.1) library includes the following functions:

#### 4.1.1 Void HemLinCam Init(void)

Purpose: Initialisation of the camera; this is the first function to call at the start of the code.

Example:

```
HemLinCam_Init( );
```

#### 4.1.2 Char HemLinCam Read Version(void)

Purpose: Read the firmware version present in the camera.

Example:

```
char Version;  
Version = HemLinCam_Read_Version( );
```

#### 4.1.3 Void HemLinCam Set Threshold(unsigned char)

Purpose: Specify the threshold value.

Example:

```
HemLinCam_Set_Threshold(127);
```

#### **4.1.4 Unsigned Char HemLinCam Read Threshold(void)**

Purpose: Read the threshold value.

Example:

```
unsigned char Threshold_Value;  
Threshold_Value = HemLinCam_Read_Threshold( );
```

#### **4.1.5 Void HemLinCam Set Exposure Time(unsigned char)**

Purpose: Specify the exposure time.

Example:

```
HemLinCam_Set_Exposition(1);
```

#### **4.1.6 Unsigned Char HemLinCam Read Exposure Time(void)**

Purpose: Read the exposure time value.

Example:

```
char Exposition_Time;  
Exposition_Time = HemLinCam_Read_Exposition( );
```

#### **4.1.7 Void HemLinCam Read Pixels(void)**

Purpose: Read all pixel values. The values are then stored in the unsigned arrays - char HemLinCam Pixels Zone1, HemLinCam Pixels Zone2, HemLinCam Pixels Zone3. Each of these is a 34-pixel array, corresponding to the left hand, centre and right hand part of the image respectively.

Example:

```
HemLinCam_Read_Pixels( );  
if(HemLinCam_Pixels_Zone2[4]>50)  
{  
...  
}
```

#### **4.1.8 Void HemLinCam Read Pixels Thresholded(void)**

Purpose: Read all pixel values after thresholding. The values are then stored in the arrays listed above.

Example:

```
HemLinCam_Read_Pixels_Thresholded( );  
if(HemLinCam_Pixels_Zone1[4] == 255)  
{  
...  
}
```

### 4.1.9 Void HemLinCam Set LED State(char)

Purpose: Specify the LED state.

Example:

```
HemLinCam_Set_Led_State(1);
```

*NB: To use these functions, you must first prefix an include command for the library at the beginning of your code: #include "HemLinCam.h"*

## 4.2 Serial Port Control

Using the 1.30 version of the Hemisson firmware available from the website (<http://www.hemisson.com/French/assistance.html>) you can access I2C modules via RS-232 commands. To do this you should consult the latest version of the Hemisson user manual.

Examples:

- W,C0,30,01 : Switches LED on.
- R,C0,00,01 : Sends the firmware version.

## A Technical Specifications

---

- Dimensions [mm]: 42[L] x 35[W] x 26[H]
- Weight [g]: Not available
- Voltage [V]: 5
- Current [mA]: 10 (during image acquisition), 1 (on standby)
- Maximum I2C frequency [KHz]: 400
- Pixel number: 102
- Grey levels: 8-bit (0-255)
- Lens: M12x0.5 type
- Maximum framerate [Hz]: 100
- Focal length: See appendix

## B Connectors

---

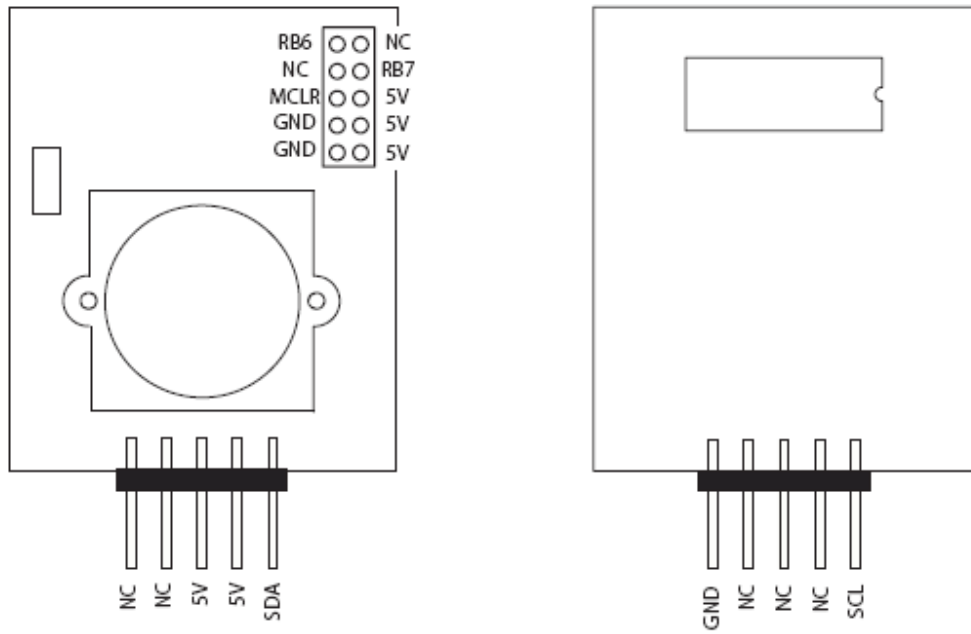


Fig. B.1 – Connector Details

---

K-Team SA  
Ch de Vuasset, CP 111  
1028 Préverenges  
Switzerland

---