

KJ-Lincam

user manual



version 1.0
JUNE 2010

Documentation Author

Frederic Lambercy
K-Team S.A.
Rue Galilee 9, Y-Park
1400 Yverdon-les-Bains
Switzerland

Email: info@k-team.com

Url: www.k-team.com

Trademark Acknowledgements:

IBM PC: International Business Machines Corp.

Macintosh: Apple Corp.

SUN Sparc-Station: SUN Microsystems Corp.

LabVIEW: National Instruments Corp.

Matlab: MathWorks Corp.

K-Junior: K-Team.

LEGAL NOTICE:

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

TABLE OF CONTENTS



1	INTRODUCTION	5
1.1	HOW TO USE THIS HANDBOOK	5
1.2	SAFETY PRECAUTIONS	5
1.3	RECYCLING	5
2	CONNECTION.....	6
2.1	INSTALL	6
2.2	UNINSTALL.....	6
3	THE K-JUNIOR LINCAM	7
3.1	OVERVIEW	7
3.2	I ² C ADDRESS.....	7
3.3	REGISTER ADDRESS	8
4	USE	10
4.1	C PROGRAMMING	10
4.2	SERIAL PORT CONTROL.....	12
A	TECHNICAL SPECIFICATION	13
B	CONNECTORS.....	14

1 INTRODUCTION



The K-Junior LinCam module allows you to add the dimension of vision to K-Junior. 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 handbook*

This manual introduces the K-Junior LinCam module dedicated to K-Junior robot. If the manual do not answer to a question, please consult K-Team website (www.k-team.com) and in particular the Forum and the FAQs

1.2 *Safety precautions*

Here are some recommendations on how to correctly use the K-Junior Robot and the LinCam:

- **Keep out from the wet places!** A contact with water can made a short circuit and damage the electronics.
- **Don't plug or unplug any connector or turret when the robot is powered!** All connections and turret insertions must be made when the robot and the interface are switched OFF. Otherwise damages can occur.
- **Never leave the K-Junior powered when it is unused.** When you have finished working with K-Junior, turn it off. It will save the battery life

1.3 *Recycling*

Think about the end of life of your robot! Parts of the robot can be recycled and it is important to do so. It is for instance important to keep batteries out of the solid waste stream. When you throw away a battery, it eventually ends up in a landfill or municipal incinerator. These batteries, which contain Lithium Polymer, can contribute to the toxicity levels of landfills or incinerator ash. By recycling the batteries through recycling programs, you can help to create a cleaner and safer environment for generations to come. For those reasons please take care to the recycling of your robot at the end of its life cycle, for instance sending back the robot to the manufacturer or to your local dealer.

Thanks for your contribution to a cleaner environment!

2 Connection



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 a wrong manipulation.

2.1 Install

To begin with, make sure the robot is switched off; then insert the K-Junior LinCam module keeping it as vertical as possible, but making sure it is positioned as below:



Figure 2.1: K-Junior LinCam module on the K-Junior

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

2.2 Uninstall

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

3 The K-Junior LinCam



3.1 Overview

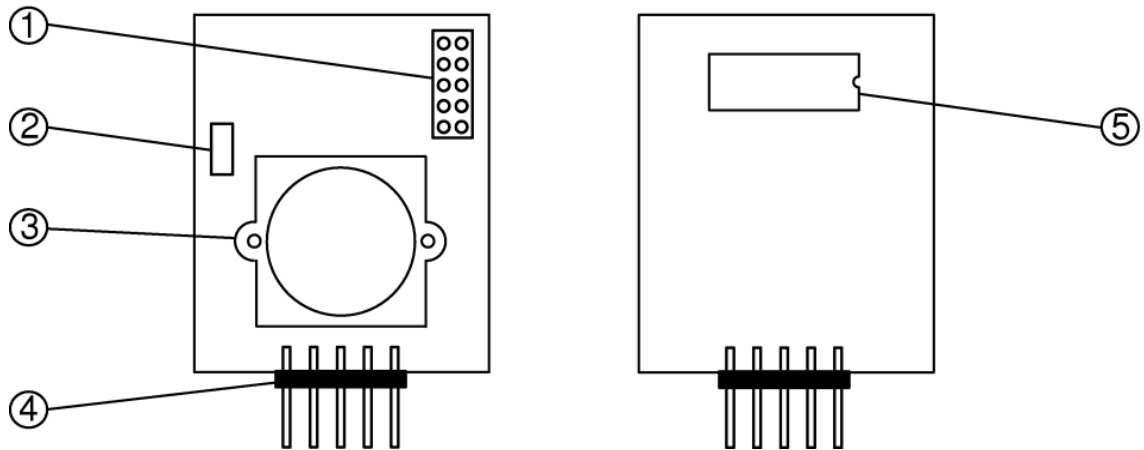


Figure 3.1: Overview of the LinCam turret layout, Front and Rear view

Make an external inspection of the turret. Note the location of the following parts:

1. Reprogramming connector
2. Led
3. M12x0.5 Lens
4. Main Connector
5. Microcontroller

3.2 I²C Address

The K-Junior Robot control the LinCam through an I²C bus. The turret address is defined by 7 bits + 1 bit for the mode selection "write" or "read"

1	1	0	0	0	0	0	R/W
---	---	---	---	---	---	---	-----

The LinCam I²C address is 0xC1 in "read" mode and 0xC0 in "write mode".

3.3 Register Address

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 K-Junior LinCam 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



There are two ways of using K-Junior LinCam with K-Junior:

- C Programming
- Serial port control

4.1 C Programming

To program K-Junior in C, you must acquire CCS C compiler (<http://www.k-team.com/mobile-robotics-products/k-junior/software#acp>). We recommend that you first read the latest version of the KJOs manual (<http://ftp.k-team.com/K-Junior/KJOSManual.doc>). This manual provides all necessary information about installing and using CCS on K-Junior modules.

- Download the latest version of KJOs, as well as the latest version of K-Junior LinCam Lib including HemLinCam.h.
- Open the project using CCS C Compiler

To use the following functions, you must first include HemLinCam.h at the beginning of your code: `#include "HemLinCam.h"`, the HemLinCam.h file must be in your project's directory. HemLinCam.h 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

Thanks to the K-Junior firmware you can access to I²C modules via the RS232 command line. For that, please refer to the latest K-Junior user manual.

Examples :

- W,C0,30,01 : Turn on the Led
- R,C0,00,01 : returns firmware version.

You can use any development environment that can access serial port to interface to K-Junior LinCam. For instance, we developed a graphical interface to read and configure the K-Junior LinCam. You must first plug your K-Junior LinCam to the K-Junior, turn on the robot and then connect the PC to the K-Junior with the USB cable.

A Technical Specification



- Dimensions [mm]: 42(L) x 35(W) x 26(H)
- Weight : NC
- Power Supply : 5 [V]
- Current[mA]: 10 (during image acquisition), 1 (on standby)
- Maximum I²C frequency: 400[kHz]
- Pixels number: 102
- Grey levels: 8 bits (0-255)
- Lens: M12 x 0.5 type
- Maximum frame rate: 100 [Hz]

B Connectors

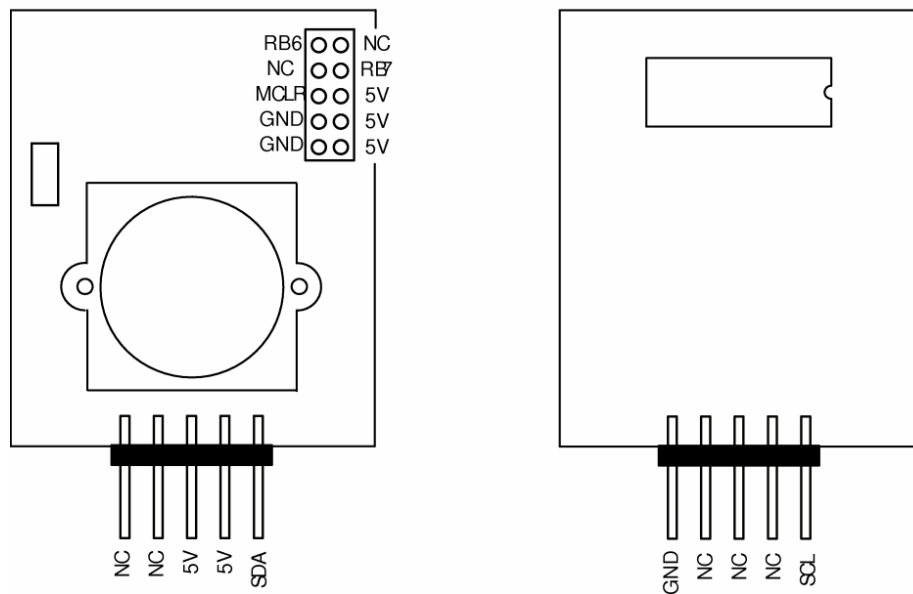
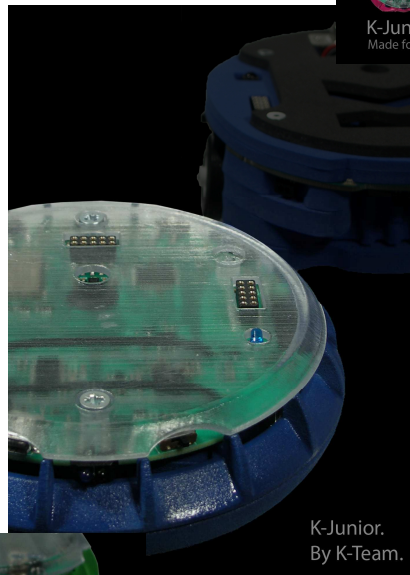
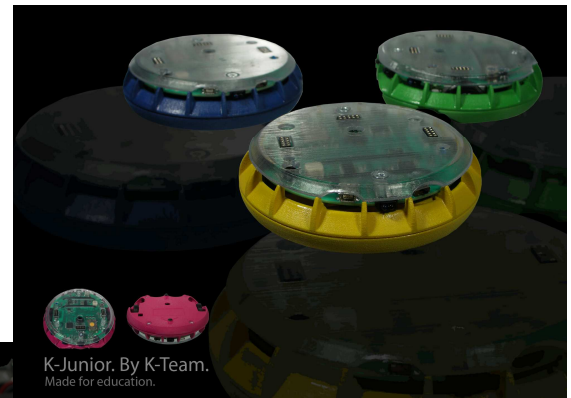


Figure B.1: Connector Details



K-Team S.A.
RUE GALILEE 9
1400 YVERDON-LES-BAINS
SWITZERLAND