Kemro K2

KeMotion
Vision
User manual 2.40

KeMotion tracking technology with vision system

KEBA®
Automation by innovation.

Document : 2.40 / article no.: 1008373
Filename : KeMotion_Vision_UserManual_en.pdf
Pages : 33

A: KEBA AG, Gewerbepark Urfahr, A-4041 Linz, Tel.: +43 732 7090-0, Fax: +43 732 7309-10, E-Mail: keba@keba.com

D: KEBA GmbH Automation, Leonard-Weiss-Straße 40, D-73037 Göppingen, Tel.: +49 7161 9741-0, Fax: +49 7161 9741-40, E-Mail: keba@keba.com

US: KEBA Corp., 100 West Big Beaver Road, Troy, MI 48084, US, Tel.: +1 248 526-0561, Fax: +1 248 526-0562, E-Mail: usa@keba.com

CN: Beijing Austrian KEBA Science and Technology Development Ltd., Room B516, Nan Xin Cang Tower, A22 Dong Si Shi Tiao, Dong Cheng District, Beijing, 100027, P.R. China, Tel. +86 10 6409-6592, Fax +86 10 6409-6312, E-Mail: china@keba.com

www.keba.com

Record of Revision

Version	Date	Change in chapter	Description	changed by
2.30	01-2010		created new	mhl
2.40	05-2011		add generic TCP camera	pen

Table of Contents

1 Introduction

1.1 Purpose of the document

This document describes how to combine a vision system with the KeMotion tracking technology.

1.2 Target group, pre-requirements

This document is made for the following persons with adequate skill pre-requirements:

Target group	Assumed knowledge
Project engineer	Technical education (Technical college, engineer or corresponding experience). Knowledge of ● the working principles of a PLC ● health and safety regulations ● how to program a PLC (IEC61131)
Programmer	Technical education (Technical college, engineer or corresponding experience). Knowledge of ● the working principles of a PLC ● health and safety regulations ● how to program a PLC (IEC61131) ● the basics of robotics
Service technician	Technical education (Technical college, engineer or corresponding experience). ● Knowledge of ● the working principles of a PLC ● health and safety regulations ● how to program a PLC (IEC61131)

1.3 Intended use

This software has been developed to control industrial robots and electrical servo drives. It must not be used for applications other than those described in this document, and it must exclusively be used for recommended or approved target systems.

WARNING!

1) **Kemro K2 KeMotion rSeries has not been designed for safety-relevant control applications such as stopping in the case of emergency or safely reduced speed.**
2) **Kemro K2 KeMotion rSeries complies only with safety category B PL a according to EN ISO 13849-1. It is therefore inadequate for implementation of safety functions to protect humans.**
3) **For safety-relevant control applications or personnel protection additional external protection measures must be taken, ensuring a safe operation condition also in case of failures.**

1.4 Notes on this document

This manual is part of the product. It must be retained over the whole lifetime and if necessary referred to subsequent owners or users of the product.

1.4.1 Contents of document

- How to handle the data from the vision system in the PLC to operate with the KeMotion tracking technology package.
- Usage of the vision system driver in the PLC.
- Software data flow overview.
- Description of the vision system activities and hardware connection.

1.4.2 Not contained in this document

- PLC Programming
- Basics of robotics
- Basics of tracking technology

1.5 Documentation for further reading

Knowledge of the following documents is assumed:

No.	Title	Target group
	PLC Robotic Programming Manual	• Project engineer • Programmer • Service technician
	Basics of Robotics	• Robot operator • Project engineer • Programmer • Service technician

	KAIRO Language Reference	• Robot operator • Project engineer • Programmer • Service technician
	Tracking User Manual	• Project engineer • Programmer • Service technician

All documents listed here may be ordered from KEBA.

2 Safety notes

2.1 Representation

At various points in this manual you will see notes and precautionary warnings regarding possible hazards. The symbols used have the following meaning:

DANGER!

- **indicates an imminently hazardous situation which will result in death or serious bodily injury if the corresponding precautions are not taken.**

WARNING!

- **indicates a potentially hazardous situation which can result in death or serious bodily injury if the corresponding precautions are not taken.**

CAUTION!

- **means that if the corresponding safety measures are not taken, a potentially hazardous situation can occur that may result in property injury or slight bodily injury.**

CAUTION

- **CAUTION used without the safety alert symbol indicates a potentially hazardous situation which, if not avoided, may result in damage to property.**

- **This symbol reminds you of the possible consequences of touching electrostatically sensitive components.**

Information

Useful practical tips and information
on the use of equipment are identified by the “Information” symbol. They do not contain any information that warns about potentially dangerous or harmful functions.

3 Hardware concept

This chapter explains the hardware wiring of a typically tracking application with a conveyor belt and a vision system.

3.1 Connection diagram

The following diagram shows the hardware connections to each important component which is needed for tracking with vision.

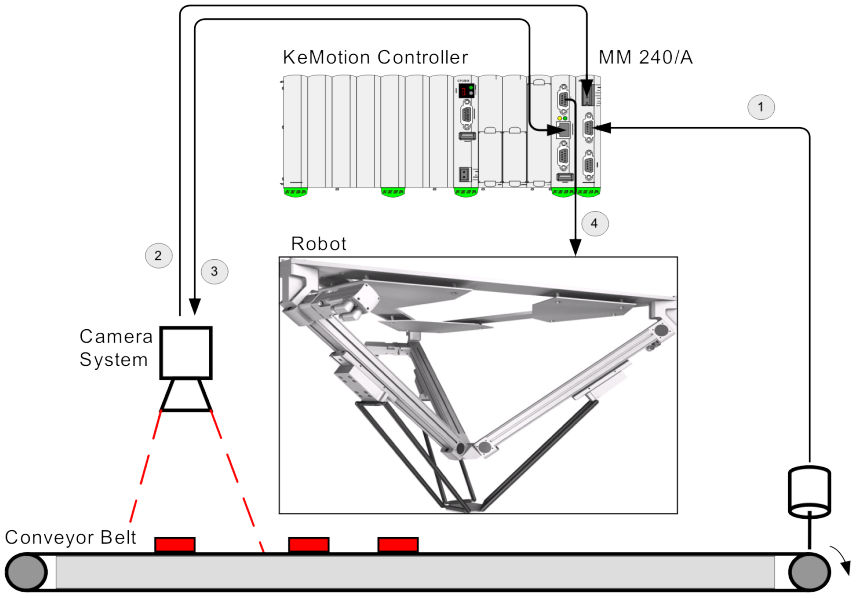


Fig.3-1: Connection diagram of tracking system with vision.

Description of connections

1) The incremental encoder is wired to the incremental encoder input of the MM 240/A module to provide the position of the conveyor belt to the controller.
2) The latch-input of the MM 240/A is wired to the digital output of the vision system. Every picture, the vision system raises this output and executes a latch interrupt event on the controller. Due to this latch event information the controller can link the position information from the vision system with the exact encoder position and can predict the further positions of the objects. Without this link, the controller would not know where or when to grab.
3) Fieldbus communication connection to the vision system
4) The controller uses the provided information from the vision system and the incremental encoder to drive the robot via fieldbus.

3.2 Compatibility of components

Robot

The KeMotion tracking technology works robot independent. So any supported kinematic type can be used.

Vision system

Following bus systems can be used to connect a vision system to a KeMotion control system:

- Ethernet
- Profibus
- Serial communication (RS232, RS422, RS485)

The required vision system operation performance depends on the tracking application.

Encoder

Incremental encoders with the following output voltages can be used:

- +/- 5 V differential (RS422)
- + 24 V single ended (HTL)

An analog output of 1V sine (peak-to-peak) is not supported. For further information on supported encoders and wiring see also the documentation of the MM 240/A encoder module.

Information

The required encoder resolution depends on the coupling of the encoder to the conveyor belt. It is recommended to use an encoder resolution such that the resulting resolution of the conveyor belt is at least 500 increments per mm. Even if most times the robot is mechanically not able to support that high accuracy of 1/500 mm, the internal processing of the encoder signals needs it to produce good results.

4 Vision system

In this chapter the vision system usage and the tasks are described which have to be done in the vision system.

4.1 Operation tasks

The vision system has to do several tasks to operate correctly with the KeMotion tracking technology package.

- The vision system has to detect and transfer all parts in the field of view (FOV).
- The vision system should send the the position of the part where the robot will take or process it (grip point). For gripping this point is typically the center of gravity.
- The position of the part (x, y, z) must be given in [mm] and not in pixels. The orientation of the part (a, b, c) must be given in [°](degree). Typically only the values of ‘x’, ‘y’, and ‘a’ are written from the vision system, all others are zero.
- The vision system has to send an impulse on a digital output when it takes the picture. This signal is wired to the latch input of the encoder module MM 240/A. Typically the digital output of the external photoflash lamp is used. This information is necessary to get the exact encoder position in the moment when the picture was taken.
- Depending on the application, the vision system can send an attribute value (DWORD) for each part. This value is a customer defined bit field in which the customer can specify bits for process necessary information which comes from the vision system. Example: Color of part, good or bad part, type of part, ...

  This attribute can be defined for each part and is accessible in the PLC program and in the KAIRO program. Depending on this attribute the PLC can handle the parts different and also the robot can decide what to do with this part.

  If one part is detected on more then one consecutive pictures (because the conveyor belt is very slow) then the attribute for this part must always be the same.

4.2 Vision system settings

4.2.1 Coordinate system of the vision system

It is important to setup the vision system coordinate system so that the positive x-direction of the coordinate system exactly fits with the conveyor belt moving direction.

Information

Most vision system software also includes a self calibrating mechanism with a calibration pattern sheet. The automatically calibration sheet can be placed on the conveyor belt and has to be aligned exactly on the conveyor belt.

It could also be helpful to mark the origin of the coordinate system of the vision on a static place (e.g. conveyor mounting) for further measurement with the robot tracking reference system.

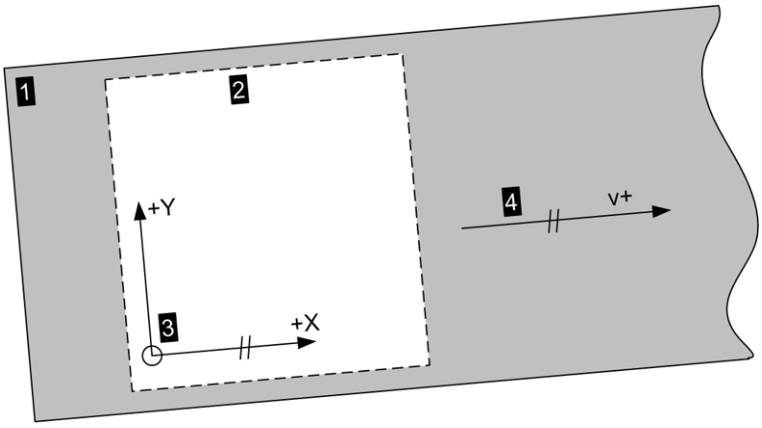


Fig.4-1: example illustration of the coordinate system of the vision system

1 ... Conveyor belt	2 ... Field of view (FOV) of vision system
3 ... Coordinate system of vision system	4 ... Conveyor belt moving direction (positive velocity).

4.2.2 Vision system picture update rate

The framerate (Frames per second, FPS) which should be executed from the vision system depends on the size of the field of view (FOV) and the conveyor belt velocity. Most time it is not necessary to work with the highest possible update rate the vision system supports. It is also not important to trigger the vision system pictures in an update rate so that every part on the conveyor is only seen one times. A good update rate of the vision system is reached when every part on the conveyor is detected 2 up to 3 times. This is because of stabilization reasons, if the vision system does not detect a part one time in a unlikely event. The update rate (FPS) is calculated as shown below:

$$update\_rate[frames/\sec] = \frac{Conveyor\_belt\_velocity[mm/\sec]}{Field\_of\_view\_of\_camera[mm]} \cdot 2,5$$

Fig.4-2: formula how to get the frames per second value of the vision system

Example

The maximum velocity of the conveyor belt is 1m/sec and the size of the vision systems filed of view is 0.5m in conveyor belt moving direction.

(1000 / 500) * 2,5 = 5

The vision system should operate with a frame rate of 5 FPS.

4.2.3 Vision system data transfer timing

To ensure that the received parts from the vision system matches the correct encoder latch data, it is important that the vision system keeps a timing prevention. A next picture can't be triggered before the last part of the picture was transferred to the controller.

It is not allowed, that the vision system takes a next picture during sending the parts to the controller, because this will raise a next latch data from the encoder module. The received parts always refer to the latest occurred encoder latch data. So the controller will use an incorrect latched encoder position for the received parts if the next picture is taken during sending the parts.

In applications, in which the trigger signal to take a next picture is sent from the controller to the vision system, this timing behavior can easily be fulfilled.

5 Software concept

In this chapter the software handling of the objects and the communication to the vision system is described.

5.1 Dataflow diagram

The handling of the incoming data from the vision system containing the object information (position and attribute) must be done in the KeMotion PLC.

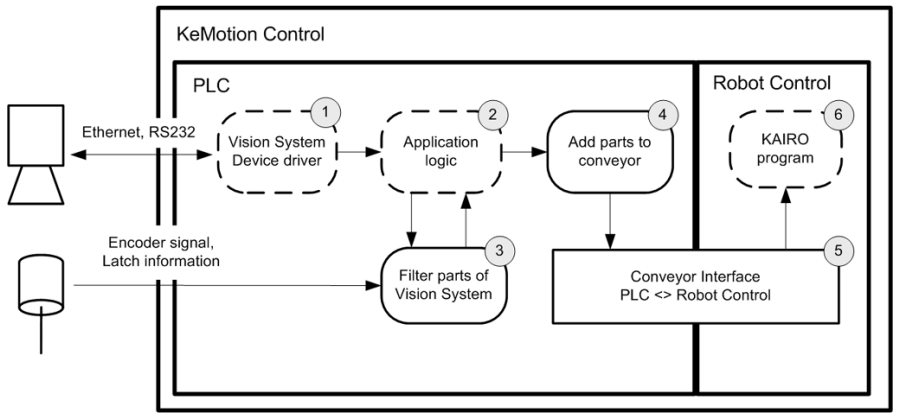


Fig.5-1: software concept

Description of software concept

- The vision system device driver is a function block which operates with the used vision system type. This driver block creates a connection to the vision system using e.g. Ethernet or serial RS232. The driver analysis the received data stream from the vision system and provide the data in lists. The vision system objects have information of their position, attributes and an ID (identification number).
- The application logic uses the received objects. Its possible to filter or split the objects out depending on the objects attribute or separate them to different robots.
- The function block for filtering the double detected objects is provided in the 'RcTracking.lib'. The name is "RCTR_FilterObjects". (See also in Tracking User Manual.) This function block detects objects from the vision system which are seen more than one times and filter them out. Additionally the latest occured latched encoder position from the MM 240/A encoder module is added to the object information which is important to get the correct position of the objects on the conveyor. The output of this filter objects function block are the new detected objects on the conveyor.
- The function block `RCTR_AddObject` adds objects to the specified conveyor interface object list. This function is in the library RcTracking.lib. (See also in Tracking User Manual.)

It is not possible to add more than one object with this function, so this function has to be called for each detected object which should be added in a conveyor object list.

- The conveyor interface contains a object list which the robot can access in the KAIRO program. The conveyor interface also contains information about the actual encoder value and its resolution.
- In the KAIRO program the robot can fetch the objects from the conveyor object list. The program gets an object which represents a object on the conveyor belt. The robot is able to move to this object and process them. After processing it mark the object as done. This will cause that this object is removed again from the objectlist.

Information

To filter a object in the PLC out there is no need to sparately kill or delete a object. Just do not add the object to the conveyor interface objectlist.

5.2 Vision system communication

The tasks of the vision system driver depends on the vision system used. The following tasks have to be performed by the driver.

- Conect to the vision system
- Select the detection program of the vision system.
- Periodically trigger the image acquisition.
- Receive the data from the camera and fill a list with the object data.
- Track the state of the camera (Errors,...)
- Optionally: Change some customer specified parameters for picture analysing. (e.g. tolerance values, ...)

5.3 Processing the objects from the vision system

The vision system usually sends all detected objects in the field of view. It is possible that one element on the conveyor is seen and sent twice or more times.

The function block "RCTR_FilterObject" can be used to filter the objects automatically. Further this function block also adds the necessary latched encoder information to the objects. See also in Tracking User Manual. Out coming objects are only the new objects on the conveyor. The objects are now deterministic.

The application logic can now handle the objects. It is possible that the objects are separated into more than one robot in case of a multi-kinematic system, so that each robot has to workout the same amount of objects.

The application logic can throw objects away depending on the attribute value of the object. For example if the vision system sends also a criterion in the

attribute if it is good or bad. Additionally it is also possible to modify or create a own attribute of the object.

For diagnostics the objects can be supplied with an identification number. This number will be used in warning and error messages and is also used and displayed at the conveyor object logger.

6 Configuration of a camera

This chapter descibes the steps to configure a camera in the system. The following camera types are supported:

- Cognex Insight
- Generic TCP

Configure PLC

Attach a new element "Vision" in the PLC configuration.

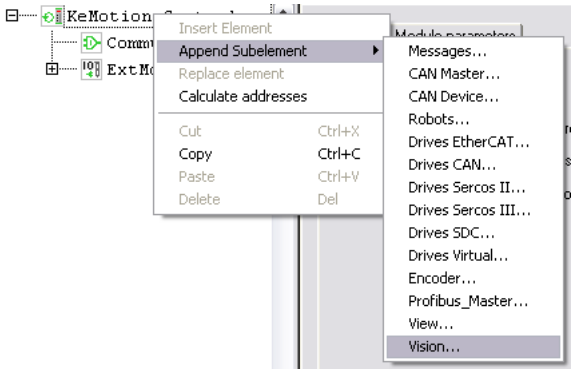


Fig.6-1: Attach device

Append a new element for each camera.

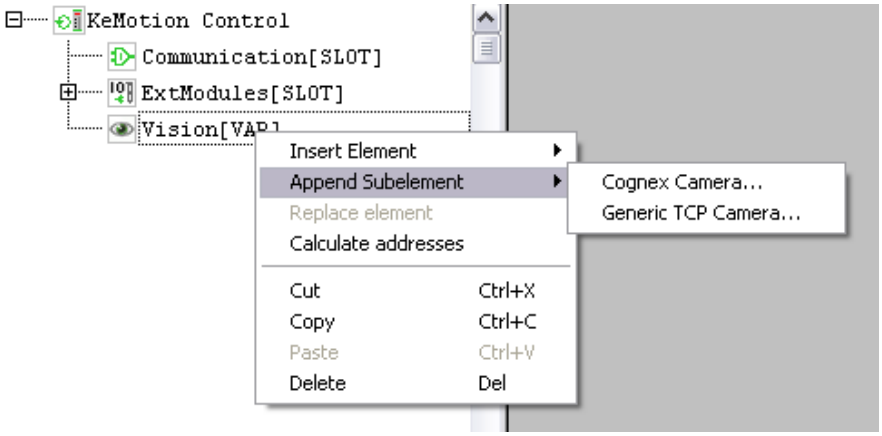


Fig.6-2: Attach camera

IO Configuration

Configure a digital high-speed output (discrete output on the camera, not an output which is on an ethernet connected io-module). Use the type “Acquisition Start” with a pulse length of 5-10ms. Connect this output to the latch input of the KeMotion encoder module MM240/A. This signal is necessary to latch the actual encoder position when the image is acquired.

IEC libraries

To use cameras in an IEC-project the libraries KVision.lib and KVisionUtils.lib are needed.

6.1 Cognex Insight

To use a Cognex Insight camera in the system, the following steps are necessary:

- Install Kemro Automation Snippet
- Configure PLC
- Configure camera
- Add Kemro Automation Snippet to the job

Install Kemro Automation Snippet

Copy the file Kemro_Automation.cxd from the delivered DVD (directory: target/vision) into the Snippet-directory of the Cognex installation (f.e. `C:\Program Files\Cognex\In-Sight\In-Sight Explorer 4.4.0\Snippets`).

Configure PLC

Configure the PLC as described previously, choose "Cognex Camera" for the camera type.

Enter the necessary parameter for the camera.

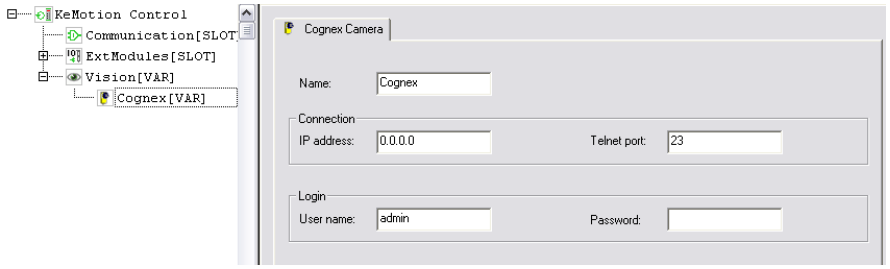


Fig.6-3: Parameter

Entry	Description
Name	The name of the camera. A variable with this name is instantiated in the PLC project and therefore it must be a unique variable name.
IP adress	The IP adress of the camera.
Telnet port	The telnet port as it is configured on the camera.
User name	The user name to login to the camera. This user must have full access to the camera.
Password	The password for the user.

Configure camera

In the Insight Explorer select the menu entry ***Sensor*** ▸ ***Startup...***.

Enable the online-checkbox in the startup and select a job. Do not select <new> because in this case the PLC detects an unsaved job after restart of the camera.

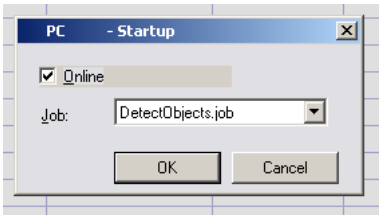


Fig.6-4: Cognex Insight Start-up dialog.

In Property sheet of the "AquireImage cell" (double click on cell A0) set the Trigger to External and the Buffer Mode to Single.

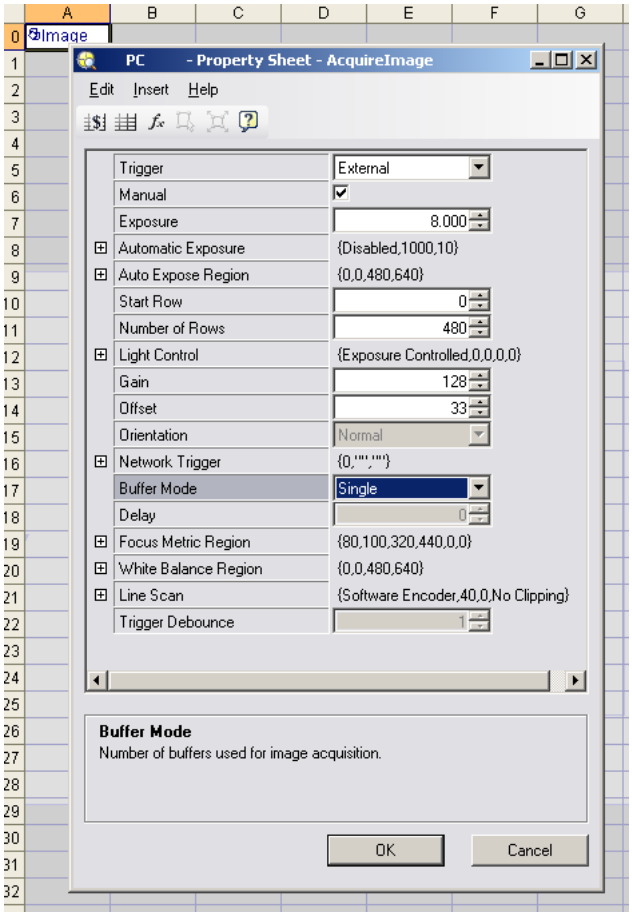
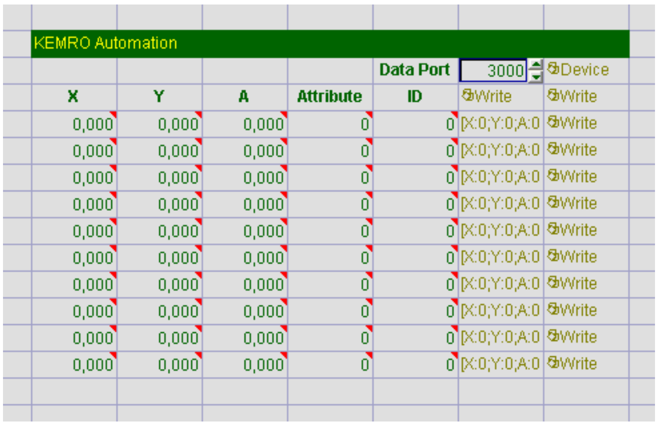


Fig.6-5: AcquireImage cnfiguration

In the Cognex Explorer select the menu entry ***Sensor ▸ User List...*** to open the users dialog. Ensure the user for the PLC has full access rights to the camera.

Spreadsheet Snippet

For the communication with the KeMotion controller you must include the KeMotion snippet in the job.



KEMRO Automation						
				Data Port	3000	Device
X	Y	A	Attribute	ID	Write	Write
0,000	0,000	0,000	0	0	[X:0;Y:0;A:0	Write
0,000	0,000	0,000	0	0	[X:0;Y:0;A:0	Write
0,000	0,000	0,000	0	0	[X:0;Y:0;A:0	Write
0,000	0,000	0,000	0	0	[X:0;Y:0;A:0	Write
0,000	0,000	0,000	0	0	[X:0;Y:0;A:0	Write
0,000	0,000	0,000	0	0	[X:0;Y:0;A:0	Write
0,000	0,000	0,000	0	0	[X:0;Y:0;A:0	Write
0,000	0,000	0,000	0	0	[X:0;Y:0;A:0	Write
0,000	0,000	0,000	0	0	[X:0;Y:0;A:0	Write
0,000	0,000	0,000	0	0	[X:0;Y:0;A:0	Write

Fig.6-6: The KeMotion spreadsheet snippet.

The snippet contains ten lines for sending object data to the controller, one line per object. If you need less lines just delete them from the end of the snippet. If you need more just copy them to the free space below the snippet.

You must link the cells in five columns (X, Y, A, Attribute and ID) on the left side of the snippet to the cells that contain the appropriate object data.

The data port is the port of the camera to which the controller connects for receiving the object data. You can change the data port in the first line of the snippet.

When one of the first five columns contains an error (#ERR) then this line is not transferred to the PLC. Thus if a tool detects less than the maximal number of parts then the detected parts are sent to the camera only.

6.2 Generic TCP

To use a generic TCP camera in the system, the following steps are necessary:

- Configure PLC
- Configure camera

Configure PLC

Configure the PLC as described previously, choose "Generic TCP Camera" for the camera type.

Enter the necessary parameter for the camera.

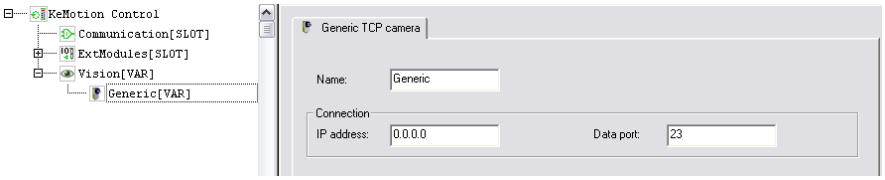


Fig.6-7: Parameter

Entry	Description
Name	The name of the camera. A variable with this name is instantiated in the PLC project and therefore it must be a unique variable name.
IP adress	The IP adress of the camera.
Data port	The data port as it is configured on the camera.

7 Camera Drivers

This Chapter describes the camera drivers that are included in the KeMotion targets.

7.1 Datatypes for camera interaction (KVisionUtils.lib)

KVisionUtils.lib offers necessary data types for an interaction with a camera. The following types are available:

Name	Description
TKVIS_CamName	Name of the camera
TKVIS_CartFrame	Cartesian frame with six coordinates for position and orientation of the object Kartesischer Frame mit 6 Koordinaten für Position und Orientierung des Objekts
TKVIS_Object	Object with cartesian frame, ID and attribut
TKVIS_ObjectList	List (fixed length) of TKVIS_Objects and numeric value (integer), how much strings are in the list
TKVIS_LineList	List of strings (fixed length) and numeric value, how much strings are in the list
TKVIS_DiagnosisData	Diagnostic data
TKVIS_Diagnosis	Diagnostic settings

7.2 Camera Drivers (KVision.lib)

The library KVision.lib provides functions to interact with a camera (e.g. Connection to PLC or triggering a picture).

A camera excecutes its program and sends the results as strings over the data port to the PLC. Image results have to start with the string 'Image' and end with the string 'Done'. Other kinds of results have to start with the string 'Result' and end with the string 'Done'. The camera driver installed on the PLC analyzes the strings. If a string within an image result has a valid object format, an object of the type TKVIS_Object is generated and added to the object-list (TKVIS_ObjectList). Otherwise the string remains unchanged and is added in the string-list (TKVIS_LineList). In this case the application decides about the further processing. Objects of the object-list can be used by the RCTracking.lib or directly by the application.

Both lists are valid for one update-cylce of the IEC-program. These lists are overwritten every cyle. Result acquisition can take more than one cycle. After completing a result acquisition the appropriate return value ('Done' or 'Image-Done') of the camera driver is set to TRUE for one cycle. After this the next acquisition starts and the return value changes to FALSE.

KVision.lib provides own data types and function blocks (camera driver) for each supported camera type.

The following camera types are supported:

- Cognex
- Generic TCP

7.2.1 Cognex

The table below provides a list of all data types and function blocks:

Name	Type	Description
CAM_COGNEX_REF	STRUCT	Data structure of Cognex camera
KVIS_CgxClient	Function block	Camera driver for Cognex Insight camera
TKVIS_CgxJobName	STRING	Name of Cognex-job

CAM_COGNEX_REF

For every configured cognex camera a global variable of type CAM_COGNEX_REF is automatically generated. Its name is identical to the name given in the PLC configuration. This variable is needed for a call to the function block KVIS_CgxClient. It also contains a member called `Diagnosis` of type TKVIS_Diagnosis that contains useful information fort the programmer.

KVIS_CgxClient

This function block contains the camera driver. Cognex cameras transmit solely images over the data port, therefore the corresponding return value is called 'ImageDone'.

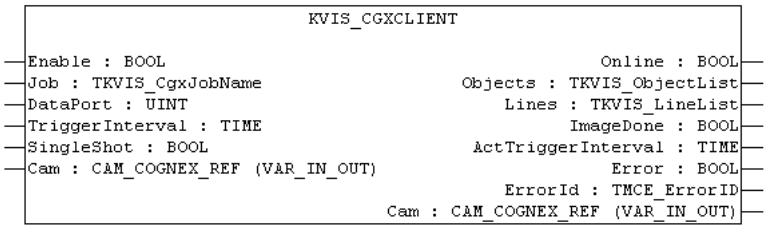


Fig.7-1: Function block

Description of the data types:

VAR_INPUT

Name	Type	Description
Enable	BOOL	Upon a rising edge the driver connects to the camera, sets the job and starts with image acquisition.
Job	TKVIS_CgxJobName	The name of the job that must be set on the camera before going online.
DataPort	UINT	Data port (as specified in the snippet)

Name	Type	Description
TriggerInterval	TIME	The minimal interval between two image acquisitions. If the processing of the image needs longer then the actual trigger interval is not relevant. With the value T#0ms the automatic trigger is deactivated. This value can be changed all the time.
SingleShot	BOOL	Upon a rising edge a single image is triggered.

VAR_IN_OUT

Name	Type	Description
Cam	CAM_COGNEX_REF	Reference to the camera

VAR_OUTPUT

Name	Type	Description
Online	BOOL	TRUE if the camera is connected and ready to acquire images.
Objects	TKVIS_ObjectList	A list containing all objects that were received from the camera in this update cycle. This output can directly be connected to the object input of the function block RCTR_FilterObjects
Lines	TKVIS_LineList	A list containing all lines that were received from the camera in this update cycle which were not converted to objects.
ImageDone	BOOL	When the last result string for an image is received from the camera this output is set to TRUE for one cycle. In the same cycle the last results may be available in the lists.
ActTriggerInterval	TIME	The actual executed trigger interval.
Error	BOOL	Indicates an error
ErrorID	TMCE_ErrorID	Detailed error description

7.2.2 Generic TCP

The table below provides a list of all data types and function blocks:

Name	Type	Description
CAM_GENTCP_REF	STRUCT	Data structure of a generic TCP camera
KVIS_TCPClient	Function block	Camera driver for a generic TCP camera
TKVIS_TcpCommand	STRING	Command for the camera

CAM_GENTCP_REF

For every configured TCP camera a global variable of type CAM_GENTCP_REF is automatically generated. Its name is identical to the name given in the PLC configuration. This variable is needed for a call to the

function block KVIS_TCPClient. It also contains a member called `Diagnosis` of type TKVIS_Diagnosis that contains useful information fort the programmer.

KVIS_TCPClient

This function block contains the camera driver.

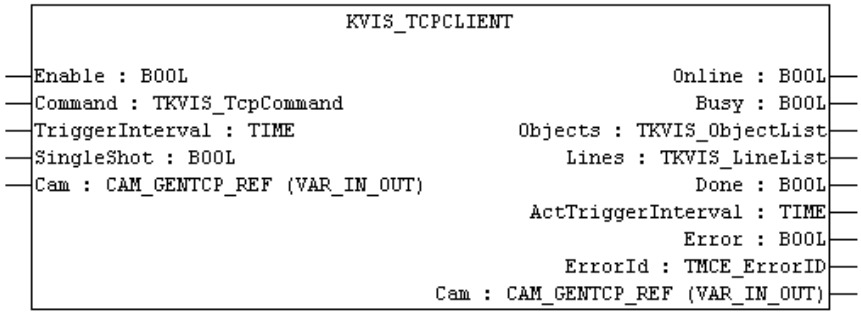


Fig.7-2: Function block

Description of the data types:

VAR_INPUT

Name	Type	Description
Enable	BOOL	Connect to the camera.
Command	TKVIS_TcpCommand	Command that should be sent to the camera.
TriggerInterval	TIME	The minimal interval between two transmissions of the command. If the processing of the command lasts longer then the next command is started immediately after completion. With the value T#0ms the automatic trigger is deactivated. The value can be changed at any time.
SingleShot	BOOL	Upon a rising edge a single command is triggered.

VAR_IN_OUT

Name	Type	Description
Cam	CAM_GENTCP_REF	Reference to the camera

VAR_OUTPUT

Name	Type	Description
Online	BOOL	TRUE if the camera is connected.
Busy	BOOL	TRUE as long as the function block needs to be maintained.

Name	Type	Description
Objects	TKVIS_Object-List	A list containing all objects that were received from the camera in this update cycle. This output can directly be connected to the object input of the function block RCTR_FilterObjects
Lines	TKVIS_LineList	A list containing all lines that were received from the camera in this update cycle which were not converted to objects.
Done	BOOL	When the last result string for an command is received from the camera this output is set to TRUE for one cycle. In the same cycle the last results may be available in the lists.
ActTriggerInterval	TIME	The actual executed trigger interval.
Error	BOOL	Indicates an error
ErrorID	TMCE_ErrorID	Detailed error description

8 Diagnosis

In this chapter some possibilities for the diagnosis of the vision components are described.

8.1 String format

The object strings that are generated by the camera and sent to the PLC contain tagged values in square brackets, separated by semicolons. A complete object string is for example `[X:<x>;Y:<y>;A:<a>;ATTR:<attr>;ID:<id>]`, where the parts in angle brackets are the actual values (<x>: x-coordiante, <y>: y-coordinate, <a>: angle, <attr>: user defined attribute, <id>: user defined id). When one of the tagged values is missing then the parser sets it to 0. The order of the tagged values does not matter. Thus `[ID:<id>;X:<x>;Y:<y>]` would also be a valid object string.

8.2 PLC camera driver

The variable that is instantiated for a configured camera contains a member named `Diagnosis` of type `TKVIS_Diagnosis`. This member contains information about the current state of the camera driver. When you debug your PLC application you can see the following information in the diagnosis struct.

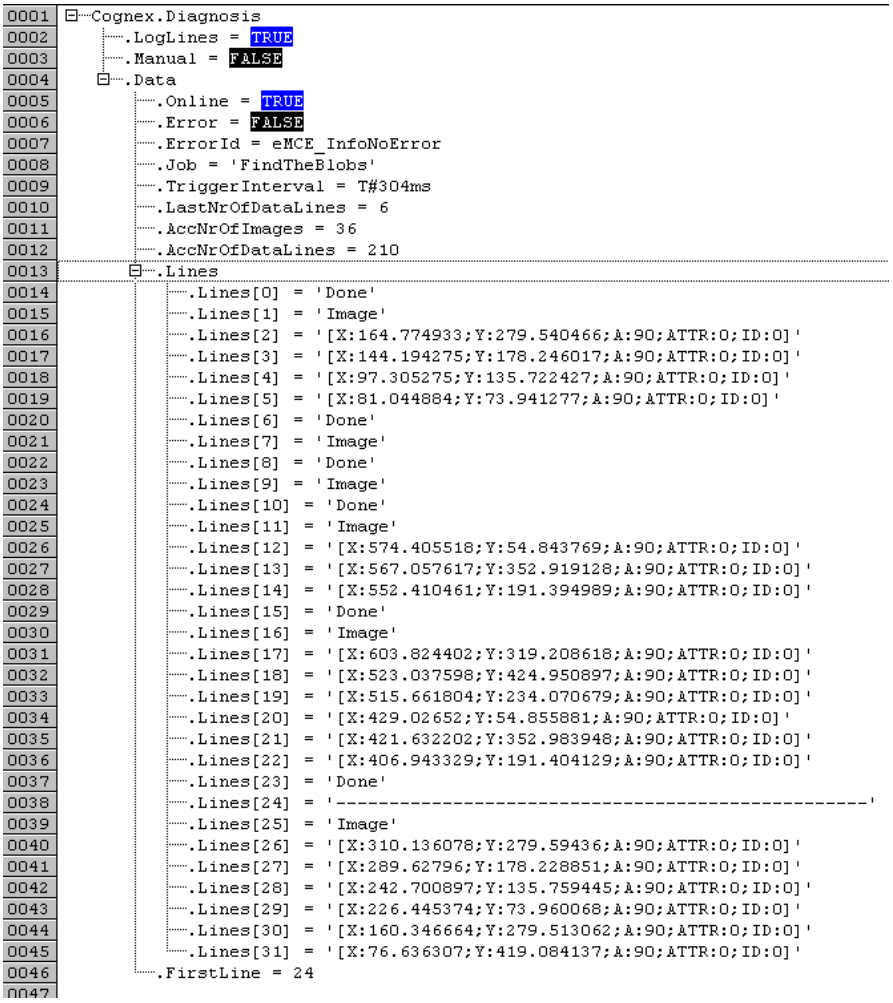


Fig.8-1: The vision diagnosis struct.

Manual, LogLines, Data.Lines

`Data.Lines` is a ring buffer of length `gcKVIS_DiagnosisLineBufLen` that containes all the lines that where received by the camera driver. In the diagnosis buffer the lines are truncated to a length of `gcKVIS_DiagnosisLineLen`. A line of dashes is inserted at the end of the ring buffer. If you set `Manual` to FALSE then every image is logged into the buffer. If you set `Manual` to TRUE then you can trigger the logging of one image by setting `LogLines` to TRUE. `Manual` affects logging only and has no influence on the camera driver.

Data.Online, Data.Error, Data.ErrorId, Data.Job, Data.TriggerInterval

Those members reflect the inputs and outputs of the camera driver function block. `Data.TriggerInterval` is the actual trigger interval.

Data.LastNrOfDataLines

The numer of data lines from the last image that was logged into the ring buffer, not counting control lines like `Image` or `Done`.

Data.AccNrOfDataLines, Data.AccNrOfImages

The accumulated number of data lines and images. These values also count the images that where not logged into the ring buffer.

Index