

Kemro K2

KeMotion Vision User manual 2.40

KeMotion tracking technology with vision system



Automation by innovation.

Dokument : 2.40 / Dokument Nr.: 1008373
Dateiname: KeMotion_Vision_UserManual_en.pdf
Seitenzahl : 33

© KEBA 2010

Änderungen im Sinne der technischen Weiterentwicklung vorbehalten. Angaben erfolgen ohne Gewähr.

Wir wahren unsere Rechte.

A: KEBA AG, Gewerbestraße 1, A-4041 Linz, Telefon: +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, Telefon: +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, Telefon: +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, Telefon +86 10 6409-6592, Fax +86 10 6409-6312, E-Mail: china@keba.com

www.keba.com

Änderungsverlauf

Version	Datum	Änderung in Kapitel	Beschreibung	geändert von
2.30	01-2010		created new	mhl
2.40	05-2011		add generic TCP camera	pen

Inhaltsverzeichnis

1	Einleitung	7
1.1	Zweck des Dokuments	7
1.2	Voraussetzungen	7
1.3	Bestimmungsgemäßer Gebrauch	7
1.4	Hinweise zu diesem Dokument	8
1.4.1	Inhalt des Dokuments	8
1.4.2	In diesem Dokument nicht enthalten	8
1.4.3	Weiterführende Dokumentationen	8
2	Sicherheitshinweise	10
2.1	Darstellung	10
3	Hardware Aufbau	11
3.1	Verbindungsdiagramm	11
3.2	Kompatibilität der Komponenten	12
4	Kamerasystem	13
4.1	Arbeitsablauf	13
4.2	Eigenschaften	13
4.2.1	Koordinatensystem	13
4.2.2	Aufnahmefrequenz (Framerate)	14
4.2.3	Übertragungsgeschwindigkeit	15
5	Software Konzept	16
5.1	Datenfluss Diagramm	16
5.2	Kommunikation mit dem Kamera-System	17
5.3	Bearbeitung der Objekte des Kamera-Systems	17
6	Konfiguration einer Kamera	19
6.1	Cognex Insight	20
6.2	Generic TCP	22
7	Kameratreiber	24
7.1	Datentypen für Kamerainteraktion (KVisionUtils.lib)	24
7.2	Kameratreiber (KVision.lib)	24
7.2.1	Cognex	25
7.2.2	Generic TCP	26
8	Diagnose	29
8.1	String format	29
8.2	Kameratreiber der Steuerung	29
	Index	33

1 Einleitung

1.1 Zweck des Dokuments

Dieses Dokument beschreibt, wie ein Kamera-System gemeinsam mit der KeMotion Tracking Technologie verwendet wird.

1.2 Voraussetzungen

Dieses Dokument richtet sich an folgende Personen mit entsprechenden Voraussetzungen:

Zielgruppe	Voraussetzung an Wissen und Können
Projektierer	Technische Grundausbildung (Fachhochschule, Ingenieur-Ausbildung oder entsprechende Berufserfahrung). Kenntnisse über: <ul style="list-style-type: none"> • die Arbeitsweise einer SPS, • Sicherheitsvorschriften, • Steuerungsprogrammierung (IEC61131)
Programmierer	Technische Grundausbildung (Fachhochschule, Ingenieur-Ausbildung oder entsprechende Berufserfahrung). Kenntnisse über: <ul style="list-style-type: none"> • die Arbeitsweise einer SPS, • Sicherheitsvorschriften, • Steuerungsprogrammierung (IEC61131) • Grundlagen der Robotik
Servicetechniker	Technische Grundausbildung (Fachhochschule, Ingenieur-Ausbildung oder entsprechende Berufserfahrung). Kenntnisse über: <ul style="list-style-type: none"> • die Arbeitsweise einer SPS, • Sicherheitsvorschriften, • Steuerungsprogrammierung (IEC61131)

1.3 Bestimmungsgemäßer Gebrauch

Die beschriebene Software wurde für die Steuerung von Industrierobotern und elektrischen Servoantrieben entwickelt. Sie darf nur in den beschriebenen Einsatzfällen und nur in Verbindung mit empfohlenen bzw. zugelassenen Ziel-systemen verwendet werden.



WARNUNG!

- 1) Das Kemro K2-System ist nicht für sicherheitsrelevante Steuerungsaufgaben (z.B.: Stillsetzen im Notfall oder Überwachung reduzierter Geschwindigkeiten) vorgesehen.
- 2) Das Kemro K2-System entspricht lt. EN ISO 13849-1 nur Sicherheits-Kategorie B PL a und ist somit für die Realisierung von Sicherheitsfunktionen für den Personenschutz nicht ausreichend.
- 3) Für sicherheitsrelevante Steuerungsaufgaben oder Personensicherheit müssen zusätzliche externe Schutzmaßnahmen realisiert werden, die auch im Fehlerfall einen sicheren Betriebszustand des Gesamtsystems gewährleisten.

1.4 Hinweise zu diesem Dokument

Dieses Handbuch ist Teil des Produktes. Es ist über seine gesamte Lebensdauer aufzubewahren und gegebenenfalls an nachfolgende Besitzer oder Benutzer des Produkts weiterzugeben

1.4.1 Inhalt des Dokuments

- Bearbeitung der Daten des Kamera-Systems in der Steuerung mit integrierter KeMotion Tracking Technologie
- Verwendung des Kameratreibers in der Steuerung
- Überblick über den Software Datenfluss
- Beschreibung der Aktivitäten des Kamera-Systems und der Hardware-Verbindung.

1.4.2 In diesem Dokument nicht enthalten

- Steuerungsprogrammierung
- Grundlagen der Robotik
- Grundlagen der Tracking Technologie

1.4.3 Weiterführende Dokumentationen

Folgende Dokumente sind zu beachten:

Nr	Bezeichnung	Zielgruppe
	PLC Robotic Programmierhandbuch	<ul style="list-style-type: none"> • Projektierer • Programmierer • Servicetechniker
	Grundlagen der Robotik	<ul style="list-style-type: none"> • Roboter Bediener • Projektierer • Programmierer • Servicetechniker

	KAIRO Sprachreferenz	<ul style="list-style-type: none">• Roboter Bediener• Projektierer• Programmierer• Servicetechniker
	Tracking Benutzerhandbuch	<ul style="list-style-type: none">• Projektierer• Programmierer• Servicetechniker

Die Dokumente können bei Fa. KEBA bezogen werden.

2 Sicherheitshinweise

2.1 Darstellung

Im Handbuch finden Sie an verschiedenen Stellen Hinweise und Warnungen vor möglichen Gefahren. Die verwendeten Symbole haben folgende Bedeutung:



GEFAHR!

- bedeutet, dass Tod oder schwere Körperverletzung eintreten werden, wenn die entsprechenden Vorsichtsmaßnahmen nicht getroffen werden.
-



WARNUNG!

- bedeutet, dass Tod oder schwere Körperverletzung eintreten kann, wenn die entsprechenden Vorsichtsmaßnahmen nicht getroffen werden.
-



VORSICHT!

- bedeutet, dass ein Sachschaden oder leichte Körperverletzung eintreten kann, wenn die entsprechenden Vorsichtsmaßnahmen nicht getroffen werden.
-

ACHTUNG

- bedeutet, dass ein Sachschaden eintreten kann, wenn die entsprechenden Vorsichtsmaßnahmen nicht getroffen werden.
-



- Mit dieser Warnung wird auf die möglichen Folgen beim Berühren von elektrostatisch empfindlichen Bauteilen hingewiesen.
-

Information

Anwendungstipps und nützliche Informationen werden mit "Information" gekennzeichnet. Sie enthalten keine Informationen, die vor einer gefährlichen oder schädlichen Funktion warnen.

3 Hardware Aufbau

Dieses Kapitel beschreibt die Einbindung eines Kamerasystems anhand einer typischen Tracking-Anwendung. Die Kamera erfasst Objekte auf einem Förderband und stellt diese Informationen einer Robotersteuerung zur Verfügung.

3.1 Verbindungsdiagramm

Folgende Abbildung zeigt die benötigten Hardwarekomponenten und deren Zusammenspiel an:

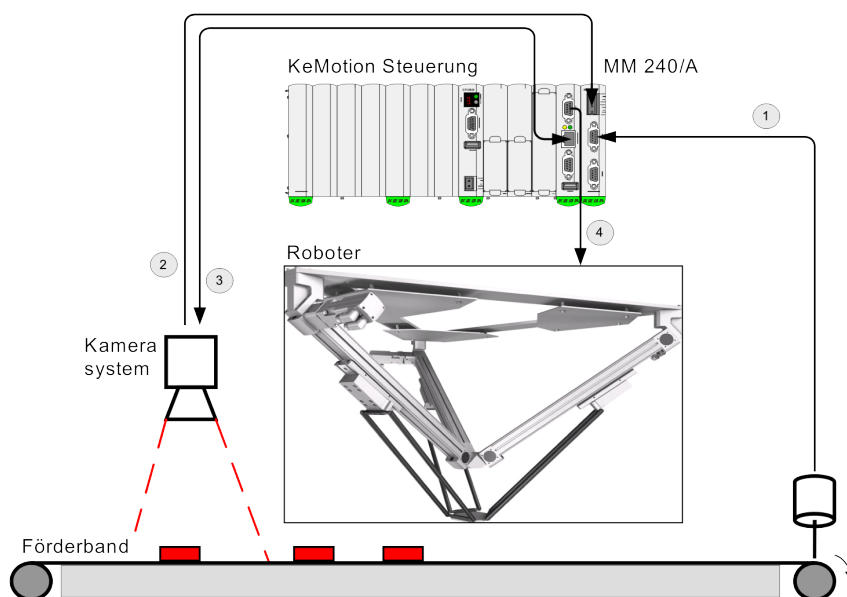


Abb.3-1: Verbindungsdiagramm eines Kameratracking-Systems

Beschreibung der Verbindungen

- 1) Der Inkrementalgeber ist mit dem Inkrementalgeber-Eingang der MM 240/A verbunden um die Position des Förderbandes an die Steuerung zu übermitteln.
- 2) Der Latch-Eingang der MM 240/A ist mit dem digitalen Ausgang der Kamera verbunden. Jede Bildaufnahme der Kamera löst somit im selben Moment einen Interrupt auf der Steuerung aus. Nun kann die Steuerung die Positionsinformation der Kamera mit dem Wert des Inkrementalgebers verknüpfen, um die Positionen des Objekts vorauszusagen. Ohne diese Verknüpfung wüsste die Steuerung nicht, wann und wo das Objekt für den Roboter zu greifen ist.
- 3) Verbindung zwischen Kamera-System und Steuerung via Feldbus
- 4) Die Steuerung benutzt die vom Kamera-System und dem Inkrementalgeber gewonnenen Informationen und bewegt den Roboter dementsprechend via Feldbus.

3.2 Kompatibilität der Komponenten

Roboter

Die KeMotion Tracking-Technologie ist Roboter unabhängig. Jeder unterstützter Kinematic-Typ kann verwendet werden.

Kamera-System

Bei einer KeMotion Steuerung können Kamera-Systeme mit folgenden Bussystemen verwendet werden:

- Ethernet
- PROFIBUS
- Serielle Kommunikation RS232, RS422 oder RS485

Die benötigte Kamerasystem-Geschwindigkeit hängt von der jeweiligen Anwendung ab.

Geber

Digitale Inkrementalgeber mit folgenden Ausgangssignalpegeln können verwendet werden:

- +/- 5 V differenziell (RS422)
- + 24 V einpolig geerdet (HTL)

Analoge Ausgangssignale von 1V Sinus (peak-to-peak) werden nicht unterstützt. Für nähere Informationen zu unterstützten Inkrementalgebern und deren Verdrahtung verwenden Sie bitte das Handbuch des Encoder-Moduls MM240/A.

Information

Die benötigte Auflösung des Inkrementalgebers ist abhängig vom Einbau in das Förderband. Es wird empfohlen die Auflösung des Inkrementalgebers so zu wählen, dass die Auflösung am Förderbandes mindestens 500 Schritte pro Millimeter beträgt, auch wenn der Roboter diese Genauigkeit von 1/500 mm nicht unterstützt. Diese genauen Werte werden für interne Rechenoperationen benötigt, um genaue Ergebnisse zu erhalten.

4 Kamerasystem

Dieses Kapitel beschreibt die Verwendung und Abarbeitung des Kamerasystems.

4.1 Arbeitsablauf

Das Kamerasystem muss folgende Aufgaben erfüllen, um korrekt mit der KeMotion Tracking Technologie zusammenzuarbeiten:

- Erkennen und Übertragen aller Objekte im Sichtfeld (FOV: Field of view). Das Kamerasystem soll die Position des Teils senden, wo der Roboter zugreifen muss (grip point). Dieser Zugriffspunkt ist typischer Weise der Lastschwerpunkt.
- Die Position des Teils (x, y, z) muss in mm und nicht in Pixels angegeben werden. Die Ausrichtung (a, b, c) wird in Grad [°] angegeben. Typisch werden durch das Kamera-System die Variablen "x", "y" und "a" geschrieben, alle anderen sind 0.
- Das Kamera-System sendet bei der Bildaufnahme an den digitalen Ausgang einen Impuls. Das Signal wird mit dem Latch-Eingang des Gebermoduls MM 240/A verbunden. Üblicher Weise wird der digitale Ausgang für den externen Biltz verwendet. Diese Information wird verwendet, um die richtige Encoder-Position zum Zeitpunkt des Auslösens zu finden.
- Abhängig von der Applikation kann das Kamera-System die Werte der Attribute (DWORD) für jedes Teil senden. Dieser Wert ist ein benutzerdefiniertes Bit-Feld, wo alle für den Prozess notwendigen Informationen vom Kamera-System gespeichert werden. Beispiele: Farbe des Teils, Größe des Teils, Typ des Teils, ...
- Die Attribute können für jedes Teil definiert werden und sind vom Steuerungsprogramm sowie vom KAIRO-Programm aus zugänglich. Abhängig vom Attribut kann die Steuerung Teile unterschiedlich behandeln und der Roboter kann entscheiden, was mit diesem Teil zu tun ist.
Wenn ein Teil mehrmals als Bild erfasst wurde (z.B. weil das Förderband sehr langsam ist) dann müssen die Attribute für dieses Teil immer die selben sein.

4.2 Eigenschaften

4.2.1 Koordinatensystem

Um die optimale Funktion des Kamerasystems zu gewährleisten, muss die positive x-Koordinate genau mit der Bewegungsrichtung des Förderbandes übereinstimmen.

Information

Die meisten Kamera-Systeme enthalten eine Selbstkalibrierung mit einem Kalibrierungsmusterblatt. Das Kalibrierungsmusterblatt kann auf das Förderband gelegt werden und muss exakt am Förderband ausgerichtet sein.

Es kann hilfreich sein, den Ursprung des Koordinatensystems an einem fixen Punkt für weitere Messungen des Roboter Tracking Referenz-Systems zu markieren.

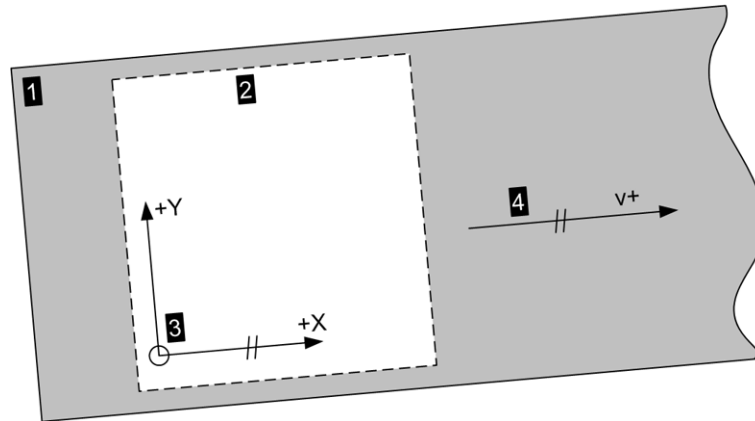


Abb.4-1: Beispiel für eine Kalibrierung des Kamerasystems

1 ... Förderband	2 ... Sichtfeld der Kamera (FOV)
3 ... Koordinatensystem des Kamera-Systems	4 ... Bewegungsrichtung des Förderbandes (positive Geschwindigkeit)

4.2.2 Aufnahmefrequenz (Framerate)

Die Aufnahmefrequenz des Kamerasystems ist abhängig von der Sichtfeldgröße und der Förderbandgeschwindigkeit. Für die meisten Anwendungen ist es nicht nötig, die maximal mögliche Aufnahmegeschwindigkeit des Kamerasystems zu verwenden. Es ist nicht wichtig, die Bildaufnahmegeschwindigkeit so zu takten, daß jedes Teil am Förderband nur einmal aufgenommen wird. Eine passende Update-Rate des Kamera-Systems ist erreicht, wenn jeder Teil am Förderband 2 oder 3 x aufgenommen wird. Somit wird mehr Sicherheit erreicht, auch jedes Teil aufzunehmen. Die Update-Rate wird wie folgt berechnet:

$$\text{Aufnahmefrequenz[Bilder/sek]} = \frac{\text{Förderbandgeschwindigkeit[mm/sek]}}{\text{Kamerasichtfeld.xLaenge[mm]}} * 2,5$$

Abb.4-2: Formel für die optimale Aufnahmefrequenz des Kamerasystems

Beispiel

Die maximale Förderbandgeschwindigkeit ist 1m/s (= 1000 mm/sek) und die Größe des Kamerasichtfeld 0,5 m in Förderband-Laufrichtung.

$$(1000 / 500) * 2,5 = 5$$

Das Kamerasystem soll mit einer Frequenz von 5 Bildern pro Sekunde arbeiten.

4.2.3 Übertragungsgeschwindigkeit

Um sicherzustellen, dass die erhaltenen Daten des Kamerasystems mit den Geber Latch-Daten übereinstimmen ist es wichtig, im Kamerasystem eine gewisse Zeitspanne einzuhalten. Das nächste Bild kann erst dann aufgenommen werden, wenn das letzte Bild komplett zur Steuerung übertragen wurde.

Es ist nicht erlaubt, während des Sendens von Daten zur Steuerung ein Bild aufzunehmen, weil dies eine Erhöhung der Latch-Daten des Geber-Moduls auslöst. Die erhaltenen Daten verweisen immer auf die letzten aufgetretenen Latch-Daten. Die Steuerung würde eine falsche Geber-Position für die erhaltenen Daten verwenden, wenn in dieser Zeit ein neues Bild aufgenommen wird.

In Applikationen, die die Bildaufnahme selbst von der Steuerung aus starten, muss dieses Verhalten zeitlich berücksichtigt werden.

5 Software Konzept

Im folgenden Kapitel wird der softwaremäßige Bearbeitung der Objekte und die Kommunikation mit der im System integrierten Kamera beschrieben.

5.1 Datenfluss Diagramm

Die Bearbeitung der eingehenden Daten (Objekt-Informationen: Position und Attribute) der Kamera muss von der Steuerung erledigt werden.

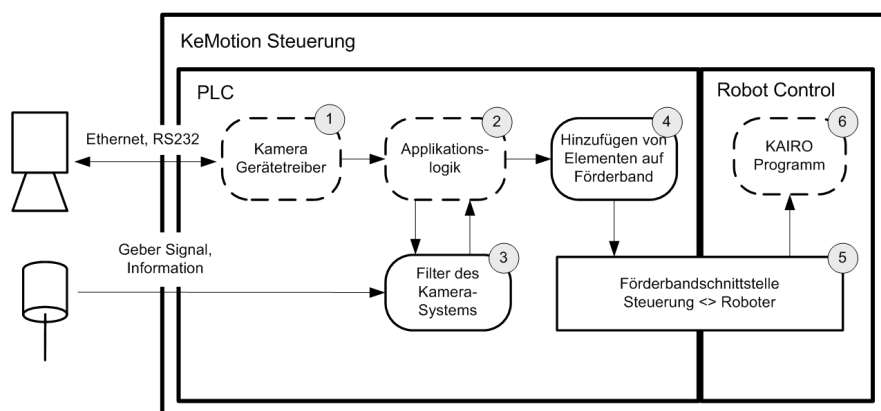


Abb.5-1: Software-Konzept

Beschreibung des Software-Konzeptes:

- Der Kameratreiber für das Kamera-System ist ein Funktionsblock, um mit dem ausgewählten Kamera-System-Typ zu interagieren. Dieser Funktionsblock baut eine Verbindung zur Kamera unter Verwendung von Ethernet oder RS232 auf. Der Treiber analysiert den erhaltenen Datenstrom der Kamera und stellt die Ergebnisse in Listeform zur Verfügung. Die Objekte der Kamera enthalten Informationen über die Position, Attribute und eine ID (Identifikationsnummer).
- Die Applikationslogik verarbeitet die erhaltenen Daten. Es ist möglich, die Objekte abhängig von ihren Attributen zu filtern oder unterschiedlichen Robotern zuzuteilen.
- Der Funktionsblock zur Filterung doppelt erhaltener Objekte wird durch die Bibliothek RCTracking.lib zur Verfügung gestellt. Der Name davon ist "RCTR_FilterObjects". (Siehe auch Tracking Benutzerhandbuch.) Dieser Funktionsblock erkennt Objekte der Kamera, die öfter erkannt wurden und filtert diese aus. Zusätzlich wird die letzte Geberposition der MM 240/A den Objektinformationen hinzugefügt. Dies ist wichtig, um die korrekte Position des Objektes am Förderband zu erhalten. Als Ergebnis liefert dieser Funktionsblock die erkannten Objekte des Förderbandes zurück.
- Der Funktionsblock `RCTR_AddObject` fügt die Objekte in die Objekt-Liste des spezifizierten Förderbandes. Die Funktion befindet sich in der Bibliothek `RcTracking.lib`. (Siehe auch Tracking Benutzerhandbuch)

Es ist möglich mit dieser Funktion mehr als ein Objekt einzufügen, darum muss die Funktion bei jedem erkannten Objekt aufgerufen werden, das in die Objekt-Liste eingefügt werden soll.

- Die Förderband-Schnittstelle enthält eine Objekt-Liste, auf die ein Roboter im KAIRO-Programm zugreifen kann. Die Förderband-Schnittstelle enthält Informationen über die aktuellen Geberwerte und deren Auflösung.
- Im KAIRO-Programm kann der Roboter die Objekte aus der Objekt-Liste aufrufen. Das Programm erhält ein Objekt welches ein Objekt des Förderbands darstellt. Der Roboter kann zu diesem Objekt fahren und es verarbeiten. Nach der Verarbeitung wird das Objekt als "fertig" markiert. Das Objekt wird aus der Objekt-Liste entfernt.

Information

Um ein Objekt in der Steuerung wegzufiltern, ist es nicht notwendig, das Objekt zu löschen. Fügen Sie das Objekt nicht in die Objektliste ein.

5.2 Kommunikation mit dem Kamera-System

Der Funktionen des Kamera-Systems hängt vom verwendeten System ab. Die folgenden Funktionen müssen im Kamera-Treiber ausgeführt werden:

- Verbindung zum Kamera-System
- Auswahl des Programmes des Kamera-Systems
- Periodisches Auslösen der Bildaufnahme
- Erhalten der Daten von der Kamera und Befüllen der Objekt-Listen mit Daten
- Zustandsüberwachung der Kamera (Fehler, ...)
- Optional: Einstellen von Kundenspezifischen Parameter für die Bildanalyse (z.B. Toleranzwerte, ...)

5.3 Bearbeitung der Objekte des Kamera-Systems

Das Kamera-System sendet normalerweise alle erkannten Objekte im Sichtfeld. Es ist möglich, daß ein Gegenstand am Förderband mehrmals erkannt wird.

Der Funktionsblock RCTR_FilterObject kann zur automatischen Filterung dieser Objekte verwendet werden. Dieser Funktionsblock fügt ebenfalls die notwendigen erhaltenen Geber-Informationen zu den Objekten hinzu. (Siehe auch Tracking Benutzerhandbuch.) Ausgehende Objekte sind alle neuen Objekte am Förderband. Die Objekte sind nun deterministisch.

Die Applikationslogik kann nun die Objekte verarbeiten. Im Fall eines Multikinematik-Systems ist es möglich, daß ein Objekt auf mehrere Roboter aufgeteilt ist, so daß jeder Roboter die selbe Anzahl von Objekte abarbeiten muss.

Die Applikationslogik kann Objekte abhängig ihrer Attributwerte verwerfen. Z.B. wenn das Kamera-System Objekte über das Attribut mit "gut" oder "schlecht" klassifiziert. Zusätzlich ist es möglich, für ein Objekt eigene Attribute zu erzeugen oder die Attribute zu verändern.

Zu Diagnosezwecke kann ein Objekt mit einer Identifikationsnummer versehen werden. Diese Nummer wird bei Warnungen oder Fehlermeldungen ausgegeben und kann auch am Förderband mitaufgezeichnet werden.

6 Konfiguration einer Kamera

Dieses Kapitel beschreibt die notwendigen Schritte für eine Konfiguration eines Kamera-Systems. Folgende Kamera-Typen werden unterstützt:

- Cognex Insight
- Generic TCP

Steuerung konfigurieren

Legen Sie in der Steuerungskonfiguration ein neues Element "Vision" an.

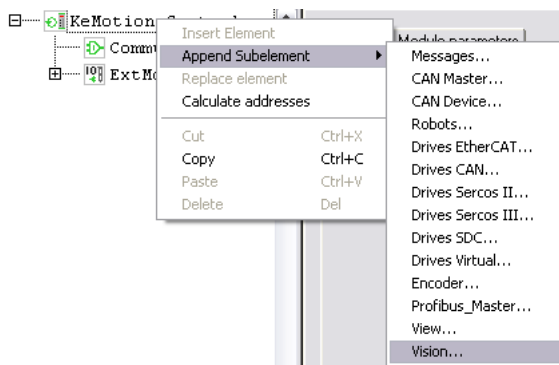


Abb.6-1: Anlegen eines "Vision" Elements

Legen Sie für jede Kamera ein neues Element an.

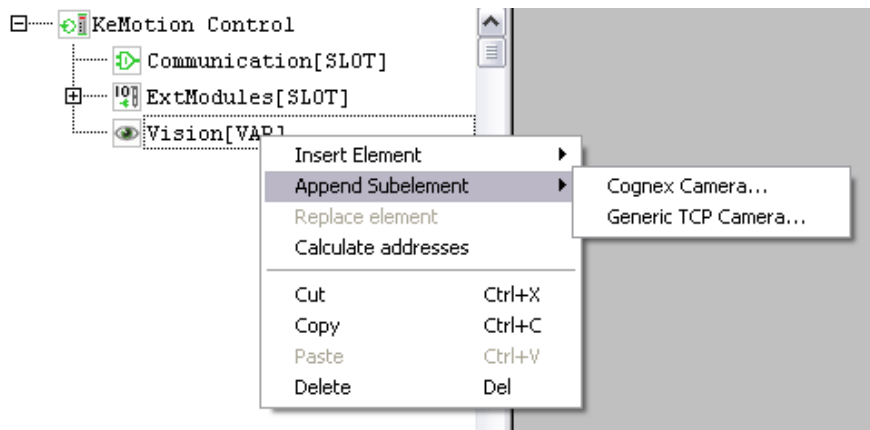


Abb.6-2: Anlegen einer Kamera

IO konfigurieren

Konfigurieren Sie einen Digitalen High-Speed-Ausgang der Kamera (getrennter Ausgang an der Kamera, nicht der Ausgang an dem das Ethernet-IO-Modul angeschlossen ist). Verwenden Sie den Typ "Acquisition Start" mit einer Puls-Länge von 5-10 ms. Verbinden Sie den Ausgang mit dem Latch-Eingang eines Geber-Moduls (z.B. MM 240/A). Dieses Signal ist notwendig um die aktuelle Geber Position bei der Bilderkennung auszuwerten.

IEC Bibliotheken

Um Kameras in ein IEC-Projekt einzubinden werden die Bibliotheken KVisi-on.lib und KVisionUtils.lib benötigt.

6.1 Cognex Insight

Um eine Cognex Insight Kamera im System zu verwenden, sind folgende Schritte notwendig:

- Kemro Automation Snippet installieren
- Steuerung konfigurieren
- Kamera konfigurieren
- Kemro Automation Snippet zum Job hinzufügen

Kemro Automation Snippet installieren

Kopieren Sie die Kemro_Automation.cxd auf der mitgelieferten DVD (Verzeichnis: target/vision) in das Snippet-Verzeichnis der Cognex-Installation (z.B. C:\Program Files\Cognex\In-Sight\In-Sight Explorer 4.4.0\Snippets).

Steuerung konfigurieren

Konfigurieren Sie die Steuerung wie bereits beschrieben, wählen Sie "Cognex Camera" für den Kamera-Typ.

Stellen Sie für die Kamera die notwendigen Parameter ein.

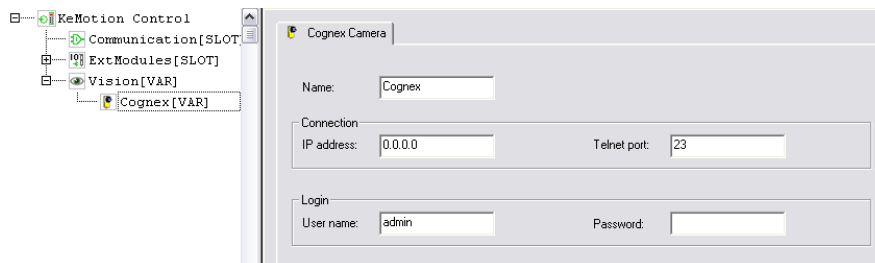


Abb.6-3: Parametereinstellung

Eintrag	Beschreibung
Name	Eindeutiger Name der Kamera. Mit diesem Namen wird eine Zugriffsvariable im IEC-Projekt angelegt.
IP adress	IP-Adresse der Kamera
Telnet port	Telnet Port der Kamera
User name	Benutzer für das Anmelden an der Kamera. Der Benutzer muss Vollzugriff haben.
Password	Passwort des Benutzers

Kamera konfigurieren

Wählen Sie im Insight Explorer **Sensor ▶ Startup....** Im Startup-Dialog die Online-Checkbox und einen Job auswählen. Wählen Sie keinesfalls <new> aus - in diesem Fall erkennt die Steuerung einen ungesicherten Job nach dem Neustart der Kamera.

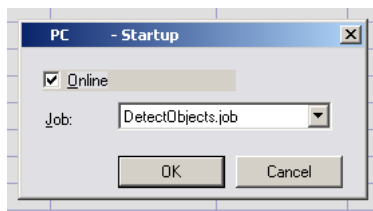


Abb.6-4: Cognex Insight Start-up Dialog.

Setzen Sie im "Property sheet" der "AcquireImage cell" (Doppelklick auf Zelle A0) den Trigger auf "External" und den Buffer-Mode auf "Single".

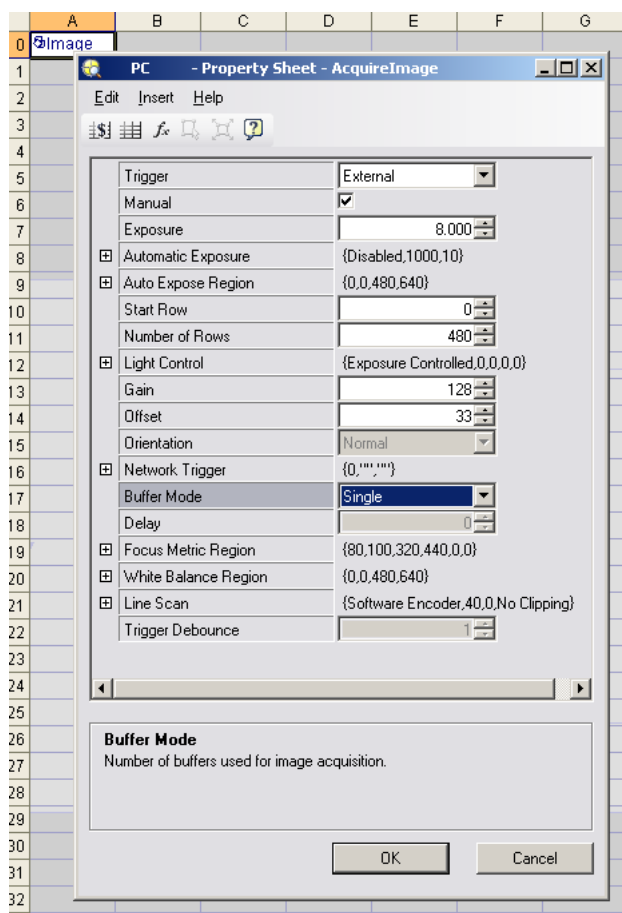


Abb.6-5: AcquireImage Konfiguration

Wählen Sie im Cognex Explorer **Sensor ▶ User List...** aus, um den Dialog zu öffnen. Stellen Sie sicher, dass der eingestellte Benutzer Vollzugriff auf die Kamera hat.

Kemro Automation Snippet zum Job hinzufügen

Für die Kommunikation mit der Steuerung müssen Sie das Kemro Automation Snippet zum Job hinzufügen.

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

Abb.6-6: Kemro Automation Snippet

Das Snippet enthält 10 Zeilen (eine Zeile pro Objekt), die für das Senden von Daten an die Steuerung verwendet werden. Werden weniger Zeilen benötigt, können die überflüssigen Zeilen am Ende des Snippets gelöscht werden. Reichen die Zeilen nicht aus, können Zeilen durch Kopieren und Einfügen unterhalb des Snippets angelegt werden.

Verbinden Sie die Zellen der fünf Spalten (X, Y, A, Attribute und ID) auf der linken Seite des Snippets mit den Zellen der entsprechenden Objekt-Daten.

Der Daten-Port ist der Port der Kamera, zu dem sich die Steuerung verbindet um Objekt-Daten zu erhalten. Sie können den Daten-Port in der ersten Zeile des Snippets umstellen.

Wenn eine der ersten fünf Spalten einen Error enthält (#ERR), dann werden die Daten dieser Zeile nicht zur Steuerung übertragen.

6.2 Generic TCP

Um eine gerische TCP Kamera im System zu verwenden, sind folgende Schritte notwendig:

- Steuerung konfigurieren
- Kamera konfigurieren

Steuerung konfigurieren

Konfigurieren Sie die Steuerung wie bereits beschrieben, wählen Sie "Generic TCP Camera" für den Kamera-Typ.

Stellen Sie für die Kamera die notwendigen Parameter ein.

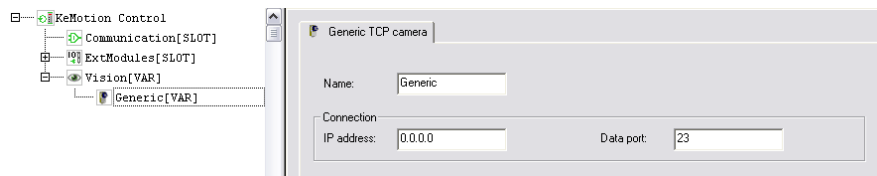


Abb.6-7: Parametereinstellung

Eintrag	Beschreibung
Name	Eindeutiger Name der Kamera. Mit diesem Namen wird eine Zugriffsvariable im IEC-Projekt angelegt.
IP adress	IP-Adresse der Kamera
Data port	Der Daten-Port, der auf der Kamera konfiguriert ist.

7 Kameratreiber

Dieses Kapitel beschreibt die Kameratreiber, die in den KeMotion Targets enthalten sind.

7.1 Datentypen für Kamerainteraktion (KVisionUtils.lib)

KVisionUtils.lib enthält die wichtigsten Datentypen, die für eine Interaktion mit einer Kamera notwendig ist.

Folgende Typen sind verfügbar:

Name	Beschreibung
TKVIS_CamName	Kameraname
TKVIS_CartFrame	Kartesischer Frame mit 6 Koordinaten für Position und Orientierung des Objekts
TKVIS_Object	Objekt mit kartesischem Frame, ID und Attribut
TKVIS_ObjectList	Liste (fixe Länge) von TKVIS_Objects und numerischer Wert (Integer), wieviele Objekte in der Liste enthalten sind
TKVIS_LineList	Liste von Strings (fixe Länge) und numerischer Wert (Integer), wieviele Strings in der Liste enthalten sind
TKVIS_DiagnosisData	Diagnosedaten
TKVIS_Diagnosis	Diagnoseeinstellungen

7.2 Kameratreiber (KVision.lib)

Die Bibliothek KVision.lib bietet Funktionen an, um mit einer Kamera zu interagieren (z.B: Verbindung zur Kamera herstellen oder Bildaufnahme auslösen).

Eine Kamera arbeitet ihr Programm ab und liefert die Ergebnisse als Strings über den Daten-Port zur Steuerung zurück. Bild-ergebnisse müssen mit dem String 'Image' beginnen und mit dem String 'Done' enden. Andere Ergebnisse müssen mit dem String 'Result' beginnen und mit dem String 'Done' enden. Der Kameratreiber auf der Steuerung wertet die Strings aus. Ist der String innerhalb eines Bild-ergebnisses in einem gültigen Objektformat, wird daraus ein Objekt vom Typ TKVIS_Object generiert und in die Objekt-Liste (TKVIS_ObjectList) eingetragen. Ansonsten wird der String unverändert in die String-Liste (TKVIS_LineList) eingetragen. Die Applikation entscheidet in diesem Fall, wie eine Weiterverarbeitung erfolgt. Die Objekte aus der Objekt-Liste können entweder von der RCTracking.lib oder ebenfalls direkt von der Applikation verwendet werden.

Die beiden Listen sind genau einen Update-Zyklus des IEC-Programms gültig. Im nächsten Zyklus werden die Listen wieder neu befüllt. Die Ergebniserfassung kann mehrere Zyklen dauern. Nach Abschluss der Ergebniserfassung wird der passende Rückgabewert des Kameratreibers ('Done' oder 'ImageDone') einen Zyklus lang auf `TRUE` gesetzt. Danach beginnt die nächste Ergebniserfassung und der Rückgabewert wechselt auf `FALSE`.

KVision.lib stellt für jeden Kamertyp eigene Datentypen und einen eigenen Funktionsblock (Kameratreiber) zur Verfügung.

Folgende Kamertypen werden unterstützt:

- Cognex
- Generic TCP

7.2.1 Cognex

Folgende Tabelle beschreibt die Datentypen und Funktionsblöcke:

Name	Typ	Beschreibung
CAM_COGNEX_REF	STRUCT	Datenstruktur der Cognex Kamera
KVIS_CgxClient	Funktionsblock	Treiber für Cognex Insight Kameras
TKVIS_CgxJobName	STRING	Name eines Cognex-Jobs

CAM_COGNEX_REF

Für jede konfigurierte Cognex-Kamera wird automatisch eine globale Variable des Typs CAM_COGNEX_REF erzeugt. Der Name ist identisch mit dem Namen der Kamera in der Steuerungskonfiguration. Diese Variable wird zum Aufruf des Funktionsblocks KVIS_CgxClient benötigt. Zusätzlich enthält sie eine Member *Diagnosis* vom Typ TKVIS_Diagnosis, in der zusätzliche Diagnoseinformationen für den Programmierer enthalten sind.

KVIS_CgxClient

Dieser Funktionsblock enthält den Kameratreiber. Cognex Kameras übertragen über den Daten-Port ausschließlich Bilder, daher heißt der entsprechende Rückgabewert 'ImageDone'.

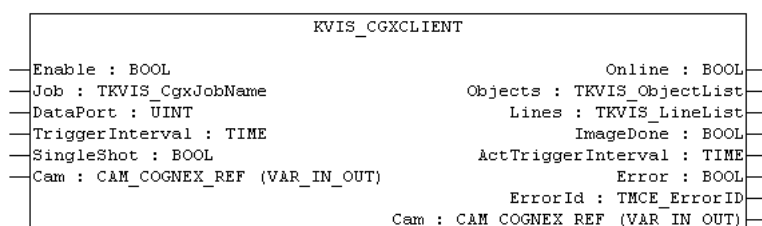


Abb.7-1: Funktionsblock

Beschreibung der Datentypen:

VAR_INPUT

Name	Typ	Beschreibung
Enable	BOOL	Bei steigender Flanke im Eingangssignal verbindet sich die Steuerung mit der Kamera und die Bildaufnahme startet.
Job	TKVIS_CgxJob-Name	Name des Jobs. Muss vor dem Anstecken der Kamera festgelegt werden.
DataPort	UINT	Angabe des Datenports (wie im Snippet eingestellt)
TriggerInterval	TIME	Legt die minimale Zeitspanne zwischen zwei Bildaufnahmen fest. Dauert die Bildverarbeitung länger, hat das Intervall keine Bedeutung. Durch Setzen von T#0ms wird das automatische Intervall deaktiviert.
SingleShot	BOOL	Einzelbildaufnahme bei steigender Flanke im Eingangssignal

VAR_IN_OUT

Name	Typ	Beschreibung
Cam	CAM_COGNEX_REF	Referenz der Kamera

VAR_OUTPUT

Name	Typ	Beschreibung
Online	BOOL	TRUE, wenn die Kamera verbunden und bereit zur Bilderkennung ist.
Objects	TKVIS_Object-List	Liste mit allen gültigen Objekten, die im letzten Aktualisierungsvorgang empfangen wurden. Diese Liste kann direkt mit dem dem Objekt-Eingang des Funktionsblocks RCTR_FilterObjects verbunden werden. Diese Liste wird bei jedem Empfangszyklus neu befüllt.
Lines	TKVIS_LineList	Liste mit allen Strings die während eines Aktualisierungsvorgangs empfangen, aber nicht in Objekte umgewandelt wurden.
ImageDone	BOOL	Nach Abschluss der Bilderkennung wird der Rückgabewert auf TRUE gesetzt. Im selben Zyklus können die letzten Ergebnisse in den Listen stehen.
ActTriggerInterval	TIME	Dauer des aktuell ausgeführten Auslöse-Intervalls
Error	BOOL	Indikator für den Fehlerfall
ErrorID	TMCE_ErrorID	Enthält die genaue Fehlerbeschreibung

7.2.2 Generic TCP

Folgende Tabelle beschreibt die Datentypen und Funktionsblöcke:

Name	Typ	Beschreibung
CAM_GENTCP_REF	STRUCT	Datenstruktur einer generischen TCP Kamera
KVIS_TCPClient	Funktionsblock	Treiber für eine generische TCP Kamera
TKVIS_TcpCommand	STRING	Kommando für die Kamera

CAM_GENTCP_REF

Für jede konfigurierte TCP-Kamera wird automatisch eine globale Variable des Typs CAM_GENTCP_REF erzeugt. Der Name ist identisch mit dem Namen der Kamera in der Steuerungskonfiguration. Diese Variable wird zum Aufruf des Funktionsblocks KVIS_TCPClient benötigt. Zusätzlich enthält sie eine Member-Variable *Diagnosis* vom Typ TKVIS_Diagnosis, in der zusätzliche Diagnoseinformationen für den Programmierer enthalten sind.

KVIS_TCPClient

Dieser Funktionsblock enthält den Kameratreiber.

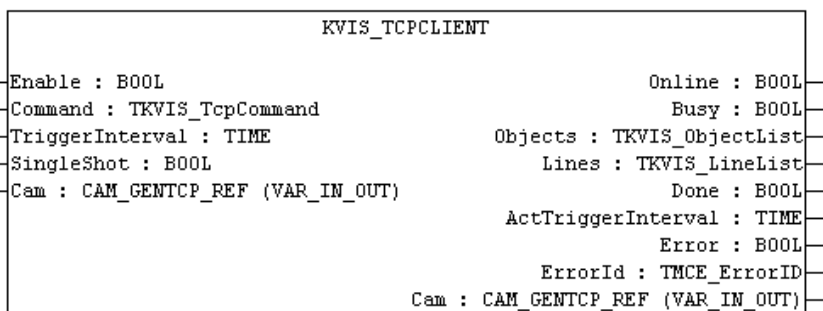


Abb.7-2: Funktionsblock

Beschreibung der Datentypen:

VAR_INPUT

Name	Typ	Beschreibung
Enable	BOOL	Zur Kamera verbinden.
Command	TKVIS_TcpCommand	Kommando das zur Kamera gesendet werden soll.
TriggerInterval	TIME	Legt die minimale Zeitspanne zwischen zwei Bildaufnahmen fest. Dauert die Bildverarbeitung länger, dann wird das nächste Kommando sofort nach der Beendigung gestartet. Durch Setzen von T#0ms wird das automatische Intervall deaktiviert.
SingleShot	BOOL	Absetzen eines einzelnen Kommandos bei steigender Flanke im Eingangssignal.

VAR_IN_OUT

Name	Typ	Beschreibung
Cam	CAM_COG-NEX_REF	Referenz der Kamera

VAR_OUTPUT

Name	Typ	Beschreibung
Online	BOOL	TRUE, wenn die Kamera verbunden ist.
Busy	BOOL	TRUE, solange der Funktionsblock gewartet werden muss.
Objects	TKVIS_Object-List	Liste mit allen gültigen Objekten, die im letzten Aktualisierungsvorgang empfangen wurden. Diese Liste kann direkt mit dem dem Objekt-Eingang des Funktionsblocks RCTR_FilterObjects verbunden werden. Diese Liste wird bei jedem Empfangszyklus neu befüllt.
Lines	TKVIS_LineList	Liste mit allen Strings die während eines Aktualisierungsvorgangs empfangen, aber nicht in Objekte umgewandelt wurden.
Done	BOOL	Nach Abschluss der Ergebniserkennung wird der Rückgabewert auf TRUE gesetzt. Im selben Zyklus können die letzten Ergebnisse in den Listen stehen.
ActTriggerInterval	TIME	Dauer des aktuell ausgeführten Auslöse-Intervalls
Error	BOOL	Indikator für den Fehlerfall
ErrorID	TMCE_ErrorID	Enthält die genaue Fehlerbeschreibung

8 Diagnose

In diesem Kapitel werden die Diagnosemöglichkeiten für ins System eingebundene Kameras beschrieben

8.1 String format

Die generierten Strings, die von der Kamera zur Steuerung gesendet werden, enthalten markierte Werte in eckigen Klammern, die durch Beistrich getrennt sind. Ein gültiger Objektstring ist z.B:

[X:<x>;Y:<y>;A:<a>;ATTR:<attr>;ID:<id>]. Fehlt ein Wert, wird dieser auf 0 gesetzt. Die Reihenfolge der Wertepaare ist nicht vorgeschrieben.

8.2 Kameratreiber der Steuerung

Die Variable, die von einer konfigurierten Kamera instanziiert ist hat eine Member-Variable `Diagnosis` vom Typ `TKVIS_Diagnosis`. Diese Variable enthält Informationen über den aktuellen Zustand des Kameratreibers. Wenn die Applikation der Steuerung debugged wird, können folgende Informationen ausgelesen werden:

```

0001  └─Cognex.Diagnosis
0002     └─.LogLines = TRUE
0003     └─.Manual = FALSE
0004     └─.Data
0005         └─.Online = TRUE
0006         └─.Error = FALSE
0007         └─.ErrorId = eMCE_InfoNoError
0008         └─.Job = 'FindTheBlobs'
0009         └─.TriggerInterval = T#304ms
0010         └─.LastNrOfDataLines = 6
0011         └─.AccNrOfImages = 36
0012         └─.AccNrOfDataLines = 210
0013     └─.Lines
0014         └─.Lines[0] = 'Done'
0015         └─.Lines[1] = 'Image'
0016         └─.Lines[2] = '[X:164.774933;Y:279.540466;A:90;ATTR:0;ID:0]'
0017         └─.Lines[3] = '[X:144.194275;Y:178.246017;A:90;ATTR:0;ID:0]'
0018         └─.Lines[4] = '[X:97.305275;Y:135.722427;A:90;ATTR:0;ID:0]'
0019         └─.Lines[5] = '[X:81.044884;Y:73.941277;A:90;ATTR:0;ID:0]'
0020         └─.Lines[6] = 'Done'
0021         └─.Lines[7] = 'Image'
0022         └─.Lines[8] = 'Done'
0023         └─.Lines[9] = 'Image'
0024         └─.Lines[10] = 'Done'
0025         └─.Lines[11] = 'Image'
0026         └─.Lines[12] = '[X:574.405518;Y:54.843769;A:90;ATTR:0;ID:0]'
0027         └─.Lines[13] = '[X:567.057617;Y:352.919128;A:90;ATTR:0;ID:0]'
0028         └─.Lines[14] = '[X:552.410461;Y:191.394989;A:90;ATTR:0;ID:0]'
0029         └─.Lines[15] = 'Done'
0030         └─.Lines[16] = 'Image'
0031         └─.Lines[17] = '[X:603.824402;Y:319.208618;A:90;ATTR:0;ID:0]'
0032         └─.Lines[18] = '[X:523.037598;Y:424.950897;A:90;ATTR:0;ID:0]'
0033         └─.Lines[19] = '[X:515.661804;Y:234.070679;A:90;ATTR:0;ID:0]'
0034         └─.Lines[20] = '[X:429.02652;Y:54.855881;A:90;ATTR:0;ID:0]'
0035         └─.Lines[21] = '[X:421.632202;Y:352.983948;A:90;ATTR:0;ID:0]'
0036         └─.Lines[22] = '[X:406.943329;Y:191.404129;A:90;ATTR:0;ID:0]'
0037         └─.Lines[23] = 'Done'
0038         └─.Lines[24] = '-----'
0039         └─.Lines[25] = 'Image'
0040         └─.Lines[26] = '[X:310.136078;Y:279.59436;A:90;ATTR:0;ID:0]'
0041         └─.Lines[27] = '[X:289.62796;Y:178.228851;A:90;ATTR:0;ID:0]'
0042         └─.Lines[28] = '[X:242.700897;Y:135.759445;A:90;ATTR:0;ID:0]'
0043         └─.Lines[29] = '[X:226.445374;Y:73.960068;A:90;ATTR:0;ID:0]'
0044         └─.Lines[30] = '[X:160.346664;Y:279.513062;A:90;ATTR:0;ID:0]'
0045         └─.Lines[31] = '[X:76.636307;Y:419.084137;A:90;ATTR:0;ID:0]'
0046         └─.FirstLine = 24
0047

```

Abb.8-1: Diagnosedaten

Manual, LogLines, Data.Lines

Data.Lines ist ein Ringbuffer von der Länge `gcKVIS_DiagnosisLineBufLen`, der alle Einträge enthält, die von der Kamera zur Steuerung übertragen werden. Im Diagnose-Buffer werden die Einträge auf die Länge von `gcKVIS_DiagnosisLineLen` gekürzt. Ein Linie mit Bindestrichen markiert das Ende des Ringbuffers. Wird `Manual` auf `FALSE` gesetzt, dann wird jede Information zum Bild im Ring-Buffer abgespeichert. Wird `Manual` auf `TRUE` gesetzt, dann kann über die Applikation über den Parameter `LogLines` (auf `TRUE` setzten) getriggert werden, welches Informationen zum Bild gespeichert/protokolliert wird. `Manual` wirkt sich nur auf das Speichern aus und hat sonst keinen Effekt auf den Kameratreiber.

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

Diese Variablen bilden den Input und den Output des Kamera-Treibers. `Data.TriggerInterval` ist das aktuelle Trigger-Intervall.

Data.LastNrOfDataLines

Die Anzahl der Einträge des letzten Bildes, das in den Ringbuffer gespeichert wurde, Einträge wie `Image` oder `Done` nicht mitgezählt.

Data.AccNrOfDataLines, Data.AccNrOfImages

Die aufgerechnete Anzahl aller Einträge und Bilder. Es werden auch Einträge von Bilder mitgezählt, die nicht im Ringbuffer gespeichert wurden.

Index

K

KVisionUtils.lib 24

