



KUKA System Technology  
KUKA.Ethernet KRL 3.2  
适用于 KUKA System Software 8.7



发布日期: 17.10.2019  
KST Ethernet KRL 3.2 V1  
KUKA Deutschland GmbH

©版权说明 2019

KUKA Deutschland GmbH  
Zugspitzstraße 140  
D-86165 Augsburg  
德国

此文献或节选只有在征得 KUKA Deutschland GmbH 明确同意的情况下才允许复制或对第三方开放。除了本文献中说明的功能外，控制系统还可能具有其他功能。但是在新供货或进行维修时，无权要求 KUKA Deutschland GmbH 提供这些功能。

我们已就印刷品的内容与描述的硬件和软件内容是否一致进行了校对。但是不排除有不一致的情况，我们对此不承担责任。但是我们定期校对印刷品的内容，并在之后的版本中作必要的更改。

我们保留在不影响功能的情况下进行技术更改的权利。

KIM-PS5-DOC

原版文件的译文

出版物: Pub KST Ethernet KRL 3.2 (PDF) zh  
PB12988

书页构造: KST Ethernet KRL 3.2 V1.1  
BS12038

版本: KST Ethernet KRL 3.2 V1

# 目录

<b>1</b>	<b>引言.....</b>	<b>7</b>
1.1	目标群体.....	7
1.2	工业机器人文献.....	7
1.3	提示的图示.....	7
1.4	所用概念.....	8
1.5	商标.....	9
1.6	许可证.....	9
<b>2</b>	<b>产品说明.....</b>	<b>11</b>
2.1	产品说明.....	11
2.2	连接配置.....	11
2.2.1	传输协议.....	11
2.2.2	失去连接时的特性.....	11
2.2.3	监控连接.....	11
2.3	数据交换.....	12
2.4	数据保存.....	12
2.5	客户端-服务器运行模式.....	14
2.6	记录类型.....	14
2.7	事件信息.....	14
2.8	错误处理.....	14
2.9	按规定使用和违规使用.....	15
<b>3</b>	<b>安全.....</b>	<b>17</b>
<b>4</b>	<b>安装.....</b>	<b>19</b>
4.1	系统要求.....	19
4.2	通过 WorkVisual 安装.....	19
4.2.1	安装或更新 KUKA.Ethernet KRL 3.2.....	19
4.2.2	卸载 KUKA.Ethernet KRL 3.2.....	20
4.3	通过 smartHMI 安装.....	20
4.3.1	安装或更新 KUKA.Ethernet KRL 3.2.....	20
4.3.2	拆卸 KUKA.Ethernet KRL 3.2.....	21
<b>5</b>	<b>配置.....</b>	<b>23</b>
5.1	通过机器人控制系统的 KLI 进行网络连接.....	23
<b>6</b>	<b>编程.....</b>	<b>25</b>
6.1	配置以太网连接.....	25
6.1.1	连接属性的 XML 结构.....	25
6.1.2	用于数据接收的 XML 结构.....	28
6.1.3	用于发送数据的 XML 结构.....	30
6.1.4	按照 XPath 架构配置.....	31
6.2	用于数据交换的函数.....	32
6.2.1	编程提示.....	33
6.2.2	初始化和删除连接.....	33
6.2.3	打开和关闭连接.....	35
6.2.4	发送数据.....	36
6.2.5	读取数据.....	38

6.2.6	删除数据存储器.....	40
6.2.7	EKI_STATUS: 函数特定返回值的结构.....	41
6.2.8	配置事件信息.....	42
6.2.9	接收完整的 XML 数据组.....	43
6.2.10	处理不完整数据组.....	43
6.2.11	EKI_CHECK(): 检查函数是否出现错误.....	44
<b>7</b>	<b>示例配置和示例程序.....</b>	<b>45</b>
7.1	接入服务器程序和示例.....	45
7.1.1	服务器程序操作界面.....	46
7.1.2	服务器程序中的通信参数.....	47
7.2	"BinaryFixed"示例.....	48
7.3	"BinaryStream"示例.....	49
7.4	"XmlTransmit"示例.....	50
7.5	"XmlServer"示例.....	51
7.6	"XmlCallback"示例.....	52
<b>8</b>	<b>诊断.....</b>	<b>57</b>
8.1	显示诊断数据.....	57
<b>9</b>	<b>信息.....</b>	<b>59</b>
9.1	错误记录 (EKI 日志).....	59
9.2	有关信息提示的说明.....	59
9.3	模块中的系统信息: EthernetKRL (EKI).....	59
9.3.1	EKI00002.....	59
9.3.2	EKI00003.....	61
9.3.3	EKI00006.....	63
9.3.4	EKI00007.....	64
9.3.5	EKI00009.....	64
9.3.6	EKI00010.....	66
9.3.7	EKI00011.....	67
9.3.8	EKI00012.....	69
9.3.9	EKI00013.....	70
9.3.10	EKI00014.....	71
9.3.11	EKI00015.....	74
9.3.12	EKI00016.....	75
9.3.13	EKI00017.....	78
9.3.14	EKI00018.....	82
9.3.15	EKI00019.....	83
9.3.16	EKI00020.....	85
9.3.17	EKI00021.....	88
9.3.18	EKI00022.....	90
9.3.19	EKI00023.....	91
9.3.20	EKI00024.....	92
9.3.21	EKI00027.....	94
9.3.22	EKI00512.....	95
9.3.23	EKI00768.....	96
9.3.24	EKI01024.....	97
9.3.25	EKI01280.....	97
9.3.26	EKI01536.....	100

9.3.27	EKI01792.....	101
9.3.28	EKI02048.....	102
<b>10</b>	<b>附录.....</b>	<b>103</b>
10.1	连接属性的扩展 XML 结构.....	103
10.2	增加内存.....	103
10.3	禁用信息输出和信息记录.....	104
10.4	指令参考.....	105
10.4.1	初始化、打开、关闭和删除连接.....	105
10.4.2	发送数据.....	106
10.4.3	写入数据.....	106
10.4.4	读取数据.....	108
10.4.5	检查函数是否出现错误.....	111
10.4.6	删除、禁用、解禁、检查数据存储器.....	112
<b>11</b>	<b>KUKA 服务.....</b>	<b>115</b>
11.1	技术支持咨询.....	115
11.2	KUKA 客户支持系统.....	115
	<b>索引</b>	<b>123</b>



1 引言

1.1 目标群体

本文件针对具有下列知识的用户：

- KRL 高级编程知识
- 机器人控制系统的专业系统知识
- 高级 XML 知识
- 高级网络知识



我们建议在 KUKA 学院接受培训，以便能以最佳方式使用 KUKA 产品。有关培训项目信息请访问公司主页 [www.kuka.com](http://www.kuka.com) 或直接在公司分支机构处获得。

1.2 工业机器人文献

工业机器人文档由以下几部分组成：

- 机器人机械装置文献
- 机器人控制系统文献
- smartPAD-2 的文件资料（如果使用）
- 系统软件操作及编程指南
- 选项及附件指南
- KUKA Xpert 中的备件概览

每份指南均独立成篇。

1.3 提示的图示

安全

这些说明是安全提示，**必须** 遵守。



**危险**

该提示表示，如果不采取预防措施，则**很可能**将导致死亡或重伤。



**警告**

该提示表示，如果不采取预防措施，则**可能**导致死亡或重伤。



**小心**

该提示表示，如果不采取预防措施，则**可能**导致轻伤。

**提示**

该提示表示，如果不采取预防措施，则**可能**导致财产损失。



该提示包含安全相关信息的说明或通用安全措施。  
该提示不针对个别的危险或个别的预防措施。

此提示提醒您注意用于预防或消除紧急情况或故障的操作步骤：

**安全须知**

必须严格遵守以下操作步骤！

必须严格遵守用此提示所标记的操作步骤。

提示

这些提示可使工作便利或提供进一步信息的说明。



用来方便工作或提供补充信息的提示。

1.4 所用概念

术语	说明
数据流	连续的数据组序列，其末尾无法提前预见： <ul style="list-style-type: none"><li>• 各个数据组具有任意而固定的类型。</li><li>• 每个时间单位的数据组量（数据率）可能有所不同。只能按顺序访问这些数据。</li></ul>
EKI	Ethernet KRL Interface（以太网 KRL 接口）
EOS	End Of Stream（末尾字符串） 标记数据组末尾的字符串
以太网	局部数据列表的数据网络技术 以太网实现使数据以数据框架的形式在所连接用户之间的交换。
FIFO	First In First Out（先进先出） 用于执行数据存储器的工艺方法 最先保存的元素将最先重新从存储器中取出。
IP	Internet Protocol（互联网协议） 互联网协议的任务是通过物理 MAC 地址定义子网络。
KLI	KUKA Line Interface（KUKA 线路接口） 机器人控制系统用于外部通讯（非实时）的以太网接口
KR C	KUKA Robot Control 机器人控制系统
KRL	KUKA Robot Language KUKA 机器人编程语言
LIFO	Last In First Out（后进先出） 用于执行数据存储器的工艺方法 最后保存的元素将最先重新从存储器中取出。
smarthMI	smart Human-Machine Interface（smart 人机界面） smartPAD 上的操作界面



smartPAD	<p>机器人控制系统的手持编程器</p> <p>smartPAD 具有工业机器人操作和编程所需的各种操作和显示功能。有以下两种模型：</p> <ul style="list-style-type: none"><li>• smartPAD</li><li>• smartPAD-2</li></ul> <p>每个模型又有诸如连接线长度不同的类型。</p> <p>除非没有明确区分，否则“KUKA smartPAD”或“smartPAD”名称指的是两种模型。</p>
套接字	将 IP 地址和端口号相互连接的软件接口
TCP	<p>Transmission Control Protocol（传输控制协议）</p> <p>有关网络用户间数据交换的协议。TCP 可在网络连接的 2 个终点之间建立一个虚拟通道。在这个通道上可双向传输数据。</p>
UDP	<p>User Datagram Protocol（用户数据报协议）</p> <p>有关网络用户间数据交换的无连接协议</p> <p>UDP 未与网络建立连接。</p>
XML	<p>Extensible Markup Language（可扩展的标记语言）</p> <p>以规定的树状结构创建人机可读文档的标准</p>
XPath	<p>XML Path Language（XML 路径语言）</p> <p>用于描述和读取 XML 文件部分的语言</p>

1.5 商标

**.NET Framework** 是微软公司（Microsoft Corporation）的商标。  
**Windows** 是微软公司（Microsoft Corporation）的商标。

1.6 许可证

KUKA 许可证条件和所用开源软件的许可证条件可在下列文件夹中找到：

- 在带 KUKA 软件安装文件的数据载体上的 .\LICENSE 下
- 机器人控制系统上安装后，在 D:\KUKA\_OPT\备选软件包名\LICENSE 下
- 在 WorkVisual 中安装后，在**选项**编目中名称为备选软件包的许可证文件夹下



关于开源许可证的更多信息可通过下面的邮件地址获取：  
opensource@kuka.com



## 2 产品说明

### 2.1 产品说明

#### 说明

KUKA.Ethernet KRL 3.2 是用于在提交解释器或机器人解释器程序运行时通过以太网在机器人控制系统和外部系统之间进行数据交换的备选软件包。

#### 功能

- 接收外部系统的 XML 数据
- 将 XML 数据发送给外部系统
- 接收外部系统的二进制数据
- 将二进制数据发送给外部系统

### 2.2 连接配置

用于数据交换的以太网连接通过一个 xml 文件进行配置。针对每个连接，必须在机器人控制系统的目录 C:\KRC\ROBOTER\Config\User\Common\EthernetKRL 中定义一个配置文件。在连接初始化时将读入该配置。

以太网连接可由机器人或提交解释器创建和操作。通道可交叉使用，例如由提交解释器打开的通道同样可由机器人解释器操作。

删除连接可与机器人或提交解释器的操作或系统操作挂钩。

交叉使用时，必须遵守特定准则，避免出现意外的程序性能。

(>>> 6.2.1 “编程提示” 页面 33)

#### 2.2.1 传输协议

数据在默认情况下通过 TCP/IP 协议传输。可以使用 UDP/IP 协议，但不建议这样做（无连接网络协议，例如无法识别数据丢失）。

#### 2.2.2 失去连接时的特性

EKI 的下列特性和功能可确保可靠地处理已接收的数据：

- 达到数据存储器的极限时，将自动关闭连接。
- 在数据接收中出错时，将自动关闭连接。
- 关闭连接时，仍继续读取错误存储器。
- 失去连接时，可恢复连接且不会影响已保存的数据。
- 可通过旗标等显示失去连接的情况。
- 对于导致失去连接的错误，可在 smartHMI 上输出错误信息。

#### 2.2.3 监控连接

通过向外部系统发送 Ping 命令，可以监控连接。（连接配置中的 <ALIVE.../> 元素）

连接成功时，根据配置不同，可给旗标或输出端赋值。只要定期发送 Ping 命令，且与外部系统的连接处于激活状态，旗标或输出端就会赋值。如果与外部系统的连接中止，则输出端或旗标将被删除。

## 2.3 数据交换

### 概览

通过 EKI，机器人控制系统既能从外部系统接收数据，也能向外部系统发送数据。

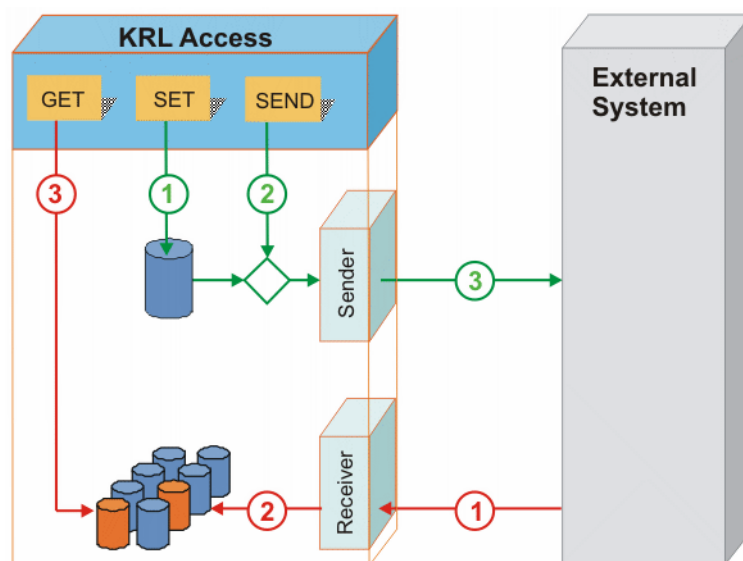


图 2-1: 系统概览

### 数据接收

基本流程（标红色）（>>> 图 2-1）：

1. 外部系统发送数据，通过协议传输数据并由 EKI 接收。
2. 数据以一定的结构储存在数据存储器中。
3. 从 KRL 程序以一定的结构存取数据。利用 KRL 指令读取数据并复制到 KRL 变量中。

### 数据发送

基本流程（标绿色）（>>> 图 2-1）：

1. 利用 KRL 指令将数据以一定的结构写入数据存储器中。
2. 用 KRL 指令从存储器中读取数据。
3. EKI 通过协议将数据发送给外部系统。



可直接发送数据，而无需事先将数据储存在存储器内。

## 2.4 数据保存

### 说明

所有已接收的数据将自动保存，从而可供 KRL 使用。保存时，对 XML 和二进制数据的处理不同。

每个数据存储器均作为后进先出存储器来执行。在 FIFO 或 LIFO 模式下读取各个存储器。

### XML 数据

接收的数据被提取出来并根据类型正确地储存到多个存储器内（每个数值储存到一个存储器）。

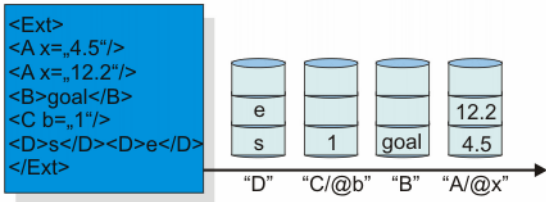


图 2-2: XML 数据存储器

二进制数据

接收的数据不被提取出来或解释。对于二进制模式下的连接，只存在一个存储器。

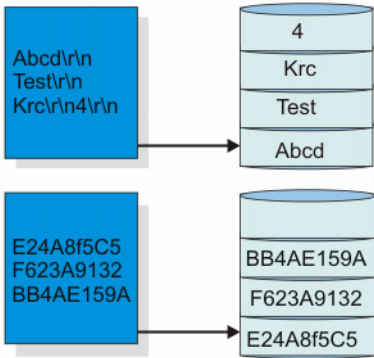


图 2-3: 二进制数据存储器

读取程序

数据元素按照储存的顺序从存储器中取出（FIFO）。与之相反的程序则可配置，即最后一次储存在存储器中的数据元素将最先取出（LIFO）。每个存储器都会分配一个最大可保存数据的共同极限。超出该极限时，将立即关闭以太网连接，以防止接收其他数据。当前接收的数据仍将保存，内存增加“1”。可继续处理存储器。可通过函数 EKI\_OPEN() 重新打开连接。

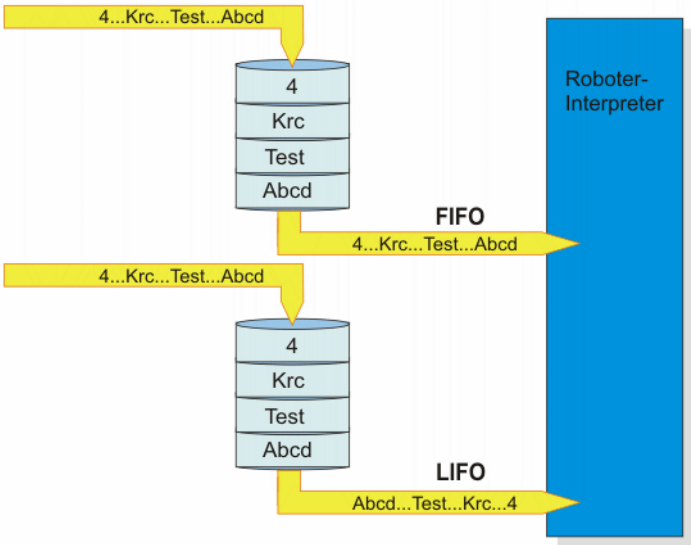


图 2-4: 读取程序概览

## 2.5 客户端-服务器运行模式

机器人控制系统和外部系统作为客户端和服务端相互连接。其中，外部系统可以是客户端或服务端。激活的连接数量限制在 16 个。

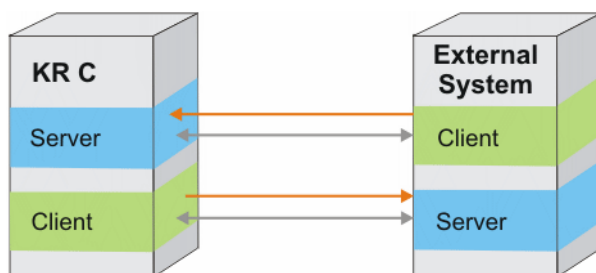


图 2-5：客户端-服务器运行模式

将 EKI 配置为服务器时，只能有一个单独的客户端与服务器连接。如果需要多个连接，则还要创建多台 EKI 服务器。可以同时 EKI 内部运行多个客户端和服务端。

## 2.6 记录类型

传输的数据可打包为不同格式。

支持下列格式：

- 可任意配置的 XML 结构
- 有固定长度的二进制数据组
- 带末尾字符串的可变二进制数据组

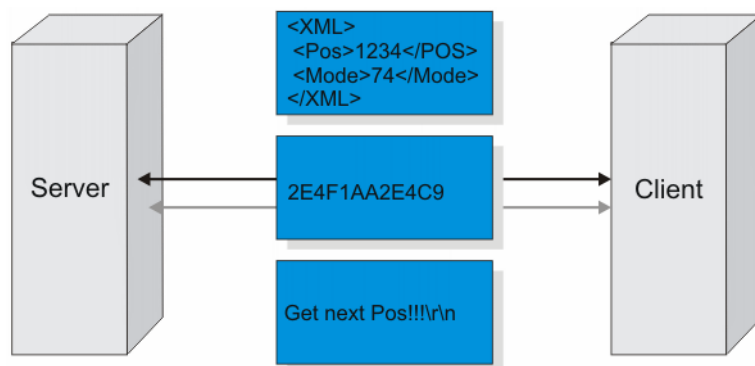


图 2-6：记录类型

## 2.7 事件信息

通过给输出端或旗标赋值，可报告下列事件：

- 连接已激活。
- 出现一个单独的 XML 元素。
- 出现一个完整的 XML 结构或一个完整的二进制数据组。

(>>> [6.2.8 “配置事件信息” 页面 42](#))

## 2.8 错误处理

KUKA.Ethernet KRL 3.2 提供相应的函数用于在机器人控制系统和外部系统之间数据交换。

这些函数的每一个都会返回值。可以在 KRL 程序中查询和分析这些返回值。根据函数不同，将返回下列值：

- 错误编号
- 存取后存储器中仍存在的元素数量
- 从存储器中读取的元素数量
- 是否存在连接的信息
- 从存储器中取出的数据元素的时间戳

(>>> 6.2.7 “EKI\_STATUS: 函数特定返回值的结构” 页面 41)

对于每个错误都会在 smartHMI 上和 EKI 日志中输出一条信息。可禁用信息的自动输出。

## 2.9 按规定使用和违规使用

### 使用

备选软件包 KUKA.Ethernet KRL 3.2 是为在提交解释器或机器人解释器程序运行时通过以太网在机器人控制系统和外部系统之间进行数据交换而设计的。

### 违规使用

所有不符合规定的使用都属于违规使用并且均被禁止。生产商对因此引起的损失不承担任何责任。这种风险须由运营者单独承担。

违规使用例如包括：

- 通过以太网通讯不能完全替代现场总线系统，并且不是为操作外部自动运行接口而设计的。
- KUKA.Ethernet KRL 3.2 无法确保确定性的时间特性，也就是说，不能认为调出的数据在经过恒定的延迟后即可使用。
- KUKA.Ethernet KRL 3.2 不适用于建立按机器人控制系统实时节拍的循环通讯。





### 3 安全

该文件包含针对此处所述选项包的安全提示。

有关工业机器人的基本安全信息请参阅系统集成商操作和编程指南或最终用户操作及编程指南中的安全一章。

**注意与安全相关的信息**

安全使用本产品需要掌握知识和遵守基本安全措施。否则有可能造成死亡、严重身体伤害或财产损失。

- 必须注意遵守系统软件操作及编程指南中的“安全”一章。



## 4 安装

该备选软件包可以通过 smarthMI 或通过 WorkVisual 安装到机器人控制系统中。

### 4.1 系统要求

#### 硬件

- KR C5 机器人控制系统

#### 软件

- KUKA System Software 8.7
- WorkVisual 6.0

### 4.2 通过 WorkVisual 安装

#### 4.2.1 安装或更新 KUKA.Ethernet KRL 3.2

#### 说明

在 WorkVisual 中安装选项包并添加到项目中。在传输项目时，自动将该选项包安装在机器人控制系统上。



我们建议在软件更新前将所有相关数据存档。

#### 前提条件

- 专家用户组
- 运行模式 T1 或 T2
- 没有选定任何程序。
- 机器人控制系统的网络连接
- 已有 KOP 文件形式的备选软件包。

#### 操作步骤

1. 在 WorkVisual 中安装备选软件包 KUKA.Ethernet KRL 3.2。
2. 从机器人控制系统载入激活的项目。
3. 将备选软件包 KUKA.Ethernet KRL 3.2 添加到项目中。
4. 根据需要在 WorkVisual 中配置该备选软件包。
5. 将项目从 WorkVisual 传输到机器人控制系统上并激活。
6. 在 smarthMI 上显示安全询问“允许激活项目 [...] 吗？”。激活时，当前活动的项目被覆盖。如果没有相关的项目要覆盖：点击是确认询问。
7. 此时 smarthMI 上显示修改概览以及一条安全询问。用是回答该询问。该备选软件包被安装并且机器人控制系统执行重启。



关于 WorkVisual 中流程步骤的信息见 WorkVisual 的文件资料。

#### 日志文件

C:\KRC\ROBOTER\LOG 下生成一个 LOG 文件。

## 4.2.2 卸载 KUKA.Ethernet KRL 3.2

### 说明

可通过 WorkVisual 卸载选项包。



我们建议在软件卸载前将所有相关数据存档。

### 前提条件

- 专家用户组
- 运行模式 T1 或 T2
- 没有选定任何程序。
- 机器人控制系统的网络连接

### 操作步骤

1. 从机器人控制系统载入项目。
2. 从项目中删除备选软件包 KUKA.Ethernet KRL 3.2。显示带更改的窗口。
3. 将项目从 WorkVisual 传输到机器人控制系统上并激活。
4. 点击 **是** 确认 smartHMI 上的安全询问“允许激活项目 [...] 吗?”。
5. 此时 smartHMI 上显示修改概览以及一条安全询问。用 **是** 答复该询问。卸载该备选软件包并且机器人控制系统执行重启。



关于 WorkVisual 中流程步骤的信息见 WorkVisual 的文件资料。

### 日志文件

C:\KRC\ROBOTER\LOG 下生成一个 LOG 文件。

## 4.3 通过 smartHMI 安装

### 4.3.1 安装或更新 KUKA.Ethernet KRL 3.2



我们建议在软件更新前将所有相关数据存档。

### 前提条件

- 用户权限：功能组**一般配置**  
但至少是专家用户组
- 运行模式 T1 或 T2
- 没有选定任何程序。
- 含备选软件包（KOP 文件）的 U 盘

#### 提示

#### 第三方制造商的 U 盘可能导致数据丢失

在机器人控制系统上操作时如果使用 KUKA 以外其他制造商的 U 盘，可能导致数据丢失。

- 对于机器人控制系统上需要 U 盘的操作，请使用 KUKA U 盘。  
KUKA U 盘已获得验证，可在机器人控制系统上使用。

### 操作步骤

1. 将 U 盘插在机器人控制系统或 smartPAD 上。
2. 在主菜单中选择**投入运行** > **辅助软件**。

3. 按下**新软件：名称** 列中必须显示条目 “KUKA.Ethernet KRL 3.2”，**路径** 列中必须显示驱动器 **E:\** 或 **K:\**。  
否则按下**刷新**。
4. 如果此时显示上述条目，则继续进行步骤 5。  
否则必须先配置待安装程序的路径：
  - a. 按下按钮**配置**。
  - b. 在区域**选项的安装路径**中选中一个行。  
**提示：**如果该行已经包含一个路径，则该路径被盖写。
  - c. 点击 **路径选择**。即显示现有的驱动器。
  - d. 如果 U 盘插接到机器人控制系统上：在 **E:\**上选中软件所在的目录。  
如果 U 盘插接到 smartPAD 上：则 **K:\** 替代 **E:\**
  - e. 点击 **保存**。再次显示区域**选项的安装路径**。它此时含有新的路径。
  - f. 选中带有新路径的行并再次按下**保存**。
5. 勾选 “KUKA.Ethernet KRL 3.2”，然后点击 **安装**。点击 **OK** 确认安装提示。
6. 将显示安全询问 **允许激活项目 [...] 吗？**。激活时，当前活动的项目被覆盖。如果没有相关的项目要覆盖：点击**是**确认询问。
7. 此时将显示修改概览以及一条安全询问。用**是**回答该询问。该备选软件包被安装并且机器人控制系统执行重启。
8. 拔出 U 盘。

日志文件

C:\KRC\ROBOTER\LOG 下生成一个 LOG 文件。

4.3.2 拆卸 KUKA.Ethernet KRL 3.2



我们建议在软件卸载前将所有相关数据存档。

前提条件

- 用户权限：功能组**一般配置**  
但至少是专家用户组
- 运行模式 T1 或 T2
- 没有选定任何程序。

操作步骤

1. 在主菜单中选择**投入运行 > 辅助软件**。
2. 勾选 KUKA.Ethernet KRL 3.2，然后点击 **卸载**。点击**是**确认安全询问。
3. 点击 **是** 确认安全询问 “**允许激活项目 [...] 吗？**”。
4. 此时将显示修改概览以及一条安全询问。用 **是** 答复该询问。卸载该备选软件包并且机器人控制系统执行重启。

日志文件

C:\KRC\ROBOTER\LOG 下生成一个 LOG 文件。



## 5 配置

### 5.1 通过机器人控制系统的 KLI 进行网络连接

在通过以太网进行数据交换时，必须通过机器人控制系统的 KLI 建立一个网络连接。



有关 KLI 网络配置的详细信息可在系统软件的系统集成商的操作及编程指南中找到。



可用的 KLI 以太网接口详细信息请见机器人控制系统的操作或安装指南。





## 6 编程

### 6.1 配置以太网连接

以太网连接通过 xml 文件进行配置。针对每个连接，必须在机器人控制系统的目录 C:\KRC\ROBOTER\Config\User\Common\EthernetKRL 中定义一个配置文件。



xml 文件都是“区分大小写”的。必须注意区分大小写。

xml 文件的名称同时也是 KRL 的访问键。  
示例: ...\EXT.xml → EKI\_INIT("EXT")

```
<ETHERNETKRL>
<CONFIGURATION>
  <EXTERNAL></EXTERNAL>
  <INTERNAL></INTERNAL>
</CONFIGURATION>
<RECEIVE>
  <ELEMENTS></ELEMENTS>
</RECEIVE>
<SEND>
  <ELEMENTS></ELEMENTS>
</SEND>
</ETHERNETKRL>
```

段	说明
<CONFIGURATION> ... </CONFIGURATION>	配置外部系统和 EKI 之间的连接参数 (>>> 6.1.1 “连接属性的 XML 结构” 页面 25)
<RECEIVE> ... </RECEIVE>	配置机器人控制系统接收的接收结构 (>>> 6.1.2 “用于数据接收的 XML 结构” 页面 28)
<SEND> ... </SEND>	配置机器人控制系统发送的发送结构 (>>> 6.1.3 “用于发送数据的 XML 结构” 页面 30)

#### 6.1.1 连接属性的 XML 结构

##### 说明

在段落 <EXTERNAL> ... </EXTERNAL> 中定义外部系统的设置:

元素	说明
TYPE	<p>确定外部系统是作为服务器还是客户端与机器人控制系统通讯（可选）</p> <ul style="list-style-type: none"> <li>• <b>Server:</b> 外部系统是服务器。</li> <li>• <b>Client:</b> 外部系统是客户端。</li> </ul> <p>默认值: <b>服务器</b></p>
IP	<p>外部系统定义为服务器时的 IP 地址 (TYPE = 服务器)</p> <p>如果 TYPE = 客户端, 将忽略 IP 地址。</p>
PORT	<p>外部系统定义为服务器时的端口号 (TYPE = 服务器)</p> <ul style="list-style-type: none"> <li>• 1 ... 65534</li> </ul> <p><b>提示:</b> 选择端口时须注意, 不能有其他服务 (如操作系统) 使用该端口。否则无法通过该端口建立连接。</p> <p>如果 TYPE = 客户端, 将忽略端口号。</p>

在段 <INTERNAL> ... </INTERNAL> 中定义 EKI 的设置:

元素	属性	说明
ENVIRONMENT	———	<p>将删除连接与操作相结合（可选）</p> <ul style="list-style-type: none"> <li>• <b>Program:</b> 在机器人解释器操作后删除连接: <ul style="list-style-type: none"> <li>– 程序复位。</li> <li>– 取消程序。</li> <li>– 重新配置输入/输出端。</li> </ul> </li> <li>• <b>Submit:</b> 在系统提交解释器操作后删除连接: <ul style="list-style-type: none"> <li>– 取消选择系统提交解释器。</li> <li>– 重新配置输入/输出端。</li> </ul> </li> <li>• <b>ExtSubmit1, ExtSubmit2 ... ExtSubmit7:</b> 设置的 Extended 提交解释器操作后删除连接: <ul style="list-style-type: none"> <li>– 取消选择 Extended 提交解释器。</li> <li>– 重新配置输入/输出端。</li> </ul> </li> <li>• <b>System:</b> 在系统操作后删除连接 <ul style="list-style-type: none"> <li>– 重新配置输入/输出端。</li> </ul> </li> </ul> <p>默认值: <b>Program</b></p>
BUFFERING	Mode	<p>处理所有数据存储器的方法（可选）</p> <ul style="list-style-type: none"> <li>• <b>FIFO:</b> First In First Out (先进先出)</li> <li>• <b>LIFO:</b> Last In First Out (后进先出)</li> </ul> <p>默认值: <b>FIFO</b></p>
	Limit	<p>数据存储器能够容纳的数据元素最大数量（可选）</p> <ul style="list-style-type: none"> <li>• 1 ... 512</li> </ul> <p>默认值: <b>16</b></p>

元素	属性	说明
BUFSIZE	Limit	能够接收而无需解释的字节最大数量（可选）  • 1 ... 65,534 字节  默认值：16,384 字节
TIMEOUT	Connect	尝试建立连接的时间，超过这段时间后将中止尝试（可选）  单位：ms  • 0 ... 65,534 ms  默认值：2,000 ms
ALIVE	Set_Out	连接成功时给输出端或旗标赋值（可选）  输出端编号：  • 1 ... 4,096  旗标编号：  • 1 ... 1,024  只要有与外部系统的连接处于激活状态，旗标或输出端就处于已赋值状态。如果与外部系统的连接中止，则输出端或旗标将被删除。
	Set_Flag	
	Ping	发送监控与外部系统连接的 Ping 命令的周期（可选）  • 1 ... 65,534 s
IP	———	EKI 定义为服务器时的 IP 地址（EXTERNAL/TYPE = Client）  如果 EXTERNAL/TYPE = Server，将忽略 IP 地址。
PORT	———	EKI 定义为服务器时的端口号（EXTERNAL/TYPE = Client）  • 54,600 ... 54,615  如果 EXTERNAL/TYPE = Server，将忽略端口号。
PROTOCOL	———	传输协议（可选）  • TCP • UDP  默认值：TCP  建议始终使用 TCP/IP 协议。

元素	属性	说明
MESSAGES	Logging	禁用在 EKI 日志中写入信息（可选）。 <ul style="list-style-type: none"><li>• <b>warning:</b> 记录警告信息和错误信息。</li><li>• <b>error:</b> 仅记录错误信息。</li><li>• <b>disabled:</b> 记录已禁用。</li></ul> 默认值: <b>error</b>
	显示屏	禁用 smartHMI 上的输出信息（可选）。 <ul style="list-style-type: none"><li>• <b>error:</b> 信息输出激活。</li><li>• <b>disabled:</b> 信息输出已被禁用。</li></ul> 默认值: <b>error</b>
	(>>> 10.3 “禁用信息输出和信息记录” 页面 104)	

示例

```
<CONFIGURATION>
<EXTERNAL>
  <IP>172.1.10.5</IP>
  <PORT>60000</PORT>
  <TYPE>Server</TYPE>
</EXTERNAL>
<INTERNAL>
  <ENVIRONMENT>Program</ENVIRONMENT>
  <BUFFERING Mode="FIFO" Limit="10"/>
  <BUFSIZE Limit="16384"/>
  <TIMEOUT Connect="60000"/>
  <ALIVE Set_Out="666" Ping="200"/>
  <IP>192.1.10.20</IP>
  <PORT>54600</PORT>
  <PROTOCOL>TCP</PROTOCOL>
  <MESSAGES Logging="error" Display="disabled"/>
</INTERNAL>
</CONFIGURATION>
```

6.1.2 用于数据接收的 XML 结构

说明

配置取决于是接收 XML 数据还是二进制数据。

- 如果是接收 XML 数据，必须定义 XML 结构: <XML> ... </XML>
- 如果是接收二进制数据，必须定义原始数据: <RAW> ... </RAW>

XML 结构 <XML> ... </XML> 的元素中的属性:

元素	属性	说明
ELEMENT	日	元素名称 在此定义用于接收数据的 XML 结构 (XPath)。 (>>> 6.1.4 “按照 XPath 架构配置” 页面 31)
ELEMENT	型号	元素的数据类型 <ul style="list-style-type: none"><li>• STRING</li><li>• REAL</li><li>• INT</li><li>• BOOL</li><li>• FRAME</li></ul> <b>提示：</b> 当标签仅用于事件信息时，可选。在这种情况下不为该元素预留存储空间。 <b>事件旗标示例：</b> <ELEMENT Tag="Ext" Set_Flag="56"/>
ELEMENT	Set_Out	接收到元素时给输出端或旗标赋值（可选） 输出端编号： <ul style="list-style-type: none"><li>• 1 ... 4096</li></ul> 旗标编号： <ul style="list-style-type: none"><li>• 1 ... 1024</li></ul> <b>提示：</b> 如果通过 Set_Out/Set_Flag 设置一个输出端或插上一个旗标，那么程序代码中的相应系统变量 \$OUT[]/\$FLAG[] 必须被重新复位。
	Set_Flag	
ELEMENT	Mode	处理数据存储器中的数据组的方法 <ul style="list-style-type: none"><li>• FIFO: 先进先出</li><li>• LIFO: 后进先出</li></ul> 只有当处理各个数据组的方法与 BUFFERING 下为 EKI 配置的不同时才相关。

原始数据 <RAW> ... </RAW> 中的元素的属性:

元素	属性	说明
ELEMENT	日	元素名称
ELEMENT	型号	元素的数据类型 <ul style="list-style-type: none"><li>• BYTE: 有固定长度的二进制数据组</li><li>• STREAM: 有可变末尾字符串和可变长度的二进制数据组</li></ul>
ELEMENT	Set_Out	接收到元素时给输出端或旗标赋值（可选） 输出端编号： <ul style="list-style-type: none"><li>• 1 ... 4096</li></ul> 旗标编号： <ul style="list-style-type: none"><li>• 1 ... 1024</li></ul> <b>提示：</b> 如果通过 Set_Out/Set_Flag 设置一个输出端或插上一个旗标，那么程序代码中的相应系统变量 \$OUT[]/\$FLAG[] 必须被重新复位。
	Set_Flag	

元素	属性	说明
ELEMENT	EOS	基本信息的末尾字符串（只有当 Type = "STREAM" 时，才相关）  • ASCII 编码：1 … 32 个字符 • 备选结尾通过符号 " " 隔开。  示例：  • <ELEMENT … EOS="123, 134, 21"/> • <ELEMENT … EOS="123, 134, 21 I 13, 10"/>
ELEMENT	Size	当 Type = "BYTE" 时，信息的固定大小  • 1 … 3600 字节

示例

```
<RECEIVE>
<XML>
  <ELEMENT Tag="Ext/Str" Type="STRING"/>
  <ELEMENT Tag="Ext/Pos/XPos" Type="REAL" Mode="LIFO"/>
  <ELEMENT Tag="Ext/Pos/YPos" Type="REAL"/>
  <ELEMENT Tag="Ext/Pos/ZPos" Type="REAL"/>
  <ELEMENT Tag="Ext/Temp/Cpu" Type="REAL" Set_Out="1"/>
  <ELEMENT Tag="Ext/Temp/Fan" Type="REAL" Set_Flag="14"/>
  <ELEMENT Tag="Ext/Integer/AState" Type="INT"/>
  <ELEMENT Tag="Ext/Integer/BState" Type="INT"/>
  <ELEMENT Tag="Ext/Boolean/CState" Type="BOOL"/>
  <ELEMENT Tag="Ext/Frames/Frame1" Type="FRAME"/>
  <ELEMENT Tag="Ext/Attributes/@A1" Type="STRING"/>
  <ELEMENT Tag="Ext/Attributes/@A2" Type="INT"/>
  <ELEMENT Tag="Ext" Set_Flag="56"/>
</XML>
</RECEIVE>
```

```
<RECEIVE>
<RAW>
  <ELEMENT Tag="RawData" Type="BYTE" Size="1408"
    Set_Flag="14"/>
</RAW>
</RECEIVE>
```

```
<RECEIVE>
<RAW>
  <ELEMENT Tag="MyStream" Type="STREAM" EOS="123, 134, 21"
    Size="836" Set_Flag="14"/>
</RAW>
</RECEIVE>
```

6.1.3 用于发送数据的 XML 结构

说明

配置取决于是否发送 XML 数据还是二进制数据。

- 如果是发送 XML 数据，必须定义 XML 结构：<XML> … </XML>



发送时，XML 结构将按照配置的顺序进行创建。

- 发送二进制数据直接在 KRL 编程中实现。无须指定配置。

XML 结构 <XML> ... </XML> 的元素中的属性:

属性	说明
日	元素名称 在此定义用于发送数据的 XML 结构 (XPath) 。

示例

```
<SEND>
<XML
  <ELEMENT Tag="Robot/Data/ActPos/@X"/>
  <ELEMENT Tag="Robot/Data/ActPos/@Y"/>
  <ELEMENT Tag="Robot/Data/ActPos/@Z"/>
  <ELEMENT Tag="Robot/Data/ActPos/@A"/>
  <ELEMENT Tag="Robot/Data/ActPos/@B"/>
  <ELEMENT Tag="Robot/Data/ActPos/@C"/>
  <ELEMENT Tag="Robot/Status"/>
  <ELEMENT Tag="Robot/Mode"/>
  <ELEMENT Tag="Robot/Complex/Tickcount"/>
  <ELEMENT Tag="Robot/RobotType/Robot/Type"/>
</XML>
</SEND>
```

6.1.4 按照 XPath 架构配置

说明

使用 XML 进行数据交换时，交换的 XML 文件必须按照相同架构构建。  
Ethernet KRL 使用 XPath 架构进行 XML 文件的描述和读取。

根据 XPath 需区分下列情况：

- 描述和读取元素
- 描述和读取属性

元素写法

- 保存的用于发送数据的 XML 文件：

```
<Robot>
  <Mode>...</Mode>
  <RobotLamp>
    <GrenLamp>
      <LightOn>...</LightOn>
    </GrenLamp>
  </RobotLamp>
</Robot>
```

- 为发送数据所配置的 XML 结构：

```
<SEND>
<XML>
  <ELEMENT Tag="Robot/Mode" />
  <ELEMENT Tag="Robot/RobotLamp/GrenLamp/LightOn" />
</XML>
<SEND />
```

属性写法

- 保存的用于发送数据的 XML 文件：

```
<Robot>
  <Data>
    <ActPos X="...">
  </ActPos>
  <LastPos A="..." B="..." C="..." X="..." Y="..." Z="...">
  </LastPos>
  </Data>
</Robot>
```

- 为发送数据所配置的 XML 结构:

```
<SEND>
  <XML>
    <ELEMENT Tag="Robot/Data/LastPos/@X" />
    <ELEMENT Tag="Robot/Data/LastPos/@Y" />
    ...
    <ELEMENT Tag="Robot/Data/ActPos/@X" />
  </XML>
<SEND />
```

6.2 用于数据交换的函数

EKI 提供相应的函数用于在机器人控制系统和外部系统之间数据交换。  
函数的详细说明可参见附录。（>>> 10.4 “指令参考” 页面 105）


初始化、打开、关闭和删除连接
EKI_STATUS = EKI_Init (CHAR[])
EKI_STATUS = EKI_Open (CHAR[])
EKI_STATUS = EKI_Close (CHAR[])
EKI_STATUS = EKI_Clear (CHAR[])
发送数据
EKI_STATUS = EKI_Send (CHAR[], CHAR[], INT)
写入数据
EKI_STATUS = EKI_SetReal (CHAR[], CHAR[], REAL)
EKI_STATUS = EKI_SetInt (CHAR[], CHAR[], INT)
EKI_STATUS = EKI_SetBool (CHAR[], CHAR[], BOOL)
EKI_STATUS = EKI_SetFrame (CHAR[], CHAR[], FRAME)
EKI_STATUS = EKI_SetString (CHAR[], CHAR[], CHAR[])
读取数据
EKI_STATUS = EKI_GetBool (CHAR[], CHAR[], BOOL)
EKI_STATUS = EKI_GetBoolArray (CHAR[], CHAR[], BOOL[])
EKI_STATUS = EKI_GetInt (CHAR[], CHAR[], INT)
EKI_STATUS = EKI_GetIntArray (CHAR[], CHAR[], INT[])
EKI_STATUS = EKI_GetReal (CHAR[], CHAR[], REAL)
EKI_STATUS = EKI_GetRealArray (CHAR[], CHAR[], REAL[])
EKI_STATUS = EKI_GetString (CHAR[], CHAR[], CHAR[])
EKI_STATUS = EKI_GetFrame (CHAR[], CHAR[], FRAME)
EKI_STATUS = EKI_GetFrameArray (CHAR[], CHAR[], FRAME[])




检查函数是否出现错误
EKI_CHECK(EKI_STATUS, EKrlMsgType, CHAR[])
删除、禁用、解禁、检查数据存储器
EKI_STATUS = EKI_Clear(CHAR[])
EKI_STATUS = EKI_ClearBuffer(CHAR[], CHAR[])
EKI_STATUS = EKI_Lock(CHAR[])
EKI_STATUS = EKI_Unlock(CHAR[])
EKI_STATUS = EKI_CheckBuffer(CHAR[], CHAR[])

6.2.1 编程提示

- 在提交解释器中创建连接时必须注意下列事项：
  - 在连接配置中必须通过元素 <ENVIRONMENT> 说明涉及哪个提交通道。
  - 在提交解释器中打开的通道同样可由机器人解释器触发。
  - 如果取消选择提交解释器，将通过配置自动删除连接。
- 当提交解释器和机器人解释器同时访问一个通道时，需注意下列事项：
  - 通过 EKI\_Get...() 读取数据时，将删除该存储器的内容。因此，不建议同时通过提交解释器和机器人解释器读取数据。
  - 提交解释器和机器人解释器可用 EKI\_Set...() 以写入的方式访问同一个通道。当访问同一个目标元素，且在此期间未调用 EKI\_Send() 时，这些调用操作会相互覆盖这些数据。
  - 不同解释器内的执行时间点会相互发生变化，使得竞相进行读写访问时，性能可能意外改变。



建议为每个发送和接收任务打开一个单独的通道。



在预进中执行 EKI 指令！

- 如果要在主进中运行 EKI 指令，必须使用会触发预进停止的指令，如 WAIT SEC。
- 由于每次访问存储器都需花费时间，因此建议用数组存取函数 EKI\_Get...Array() 调出较大的数据量。
- 通过函数 EKI\_Get...Array() 可以访问最多 512 个数组元素。尽管可在 KRL 中创建较大数组，如 myFrame[1000]，但始终只有最多 512 个元素可读。
- 等待数据可有多种途径：
  - 通过给旗标或输出端赋值，可显示接收了某个特定的数据元素或一个完整的数据组。（用于接收数据的 XML 结构中的元素 Set\_Flag 或 Set\_Out）。
  - 示例：通过 WAIT FOR \$FLAG[x] 中断 KRL 程序  
(>>> 7.6 “XmlCallback”示例 页面 52)
  - 通过函数 EKI\_CheckBuffer() 可周期性查询存储器中是否有新的数据元素。

6.2.2 初始化和删除连接

说明

- 必须用函数 EKI\_Init() 创建一个连接并初始化。

- 此时将读入函数中给出的连接配置。
- 既可在机器人解释器，也可在提交解释器中创建一个连接。
- 可通过函数 `EKI_Clear()` 重新删除一个连接。
  - 既可在机器人解释器，也可在提交解释器中删除一个连接。
- 可选择通过连接配置中的 `<ENVIRONMENT>` 元素配置：
  - 删除连接可与机器人解释器或提交解释器的操作或系统操作挂钩。

(>>> 6.1.1 “连接属性的 XML 结构” 页面 25)

#### ENVIRONMENT “Program”

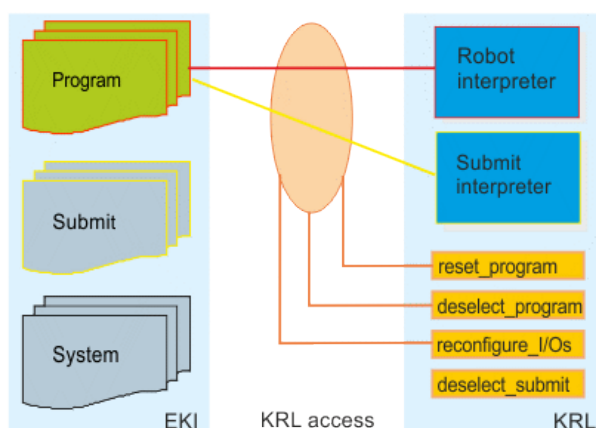


图 6-1: 配置 “Program”

进行该配置时，将在机器人解释器的下列操作后删除一个连接：

- 程序复位。
- 取消程序。
- 重新配置输入/输出端。



重新配置输入/输出端时，将重新载入驱动程序并删除所有初始化。

#### ENVIRONMENT “Submit” 或 “ExtSubmit1” ... “ExtSubmit7”

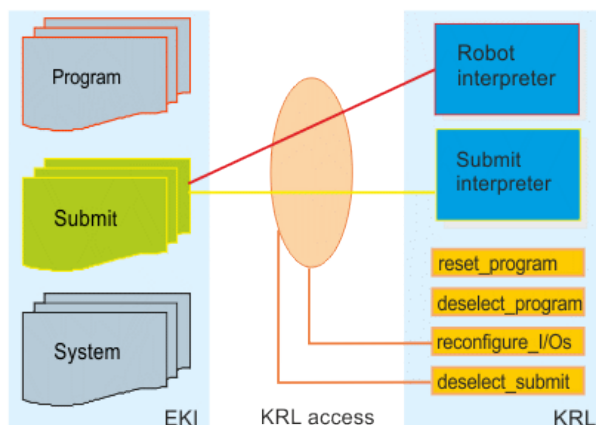


图 6-2: 配置 “Submit”

进行该配置时，将在所设置提交解释器的下列操作后删除一个连接：

- 反选提交解释器。
- 重新配置输入/输出端。



重新配置输入/输出端时，将重新载入驱动程序并删除所有初始化。

ENVIRONMENT“系统”

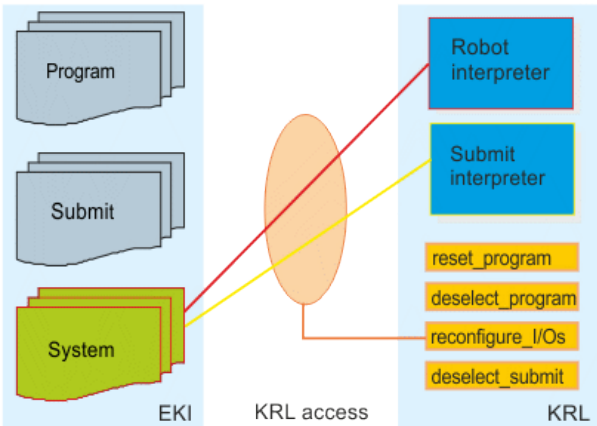


图 6-3：配置 “System”

进行该配置时，将在下列系统操作后删除一个连接：

- 重新配置输入/输出端。



重新配置输入/输出端时，将重新载入驱动程序并删除所有初始化。

6.2.3 打开和关闭连接

说明

通过一个 KRL 程序建立与外部系统的连接。大多数 KRL 程序架构如下：

```
1 DEF Connection()
2 ...
3 RET=EKI_Init("Connection")
4 RET=EKI_Open("Connection")
5 ...
6 Write data, send data or get received data
7 ...
8 RET=EKI_Close("Connection")
9 RET=EKI_Clear("Connection")
10 ...
11 END
```

行	说明
3	EKI_Init() 将 EKI 与外部系统连接的通道初始化。
4	EKI_Open() 打开通道。
6	用于将数据写入存储器、发送数据或访问已接收的数据的 KRL 指令
8	EKI_Close() 关闭通道。
9	EKI_Clear() 删除通道。

编程时须注意 EKI 是配置为服务器还是客户端。

服务器运行模式

当外部系统配置为客户端时，EKI\_Open() 会将 EKI (= 服务器) 置于监听状态。服务器等待客户端的连接问询，但不中断程序运行。如果配置文件中未描述元素 <TIMEOUT Connect="..." />，服务器将一直等待，直至客户端请求连接为止。

客户端的连接问询通过访问 EKI 或通过事件信息表示，例如通过元素 `<ALIVE SET_OUT="...">`。

如果在服务器期待连接问询的过程中要中断程序运行，必须编程设定一个事件旗标或输出端，例如 `WAIT FOR $OUT[...]`。



建议在服务器运行模式下不要使用 `EKI_Close()`。在服务器运行模式下，将从外部客户端关闭通道。

## 客户端运行模式

当外部系统配置为服务器时，`EKI_Open()` 会中断程序运行，直至与外部系统的连接激活为止。`EKI_Close()` 关闭与外部服务器的连接。

### 6.2.4 发送数据

#### 说明

根据配置和编程不同，可用 `EKI_Send()` 发送下列数据：

通过 XML 结构通讯时：

- 完整的 XML 结构
- XML 结构的一部分
- 可变长度的任意字符串

通过原始数据通讯时：

- 对于有固定长度的二进制数据组（属性 `Type = "BYTE"`）：有固定长度的任意字符串
  - 用字节表示的字符串大小必须与配置的属性 `Size`（大小）完全相符。超出时将输出错误信息，低于时将输出警告信息。
  - 必须在 KRL 程序中用 `CAST_TO()` 读入固定长度的二进制数据组。只可读取类型为 `REAL`（4 字节）的数据，非 `Double`（双精度）型。



关于指令 `CAST_TO()` 的详细信息参见文献 `CREAD/CWRITE`。

- 对于有可变末尾字符串的二进制数据组（属性 `Type = "STREAM"`）：有可变长度的任意字符串  
自动一并发送末尾字符串。

#### 示例

发送 XML 数据：

发送完整的 XML 结构：

- 保存的用于发送数据的 XML 结构：

```
<Robot>
  <ActPos X="1000.12"></ActPos>
  <Status>12345678</Status>
</Robot>
```

- 编程：

```
DECL EKI_STATUS RET
RET=EKI_Send("Channel_1","Robot")
```

- 发送的 XML 结构：

```
<Robot>
  <ActPos X="1000.12"></ActPos>
```

```
<Status>12345678</Status>
</Robot>
```

#### 发送 XML 结构的一部分：

- 保存的用于发送数据的 XML 结构：

```
<Robot>
  <ActPos X="1000.12"></ActPos>
  <Status>12345678</Status>
</Robot>
```

- 编程：

```
DECL EKI_STATUS RET
RET=EKI_Send("Channel_1", "Robot/ActPos")
```

- 发送的 XML 结构：

```
<Robot>
  <ActPos X="1000.12"></ActPos>
</Robot>
```

#### 直接发送字符串：

- 保存的用于发送数据的 XML 结构（直接发送时不使用）：

```
<Robot>
  <ActPos X="1000.12"></ActPos>
  <Status>12345678</Status>
</Robot>
```

- 编程：

```
DECL EKI_STATUS RET
RET=EKI_Send("Channel_1", "<POS><XPOS>1</XPOS></POS>")
```

- 发送的字符串：

```
<POS><XPOS>1</XPOS></POS>
```

#### 发送二进制数据：

##### 发送固定长度（10 字节）的二进制数据组：

- 配置的原始数据：

```
<RAW>
  <ELEMENT Tag="Buffer" Type="BYTE" Size="10" />
</RAW>
```

- 编程：

```
DECL EKI_STATUS RET
CHAR Bytes[10]
OFFSET=0
CAST_TO(Bytes[], OFFSET, 91984754, 913434.2, TRUE, "X")
RET=EKI_Send("Channel_1", Bytes[])
```

- 发送的数据：

```
"r{ ? _I X"
```

##### 发送有末尾字符串和可变长度的二进制数据组：

- 配置的原始数据:

```
<RAW>
  <ELEMENT Tag="Buffer" Type="STREAM" EOS="65, 66" />
</RAW>
```

- 编程:

```
DECL EKI_STATUS RET
CHAR Bytes[64]
Bytes[]="Stream ends with:"
RET=EKI_Send("Channel_1",Bytes[])
```

- 发送的数据:

```
"Stream ends with:AB"
```

发送有末尾字符串和可变长度、限制为字符最大值的二进制数据组:

- 配置的原始数据:

```
<RAW>
  <ELEMENT Tag="Buffer" Type="STREAM" EOS="65, 66" />
</RAW>
```

- 编程:

```
DECL EKI_STATUS RET
CHAR Bytes[64]
Bytes[]="Stream ends with:"
RET=EKI_Send("Channel_1",Bytes[],6)
```

- 发送的数据:

```
"StreamAB"
```

## 6.2.5 读取数据



读取数据时，必须通过数值赋值等方式，将相应的 KRL 变量初始化。

### 说明

保存和读取数据时，对 XML 和二进制数据的处理不同:

- XML 数据被 EKI 提取出来并根据类型储存在多个存储器内。可单独存取每个保存的数值。  
如要读取 XML 数据，可使用所有存取函数 EKI\_Get...()。
- 二进制数据组不被 EKI 解释，而是作为整体储存在存储器内。  
如要从存储器中读取二进制数据组，必须使用存储函数 EKI\_GetString()。以字符串形式从存储器中读取二进制数据组。  
必须在 KRL 程序中用 CAST\_FROM() 将固定长度的二进制数据组重新分为单个变量。



关于指令 CAST\_FROM() 的详细信息请参见文献 CREAD/CWRITE。

### 示例

**读取 XML 数据:**

用于数据接收的 XML 结构:

```

<RECEIVE>
  <XML>
    <ELEMENT Tag="Sensor/Message" Type="STRING" />
    <ELEMENT Tag="Sensor/Status/IsActive" Type="BOOL" />
  </XML>
</RECEIVE>

```

编程:

```

; Declaration
INT i
DECL EKI_STATUS RET
CHAR valueChar[256]
BOOL valueBOOL

; Initialization
FOR i=(1) TO (256)
  valueChar[i]=0
ENDFOR
valueBOOL=FALSE

RET=EKI_GetString("Channel_1", "Sensor/Message", valueChar[])
RET=EKI_GetBool("Channel_1", "Sensor/Status/IsActive", valueBOOL)

```

用传感器数据描述的 XML 文件（已接收的数据）：

```

<Sensor>
  <Message>Example message</Message>
  <Status>
    <IsActive>1</IsActive>
  </Status>
</Sensor>

```

读取固定长度（10 字节）的二进制数据组：

- 配置的原始数据：

```

<RECEIVE>
  <RAW>
    <ELEMENT Tag="Buffer" Type="BYTE" Size="10" />
  </RAW>
</RECEIVE>

```

- 编程：

```

; Declaration
INT i
INT OFFSET
DECL EKI_STATUS RET
CHAR Bytes[10]
INT valueInt
REAL valueReal
BOOL valueBool
CHAR valueChar[1]

; Initialization
FOR i=(1) TO (10)
  Bytes[i]=0
ENDFOR

OFFSET=0
valueInt=0

```

```

valueBool=FALSE
valueReal=0
valueChar[1]=0

RET=EKI_GetString("Channel_1", "Buffer", Bytes[])

OFFSET=0
CAST_FROM(Bytes[], OFFSET, valueReal, valueInt, valueChar[], valueBool)

```

读取带末尾字符串的二进制数据组：

- 配置的原始数据：

```

<RECEIVE>
  <RAW>
    <ELEMENT Tag="Buffer" Type="STREAM" EOS="13, 10" />
  </RAW>
</RECEIVE>

```

- 编程：

```

; Declaration
INT i
DECL EKI_STATUS RET
CHAR Bytes[64]

; Initialization
FOR i=(1) TO (64)
  Bytes[i]=0
ENDFOR

RET=EKI_GetString("Channel_1", "Buffer", Bytes[])

```

## 6.2.6 删除数据存储器

### 说明

在删除数据存储器时，需区别下列几种情况：

- 通过 EKI\_Clear() 删除：删除所有数据存储器，退出以太网连接
- 通过 EKI\_ClearBuffer() 删除：删除特定数据存储器，不退出以太网连接

### EKI\_ClearBuffer()

哪些数据存储器可用指令 EKI\_ClearBuffer() 删除，取决于是否通过原始数据，还是通过 XML 结构进行通讯。

通过原始数据通讯时：

- 可删除用于已接收的数据的数据存储器。

通过 XML 结构通讯时：

- 通过给出配置的 XML 结构中一个元素的 XPath 表达式，可删除其数据存储器。这既适用于已接收的数据，也适用于待发送的数据。



XML 数据被 EKI 提取出来并根据类型储存在多个存储器内。删除各个存储器时必须确保不丢失关联数据。

### 示例

通过原始数据通讯：



删除用于已接收的数据的数据存储器：

- 配置的原始数据：

```
<RECEIVE>
<RAW>
  <ELEMENT Tag="RawData" Type="BYTE" Size="1408"
    Set_Flag="14"/>
</RAW>
</RECEIVE>
```

- 编程：

```
DECL EKI_STATUS RET
RET=EKI_ClearBuffer("Channel_1","RawData")
```

通过 XML 结构通讯：

删除用于标签 <Flag> 的数据存储器：

- 为接收数据所配置的 XML 结构：

```
<RECEIVE>
<XML>
  <ELEMENT Tag="Ext/Activ/Value" Type="REAL"/>
  <ELEMENT Tag="Ext/Activ/Flag" Type="BOOL"/>
  <ELEMENT Tag="Ext/Activ/Flag/Message" Type="STRING"/>
</XML>
</RECEIVE>
```

用于标签 <Message> 的数据存储器也同样被删除，因为该标签从属于标签 <Flag>。

- 编程：

```
DECL EKI_STATUS RET
RET=EKI_ClearBuffer("Channel_1","Ext/Activ/Flag")
```

6.2.7 EKI\_STATUS：函数特定返回值的结构

说明

EKI 的每项函数都会返回函数特定的数值。EKI\_STATUS 是这些数值写入的全局结构变量。

句法

GLOBAL STRUC EKI\_STATUS INT Buff, Read, Msg\_No, BOOL Connected, INT Counter

句法说明

元素	说明
Buff	存取后存储器中仍存在的元素数量
Read	从存储器中读取的元素数量
Msg_No	调用函数或接收数据时出现的错误的错误编号 如果自动信息输出被禁用，可用 EKI_CHECK() 读取错误编号，并在 smartHMI 上输出错误信息。

元素	说明
Connected	说明是否存在连接 <ul style="list-style-type: none"><li>• TRUE = 存在连接</li><li>• FALSE = 连接已中断</li></ul>
计数器	已接收的数据包的时间戳 给到达存储器的数据包进行连续编号，即按照其在存储器中储存的顺序。 读取单个数据时，会给结构元素 <i>Counter</i> 分配该数据元素来自的数据包的时间戳。 (>>> 6.2.10 “处理不完整数据组” 页面 43)

返回值

根据函数不同，将描述 EKI\_STATUS 结构的下列元素：

功能	Buff	Read	Msg_No	Connected	计数器
EKI_Init()	✗	✗	✓	✗	✗
EKI_Open()	✗	✗	✓	✓	✗
EKI_Close()	✗	✗	✓	✗	✗
EKI_Clear()	✗	✗	✓	✗	✗
EKI_Send()	✗	✗	✓	✓	✗
EKI_Set…()	✗	✗	✓	✓	✗
EKI_Get…()	✓	✓	✓	✓	✓
EKI_ClearBuffer()	✗	✗	✓	✓	✗
EKI_Lock()	✗	✗	✓	✓	✗
EKI_UnLock()	✗	✗	✓	✓	✗
EKI_CheckBuffer()	✓	✗	✓	✓	✓

6.2.8 配置事件信息

通过给输出端或旗标赋值，可报告下列事件：

- 连接已激活。
- 出现一个单独的 XML 元素。
- 出现一个完整的 XML 结构或一个完整的二进制数据组。

事件输出端

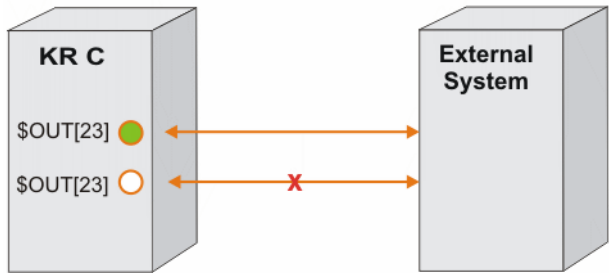


图 6-4：事件输出端（连接已激活）

只要与外部系统的连接处于激活状态，\$OUT[23] 即被赋值。连接不再激活时，将重置 \$OUT[23]。



只能用函数 EKI\_OPEN() 恢复连接。

事件旗标

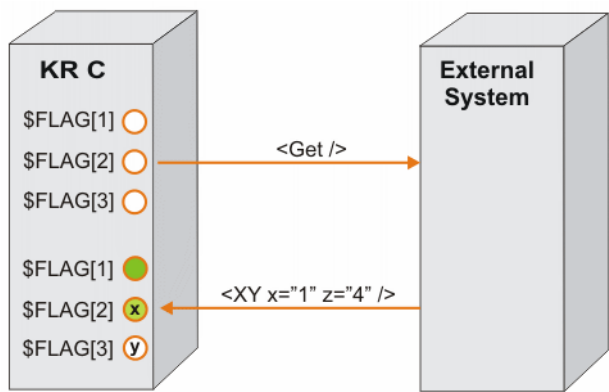


图 6-5：事件旗标（完整的 XML 结构）

XML 结构 <XY /> 包含数据元素 “XY/@x” 和 “XY/@z”。由于出现完整的 XML 结构，因此 \$FLAG[1] 将赋值。由于 “XY” 中包含元素 “x”，因此 \$FLAG[2] 将赋值。由于未传输元素 “y”，因此 \$FLAG[3] 不赋值。

示例

(>>> 7.6 “XmlCallback”示例” 页面 52)

6.2.9 接收完整的 XML 数据组

说明

存取函数 EKI\_Get... () 会一直保持禁用，直至一个 XML 数据组中的所有数据都位于存储器中为止。

当配置为 LIFO，且有 2 个或多个 XML 数据组相继出现时，不再确保从存储器中连续地取出一个数据组。例如可能出现这样的情况：尽管第一个数据组尚未完全处理好，但是第二个数据组的数据已储存在存储器中。由于在 ILFO 运行模式下，始终是先存取最后一次保存的数据，因此导致 KRL 中可用的数据组不一致。

为防止在 ILFO 运行模式下数据组出现碎片化现象，必须禁止处理新接收的数据，直至从存储器中取出所有关联的数据为止。

示例

```
...
RET=EKI_Lock("MyChannel")
RET=EKI_Get... ()
RET=EKI_Get... ()
...
RET=EKI_Get... ()
RET=EKI_Unlock("MyChannel")
...
```

6.2.10 处理不完整数据组

可能会出现外部系统发送不完整数据组的情况。单个 XML 元素为空或完全缺失，使得在一个班次的存储器中存在来自多个数据包的数据。

如果 KRL 中的数据组必须是连续的，则可使用变量 EKI\_STATUS 的结构元素 Counter。使用 EKI\_Get...Array 函数时，可从返回 Counter = 0 识别出时间上不连续的数据。

### 6.2.11 EKI\_CHECK(): 检查函数是否出现错误

#### 说明

KUKA.Ethernet KRL 3.2 会在出现任何错误时在 smarthMI 上输出一条信息。可禁用信息的自动输出。

(>>> [10.3 “禁用信息输出和信息记录” 页面 104](#))

如果自动信息输出已被禁用，建议用 EKI\_CHECK() 检查在运行函数时是否出错：

- 将读取错误编号并在 smarthMI 上输出相应的信息。
- 如果在 EKI\_CHECK() 中给出了通道名称，则在接收数据时会询问是否存在错误。

(>>> [10.4.5 “检查函数是否出现错误” 页面 111](#))

每次调用 EKI\_CHECK() 时将调用程序 KRC:\R1\TP\EthernetKRL\EthernetKRL\_USER.SRC。在该程序中可编程设定用户特定的错误反应。

#### 示例

接收时出错时将始终关闭连接。以太网连接中止时，作为错误策略，可编程设定一个中断。

- 在配置文件 XmlTransmit.XML 中定义了成功连接时 FLAG[1] 将赋值。连接丢失时将重置 FLAG[1]。

```
<ALIVE Set_Flag="1"/>
```

- 中断在 KRL 程序中声明并接通。重置 FLAG[1] 时，将执行中断程序。

```
;FOLD Define callback
  INTERRUPT DECL 89 WHEN $FLAG[1]==FALSE DO CON_ERR()
  INTERRUPT ON 89
;ENDFOLD
```

- 在中断程序中，用 EKI\_CHECK() 查询发生了什么错误，然后重新打开连接。

```
DEF CON_ERR()
  DECL EKI_STATUS RET
  RET={Buff 0,Read 0, Msg_no 0, Connected false}
  EKI_CHECK(RET,#Quit,"XmlTransmit")
  RET=EKI_OPEN("XmlTransmit")
END
```

## 7 示例配置和示例程序

### 7.1 接入服务器程序和示例

#### 说明

软件的供货范围包括一个服务器程序及多个示例配置和示例程序。利用这些示例配置和示例程序，可以在服务器程序与机器人控制系统之间建立通讯。

组件	目录
服务器程序 <ul style="list-style-type: none"><li>EthernetKRL_Server.exe</li></ul>	DOC\Example\Application
KRL 中的示例程序 <ul style="list-style-type: none"><li>BinaryFixed.src</li><li>BinaryStream.src</li><li>XmlCallback.src</li><li>XmlServer.src</li><li>XmlTransmit.src</li></ul>	DOC\Example\Program
XML 中的示例配置 <ul style="list-style-type: none"><li>BinaryFixed.xml</li><li>BinaryStream.xml</li><li>XmlCallBack.xml</li><li>XmlServer.xml</li><li>XmlTransmit.xml</li><li>XmlFullConfig.xml</li></ul>	DOC\Example\Config

#### 前提条件

外部系统:

- 已安装 .NET-Framework 3.5 或更高版本的 Windows 操作系统

机器人控制系统:

- 专家用户组
- 运行方式 T 1 或 T 2

#### 操作步骤

- 将服务器程序复制到外部系统上。
- 将 XML 示例配置复制到机器人控制系统的目录 C:\KRC\ROBOTER\Config\User\Common\EthernetKRL 中。
- 将 KRL 示例程序复制到机器人控制系统的目录 C:\KRC\ROBOTER\KRC\R1\Program 中。
- 启动外部系统上的服务器程序。  
(>>> 7.1.1 “服务器程序操作界面” 页面 46)
- 在服务器程序中选择菜单按钮。窗口**通讯属性**打开。
- 根据需要设置通讯参数。  
(>>> 7.1.2 “服务器程序中的通信参数” 页面 47)
- 关闭**通讯属性**窗口并且在服务器程序中选择启动按钮。用于通讯的端口会显示在信息窗口中。
- 在所需的 xml 文件中设置外部系统的 IP 地址。

7.1.1 服务器程序操作界面

说明

服务器程序可通过与机器人控制系统建立稳定连接，测试外部系统和机器人控制系统之间的通讯。

服务器程序包含以下功能：

- 发送和接收数据（自动或手动）
- 显示接收的数据
- 显示发送的数据

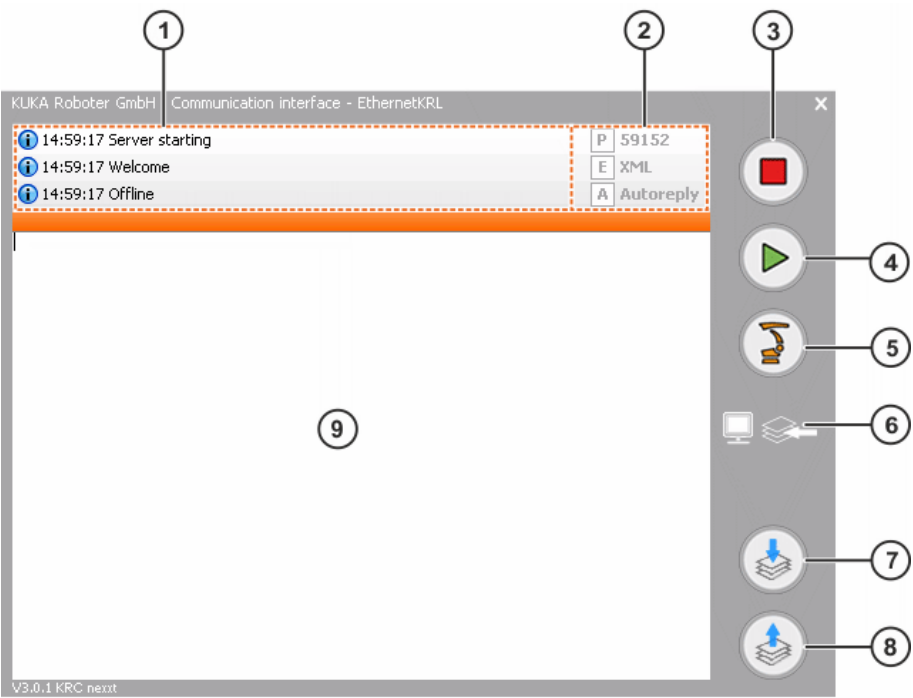


图 7-1：服务器程序操作界面

序号	说明
1	信息窗口
2	<div>显示所设置的通讯伙伴</div> <div>(&gt;&gt;&gt; 7.1.2 “服务器程序中的通信参数” 页面 47)</div> <div><ul style="list-style-type: none"><li>• P: 端口编号</li><li>• E: 示例数据<ul style="list-style-type: none"><li>– Xml: XML 数据</li><li>– BinaryFixed: 有固定长度的二进制数据</li><li>– BinaryStream: 带末尾字符串的可变二进制数据流</li></ul></li><li>• A: 通讯模式<ul style="list-style-type: none"><li>– Autoreply: 服务器自动回答每个已接收的数据包。</li><li>– Manual: 仅手动接收数据或发送数据</li></ul></li></ul></div>
3	<div>停止按钮</div> <div>结束与机器人控制系统的通讯并且重置服务器。</div>
4	<div>开始按钮</div> <div>分析服务器程序和机器人控制系统之间的数据交换。连接第一个到达的连接请求并用作通讯适配器。</div>

序号	说明
5	用于设置通讯参数的菜单按钮 (>>> 7.1.2 “服务器程序中的通信参数” 页面 47)
6	显示选项 <ul style="list-style-type: none"><li>箭头向左：显示接收的数据。（默认）</li><li>箭头向右：显示发送的数据。</li></ul>
7	手动接收数据的按钮
8	手动发送数据的按钮
9	显示窗口 根据所设置显示选项的不同，显示发送或接收的数据。

7.1.2 服务器程序中的通信参数

说明

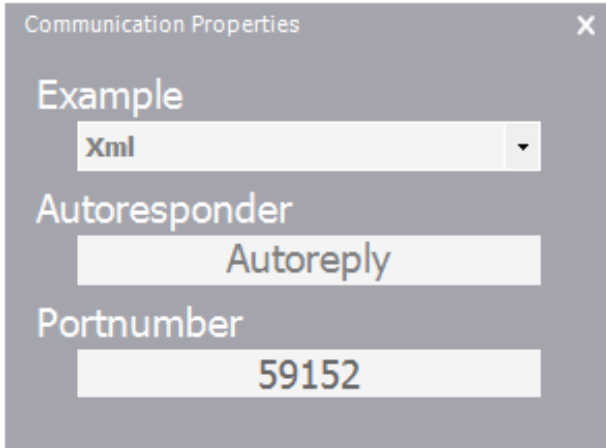


图 7-2：窗口通讯属性

参数	说明
Example	选择示例数据。 <ul style="list-style-type: none"><li>Xml: XML 数据</li><li>BinaryFixed: 有固定长度的二进制数据</li><li>BinaryStream: 带末尾字符串的可变二进制数据流</li></ul> 默认: Xml
Autoresponder	选择通讯模式。 <ul style="list-style-type: none"><li>Autoreply: 服务器自动回答每个已接收的数据包。</li><li>Manual: 仅手动接收数据或发送数据</li></ul> 默认: Autoreply
Portnumber	输入插座连接的端口编号。 外部系统在该端口上等待机器人控制系统的连接请求。 必须选择一个没有作为标准服务占用的空号。 默认: 59152 <b>提示:</b> 选择端口时, 须注意不能有其他服务 (如操作系统) 使用该端口。否则无法通过该端口建立连接。

## 7.2 “BinaryFixed”示例



与机器人控制系统通讯时，必须在服务器程序中设置合适的示例数据，例如这里是“BinaryFixed”。

EKI 配置为客户端。通过连接仅能接收固定长度为 10 字节、元素名称为“Buffer”的二进制数据组。服务器程序发送一个数据组。EKI 接收到外部数据后，\$FLAG[1] 将被赋值。

### XML 文件

```
<ETHERNETKRL>
  <CONFIGURATION>
    <EXTERNAL>
      <IP>x. x. x. x</IP>
      <PORT>59152</PORT>
    </EXTERNAL>
  </CONFIGURATION>
  <RECEIVE>
    <RAW>
      <ELEMENT Tag="Buffer" Type="BYTE" Set_Flag="1"
        Size="10" />
    </RAW>
  </RECEIVE>
  <SEND/>
</ETHERNETKRL>
```

必须在 KRL 程序中用 CAST\_TO() 和 CAST\_FROM() 读入和读取固定长度的二进制数据组。只可读取类型为 REAL (4 字节) 的数据，非 Double (双精度) 型。



关于指令 CAST\_TO() 和 CAST\_FROM() 的详细信息请参见文献 CREAD/CWRITE。

### 程序

```
1 DEF BinaryFixed( )
2 Declaration
3 INI
4 Initialize sample data
5
6 RET=EKI_Init("BinaryFixed")
7 RET=EKI_Open("BinaryFixed")
8
9 OFFSET=0
10 CAST_TO(Bytes[], OFFSET, 34.425, 674345, "R", TRUE)
11
12 RET=EKI_Send("BinaryFixed", Bytes[])
13
14 WAIT FOR $FLAG[1]
15 RET=EKI_GetString("BinaryFixed", "Buffer", Bytes[])
16 $FLAG[1]=FALSE
17
18 OFFSET=0
19 CAST_FROM(Bytes[], OFFSET, valueReal, valueInt, valueChar[], valueBool)
20
21 RET=EKI_Close("BinaryFixed")
22 RET=EKI_Clear("BinaryFixed")
23 END
```



行	说明
4	通过数值赋值将 KRL 变量初始化
6	EKI_Init() 将 EKI 与外部系统连接的通道初始化。
7	EKI_Open() 打开通道并与服务器连接。
9、10	CAST_TO 将数值写入 CHAR 数组 Bytes[]。
12	EKI_Send() 将 CHAR 数组 Bytes[] 发送给外部系统。
14 ... 16	\$FLAG[1] 表示接收配置的数据元素。 EKI_GetString 访问存储器并将数据复制到 CHAR 数组 Bytes[] 中。 \$FLAG[1] 被重置。
18、19	CAST_FROM 读取 CHAR 数组 Bytes[] 包含的数值，并根据类型将其复制到给出的变量中。
21	EKI_Close() 关闭通道。
22	EKI_Clear() 删除通道。

7.3 “BinaryStream”示例



与机器人控制系统通讯时，必须在服务器程序中设置合适的示例数据，例如这里是“BinaryStream”。

EKI 配置为客户端。通过连接仅能接收长度最大为 64 字节、元素名称为“Buffer”的二进制数据组。二进制数据组的末尾必须用末尾字符串 CR、LF 标记。当 EKI 接收到该元素时，\$FLAG[1] 将赋值。

XML 文件

```
<ETHERNETKRL>
  <CONFIGURATION>
    <EXTERNAL>
      <IP>x. x. x. x</IP>
      <PORT>59152</PORT>
    </EXTERNAL>
  </CONFIGURATION>
  <RECEIVE>
    <RAW>
      <ELEMENT Tag="Buffer" Type="STREAM" Set_Flag="1"
        Size="64" EOS="13,10" />
    </RAW>
  </RECEIVE>
  <SEND/>
</ETHERNETKRL>
```

程序

```
1 DEF BinaryStream( )
2 Declaration
3 INI
4 Initialize sample data
5
6 RET=EKI_Init("BinaryStream")
7 RET=EKI_Open("BinaryStream")
8
9 Bytes[]="Stream ends with CR,LF"
10
```

```
11 RET=EKI_Send("BinaryStream",Bytes[])
12
13 WAIT FOR $FLAG[1]
14 RET=EKI_GetString("BinaryStream","Buffer",Bytes[])
15 $FLAG[1]=FALSE
16
17 RET=EKI_Close("BinaryStream")
18 RET=EKI_Clear("BinaryStream")
19
20 END
```

行	说明
4	通过数值赋值将 KRL 变量初始化
6	EKI_Init() 将 EKI 与外部系统连接的通道初始化。
7	EKI_Open() 打开通道并与服务器连接。
9	用数据描述 CHAR 数组 Bytes[]。
11	EKI_Send() 将 CHAR 数组 Bytes[] 发送给外部系统。
13 ... 15	\$FLAG[1] 表示接收配置的数据元素。 EKI_GetString 从存储器读取 CHAR 数组 Bytes[] 的字符串。 \$FLAG[1] 被重置。
17	EKI_Close() 关闭通道。
18	EKI_Clear() 删除通道。

7.4 “XmlTransmit”示例



与机器人控制系统通讯时，必须在服务器程序中设置合适的示例数据，例如这里是“Xml”。

EKI 配置为客户端。将发送机器人数据，等待 1 秒后，从存储器读取接收的传感器数据。

XML 文件

```
<ETHERNETKRL>
  <CONFIGURATION>
    <EXTERNAL>
      <IP>x. x. x. x</IP>
      <PORT>59152</PORT>
    </EXTERNAL>
  </CONFIGURATION>
  <RECEIVE>
    <XML>
      <ELEMENT Tag="Sensor/Message" Type="STRING" />
      <ELEMENT Tag="Sensor/Positions/Current/@X" Type="REAL" />
      <ELEMENT Tag="Sensor/Positions/Before/X" Type="REAL" />
      <ELEMENT Tag="Sensor/Nmb" Type="INT" />
      <ELEMENT Tag="Sensor/Status/IsActive" Type="BOOL" />
      <ELEMENT Tag="Sensor/Read/xyzabc" Type="FRAME" />
      <ELEMENT Tag="Sensor/Show/@error" Type="BOOL" />
      <ELEMENT Tag="Sensor/Show/@temp" Type="INT" />
      <ELEMENT Tag="Sensor/Show" Type="STRING" />
      <ELEMENT Tag="Sensor/Free" Type="INT" />
    </XML>
  </RECEIVE>
</ETHERNETKRL>
```

```
</RECEIVE>
<SEND>
  <XML>
    <ELEMENT Tag="Robot/Data/LastPos/@X" />
    <ELEMENT Tag="Robot/Data/LastPos/@Y" />
    <ELEMENT Tag="Robot/Data/LastPos/@Z" />
    <ELEMENT Tag="Robot/Data/LastPos/@A" />
    <ELEMENT Tag="Robot/Data/LastPos/@B" />
    <ELEMENT Tag="Robot/Data/LastPos/@C" />
    <ELEMENT Tag="Robot/Data/ActPos/@X" />
    <ELEMENT Tag="Robot/Status" />
    <ELEMENT Tag="Robot/Mode" />
    <ELEMENT Tag="Robot/RobotLamp/GrenLamp/LightOn" />
  </XML>
</SEND>
</ETHERNETKRL>
```

程序

```
1 DEF XmlTransmit( )
2 Declaration
3 Communicated data
4 INI
5 Initialize sample data
6
7 RET=EKI_Init("XmlTransmit")
8 RET=EKI_Open("XmlTransmit")
9
10 Write data to connection
11 Send data to external program
12 Get received sensor data
13
14 RET=EKI_Close("XmlTransmit")
15 RET=EKI_Clear("XmlTransmit")
16
17 END
```

行	说明
5	通过数值赋值将 KRL 变量初始化
7	EKI_Init() 将 EKI 与外部系统连接的通道初始化。
8	EKI_Open() 打开通道并与外部系统连接。
10	将数据写入保存的用于发送数据的 XML 文件。
11	将描述的 XML 文件发送给外部系统。
12	从存储器读取已接收的传感器数据。
14	EKI_Close() 关闭通道。
15	EKI_Clear() 删除通道。

7.5 “XmlServer”示例

EKI 配置为服务器。只要与外部系统存在连接，\$FLAG[1] 就处于赋值状态。



将 EKI 配置为服务器时，无法在外部系统上使用服务器程序。可通过 Windows HyperTerminal 实现简单的客户端。

XML 文件

```
<ETHERNETKRL>
  <CONFIGURATION>
    <EXTERNAL>
      <TYPE>Client</TYPE>
    </EXTERNAL>
    <INTERNAL>
      <IP>x. x. x. x</IP>
      <PORT>54600</PORT>
      <ALIVE Set_Flag="1" />
    </INTERNAL>
  </CONFIGURATION>
  <RECEIVE>
    <XML>
      <ELEMENT Tag="Sensor/A" Type="BOOL" />
    </XML>
  </RECEIVE>
  <SEND>
    <XML>
      <ELEMENT Tag="Robot/B" />
    </XML>
  </SEND>
</ETHERNETKRL>
```

程序

```
1 DEF XmlServer( )
2 Declaration
3 INI
4
5 RET=EKI_Init("XmlServer")
6 RET=EKI_Open("XmlServer")
7
8 ; wait until server is connctected
9 wait for $FLAG[1]
10 ; wait until server is deconnected
11 wait for $FLAG[1]==FALSE
12
13 RET=EKI_Clear("XmlServer")
14 END
```

行	说明
5	EKI_Init() 将外部系统与 EKI 连接的通道初始化。
6	EKI_Open() 打开通道。
9	当外部客户端与服务器成功连接时，\$FLAG[1] 将赋值。
11	由于将 EKI 配置为服务器，所以机器人控制系统期待通道由外部客户端关闭。如果是这种情况，则删除 \$FLAG[1]。
13	EKI_Clear() 删除通道。

7.6 “XmlCallback”示例



与机器人控制系统通讯时，必须在服务器程序中设置合适的示例数据，例如这里是“Xml”。

EKI 配置为客户端。将发送机器人数据，接收传感器数据，然后等待 \$FLAG[1]。\$FLAG[1] 表示已读取传感器数据。

XML 文件中配置了当 EKI 接收了所有传感器数据后，\$FLAG[998] 将赋值。该旗标将触发程序中断。通过将标签“Sensor”配置为事件标签，可确保当所有数据都位于存储器中时，才取出传感器数据。

已读取传感器数据时，将重置 \$FLAG[998] 并给 \$FLAG[1] 赋值。

## XML 文件

```
<ETHERNETKRL>
  <CONFIGURATION>
    <EXTERNAL>
      <IP>x. x. x. x</IP>
      <PORT>59152</PORT>
    </EXTERNAL>
  </CONFIGURATION>
  <RECEIVE>
    <XML>
      <ELEMENT Tag="Sensor/Message" Type="STRING" />
      <ELEMENT Tag="Sensor/Positions/Current/@X" Type="REAL" />
      <ELEMENT Tag="Sensor/Positions/Before/X" Type="REAL" />
      <ELEMENT Tag="Sensor/Nmb" Type="INT" />
      <ELEMENT Tag="Sensor/Status/IsActive" Type="BOOL" />
      <ELEMENT Tag="Sensor/Read/xyzabc" Type="FRAME" />
      <ELEMENT Tag="Sensor/Show/@error" Type="BOOL" />
      <ELEMENT Tag="Sensor/Show/@temp" Type="INT" />
      <ELEMENT Tag="Sensor/Show" Type="STRING" />
      <ELEMENT Tag="Sensor/Free" Type="INT" Set_Out="998" />
      <ELEMENT Tag="Sensor" Set_Flag="998" />
    </XML>
  </RECEIVE>
  <SEND>
    <XML>
      <ELEMENT Tag="Robot/Data/LastPos/@X" />
      <ELEMENT Tag="Robot/Data/LastPos/@Y" />
      <ELEMENT Tag="Robot/Data/LastPos/@Z" />
      <ELEMENT Tag="Robot/Data/LastPos/@A" />
      <ELEMENT Tag="Robot/Data/LastPos/@B" />
      <ELEMENT Tag="Robot/Data/LastPos/@C" />
      <ELEMENT Tag="Robot/Data/ActPos/@X" />
      <ELEMENT Tag="Robot/Status" />
      <ELEMENT Tag="Robot/Mode" />
      <ELEMENT Tag="Robot/RobotLamp/GrenLamp/LightOn" />
    </XML>
  </SEND>
</ETHERNETKRL>
```

## 程序

```
1 DEF XmlCallBack( )
2 Declaration
3 Communicated data
4 INI
5 Define callback
6
7 RET=EKI_Init("XmlCallBack")
8 RET=EKI_Open("XmlCallBack")
9
10 Write data to connection
```

```
11 RET=EKI_Send("XmlCallBack","Robot")
12
13 ;wait until data read
14 WAIT FOR $FLAG[1]
15
16 RET=EKI_Close("XmlCallBack")
17 RET=EKI_Clear("XmlCallBack")
18 END
19
20 DEF GET_DATA()
21 Declaration
22 Initialize sample data
23 Get received sensor data
24 Signal read
```

行	说明
5	声明并接通中断
7	EKI_Init() 将 EKI 与外部系统连接的通道初始化。
8	EKI_Open() 打开通道。
10	将数据写入保存的用于发送数据的 XML 文件。
11	发送数据。
14	等待 \$FLAG[1]。 事件旗标报告已读取所有数据。
16	EKI_Close() 关闭通道。
17	EKI_Clear() 删除通道。
20 ... 24	通过数值赋值和读取数据将 KRL 变量初始化 已读取所有数据时，\$FLAG[1] 将赋值。

数据发送

XML 文件由 KRL 程序用机器人数据来描述并通过 EKI 发送给外部系统。

```
<Robot>
  <Data>
    <LastPos X="..." Y="..." Z="..." A="..." B="..." C="...">
    </LastPos>
    <ActPos X="1000.12">
    </ActPos>
  </Data>
  <Status>12345678</Status>
  <Mode>ConnectSensor</Mode>
  <RobotLamp>
    <GrenLamp>
      <LightOn>1</LightOn>
    </GrenLamp>
  </RobotLamp>
</Robot>
```

数据接收

XML 文件由服务器程序用传感器数据来描述并由 EKI 接收。

```
<Sensor>
  <Message>Example message</Message>
  <Positions>
    <Current X="4645.2" />
```

```
<Before>
  <X>0.9842</X>
</Before>
</Positions>
<Nmb>8</Nmb>
<Status>
  <IsActive>1</IsActive>
</Status>
<Read>
  <xyzabc X="210.3" Y="825.3" Z="234.3" A="84.2" B="12.3"
    C="43.5" />
</Read>
<Show error="0" temp="9929">Taginfo in attributes</Show>
<Free>2912</Free>
</Sensor>
```





## 8 诊断

### 8.1 显示诊断数据

#### 前提条件

- 用户权限：功能组 **诊断功能**

#### 操作步骤

1. 在主菜单中选择**诊断** > **诊断显示器**。
2. 在**模块**栏中选择模块 **EKI (EthernetKRL)**。

#### 说明

名称	说明
总内存	总共可用的内存（字节）
占用的内存	用户内存（字节）
机器人程序连接	机器人解释器初始化的连接数量
提交程序连接	提交解释器初始化的连接数量
系统连接	系统初始化的连接数量
以太网连接	打开的连接数量
处理时间	处理已接收的数据所需的最长时间（每 5 秒刷新一次）
警告信息	警告信息数量
错误信息	错误信息数量
	禁用自动信息输出和信息记录后，仍会对警告和错误信息进行计数。



## 9 信息

### 9.1 错误记录（EKI 日志）


EKI 的所有错误信息记录在 C:\KRC\ROBOTER\LOG\EthernetKRL 下的 LOG 文件中。

### 9.2 有关信息提示的说明

如果在调用函数或接收数据时出错，EKI 会反馈错误编号。错误编号对应于 smartHMI 显示的一条信息文本。如果自动输出信息已被禁用，可通过 EKI\_CHECK() 继续在 smartHMI 上显示信息。  
“信息”一章中包含了所选的各种信息。但并不涉及在信息窗口中显示的所有信息。

### 9.3 模块中的系统信息：EthernetKRL（EKI）

#### 9.3.1 EKI00002

信息代码	EKI00002
信息文本	系统存储器已耗尽
信息类型	故障信息
	
作用	轨道紧急停止 被激活的指令（机器人运动，程序启动）的输入被禁。
可能的原因	<b>原因：</b> 预留内存不足（>>> 页面 59） <b>解决方案：</b> 增加内存（>>> 页面 60）  <b>原因：</b> 连接配置占用过多内存（>>> 页面 60） <b>解决方案：</b> 更改连接配置（>>> 页面 60）  <b>原因：</b> 激活了数据传输量大的多个连接（>>> 页面 61） <b>解决方案：</b> 调整编程，以减少占用内存（>>> 页面 61）

#### 原因：预留内存不足

#### 说明

为 Ethernet KRL 预留的内存不足。

#### 检测说明

- 检查是否可通过另一种配置或编程减少所需内存。  
如果不可行，允许在与 KUKA Deutschland GmbH 协商后增加内存。

解决方案：增加内存



仅允许在与 KUKA Deutschland GmbH 协商后增加内存。 (>>> 11 “KUKA 服务” 页面 115)

前提条件

- 用户权限：
  - 专家用户组



视配置和使用的系统软件而定，可能需要更高的用户权限。

- 运行模式 T1、T2 或 AUT

操作步骤

1. 在目录 C:\KRC\ROBOTER\Config\User\Common 中打开文件 EthernetKRL.XML。
2. 在段 <EthernetKRL> 的元素 <MemSize> 中输入所需的内存容量，单位：字节。

```
<EthernetKRL>
  <Interface>
    <MemSize>1048576</MemSize>
  ...
</EthernetKRL>
```

3. 关闭文件，对于是否要保存更改的安全询问，回答是。
4. 重新启动机器人控制系统，并采用设置冷启动和重新读入文件。

原因：连接配置占用过多内存

说明

配置的以太网连接占用过多内存。

配置文件

针对每个以太网连接，必须配置一个 xml 文件。xml 文件的名称同时也是 KRL 的访问键。

目录	C:\KRC\ROBOTER\Config\User\Common\EthernetKRL
文件	示例：... \EXT.xml —> EKI_INIT("EXT")

检测说明

- 检查 xml 文件，以确定在配置以太网连接时是否可减少占用内存。

解决方案：更改连接配置

说明

更改连接配置，以减少占用内存。

前提条件

- 专家用户组
- 运行方式 T1 、T2 或 AUT

操作步骤

- 1. 根据需要更改 xml 文件并保存更改。
- 2. 如果已离线更改 xml 文件，将其复制到规定的目录中，覆盖旧的 xml 文件。

原因：激活了数据传输量大的多个连接

说明

如果同时激活了数据传输量大的多个连接，可能会导致占用过多内存。

检测说明


- 接着检查编程，以确定是否可对其进行相应更改，以减少占用内存。

解决方案：调整编程，以减少占用内存

说明

必须调整编程，以减少占用内存。

9.3.2 EKI00003

信息代码	EKI00003
信息文本	访问文件失败
信息类型	故障信息
	
作用	轨道紧急停止
	被激活的指令（机器人运动，程序启动）的输入被禁。
可能的原因	<p><b>原因：</b>包含连接配置的 xml 文件不存在 (&gt;&gt;&gt; 页面 61)</p> <p><b>解决方案：</b>恢复包含连接配置的 xml 文件 (&gt;&gt;&gt; 页面 62)</p> <p><b>原因：</b>包含连接配置的 xml 文件不可读 (&gt;&gt;&gt; 页面 62)</p> <p><b>解决方案：</b>恢复包含连接配置的 xml 文件 (&gt;&gt;&gt; 页面 63)</p>

原因：包含连接配置的 xml 文件不存在

说明

由于没有以函数 EKI\_Init() 中给出的名称保存的 xml 文件，因此以太网连接未初始化。

配置文件

针对每个以太网连接，必须配置一个 xml 文件。xml 文件的名称同时也是 KRL 的访问键。

目录	C:\KRC\ROBOTER\Config\User\Common\EthernetKRL
文件	<b>示例：</b> ...\EXT.xml —> EKI_INIT("EXT")

因此可以检查 xml 文件是否存在：

前提

- 专家用户群
- 已选定或者已打开程序。

检测说明

1. 记下函数 EKI\_Init() 中使用的文件名。  
RET = EKI\_Init("文件名")
  - 文件名: 包含连接配置的 xml 文件名称
2. 打开存储有 xml 文件的目录, 检查是否存在有函数 EKI\_Init() 中所给出名称的 xml 文件。

解决方案: 恢复包含连接配置的 xml 文件

说明

必须恢复包含连接配置的 xml 文件并复制到目录 C:\KRC\ROBOTER\Config\User\Common\EthernetKRL 中。

前提条件

- 专家用户组
- 运行方式 T1 、T2 或 AUT

操作步骤

- 将包含连接配置的 xml 文件复制到规定目录中。

原因: 包含连接配置的 xml 文件不可读

说明

由于函数 EKI\_Init() 中给出的 xml 文件不可读, 因此以太网连接未初始化。xml 文件损坏, 无法打开。

配置文件

针对每个以太网连接, 必须配置一个 xml 文件。xml 文件的名称同时也是 KRL 的访问键。

目录	C:\KRC\ROBOTER\Config\User\Common\EthernetKRL
文件	示例: ...\EXT.xml —> EKI_INIT("EXT")

因此可以检查 xml 文件是否损坏:

前提

- 专家用户群
- 已选定或者已打开程序。

检测说明

1. 记下函数 EKI\_Init() 中使用的文件名。  
RET = EKI\_Init("文件名")
  - 文件名: 包含连接配置的 xml 文件名称
2. 打开存储有 xml 文件的目录, 检查是否可打开 xml 文件。

解决方案：恢复包含连接配置的 xml 文件

说明

必须恢复包含连接配置的 xml 文件并复制到目录 C:\KRC\ROBOTER\Config\User\Common\EthernetKRL 中。


前提条件

- 专家用户组
- 运行方式 T1 、T2 或 AUT

操作步骤

- 将包含连接配置的 xml 文件复制到规定目录中。

9.3.3 EKI00006

信息代码	EKI00006
信息文本	解释配置失败
信息类型	故障信息
	
作用	轨道紧急停止
	被激活的指令（机器人运动，程序启动）的输入被禁。
可能的原因	<b>原因：</b> 包含连接配置的 xml 文件中的架构错误（>>> 页面 63） <b>解决方案：</b> 排除 xml 文件中的错误（>>> 页面 63）

原因：包含连接配置的 xml 文件中的架构错误

说明

包含连接配置的 xml 文件可能由于架构错误而无法读取。  
KUKA.Ethernet KRL 使用 XPath 架构。必须严格遵守架构规定的句法。例如，不允许缺少句号或结构元素。  
xml 文件中元素和属性的写法（包括大写和/或小写）均有规定，必须严格遵守。



关于 XPath 架构的更多信息请参见 KUKA.Ethernet KRL 的文档。

配置文件

针对每个以太网连接，必须配置一个 xml 文件。xml 文件的名称同时也是 KRL 的访问键。

目录	C:\KRC\ROBOTER\Config\User\Common\EthernetKRL
文件	<b>示例：</b> ...\EXT.xml —> EKI_INIT("EXT")

解决方案：排除 xml 文件中的错误

说明

必须排除 xml 文件中的错误。


前提条件

- 专家用户组
- 运行方式 T1 、 T2 或 AUT

操作步骤

1. 根据需要更改 xml 文件并保存更改。
2. 如果已离线更改 xml 文件，将其复制到规定的目录中，覆盖旧的 xml 文件。

9.3.4 EKI00007

信息代码	EKI00007
信息文本	写入需要发送的数据失败
信息类型	故障信息
	
作用	轨道紧急停止  被激活的指令（机器人运动，程序启动）的输入被禁。
可能的原因	<b>原因：</b> 用于发送数据的 XML 结构构建错误（>>> 页面 64） <b>解决方案：</b> 根据待发送的 XML 文件建立 XML 结构（>>> 页面 64）

原因：用于发送数据的 XML 结构构建错误

说明

由于在构建用于发送数据的 XML 结构时采用的架构与待发送的 XML 文件不同，因此无法描述该结构。KUKA.Ethernet KRL 使用 XPath 架构。




关于 XPath 架构的更多信息请参见 KUKA.Ethernet KRL 的文档。

解决方案：根据待发送的 XML 文件建立 XML 结构

说明

必须根据待发送的 XML 文件，按照 XPath 架构建立 XML 结构。

9.3.5 EKI00009

信息代码	EKI00009
信息文本	连接不存在
信息类型	故障信息
	
作用	轨道紧急停止  被激活的指令（机器人运动，程序启动）的输入被禁。



可能的原因	<b>原因：</b> EKI_Init() 未编程或编程错误 (>>> 页面 65) <b>解决方案：</b> 给函数正确编程 (>>> 页面 65)
-------	--

原因：EKI\_Init() 未编程或编程错误

说明

由于函数 EKI\_Init() 未编程或编程错误，因此以太网连接未初始化。  
必须始终用函数 EKI\_Init() 创建连接并初始化。此时将读入函数中给出的包含连接配置的 xml 文件。

RET = EKI_Init(CHAR[])	
功能	初始化一个通道，用于进行以太网通讯 将执行下列操作： <ul style="list-style-type: none"><li>• 读入连接配置</li><li>• 创建数据存储器</li><li>• 准备以太网连接</li></ul>
参数	类型：CHAR 通道名称 (= 包含连接配置的 xml 文件名称)
RET	类型：EKI_STATUS 函数的返回值
示例	RET = EKI_Init("Channel_1")

配置文件

针对每个以太网连接，必须配置一个 xml 文件。xml 文件的名称同时也是 KRL 的访问键。

目录	C:\KRC\ROBOTER\Config\User\Common\EthernetKRL
文件	<b>示例：</b> ...\EXT.xml —> EKI_INIT("EXT")

因此可以检查函数是否正确编程：

前提

- 专家用户群
- 已选定或者已打开程序。

检测说明

1. 检查是否编程设定了下列行：  
RET = EKI\_Init("文件名")
  - 文件名: 包含连接配置的 xml 文件名称
2. 检查函数 EKI\_Init() 中给出的文件名是否与包含连接配置的 xml 文件名称一致。

解决方案：给函数正确编程

说明

函数必须正确编程。

前提条件

- 用户权限：
  - 专家用户组



视配置和使用的系统软件而定，可能需要更高的用户权限。

- 运行模式 T1、T2 或 AUT

操作步骤

1. 在导航器中选定程序并按**打开**。在编辑器中显示该程序。
2. 搜索和编辑程序中相应的位置。
3. 关闭文件，对于是否要保存更改的安全询问，回答**是**。

9.3.6 EKI00010

信息代码	EKI00010
信息文本	以太网断开
信息类型	故障信息
	
作用	轨道紧急停止  被激活的指令（机器人运动，程序启动）的输入被禁。
可能的原因	<b>原因：</b> EKI_Open() 未编程或编程错误 (>>> 页面 66) <b>解决方案：</b> 给函数正确编程 (>>> 页面 67)

**原因：** EKI\_Open() 未编程或编程错误

说明

由于函数 EKI\_Open() 未编程或编程错误，因此以太网连接已初始化但尚未打开。

RET = EKI_Open (CHAR[])	
功能	打开初始化的通道  EKI 配置为客户端时，EKI 将与外部系统 (= 服务器) 连接。  EKI 配置为服务器时，EKI 将等待外部系统 (= 客户端) 的连接询问。
参数	类型：CHAR  信道名称
RET	类型：EKI_STATUS  通道名称 (= 包含连接配置的 xml 文件名称)
示例	RET = EKI_Open ("Channel_1")

配置文件

针对每个以太网连接，必须配置一个 xml 文件。xml 文件的名称同时也是 KRL 的访问键。

目录	C:\KRC\ROBOTER\Config\User\Common\EthernetKRL
文件	示例: ...\EXT.xml —> EKI_INIT("EXT")

因此可以检查函数是否正确编程：

前提

- 专家用户群
- 已选定或者已打开程序。

检测说明

1. 检查是否编程设定了下列行：  
RET = EKI\_Open("文件名")
  - 文件名: 包含连接配置的 xml 文件名称
2. 检查函数 EKI\_Open() 中给出的文件名是否与函数 EKI\_Init() 中使用的文件名一致。

解决方案：给函数正确编程

说明

函数必须正确编程。

前提条件

- 用户权限：
  - 专家用户组




视配置和使用的系统软件而定，可能需要更高的用户权限。

- 运行模式 T1、T2 或 AUT

操作步骤

1. 在导航器中选定程序并按**打开**。在编辑器中显示该程序。
2. 搜索和编辑程序中相应的位置。
3. 关闭文件，对于是否要保存更改的安全询问，回答**是**。

9.3.7 EKI00011

信息代码	EKI00011
信息文本	已存在与外部系统的以太网连接
信息类型	故障信息 
作用	轨道紧急停止 被激活的指令（机器人运动，程序启动）的输入被禁。

可能的原因	<b>原因：</b> 已用 EKI_Open() 打开以太网连接 (>>> 页面 68) <b>解决方案：</b> 删除过度编程的函数 (>>> 页面 68)
-------	--

**原因：**已用 EKI\_Open() 打开以太网连接

**说明**

已用函数 EKI\_Open() 打开以太网连接。

RET = EKI_Open (CHAR[])	
功能	打开初始化的通道  EKI 配置为客户端时，EKI 将与外部系统 (= 服务器) 连接。  EKI 配置为服务器时，EKI 将等待外部系统 (= 客户端) 的连接问询。
参数	类型：CHAR  信道名称
RET	类型：EKI_STATUS  通道名称 (= 包含连接配置的 xml 文件名称)
示例	RET = EKI_Open("Channel_1")

**配置文件**

针对每个以太网连接，必须配置一个 xml 文件。xml 文件的名称同时也是 KRL 的访问键。

目录	C:\KRC\ROBOTER\Config\User\Common\EthernetKRL
文件	<b>示例：</b> ...\EXT.xml —> EKI_INIT("EXT")

**因此可以检查是否已打开连接：**

**前提**

- 专家用户群
- 已选定或者已打开程序。

**检测说明**

- 检查是否在连接的初始化和关闭之间多次编程设定下列行：  
RET = EKI\_Open("文件名")
  - 文件名：包含连接配置的 xml 文件名称

**解决方案：**删除过度编程的函数

**说明**

必须删除过度编程的函数。

**前提条件**

- 用户权限：
  - 专家用户组




视配置和使用的系统软件而定，可能需要更高的用户权限。

- 运行模式 T1、T2 或 AUT

操作步骤

- 在导航器中选定程序并按**打开**。在编辑器中显示该程序。
- 搜索和编辑程序中相应的位置。
- 关闭文件，对于是否要保存更改的安全询问，回答**是**。

9.3.8 EKI00012

信息代码	EKI00012
信息文本	创建服务器失败
信息类型	故障信息
	
作用	轨道紧急停止
	被激活的指令（机器人运动，程序启动）的输入被禁。
可能的原因	<b>原因：</b> EKI 的 IP 地址和/或端口号未给出或给出错误（>>> 页面 69） <b>解决方案：</b> 在 xml 文件中输入正确的 IP 地址和端口号（>>> 页面 70）

原因：EKI 的 IP 地址和/或端口号未给出或给出错误

说明

EKI 配置为服务器，而外部系统则配置为客户端。在包含连接配置的 xml 文件中，EKI 的 IP 地址和/或端口号未给出或给出的形式错误。为了确保网络安全，可以规定能够从哪个子网访问 EKI。

示例

只可以通过子网 192.168.X.X 连接端口 54600：

```
<INTERNAL>
<IP>192.168.23.1</IP>
<PORT>54600</PORT>
</INTERNAL>
```

可以通过所有已在 KLI 中设置的子网来连接端口 54600：

```
<INTERNAL>
<IP>0.0.0.0</IP>
<PORT>54600</PORT>
</INTERNAL>
```

配置文件

针对每个以太网连接，必须配置一个 xml 文件。xml 文件的名称同时也是 KRL 的访问键。

目录	C:\KRC\ROBOTER\Config\User\Common\EthernetKRL
文件	示例：...\EXT.xml → EKI_INIT("EXT")

检测说明

- 在包含连接配置的 xml 文件中检查是否正确输入了 IP 地址和端口号。

解决方案：在 xml 文件中输入正确的 IP 地址和端口号


前提条件

- 专家用户组
- 运行方式 T1 、T2 或 AUT

操作步骤

- 根据需要更改 xml 文件并保存更改。
- 如果已离线更改 xml 文件，将其复制到规定的目录中，覆盖旧的 xml 文件。

9.3.9 EKI00013

信息代码	EKI00013
信息文本	未能初始化以太网参数
信息类型	故障信息
	
作用	轨道紧急停止  被激活的指令（机器人运动，程序启动）的输入被禁。
可能的原因	<b>原因：</b> 外部系统的 IP 地址和/或端口号未给出或给出错误（>>> 页面 70） <b>解决方案：</b> 在 xml 文件中输入正确的 IP 地址和端口号（>>> 页面 71）

原因：外部系统的 IP 地址和/或端口号未给出或给出错误

说明

EKI 配置为客户端，而外部系统则配置为服务器。在包含连接配置的 xml 文件中，外部系统的 IP 地址和/或端口号未给出或给出的形式错误。

必须在 xml 文件的段 <EXTERNAL> ... </EXTERNAL> 中按下列方式给出外部系统的 IP 地址和端口号：

- <IP>IP 地址</IP>
- <PORT>端口号</PORT>

配置文件

针对每个以太网连接，必须配置一个 xml 文件。xml 文件的名称同时也是 KRL 的访问键。

目录	C:\KRC\ROBOTER\Config\User\Common\EthernetKRL
文件	示例: ...\EXT.xml —> EKI_INIT("EXT")

检测说明

- 在包含连接配置的 xml 文件中检查是否正确输入了 IP 地址和端口号。

解决方案：在 xml 文件中输入正确的 IP 地址和端口号

前提条件

- 专家用户组
- 运行方式 T1 、T2 或 AUT

操作步骤

- 根据需要更改 xml 文件并保存更改。
- 如果已离线更改 xml 文件，将其复制到规定的目录中，覆盖旧的 xml 文件。

9.3.10 EKI00014

信息代码	EKI00014
信息文本	未能与外部系统建立以太网连接
信息类型	故障信息
	
作用	轨道紧急停止  被激活的指令（机器人运动，程序启动）的输入被禁。
可能的原因	<p><b>原因：</b>给出的 IP 地址和/或端口号错误 (&gt;&gt;&gt; 页面 72) <b>解决方案：</b>在 xml 文件中输入正确的 IP 地址和端口号 (&gt;&gt;&gt; 页面 72)</p> <p><b>原因：</b>Realtime OS 网络适配器和 KUKA.VisionTech 中的不同子网 (&gt;&gt;&gt; 页面 72) <b>解决方案：</b>在 VisionTechConfig.xml 中调整 IP 地址 (&gt;&gt;&gt; 页面 73)</p> <p><b>原因：</b>网线损坏或未正确连接 (&gt;&gt;&gt; 页面 73) <b>解决方案：</b>更换或正确插入网线 (&gt;&gt;&gt; 页面 73)</p> <p><b>原因：</b>因外部系统软件错误而无以太网连接 (&gt;&gt;&gt; 页面 73) <b>解决方案：</b>排除外部系统软件中的错误 (&gt;&gt;&gt; 页面 74)</p> <p><b>原因：</b>因外部系统硬件故障而无以太网连接 (&gt;&gt;&gt; 页面 74) <b>解决方案：</b>排除外部系统硬件故障 (&gt;&gt;&gt; 页面 74)</p>

原因：给出的 IP 地址和/或端口号错误

说明

包含连接配置的 xml 文件中给出的 IP 地址和/或端口号错误。它们与外部系统或 EKI 不符。

IP 地址和端口号应按下列方式给出，取决于外部系统是配置为服务器还是客户端：

- 段 <EXTERNAL> ... </EXTERNAL>
  - <TYPE>Server</TYPE> 或 <TYPE> 未给出：外部系统配置为服务器。
  - <TYPE>Client</TYPE>：外部系统配置为客户端。
  - 如果 TYPE = Server，必须在此输入外部系统的 IP 地址和端口号：
    - <IP>IP 地址</IP>
    - <PORT>端口号</PORT>
  - 如果 TYPE = Client，将忽略在此给出的连接参数。
- 段 <INTERNAL> ... </INTERNAL>
  - 如果 TYPE = Client，必须在此输入 EKI 的 IP 地址和端口号：
    - <IP>IP 地址</IP>
    - <PORT>端口号</PORT>
  - 如果 TYPE = Server，将忽略在此给出的连接参数。

配置文件

针对每个以太网连接，必须配置一个 xml 文件。xml 文件的名称同时也是 KRL 的访问键。

目录	C:\KRC\ROBOTER\Config\User\Common\EthernetKRL
文件	示例：... \EXT.xml —> EKI_INIT("EXT")

检测说明

- 在包含连接配置的 xml 文件中检查是否正确输入了 IP 地址和端口号。

解决方案：在 xml 文件中输入正确的 IP 地址和端口号

前提条件

- 专家用户组
- 运行方式 T1 、 T2 或 AUT

操作步骤

1. 根据需要更改 xml 文件并保存更改。
2. 如果已离线更改 xml 文件，将其复制到规定的目录中，覆盖旧的 xml 文件。

原因：Realtime OS 网络适配器和 KUKA.VisionTech 中的不同子网

说明

通过以太网连接，KUKA.VisionTech 可以在控制系统中的 VxWorks 与 Windows 之间进行内部通信。Realtime OS 网络适配器和 VisionTechConfig.xml 中的 IP 地址不在同一个子网中。



## 配置文件

目录	C:\KRC\Roboter\Config\User\Common\EthernetKRL
文件	VisionTechConfig.xml

因此可以检查 IP 地址是否在同一个子网中：

## 检测说明

1. 在主菜单中选择**投入运行** > **网络配置**。**网络配置** 窗口打开。
2. 点击**高级**。用于高级网络配置的窗口打开。
3. 选择选项卡**内部子网**。在**共享内存网络**区域显示编号范围。
4. 打开文件。
5. 查找 <External>。  
IP 地址位于 <IP> 下。
6. 检查 KUKA.HMI 上和文件中的编号范围是否一致。

**解决方案：**在 VisionTechConfig.xml 中调整 IP 地址

## 说明

必须在 VisionTechConfig.xml 中调整 IP 地址。

## 操作步骤

1. 打开配置文件。
2. 查找 <External>。  
在 <IP> 下输入 Realtime OS 网络适配器的 IP 地址。

**原因：**网线损坏或未正确连接

## 说明

网线损坏或插接错误。

如此可检查网线和连接插头是否正确连接：

## 检测说明

1. 检查网线是否插接正确以及是否牢固。
2. 相互调换网线。

**解决方案：**更换或正确插入网线

## 操作步骤

- 更换或正确插入网线。

**原因：**因外部系统软件错误而无以太网连接

## 说明

由于外部系统的软件存在错误，因此没有以太网连接。

## 检测说明

- 检查外部系统是否存在软件错误。

解决方案：排除外部系统软件中的错误

说明

必须排除外部系统软件中的错误。

原因：因外部系统硬件故障而无以太网连接

说明

由于外部系统存在硬件故障，例如插口接触不良、网卡损坏等，因此没有以太网连接。

检测说明


- 检查外部系统是否存在硬件故障。

解决方案：排除外部系统硬件故障

说明

必须排除外部系统硬件故障。

9.3.11 EKI00015

信息代码	EKI00015
信息文本	访问空的接收存储器
信息类型	故障信息
	
作用	轨道紧急停止 被激活的指令（机器人运动，程序启动）的输入被禁。
可能的原因	<b>原因：</b> 用 EKI_Get... () 访问时数据存储器为空 (>>> 页面 74) <b>解决方案：</b> 更改程序 (>>> 页面 75)

原因：用 EKI\_Get... () 访问时数据存储器为空

说明

用 EKI\_Get... () 函数访问了一个空的数据存储器。  
通过查询和分析存取函数的相应返回值，可防止访问空的存储器。  
EKI\_STATUS 是函数的返回值写入的全局结构变量。与分析相关的是 EKI\_STATUS 的 Buff 元素。  
Buff 包含下列数值：

- 存取后存储器中仍存在的元素数量

句法

GLOBAL STRUC EKI\_STATUS INT Buff, Read, Msg\_No, BOOL Connected,  
INT Counter

示例

```
...
REPEAT
    ret = EKI_GetInt("MyChannel", "Root/Number", value)
    DoSomething(value)
UNTIL (ret.Buff < 1)
...
```

解决方案：更改程序


前提条件

- 专家用户组
- 运行方式 T1 、T2 或 AUT

操作步骤

1. 在导航器中选定程序并按**打开**。在编辑器中显示该程序。
2. 搜索和编辑程序中相应的位置。
3. 关闭文件，对于是否要保存更改的安全询问，回答**是**。

9.3.12 EKI00016

信息代码	EKI00016
信息文本	未能找到元素
信息类型	故障信息
	
作用	轨道紧急停止
	被激活的指令（机器人运动，程序启动）的输入被禁。
可能的原因	<b>原因：</b> 用于接收数据的 XML 结构中的元素未配置或配置错误（>>> 页面 75） <b>解决方案：</b> 排除 xml 文件中的错误（>>> 页面 77）  <b>原因：</b> 存取函数中的元素名称编程错误（>>> 页面 77） <b>解决方案：</b> 给函数正确编程（>>> 页面 78）

原因：用于接收数据的 XML 结构中的元素未配置或配置错误

说明

存取函数中给出的元素在用于数据接收的 XML 结构中未配置或与配置的元素不一致。

存取函数

一个存取函数中的参数 2 分别给出要存取的元素。

存取函数
EKI_STATUS = EKI_GetBool (CHAR[], CHAR[], BOOL)
EKI_STATUS = EKI_GetBoolArray (CHAR[], CHAR[], BOOL[])
EKI_STATUS = EKI_GetInt (CHAR[], CHAR[], INT)
EKI_STATUS = EKI_GetIntArray (CHAR[], CHAR[], INT[])
EKI_STATUS = EKI_GetReal (CHAR[], CHAR[], REAL)
EKI_STATUS = EKI_GetRealArray (CHAR[], CHAR[], REAL[])
EKI_STATUS = EKI_GetString (CHAR[], CHAR[], CHAR[])
EKI_STATUS = EKI_GetFrame (CHAR[], CHAR[], FRAME)
EKI_STATUS = EKI_GetFrameArray (CHAR[], CHAR[], FRAME[])

配置文件

针对每个以太网连接，必须配置一个 xml 文件。xml 文件的名称同时也是 KRL 的访问键。

目录	C:\KRC\ROBOTER\Config\User\Common\EthernetKRL
文件	示例：…\EXT.xml → EKI_INIT("EXT")

在 xml 文件的段 <RECEIVE> … </RECEIVE> 中配置了接收结构。属性 Tag 定义了可存取的元素。

检测说明

- 1. 检查尝试访问的元素在接收结构中是否已配置。
- 2. 检查编程的元素名称是否与配置的元素名称一致。

示例

读取 XML 数据：

XML 结构：

```
<RECEIVE>
  <XML>
    <ELEMENT Tag="Sensor/Message" Type="STRING" />
    <ELEMENT Tag="Sensor/Status/IsActive" Type="BOOL" />
  </XML>
</RECEIVE>
```

编程：

```
RET=EKI_GetString("Channel_1", "Sensor/Message", valueChar[])
RET=EKI_GetBool("Channel_1", "Sensor/Status/IsActive", valueBOOL)
```

读取固定长度（10 字节）的二进制数据组：

- 原始数据：

```
<RECEIVE>
  <RAW>
    <ELEMENT Tag="Buffer" Type="BYTE" Size="10" />
  </RAW>
</RECEIVE>
```

- 编程：

```
RET=EKI_GetString("Channel_1", "Buffer", Bytes[])
```

解决方案：排除 xml 文件中的错误

说明

必须排除 xml 文件中的错误。

前提条件

- 专家用户组
- 运行方式 T1 、T2 或 AUT

操作步骤

1. 根据需要更改 xml 文件并保存更改。
2. 如果已离线更改 xml 文件，将其复制到规定的目录中，覆盖旧的 xml 文件。

原因：存取函数中的元素名称编程错误

说明

存取函数中给出的元素名称与用于接收数据的 XML 结构中配置的元素名称不一致。

存取函数

一个存取函数中的参数 2 分别给出要存取的元素。

存取函数	
EKI_STATUS	= EKI_GetBool (CHAR[], CHAR[], BOOL)
EKI_STATUS	= EKI_GetBoolArray (CHAR[], CHAR[], BOOL[])
EKI_STATUS	= EKI_GetInt (CHAR[], CHAR[], INT)
EKI_STATUS	= EKI_GetIntArray (CHAR[], CHAR[], INT[])
EKI_STATUS	= EKI_GetReal (CHAR[], CHAR[], REAL)
EKI_STATUS	= EKI_GetRealArray (CHAR[], CHAR[], REAL[])
EKI_STATUS	= EKI_GetString (CHAR[], CHAR[], CHAR[])
EKI_STATUS	= EKI_GetFrame (CHAR[], CHAR[], FRAME)
EKI_STATUS	= EKI_GetFrameArray (CHAR[], CHAR[], FRAME[])

配置文件

针对每个以太网连接，必须配置一个 xml 文件。xml 文件的名称同时也是 KRL 的访问键。

目录	C:\KRC\ROBOTER\Config\User\Common\EthernetKRL
文件	示例：... \EXT.xml —> EKI_INIT ("EXT")

在 xml 文件的段 <RECEIVE> ... </RECEIVE> 中配置了接收结构。属性 Tag 定义了可存取的元素。

检测说明

- 检查编程的元素名称是否与配置的元素名称一致。

示例

读取 XML 数据：  
XML 结构：

```
<RECEIVE>
  <XML>
    <ELEMENT Tag="Sensor/Message" Type="STRING" />
    <ELEMENT Tag="Sensor/Status/IsActive" Type="BOOL" />
  </XML>
</RECEIVE>
```

编程:

```
RET=EKI_GetString("Channel_1", "Sensor/Message", valueChar[])
RET=EKI_GetBool("Channel_1", "Sensor/Status/IsActive", valueBOOL)
```

读取固定长度（10 字节）的二进制数据组:

- 原始数据:

```
<RECEIVE>
  <RAW>
    <ELEMENT Tag="Buffer" Type="BYTE" Size="10" />
  </RAW>
</RECEIVE>
```

- 编程:

```
RET=EKI_GetString("Channel_1", "Buffer", Bytes[])
```

解决方案：给函数正确编程

说明

函数必须正确编程。

前提条件

- 用户权限:
  - 专家用户组




视配置和使用的系统软件而定，可能需要更高的用户权限。

- 运行模式 T1、T2 或 AUT

操作步骤

1. 在导航器中选定程序并按**打开**。在编辑器中显示该程序。
2. 搜索和编辑程序中相应的位置。
3. 关闭文件，对于是否要保存更改的安全询问，回答**是**。

9.3.13 EKI00017

信息代码	EKI00017
信息文本	汇集需要发送的数据失败
信息类型	故障信息
	
作用	轨道紧急停止
	被激活的指令（机器人运动，程序启动）的输入被禁。

可能的原因	原因：用于发送数据的 XML 结构配置错误 (>>> 页面 79)
	解决方案：排除 xml 文件中的错误 (>>> 页面 80)
	原因：EKI_Send() 编程错误 (>>> 页面 80)
	解决方案：给函数正确编程 (>>> 页面 81)

原因：用于发送数据的 XML 结构配置错误

说明

用于发送数据的 XML 结构与应用 XML 数据描述的 XML 文件不符。可能也与函数 EKI\_Send() 的编程不符。

RET = EKI_Send(CHAR[], CHAR[], INT)	
功能	通过通道发送数据 (>>> 6.2.4 “发送数据” 页面 36)
参数 1	类型：CHAR 应通过其发送的打开的通道名称
参数 2	类型：CHAR 定义待发送数据的范围。 通过 XML 结构通讯时： <ul style="list-style-type: none"><li>为发送数据所配置的 XML 结构中的待发送元素的 XPath 表达式。<ul style="list-style-type: none"><li>如果仅给出了根元素，将传输整个 XML 结构。参见示例 1。</li><li>如果只应传输 XML 结构的一部分，即某个下级层的元素，必须给出从根元素至该元素的路径。参见示例 2。</li></ul></li><li>可选：应传输的可变长度的任意字符串</li></ul> 通过原始数据通讯时： <ul style="list-style-type: none"><li>应传输的任意字符串：<ul style="list-style-type: none"><li>对于有固定长度的二进制数据组（属性 Type = “BYTE”）：有固定长度的任意字符串 用字节表示的字符串大小必须与配置的属性 Size（大小）完全相符。超出时将输出错误信息，低于时将输出警告信息。</li><li>对于有可变末尾字符串的二进制数据组（属性 Type = “STREAM”）：有可变长度的任意字符串 自动一并发送末尾字符串。</li></ul></li></ul> <b>提示：</b> 如果可变长度的任意字符串包含一个 ASCII 零字符，将仅传输到该字符的部分。如果此非所愿，则必须相应地定义参数 3。
参数 3（可选）	类型：INT 只有在发送长度可变的任意字符串时才相关（见参数 2）：最大待发送的字符数量
RET	类型：EKI_STATUS 函数的返回值
示例 1	RET = EKI_Send(“Channel_1”, “Root”)
示例 2	RET = EKI_Send(“Channel_1”, “Root/Test”)
示例 3	RET = EKI_Send(“Channel_1”, MyBytes[])
示例 4	RET = EKI_Send(“Channel_1”, MyBytes[], 6)

配置文件

针对每个以太网连接，必须配置一个 xml 文件。xml 文件的名称同时也是 KRL 的访问键。

目录	C:\KRC\ROBOTER\Config\User\Common\EthernetKRL
文件	示例: ...\EXT.xml —> EKI_INIT("EXT")

在 xml 文件的段 <SEND> ... </SEND> 中配置了发送结构。

检测说明

- 1. 检查 xml 文件中的发送结构配置。
- 2. 检查所发送数据的编程是否与配置相符。

解决方案：排除 xml 文件中的错误

说明

必须排除 xml 文件中的错误。

前提条件

- 专家用户组
- 运行方式 T1 、 T2 或 AUT

操作步骤

- 1. 根据需要更改 xml 文件并保存更改。
- 2. 如果已离线更改 xml 文件，将其复制到规定的目录中，覆盖旧的 xml 文件。

原因：EKI\_Send() 编程错误

说明

函数 EKI\_Send() 中给出的待发送的数据错误。可能与为发送数据所配置的 XML 结构不符。

RET = EKI_Send(CHAR[], CHAR[], INT)	
功能	通过通道发送数据 (>>> 6.2.4 “发送数据” 页面 36)
参数 1	类型: CHAR 应通过其发送的打开的通道名称



RET = EKI_Send(CHAR[], CHAR[], INT)	
参数 2	<p>类型：CHAR</p> <p>定义待发送数据的范围。</p> <p>通过 XML 结构通讯时：</p> <ul style="list-style-type: none"><li>为发送数据所配置的 XML 结构中的待发送元素的 XPath 表达式。<ul style="list-style-type: none"><li>如果仅给出了根元素，将传输整个 XML 结构。参见示例 1。</li><li>如果只应传输 XML 结构的一部分，即某个下级层的元素，必须给出从根元素至该元素的路径。参见示例 2。</li></ul></li><li>可选：应传输的可变长度的任意字符串</li></ul> <p>通过原始数据通讯时：</p> <ul style="list-style-type: none"><li>应传输的任意字符串：<ul style="list-style-type: none"><li>对于有固定长度的二进制数据组（属性 Type = "BYTE"）：有固定长度的任意字符串 用字节表示的字符串大小必须与配置的属性 Size（大小）完全相符。超出时将输出错误信息，低于时将输出警告信息。</li><li>对于有可变末尾字符串的二进制数据组（属性 Type = "STREAM"）：有可变长度的任意字符串 自动一并发送末尾字符串。</li></ul></li></ul> <p><b>提示：</b>如果可变长度的任意字符串包含一个 ASCII 零字符，将仅传输到该字符的部分。如果此非所愿，则必须相应地定义参数 3。</p>
参数 3（可选）	<p>类型：INT</p> <p>只有在发送长度可变的任意字符串时才相关（见参数 2）：最大待发送的字符数量</p>
RET	<p>类型：EKI_STATUS</p> <p>函数的返回值</p>
示例 1	RET = EKI_Send("Channel_1", "Root")
示例 2	RET = EKI_Send("Channel_1", "Root/Test")
示例 3	RET = EKI_Send("Channel_1", MyBytes[])
示例 4	RET = EKI_Send("Channel_1", MyBytes[], 6)

配置文件

针对每个以太网连接，必须配置一个 xml 文件。xml 文件的名称同时也是 KRL 的访问键。

目录	C:\KRC\ROBOTER\Config\User\Common\EthernetKRL
文件	<b>示例：</b> ...\EXT.xml —> EKI_INIT("EXT")

在 xml 文件的段 <SEND> ... </SEND> 中配置了发送结构。

检测说明

1. 检查 xml 文件中的发送结构配置。
2. 检查所发送数据的编程是否与配置相符。

解决方案：给函数正确编程

说明

函数必须正确编程。

前提条件

- 用户权限：
  - 专家用户组




视配置和使用的系统软件而定，可能需要更高的用户权限。

- 运行模式 T1、T2 或 AUT

操作步骤

1. 在导航器中选定程序并按**打开**。在编辑器中显示该程序。
2. 搜索和编辑程序中相应的位置。
3. 关闭文件，对于是否要保存更改的安全询问，回答**是**。

9.3.14 EKI00018

信息代码	EKI00018
信息文本	发送数据失败
信息类型	故障信息
	
作用	轨道紧急停止  被激活的指令（机器人运动，程序启动）的输入被禁。
可能的原因	<b>原因：</b> 网线损坏或未正确连接（>>> 页面 82） <b>解决方案：</b> 更换或正确插入网线（>>> 页面 82）  <b>原因：</b> 因外部系统软件错误而无以太网连接（>>> 页面 83） <b>解决方案：</b> 排除外部系统软件中的错误（>>> 页面 83）  <b>原因：</b> 因外部系统硬件故障而无以太网连接（>>> 页面 83） <b>解决方案：</b> 排除外部系统硬件故障（>>> 页面 83）

原因：网线损坏或未正确连接

说明

网线损坏或插接错误。  
如此可检查网线和连接插头是否正确连接：

检测说明

1. 检查网线是否插接正确以及是否牢固。
2. 相互调换网线。

解决方案：更换或正确插入网线

操作步骤

- 更换或正确插入网线。

原因：因外部系统软件错误而无以太网连接

说明

由于外部系统的软件存在错误，因此没有以太网连接。

检测说明

- 检查外部系统是否存在软件错误。

解决方案：排除外部系统软件中的错误

说明

必须排除外部系统软件中的错误。

原因：因外部系统硬件故障而无以太网连接

说明

由于外部系统存在硬件故障，例如插口接触不良、网卡损坏等，因此没有以太网连接。

检测说明


- 检查外部系统是否存在硬件故障。

解决方案：排除外部系统硬件故障

说明

必须排除外部系统硬件故障。

9.3.15 EKI00019

信息代码	EKI00019
信息文本	不存在需要发送的数据
信息类型	故障信息
	
作用	轨道紧急停止 被激活的指令（机器人运动，程序启动）的输入被禁。
可能的原因	<b>原因：</b> EKI_Send() 未包含待发送的数据（>>> 页面 83） <b>解决方案：</b> 给函数正确编程（>>> 页面 84）

原因：EKI\_Send() 未包含待发送的数据

说明

发送函数 EKI\_Send() 未包含待发送的数据（参数 2）。

RET = EKI_Send(CHAR[], CHAR[], INT)	
功能	通过通道发送数据 (>>> 6.2.4 “发送数据” 页面 36)
参数 1	类型: CHAR 应通过其发送的打开的通道名称
参数 2	类型: CHAR 定义待发送数据的范围。 通过 XML 结构通讯时: <ul style="list-style-type: none"><li>为发送数据所配置的 XML 结构中的待发送元素的 XPath 表达式。<ul style="list-style-type: none"><li>如果仅给出了根元素, 将传输整个 XML 结构。参见示例 1。</li><li>如果只应传输 XML 结构的一部分, 即某个下级层的元素, 必须给出从根元素至该元素的路径。参见示例 2。</li></ul></li><li>可选: 应传输的可变长度的任意字符串</li></ul> 通过原始数据通讯时: <ul style="list-style-type: none"><li>应传输的任意字符串:<ul style="list-style-type: none"><li>对于有固定长度的二进制数据组 (属性 Type = “BYTE”): 有固定长度的任意字符串 用字节表示的字符串大小必须与配置的属性 Size (大小) 完全相符。超出时将输出错误信息, 低于时将输出警告信息。</li><li>对于有可变末尾字符串的二进制数据组 (属性 Type = “STREAM”): 有可变长度的任意字符串 自动一并发送末尾字符串。</li></ul></li></ul> <b>提示:</b> 如果可变长度的任意字符串包含一个 ASCII 零字符, 将仅传输到该字符的部分。如果此非所愿, 则必须相应地定义参数 3。
参数 3 (可选)	类型: INT 只有在发送长度可变的任意字符串时才相关 (见参数 2): 最大待发送的字符数量
RET	类型: EKI_STATUS 函数的返回值
示例 1	RET = EKI_Send(“Channel_1”, “Root”)
示例 2	RET = EKI_Send(“Channel_1”, “Root/Test”)
示例 3	RET = EKI_Send(“Channel_1”, MyBytes[])
示例 4	RET = EKI_Send(“Channel_1”, MyBytes[], 6)

检测说明

- 在 KRL 程序中检查是否给出待发送的数据。

解决方案：给函数正确编程

说明

函数必须正确编程。

前提条件

- 用户权限:
  - 专家用户组



视配置和使用的系统软件而定，可能需要更高的用户权限。




- 运行模式 T1、T2 或 AUT

操作步骤

1. 在导航器中选定程序并按**打开**。在编辑器中显示该程序。
2. 搜索和编辑程序中相应的位置。
3. 关闭文件，对于是否要保存更改的安全询问，回答**是**。

9.3.16 EKI00020

信息代码	EKI00020
信息文本	数据类型不匹配
信息类型	故障信息
	
作用	轨道紧急停止  被激活的指令（机器人运动，程序启动）的输入被禁。
可能的原因	<b>原因：</b> 用于接收数据的 XML 结构中的数据类型配置错误（>>> 页面 85） <b>解决方案：</b> 排除 xml 文件中的错误（>>> 页面 86）  <b>原因：</b> 存取函数中的数据类型编程错误（>>> 页面 87） <b>解决方案：</b> 给函数正确编程（>>> 页面 88）

原因：用于接收数据的 XML 结构中的数据类型配置错误

说明

存取函数中给出的元素的数据类型与用于接收数据的 XML 结构中为此配置的数据类型不一致。

存取函数

一个存取函数中的参数 3 分别给出要存取的元素的数据类型。

存取函数
EKI_STATUS = EKI_GetBool (CHAR[], CHAR[], BOOL)
EKI_STATUS = EKI_GetBoolArray (CHAR[], CHAR[], BOOL[])
EKI_STATUS = EKI_GetInt (CHAR[], CHAR[], INT)
EKI_STATUS = EKI_GetIntArray (CHAR[], CHAR[], INT[])
EKI_STATUS = EKI_GetReal (CHAR[], CHAR[], REAL)
EKI_STATUS = EKI_GetRealArray (CHAR[], CHAR[], REAL[])
EKI_STATUS = EKI_GetString (CHAR[], CHAR[], CHAR[])
EKI_STATUS = EKI_GetFrame (CHAR[], CHAR[], FRAME)
EKI_STATUS = EKI_GetFrameArray (CHAR[], CHAR[], FRAME[])

配置文件

针对每个以太网连接，必须配置一个 xml 文件。xml 文件的名称同时也是 KRL 的访问键。

目录	C:\KRC\ROBOTER\Config\User\Common\EthernetKRL
文件	示例: ...\EXT.xml → EKI_INIT("EXT")

在 xml 文件的段 <RECEIVE> ... </RECEIVE> 中配置了接收结构。属性 Type 定义了元素的数据类型。

检测说明

- 检查编程的数据类型是否与接收结构中配置的元素的数据类型一致。

示例

读取 XML 数据:

XML 结构:

```
<RECEIVE>
  <XML>
    <ELEMENT Tag="Sensor/Message" Type="STRING" />
    <ELEMENT Tag="Sensor/Status/IsActive" Type="BOOL" />
  </XML>
</RECEIVE>
```

编程:

```
...
CHAR valueChar[256]
BOOL valueBOOL
...
RET=EKI_GetString("Channel_1", "Sensor/Message", valueChar[])
RET=EKI_GetBool("Channel_1", "Sensor/Status/IsActive", valueBOOL)
```

读取固定长度（10 字节）的二进制数据组:

- 原始数据:

```
<RECEIVE>
  <RAW>
    <ELEMENT Tag="Buffer" Type="BYTE" Size="10" />
  </RAW>
</RECEIVE>
```

- 编程:

```
...
CHAR Bytes[10]
...
RET=EKI_GetString("Channel_1", "Buffer", Bytes[])
```

解决方案: 排除 xml 文件中的错误

说明

必须排除 xml 文件中的错误。

前提条件

- 专家用户组

- 运行方式 T1 、T2 或 AUT

操作步骤

1. 根据需要更改 xml 文件并保存更改。
2. 如果已离线更改 xml 文件，将其复制到规定的目录中，覆盖旧的 xml 文件。

原因：存取函数中的数据类型编程错误

说明

存取函数中给出的元素的数据类型与用于接收数据的 XML 结构中为此配置的数据类型不一致。

存取函数

一个存取函数中的参数 3 分别给出要存取的元素的数据类型。

存取函数
EKI_STATUS = EKI_GetBool (CHAR[], CHAR[], BOOL)
EKI_STATUS = EKI_GetBoolArray (CHAR[], CHAR[], BOOL[])
EKI_STATUS = EKI_GetInt (CHAR[], CHAR[], INT)
EKI_STATUS = EKI_GetIntArray (CHAR[], CHAR[], INT[])
EKI_STATUS = EKI_GetReal (CHAR[], CHAR[], REAL)
EKI_STATUS = EKI_GetRealArray (CHAR[], CHAR[], REAL[])
EKI_STATUS = EKI_GetString (CHAR[], CHAR[], CHAR[])
EKI_STATUS = EKI_GetFrame (CHAR[], CHAR[], FRAME)
EKI_STATUS = EKI_GetFrameArray (CHAR[], CHAR[], FRAME[])

配置文件

针对每个以太网连接，必须配置一个 xml 文件。xml 文件的名称同时也是 KRL 的访问键。

目录	C:\KRC\ROBOTER\Config\User\Common\EthernetKRL
文件	示例： ...\EXT.xml → EKI_INIT("EXT")

在 xml 文件的段 <RECEIVE> ... </RECEIVE> 中配置了接收结构。属性 Type 定义了元素的数据类型。

检测说明

- 检查编程的数据类型是否与接收结构中配置的元素的数据类型一致。

示例

读取 XML 数据：

XML 结构：

```
<RECEIVE>
  <XML>
    <ELEMENT Tag="Sensor/Message" Type="STRING" />
    <ELEMENT Tag="Sensor/Status/IsActive" Type="BOOL" />
  </XML>
</RECEIVE>
```

编程：

```
...
CHAR valueChar[256]
BOOL valueBOOL
...
RET=EKI_GetString("Channel_1", "Sensor/Message", valueChar[])
RET=EKI_GetBool("Channel_1", "Sensor/Status/IsActive", valueBOOL)
```

读取固定长度（10 字节）的二进制数据组：

- 原始数据：

```
<RECEIVE>
<RAW>
  <ELEMENT Tag="Buffer" Type="BYTE" Size="10" />
</RAW>
</RECEIVE>
```

- 编程：

```
...
CHAR Bytes[10]
...
RET=EKI_GetString("Channel_1", "Buffer", Bytes[])
```

解决方案：给函数正确编程

说明

函数必须正确编程。

前提条件

- 用户权限：
  - 专家用户组




视配置和使用的系统软件而定，可能需要更高的用户权限。

- 运行模式 T1、T2 或 AUT

操作步骤

1. 在导航器中选定程序并按**打开**。在编辑器中显示该程序。
2. 搜索和编辑程序中相应的位置。
3. 关闭文件，对于是否要保存更改的安全询问，回答**是**。

9.3.17 EKI00021

信息代码	EKI00021
信息文本	最大数据保存时系统存储器空间不足
信息类型	故障信息
	
作用	轨道紧急停止
	被激活的指令（机器人运动，程序启动）的输入被禁。



可能的原因	原因：预留内存不足 (>>> 页面 89)
	解决方案：增加内存 (>>> 页面 89)
	原因：连接配置占用过多内存 (>>> 页面 89)
	解决方案：更改连接配置 (>>> 页面 90)

原因：预留内存不足

说明

为 Ethernet KRL 预留的内存不足。

检测说明

- 检查是否可通过另一种配置或编程减少所需内存。
- 如果不可行，允许在与 KUKA Deutschland GmbH 协商后增加内存。

解决方案：增加内存



仅允许在与 KUKA Deutschland GmbH 协商后增加内存。 (>>> 11 “KUKA 服务” 页面 115)

前提条件

- 用户权限：
  - 专家用户组



视配置和使用的系统软件而定，可能需要更高的用户权限。

- 运行模式 T1、T2 或 AUT

操作步骤

1. 在目录 C:\KRC\ROBOTER\Config\User\Common 中打开文件 EthernetKRL.XML。
2. 在段 <EthernetKRL> 的元素 <MemSize> 中输入所需的内存容量，单位：字节。

```
<EthernetKRL>
  <Interface>
    <MemSize>1048576</MemSize>
  ...
</EthernetKRL>
```

3. 关闭文件，对于是否要保存更改的安全询问，回答是。
4. 重新启动机器人控制系统，并采用设置冷启动和重新读入文件。

原因：连接配置占用过多内存

说明

配置的以太网连接占用过多内存。

配置文件

针对每个以太网连接，必须配置一个 xml 文件。xml 文件的名称同时也是 KRL 的访问键。

目录	C:\KRC\ROBOTER\Config\User\Common\EthernetKRL
文件	示例: ...\EXT.xml → EKI_INIT("EXT")

检测说明

- 检查 xml 文件，以确定在配置以太网连接时是否可减少占用内存。

解决方案：更改连接配置

说明

更改连接配置，以减少占用内存。


前提条件

- 专家用户组
- 运行方式 T1 、T2 或 AUT

操作步骤

1. 根据需要更改 xml 文件并保存更改。
2. 如果已离线更改 xml 文件，将其复制到规定的目录中，覆盖旧的 xml 文件。

9.3.18 EKI00022

信息代码	EKI00022
信息文本	读取配置时出错。XML 错误。
信息类型	故障信息
	
作用	轨道紧急停止  被激活的指令（机器人运动，程序启动）的输入被禁。
可能的原因	<b>原因：</b> 包含连接配置的 xml 文件中的架构错误（>>> 页面 90） <b>解决方案：</b> 排除 xml 文件中的错误（>>> 页面 91）

原因：包含连接配置的 xml 文件中的架构错误

说明

包含连接配置的 xml 文件可能由于架构错误而无法读取。  
KUKA.Ethernet KRL 使用 XPath 架构。必须严格遵守架构规定的句法。例如，不允许缺少句号或结构元素。  
xml 文件中元素和属性的写法（包括大写和/或小写）均有规定，必须严格遵守。



关于 XPath 架构的更多信息请参见 KUKA.Ethernet KRL 的文档。

配置文件

针对每个以太网连接，必须配置一个 xml 文件。xml 文件的名称同时也是 KRL 的访问键。

目录	C:\KRC\ROBOTER\Config\User\Common\EthernetKRL
文件	示例: ...\EXT.xml —> EKI_INIT("EXT")

解决方案：排除 xml 文件中的错误

说明

必须排除 xml 文件中的错误。


前提条件

- 专家用户组
- 运行方式 T1 、T2 或 AUT

操作步骤

1. 根据需要更改 xml 文件并保存更改。
2. 如果已离线更改 xml 文件，将其复制到规定的目录中，覆盖旧的 xml 文件。

9.3.19 EKI00023

信息代码	EKI00023
信息文本	初始化已完成。
信息类型	故障信息
	
作用	轨道紧急停止  被激活的指令（机器人运动，程序启动）的输入被禁。
可能的原因	<b>原因：</b> 已用 EKI_Init() 初始化以太网连接 (>>> 页面 91) <b>解决方案：</b> 删除过度编程的函数 (>>> 页面 92)

原因：已用 EKI\_Init() 初始化以太网连接

说明

以太网连接已用函数 EKI\_Init() 初始化。

RET = EKI_Init (CHAR[])	
功能	初始化一个通道，用于进行以太网通讯 将执行下列操作： <ul style="list-style-type: none"><li>• 读入连接配置</li><li>• 创建数据存储器</li><li>• 准备以太网连接</li></ul>
参数	类型：CHAR 通道名称 (= 包含连接配置的 xml 文件名称)
RET	类型：EKI_STATUS 函数的返回值
示例	RET = EKI_Init ("Channel_1")

因此可以检查连接是否已初始化：

前提

- 专家用户群
- 已选定或者已打开程序。

检测说明

- 检查是否在连接的首次初始化和关闭之间多次编程设定下列行：  
RET = EKI\_Init ("文件名")
  - 文件名：包含连接配置的 xml 文件名称

解决方案：删除过度编程的函数

说明

必须删除过度编程的函数。

前提条件

- 用户权限：
  - 专家用户组



视配置和使用的系统软件而定，可能需要更高的用户权限。


- 运行模式 T1、T2 或 AUT

操作步骤

1. 在导航器中选定程序并按**打开**。在编辑器中显示该程序。
2. 搜索和编辑程序中相应的位置。
3. 关闭文件，对于是否要保存更改的安全询问，回答**是**。

9.3.20 EKI00024

信息代码	EKI00024
信息文本	绑定到内部参数（端口，IP）失败

信息类型	故障信息 
作用	轨道紧急停止 被激活的指令（机器人运动，程序启动）的输入被禁。
可能的原因	<b>原因：</b> EKI 的 IP 地址和/或端口号未给出或给出错误（>>> 页面 93） <b>解决方案：</b> 在 xml 文件中输入正确的 IP 地址和端口号（>>> 页面 93）

原因：EKI 的 IP 地址和/或端口号未给出或给出错误

说明

EKI 配置为服务器，而外部系统则配置为客户端。在包含连接配置的 xml 文件中，EKI 的 IP 地址和/或端口号未给出或给出的形式错误。为了确保网络安全，可以规定能够从哪个子网访问 EKI。

示例

只可以通过子网 192.168.X.X 连接端口 54600：

```
<INTERNAL>
<IP>192.168.23.1</IP>
<PORT>54600</PORT>
</INTERNAL>
```

可以通过所有已在 KLI 中设置的子网来连接端口 54600：

```
<INTERNAL>
<IP>0.0.0.0</IP>
<PORT>54600</PORT>
</INTERNAL>
```

配置文件

针对每个以太网连接，必须配置一个 xml 文件。xml 文件的名称同时也是 KRL 的访问键。

目录	C:\KRC\ROBOTER\Config\User\Common\EthernetKRL
文件	示例：...\EXT.xml —> EKI_INIT("EXT")

检测说明

- 在包含连接配置的 xml 文件中检查是否正确输入了 IP 地址和端口号。

解决方案：在 xml 文件中输入正确的 IP 地址和端口号


前提条件

- 专家用户组
- 运行方式 T1 、T2 或 AUT

操作步骤

- 1. 根据需要更改 xml 文件并保存更改。
- 2. 如果已离线更改 xml 文件，将其复制到规定的目录中，覆盖旧的 xml 文件。

9.3.21 EKI00027

信息代码	EKI00027
信息文本	KRL CHAR[] 数组过小。
信息类型	故障信息
	
作用	轨道紧急停止  被激活的指令（机器人运动，程序启动）的输入被禁。
可能的原因	<b>原因：</b> CHAR 数组对于已接收的数据过小（>>> 页面 94） <b>解决方案：</b> 增加 CHAR 数组的存储空间（>>> 页面 94）

原因：CHAR 数组对于已接收的数据过小

说明

KRL 程序中定义的 CHAR 数组对于已接收的数据过小。

检测说明

- 检查外部系统发送的数据组最大可以有多长。

解决方案：增加 CHAR 数组的存储空间

说明

必须增加 CHAR 数组的存储空间。

前提条件

- 用户权限：
  - 专家用户组




视配置和使用的系统软件而定，可能需要更高的用户权限。

- 运行模式 T1、T2 或 AUT

操作步骤

- 1. 在导航器中选定程序并按**打开**。在编辑器中显示该程序。
- 2. 搜索和编辑程序中相应的位置。
- 3. 关闭文件，对于是否要保存更改的安全询问，回答**是**。

9.3.22 EKI00512

信息代码	EKI00512
信息文本	以太网连接故障
信息类型	故障信息
	
作用	轨道紧急停止 被激活的指令（机器人运动，程序启动）的输入被禁。
可能的原因	<b>原因：</b> 网线损坏或未正确连接（>>> 页面 95） <b>解决方案：</b> 更换或正确插入网线（>>> 页面 95）  <b>原因：</b> 因外部系统软件错误而无以太网连接（>>> 页面 95） <b>解决方案：</b> 排除外部系统软件中的错误（>>> 页面 95）  <b>原因：</b> 因外部系统硬件故障而无以太网连接（>>> 页面 96） <b>解决方案：</b> 排除外部系统硬件故障（>>> 页面 96）

原因：网线损坏或未正确连接

说明

网线损坏或插接错误。  
如此可检查网线和连接插头是否正确连接：

检测说明

1. 检查网线是否插接正确以及是否牢固。
2. 相互调换网线。

解决方案：更换或正确插入网线

操作步骤

- 更换或正确插入网线。

原因：因外部系统软件错误而无以太网连接

说明

由于外部系统的软件存在错误，因此没有以太网连接。

检测说明

- 检查外部系统是否存在软件错误。

解决方案：排除外部系统软件中的错误

说明

必须排除外部系统软件中的错误。

原因：因外部系统硬件故障而无以太网连接

说明

由于外部系统存在硬件故障，例如插口接触不良、网卡损坏等，因此没有以太网连接。

检测说明


- 检查外部系统是否存在硬件故障。

解决方案：排除外部系统硬件故障

说明

必须排除外部系统硬件故障。

9.3.23 EKI00768

信息代码	EKI00768
信息文本	Ping 报告没有连接
信息类型	故障信息
	
作用	轨道紧急停止 被激活的指令（机器人运动，程序启动）的输入被禁。
可能的原因	<b>原因：</b> 网线损坏或未正确连接（>>> 页面 96） <b>解决方案：</b> 更换或正确插入网线（>>> 页面 96） <b>原因：</b> 因外部系统硬件故障而无以太网连接（>>> 页面 97） <b>解决方案：</b> 排除外部系统硬件故障（>>> 页面 97）

原因：网线损坏或未正确连接

说明

网线损坏或插接错误。  
如此可检查网线和连接插头是否正确连接：

检测说明

1. 检查网线是否插接正确以及是否牢固。
2. 相互调换网线。

解决方案：更换或正确插入网线

操作步骤

- 更换或正确插入网线。



原因：因外部系统硬件故障而无以太网连接

说明

由于外部系统存在硬件故障，例如插口接触不良、网卡损坏等，因此没有以太网连接。

检测说明


- 检查外部系统是否存在硬件故障。

解决方案：排除外部系统硬件故障

说明

必须排除外部系统硬件故障。

9.3.24 EKI01024

信息代码	EKI01024
信息文本	读取接收的 XML 数据时出错
信息类型	故障信息
	
作用	轨道紧急停止 被激活的指令（机器人运动，程序启动）的输入被禁。
可能的原因	<b>原因：</b> 发送的 XML 文件与 XPath 架构不符（>>> 页面 97） <b>解决方案：</b> 按照 XPath 框架构建发送的 XML 文件（>>> 页面 97）

原因：发送的 XML 文件与 XPath 架构不符

说明

外部系统发送的 XML 文件与 XPath 架构不符。




关于 XPath 架构的更多信息请参见 KUKA.Ethernet KRL 的文档。

解决方案：按照 XPath 框架构建发送的 XML 文件

说明

必须按照 XPath 框架构建外部系统发送的 XML 文件。

9.3.25 EKI01280

信息代码	EKI01280
信息文本	已达到可存储元素的极限
信息类型	故障信息
	

作用	轨道紧急停止  被激活的指令（机器人运动，程序启动）的输入被禁。
可能的原因	<b>原因：</b> 达到存储器极限（每个存储器的数据元素数量）（>>> 页面 98） <b>解决方案：</b> 在 xml 文件中提高存储器极限（>>> 页面 98）  <b>原因：</b> 达到存储器极限（每个存储器的数据元素数量）（>>> 页面 99） <b>解决方案：</b> 周期性检查数据存储器并在达到极限前禁用（>>> 页面 99）  <b>原因：</b> EKI 存储器不被读取（>>> 页面 100） <b>解决方案：</b> 更改程序（>>> 页面 100）

原因：达到存储器极限（每个存储器的数据元素数量）

说明

可保存的数据元素数量有限制。已达到这一存储器极限。以太网连接已关闭，以防止接收其他数据。

在包含连接配置的 xml 文件中定义了可保存元素的最大数量：

- 段 <INTERNAL> ... </INTERNAL>
- 元素 <BUFFERING ... Limit="..." />
  - 最大可能的数量：512
- 如果未配置元素 BUFFERING，则采用默认值：
  - <BUFFERING Mode="FIFO" Limit="16" />

配置文件

针对每个以太网连接，必须配置一个 xml 文件。xml 文件的名称同时也是 KRL 的访问键。

目录	C:\KRC\ROBOTER\Config\User\Common\EthernetKRL
文件	<b>示例：</b> ...\EXT.xml —> EKI_INIT("EXT")

检测说明

- 在包含连接配置的 xml 文件中检查可保存元素的数量限制为哪个值。

解决方案：在 xml 文件中提高存储器极限

说明

可在 xml 文件中提高存储器极限。

前提条件

- 专家用户组
- 运行方式 T1 、T2 或 AUT

操作步骤

1. 根据需要更改 xml 文件并保存更改。
2. 如果已离线更改 xml 文件，将其复制到规定的目录中，覆盖旧的 xml 文件。

原因：达到存储器极限（每个存储器的数据元素数量）

说明

可保存的数据元素数量有限制。已达到这一存储器极限。以太网连接已关闭，以防止接收其他数据。  
在包含连接配置的 xml 文件中定义了可保存元素的最大数量：

- 段 <INTERNAL> ... </INTERNAL>
- 元素 <BUFFERING ... Limit="..." />
  - 最大可能的数量：512
- 如果未配置元素 BUFFERING，则采用默认值：
  - <BUFFERING Mode="FIFO" Limit="16" />

配置文件

针对每个以太网连接，必须配置一个 xml 文件。xml 文件的名称同时也是 KRL 的访问键。

目录	C:\KRC\ROBOTER\Config\User\Common\EthernetKRL
文件	示例：... \EXT.xml —> EKI_INIT("EXT")

检测说明

- 在包含连接配置的 xml 文件中检查可保存元素的数量限制为哪个值。

解决方案：周期性检查数据存储器并在达到极限前禁用

说明

通过编程可防止达到存储器极限。为此，周期性检查当前在数据存储器中有多少数据元素，并在达到极限前暂时禁用存储器。  
需要以下函数：

- EKI\_STATUS = EKI\_CheckBuffer (CHAR[], CHAR[])  
检查存储器中还有多少数据。不改变存储器。
- EKI\_STATUS = EKI\_Lock (CHAR[])  
禁用对已接收的数据进行处理。数据无法再储存在存储器中。
- EKI\_STATUS = EKI\_Unlock (CHAR[])  
解禁对已接收的数据进行处理。数据重新储存在存储器中。

EKI\_STATUS 是函数的返回值写入的全局结构变量。与分析相关的是 EKI\_STATUS 的 Buff 元素。

Buff 包含下列数值：

- 存取后存储器中仍存在的元素数量

句法

GLOBAL STRUC EKI\_STATUS INT Buff, Read, Msg\_No, BOOL Connected, INT Counter

示例

```
...
DECL INT limit
DECL EKI_STATUS ret
...
limit = 16
```

```
ret = EKI_CheckBuffer("MyChannel", "MyData")

IF(ret.Buff >= limit - 1) THEN
    ret = EKI_Lock("MyChannel")
ELSE
    ret = EKI_Unlock("MyChannel")
ENDIF
```

原因：EKI 存储器不被读取

说明

外部系统将数据写入 EKI 存储器。如果这些数据无法调用，EKI 存储器即被存满。为了避免产生数据损失，在 EKI 存储器被存满后，连接就会被关闭。TCP 协议识别出连接被关闭，并中断数据交换。最大数量的数据元素被写入到 EKI 存储器中，同时并不对其进行读取。

因此可以检查在 KRL 程序中 EKI 存储器是否被读取：

前提

- 已选定或者已打开程序。

检测说明

- 检查是否通过 EKI\_Get...() 读取数据。

解决方案：更改程序


前提条件

- 专家用户组
- 运行方式 T1 、T2 或 AUT

操作步骤

1. 在导航器中选定程序并按**打开**。在编辑器中显示该程序。
2. 搜索和编辑程序中相应的位置。
3. 关闭文件，对于是否要保存更改的安全询问，回答**是**。

9.3.26 EKI01536

信息代码	EKI01536
信息文本	接收的字符串过长
信息类型	故障信息
	
作用	轨道紧急停止
	被激活的指令（机器人运动，程序启动）的输入被禁。
可能的原因	<b>原因：</b> 外部系统发送的字符串过长（>>> 页面 101） <b>解决方案：</b> 调整外部系统的编程（>>> 页面 101）

原因：外部系统发送的字符串过长

说明

外部系统发送的字符串超出了允许的最大长度。最多允许 3600 个字符。

检测说明


- 检查外部系统发送的数据长度。

解决方案：调整外部系统的编程

说明

必须调整外部系统的编程，以便不再发送过长的字符串。

9.3.27 EKI01792

信息代码	EKI01792
信息文本	已达到接收存储器的限制
信息类型	故障信息
	
作用	轨道紧急停止  被激活的指令（机器人运动，程序启动）的输入被禁。
可能的原因	<b>原因：</b> 达到存储器极限（总内存字节数）（>>> 页面 101） <b>解决方案：</b> 在 xml 文件中提高存储器极限（>>> 页面 102）

原因：达到存储器极限（总内存字节数）

说明

可保存的字节数有限制。已达到这一存储器极限。以太网连接已关闭，以防止接收其他数据。

在包含连接配置的 xml 文件中定义了可保存字节的最大数量：

- 段 <INTERNAL> ... </INTERNAL>
- 元素 <BUFSIZE Limit="..." />
  - 最大可能的数量：65534
- 如果未配置元素 BUFSIZE，则默认值是：
  - <BUFSIZE Limit="16384" />

配置文件

针对每个以太网连接，必须配置一个 xml 文件。xml 文件的名称同时也是 KRL 的访问键。

目录	C:\KRC\ROBOTER\Config\User\Common\EthernetKRL
文件	<b>示例：</b> ...\EXT.xml —> EKI_INIT("EXT")

检测说明

- 在包含连接配置的 xml 文件中检查字节数限制为哪个值。

解决方案：在 xml 文件中提高存储器极限

说明

可在 xml 文件中提高存储器极限。


前提条件

- 专家用户组
- 运行方式 T1 、T2 或 AUT

操作步骤

1. 根据需要更改 xml 文件并保存更改。
2. 如果已离线更改 xml 文件，将其复制到规定的目录中，覆盖旧的 xml 文件。

9.3.28 EKI02048

信息代码	EKI02048
信息文本	已达到服务器时间限制
信息类型	故障信息
	
作用	轨道紧急停止 被激活的指令（机器人运动，程序启动）的输入被禁。
可能的原因	<b>原因：</b> 外部系统上的软件错误导致超时（>>> 页面 102） <b>解决方案：</b> 排除外部系统软件中的错误（>>> 页面 102）

原因：外部系统上的软件错误导致超时

说明

EKI 配置为服务器，并等待外部系统的连接问询。  
包含连接配置的 xml 文件中给出了 <TIMEOUT Connect=“...”/> 的值。这段时间已过，但外部系统并未与服务器连接。原因是外部系统软件中存在错误。

检测说明

- 检查外部系统是否存在软件错误。

解决方案：排除外部系统软件中的错误

说明

必须排除外部系统软件中的错误。

10 附录

10.1 连接属性的扩展 XML 结构



仅在与 KUKA Deutschland GmbH 协商后，才允许使用扩展的 XML 结构。  
(>>> 11 “KUKA 服务” 页面 115)

说明

在配置文件的段 <INTERNAL> ... </INTERNAL> 中，可为 EKI 配置其他属性：

元素	属性	说明
TIMEOUT	Receive	尝试接收数据的时间，超过这段时间后将中止尝试（可选）  • 0 ... 65 534 ms 默认值：0 ms
	Send	尝试发送数据的时间，超过这段时间后将中止尝试（可选）  • 0 ... 65 534 ms 默认值：0 ms
BUFFSIZE	Receive	接收数据时所用的套接字大小（可选）  • 1 ... 65 534 字节 默认值：由系统规定
	Send	发送数据时所用的套接字大小（可选）  • 1 ... 65 534 字节 默认值：由系统规定

10.2 增加内存



仅允许在与 KUKA Deutschland GmbH 协商后增加内存。（>>> 11 “KUKA 服务” 页面 115）

说明

- 如果可用的内存不够，建议检查 KRL 中的编程方法和配置。
- 检查在配置连接时，是否使得内存完全被接收的数据占用。
  - 检查是否定义并激活了数据传输量大的多个连接。

前提条件

- 用于编辑文件的用户权限：
  - 功能组文件操作
- 用于用重新读入文件关闭的用户权限：
  - 功能组临界配置
- 运行模式 T1、T2 或 AUT

## 操作步骤

1. 在目录 C:\KRC\ROBOTER\Config\User\Common 中打开文件 EthernetKRL.XML。
2. 在段 <EthernetKRL> 的元素 <MemSize> 中输入所需的内存容量，单位：字节。

```
<EthernetKRL>
  <Interface>
    <MemSize>1048576</MemSize>
  ...
</EthernetKRL>
```

3. 关闭文件，对于是否要保存更改的安全询问，回答是。
4. 重新启动机器人控制系统，并采用设置冷启动和重新读入文件。

## 10.3 禁用信息输出和信息记录

### 说明

在下列情况下，建议禁用 smartHMI 上的自动输出信息：

- 出现运行时间错误。
- 在提交解释器中使用 EKI。

自动信息输出被禁用时，默认情况下继续记录所有错误信息。如果有意识地忽略这些错误或警告（例如因为记录信息会导致系统负荷过高或系统运行速度放慢），则同样可禁用该机制。

### 前提条件

- 用户权限：功能组**文件操作**  
但至少为用户组 **专家**

### 操作步骤

1. 在机器人控制系统的目录 C:\KRC\ROBOTER\Config\User\Common \EthernetKRL 中打开以太网连接的配置文件。
2. 在 xml 文件的段 <INTERNAL> ... </INTERNAL> 中输入下列行：

```
<MESSAGES Logging="disabled" Display="disabled"/>
```

3. 保存更改并关闭文件。



如果自动信息输出已禁用，可使用函数 EKI\_CHECK() 检查各个 EKI 指令是否存在错误。



10.4 指令参考

10.4.1 初始化、打开、关闭和删除连接

RET = EKI_Init (CHAR[])	
功能	初始化一个通道，用于进行以太网通讯 将执行下列操作： <ul style="list-style-type: none"><li>• 读入连接配置</li><li>• 创建数据存储器</li><li>• 准备以太网连接</li></ul>
参数	类型：CHAR 通道名称 (= 包含连接配置的 xml 文件名称)
RET	类型：EKI_STATUS 函数的返回值
示例	RET = EKI_Init ("Channel_1")
RET = EKI_Open (CHAR[])	
功能	打开初始化的通道  EKI 配置为客户端时，EKI 将与外部系统 (= 服务器) 连接。  EKI 配置为服务器时，EKI 将等待外部系统 (= 客户端) 的连接问询。
参数	类型：CHAR 信道名称
RET	类型：EKI_STATUS 通道名称 (= 包含连接配置的 xml 文件名称)
示例	RET = EKI_Open ("Channel_1")
RET = EKI_Close (CHAR[])	
功能	关闭打开的通道
参数	类型：CHAR 通道名称 (= 包含连接配置的 xml 文件名称)
RET	类型：EKI_STATUS 函数的返回值
示例	RET = EKI_Close ("Channel_1")
RET = EKI_Clear (CHAR[])	
功能	删除一个通道及所有相关数据存储器，退出以太网连接。
参数	类型：CHAR 通道名称 (= 包含连接配置的 xml 文件名称)
RET	类型：EKI_STATUS 函数的返回值
示例	RET = EKI_Clear ("Channel_1")

关于相应函数的返回值的信息请参见： (>>> [“返回值” 页面 42](#))

10.4.2 发送数据

RET = EKI_Send(CHAR[], CHAR[], INT)	
功能	通过通道发送数据 (>>> 6.2.4 “发送数据” 页面 36)
参数 1	类型: CHAR 应通过其发送的打开的通道名称
参数 2	类型: CHAR 定义待发送数据的范围。 通过 XML 结构通讯时: <ul style="list-style-type: none"><li>为发送数据所配置的 XML 结构中的待发送元素的 XPath 表达式。<ul style="list-style-type: none"><li>如果仅给出了根元素, 将传输整个 XML 结构。参见示例 1。</li><li>如果只应传输 XML 结构的一部分, 即某个下级层的元素, 必须给出从根元素至该元素的路径。参见示例 2。</li></ul></li><li>可选: 应传输的可变长度的任意字符串</li></ul> 通过原始数据通讯时: <ul style="list-style-type: none"><li>应传输的任意字符串:<ul style="list-style-type: none"><li>对于有固定长度的二进制数据组 (属性 Type = “BYTE”): 有固定长度的任意字符串 用字节表示的字符串大小必须与配置的属性 Size (大小) 完全相符。超出时将输出错误信息, 低于时将输出警告信息。</li><li>对于有可变末尾字符串的二进制数据组 (属性 Type = “STREAM”): 有可变长度的任意字符串 自动一并发送末尾字符串。</li></ul></li></ul> <b>提示:</b> 如果可变长度的任意字符串包含一个 ASCII 零字符, 将仅传输到该字符的部分。如果此非所愿, 则必须相应地定义参数 3。
参数 3 (可选)	类型: INT 只有在发送长度可变的任意字符串时才相关 (见参数 2): 最大待发送的字符数量
RET	类型: EKI_STATUS 函数的返回值
示例 1	RET = EKI_Send(“Channel_1”, “Root”)
示例 2	RET = EKI_Send(“Channel_1”, “Root/Test”)
示例 3	RET = EKI_Send(“Channel_1”, MyBytes[])
示例 4	RET = EKI_Send(“Channel_1”, MyBytes[], 6)

关于相应函数的返回值的信息请参见: (>>> “返回值” 页面 42)

10.4.3 写入数据

RET = EKI_SetBool(CHAR[], CHAR[], BOOL)	
功能	将一个布尔值写入存储器
参数 1	类型: CHAR 打开的通道名称

RET = EKI_SetBool(CHAR[], CHAR[], BOOL)	
参数 2	类型: CHAR XML 结构中的位置名称
参数 3	类型: BOOL 写入存储器的值
RET	类型: EKI_STATUS 函数的返回值
示例	RET = EKI_SetBool("Channel_1", "Root/Activ", true)

RET = EKI_SetFrame(CHAR[], CHAR[], FRAME)	
功能	将一个 FRAME 类型的值写入存储器
参数 1	类型: CHAR 打开的通道名称
参数 2	类型: CHAR XML 结构中的位置名称
参数 3	类型: FRAME 写入存储器的值
RET	类型: EKI_STATUS 函数的返回值
示例	RET= EKI_SetFrame("Channel_1", "Root/BASE", {X 0.0, Y 0.0, Z 0.0, A 0.0, B 0.0, C 0.0})

RET = EKI_SetInt(CHAR[], CHAR[], INT)	
功能	将一个整数值写入存储器
参数 1	类型: CHAR 打开的通道名称
参数 2	类型: CHAR XML 结构中的位置名称
参数 3	类型: INT 写入存储器的值
RET	类型: EKI_STATUS 函数的返回值
示例	RET = EKI_SetInt("Channel_1", "Root/List", 67234)

RET = EKI_SetReal(CHAR[], CHAR[], REAL)	
功能	将一个浮点值写入存储器
参数 1	类型: CHAR 打开的通道名称
参数 2	类型: CHAR XML 结构中的位置名称
参数 3	类型: REAL 写入存储器的值

RET = EKI_SetReal (CHAR[], CHAR[], REAL)	
RET	类型: EKI_STATUS 函数的返回值
示例	RET = EKI_SetReal ("Channel_1", "Root/Number", 1.234)
RET = EKI_SetString (CHAR[], CHAR[], CHAR[])	
功能	将一个字符串写入存储器
参数 1	类型: CHAR 打开的通道名称
参数 2	类型: CHAR XML 结构中的位置名称
参数 3	类型: CHAR 写入存储器的字符串 最大字符数: <ul style="list-style-type: none"><li>• 3 600</li></ul>
RET	类型: EKI_STATUS 函数的返回值
示例	RET = EKI_SetString ("Channel_1", "Root/Message", "Hello")

关于相应函数的返回值的的信息请参见： (>>> [“返回值” 页面 42](#))

10.4.4 读取数据

RET = EKI_GetBool (CHAR[], CHAR[], BOOL)	
功能	从存储器读取布尔值
参数 1	类型: CHAR 打开的通道名称
参数 2	类型: CHAR XML 结构中的位置名称
参数 3	类型: BOOL 从存储器中读取的值
RET	类型: EKI_STATUS 函数的返回值
示例	RET = EKI_GetBool ("Channel_1", "Root/Activ", MyBool)
RET = EKI_GetBoolArray (CHAR[], CHAR[], BOOL[])	
功能	从存储器读取布尔值，并将该值复制到 KRL 程序传输的数组中 将一直读取值，直至数组满或不再有元素为止。
参数 1	类型: CHAR 打开的通道名称

RET = EKI_GetBoolArray(CHAR[], CHAR[], BOOL[])	
参数 2	类型: CHAR XML 结构中的位置名称
参数 3	类型: BOOL 从存储器中读取的数组 可读取数组元素的最大数量: • 512
RET	类型: EKI_STATUS 函数的返回值
示例	RET = EKI_GetBoolArray("Channel_1", "Root/ Activ", MyBool[])
RET = EKI_GetInt(CHAR[], CHAR[], INT)	
功能	从存储器读取整数值
参数 1	类型: CHAR 打开的通道名称
参数 2	类型: CHAR XML 结构中的位置名称
参数 3	类型: INT 从存储器中读取的值
RET	类型: EKI_STATUS 函数的返回值
示例	RET = EKI_GetInt("Channel_1", "Root/Numbers/ One", MyInteger)
RET = EKI_GetIntArray(CHAR[], CHAR[], INT[])	
功能	从存储器读取整数值，并将该值复制到 KRL 程序传输的数组中 将一直读取值，直至数组满或不再有元素为止。
参数 1	类型: CHAR 打开的通道名称
参数 2	类型: CHAR XML 结构中的位置名称
参数 3	类型: INT 从存储器中读取的数组 可读取数组元素的最大数量: • 512
RET	类型: EKI_STATUS 函数的返回值
示例	RET = EKI_GetIntArray("Channel_1", "Root/ Numbers/One", MyInteger[])

RET = EKI_GetReal(Char[], Char[], Real)	
功能	从存储器读取浮点值
参数 1	类型: CHAR 打开的通道名称
参数 2	类型: CHAR XML 结构中的位置名称
参数 3	类型: REAL 从存储器中读取的值
RET	类型: EKI_STATUS 函数的返回值
示例	RET = EKI_GetReal("Channel_1", "Root/Position", MyReal)

RET = EKI_GetRealArray(Char[], Char[], Real[])	
功能	从存储器读取浮点值，并将该值复制到 KRL 程序传输的数组中 将一直读取值，直至数组满或不再有元素为止。
参数 1	类型: CHAR 打开的通道名称
参数 2	类型: CHAR XML 结构中的位置名称
参数 3	类型: REAL 从存储器中读取的数组 可读取数组元素的最大数量: • 512
RET	类型: EKI_STATUS 函数的返回值
示例	RET = EKI_GetRealArray("Channel_1", "Root/Position", MyReal[])

RET = EKI_GetString(Char[], Char[], Char[])	
功能	从存储器读取字符串
参数 1	类型: CHAR 打开的通道名称
参数 2	类型: CHAR XML 结构中的位置名称或原始数据中的元素名称
参数 3	类型: CHAR 从存储器中读取的字符串 最大字符数: • 3 600
RET	类型: EKI_STATUS 函数的返回值
XML 示例	RET = EKI_GetString("Channel_1", "Root/Message", MyChars[])

RET = EKI_GetString(CHAR[], CHAR[], CHAR[])	
二进制示例	RET = EKI_GetString("Channel_1", "Streams", MyStream[])
RET = EKI_GetFrame(CHAR[], CHAR[], FRAME)	
功能	从存储器读取 FRAME 类型的值
参数 1	类型: CHAR 打开的通道名称
参数 2	类型: CHAR XML 结构中的位置名称
参数 3	类型: FRAME 从存储器中读取的值
RET	类型: EKI_STATUS 函数的返回值
示例	RET = EKI_GetFrame("Channel_1", "Root/TCP", MyFrame)
RET = EKI_GetFrameArray(CHAR[], CHAR[], FRAME[])	
功能	从存储器读取 FRAME 类型的值，并将该值复制到 KRL 程序传输的数组中 将一直读取值，直至数组满或不再有元素为止。
参数 1	类型: CHAR 打开的通道名称
参数 2	类型: CHAR XML 结构中的位置名称
参数 3	类型: FRAME 从存储器中读取的数组 可读取数组元素的最大数量: <ul style="list-style-type: none"><li>• 512</li></ul>
RET	类型: EKI_STATUS 函数的返回值
示例	RET = EKI_GetFrameArray("Channel_1", "Root/TCP", MyFrame[])

关于相应函数的返回值的详细信息请参见： (>>> [“返回值” 页面 42](#))

10.4.5 检查函数是否出现错误

EKI_CHECK(EKI_STATUS, EKrlMsgType, CHAR[])	
功能	检查运行函数时是否出错: <ul style="list-style-type: none"><li>• 将读取错误编号并在 smarthMI 上输出相应的信息。（见参数 1）</li><li>• 可选: 如果给出了通道名称，则在接收数据时会询问是否存在错误（见参数 3）</li></ul>
参数 1	EKI_STATUS 所检查函数的返回值

EKI_CHECK(EKI_STATUS, EKrlMsgType, CHAR[])	
参数 2	类型: ENUM 在 smartHMI 上输出的信息类型: <ul style="list-style-type: none"><li>• #NOTIFY: 提示信息</li><li>• #STATE: 状态信息</li><li>• #QUIT: 确认信息</li><li>• #WAITING: 等待信息</li></ul>
参数 3 (可选)	类型: CHAR 应检查的打开的通道名称
示例 1	EKI_CHECK(RET, #QUIT)
示例 2	EKI_CHECK(RET, #NOTIFY, "MyChannelName")

关于相应函数的返回值的的信息请参见: (>>> [“返回值” 页面 42](#))

10.4.6 删除、禁用、解禁、检查数据存储器

RET = EKI_Clear(CHAR[])	
功能	删除一个通道及所有相关数据存储器，退出以太网连接。
参数	类型: CHAR 通道名称 (= 包含连接配置的 xml 文件名称)
RET	类型: EKI_STATUS 函数的返回值
示例	RET = EKI_Clear("Channel_1")

RET = EKI_ClearBuffer(CHAR[], CHAR[])	
功能	删除定义的数据存储器，但无需退出以太网连接 (>>> <a href="#">6.2.6 “删除数据存储器” 页面 40</a> )
参数 1	类型: CHAR 信道名称
参数 2	类型: CHAR 定义待删除的数据存储器 通过原始数据通讯时: <ul style="list-style-type: none"><li>• 为接收数据所配置的 XML 结构中要删除的原始数据名称</li></ul> 通过 XML 结构通讯时: <ul style="list-style-type: none"><li>• 为接收数据所配置的 XML 结构中或用于发送数据的 XML 结构中要删除数据存储器的元素的 XPATH 表达式</li></ul>
RET	类型: EKI_STATUS 函数的返回值
示例 1	RET = EKI_ClearBuffer("Channel_1", "RawData")
示例 2	RET = EKI_ClearBuffer("Channel_1", "Ext/Activ/Flag")



RET = EKI_Lock (CHAR[])	
功能	禁用对已接收的数据进行处理。数据无法再储存在存储器中。
参数	类型: CHAR 通道名称
RET	类型: EKI_STATUS 函数的返回值
RET = EKI_Unlock (CHAR[])	
功能	解禁对已接收的数据进行处理。数据重新储存在存储器中。
参数	类型: CHAR 通道名称
RET	类型: EKI_STATUS 函数的返回值
RET = EKI_CheckBuffer (CHAR[], CHAR[])	
功能	检查存储器中还有多少数据。不改变存储器。 此外还将返回准备好从存储器中取出的下一个数据元素的时间戳。
参数 1	类型: CHAR 信道名称
参数 2	类型: CHAR 定义待检查的数据存储器 通过原始数据通讯时: <ul style="list-style-type: none"> <li>为接收数据所配置的 XML 结构中的待检查原始数据的名称</li> </ul> 通过 XML 结构通讯时: <ul style="list-style-type: none"> <li>为接收数据所配置的 XML 结构中或用于发送数据的 XML 结构中要检查数据存储器的元素的 XPATH 表达式</li> </ul>
RET	类型: EKI_STATUS 函数的返回值
示例 1	RET = EKI_CheckBuffer ("Channel_1", "RawData")
示例 2	RET = EKI_CheckBuffer ("Channel_1", "Ext/Activ/Flag")

关于相应函数的返回值的信息请参见: (>>> [“返回值” 页面 42](#))



## 11 KUKA 服务

### 11.1 技术支持咨询

#### 引言

该文献将提供有关机器运行及操作的信息，并可帮助您排除故障。当地各分支机构将乐于为您提供详细咨询。

#### 信息

##### 提供咨询时需要以下信息：

- 问题描述，包括故障持续时间及频率的说明
- 关于整个系统硬件和软件组件的尽可能全面的信息

以下列表提供了通常是相关信息的要点：

- 运作系统（例如机械手）的型号及序列号
- 控制系统型号及序列号
- 能量供应系统型号及序列号
- 系统软件名称及版本
- 更多/其他软件组件的名称及版本或修正版
- 诊断程序包 KRCDiag

针对 KUKA Sunrise 另外还需要：现有项目，包括应用程序

针对早于 V8 的 KUKA System Software 版本：软件档案（KRCDiag 在此尚不可用。）

- 现有的应用程序
- 现有的附加轴

### 11.2 KUKA 客户支持系统

#### 可用性

在许多国家内均可使用 KUKA 客户支持系统。如果有问题的话，我们非常乐意为您提供服务！

#### 阿根廷

Ruben Costantini S.A. (Agentur)  
Luis Angel Huergo 13 20  
Parque Industrial  
2400 San Francisco (CBA)  
阿根廷  
电话 +54 3564 421033  
传真 +54 3564 428877  
ventas@costantini-sa.com

#### 澳大利亚

KUKA Robotics Australia Pty Ltd  
45 Fennell Street  
Port Melbourne VIC 3207  
澳大利亚  
电话 +61 3 9939 9656  
info@kuka-robotics.com.au  
www.kuka-robotics.com.au

**比利时**

KUKA Automatisering + Robots N.V.  
Centrum Zuid 1031  
3530 Houthalen  
比利时  
电话 +32 11 516160  
info@kuka.be  
www.kuka.be

**巴西**

KUKA Roboter do Brasil Ltda.  
Travessa Claudio Armando, nº 171  
Bloco 5 - Galpões 51/52  
Bairro Assunção  
CEP 09861-7630 São Bernardo do Campo - SP  
巴西  
电话 +55 11 4942-8299  
传真 +55 11 2201-7883  
info@kuka-roboter.com.br  
www.kuka-roboter.com.br

**智利**

Robotec S.A. (Agency)  
Santiago de Chile  
智利  
电话 +56 2 331-5951  
传真 +56 2 331-5952  
robotec@robotec.cl  
www.robotec.cl

**中国**

KUKA Robotics China Co., Ltd.  
No. 889 Kungang Road  
Xiaokunshan Town  
Songjiang District  
201614 Shanghai  
中华人民共和国  
电话 +86 400 820 8865  
传真 +86 21 5707 2607  
864008208865@kuka.com oder CN-ROB-CS-Hotline@kuka.com  
www.kuka.com

**德国**

KUKA Deutschland GmbH  
Zugspitzstr. 140  
86165 Augsburg  
德国  
电话 +49 821 797-1926  
传真 +49 821 797-41 1926  
CustomerService@kuka.com  
www.kuka.com

**法国**

KUKA Automatisme + Robotique SAS  
Techvallée  
6, Avenue du Parc  
91140 Villebon S/Yvette  
法国  
电话 +33 1 6931660-0  
传真 +33 1 6931660-1  
commercial@kuka.fr  
www.kuka.fr

**印度**

KUKA India Pvt. Ltd.  
Office Number-7, German Centre,  
Level 12, Building No. - 9B  
DLF Cyber City Phase III  
122 002 Gurgaon  
Haryana  
印度  
电话 +91 124 4635774  
传真 +91 124 4635773  
info@kuka.in  
www.kuka.in

**意大利**

KUKA Roboter Italia S.p.A.  
Via Pavia 9/a - int.6  
10098 Rivoli (TO)  
意大利  
电话 +39 011 959-5013  
传真 +39 011 959-5141  
kuka@kuka.it  
www.kuka.it

**日本**

KUKA Japan K.K.  
YBP Technical Center  
134 Godo-cho, Hodogaya-ku  
Yokohama, Kanagawa  
240 0005  
日本  
电话 +81 45 744 7531  
传真 +81 45 744 7541  
info@kuka.co.jp

**加拿大**

KUKA Robotics Canada Ltd.  
2865 Argentia Road, Unit 4-5  
Mississauga  
Ontario L5N 8G6  
加拿大  
电话 +1 905 858-5852  
传真 +1 905 858-8581  
KUKAFocusCenter@KUKARobotics.com  
www.kukarobotics.ca

**韩国**

KUKA Robotics Korea Co. Ltd.  
RIT Center 306, Gyeonggi Technopark  
1271-11 Sa 3-dong, Sangnok-gu  
Ansan City, Gyeonggi Do  
426-901  
韩国  
电话 +82 31 501-1451  
传真 +82 31 501-1461  
info@kukakorea.com

**马来西亚**

KUKA Robot Automation (M) Sdn Bhd  
South East Asia Regional Office  
No. 7, Jalan TPP 6/6  
Taman Perindustrian Puchong  
47100 Puchong  
Selangor  
马来西亚  
电话 +60 (03) 8063-1792  
传真 +60 (03) 8060-7386  
info@kuka.com.my

**墨西哥**

KUKA de México S. de R.L. de C.V.  
Progreso #8  
Col. Centro Industrial Puente de Vigas  
Tlalnepantla de Baz  
54020 Estado de México  
墨西哥  
电话 +52 55 5203-8407  
传真 +52 55 5203-8148  
info@kuka.com.mx  
www.kuka-robotics.com/mexico

**挪威**

KUKA Sveiseanlegg + Roboter  
Sentrumsvegen 5  
2867 Hov  
挪威  
电话 +47 61 18 91 30  
传真 +47 61 18 62 00  
info@kuka.no

**奥地利**

KUKA CEE GmbH  
Gruberstraße 2-4  
4020 Linz  
奥地利  
电话 +43 732 784 752 0  
传真 +43 732 793 880  
KUKAAustriaOffice@kuka.com  
www.kuka.at

**波兰**

KUKA CEE GmbH Poland  
Spółka z ograniczoną odpowiedzialnością  
Oddział w Polsce  
Ul. Porcelanowa 10  
40-246 Katowice  
波兰  
电话 +48 327 30 32 13 or -14  
传真 +48 327 30 32 26  
ServicePL@kuka.com

**葡萄牙**

KUKA Robots IBÉRICA, S.A.  
Rua do Alto da Guerra n° 50  
Armazém 04  
2910 011 Setúbal  
葡萄牙  
电话 +351 265 729 780  
传真 +351 265 729 782  
info.portugal@kukapt.com  
www.kuka.com

**俄罗斯**

KUKA Russia 000  
1-y Nagatinskiy pr-d, 2  
117105 Moskau  
俄罗斯  
电话 +7 495 665-6241  
support.robotics.ru@kuka.com

**瑞典**

KUKA Svetsanläggningar + Robotar AB  
A. Odhners gata 15  
421 30 Västra Frölunda  
瑞典  
电话 +46 31 7266-200  
传真 +46 31 7266-201  
info@kuka.se

**瑞士**

KUKA Roboter CEE GmbH  
Steyregg, Zweigniederlassung Buchs  
Heinrich Wehrli-Strasse 27  
5033 Buchs  
瑞士  
电话 +41 62 837 43 20  
info.robotics.ch@kuka.com

**斯洛伐克**

KUKA CEE GmbH  
organizačná zložka  
Bojnická 3  
831 04 Bratislava  
斯洛伐克  
电话 +420 226 212 273  
support.robotics.cz@kuka.com

**西班牙**

KUKA Iberia, S.A.U.  
Pol. Industrial  
Torrent de la Pastera  
Carrer del Bages s/n  
08800 Vilanova i la Geltrú (Barcelona)  
西班牙  
电话 +34 93 8142-353  
comercial@kukarob.es

**南非**

Jendamark Automation LTD (Agentur)  
76a York Road  
North End  
6000 Port Elizabeth  
南非  
电话 +27 41 391 4700  
传真 +27 41 373 3869  
www.jendamark.co.za



**台湾**

KUKA Automation Taiwan Co. Ltd.  
1F, No. 298 Yangguang ST.,  
Nei Hu Dist., Taipei City, Taiwan 114  
台湾  
电话 +886 2 8978 1188  
传真 +886 2 8797 5118  
info@kuka.com.tw

**泰国**

KUKA (Thailand) Co. Ltd.  
No 22/11-12 H-Cape Biz Sector Onnut  
Sukhaphiban 2 road, Prawet  
Bangkok 10250  
泰国  
电话 +66 (0) 90-940-8950  
HelpdeskTH@kuka.com

**捷克**

KUKA Roboter CEE GmbH  
organizační složka  
Pražská 239  
25066 Zdiby  
捷克  
电话 +420 226 212 273  
support.robotics.cz@kuka.com

**土耳其**

KUKA CEE GmbH  
Şerifali Mah.  
Bayraktar Bulv. Beyit Sok. No:9-11  
34775 Ümraniye/Istanbul  
土耳其  
电话 +90 216 508 14 04  
hotline-tr@kuka.com; servis-tr@kuka.com  
www.kuka.com

**匈牙利**

KUKA HUNGÁRIA Kft.  
Fő út 140  
2335 Taksony  
匈牙利  
电话 +36 24 501609  
传真 +36 24 477031  
info@kuka-robotics.hu

## USA

KUKA Robotics Corporation  
51870 Shelby Parkway  
Shelby Township  
48315-1787  
Michigan  
USA  
电话 +1 866 873-5852  
传真 +1 866 329-5852  
CustomerService@kuka.com  
www.kuka.com

## 英国

KUKA Robotics UK Ltd  
Great Western Street  
Wednesbury West Midlands  
WS10 7LL  
英国  
电话 +44 121 505 9970  
传真 +44 121 505 6589  
service@kuka-robotics.co.uk  
www.kuka-robotics.co.uk

## 索引

## C

CAST\_FROM()..... 38, 48  
 CAST\_TO()..... 36, 48

## E

EKI..... 8  
 EKI\_CHECK()..... 41, 44, 111, 112  
 EKI\_CheckBuffer()..... 33, 113  
 EKI\_Clear()..... 40, 105, 112  
 EKI\_ClearBuffer()..... 40, 112  
 EKI\_Close()..... 36, 105  
 EKI\_GetBool()..... 108  
 EKI\_GetBoolArray()..... 108, 109  
 EKI\_GetFrame()..... 111  
 EKI\_GetFrameArray()..... 111  
 EKI\_GetInt()..... 109  
 EKI\_GetIntArray()..... 109  
 EKI\_GetReal()..... 110  
 EKI\_GetRealArray()..... 110  
 EKI\_GetString()..... 38, 110, 111  
 EKI\_Init()..... 33  
 EKI\_Lock()..... 113  
 EKI\_Open()..... 35, 36, 66, 68, 105  
 EKI\_Send()..... 36, 79 - 81, 84, 106  
 EKI\_SetBool()..... 106, 107  
 EKI\_SetFrame()..... 107  
 EKI\_SetInt()..... 107  
 EKI\_SetReal()..... 107, 108  
 EKI\_SetString()..... 108  
 EKI\_STATUS..... 41  
 EKI\_Unlock()..... 113  
 EOS..... 8  
 EthernetKRL\_Server.exe..... 45

## F

FIFO..... 8, 12, 13

## I

IP..... 8

## K

KLI..... 8, 23  
 KR C..... 8  
 KRL..... 8  
 KUKA 服务..... 115  
 KUKA 客户支持系统..... 115

## L

LIFO..... 8, 12, 13, 43

## P

Ping..... 11

## S

smartHMI..... 8  
 smartPAD..... 9

## T

TCP..... 9

## U

UDP..... 9

## X

XML..... 9  
 XPath..... 9, 29, 31

## 安

安全..... 17  
 安全提示..... 7  
 安装..... 19  
     通过 smartHMI..... 20  
     通过 WorkVisual..... 19

## 按

按规定使用..... 15

## 编

编程..... 25  
 编程提示..... 33

## 拆

拆卸..... 21  
     通过 smartHMI..... 21

## 产

产品说明..... 11

## 传

传输协议..... 11

## 错

错误处理..... 14  
 错误反应, 编程..... 44  
 错误记录..... 59

## 服

服务器程序.....	45
服务器程序，操作界面.....	46
服务器程序，通信参数.....	47
服务器运行模式.....	14, 35

## 附

附录.....	103
---------	-----

## 概

概念，所用的.....	8
-------------	---

## 更

更新，通过 smarthMI.....	20
更新，通过 WorkVisual.....	19

## 功

功能.....	11
---------	----

## 函

函数，数据交换.....	32
--------------	----

## 记

记录类型.....	14
-----------	----

## 技

技术支持咨询.....	115
-------------	-----

## 监

监控，连接.....	11
------------	----

## 开

开源.....	9
---------	---

## 客

客户端运行模式.....	14, 36
--------------	--------

## 连

连接，监控.....	11
连接配置.....	11

## 末

末尾字符串.....	8
------------	---

## 目

目标群体.....	7
-----------	---

## 内

内存，增加.....	103
------------	-----

## 培

培训.....	7
---------	---

## 配

配置.....	23
配置，以太网连接.....	25

## 日

日志.....	59
---------	----

## 商

商标.....	9
---------	---

## 失

失去连接.....	11
-----------	----

## 时

时间戳.....	42
----------	----

## 示

示例，接入.....	45
示例程序.....	45
示例配置.....	45

## 事

事件信息.....	14, 42
-----------	--------

## 数

数据保存.....	12
数据交换.....	12
数据交换，函数.....	32
数据流.....	8

## 碎

碎片化.....	43
碎片整理.....	43

## 所

所用概念.....	8
-----------	---

**套**  
套接字..... 9

知识，必需..... 7

**提**  
提示..... 7

**指**  
指令参考..... 105

**网**  
网络连接..... 23

**违**  
违规使用..... 15

**文**  
文献，工业机器人..... 7

**系**  
系统要求  
    软件..... 19  
    硬件..... 19

**卸**  
卸载  
    通过 WorkVisual..... 20

**信**  
信息..... 59  
信息，禁用..... 104

**许**  
许可证..... 9

**以**  
以太网..... 8  
以太网，接口..... 23  
以太网连接，配置..... 25

**引**  
引言..... 7

**诊**  
诊断..... 57  
诊断数据，显示..... 57  
诊断显示器（菜单项）..... 57

**知**