

应用手册 Integrated Vision

Trace back information:
Workspace R15-2 version a20
Checked in 2015-10-22
Skribenta version 4.6.176

应用手册
Integrated Vision

RobotWare 6.02

文档编号: 3HAC044251-010

修订: E

本手册中包含的信息如有变更，恕不另行通知，且不应视为 ABB 的承诺。ABB 对本手册中可能出现的错误概不负责。

除本手册中有明确陈述之外，本手册中的任何内容不应解释为 ABB 对个人损失、财产损失或具体适用性等做出的任何担保或保证。

ABB 对因使用本手册及其中所述产品而引起的意外或间接伤害概不负责。

未经 ABB 的书面许可，不得再生或复制本手册和其中的任何部件。

可从 ABB 处获取此手册的额外复印件。

本出版物的原始语言为英语。所有其他语言版本均翻译自英语版本。

© 版权所有 2013-2015 ABB。保留所有权利。

ABB AB
Robotics Products
Se-721 68 Västerås
瑞典

目表

手册概述	7
产品文档, IRC5	9
安全	11
1 Integrated Vision 简介	13
1.1 系统概述	13
1.2 图像安全	14
1.3 Integrated Vision 入门	15
1.4 术语表	16
2 安装	17
2.1 安装硬件	17
2.2 安装软件	19
3 RobotStudio 用户界面	21
3.1 主窗口	21
3.2 在线帮助	22
3.3 菜单条	23
3.4 控制器浏览器	26
3.5 图像采集和配置区域	27
3.6 胶卷	29
3.7 调色板窗口	30
3.8 上下文窗口	31
3.9 Options (选项) 对话框	32
3.10 电子表格视图	33
4 FlexPendant 用户界面	37
4.1 RobotWare Integrated Vision	37
4.2 操作员视图	38
5 配置 Integrated Vision	41
5.1 建议的工作步骤	41
5.2 准备工作	42
5.3 设置摄像头	43
5.3.1 基本步骤	43
5.3.2 其它摄像头配置	45
5.3.3 限制用户访问	49
5.4 设置新图像作业	53
5.5 设置图像	54
5.6 校准	55
5.7 添加图像工具	58
5.8 输出到 RAPID	60
5.9 I/O 处理	64
5.10 准备 RAPID 程序	65
5.10.1 RobotStudio 中的 RAPID 代码片段	65
5.10.2 基本编程示例	66
5.10.3 高级编程示例	68
5.11 开始生产	70
6 参考信息	71
6.1 坐标系统之间的关系	71
6.2 校准理论	74
6.3 最佳做法	76
6.3.1 在采用解决方案前先评估性能	76
6.3.2 如何安装摄像头	77

6.3.3	获取精度	78
6.3.4	获取良好的照明	79
6.3.5	图像作业的结构组织	80
6.3.6	启动程序	81
6.3.7	运行时间内启用和禁用图像工具	82
6.3.8	避免超出摄像头的存储空间	84
6.3.9	备份摄像头到控制器	85
6.3.10	将不同类型的项目分类	86
6.3.11	寻找同一类型的多个项目	88
6.3.12	总是检查确认图像目标在预期限制内	90
7	RAPID 参考信息	91
7.1	指令 :	91
7.1.1	CamFlush - 从摄像头删除集合数据	91
7.1.2	CamGetParameter - 获取不同名称的摄像头参数	92
7.1.3	CamGetResult - 从集合获取摄像头目标	94
7.1.4	CamLoadJob - 加载摄像头任务到摄像头	96
7.1.5	CamReqlmage - 命令摄像头采集图像	98
7.1.6	CamSetExposure - 设置具体摄像头的的数据	100
7.1.7	CamSetParameter - 设置不同名称的摄像头参数	102
7.1.8	CamSetProgramMode - 命令摄像头进入编程模式	104
7.1.9	CamSetRunMode - 命令摄像头进入运行模式	105
7.1.10	CamStartLoadJob - 开始加载摄像头任务到摄像头	106
7.1.11	CamWaitLoadJob - 等待摄像头任务加载完毕	108
7.2	函数	109
7.2.1	CamGetExposure - 获取具体摄像头的的数据	109
7.2.2	CamGetLoadedJob - 获取所加载摄像头任务的名称	111
7.2.3	CamGetName - 获取所使用摄像头的名称	112
7.2.4	CamNumberOfResults - 获取可用结果的数量	113
7.3	数据类型	115
7.3.1	cameradev - 摄像头设备	115
7.3.2	cameratarget - 摄像头数据	116
	索引	119

手册概述

关于本手册

本手册包含选件 Integrated Vision 安装、配置和日常操作的说明。

手册用法

在安装、配置和维护带有 Integrated Vision 选件的系统时，应使用本手册。

本手册的阅读对象

本手册面向：

- 安装人员
- 程序员
- 操作员

操作前提

使用 ABB 机器人的维护/维修/安装技工必须经过 ABB 的培训，且具备机械和电气安装/维护/维修工作所需的知识。

参考信息

参考文档	文档编号
操作员手册 - RobotStudio	3HAC032104-010
操作员手册 - 带 FlexPendant 的 IRC5	3HAC050941-010
操作员手册 - IRC5 故障排除	3HAC020738-010
技术参考手册 - RAPID指令、函数和数据类型	3HAC050917-010
技术参考手册 - RAPID语言概览	3HAC050947-010
技术参考手册 - 系统参数	3HAC050948-010
Product manual - IRC5	3HAC021313-001
产品手册 - IRC5 Compact	3HAC035738-010
Product manual - IRC5 Panel Mounted Controller (IRC5 及主计算机 DSQC 639)	3HAC027707-001
Product manual - IRC5	3HAC047136-001
产品手册 - IRC5 Compact	3HAC047138-010
Product manual - IRC5 Panel Mounted Controller (IRC5 及主计算机 DSQC1000)	3HAC047137-001
产品手册 - IRB 14000	3HAC052983-010

外部参考

在用户文档 DVD 中包含了下列外部手册：

参考文档	文档编号
Cognex In-Sight® 7000 Series Vision System Installation Manual 本手册介绍了如何安装摄像头。	P/N 597-0138-01

下一页继续

修订版

版本号	描述
-	随 RobotWare 5.15.01 发布。 第一版
A	与 RobotWare 5.15.03 和 RobotWare 5.60 一起发行。 除了别的以外，整个手册中更新和校正了以下内容： <ul style="list-style-type: none"> • 更新了限值列表，请参阅第13页的Integrated Vision 简介。 • “安装”一章进行了更新和结构重组，请参阅第17页的安装。 • 增加了选项对话框，请参阅第32页的Options (选项) 对话框。 • 更改了图像参数的语言设置，请参阅第33页的更改语言。 • 更新和增加了有关移除摄像头、更新摄像头固件和连接摄像头模拟器的信息，请参阅第43页的设置摄像头。 • “最佳做法”一章中新增了几节，请参阅第76页的最佳做法。
B	随 RobotWare 5.61 发布。 除了别的以外，整个手册中更新和校正了以下内容： <ul style="list-style-type: none"> • 在多个 RAPID 说明中增加了错误恢复 <code>ERR_CAM_COM_TIMEOUT</code>。
C	随 RobotWare 6.0 发布。 <ul style="list-style-type: none"> • 限制用户访问的新功能，请参阅第49页的限制用户访问。 • 预先印制的校准盘现在上市，请参阅第55页的摄像头校准。 • “最佳做法”一章中新增了一节，请参阅第90页的总是检查确认图像目标在预期限制内。 • 细微纠正。
D	随 RobotWare 6.01 发布。 <ul style="list-style-type: none"> • 更新了对其他手册的参考，请参阅第7页的参考信息。 • 更新了必要硬件和软件的列表，请参阅第15页的检查表。 • 更新了安装操作程序，请参阅第17页的安装硬件。 • 细微纠正。
E	随 RobotWare 6.02 一同发布。 <ul style="list-style-type: none"> • 更新了有关第54页的图像触发事件的信息。 • 添加了 <code>\AwaitComplete</code> 变元到 RAPID 指令第98页的CamReqlmage - 命令摄像头采集图像。 • RAPID 指令、函数与数据类型现在也可以在技术参考手册 - RAPID 指令、函数和数据类型 中使用。

产品文档, IRC5

ABB 机器人用户我能当的类别

ABB 机器人用户文档分为多个类别。以下列表基于文档的信息类型编制, 而未考虑产品为标准型还是选购型。

所有列出的文档都可以向 ABB 订购 DVD。列出的文档适用于 IRC5 机器人系统。

产品手册

机械手、控制器、DressPack/SpotPack 和其他大多数硬件交付时一般都附有包含以下内容的产品手册：

- 安全信息。
- 安装与调试（介绍机械安装或电气连接）。
- 维护（介绍所有必要的预防性维护程序, 包括间隔周期和部件的预计使用寿命）。
- 维修（介绍所有建议的维修程序, 包括零部件）。
- 校准。
- 停用。
- 参考信息（安全标准、单位换算、螺钉接头和工具列表）。
- 带分解图的备件列表（或者单个备件列表的参考信息）。
- 电路图（或电路图参考信息）。

技术参考手册

技术参考手册介绍了机器人产品的参考信息。

- *Technical reference manual - Lubrication in gearboxes* : 关于操纵器齿轮箱的润滑油类型和用量的说明。
- 技术参考手册 - *RAPID*语言概览 : *RAPID* 编程语言概述。
- 技术参考手册 - *RAPID*指令、函数和数据类型 : 所有 *RAPID* 指令、函数和数据类型的说明和语法。
- 技术参考手册 - *RAPID*语言内核 : *RAPID* 编程语言的正式描述。
- 技术参考手册 - 系统参数 : 系统参数和配置工作流程说明。

应用手册

特定的应用产品（例如软件或硬件选项）在应用手册中介绍。一本应用手册可能涵盖一个或多个应用产品。

应用手册通常包含以下信息：

- 应用产品用途（作用及使用场合）。
- 随附的内容（如电缆、I/O 电路板、*RAPID* 指令、系统参数和计算机软件光盘）。
- 如何安装所包含的或所需的硬件。
- 如何使用应用产品。
- 应用产品使用示例。

下一页继续

操作员手册

操作员手册说明了产品的实际处理过程。手册面向与产品有着直接接触操作的人员,即生产车间操作员、程序员和故障排除人员。

此套手册包括 (特别是) :

- 操作员手册 - 紧急安全信息
- 操作员手册 - 一般安全信息
- 操作员手册 - 使用入门、IRC5 和 RobotStudio
- *Operating manual - Introduction to RAPID*
- 操作员手册 - 带 FlexPendant 的 IRC5
- 操作员手册 - RobotStudio
- 操作员手册 - IRC5 故障排除, 用于控制器和操纵器。

安全

人员安全

在机器人控制器内部工作时，必须注意电相关的风险。

以下部件有高压危险：

- 控制器内部设备，例如 I/O 设备之类的设备，可从外部电源供电。
- 主电源/主开关。
- 电源单元。
- 计算机系统的电源装置 (230 VAC)。
- 整流器装置 (400-480 VAC 和 700 VDC) 。电容器！
- 驱动装置 (700 VDC)。
- 维修插座 (115/230 VAC)。
- 机械加工过程中的工具电源装置或特殊电源装置。
- 即使机器人已断开与主电源的连接，控制器连接的外部电压仍存在。
- 附加连接。

因此，在进行机械或电气安装工作前，请确保已经遵循了所有安全守则。

安全守则

在开始机械和/或电气安装前，请确保已经熟悉了操作员手册 - 一般安全信息¹中所述的安全规定。

¹ 本手册包含操纵器与控制器的产品手册中所含的全部安全说明。

此页刻意留白

1 Integrated Vision 简介

1.1 系统概述

什么是 Integrated Vision ?

ABB 的 Integrated Vision 系统插件提供了可靠且易用的图像系统，可以满足图像引导机器人 (VGR) 应用的一般需求。

系统包括一套完整的软硬件解决方案与 IRC5 机器人控制器以及 RobotStudio 编程环境完全集成。图像功能基于 *Cognex In-Sight®* 智能摄像头家族，配有嵌入式图形处理功能和以太网通信接口。

RobotStudio 配备图形编程环境，可调用 *Cognex EasyBuilder®* 功能的全调色板，同时具备部件位置、部件检查和识别的可靠工具。RAPID 编程语言已经添加了摄像头操作和图像引导专用指令和错误跟踪功能。

Hardware

摄像头系统基于 *Cognex In-Sight® 7000 series*，但可以使用任何 *Cognex In-Sight®* 摄像头。摄像头从控制器接入 24 VDC 和以太网。

摄像头连接到附带的以太网交换机。摄像头最多为三个。

有关更多信息，请参阅 *Cognex In-Sight® 7000 Series Vision System Installation Manual*。

软件

软件解决方案基于 3 个组件 – RobotStudio、使用 RAPID 编程语言的 IRC5 控制器以及 FlexPendant。

RobotStudio 并排显示图像和配置参数，提供了便利的 VGR 编程环境。

IRC5 控制器可以实现轻松创建 RAPID 程序，充分利用摄像头系统的功能。除了其他功能外，控制器还有一个 RAPID 接口，包括预建的摄像头通信接口和图像目标队列处理功能。

FlexPendant 配备了操作员界面，可以在生产中实现对系统的管理。

限制

- 需要一个 FlexPendant（类型为 SxTPU3）来运行 Integrated Vision。
- Integrated Vision 插件不能在 64 位版本的 RobotStudio 上运行。
- 集成摄像头仅应在机器人控制器的服务端口网络内使用，并且必须使用 RobotStudio Integrated Vision 插件配置。
- 用 RobotStudio 创建的图像程序包含专用配置数据。用 *Cognex EasyBuilder®* 制作的现有图像程序必须用 Integrated Vision 插件进行修改，以便与 IRC5 控制器兼容。

任何其他用途都未经过验证，可能会导致无法预料的行为。

1 Integrated Vision 简介

1.2 图像安全

1.2 图像安全

一般原则

使用机器人引导图像感应器需要用户在处理、安装、配置系统时特别小心。
用户必须始终确保图像传感器打开，即使操纵器并未移动。
在进入操纵器的工作范围前，用户必须采取以下预防措施以防操纵器开始移动。



警告

如果必须在操纵器工作范围内执行工作，请务必遵守以下几点：

- 控制器上的操作模式选择器必须处于手动模式位置，以保证启用装置正常工作并阻止来自图像传感器、计算机链路或远程控制面板的操作。
- 任何人进入操纵器工作空间时必须始终随身携带 FlexPendant。这是为了防止其他人在其不知情的情况下控制操纵器。
- 任何时候都必须保证使动装置可以正常工作。
- 在编程和测试过程中，操纵器无需移动时必须尽快释放使动装置。

1.3 Integrated Vision 入门

检查表

在设置系统前，请确保完成了必要的准备措施。

Hardware

- 为 IRB 机器人系统配齐操纵器、IRC5 控制器和一个 FlexPendant（类型为 SxTPU3）。
- 选件 *Integrated Vision interface*，其中包括：
 - Ethernet 网关。
 - 用于将摄像头连接到网关的 Ethernet 网线。
 - 用于将网关连接到 IRC5 主计算机服务端口的 Ethernet 网线。
 - 用于将网关连接到机柜服务端口接口的 Ethernet 网线。
 - 客户电源。
 - 从客户电源向 Ethernet 网关提供 24 VDC 的电线。
 - 为摄像头提供 24 VDC 的电线。
- 用于安装在机器人上以实现精确定义工件的指点工具。
- PC。
- 用于将 PC 连接到控制器的 Ethernet 电缆。
- *Cognex In-Sight®* 摄像头。
- 摄像头镜头。
- 照明设备。

软件

- RobotStudio 5.60 或更高版本，完整安装。不需要 RobotStudio 许可。
- RobotWare 5.60 或更高版本。
- RobotWare 许可，启用了 *Integrated Vision* 选件。

有关如何配置 Integrated Vision 的信息，请参阅第41页的配置 [Integrated Vision](#)。

1 Integrated Vision 简介

1.4 术语表

1.4 术语表

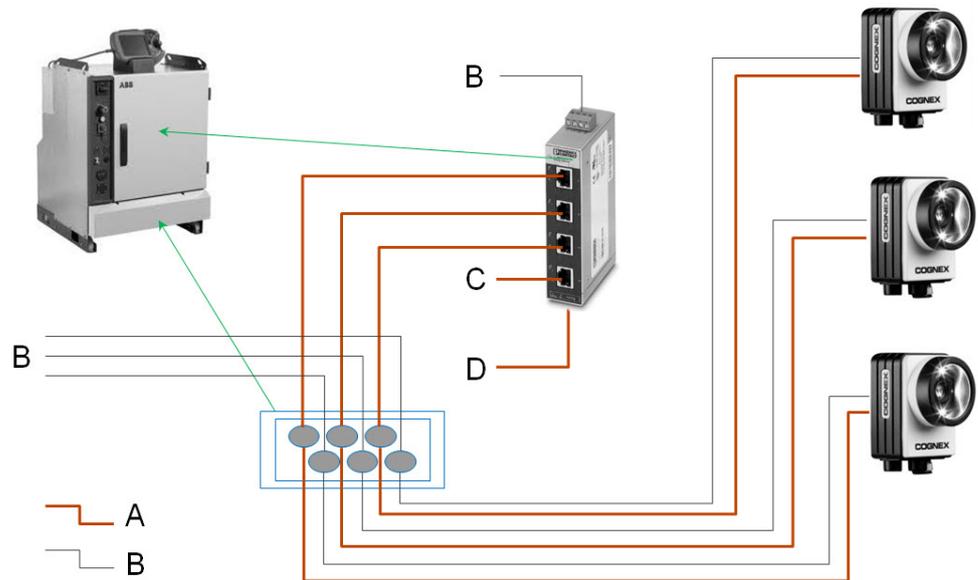
术语列表

术语	说明
摄像头校准	计算图像中的像素坐标与空间中实际坐标的转换。校准后的帧通常使用一个国际象棋盘校准板定义。
摄像头到机器人校准	校准后摄像头帧、摄像头校准计算的帧以及机器人基础帧之间的关系。结果通常存储为一个工件用户帧。
国际象棋盘校验板	校准摄像头要使用校准图样。图样在校准期间放置在摄像头视野范围内。
<i>Cognex EasyBuilder®</i>	<i>Cognex</i> 的摄像头软件已经无缝集成到了 <i>RobotStudio</i> ，以提供 <i>Integrated Vision</i> 。
<i>Cognex In-Sight®</i>	<i>Integrated Vision</i> 支持的特定产品系列的智能摄像头。
基准	基准标记或基准就是放在摄像头校准板图像上，用于作为原始坐标参照点的物体。
指点工具	具有精确定义的工具中心点 (TCP) 的机器人工具，用于向机器人准确指引目标。
智能摄像头	具有嵌入式图像处理能力的摄像头。
代码片段	代码片段是指您可以插入 <i>RAPID</i> 程序中的预定义的 <i>RAPID</i> 代码段。
VGR	图像引导机器人
图像校准	将摄像头校准和摄像头到计算机的校准结合在一起，其结果是一个常用的图像帧，可以让机器人准确的移动到图像目标。
图像作业	图像作业或作业是加载到摄像头的图像程序。

2 安装

2.1 安装硬件

概述



xx120000992

A	以太网
B	从客户电源接入到网关和摄像头的 24V 电源
C	网关和控制柜服务端口（内部）之间的以太网连接
D	网关和主计算机服务端口之间的以太网连接

安装程序（步骤）

如图连接下列部件和线缆。

	操作
1	确保控制器电源开关已经关闭。
2	将 Ethernet 电缆从控制器机柜服务端口（内部）连接到交换机上四个 Ethernet 接口中的一个。
3	通过控制器机柜上的电缆密封套将 Ethernet 电缆从每个摄像头连接到交换机上的任何可用的 Ethernet 接口。 小心地剥掉 20 毫米绝缘，将电缆捆扎在电缆密封套上的接地片上。
4	通过控制器机柜上的电缆密封套将 24 VDC 电源电缆从每个摄像头连接到 24 VDC 电源。 小心地剥掉 20 毫米绝缘，将电缆捆扎在电缆密封套上的接地片上。

下一页继续

2 安装

2.1 安装硬件

续前页

有关详细信息，请参阅 *Cognex In-Sight® 7000 Series* 手册、对应控制器的产品手册和电路图，请参阅 [第7页的参考信息](#)。



小心

使用机器人携带的或采用其他方式安装的移动摄像头时，沿机器人手臂合理布线非常重要。

布设电缆时必须小心谨慎，避免连接器承受机械应力，让电缆有足够的弯曲半径，尽量减小电缆磨损。此外还建议在固定点处和特别暴露的区域在电缆上额外加装防磨损保护。

2.2 安装软件

安装 RobotStudio

Integrated Vision 配置环境是以 RobotStudio 插件的形式设计的，包含在标准安装中。

操作	
1	安装 RobotStudio。选择完全安装。
2	启动 RobotStudio。
3	<p>转到菜单条上的控制器选项卡并启动 Integrated Vision 插件。在插件加载完成后，会显示一个新的选项卡图像 (Vision)。</p> <p> 提示</p> <p>连接了一个控制器时，可以从控制器浏览器中的控制器节点、图形系统节点或摄像头节点的右键菜单启动 Integrated Vision 插件。</p>

有关更多信息，请参阅 操作员手册 - *RobotStudio*。



注意

建议在开始配置 Integrated Vision 系统后，不要更改 RobotStudio 的语言。
有关详细信息，请参阅[第33页的更改语言](#)。



提示

RobotStudio 库中包含摄像头的 3D 模型。

将 RobotWare 安装到 IRC5 控制器

要运行 Integrated Vision，需要有一个 RobotWare 授权许可密钥（启用了 Integrated Vision 选项）。

使用 RobotStudio 来配置、构建和下载 RobotWare 系统到 IRC5 控制器。

有关更多信息，请参阅 操作员手册 - *RobotStudio*。

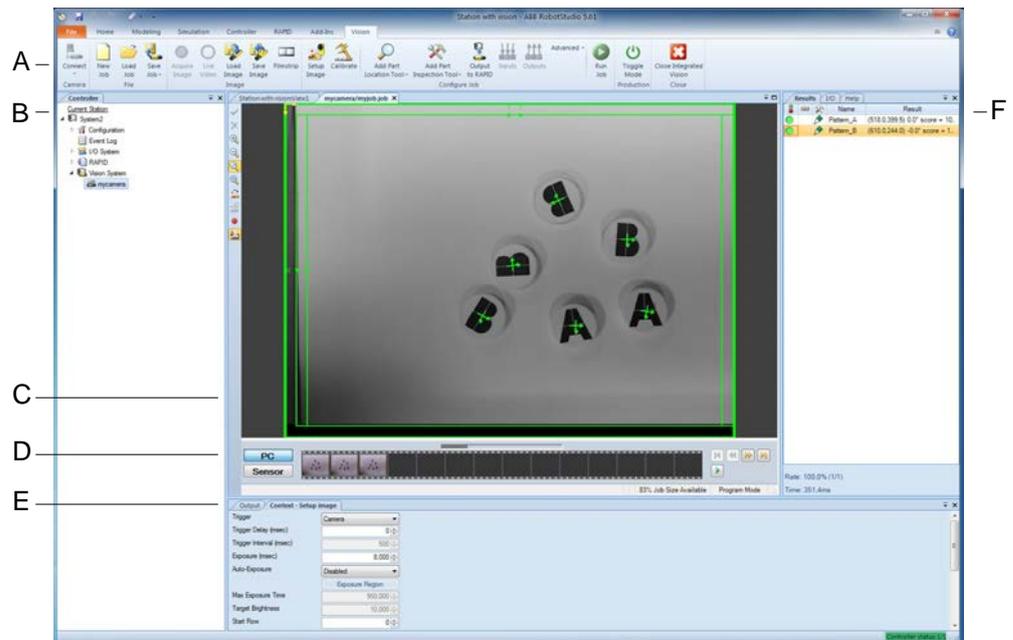
此页刻意留白

3 RobotStudio 用户界面

3.1 主窗口

主窗口概述

本节介绍了 Integrated Vision 图形用户界面概况。



xx120000989

	部件说明	描述
A	菜单条	根据不同的功能类型，分组显示图标。
B	控制器浏览器	图像系统节点会显示网络中的所有摄像头。
C	图像采集和配置区域	显示由摄像头获取到的图像，附带有定位和检查部件的说明。
D	胶卷工具条	用于记录一系列图像以便以后分析。
E	上下文窗口	包含所选控件的可用属性、设置以及事件。
F	调色板窗口	可用选项卡如下： <ul style="list-style-type: none"> • Results (结果) 选项卡 - 显示活动图像作业的设置以及所有已使用位置和检查工具的列表。 • I/O 选项卡 - 显示 I/O 设置。 • 帮助 选项卡 - 提供在线帮助。



提示

如果意外关闭了任一窗口，可以从自定义快速访问工具栏恢复该窗口。

3 RobotStudio 用户界面

3.2 在线帮助

3.2 在线帮助

概述

有经验的摄像头用户熟悉 *Cognex* 产品线，应该会认识大部分 RobotStudio 中有关图像的用户图形界面。

Cognex EasyBuilder® 是 *Cognex* 提供的摄像头软件，无缝集成到 RobotStudio，创建出 IRC5 的 Integrated Vision 系统。为提供图形引导机器人的最佳性能，用户图形界面仅经过轻微修改。

这与在线帮助选项卡有关，因为在线帮助的内容是未经修改直接集成的。

所有参数、设置和技术解释的参考都是正确的。但是任何涉及到 GUI 的参考可能会有轻微的不准确，具体情况取决于 RobotStudio 的版本。

因此可以使用在线 Help（帮助）选项卡作为所有参数和设置的技术参考手册，使用本应用软件手册作为用户图形界面的参考。

在线帮助选项卡

在线帮助选项卡位于调色板窗口。

在线帮助选项卡是一个针对所有 Integrated Vision 参数和设置的上下文敏感的参考手册。也就是说帮助选项卡的内容会根据当前在 Integrated Vision 中的操作自动调整。例如设置图像、校准、添加位置工具等。



注意

大部分参数和设置只在帮助选项卡中介绍，并未包含在本手册中。

术语

下面的列表介绍了在线帮助选项卡加与 Integrated Vision 应用程序中术语部分最主要的区别。

如果在线帮助选项卡说...	...则表示：
<i>EasyBuilder®</i>	机器人图像插件(RobotStudio vision add-in)选项卡
编辑采集设置(edit acquisition settings) (群框)	设置图像(Setup Image)按钮
按实际部件校准图像(calibrating the image to real world units) (组框)	校准(Calibrate)按钮
在线	运行模式
离线	编程模式

3.3 菜单条

菜单条布局

Integrated Vision选项卡包括按照功能的逻辑关系组织的几组命令，以便简化Integrated Vision项目的管理。

选项卡包含以下各组：

组	用于以下对象的功能：
摄像头	连接到摄像头和设置摄像头属性
File (文件)	创建、保存和加载图像作业。
图像	加载和保存图像。图像可以从摄像头获取，也可以从此前保存的文件中加载。
配置作业	配置图像作业的工作流，按照功能的逻辑关系组织。
生产	设置Integrated Vision系统为编程模式或运行模式。（类似于控制面板上机器人控制器的自动/手动切换开关。）

摄像头组

摄像头组用于连接到摄像头和配置摄像头。

Button	描述
连接	连接到所选的摄像头。
断开	断开所选的摄像头。  注意 当图像采集和配置区中的相应选项卡关闭时，摄像头也会断开。
重命名	重命名控制器以便使摄像头在 RAPID 中可以通过一个名称调用。
网络设置	更改已连接摄像头的 IP 设置。
设置日期和时间	更改摄像头的日期和时间。
用户访问设置	编辑摄像头上的用户列表及其相关权限。
设置控制器用户	选择在与摄像头通信时由控制器使用的用户档案。
添加传感器	在网络上找到摄像头并更改其 IP 设置。
Properties (属性)	查看摄像头属性。

从摄像头节点右键菜单也可以访问同样的设置，请参阅[第26页的摄像头节点右键菜单](#)。

文件组

文件组用于加载和保存作业。

Button	描述
新作业	创建新的图像作业。
加载作业	从文件加载一个图像作业到摄像头。
保存作业	保存图像作业到文件。
将作业另存为	

下一页继续

3 RobotStudio 用户界面

3.3 菜单条 续前页

有关作业以及作业保存位置的更多信息，请参阅[第53页的设置新图像作业](#)。

图像组

图像组用于处理图像。

Button	描述
获取图像	获取新图像。
实时视频	开/关摄像头实时视频模式。
加载图像	将图像从文件加载到摄像头。
保存图像	将当前图像保存到文件。
显示胶卷	编辑录制/播放设置

配置作业组

配置作业组用于对当前图像作业进行必要设置。例如摄像头应当定位的对象、如何检查对象以及最后图像数据应如何传输到 RAPID 程序。

Button	描述
设置图像	修改图像获取参数。
校准	按实际部件校准图像
添加部件位置工具	添加位置部件工具到图像作业。
添加部件检查工具	添加检查工具到图像作业。
输出到 RAPID	选择要在 RAPID 中使用的图像作业结果。
输入	定义摄像头输入。
输出	定义摄像头输出。
Advanced (高级)	高级设置和编辑模式。
运行作业	运行图像作业。

有关配置Integrated Vision的更多信息，请参阅[第41页的配置 Integrated Vision](#)。

生产组

生产组中的运行模式按钮用于手动在编程模式和运行模式间切换摄像头。

当在生产中运行图像系统时，编程模式和运行模式的切换是在 RAPID 程序中进行的。

Button	描述
运行模式	在编程模式（按钮不可用）和运行模式（按钮可用）之间切换

有关从 RAPID 程序设置运行模式的更多信息，请参阅[第65页的准备 RAPID 程序](#)。

高级设置

以下功能和设置可以在高级按钮下找到。

Button	描述
查看电子表格	高级模式编辑。 请参阅 第33页的电子表格视图 一节。
不保护作业 / 保护作业	保护电子表格中锁定的单元格，防止被修改。 请参阅 第33页的电子表格视图 一节。

下一页继续

Button	描述
显示图像设置	配置实时和在线图像的分辨率和帧速 请参阅第25页的显示图像设置对话框一节。
作业大小	显示当前作业的大小 请参阅第25页的作业大小对话框一节。
更新固件	更新摄像头固件。 请参阅第46页的更新摄像头固件一节。

从摄像头节点右键菜单也可以访问同样的设置，请参阅第26页的摄像头节点右键菜单。

显示图像设置对话框

显示图像设置对话框用于配置实时和在线图像的分辨率和帧速。

设置	描述
实时采集 • 解析度	实时图像的全分辨率、半分辨率和四分之一分辨率。
实时采集 • 限制最大帧速	启用最大速率 (帧/秒) 文本框。
实时采集 • 最大速率 (帧/秒)	指定允许传感器每秒可以发送的图像的最大数量(0.016 to 100)。此功能可以用于通过限制发送的图像数量来减少网络通信量。
在线 (运行模式) • 解析度	实时图像的全分辨率、半分辨率和四分之一分辨率。 已经优化，在摄像头或连续模式下提供半分辨率，在手动模式下提供全分辨率。
在线 (运行模式) • 限制最大帧速	启用最大速率 (帧/秒) 文本框。
在线 (运行模式) • 最大速率 (帧/秒)	指定允许传感器每秒可以发送的图像的最大数量(0.016 to 100)。此功能可以用于通过限制发送的图像数量来减少网络通信量。

作业大小对话框

作业大小对话框用于在可供图像系统使用的内存中配置分配给作业的内存大小。滑块控件可以控制图像系统的内存分配。调整滑块设置作业的内存大小分配。

调整图像系统的作业大小限制要求图像系统重启，并且会从内部闪存清除所有作业和设置文件。请在继续前保存或备份作业。

3 RobotStudio 用户界面

3.4 控制器浏览器

3.4 控制器浏览器

控制器浏览器的布局

控制器浏览器采用分级结构视图，显示 RobotStudio 控制器选项卡中的控制器和配置元素。

有关控制器选项卡视图总体内容和功能的详细介绍，请参阅操作员手册 - *RobotStudio*。

对于 Integrated Vision 提供了一个单独的图像系统节点。在此节点下，所有的摄像头都连接到显示的机器人上。命名的摄像头按其名称识别，未命名的摄像头按其 MAC 地址识别。



提示

连接了一个控制器时，可以从控制器浏览器中的控制器节点、图形系统节点或摄像头节点的右键菜单启动 Integrated Vision 插件。

摄像头节点右键菜单

具体每个摄像头节点的右键菜单与菜单条上 Camera (摄像头) 组和功能区上的 Advanced (高级) 按钮的内容相同。

Button	描述
Integrated Vision	启动 Integrated Vision 插件。
连接	连接到所选的摄像头。
断开	断开所选的摄像头。  注意 当图像采集和配置区中的相应选项卡关闭时，摄像头也会断开。
重命名	重命名控制器以便使摄像头在 RAPID 中可以通过一个名称调用。
Restart (重新启动)	重启摄像头。
Properties (属性)	查看摄像头属性。
Advanced (高级)	与 Configure Job (配置作业) 组上 Advanced (高级) 按钮下的设置相同，请参阅 第24页的高级设置 。

3.5 图像采集和配置区域

图像采集和配置区域布局

图像采集和配置区域用于显示应该由Integrated Vision处理的图像。图像可以由任何在线摄像头实时采集，也可以从文件加载。

根据情况不同，在此区域会显示用于定位和检查部件的不同图像配置向导、坐标系统和图像数据。

图像采集和配置区域的按钮

Button	描述
接受更改 (ENTER)	在有些对话框中用来接受更改，如修改校准区时。
取消更改 (ESC)	在有些对话框中用来取消更改，如修改校准区时。
放大	缩放按钮
缩小	
适应	
填满	
旋转图像	90 度旋转图像。
只显示所选工具的图像	只显示所选工具的图像。
胶卷	显示胶卷工具条。
录制	开始录制图像到 PC。

图像采集和配置区域的快捷键

功能	快捷键
缩放	SHIFT + 滚轮。
使用窗口缩放	CTRL + 鼠标左键并拖动鼠标。
平移	CTRL + SHIFT + 鼠标左键并拖动鼠标。

图像采集和配置区域的选项卡

每个摄像头都在图像采集和配置区域单独的选项卡中显示。选项卡的标题栏会显示摄像头以及当前作业的名称。



注意

关闭选项卡等于断开摄像头。

状态条

在图像采集和配置区域有一个状态条，显示以下信息：

Information	描述
(R, G, B)@(x, y)	语法：(红, 绿, 蓝)@(x 坐标, y 坐标) 此信息仅在光标在图像区域移动时可见。

下一页继续

3 RobotStudio 用户界面

3.5 图像采集和配置区域

续前页

Information	描述
% 作业可用大小	显示摄像头的内存状态。百分比值显示了剩余的可用内存。当此值接近 0 时，则内存耗尽。
编程模式 / 运行模式	在菜单条显示与运行模式按钮相同的状态。

3.6 胶卷

简介

胶卷用于播放录制到 PC 的图像或查看存储到传感器的图像和结果。

此功能在故障排除和错误追查时会很有用，例如在生产中出现了部件定位和检查的间歇性错误时。

胶卷设置

检查菜单条上的大胶卷按钮打开显示和录制对话框设置窗口。

有 3 个主要设置组：传感器设置、录制设置和播放设置。

有关不同胶卷设置的更多信息，请参阅在线帮助选项卡的胶卷一节。

胶卷条

在图像采集和配置区域单击胶卷按钮以显示胶卷工具条。

通过的图像以绿色表示，失败的图像则以红色表示。

有关图像工具的通过与失败状态的更多信息，请参阅[第59页的图像工具的通过与失败](#)。

PC 胶卷

配置了胶卷设置后，按下录制按钮，则胶卷工具条将录制并显示通过和失败图像。图像可以从在线或离线摄像头保存到 PC 上的任何文件夹。

PC 播放模式可用于播放录制的图像，图像发送到摄像头，作业对正在被播放的图像执行。PC 上最多可保存 10000 个图像。

传感器胶卷

当传感器在线并采集图像时，传感器胶卷可用于监视作业的性能。

当采集到图像时，包括采集到的图像和相关的作业数据在内的作业结果保存在传感器的内存中。当结果保存到传感器时，会添加通过或失败图片到胶卷。

当胶卷中的某个结果被选中时，胶卷显示从图片变为缩微图，对应的图片会加载到显示区域。

使用显示和录制对话框窗口传感器设置选项卡的控件可配置传感器的行为。

保存数量的数量取决于图像系统的分辨率和可用内存，最多可保存 20 个结果。

3 RobotStudio 用户界面

3.7 调色板窗口

3.7 调色板窗口

结果选项卡

结果选项卡显示每个工具的性能，并可以在此进行故障排除或作业设置的优化。

结果选项卡可用于：

- 通过监视每个工具的信号快速确定工具通过（绿）或是失败（红）。
- 查看工具图标，以视觉方式辨别工具类型。
- 按名称识别工具。
- 双击工具进行修改。工具随后会显示在上下文窗口。按需要编辑参数或图像。

在结果选项卡右键单击工具也可以打开一个简单的编辑菜单，这个菜单可用于复制、粘贴、删除或编辑所选工具。

I/O 选项卡

I/O 选项卡显示每个作业的输入和输出线路的活动（绿）或不活动（灰）状态。这可以用于监视 I/O 线路。

帮助选项卡

请参阅[第22页的在线帮助](#)一节。

3.8 上下文窗口

简介

上下文窗口会自动更新，显示所选应用程序步骤的参数。上下文窗口旨在通过在左侧提供一般性参数以及在右侧显示更具体的参数，从而引导用户完成步骤。

此外，根据添加的工具，定位和检查部件步骤的窗口也可能显示可视帮助，例如反馈表或图形（以帮助配置应用程序参数）以及范围限制选项卡（用于设置通过/失败标准）。创建了作业后，可以按任何顺序再次访问各个步骤，所以可以修改和微调作业参数知道实现所需要的结果。

大部分步骤都要求按特定顺序配置参数。有关设计作业时每个步骤的说明，请参阅调色板窗口的帮助选项卡。

手册用法

下列功能使用上下文窗口进行设置和配置。

- 胶卷
- 设置图像
- 校准
- 添加部件位置工具
- 添加部件检查工具
- 输出到 RAPID
- 输入
- 输出

3 RobotStudio 用户界面

3.9 Options (选项) 对话框

3.9 Options (选项) 对话框

访问 Options (选项) 对话框

	操作
1	单击 File (文件) 选项卡。
2	单击 Options (选项) 。
3	滚动到 Integrated Vision 部分。

Integrated Vision 选项

主题	设置	描述
行为	Immediate Feedback (立即反馈)	如果勾选, 当添加和配置图像工具时, 将实时处理图像采集和配置区中的搜索区域。 如果未勾选 (默认), 当添加和配置图像工具时, 将在鼠标点击时处理图像采集和配置区中的搜索区域。
摄像头模拟器	Camera Model (摄像头型号)	决定摄像头模拟器的行为。当使用模拟器查看图像时, 图像被剪修到被模拟的传感器的尺寸, 因此需要选择要模拟的正确摄像头型号。 下拉式列表包括所有支持的 <i>Cognex In-Sight®</i> 图像系统, 以及那些传感器上可用的图像工具和功能。 默认摄像头型号为 is7200。 有关详细信息, 请参阅 第47页的将摄像头模拟器连接到虚拟控制器 。

3.10 电子表格视图

简介

Integrated Vision在添加图像工具或编辑作业时会将代码片段插入电子表格。通常这不会显示给用户。

电子表格视图是一种高级模式，显示已配置的作业以及电子表格模式包含的所有图像工具。这主要是为熟悉 *Cognex In-Sight Explorer®* 软件的高级用户设计的。



小心

电子表格使用不正确可能会导致在图像作业或 RAPID 程序中出现无法控制的错误。

概述

就标准操作和功能（例如操作表格块、编辑单元格、引用单元格和插入功能等）来说，电子表格类似于其他电子表格程序很相似。单元格被组织成 400 行（编号 0 到 399）26 列（标签为 A 到 Z），每个单元格按其列字母与行号来识别。例如单元格 A2 位于 A 列和第二行的交叉处。

电子表格每次配置一个单元格。每个单元格的内容定义为公式，插入单元格的任何信息（无论是单个数字还是负责的图像处理函数）都会被当成是公式的一部分。

电子表格相当于摄像头内存。这就是说当电子表格满了时，摄像头内存也满了。

有关使用电子表格的更多信息，请参阅 *Cognex In-Sight Explorer®* 帮助。



注意

电子表格最多可包含 4096 个活动单元格。如果插入的函数（例如 *FindBlobs*）会在电子表格中插入多个图像数据访问函数，则可能导致电子表格尝试超出限制。在这种情况下，不会显示警告对话框。超出 4096 限制的单元格不会被插入。

传输数据到 RAPID

使用输出到 RAPID 表，可以将任何随着每个图像采集更新的数据都可以从电子表格传输到 RAPID。要让数据在输出到 RAPID 表中可见，必须在保存数据的单元格添加符号标签。

标签的名称必须采用 `<Group>.<Result>` 格式，例如 `mydata.data`。

符号标签是从右键菜单添加的。

更改语言

当在 RobotStudio 中更改应用程序 GUI 语言时，Integrated Vision 插件的 GUI 语言也会随之改变。

集成的 *Cognex EasyBuilder®* 软件可用当前设置的 GUI 语言创建新的作业，并在电子表格的单元格中填充翻译的名称。这可造成当用另一 GUI 语言设置打开作业时，许多数据的名称将是首次创建该作业时设置的语言。任何后续添加均以当前设置的语言创建。

出于这个原因，强烈建议在编程和使用作业时使用相同的 GUI 语言。

下一页继续

3 RobotStudio 用户界面

3.10 电子表格视图

续前页

即使有些描述数据的标签和设置是用本地语言创建的，RobotStudio 被设置为创建具有英文名称（符号标签）的数据。原因是可以从 RAPID 访问的数据应不需要语言编码，例如当使用 `CamGetParameter` 时。



注意

建议在开始配置 Integrated Vision 系统后，不要更改 RobotStudio 的语言。

电子表格视图的快捷键

快捷键	功能
CTRL + 1	打开格式化单元格对话框

右键菜单

在电子表格内右键单击会显示一个菜单，其中包含可执行的多个电子表格操作。

设置	描述
剪切	剪切所选单元格。
复制	将所选单元格复制到剪贴板。
粘贴	粘贴剪切或复制到剪贴板的单元格。
插入	按所选行列数插入相同数量的行或列。如果没有选中任何行或列，则会显示插入对话框以插入单元格、行或列。
Delete (删除)	删除所选行或列。如果没有选中任何行或列，则会显示插入对话框以插入单元格、行或列。，则会显示删除对话框以删除单元格、行或列。
清除内容	清除活动单元格的内容。
插入函数	打开对话框插入函数到活动单元格。此选项仅在活动单元格不包含函数时可用。
编辑函数	如果活动单元格包含一个有关联属性表的函数，则打开属性表，否则打开插入函数对话框。此选项仅在活动单元格包含函数时可用。
插入绝对引用	在活动单元格插入一个绝对单元格引用。
插入相对引用	在活动单元格插入一个相对单元格引用。
插入批注	打开对话框插入新批注到活动单元格。此选项仅在活动单元格不包含批注时可用。
编辑批注	打开对话框编辑活动单元格的现有批注。此选项仅在活动单元格包含批注时可用。
插入符号标签	打开符号标签编辑器以插入新符号标签到活动单元格。此选项仅在活动单元格不包含符号标签时可用。
编辑符号标签	打开符号标签编辑器以编辑活动单元格的符号标签。此选项仅在活动单元格包含符号标签时可用。
单元格图像	进入交互图像模式以编辑单元格图像。此选项仅在活动单元格包含一个有关联单元格图像的函数时可用。
单元格状态	根据引用单元格的值，打开对话框启用或禁用单元格的执行（显式或条件方式）。
设置作业通过/失败	打开设置对话框监视一个包含数字结果的单元格，以确定作业整体通过/失败状态。作业状态可以发送到 RAPID 程序。
自定义视图设置	打开对话框配置电子表格自定义视图的属性。
便捷视图设置	打开对话框自定义数据显示的方式。

下一页继续

设置	描述
格式	<ul style="list-style-type: none"> • 格式化单元格：打开对话框设置所选单元格中数字、对齐、字体的格式以及保护。 • 行高：打开对话框调整一个或多个电子表格行的高度。 • 列宽：打开对话框调整一个或多个电子表格列的宽度。 • 隐藏：隐藏所选的行或列。 • 取消隐藏：显示所选行或列之间的隐藏行或列。
代码片段	<ul style="list-style-type: none"> • 导入：打开对话框将保存为 .CXD 文件的代码片段导入电子表格。此数据文件可以从 PC 的代码片段文件夹加载。 • 导出：打开对话框将保存为 .CXD 文件的代码片段导出到 PC 的代码片段文件夹。
导入单元格	将单元格数据以 .CXD 文件形式导入电子表格。
导出单元格	将单元格数据导出为 .CXD 文件。

不保护作业 / 保护作业对话框

保护作业对话框防止对作业中所有锁定单元格的修改。当作业中有单元格必须保持不变作业才能正常进行时很有用，可以让作业的其余部分由用户自定义。

作业以用户定义的密码保护，也可以空密码。要解除作业的保护需要密码。

当作业被保护后，任何锁定的单元格中的公式都无法被修改，但是锁定单元格的格式（字体、颜色等）仍然可以修改。当鼠标指针停留在锁定单元格上时，锁定单元格的公式仍然可见，但显示为灰色。



注意

使用格式化单元格对话框中的保护选项卡锁定单元格。

此页刻意留白

4 FlexPendant 用户界面

4.1 RobotWare Integrated Vision

简介

本节提供了 FlexPendant 上 Integrated Vision 应用程序图形用户界面的概述。视图、按钮以及用户界面的其他部分根据其内容以及访问方式介绍。

有关更多如何使用 FlexPendant 的一般信息，请参阅操作员手册 - 带 *FlexPendant* 的 *IRC5*。

使用下列步骤启动 RobotWare Integrated Vision。

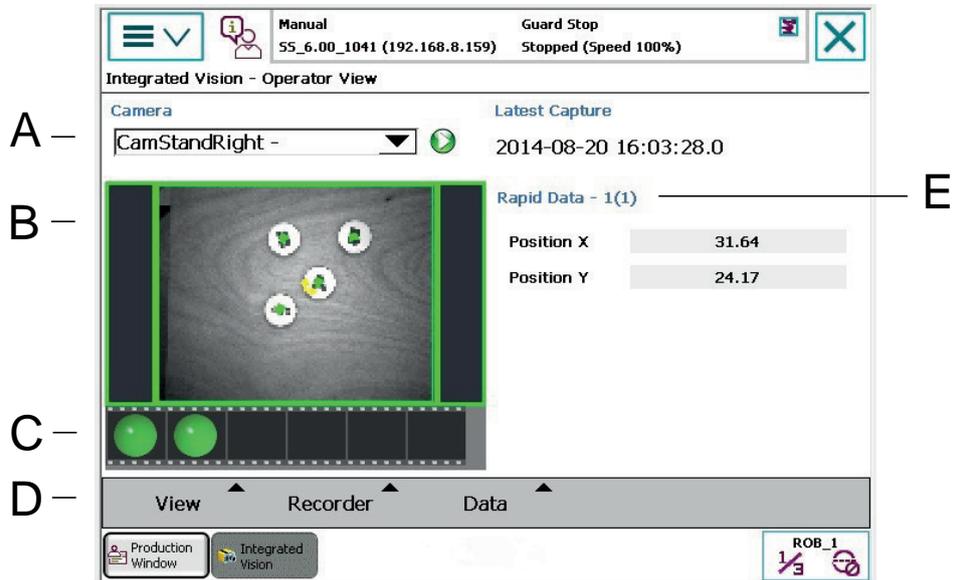
	操作
1	点击 ABB 菜单。
2	点击集成图像。

4 FlexPendant 用户界面

4.2 操作员视图

4.2 操作员视图

操作员视图界面



xx120000987

	部件说明	描述
A	摄像头	选择连接的摄像头和显示活动作业。
B	图像区域	显示摄像头采集的图像。
C	记录器工具条	用于记录一系列图像以便以后分析。
D	View (查看)	用于更改视图以显示图像和/或结果。
	记录器	用于显示和冻结记录器工具条以及保存图像。
	数据	用于设置结果视图。
E	结果区域	用于显示摄像头结果或 RAPID 数据。用户配置。

视图设置

默认设置是并排显示图像和结果，但也可以只显示图像或结果。

设置	描述
图像和结果	并排显示图像和结果
图像	只显示图像
结果	只显示结果



提示

点击图像区域进入全屏模式。全屏模式会保持 30 秒，然后会自动退出。

下一页继续

记录器设置

FlexPendant 中的记录器工具条是一个 RobotStudio 中胶卷工具条的简化版，请参阅 [第29页的胶卷](#)。

通过的图像以绿色表示，失败的图像则以红色表示。

设置	描述
显示	显示记录器工具条。
冻结	冻结记录器工具条。 当记录器工具条冻结时，可以单击具体的图像查看其内容。
Save (保存)	只有在记录器工具条冻结时方可执行保存。 活动图像会以当前时间信息保存为 .bmp 文件，保存在控制器闪存盘的 "...HOME\IV" 文件夹。

数据设置

结果区域可以自定义为显示任何用户感兴趣的 RAPID 数据。其布局取决于作业，所以每个作业都需要创建一个新布局。另外，还可以创建一个默认布局，用于在没有加载作业时显示。



注意

只有声明为持续 (PERS) 的 RAPID 数据才能显示在结果区域。

设置	描述
配置	通过添加标签和 RAPID 数据配置作业结果区域的布局
摄像头结果	显示在输出到 RAPID对话框映射的摄像头参数。 请参阅 第60页的输出到 RAPID 。
默认	显示在没有加载作业时显示的默认结果区域布局。

配置保存在控制器闪存盘 "...HOME\IV" 文件夹的 IVSetup.xml 文件。熟悉 XML 的用户可以直接编辑 IVSetup.xml 文件，而无需使用 FlexPendant 的配置工具。重启 FlexPendant 的 RobotWare Integrated Vision 应用程序以应用修改。

此页刻意留白

5 配置 Integrated Vision

5.1 建议的工作步骤

概述

本节介绍在创建新图像应用程序时的推荐工作步骤。此工作步骤有助于了解不同对象之间的依存关系。创建新应用程序的好方法是从基本功能开始。当基本功能按实现后，再扩展应用程序。

前提是必须先执行完软硬件安装步骤中的所有步骤，请参阅[第17页的安装](#)。

基本步骤

使用如下步骤创建新图像应用程序。

	操作	更多信息
1	进行初始准备。	第42页的准备工作
2	设置摄像头。	第43页的设置摄像头
3	创建新的图像作业。	第53页的设置新图像作业
4	调整摄像头的图像设置	第54页的设置图像
5	校准摄像头和机器人	第55页的校准 第74页的校准理论
6	添加图像工具以定位和检查图像中的部件。	第58页的添加图像工具
7	使图像数据对 RAPID 程序可用。	第60页的输出到 RAPID
8	设置摄像头的输入和输出，如有。	第64页的I/O 处理
9	在控制器上准备 RAPID 程序。	第65页的准备 RAPID 程序 第91页的RAPID 参考信息
10	开始生产	第70页的开始生产

有关设置图像系统的更多信息，请参阅[第76页的最佳做法](#)。

5 配置 Integrated Vision

5.2 准备工作

5.2 准备工作

准备工作

经验显示，在使用干净的系统开始时，最好先加载一个 RAPID 程序并进行一些初始准备。

- 创建全部所需工具的工具数据并定义 TCP。
- 创建全部所需固定装置的工件数据并定义。
- 等。

建议创建一个模块并添加 `MoveToDetectedObject` 代码片段。这样所有在校准、抓取点培训等期间编辑的数据都将会到位。



提示

在创建新图像程序时，使用 `MoveToDetectedObject` 代码片段作为基础。

有关添加图像代码的更多数据，请参阅第65页的 *RobotStudio* 中的 *RAPID* 代码片段。

5.3 设置摄像头

5.3.1 基本步骤

配置摄像头网络并连接到摄像头

当所有的摄像头都实际连接号后，还需要为每个摄像头配置一个 IP 地址和一个名称。摄像头的 IP 地址默认由控制器使用 DHCP 自动分配，但也可以使用静态 IP 地址。摄像头名称在系统的所有部分（例如 RobotStudio、RAPID 程序）中作为一个唯一的识别符。这可以实现在无需修改程序的情况下修改摄像头的 IP 地址（例如在更换了摄像头时）。

RobotStudio 中的 IRC5 控制器浏览器有一个叫图像系统的节点。用于配置和连接到摄像头。与摄像头的链接是通过机器人控制实现的。摄像头作为 FTP 远程加载磁盘连接。

使用下列步骤将摄像头分配给控制器。

	操作
1	确保 PC 的网卡已经设置为自动获取 IP 地址。
2	确保 PC 上安装的防火墙已经关闭或允许与摄像头的通信。
3	将 PC 连接到 IRC5 控制器的服务端口。
4	启动 RobotStudio。
5	连接控制器并通过控制器请求写入权限。
6	转到菜单条的控制器选项卡并启动 Integrated Vision 插件。
7	转到菜单条的图像选项卡。
8	<p>展开控制器浏览器中的图像系统节点。 命名的摄像头会显示其名称，未命名的摄像头则显示其 MAC 地址。</p> <p> 注意</p> <p>如果摄像头没有显示在图像系统列表，则可能是摄像头的 IP 地址被设置成了另外的子网。有关详细信息，请参阅第45页的将摄像头连接到不同的子网。</p>
9	<p>右键单击摄像头并选择连接。 来自摄像头的图像应该显示在图像采集和配置区域的一个单独的选项卡中。请使用摄像头图像识别正确的摄像头。</p>
10	如果需要，可按 Acquire Image （获取图像）按钮更新图像。
11	右键单击摄像头并选择 Rename （重命名）。
12	<p>在 Rename（重命名）对话框中，在 RAPID Camera Name（RAPID 摄像头名称）字段中输入摄像头名称。</p> <p> 注意</p> <p>建议将摄像头的主机名设置为与 RAPID 摄像头名称相同的名称。</p>
13	<p>重启控制器和摄像头。</p> <p> 注意</p> <p>控制器和摄像头都需要重启，这一点非常重要。</p>

下一页继续

5 配置 Integrated Vision

5.3.1 基本步骤

续前页

操作	
14	配置好的摄像头现在应该显示在控制器浏览器的图像系统节点。



注意

配置好的摄像头的名称存储在控制器的系统参数中，主题为 **Communication (通信)** (SIO.cfg)。IP 地址设置存储在摄像头中。

断开摄像头

要断开一个摄像头，只需单击 **Disconnect (断开)** 或在图像采集和配置区关闭对应的选项卡。

移除摄像头

要移除配置好的摄像头，必须先控制器的系统参数中删除摄像头，主题为 **Communication (通信)** (SIO.cfg)。

使用以下步骤移除：

操作	
1	使用 RobotStudio 中的 Configuration Editor (配置编辑器) 打开控制器的系统参数，主题为 Communication (通信) (SIO.cfg)。
2	删除所选摄像头的应用协议： COM_APP: -Name "MyCamera" -Type "CAMERA" -Trp "TCPIP1" -MAC "..."
3	重启控制器。

有关使用 RobotStudio 中的 **Configuration Editor (配置编辑器)** 的详细信息，请参阅操作员手册 - *RobotStudio*。

有关系统参数的详细信息，请参阅技术参考手册 - 系统参数。

5.3.2 其它摄像头配置

更改摄像头的 IP 地址

摄像头的 IP 地址可以在网络设置对话框修改。

建议将摄像头的主机名设置为与配置对话框中 **RAPID** 摄像头名称一样的名称。不这样设置系统仍然可以工作，但是部分 RobotStudio 中集成的 *Cognex EasyBuilder*® 对话框会使用主机名来识别摄像头。

请使用下列步骤来更改网络设置和摄像头主机名：

	操作
1	选择控制器浏览器中的图像系统节点中的摄像头。
2	单击连接按钮下拉列表菜单并选择网络设置。
3	设置与控制器和 PC 同一个子网的固定 IP 或者启用 DHCP。
4	建议将摄像头的 Host Name (主机名) 设置为与 Configuration (配置) 对话框中给出的 RAPID Camera Name (RAPID 摄像头名称) 相同的名称。  注意 请勿更改 Telnet 端口或其他设置。
5	单击确定。
6	重启摄像头。
7	重启控制器。

将摄像头连接到不同的子网

如果摄像头的 IP 地址与控制器和 PC 不处于同一个子网，在控制器浏览器的图像系统接口就不会显示摄像头，因此就无法使用网络设置对话框来设置新 IP 地址。

使用下列步骤将摄像头连接到不同的子网。

	操作
1	单击连接按钮下拉列表菜单并选择添加传感器。
2	单击列表中的摄像头，在与控制器和 PC 相同的子网设置一个固定 IP 地址或者启用 DHCP。
3	点击 Apply (应用) 。
4	重启摄像头。
5	重启控制器。

更改摄像头的日期和时间

使用下列步骤更改摄像头的日期和时间：

	操作
1	选择控制器浏览器中的图像系统节点中的摄像头。
2	单击连接按钮下拉列表菜单并选择设置日期和时间。
3	更改日期和时间设置。
4	单击确定。
5	重启摄像头。

下一页继续

5 配置 Integrated Vision

5.3.2 其它摄像头配置

续前页

更新摄像头固件

当使用不是与 Integrated Vision 备选件一起供应的图像摄像头时，可能需要升级摄像头固件。

Cognex In-Sight® 摄像头的最新的固件包含在 RobotStudio 安装程序中。

使用下列步骤更新摄像头固件：

操作	
1	选择或右键单击控制器浏览器中的图像系统节点中的摄像头。
2	单击高级下拉列表菜单并选择更新固件。 此时会显示摄像头型号、当前版本以及新版本固件。
3	单击更新。
4	重启摄像头。
5	重启控制器。



注意

运行 Integrated Vision 时，建议摄像头固件版本为 4.08.02 (023) 或更高。

将摄像头连接到虚拟控制器

如果没有实际的控制器，可以使用虚拟控制器。

将摄像头连接到虚拟控制器的步骤与连接实际控制器的步骤相同。

但在连接摄像头前，必须在 PC 的虚拟控制器 HOME 目录添加网络配置文件。

使用下列步骤创建、添加和修改网络配置文件。

操作	
1	根据以下具体说明创建 <code>vc_network_definition.xml</code> 文件。
2	将 <code>vc_network_definition.xml</code> 文件放在 PC 上虚拟控制器 HOME 目录下。
3	检查摄像头连接的 PC 网卡的 IP 地址。
4	将 XML 文件中的服务端口 IP 地址更改为 PC 网卡的 IP 地址。 <pre><PORT name="SERVICE"> <IPAddress>192.168.125.100</IPAddress></pre>
	 注意 请勿更改任何其他设置。
5	重启虚拟控制器。
6	按连接实际控制器相同的步骤将摄像头连接到虚拟控制器。 请参阅 第43页的配置摄像头网络并连接到摄像头 。

创建 `vc_network_definition.xml` 文件

将下列内容使用记事本或者类似程序复制到文本文件。将文件命名为 `vc_network_definition.xml`。

```
<?xml version="1.0" encoding="ISO-8859-1" ?>  
<VC_NETWORK_DEF>  
  <PORT name="LAN">  
    <IPAddress>10.46.77.126</IPAddress>
```

下一页继续

```

    <SubnetMask>255.255.252.0</SubnetMask>
  </PORT>
  <PORT name="SERVICE">
    <IPAddress>192.168.125.100</IPAddress>
    <SubnetMask>255.255.255.0</SubnetMask>
  </PORT>
</VC_NETWORK_DEF>

```

限制



注意

- 在与摄像头通信时，虚拟控制器的性能不如实际控制器。
- 虚拟控制器以“现状”方式提供，因此仅应用于评估、培训、支持、测试和类似用途。

将摄像头模拟器连接到虚拟控制器

当创建一个具有虚拟控制器的系统时，一个摄像头模拟器处在可用状态。如果没有真实的摄像头可用，可使用该模拟器进行评估、培训、支持、测试和类似功能。在模拟器中，始终评估当前加载的图像。模拟器没有遍历图像的功能。

连接摄像头模拟器的步骤与连接真实摄像头相同。

模拟器默认由 Integrated Vision 插件启动，并将自己连接到 PC 上一个可用的网卡。可以更改它模拟的摄像头的类型，请参阅 [第32页的Integrated Vision 选项](#)。

使用此步骤检测摄像头模拟器：

操作	
1	<p>如果可能，禁用 PC 上的所有网卡，只留下一个。</p> <p> 注意</p> <p>如果有多个可用的网卡，可能需要重复下面的步骤，直到找到正确的网卡。</p>
2	<p>检查 PC 网卡 IP 地址。</p> <p> 注意</p> <p>如果 PC 上的网卡设置为 DHCP，PC 重启时 IP 地址会改变。</p>
3	<p>将 <code>vc_network_definition.xml</code> 文件中的 SERVICE 端口 IP 地址更改为 PC 网卡的 IP 地址。</p> <pre> <PORT name="SERVICE"> <IPAddress>192.168.1.1</IPAddress> </PORT> </pre> <p> 注意</p> <p>请勿更改任何其他设置。</p>
4	重启虚拟控制器。
5	此时在控制器浏览器的图像系统节点中检测到了摄像头模拟器。

下一页继续

5 配置 Integrated Vision

5.3.2 其它摄像头配置

续前页

限制



注意

- 只有在使用虚拟控制器时才可以访问摄像头模拟器。
- 摄像头模拟器以“现状”方式提供，因此仅应用于评估、培训、支持、测试和类似用途。
- 不可以为摄像头模拟器设置一个主机名。

5.3.3 限制用户访问

简介

RobotStudio 直接与摄像头通信。因此摄像头本身带有用户身份验证方法，以便有可能限制特定功能的使用。

RobotStudio 和控制器均使用密码来登录到摄像头。默认情况下，用户名为 "admin"，密码为空 ""。

摄像头上的用户列表及其相关权限是可以编辑的，另外，也可以选择当控制器与摄像头通信时所使用的用户名和密码。当更新摄像头用户列表时，确保单击 **Set controller user** (设置控制器用户) 来同样更新机器人控制器使用的用户档案。

用户身份验证系统 (UAS)

摄像头使用的用户身份验证方法并不属于机器人控制器使用的用户身份验证系统 UAS。

部分涉及控制器的操作要求写访问权限，例如摄像头命名。这时由于该信息是存储在控制器配置中的。所有其它操作则适用摄像头的用户身份验证。

有关 UAS 的详细信息，请参阅操作员手册 - *RobotStudio*。

编辑用户访问设置

User Access Settings (用户访问设置) 对话框用于维护 *Cognex In-Sight*® 图像系统和仿真器授权用户的访问级别和 FTP 读写权限。**User Access Settings** (用户访问设置) 对话框可确定允许哪些用户登录到特定摄像头以及他们能对活动作业执行的修改类型。每个摄像头都有其自己的用户列表，独立于网络上的其它摄像头。如果用户需要访问特定摄像头，则必须知道该摄像头用户列表中已经存在的用户名和密码。



注意

每个摄像头都预先配置了三个用户：*admin*、*operator* 和 *monitor*。这些用户分别对应着 *Full*、*Protected* 和 *Locked* 访问级别。



注意

允许添加到一个摄像头的最大用户数量为 32 个。

设置控制器用户

请使用以下步骤来选择在与摄像头通信时由控制器使用的用户档案。

	操作
1	申请控制器写权限。
2	单击连接按钮下拉列表菜单并选择 Set Controller user (添加控制器用户)。
3	单击 Select User (选择用户) 下拉列表菜单并选择一个用户。
4	单击 OK (确定)。
5	重启控制器。

下一页继续

5 配置 Integrated Vision

5.3.3 限制用户访问 续前页

添加新用户

使用下列步骤添加新用户。

操作	
1	单击连接按钮下拉列表菜单并选择 User Access Settings (用户访问设置)。
2	单击添加。
3	在 User Name (用户姓名) 字段输入一个字母字符串。  注意 用户名和密码的长度不能超过 30 个字符，两者均为大小写敏感。
4	输入新用户的密码。
5	单击 Access (访问) 下拉列表菜单并为新用户选择一个访问级别。 有关详细信息，请参阅第50页的访问级别。
6	或者，禁用允许运行模式/程序模式复选框以限制具有 <i>Protected</i> 访问权限的用户切换传感器的程序模式/运行模式状态。 当所选访问级别为 <i>Full</i> 或 <i>Locked</i> 时此复选框为灰色。
7	或者，启用允许运行模式作业保存复选框以便让具有 <i>Full</i> 或 <i>Protected</i> 访问权限的用户在运行模式中保存作业。 当所选访问级别为 <i>Locked</i> 时此复选框为灰色。  注意 选中后，具有 <i>Protected</i> 访问权限的用户将可以运行模式保存作业，即使未启用 FTP 写入权限也不受影响。
8	指定用户的 FTP 读写权限。 有关详细信息，请参阅第51页的 <i>FTP</i> 权限。
9	点击 OK (确定)。

访问级别

访问级别控制允许当前用户进行多少交互活动，以防止对配置造成意外或未经授权的更改。所选的访问级别将在任何人使用所选用户名和密码登录到 RobotStudio 图像插件时生效。

有三个访问级别可选：

访问级别	描述
完全	提供对摄像头全面的不受限制的访问权限。 可以加载、修改或保存在 RobotStudio 中创建的任何作业，所有的菜单选项均可以使用。默认的 <i>admin</i> 用户帐户具有完全访问权限。
受保护	用户对传感器具备受限制的访问权限。 受保护模式可以让您访问实时视频模式、切换摄像头的运行模式/程序模式（如果运行模式/程序模式权限允许）、以及打开或保存作业（如果 FTP 读写权限允许）。 在标准视图下（非电子表格视图），“受保护”访问权限可以让您编辑工具参数，但不能添加或删除工具。 在电子表格视图下，“受保护”访问权限会始终显示电子表格的自定义视图。在保护模式下的用户可以编辑在自定义视图中可见的任何图像控制功能的值，但不能更改功能本身。默认的操作员用户帐户具有“受保护”访问权限。 如需了解更多有关电子表格视图和自定义视图的信息，请参阅 Cognex 网站上的 <i>Cognex In-Sight® Explorer User's Guide</i> 。

下一页继续

访问级别	描述
锁定	提供限制最大的访问权限。 您只能监视当前摄像头的运行。默认的监视用户帐户具有“锁定”访问权限。

FTP 权限

FTP 读权限适用于打开来自摄像头的作业或图片文件，FTP 写权限适用于将作业或图像保存到摄像头。

此外，这些权限在用户试图登录用网络的远程 FTP 客户端登录到当前摄像头时也有效。具体来说，要让控制器能在摄像头上执行文件操作，则需要启用这些权限。

控制器用户所需的权限和访问级别

机器人使用的用户配置文件需要 FTP 读写权限才能在摄像头和控制器之间实现文件传输。

所需的访问级别为保护，因为这是允许在运行模式和程序模式之间切换的最低级别。请确保选中了允许运行模式/程序模式。



注意

控制器用于登录到摄像头的用户配置文件和关联的密码以明文方式存储在机器人配置中。请将对此用户配置文件的权限尽量保持最低。

编辑已有用户

使用下列步骤编辑已有用户。

	操作
1	单击连接按钮下拉列表菜单并选择 User Access Settings (用户访问设置)。
2	单击 Edit (编辑)。
3	修改必要的用户设置。 关于设置的详细信息请参阅 第50页的添加新用户 。



注意

- 如果修改了 *admin* 密码，必须注销并用新密码重新登录，否则可能会出现错误。
- 访问级别、允许运行模式/程序模式权限、FTP 权限以及 *admin* 用户名无法修改。
- **Show Custom View at Log On** (登录时显示自定义视图) 复选框只影响电子表格视图。标准 RobotStudio 视图并不支持自定义视图。

删除已有用户

使用下列步骤删除已有用户。

	操作
1	单击连接按钮下拉列表菜单并选择 User Access Settings (用户访问设置)。
2	单击 Delete (删除)。
3	确认删除。

下一页继续

5 配置 Integrated Vision

5.3.3 限制用户访问

续前页



注意

admin 用户无法删除。

5.4 设置新图像作业

连接到摄像头

打开 RobotStudio 并确保摄像头根据第43页的配置摄像头网络并连接到摄像头的要求连接并经过测试。

创建并保存新作业

Integrated Vision 中的所有摄像头配置和设置统称为作业。

活动作业存储在摄像头的工作内存中，在停电时会丢失。作业应在摄像头闪存盘或机器人控制器的闪存盘上永久性保存为作业文件 (.job)。

为了能从 RAPID 图像指令加载 job 文件，作业必须保存在摄像头闪存盘。但是我们强烈建议同时在其他地方创建一个备份，以防摄像头损坏。

如果使用机器人控制器作为作业存储空间，则必须将从控制器盘复制 job 文件到摄像头闪存盘，然后才能从摄像头闪存盘加载。这是通过标准的 RAPID 文件处理指令完成的。



提示

有关传输示例，请参阅 RobotStudio 中的备份和还原代码片段。

使用如下步骤可创建新作业。

	操作
1	确保摄像头处于编程模式。
2	单击菜单条中的新建作业。
3	单击是清除当前作业全部数据。
4	在菜单条单击保存作业或将作业另存为。 因为作业此前未保存过，所以会显示另存为对话框。
5	浏览到希望的位置，最好是摄像头闪存盘。
6	给作业命名并单击保存。 作业的名称将会显示在图像采集和配置区域的图像选项卡。

放置摄像头

如果摄像头安装在机器人的活动部件上，应该将微调确定摄像头的位置并保存此位置。在采集新图像前，摄像头始终需要准确回到同一个位置以保证精确性。

5 配置 Integrated Vision

5.5 设置图像

5.5 设置图像

简介

最常用的设置为曝光时间。时间越长，允许进入摄像头的光就越多，图像也就越亮。调整图像作业的设置通常是一种重复性的作业，常常需要在作业准备就绪前再一次或多次修改曝光时间。有时，得到用于校准的清晰图像所需的设置可能与检测产品的理想设置不完全相同。如果在后续步骤中证实图像设置并不理想，请勿匆忙地进行修改。



提示

有关详细信息请参阅第76页的最佳做法一章，特别是第79页的获取良好的照明和第77页的如何安装摄像头这两节。

图像触发事件

图像触发事件设置决定触发摄像头采集图像的事件。为了能从Integrated Vision RAPID指令触发图像采集，必须将触发事件设置为摄像头或外部。



注意

如果使用了RAPID指令CamReqImage，设置摄像头图像触发类型为摄像头。如果同一个RAPID指令CamReqImage搭配可选变元\AwaitComplete使用，则摄像头图像触发类型必须设置为外部。

图像设置

使用下列步骤可设置图像。

	操作
1	确保摄像头处于编程模式。
2	单击菜单条中的设置图像。
3	如果在搭配可选变元\AwaitComplete使用CamReqImage，在上下文窗口将触发设置改为摄像头或外部。
4	如有必要，调整其他设置以取得最佳图像质量。
5	保存作业。

有关所有参数和设置的信息，请参阅在线帮助选项卡中的设置图像/调整编辑采集设置一节。



注意

图像设置直接连接到摄像头内存。所有更改都会立即生效。因此没有“应用”、“保存”或“撤销”选项。请经常保存和备份作业以防止数据丢失。

5.6 校准

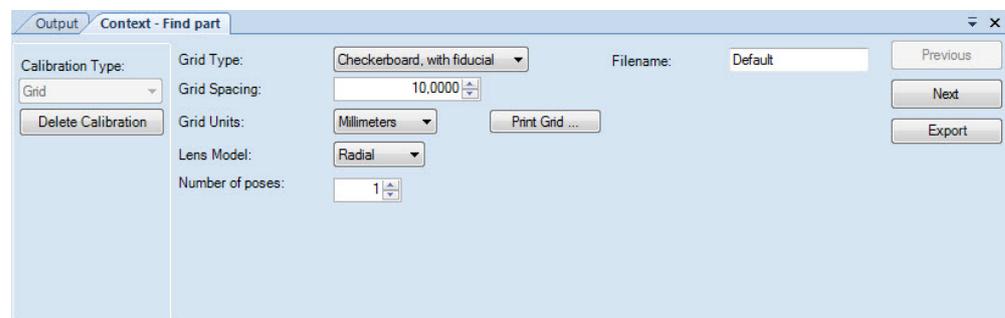
简介

图像由像素组成，因此为了获取毫米级的结果，需要先校准摄像头。校准功能用于按实际部件校准摄像头。

校准有两个基本步骤。首先是摄像头校准，即将图像像素转换为毫米，然后是摄像头对机器人的校准，将摄像头坐标与机器人机架（工件）关联起来。

设置显示在上下文窗口。在线帮助选项卡以及[第74页的校准理论](#)一节介绍了不同的摄像头校准类型。

为了取得最好的精度，Integrated Vision 的建议校准类型是使用一个带有基准标记的国际象棋盘，请参阅[第57页的摄像头对机器人校准](#)。基准标记提供了清晰的参照，以后可以用于定义对应的工件。



xx1300001097



注意

最重要的是设置校准网格的正确尺寸，否则校准会不正确。

摄像头校准

按照以下步骤可校准传感器。

	操作
1	确保摄像头处于编程模式。
2	单击菜单条中的校准。
3	在上下文窗口，将校准类型修改为网格。
4	从网格类型下拉菜单选择一个带有基准标记参照点的棋盘校准板。
5	如有必要调整间隔、单位、镜头型号以及姿势数量设置。 <ul style="list-style-type: none"> 使用 mm 作为单位 镜头型号取决于预计的最大失真位置。或者是由于摄像头的角度问题（透视），或是因为镜头本身会使图像失真（径向）。 姿势数量可以允许使用超过一个校准板图像来校准摄像头，以防校准板不能覆盖整个视野。

下一页继续

5 配置 Integrated Vision

5.6 校准 续前页

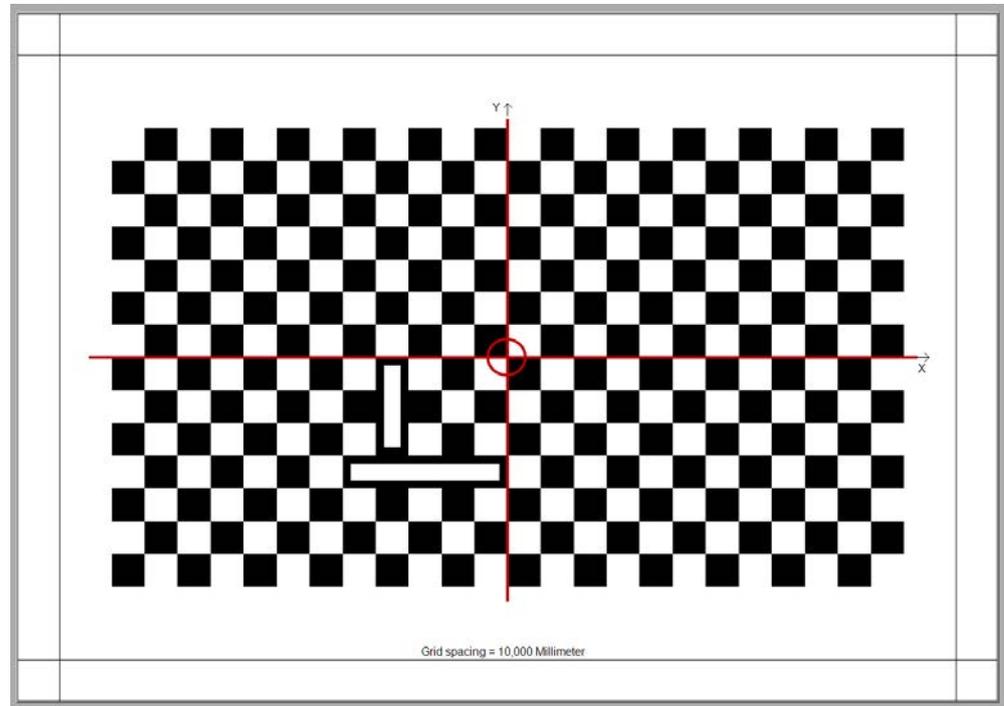
操作	
6	<p>单击打印网格以打印校准板。打印的图像必须具有高对比度且纸张不得反光（高光）。使用尺子验证方块是否比例成比例。</p> <p> 提示</p> <p>在 RobotStudio 安装文件夹下有预先印制的校准板。如果在打印时无法进入校准向导可以使用这些校准板： <i>C:\... \ABB Industrial IT\Robotics IT\RobotStudio x.xx\Bin\Addins\IntegratedVision\PDF Grids</i></p>
7	<p>将校准板放在摄像头图像中心的一个固定位置，与摄像头要识别的物体的相同高度。校准纸必须完全平整、得到充分照明且没有反光和阴影。旋转校准板使 X 和 Y 箭头对应摄像头工件的预期朝向。</p>
8	<p>在上下文窗口单击下一步。 校正由摄像头计算中，此时会显示找到的特征点数量。</p>
9	<p>单击 Next (下一步)。</p>
10	<p>单击校准应用校准结果。</p>
11	<p>单击完成退出校准。</p>
12	<p>保存作业。</p>
13	<p>在工件定义好前不要更改校准板的位置。</p>

有关所有参数和设置的信息，请参阅在线帮助选项卡中的设置图像/根据实际部件校准图像一节。

摄像头对机器人校准

摄像头通过定义与校准板相同坐标原点的工件校准到机器人。

在带有基准的国际象棋盘校准板上，坐标原点位于 X 和 Y 箭头延伸的交叉处（如下图）。



xx120000994



注意

在定义工件前，确保检查原点当前使用的校准板上。

按照以下步骤校准机器人。

操作	
1	创建一个支点工具并使用精确的方法定义工具的 TCP。
2	为每个摄像头创建一个工件。
3	激活支点工具，并沿着校准板的对应 X 和 Y 轴定义摄像头工件的用户框架。将对象框架留空。
4	通过控制机器人在工件上微动测试校准是否正确。
5	现在可以移掉校准板了。

有关如何创建和定义工具和工件的信息，请参阅操作员手册 - 带 *FlexPendant* 的 *IRC5* 中的编程与测试一章。

5 配置 Integrated Vision

5.7 添加图像工具

5.7 添加图像工具

简介

定位工具用于定义图像中的某个特征以便提供位置数据。定位工具会创建一个参照点，用于在图像中快速可靠的定位一个部件，即使部件处于旋转检查或出现在图像的不同位置。

检查工具用于检查部件是否位于定位工具规定的位置。根据当前应用程序的要求，有检查出现/缺少、测量、计数、几何等的不同工具可用。

设置显示在上下文窗口中。

添加定位工具

使用下列步骤可添加定位工具。

	操作
1	加载或采集一个新图像。 确保要定位的部件出现在图像中可能出现部件的区域中。
2	单击添加部件定位工具，然后单击下拉菜单中所需的工具。
3	按上下文窗口中的工具相关说明操作。
4	如有必要，调整设置以取得最佳性能。
5	保存作业。

有关所有参数和设置的信息，请参阅在线帮助选项卡的定位部件一节。

最常用的定位工具

部件定位工具	描述
PatMax® 图案 PatMax® 图案 (1-10)	使用 PatMax® 算法定位一个图案或最多 10 个图案，并报告所发现图案的 x 和 y 坐标、角度和分数。
Blob Blobs (1-10)	定位一组或最大 10 组的深色或浅色相对应像素，称为 blob，并报告所发现 blob 的 X 和 Y 坐标。 此工具通常用来作为引导其他图形工具的固定装置。

最常用的定位工具设置

设置	描述
Number To Find (要找到的数量)	定义要检测的实例的数量。默认值通常是 1，要检测多个实例时必须增大。
Rotation Tolerance (旋转公差)	定义找到的图案从训练图案旋转多大角度时仍能被视为有效图案。默认值 +/-10° 到 15° 常常过小，需要增大。

添加检查工具

使用下列步骤可添加检查工具。

	操作
1	加载或采集一个新图像。 确保要定位的图案出现在图像中可能出现图案的区域中。
2	单击添加部件检查工具，然后单击下拉菜单中所需的工具。
3	按上下文窗口中的工具相关说明操作。

下一页继续

	操作
4	如有必要，调整设置以取得最佳性能。
5	保存作业。

有关所有参数和设置的信息，请参阅在线帮助选项卡的检查部件一节。

最常用的检查工具

工具组和部件检查工具	描述
出现/缺少工具 • PatMax® 图案	使用 PatMax® 算法确定某个训练图案是出现还是缺少。如果图案出现且处于限制内则报告通过，如果超出限制则报告失败。
出现/缺少工具 • Blob Blobs (1-10)	确定是否出现或缺少一组深色或浅色的关联像素。如果出现 blob 特征且处于限制内则报告通过，如果超出限制则报告失败。
测量工具 • 距离	测量任何两个特征（例如边缘、圆环、图案和/或 blob）之间的距离。以毫米或像素表示距离（除非图像已经校准），否则如果报告的距离超出限制则报失败。
识别工具 • PatMax® 图案 (1-10)	使用 PatMax® 算法在训练图案库中确定最符合图像中图案的图案。报告发现图案的名称以及其分数与培训模型的对比。如果找到图案且位于限制内则报告通过；如果超出限制或未找到图案则报失败。

图像工具的通过与失败

每个定位和检查工具都有一个复选框用于定义是否工具的通过/失败状态应包含在作业的总体通过/失败状态。默认情况下，会检查此数据并会包含在作业整体的通过/失败状态中。

清空此对话框将工具的通过/失败状态与作业其他工具分开来。

如果工具预计会失败则应不要选中此空间。例如，如果两个识别工具被用于确定是否部件为右手部件还是左手部件。如果选择了此复选框，则会导致无论识别部件的那个面，作业每次都失败。

图像工具之间的链接

一个图像工具的输出结果可以被用作另一个工具的输入参数。然后两个工具就链接了。

工具属性分成输入和输出属性，链接只能从输出属性建立到输入属性。输出属性可以链接到多个不同的输入属性，而输入属性只能接受一个输出属性的链接。最常见的例子是定位工具被当成检查工具的固定装置这样的情况。

链接在结果选项卡显示为箭头。图形箭头无法编辑。要删除或修改链接，必须从图像工具删除参数。

5 配置 Integrated Vision

5.8 输出到 RAPID

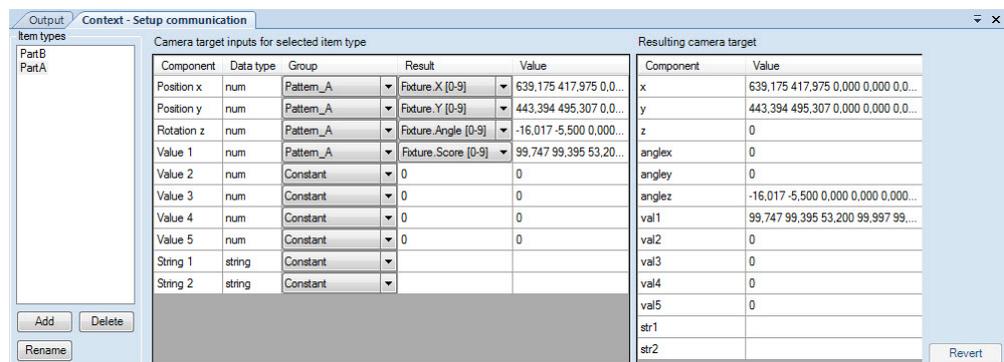
5.8 输出到 RAPID

简介

摄像头作业随着采集的每个图像会产生一些参数。最重要的是配置的图像工具的输出，但是所使用的曝光时间等附加数据也很重要。

输出到 RAPID 中的映射对话框提供了一个简单但灵活的方法，可以让用户选择哪些数据要转成 RAPID 变量。其目的在于允许用户进行试验并调整图像工作而无需修改利用图像数据的 RAPID 程序。

此对话框可以让用户点击选择要映射到某个 RAPID `cameratarget` 记录属性的图像输出参数。另外实现了通过定义图像作业产生的摄像头目标的名称属性来创建摄像头目标类别。



xx1300001098

设置通信

输出到 RAPID 功能显示在上下文窗口。图像摄像头数据显示在左边，结果 RAPID `cameratarget` 显示在右边。来自摄像头的的数据通过从下拉菜单选择所需的数据和参数，映射到 `cameratarget`。

数据类型 `cameratarget` 有一些预定义的数据组件。只要注意数据类型，这些数据可以按任何方式使用。如需数据类型 `cameratarget` 的完整说明，请参阅 [第 116 页的 cameratarget - 摄像头数据](#)。

查看	描述
项目类型	用于定位或检查工具识别的部件的说明性名称（例如“螺母”、“螺丝”、“螺栓”等）。 此名称将会被传输给后续产生的 <code>cameratarget</code> 的 <code>name</code> 参数。
所选项目类型的摄像头目标输入	<ul style="list-style-type: none">• 组件：对应后续产生的 <code>cameratarget</code> 参数的组件。• 数据类型：对应结果 <code>cameratarget</code> 参数的数据类型。• 组：组是数据源，通常是作业中配置的检查或定位工具。也可能从作业本身、输入信号、摄像头或常数获取数据。每个数据组都是一个包含很多结果（参数）的数组。• 结果：组数据的结果（参数）。通常是 x, y, z, 以及机器人的角度坐标。根据数据的不同有一些参数可用。这些参数在每个具体工具的 帮助选项卡中可以查到，位于“... 工具输入/输出属性”一节下。• 值：结果的值。通常是一个数字值或文字字符串。

下一页继续

查看	描述
结果摄像头目标	<p>显示将要传输到结果 RAPID cameratarget 的值。</p> <p> 注意</p> <p>摄像头只能识别已定位部件的旋转，z 角度。此值在 RAPID 中转换为四元数，因为对应的 RAPID 组件是 pose。</p>

映射数据

使用下列步骤使图像数据可以被 RAPID 程序使用。

	操作
1	单击输出到 RAPID。
2	单击项目类型视图中的添加来创建一个新项目类型。
3	单击重命名来为项目类型添加一个说明性的名称。 例如定位或检查工具识别的部件的名称（例如“螺母”、“螺丝”、“螺栓”等）。名称会被复制到结果 cameratarget 的 type 参数。
4	通过从组（后接结果下拉菜单）选择图像数据来将添加图像数据到 RAPID cameratarget。
5	在结果摄像头目标视图。
6	保存作业。
7	运行 RAPID 指令 CamSetRunMode 将当前输出的控制器更新为 RAPID 配置。

有关不同定位工具及其设置的信息，请参阅在线帮助选项卡的定位部件一节。

有关不同检查工具及其设置的信息，请参阅在线帮助选项卡的检查部件一节。



注意

建议在开始配置 Integrated Vision 系统后，不要更改 RobotStudio 的语言。

有关详细信息，请参阅[第33页的更改语言](#)。

限制

在完成输出配置时有些需要注意的限制。

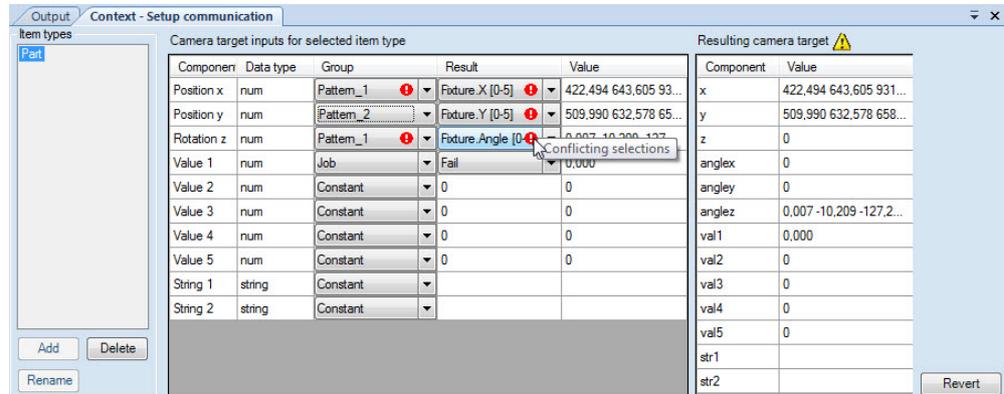
映射到每个组件的值必须配置。例如如果一个工具找到了 10 个值，比如 PatMax® Patterns (1-10)，映射到 X 坐标，建议不要映射不同大小的工具输出到另一个组件。那样可能会导致无法预料的结果。

5 配置 Integrated Vision

5.8 输出到 RAPID

续前页

请遵守 Integrated Vision 插件给出的警告和注意事项：



xx1300001100

此外，只有预计随着图像采集变化的值才要包含在结果输出中。在运行时需要的其他参数值建议使用 RAPID 指令 `CamGetParameter` 来预先获取。

最常用的图像数据

定位工具数据

下列数据结果通常在定位部件时用到：

符号名称	描述
固定装置 X	定位部件的 X 坐标。
固定装置 Y	定位部件的 Y 坐标。
固定装置角度	所选部件的旋转。  注意 摄像头只能识别已定位部件的旋转，z 角度。此值在 RAPID 中转换为四元数，因为对应的 RAPID 组件是 <code>pose</code> 。
Fixture.Score	论及找到的图案与培训图案的相似程度的百分数值。
通过	如果找到部件则设置为 1，如果没有找到则设置为 0。
失败	如果找到部件则设置为 0，如果没有找到则设置为 1。

检查工具数据

下列数据结果通常在检查部件时用到：

符号名称	描述
距离	两点之间的测量距离。
结果	如果找到训练对象则返回字符串 <i>Present</i> ，如果未找到则返回 <i>Not Present</i> 。
字符串	将识别的条形码数字值作为字符串返回。

作业数据

下列作业产生的数据经常被用到：

符号名称	描述
通过	如果设置为包含在作业中通过的所有图像工具通过则设置为 1，如果一个或多个包含的图像工具失败则设置为 0。

下一页继续

符号名称	描述
失败	如果设置为包含在作业中通过的所有图像工具为通过则设置为 0，如果一个或多个包含的图像工具失败则设置为 1

5 配置 Integrated Vision

5.9 I/O 处理

5.9 I/O 处理

简介

在应用程序包括需要由摄像头控制的外部照明最可能用到 I/O。如果没有 I/O 信号连接到摄像头，则无需配置。



注意

Inputs Outputs 函数仅用于定义摄像头的输入和输出而非机器人控制器的输入和输出。

概述

通过自定义输入线路名称、设置信号类型、选择信号沿变化以及强制输入测试应用程序，**Inputs** 函数可用于定义已安装的 *Cognex* 摄像头 I/O 模块的独立输入设置。

通过自定义输出线路名称、设置信号类型、选择信号沿变化以及强制输出测试应用程序，**Outputs** 函数可用于定义已安装的 *Cognex* 摄像头 I/O 模块的独立输出设置。部分摄像头具有集成的输出和 LED，其输出线路已经预定义，不能用于其他用途。

设置显示在上下文窗口中。

配置输入

使用下列步骤配置摄像头或任何已安装摄像头 I/O 模块的输入。

	操作
1	单击输入。 可用的 I/O 线路显示在上下文窗口中。
2	使用信号类型、信号沿类型以及强制输入下拉菜单可配置输入的预期行为。
3	要测试输入，摄像头必须处于运行模式。
4	保存作业。

有关所有参数和设置的信息，请参阅在线帮助选项卡的输入一节。

配置输出

使用下列步骤配置摄像头或任何已安装摄像头 I/O 模块的输出。

	操作
1	单击输出。 可用的 I/O 线路显示在上下文窗口中。
2	使用信号类型、作业结果以及强制输出下拉菜单可配置输出的预期行为。
3	要测试输出，摄像头必须处于运行模式。
4	保存作业。

有关所有参数和设置的信息，请参阅在线帮助选项卡的输出一节。

5.10 准备 RAPID 程序

5.10.1 RobotStudio 中的 RAPID 代码片段

简介

代码片段是指您可以插入 RAPID 程序中的预定义的 RAPID 代码段。为帮助编程和图像系统调试，已经提供了一些代码片段。



提示

在创建新图像程序时，使用 MoveToDetectedObject 代码片段作为基础。

添加代码片段

RAPID 代码片段位于 RobotStudio RAPID 选项卡的代码片段下拉列表。

使用下列步骤添加代码片段。

	操作
1	申请控制器写权限。
2	展开控制器浏览器的 RAPID 节点，找到程序模块。
3	创建新程序模块或打开现有模块。将光标放在要插入代码片段的位置。
4	单击代码片段下拉列表并选择集成图像以查看包含的一系列代码片段。
5	单击一个代码片段将其包含在程序模块中。

有关更多如何使用 RobotStudio RAPID 选项卡进行控制器变成的信息，请参阅 操作员手册 - *RobotStudio*。

5 配置 Integrated Vision

5.10.2 基本编程示例

5.10.2 基本编程示例

简介

本节介绍了如何编写基本的图像引导机器人程序。主要目的是提供需要调用的指令的大致情况。另外也提供了一些高级示例。

有关图像相关 RAPID 指令、函数和数据类型的详细信息，请参阅 [第91页的RAPID 参考信息](#)。



注意

在运行 RAPID 程序前，必须先完成本章前述的所有配置步骤。请参阅 [第41页的配置 Integrated Vision](#)。

创建基本 RAPID 程序

以下示例说明了运行图像引导机器人程序的基本步骤。为了更好的概括，此处省略了错误处理和其他增强功能。

以下 RAPID 程序的目的是将机器人移动到可以捡起摄像头检测到部件的位置。本程序基于 RobotStudio 中的 MoveToDetectedObject 代码片段。有关更多信息，请参阅 [第65页的RobotStudio 中的 RAPID 代码片段](#)。

```
1    ...
2    PERS wobjdata mywobj := ... ;
3    PERS tooldata mytool := ... ;
4    PERS robtargt myrobtargt := ... ;
5    CONST string myjob := "myjob.job";
6    VAR cameratargt mycameratargt;
7    ...
8    PROC MoveToDetectedObject()
9        CamSetProgramMode mycamera;
10       CamLoadJob mycamera, myjob;
11       CamSetRunMode mycamera;
12       CamReqImage mycamera;
13       CamGetResult mycamera, mycameratargt;
14       mywobj.oframe := mycameratargt.cframe;
15       MoveL myrobtargt, v100, fine, mytool \WObj:=mywobj;
16    ENDPROC
17    ...
```

行	备注
2 - 6	声明数据。
9 - 10	设置摄像头为编程模式并加载作业。
11 - 12	设置摄像头为编程模式并采集图像。
13	获取图像结果并将其保存在摄像头目标中。
14	从摄像头目标向工件的对象框架复制图像坐标。
15	将机器人移动至其捡拾位置。
-	(捡起部件。)

下一页继续

有关如何创建模块、创建例行程序和添加 RAPID 指令的信息，请参阅操作员手册 - 带 FlexPendant 的 IRC5 中的编程与测试一章。

训练部件夹持位置

夹持部件通常与移动 TCP 到摄像头报告的目标并不一样。通常必须首先复位此位置然后旋转一定值以实现良好夹持。

最简单的方法是微调机器人到指定位置，然后修改位置。

	操作
1	确保摄像头只能看到一个部件。
2	运行前面的示例程序到 MoveL 指令并停止执行。 此时 mywobj 的对象框架已经修改，正确的工具 mytool 已经激活。
3	确认图像摄像头成功地找到了该部件的位置。
4	微动调节机器人到适合夹持的位置。
5	标记位置 myrobtargget 并点击 Modify Position (修改位置) 。
6	从头与运行程序，确保部件是按预定的位置夹持的。
7	移动部件，然后再次重头运行程序。

5 配置 Integrated Vision

5.10.3 高级编程示例

5.10.3 高级编程示例

简介

下列示例展示了部分可以在运行图像引导机器人程序时使用的较高级功能。示例展示了错误处理、场景ID的使用以及其他增加生产效率和节省周期时间的方法。

数据声明

```
CONST string myjob := "myjob.job";
PERS robtarget myrobtarget :=
    [[100,200,300],[1,0,0,0],[0,0,0,0],[9E9,9E9,9E9,9E9,9E9,9E9]];
VAR cameratarget mycameratarget;
...
PROC integratedvisionadvanced()
    VAR num mysceneid:=0;
    VAR num myexposuretime:=10;
    VAR bool targetsavailable := TRUE;
    ...
    ! Insert the code here
    ...
ENDPROC
```

异步加载

在移动机器人时异步加载作业，然后在作业完成加载后立即采集图像。这可以让机器人在加载时执行其他任务，从而节省了周期时间。

```
CamSetProgramMode mycamera;
CamStartLoadJob mycamera, myjob;
!MoveJ myrobtarget, v100, fine, toolvision \WObj:=wobjvision;
CamWaitLoadJob mycamera;
CamSetRunMode mycamera;
CamReqImage mycamera;
```

使用 SceneId 参数

获取图像并在结果可用时立即获取结果。使用 SceneId 参数可以实现在最新图片中没有找到目标时快速通知用户。

```
CamReqImage mycamera \SceneId:=mysceneid;
CamGetResult mycamera, mycameratarget \SceneId:=mysceneid;
```

更换参数

在当前曝光时间上增加 5 ms

```
myexposuretime := CamGetExposure (mycamera \ExposureTime);
myexposuretime := myexposuretime + 5;
CamSetExposure mycamera \ExposureTime:=myexposuretime;
```

禁用一个图像工具

在指定作业禁用特定图像工具。为了防止耗时的作业加载，可能有人会喜欢在一个作业中配置多个配置图像工具。通过启用和禁用当前未使用的工具，可以将处理时间降到最低。

```
CamSetParameter mycamera, "Pattern_1.Tool_Enabled" \BoolVal:=FALSE;
```

下一页继续

查找多个部件

在一个图像中查找多个部件并逐个使用目标。当没有目标剩下时会产生错误 ERR_CAM_NO_MORE_DATA, 请参阅 [第69页的ERROR 处理程序](#)。

```
CamReqImage mycamera \SceneId:=mysceneid;
WHILE targetsavailable DO
  CamGetResult mycamera, mycameratarget \SceneId:=mysceneid;
  TPWrite "Current camera target is: "
        \Pos:=mycameratarget.cframe.trans;
ENDWHILE
```

检查作业是否已经加载

为了节省周期时间, 应该只有在作业尚未加载时才可以被加载到摄像头内存。

```
IF CamGetLoadedJob(mycamera) <> myjob THEN
  CamLoadJob mycamera, myjob;
ENDIF
```

按名称将部件分类

根据部件的名称属性将部件分类。

```
CamGetResult mycamera, mycameratarget;
IF mycameratarget.name = "wrench" THEN
  !Do something with the wrench
ELSEIF mycameratarget.name = "screwdriver" THEN
  !Do something with the screwdriver
ENDIF
```

ERROR 处理程序

```
ERROR
IF ERRNO = ERR_CAM_NO_MORE_DATA THEN
  TPWrite "There are no more targets originating from image with
        scene id "\Num:=mysceneid;
  targetsavailable:=FALSE;
  TRYNEXT;
ENDIF
```

5 配置 Integrated Vision

5.11 开始生产

5.11 开始生产

设置摄像头为运行模式

要开始并进行生产，摄像头必须处于运行模式。摄像头可以手动从 RobotStudio 设置为运行模式。这主要用于测试目的。

推荐的做法是使用 CamSetRunMode 指令从 RAPID 设置运行模式。

启动机器人程序

从 FlexPendant 的生产窗口启动程序。

6 参考信息

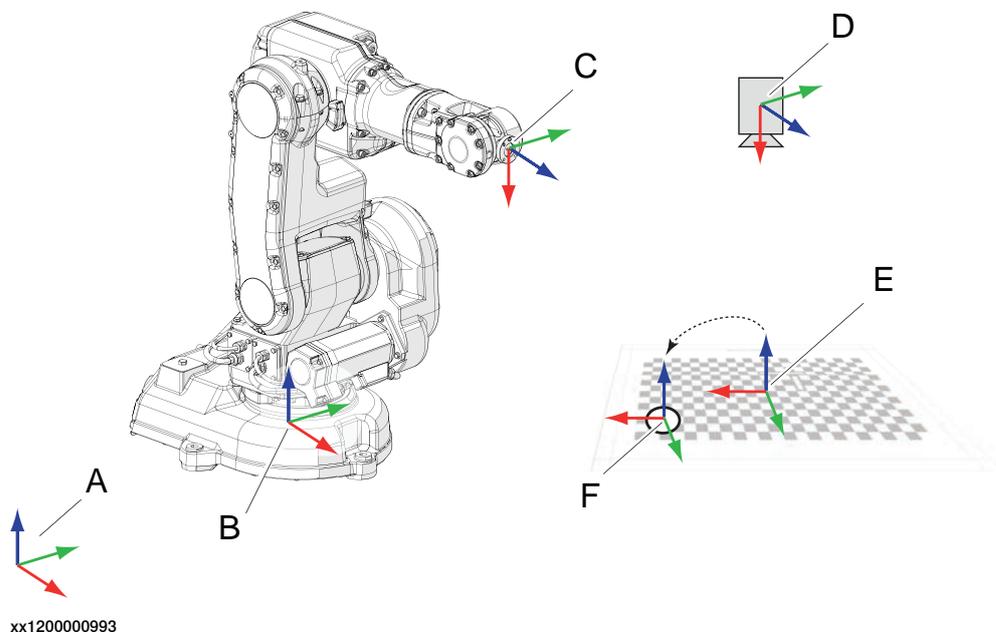
6.1 坐标系统之间的关系

简介

机器人控制器有一些内建的坐标系统，例如 WORLD、BASE、工具、工件等。摄像头也有坐标系统，用于定义图像的原点以及定义图像中距离定位部件的距离(mm)。Integrated vision提供了将摄像头坐标系统与机器人控制器坐标系统同步的方法。有关坐标系统以及如何使用的更多信息，请参阅操作员手册 - 带 FlexPendant 的 IRC5。

坐标系统概述

图像显示了机器人控制器最常用的坐标系统。WORLD 坐标系统并不一定要定义，但是机器人控制器需要知道所有其他坐标系统。



A	WORLD 坐标系统。
B	BASE 坐标系统。
C	工具坐标系 (tool0)。
D	<p>在空中的固定摄像头位置。只有在摄像头由机器人握持时，机器人控制器才会知道摄像头位置。</p> <p> 注意</p> <p>如果摄像头由机器人握持，则机器人必须在每次采集到图像时移动都同一个位置 (robtarger)。</p>
E	<p>Work object - user frame (wobj.uframe).</p> <p>当执行摄像头对机器人校准时同时进行了摄像头框架的匹配。</p>
F	<p>Work object - object frame (wobj.oframe).</p> <p>建议对已定位的部件使用此坐标系统。</p>

下一页继续

6 参考信息

6.1 坐标系统之间的关系

续前页

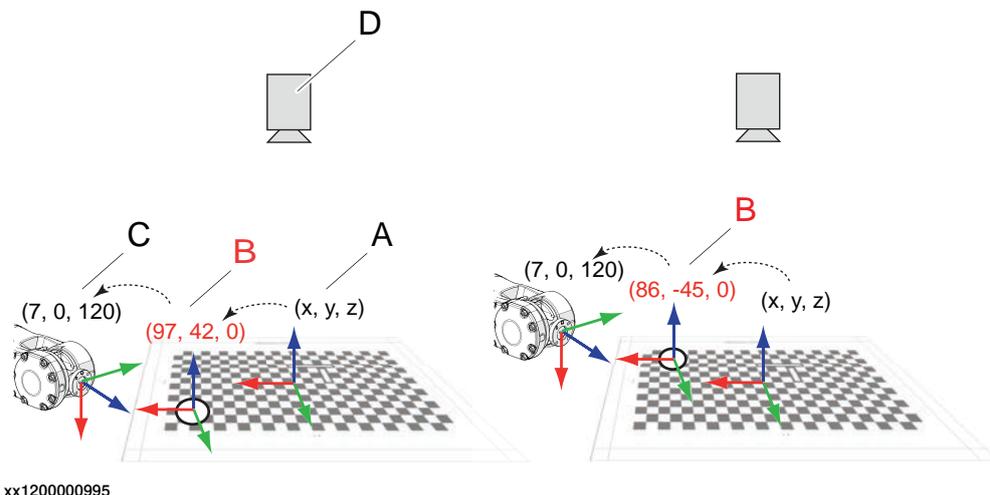
校准的摄像头框架 (work object)

以下插图和示例展示了摄像头坐标系统与机器人控制器坐标系统的同步。



注意

在摄像头对机器人校准后，就不再需要网格，因此可以去除网格。



xx120000995

A	Work object - user frame (<code>wobj.uframe</code>). 当执行摄像头对机器人校准时进行了摄像头框架的匹配。
B	Work object - object frame (<code>wobj.oframe</code>). 此坐标系统用于已定位的部件。
C	机器人的夹持位置 (<code>robtarget</code>)。 (图中使用的是 <code>tool0</code>) 夹持位置是相对于摄像头工件的。
D	在空中的固定摄像头位置 (未由机器人握持)。

示例

左图说明了摄像头定位部件的基本设置。为保证总体效果，此处略去了朝向和角度。

- 摄像头位于空中的固定位置。
- 摄像头使用 10 mm 的校准网格校准以便摄像头能讲图像像素转换为 x 和 y 坐标中的 mm。
坐标原点位于基准标记的交叉处。
- 校准网格根据 *work object - user frame* 对机器人控制器校准。
工件原点也放置在基准标记的交叉处。
- 摄像头定位了一个部件坐标，被发送到机器人控制器。
网格间隔是 10 mm，提供了 x y z 坐标(97, 42, 0)。
- 部件坐标写入 *work object - object frame*。
- 机器人移动捡起部件。
捡取位置已根据 *work object - object frame* 修改。例如，部件上方 120 mm，x 稍微偏移 7 mm，得到坐标 (7, 0, 120)。
- 这就是说无论部件在哪里被定位，捡取位置都可以保持不变。

下一页继续

在右图中，摄像头定位了一个新部件。

- 摄像头位置相同
- 摄像头校准相同
- 相对于部件的捡取位置相同。
- 部件的位置是新的 (86, -45, 0)。
部件坐标写入 *Work object - object frame*。
- 现在，机器人就可以捡取部件了。

6 参考信息

6.2 校准理论

6.2 校准理论

简介

大部分 VGR 应用程序的目标是从摄像头提供定位数据从而引导机器人。这要求图像系统在一个与机器人共享的坐标系统中提供目标。创建图像校准可以分成两步，摄像头校准和摄像头对机器人校准。

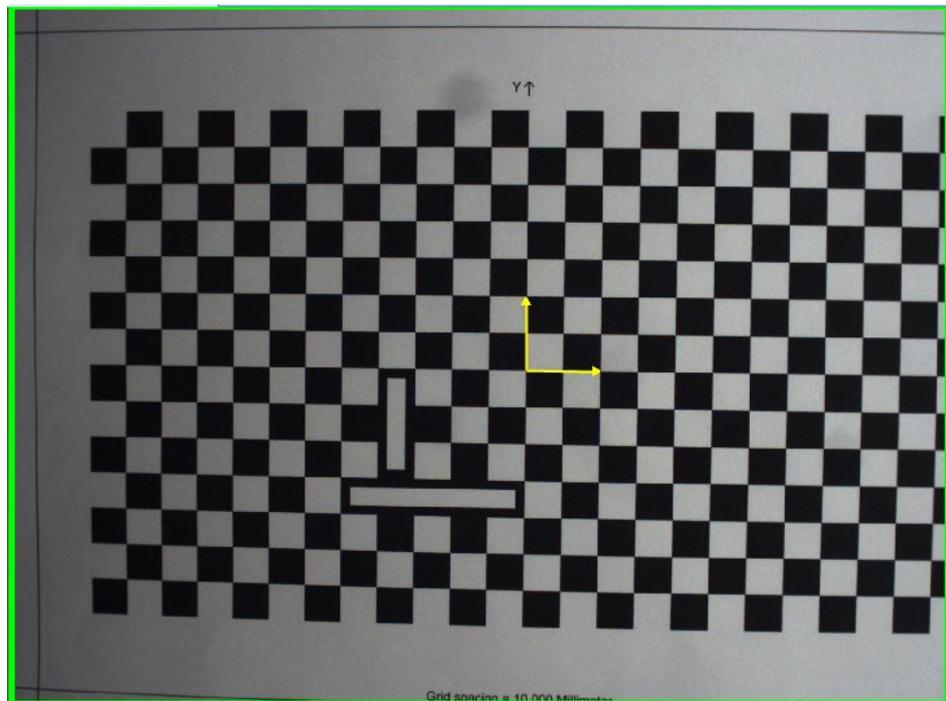
摄像头校准

摄像头校准的目的是计算出用于转换图像中像素坐标为空间物理坐标的转换方法（摄像头框架与期望框架之间的关系）。

Integrated Vision 系统以 2D 方式工作，即所有图像中的部件都必须位于一个校准的平面上，否则结果将会不准确。

Integrated Vision 系统提供了多种方法可以校准摄像头，但最准确最方便的方法是使用棋盘板。这是一个具有已知尺寸的图样，其中包括一个变啊的坐标系统。摄像头与期望直接的关系固定的，这表示摄像头始终必须从与校准时所处位置相同的点采集场景图像。

如果摄像头安装在机器人上，则在生产过程中要拍照机器人必须先移动到校准姿势。摄像头校准后，所有的图像处理输出都以相对于棋盘原点的毫米表示。



xx120000996

有关如何执行摄像头校准的信息，请参阅[第55页的摄像头校准](#)。

摄像头校准方法

下表介绍了Integrated Vision中提供的摄像头校准方法。

X/Y 刻度、边到边、X/Y 边到边以及圆圈校准法在部件或对象的实际尺寸已知时的应用中很有帮助。

下一页继续

只有网格校准提供了可用于指导工件的参照标记。其他校准法也会创建像素-毫米关系，但是较不适合部件定位应用。



注意

刻度校准法不考虑镜头失真，因此为了最大程度保证准确性，请确保图像系统安装的位置与检查面垂直。否则，请考虑使用网格校准来消除失真以更精确的校准图像。

组	描述
X/Y 刻度	当用于计算简单校准的 X 和 Y 刻度值已知时使用。
边到边	当用于计算简单校准的两条边直接的距离已知时使用。
X/Y 边到边	当两对边之间的距离已知时使用。一组处于水平方向，另一组处于垂直方向。结合水平与垂直的测量和尺寸可以创建两个独立的刻度。
圆圈	当一个圆的直径已知时使用。
网格	网格校准法通过采集由点或者棋盘方块组成的网格图案的图像，创建出一张图像区域的“地图”。这种方法随后会将找到的网格特征（点或者方块的交叉处）与网格图案中用户指定的网格间隔关联。此过程可以创建高度精确的计算结果，以后可以用于转换失真的特征点为正确的形状。
导入	导入校准类型用于加载在配置步骤导出的网格校准文件。在加载使用校准的作业时，校准文件 (.cxd) 必须位于摄像头的闪存盘上。

摄像头对机器人校准

摄像头校准后任何图像处理输出都以校准后的摄像头框架表示。要创建机器人和摄像头的公用框架以便机器人能正确解读图像目标的位置，必须告诉机器人校准后的摄像头框架。

实现这一目标的方法是教导与校准后摄像头框架一致的工件。换句话说，如果使用棋盘图案工件将使用与棋盘框架完全相同的位置（和朝向）。工件由两个框架组成，用户框架和对象框架。在本校准中，用户框架将被修改，对象框架将保持不变。

有关如何执行摄像头对机器人校准的信息，请参阅[第57页的摄像头对机器人校准](#)。

6 参考信息

6.3.1 在采用解决方案前先评估性能

6.3 最佳做法

6.3.1 在采用解决方案前先评估性能

简介

当决定部署图像解决方案时，应该屁股是否能实现预期的结果。

确保所需的结果能实现的最安全的方法是执行测试，实验室环境与实际环境越接近，结果越可靠。

当然不可能完全按照实际全面铺开，但是必须考虑以下以及类似关键因素：

- 要求的视野
- 要求的精度
- 可靠性

确认预期

在进入设计阶段前必须确认预期目标，以便采取适当行动达成最佳效果。

精度概念相对容易理解，在很多情况下系统要求的精度很明确。但是机器人引导图像系统的总体精度，甚至仅仅是图像系统本身的总体精度，都很难预见。

虽然单独的图像工具（例如图案识别工具或卡尺）可能有极其优秀的理论精度，但其真实精度高度依赖外部因素（例如要识别的产品和照明条件）。

评估图像精度

总体建议是使用最终生产线上实际要处理的部件或产品来评估图像精度。此外照明条件对于精度和可靠性都非常重要。

照明在很多情况下是实现或破坏最终结果的因素。另外还必须注意到，这在生产线设计完成并开始部署时将会很难修改。

因此当部署图像系统前，先在实验室环境下进行图像测试是有意义的。

安装检查表

作为最佳做法，应鉴别、量化和核实以下要求：

要求	描述
样品	采集用于评价的实际客户产品的好样品和坏样品。
精确度	要求的精度是多少？总数量结合机器人的精度，部件变化、照明等因素的影响。
公差	部件尺寸能变化吗？规则还是不规则？
循环时间	图像系统需要处理时间。视应用不同，这可能影响循环时间，也可能不影响。
部件定位	确保知道摄像头观察物体的观察点，像从侧面观看物体这样的简单做法可能会影响结果。
过程中的变化	除已查证的变量外，是否还会发生其他变化？
照明需求	照明极其重要。应遮挡环境光并应用能突出所需部件特性的光。
物理空间限制	综合各方面的因素，如视野、照明方案和观察点等，一切是否配合良好？

6.3.2 如何安装摄像头

简介

视应用要求和物理限制不同，可以用不同的方式安装摄像头。

一般情况下可以说将摄像头安装在固定结构上会更为有效，除非要求由机器人携带摄像头。

固定摄像头

固定摄像头一般可提供更快的循环时间，因为机器人不用在其路径上停下来采集图像。

固定摄像头的设置和校准一般来说更为容易，因为采集图像时所处的点是固定的。

当将摄像头安装在固定结构上时，重要的一点是不能让摄像头承受会导致运动模糊的震动。

机器人携带的摄像头

如果不能将摄像头安装在固定位置，也可将摄像头安装在操纵器的运动部件上。在这种情况下，一般会将摄像头安装在机器人的工具上，以避免遮挡。每种应用场合的情况不同，而且工具设计和电缆捆扎也各不相同。

将摄像头安置在移动位置时，用户应负责确保摄像头不承受大于摄像头技术规格中规定的机械力。电缆虽为柔软型，但磨损大大取决于电缆布设和机器人的编程路径两个因素。



小心

使用机器人携带的或采用其他方式安装的移动摄像头时，布线合理非常重要。

布设电缆时必须小心谨慎，避免连接器承受机械应力，确保电缆有足够的弯曲半径，尽量减小电缆磨损。还建议在接头处和特别暴露的区域在电缆上额外加装防磨损保护。

6 参考信息

6.3.3 获取精度

6.3.3 获取精度

简介

本节包含优化图像引导机器人系统总体精度的有用技巧。知道哪些因素影响精度对于避免最常见的问题非常有帮助。

训练图像工具

当训练一个图像系统识别或测量一个部件时，首先选择部件的哪些部分是最重要的。这涉及相关特征的可见性及其一致性之间的平衡。

待检测的特征的一致性是非常重要的，影响着多个方面。每个不同部件上相关特征是否出现以及是否外观相似？当部件在视野中移动时是否还能被识别出来？这里照明条件就起到至关重要的作用了。如果没有采用合适的照明，则图像处理将很容易失败，无论是部件以不同的位置出现在摄像头前，或是因为待检测的特征在所有部件上并不都能足够可见。

总之，关键在于选择好要测量部件的部位以及安排好照明，使得待测特征总是能以一致的方式出现。

有关详细信息，请参阅[第79页的获取良好的照明](#)。

视野

很多时候，要执行所需的测量或识别，可能不必将整个部件都容纳在视野中。

最重要的是相关的特征足够清晰且稳定可见。缩小视野不仅可以增加像素-mm的转换比例从而提供更高的精度，同时缩小了需要提供合适照明的范围。此外，应注意摄像头拍摄部件的角度也可能影响结果。

一个通行的规则是，摄像头（如有可能）应该从右侧角度拍摄部件。在无法满足此条件时，非线性校准方式（例如棋盘校准）可以有助于补偿非垂直视角带来的影响。虽然这可以实现精确的将从有倾角的位置获取的坐标进行转换，但并不能自动解决由于角度或镜头效果导致图像失真时，图像工具可能无法稳定运行的问题。

当图像由于光学原因或查看角度导致失真时，作为弥补，可以采用一个能清晰呈现图像的纠正性滤镜。在添加部件检查工具和图像滤镜工具下，选择变形。此工具从网格校准获取结果并使用该信息计算出图像的“不失真”版本。对于后续添加的图像工具，用户可以选择是在修正后的图像还是在原始图像上运行。

棋盘/校准板

正如前面提到的，有多重因素会影响场景的成像，而后续图像处理会对结果精度有很大的影响。

但是如果系统的几何设置或校准与真实环境不一致的，这一切都无关紧要。摄像头坐标系统的建立，通常是通过在摄像头下方放置一个带有原点标记的校准板，并执行校准例行程序完成的。

摄像头的校准结果经常是由机器人以工件的形式作为参照使用，其中用户框架放置在与摄像头坐标系统完全相同的位置。

显然如果工件没有根据原点标记的位置准确定义，则将会损失精度。更重要的是，图像系统报告的结果始终是所识别和测量特征投影在校准板上的投影。这就意味着为了获取准确的结果，要测量的特征应该位于校准板上。

6.3.4 获取良好的照明

简介

必须要注意的是，对于机器图像来说的良好照明与人类眼中的良好照明并不相同。待检查的具体特征应该得到清晰且稳定的照明，以保证在整个视野有效区域中都能得到同样的图像处理结果。

常见问题

最常见的问题来自不均匀的光线分布、反射、阴影以及眩光。消除或至少尽量减少此类不良因素对于图像处理的结果至关重要。选择照明系统时，透明塑料表面的金属物品等发亮的部件通常需要精心考虑。

并没有什么适合所有部件和情况的通用照明技术。但是，了解图像处理的原理以及如何调节照明以提供良好的输入数据，可以大大提高性能和稳定性。

照明技术

可以说控制照明的关键，就是屏蔽不想要的光线以及提供适合应用场合的照明。依靠环境照明（无论是部分还是全部）并不可取。

屏蔽光线的常用技巧是在需要观察的工件位置周围加一个柜子，形成一个照明中心。在柜子内部安装最能突出待测特征的照明类型。另一种遮挡不想要的光线的方法是使用安装在镜头上的光学带通滤镜，同时使用彩光照明。

安装滤镜的目的是只让照明系统提供的相同波长（或颜色）的光通过。

另一种叫“背光”的技术在很多情况下的应用都很成功。这种技术并不照亮部件的正面，光线从产品的边上投射过来从而形成剪影。这可以提供一个简化的白底黑图，通常可以使图像系统工具的处理更容易。

如何选择照明系统

目前来说，确定合适的照明系统的最安全方法，是使用真实的部件并从不同角度照射不同的光源进行实际图像评估。

在评估中，必须考虑照射距离、视野大小、环境光存在情况以及要使用的图像工具等因素。所有这些因素会相互影响所以必须要调整到适合。

6.3.5 图像作业的结构组织

简介

图像系统常常需要在每个图像中或不同的班次之间检测不同的部件。

视当前要求，可使用不同的方法来组织图像作业的结构：

对每个产品或任务创建单独的作业

创建单独的作业可提供结构化的设置，但作业需要几秒钟甚至一分钟时间来加载。每项作业还包含自己的摄像头校准，这在不同的任务要求不同的校准时是有用的。

对所有产品和任务创建一个作业

创建一个包含所有不同部件和生产场景的图像工具的作业。

如果要求系统在单一循环中观看不同的部件，一般使用这种方法。添加工具会使作业文件变大，最终达到最大值。所有工具都随着每次图像采集而执行，这会大大降低图像执行的速度，除非将工具禁用。将所有图像工具放在一个作业中时，一般使用 RAPID 调用打开和关闭各种工具，请参阅[第82页的运行时间内启用和禁用图像工具](#)。

6.3.6 启动程序

描述

上电或重启控制器后，一定要运行一次启动程序。这可确保加载正确的作业，并确保控制器和摄像头处在正确模式。

事件	描述
断电后：	<ul style="list-style-type: none"> • 摄像头将丢失作业。 • 控制器将丢失“输出到 RAPID”配置。
控制器重新启动后：	<ul style="list-style-type: none"> • 控制器将丢失摄像头、程序模式/运行模式的当前状态。 • 控制器将丢失“输出到 RAPID”配置。

示例

```

PROC IV_Init(VAR cameradev cam,string jobname)
  VAR num maxloadtime:=15;

  CamSetProgramMode cam;
  CamLoadJob cam,jobname\MaxTime:=maxloadtime;
  CamSetRunMode cam;

  MoveAbsJ safepos,v100,fine,tool0;

ERROR
  IF ERRNO=ERR_CAM_BUSY THEN
    TPWrite "ERR_CAM_BUSY. Calling RETRY";
    WaitTime 1.0;
    RETRY;
  ELSEIF ERRNO=ERR_CAM_MAXTIME THEN
    TPWrite "ERR_CAM_MAXTIME. Increasing timeout by 10s and calling
      RETRY";
    maxloadtime:=maxloadtime+10;
    WaitTime 1.0;
    RETRY;
  ELSEIF ERRNO=ERR_CAM_NO_PROGMode THEN
    TPWrite "ERR_CAM_NO_PROGMode. Setting camera to program mode
      and calling RETRY";
    WaitTime 1.0;
    CamSetProgramMode cam;
    RETRY;
  ENDIF
ENDPROC

```

6.3.7 运行时间内启用和禁用图像工具

描述

有时我们知道要观看图像中哪种类型的部件，如果是这样，可将目前不需要的图像工具禁用，以缩短摄像头的处理时间。

我们考虑有两个图像工具 `Pattern_1` 和 `Pattern_2` 的示例，它们分别产生类型为 `Item1` 和 `Item2` 的项目。下面的步骤描述如何打开和关闭这两个图像工具。

被禁用的工具仍然用最后一次有效执行得到的值产生结果。为了不使用这些目标，应将它们从 `RAPID` 程序中剔除出去。

示例

此步骤显示如何打开和关闭工具，以缩短摄像头的处理时间。也显示禁用工具时如何处理结果队列。

```
PROC Enable_Disable_Tools(VAR cameradev cam,bool enabletool1,bool
    enabletool2)
VAR cameratarget mycameratarget;
VAR string tool1_propertyname:="Pattern_1.Tool_Enabled";
VAR string tool2_propertyname:="Pattern_2.Tool_Enabled";
VAR string tool1_corresp_item_name:="Item1";
VAR string tool2_corresp_item_name:="Item2";
VAR num maxresulttime:=5;
CamSetProgramMode cam;
CamSetParameter cam,tool1_propertyname\BoolVal:=enabletool1;
CamSetParameter cam,tool2_propertyname\BoolVal:=enabletool2;
CamSetRunMode cam;
CamReqImage cam;
CamGetResult cam,mycameratarget;
!Before using the result, make sure that it originates from an
    enabled tool.
!Disabling tools reduces the processing time, but the latest
    result produced by the tool is still communicated.
IF mycameratarget.name=tool1_corresp_item_name AND enabletool1
    THEN
    TPWrite "Received target of type "+mycameratarget.name+" with
        position "\Pos:=mycameratarget.cframe.trans;
    !Run the robot..
ELSEIF mycameratarget.name=tool2_corresp_item_name AND enabletool2
    THEN
    TPWrite "Received target of type "+mycameratarget.name+" with
        position "\Pos:=mycameratarget.cframe.trans;
    !Run the robot..
ENDIF
!Flush the result queue to get rid of any remaining targets.
CamFlush cam;
ERROR
IF ERRNO=ERR_CAM_BUSY THEN
    TPWrite "ERR_CAM_BUSY. Calling RETRY";
    WaitTime 1.0;
    RETRY;
```

```
ELSEIF ERRNO=ERR_CAM_MAXTIME THEN
    TPWrite "ERR_CAM_MAXTIME. Increasing timeout by 5s and acquiring
           new image";
    CamFlush cam;
    CamReqImage cam;
    maxresulttime:=maxresulttime+5;
    WaitTime 1.0;
    RETRY;
ELSEIF ERRNO=ERR_CAM_NO_RUNMODE THEN
    TPWrite "ERR_CAM_NO_RUNMODE. Setting camera to run mode and
           calling RETRY";
    WaitTime 1.0;
    CamSetRunMode cam;
    RETRY;
ELSEIF ERRNO=ERR_CAM_NO_PROGMODE THEN
    TPWrite "ERR_CAM_NO_PROGMODE. Setting camera to program mode
           and calling RETRY";
    WaitTime 1.0;
    CamSetProgramMode cam;
    RETRY;
ENDIF
ENDPROC
```

6.3.8 避免超出摄像头的存储空间

描述

摄像头的闪存盘存储容量有限。如果机器人单元需要执行很多不同的作业，有时候摄像头可能会无法存储下所有作业。

因此只将必要的作业保存在摄像头上，将其余作业保存在机器人控制器上就是一种方便的好做法。在需要的时候，可以从控制器上传作业到摄像头的闪存盘。

在摄像头和控制器之间移动文件可以使用 RAPID 或 FlexPendant explorer 进行。在后者会将摄像头和控制器驱动器并排显示。

下面的示例显示了如何将系统主目录中的文件移动到摄像头。

另请参阅[第85页的备份摄像头到控制器](#)

示例

```
VAR string campath;  
VAR string controllerpath;  
...  
campath := CamGetName(mycamera) + "://" + "myjob.job";  
controllerpath := "HOME:/myjob.job";  
CopyFile controllerpath, campath;
```

6.3.9 备份摄像头到控制器

描述

在执行机器人控制器的常规备份时，摄像头的内容并不会自动备份。但是以下 RAPID 例行程序会从摄像头复制到控制器的主目录。这可以用来作为执行常规备份时调用的服务例行程序。



提示

相关代码在 RobotStudio 中作为代码片段听。另外也提供了还原文件的代码片段。

示例

```
PROC BackupCamToCtrl(var cameradev cam,bool replaceexistingfiles)
  VAR string ctrldirname:="HOME/IV/";
  VAR dir camdirectory;
  VAR string camdirname;
  VAR string tempfilename;
  VAR string tempcamfilepath;
  VAR string tempctrlfilepath;
  ...
  camdirname:=CamGetName(cam)+"/";
  ctrldirname:=ctrldirname+CamGetName(cam)+"/";
  MakeDir ctrldirname;
  OpenDir camdirectory,camdirname;
  WHILE ReadDir(camdirectory,tempFileName) DO
    tempcamfilepath:=camdirname+tempfilename;
    tempctrlfilepath:=ctrldirname+tempfilename;
    CopyFile tempcamfilepath,tempctrlfilepath;
  ENDWHILE
  CloseDir camdirectory;
ERROR
  IF ERRNO=ERR_FILEEXIST THEN
    IF replaceexistingfiles THEN
      RemoveFile tempctrlfilepath;
      RETRY;
    ELSE
      TRYNEXT;
    ENDIF
  ENDIF
ENDPROC
```

6 参考信息

6.3.10 将不同类型的项目分类

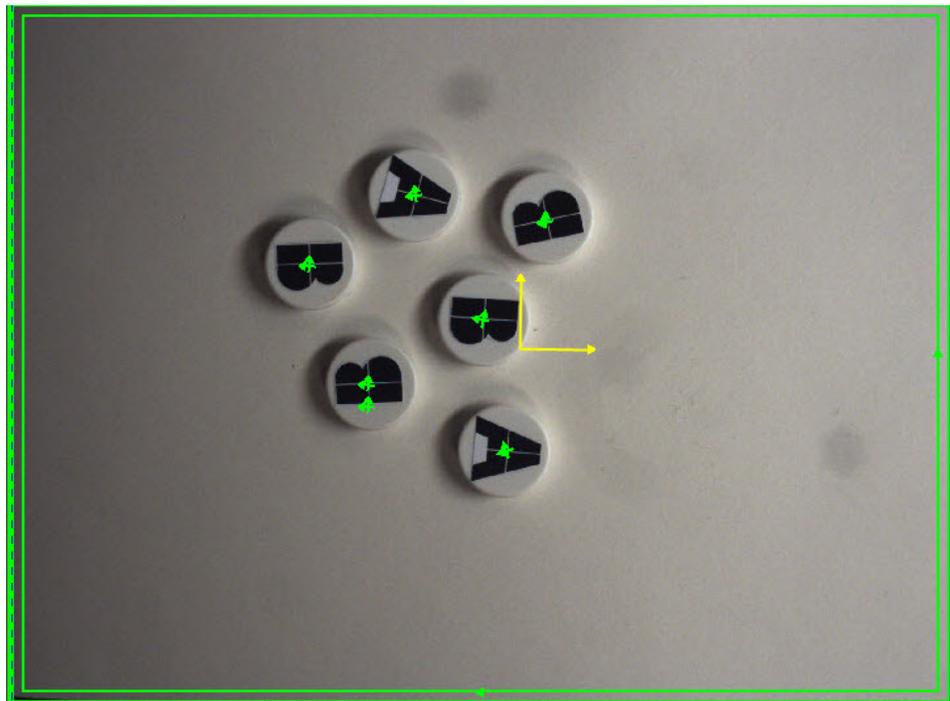
6.3.10 将不同类型的项目分类

描述

有时候需要区分不同类型的产品。以下示例介绍了如何分辨分别印有字母 A 和 B 的两种芯片。

解决方案可以分为 3 个基本步骤：

- 1 先安排一个图像作业，用于区分两种不同芯片类型
- 2 配置输出到 RAPID，以便 RAPID 程序能获取对象的位置和类型。
- 3 将 RAPID 程序编程为获取图像对象，并在操作前检查其类型。



xx1300000190

安排图像作业。

所有，必须教会图像系统分辨两种不同的部件类型。

依靠现有的图像工具，此问题有多种不同解决方法。在本例中，使用了两款不同的部件定位工具 (*PatMax*®)，并针对两种产品类型之间的区分特征字母 A 和 B 进行安排。不包括两种型号上都有的圆圈有两个原因。第一个原因是圆圈在两个部件上是相同的，这使得它们从机器图像的角度来看更相似。第二个原因是圆圈特征占据了型号内容的很大比例却不能提供任何与部件角度相关的信息。这就会导致角度上的不确定性。

配置输出到 RAPID

第二，必须配置输出到 RAPID，以便机器人了解哪些部分需要保留以及可用于操作。要实现这一点，要根据两个项目所代表的部件类型 (A_Chip 和 B_chip) 配置并命名。识别 A 芯片的工具输出连接到 A_Chip 组件，B 芯片也相应操作。

编写 RAPID 程序

最后必须根据所识别的类型编写要执行的 RAPID 程序。

下一页继续

示例

```
...
CamGetResult mycamera, mycamtarget;
IF mycamtarget.name = "A_Chip" THEN
    !Do something with the A-Chip
ELSEIF mycamtarget.name = "B_Chip" THEN
    !Do something with the B-Chip
ENDIF
...
```

6.3.11 寻找同一类型的多个项目

描述

图像任务常常涉及寻找散布在摄像头视野中的同一类型的多个项目。配置这样的任务需要进行图像作业和 RAPID 程序的某些配置。

- 1 一般使用一个能找到多个项目的图像工具 `PatMax[1-10]` 或 `Blob[1-10]`。
- 2 在 **Output to RAPID (输出到 RAPID)** 对话框中，图像工具被链接到单一的项目类型，如称为 *Part_A*。这意味着对每个找到的项目，将产生一个名称属性为 *Part_A* 的摄像头目标，并发送到图像队列。
- 3 RAPID 程序的任务是遍历每个产生的 `cameratarget` 并选取相应的对象。

示例

下面的示例采集一个图像并移动到每个报告的目标处。摄像头作业可以对每个采集的图像产生多个目标。

```
PROC Multiple_Target_Pick(VAR cameradev cam)
  VAR bool continueloop:=TRUE;
  VAR num maxresulttime:=5;
  VAR cameratarget mycameratarget;
  VAR num zoffset:=200;
  CamReqImage cam;
  WHILE continueloop DO
    CamGetResult cam, mycameratarget;
    camwobj.oframe:=mycameratarget.cframe;
    MoveL offs(picktargt,0,0,zoffset), v500, z0, picktool
      \WObj:=camwobj;
    MoveL picktargt, v100, fine, picktool \WObj:=camwobj;
    WaitTime 1.0;
    MoveL offs(picktargt,0,0,zoffset), v500, z0,picktool
      \WObj:=camwobj;
    IF CamNumberOfResults(cam)<1 THEN
      continueloop:=FALSE;
    ENDIF
  ENDWHILE
ERROR
  IF ERRNO=ERR_CAM_BUSY THEN
    TPWrite "ERR_CAM_BUSY. Calling RETRY";
    WaitTime 1.0;
    RETRY;
  ELSEIF ERRNO=ERR_CAM_MAXTIME THEN
    TPWrite "ERR_CAM_MAXTIME. Increasing timeout by 5s and acquiring
      new image";
    CamFlush cam;
    CamReqImage cam;
    maxresulttime:=maxresulttime+5;
    WaitTime 1.0;
    RETRY;
  ELSEIF ERRNO=ERR_CAM_NO_RUNMODE THEN
    TPWrite "ERR_CAM_NO_RUNMODE. Setting camera to run mode and
      calling RETRY";
```

下一页继续

```
WaitTime 1.0;  
CamSetRunMode cam;  
RETRY;  
ENDIF  
ENDPROC
```

6.3.12 总是检查确认图像目标在预期限制内

描述

使用传感器时，在视图接近机器人的位置前，务必要始终确保探测位置处在预期的工作空间内。

训练不当的模型或校准错误都可能会使目标处于无法预料的位置。因此应使用下述步骤（另有代码片段提供）来尽量减少此类问题的风险。

示例

```
IF (CamCheckLimits(mycameratarget.cframe, -100, 100, -100, 100,
                  -90, 90)) THEN
    !
ELSE
    !
ENDIF
FUNC BOOL CamCheckLimits(pose current_pose, num X_min, num X_max,
                        num Y_min, num Y_max, num Angle_min, num Angle_max)
    !
    IF (current_pose.trans.X < X_min) RETURN FALSE;
    IF (current_pose.trans.X > X_max) RETURN FALSE;
    IF (current_pose.trans.Y < Y_min) RETURN FALSE;
    IF (current_pose.trans.Y > Y_max) RETURN FALSE;
    IF (EulerZYX(\Z, current_pose.rot) < Angle_min) RETURN FALSE;
    IF (EulerZYX(\Z, current_pose.rot) > Angle_max) RETURN FALSE;
    RETURN TRUE;
ENDFUNC
```

7 RAPID 参考信息

7.1 指令：

7.1.1 CamFlush - 从摄像头删除集合数据

手册用法

CamFlush 用于清空（删除）摄像头的 cameratarget 集合数据。

基本示例

以下示例介绍了 CamFlush 指令。

例 1

```
CamFlush mycamera;
```

摄像头的 mycamera 集合数据被删除。

变元

```
CamFlush Camera
```

Camera

数据类型：cameradev
摄像头名称。

语法

```
CamFlush  
[ Camera ::= ] < variable (VAR) of cameradev > ';' ;
```

7 RAPID 参考信息

7.1.2 CamGetParameter - 获取不同名称的摄像头参数

7.1.2 CamGetParameter - 获取不同名称的摄像头参数

手册用法

CamGetParameter 用于获取摄像头可能提供的命名的参数与。用户必须知道参数的名称及其返回类型才能获取其值。

基本示例

以下示例介绍了 CamGetParameter 指令。

例 1

```
VAR bool mybool:=FALSE;
...
CamGetParameter mycamera, "Pattern_1.Tool_Enabled_Status"
  \BoolVar:=mybool;
TPWrite "The current value of Pattern_1.Tool_Enabled_Status is: "
  \Bool:=mybool;
```

获得命名的布尔参数 Pattern_1.Tool_Enabled_Status 并将值写在 FlexPendant 上。

变元

```
CamGetParameter Camera ParName [\Num] | [\Bool] | [\Str]
```

Camera

数据类型：cameradev
摄像头名称。

ParName

Parameter Name
数据类型：string
摄像头中参数的名称。

[\NumVar]

数据类型：num
变量 (VAR) 用于存储所获取的数据对象的数字值。

[\BoolVar]

数据类型：bool
变量 (VAR) 用于存储所获取的数据对象的布尔值（真假值）。

[\StrVar]

数据类型：string
变量 (VAR) 用于存储所获取的数据对象的布尔值（真假值）。

程序执行

在此指令被执行时会直接读取指定参数并返回值。
如果指令被用于从图像分析读取结果，请在获取数据前确保摄像头已经完成图像处理。

下一页继续

错误处理

可能会产生下列可恢复错误。错误可以由错误处理程序处理。系统变量 `ERRNO` 将设置为：

名称	错误原因
<code>ERR_CAM_BUSY</code>	摄像头正忙于处理其他请求，无法执行当前命令。
<code>ERR_CAM_COM_TIMEOUT</code>	与摄像头通信错误。摄像头可能已断开。
<code>ERR_CAM_GET_MISMATCH</code>	用 <code>CamGetParameter</code> 指令从摄像头获取的参数的数据类型错误。

语法

```
CamGetParameter
  [ Camera ::= ] < variable (VAR) of cameradev > ','
  [ ParName ::= ] < expression (IN) of string >
  [ '\NumVar ::= ' < variable (VAR) of num > ]
  | [ '\BoolVar ::= ' < variable (VAR) of bool > ]
  | [ '\StrVar ::= ' < variable (VAR) of string > ] ';'

```

7 RAPID 参考信息

7.1.3 CamGetResult - 从集合获取摄像头目标

7.1.3 CamGetResult - 从集合获取摄像头目标

手册用法

`CamGetResult` (摄像头获取结果) 用于从图像结果集合获取摄像头目标。

基本示例

以下示例介绍了 `CamGetResult` 指令。

例 1

```
VAR num mysceneid;  
VAR cameratarget mycamtarget;  
...  
CamReqImage mycamera \SceneId:= mysceneid;  
CamGetResult mycamera, mycamtarget \SceneId:= mysceneid;
```

命令摄像头 `mycamera` 采集图像。使用 `SceneId` 获取从图像生成的图像结果。

变元

```
CamGetResult Camera CamTarget [\SceneId] [\MaxTime]
```

Camera

数据类型: `cameradev`

摄像头名称。

CamTarget

摄像头目标

数据类型: `cameratarget`

作为图像结果保存位置的变量。

[\SceneId]

场景识别

数据类型: `num`

`SceneId` 是一个识别程序, 指定 `cameratarget` 是从哪个图像生成的。

[\MaxTime]

最大时间

数据类型: `num`

程序执行可以等待的最大时间 (以秒为单位)。允许的最大值是 120 秒。

程序执行

`CamGetResult` 从图像结果集合获取摄像头目标。如果没有使用 `SceneId` 或 `MaxTime`, 则不会获取结果, 指令将永远停止。如果在 `CamGetResult` 中使用了 `SceneId`, 则结果将在 `CamReqImage` 指令后生成。

`SceneId` 仅在已经从指令 `CamReqImage` 请求了图像时可用。如果图像是由外部 I/O 信号生成的, 则 `SceneId` 不能在指令 `CamGetResult` 中使用。

下一页继续

错误处理

可能会产生下列可恢复错误。错误可以由错误处理程序处理。系统变量 `ERRNO` 将设置为：

名称	错误原因
<code>ERR_CAM_BUSY</code>	摄像头正忙于处理其他请求，无法执行当前命令。
<code>ERR_CAM_MAXTIME</code>	在超时时间不能获取任何结果。
<code>ERR_CAM_NO_MORE_DATA</code>	不能为已经使用的 <code>SceneId</code> 获取更多图像结果，否则在超时时间无法获取结果。

语法

```
CamGetResult
[ Camera ::= ] < variable (VAR) of cameradev > ','
[ CamTarget ::= ] < variable (VAR) of CameraTarget >
[ '\SceneId ::= ' < expression (IN) of num > ]
[ '\MaxTime ::= ' < expression (IN) of num > ] ';'

```

7 RAPID 参考信息

7.1.4 CamLoadJob -加载摄像头任务到摄像头

7.1.4 CamLoadJob -加载摄像头任务到摄像头

手册用法

CamLoadJob (摄像头加载作业) 加载摄像头任务, *job*, 说明了曝光参数、校准以及要应用的图像工具。

基本示例

以下示例介绍了 CamLoadJob 指令。

例 1

```
CamSetProgramMode mycamera;  
CamLoadJob mycamera, "myjob.job";  
CamSetRunMode mycamera;
```

作业 myjob 加载到名为 mycamera 的摄像头。

变元

```
CamLoadJob Camera JobName [\KeepTargets] [\MaxTime]
```

Camera

数据类型: cameradev

摄像头名称。

Name

数据类型: string

加载到摄像头的作业名称。

[\KeepTargets]

数据类型: switch

此参数用于指定是否保留摄像头产生的任何现有摄像头目标。

[\MaxTime]

数据类型: num

程序执行可以等待的最大时间 (以秒为单位)。允许的最大值是 120 秒。

程序执行

CamLoadJob 的执行将会等到作业加载完毕或经过超时错误失败。如果使用可选参数 KeepTargets, 则保留指定摄像头的集合数据。默认的操作是删除 (清空) 就集合数据。

错误处理

可能会产生下列可恢复错误。错误可以由错误处理程序处理。系统变量 ERRNO 将设置为:

名称	错误原因
ERR_CAM_BUSY	摄像头正忙于处理其他请求, 无法执行当前命令。
ERR_CAM_COM_TIMEOUT	与摄像头通信错误。摄像头可能已断开。
ERR_CAM_MAXTIME	摄像头作业不会在超时时间加载。
ERR_CAM_NO_PROGMode	摄像头未处于编程模式

下一页继续

限制

当摄像头设置为编程模式时，才可以执行 CamLoadJob。使用指令 CamSetProgramMode 可将摄像头设置为编程模式。
为了能加载作业，作业文件必须存储在摄像头的闪存盘。

语法

```
CamLoadJob
  [ Camera ':= ' ] < variable (VAR) of cameradev > ', '
  [ JobName ':= ' ] <expression (IN) of string >
  [ '\KeepTargets ]
  [ '\MaxTime ':= ' <expression (IN) of num>'];'
```

7 RAPID 参考信息

7.1.5 CamReqImage - 命令摄像头采集图像

7.1.5 CamReqImage - 命令摄像头采集图像

手册用法

CamReqImage (摄像头请求图像)命令摄像头采集图像。

基本示例

以下示例介绍了 CamReqImage 指令。

例 1

```
CamReqImage mycamera;
```

命令摄像头 mycamera 采集图像。

变元

```
CamReqImage Camera [\SceneId] [\KeepTargets] [\AwaitComplete]
```

Camera

数据类型：cameradev

摄像头名称。

[\SceneId]

场景识别

数据类型：num

可选参数 SceneId 是所采集图像的一个标识符。这是由 CamReqImage 加上可选变量 SceneId 执行生成的。标识符是一个 1 到 8388608 之间的整数。如果没有使用 SceneId, 则标识符值设置为 0。

[\KeepTargets]

数据类型：switch

此参数用于指定是否保留指定摄像头的旧集合数据。

[\AwaitComplete]

数据类型：switch

如果指定可选参数 \AwaitComplete, 则指令等待, 直至已经收到来自图像的结果。如果未产生任何结果, 例如, 因为图像中没有一部分, 则会产生错误 ERR_CAM_REQ_IMAGE。

当使用 \AwaitComplete 时, 必须将相机触发类型设置为外部。

程序执行

CamReqImage 用于命令指定摄像头采集图像。如果使用了可选参数 SceneId, 则所采集图像的可用图像结果使用该指令生成的唯一数字标记。

如果使用可选参数 KeepTargets, 则保留指定摄像头的旧集合数据。默认的操作是删除 (清空) 所有旧集合数据。

下一页继续

错误处理

可能会产生下列可恢复错误。错误可以由错误处理程序处理。系统变量 `ERRNO` 将设置为：

名称	错误原因
<code>ERR_CAM_BUSY</code>	摄像头正忙于处理其他请求，无法执行当前命令。
<code>ERR_CAM_COM_TIMEOUT</code>	与摄像头通信错误。摄像头可能已断开。
<code>ERR_CAM_NO_RUNMODE</code>	摄像头未处于运行模式
<code>ERR_CAM_REQ_IMAGE</code>	相机无法生成任何图像结果。

限制

当摄像头设置为运行模式时，才可以执行 `CamReqImage`。使用指令 `CamSetRunMode` 可将摄像头设置为运行模式。

语法

```
CamReqImage
  [ Camera ':= ' ] < variable (VAR) of cameradev > ', '
  [ '\SceneId ' := ' < variable (VAR) of num > ]
  [ '\KeepTargets ]
  [ '\AwaitComplete ]';'
```

7 RAPID 参考信息

7.1.6 CamSetExposure - 设置具体摄像头的的数据

7.1.6 CamSetExposure - 设置具体摄像头的的数据

手册用法

CamSetExposure (摄像头设置曝光) 设置具体摄像头的的数据并可以实现根据环境照明条件调整图像参数。

基本示例

以下示例介绍了 CamSetExposure 指令。

例 1

```
CamSetExposure mycamera \ExposureTime:=10;
```

命令摄像头 mycamera 将曝光时间修改为 10 ms。

变元

```
CamSetExposure Camera [\ExposureTime] [\Brightness] [\Contrast]
```

Camera

数据类型：cameradev
摄像头名称。

[\ExposureTime]

数据类型：num
如果使用了本可选参数，则摄像头的曝光时间会更新。该值以毫秒 (ms) 为单位。

[\Brightness]

数据类型：num
如果使用了本可选参数，则将更新摄像头的亮度设置。其值通常以 0 到 1 之间的刻度表示。

[\Contrast]

数据类型：num
如果使用了本可选参数，则将更新摄像头的对比度设置。其值通常以 0 到 1 之间的刻度表示。

程序执行

如果具体摄像头的对应参数可能更新，则此指令更新曝光时间、亮度和对比度。如果摄像头不支持某个设置，则会向用户显示错误消息，程序停止执行。

错误处理

可能会产生下列可恢复错误。错误可以由错误处理程序处理。系统变量 ERRNO 将设置为：

名称	错误原因
ERR_CAM_COM_TIMEOUT	与摄像头通信错误。摄像头可能已断开。

语法

```
CamSetExposure  
[ Camera ':= ' ] < variable (VAR) of cameradev > ','
```

下一页继续

```
[ '\ExposureTime :=' < variable (IN) of num > ]  
[ '\Brightness :=' < variable (IN) of num > ]  
[ '\Contrast :=' < variable (IN) of num > ] ;'
```

7 RAPID 参考信息

7.1.7 CamSetParameter - 设置不同名称的摄像头参数

7.1.7 CamSetParameter - 设置不同名称的摄像头参数

手册用法

`CamSetParameter` 用于设置摄像头可能接触的不同名称的摄像头参数。在此命令中可以在运行时修改摄像头中的不同参数。用户须知道参数名称及其类型才能设置其值。

基本示例

以下示例介绍了 `CamSetParameter` 指令。

例 1

```
CamSetParameter mycamera, "Pattern_1.Tool_Enabled" \BoolVal:=FALSE;  
CamSetRunMode mycamera;
```

在本例中名为 "Pattern_1.Tool_Enabled" 的参数被设为假，这表示在采集到图像时不应执行指定的图像工具。

这将使图像工具的执行更快。但是，工具仍然用最后一次有效执行得到的值产生结果。为了不使用这些目标，应将它们从 RAPID 程序中剔除出去。

变元

```
CamSetParameter Camera ParName [\Num] | [\Bool] | [\Str]
```

Camera

数据类型：cameradev

摄像头名称。

ParName

数据类型：string

摄像头中参数的名称。

[\NumVal]

数据类型：num

摄像头的数值在参数 `ParName` 设置名称时设置。

[\BoolVal]

数据类型：bool

摄像头的布尔值在参数 `ParName` 设置名称时设置。

[\StrVal]

数据类型：string

摄像头的字符串值在参数 `ParName` 设置名称时设置。

错误处理

可能会产生下列可恢复错误。错误可以由错误处理程序处理。系统变量 `ERRNO` 将设置为：

名称	错误原因
<code>ERR_CAM_BUSY</code>	摄像头正忙于处理其他请求，无法执行当前命令。
<code>ERR_CAM_COM_TIMEOUT</code>	与摄像头通信错误。摄像头可能已断开。

下一页继续

名称	错误原因
ERR_CAM_SET_MISMATCH	使用命令 <code>CamSetParameter</code> 写入摄像头的参数数据类型错误, 或者其值超出范围。

语法

```
CamSetParameter
[ Camera ::= ] < variable (VAR) of cameradev > ','
[ ParName ::= ] < expression (IN) of string >
[ '\NumVal ::= ' < expression (IN) of num > ]
| [ '\BoolVal ::= ' < expression (IN) of bool > ]
| [ '\StrVal ::= ' < expression (IN) of string > ] ';' ;
```

7 RAPID 参考信息

7.1.8 CamSetProgramMode - 命令摄像头进入编程模式

7.1.8 CamSetProgramMode - 命令摄像头进入编程模式

手册用法

CamSetProgramMode (摄像头设置编程模式) 命令摄像头进入编程模式（离线）。

基本示例

以下示例介绍了 CamSetProgramMode 指令。

例 1

```
CamSetProgramMode mycamera;  
CamLoadJob mycamera, "myjob.job";  
CamSetRunMode mycamera;  
...
```

首先，将摄像头改为编程模式。然后加载 myjob 到摄像头。然后命令摄像头进入运行模式。

变元

```
CamSetProgramMode Camera
```

Camera

数据类型：cameradev
摄像头名称。

程序执行

当使用 CamSetProgramMode 指令命令摄像头进入编程模式时，可以修改设置并加载作业到摄像头。

错误处理

可能会产生下列可恢复错误。错误可以由错误处理程序处理。系统变量 ERRNO 将设置为：

名称	错误原因
ERR_CAM_BUSY	摄像头正忙于处理其他请求，无法执行当前命令。
ERR_CAM_COM_TIMEOUT	与摄像头通信错误。摄像头可能已断开。

语法

```
CamSetProgramMode  
[ Camera '[:=' ] < variable (VAR) of cameradev > '];'
```

7.1.9 CamSetRunMode - 命令摄像头进入运行模式

手册用法

CamSetRunMode (摄像头设置的运行模式) 命令摄像头进入运行模式 (在线), 并将控制器更新为“输出到 RAPID”的当前配置。

基本示例

以下示例介绍了 CamSetRunMode 指令。

例 1

```
CamSetProgramMode mycamera;
CamLoadJob mycamera, "myjob.job";
...
CamSetRunMode mycamera;
```

首先, 将摄像头改为编程模式。然后加载 myjob 到摄像头。然后使用 CamSetRunMode 指令命令摄像头进入运行模式。

变元

```
CamSetRunMode Camera
```

Camera

数据类型: cameradev

摄像头名称。

程序执行

在使用 CamSetRunMode 命令摄像头进入运行模式时, 可以开始采集图像。

错误处理

可能会产生下列可恢复错误。错误可以由错误处理程序处理。系统变量 ERRNO 将设置为:

名称	错误原因
ERR_CAM_BUSY	摄像头正忙于处理其他请求, 无法执行当前命令。
ERR_CAM_COM_TIMEOUT	与摄像头通信错误。摄像头可能已断开。

语法

```
CamSetRunMode
[ Camera ':= ' ] < variable (VAR) of cameradev > ' ; '
```

7 RAPID 参考信息

7.1.10 CamStartLoadJob - 开始加载摄像头任务到摄像头

7.1.10 CamStartLoadJob - 开始加载摄像头任务到摄像头

手册用法

CamStartLoadJob 将开始加载作业到摄像头，然后执行将在下一个指令继续进行。当加载进行时，其他指令可以并行执行。

基本示例

以下示例介绍了 CamStartLoadJob 指令。

例 1

```
...
CamStartLoadJob mycamera, "myjob.job";
MoveL p1, v1000, fine, tool2;
CamWaitLoadJob mycamera;
CamSetRunMode mycamera;
CamReqImage mycamera;
...
```

首先开始加载作业到摄像头，在加载进行时，执行了一个移动到位置 p1 的操作。当移动就绪后，加载也完成了，图像也采集好了。

变元

```
CamStartLoadJob Camera Name [\KeepTargets]
```

Camera

数据类型：cameradev
摄像头名称。

Name

数据类型：string
加载到摄像头的作业名称。

[\KeepTargets]

数据类型：switch
此参数用于指定是否保留指定摄像头的旧集合数据。

程序执行

CamStartLoadJob 的执行将会命令开始加载，然后无需等待加载完成直接继续下一个指令，如果使用了可选参数 \KeepTargets，则不会删除指定摄像头的旧集合数据。默认操作是删除（清空）指定摄像头的旧集合数据。

错误处理

可能会产生下列可恢复错误。错误可以由错误处理程序处理。系统变量 ERRNO 将设置为：

名称	错误原因
ERR_CAM_BUSY	摄像头正忙于处理其他请求，无法执行当前命令。

下一页继续

限制

当摄像头设置为编程模式时，才可以执行 CamStartLoadJob。使用指令 CamSetProgramMode 可将摄像头设置为编程模式。

当作业的加载在执行中时，无法使用任何其他指令或函数访问对应的摄像头。后续的摄像头指令或函数必须是一个 CamWaitLoadJob 指令。

为了能加载作业，作业文件必须存储在摄像头的闪存盘。

语法

```
CamStartLoadJob
  [ Camera ':= ' ] < variable (VAR) of cameradev > ', '
  [ Name ':= ' ] < expression (IN) of string >
  [ '\KeepTargets ]';'
```

7 RAPID 参考信息

7.1.11 CamWaitLoadJob – 等待摄像头任务加载完毕

7.1.11 CamWaitLoadJob – 等待摄像头任务加载完毕

手册用法

CamWaitLoadJob (摄像头等待加载作业) 将会等待作业加载到摄像头完成。

基本示例

以下示例介绍了 CamWaitLoadJob 指令。

例 1

```
...
CamStartLoadJob mycamera, "myjob.job";
MoveL p1, v1000, fine, tool2;
CamWaitLoadJob mycamera;
CamSetRunMode mycamera;
CamReqImage mycamera;
...
```

首先开始加载作业到摄像头，在加载进行时，执行了一个移动到位置 p1 的操作。当移动就绪后，加载也完成了，图像也采集好了。

变元

CamWaitLoadJob Camera

Camera

数据类型：cameradev
摄像头名称。

错误处理

可能会产生下列可恢复错误。错误可以由错误处理程序处理。系统变量 ERRNO 将设置为：

名称	错误原因
ERR_CAM_COM_TIMEOUT	与摄像头通信错误。摄像头可能已断开。

限制

当摄像头设置为编程模式时，才可以执行 CamWaitLoadJob。使用指令 CamSetProgramMode 可将摄像头设置为编程模式。

当作业的加载在执行中时，无法使用任何其他指令或函数访问对应的摄像头。后续的摄像头指令或函数必须是一个 CamWaitLoadJob 指令。

语法

```
CamWaitLoadJob
[ Camera ':' = ] < variable (VAR) of cameradev > ';' ;
```

7.2 函数

7.2.1 CamGetExposure - 获取具体摄像头的数据

手册用法

CamGetExposure (Camera Get Exposure) 是一个读取当前摄像头设置的函数。使用此函数与 CamSetExposure 指令可以在运行时根据环境调整摄像头图像。

基本示例

以下示例介绍了 CamGetExposure 函数。

例 1

```
VAR num exposuretime;
...
exposuretime:=CamGetExposure(mycamera \ExposureTime);
IF exposuretime = 10 THEN
  CamSetExposure mycamera \ExposureTime:=9.5;
ENDIF
```

如果当前曝光时间设置为 10 ms，则命令摄像头 mycamera 将曝光时间改为 9.5 ms。

返回值

数据类型：num

从摄像头以数值方式返回的曝光时间、亮度或对比度中的某个设置。

变元

```
CamGetExposure (Camera [\ExposureTime] | [\Brightness] |
[\Contrast])
```

Camera

数据类型：cameradev

摄像头名称。

[\ExposureTime]

数据类型：num

返回摄像头曝光时间。其值以毫秒 (ms) 为单位。

[\Brightness]

数据类型：num

返回摄像头的亮度设置。

[\Contrast]

数据类型：num

返回摄像头的对比度设置。

下一页继续

7 RAPID 参考信息

7.2.1 CamGetExposure - 获取具体摄像头的数据

续前页

错误处理

可能会产生下列可恢复错误。错误可以由错误处理程序处理。系统变量 `ERRNO` 将设置为：

名称	错误原因
<code>ERR_CAM_BUSY</code>	摄像头正忙于处理其他请求，无法执行当前命令。
<code>ERR_CAM_COM_TIMEOUT</code>	与摄像头通信错误。摄像头可能已断开。

语法

```
CamGetExposure '('  
  [ Camera ':=' ] < variable (VAR) of cameradev >  
  ['\ExposureTime]  
  | ['\Brightness]  
  | ['\Contrast] ')'
```

返回值的数据类型是 `num` 的函数。

7.2.2 CamGetLoadedJob - 获取所加载摄像头任务的名称

手册用法

CamGetLoadedJob (摄像头获取加载作业) 是一个从摄像头读取当前所加载作业名称并以字符串形式返回的函数。

基本示例

以下示例介绍了 CamGetLoadedJob 函数。

例 1

```
VAR string currentjob;
...
currentjob:=CamGetLoadedJob(mycamera);
IF CurrentJob = "" THEN
  TPWrite "No job loaded in camera "+CamGetName(mycamera);
ELSE
  TPWrite "Job "+CurrentJob+" is loaded in camera "
    "+CamGetName(mycamera);
ENDIF
```

在 FlexPendant 上写入加载的作业名称。

返回值

数据类型：string
指定摄像头当前加载的作业名称。

变元

CamGetLoadedJob (Camera)

Camera

数据类型：cameradev
摄像头名称。

程序执行

CamGetLoadedJob 函数从摄像头获取当前加载的作业名称。如果没有作业加载摄像头，则返回空字符串。

错误处理

可能会产生下列可恢复错误。错误可以由错误处理程序处理。系统变量 ERRNO 将设置为：

名称	错误原因
ERR_CAM_BUSY	摄像头正忙于处理其他请求，无法执行当前命令。
ERR_CAM_COM_TIMEOUT	与摄像头通信错误。摄像头可能已断开。

语法

```
CamGetLoadedJob '('
  [ Camera ':=' ] < variable (VAR) of cameradev > ')'
```

返回值的数据类型是 string 的函数。

7 RAPID 参考信息

7.2.3 CamGetName - 获取所使用摄像头的名称

7.2.3 CamGetName - 获取所使用摄像头的名称

手册用法

CamGetName (Camera Get Name) 用于获取摄像头的配置名称。

基本示例

以下示例介绍了 CamGetName 函数。

例 1

```
...
logcameraname camera1;
CamReqImage camera1;
...
logcameraname camera2;
CamReqImage camera2;
...
PROC logcameraname(VAR cameradev camdev)
  TPWrite "Now using camera: "+CamGetName(camdev);
ENDPROC
```

此步骤记录当前使用的摄像头的名称到 FlexPendant。

返回值

数据类型: string

当前使用的摄像头的名称以字符串返回。

变元

CamGetName(Camera)

Camera

数据类型: cameradev

摄像头名称。

语法

```
CamGetName( '('
  [ Camera ':=' ] < variable (VAR) of cameradev > ')'
```

返回值的数据类型是 string 的函数。

7.2.4 CamNumberOfResults - 获取可用结果的数量

手册用法

CamNumberOfResults (Camera Number of Results) 是读取获取可用结果数量，并以数值形式返回的函数。

基本示例

以下示例介绍了 CamNumberOfResults 函数。

例 1

```
VAR num foundparts;
...
CamReqImage mycamera;
WaitTime 1;
FoundParts := CamNumberOfResults(mycamera);
TPWrite "Number of identified parts in the camera image:
        "\Num:=foundparts;
```

采集图像。等待图像处理完成，在本例中为 1 秒。读取识别的部件并将其写入 FlexPendant。

返回值

数据类型：num

返回指定摄像头集合中结果的数量。

变元

CamNumberOfResults (Camera [\SceneId])

Camera

数据类型：cameradev

摄像头名称。

[\SceneId]

场景识别

数据类型：num

SceneId 是一个标识符，指定从哪个图像读取识别的部件的编号。

程序执行

CamNumberOfResults 是读取可用图像结果数量，并将其以数值方式返回的函数。可以用于遍历所有可用结果。

此功能被执行时直接返回队列级别。如果在请求图片后直接执行功能，则结果为 0，因为摄像头尚未完成图片处理。

错误处理

可能会产生下列可恢复错误。错误可以由错误处理程序处理。系统变量 ERRNO 将设置为：

名称	错误原因
ERR_CAM_BUSY	摄像头正忙于处理其他请求，无法执行当前命令。

下一页继续

7 RAPID 参考信息

7.2.4 CamNumberOfResults - 获取可用结果的数量

续前页

语法

```
CamNumberOfResults '('  
  [ Camera ':=' ] < variable (VAR) of cameradev >  
  [ '\SceneId ':=' < expression (IN) of num > ] ')'
```

返回值的数据类型是 num 的函数。

7.3 数据类型

7.3.1 cameradev - 摄像头设备

手册用法

cameradev (摄像头设备)用于定义可以从 RAPID 程序控制和访问的不同摄像头设备。数据类型 cameradev 用于与摄像头的指令和函数通信。摄像头设备的名称在系统参数中定义，因此切勿在程序中定义。

描述

cameradev 类数据仅包含对摄像头设备的引用。

限制

cameradev 类数据部的在程序中定义。但是如果定义了，则会任何引用此 cameradev 的指令或函数执行时立即显示一个错误消息。另外，此数据类型也可以在声明例行程序时用作参数。

预定义数据

系统参数中定义的所有摄像头在每个程序任务中预定义。

基本示例

以下示例介绍了 cameradev 数据类型。

例 1

```
CamLoadJob mycamera, "myjob.job";
```

特征

cameradev 是一个非值数据类型。这就表示此类数据不允许面向数值的操作。

7 RAPID 参考信息

7.3.2 cameratarget - 摄像头数据

7.3.2 cameratarget - 摄像头数据

手册用法

`cameratarget` 用于从摄像头图像交换图像数据到 RAPID 程序。

描述

`cameratarget` 类的数据是一个用户定义的集合数据，可以用于设置为从摄像头图像交换数据到 RAPID 程序。

此数据有一系列组件，可根据当前图像应用的具体需要设置。`cframe` 组件旨在传输对象的位置信息，数值以及字符串用于保存检查数据。

组件

此数据类型有以下组件：

`name`

数据类型：`string`

`cameratarget` 的名称标识符。

`cframe`

current frame

数据类型：`pose`

用于存储位置数据，通常用于通过修改工件引导机器人。

`val1`

value 1

数据类型：`num`

用于存储测量结果等数值输出。

...

`val5`

value 5

数据类型：`num`

用于存储测量结果等数值输出。

`string1`

数据类型：`string`

用于存储检查或识别输出等数值图像输出。

`string2`

数据类型：`string`

用于存储检查或识别输出等数值图像输出。

`type`

数据类型：`num`

摄像头对象的数字标识符。与 `name` 组件作用类似。

下一页继续

cameraname

数据类型 : string

摄像头名称。

sceneid

scene identification

数据类型 : num

用于产生 cameratarget 的图像的唯一标识符。

基本示例

以下示例介绍了 cameratarget 数据类型。

例 1

```

VAR cameratarget target1;
...
wobjmycamera.oframe := target1.cframe;
MoveL pickpart, v100, fine, mygripper \WObj:= wobjmycamera;

```

cframe 坐标转换分配到工件的对象框架。robtarget pickpart 此前已经在工件框架内微调为正确的捡取位置。

结构

```

< dataobject of cameratarget >
  < name of string >
  < cframe of pose >
    < trans of pos >
    < rot of orient >
  < val1 of num >
  < val2 of num >
  < val3 of num >
  < val4 of num >
  < val5 of num >
  < string1 of string >
  < string2 of string >
  < type of num >
  < cameraname of string >
  < sceneid of num >

```

此页刻意留白

索引

C

cameradev, 115
 cameratarget, 116
 CamFlush, 91
 CamGetExposure, 109
 CamGetLoadedJob, 111
 CamGetName, 112
 CamGetParameter, 92
 CamGetResult, 94
 CamLoadJob, 96
 CamNumberOfResults, 113
 CamReqImage, 98
 CamSetExposure, 100
 CamSetParameter, 102
 CamSetProgramMode, 104
 CamSetRunMode, 105
 CamStartLoadJob, 106
 CamWaitLoadJob, 108
 Cognex EasyBuilder®, 13, 16
 Cognex In-Sight®, 13, 16

I

I/O, 64
 IP 地址, 45

O

Options (选项) 对话框, 32

R

RAPID 程序, 65

V

vc_network_definition, 46
 VGR, 16

上

上下文窗口, 21, 31

代

代码片段, 16, 65

作

作业, 16

保

保护作业, 35

入

入门, 15

固

固件, 46

图

图像作业, 16
 图像工具, 58

在

在线帮助, 22

坐

坐标系统, 71

基

基准, 16, 57

备

备份, 85

夹

夹持部件, 67

子

子网, 45

安

安全, 11
 安装
 RobotStudio, 19
 RobotWare, 19
 硬件, 17
 软件, 19

定

定位工具, 58

开

开始生产, 70

快

快捷键, 27, 34

指

指点工具, 16

控

控制器浏览器, 21, 26

摄

摄像头模拟器, 32, 47
 摄像头网络, 43

数

数据映射, 61

断

断开摄像头, 23, 26, 44

日

日期, 45

时

时间, 45

智

智能摄像头, 16

术

术语, 22
 术语列表, 16
 术语表, 16

校

校准
 国际象棋盘, 16
 图像, 16
 摄像头, 16, 55, 74
 摄像头到机器人, 16
 摄像头对机器人, 57, 75
 校准的摄像头框架, 72

检

检查工具, 58
 检查表, 15

模

模拟器, 摄像头, 32, 47

照

照明, 79

状

状态条, 27

用

用户界面

FlexPendant, 37

RobotStudio, 21

电

电子表格, 33

硬

硬件, 13, 15

移

移除摄像头, 44

网

网络设置, 45

胶

胶卷, 21, 29

菜

菜单条, 21, 23

虚

虚拟控制器, 46

语

语言, 33

调

调色板窗口, 21, 30

软

软件, 13, 15

输

输入, 64

输出, 64

还

还原, 85

通

通过与失败, 59

配

配置Integrated Vision, 41

链

链接, 59

限

限制, 61

项

项目分类, 86

高

高级设置, 24

Contact us

ABB AB

**Discrete Automation and Motion
Robotics**

S-721 68 VÄSTERÅS, Sweden
Telephone +46 (0) 21 344 400

ABB AS, Robotics

Discrete Automation and Motion

Nordlysvegen 7, N-4340 BRYNE, Norway
Box 265, N-4349 BRYNE, Norway
Telephone: +47 51489000

ABB Engineering (Shanghai) Ltd.

No. 4528 Kangxin Highway
PuDong District
SHANGHAI 201319, China
Telephone: +86 21 6105 6666

www.abb.com/robotics