

B&R Automation Studio 使用入门手册



1 · 介绍	3
1.1 如何获取Automation Studio™的相关信息	5
2 · 安装	8
2.1 安装步骤	8
2.2 注册	10
2.3 目录结构	14
3 · 运行 Automation Studio™	15
4 · 第一个项目	16
4.1 运行仿真器	16
4.2 建立连接	17
4.3 创建一个新项目	20
4.4 创建一个梯形图	23
4.5 编译和传送	26
4.6 监控模式	28
5 · Automation Studio™ 使用原则	29
5.1 编辑窗口的结构	29
5.2 项目结构	38
5.3 窗体管理	33
6 · 变量	33
6.1 变量和常量的范围	33
6.2 数据类型	34
6.3 变量和常量声明	35
6.4 数组	39
7 · 初始化	40
8 · 编程语言	41
8.1 综述	41
8.2 可能性	42
9 · Automation Studio™ 调试诊断工具	48
9.1 综述	49
9.2 状态栏	49
9.3 目标系统在线信息	50
9.4 错误记录本	51
9.5 强制操作 (FORCE)	51
9.6 监控	53
9.7 监视	56
9.8 轨迹跟踪	59
9.9 NC诊断	60
9.10 PROFILER	62
9.11 调试器	63

目录

10 · 小结	64
11 · 附录	67
11.1 编程计算机和B&R控制器之间建立连接	67
11.2 I/O变量的申明定义	68
11.3 操作系统的安装	68
11.4 B&R Automation Studio 多版本安装	71
11.5 模块化培训手册一览	72

1、介绍

Automation Studio™ 是针对贝加莱(B&R)所有工业自动化产品的集成化的软件开发环境，可应用于任何规模任何范围的项目。我们不仅可在这个编程环境中处理工程项目开发中的每个步骤，提供有多语言编程和大量的诊断工具，并且可在同一个用户界面中处理贝加莱(B&R)控制系统，人机界面系统，运动控制系统的操作控制。对于初学者来说，相当容易入门，操作原理一经掌握，将大大节约了开发成本，减少了设备维护工作。

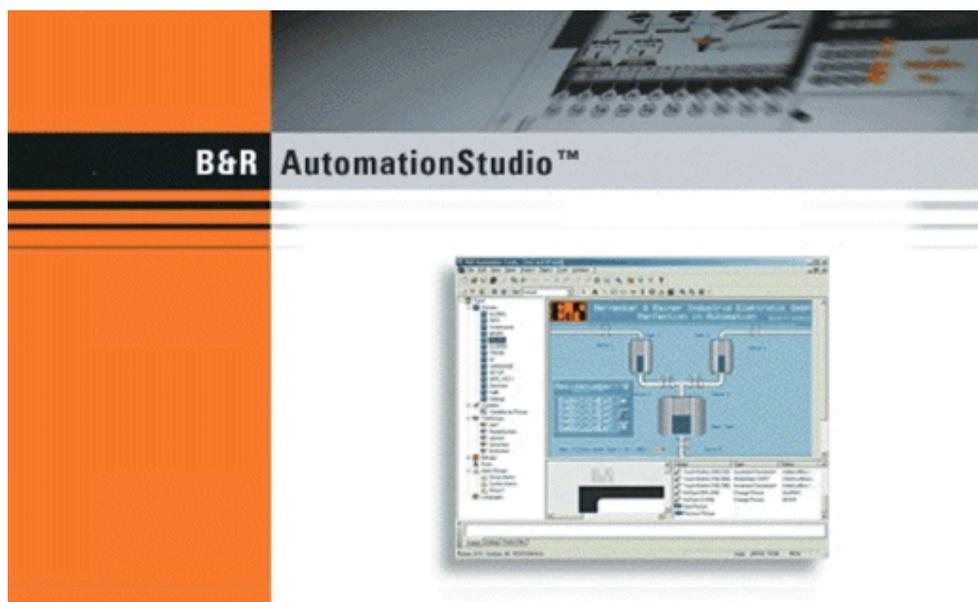


Fig. 1 Automation Studio™ 启动画面

Automation Studio™ 中集成了不同的开发工具：

- ◎ 项目的硬件配置
 - 在线自动识别目标控制器模块型号
 - 关于控制器的具体信息
 - 简单的控制器硬件分布定义
 - 符号性变量名直接分配到物理端口
- ◎ 项目的软件配置
 - 控制系统的有效配置
 - 项目等级的清晰显示
 - 简单的软件分布定义

介绍

◎ 编程语言

- ◎ B&R Automation Basic
- ◎ ANSI C
- ◎ IEC61131-3 梯形图——LD
- ◎ IEC61131-3 顺序功能图——SFC
- ◎ IEC61131-3 指令表——IL
- ◎ IEC61131-3 结构文本——ST

◎ 库和功能函数

Automation Studio™ 中提供了很多标准的功能函数，这些功能函数可以使编程人员在解决标准问题时，更加简单省时高效。

◎ 图文显示

在Automation Studio™ 配置和编程环境中，Visual Components 集成了创建高性能优化用户界面所需要的所有工具。

◎ 运动控制配置

Motion Components 集成了Automation Studio™ 中所有的定位工具，支持以下功能：

- ◎ 利用参数表进行运动控制器配置
- ◎ 示波器功能实现实时运动分析
- ◎ 利用跟踪功能记录所有相关状态和参数
- ◎ 利用集成的NC轴测试功能检测每种运动
- ◎ PLCopen 运动控制功能块的使用缩短了开发时间
- ◎ Smart Process Technology 自由配置技术库实现经济的方案和更快的生产速度
- ◎ 凸轮编辑器，轻松链接复杂的运动
- ◎ 集成CNC系统

◎ 维护及诊断

- ◎ 在线变量强制
- ◎ 在线变量监控

- 曲线跟踪
- 使用断点，单步，执行行显示等功能调试源程序
- 信息记录本使系统状态确定更方便
- Profiler性能测试工具测量任务和系统的运行

1.1 如何获取Automation Studio™的相关信息

1.1.1 运用AS在线帮助系统

Automation Studio™在线帮助是在运用Automation Studio™的关键指导，它包括所有的你在操作Automation Studio™时所需要的信息，它包括编译，用户接口以及硬件模块文件等重要的信息。



Fig. 2在线帮助系统的主要标题

你可以从Automation Studio™在线帮助上获得一切你想要的信息，象Automation Studio™本身一样，帮助的文件也会不断的修改更新。

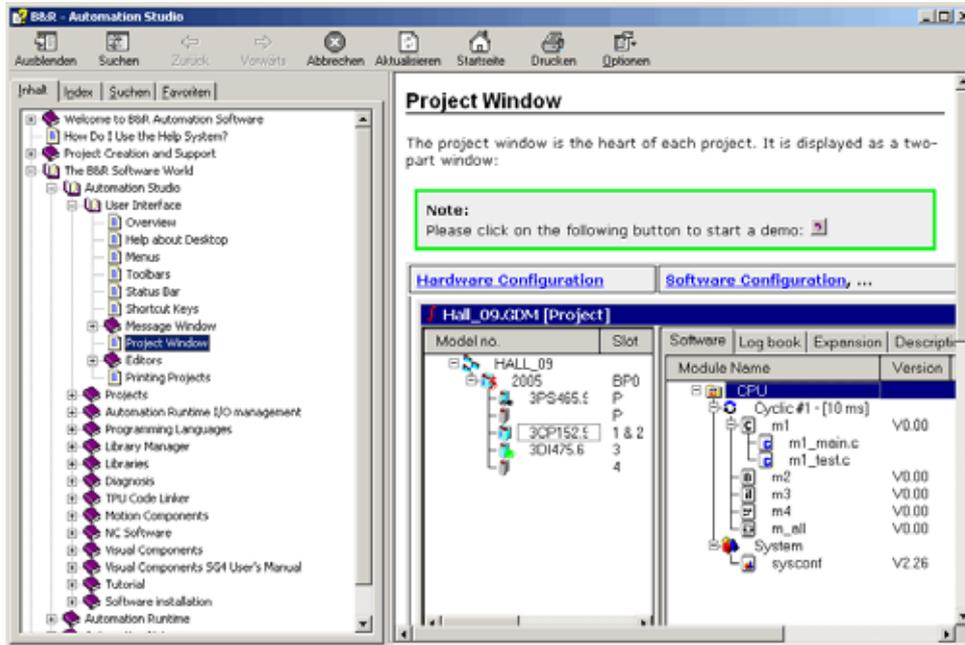


Fig. 3 Automation Studio™在线帮助

按F1键打开Automation Studio™选择部分的帮助主题，你也可以用帮助中的搜索功能查找具体信息的主题。

1.1.2 《使用入门手册》

在这本手册中，提供了Automation Studio™使用的基本信息，通过一个典型的例子来学习如何使用操作Automation Studio™。

掌握数据和变量的类型以及他们的定义声明。

对不同的编程语言和使用的可能性有个大致的了解。

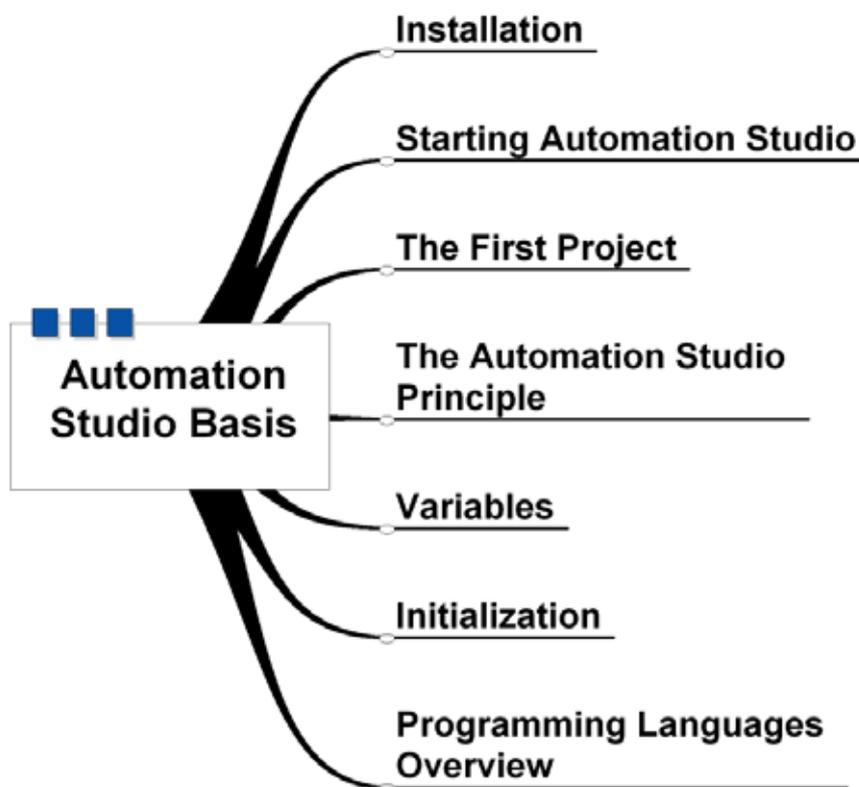


Fig. 4 综述

1.1.3 服务和技术支持

当你使用中，无法在在线帮助和相关的文档资料中找到你问题的答案，或者你希望得到更多的信息或建议，你可以和贝加莱（B&R）中国在当地的技术服务中心联系，我们完整、系统的技术支持体系将高效迅捷地解答你的疑问。

安装

2、安装

2.1 安装步骤

安装Automation Studio™计算机的配置要求：

- > 操作系统： Windows XP
- > 处理器： 至少Pentium II 400 MHz
- > 内存： 至少 256M
- > 硬盘空间： 至少 500 MB
- > 显示： XGA（1024 * 768）

计算机必须要有一个闲置的在线串行接口（COM1 – COM4）连接计算机和控制
系统。

安装Automation Studio™ 前需预先安装 "Internet Explorer 6.0"和"Microsoft .NET Framework 1.1"。无论你用什么Windows版本，这两个Windows部件必须在安装Automation Studio™前安装，同样这些部件的安装文件在Automation Studio™ 工具CD 上。

2.1.1 打开安装菜单

每张 Automation Studio™ 安装CD读取时会自动弹出一个安装菜单，也可以通过点击CD中 "BrMenu.exe"文件来打开该窗体。

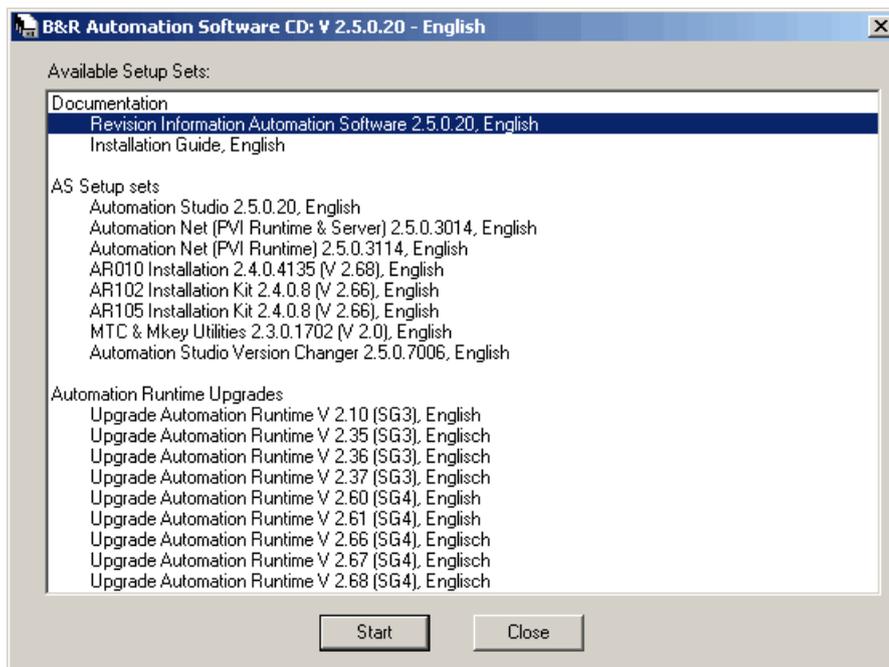


Fig. 5 安装菜单

2.1.2 安装 Automation Net

为了建立Automation Studio™和目标控制系统的通信连接，必须先运用"Automation Net (PVI Runtime & Server)" 菜单来安装Automation Net，安装向导将会引导你安装步骤。

在以下窗体中选择你要安装的部件。

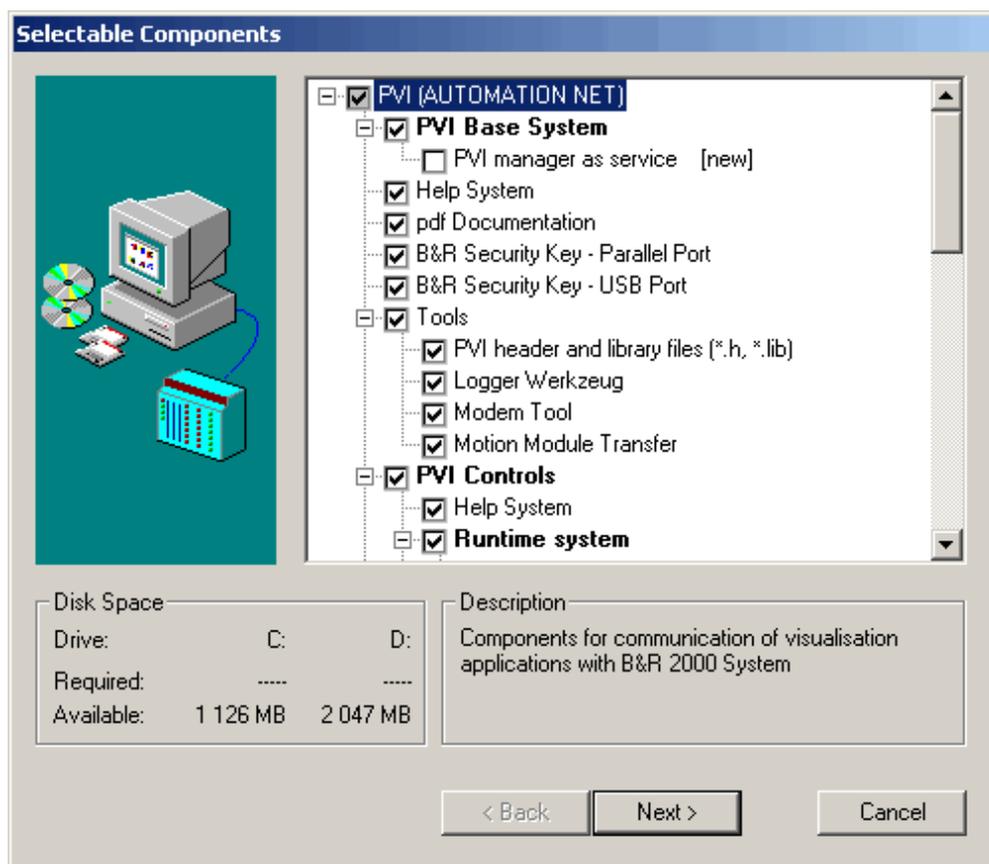


Fig. 6 选择要安装的部分

按照安装向导的步骤重新启动计算机。

安装

2.1.3 安装 Automation Studio™

Automation Studio™通过安装菜单上相应的菜单选项来安装，安装向导将会引导你安装步骤。

在以下窗体中选择你要安装的部分。

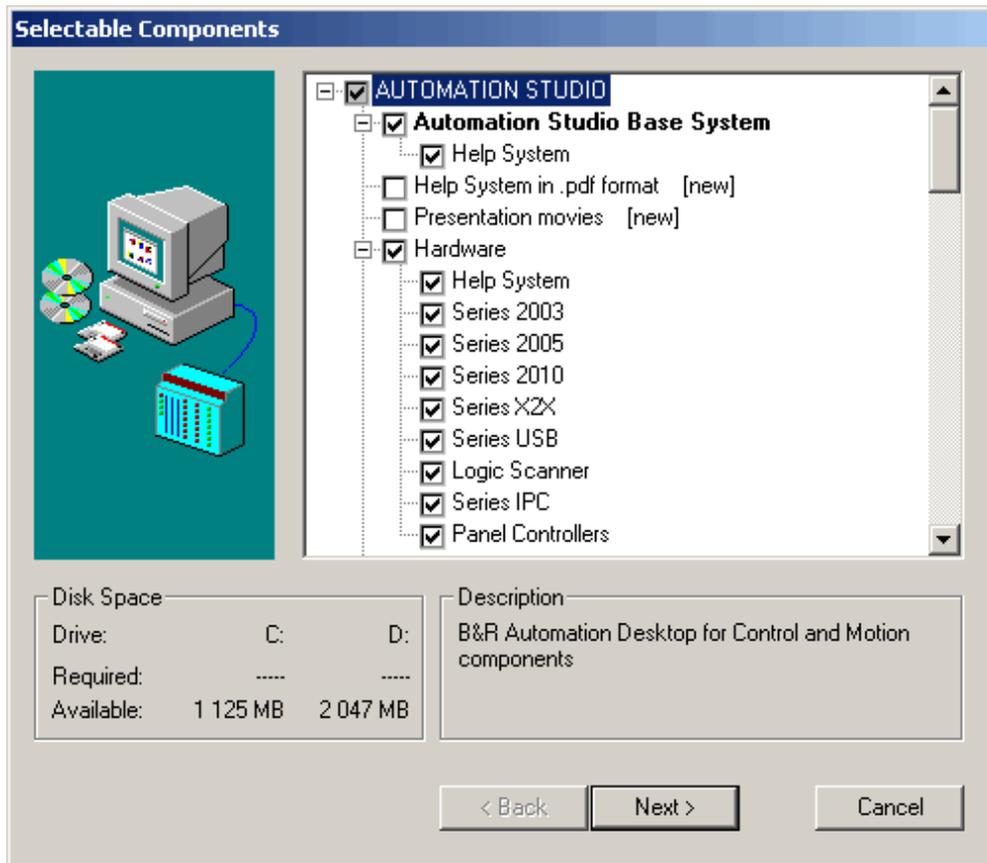


Fig. 7 选择要安装的部分

如果你需要进行计算机仿真编程测试，那必须安装AR000 runtime emulator这个部件。按照安装向导重新启动计算机。

2.2 注册

注册 Automation Studio™必须要有一个序列号，注意这个序列号不包括在 DEMO的安装 CD 上，如需购买请与贝加莱（B&R）中国联系。

当第一次启动 Automation Studio™ 时，你将会要求填写软件的序列号，如果不注册 Automation Studio™ 会允许你有30天的试用时间。



Fig. 8 软件没注册

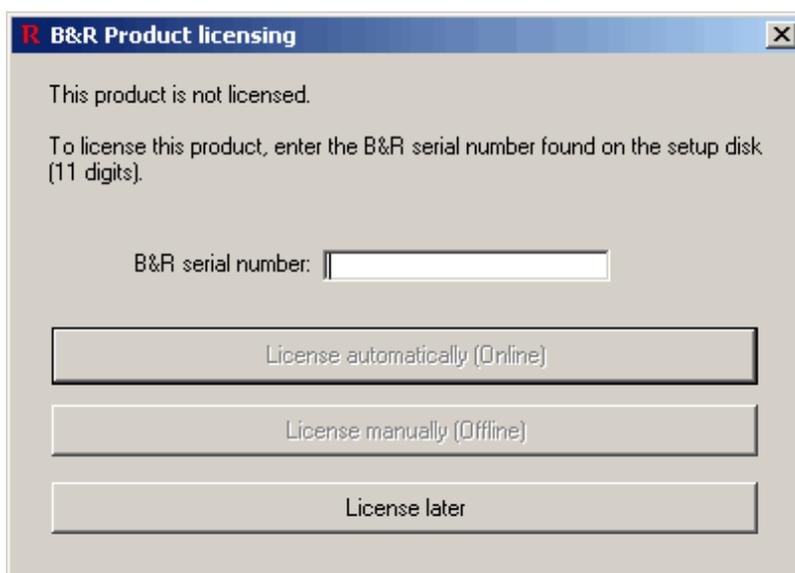


Fig. 9 输入序列号

当序列号填入后会有以下选项：

- License automatically (online)

自动通过Internet 注册Automation Studio™。

- License manually (offline)

软件必须手动到 B&R主页上注册 (www.br-automation.com)。

当手动到主页上注册的时候，步骤如下：

- 复制系统码
- 在 B&R主页上注册
- 输入注册码

备注: 如果 "Invalid ID" 错误信息 被返回，说明在输入序列号的过程中有错误。

序列号成功输入后将会弹出以下窗体：

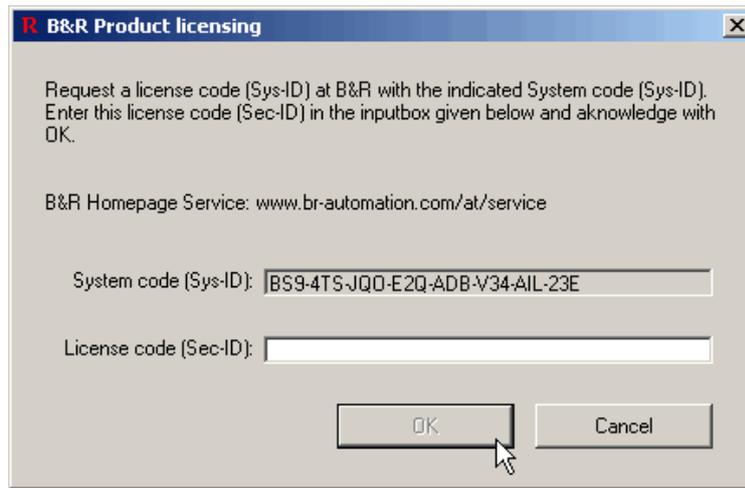


Fig.10复制系统码

复制系统码到剪贴板。

打开你的网页到www.br-automation.com按 "Services – Software Registration – Register" 到注册界面。

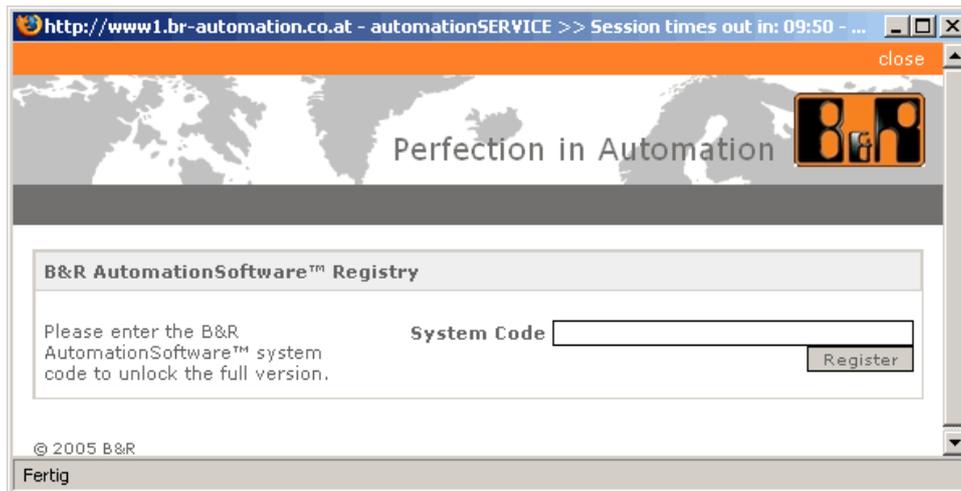


Fig. 11 软件注册

在 "System Code" 条中输入你的系统码，然后点击 Register 按钮。

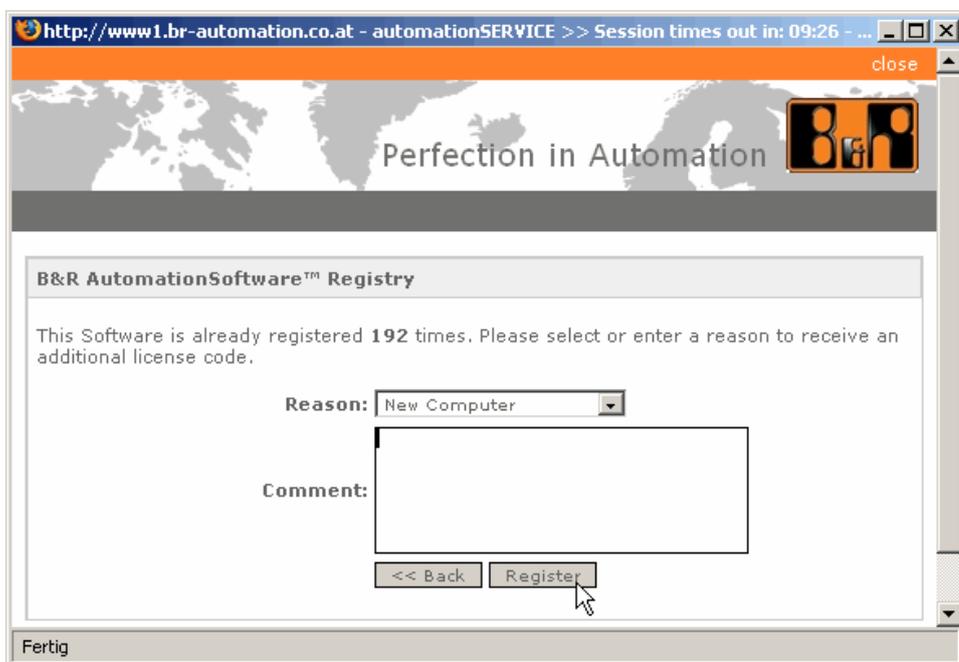


Fig. 12 说明注册理由

输入你注册的理由然后点击(Register) 按钮。

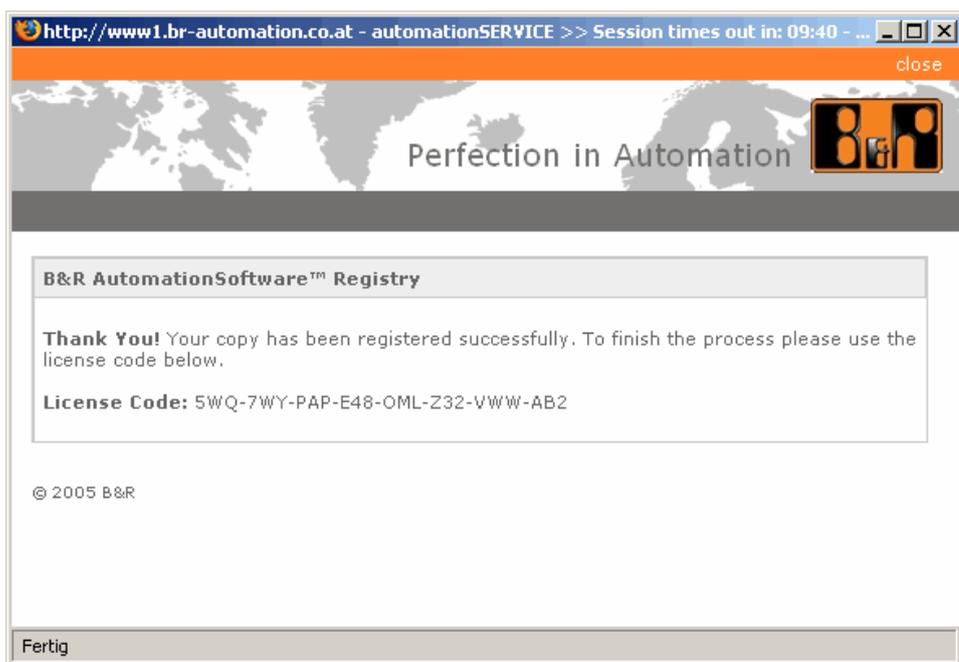


Fig. 13 接收许可码

你将会得到注册码，将注册码复制到剪贴板。



Fig. 14 在Automation Studio™中输入许可码

在 Automation Studio™里输入注册码，Automation Studio™ 注册就完成了。

2.3 目录结构

软件安装分成两部分：

Automation Net(PVI)的安装（目标路径：\BrAutomation）

Automation Studio™的安装（目标路径：\BR_AS_XXX_L0XX）

Automation Studio™ 被安装了以后，以下文件夹 将会自动安装到你安装时选择的目标路径下面：

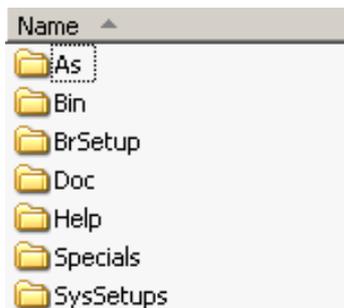


Fig. 15 AS安装路径结构

文件夹	内容
..As\gnuinst	GNU C 编译器
..As\Library	B&R 提供的函数库
..As\System	操作系统（库文件夹下每个操作系统版本的完整库）
Help	Automation Studio™完整的CHM 格式文件，这个帮助系统还可以通过打开 AS.chm来读

3、运行 Automation Studio™

正确安装后，可以在Windows开始菜单的程序组中找到Automation Studio™，并启动，你将会看到Automation Studio™的用户界面。

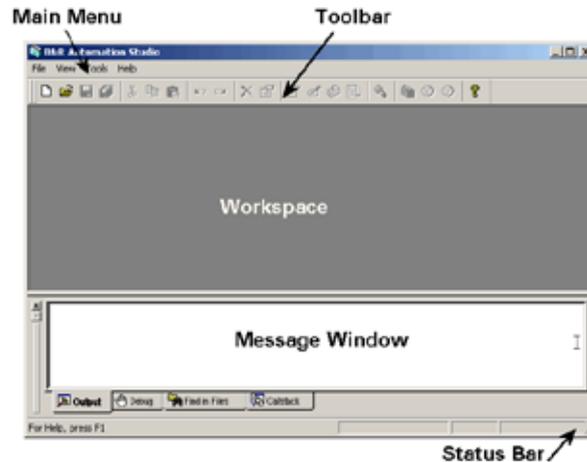


Fig. 16 Automation Studio™用户界面

Automation Studio™ 用户界面包括以下元素：

- ◎ 主菜单
可以通过B&R Automation Studio™主菜单运行所有的功能。
- ◎ 工具条
可以通过点击工具条中的按钮快速的运行命令和功能。
- ◎ 工作区
当你打开一个项目的时候该项目就在此区域显示，项目窗口可以最大化填满整个区域，也可以手动调节大小。
- ◎ 信息窗体
信息窗体位于编程窗体的最下端，它用来显示编译信息，调试信息等。此外，它还显示功能 "Find in Files"的搜索结果。
- ◎ 状态条
状态条位于整个窗体的下方，显示以下信息：
 - 菜单命令或工具条图表的简短帮助
 - 编译步骤的简短信息
 - 编程设备和目标系统的在线连接状态
 - RUN 表示运行模式，CPU运行状态
 - SERV 表示维护模式，CPU停止状态
 - DIAG 表示诊断模式，CPU最小配置模式
 - 当前活动窗体的状态数据

第一个项目

4、第一个项目

我们创建一个新的项目，编写程序，下载到目标系统中。

在此，我们每个步骤将会一一做详细解释。

一些编译的更进一步信息和如何使用它们，包括应用的原则，将会在下几个部分做详细阐述。

我们使用 AR000 runtime 仿真器，作为我们的目标系统。这样我们就可以实现不用实际控制器硬件来创建、编写和测试一个项目。

4.1 运行仿真器

虽然 AR000 runtime 仿真器不具有实时的功能，但它可以让我们在没有任何实际硬件设备的情况下测试不同的程序。

点击 Tools下面的AR000选项。

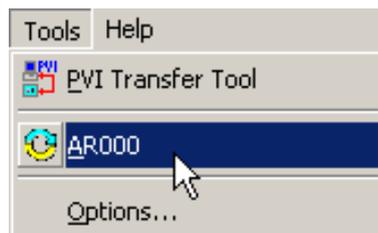


Fig. 17 运行仿真器

仿真器启动了

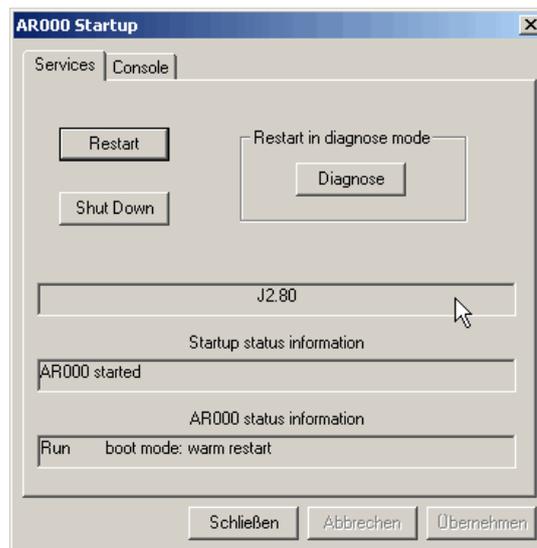


Fig. 18 AR000 runtime 仿真器

现在我们可以用 AR000作为一个目标系统去测试我们的第一个项目。

4.2 建立连接

为了可以对一个控制器操作，我们必须建立和它的连接，这是因为我们必须要把项目传输到目标系统后，我们才可以检测它。

打开Tools下的Options... 选项。

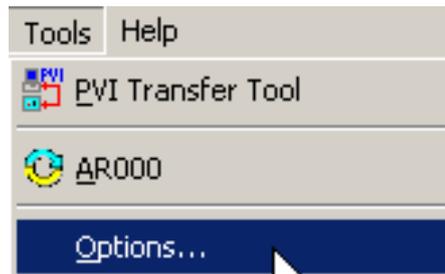


Fig. 19 连接菜单

将会弹出以下窗体



Fig. 20 连接设置 I

可以通过Add... 来新添加一个连接

第一个项目

点击后将会弹出以下窗体：

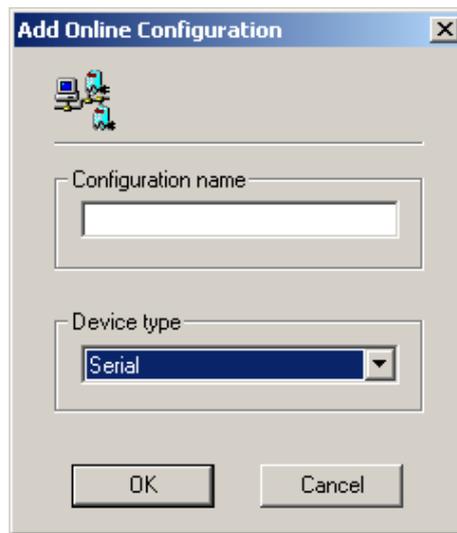


Fig. 21 连接设置 II

给你的连接一个 configuration name 然后把device type 改成 TCPIP ， 然后就会出现以下窗体：：

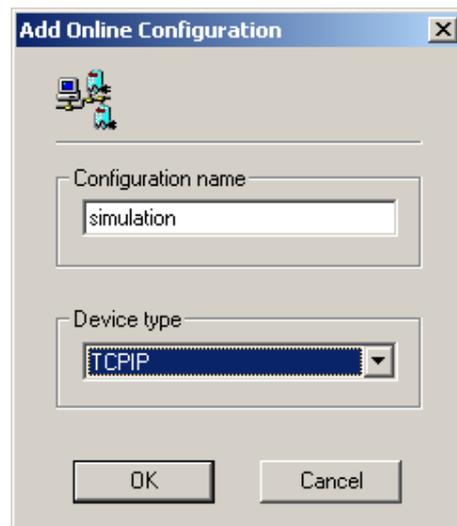


Fig. 22 连接设置 III

点击OK保存设置

现在你可以回到主窗体，选择 Properties... 按钮。

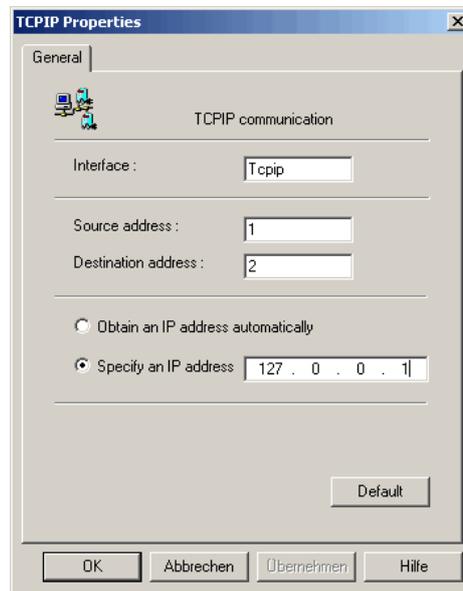


Fig. 23 连接设置 IV

按上图改变设置，然后点击OK.保存设置。

现在你就回到了主窗体。

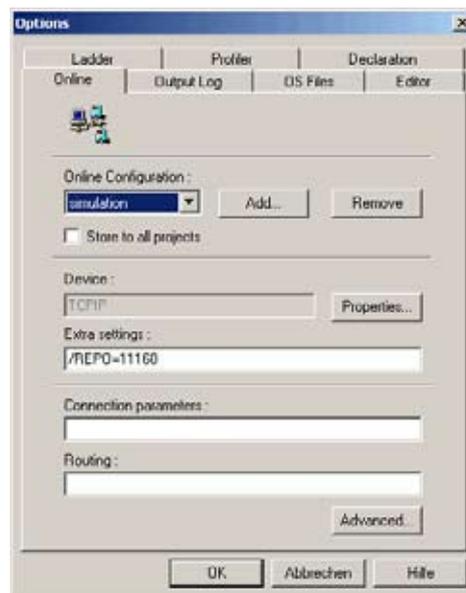


Fig. 24 连接设置 V

在 Extra settings 条中输入 /REPO=11160。

第一个项目

这样仿真器的连接就完全设置好了，一旦在指定的设置名称中保存设置后以后就不用每次再输入一次信息了。

点击OK.确认设置。

4.3 创建一个新项目

在Automation Studio™中要创建一个新项目，点击 File 下的New project... 选项。

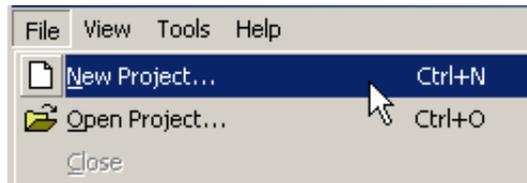


Fig. 25 创建一个新的项目 I

新项目向导将会帮助你完成创建。

执行如下设置：



Fig. 26 创建一个新的项目II

Upload hardware from target 选项意味着当前的硬件配置会可以通过连接上传上来。

- > 选择你的项目所要保存的路径，比如： D:\Projects。
- > 给你的项目起个名字，比如 "sim" (最大8个字符)。
- > 点击Next继续向导。

如果硬件上传成功，你将会看到以下画面：

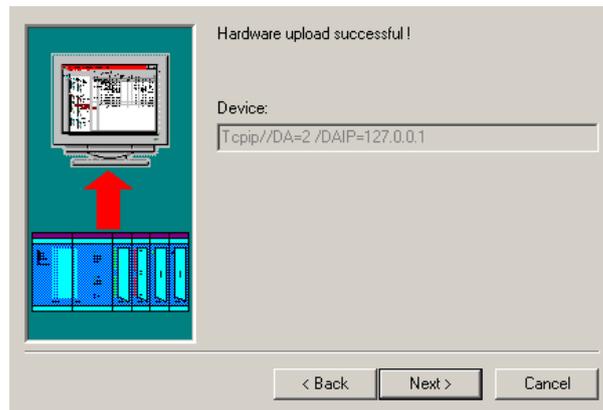


Fig. 27 创建一个新的项目III

点击Next继续向导。

如果上传硬件时出现错误，你将会看到如下窗体：

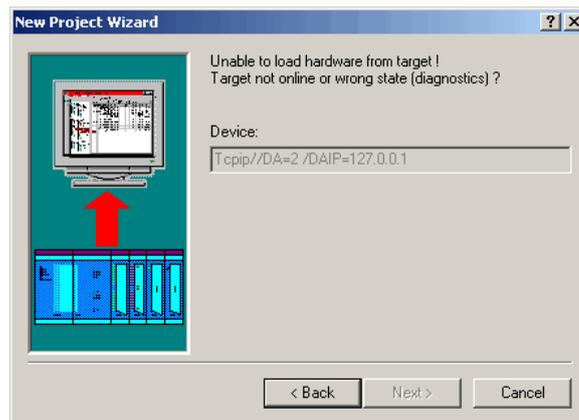


Fig. 28 不能从目标系统上上载硬件信息

在上一级窗口中取消选择Upload hardware from target，手动选择硬件。

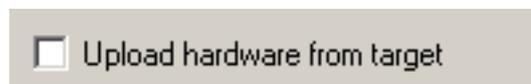


Fig. 29 不自动上载硬件

选择 AR000 runtime 仿真器然后点机击Next。

第一个项目

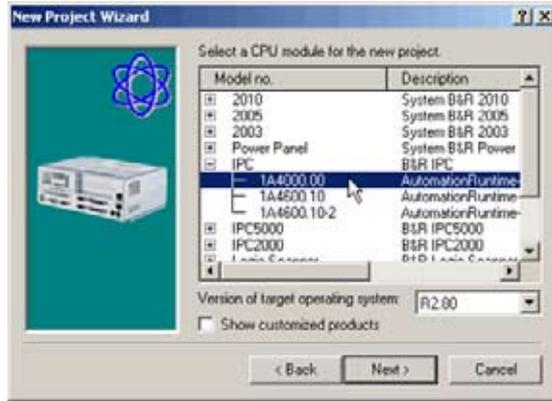


Fig. 30 选择 AR000 runtime 仿真器

项目的一个概要将会象如下一样显示出来：

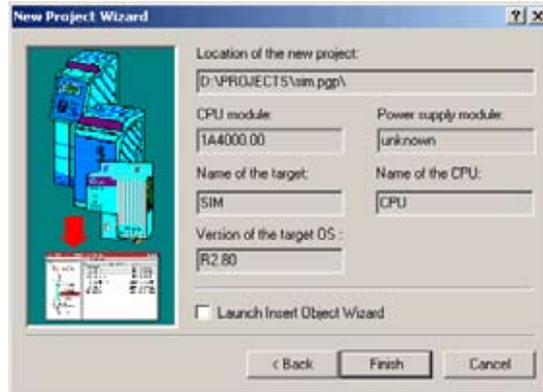


Fig. 31 创建一个新的项目 IV

点击Finish完成向导。

现在，你可以看到如下界面：

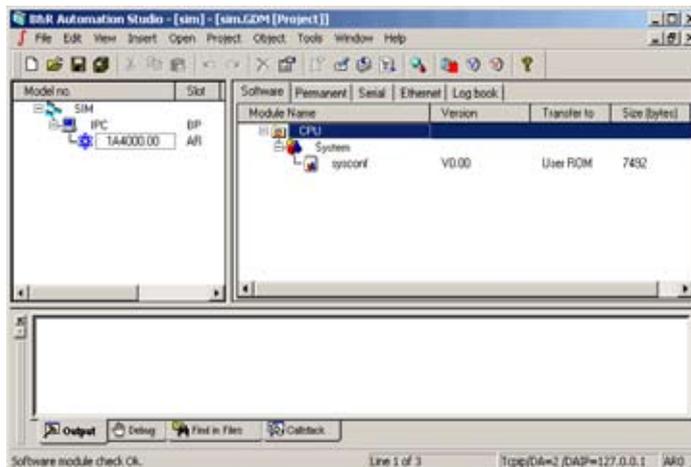


Fig. 32 新对象

4.4 创建一个梯形图

以下是在项目中创建一个梯形图的必要步骤：

- > 创建一个梯形图任务
- > 声明变量
- > 用梯形图编程

4.4.1 插入一个梯形图

在快捷菜单中选择 Insert object...



Fig. 33 插入一个对象

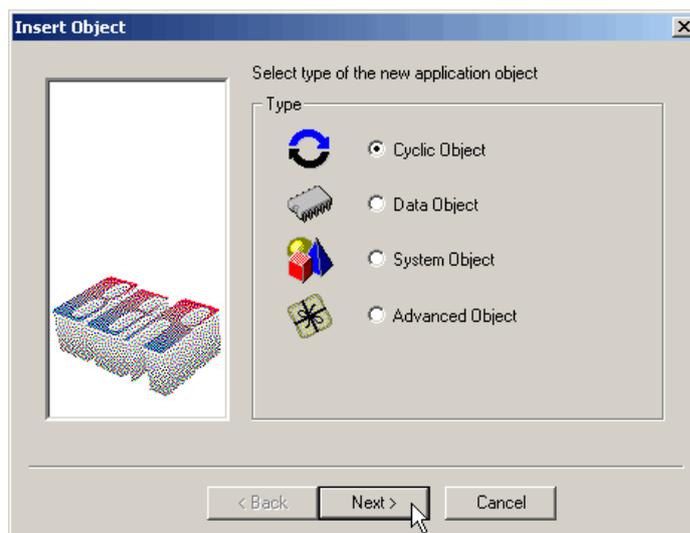


Fig. 34 选择对象类型

Cyclic Object: 一个循环对象在其定义的循环时间内执行，操作系统监控运行时间是否超出。

Data Object: 用于存储数据

System Object: 包括 *.br 文件或库文件，多数系统文件Automation Studio™会自动加载，无需手动添加。

Advanced Object: NC 轴对象、CAM 曲线、可视化对象等

第一个项目

选择 Cyclic object后点击Next按钮。

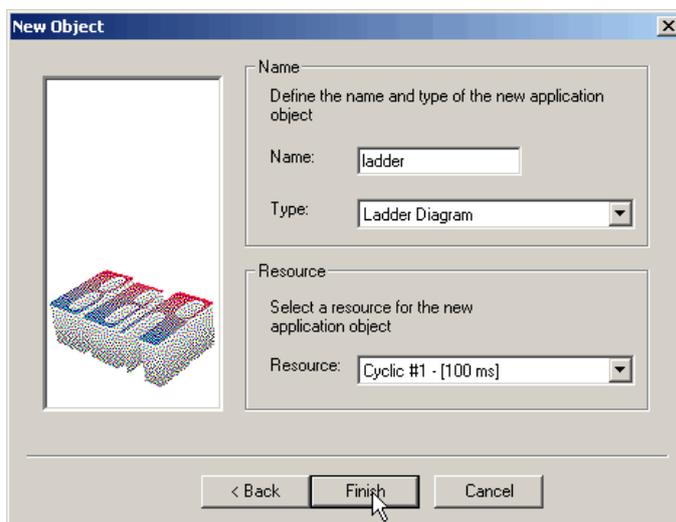


Fig. 35 选择对象参数

在 Name 栏中输入任务的名字，名字的长度必须少于等于8个字符，且不可以包含特殊字符。

在 Type 选项中，可以选择不同的编程语言：

- B&R Automation Basic
- ANSI C
- IEC61131-3 梯形图——LD
- IEC61131-3顺序功能图——SFC
- IEC61131-3指令表——IL
- IEC61131-3结构文本——ST

我们将Type 选项改变成 Ladder Diagram。

在 Resource 选项，定义任务的级别：

Cyclic #x: 由B&R Automation Runtime操作系统调用这些对象。

Timer #x: 由硬件定时器操作这些对象。

控制器中定性分时多任务的操作系统，使得高级别的任务会中断低级别的任务优先运行。

使用Resource 默认设置选项，点击 Finish 完成输入。

梯形图已经在你的项目中创建，梯形图的编辑器也随之打开，出现以下界面：

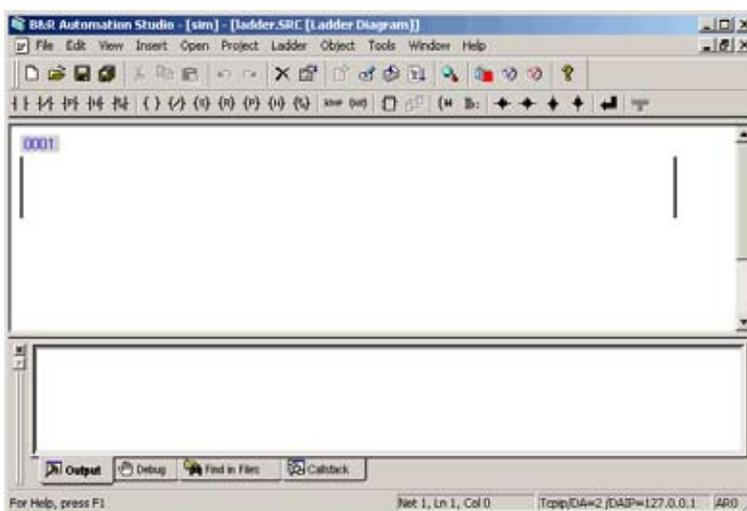


Fig. 36 梯形图编辑器

一个新的工具条将会出现在梯形图编辑器环境中，这些图标可以用来插入连接到梯形图上的开关。

4.4.2 梯形图编程

光标当前位置（显示为十字）会在编辑状态闪烁。

现在点击图标  来插入一个常开开关，在此开关上将出现闪烁的当前光标位置。输入开关名并点击ENTER确认。

然后会弹出一个变量声明的框来声明该变量，点击OK完成设置。

第一个项目

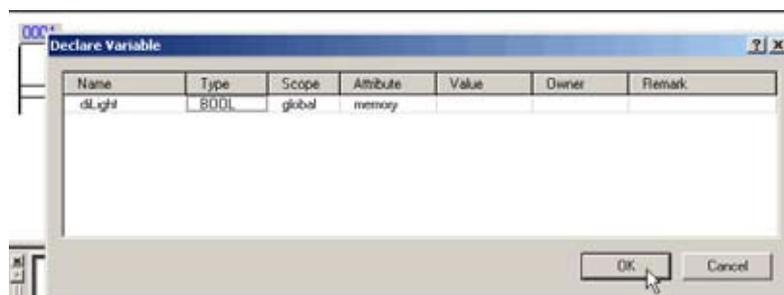


Fig. 37 声明常开开关变量

点击  图标插入一个输出开关，重复常开开关一样的设置。

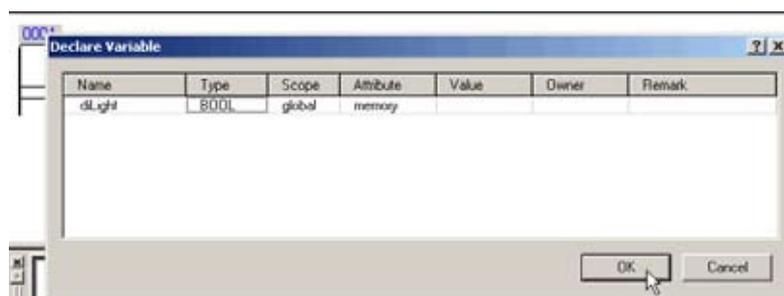


Fig. 38 声明变量表

这时你梯形图应如下图所示：



Fig. 39 完成梯形图

用  图标可以保存。

4.5 编译和传送

现在你成功地创建了一个项目和一个梯形图任务，然后当要执行任务时我们要把程序传到目标系统中。

点击  图标把梯形图任务传到目标系统中。

如果控制器CPU中已经存在项目，你将会看到以下窗口。

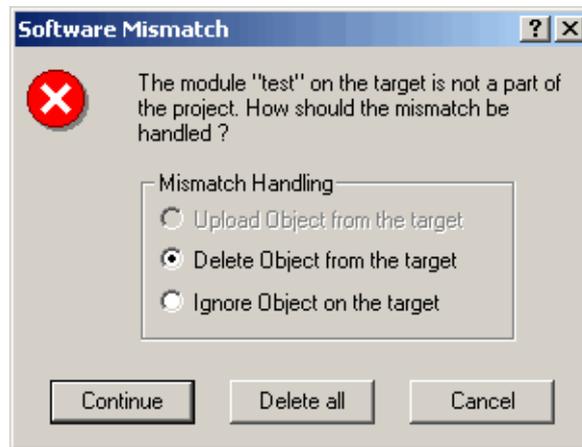


Fig. 40 删除目标系统上其他项目的部分

点击Delete all来删除所有与本项目无关的项目。

然后你将会被系统要求进行热启动或冷启动，重新启动后只有操作系统参数还保留在控制器内。



Fig. 41 目标系统重起

备注:

冷启动会删除RAM中的所有内容，所以在必须要求重新启动时才执行重起。

梯形图程序已经在目标系统中运行，现在我们可以测试 程序是否正确工作。

第一个项目

4.6 监控模式

点击  图标启动监控模式来监控梯形图程序。

出现以下显示。



Fig. 42 监控梯形图

现在你可以检测输出是否随输入改变而改变。

以上我们在Automation Studio™中创建了一个空项目，建立了和目标系统的连接，创建了一个新程序，并添加了变量，这个变量与梯形图编译相连，传送程序以后，我们用图表监控来检测不同输入的结果。

5、AUTOMATION STUDIO™ 使用原则

Automation Studio™的项目编辑中，可以访问很多窗口和菜单，因此必须熟悉和理解这些部分的联系和区别，这些窗口和菜单的设置，这些信息都可以从AS在线帮助中找到。

练习：

在Automation Studio™中寻找监控模式的信息，看看哪些关于监控模式的信息。在线帮助中寻找 "The B&R Software World - Diagnosis"信息。



5.1 编辑窗口的结构

在Automation Studio™中，硬件和软件是在不同的窗体中设置的，硬件配置树位于画面的左面，软件配置树位于画面的右面，信息窗体位于画面的下方，提供用户信息。

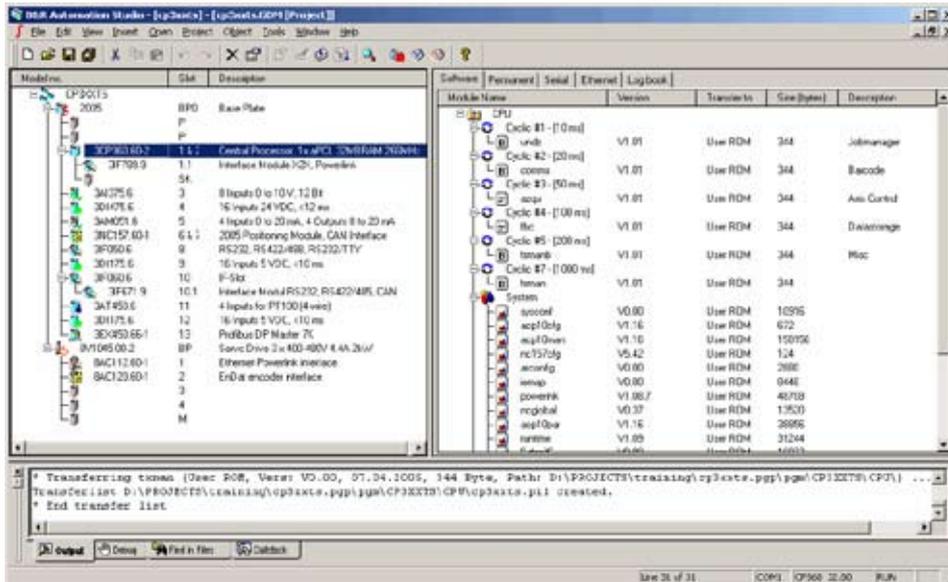


Fig. 43 Automation Studio™ 用户界面

Automation Studio™ 使用原则

5.1.1 硬件配置树

硬件配置树反映在此项目中用到的硬件，最上面显示运用的系统，在它下面是CPU连接的I/O模块和远程 I/O 模块。

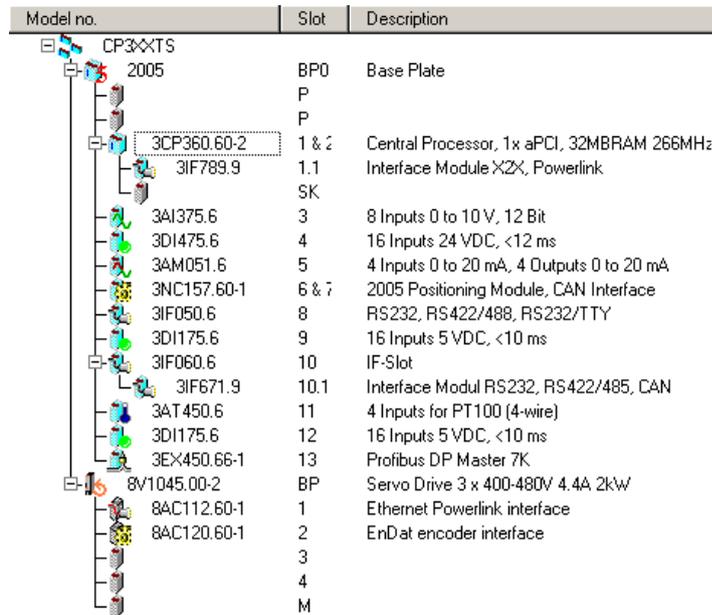


Fig. 44 Automation Studio™ 硬件树

每个单独模块的后面一列显示的是对于选择I/O数据点很重要的槽信息，最后一列是对插入模块的简短说明。

在硬件配置树中可以把变量定义到I/O模块的通道上。

按F1键可以查看选择模块的帮助信息。

硬件配置树包括以下几个方面：

- ⊙ 模块的示意显示
- ⊙ 模块在底板和远程 I/O 结点的次序
- ⊙ 槽信息
- ⊙ 模块的清晰描述
- ⊙ I/O模块上的变量定义

在线连接的情况下，如果硬件与实际连线的硬件不一致，在模块的左边将会有醒目的标注。

▲ 表示实际控制器上这一插槽上模块的型号不一致

？ 表示实际控制器上这一插槽上没有模块

5.1.2 软件配置树

软件配置树包括一个项目中所有创建和插入的软件对象，比如任务对象，系统对象，轴对象，图文显示对象，数据模块，这个区域不仅显示这些软件对象，还用图表显示它们的资源，包括它们的版本号，文件大小，存储内存和描述。

用户可以自己定义任务的版本号和描述。

Module Name	Version	Transfer to	Size (Bytes)	Description
CPU				
Cycle #1 - [10 ms]				
unde	V1.01	User ROM	344	Jobmanager
Cycle #2 - [20 ms]				
comrx	V1.01	User ROM	344	Barcode
Cycle #3 - [50 ms]				
acqu	V1.01	User ROM	344	Axis Control
Cycle #4 - [100 ms]				
fbc	V1.01	User ROM	344	Datastorage
Cycle #5 - [200 ms]				
bemar	V1.01	User ROM	344	Misc
Cycle #7 - [1000 ms]				
benan	V1.01	User ROM	344	
System				
sprconf	V0.00	User ROM	10916	
acp10crg	V1.16	User ROM	672	
acp10man	V1.16	User ROM	150156	
nc157crg	V0.42	User ROM	124	
acrcrg	V0.00	User ROM	2990	
ionap	V0.00	User ROM	8448	
powertrk	V1.00.7	User ROM	48708	
ncglobal	V0.37	User ROM	13520	
acp10par	V1.16	User ROM	38856	
nufine	V1.09	User ROM	31244	
Ddon65	V0.00	User ROM	16072	
Ddtr65	V0.00	User ROM	24728	
Data objects				
acp10dyn	V1.16	User ROM	389908	
dshcp	V0.00	User ROM	0	Recipe Data
ACP10 Axis				
test	V0.00	User ROM	388	

Fig. 45 Automation Studio™ 软件配置树

可以在软件配置树中添加对象再和项目一起下载到目标系统中。

软件配置树包括以下几个部分：

- ◎ 图表显示的软件对象
- ◎ 对象的资源分配
- ◎ 版本号，文件大小，存储内存和描述信息
- ◎ 项目中所有软件对象的处理

Automation Studio™ 使用原则

5.1.3 信息窗体

信息窗体用来显示警报，错误和一些其他信息，这些信息在编译和查找错误时十分有用。

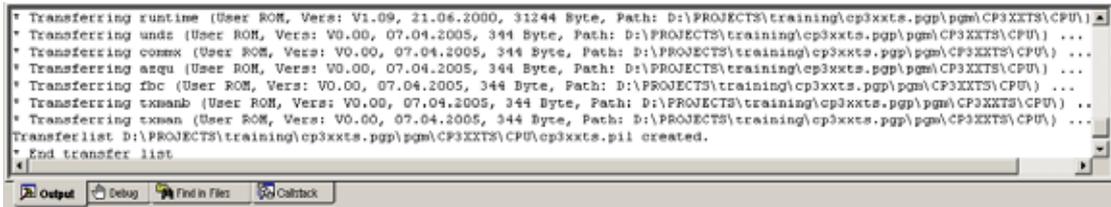


Fig. 46 Automation Studio™ 信息窗口

信息窗体包括以下几个部分：

- ⊙ 编译时的警报和错误信息
- ⊙ 双击显示的错误信息就可以看到是什么引起的，出现的位置
- ⊙ 下载项目的进程和状态
- ⊙ 在项目或目标系统中插入和删除对象的信息显示
- ⊙ 在项目文件中搜索的输出结果
- ⊙ 调试信息的输出窗口
- ⊙ 在项目中运用"Find in Files"功能的输出结果

5.2 项目结构

项目中所有的文件存储在<项目名>.pgp的目录下，被分割成好几个文件夹和文件。

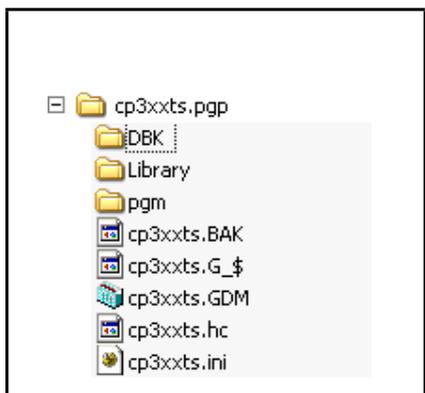
 <pre> cp3xxts.pgp ├── DBK ├── Library ├── pgm ├── cp3xxts.BAK ├── cp3xxts.G_\$ ├── cp3xxts.GDM ├── cp3xxts.hc └── cp3xxts.ini </pre>	<ul style="list-style-type: none"> • 库 在项目中应用到的库 • DBK 项目的数据库，包括数据类型和变量的声明 • PGM 源文件和编译对象 • XXXXX.GDM 在Automation Studio™可以打开的项目文件 • XXXXX.hc 项目的硬件设置文件 • 项目的具体设置和编者设置
--	--

Fig. 47 项目结构

5.3 窗体管理

当我们用Automation Studio™时，允许在工作区同时打开10个窗口。

5.3.1 通过菜单或快捷键切换

可以用菜单中Window下拉或按<CTRL + TAB>切换窗体。

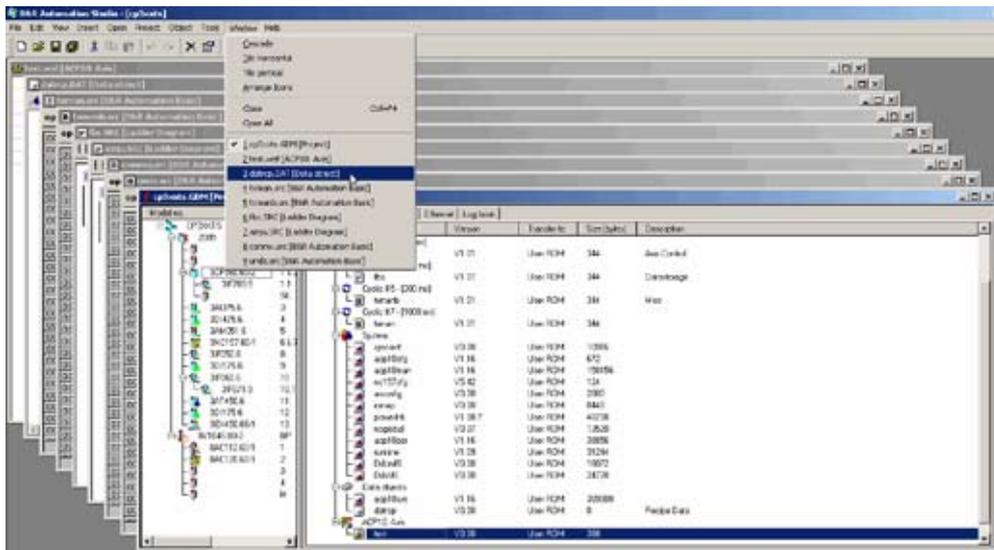


Fig. 48 独立窗口间切换

备注:

最好把不用的窗体关闭（已经完成编译有一段时间的窗体），保留那些真正需要的窗体，这样可以使处理起来更快更清晰。

6、变量

变量在程序中作为一个符号，通过变量访问读写内存中的数值，使用符号性变量，用户就不必考虑内存分配，程序任务会自行处理这些问题。

常量和变量很相象，与变量不同的是，在软件创建的时候常量就被设为一个定值，在运行时它的值不能改变（因此叫常量）。

6.1 变量和常量的范围

变量有2种范围:

- ◎ 全局变量和全局常量在整个项目中全有效，可以在任何任务中被调用。
- ◎ 局部变量和局部常量只有在当前任务有效，其他任务中不能调用局部量。

6.2 数据类型

数据类型是描述一个变量的性质，比如，它可以包括变量的范围，精度，和可以对它做什么操作。

6.2.1 基本数据类型

以下的数据类型被成为基本（原始）数据类型，它们在所有的编程语言都可以被使用。

Binary	Unsigned	Signed	Floating point	Time, date, string
BOOL	USINT	SINT	REAL	TIME
	UINT	INT		DATE_AND_TIME
	UDINT	DINT		STRING

数据类型	长度 [bytes]	值范围
BOOL	1	TRUE (1), FALSE (0) 数字量输入输出
SINT	1	-128 ... +127
INT	2	-32768 ... +32767 模拟量输入输出
DINT	4	-2147483648 ... +2147483647
USINT	1	0 ... 255
UINT	2	0 ... 65535
UDINT	4	0 ... 4294967295
REAL	4	-3.4E38 ... +3.4E38
TIME	4	T#24d_20h_31m_23s_648ms ... T#24d_20h_31m_23s_647ms
DATE_AND_TIME	4	D T # 1 9 7 0 - 0 1 - 0 1 - 0 0 : 0 0 : 0 0 ... DT#2106-02-07-06:28:15
STRING	Variable	字符串显示

6.3 变量和常量声明

变量和常量在Automation Studio™按如下声明。

6.3.1 一个任务的声明

一个任务的声明窗体会显示所有在此任务中调用的变量，包括其局部变量。

在软件配置树中选择你想要看它声明的任务。

可以通过菜单Open – Declaration或点击图标。

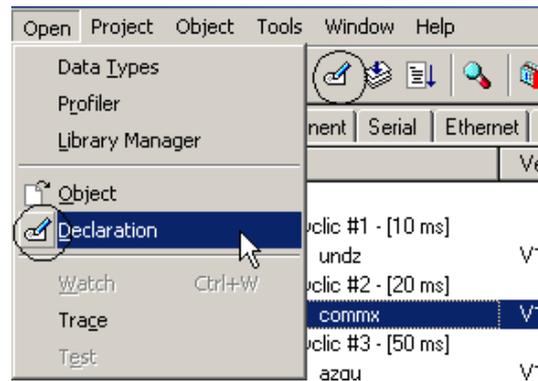


Fig. 49 打开声明窗口

将会弹出以下窗体：

Name	Type	Scope	Attribute	Value	Owner	Remark

Fig. 50 声明窗口

点击右键按New variable添加新的变量，也可以点击Insert key (INS)。

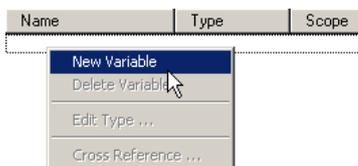


Fig. 51 添加一个新变量

然后会出现一个空条，可以输入变量的名称。

不同的列可以编辑新的或已经存在变量的类型，范围等。

Name	Type	Scope	Attribute	Value	Owner	Remark
diSwitch	BOOL	global	memory			
doSwitch	BOOL	global	memory			

Fig. 52 改变范围(scope)

双击"Data type"的空白处可以打开以下窗体：

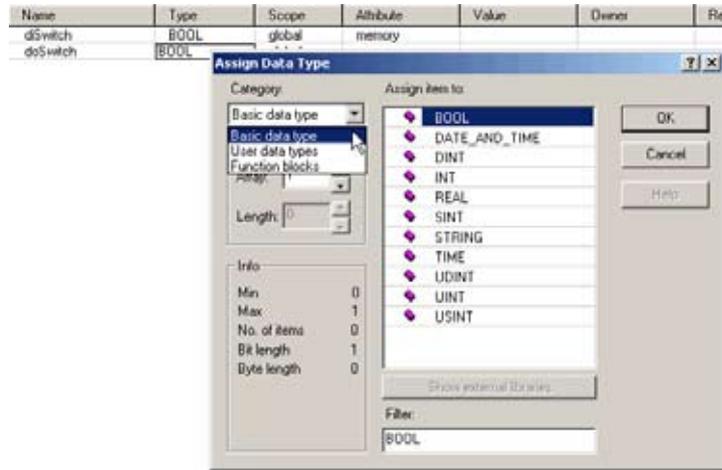


Fig. 53 改变数据类型

在种类处可以选择basic data types, user data types和 function blocks。

此外, 在Array处可以定义变量的长度。

Attribute这一栏可以定义变量或常量, 当把它改成常量的时候, 缺省显示为绿色。

Name	Type	Scope	Attribute	Value
diSwitch	BOOL	global	memory	
doSwitch	BOOL	global	memory	FALSE
VIS_MAIN	BOOL	global	constant	FALSE

Fig. 54 改变属性

Value这一栏可以用来定义常量的值。

点击图标  保存你的设置。

6.3.2 全局变量声明

全局变量声明窗体将会显示项目中调用的所有全局变量, 还可以添加变量。

在软件配置树中选择CPU, 然后如上打开变量声明就可以打开全局变量声明窗体。

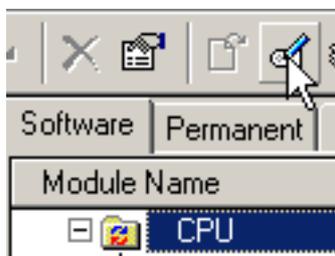


Fig. 55 打开全局变量声明窗口

6.3.3 结构（用户数据类型）

变量如果是基本变量类型的话，该变量同时只能有一个值。

结构变量定义一个变量如何构成，使一个变量可以包括多个独立的成分，它把不同分散的变量在一起组成反映一个功能或任务的结构变量。

例子:

下面我们来设计创建一个烘面包的程序。

原料是水，面粉，盐和发酵粉，他们组成面包的比例如下：

Water := 3

Flour := 5

Salt := 1

Yeast := 1

若有不同的烘烤配方，你就需要大量不同的独立变量。

所以有时候必须把烘烤配方的变量组合起来以减少整体变量的数量。

创建一个用户数据类型：

要在Automation Studio™中创建用户数据类型，点击菜单Open – Data Types将会出现以下界面：

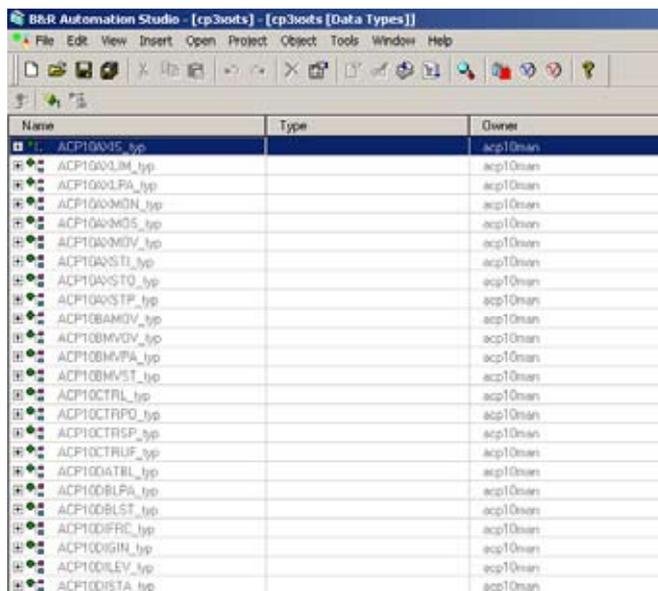


Fig. 56 数据类型编辑

变量

项目中的多有数据类型将全部在这显示。

右键点击Insert New Type 或点图示工具栏上画圈的图标来添加一个新的数据类型。

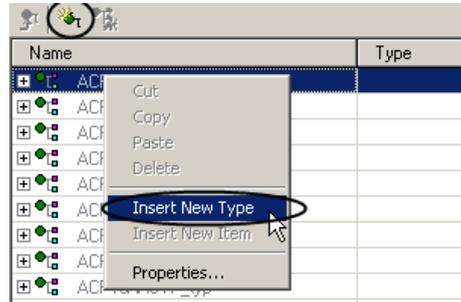


Fig. 57 添加一个新的数据类型

给数据类型起个名字

鼠标右键点击Insert New Item或点击下图工具栏中画圈的图标可以给数据类型添加一个数据元素。

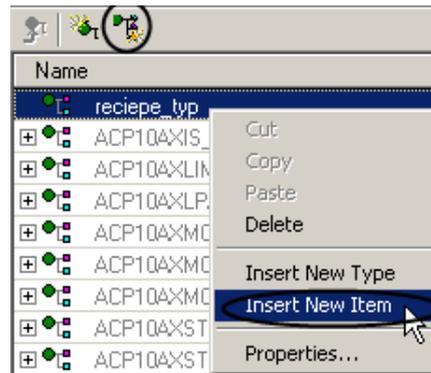


Fig. 58 添加元素

完成后的数据类型如下图所示：

Name	Type
reciepe_typ	
├─ water	USINT
├─ flour	USINT
├─ salt	USINT
└─ yeast	USINT

Fig. 59 数据类型

6.3.4 功能块数据类型

每个功能块都有输入和输出，这些参数集中在一个结构体中，当功能块被调用时，当前程序就会接受到其数据结构，当一个功能块被添加后，在Watch窗口中，我们可以清楚的看到这个功能块及其包括的独立元素。

6.4 数组

数组也是个变量，它包括许多有相同数据类型的元素，这些元素通过索引来访问，这些元素既可以声明为基本数据类型（简单数组），也可以声明为用户数据类型（结构数组）。

数组的索引总是从0开始的。

要访问简单数组的元素如下所示：

- `ArrayVariable[ArrayIndex]`

结构数组如下所示：

- `ArrayVariable[ArrayIndex].Element`

在 Automation Studio™ 中，当选择数据类型的时候，变量可以在变量声明窗口中定义为数组。

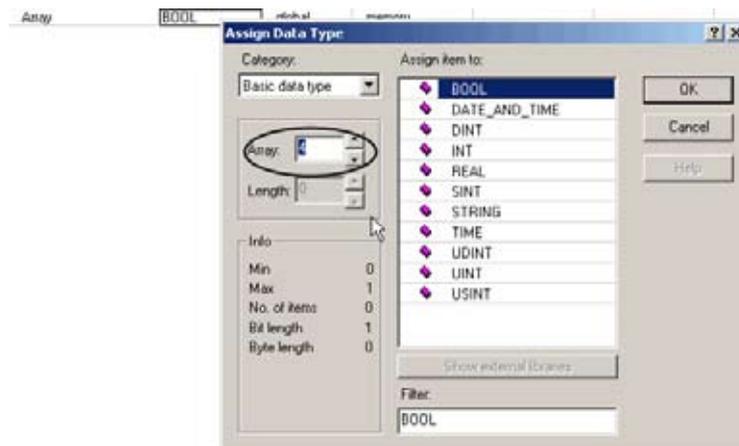


Fig. 60 设置数组的大小

当程序需要一系列相同数据类型的变量来执行相同的动作时，使用数组（基本数据类型或结构）是十分必要的。

初始化

7、初始化

变量声明窗口：变量和常量的初始化值可以在变量声明窗口中输入。

Value这一列就是用来设变量初值的，有如下两种情况：

- 变量可以设为一个固定的值（变量范围内的值）。
- 变量也可以设为remanent，这些值在系统重起前存储在缓冲存储区中，系统重起时复位（热起时保留）。

Name	Type	Scope	Attribute	Value
VIS_MAIN	BOOL	global	constant	FALSE
diSwitch	BOOL	global	memory	remanent
doSwitch	BOOL	global	memory	FALSE
Array	BOOL[10]	global	memory	

Fig. 61 声明

任务初始化：如果任务有效的话，当循环系统启动时（在程序循环部分执行前运行），每个循环任务都要通过他的初始化子程序（Init-Sp）。

初始化子程序Init-Sp可以包括定义变量初值的程序代码。

循环任务部分：系统的循环部分在变量声明和任务初始化后启动，变量直到再次被赋值或系统重起前保持它的设定值（参照变量声明和remanent变量部分）。

Remanent 变量和永久（permanent）变量：和前面提过的一样，系统重起时（热起或掉电）remanent储存在一个安全的存储区，当系统重起完毕，它的值可以从存储中再次读入，Permanent变量处理上也非常相近，不同的是它在冷起的时候也可以保持，在任何情况下，CPU或底板上的缓冲（电池，可再充电电池）会保持它的值。

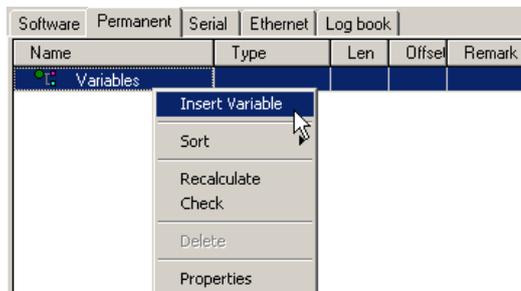


Fig. 62 插入一个永久变量

要在permanent这一栏中创建变量，你必须将变量声明窗口中Value一列设置为"remanent"。

8、编程语言

8.1 综述

在Automation Studio™中可以创建多种编程语言，因此，在一个项目中可以任
你希望选择多种语言进行编程来达到你的目的。

可以运用以下编程语言：

编程语言	备注
梯形图 (LAD)	图形化
顺序功能表 (SFC)	图形化和文本化
IL指令表 (IL)	文本化
结构文本 (ST)	文本化
高级语言 (AB)	文本化
C语言 (C)	文本化

在Automation Studio™中，所有的文本编程语言用的是 同一个编译器，同一个
诊断工具，处理起来也用同一种方式，这使其工作更为简单，增加效率。

不管编程语言是文本化的还是图形化， 查看（Watch）窗口中检测和设值的处理
是一样的。

备注：

B&R标准库中的功能块可以在所有的编程语言中调用。

编程语言

8.2 可能性

虽然理论上可以用任何编程语言去创建用户想要的功能，但每种语言都有其特有的专长。

下表中横向表示编程语言，纵向表示不同的功能。

逻辑	LAD	FBD	IL	ST	AB	C
算术	√	√	√	√	√	√
判断			√	√	√	√
循环			√	√	√	√
步序			√	√	√	√
动态变量				(√)	√	√
功能块	√	√	√	√	√	√

备注：

运用功能块可以扩展那些编程语言不能实现的功能。

图形化编译器用来创建梯形图的逻辑。

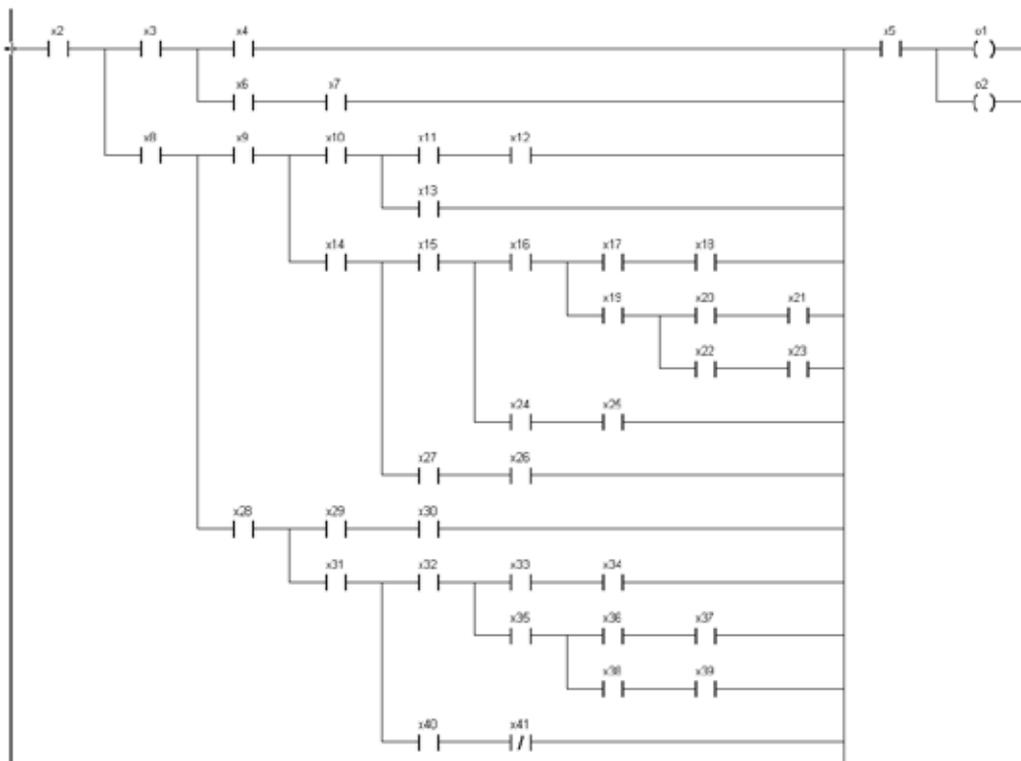


Fig. 63 梯形图编程

B&R Aotomation Basic是一种基于文本的高级编程语言。

```
TON_Temp_Mittel.PT = TW2s          : 1950ms.....Differenzwert jeder Sekunde messen
TON_Temp_Mittel FUB TON()
if TON_Temp_Mittel.Q = 1 then
  l_M_UI_Temp_Mittel = ((g_M_UI_Cnt3 + l_M_UI_oldCnt_Temp) / 4)
  l_M_UI_oldCnt_Temp = g_M_UI_Cnt3
  if ((l_M_UI_Temp_Mittel >= 0) and (l_M_UI_Temp_Mittel <= 200)) then      : Diffe:
    l_M_UI_Temp_Mittelwert = l_M_UI_Temp_Mittel
  endif
  g_M_ResetCnt3 = 1
  TON_Temp_Mittel.IN = 0
  TON_Temp_Mittel FUB TON()
endif

:---- Mittelwert in Temperatur umrechnen -----
if g_M_ResetCnt3 = 0 then
  l_M_UI_Vorige_Anzeige = g_M_IN_Temperaturistwert
  g_M_IN_Temperaturistwert = (int(((real(l_M_UI_Temp_Mittelwert))-24)/(1.2)))
  if (g_M_IN_Temperaturistwert < - 20) or (g_M_IN_Temperaturistwert > 100) then
    g_M_IN_Temperaturistwert = l_M_UI_Vorige_Anzeige
  endif
  : Abgleich der Temperatur über Offset-Einstellung
  g_M_IN_Isttemperatur = g_M_IN_Temperaturistwert
  if Einstellungen.Konfiguration.Temperaturumschaltung = 2 then
    g_M_IN_Isttemperatur = INT(((g_M_IN_Isttemperatur) * 9) / 5) + 32)
  else
    g_M_IN_Isttemperatur = g_M_IN_Isttemperatur
  endif
  g_M_IN_Isttemperatur = g_M_IN_Isttemperatur + Einstellungen.Offset.Temperatur
endif

:-----
```

Fig. 64 Aotomation Basic编程

ANSI C也是基于文本的高级编程语言，它有和汇编语言不同的符号和语法。

```
/*-----*/
int _CYCLIC main( void ) /* Zyklischer Teil des C-Objektes */
{
  int i;
  int NachfOk;

  if(GrpSelected) /* Weg in Listbox ausgewählt */
  {
    WegPtr[1].Bez=&WegBez1;
    strncpy(BezEdit, WegPtr[GrpSelected].Bez->Bez[WegSelected[GrpSelected]], 21);
    SaveGrpSelected=GrpSelected;
    GrpSelected=FALSE;
  }
  /*
  */
}

if(TimerHupe) TimerHupe--;
for(i=1;i<=ANZ_GRP;i++)
{ /* Abschaltverzögerung bei Verlust des Nachfolgers
  in Betriebsart verriegelt */
  if(StoerTimer[i]) StoerTimer[i]--;
  /* Verriegelung ein bei Automatik */
  if(GrpStatus[i].autom) GrpStatus[i].verriegelt=TRUE;
}

/* for(i=1;i<=7;i++) nur für Test */
for(i=1;i<=ANZ_M;i++)
{
  /* Tastenfarben zuweisen */
  Color[i]=0; /* alle Bits rücksetzen */
  if(Taste[i]) Color[i]=TASTE_EIN;
  else Color[i]=TASTE_AUS;
}
```

Fig. 65 ANSI C编程

下面我们举例说明梯形图编程语言，关于其它编程语言的具体信息可以参见相关的在线帮助系统和手册。

编程语言

我们将编写一个应用程序控制传输带。

第一步：

编写一个通过按下“btnConvStart”按钮来控制传输带电动机的数字量输出“gDoConvMotor”的程序。不要把输入和输出直接相连。象下面的图示所示，使用“cmdManConvMotor”作为中间变量。

程序中所有新的部分用绿色显示-> .

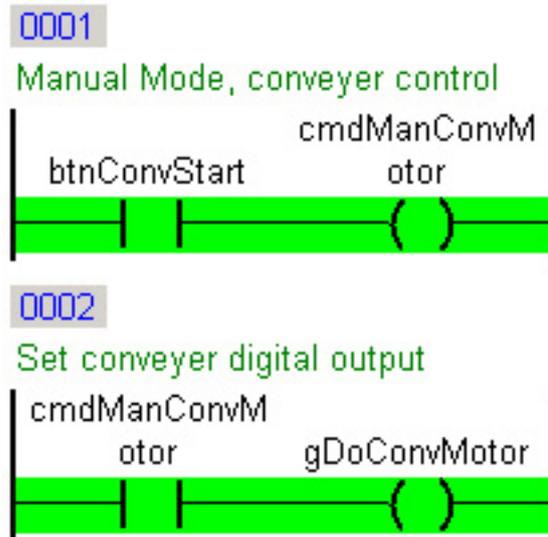


Fig. 66 第一步 源程序



Fig. 67 传送带

第二步:

我们已经能通过一个按钮来控制传输带，现在我们要加入一些新的功能:

如果传输带的终端传感器“gDiLoadConvEnd”没有检测到原料。

如果传输带的终端传感器检测到原料并且机器在数字量输入

“gDiMachAskMat”说明需要更多的原料。

→ 启动传输带

如果传输带的终端传感器检测到原料并且机器不需要更多的原料。

→ 停止传输带

那么程序可以这样编写:

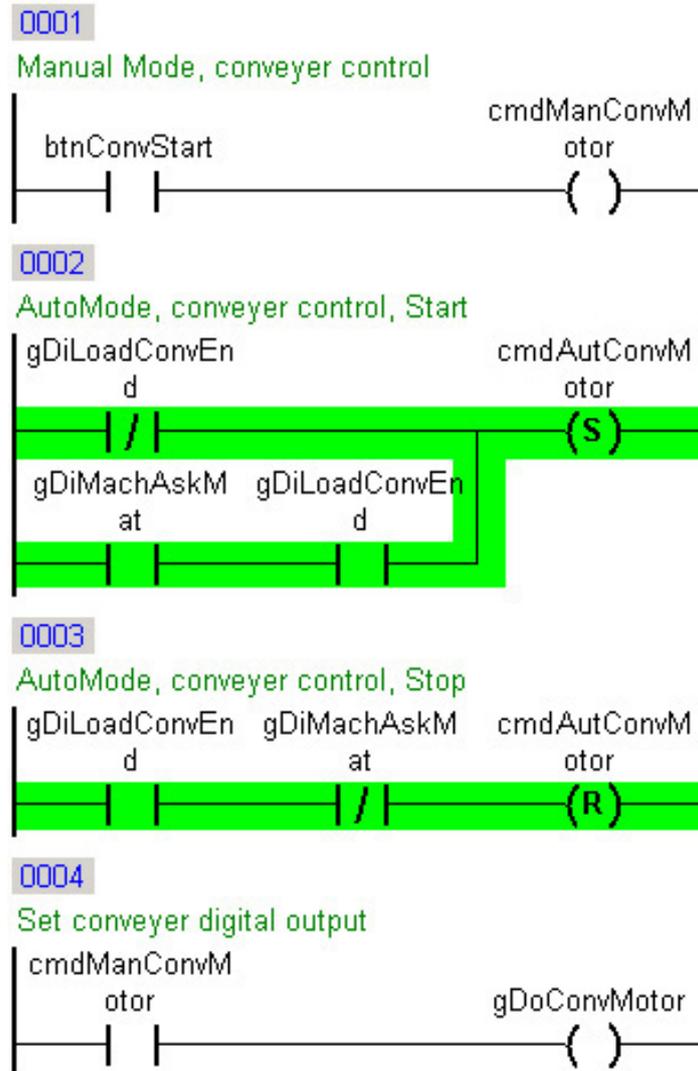


Fig.69. 第二步程序

编程语言

第三步:

现在我们要加入输入量“gDiAutoMode”来切换手动和自动模式。

如果“gDiAutoMode”是TRUE，传送带执行自动模式。

如果“gDiAutoMode”是FALSE，传送带执行手动模式。

使用条件跳转，在自动模式中使用一个新的变量“cmdAutConvMotor”。

那么程序可以这样编写:

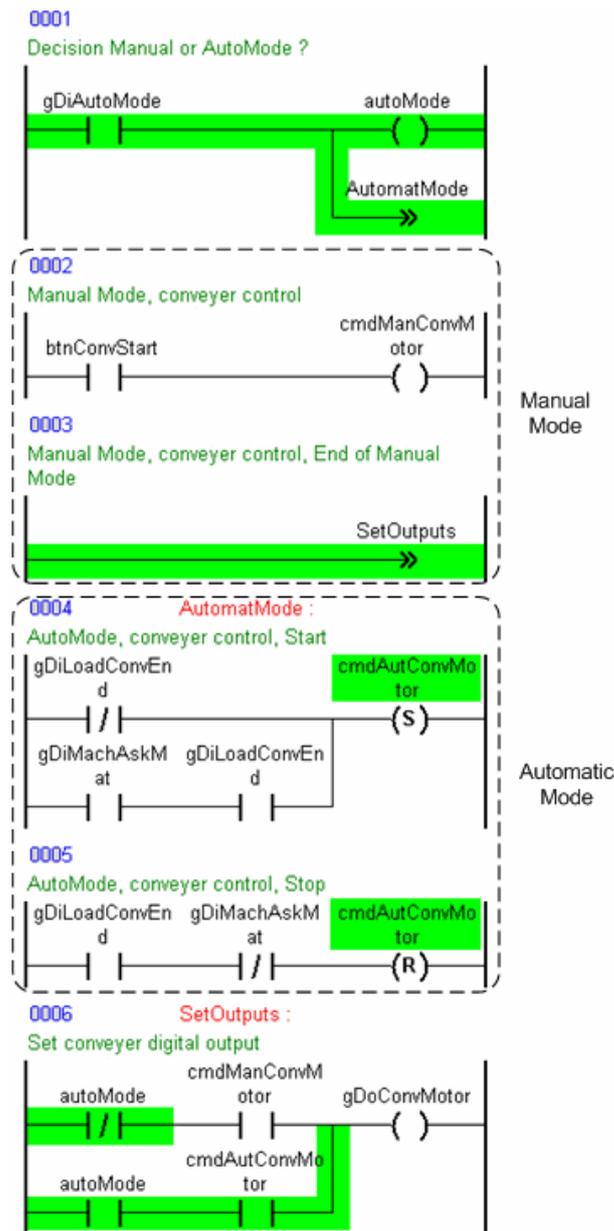


Fig.69. 第三步程序

第四步:

在自动模式测量传输带的原料数，使用在STANDARD库中的CTU功能块。

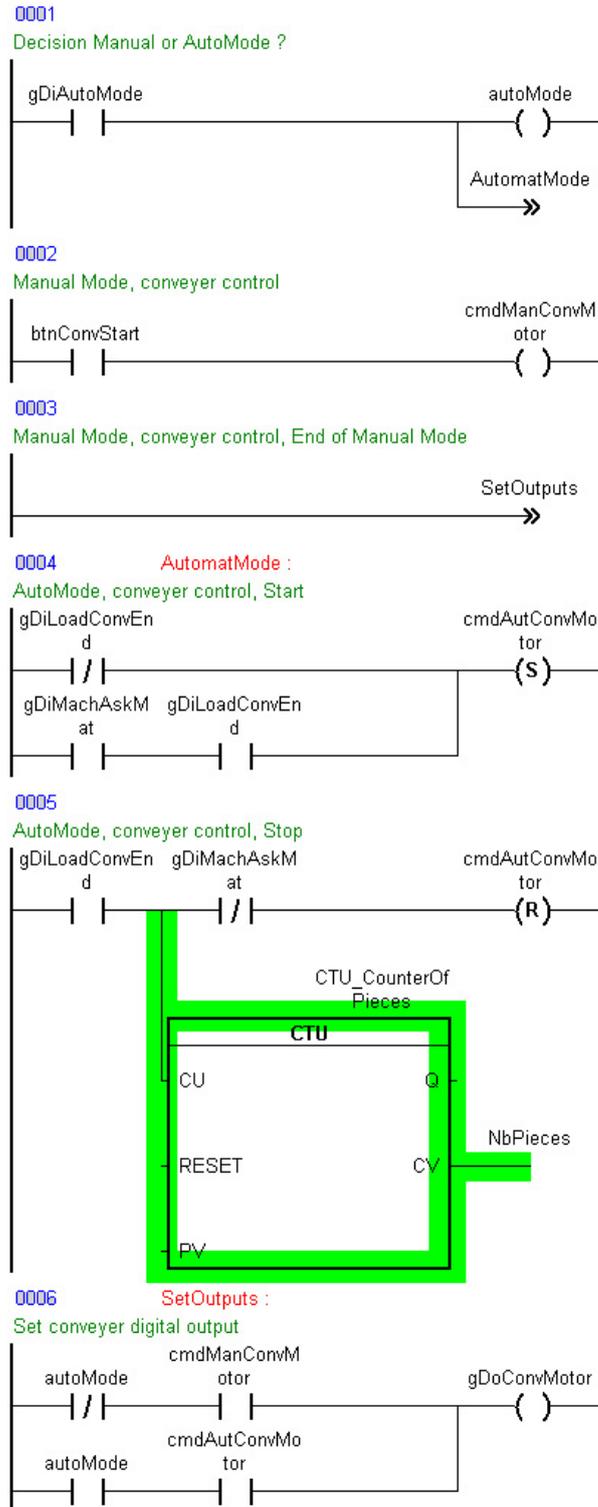


Fig.69. 第四步程序

Automation Studio™ 调试诊断工具

9、Automation studio™ 调试诊断工具

任何一个设计好的程序只有通过调试、发现问题、通过修改，再调试、再修改，如此反复才能最终达到完全正确，只有调试才能最终来验证你编制程序的正确性。

Automation Studio™ 为控制系统提供了多个标准化诊断工具。

所有重要的数据可以读取、生成和保存，也可以用作文档。

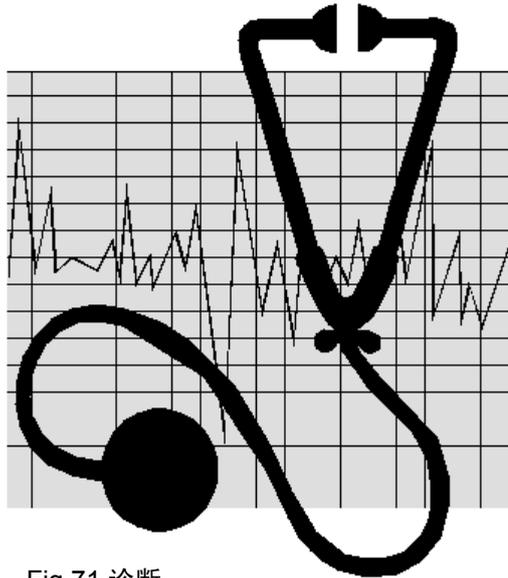


Fig.71 诊断

Automation Studio™ 中的诊断工具不仅为运行中查找错误，而且在软件开发方面提供不可替代的支持。

在Automation Studio™ 在线帮助中可以找到关于如何操作每个工具的大量信息。下面介绍如何和在哪里使用这些工具。

9.1 综述

Automation Studio™ 提供了大量的不同的诊断工具。有些工具只显示正在运行概要，有些工具允许读取、存储和修改信息，还有些可以深入系统并优化它。

有以下几种诊断工具：

- 状态栏
- 目标系统在线信息
- 错误记录本
- 监控器 (系统，硬件，I/O，强制，编程语言)
- 监视 (watch) & 存档
- 轨迹跟踪
- Nc诊断
- PROFILER
- 调试器

在工程中，我们有必要学习和掌握大量不同的工具。正确的选择工具决定你能否可能在最短的时间内获到你想要的信息。

9.2 状态栏

在Automation Studio™中，状态栏位于窗口的底部。在窗体的右下方会显示以下信息：

- OFFLINE (离线) 与目标系统没有连接
- 连接到目标系统：
 - CPU 类型 + 控制器上操作系统版本
 - 控制器状态



Fig.72 Offline –与控制器没有连接



Fig.73 Online –与控制器有连接



Fig.74 Online –服务模式



Fig.75 Online –诊断模式



Fig.76 Online –启动模式 (装入默认的AR)

Automation Studio™ 调试诊断工具

9.3 目标系统在线信息

在线信息使你可以找到关于目标系统信息的观察窗口。包括关于可以运用的存储器容量、电池状态、CAN总线或以太网连接的节点数量和在线目标系统上的时钟设置。

可以在CPU快捷菜单中选择Online info...打开在线信息。

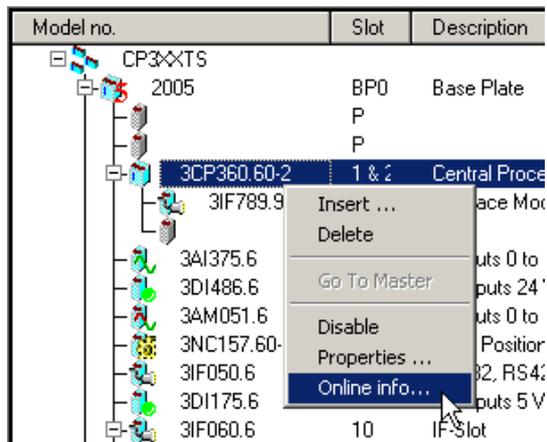


Fig.77 打开在线信息窗口

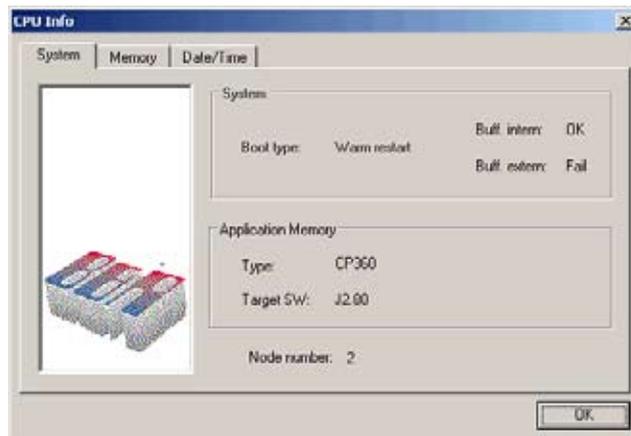


Fig.78 在线信息

Automation Studio™ 调试诊断工具

9.5.1 监控窗口 (WATCH)

监视窗口用红色的灯显示输出变量，用绿色的灯显示输入变量。双击某个变量的VALUE框会弹出强制选项窗体，确认后就可以强制改变变量的值。当强制开启时，在监视窗口中输入的值被写到软件或硬件中，与I/O数据点的状态值无关。

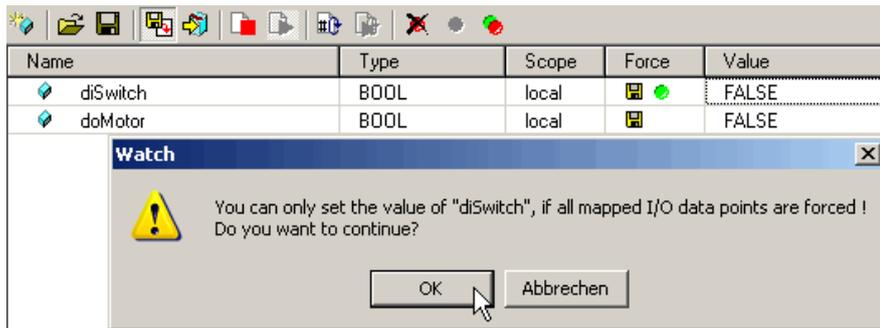


Fig.81 在监控窗口下强制

9.5.2 I/O 监控

你可以点击硬件树下的任何一个I/O点来观察和强制它的值。只需要首先转换到监控模式。

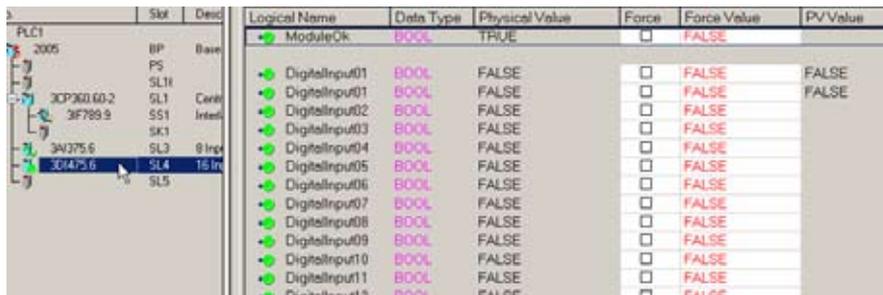


Fig.82 在监控模式下的 I/O设置

任何的I/O数据点都可选择强制选项来设置强制值。

输入和输出都可以在任务监控模式下强制。

备注:

可以取消每个单独的输入/输出的强制。通过选择 Project – Services – Force Global Off菜单或重新启动来立刻取消所有的强制操作。

9.6 监控

监控用来获得目标系统的信息。所有的监控由  启动。

9.6.1 系统监控

系统监控用于提供目标系统的软件版本信息。不仅操作系统有版本号，用户开发的软件也可以有版本号。知道哪个版本在目标系统上运行是十分重要的。为了确定版本必须知道对象的大小、版本、日期和时间。

通过这些信息，你可以决定软件版本是什么时候创建的，是否记录变成最新版本的转变，在机器上的或在工程中的软件版本的普遍程度。

系统监控用  启动。

Module Name	Target vs. Project	Location	State	Description
CPU				
Cyclic #2 - [20 ms]			RUN	
combx	not on target			Barcode
g1	not on target			
Cyclic #3 - [50 ms]			RUN	
azqu	not on target			Axis Control
Cyclic #4 - [100 ms]			RUN	
lbc	not on target			Datastorage
Cyclic #5 - [200 ms]			RUN	
bmanb	not on target			Misc
Cyclic #7 - [1000 ms]			RUN	
bman	not on target			
Exception				
undz	not on target			Jobmanager
System				
sysconf	older on target	System ROM	USE	
ocp10clg	not on target			
ocp10man	not on target			
nc157clg	not on target			
arconfig	not on target			
iomap	not on target			
powerlink	not on target			
reglobal	not on target			
ocp10par	not on target			
runtime	not on target			
Ddonat6	not on target			
Ddlatf6	not on target			
cs_bmod	on target only	System ROM	USE	
can2000	on target only	System ROM	USE	
lbcan	on target only	System ROM	USE	
dbracer	on target only	System ROM	USE	
profiler	on target only	System ROM	USE	
librtuadp	on target only	System ROM	USE	
iqvw	on target only	System ROM	USE	
lnaiaqdrv	on target only	System ROM	USE	
lociaqdrv	on target only	System ROM	USE	
Data objects				
ocp10ays	not on target			
datcp	not on target			Recipe Data
ACP10: Axis				
test	not on target			

Fig. 83 系统监控

在每个对象的旁边的信息表明对象是否只在目标上或工程中，或者在目标系统上的和在工程中的软件版本是否相同。那些信息也包括在目标上的对象是否正在被执行以及它位于哪个目标存储器中。

另外，软件对象可以停止、启动和删除。在系统监控中也可以启动监视窗口和跟踪。终止软件的某些特殊部分可以停止整个机器功能的循环动作（测试单独的软件对象）。

Automation Studio™ 调试诊断工具

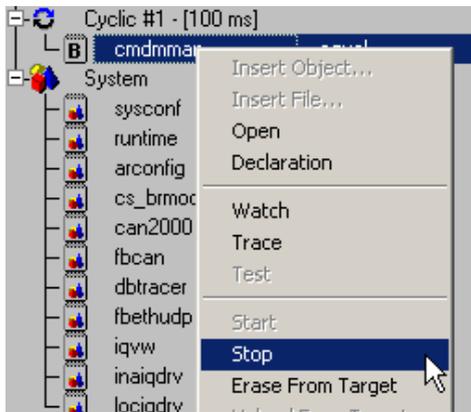


Fig.84 停止一个任务

9.6.2 监控硬件配置

在监控模式下的硬件配置辨别硬件设置定义和实际配置是否匹配。不像工程，这里用图标表示

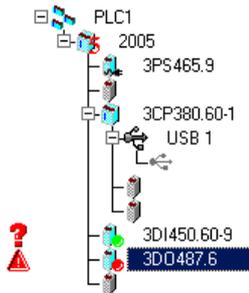


Fig.85 在监控模式下的硬件配置

对图标的解释：

- 问号：模块不在这个槽中或没被发现。
- 感叹号：在槽中的不是配置的模块。

9.6.3 监控I/O配置

在监控模式下的I/O配置辨别工程中配置的I/O模块是否确实存在，这被映射成模块状态。每个I/O模块通道的实际值也可以显示、输出，也可以设置（查看Force）。

Logical Name	Data Type	Physical Value	Force	Force Value	PV Value
ModuleOk	BOOL	TRUE	<input type="checkbox"/>	FALSE	
DigitalInput01	BOOL	FALSE	<input type="checkbox"/>	FALSE	FALSE
DigitalInput02	BOOL	FALSE	<input type="checkbox"/>	FALSE	FALSE
DigitalInput03	BOOL	FALSE	<input type="checkbox"/>	FALSE	FALSE
DigitalInput04	BOOL	FALSE	<input type="checkbox"/>	FALSE	FALSE
DigitalInput05	BOOL	FALSE	<input type="checkbox"/>	FALSE	FALSE
DigitalInput06	BOOL	FALSE	<input type="checkbox"/>	FALSE	FALSE
DigitalInput07	BOOL	FALSE	<input type="checkbox"/>	FALSE	FALSE
DigitalInput08	BOOL	FALSE	<input type="checkbox"/>	FALSE	FALSE
DigitalInput09	BOOL	FALSE	<input type="checkbox"/>	FALSE	FALSE
DigitalInput10	BOOL	FALSE	<input type="checkbox"/>	FALSE	FALSE
DigitalInput11	BOOL	FALSE	<input type="checkbox"/>	FALSE	FALSE
DigitalInput12	BOOL	FALSE	<input type="checkbox"/>	FALSE	FALSE

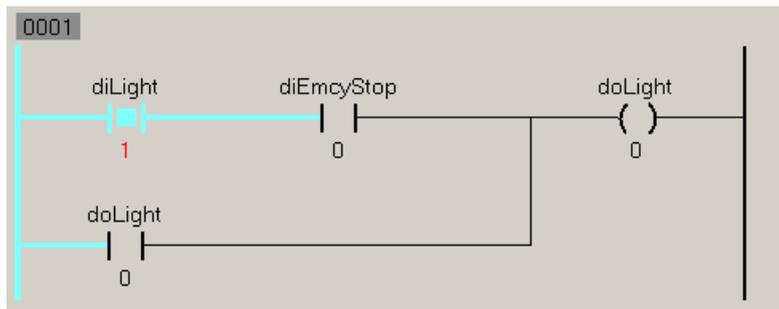
Fig.86 在监控模式下的 I/O 配置

9.6.4 监控编程语言

监控模式适用与所有的编程语言。但是不能与系统监视器混淆。在监视模式下，可以看到程序以及它的变量（包括它们的值）。这样可以有利于你推断程序的流程和它的结果。

9.6.4.1 图形化编程语言

在图形化编程语言中，变量值就显示在符号的旁边。也有可视化的辅助手段以十分简单的方式显示程序的流程。



图Fig.87 梯形图信号流程显示

在上面的图形中，信号的流程用带颜色的线和符号表示。除了程序中的变量值，这提供了除了程序中的变量值以外的另一种诊断的方法。

可以点击  图形打开信号流程显示。

9.6.4.2 文本编程语言

在文本编程语言中，变量值在屏幕右方显示与代码列对齐。

0001	(* cyclic program *)	
0002	CASE condition OF	condition = 0
0003	0: setVal := 1;	setVal = 1
0004	aoLimit := 255;	aoLimit = 255
0005	1: setVal := 2;	setVal = 1
0006	aoLimit := 256;	aoLimit = 255
0007	2: setVal := 3;	setVal = 1
0008	aoLimit := 267;	aoLimit = 255
0009	3: setVal := 4;	setVal = 1
0010	aoLimit := 268;	aoLimit = 255
0011	END_CASE	

Fig.88 高级语言监控画面(ST)

代码显示和变量列表的窗口宽度可以在Project – Settings – IEC Editor Settings 栏中配置。

Automation Studio™ 调试诊断工具

9.6.4.3 行检测

行检测可以显示在某个特殊时段哪些程序行和控制器系统正在执行。

0001	(* cyclic program *)	
0002	CASE condition OF	condition = 3
0003	0: setVal := 1;	setVal = 4
0004	aoLimit := 255;	aoLimit = 268
0005	1: setVal := 2;	setVal = 4
0006	aoLimit := 256;	aoLimit = 268
0007	2: setVal := 3;	setVal = 4
0008	aoLimit := 267;	aoLimit = 268
0009	3: setVal := 4;	setVal = 4
0010	aoLimit := 268;	aoLimit = 268
0011	END_CASE	
0012		

Fig.89 行检测(ST)

9.7 监视 (WATCH)

可以在监视窗口下显示、监控、修正目标系统的变量。

9.7.1 监视 (WATCH) 窗口

当打开监视窗口时，监视窗口的所有功能与软件树中选择的任务有关。换句话说，只能显示选择任务中的变量。如果监控窗口是在选中软件树中CPU时打开的，所有的全局变量能被显示。

通过点击 快捷菜单中的Watch可以打开监视窗口。

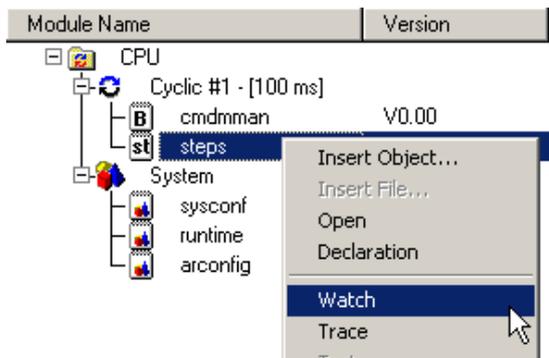


Fig.90 打开监视窗口

监视窗口的首要目的是观察和修正控制器的变量。除了变量值，其余的重要的信息也可以显示（数据类型、范围、I/O数据点等）。

变量可以以列表的形式来管理，不同的配置（变量组）也可以存储。

通过点击快捷菜单中Insert Variable... 可以在Watch（监视）窗口中添加选择任务的允许变量。

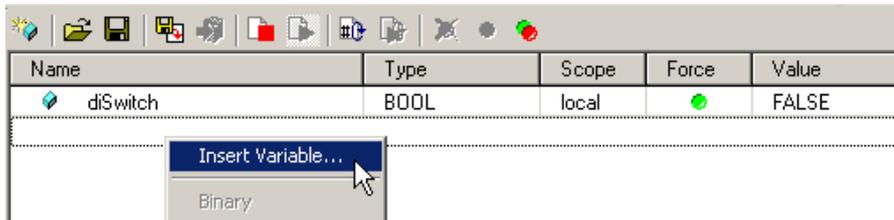
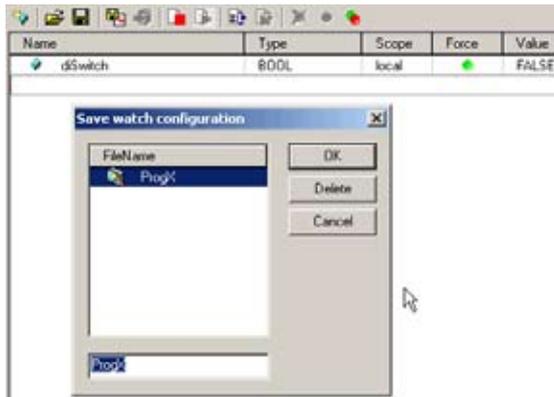


Fig.91 添加变量

在任何时候都可以保存、重新打开和重新存储Watch窗口的当前配置。你也可以用不同的名称下保存几个Watch设置。



用Watch窗口下的图标可以执行下面的操作（功能以从左到右的顺序列出）：



Fig.92 Watch窗口的工具栏

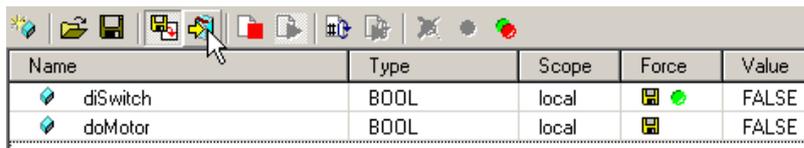
- 增加变量
- 装载配置
- 保存配置
- 存档模式
- 把存档数据写到PCC
- 结束任务
- 开始任务
- 设定循环数
- 执行循环数(只有当任务停止时才能执行)
- 取消强制
- 打开强制
- 取消全部强制

Automation Studio™ 调试诊断工具

9.7.2 存档

在监视窗口下改变一个变量是没有问题的。当确认输入后，新的变量马上传输到目标系统中。从这一刻起。任务以新的变量运行。

然而，如果你想改变多个变量，你可以用存档格式。当前的变量值在存档格式下不显示。只用一个命令（Write Values），你可以同时将所有值上载到目标系统中。你甚至可以把小配方（大量相互关联的变量值）放到一起并在任何时候都能装载和保存它们。当保存Watch的配置时，变量值也被保存。这个数据叫存档数据。



Name	Type	Scope	Force	Value
diSwitch	BOOL	local	 	FALSE
doMotor	BOOL	local		FALSE

Fig.93 存档

现在，保存的设置可以用作执行任何可行的附加处理。

存档数据被存在各自的Watch配置中。如果你打开这些配置中的一个，你会被建议进入存档模式。

备注:

你可以用Watch窗口和存档模式将重要的变量保存成文件以便以后需要时可以恢复。

9.8 轨迹跟踪

现在，不同的监视，行检测和Watch窗口使我们完全可以观察静态和曾经改变的变量。但是，也有一些变量，它们变化的太快，Watch窗口无法跟上它们的变化。因为它们的变化好像是随机的，所以很难确切的说出如何运行。于是就有了跟踪器（Tracer）。这个工具可以报告变量值随时间的变化并以图表的形式加以显示。图表结果可以存储并用作以后的观察和处理。

当系统监控启动时，可以通过点击 快捷菜单下的Trace来打开它。

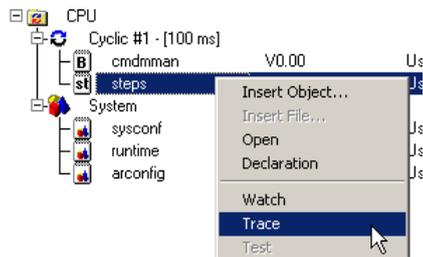


Fig.94 打开 tracer

轨迹跟踪记录也可被存成文件和传输到其它的相关方面。



Fig.95 轨迹跟踪记录

Automation Studio™ 调试诊断工具

9.9 NC诊断

在Automation Studio™ 中的Nc诊断工具可以执行日志指定的驱动参数和读取状态信息，以便于分析。这必须有确定的顺序，例如：定位，有选择的再调整。

9.9.1 Nc监视

Nc监视可以在列表中显示状态信息、实际值、目标值等。对分析起决定性作用的参数可以随时加进列表中。

通过选择工具栏的功能也可以显示驱动的错误信息。

Parameter	Wert	Einh.
x_achse.netzwerk.ini	ncWAHR	
x_achse.netzwerk.se		
x_achse.netzwerk.se	0	
x_achse.netzwerk.se	0	
x_achse.monitor.s	4182	[Einh.]
x_achse.monitor.v	5.960464	[Einh./s]
x_achse.meldung.an	0	
x_achse.meldung.an	0	
x_achse.meldung.sal	0	
x_achse.meldung.sal	0	
x_achse.meldung.sal	0	

Fig.96 Nc监视

选择工具栏的功能也可以显示驱动的错误

9.9.2 Nc跟踪

接下来，用户可以打开Nc跟踪来建立评估指定的某些驱动参数的记录。

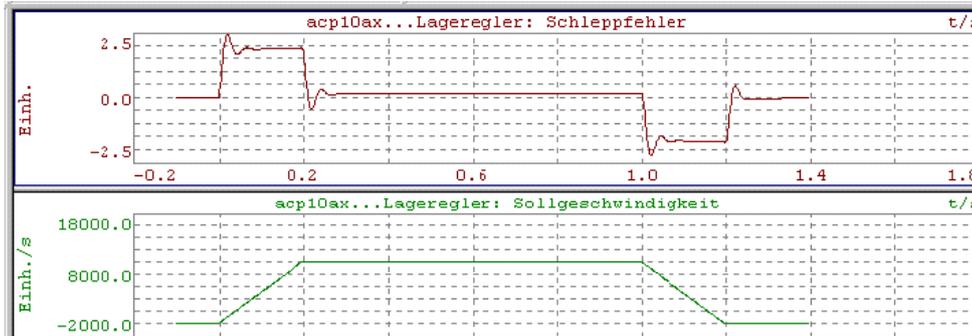


Fig.97 Nc跟踪

备注:

Nc跟踪可以在测试中心或直接在目标轴下打开。直接在目标轴下打开不会中断应用程序的运行。

对跟踪配置的描述可以在Automation Studio™ 在线帮助下的The B&R Software World – NC Software – ACP10 – Trace – Trace Configuration 找到。

9.9.3 NC测试 (NcTest)

NcTest 结合了驱动分析和测试的所有功能。NcTrace (循环或通信数据的轨迹)、NcWatch、参数接口和命令接口都在NcTest的范围内。

NcTest可以校准驱动的所有功能并可执行任务。在目标系统上不需要另外的控制器软件。



Fig.98 NcTest

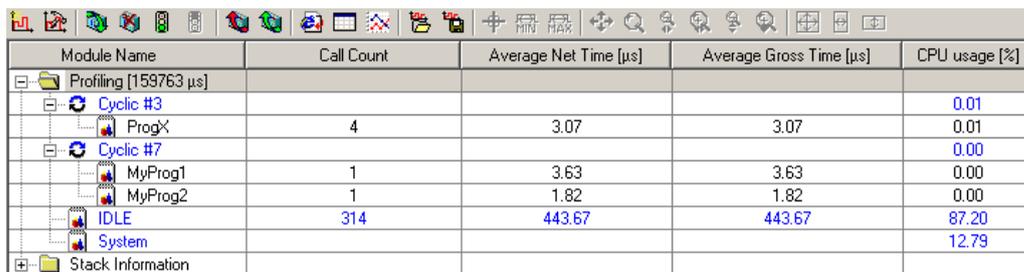
Automation Studio™ 调试诊断工具

9.10 PROFILER

用 Profiler可以测量和显示重要的系统文件:

- 任务运行时间
- 堆栈利用率
- 系统利用率

这样就可以根据系统利用率（负载）来分析系统运行时间。这个信息可以帮助优化系统如何使用资源。



Module Name	Call Count	Average Net Time [μs]	Average Gross Time [μs]	CPU usage [%]
Profiling [159763 μs]				
Cyclic #3				0.01
ProgX	4	3.07	3.07	0.01
Cyclic #7				0.00
MyProg1	1	3.63	3.63	0.00
MyProg2	1	1.82	1.82	0.00
IDLE	314	443.67	443.67	87.20
System				12.79
Stack Information				

Fig.99 Profiler 测量记录

测量结果的最大值、最小值和平均值显示在表格中。Profiler系统能以绘图的形式输出这些测量结果。配置和测量结果可以保存，如果需要，也可以以邮件形式发送。

备注:

在目标系统上的激活的Profiler配置也可以用“logging”库中的功能从软件中手动使能。

备注:

我们可以通过详细分析Profiler的测量数据，这些测量结果对软件开发人员是特别重要的。

9.11 调试器

调试器可以捕捉到在文本编程语言中隐藏的软件错误。通过在源程序中设置断点，可以使程序在设置断点的确切代码行停止执行。

这就可以决定变量值与当前被执行的程序行直接联结关系。使用 Step into（进入）和 Step over（跳过）功能键可以使你跳过函数或跳进函数内部对正在运行程序有一个详细的了解。

0001	(* cyclic program *)	
0002	CASE condition OF	condition = 0
0003	0: setVal := 1;	setVal = 1
0004	aoLimit := 255;	aoLimit = 255
0005	1: setVal := 2;	setVal = 1
0006	aoLimit := 256;	aoLimit = 255
0007	2: setVal := 3;	setVal = 1
0008	aoLimit := 267;	aoLimit = 255
0009	3: setVal := 4;	setVal = 1
0010	aoLimit := 268;	aoLimit = 255
0011	END_CASE	

Fig.100 在编程语言监控时设置断点

Step into 模式一行一行的检查程序。Step over把相关结构（循环、步序程序）当作单独的程序行来处理。

警告:

如果程序进入断点，整个应用程序停止。不推荐在正在实际生产的机器上尝试它。在重新启动时断点会被自动删除。

10、小结

Automation Studio™用来编程开发所有的贝加莱(B&R)自动化产品。



Fig. 101 Automation Studio™

通过这本《使用入门手册》，你已经熟悉了Automation Studio™并开始了解在你将来工作时，可以通过Automation Studio™在线帮助寻求帮助。

各种编程语言使你能选择最合适你应用的语言进行编程。

更详细的信息可参加贝加莱定期开设的培训课程，具体课程内容如下：

课程编号：ASINT

参加条件：具有一般的电气工程技术知识，具有WINDOWS的操作经验。

目的：熟悉B&R控制系统的特点及硬件配置，能够使用不同的编程语言编写和测试一个简单的应用,学习有关设备的维护。

课程内容：Automation Studio (Windows)基础课程

- ◎ 系统概述
- ◎ 控制系统的硬件配置
- ◎ 编程环境 ---- Automation Studio
- ◎ Automation Runtime ----B&R2000的操作系统
- ◎ 编程语言（梯形图LAD、指令表IL、结构文本ST、B&R AB、ANSI C）
- ◎ 库管理器
- ◎ 维护信息
- ◎ 项目管理

时 间：四天

课程编号：ASPROG

参加条件：参加过ASINT培训。

目的：创建和编写复杂的应用项目。

课程内容：Automation Studio(Windows) 中级课程

- ◎ 数据类型及寻址方式
- ◎ 数据的处理
- ◎ 中断任务和例外任务
- ◎ 网络通信(本地扩展、远程扩展、CAN、FRAME DRIVER、ETHERNET)
- ◎ Automation NET
- ◎ TPU功能
- ◎ 项目的编写规则

时间：四天

课程编号：ACOPOS

参加条件：参加过ASPROG培训。

目的：熟悉ACOPOS系列产品，创建、编写定位控制项目及驱动参数调试。

课程内容：ACOPOS定位系统

- ◎ 定位系统概述
- ◎ 传动原理
- ◎ 驱动规格
- ◎ NC定位概念
- ◎ 运动组件 ---- Motion Component
- ◎ ACP10单轴软件
- ◎ 驱动参数调试
- ◎ 多轴定位

时间：三天

小结

课程编号：VISCOMP

参加条件：参加过ASINT培训。

目的：创建、编写图文显示应用项目。

课程内容：Visual Component 课程

- ◎ 人机界面系统概述
- ◎ 可视化组件 ---- Visual Component
- ◎ 创建画面
- ◎ 画面的编辑工具
- ◎ 报警系统
- ◎ 参数管理
- ◎ 多语言切换
- ◎ VISAPI函数的应用

时间：二天

培训报名方式：

1. 选择培训课程：客户应根据设备情况，参加人员的技术基础选择合适的课程，请参看培训课程描述。

2. 选择培训地点：贝加莱在全国有四个培训中心，在上海和西安定期开课，在济南、广州和北京我们也可应客户要求由适当数量的学员组班开课。客户可根据需要选择地点。但请注意：各培训中心的课程种类有所不同，每个中心的开课情况请参看培训日期安排表。

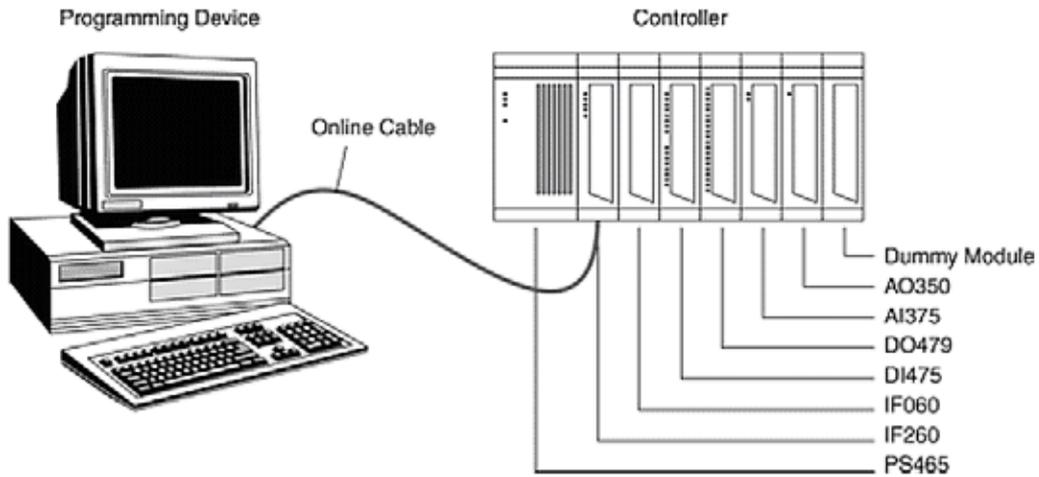
3. 选择培训时间：上海和西安培训中心有定期开课计划，可以电话垂询各个培训中心或通过贝加莱网站（www.br-automation.cn）得到最新的培训计划及详细时间安排。我们也可应客户要求由适当数量的学员组班不定期开课。

4. 报名：为了保证培训质量，学员有足够的上机操作练习的时间，每期培训都有人数限制，因此学员应尽可能提前报名参加培训。填好“培训回执表”并由公司盖章后，传真发给中心的培训联系人。收到您的培训回执后，我们将尽快确认开课日期。若该课程不能如期召开，我们将及时通知申请人并安排其他开课日期。通常，开课前学员会收到一份开课通知，即确认函，告知上课的有关信息。

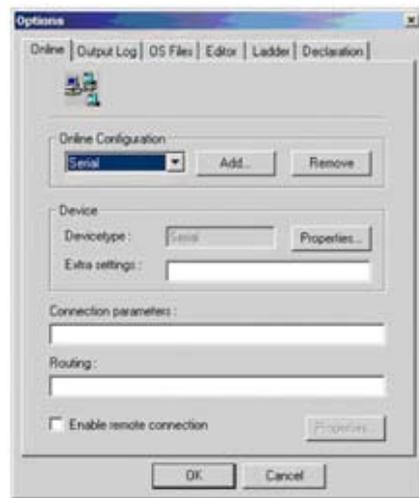
11、附录

11.1 编程计算机和B&R控制器之间建立连接

在编程计算机和控制器之间建立连接需使用一根标准的RS232连接电缆（Model No.: 0G0001.00-090）。



连接电缆将编程计算机的串行接口和控制器的RS232接口连接，并保证控制器的电源供电打开。然后通过Automation Studio™的Tools/Options 选项进行在线连接配置。



并且通过状态行显示信息，判断是否激活连接：

连接激活显示： Line 1 of 3 COM2 IF260 V2.29 RUN

离线显示： Line 1 of 3 COM2 OFFLINE

11.2 I/O变量的申明定义

B&R 控制系统中对于I/O变量采用的是符号性的变量名，通过变量申明表与实际I/O信号建立连接。

在硬件配置树上，选中你所需要定义的I/O模块，在软件窗体中将显示变量的定义：

I/O			
Name	Data Type	PV Name	Remark
digital input 01	BOOL	Key_1	10ms switching delay
digital input 02	BOOL		10ms switching delay
digital input 03	BOOL		10ms switching delay
digital input 04	BOOL		10ms switching delay
digital input 05	BOOL		10ms switching delay
digital input 06	BOOL		10ms switching delay
digital input 07	BOOL		10ms switching delay
digital input 08	BOOL		10ms switching delay
digital input 09	BOOL		10ms switching delay
digital input 10	BOOL		10ms switching delay
digital input 11	BOOL		10ms switching delay
digital input 12	BOOL		10ms switching delay
digital input 13	BOOL		10ms switching delay
digital input 14	BOOL		10ms switching delay
digital input 15	BOOL		10ms switching delay
digital input 16	BOOL		10ms switching delay

在 PV Name 项，你可键入你对应不同通道的符号性变量名，或通过鼠标双击从项目的数据库变量中选择符合定义要求的变量。

这样，你所定义的符号性变量名，就和外部的I/O物理端口建立了连接。

11.3 操作系统的安装

B&R Automation Runtime操作系统存储在控制器的系统ROM中，你可以使用B&R Automation Studio™下载或更新控制器的操作系统。

控制器系统有不同的运行模式，可以通过控制器上的拨码设置：

BOOT 模式：通常控制器的拨码设置在 0 位置时

DIAGNOSE 模式：通常控制器的拨码设置在 F 位置时

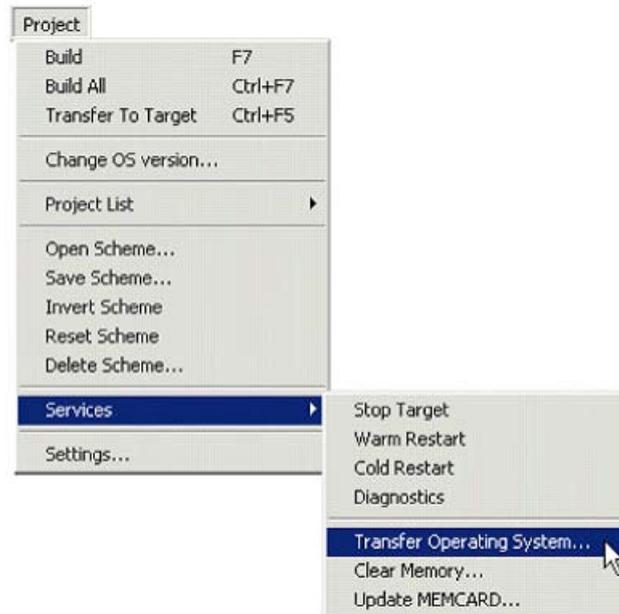
RUN 模式：通常控制器的拨码设置在 非0或非F 位置时

具体可查阅相关模块的帮助信息或文档。

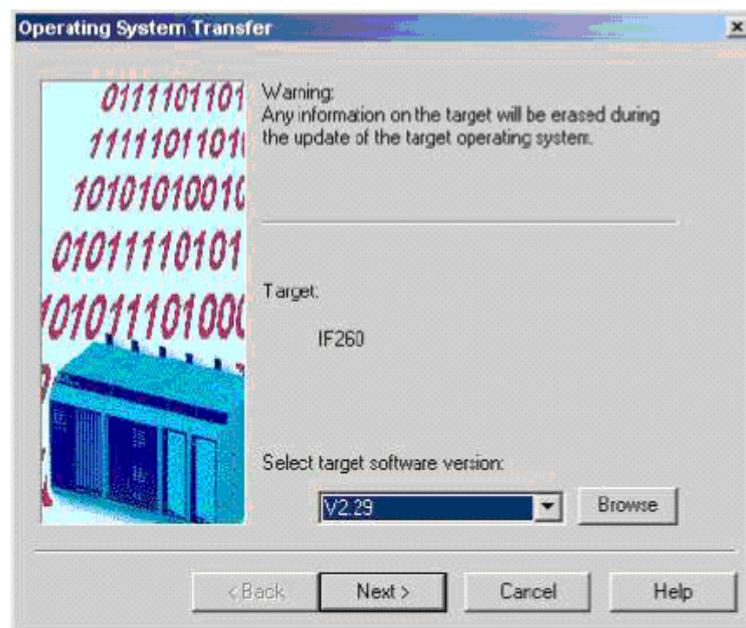
操作系统的安装可以在BOOT 模式或 RUN模式下：

1. 控制器和计算机之间建立连接
2. 运行B&R Automation Studio™
3. 通过File/ Open Project 命令打开与当前连接控制器硬件模块型号一致的项目

4、选择Project / Service / Transfer Operating System 命令，打开下载操作系统对话框

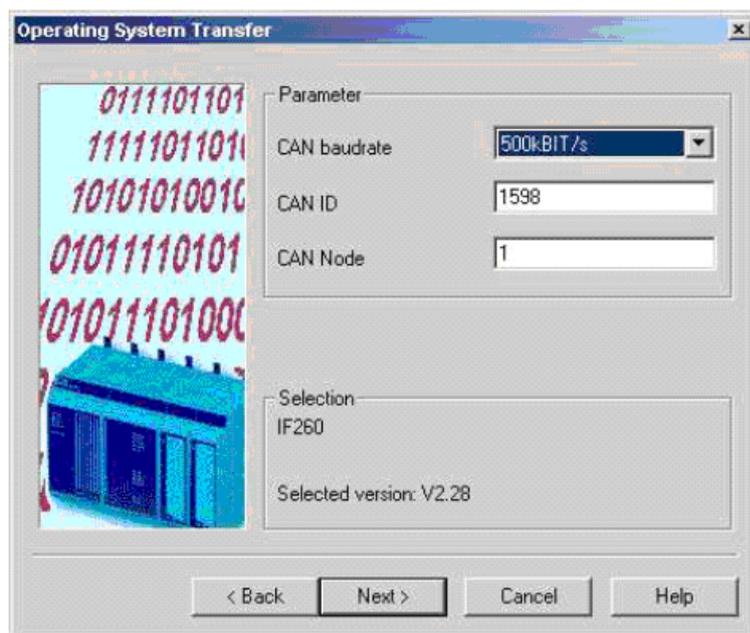


5、选择Automation Runtime的版本，点击 Next



附录

6、设置CAN通信的参数，如果你的控制器在CAN通信网络中需要其他设置值时，在此修改。



7、设定CAN参数后，选择 Next ，Flash PROM 上的内容被清除，选择的操作系统的版本被传送到Flash PROM 上，整个下载进度在屏幕中显示，需要几分钟时间。

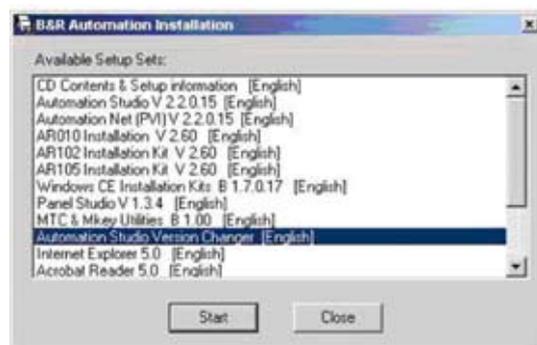
8、如果你是在BOOT模式下下载的，完成后将控制器设置成RUN模式。

11.4 B&R Automation Studio™ 多版本安装

从2.2版本开始，Automation Studio™就支持多版本安装，我们可以使用 版本切换器在多个版本之间切换，同一时间只有一个版本是被激活的。

版本切换器的安装

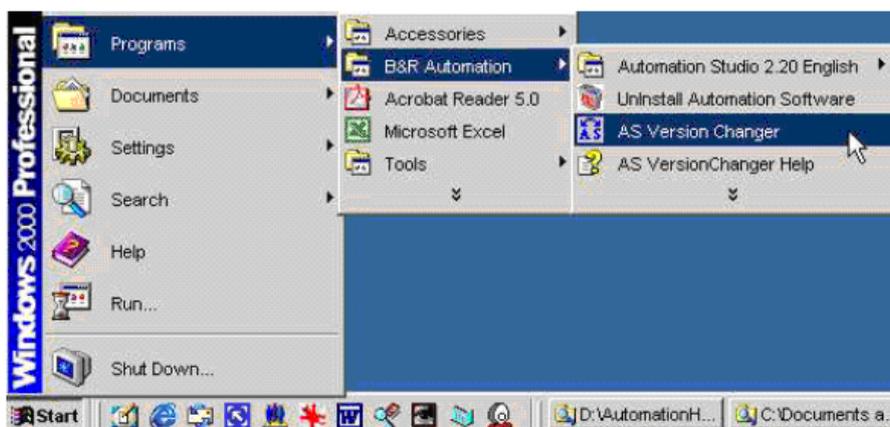
每张 Automation Studio™ 安装CD读取时会自动弹出一个安装菜单，也可以通过点击CD中 "BrMenu.exe"文件来打开该窗体



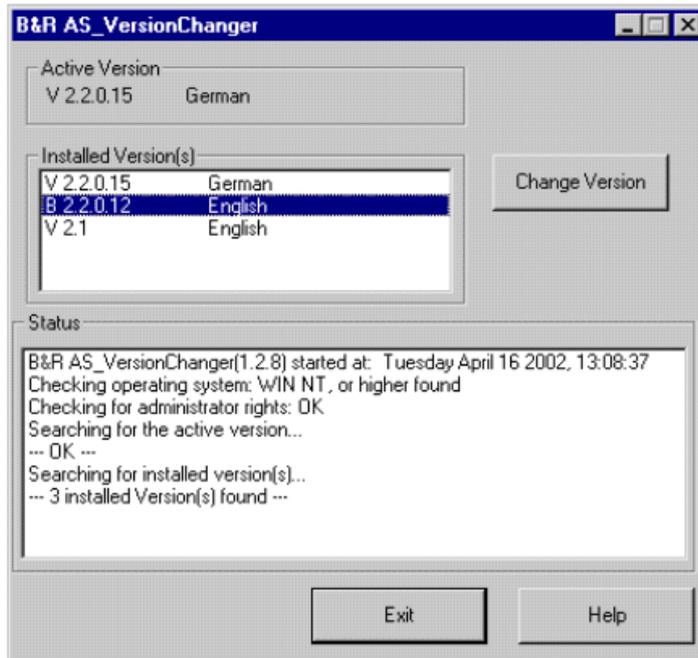
选择 Automation Studio Version Changer 的安装。

版本切换过程

从开始菜单中启动 AS Version Changer



打开 AS Version Changer 的对话框，通过 UP/DOWN 键盘选择你使用的版本，点击 Change Version 按钮，几分钟后将完成版本的切换。



Exit 按钮关闭 Version Changer 对话框。

11.5 模块化培训手册一览

TM200 – B&R Company Presentation **
TM201 – B&R Product Spectrum **
TM210 – The Basics of Automation Studio
TM211 – Automation Studio Online Communication
TM212 – Automation Target **
TM213 – Automation Runtime
TM220 – The Service Technician on the Job *
TM221 – Automation Components and Sources of Errors *
TM223 – Automation Studio Diagnostics
TM230 – Structured Software Generation
TM231 – Automation Studio Machine Variations *
TM240 – Ladder Diagram (LAD)
TM241 – Function Block Diagram (FBD) *
TM242 – Continuous Function Chart (CFC) *
TM243 – Sequential Function Chart (SFC) *
TM245 – Instruction List (IL) *
TM246 – Structured Text (ST)
TM247 – Automation Basic (AB) *
TM248 – ANSI C
TM250 – Memory Management and Data Storage
TM260 – Automation Studio Libraries I

TM261 – Automation Studio PID Library *
TM400 – The Basics of Motion Control *
TM401 – B&R Motion Control Products **
TM402 – Dimensioning Motion Control Systems *
TM410 – The Basics of ASiM
TM440 – ASiM Basic Functions
TM441 – ASiM Multi-Axis Functions
TM442 – ACOPOS Smart Process Technology *
TM445 – ACOPOS ACP10 Software
TM450 – ACOPOS Control Concept and Adjustment
TM460 – Starting up B&R Motors *
TM461 – Starting up Motors from 3rd Party Manufacturers *
TM470 – CNC *
TM600 – The Basics of Visualization
TM601 – B&R HMI Products **
TM610 – The Basics of ASiV
TM620 – Visual Components Service *
TM630 – Visualization Programming Guide
TM640 – ASiV Alarm System
TM650 – ASiV Internationalization
TM660 – ASiV Remote
TM670 – ASiV Advanced
TM800 – APROL System Concept
TM801 – APROL Basis Engineering
TM810 – APROL Installation, Configuration and Recovery *
TM811 – APROL Runtime System *
TM812 – APROL Operator Management
TM813 – APROL XML Queries and Audit Trail*
TM820 – APROL Service *
TM830 – APROL Project Engineering
TM840 – APROL Parameter Management and Recipes
TM850 – APROL Controller Configuration and INA
TM860 – APROL Library Engineering
TM861 – APROL Couplings *
TM865 – APROL Library Guide Book
TM870 – APROL Python Programming *
TM880 – APROL Report *
*) 即将提供
**) 参看产品目录

全球总部

Bernecker+Rainer Industrie-Elektronik Ges.m.b.H.

B&R Straße 1

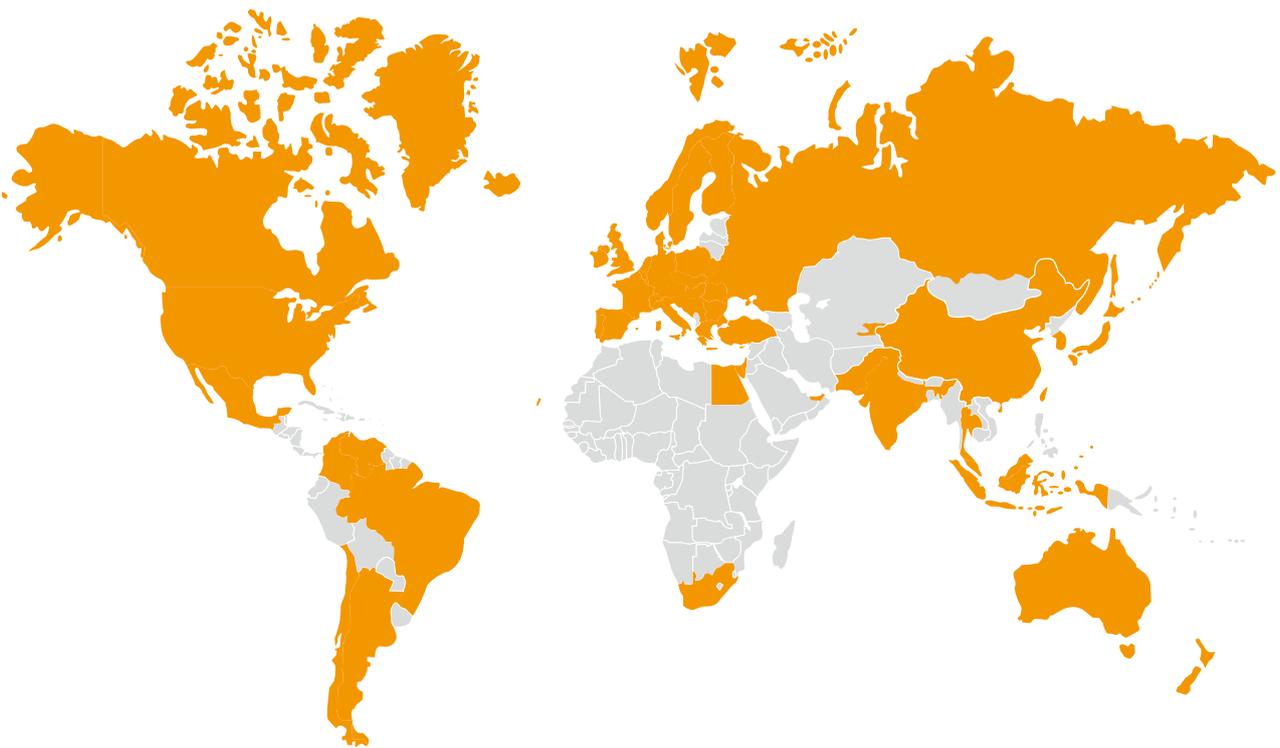
A-5142 Eggelsberg 奥地利

中国总部

贝加莱工业自动化（上海）有限公司

上海市田林路487号宝石园21号楼

全球60个国家155个分支机构



Austria · Australia · Belgium · Belarus · Brazil · Bulgaria · Canada · Chile · China · Croatia · Cyprus · Czech Republic
Denmark · Egypt · Emirates · Finland · France · Germany · Greece · Hungary · India · Indonesia · Ireland · Israel · Italy · Korea
Kyrgyzstan · Malaysia · Mexico · The Netherlands · Norway · Pakistan · Poland · Portugal · Romania · Russia · Singapore
Slovakia · Slovenia · South Africa · Spain · Sweden · Switzerland · Thailand · Turkey · Ukraine · United Kingdom · USA