

# **B&R SYSTEM 2000**

## **B&R AUTOMATION STUDIO™**

### **Quick Start**

Version: **1.0 (Oct. 1999)**  
Model No.: **MASYS2ASQS-E**

Pentium® is a registered trademark of the Intel Corp.  
Windows® is a registered trademark of the Microsoft Corp.  
Windows® 95 is a registered trademark of the Microsoft Corp.  
Windows® 98 is a registered trademark of the Microsoft Corp.  
Windows® NT 4.0 is a registered trademark of the Microsoft Corp.

We reserve the right to change the contents of this manual without warning. The information contained herein is believed to be accurate as of the date of publication, however, Bernecker and Rainer Industrie-Elektronik Ges.m.b.H. makes no warranty, expressed or implied, with regards to the products or the documentation contained within this book. Bernecker and Rainer Industrie-Elektronik Ges.m.b.H. shall not be liable in the event of incidental or consequential damages in connection with or arising from the furnishing, performance or use of these products.

**1. INTRODUCTION / INSTALLATION**

**2. THE FIRST STEPS**

**A. FREQUENTLY ASKED QUESTIONS**

**B. GLOSSARY**

**C. INDEX**



# Table of Contents

## Chapter 1

<b>Introduction / Installation .....</b>	<b>7</b>
1. Introduction .....	8
1.1 Most Important Tools .....	9
1.2 What can I find in this manual ? .....	11
1.3 Conventions .....	12
2. Where do I find answers to my questions ? ..	13
2.1 Online Help .....	13
2.2 This Manual .....	13
2.3 Service and Support .....	13
3. Installation .....	14
3.1 System Requirements .....	14
3.2 Installation in Windows 95 or Windows 98 .....	15
3.3 Installation in Windows NT .....	17

## Chapter 2

<b>The First Steps .....</b>	<b>19</b>
1. First Program Start .....	20
1.1 Starting B&R Automation Studio™ .....	21
1.2 Exiting B&R Automation Studio™ .....	23
1.3 Keyboard and Mouse Operation .....	23
2. Example Program .....	24
2.1 Task Description .....	24
2.2 Hardware Requirements .....	24
3. How do I create a project? .....	26
3.1 Preparation .....	26
3.2 How do I make the connection between the programming device and PCC? .....	27
3.3 Establishing the Online Connection .....	28
3.4 Creating a New Project .....	30
3.5 Connection Status .....	35
3.6 Work Environment .....	36
4. Variable Declaration .....	38
4.1 Digital Input .....	38
4.2 Digital Output .....	39
5. Creating a Cyclic Object .....	40
5.1 Inserting a Cyclic Object .....	40
5.2 Define Object Name, Object Type and Resource .....	42
5.3 Entering the Ladder Diagram .....	45
6. Compiling and Transferring Objects to the Controller .....	49
7. Testing the Object .....	52
7.1 Ladder Diagram Monitor .....	52
7.2 System Monitor .....	55
7.3 Watch .....	58

## Appendix A

<b>Frequently Asked Questions .....</b>	<b>63</b>
1. How do I install the PCCSW Operating system? .....	64
1.1 Processors with Application Memory .....	64
1.2 Processors without Application Memory .....	68
2. How can I work without hardware? .....	71

## Appendix B

<b>Glossary .....</b>	<b>73</b>
-----------------------	-----------

## Appendix C

<b>Index .....</b>	<b>81</b>
--------------------	-----------



# CHAPTER 1

## INTRODUCTION / INSTALLATION

# 1. Introduction

---

B&R Automation Studio™ provides you with a development system for all automation tasks.

B&R Automation Studio™ characteristics:

## Microsoft Windows

### Look & Feel

The B&R Automation Studio™ desktop strictly complies with the Microsoft design guidelines for Windows programs. This makes it easier to become familiar with the program and reduces project engineering costs.

## Supports many

### Target Systems

The B&R Automation Studio™ development environment supports many different target systems. This allows the automation platform to be scaled to meet your exact requirements.

## More than IEC 1131

In addition to IEC 1131 languages, B&R Automation Studio™ offers high level language programming using ANSI C and complete line of high performance technology functions which greatly simplify project engineering for your machines and systems.

## Start-up

All diagnosis and service tools clearly show system behavior using graphic displays. Synchronous recording of data replaces an oscilloscope and allows exact settings to be made for machine parameters.

## Contains all

### Information

B&R has completely integrated all product information that is relevant during the creation and development of your B&R Automation Studio™ projects. Standard help functions provide easy access of all details related to the operation of this tool. Hardware specific data is placed in user-friendly registers.



## 1.1 Most Important Tools

The following pages contain an overview of the most important tools included in B&R Automation Studio™ and their functionality.

### Hardware Configuration

Hardware Configuration Characteristics:

- Automatic recognition of the target system
- Graphic display of the target system
- Detailed information about the target system
- Simple definition of target system profiles
- Direct assignment of symbolic variables to physical terminals

### Software Configuration

Software Configuration Characteristics:

- Efficient configuration of the target system
- Clear display of project hierarchy
- Simple definition of software profiles

### Programming Languages

Ladder Diagram ..... IEC 1131 (abbr.: LAD)  
Instruction List ..... IEC 1131 (abbr.: IL)  
Structured Text ..... (abbr.: ST)  
Sequential Function Chart .... (abbr.: SFC)  
B&R Automation Basic ..... Previously "PL2000" (abbr.: AB)  
C Language ..... ANSI C

### Libraries and Function Blocks

Many standard functions (function blocks) are included in the delivery of B&R Automation Studio™ which are grouped together in various libraries. Each library contains function blocks which can save you a lot of time and effort when creating solutions to standard problems.

## Technology Functions

Modular technology plug-ins allow homogenous integration of high performance tools.

- Set, analyse and diagnose axis / drive parameters
- CNC editor
- CAM editor

## Diagnosis, Service, etc.

- Online variable force
- Online variable monitor
- Realtime trace
- Complete source level debugger with break point, single step, line coverage and disassembler functions
- Logbook entries for error analysis

## 1.2 What can I find in this manual ?

This manual provides you with an introduction to B&R Automation Studio™ allowing you to get started quickly. It will also provide an overview and helpful tips.

Don't be shocked by the size of this manual ! You should only need approx. 1 to 2 hours to work through the most important chapters (Chapter 1 "Introduction / Installation" and Chapter 2 "The First Steps").

### Chapter 1

Chapter 1 "Introduction / Installation" provides you with all of the important information required to install B&R Automation Studio™.

### Chapter 2

Chapter 2 "The First Steps" uses examples to introduce you to the functions available in B&R Automation Studio™. If you work through this chapter, you will become familiar with B&R Automation Studio™ and also collect a lot of useful information.

### Appendix A

Appendix A "Frequently Asked Questions" gives answers to the most frequent questions as well as solutions to common problems.

### Appendix B

Appendix B "Glossary" contains a listing of important terms and their descriptions.

### Appendix C



Appendix C "Index" allows you to find the information you are looking for quickly.

## 1.3 Conventions

In order to be able to find and interpret information quickly and easily in this manual, visual orientation guidelines such as standard text formats, special terms symbols are used. The following conventions are used.

### Visual Orientation

In this manual, you will continually find the following typographic conventions:

- CAPITALS**  
Directory names, file names, etc. are shown in capitals.
- Bold Upper / Lower Case Letters**  
Menu items, command names, names of dialog fields, buttons and options are shown in bold.
- Bold Italics**  
Important terms and information are shown in bold italics.
- "In Quotation Marks"  
All text that has to be entered in fields are shown in quotation marks.
- SMALL CAPITALS**  
Names of keys on your keyboard, e.g. CTRLKEY, ESC, etc.
- Important notes and tips are shown using the  symbol.
- Explanations to new terms are shown using the  symbol.

### Key Combinations and Sequences

Key combinations and sequences are differentiated as follows:

- KEY1+KEY2**  
A plus (+) between key names means that the first key is held down while the second is pressed.
- KEY1, KEY2**  
A comma (,) between key names means that both keys are pressed and released in sequence.

## 2. Where do I find answers to my questions ?

---

### 2.1 Online Help

---

The B&R Automation Studio™ online help can be called any time, providing you with information concerning the operation and function of B&R Automation Studio™. To start the online help, click on the **Help** menu item found in the **?** menu.

If you need help concerning the currently active window or dialog box, simply press the **F1** key.

### 2.2 This Manual

---

In addition to basic information about B&R Automation Studio™, this manual also contains descriptions of often used procedures. Appendix A "Frequently Asked Questions", contains answers to often asked question and Appendix B "Glossary" contains a listing of important terms and their meanings.

### 2.3 Service and Support

---

B&R will be there for you when you need information or advice. If you have questions that are not answered in our documentation, get in touch with your service partner at B&R.

#### Internet / E-Mail

B&R provides an Internet page at the following address: <http://www.br-automation.com>. This address provides you with information concerning our products and which email addresses you can use to reach us. New program versions and updates are also available on our WWW pages.

#### Mailbox

Newest program versions, drivers, libraries, etc. can be found in our company mailbox. The B&R Mailbox can be reached with the following numbers:

1. +43 (77 48) 65 86 - 623
2. +43 (77 48) 65 86 - 86

## 3. Installation

---

In this section, you will find all of the important information you will need for the installation of B&R Automation Studio™.

Read the following notes before you start the installation:

- Before the installation, check if your PC meets the criteria listed in section 3.1 "System Requirements".
- Close all other Windows programs.
- Microsoft Internet Explorer 4.01 (or higher) is required to use B&R Automation Studio™ in Windows. Install Internet Explorer first and then B&R Automation Studio™ (see section 3.2 "Installation in Windows 95 or Windows 98" or 3.3 "Installation in Windows NT").

---

### 3.1 System Requirements

The computer where you are going to install B&R Automation Studio™ has to meet the following requirements:

	Minimum	Recommended
<b>Operating System</b>	Windows 95, Windows 98 or Windows NT 4.0	
<b>Processor</b>	80486 100 MHz	Pentium 166 MHz
<b>Working Memory</b>	16 MB	64 MB
<b>Hard Disk</b>	Approx. 80 MB (full version)	
<b>Graphics</b>	SVGA (800x600)	XGA (1024x768)

Your computer has to have a free online interface for the online connection between your programming device (computer) and the target system (controller). For example, a free serial interface (COM1, COM2, COM3 or COM4) or CAN interface can be used as online interface.

### 3.2 Installation in Windows 95 or Windows 98


Microsoft Internet Explorer 4.01 (or higher) has to be installed before installing B&R Automation Studio™. The installation software for Internet Explorer is available on the B&R Automation Studio™ installation CD.



If you already have Microsoft Internet Explorer Version 4.01 (or higher) on your computer, you don't have to carry out the following steps.

#### Microsoft Internet Explorer

Microsoft Internet Explorer is installed as follows in Windows 95 or Windows 98:

1. Start Windows 95 or Windows 98.
2. Place the installation CD in the CD-ROM drive in your computer. Click on the **Start**  button and select **Run...** To install the English Version of Microsoft Internet Explorer, enter "E:\IE401\ENGLISH\IE4SETUP.EXE" in the command line of the **Run** dialog box and click on the **OK** button. To install the German Version of Microsoft Internet Explorer, enter "E:\IE401\DEUTSCH\IE4SETUP.EXE" in the command line of the **Run** dialog box and click on the **OK** button.

The letter E in

"E:\IE401\ENGLISH\IE4SETUP.EXE"

(or "E:\IE401\DEUTSCH\IE4SETUP.EXE")


stands for the drive designation of your CD-ROM drive. Naturally, this drive can also have a different drive designation !

The installation program is started. The installation procedure is menu driven, i.e. you only have to follow the steps shown on the screen.

Additional information concerning the Microsoft Internet Explorer can be found in the Microsoft documentation.

B&R Automation Studio™ is installed as follows in Windows 95 or Windows 98:

1. Start Windows 95 or Windows 98.
2. Place the installation CD in the CD-ROM drive in your computer. The autoplay function for the CD-ROM drive automatically starts the installation program on the CD.

If the installation program is not started automatically, click on the **Start**  button and select **Run...** Enter "E:\SETUP\SETUP.EXE" in the command line of the **Run** dialog box and click on the **OK** button.

The letter E in "E:\SETUP\SETUP.EXE" stands for the drive designation of your CD-ROM drive. Naturally, this drive can also have a different drive designation !

The installation program is started. The installation procedure is menu driven, i.e. you only have to follow the steps shown on the screen.



The installation procedure can take a few minutes !



### 3.3 Installation in Windows NT


Microsoft Internet Explorer 4.01 (or higher) has to be installed before installing B&R Automation Studio™. The installation software for Internet Explorer is available on the B&R Automation Studio™ installation CD.



If you already have Microsoft Internet Explorer Version 4.01 (or higher) on your computer, you don't have to carry out the following steps.

#### Microsoft Internet Explorer

To install Microsoft Internet Explorer in Windows NT, you need network administrator rights. If necessary, get in touch with your network administrator.

1. Start Windows NT.
2. Log on as network administrator.
3. Place the installation CD in the CD-ROM drive in your computer. Click on the **Start**  button and select **Run....** To install the English Version of Microsoft Internet Explorer, enter "E:\IE401\ENGLISH\IE4SETUPEXE" in the command line of the **Run** dialog box and click on the **OK** button. To install the German Version of Microsoft Internet Explorer, enter "E:\IE401\DEUTSCH\IE4SETUPEXE" in the command line of the **Run** dialog box and click on the **OK** button.

The letter E in

"E:\IE401\ENGLISH\IE4SETUPEXE"

(or "E:\IE401\DEUTSCH\IE4SETUPEXE")

stands for the drive designation of your CD-ROM drive. Naturally, this drive can also have a different drive designation !


The installation program is started. The installation procedure is menu driven, i.e. you only have to follow the steps shown on the screen.

Additional information concerning the Microsoft Internet Explorer can be found in the Microsoft documentation.

## B&R Automation Studio™

To install B&R Automation Studio™ in Windows NT, you need network administrator rights. If necessary, get in touch with your network administrator.

1. Start Windows NT.
2. Log on as network administrator.
3. Place the installation CD in the CD-ROM drive in your computer. The autoplay function for the CD-ROM drive automatically starts the installation program on the CD.

If the installation program is not started automatically, click on the **Start**  button and select **Run....** Enter "E:\SETUP\SETUPEXE" in the command line of the **Run** dialog box and click on the **OK** button.

The letter E in "E:\SETUP\SETUPEXE" stands for the drive designation of your CD-ROM drive. Naturally, this drive can also have a different drive designation !

The installation program is started. The installation procedure is menu driven, i.e. you only have to follow the steps shown on the screen.



The installation procedure can take a few minutes !

# CHAPTER 2

## THE FIRST STEPS

# 1. First Program Start

---

In the following sections of this chapter, we will assume that the following standard settings were entered during the installation of B&R Automation Studio™:

Path to target directory for the installation:


C:\BRAUTOMATION

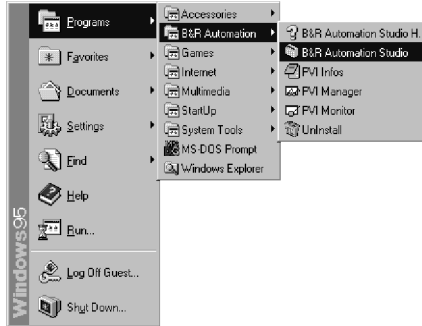
Name of the B&R Automation Studio™ program group:

B&R Automation

## 1.1 Starting B&R Automation Studio™

B&R Automation Studio™ is started in the following manner:

- Click on the **Start**  button and go to the **Program** directory.
- Then go to the **B&R Automation** directory and click on the program **B&R Automation Studio**:

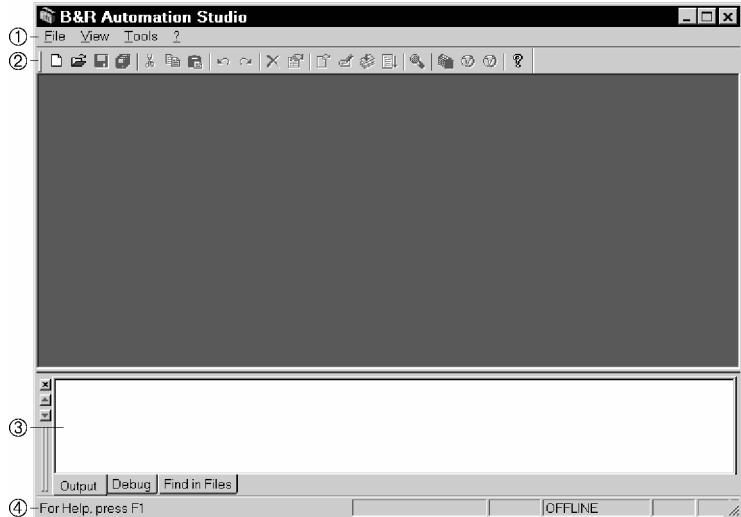


After starting B&R Automation Studio™, the start screen is shown for a short time and then B&R Automation Studio™ is opened.

### Start Screen



B&R Automation Studio™ window:



### ① **Main Menu**

The B&R Automation Studio™ main menu changes according to the active editor. In this way, you can select from the functions needed for the active editor.

### ② **Tool Bar**

Contains tool bar icons that provide fast access to various commands and functions. If you place the mouse pointer over the icons, the name is shown. Use the **Toolbars...** command in the **View** menu to show or hide tool bars

### ③ **Message Window**

The message window is shown in the lower part of the program window after the first time B&R Automation Studio™ is started. The message window shows e.g. compiler and debugger messages.

### ④ **Status Bar**

The status bar at the bottom of the window shows information such as:


- Short help information for menu commands or tool bar icons.
- Short information concerning editing procedures.
- Status of the online connection between the programming device and the target system (see section "3.4 Creating a New Project").

## 1.2 Exiting B&R Automation Studio™

---

Exit B&R Automation Studio™ by ...

... clicking on **Exit** in the **File** menu, or

... clicking on the  button in the top right corner of the B&R Automation Studio™ window.

## 1.3 Keyboard and Mouse Operation

---

The B&R Automation Studio™ desktop strictly complies with the Microsoft design guidelines for Windows programs. Keyboard and mouse operation also correspond to the Windows standard. If you have already worked with Windows 95, Windows 98 or Windows NT, you will be able to find your feet quickly in B&R Automation Studio™. Additional information concerning the operation of B&R Automation Studio™ can be found in the online help. To start online help, select **Help** from the **?** menu or press F1.

## 2. Example Program

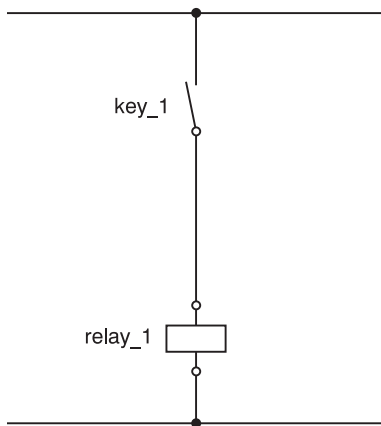
---

In the following sections, we will get the ball rolling by writing a short example program in B&R Automation Studio™ using the ladder diagram (LAD) programming language.

### 2.1 Task Description

---

We will create a project called "AS\_QS" in directory "C:\PROJECTS". Then we will open the project and enter the circuit plan shown in the following illustration using LAD.

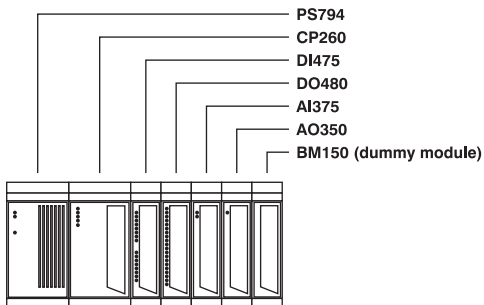


This example program be saved in the project "AS\_QS" under the name "logic1".

### 2.2 Hardware Requirements

---

In this manual, the following hardware is used as the target system for all example programs:





PS794 ..... Power supply module  
CP260 ..... CPU  
DI475 ..... Digital input module  
DO480 .... Digital output module  
AI375 ..... Analog input module  
AO350 ..... Analog output module  
BM150 .... Dummy module (to fill unused slots)

Many pictures and information in the following sections refer to this hardware configuration.

However, you can also use different hardware. To run this example program, you will need at least the following hardware components:

- Power supply module
- CPU
- Digital input module
- Digital output module

Without the hardware components listed above, you can program the example but you can't test it.

## 3. How do I create a project?

---

A project contains all information for a system / machine, including all programs and the corresponding data banks. In order to be able to enter an example program, you have to create a project.

### 3.1 Preparation

---

B&R Automation Studio™ is a hardware oriented programming system. That means the hardware has to be defined when creating a project. B&R Automation Studio™ can carry out automatic hardware recognition when creating a project. For inexperienced users, this is the simplest type of hardware definition.

The online connection has to be configured properly so that B&R Automation Studio™ can determine the hardware configuration on the target system. Before creating a project, check the connection between the programming device (computer) and target system (controller). In order to establish an online connection, the PCCSW operating system has to be installed on the target system. The controller **cannot** be used without the PCCSW operating system !



The PCCSW operating system (PCCSW stands for PCC software) is an adjustable, deterministic realtime-multitasking operating system.

Before continuing with the next section, make sure that the PCCSW operating system is installed on the target system. The PCCSW operating system is not installed in the following cases:

- For processors **with application memory** (CP10x, CP2x0, MP100 and CP15x): All application memory modules are delivered without an operating system. Therefore the PCCSW operating system is not installed on new application memory modules.
- For processors **without application memory** (CP260, XP152, IF152, IF260, IPxxx, CP47x, CP77x, etc.): Processor modules without application memory are delivered without an operating system installed. This status is indicated by the following switch settings:
  - CAN Node Number Switch** in position 00 (e.g. XP152, IF152 and IP151).
  - Operating Mode Switch** in position 0 (e.g. CP260) or 00 (e.g. IP350).
  - MODE Switch** in position 0 (e.g. CP47x and CP77x).

You can also have a look in the hardware documentation for additional information.

If the operating system is not installed, an operating system download has to be carried out. An exact description of this procedure can be found in Appendix A "Frequently Asked Questions", Section 1 "How do I install the PCCSW operating system?".

### 3.2 How do I make the connection between the programming device and PCC?

An online cable (RS232 cable) is delivered with B&R Automation Studio™ for the connection between the programming device and controller. This cable can also be ordered from B&R separately:

**Online Cable** for connection between programming device (computer) and target system (controller), **Model Number 0G0001.00-090**.

Take note of the following points when making an online connection using this cable:

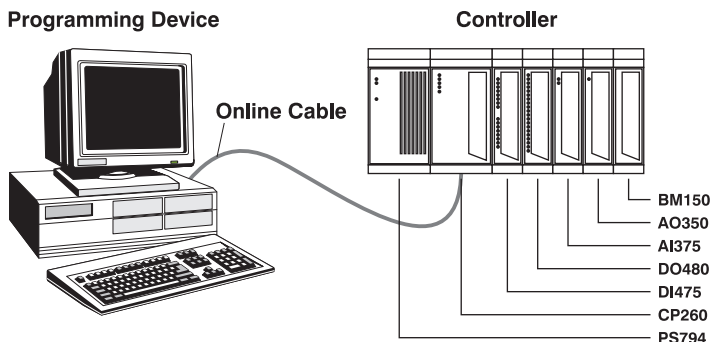
- COM1, COM2, COM3 or COM4 has to be free on your computer.
- The online cable has two 9 pin D-type sockets. The serial interfaces on some computer use a 25 pin D-type plugs, therefore you might need an adapter. These type of adapters are often included with a mouse. Otherwise you can get an adapter from a local supplier.
- The online cable is only for a connection between two RS232 interfaces. Do not use an adapter to connect the serial interface COMx with an RS485/RS422 interface. You need a special interface converter to do this !

If your computer and the controller both have a CAN interface, communication can also take place via CAN.

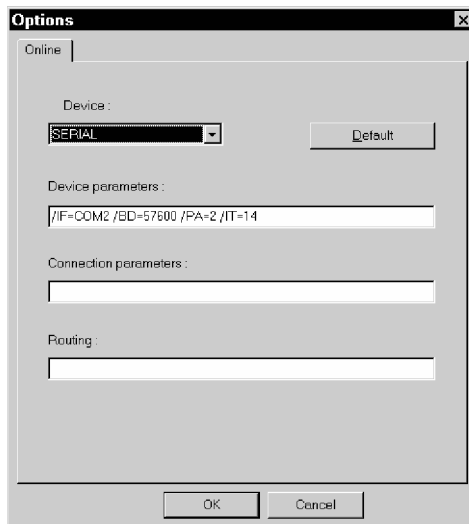
When using a LS251 Logic Scanner, the online connection is made via the PCI Bus and SRAM on the LS251.

### 3.3 Establishing the Online Connection

In our example, communication with the controller should take place via the COM2 serial interface on the programming device. Before the programming device and the controller are connected with each other, make sure that the controller is turned on (is power applied?). Then take the online cable and connect the RS232 interface on the CPU with the serial interface COM2 on the programming device:



After this is done, the online connection has to be configured. To make the configuration, call the **Options...** command from the **Tools** menu in B&R Automation Studio™. The online interface can be configured in the dialog box shown. The following parameters have to be set for communication via the COM2 serial interface:



The following section contains a short description of the parameters:

### Device

Interface used to communicate with the controller:

**SERIAL** ..... Serial interface  
**SHARED** ..... LS251 (online connection via PCI bus and SRAM)  
**CAN** ..... CAN interface

For communication via the COM2 serial interface, the **SERIAL** setting has to be used.

### Device Parameters

Configuration of the interface using a parameter line.

The following entries have to be made for communication via COM2:

**/IF=COM2** ..... Interface (COM1, COM2, COM3, COM4).  
 Only RS232 interfaces are supported!  
**/BD=57600** ..... Baudrate (9600, 19200, 38400, 57600, 115200 Bit/sec)  
**/PA=2** ..... Parity:  
 0 = no parity  
 1 = odd parity  
 2 = even parity  
 3 = parity always 1  
 4 = parity always 0  
**/IT=14** ..... Interval Timeout (in milliseconds).  
 Maximum time that can pass between receiving two consecutive characters. This time is used for communication between the programming device and the controller to differentiate between individual frames (data packages).

### Connection Parameters

The field **Connection parameters** has no meaning for the online connection via a serial interface. In our example, this field should remain empty !

### Routing

The field **Routing** has no meaning for the online connection via a serial interface. In our example, this field should remain empty !

The defined parameters are accepted when this dialog box is closed by pressing the **OK** button.

If the online interface is configured correctly, the online connection between the programming device and the controller will be established **automatically** when the project is opened (see Section 3.4 "Creating a New Project").

### 3.4 Creating a New Project

B&R Automation Studio™ is a hardware oriented programming system. That means the hardware has to be defined when creating a project. We will differentiate between **Automatic Hardware Recognition** and **Manual Hardware Definition**.

#### Automatic Hardware Recognition

When creating a project, B&R Automation Studio™ can carry out automatic hardware recognition. For inexperienced users, this is the simplest type of hardware definition.

To create a new project, select the **New Project...** command from the **File** menu. Then the **New Project Wizard** is opened. In this dialog box, you can enter the desired project name (**Name**) and the path to the project directory (**Path**). For our example project, we will use the project name "AS\_QS" and the project should be saved in the "C:\PROJECTS" directory:



If you want to use a project name other than "AS\_QS", read the limitations for project names given in context sensitive help !



In order to be able to upload the hardware configuration from the controller automatically (automatic hardware recognition), select **Upload hardware from target** !

If the directory "C:\PROJECTS" doesn't exist, you will be asked if it should be created when you press the **Next >** button. Answer this question with **OK**.

B&R Automation Studio™ then carries out automatic hardware recognition. The progress is shown in a dialog box. When the hardware recognition is complete, you can open the next dialog box by clicking on **Next >**:



This dialog box shows all previously defined data along with the modules found during hardware recognition (model number of CPU and PS).

Check the data again. You can go back and correct data by clicking on **< Back**. If the data is correct, click on **Finish** to create and open the project.

## Manual Hardware Definition

Manual hardware definition is mostly used when the target system is not available. The following section describes the procedure for project creation using manual hardware definition.

To create a new project, select the **New Project...** command from the **File** menu. Then the **New Project Wizard** is opened. In this dialog box, you can enter the desired project name (**Name**) and the path to the project directory (**Path**). For our example project, we will use the project name "AS\_QS" and the project should be saved in the "C:\PROJECTS" directory:



If you want to use a project name other than "AS\_QS", read the limitations for project names given in context sensitive help !

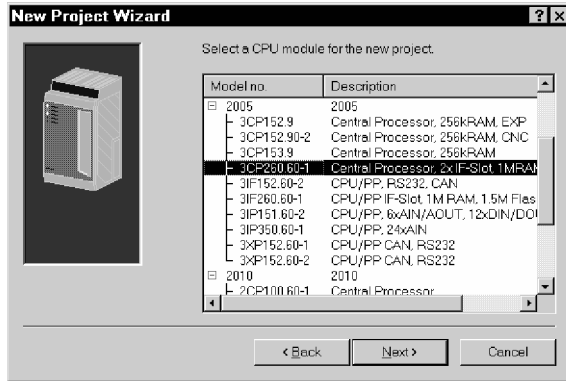


For manual hardware definition, **Upload hardware from target** has to be deactivated !

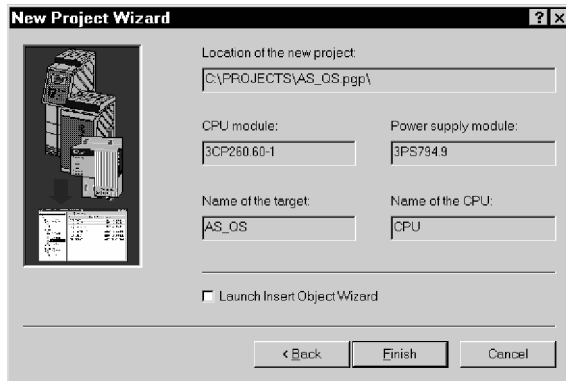
If the directory "C:\PROJECTS" doesn't exist, you will be asked if it should be created when you press the **Next >** button. Answer this question with **OK**.



Then you have to define the CPU on the target system. A list of all possible modules is shown:



Select the desired CPU from the list and then click on **Next >**. Another dialog box is then opened where you can select from a list of all power supply modules that can be used with for the selected CPU. Select the desired power supply and then click on **Next >**. The next dialog box shows all previously defined data and the modules defined manually (model number of the CPU and power supply modules):



Check the data again. You can go back and correct data by clicking on **< Back**. If the data is correct, click on **Finish** to create and open the project.



During manual hardware definition, only the CPU and power supply modules are defined. All other modules (e.g. digital or analog I/O modules) have to be set after manual hardware definition. A detailed description of this procedure can be found in Appendix A "Frequently Asked Questions", Section 2 "How can I program without hardware?".

### 3.5 Connection Status

If the online interface is configured correctly, the **online connection** between the programming device and the target system is established **automatically** when a project is opened. The connection status is shown in the status bar:

#### Active Connection

**Active connection** between programming device and target system:



The text in the status bar contains the following information:

- COM2** ..... Shows the interfaced used (online interface on the computer).
- CP260 V2.10** ..... Shows the processor type (name of the CPU) and the operating system version used.
- RUN** ..... Mode that the processor is in:
  - RUN = RUN Mode
  - SERV = SERVICE Mode
  - DIAG = DIAGNOSE Mode

#### Offline

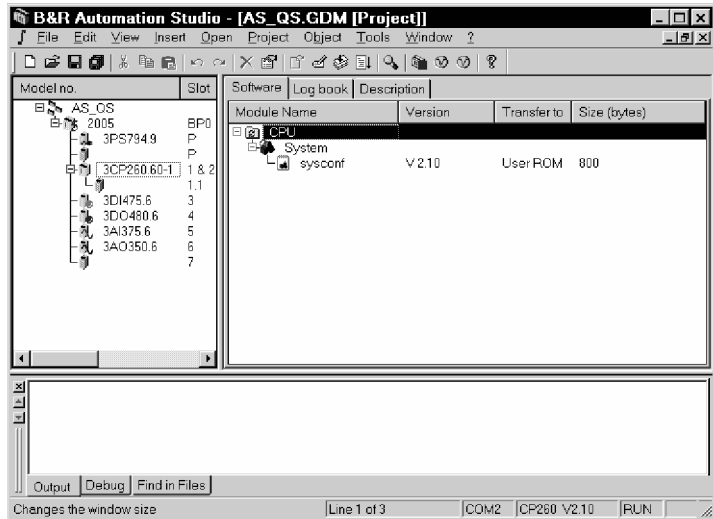
**No connection** between programming device and target system:



If a connection cannot be established even though the interface configuration is correct, check again to see if the PCCSW operating system is installed on the controller ! Have a look at Section 3.1 "Preparation".

### 3.6 Work Environment

You probably noticed that the B&R Automation Studio™ main menu changes when you open a project. A "Project Window" is also opened. The project window is the heart of every project. It is shown as a two part window:



The **left** section of the window gives an overview of the hardware used in the project (hardware configuration).

The **right** section of the window provides additional information and settings for the marked module.

#### Left Section of Project Window

The left section of the window gives an overview of the hardware used in the project .





The left section of the project window is also called the **hardware configuration**.

The easiest way to define the hardware in a project is using automatic hardware recognition (see Section 3.4 "Creating a New Project"):

- When creating a new project, the user can let B&R Automation Studio™ upload the available hardware from the controller.
- If there is an active connection between the programming device and the target system, the current hardware on the target system is compared with the hardware configuration (shown in the project window).

If there are hardware differences, the modules in the current project that are different from the modules on the controller are marked:

 ..... A **different module** is in this slot on the controller ! You can replace the module by marking it and calling the context menu (right mouse button).

 ..... **No module** is in this slot on the controller !

### Right Section of Project Window

The right section of the window contains additional information and settings for the module marked in the left section. The right section is divided into registers in order to show the information more clearly. Which register is shown depends on which module you have marked in the left section.

#### Example CP260

**Software** ..... This register card allows access to the software configuration (see below).

**Log book** ..... This register card shows entries from the logbook on the PCC.

**Description** ..... Display of hardware and software specific information for the module marked. Displayed in HTML format.



The **software configuration** is integrated into the right section of the project window. Each module that uses software (CPU, multiprocessor or parallel processor) has a **Software** register card. This register card is also called the software configuration.

You can configure the PCCSW operating system performance to suit your application using the software configuration (you can find an in-depth description of the software configuration in the online help).

#### Example DI475

**I/O** ..... Module configuration: Displays all data points on the module. This list varies according to the properties of a module.

**Description** ..... Display of hardware and software specific information for the module marked. Displayed in HTML format.

## 4. Variable Declaration

We need the following **process variables** for our short example program (abbr. **PVs**):

key\_1 ..... digital input  
relay\_1 ..... digital output

There are several ways to declare these two process variables. The variable declaration method shown on the following pages is the easiest for beginners.

### 4.1 Digital Input

The digital input "key\_1" can be defined as follows:

- In the left section of the project window, click on the symbol for the DI475 module (digital input module). The registers **I/O** and **Description** are then shown in the right section of the window.
- If you click on the **I/O** register card, a list of all data points on the DI475 module is shown.
- To assign the variable name "key\_1" to the first digital input on the DI475 module ...
  - ... double click on the **PV Name** column in the first line of the list and enter the variable name "key\_1". The entry for the variable name is completed when the **ENTER** key is pressed.
  - ... place the cursor on the **PV Name** column in the first line of the list and press the space bar. Then you can enter the variable name "key\_1". The entry is completed when the **ENTER** key is pressed.

I/O		Description	
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

Variable names can be up to 32 characters long.

- The **Remark** column normally contains a short function description for I/O on the module marked.

You can enter an optional description with up to 60 characters in this column. To do this, double click on the **Remark** column and enter the desired text. The entry is completed when the `ENTER` key is pressed.

The additional description is only for documentation and is not transferred to the controller.

## 4.2 Digital Output

The digital output "relay\_1" can be defined as follows:

- In the left section of the project window, click on the symbol for the DO480 module (digital output module). The registers **I/O** and **Description** are then shown in the right section of the window.
- If you click on the **I/O** register card, a list of all data points on the DO480 module is shown.
- To assign the variable name "relay\_1" to the first digital output on the DO480 module ...
  - ... double click on the **PV Name** column in the first line of the list and enter the variable name "relay\_1". The entry for the variable name is completed when the `ENTER` key is pressed.
  - ... place the cursor on the **PV Name** column in the first line of the list and press the space bar. Then you can enter the variable name "relay\_1". The entry is completed when the `ENTER` key is pressed.

I/O	Description		
Name	Data Type	PV Name	Remark
digital output 01	BOOL	relay_1	Transistor, 2A/24VDC
digital output 02	BOOL		Transistor, 2A/24VDC
digital output 03	BOOL		Transistor, 2A/24VDC
digital output 04	BOOL		Transistor, 2A/24VDC
digital output 05	BOOL		Transistor, 2A/24VDC
digital output 06	BOOL		Transistor, 2A/24VDC
digital output 07	BOOL		Transistor, 2A/24VDC
digital output 08	BOOL		Transistor, 2A/24VDC
digital output 09	BOOL		Transistor, 2A/24VDC
digital output 10	BOOL		Transistor, 2A/24VDC
digital output 11	BOOL		Transistor, 2A/24VDC
digital output 12	BOOL		Transistor, 2A/24VDC
digital output 13	BOOL		Transistor, 2A/24VDC

Variable names can be up to 32 characters long.

- The **Remark** column normally contains a short function description for I/O on the module marked.

You can enter an optional description with up to 60 characters in this column. To do this, double click on the **Remark** column and enter the desired text. The entry is completed when the `ENTER` key is pressed.

The additional description is only for documentation and is not transferred to the controller.

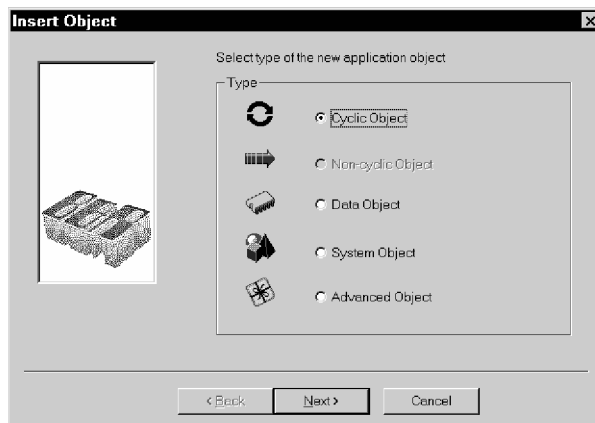
## 5. Creating a Cyclic Object

Now we will enter our example program using the ladder diagram (LAD) programming language. We will create a cyclic object for the example program because it is to be processed cyclically.

### 5.1 Inserting a Cyclic Object

A cyclic object is inserted in the following manner:

- In the left section of the project window, select a module that can contain a software object (CPU, multiprocessor or parallel processor). In our example, this is the CP260 module.
- Select the **Software** register in the right section of the window.
- Insert a new object by...
  - ... clicking on **New Object...** in the **Insert** menu, or
  - ... click on the CPU symbol with the right mouse button and select the **Insert Object...** command from the context menu shown, or
  - ... click on the CPU symbol and then press the **Ins** key. Then you can select **Insert Object**.
- In the dialog box shown, select resource type **Cyclic Object** and acknowledge the selection by pressing the **Next >** button.





## Resource Types

The **Insert Object** dialog box allows the following resource types to be selected:

### Cyclic Object

A **cyclic object** is processed at regular intervals using a defined cycle time. The operating system monitors the cycle times to make sure they are not exceeded.

Priorities can be set by assigning various resources to the cyclic objects (different cycle times). Cyclic objects that carry out important, time critical tasks are assigned a resource with a shorter cycle time and are therefore carried out more often than objects that are assigned a resource with a longer cycle time. Cyclic objects include:

- Cyclic objects
- Timer objects
- Interrupt objects
- Exception objects

### Non-Cyclic Object

**Non-cyclic objects** (idle time objects) run without cycle time monitoring. They are processed during system idle time. That means they are only carried if both the PCCSW operating system and cyclic objects do not require system time (CPU time).

### Data Object

**Data objects** are modules which are used to store data.

### System Object

B&R modules (\*.BR files) can be transferred to the controller as **system objects**. System objects include libraries or objects with additional system functions. Many of these objects are loaded automatically by B&R Automation Studio™ and don't have to be inserted manually.



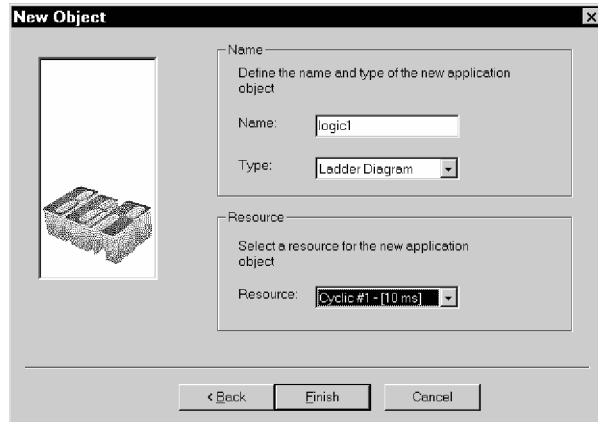
A **library** is a collection of standard functions that is represented in B&R Automation Studio™ as a B&R module. Each library contains functions that can save you time and effort when creating solutions to standard problems. If a function from a library is used, this library is automatically imported into the project and installed on the controller during a download.

### Advanced Object

NC axes, CAM profiles, etc.

## 5.2 Define Object Name, Object Type and Resource

After defining the resource type in the **Insert Object** dialog box, the **New Object** dialog box use to define the object name, object type and resource is opened:



Enter the desired name in the **Name** field ("logic1" in our example) for the new object. Take note of the limitations for object names described in the context sensitive help ! Then select **Ladder Diagram** for the object type (**Type** field) and **Cyclic #1 - [10 msec]** for the resource (**Resource** field).



Priorities for the individual cyclic objects can be defined by assigning various **resources** to the objects. A different cycle time can be set for each cyclic resource and each timer resource. All objects which have been assigned a certain resource are processed once within the cycle time defined for that resource. Objects that carry out important, time critical tasks are assigned a resource with a shorter cycle time and are therefore carried out more often than objects that are assigned a resource with a longer cycle time.

After acknowledging these entries by clicking on the **Finish** button, the cyclic object is entered in the software configuration and the LAD editor is opened.

## Object Types

The following object types are presently possible for cyclic objects (see **Type** in the **New Object** dialog box):

- Ladder Diagram ..... IEC 1131 (abbr.: LAD)
- Instruction list ..... IEC 1131 (abbr.: IL)
- Structured text ..... (abbr.: ST)
- Sequential function chart ..... (abbr.: SFC)
- C Language ..... ANSI C

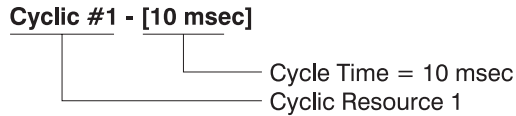
## Resources

The following resources are possible for cyclic objects (see **Resource** in the **New Object** dialog box):

### Cyclic #x - [...msec]

Objects that have this resource assigned to them are called cyclically. The objects are called by the PCCSW operating system. Cyclic resources have a fixed name (**Cyclic**) which the cycle time is added to (time for one cycle).

Example:



### Timer #x - [...msec]

Objects that have this resource assigned to them are called cyclically by a special hardware timer (interrupt). Timer resources have very consistent timing and a higher priority than cyclic resources. Timer resources have a fixed name (**Timer**) which the cycle time is added to (time for one cycle).

Example:



All resources described up to this point (cyclic resources and timer resources) are time controlled. However, there are also resources that are not time controlled, instead are carried out when a certain event occurs. We will differentiate between two types of event controlled resources: interrupt resource and exception resource.

## Interrupt

Interrupts are asynchronous events (triggered by hardware) that interrupt cyclic program execution.

Interrupts can only be generated by special hardware. An interrupt capable module is e.g. the DI400 digital input module (System B&R 2010). On this module, 8 digital inputs can be used as CSI inputs (CSI = "Change of State Interrupt"). If the state of an enabled CSI input changes, an interrupt is triggered on the CPU.

An object can be created for each interrupt capable module which is used to handle the event.

## Exception


Exceptions are signals that come from the PCCSW operating system and indicate a certain exception status. They can be compared to an interrupt triggered by the system that should indicate a fatal error. These errors can be reacted to using objects which have the exception resource assigned to them.

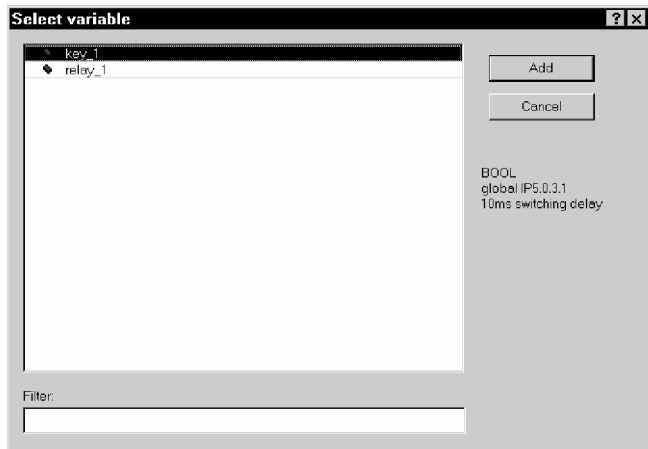
## 5.3 Entering the Ladder Diagram

Now we will program the circuit diagram (see Section 2.1 "Task Description") in the ladder diagram editor (abbr.: LAD Editor).

### Placing Digital Inputs





Digital Inputs can be placed in the LAD editor as follows:

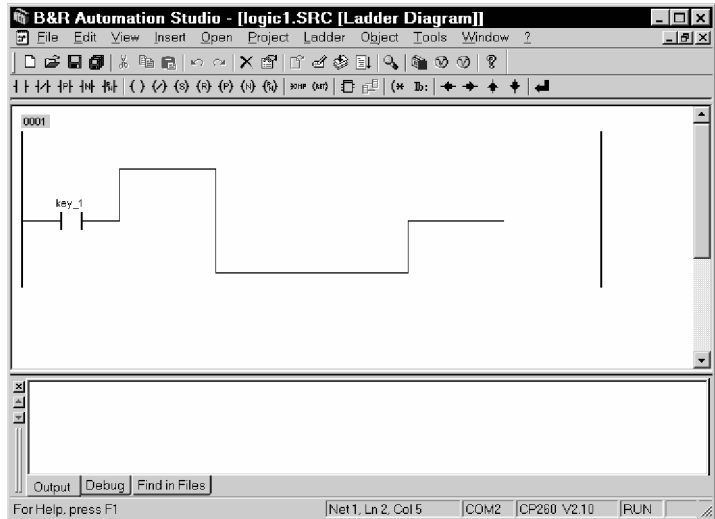
- Add a digital input symbol by ...
  - ... clicking on the  button in the toolbar, or
  - ... pressing the `c` key, or
  - ... clicking on the **Contact** command in the **Insert** menu and selecting **Normally Open Contact** from the submenu shown.
- Then assign the digital input "key\_1" to the contact by ...
  - ... entering the variable name "key\_1" and pressing the ENTER key. or
  - ... pressing the space bar. Then a list of all defined process variables will be shown. Select "key\_1" from this list and click on the **Add** button.



The list only contains variables that are compatible with the data type of the new contact !


## Drawing Connection Lines

Connection Lines can be drawn using the key combination ALT + CURSOR or with the , ,  and  buttons.




## Placing Digital Outputs

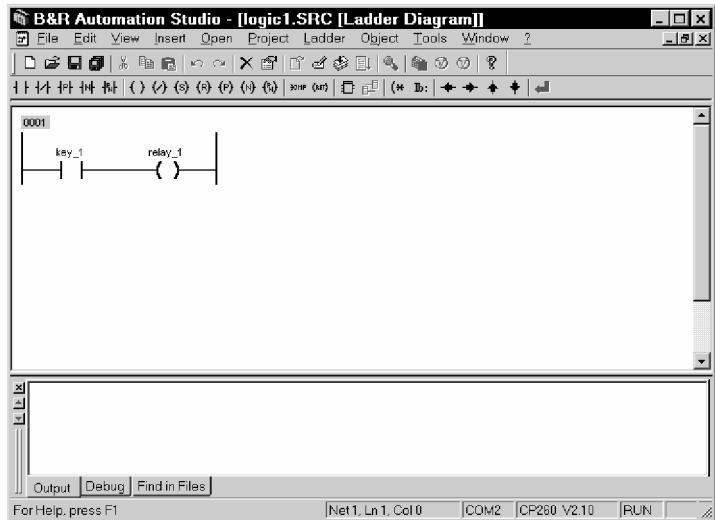
Digital outputs can be placed in the LAD editor as follows:

- Add a digital output symbol by ...
  - ... clicking on the  button in the toolbar, or
  - ... pressing the key combination SHIFT + C, or
  - ... clicking on the **Coil** command in the **Insert** menu and selecting **Coil** from the submenu shown.
  
- Then assign the digital output "relay\_1" to the coil by ...
  - ... entering the variable name "relay\_1" and pressing the ENTER key, or
  - ... pressing the space bar. Then a list of all defined process variables will be shown. Select "relay\_1" from this list and click on the **Add** button.

The list only contains variables that are compatible with the data type of the new coil !

## Checking and Optimizing LAD

If you click on the  button in the toolbar or press the ENTER key, the ladder diagram is checked for open connection lines or other errors and optimized (extra connection lines are deleted, etc.). Check and optimize ladder diagram:



## View Information On/Off


In the ladder diagram, it is possible to show additional information. The commands **Type**, **Scope** and **Remark** from the **View** menu and the context menu of the LAD editor (called with the right mouse button in the LAD editor) can be used to select which information is to be shown. A check ✓ to the left of the menu command means that this information will be shown.

Meaning of the additional information:

- Type** ..... Data type of process variables (detailed information concerning "data type" can be found in the online help).
- Scope** ..... Valid range of process variables (detailed information concerning "scope" can be found in the online help).
- Remark** ..... Up to 60 character long description of the process variable (see Section 4. "Variable Declaration").


## Saving and Closing LAD

To save the ladder diagram "logic1" on the hard drive of the programming device (under C:\PROJECTS\AS\_QS\...) ...

- ... click on the  button in the toolbar, or
- ... press the key combination `CTRL + S`, or
- ... select the **Save** command from the **File** menu.

When saving, the ladder diagram is checked and optimized again. After saving, you can close the LAD editor.

Close the LAD editor by ...

- ... clicking on the **Close** command in the **File** menu, or
- ... clicking on the  in the upper right corner of the LAD editor window.



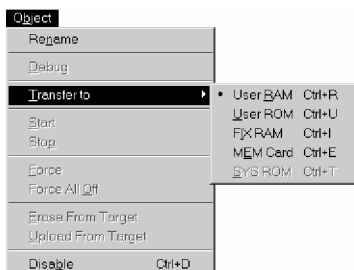
## 6. Compiling and Transferring Objects to the Controller

The ladder diagram has to be converted to machine language for the processor on the controller. B&R Automation Studio™ has a compiler for this purpose. The compiler provides an executable program module (B&R module) in machine language for the processor.

Before compiling the example program "logic1" and transferring it to the controller, you have to define the memory area that will be used as target for object transfer.

### Defining the Memory Area

To define the target memory area for the object "logic1", click on the object symbol in the software configuration and select the **Transfer to** command from the **Object** menu. A list of all memory areas possible for this object is shown:



Select the memory area **User RAM** from this list for the example program. Then the object will be stored in User RAM on the controller during transfer.

Overview of possible memory areas:

#### User RAM

**User RAM** is a conditionally nonvolatile memory area (buffered when power is not applied to the controller) and is available for application software (cyclic objects, idle time objects, data objects, etc.).

#### User ROM

**User ROM** is a nonvolatile memory area that can be used to store cyclic objects, idle time objects, data objects, etc. User ROM is in the form of a FlashPROM.

## FIX RAM

Using B&R Automation Studio™ (software configuration), you can configure a part of User RAM so it behaves like User ROM during a COLDSTART. This memory area is called **FIX RAM**. Objects stored in FIX RAM are not deleted during a COLDSTART.



The boot mode **COLDSTART** corresponds to the first initialization of the processor. During a COLDSTART, the operating system is completely restarted and all RAM data is deleted (exception: objects stored in FIX RAM).

## MEM Card

Some processor modules can be equipped with a **PCMCIA Memory Card** (also called **MEM Card**). A MEM Card is a nonvolatile memory area where you can store cyclic objects, idle time objects, data objects, etc. That means this memory area corresponds to either User ROM (MEM Card as FlashPROM) or FIX RAM (MEM Card as SRAM).

## SYS ROM

The PCCSW operating system is stored in **System ROM**. System ROM is a nonvolatile memory area in the form of a FlashPROM.

## Compile and Transfer

To compile the example program and transfer it to User RAM, select the **Transfer To Target** command from the **Project** menu. Using this command, the entire project is compiled and then the objects are transferred to the defined memory area and started.

The progress and completion status of the compile procedure is shown in the message window:

```
* Compiling logic1 ...
* Compiling logic1 ok
0 Error(s) - 0 Warning(s)
* Transferring logic1 (User RAM, Vers: 0.00, 05.08.1999, 316 Byte, Pe
* Transferring logic1 ok
```

Output | Debug | Find in Files

To avoid errors, B&R Automation Studio™ checks the structure of the target system hardware, the version of the operating system and all previously installed software objects during transfer. Known problems will be clearly pointed out so that additional errors can be avoided.

The following message is given after the project has been successfully transferred:



The example program is now running in User RAM. The digital output is set or reset according to the logical state of the digital input.


## 7. Testing the Object

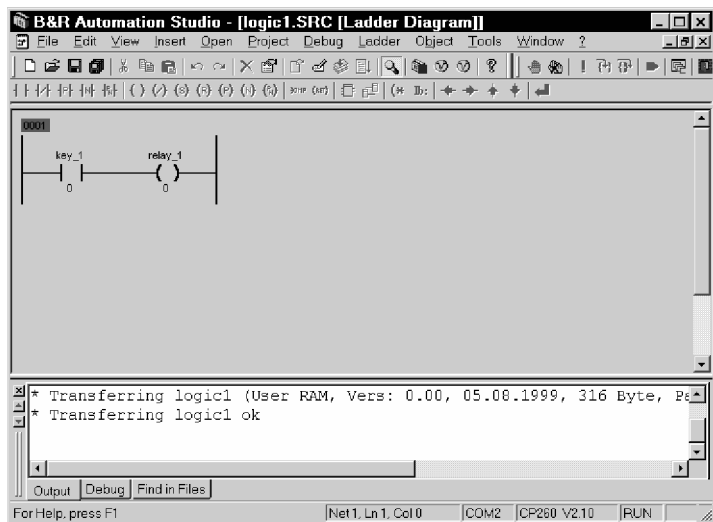
B&R Automation Studio™ recognizes many problems during programming, however it is very unlikely that large applications function perfectly after the first transfer. For this reason, B&R Automation Studio™ provides a number of powerful diagnostics tools that will make it easier to find errors.

### 7.1 Ladder Diagram Monitor

With the *ladder diagram monitor (LAD monitor)*, it is possible show and force process variables from a ladder diagram running on the controller directly in the ladder diagram mask.

To call the LAD monitor, you have to open the LAD editor by double clicking on the symbol for "logic1" in the software configuration. You can then open the LAD monitor from the LAD editor by ...

- ... clicking on the  button on the toolbar, or
- ... entering the key combination CTRL + M, or
- ... selecting the **Monitor** command in the **View** menu.



## Forcing a Variable

Using the Force function in the LAD monitor, you can change the state of digital inputs from B&R Automation Studio™. This function can be used to test the example program "logic1".



If a process variable is **forced**, it is set to a certain value and remains at this value. If an object writes to the process variable, the changed value only remains valid for the current cycle. Then the variable is overwritten by the force value.

The **Force function** is also available in other diagnostics tools in B&R Automation Studio™.

To force the variable "key\_1", click on the symbol for the digital input "key\_1" with the right mouse button and select **Force** from the context menu shown. Then a dialog box is shown asking if the variable should be forced. Answer this question with **OK**.

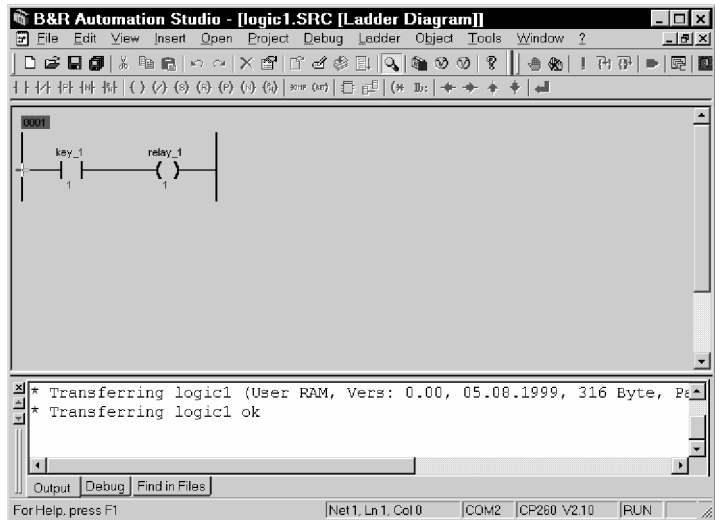


A check ✓ to the left of the **Force** menu item in the context menu means that the variable is forced ! The values of forced variables are also shown in **red** in the LAD monitor.

Then you can set the value of the digital input "key\_1" to 1, by ...

- ... clicking on the symbol for the digital input with the right mouse button again and selecting **Value** from the context menu. The value "1" can now be entered. Press **ENTER** to complete the entry.
- ... click on the symbol for the digital input and press the space bar. The value "1" can now be entered. Press **ENTER** to complete the entry.

This causes the digital output "relay\_1" to be set:



## Closing the LAD Monitor

Close the LAD monitor by ...

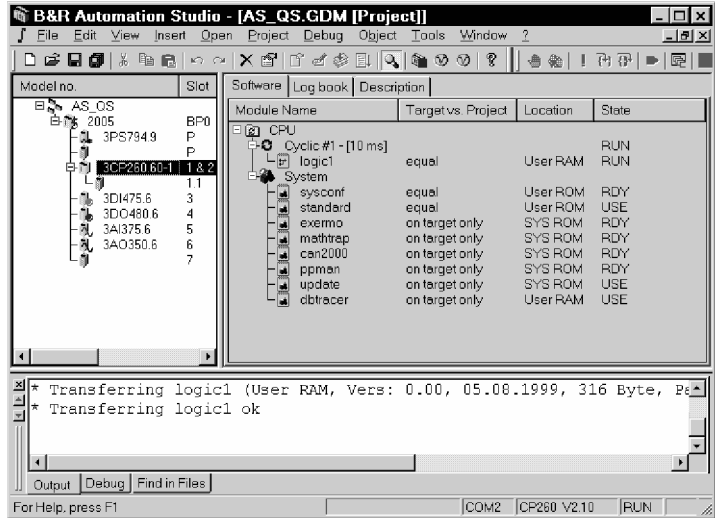
... clicking on **Close** in the **File** menu, or

... clicking on  in the upper right corner of the LAD monitor window.


Additional information concerning "LAD Monitor" can be found in the B&R Automation Studio™ online help.

## 7.2 System Monitor

When you close the LAD monitor window, B&R Automation Studio™ remains in monitor mode and the **system monitor** (system monitor = software configuration in monitor mode) is shown:



To turn monitor mode on or off, ...

- ... click on the  button in the toolbar, or
- ... enter the key combination CTRL + M, or
- ... select the **Monitor** command from the **View** menu.

Different information is shown in the software configuration depending on if B&R Automation Studio™ is in monitor mode or not:

### Monitor Mode Turned Off

In this mode, all objects in the current project are shown. Objects only found on the controller are not shown in this mode. The following information is shown in the right section of the project window:

Module Name	Version	Transfer to	Size (bytes)
CPU			
Cyclic #1 - [10 ms]			
logic1	v 0.00	User RAM	316
System			
sysconf	v 2.10	User ROM	800
standard	v 1.22	User ROM	28872

### Module Name

Symbol and name of the software object.

### Version

Version number of the software object.

### Transfer to

Target memory for the software object (valid for the next transfer).

### Size (bytes)

Size of the compiled software object in bytes. If the object has not been compiled, the value 0 is shown.

## Monitor Mode Turned On

In this view, the differences between the current project (computer) and the controller are shown. The following information is provided in addition to the symbol and name of an object:

Module Name	Target vs. Project	Location	State
CPU			
Cyclic #1 - [10 ms]			RUN
logic1	equal	User RAM	RUN
System			
sysconf	equal	User ROM	RDY
standard	equal	User ROM	USE
exerms	on target only	SYS ROM	RDY
mathtrap	on target only	SYS ROM	RDY
can2000	on target only	SYS ROM	RDY
ppman	on target only	SYS ROM	RDY
update	on target only	SYS ROM	USE
dbtracer	on target only	User RAM	USE

### Module Name

Symbol and name of the software object.

### Target vs. Project

Objects in the project and on the controller are compared and the result is shown:

on target only ..... Object is only on the controller.

not on target ..... Object is not on the controller (only in project).

older on target ..... Object in the project is more current than the one on the controller.

newer on target .... Object on the controller is more current than the one in the project.

equal ..... Object in the project and on the controller are identical.

### Location

Memory where the object can be found on the controller.



**State**

Status of the object.

The status of cyclic objects (cyclic objects, timer objects, interrupt and exception objects) and idle time objects:

RUN ..... Software object being executed.

STOP ..... Software object stopped (NOT being executed).

The status of the system and data objects:

RDY ..... The software object (system or data object) is ready to be used by other objects.

USE ..... The software object (system or data object) is being used by other objects.

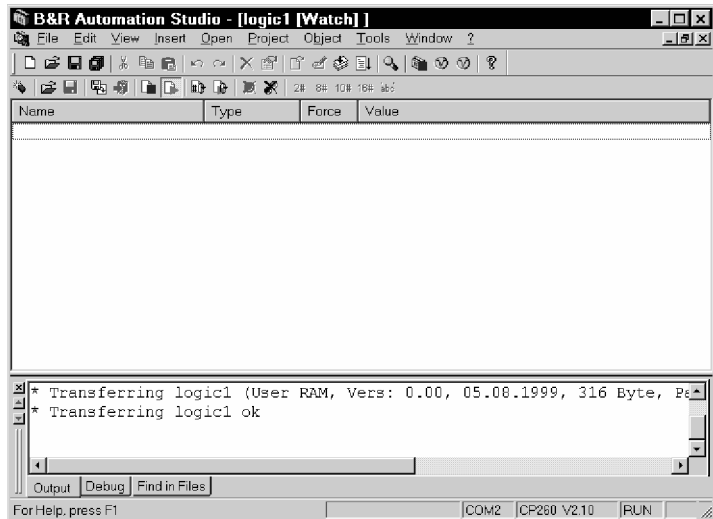
Additional information concerning the "System Monitor" can be found in the B&R Automation Studio™ online help.

### 7.3 Watch

Using the **Watch function**, you can view, monitor and change the values of process variables found on the controller.  
Open the watch window for the object "logic1" by ...

- ... clicking on the symbol "logic1" in the software configuration and selecting the **Watch** command in the **Open** menu, or
- ... clicking on the symbol "logic1" with the right mouse button in the software configuration and selecting the **Watch** command from the context menu shown.

The following information is shown in the watch window:



#### Name

Names of process variables are shown in this column.

#### Type





The data type of each variable is shown in this column.



The memory requirements of a process variable and its value range are defined using **data type** (detailed information concerning this topic can be found in the online help).

## Force

The following information is shown in this column:

Symbol	Description	Force Status
	Force symbol for input	not active
	Force symbol for input	active
	Force symbol for output	not active
	Force symbol for output	active
no symbol	Variable in memory	---

## Value

The value of variables is shown in this column. You can select from several view formats:


- binary
- octal
- decimal
- hexadecimal
- string

**Binary, Octal, Decimal, Hexadecimal** and **String** in the **View** menu and in the context menu for the watch window (called with the right mouse button in the watch window) can be used to select the view format. A dot ● to the left of the menu item means that the variable values will be displayed in this format.

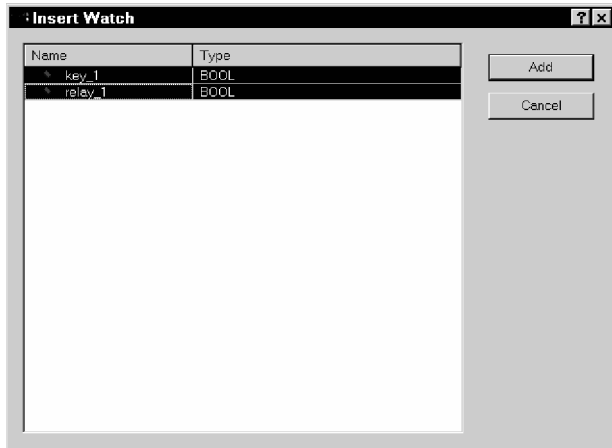
## Inserting a Variable

All functions in the watch window refer to the software object that was marked in the software configuration when the watch function was called. That means you can only insert variables that exist in the selected object.

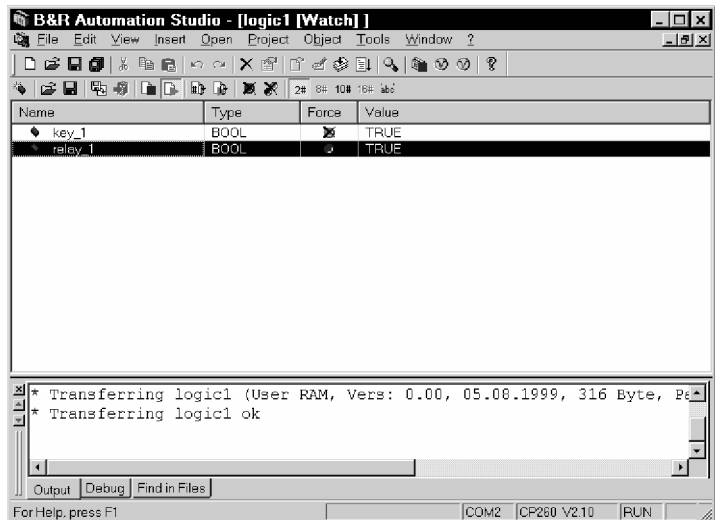
You can insert a variable in the watch window as follows:

- The watch window has to be the active window !
- Insert a new variable by ...
  - ... clicking on  in the toolbar, or
  - ... pressing `INS` and then the space bar, or
  - ... clicking on **Variable** in the **Insert** menu, or
  - ... clicking in the watch window with the right mouse button and select the **Insert Variable** command in the context menu shown.
- Select the desired variable (e.g. "key\_1") in the **Insert Watch** dialog box and click then press the **Add** button.

You can also select several variable at the same time in the **Insert Watch** dialog box and insert them into the watch window. To do this, press and hold **SHIFT** when selecting the variables (with the mouse or **CURSOR** key):



After closing the dialog box by pressing the **Add** button, all marked variables are inserted into the watch window:



## Editing a Variable

You can change the value of a variable as follows:

- Mark a variable in the watch window.
- Change the value of this variable by ...
  - ... pressing the space bar (the cursor has to be in the **Value** column) and entering the desired value, or
  - ... clicking on **Variable** in the **Edit** menu and entering the desired value, or
  - ... entering the desired value directly (the cursor has to be in the **Value** column).
- Press **ENTER** to accept the new value or **ESC** to disregard the change.



If the variable to be changed is an input or output as with our example (digital input "key\_1" or digital output "relay\_1"), the variable is **automatically** forced after acknowledgement.

## Activate/Deactivate Force Status

The force status of a variable can be activated **manually**. To force a variable, mark it in the watch window and click on **Force** in the **Object** menu or context menu (called with the right mouse button). Then a dialog box is shown asking if the force status of the variable should be changed. Answer this question with **OK**.



A check  to the left of the **Force** menu item from the **Object** menu and context menu means that the variable is forced ! The following symbols in the **Force** column of the watch window also indicate a forced variable:

Symbol	Description	Force Status
	Force symbol for input	active
	Force symbol for output	active

To deactivate the force status of a variable, mark it in the watch window and click on **Force** again in the **Object** menu or context menu (called with the right mouse button). Then a dialog box is shown asking if the force status of the variable should be changed. Answer this question with **OK**.

If you want to deactivate the force status of all variables at the same time, click on the button in the toolbar or on **Force All Off** in the **Object** menu.

## Closing the Watch Window

Close the watch window by ...

... clicking on **Close** in the **File** menu, or

... clicking on  in the upper right corner of the watch window.

Additional information concerning all functions of the watch window can be found in the B&R Automation Studio™ online help.

# APPENDIX A FREQUENTLY ASKED QUESTIONS

# 1. How do I install the PCCSW operating system?

The PCCSW operating system is stored in System ROM on the processor (CPU, multiprocessor or parallel processor).



System ROM is a nonvolatile memory area in the form of a FlashPROM.

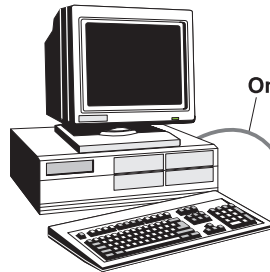
You can install the PCCSW operating system using B&R Automation Studio™ (operating system download).

## 1.1 Processors with Application Memory

Application memory is always delivered without an operating system installed. The first operating system installation (operating system download) can be carried out as follows for processors with application memory (CP10x, CP2x0, MP100 and CP15x):

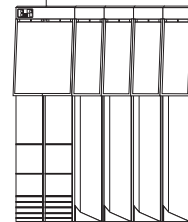
1. Turn off power to the controller. This is necessary because the application memory is only allowed to be removed or inserted without power applied !
2. Insert new FlashPROM application memory and return power. Make sure that the write protect switch on the front of the application memory is on the write position !
3. Take the online cable and connect the computer to the processor (CPU or multiprocessor) where the operating system should be installed.

Programming Device



Online Cable

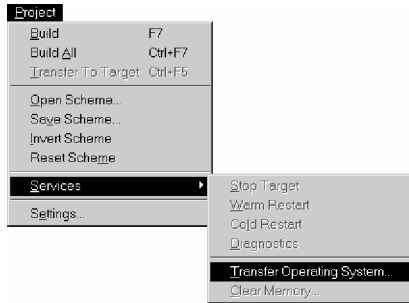
Processor with Application Memory (e.g. CP100)



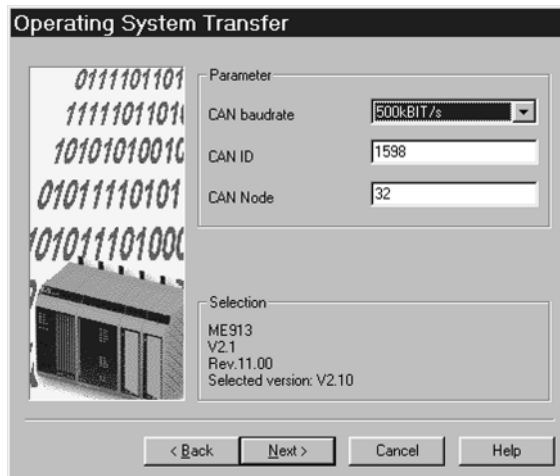
4. Start B&R Automation Studio™.



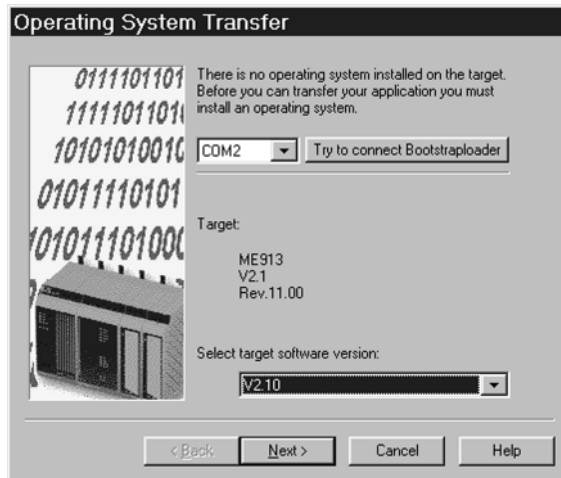
- Open a project in B&R Automation Studio™ by calling the **Open Project...** command from the **File** menu.
- Start the download procedure by calling the **Services** command from the **Project** menu. Select **Transfer Operating System...** from the menu shown.



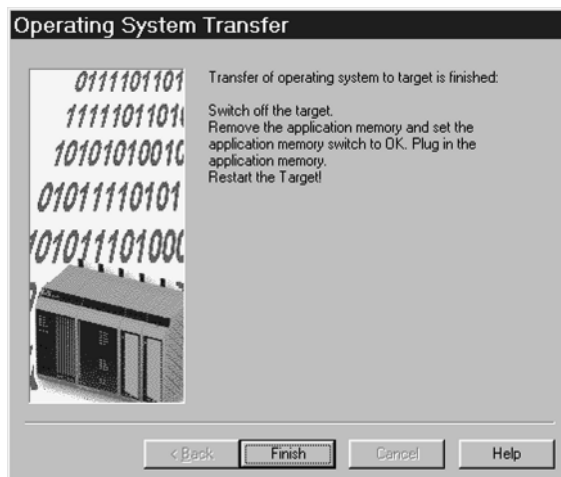
- B&R Automation Studio™ now gathers all information required for the operating system download (type and revision of the application memory, etc.) from the controller. Then a dialog box is shown where you can select the interface used on your computer for the connection to the target system (e.g. COM2) and the desired operating system version (e.g. V2.10).



- When you press **Next >** , a dialog box will be opened where you can define a few CAN parameters (baudrate, CAN identifier and CAN node number):



- The default settings in this dialog box only has to changed if your controller is in a CAN network that requires other settings !
- After closing this dialog box by pressing **Next >** , the FlashPROM is deleted. Then the selected version of the operating system is downloaded to FlashPROM. Download progress is shown on the screen.  
The download procedure can take a few minutes !
- The following dialog box is shown when the download procedure is finished:



12. After closing this dialog box by pressing **Finish**, you have to turn the controller off and on again.

13. The controller is now ready for operation.



If the download procedure is interrupted, the operating system installation has to be started again !

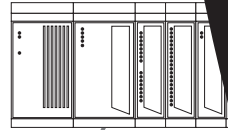
**Processors**  
**Application**  
**Memory**

An operating system is not installed on memory (CP 260, XP152, IF152) at delivery. The first operating system load can be carried out as follows:

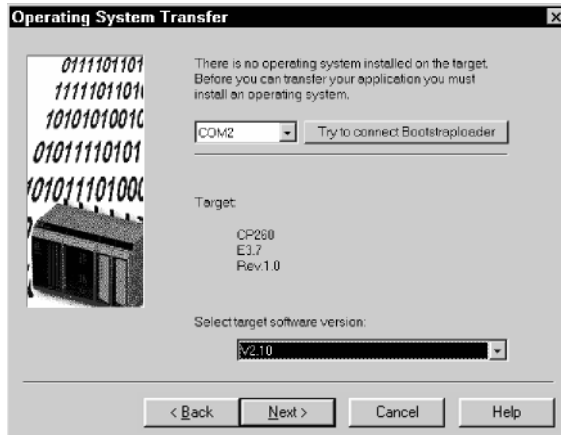
1. Take the online cable and connect it to the computer (CPU or parallel processor) which is installed.

**Programming Device**

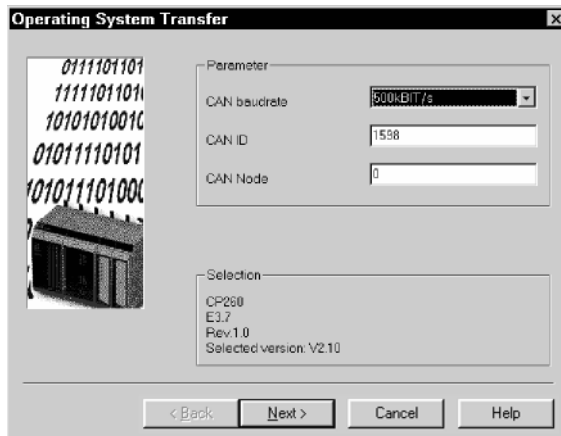
Online C



5. B&R Automation Studio™ now gathers all information required for the operating system download (type and revision of the application memory, etc.) from the controller. Then a dialog box is shown where you can select the interface used on your computer for the connection to the target system (e.g. COM2) and the desired operating system version (e.g. V2.10).

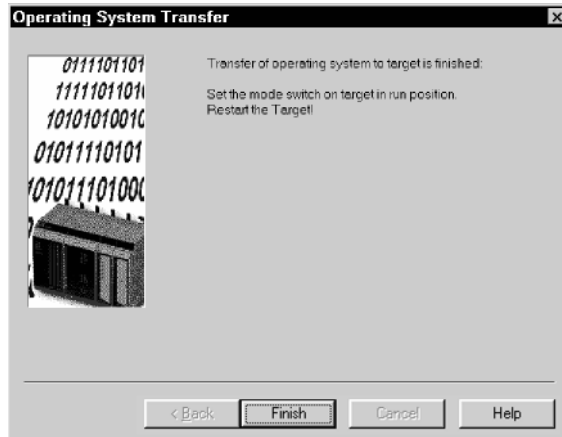


6. When you press **Next >**, a dialog box will be opened where you can define a few CAN parameters (baudrate, CAN identifier and CAN node number):



7. The default settings in this dialog box only has to changed if your controller is in a CAN network that requires other settings !

8. After closing this dialog box by pressing **Next >**, the FlashPROM is deleted. Then the selected version of the operating system is downloaded to FlashPROM. Download progress is shown on the screen.  
The download procedure can take a few minutes !
9. The following dialog box is shown when the download procedure is finished:



10. After closing this dialog box by pressing **Finish**, you have to ...
  - ... set a number other than 00 and FF on the CAN node number switches (valid e.g. for modules XP152, IF152 and IP151).
  - ... set a number other than 00 and FF on the operating mode switches (valid for module IP350).
  - ... set a number other than 0 and F on the operating mode switch (valid for module CP260).
  - ... set the operating mode switch to 1 - "Run" mode (valid for module IF260).
  - ... set a number between 1 and 8 on the MODE switch (valid e.g. for modules CP47x and CP77x).

See hardware documentation for details.

11. Turn controller off and then on again.
12. The controller is now ready for operation.



If the download procedure is interrupted, the operating system installation has to be started again !

## 2. How can I work without hardware?

B&R Automation Studio™ is a hardware oriented programming system. That means the hardware has to be defined when creating a project. **Automatic Hardware Recognition** and **Manual Hardware Definition** can be carried out.

If you want to work without hardware, you have to use manual hardware definition when creating a project.

### Manual Hardware Definition

A detailed description of the steps required for manual hardware definition can be found in Chapter 2 "The First Steps", Section 3.4 "Creating a New Project".

If the target system hardware is not available, the programming device can work OFFLINE (no connection between computer and controller).

### Define Additional Modules

After creating a project, additional modules can be defined in the left section of the project window (e.g. digital or analog I/O modules). This is done as follows:

- Place the cursor in the hardware configuration on the position where you want to insert the module.
- Insert a new module by ...
  - ... clicking on **Module...** in the **Insert** menu, or
  - ... pressing the **INS** key.
- Select the desired module in the **Select Module** dialog box.
- Acknowledge your selection by pressing the **OK** button.

You can create any hardware configuration by simply repeating this procedures for all desired modules.

Then you can declare variables, create objects, etc.



Take note that you can program a project without hardware, but you can't test it !





# APPENDIX B GLOSSARY

## User RAM

User RAM is a conditionally nonvolatile memory area (buffered when power is not applied to the controller) and is available for application software (cyclic objects, idle time objects, data objects, etc.).

## User ROM

User ROM can be used to store cyclic objects, idle time objects, data objects, etc. in nonvolatile memory.

User ROM is in the form of a FlashPROM.

## PCCSW

### Operating System

The PCC operating system (PCCSW stands for PCC software) is an adjustable, deterministic realtime-multitasking operating system. The B&R Automation Studio™ software configuration can be used to define the performance of the PCCSW operating system to meet the requirements of your application (additional information concerning the software configuration can be found in the online help).

The controller **cannot be used** without an operating system !

### Library

A collection of standard functions is represented in B&R Automation Studio™ by a B&R module (library). Each library contains functions that can save you time and effort solving standard problems. If a function from a library is used, that library is automatically imported into the project and installed on the controller during the next transfer.

### Boot Mode

"Booting" is the procedure carried out when the PCCSW operating system is restarted. A boot procedure can be triggered either by the user (e.g. with B&R Automation Studio™) or by a fatal error. B&R CPUs support various boot modes:

- COLDSTART
- WARMSTART
- RESET
- ERROR
- DIAGNOSE

**Data Object**

Data objects are modules used to store data.

**Data Type**

The data type defines the memory requirements and value range for a process variable (detailed information can be found in the online help).

**DIAGNOSE**

In DIAGNOSE mode, the controller boots with the PCCSW, i.e. all software objects are ignored and are therefore inactive.

**ERROR**

If a fatal error occurs during cyclic operation, the error is entered in the logbook. Then the controller automatically boots in ERROR mode and goes into SERVICE mode.

**Exception Resource**

Exceptions are signals that come from the PCCSW operating system and indicate certain exception states. They can be compared to interrupts triggered by the system to indicate a fatal error. Objects which are assigned an exception resource, can be used to react to these fatal errors.

**FIX RAM**

Using B&R Automation Studio™ (software configuration), you can configure a part of User RAM to behave like User ROM during a COLDSTART. This memory area is called **FIX RAM**. Objects stored in FIX RAM are not deleted during a COLDSTART.

**FlashPROM**

Read only memory that can be deleted electrically.

**Force Status**

If a process variable is forced, it is assigned a specific value and remains at this value. If an object writes to a forced process variable, the changed value is only valid for the current cycle. Then the variable is overwritten with the force value entered.

## Hardware Configuration

The left section of the project window is called the **hardware configuration**. The hardware configuration provides an overview of all hardware in the project.

## Initialization Subprogram

With B&R Automation Studio™, an initialization subprogram can be created for each cyclic object (INIT SP). Initialization subprograms are only executed **in the controller's boot phase** (after a COLDSTART or WARMSTART) or after an object is transferred to the controller. INIT SPs are especially useful for the initialization of values in a program.

### INIT SP

See *Initialization Subprogram*.

## Interrupt Resource

Interrupts are asynchronous event (triggered by hardware) that interrupt cyclic program execution.

Interrupts can only be generated by special hardware. An example of an interrupt capable module is the DI400 digital input module (B&R 2010). 8 digital inputs can be used as CSI inputs (CSI = "Change of State Interrupt") on this module. If the state of an enabled CSI input changes, an interrupt is triggered on the CPU.

An object can be created for each interrupt capable module to handle the event.

## COLDSTART

COLDSTART boot mode corresponds to the first time a processor is initialized. The operating system is restarted and all RAM data (except object stored in FIX RAM ) are deleted.

## MEM Card

Some processor modules can be equipped with a **PCMCIA memory card** (also called **MEM card**). Cyclic objects, idle time objects, data objects, etc. can be stored in nonvolatile memory on the MEM card. That means this memory is equivalent to User ROM (MEM card in FlashPROM form) or FIX RAM (MEM card in SRAM form).

## Multitasking

During multitasking operation, several objects (programs) share the processor's processing time. The objects run appear to be running at the same time. Switching between the objects quickly gives the impression that they are running simultaneously.

## PCMCIA Memory Card

See *MEM Card*.

## Project Window

When opening a project in B&R Automation Studio™, the **project window** is shown. The project window is the heart of each project. The window has two sections.

The **left section** provides an overview of all hardware in the project (hardware configuration).

The **right section** provides additional information and settings for the module marked.

## RESET

In RESET mode (similar to WARMSTART), all battery buffered RAM data (objects in User RAM, variable values, etc.) remains in tact after booting. After the RESET, the controller goes into SERVICE mode, i.e. software objects in User RAM, User ROM and FIX RAM are not started. In SERVICE mode, only the PCCSW is active, all software objects are inactive.

## Resource

You can assign various **resources** to object to define priority between the individual cyclic objects. A cycle time can be set for each cyclic resource and each timer resource. All objects assigned this resource are executed once within this cycle time.

Objects that carry out important, time critical tasks are assigned a resource with a shorter cycle time and are therefore executed more often than objects assigned a resource with a longer cycle time.

## Idle Time Object

Idle time objects run without cycle time monitoring. They use the system idle time. That means they are only executed if the PCCSW operating system and cyclic objects do not require the system time (CPU time).

## PCC

PCC (Programmable Computer Controller).

The difference to a standard PLC (Programmable Logic Controller) is the integration of computer functionality such as the ability to use high level programming languages, simultaneous execution of several programs on the CPU and memory in the megabyte range.

## PCCSW

See *PCCSW Operating System*.

## SERVICE

SERVICE mode is not a boot mode, instead a mode that can follow a boot procedure. In SERVICE mode, only the PCCSW is running, all software objects are inactive.

## Software Configuration

The **software configuration** is integrated in the right section of the project window. Each module that can run software (CPU, multiprocessor or parallel processor) has a register card **software**. This register card is also called the software configuration.

Using the software configuration, you can adjust the performance of the PCCSW operating system to meet the requirements of your application.

## System Object

B&R modules (\*.BR files) can be transferred to the controller as system objects. System objects include libraries or objects with additional system functions. Many of these objects are loaded by B&R Automation Studio™ automatically and don't have to be inserted manually.

## System ROM

The PCCSW operating system is stored in System ROM.

System ROM is a nonvolatile memory area in the form of a FlashPROM.

## Timer Resource

Objects with this resource are carried out cyclically using special hardware timers (interrupts). Timer resources offer a high degree of consistency and have higher priority than cyclic resources.

## **WARMSTART**

During a WARMSTART, all battery buffered RAM data (objects in User RAM, variable values, etc.) remains in tact after booting. After a WARMSTART, all objects have the same status (object being executed or object not being executed) as they did before the WARMSTART.

## **Cyclic Resource**

Objects assigned this resource are executed cyclically. The PCCSW operating system calls these objects.

## **Cyclic Object**

A cyclic object is executed in a defined time frame (cycle time). The operating system monitors to make sure the cycle time is not exceeded. To assign priorities, cyclic objects can be assigned various resources (different cycle times). Cyclic objects that carry out important, time critical tasks are assigned a resource with a shorter cycle time and are therefore executed more often than objects assigned a resource with a longer cycle time. Cyclic objects include:

- Cyclic objects
- Timer objects
- Interrupt objects
- Exception objects





# APPENDIX C INDEX

## A

ANSI C .....	43
Automatic Hardware Recognition .....	30

## B

B&R Automation Studio	
Exiting .....	23
Starting .....	21
Boot Mode .....	74
COLDSTART .....	76
DIAGNOSE .....	75
ERROR .....	75
RESET .....	77
SERVICE .....	78
WARMSTART .....	79

## C

COLDSTART .....	50, 76
Compiler .....	49
Connection Status .....	35
Conventions .....	12
Cyclic Object .....	41, 79
Creating .....	40
Cyclic Resource .....	43, 79

## D

Data Object .....	41, 75
Data Type .....	58, 75
DIAGNOSE .....	75

## E

ERROR .....	75
Example Program .....	24
Exception .....	44
Exception Resource .....	44, 75
Exiting B&R Automation Studio .....	23

## F

FIX RAM .....	50, 75
FlashPROM .....	64, 75
Force Function .....	53
Force Status .....	61, 75
Forced .....	53

## H

Hardware Configuration .....	36, 76
Hardware Recognition	
Automatic .....	30
Hardware Definition	
Manual .....	32
Help System .....	See Online Help

## I

Idle Time Object .....	41, 77
Installation	
PCCSW Operating System .....	64
Windows NT .....	17
Windows 95 .....	15
Windows 98 .....	15
Instruction List .....	43
Interrupt .....	44
Interrupt Resource .....	44, 76

## K

Keyboard Operation .....	23
LAD Editor .....	See Ladder Diagram Editor
LAD Monitor .....	See Ladder Diagram Monitor

## L

Ladder Diagram .....	43
Ladder Diagram Editor .....	45
Ladder Diagram Monitor .....	52
Library .....	41, 74

## M

Main Menu .....	22
Manual Hardware Definition .....	71
Manual Hardware Definition .....	32
MEM Card .....	See PCMCIA Memory Card
Memory Area .....	49
Memory Areas	
FIX RAM .....	See FIX RAM
MEM Card .....	See PCMCIA Memory Card
SYS ROM .....	See System ROM
User RAM .....	See User RAM
User ROM .....	See User ROM
Message Window .....	22
Mouse Operation .....	23
Multitasking .....	77

## N

Non-Cyclic Object ..... See Idle Time Object

## O

### Object

Compile ..... 49, 50  
Testing ..... 52  
Transfer ..... 49, 50

Object Type (programming language) ..... 43

### Object Type

C Language ..... See ANSI C  
Instruction List ..... See Instruction List  
Ladder Diagram ..... See Ladder Diagram  
Sequential Function Chart ..... See Sequential  
Function Chart  
Structured Text ..... See Structured Text

Online Connection ..... 27, 28

Online Help ..... 13

Online Cable ..... 27

Online Connection ..... 35

Operating System Download ..... 26, 64

Processors with Application Memory ..... 64

Processors without Application Memory ..... 68

## P

PCC ..... 78

PCCSW Operating System ..... 26, 64, 74

PCMCIA Memory Card ..... 50, 76

Process Variables ..... 38

Program Start ..... 20

Programming Language ..... See Object Type

Project ..... 26

Creation ..... 26, 30

Open ..... 31, 33

Project Window ..... 36, 77

Left Section ..... 36

Right Section ..... 37

PV ..... See Process Variable

## R

RESET ..... 77

Resource ..... 42, 43, 77

Cyclic Resource ..... 43

Exception Resource ..... 44

Interrupt Resource ..... 44

Timer Resource ..... 43

Resource Type ..... 41

### Resource Types

Cyclic Object ..... See Cyclic Object

Data Object ..... See Data Object

Non-Cyclic Object ..... See Idle Time Object

System Object ..... See System Object

## S

Sequential Function Chart ..... 43

SERVICE ..... 78

Software Configuration ..... 37, 78

Starting B&R Automation Studio ..... 21

Status Bar ..... 22

Structured Text ..... 43

System Monitor ..... 55

System Object ..... 41, 78

System Requirements ..... 14

System ROM ..... 50, 64, 78

## T

Timer Resource ..... 43, 78

Tool Bar ..... 22

## U

User RAM ..... 49, 74

User ROM ..... 49, 74

## V

Variable Declaration ..... 38

Variable Declaration ..... 38

## W

WARMSTART ..... 79

Watch ..... See Watch Function

Watch Function ..... 58

