POSITIONING AND CONTOURING CONTROL SYSTEM

APCI-8001 AND APCI-8008

COMMISSIONING MANUAL / CM

Last updated: 29/07/2021, from disk V2.53VM
Board revision: APCI-8001 Rev. D
APCI-8008 Rev.C
APCI-8008-STP-EVAI Rev.B
Rev. 17/052022

www.addi-data.com

Warranty and liability

Warranty and liability claims for personal injury and damage to property shall be excluded, if they are attributable to one or more of the following causes:

- Use of the board not in accordance with the regulations
- Improper installation, commissioning, operation and maintenance of the board
- Operation of the board in defective security installations or in improperly fitted or non-operational security and protection appliances
- Non-compliance with the notes in the operating instructions regarding transport, storage, installation, commissioning, operation, thresholds, maintenance of the board, standard software
- Arbitrary changes to the board
- Arbitrary changes to the source code files
- Insufficient monitoring of parts which are subject to wear
- Catastrophes due to effects by foreign bodies or acts of God.

ADDI-DATA software product licence

Please read this licence carefully before using the standard software. The customer is only granted the right to use this software if he/she agrees with the conditions of this licence.

- The standard software must only be used to set up the ADDI-DATA boards
- Reproduction of the software is forbidden (except for back-up and for exchange of faulty data carriers)
- Disassembly, decompilation, decryption and reverse engineering of the software are forbidden.
- This licence and the software may be transferred to a third party if this party has acquired a board by purchase, has agreed to all the conditions in this licence contract and the original owner does not keep any copies of the software.

Warning!

The following risks result from the improper implementation of the board and from use contrary to the regulations:

Personal injury

Damage to the board, the PC and peripherals

Pollution of the environment.

- Protect yourself, others and the environment!

- Read the safety precautions (yellow leaflet) carefully!

  If this leaflet is not enclosed with the documentation, please contact us and ask for it.

- Observe the instructions of this manual!

  Make sure that you do not forget or skip any step!
  We are not liable for damages resulting from the wrong use of the board.

- Pay attention to the following symbols:

ℹ **NOTICE!**
Designates hints and other useful information.

ℹ **NOTICE!**
Designates a possibly dangerous situation.
If the instructions are ignored, the board, the PC and/or peripherals may be **destroyed**.

⚠ **WARNING!**
Designates a possibly dangerous situation.
If the instructions are ignored, the board, the PC and/or peripherals may be **destroyed** and persons may be **endangered**.

1 Introduction

What is this manual for?	This manual describes how to commission all the necessary system components for using the **xPCI-800x** positioning and contouring control system. The complete manual is made up of three parts: OM (Operating Manual), CM (Commissioning Manual) and PM (Programming and Reference Manual).
What devices belong to the xPCI-800x range?	The xPCI-800x range of products are third-generation positioning and contouring control systems. At present, these include the APCI-8001 and APCI-8008 positioning and contouring control systems.
Other remarks	Where the functions described in this manual do not apply for all xPCI-800x products, this is marked clearly. In this case, the corresponding function only applies for the product indicated.

2 Intended use

The xPCI-800x motion control boards (APCI-8001 and APCI-8008) form the interface between industrial process, automation and drive technology and a personal computer (PC).

The board is suitable for use in a PC that has been fitted with free PCI slots. The PC is subject to the European EMC directive and must satisfy the corresponding EMC protection requirements.

Products which satisfy these requirements have the CE mark.

2.1 Special notes for the APCI-8001 and APCI-8008

Data exchange between the APCI-8001 and APCI-8008 boards and the peripherals takes place via a shielded cable. This cable must be connected to the 50-pin SUB-D male connector of the APCI-8001 or APCI-8008.
The APCI-8001 or APCI-8008 also has digital outputs for processing 24 V signals. An external 24 V supply voltage is required to operate these outputs.
The PX8001 screw terminal panel enables connection of the 24 V supply voltage via a shielded cable.
The use of the APCI-8001 or APCI-8008 in combination with external screw terminal boards requires technical installation in a closed switch cupboard. Check the screen attenuation measure of PC housing and cable shield before commissioning the device.

Use of the standard ST8001 cable satisfies the minimum requirements:

- metallised connector shells,
- shielded cable,
- cable screen folded back over insulation and screwed down on both sides with the connector shell.

2.2 Special notes for the APCI-8008-STP-EVAI

The board APCI-8008-STP-EVAI enables the user to control 6 stepper axes with encoder verification.
For this purpose, the board is equipped with 16 digital inputs and a maximum of 8 digital outputs.

The inputs and outputs are shared for all 6 axes. Hardware latch inputs are:

I11 for axis 1
I12 for axis 2
I13 for axis 3
I14 for axis 4
I15 for axis 5
I16 for axis 6

The outputs are optional. The +24 V supply at pin 59 of X1 is required only if digital outputs are used. At the respective pins (digital outputs), also CNC ready relays or analog inputs can be made available.
The corresponding configuration is fitted ex works. The external components are connected via a 78-pin SUB-D female connector (X1). The OPMF-8008 cannot be used for this board version.

The scope of software is more or less identical to that of the other xPCI-800x boards. However, an interpolation is only possible with the first 4 axes. The SMLA and SMLR commands can be called up for axes 5 and 6 as well, but only as single-axis traversing commands.

2.3 Limits of use

By using the boards in a PC, the resistance to interference and emission values of the PC may change. Increased emissions or reduced resistance to interference can mean that the conformity of the system is no longer ensured.

EMERGENCY STOP functions must not be taken over by any of the above-mentioned modules.

The EMERGENCY functions must be protected separately. This protection must not be influenced by the boards and the PC.

The board must remain in its anti-static packaging until it is installed.

The warranty claim becomes invalid if the identification numbers are deleted or changed.

2.4 Users

2.4.1 Qualification

The following activities must only be executed by trained electronics employees:

- Installation
- Commissioning
- Operation
- Maintenance

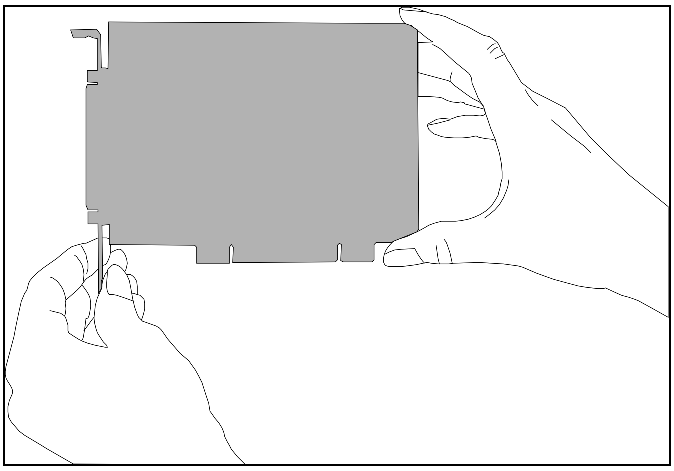
2.4.2 Personal protection equipment

Please note the national regulations regarding:

- Accident prevention
- Installation of electrical and mechanical equipment
- Electromagnetic compatibility (EMC).

2.5 Handling the board

Fig. 2.1: How to handle the board correctly



Please note:
The following procedures must be observed during commissioning:

- Installing the xPCI-800x TOOLSET software [section 3] – Installing the board [section 4]
- Configuring and wiring the xPCI-800x board [section 5]
- Settings and planning as described in section 6 and section [OM / section 4.2].

2.6 Important!

Before installing or removing any modules, ensure that the PC and the external power supplies have been switched off.
If this instruction is ignored, the board or computer could be destroyed.

The manufacturer shall not be liable for damage which could result from the use or implementation of its products.

The manufacturer shall not be liable for errors of any type, which may be contained in these manuals. The manufacturer further reserves the right to alter this manual, and the specifications of the product described, without having to divulge or announce these alterations in any form to any person or persons.

3 Installing and configuring the xPCI-800x TOOLSET software

3.1 Scope of delivery of the xPCI-800x TOOLSET software

The xPCI-800x TOOLSET software [TSW] is delivered on a CD. This essentially contains the following parts:

- INF file with Miniport device driver
- mcfg utility program
- fwsetup utility program
- Command line program
- Firmware program and files
- Libraries + programming examples
- Documentation

3.2 Installing the miniport (rnwmc device driver)

The sub-directory \Inf\WIN_2K_XP_Vista_7 of the xPCI-800x TOOLSET CD contains the file rnwmc.inf. This file has to be specified as an installation file for the plug&play installation and accordingly in the device manager. Thus, the current hardware driver version will be installed.
For earlier operating systems, the INF file of the respective directory has to be indicated. In addition, the Miniport driver needs to be installed using ksetup.exe.

3.3 Installing MCFG

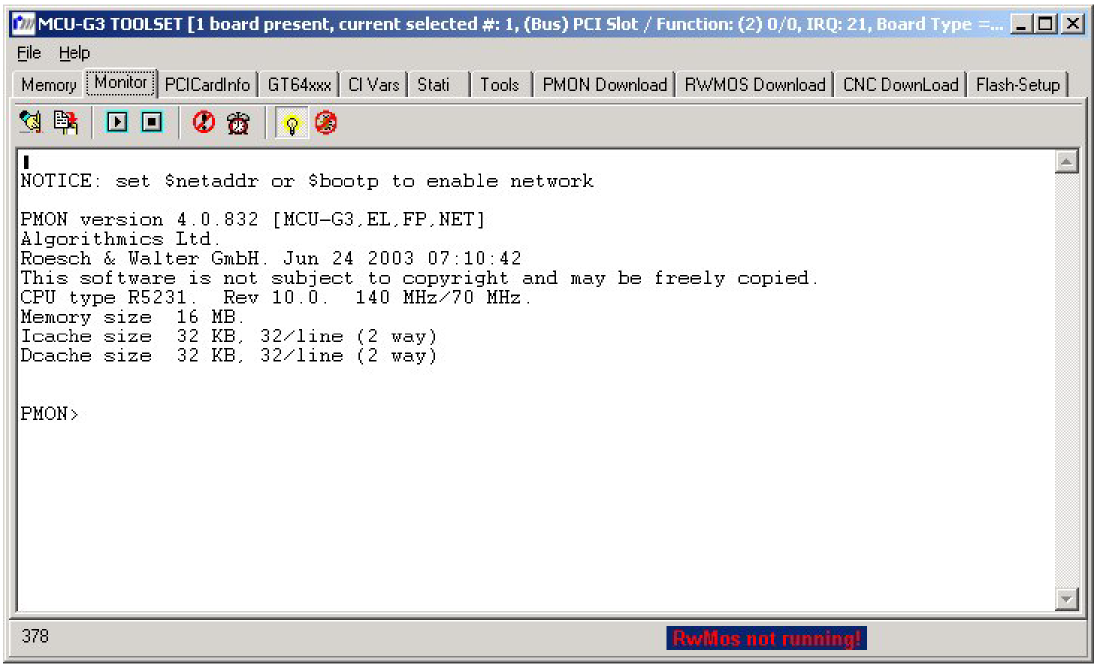
Call the setup.exe installation program in the mcfg sub-directory of the xPCI-800x TOOLSET CD. This is an installation package based on the Microsoft Installer.

3.4 Installing FWSETUP

Call the setup.exe installation program in the fwsetup sub-directory of the xPCI-800x TOOLSET CD. This is an installation package based on the Microsoft installer.

3.5 Starting FWSETUP

Start the fwsetup.exe application. The following screen output or any identical one should be displayed on the "Monitor" page:



```
NOTICE: set $netaddr or $bootp to enable network

PMON version 4.0.832 [MCU-G3,EL,FP,NET]
Algorithmics Ltd.
Roesch & Walter GmbH. Jun 24 2003 07:10:42
This software is not subject to copyright and may be freely copied.
CPU type R5231.  Rev 10.0.  140 MHz/70 MHz.
Memory size  16 MB.
Icache size  32 KB, 32/line (2 way)
Dcache size  32 KB, 32/line (2 way)


PMON>
```

The header line of the fwsetup application includes information on the number of identified xPCI-800x controllers and the board type, among other things. The version of the monitor program PMON stored for the control system, the CPU frequency and the memory status for the control system are displayed in the window. You can reset the xPCI-800x controller by pressing the soft reset button. In this case, the screen output in the monitor window should be filled in with other screen messages. In this state, environment variables can be set in the control system. These can be used to set properties of the control system. You will find more information on this in Chapter 3.12.
In the figure above, the so-called fwsetup monitor screen is shown. As long as the control system is not booted yet, the system is in the PMON monitor program. Here, different commands may be entered. The available commands can be listed using the h command. However, only operations that are really wanted should be called here, as otherwise, the system could become unusable.

3.6 Creating a system directory

- Create a file for storing the most important system files.
- In this folder, generate a system.dat file using one of the 3 methods outlined below:

  By creating a new file using the sysgen.exe help program from the Toolset sub-directory of the xPCI-800x TOOLSET CD or

  by copying a file from the firmware and system.dat files sub-directory\{Controller name of the xPCI-800x TOOLSET CD (**Notice:** Delete read only attribute after the copying process) or

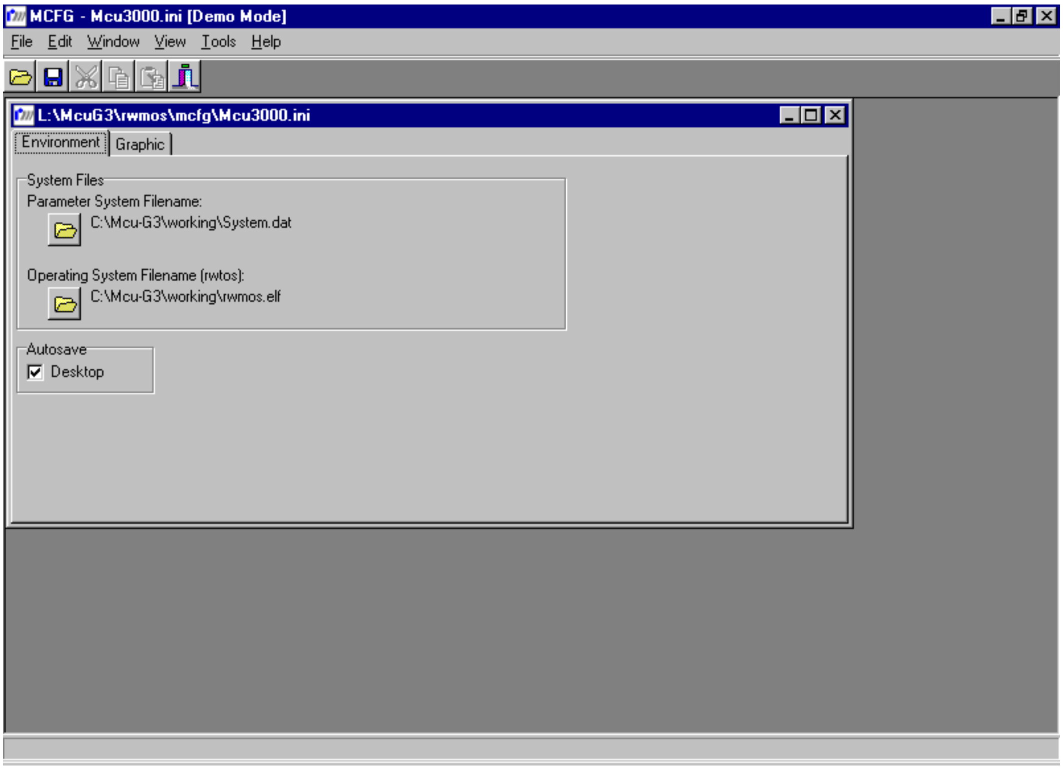
- by converting an existing system.dat file using the sysconv.exe help program from the Toolset sub-directory of the xPCI-800x TOOLSET CD.

- Likewise, copy the RWMOS.ELF file from the Firmware and system.dat files sub-directory\ { Controller name } into this newly created system directory.

Important note: The RWMOS.ELF and SYSTEM.DAT files are specifically configured for the various module types. The SYSTEM.DAT file can be converted where required.
The RWMOS.ELF file requires the right type to be selected from the Toolset software. This also applies to the SYSTEM.DAT file, but where applicable, this file may as well be adapted through conversion using sysconv.exe.

3.7 Setting up the MCFG project environment

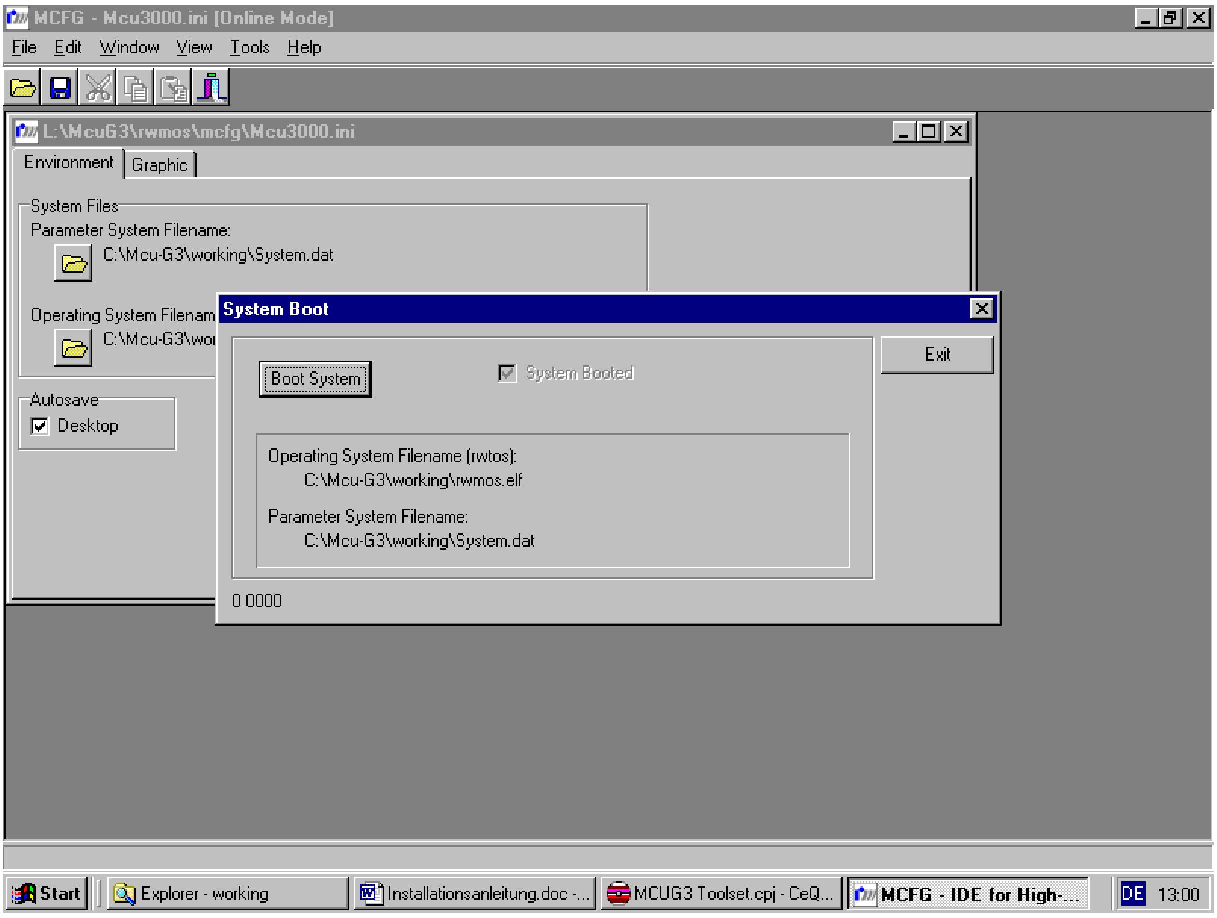
- Start the mcfg.exe application
- In the [File][Project Parameter] menu, update the entries according to the system directory you have selected (see above).
- Save the project settings via [File][Save As] or [File][Save].


MCFG - Mcu3000.ini [Demo Mode]
File
Edit
Window
View
Tools
Help
L:\McuG3\rwmos\mcfg\Mcu3000.ini
Environment
Graphic

System Files
Parameter System Filename: C:\Mcu-G3\working\System.dat
Operating System Filename (rwtos): C:\Mcu-G3\working\rwmos.elf
Autosave
☑ Desktop

3.8 Booting the xPCI-800x controller

- Open the dialogue box [Tools][System Boot]
- Press the [Boot System] button
- After a few seconds, the [System Booted] check box must be marked and the [Online Mode] entry should appear in the header line of the mcfg application.



Parameter System Filename:
C:\Mcu-G3\working\System.dat

Operating System Filenam
C:\Mcu-G3\wo

☑ Desktop

☑ System Booted

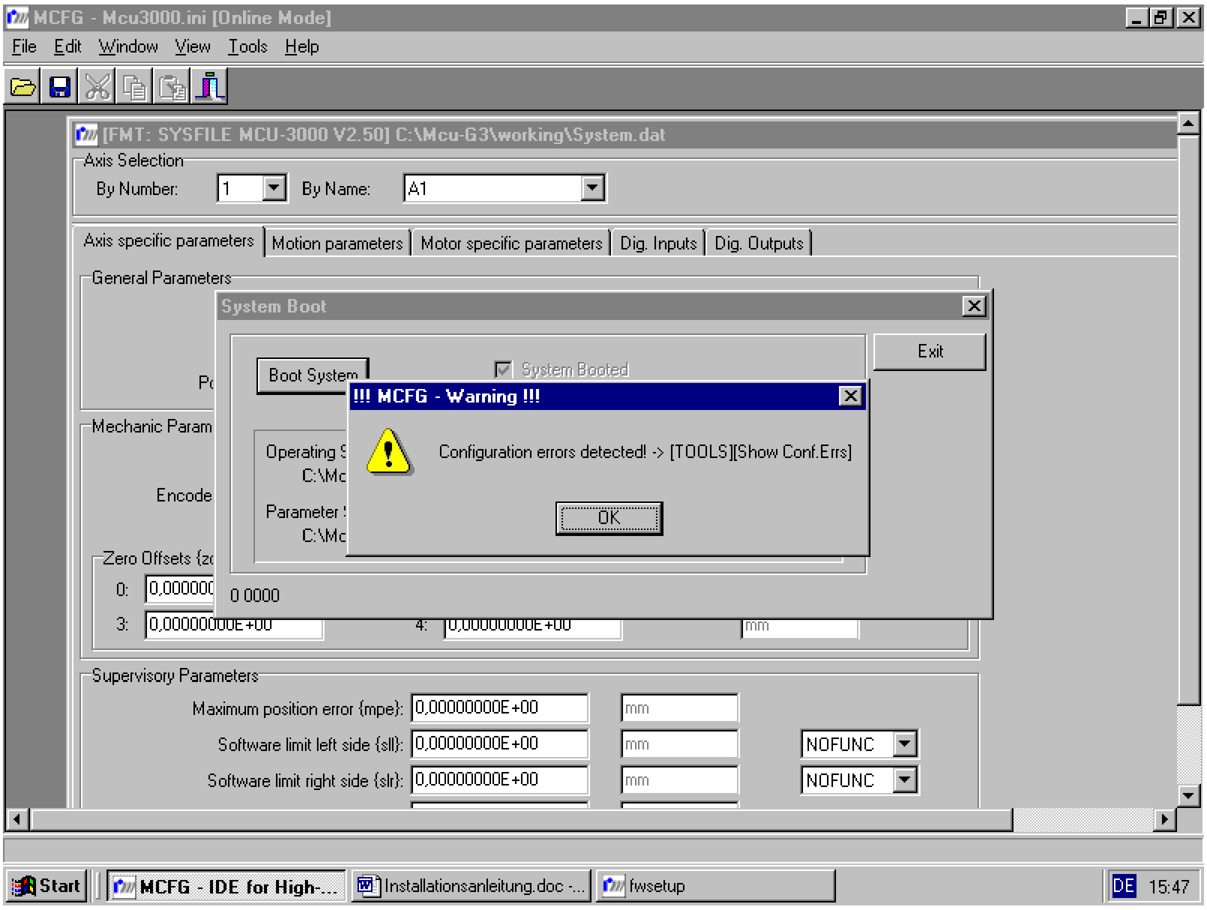
Operating System Filename (rwtos):
C:\Mcu-G3\working\rwmos.elf

Parameter System Filename:
C:\Mcu-G3\working\System.dat

0 0000

3.9 Configuration errors

If the following error message is displayed on the screen after you have booted the control system, this is caused by the following:
The system file (system.dat) and system data selected in the project, which are stored residually in the flash memory of the xPCI-800x controller control system, are different. This error can be cleared through a storage procedure as described in the next section.



MCFG - Mcu3000.ini [Online Mode]

[FMT: SYSFILE MCU-3000 V2.50] C:\Mcu-G3\working\System.dat

Axis Selection: By Number: 1 By Name: A1

!!! MCFG - Warning !!!

Configuration errors detected! -> [TOOLS][Show Conf.Errs]

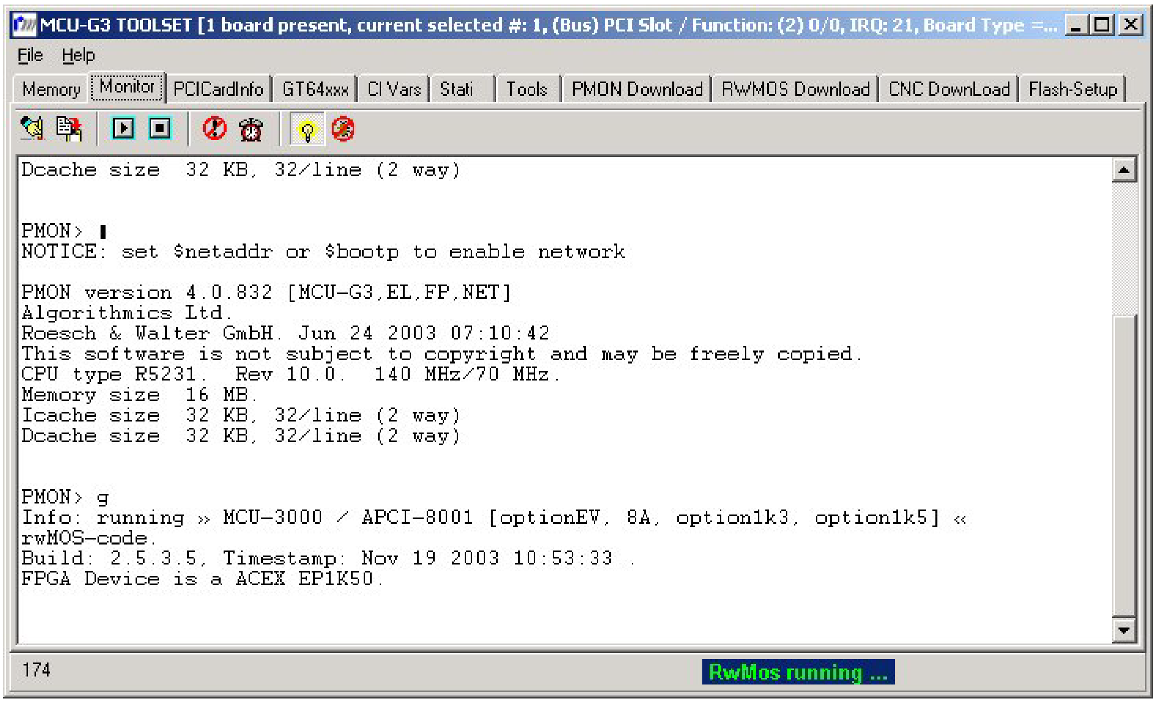
Maximum position error {mpe}: 0,00000000E+00 mm

Software limit left side {sll}: 0,00000000E+00 mm NOFUNC

Software limit right side {slr}: 0,00000000E+00 mm NOFUNC

3.10 New calling of fwsetup.exe

By calling the program fwsetup again an output identical to the following examples is displayed in the monitor page:

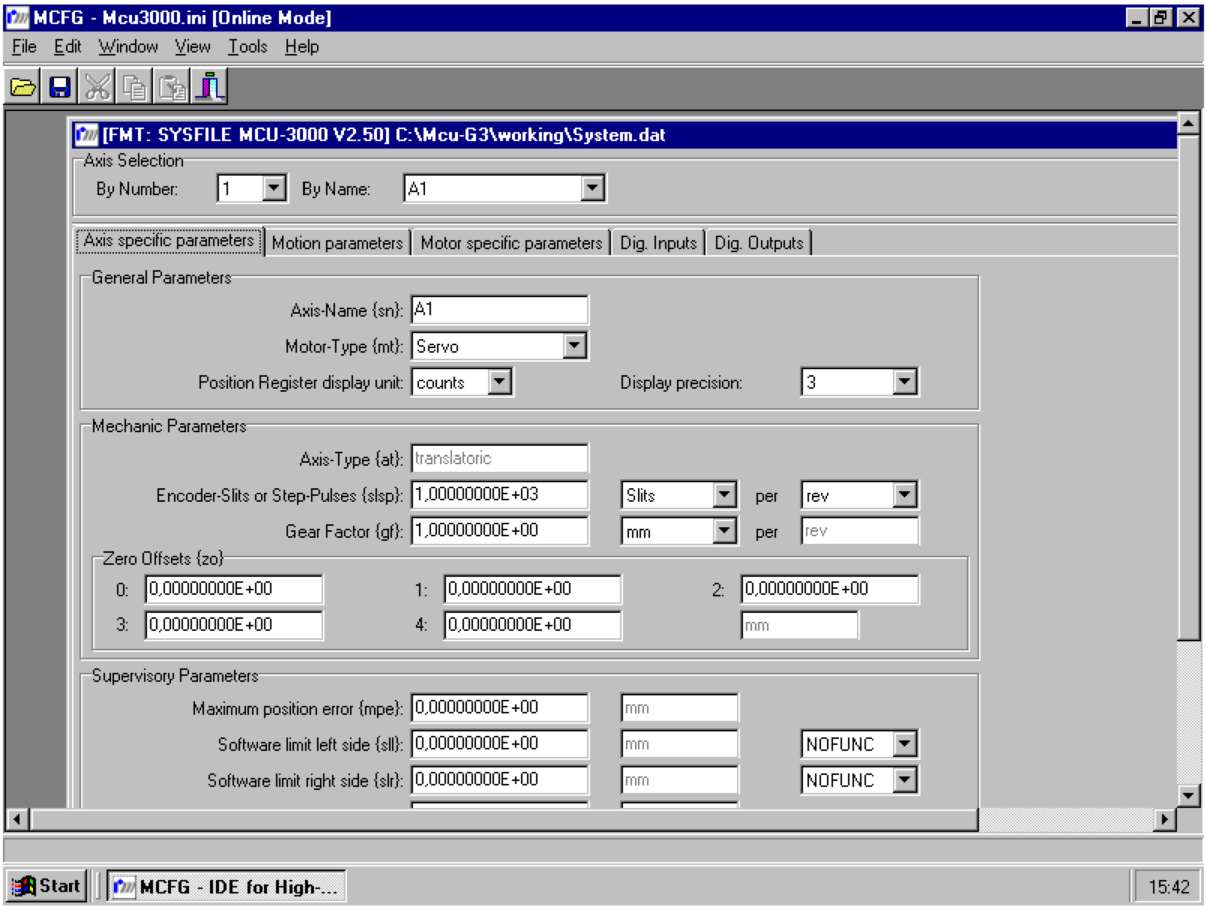


```
Dcache size  32 KB, 32/line (2 way)


PMON> ▮
NOTICE: set $netaddr or $bootp to enable network

PMON version 4.0.832 [MCU-G3,EL,FP,NET]
Algorithmics Ltd.
Roesch & Walter GmbH. Jun 24 2003 07:10:42
This software is not subject to copyright and may be freely copied.
CPU type R5231.  Rev 10.0.  140 MHz/70 MHz.
Memory size  16 MB.
Icache size  32 KB, 32/line (2 way)
Dcache size  32 KB, 32/line (2 way)


PMON> g
Info: running » MCU-3000 / APCI-8001 [optionEV, 8A, option1k3, option1k5] «
rwMOS-code.
Build: 2.5.3.5, Timestamp: Nov 19 2003 10:53:33 .
FPGA Device is a ACEX EP1K50.
```

In addition to the screen displayed in section 3.5 the footer indicates that rwMos is running. In the monitor window data about the started operating system variant RWMOS.ELF are listed. These are first the software options available in the booted RwMos variant. Then the operating system version with creation date and information about the hardware equipment of the control variants (here ACEX EP1K50). This display is to be noticed when errors occurs by booting or during the operation. You can find here in most cases indications with which the problem cause can be found.

3.11 Entering and storing system data

In the next step, you should edit the system data for the axis channel being used in the programm mcfg. To do this, proceed as follows:


MCFG - Mcu3000.ini [Online Mode]

File Edit Window View Tools Help

[FMT: SYSFILE MCU-3000 V2.50] C:\Mcu-G3\working\System.dat

Axis Selection
By Number: 1
By Name: A1

Axis specific parameters | Motion parameters | Motor specific parameters | Dig. Inputs | Dig. Outputs

General Parameters
Axis-Name {sn}: A1
Motor-Type {mt}: Servo
Position Register display unit: counts
Display precision: 3

Mechanic Parameters
Axis-Type {at}: translatoric
Encoder-Slits or Step-Pulses {slsp}: 1,00000000E+03 Slits per rev
Gear Factor {gf}: 1,00000000E+00 mm per rev

Zero Offsets {zo}
0: 0,00000000E+00
1: 0,00000000E+00
2: 0,00000000E+00
3: 0,00000000E+00
4: 0,00000000E+00
mm

Supervisory Parameters
Maximum position error {mpe}: 0,00000000E+00 mm
Software limit left side {sll}: 0,00000000E+00 mm NOFUNC
Software limit right side {slr}: 0,00000000E+00 mm NOFUNC

- The control system must be booted [Online Mode in the header line of the mcfg.exe application]
- mcfg.exe: [File] [System Data] enter the axis parameters for the respective axes
- Save the new system data using the [File][Save] command.

During the save process, the different parameters are stored residually in the flash memory of the xPCI-800x controller. If a configuration error was displayed, this must not be displayed again after the system data has been stored. The save process is finished after a few seconds.

3.12 Control system hardware environment variables

Environment variables can be set in the control system hardware environment, to configure the control system hardware or software. These environment variables are set and reset using the fwsetup.exe configuration program when the system is not booted. To set environment variables, the instruction:

set variable value

is entered in the data window of the “Monitor” tab page. A repeated assignment overwrites a previous assignment. To delete the environment variables, the instruction:

unset variable

is entered. Note that the spelling of these instructions and parameters must be exactly right (high/low case). The current status of the environment variables can be displayed by entering the instruction

set

(without parameters). If the display is not complete, press “Enter” to display another environment variable until all the variables have been listed in the monitor window.

From RWMOS.ELF V2.5.3.37 andmcug3.dll V2.5.3.25 on, in the PCAP programming environment, environment variables of the control can be read out. Hereto the function getEnvStr() is available. However, in this way the setting of environment variables is not possible.

Important note: The status of the environment variable is an important characteristic of the respective control system and must be documented. For example before using a replacement device or by reproduction of a system, the entries made by the user must be restored. The screen contents can be copied into a text file via the Windows clipboard, for documentation purposes.
Please note also that the formatting of these instructions and parameters must be exactly correct in terms of upper and lower-case, or the entry will be invalid.
These environment variables are **only required in special cases**. For normal servo and stepper motor axes, environment variables do not need to be set. Incorrect use of these variables may compromise the operation of the module.

3.12.1 The environment variable MT (MotorType)

This environment variable can adapt the control system to different axis systems. The axis allocation occurs by attaching the axis index to MT (z.B. MT0). The different configurations possible are listed in the table below. To select an option, enter the value of the corresponding system variable MT.

Important note: The following system variables are only needed in special cases (e.g. where an SSI absolute value encoder is used). For standard applications with stepper motor systems, servo systems and incremental encoder systems, these variables need not, and in fact must not, be set.

Table: Motor types

Value	Name	Description
2	SSI	Axis with actual value acquisition of SSI encoder SSI encoder and analog manipulated variable output.
3	INC PWM	Axis with PWM manipulated value signal (RS 422) to the pulse output and actual value acquisition through incremental encoder. The directional signal is transmitted to the outputs *Sign*.
4	STEPPER SSI	Axis with actual value acquisition of SSI encoder and manipulated variable output of stepper motors.
5	ANALOG PWM	Axis with PWM manipulated value signal (RS 422) to pulse output and actual value acquisition through analog input. The directional signal is transmitted to the outputs *Sign*.
6	STEPPER NDX	This axis type enables to transmit the directional signal of the stepper motor to a digital output (24V). The pins *NDX/Sign* are inputs and can be used for the evaluation of a zero track (index).
7	ANALOG / ANALOG	Axis with analog manipulated variable output and actual value acquisition through analog input.
8	Encoder emulation	With this axis type, an incremental signal is output as manipulated value (Encoder simulation). The axis must also be defined in mcfg as stepper motor axis.
9	Piezo motor	This motor type is optimised for driving Piezo motors of the Nano-Motion Company.
10	PSM	Axis with digital performance final level PSM-1150 over PSM bus.
11	ENDAT2.1	Axis with ENDAT encoder (serial data interface) and with incremental actual value acquisition and analog manipulated variable output.
12	INC_PULSE	Servo axis with pulse direction interface and incremental encoder actual value acquisition (controlled stepper motor)
13	VIRTUAL	Virtual axes: Virtual axes cannot be used for axes control, but they can be used for profile generator calculation.
14	GEOADD	Axis type for the graphical view of trackdata of virtual axes.
15	UPDOWNSIGNALS	Option for presenting encoder signals of an axis channels as UP or DOWN counting signals.
16	ENDAT 2.2	Axis with ENDAT encoder (serial data interface) without incremental actual value acquisition and analog manipulated variable output.
19	ANA_SIGN	Axis with analog manipulated variable output and actual value acquisition through incremental encoder. The analog output signal is always positive; the directional information is output through the digital output.
20	CI / ANALOG	Axis with analog manipulated variable output and actual value acquisition through Common Integer variable (CI).
21	CD / ANALOG	Axis with analog manipulated variable output and actual value acquisition through Common Double variable (CD).
22	STEPPER / ENDAT 2.2	Axis with step/direction output and ENDAT 2.2 encoder verification

Value	Name	Description
23	ETM	Customer-specific special version with actual value acquisition through pulse duration measurement
24	ANA_SIGN_SSI	Axis with analog manipulated variable output and actual value acquisition through SSI absolute encoder. The analog output signal is always positive; the directional information is output through the digital output.
25	ANA_SIGN ENDAT2_2	Axis with analog manipulated variable output and actual value acquisition through ENDAT 2.2 absolute encoder. The analog output signal is always positive; the directional information is output through the digital output.
26	SSI_PULSE	Servo axis with pulse direction interface and SSI absolute encoder actual value acquisition (controlled stepper motor)
27	ENDAT22_PULSE	Servo axis with pulse direction interface and ENDAT 2.2 absolute encoder actual value acquisition (controlled stepper motor) In preparation!
28	INC_PULSE_NDX	Servo axis with pulse direction interface and incremental encoder actual value acquisition (controlled stepper motor) Contrary to motor type 12, with this type of axis, the stepper motor direction signal is output at a digital output (24 V). The NDX/Sign pins are inputs here and can be used for the analysis of an index trace.
29	NDX_DIAG	Special version for diagnostic purposes
30	ENDAT_SNIFFER	Passive retrieval at an Endat 2.2 actual value channel. Only available with appropriate operating system version.

Example: set MT2 6

3.12.1.1 SSI motor type (2)

With this motor type, an SSI absolute value encoder can be used for the acquisition of actual values. This type is valid for stepper and servo axes. For the configuration of the SSI parameters, the environment variables SSIF and SSIP can still be set. These have to be set in an axis-specific way. The “?” character represents the axis index 0...7.
With SSIF the clock frequency can be reduced for reading the encoder. The wished frequency is entered as a value in Hz between 100 kHz and 4 MHz. The default value is 500 kHz. For long transmission lines the frequency must be generally reduced.
By assigning the environment variable SSIBIN? with the value 1, encoder axes can be set on binary code. Default: Gray code (see also section 5.2.6.1.).

Further notes on the configuration of SSI absolute encoders:

- The unit for the encoder resolution in the system file must usually be set to “Pulses” with SSI encoders (With “Slits”, the electronic quadruplication of incremental encoder signals is taken into account).
- The number of SSI pulses indicated with SSIP exceeds the number of user data bits by 2 (the default for SSIP is 26 – suitable for 24-bit encoders)

3.12.1.2 INC PWM motor type (3)

This motor type has an incremental encoder for the acquisition of actual value and a pulse width modulated output signal (PWM) as command value. The PWM output has a basic frequency of 20 kHz and a resolution of 3500 steps with the APCI-8001 (3333 steps with the APCI-8008) plus sign and is available as RS422 signal at the pins servo/puls+ and AGND/puls-. For this option a specific firmware RWMOS.ELF is necessary. In mcfg, this axis has to be set as SERVO. This option is not available for axes which can also be defined as STEPPER.

3.12.1.3 STEPPER SSI motor type (4)

Alternative method to define a stepper axis with SSI encoder feedback.

3.12.1.4 ANALOG PWM motor type (5)

This motor type has an analog input for actual value acquisition and a pulse width modulated output signal (PWM) as a manipulated variable. The PWM output has a basic frequency of 20 kHz and a resolution of 3500 steps with the APCI-8001 (3333 steps with the APCI-8008) plus sign. For this option a specific firmware RWMOS.ELF is necessary. In mcfg, this axis has to be set as SERVO.

With axes that can also be defined as STEPPER, this option is only possible to a limited extent:

- The axes must not include any SSI option.
- RWMOS.ELF must include the option optionSTPPWM.
- PWM and direction output are the pins CHA-CLKSSI and CHB-DATSSI.

3.12.1.5 STEPPER NDX motor type (6)

For this motor type, the directional signal is not given to the RS422 outputs Sign+ and Sign- but through the digital output. This digital output must be configured in mcfg (from the version V2.5.3.3) by selecting the option „SIGN SPEC“. Herewith, the zero track of the encoder can also be connected and evaluated when the encoder inputs are used.
This option is available from the version RWMOS.ELF V2.5.3.3.

3.12.1.6 ANALOG/ANALOG motor type (7)

Motor type with analog manipulated variable output and analog actual value acquisition. For this, also the environment variable FBCH? (Section 0) can be set to assign an analog actual value channel to an axis.

3.12.1.7 Encoder Emulation motor type (8)

Stepper motor type: Yet the command value is not output as a step or directional signal but as an emulated incremental encoder signal. For this option, a specific firmware RWMOS.ELF is necessary.

3.12.1.8 Piezo-Motor motor type (9)

The motor type is adapted to Piezo motors of the company Nano-Motion. The filter parameters kp, ki, kd and kfcv have the same significance as for standard controllers except the integral component which is handled in a different way. For the controller parameter kpl an amplitude value (peak/peak) can be entered in digits. This is superposed to the command value with half the sampling frequency. The position value entered in the target window is used for commuting the controller structure. Therefore the value to be entered must absolutely be targeted as resolution.
Further important parameters for this motor type are the compensation voltages mcpcp and mcpcn.

3.12.1.9 PSM motor type (10)

This motor type is a motor that is controlled through a digital performance end level at the serial field bus PSM bus. At the moment hereto the module PSM-1150 is available, that was designed for the control of brush covered direct current motors for the performance class up to approx. 12 A (nominal current) 60 V.

3.12.1.10 ENDAT motor type (11+16)

For this motor type, an ENDAT absolute value encoder or an incremental encoder for the actual value acquisition can be used. Here you have to distinguish between the Endat versions 2.1 and 2.2 (MT = 11 or MT = 16). Currently, this type is only available for servo axes. For the additional configuration of ENDAT parameters, the environment variable ENDATF can be set. With ENDATF, the clock frequency can be projected for reading and writing of the encoder. The value to be entered is the desired frequency in Hz (between 100 kHz and 2 MHz). The default value is 500 kHz. For long transfer lines, the frequency must be reduced. As each ENDAT axis can be projected with different frequency, the axis index must be attached to ENDATF (e.g. ENDATF3)

3.12.1.11 INC_PULSE motor type (12)

With this motor type, a position control is carried out in the same way as with default servo systems. The Motor-Type {mt} variable in mcfg must be set to SERVO. The default frequency range of the pulse output is +/-2 MHz. The frequency range can be limited using the variables {mcpmax} and {mcpmin}. The unit of these variables is 200 kHz. If a setpoint value jump (OL Response) is output, the output frequency is also indicated in the unit of 200 kHz.
The position controller must be set according to the same criteria as a speed-controlled system.
Notice: An axis can only be configured for this motor type if in RWMOS, the resources incremental encoder analysis and pulse output are available for the corresponding axis.

3.12.1.12 VIRTUAL motor type (13)

With the help of virtual axes the track velocity can be calculated in cartesian coordinates. The real axes can participate in this interpolation as Non-Feed-Rate axes. Thus a constant track velocity is guaranteed in axis systems with non-cartesian design.
The maximum number of virtual axes is compiled in RWMOS.ELF without any relation to the hardware.
This value can be determined in the RWMOS boot message in fwsetup. The actual number of virtual axes is then set using the environment variable VirtualAxis.
With virtual axes it is possible to realise, for example, a system with eight real axes and three additional virtual axes. The axis type VIRTUAL is not given manually, but is an internal constant that is automatically assigned to the virtual axes.
So virtual axes are the axes with an index higher than NumberAxis.

3.12.1.13 GEOADD motor type (14)

With the help of this axis type the track data of virtual axes can be visualized in the graphical system analysis. In this way the track velocity and the traverse path of virtual axes can be visualized. This axis type is only available for diagnostics.

3.12.1.14 UPDOWNSIGNALS motor type (15)

This option can be used only with an adequate RWMOS.ELF operating system version.

3.12.1.15 ANA_SIGN motor type (19)

Axis with analog manipulated variable output and actual value acquisition through incremental encoder (similar to standard servo axis). The analog output signal is always positive; the directional information is output through the digital output. Here, for positive manipulated variable output, an environment variable SIGNOUTPOS? can be defined, and for negative manipulated variable output, an environment variable SIGNOUTNEG?, with the attached „?“ representing the axis index. In the SIGNOUT... value, the digital output is specified in a bit-coded form (output 1 = 1, output 2 = 2, output 3 = 4, ... output 8 = 128, etc.). Here, also none or more outputs can be specified.

Example:
2nd axis, positive manipulated value = output 4, negative manipulated value display is not required

```
set MT1 19
set  SIGNOUTPOS1  8
set  SIGNOUTNEG1  0
```

In this case, SIGNOUTNEG1 can also remain undefined.

3.12.1.16 CI / ANALOG and CD / ANALOG motor types (20+21)

Motor type with analog manipulated variable output and actual value acquisition through CI (MT 20) or CD (MT 21) variable. By default, each axis is assigned the content of the common variable using the index of the corresponding axis. To assign a different common variable to an axis, the environment variable FBCH? (Chapter 0) can be set.

3.12.1.17 STEPPER_ENDAT2_2 motor type (22)

3.12.1.18 ETM motor type (23)

Application-specific motor type with pulse duration measurement for actual value acquisition

3.12.1.19 ANA_SIGN_SSI motor type (24)

Axis with analog manipulated variable output and actual value acquisition through SSI absolute encoder (similar to SSI type 2 and ANA_SIGN type 19). The analog output signal is always positive; the directional information is output through the digital output. Here, for positive manipulated variable output, an environment variable SIGNOUTPOS? can be defined, and for negative manipulated variable output, an environment variable SIGNOUTNEG?, with the attached „?“ representing the axis index. In the SIGNOUT... value, the digital output is specified in a bit-coded form (output 1 = 1, output 2 = 2, output 3 = 4, ... output 8 = 128, etc.). Here, also none or more outputs can be specified.

Example:
2nd axis, positive manipulated value = output 4, negative manipulated value display is not required
set MT1 24
set SIGNOUTPOS1 8
set SIGNOUTNEG1 0

In this case, SIGNOUTNEG1 can also remain undefined.

3.12.1.20 ANA_SIGN_ENDAT2_2 motor type (25)

Axis with analog manipulated variable output and actual value acquisition through ENDAT 2.2 absolute encoder (similar to ENDAT 2.2 type 16 and ANA_SIGN type 19). The analog output signal is always positive; the directional information is output through the digital output. Here, for positive manipulated variable output, an environment variable SIGNOUTPOS? can be defined, and for negative manipulated variable output, an environment variable SIGNOUTNEG?, with the attached „?“ representing the axis index. In the SIGNOUT... value, the digital output is specified in a bit-coded form (output 1 = 1, output 2 = 2, output 3 = 4, ... output 8 = 128, etc.). Here, also none or more outputs can be specified.

Example:
2nd axis, positive manipulated value = output 4, negative manipulated value display is not required
set MT1 24
set SIGNOUTPOS1 8
set SIGNOUTNEG1 0

In this case, SIGNOUTNEG1 can also remain undefined.

3.12.1.21 SSI_PULSE motor type (26)

With this motor type, a position control is carried out in the same way as with default servo systems, but with a step direction output. The Motor-Type {mt} variable in mcfg must be set to SERVO. The default frequency range of the pulse output is +/-2 MHz. The frequency range can be limited using the variables {mcpmax} and {mcpmin}. The unit of these variables is 200 kHz. If a setpoint value jump (OL Response) is output, the output frequency is also indicated in the unit of 200 kHz.
The position controller must be set according to the same criteria as a speed-controlled system.
Notice: An axis can only be configured for this motor type if in RWMOS, the resources SSI absolute encoder and pulse output are available for the corresponding axis.
This motor type is only available from RWMOS V2.5.3.132.

3.12.2 The environment variable NumberAxis

This environment variable enables to set the axis number. The default value is 3. Normally, this value is set for operation and is only to be modified with values which are supported by the hardware and software configuration of the control.

3.12.3 The environment variable SampleTime

Through this environment variable the scan time (control cycle time and interpolation cycle time) of the control is set in microseconds. The values are between 100 and 5000. The default value is 1280. This value can only be modified with values which can be reached by the hardware and software configuration of the control. Realistic values begin from approximately 300. Please note that by modifying the scan time, the filter parameters must generally be adapted as well, especially the pre-control coefficients kfcv and kfca.
If this setting is required for the application, it should also be checked in the application program to avoid faults when the system is reproduced or serviced. This setting is a device option, which is stored in the flash memory of the PCI board. The sampling time can be checked with the DLL function rdSampleTime(). The value of environment variables can be read with the DLL function getEnvStr ().

3.12.4 The environment variable SZTSK?

Herewith the program memory of the task environment can be modified in bytes according to the task. The ? stands for the task number (0..3) and must be set. The default value is 100.000 bytes.

Example:

```
set SZTSK3 1000000
```

3.12.5 APCI-8001: Configuration of the analog input voltage ranges

With the environment variable **MAX1270CH?**, the input voltage range of the available analog inputs can be configured for each channel. By setting this value, an analog input channel is activated simultaneously.
The ? stands for the index of the analog channel (0..7).

Value	Voltage range
5VU	5 V unipolar (0 V..+5 V)
5VB	5 V bipolar (-5 V..+5 V)
10VU	10 V unipolar (0 V..+10 V)
10VB	10 V bipolar (-10 V..+10 V)

Example:

```
set MAX1270CH0 5VU
```

Further information about the use of analog inputs is to be found in the Options Manual.

3.12.6 APCI-8008: Configuration of the analog input voltage ranges

With the environment variable **AD7606RNG,** the input voltage range of the available analog inputs can be configured. Here, the configuration cannot be set individually for each channel but for all analog inputs together. Unlike the APCI-8001, there is no longer a unipolar operating mode with the APCI-8008. Instead, the analog inputs have a 16-bit resolution.

Value	Voltage range
5VB	5 V bipolar (-5 V..+5 V)
10VB	10 V bipolar (-10 V..+10 V)

Examplel:

```
set AD7606RNG 5VB
```

Further information about the use of analog inputs is to be found in the Options Manual.

With the APCI-8008, the environment variable **AD7606OS** offers another configuration possibility:
The analog inputs have the option for hardware oversampling, i.e. the input voltage is read in several times and an average value is computed automatically. The following values can be programmed:

Value	Oversampling factor	Conversion time / µs
0	No oversampling	**5**
1	2	**10**
2	4	**20**
3	8	**40**
4	16	**80**
5	32	**160**
6	64	**320**

If other values are programmed, oversampling will be deactivated. The default value is 0.

Example:

```
set AD7606OS 6
```

3.12.7 The environment variable FBCH?

Feedback channel: An available analog input can be allocated to an analog actual value channel. This variable is only significant for axes with motor types “ANALOG PWM” (Section 3.12.1.4), “ANALOG / ANALOG” (Section 3.12.1.6), “CI_ANALOG” and “CD_ANALOG” (Section 3.12.1.16).
FBCHx y: x is the index of the axis to which the channel is assigned; y is the index of the analog input (0...7) or of the common variable (0...999) that is to be read as a feedback channel. If this variable is not set, each axis channel 0..7 is assigned the respective analog input channel 0..7.

Example:

```
set FBCH0 0
```

3.13 Special features for system parameters for servo and stepper motor axes

Please note the system parameters listed below, which must be set accordingly in the *system.dat* system file using the *mcfg.exe* application.

Operating mode	Parameter/page	Value/meaning
Servo motor	Motor type {mt} / motion parameters	Servo/operating mode selection
	Encoder slits or step pulses / motion parameters	Slits/strokes (electronic quadrupling is considered)
Stepper motor	Motor type {mt} / motion parameters	Stepper/operating mode selection
	Encoder slits or step pulses / motion parameters	Pulses/steps (no electronic quadrupling)
	Filter parameter {kp} / motor-specific parameters	0.04 (is set by the system))/ any other value will lead to instable control behaviour Notice: As of *mcfg.exe* version 2.5.0.45, {kp} can no longer be edited for stepper motors! You require *rwmos.elf* firmware version 2.5.0.4 or higher.

What next?

If you have reached this point without any errors, the xPCI-800x controller has been successfully set up and is ready for use. To complete the installation, you can now execute the following steps in the *mcfg.exe* application:

- [File] [Dialog Functions][Show Axis Status] to display the current position actual and setpoint values as well as the axis status information
- [File] [Dialog Functions][Show Digital Inputs / Status] to display the digital inputs, axis status and interface status information
- [File] [Dialog Functions][Edit Digital Outputs] to set or reset the digital outputs
- [File] [Motion Tools] for manual procedure of the drive axes. In doing this, note that the [Close Loop] button must also be activated for a stepper motor axis, before the axis can be manually operate using the [Jog Start], [Jog Stop] or [Jog Back] buttons. This is necessary, as stepper motor axes are also managed via an internal control algorithm.

If you are happy with your settings, you can now start to configure the PC application program. For help with this, you will find program libraries and example programs in the Drivers, libraries and examples sub-directory on the xPCI-800x TOOLSET CD.

3.14 Additional installation information for Windows NT

If problems occur when allocating physical storage under Windows NT, these are displayed on screen through explicit xPCI-800x TOOLSET software error messages. In this case, a value must be changed in the registration database of your PC.

To do this, proceed as follows:

- Start the regedit Window utility program as the administrator, e.g., via Start, Run, regedit.
- Change the HKEY_LOCAL_MACHINES\System\CurrentControlSet\Control\Session Manager\Memory Management\SystemPages key, which should usually have the value 0, to the value 10000 hex or higher.
- Reboot the PC.

3.15 Updating the xPCI-800x Flash firmware (PMON)

If it is necessary to update the Flash firmware on the xPCI-800x controller:

- Start the fwsetup application.
- Open the [Pmon Download] page.
- Select the pmon.elf file from the Firmware and system.dat files\Pmon sub-directory.
- After you have made your selection, the xPCI-800x controller’s flash will be deleted and then the corresponding data will be written from the PMON.ELF file.
- Please note any error messages during programming.
- Select the [Monitor] page.
- Press the [Soft Reset] button.

The monitor should immediately display new screen messages.

3.16 Trouble-shooting

E-mail: info@addi-data.com

Phone: +49 7229 1847-0

4 How to install the motion control board in the PC

You need:

- Motion control board APCI-8001 or APCI-8008
- TOOLSET CD

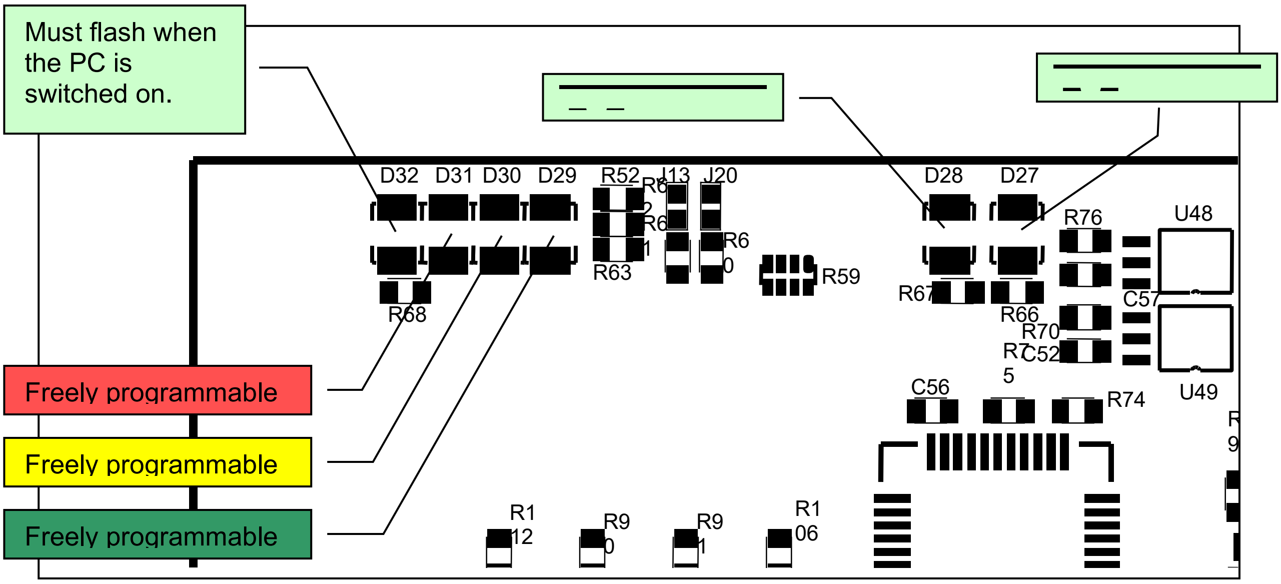
4.1 Installing the xPCI-800x controller

- Switch the PC off
- Electrostatically discharge yourself
- Insert the board vertically from above into the selected PCI slot
  (Ensure that you do not touch the gold contacts, if necessary the contacts can first be cleaned with alcohol)
- Switch the PC on and start Windows.
  Supply voltages of 3.3 V and 5 V must be available at the PCI bus of the target system (PC motherboard). This can be identified using the diagnosis LEDs listed below, after switching on the PC. All diagnosis LEDs are on the back (soldered side) of the xPCI-800x controller, on the top edge of the board. These are small SMD (surface-mounted) light-emitting diodes. See also the sub-component insertion diagram on the next page.

Device	LED	Function
APCI-8001	D27	PC supply voltage 5 V (1st LED seen from the board holder). Must be lit up.
	D28	PC supply voltage 3.3 V (2nd LED seen from the board holder). Must be lit up.
	D32	Flashes, showing that the monitor program is running (1st LED on the side facing away from the board holder). Must flash when the PC is switched on.
APCI-8008	D50	PC supply voltage 5 V (1st LED seen from the board holder). Must be lit up.
	D51	Internal supply voltage 3.3 V (2nd LED seen from the board holder). Must be lit up.
	D57	Flashes, showing that the monitor program is running (1st LED on the side facing away from the board holder). Must flash when the PC is switched on.

For most PC systems, the results of the PCI plug and play bios are displayed during the boot process. An xPCI-800x controller with the following identity (APCI-8001) should be listed there:
Vendor ID: 11AB (hex) and Device ID: 4611 (hex)
Windows 95, 98, Me and Windows 2000 automatically recognises the xPCI-800x board.
For installation, you now require the xPCI-800x TOOLSET CD. There you will find the necessary INF file to install the board, under the *inf* sub-directory. You must register the xPCI-800x controller as a multi-function board if necessary.

Sub-component insertion diagram (soldered side) of the APCI-8001



4.1.1 Installation under W98

If the board is displayed in the device manager with an exclamation mark:
- Select it with the right mouse key and select properties : Driver – Update driver. You also can delete the wrong device in the device manager and then select “Update”. Then the device is detected as a “PCI Memory Controller”. Then do the following steps:

- Select: Show a driver list in a specific directory…
- Select multifunction board
- Select the floppy disc
- Select mcug3.inf (e.g. on CD)
- Select the correct product (e.g. APCI-8001)
- Under Win98 the following message is displayed: “The driver is not written for the selected hardware…“ etc.
  This happens because Win98 overwrites the Sub-Vendor ID of the board and therefore does not recognize it. Confirm this message with OK.
- Then confirm it with Yes and install it.

Now the device must be correctly registered in the device manager under “Multifunction boards”.

Please note that for Windows 98 the Miniport driver V7.01b (CD sub-directory V9x) must be used.
We recommend you to use a Windows version later than Windows 9x.

5 Configuring and wiring the motion control board

5.1 Installation, commissioning and replacement

If the module is recommissioned or replaced, various system data must be saved from the *system.dat* system file to the motion control board. This is done using the *mcfg.exe* utility programmed in the [Save Changes] menu. If the saved information is not conform with the information saved in the *system.dat* system file, the *cef* flag is set.

5.2 Environment

The xPCI-800x board was specifically designed for industrial use. All inputs are available in potential-free form. The output signals are electronically isolated from the logic supply and have a shared earth reference potential. This means that incidents from the peripheral electronics are almost completely suppressed.
As the xPCI-800x board is fitted with a microprocessor module, you should avoid installing it in an environment subject to severe electromagnetic interference. Otherwise, this could lead to an uncontrolled process behaviour by the microcontroller. In this case, you must expect the watchdog logic of the xPCI-800x module to be triggered and cause a hardware reset.

Hardware interfaces, connection assignments
Depending on the level of completion, the peripheral electronics are connected to the APCI-8001 / APCI-8008 via a 50-pin SUB-D connector (X1). For rapid and simple wiring, an ST8001 connection cable and a PX8001 screw terminal panel can also be used optionally.

5.2.1 Connector X1: 50-pin SUB-D male connector APCI-8001 / APCI-8008

Pin	Name	Group
1	SERVO1 / PULSE1+	Setpoint value 1/stepper 1
2	AGND1 / PULSE1-	Setpoint value 1/stepper 1
3	CHA1+ / CLKSSI1+ / ENDATCLK1+	Actual value 1
4	CHA1- / CLKSSI1- / ENDATCLK1-	Actual value 1
5	CHB1+ / DATSSI1+ / ENDAT_Data1+	Actual value 1
6	CHB1- / DATSSI1- / ENDAT_Data1-	Actual value 1
7	NDX1+ / SIGN1+	Actual value 1/stepper 1
8	NDX1- / SIGN1-	Actual value 1/stepper 1
9	I1	Digital inputs 1-8 (24V) Assignment to axis channel 1, 2 and 3
10	I2	
11	I3	
12	I4	
13	I5	
14	I6	
15	I7	
16	I8	
17	+24V	Power supply for the digital outputs 24V, if digital outputs are being used, this voltage must be supplied externally.
18	SERVO2 / PULSE2+	Setpoint value 2/stepper 2
19	AGND2 / PULSE2-	Setpoint value 2/stepper 2
20	CHA2+ / CLKSSI2+ / ENDATCLK2+	Actual value 2
21	CHA2- / CLKSSI2- / ENDATCLK2-	Actual value 2
22	CHB2+ / DATSSI2+ / ENDAT_Data2+	Actual value 2
23	CHB2- / DATSSI2- / ENDAT_Data2-	Actual value 2
24	NDX2+ / SIGN2+	Actual value 2/stepper 2
25	NDX2- / SIGN2-	Actual value 2/stepper 2
26	O1	Digital outputs 1..8 (24V) Assignment to axis channel 1, 2 and 3
27	O2	
28	O3	
29	O4	
30	O5	
31	O6	
32	O7	
33	O8	
34	SERVO3 / PULSE3+	Setpoint value 3/stepper 3
35	AGND3 / PULSE3-	Setpoint value 3/stepper 3
36	CHA3+ / CLKSSI3+ / ENDATCLK3+	Actual value 3
37	CHA3- / CLKSSI3- / ENDATCLK3-	Actual value 3
38	CHB3+ / DATSSI3+ / ENDAT_Data3+	Actual value 3
39	CHB3- / DATSSI3- / ENDAT_Data3-	Actual value 3
40	NDX3+ / SIGN3+	Actual value 3/stepper 3
41	NDX3- / SIGN3-	Actual value 3/stepper 3

Pin	Name	Group
42	I9	Digital inputs 9-16 (24V)
43	I10	Assignment to axis channel 1, 2 and 3
44	I11	
45	I12	
46	I13	
47	I14	
48	I15	Faster latch input axis channel 1
49	I16	Faster latch input axis channel 2 Faster latch input axis channel 3
50	GND-D	Reference potential for all signal sources. These include digital inputs and outputs and the transmitter actual value. GND-D must be connected with the earth potential of the external device electronics.

5.2.2 Counting for the 50-pin SUB-D male connector X1

▪ 34		▪ 1	Top
	▪ 18		
▪ 35		▪ 2	
	▪ 19		
▪ 36		▪ 3	
	▪ 20		
▪ 37		▪ 4	
	▪ 21		
▪ 38		▪ 5	
	▪ 22		
▪ 39		▪ 6	
	▪ 23		
▪ 40		▪ 7	
	▪ 24		
▪ 41		▪ 8	
	▪ 25		
▪ 42		▪ 9	
	▪ 26		
▪ 43		▪ 10	
	▪ 27		
▪ 44		▪ 11	
	▪ 28		
▪ 45		▪ 12	
	▪ 29		
▪ 46		▪ 13	
	▪ 30		
▪ 47		▪ 14	
	▪ 31		
▪ 48		▪ 15	
	▪ 32		
▪ 49		▪ 16	
	▪ 33		
▪ 50		▪ 17	Bottom / PC bus connector

5.2.3 Connector X1: 78-pin SUB-D female connector APCI-8008-STP-EVAI

Row 1 Pins 1-20		**Row 2 Pins 21–39**		**Row 3 Pins 40–59**		**Row 4 Pins 60–78**	
1	Puls+ CH1	21	Puls– CH1	40	Sign+ CH1	60	Sign– CH1
2	CHA+ CH1	22	CHA– CH1	41	CHB+ CH1	61	CHB– CH1
3	Puls+ CH2	23	Puls– CH2	42	Sign+ CH2	62	Sign– CH2
4	CHA+ CH2	24	CHA– CH2	43	CHB+ CH2	63	CHB– CH2
5	Puls+ CH3	25	Puls– CH3	44	Sign+ CH3	64	Sign– CH3
6	CHA+ CH3	26	CHA– CH3	45	CHB+ CH3	65	CHB– CH3
7	Puls+ CH4	27	Puls– CH4	46	Sign+ CH4	66	Sign– CH4
8	CHA+ CH4	28	CHA– CH4	47	CHB+ CH4	67	CHB– CH4
9	Puls+ CH5	29	Puls– CH5	48	Sign+ CH5	68	Sign– CH5
10	CHA+ CH5	30	CHA– CH5	49	CHB+ CH5	69	CHB– CH5
11	Puls+ CH6	31	Puls– CH6	50	Sign+ CH6	70	Sign– CH6
12	CHA+ CH6	32	CHA– CH6	51	CHB+ CH6	71	CHB– CH6
13	D-Inp 01	33	D-Inp 02	52	D-Inp 03	72	D-Inp 04
14	D-Inp 05	34	D-Inp 06	53	D-Inp 07	73	D-Inp 08
15	D-Inp 09	35	D-Inp 10	54	D-Inp 11	74	D-Inp 12
16	D-Inp 13	36	D-Inp 14	55	D-Inp 15	75	D-Inp 16
17	(Out 01)	37	(Out 02)	56	CNC-Ready+ (Out 03)	76	CNC-Ready- (Out 04)
18	AIN6– (Out 05)	38	AIN6+ (Out 06)	57	AIN5– (res. Out 07)	77	AIN5+ (res. Out 08)
19	AIN2–	39	AIN2+	58	AIN1–	78	AIN1+
20	GND			59	res. (+24V)		

5.2.4 Counting for the 78-pin SUB-D female connector X1

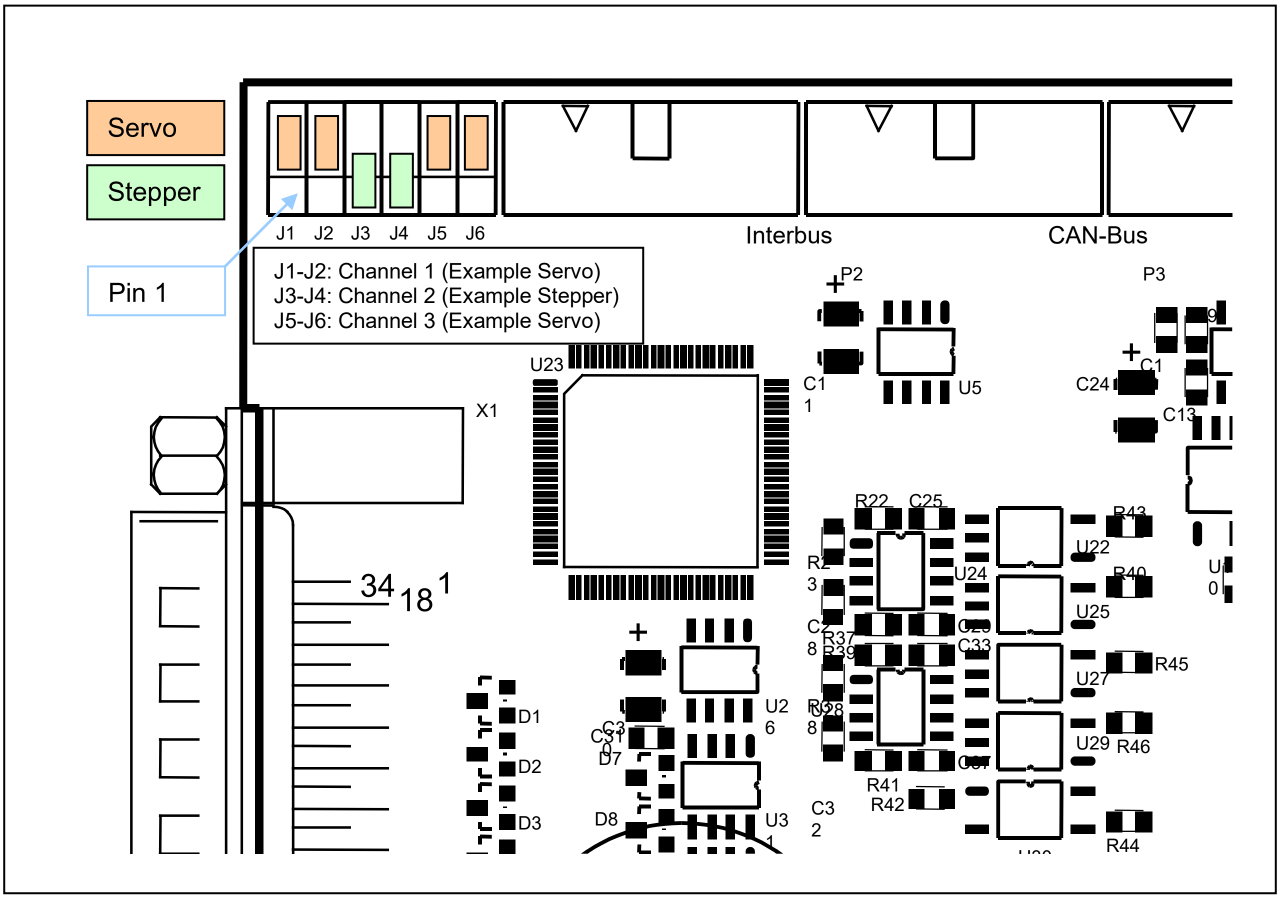
	▪ 59		▪ 20	Top
▪ 78		▪ 39		
	▪ 58		▪ 19	
▪ 77		▪ 38		
	▪ 57		▪ 18	
▪ 76		▪ 37		
	▪ 56		▪ 17	
▪ 75		▪ 36		
	▪ 55		▪ 16	
▪ 74		▪ 35		
	▪ 54		▪ 15	
▪ 73		▪ 34		
	▪ 53		▪ 14	
▪ 72		▪ 33		
	▪ 52		▪ 13	
▪ 71		▪ 32		
	▪ 51		▪ 12	
▪ 70		▪ 31		
	▪ 50		▪ 11	
▪ 69		▪ 30		
	▪ 49		▪ 10	
▪ 68		▪ 29		
	▪ 48		▪ 9	
▪ 67		▪ 28		
	▪ 47		▪ 8	
▪ 66		▪ 27		
	▪ 46		▪ 7	
▪ 65		▪ 26		
	▪ 45		▪ 6	
▪ 64		▪ 25		
	▪ 44		▪ 5	
▪ 63		▪ 24		
	▪ 43		▪ 4	
▪ 62		▪ 23		
	▪ 42		▪ 3	
▪ 61		▪ 22		
	▪ 41		▪ 2	
▪ 60		▪ 21		
	▪ 40		▪ 1	Bottom / interlocking

5.2.5 Setpoint value channels

Each axis channel of the APCI-8001 or APCI-8008 can be operated as a servo or stepper motor channel. The following configuration table, or the following component insertion diagram, can be used to determine the jumper configuration.
In the factory, the devices are supplied for servo motor axes.
The software planning and selection of the required motor system must be adapted additionally using the *mcfg.exe* TOOLSET program.

Channel	Jumper	Position	Mode
1	J1, J2	1-2	Stepper
		2-3	Servo
2	J3, J4	1-2	Stepper
		2-3	Servo
3	J5, J6	1-2	Stepper
		2-3	Servo

Component mounting diagram (components side) of the APCI-8001 (applies to the APCI-8008 to a limited extent)



5.2.5.1 Setpoint value channel for servo motor axes APCI-8001 / APCI-8008

The analog output signal is used to control a power amplifier, which is activated as a speed or moment controller (current amplifier). The offset of this setpoint value channel is stored in the factory in the non-volatile flash memory of the APCI-8001 and is taken into consideration by the software during output. The analog value output is only supported for *SERVO* planned axes.

5.2.5.1.1 Pin assignment for connector X1, axis channel 1

Pin	Name	Group	Description
1	SERVO1	Setpoint 1	Analog output signal 1 for controlling a power amplifier (+/-10 V, 5 mA). This signal is electrically isolated from the APCI-8001 / APCI-8008 and has reference potential AGND1.
2	AGND1	Setpoint 1	Reference potential for SERVO1. This potential is electrically isolated from the APCI-8001 / APCI-8008 system electronics.

Notice: Jumpers J1 and J2 must be bridged in position 2-3, so that the signals listed in the table are available at connector X1.

5.2.5.1.2 Pin assignment for connector X1, axis channel 2

Pin	Name	Group	Description
18	SERVO2	Setpoint 2	Analog output signal 2 for controlling a power amplifier (+/-10 V, 5 mA). This signal is electrically isolated from the APCI-8001 / APCI-8008 and has reference potential AGND2.
19	AGND2	Setpoint 2	Reference potential for SERVO2. This potential is electrically isolated from the APCI-8001 / APCI-8008 system electronics.

Notice: Jumpers J3 and J4 must be bridged in position 2-3, so that the signals listed in the table are available at connector X1.

5.2.5.1.3 Pin assignment for connector X1, axis channel 3

Pin	Name	Group	Description
34	SERVO3	Setpoint 3	Analog output signal 3 for controlling a power amplifier (+/-10 V, 5 mA). This signal is electrically isolated from the APCI-8001 / APCI-8008 and has reference potential AGND3.
35	AGND3	Setpoint 3	Reference potential for SERVO3. This potential is electrically isolated from the APCI-8001 / APCI-8008 system electronics.

Notice: Jumpers J5 and J6 must be bridged in position 2-3, so that the signals listed in the table are available at connector X1.

5.2.5.2 Setpoint value channel for stepper motor axes APCI-8008 / APCI-8008

There are four output signals to control a stepper motor power module. These comprise a pulse signal, a directional signal, and their inverted signals according to EIA standard RS422. All outputs deliver a typical output current of -60 mA (max. -150 mA). The maximum pulse frequency of the stepper signals is 10 MHz.
Notice: The positive edge of the PULSx+ stepper signal or the negative edge of the PULSx- stepper signal are crucial for the correct number of steps to be executed.

5.2.5.2.1 Pin assignment for connector X1, axis channel 1

Pin	Name	Group	Description
1	PULSE1+	Stepper 1	Pulse signal
2	PULSE1-	Stepper 1	Inverted pulse signal
7	SIGN1+	Stepper 1	Directional signal
8	SIGN1-	Stepper 1	Inverted directional signal

Notice: Jumpers J1 and J2 must be bridged in position 1-2, in order for the above-mentioned signals to be available at connector X1.

5.2.5.2.2 Pin assignment for connector X1, axis channel 2

Pin	Name	Group	Description
18	PULSE2+	Stepper 2	Pulse signal
19	PULSE2-	Stepper 2	Inverted pulse signal
24	SIGN2+	Stepper 2	Directional signal
25	SIGN2-	Stepper 2	Inverted directional signal

Notice: Jumpers J3 and J4 must be bridged in position 1-2, in order for the above-mentioned signals to be available at connector X1.

5.2.5.2.3 Pin assignment for connector X1, axis channel 3

Pin	Name	Group	Description
34	PULSE3+	Stepper 3	Pulse signal
35	PULSE3-	Stepper 3	Inverted pulse signal
40	SIGN3+	Stepper 3	Directional signal
41	SIGN3-	Stepper 3	Inverted directional signal

Notice: Jumpers J5 and J6 must be bridged in position 1-2, in order for the above-mentioned signals to be available at connector X1.

5.2.5.3 Analog outputs with the APCI-8008

With the APCI-8008, a total of 4 analog outputs with 16-bit resolution each are available.
The first 3 of these outputs are used as manipulated value outputs for axis channels 1-3 (see previous chapters). However, all 4 analog outputs are also led to connector X2. With servo systems, the 4th output channel can be used for other purposes in this case.
If several or all axis channels are used as stepper channels, all analog outputs can be used for other purposes. Via the “Universal Object Interface”, these outputs are assigned a value by means of the resource #83. Using a SUB-D adapter, the signals can be led to a 9-pin SUB-D male or female connector. This adapter must be ordered separately. The AGND signals are all connected with one other.
Notice: If the analog outputs of the channels are used for axis control, they should not be connected to the connection adapter.

Pin (SUB-D)	Name	Function	Pin at X2 (FB)
1	AOUT0	Analog output of the 1st axis channel	1
2	AOUT1	Analog output of the 2nd axis channel	3
3	AOUT2	Analog output of the 3rd axis channel	5
4	AOUT3	4th analog output	7
5		not assigned	9
6	AGND0	Reference potential for analog output	2

Pin (SUB-D)	**Name**	**Function**	**Pin at X2 (FB)**
7	AGND1	Reference potential for analog output	4
8	AGND2	Reference potential for analog output	6
9	AGND3	Reference potential for analog output	8

5.2.6 Pulse acquisition channels

The APCI-8001 / APCI-8008 is fitted with up to three pulse acquisition channels, to which various encoder types, such as linear scales or incremental or absolute encoders can be connected. Two 90° phase-shifted quadrature signals are processed as input signals, with a maximum pulse frequency of 2.0 MHz (optionally 5 MHz) and TTL sensors. A zero track (index signal) can also be evaluated. The signal levels acquired by the encoders are electronically quadrupled and processed internally as floating-point numbers with double accuracy. This means that the value range for the traverse path is virtually unrestricted.
The pulse acquisition of the APCI-8008 is provided with effective line break monitoring.

5.2.6.1 SSI absolute encoder

If SSI absolute encoders are to be used for position feedback, corresponding environment variables must be set in fwsetup.exe for the respective axis channels (see section on “Environment variables for control system hardware”.

Table: Environment variables for SSI absolute encoders

Variable	Value	Comment
MT?	2 4	Servo motor with SSI absolute encoder Stepper motor with SSI absolute encoder ? is the number of the respective axis channel (starting from 0)
SSIF?	X	X = Numerical value for SSI frequency in Hz Permitted values: 100,000 to 4,000,000 Default value: 500000 Notice: With bigger cable lengths, the SSI frequency must be reduced in order to compensate the signal runtimes. ? is the number of the respective axis channel (starting from 0)
SSIP?	X	X = Numerical value for the number of SSI bits Permitted values: 2 to 30 Default value: 26 (25 bits + 1 extra bit) ? is the number of the respective axis channel (starting from 0)
SSIBIN?	1	The binary code at the SSI data word is selected with a nonzero value. Default: Gray code. ? is the number of the respective axis channel (starting from 0)
SSIPCK SIZE?	1	(from RWMOS.ELF V2.5.3.130) With this variable, the word length of the encoder is indicated in bits. Only if this value is correctly indicated, an extension of the traversing range via the measurement range of the encoder and the addition of a sign to the position value is possible. The default value is 24 bits. ? is the number of the respective axis channel (starting from 0)
SSIOUT OF RANGE?	1	(from RWMOS.ELF V2.5.3.130) With this variable, an error bit can be identified and displayed. This bit is indicated in the system variable DIGI in the EnkoderError bit. Moreover, in the IFS register, the bef bit is set. Thus, also an event can be triggered in the SAP programming environment.

Furthermore, for the application of SSI absolute encoders, an operating system version (RWMOS.ELF) with the option “option SSI” must be used. The available RWMOS.ELF options can be verified in the “fwsetup” program, on the “Monitor” tab when the control unit is booted.

Example:

Info: running » APCI-8001 [option1k5, optionPCI, optionRESOURCE]

For the connection of the SSI interface, the pins with the designation CLKSSI for the SSI clock signals and DATSSI for the SSI data lines have to be used.

5.2.6.2 Endat absolute encoder

If Endat absolute encoders are to be used for position feedback, corresponding environment variables must be set in fwsetup.exe for the respective axis channels (see section on “Environment variables for control system hardware”.

Table: Environment variables for Endat absolute encoders

Variable	Value	Comment
MT?	16	Servo motor with Endat 2.2 absolute encoder ? is the number of the respective axis channel (starting from 0)
ENDATF?	X	X = Numerical value für Endat frequency in Hz Permitted values: 100,000 to 2,000,000 Default value: 500000 Notice: With bigger cable lengths, the Endat clock frequency must be reduced in order to compensate the signal runtimes. ? is the number of the respective axis channel (starting from 0)

Furthermore, for the application of Endat absolute encoders, an operating system version (RWMOS.ELF) with the option “optionEndat” must be used. The available RWMOS.ELF options can be verified in the “fwsetup” program, on the “Monitor” tab when the control unit is booted.

Example:

Info: running » MCU-3000 / APCI-8001 [option1k5, optionPCI, optionRESOURCE, optionENDAT]

For the connection of the Endat interface, the pins with the designation ENDATCLK for the Endat clock signals and ENDAT_Data for the Endat data lines have to be used.

5.2.6.3 Incremental encoders with inverted signals (symmetrical circuitry)

Incremental encoders with symmetrical outputs are particularly suitable for industrial use and are preferred, as the output signals are available with inverted and non-inverted signal sensors for all tracks. This enables pulses to be acquired reliably, even in environments that are subject to severe electromagnetic interference. The evaluation electronics on the APCI-8001 / APCI-8008 are based on the RS422 standard and form a signal difference between the inverted and non-inverted input signals. Interference that is linked with transmission lines can thus effectively be suppressed.

Important: In the factory, the APCI-8001 / APCI-8008 is delivered for incremental encoders with symmetrical outputs, but can be configured by the user for asymmetrical encoders (see next table).

5.2.6.4 Incremental encoders without inverted signals (asymmetrical circuitry)

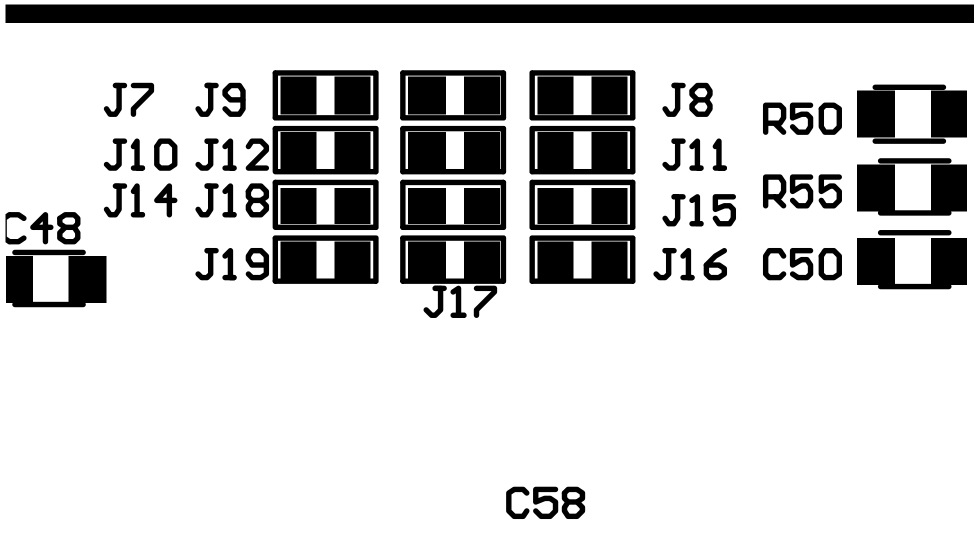
It is also possible to process incremental encoders without inverted pulse trains. However, these should only be used in environments that are not subject to severe electromagnetic interference, e.g. in laboratory applications. Please also ensure that the encoder cable is only a few metres long, especially for high pulse frequencies.

Table 5-1: Configuration of the incremental encoders for symmetrical and asymmetrical operating modes

Axis channel	Signal source	Solder jumper	Asymmetrical	Symmetrical
1	CHA1-	J8 (8000) J10 (8008)	Bridged Do not connect pin 4 / X1	Unbridged Connect pin 4 / X1
	CHB1-	J11 (8000) J9 (8008)	Bridged Do not connect pin 6 / X1	Unbridged Connect pin 6 / X1
	NDX1-	J15 (8000) J8 (8008)	Bridged Do not connect pin 8 / X1	Unbridged Connect pin 8 / X1
2	CHA2-	J9 (8000) J13 (8008)	Bridged Do not connect pin 21 / X1	Unbridged Connect pin 21 / X1
	CHB2-	J12 (8000) J12 (8008)	Bridged Do not connect pin 23 / X1	Unbridged Connect pin 23 / X1
	NDX2-	J18 (8000) J11 (8008)	Bridged Do not connect pin 25 / X1	Unbridged Connect pin 25 / X1
3	CHA3-	J7 (8000) J17 (8008)	Bridged Do not connect pin 37 / X1	Unbridged Connect pin 37 / X1
	CHB3-	J10 (8000) J16 (8008)	Bridged Do not connect pin 39 / X1	Unbridged Connect pin 39 / X1
	NDX3-	J14 (8000) J14 (8008)	Bridged Do not connect pin 41 / X1	Unbridged Connect pin 41 / X1

Notice: The solder jumpers listed in the table are located on the top right-hand corner of the solder side of the APCI-8001 / APCI-8008 board.

Component mounting diagram (soldered side) of the APCI-8001



5.2.6.5 Optical decoupling of the pulse acquisition channels

All pulse acquisition channels of the APCI-8001 / APCI-8008 are optically decoupled. This is advantageous particularly in an environment that is subject to severe electromagnetic interference.

5.2.6.6 Pin assignment for the pulse acquisition channels with incremental encoders

5.2.6.6.1 Pin assignment X1, channel 1

Pin	Name	Function
3	CHA1+	Incremental signal (TTL square-wave pulse trains) track A
4	CHA1-	Inverted incremental signal track A
5	CHB1+	Incremental signal track B with 90° electrical phase shift to track A
6	CHB1-	Inverted incremental signal track B
7	NDX1+	Reference signal track 0
8	NDX1-	Inverted reference signal track 0

5.2.6.6.2 Pin assignment X1, channel 2

Pin	Name	Function
20	CHA2+	Incremental signal (TTL square-wave pulse trains) track A
21	CHA2-	Inverted incremental signal track A
22	CHB2+	Incremental signal track B with 90° electrical phase shift to track A
23	CHB2-	Inverted incremental signal track B
24	NDX2+	Reference signal track 0
25	NDX2-	Inverted reference signal track 0

5.2.6.6.3 Pin assignment X1, channel 3

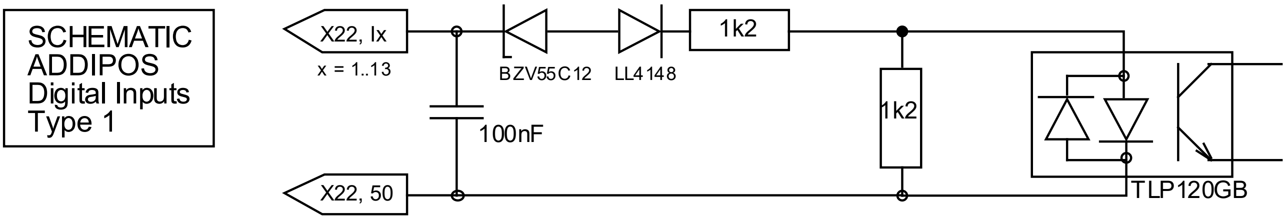
Pin	Name	Function
36	CHA3+	Incremental signal (TTL square-wave pulse trains) track A
37	CHA3-	Inverted incremental signal track A
38	CHB3+	Incremental signal track B with 90° electrical phase shift to track A
39	CHB3-	Inverted incremental signal track B
40	NDX3+	Reference signal track 0
41	NDX3-	Inverted reference signal track 0

5.2.7 Pin assignment connector X1, Digital inputs (APCI-8001 / APCI-8008)

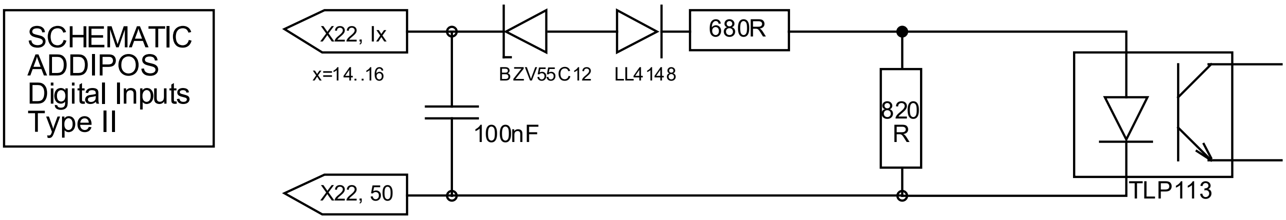
The basic wiring diagram of the digital inputs I1..I13 listed below are printed in [section 5.2.7.1] and inputs I14…I16 are printed in [section 5.2.7.2].

Pin	Name	Function
9	I1	Digital input 1
10	I2	Digital input 2
11	I3	Digital input 3
12	I4	Digital input 4
13	I5	Digital input 5
14	I6	Digital input 6
15	I7	Digital input 7
16	I8	Digital input 8
42	I9	Digital input 9
43	I10	Digital input 10
44	I11	Digital input 11
45	I12	Digital input 12
46	I13	Digital input 13
47	I14	Digital input 14 and faster hardware latch input to save the actual position of axis channel 1
48	I15	Digital input 15 and faster hardware latch input to save the actual position of axis channel 2
49	I16	Digital input 16 and faster hardware latch input to save the actual position of axis channel 3

5.2.7.1 Basic wiring diagram of xPCI-800x digital inputs I1 … I13



5.2.7.2 Basic wiring diagram of xPCI-800x digital inputs I14 … I16

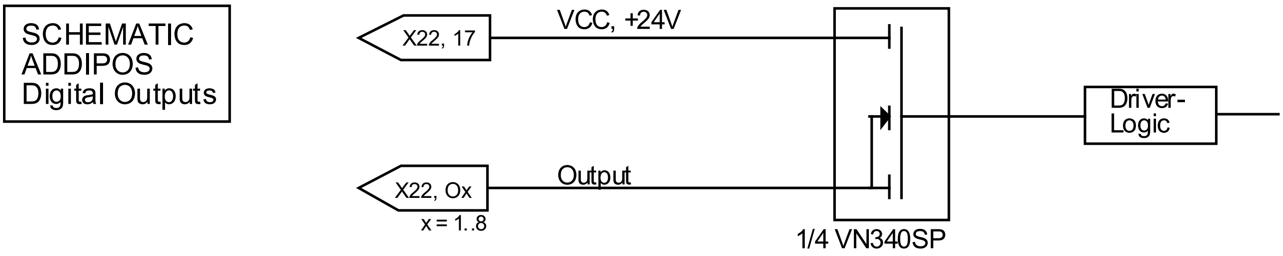


5.2.8 Pin assignment connector X1, digital outputs (APCI-8001 / APCI-8008)

The basic wiring diagrams for the digital outputs O1..O8 listed below are printed in [section 5.2.8.1].

Pin	Name	Function
26	O1	Digital output 1
27	O2	Digital output 2
28	O3	Digital output 3
29	O4	Digital output 4
30	O5	Digital output 5
31	O6	Digital output 6
32	O7	Digital output 7
33	O8	Digital output 8

5.2.8.1 Basic wiring diagram of the xPCI-800x digital outputs O1..O8



5.2.9 APCI-8001 pin assignment connector P5, release relay

At connector P5 of the APCI-8001 (corresponds to X6 on the APCI-8008), relay points are provided for the CNC-ready request and amplifier release. These are ‘normally open’ contacts. After the PC is switched on, all relays are switched off after a reset action or an error.
The release relay is activated for the respective selected axis channel by using the *cl()* command for PCAP and the *CL()* command for SAP.

Notice: Depending on the configuration level of the APCI-8001 / APCI-8008, 1 to 3 relay outputs are available. The signal assignment is described here for the FB-RELAIS-3000 adapter (SUB-D 9-pin), which is connected to P5 (X6 on the APCI-8008).
If there are more than 3 axis channels, the connections might be made via a 25-pin SUB-D connector (see Options Manual). The relay is a semi-conductor relay with switching on resistance of max. 25 ohms. The switching power is 25 mA, switching voltage max. 60 V.

Pin **(SUB-D)**	**Name**	**Function**	**Pin at P5 or X6 (FB)**
1	R3-R	Relay S3(4), P contact, CNC ready	1
2	R1-R	Relay S1(3), P contact, release of power amplifier axis channel 1	3
3	R4-R	Relay S4(1), P contact, release of power amplifier axis channel 2	5
4	R2-R	Relay S2(5), P contact, release of power amplifier axis channel 3	7
5		Not connected	9
6	R3-S	Relay S3(4), contact, CNC ready	2
7	R1-S	Relay S1(3), contact, release of power amplifier axis channel 1	4
8	R4-S	Relay S4(1), contact, release of power amplifier axis channel 2	6
9	R2-S	Relay S2(5), contact, release of power amplifier axis channel 3	8

5.2.10 Connection and wiring instructions

5.2.10.1 Earth and current supplies

The xPCI-800x is divided up electrically into two zones. Each zone has its own reference potential, although the different zones are electrically isolated from each other. The first zone contains the xPCI-800x system logic (CPU, memory, etc), and the second zone contains the pulse acquisition (encoder), setpoint value generation and the digital input-output logic. This separation offers maximised protection of the various modules, prevents earth loops and circuits, and provides a high level of immunity to interference from interfering signals, which are often spread from the drives via signal and earth connections.

5.2.10.2 Potential equalisation

Since the above-mentioned supply zones are completely electrically isolated from each other, potential differences of several kV can occur between these zones. To prevent this, potential equalization should take place between the individual zones. This can take place by earthing all supply voltages or through potential equalisation networks on the xPCI-800x.

5.2.10.3

5.2.10.4 Fitting shields

All connections to the APCI-8001 / APCI-8008 must be shielded. The shields must be fitted on both sides on the casing earth wire (not on an internal earth wire, such as pin 50 on X1). Therefore, for SUB-D connectors, massive metal caps (not insulating plastic caps) must be used. Only through correct shielding of all connections is interference-free operation, in particular of the counter inputs, ensured.

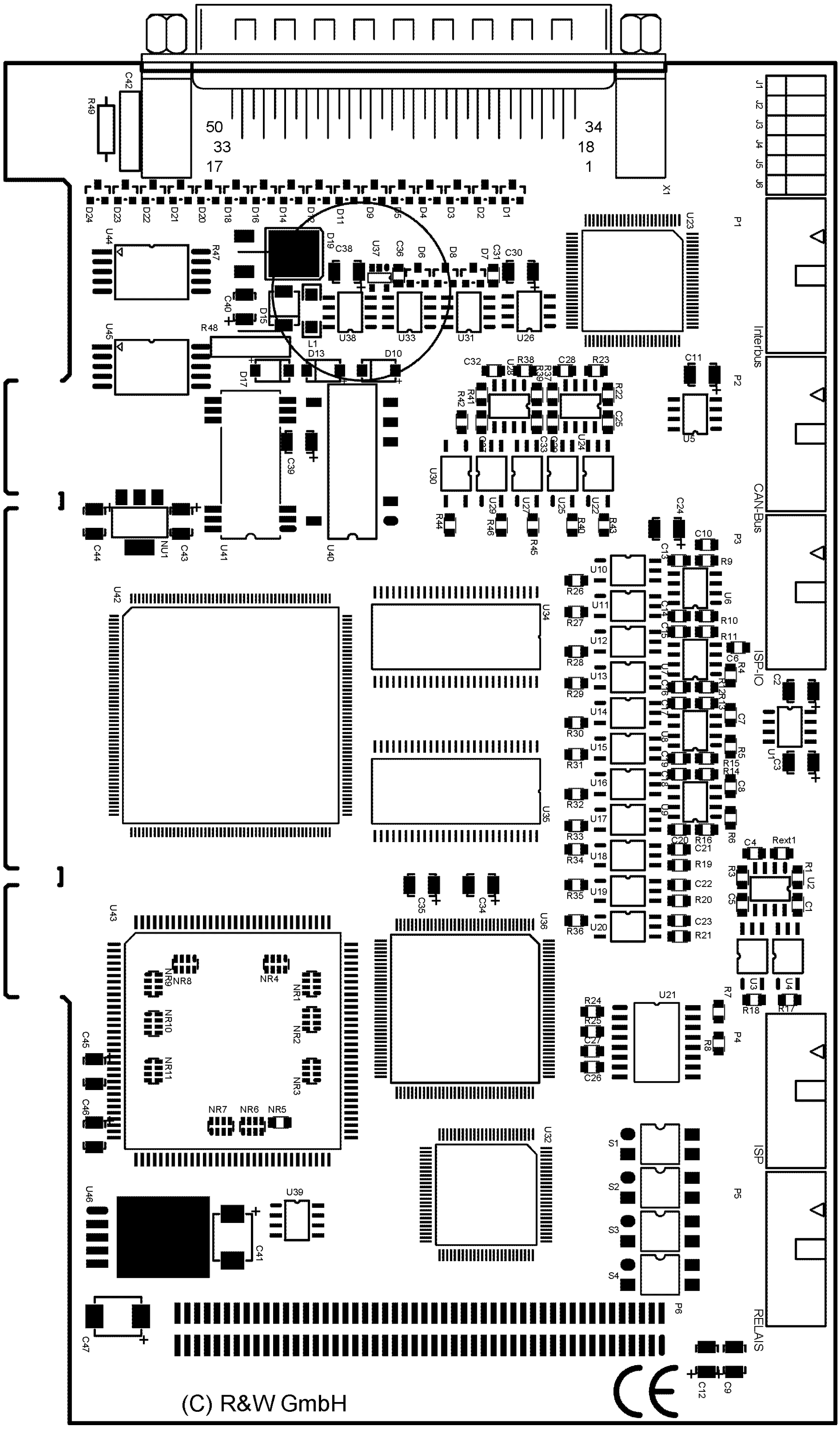
5.3 Using several xPCI-800x controllers in a PC

The driver software contained in the standard delivery allows using several xPCI-800x controllers in a PC. In this case the following features must be observed:

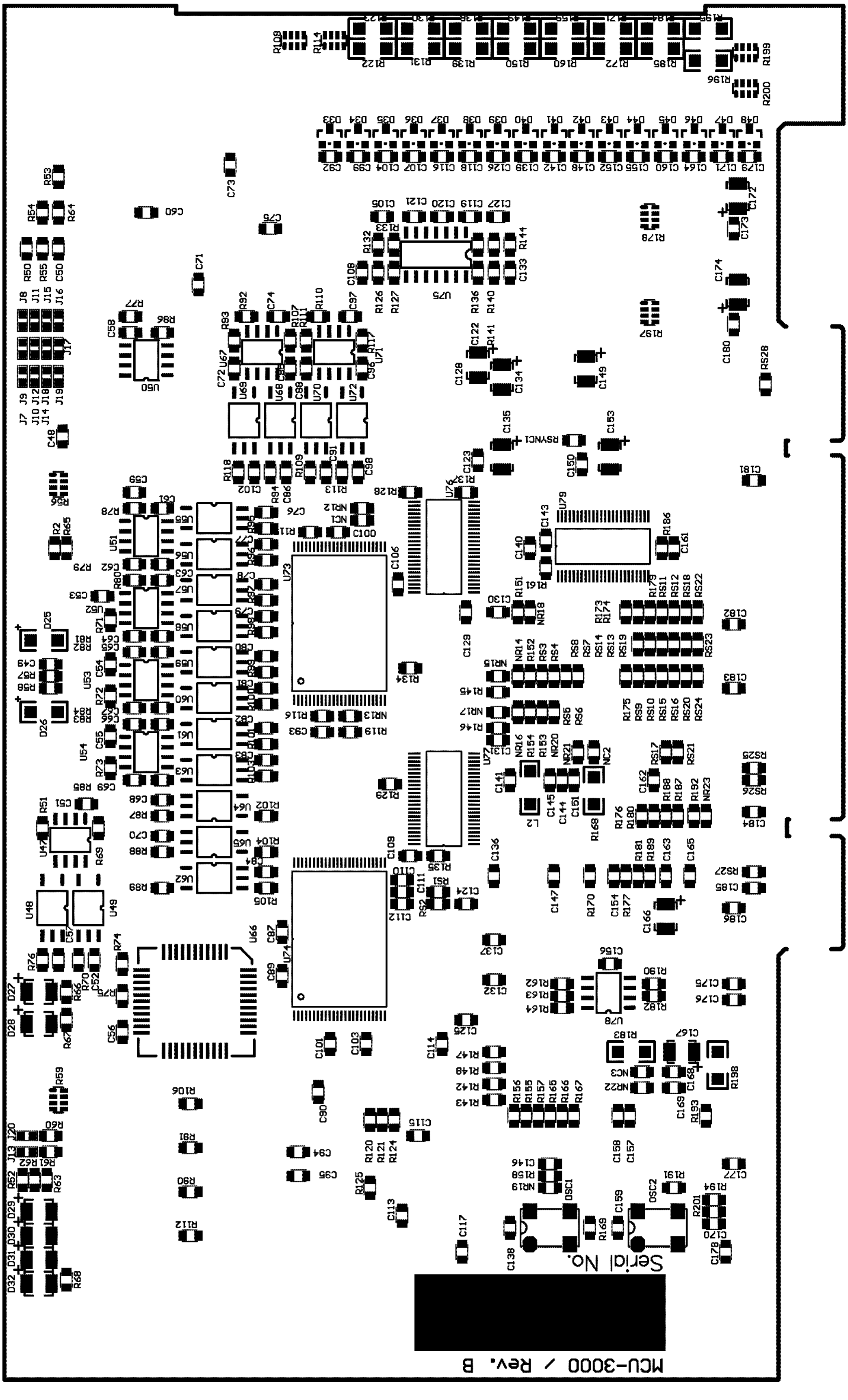
- The boards that are available in the system will be numbered continuously, beginning with 0. If there is only one device in the system, a 0 will be allocated. The numbering is realized by the Plug & Play Bios or the operating system and depends from the slot in the PC.
- An xPCI-800x controller can be selected with the dll-function CardSelect() (see programming manual). After successful selection the number of the board is returned (= parameter). If there was no selection, always device 0 is active when starting an application. After the selection of a device with CardSelect it will be called via the DLL-functions until another device is selected.
- If the selected board is not available in the system -1 will be returned. In this way you can find out how many systems are really installed in the PC. After successful calling of CardSelect, device 0 is selected.
- The board number depends from the slot, in which the respective devices are installed, but not be board itself. Thus, the installation position must be kept at any time, as otherwise the allocation could be changed.
- Firstly, in the user software, the command InitMcuSystem3 () must be called as usual. After this, with CardSelect any device that is installed in the PC can be selected. Then for this and all following devices the command InitMcuSystem3() must be called newly. If necessary, also each control must be booted separately. In this case it must be considered, that for each device in the system, global data structures (e.g. tsrp[]) must be declared separately.
- When calling a device, the data structures (especially TSRP) that are declared for each device must be used.
- A board can be selected in the program mcfg in the window „Projekt Parameter“ on the register card „Environment“. This information will be saved when quitting mcfg and will be activated automatically after new calling. Be very cautious if axes movements shall be done in mcfg. It must be ensured that also the required device is selected.
- A board can be selected in the program fwsetup on the register card „Tools“. However, this information will be not saved when quitting fwsetup.
- In order to guarantee that the required device is called according to the device number, for example according to the pin labelling but not according to the slot, each device can get an Environment Variable for labelling. Then this variable can be requested with getEnvStr(). In this way it is possible to use the device numbers variably. For this, please read section 3.12 et sqq. and programming manual, section 4.4.12.

Be very cautious at multi-thread-applications because due to a thread-change, the execution of the program codes can be switched at any time. A thread-switch during access to the APCI-8001 / APCI-8008 must be avoided, if after a thread-switch an other than the currently selected device shall be called.

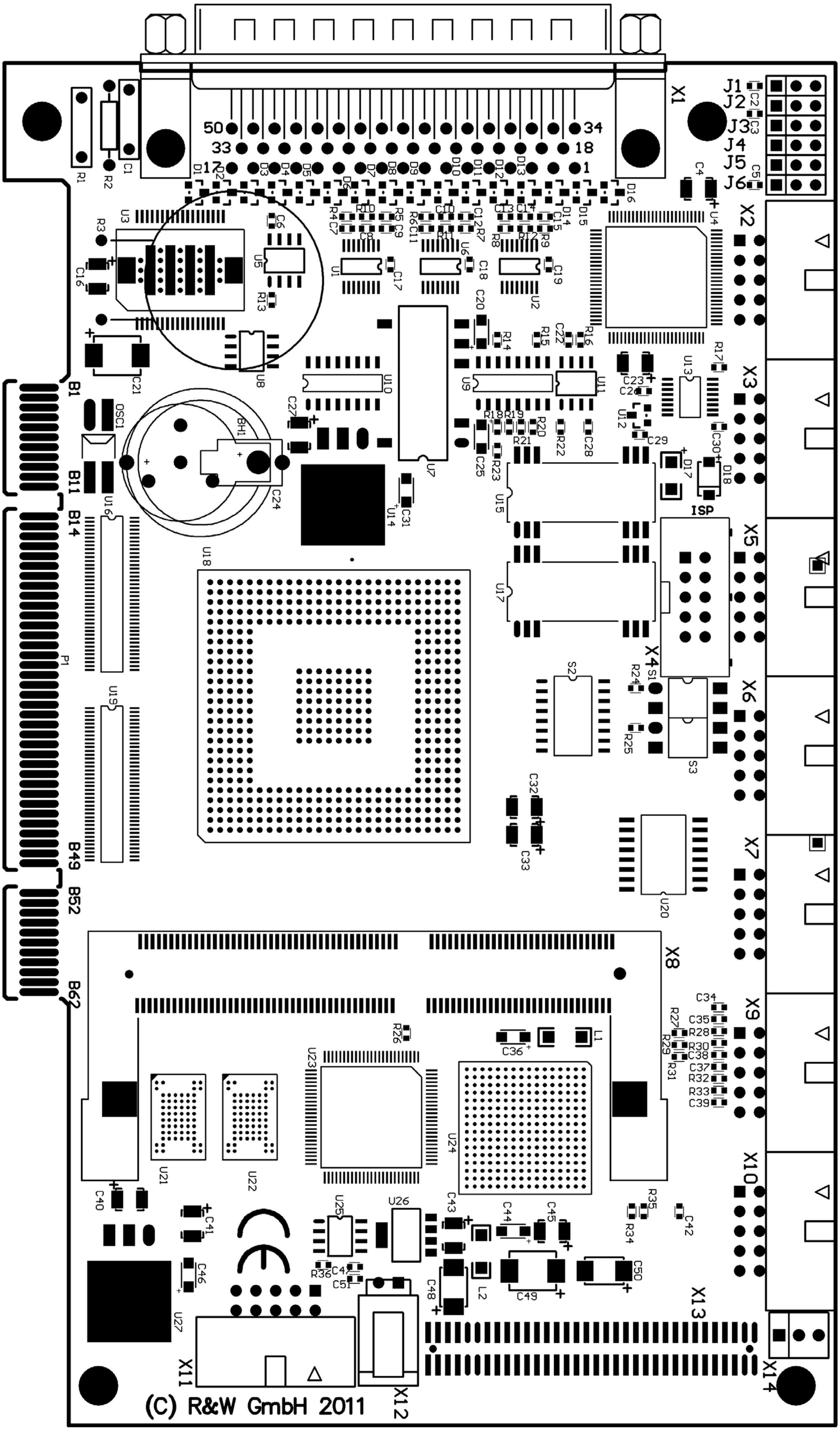
5.4 APCI-8001 component mounting diagram



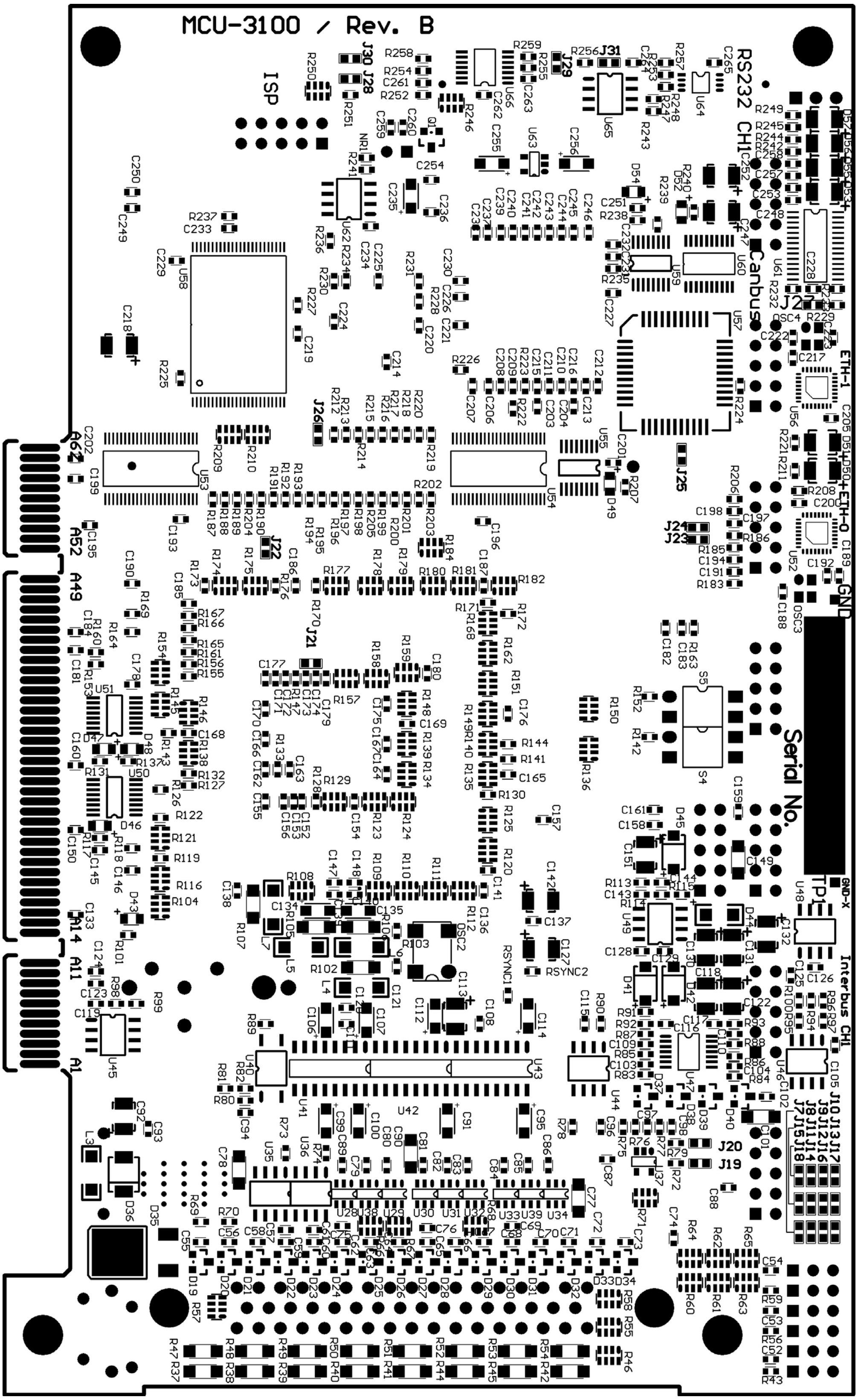
5.5 APCI-8001 component mounting diagram (bottom side)



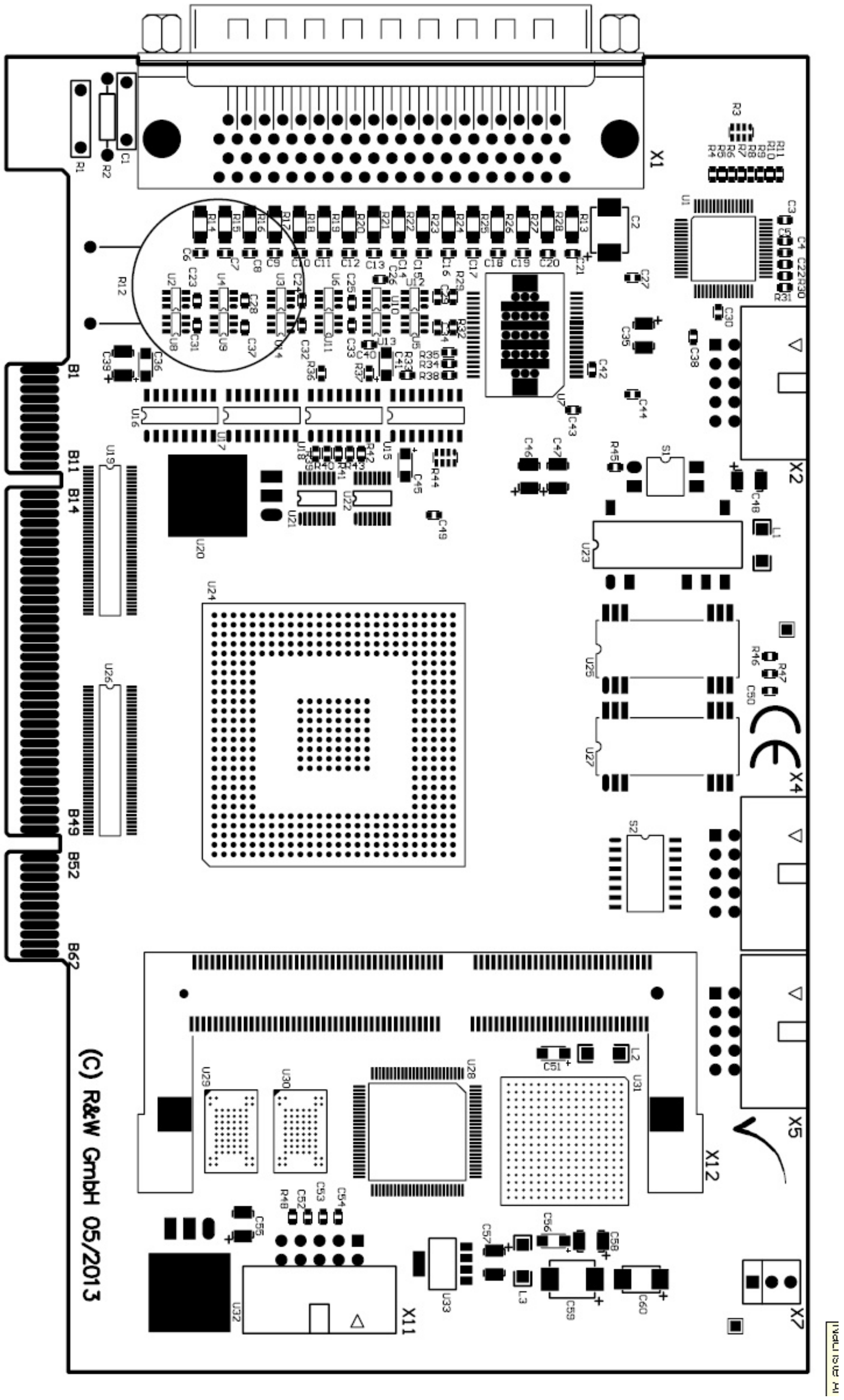
5.6 APCI-8008 component mounting diagram



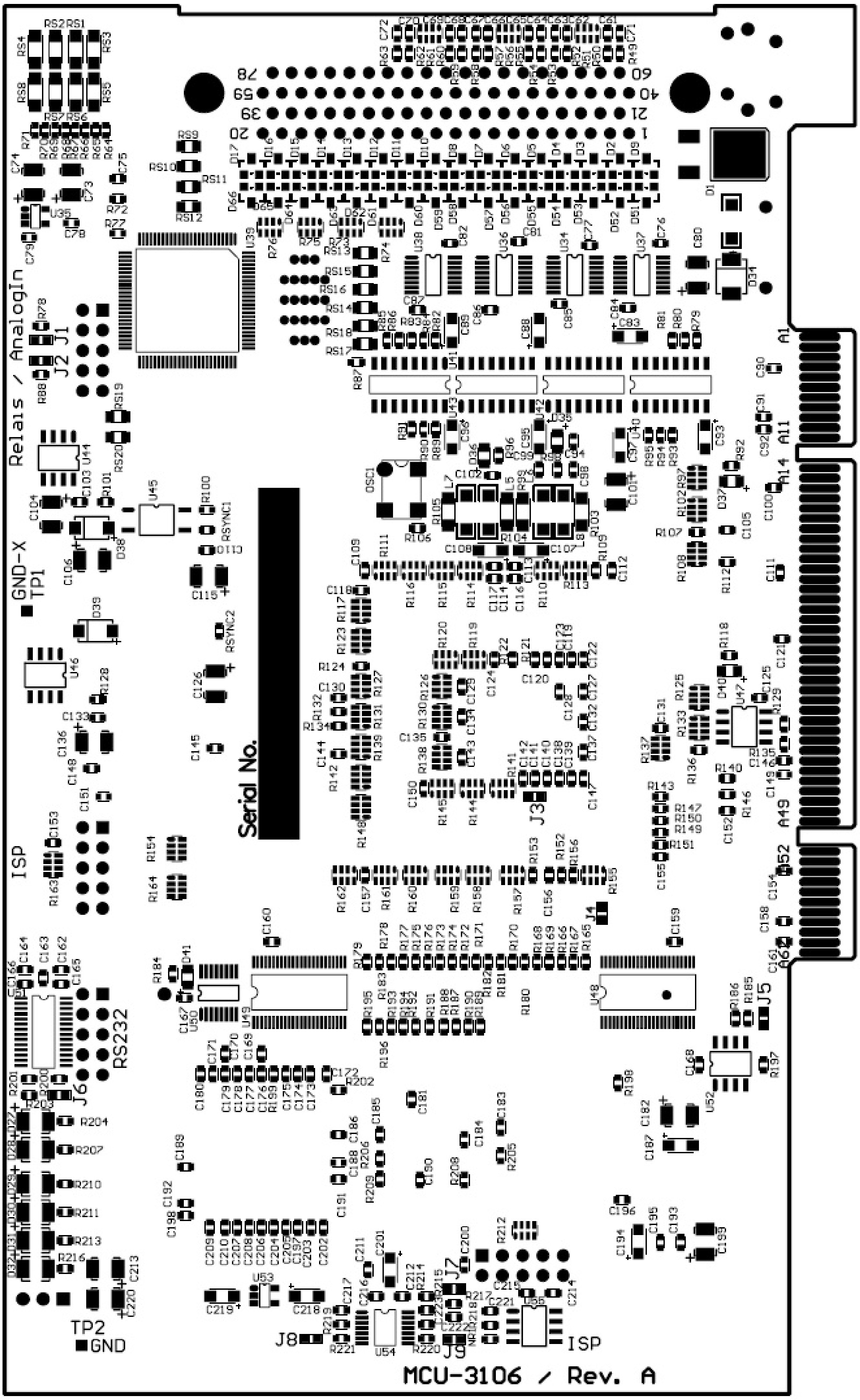
5.7 APCI-8008 component mounting diagram (bottom side)



5.8 APCI-8008-STP-EVAI component mounting diagram



5.9 APCI-8008-STP-EVAI component mounting diagram (bottom side)



5.10 Technical data of APCI-8001

Axes:	1, 2, or 3. extension up to 8 axes with option print OPMF Mixed operation of servo or stepper motors possible
Axis processor:	RISC, MIPS R5K range Elementary frequency: 150 MHz (250 MHz Option), word length: 64 Bit Peak instruction rate: 325 Dhrystone 2.1 Mips and 500 MFlops
Memory:	APCI-8001: 16 MB SDRAM 1 MByte Option 8 kB FLASH for hardware system parameters
Bus:	PCI Universal, word length: 32 bit, Bus frequency: 33MHz or 66 MHz
Addressing:	PCI Plug & Play approx. 80 MB of physical address memory is occupied (not PC main memory)
Encoder inputs:	Directional discriminator for incremental encoders with two 90° phase-shifted pulse trains and reset pulse, or its inverted pulse train (6 channels) SSI absolute encoder
Pulse signal	5 V RS422 or TTL
Incremental encoder evaluation:	x4, 32 bit with sign, 5.0 MHz (20 MHz after quadrupling)
SSI encoder evaluation:	1..32 bit, gray/binary codes, variable frequency 30 kHz .. 10 MHz
Encoder supply:	External auxiliary voltage depending on encoder type (5..30 V)
Setpoint value outputs for servo power transformers:	16-bit-DA converter, +/-10 V, 5 mA, potential-free
Setpoint value outputs for stepper motor transformers:	RS422 pulse and directional signals and their inverted pulse trains, output current usually: -60 mA (max. -150 mA) Pulse frequency: max. 10 MHz
Digital inputs:	16 optically decoupled inputs 18..36 V, input current at 24 V approx. 8 mA. Function is freely programmable Low level: 0..10 V – High level: 16..30 V
Digital outputs:	8 optically decouple outputs, output type: PNP 24V, 500 mA (internal current limit at 1 A) Function mode is freely programmable Setpoint state programmable after reset Relay outputs max. 60 V/100 mA
Safety functions:	Watchdog circuit, power-on reset, efficient CPU exception model

External power supply:	24 V power consumption depending on load of the digital outputs
Design:	Short plug-in boards, x8 multi-layer, 1 slot required
PC power supply:	3.3 V/0.8 A, Notice: 3.3 V supply voltage is sometimes not provided by older motherboards. 5V/1.0A,
Cascading:	APCI-8001: up to a total of 8 axes with OPMF option
Controller software:	PIDF (PID controller with forward compensation)
Control times:	1.28 ms (idle time approx. 0.05 ms) Optional approx. 0.3 ms to 4 ms
Interpolation:	2D .. 3D linear, 2D .. 3D circular, helical, asynchronous and synchronous interpolation with secondary axes
Connectors:	APCI-8001: 50-pin SUB-D connector complete peripheral connection 10-pin FB connector with 3 potential-free relay points 10-pin FB connector for CAN-Bus (Option), 10-pin FB connector for Interbus (Option)
Other options:	Spline and CAD interpolation, electronic gear unit (e.g. Gantry axes), G-Code programming, velocity limitation at profile transition through look ahead, surface area processing, support of non-feed rate axes, scanner functionality, ELCAM functionality (cam plate control), flying saw, SSI absolute value encoder, PWM manipulated value outputs.
Manufacturing:	The board is manufactured according to DIN ISO 9001.
Test:	The board is tested according to CE directives.

6 Settings and plans

After the xPCI-800x hardware and software components have been installed correctly, the axis and motor-specific settings and plans can be made using the *mcfg.exe* TSW program, as described in the following sections.

6.1 Isolating output for power output stage

It is sometimes necessary to isolate the power output stage only if the control loop is closed. You can do this using a programmable xPCI-800x digital output, or the release relay provided for this purpose [section 5.2.9], which is configured with PAE function (MCFG / section 1.7.2.6). This output is activated by closing the control loop. In addition, this output can be used to control a fail-safe brake. If a speed controller is being used, however, the amplifier must be disabled, as otherwise the drift may cause torque to build up.

6.2 Determining the PIDF filter parameters

The axis and motor-specific filter parameters *kp. ki, kd* and *kpl* can be set empirically or analytically. The *mcfg.exe* program offers the option of displaying the system behaviour graphically. This enables the control response to be assessed accurately. Every time you set the filter parameters, you should check if the manipulated variable output and the position feedback are being executed with the correct phase angle, as otherwise the motor axis will immediately drift out of control after the control loop has been closed, if a system deviation occurs.
For all experimental settings when the motor axis is connected, you must remember that the system may vibrate at considerable amplitudes and with high accelerations. Any danger to persons or the machine itself must be prevented by appropriate precautionary measures. Furthermore, a system which initially appears to be stable may start to oscillate due to excitation.
Possible protective measures are an EMERGENCY STOP switch, decoupling the motor axis from the load, etc.
Contouring error monitoring is also possible here.

Notice: For more information on the PIDF filter, see [PM/section 2.1.2].

6.2.1 Speed controllers

A proportional controller is sufficient to control a control system with a subordinate speed controller. To set this, first set all filter parameters to zero and *kp* to 1, for example. Now you can use *kp* to vary the setting until you find a suitable control response.
An additional improvement in the control behaviour can be reached through the velocity pre-control. In order to determine this value experimentally, the proportional part (kp) is set on 0. When a suitable value is found, the previously determined value is entered again from kp.

Caution: If kp is set to 0, the axis is not regulated and will only be operated in a controlled way. Thus, significant deviations between real value and set value are possible. In this situation, the system must be monitored permanently by the user. Before re-adding the proportional part (kp), the control loop must be opened as otherwise unexpected axis movement may occur.
By an additional integral part *ki* , a permanent control deviation at position control, e.g. the input offset of the speed controller can be avoided.

6.2.2 Current amplifier

When using a power module that has been designed as a current amplifier, a PD controller (*kp, kd, kpl*) is required. To prevent the contouring error during traversing and in the event of static loading of the motor shaft, an additional integral component can also be used here. As a rule of thumb, you can assume the following equation:

$$T_N >= 5 * T_V$$

and

$$T_V >= 5 * T_A$$

Now you can use *kp* to vary the settings until you find the best working point. At this working point, you can use *kpl* to vary the settings again. If the system becomes stable, but is too soft, the T_V / T_A ratio can be reduced.

6.2.3 Voltage amplifier

When using a power module that has been designed as a voltage amplifier, a P or PD controller is required. The position error under static loading can be prevented by an additional integral component. Experimental filter parameter settings are made in the same manner as for the speed controller. The controller hardness can be improved by the derivative component with which the motor's mechanical or the electrical time constant can be compensated.

6.2.4 Stepper motor power amplifiers

6.2.4.1 Stepper motor system without position feedback

When using stepper motor power amplifiers without displacement feedback, all that is needed is a proportional controller with an amplification of *kp* = 0.04 and a forward compensation. All other filter parameters are automatically set to 0. The controller parameters are automatically set by the system. It is not possible for the user to make any settings.
Although this seems to be an open-loop system, the control loop must still be closed with CloseLoop when the axis is handled. Here, the actual value is the number of steps that are actually output. In this mode, verification with actual value encoders is possible. However, the encoder value is kept in the aux variable (rdaux, wraux functions) und in the digits unit without conversion.

6.2.4.2 Power amplifier with step/direction setpoint value input and position control

For the control unit, the same specifications as with stepper motors without position feedback apply when power amplifiers with a step/direction setpoint value specification are used, which are operated in the Closed-Loop mode, i.e. with displacement feedback. Here, however, a permanent step error will not occur as the position is controlled by the power amplifier. In such a configuration, different motor types like direct current motors, asynchronous motors, electronically commutated motors or stepper motors can be used.

6.2.5 Pre-control

Using parameters *kfca* and *kfcv*, an acceleration and velocity pre-control signal can be created. The pre-control feature enables the contouring error to be reduced during positioning operations. The stability of the control loop is not affected by the pre-control function.

6.2.5.1 Determining the coefficients

To determine the pre-control coefficients experimentally, first run a short trapezoidal profile, and use the graphical system analysis to assess it, in order to set suitable profile data and scaling parameters. Run the profile with medium acceleration and velocity. The acceleration ramp, deceleration ramp and linear traversing range should be more or less evenly distributed on the screen and displayed in full. The control algorithm is then deactivated by resetting the *kp, ki* and *kd* parameters. The pre-control parameters can now be modified until the setpoint and actual speed characteristics coincide in good approximation after the preset profile has been run. After every modification to the parameters, you must select the [Clear Position] and [Update Filter] menu items, so that the new parameters can be applied.

When using a current amplifier, first set the acceleration pre-control *kfca* so that the acceleration ramps for the setpoint and actual speed coincide. Then set the velocity pre-control so that the speed in the linear speed range runs parallel. Now both values can be slightly altered alternately until setpoint and actual speed characteristics best coincide.

When using a speed controller, begin with the velocity pre-control *kfcv*.

After the best possible parameters have been found, the *kp, ki* and *kd* filter parameters are entered again and the behaviour is checked again. Ensure that the data you have set is saved.