

# IPCOMM

phytron®

## Communication Software for Stepper Motors Control Units IPP, GSP, GCD and GLD



**Manual 1120-A010 GB**

**customized solutions**  
in motion



## **IPCOMM**

# **Communication Software for Stepper Motor Control Units IPP, GSP, GCD and GLD**

© 2005

All rights with:

Phytron GmbH

Industriestraße 12

82194 Gröbenzell, Germany

Tel.: +49(0)8142/503-0

Fax: +49(0)8142/503-190

Every possible care has been taken to ensure the accuracy of this technical manual. All information contained in this manual is correct to the best of our knowledge and belief but cannot be guaranteed. Furthermore we reserve the right to make improvements and enhancements to the manual and / or the devices described herein without prior notification.

We appreciate suggestions and criticisms for further improvement. Please send your comments to the following

E-mail address: [doku@phytron.de](mailto:doku@phytron.de)

## Contents

1 What is IPCOMM ? .....	4	5.3.9 RS Monitor >> .....	23
1.1 First steps with IPCOMM .....	4	5.3.10 SFI >> .....	24
1.2 The Main Features of IPCOMM on ONE View .....	5	6 Error Messages .....	25
2 System Requirements .....	5	6.1 Interface Problems .....	25
3 Program Installation .....	5	7 Adapter Cables .....	26
3.1 How to Install the Program .....	6	8 Copyright and Limitation of Warranty .....	28
3.2 File Names .....	8	9 IPCOMM Protocol .....	29
3.3 Update of IPCOMM .....	8	9.1 Structure of the Master-Slave- Telegram .....	30
4 The Working Environment Used by IPCOMM .....	9	9.2 Structure of the Slave-Master- Telegram .....	31
5 The IPCOMM Menus .....	10	9.3 Handling of Transmission Errors .....	32
5.1 Starting and Closing of IPCOMM .....	10	9.4 Status Information of the PSC .....	33
5.2 The Menu <i>File</i> .....	10	9.4.1 Short Status .....	33
5.2.1 New .....	10	9.4.2 Extended Status .....	34
5.2.2 Open .....	10	9.5 Definition of the Commands .....	36
5.2.3 Save .....	10	9.5.1 Stop Commands "H" and "B" .....	36
5.2.4 Save as .....	11	9.5.2 Motion Command "G" .....	37
5.2.5 Close .....	11	9.5.3 Parameter Command .....	38
5.2.6 Print .....	11	9.5.4 I/O, Status and Info Commands "I" .....	39
5.2.7 Exit Program .....	11	9.5.5 Clear Commands "C" .....	40
5.3 The Main Menu Window .....	12	9.5.6 Flash Programming Commands "F" .....	41
5.3.1 Transmit / Close .....	12	9.5.7 Write Commands "W" .....	41
5.3.2 Motion Commands .....	13	9.5.8 Extended Commands "E" .....	42
5.3.3 Parameters .....	14	9.5.9 Retry Command "R" .....	43
5.3.3.1 Current >> .....	14	9.5.10 Special Commands "S" .....	43
5.3.4 Range Register .....	15	9.6 Online Operation Mode of the PSC .....	44
5.3.5 Short Status of the Axes .....	15	9.7 PLC Operation Mode of the PSC .....	44
5.3.5.1 Status >> .....	16	9.8 Additional Information .....	47
5.3.5.2 Reset>> .....	17	9.8.1 Tools .....	47
5.3.6 Digital Input / Output Status .....	17	9.8.2 Structure of Software .....	48
5.3.7 Program Axes .....	18	9.8.3 Configuration of the PLC Sequences .....	48
5.3.7.1 PLC >> .....	18	9.8.4 Extended PLC Sequences .....	50
5.3.7.2 PLC Ext. >> .....	19	9.8.5 Interface Communication .....	52
5.3.7.3 Ramps >> .....	21	9.8.6 Baud Rate Selection .....	54
5.3.7.4 Actual Parameters .....	22	10 Index .....	55
5.3.8 Optional add-on card >> .....	22		

## 1 What is IPCOMM ?

---

IPCOMM is a Windows® program, which is delivered as setting-up software with every PSC (Phytron **S**tepper motor **C**ontrol unit). IPCOMM is used to configurate the “Phytron Stepper motor Control units” IPP, GSP, GCD and GLD. You can set parameters and program PLC sequences.

### 1.1 First steps with IPCOMM

---

**Remark:** For most of the users Microsoft® Windows® is a familiar operating system. Therefore, we take for granted the application of windows specific inputs for the MC-COMM operation, switches, mouse clicks. You will find helpful information about your Windows operating system in your Windows user's guide.

- Please check whether your PC fulfills the hardware and software requirements as shown on page 5 (chap. 2).
- Please install IPCOMM following the instructions starting on page 6 (chap. 3.1).
- Start the program by clicking the button **Start** on the taskbar. In the menu *Start* select **Programs/ IPComm V2/IPCommV2** by clicking the left mouse key. Or start the program by double-clicking on the selected IPCOMM-Icon.

**Important:**

- Now it is the time simply to play with the program and to test some of its functions. You will see that the handling is very simple and self-explaining.
- For further instructions please refer to the appropriate chapters in this manual. You will find a summary, many cross references and an index at the end of the manual. This will help you to get an overview about all functions of the program.
- For programming commands and other information about your control units, please refer to the special programming manuals and hardware manuals of your devices.
- Parameters stored in your control unit and new programs created with IPCOMM can be printed.

## 1.2 The Main Features of IPCOMM on ONE View

- Desktop with pull-down menus and mouse support
- Hotkeys for frequently used commands
- Menu dialog in 3 languages (German, English, French)
- IPCOMM runs in 2 modes: Offline and Online
- Simple and fast parameter settings

## 2 System Requirements

IPCOMM can be used on PCs with at least the following features:

- Pentium 100 MHz
- Operating system Windows® 95, 98, 2000, NT or XP®
- At least 32 MB RAM free memory
- Mouse
- CD-ROM drive, in case of CD program installation
- Free interface RS232 for controller connection

## 3 Program Installation

The following folders are saved on CD:

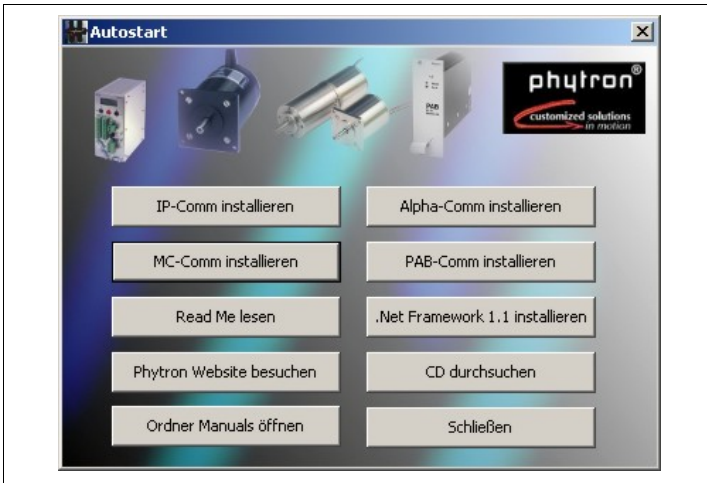
English	Includes the files for a setup in English
French	Includes the files for a setup in French
German	Includes the files for a setup in German

In every folder the following files are stored:

Setup.exe	Installation file
Setup.ini	Configuration settings
IP-Comm Setup Eng.msi	Windows Installer Package
InstMsiA.exe	Packed installer file A
InstMsiW.exe	Packed installer file W

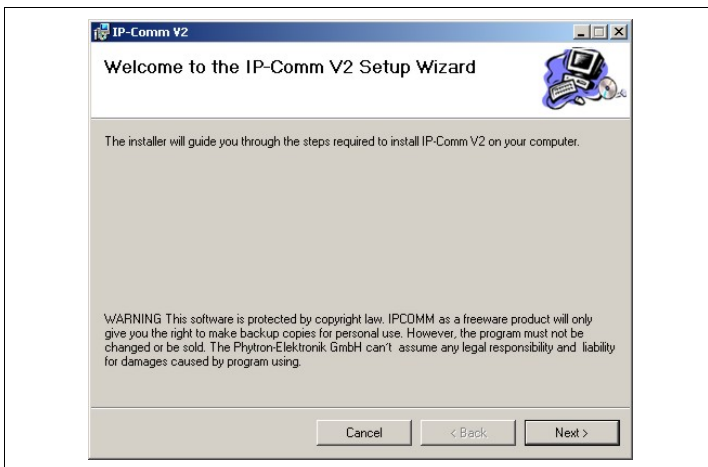
## 3.1 How to Install the Program

1. Insert the program CD into the CD ROM drive.
2. If the autostart program does not start automatically, you'll open the Phytron installation program [CD-ROM address]\Autostart.exe by mouse click.
3. The Phytron installation program starts the following menu:



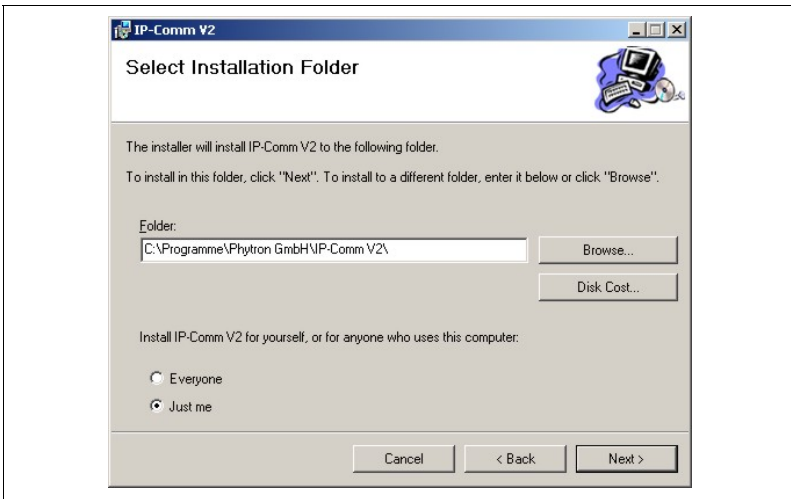
Please start the IPCOMM installation program by mouse click on **Install IP-COMM**.

4. The installation program copies the files into the RAM after program start.
5. The screen shows the following message:





6. Please click on **Next >** and the following message is on the screen:



7. **c:\Programs\ Phytron GmbH\IPComm V2** is the standard directory of IPCOMM. With **Browse** another directory can be selected. Click on **Next >** to start the installation.
8. The files will be stored into the defined directory, some files will be stored into the Windows directory.
9. When the message '*IPCommV2 successfully finished. Click on close*' on the desktop, the installation is successfully finished. Click on **Close** to finish.
10. Remove the CD from the CD-ROM drive.
11. For quick program start we recommend to click on the IPCommV2 icon on the desktop.

**Remark:**

**If you have got the IPCOMM files on another way (e. g. email), the program installation can be done like in chapter 3.1 (beginning at item 2). Instead of CD-ROM drive select the corresponding directory, on which the file SETUP.exe is stored.**

# Manual IPCOMM

---

## 3.2 File Names

---

After installation the following files are stored in the directory:

IPComm.exe	Program IPCOMM
IPS2000.exe	Program IPSearch (see chap. 9.8.1)
opm.exe	Program opm (see chap. 9.8.1)
comdlg32.dll	Program library
oledlg.dll	Program library
SFI	Folder with stored data for the power stage GSP with step failure indication (SFI)

## 3.3 Update of IPCOMM

---

If you want to install an update in case of an existing IPCOMM copy, the installation program requests your confirmation and deletes the old version before installing the new.

## 4 The Working Environment Used by IPCOMM

After the successful program start of IPCOMM the desktop will show up like in Fig.1:

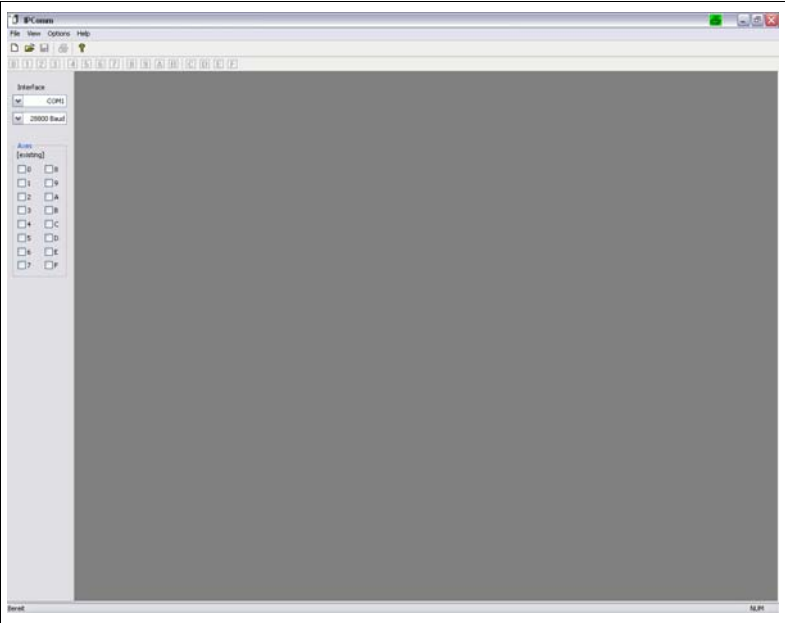







Fig. 1: Desktop after program start

The working environment of IPCOMM consists of six main components:

1. The selection menu in the top line on the desktop: File, Edit, Options, Window, Help
2. The selection menu of some commands with icons (see table)
3. Selection of COM interface and baudrate
4. Selection of the device address: address 0 to 9 and A to F
5. The working area
6. Status line

Icon	Signification
	Write new parameter file
	Open new parameter file
	Store the parameter file
	Print the parameter file
	Information about IPCOMM

# Manual IPCOMM

---

## 5 The IPCOMM Menus

---

In this chapter you will find all you need for starting and leaving IPCOMM as well as a detailed description of each menu command, dialog box, switch etc.

### 5.1 Starting and Closing of IPCOMM

---

Select **Start / Programs / Phytron GmbH / IPComm V2** by clicking to program start with the left mouse key. Or click on the IPCOMM icon, if it exists on your desktop.

Use the menu command **File / End** to leave IPCOMM. IPCOMM will display a warning message if there is unsaved data in one of its editor windows.

### 5.2 The Menu *File*

---

In the menu **File** you will find commands to open files, to create new files, to save changed files, to save a parameter file by a new name, to print files and to leave IPCOMM.

#### 5.2.1 New

---

In the submenu **File / New** you select the editor window for creating a new parameter file in the offline or online mode .

IPCOMM allocates files like <NAME>.IPC.

#### 5.2.2 Open

---

In the sub menu **File / Open** you can select an existing parameter file. The dialog box contains an edit box, a file list and action buttons **Open** and **Cancel**.

Select a file name in the file list. The file will be opened and loaded into a new editor window.

#### 5.2.3 Save

---

**File / Save** stores the data from the active editor window into a file.

.

### 5.2.4 Save as

---

With **File / Save as** you can define a new file name for the data you are editing in the current editor window, or you can store the data with the same file name, but into another directory or drive.

### 5.2.5 Close

---

With **File / Close** the file in the active editor is closed. The IPCOMM program remains opened.

### 5.2.6 Print

---

**File / Print** opens the Windows specific print sub menu where you can select the printer type and the print properties.

### 5.2.7 Exit Program

---

With **File / Exit** you finish IPCOMM. If files are open which are not yet saved, IPCOMM will ask you to save them.

# Manual IPCOMM

### 5.3 The Main Menu Window

With **File / New or Open** you open a window, where you can define parameters, motion commands, Axes, range registers or interrogate axe status and digital I/O:

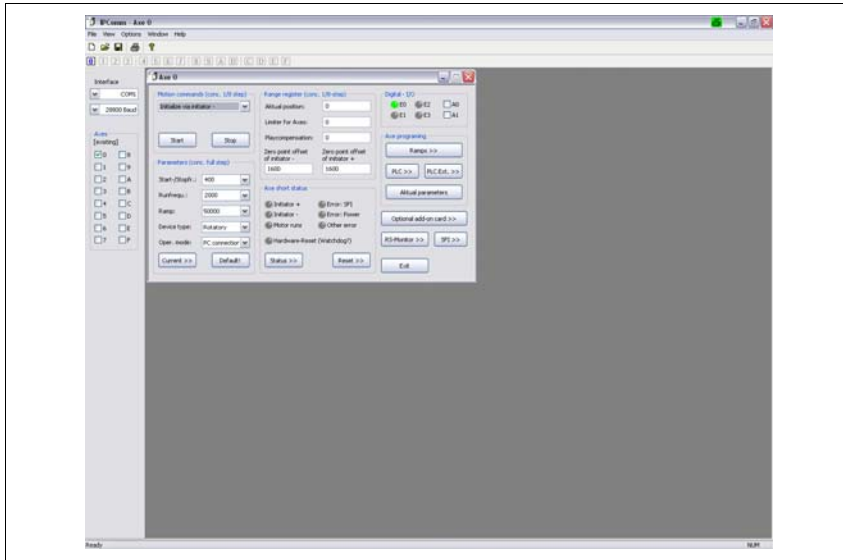


Fig. 2: The main menu window

### 5.3.1 Transmit / Close

## Exit

The parameter file <NAME>.ipc is closed without storing.

## Transmit

In the OFFLINE mode:  
The preset values are transmitted to the control unit.

**Remark:**

If you create a new parameter file, the parameters are installed for those axes, which have been selected by mouse click to the address bar.

### 5.3.2 Motion Commands

Here you can select the motor running mode by clicking the DropDownkey [!]. The motion commands refer to 1/8-step:

Motion commands	Signification
Initialize via Initiator +	The axe is moved in + direction to a reproducible reference point.
Initialize via Initiator –	The axe is moved in – direction to a reproducible reference point.
Position absolute...	Motion to absolute position
Go relative...	Motion from the actual position by x-counter values
Free running dir. +	Free motor running in + direction
Free running dir. –	Free motor running in – direction
A step dir. +	1 motor step in + direction
A step dir. –	1 motor step in – direction

**Start** Only in ONLINE mode: The preset command is executed.

**Stop** The motor is stopped.

5.3.3 Parameters

In this window you can set different parameters (referred to a full step) by clicking the DropDownkey [↓], .

Parameters	Values	Signification
Start /Stop frequency	50...1000	Frequency in Hz
Run frequency	50...10000	Frequency in Hz
Ramp	programmable	Run frequency of the stepper motor during acceleration and deceleration
Device type	Rotatory	Motor drives axis without limiter (rotary axis).
	Linear	Motor drives axes with limiter (f.ex. linear axis or circular table with stopper)
Operation mode	PC connection PLC mode	Direct PC connection to the power stage PLC connection

5.3.3.1 Current >>

The current setting submenu starts by clicking **Current >>** .

Run, Stop and Boost current can be set incrementally by clicking the DropDownkey [↓]:

Step 0 until 9, A until F.

Range of the current delay time: 10 to 1000 ms.

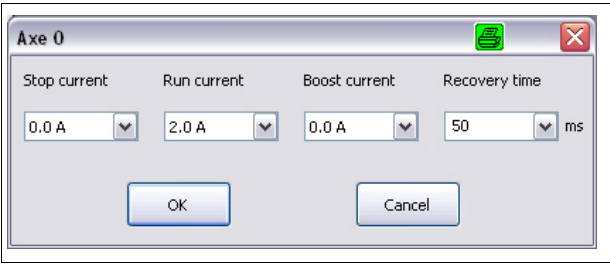


Fig. 3: Current setting

**Default!** The parameter values are set to the internal delivery data (default values).





### 5.3.4 Range Register

Here you can read or change different range register data. The values refer to 1/8 step:

Range register	Signification
Actual Position	Gives the actual position of the range register
Limiter for axes	Value used during free running and initialization. If the number of steps has been reached the running will be stopped.
Compensation for play	During positioning of mechanics typically having a play, the play is eliminated and defined by this value.
Zero point offset from initiator –	After the reference point has been reached the motor is moved by this value in – direction.
Zero point offset from initiator +	After the reference point has been reached the motor is moved by this value in + direction.

### 5.3.5 Short Status of the Axes

Here you can see 7 axes status values displayed by pilot lamps  :

Colour of the 	Signification
green	Active / o.k.
Yellow	Warning
Red	Error
Grey	not active / not available

Status	Pilot lamp does not shine
Initiator +	red or grey
Initiator –	red or grey
Motor is running	green or grey
Error: SFI	red or grey
Error: Power stage	red or grey
Other errors	red or grey
Hardware Reset (Watchdog?)	yellow or grey

### 5.3.5.1 Status >>

By clicking **Status >>** the submenu starts the extended status query.

Here you can see the axes status values displayed by pilot lamps:

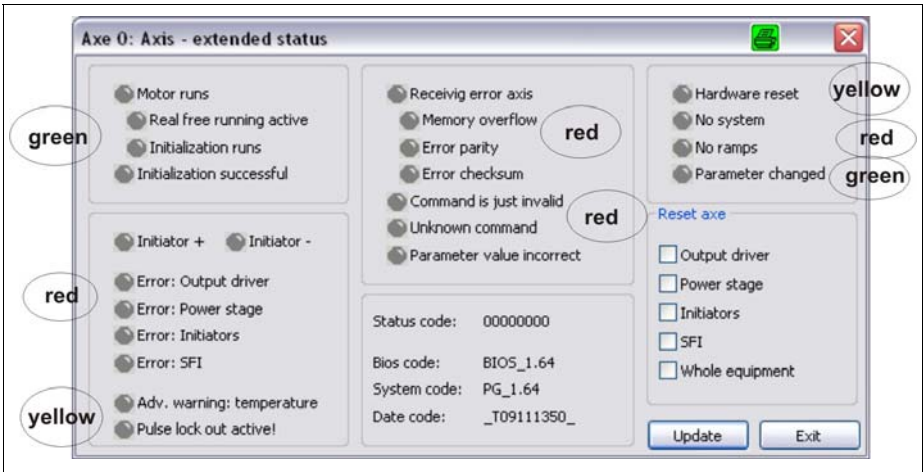



Fig. 4: Extended status of the axes


## Reset axes

Here you can reset the output drivers, power stage, initiators, SFI and/or the whole equipment.


If 1 of the 5 equipments are marked by mouse click with , the

-Button has the marking  in order to reset by mouse click.

 A new status query starts.

 Close the submenu.


### 5.3.5.2 Reset>>


By clicking  the submenu Equipment Reset starts.

You can select the output drivers, power stage, initiators, SFI and the whole equipment by clicking the mouse.




Fig. 5: Submenu: Reset

 Reset of the equipments, which are selected by clicking the mouse.

 The submenu is quitted without Reset activation.

## 5.3.6 Digital Input / Output Status

In this chapter the status of the 4 inputs (E0 to E3) are monitored:

E0 to E3 shine green or grey. The outputs A0 and A1 can be set by clicking the mouse 

5.3.7 Program Axes

5.3.7.1 PLC >>

By clicking **PLC >>** the submenu PLC functions starts.

Here you can write up to 256 (0...256) motion commands (sequences) for the PLC mode. In addition you can test sequences in a PLC protocol simulation and set output signals.



Fig. 6: Submenu: PLC programming

<b>Test</b>	Test of Sequences
<b>Begin</b>	Start of the PLC protocol simulation, which is finished by clicking <b>End</b> .
<b>Copy motion command</b>	The marked sequence is copied to the following sequences.
<b>Program</b>	The selected motion commands are saved into the control unit in online mode.
<b>Transfer</b>	In the offline mode the selected motion commands are filed in the buffer store and can be stored into the parameter file later.
<b>! Delete all !</b>	Delete all motion commands.

<b>Stop</b>	Stop the motion command.
<b>Exit</b>	Quit the submenu.

In **Actual position** the actual counter value, in **Status** the axes short status is shown in hexadecimal numbers.

5.3.7.2 PLC Ext. >>

The submenu extended PLC functions starts by clicking **PLC Ext. >>**.

The selected motion commands (up to 256 (0..255)) are executed automatically in rotation.  
This processing goes as long as the signals STROBE and ENABLE are active.  
(See chap. 9.8.4)

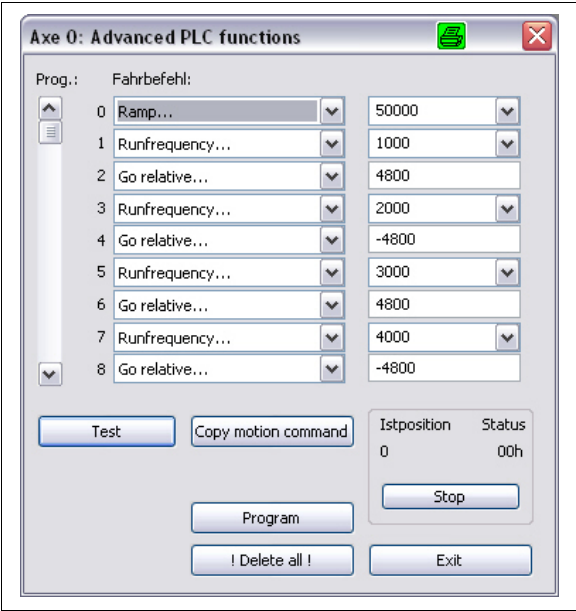


Fig. 7: Submenu: Extended PLC sequences

<b>Test</b>	Test of Sequences
<b>Copy motion command</b>	The marked sequence is copied to the following sequences.

## Manual IPCOMM

---

<b>Program</b>	The selected motion commands are saved into the control unit in online mode.
<b>Transfer</b>	In the offline mode the selected motion commands are filed in the buffer store and can be stored into the parameter file later.
<b>! Delete all !</b>	Delete all motion commands.
<b>Stop</b>	Stop the motion command.
<b>Exit</b>	Quit the submenu.

5.3.7.3 Ramps >>

The submenu ramp programming starts by clicking **Ramps >>**.

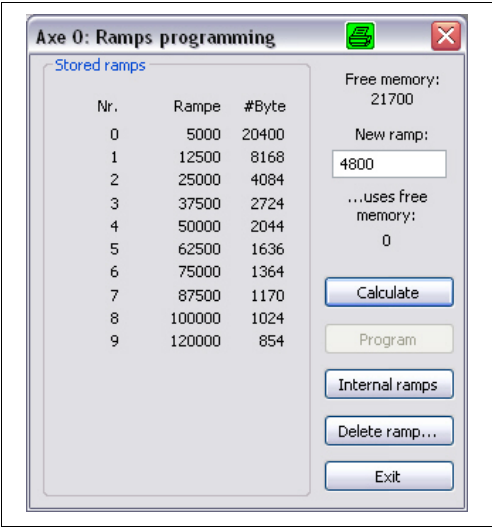


Fig. 8: Submenu: Ramp programming

On the left part of the window you see the **stored ramps (Nr.)** with the required **memory (#Byte)**. 16 ramps (0 until 15) are programmable.

**Free memory** shows the free memory for new ramps.

The smallest possible ramp value is written in the task bar **New Ramp** which is calculated by the free memory value. This value can be overwritten.

In .... **uses free memory**: the memory value is shown which is calculated for the ramp.

<b>Calculate</b>	The memory size required for the entered ramp value is calculated.
<b>Program</b>	The ramp values are stored into the control unit.
<b>Internal Ramps</b>	Device-specific default ramp values are loaded.
<b>Delete Ramp...</b>	A new window is opened: Ramps marked by <input checked="" type="checkbox"/> are deleted and the other ramp values are reprogrammed.
<b>Exit</b>	The submenu is quitted.

## 5.3.7.4 Actual Parameters

By clicking **Actual Parameters** all preset values are stored into the control unit (ONLINE mode).

## 5.3.8 Optional add-on card >>

By clicking **Optional Add-on-card >>** the submenu *Optional Add-on-card* is started.

If the control unit has got more than 4 inputs and more than 2 outputs, it is now possible to monitor all 12 inputs and 6 outputs by the pilot lamps.

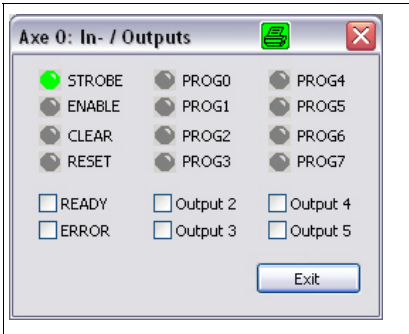


Fig. 9: Submenu: Added inputs and outputs



5.3.9 RS Monitor >>

By clicking **RS Monitor >>** the submenu RS-Monitor is started to send commands directly to the control unit.

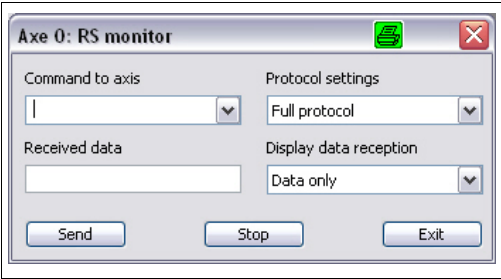


Fig. 10: Submenu: RS-Monitor

The command code is written in **Command to axis**.

The following **Protocol settings** are possible:

- Only STX/ETX
- STX/ETX and checksum
- STX/ETX and address
- Full protocol

The response telegram is defined in **Display data reception**:

- Data without control character
- Short status and data
- All characters

<b>Send</b>	The direct command is sent to the control unit.
<b>Stop</b>	The motor running is stopped.
<b>Exit</b>	The submenu is quitted.

5.3.10 SFI >>

If the control unit is provided with step failure identification (SFI), the submenu **SFI >>** is activated. The data (pref. direction, motor step number, encoder resolution) are configured for the step failure identification (SFI):

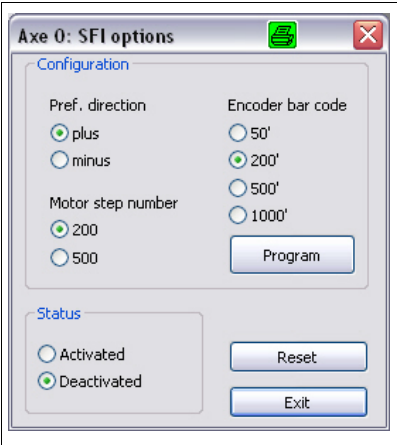


Fig. 11: Submenu: SFI Options

SFI can be activated or deactivated in **Status**.

<b>Program</b>	The preset values are stored in the control unit.
<b>Reset</b>	Reset the SFI.
<b>Exit</b>	The submenu is quitted.

## 6 Error Messages

In this appendix you will find hints how to handle error messages of IPCOMM. In the following text these messages are written in bold or underscored italics. Messages which do not need any additional information are not contained in this appendix.

### Remark:

Error messages resulting from problems with any peripheral device are very often caused by trivial facts like: device is not switched on, cables are missing, cable is defect and so. Please check these items.

### 6.1 Interface Problems

#### **An interface timeout has occurred**

Possible reasons for this problem may be:

- The cabling is wrong (RD and TD pins interchanged).
- The control unit is not in the REMOTE operation mode (Switch Remote/Local).
- The cabling is defect or interrupted.
- The interface parameters selected in IPCOMM do not match the control unit type used.
- No control unit connected

#### **COM Interface already open**

Possible reason for this problem may be:

- The COM interface is already used by another program.

#### **In the Main menu window no COM Interface is displayed**

Possible reason for this problem may be:

- PC has no serial interface

7 Adapter Cables

**Important:** If you connect the control unit to the PC, only use cables with the following pin assignment (see above)! Additional wiring can cause transmission errors!

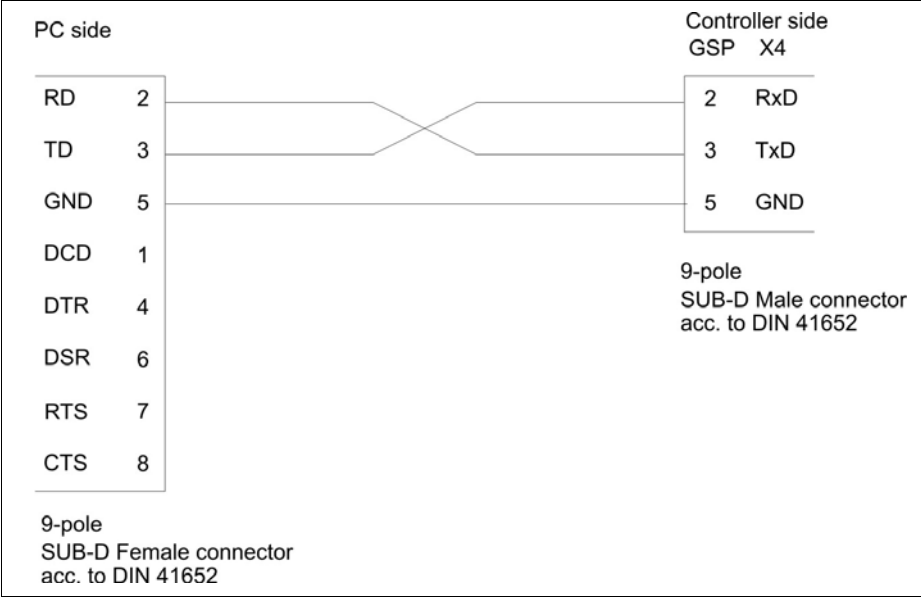


Fig. 12: PC (9-pole female connector) <-----> control unit GSP

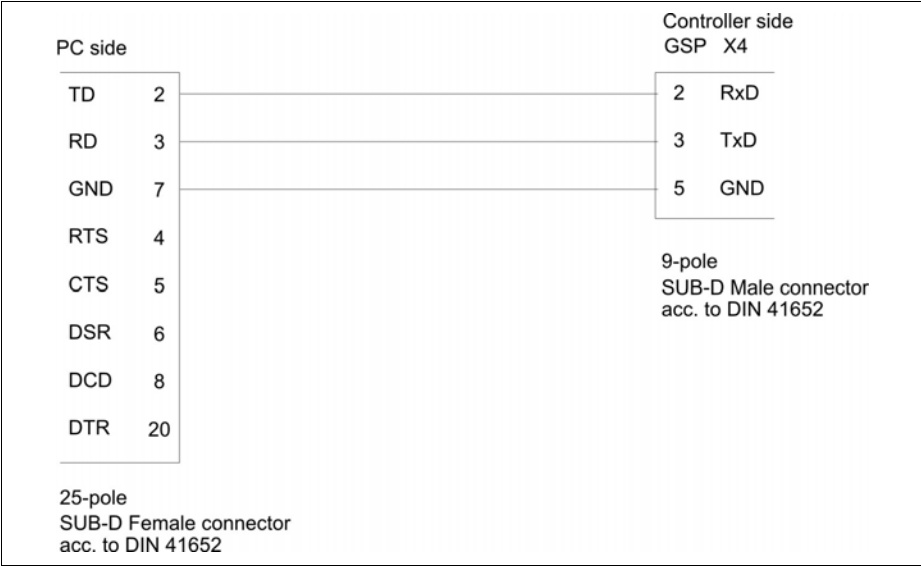


Fig. 13: PC (25-pole female connector) <-----> control unit GSP

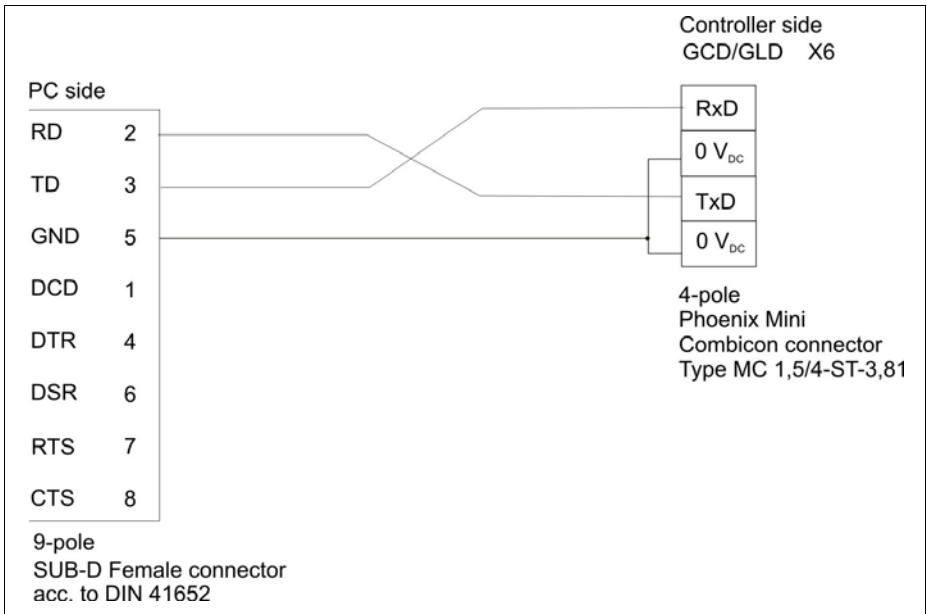


Fig. 14: PC (9-pole female connector) <-----> control unit GCD/GLD

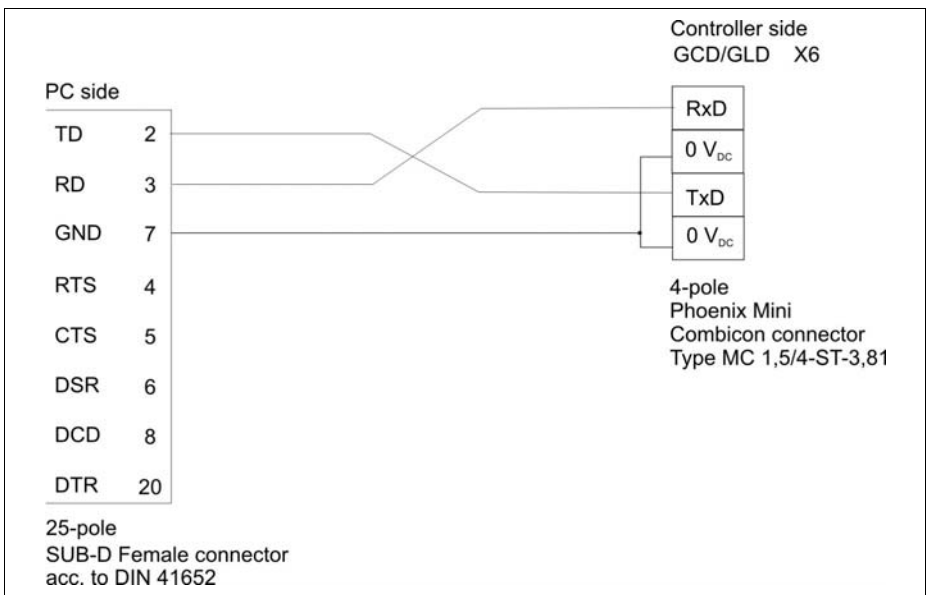


Fig. 15: PC (25-pole female connector) <-----> control unit GCD/GLD

## 8 Copyright and Limitation of Warranty

---

The software IPCOMM and any documentation delivered with it are protected by copyright law.

IPCOMM as a freeware product will only give you the right to make backup copies for personal use. However, the program must not be changed or be sold.

### Limitation of warranties

The software IPCOMM and its documentation have been made with great carefulness and have been reproduced under effective controlling measures. Nevertheless there might be mistakes. Phytron-Elektronik GmbH therefore will take no responsibility for any damage resulting from the use of the software and / or the manual. If you detect any mistake in the manual or in the software, we would appreciate any information about this to increase the quality of the product.

We refer to our delivery conditions and terms of payment, particularly to item VII Liability and item IX Software utilization.

By installing, copying or using otherwise the software product, you agree with our delivery conditions and terms of payment .

If you don't agree to these delivery conditions, you won't be authorized to install or to use the software product.

### Trade marks

In this manual several trade marks are used which are no longer explicitly marked as trade marks within the text. The lack of this signs may not be used to draw the conclusion that these products are free of rights of third parties. Some product names used herein are for instance

- Microsoft is a registered trade mark and WINDOWS is a trade mark of the Microsoft Corporation in the USA and other countries.

## 9 IPCOMM Protocol

This manual describes the “Communication Protocol” which is used to control the “Phytron Stepper motor Control units” of the IPP, GSP, GCD and GLD devices.

In the following the “**Phytron Stepper motor Control unit**” will be named “**PSC**”.

The IPCOMM protocol uses a stringent single master/multiple slaves architecture. The host computer or a PC (master) transmits the instruction code like a data telegram to one or several of max. 16 PSCs (slaves). The data message consists of a start bit, an address byte, several data bytes, checksum and a stop bit.

The slaves (one or several PSCs) response to the instruction code with a data message of their own. In typical applications there is a need to transmit regularly the internal status of the PCSs to the master but not reverse. This makes it necessary to differentiate between the format of the instruction code (master to PSC) and the format of the response message (PSC to master).

When defining the IPCOMM protocol, attention was paid to make the contents of the data telegram easily legible. All data, except the start and stop bits, are ASCII characters. In this way, one can manually set and transmit single instructions to the PSC by means of a low-grade data terminal program.

Each PSC responses only to message telegrams whose first character has been recognized as its indicated address (identity). This address corresponds to the address which has been preset by means of the code switch on the front panel of the IPP and GSP devices or the menu buttons for the GCD and GLD devices.

In bus operation mode (e. g.: one RS485 interface serves several axes), it must be ensured that each bus address is assigned only once. To avoid bus conflicts, caused by simultaneous response telegrams, the master computer must only transmit the next data telegram after receiving the ETX signal from the PSC and/or after a time delay.

It is possible to run PSCs of different device series, e. g.: IPP, GSP, GCD and GLD, within the same bus system.

As the preferential transmission rate, used by the IPCOMM, a bit rate of 28800 Baud<sup>1</sup> has been selected (applied to 8 bits/byte, 1 stop bit, no parity).

This bit rate is automatically set by IPCOMM. The value, already set by Windows®, will not be used.

If you intend to write PSC software of your own, a bit rate selection routine must be implemented<sup>2</sup>.

<sup>1</sup> The bit rate 9600 Baud can optionally be selected by software. With the installation of the IPCOMM software, the required auxiliary program BAUDRATE.EXE is also installed within the Phytron program group.

<sup>2</sup> For programming example refer to chap. 9.8.6 of this manual.

## 9.1 Structure of the Master-Slave-Telegram

The telegram format sent from a master computer to a PSC is defined as shown below:

<STX> | address | data bytes | separator | checksum | <ETX>

Example: Transmission of the motion command "Go 1000 steps relative ":

ASCII: | <STX> | 1 | G | R | 1 | 0 | 0 | 0 | : | 1 | F | <ETX>

HEX     02    31   47   52   31   30   30   30   3A   31   46   03

Character	Meaning
1	<STX> (Start of Text, 02 <sub>h</sub> ): a PSC can easily synchronize to this character because it is exclusively used as the start code for a new telegram.
2	Address of the PSC (slave), the range of the address byte is 0 to 9 and A to F (30 <sub>h</sub> ...39 <sub>h</sub> and 41 <sub>h</sub> ...46 <sub>h</sub> ). Additional the Broadcast <sup>1</sup> address @ (40 <sub>h</sub> ) is used.
3 to n-4	As usable data all ASCII characters with HEX code 20 <sub>h</sub> and above are allowed, except the colon : (3A <sub>h</sub> ), which is used as separator. Only capital letters must be used.
n-3	Colon (3A <sub>h</sub> ) as separator, to distinguish between usable data and checksum.
n-2	Upper byte of the checksum value (see below for the algorithm to calculate the checksum)
n-1	Lower byte of the checksum value (calculation see below)
n	<ETX> (End of Text, 03 <sub>h</sub> ), this code indicates the end of the telegram.

The checksum is defined by summing up all bytes, beginning with the address byte and including the separator (:) in an exclusive-OR-operation:

$$CS = \text{address} \oplus \text{data byte 1} \oplus \text{data byte 2} \dots \oplus \text{data byte n} \oplus \text{separator}$$

The checksum is calculated as one 8-bit binary value (00<sub>h</sub> to FF<sub>h</sub>). This byte is taken apart in its upper and lower byte (nibbles). After the HEX values of the two nibbles have been transferred to the corresponding two ASCII characters (0 to 9 instead of 0<sub>h</sub> to 9<sub>h</sub> and A to F instead of A<sub>h</sub> to F<sub>h</sub>, that means to each nibble 30<sub>h</sub> or rather 37<sub>h</sub> is mathematically added), the checksum is written in the telegram.

The PSC also calculates (Exclusive OR) the checksum of the received data. The telegram will be rejected if a difference to the received checksum is detected and an error bit is set in the status byte.

If there is no need to validate the contents of the telegram, the checksum monitoring can be set off. Instead of the checksum bytes, two X characters will be accepted, e. g.:

<STX> | 1 | G | R | 1 | 0 | 0 | 0 | : | X | X | <ETX>

<sup>1</sup> Broadcast: All axes, connected to a RS485 bus-system, receive and evaluate the telegram. To avoid bus-conflicts caused by the response of all axes nearly within the same time, the response of the controllers is suppressed by addressing with "@".



## 9.2 Structure of the Slave-Master-Telegram

The response telegram from the PSC to the Master computer is defined as:

<STX> | addr. | status | sep. | data | sep. | checks. | <ETX>

Example: Response to the motion command "Go 1000 steps relative", the PSC only transmits a status information. The response telegram does not contain further usable data.

ASCII: | <STX> | 1 | 0 | 1 | : | : | 0 | A | <ETX>  
 HEX     02   31   30   31   3A   3A   30   41   03

Character	Meaning
1	<STX> (Start of Text, 02 <sub>h</sub> ) as start code
2	Address of the corresponding PSC. The range of the address byte is 0 to 9 and A to F (30 <sub>h</sub> ...39 <sub>h</sub> and 41 <sub>h</sub> ...46 <sub>h</sub> ).
3 and 4	Short status, 8 bits of information coded as two ASCII characters (hexadecimal representation)
5	Separator (3A <sub>h</sub> ) to distinguish between status and usable data.
6 to n-4	Optional usable data. All ASCII characters with HEX code 20 <sub>h</sub> and above are applicable, except the colon : (3A <sub>h</sub> ), which is used as separator. Only capital letters are used in this protocol. If there are no usable data available, the PSC only transmits its internal short status followed by two separators ::.
n-3	Colon (3A <sub>h</sub> ) as separator, to distinguish between usable data and checksum.
n-2	Upper byte of the checksum value (see below for the algorithm to calculate the checksum)
n-1	Lower byte of the checksum value (calculation see below)
n	<ETX> (End of Text, 03 <sub>H</sub> ) this code indicates the end of the telegram.

The checksum of the response telegram is defined in the same way as for the transmit telegram.

All bytes, beginning with the address byte and including the 2 separators (:) have to be summed up in an exclusive-OR-operation:

CS = address ⊕ status byte 1 ⊕ status byte 2 ⊕ separator ⊕  
 data byte 1 ⊕ data byte 2 . . . . data byte n ⊕ separator

## 9.3 Handling of Transmission Errors

The block transfer of each data telegram is validated by the IPCOMM Protocol evaluating the checksum of each single block.

In case of disturbances during the transmission, the master computer (PC, PLC) decides whether a telegram, which was detected to be faulty should be transmitted a second time or rejected.

The overriding controller should also be able to detect missed response telegrams by time-out monitoring.

Error Source	Error indicator	Recommended Reaction
The transmission of the telegram from master to PSC has been disturbed, but the address byte has been recognized.	The PCS sends the status information with the status bit "Protocol Error" or "Command Unknown" set.	The Master-Slave telegram must be transmitted a second time.
The heading bytes (STX character and the address byte) of the MS telegram have been disturbed.	The PSC does not recognize any telegram.	The master computer must detect this failure by time out monitoring. The transmission has to be repeated.
The heading bytes (STX character and address byte) of the response (Slave-Master) telegram have been disturbed.	The master computer does not recognize the telegram heading bytes.	The master computer must detect this error by time out monitoring. The transmission should be repeated if necessary. Warning: The PSC status information has to be checked! If a relative motion command has to be repeated, it must be made sure by reading the position counter of the PSC whether the first command has been performed or not.
Slave-Master telegram is disturbed.	The master computer detects an incorrect checksum.	The master computer recognizes the incorrect checksum and repeats the last command or rather checks the internal status of the PSC.

## 9.4 Status Information of the PSC

In response to each received command, the PSC transmits a short status message to the master computer (exception: a Broadcast telegram was received). In case of a direct status request from the master computer, the PSC sends an extended status message consisting of the short status byte (located before the first separator) and the long status information (located behind the first separator). In all, there are 32 bits coded for the status information of each PSC device. The most important status bits are located in the short status byte.

### 9.4.1 Short Status

The short status is defined as an 8 bit information, whose upper and lower byte is transferred to two hexadecimal numbers.

The meaning of each single bit is defined as follows:

Bit ID (Decimal value)	Meaning	Comment
MSB 7 (128)	Cold start occurred	This bit is set during power on or a Reset has been carried out and is cleared by a status request.
6 (64)	Any Error	Summary of some other error codes defined in the 3 <sup>rd</sup> byte of the extended status code.
5 (32)	RX Error	Summary of some other error codes defined in the 2 <sup>nd</sup> byte of the extended status code.
4 (16)	SFI Error	Step failure has been detected (available only with optional SFI module).
3 (8)	Power stage error	Electrical failure of the power stage, e. g. low mains voltage, short circuit condition or high temperature.
2 (4)	Limit switch –	Status of Limit switch – 0=passive
1 (2)	Limit switch +	Status of Limit switch + 1=active (damped)
LSB 0 (1)	Motor runs	The motor is running or the recovery time has not been finished.

## 9.4.2 Extended Status

Requesting the extended status or rather the complete status (IS command), the master computer gets the internal status bit information of the PSC. This response telegram is assembled of one byte of short status and three bytes of extended status (coded as 6 hexadecimal numbers). If possible, the reading of the extended status also causes a reset of the error bits in the extended status. "Possible" means, that the error condition does no longer linger on.

The 2<sup>nd</sup> status byte: Interface status

Bit ID (Decimal value)		Meaning	Comment
MSB 7	(128)	Checksum	Checksum error has been detected
6	(64)		Not used
5	(32)	Overrun	Input buffer overflow
4	(16)	Not Now	Command not allowed just now. Example: While the motor is running no frequency changes of start/stop or running frequencies are allowed.
3	(8)	Unknown	Command unknown by the PSC.
2	(4)	Bad Value	The transmitted value is not permissible.
1	(2)	Out of Limits	The parameter is out of the permissible range.
LSB 0	(1)		Not used

Status byte 3: Additional status

Bit ID (Decimal value)		Meaning	Comment
MSB 7	(128)	No system	No firmware code available. For programming the system, appropriate download utilities are supplied by Phytron.
6	(64)	No ramps	There is no valid ramp table defined in the PSC. To program ramp tables, IPCOMM protocol can be used.
5	(32)	Parameter changed	An actual PSC parameter (e. g.: run frequency) has been changed. The bit will be cleared if the data have been stored in the permanent parameter memory.
4	(16)	Just busy	The PSC cannot respond to any command. The PSC is busy, e. g. reprogramming internal memory.
3	(8)	Programming error	Hardware fault, the flash eeprom seems to be defect.
2	(4)	Temperature	The power stage temperature has reached the warning level >80 °C. If the temperature increases by another 10 °C, the power stage will switch off.
1	(2)	Limit switch error	A limit switch failure has been detected and must be cleared by the CI command.
LSB 0	(1)	Internal error	Error occurred during calculating internal parameters.

Status byte 4: Additional status

Bit ID (Decimal value)		Meaning	Comment
MSB 7	(128)	Driver error	Overload or short circuit on 24 V-outputs detected.
6	(64)		Not used.
5	(32)	Wait for SYNC	PCS is waiting for the command "Start synchronous motion" after having received the motion command data.
4	(16)	Linear move	Type of motion: Linear
3	(8)	Free running	Axis is running without distance specification
2	(4)	Init successful	Initialization of the axis has been completed successfully and the reference point has been found. The bit is cleared if starting a new initialization.
1	(2)	HW disable	Only with GSP devices: The input X5.6 HW Enable is not activated. The connection between processor and power stage is interrupted.
LSB 0	(1)	Initialization is running	Initialization cycle of the axis has been started.

## 9.5 Definition of the Commands

The commands are sent within a master-slave telegram to the PSCs.

The commands are allocated to different groups, which are indicated by the first character of the command.

Code	Command Group	Command Meaning	refer to chapter:
B	<b>Break</b>	Motor stop with emergency stop ramp	7.1
H	<b>Halt</b>	Motor stop	7.1
G	<b>Go</b>	Execute motion command	7.2
P	<b>Parameter</b>	Read and change Parameter	7.3
I	<b>Info</b>	I/O and status information	7.4
C	<b>Clear</b>	Reset commands	7.5
F	<b>Flash</b>	Read PCS memory	7.6
W	<b>Write</b>	Write PCS memory	7.7
E	<b>Extension</b>	Hardware extension	7.8
R	<b>Retry</b>	Request for transmission repeat	7.9
S	<b>Special</b>	Special commands, due to software	7.10

The 2<sup>nd</sup> character is used as a command indicator within one group, from up the 3<sup>rd</sup> character the command string contains optional data. For reading the preset parameters, the 3<sup>rd</sup> character of the command had to be a "?". To change a parameter, the new value has to be transmitted within the command.

The valid range for the positioning data covers a binary value of 32 bits (Long Integer Format) and will always be presented as a decimal number with its operational sign (Maximum value: "GA-2147483647").

Straight direction commands, as for Initialization, Free Running and Single Step Mode are defined by the characters "+" and "-" (e. g. "GF+"). Binary coded parameters are transferred as hexadecimal numbers (e. g.: the current range to be set).

### 9.5.1 Stop Commands "H" and "B"

In the online modus a positioning motion or a free running task can be stopped by the "H" (Halt) command and also by the "B" (Break) command.

The stop command "H" terminates the motion task according to the programmed deceleration ramp ("PN" command), the stop command "B" uses the "emergency stop ramp" which is a multiple of the deceleration ramp. The factor is set by the "PH" command. When stopping the motion task by command "B" (emergency stop), the step width is fixed and no step width adaptation will be performed.

Both stop procedures are controlled, that means the value of the position counter is valid.

## 9.5.2 Motion Command “G”

The character following the “G” defines the type of the motion command.

Code	Command
GA	Absolute Motion
GR	Relative Motion
GF	Free running
GI	Initialization cycle
GS	Single Step Mode
GW	Beginning of the synchronous start sequence, refer to example
GX	Performing the synchr. start sequence, start of the axis, ref. to example
GB	Termination of an already running synchronous start sequence

Examples:

The following examples of the master slave telegram show only the usable raw data and the address. The obligate separators, checksum and delimiters (<STX>, <ETX>) have to be added:

Go absolute to position 1000	“1GA1000”
Go relative –200 in 1/8th step mode	“1GR–200”
Initialization start on limit switch +	“1GI+”
Initialization start on limit switch –	“2GI–”
One 1/8th step in positive direction	“1GS+” or “1GR+1”

Synchronous start sequence:

Preparing axis 1	“1GW”
Positioning value axis 1	“1GR1000”
Preparing axis 2	“2GW”
Positioning value axis 2	“2GR800”
....and the synchronous start command	“@GX”

Note:

The character “@” is used as the Broadcast address, that means all axes will receive and evaluate this telegram, but only the axes which have in advance got a clearing by a “GW/GR” command will perform the motion instruction.

To avoid bus conflicting caused by PSC response telegrams which would use the bus nearly at the same time, the master computer suppresses the response telegrams by sending the “@”.

## 9.5.3 Parameter Command

Parameters are set by the master computer by sending the new values within the “P” command or can be read by sending a “?” sign.

Code	Parameter	Range of Value	Default	Example
PA	Acceleration current (Boost current)	$0^2 \dots F_h^3$ (PAX.x GCD/GLD)	0	PAC (PA6.1)
PB	Reset to BIOS-Parameter	N. A.		
PC	Actual Position (Position Count)	32 bits	0	PC?
PD	Operation mode (Digital Mode)	0 = Online mode 1 = PLC mode	0	PD0
PE <sup>6</sup>	IPCOMM internal (Hardware configuration)			
PF	Run Frequency	$1 \dots 10000 (f_{max})$	2000	PF4700
PG	Limiter for axes, active only on free running and Initialization, stop if exceeding.	32 bits	1000000	PG?
PH	Emergency Halt Ramp	$0 \dots 250 \cdot PN$	0	PH5
PI	SFI Activation/ Deactivation (only optional)	0 = OFF 1 = ON	0	PI?
PL	Type of motion (Linear axis)	0 = rotating 1 = linear	0	PL1
PM	Offset from limit switch –(Minus)	$0 \dots 40000$	0	PM1600
PN	Ramp Number	$0 \dots R_{max}$	0	PN2
PO	Offset or Start/Stop frequency	$0 \dots 1250 (f_{max}/8)$	400	PO350
PP	Offset from limit switch+ (Plus)	$0 \dots 40000$	0	PP1200
PR	Run current	$1 \dots F_h^5$ PRx.x GCD / GLD	4	PR7 (PR3.9)
PS	Stop current	$0 \dots F_h^5$ PSx.x GCD / GLD	2	PS (PS0.1)
PT	Run current recovery time (msec)	$0 \dots 4000$	20	PT25
PU <sup>4</sup>	IPCOMM internal (User specific data)			
PX <sup>6</sup>	IPCOMM internal (Ramp data)			
PW	Compensation for Play	$-30000 \dots 30000$	0	PW0

<sup>2</sup> Zero inactivates the Boost mode, the run current is set also during acceleration/deceleration (ramps).

<sup>3</sup> The phase current of all controllers is set with 16 steps. These steps are corresponding to the different values of the motor current, which are specified in the single device documents. The current of the controller type GCD and GLD can as well be adjusted in steps of 0.1 A by using the format “PRx.x”. But current setting in the IPCOMM mode is also supported. The Read Out format “PR?” supplies the adjusted step, the format “PR??” supplies the exact value of the current.

<sup>6</sup> The IPCOMM uses the commands “PE”, “PU” and “PX” for depositing the information of the special DLLs which have been used for programming in the PSC.



Three copies of the parameter values are deposited in the PSC:

In the Program Memory (BIOS-Default in Flash-EPROM), in the Permanent Data Memory (serial EPROM) and in the Operating Memory (volatile RAM). These tables contain the ranges and standard values which will be valid after all parameters have been reset. On delivery, all parameters are in a defined state, the so-called BIOS-Default.

When the PSC has been switched ON or when a reset has been performed, the data are copied from the permanent memory into the operating memory.

During operation, all changes to the parameters only concern to the data in the volatile RAM and are only valid until reset. To change any parameters permanently, the write command "WP" has to be used.

Changing the parameters is only permissible when the axis is in the standstill and the run current recovery time has run out (PSC sends "standstill" with its status information).

### 9.5.4 I/O, Status and Info Commands "I"

By means of the command group "I" the master computer gets general information, for example about the internal state of the PSC. The character after the "I" specifies the type of the required information. All commands, except "IO" and "IZ", are inquiry commands.

Code	Parameter
IA?	Find <b>a</b> ddress of next free storage location for ramp
IB?	Read version ID of <b>B</b> IOS
IC?	Read date <b>c</b> ode of system software
ID?	Read date code of BIOS software
IE?	Read ID of <b>e</b> xtension board
IF?	Read maximum adjustable frequency <b>f</b> <sub>max</sub>
II?	Read Inputs, provides the status of the standard inputs
IN?	<b>N</b> umber of programmed ramps
IQ?	CPU Clock ( <b>q</b> uartz)
IS?	Read <b>S</b> tatus
IV?	Read <b>v</b> ersion ID of system software
IO	Read or set the standard <b>o</b> utputs
IZ	Read or set Delta <b>Z</b> ero

Note:

The commands “II” and “IO” only refer to the standard I/O, which have already been integrated in the first controller type IPP without extension board. All additional inputs and outputs are controlled by the extension commands “E” (refer to chapter 9.5.8).

## 9.5.5 Clear Commands “C”

---

By means of the Clear command, a single error status information can be reset. It is also possible to perform a hardware reset of the PSC by software instructions. The second character of the “C” command defines which hardware devices have to be reset.

Code	Parameter
CA	Power stage reset ( <b>A</b> mplifier), clear the error bit and initiates the Power stage reset.
CI	Limit switch error, <b>must</b> be confirmed with CI in any case.
CO	Reset <b>o</b> utput driver (24 VDC digital output), clears the error bit and initiates the reset signal for the output drivers.
CR	Causes a hardware <b>r</b> eset of the entire PSC.
CS	Initiates Reset of the optional SFI device (Step Failure Indication only available for IPP or GSP, not for GCD and GLD).

Notes referring to:

### “Power Stage Error”

The power stage error can be caused by different reasons: a short-time voltage collapse, an increase of the internal DC voltage or a short circuit. A short circuit is supposed, if the error bit cannot be cleared by the power stage reset.

### “Limit Switch Error”

- Both limit switches have responded simultaneously (e. g. the supply voltage of the switches has only been switched ON after the main power of the PSC or one limit switch has not been wired).
- A limit switch responds during positioning in the linear mode (in this operating mode the switches also function as terminal switches).
- When searching the reference position in the linear mode, the wrong switch has responded.
- A limit switch has been overridden, that means after standstill of the axis the signal is no longer active (the limit switches of a linear axis should be damped up to the end of the motion distance).
- No limit switch has been found within the limited positioning ranges.

### “Output Driver Error”

This error bit indicates that an output driver, caused by a short circuit or by overheating, has been switched off. Normally this failure can be cleared by a software reset instruction, sometimes it is necessary to switch the 24 V auxiliary voltage ON/OFF. If the error bit cannot be cleared, a short circuit is still present or the driver is damaged permanently.

## 9.5.6 Flash Programming Commands “F”

Using the Flash programming Commands presupposes the precise knowledge of the hardware configuration of the devices. It is presumed that the software modules, located in the Flash-EPRPOM of the PSC, will only be programmed by means of the IPCOMM configuration program. Because of this, the following commands as also the commands described in the next chapter are only explained for completion.

Code	Parameter
FA	Current chart <sup>7</sup> ( <b>A</b> mplifier)
FC	Flash Enable ( <b>C</b> lear)
FR	Read Ramp Chart
FU	Read PLC ( <b>U</b> ser) range, Format FUx. The program number (x) must be indicated as a hexadecimal number (0...F <sub>h</sub> ).
FF	Read serial EPROM block by block
FQ	Read Flash memory block by block

## 9.5.7 Write Commands “W”

Please read also the explanation of commands already described in chapter 9.5.6.

Code	Parameter
WA	Rewrite Ramp Chart as preset on delivery
WB	Break off programming
WD	Clear Ramp Charts and PLC user range
WE	Clear PLC user range
WF	Write block by block: serial EPROM
WP	Write actual parameter, located in RAM, into serial EPROM
WQ	Write PLC user range into Flash-EPROM block by block
WR	Write into Ramp Chart block by block
WS	Write into System range block by block

<sup>7</sup> The current table can only be read by the command “FA?”. The response is a string consisting of 16 hexadecimal coded digits each corresponding to one of the 16 adjustable current steps. Each value corresponds to the tenfold of the adjusted effective current according to the particular current step, e. g. the last current value of a type GSP 92-70 device is transmitted as 3F<sub>h</sub>=63<sub>d</sub>, which means that the maximum motor current is 6.3 Ampere.

Code	Parameter
WU	Write PLC user range, format WUXSTRING, x = Program number (0...F <sub>h</sub> ), STRING = PLC String command)
WX	EXIT Programming

## 9.5.8 Extended Commands “E”

The IPCOMM Protocol for managing the data transfer between a PSC device and the Installation Software IPCOMM has been developed in parallel with the controller device IPP and the Installation Software IPCOMM itself.

The IPP possesses four inputs and two outputs. An optional I/O Extension board with additional eight inputs and four outputs is available. The “E” commands are based on this hardware extension. These inputs are indicated as Program Selection Inputs.

For the later on developed controllers, GSP and GCD the compatibility to the IPCOMM software has been guaranteed and the used command groups are the same.

The GSP possesses 12 inputs and 6 outputs, the GCD 10 inputs and 4 outputs. The outputs “OUT0” and “OUT1” correspond to the two outputs of the IPP basic version, from up “OUT2” the outputs correspond to the additional outputs.

Code	Parameter
EA <sup>8</sup>	Reading and Setting of all 4 outputs of GCD/GLD EA? provides a hexadecimal number as response EAx sets the outputs (x = 0...F <sub>h</sub> )
EI	Reading of the Program Selection Inputs Prog.0 ...Prog n-1 EI? provides 8 bits binary data, coded as 2 hexadecimal numbers. GCD, GLD, GSP and IPP with extension boards have n=8 inputs, GCD and GLD with T-Option have n=4 inputs.
EO	Reading and Setting of the 4 additional outputs (GCD/GLD <sup>9</sup> 2 add. outputs) EO? provides one hexadecimal number as a response, EOx sets the outputs (x = 0..F <sub>h</sub> for GSP/IPP and 0 .. 3 for GCD/GLD). The both standard outputs are served with the command IO.
ER	Reading of PLC user sequences of the format ERxx. 256 user sequences are possible (IPP basic version only 7). The address' are xx = 00...FF <sub>h</sub> .
EW	Writing of the PLC sequences, format: EWxxSTRING, xx = Program number (00...FF <sub>h</sub> ), STRING = PLC String command)

<sup>8</sup> Only GCD and GLD devices.

<sup>9</sup> Not available in combination with the T-Modul, in this option these outputs are reserved for the DEGAS-process.

## 9.5.9 Retry Command “R”

The Retry command “R” instructs the PSC to repeat the last transmitted message.

## 9.5.10 Special Commands “S”

The command group “S” is used for extensions which do not generally concern every PSC. The following commands apply only to the controller devices GCD and GLD concerning in the first place the operating of the T-Module (Temperature measuring module).

Code	Parameter
SA?	Read power stage ( <b>A</b> mplifier) temperature
SC?	Read motor <b>c</b> urrent (present value)
SD	Read or set <b>D</b> elta Zero
SU?	Read internal $V_{DC}$
SW?	Read the adapted temperature sensor (the answer is “PT”, “KT” or “NO”)
SH? <sup>10</sup>	Start of DEGAS cycle
SM? <sup>10</sup>	Read motor temperature (a sensor must be installed in the motor)
SR? <sup>10</sup>	Read reference temperature (measured near the input connecting terminals)
ST <sup>10</sup>	Read and set the DEGAS time (minutes)
SI? <sup>11</sup>	Read internal reference voltage
SV? <sup>11</sup>	Read reference voltage of T-module
SX <sup>11</sup>	Read and set temperature 1
SY <sup>11</sup>	Read and set temperature 2
SZ <sup>11</sup>	Read and set temperature 3

<sup>10</sup> Only with T-Modul installed.

<sup>11</sup> Only with T-Modul installed and PT100 sensor connected.

## 9.6 Online Operation Mode of the PSC

---

Online operation mode (PD0) means that a master computer transmits the motion commands to the PSC by using a serial interface.

If it is intended to operate a PSC with a controller software of your own, there is no need to implement all commands of the IPCOMM Protocol. For plain applications the controller software can exist of only a few commands: The basis configuration of the axis can be performed by IPCOMM. The parameter of the axis are stored by the PSC and will become valid also after a reset.

To have an axis run, only a few commands are needed:

IS?	Read status of axis
GI+ and GI-	Go to limit switch (initialize)
GAXxxx	Go to absolute position
GRxxxx	Go relative
PNx	Set ID number of ramp to be used
PFxxxx	Set running frequency
POxxxx	Set offset frequency
CI	Clear limit switch error condition
CA	Clear power stage error condition

## 9.7 PLC Operation Mode of the PSC

---

PLC operation mode (PD1) means that a master computer transmits the motion commands to the PSC by using the digital inputs. In this operation mode handshaking with the master computer is performed through the digital outputs.

A reliable data exchange can only be guaranteed by using a definite protocol. The PLC has to wait for the READY signal of the PSC. After this, the program information can be put on the inputs and will be activated by a STROBE signal.

The PSC takes the program information and processes the command. The READY signal becomes passive.

In PLC operation mode the digital inputs have following function:

Input 0	STROBE: activates the prepared program information
Input 1	SW-ENABLE <sup>12</sup> : Must be active, so that the axis is released and an axis start could be performed. By inactivating the signal a running positioning can be stopped. The motor is stopped according to the programmed ramp and the position counter is valid.
Input 2	CLEAR ERROR <sup>12</sup> : Setting this input and activating the Strobe signal clears the error status of the PSC (limit switch or power stage error).The corresponding failure output of the PSC is cleared at the same time.
Input 3	RESET <sup>12</sup> : This input is corresponding to the function of a RESET push button. Setting this input and activating the STROBE signal, the PSC performs a RESET.
Prog x	The program select inputs can be used to select 1 of 256 user programs <sup>13</sup> out of the PLC program table. The inputs are interpreted binary. After the signals are steadily put to the inputs, the selected program can be started by activating the STROBE signal.

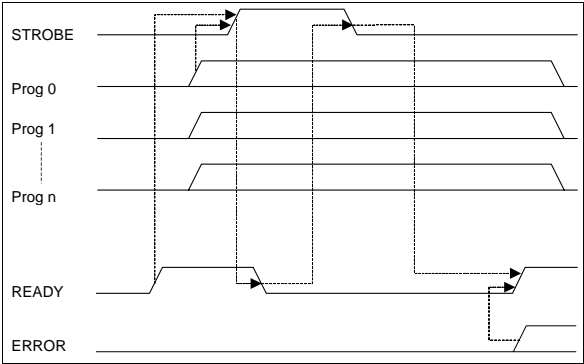
The outputs have following function:

Output 0	READY: The PSC is ready for receiving the next motion command. This signal is switched OFF when the motor starts and is only switched ON again when the motor is in the standstill <b>and</b> the run current recovery time has run out <b>and</b> the STROBE signal is deactivated.
Output 1	ERROR: There are error conditions, possibly caused by: <ul style="list-style-type: none"> <li>▪ The selected program has not been programmed</li> <li>▪ A limit switch error has occurred</li> <li>▪ There is a power stage failure</li> <li>▪ A step failure has been detected (only IPP/GSP with optional SFI)</li> </ul>

<sup>12</sup> Not valid for GCD/GLD and IPP(when only basically equipped). These controllers have only 4 inputs in all, the first input is used as STROBE, the 3 remaining inputs are used for program selection (1 of 8).

<sup>13</sup> For the basically equipped IPP are the number of 8 user programs and for GCD or GLD with the T-Modul installed are 16 user programs available.

The digital data exchange between PLC and PSC can be represented as in the following diagram:





## 9.8 Additional Information

The appendix contains additional information about IPCOMM protocol.

### 9.8.1 Tools

**IPS2000:** The program can be used to set the serial communication between the master computer and the PSC in operation.

This program is only available in German and it supports the following three commands:

- Suche!

The program requests the version over all control addresses from 0 to F at all comports COM1 to COM4. Furthermore, it uses the baudrates 9600 and 28800. If a control responds, the program displays the address, the firmware version and the used baudrate.

- Blinke!

By using this command, you can ascertain if the cable from the PC to the control is ok. The program sends every 250 ms on the chosen comports with a baudrate 9600 or 28800 alternative the characters <STX> and <ETX>. On the control the red LED "Status" is blinking.

If the command "Blinke!" is successful and the command "Suche!" does not find any control, it might be the hardware interrupt (IRQ) of the comport is not set correctly.

- LoopBack!

By using this command, you can ascertain if the comport of your PC is correctly configured. For this purpose you have temporary to bridge the connector pins 2 and 3, e.g. with a test adapter or with a wire.

On the chosen comport the character 'U' (55h) will be permanently send with a baudrate 9600 or 28800.

If everything is ok, the received number have to be equal to the transmitted number.

**OPM:** OPM is a independent software tool to record the project data. All parameters and PLC sequences will be read by the PSC and stored in a text file. The parameters will be written line-by-line according to the format of the IPCOMM-Protocol and could be edited if necessary. The edited parameter file can also be restored to the PSC by means of the OPM. (See also manual OPM).

## 9.8.2 Structure of Software

---

The programming language of the PSC software is “C”. The software consists of two parts: “BIOS” and “System”. Deposited in the BIOS are all communications procedures, upgrade and library procedures and the constants. The SYSTEM part is reserved for the effective motor controlling procedures.

These two software parts are placed in three of eight blocks of the permanent controller memory (block by block erasable Flash-EPROM).

The PLC sequences (user defined motion commands) are stored in a 4<sup>th</sup> memory block. In the PLC operation mode, these commands become valid by activating the corresponding digital inputs.

The remaining 4 blocks of the Flash-EPROM contain the ramp data. These tables are used to generate the motor pulse during acceleration.

Caused by the divided memory area, it is possible to reprogram parts of the memory when the PSC is already locally installed. Normally the reprogramming is performed when the PSC was set in operation for the first time. The IPCOMM configuration software, delivered with the PSC, can be used to program new ramps or new PLC sequences.

Sometimes it is also necessary to actualize the operating system of the PSC or to replace the operating system by a specific user defined version. Phytron provides for this purpose the corresponding download software. The software has been carefully established to the best of our knowledge. New program versions, which are due to program changes or hardware extensions, will also have been debugged to eliminate known malfunctions.

This manual is based on the software version 1.11 of GCD/GLD and version 1.55 of the GSP/IPP software.

Extensions which are due to the optional temperature module are implemented in the version of the GCD/GLD software.

As a new feature, the Emergency Stop Ramp was implemented to the GSP/IPP software version.

All software versions have an extended PLC functionality compared to former versions.

## 9.8.3 Configuration of the PLC Sequences

---

The standard PLC commands are exclusively motion commands. Start/stop frequency, running frequency, ramp and the type of motion will be written in one PLC command line.

Already during programming, the parameters, contained in the command line, will be converted to controller internal parameters. When starting one of these PLC sequences, the parameters will be loaded from memory. A few parameters must be calculated in advance and the start of the axis follows with only a few microseconds delay.

It is recommended to perform the programming of the PLC commands by means of IPCOMM or OPM configuration software. If it is necessary to change the PLC commands by means of your own software, the example below shows a command sequence, which can be used to change PLC commands.

The basic commands are:

- |      |                                |             |
|------|--------------------------------|-------------|
| 1.   | Read release code              | command IC? |
| 2.   | Enable flash programming       | command FC  |
| (3.) | Delete contents of Flash Eprom | command WE  |
| 4.   | Load new Flash Eprom code      | command EW  |
| 5.   | Start of programming cycle     | command WX  |

Caused by the block structure of the Flash-Eprom, the command WE erases the whole memory block which is used for the PLC sequences. If only free and unused memory addresses had to be newly programmed, it is not necessary to use the WE command.

Each EW command line must contain the specific data of a PLC command sequence. The information is coded identically with the online commands.

The data required are:

1. Number of the PLC program (memory location)
2. Start/stop frequency
3. Running frequency
4. Motion command
5. Ramp table index (ID number of ramp to be used)

Example: The PLC sequence for program 4 is to be redefined with

$f_{ss}$  = 100 Hz

$f_R$  = 1000 Hz

motion type = go relative

distance = 2300 steps

ramp ID = 3 (that means an internal given steepness of 37500 Hz/sec)

The corresponding EW command is: EW04 PO100 PF1000 GR2300 PN3

## 9.8.4 Extended PLC Sequences

To enlarge the functionality of the PLC commands, where its parameters were calculated during programming and stored within the command line, special commands of the format “\$OnlineCommand” have been added.

The special commands for the extended PLC mode can easily be programmed by IPCOMM V2. The programming of these commands in the controller can also be done locally with OPM.

Some of the normal online commands are used as special PLC commands:

Code	Command
\$PAx	Set boost current to value x (\$PAx.x for GCD)
\$PB	Overwrite actual parameters with predefined BIOS parameters
\$PCxxx	Set position counter to xxx
\$PFxxx	Set running frequency to xxx
\$PNx	Change ramp ID number
\$POxxx	Change start/stop frequency
\$PRx	Set run current to x (\$PRx.x for GCD)
\$PSx	Set stop current to x (\$PSx.x for GCD)
\$PT	Change run current recovery time
\$CA	Reset power stage
\$CI	Clear limit switch error
\$CO	Clear output driver
\$CR	Reset of PSC hardware
\$CS	Reset SFI (only GSP)
\$GAxxx	Go absolute
\$GI	Start Initialization
\$GRxxx	Go relative
\$SH	Start heating cycle of the motor (only for GCD)
\$&Nxx	Set program pointer (command in line xx <sub>H</sub> is next to be executed)
\$Txxx	Take a break of xxx msec

Besides the "\$OnlineCommand" exists the command of the format "\$&OnlineCommand". The & sign indicates that after the actual command line has been executed every next command line has also to be executed as long as the STROBE signal is active. In this way a "run off control" can be implemented on its minimum.

For illustration a part of an OPM file is shown below. The program sequence 8000 steps ahead, 8000 steps back, 1600 steps ahead, 1600 steps back, 2000 msec delay, jump to line 03H is repeated as long as the STROBE input is high (24 V).

```
; Phytron parameter file (Date: 26.06.2000)
; -----
;
; [comment]
; You may freely edit this file to add your own
comment.
; Any comment line must be preceeded by a semicolon.
;
; [GCD]
;
; Q32 F10
; BIOS_1.11 _P25051845_
; MK_1.11 _P25051850_
;
; [parameters]
;
PD1
PA0
PR3.4
PS0.8
PF2000
PG1000000
PH0
PL1
PM0
PN0
PO400
PP0
PT20
PW0
;
; [PLC sequences]
;
EW00$&PO250
EW01$&PF1500
EW02$&PN2
EW03$&GR+8000
EW04$&GR-8000
EW05$&GR+1600
EW06$&GR-1600
EW07$&T2000
EW08$&N03
```

## 9.8.5 Interface Communication

The following table shows an extract from the communication record between a PSC and the configuration program IPCOMM. It is an example for controlling the PSC by means of the serial interface.

Note: It is possible by using an option in the command line, to write a copy of the entire interface communication record on a hard disk of the master computer/PC. For this purpose the communication program IPCOMM has to be installed with the parameter: IPCOMM/D:S.

The destination file name of the data is IPC\_DSxxx, directory: TEMP.

Attention: The files can become very large, because IPCOMM is permanently reading the status of the PSC.

Command from Master to PLC	Response from PLC to Master	Comment
<STX>1B?:3F<ETX>	<STX>100:BIOS_1.04:62<ETX>	read out BIOS type
<STX>1IC?:3E<ETX>	<STX>100:_K05051043_:7C<ETX>	read out date code
<STX>1IV?:2B<ETX>	<STX>100:IPP_1.04:3C<ETX>	read out version ID
<STX>1IF?:3B<ETX>	<STX>100:10000:00<ETX>	read out maximum frequency (10000Hz)
<STX>1PG?:23<ETX>	<STX>100:10000000:30<ETX>	read out axis limit
<STX>1IO?:32<ETX>	<STX>100:0:01<ETX>	read out current state of digital outputs
<STX>1PL?:28<ETX>	<STX>100:0:01<ETX>	read out type of axis
<STX>1PD?:20<ETX>	<STX>100:0:01<ETX>	read out operation mode (0 = Online at PC)
<STX>1II?:34<ETX>	<STX>100:0:01<ETX>	read out digital inputs
<STX>1PF?:22<ETX>	<STX>100:5:04<ETX>	read out frequency of run
<STX>1PC?:27<ETX>	<STX>100:666:07<ETX>	read out current position
<STX>1PO?:2B<ETX>	<STX>100:400:05<ETX>	read out start/stop frequency
<STX>1PM?:29<ETX>	<STX>100:800:09<ETX>	read out offset to limit switch –
<STX>1PP?:34<ETX>	<STX>100:8000:39<ETX>	read out offset to limit switch +
<STX>1IS?:2E<ETX>	<STX>100:000008:39<ETX>	read out extended status
<STX>1PU?:31<ETX>	<STX>100:PSNORMAL 1.0.000:30<ETX>	(IPCOMM internal)

Command from Master to PLC	Response from PLC to Master	Comment
<STX>IIS?:2E<ETX>	<STX>100:000008:39<ETX>	read extended status: All is ok, one PLC program is stored in the memory but not activated (user area programmed, operation mode online)
<STX>IPX?:3C<ETX>	<STX>100:PRLINEAR 1.0.000:3F<ETX>	read out ramp table ID (linear ramps created with DLL PRLINEAR, Version 1.0.000)
<STX>IIN?:33<ETX>	<STX>100:10:30<ETX>	read out number of ramps in table
<STX>IFR0050:1A<ETX>	<STX>100:0032007D00FA017701F 4027102EE036B:47<ETX>	read out detailed ramp table information
<STX>IFR0060:19<ETX>	<STX>100:03E804B0FFFFFFFFFFFF FFFFFFFFFFFFFF:39<ETX>	..continue
<STX>IPN?:2A<ETX>	<STX>100:0:01<ETX>	read out "current ramp selection"
<STX>III?:34<ETX>	<STX>100:0:01<ETX>	read out digital inputs: Input 0 is active (1), inputs 1,2,3 are inactive (0)
<STX>IGR1234:1A<ETX>	<STX>101::30<ETX>	Motion command: Go relative 1234 decimal steps each 1/8 full step in positive direction
<STX>IPC?:27<ETX>	<STX>101:670:01<ETX>	read out current position
<STX>III?:34<ETX>	<STX>101:0:00<ETX>	read out current position and input status repeatedly
<STX>IPC?:27<ETX>	<STX>101:674:05<ETX>	
<STX>III?:34<ETX>	<STX>101:0:00<ETX>	
<STX>IPC?:27<ETX>	<STX>101:1603:34<ETX>	
<STX>III?:34<ETX>	<STX>101:0:00<ETX>	

## 9.8.6 Baud Rate Selection

As transmission speed, a bit rate of 28800 Baud (automatically preset by IPCOMM) or a bit rate of 9600 Baud can be selected.

The IPCOMM communication program offers a “check box” on its main user surface for adjusting the bit rate of the master computer to the bit rate of the PSC.

For changing the bit rate of the PSC, the auxiliary program “BAUDRATE.EXE”, which is part of the delivered IPCOMM software, must be used.

When using a PSC software of your own, make sure that a bit rate selection routine has been implemented.

Refer to the following programming example (TurboPascal):

```
THR          = $00;  { Transmitter Holding Register }
RBR          = $00;  { Receiver Buffer Register }
DLL          = $00;  { Divisor Latch, Least Significant }
DLM          = $01;  { Divisor Latch, Most Significant }
IER          = $01;  { Interrupt Enable Register }
IIR          = $02;  { Interrupt Identifier Register }
LCR          = $03;  { Line Control Register }
MCR          = $04;  { Modem Control Register }
LSR          = $05;  { Line Status Register }
MSR          = $06;  { Modem Status Register }
MR1          = $21;  { Interrupt-Maskenregister 1 }
MR2          = $A1;  { Interrupt-Maskenregister 2 }

DTR          = $01;
RTS          = $02;
OUT1         = $04;
OUT2         = $08;

type
  PariType    = ( None, xxxx, ODD, EVEN );
const
  Portnummer: byte   = 0;           { 0 für COM1 oder 1 für COM2 }
  Interrupt: byte    = 4;           { oder 3 für COM2 }
  COM_Port: word     = $03F8;       { oder $02F8 für COM2 }
  Baudrate: longint  = 28800;
  Paritaet: PariType = None;
  Wortlaenge: word   = 8;
  Stopbits: word     = 1;

.....

procedure SetUART ( Baudrate:longint; parity:PariType; wlength, stops: byte );
{-----}
begin
  CLI;
  port[LCR+COM_Port] := $80;
  port[DLL+COM_Port] := lo(word(115200 div baudrate));
  port[DLM+COM_Port] := hi(word(115200 div baudrate));
  port[LCR+COM_Port] := (wlength-5) or ((stops-1) shl 2) or (ord(parity) shl 3);
  port[IER+COM_Port] := $01;
  port[MCR+COM_Port] := Port[MCR+COM_Port] OR $0B;
  if IRQ_Nummer < $10 then Port[MR1] := Port[MR1] AND ((IRQ_Maske) XOR $FF)
    else Port[MR2] := Port[MR2] AND ((IRQ_Maske) XOR $FF);

  Dummy := Port[LSR+COM_Port];
  Dummy := Port[RBR+COM_Port];
  STI;
end;
```



## 10 Index

### A

Actual position 15  
 Adapter cable 26  
 Axis status 15, 16

### B

Baud rate 54  
 Bit rate 29  
 Broadcast 30

### C

Cabling 25  
 CD ROM 6  
 Checksum 30  
 COM Interface 25  
 Command code 23  
 Commands  
   Clear command 40  
   Extended command 42  
   Flash programming command 41  
   I/O command 39  
   Info command 39  
   Motion command 37  
   Parameter command 38  
   Retry command 43  
   Special command 43  
   Status command 39  
   Stop command 36  
   Write command 41  
 Compensation for play 15  
 Copyright 2  
 Current delay time 14  
 Current setting 14

### D

Default 14  
 Defect cabling 25  
 Device address 9  
 Device type 14

### E

Error 15  
 Extended status 34

### I

Installation program 6

Installationsprogramm 6  
 Interface problems 25  
 IPCOMM-Icon 10  
 IPS2000 8

### L

Limiter for axes 15

### M

Main features 5  
 Master computer 30  
 Motion commands 13, 18, 20

### O

Online operation mode 44  
 Operating system 5  
 Operation mode 14  
 OPM 8, 47  
 Optional add-on-card 22  
 Output drivers 17

### P

Parameters 14, 22  
 Pilot lamp 15, 16  
 PLC operation mode 44  
 PLC protocol simulation 18  
 PLC sequences 48  
 Programm installation 5

### R

RAM 5  
 Ramps 14, 21  
 Range register 12, 15  
 Reset 17, 18  
 Response telegram 31  
 RS Monitor 23  
 Run frequency 14

### S

Selection menu 9  
 Send telegram 30  
 Separator 30  
 Setup.exe 6  
 SFI 8, 16, 17, 24  
 Short status 33  
 Start 4  
 Start/Stop frequency 14

# Manual IPCOMM

---

Status line 9  
Structure of software 48  
System requirements 5

**W**

Windows 4

Working environment 9

**Z**

Zero point offset 15



**Phytron GmbH • Industriestraße 12 • 82194 Gröbenzell, Germany**  
**Tel. +49(0)8142/503-0 • Fax +49(0)8142/503-190 • E-Mail [info@phytron.de](mailto:info@phytron.de) • [www.phytron.de](http://www.phytron.de)**

**Phytron, Inc. • 600 Blair Park Road Suite 220 • Williston, VT 05495 USA**  
**Tel. +1-802-872-1600 • Fax +1-802-872-0311 • Email [info@phytron.com](mailto:info@phytron.com) • [www.phytron.com](http://www.phytron.com)**