Operation instructions manual

ComPro

Project planning and diagnostic

DOS program

Hilscher Gesellschaft für Systemautomation mbH
Rheinstraße 15
D-65795 Hattersheim
Germany

Tel. +49 (0) 6190/9907-0
Fax. +49 (0) 6190/9907-50

Sales: +49 (0) 6190/9907-0
Hotline and support: +49 (0) 6190/9907-99

e-mail: hilscher@hilscher.com
Web: http://www.hilscher.com

Index	Date	Version	Chapter	Revision
1	1.12.93	1.00	all	draw up; based on V 1.00 of the German manual
6	13.09.95	2.12	all	draw up; based on V 2.11 of the German manual timestamp using with trace. Not compatible to previous versions of ComPro
7	21.02.96 13.06.96	from 2.20	all all	Improved system conditions Expanded menu ONLINE with submenu driver Expanded menu ONLINE-SYSTEM with submenu device
8	21.11.96	from 2.224	 all	New command line parameter /?, /H and /HELP Some small additional features Bug fixes Spelling corrections
9	26.10.97	from 2.224	1, 2	Using the program ComPro under Windows NT

Although this program has been developed with great care and intensively tested, Hilscher Gesellschaft für Systemautomation mbH cannot guarantee the suitability of this program for any purpose not confirmed by us in writing.

Guarantee claims shall be limited to the right to require rectification. Liability for any damages which may have arisen from the use of this program or its documentation shall be limited to cases of intent.

We reserve the right to modify our products and their specifiactions at any time in as far as this contributes to technical progress. The version of the manual supplied with the program applies.

1 General

The 'ComPro' program is the configuration and diagnostic program for all equipment supplied by Hilscher.
The configuration data for the devices are stored in a database on the PC. With the aid of ComPro, this database can be processed and transferred to the target hardware.
Extensive status functions allow the user to monitor the device status at any time and change it where necessary. In addition, ComPro includes functions for supervision and manipulation of message traffic.

The databases have their own format, and can only be processed with ComPro. They are stored in the COMPRO directory as '*.DBM' files.

1.1 System conditions

The PC should have a main memory capacity of 2 MByte and a free disk memory capacity of 1 MByte. It is recommended to have a 386 as a processor or larger. It is a executable program in a DOS-Box for Windows and OS/2.

Following points are mind by configuration at the PC:

- Every PC should have at least 510 kByte DOS memory free.
- By using of ComPro with a CIF card the memory manager (QEMM, 386MAX, EMM386 etc.) must exlude a memory utilization of 2 kByte for **every** CIF card.
  (for example: QEMM => X=CA00-CA7F by CIF-Basicaddress=CA00:0000).

It will be checked how many memory is available while editing the data base. If there is no more memory available there will be displayed an error notifiction. Use in this case the external data base editor DBEDIT.EXE.

1.2 Installation and program start

To install ComPro, make the drive with the installation disk the current drive. Then starte the installation program INSTALL.EXE and follow the menue.

To start ComPro, switch to the '*target directory*' and enter

CPRUN <*RETURN*>

1.3 DOS-adjustment for OS/2

Following DOS-adjustments for OS/2 are to use:

parameter	adjustment
COM_DIRECT_ACCESS	ON
COM_HOLD	ON
COM_SELECT	COM1 or COM2
DOS_BACKROUND_EXECUTION	ON
DOS_HIGH	ON
DOS_RAMSIZE	640
(VIDEO_MODE_RESTRICTION	CGA)

DOS-adjustment for OS/2

1.4 Windows NT

The program ComPro can only run under Windows NT using a serial connection between the PC and the CIF for parameterization or diagnostic purposes.

The requirements are that the CIF has a diagnostic interface and a RS232 cable. The you can connect via the RS232 cable COM1 of the PC to the diagnostic interface of the CIF. Alternatively you can use COM2 of the PC.

The necessary parameter for starting program ComPro is described in chapter *The command line* in section *Parameter for the CIF* in this manual.

The pinning of the RS232 cable is desribed in the device manual of the device. With order number KAB-SRV this cable is available.

2 The command line

The batch ile stored all specific ComPro parameters for all different devices!

As ComPro can be installed for all our equipments (PKV, KPO, CIF and so on) equally. Because of this the various parameters have to be entered when the program is started. The program must know the connected device. **For every device we deliver a batch file called 'CPRUN.BAT', in which your settings are stored.**

2.1 General parameters

parameter	meaning
/? /H /HELP	Displays all command line parameters
/B:Keyfile	Name of the keyfiles (Macros), automatic execution by starting the program
/D:Datenbank.DBM	Name of the database, automatic loading by starting the program. The database must be installed in the COMPRO directory.
/MONO	Independent of the used graphic adapter the monochrome mode switched on.
/RT:xx	set up device basis address (see SETUP).
/CB:xx	set up ComPro basis address (see SETUP).

General call parameters

2.2 Parameter for the PKV and the KPO

parameter	meaning
/S:1	Selection of the PCs serial interface COM1 for the connection to the device diagnostic interface.
/S:2	Selection of the PCs serial interface COM2 for the connection to the device diagnostic interface.

Call parameters for PKV and KPO

2.3 Parameter for the CIF

parameter	meaning
/A:CIF-SEGMENT	Memory address of CIF jumpers on the Hardware side. For example: */A:CA00*
/S:1	Selection of the PCs serial interface COM1 for the connection to the device diagnostic interface.
/S:2	Selection of the PCs serial interface COM2 for the connection to the device diagnostic interface.
/DPM:xx	Size of the dual port memory in kByte. Possible parameters are 1, 2, 4, 8, 16. Default-parameter is 2 kByte.
/T:Data base.DBM	Name of the data base, which automatically downloaded on the CIF at start time. The data base must be in the COMPRO-directory.
/R:1	Force a warm start on the CIF.
/R:2	Force a cold start on the CIF.

Call parameters for CIF

The program ComPro has access to the CIF over the dual-port memory. The parameter is /A: and the segment address for example /A:CA00. This parameter is used in file CPRUN.BAT as default parameter.

As an alternativ a serial connection between ComPro and the CIF is usable, if the CIF has a diagnostic interface. The parameter for ComPro is in this case /S:1 (for COM1) or /S:2 (for COM2). To use a serial connection the parameter /A: does not be used anymore. The batch file CPRUN has to be modified to COMPRO /S:1 or COMPRO /S:2.

To build up a serial connection between CIF and PC a RS232 cable is necessary.

3 Description of ComPro functions

The ComPro main window consists of the menu line and the protocol window. The protocol window notes all the actions by the user, thus providing full traceability of all user actions and how the connected system or ComPro itself reacted to them.

User dialogue box:

'F10' key: jump to <ok-enter>

'ESC' key: jump to <cancel-esc>

In a user dialog box, the fields <ok-enter> und <cancel-esc> are permanently available. The <ok-enter> field can be reached with the 'F10' key, and the <cancel-esc> field with the 'ESC' key. Pressing these keys does **not** however terminate the dialog, but merely places the cursor on the relevant field. In order to terminate the dialog, the selected field has to be further acknowledged with *<enter>*.

The program is structured in six parts:

- *database* Off-line configuration and parameterization of the link (*edit, save, load, print*)
- *online* On-line configuration and status functions (*system, task, database, software, driver*)
- *diagnostic* Diagnostic functions (*filter, trace, protocol, monitor*)
- *setup* setting the address *(address, password)*
- *macro* macro functions *(recording, playing, delay)*
- *exit* Terminate ComPro session.

3.1 Offline processing of a database - *database*

The database records are the specific configuration data of a device. This is a file with the ending *.DBM, it is kept in the COMPRO directory. Thus it is possible to manage all configuration datas for several devices with one ComPro.

Every database subdivides serveral database tables, they contain information about the whole system (device hardware, device firmware and ComPro). With the database ComPro can adapt to the connected devices.

The first menu option provides all the functions for offline processing of a database.

- *edit* Editing of a database file
- *save* Saving a database to the hard disk
- *load* Loading a database from the hard disk
- *print* Printing a database table to a printer or a file

3.1.1 Editing a database - *edit*

A internal special database editor is provided for editing of configuration data. The editor checks all entries, and only permits the user to exit from an input field when the values entered in that field are within the valid range. The value range for the field concerned is shown in the last line of the editor window. All entries made are only stored in the PC's internal database when the <ok> field has been selected and acknowledged. If the user exit the editor with <cancel>, all entries are cancelled. There is no safety prompt!

By selecting of the menu option 'edit' all available tables of the loaded database were displayed. By means of <*cursor up*> respectively <*cursor down*> can be choosed a table of a list and opened with the <*enter*> key.

The key assignment in the editor is as follows:

- *TAB* - one field to the right
- *shift-TAB* - one field to the left
- *cursor up* - one field up
- *cursor down* - one field down
- *grey +* - insert protocol (not available on every tables)
- *grey -* - delete protocol (not available on every tables)
- *escape* - Cursor to <cancel> field
- *F10* - Cursor to <ok> field

3.1.2 Saving a database - *save*

A database from the hard-disk would always saved in topical directory. The file is stored with the extension 'DBM'.

A selection of the files for existing databases is displayed. The database can be saved by a existing or a new name.

By selection of a name from the list the current database will saved to the hard-disk by this name. The old database from the hard disk will be over written, after creating a **.BAK*-File.

If the database is to be saved with a new name, select the 'new' option. Then enter a file name with which the database is to be stored in the current directory.

The database is only saved when the entry is confirmed by pressing *'Enter'*.

3.1.3 Loading a database - *load*

A selection of the files for existing databases is displayed. If one of these databases is to be loaded, the name should be selected with *'cursor up'* or *'cursor down'* keys and *'enter'* pressed to confirm.

If there is a database in RAM which has not been saved, a prompt first appears enquiring whether the current database should be saved before a new database is loaded. If this prompt is answered with *<cancel-esc>*, the database in RAM is cancelled.

3.1.4 Printing a database - *print*

There is a facility for printing out a database table as formatted text respectively printing out to a file. First select a table of a list. The name of the file is stored with the extension **.PRN*. The default line length is 80 characters and the default page length 65 lines.

3.2 The online functions - *online*

The functions under this menu option directly influence (online) the connected device (CIF, PKV, KPO etc.). If an online function is activated and no device is connected, an error message is generated.

The menu option is subdivided into four submenus:

- *system* Reading of equipment configuration, firmware version, display of the modules, reading of global status, reading and setting the time of the device, coldstart, warmstart and bootstart.
- *task* Display the status of all tasks, display or edit the status of individual tasks, display the task versions.
- *database* Overview of the existing segments, transfer of the data base between the connected device and the ComPro, deletion of individual database segments.
- *software* Overview of the existing segments, loading of the device firmware or individual device functions (e.g SPC module) into the connected device.
- *driver* Switch driver free and display licensed drivers.

3.2.1 The device system functions- *system*

3.2.1.1 Device information - *device*

The device number (GNR), the serial number of the device (SNR) and the manufacturing date will be displayed.

name	display
date	21.02.1996
device number	9 2 0 9 0 5 4 0
serial number	0 0 0 0 0 5 4 3

Device information

3.2.1.2 Display of the firmware version - *firmware*

This function can be used to read out the name and the version of the firmware running on the device and its checksum.
Firmware indicates the file name of the firmware loaded and the device.
The date in the version is identical to the file date.

name	display
firmware	MODBUS CIF10/11
version	V1.234 01.02.95
checksum	0X1234

A firmware display for example

3.2.1.3 Display of the device configuration - *configuration*

This function is used to display the configuration of the connected device. Following selection of the *configuration* menu, the operator can choose between *hardware* and *software*.

On selection of the hardware configuration, the following table appears. The *values* are example values.

The unit code contains the file extension of the firmware.

parameter	display
unit code	H41
processor type	80C188
ram type	1 x 32kByte RAM
res. memory type	no present
boot memory	1 x 128k-FLASH (29F010)
SCC-count	2
system frequency	16 MHz
realtime clock	no present
software number	0
SCA-type	no present

A hardware configuration for example

On selection of the software configuration, the following table appears. The *values* are example values.

parameter	display
cycletime [ms]	2
max. taskcount	8
segment count	31
segment size	288
message size	255
RCS-version	01.100
device address	0
first SW number	0
cnt. SW-number	0

A software configuration for example

3.2.1.4 The device system functions - *modules*

This function can be used to display the modules composing the firmware. The firmware is composed of RCS, LIB and MCL modules. Together with the module name, the version, start address and status of the module are also displayed.

3.2.1.4.1 List of RCS modules

The modules of the RCS operating system present in the firmware are shown in a list.

number	name	version	start address	state
01:	RCS_SYS	1.007	D2AD:2CED	ok.
02:	RCS_FNC	1.006	DDAD:3CC3	ok.
03:	RCS_COM	1.006	D78A:622C	ok.
04:	RCS_SCC	1.007	E17D:2819	ok.
05:	RCS_DBM	1.001	E402:0175	ok.
06:		0.000	0000:0000	
07:		0.000	0000:0000	

List of RCS modules for example

3.2.1.4.2 List of LIB modules

The library modules present in the firmware are shown in a list.

number	name	version	start address	state
01:	LIB_SOR	1.003	F324:0038	ok
02:		0.000	0000:0000	
03:		0.000	0000:0000	
04:		0.000	0000:0000	
05:		0.000	0000:0000	
06:		0.000	0000:0000	
07:		0.000	0000:0000	

List of LIB modules for example

3.2.1.4.3 List of MCL modules

The communications modules present in the firmware are shown in a list.

number	name	version	start address	state
01:	MCL_KPO	2.002	E5A9:0048	ok.
02:	MCL_SCC	1.006	E4FE:071E	ok.
03:		0.000	0000:0000	

List of MCL modules for example

3.2.1.5 The device system function - *state*

The states of global variables are displayed. The number of the segments in the system, the free RAM length and the number of running tasks are also displayed. The idle task of the operating system is also counted.

free segment	29
free ram length	3.248
running task	2

Status display for example

3.2.1.6 Setting the system time - *time*

The device system time is not available on every device.

The system time is read out and can be changed.

3.2.1.7 Cold start - *coldstart*

A cold start is practicable. It is equivalent to switching the device off and on. During a cold start, all the data in RAM are deleted and the system was started new. A cold start interrupt the communication.

There is a safety prompt before this action is executed.

3.2.1.8 Warm start - *warmstart*

During a warm start, the hardware is reinitialized. In contrast to a cold start, the current parameters are retained. The system was started new with the current parameters. A warm start interrupts the communication.

There is a safety prompt before this action is executed.

3.2.1.9 The initial start - *bootstart*

On an bootstart, the connected device is placed in its booting condition. In this condition, the device only accepts the commands load firmware, database overview and configuration.

The system is reset and the initial program loader is activated without any possibly present firmware being started.

There is a safety prompt before this action is executed.

3.2.2 The task functions - *task*

The activities of the tasks on the device can be monitored.

3.2.2.1 Reading the version data - *version*

The version data for all tasks running on the connected device are displayed.

	name	version	prio	startidx	state
00:	RCS	1.000	0	0	ok
01:	TASK1	1.000	1	1	ok
02:	TASK2	1.000	2	2	ok
03:					
04:					
05:					
06:					
07:					

task - version for example

Prio indicates the priority of the task, and *Startidx* the index of the task.

3.2.2.2 Displaying the task status - *status read*

The call up of select status is dependent on the device software and not always possible. Please consult your protocol description.

On selection of the submenu *all* the current status of all tasks on the connected device is displayed online. This can be used to check what action the individual task is currently performing. In addition, it is possible to determine whether there is a connection between the diagnostic PC and the device.

On selection of *all*, the window appears with the task numbers, the individual task names and the status of the relevant task. The task names and task status indicated in the following table are intended as examples only.

number of task	task name	task state
00:	RCS	running
01:	TASK1	send
02:	TASK2	ready
03:		
04:		
05:		
06:		
07:		

Taskstate for example

In addition to the *all* status for the device, each task provides one or more individual states. These individual states contain, among other data, send, receive and error counters, and further task specific status information. Further details on the task status can be found in your protocol description.

Apart from the 'read only' task status, some devices also have writable task status fields. These are displayed on the left hand half of the screen, with the indication that edit mode can be entered with key '*F9'*. The values displayed can then be edited in the edit window and transferred to the connected device. The user must indicate which values in the status field are to be written to the hardware. Selection takes place by setting the selection field with the '*space bar*'. On confirmation of the dialog with <*ok*> the data are transferred. It is to be noted that no current status information is displayed in the left hand status field during edit mode.

3.2.2.3 Resetting the task status - *status reset*

With this function, status fields for individual tasks or all tasks can be reset.

3.2.3 Device configuration - *database*

In most devices, the configuration data are stored in a flash EPROM, protected from zero voltage. A number of CIF cards which store the configuration data in a dual port memory are an exception.

The flash EPROM is divided into several segments. These segments are used by the firmware for different purposes. The segmentation is specified by the firmware and can only be altered by exchanging the firmware EPROM.

3.2.3.1 Overview of the database segments - *overview*

With this submenu, the segmentation of the flash EPROM and the status of the individual segments can be viewed.

Segments of type *DBM* are used for storage of configuration data. Segments of type *etc* (for 'etcetera') form add on segments and thus enlarge the previous segment. This is the case, for example, with large configuration databases. The segmentation is specified by the firmware and can only be altered by replacing the firmware. The current length of the first entry is the total length.

	name	type	mode	address	max. len	act. len	state
0	PROFIBUS	DBM	0x00	4000:0000	16.384	12.230	ok
1			0x00	0000:0000	16.384	0	ok
2			0x00	0000:0000	16.384	0	ok
3			0x00	0000:0000	16.384	0	ok
4			0x00	0000:0000	16.384	0	ok
5			0x00	0000:0000	16.384	0	ok
6			0x00	0000:0000	16.384	0	ok
7			0x00	0000:0000	16.384	0	ok

Overview for example

Segment errors can occur if an illegal database or no database at all is loaded. An incorrect flash module or a defective configuration can also lead to errors. Incorrect configuration data such as incorrect baud rate or parity, etc. are then not detected!

The maximum permissible length of the configuration data is determined from the tables displayed. If an attempt is made to load a longer database than permissible into the device, this is acknowledged with an error message on downloading. The configuration data in the device are then invalid!

3.2.3.2 Transferring the configuration data into the device - *download*

Transfer of the database into the connected device is activated in this submenu. All the tasks in the device have to be stopped for transfer of the database. This is effected by the ComPro after a safety prompt. Transfer is then initialized and the data then downloaded. The device is then re-initialized. The parameters from the database transferred are accepted by the firmware as new parameters.

Prior to downloading, the ComPro checks whether the database loaded is suitable for the device firmware. If the database and device firmware are incompatible, error message *Message (53): RCS_FLASH_FILE* appears.

3.2.3.3 Reading the configuration data from the device - *upload*

The database is transferred from the connected device to the PC. The function of the device remains unaffected. If there is a database in the PC memory which has not been saved, the user has an opportunity to save this prior to uploading.

3.2.3.4 Deletion of the configuratoin data in the device - *delete*

Individual segments in the flash EPROM can be deleted with this function. Depending on the firmware, the device will stop operation immediately, or following the next warm start/cold start at the latest.

3.2.4 The device system functions - *software*

3.2.4.1 Overview of the firmware modules - *overview*

With this submenu, the segmentation of the flash EPROM and the status of the individual segments can be viewed.

3.2.4.2 Modification of the firmware modules - *program load*

With this function, an individual module is downloaded from the PC into the connected device. Individual functions of the firmware can thus be systematically modified or adapted.

There is a safety prompt before the action is executed.

3.2.4.3 Loading the firmware modules - *firmware load*

This function downloads the complete firmware from the PC into the connected device. It is dependent on the firmware being stored in a flash EPROM in the device.

This function allows firmware modification to be performed at any time.

3.2.5 Driver activation - *driver*

3.2.5.1 Display drivers - *display*

With this menu item all licensed drivers will be displayed.

3.2.5.2 Activate driver - *activate*

It appears the request to enter the activation code. The input of the code is in hexadecimal description. The code can have a length up to 16 bytes and is only for licensed drivers.

First, licensed drivers must be acquired from us. Therefore we need the serial number of the device hardware (CIF communication interface, COM Communication module). Then the activation code is spent by us. Finally, enter the activation code to free the driver.

3.3 The diagnostic functions

The diagnostic functions described below can be used for monitoring and manipulation of the function of the communications device.

3.3.1 The trace filter - *filter*

The filter dialog provides for an indication of what information the tasks should enter in the trace buffer. The trace filters can be activated individually for each task or globally for all tasks.

With the activation of the trace filters, the selected tasks enter the required information in the trace buffer. This information can be displayed using the submenu *trace.*

To activate a trace filter, an *event* and an *entry* are to be specified. Under *event*, the user sets the conditions under which a task is to effect an entry in the trace. Under *entry*, the user selects the extent to which the task stores data in the trace.

Specification of the event, on the basis of which entries are made:

- *off* no entries
- *send/receive* on sending and receipt of a message
- *sending* on sending of a message
- *receiving* on receipt of a message
- *error* on occurrence of an error
- *always* on any event

Specification of the information to be entered in the trace buffer (Entry):

With some protocols, no distinction is made between header and user data, and therefore the settings for header data/user data/data are identical.

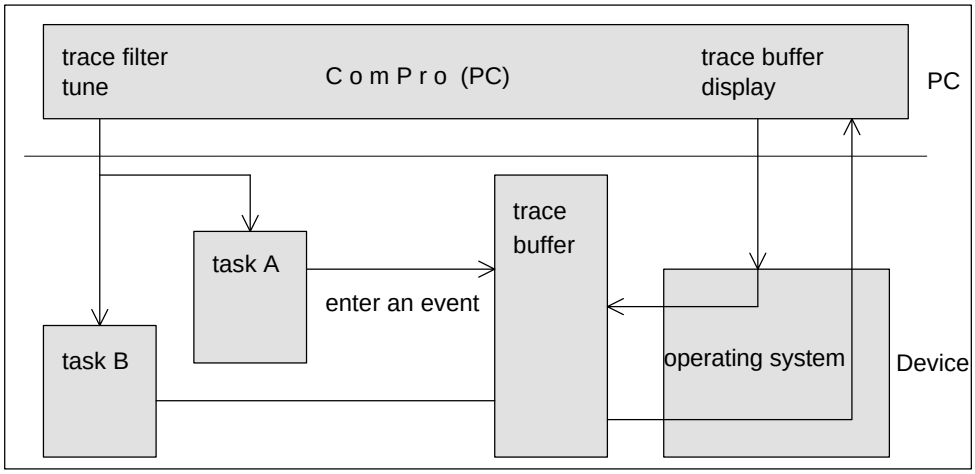
- *off* no data
- *telegram header* Only the header data of the telegrams transmitted/ received
- *telegram data* Only the user datas of a telegram transmitted/received
- *telegram* Only a telegram transmitted/received with its user and header data
- *message header* Only the header data of a system message input or output
- *message data* Only the user data of a system message input or output
- *message* A system message input or output complete with its user and header data

3.3.2 The trace - *trace*

The setting of which task enters signals in the trace buffer on what event is effected with the FILTER submenu.

The diagnostic functions of the ComPro are used to test a link and to locate errors. For this purpose, the device has a trace buffer, in which the tasks enter certain data on occurrence of an event. **The entries are also made when the ComPro is not connected!** Entry of data in the trace buffer is started by the setting of a trace filter.

The operating system on the device transfers the contents of the trace buffer to the ComPro when the *Trace* submenu has been selected and the subsequent prompt answered. As the trace buffer is designed as a ring buffer, a buffer overflow may occur. This means that the trace protocol is incomplete. A trace buffer overflow is displayed on reading by ComPro.



Course of the trace function

Following selection of the trace submenu, there is a prompt as to whether the contents already stored in the trace buffer are to be displayed.

Trace buffer **output** is activated with <*ok*> without previously deleting the trace buffer.

An entry of <*no*> also activates the trace buffer output. In this case, however, all trace entries, i.e. the entire trace buffer, are deleted! The entries from the tasks which then follow are output.

Stopping acts on the output only, and trace buffer entries continue to be made by the tasks.

All entries in the trace buffer are transmitted to the PC and displayed. Trace buffer output is terminated with '*esc*'. The **Output** can be stopped with the key combination '*ctrl-S*' and startes again with '*ctrl-Q*'.

3.3.3 Recording trace outputs in a file - *protocol*

On activation of the protocol function, all trace buffer entries displayed are recorded in a file. On opening of a protocol file, any existing file of the same name is deleted! The file created has the name <*filename*>.*PRT* .

The protocol file is an ASCII file and can be viewed with any ASCII editor and with the DOS command *type* <*filename*> .

If recording is active, repeated selection of the submenu activates the recording function.

3.3.4 The message monitor - *monitor*

With the message monitor, you have the opportunity to intervene actively in the working sequence of the device software by sending messages to individual tasks and receiving messages from them. For this purpose, you can use the message monitor to compile, edit, save, load and of course send and receive messages.

Each task on the device 'understands' certain commands, which are transmitted to it in the form of a message. On receipt of a command, the task will execute and acknowledge that command. The transmission of a command to a task is initially independent from the sending of a message via an interface. Each interface task does however know commands which cause it to send a message.

The task commands can be found in the manual on the protocol.

The structure of a command is dependent on the task which performs it. Please consult the manual for the protocol for the command supported by your software.

3.3.4.1 The windows of the message monitor

The monitor screen is divided into four windows. The two left hand windows are output windows for the messages received (receive windows), and the two right hand windows are used to enter, edit and send messages (send windows). Messages in a send window can be saved to the hard disk and loaded again.

Messages are always stored in the subdirectory *MSG*. This directory must be created before messages are saved.

Both windows are divided into message headers and message data. The message header has a fixed structure, which is defined as follows:

Rx Receiver of the message

Tx Sender of the message

Ln Length of the utilization data (calculated!)

Nr Message number for identification of the message

A Answer identifier

F Error identifier

B Command identifier

E Extension

The contents of a message are described in detail in the manual on the protocol. Entry and display are hexadecimal.

3.3.4.2 The functions of the monitor

Key	Function
'Esc'	exit from the monitor
'TAB'	toggle between message header and message data windows
'F1'	send message to the selected receiver
'F2'	load a stored message from the hard disk. The message is searched for in the subdirectory *MSG*.MSG* Subdirectory *MSG* must exist.
'F3'	save message to hard disk. The message is stored in the subdirectory *MSG* under the name *name.MSG* . The subdirectory *MSG* must exist.
'F4'	remove a stored message from the hard disk. A selected message *Name.MSG* is deleted from the subdirectory *MSG*. The subdirectory *MSG* must exist.
'F5'	activate protocol When the protocol is activated, **all** messages sent and received are stored in an ASCII file on disk. The file has the extension **.ASC*.
'F6'	Reset Counter The counters for messages sent and messages received are reset to zero.
'F7'	Nr.Inc The user can toggle between automatic incrementation of the message number on each transmission and no incrementation.

Keys for the monitor

3.4 Settings - *setup*

3.4.1 Setting the basic addresses - *address*

A device can contain several subsystems. Intelligent communications cards for the PROFIBUS are an example. These subsystems can be described as a device within the device. Each system is addressed by ComPro by means of a unique basic address. Device address zero is reserved for the main system and the default settings. The addresses of the subsystems can be found in the relevant device description.

The basic address of ComPro is only of significance for devices with several diagnostic interfaces. In such a case, a unique identification of the ComPro must be set.

3.4.2 Function - *password*

Prepared for a future function.

3.5 Macro

There is a facility for recording keystrokes with the macro recorder and replaying them. A recorded macro is always stored on the hard disk as an ASCII file.

3.5.1 Recording macros - *recording*

In order to start recording the macro, it must first be given a name. The name of the macro corresponds to the file name under which it is stored on the hard disk. When entry of the name has been acknowledged with '*enter*', the following message appears on the screen:

**** Macro recording active: Name.MAK ****
Macro end with 'CTRL-A'
Insert pause with 'CTRL-P'
Insert wait key with 'CTRL-K'

When recording is active, all keystrokes are stored in the macro apart from the control keys mentioned above.

Key	Function
'Ctrl-X'	End macro recording Completion of recording is acknowledged with the message ****Macro recording ended**** even if the main screen is not activated (e.g. diagnostic, trace).
'Ctrl-P'	When playing the macro, a pause of approx. 1 second is inserted at the point where these keys have been pressed. Longer pauses can be inserted by pressing the keys several times.
'Ctrl-K'	When playing the macro, it waits for a key to be pressed at this point

These control keys can be used at any time, even if for example the message monitor is active.

During recording, the menu options *Play* and *Delay* are not available.

3.5.2 Playing the macro - *playing*

On selection of a macro, it is played with the delay set. **No** interruption is currently possible! A macro can be automatically executed on the start of the ComPro program (see 'The command line').

3.5.3 Delay macro - *delay*

A pause period can be specified, with the result that the pause occurs between re-playing of the individual keystrokes.

3.6 Terminate ComPro - *exit*

ComPro is terminated, a prompt as to whether the user wishes to exit from the program follows.

If the current database has not yet been saved there is then a safety prompt.