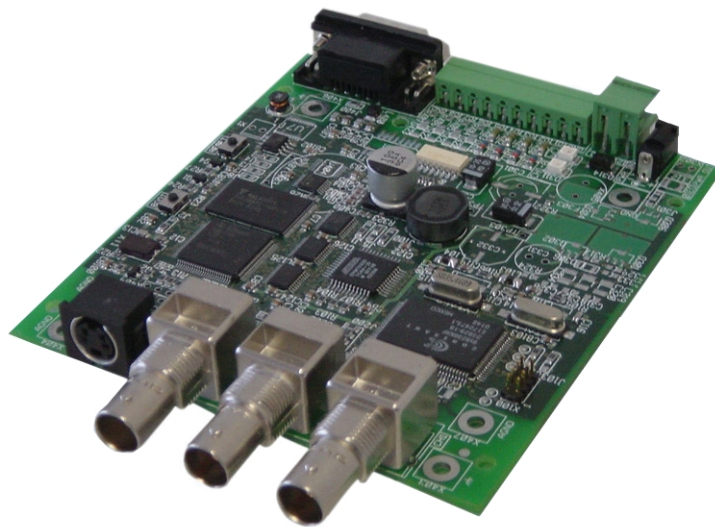


# grabbMODUL-4



## Softwareextensions

Release: July 2004

## Table of contents

1.	General information to the grabbMODUL-4 software .....	3
2.	MODEM_COM: Firmware for remote image transmission .....	4
2.1	Function of the firmware .....	4
2.1.1	Notes to the function of the firmware .....	7
2.1.2	Flowdiagram MODEM_COM .....	9
2.1.3	Overview on the commands and command structure .....	10
2.1.4	Block incoming calls .....	11
2.1.5	Terminate connection .....	11
2.1.6	Resetting device.....	11
2.2	Testing MODEM_COM with a terminal program.....	12
2.3	Notes on developing the PC program.....	18
2.4	Modem.lib – grabbMODUL-4 – driver for the AT instruction set.....	20
2.4.1	Modem_Init .....	21
2.4.2	Modem_Call .....	21
2.4.3	Set_Modem.....	22
2.4.4	Read_Modem.....	22
2.4.5	Read_Modem_without_OD0A .....	22
2.4.6	Check_AT.....	23
2.4.7	Read_Error_Code_Modem.....	23

## Figure listing

figure 1:	Device configuration for MODEM_COM - firmware .....	4
figure 2:	Direct serial connection .....	5
figure 3:	Imaginary serial connection.....	5
figure 4:	Flowdiagram MODEM_COM.....	9
figure 5:	Device configuration for the MODEM_COM firmware.....	12
figure 6:	HyperTerminal: setup program.....	13
figure 7:	HyperTerminal: name program .....	13
figure 8:	HyperTerminal: connection properties .....	14
figure 9:	HyperTerminal: configure program.....	14
figure 10:	HyperTerminal: start program .....	15
figure 11:	HyperTerminal: modem test / synchronisation .....	15
figure 12:	HyperTerminal: establish connection .....	16
figure 13:	HyperTerminal: testing PC – modul communication.....	16
figure 14:	HyperTerminal: terminate connection.....	17
figure 15:	Communication using the AT instruction set (Example) .....	20

## **1. General information to the grabbMODUL-4 software**

In this manual you will find the description for the additional software of the grabbMODUL-4 from Phytec Meßtechnik GmbH. The following additional software is explained in this manual:

- **MODEM\_COM (SO-670-X1):** Firmware for remote image transmission  
The firmware MODEM\_COM is a complete software for the modul, by which image and controldata can be easily transmitted – by a telephonenumber for example - over large distances.

## 2. MODEM\_COM: Firmware for remote image transmission

### 2.1 Function of the firmware

With the firmware MODEM\_COM, PHYTEC supplies you with a software by which image and control data can be transmitted easily over a large distance. To do so, the grabbMODUL-4 is connected to a modem which in turn is connected to the telephonenet. The transmission of the data (control commands and images) occurs over the telephonenet.

With this firmware the grabbMODUL-4 in combination with a modem can receive calls and communicate by a defined protocol with the caller. In addition to the grabbMODUL-4/camera device, on the modul side you will need a modem (telephone connection) and on the other side you need an appropriate remote station (control station or a PC incl. modem).

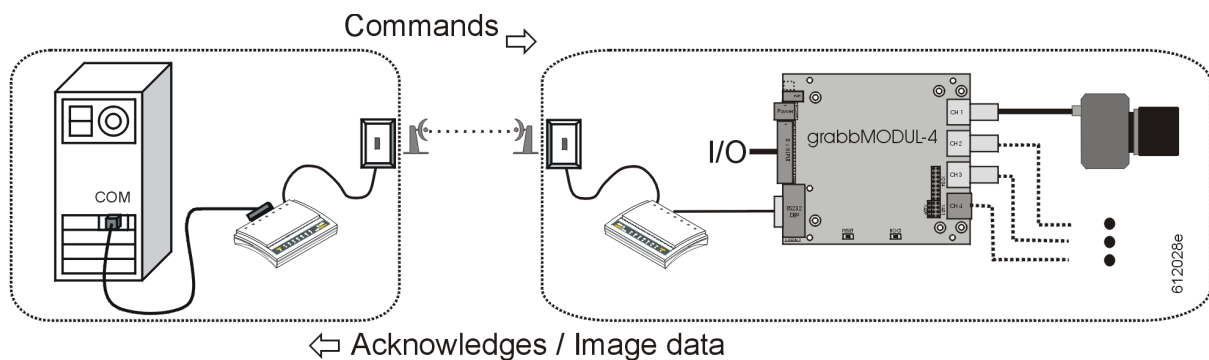


figure 1: Device configuration for MODEM\_COM - firmware

The software on the modul is based on the proven and well described firmware „LOCAL\_COM“ (see manual grabbMODUL-4, L-612). The „LOCAL\_COM“ software needs a serial connection between the grabbMODUL-4 and a control station-/host PC. Over this serial connection commands and acknowledgements are transmitted by which the grabbMODUL-4 can be setup and controlled. The through a certain event or on demand recorded images can be transmitted in different formats by the modul.

It is not important if there is a direct serial connection (see figure 2)

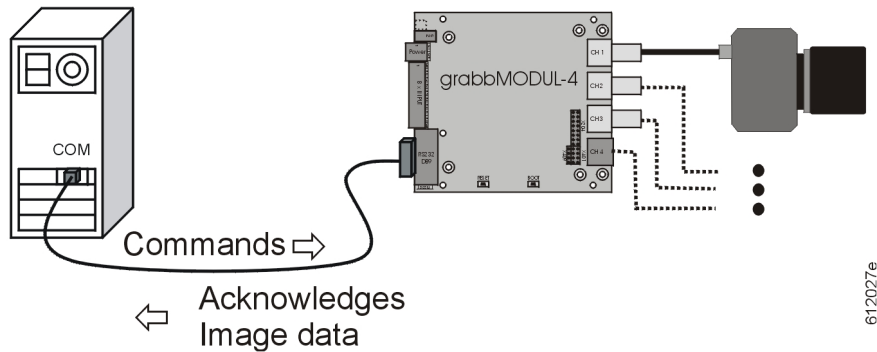


figure 2: Direct serial connection

or an „imaginary“ serial connection by a modem-modem setup (see figure 3).

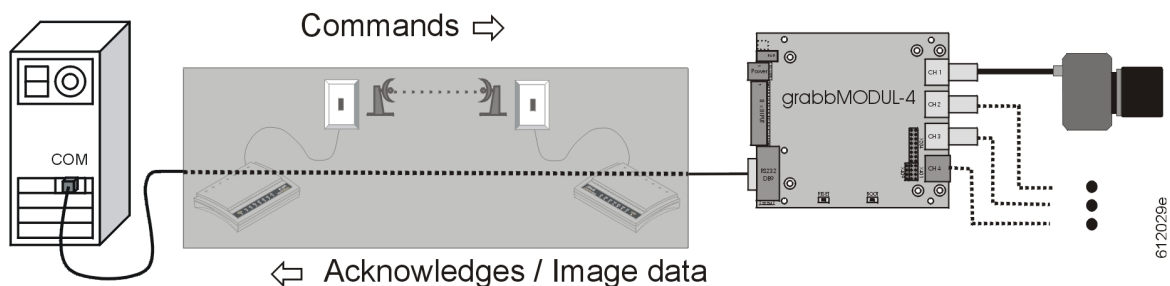


figure 3: Imaginary serial connection

The firmware „MODEM\_COM“ expands the capabilities of the grabbMODUL-4 to communicate via a serial interface with a modem and to put the modem in the „imaginary serial connection“.

The communication between the grabbMODUL-4 and the modem is achieved by AT-commands. The AT-commands are standardized and allow the usage of any modem which understands these types of commands. The following sequence is realized by a grabbMODUL-4 modem communication:

- initialize modem (command mode)
- waiting for call and taking call (command mode)
- after taking the call switching to the „transparent“ serial mode (data mode)
- communication according to the „LOCAL\_COM“ protocol.
- changing mode by a defined command (data mode to command mode) and termination of the connection
- waiting for further calls.

Please note:

- The firmware MODEM\_COM must be installed on the grabbMODUL-4. To achieve this the file *modem\_com\_Vxx.h86* must be downloaded to the moduls FLASH-memory by using the PHYTEC-flashtools. (xx is the version number). When the modul is delivered, the software „LOCAL\_COM“ is in the FLASH memory of the grabbMODUL-4.
- The grabbMODUL-4 and the modem must be connected by a fully connected serial cable. If possible, apply a ready to use cable which normaly comes with the modem. The modem must be connected with a phoneplug and switched on.
- The communication of the grabbMODUL-4 with a host/PC by a modem is only possible if the PC is also connected with a modem. Furthermore you have to develope a PC program which configures the modem of the PC, makes a call and switches to the data mode (imaginary serial connection). After that, the developed PC software can communicate by the commands described in the „LOCAL\_COM“ with the grabbMODUL-4.
- After the download to the grabbMODUL-4, the software „LOCAL\_COM“ can be tested with a terminal program (like HyperTerminal see section 2.2) or your own developed windows software. If the firmware „MODEM\_COM“ is in the FLASH memory of the grabbMODUL-4, after a new start the modul will look for a modem on his serial interface. This is the reason why a direct communication of a PC and the grabbMODUL-4 is not possible.

### 2.1.1 Notes to the function of the firmware

The sequence of functions is shown in section 2.1.2 . The explanation follows in this section:

- The software of the grabbMODUL-4 initializes the serial interface with 57600 baud, 8 data bits, no parity, one stop bit and hardware handshake (CTS/RTS).

#### **NOTE:**

The connection speed 57600 baud between the grabbMODUL-4 and the modem does not correspond to the speed of the modem-modem connection. If the modem-modem connection is setup with a different speed (9600 Baud for example) communication is still possible because data is buffered in the modems.

- After that the software will wait for 2 sec. (Power On Modem) und tries to establish the first connection.
- ① sending the sequence „+++“ (terminates the data mode)
- sending the sequence „ATH0“ (terminates a connection)
- sending the sequence „AT“ (synchronizing with the modem)
- waiting for the answer „OK“
- If no „OK“ is received a software reset on the modul will be the result, the grabbMODUL-4 will then continously try to setup a communication to the modem, jump back to ①.
- ② If a „OK“ is received the modul waits for a call („RING“), which is recognized and relayed by the modem.
- During this period the input channels are active. That means at a any change of edge at the inputs IO\_1 - I/O\_4 an image will be grabbed from the corresponding channel.

#### **NOTE:**

After a connection is setup, the automatic input poll is inactive and an image can only be recorded by the grabb command „G“

- If a „RING“ is sensed, the grabbMODUL-4 sends the „ATA“ character sequence so the modem can take the call.
- At this point the setup time of the modem-modem connection can vary. Therefore a up to 50 sec wait state is inserted until an answer from the other modem is received. Since this answer may vary according to the used modem and quality of connection, the defined string „CONNECT“ included in the answer will be checked while the rest of the answer is discarded.

- If a „CONNECT“ is recognized, the sequence jumps back to ① and by the „AT“ command it is checked if the modem is still active. If the modem is still active the sequence jumps back to point ② and continues from there, if not a software reset is initiated.
- If a „CONNECT“ is recognized then the modem has switched to the „transparenten“ serial mode (data mode). It is now possible to communicate with the remote station, as if there would be a direct serial connection to the grabbMODUL-4.
- The grabbMODUL-4 now waits for control characters from the host according to the „LOCAL\_COM“-specification. The used protocol is explained in detail in the manual grabbMODUL-4, L-612, Firmware „LOCAL\_COM“. For the use with a modem the instruction set of the „LOCAL\_COM“ protocol has been extended. These additions are explained in the section 2.1.3 of this manual.
- If no characters are exchanged for 4 min. in this mode, a connection malfunction is assumed. In this case the grabbMODUL-4 tries to put itself in a defined state. That means, the command to terminate the connection is sent and the answer from the modem is checked. If the modem answers the sequence jumps back to point ②, else a software reset is initiated on the grabbMODUL-4.

**NOTE:**

- With the AT-command communication from the modul the control commands 0x0D and 0x0A as well as the from the modem echo function returned characters will be polled.
- The grabbMODUL-4 only uses the AT-commands for the communication with a modem. Therefore the use of any modem should be possible. However, when selecting a modem please check if it can handle the LOCAL\_COM – firmware in all operating states.
- There is a possibility to suspend an incoming call for 2 min. (see section 2.1.4). That means the modul will stay in the input poll routine ② although there is a „RING“.The incoming call is not taken. This enables to use an additional device at the same phone plug.
- While an on-line connection is those, described in the grabbMODUL-4 manual L-612 (section „picture recording by alarm entrances“), automatic picture grabbing deactivated.



### 2.1.2 Flowdiagram MODEM\_COM

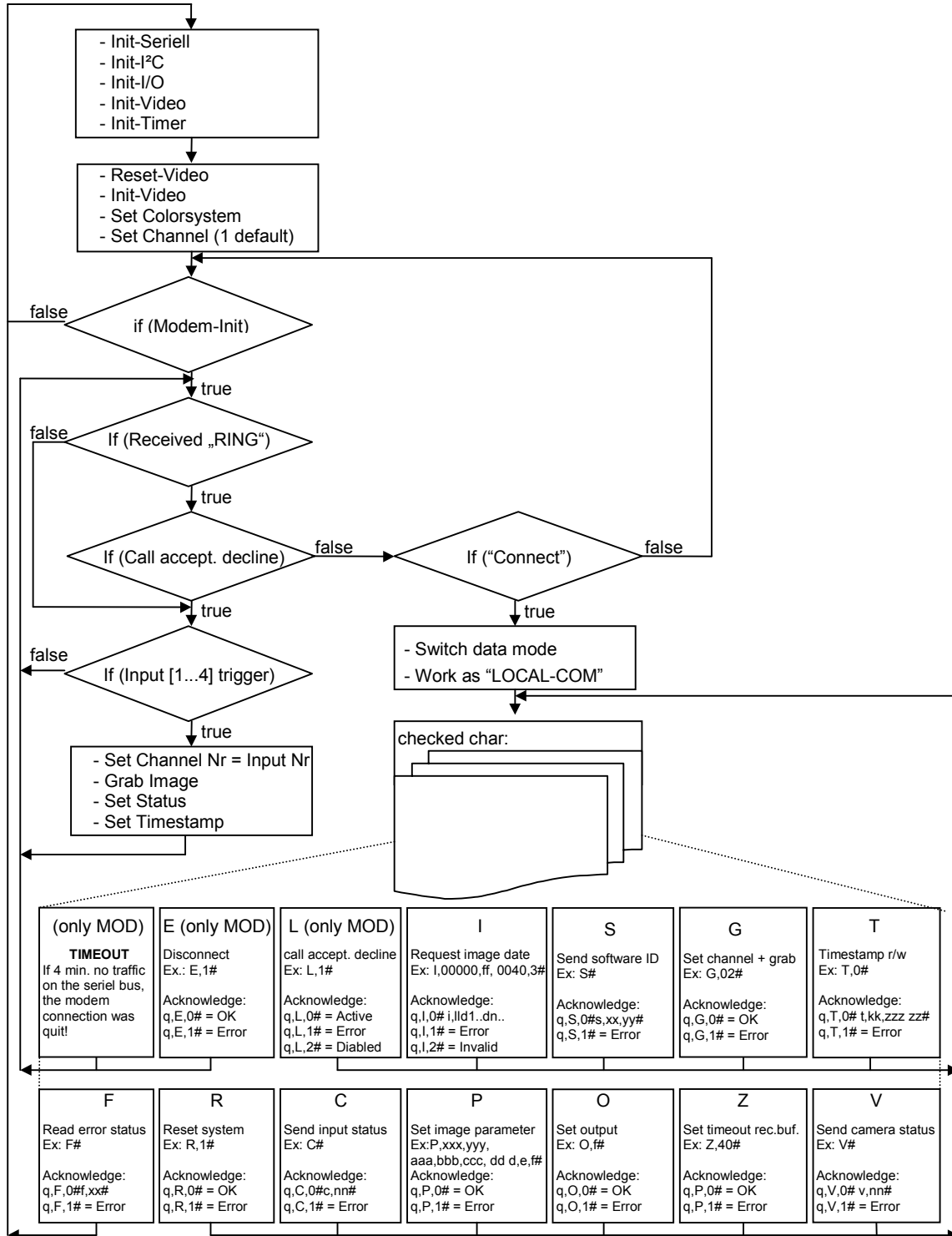


figure 4: Flowdiagram MODEM\_COM

### 2.1.3 Overview on the commands and command structure

The structure and function of the commands and handshakes are described in the manual grabbMODUL-4, L-612, section „LOCAL\_COM“. In this section only the additional commands for the use with a modem are explained.

The following commands are implemented in the LOCAL\_COM firmware:

- „C“ - Polling input status
- „F“ - Polling error status
- „G“ - grabb camera image from channel k
- „I“ - request image data
- „O“ - set output status
- „P“ - set image size and scaling
- „R“ - reset device (**extension for modem use !**)
- „S“ - poll software ID
- „T“ - read / reset timestamp
- „V“ - request camera status
- „Z“ - set receive timeout for interface
- 0x0D - reserved (control comands for modem)
- 0x0A - reserved (control comands for modem)

The following commands are an addition to the MODEM\_COM firmware:

- „L“ - block incoming call
- „E“ - terminate connection

Please note, that the grabbMODUL-4 in idle state (with no connection established) communicates by the AT-instruction set with the modem. Only after a connection initiated by a „RING“ has been set up, the modem and the remote modem switch to the „transparent“ mode. After that the modul talks virtually serial with the system (PC for example) on the remote side.

If no characters are exchanged for 4 min. in this mode, a connection malfunction is assumed. In this case the grabbMODUL-4 tries to put itself in a defined state. That means, the command to terminate the connection is sent and the answer from the modem is checked.

### 2.1.4 Block incoming calls

Instruction code: L,x#  
Parameter: 0 = blocking deactivated  
              1 = blocking activated  
Example: L,1#  
Acknowledge: q, L, 0# = acknowledge, blocking active  
              q, L, 1# = parameter error, no action  
              q, L, 2# = acknowledge, blocking activated

Remark: The blocking is only active after the current connection has been terminated by the „E“ command. After that blocking is active for 2 min., independent of how often a „RING“ is sent by the modem.

### 2.1.5 Terminate connection

Instruction code: E,x#  
Parameter: 1 = terminate connection  
           all others: terminate instruction  
Example: E,1#  
Acknowledge: q, E, 0# = acknowledge, connection will be terminated  
              q, E, 1# = parameter error, connection will remain

Remark: A current connection should always be terminated by the host through the „E“ command. If the remote station hangs up, the modul will not sense this hang up. In this case the modul notices the interruption after the expiration of the preset 4 min timeintervall. Only after tthis time span the modul is able to receive a new call.

After the modul has sent the acknowledgement the hang up as such will follow after approx. 1.1 sec. (so called Guard Time) from the modul side by sending the escape sequence („+++“) and the hang up command („ATH0“). After the modul-modem has hung up, the host-modem will notice that and will also hang up. According to the used modem the hang up time will vary. If you would like to dial again immediatly from the host, it is recommended to enforcce an immediate hang up through an on/off transission of the DTR line.

### 2.1.6 Resetting device

Instruction code: see manual grabbMODUL-4, L-612, section „Resetting device“.

Remark: In addition to the described functionality this command has been extended as follows: Prior to resetting the modul by software, the grabbMODUL-4 sends a control command to terminate a current connection.

## 2.2 Testing MODEM\_COM with a terminal program

For testing the „MODEM\_COM“ you will need the in figure 5 shown device configuration.

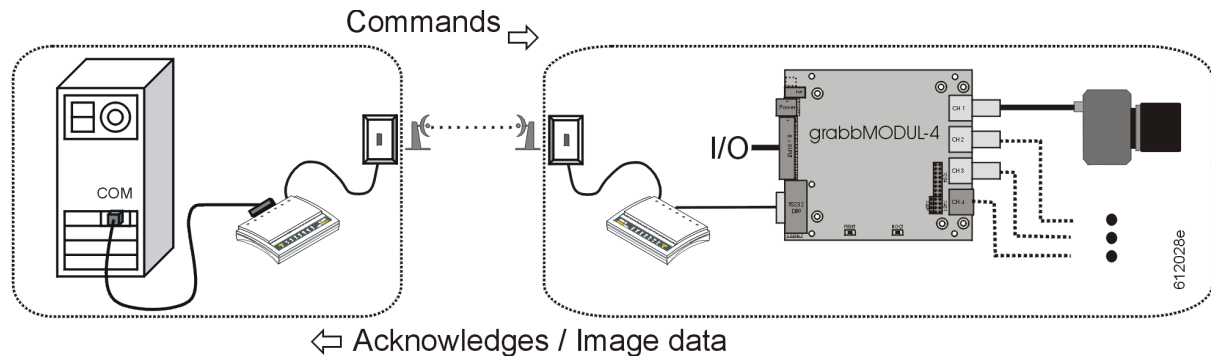


figure 5: Device configuration for the MODEM\_COM firmware

The basis of this configuration is the grabbMODUL-4 incl. a camera and the needed connection cables. The common modul operation is described in detail in the manual grabbMODUL-4, L-612, section „Rapid-Development-Kit operation“.

In addition to this base configuration the following components are needed for a „MODEM\_COM“ connection:

- A modem, directly connected to the serial interface of the grabbMODUL-4 and to a free phone plug.
- A second modem directly connected to a PC (serial, USB or integrated) and to a free phone plug (acts as the remote station).

Proir to testing this configuration you must assure that the „MODEM\_COM“ firmware is in the Flash memory of the grabbMODUL-4. To do so follow the steps „Installation of the PHYTEC FlashTools and downloading program code“ in the manual L-612 and download the file „Modem\_Com\_Vxx.h86“ (**SO-670-X1**) to the modul.

### **Note:**

If during operation the modem is switched off/on, you must also reset the grabbMODUL-4 to reinitialize the modem-modul connection.

For testing the function and the protocol a terminal program on the PC side can be used. For example the windows supplied program „HyperTerminal“ can be used. Command strings can be send to the grabbMODUL-4, the answers and raw image data (ASCII format) can be displayed. The following procedure may not be the same for different operating systems.

First of all, make sure that a modem is connected to the PC. The COM port which the modem uses must be known. When using an internal modem this information is provided by the hardware manager.

- Start the program *HyperTerminal* in the directory *Start\Programme\Zubehör* or *Start\Programme\Zubehör\Kommunikation*. (The directory may differ according to the operating system).
- The following *HyperTerminal* window will open:

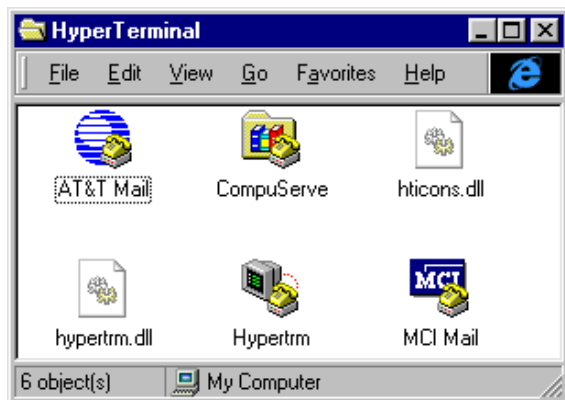


figure 6: *HyperTerminal*: setup program

- By double clicking the „*HyperTerminal*“ symbol a new *HyperTerminal* connection is started.

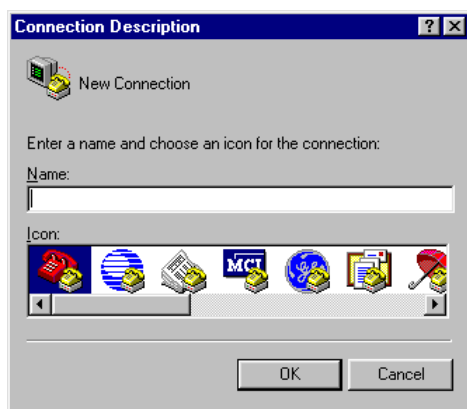


figure 7: *HyperTerminal*: name program

- The window „*Connection Description*“. Enter „*MODEM connection*“ in the name field.
- After confirming by clicking „*OK*“ a new *HyperTerminal* connection with the name „*MODEM connection*“ is created.

- Specify the properties of the COM connection in the „New connection properties“ windows.

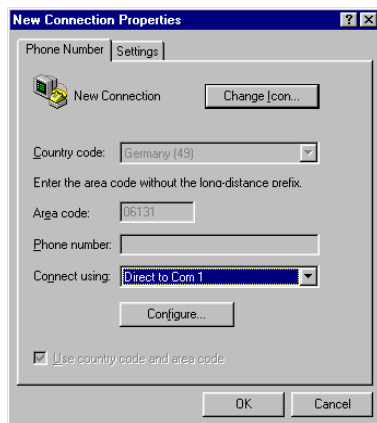
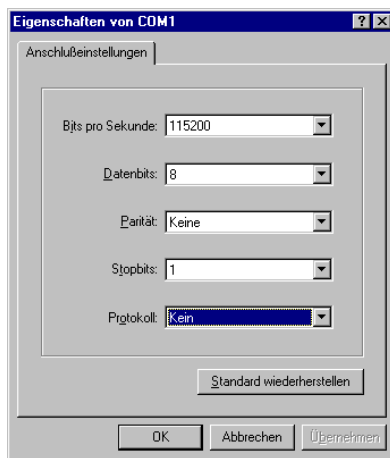


figure 8: HyperTerminal: connection properties

- Choose in the „Connection using:“ window the COM port which your modem uses (COM 1 for example) and confirm.
- By clicking on the button „Configure“ the window „Properties of COM1“. In this window set the COM parameters as follows:



- Bits per second = 115200
- data bits = 8
- parity = None
- stop bits = 1
- protocol = None

figure 9: HyperTerminal: configure program

- Click on „OK“ and switch to the monitor window “MODEM connection – HyperTerminal”.

Close and save HyperTerminal and start the newly created connection icon „Modem connection“, so the settings are applied ! Depending on which operating system is used, the icon may be in a separate folder called „HyperTerminal“.

- After starting „MODEM connection – HyperTerminal“ you will see the monitor window. In the lower window border some status information about the connection is displayed.

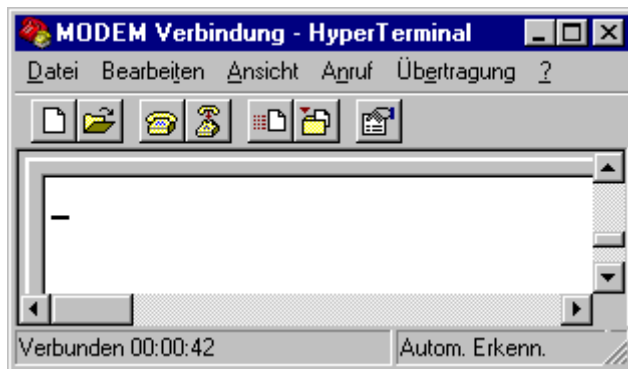


figure 10: HyperTerminal: start program

- Initiate a reset of the grabbMODUL-4 by pressing the onboard reset button. This will start the „MODEM\_COM“ program from the FLASH memory.
- At first the communication PC-modem should be tested and synchronized. Enter „AT<ENTER>“ in the HyperTerminal window. The modem should answer with „OK<ENTER>“. If this is not the case please check the PC-modem connection.

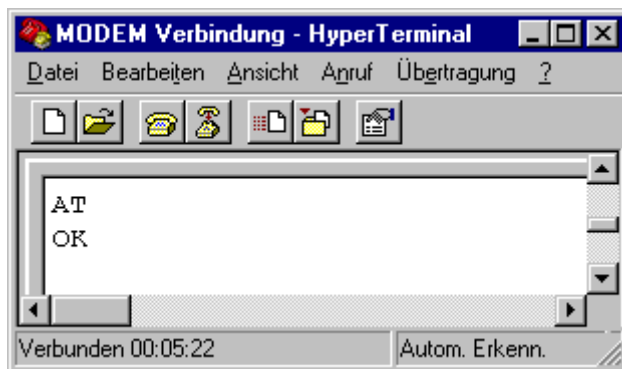


figure 11: HyperTerminal: modem test / synchronisation

- The next step is to call the grabbMODUL-4 modem and establish a connection. The call is done by „ATDxxx<ENTER>“ (**xx stands for the MODUL-modem calling number!**). The grabbMODUL-4 on the remote side will answer the call after the first „RING“. After the two modems have connected the PC-modem will send an answer. If the connection was successful this answer will always begin with „CONNECT ...“ (see figure 12).

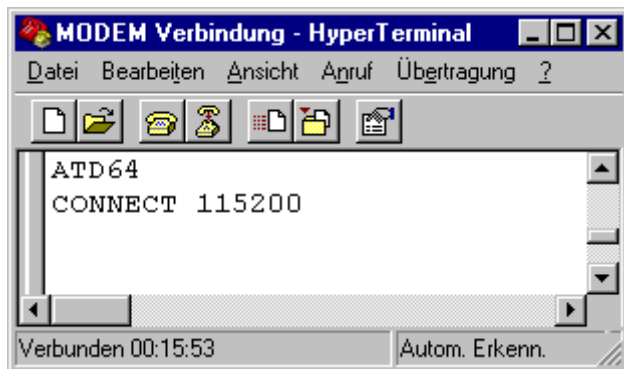


figure 12: HyperTerminal: establish connection

- After a successful connection both modems are in the data mode. That means, the PC and the grabbMODUL-4 can communicate via a serial connection as if they were directly connected with each other. As an example the function „poll software ID“ is tested:

Enter the following strings: „S#“

The modul answers: „q, S, 0#s, 03, 05#“ (depending on the version)

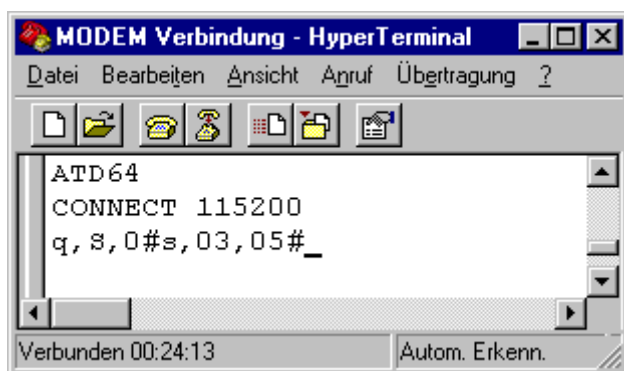


figure 13: HyperTerminal: testing PC – modul communication

- The termination of a connection should always be realized by the defined protocol (see section 2.1.5). To do so enter the command „E,1#“. The grabbMODUL-4 on the remote side first sends the acknowledgement and then after approx. 1.1s (Guard Time) it switches to the command mode („+++“) and the connection is terminated („ATH0“ is not always echoed back !). This will be displayed on the PC/modem side by „NO CARRIER“ (see figure 14). The line is now ready for a new connection.

After the modul-modem has hung up, the host-modem recognizes this and also hangs up. Depending on the type of modem used this timeperiod to the hang up may vary. If you would like to dial immediately after a long timespan it is recommended to force the hang up immediately by switching the modem off and on.



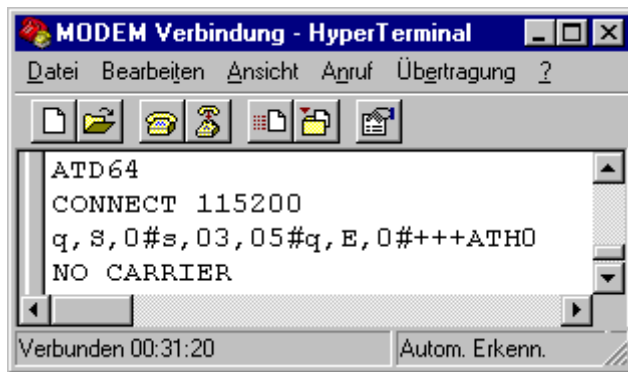


figure 14: HyperTerminal: terminate connection

**Note:**

If a connection is not terminated by the protocol command „E,l#“ but the PC side hangs up, then the modem will not notice that. In this case the modul will recognize this malfunction after the preset 4 min. . Only after this timespan it will be ready to receive a new call.

- Now, all protocol functions can be tested in the described manner.

**Note:**

By testing the „I“-command (requesting image data) you have to enter manually after each data block the acknowledgement „q, i, 0#“ to send it via *HyperTerminal*.

So that the manual input is not interrupted by the preset timeout, it is recommended to raise it by the Z-command („Z, 30#“), as explained in the manual grabbMODUL-4, L-612 section „Setting receive timeout of the interface“.

Since the characterstring „0x2B,0x2B,0x2B“ („+++“, can randomly exist in image data) is interpreted as a command in the data mode (the modem would then leave the data mode), the grabbMODUL-4 will suppress this sequence if it is sending image data.

- To terminate the connection between the PC and the modem simply klick on the „*Terminate connection*“ button in HyperTerminal or close the program.



“*Terminate connection*”

If no output is displayed in the HyperTerminal window, please check the power supply, the COM-port - parameter and the RS-232 connection.

## 2.3 Notes on developing the PC program

At this point a few notes on developing the PC program:

- Before developing a program for modem transmission, you should first establish a local connection with the grabbMODUL-4, develop and test a general code framework including the most important control commands for the local mode. The modul program „LOCAL\_COM“ is on the driver CD SO-670. You will find the description in the manual to the grabbMODUL-4 (Dokument L-612).
- To make the PC-program communicate via a modem, the following steps have to be taken on the PC side:
  - Initializing the modems
  - Dialing of a certain telephone number (remote station on the modul).
  - Checking if there is a connection.
  - Exchanging the defined strings of the protocol in serial mode.
  - Possibility to display the raw image data as a picture.
  - Maintaining the protocol defined timeouts.
  - Possibility to control and check the connection.
  - Termination of a connection („E,I#“).
- First tests can be conducted with internal telephone equipment to reset the systems fast in case of troubleshooting and debugging.

### **Hinweis:**

- Depending on which mode you currently work with, local or modem mode, the corresponding counterpart must be downloaded to the grabbMODUL-4.
- The modems in use should support the same transmission standard (i.e. analog analog or analog ISDN-device into analog-device-mode). A mixed connection like analog – ISDN for example is not possible.
- If one of the modems on the phone side is in an undefined state (malfunction etc.) it may be possible to resolve this problem by powering the modem off/on. In this case, to synchronize the modul and the modem the modul must also be powered off/on. If this state occurs quite often, we recommend using a modem with an internal „watchdog“ function.
- Please note, that some modems may use slightly different communication protocols or differ in case of country properties (signal line definition). Therefore check prior to use in detail the modems usability and reliability. In this context we recommend a field test where necessary.
- After the modul-modem has hung up the host-modem will sense this and also hangs up. Depending on the used modem type the timespan until hanging up may vary. If you wish to dial immediatly from the host side, it is

recommended to force an immediate hang up by an on/off transission of the DTR line.

## 2.4 Modem.lib – grabbMODUL-4 – driver for the AT instruction set

The *Modem.lib* includes functions to initialize and to communicate with a device (modem) by the AT instruction set.

The definition BYTE stands for „unsigned char“ and the definition of WORD for „unsigned int“.

**IMPORTANT:** To use a function from the *Modem.lib* the *Modem.h* must be included in your project.

The following functions are supplied by the *Modem.lib*:

- Modem\_Init()
- Modem\_Call()
- Set\_Modem()
- Read\_Modem()
- Read\_Modem\_without\_OD0A()
- Check\_AT()
- Read\_Error\_Code\_Modem()

The AT command communication is structured as follows:

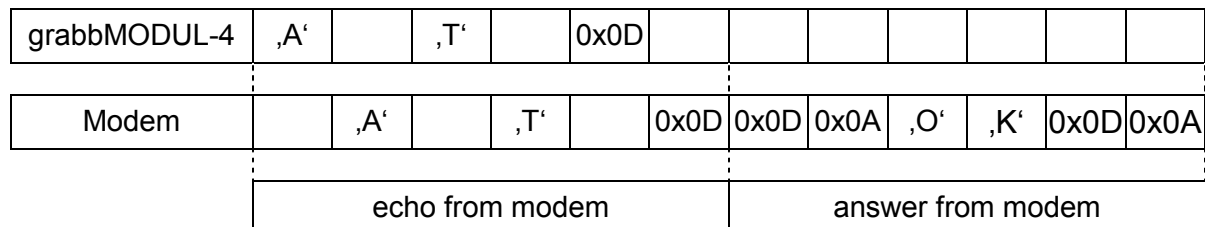


figure 15: Communication using the AT instruction set (Example)

**IMPORTANT:** The functions of the *Modem.lib* do analyze echos and control strings. Therefore it is necessary that the used modem understands this standard protocol and the ECHO function of the device is not turned off. Is this the case, communication errors will be the result !

### 2.4.1 Modem\_Init

**Function:** Initializing and synchronizing the baud rate with a device (modem) by an AT instruction.

#### BYTE Modem\_Init (void)

Return value:     0 = device was identified, synchronization successful  
                  1 = no device could be identified

The function **Modem\_Init()** must ensure that the modem is in a defined state. Therefore the string „+++“ (switching from data to command mode) is sent first. Next, by the command „ATH0“ it is ensured that an existing connection is terminated. These two commands are sent „blind“ without analyzing the answer. After approx. 5 sec the synchronization itself will be by the command „AT“ and testing the answer „OK“ be used to decide for a successful initialization. If an error occurs with the „AT“-synchronisation, the field variable „ERROR\_CODE\_MODEM“ errorbit\_0 (bit value 1) is set.

### 2.4.2 Modem\_Call

**Function:** Senses on the serial connection a call („RING“), sends a command to take the call and tests the execution.

#### Byte Modem\_Call (void)

Return value:     0 = connection successful  
                  1 = connection not successful

The **Modem\_Call()** function compares a complete received string with the „RING“ string. If the strings are equal, the command „ATA“ is sent to the modem to take the call. After that the modem will connect autonomously with the remote station. Depending on the line quality and current transfer this may take some time. Therefore a timespan of 30 sec is used to wait for the result of the established connection. If the connection is successful it will be signaled to the modem by the „CONNECT ...“ command. Depending on the used modem and phoneline condition there may follow further information after the „CONNECT“. Therefore only the preliminary „CONNECT“ is checked while the rest of the broadcast is discarded.

### 2.4.3 Set\_Modem

**Function:** .

**BYTE Set\_Modem (char \*at\_out\_array)**

\*at\_out\_array: String to send

Return value: 0 = sending of string successful  
1 = wrong or no answer from the modem, transmission aborted

The **Set\_Modem()** function sends the assigned string incl. the end of sign indicator 0x0D. After each character the echo is checked before the next character is sent (figure 15: Communication using the AT instruction set (Example)).

### 2.4.4 Read\_Modem

**Function:** Receives a character from the serial interface according to the AT protocol.

**BYTE Read\_Modem (char \*at\_in\_array)**

\*at\_in\_array: String in in which the received string is stored

Return value: 0 = reception of an AT sequence successful  
1 = begin and end indicator on the interface is wrong, or no characters

A transmission from a device with AT instruction set is always introduced by 0x0D, 0x0A and will be terminated 0x0D, 0x0A (see figure 15: Communication using the AT instruction set (Example)). The **Read\_Modem()** waits for a string from the interface which are sent with these indicators. The string (without the indicators) is written into the „\*at\_in\_array“ string. If the protocol: is incorrect, the sending of characters fails or the number of characters greater then 50, then the function will return with the error parameter = 1 and the „ERROR\_CODE\_MODEM“ error variable errorbit\_1 (bit value 2) is set.

### 2.4.5 Read\_Modem\_without\_0D0A

**Function:** Receives a character from the serial interface according to the AT protocol without the startsequence 0x0D and 0x0A.

**BYTE Read\_Modem\_without\_0D0A (char \*at\_in\_array)**

\*at\_in\_array: String in in which the received string is stored

Return value:     0 = reception of an AT sequence successfull  
                  1 = begin and end indicator on the interface is wrong, or no  
                  characters

A transmission from a device with AT instruction set is always introduced by 0x0D, 0x0A and will be terminated 0x0D, 0x0A (see figure 15: Communication using the AT instruction set (Example)). The **Read\_Modem\_without\_0D0A()** waits for a string from the interface which are removed the startsequence (0x0D, 0x0A). The string (without the indicators) is written into the „\*at\_in\_array“ string. If the protocol: is incorrect, the sending of characters fails or the number of characters greater then 50, then the function will return with the error parameter = 1 and the „ERROR\_CODE\_MODEM“ error variable errorbit\_1 (bit value 2) is set.

#### 2.4.6 Check\_AT

**Function:** Synchronizing the baud rate with a device (modem) by an AT instruction.

#### **BYTE Check\_AT (void)**

Return value:     0 = Device identified, synchronization successfull  
                  1 = No device was identified

The function **Check\_AT ()** sends the „AT“ command and checks the answer „OK“ to ensure that the connected device (modem) is still ready for operation. If the „AT“ synchronization is incorrect the „ERROR\_CODE\_MODEM“ error variable errorbit\_0 (bit value 1) is set.

This command should only be used if you are working in the command mode.

#### 2.4.7 Read\_Error\_Code\_Modem

**Function:** Reading the error variable „ERROR\_CODE\_MODEM“.

#### **BYTE Read\_Error\_Code\_Modem (void)**

Return value:     0x00 = no error ocured  
                  0x01 = failure initializing modem  
                  0x02 = answer from modem wrong