Windows DLL MEGSV4w32.dll

Manual for measuring amplifier BSC4D & BSC4BT

NOTE: GSV4 = BSC4

Last changes: 08.05.2017

Table of contents

1 Description

Dynamic Link Libraries (DLL) are available for programming the GSV-4 assembly on Windows (Microsoft Corp.)-wide, one for 32-bit and one for 64-bit. Both contain the same features that often match the firmware commands, but sometimes include additional functionality or combine multiple commands to simplify programming. They also include functions for opening a serial interface with simultaneous communication with the GSV-4 and for closing a serial interface thus open. The functions of the DLL can be imported via its index (order number).

The activated interfaces and GSVs are managed separately for each program that uses this DLL. The use of an interface (with GSV) that is other process is not possible. Different threads of a process, their activated interfaces can be shared in parallel or pseudo-parallel use.

2. Versions and variants

The variants listed here are identical in call stack, return type, and functionality. For all variants, Wrapper VIs are available for easier integration with NI LabView, as well as a C-lib for use with C/C++.

1.1. MEGSV4.DLL

The last version of this 32-bit DLL shipped is 1.10. It will not be further maintained from November 2013. The calling convention is cdecl.

1.2 MEGSV4w32.DLL

This 32-bit DLL will be delivered in version 1.13.00.32 from November 2013. The calling convention is __stdcall. Therefore, this variant is suitable for use with VisualBasic and many other programming languages. A BAS file is available for integration into VB. A WinSxS manifest is embedded.

1.3 MEGSV4w64.DLL

This 64-bit DLL will be delivered in version 1.13.00.64 from December 2013.

Calling convention is __fastcall. Apart from that, the same applies as for the MEGSV4w32.DLL.

3. Principal approach

To create an application program that uses the DLL, the following is usually done:

3.1. Inclusion of the DLL

The DLL and its declaration files (headers) must be included in the application program that uses the DLL functions. Exactly how this is done depends on the programming language and development environment. In general, DLLs are integrated at program time, i.e. dynamically. A LIB import library is available for static integration for use with C/C++ programs.

3.2. Program

Most application programs that read and evaluate measured values can be roughly divided into three parts: initialization, main loop, and termination. Simple sample programs in C and LabView are available for download on www.me-systeme.de or are in the archive that contains this guide.

3.2.1. Initialization

Initialization is normal for a program run, either once or by user input. It can be divided into two or three parts: integration of the DLL (only necessary for dynamic binding), activation of the interface and testing and, if necessary, adjusting relevant settings of the measuring amplifier.

Example:

Let the DLL be already statically integrated. The first thing to do is to open the interface by calling GSV4actExt or GSV4activateExtended. Subsequently, it is often useful to check the status of the continuous measurement transfer with GSV4getTXmode and, if necessary, adjust it. Also, the display standardization (reading with GSV4DispGetNorm) is often used to evaluate the later reading measuring values in the physically correct display.

Note:

It is not recommended to set operating parameters, which the GSV-4 itself stores non-volatilely in its EEPROM, in general terms - especially not within a program loop - because the EEPROM is worn out by write access (specified for 1 million writes).

3.2.2. Main loop or thread

If the amplifier continuously sends measurements automatically, it is expedient and efficient to proceed as follows in the case of repeated reading and evaluation:

- Ask if measured values are in the buffer. This can be done by returning one of the reading functions (GSV4read, GSV4readAll or GSV4readMultiple) or by the GSV4received function.

- If so, evaluate the measured value(s), by using the content of the pointer passed to the previously called read function, or by using GSV4received: calling the read function(s). Subsequently, the raw measured value is often multiplied by the display scale read in the initialization in order to obtain physically correct measured values.

- If no: maintain or delay the measurement processing thread so that the operating system is not unnecessarily utilized (depending on the programming environment). It is efficient to choose this waiting time slightly less than the measurement data end period of the GSV-4.

If the amplifier does not automatically send measured values, the GSV4getValue function must first be called (i.e. before the buffer is queryed) to trigger the sending of a measured frame by the GSV-4. The triggering of the measured value frame transmission is also possible by a digital I/O line, see GSV4getDigitalIOtype.

3.2.3. Termination

GSV4release must be called to terminate the program or communicate with the GSV-4.

4. The DLL functions

The functions are listed here in order of relevance.

If the detailed description of a parameter is omitted, it can be assumed that the preceding description of the parameter of the same name applies the same way.

The return value is often (so) an error status value. If this does not signal an error, this does not have to mean that the firmware executed the command correctly (do not check the DLL functions), but it simply means that no error has occurred within the function itself. This is especially true for write commands. Therefore, for commands that change the operating parameters of the device, it is recommended to read the value again after writing with the appropriate read function to verify it.

4.1. GSV4actExt

This function allocates necessary memory resources and attempts to open the specified COM interface and then start communicating with the GSV-4.

GSV-4 firmware commands used: get_mode (0x27), set_mode (0x26), get_tx_status (0x29), get_firmware_version (0x2B)

EEPROM-Storage: no

Minimum firmware version number: 0x05

DLL-Atomic number (Index): 1

Transfer parameters:

Name	Data type	Short description	Value range
ComNo	Signed Int32	Number of the COMport to be opened	1 to 256
Bitrate	Signed Int32	Bitrate of COMport communication	Usually only **115200**
BufSize	Signed Int32	Buffer size in measured values.	1 to 3*10^8

Description:

ComNo: This number is assigned by the system or by communication drivers (e.g. Bluetooth, USB). It can be looked up in the device manager of the control panel.

Bitrate With almost all GSV-4 devices this is 115200 bits/s. However, since it can also be adjusted (with restrictions), the following different baud rates are possible: 9600, 19200, 38400, 57600 and 230400.

BufSize: This buffer is created for each of the 4 channels and incoming measured values from the GSV-4 are stored in it until they are fetched by one of the read functions. If the latter does not happen, older values are overwritten so that they are lost. A buffer overflow error is not triggered, i.e. everything continues to work normally.

Return parameters

Data type	Short description	Value range
Signed Int32	Error status	-1, 0

Description:
If the function was executed successfully, GSV_OK = 0 is returned, otherwise, if an error has occurred, GSV_ERROR = -1. In the event of an error, GSV4getLastError can be used to fetch an error code that is to be interpreted as an indication of a possible cause of the error. See Appendix A. In addition to the error codes listed there, GSV4getLastError can also return one of the Windows system error codes after an unsuccessful opening attempt.

4.2. GSV4activateExtended

Like GSV4actExt, but with an additional flag parameter with which alternative settings can be selected when opening.

Used GSV-4 Firmware commands: get_mode (0x27), ggf. set_mode (0x26), get_tx_status (0x29), get_firmware_version (0x2B), ggf. stop_transmission (0x23)

EEPROM-storage: no

Mindest-Firmware version number: 0x05

DLL-atomic number (Index): 36

Transfer parameters:

Name	Data type	Short description	Value range
ComNo	Signed Int32	Number of the COMport to be opened	1 to 256
Bitrate	Signed Int32	Bitrate of COMport communication	Usually only 115200
BufSize	Signed Int32	Buffer size in measured values.	1 to 3*10^8
flags	Signed Int32	Additional settings	Flag value: 0 to 772

Description:

ComNo: like GSV4actExt

Bitrate like GSV4actExt

BufSize: like GSV4actExt

flags: The following values can be transferred individually or combined by bitwise:

ACTEX_FLAG_HANDSHAKE value 0x004. This activates hardware handshake. This setting is only recommended with GSV-4BT Bluetooth devices that were delivered before the end of 2012. If the opening still fails, an attempt should be made to call the function without handshake.

ACTEX_FLAG_WAIT_EXTENDED value 0x0100. For devices that only switch on their hardware by opening the interface (e.g. GSV4-BT), this means that there is a longer wait until communication with the device is excluded (namely approx. 4 seconds instead of approx. 0.5 s without ACTEX_FLAG_WAIT_EXTENDED). When communication is started, the first read command is carried out up to 5 times if the GSV-4 has not yet responded.

ACTEX_FLAG_STOP_TX value 0x0200 With this flag, the constant transmission of measurement data is stopped in general when opening. If many devices that use a common communication channel (e.g. several GSV-4BT) are opened or if the data rate of the constant transmission of measured values is relatively high (from 125 values/s), this can make opening the interfaces easier.

Return parameter: like GSV4actExt

4.3. GSV4release

This function closes the interface and all allocated resources are released. The call is required before ending an application that has opened an interface with GSV4actExt or GSV4activateExtended.

Used GSV-4 Firmware commands: no

DLL-atomic number (Index): 10

Return parameter:

Name	Data type	Short description	Value range
ComNo	Signed Int32	Number of the COMport to be opened	1 to 256

Description:

ComNo is the number of the communication port, which has been opened before through GSV4actExt oder GSV4activateExtended.

Return parameters

Data type	Short description	Value range
Signed Int32	Error status	-1, 0

Description:
If the function succeeds, GSV_OK =0 is returned, otherwise if an error has occurred, GSV_ERROR =-1. In the event of an error, GSV4getLastError can be used to retrieve an error code that is to be interpreted as an indication of a possible cause of the error. See Annex A.

4.4. GSV4dllVersion

This function asks for the version number of the DLL version.

Used GSV-4 firmware commands: No

DLL-Atomic number (Index): 39

Transfer parameters: No

Return parameters

Data type	Short description	Value range
Signed Int32	DLL-Version	65536..6553600

Description:
The return parameter is: VER_H x 65536 + VER_L
The DLL version resource contains the following 4 numbers:
VER_H . VER_L . 00 . XX. The last (XX) is either 32 for the MEGSV4w32.DLL or 64 for the MEGSVw64.DLL.

4.5. GSV4getTXmode

With this function, the status of the continuous measurement transmission by the GSV-4 is determined.

Used GSV-4 Firmware commands: get_tx_status (0x29)

EEPROM-storage: No

Minimum firmware version number: 0x08

DLL-atomic number (Index): 51

Transfer parameters:

Name	Data type	Short description	Value range
ComNo	Signed Int32	Number of the COMport to be opened	1 to 256

Return parameters

Name	Data type	Short description
Signed Int32	TX-Status or error	-1 to 3

Description:
The return value is -1 in the event of an error (then an error code can be retrieved with GSV4getLastError, see Appendix A) or it corresponds to one of the following values:
0: Independent measurement data sending permanently (i.e. non-volatile stored) and up-to-date
1: Independent measurement data sending permanently (i.e. non-volatile stored), but temporarily (i.e. volatile) switched off, because GSV4setTXmode or GSV4stopTX was given accordingly.
2: Independent measurement data sending permanently (i.e. non-volatile stored), but temporarily (i.e. volatile) switched on, because GSV4setTXmode or GSV4startTX was given accordingly.
3: Independent measurement data sending permanently (i.e. non-volatile stored) and up-to-date.

4.6. GSV4setTXmode

This function can be used to change the status of the continuous measurement transmission of the GSV-4.

Used GSV-4 Firmware commands: set_tx_status (0x28)

EEPROM-storage: yes, changed if non-volatile state (see above)

Minimum firmware version number: 0x08

DLL-atomic number (Index): 50

Transfer parameters:

Name	Data type	Short description	Value range
ComNo	Signed Int32	Number of the COMport to be opened	1 to 256
val	Signed Int32	TX status to be set	Flag value 0 to 3

Description:

val: see description of the return parameter of the GSV4getTXmode.

Return parameters

Name	Data type	Short description
Signed Int32	Error status	-1, 0

Description:
if the function succeeds, GSV_OK =0 is returned, otherwise if an error has occurred, GSV_ERROR =-1. In the event of an error, GSV4getLastError can be used to retrieve an error code that is to be interpreted as an indication of a possible cause of the error. See Annex A.

4.7. GSV4startTX

With this function, the independent measurement data transmission is switched on, so that the GSV-4 then sends measured values permanently with the set data rate.

Used GSV-4 Firmware commands: start_transmission (0x24)

EEPROM-storage: No

Minimum firmware version number: 0x04

DLL-atomic number (Index): 8

Transfer parameters:

Name	Data type	Short description	Value range
ComNo	Signed Int32	Number of the COMport to be opened	1 to 256

Return parameters

Name	Data type	Short description
Signed Int32	Error status	-1, 0

Description: see above

4.8. GSV4stopTX

With this function, the independent measurement data transmission is switched off, so that the GSV-4 does not then send measurements independently.

Used GSV-4 Firmware commands: stop_transmission (0x23)

EEPROM-storage: No

Minimum firmware version number: 0x04

DLL-atomic number (Index): 9

Transfer parameters:

Name	Data type	Short description	Value range
ComNo	Signed Int32	Number of the COMport to be opened	1 to 256

Return parameters

Name	Data type	Short description
Signed Int32	Error status	-1, 0

Description: see above

4.9. GSV4firmwareVersion

This function reads the version number of the GSV-4 firmware.

Used GSV-4 Firmware commands: get_firmware_version (0x2B)

EEPROM-storage: No

Minimum firmware version number: 0x04

DLL-atomic number (Index): 22

Transfer parameters:

Name	Data type	Short description	Value range
ComNo	Signed Int32	Number of the COMport to be opened	1 to 256

Return parameters

Data type	Short description	Value range
Signed Int32	Firmwareversion or error	-1 to 255

Description:

It is this version number that is specified as the "minimum firmware version number" in the command descriptions.

4.10. GSV4getMode

This function reads the command mode of the GSV-4. Most commands can only be used if GSV-4 is in command mode 1. The Activation functions GSV4actExt and GSV4activateExtended already set the command mode to 1 if the interface could be opened.

The only commands that can also be used in command mode 0 are:

–GSV4getMode

–GSV4setMode

–GSV4getTXmode

–GSV4getValue

–GSV4firmwareVersion

Used GSV-4 Firmware commands: get_mode (0x27)

EEPROM-storage: No

Minimum firmware version number: 0x04

DLL-atomic number (Index): 21

Transfer parameters:

Name	Data type	Short description	Value range
ComNo	Signed Int32	Number of the COMport to be opened	1 to 256

Return parameters

Data type	Short description	Value range
Signed Int32	Command mode or error	-1 to 2

Description:
Mode =0 means that only the three functions listed above can be executed. In mode 1, all functions accessible to the user can be performed.

4.11. GSV4setMode

This function can be used to set the command mode of the GSV-4. Most commands can only be used if GSV-4 is in command mode 1. The GSV4actExt and GSV4activateExtended activation functions set the command mode to 1 if the interface could be opened. This function can therefore be used to set the command mode to 0 to lock the amplifier against unauthorized use and to reset this state to 1.

The only commands that can also be used in command mode 0 are:

–GSV4getMode

–GSV4setMode

–GSV4getTXmode

–GSV4getValue

–GSV4firmwareVersion

Used GSV-4 Firmware commands: set_mode (0x26)

EEPROM-storage: No

Minimum firmware version number: 0x05

DLL-atomic number (Index): 20

Transfer parameters:

Name	Data type	Short description	Value range
ComNo	Signed Int32	Number of the COMport to be opened	1 to 256
Mode	Signed Int32	Command mode	0 to 1
pw	Pointer to byte string (0x00 terminated)	Password „berlin“	„berlin“

Description:

ComNo: see above

Mode: see above

pw: The password is designed to protect against accidental or unauthorized change in command mode. A pointer is passed to an unsigned Int8 (Char) array terminated with 0x00, containing the following 7 values:

0x62,0x65,0x72,0x6C,0x69,0x6E,0x00

Return parameters

Data type	Short description	Value range
Signed Int32	Error status	-1, 0

Description:
If the function succeeds, GSV_OK =0 is returned, otherwise if an error has occurred, GSV_ERROR =-1. In the event of an error, GSV4getLastError can be used to retrieve an error code that is to be interpreted as an indication of a possible cause of the error. See Annex A.

4.12. GSV4getLastError

This function can be used to retrieve an error code if a previous function call had returned GSV_ERROR = -1.

Used GSV-4 Firmware commands: No

DLL-atomic number (Index): 2

Transfer parameters:

Name	Data type	Short description	Value range
ComNo	Signed Int32	Number of the COMport to be opened	1 to 256

Return parameters

Data type	Short description	Value range
Signed Int32	Error code	0 to 805310464

Description: If the return value is 0, no error has occurred before. Otherwise, see Appendix 1 for a description of the error codes.

4.13. GSV4readAll

With this function, a measured value of all 4 channels can be read from the DLL buffer.

Used GSV-4 Firmware commands: No

DLL-atomic number (Index): 5

Transfer parameters:

Name	Data type	Short description	Value range
ComNo	Signed Int32	Number of the COMport to be opened	1 to 256
out0	Zeiger auf Double	Read measuring value channel 1	-1,05 to 1,05
out1	Zeiger auf Double	Read measuring value channel 2	-1,05 to 1,05
out2	Zeiger auf Double	Read measuring value channel 3	-1,05 to 1,05
out3	Zeiger auf Double	Read measuring value channel 4	-1,05 to 1,05

Description:

out0 to out3 are passed as pointers to previously allocated variables of type Double. If the return value is GSV_TRUE =1, the function here writes the raw values of each measured value of all 4 channels. If there were multiple readings in the buffers, the oldest unread and untitled value is read (see BufSize description in GSV4actExt).

The measured value 1 corresponds to the positive nominal control of the measuring amplifier. The physical value of this corresponds depends on the setting of the input type and the connected sensor.

Example:

The input type AinTypeBridge2 (see GSV4setAnalogueType), i.e. the input type is bridge input with 2mV/V input sensitivity. Then the raw reading 1 corresponds to the bridge deflection 2mV/V.

Return parameters

Data type	Short description	Value range
Signed Int32	Status code	-1 to 1

Description:

GSV_ERROR = -1 indicates that an error has occurred. In this case, GSV4getLastError can be used to retrieve an error code that is to be interpreted as an indication of a possible cause of the error. See Annex A.

GSV_OK = 0 means that the function was executed correctly, but there are no measured values in the buffers. In this case, nothing is written into it after out0, out1, out2 and out3.

GSV_TRUE = 1 means that no error occurred and the function has placed a measured value in out0, out1, out2 and out3.

4.14. GSV4readMultiple

With this function, several readings of a specific or all 4 channels can be read from the DLL buffer. It can also be used to retrieve all the measured values in all 4 DLL buffers, so that they are empty immediately after execution. They are recommended for relatively high data rates of the GSV-4 or when the operating system is busy due to other tasks. Buffer overwriting by the dLL's read thread is very unlikely due to efficient use of this function (see also chap. 3).

Used GSV-4 Firmware commands: No

DLL-atomic number (Index): 7

Transfer parameters:

Name	Data type	Short description	Value range
ComNo	Signed Int32	Number of the COMport to be opened	1 to 256
Chan	Signed Int32	Channel value	0 to 4
out	Pointer at Double Array	Read measuring values	-1,05 to 1,05
count	Signed Int32	Number of measuring values to be read	1 to BufSize
valsread	Pointer at Signed Int32	Number of measuring values actually read	0 to count

Description:

ComNo: see above

Chan: the channel value can be either 0 or 1..4. If it is not 0, it is the channel number of the channel whose readings are to be read. When reading metrics, the oldest value is at the beginning of the array (index 0). If the channel value is =0, all readings of all channels are read. If measurements have been read, the oldest value of channel 1 is at the beginning of the array (index 0), then channel 2 follows, and so on, until index 3. Index 4 is then the second oldest value of channel 1, and so on.

See example below.

If the application evaluates multiple channels, it is highly recommended to use GSV4readMultiple with Chan = 0, even if not all 4 channels are of interest. In the latter case, the irrelevant channels should be ignored or skipped when reading the array. Otherwise, the time allocation of the relevant channels can be very difficult.

out is passed as a pointer to a previously allocated one-dimensional array of type Double. The size of this array should be the highest value of count ever used. The array size cannot be checked by the function. If the array is too small, it can happen that it is written to a "forbidden" memory area, which should be avoided at all times!

If the return value is GSV_TRUE =1, the function writes the raw values of the measured values in here.

The measured value 1 corresponds to the positive nominal control of the measuring amplifier. The physical reading corresponds to this depends on the setting of the input type and the connected sensor, see description for GSV4readAll.

count: Number of maximum readings. If there are more measured values in the buffer than is passed by count, exactly count values are read, otherwise all measured values in the buffer are read. If the measured values of all channels are to be read (Chan =0), count must be divisible by 4, because this parameter indicates the total number of measured values of all channels. Therefore, the allocated array must also be at least count in size, see warning above. Therefore, you often choose the size of the DLL buffers = count or count = BufSize x 4, if Chan =0, so that the function picks up all the measured values in the buffer(s).

valsread: passed as a pointer to a previously assigned variable of type Signed Int32. This always (except in the event of an error) the number is actually stored according to <out>written measured values. If the return value is GSV_TRUE = 1, this value should be evaluated because it indicates the number of measurements written by <out>, i.e. the size of the (lower) part of out that contains measured values.

Return parameters

Data type	Short description	Value range
Signed Int32	Status code	-1 to 1

Description:

GSV_ERROR = -1 indicates that an error has occurred. In this case, GSV4getLastError can be used to retrieve an error code that is to be interpreted as an indication of a possible cause of the error. See Annex A.

GSV_OK = 0 means that the function was executed correctly, but there are no measured values in the buffers. In this case, nothing is written into it after out and <valsread>is also 0.

GSV_TRUE = 1 means that no error occurred and the function has stored readings after <**out**>.

Example 1:

One has been allocated to <out>a double array of size 77.

One passes to GSV4readMultiple Chan =2 and count = 77. The return value is 1 and <valsread>= 3.

Then the following 3 readings were written according to <out>:

{ oldest value channel 2, second oldest value channel 2, latest value channel 2 }

Example 2:

GSV4actExt was called successfully with BufSize = 1024.

One has been allocated to <out>a double array of size 4096.

Pass to GSV4readMultiple Chan = 0 and count = 4096. The return value is 1 and <valsread>= 8.

Then the following 8 readings were written according to <out>:

{ oldest value channel 1, oldest value channel2, oldest value channel3, oldest value channel4, latest value channel 1, latest value channel2, latest value channel3, newest value channel3, latest value channel4 }

4.15. GSV4read

This function can be used to read a measurement of a specific channel from the DLL buffer. The use of this function is only recommended if only one channel is used, because the buffers of the unread channels are overwritten by the read thread of the DLL and their temporal allocation is thus lost.

Used GSV-4 firmware commands: No

DLL (index): 6

Transfer parameters:

Name	Data type	Short description	Value range
ComNo	Signed Int32	Number of the COMport to be opened	1 to 256
Chan	Signed Int32	Channel	1 to 4
out	Pointer at Double	Read measuring value	-1,05 to 1,05

Description:

ComNo: see above

Chan: number of the channel to be read (1 to 4).

out is passed as a pointer to a previously allocated variable of type Double.

If the return value is GSV_TRUE =1, the function writes down the raw value of the measured value to be read. If there are multiple readings in the buffer, the oldest unread and untitled value is read (see BufSize description in GSV4actExt).

The measured value 1 corresponds to the positive nominal control of the measuring amplifier. The physical reading corresponds to this depends on the setting of the input type and the connected sensor, see description for **GSV4readAll**.

Return parameters

Data type	Short description	Value range
Signed Int32	Status code	-1 to 1

Description:

GSV_ERROR = -1 indicates that an error has occurred. In this case, GSV4getLastError can be used to retrieve an error code that is to be interpreted as an indication of a possible cause of the error. See Annex A.

GSV_OK = 0 means that the function was executed correctly, but there are no measured values in the buffer. In this case, nothing is written into it after out.

GSV_TRUE = 1 means that no error occurred and the function has placed a reading in <out>.

4.16. GSV4received

This function can be used to determine how many measured values are in the measured value buffer. The levels of the 4 measured value buffers can only be different if GSV4read or GSV4readMultiple is used with Chan >0, because the GSV-4 sends whole measured frame, which always contains the 4 values of all 4 channels.

Used GSV-4 firmware commands: No

DLL (index): 4

Transfer parameters:

Name	Data type	Short description	Value range
ComNo	Signed Int32	Number of the COMport to be opened	1 to 256
Chan	Signed Int32	Channel value	0 to 5

Description:

ComNo: see above

Chan: If this channel value is =0, the lowest level of the 4 measured buffers is returned. If this value is = 5, the highest level of the 4 measured buffers is returned. Otherwise, it is the number of the channel (1 to 4) whose level is returned.

Return parameters

Data type	Short description	Value range
Signed Int32	Level or error	-1 to 1

Description:

GSV_ERROR = -1 indicates that an error has occurred. In this case, GSV4getLastError can be used to retrieve an error code that is to be interpreted as an indication of a possible cause of the error. See Annex A.

Otherwise, the number of readings in the buffer will return.

4.17. GSV4getValue

This function causes the amplifier to send a measured value frame. The measured values of all 4 channels are then stored in the DLL buffers, where they can be picked up with a read function (e..B g. GSV4readAll). GSV4getValue is therefore particularly suitable if the amplifier does not automatically transmit measured values (see .B. GSV4getTXmode).

Used GSV-4 firmware commands: get_value (0x3B)

EEPROM Storage: No

Minimum firmware version number: 0x04

DLL (index): 23

Transfer parameters:

Name	Data type	Short description	Value range
ComNo	Signed Int32	Number of the COMport to be opened	1 to 256

Description:

ComNo: see above

Return parameters

Data type	Short description	Value range
Signed Int32	Error status	-1, 0

Description:
If the function succeeds, GSV_OK =0 is returned, otherwise if an error has occurred, GSV_ERROR =-1. In the event of an error, GSV4getLastError can be used to retrieve an error code that is to be interpreted as an indication of a possible cause of the error. See Annex A.

4.18. GSV4clearDLLbuffer

This function can be used to delete the contents of all 4 DLL buffers. This function is especially used when no values have been read from the buffers for a long time and those readings that are still in them are considered irrelevant. This can be done, for example.B immediately before entering a reading loop.

Used GSV-4 firmware commands: No

DLL (index): 3

Transfer parameters:

Name	Data type	Short description	Value range
ComNo	Signed Int32	Number of the COMport to be opened	1 to 256

Description:

ComNo: see above

Return parameters

Data type	Short description	Value range
Signed Int32	Error status	-1, 0

Description: see above

4.19. GSV4setZero

With this function, the amplifier is made to set the currently processed measured value of the passed channel (or all 4 channels) to 0. The offset value thus determined is stored by the GSV-4 and added by it to all future measured values. In this way, zero taration can be performed.

Used GSV-4 firmware commands: set_zero (0x0C)

EEPROM Storage: yes

Minimum firmware version number: 0x04

DLL (index): 11

Transfer parameters:

Name	Data type	Short description	Value range
ComNo	Signed Int32	Number of the COMport to be opened	1 to 256
Chan	Signed Int32	Channel value	0 to 4

Description:

ComNo: see above

Chan: If the channel value is =0, all 4 channels are set to zero, otherwise only the one specified by this value.

Return parameters

Data type	Short description	Value range
Signed Int32	Error status	-1, 0

Description: see above

4.20. GSV4readFreq

This function reads the data rate set in the amplifier. This is to be understood as a single sampling rate, i.e. it applies to each of the 4 channels (if it is e.B. =5, 5 values of each channel are sent per second, i.e. a total of 20 values/s).

The GSV-4 cannot fully meet this data rate, especially at higher values of 25 or more, i.e. the actual data rate may be lower than the specified data rate.

The GSV-4 continuously determines measured values with this data rate, regardless of the set TX mode. But only with the continuous data transmission activated, measured values are also sent with this rated data rate.

Used GSV-4 Firmware commands: get_frequency (0x16)

EEPROM-storage: No

Minimum Firmware version number: 0x04

DLL (Index): 15

Transfer parameters:

Name	Data type	Short description	Value range
ComNo	Signed Int32	Number of the COMport to be opened	1 to 256

Description:

ComNo: see above

Return parameters

Data type	Short description	Value range
Double	Data rate or error	-1 or 0,625 to 1000

GSV_ERROR = -1 indicates that an error has occurred. In this case, GSV4getLastError can be used to retrieve an error code that is to be interpreted as an indication of a possible cause of the error. See Annex A.

Otherwise, the data rate is returned.

4.21. GSV4writeFreq

With this function, the data rate set in the amplifier can be changed, see description for GSV4readFreq. Only certain, discrete data rates are set in the amplifier. If the function passes other than those listed below, the closest existing value is set.

Used GSV-4 Firmware commands: set_frequency (0x12)

EEPROM-storage: yes

Minimum Firmware version number: 0x04

DLL order number (Index): 14

Transfer parameters:

Name	Data type	Short description	Value range
ComNo	Signed Int32	Number of the COMport to be opened	1 to 256
Freq	Double	Data rate to be set	0,625 to 250 (937,5)[1]

Description:

ComNo: see above

Freq: The following values are set in the device (all others are approximated to one of these values):

Data frequency in Hz (nominal)	Data frequency in Hz (effectively)
0,63	0,625
1,25	1,250
2,5	2,500
3,75	3,750
6,25	6,250
7,5	7,500
12,5	12,400
15	14,7
25	24,4
125	114
250	208
500	297

[1] The maximum possible data rate is 250 Hz to firmware version 0x0F. The DLL checks this maximum value. From version 0x10 can be passed up to 1000 Hz, it is then set nominally 937.5 Hz. Functional limitations may be expected.

Return parameters

Data type	Short description	Value range
Signed Int32	Error status	-1, 0

Description:
If the function succeeds, GSV_OK =0 is returned, otherwise if an error has occurred, GSV_ERROR =-1. In the event of an error, GSV4getLastError can be used to retrieve an error code that is to be interpreted as an indication of a possible cause of the error. See Annex A.

4.22 GSV4readSerialNo

With this function, the individually assigned serial number of the device copy can be read.

Used GSV-4 Firmware commands: get_serial_number (0x1F)

EEPROM-storage: No

Minimum Firmware version number: 0x04

DLL order number (Index): 17

Transfer parameters:

Name	Data type	Short description	Value range
ComNo	Signed Int32	Number of the COMport to be opened	1 to 256
SerNo	Pointer at Byte-String	Serial number as ASCII string	„07000000“ to „99999999“

Description:

ComNo: see above

SerNo: Pointer to an array to be apreviously approved with the data type Unsigned Int8 (Char/Byte), which must be 9 bytes in size. If executed successfully, the function writes an ASCII-encoded string with 0x00 terminated string, which specifies the serial number.

Return parameters

Data type	Short description	Value range
Signed Int32	Error status	-1, 0

Description: see above

4.23 GSV4DispGetNorm

With this function, a pre-written display standardization can be read, for each channel individually. This value is usually used to calculate physically correctly scaled measured values from the raw values collected with the above-date reading functions by multiplying the raw values by the display standardization. If you want the measurement display to be consistent with the existing evaluator software (e.B.g. GSVmultichannel), it is recommended that this parameter be used.

Used GSV-4 Firmware commands: get_user_scale (0xBB)

EEPROM-storage: no

Minimum Firmware version number: 0x05

DLL order number (Index): 31

Transfer parameters:

Name	Data type	Short description	Value range
ComNo	Signed Int32	Number of the COMport to be opened	1 to 256
Chan	Signed Int32	Channel	1 to 4

Description:

ComNo: see above

Chan: Channel for which display standardization has been saved there.

Return parameters

Data type	Short description	Value range
Double	Display standardization or 0 = error	1,175 E-38 to 3.402 E38

Description:

A value of 0 means (exceptionally) that an error has occurred. In this case, GSV4getLastError can be used to retrieve an error code that is to be interpreted as an indication of a possible cause of the error. See Annex A.

Otherwise, the return value is the stored display standardization.

4.24 GSV4DispSetNorm

With this function, a display standardization can be written for each channel individually. The GSV-4 does not use this value itself, but simply stores it non-volatilely. For a measurement display that matches existing evaluator software (e.B. GSVmultichannel), the use of this parameter is recommended.

Used GSV-4 Firmware commands: set_user_scale (0xBA)

EEPROM-storage: Yes

Minimum Firmware version number: 0x05

DLL order number (Index): 30

Transfer parameters:

Name	Data type	Short description	Value range
ComNo	Signed Int32	Number of the COMport to be opened	1 to 256
Chan	Signed Int32	Channel	1 to 4
Norm	Double	Display standardization	-3,402E38 to 3,402E38 except 0

Description:

ComNo: see above

Chan: Channel for which display standardization is to be saved.

Norm: Displayed standardization passed in. A value of 0 is not allowed, but negative values are not allowed. With these, the direction of the sensor deflection can be virtually reversed without the need to change the sensor connection to hardware.

Return parameters

Data type	Short description	Value range
Signed Int32	Error status	-1, 0

Description:

If the function succeeds, GSV_OK =0 is returned, otherwise if an error has occurred, GSV_ERROR =-1. In the event of an error, GSV4getLastError can be used to retrieve an error code that is to be interpreted as an indication of a possible cause of the error. See Annex A.

4.25 GSV4getGain

This function can be used to read out the setting of the analog input type of the GSV-4 for the channel passed in. This input type should match the output signal type of the connected sensor.

Used GSV-4 Firmware commands: get_gain (0xB3)

EEPROM-storage: No

Minimum Firmware version number: 0x08 (0x09)

DLL order number (Index): 29

Transfer parameters:

Name	Data type	Short description	Value range
ComNo	Signed Int32	Number of the COMport to be opened	1 to 256
Chan	Signed Int32	Channel value	0 to 4

Description:

ComNo: see above

Chan: If channel = 0, the input types of all 4 channels are read. Otherwise (1 to 4) it is the number of the channel whose input type is to be read.

Return parameters

Data type	Short description	Value range
Signed Int32	Input type or error	-1 to 0x07070707

Description:

If an error has occurred, the function returns GSV_ERROR =-1. Then GSV4getLastError can be used to retrieve an error code that is to be interpreted as an indication of a possible cause of the error. See Annex A.

Otherwise, the input type is encoded as follows:

Return-value	Input type = sensor type	Display standardization for sensor	Unit[1]
1	Bridge sensor 2mV/V	2	mV/V
2	Bridge sensor 10mV/V	10	mV/V
3	Voltage input 0..5V	5	V
4	Temperature PT1000	1000	°C
6	Cable temperature sensor type K	1000	°C
7	Voltage input 0..10V	10	V

If the transfer parameter is Chan =0, the above values for the 4 channels within the return value are as follows:

Bits	Channel
31 to 24	1
23 to 16	2
17 to 8	3

[1] The last two columns of this table are not directly relevant to this function. They are specified only to make basic parameterization easier for the user, because the GSV-4 never changes the unit and display standardization itself.

7 to 0	4

Otherwise (Chan = 1..4) the value in bits is 0 to 7.

4.26 GSV4setGain

This function can be used to set the setting of the analog input type of the GSV-4 for the channel passed in. This input type should match the output signal type of the connected sensor.

Used GSV-4 Firmware commands: set_gain (0xB2)

EEPROM-storage: Yes

Minimum Firmware version number: 0x08 (0x09)

DLL order number (Index): 28

Transfer parameters:

Name	Data type	Short description	Value range
ComNo	Signed Int32	Number of the COMport to be opened	1 to 256
Chan	Signed Int32	Channel value	0 to 4
gain	Signed Int32	Input type	1 to 7, except 5

Description:

ComNo: see above

Chan: If channel = 0, the input types of all 4 channels are set to the same value. Otherwise (1 to 4) it is the number of the channel for which the input type should be set.

gain: The input type is encoded in the same way as the return type of the GSV4getGain (see above) function when reading a single channel. Types 6 and 7 are only available from firmware version 0x09.

Return parameters

Data type	Short description	Value range
Signed Int32	Error status	-1, 0

Description:

If the function succeeds, GSV_OK =0 is returned, otherwise if an error has occurred, GSV_ERROR =-1. In the event of an error, GSV4getLastError can be used to retrieve an error code that is to be interpreted as an indication of a possible cause of the error. See Annex A.

4.27 GSV4getAnalogType

This function can be used to read the setting of the analog input type of the GSV-4, as well as with GSV4getGain. However, for each channel, a pointer is passed on which a variable must first be alloated into which the input types are written. For use with C/C++ the enumdata data type (enum) AinType is provided, see header MEGSV4wXX.h.

Used GSV-4 Firmware commands: get_gain (0xB3)

EEPROM-storage: No

Minimum Firmware version number: 0x08 (0x09)

DLL order number (Index): 49

Transfer parameters:

Name	Data type	Short description	Value range
ComNo	Signed Int32	Number of the COMport to be opened	1 to 256
typCh1	Pointer at Signed Int32	Input type channel 1	1 to 7, except 5
typCh2	Pointer at Signed Int32	Input type channel 2	1 to 7, except 5
typCh3	Pointer at Signed Int32	Input type channel 3	1 to 7, except 5
typCh4	Pointer at Signed Int32	Input type channel 4	1 to 7, except 5

Description:

ComNo: see above

typCh1 up to typCh4: Pointers to variables that, if successfully executed, contain the input types of the 4 channels. Coding see GSV4getGain.

Return parameters

Data type	Short description	Value range
Signed Int32	Error status	-1, 0

Description: see above

4.28 GSV4setAnalogueType

This function is almost identical to GSV4setGain, except that the transfer parameters *type* uses the C/C++ enumeration type AinType (see. MEGSV4wXX.h).

Used GSV-4 Firmware commands: set_gain (0xB2)

EEPROM-storage: Yes

Minimum Firmware version number: 0x08 (0x09)

DLL order number (Index): 48

Transfer parameters:

Name	Data type	Short description	Value range
ComNo	Signed Int32	Number of the COMport to be opened	1 to 256
Chan	Signed Int32	Channel value	0 to 4
type	Signed Int32	Input type	1 to 7, except 5

Anything else: see GSV4setGain.

4.29 GSV4saveAll

With this function, a complete parameterization record can be saved as an alternative. The GSV-4 saves all settings as soon as they are changed and restores these settings when turned on. If a different parameterization is desired, or different parameter records, these can be stored in the EEPROM of the GSV-4 using this function. These can then be restored with GSV4getAll.

Used GSV-4 Firmware commands: save_konfiguration (0x09).

EEPROM-storage: Yes

Minimum Firmware version number: 0x04

DLL order number (Index): 12

Transfer parameters:

Name	Data type	Short description	Value range
ComNo	Signed Int32	Number of the COMport to be opened	1 to 256
DataNo	Signed Int32	Record No.	2 to 4

Description:

ComNo: see above

DataNo: Here the number of the data record under which the operating parameters are to be stored and under which they can be restored with GSV4getAll is specified.

Return parameters

Data type	Short description	Value range
Signed Int32	Error status	-1, 0

Description: see above

4.30 GSV4getAll

This function can be used to restore an operating parameter record previously saved with GSV4saveAll. In addition, data no=1 can be used to load manufacturer settings. The loaded settings are also restored (as always all non-volatile settings) after the next power on.

Used GSV-4 Firmware commands: restore_konfiguration (0x0A).

EEPROM-storage: Yes

Minimum Firmware version number: 0x04

DLL order number (Index): 13

Transfer parameters:

Name	Data type	Short description	Value range
ComNo	Signed Int32	Number of the COMport to be opened	1 to 256
DataNo	Signed Int32	Record No.	1 to 4

Description:

ComNo: see above

DataNo: specifies the number of the record to load, where 1 is the manufacturer record that contains default values that are configured only at the factory. Under DataNo 2 to 4, records stored with GSV4saveAll can be loaded beforehand.

Return parameters

Data type	Short description	Value range
Signed Int32	Error status	-1, 0

Description: see above

4.31 GSV4readThreshold

The GSV-4 has 8 configurable threshold switches. These are switched on when the measured value of the input channel associated with the threshold switch exceeds the duty threshold; or switched off when the switch-off threshold is exceeded. The threshold switch should be mapped to one of the digital I/O lines; this can be done with the **GSV4setDigitalIOtype** function. With **GSV4readThreshold**, the on and off thresholds of the specified threshold switch are read out.

Used GSV-4 Firmware commands: get_threshold (0x21).

EEPROM-storage: No

Minimum Firmware version number: 0x04

DLL order number (Index): 19

Transfer parameters:

Name	Data type	Short description	Value range
ComNo	Signed Int32	Number of the COMport to be opened	1 to 256
ThresNo	Signed Int32	Threshold No.	1 to 8
ThresOn	Pointer at Double	Switch-on threshold	-1,05 to 1,05
ThresOff	Pointer at Double	Switch-off threshold	-1,05 to 1,05

Description:

ComNo: see above

ThresNo: The number of the threshold value is assigned to the channels as follows, with two threshold switches each responding to the same channel:

Threshold No.	Channel
1 and 2	1
3 and 4	2
5 and 6	3
7 and 8	4

ThresOn: Pointer to a previously allocated variable of type Double, which contains the one-switch threshold after successful execution. This is the threshold value, which the threshold switch is switched on when exceeded by the measured value of the corresponding channel.

ThresOff: Pointer to a previously allocated variable of type Double, which contains the off switching threshold after successful execution. This is the threshold value at which the threshold value is switched off by the measured value of the corresponding channel.

Both switching thresholds are denormalized, i.e. the raw readings (see **GSV4readAll**) are compared directly with them. If necessary, the threshold raw values read by the function must be multiplied by the display standardization.

Return parameters

Data type	Short description	Value range
Signed Int32	Error status	-1, 0

Description: see above

4.32 GSV4writeThreshold

With this function, the on and off thresholds of the specified threshold switch can be set. This applies to the threshold switches and the threshold values in the description of GSV4readThreshold.

Used GSV-4 Firmware commands: set_threshold (0x20).

EEPROM-storage: Yes

Minimum Firmware version number: 0x04

DLL order number (Index): 18

Transfer parameters:

Name	Data type	Short description	Value range
ComNo	Signed Int32	Number of the COMport to be opened	1 to 256
ThresNo	Signed Int32	Threshold No.	1 to 8
ThresOn	Double	Switch-on threshold	-1,05 to 1,05
ThresOff	Double	Switch-off threshold	-1,05 to 1,05

Description:

ComNo: see above

ThresNo: The number of the threshold value is assigned to the channels as follows, i.e. two threshold switches each respond to the same channel:

Threshold No.	Channel
1 and 2	1
3 and 4	2
5 and 6	3
7 and 8	4

ThresOn: Value of the switch-on threshold. This is the threshold value, which the threshold switch is switched on when exceeded by the measured value of the corresponding channel.

ThresOff: Value of the off switching threshold. This is the threshold value at which the threshold value is switched off by the measured value of the corresponding channel.

Both switching thresholds are denormalized, i.e. the raw readings (see GSV4readAll) are compared directly with them. If necessary, the desired threshold scaled into physical units must be split by display standardization before passing.

Return parameters

Data type	Short description	Value range
Signed Int32	Error status	-1, 0

Description: see above

4.33 GSV4getDigitalIOtype

The GSV-4 has 12 configurable digital I/O lines. This function can be used to read the type of I/O line number passed by parameter. For firmware versions 0x08 and 0x09, there are only 8 I/O lines.

Used GSV-4 Firmware commands: get_digital (0xB7).

EEPROM-storage: No

Minimum Firmware version number: 0x10 (0x08)

DLL order number (Index): 45

Transfer parameters:

Name	Data type	Short description	Value range
ComNo	Signed Int32	Number of the COMport to be opened	1 to 256
DigIO_No	Signed Int32	Digital I/O line number	1 to 12 (8)

Description:

ComNo: see above

DigIO_No: Number of the digital I/O line whose type to read.

Return parameters

Data type	Short description	Value range
Signed Int32	Digital I/O type or error	-1 to 17

Description:
If an error has occurred, the function returns GSV_ERROR =-1. Then GSV4getLastError can be used to retrieve an error code that is to be interpreted as an indication of a possible cause of the error. See Annex A.
Otherwise, the type of the specified digital I/O line is specified as follows:

Type-No	Function	Input/ Output	GSV-4 Register value
0	General entrance	Input	0x00
1	General output, manually switchable	Output	0x01
2	Get Value. At Hi-Flanke, a measured value frame is transmitted	Input	0x02
3	Zero All. For Hi-Flanke, all 4 analog measurement channels are set to	Input	0x0A

Type-No	Function	Input/ Output	GSV-4 Register value
	zero.		
4	Zero Chan 1. For Hi-Flanke, the measured value of analog channel 1 is set to zero.	Input	0x0B
5	Zero Chan 2. For Hi-Flanke, the measured value of analog channel 2 is set to zero.	Input	0x0C
6	Zero Chan 3. For Hi-Flanke, the measured value of analog channel 3 is set to zero.	Input	0x0D
7	Zero Chan 4. For Hi-Flanke, the measured value of analog channel 4 is set to zero.	Input	0x0E
8	Threshold output 1. The line is assigned to the threshold switch 1.	Output	0x11
9	Threshold output 2. The line is assigned to the threshold switch 2.	Output	0x12
10	Threshold output 3. The line is assigned to the threshold switch 3.	Output	0x13
11	Threshold output 4. The line is assigned to the threshold switch 4.	Output	0x14
12	Threshold output 5. The line is assigned to the threshold switch 5.	Output	0x15
13	Threshold output 6. The line is assigned to the threshold switch 6.	Output	0x16
14	Threshold output 7. The line is assigned to the threshold switch 7.	Output	0x17
15	Threshold output 8. The line is assigned to the threshold switch 8.	Output	0x18
16	TrigSyncSlave. On this line, another GSV-4 device configured as a master can trigger the synchronous transmission of a measured frame.	Input	0x0F
17	TrigSyncMaster. When a measured value is transmitted, this GSV-4, configured as a master, controls other devices configured as slaves for synchronous transmission of the measured value frames.	Output	0x10

4.34 GSV4setDigitalIOtype

The GSV-4 has 12 configurable digital I/O lines. This function can be used to set the type of I/O line passed by parameter. For firmware versions 0x08 and 0x09, there are only 8 I/O lines.

Used GSV-4 Firmware commands: set_digital (0xB6)

EEPROM-storage: Yes

Minimum Firmware version number: 0x10 (0x08)

DLL order number (Index): 44

Transfer parameters:

Name	Data type	Short description	Value range
ComNo	Signed Int32	Number of the COMport to be opened	1 to 256
DigIO_No	Signed Int32	Digital I/O line number	0 to 12 (8)
type	Signed Int32 / enum	Type of digital I/O line to be set	0 to 17

Description:

ComNo: see above

DigIO_No: If this parameter is =0, all digital I/O lines are set to the same passed type value. Otherwise, it is the number of the digital I/O line whose type is to be set.

type: This type-No defines the type of digital I/O line to be changed. It is encoded like the return value of the GSV4getDigitalIOtype function, see table above.

Return parameters

Data type	Short description	Value range
Signed Int32	Error status	-1, 0

Description:
If the function succeeds, GSV_OK =0 is returned, otherwise if an error has occurred, GSV_ERROR =-1. In the event of an error, GSV4getLastError can be used to retrieve an error code that is to be interpreted as an indication of a possible cause of the error. See Annex A.

4.35 GSV4getDigitalIOvalue

This function can be used to read the state of the digital I/O lines.

Used GSV-4 Firmware commands: get_digital_port (0xB9, 0xBF)

EEPROM-storage: No

Minimum Firmware version number: 0x10 (0x08)

DLL order number (Index): 47

Transfer parameters:

Name	Data type	Short description	Value range
ComNo	Signed Int32	Number of the COMport to be opened	1 bis 256
DigIO_No	Signed Int32	Nummer der digitalen I/O-Leitung	0 bis 12 (8)

Description:

ComNo: see above

DigIO_No: If this parameter is =0, all (available) digital I/O lines are read. Otherwise, it is the number of the digital I/O line to be read. I/O lines 8 to 12 are only available from firmware version 0x10 (d16).

Return parameters

Data type	Short description	Value range
Signed Int32	I/O state or error	-1 to 0x00000FFF

Description:

If an error has occurred, the function returns GSV_ERROR =-1. Then GSV4getLastError can be used to retrieve an error code that is to be interpreted as an indication of a possible cause of the error. See Annex A.
Otherwise, the return =1 is if a single line is to be read and this is high and =0 if a single line is to be read and that is low. If all lines are to be read, the states in the bits <11:0> are assigned to the return as follows:

Bit	Dig.I/O No.
11	12
10	11
9	10
8	9
7	8
6	7
5	6
4	5
3	4
2	3
1	2
0	1

4.36 GSV4setDigitalIOvalue

This function can be used to set the state of the digital I/O lines if they are configured as output (see GSV4getDigitalIOtype).

Used GSV-4 Firmware commands: set_digital_on_off (0xB8)

EEPROM-storage: Yes

Minimum Firmware version number: 0x10 (0x08)

DLL order number (Index): 46

Transfer parameters:

Name	Data type	Short description	Value range
ComNo	Signed Int32	Number of the COMport to be opened	1 to 256
DigIO_No	Signed Int32	Digital I/O line number	0 to 12 (8)
value	Signed Int32	Value to be set	0 to 1

Description:

ComNo: see above

DigIO_No: If this parameter is =0, all (available) digital I/O lines are set to the same value. Otherwise, it is the number of the digital I/O line to be set. I/O lines 8 to 12 are only available from firmware version 0x10 (d16).

value: Value =0 to set the line(s) to Low and =1 to set it to High.

Return parameters

Data type	Short description	Value range
Signed Int32	Error status	-1, 0

Description:
Wenn die Funktion erfolgreich ausgeführt wurde, wird GSV_OK =0 zurückgegeben, andernfalls, wenn ein Fehler aufgetreten ist, GSV_ERROR =-1. Im Fehlerfall kann mit GSV4getLastError ein Fehlercode abgeholt werden, der als Hinweis auf eine mögliche Fehlerursache zu interpretieren ist. Siehe dazu Anhang A.

4.37 GSV4getUnit

This function allows you to read a previously saved number for each channel that represents a unit. This unit is displayed by existing application programs, such as .B gsVmultichannel.

Used GSV-4 Firmware commands: get_unit (0xB5).

EEPROM-storage: No

Minimum Firmware version number: 0x0A

DLL order number (Index): 41

Transfer parameters:

Name	Data type	Short description	Value range
ComNo	Signed Int32	Number of the COMport to be opened	1 to 256
Chan	Signed Int32	Channel	1 to 4

Description:

ComNo: see above

Chan: Number of the channel whose unit to read.

Return parameters

Data type	Short description	Value range
Signed Int32	Unit number or error	-1 to 42

Description:

If an error has occurred, the function returns GSV_ERROR =-1. Then GSV4getLastError can be used to retrieve an error code that is to be interpreted as an indication of a possible cause of the error. See Annex A.
Otherwise, the unit encoded as follows was read:

Value	Unit
0	mV/V
1	kg

Value	Unit
2	g
3	N
4	cN
5	V
6	µm/m
7	<keine>
8	t
9	kN
10	lb
11	oz
12	kp
13	lbf
14	pdl
15	mm
16	m
17	cNm
18	Nm
19	°C
20	°F
21	K
22	oztr
23	dwt
24	kNm
25	%
26	0/00
27	W
28	kW
29	rpm
30	bar
31	Pa
32	hPa
33	MPa

Value	Unit
34	N/mm²
35	°
36	Hz
37	m/s
38	km/h
39	m³/h
40	mA
41	A
42	m/s²

4.38 GSV4setUnit

This function can be used to store a previously stored number for each channel, which represents the unit. This unit is displayed by existing application programs, such as .B gsVmultichannel. The GSV-4 does not use this value itself, but only stores it non-volatilely.

Used GSV-4 Firmware commands: set_unit (0xB4).

EEPROM-storage: Yes

Minimum Firmware version number: 0x0A

DLL order number (Index): 40

Transfer parameters:

Name	Data type	Short description	Value range
ComNo	Signed Int32	Number of the COMport to be opened	1 to 256
Chan	Signed Int32	Channel	1 to 4
UnitNo	Signed Int32	Unit number	0 to 42

Description:

ComNo: see above

Chan: Number of the channel whose unit is to be set.

UnitNo: Number of the unit to be set. For their encoding so that it is identical to existing application programs, see Return parameters from **GSV4getUnit**.

Return parameters

Data type	Short description	Value range
Signed Int32	Error status	-1, 0

Description:

If the function succeeds, GSV_OK =0 is returned, otherwise if an error has occurred, GSV_ERROR =-1. In the event of an error, GSV4getLastError can be used to retrieve an error code that is to be interpreted as an indication of a possible cause of the error. See Annex A.

4.39 GSV4getUnitText

This function allows you to read a previously saved unit for each channel. This channel-specific unit is displayed by existing application programs, e.B. gsVmultichannel. You can also read a unit (or any text) previously saved as text by the user.

Used GSV-4 Firmware commands: get_unit (0xB5) or get_user_string (0xBD)

EEPROM-storage: No

Minimum Firmware version number: 0x0B

DLL order number (Index): 43

Transfer parameters:

Name	Data type	Short description	Value range
ComNo	Signed Int32	Number of the COMport to be opened	1 to 256
Chan	Signed Int32	Channel value	0 to 4
UnitTxt	Pointer at Byte-String	Pointer to the text to be read	Max string length: 5 Bytes

Description:

ComNo: see above

Chan: If this value is =0, any previously saved text is read. Otherwise, it is the number of the channel whose unit is to be read.

UnitTxt: Pointer to the string to read. This must be a previously aggregated byte array whose length must be at least 6 bytes. After successful execution, a 0x00-terminated byte string is written in here.

Return parameters

Data type	Short description	Value range
Signed Int32	Error status	-1, 0

Description: see above

4.40 GSV4setUnitText

With this function, one unit can be set for each channel. This channel-specific appliance is displayed by existing application programs, e.B. GSVmultichannel, and must correspond to one of the units tabulated in **GSV4getUnit**. Alternatively, any text consisting of Int8 values, which can be a maximum of 5 characters long, can be saved.

Used GSV-4 Firmware commands: set_unit (0xB4) or set_user_string (0xBC)

EEPROM-storage: Yes

Minimum Firmware version number: 0x0B

DLL order number (Index): 42

Transfer parameters:

Name	Data type	Short description	Value range
ComNo	Signed Int32	Number of the COMport to be opened	1 to 256
Chan	Signed Int32	Channel value	0 to 4
UnitTxt	Pointer at Byte-String	Pointer to the text to be written	Max. string length: **5 Bytes**

Description:

ComNo: see above

Chan: If this value is =0, any text is saved. Otherwise, it is the number of the channel whose unit is to be stored.

UnitTxt: Pointer to the string to be saved. Its length is a maximum of 5 bytes and ideally ASCII is encoded. The non-ASCII characters should match the UTF-8 code.

Return parameters

Data type	Short description	Value range
Signed Int32	Error status	-1, 0

Description: see above

4.41 GSV4getCanID

With this function, the CAN ID for the passed number of one of the four different services can be read for GSV-4 CAN devices.

Used GSV-4 Firmware commands: get_can_id (0xC6)

EEPROM-storage: No

Minimum Firmware version number: 0x04

DLL order number (Index): 35

Transfer parameters:

Name	Data type	Short description	Value range
ComNo	Signed Int32	Number of the COMport to be opened	1 to 256
IDno	Signed Int32	CAN service number	1, 2, 5, 6

Description:

ComNo: see above

IDno: Number of the service for which to read the CAN ID. There are the following 4 different services:

CAN-ID:		Description
TX_1	1	Measuring value send (610)
TX_2	2	Send command response (611)
-	-	
-	-	
RX_1	5	Get_Value (CANSYNC 110)
RX_2	6	Receive Commands (111)

Return parameters

Data type	Short description	Value range
Signed Int32	CAN-ID or error	-1 to 2047

Description:
If an error has occurred, the function returns GSV_ERROR =-1. Then GSV4getLastError can be used to retrieve an error code that is to be interpreted as an indication of a possible cause of the error. See Annex A.
Otherwise, the CAN ID was read.

4.42 GSV4setCanID

With this function, the CAN ID for the passed number of one of the four different services can be set for GSV-4 CAN devices.

Used GSV-4 Firmware commands: set_can_id (0xC5)

EEPROM-storage: Yes

Minimum Firmware version number: 0x04

DLL order number (Index): 34

Transfer parameters:

Name	Data type	Short description	Value range
ComNo	Signed Int32	Number of the COMport to be opened	1 to 256
IDno	Signed Int32	CAN service number	1, 2, 5, 6
CanID	Signed Int32	CAN-ID to be set	0 to 2047

Description:

ComNo: see above

IDno: Number of the service for which to set the CAN ID. There are the following 4 different services:

CAN-ID:		Description
TX_1	1	Send reading (610)
TX_2	2	Send Command Response (611)
-	-	
-	-	
RX_1	5	Get_Value (CANSYNC 110)
RX_2	6	Receive Commands (111)

Return parameters

Data type	Short description	Value range
Signed Int32	Error status	-1, 0

Description: see above

Annex A Error codes.

The error codes larger than 0x30.00.00.00 (d805.306.368) are defined by the DLL itself and are triggered by its functions.

Those in the range of 0x20.00.00.00 (d536.870.912) to 0x2F.FF.FF.FF (d805306367) are Windows System Error Codes, which are passed with 0x20.00.00 .00 (bitwise OR).

The underlying ones are Windows System Error Codes. These can be viewed .B online in the MSDN documentation. They were triggered by failed operating system functions.

The first 6 tabulated codes are also generated by the activation functions GSV4actExt and GSV4activateExtended, especially if the connection to the amplifier failed and therefore occur more frequently than the rest.

No. Hex	No. Dez.	Name	Possible cause
0x00000002	2	ERROR_FILE_NOT_FOUND	Incorrect COMport number or error with communication device
0x00000005	5	ERROR_ACCESS_DENIED	1) The COMport is already open by another application or is still open by previously failing to close the COMport. Close any applications that may be accessing this port. If this does not help or (apparently) no such application is open, restart the system. Or 2) The logged-on user does not have the necessary rights to access this port. Please log in as a user with such rights, change the rights, and if you don't know how, contact your system administrator if necessary.
0x0000001F	31	ERROR_GEN_FAILURE	Hardware problem with this port or with the communication device (e..B missing or defective Bluetooth dongle) or problem with the driver of the communication device or incorrect configuration of it. Also possible: (especially with wireless devices

No. Hex	No. Dez.	Name	Possible cause
			such as GSV-3BT and GSV-4BT): Battery empty or not available. Check the port settings in Control Panel or, if available, in the COMMUNICATION device driver's GUI.
0x3000001F	805306399	ERR_COM_GEN_FAILURE	If the ACTEX_FLAG_HANDSHAKE flag was set in GSV4activateExtended: Problem with hardware handshaking or any of the reasons given for error No 31 (see above).
0x30000058	805306456	ERR_NO_GSV_ANSWER	Activation functions: The COMport could be opened correctly, but no GSV4 device was found on it. For read functions: The GSV-4 does not respond within the timeout of about 6 sec. Check the power supply of your GTV and also the communication lines. Certain device settings may make communication with the GSV difficult (e.B. For the GSV-4 with a data rate of 0.625 /s or devices with high data rates, e.B. from 250 /s). In this case, try to open the device with another software (GSVmultichannel or gsv4test.exe) and reduce the data rate, e..B to 12.5 values/s.
0x30000101	805306625	ERR_WRONG_COMNO	The COMport number is less than 1 or greater than 255. Please dial the correct COMport number and try again.
0x300000F6	805306614	ERR_COM_ALREADY_OPEN	The COMport is already open. It cannot be reopened by a second instance.
0x30000100	805306624	ERR_WRONG_PARAMETER	Incorrect function transfer parameters, e.B. value range overrun.
0x30000062	805306466	ERR_NOT_SUPPORTED	A firmware command to be called is not supported by the connected GSV-4. As a rule, its firmware is too old.
0x300000F3	805306611	ERR_MEM_ALLOC	The attempt to allocate PC memory space failed. The allocated memory may be too small.
0x300000F0	805306608	ERR_MUTEXFAILED	The attempt to log on to a protected or managed access (Mutex) from the OS was rejected.
0x300000F1	805306609	ERR_EVENTFAILED	The attempt to log on an event handle from the OS was rejected.
0x300000F5	805306613	ERR_BYTES_WRITTEN	Not enough bytes could be written to the interface.
0x30000059	805306457	ERR_WRONG_ANSWER_NUM	The response frame of the GSV-4 does not appear to contain the defined number of bytes. This may also be due to a missynchronization or a hardware failure.
0x30000060	805306464	ERR_WRONG_ANSWER	The bytes evaluated as payload data in the response frame of the GSV-4 do not match your definition.

No. Hex	No. Dez.	Name	Possible cause
			This may also be due to a missynchronization or a hardware failure.
0x30000061	805306465	ERR_WRONG_FRAME_SUFFIX	The evaluated communication frame of the GSV-4 has an incorrect suffix. This may also be due to a missynchronization or a hardware failure.