VGCSxxx control protocol

The VGCSxxx 200/600 series can be controlled using the VGCSxxx control protocol. This protocol permits the higher system to control the VGCSxxx and read out the measuring data. If more than one VGCSxxx is connected, up to 128 instruments can be on the same bus in parallel. In order to control the VGCSxxx the following steps have to be done before the protocol can be used.

1. Buy a VGCSxxx Win code for your VGCSxxx. Key in this code according to the VGCSxxx Win description located on your VGCSxxx Win disk in the \doc\Softcode_1.00.pdf. Start the VGCSxxx Win and verify, if the VGCSxxx Win is capable to connect to your instrument.

2. Select an instrument address you want to use with your VGCSxxx (1 – 127). Turn of your VGCSxxx, press and hold the keys F1 and F2 (the two right keys underneath the LCD display). Move to the menu item “Enable data comm.”, press ENTER and select “ON” with the + key. Now press ENTER again. The instrument now requests the “Inst. address=”, which is the address you have chosen above. Set the instrument to the desired address with the + and – key and confirm by pressing ENTER again.

3. The instrument is now ready to communicate and can be accessed with the installed address.

NOTE: When this protocol is activated, the VG WIN Software protocol is deactivated. If you wish to use VG Win again, you have to deactivate this protocol (reverse item 2)!

Detailed description of the VGCSxxx control protocol:

Numbers in this Document:	Hex numbers	0x3B
	Decimal numbers	59
	ASCII Chraracters	“;”

Calculated check sum: 256 -((Byte 2 + Byte 3 + Byte 4 + Byte 5 + Byte 6 + Byte 7) and 0xFF), then convert result to 2 ASCII bytes and put MSB to Byte 8 and LSB to Byte 9
Example of request below:
256 – ((0x01 + 0x00 + 0x00 + 0x00 + 0x03 + 0xE8) and 0xFF) = 0x14
Byte 8 = “1”, Byte 9 = “4”

Timing: After issuing a command, start a time out of at least 500ms before reissuing another command. VGCSxxx sends the answer back within this time. If no answer received, the command has not been received.

PC	System that controls the VGCSxxx
VGCSxxx	Instrument of the VGCSxxx series with an individual address
Adr	Actual VGCSxxx address
AdrPC	System address
CMD	Main command 0x00 - 0x7F
Cmd	Command
Data	Return data
CKH	Check sum higher Byte ASCII
CKL	Check sum lower Byte ASCII
CR	Carriage Return
LF	Line Feed
Start	Start character “;”

VGCSxxx control protocol

Request from PC to VGCSxxx

The VGCSxxx control protocol consists of a 11 byte request. A request is issued by the PC to the VGCSxxx. This request has the following format:

In this Example we use the following data:

Address	0x01
CMD	0x00
Cmd	1000.0, 1st Byte (LSB) 0x0, 2nd. Byte 0x0, 3rd.Byte 0x03, 4th.Byte (MSB) 0xE8
CKH	“1”
CKL	“4”
CR	0x0D
LF	0x0A

Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8	Byte 9	Byte 10	Byte 11
Start “:” (0x3B)	Adr (0x01)	CMD (0x00)	1st.Byte Cmd (0x00)	2nd.Byte Cmd (0x00)	3rd.Byte Cmd (0x03)	4th.Byte Cmd (0xE8)	CKH “1” (0x31)	CKL “4” (0x34)	CR 13 (0x0D)	LF 10 (0x0A)

VGCSxxx control protocol

Answer from VGCSxxx to PC

The VGCSxxx has two possibilities to answer the above request if the address (Adr) matched the address previously set in the VGCSxxx. If the address does not match, VGCSxxx **will not answer** at all!

1. The address was ok, but the calculated check sum did not match. The answer has 22 Bytes and looks as follows:

";"	"F"	"E"	"H"	"L"	"E"	"R"	"4"	"A"	CR (0x0D)	LF (0x0A)
";"	"R"	"E"	"T"	"O"	"R"	"E"	"2"	"F"	CR (0x0D)	LF (0x0A)

2. The address was ok, the calculated check sum did match. The answer has 22 Bytes and looks as follows (IEEE Float Data = 304,6 = 0x43984CCD):

Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8	Byte 9	Byte 10	Byte 11
Start ";" (0x3B)	AdrPC (0x00)	CMD (0x80)	1st.Byte Data (0xCD)	2nd.Byte Data (0x4C)	3rd.Byte Data (0x98)	4th.Byte Data (0x43)	CKH "8" (0x31)	CKL "C" (0x34)	CR 13 (0x0D)	LF 10 (0x0A)
Byte 12	Byte 13	Byte 14	Byte 15	Byte 16	Byte 17	Byte 18	Byte 19	Byte 20	Byte 21	Byte 22
";"	"R"	"E"	"T"	"O"	"R"	"E"	"2"	"F"	CR (0x0D)	LF (0x0A)

In this Example we use the following data:

Address PC	0x00
CMD	0x80 = Request CMD with MSBit set
Data	304,6, 1st Byte (LSB) 0xCD, 2nd. Byte 0x4C, 3rd.Byte 0x98, 4th.Byte (MSB) 0x43
CKH	"8"
CKL	"C"
CR	0x0D
LF	0x0A

VGCSxxx control protocol

Usable commands from PC to VGCSxxx

CMD:Cmd **Instrument status**

0x00:100.0 Get instrument status
Data & 0x01 = 0x01, Continuous Mode on
Data & 0x01 = 0x00, Continuous Mode off
Data & 0x02 = 0x02, Temperature compensation on
Data & 0x02 = 0x00, Temperature compensation off
Data & 0x04 = 0x04, Current clamp on
Data & 0x04 = 0x00, Current clamp off
Data & 0x08 = 0x08, Measurement on
Data & 0x08 = 0x00, Measurement off
Data & 0x10 = 0x10, LED Ramp up on
Data & 0x10 = 0x00, LED Ramp up off
Data & 0x20 = 0x20, LED Ramp hold on
Data & 0x20 = 0x20, LED Ramp hold off
Data & 0x40 = 0x40, LED Ramp down on
Data & 0x40 = 0x00, LED Ramp down off
Data & 0x80 = 0x80, LED Error on
Data & 0x80 = 0x00, LED Error off
Data & 0x100 = 0x100, Polarity sense line inverse
Data & 0x100 = 0x000, Polarity sense line normal
Data & 0x200 = 0x200, Polarity clamp line inverse
Data & 0x200 = 0x000, Polarity clamp line normal
Data & 0x400 = 0x400, Result ready (this flag can only be read once, then it is reset)
Data & 0x400 = 0x000, No new result available

Example: Record PC → VGCSxxx (Hex) 3B 01 00 00 00 00 64 39 42 0D 0A
Record VGCSxxx → PC (Hex) 3B 00 80 00 80 80 44 33 43 0D 0A
+ (";RETORE2F"CRLF)

0x00:101.0 Get firmware version
Firmware version = Data (Example: 5.40)

Example: Record PC → VGCSxxx (Hex) 3B 01 00 00 00 00 65 39 41 0D 0A
Record VGCSxxx → PC (Hex) 3B 00 80 CD CC AC 40 46 42 0D 0A
+ (";RETORE2F"CRLF)

0x00:102.0 Get internal board temperature in °C
Internal board temperature = Data (Example: 27.2)

Example: Record PC → VGCSxxx (Hex) 3B 01 00 00 00 00 66 39 39 0D 0A
Record VGCSxxx → PC (Hex) 3B 00 80 00 70 D9 41 46 36 0D 0A
+ (";RETORE2F"CRLF)

VGCSxxx control protocol

CMD:Cmd **Get Measurement values**
0x00:1000.0 Get measuring value
Measurement value [µΩ] = Data (Example: 304,6)

Example: Record PC → VGCSxxx (Hex) 3B 01 00 00 00 03 E8 31 34 0D 0A
Record VGCSxxx → PC (Hex) 3B 00 80 CD 4C D6 43 34 45 0D 0A
+ (";RETORE2F"CRLF)

0x00:1001.0 Get measuring current
Measuring current [A] = Data (Example: 120,0)

0x00:1002.0 Get temperature
Temperature [°C] = Data (Example: 20,0)

0x00:1003.0 Get sense voltage
Sense voltage, last 25 meas. [µV / 10] = Data (Example: 979.4 → 979.4 * 4 /100.0=39.17mV)

0x00:1004.0 Get shunt voltage
Shunt voltage, last 25 meas. [µV / 10] = Data (Example: 979.4 → 979.4 * 4 /100.0=39.17mV)

0x00:1005.0 Get clamp voltage
Clamp voltage, last 25 meas. [µV / 10] = Data (Example: 979.4 → 979.4 * 4 /100.0=39.17mV)

CMD:Cmd **Set instrument status**
0x01:100.0 Set start measurement flag
→ starts the measurement immediately

Example: Record PC → VGCSxxx (Hex) 3B 01 01 00 00 00 64 39 41 0D 0A
Record VGCSxxx → PC (Hex) (";RETORE2F"CRLF)

CMD:Data **Set instrument current**
0x14:100.0 [A] Set the instrument current as a 32 bit float number (5.0A – max measuring current)
→ sets the current of the instrument immediately in the instrument RAM only

Example: Set the Mjölner to 100.0 A measuring current (IEEE float 42C80000)
Record PC → Mjölner (Hex) 3B 01 14 00 00 C8 42 45 31 0D 0A
Record Mjölner → PC (Hex) (";RETORE2F"CRLF)