

EmStatTM

Communications protocol

For use with firmware v7.6

Contents

Contents	1
1 Important to know before you start	3
1.1 Two ways of measuring with EmStat	3
1.2 Idle state.....	3
1.3 Connection viewer in PSTrace	4
1.4 Overview of commands	5
1.5 Receiving from EmStat	6
1.6 Values to be used for different EmStats	6
2 Receiving data from EmStat	7
2.1 'U' packages	7
2.2 'T' packages.....	9
2.3 'P' packages.....	10
3 Remote control of EmStat.....	11
3.1 Fast response mode (J and j)	11
3.2 Set potential (D)	11
3.3 Set cell on/off and current range (G)	12
3.4 Read current, potential or auxiliary analog input on CON2 (a).....	13
3.5 Set analog output line DAC1 on CON2 (d)	14
3.6 Set digital output lines of CON1.....	14
3.7 Set mux channel (m).....	15
3.8 Set mux settings(s)	15
3.9 Read mux info(i).....	16
3.10 Read digital input line of CON2 (r)	16
4 Other commands.....	17
4.1 Receiving EmStat version (t)	17
4.2 Calibrate analog electronics (N)	17
4.3 Read serial number from permanent memory / EEPROM (h).....	18
4.4 Write BiPot mode to permanent memory / EEPROM (W)	19
4.5 Changing the baudrate (B).....	20
5 Setting up and running electrochemical measurements.....	21
5.1 Sending method parameters to EmStat.....	21
5.2 Formatting method parameters	21
5.3 Aborting or pausing a measurement.....	22
5.4 Calculating potential and current range values	22
5.5 Current sampling window; calculating nadmean, d1 and d16	23
5.6 Determining nPoints.....	27
6 Method specific parameters.....	27
6.1 Linear Sweep Voltammetry measurement parameters:	27
6.2 Differential Pulse Voltammetry measurement parameters	28
6.3 Square Wave Voltammetry measurement parameters	29
6.4 Normal Pulse Voltammetry measurement parameters.....	30
6.5 Cyclic Voltammetry measurement parameters (preliminary).....	31
6.6 Measurement sequence for LSV, NPV, DPV and SWV	31
6.7 Amperometric measurement parameters	32
6.8 Pulsed amperometric measurement parameters.....	33
6.9 Open circuit potentiometry (OCP) measurement parameters	34
6.10 Multiple pulse amperometric (MPAD) measurement parameters.....	35
6.11 Measurement sequence for AD, PAD and MPAD.	35
Appendix A: Manual control flow chart	36
Appendix B: Version changes	37

1 Important to know before you start

The EmStat is connected to a PC by means of a standard USB cable or serial (RS-232) cable and to a microcontroller (uC) by means of a serial interface using Tx/Rx.

The default communication port settings are: 230400 Baud,N,8,1 with No handshaking.

The baud rate can be changed, see section 4.5.

Communication is done by sending ASCII characters.

Characters A-F should always be capitals when sending HEX values.

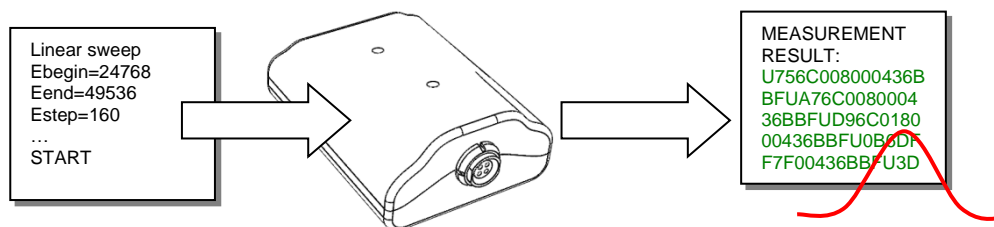
All hexadecimal numbers in this document have prefix 0x.

All byte values are represented as two hexadecimal characters.

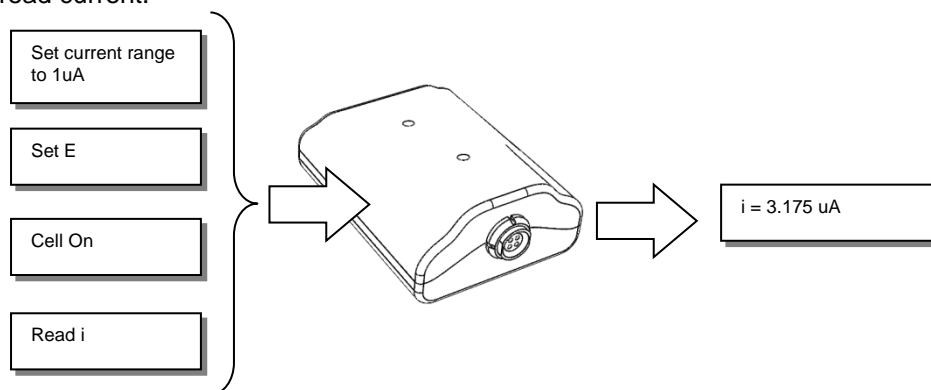
1.1 Two ways of measuring with EmStat

There are two ways of doing a measurement with EmStat;

1. Load EmStat with method parameters (see section 5.1 and section 5.6), and run a pre-defined measurement like squarewave, linear sweep, differential pulse voltammetry, amperometry, etc. EmStat will then run the method autonomous when it has received the 'Start measurement' command (see section 5.1) and send data packages back (see section 2.1). Auto ranging is done automatically if multiple current ranges are enabled for the measurement.



2. Control the cell of EmStat manually by changing the cell state (on/off), current range, set potential or read current.



See section 3 for information about manual control of EmStat.

1.2 Idle state

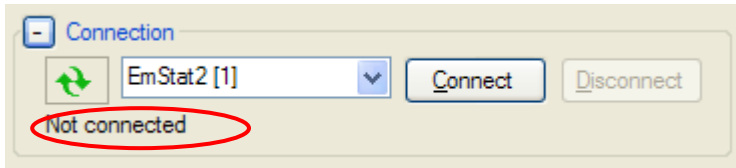
When EmStat is idle it sends a T-package with readings every second (see section 2.2).

EmStat can also be set in a 'Fast Response Mode' where it will respond faster to commands, but will not send the T-packages (see section 3.1).

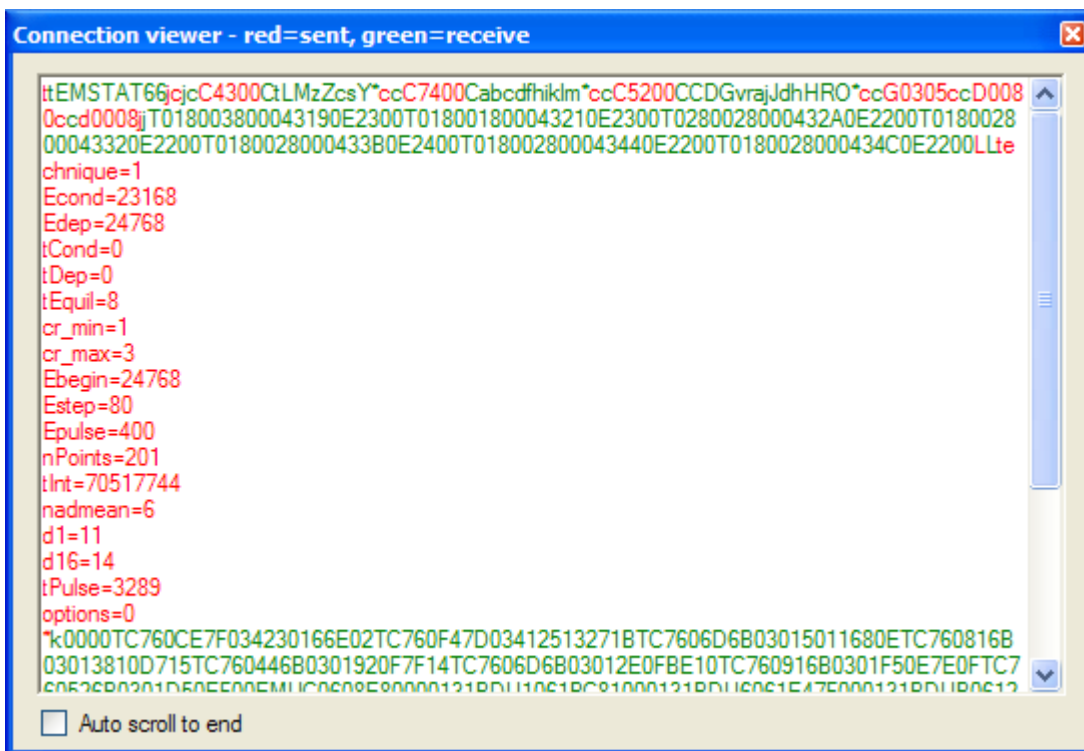
1.3 Connection viewer in PSTrace

PSTrace version 2.6 or higher has a hidden feature, which is useful when the communication protocol is used for development of software for EmStat.

PSTrace will open the 'Connection viewer' window when you double click just below the Connect button, before connecting to EmStat.



Double click on the text in this area before connecting to open the Connection viewer



The connection viewer window. All information in red is sent from the PC to EmStat and the green information is sent by EmStat.

1.4 Overview of commands

EmStat is controlled by the PC or uC using the following characters or commands (without quotes):

Character	Description	See section
'L'	Load EmStat with method parameters from PC.	5.1
'M'	Start measurement using the loaded method, stored in memory	5.1
'Z'	Abort current measurement.	5.3
'z'	Pause LSV, CV or AD or PAD measurement, until 'z' is received again	5.3
'R'	Reverse scan direction during CV measurement	
't'	Receive hardware version, software (firmware) version	4.1
'N'	Calibrate analog electronics	4.2
'J'	Set EmStat in the 'Fast' mode	3.1
'j'	Set EmStat in the 'Normal' mode	3.1
'B'	Set baud	4.5

Manual cell control commands:

Character	Description	See section
'c'	Make EmStat ready to receive remote control command	3 and Appendix A
'D'	Set potential	3.2
'G'	Set cell on/off and current range	3.3
'a'	Read current, potential or analog input signal ADC2 on CON2	0
'd'	Set analog output line DAC1 on CON2	0
'v'	Set digital output lines of CON1 or select channel of MUX	3.6
'r'	Read digital input line of CON2	0

See section 3 for more information about manual control.

1.5 Receiving from EmStat

EmStat sends measurement data preceded by the following characters:

Character	Description	See section
'U'	A single data point is sent: potential, current and status byte	2.1
'T'	Measured values of potential, current, status byte, temperature and noise are sent. These values are measured during idle stage, or conditioning, deposition and equilibration stages	2.2
'P'	EmStat/Mux send measured data of 8 to 16 channels using the method amperometric detection, measuring all channels alternatingly	2.3

Other data sent by EmStat:

'rst'	This string is sent by EmStat when EmStat has been reset, so when the power is switched on or the USB connection is made	
-------	--	--

1.6 Values to be used for different EmStats

This document can be used for EmStat2, EmStat3 and EmStat3+. Some calculations in this document use hardware specific variables.

These variables are for each instrument:

	Efactor	DACfactor	Vrange
EmStat2	1	1	2
EmStat3	1.5	1.599	3
EmStat3+	2	2	4

2 Receiving data from EmStat

2.1 'U' packages

'U' packages can contain measurement results of the following techniques:

- Linear sweep voltammetry LSV
- Differential pulse voltammetry DPV
- Square wave voltammetry SWV
- Normal pulse voltammetry NPV
- Cyclic voltammetry CV
- Amperometric detection AD
- Pulsed amperometric detection PAD
- Open Circuit Potentiometry OCP

'U' is followed by 16 characters of hexadecimal numbers '0' --- '9', 'A' --- 'F'. Hexadecimal numbers in the table below have prefix 0x.

Character	1-4 Potential (except OCP)		5-8 Current (except OCP)		9-10 DPV and SWV	11-12 I status and range	...
	Low Byte	High Byte	Low Byte	High Byte	Correction Byte	IntStatus Byte	...
U	0x00-0xFF	0x00-0xFF	0x00-0xFF	0x00-0xFF	01 or FF	0x00-0xFF	...

...	13-16 Auxiliary input ¹	
...	Low Byte	High Byte
...	0x00-0xFF	0x00-0xFF

Each two characters give an integer in the range 0 to 255 (0x00-0xFF).

Characters 1-4 (Potential)

The first four characters represent the potential

$$E = ((\text{HighByte} * 256 + \text{LowByte}) / 65536 * 4.096 - 2.048) * \text{DACfactor} \text{ (see footnote 2)}$$

For OCP these values represent current and will be all zeros.

Characters 5-8 (Current)

The next four characters represent the current:

$$I = ((\text{HighByte} * 256 + \text{LowByte}) / 65536 * 4.096 - 2.048) * \text{current range.}$$

See "I status and range" for current range.

For OCP these values represent potential values and calculation is similar to characters 1-4, except that the calculation uses Efactor instead of DACfactor.

¹ Available from firmware version 2.5 (or 6.5 for ES2)

² See table in section 1.6

DPV and SWV current correction

The next two characters are used only in DPV and SWV. The current I, as calculated with the two previous characters, is corrected:

if this byte is equal to '01' the current $I = I + (4.096 * \text{current range})$

if this byte is equal to 'FF' the current $I = I - (4.096 * \text{current range})$

I status and range

The next two characters (IntStatus) give the status and the applied current range:

Current range = $10^{(\text{IntStatus AND } 0x0F)}$ in nA units.

Current overload is detected when (IntStatus AND 0x20) is 0x20.

A higher current range is required for proper measurements.

Current underload is detected when (IntStatus AND 0x40) is 0x40.

A lower current range can be applied for better resolution.

2.2 'T' packages

'T' is followed by 20 characters.

Character	1-4 Potential		5-8 Current		9-10 Stage	11-12 I status and range	...
	Low Byte	High Byte	Low Byte	High Byte	Stage Byte	IntStatus Byte	
T	0x00-0xFF	0x00-0xFF	0x00-0xFF	0x00-0xFF	0x00-0x03	0x00-0xFF	...

...	13-16 Auxiliary input ¹		17-20 Noise level	
	Low Byte	High Byte	Low Byte	High Byte
...	0x00-0xFF	0x00-0xFF	0x00-0xFF	0x00-0xFF

Potential

The first four characters represent the potential:

$$E = ((\text{HighByte} * 256 + \text{LowByte}) / 65536 * 4.096 - 2.048) * E_{\text{factor}} \quad (\text{see footnote 2})$$

Current

The next four characters represent the current:

$$I = ((\text{HighByte} * 256 + \text{LowByte}) / 65536 * 4.096 - 2.048) * \text{current range}$$

See "I status and range" for current range.

Stage

These two characters are used to tell the PC or other host the stage of the pretreatment as performed before the actual measurement:

0x00: no pretreatment stage (idle)

0x01: conditioning stage

0x02: deposition stage

0x03: equilibration stage.

I status and range

The next two characters (IntStatus) give the status and the applied current range:

Current range = $10^{\text{IntStatus AND } 0x0F}$ in nA units.

Current overload is detected when (IntStatus AND 0x20) is 0x20.

A higher current range is required for proper measurements.

Current underload is detected when (IntStatus AND 0x40) is 0x40.

A lower current range can be applied for better resolution.

Auxiliary input

Four characters with the reading of the auxiliary analog input channel, which for instance can be connected to a Pt1000 temperature sensing unit.

¹ Available from firmware version 2.5 (or 6.5 for ES2)

² See table in section 1.6

Noise level

The last four characters specify the noise level:

$$\text{Noise} = (\text{LowByte} + \text{HighByte} * 256) / 65536 * 4.096$$

The noise value is given as a mean of the absolute deviation from the average current, expressed in the active current range.

Some examples

Please note that the spaces are added for readability.

T 4A9F 2D9F 00 03 0000 0100 yields :

$$E = 0.501 \text{ V (4A9F)}$$

$$I = 0.4988 \text{ E -06 A (2D9F + 0x03)}$$

$$\text{Noise} = 6.25\text{E-05 (0100)}$$

Status: no overloads or underload (0x03)

T C679 2641 03 02 0000 1B00 yields :

$$E = -0.100 \text{ V (C679)}$$

$$I = -1.0056 \text{ E-07 A (2641 + 0x02)}$$

Stage equilibration (0x03)

$$\text{Noise} = 1.6875\text{E-03 (1B00)}$$

Status: no overloads or underload (0x03)

2.3 'P' packages

'P' packages are only sent with Amperometric Detection using a multiplexer in alternating mode. This means that all channels are scanned quickly at each interval.

P followed by eight or sixteen times eight characters LLMMHHSS where

LL=low byte MM=middle byte HH=high byte

HH is always 00 and reserved for future use.

SS=intStatus with (see also section 2.1):

I status and range

intStatus gives the status and the applied current range:

Low nibble is the status and high nibble is the current range, so:

$$\text{Current range} = 10^{\wedge} (\text{intStatus AND } 0x0F) \text{ in nA units.}$$

Current overload is detected when (IntStatus AND 0x20) is 0x20.

A higher current range is required for proper measurements.

Current underload is detected when (IntStatus AND 0x40) is 0x40.

A lower current range can be applied for better resolution.

For Chronoamperometry in Alternating MUX mode

In case of scanning across all channels during each interval time (alternating MUX mode) the current underload or overload status is determined for each channel.

If only one channel is in overload the next interval will be measured at a higher current range.

If all channels are in underload the next interval will be measured at a lower current range.

Example package P is followed by F5310004 where LL = 0xF5 or 245, MM = 0x31 or 49 and HH = 0 and current range is

$$10^4 \text{ nA or } 10 \text{ uA. So } I = ((49 * 256 + 245) / 65536 * 4.096 - 2.048) * 10 \text{ uA} = -12.49 \text{ uA}$$

P 13010024 0F010024 160E0024 60220004 F5310004 E8400004 A74A0004 18630004

3 Remote control of EmStat

The EmStat is remotely controlled by sending the character 'c'. As soon as EmStat returns a 'c', the actual command can be sent as a character and two integers each of two bytes with a value from 0 (0x00) to 255 (0xFF). Be sure that after 'c' has been returned by EmStat, only a character and two hexadecimal values and no other information is sent to EmStat.

See Appendix A for a flow chart.

3.1 Fast response mode (J and j)

EmStat normally performs continuously measurements of the potential, current and temperature when idle. This means that remote control may have a response of up to 100 ms.

To get a faster response time, the EmStat can be set to 'Fast Response Mode', with the 'J' command. In this mode no idle packages will be sent. If the command 'j' is sent, the EmStat returns to its normal remote-control mode.

3.2 Set potential (D)

Setting the potential is done by sending the character 'D' followed by 4 characters:

Command	Low Byte intL	High Byte intH
D	0xFF	0xFF

The potential is set by sending a 16 bit value.

Set E with setting intH/intL as high/low bytes respectively. The values intH and intL are calculated according to the algorithm given below.

$$\text{intInt} = (\text{Vapplied} / \text{DACfactor}^1 + 2.048) * 16000$$

$$\text{intH} = \text{intInt} / 256 \quad \text{'calculate high byte as integer'}$$

$$\text{intL} = \text{intInt} - 256 * \text{intH} \quad \text{'calculate low byte as integer'}$$

The range of Vapplied is: $-2.048 * \text{DACfactor}$ to $2.047 * \text{DACfactor}$.

The applied potential E is equal to Vapplied if the mode is potentiostatic.

Example: Send 'c', wait until 'c' has been received back and then send 'D0080'.
 Set E to 0.000 V

Note: The previous versions of the firmware used the instrument as a 12-bit device. So E was set with intInt in the range 0 – 4095.

¹ See table in section 1.6

3.3 Set cell on/off and current range (G)

Setting the cell and current range is done by sending the character 'G' followed by 4 characters:

Command	Current range Byte	Set Cell Byte
G	0x00 - 0x07	ON: 0x03 OFF: 0x05

The valid settings of intCR for EmStat are:

- 0: 1 nA
- 1: 10 nA
- 2: 100 nA
- 3: 1 uA
- 4: 10 uA
- 5: 100 uA

for EmStat2 and 3 also

- 6: 1 mA
- 7: 10 mA

for EmStat 3+ also

- 8: 100 mA

The valid settings of intCellOnOff are:

- 5: Cell off
- 3: Cell on

Some examples

- G0205 cell off and 100 nA
- G0203 cell on at 100 nA
- G0403 cell on at 10 uA

3.4 Read current, potential or auxiliary analog input on CON2 (a)

The analog input signal on a specific channel (ichannel) is read as following:

Command	ichannel Byte	Dummy Byte
a	0x00 - 0x03	0xFF

The applicable channels are:

0: current

1: potential

2: auxiliary input (pin 9 CON 2)

Return value:

Character	Low Byte intL	High Byte intH
a	0x00 - 0xFF	0x00 - 0xFF

The value (V_{in}) which is returned is an average of 64 conversions done in 20 ms (50 Hz).

$$V_{in} = (\text{intH} * 256 + \text{intL}) / 65536 * 4.096$$

Channel 0 – current:

If channel 0 is used, the current value in the active current range is: $V_{in} - 2.048$

Channel 1 – potential:

The potential read in Volt from channel 1 is: $(V_{in} - 2.048) * E_{factor}^1$

Channel 2 – analog input:

The analog input signal in Volt is equal to V_{in} .

Some examples

Reading current and selecting optimum current range

Read the output voltage of the current amplifier, which is connected to channel 0.

Send 'c', wait until 'c' has been received back and then send 'a0000'.

Please note that a reasonable selected current range yields a minimum reading of + or - 0.05 V and a maximum reading of + or - V_{Range}^2 . If the reading is less than 0.05 V, use a lower current range and when it exceeds for instance $0.9 * V_{range}$, use a higher current range.

Reading auxiliary input voltage

Read the input voltage at pin 9 of CON2, which is connected to channel 2:

Send 'c', wait until 'c' has been received back and then send 'a02FF'.

¹ See table in section 1.6

² See table in section 1.6

3.5 Set analog output line DAC1 on CON2 (d)

The analog output line DAC1 on CON2 is set as following:

Command	Low Byte intL	High Byte intH
d	0x00 - 0xFF	0x00 - 0xFF

Set analog output (pin 10 of CON2) with setting intH/intL as high/low bytes respectively. The values intH and intL are calculated according to the algorithm given below. The range of $V_{applied}$ is 0 V to 4.095 V.

$intInt = V_{applied} * 16000$

$intH = \text{Int}(intInt / 256)$ 'calculate high byte as integer

$intL = \text{Int}(intInt - 256 * intH)$ 'calculate low byte as integer

Example

Set analog output to 2.048 V:

Send 'c', wait until 'c' has been received back and then send 'd0080'.

Note: The previous versions of the firmware used the instrument as a 12-bit device. So DAC1 was set with intInt in the range 0 – 4095.

3.6 Set digital output lines of CON1

The digital output lines (pin 4-1 on CON1) are set as following:

Command	Low Byte intOut	High Byte intDummy
v	0x00 - 0x0F	0xFF

Set external digital output lines with setting intOut. Only d0, d1, d2 and d3 are relevant.

The range of intOut therefore is 0 to 15. The value of intDummy is not relevant and can have any value from 0 to 255. Setting the digital outputs should not be done when a mux is connected to the EmStat.

Example

Send 'c', wait until 'c' has been received back and then send 'v07FF'.

Stirrer control

In case a stirrer is controlled by EmStat, d0 and d1 are used.

The stirrer is switched on by sending: 'v0100'

The stirrer is switched off by sending: 'v0200'.

3.7 Set mux channel (m)

The mux channel is set as following:

Command	Low Byte byteChan	High Byte byteDummy
m	0x00 - 0xFF	0xFF

Set the mux channel by sending the channel number in byteChan. The range of byteChan is 0 to 255, although the actual amount of supported channels depend on the mux (or muxes) that is connected. Supports both Rev1 and Rev2 muxes. To use a Rev2 mux, the "Set mux settings(s)" command must be used to set the mux type to a Rev2 mux, before using the "Set mux channel(m)" command.

Example

Send 'c', wait until 'c' has been received back and then send 'm0300'. This will set the mux to channel 3.

3.8 Set mux settings(s)

The mux settings are set as following:

Command	Low Byte intL	High Byte intH
s	0x00 - 0xFF	0xFF

Where intL and intH are the low and high bytes of a 16 bit settings field. The settings field of this command is only supported by the Rev2 mux.

Settings field:

Bit 16-13	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1 - 0
Mux type	OCP mode	Common RE/CE	RE to CE	Use sense lead	Unselected WE setting
0 = Rev1 Mux 1 = Rev2 Mux	0 = OFF 1 = ON	0 = OFF 1 = ON	0 = OFF 1 = ON	0 = ON 1 = OFF	0 = Floating 2 = Ground 3 = VStandby

Mux type:

Sets the type of mux that is connected. Options are a Rev1 mux or a Rev2 mux.

OCP mode:

Decreases switching time if turned on, useful for OCP measurements.

Common RE/CE:

If turned on, all channels with use the RE and CE leads of channel 1.

RE to CE:

If turned on, RE will be internally connected to CE. Effectively disabling the reference.

Use sense lead:

If turned on, the sense lead will be enabled (otherwise, it is internally connected to WE, disabling it).

Unselected WE setting:

Determines what happens to the WE of unselected channels.

Example

Send 'c', wait until 'c' has been received back and then send 's0801'. This will set the mux to a Rev2 mux and connect RE to CE.

3.9 Read mux info(i)

The mux information is read as following:

Command	Low Byte intDummy	High Byte intDummy
i	0x0	0x0

The parameters are dummy values.

Return value:

Command	Byte 0 intLowID	Byte 1 intHighID	Byte 2 intLowNChan	Byte3 intHighNChan
i	0x0-0xFF	0x0-0xFF	0x0-0xFF	0x0-0xFF

Where intLowID and intHighID are part of a 16 bit mux type ID. This ID will be "15291" for a Rev2 mux. The bytes intLowNChan and intHighNChan form a 16 bit number that contains the number of available channels. This command is only supported by Rev2 muxes.

Example

Send 'c', wait until 'c' has been received back and then send 'i0000'. A return value of 'iBB3B0800' means that this is a Rev2 mux that has 8 channels.

3.10 Read digital input line of CON2 (r)

The state of the digital input line (pin 5 on CON2) is read as following:

Command	Low Byte intDummy	High Byte intDummy
r	0xFF	0xFF

The status of the input line is returned as following:

Only d0 of intDigIn is relevant and can be either 0 or 1. The value of intDummy is not relevant and can have any value from 0 to 255.

Example

Send 'c', wait until 'c' has been received back and then send 'rFFFF'. The results is returned by means of the string 'r0000' or 'r0100'.

4 Other commands

4.1 Receiving EmStat version (t)

The EmStat returns the model and firmware version when it receives the character 't'.
EmStat's reply depends on the model:

EmStat1 and 2:	"EMSTAT###" (if version is <6.2, model is EmStat1, otherwise EmStat2)
EmStat3:	"EMST 3 ###"
EmStat3+:	"EMST3P###"

represents the firmware version without the decimal point, e.g. when ## = 66, the firmware version is 6.6.
REMARK: Sending a 't' will also set the **cell off** and the instrument in **idle** mode.

4.2 Calibrate analog electronics (N)

The offsets of the potential setting and the current measurement are calibrated. The calibration data are stored permanently until a new command 'N' is received or when the firmware is updated.
All instruments provided by PalmSens BV are already calibrated and this command should only be used in case unwanted current or potential offsets are observed. Leave the cell cable unconnected when calibrating.

4.3 Read serial number from permanent memory / EEPROM (h)

EmStat stores some information in its EEPROM, for example the device serial number. This can be useful for identification when more than one EmStats are connected to the same PC or controller.

The serial number can be retrieved by sending the manual control command 'h0001'.

A package of four 8-bit values is returned in the following order:

Header character	Byte value (2 chars)	Byte value (2 chars)	Byte value (2 chars)	Byte value (2 chars)
'h'	Serial no. HB	Serial no. LB	Batch letter	Year

The first two bytes make up the 16-bit serial number.

The Batch letter is returned as 8-bit value which correlates to the position in the Alphabet where A=1 and Z=26. So for example 0x11 corresponds to the batch letter 'Q'.

The Year number is returned as 8-bit value which correlates to the production year starting from 2000. So 0x0F corresponds to the year 2015.

Example:

Send 'c', wait until 'c' is returned and then send 'h0001'.

EmStat returns:

'hEC00110F'

Which corresponds to serial no 236, batch Q, production year 2015.

4.4 Write BiPot mode to permanent memory / EEPROM (W)

The function of this command depends on the firmware build running on the EmStat.

If the EmStat returns an identification (see 4.1) of 'EMST3B##' where ## is version the firmware is the special BiPot build.

Mode settings for 'BiPot build' of the firmware

In case the special BiPot build of the firmware is running, two modes are applicable for each 'BiPot' board:

Mode 0 = E at independent constant potential

Mode 1 = E at dc-offset from main board E(WE1)

The mode is set by sending

'W', intMode, intDummy

Where intMode can be 0 or 1

Reading the setting back, can be done using the 'h' command (see 0)

Example:

Send 'c', wait until 'c' is returned and then send 'h0004' returns the mode in the first byte

Mode settings for normal build of the firmware

If a value for intMode of 2 is sent to EmStat, the board will send pulses at each measurement interval on d3 (CON1 – PIN1). Each pulse lasts about 10 clock cycles so it can trigger an interrupt on connected BiPot-EmStats. This pulse is used to synchronize measurements between a normal EmStat and additional BiPot-EmStat's and is used in a polypotentiostat.

Enable pulsating on d3:

'W', 2, intDummy

The setting is saved in the permanent part of its memory.

4.5 Changing the baudrate (B)

EmStat works with a default baudrate of 230400 bps. Some microcontrollers do not support this baud rate and can only go up to 115200 bps.

The EmStat's baudrate can be changed by sending

'B', b, intDummy

Where b can be:

b	Corresponding baudrate
0x05	9600
0x06	19200
0x07	38400
0x08	57600
0x09	115200
0x0A	230400 (default)

Example:

Send 'c', wait for returned 'c'
Send 'B0800' to set the device at 57600 baud.

Or:

Send 'c', wait for returned 'c'
Send 'B0A00' to set the device at 230400 baud.

If the new baudrate settings are received and confirmed EmStat will reply 'OK' at the old baudrate before switching to the new baudrate settings.

If the new settings contain an error a '?' is returned.

Other settings for the serial port are:

- Mode 1
- 8 bit
- no parity
- 1 stop bit

The new baudrate settings are stored in the permanent memory of EmStat.

5 Setting up and running electrochemical measurements

EmStat supports the following built-in electrochemical techniques:

- Linear sweep voltammetry (LSV), technique=0
- Differential pulse voltammetry (DPV), technique=1
- Square wave voltammetry (SWV), technique=2
- Cyclic voltammetry (CV), technique=5

- Amperometric detection (AD), technique=7
- Pulse Amp. Detection (PAD), technique=8
- Open Circuit Potentiometry (POT), technique = 10
- Multiple Pulse Amperometric Detection (MPAD), technique= 11

5.1 Sending method parameters to EmStat

An electrochemical measurement is performed by first sending a complete set of parameters to EmStat. EmStat needs to be prepared to receive the method parameters. This is done by sending an 'L' while EmStat is idle. EmStat returns an 'L' to say it's ready. As soon as all parameters are sent and a '*' is sent to mark the end of the method, EmStat will start measuring. If EmStat cannot process one of the parameters it will reply with a '?'.

If the proper method has been sent, only 'M' is required to start a new measurement with the same parameters.

**It's strongly recommended to use the
Connection viewer in PStace. See section 1.3.**

5.2 Formatting method parameters

Measurement parameters are sent as simple key-value data after a 'L' character is received. The format for each parameter is:

NAME = VALUE '\n'

Where '\n' is the end-of-line character (ASCII 0x10).

The sequence of parameters should be terminated by sending a simple '*' to indicate the end of the measurement parameters. The order of the parameters is not relevant. Unspecified parameters remain unchanged.

Example method parameters for Differential Pulse Voltammetry:

technique=1	<i>(Differential Pulse Voltammetry)</i>
Econd=23168	<i>(-0.6 V)</i>
Edep=24768	<i>(-0.5 V)</i>
tCond=5	<i>(5 seconds)</i>
tDep=5	<i>(5 seconds)</i>
tEquil=2	<i>(2 seconds)</i>
cr_min=0	<i>(1 nA)</i>
cr_max=2	<i>(100 nA)</i>
cr =1	<i>(Start at 10 nA)</i>
Ebegin=24768	<i>(-0.5 V)</i>
Estep=80	<i>(0.005 V)</i>
Epulse=400	<i>(0.025 V)</i>
nPoints=201	

```

tInt=68881734      (scanrate of 0.05 V/s)
nadmean=6         "
d1=0              "
d16=0             "
tPulse=2355       (tPulse of 0.05 s)
options=0         (no peripherals or auxiliary)
*
```

See section 5.6 for all method specific parameters.

5.3 Aborting or pausing a measurement

As soon as any measurement has started, it can be aborted at any time by sending a 'Z'.

The measurement techniques LSV, CV or AD or PAD can be paused during a measurement by sending a 'z'. The measurement will be paused until 'z' is received again

5.4 Calculating potential and current range values

Calculation of some potential values

In the next calculation E can be Ebegin, Evtx1, Evtx2, Econd, Edep or Estby:

```
UInt32 UIntValue = (UInt32) ((E / DACfactor1 + 2.048) * 16000.0)
```

Calculation of potential step and pulse or amplitude values

In the next calculation E can be Estep or Epulse:

```
Int32 IntValue = Sign(Eend - Ebegin) * (Int32)(E / DACfactor1 * 16000.0);
if (E < 0.0)
{
  IntValue += 0x10000;
}
```

Where:

```

>0: Sign returns 1
=0: Sign returns 0
<0: Sign returns -1
```

The value 16000 is derived from *bit range / Vref*, which is 65536 / 4.096.

Note that the signed integer values should be sent as unsigned integers. The "if" statement converts negative integers to positive integers.

Current ranges:

The EmStat allows autoranging of the current range (CR).

In the next example CR is the initial, lowest or highest current range, specified in uA. When cr_min is lower than cr_max, autoranging is specified between the specified ranges cr_min and cr_max.

```
IntCR = (Log (CR) / Log (10) + 3.5)
```

(3.5 in fact can be 3, but is used in order to prevent an error in the conversion to integers).

So: intCR = 0 : CR = 1 nA (10⁻³ uA) and intCR=4: CR = 10 uA.

¹ See table in section 1.6

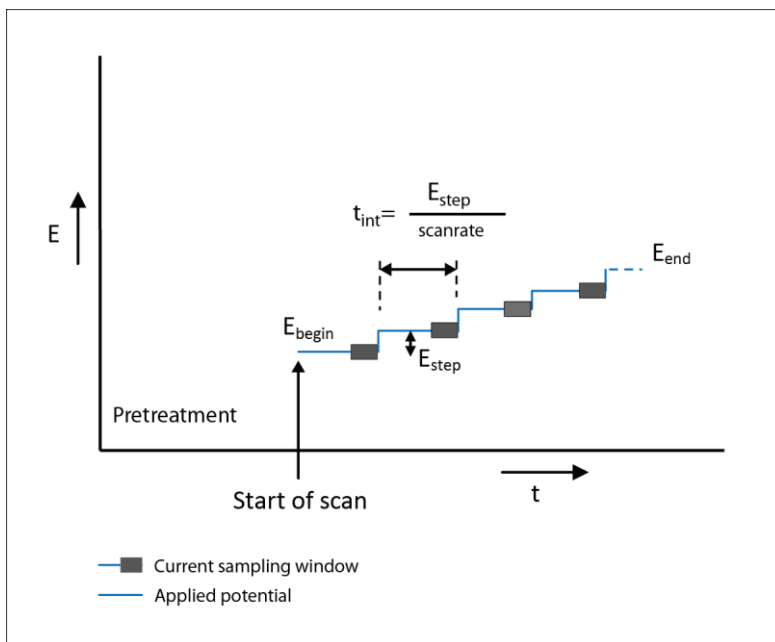
5.5 Current sampling window; calculating nadmean, d1 and d16

The current is sampled during a specified sampling time. During this time as many analog-to-digital conversions as possible are done.

If the interval time is long enough, the current is sampled during an integer number of the cycle time of the power supply. This power supply cycle time is 0.020 s for 50 Hz regions and 0.016667 for 60 Hz regions.

Sampling time during each interval: tmeas

The advised sampling time tmeas is a fraction of the interval or pulse time.



Sampling window during a Linear Sweep scan

We advise to use a fraction of 1 / 2 or 1 / 3. So for the available techniques you can use:

$t_{int}/2$ or $t_{int}/3$ (LSV, CV, OCP and AD)

$t_{pulse}/2$ or $t_{pulse}/3$ (DPV, NPV and PAD)

$1/(4 * freq)$ or $1/(6 * freq)$ (SWV)

So for instance for DPV with a pulse time of 0.06 s, we can have a sampling time tmeas of $0.06s / 3 = 0.02$ s.

Calculating the value for tmeas: nadmean, d1 and d16

The value of tmeas has to be used to calculate the values of:

nadmean, d1 and d16

The actual sampling time sampling or measurement time of the current (or potential) is equal to $t_{meas} = 2^{nadmean} * (0.222 + d1 * 0.0076 + d16 * 0.0005)$

The result for tmeas is in ms.

See next page for a code example in C#.

A possible algorithm to calculate the values of nadmean, d1 and d16 is (C# code):

```
private void ADTimingEmstat(ref float tmeas)
{
    byte nadmean, d1, d16;           // Values to calculate and send to EmStat
                                     // that define current sampling window
    double ADT16ad;

    if (PowerFreq == 60)           // Where PowerFreq is the mains frequency in Hz
    {                               // Normally 50 Hz in Europe
        ADT16ad = 0.0002604;
        d1 = 5;
        d16 = 1;
    }
    else
    { // Power frequency = 50
        ADT16ad = 0.0003125;
        d1 = 11;
        d16 = 14;
    }

    if (tmeas < PowerLinePeriod) // The sampling time is shorter than
    {                               // PowerLinePeriod (1/PowerFreq)
        ADT16ad = 0.000222;       // So PowerLinePeriod=0.020 for 50 Hz
        d1 = 0; d16 = 0;
    }

    int ncycles = (int)(tmeas / ADT16ad);

    if (ncycles < 1)
        ncycles = 1;              // Prevent errors

    //Where Math.Log calculates natural log:
    nadmean = Convert.ToByte((int)(Math.Log(ncycles) / Math.Log(2.0)));

    if (nadmean < 0)
        nadmean = 0;
    if (nadmean > 11)
        nadmean = 11;

    // the following value can be used to verify the result:
    tmeas = Convert.ToSingle((Math.Pow(2.0, nadmean)) * ADT16ad);

    // so tmeas is now the actual sampling time based on the device resolution.
}
}
```

Some notes:

With nadmean = 6, d1 = 11 and d16 = 14, the sampling time is 20.0 ms (for 50 Hz power supplies).
 With nadmean = 6, d1 = 5 and d16 = 1, the sampling time is 16.67 ms (for 60 Hz power supplies).

The maximum value for nadmean = 11, so the maximum sampling is $2^{(11-6)} * 20$ ms or 16.67 ms, so 0.64 s or 0.53 s.

**It's strongly recommended to use the
 Connection viewer in PSTrace. See section 1.3.**

Interval time (tInt):

The time between two measurements or two potential settings or two pulses is called interval time or tInt. This interval time is for scanning methods equal to Estep / scan rate. The parameter tInt is a four byte integer and is used for two different ranges:

- Region 1 with an advised minimum interval time of 3 ms and a maximum of 0.98 s or less
- Region 2 with a minimum interval time of 1/128 s and a maximum of hours. The resolution of the interval times is limited to discrete values with sometimes a poor resolution.

Region 1:

The parameter is specified by four bytes:
Rangebyte, Highbyte, Middlebyte and Lowbyte
This region is used when Rangebyte is set to 4.

The total interval time in s equals: $\text{Highbyte} * (65536 - (\text{Middlebyte} * 256 + \text{Lowbyte})) / 16.777E6$
Thus giving a maximum value of $255 * 65535 / 16.777E6 = 0.996$ s
So the heximal representation is given by: 04 HB MB LB (with HB, MB or LB is from 00 to FF)

Examples:

Interval time	Decimal value	Hexadecimal value
1 / 100 s	67394601	04 04 5C 29
1 / 2 s	75563516	04 81 01 FC

Region 2:

The parameter tInt has a range of 1/128 s to hours and is specified by four bytes:
Rangebyte, Highbyte, Middlebyte and Lowbyte
The value of Highbyte and Middlebyte are set to 0.

Range:

Hours: This region is used when Rangebyte is set to 3.

The interval time equals Lowbyte hours.

Minutes: This region is used when Rangebyte is set to 2.

The interval time equals Lowbyte minutes.

Seconds: This region is used when Rangebyte is set to 1.

The interval time equals Lowbyte seconds.

1/128 s to 1 s: This region is used when Rangebyte is set to 0.

The interval time equals Lowbyte/128 s.

Examples:

Interval time	Decimal value	Hexadecimal value
1 s	128	00 00 00 80
5 s	16777221	01 00 00 05

A possible algorithm to calculate tInt (C# code):

```

private UInt32 CalctInt(float duration)
{
    UInt16 T2HL = 0;
    UInt16 T2M = 0;
    UInt32 value = 0;
    float T2Value;
    float CoreFreq = 16.7772E+06F;           // clock frequency

    if (duration < 0.98) // maximum interval time is 255 * 65536 / 16.7772E6 =
                          // 0.996 s
    {
        T2Value = duration * CoreFreq;
        T2M = Convert.ToUInt16(T2Value / 65536 + 1);
        // set T2 to highest possible values, in order to reduce ISR rate.
        if (T2M > 255)
        {
            T2M = Convert.ToByte(255);
        }
        if (T2M < 1)
        {
            T2M = Convert.ToByte(1);
        }
        T2HL = Convert.ToUInt16(65536 - T2Value / T2M);
        value = (4 << 24);
        value += Convert.ToUInt32(T2M * 65536);
        value += T2HL;
    }
    else
    {
        if (duration / (60 * 60) >= 0x100)
        {
            throw new OverflowException();
        }
        else if (duration / 60 >= 0x100)
        {
            value = 3 << 24; //Hours
            value += Convert.ToByte(duration / (60 * 60));
        }
        else if (duration >= 0x100)
        {
            value = 2 << 24; //Minutes
            value += Convert.ToByte(duration / 60);
        }
        else if (duration * 128 >= 0x100)
        {
            value = 1 << 24; //Seconds
            value += Convert.ToByte(duration);
        }
        else
        {
            value = 0; //128th seconds
            value += Convert.ToByte(duration * 128);
        }
    }
    return value;
}

```

5.6 Determining nPoints

nPoints is the number of current samples (or potential in case of OCP) that will be measured. So if a begin potential of 1.000V is specified with a step potential of 0.1V and a scan should be made up to 2.000V then nPoints should be 11.

6 Method specific parameters

The following sections give an overview of all applicable method parameters.

6.1 Linear Sweep Voltammetry measurement parameters:

The following parameters are supported:

Name	Description	Type
technique=0		
Econd (0-65535)	Conditioning potential	UInt16
tCond (0-65535)	Conditioning Time in units of s	UInt16
Edep (0-65535)	Deposition potential	UInt16
tDep (0-65535)	Deposition time in units of s	UInt16
tEquil (0-65536)	Equilibration time in units of s	UInt16
cr_min (0-5 or 0-7))	Lowest current range	UInt8
cr_max (0-5 or 0-7))	Highest current range	UInt8
cr (0-5 or 0-7))	Starting current range	UInt8
Ebegin (0-65535)	Measurement potential	UInt16
Estep (0-65535)	Step potential	UInt16
nPoints (1-65535)	Number of points	UInt16
ChAux (0-12)	Aux input channel, measured when options bit set	UInt8
Estby (0-65535)	Standby potential	UInt16
tlnt (see description)	Interval time	UInt32
d1(0-255)	Used for ADC's	UInt8 (default: 11)
d16 (0-255)	Used for ADC's	UInt8 (default: 14)
options (0-65535)	Misc. boolean settings. Currently supported: 0x4: Cell on after measurement, with E = Estby 0x8: Stirrer present 0x400: Measure Aux. Input	UInt16
nadmean (0-11)	Used for ADC's	UInt8 (default: 6)

Notes:

1. The interval time tlnt is equal to potential step / scan rate.
2. The sampling time t(meas) can be set to half the interval time or less.
So for instance t(meas) = tlnt / 2.

6.2 Differential Pulse Voltammetry measurement parameters

The following parameters are supported:

Name	Description	Type
technique=1		
Econd (0-65535)	Conditioning potential	UInt16
tCond (0-65535)	Conditioning Time in units of s	UInt16
Edep (0-65535)	Deposition potential	UInt16
tDep (0-65535)	Deposition time in units of s	UInt16
tEquil (0-65536)	Equilibration time in units of s	UInt16
cr_min (0-5 or 0-7))	Lowest current range	UInt8
cr_max (0-5 or 0-7))	Highest current range	UInt8
cr (0-5 or 0-7))	Starting current range	UInt8
Ebegin (0-65535)	Measurement potential	UInt16
Estep (0 - 65535)	Step potential	UInt16
Epulse(0-65535)	Pulse potential	UInt16
Estby (0-65535)	Standby potential	UInt16
nPoints (1-65535)	Number of points	UInt16
tlnt (see description)	Interval time	UInt32
tPulse (in units of 15.2 us)	Pulse time – tmeas (see note)	UInt16
d1(0-255)	Used for ADC's	UInt8 (default: 11)
d16 (0-255)	Used for ADC's	UInt8 (default: 14)
options (0-65535)	Misc. boolean settings. Currently supported: 0x4: Cell on after measurement 0x8: Stirrer present	UInt16
nadmean (0-11)	Used for ADC's	UInt8 (default: 6)

Notes:

The sampling time $t(\text{meas})$ can for instance be set to half the pulse time. So $t(\text{meas}) = t\text{Pulse} / 2$. The value of $t\text{Pulse}$ is corrected for $t(\text{meas})$.

The interval time is equal to potential step / scan rate.

The value of $t\text{Pulse}$ as sent to EmStat is set to $(\text{Pulse time} - t\text{meas})/0.0000152$, where Pulse time is specified by the user and $t\text{meas}$ is the actual sampling time as determined by the values of $nadmean$, $d1$ and $d16$ (see algorithm `void ADTimingEmstat(ref float tmeas)`).

So for instance with Pulse time is 0.07 s and $t\text{meas}$ is 0.02 s (one cycle time of 50 Hz), the value of $t\text{Pulse}$ is $(0.07 - 0.02)/0.000015\text{s} = 3289$.

6.3 Square Wave Voltammetry measurement parameters

The following parameters are supported:

Name	Description	Type
technique=2		
Econd (0-65535)	Conditioning potential	UInt16
tCond (0-65535)	Conditioning Time in units of s	UInt16
Edep (0-65535)	Deposition potential	UInt16
tDep (0-65535)	Deposition time in units of s	UInt16
tEquil (0-65536)	Equilibration time in units of s	UInt16
cr_min (0-5 or 0-7)	Lowest current range	UInt8
cr_max (0-5 or 0-7)	Highest current range	UInt8
cr (0-5 or 0-7)	Starting current range	UInt8
Ebegin (0-65535)	Measurement potential	UInt16
Estep(0-65535)	Step potential	UInt16
Epulse(0-65535)	Pulse potential	UInt16
Estby (0-65535)	Standby potential	UInt16
nPoints (1-65535)	Number of points	UInt16
tlnt (see description)	Interval time or 1 / Freq	UInt32
tPulse (in units of 15.2 us)	Pulse time – tmeas (see note)	UInt16
d1(0-255)	Used for ADC's	UInt8 (default: 11)
d16 (0-255)	Used for ADC's	UInt8 (default: 14)
options (0-65535)	Misc. boolean settings. Currently supported: 0x4: Cell on after measurement 0x8: Stirrer present 0x400: Measure lforwardReverse	UInt16
nadmean (0-11)	Used for ADC's	UInt8 (default: 6)

Notes:

The sampling time $t(\text{meas})$ can for instance be set to one third the pulse time. So $t(\text{meas}) = 1 / (6 * \text{Frequency})$.

The interval time is equal to $1 / \text{Frequency}$ and the pulse time is equal to $1 / (2 * \text{Frequency})$.

The value of $t\text{Pulse}$ as sent to EmStat is set to $(1/(2 * \text{Frequency}) - t\text{meas}) / 0000152$. The value of $t\text{meas}$ is the actual sampling time as determined by the values of $nadmean$, $d1$ and $d16$ (see algorithm `void ADTimingEmstat(ref float tmeas)`).

So for instance with a frequency of 20 Hz the pulse time is 0.025 s and $t\text{meas}$ can be 0.071 s, the value of $t\text{Pulse}$ is $(0.025 - 0.071) / 0.0000152 \text{ s} = 1177$

6.4 Normal Pulse Voltammetry measurement parameters

The following parameters are supported:

Name	Description	Type
technique=3		
Econd (0-65535)	Conditioning potential	UInt16
tCond (0-65535)	Conditioning Time in units of s	UInt16
Edep (0-65535)	Deposition potential	UInt16
tDep (0-65535)	Deposition time in units of s	UInt16
tEquil (0-65536)	Equilibration time in units of s	UInt16
cr_min (0-5 or 0-7))	Lowest current range	UInt8
cr_max (0-5 or 0-7))	Highest current range	UInt8
cr (0-5 or 0-7))	Starting current range	UInt8
Ebegin (0-65535)	Begin potential	UInt16
Estep (0 - 65535)	Step potential	UInt16
Estby (0-65535)	Standby potential	UInt16
nPoints (1-65535)	Number of points	UInt16
tlnt (see description)	Interval time	UInt32
tPulse (in units of 15.2 us)	Pulse time – tmeas (see note)	UInt16
d1(0-255)	Used for ADC's	UInt8 (default: 11)
d16 (0-255)	Used for ADC's	UInt8 (default: 14)
options (0-65535)	Misc. boolean settings. Currently supported: 0x4: Cell on after measurement 0x8: Stirrer present	UInt16
nadmean (0-11)	Used for ADC's	UInt8 (default: 6)

Notes:

The sampling time $t(\text{meas})$ can for instance be set to half the pulse time. So $t(\text{meas}) = \text{Pulse time} / 3$. The value of $t\text{Pulse}$ is corrected for the actual sampling time $t\text{meas}$.

So for instance with Pulse time is 0.07 s and $t\text{meas}$ is 0.02 s (one cycle time of 50 Hz), the value of $t\text{Pulse}$ is $(0.07 - 0.02)/0.000015\text{s} = 3289$.

The interval time is equal to potential step / scan rate.

6.5 Cyclic Voltammetry measurement parameters (preliminary)

The following parameters are supported:

Name	Description	Type
technique=5		
Econd (0-65535)	Conditioning potential	UInt16
tCond (0-65535)	Conditioning Time in units of s	UInt16
Edep (0-65535)	Deposition potential	UInt16
tDep (0-65535)	Deposition time in units of s	UInt16
tEquil (0-65536)	Equilibration time in units of s	UInt16
cr_min (0-5 or 0-7))	Lowest current range	UInt8
cr_max (0-5 or 0-7))	Highest current range	UInt8
cr (0-5 or 0-7))	Starting current range	UInt8
Ebegin (0-65535)	The potential where the measurements starts and ends	UInt16
Evtx1(0 - 65535)	The lowest vertex potential	UInt16
Evtx2(0 - 65535)	The highest vertex potential	UInt16
Estep(0 - 65535)	Step potential. Its sign determines the direction of the first segment of the scan.	UInt16
Estby (0-65535)	Standby potential	UInt16
nScans (1- 255)	Number of scans	UInt8
tlnt (see description)	Interval time	UInt32
d1(0-255)	Used for ADC's	UInt8 (default: 11)
d16 (0-255)	Used for ADC's	UInt8 (default: 14)
options (0-65535)	Misc. boolean settings. Currently supported: 0x4: Cell on after measurement 0x8: Stirrer present 0x40: Measure Aux. Input	UInt16
nadmean (0-11)	Used for ADC's	UInt8 (default: 6)

Notes:

1. The sampling time $t(\text{meas})$ can be set to half the interval time or less.
So for instance $t(\text{meas}) = \text{tlnt} / 2$. The interval time is equal to potential step / scan rate.
2. A CV scan consists of either 2 or 3 segments.
3. The scan starts at Ebegin and steps are made with a value Estep.

6.6 Measurement sequence for LSV, NPV, DPV and SWV

1. If $t\text{Cond} > 0$ then Econd is applied is for tCond seconds.
2. If $t\text{Dep} > 0$ then Edep is applied is for tDep seconds.
3. Ebegin is the potential at which the measurements are done. Before the actual measurements start, the program waits tEquil seconds.
4. The highest current range is applied. If the current is below $0.05 * \text{current range}$, a lower current range is set. If the current exceeds $1.6 * \text{current range}$ a higher current range is set. However the current range will never be set to a higher value than cr_max and never lower than cr_min.
5. After the equilibration time, the measurement is performed, by measuring nPoints.
6. After completing the measurement, EmStat sends the character (*) as a kind of end-of-file sign.
7. If the 'Cell on after measurement bit is set, the applied potential is set to Estby. If this bit is not set, the cell is switched off.

6.7 Amperometric measurement parameters

The following parameters are supported:

Name	Description	Type
technique=7		
Econd (0-65535)	Conditioning potential	UInt16
tCond (0-65535)	Conditioning Time in units of s	UInt16
Edep (0-65535)	Deposition potential	UInt16
tDep (0-65535)	Deposition time in units of s	UInt16
tEquil (0-65536)	Equilibration time in units of s	UInt16
cr_min (0-5 or 0-7))	Lowest current range	UInt8
cr_max (0-5 or 0-7))	Highest current range	UInt8
cr (0-5 or 0-7))	Starting current range	UInt8
Ebegin (0-65535)	Measurement potential	UInt16
Estby (0-65535)	Standby potential	UInt16
nPoints (1-65535)	Number of points.	UInt16
tlnt (see description)	Interval time.	UInt32
mux_delay (0-65536)	Delay in 15.2 us per unit after selecting a new channel of the multiplexer.	UInt16 (default: 320)
nmux (1 – 16)	Number of channels to be measured.	UInt8 (default 16)
d1(0-255)	Used for ADC's	UInt8 (default: 11)
d16 (0-255)	Used for ADC's	UInt8 (default: 14)
options (0-65535)	Misc. boolean settings. Currently supported: 0x2: Alternating Multiplexer 0x4: Cell on after measurement 0x8: Stirrer present 0x40: Measure Aux. Input	UInt16
nadmean (0-11)	Used for ADC's	UInt8 (default: 6)

Note:

The sampling time $t(\text{meas})$ can be set to half the interval time or less. So for instance $t(\text{meas}) = \text{tlnt} / 2$.

6.8 Pulsed amperometric measurement parameters

The following parameters are supported:

Name	Description	Type
technique=8		
Econd (0-65535)	Conditioning potential	UInt16
tCond (0-65535)	Conditioning Time in units of s	UInt16
Edep (0-65535)	Deposition potential	UInt16
tDep (0-65535)	Deposition time in units of s	UInt16
tEquil (0-65536)	Equilibration time in units of s	UInt16
cr_min (0-5 or 0-7))	Lowest current range	UInt8
cr_max (0-5 or 0-7))	Highest current range	UInt8
cr (0-5 or 0-7))	Starting current range	UInt8
Ebegin (0-65535)	Measurement potential	UInt16
nPoints (1-65535)	Number of points.	UInt16
Epulse(0-65535)	Pulse potential	UInt16
Estby (0-65535)	Standby potential	UInt16
tPulse (in units of 15.2 us)	Pulse time – tmeas (see note)	UInt16
tlnt (see description)	Interval time.	UInt32
mux_delay (0-65536)	Delay in 15.2 us per unit after selecting a new channel.	UInt16 (default: 320)
d1(0-255)	Used for ADC's	UInt8 (default: 11)
d16 (0-255)	Used for ADC's	UInt8 (default: 14)
options (0-65535)	Misc. boolean settings. Currently supported: 0x4: Cell on after measurement 0x8: Stirrer present 0x10: PAD mode 2 0x20: PAD mode 3	UInt16
nadmean (0-11)	Used for ADC's	UInt8 (default: 6)

Note:

The sampling time $t(\text{meas})$ can be set to half the pulse time or less. So for instance $t(\text{meas}) = t\text{Pulse} / 2$.

So for instance with Pulse time is 0.07 s and the actual sampling time t_{meas} is 0.02 s (one cycle time of 50 Hz), the value of $t\text{Pulse}$ is $(0.07 - 0.02) / 0.000015\text{s} = 3289$.

6.9 Open circuit potentiometry (OCP) measurement parameters

The following parameters are supported:

Name	Description	Type
technique=10		
Econd (0-65535)	Conditioning potential	UInt16
tCond (0-65535)	Conditioning Time in units of s	UInt16
Edep (0-65535)	Deposition potential	UInt16
tDep (0-65535)	Deposition time in units of s	UInt16
tEquil (0-65536)	Equilibration time in units of s	UInt16
nPoints (1-65535)	Number of points.	UInt16
tlnt (see description)	Interval time.	UInt32
mux_delay (0-65536)	Delay in 15.2 us per unit after selecting a new channel of the multiplexer.	UInt16 (default: 320)
nmux	Number of channels to be measured	UInt8 (default: 16)
d1(0-255)	Used for ADC's	UInt8 (default: 11)
d16 (0-255)	Used for ADC's	UInt8 (default: 14)
options (0-65535)	Misc. boolean settings. Currently supported: 0x2: Alternating Multiplexer 0x4: Cell on after measurement 0x8: Stirrer present 0x40: Measure Aux. Input	UInt16
nadmean (0-11)	Used for ADC's	UInt8 (default: 6)

Note:

The sampling time $t(\text{meas})$ can be set to half the interval time or less. So for instance $t(\text{meas}) = \text{tlnt} / 2$.

6.10 Multiple pulse amperometric (MPAD) measurement parameters

The following parameters are supported:

Name	Description	Type
technique=11		
Econd (0-65535)	Conditioning potential	UInt16
tCond (0-65535)	Conditioning Time in units of s	UInt16
Edep (0-65535)	Deposition potential	UInt16
tDep (0-65535)	Deposition time in units of s	UInt16
tEquil (0-65536)	Equilibration time in units of s	UInt16
cr_min (0-5 or 0-7))	Lowest current range	UInt8
cr_max (0-5 or 0-7))	Highest current range	UInt8
cr (0-5 or 0-7))	Starting current range	UInt8
E1 (0-65535)	Measurement potential	UInt16
E2 (0-65535)	First potential to regenerate the electrode	UInt16
E3 (0-65535)	Second potential to regenerate the electrode	UInt16
nPoints (1-65535)	Number of points.	UInt16
Estby (0-65535)	Standby potential	UInt16
tlnt (see description)	Interval time.	UInt32
t2	Time of potential E2 in unit of 71.5E-06 s	UInt16
t3	Time of potential E3 in unit of 71.5E-06 s	UInt16
d1(0-255)	Used for ADC's	UInt8 (default: 11)
d16 (0-255)	Used for ADC's	UInt8 (default: 14)
options (0-65535)	Misc. boolean settings. Currently supported: 0x4: Cell on after measurement 0x8: Stirrer present	UInt16
nadmean (0-11)	Used for ADC's	UInt8 (default: 6)

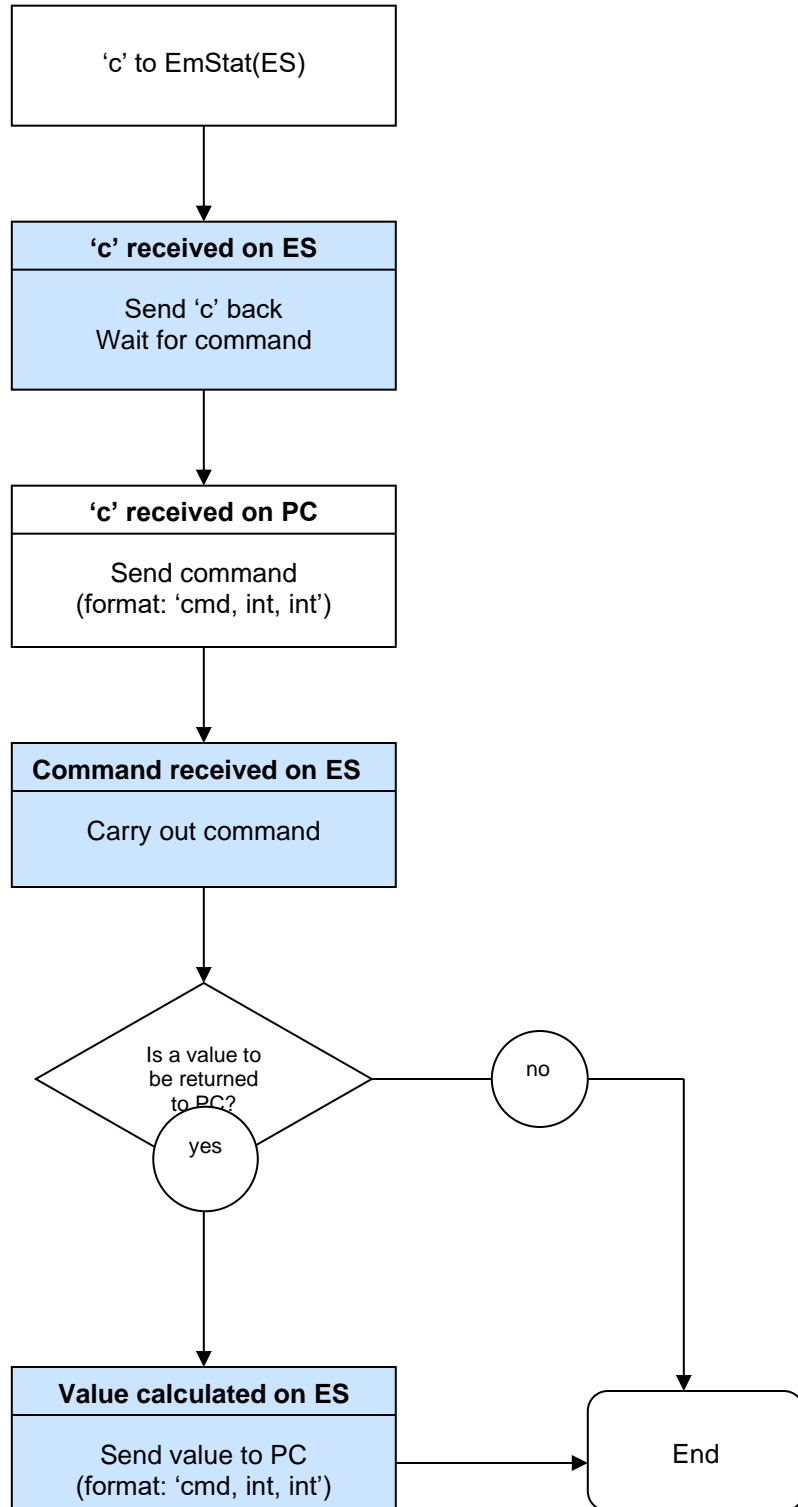
Notes:

The sampling time $t(\text{meas})$ can be set to half the E1 stage. So for instance $t(\text{meas}) = (tlnt - t2 - t3) / 2$.

6.11 Measurement sequence for AD, PAD and MPAD.

1. If $t\text{Cond} > 0$ then Econd is applied is for tCond seconds.
2. If $t\text{Dep} > 0$ then Edep is applied is for tDep seconds.
3. Ebegin is the potential at which the measurements are done. Before the actual measurements start, the program waits tEquil seconds.
4. The highest current range is applied. If the current is below $0.05 * \text{current range}$, a lower current range is set. If the current exceeds $1.6 * \text{current range}$ a higher current range is set. However the current range will never be set to a higher value than cr_max and never lower than cr_min.
5. After the equilibration time, the measurement is performed, by measuring nPoints.
6. The currents are sampled just before Epulse is applied. So the measurement takes place at Ebegin. Epulse is used to regenerate the working electrode (PAD only). If MPAD is applied, the current is sampled at the end of the E1 stage. E2 and E3 are applied to regenerate the electrode.
6. After completing the measurement, EmStat sends the character (*) as a kind of end-of-file sign.
7. If the 'Cell on after measurement bit is set, the applied potential is set to Estby. If this bit is not set, the cell is switched off.

Appendix A: Manual control flow chart



Appendix B: Version changes

Version 7.3

- Added “cr” method parameter for the starting current range, this is a mandatory parameter.
- Corrected U package potential using Efactor instead of DACfactor.
- Option bit for SWV IReverse changed from 0x40 to 0x400 for compatibility reasons

Version 7.4

- Implemented new mux protocol for MUX8-R2

Version 7.5

- No protocol changes

Version 7.6

- No protocol changes