

Programmer's Guide

▶ www.bmcm.de

meM-LOG

Programming of the
Data Logger meM-LOG

Table of Contents

1 Introduction	5
1.1 Copyrights	5
1.2 BMC Messsysteme GmbH	6
2 Programming	7
2.1 General Notes	7
2.2 Overview	7
2.3 Programming Commands	9
2.3.1 Start/stop scanning	9
2.3.2 Read scan status	9
2.3.3 Set scan configuration	10
2.3.4 Read scan configuration	11
2.3.5 Set to fast/normal mode	12
2.3.6 Read fast/normal mode	12
2.3.7 Adjust offset of all channels	13
2.3.8 Set analog alarm limits	13
2.3.9 Read analog alarm limits	14
2.3.10 Set digital alarm levels	14
2.3.11 Read digital alarm levels	15
2.3.12 Set alarm strategy	16
2.3.13 Read alarm strategy	16
2.3.14 Read one analog input channel	17
2.3.15 Read all analog input channels	17
2.3.16 Read digital lines	18
2.3.17 Get number of stored standard records (continuous mode)	19
2.3.18 Get number of stored event records (alarm mode)	19
2.3.19 Read a record	20
2.3.20 Synchronized sampling	21
2.3.21 Read synchronized data	21
2.3.22 Set device configuration	22
2.3.23 Read device configuration	23
2.3.24 Read the firmware version	24
2.3.25 Read the name of the device	24

2.3.26	Read the serial number	24
2.3.27	Get internal times	25
2.3.28	Set internal time	25
3	Index	27

1 Introduction

1.1 Copyrights

The data logger **meM-LOG** has been developed and tested with utmost care. BMC Messsysteme GmbH does not provide any guarantee in respect of this manual and its contents, its quality, its performance or fitness for a particular purpose. BMC Messsysteme GmbH is not liable in any case for direct or indirect damages or consequential damages, which may arise from improper operation or any faults whatsoever of the system. The system is subject to changes and alterations which serve the purpose of technical improvement.

The data logger **meM-LOG**, this manual and all names, brands, pictures, other expressions and symbols are protected by law as well as by national and international contracts. The rights established therefrom, in particular those for translation, re-print, extraction of depictions, broadcasting, photomechanical or similar way of reproduction - no matter if used in part or in whole - are reserved. Reproduction of the programs and the manual as well as passing them on to others is not permitted. Illegal use or other legal impairment will be prosecuted by criminal and civil law and may lead to severe sanctions.

Copyright
© 2002

Last updated:
03/12/2002

BMC Messsysteme GmbH
Hauptstr. 21, 82216 Maisach,
GERMANY

Phone: +49 8141/40 41 80-0
Fax +49 8141/40 41 80-9

E-Mail: info@bmcm.de

1.2 BMC Messsysteme GmbH

BMC Messsysteme GmbH stands for innovative measuring technology made in Germany. We provide all components required for the measuring chain, from sensor to software.

Our hard- and software components are perfectly tuned with each other to produce an extremely user-friendly integrated system. We put great emphasis on observance of current industrial standards, which facilitate the interaction of many components.

Products by BMC Messsysteme are applied in industrial large-scale enterprises, in research and development and in private application. We produce in compliance with ISO-9000-standards, since standards and reliability are of paramount importance to us - for your profit and success.

Please visit us on the Web (<http://www.bmcm.de>) for detailed information and latest news.



▶ w w w . b m c m . d e

2 Programming

2.1 General Notes

Each line sent or received to/from the **meM-LOG** device must end with a *CarriageReturn* (0x0d).

In the following description, 'AA' is always the address of the **meM-LOG** device in hex notation.

If a command was recognized successfully by the device, an exclamation mark '!' is returned followed by the device address. In case of an error, otherwise a question mark '?' is returned together with the device address (in hex notation).

Example:

- '?03' means that device 03 detected a parameter or syntax error.
- '!03' means that the command was transmitted successfully to the device 03.

2.2 Overview

You will find all available commands with a short explanation listed in the following table. For further details please refer to the respective chapter.

Command	Function
@AASN	<i>Start/stop scanning</i>
@AAT	<i>Read scan status</i>
@AACZZZMLSFFFDDXXXX	<i>Set scan configuration</i>
@AAD	<i>Read scan configuration</i>
*AAF _M	<i>Set to fast/normal mode</i>
*AAF?	<i>Read fast/normal mode</i>

\$AA1	<i>Adjust offset of all channels</i>
@AAACSDHHHSDLLLL	<i>Set analog alarm limits</i>
@AABC	<i>Read analog alarm limits</i>
*AASANN	<i>Set digital alarm levels</i>
*AAGA	<i>Read digital alarm levels</i>
*AAAM	<i>Set alarm strategy</i>
*AAA?	<i>Read alarm strategy</i>
#AAC	<i>Read one analog input channel</i>
\$AAA	<i>Read all analog input channels</i>
*AAN	<i>Get number of stored standard records (continuous mode)</i>
@AAN	<i>Get number of stored standard records (continuous mode)</i>
@AAL	<i>Get number of stored event records (alarm mode)</i>
@AARNNNNNNNNXX	<i>Read a record</i>
#**	<i>Synchronized sampling</i>
\$AA4	<i>Read synchronized data</i>
%AANNRRBBFF	<i>Set device configuration</i>
\$AA2	<i>Read device configuration</i>
\$AAF	<i>Read the firmware version</i>
\$AAM	<i>Read the name of the device</i>
*AAS	<i>Read the serial number</i>
*AAGTM	<i>Get internal times</i>
*AASTXTTTTTTTT	<i>Set internal time</i>

2.3 Programming Commands

2.3.1 Start/stop scanning

Syntax:	@AASN
AA:	device address in hex notation
S:	command
N:	'0' = stop scanning; any other character = start scanning
Reply:	!'AA' on success, else '?AA'

Example: '@04S1' → '!04'

The command orders the device with the address 0x04 to start ('1') the scan. The device replies that the scan was started ('!04').

Note: The "start scan" command starts a scan immediately. It overrides any pending scan start and destroys all previous stored data in the logger.

2.3.2 Read scan status

Syntax:	@AAT
AA:	device address in hex notation
T:	command
Reply:	!'AAS' S: 0 = not scanning 1 = scanning 2 = wait for scan start at specific time

Example: '@04T' → '!042'

Asks the device 0x04 for the current scan status. The device replies that it is waiting for a scan start at a previous programmed time ('2').

2.3.3 Set scan configuration

Syntax:	@AACZZZZMLSFFFFDDXXXX
AA:	device address in hex notation
C:	command
ZZZZ:	bitfield of the 16 channels to be scanned: cleared bit = do not store channel, set bit = store channel
M:	standalone mode ('1'=active): This is ignored by meM-LOG as it is always in standalone mode (waiting for an event to start a scan).
L:	logging mode: '0' = continuous mode; '1' = alarm mode
S:	storage type: '0' = stop if RAM is full; '1' = use RAM as ring buffer
FFFF:	sampling interval in hex notation (0000..FFFF), unit in seconds or 1/100 seconds depending on which sampling mode (normal/fast) has been selected (see "Set to fast/normal mode", p. 12)
DD:	bitfield of the digital lines to be scanned
XXXX:	number of seconds in hex notation before next scan shot where the analog part of the meM-LOG device is powered up (0001..sampling frequency); only valid in normal mode
Reply:	!AA on success, else '?AA'

Example: '@04C00011100001000001' → '!04'

The command orders the device with the address 0x04 to scan the first analog channel (ZZZZ = '0001'). The device is set to standalone mode (M = '1') and logs data in alarm mode (L = '1'). The scan automatically finishes at full RAM (S = '0') and a sampling frequency of 1 (FFFF = '0001') is used. No digital channels are stored (DD = '00'). The device is powered up 1sec. before next sampling (XXXX = '0001').

The device replies that the scan parameters are set ('!04').

Note: This command will clear all previously stored data in the meM-LOG-device!

2.3.4 Read scan configuration

Syntax:	@AAD
AA:	device address in hex notation
D:	command
Reply:	<p>'!AAZZZZMLSFFFFDD'</p> <p><i>ZZZZ</i>: bitfield of the 16 channels to be scanned: cleared bit = do not store channel, set bit = store channel</p> <p><i>M</i>: standalone mode (always returns '1')</p> <p><i>L</i>: logging mode: '0' = continuous mode; '1' = alarm mode</p> <p><i>S</i>: storage type: '0' = stop if RAM is full; '1' = use RAM as ring buffer</p> <p><i>FFFF</i>: sampling interval in hex-notation (0000..FFFF), unit in seconds or 1/100 seconds depending on which sampling mode (normal/fast) has been selected (see "Set to fast/normal mode", p. 12)</p> <p><i>DD</i>: bitfield of the digital lines to be scanned</p>

Example: '@04D' → '!040001110000100'

The command asks the device 0x04 for its current scan configuration. The device replies that the first analog channel is scanned (*ZZZZ* = '0001'). It is in standalone mode (*M* = '1') and logs data in alarm mode (*L* = '1'). The scan automatically finishes at full RAM (*S* = '0') and a sampling frequency of 1 (*FFFF* = '0001') is used. No digital channels are stored (*DD* = '00').

2.3.5 Set to fast/normal mode

Syntax:	*AAF M
AA:	device address in hex notation
F:	command
M:	'0' = normal mode; '1' = fast mode
Reply:	'!AA' on success, else '?AA'

Note: The value for the sampling interval given in the command "Set scan configuration", p. 10, is either interpreted in seconds (normal mode) or 1/100 seconds (fast mode). E.g. in normal mode the value 2 means, that sampling is done every 2sec. (0.5Hz), whereas in fast mode the sampling interval is 0.02sec. (50Hz).

2.3.6 Read fast/normal mode

Syntax:	*AAF?
AA:	device address in hex notation
F?:	command
Reply:	'!AAM' M: '0' = normal mode; '1' = fast mode (see "Set to fast/normal mode", p. 12)

2.3.7 Adjust offset of all channels

Syntax:	\$AA1
AA:	device address in hex notation
1:	command
Reply:	'!AA' on success, else '?AA'

Example: '\$041' → '!04'

The command sets the offset of all channels of device 0x04. The value measured at the analog inputs will be added to all further values of the respective channel. The offset values are stored in the meM-LOG's internal EEPROM.

2.3.8 Set analog alarm limits

Syntax:	@AAACSDHHHHSLLLL
AA:	device address in hex notation
A:	command
C:	channel number in hex notation
S:	sign of the value: '+' or '0' = positive sign; any other character is a negative sign
D:	number of decimals after the point (e.g. 0.000: D=3; 0.0000: D=4)
HHHH:	value in hex format of the upper alarm limit
LLLL:	value in hex format of the lower alarm limit
Reply:	'!AA' on success, else '?AA'

Example: '@04AE+308CA-30550' → '!04'

The command sets the alarm limits of channel 14 (C = 'E') of device 0x04 to +2.250V (upper limit HHHH = '08CA') and -1.360V (lower

limit LLLL = '0550'). Both values have 3 decimals after the decimal point (D = '3'), the upper limit is a positive value (S = '+'), the lower one a negative (S = '-'). The device replies that the limits are set ('!04').

2.3.9 Read analog alarm limits

Syntax:	@AABC
AA:	device address in hex notation
B:	command
C:	channel number in hex notation
Reply:	'!AASDHHHSDLLLL' on success, else '?AA' S: sign of the value: '+' = positive sign; '-' = negative sign D: number of decimals after the point (e.g. 0.00: D=2; 0.000: D=3) HHHH: value in hex format of the upper alarm limit LLLL: value in hex format of the lower alarm limit

Example: '@04B9' → '!049+3127A-30230'

The command asks the device 0x04 for the alarm limits of channel 9 (C = '9'). The device returns +4.730V (S = '+'; D = '3'; HHHH = '127A') as upper value and -0.560V (S = '-'; D = '3'; LLLL = '0230') as lower limit.

2.3.10 Set digital alarm levels

Syntax:	*AASANN
AA:	device address in hex notation
SA:	command
NN:	bitmask with alarm levels of the digital lines (in hex notation)
Reply:	'!AA'

Example: '*04SA66' -> '!04'

This command sets the alarm levels of the digital lines 2, 3, 6, 7 to active high and of the digital lines 1, 4, 5, 8 to active low (NN = '66' = 01100110 (bin)). The device replies that the settings were done.

Note: As meM-LOG has only two digital inputs only changes of the last 2 bits have an effect on the digital alarm levels of the device.

2.3.11 Read digital alarm levels

Syntax:	*AAGA
AA:	device address in hex notation
GA:	command
Reply:	'!AADD' DD: bitmask with alarm levels of the digital lines (in hex notation)

Example: '*04GA' -> '!041C'

This command asks for the digital alarm levels of device 0x04. The device replies that the alarm levels of line 3, 4 and 5 are active high and the rest is active low (DD = '1C' = 00011100 (bin)).

Note: As meM-LOG has only two digital inputs only the last 2 bits give information about the digital alarm levels of the device.

2.3.12 Set alarm strategy

Syntax:	*AAAM
AA:	device address in hex notation
A:	command
M:	'0' = save only enabled channels which exceeded their alarm limits; '1' = save all enabled channels if the alarm limits of one enabled channel were exceeded
Reply:	'!AA' on success, else '?AA'

Example: '*04A1' → '!04'

This command sets the alarm strategy of device 0x04 to save all enabled channels in case of alarm (M = '1'). The device replies that the strategy was set.

2.3.13 Read alarm strategy

Syntax:	*AAA?
AA:	device address in hex notation
A?:	command
Reply:	'!AAM' M: '0' = save only enabled channels which exceeded the their alarm limits; '1' = save all enabled channels if the alarm limits of one enabled channel were exceeded

Example: '*04A?' -> '!040'

This command asks for the alarm strategy of device 0x04. The device replies that the current alarm strategy is to save only channels, which exceeded their alarm limits (M = '0').

2.3.14 Read one analog input channel

Syntax:	#AAC
AA:	device address in hex notation
C:	channel number in hex notation
Reply:	'>value' on success, else '?AA' value: depends on the selected data format: engineering, percent or hexadecimal

Example: engineering format: '#04D' → '>3.650'
 % format: '#04D' → '>35'
 hex format: '#04D' → '>0E42'

The command asks the device 0x04 for the analog value of channel 13 (C = 'D'). The device returns a value of 3.650 (engineering) or 35 (percent) or 0E42 (hexadecimal).

Note: For valid data formats see "Set device configuration", p. 22.

2.3.15 Read all analog input channels

Syntax:	\$AAA
AA:	device address in hex notation
A:	command
Reply:	'>value>value ... >value>value' value: depends on the selected data format: engineering, percent or hexadecimal

Example: '\$04A' →
 '>1.234>0.023>4.125>2.850>1.234>0.023>4.125>2.850...>2.850'

Depending on the data format in the device configuration command, the returned values are either in engineering format, hexadecimal or in

percent (here: engineering). In this examples the device 0x04 returns the values of the 16 analog inputs (listed above: first 8 and last channel)

Note: For valid data formats see "Set device configuration", p. 22.

2.3.16 Read digital lines

Syntax:	*AAD?
AA:	device address in hex notation
D?:	command
Reply:	'>XXYY' XX: digital-in lines in hex notation YY: digital-out lines in hex notation

Example: '*04D?' → '>0300'

The command asks the device 0x04 for the current values of the digital lines. The device replies that the two low in-lines are set ('03') and that no out-line is set ('00').

Note: The data logger meM-LOG provides for two digital input lines only.

2.3.17 Get number of stored standard records (continuous mode)

Syntax:	@AAN
AA:	device address in hex notation
N:	command
Reply:	'!AANNNNNNNN' NNNNNNNN: number of available standard records in meM-LOG (hex notation)

Example: '@04N' → '!040000000A'

The command asks for the number of stored standard records. The device replies that there are 10 ('0000000A') records available.

Note: In case of an error the returned number of records is also '00000000'.

2.3.18 Get number of stored event records (alarm mode)

Syntax:	@AAL
AA:	device address in hex notation
L:	command
Reply:	'!AANNNNNNNN' NNNNNNNN: number of available event records in meM-LOG (hex notation)

Example: '@04L' → '!0400000008'

The command asks the device 0x04 for the number of stored event records. The device replies that there are 8 ('00000008') records available.

Note: In case of an error the returned number of records is also '00000000'.

2.3.19 Read a record

Syntax:	@AARNNNNNNNNXX
AA:	device address in hex notation
R:	command
NNNNNNNN:	number of records to read in hex notation (00000000..FFFFFFFF)
XX:	number of records in hex notation to read as a block (01..FF) Attention: If using the USB interface do not exceed the following limits to avoid internal buffer overflow (in decimals): continuous mode: max = 28, alarm mode: max = 14
Reply:	!AACSVVVVTTTTTTTTDDCSVVVVTTTTTTTTDD ... CSVVVVTTTTTTTTDD' on success, else '?AA' C: channel number (hex notation) S: bitfield with sign and decimal point of the value MSB → $\{ \overset{\cdot}{\underset{\cdot}{X}} \overset{\cdot}{\underset{\cdot}{X}} \overset{\cdot}{\underset{\cdot}{X}} \overset{\cdot}{\underset{\cdot}{X}} \overset{\cdot}{\underset{\cdot}{D}} \overset{\cdot}{\underset{\cdot}{D}} \overset{\cdot}{\underset{\cdot}{D}} \overset{\cdot}{\underset{\cdot}{S}} \}$ ← LSB with: S: sign (0 = positive; 1 = negative) D: number of dec. points (always 3) X: undefined VVVV: sample value (in hex notation) TTTTTTTT: timestamp (in hex notation) in relation to scan start (only written if alarm mode was selected) DD: digital lines in hex notation (only sent if digital lines were enabled for scanning)

Example: '@04R0000001D01' → '!042705460000000D'

The command asks the device 0x04 for the stored record 29 (NNNNNNNN = '000001D'), which should be returned as one record (X = '01'). In this example it is channel 2 and -1.350V.

The device returns the channel number (C = '2'), a negative sign and the decimal point at 3 (B = '7'), the value (VVVV = '0546') and a timestamp of 13 (TTTTTTTT = '0000000D'). This tell us that the alarm mode was selected (because of the existance of a timestamp) and that it was sampled 13 ticks after scan start. A tick is either 1sec. (normal mode) or 1/100sec. (fast mode) (see "Set to fast/normal mode", p. 12). No digital channels were scanned.

Note: If sending records at once (as defined in the 'XX'-part) do not exceed the total number of available records, otherwise the additional returned records are undefined.

If the device is scanning in loop mode (ring buffer), reading records affects the number of available records reported by the device. So, if reading a block of records, those records are **cleared** in the device and the number of available records decreased respectively. The next block reading must start again at record number '00000000'!

2.3.20 Synchronized sampling

Syntax:	#**
Reply:	no reply from the devices

Note: The value of channel 1 is stored.

2.3.21 Read synchronized data

Syntax:	\$AA4
AA:	device address in hex notation
4:	command
Reply:	'!AAS>value' S: '1' = value was not read yet after the '#**' command (see "Synchronized sampling", p. 21); '0' = value was read before value: depends on selected data format: engineering, percent or hex

Example: '\$044' → '!041>-0.007'

The command reads the data of the last synchronized sampling command of device 0x04. The device replies that the data is read for the first time (S = '1') and returns a value of -0.007. The device uses the engineering data format.

Note: For valid data formats see "Set device configuration", p. 22.

2.3.22 Set device configuration

Syntax:	%AANNRRBBFF
AA:	current device address in hex notation
NN:	new device address in hex notation (00..FF)
RR:	input range of analog inputs in hex notation (meM-LOG accepts only '05')
BB:	index of the baudrate (*)
FF:	index of the data format for transmitting analog values (**)
Reply:	'!AA' on success, else '?AA'

* baudrate index: '04' = 2400bps;
 '05' = 4800bps;
 '06' = 9600bps;
 '07' = 19200bps;
 '08' = 38400bps (or any other character)

** data format index: '00' = engineering
 '01' = percent
 any other character = hexadecimal

Example: '%0403050800' → '!04'

The command sets the device 0x04 to the new address 0x03 (NN = '03'). The input-range is ±5V (RR = '05'), the baudrate is 38400bps (BB = '08') and the data format is set to engineering (FF = '00').

The device replies that all settings are done correctly.

Note: The baudrate and the device address are changed immediately after the reply '!AA'.

Data formats:

- Engineering: [-5.120..5.120]; -5.12V = -5.120, 5.12V = 5.120; number of decimals after the point: 3
- Percent: [-50..50]; -5.12V = -50%, 0V = 0%, 5.12V = 50%
- Hexadecimal: [0000..FFFF]; -5.12V = 0000, 0V = 8000, 5.12V = FFFF

2.3.23 Read device configuration

Syntax:	\$AA2
AA:	device address in hex notation
2:	command
Reply:	<p>'!AARRBBFF'</p> <p>RR: input range of analog inputs in hex notation (meM-LOG accepts only '05')</p> <p>BB: index of the baudrate</p> <p>FF: index of the data format for transmitting analog values (**)</p>

Example: '\$042' → '!04050702'

The command asks the device 0x04 for its current settings. The device replies that the input-range is $\pm 5V$ (RR = '05'), the baudrate is 19200bps (BB = '07') and that the data-format is hexadecimal (FF = '02').

Note: For valid indexes of the baudrate and the data format see "Set device configuration", p. 22.

2.3.24 Read the firmware version

Syntax:	\$AAF
AA:	device address in hex notation
F:	command
Reply:	'!AA<version> version: string containing the firmware version (e.g. '1.0.11')

2.3.25 Read the name of the device

Syntax:	\$AAM
AA:	device address in hex notation
M:	command
Reply:	'!AA<name> name: string containing the module name ('meM-LOG')

2.3.26 Read the serial number

Syntax:	*AAS
AA:	device address in hex notation
S:	command
Reply:	'!AANNNNNNNN' NNNNNNNN: serial number in hex notation

Example: '*04S' → '!04FEEDCODE'

The command asks the device 0x04 for its serial number. The device returns the serial number 4.276.994.270 (NNNNNNNNN = 'FEEDCODE')

2.3.27 Get internal times

Syntax:	*AAGTM
AA:	device address in hex notation
GT:	command
M:	'0' = current time of the device; '1' = time of the last scan start; '2' = time to start scan (pending scan start, see "Set internal time", p. 25)
Reply:	'!AATTTTTTTTT' on success, else '?AA' TTTTTTTT: time in UTC format (time in sec. since 01.01.1970) and hex notation

Example: `*04GT0' → '!043BC2DC7D'`

The command asks the device 0x04 for the actual time of the device (M = '0'). The device returns the current date (d/m/y) 09.10.2001 at (h/min/sec) 13:16:12 (TTTTTTTT = '3BC2DC7D').

2.3.28 Set internal time

Syntax:	*AASTTTTTTTTT
AA:	device address in hex notation
ST:	command
X:	'0' = current time of the device; '2' = time to start a scan
TTTTTTTT:	time in UTC format (time in sec. since 01.01.1970) and hex notation
Reply:	'!AA' on success, else '?AA'

Example: '*04ST03BC2DC17' → '!04'

The command sets the internal time (X = '0') of device 0x04 to (d/m/y) 09.10.2001 at (h/min/sec) 13:14:30 (TTTTTTTT = '3BC2DC17'). The device replies that the actual time was set.

Note: If X = 2, a value of "00000000" disables the pending scan start. Otherwise a scan will be started immediately if the value is below or equal to the logger's internal time.

If the value is above the loggers internal time, it is waiting for the programmed time of the scan start. This will only be successful if no values are stored in the logger's RAM from a previous scan. So make sure to send a scan configuration command before setting the pending scanstart time (see "Set scan configuration", p. 10).

In case of a value different from "00000000" do not send an extra "start scan" command (see "Start/stop scanning", p. 9).

3 Index

A

- alarm
 - store channels 16
 - strategy 16
- alarm levels 14, 15
- alarm limits 13, 14
- alarm mode 10, 11, 13, 14, 15, 16, 19, 20
- analog channel
 - read all values 17
 - read value 17
 - set offset 13

B

- baudrate 22, 23
- block reading 21

C

- channel
 - store 10, 11
 - store alarm channel 16
- command 7
- continuous mode 10, 11, 19, 20

D

- data format 17, 22, 23
 - engineering 23
 - hexadecimal 23
 - percent 23
- device
 - firmware version 24
 - internal time 25
 - name 24
 - serial number 24
- device address 7, 22
- device configuration 22, 23

- digital lines
 - read 18

E

- error 7, 19, 20

F

- fast mode 12, 21
- firmware version 24

I

- input range 22, 23
- internal time 25
- internet address 6

L

- loop mode 10, 11, 21

N

- name 24
- normal mode 12, 21
- number of event records 19
- number of standard records 19

O

- offset 13

P

- power-up device 10

R

- read record 20
- read synchronized data 21
- record
 - number 19

read 20
read in blocks 21

S

sampling interval 10, 11, 12
scan
 pending 9, 25
 read configuration 11
 set configuration 10
 start 9, 25
 status 9
 stop 9
serial number 24
synchronized sampling 21

T

time 25
timestamp 20