A. REMOTE CONTROL

The USB 2.0 interface is the serial one working with 480 MHz clock which enables one to control remotely the unit. Its speed is relatively high and it ensures the common usage of USB in all produced nowadays Personal Computers.

The functions, which are developed in order to control data flow in the serial interfaces, ensure:

- Bi-directional data transmission,
- Remote control of the instrument.

The user, in order to programme the serial interface, has to:

1. send "the function code",
2. send an appropriate data file

or

3. receive a data file.

A.1. Input / Output transmission types

The following basic input / output transmission types (called functions) are available:

#1 input/output of the control setting codes,

#2 read-out of the measurement results in the **VLM** mode,

#**3** read-out of the measurement results in the **1/1 OCTAVE** or **1/3 OCTAVE** analysis mode

#4 read-out of the data file from the internal flash-disc or RAM memory,

#7 special control functions,

#9 writing the data file into the internal flash-disk.

#D read/write the data file from the external memory (SD Card),

A.2. Function #1 - Input/Output of the control setting codes

#1 function enables the user to send the control setting codes to the instrument and read out a file containing the current control state. A list of the control setting codes is given in Tab. A.1. The format of #1 function is defined as follows:

#1,Xccc,Xccc,(...),Xccc;

or

#1,Xccc,X?,Xccc,(...),X?,Xccc;

where:

X - the group code, **ccc** - the code value,
X? - the request to send the current X code setting.

The instrument outputs in this case a control settings file for all requests X? in the following format:

#1,Xccc,Xccc,(...),Xccc;

In order to read out all current control settings the user should send to the device the following sequence of characters:

#1;

The instrument outputs in this case a file containing all control settings given in Tab. A1 in the format:

#1,Xccc,Xccc,(...),Xccc;

Example: The instrument sends the following sequence of characters as an answer for the mentioned above request:

#1,U103,N1234,W1.06.1,Q0.01:1,Q0.03:2,Q0.05:3,Q0.40:4,q140.00,M4,G9,g65,d1s,D10s,K5,Y3,y0,S0,T1,e480,m0,s4,l120,k1,p0,n10,Xa1,Xf250,Xb500,XV2,XT0,XQ4,XL120,Xg0,Xj1,Xk120,Xp0,Xq0,XG0,XJ2,XK120,XB0,Xc10,XC4,XD0;

means that:

- the SV 103 is investigated (U103);
- its number is 1234 (N1234);
- the software version number 1.06.1 (W1.06.1);
- the calibration factor is equal to 0.01 dB (Q0.01:1) in channel X, calibration factor is equal to 0.03 dB (Q0.03:2) in channel Y and calibration factor is equal to 0.05 dB (Q0.05:3) in channel Z, calibration factor is equal to 0.40 dB (Q0.40:4) in force channel;
- the calibration level is equal to 140.00 dB (q140.00);
- the **DOSE METER** mode is selected (M4);
- the **PEAK** and **RMS** values are stored in the files of the logger from all channel (G9);
- the summary results **MAIN** and **FORCE** are stored in the files of the logger from all channel (g65);
- the results are stored in a logger's file every 1 second (d1s);
- the integration period is equal to 10 seconds (D10s);
- the measurement has to be repeated 5 times (K5);
- the delay of the start of the measurements is equal to 3 seconds (Y3);
- the synchronization the start of measurement with RTC is switched off (y0);
- the instrument is in the Stop state (S0);
- the logger is active (T1);
- the exposition time is set to 8 hours (e480);
- the time-domain signal recording is switched off (m0);
- the RMS value from channel Z is treated as a source for trigger in the time-domain signal recording (s4);
- the time-domain trigger level is equal to 120 dB (l120);
- the signal from channel X will be recorded in time-domain signal recording (k1);
- the additional recording time before the triggering in time-domain signal recording is switched off (p0);
- the recording time in time-domain signal recording is equal to 10 seconds (n10).
- the reference level is equal 1 $\mu m/s^2$ (Xa1)
- the **E**xposure **A**ction **V**alue is equal to 2.50 m/s^2 (Xf250:1);
- the **E**xposure **L**imit **V**alue is equal to 5.00 m/s^2 (Xb500:1);
- the **ELV** alarm source is selected (XV2);
- the measure triggering is switched off (XT0);
- the RMS value from channel Z is treated as a source of the measure triggering signal (XQ4);
- the measure trigger level is equal to 120 dB (XL120);
- the logger triggering is switched off (Xg0);
- the RMS value from channel Z is treated as a source of the logger triggering signal (Xj1);
- the logger trigger level is equal to 120 dB (Xk120);
- the number of the records before the triggering saved in a file of the logger is equal to 0 (Xp0);
- the number of records registered, after the moment in which the measured signal does not fulfil any longer the condition of the triggering, is equal to 0 (Xq0);
- the wave trigger signal recording is switched off (XG0);
- the RMS value from channel Y is treated as a source of the wave triggering signal (XJ2);
- the wave trigger level is equal to 120 dB (XK120);
- the additional recording time before the triggering in wave trigger is switched off (XB0);
- the recording time in wave trigger mode is equal to 10 seconds (Xc10).

- the time-domain signal from channel Z will be recorded in wave trigger (XC4);
- the PCM wave file format is selected (XD0);

Note: *All bytes of that transmission are ASCII characters.*

A.3. Function #2 - measurement results read-out in the VLM mode

#2 function enables one to read-out the current measurement result from the selected channel in the **VLM** mode.

Notice: *This function can also be programmed while measurements are taking place. In this case, the RMS values measured **after entering #2 function** are sent out.*

#2 function has the format defined as follows:

#2,p,X?,X?,X?,(...),X?;

where:
X - the code of the result,
p - the number of channel:
1 - channel **X,** profile 1 (Wh filter),
2 - channel **Y,** profile 1 (Wh filter),
3 - channel **Z,** profile 1 (Wh filter).
4 - channel **X,** profile 2 (Band Limiting Wh filter),
5 - channel **Y,** profile 2 (Band Limiting Wh filter),
6 - channel **Z,** profile 2 (Band Limiting Wh filter).

Notice: *After finishing the measurement, **#2** function is no longer active and has to be reprogrammed in order to read-out successive measurements.*

The instrument sends the values of results in the format defined as follows:

#2,p,Xccc,Xccc,Xccc,(...),Xccc; (where **p** - the number of channel)

or

#2,?; (when the results are not available).

The codes of the results from the **DOSE METER** mode are defined as follows:

v the under-range flag (ccc equals to 0 when the overload did not occur, 1 when the under-range took place during the last measurement period);
V the overload flag (ccc equals to 0 or 1);
T time of the measurement (ccc – value in seconds);
P the **PEAK** value (ccc – the value in dB);
Q the **P_P** value (ccc – the value in dB);
M the **MAX** value (ccc – the value in dB);
R the **RMS** value (ccc – the value in dB);
O the **AEQ** result (ccc – the value in dB);
c the **CExp** result (ccc – the value in dB);
o the **CExp** result (ccc – the value in points);

f the **A(8)** result (ccc – the value in dB);
p the **A(8)** result (ccc – the value in points);
g the **EAVTT** result (ccc – the value in s);
h the **EAVTL** result (ccc – the value in s);
i the **ELVTT** result (ccc – the value in s);
j the **ELVTL** result (ccc – the value in s);
l the **FUT** result (ccc – the value in s);

The exemplary results of the instrument's response after sending to it the following sequence of characters: **#2,1;** coming from the channel **X** are given below:

#2,1,v0,V0,T1,P126.20,Q132.22,M123.19,R123.19,O127.96,c83.37,o0,f127.96,p100,g28807,h28806, i115212,j115211,m41.56,n40.65,k40.65,l0;

Notice: *The presented above order of the measurement results sent out by the instrument does not depend about the characters sent to the unit.*

Example: After sending to the instrument the string:

#2,1,T?,R?,V?,P?;

the unit sends out the results of measurement coming from the channel X in predefined, described above, order:

#2,1,V0,T1,P126.20,R123.19;

Notice: *The value displayed on the screen during the result's presentation is sent out from the instrument in the case when **nn** is not given after **X** character.*

Notice: *All bytes of that transmission are ASCII characters.*

A.4. Function #3 - Read-out of the measurement results in 1/1 octave or 1/3 octave modes

#3 function enables one to read out the current measurement results in **1/1 OCTAVE** or **1/3 OCTAVE** mode.

#3 function format is defined as follows:

#3; The device responds, sending the last averaged spectrum.
#3,A; The device responds, sending the last averaged spectrum
#3,I; The device responds, sending the last instantaneous spectrum
#3,M; The device responds, sending the last max spectrum
#3,N; The device responds, sending the last min spectrum

The device responds, sending the last measured spectrum (when the instrument is in STOP state) or currently measured spectrum (when the instrument is in RUN state) in the following format:

#3;<Status Byte> <LSB of the transmission counter> <MSB of the transmission counter> <X channel data byte> (...) <X channel data byte> <Y channel data byte> (...) <Y channel data byte> <Z channel data byte> (...) <Z channel data byte>

Status Byte gives the information about the current state of the instrument.

D7	D6	D5	D4	D3	D2	D1	D0

where:

D7 = 0 means that "overload does not happen" in Z channel,
= 1 means that “overload appeared” in Z channel,
D6 = 0 means that "overload does not happen" in Y channel,
= 1 means that “overload appeared” in Y channel,
D5 = 0 means that "overload does not happen" in X channel,
= 1 means that “overload appeared” in X channel,
D4 = 0 the instantaneous current result (RUN State),
= 1 the final result (STOP State),
D3 = 1 the **1/3 OCTAVE** results,
D2 = 1 the **1/1 OCTAVE** results,
D1,D0 – type of spectrum:
00 means that averaged spectrum,
01 means that instantaneous spectrum,
10 means that max spectrum,
11 means that min spectrum,

Note: *The measurement result is coded in binary form as dB•100 (e.g. 34.52 dB is sent as binary number 3452).*

A.5. Function #4 - read-out of the data file from the internal flash-disc

#4 function enables the user to read-out the data file from the internal Flash-disc memory. The data file formats are given in Appendix B.

#4 function formats are defined as follows:

#4,0,\; the file containing the catalogue,
#4,0,?; the count of the files,
#4,0,index,count; the part of the file containing the catalogue,
where:
index - first record,
count - number of records in the catalogue.

#4,1,fname; the file containing the measurement results,
#4,1,fname,?; file size,
#4,1,fname,offset,length; the part of the file containing the measurement results,
where:
fname - name containing not more than eight characters,
offset - offset from the beginning of the file,
length - number of bytes to read,

#4,4; the Settings file,
#4,4,?; size of Settings file,
#4,4,offset,length; the part of Settings file,
where:
offset - offset from the beginning of the Settings file,
length - number of bytes to read,

Notice: *The "\" character is treated as the file name of the catalogue and must be sent to the instrument.*

All data words are sent as **<LSB>**,**<MSB>**.

When an error is detected in the file specification or data, the instrument will send:

#4,?;

The catalogue of the files is a set of the records containing 16 words (16 bits each). Each record describes one file saved in the instrument's Flash-disc. The record structure is as follows:

words 0 - 3	8 characters of the file name,
word 4	file type (binary number),
word 5	reserved,
word 6	the least significant word of the file size,
word 7	the most significant word of the file size,
words 8 - 15	reserved.

A.6. Function #D - Read / Write the data files from the external memory (SD Card)

<disk>	logical disk number: 0 – SD Card, 1 – USB Disk (not implemented), 2 – Internal Memory (not implemented)
<address>	directory address (cluster numer) – for internal memory 0
<offsetB>	offset the first byte to read (an even number).
<nB>	number of bytes to read (an even number)
<data>	binary data.
<count>	directory size in bytes
<name>	file name in format XXXXXXXX.YYY (XXXXXXXX – file name, YYY- file name extension)
<dirName>	directory name
<nBwr>	number of bytes to write

1) #D,c,?; this function returns the list of available disks in format:

   #D,c,<disk1>[,<disk2>[,<disk3>]];

2) #D,d,?; this function returns the parameters of the working directory in format:

   #D,d,<disk>,<address>,<count>;

3) #D,d,<disk>,<address>; this function enables to change the working directory

   Response:

   #D,d; - command was executed

   #D,d,?; - command cannot be executed

4) #D,r,<disk>,<address>,<offsetB>,<nB>; function enables the user to read the file (except of internal memory):

Response:
#D,r,<disk>,<address>,<offsetB>,<nB>; [<data>]

5) #D,w,<name>,<nBwr>;<data> function enables the user to write the file to working directory:

Response:
#D,w; - command was executed
#D,w,?; - command cannot be executed

6) #D,e,<name>; function enables the user to delete the file in working directory:

Response:
#D,e; - command was executed
#D,e,?; - command cannot be executed

7) #D,e; function enables the user to delete all files in in working directory:

Response:
#D,e; - command was executed
#D,e,?; - command cannot be executed

8) #D,m,<address>,<dirName>; function enables the user to create a subdirectory in the directory defined by <address>:

Response:
#D,m; - command was executed
#D,m,?; - command cannot be executed

9) #D,f,<address>; function enables the user to delete directory and its contents (files and subdirectories):

Response:
#D,f; - command was executed
#D,f,?; - command cannot be executed

A.7. Function #7 - special control functions

#7 function enables the user to perform special control functions. **Some of them should be used with the extreme care.**

#7 function formats are defined as follows:

#7,AF;

Get Alarm Flags.
Response format:
#7,AF,xx;
where **xx** defined as a sum of the following flags:
1 – EAV.

2 – ELV.

#7,AS;

Get settings for the Auto-Run function.
Response format:
#7,AS,e,HH,MM,hh,mm,dW,mR;
where:
e – On (e=1), Off (e=0),
HH – hour of the measurement start,
MM – minutes of the measurement start,
hh – hour of the measurement stop,
mm – minutes of the measurement stop,
dW – day of week in which the measurement will be done:
bit:0 – Monday,
...
bit:6 – Sunday
mR – maximum number of the measurement days,

#7,AS, e,HH,MM,hh,mm,dW,mR;

where:
e – On (e=1), Off (e=0),
HH – hour of the measurement start,
MM – minutes of the measurement start,
hh – hour of the measurement stop,
mm – minutes of the measurement stop,
dW – day of week in which the measurement will be done:
bit:0 – Monday,
...
bit:6 – Sunday
mR – maximum number of the measurement days,
Response format:
#7,AS;

#7,BC;

Reserved.

#7,BD;

Reserved.

#7,BN;

Get number of logger files created to the current time.
Response format:
#7,BN,xxx;
where:
xxx – number of logger files.

#7,BS;

Get battery state.
Response format:
#7,BS,x;
where:
x – battery state in [%].

#7,BV;

Get battery voltage.
Response format:
#7,BV,xxx;
where:
xxx – battery voltage in [10 mV].

#7,CP;

Get selected **Standard**.
Response format:
#7,CP,xx;
where:
xx = **GE** – German,
xx = **UK** – English,
xx = **IT** – Italian,
xx = **PL** – Polish,
xx = **FR** – French,
xx = **UD** – User defined,

#7,CP,xx;

Set **Standard** for.
where:
xx = **GE** – German,
xx = **UK** – English,
xx = **IT** – Italian,
xx = **PL** – Polish,
xx = **FR** – French,
xx = **UD** – User defined,
Response format:
#7,CP;

#7,CS;

Clear setup (restore factory settings).
Response format:
#7,CS;

#7,DS,name;

Delete setup file in SETUP directory.
where:
name – setup file name,
Response format:
#7,DS;

#7,DU;

Get display units.
Response format:
#7,DU,x;
where:
x = 0 – logarithmic,
x = 1 – linear metric,
x = 2 – linear non-metric.

#7,DU,x;

Set display units.

where:

x = 0 – logarithmic,
x = 1 – linear metric,
x = 2 – linear non-metric.

Response format:

#7,DU;

#7,ED;

Delete all files and directories on SD card. The function is not accepted while the instrument is in the RUN state.

Response format:

#7,ED;

#7,EV;

Get external supply voltage.

Response format:

#7,EV,xxx;

where:

xxx – external supply voltage in [10 mV].

#7,FS;

Get file system version.

Response format:

#7,FS,x.xx;

where:

x.xx – file system version number.

#7,FT;

Get SD-card fat type.

Response format:

#7,FT,x;

where:

x = -1 – SD-card is not available,
x = 1 – FAT16,
x = 2 – FAT32,
x = 3 – FAT12.

#7,IC;

Reserved.

#7,KL;

Get the states of keyboard lock.

Response format:

#7,KL,x;

where:

x = 0 – keyboard lock off,
x = 1 – keyboard lock on.

#7,KL,x;

Get the states of keyboard lock.

where:

x = 0 – keyboard lock off,
x = 1 – keyboard lock on.

Response format:

#7,KL;

#7,LA;

Get interface language.
Response format:
#7,LA,name;
where:
xx = **GE** – German,
xx = **EN** – English,
xx = **IT** – Italian,
xx = **PL** – Polish,
xx = **HU** – Hungarian,
xx = **TU** – Turkish,
xx = **RU** – Russian,
xx = **NL** – Flemish,
xx = **FR** – French,
xx = **SP** – Spanish,

#7,LB;

Get the name of the last logger filename.
Response format:
#7,LB,name;
where:
name – last logger filename.

#7,LS,name;

Load setup and writes settings into EEPROM. The selected file must exist.
where:
name – setup file name.
Response format:
#7,LS;

#7,LW;

Get name of last created wave file**;**
Response format:
#7,LW,name;
where:
name – wave file name.

#7,MC;

Get acceleration sensor compensation.
Response format:
#7,MC,x;
where:
x = 0 – Off,
x = 1 – On,

#7,MC,x;

Set acceleration sensor compensation.
where:
x = 0 – Off,
x = 1 – On,
Response format:
#7,MC;

#7,NF;

Get SD-card number of free sectors (sector = 512 bytes).
Response format:
#7,NF,x;
where:
x – number of free sectors (in case of **x = -1** SD-card is not available).

#7,NS;

Get SD-card number of sectors (sector = 512 bytes).
Response format:
#7,NS,x;
where:
x – number of sectors (in case of **x = -1** SD-card is not available).

#7,PC;

Get Post Calibration.
Response format:
#7,PC,x;
where:
x = 0 – off,
x = 1 – last file,
x = 2 – files after last calibration.

#7,PC,x;

Set Post Calibration.
where:
x = 0 – off,
x = 1 – last file,
x = 2 – files after last calibration.
Response format:
#7,PC;

#7,PI;

Get internal microcontroller firmware version.
Response format:
#7,PI,x.xx;
where:
x.xx – internal microcontroller firmware version number.

#7,PO;

Power off the instrument.
Response format:
#7,PO;

#7,RT;

Get current real time clock settings.
Response format:
#7,RT,hh,mm,ss,DD,MM,YYYY;
where:
hh:mm:ss – time,
DD/MM/YYYY – date.

#7,RT,hh,mm,ss,DD,MM,YYYY;

Set current real time clock and date settings;
where:
hh:mm:ss – time,
DD/MM/YYYY – date.
Response format:
#7,RT;

#7,SC;

Reserved.

#7,SD;

Get date and time of last loaded setup file;
Response format:
#7,RT,hh,mm,ss,DD,MM,YYYY;
where:
hh:mm:ss – time,
DD/MM/YYYY – date.

#7,SE;

Reserved.

#7,SF;

Reserved.

#7,SN;

Get last loaded setup file name;
Response format:
#7,SN,name;
where:
name – setup file name.

#7,SP;

Reserved.

#7,SS,name;

Create setup file based on the current settings and save on SD card.
where:
name – name of the setup file.
Response format:
#7,SS;

#7,ST;

Get Standby Delay.
Response format:
#7,ST,xxx;
where:
xxx – time to standby in [s].

#7,ST,x;

Set Standby Delay.
where:
xxx – time to standby in [s].
Response format:

#7,ST;

#7,TP;

Reserved.

#7,UF;

Get usb speed;
Response format:
#7,UF,x;
where:
x = 0 – High speed (480Mbps).
x = 1 – Full speed (12Mbps).

#7, UF,x;

Set usb speed;
where:
x = 0 – High speed (480Mbps).
x = 1 – Full speed (12Mbps).
Response format:
#7,UF;

#7,UN;

Get unit name;
Response format:
#7,UN,name;
where:
name – unit name.

#7,UN,name;

Sets the unit name;
where:
name – unit name.
Response format:
#7,UN;

#7,US;

Get unit subtype.
Response format:
#7,US,x;
where:
x – subtype number.

#7,UV;

Get USB voltage.
Response format:
#7,UV,xxx;
where:
xxx – USB voltage in [10 mV].

#7,VB;

Get bootstrap program version.
Response format:
#7,VB,x.xx;
where:
x.xx – bootstrap program version number.

#7,VH;

Get hardboot program version.
Response format:
#7,VH,x.xx;
where:
x.xx – hardboot program version number.

For the unknown function and/or in the case of the other error, all these functions return the following sequence of characters: **#7,?;**

A.8. Function #9 - write-in the data file into the internal flash-disc

#9 function enables the user to write-in the data file into the internal Flash-disc memory. The data file formats are given in Appendix B.

#9 function formats are defined as follows:

#9,FILE_TYPE,FILE_LENGTH,DATA
where:

FILE_TYPE	type of the file 1 - result file, 2 - setup file,
FILE_LENGTH	length of the file in bytes,
DATA	binary content of the file.

A.9. Control setting codes

The control setting codes used in the SV 103 instrument (the internal software revision 1.06.1) are given in the table below.

Table A.1. Control setting codes

Group name	Group code	Code description
Unit type	**U**	`U103` (read only)
Serial number	**N**	`Nxxxx` (read only)
Software version	**W**	`Wx.xx.x` x.xx.x - revision number (read only)

Group name	Group code	Code description
Calibration factor	**Q**	`Qn.nn:c` `n.nn` - real number with the value of the calibration factor ∈ (-1.2 ÷ 3.0) for X, Y, Z channel and ∈ (-19.0 ÷ 19.0) for Force channel `c` - the number of channel: 1:X, 2:Y, 3:Z, 4:Force
Calibration level	**q**	`Qn.nn` n.nn - real number with the value of the calibration level ∈ (115.0 ÷ 145.0)
Measurement function	**M**	`M2` - **1/1 OCTAVE** analyser `M3` - **1/3 OCTAVE** analyser `M4` - **DOSE METER**
Logger type	**G**	`Gx` - x - sum of the following flags flags: `1` - logger with **PEAK** values `2` - logger with **P–P** values `4` - logger with **MAX** values `8` - logger with **RMS** values `16` - logger with **Vector** values `32` - logger with **Spectrum** values `64` - logger with **Force** values
Summary results	**g**	`gx` - x - sum of the following flags flags: `1` - **Main Results** values `2` - **Spectrum** values `4` - **Spectrum MAX** values `8` - **Spectrum MIN** values `16` - **Force** values
Logger step	**d**	`dn` n = number in milliseconds ∈ (**100**, **200**, **500**, **1000**) `dns` n = number in seconds ∈ (1 ÷ 60) `dnm` n = number in minutes ∈ (1 ÷ 60)
Integration period	**D**	`D0` - infinity (measurement finished by pressing the **<STOP>** push-button or remotely by sending S0 control code) `Dnns` nn number in seconds `Dnnm` nn number in minutes `Dnnh` nn number in hours
Repetition of the measurement cycles (RepCycle)	**K**	`K0` - infinity (measurement finished by pressing the **Stop** or remotely - by sending S0 control code) `Knnnn`- nnnn number of repetitions ∈ (1 ÷ 1000)
Exposure Time	**e**	`ennn` - nnn time in minutes ∈ (1 ÷ 480)
Logger	**T**	`T0` - switched **Off** `T1` - switched **On**
Delay in the start of measurement	**Y**	`Ynn` - nn delay given in seconds ∈ (0 ÷ 60)

Group name	Group code	Code description
Synchronization the start of measurement with RTC	**y**	`y0` - switched off (**OFF**) `y1` - synchronization to 1 min. `y15` - synchronization to 15 min. `y30` - synchronization to 30 min. `y60` - synchronization to 1 hour.
State of the instrument (Stop, Start or Pause)	**S**	`S0` - **STOP** `S1` - **START** `S2` - **PAUSE**
Time-domain signal recording mode	**m**	`m0` - switched off (**OFF**) `m1` - recording all measurement `m2` - recording on trigger **SLOPE +** `m3` - recording on trigger **SLOPE –** `m4` - recording on trigger **LEVEL +** `m5` - recording on trigger **LEVEL –**
Time-domain signal recording: stored channel	**k**	`kx` - x - sum of the following flags: `1` - channel X `2` - channel Y `4` - channel Z
Time-domain signal recording: source of the triggering signal	**s**	`sx` - x - sum of the following flags: `1` - the RMS in channel X `2` - the RMS in channel Y `4` - the RMS in channel Z
Time-domain signal recording: triggering level	**l**	`lnnn` - nnn level in dB $\in$ (80 ÷ 160)
Time-domain signal recording: pre-trigger time	**p**	`p0` - switched **Off** `p1` - switched **On**
Time-domain signal recording: recording time	**n**	`nkkk` - kkk time in second $\in$ (1 ÷ 1800) `n0` - recording to the end of measurement
Reference Level	**Xa**	`Xax` - x = reference level $\in$ (1 ÷ 100) in µm/s^2
Exposure **A**ction **V**alue for **Standard** set to User Defined	**Xf**	`Xfnnn` nnn **E**xposure **A**ction **V**alue given in 0.01 m/s^2
Exposure **L**imit **V**alue for **Standard** set to User Defined	**Xb**	`Xbnnn` nnn **E**xposure **L**imit **V**alue given in 0.01 m/s^2
Alarm Mask	**XV**	`XVx` - x - activated alarm defined as a sum of the following flags: `1` - **EAV** `2` - **ELV**
Measure Triggering mode	**XT**	`XT0` - switched off (**OFF**) `XT2` - **SLOPE +** `XT3` - **SLOPE –** `XT4` - **LEVEL +** `XT5` - **LEVEL –**
Source of the measure triggering signal	**XQ**	`XQx` - x - sum of the following flags: `1` - the RMS in channel X `2` - the RMS in channel Y `4` - the RMS in channel Z
Measure Triggering level	**XL**	`XLnnn`- nnn level in dB $\in$ (80 ÷ 160)

Group name	Group code	Code description
Logger Triggering mode	**Xg**	`Xg0` - switched off (**OFF**) `Xg4` - **LEVEL +** `Xg5` - **LEVEL –**
Source of the logger triggering signal	**Xj**	`Xjx` - x - sum of the following flags: `1` - the RMS in channel X `2` - the RMS in channel Y `4` - the RMS in channel Z
Logger Triggering level	**Xk**	`Xknnn`- nnn level in dB ∈ (80 ÷ 160)
Logger Pre-Triggering	**Xp**	`Xpx` - x = number of records saved in logger before the fulfilment of the triggering condition; x ∈ (0 ÷ 8)
Logger Post-Triggering	**Xq**	`Xpx` - x = number of records saved in logger after the fulfilment of the triggering condition; x ∈ (0 ÷ 200)
Wave signal recording mode	**XG**	`XG0` - switched off (**OFF**) `XG1` - recording all measurement `XG2` - recording on trigger **SLOPE +** `XG3` - recording on trigger **SLOPE –** `XG4` - recording on trigger **LEVEL +** `XG5` - recording on trigger **LEVEL –**
Wave signal recording: stored channel	**XC**	`XCx` - x - sum of the following flags: `1` - channel X `2` - channel Y `4` - channel Z
Wave signal recording: source of the triggering signal	**XJ**	`XJx` - x - sum of the following flags: `1` - the RMS in channel X `2` - the RMS in channel Y `4` - the RMS in channel Z
Wave signal recording: triggering level	**XK**	`XKnnn`- nnn level in dB ∈ (80 ÷ 160)
Wave signal recording: pre-trigger time	**XB**	`XB0` - switched **Off** `XB1` - switched **On**
Wave signal recording: recording time	**Xc**	`Xckkk`- kkk time in second ∈ (1 ÷ 1800) `Xc0` - recording to the end of measurement
Wave signal recording: File format	**XD**	`XD0` - PCM wave file format `XD1` - Extensible wave file format