

**Data are stored in memory by blocks with this format**

4 bytes copy of actual header

4 bytes pointer to free space – only for internal use

256 data bytes – time stamps + measured values

**This blocks is possible get by this procedure****Command 1S**

Clear whole memory

**Command 2S**

It is not implemented in new instrument.

**Command 3S**

It send blocks with variable length. Before using this command it is necessary use 4S a 5S for memory analyse. Command 3S is necessary call as many time as lines have answer from 5S.

Sended block should be as:

```
>HHHHHHHH(CR)      ... header
>HHHHHHHH(CR)      ... time
>HHHHHHHH(CR)      ... one channel
>HHHHHHHH(CR)      ... time
>HHHHHHHH(CR)      ... one channel
>HHHHHHHH(CR)      ... one channel
>HHHHHHHH<(CR)     ... checksum
```

or

```
>HHHHHHHH(CR)      ... header
>HHHHHHHH(CR)      ... time
>HHHHHHHH(CR)      ... one channel
>HHHHHHHH(CR)      ... time
>HHHHHHHH(CR)      ... one channel
>HHHHHHHH(CR)      ... one channel
>HHHHHHHH(CR)      ... header
>HHHHHHHH(CR)      ... time
>HHHHHHHH(CR)      ... one channel
>HHHHHHHH(CR)      ... second channel
>HHHHHHHH(CR)      ... time
>HHHHHHHH(CR)      ... one channel
>HHHHHHHH(CR)      ... second channel
>HHHHHHHH(CR)      ... time
>HHHHHHHH(CR)      ... one channel
>HHHHHHHH(CR)      ... second channel
>HHHHHHHH<(CR)     ... checksum
```

**Command 4S**

Return HEX number &gt;HHHH(CR), which respond to number of blocks

**Command 5S**

Return block with number of lines by the value returned 4S.

&gt;HHHH(CR)

&gt;HHHH(CR)

&gt;HHHH(CR)

&gt;HHHH(CR)

Each line contain number of data block lines.

**Header format**

(0xFF)(data length)(other channels OC)(measure channels MC)

(0xFF) - header identifier

(data length) - number of bytes in one record

0 - 3	time in compressed format
4 - 7, 8 - 11, ..., 64 - 67	single float in order MC0, MC1, ... MC7, OC0, OC1, ... OC7

(other channels) - bit oriented

0x80	... mathematic function
0x40	... integrator
0x20	... (channel 14)
0x10	... (channel 13)
0x08	... (channel 12)
0x04	... (channel 11)
0x02	... (channel 10)
0x01	... (channel 9)

(measure channels) - bit oriented

0x80	channel 8, H
0x40	channel 7, G
0x20	channel 6, F
0x10	channel 5, E
0x08	channel 4, D, F
0x04	channel 3, C, P
0x02	channel 2, B, U
0x01	channel 1, A, I

**Time stamp format**

Time is stored as 32-bit number:

$((((YY*13+MO)*32+DD)*24+HH)*60+MI)*60+SS$

where YY is year, MO is month, DD is day, HH is hours, MI is minutes and SS are seconds.

**Measured values format**

Values are stored in IEEE floating point format, named single too.

**Communication example**

#00R4S

>0001

-----  
#00R5S

>0008

-----  
#00R3S

>0DABFACC	DD.MM.YY HH:MM:SS	30.4.6 19:11:8
>C1CBB42E	value	-25,46297836
>0DABFAEA	DD.MM.YY HH:MM:SS	30.4.6 19:11:38
>C1CB963A	value	-25,44835281
>0DABFB08	DD.MM.YY HH:MM:SS	30.4.6 19:12:8
>C1CB9758	value	-25,44889832
>413C4B21<		