Communication with dataloggers

Content:

1. Introduction

This manual contains only basic communication commands and special commands for dataloggers. If you need to use more device depends commands, please contact us on support@meret.sk.

2. Communication protocol

1. Settings of serial link

Any devices communicate at the same transfer speed which is 9600 bauds with no parity and byte has 8 bits. Any transfer starts with one start bit and finishes with one stop bit. Only TXD and RXD pins are used for communication.
Parameters of communication are:

Speed: 9600 bauds (bits per second)
Parity: None
Bits: 8
Stop bit: 1
Start bit: 1

2. Packet structure

Here is a structural diagram of any packet:

Byte	Short	Description
1.	SYNC	Synchronization byte (always is (dec) 85 or (hex) 55)
2.	DADD	Destination device address
3.	SADD	Source device address (computer or master device)
4.	LEN	Length of whole packet (including synchronization byte) max. 255
5.	CMD	Command
6.	…	… command's parameters and data
…		
n.	CHCK	Checksum

Byte:	1.	2.	3.	4.	5.	6. … (n-1).	n.
Description:	SYNC	DADD	SADD	LEN	CMD	parameters	CHCK

3. Device address

Device address is for communicate on line where are more devices than one. So each device must has its own unique address in range from 0 to 254 but 0 is commonly used for master device (computer for example). Any device will respond on address 255 this is called broadcast address. So when master device send request with DADD = 255 it doesn't need to know address of connected device but only one device must be connected on the line in broadcast moment. Master device then receives packet with current device's address. Device take received packet and switch addresses so new DADD will be original SADD.

4. Counting checksum

Checksum is counting as the sum of every (n-1) bytes in packet, this is substracted from 0 and value is finally converted to byte. Take this packet for example:

Byte:	1.	2.	3.	4.	5.	6.	7.	8. (CHCK)
Value (hex):	55	FF	00	08	05	06	00	99

5. Commands

a. Set device address

This command set address of device. The packet is sent with current address in destination address array and new address is sent as Parameter 1. Master device receive acknowledgement with old device address, but any other acknowledgement will be received with new device address.

	Value	Type	Description
Command:	**(hex) 00**	Byte	Write value to device
Parameter 1:	***(dec) 101***	Byte	New device address (101)

Example of packet:

Changing address from 100 to 101

Sent (hex):	55	64	00	07	**00**	***65***	***DB***
Received (hex):	55	00	64	06	**00**	92	

b. Read actual value from device

This command is used to read actual measured value of pressure from device. The actual pressure is selected via Parameter 2 in packet by set it to value of 0 (zero). Device respond with floating point value in standard IEEE format. This float number is in basic unit from manufacturer calibration.

	Value	Type	Description
Command:	**(hex) 05**	Byte	Read value from device
Parameter 1:	***(hex) 10***	Byte	Actual value
Parameter 2:	***(hex) 00***	Byte	Pressure value

Example of packet:

Reading pressure actual value of 100:

Sent (hex):	55	FF	00	08	**05**	***10***	***00***	8F			
Received (hex):	55	00	FF	0B	**05**	***10***	*00*	*00*	*C8*	*42*	82

c. Write mathematical input filtration

This command set mathematical input filtration. The filtration is computed by following formula:
New_value = Old_value + (1-Filtration)*ADC_value
New_value is new actual value which one is transferred into Old_value each counting cycle
Old_value is New_value from previous counting cycle
Filtration is float number which represent level of filtration must be in range (0;1>. For full filtration you may choose 0.99999 or for no filtration you may choose 0.

	Value	Type	Description
Command:	**(hex) 06**	Byte	Write value to device
Parameter 1:	***(hex) 64***	Byte	Special values
Parameter 2:	***(hex) 0D***	Byte	Mathematical filtration
Data 1:	*0.25*	Float	Floating number for filtration in range (0;1>

Example of packet:

Set filtration to 75%:

Sent (hex):	55	FF	00	0C	**06**	***64***	***0D***	*00*	*00*	*80*	*3E*	6B
Received (hex):	55	00	FF	07	**06**	***0D***	92					

6. Special dataloggers commands

a. Read archive memory size

This command is used to read memory size in datalogger. Device respond with floating point value in standard IEEE format. This float number represent size of installed memory.

	Value	Type	Description
Command:	**(hex) 1E**	Byte	Read archive values
Parameter 1:	***(hex) 1C***	Byte	Archive memory size

Example of packet:

Reading memory size capacity of 1 081 344 Bytes:

Sent (hex):	55	FF	00	07	**1E**	***1C***	6A					
Received (hex):	55	00	FF	0B	**1E**	***1C***	*00*	*00*	*84*	*49*	9A	

b. Read type of recorded data in memory

This command is used to read type of recorded data in memory. Type of recorded data is used to determine length of record in bytes. There are two types of recorded data in dataloggers. First one records pressure value and temperature value (code 03 hex), length of record this case is 14 bytes. Second one records only pressure values (code 04 hex), length of record in this case is 10 bytes.

	Value	Type	Description
Command:	**(hex) 1E**	Byte	Read archive values
Parameter 1:	***(hex) 21***	Byte	Archive record type

Example of packet:

Reading first type (code 03 hex) of recorded data:

Sent (hex):	55	FF	00	07	**1E**	***21***	66					
Received (hex):	55	00	FF	09	**1E**	***21***	*00*	*03*	61			

c. Write type of recorded data in memory

If you want to change datalogger from one type to another, it is very important to do this when datalogger has empty memory. If the memory is not empty, all address of new sample will be counted with new record type. This can lead to meshed value in memory. Samples written before change can be read incorrectly.

	Value	Type	Description
Command:	**(hex) 1F**	Byte	Write archive values
Parameter 1:	***(hex) 21***	Byte	Archive record type
Data 1:	*3*	Integer	Two bytes of new type of recorded data in memory

Example of packet:

Writing first type of recorded data:

Sent (hex):	55	FF	00	09	**1F**	***21***	*00*	*03*	60			
Received (hex):	55	00	FF	07	**1F**	***21***	65					

d. Read samples count

This command is used to read number of samples in memory. Device respond with floating point value in standard IEEE format. This float number represents number of samples saved in memory.

	Value	Type	Description
Command:	**(hex) 1E**	Byte	Read archive values
Parameter 1:	***(hex) 22***	Byte	Samples count

Example of packet:

Reading number of 100 samples saved in memory:

Sent (hex):	55	FF	00	07	**1E**	***22***	6A					
Received (hex):	55	00	FF	0B	**1E**	***22***	*00*	*00*	*C8*	*42*	57	

e. Write samples count or clear memory

This command writes number of samples to memory. If you want to erase entire memory you can just write zero as number of samples. Number of samples is represented in standard IEEE format.

	Value	Type	Description
Command:	**(hex) 1F**	Byte	Write archive values
Parameter 1:	***(hex) 22***	Byte	Samples count
Data 1:	*0*	Float	Floating number as number of samples

Example of packet:

Writing first type of recorded data:

Sent (hex):	55	FF	00	0B	**1F**	***21***	*00*	*00*	*00*	*00*	61	
Received (hex):	55	00	FF	07	**1F**	***21***	65					

f. Read bytes from memory

This command reads 140 bytes from set address. Address is sent as data after parameter and it is represented in standard IEEE format. This is the only command how to read whole memory. The first two bytes are type of recorded data in memory, same as in chapter **b.** of section **6**. Next four bytes represent samples count stored in memory in standard IEEE format, same as in chapter **d.** of section **6.** There are stored samples after this 6 bytes. Stored samples are organized into 10 bytes or 14 bytes group, depending on type of recorded data in memory. So the total bytes occupied in memory is 6 + samples_count * 10 (code 04 hex for type of recorded data in memory) or 6 + samples_count * 14 (code 03 hex for type of recorded data in memory).

Detailed view on memory organization is in this table:

Type	***Count***	*Sample 1.*	*Sample 2.*	*Sample 3.*	*...*	*Sample n.*

Legend:
Type (2 bytes) – type of recorded data in memory (code 03 hex for 14 bytes, code 04 hex for 10 bytes sample record length)
Count (4 bytes) – number of stored samples
Sample x (14 or 10 bytes) – stored date, time and values of sample (length is depending on type of recorded data in memory)

Detailed view on memory organization of stored sample for 14 bytes length:

Stored date and time	*Float value of pressure*	*Float value of temperature*

Legend:
Stored date and time (6 bytes) – date and time when was sample taken
Float value of pressure (4 bytes) – recorded value of pressure in standard IEEE format
Float value of temperature (4 bytes) – recorded value of temperature in standard IEEE format

Detailed view on memory organization of stored date and time of each sample:

Seconds	***Hours***	***Minutes***	***Day***	***Month***	***Dow***	***Year***

Note:
This memory organization of stored date and time of each sample is not depending on type of recorded data in memory.

Legend:
Seconds (8 bits) – seconds of stored time
Hours (5 bits) – hours of stored time
Minutes (6 bits) – minutes of stored time
Day (5 bits) – day of stored time
Month (5 bits) – month of stored time
Dow (3 bits) – day of week of stored time
Year (16 bits) – year of stored time

Packet structure:

	Value	Type	Description
Command:	**(hex) 1E**	Byte	Read archive values
Parameter 1:	***(hex) 23***	Byte	Archive bytes
Data 1:	*0*	Float	Floating number for starting address

Example of packet:

Reading first 140 bytes of memory from address 0:

Sent (hex):	55	FF	00	0B	**1E**	***23***	*00*	*00*	*00*	*00*	61	
Received (hex):	55	00	FF	93	**1E**	***23***	*00*	*04*	*00*	*00*	*00*	*00*
	00	*00*	*00*	*00*	*00*	*00*	*00*	*00*	*00*	*00*	*00*	*00*
	...											
	00	*00*	*00*	*00*	*00*	*00*	*00*	*00*	*00*	*00*	*00*	D4

Note: As you can see type of recorded data in memory has code 04 hex and there are no samples stored in memory.

g. Read actual time of clock

This command is used to read actual time and date of internal datalogger clock. Time and date are sent directly in bytes format as shown in example. Day of week is not important for correct sampling.

	Value	Type	Description
Command:	**(hex) 1E**	Byte	Read archive values
Parameter 1:	***(hex) 24***	Byte	Archive clock

Example of packet:

Reading time of 22:36:02 (hh:mm:ss) and date of 06.03.2008 (dd.mm.yyyy) day of week is zero:

Sent (hex):	55	FF	00	07	**1E**	***24***	63								
Received (hex):	55	00	FF	0F	**1E**	***24***	*16*	*24*	*02*	*06*	*03*	*07*	*D8*	*00*	37

h. Write actual time of clock

This command is used to write any time and date as actual time and date of internal datalogger clock. Time and date are sent directly in bytes format as shown in description table. Day of week is not important to correct sampling.

	Value	Type	Description
Command:	**(hex) 1F**	Byte	Write archive values
Parameter 1:	***(hex) 24***	Byte	Archive clock
Data 1:	*17*	Byte	Hours of new time
Data 2:	*0*	Byte	Minutes of new time
Data 3:	*0*	Byte	Seconds of new time
Data 4:	*1*	Byte	Day of new date
Data 5:	*1*	Byte	Month of new date
Data 6:	*2008*	Integer	Year of new date
Data 7:	*0*	Byte	Day of week of new date

Example of packet:

Writing time of 17:00:00 (hh:mm:ss) and date of 01.01.2008 (dd.mm.yyyy) day of week is zero:

Sent (hex):	55	FF	00	0F	**1F**	***24***	*11*	*00*	*00*	*01*	*01*	*07*	*D8*	*00*	69
Received (hex):	55	00	FF	07	**1F**	***24***	62								

i. Read archive interval

This command is used to read interval time to take sample. Interval time is sent in byte format in hours, minutes and seconds.

	Value	Type	Description
Command:	**(hex) 1E**	Byte	Read archive values
Parameter 1:	***(hex) 25***	Byte	Archive interval time

Example of packet:

Reading archive interval time of 00:00:05 (hh:mm:ss):

Sent (hex):	55	FF	00	07	**1E**	***25***	62			
Received (hex):	55	00	FF	0A	**1E**	***25***	*00*	*00*	*05*	5A

j. Write archive interval

This command is used to write any time as interval time to take sample. Interval time is sent in byte format and can be set only for hours, minutes and seconds. There is no possibility to set interval for more than one day.

	Value	Type	Description
Command:	**(hex) 1F**	Byte	Write archive values
Parameter 1:	***(hex) 25***	Byte	Archive interval
Data 1:	*0*	Byte	Hours of new interval
Data 2:	*0*	Byte	Minutes of new time
Data 3:	*10*	Byte	Seconds of new time

Example of packet:

Writing archive interval time of 00:00:10 (hh:mm:ss):

Sent (hex):	55	FF	00	0A	**1F**	***25***	*00*	*00*	*0A*	84
Received (hex):	55	00	FF	07	**1F**	***25***	61			

k. Read archive wake up

This command is used to read wake up time. This time is time when the first sample will be taken.

	Value	Type	Description
Command:	**(hex) 1E**	Byte	Read archive values
Parameter 1:	***(hex) 26***	Byte	Archive wake up time

Example of packet:

Reading archive wake up time of 10:00:00 (hh:mm:ss) on date 10.03. (dd.mm) current year:

Sent (hex):	55	FF	00	07	**1E**	***26***	61					
Received (hex):	55	00	FF	0C	**1E**	***26***	*0A*	*00*	*00*	*0A*	*03*	45

l. Write archive wake up

This command is used to write any time as wake up time to take sample so you can plan when the datalogger has to take the first sample. Wake up time time is sent in byte format and can be set only on hours, minutes, seconds, day and month of the current year.

	Value	Type	Description
Command:	**(hex) 1F**	Byte	Write archive values
Parameter 1:	***(hex) 26***	Byte	Archive wake up time
Data 1:	*10*	Byte	Hours of wake up date
Data 2:	*0*	Byte	Minutes of wake up date
Data 3:	*0*	Byte	Seconds of wake up date
Data 4:	*10*	Byte	Day of wake up date
Data 5:	*3*	Byte	Month of wake up date

Example of packet:

Writing archive wake up time on 10:00:00 (hh:mm:ss) and date on 10.03 (dd.mm):

Sent (hex):	55	FF	00	0C	**1F**	***26***	*0A*	*00*	*00*	*0A*	*03*	44
Received (hex):	55	00	FF	0A	**1F**	***26***	60					