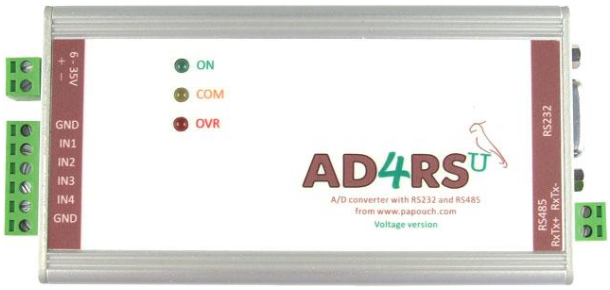
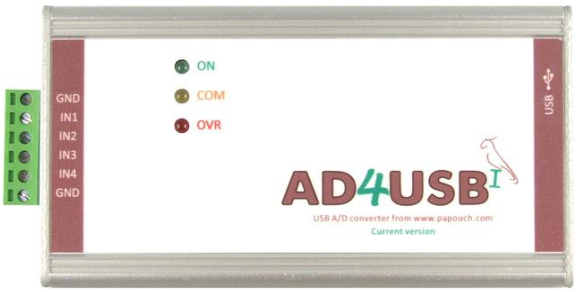
Communication protocol Spinel used in converters AD4xxx

Complete protocol description

+ hysteresis principal addition





Spinel in AD4xxx

Datasheet

Created: 9/7/2007

Last update: 4/2/2008 9:52 AM

Pages: 52

Papouch s.r.o.

Address:

Strašnická 3164/1a
102 00 Praha 10
Czech Republic

Telephone:

+420 267 314 267-8
+420 602 379 954

Fax:

+420 267 314 269

Internet:

www.papouch.com

E-mail:

info@papouch.com

CONTENT

Description....4
Change log....5
Complete description of Spinel protocol....6
Format 97....6
Structure....6
Glossary....6
Format 66....8
Structure....8
Glossary....8
Complete instructions overview....10
Measuring....11
Single measuring....11
Continuous Measuring – start....13
End of measuring....18
Continuous Measuring Setup....19
Reading settings of continuous measuring....19
Communication line and address settings....20
Configuration Permission....20
Communication Parameters Setup....21
Communication Parameters Reading....23
Address Setup using Serial Number....24
Calibration....25
Calibration constants writing....25
Reading of calibration constants....25
Measurement type setting....26
Reading of measurement type settings....26
RAW measurement....27
Additional....28
Name and version reading....28
Manufacturer data reading....29
User Data Saving....30
Saved User Data Reading....31
Input Name Saving....32
Reading of input title....33
Status Setup....34
Reading of status....35
Reading of communication errors....36
Checksum Permission....37
Checksum – settings reading....37
Reset....38
Auxiliary Instructions....39
Single Measurement with Conversion....39
Conversion and Display Setup....40
Conversion and Display Reading....43
Values Monitoring Setup....44
Reading the settings of values monitoring....48
Addition A – Hysteresis....49

DESCRIPTION

This is document describing communication protocol in AD4RS, AD4USB and AD4ETH converters. Complete hardware documentation and functionality description is at disposal on our WEB site http://www.papouch.com/ (detailed documentation also available in PDF).

Basic communication parameters:

AD4RS

Communication line..... RS232 and RS485

Communication speed ..... adjustable 300 Bd to 230,4 kBd (default: 9,6 kBd)

Number of data bits..... 8

Parity..... no parity

Number of stop-bits..... 1

Minimum response delay .....2 ms[1]

AD4USB

Communication line..... USB version 1.1 (USB 2.0 compatible)

Communication speed ..... 115 200 Bd (fixed)

Data bits..... 8

Parity..... none

Stop bits..... 1

AD4ETH

Communication line..... 10/100 Ethernet

Default IP address 192.168.1.254

Default network mask..... 255.255.255.0

Default gateway 0.0.0.0

Default port number for TCP 10001

Virtual port – communication speed.. 115 200 Bd (fixed)

Virtual port – data bits 8

Virtual port – parity..... none

Virtual port – stop bits 1

[1] Due to the delay when switching the communication direction over to RS485.

Change log

Version 04

Added standard current resolutions 0 – 20 mA and 4 – 20 mA. This change is related to instructions 1AH (Measurement type settings), 1BH (Measurement type read), 1EH (Conversion and display settings) and 1FH (Conversion and display read).

- 1AH/1BH: Added value 02H in Measurement type parameter.
- 1EH/1FH: Added parameter Type, identical to parameter in instructions 1AH/1BH.

Corrected description of input parameters in 1BH instruction (Measurement type read). This instruction was already without input parameters in version 2. Always sent all channels' settings in response.

COMPLETE DESCRIPTION OF SPINEL PROTOCOL

Standardized protocol Spinel is implemented to all AD4 modules in formats 66 (ASCII) and 97 (binary). AD4xxx communicates using this protocol in case of TCP/IP or UDP/IP level of communication.

Format 97

Format 97 uses 8bit bytes for communication (0 to 255 in decadal range). For easy communication debugging can be used Spinel Terminal. Instructions are split to Requests and Responses.

Structure

Request:

```
PRE FRM NUM NUM ADR SIG INST DATA… SUMA CR
```

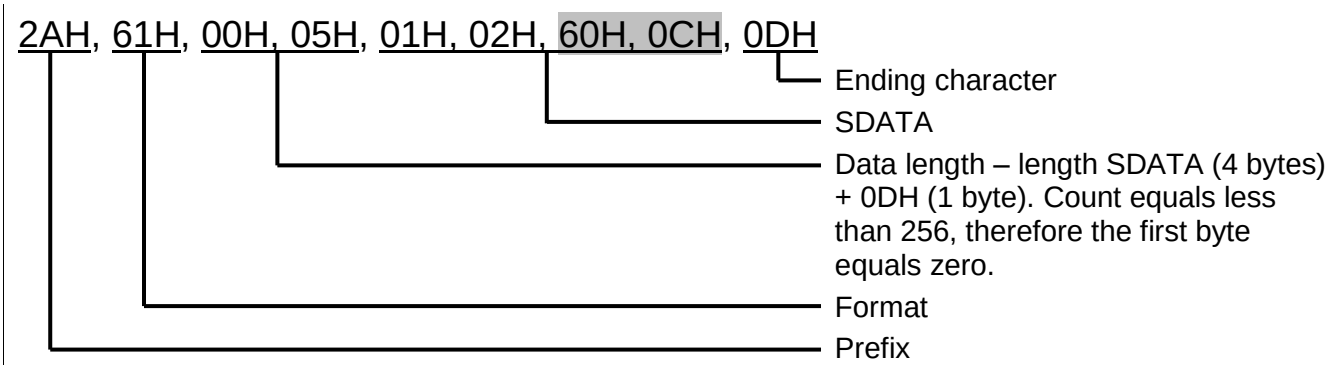
Response:

```
PRE FRM NUM NUM ADR SIG ACK  DATA… SUMA CR
```

PRE	Prefix, 2AH (character "*").
FRM	Format number 97 (61H).
NUM	Byte count of the instruction from following byte to the end of the frame.
ADR	Address of the module sending request or responding to it.
SIG	Instruction signature – any number in range from 00H to FFH. The same number sent in request returns in response to determine which request the response replies on.
INST[2]	Instruction code – Module instructions are described in detail in chapter "Complete instructions overview" on page 10.
ACK	Request confirmation (Acknowledge); if and how was the request performed. ACK lies in interval of 00H to 0FH.
DATA[2]	Data. Described in detail in chapter "Complete instructions overview" (page 10) for each instruction.
SUMA	Checksum.
CR	Ending character (0DH).

Glossary

Example



[2] Data and Instructions are highlighted like this.

Data Length (NUM)

Sixteen-bit value defining the number of bytes until the end of the instruction; number of all bytes found after NUM up to CR (including). It takes the values from 5 to 65535. If lower than 5, the instruction is considered faulty and it is answered (if intended for the relevant device) with ACK "Invalid Data" instruction.

Process of NUM creation:

Ad up the number of bytes following both NUM bytes (i.e. the number of SDATA bytes + 1 CR byte). View the resulting sum as a sixteen-bit number. Divide it into the upper and lower byte. The first NUM byte id the upper byte of the amount, the second NUM byte is the lower byte of the amount. (If the amount of bytes is lower than 256, the first NUM byte is 00H.)

Address (ADR)

The FFH address is reserved for broadcast. If the enquiry contains the FFH address, the device operates as if its own address is entered. No response is sent to enquiries with this address.

The FEH address is a universal address. If the enquiry contains the FEH address, the device operates as if its own address is entered. The device enters real, currently set address into the response. The universal address is used in cases where only one device is connected on the line.

Enquiry Acknowledgement (ACK)

ACK informs the superior device on the way of the received instruction processing. Acknowledgement codes:

00H ..... EVERYTHING OK
The instruction was properly received and completely executed.

01H ..... UNSPECIFIED ERROR
Unspecified device error.

02H ..... INVALID CODE OF INSTRUCTION
The received instruction code is unknown.

03H ..... INVALID DATA
Data are of invalid length or contain an invalid value.

04H ..... ENTRY NOT ALLOWED/ACCESS DENIED
- The enquiry was not performed, as some conditions had not been fulfilled.
- Attempt to enter data into inaccessible memory.
- Attempt to activate a device function requiring a different configuration (e.g. higher communication speed).
- Attempt to change configuration without previous setup permission.
- Access into memory protected by a password.

05H ..... DEVICE FAILURE
- Device failure requiring service action.
- Device internal memory error or setup memory error.
- Device internal error (operation error or start-up error).
- Any other error affecting the device proper functioning.

06H ..... NO DATA AVAILABLE

0DH..... INSTRUCTION SENT AUTOMATICALLY – CHANGE OF DIGITAL INPUT STATE

0EH..... INSTRUCTION SENT AUTOMATICALLY – CONTINUOUS MEASURING
- Periodical sending of measured values.

0FH ..... INSTRUCTION SENT AUTOMATICALLY – LIMITS OR RANGE EXCEEDING

Check Sum (SUMA)

The sum of all instruction bytes (added all transmitted data except for CR) subtracted from 255.

Calculation: SUMA = 255 – (PRE + FRM + NUM + ADR + SIG + ACK (INST) + DATA)

Incorrect checksum is not answered. (Device is waiting for CR even if checksum is incorrect.)

Format 66

Format 66 uses only decimal variables or character that can be typed in using a usual keyboard. That makes this format suitable for synchronizing applications with Spinel. Pauses between individual characters must not be longer than 5 seconds. The instructions are divided into enquiry – response:

Structure

Request:

```
PRE FRM ADR INST DATA… CR
```

Response:

```
PRE FRM ADR ACK  DATA… CR
```

PRE Prefix, 2AH (character "*").

FRM Number of Format 66 (character „B").

ADR Address of the module that receives request or sends response.

INST[2] Instruction code – Instruction codes of a given device. Those are ASCII codes of letters „A" to „Z" and „a" to „z" and numbers „0" to „9". Instructions of the module are described in detail in chapter Complete instructions overview (page 10).

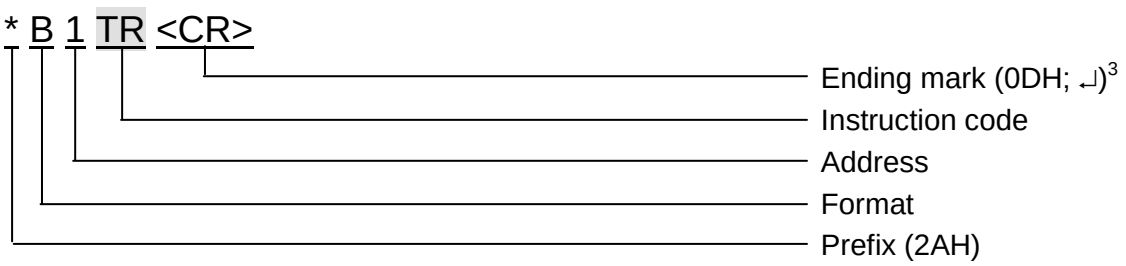
ACK Request acknowledge, if and how was the request performed. ACK lies in interval 00H to 0FH.

DATA[2] Data. Carried instructions in ASCII code. It is recomened to carry the tada in ordinary shape and units. DATA can contain neither prefix nor CR. in detail in chapter Complete instructions overview (page 10).

CR Ending mark (0DH).

Glossary

Example – single measure



Address (ADR)

Address is one character which unambiguously identifies and distinguishes a particular device from others on a single communication line. A device always uses this number for its identification in responses to enquiries from the superior system. The following ASCII characters can form an address: numbers "0" to "9", lower case letters "a" to "z" and capital letters "A" to "Z". The address must not be identical with a prefix or CR.

The "%" address is reserved for broadcast. If the enquiry contains the "%" address, the device operates as if its own address is entered. No response is sent to enquiries with this address.

[3] Ending mark <CR> is not written out in examples from chapter Complete Instructions overview (replaced by character ↵.)

The “$“ address in the universal address. If the enquiry contains the “$“ address, the device operates as if its own address is entered. The device enters real, currently set address into the response. The universal address is used in cases where only one device is connected on the line.

Instruction Code (INST)

Device instruction code.

If a valid instruction is received (correct ADR) and a flag of the received message is set,, the device must respond to such instruction.

Enquiry Acknowledgement (ACK)

ACK informs the master device about the performance of received request.

Acknowledge codes:

0.....EVERYTHING OK
The instruction was properly received and completely executed.

1.....UNSPECIFIED ERROR
Unspecified device error.

2.....INVALID CODE OF INSTRUCTION
The received instruction code is unknown.

3.....INVALID DATA
Data are of invalid length or contain invalid value.

4.....ENTRY NOT ALLOWED/ACCESS DENIED
- The enquiry was not performed, as some conditions had not been fulfilled.
- Attempt to enter data into inaccessible memory.
- Attempt to activate a device function requiring a different configuration (e.g. higher communication speed).
- Attempt to change configuration without previous setup permission.
- Access into memory protected by a password.

5.....DEVICE FAILURE
- Device failure requiring service action.
- Device internal memory error or setup memory error.
- Device internal error (operation error or start-up error).
- Any other error affecting the device proper functioning.

6.....NO DATA AVAILABLE

D .....INSTRUCTION SENT AUTOMATICALLY – CHANGE OF DIGITAL INPUT STATE

E .....INSTRUCTION SENT AUTOMATICALLY – CONTINUOUS MEASURING
- Periodical sending of measured values.

F.....INSTRUCTION SENT AUTOMATICALLY – LIMITS OR RANGE EXCEEDING

Data (DATA)

Instruction data.

COMPLETE INSTRUCTIONS OVERVIEW

Instruction	Code 97	Code 66	Page
Measuring			
Single measuring	51H	MR0	11
Continuous Measuring – start	52H	MC	13
End of measuring	53H		18
Continuous Measuring Setup	54H		19
Reading settings of continuous measuring	55H		19
Communication line and address settings			
Configuration Permission	E4H	E	20
Communication Parameters Setup	E0H	AS a SS	21
Communication Parameters Reading	F0H	CP	23
Address Setup using Serial Number	EBH		24
Calibration			
Calibration constants writing	12H		25
Reading of calibration constants	13H		25
Measurement type setting	1AH		26
Reading of measurement type settings	1BH		26
RAW measurement	5FH		27
Additional			
Name and version reading	F3H	?	28
Manufacturer data reading	FAH		29
User Data Saving	E2H	DW	30
Saved User Data Reading	F2H	DR	31
Input Name Saving	2BH		32
Reading of input title	3BH		33
Status Setup	E1H	SW	34
Reading of status	F1H	SR	35
Reading of communication errors	F4H		36
Checksum Permission	EEH		37
Checksum – settings reading	FEH		37
Reset	E3H	RE	38
Auxiliary Instructions			
Single Measurement with Conversion	58H		39
Conversion and Display Setup	1EH		40
Conversion and Display Reading	1FH		43
Values Monitoring Setup	1CH		44
Reading the settings of values monitoring	1DH		48

To keep the lucidity, Instructions (INST), acknowledgement (ACK) and data (DATA) are described in detail. Address (ADR), Signature (SIG) and CheckSum (SUMA) are described in detail in upper part of the document where the protocol is described.

Measuring

Single measuring

This instruction reads last measured values from all channels of the converter.

Request:

Instruction code: 51H

Parameters: (const)

const	Constant	length: 1 byte
Constant 00H. This byte ensures compatibility with other devices and allows further features installed.		

Response:

Acknowledge code: ACK 00H

Parameters: {(chn_1)($status_1$)($value_1$)} {…} {…} {(chn_4)($status_4$)($value_4$)}

chn	Channel number	length: 1 byte
This byte designates channel number and it holds for all following bytes until next *chn* byte. In essence that means that following bytes (status, value) attach to channel with designated number. From range 01H to 04H depending on channel number.		

status	Status of the measured-out value	length: 1 byte
Status of measured-out value for the channel with the number stated in *chn* byte.		
bit 3,2	00 = value is in measuring range	
	01 = value is below the lower limit (underflow)	
	10 = value is above upper limit (overflow)	
bit 7 (MSb)	0 = invalid value	
	1 = valid value	

value	Measured-out value	length: 2 bytes
Measured value from the channel with number stated in previous byte *chn*. 16-bit number from range 0 – 10 000. Bytes are in order MSB:LSB.		

Examples:

Request:
2AH,61H,00H,06H,31H,02H,51H,00H,EAH,0DH
Request for single measuring.
Response:
2AH,61H,00H,15H,31H,02H,00H,01H,80H,15H,F3H,02H,80H,00H,00H,03H,80H,22H,7BH,04H,88H,28H,2BH,22H,0DH
Meaning of items is described in highlighted part. Highlighted are: Status and then alternately single channels in different hue of grey. 00H – command received OK *Chanel 1 with value 5619:* 01H – byte *chn* with number of channel 1 80H – *status* of measured value for channel 1. Value is valid and in range. 15H,F3H – measured value from channel 1 as a 16-bit number between 0 to 10000 *Chanel 2 with value 0:* 02H – byte *chn* with number of channel 2 80H – *status* of measured value for channel 2. Value is valid and in range. 00H,00H – measured value from channel 2 as a 16-bit number between 0 to 10000 *Chanel 3 with value 8827:* 03H – byte *chn* with number of channel 3 80H – status of measured value for channel 3. Value is valid and in range. 22H,7BH – measured value from channel 3 as a 16-bit number between 0 to 10000 *Chanel 4 with value above the upper limit (value 10283):* 03H – byte *chn* with number of channel 4 88H – status of measured value for channel 4. Value is valid and is above upper limit. 28H,2BH – measured value from channel 3 as a 16-bit number.

In 66 Format:

Request: „MR0“ *(Measure Read)*

Response: (ACK „0“)(ch1)(stat1)(val1)…(ch4)(stat4)(val4)

Legend:

(chX) 2 characters; character break and number of channel; „1“ for first channel, „2“ for second channel, etc.

(statX) 3 character; character break and two characters as a status of measured-out value:

80 … measured-out value is valid

88 … value exceeded the upper limit of measuring range.

(valX) 2 through 6 characters; character break and measured-out value rounded off three digits. Rounding off is adjustable by instruction Conversion and Display Setup on page 40.

Example: *Request*

**B1MR0↵*

Response – channel 1: valid value 806.00; channel 2: valid value 0.00; channel 3: value exceeded the upper limit; channel 3: valid value 1874.50; rounding off set to two digits.

**B10 1 80 809.00 2 80 0.00 3 88 655.47 4 80 1874.50↵*

Continuous Measuring – start

Continuous measuring is a function by means of which AD4 performs measurements in set intervals and sends the measured values automatically via the communication line.

This instruction runs continuous measuring from all channel in the set intervals. Depending on the setup, the measured values are converted to decimal numbers or sent without conversion as numbers ranging from 0 to 10000. The number of measurements can be limited to a certain number or it is possible to continue in measuring until End of measuring instruction has been activated.

It is not recommended to communicate with the module when the automatical sending of values is in operation (with the exception of End of measuring instruction).

Request:

Instruction Code: 52H

Parameters: [interval][sample_counter][flags]

Not all the aforesaid parameters must be present and it is not necessary to adhere to the sequence specified above either. Each parameter shall be preceeded by a parameter code shown in the following description as *id*. The parameters shall thus be entered in the following structure (id_ofparameterA)(parameterA)(id_ofparameterB)(parameterB), etc. The last setting shall be used for parameters that have not been introduced, or default seting shall be used (if they have not been entered yet).

interval Setting the period of continuous measuring	length: 2 bytes id: 01H
Interval sets the period of sending automated response containing measured-out value. Measuring period comes up from this relation: $$\text{period} = \text{interval} \times 406\ \text{ms}$$ Value 0 is not an option. Therefore it is possible to set the period from 406 ms to 7.5 hours. Bytes are in order MSB:LSB. Default value is 406 ms.	

sample_counter Number of samples	length: 2 bytes id: 02H
This parameter sets the number of samples that are to be measured. If the value of this parameter equals zero (0), measuring continues indefinitely until the End of measuring instruction is received. Bytes are in order MSB:LSB. Default value: 0	

flags Other parameters		length: 1 byte id: 03H
Some other parameters are determined in this byte. Default value: 00H		
bit 0 (LSb)	*Non conversion measuring* 0 = values transmitted during continuous measuring are from interval 0 to 10000.	
	Measuring and conversion 1 = values transmitted during continuous measuring are converted according to the actual setting. See page 40 for Conversion and Display Setup.	

Response:

Acknowledge code: ACK 00H

Automated responses:

Automated responses are transmitted according to parameters set above. As the first and the last is transmitted packet containing byte frame_identifier. Between these two packets the measured-out values are transmitted (all channels at once).

Parameters of the first and the last packet: (frame_identifier)

Parameters of the packet of measuring: {(chn_1)($status_1$)($value_1$)} {…} {…} {(chn_4)($status_4$)($value_4$)}

frame_identifier	Status of starting/ending automated response	length: 1 byte
Is byte that is present in the first and the last packet of automated responses. It determines whether the packet is starting or ending. Also reason of the end of measuring can be identified from it.		
bit 0 (LSb)	0 = end of measuring	
	1 = start of measuring	
bit 2	0 = measuring terminated manually using the instruction stop	
	1 = measuring terminated after specified samples are measured-out. (termination by parameter sample_counter)	

chn	Channel number	length: 1 byte
This byte designates channel number and it holds for all following bytes until next *chn* byte. In essence that means that following bytes (status, value) attach to channel with designated number. It ranges from 01H to 04H according to channel number.		

status	Status of the measured-out value	length: 1 byte
Status of measured-out value for the channel with the number stated in *chn* byte.		
bit 3	0 = value is in measuring range	
	1 = value is above upper limit (overflow)	
bit 7 (MSb)	0 = invalid value	
	1 = valid value	

value	Measured-out value	
Measured value from the channel with number stated in previous byte *chn*. Length differs according to settings of continuous measuring in *flags* byte.		
Non conversion measuring		length: 2 bytes
16-bit number from range 0 – 10 000. Bytes are in order MSB:LSB.		
Measuring and conversion		length: 14 bytes
In this case values are transmitted in two formats simultaneously. In 32-bit float format according to the IEEE 754[4] and in ASCII as ten characters of decimal number. Values are in following order: *Example:* Value 9215,85 is formulated as: `46H,0FH,FFH,66H,20H,20H,20H,39H,32H,31H,35H,2EH,38H,35H` IEEE 754 part: `46H,0FH,FFH,66H` ASCII part: `20H,20H,20H,39H,32H,31H,35H,2EH,38H,35H` (9215.85)		

[4] Description of IEEE 754 is available here: http://en.wikipedia.org/wiki/IEEE_754

Examples:

Request:
2AH,61H,00H,05H,31H,02H,52H,EAH,0DH
Here is an example of Measurement with no entered parameters. Only instruction code 52H is stated. For parameters are used last previously entered parameters. In case the parameters have never been entered, default parameters are used.
Response:
2AH,61H,00H,05H,31H,02H,00H,3CH,0DH
Receiving the command for start of measuring has been confirmed.
First automatic response (start of measuring):
2AH,61H,00H,06H,31H,00H,0EH,01H,2EH,0DH
First automatic response contains identification 0EH and byte *frame_identifier* – value 01H means start of measuring.
Last automatic response (end of measuring):
2AH,61H,00H,06H,31H,33H,0EH,04H,F8H,0DH
Last automatic response contains identification 0EH and byte *frame_identifier* – value 04H means end of measuring after previously set number of measurements.
One of automatic responses with measured-out values – no conversion measuring.
2AH,61H,00H,15H,31H,52H,0EH,01H,80H,15H,F3H,02H,80H,00H,00H,03H,80H,22H,7BH,04H,88H,28H,2BH,C4H,0DH 2AH,61H,00H,15H,31H,01H,0EH,01H,80H,15H,F3H,02H,80H,00H,00H,03H,80H,28H,2BH,04H,88H,FFH,FFH,B4H,0DH
The meaning of particular items in highlighted part. Highlighted are status and alternately particular channels in various shade of grey. 0EH – Automatic response identification *Channel 1 with value 5619:* 01H – byte *chn* and channel number 1 80H – *status* of measured-out value for channel 1. Value is valid and in range. 15H,F3H – measured-out value from channel 1 as a 16 bit number from interval 0 to 10000 *Channel 2 with value 0:* 02H – byte *chn* and channel number 2 80H – *status* of measured-out value for channel 2. Value is valid and in range. 00H,00H – measured-out value from channel 1 as a 16 bit number from interval 0 to 10000 *Channel 3 with value 8827:* 03H – byte *chn* and channel number 3 80H – *status* of measured-out value for channel 3. Value is valid and in range. 22H,7BH – measured-out value from channel 1 as a 16 bit number from interval 0 to 10000 *Channel 4 with exceeded limit of range (value 10283):* 03H – byte *chn* and channel number 4 88H – *status* of measured-out value for channel 4. Value is valid but above upper limit of range. 28H,2BH – measured-out value from channel 1 as a 16 bit number from interval 0 to 10000

One of automatic responses with measured-out values – measuring and conversion.
2AH,61H,00H,45H,31H,08H,0EH,01H,80H,40H,96H,A7H,F0H,20H,20H,20H,20H,20H,20H,34H,2EH,37H,31H,02H,80H,C1H,98H,C2H,8CH,20H,20H,20H,2DH,31H,39H,2EH,30H,39H,35H,03H,80H,00H,00H,00H,00H,20H,20H,20H,20H,20H,30H,2EH,30H,30H,30H,04H,80H,00H,00H,00H,00H,20H,20H,20H,20H,20H,30H,2EH,30H,30H,30H,61H,0DH
The meaning of particular items in highlighted part. Highlighted are status and alternately particular channels in various shade of grey. 0EH – Automatic response identification *Channel 1 with value 4.71:* 01H – byte *chn* and channel number 1 80H – *status* of measured-out value for channel 1. Value is valid and in range. 40H,96H,A7H,F0H – measured-out value from channel 1 in float format 20H,20H,20H,20H,20H,20H,34H,2EH,37H,31H – measured-out value from channel 1 in ASCII *Channel 2 with value -19.095:* 02H – byte *chn* and channel number 2 80H – *status* of measured-out value for channel 2. Value is valid and in range. C1H,98H,C2H,8CH – measured-out value from channel 2 in float format 20H,20H,20H,2DH,31H,39H,2EH,30H,39H,35H – measured-out value from channel 1 in ASCII *Channel 3 with value 0.000.* *Channel 4 with value 0.000.*
Note: Description of SIG instruction is automatically incremented in automatic responses.

In format 66:

Request: „MC“(interval)

Response: (ACK „0“)

Automatic Response: (ACK „E“) (value)

Legend: (interval) 1 through 5 characters; Send interval of automatic response with measured-out value. Period of measuring is given by relation: (interval) * 406 [ms]. Therefore it is possible to set time from 406 ms to approximately 7 hours and 23.5 minutes. Default value is 406 ms. If it is set to 0, measuring will terminate.

Example: *Request*

**B1MC1↵*

Response

**B10↵*

Automatic responses – channel 1: valid value 4.71; channel 2: valid value -19.095; channel 3: valid value 0.000; channel 4: valid value 0.000:

**B1E 1 80 4.71 2 80 -19.095 3 80 0.000 4 80 0.000↵*

End of measuring

Continuous measuring is a function by means of which AD4 performs measurements in set intervals and sends the measured-out values automatically via the communication line.

This instruction deactivates the ongoing continuous measuring. This way it is possible to terminate continuous measuring which has been initiated without limiting the number of samples as well as continuous measuring which has not yet measured the required number of samples.

Request:

Instruction code: 53H

Response:

Acknowledge code: ACK 00H

Examples:

Request:
2AH,61H,00H,05H,01H,02H,53H,19H,0DH
End of measuring.
Response:
2AH,61H,00H,05H,01H,02H,00H,6CH,0DH
Command receiving confirmed.

66 format:

The measuring is stopped by command MC0 in 66 format – see format 66, instruction “Continuous Measuring – start”.

Continuous Measuring Setup

Continuous measuring is a function by means of which AD4 performs measurements in set intervals and sends the measured values automatically via the communication line.

This instruction can be used to set the parameters of continuous measuring independently of the start of the measuring. This instruction can be used only when continous measuring is not in operation.

Request:

Instruction code: 54H

Parameters: (interval) (sample_counter) (flags)

(Parameters are the same as for instruction „Continuous Measuring – start“ on the page 13.)

Response:

Acknowledge code: ACK 00H

Examples:

Request:
2AH,61H,00H,0BH,31H,02H,54H,01H,00H,05H,02H,00H,32H,A8H,0DH
Meaning of particular highlighted items: 54H – instruction code 01H – *interval* parameter ID 00H,05H – measuring interval 2.03 seconds (5 x 406 ms) 02H – *sample_counter* parameter ID 00H,32H – parameter *sample_counter* with value 32H, that gives 50 measurings

Reading settings of continuous measuring

This instruction allows us to read the parameters set by previous instruction.

Request:

Instruction code: 55H

Response:

Acknowledge code: ACK 00H

Parameters: (interval) (sample_counter) (flags)

(Parameters are the same as for instruction „Continuous Measuring – start“ on the page 13.)

Examples:

Request:
2AH,61H,00H,05H,31H,02H,55H,E7H,0DH
Command for reading parameters of continuous measuring.
Response:
2AH,61H,00H,0BH,31H,02H,00H,01H,00H,05H,02H,00H,32H,FCH,0DH
Meaning of particular highlighted items: 00H – acknowledge code 01H – *interval* parameter ID 00H,05H – measuring interval 2,03 seconds (5 x 406 ms) 02H – *sample_counter* parameter ID 00H,32H – parameter *sample_counter* with value 32H, that gives 50 measurements

Communication line and address settings

Configuration Permission

This instruction enables configurations to be carried out. It must immediately precede some instructions for communication parameters setup. After a following instruction (even an invalid one) the configuration is automatically disabled again.

This instruction cannot be used with the universal address. There must always be entered an address of a particular device.

Request:

Instruction code: E4H

Response:

Acknowledge code: ACK 00H

Examples:

Request:
2AH,61H,00H,05H,01H,02H,E4H,88H,0DH
Configuration permission.
Response:
2AH,61H,00H,05H,01H,02H,00H,6CH,0DH
Command receiving confirmed.

In 66 format:

Request: „E“ *(Enable)*

Response: (ACK „0“)

Example: *Request*

**B1E↵*

Response

**B10↵*

Communication Parameters Setup

This instruction set the address in Spinel protocol and the communication speed. It is not possible to use the universal address in this instruction.

In case the address is not known and no other device is connected on the line, the address can be found out using the instruction of "Communication Parameters Reading". (Use the universal FEH address as the device address.) If this is not possible (there are other devices on the same communication line), you can assign an address to the required device by using the instruction of "Address Setup using Serial Number " (page 24).

The setup of communication parameters must be preceeded by the instruction for "Configuration Permission" (page 20).

Request:

Instruction code: E0H

Parameters: (address) (speed)

address	New address of the device	length: 1 byte
New address of the device in Spinel protocol. Address ranges from 00H to FDH. If the device uses 66 format, it is necessary to use only addresses from ASCII range (see Address part on page 7.) Default address: 31H		

speed	New communication speed		length: 1 byte
This parameter sets new communication speed of the device. Speed is unchangeable in AD4ETH and AD4USB devices and it is set to 115 200 Bd. Default communication speed in AD4RS is 9 600 Bd. Codes of communication speeds are in table on the right side:	Speed [Bd]	Format 97 code	Format 66 code
	110	00H	0
	300	01H	1
	600	02H	2
	1 200	03H	3
	2 400	04H	4
	4 800	05H	5
	9 600	06H	6
	19 200	07H	7
	38 400	08H	8
	57 600	09H	9
	115 200	0AH	A
	230 400	0BH	B

Response:

Acknowledge code: ACK 00H

New address and communication speed is set after sending the response.

Examples:

Request:
2AH,61H,00H,07H,01H,02H,E0H,02H,0AH,7EH,0D
Address set to 02H and speed to 115200 Bd.
Response:
2AH,61H,00H,05H,01H,02H,00H,6CH,0DH
New address and communication speed is set after sending the response.

In 66 format:

Request: „AS“(address)[5] (Address Set)

Response: (ACK „0“)

Legend: (address) see part Address on page 8.

Example: Request: Address 4

**B1AS4↵*

Response

**B10↵*

Request: „SS“(code)[5] (Speed Set)

Response: (ACK „0“)

Legend: (code) Communication speed code according to the table on the previous page.

Example: Request: Speed 19200Bd (code 7)

**B1SS7↵*

Response

**B10↵*

[5] Address and communication speed has to be set by two separate commands in format 66. (In format 97 this can be done in one instruction.)

Communication Parameters Reading

This instruction is designed for the detection of the set address of a required device in case it is unknown. The enquiry is sent to the FEH universal address. If even the communication speed is not known, it is necessary to try out all communication speeds available for the particular device. However, no other device can be connected on the line when finding out the required device address using the universal address.

Request:

Instruction code: F0H

Response:

Acknowledge code: ACK 00H

Parameters: (address) (speed)

address	Device's address	length: 1 byte
Device's address in Spinel protocol.		

speed	Communication speed		length: 1 byte
Communication speed code. Speed is unchangeable in AD4ETH and AD4USB devices and it is set to 115 200 Bd. Default communication speed in AD4RS is 9 600 Bd. Codes of communication speeds are in table on the right side:	Speed [Bd]	Format 97 code	Format 66 code
	110	00H	0
	300	01H	1
	600	02H	2
	1 200	03H	3
	2 400	04H	4
	4 800	05H	5
	9 600	06H	6
	19 200	07H	7
	38 400	08H	8
	57 600	09H	9
	115 200	0AH	A
	230 400	0BH	B

Examples:

Request:
2AH,61H,00H,05H,FEH,02H,F0H,7FH,0DH
Communication parameters reading using universal address FEH.
Response:
2AH,61H,00H,07H,04H,02H,00H,04H,06H,5DH,0DH
Address 04H, communication speed 9600 Bd.

In 66 format:

Request: „CP“ (Communication Parameter)

Response: (ACK „0“)(address)(speed)

Legend: (address) see part Address on page 8.

(speed) Communication speed code according to the table for speed parameter.

Example: *Request with universal address:* *$1CP↵

Response – Address B, speed 9600Bd (code 6): *B10B6↵

Address Setup using Serial Number

This istruction enables the user to set the address based on the unique serial number of the device. This instruction is handy when the superior system or operating staff lose the address of a device which is on the same communication line with other devices.

The serial number is shown on the product in the following structure *[product-numbr].[hardware-version].[software-version]/[serial-number]* for example like this: *0227.00.03/0001*

Request:

Instruction code: EBH

Parameters: (new_address)(product_number)(serial_number)

new_address	New address of the device	length: 1 byte
New device's address in Spinel protocol.		

product_number	Product number	length: 2 bytes
Product number is stated on the label. For the device with 0227.00.03/0001 on the label, 227 is product number.		

serial_number	Serial number of the product	length: 2 bytes
Serial number is stated on the label. For the device with 0227.00.03/0001 on the label, 1 is product number. This number can be read by instruction „Manufacturer data reading“ (see page 29).		

Response:

Acknowledge code: ACK 00H

Examples:

Request:
2AH,61H,00H,0AH,FEH,02H,EBH,32H,00H,C7H,00H,65H,21H,0DH
New address 32H, product number 199 (= 00C7H), serial number 101 (= 0065H).
Response:
2AH,61H,00H,05H,32H,02H,00H,3BH,0DH
Address has been changed – device responses with new address.

Calibration

It is not necessary to calibrate the device before use. It has been calibrated in the factory based on the requirements specified in the order.

Calibration procedure:

By improper modification of calibration constants you can deteriorate the fuctioning of the device. The calibration shall be only carried out using devices with al least one-level higher accuracy compared to AD4xxx!

1) On the calibrated input bring the **H** value close to the **R** input range.
2) Carry out several measurements (for example 5) and calculate the average. The resulting value is **Mh**.
3) Convert **Mh** to the whole range:

$$M = Mh \times \frac{R}{H}$$

4) Calculate the constant **K**:

$$K = D \times \frac{65536}{M}$$

5) Write **K** constant into AD4xxx and verify the correctness by measuring.

Calibration constants writing

Instruction writes calibration constants. It is possible to write one through four constants at once.

Request:

Instruction code: 12H

Parameters: (channel)(constant)

channel	Number of channel	length: 1 byte
Channel number from range 1 to 4.		

constant	Calibration constant	length: 2 bytes
Calibration constant. Bytes are entered in order MSB:LSB.		

Response:

Acknowledge code: ACK 00H

Reading of calibration constants

Instruction reads calibration constants including the data about how many divisions are for minimal and maximal limits.

Request:

Instruction code: 13H

Parameters: (channel)

Channel	Channel number	length: 1 byte
Chanel number from 1 to 4.		

Response:

Acknowledge code: ACK 00H

Parameters: (constant)(min)(max)

constant	Calibration constant	length: 2 bytes
Calibration constant. Two bytes in order MSB:LSB		

min	Minimal value	length: 2 bytes
States how many divisions matches minimal value. 16bit value with sign. Typically 0.		

max	Maximal value	length: 2 bytes
States how many divisions matches maximal value. 16bit value with sign. Typically 10 000.		

Measurement type setting

Sets the way of conversion for measured values. Available are voltage, current and current range from 4 to 20 mA.

Request:

Instruction code: 1AH

Parameters: (channel)(type)

channel	Channel number	length: 1 byte
Chanel number from 1 to 4.		

typ	Measuring type	length: 1 byte
00H for standard conversion – voltage ranges 01H for current range from 4 to 20 mA 02H for standard conversion – other current ranges		

Response:

Acknowledge code: ACK 00H

Reading of measurement type settings

Finds out what type of measuring is set. It can be voltage, current or special current from 4 to 20 mA.

Request:

Instruction code: 1BH

Response:

Acknowledge code: ACK 00H

Parameters: (typ)

Type	Measurement type	length: 1 byte
00H for standard conversion – voltage ranges 01H for current range from 4 to 20 mA 02H for standard conversion – other current ranges		

RAW measurement

Reads the last measured values from all the channels of the converter. The value measured directly by the internal A/D converter is sent without any subsequent calculation. This instruction facilitates the procedure of calibration and synchronization. When the device is in operation, it is more suitable to use the instruction for "Single measuring" as decribed on page 11.

Request:

Instruction code: 5FH

Parameters: (const)

const	constant	length: 1 byte
Constant 00H. This byte ensures compatibility with other devices and allows further features installed.		

Response:

Acknowledge code: ACK 00H

Parameters: {(chn_1)($status_1$)($value_1$)} {…} {…} {(chn_4)($status_4$)($value_4$)}

chn	Channel number	length: 1 byte
This byte designates channel number and it holds for all following bytes until next *chn* byte. In essence that means that following bytes (status, value) attach to channel with designated number. It ranges from 01H to 04H according to channel number.		

status	Status of the measured-out value	length: 1 byte
Status of measured-out value for the channel with the number stated in *chn* byte.		
bit 3	0 = value is in measuring range	
	1 = value is above upper limit (overflow)	
bit 7 (MSb)	0 = invalid value	
	1 = valid value	

value	Measured-out value	length: 2 bytes
Measured value from the channel with number stated in previous byte *chn*. 16-bit measured-out value. Bytes are in order MSB;LSB.		

Additional

Name and version reading

Reads name of the device, version of inner software and list of possible communication formats. Set by manufacturer.

Request:

Instruction code: F3H

Response:

Acknowledge code: ACK 00H

Parameters: (string)

string	Name and version	length: 1 byte
One of following texts according to the type of device: AD4ETH; v0293.01.04; f66 97 AD4USB; v0295.01.04; f66 97 AD4RS; v0294.01.04; f66 97 There can also be other information in the string. Such information's are separated by semicolon and break. Lower case character states the type of information. *Example:* AD4ETH; v0293.01.04; f66 97; t1; s358; dDG21		

Examples:

Request:
2AH,61H,00H,05H,FEH,02H,F3H,7CH,0DH
Command for reading of name and version.
Response:
2AH,61H,00H,20H,31H,02H,00H,41H,44H,34H,45H,54H,48H,3BH,20H,76H,30H,32H,39H,33H,2EH,30H,31H,2EH,30H,32H,3BH,20H,66H,36H,36H,20H,39H,37H,0CH,0DH
Example of device's response: AD4ETH (AD4ETH; v0293.01.02; f66 97).

In 66 format:

Request: „?“

Response: (ACK „0“)

Example: *Request*

**B1?↵*

Response:

**B10 AD4 ETH; V0293.01.02; F66 97↵*

Note: There can also be other information in the instruction. Such informations are separated by semicolon and break. Lower case character states the type of information.

(Example: AD4ETH*; v0293.01.02; f66 97; t1; s358; dDG21*)

Manufacturer data reading

Instruction reads the data set by manufacturer.

Request:

Instruction code: FAH

Response:

Acknowledge code: ACK 00H

Parameters: (product_number)(serial_number)(other)

product_number	length: 2 bytes
Product number. For the device with 0227.00.03/0001 on the label, 227 is product number.	

serial_number	length: 2 bytes
Device's serial number. For the device with 0227.00.03/0001 on the label, 1 is serial number.	

other	length: 4 bytes
Other Manufacturer data.	

Examples:

Request:
2AH,61H,00H,05H,FEH,02H,FAH,75H,0DH
Response:
2AH,61H,00H,0DH,35H,02H,00H,00H,C7H,00H,65H,20H,05H,09H,23H,B3H,0DH
Product number is 199 (= 00C7H) and serial number is 101 (= 0065H).

User Data Saving

The instruction saves the required user data into relevant memory. The device will remember data saved in user data memory even after power supply failure of reset. This memory space is suitable for example for the specification of the device's location etc.

Request:

Instruction code: E2H

Parameters: (position)(data)

position	length: 1 byte
Address of position in memory where the entered data will be saved. Ranges from 00H to 0FH.	

data	length: 1 to 16 bytes
Any user data. Memory has a capacity of 16 bytes, if the first entry is saved in first position. In case there is not enough space in memory for the user data, the device answers with error and writes nothing. (For example if you try to write to memory address 0CH, you can only write 4 bytes max.)	

Response:

Acknowledge code: ACK 00H

Examples:

Request:
2AH,61H,00H,0FH,31H,02H,E2H,00H,53H,74H,6FH,72H,61H,67H,65H,20H,41H,1AH,0DH
Writing of string *Storage A* (53H,74H,6FH,72H,61H,67H,65H,20H,41H).
Response:
2AH,61H,00H,05H,31H,02H,00H,3CH,0DH
String has been written.

In 66 format:

Request: „DW"(position)(data) *(Data Write)*

Response: (ACK „0")

Legend: (position) Address of position in memory where the entered data will be saved. Ranges from 0-9 or A-F.

(data) 1 through 16 bytes; Any user data. From interval 0-9 or A-F.

Example: *Request*

**B1DW0STORAGE A↵*

Response

**B10↵*

Saved User Data Reading

This instruction reads the data saved in the user data memory. The device will remember data saved into this memory even after power supply failure of reset. This memory space is suitable for example for the specification of measuring points.

Request:

Instruction code: F2H

Response:

Acknowledge code: ACK 00H

Parameters: (data)

data	length: 16 bytes
User data.	

Exampley:

Request:
2AH,61H,00H,05H,31H,02H,F2H,4AH,0DH
Response:
2AH,61H,00H,15H,31H,02H,00H,53H,74H,6FH,72H,61H,67H,65H,20H,41H,20H,20H,20H,20H,20H,20H,20H,16H,0DH
There is a string „Storage A“ saved in user data.

In 66 format:

Request: „DR“ *(Data Read)*

Response: (ACK „0“)(data)

Legend: (data) 1 through 16 bytes; User data.

Example: *Request*

**B1DR↵*

Response

**B10STORAGE A↵*

Input Name Saving

Enables the user to save a unique character string for each input. This function is suitable for the specification of individual inputs.

This memory space uses the control software and is also used in AD4ETH for input names saving. For these reasons we do not recommend manipulating with this memory space AD4.

Request:

Instruction code: 2BH

Parameters: (input)(data)

input	length: 1 byte
Number of input (channel) from 01H to 04H.	

data	length: 21 bytes
Any user data.	

Response:

Acknowledge code: ACK 00H

Examples:

Request:
2AH,61H,00H,1BH,31H,02H,2BH,01H,30H,4BH,6FH,74H,65H,6CH,6EH,61H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,FCH,0DH
Writing of title "0Storage" for input 1 (unused bytes are filled by zeros “0”).
Response:
2AH,61H,00H,05H,31H,02H,00H,3CH,0DH
String has been written.

Reading of input title

Read input title.

Request:

Instruction code: 3BH

Parameters: (input)

input	length: 1 byte
Number of input (channel) from 01H to 04H.	

Response:

Acknowledge code: ACK 00H

Parameters: (data)

data	length: 21 byte
Data saved for given input.	

Examples:

Request:
2AH,61H,00H,06H,31H,02H,3BH,01H,FFH,0DH
Response:
2AH,61H,00H,1AH,31H,02H,00H,30H,4BH,6FH,74H,65H,6CH,6EH,61H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,29H,0DH
Saved string is "0Storage" (unused bytes are filled by zeros “0”).

Status Setup

Sets the status of the device. User defined byte which can be used to find out the device status. It is possible for the user to enter this byte as requested. It serves as memory space suitable for example of user's own specification of device status. (Set to zero after reset or power supply restoration.)

Request:

Instruction code: E1H

Parameters: (status)

status	length: 1 byte
Device's status. After turning the device on or reset (even software one) the device, default status 00H is set. If it is changed by instruction Status Setting, it is easily identifiable at any point.	

Response:

Acknowledge code: ACK 00H

Examples:

Request:
2AH,61H,00H,06H,01H,02H,E1H,12H,78H,0DH
Setting of status 12H.
Response:
2AH,61H,00H,05H,01H,02H,00H,6CH,0DH
Acknowledge.

In 66 format:

Request: „SW"(status) (Status Write)

Response: (ACK „0")

Legend: (status) character frim interval „break" to „~" (32 – 126 in ASCII code)

Example: *Request – character A*

**B1SWA↵*

Response

**B10*

Reading of status

Reads device's status. It is user defined byte, which can be used for finding out the device's status.

Request:

Instruction code: F1H

Response:

Acknowledge code: ACK 00H

Parameters: (status)

status	length: 1 byte
Device's status. After turning the device on or reset (even software one) the device, default status 00H is set.	

Examples:

Request:
2AH,61H,00H,05H,01H,02H,F1H,7BH,0DH
Response:
2AH,61H,00H,06H,01H,02H,00H,12H,59H,0DH
Device's status is set to 12H.

In 66 format:

Request: „SR“ (Status Read)

Response: (ACK „0“)(character)

Legend: (character) character from interval „break“ to „~“ (32 – 126 in ASCII code)

Example: *Request*

**B1SR↵*

Response

**B10A↵*

Reading of communication errors

Instruction returns number of communication errors that occurred from the time when the device has been turned on, or from the time of last reading.

Request:

Instruction code: F4H

Response:

Acknowledge code: ACK 00H

Parameters: (errors)

errors	length: 1 byte
Number of communication errors that occurred from the time when the device has been turned on, or from the time of last reading. Following events are considered as errors: • Prefix is expected, but another byte arrives instead. • Checksum SUMA does not match. • Message is incomplete.	

Examples:

Request:
2AH,61H,00H,05H,01H,02H,F4H,78H,0DH
Response:
2AH,61H,00H,06H,01H,02H,00H,05H,66H,0DH
5 errors occurred since the device has been turned on.

Checksum Permission

This function enables the user to switch off the checkup of the checksum accuracy. This instruction is handy for the synchronization of applications. When entering instructions manually using a terminal, it is not necessary to accurately enter the checksum (the one but last byte).

We do not recommend you to switch the checkup off in other situations than in testing operation of the device. The checksum serves as a protection against data corruption during their transfer over the communication line. The checkup is switched on by default.

Request:

Instruction code: EEH

Parameters: (state)

State	length: 1 byte
00H for turning off the control of checksum. 01H for turning on the control of checksum.	

Response:

Acknowledge code: ACK 00H

Examples:

Request:
2AH,61H,00H,06H,01H,02H,EEH,01H,7CH,0DH
Control turned on.
Response:
2AH,61H,00H,05H,01H,02H,00H,6CH,0DH
Command confirmed.

Checksum – settings reading

Finds out current checksum control settings. (See the description of previous instruction „Checksum Permission“.)

Request:

Instruction code: FEH

Response:

Acknowledge code: ACK 00H

Parameters: (state)

state	length: 1 byte
00H checksum control off. 01H checksum control on.	

Examples:

Request:
2AH,61H,00H,05H,01H,02H,FEH,6EH,0DH
Response:
2AH,61H,00H,06H,01H,02H,00H,01H,6AH,0DH
Checksum control on.

Reset

Executes reset of the device. Device gets in the same state as if it was just turned on.

Request:

Instruction code: E3H

Response:

Acknowledge code: ACK 00H

Examples:

Request:
2AH,61H,00H,05H,01H,02H,E3H,89H,0DH
Response:
2AH,61H,00H,05H,01H,02H,00H,6CH,0DH
Reset is executed after this response is sent.

In 66 format:

Request: „RE“ *(Reset)*

Response: (ACK „0“)

Example: *Request:* **B1RE↵*

Response: **B10↵*

Note: Reset is executed after this response is sent.

Auxiliary Instructions

The following instructions have been added purposefully to support the WEB interface in the Ethernet version of AD4ETH. They can also be found in AD4USB and AD4RS. They are system instructions designed only for advanced users.

Single Measurement with Conversion

This instruction reads the last measured values from all or just some of the converter channels. It returns values converted into decimal numbers (32bit float according to to IEEE 754), and into a string of ten characters including the decimal point.

Request:

Instruction code: 58H

Parameters: (chn)

chn	Channel	length: 1 až 4 byte
One through four numbers of channels that are to be red. If the value 00H is entered, all channels will be red.		

Response:

Acknowledge code: ACK 00H

Parameters: {(chn_1)($status_1$)($value_1$)} {…} {…} {(chn_4)($status_4$)($value_4$)}

chn	Channel number	length: 1 byte
This byte designates channel number and it holds for all following bytes until next *chn* byte. In essence that means that following bytes (status, value) attach to channel with designated number. It ranges from 01H to 04H according to channel number.		

status	Status of the measured-out value	length: 1 byte
Status of measured-out value for the channel with the number stated in *chn* byte.		
bit 3	0 = value is in measuring range	
	1 = value is above upper limit (overflow)	
bit 7 (MSb)	0 = invalid value	
	1 = valid value	

value	Measured-out value	length: 14 byte
Measured value from the channel with number stated in previous byte *chn*. Values are sent in three formats simultaneously. First is 16bit value from 0 to 10 000 (integer in order MSB:LSB). Next are two values converted for actual range according to actual settings. As a 32 bit float in complaint with IEEE 754[6] and ASCII as ten characters of decimal number. Values are sent in described order. *Example:* Value 9215.85 is formulated as: `0AH,58H,46H,0FH,FFH,66H,20H,20H,20H,39H,32H,31H,35H,2EH,38H,35H` Part INT: `0AH,58H` (2648) Part IEEE 754: `46H,0FH,FFH,66H` Part ASCII: `20H,20H,20H,39H,32H,31H,35H,2EH,38H,35H` (9215.85)		

[6] Description of IEEE 754 is available here: http://en.wikipedia.org/wiki/IEEE_754

Examples:

Request:
2AH,61H,00H,06H,31H,02H,58H,02H,E1H,0DH
Command for reading channel 2.
Response:
2AH,61H,00H,17H,31H,02H,00H,02H,80H,15H,3AH,41H,ADH,E3H,53H,20H,20H,20H,20H,20H,32H,31H,2EH,37H,34H,99H,0DH
From channel 2 was measured value 21.74. Channel number: 02H Status: 80H Part INT: 15H,3AH (5434) Part IEEE 754: 41H,ADH,E3H,53H Part ASCII: 20H,20H,20H,20H,20H,32H,31H,2EH,37H,34H (21.74)

Conversion and Display Setup

This instruction enables the user to set simultaneously various values used for the WEB interface, namely the name of the channel, text description of the measuring range, text description of units, display setup, number of decimal positions for rounding as well as the additive and multiplicative constant for the equation of the line based on which the basic range of 0 to 10000 divisions shall be converted into the range set by the user.

Request:

Instruction code: 1EH

Parameters: {[channell][name][range][units][display][float][multi-f][multi-a][add-f][add-a][type]}{…}

channel Channel number	length: 1 byte id: 01H
This byte designates channel number and it holds for all following bytes until next *channel* byte. In essence that means that following bytes (status, value) attach to channel with designated number. It ranges from 01H to 04H according to channel number.	

name Channel name	length: 21 bytes id: 11H
Memory position for saving the channel label. It is the same string as from instruction “Input Name Saving” on page 32.	

range Measuring range	length: 15 bytes id: 12H
Text string defining measured range. For example: „0 – 100 V“	

units Set unit	length: 5 bytes id: 13H
Text string describing set units. For example: „V“, „A“, „pA“, „m“, etc.	

display Parameters of display	length: 5 bytes id: 14H
Parameters of display of channel on the WEB page. (This variable is not yet used.)	

decimals Number of decimals	length: 1 byte id: 15H
Number of displayed rounded-off decimals.	

multi-f Multiplicative constant – float	length: 4 bytes id: 16H
Multiplicative constant for conversion of measured-out value. Current value from converter is converted to decimal number according to this relation: $$result = multi \times measured + add$$ multi – *multi-f* represents multiplicative constant as float[7] measured – measured-out value from A/D converter add – additive constant *Note:* It is possible to enter either parameter multi-f or the parameter multi-a for single channel.	

multi-a Multiplicative constant – ASCII	length: 10 bytes id: 17H
Multiplicative constant for conversion of measured-out value. Current value from converter is converted to decimal number according to this relation: $$result = multi \times measured + add$$ multi – *multi-a* represents multiplicative constant as ASCII value. ASCII value is 10bytes including decimal point. measured – measured-out value from A/D converter add – additive constant *Note:* It is possible to enter either parameter multi-f or the parameter multi-a for single channel.	

add-f Additive constant – float	length: 4 bytes id: 18H
Additive constant for conversion of measured-out value. Current value from converter is converted to decimal number according to this relation: $$result = multi \times measured + add$$ multi – multiplicative constant measured – measured-out value from A/D converter add – *add-f* represents multiplicative constant as float[7] *Note:* It is possible to enter either parameter add-f or the parameter add-a for single channel.	

[7] Description of IEEE 754 is available here: http://en.wikipedia.org/wiki/IEEE_754

add-a Additive constant – ASCII	length: 10 byte id: 19H
Multiplicative constant used for conversion of measured-out value. Current measured-out value from converter is converted to decimal number based on this relation: $$result = multi \times measured + add$$ multi – multiplicative constant measured – measured-out value from A/D converter add – *add-a* represents additive constant (add) entered as an ASCII value. ASCII value is 10 bytes aligned to the right including decimal point. *Note:* It is possible to enter either parameter add-f or the parameter add-a for single channel.	

typ Measurement type	length: 1 byte id: 20H
Measurement type determines whether it is voltage range (00H), special current range (01H; 4 to 20 mA) or other current ranges (02H).	

Response:

Acknowledge code: ACK 00H

Examples:

Request:
`2AH,61H,00H,15H,31H,02H,1EH,01H,01H,13H,20H,20H,20H,B0H,43H,01H,03H,13H,20H,20H,6BH,50H,61H,33H,0DH`
Setting the range to °C for channel 1 and kPa for channel 3. Channel number (id 01H): `01H` Units (id 13H): `20H,20H,20H,B0H,43H` (°C) Channel number (id 01H): `03H` Units (id 13H): `20H,20H,6BH,50H,61H` (kPa)
Response:
`2AH,61H,00H,05H,31H,02H,00H,3CH,0DH`
Command executed.

Conversion and Display Reading

This instruction read the setup of the name of the channel, text description of the measuring range, text description of units, display setup, number of decimal positions for rounding as well as the additive and multiplicative constant.

Request:

Instruction code: 1FH

Parameters: (channel)

channel	Channel	length: 1 byte
Number of the channel that is to be red.		

Response:

Acknowledge code: ACK 00H

Parameters: {[channel][name][range][units][display][decimals][multi-f][multi-a][add-f][add-a][type]}{…}

Description of the parameters is the same as for the previous instruction.

Examples:

Request:

`2AH,61H,00H,06H,31H,02H,1FH,01H,1BH,0DH`

Reading parameters for channel 1.

Response:

`2AH,61H,00H,5DH,31H,02H,00H,01H,01H,11H,20H,53H,74H,75H,64H,6EH,61H,20H,7AH,61H,20H,68H,75H,6DH,6EH,79H,20H,20H,20H,20H,20H,12H,20H,20H,20H,20H,20H,2DH,35H,35H,20H,2BH,31H,35H,30H,B0H,43H,13H,20H,20H,20H,B0H,43H,14H,41H,42H,43H,44H,45H,15H,02H,16H,3CH,B4H,39H,58H,17H,20H,20H,20H,20H,20H,30H,2EH,30H,32H,32H,18H,C2H,5CH,00H,00H,19H,20H,20H,20H,2DH,35H,35H,2EH,30H,30H,30H,20H,01H,F4H,0DH`

Channel 1 parameters are:

Channel number (id 01H): `01H`
Name (id 11H): `20H,53H,74H,75H,64H,6EH,61H,20H,7AH,61H,20H,68H,75H,6DH,6EH,79H,20H,20H,20H,20H,20H` (STORAGE)
Range (id 12H): `20H,20H,20H,20H,20H,2DH,35H,35H,20H,2BH,31H,35H,30H,B0H,43H` (-55 +150C)
Units (id 13H): `20H,20H,20H,B0H,43H` (°C)
Display parameters (id 14H): `41H,42H,43H,44H,45H`
Number of decimals (id 15H): `02H`
Multiplicative constant as float (id 16H): `3CH,B4H,39H,58H` (0.022)
Multiplicative constant as ASCII (id 17H): `20H,20H,20H,20H,20H,30H,2EH,30H,32H,32H` (0,022)
Additive constant as float (id 18H): `C2H,5CH,00H,00H` (-55)
Additive constant as ASCII (id 19H): `20H,20H,20H,2DH,35H,35H,2EH,30H,30H,30H` (-55)
Measurement type (id 20H): `01H` (Current range from 4 to 20 mA)

Values Monitoring Setup

This instruction enables the user to set for each channel an upper and lower limit to be monitored. Exceeding the upper limit or dropping below the lower limit will result in sending an automatic message to the superior system.

This instruction can also be used for setting the hysteresis of the set limits. The hysteresis applies below the upper limit and above the lower limit. The principle of hysteresis is described in Addition A – Hysteresis on page 49.

Depending on the setup, it is possible in AD4ETH to allow messages to be sent via the WEB interface in case these limits are violated. The WEB interface graphically signalizes any violation of the limits.

Request:

Instruction code: 1CH

Parameters: {[channell][attributes][lower-limit-f][lower-limit-a][upper-limit-f][upper-limit-a][hysteresis-f][hysteresis-a][error]}{…}

channel Channel number	length: 1 byte id: 01H
This byte designates channel number and it holds for all following bytes until next *channel* byte. In essence that means that following bytes (status, value) attach to channel with designated number. It ranges from 01H to 04H according to channel number.	

attributes Other parameters	length: 1 byte id: 12H
Some other parameters are in this byte. Default value: 00H	
bit 7 (MSb)	0 = Values monitoring off for this channel.
	1 = Values monitoring on for this channel.

lower-limit-f Lower limit – float	length: 4 bytes id: 13H
Lower monitoring limit in float format.[8] *Note:* It is possible to enter either parameter lower-limit-f or the parameter lower-limit-a for single channel.	

lower-limit-a Lower limit – ASCII	length: 10 bytes id: 14H
Lower monitoring limit in ASCII format. ASCII value is 10 bytes aligned to the right side including decimal point. *Note:* It is possible to enter either parameter lower-limit-f or the parameter lower-limit-a for single channel.	

[8] Popis normy IEEE 754 je k dispozici naExample zde: http://en.wikipedia.org/wiki/IEEE_754

upper-limit-f Upper limit – float	length: 4 bytes id: 15H
Upper monitoring limit in float format.[8] *Note:* It is possible to enter either parameter upper-limit-f or the parameter upper-limit-a for single channel.	

Upper-limit-a Upper limit – ASCII	length: 10 bytes id: 16H
Upper monitoring limit in ASCII format. ASCII value is 10 bytes aligned to the right side including decimal point. *Note:* It is possible to enter either parameter upper-limit-f or the parameter upper-limit-a for single channel.	

hysteresis-f Hysteresis – float	length: 4 bytes id: 17H
Hysteresis in float format.[9]	

hysteresis-a Hysteresis – ASCII	length: 10 byte id: 18H
Hysteresis in ASCII format. ASCII value is 10 bytes aligned to the right side including decimal point. *Note:* It is possible to enter either parameter hysteresis-f or the parameter hysteresis-a for single channel.	

error Overflow behavior	length: 1 byte id: 1AH
This byte states what happens if A/D converter range overflow occurs.	
00H	No information about overflow is sent.
01H	Information about overflow is sent automatically.

Response:

Acknowledge code: ACK 00H

[9] Popis normy IEEE 754 je k dispozici naExample zde: http://en.wikipedia.org/wiki/IEEE_754

Automatic Response:

This response is generated in case there are set limits and they are violated or in case the measured value deviates from the physical range of the internal A/D converter.

Acknowledge code: ACK 0FH

Parameters: [event][channel][status][value]

event Number of the event source	length: 1 byte id: 01H
This byte specifies the source of event. In case of exceeding limits or range, the automatic message can contain this byte to separate such errors. This byte's value is 30H.	

channel Channel number	length: 1 byte id: 02H
This byte designates the number of channel responsible for sending automatic instructions. It ranges from 01H to 04H according to channel number.	

status Measured-out value status		length: 1 byte id: 03H
Status of measured-out value for channel with channel number given in byte *channel*.		
bites 0 to 3 (lower nibble)	0000 = measured-out value is in range	
	0001 = exceeding lower limit of measured range	
	0010 = exceeding upper limit of measured range	
	0100 = underflow of physical range of A/D converter (can occur only at 4 to 20 mA current range)	
	1000 = overflow of physical range of A/D converter	
bit 7 (MSb)	0 = measured-out value is invalid	
	1 = measured-out value is valid	

value Measured-out value	length: 14 byte id: 04H
Measured value from the channel with number stated in previous byte *channel*. Values are sent in three formats simultaneously. First is 16bit value from 0 to 10 000 (integer in order MSB:LSB). Next are two values converted for actual range according to actual settings. As a 32 bit float in complaint with IEEE 754[10] and ASCII as ten characters of decimal number. Values are sent in described order. *Example:* Value 9215,85 is formulated as: `0AH,58H,46H,0FH,FFH,66H,20H,20H,20H,39H,32H,31H,35H,2EH,38H,35H` Part INT: `0AH,58H` (2648) Part IEEE 754: `46H,0FH,FFH,66H` Part ASCII: `20H,20H,20H,39H,32H,31H,35H,2EH,38H,35H` (9215.85)	

[10] Description of IEEE 754 is available here: http://en.wikipedia.org/wiki/IEEE_754

Examples:

Request:
2AH,61H,00H,19H,31H,02H,1CH,01H,01H,12H,80H,14H,20H,20H,20H,20H,32H,35H,2EH,30H,30H,30H,15H,41H,A0H,00H,00H,C9H,0DH
Setting the upper and lower limit for channel 1. Meaning of particular items: Channel number (id 01H): 01H Attributes (id 12H): 80H Upper limit as ASCII (id 14H): 20H,20H,20H,20H,32H,35H,2EH,30H,30H,30H (25.000) Lower limit as float (id 15H): 41H,A0H,00H,00H (20.000)
Response:
2AH,61H,00H,05H,31H,02H,00H,3CH,0DH
Command receiving confirmation.
Automatic response:
2AH,61H,00H,1CH,31H,13H,0FH,01H,30H,02H,02H,03H,82H,04H,18H,BBH,41H,CAH,97H,8CH,20H,20H,20H,20H,20H,32H,35H,2EH,33H,32H,ACH,0DH
Automatic information about exceeding the upper limit 25.0 on channel 2. Current value is 25.23. Event number (id 01H): 30H Channel number (id 02H): 02H Attributes (id 03H): 82H Current value (id 04H): As INT: 18H,BBH as float: 41H,CAH,97H,8CH As ASCII: 20H,20H,20H,20H,20H,32H,35H,2EH,33H,32H

Reading the settings of values monitoring

This instruction allows us to read data written by previous instruction.

Request:

Instruction code: 1DH

Parameters: [channel]

Response:

Acknowledge code: ACK 00H

Parameters: {[channell][attributes][lower-limit-f][lower-limit-a][upper-limit-f][upper-limit-a][hysteresis-f][hysteresis-a][error]}{…}

Meaning of all parameters is stated in previous instruction description.

Examples:

Request:
2AH,61H,00H,06H,31H,02H,1DH,01H,1DH,0DH
Command for reading the parameters of monitoring values from channel 1.
Response:
2AH,61H,00H,3BH,31H,02H,00H,01H,01H,12H,80H,13H,41H,C8H,00H,00H,14H,20H,20H,20H,20H,32H,35H,2EH,30H,30H,30H,15H,41H,A0H,00H,00H,16H,20H,20H,20H,20H,32H,30H,2EH,30H,30H,30H,17H,3EH,A6H,66H,66H,18H,20H,20H,20H,20H,20H,30H,2EH,33H,32H,35H,1AH,00H,60H,0DH
Meaning of highlighted items Channel number (id 01H): 01H Attributes (id 12H): 80H Upper limit as float (id 13H): 41H,C8H,00H,00H (25.000 – twenty-five point zero) Upper limit as ASCII (id 14H): 20H,20H,20H,20H,32H,35H,2EH,30H,30H,30H Lower limit as float (id 15H): 41H,A0H,00H,00H (20.000) Lower limit as ASCII (id 16H): 20H,20H,20H,20H,32H,30H,2EH,30H,30H,30H Hysteresis as float (id 17H): 3EH,A6H,66H,66H (0.325) Hysteresis as ASCII (id 18H): 20H,20H,20H,20H,20H,30H,2EH,33H,32H,35H,1AH,00H Error (id 1AH): 00H

ADDITION A – HYSTERESIS

Hysteresis enables user to eliminate the fluctuation of the measured values. During measuring the measured value often happens to be unstable and it quickly fluctuates around the real value either naturally or due to the noise of the device.

In AD4 it is possible to set an upper and lower limit to be monitored. Exceeding the upper limit (or dropping below the lower limit) will result in an automatic message sent to the superior system by e-mail etc.

If the measured value fluctuates quickly around the upper or lower limit, the information about limit violations would be sent repeatedly even though the value does not change at all. This can be avoided by setting a so called hysteresis. It defines dead bands in which the information is not being sent.

The principle of the hysteresis is demonstrated in the following two charts showing temperature development.

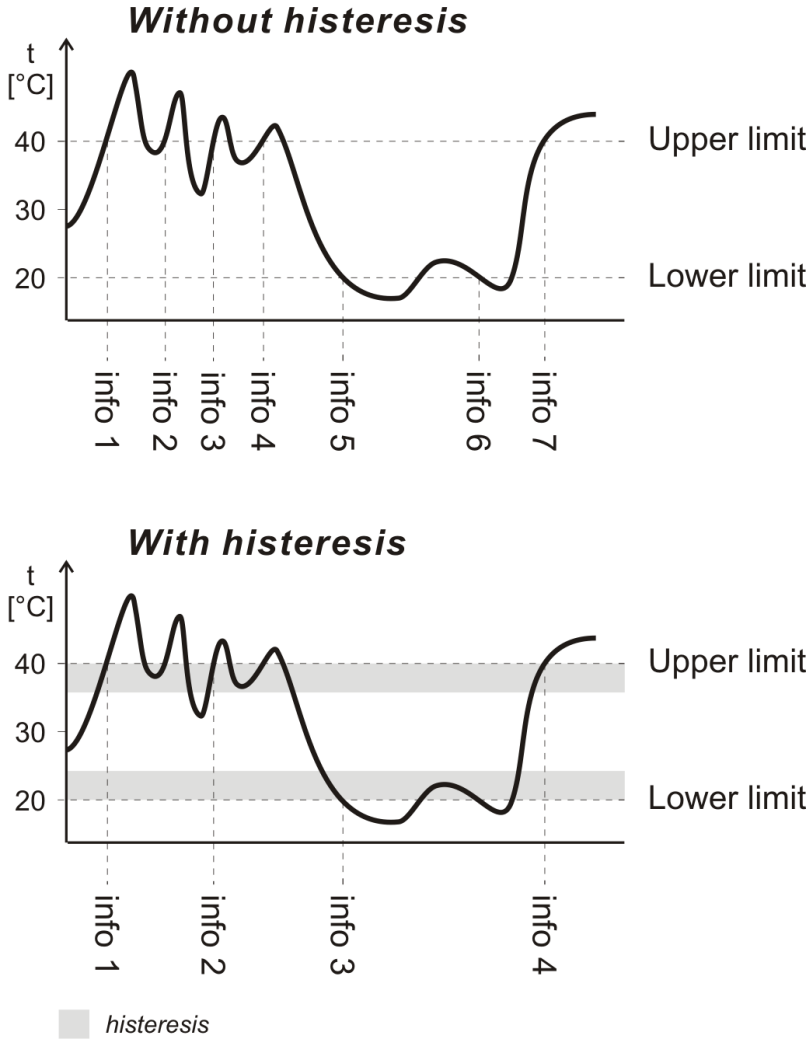


Fig. 1 – hysteresis of temperature limits

The hysteresis value applies below the upper limit (or above the lower limit) – see Fig. 1.

In the upper chart the hysteresis has been switched off (set to 0). The measured value fluctuates around the set limits causing frequent sending of relevant information to the superior system. If the hysteresis is set – like in the second chart – the information is sent only in case of a more significant change, but not just during the fluctuation of the value around the limit.

The hysteresis range needs to be chosen based on the measured value and the size of measured value fluctuation.

Papouch s.r.o.

Data transmission in industry, line and protocol conversions, RS232/485/422/USB/Ethernet/GPRS/ WiFi, measurement modules, intelligent temperature sensors, I/O modules, and custom-made electronic applications.

Address:

Strašnická 3164/1a
102 00 Praha 10
Czech Republic

Telephone:

+420 267 314 267-8
+420 602 379 954

Fax:

+420 267 314 269

Internet:

www.papouch.com

E-mail:

info@papouch.com