UMB-ASCII 2.0
Universal-Measurement-Bus
Communication Protocol
for Meteorological Sensors

WS3000-UMB
WS3100-UMB
WS100-UMB
VS100k-UMB
SHM31-UMB

Stand 26.02.2018
Protocol Version 2.0
Document Version 1.1

G. LUFFT MESS- UND
REGELTECHNIK GMBH

POSTFACH 4252
70719 FELLBACH
TEL. 49 (711) - 51822-0
FAX 49 (711) - 51822-41

Contents

1 Version History

Document Version	Date	By	Modifications
1.1	14.02.2018	GS/ITSC	Translation of German version 1.1
	26.02.2018	RZF	Error codes added

2 Introduction

UMB-ASCII 2.0 is a human readable protocol for data acquisition from meteorological sensors. When designing the protocol, the focus was on a high degree of adaptability to meet the requirements of a wide range of measurement acquisition devices.

The protocol permits automatic telegram transmission for measurement data acquisition without prior request.

3 Syntax

Expressions in brackets [] are optional:

Request <Add>:<Nr>:<Payload><CR><LF>
Response <STX><Add>:<Nr>:<Payload>:<Status>:<Checksum><CR><LF><EOT>

Payload <Cmd>[;<Param$_0$>;…;Param$_N$][=<Value$_0$>;…;<Value$_N$>]

The blocks of the protocol are separated by colon ':'. This block separator character is adjustable. <Add> is the UMB address of the sensor, four hexadecimal characters with leading zeros, in the range from 0001 to FFFF. Messages with wrong address are ignored by the sensors. The transmitter sets the field <Nr> of the request to two arbitrary hexadecimal characters. The sensor includes the received characters into the <Nr> field of the response. <Nr> may be used e.g. as time reference if it may happen that the response does not follow directly to the request. <Payload> holds the user data of the message. This includes the command with its parameters and values. The <Status> field has two hexadecimal characters with leading zero. In case of unknown or erroneous commands, the payload of the response consists of the command and the related error code. Requests and responses are terminated by characters <CR><LF>. The termination characters are adjustable. The response of a sensor is additionally framed by control characters <STX> und <EOT>.

The payload contains the command with its related data. Commands may have one or more parameters, separated by semicolon. The parameter separator is adjustable. If the payload contains an equals sign (=) the command is handled as a write command. The field <value> may hold, depending on the command, any of the supported data types, including string. More than one value may be transmitted, separated by the parameter separator (semicolon). The (adjustable) decimal separator is a period (.).

The length of a request is limited to 128 characters, the length of a response to 512 characters. Devices may permit longer telegrams. Such extensions are defined in the individual device manuals.

3.1 Checksum

Sensor responses have a check sum <Checksum> for data integrity check. The check sum is the two's complement of the 8-bit sum of all characters, including control characters <STX> and <EOT>, excluding the check sum itself. The check sum is presented as two hexadecimal characters with leading zero.

3.2 Examples

Assumption for following examples: <Add> = 7001, <Nr> =4E

Reset	7001:4E:RST<CR><LF> <STX>7001:4E:RST:00:<CheckSum><CR><LF><EOT>	
Read Float	7001:4E:EF;120<CR><LF> <STX>7001:4E:EF;120=34.5654:00:<CheckSum><CR><LF><EOT>	
WriteByte	7001:4E:ES;34=1<CR><LF> <STX>7001:4E:ES;34=1:00:<CheckSum><CR><LF><EOT>	
Read Channel	7001:4E:CHN;100<CR><LF> <STX>7001:4E:CHN;100=+23.45:00:<CheckSum><CR><LF><EOT>	
	7001:4E:CHN;110<CR><LF> <STX>7001:4E:CHN;110:28:<CheckSum><CR><LF><EOT>	(Channel busy)
	7001:4E:CHN;523<CR><LF> <STX>7001:4E:CHN;523:24:<CheckSum><CR><LF><EOT>	(Channel unknown)

4 Commands

4.1 Overview

The Parameters of the standardized part of the UMB parameter list (1-199) can be adjusted by corresponding setting commands. Each device type may specify additional setting commands for device specific parameters.

EERPOM commands are for internal use only.

Command		R/W	Description
Allgemeine Befehle			
ATI		RW	Auto Transmit Intervall
ATM		RW	Auto Transmit Mode
BAU	BAUdrate	RW	Baudrate
DSC	Device DeSCription	RW	Device Description
ID		RW	Device ID
NAM	Device NAMe	R	Device Name
PBS	Protocol Block Separator	RW	Block Separator (:)
PCR	Protocol Carriage Return	RW	Line Terminator(CRLF)
PDS	Protocol Decimal Separator	RW	Decimal Separator (.)
PEN	Protocol ENd	RW	Message Terminator (EOT)
PPS	Protocol Parameter Separator	RW	Parameter Separator (;)
PRT	PRoTocol	RW	Protocol
PRY	PaRitY	RW	Parity Setting
PST	Protocol STart	RW	Message Start Character (STX)
SRN	SeRialNumber	R	Lufft Serial Number
ES;<Add>[;<Pin>]		RW	EEPROM uint8
EI;<Add>[;<Pin>]		RW	EEPROM uint16
EL;<Add>[;<Pin>]		RW	EEPROM uint32
EF;<Add>[;<Pin>]		RW	EEPROM float32
ED;<Add>[;<Pin>]		RW	EEPROM float64
RST[=<Value>]	ReSeT	RW	Reset, immediately /after <Value> ms
RSD	ReSet Default	R	Reset to Factory Settings
IFO;<Info>	InFO	R	Device Informations
UMB			Change protocol temporarily to UMB-Binary
CHN;<ChnNum>	CHaNnel	R	Read Meas. Channel <ChnNum>
SS;<Num>		R	Read Standard Set <Num>
TNL;<BusNum>;<Len>	TuNnel	W	Tunnel data with length <Len> to <BusNum>

4.2 Error Codes

The values of the <Status> field conform to the definitions of UMB-Binary protocol. UMB-ASCII 2.0 uses a subset of the following UMB-Binary error code list:

<status>	Define	Description
00h	OK	Command succeeded; no error; everything ok
10h	UNBEK_CMD	Unknown command; not supported by this device
11h	UNGLTG_PARAM	Invalid parameter
12h	UNGLTG_HEADER	Invalid header version
13h	UNGLTG_VERC	Invalid command version
14h	UNGLTG_PW	Wrong password for command
15h	UNGLTG_WERT	Invalid value
20h	LESE_ERR	Read error
21h	SCHREIB_ERR	Write error
22h	ZU_LANG	Message too long; max. permitted length given in <maxlength>

23h	UNGLTG_ADRESS	Invalid address / storage location
24h	UNGLTG_KANAL	Invalid channel
25h	UNGLTG_CMD	Command not accepted in current operation mode
26h	UNBEK_CAL_CMD	Unknown test / calibration command
27h	CAL_ERROR	Calibration error
28h	BUSY	Device not ready; e.g. initialisation / calibration active
29h	LOW_VOLTAGE	Low voltage
2Ah	HW_ERROR	Hardware error
2Bh	MEAS_ERROR	Measurement error
2Ch	INIT_ERROR	Error during device initialisation
2Dh	RTOS_ERROR	Operation system error
2Eh	COM_ERROR	Communication error of an internal interface
2Fh	HW_SW_MISMATCH	Hardware and software version do not match
30h	E2_DEFAULT_KONF	Configuration error, device was set to default configuration
31h	E2_CAL_ERROR	Error in calibration data / invalid calibration, measurement not possible
32h	E2_CRC_KONF_ERR	CRC error when loading configuration, device was set to default configuration
33h	E2_CRC_KAL_ERR	CRC error when loading calibration data, measurement not possible
34h	ADJ_STEP1	Calibration Step 1
35h	ADJ_OK	Calibration OK
36h	KANAL_AUS	Channel deactivated
37h	SERVICE_MODE	Service mode active
50h	VALUE_OVERFLOW	Measurement value (+Offset) out of defined display range
51h	VALUE_UNDERFLOW	Measurement value (+Offset) out of defined display range
52h	CHANNEL_OVERRANGE	Measurement value (physically) out of measurement range (e.g. ADC-over range)
53h	CHANNEL_UNDERRANGE	Measurement value (physically) out of measurement range (e.g. ADC-under range)
54h	DATA_ERROR	Data error in measurement data or no valid data available
55h	MEAS_UNABLE	Device / Sensor unable to do valid measurement due to environmental conditions
0x60	FLASH_CRC_ERR	CRC error in flash data
0x61	FLASH_WRITE_ERR	Error when writing to flash memory, e.g. storage location not erased
0x62	FLASH_FLOAT_ERR	Flash memory contains invalid float values
0x63	FLASH_ERR	Flash memory defective, initialisation error
0x64	CONF_ERR	Configuration error
F0h - FEh	Reserved!!	reserved
FFh	UNBEK_ERR	Unknown error

4.3 Commands

UMB-ASCII 2.0 commands are case-insensitive, i.e. upper and lower case are not handled differently: RST = rst = rSt.

4.3.1 Auto Transmit Interval

Command <cmd>: ATI
Parameter: none
Request: ATI ATI=<time>
Response: ATI=<time>

Description: this command reads / sets the interval [sec] between two automatically transmitted standard set messages.

4.3.2 Auto Transmit Mode

Command <cmd>: ATM
Parameter: none
Request: ATM ATM=<setNumber>
Response: ATM=<setNumber>

Description: this command reads / sets the number of the standard set to be transmitted automatically. If <setNumber> is 0, the automatic message transmission is deactivated.

4.3.3 Baudrate

Command <cmd>: BAU
Parameter: none
Request: BAU BAU=<value>
Response: BAU=<value>

Description: this command reads / sets the baud rate using the coding shown in the following table:

<value>	Baud Rate
2	57600 Bd
4	28800 Bd
6	19200 Bd
8	14400 Bd
12	9600 Bd
24	4800 Bd
48	2400 Bd
96	1200 Bd

4.3.4 Device Description

Command <cmd>: DSC
Parameter: none
Request: DSC DSC=<desc>
Response: DSC=<desc>^{40}

Description: this command reads / writes the device description with max. 40 characters. If <desc> has less than 40 characters, the remaining are blank padded.

4.3.5 Device ID

Command <cmd>: ID

Parameter: none

Request: ID ID=<id>

Response: ID=<id>

Description: this command reads / sets the device ID in the range: 1-255

4.3.6 Device Name

Command <cmd>: NAM

Parameter: none

Request: NAM

Response: NAM=<name>^{40}

Description: this Command reads the device name with max. 40 characters. If <name> has less than 40 characters, the remaining are blank padded

4.3.7 Block Separator

Command <cmd>: PBS

Parameter: none

Request: PBS PBS=<asciiValue>

Response: PBS=<asciiValue>

Description: this command reads / sets the ASCII value of the protocol block separator. Default value is 58, a colon ‘:‘. Only special characters (punctuation marks …) are permitted as <asciiValue>.

4.3.8 Line Termination Characters

Command <cmd>: PCR

Parameter: none

Request: PCR PCR=<value>

Response: PCR=<value>

Description: this command reads / sets the line termination characters using the coding shown in following table:

<value>	Line Terminator
0	CRLF (0Dh, 0Ah)
1	CR (0Ah)
2	LF (0Dh)

4.3.9 Decimal Separator

Command <cmd>: PDS

Parameter: none

Request: PDS PDS=<asciiValue>

Response: PDS=<asciiValue>

Description: this command reads / sets the ASCII value of the decimal separator. Default value is 46, a period ‘.‘.Only special characters (punctuation marks …) are permitted as <asciiValue>.

4.3.10 Message End Character

Command <cmd>: PEN

Parameter: none

Request: PEN PEN=<asciiValue>

Response: PEN=<asciiValue>

Description: this command reads / sets the ASCII value of the message end character. Default value is 4, an „end of text“ ‘EOT‘.

4.3.11 Parameter Separator

Command <cmd>: PPS

Parameter: none

Request: PPS PPS=<asciiValue>

Response: PPS=<asciiValue>

Description: this command reads / sets the ASCII value of the protocol parameter separator. Default value ist 59, a semicolon ‘;‘. Only special characters (punctuation marks …) are permitted as <asciiValue>.

4.3.12 Message Start Character

Command <cmd>: PST

Parameter: none

Request: PST PST=<asciiValue>

Response: PST=<asciiValue>

Description: this command reads / sets the ASCII value of the protocol start character. Default value is 2, a „start of text" ‘STX‘.

4.3.13 Protocol

Command <cmd>: PRT

Parameter: none

Request: PRT PRT=<value>

Response: PRT=<value>

Description: this command reads / sets the communication protocol using the coding shown in following table:

<value>	Protocol
0	UMB binary
1	UMB-ASCII
2	Terminal
3	SDI-12
5	MODBUS-RTU
6	MODBUS-ASCII
7	XDR
9	UMB-ASCII 2.0

NOTE: The listed protocols are not supported by all device types. Please refer to the device manual or data sheet for detailed information.

NOTE: The change of communication protocol is effective after restart of the device. Then the access to the device is only possible by the new protocol!

4.3.14 Parity

Command <cmd>: PRY

Parameter: none

Request: PRY PRY=<value>

Response: PRY=<value>

Description: this command reads / sets the parity setting using the coding of following table:

<value>	Protokoll
0	none (8N1)
1	even (8E1)
2	even (7E1)
3	none (8N2)

NOTE: The parity setting should only be modified for Modbus or SDI12 communication.

NOTE: The change of the parity setting is effective after restart of the device!

4.3.15 Serial Number

Command <cmd>: SRN

Parameter: none

Request: SRN

Response: SRN=<serialNumber>

Description: this command returns the serial no. of the device formatted as:

<consecutive number>[3].<month year>[4].<project number>[4].<device version>[3]

4.3.16 EEPROM

Command <cmd>: Ex

Parameter: <address>

Request: Ex;<address> Ex;<address>[;<Pin>]=<value>

Response: Ex;<address>=<value>

Description: this command reads / writes a value <value> to address <address> of the internal EEPROM. For areas otherwise write protected a PIN code can be supplied. ‘x‘ stands for one of the following data types:

x	Data Type
S	Short/ uint8
I	Int / uint16
L	Long / uint32
F	Float / float32
D	Double / float64

NOTE: This command is for internal use only. Inadequate setting may render the device unusable.

4.3.17 Reset

Command <cmd>: RST

Parameter: none

Request: RST RST=<time>

Response: RST=<time>

Description: this command causes a restart of the device. If a value <time> is given the restart will be executed after <time> milliseconds, otherwise immediately.

4.3.18 Reset to Factory Settings

Command <cmd>: RSD

Parameter: none

Request: RSD RSD=<time>

Response: RSD=<time>

Description: this command causes a restart of the device. If a value <time> is given the restart will be executed after <time> milliseconds, otherwise immediately. All user settings are reset to factory settings, except of the device ID.

4.3.19 Device Information

Command <cmd>: IFO

Parameter: <info>;<option>

Request: IFO;<info>;<option>

Response: IFO;<info>;option>=<value>

Description: this command returns device information as defined in following table:

<info>	<option>	Description	<value>
10	none	Device Name	<bez>^{40} z.B. ‚Visibility-Sensor VS20 ’
11	none	Device Description	<beschr>^{40} z.B. ‚Sichtweite A92 West ’
12	None	Hard- and Software Version	<hardware>^{3}; <software>^{3} Version 2.3 = 17h = 23d
13	none	extended Version Information	<Lfd.-Nr>^{3}; <MMJJ>^{4}; <Projekt>^{4}; <Stüli>^{3}; <SPlan>^{3}; <hardware>^{3}, <software>^{3}; <e2version>^{3}; <geräteversion>^{3}
14	none	EEPROM Size	<e2_size>^{5}
15	none	Number of Channels	<channels>^{5}; <blocks>^{3} max. 75 channels per block
16	<block>	Channel Number List	<channels>^{5}; [<channel>^{5}]$^{\text{<channels>}}$
20	<channel>	Channel Measurement Value	<meas_value>^{20} e.g. ‚visibility ’
21	<channel>	Channel Range	<min>^{n}; <max>^{n}
22	<channel>	Channel Unit	<unit>^{15} e.g. ‚m ’
23	<channel>	Channel Data Type	<data_type>^{2} e.g. 16h for float
24	<channel>	Channel Value Type	<mw_type>^{2} e.g. 13h for Mittelwert
30	<channel>	Complete Channel Information	<meas_value>^{20}; <unit>^{15}; <mw_type>^{2}; <data_type>^{2}; <min>^{n}; <max>^{n}

Please refer to UMB-Binary Specification for data type and value type coding

4.3.20 Temporary Protocol Change to UMB-Binary

Command <cmd>: UMB
Parameter: none
Request: UMB
Response: UMB
Description: this command switches the communication protocol immediately after reception to UMB-Binary. After 10min the protocol will automatically switch back to UMB-ASCII 2.0.

NOTE: After receiving this command will respond only to UMB-Binary protocol!

4.3.21 Channel Request

Command <cmd>: CHN
Parameter: <channelNumber>
Request: CHN;<channelNumber>
Response: CHN;<channelNumber>=<value>
Description: this command reads the measurement value of the specified channel. Length and format of <value> depends on the selected channel.

4.3.22 Standard Set Request

Command <cmd>: SS
Parameter: <setNumber>
Request: SS;<setNumber>
Response: SS;<setNumber>=<value_list>
Description: this command returns the measurement values of a Standard Set.

The definition of the Standard Sets depends on the device type. Each device type may have several Standard Sets. The <value_list> of a Standard Set may contain one or more measurement values and additional information, e.g. device type and version. The definitions of Standard Sets can be found in the device manuals.

4.3.23 Tunnel

Command <cmd>: TNL
Parameter: <BusNum>;<Len>
Request: TNL;<BusNum>;<Len>=<Data>^{Len}
Response: TNL;<BusNum>;<Len>=<Data>^{Len}
Description: this command tunnels binary data <Data> of length <Len> to a device internal communication bus. Valid values for <BusNum> are defined in the individual device manual. <BusNum> and <Len> are two hexadecimal characters each.

NOTE: When using this command, the parser of the receiver MUST evaluate the <len> parameter, as the bytes of <Data> may take all values between 0 and 255!

NOTE <Len> and the length of <Data> MAY NOT exceed 228!