Microtronics
We live M2M

User Manual for the stationary device myDatalogMUCmini



Rev. 02 valid from:

- Firmware version: 03v011
- Modem version: 03v016
- Server version: 42.2
- Hardware version: 2.0

CE

Microtronics Engineering GmbH | www.microtronics.at
Hauptstrasse 7 | 3244 Ruprechtshofen | Austria
Tel +43 2756 77180 23 | Fax +43 2756 77180 33 | support@microtronics.at

6872
6872
6872

Chapter 1 Table of contents

User Manual for the stationary device myDatalogMUCmini 1

Chapter 1 Table of contents 3

Chapter 2 Declaration of conformity 15

Chapter 3 Technical data 17

Chapter 4 General specifications 21

4.1 Translation 21

4.2 Copyright 21

4.3 General descriptive names 21

4.4 Safety instructions 21

4.4.1 Use of the hazard warnings 22

4.4.2 General safety instructions 22

4.4.3 Safety and preventative measures for handling GSM/GPRS modems 23

4.4.3.1 Safety and precautionary measures for the installation of the GSM/GPRS modem 23

4.4.3.2 Safety measures for installing the antenna 23

4.5 Overview 24

4.5.1 Block diagram 25

4.6 Intended use 25

4.7 General product information 26

4.8 Device labelling 27

4.9 Installation of replacement and wear parts 28

4.10 Storage of the product 28

4.11 Obligation of the operator 28

Chapter 5 Functional principle 29

5.1 Internal processing of the measurement values 31

5.1.1 Filter module 33

5.1.2 Overflow module 34

5.1.3 Scale module (inputs) 35

5.1.4 Decay module ... 36
5.1.5 Hold module ... 37
5.1.6 Alarm/trigger module ... 38
5.1.7 Control module ... 39
5.1.8 Record module ... 40
5.1.9 Setpoint module ... 42
5.1.10 Scale module (outputs) ... 43
5.2 Functionality of the internal data memory ... 44
5.3 Procedure in case of connection aborts ... 45
5.4 ALOHA transmission mode ... 45
5.5 Automatic selection of the GSM network ... 46
5.5.1 Ban list ... 47
5.6 Script eradication ... 48
5.6.1 Instruction list ... 48
5.6.2 Pawn ... 48
5.6.2.1 Script parsing ... 48
Chapter 6 Storage, delivery and transport ... 53
6.1 Inspection of incoming deliveries ... 53
6.2 Scope of supply ... 53
6.3 Storage ... 53
6.4 Transport ... 53
6.5 Return ... 54
Chapter 7 Installation ... 55
7.1 Dimensions ... 55
7.2 Installing the myDatalogMUCmini ... 55
7.2.1 Wall mounting ... 56
7.2.2 Top-hat rail assembly ... 57
7.2.3 Pipe assembly ... 58
7.2.4 Assembly in a control cabinet ... 58
7.3 Safety instructions for the cabling ... 59

7.3.1 Information on preventing electrostatic discharges (ESD) ... 59
7.4 Electrical installation ... 60
7.4.1 Connecting the sensors, actuators and power supply ... 60
7.4.1.1 Connection examples ... 63
7.4.1.1.1 Connection examples: Digital modes ... 63
7.4.1.1.2 Connection examples: Analogue modes ... 64
7.4.2 Connection of the GSM antenna ... 64
7.4.3 Technical details about the universal inputs ... 65
7.4.3.1 0/4 to 20mA mode ... 65
7.4.3.2 0 to 2V mode ... 65
7.4.3.3 0 to 10V mode ... 65
7.4.3.4 Standard digital modes (PWM, frequency, digital, day counter, impulse counter) ... 66
7.4.4 Technical details regarding the modbus interface (Com1) ... 66
7.4.4.1 Modbus slave mode ... 68
7.4.5 Technical details about the RS232 interface (Com2) ... 68
7.4.5.1 Error codes of the RS232 interface ... 70
7.4.6 Technical details about the outputs ... 70
7.4.6.1 mA OUT 1: Analogue output (mA OUT, GND) ... 70
7.4.6.2 Relay 1: Isolated switch contact (CC, NO) ... 70
7.4.7 Technical details about the integrated rechargeable buffer battery ... 71
7.4.8 Technical details regarding the energy supply ... 74
Chapter 8 Initial Start-Up ... 75
8.1 User information ... 75
8.2 General principles ... 75
8.3 Placing the system into operation ... 75
8.4 Testing communication with the device ... 76
Chapter 9 User interfaces ... 79
9.1 User interface on the myDatalogMUCmini ... 79
9.1.1 Operating elements ... 79

9.1.1.1 Button to initiate ALOHA transmission mode ... 79
9.1.1.2 Status LED ... 80
9.2 User interface on the myDatanet-Server ... 81
9.2.1 Site configuration ... 81
9.2.1.1 Site ... 81
9.2.1.2 Comments ... 81
9.2.1.3 Control ... 82
9.2.1.4 Interfaces ... 83
9.2.1.4.1 Basis ... 83
9.2.1.4.1.1 Com1 (modbus master, RS485, script parsing inactive) ... 83
9.2.1.4.1.2 Com1 (modbus slave, RS485, script parsing inactive) ... 86
9.2.1.4.1.3 Com1 (RS485, script parsing active) ... 87
9.2.1.4.1.4 Com2 (serial, RS232) ... 90
9.2.1.4.2 Config ... 94
9.2.1.4.2.1 Com1 ... 94
9.2.1.4.2.2 Com2 ... 95
9.2.1.5 Sequences for COM2 ... 95
9.2.1.6 Measurement channels ... 96
9.2.1.6.1 Basis ... 96
9.2.1.6.2 Config ... 99
9.2.1.6.3 Alarms ... 114
9.2.1.6.4 Trigger ... 114
9.2.1.7 Interface channels 1-32 ... 115
9.2.1.7.1 Basis ... 116
9.2.1.7.2 Config ... 117
9.2.1.7.2.1 Com1 (modbus master, RS485) ... 117
9.2.1.7.2.2 Com1 (modbus slave, RS485) ... 118
9.2.1.7.2.3 Com2 (serial, RS232) ... 120
9.2.1.7.3 Scale ... 120
9.2.1.7.3.1 Com1 (modbus, RS485) ... 120

9.2.1.7.3.2 Com2 (serial, RS232) 121
9.2.1.7.4 Alarms 121
9.2.1.7.5 Trigger 122
9.2.1.8 Interface channels 33-64 122
9.2.1.8.1 Basis 122
9.2.1.9 Interface output channels 1-32 122
9.2.1.9.1 Basis 123
9.2.1.9.2 Config 124
9.2.1.9.2.1 Com1 (modbus master, RS485) 124
9.2.1.9.2.2 Com1 (modbus slave, RS485) 126
9.2.1.9.2.3 Com2 (serial, RS232) 127
9.2.1.9.3 Scale 127
9.2.1.9.3.1 Com1 (modbus, RS485) 128
9.2.1.9.3.2 Com2 (serial, RS232) 128
9.2.1.10 Interface output channels 33-64 128
9.2.1.11 Calculated channels 129
9.2.1.11.1 Basis 129
9.2.1.11.2 Calculation 131
9.2.1.11.3 Alarms 133
9.2.1.12 Output channels 133
9.2.1.12.1 Basis 133
9.2.1.12.2 Config 138
9.2.1.13 Internal channels 139
9.2.1.13.1 Basis 139
9.2.1.13.2 Alarms 139
9.2.1.13.3 Trigger 140
9.2.1.14 Alarm settings 141
9.2.1.15 Basic settings 142
9.2.1.16 FTP export settings 144
9.2.2 Device configuration 144

9.2.2.1 Comments ... 144
9.2.2.2 Measurement instrument ... 144
9.2.2.3 Device-specific settings ... 146
9.2.2.4 GPRS ... 146
Chapter 10 myDatanet server ... 147
10.1 Overview ... 147
10.1.1 Explanation of the symbols ... 147
10.2 "Customer" area ... 148
10.3 "Site" area at customer level ... 149
10.3.1 Reports ... 150
10.3.2 Map view ... 150
10.4 Recommended procedure ... 150
10.4.1 Creating the site ... 150
10.4.2 Assigning a device to the customer ... 151
Chapter 11 Instruction list ... 153
11.1 General ... 153
11.1.1 Direct entry of an instruction list ... 153
11.1.2 Using a script template ... 153
11.2 Implementation details ... 154
11.2.1 Common elements ... 154
11.2.2 IL syntax elements ... 156
11.3 Instruction list error codes ... 158
11.4 Expansions ... 160
11.5 Device-specific implementation ... 162
11.6 Example ... 164
11.6.1 Saw-tooth generator ... 164
11.6.2 Weekday designation ... 164
Chapter 12 Pawn script ... 167
12.1 General ... 167
12.1.1 Direct input of a pawn script ... 167

12.1.2 Uploading a binary file ... 167
12.2 rapidM2M API ... 168
12.2.1 Core functions ... 168
12.2.2 Console functions ... 171
12.2.3 Mdn functions ... 172
12.2.3.1 Arrays with symbolic indices ... 172
12.2.3.2 Constants ... 173
12.2.3.3 Functions ... 176
12.2.4 Mathematical functions ... 191
12.2.5 String functions ... 193
12.2.6 Help functions ... 198
12.3 Device-dependent channel numbers ... 199
12.4 Device-specific interface numbers ... 199
12.5 Predefined log entries ... 200
12.6 Pawn script error codes ... 200
12.7 Syntax ... 202
12.7.1 General syntax ... 202
12.7.1.1 Format ... 202
12.7.1.2 Optional semicolons ... 202
12.7.1.3 Comments ... 202
12.7.1.4 Identifier ... 202
12.7.1.5 Reserved keywords ... 203
12.7.1.6 Numerical constants ... 203
12.7.1.6.1 Numerical integer constants ... 203
12.7.1.6.2 Numerical floating-point constants ... 203
12.7.2 Variables ... 204
12.7.2.1 Declaration ... 204
12.7.2.2 Local declaration ... 204
12.7.2.3 Global declaration ... 204
12.7.2.4 Static local declaration ... 204

12.7.2.5 Static global declaration 204
12.7.2.6 Floating point values 205
12.7.3 Constant variables 205
12.7.4 Array variables 205
12.7.4.1 One-dimensional arrays 205
12.7.4.2 Initialisation 205
12.7.4.3 Progressive initialisation for arrays 206
12.7.4.4 Multi-dimensional arrays 206
12.7.4.5 Arrays and the "sizeof" operator 207
12.7.5 Operators and expressions 207
12.7.5.1 Notational conventions 207
12.7.5.2 Expressions 208
12.7.5.3 Arithmetic 208
12.7.5.4 Bit manipulation 209
12.7.5.5 Assignment 209
12.7.5.6 Comparative operators 210
12.7.5.7 Boolean 210
12.7.5.8 Other 211
12.7.5.9 Priority of the operators 211
12.7.6 Statements 212
12.7.6.1 Statement label 213
12.7.6.2 Composite statements 213
12.7.6.3 Expression statement 213
12.7.6.4 Empty statement 213
12.7.6.5 Assert expression 213
12.7.6.6 Break 214
12.7.6.7 Continue 214
12.7.6.8 Do statement while (expression) 215
12.7.6.9 Exit expression 215
12.7.6.10 For (expression 1; expression 2; expression 3) statement 215

12.7.6.11 Goto label ... 216
12.7.6.12 If (expression) statement 1 else statement 2 ... 216
12.7.6.13 Return expression ... 217
12.7.6.14 switch (expression) {case list} ... 217
12.7.6.15 While (expression) statement ... 218
12.7.7 Functions ... 218
12.7.7.1 Function arguments ("call-by-value" versus "call-by-reference") ... 219
12.7.7.2 Named parameters versus fixed parameters ... 221
12.7.7.3 Standard values of function arguments ... 222
12.7.8 Example ... 223
12.7.8.1 Saw-tooth generator ... 223
12.7.8.2 Weekday designation ... 224
12.7.8.3 Calculating the flow rate with the table ... 226
12.7.8.4 Script parsing ... 228
12.7.9 Differences to C ... 230
Chapter 13 XML interface ... 233
13.1 General ... 233
13.2 Structure of the database ... 234
13.2.1 Generally applicable table identifiers ... 236
13.2.1.1 mydatalogmucminiid ... 236
13.2.1.2 geraetid ... 245
13.2.1.3 bookingline (read only) ... 246
13.2.1.4 kundeid ... 247
13.2.2 Device-specific table identifier ... 249
13.2.2.1 mydatalogmucmini_data ... 249
13.2.2.2 mydatalogmucmini_channel ... 250
13.2.2.3 mydatalogmucmini_output ... 255
13.2.2.4 mydatalogmucmini_log ... 258
13.2.2.5 mydatalogmucmini_user1 ... 258
13.2.2.6 mydatalogmucmini_user2 ... 260

13.2.2.7 mydatalogmucmini_state ... 260
13.2.2.8 mydatalogmucmini_alarm ... 261
13.2.2.9 mydatalogmucmini_compcfg ... 262
13.2.2.10 mydatalogmucmini_aloha ... 266
13.2.2.11 mydatalogmucmini_pos ... 267
13.2.2.12 mydatalogmucmini_ctrlscript ... 268
13.2.2.13 mydatalogmucmini_cfg ... 268
13.2.2.14 mydatalogmucmini_scale ... 275
13.2.2.15 mydatalogmucmini_initseq ... 277
13.2.2.16 mydatalogmucmini_measseq ... 277
13.3 XML export ... 277
13.3.1 Configuration of a HTTP-GET command ... 277
13.3.1.1 Syntax of the xPath query ... 278
13.3.1.1.1 List of the fields in a table ... 278
13.3.1.1.2 List of site types ... 279
13.3.1.1.3 Filter conditions ... 279
13.3.1.1.3.1 Example: ... 280
13.3.1.1.4 Fields ... 281
13.3.1.1.4.1 Example ... 281
13.3.1.1.5 Data retrieval including index specifications ... 281
13.3.1.1.5.1 Example ... 282
13.3.1.2 Data record compression ... 283
13.3.1.3 Updating data fields ... 285
13.3.1.3.1 Example ... 285
13.3.2 Output formats of the HTTP-GET command ... 286
13.3.2.1 Format of the XML result ... 287
13.3.2.1.1 Example: ... 287
13.3.2.2 Format of the JSON result ... 287
13.3.2.2.1 Example ... 288
13.4 XML import ... 289

13.4.1 Transferring data to the myDatanet server ... 289
13.4.1.1 Configuration of a HTTP-POST command ... 291
13.4.1.2 Structure of the XML file ... 292
13.4.1.2.1 Changing existing data ... 292
13.4.1.2.2 Importing data ... 292
Chapter 14 Maintenance ... 295
14.1 General maintenance ... 295
14.2 Fuse replacement ... 295
Chapter 15 Removal/disposal ... 297
Chapter 16 Troubleshooting and repair ... 299
16.1 General problems ... 299
16.2 Log entries and error codes ... 302
16.2.1 Modem error ... 307
16.3 Evaluating the device log ... 310
16.3.1 Evaluating the device log on the myDatanet server ... 310
Chapter 17 Spare parts and accessories ... 311
17.1 Assembly sets ... 311
17.2 Antennas ... 311
17.3 Power supply ... 311
17.4 Adapter ... 311
17.5 Other accessories ... 311
Chapter 18 Document history ... 313
Chapter 19 Glossary ... 317
Chapter 20 Contact information ... 319

6872
6872
6872

Chapter 2 Declaration of conformity

CE

Konformitätserklärung
Declaration of Conformity
Déclaration de conformité

Microtronics at

Microtronics Engineering GmbH
Hauptstrasse 7
A-3244 Ruprechtshofen
www.microtronics.at

erklärt in alleiniger Verantwortung, dass das Produkt /
declare under our sole responsibility that our product /
déclare sous sa seule responsabilité que le produit

myDatalogMUCmini

auf das sich diese Erklärung bezieht, mit den folgenden Normen übereinstimmt /
to which this declaration relates is in comformity with the following standards /
auquel se réfère cette déclaration est conforme aux normes

ÖVE/ÖNORM EN61326-1:2006-12-01
ÖVE/ÖNORM EN61010-1:2011-08-01
ETSI EN301489-1 V1.9.2:2011-09; ETSI EN301489-7 V1.3.1:2005-11;
ETSI EN301511 V9.0.2 :2003-03

gemäß den Bestimmungen der Richtlinien /
following the provision of Directives /
conformément aux dispositions des directives

2004/108/EG EMC Directive
2006/95/EG LVD Directive
1999/5/EG RTTE Directive

Ruprechtshofen, 16.04.2013

Buber Hans-Peter, Managing Director

6872
6872
6872

Chapter 3 Technical data

Voltage supply	12...30VDC (+/- 10%) Additional information is provided in "Technical details regarding the energy supply" on page 74.
Power consumption	typ. 1W (without sensors) max. 3W (without sensors)
Integrated rechargeable buffer battery	LiPo rechargeable battery with 240mAh to issue a message in the event of a power supply failure. Additional information is provided in "Technical details about the integrated rechargeable buffer battery" on page 71.
Housing	Material: PC/Noryl, UL94-Vo Weight: 160g Protection class: IP20 Dimensions (WHD): 104 x 85 x 32mm (without antenna)
Operating temperature	-20...+60°C, 15...90%rH non-condensing
Storage temperature	-20...+60°C
Charging temperature (rechargeable buffer battery)	0...+40°C
Display	LED for indicating the operating state and error codes
Operation	Button to initiate ALOHA transmission mode
Antenna connector	SMA
Universal inputs	3 x analogue or digital Modes: • 0...20mA: Resolution 6,3µA, max. 23,7mA, load 96Ω • 4...20mA: Resolution 6,3µA, max. 23,7mA, load 96Ω • 0...2V: Resolution 610µV, max. 2,5V, load 10k086 • 0...10V: Resolution 7,97mV, max. 32V, load 4k7 • PWM: 1...99%, max. 100Hz, min. pulse length 1ms, load 4k7 • Frequency: 1...1000Hz, 4k7 • Digital: max. 32V, low ‹1,31V, high ›2,61V, load 4k7 • Day counter: min. pulse length 1ms, load 4k7 • Interval counter: min. pulse length 1ms, load 4k7 Additional information is provided in "Technical details about the universal inputs" on page 65.

Modbus	1 x RS485 switchable between modbus master and modbus slave • Baud rate: 300-115200 • Stop bits: 1, 2 • Parity: N, E, O • Data bits: 7, 8 Modes: • RTU • ASCII Number of processable interface channels • Inputs: modbus channels + serial channels = 64 • Outputs: modbus channels + serial channels = 64 Additional information is provided in "Technical details regarding the modbus interface (Com1)" on page 66.
Serial interface	1 x RS232 for the connection of a digital sensor • Baud rate: 300-115200 • Stop bits: 1, 2 • Parity: N, E, O • Data bits: 7, 8 Modes: • ASCII Number of processable interface channels • Inputs: modbus channels + serial channels = 64 • Outputs: modbus channels + serial channels = 64 Additional information is provided in "Technical details about the RS232 interface (Com2)" on page 68.

Outputs	1 x isolated switch contact • Galvanically isolated • Parallel connection of an optoswitch and relay Relay (modes: ext. warmup time, digital) • U: 250V AC, 30V DC • I: 3A Optoswitch (modes: frequency, PWM, pulse) • U_{max}: 350V peak AC and DC • $I_{max\ cont.}$: 130mA • P_{max}: 500mW • R_{on}: 18...25Ω • f_{max}: 500Hz 1 x analogue output (not galvanically isolated) • Load voltage: Corresponds to the supply voltage (12...30VDC , +/- 10%) • max. load (32V, 20mA): 1200Ω • $I_{out\ max}$: 20mA Modes: • 4-20mA • 0-20mA Additional information is provided in "Technical details about the outputs" on page 70.
Data memory	Internal flash memory for up to 3.128 measurement cycles Additional information is provided in "Functionality of the internal data memory" on page 44.
Data type	f32 (32Bit floating point) Exception: Modbus register (see "format" configuration parameters in the chapter "Interface channels 1-32" on page 115)
Data transmission	By means of GSM/GPRS quad-band modem to the respective myDatanet server
SIM	The myDatalogMUCmini is equipped with an integrated SIM chip.
Monthly data volume	d.b.d. at 2 min. measurement cycle and 120 min. transmission cycle

6872
6872
6872

Chapter 4 General specifications

The information in this manual has been compiled with great care and to the best of our knowledge. The manufacturer, however, assumes no liability for any incorrect specifications that may be provided in this manual. The manufacturer is not responsible for direct, indirect, accidental or consequential damages which arise from errors or omissions in this manual even if advised of the possibility of such damages. In the interest of continuous product development, the manufacturer reserves the right to make improvements to this manual and the products described in it at any time and without prior notification or obligation.

> ***Note:*** *The specifications in this manual are valid as of the versions listed on the front page. Revised versions of this manual, as well as software and driver updates are available in the service area of the myDatanet server.*

4.1 Translation

For deliveries to countries in the European Economic Area, the manual must be translated into the language of the respective country. If there are any discrepancies in the translated text, the original manual (German) must be referenced or the manufacturer contacted for clarification.

4.2 Copyright

The copying and distribution of this document as well as the utilisation and communication of its contents to others without express authorisation is prohibited. Contraventions are liable to compensation. All rights reserved.

4.3 General descriptive names

The use of general descriptive names, trade names, trademarks and the like in this manual does not entitle the reader to assume they may be used freely by everyone. They are often protected registered trademarks even if not marked as such.

4.4 Safety instructions

For the connection, commissioning and operation of the myDatalogMUCmini , the following information and higher legal regulations of the country (e.g. ÖVE), such as valid EX regulations as well as the applicable safety and accident prevention regulations for the respective application case must be observed.

Please read this manual completely before unpacking, setting up or operating this device. Observe all hazard, danger and warning information. Non-observance can lead to serious injuries to the operator and/or damage to the device.

Ensure that the safety equipment of this measurement device is not impaired. Install and use the measuring system only in the manner and method described in this manual.

Important note: *The product is not approved for use outdoors as it is not protected from penetrating moisture and only provides minimal protection against the ingression of dust.*

4.4.1 Use of the hazard warnings



DANGER:
Indicates a potential or threatening hazardous situation that will result in death or serious injuries if not avoided.



WARNING:
Indicates a potential or threatening hazardous situation that can result in death or serious injuries if not avoided.



CAUTION:
Indicates a potential hazardous situation that can result in minor or moderate injury.

Note: *Indicates a situation that does not result in any injury to persons.*

Important note: *Indicates a situation that can result in damages to this instrument if it is not avoided. Information that must be particularly emphasised.*

Note: *Information that supplements the specifications in the main text.*

4.4.2 General safety instructions



DANGER:
Always switch off all of the used power supplies for the device before installing it, completing any maintenance work or resolving any faults.



DANGER:
Ensure that the device is fully deactivated and cannot activate automatically when sending/returning it as air freight. Information on this is provided in chapter "Storage of the product" on page 28.If you have any unanswered questions, contact the manufacturer (see "Contact information" on page 319).



WARNING:
Never use this device in areas where the use of wireless equipment is prohibited. This device must not be used in hospitals and/or in the vicinity of medical equipment, such as heart pacemakers or hearing aids, as their functionality could be compromised by the GSM/GPRS modem contained in the device.



WARNING:
Never use this device in potentially explosive atmospheres and in the vicinity of highly combustible areas (fuel stations, storage areas for combustible material, chemical plants and detonation sites) or in the vicinity of flammable gases, vapours or dust.

4.4.3 Safety and preventative measures for handling GSM/GPRS modems

The following safety and preventative measures must be observed during all phases of installation, operation, maintenance or repair of a GSM/GPRS modem . The manufacturer is not liable if the customer disregards these preventative measures.



CAUTION:
The GSM/GPRS modem connection must not be used in hazardous environments.

No guarantee of any kind, whether implicit or explicit, is given by the manufacturer and its suppliers for the use with high risk activities.

In addition to the following safety considerations, all directives of the country in which the device is installed must be complied with.

> *Important note: No liability shall be assumed at any time and under no circumstances for connections via a GSM/GPRS modem for which wireless signals and networks are utilized. The GSM/GPRS modem must be switched on and operated in an area where sufficient signal strength is present.*

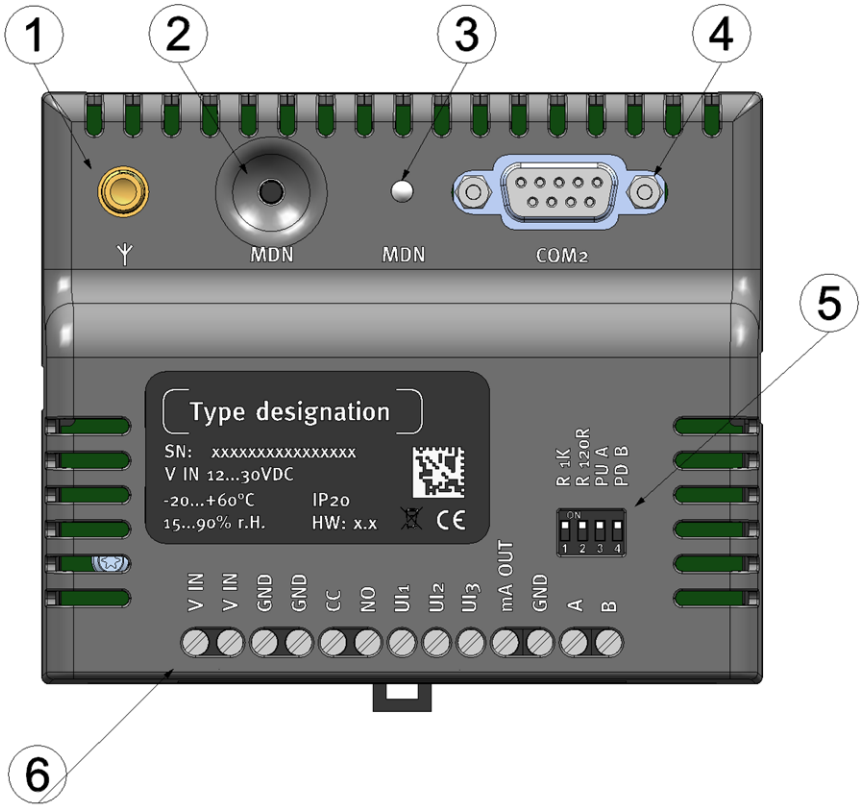
4.4.3.1 Safety and precautionary measures for the installation of the GSM/GPRS modem

- This device must only be installed by a trained technician who applies the recognised installation practices for a radio frequency transmitter including the correct grounding of external antennas.
- The device must not be operated in hospitals and/or in the vicinity of medical equipment such as heart pacemakers or hearing aids.
- The device must not be operated in highly flammable areas such as petrol filling stations, fuel storage sites, chemical factories and explosion sites.
- The device must not be operated in the vicinity of flammable gases, vapours or dusts.
- The device must not be subjected to strong vibrations or impacts.
- The GSM/GPRS modem can cause interferences if it is located in the vicinity of television sets, radios or computers.
- Do not open the GSM/GPRS modem. Any modification to the equipment is prohibited and will result in the operating licence being revoked.
- The use of GSM services (SMS messages/data communication/GPRS, etc.) may incur additional costs. The user is solely responsible for any resulting damages and costs.
- Do not install the device in any other way to the one described in the operating instructions. Improper use will invalidate the warranty.

4.4.3.2 Safety measures for installing the antenna

- Only use antennas that are recommended or supplied by the manufacturer.
- The antenna must be installed at a distance of at least 20 cm from individuals.
- The antenna must not be extended outside protected buildings and must be protected against lightning strikes.
- The voltage supply must be switched off before replacing the antenna.

4.5 Overview

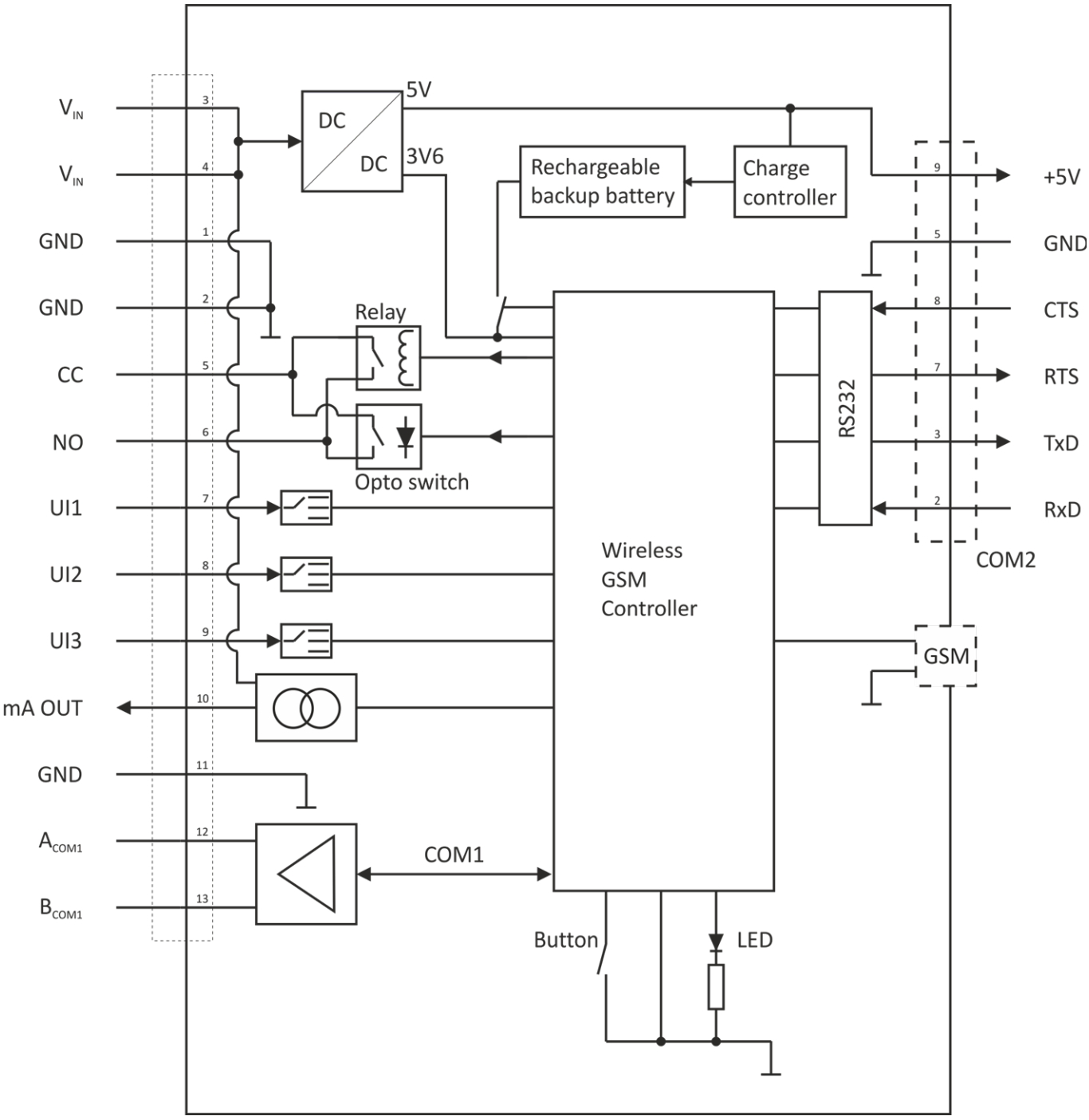


Overview myDatalogMUCmini

1 Antenna connector	4 Sub-D connector of the serial interface (Com2)
2 Button to initiate ALOHA transmission mode	5 Dip switch (dip switch 1) for activating/deactivating the load or clamp resistances for Com1 (RS485)
3 Signalling LED	6 Connection strip

4.5.1 Block diagram

Important note: *The current firmware does not support the hardware handeshake of the RS232 interface (CTS, RTS).*



Block diagram of the myDatalogMUCmini

4.6 Intended use

The stationary measurement device provides a universal solution for a variety of signal recording and control tasks. The device requires a continuous power supply. The measured and recorded data

is stored on a non-volatile memory medium. This stored data is sent via the mobile network to a central server for further processing. The device is equipped with an integrated SIM chip for this purpose. The maximum permissible limit values specified in chapter "Technical data" on page 17 must be observed. The manufacturer shall not be liable for any operational cases that deviate from these limit values and have not been approved by the manufacturer in writing.

> ***Note:*** *This device is exclusively intended to be used for the purposes as described above. Any other use or use beyond what is specified or a modification of the device shall be deemed to be not for the intended purpose and is not permitted without the express written consent of the manufacturer. The manufacturer shall not be held liable for any damages that may result from such unauthorised use or modification. The operator alone bears the associated risk.*
>
> ***Note:*** *The manufacturer is not liable for data loss of any kind if the device is damaged and data is not able to be stored correctly.*
>
> ***Note:*** *The integrated SIM chip provides a mobile communications connection to a variety of international service providers. In order to be able to utilise all functions of the device, you must ensure that the device is located in the service area of one of these service providers. You can find a list of all supported countries and associated service providers under www.microtronics.at/footprint. A Managed Service contract with Microtronics Engineering GmbH is required for use of the mobile data transmission (see www.microtronics.at/managedservice). This includes the provisioning of the mobile communications connection via the network of the service provider included in the above-mentioned list.*

4.7 General product information

The device is a stationary device that can be used for a variety of control and regulating tasks in addition to recording and transmitting measurement data from various sources.

The following interfaces are available for recording measurement data:

- 3 universal inputs that can be operated in various analogue and digital modes
- A RS485 interfaces that can be configured as a modbus master or modbus slave
- A RS232 interface to which a digital sensor can be connected that provides its data in ASCII format separated in blocks by means of a delimiter

It should be noted that there are only 64 channels available for recording the measurement data read from the 2 serial interfaces (1 x RS485, 1 x RS232). These 2 serial interfaces can also be used to return data to the sensors and actuators. A total of 64 channels are provided for this purpose. The output data can either be specified via the input screen of the myDatanet server or can be calculated by the myDatalogMUCmini after each measurement cycle via the script (see "Instruction list" on page 153 or "Pawn script" on page 167).

In addition to a 0/4-20 mA analogue output, an isolated switch contact is also available for the output of regulating and control commands. The isolated switch contact is the parallel connection of a relay (high switching current, modes: ext. warmup time, digital) and an optoswitch (high

switching frequencies, modes: frequency, PWM, pulse). The output value of the analogue output can either be assigned wirelessly from a central location or calculated by the myDatalogMUCmini by means of a script (see "Instruction list" on page 153 or "Pawn script" on page 167). The isolated switch contact can be configured so that it can be switched by the device itself prior to a measurement (to supply a sensor) or so that the setpoint can be specified wirelessly from a central location. The myDatalogMUCmini can also determine the setpoint for the isolated switch contact itself by means of the script.

All of the input measurement data is temporarily saved to an internal data memory along with the output states and wirelessly transferred to a central location at a freely selected interval. The device is also configured via this connection. The device is equipped with an integrated SIM chip.

4.8 Device labelling

The specifications in this user manual apply exclusively to the myDatalogMUCmini device type. The type plate is located on the top of the device and contains the following specifications:

- Type designation
- Serial number
- Voltage supply specifications
- Environmental conditions during operation
- Country list profile of the SIM chip
- Protection class
- Hardware revision
- CE marking

The correct specification of the type designation and serial number is important for all queries and spare part orders. Only then can we process requests promptly and properly.



Type plate myDatalogMUCmini



Note: *This symbol indicates the country list profile (see www.microtronics.at/footprint) of the SIM chip installed in the device.*

Note: *These operating instructions are part of the device and must be available to the user at all times. The safety instructions contained therein must be observed.*



DANGER:
It is strictly prohibited to disable the safety equipment or modify its mode of operation.

4.9 Installation of replacement and wear parts

Please be advised that replacement and accessory parts that have not been supplied by the manufacturer have also not been inspected or approved by it. Installing or using such products and similar can constructively modify the specified properties of the device in a negative way. The manufacturer shall not be liable for any damages that arise from the use of non-original parts and non-original accessory parts.

4.10 Storage of the product

To safeguard the myDatalogMUCmini , ensure that all of the relevant data was transmitted to the myDatanet server. If necessary initiate the ALOHA transmission mode using the button (see "ALOHA transmission mode" on page 45) and then check again that all of the relevant data has been transmitted. Then disconnect the device from the supply voltage. If possible, switch off the supply voltage before disconnecting the cables from the V_{IN} and GND clamps (see "Connecting the sensors, actuators and power supply" on page 60). The remaining cables and the antenna can then be removed. Store the myDatalogMUCmini in its original packaging.

The configuration and most recently determined data are retained. However, the system clock will not continue to work without an energy supply. This means that following recommissioning, the time must be synchronised to ensure that the measurement and log data timestamps are correct. This is completed automatically during the first connection to the myDatanet server.

4.11 Obligation of the operator



WARNING:
In the EEA (European Economic Area), the national implementation of the framework directive (89/391/EEC) as well as the associated specific directives and from these in particular, the directive (89/655/EEC) about the minimum safety and health requirements for use of work equipment by workers at work, each in their respective version are to be complied with.

The operator must obtain the local operating licence and the associated documents.

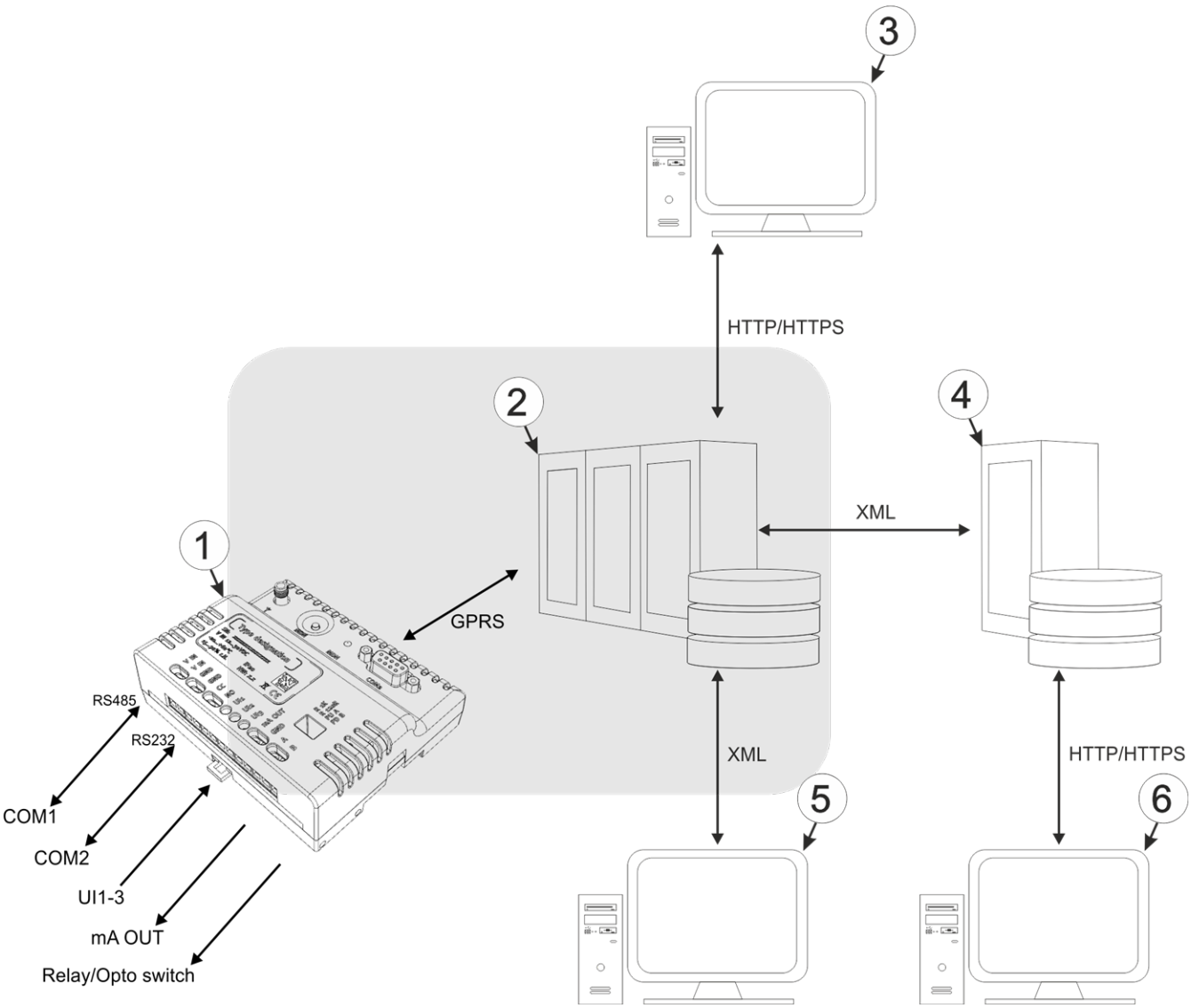
In addition, the operator must comply with the local legal requirements for

- the safety of the personnel (accident prevention measures)
- the safety of the equipment (protective equipment and maintenance)
- the product disposal (waste disposal law)
- the material disposal (waste disposal law)
- the cleaning (cleaning agents and disposal)
- and the environmental protection amendments.

Before commissioning, the operator must ensure that the installation and commissioning – provided these were performed by the operator himself – are in compliance with the local regulations.

Chapter 5 Functional principle

In the graphic below, all of the components that are part of the myDatanet are illustrated in grey. All of the other components must be provided/created by the customer.



Functional principle

1	myDatalogMUCmini with integrated managed service SIM chip (including data transmission)
2	myDatanet server to which the data is transferred
3	Client that accesses the interface of the myDatanet server via the web browser
4	Customer-specific server that provides clients with their own interface. The customer-specific server obtains the data via the XML interface of the myDatanet server (see "XML interface" on page 233).
5	Client, on which a PC program is running, that obtains its data via the XML interface of the myDatanet server (see "XML interface" on page 233)
6	Client that accesses the interface of the customer-specific server via the web browser

Functions and components provided by myDatanet :

- myDatalogMUCmini

  The myDatalogMUCmini is a stationary device for connecting sensors (UI1-3, COM1-2) and actuators (relay/optoswitch, mA OUT, COM1-2) to the myDatanet server (GPRS).

- Managed Service

  Managed Service is the basis for operating your devices and provides you with a wide range of services. Managed Service includes updates for device firmware, mobile data transmission on a global scale and free support - providing you with one contact person for the entire solution.

- myDatanet server

  Database for saving the measurement data and configurations. Data is either accessed via the XML interface of the server (see "XML interface" on page 233) or web interface of the server.

Functions and components provided by the customer:

- Sensor and actuators

  Sensor and actuators that include interfaces that are compatible with the specifications listed in the chapter "Specifications" (see "Technical data" on page 17).

- Customer-specific server with web interface for the clients (optional)

  It is therefore possible to create an individual web interface for the clients. Using this method, the data is read out of the myDatanet server via the XML interface (see "XML interface" on page 233) by the customer-specific server.

5.1 Internal processing of the measurement values

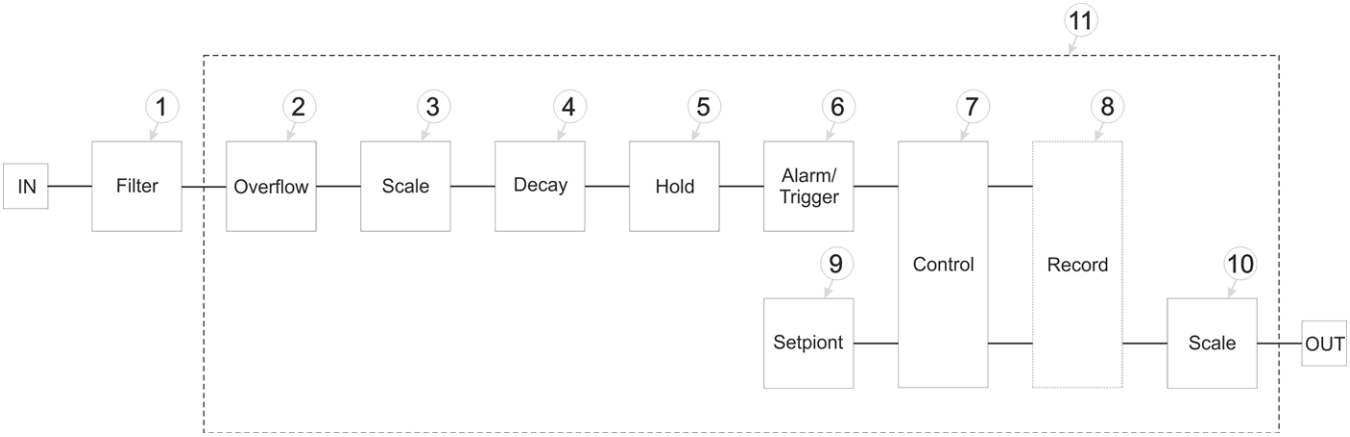


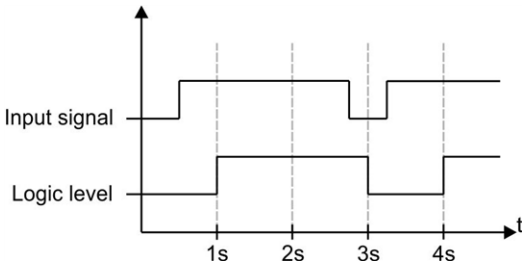
Diagram of the internal processing of the measurement values

1 Filter to compensate for brief signal fluctuations (see "Filter module" on page 33). The filter module is operated permanently.	**7** Determination of the setpoints, script processing (see "Control module" on page 39)
2 Monitoring of the measurement range limits (see "Overflow module" on page 34)	**8** As the record interval and measurement cycle can be selected individually, the measurement values and setpoints are not recorded at the time of every measurement (see "Record module" on page 40).
3 Rescaling from the raw value to the measurement value (see "Scale module (inputs)" on page 35)	**9** Specification of the setpoints for the outputs (see "Setpoint module" on page 42)
4 Decay module to summarise several measurement values (see "Decay module" on page 36)	**10** Rescaling from the setpoint to the physical size for the output (see "Scale module (outputs)" on page 43)
5 Module to maintain the last valid value in the event of invalid measurement values (see "Hold module" on page 37)	**11** This module chain is started at the time of every measurement and is completed once. For universal inputs that are operated in Digital mode, this module chain (with exception of the control module) is also executed at second intervals to be able to react to level changes as quickly as possible.
6 Monitoring of the alarm limits and trigger levels (see "Alarm/trigger module" on page 38)	

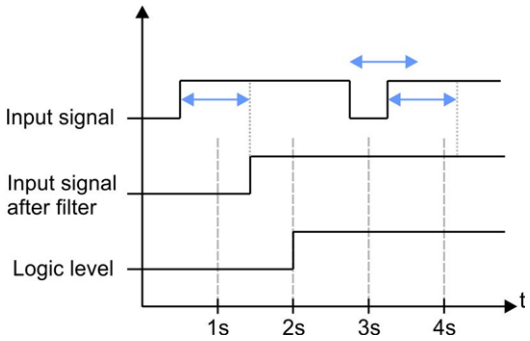
Note: *Additional explanation on universal inputs that are operated in Digital mode.*

Measurement channels -> Basis		*Mode*	*Digital*
Measurement channels -> Config.	⟷	*Filter time*	*800ms*

Input signal	*Input signal at the universal input*
Input signal after filter	*Input signal taking the "filter time" into consideration*
Logic level	*The input signal at the universal input is analysed once per second.*



Filter module to compensate for brief signal fluctuations disabled ("filter time" = 0)



Filter module to compensate for brief signal fluctuations active ("filter time" not equal to 0)

Explanation: *To react to a level change at the universal inputs as quickly as possible, the module chain detailed in "Diagram of the internal processing of the measurement values" on page 31 (with exception of the control module) for the relevant universal input is operated once per second in Digital operating mode. This ensures that alarms and triggers are detected asynchronously to the measurement cycle. The signal must be present for at least 1 second to definitely detect a pulse safely. Additionally, any brief faults can be eliminated with the help of the "filter time".*

5.1.1 Filter module

The filter module is designed to compensate for brief fluctuations to the input signal. This module is only available for the 3 universal inputs.

The following table specifies the relevant parameters for the module:

Configuration section	Mode	Parameter	Explanation
Measurement channels -›Config.	Digital	Filter time	Time during which the signal must remain constant to initiate a level change. Used to suppress brief faults (debouncing).
	Cnt.Day		
	Cnt.Intervl.		
	Freq		
	PWM		
	4-20 mA	Filter time	Time during which the analogue signal is averaged for signal smoothing. Used to suppress signal noise (also see "Output channels" on page 133).
	0-20 mA		
	0-2V		
	0-10V		

5.1.2 Overflow module

This module monitors the measurement range limits of the raw value. If a universal input was, for example, switched to "4-20 mA" mode, a raw value of 2 mA will violate the measurement range. The overflow module is only available for the 3 universal inputs in channel modes "Freq", "PWM", "4-20 mA", "0-20 mA", "0-2V" and "0-10V" as well as for the interface channels (Com1 only) with activated scaling.

The following table specifies the relevant parameters for the module:

Configuration section	Modus/interface	Parameter	Explanation
Measurement channels -›Config.	Digital	---	---
	Cnt.Day		
	Cnt.Intervl.		
	Freq	Overflow	Procedure in the event of measurement range violations
	PWM		
	4-20 mA		
	0-20 mA		
	0-2V		
	0-10V		
Interface channels 1-64 -› Scaling	Com1	Overflow	Procedure in the event of measurement range violations

5.1.3 Scale module (inputs)

This module rescales the raw value (e.g. mA) to the desired measurement value (e.g. mm). The scale module is only available for the 3 universal inputs and for the interface channels (Com1 only).

The following table specifies the relevant parameters for the module:

Configuration section	Modus/interface	Parameter	Explanation
Measurement channels -› Basis	Digital	Invert	Inverts the input signal
	Cnt.Day	Pulse	Metered measurand of a pulse in the measurement unit
	Cnt.Intervl.		
	Freq	Factor	Factor by which the input signal is multiplied
	PWM 4-20 mA 0-20 mA 0-2 V 0-10 V	0%	Start of the measurement range in the measurement unit
		100%	End of the measurement range in the measurement unit
Interface channels 1-64 -› Scaling	Com1	0%	Start of the measurement range in the measurement unit
		100%	End of the measurement range in the measurement unit
		0% modbus	Start of the measurement range in the unit of the modbus slave
		100% modbus	End of the measurement range in the unit of the modbus slave

5.1.4 Decay module

The decay module is designed to summarise several measurement values. The average value over a required time frame or the minimum value within a required time frame can be determined, for example. The decay module is only available for the 3 universal inputs, with the exception of "Cnt.Day" channel mode.

The following table specifies the relevant parameters for the module:

Configuration section	Mode	Parameter	Explanation
Measurement channels - ›Config.	Digital	Decay	Temporal function in the measurement cycle
		Time	Time x, that is used for decay modes "up", "down" and "up & down"
	Cnt.Day	---	---
	Cnt.Intervl. Freq PWM 4-20 mA 0-20 mA 0-2 V 0-10 V	Decay	Temporal function in the measurement cycle
		Time	Decay period. To calculate the number of considered measurement values see "Example to clarify the record, measurement and burst interval (in conjunction with the decay)" on page 143. The "DECAY MEM ERR" error is entered in the device log if no temporary memory could be reserved to take another measurement value into consideration (see "Log entries and error codes " on page 302).

5.1.5 Hold module

With the help of the hold module, it can be determined how to proceed in the event of invalid measurement values. It ensures the last valid measurement value is retained until a new valid measurement value is available or passes the error on after a certain number of invalid measurements. The hold module is only available for the universal inputs in channel modes "Freq", "PWM", "4-20 mA", "0-20 mA", "0-2V" and "0-10V" as well as the interface channels.

The following table specifies the relevant parameters for the module:

Configuration section	Modus/interface	Parameter	Explanation
Interfaces -› Basis	Com1	Hold	Hold the last valid measurement value for x measurement cycles
	Com2		
Measurement channels -›Config.	Digital	---	---
	Cnt.Day		
	Cnt.Intervl.		
	Freq	Hold	Hold the last valid measurement value for x measurement cycles
	PWM		
	4-20 mA		
	0-20 mA		
	0-2V		
	0-10V		

5.1.6 Alarm/trigger module

The alarm limits and trigger levels are monitored by this module. An entry is created in the alarm list if required. The global triggers are also set (see "Alarm flags" or "Trigger flags" in chapter "Constants" on page 173). The alarm/trigger module is only available for inputs (universal inputs, interface channels, internal channels).

All of the parameters of the "Alarms" and "Trigger" tabs of the "Measurement channels" (see "Measurement channels" on page 96), "Interface channels 1-32" (see "Alarms" on page 121 and "Trigger" on page 122) and "Internal channels" (see "Internal channels" on page 139) configuration sections are relevant to this module. Additionally, the parameters in the following tables are also important:

Configuration section	Parameter		Explanation
Alarm	On alarm	A	The alarm is recorded in the alarm list.
	On warning	A	The warning is recorded in the alarm list.
	On fault alarm	A	The fault alarm is recorded in the alarm list.
	On fault warning	A	The fault warning is recorded in the alarm list.

5.1.7 Control module

The control module completes the script processing and determines the setpoints for the outputs. When determining the setpoints for the outputs, it must be taken into account that a value calculated by means of a script overwrites the value determined via the input screen on the myDatanet server.The values of a measurement channel can also still be changed by the script before it is recorded by the record module. If the "Pawn" script type was selected, the alarm status of a channel can be changed via the "Mdn_SetAlarm()" function (see Mdn_SetAlarm()). An entry created by the alarm/trigger module in the alarm list cannot be changed during this process, only an additional entry for the relevant channel can be created.

The control module execute the complete script if the "IL" (Instruction list) script type was selected. The "Mdn_CtrlFinish()" function (see Mdn_CtrlFinish()) is called up by the control module if the "Pawn" script type was selected.

In addition to the parameters in the following table, all of the parameters of the "Control" configuration section (see "Control" on page 82) are also relevant:

Configuration section	Modus/interface	Parameter	Explanation
Interface output channels 1-64 -› Basis	Off	---	---
	Com1		
	Com2		
Output channels	Off	---	---
	0-20 mA		
	4-20mA		
	Ext warmup time	Ext. warmup time	The output channel is switched on "ext warmup time" seconds prior to the measurement.
	Digital	---	---
	Freq		
	PWM		
	Pulse		

5.1.8 Record module

The record module records the measurement values. As the record interval and measurement cycle can be selected individually, a record is not saved at the time of every measurement. Depending on the global triggers set by the alarm/trigger module (see "Alarm flags" or "Trigger flags" in chapter "Constants" on page 173), the record interval is modified, if necessary, the transmission is initiated or a new measurement is triggered.

The following table specifies the relevant parameters for the module:

Configuration section	Modus/interface	Parameter	Explanation
Measurement channels -> Trigger	Digital	QU	Fastrecording (record interval = record interval / factor)
		SL	Slow recording (record interval = record interval * factor)
		MS	Start measurement cycle immediately
		XM	Initiate transmission
		ON	Activate online mode
	Cnt.Day Cnt.Intervl. Freq PWM 4-20mA 0-20mA 0-2V 0-10V	QU	Fastrecording (record interval = record interval / factor)
		SL	Slow recording (record interval = record interval * factor)
		RO	Switch on recording
		RF	Switch off recording
		XM	Initiate transmission
		ON	Activate online mode
Interface channels 1-64 -> Trigger	Com1 Com2	QU	Fastrecording (record interval = record interval / factor)
		SL	Slow recording (record interval = record interval * factor)
		RO	Switch on recording
		RF	Switch off recording
		XM	Initiate transmission
		ON	Activate online mode

Configuration section	Modus/interface	Parameter	Explanation
Internal channels -› Trigger	---	QU	Fastrecording (record interval = record interval / factor)
		SL	Slow recording (record interval = record interval * factor)
		RO	Switch on recording
		RF	Switch off recording
		XM	Initiate transmission
		ON	Activate online mode

Configuration section	Parameter		Explanation
Alarm	On alarm On warning On fault alarm On fault warning	Ü	An immediate transmission is initiated.

Configuration section	Parameter	Explanation
Basic settings	Record interval	Time between measurement data recordings

5.1.9 Setpoint module

This module uses the setpoints for the outputs entered via the configuration interface of the myDatanet server.

The following table specifies the relevant parameters for the module:

Configuration section	Modus/interface	Parameter	Explanation
Interface output channels 1-64 -› Basis	Off	---	
	Com1	Setpoint	Output value in the measurement unit
	Com2		
Output channels -› Basis	Off	---	---
	0-20 mA	Setpoint	Output value in the measurement unit
	4-20mA		
	Ext. warmup time[1)]	---	---
	Digital	Setpoint	Setpoint (on/off) that should be issued
	Freq	Setpoint	The setpoint multiplied by the factor equals the frequency in Hertz.
	PWM	Setpoint	Output value in the measurement unit
	Pulse	Setpoint	The setpoint multiplied by the factor equals the pulse/min.

[1)] In "Ext. warmup time" mode, the output is actuated by the device itself to activate the sensor supply for the measurement (see "Output channels" on page 133).

5.1.10 Scale module (outputs)

This module rescales the setpoint (e.g. mm) into the desired physical size (e.g. frequency) for the output.

The following table specifies the relevant parameters for the module:

Configuration section	Modus/interface	Parameter			Explanation
Interface output channels 1-64 -› Scale	Com1	Scale	Off	---	---
			On	0% modbus	Start of the output range in the unit of the modbus slave
				100% modbus	End of the output range in the unit of the modbus slave
				0%	Start of the output range in the measurement unit
				100%	End of the output range in the measurement unit
Output channels	Off	---			---
	Ext. warmup time[1)]				
	Digital	Basis -› Invert			Inverts the level issued on the device.
	Freq	Basis -› Factor			The setpoint multiplied by the factor equals the frequency in Hertz.
	PWM	Basis -› 0%			Start of the output range in the measurement unit
		Basis -› 100%			End of the output range in the measurement unit
	Pulse	Basis -› Factor			The setpoint multiplied by the factor equals the pulse/min.
		Config -› Duration			Duration of an output pulse [ms]

[1)] In "Ext. warmup time" mode, the output is actuated by the device itself to activate the sensor supply for the measurement (see "Output channels" on page 133).

5.2 Functionality of the internal data memory

Structure	Circular buffer
Total size	3.128 measurement cycles
Number of sectors	4
Sector size	782 measurement cycles

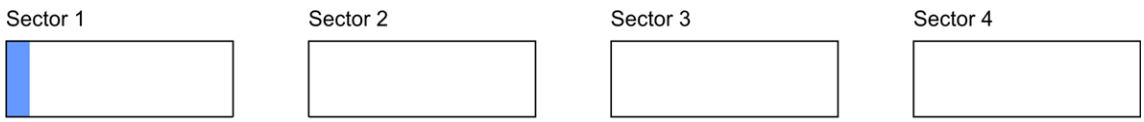
The internal data memory of the myDatalogMUCmini is designed as a circular buffer with 4 sectors. If the maximum number of data records (3.128) is achieved, the sector with the oldest data is deleted fully before new data can be saved in this sector again. This means that the internal data memory at the very least contains the measurement values of the last 2.346 cycles, however at most the measurement values of the last 3.128 cycles.

For this reason, it is recommended to coordinate the transmission cycle and record interval in such a way that a maximum of 2.346 measurement cycles have to be recorded between two transmissions. Please note, that if the measurement cycle is shorter than the record interval, the record interval still has to be used for the calculation. In this case, the reason for this is that although the measurement is completed in the measurement cycle, the determined data is saved in the data memory in the record interval. If it can be expected that individual transmissions fail due to poor network coverage or the measure quick divisor is activated via the trigger, this must also be taken into consideration when calculating the measurement cycles that are to be saved.

Note:

Additional explanation regarding the functionality of the circular buffer

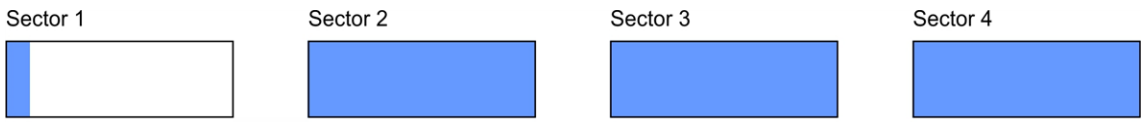
Data memory after the first measurement cycle:



Data memory after 3.128 measurement cycles:



Data memory after 3.128 +1 measurement cycles:



5.3 Procedure in case of connection aborts

If the connection is terminated, another attempt to establish a connection is made after 2min for all connections, except for ones that were established by initiating ALOHA transmission mode. The connection is attempted up to 2 times, as long as the previous attempt to establish a connection was successful. This means that if the connection fails and it is not possible to re-establish the connection during the first retry after 2min , a further retry will not be completed after 2min .

This process also applies if the connection is terminated during online mode.

5.4 ALOHA transmission mode

ALOHA transmission mode is a special connection mode whereby the myDatalogMUCmini establishes a connection to the myDatanet server for a period of time configured via the "Basic settings" configuration section (see "Aloha/wakeup duration" in chapter "Basic settings" on page 142).

There are two options for initiating the ALOHA transmission mode:

- Directly on the device using a button (see "Button to initiate ALOHA transmission mode" on page 79). The GSM network is re-selected during this process. This means that a list of all of the GSM networks in the receiving range of the device is created from which the strongest GSM network is then selected that is not on the "Ban list" (see "Ban list" on page 47) and that is part of one of the service providers supported by the integrated SIM chip (see www.microtronics.at/footprint). The internal fault counter is pre-initialised in such a way that if it was not possible to log into the selected GSM network, it is immediately added to the "Ban list". In this case, the next GSM network from the previously created list is used immediately. If the connection with all of the available GSM networks fails, the ban list is deleted and the connection attempt is aborted. If ALOHA transmission mode is then initiated again, the ban list is empty and all of the GSM networks in the receiving range, that are part of one of the service providers supported by the integrated SIM chip, are taken into consideration when selecting the network. The network selection also takes into consideration whether GPRS is available on the selected GSM network.
- Via the myDatanet server if the "Interval & wakeup" connection type has been selected (see "Basic settings" on page 142).



A speech bubble in the measurement instrument list (see "myDatanet Server Manual " 805002) with the title "Aloha" indicates that a device is in ALOHA transmission mode.



Clicking on the speech bubble with the "Aloha" title opens the ALOHA data window (see "myDatanet Server Manual " 805002), It contains the internal "Voltage" and "GSM level" measurement values, as well as the measurement values of the 3 universal inputs. The measurement values for the last thirty minutes are always shown in the ALOHA data window, which means that data from a previous ALOHA transmission may be included. The ALOHA data is generated every three seconds independently of the normal measurement values and is thus not saved with the standard measurement data.

If a measurement dataset is recorded during ALOHA transmission mode, it is immediately transferred to the myDatanet server and saved in the standard measurement data.

If a display element that uses one of the device's measurement values is shown on the server interface, this data is also transferred to the myDatanet server each time a measurement value is generated. If the measurement is completed more frequently than the recording ("measurement cycle" is not equal to zero and/or "burst interval" is not equal to zero) (see "Example to clarify the record, measurement and burst interval (in conjunction with the decay)" on page 143), this increases the data transmission. However, this data is not permanently stored on the server or on the device, but is instead used to update the display elements on the server (see "myDatanet Server Manual " 805002).

5.5 Automatic selection of the GSM network

The GSM network to which the device should register must be selected, as the myDatalogMUCmini is equipped with a SIM chip that provides a mobile connection via a variety of international service providers (see www.microtronics.at/footprint). This is completed automatically by the device. During this process it is taken into consideration if a connection to a certain GSM network is possible although the use of this GSM network repeatedly causes transmission errors. Such a GSM network is added to a ban list and is ignored during the next login attempt.

The following conditions ensure that the myDatalogMUCmini logs itself into a GSM network:

- PowerOn
- Expiry of the transmission cycle if the "Interval" connection type was selected (see "Basic settings" on page 142)
- Activation of the "Interval & Wakeup" or "Online" connection type (see "Basic settings" on page 142)
- Initiation of ALOHA transmission mode (see "ALOHA transmission mode" on page 45)

1. The myDatalogMUCmini checks whether any errors occurred during the last login attempt.

   No errors:

   The myDatalogMUCmini logs into the same GSM network that was also used during the last attempt.

   Error:

   Continue with step 2

   > ***Note:*** *A transmission error is not the same as an error that occurs when logging into the GSM network. It is possible for the login to the GSM network to work correctly but that a transmission error still occurs.*

The following steps are only completed if an error was detected during step 1:

2. The device searches for GSM networks within its receiving range and creates a list of available GSM networks.
3. A search of the list of available GSM networks is completed and the strongest GSM network is selected that is not on the "ban list" and is assigned to one of the service providers supported by the integrated SIM chip (see www.microtronics.at/footprint).

   > ***Note:*** *An explanation detailing the conditions when a GSM network is added to the "ban list" is provided in "Ban list" on page 47.*

5.5.1 Ban list

The ban list provides space for 5 entries and is not saved in a volatile memory. This means that it is also still available following a PowerOn.

If there is a GPRS error (see error codes -975 to -993 in chapter "Modem error" on page 307), then an internal counter is increased by 2 . If a transmission was completed without errors, i.e. all of the data was transferred, this internal counter is reduced by 1 . The current GSM network is added to the ban list if the internal counter reaches the value 6 . The relevant GSM network is automatically added to the ban list, if the "MODEM CMME ERROR" error occurs (see "Modem error" on page 307) with parameter 32 (NETWORK NOT ALLOWED) or 529 (MODEM CMME ERROR). The ban list is deleted if no GSM network can be selected from the list of available GSM networks in the receiving range taking the ban list into consideration. Another search for GSM networks in the receiving range is completed once the ban list is deleted and the network selection is completed with the ban list that is now empty (see step 3 in chapter"Automatic selection of the GSM network" on page 46).

The quality of the data transfer is monitored in addition to the establishment of the GPRS connection. If the data transmission fails 10 times in succession despite an existing GPRS connection, the currently selected GSM network is also added to the ban list. The same applies in this case, which means that if, when taking the ban list into consideration, there is no further GSM network available, the ban list is deleted and selection of the strongest GSM network is restarted.

5.6 Script eradication

5.6.1 Instruction list

The instruction list is run through at the time of every measurement once all of the measurement values have been generated. If an error occurs during this process, the script execution will stop and enter the relevant error code in the device log (see "Instruction list error codes" on page 158). Only then will the measured data and data calculated by the instruction list be recorded, if the record interval has also expired.

> ***Note:*** *If an instruction list is used to write directly to an output, the setpoint entered via the input screen of the myDatanet server is overwritten with the calculated value.*

5.6.2 Pawn

The PAWN script must include the following two functions:

main();

This function is executed during a PowerOn and when the PAWN script is exchanged. It should include all initialisations that only have to executed once during program start-up.

Mdn_CtrlFinish();

This function is called up at the time of every measurement once all of the measurement values have been generated. It should comprise all of the calculations and functions that should be performed cyclically.

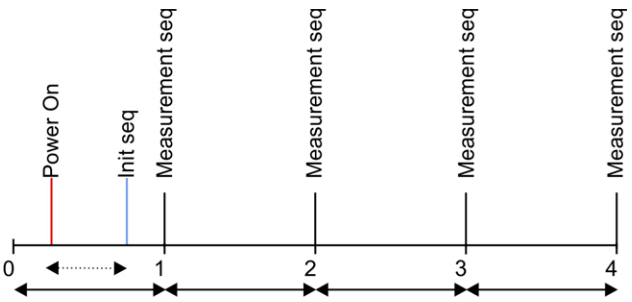
> ***Note:*** *If a pawn script is used to write directly to an output, the setpoint entered via the input screen of the myDatanet server is overwritten with the calculated value.*

If an error occurs during this process, the script execution is stopped and deactivated. The relevant error code is also entered in the device log (see "Pawn script error codes" on page 200).

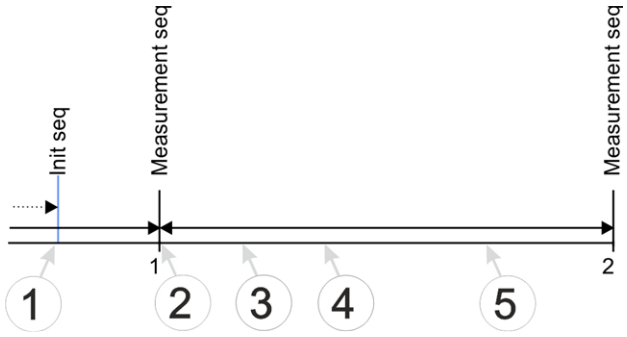
5.6.2.1 Script parsing

If script parsing was activated for one of the COM interfaces, the relevant interfaces can be accessed by means of the "Mdn_SerialEvent()", "Mdn_SerialRx()", "Mdn_SerialTx()" and "Mdn_SerialFinish()" PAWN script functions. The baud rate, stop bits, parity, data bits, frame timeout and activation of a retry are still configured via the "Interface" configuration section (see "Interfaces" on page 83). The following two graphics indicate at what times the "Mdn_SerialEvent()" (see Mdn_SerialEvent()) and "Mdn_SerialRx()" (see Mdn_SerialRx()) callback functions are called up and what parameters are transferred during this process.

Basic setting	◄——►	Measurement cycle	60 sec.
Interfaces ->Config.	◄·····►	Warmup time	30 sec.



Sequence of the serial communication



Sequence of the serial communication in detail

1 Function: Mdn_SerialEvent() Event: MDN_SERIAL_EVENT_INIT	4 Only when "Retry" is active Function: Mdn_SerialEvent() Event: MDN_SERIAL_EVENT_TIMEOUT or Mdn_SerialRx()
2 Function: Mdn_SerialEvent() Event: MDN_SERIAL_EVENT_MEASURE	5 Function: Mdn_CtrlFinish()
3 Function: Mdn_SerialEvent() Event: MDN_SERIAL_EVENT_TIMEOUT Event: MDN_SERIAL_EVENT_MEASURE or Mdn_SerialRx()	

Explanation of 1

A delay can be set between the PowerOn and the dispatch of the init sequence via the warmup time (call up of the "Mdn_SerialEvent()" callback function by the system with the "MDN_SERIAL_EVENT_INIT" event). However, the warmup time can only be entered for Com2 via the configuration interface. The relevant parameter for configuring the interface is located in the "Config" tab of the input screen (see "Config" on page 94). The warmup time for interface Com1 is always 0 sec. The user should react to the "MDN_SERIAL_EVENT_INIT" event by using the "Mdn_SerialTx()" function to send the init sequence via the interface.

Explanation of 2

If it is time to initiate a measurement, the "Mdn_SerialEvent()" callback function is called up by the system with the "MDN_SERIAL_EVENT_MEASURE" event. The internal processing of the measurement values also starts at this time (see "Internal processing of the measurement values" on page 31). The user must react to the "MDN_SERIAL_EVENT_MEASURE" event by using the "Mdn_SerialTx()" function to send the measurement sequence via the interface.

> ***Note:*** *The time of the first measurement following the PowerOn is calculated and is not completed exactly according to the time after the PowerOn specified via the measurement cycle. If the measurement cycle is 1 minute long, the first time of the measurement is selected in such a way that it is completed at the full minute mark. This means that if the PowerOn is completed at 12:05:34, the first measurement is taken at 12:06:00, i.e. 26 sec. after the PowerOn.*

Explanation of 3

This point is achieved if data has been received or a timeout has occurred. The time for the timeout is entered via the "Basis" tab of the input screen for configuring the interface (see "Config" on page 94).

If a retry is activated, the "Mdn_SerialEvent()" callback function with the "MDN_SERIAL_EVENT_MEASURE" event is called up by the system in the event of a timeout. The user must react to this event by using the "Mdn_SerialTx()" function to resend the measurement sequence.

If a retry is not activated, the "Mdn_SerialEvent()" callback function with the "MDN_SERIAL_EVENT_TIMEOUT" event is called up by the system in the event of a timeout. In this case, the receipt of data must be terminated by the "Mdn_SerialFinish()" function.

However, if valid data has been received, the "Mdn_SerialRx()" callback function is called up by the system. It supplies the received data as an array. Following the evaluation of the data, the user must copy the required measurement values to the interface channel by means of the "Mdn_SetCh()" function (see Mdn_SetCh()).

> ***Important note:*** *During this process, only interface channels, that are connected to an interface for which script parsing was activated, should be used. Otherwise the data generated by the system will be overwritten.*

If all of the required data has been received, the receipt of data must be terminated by the "Mdn_SerialFinish()" function. Otherwise the internal processing of the measurement values cannot be continued. Following the call up of the "Mdn_SerialFinish()" function the internal processing of the measurement values is continued from the decay module onwards (see "Internal processing of the measurement values" on page 31). This means the levels for the alarms or triggers can still be edited via the "Alarm" and "Trigger" tabs in the input screen for configuring interface channels 1-64 (see "Alarms" on page 121 or "Trigger" on page 122).

Explanation of 4 (only when "Retry" is activated)

In the event of a timeout, the "Mdn_SerialEvent()" callback function is called up by the system with the "MDN_SERIAL_EVENT_TIMEOUT" event. In this case, the receipt of data must be terminated by the "Mdn_SerialFinish()" function. However, if valid data has been received, the "Mdn_SerialRx()" callback function is called up by the system. It supplies the received data as an array. Following the evaluation of the data, the user must copy the required measurement values to the interface channel by means of the "Mdn_SetCh()" function (see Mdn_SetCh()).

> ***Important note:*** *During this process, only interface channels, that are connected to an interface for which script parsing was activated, should be used. Otherwise the data generated by the system will be overwritten.*

If all of the required data has been received, the receipt of data must be terminated by the "Mdn_SerialFinish()" function. Otherwise the internal processing of the measurement values cannot be continued. Following the call up of the "Mdn_SerialFinish()" function the internal processing of the measurement values is continued from the decay module onwards (see "Internal processing of the measurement values" on page 31). This means the levels for the alarms or triggers can still be edited via the "Alarm" and "Trigger" tabs in the input screen for configuring interface channels 1-64 (see "Alarms" on page 121 or "Trigger" on page 122).

Explanation of 5

The "Mdn_CtrlFinish()" (see Mdn_CtrlFinish()) callback function is called up if the internal processing of the measurement values reaches the control module. It is then possible to send additional commands via the interface by means of the "Mdn_SerialTx()" function to, for example, control the actuators.

6872
6872
6872

Chapter 6 Storage, delivery and transport

6.1 Inspection of incoming deliveries

Please check the shipment immediately after receiving for completeness and intactness. Immediately report any discovered transport damages to the delivering carrier. Also notify the manufacturer in writing about this without delay. Report any incompleteness in the delivery to the responsible representative or directly to the company headquarters of the manufacturer within 2 weeks (see "Contact information" on page 319).

Note: *Any claims received thereafter will not be acknowledged!*

6.2 Scope of supply

The standard scope of supply of the myDatalogMUCmini (3000051) includes:

- myDatalogMUCmini
- Antenna 900 SMA-M angled (300026)
- myDatanet Tool Pen (206.646)
- Type plate sticker to attach to the control cabinet
- 3x stickers with the type designation "myDatalogMUCmini "

Additional accessories such as assembly sets, antennas, power supplies, sensors, etc. depending on the order specifications. Please check this against the delivery slip.

6.3 Storage

The following storage conditions must be observed:

myDatalogMUCmini	Storage temperature	-20...+60°C
	Humidity	15...90%rH

Store the measurement technology so that it is protected against corrosive and organic solvent vapours, radioactive emissions and strong electromagnetic radiation.

6.4 Transport

The myDatalogMUCmini should not be subjected to heavy shocks, bumps, impacts or vibrations. The original packaging must always be used for transport.

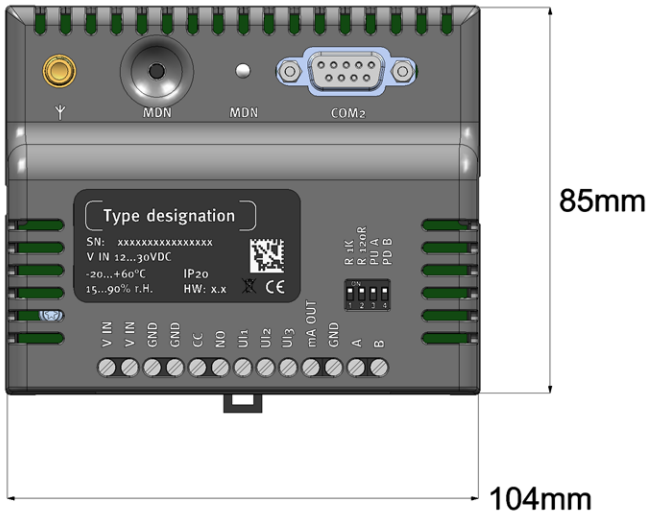
6.5 Return

Every return must be accompanied by a fully field-out return form. This return form is available in the service area of the myDatanet-Server. An RMA number is mandatory for any returns and can be obtained from the Support & Service Centre (see "Contact information" on page 319). The return shipment of the myDatalogMUCmini must occur in the original packaging and with freight and insurance paid to Microtronics Engineering GmbH (see "Contact information" on page 319). Insufficiently cleared return shipments will otherwise not be accepted!

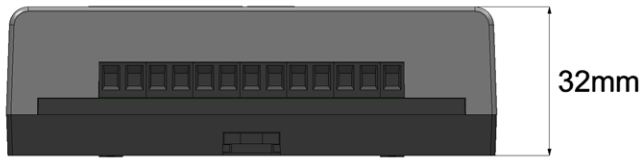
Chapter 7 Installation

Important note: *To prevent any damage to the device, the work described in this section of the instructions must only be performed by qualified personnel.*

7.1 Dimensions



Dimensions: Width and height



Dimensions: Depth

7.2 Installing the myDatalogMUCmini

Important note:

- *Ensure installation is completed correctly.*
- *Comply with existing legal and/or operational directives.*
- *Improper handling can cause injuries and/or damage to the devices.*

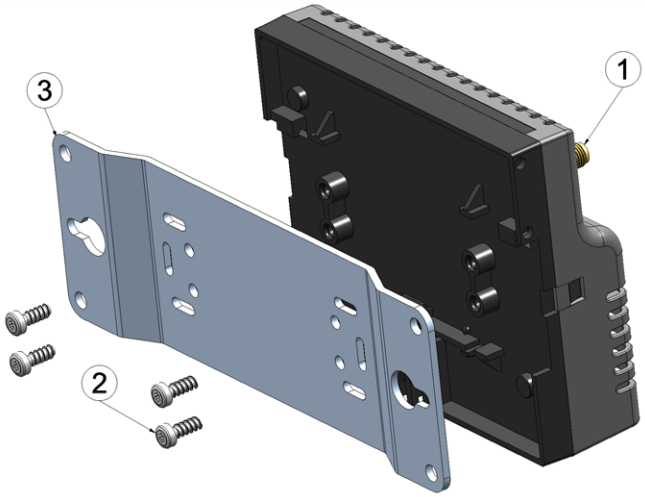
The installation site must be selected according to specific criteria. The following conditions must be avoided in any case:

- Direct sunlight
- Objects that radiate intense heat (maximum ambient temperature: -20...+60°C)
- Objects with a strong electromagnetic field (frequency converter or similar)
- Corrosive chemicals or gases
- Mechanical impacts
- Direct installation on paths or roads
- Vibrations
- Radioactive emissions

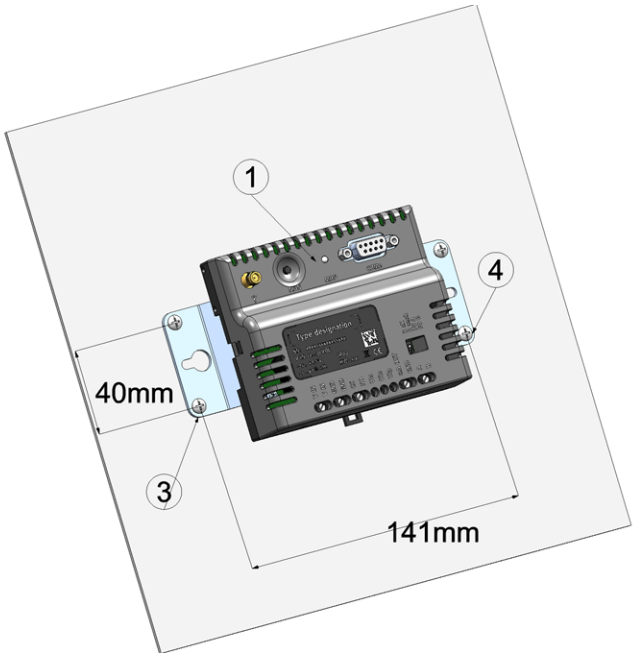
Note: *Approx. 2-5 cm of space must be left below the device for the cable connections. The antenna connection is located on the front of the device. The space required depends on the antenna used. Further information about the installation dimensions can be found in the relevant sub-chapter.*

7.2.1 Wall mounting

For wall mounting the optional "Universal bracket for housing myDatanet 86x126 (206.640)" equipment is required.



Step 1 of the wall mounting



Step 2 of the wall mounting

1 myDatalogMUCmini	3 Assembly loop (included in the delivery scope of 206.640)
2 Delta PT M3.5x8 Torx 15 (included in the delivery scope of 206.640)	4 Raised head tapping screw 3.5x32 (included in the delivery scope of 206.640)

1. First attach the assembly loop (3) using the screws (2) included in the "Universal bracket for housing myDatanet 86x126 (206.640)" equipment set to the myDatalogMUCmini (see "Step 1 of the wall mounting" on page 56).
2. Drilling the holes for mounting:

   If you want to use the tapping screws (4) included in the equipment set to secure the assembly loop (3) to the wall, drill four 6-mm holes in the wall using the drill template as a guide in accordance with the dimensions in the diagram "Step 2 of the wall mounting" on page 56.

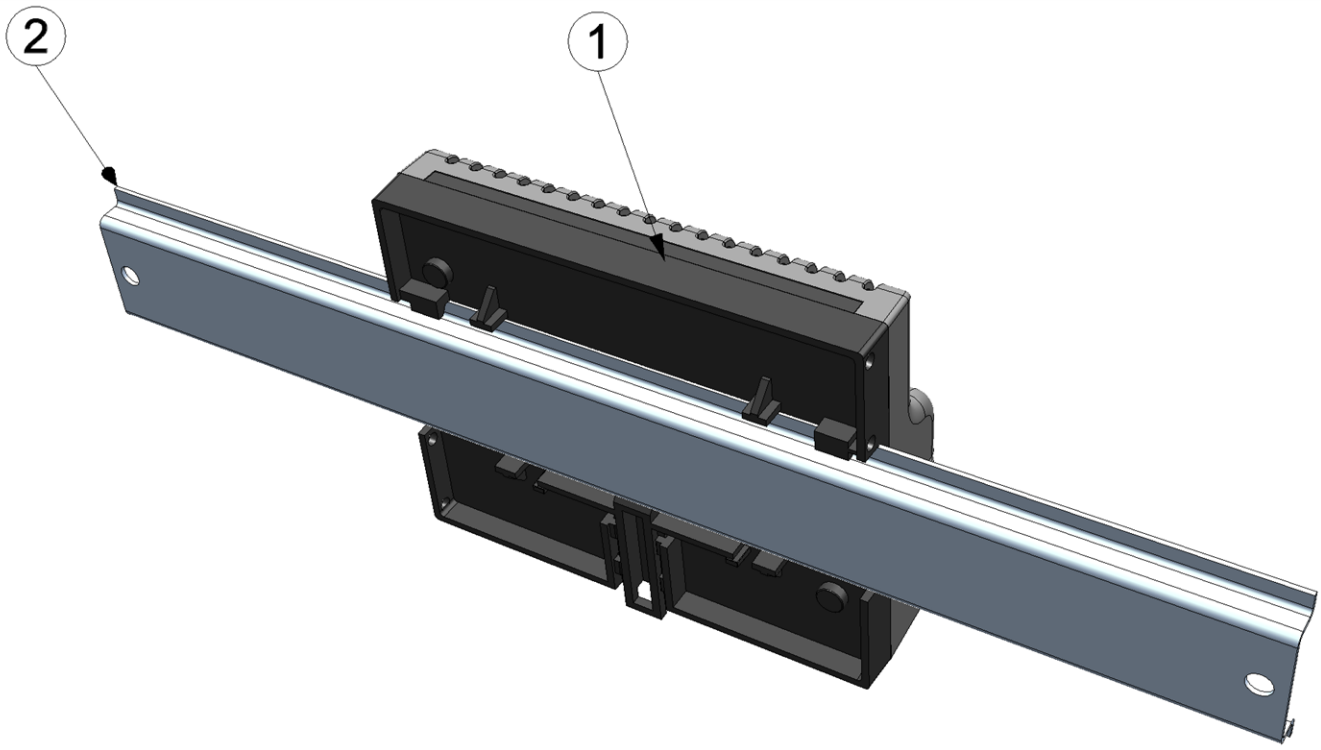
   If you want to use your own fastening screws you can use the drill guide provided in the equipment set to determine the position of the holes. The diameters are then determined by the screws that are being used and any wall plugs that may be required.

3. Securing the myDatalogMUCmini to the wall:

   If you want to use the tapping screws (4) included in the equipment set, first of all insert the supplied wall plugs into each of the four drill holes before screwing the myDatalogMUCmini with the attached assembly loop (3) to the wall (see "Step 2 of the wall mounting" on page 56).

   If you are using your own screws, you must also insert the wall plugs into the holes before mounting the myDatalogMUCmini with the attached assembly loop (3) to the wall (see "Step 2 of the wall mounting" on page 56).

7.2.2 Top-hat rail assembly



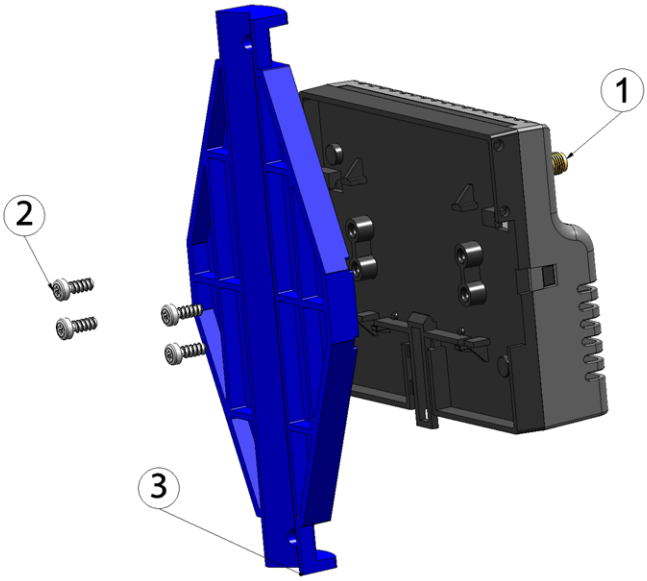
Top-hat rail assembly

1 myDatalogMUCmini	2 Top-hat rail

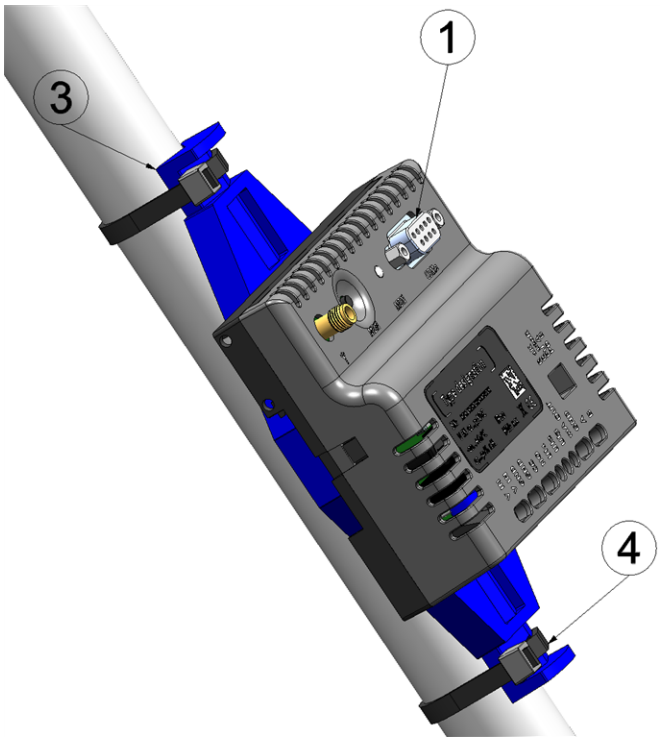
1. Place the myDatalogMUCmini on to the top edge of the top-hat rail. Turn slightly around the horizontal axis so that the myDatalogMUCmini clicks into the top-hat rail (see Figure "Top-hat rail assembly" on page 57).

7.2.3 Pipe assembly

The optional "Pipe mounting kit for myDatanet housing 86x126 (206.660)" equipment is required for the pipe assembly.



Step 1 of the pipe assembly



Step 2 of the pipe assembly

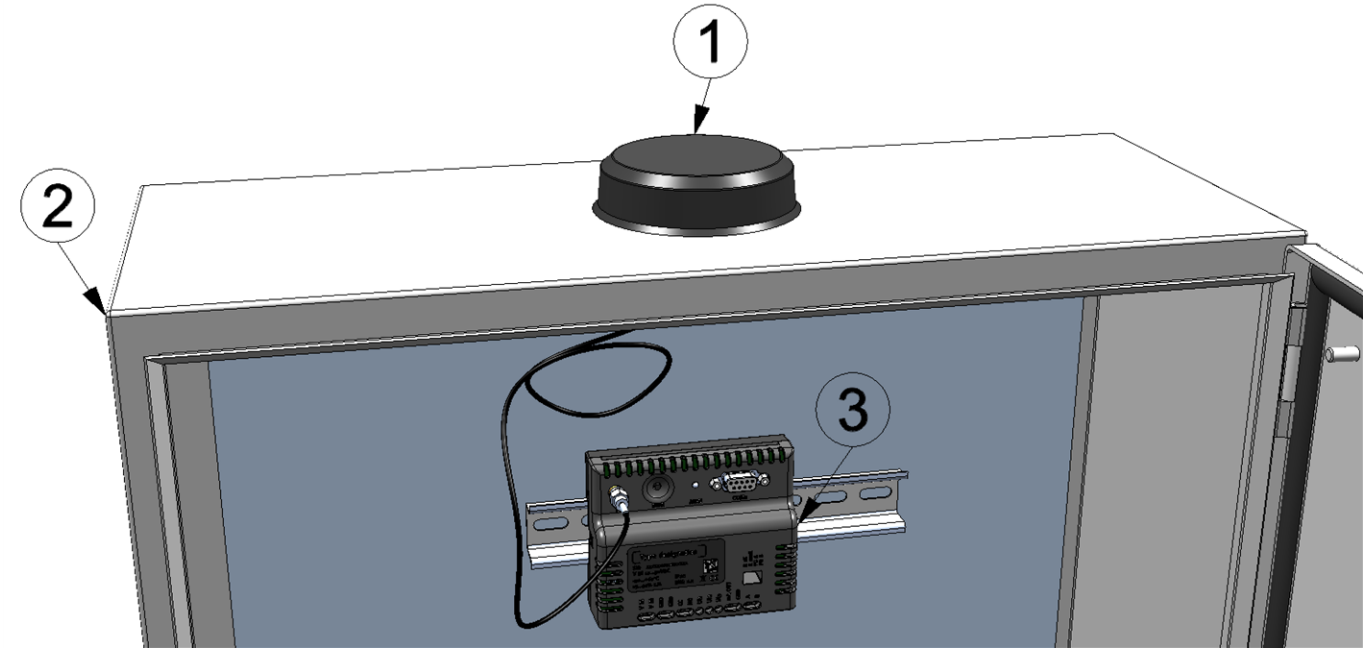
1	myDatalogMUCmini	3	Assembly loop (included in the scope of delivery of 206.660)
2	Delta PT M3.5x8 Torx 15 (included in the scope of delivery of 206.660)	4	Cable binder (included in the scope of delivery of 206.660)

1. First attach the assembly loop (3) using the screws (2) included in the "Pipe mounting kit for myDatanet housing 86x126 (206.660)" equipment set to the myDatalogMUCmini (see "Step 1 of the pipe assembly " on page 58).
2. Position the myDatalogMUCmini with the attached assembly loop (3) on the pipe and use the supplied cable binders (4) to secure the myDatalogMUCmini (see "Step 2 of the pipe assembly" on page 58).

7.2.4 Assembly in a control cabinet

The Antenna 900 SMA-M angled (300026) included in the scope of supply is not suitable for assembly within a control cabinet as the GSM signal is shielded by the metal of the cabinet. In this case, the manufacturer recommends using the Flat antenna Smart Disc SMA-M 2,5m (206.816) or Flat Antenna Smart Disc US SMA-M 2,5m (206.818) that are available as accessories.

Important note: *The installation of the Flat antenna Smart Disc SMA-M 2,5m or Flat Antenna Smart Disc US SMA-M 2,5m is only possible up to a wall thickness of 3mm .*



Control cabinet with mounted Flat antenna Smart Disc SMA-M 2,5m (206.816)

1	Flat antenna Smart Disc SMA-M 2,5m (206.816)	3	myDatalogMUCmini
2	Control cabinet		

7.3 Safety instructions for the cabling

Important note: *To avoid any damage, always switch off the voltage supply to the device when performing electrical connections.*

When connections are made to the myDatalogMUCmini , the following warnings and information must be observed, in addition to the warnings and information found in the individual chapters on the installation. Further safety information is included in "Safety instructions" on page 21.

7.3.1 Information on preventing electrostatic discharges (ESD)

Important note: *Maintenance procedures that do not require the device to be connected to the power supply should only be performed once the device has been disconnected from the mains power supply to minimise hazards and ESD risks.*

The sensitive electronic components inside the device can be damaged by static electricity, which can impair the device performance or even cause the device to fail. The manufacturer recommends the following steps to prevent any damage to the device caused by electrostatic discharges:

- Discharge any static electricity present on your body before handling the electronic components of the device (such as circuit boards and components attached thereto). To do this, you can touch a grounded metallic surface such as the housing frame of a device or a metal pipe.
- Avoid any unnecessary movements to prevent the build-up of static charges.
- Use antistatic containers or packaging to transport components that are sensitive to static.
- Wear an antistatic wristband that is grounded via a cable to discharge your body and keep it free of static electricity.
- Only touch components that are sensitive to electric charges in an antistatic working area. If possible, use antistatic mats and work pads.

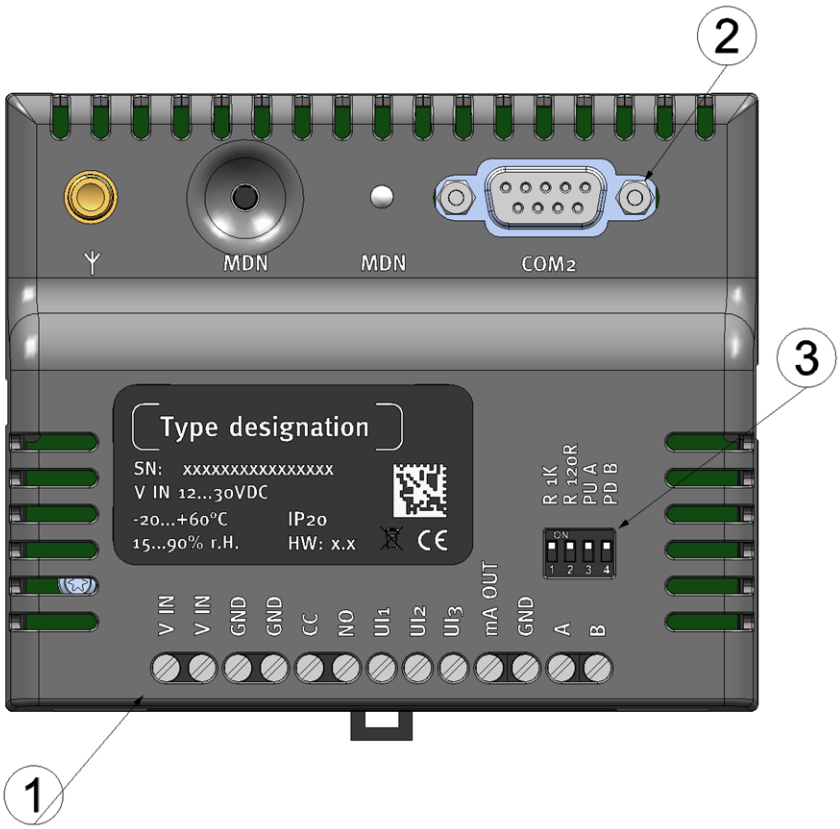
7.4 Electrical installation

Important note: *Only qualified personnel should undertake the installation described in this chapter of the operating instructions to avoid any damage to the device.*

7.4.1 Connecting the sensors, actuators and power supply

Important note:

- *All wiring work must be performed in the de-energised state.*
- *Ensure installation is completed correctly.*
- *Comply with existing legal and/or operational directives.*
- *Improper handling can cause injuries and/or damage to the instruments!*
- *Run all data and power cables so that they do not pose a trip hazard and ensure that cables do not have any sharp bends.*



Connection of the sensors and power supply

1 Connection strip	**3** Dip switch (dip switch 1) for activating/deactivating the load or clamp resistances for Com1 (RS485)
2 Sub-D connector of the serial interface (Com2)	

Assignment of the main terminal block

1	V IN	Supply voltage: 12...30VDC (+/- 10%), max. 3W grounded
2	V IN	
3	GND	Ground
4	GND	
5	CC	Isolated switch contact (parallel connection of an optoswitch and a relay; normally open)
6	NO	
7	UI1	Universal input 1
8	UI2	Universal input 2
9	UI3	Universal input 3
10	mA OUT	Analogue output (0/4-20mA)
11	GND	
12	A	COM1 (RS485)
13	B	

Dip switch (dip switch 1) for activating/deactivating the load or clamp resistances for Com1 (RS485)

R 1K	1k load resistance between RS485 A and B
R 120Ω	120Ω load resistance between RS485 A and B
PU A	1k pull up to RS485 A
PD B	1k pull down to RS485 B

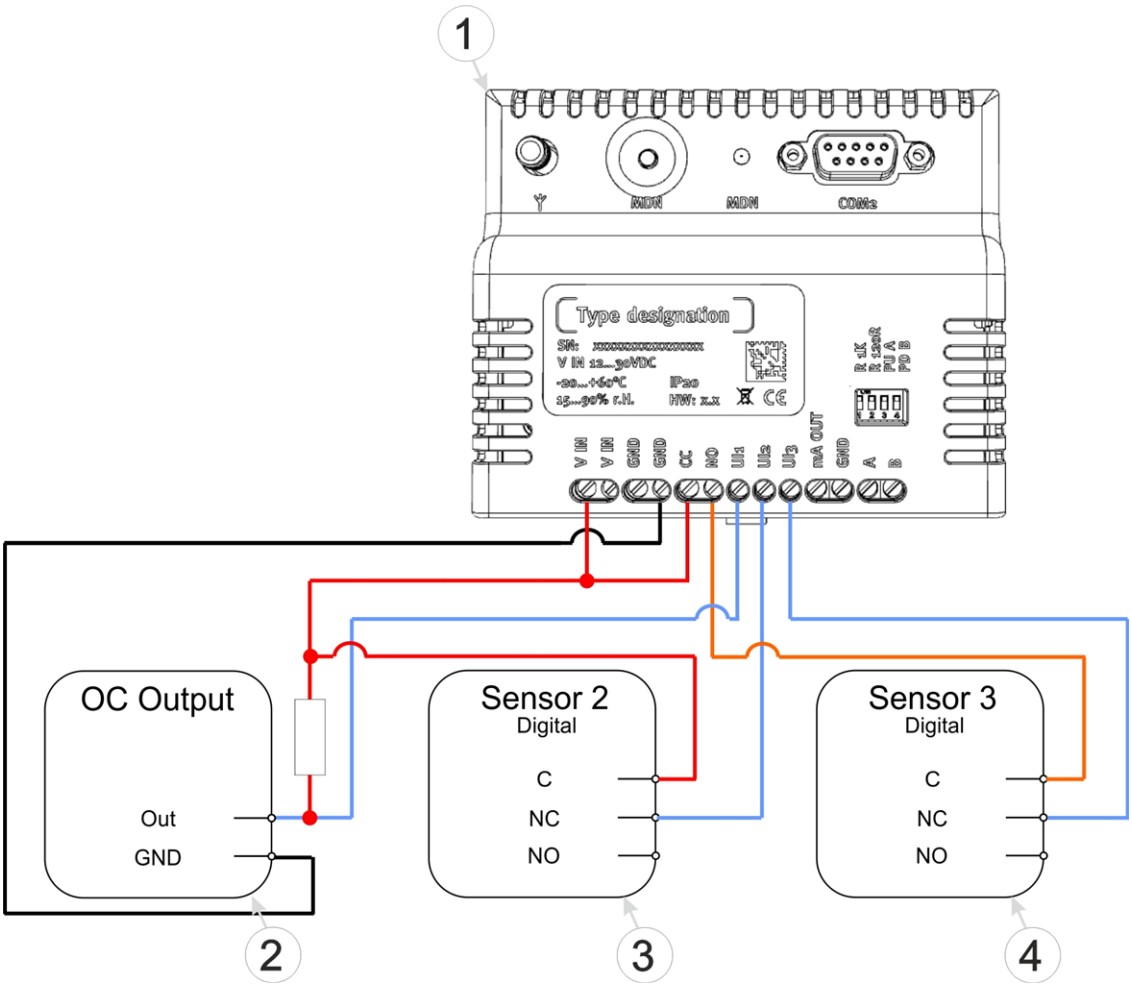
1. Connect your sensors and actuators with the inputs and outputs. Ensure that no current is present! Ensure the supply cables for the myDatalogMUCmini are in a de-energised state when being connected to the supply connectors.
2. Connect the antenna (see "Connection of the GSM antenna" on page 64).
3. Switch on the 12...30VDC supply voltage of the myDatalogMUCmini . The status LED should then start to flicker (see "Status LED" on page 80), indicating that a connection is being established.

The following step is not mandatory.

4. Check whether the connection to the myDatanet has worked correctly (see "Testing communication with the device" on page 76).

7.4.1.1 Connection examples

7.4.1.1.1 Connection examples: Digital modes

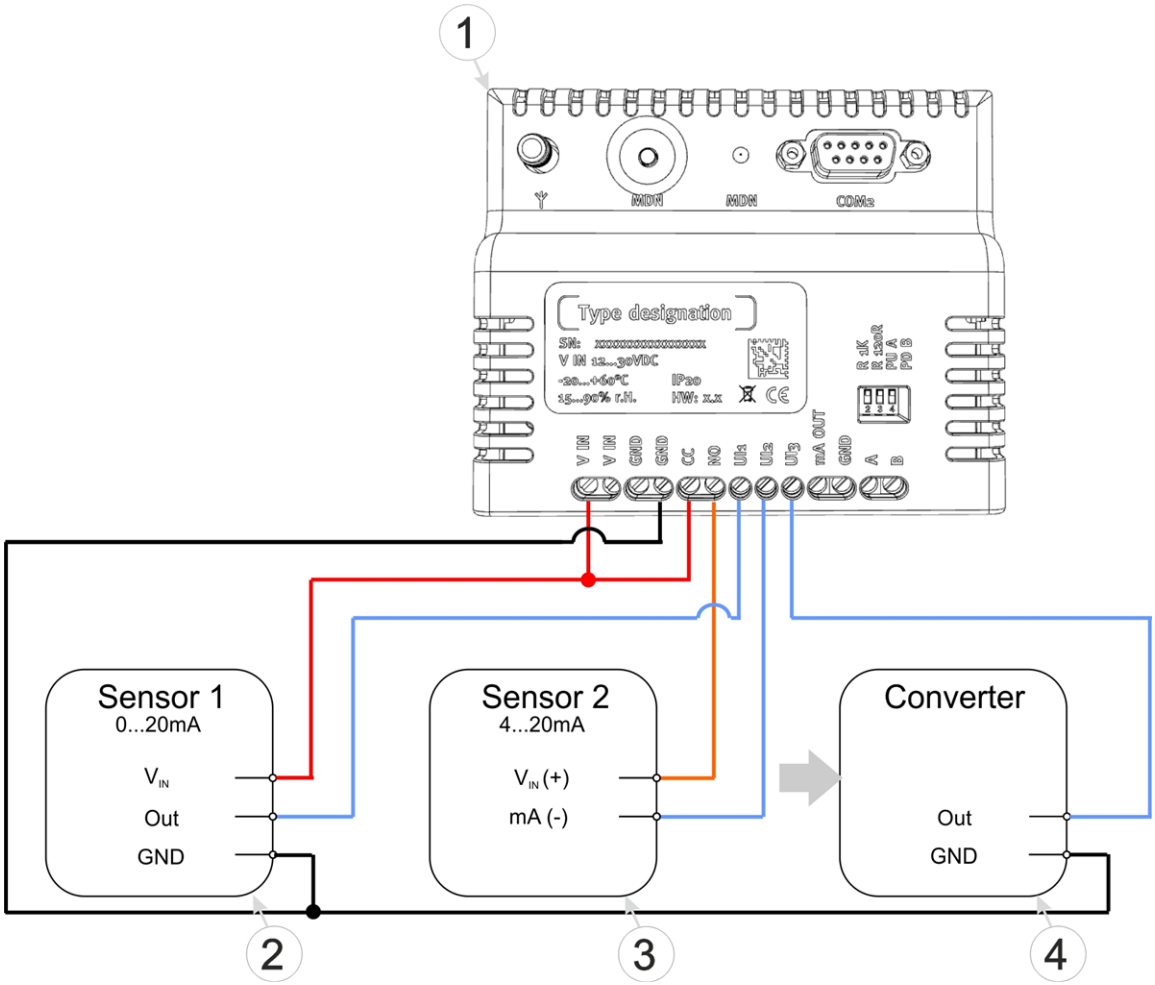


Connection examples: Digital modes

1	myDatalogMUCmini	3	Isolated relay contact (directly connected to V IN)
2	Sensor with open collector output	4	Isolated relay contact

Note: *In the connection example above, "Sensor 3" is not directly connected to the supply voltage V IN but is conducted via the isolated switch contact (CC, NO). This ensures that the supply of "Sensor 3" (provided the isolated switch contact is operated in "Ext. warmup time" mode) is switched in accordance with the "Ext. warmup time" (see "Output channels" on page 133). This means that when the switch contact is closed, a permanent power consumption of 2,55...6,38mA (depending on supply voltage V IN) due to a load of 4k7 is prevented. However, this method is not suitable for the "Cnt.Day", "Cnt.Intervl.", "Freq" and "PWM" operating modes as these modes require a permanent supply of the sensors. This method is not suitable even if an input operated in "Digital" mode is to be used to initiate a trigger event.*

7.4.1.1.2 Connection examples: Analogue modes



Connection examples: Analogue modes

1	myDatalogMUCmini	3	2-wire mA sensor
2	3-wire mA sensor (directly connected to V IN)	4	Signal converter, isolating transducer

> ***Note:*** *In the connection example above, "Sensor 2" is not directly connected to the supply voltage V IN but is conducted via the isolated switch contact (CC, NO). This ensures that the supply of "Sensor 2" (provided the isolated switch contact is operated in "Ext. warmup time" mode) is switched in accordance with the "Ext. warmup time" (see "Output channels" on page 133). This means that the power consumption can be reduced and the service life of the sensor extended, as it is only activated while the measurement is being completed.*

7.4.2 Connection of the GSM antenna

> ***Important note:*** *To ensure the correct functionality, only use antennas that are supplied by the manufacturer.*

The standard antenna is directly attached to the antenna connector (see "Overview" on page 24) of the myDatalogMUCmini . In the event of a low radio signal strength, you can use the Flat antenna Smart Disc SMA-M 2,5m (206.816) or Flat Antenna Smart Disc US SMA-M 2,5m (206.818).

If the distance between the antenna position and the myDatalogMUCmini is too great, you can use a 2.5m Extension cable for antenna SMA-M/SMA-F 2,5m (206.807).

1. Ensure that the myDatalogMUCmini is de-energised.
2. If you need an antenna extension, connect this to the antenna first.
3. Connect the antenna extension or antenna directly to the antenna connector of the myDatalogMUCmini (see "Overview" on page 24).

   Important note: Do not apply too much force when tightening the antenna. Do not use any tools to tighten the antenna or antenna extension; only tighten it manually.
4. Switch the voltage supply of the myDatalogMUCmini back on.

The following step is not mandatory.

5. Check whether the connection to the myDatanet has worked correctly (see "Testing communication with the device" on page 76).

7.4.3 Technical details about the universal inputs

Note: The universal inputs are not galvanically isolated.

7.4.3.1 0/4 to 20mA mode

Note: Above 23,7mA, the relevant input becomes highly resistive (safety shutdown to prevent damage to the universal input).

Resolution	6,3µA
I_{max}	23,7mA
Load	96Ω

7.4.3.2 0 to 2V mode

Resolution	610µV
U_{max}	2,5V
Load	10k086

7.4.3.3 0 to 10V mode

Resolution	7,97mV
U_{max}	32V
Load	4k7

7.4.3.4 Standard digital modes (PWM, frequency, digital, day counter, impulse counter)

General	U_{max}	32V
	Low	‹1,31V
	High	›2,61V
	Load	4k7
PWM	Measurement range	1...99%
	f_{max}	100Hz
	Minimum impulse length	1ms
Frequency	Measurement range	1...1000Hz
Day and impulse counter	Minimum impulse length	1ms

7.4.4 Technical details regarding the modbus interface (Com1)

Note: *The modbus interface Com1 corresponds to standard EIA-485.*

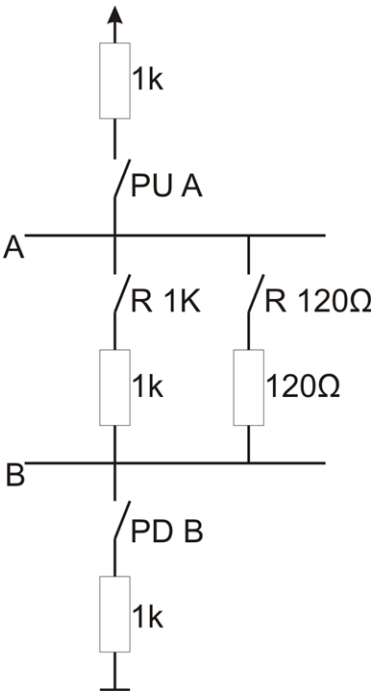
The modbus interface Com1 includes an input common mode range that covers the full area specified for RS485 (-7V...+12V). Higher voltages damage the interface. Differential signals of more than +/- 200mV within the specified input common mode range are detected correctly. In send mode, the output signal is in the range of 1,5...3,3V .

Baud rate	300-115200
Stop bits	1, 2
Parity	N, E, O
Data bits	7, 8
Modes	RTU ASCII

Com1 is a RS485 interface that can be configured as a modbus master and a modbus slave (see "Interfaces" on page 83). In modbus master mode, one or more slaves can be connected to the myDatalogMUCmini . The load or clamp resistances can be connected via dip switch 1 (see "Connecting the sensors, actuators and power supply" on page 60).

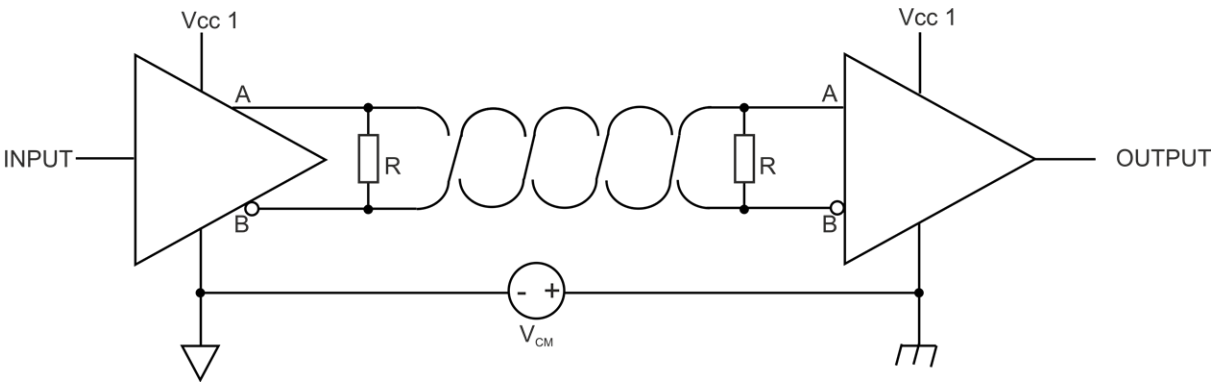
Dip switch (dip switch 1) for activating/deactivating the load or clamp resistances for Com1 (RS485)

R 1K	1k load resistance between RS485 A and B
R 120Ω	120Ω load resistance between RS485 A and B
PU A	1k pull up to RS485 A
PD B	1k pull down to RS485 B



Schematic diagram of the switchable load resistances

Note: *Additional explanation regarding the connection of two RS485 bus participants*



Schematic diagram: Connection of two RS485 bus participants

A problem occurs if there is no connection between the GND potentials of the sender and recipient. A common mode voltage (V_{CM}) occurs in this case. The GND potential difference must not exceed max. +/- 7V . Higher voltages will damage the interface. Temporary overvoltages (ESD, EFT and surge) are, however, absorbed by protective circuits.

Note: The common mode input voltage range of -7V...+12V specified for the RS485 is determined from the max. permissible GND potential difference (+/- 7V) and the max. permissible output voltage range of 0...5V for RS485.

7.4.4.1 Modbus slave mode

The following table details the possible access functions depending on the data type of the interface channel:

Modbus add.	Data type	Read function	Write function
0x0000 : 0x003F	Digital	Read coils (FC 01)	Write single coil (FC 05) Write multiple coils (FC 15)
0x0000 : 0x007F	Signed 16/32 bit Unsigned 16/32 bit Float	Read holding registers (FC 03)	Write single register (FC 06) Write multiple registers (FC 16)

The following table details the possible access functions depending on the data type of the interface output channel:

Modbus add.	Data type	Read function	Write function
0x0800 : 0x083F	Digital	Read discrete inputs (FC 02)	---
0x0800 : 0x087F	Signed 16/32 bit Unsigned 16/32 bit Float	Read input registers (FC 04)	---

7.4.5 Technical details about the RS232 interface (Com2)

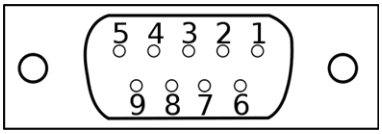
Note: *The modbus interface Com2 is compatible with standard TIA/EIA-232-F.*

Important note: *The RS232 interface of the myDatalogMUCmini does not support hardware handshakes.*

The output drivers are protected against overloading and are not damaged by a short circuit to the GND or +/-15 V. The inputs are equipped with a 5 kΩ load resistance.

Baud rate	300- 115200
Stop bits	1, 2
Parity	N, E, O
Data bits	7, 8
Modes	ASCII

The direction of the signals corresponds to that of a DCE (e.g. modem).



9-pin Sub-D(f)

Assignment of the Sub-D connector

Pin	Signal	Type
1	NC	
2	RXD	O (low: -5,7V; high: 6,2V)
3	TXD	I (low: ‹0,8V; high: ›2,5V)
4	NC	
5	GND	
6	NC	
7	RTS[1)]	I (low: ‹0,8V; high: ›2,5V)
8	CTS[1)]	O (low: -5,7V; high: 6,2V)
9	5V [2)] max. 750mA	

[1)] The current firmware does not support the hardware handeshake.

[2)] Supply voltage (reserved for extensions)

If your sensor also comprises a SUB-D(f) connector, you can use the Gender changer 9-pin D-Sub male/male (206.684) provided as an accessory. If the connection characteristics (transmission direction of the individual signal lines) of your sensor also correspond to that of a DCE (e.g. modem), you can use the Null modem adapter 9-pin D-Sub female/male (206.686) that is available as an accessory.

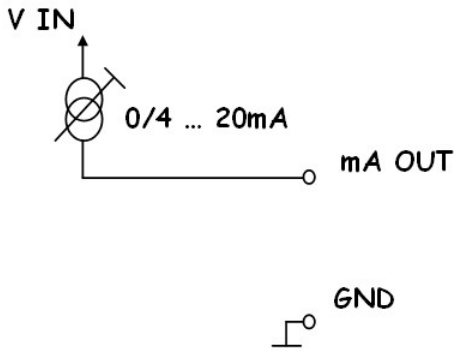
7.4.5.1 Error codes of the RS232 interface

Log entry		Parameter		Description
Code	Plain text	Code	Plain text	
2000	LOG MODULE ERR	0	CHAR TIMEOUT	The time lag between two received characters is larger than the character timeout entered via the input screen for configuring the interface (see "Config" on page 94).
		1	FRAME TIMEOUT	The complete answer was not received within the frame timeout. This period of time is entered via the input screen for configuring the interface (see "Com2 (serial, RS232)" on page 90).
		2	INVALID FRAME	The answer frame of the sensor is invalid if, for example, the number of columns of the sensor does not correspond to the number entered via the configuration interface (see "Column" in chapter "Config." on page 117).

7.4.6 Technical details about the outputs

7.4.6.1 mA OUT 1: Analogue output (mA OUT, GND)

Note: *The analogue output of the myDatalogMUCmini is an active, not galvanically isolated current output.*

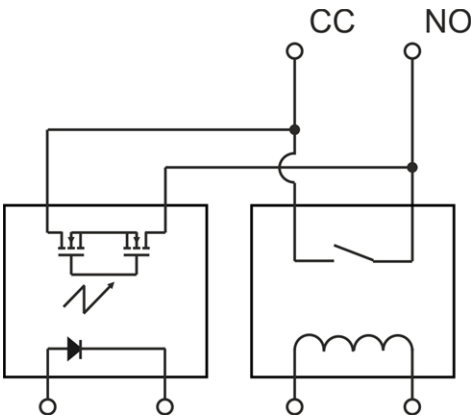


Replacement circuit diagram for the analogue output

Load voltage	Corresponds to the supply voltage (12...30VDC)
max. load (32V, 20mA)	1200Ω
$I_{out\ max}$	20mA

7.4.6.2 Relay 1: Isolated switch contact (CC, NO)

In idle mode, the operating contact of the relay and optoswitch is open (normally open).



Equivalent circuit diagram for the isolated switch contact

General	Dielectric strength (CC, NO)	350V AC
	Galvanic isolation	4kV AC
Relay	U_{max}	AC: 250V DC: 30V
	I	3A, max. 5A
	P_{max}	1250VA , 150W
Optoswitch	U_{max}	350V peak AC and DC
	$I_{max\ cont.}$	130mA
	$I_{max\ peak}$	400mA (100ms (1 shot), DC)
	P_{max}	500mW
	R_{on}	18...25Ω
	f_{max}	500Hz

7.4.7 Technical details about the integrated rechargeable buffer battery

The integrated rechargeable buffer battery enables a message to be issued in the event of a supply voltage failure. The alarm for the internal "voltage" measurement value (see "Internal channels" on page 139) must be configured for this purpose. The value that is entered for the "value low" alarm level is not important as the internal "voltage" measurement value is set to zero and the supply is switched to the rechargeable buffer battery as soon as the supply voltage falls below 9,5V . This initiates the malfunction message to be issued, as long as a time delay for issuing the malfunction message has not been entered via the "Delay for power off alarm" parameter in the "Basic settings" configuration section (see "Basic settings" on page 142). The switchover to the rechargeable buffer battery is definitely entered in the device log by the "BACKUP SUPPLY, 1" log entry. If a delay time has been configured, the myDatalogMUCmini , apart from the components that fail when supplied by the rechargeable buffer battery, continues to operate normally until the time delay has elapsed. If the supply voltage increases to above 10,5V again during the time delay, the supply is switched from the rechargeable buffer battery back to the supply voltage. This is recorded in the device log

with the "BACKUP SUPPLY, 0" log entry. Issuing the malfunction message is initiated if the supply voltage is below 10,5V when the time delay elapses.

The following components fail as soon as the myDatalogMUCmini is only supplied by the rechargeable buffer battery:

- Analogue output
- Control of the relay (i.e. the closed-circuit contact switches to "NO" idle mode). The control of the optoswitch is not affected by this.
- 5V supply to COM2

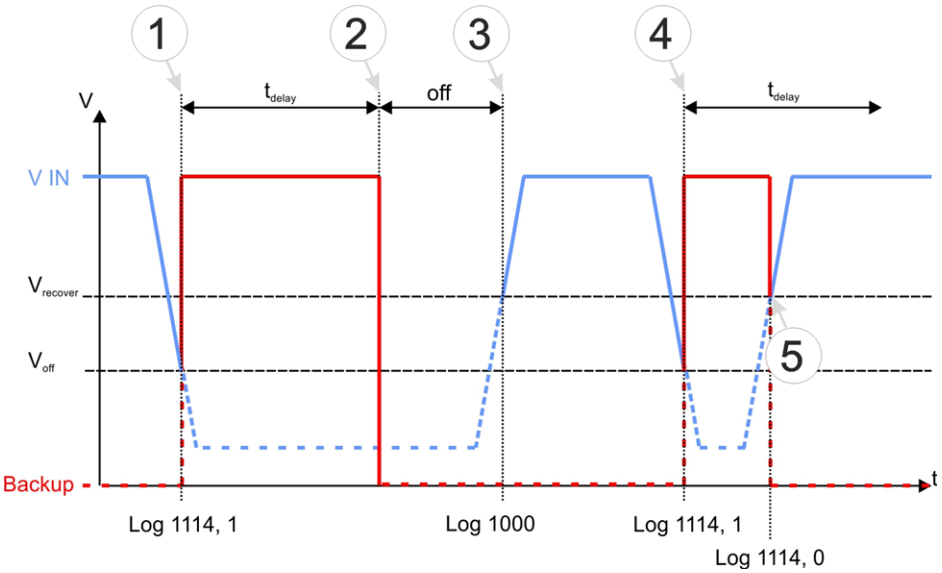
The device establishes a connection to the server to issue the malfunction message. Any outstanding data is also transmitted to the server during this process. The rechargeable buffer battery is then deactivated . If the supply voltage has not broken down completely at this time, the supply voltage is only monitored on the device. All of the other operations are terminated. This state is maintained until the supply fails completely or increases above 10,5V . The myDatalogMUCmini resumes normal operation above 10,5V .

Devices that have informed the myDatanet server about the failure of their supply voltage are indicated in the site list with the symbol pictured below.



A hardware-regulated controller ensures that the rechargeable buffer battery is only charged when the ambient temperature is in the permitted range (0...+40°C).

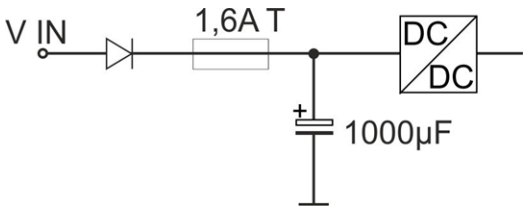
V IN	Supply voltage: 12...30VDC (+/- 10%)
$V_{recover}$	Threshold for switching to normal operation: 10,5V
V_{off}	Threshold for switching to supply via the rechargeable buffer battery: 9,5V
t_{delay}	Time delay for issuing the malfunction message. Is configured via the "Delay for power off alarm" parameter located in the "Basic settings" configuration section (see "Basic settings" on page 142).
Backup	Voltage of the rechargeable buffer battery



Supply of the myDatalogMUCmini

1	• The power supply is switched to the rechargeable buffer battery. • The "BACKUP SUPPLY, 1" log entry is created. • The device continues to operate normally (except for the components that fail when supplied by the rechargeable buffer battery).
2	• The time delay for issuing the malfunction message has elapsed. • The malfunction message is issued. The rechargeable buffer battery is then deactivated. The myDatalogMUCmini is subsequently switched off.
3	• The supply voltage exceeds the threshold for switching to normal operation again. • The "POWER ON" log entry is created. • The myDatalogMUCmini resumes normal operation again.
4	• The power supply is switched to the rechargeable buffer battery. • The "BACKUP SUPPLY, 1" log entry is created. • The device continues to operate normally (except for the components that fail when supplied by the rechargeable buffer battery).
5	• The supply voltage exceeds the threshold for switching to normal operation again. • The time delay for issuing the malfunction message has not elapsed yet. • The "BACKUP SUPPLY, 0" log entry is created. • NO malfunction message is issued.

7.4.8 Technical details regarding the energy supply



Schematic diagram of the energy supply

V IN	12...30VDC (+/- 10%)
Power consumption [1)] (without sensors)	typ. 1W max. 3W
Input capacity	1000µF
Fuse	1,6A T
Reverse voltage protection	Yes

[1)] applies to ongoing operation. A current peak is caused by the input capacity at the time of activation.

The myDatalogMUCmini is equipped with a relatively large input capacity (1000µF) to ensure reliable switchover to the integrated rechargeable buffer battery in the event of a supply voltage failure. When selecting the power supply please ensure that it is able to supply the required initial current. A selection of compatible power supplies is included in the chapter "Power supply" on page 311. The supply voltage input is also equipped with a diode to protect against polarity reversal and a 1,6A T fuse.

Chapter 8 Initial Start-Up

8.1 User information

Before you connect the myDatalogMUCmini and place it into operation, you must observe and comply with the following user information!

This manual contains all information that is required for using the device.

Is intended for technically qualified personnel who have the relevant knowledge and experience in the area of measurement technology.

Read this manual carefully and completely in order to ensure the proper functioning of the myDatalogMUCmini .

Please contact our technical department if anything is unclear or if you encounter difficulties with regard to installation, connection or configuration (see "Contact information" on page 319).

If necessary, please also refer to the manuals of the accessory parts when commissioning the complete system. These manuals are included in the scope of delivery of the accessory parts.

8.2 General principles

The entire measurement system may only be placed into operation after completion and inspection of the installation. Study the manual thoroughly before placing into operation to prevent faulty or incorrect configuration.

Utilise the manual to familiarise yourself with the operation of the myDatalogMUCmini and the input screens of the myDatanet-Server before you begin with the configuration.

8.3 Placing the system into operation

> ***Note:*** *It is recommended that the myDatalogMUCmini is first placed into operation in the office before mounting the device permanently at the place of use. During this process, you should set a site for the later operation on the myDatanet server (see "myDatanet Server Manual " 805002) and determine a site configuration (see "Site configuration" on page 81). Take the opportunity to get to know the functions of the device in a stable environment. You can also use suitable test signals to simulate the sensors to establish the optimum configuration of the myDatalogMUCmini prior to its actual first use. This reduces the amount of time required for on-site installation to a minimum.*

1. Once you have completed all the steps described in the chapter "Connecting the sensors, actuators and power supply" on page 60, the myDatalogMUCmini is ready for operation and should have made its first connection to the myDatanet server.
2. Create a site for the operation on the myDatanet server (see "myDatanet Server Manual " 805002).

3. Configure the created site according to your requirements (see "Site configuration" on page 81).
4. Link the myDatalogMUCmini with the created site (see "Site" on page 81).
5. Activate the ALOHA transmission mode (see "ALOHA transmission mode" on page 45) so that the site configuration is transmitted to the myDatalogMUCmini .

8.4 Testing communication with the device

1. Create a site for the operation on the myDatanet server (see "myDatanet Server Manual " 805002).
2. Configure the created site according to your requirements (see "Site configuration" on page 81).
3. Link the myDatalogMUCmini with the created site (see "Site" on page 81).
4. Initiate the ALOHA transmission mode (see "ALOHA transmission mode" on page 45) so that the configuration of the site is transmitted to the myDatalogMUCmini .
5. Wait until it is indicated in the list of measurement instruments that the device is in ALOHA transmission mode. This is indicated by a speech bubble with the "Aloha" inscription.



The following steps are only required if you also wish to test the measurement value acquisition and data transmission.

6. Stop the ALOHA transmission mode by clicking the cross in the speech bubble with the "Aloha" inscription or wait for the duration of the ALOHA transmission mode. This period can be set in the basic settings (see "Basic settings" on page 142) of the site configuration. The default setting is 10 minutes.
7. Then wire up the sensors (see "Connecting the sensors, actuators and power supply" on page 60) and restart the ALOHA transmission mode.

   Important note: *All wiring work must be performed in the de-energised state.*

8. Check the incoming data in the ALOHA data window of the myDatanet server, which can be accessed by clicking on the speech bubble with the "Aloha" inscription (see "myDatanet Server Manual " 805002). Particular attention must be paid to the internal "GSM level" and "voltage" measurement values.



Note: *Supplementary explanation on evaluating the "GSM level":*

"GSM level"	
›-64dBm	
-64 to -73dBm	
-74 to -83dBm	
-84 to -93dBm	
-94 to -107dBm	
‹= -108dBm	

Note: *Additional explanation about evaluating the "voltage":*

The displayed voltage should only deviate slightly from the supply voltage applied to the device.

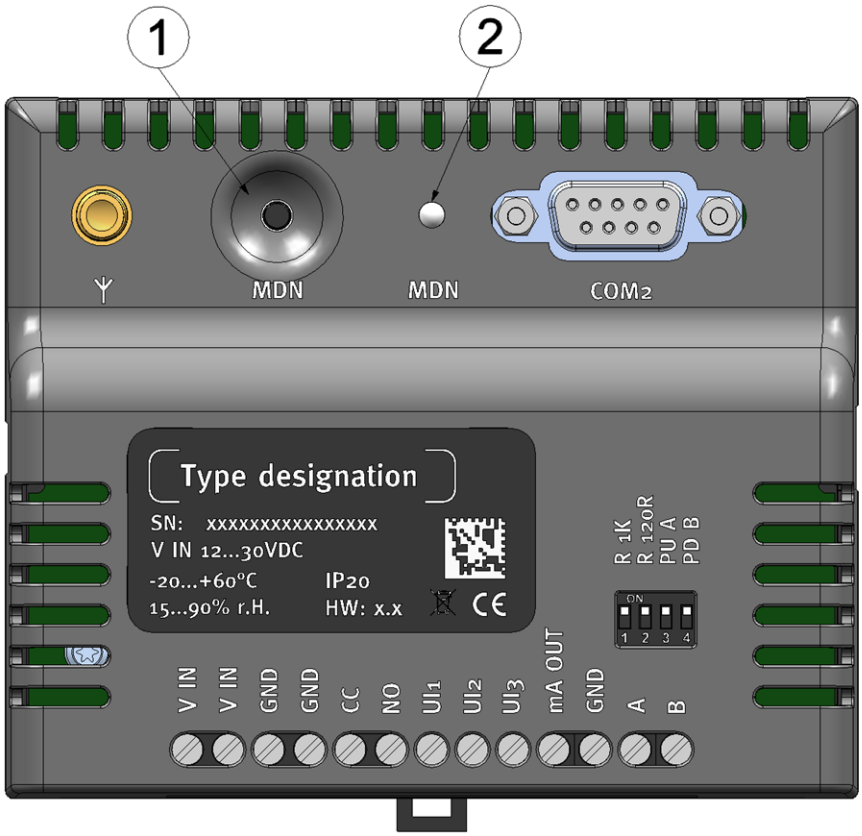
6872
6872
6872

Chapter 9 User interfaces

The configuration of the myDatalogMUCmini is carried out via the web interface on the myDatanet server (see "User interface on the myDatanet-Server" on page 81), which your responsible sales partner will provide to you.

9.1 User interface on the myDatalogMUCmini

9.1.1 Operating elements



Operating elements

1	Button to initiate ALOHA transmission mode	2	Status LED

9.1.1.1 Button to initiate ALOHA transmission mode

The button can be used to initiate the ALOHA transmission mode or to instruct the myDatalogMUCmini to immediately issue the error/status code.

User action	Device response	Operation after releasing the button
Press briefly for approx. one second	Status LED illuminates	Error/status code is displayed (see "Status LED" on page 80)
Press and hold for five seconds	Status LED flashes three times and then remains on	ALOHA transmission mode

9.1.1.2 Status LED

The status LED is used both to display the error/status codes and to indicate the current operating state. If the ALOHA transmission mode was activated or the power supply was provided (PowerOn), the status LED shows the current operating state for ten minutes. During these ten minutes, the error/status codes are transmitted every three seconds.

Error/status codes

Flashing code	Description	Solution/cause
0x	Transport lock (GPRS OFF, measurement OFF)	If the ALOHA transmission mode is initiated by the button , the myDatalogMUCmini switches back into "RUN" mode (GPRS ON, measurement ON).
1x	Last connection OK	---
2x	Last transmission faulty	Try again later
4x	Standby (GPRS ON, measurement OFF)	see "transport lock"
6x	Offline (GPRS OFF, measurement ON)	see "transport lock"
7x	Network block/no matching provider	• Improve position of the antenna • Check whether the device is located in the coverage area of a service provider that is supported by the integrated SIM chip (www.microtronics.at/footprint) • Lift the network block (see "myDatanetDeviceConfig Manual " 805004)
8x	No GSM network	• Try again later • Improve position of the antenna
10x	No GPRS connection	Improve position of the antenna
11x	No myDatanet server accessible	• Check whether port 51241 is enabled on the myDatanet server • Try again later
12x	Faulty SIM chip	Contact support

Operating states

Status LED	Description
Flickering	Establishing connection
Lights up	GPRS connection established or button actuated
Off	Normal measuring operation according to configuration until the next transmission

9.2 User interface on the myDatanet-Server

9.2.1 Site configuration

> ***Note:*** *Several of the configuration fields in the following sub chapters may possibly be hidden depending on the respective user level. In this case, please contact the myDatanet-Server administrator.*

Click on the name of the site in the list of sites to get to the input screen for configuring the site (see "myDatanet Server Manual " 805002).

9.2.1.1 Site

Customer
Specifies to which customer the site is assigned.

 **icon**
Assign site to another customer

Name
Site designation (not relevant for the device or data assignment)[2-50 characters]

Instrument S/N
Serial number of the device that is linked to the site (device assignment!)

9.2.1.2 Comments

Comments
Free comment field (is also displayed below the device type in the site list)

9.2.1.3 Control

Script type

Selection of the script type

Off	*Script deactivated*			
IL	*Activates script processing and informs the myDatalogMUCmini that the script is an "Instruction list" (see "Instruction list" on page 153)*			
	Script	*Input window for editing the script that is loaded in the device*		
Pawn	*Activates script processing and informs the myDatalogMUCmini that the script is a "Pawn script" (see "Pawn script" on page 167)*			
	Script source	*Script*	*The script is directly entered and compiled via the server interface.*	
			Script	*Input window for editing the script that is loaded in the device*
		Upload a compiled script	*A script binary file (*.amx), that has already been compiled, should be uploaded.*	
			File upload	*Selection of the script binary file (*.amx) that is uploaded to the myDatanet server and is loaded into the myDatalogMUCmini during the next connection. The file path is only displayed as long as the input screen for configuring the site has not been closed.*

9.2.1.4 Interfaces

9.2.1.4.1 Basis

Note: *Modbus mode (master/slave) for interface Com1 is selected in the "Config" tab (see "Config" on page 94).*

9.2.1.4.1.1 Com1 (modbus master, RS485, script parsing inactive)

Mode

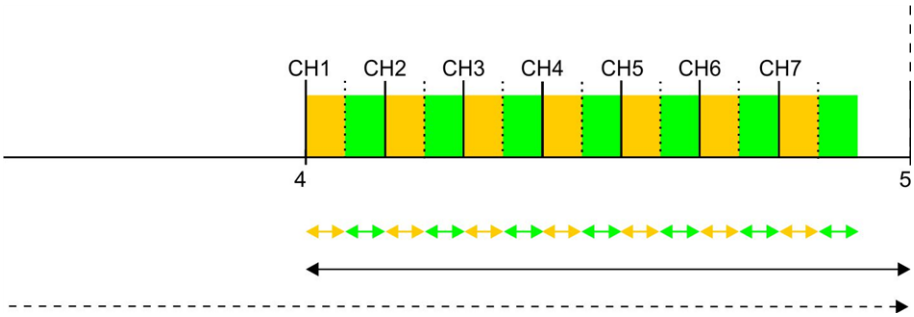
Off	*Interface deactivated*		
Modbus (1/2)	*activates the modbus functionality of the interface*		
	Baud rate	*300*	*Selection of the required baud rate*
		600	
		1200	
		2400	
		4800	
		9600	
		19200	
		38400	
		57600	
		115200	
	Stop bits	*1*	*Selection of the required stop bits*
		2	
	Parity	*None*	*Selection of the required parity*
		Odd	
		Even	
	Data bits	*7*	*Number of data bits to be used*
		8	
	Response timeout	*Time during which the modbus slave must react to the command from the device*	

Modbus *(2/2)*	*Hold*	*Hold the last valid measurement value for x measurement cycles*	
		Off	*Function deactivated*
		1-5	*Number of measurement cycles for which the measurement value is held until the error value is issued*
		On	*In the event of an error, the last valid measurement value is held until a new valid measurement value is present.*
	Retry	*In the event of a communication error, the corresponding command is repeated once. The error is only indicated if this second attempt also fails.*	
Script parsing	*Activates script parsing. This means that the interface can be accessed via the "Mdn_SerialEvent()", "Mdn_SerialRx()", "Mdn_SerialTx()" and "Mdn_ SerialFinish()" PAWN script functions (see "Script parsing" on page 48). This deactivates the modbus functionality of the interface.* *Details on configuring the "Script Parsing" are provided in "Com1 (RS485, script parsing active)" on page 87.*		

Important note:

Example to explain the link between the "Response timeout", "Retry" and "Measurement cycle"

Basic setting		*Record interval*	*5min.*
		Measurement cycle	*1min.*
Interfaces -› Basis	*First attempt*	*Response timeout*	*4sec.*
	Retry		
		Retry	*Active*



Explanation: *In this example, seven channels are activated, however, the modbus slave does not answer. At the time of a measurement, an initial attempt is made to read the data for interface channel 1. As "Retry" is activated, another attempt is made to read the data for interface channel 1 from the modbus slave following expiry of the "Response timeout". Once the "Response timeout" has expired again, the error value "OL" (open loop) is set for interface channel 1 and the first attempt is made to read the data for interface channel 2.*

Therefore, the number of activated channels (interface channels + interface output channels), the "Response timeout", "Retry" and measurement cycle must be selected as follows:

Retry not active: "Response timeout" * number of active channels ‹ "Measurement cycle"

Retry active: "Response timeout" * 2 * number of active channels ‹ "Measurement cycle"

9.2.1.4.1.2 Com1 (modbus slave, RS485, script parsing inactive)

Mode

Off	*Interface deactivated*		
Modbus	*activates the modbus functionality of the interface*		
	Baud rate	*300*	*Selection of the required baud rate*
		600	
		1200	
		2400	
		4800	
		9600	
		19200	
		38400	
		57600	
		115200	
	Stop bits	*1*	*Selection of the required stop bits*
		2	
	Parity	*None*	*Selection of the required parity*
		Odd	
		Even	
	Data bits	*7*	*Number of data bits to be used*
		8	
	Monitoring timeout	*0...no monitoring* *Time during which the registers have to be renewed by the modbus master. The measurement values are labelled as "OL" (open loop) in the event of a timeout.*	
	Hold	*Hold the last valid measurement value for x measurement cycles*	
		Off	*Function deactivated*
		1-5	*Number of measurement cycles for which the measurement value is held until the error value is issued*
		On	*In the event of an error, the last valid measurement value is held until a new valid measurement value is present.*

Script parsing	*Activates script parsing. This means that the interface can be accessed via the "Mdn_SerialEvent()", "Mdn_SerialRx()", "Mdn_SerialTx()" and "Mdn_SerialFinish()" PAWN script functions (see "Script parsing" on page 48). This deactivates the modbus functionality of the interface.* *Details on configuring the "Script Parsing" are provided in "Com1 (RS485, script parsing active)" on page 87.*

9.2.1.4.1.3 Com1 (RS485, script parsing active)

Mode

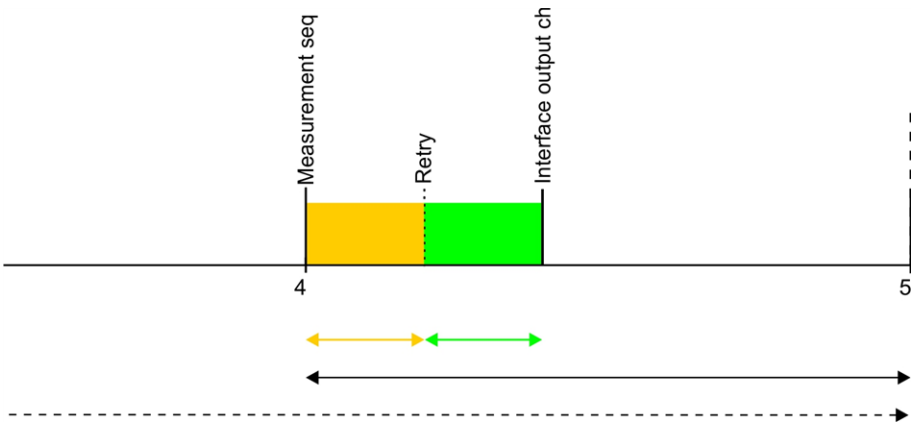
Off	*Interface deactivated*		
Modbus	*activates the modbus functionality of the interface* *Details on configuring "Modbus" mode are provided in "Basis" on page 83 or "Com1 (modbus slave, RS485, script parsing inactive)" on page 86.*		
Script parsing *(1/2)*	*Activates script parsing. This means that the interface can be accessed via the "Mdn_SerialEvent()", "Mdn_SerialRx()", "Mdn_SerialTx()" and "Mdn_SerialFinish()" PAWN script functions (see "Script parsing" on page 48). This deactivates the modbus functionality of the interface.*		
	Baud rate	*300*	*Selection of the required baud rate*
		600	
		1200	
		2400	
		4800	
		9600	
		19200	
		38400	
		57600	
		115200	
	Stop bits	*1*	*Selection of the required stop bits*
		2	

Script parsing (2/2)	*Parity*	*None*	*Selection of the required parity*
		Odd	
		Even	
	Data bits	*7*	*Number of data bits to be used*
		8	
	Frame timeout	*0...no timeout* *Time during which the connected digital sensor must send the full answer to a request. A communication error is detected if the time is exceeded.*	
	Hold	*Hold the last valid measurement value for x measurement cycles*	
		Off	*Function deactivated*
		1-5	*Number of measurement cycles for which the measurement value is held until the error value is issued*
		On	*In the event of an error, the last valid measurement value is held until a new valid measurement value is present.*
	Retry	*In the event of a communication error, the corresponding command is repeated once. The error is only indicated if this second attempt also fails.*	

Important note:

Example to explain the link between the "Frame timeout", "Retry" and "Measurement cycle"

Basic setting		*Record interval*	*5min.*
		Measurement cycle	*1min.*
Interfaces -› Basis	*First attempt*	*Frame timeout*	*12sec.*
	Retry		
		Retry	*Active*



Explanation: *At the time of a measurement, the measurement data sequence is sent via the "Mdn_SerialTx()" function (see "Script parsing" on page 48), although the sensor does not answer. As "Retry" is activated, the measurement data sequence is sent again once the "Frame timeout" has expired. If the "Frame timeout" expires again, the error value "OL" (open loop) is set for all of the interface channels that obtain their data via the relevant Com by means of the "Mdn_SetCh()" function (see Mdn_SetCh()). Afterwards, the data record that contains the setpoints of the interface output channels is sent via the interface by means of the "Mdn_SerialTx()" function.*

The "Frame timeout", "Retry" and measurement cycle must therefore be selected as follows:

Retry not active: "Response timeout" ‹ "Measurement cycle"

Retry active: "Response timeout" * 2 ‹ "Measurement cycle"

9.2.1.4.1.4 Com2 (serial, RS232)

Mode

Off	*Interface deactivated*		
ASCII (1/2)	*Baud rate*	*300*	*Selection of the required baud rate*
		600	
		1200	
		2400	
		4800	
		9600	
		19200	
		38400	
		57600	
		115200	
	Stop bits	*1*	*Selection of the required stop bits*
		2	
	Parity	*None*	*Selection of the required parity*
		Odd	
		Even	
	Data bits	*7*	*Number of data bits to be used*
		8	
	Frame timeout	*0...no timeout* *Time during which the connected digital sensor must send the full answer to a request. A communication error is detected if the time is exceeded.*	

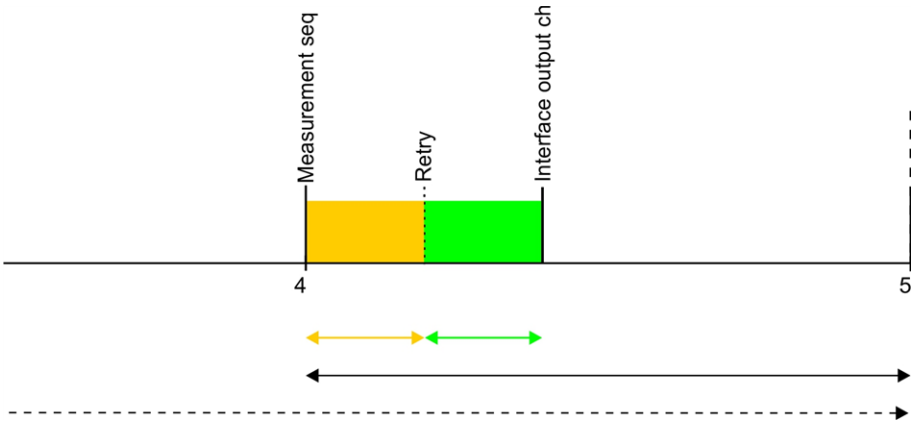
ASCII (2/2)	*Hold*	*Hold the last valid measurement value for x measurement cycles*	
		Off	*Function deactivated*
		1-5	*Number of measurement cycles for which the measurement value is held until the error value is issued*
		On	*In the event of an error, the last valid measurement value is held until a new valid measurement value is present.*
	Retry	*In the event of a communication error, the corresponding command is repeated once. The error is only indicated if this second attempt also fails.*	
Script parsing (1/2)	*Activates script parsing. This means that the interface can be accessed via the "Mdn_SerialEvent()", "Mdn_SerialRx()", "Mdn_SerialTx()" and "Mdn_SerialFinish()" PAWN script functions (see "Script parsing" on page 48). The option to determine via the "Config" tab of the "Interfaces" configuration section how the received characters should be evaluated is thus deactivated.*		
	Baud rate	*300*	*Selection of the required baud rate*
		600	
		1200	
		2400	
		4800	
		9600	
		19200	
		38400	
		57600	
		115200	
	Stop bits	*1*	*Selection of the required stop bits*
		2	
	Parity	*None*	*Selection of the required parity*
		Odd	
		Even	
	Data bits	*7*	*Number of data bits to be used*
		8	

Script parsing *(2/2)*	*Frame timeout*	*0...no timeout* *Time during which the connected digital sensor must send the full answer to a request. A communication error is detected if the time is exceeded.*	
	Hold	*Off*	*Function deactivated*
		1-5	*Number of measurement cycles for which the measurement value is held until the error value is issued*
		On	*In the event of an error, the last valid measurement value is held until a new valid measurement value is present.*
	Retry	*In the event of a communication error, the corresponding command is repeated once. The error is only indicated if this second attempt also fails.*	

Important note:

Example to explain the link between the "Frame timeout", "Retry" and "Measurement cycle"

Basic setting		*Record interval*	*5min.*
		Measurement cycle	*1min.*
Interfaces -› Basis	*First attempt*	*Frame timeout*	*12sec.*
	Retry		
		Retry	*Active*



Explanation: *At the time of the measurement, the measurement data sequence is sent (see "Sequences for COM2" on page 95), although the sensor does not answer. As "Retry" is activated, the measurement data sequence is sent again once the "Frame timeout" has expired. If the "Frame timeout" expires again, the error value "OL" (open loop) is set for all of the interface channels that obtain their data via Com2. The data record that contains the setpoints of the interface output channels is then sent.*

The "Frame timeout", "Retry" and measurement cycle must therefore be selected as follows:

Retry not active: "Response timeout" ‹ "Measurement cycle"

Retry active: "Response timeout" * 2 ‹ "Measurement cycle"

9.2.1.4.2 Config

9.2.1.4.2.1 Com1

Mode

Off	*Interface deactivated*				
Modbus	*Modbus mode*	*Master*	*Com mode*	*RTU*	*Data is transferred in binary format.*
				ASCII	*Data is transferred in ASCII format.*
		Slave	*Com mode*	*RTU*	*Data is transferred in binary format.*
				ASCII	*Data is transferred in ASCII format.*
			Slave Ad.	*Modbus slave address*	
Script Parsing	*---*				

9.2.1.4.2.2 Com2

Mode

Off	*Interface deactivated*		
ASCII	*Warmup time*	*Time from switching on the myDatalogMUCmini until the init sequence is sent.*	
	Frame structure	*...[CR][LF]*	*Each frame must be terminated with both "[CR][LF]" characters.*
		...[ETX]	*Each frame must be terminated with the "[ETX]" character.*
		[STX]...[CR][LF]	*Each frame must start with the "[STX]" character and terminate with both of the "[CR][LF]" characters.*
		[STX]...[ETX]	*Each frame must start with the "[STX]" character and terminate with the "[ETX]" character.*
	Character timeout	*0...no timeout* *Maximum permissible time interval between the full receipt of two characters. A communication error is detected if the time is exceeded. The monitoring process starts following receipt of the first character of an answer from the digital sensor.*	
	Number delimiter	*Divides the ASCII data message of the digital sensor into individual measurement values. Up to ten different delimiters can be entered via the input field. The following escape codes are also supported as delimiters:* *\a \b \f \n \r \t \v \\ \? \' \" \xhh* *The maximum number of delimiters is reduced when escape codes are used.*	
	Number format	*1,000.00*	*"," is used as the thousand separator* *"." is used as the decimal point*
		1.000,00	*"." is used as the thousand separator* *"," is used as the decimal point*
Script Parsing	*Warmup time*	*Time from switching on the myDatalogMUCmini until the init sequence is sent.*	

9.2.1.5 Sequences for COM2

Note: *This configuration section is only visible if "ASCII" mode is activated for the COM2 interface in the "Interfaces" configuration section (see "Basis" on page 83). If "Script parsing" mode is activated, the init sequence and measurement data sequence must be generated by the PAWN script.*

Init sequence

Initialisation command that is sent to the digital sensor following expiry of the warmup time when the myDatalogMUCmini is switched on.

The following escape codes are supported:

\a \b \f \n \r \t \v \\ \? \' \" \xhh

Measurement data sequence

Command that is sent to the digital sensor at the time of every measurement [0-512 characters]

The following escape codes are supported:

\a \b \f \n \r \t \v \\ \? \' \" \xhh

9.2.1.6 Measurement channels

9.2.1.6.1 Basis

Title 1-3

Freely selectable channel title for the universal inputs

Mode

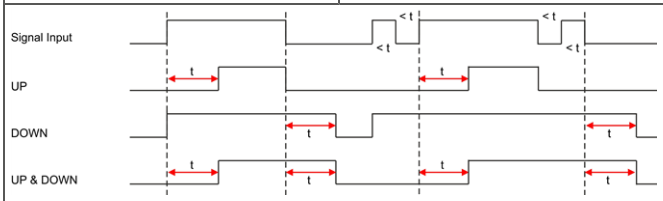
Basic settings for the measurement channel

Universal inputs *(digital modes)*	*Off*	*---*	*Measurement channel deactivated*
	Digital	*Invert*	*Inverts the input signal*
	Cnt.Day	*Impulse*	*Metered measurand of a pulse in the measurement unit*
		Unit	*String that is used as a measurement unit by all of the server display elements [0-16 characters]*
		Decimal places	*Number of decimal places that are used by all of the server display elements*
	Cnt.Intervl.	*Impulse*	*Metered measurand of a pulse in the measurement unit*
		Unit	*String that is used as a measurement unit by all of the server display elements [0-16 characters]*
		Decimal places	*Number of decimal places that are used by all of the server display elements*
	Freq	*Factor*	*Factor by which the input signal is multiplied*
		Unit	*String that is used as a measurement unit by all of the server display elements [0-16 characters]*
		Decimal places	*Number of decimal places that are used by all of the server display elements*
	PWM	*0%*	*Start of the measurement range in the measurement unit*
		100%	*End of the measurement range in the measurement unit*
		Unit	*String that is used as a measurement unit by all of the server display elements [0-16 characters]*
		Decimal places	*Number of decimal places that are used by all of the server display elements*

Universal inputs *(analogue modes)*	*0-20 mA*	*0%*	*Start of the measurement range in the measurement unit*
		100%	*End of the measurement range in the measurement unit*
		Unit	*String that is used as a measurement unit by all of the server display elements [0-16 characters]*
		Decimal places	*Number of decimal places that are used by all of the server display elements*
	4-20 mA	*0%*	*Start of the measurement range in the measurement unit*
		100%	*End of the measurement range in the measurement unit*
		Unit	*String that is used as a measurement unit by all of the server display elements [0-16 characters]*
		Decimal places	*Number of decimal places that are used by all of the server display elements*
	0-2V	*0%*	*Start of the measurement range in the measurement unit*
		100%	*End of the measurement range in the measurement unit*
		Unit	*String that is used as a measurement unit by all of the server display elements [0-16 characters]*
		Decimal places	*Number of decimal places that are used by all of the server display elements*
	0-10V	*0%*	*Start of the measurement range in the measurement unit*
		100%	*End of the measurement range in the measurement unit*
		Unit	*String that is used as a measurement unit by all of the server display elements [0-16 characters]*
		Decimal places	*Number of decimal places that are used by all of the server display elements*

9.2.1.6.2 Config

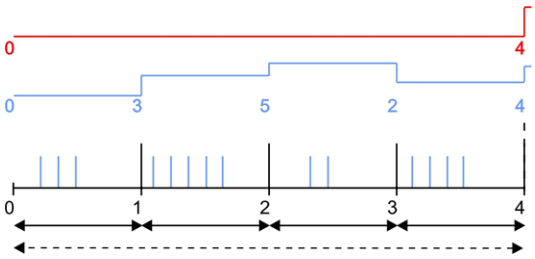
<table>
<tr><td rowspan="8">Universal inputs

(digital modes)</td><td>Off</td><td>---</td><td colspan="2">---</td></tr>
<tr><td rowspan="7">Digital</td><td>Filter time</td><td colspan="2">Time during which the signal must remain constant to initiate a level change. Used to suppress brief faults (debouncing).</td></tr>
<tr><td rowspan="5">Decay</td><td colspan="2">Temporal function in the measurement cycle</td></tr>
<tr><td>Off</td><td>Decay deactivated</td></tr>
<tr><td>up</td><td>Minimum signal length for x seconds with a rising edge</td></tr>
<tr><td>down</td><td>Minimum signal length for x seconds with a falling edge</td></tr>
<tr><td>up&down</td><td>Minimum signal length for x seconds on both edges</td></tr>
<tr><td colspan="2">
</td></tr>
<tr><td>Time</td><td colspan="2">Time x, that is used for decay modes "up", "down" and "up & down"</td></tr>
</table>

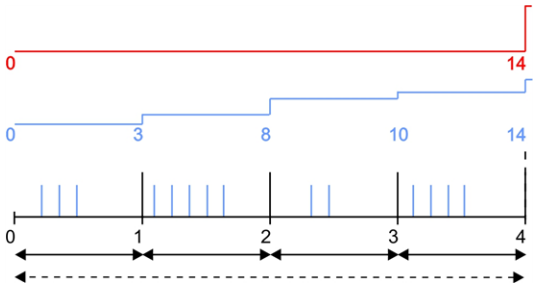
Universal inputs (Counter modes)	Cnt.Day	Filter time	Time during which the signal must remain constant to initiate a level change. Used to suppress brief faults (debouncing).	
		Reset at	Reset time of the day counter	
	Cnt.Intervl.	Filter time	Time during which the signal must remain constant to initiate a level change. Used to suppress brief faults (debouncing).	
		Decay	Temporal function in the measurement cycle	
			Off	Decay deactivated
			min	The minimum of the last x measurement values is recorded.
			max	The maximum of the last x measurement values is recorded.
			avg	The arithmetic mean of the last x measurement values is recorded.
			med	The median of the last x measurement values is recorded.
			rms	The root mean square of the last x measurement values is recorded.
		Time	Decay period. To calculate the number of considered measurement values see "Example to clarify the record, measurement and burst interval (in conjunction with the decay)" on page 143. The "DECAY MEM ERR" error is entered in the device log if no temporary memory could be reserved to take another measurement value into consideration (see "Log entries and error codes " on page 302).	

Note: *Additional explanation regarding the difference between "Cnt.Day" and "Cnt.Intervl."*

Basic setting	◄- - - - - -►	*Record interval*	*4 min.*	*Recorded value*	*red line*
	◄——►	*Measurement cycle*	*1 min.*	*Measurement value*	*blue line*



"Cnt.Intervl." mode: The pulses are added up and reset every time a measurement value is generated.



"Cnt.Day" mode: All of the pulses up to the reset time are added up.

Note on "Cnt.Intervl." mode: *If the record interval is larger than the measurement cycle, only the number of pulses registered the last time the measurement value was generated is recorded. In this case, the individual measurement values can be summarised by means of the decay.*

Universal inputs (Frequency mode 1/2)	Freq (1/2)	Filter time	Time during which the signal must remain constant to initiate a level change. Used to suppress brief faults (debouncing).	
		Decay	Temporal function in the measurement cycle	
			Off	Decay deactivated
			min	The minimum of the last x measurement values is recorded.
			max	The maximum of the last x measurement values is recorded.
			avg	The arithmetic mean of the last x measurement values is recorded.
			med	The median of the last x measurement values is recorded.
			rms	The root mean square of the last x measurement values is recorded.
		Time	Decay period. To calculate the number of considered measurement values see "Example to clarify the record, measurement and burst interval (in conjunction with the decay)" on page 143. The "DECAY MEM ERR" error is entered in the device log if no temporary memory could be reserved to take another measurement value into consideration (see "Log entries and error codes " on page 302).	
		Hold	Hold the last valid measurement value for x measurement cycles	
			Off	Function deactivated
			1-5	Number of measurement cycles for which the measurement value is held until the error value is issued
			On	In the event of an error, the last valid measurement value is held until a new valid measurement value is present.

Universal inputs (Frequency mode 2/2)	Freq (2/2)	Overflow	Procedure in the event of measurement range violations	
			Ignore	The measurement value is calculated beyond the range limits.
			Silent cutoff	The measurement value is truncated at the range limits.
			Out of range	• If the measurement value is below 1Hz, the error value "UF" (underflow) is issued. • The error value "OF" (overflow) is issued, if the measurement value is above 1000Hz.
			NAMUR borders	• If the measurement value is below 1Hz, the error value "UF" (underflow) is issued. • The error value "OF" (overflow) is issued, if the measurement value is above 1000Hz.

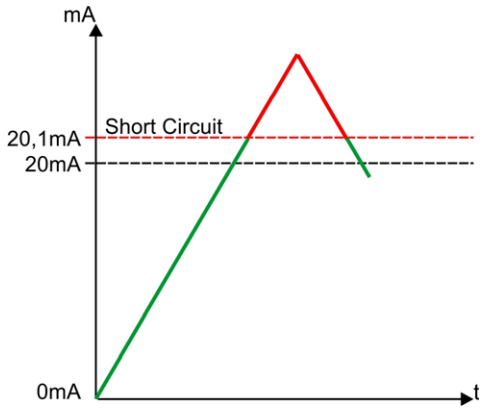
Universal inputs (PWM mode 1/2)	PWM (1/2)	Filter time	Time during which the signal must remain constant to initiate a level change. Used to suppress brief faults (debouncing).	
		Decay	Temporal function in the measurement cycle	
			Off	Decay deactivated
			min	The minimum of the last x measurement values is recorded.
			max	The maximum of the last x measurement values is recorded.
			avg	The arithmetic mean of the last x measurement values is recorded.
			med	The median of the last x measurement values is recorded.
			rms	The root mean square of the last x measurement values is recorded.
		Time	Decay period. To calculate the number of considered measurement values see "Example to clarify the record, measurement and burst interval (in conjunction with the decay)" on page 143. The "DECAY MEM ERR" error is entered in the device log if no temporary memory could be reserved to take another measurement value into consideration (see "Log entries and error codes " on page 302).	
		Hold	Hold the last valid measurement value for x measurement cycles	
			Off	Function deactivated
			1-5	Number of measurement cycles for which the measurement value is held until the error value is issued
			On	In the event of an error, the last valid measurement value is held until a new valid measurement value is present.

Universal inputs (PWM mode 2/2)	PWM (2/2)	Overflow	Procedure in the event of measurement range violations	
			Ignore	The measurement value is calculated beyond the range limits.
			Silent cutoff	The measurement value is truncated at the range limits.
			Out of range	• If the measurement value is below 1%, the error value "UF" (underflow) is issued. • The error value "OF" (overflow) is issued, if the measurement value is above 99%.
			NAMUR borders	• If the measurement value is below 1%, the error value "UF" (underflow) is issued. • The error value "OF" (overflow) is issued, if the measurement value is above 99%.

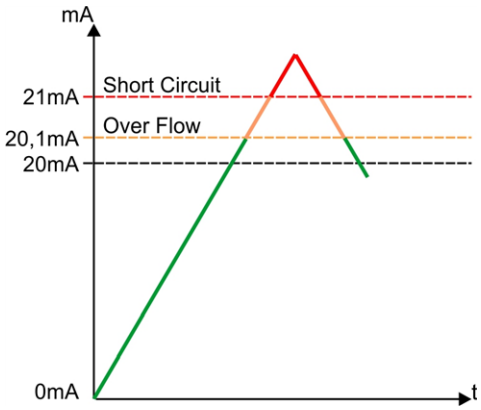
Universal inputs (0-20mA mode 1/2)	0-20mA (1/2)	Filter time	Time during which the analogue signal is averaged for signal smoothing. Used to suppress signal noise (also see "Output channels" on page 133).	
		Decay	Temporal function in the measurement cycle	
			Off	Decay deactivated
			min	The minimum of the last x measurement values is recorded.
			max	The maximum of the last x measurement values is recorded.
			avg	The arithmetic mean of the last x measurement values is recorded.
			med	The median of the last x measurement values is recorded.
			rms	The root mean square of the last x measurement values is recorded.
		Time	Decay period. To calculate the number of considered measurement values see "Example to clarify the record, measurement and burst interval (in conjunction with the decay)" on page 143. The "DECAY MEM ERR" error is entered in the device log if no temporary memory could be reserved to take another measurement value into consideration (see "Log entries and error codes " on page 302).	
		Hold	Hold the last valid measurement value for x measurement cycles	
			Off	Function deactivated
			1-5	Number of measurement cycles for which the measurement value is held until the error value is issued
			On	In the event of an error, the last valid measurement value is held until a new valid measurement value is present.

Universal inputs (0-20mA mode 2/2)	0-20mA (2/2)	Overflow	Procedure in the event of measurement range violations	
			Ignore	The measurement value is calculated beyond the range limits.
			Silent cutoff	The measurement value is truncated at the range limits.
			Out of range	• If the measurement value is above 20.1 mA, the error value "SC" (short circuit) is issued.
			NAMUR borders	• The error value "OF" (overflow) is issued, if the measurement value is above 20.1mA. • If the measurement value is above 21mA, the error value "SC" (short circuit) is issued.

Note: *Additional explanation regarding the "Out of range" and "Namur borders" modes.*



"Out of range" mode

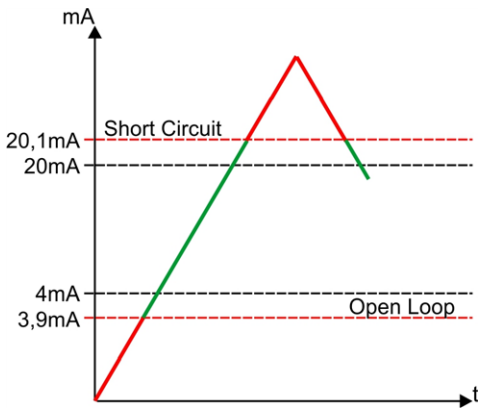


"Namur borders" mode

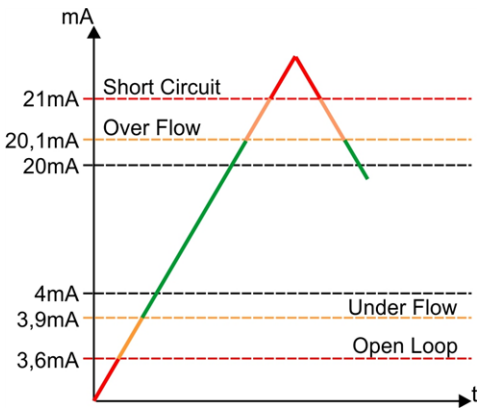
Universal inputs (4-20mA mode 1/2)	4-20 mA (1/2)	Filter time	Time during which the analogue signal is averaged for signal smoothing. Used to suppress signal noise (also see "Output channels" on page 133).	
		Decay	Temporal function in the measurement cycle	
			Off	Decay deactivated
			min	The minimum of the last x measurement values is recorded.
			max	The maximum of the last x measurement values is recorded.
			avg	The arithmetic mean of the last x measurement values is recorded.
			med	The median of the last x measurement values is recorded.
			rms	The root mean square of the last x measurement values is recorded.
		Time	Decay period. To calculate the number of considered measurement values see "Example to clarify the record, measurement and burst interval (in conjunction with the decay)" on page 143. The "DECAY MEM ERR" error is entered in the device log if no temporary memory could be reserved to take another measurement value into consideration (see "Log entries and error codes " on page 302).	
		Hold	Hold the last valid measurement value for x measurement cycles	
			Off	Function deactivated
			1-5	Number of measurement cycles for which the measurement value is held until the error value is issued
			On	In the event of an error, the last valid measurement value is held until a new valid measurement value is present.

Universal inputs (4-20mA mode 2/2)	4-20mA (2/2)	Overflow	Procedure in the event of measurement range violations	
			Ignore	The measurement value is calculated beyond the range limits.
			Silent cutoff	The measurement value is truncated at the range limits.
			Out of range	• The error value "OL" (open loop) is issued, if the measurement value is below 3.9mA. • If the measurement value is above 20.1 mA, the error value "SC" (short circuit) is issued.
			NAMUR borders	• The error value "OL" (open loop) is issued, if the measurement value is below 3.6mA. • The error value "UF" (underflow) is issued, if the measurement value is below 3.9mA. • The error value "OF" (overflow) is issued, if the measurement value is above 20.1mA. • If the measurement value is above 21mA, the error value "SC" (short circuit) is issued.

Note: *Additional explanation regarding the "Out of range" and "Namur borders" modes.*



"Out of range" mode

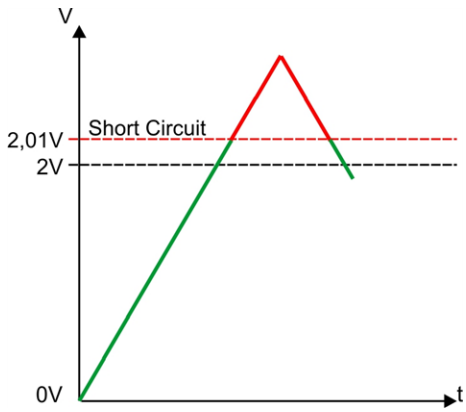


"Namur borders" mode

Universal inputs (0-2V mode 1/2)	0-2V (1/2)	Filter time	Time during which the analogue signal is averaged for signal smoothing. Used to suppress signal noise (also see "Output channels" on page 133).	
		Decay	Temporal function in the measurement cycle	
			Off	Decay deactivated
			min	The minimum of the last x measurement values is recorded.
			max	The maximum of the last x measurement values is recorded.
			avg	The arithmetic mean of the last x measurement values is recorded.
			med	The median of the last x measurement values is recorded.
			rms	The root mean square of the last x measurement values is recorded.
		Time	Decay period. To calculate the number of considered measurement values see "Example to clarify the record, measurement and burst interval (in conjunction with the decay)" on page 143. The "DECAY MEM ERR" error is entered in the device log if no temporary memory could be reserved to take another measurement value into consideration (see "Log entries and error codes " on page 302).	
		Hold	Hold the last valid measurement value for x measurement cycles	
			Off	Function deactivated
			1-5	Number of measurement cycles for which the measurement value is held until the error value is issued
			On	In the event of an error, the last valid measurement value is held until a new valid measurement value is present.

Universal inputs (0-2V mode 2/2)	0-2V (2/2)	Overflow	Procedure in the event of measurement range violations	
			Ignore	The measurement value is calculated beyond the range limits.
			Silent cutoff	The measurement value is truncated at the range limits.
			Out of range	The error value "OF" (overflow) is issued, if the measurement value is above 2.01V.
			NAMUR borders	The error value "OF" (overflow) is issued, if the measurement value is above 2.01V.

Note: *Additional explanation regarding the "Out of range" and "Namur borders" modes.*

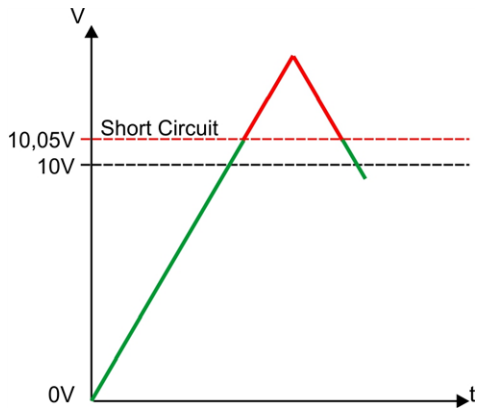


"Out of range" or "Namur borders" mode

Universal inputs (0-10V mode 1/2)	0-10V (1/2)	Filter time	Time during which the analogue signal is averaged for signal smoothing. Used to suppress signal noise (also see "Output channels" on page 133).	
		Decay	Temporal function in the measurement cycle	
			Off	Decay deactivated
			min	The minimum of the last x measurement values is recorded.
			max	The maximum of the last x measurement values is recorded.
			avg	The arithmetic mean of the last x measurement values is recorded.
			med	The median of the last x measurement values is recorded.
			rms	The root mean square of the last x measurement values is recorded.
		Time	Decay period. To calculate the number of considered measurement values see "Example to clarify the record, measurement and burst interval (in conjunction with the decay)" on page 143. The "DECAY MEM ERR" error is entered in the device log if no temporary memory could be reserved to take another measurement value into consideration (see "Log entries and error codes " on page 302).	
		Hold	Hold the last valid measurement value for x measurement cycles	
			Off	Function deactivated
			1-5	Number of measurement cycles for which the measurement value is held until the error value is issued
			On	In the event of an error, the last valid measurement value is held until a new valid measurement value is present.

Universal inputs (0-10V mode 2/2)	0-10V (2/2)	Overflow	Procedure in the event of measurement range violations	
			Ignore	The measurement value is calculated beyond the range limits.
			Silent cutoff	The measurement value is truncated at the range limits.
			Out of range	The error value "OF" (overflow) is issued, if the measurement value is above 10.05V.
			NAMUR borders	The error value "OF" (overflow) is issued, if the measurement value is above 10.05V.

Note: *Additional explanation regarding the "Out of range" and "Namur borders" modes.*



"Out of range" or "Namur borders" mode

9.2.1.6.3 Alarms

"Digital" mode	WA	A "high" at the universal input triggers a "warning".	
	AL	A "high" at the universal input triggers an "alarm".	
	SW	A "high" at the universal input triggers a "fault warning".	
	SA	A "high" at the universal input triggers a "fault alarm".	
All other modes	Warning	Value low	A warning is triggered, if the measurement value drops to or below this value.
		Value high	A warning is triggered, if the measurement value meets or exceeds this value.
	Alarm	Value low	An alarm is triggered, if the measurement value drops to or below this value.
		Value high	An alarm is triggered, if the measurement value meets or exceeds this value.
	Hyst %	Hysteresis for the all-clear in the event of an alarm/warning (e.g. Hyst=5%, alarm or warning at 100 -› all-clear at 95) or reset the trigger (e.g. Hyst=5%, level = greater or equal, trigger at 100 -› reset at 95)	

9.2.1.6.4 Trigger

If an universal input is operated in digital mode , there are two different types of triggers:

- Event trigger (MS, XM, I1-I4)

  In contrast to the level triggers, the relevant operation (e.g. start measurement) is only executed once when the trigger event occurs. With the help of the "Edge" configuration parameter, it is specified whether the rising, falling or both edges initiate the trigger event.

- Level trigger (QU, SL, ON)

  A "high" at the universal input initiates the trigger. A "low" at the universal input resets the trigger. The relevant operation (e.g. fast recording) is executed as long as the trigger is active. The option selected via the "Edge" configuration parameter is not relevant to the level trigger. If it is necessary for the trigger to be initiated by a "low" at the universal input and for it to be reset by a "high", the input signal must then be inverted using the "Invert" configuration parameter located in the "Basis" tab.

"Digital" mode	Event trigger	MS	Start measurement cycle immediately	
		XM	Initiate transmission	
		I1	Internal use with the control program	
		I2	Internal use with the control program	
		I3	Internal use with the control program	
		I4	Internal use with the control program	
		Edges	Selection of the edge at which the trigger should be initiated	
			rising	The rising edge initiates the trigger.
			falling	The falling edge initiates the trigger.
			both	Both edges initiate the trigger.
	Level trigger	QU	Fastrecording (record interval = record interval / factor)	
		SL	Slow recording (record interval = record interval * factor)	
		ON	Activate online mode	
		Edges	not relevant for the level triggers	
All other modes		QU	Fastrecording (record interval = record interval / factor)	
		SL	Slow recording (record interval = record interval * factor)	
		RO	Switch on recording	
		RF	Switch off recording	
		XM	Initiate transmission	
		ON	Activate online mode	
		I1	Internal use with the control program	
		I2	Internal use with the control program	
		I3	Internal use with the control program	
		I4	Internal use with the control program	
		Level	Levels for initiating the trigger. The hysteresis from the "Alarm" tab is used to determine the level to reset the trigger.	
			Greater or equal	The measurement value must be greater or equal to the level to initiate the trigger.
			Less or equal	The measurement value must be less or equal to the level to initiate the trigger.

9.2.1.7 Interface channels 1-32

Note: *Some of the configuration parameters included in this configuration section also affect parameters that are not included in the tab that is currently open. In the following section, this is indicated by the "›" symbol, for example "Basis › Min" means that the "Min" configuration parameter is located in the "Basis" tab.*

An interface channel is designed to record a single measurement value that is read from one of the 2 serial interfaces (1 x RS485, 1 x RS232). The data type that should be used and the interface, from which the data is read, can be determined for every interface channel independently of the other channels. The following chapters contain a description on configuring the interface channels.

9.2.1.7.1 Basis

Title 1-32

Freely selectable channel title for the interface channels [0-16 characters]

Interface

Basis settings for the measurement channel

Off	*---*	*Measurement channel deactivated*		
Com1	*Config › Format* *(see "Config." on page 117)*	*Digital*	*---*	
		Signed	*Unit*	*String that is used as a measurement unit by all of the server display elements [0-16 characters]*
			Decimal places	*Number of decimal places that are used by all of the server display elements*
		Unsigned	*Unit*	*String that is used as a measurement unit by all of the server display elements [0-16 characters]*
			Decimal places	*Number of decimal places that are used by all of the server display elements*
		Float	*Unit*	*String that is used as a measurement unit by all of the server display elements [0-16 characters]*
			Decimal places	*Number of decimal places that are used by all of the server display elements*
Com2	*Unit*	*String that is used as a measurement unit by all of the server display elements [0-16 characters]*		
	Decimal places	*Number of decimal places that are used by all of the server display elements*		

Mode

Specifies how the interface channel should be handled by the evaluation elements of the server

Analogue/digital	*Analogue or digital measurement value. This means that the system deals with each measurement value independently of the measurement values before or afterwards.*
Cnt.Day	*Day counter. This means that the system anticipates that the measurement value of the channel will continuously increase and is reset once per day.*
Cnt.Intrvl.	*Interval counter. This means that the system assumes that the counter reading is reset each time a measurement is recorded.*
Cnt.Inf.	*Infinite counter. This means that the system expects that the measurement value of the channel will continuously increase and never be reset.*

9.2.1.7.2 Config.

Note: *The following parameters are not available for an interface channel that is connected to an interface for which "Script parsing" was activated (see "Basis" on page 83).*

9.2.1.7.2.1 Com1 (modbus master, RS485)

Slave Ad.

Address of the modbus slave

Modbus Ad.

Address of the register that should be read

Format

Data type

Digital	*A digital value should be read.*	
	Function	*Read coils (FC 01)*
		Read discrete inputs (FC 02)

Signed	*A signed integer value should be read.*				
	Bit	*16*	*16-bit integer*		
		32	*32-bit integer. Two registers must be read from the modbus slave for this purpose.*		
			Word order	*HI-LO*	*HI word on the lower register address and the LO word on the higher register address*
				LO-HI	*LO word on the lower register address and the HI word on the higher register address*
	Function	*Read holding registers (FC 03)*			
		Read input registers (FC 04)			
Unsigned	*The integer value to be read is not signed.*				
	Bit	*16*	*16-bit integer*		
		32	*32-bit integer. Two registers must be read from the modbus slave for this purpose.*		
			Word order	*HI-LO*	*HI word on the lower register address and the LO word on the higher register address*
				LO-HI	*LO word on the lower register address and the HI word on the higher register address*
	Function	*Read holding registers (FC 03)*			
		Read input registers (FC 04)			
Float	*A 32-bit float should be read.*				
	Word order	*HI-LO*	*HI word on the lower register address and the LO word on the higher register address*		
		LO-HI	*LO word on the lower register address and the HI word on the higher register address*		
	Function	*Read holding registers (FC 03)*			
		Read input registers (FC 04)			

9.2.1.7.2.2 Com1 (modbus slave, RS485)

The following table details the possible access functions depending on the data type of the interface channel:

Modbus add.	Data type	Read function	Write function
0x0000 : 0x003F	Digital	Read coils (FC 01)	Write single coil (FC 05) Write multiple coils (FC 15)
0x0000 : 0x007F	Signed 16/32 bit Unsigned 16/32 bit Float	Read holding registers (FC 03)	Write single register (FC 06) Write multiple registers (FC 16)

Modbus Ad.

Address of the register that should be read

Format

Data type

Digital	*A digital value should be read.*				
Signed	*A signed integer value should be read.*				
	Bit	*16*	*16-bit integer*		
		32	*32-bit integer. Two registers must be read from the modbus slave for this purpose.*		
			Word order	*HI-LO*	*HI word on the lower register address and the LO word on the higher register address*
				LO-HI	*LO word on the lower register address and the HI word on the higher register address*
Unsigned	*The integer value to be read is not signed.*				
	Bit	*16*	*16-bit integer*		
		32	*32-bit integer. Two registers must be read from the modbus slave for this purpose.*		
			Word order	*HI-LO*	*HI word on the lower register address and the LO word on the higher register address*
				LO-HI	*LO word on the lower register address and the HI word on the higher register address*
Float	*A 32-bit float should be read.*				
	Word order	*HI-LO*	*HI word on the lower register address and the LO word on the higher register address*		
		LO-HI	*LO word on the lower register address and the HI word on the higher register address*		

9.2.1.7.2.3 Com2 (serial, RS232)

Column

The ASCII data message from the digital sensor is split into individual measurement values by the "number delimiter" (see "Interfaces" on page 83). The "column" configuration parameter specifies which of these measurement values is linked with/recorded on the interface channel.

9.2.1.7.3 Scale

Note: *The following parameters are not available for an interface channel that is connected to an interface for which "Script parsing" was activated (see "Basis" on page 83).*

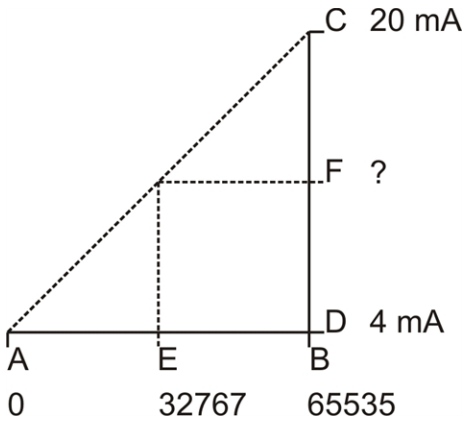
9.2.1.7.3.1 Com1 (modbus, RS485)

Scale	Off	---		
	On	0% modbus	Start of the measurement range in the unit of the modbus slave	
		100% modbus	End of the measurement range in the unit of the modbus slave	
		0% modbus	Start of the measurement range in the measurement unit	
		100% modbus	End of the measurement range in the measurement unit	
		Overflow	Procedure in the event of measurement range violations	
			Ignore	The measurement value is calculated beyond the range limits.
			Silent cutoff	The measurement value is truncated at the range limits.
			Overflow	• The error value "UF" (underflow) is issued, if the measurement value is below the lower limit. • The error value "OF" (overflow) is issued, if the measurement value is above the upper limit.

Note: *Calculation of the measurement value during active scaling:*

Basis › min	*4mA*	*D*
Basis › max	*20mA*	*C*
Min modbus	*0*	*A*
Max modbus	*65535*	*B*

Measurement value in the unit of the modbus slave	*32767*	*E*
Scaled measurement value	*11.9998mA*	*F*



F = ((C - D)/(B - A)) * (E - A) + D

F = ((20mA - 4mA) /(65535 - 0)) * (32767 - 0) + 4mA =11.9998 mA

9.2.1.7.3.2 Com2 (serial, RS232)

The scaling is not available for the serial interface.

9.2.1.7.4 Alarms

Warning	Value low	A warning is triggered, if the measurement value drops to or below this value.
	Value high	A warning is triggered, if the measurement value meets or exceeds this value.
Alarm	Value low	An alarm is triggered, if the measurement value drops to or below this value.
	Value high	An alarm is triggered, if the measurement value meets or exceeds this value.
Hyst %	Hysteresis for the all-clear in the event of an alarm/warning (e.g. Hyst=5%, alarm or warning at 100 -› all-clear at 95) or reset the trigger (e.g. Hyst=5%, level = greater or equal, trigger at 100 -› reset at 95)	

9.2.1.7.5 Trigger

QU	Fastrecording (record interval = record interval / factor)	
SL	Slow recording (record interval = record interval * factor)	
RO	Switch on recording	
RF	Switch off recording	
XM	Initiate transmission	
ON	Activate online mode	
I1	Internal use with the control program	
I2	Internal use with the control program	
I3	Internal use with the control program	
I4	Internal use with the control program	
Level	Levels for initiating the trigger. The hysteresis from the "Alarm" tab is used to determine the level to reset the trigger.	
	Greater or equal	The measurement value must be greater or equal to the level to initiate the trigger.
	Less or equal	The measurement value must be less or equal to the level to initiate the trigger.

9.2.1.8 Interface channels 33-64

9.2.1.8.1 Basis

Title 33-64

Freely selectable channel title for the interface channels

The significance of the remaining configuration parameters in this configuration section corresponds to the "Interface channels 1-32" configuration section (see "Interface channels 1-32" on page 115).

9.2.1.9 Interface output channels 1-32

> ***Note:*** *Some of the configuration parameters included in this configuration section also affect parameters that are not included in the tab that is currently open. In the following section, this is indicated by the ">" symbol. For example, "Basis › Min" means that the "Min" configuration parameter is located in the "Basis" tab.*

An interface output channel is designed to receive a single setpoint that is transferred to a sensor or actuator via one of the 2 serial interfaces (1 x RS485, 1 x RS232). The data type that should be used and the interface, via which the data should be transferred, can be determined for every interface output channel independently of the other channels. The following chapters contain a description on configuring the interface output channels.

9.2.1.9.1 Basis

Title 1-32

Freely selectable channel title for the interface output channels [0-16 characters]

Interface

Basis settings for the output channel:

Off [1)]	*Unit*	*String that is used as a measurement unit by all of the server display elements [0-16 characters]*		
	Decimal places	*Number of decimal places that are used by all of the server display elements*		
	Setpoint	*Output value in the measurement unit*		
Com1	*Config › Format* *(see "Config" on page 124)*	*Digital*	*---*	
		Signed	*Unit*	*String that is used as a measurement unit by all of the server display elements [0-16 characters]*
			Decimal places	*Number of decimal places that are used by all of the server display elements*
		Unsigned	*Unit*	*String that is used as a measurement unit by all of the server display elements [0-16 characters]*
			Decimal places	*Number of decimal places that are used by all of the server display elements*
		Float	*Unit*	*String that is used as a measurement unit by all of the server display elements [0-16 characters]*
			Decimal places	*Number of decimal places that are used by all of the server display elements*
	Setpoint	*Output value in the measurement unit*		
Com2	*Setpoint*	*Output value in the measurement unit*		

[1)] Although the setpoint is not issued via one of the interfaces it can be used by a script. A possible application is to specify the setpoint via the server-PLC and to process it further via a script on the device.

Mode

Specifies how the interface output channel should be handled by the evaluation elements of the server

Analogue/digital	*Analogue or digital measurement value. This means that the system deals with each measurement value independently of the measurement values before or afterwards.*
Cnt.Day	*Day counter. This means that the system anticipates that the measurement value of the channel will continuously increase and is reset once per day.*
Cnt.Intrvl.	*Interval counter. This means that the system assumes that the counter reading is reset each time a measurement is recorded.*
Cnt.Inf.	*Infinite counter. This means that the system expects that the measurement value of the channel will continuously increase and never be reset.*

9.2.1.9.2 Config

Note: *The following parameters are not available for an interface output channel that is connected to an interface for which "Script parsing" was activated (see "Basis" on page 83).*

9.2.1.9.2.1 Com1 (modbus master, RS485)

Slave add.

Address of the modbus slave

Modbus add.

Address of the register that should be written to

Format

Data type

Digital	*A digital value must be written.*				
	Function	*Write single coils (FC 05)*			
		Write multiple coils (FC 15)			
Signed	*A signed integer value should to be written.*				
	Bit	*16*	*16-bit integer*		
			Function	*Write multiple registers (FC 16)*	
				Write single register (FC 06)	
		32	*32-bit integer. Two registers must be written in the modbus slave for this purpose.*		
			Function	*Write multiple registers (FC 16)*	
			Word order	*HI-LO*	*HI word on the lower register address and the LO word on the higher register address*
				LO-HI	*LO word on the lower register address and the HI word on the higher register address*
Unsigned	*The integer value to be written is not signed.*				
	Bit	*16*	*16-bit integer*		
			Function	*Write multiple registers (FC 16)*	
				Write single register (FC 06)	
		32	*32-bit integer. Two registers must be written in the modbus slave for this purpose.*		
			Function	*Write multiple registers (FC 16)*	
			Word order	*HI-LO*	*HI word on the lower register address and the LO word on the higher register address*
				LO-HI	*LO word on the lower register address and the HI word on the higher register address*
Float	*A 32-bit float should to be written.*				
	Word order	*HI-LO*	*HI word on the lower register address and the LO word on the higher register address*		
		LO-HI	*LO word on the lower register address and the HI word on the higher register address*		
	Function	*Write multiple registers (FC 16)*			

9.2.1.9.2.2 Com1 (modbus slave, RS485)

The following table details the possible access functions depending on the data type of the interface output channel:

Modbus add.	Data type	Read function	Write function
0x0800 : 0x083F	Digital	Read discrete inputs (FC 02)	---
0x0800 : 0x087F	Signed 16/32 bit Unsigned 16/32 bit Float	Read input registers (FC 04)	---

Modbus add.

Address of the register that should be written to

Format

Data type

Digital	*A digital value must be written.*				
Signed	*A signed integer value should to be written.*				
	Bit	*16*	*16-bit integer*		
		32	*32-bit integer. Two registers must be written in the modbus slave for this purpose.*		
			Word order	*HI-LO*	*HI word on the lower register address and the LO word on the higher register address*
				LO-HI	*LO word on the lower register address and the HI word on the higher register address*
Unsigned	*The integer value to be written is not signed.*				
	Bit	*16*	*16-bit integer*		
		32	*32-bit integer. Two registers must be written in the modbus slave for this purpose.*		
			Word order	*HI-LO*	*HI word on the lower register address and the LO word on the higher register address*
				LO-HI	*LO word on the lower register address and the HI word on the higher register address*
Float	*A 32-bit float should to be written.*				
	Word order	*HI-LO*	*HI word on the lower register address and the LO word on the higher register address*		
		LO-HI	*LO word on the lower register address and the HI word on the higher register address*		

9.2.1.9.2.3 Com2 (serial, RS232)

Column

The ASCII output string is split into individual measurement values by the "number delimiter" (see "Interfaces" on page 83). The "column"configuration parameter specifies where the output value is placed in the ASCII output string.

9.2.1.9.3 Scale

Note: *The following parameters are not available for an interface output channel that is connected to an interface for which "Script parsing" was activated (see "Basis" on page 83).*

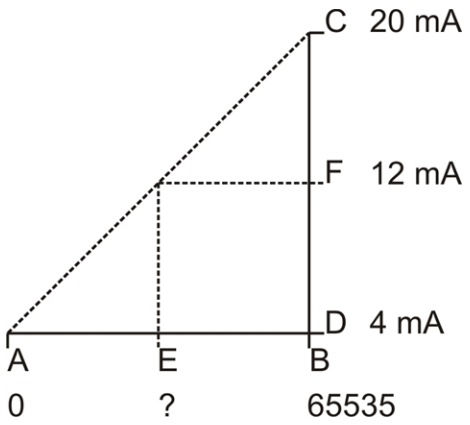
9.2.1.9.3.1 Com1 (modbus, RS485)

Scale	off	--	
	on	0% Modbus	Start of the output range in the unit of the modbus slave
		100% Modbus	End of the output range in the unit of the modbus slave
		0%	Start of the output range in the measurement unit
		100%	End of the output range in the measurement unit

Note: *Calculation of the output value during active scaling:*

Basis › min	*4mA*	*D*
Basis › max	*20mA*	*C*
Min modbus	*0*	*A*
Max modbus	*65535*	*B*

Basis › setpoint	*12mA*	*F*
Scaled output value	*32767*	*E*



E = ((B - A)/(C - D)) * (F - D) + A
F = ((65535 - 0) /(20mA - 4mA)) * (12mA - 4mA) + 0 =32767

9.2.1.9.3.2 Com2 (serial, RS232)

The scaling is not available for the serial interface.

9.2.1.10 Interface output channels 33-64

Title 33-64

Freely selectable channel title for the interface output channels

The significance of the remaining configuration parameters in this configuration section corresponds to the "Interface output channels 1-32" configuration section (see "Interface output channels 1-32" on page 122).

9.2.1.11 Calculated channels

> ***Note:*** *The values of the calculated channels are directly calculated for every data output (display on the myDatanet server or download from the myDatanet server). They are not saved in the server database.*

9.2.1.11.1 Basis

Title 1-5

Freely selectable channel title for the calculated channels [0-16 characters]

Mode

Possible calculation modes for the calculated channels

Off	*---*	*Calculated channel deactivated*
Table	*Unit*	*String that is used as a measurement unit by all of the server display elements [0-16 characters]*
	Decimal places	*Number of decimal places that are used by all of the server display elements*
Digital	*Invert*	*Inverts the input signal*
+, -, x, /	*Unit*	*String that is used as a measurement unit by all of the server display elements [0-16 characters]*
	Decimal places	*Number of decimal places that are used by all of the server display elements*
Delta	*Determines the difference between two measurement values and divides the result by the time difference between the time stamps of the measurement values. The time unit (value/sec., value/min., ...) for the result can be selected via the "Time basis" parameter located in the "Calculation" tab. It is thus possible to convert the counter reading (m^3) of the source in to a flow rate (m^3/min.). (see "Additional explanation: Delta mode" on page 132)*	
	Unit	*String that is used as a measurement unit by all of the server display elements [0-16 characters]*
	Decimal places	*Number of decimal places that are used by all of the server display elements*
clone	*Creates a clone of a measurement channel. This ensures it is possible to rename channels, to select different ends of the scale for the pointer instruments, to determine a new string as a measurement unit and to adjust the number of decimal places. The measurement values (numerical value without unit) correspond exactly to those of the source.*	
	Unit	*String that is used as a measurement unit by all of the server display elements [0-16 characters]*
	Decimal places	*Number of decimal places that are used by all of the server display elements*



Shift element down



Shift element up

9.2.1.11.2 Calculation

Off	---	Calculated channel deactivated
Table	Source	Selection of the channel from which the input data is used
		Opens the screen for entering the values table (the table rows are interpolated linearly, values outside of the defined table are extrapolated linearly.)
Digital	Source	Selection of the channel from which the input data is used
	High level	Signal recognition level
+, -, x, /	Source	Selection of the channel from which the input data is used
	+, -, x, /	
	Source	Selection of the channel from which the input data is used
Delta	Source	Selection of the channel from which the input data is used
	Time basis	Specifies the desired time unit (value/sec., value/min., ...) for the result
	Offset	Offset that is added following multiplication with the "Factor" parameter. The result is converted in to the desired time unit before multiplication with the "Factor" parameter.
	Factor	Factor with which the result is multiplied once it has been converted into the desired time unit. The "Offset" parameter is then added.
Clone	Source	Selection of the channel that should be cloned

Note:

Additional explanation: Delta mode

Assumption: *The source channel contains the counter reading of an infinite counter in m^3. The calculated channel 1 should contain the flow rate in m^3/s and calculated channel 2 should contain the flow rate in l/h.*

Required configuration

Parameter	*Value channel 1*	*Value channel 2*
Basis -> mode	*Delta*	*Delta*
Basis -> unit	*m^3/s*	*l/h*
Calculation -> time basis	*Seconds*	*Hours*
Calculation -> offset	*0*	*0*
Calculation -> factor	*1*	*1000*

Source		*Calculated channel 1*	*Calculated channel 2*
Date/time	*Infinite counter [m^3]*	*Flow rate [m^3/s]*	*Flow rate [l/h]*
26.03.2013 12:50	*900*	*0 [1)]*	*0 [1)]*
26.03.2013 12:51	*960*	*1*	*3,600,000*
26.03.2013 12:52	*990*	*0.5*	*1,800,000*
26.03.2013 12:53	*1005*	*0.25*	*900,000*
26.03.2013 12:54	*1065*	*1*	*3,600,000*

[1)] Calculation not possible as there is no measurement value before 12:50.

Explanation: *No values can be determined for the measurement at 12:50 for the calculated channels as there is no previous value and the difference between the counter readings cannot be determined. For the measurement at 12:51, the difference to the counter reading is 60m^3 and the time difference is 60sec.*

Result = ((value difference / time difference [sec.]) * time basis [sec] * factor) + offset

The result for calculated channel 1 (time basis "Seconds", offset "0" and factor "1") is calculated as follows:

Channel 1 = ((60m^3 / 60sec.) * 1 * 1) + 0 = 1m^3/s

The result for calculated channel 2 (time basis "Hours", offset "0" and factor "1000") is calculated as follows:

Channel 2 = ((60m^3 / 60sec.) * 3600 * 1000) + 0 = 3,600,000l/h

9.2.1.11.3 Alarms

Note: The evaluation of the alarm thresholds for calculated channels can only occur if the device has transferred the measurement data to the myDatanet server.

Alarm low	An alarm is triggered, if the measurement value drops to or below this value.
Alarm high	An alarm is triggered, if the measurement value meets or exceeds this value.
Hyst %	Hysteresis for the all-clear in the event of an alarm/warning (e.g. Hyst=5%, alarm or warning at 100 -› all-clear at 95) or reset the trigger (e.g. Hyst=5%, level = greater or equal, trigger at 100 -› reset at 95)

9.2.1.12 Output channels

9.2.1.12.1 Basis

Ext. warmup time

The output channel is switched on "ext warmup time" seconds prior to the measurement.

mA OUT 1 title

Freely selectable channel title for the analogue output (not galvanically isolated) [0-16 characters]

Relay 1 title

Freely selectable channel title for the isolated switch contact (parallel connection of an optoswitch and relay) [0-16 characters]

Mode

Basis settings for the analogue output

Off	*---*	*Output channel deactivated*
0-20 mA	*0%*	*Start of the output range in the measurement unit*
	100%	*End of the output range in the measurement unit*
	Setpoint	*Output value in the measurement unit*
	Unit	*String that is used as a measurement unit by all of the server display elements [0-16 characters]*
	Decimal places	*Number of decimal places that are used by all of the server display elements*
4-20mA	*0%*	*Start of the output range in the measurement unit*
	100%	*End of the output range in the measurement unit*
	Setpoint	*Output value in the measurement unit*
	Unit	*String that is used as a measurement unit by all of the server display elements [0-16 characters]*
	Decimal places	*Number of decimal places that are used by all of the server display elements*

Basis setting for the isolated switch contact (parallel connection of an optoswitch and relay)

Off	*---*	*Output channel deactivated*
Ext warmup time	*If the value is "0", the output channel is not switched on at all.*	
Digital	*Invert*	*Inverts the level issued on the device. (see "Supplementary explanation regarding "Digital" mode" on page 135)*
	Setpoint	*Setpoint (on/off) that should be issued*
Freq	*Factor*	*The setpoint multiplied by the factor equals the frequency in Hertz.*
	Setpoint	
	Unit	*String that is used as a measurement unit by all of the server display elements [0-16 characters]*
	Decimal places	*Number of decimal places that are used by all of the server display elements*

PWM	*0%*	*Start of the output range in the measurement unit*
	100%	*End of the output range in the measurement unit*
	Setpoint	*Output value in the measurement unit*
	Unit	*String that is used as a measurement unit by all of the server display elements [0-16 characters]*
	Decimal places	*Number of decimal places that are used by all of the server display elements*
Pulse	*Factor*	*The setpoint multiplied by the factor equals the pulse/min.*
	Setpoint	
	Unit	*String that is used as a measurement unit by all of the server display elements [0-16 characters]*
	Decimal places	*Number of decimal places that are used by all of the server display elements*

Note:

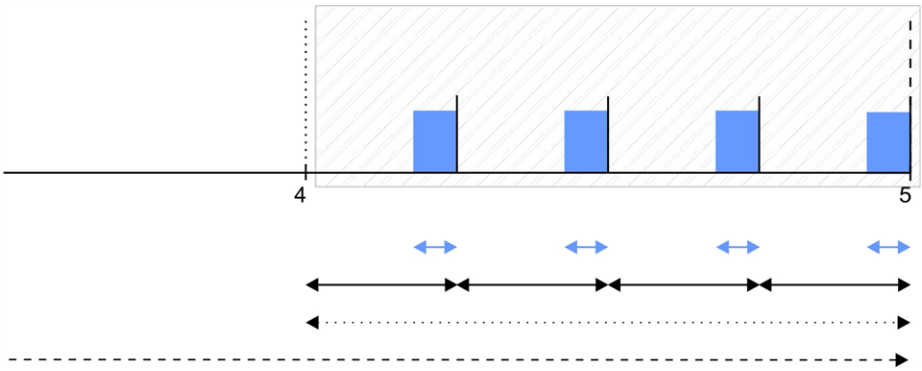
Supplementary explanation regarding "Digital" mode

Invert	*Setpoint*		*Output on the device*
Off	*Off*	=	*Off (low)*
Off	*On*	=	*On (high)*
On	*Off*	=	*On (high)*
On	*On*	=	*Off (low)*

Note:

Example to explain the burst interval in conjunction with the ext. warmup time (ext. warmup time < measurement cycle):

Basic setting		*Record interval*	*5min.*
		Burst interval	*60 sec.*
		Measurement cycle	*15 sec.*
Output channels		*Ext warmup time*	*5 sec.*
Measurement channels - >Config.		*Decay*	*med*
		Time	*60 sec.*
Output on the device		*Sensor supply*	

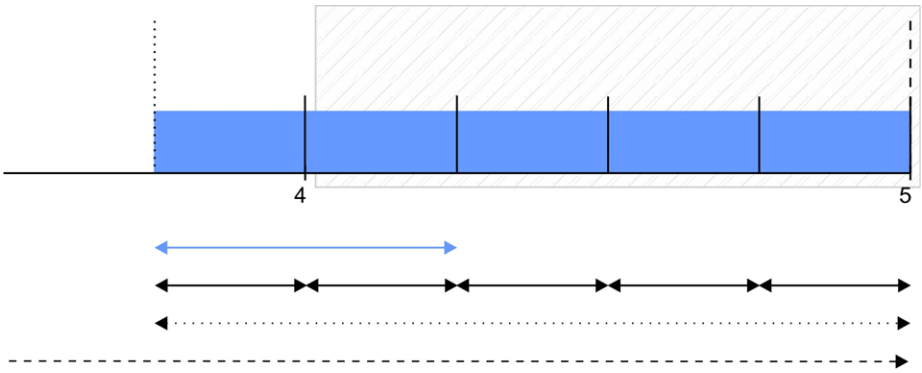


Formation of the measurement value: *In each case, the sensor supply is activated 5 sec before expiry of the measurement cycle. This results in four valid measurements that are used to determine a median and are recorded as the measurement value.*

Note:

Example to explain the burst interval in conjunction with the ext. warmup time (ext. warmup time › measurement cycle):

Basic setting		*Record interval*	*5min.*
		Burst interval	*75 sec.*
		Measurement cycle	*15 sec.*
Output channels		*Ext warmup time*	*30 sec.*
Measurement channels - ›Config.		*Decay*	*med*
		Time	*60 sec.*
Output on the device		*Sensor supply*	



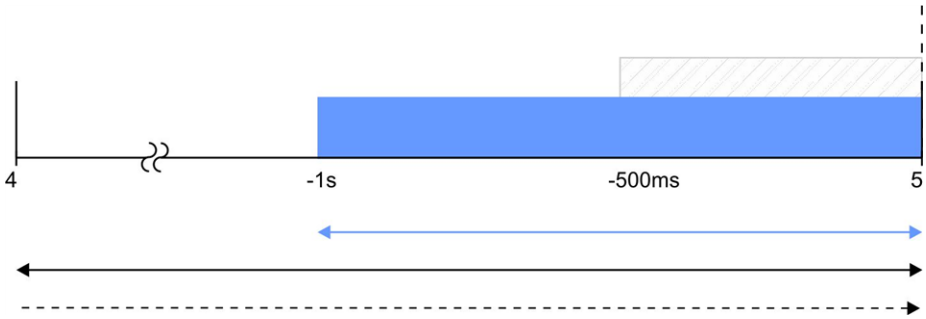
Explanation: *As the ext. warmup time is greater than the measurement cycle in this example, the sensor supply is activated right at the start of the burst interval and is only deactivated again when the recording starts. The first measurement is completed 15 sec. (measurement cycle) after the burst interval has started. At this time, the sensor supply had not yet been active for the duration of the ext. warmup time. This means that the measurement value of the connected sensor is probably not stable yet and the measurement is thus invalid. In this example, only the second measurement after the start of the burst interval is valid. To exclude the invalid measurements during the formation of the measurement value, the selected decay period must be appropriately smaller than the burst interval. The burst interval has to be increased from 60 sec. to 75 sec. to receive four valid measurements in exactly the same way as in the previous example (ext warmup time ‹ measurement cycle).*

Formation of the measurement value: *The sensor supply is activated at the start of the burst interval. The first measurement is completed 15 sec. later. Five measurements are thus generated. The first measurement is invalid (ext. warmup time has not expired yet). As the decay period only lasts 60 sec., only the second to fifth measurement are used to determine the median and are recorded as the measurement value. The first invalid measurement is thus ignored.*

Note:

Example to explain the filter time in conjunction with the ext. warmup time

Basic setting		*Record interval*	*5min.*
		Measurement cycle	*1min.*
Output channels		*Ext warmup time*	*1sec.*
Measurement channels -›Config.		*Filter time*	*500ms*
Output on the device		*Sensor supply*	



Explanation: *The sensor supply is always activated 1 sec before expiry of the measurement cycle. The filter time starts 500ms before expiry of the measurement cycle, which ensures that a valid value is available at the time of a measurement. However, this also means that the filter time must be taken into consideration when selecting the ext warmup time. In the current example, the sensor connected to the input must at the latest supply stable values 500ms following activation of the sensor supply. Otherwise, invalid values would be taken into consideration while averaging during the filter time resulting in the measurement value being distorted.*

9.2.1.12.2 Config

Isolated switch contact (parallel connection of an optoswitch and relay)

Off	---	
Ext warmup time	---	
Freq	---	
PWM	---	
Pulse	Duration	Duration of an output pulse [ms]

9.2.1.13 Internal channels

9.2.1.13.1 Basis

GSM level title	Freely selectable channel title for the GSM field level[0-16 characters]	
	Unit	String that is used as a measurement unit by all of the server display elements [0-16 characters]
Voltage title	Freely selectable channel title for the supply voltage [0-16 characters]	
	Unit	String that is used as a measurement unit by all of the server display elements [0-16 characters]

9.2.1.13.2 Alarms

Warning	Value low	A warning is triggered, if the measurement value drops to or below this value.
	Value high	A warning is triggered, if the measurement value meets or exceeds this value.
Alarm	Value low	An alarm is triggered, if the measurement value drops to or below this value.
	Value high	An alarm is triggered, if the measurement value meets or exceeds this value.
Hyst %	Hysteresis for the all-clear in the event of an alarm/warning (e.g. Hyst=5%, alarm or warning at 100 -› all-clear at 95) or reset the trigger (e.g. Hyst=5%, level = greater or equal, trigger at 100 -› reset at 95)	

9.2.1.13.3 Trigger

QU	Fastrecording (record interval = record interval / factor)	
SL	Slow recording (record interval = record interval * factor)	
RO	Switch on recording	
RF	Switch off recording	
XM	Initiate transmission	
ON	Activate online mode	
I1	Internal use with the control program	
I2	Internal use with the control program	
I3	Internal use with the control program	
I4	Internal use with the control program	
Level	Levels for initiating the trigger. The hysteresis from the "Alarm" tab is used to determine the level to reset the trigger.	
	Greater or equal	The measurement value must be greater or equal to the level to initiate the trigger.
	Less or equal	The measurement value must be less or equal to the level to initiate the trigger.

9.2.1.14 Alarm settings

Acknowledgement	Standard	The global server setting is used to determine whether alarms must be acknowledged automatically or manually. (see "myDatanet Server Manual " 805002).
	Automatic	Alarms are acknowledged automatically as soon as all of the messages have been sent. If SMS that have a tariff with a delivery confirmation function have also been sent, acknowledgement is provided after delivery confirmation.
	Manual	Alarms must be acknowledged by the user.
Offline alarm	Alarm in case the device does not respond for longer than the set number of transmission cycles	
Transfer volume	Standard	The setting for the transfer volume alarm is taken from the global server settings. (see "myDatanet Server Manual " 805002).
	Off	The transfer volume alarm is deactivated.
	Individual	The level at which the transfer volume alarm should be triggered can be entered in the adjacent field in KB.
On alarm	A	The alarm is recorded in the alarm list.
	Ü	An immediate transmission is initiated.
On warning	A	The warning is recorded in the alarm list.
	Ü	An immediate transmission is initiated.
On fault alarm [1)]	A	The fault alarm is recorded in the alarm list.
	Ü	An immediate transmission is initiated.
On fault warning [2)]	A	The fault warning is recorded in the alarm list.
	Ü	An immediate transmission is initiated.

[1)] A "fault alarm" is triggered if a measurement channel is configured in such a way that one of the error values NAN, OL (open loop), UF (underflow), OF (overflow) or SC (short circuit) is issued if the measurement range is breached and maintaining the last measurement value is not activated or the maximum number for holding the last valid measurement value has been reached.

[2)] A "fault warning" is triggered if a measurement channel is configured in such a way that one of the error values NAN, OL (open loop), UF (underflow), OF (overflow) or SC (short circuit) is issued if the measurement range is breached and the maximum number for holding the last valid measurement value has not yet been reached.

9.2.1.15 Basic settings

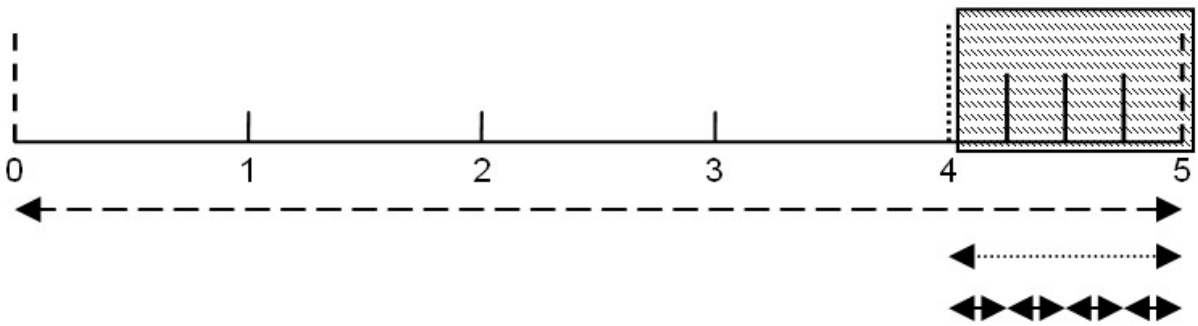
Connection type	Interval	The device connects in the transmission cycle.
	Interval & wakeup	The device connects to the server in the transmission cycle and can be placed into ALOHA transmission mode via the server. 
	online	The device does not separate the connection and continuously transmits the measurement data.
Aloha/Wakeup duration	Duration of the Aloha/wakeup connection	
Transmission cycle	Time between transmissions	
Record interval	Time between measurement data recordings	
Trigger measurement on wakeup	A measurement is also started if the device is placed in ALOHA transmission mode by the server.	
Measure quick divisor	record interval = record interval / factor (from triggering)	
Measure slow factor	record interval = record interval * factor (from triggering)	
Measurement cycle	Time between measurements (00:00 same as record interval)	
Burst interval	Period of time, during which measurement intervals are used for measuring, before recording begins (00:00 measurement cycle is permanently activated)	
Delay for power off alarm	A relevant alarm is only triggered if the supply voltage has failed for longer than the time configured here. This prevents alarms from being triggered in the event of brief supply voltage failures. However, an entry is always added to the device log.	
Time zone	Regional settings (not relevant for raw measurement data as this is stored in UTC)	
Daylight saving time	Configuration for automatic time adjustment	
	Standard	The configuration for the time adjustment is adopted by the global server setting (see "myDatanet Server Manual " 805002).
	Off	Automatic time adjustment deactivated
	USA	Predefined setting for the American area
	EU	Predefined setting for the European area
Position cycle	Position update interval (00:00 positioning is completed with every connection)	

Default report	Selection of the report that is loaded by clicking on the device link in the maps	
	Off	The default graphic is loaded.
	"Name of a report"	The selected report is loaded.
Report template	Selection of whether the default graphic or a report template to display the data is used when clicking on the symbol to display the measurement data located in the site/application list	
	(not assigned)	The default graphic is used to display the measurement data.
	"Name of a report template"	Name of the report template used to display the measurement data

Note:

Example to clarify the record, measurement and burst interval (in conjunction with the decay)

Basic setting		*Record interval*	*5 min.*
		Burst interval	*1 min.*
		Measurement cycle	*15 sec.*
Measurement channels - ›Config.		*Decay*	*med*
		Time	*60 sec.*



Formation of the measurement value: The last four measurement values are used to determine the median and recorded as the measurement value.

Note: *Additional explanation about the connection types*

Connection type	*Energy consumption*	*Data volumes*	*Response time*
online			
Interval & wakeup			
Interval			

9.2.1.16 FTP export settings

Note: This configuration section is only visible if the "FTP Agent Extended" licence for the myDatanet-Server has been enabled.

FTP export profile	off	FTP export deactivated
	"Name of an FTP export profile"	List with the FTP export profiles that were created on the myDatanet server (for creating an FTP export profile, see "myDatanet Server Manual " 805002).
Settings of the selected profile	shows an overview of the most important parameters of the selected FTP export profile	
FTP directory	makes overwriting the standard directory of the selected FTP export profile possible [0-100 characters]	

9.2.2 Device configuration

Note: Several of the configuration fields in the following sub chapters may possibly be hidden depending on the respective user level. In this case, please contact the myDatanet-Server administrator.

You can reach the input screen for configuring the device by clicking on the serial number in the site list (see "myDatanet Server Manual " 805002) or by clicking on the device name in the device name list (see "myDatanet Server Manual " 805002).

9.2.2.1 Comments

Comments

Free comment field (is also displayed below the site name in the measurement instrument list)

9.2.2.2 Measurement instrument

Customer	Name of the customer to whom the measurement instrument is assigned
Serial number	Serial number of the instrument
Instrument class	The instrument class of the site and instrument must match for an instrument to be able to be connected to a site.
Telephone number	Telephone number of the SIM card. The control SMS messages (e.g. wakeup) are sent to this number. Format: +43555837465
Instrument flags	Additional information regarding the instrument class (for internal use)
Firmware version	Current software version installed on the measurement controller
Modem version	Current software version installed on the modem controller

OS version	OS version of the modem	
Last connection	In each case, the last time stamp of the affected operation	
Last wakeup		
Last disconnection		
Last transmission error		
Last Aloha connection		
Firmware update	Off	Firmware update is deactivated.
	On	As soon as a new version of the selected firmware type is available, this is installed immediately.
	Even if tag is missing	Firmware is also transferred to the device if the device has not transmitted the current firmware version to the server (NOT RECOMMENDED!).
	Allow downgrade	Facilitates the installation of an older firmware version than the one on the device (NOT RECOMMENDED!)
	Once	Performs a single firmware update. If no new firmware is available or the firmware was installed successfully, the firmware update is automatically switched to "OFF".
	Ignore	The firmware update is deactivated and no information is provided about available firmware updates.
Firmware type	Released	Only firmware versions that have successfully undergone internal and field testing are installed (this practically eliminates malfunctions).
	Release candidate	Only firmware versions that have successfully undergone internal testing are installed (malfunctions cannot be excluded).
	Beta release	Even firmware versions that have not successfully undergone all of the internal tests are installed (malfunctions may occur).
Hardware version	Hardware version of the myDatalogMUCmini	

9.2.2.3 Device-specific settings

Operating type	hold	Measurement: OFF, transmission: ON
	run	Measurement: ON, transmission: ON
	transport	Measurement: OFF, transmission: OFF
	offline	Measurement: ON, transmission: OFF

9.2.2.4 GPRS

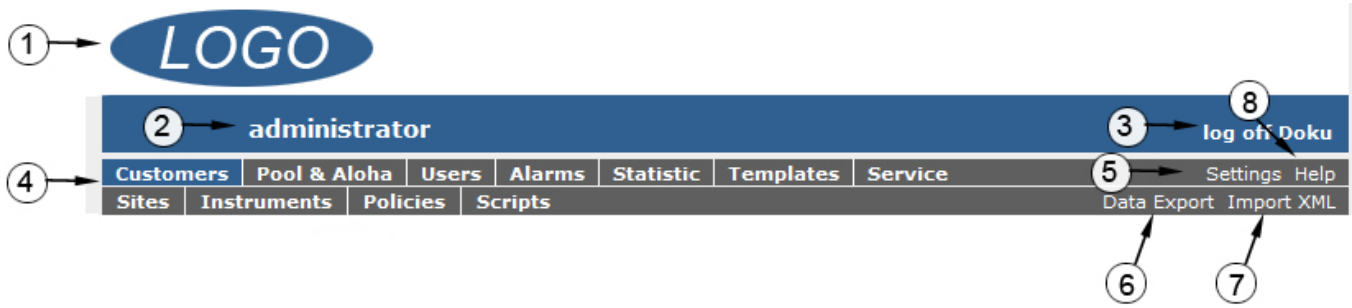
SIM tariff

Selected SIM tariff

Chapter 10 myDatanet server

Note: All of the screenshots show version 42.2 of the myDatanet server using the standard colour scheme. Newer versions may include minor changes to the appearance of the server.

10.1 Overview



Overview of the myDatanet server

1	Freely selectable logo	5	Opens the screen to input the global settings for the server
2	User group to which the user who is currently logged in is assigned	6	Switches to the "Data exports" area to configure the data export. This tab is only visible if at least the licence for one export variant is available.
3	Logout the active user	7	Opens the input screen to upload a XML file. This tab is only visible if the licence for the XML import is available.
4	Tabs to switch between the individual server areas	8	Call up the help menu

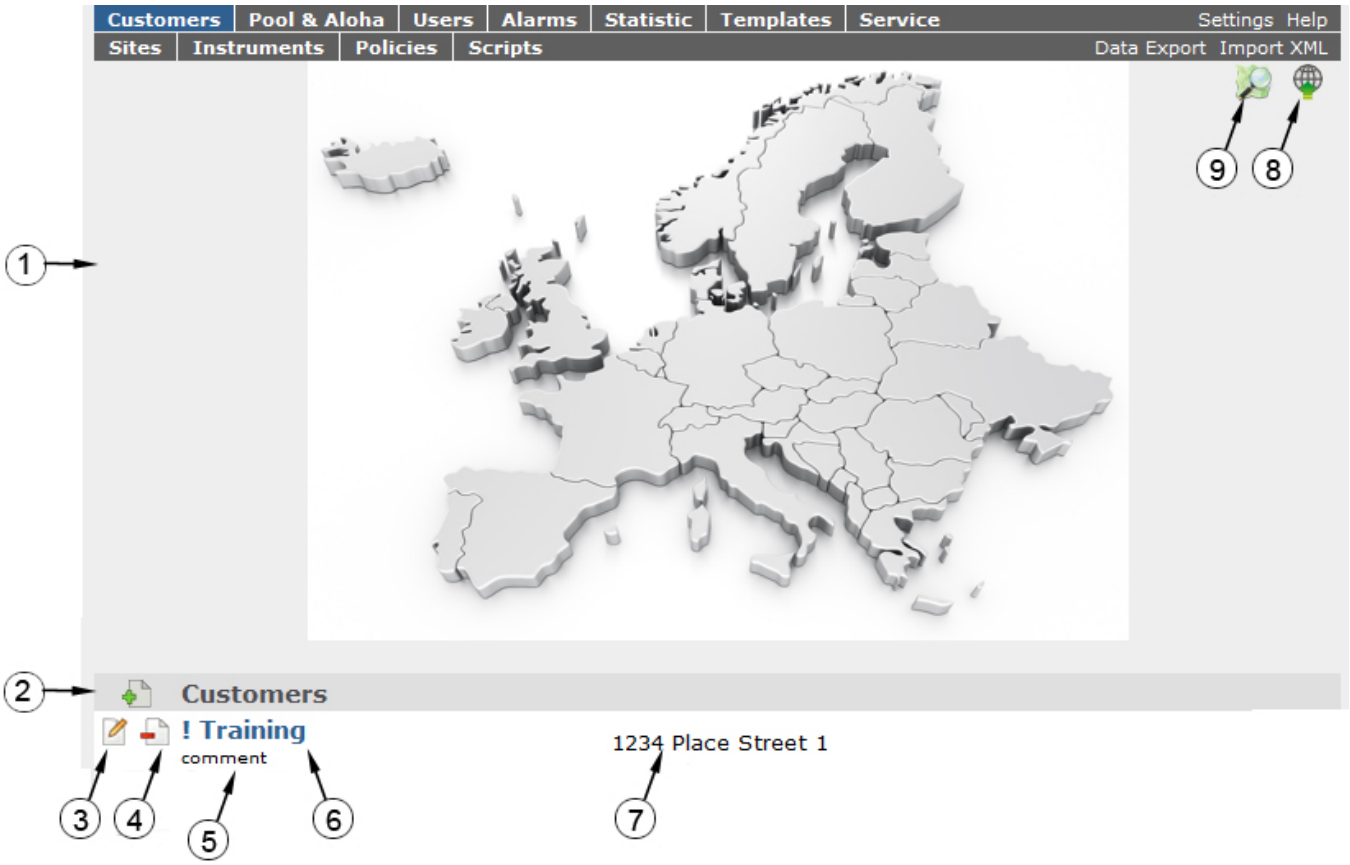
10.1.1 Explanation of the symbols

Adds a new entry to the current list (reports, sites, users, etc.).

Deletes the adjacent element (reports, sites, users, etc.) from the list.

Calls up the input screen to edit the adjacent element (reports, sites, users, etc.).

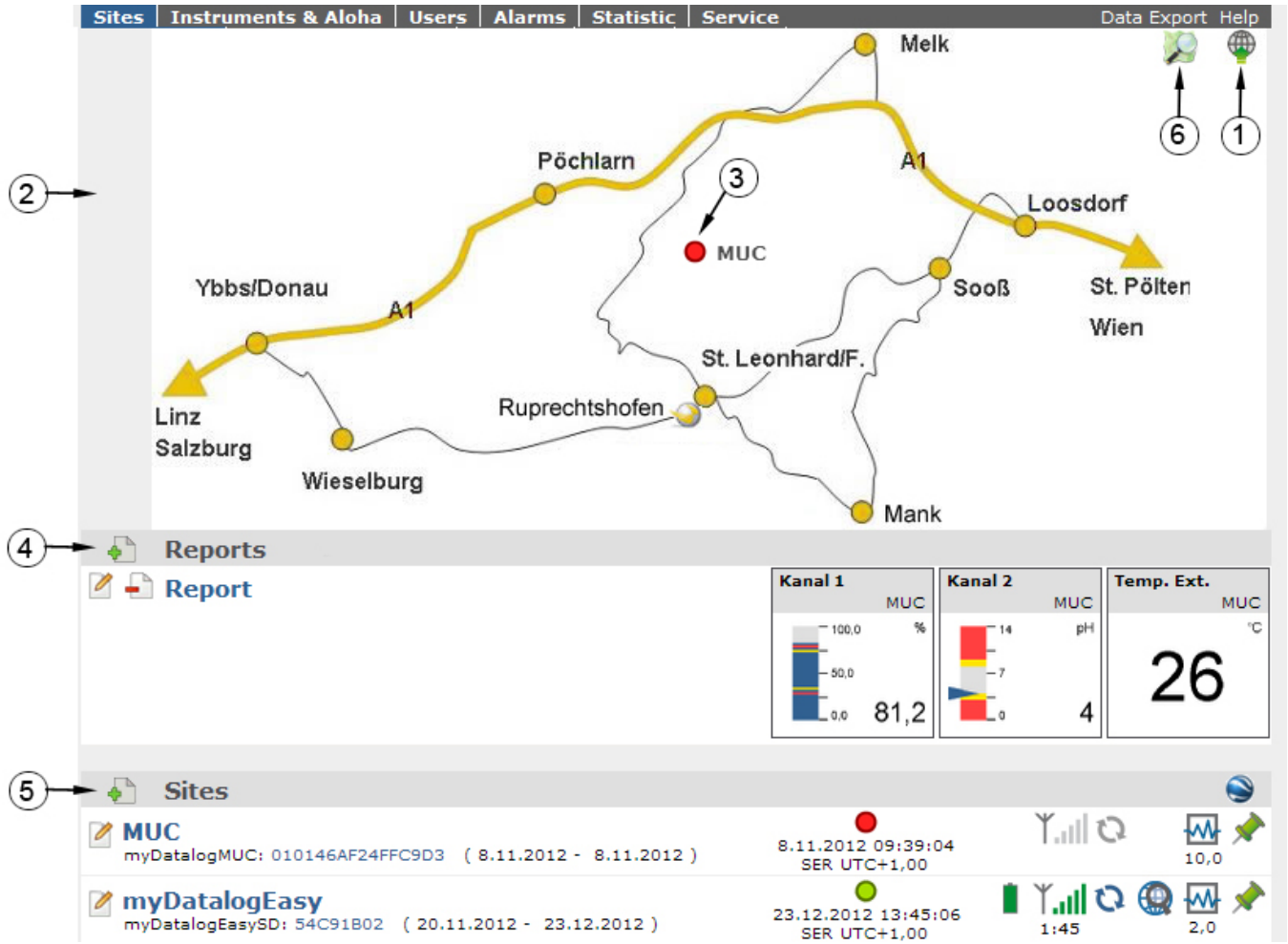
10.2 "Customer" area



Overview of the "Customer" area

1	Current "Overview map" on which the sites can be placed
2	Adds a new customer
3	Opens the input screen for configuring the customer
4	Deletes the customer
5	Comment that can be entered in the configuration of the customer
6	If a default report was defined, the default report is accessed by clicking on the name of the customer. Otherwise the "Sites" area at customer level is opened by clicking on the name of the customer (see ""Site" area at customer level" on page 149 or "Reports" on page 150).
7	Customer's address that can be entered via the input screen for configuring the customer
8	Symbol via which an image file can be loaded on to the server as an "Overview map" To remove the "Map" again, open the upload dialogue again and click on "Submit" without selecting an image file beforehand.
9	Symbol via which a OpenStreetMaps map, on which the sites are displayed, can be loaded. (see "Map view" on page 150)

10.3 "Site" area at customer level



Overview of the "Sites" area at customer level

1	Symbol via which an image file can be loaded on to the server as a "Map" To remove the "Map" again, open the upload dialogue again and click on "Submit" without selecting an image file beforehand.
2	Current "Map" on which the sites can be placed
3	Symbol that represents a site on the "Map"
4	List of reports (see "Reports" on page 150)
5	List of sites / applications (see "Site" on page 81)
6	Symbol via which a OpenStreetMaps map, on which the sites are displayed, can be loaded. (see "Map view" on page 150)

10.3.1 Reports

The reports provide a variety of options to display graphs of the data on the web interface of the myDatanetserver or to download the data from the myDatanet server. Detailed instructions on creating and handling the reports is provided in myDatanet Server Manual (805002).

10.3.2 Map view

The map view provides an overview of the geographic position of the sites. Detailed instructions on operating and configuring map view are provided in myDatanet Server Manual (805002).

10.4 Recommended procedure

10.4.1 Creating the site

> ***Note:*** *Some of the fields mentioned in the following chapters may be hidden depending on the respective user level. In this case, please contact the administrator of the myDatanet server.*

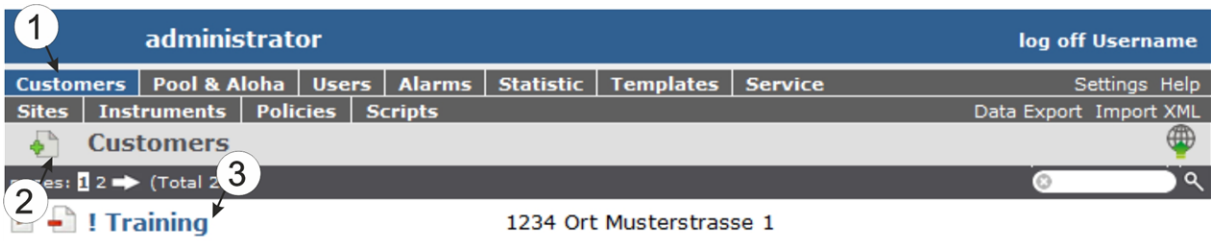
Detailed instructions on creating a new site are provided in myDatanet Server Manual (805002).

1. Log in via the web interface on the myDatanet server. You will receive the web address from your responsible sales partner.



Login form of the myDatanet server

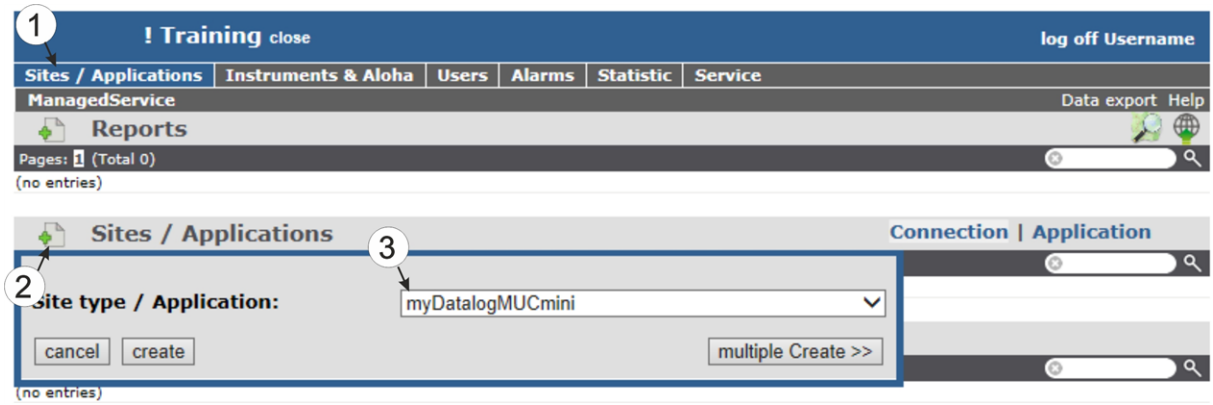
2. Click on the "Customer" menu item of the myDatanet server to call up the list of available customers. Select an existing customer or create a new customer.



Selecting the customer

1 Menu item to call up the list of customers	3 List of available customers
2 Creating a new customer	

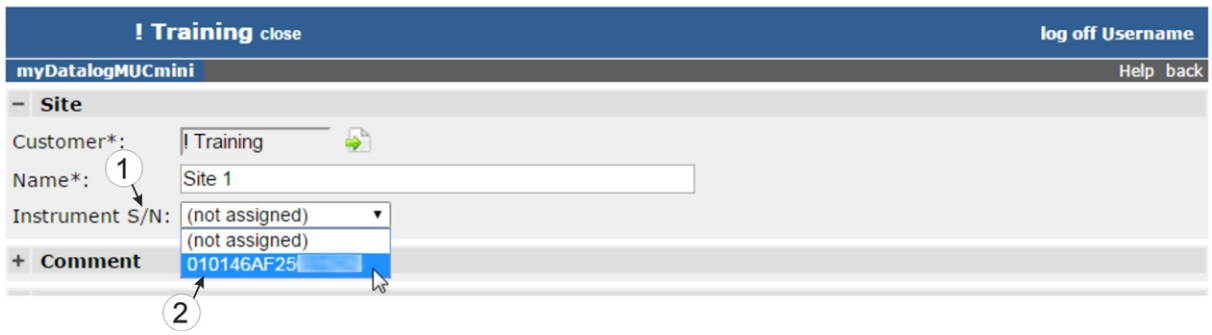
3. Click on the "Sites / Applications" menu item of the myDatanet server to call up the list of available application templates and sites. Create a new "myDatalogMUCmini " type site or a site using an application template that is compatible with the "myDatalogMUCmini " site type.



Creating the site

1 Menu item to call up the list of sites	3 Selection of the type of site
2 Creating a new site	

4. Link the site/application with the myDatalogMUCmini by selecting the serial number. If the serial number of your device is not included in the list, you must first assign the device to the customer (see "Assigning a device to the customer" on page 151) or the device is already assigned to a different site. More detailed information on this is provided in myDatanet Server Manual (805002).



Linking the device and site

1 Device assignment	2 List of devices that have not yet been linked with a site

10.4.2 Assigning a device to the customer

Note: *Some of the fields mentioned in the following chapters may be hidden depending on the respective user level. In this case, please contact the administrator of the myDatanet server.*

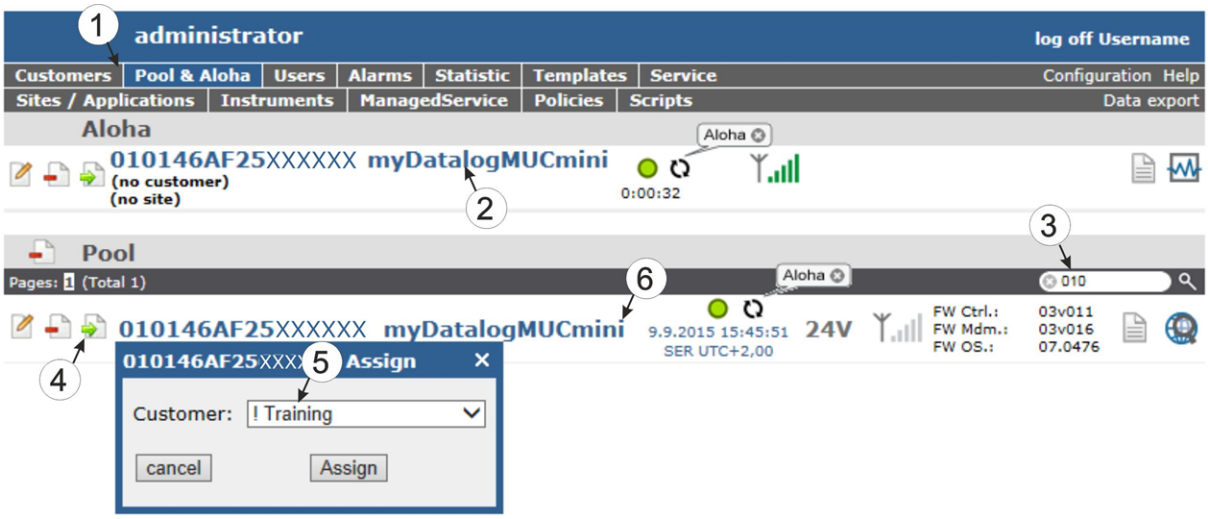
Detailed instructions on shifting the devices are provided in myDatanet Server Manual (805002).

1. Log in via the web interface on the myDatanet server. You will receive the web address from your responsible sales partner.



Login form of the myDatanet server

2. Click on the "Pool and Aloha" menu item of the myDatanet server to access the list of devices that are in ALOHA transmission mode.



Assigning a device to the customer

1	Menu item to call up the list of devices that are in ALOHA transmission mode	4	Opens the dialogue to select the customer to whom the device should be assigned
2	List of devices in ALOHA transmission mode	5	Selection of the customer to whom the device should be assigned
3	Filter for the list of measurement instruments	6	List of measurement instruments

3. Initiate ALOHA transmission mode on the device. Instructions on this are provided in "ALOHA transmission mode" on page 45.

4. Use the filter to find the required device in the list of measurement instruments. Click on the symbol to open the dialogue to select the customer (see "Assigning a device to the customer" on page 152) and assign the device to the selected customer.

Chapter 11 Instruction list

11.1 General

The following chapter describes the functionality of the instruction list parser in accordance with IEC 61131-3:2003.

There are two ways to create an instruction list for the myDatalogMUCmini :

- Direct entry in the "Script" input field in the "Control" configuration section
- Use of a script template, that was created on the myDatanet server

11.1.1 Direct entry of an instruction list

The instruction list is entered via the "Control" configuration section (see "Control" on page 82) of the input screen for configuring the site. "IL" must be selected as the "script type" so that myDatalogMUCmini interprets the commands entered in the "script" as an instruction list.

11.1.2 Using a script template

Predefined script templates can be selected via the "Control program" list selection in the "Control" configuration section (see "Control" on page 82) of the input screen for configuring the site (to create a template, see "myDatanet Server Manual " 805002). After selecting the control program, the template is copied to the "Script" input window where is can be adapted. When using this method, "IL" must also be selected as the "script type" so that myDatalogMUCmini interprets the commands as an instruction list.

> ***Note:*** *If an instruction list is used to write directly to an output, the setpoint entered via the input screen of the myDatanet is overwritten with the calculated value.*

11.2 Implementation details

11.2.1 Common elements

No. of the table	Attribute	Description	Example/explanation
1	2	Lower case letters	
	3a	Hash (#)	
2	1	Capital letters and numbers	IW215 IW215Z QX75 IDENT
3	1	Comments	(**********) (* A framed comment *) (**********)
4	1	Integer literal	-12 0 123_456 +986
	2	Real literal	-12.0 0.0 0.456 3.14159_26
	7	Boolean zero and one	0 1
7	1a	Duration without underscore: short prefix	T#14ms T#-14ms T#14.7s T#14.7m T#14.7h t#14.7d t#25h15m t#5d14h12m18s3.5ms
8	6	Literal for date and time (short prefix)	DT#1984-06-25-15:36:55 dt#1984-06-25-15:36:55
10	1	BOOL	Data type: Boolean, 1 bit
	4	DINT	Data type: double integer, 32 bit
	8	UDINT	Data type: unsigned double integer, 32 bit
	10	REAL	Data type: real number, 32 bit
	12	TIME	Data type: Duration
	15	DATE_AND_TIME or DT	Data type: Date and time
	19	DWORD	Data type: Bit sequence 32, 32 bit
15	1	I (storage location input)	%IW215 %IX2.5
	2	Q (storage location output)	%QX75 %QR1
	3	M (storage location marker)	%MD48 %MR1000
	4	X ((individual) bit size)	
	5	No ((individual) bit size)	
	6	B (byte (8 bits) size)	
	7	W (word (16 bits) size)	
	8	D (double word (32 bits) size)	

No. of the table	Attribute	Description	Example/explanation
23	1	ABS	Absolute
	2	SQRT	Square root LD %IRo SQRT
24	12	ADD	OUT := IN1 + IN2
	13	MUL	OUT := IN1 * IN2
	14	SUB	OUT := IN1 - IN2
	15	DIV	OUT := IN1 / IN2
	16	MOD	OUT := IN1 modulo IN2
25	1	SHL	OUT := IN1 « IN2
	2	SHR	OUT := IN1 » IN2
	3	ROR	OUT := IN1 rotate right IN2
	4	ROL	OUT := IN1 rotate left IN2
26	5	AND	OUT := IN1 & IN2
	6	OR	OUT := IN1 or IN2
	7	XOR	OUT := IN1 xor IN2
	8	NOT	OUT := not IN1
27	2a	MAX	Compares the value in the accu and operand and provides the larger of the two
	2b	MIN	Compares the value in the accu and operand and provides the smaller of the two
28	5	GT	OUT := IN1 › IN2
	6	GE	OUT := IN1 ›= IN2
	7	EQ	OUT := IN1 = IN2
	8	LE	OUT := IN1 ‹= IN2
	9	LT	OUT := IN1 ‹ IN2
	10	NE	OUT := IN1 ‹› IN2
30	1a	ADD	TIME := TIME + TIME
	3a	ADD	DT := DT + TIME
	4a	SUB	TIME := TIME - TIME
	8a	SUB	DT := DT - TIME
	9a	SUB	TIME := DT - DT

11.2.2 IL syntax elements

No. of the table	Attribute	Description	Example/explanation
52 (1/3)	1	LD	Synchronises the current result with the operand LD %IR0 LDN %MX4.0
	2	ST	Saves the current result to the operand address ST %QR0 STN %MX1000.2
	4	AND	Logical AND AND 8 ANDN 12
	6	OR	Logical OR OR 8 ORN 12
	7	XOR	Logical exclusive OR XOR 8 XORN 12
	7a	NOT	Logical negation (of a complement) NOT %MX0:1
	8	ADD	Addition LD 3 ADD 4 LD T#12s ADD T#1h23m LD DT#1984-06-25-15:36:55 ADD T#1h23m

No. of the table	Attribute	Description	Example/explanation
52 (2/3)	9	SUB	Subtraction LD 3 SUB 2 LD T#1d3h12s SUB T#1h23m LD DT#1984-06-25-15:36:55 SUB T#1h23m LD DT#1988-06-25-15:36:55 SUB DT#1984-06-25-15:36:55
	10	MUL	Multiplication LD 2 MUL 3
	11	DIV	Division LD 12 DIV 6
	11a	MOD	Modulo division LD 12 MOD 5
	12	GT	Comparison: › LD 12 GT 10
	13	GE	Comparison: ›= LD 12 GE 10
	14	EQ	Comparison: = LD 12 EQ 12
	15	NE	Comparison: ‹› LD 12 NE 10

No. of the table	Attribute	Description	Example/explanation
52 (3/3)	16	LE	Comparison: <= LD 12 LE 13
	17	LT	Comparison: < LD 12 LT 10
	18	JMP	Jump to the point JMP {label} JMPC {label} conditional JMPCN {label} conditional not LD %IDo LT 10 JMPCN label1 ADD 3 label1: SUB 2

11.3 Instruction list error codes

The instruction list script is run through line-by-line at the time of every measurement once all of the measurement values have been generated. If an error occurs during this process, the script execution stops and enters the relevant error code in the device log. The number of the line in which the error occurred is entered as the error code parameter. Instructions on evaluating the device log are included in the chapter "Evaluating the device log" (see "Evaluating the device log" on page 310).

Log entry		Parameter		Description
Code	Plain text	Code	Plain text	
3001	UNKNOWN ERROR	## [1)]	---	Unknown error
3002	DECODING ERROR	## [1)]	---	Error when decoding the script line
3003	OPERATOR ERROR	## [1)]	---	Invalid operator (see "IL syntax elements" on page 156 and "Expansions" on page 160)
3004	TYPE MISMATCH	## [1)]	---	Data type of the operand and accu are not compatible.
3005	PARSE ERROR	## [1)]	---	No label, comment, operator or operand found in the script line

Log entry		Parameter		Description
Code	Plain text	Code	Plain text	
3006	UNKNOWN OPERAND	## [1]	---	Invalid operand (see "Common elements" on page 154, table no. 4, no. 7, no. 8, no. 15)
3007	INVALID DIRECT ADDRESS	## [1]	---	No valid address
3008	INVALID NUMERIC OPERAND	## [1]	---	Invalid numeric operand
3009	UNKNOWN LABEL	## [1]	---	The label to which the jump is to be made does not exist.
3010	INVALID LOCATION	## [1]	---	Invalid specification of the storage location (see "Common elements" on page 154 table no. 15)
3011	SIZE NOT SUPPORTED	## [1]	---	Invalid specification of the type when reading or writing to the storage location (see "Common elements" on page 154 and "Expansions" on page 160 table no. 15 respectively)
3012	NOT ABLE TO WRITE TO OPERAND	## [1]	---	An attempt is being made to write to a read-only operand.
3013	MAX NUMBER OF LABELS REACHED	## [1]	---	The number of defined labels exceeds the maximum permissible number for this device class.
3014	LABEL ALREADY DEFINED	## [1]	---	A label with exactly the same title has already been defined in previous script lines.
3015	MAX INSTRUCTIONS PER RUN EXCEEDED	## [1]	---	The number of script commands exceeds the maximum permissible number for this device class.
3016	UNKNOWN TYPE	## [1]	---	Unknown data type of the operand
3017	INVALID TIME OPERAND	## [1]	---	Invalid specification of a "duration" type operand
3018	INVALID DATE AND TIME OPERAND	## [1]	---	Invalid specification of a "date and time" type operand

[1] Number of the script line in which the error has occurred

11.4 Expansions

The standard version was extended with the following expansions:

No. of the table	Description	Example/explanation
15	R (real number, (32 bits) size)	%IR0 %QR2 %MR1000
	T (duration - size)	LD T#1d23h45m ST %MT0
	S (date and time - size)	LD DT#2010-08-25-15:36:55 ST %MS0
52 (1/2)	DOW Day of week	Determine the day of the week from a "date and time" type of variable (* 0 = Monday *) DOW DT#2010-08-25-15:36:55
	LDS [1)] Load status	(* Load status from %IR1*) LDS %IR1
	STS [1)] Store status	(* Set %QR1 status to 2 *) LD 2 STS %QR1
	POW Power of	The variable can be exponentiated with the operator "POW". LD %IR0 POW 2.0

No. of the table	Description	Example/explanation
52 (2/2)	INC Increment	Increases a variable LD 5 INC
	DEC Decrement	Reduces a variable LD 5 DEC
	JGTZ	Jump to the label, if the value in the accu › 0 JGTZ {label} LD 5 JGTZ label1 ADD 3 label1: SUB 2
	JGEZ	Jump to the label, if the value in the accu >= 0
	JZ	Jump to the label, if the value in the accu = 0
	JNZ	Jump to the label, if the value in the accu ‹› 0
	JLEZ	Jump to the label, if the value in the accu <= 0
	JLTZ	Jump to the label, if the value in the accu ‹ 0
	CEIL	Round up the value in the accu
	FLOOR	Round down the value in the accu
	ROUND	Commercially round the value in the accu

[1)]myDatalogMUCmini uses a special coding to indicate various error statuses in the measurement channels (e.g. OL = open loop, SC = short circuit). This error code that is encoded in the normal measurement value must not be used for calculations. This error code can be read or written with both "LDS" and "STS" operators. The "LDS" operator always provides a result from the "UINT" type. The "STS" operator expects a parameter from the "UINT" type.

Possible error codes

Status	Description
0	Value valid
1	Undefined error
2	Overflow
3	Underflow
4	Open loop (interruption)
5	Short circuit

11.5 Device-specific implementation

Maximum permissible number of script commands	100
Maximum number of characters for a syntax element	25
Maximum number of characters for a script line	100
Maximum permissible number of labels	20
Input addresses	%Ix0 .. %Ix70, %I1000...%I1063 • %Ix0...%Ix2: Universal inputs 1-3 • %Ix3: Internal measurement channel "GSM level" • %Ix4: Internal measurement channel "voltage" • %Ix5: Analogue output • %Ix6: Relay • %Ix7...%Ix70 or %Ix1000...%Ix1063: Interface channels 1-64
Output addresses	%Qx0 .. %Qx1, %Qx1000...%Qx1063 • %Qx0: Analogue output • %Qx1: Relay • %Qx1000 ... %Qx1063: Interface output channels 1-64
Memory register addresses	%Mx0 .. %Mx31

Global register addresses	%Mx1000 .. %Mx1001
	• %Mx1000: Global trigger mask
	  • Bit0: Warning
	  • Bit1: Alarm
	  • Bit2: Fault warning
	  • Bit3: Fault alarm
	  • Bit4-7: I1-I4
	  • Bit8: Fast recording
	  • Bit9: Slow recording
	  • Bit10: Start measurement cycle immediately
	  • Bit11: Switch on recording
	  • Bit12: Switch off recording
	  • Bit13: Initiate transmission
	  • Bit14: 0...Trigger is initiated when "greater or equal" to the level, 1...Trigger is initiated when "less or equal" to the level
	  • Bit15: Activate online mode
	  • Bit16: Warmup time active
	  • Bit17: Measurement active
	  • Bit18-19: Reserved
	• %Mx1001: Current time stamp

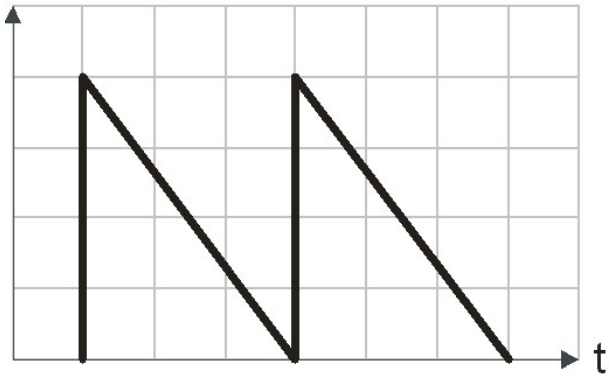
Register addresses that specify which triggers are active or inactive for the relevant input.	%Mx2000 .. %Mx2070 • Bit0-3: Reserved • Bit4-7: I1-I4 • Bit8: Fast recording • Bit9: Slow recording • Bit10: Start measurement cycle immediately • Bit11: Switch on recording • Bit12: Switch off recording • Bit13: Initiate transmission • Bit14: 0...Trigger is initiated when "greater or equal" to the level, 1...Trigger is initiated when "less or equal" to the level • Bit15: Activate online mode
Register addresses that specify which alarms are active or inactive for the relevant input.	%Mx3000 .. %Mx3070 • Bit0: Warning • Bit1: Alarm • Bit2: Fault warning • Bit3: Fault alarm
Display register addresses	nicht verfügbar

11.6 Example

11.6.1 Saw-tooth generator

This example generates a saw-tooth on the output of a device.

```
LD %MR0          (* Loads the value from the memory register 0 *)
LE 0.0           (* Checks <= 0.0 *)
JMPCN end        (* If the test of the previous line does not equate to 0,
                    jump to the "END" label *)
LD 4.0           (* Load value 4 into the accu *)
ST %MR0          (* Save value 4 from the accu to memory register 0 *)
end:             (* "END" label *)
LD %MR0          (* Loads the value of memory register 0 *)
SUB 0.1          (* Subtract 0.1 from the value in the accu *)
ST %QR0          (* Write the value from the accu to output 0 *)
ST %MR0          (* Write the value from the accu to memory register 0 *)
```



Saw-tooth generated by means of IL

11.6.2 Weekday designation

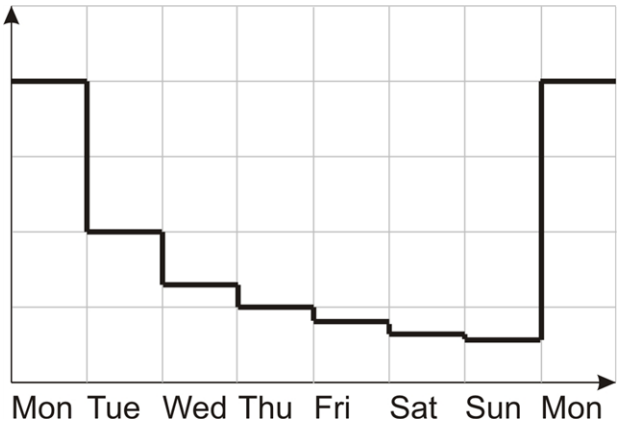
This example determines the weekday from the current time stamp of a device. An interim value is then calculated by dividing the time difference between the cycles of the script by six. With a measurement cycle of 2 minutes this equates to a value of 20. This interim value is then also divided by the determined weekday and the result is issued at output 0.

```
LD %MS0                 (* Loads the time stamp from memory register 0 *)
EQ  DT#0-00-0-0:0:0     (* Checks = 0-00-0-0:0:0 (No time stamp available yet) *)
JMPC END                (* If the test of the previous line equates to 1, jump to
                           the "END" label *)

DOW %MS1001             (* Determine weekday from the current time stamp *)
ADD 1                   (* +1 so that when the division is completed, it is not divided
                           by 0 (Monday) *)

ST %MD2                 (* Save the determined weekday in memory register 2)
LD %MS1001              (* Load current time stamp into the accu *)
SUB %MS0                (* Subtract the time stamp that is saved in memory register
                           0 from the accu *)

DIV T#6s                (* Divide by 6 *)
ST %MT1                 (* Convert the "duration" to a "real number" by
LD %MR1                    saving it in and loading it from memory register 1 *)
DIV %MR2                (* Divide by the weekday *)
ST %QR0                 (* Write the value from the accu to output 0 *)
END:                    (* "END" label *)
LD %MS1001              (* Load current time stamp *)
ST %MS0                 (* Write current time stamp to memory register 0 *)
```



Output signal generated by means of IL

6872
6872
6872

Chapter 12 Pawn script

12.1 General

The following chapter describes the functionality of the pawn script. PAWN (previously SMALL) is a C-similar scripting language that is used on embedded systems.

Additional and more detailed information is provided on the developer's website: http://www-.compuphase.com/pawn/pawn.htm

There are two ways to create a pawn script for the myDatalogMUCmini :

- Direct entry in the "Script" input field in the "Control" configuration section
- Use of a script template that was created on the myDatanet server

12.1.1 Direct input of a pawn script

The pawn script is entered via the "Control" configuration section (see "Control" on page 82) of the input screen for configuring the site. "Pawn" must be selected as the "script type" so that myDatalogMUCmini interprets the commands entered in the "script" as a pawn script.

12.1.2 Uploading a binary file

If the "Upload a compiled script" entry was selected via the "Script source" list selection in the "Control" configuration section (see "Control" on page 82) of the input screen for configuring the site, a binary file that was previously created can be uploaded to the myDatanet server. This is then loaded into the myDatalogMUCmini during the next connection. When using this method, "Pawn" must also be selected as the "script type" so that the myDatalogMUCmini interprets the commands as a pawn script.

> ***Note:*** *If a pawn script is used to write directly to an output, the setpoint entered via the input screen of the myDatanet server is overwritten with the determined value.*

12.2 rapidM2M API

12.2.1 Core functions

native heapspace();

Supplies the free memory capacity to the heap

	Explanation
Return value	*The free memory capacity to the heap. The stack and the heap have a joint memory area, so that this value specifies the number of bytes that remain for the stack or the heap.*

native funcidx(const name[]);

Supplies the index of a public function

Parameter	*Explanation*
name	*Name of the public function*

	Explanation
Return value	• *-1, if there is no function with the transferred name* • *Index of the public function*

native numargs();

Supplies the number of arguments transferred to a function

	Explanation
Return value	*The number of arguments that have been transferred to a function. numargs is useful within functions with a variable list of arguments.*

native getarg(arg, index=0);

Supplies the value of the argument

Parameter	*Explanation*
arg	*The sequence number of the argument. Use 0 for the first argument.*
index	*Index if "arg" refers to an array*

	Explanation
Return value	*This function supplies an argument from a variable argument list. If the argument is an array, the "index" specifies the index of the required array element. The return value is the value of the argument.*

native setarg(arg, index=0, value);

Sets the value of the argument

Parameter	*Explanation*
arg	*The sequence number of the argument. Use 0 for the first argument.*
index	*Index if "arg" refers to an array*
value	*Value to which the argument should be set*

	Explanation
Return value	• *true, if the value could be set* • *false, if the argument or index are invalid* *This function sets an argument in a variable argument list. If the argument is an array, the "index" specifies the index of the required array element.*

native tolower(c);

Converts a character into lower case

Parameter	*Explanation*
c	*Character that should be converted to lower case*

	Explanation
Return value	*The lower case variant of the transferred character, if available, or the unchanged character code of "c" if the letter "c" does not have a lower case equivalent.*

native toupper(c);

Converts a character into upper case

Parameter	*Explanation*
c	*Character that should be converted to upper case*

	Explanation
Return value	*The upper case variant of the transferred character, if available, or the unchanged character code of "c" if the letter "c" does not have a upper case equivalent.*

native swapchars(c);

Swaps the order of the bytes

Parameter	*Explanation*
c	*Value for which the bytes should be swapped over*

	Explanation
Return value	*Value for which the bytes in parameter "c" are swapped over (the lowest byte becomes the highest byte)*

native min(value1, value2);

Supplies the smaller of the two transferred values

Parameter	*Explanation*
value1	*Two values of which the smaller one is to be determined*
value2	

	Explanation
Return value	*The smaller of the two transferred values*

native max(value1, value2);

Supplies the larger of the two transferred values

Parameter	*Explanation*
value1	*Two values of which the larger one is to be determined*
value1	

	Explanation
Return value	*The larger of the two transferred values*

native clamp(value, min=cellmin, max=cellmax);

Checks whether the transferred value is between "min" and "max"

Parameter	*Explanation*
value	*Value that is to be checked*
min	*Lower limit*
max	*Upper limit*

	Explanation
Return value	• *"value" if the value is between "min" and "max"* • *"min" is the value is less than "min"* • *"max", if the value is greater than "max"*

12.2.2 Console functions

Important note: *The Com2 (115200 8N1) interface is used as the standard output on the myDatalogMUCmini if it is not being used otherwise.*

native print(const string[]);

Prints the specified string to the standard output

Parameter	*Explanation*
string	*The character string to be issued. This can include escape sequences.*

	Explanation
Return value	*OK*

native printf(const format[], {Float,Fixed,_}:...);

Prints the transferred format string to the standard output. The mode of operation of the functions corresponds to that of the standard ANSI-C implementation.

Parameter	*Explanation*
format[]	*The format character string to be used*

	Explanation
Return value	• *Number of printed characters* • *ERROR, if not successful*

Important note: *Although the following two functions will still be supported for the purpose of downward compatibility, they should no longer be used for devices with modem version 03v006 or higher.*

native printi(value);

Prints the specified value on the standard output

Parameter	*Explanation*
value	*The integer type value to be issued*

12.2.3 Mdn functions

12.2.3.1 Arrays with symbolic indices

Mdn_TablePoint

Two-column reference point table, float data type

```
// key     Column that is searched
// value   Column with the result values that need to be returned

#define Mdn_TablePoint[Float:.key, Float:.value]
```

Mdn_AlarmCfg

Alarm configuration of a measurement channel

```
//Relevant for all channel modes except for "Digital"
// WarnValueLow     A warning is triggered,
//                  if the measurement value drops to or below this value.

// WarnValueHigh    A warning is triggered,
//                   if the measurement value meets or exceeds this value.
// AlarmValueLow    An alarm is triggered,
//                  if the measurement value drops to or below this value.
// AlarmValueHigh   An alarm is triggered,
//                  if the measurement value meets or exceeds this value.
// Hyst             Hysteresis for all-clear in event of alarm/warning

//Relevant for the "Digital" channel mode
// Flags            MDN_FLG_WARNING:   "High" triggers a "warning".
//                  MDN_FLG_ALARM:     "High" triggers an "alarm".
//                  MDN_FLG_FAULTLOW:  "High" triggers a "fault warning".
//                  MDN_FLG_FAULTHIGH: "High" triggers a "fault alarm".

#define Mdn_AlarmCfg[ Float:.WarnValueLow, Float:.WarnValueHigh,
                      Float:.AlarmValueLow, Float:.AlarmValueHigh,
                      Float:.Hyst, .Flags                         ]
```

Mdn_TriggerCfg

Trigger configuration of a measurement channel

```
// Mode    Trigger mode (see "Trigger mode" in chapter
//         "Constants" on page 173)
// Flags   Indicates which actions should be initiated when the trigger
//         is active (see "Trigger flags" in chapter
//         "Constants" on page 173)
// Value   Levels for initiating the trigger

#define Mdn_TriggerCfg[.Mode, .Flags, Float:.Value]
```

12.2.3.2 Constants

Return codes for general purposes

```
OK                      = 0,
ERROR                   = -1,
```

Error codes for the "Mdn_TablePoint" function

```
MDN_TAB_ERR_FLOOR = -1, // searched value lower than the first table entry
MDN_TAB_ERR_CEIL  = -2, // searched value higher than the last table entry
```

Status of the measurement value/measurement channel

Coding to indicate various error statuses

```
MDN_STATUS_OK  = 0,   // Value is OK.
MDN_STATUS_NAN = 1,   // Invalid measurement value / undefined error
MDN_STATUS_OF  = 2,   // The measurement value is above the upper limit (over-
flow).
MDN_STATUS_UF  = 3,   // The measurement value is below the lower limit
(underflow).
MDN_STATUS_OL  = 4,   // Cable break was detected or no sensor was connected.
MDN_STATUS_SC  = 5,   // Short circuit was detected.
```

Alarm flags

Flags to specify the alarm type to be set for the Mdn_SetAlarm() function or to determine the alarm type supplied by the Mdn_GetAlarmCfg() function.

```
MDN_FLG_WARNING   = 0b00000001,  // Warning
MDN_FLG_ALARM     = 0b00000010,  // Alarm
MDN_FLG_FAULTLOW  = 0b00000100,  // Technical error of low priority
MDN_FLG_FAULTHIGH = 0b00001000,  // Technical error of high priority
MDN_FLG_UNDERFLOW = 0b10000000,  // Bit set:  Alarm/warning, as
                                 //               value <= level
                                 // bit deleted: Alarm/warning, as
                                 //               value >= level
```

Trigger flags

Trigger flags for the Mdn_SetTrigger() function, to evaluate the return value of the Mdn_GetTrigger() function and to evaluate the trigger configuration of a channel read by the Mdn_GetTriggerCfg() function

```
MDN_TRG_INTERN1    = 0b0000000000010000, // Device-specific
MDN_TRG_INTERN2    = 0b0000000000100000, // Device-specific
MDN_TRG_INTERN3    = 0b0000000001000000, // Device-specific
MDN_TRG_INTERN4    = 0b0000000010000000, // Device-specific
MDN_TRG_RECQUICK   = 0b0000000100000000, // Request fast recording.
MDN_TRG_RECSLOW    = 0b0000001000000000, // Request slow recording.
MDN_TRG_RECORD_ON  = 0b0000100000000000, // Start recording the current
                                         // measurement
MDN_TRG_RECORD_OFF = 0b0001000000000000, // Prevent current measurement from
                                         // being recorded
MDN_TRG_TXSTART    = 0b0010000000000000, // Request a single connection to the
                                         // server
MDN_TRG_CONTINOUS  = 0b1000000000000000, // Request an online connection to the
                                         // server
```

Trigger mode

To evaluate the trigger configuration of a channel read by the Mdn_GetTriggerCfg() function

```
MDN_TRG_MODE_NONE             = 0, // Channel deactivated, no trigger

//Relevant for all channel modes except for "Digital"
MDN_TRG_MODE_LESS_OR_EQUAL    = 1, // Initiation of the trigger, if
                                   // value <= level
MDN_TRG_MODE_GREATER_OR_EQUAL = 2, // Initiation of the trigger, if
                                   // value >= level

//Relevant for the "Digital" channel mode
MDN_TRG_MODE_DI_RISING_EDGE   = 3, // Rising edge initiates the trigger.
MDN_TRG_MODE_DI_FALLING_EDGE  = 4, // Falling edge initiates the trigger.
MDN_TRG_MODE_DI_BOTH_EDGES    = 8, // Both edges initiate the trigger.
```

Channel modes of the universal inputs

```
MDN_MODE_IN_NONE       = 0,        // Channel deactivated
MDN_MODE_IN_DIGITAL    = 1,        // Digital
MDN_MODE_IN_DCTRDAY    = 2,        // Day counter
MDN_MODE_IN_DCTRCONT   = 3,        // Interval counter
MDN_MODE_IN_DFREQ      = 4,        // Frequency
MDN_MODE_IN_DPWM       = 5,        // PWM
MDN_MODE_IN_A420MA     = 6,        // 4-20mA
MDN_MODE_IN_A020MA     = 7,        // 0-20mA
MDN_MODE_IN_A002V      = 8,        // 0-2V
MDN_MODE_IN_A010V      = 9,        // 0-10V
```

Channel modes of the output channels

```
MDN_MODE_OUT_NONE      = 100,      // Channel deactivated
MDN_MODE_OUT_EXTWARMUP = 101,      // Ext. warmup time
MDN_MODE_OUT_DIGITAL   = 102,      // Digital output
MDN_MODE_OUT_DFREQ     = 103,      // Frequency output
MDN_MODE_OUT_DPWM      = 104,      // PWM
MDN_MODE_OUT_DIMPULS   = 105,      // Pulse output
MDN_MODE_OUT_A420MA    = 106,      // 4-20mA
MDN_MODE_OUT_A020MA    = 107,      // 0-20mA
```

Serial events

Events that have to be processed by the Mdn_SerialEvent() function

```
MDN_SERIAL_EVENT_INIT    = 0,    // Script should send the init sequence.
                                 // (e.g. following activation of the sensor)
MDN_SERIAL_EVENT_MEASURE = 1,    // Script should send the measurement command.
MDN_SERIAL_EVENT_TIMEOUT = 2,    /* Timeout upon receipt of the answer to the
                                    measurement command */
```

12.2.3.3 Functions

native Mdn_GetCh(ch, &Float:value, &Mdn_ValueStatus:status = MDN_STATUS_OK);

Reads out the current value and status of a channel

Parameter	*Explanation*
ch	*Number of the channel (see "Device-dependent channel numbers" on page 199)*
value	*Value of the channel*
status	*Status of the channel (see "Status of the measurement value/measurement channel" in the chapter "Constants" on page 173) - OPTIONAL*

	Explanation
Return value	• *OK, if successful* • *ERROR, if no valid channel number was transferred*

native Mdn_GetChScale(ch, &Float:min, &Float:max, &Float:scale);

Reads the parameters of channel scaling and saves it in the transferred variables

<table>
<tr><th>Parameter</th><th>Explanation</th></tr>
<tr><td>ch</td><td>Number of the channel (see "Device-dependent channel numbers" on page 199)</td></tr>
<tr><td>min</td><td>
<table>
<tr><th>Channel type</th><th>Mode</th><th>Explanation</th></tr>
<tr><td>Universal inputs</td><td>Digital
Cnt.Day
Cnt.Intervl.
Freq</td><td>Not relevant</td></tr>
<tr><td></td><td>PWM
4-20 mA
0-20 mA
0-2 V
0-10 V</td><td>Start of the measurement range in the measurement unit</td></tr>
<tr><td>Internal measurement channel</td><td>---</td><td>---</td></tr>
<tr><td>Analogue output</td><td>All modes</td><td>Start of the output range in the measurement unit</td></tr>
<tr><td>Isolated switch contact</td><td>---</td><td>---</td></tr>
<tr><td>Interface channels</td><td>If scaling is "on"</td><td>Start of the measurement range in the measurement unit</td></tr>
<tr><td>Interface output channels</td><td>If scaling is "on"</td><td>Start of the output range in the measurement unit</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><th>Parameter</th><th>Explanation</th></tr>
<tr><td>max</td><td>
<table>
<tr><th>Channel type</th><th>Mode</th><th>Explanation</th></tr>
<tr><td>Universal inputs</td><td>Digital
Cnt.Day
Cnt.Intervl.
Freq</td><td>Not relevant</td></tr>
<tr><td></td><td>PWM
4-20 mA
0-20 mA
0-2 V
0-10 V</td><td>End of the measurement range in the measurement unit</td></tr>
<tr><td>Internal measurement channel</td><td>---</td><td>---</td></tr>
<tr><td>Analogue output</td><td>All modes</td><td>End of the output range in the measurement unit</td></tr>
<tr><td>Isolated switch contact</td><td>---</td><td>---</td></tr>
<tr><td>Interface channels</td><td>If scaling is "on"</td><td>End of the measurement range in the measurement unit</td></tr>
<tr><td>Interface output channels</td><td>If scaling is "on"</td><td>End of the output range in the measurement unit</td></tr>
</table>
</td></tr>
</table>

Parameter	Explanation
scale	

Channel type	Mode	Explanation
Universal inputs	Digital	1: Inverting "off" -1: Inverting "on"
	Cnt.Day Cnt.Intervl.	Metered measurand of a pulse in the measurement unit
	Freq	Factor by which the input signal is multiplied
	PWM 4-20 mA 0-20 mA 0-2 V 0-10 V	Not relevant
Internal measurement channel	---	---
Analogue output	All modes	Not relevant
Isolated switch contact	---	---
Interface channels	---	Not relevant
Interface output channels	---	Not relevant

	Explanation
Return value	• OK, if successful • ERROR, if no valid channel number was transferred

native Mdn_GetChMode(ch);

Returns the set mode of the channel

Parameter	Explanation
ch	Number of the channel (see "Device-dependent channel numbers" on page 199)

	Explanation
Return value	• Set channel mode (see "Channel modes of the universal inputs" or "Channel modes of the output channels" in chapter "Constants" on page 173) • ERROR, if no valid channel number was transferred

native Mdn_SetCh(ch, Float:value, Mdn_ValueStatus:status = MDN_STATUS_OK);

Sets the value and status of a channel. This function can also be used to change the value of an input before it is saved. This is possible, as the script is processed once the measurement value has been acquired but before it is recorded (see "Internal processing of the measurement values" on page 31).

Parameter	*Explanation*
ch	*Number of the channel (see "Device-dependent channel numbers" on page 199)*
value	*Value of the channel*
status	*Status of the channel (see "Status of the measurement value/measurement channel" in the chapter "Constants" on page 173) - OPTIONAL*

	Explanation
Return value	• *OK, if successful* • *ERROR, if no valid channel number was transferred*

native Mdn_GetTime(&hour=0, &minute=0, &second=0, timestamp=0);

If no time stamp was transferred (timestamp=0), the current system time (in local time) is converted to hours/minutes/seconds. Alternatively, the transferred time stamp is converted to hours/minutes/seconds.

Parameter	*Explanation*
hour	*Variable to store the hours - OPTIONAL*
minute	*Variable to store the minutes - OPTIONAL*
second	*Variable to store the seconds - OPTIONAL*
timestamp	*Time stamp that should be converted* *= 0: The current system time (in local time) is converted.* *› 0: The transferred time stamp is converted.* *(The time stamp must be specified in seconds since 31.12.1999.)*

	Explanation
Return value	• *timestamp = 0: Seconds since 31.12.1999 (current system time in local time)* • *timestamp › 0: The transferred time stamp is returned.*

native Mdn_GetDate(&year=0, &month=0, &day=0, timestamp=0);

If no time stamp was transferred (timestamp=0), the date (year, month, day) is determined for the current system time (in local time). Alternatively, the date (year, month, day) is determined for the transferred time stamp.

Parameter	*Explanation*
year	*Variable to store the year - OPTIONAL* ***Note:*** *The year specified relates to the 21st century, i.e. the value 14 refers to the year 2014.*
month	*Variable to store the month - OPTIONAL*
day	*Variable to store the day - OPTIONAL*
timestamp	*Time stamp for which the date should be determined* *= 0: The date for the current system time (in local time) is determined.* *› 0: The date for the transferred time stamp is determined.* *(The time stamp must be specified in seconds since 31.12.1999.)*

	Explanation
Return value	• *timestamp = 0: Seconds since 31.12.1999 (current system time in local time)* • *timestamp › 0: The transferred time stamp is returned.*

native Mdn_GetTimezoneOffset();

Returns the difference (in seconds) between UTC and local time configured for the site on the myDatanet server. This can be used to determine the UTC in the script by subtracting the difference from the system time (local time). The offset value is determined by the myDatanet server in accordance with the set time zone (including summer/winter time) and is synchronised during every connection to the device.

Example: Central European time (CET = UTC+1) is used for the site -› Offset = 3600 sec.

	Explanation
Return value	*Offset value in seconds*

native Mdn_DoW(timestamp);

Calculates the weekday from a given timestamp

Parameter	*Explanation*
timestamp	*Timestamp of the day in question*

	Explanation
Return value	*Weekday, 0=Monday ... 6=Sunday*

native Mdn_CalcTable(Float:key, &Float:value, const table[][Mdn_TablePoint], size = sizeof table);

Searches for a certain key in the supplied table and returns the relevant value. The value is interpolated between the two points, if the key is between two values in the table.

Parameter	*Explanation*
key	*Value that is used for the search*
value	*Includes the result of the calculation by the function*
table	*The table that is searched must be a "Mdn_TablePoint" type table.*
size	*Number of elements in the table*

	Explanation
Return value	• *OK, if the relevant value was found* • *MDN_TAB_ERR_FLOOR, if the key is smaller than the first table entry. "value" contains the first table entry.* • *MDN_TAB_ERR_FLOOR, if the key is larger than the last table entry. "value" contains the last table entry.*

native Mdn_GetAlarm(ch);

Returns the alarm status of a channel

Parameter	*Explanation*
ch	*Number of the channel (see "Device-dependent channel numbers" on page 199)*

	Explanation
Return value	*Positive, if successful (see "Alarm flags" in the chapter "Constants" on page 173), ERROR, if no valid channel was specified*

native Mdn_GetAlarmCfg(ch, Config[Mdn_AlarmCfg]);

Returns the alarm configuration of a channel

Parameter	*Explanation*
ch	*Number of the channel (see "Device-dependent channel numbers" on page 199)*
Config	*Structure for storing the alarm configuration (see "Mdn_AlarmCfg" in chapter "Arrays with symbolic indices" on page 172)*

	Explanation
Return value	• *OK, if successful* • *ERROR, if no valid channel number was transferred*

native Mdn_SetAlarm(ch, alarm, Float:value, Float:level = 0.0);

Sets the alarm status of a channel. The "MDN_FLG_UNDERFLOW" alarm flag must be set to indicate that the alarm/warning was triggered because the alarm/warning level was undercut. The "MDN_FLG_UNDERFLOW" alarm flag remains clear to indicate that the alarm/warning was triggered because the alarm/warning level was exceeded. The alarm detection of the system is executed before the script is processed (see "Internal processing of the measurement values" on page 31).

Parameter	*Explanation*
ch	*Number of the channel (see "Device-dependent channel numbers" on page 199)*
alarm	*Alarm flags (see "Alarm flags" in chapter "Constants" on page 173). The value "0" clears the alarm status of the channel.*
value	*Value of the channel used for the alarm calculation*
level	*Contains the alarm threshold - OPTIONAL*

	Explanation
Return value	• *OK, if successful* • *ERROR, if no valid channel number was transferred*

native Mdn_SetTrigger(trigger);

Sets global trigger events

Parameter	*Explanation*
trigger	*Trigger flags (see "Trigger flags" in chapter "Constants" on page 173)*

	Explanation
Return value	*OK, if successful*

native Mdn_GetTrigger();

Returns the global trigger flags

	Explanation
Return value	*Global trigger flags (see "Trigger flags" in chapter "Constants" on page 173)*

native Mdn_GetTriggerCfg(ch, Config[Mdn_TriggerCfg]);

Returns the trigger configuration of a channel

Parameter	*Explanation*
ch	*Number of the channel (see "Device-dependent channel numbers" on page 199)*
Config	*Structure for storing the trigger configuration (see "Mdn_TriggerCfg" in chapter "Arrays with symbolic indices" on page 172)*

	Explanation
Return value	• *OK, if successful* • *ERROR, if no valid channel number was transferred*

native Mdn_WriteLog(log, param);

Generates an entry in the device log

Parameter	*Explanation*
log	*Log entry to be generated (valid range 0...999). Some error codes have already been predefined in chapter "Predefined log entries" on page 200. If possible, you should use these for the described error situations.* ***Important note:** The log entries are mapped in the "MODULE ERR" (2000-2999) area for display on the server.*
param	*Additional parameter for the exact specification of the log entry (valid range -32768 .. 32767)*

	Explanation
Return value	*OK*

forward public Mdn_CtrlFinish();

This function is called up at the time of every measurement once all of the measurement values have been generated. It should comprise all of the calculations and functions that should be performed cyclically.

native Mdn_SetPacked(data{}, pos, &{Float,Fixed,_}:value, size=4, bool:bigendian=false);

Writes the transferred value to a specified position in an array

Parameter	*Explanation*
data	*Array that should contain the data*
pos	*Byte offset within the array to determine the position where the value should be written*
value	*Value that should be written in the array*
size	*Number of bytes that should be used for the value to be written*
bigendian	*Settings for the byte order that should be used when writing the value:* *true: "Big endian" is used* *false: "Little endian" is used*

	Explanation
Return value	• *OK, if successful* • *ERROR*

Note: *Additional explanation on the byte order:*

In the following example, the whole number 439.041.101 is saved as a 32-bit integer value from memory address 10000.

	Big endian			*Little endian*		
Addresses	*Hex*	*Dez*	*Binary*	*Hex*	*Dez*	*Binary*
10000	*1A*	*26*	*00011010*	*4D*	*77*	*01001101*
10001	*2B*	*43*	*00101011*	*3C*	*60*	*00111100*
10002	*3C*	*60*	*00111100*	*2B*	*43*	*00101011*
10003	*4D*	*77*	*01001101*	*1A*	*26*	*00011010*

native Mdn_SetPackedB(data{}, pos, const block{}, size);

Writes the transferred data block to the specified position in an array

Parameter	*Explanation*
data	*Array that should contain the data*
pos	*Byte offset within the array to determine the position where the data block should be written*
block	*Data block that should be written in the array*
size	*Number of bytes that should be written in the array by the data block*

	Explanation
Return value	• *OK, if successful* • *ERROR*

native Mdn_GetPacked(const data{}, pos, &{Float,Fixed,_}:value, size=4, bool:bigendian=false);

Returns the value that is located at the specified position in an array

Parameter	*Explanation*
data	*Array that contains the data*
pos	*Byte offset within the array to determine the position from which the data should be read*
value	*Variable to store the data to be read*
size	*Number of bytes that should be read*
bigendian	*Specifies how the packed data must be interpreted:* *true: The data is saved in "Big endian" format in the array.* *false: The data is saved in "Little endian" format in the array.*

	Explanation
Return value	• *OK, if successful* • *ERROR*

Note: *Additional explanation on the byte order:*

In the following example, the whole number 439.041.101 is saved as a 32-bit integer value from memory address 10000.

	Big endian			*Little endian*		
Addresses	*Hex*	*Dez*	*Binary*	*Hex*	*Dez*	*Binary*
10000	*1A*	*26*	*00011010*	*4D*	*77*	*01001101*
10001	*2B*	*43*	*00101011*	*3C*	*60*	*00111100*
10002	*3C*	*60*	*00111100*	*2B*	*43*	*00101011*
10003	*4D*	*77*	*01001101*	*1A*	*26*	*00011010*

native Mdn_GetPackedB(const data{}, pos, block{}, size);

Reads a data block that is located at the specified position in an array

Parameter	*Explanation*
data	*Array that contains the data*
pos	*Byte offset within the array to determine the position from which the data should be read*
block	*Array to store the data to be read*
size	*Number of bytes that should be read*

	Explanation
Return value	• *OK, if successful* • *ERROR*

forward public Mdn_SerialEvent(com, event);

This function is the entry point for evaluating the serial events. This function must be provided by the user if script parsing is to be used.

Parameter	*Explanation*
com	*Number of the interface (see "Device-specific interface numbers" on page 199)*
event	*Event that caused this function to be called up* • *MDN_SERIAL_EVENT_INIT (Script should send the init sequence.)* • *MDN_SERIAL_EVENT_MEASURE (Script should send the measurement command.)* • *MDN_SERIAL_EVENT_TIMEOUT (Timeout upon receipt of the answer to the measurement command)*

forward public Mdn_SerialRx(com, const data{}, len);

This function is the entry point for evaluating the received characters. This function must be provided by the user if script parsing is to be used.

Parameter	*Explanation*
com	*Number of the interface (see "Device-specific interface numbers" on page 199)*
data	*Array that contains the received characters*
len	*Number of characters that were received (max. 256)*

native Mdn_SerialTx(com, const data{}, len);

Sends the transferred data block via the specified interface

Parameter	*Explanation*
com	*Number of the interface (see "Device-specific interface numbers" on page 199)*
data	*Array that contains the data to be sent*
len	*Number of bytes that should be sent*

	Explanation
Return value	• *OK, if successful* • *ERROR*

native Mdn_SerialFinish(com);

Terminates the receipt of data via the specified interface. This function must always be called up once all of the required data has been received or the timeout event has occurred.

Parameter	*Explanation*
com	*Number of the interface (see "Device-specific interface numbers" on page 199)*

	Explanation
Return value	• *OK, if successful* • *ERROR*

12.2.4 Mathematical functions

Helpful constants

Definition	Value	Description
M_E	2.7182818284590452354	e
M_LOG2E	1.4426950408889634074	$\log_2 e$
M_LOG10E	0.43429448190325182765	$\log_{10} e$
M_LN2	0.69314718055994530942	ln 2
M_LN10	2.30258509299404568402	ln 10
M_PI	3.14159265358979323846	π
M_PI_2	1.57079632679489661923	π/2
M_PI_4	0.78539816339744830962	π/4
M_1_PI	0.31830988618379067154	1/π
M_2_PI	0.63661977236758134308	2/π
M_2_SQRTPI	1.12837916709551257390	2/sqrt(π)
M_SQRT2	1.41421356237309504880	sqrt(2)
M_SQRT1_2	0.70710678118654752440	1/sqrt(2)

native fround(Float:x);

Commercially rounds the transferred float

Parameter	*Explanation*
x	*Float that should be rounded*

	Explanation
Return value	*Commercially rounded integral value*

The mode of operation of the following functions corresponds to that of the standard ANSI-C implementation:

native Float:sin(Float:x);

Sine of x

native Float:cos(Float:x);

Cosine of x

native Float:tan(Float:x);

Tangent of x

native Float:asin(Float:x);

Arcsine(x) in the range [-π/2, π/2], x element of [-1, 1]

native Float:acos(Float:x);
Arccosine(x) in the range [0, π], x element of [-1, 1]

native Float:atan(Float:x);
Arctangent(x) in the range [-π/2, π/2]

native Float:atan2(Float:y, Float:x);
Arctangent(y/x) in the range [-π, π]

native Float:sinh(Float:x);
Hyperbolic sine of x

native Float:cosh(Float:x);
Hyperbolic cosine of x

native Float:tanh(Float:x);
Hyperbolic tangent of x

native Float:exp(Float:x);
Exponential function e^x

native Float:log(Float:x);
Natural logarithm ln(x), x > 0

native Float:log10(Float:x);
Logarithm as the basis 10 $\log_{10}(x)$, x > 0

native Float:pow(Float:x, Float:y);
x^y. An argument error has occurred if x = 0 and y <= 0, or if x < 0 and y is not a whole number.

native Float:sqrt(Float:x);
Root x, x >= 0

native Float:ceil(Float:x);
Smallest whole number that is not smaller than x

native Float:floor(Float:x);
Largest whole number that is not larger than x

native Float:fabs(Float:x);
Absolute value | x |

native Float:ldexp(Float:x, n);
$x*2^n$

native Float:frexp(Float:x, &n);
Breaks down x into a normalised mantissa in the range [1/2, 1] that is supplied as the result, and a potency of 2 that is filed in n. If x is zero, both parts of the result are zero.

native Float:modf(Float:x, &Float:ip);
Breaks down x into an integral and residual part that both have the same prefix as x. The integral part is filed in ip, while the residual part is the result.

native Float:fmod(Float:x, Float:y);
Residual floating point of x/y with the same prefix as x. The result is dependent on the implementation, if y is zero.

native isnan(Float:x);
Returns a value that is not equal to zero, if x is not a number

12.2.5 String functions

> ***Note:*** *You require the following include file to be able to use the functions in this chapter:*
> `#include <string>`

The mode of operation of the following functions essentially corresponds to that of the standard ANSI-C implementation:

native strlen(const string[]);

Supplies the length of the string (without '\0')

Parameter	*Explanation*
string	*Character string for which the length has to be determined*

	Explanation
Return value	*Number of characters without the final '\0'*

native sprintf(dest[], maxlength=sizeof dest, const format[], {Float,Fixed,_}:...);

Saves the transferred format string in the array dest. The mode of operation of the functions corresponds to that of the "snprintf" function of the standard ANSI-C implementation.

Parameter	*Explanation*
dest	*Array to store the formatted result*
maxlength	*Maximum number of characters that the array dest can store*
format	*The format character string to be used*

	Explanation
Return value	• *-1 in the event of a fault* • *Number of characters that would have been written if the array dest had been long enough (without '\0').* *The array dest is always assigned a final zero. The length of the array dest cannot be exceeded.*

native strcpy(dest[], const source[], maxlength=sizeof dest);

Copies the source character string to the array dest (including '\0').

Parameter	*Explanation*
dest	*Array to store the character string that should be copied*
source	*Character string that should be copied*
maxlength	*Number of characters to be copied - OPTIONAL*

	Explanation
Return value	*Number of copied characters*

native strcat(dest[], const source[], maxlength=sizeof dest);

Adds the source character string to the dest character string (including '\o')

Parameter	*Explanation*
dest	*Array to store the result. This array already comprises one character string to which the source character string should be added.*
source	*Character string that should be added to the character string included in the array dest*
maxlength	*Number of characters to be added - OPTIONAL*

	Explanation
Return value	*Number of added characters*

native strcmp(const string1[], const string2[], length=cellmax);

Compares character string1 and string2

Parameter	*Explanation*
string1	*The two character strings that are to be compared*
string2	
length	*The maximum number of characters that should be taken into consideration during the comparison - OPTIONAL*

	Explanation
Return value	• *1: string1 › string 2* • *0: both of the character strings are the same (at least the length that is taken into account)* • *-1: string1 ‹ string 2*

native strchr(const string[], char);

Searches for a character (first occurrence) in a character string

Parameter	*Explanation*
string	*Character string that should be searched*
char	*Character that the search is looking for*

	Explanation
Return value	• *-1, if the character that the search is looking for is not included in the character string* • *Array index for the character that the search is looking for (first character occurring in the character string)*

native strrchr(const string[], char);

Searches for a character (last occurrence) in a character string

Parameter	*Explanation*
string	*Character string that should be searched*
char	*Character that the search is looking for*

	Explanation
Return value	• *-1, if the character that the search is looking for is not included in the character string* • *Array index for the character that the search is looking for (last character occurring in the character string)*

native strspn(const string1[], const string2[]);

Searches for the position of the first character in string1 that is ***not*** *included in the character string of permitted characters (string2)*

Parameter	*Explanation*
string1	*Character string that should be searched*
string2	*Character string of permitted characters*

	Explanation
Return value	• *Length of string1 if no permitted characters are found* • *Position of the first character in the character string that should be searched that is not included in the character string of permitted characters*

native strcspn(const string1[], const string2[]);

Searches for the position of the first character in string1 that is also included in the character string of permitted characters (string2)

Parameter	*Explanation*
string1	*Character string that should be searched*
string2	*Character string of permitted characters*

	Explanation
Return value	• *Length of string1 if no permitted character has been found* • *Position of the first character in the character string that should be searched that is also included in the character string of permitted characters*

native strpbrk(const string1[], const string2[]);

Searches the array index of the first character that is also included in the character string of permitted characters

Parameter	*Explanation*
string1	*Character string that should be searched*
string2	*Character string of permitted characters*

	Explanation
Return value	• *-1, if the character that the search is looking for is not included in the character string* • *Array index of the first character in the character string that should be searched that is also included in the character string of permitted characters*

native strstr(const string1[], const string2[]);

Searches character string2 in character string1

Parameter	*Explanation*
string1	*Character string that should be searched*
string2	*Character string that the search should be for*

	Explanation
Return value	• *-1, if character string2 that is being searched for is not included in string1* • *Array index where character string2 that is being searched for starts in string1*

native strtol(const string[], base);

Converts a character string into a value

Parameter	*Explanation*
string	*Character string that is to be converted*
base	*Specifies the basis that must be used for the conversion* *2-36: The specified basis is used* *0: 8, 10 or 16 is used as the basis, depending on the character string that must be converted* *Basis 8: with a leading 0* *Basis 16: with 0x or 0X*

	Explanation
Return value	*Value that corresponds to the character string*

native Float: atof(const string[]);

Converts a character string into a float

Parameter	*Explanation*
string	*Character string that is to be converted*

	Explanation
Return value	*Float for which the numerical value corresponds to the character string*

12.2.6 Help functions

native getapilevel();

Issues the implemented API level of the script engine

	Explanation
Return value	*Implemented API level of the script engine*

native CRC16(data{}, len);

Supplies the calculated modbus CRC16 of the transferred data

Parameter	*Explanation*
data	*Array that contains the data for which the CRC16 must be calculated*
len	*Number of bytes that must be taken into consideration during the calculation*

	Explanation
Return value	*Calculated CRC16*

native CRC32(data{}, len);

Supplies the calculated Ethernet CRC32 of the transferred data

Parameter	*Explanation*
data	*Array that contains the data for which the CRC32 must be calculated*
len	*Number of bytes that must be taken into consideration during the calculation*

	Explanation
Return value	*Calculated CRC32*

native LRC(data[], len);

Returns the calculated modbus LRC of the transferred data

Parameter	*Explanation*
data	*Array that contains the data for which the LRC must be calculated*
len	*Number of bytes that must be taken into consideration during the calculation*

	Explanation
Return value	*Calculated LRC*

12.3 Device-dependent channel numbers

Definition of the constants	Explanation
MDN_CH_UI1 : MDN_CH_UI3	Universal inputs 1-3
MDN_CH_GSM	Internal measurement channel "GSM level"
MDN_CH_VIN	Internal measurement channel "Voltage"
MDN_CH_IOUT1	Analogue output
MDN_CH_REL1	Isolated switch contact
MDN_CH_IN1 : MDN_CH_IN64	Interface channels 1-64
MDN_CH_OUT1 : MDN_CH_OUT64	Interface output channels 1-64
MDN_NUM_IN_CHANNELS	Total number of interface channels
MDN_NUM_OUT_CHANNELS	Total number of interface output channels
MDN_NUM_CHANNELS	Total number of all channels (all of the in- and outputs)

12.4 Device-specific interface numbers

Definition of the constants	Explanation
MUCMINI_ITF_COM1	COM1 (RS485)
MUCMINI_ITF_COM2	COM2 (RS232)

12.5 Predefined log entries

Log entry		Parameter		Description
Code	Plain text	Code	Plain text	
0	LOG_COM2_ERROR	0	CHAR TIMEOUT	Com2: The time lag between two received characters is larger than the permissible character timeout.
		1	FRAME TIMEOUT	Com2: The complete answer was not received within the frame timeout.
		2	INVALID FRAME	Com2: The answer frame of the sensor is invalid.
1	LOG_CHAR_TIME-OUT	## [1)]	---	The time lag between two received characters is larger than the permissible character timeout.
2	LOG_FRAME_TIME-OUT	##[1)]	---	The complete answer was not received within the frame timeout.
3	LOG_FRAME_ INVALID	##[1)]	---	The answer frame of the sensor is invalid.

12.6 Pawn script error codes

The PAWN script is run through at the time of every measurement once all of the measurement values have been generated. If an error occurs during this process, the script execution is stopped and deactivated. The relevant error code is also entered in the device log and a connection to the server is established. The parameter for all log entries except "SCRIPT_ERR" contains the 32-bit instruction pointer of the PAWN abstract machine (AMX). Two entries are generated in the device log as only 16-bit values can be saved in the parameter of a log entry. The first entry contains Bit31-Bit16 and the second entry contains Bit15-Bito of 32-bit instruction pointer. Instructions on evaluating the device log are included in the chapter "Evaluating the device log" (see "Evaluating the device log" on page 310).

Log entry		Parameter		Description
Code	Plain text	Code	Plain text	
3000	SCRIPT_ERR	0	NO SCRIPT	No valid script available
		1	SCRIPT UPDATE	New script received
3001	AMX_ERR_EXIT	##	---	Abortion e.g. max. number of PAWN commands (100,000) per run reached
3002	AMX_ERR_ASSERT	##	---	Assertion failed

Log entry		Parameter		Description
Code	Plain text	Code	Plain text	
3003	AMX_ERR_STACK-ERR	##	---	Stack/heap collision (insufficient stack size)
3004	AMX_ERR_BOUNDS	##	---	Array index outside the valid range
3005	AMX_ERR_MEMAC-CESS	##	---	Invalid memory access e.g. mix-up between cell (32-bit element) access [] and byte access {}
3006	AMX_ERR_INVINSTR	##	---	Invalid statement
3007	AMX_ERR_STACK-LOW	##	---	Stack underflow
3008	AMX_ERR_HEA-PLOW	##	---	Heap underflow
3009	AMX_ERR_CALL-BACK	##	---	No (invalid) native callback function
3010	AMX_ERR_NATIVE	##	---	Native function failed
3011	AMX_ERR_DIVIDE	##	---	Divide by zero
3012	AMX_ERR_SLEEP	##	---	Sleep mode
3013	AMX_ERR_INV-STATE	##	---	Invalid state
3014	reserved			
3015	reserved			
3016	AMX_ERR_MEMORY	##	---	Out of memory
3017	AMX_ERR_FORMAT	##	---	P-code file format is invalid/not supported
3018	AMX_ERR_VERSION	##	---	File is for a newer version of AMX
3019	AMX_ERR_NOT-FOUND	##	---	File or function not found
3020	AMX_ERR_INDEX	##	---	Invalid index parameter (invalid entry point)
3021	AMX_ERR_DEBUG	##	---	Debugger cannot be executed
3022	AMX_ERR_INIT	##	---	AMX not initialised (or initialised twice)
3023	AMX_ERR_USER-DATA	##	---	User data field cannot be set (table full)
3024	AMX_ERR_INIT_JIT	##	---	JIT cannot be initialised.
3025	AMX_ERR_PARAMS	##	---	Faulty parameter
3026	AMX_ERR_DOMAIN	##	---	Domain error. The result of the expression is not in the valid range.

Log entry		Parameter		Description
Code	Plain text	Code	Plain text	
3027	AMX_ERR_GENERAL	##	---	General error (invalid or non-specific error)
3028	AMX_ERR_OVERLAY	##	---	Overlays are not supported (JIT) or are not initialised.

12.7 Syntax

12.7.1 General syntax

12.7.1.1 Format

Identifiers, numbers and characters are separated by spaces, tabs, line breaks and "form feed". A series of one or more of these separators is recognised as an empty space.

12.7.1.2 Optional semicolons

Semicolons (used to finish a statement) are optional if they are at the end of a line. Semicolons are required to separate several statements in a line. An expression can be split across several lines, though the postfix operators must be on the same line as the operand.

12.7.1.3 Comments

Text between the /* and */ symbols (both symbols can be on the same or different lines) and text following // (to the end of the line) are comments. Comments must not be nested. The compiler considers comments to be blank space. A documentation comment is a comment that starts with "/** " (two stars and space after the second star) and ends with "*/". A comment that starts with "/// " (three forward slashes and a space after the third slash) is also a documentation comment. The parser can support the documentation comment in different ways, for example, by using it to generate online help.

12.7.1.4 Identifier

Names of variables, functions and constants. Identifier comprises the characters a...z, A...Z, 0...9, _ or @. The first character must not be a number. The characters @ and _ on their own are not valid identifiers, e.g. "_Up" is a valid identifier but "_" is not. Pawn distinguishes between upper and lower case. The parser cuts identifiers off after a certain length. By default, only the first 16 characters are referenced for distinguishing purposes.

12.7.1.5 Reserved keywords

Statements	Operator	Directives	Others
assert	defined	defined	defined
break	sizeof	sizeof	sizeof
case	state	state	state
continue	tagof	tagof	tagof
default			
do			
else			
exit			
for			
goto			
if			
return			
sleep			
state			
switch			
while			

12.7.1.6 Numerical constants

12.7.1.6.1 Numerical integer constants

Binary:
0b followed by a series of 0 and 1

Decimal:
A series of numbers between 0 and 9

Hexadecimal:
0x followed by a series of numbers between 0 and 9 and the letters a to f

12.7.1.6.2 Numerical floating-point constants

A floating-point number is a number with numbers after the decimal point. A floating-point number starts with one or several numbers, includes a decimal point and has at least one number after the decimal point, e.g. "12.0" and "0.75" are valid floating-point numbers. An exponent can optionally be added. The notation is the letter "e" (lower case) followed by an integer numerical constant. For example, "3.12e4" or "12.3e-3" are valid floating-point numbers with an exponent.

12.7.2 Variables

12.7.2.1 Declaration

The keyword "new" declares a new variable. For special declarations, the keyword "new" is replaced with "static" (see "Static local declaration" on page 204). The value of the new variable is zero, provided that is not initialised explicitly.

A variable declaration can appear

- At every position at which an expression is valid - local variable
- At every position at which a function declaration or the implementation of the function is valid - global variables
- In the first expression of a "for" loop (see "For (expression 1; expression 2; expression 3) statement " on page 215) - local variable

Example:

```
new a;     // without initialisation (value is 0)
new b = 3; // with initialisation (value is 3)
```

12.7.2.2 Local declaration

A local declaration appears within a statement block. A variable can only be accessed within this block and the blocks that it comprises. A declaration within the first expression of a loop instruction is also a local declaration.

12.7.2.3 Global declaration

A global declaration appears outside of a function and a global variable can be used in any function. Global variables can only be initialised with constant expressions.

12.7.2.4 Static local declaration

A local variable is destroyed if the execution leaves the block in which the variable was created. Local variables in a function only exist during the operating time of the specified function. Each new call up of the function creates and initialises new local variables. The variable will also remain in the memory at the end of the function, if a local variable is declared with the keyword "static" instead of "new". This means that static, local variables provide permanent private storage that can only be accessed by a single function (or block). Static local variables can only be initialised with constant expressions, the same way as global variables.

12.7.2.5 Static global declaration

A static global variable acts in the same way as a global variable with the difference that the variable is only valid in the file in which it was declared. Replace the keyword "new" with "static" to declare a global variable as static.

12.7.2.6 Floating point values

The pawn supports floating point values. These can be added at every point at which a variable declaration is valid.

Example:

```
new Float:a;       // without initialisation (value is 0.0)
new Float:b = 3.0; // with initialisation (value is 3.0)
```

12.7.3 Constant variables

It is sometimes necessary to create a variable that is initialised once and is then not meant to be changed again. Such a variable acts in a similar way to a symbolic constant although it is still a variable. To declare a constant variable, place the keyword "const" between the keyword that starts the variable declaration ("new", "static") and the name of the variables.

Example:

```
new const address[4] = { 192, 0, 168, 66 }
static const status /* initialised to zero */
```

Typical situations in which you could use a constant variable, include:

- To create an "array" constant. Symbolic constants cannot be accessed via the index.
- It is a special case when array arguments are marked as "const" in a function. Arrays arguments are always transferred via a reference. If the arguments are declared to be "const", they are protected against unwanted changes. See examples of "const function arguments" in the chapter "Function arguments ("call-by-value" versus "call-by-reference")" on page 219.

12.7.4 Array variables

12.7.4.1 One-dimensional arrays

The name[constant] syntax declares the name as an array of "constant" elements, where each element is an entry. "name" is a placeholder for the name of the variable and "constant" is a positive value not equal to zero. "constant" is optional and can be omitted. If there is no value between the brackets, the number of elements is equal to the number of initial values. The array index area is "zero-based", which means that the first element is "name[0]" and the last element is "name[constant-1]".

12.7.4.2 Initialisation

Data objects can be initialised during their declaration. The initialised value from global data objects must be a constant value. Global or local arrays must also be initialised with constant values. Data that is not initialised are zero by default.

Example:

List: valid declaration

```
new i = 1
new j                                   /* j is 0 */
new k = 'a'                             /* k has the character code of 'a' */
new a[] = [1,4,9,16,25]                 /* a has 5 elements */
new s1[20] = ['a','b']                  /* the remaining 18 elements are 0 */
new s2[] = ''Hello world...''           /* an unpacked string */
```

List: invalid declaration

```
new c[3] = 4                            /* An array cannot be set to an individual
                                           value */
new i = "Good-bye"                      /* only an array can hold a string */
new q[]                                 /* Unknown size for an array */
new p[2] = { i + j, k - 3 }             /* Array initialisers must be constants */
```

12.7.4.3 Progressive initialisation for arrays

The point operator continues the initialisation of the arrays based on the last two initialised values. The point operator (three points, "...") initialises the array up to the array limit.

Example: List: array initialisers

```
new a[10] = { 1, ... }                  // sets all of the elements to 1
new b[10] = { 1, 2, ... }               // b = 1, 2, 3, 4, 5, 6, 7, 8, 9, 10
new c[8]  = { 1, 2, 40, 50, ... }       // c = 1, 2, 40, 50, 60, 70, 80, 90
new d[10] = { 10, 9, ... }              // d = 10, 9, 8, 7, 6, 5, 4, 3, 2, 1
```

12.7.4.4 Multi-dimensional arrays

(Only arrays with up to three dimensions are supported)

Multi-dimensional arrays are arrays that include references to other sub-arrays. For example, a two-dimensional array is an "array on one-dimensional arrays".

Example for the declaration of two-dimensional arrays:

```
new a[4][3]
new b[3][2] = [ [ 1, 2 ], [ 3, 4 ], [ 5, 6 ] ]
new c[3][3] = [ [ 1 ], [ 2, ...], [ 3, 4, ... ] ]
new d[2]{10} = [ "agreement", "dispute" ]
new e[2][] = [ ''OK'', ''Cancel'' ]
new f[][] = [ ''OK'', ''Cancel'' ]
```

As the last two declarations (variables "e" and "f") illustrates, the last dimension has an unspecified length. In this case, the length of the sub-array is detected by the associated initialiser. Each sub-array is a different length. In this specific example, "e[1][5]" includes the letter "l" of the word "Cancel". However, "e[0][5]" is invalid as the sub-array e[0] only comprises three entries (the letters "O", "K" and the zero terminator). The difference between the declarations of the "e" and "f" arrays is that we enable the compiler to determine the number of higher dimensions for "f". "sizeof f" and "sizeof e" are both 2 (see "Arrays and the "sizeof" operator" on page 207).

12.7.4.5 Arrays and the "sizeof" operator

The "sizeof" operator returns the number of elements of a variable. The "sizeof" result of a simple (non array) variable is always 1.

An array with one dimension comprises a number of elements and the "sizeof" operator returns this quantity. The code section below would therefore issue "5", as the array comprises four characters and the zero terminator.

```
new msg[] = ''Help''
printf(''%d'', sizeof msg);
```

The "sizeof" operator always returns the number of entries even for a "packed" array. The code section below also issues "5", as the variable comprises five entries even though it requires less memory space.

```
new msg{} = "Help"
printf(''%d'', sizeof msg);
```

For multi-dimensional arrays, the "sizeof" operator can return the number of elements for every dimension. An element in the last (lowest) dimension is a single entry, while it is a sub-array in the highest dimension. Please note that in the following code section, the "sizeof matrix" syntax returns the number of elements of the higher dimension and that the "sizeof matrix[]" syntax issues the lower dimension of the two-dimensional array. The code section issues three (higher dimension) and two (lower dimension).

```
new matrix[3][2] = { { 1, 2 }, { 3, 4 }, { 5, 6 } }
printf(''%d %d'', sizeof matrix, sizeof matrix[]);
```

The application of the "sizeof" operator on multi-dimensional arrays is particularly practical when it is used as a standard value for function arguments.

12.7.5 Operators and expressions

12.7.5.1 Notational conventions

The use of some operators is dependent on the relevant type of operand. The following notations are therefore used in this chapter:

e
: *Any expression*

v
: *Any expression that can be assigned a value ("lvalue" expression - variable)*

a
: *An array*

f
: *A function*

s
: *A symbol - this can be a variable, a constant or a function*

12.7.5.2 Expressions

An expression consists of one or several operands with an operator. The operand can be a variable, a constant or another expression. An expression followed by a semicolon is a statement.

Examples of expressions:

```
v++ f(a1, a2)
v = (ia1 * ia2) / ia3
```

12.7.5.3 Arithmetic

Operator	Example	Explanation
+	e1 + e2	Result of adding e1 and e2
-	e1 - e2	Result of subtracting e2 from e1
	-e	Result of the arithmetic negation of e (two's complement)
*	e1 * e2	Result of multiplying e1 with e2
/	e1 / e2	Result of dividing e1 by e2. The result is truncated to the closest whole number that is less or equal to the quotient. Positive and negative values are rounded down (negative infinity).
%	e1 % e2	Result is the remainder of the division of e1 by e2. The prefix is the same as that of e2
++	v++	Increases v by 1. The result of the expression is the value before the increase.
	++v	Increases v by 1. The result of the expression is the value following the increase.
--	v--	Decreases v by 1. The result of the expression is the value before the decrease.
	--v	Decreases v by 1. The result of the expression is the value following the decrease.

> ***Note:*** *The unary + is not defined in the pawn. The operators ++ and -- change the operand. The operand must be a "lvalue".*

12.7.5.4 Bit manipulation

Operator	Example	Explanation
~	~e	The result is the one's complement of e.
››	e1 ›› e2	The result of the arithmetic shift to the right of e1 by e2 bits. The shift is signed: The bit on the far left is copied to the free bits of the result.
›››	e1 ››› e2	The result of the logical shift to the right of e1 by e2 bits. The shift is unsigned. The free bits of the result are filled with 0.
‹‹	e1 ‹‹ e2	Result: Shift to the left of e1 by e2 bits. The free bits of the result are filled with 0. There is no difference between an arithmetic and a logical shift to the left.
&	e1 & e2	The result is the bitwise logical "and" of e1 and e2.
|	e1 | e2	The result is the bitwise logical "or" of e1 and e2.
^	e1 ^ e2	The result is the bitwise logical "exclusive or" of e1 and e2.

12.7.5.5 Assignment

The result of an assignment expression is the value of the operand following the assignment.

Operator	Example	Explanation
=	v = e	Assigns the value of e to the variable v
	v = a	Assigns the array a to variable v. v must be an array of the same size and with the same dimensions as a. a can be a character string or an array.

Note: *The following operators combine an assignment with an arithmetic or bitwise operation. The result of the expression is the value of the left operand following the arithmetic or bitwise operation.*

Operator	Example	Explanation
+=	v += e	Increases v by e
-=	v -= e	Decreases v by e
*=	v *= e	Multiplies v with e
/=	v /= e	Divides v by e
%=	v %= e	Assigns v the remainder of the division of v and e
››=	v ››= e	Arithmetically shifts v to the right by e bits
›››=	v ›››= e	Logically shifts v to the right by e bits
‹‹=	v ‹‹= e	Shifts v to the left by e bits
&=	v &= e	Executes a bitwise "and" from v and e and assigns the result to v
|=	v |= e	Executes a bitwise "or" from v and e and assigns the result to v
^=	v ^= e	Executes a bitwise "exclusive or" from v and e and assigns the result to v

12.7.5.6 Comparative operators

A logical "false" is represented by an integer value of 0; a logical "true" is represented by a value that is not 0. Results of a comparative expression are either 0 or 1 and the "tag" is set to "bool".

Operator	Example	Explanation
==	e1 == e2	The result is "true" if e1 and e2 are the same.
!=	e1 != e2	The result is "true" if e1 and e2 are not the same.

Note: *The following operators can be linked, the same as in the expression "e1 <= e2 <= e3". This means that the result is "1" if every single comparison is true and "0" if at least one comparison is false.*

Operator	Example	Explanation
<	e1 < e2	The result is a logical "true" if e1 is less than e2.
<=	e1 <= e2	The result is a logical "true" if e1 is less or equal to e2.
>	e1 > e2	The result is a logical "true" if e1 is greater than e2.
>=	e1 >= e2	The result is a logical "true" if e1 is greater or equal to e2.

12.7.5.7 Boolean

A logical "false" is represented by an integer value of 0; a logical "true" is represented by a value that is not 0. Results of a comparative expression are either 0 or 1 and the "tag" is set to "bool".

Operator	Example	Explanation
!	!e	The result is a logical "true", if e is logical "false".
||	e1 || e2	The result is "true", if either e1 or e2 (or both) are logical "true". The expression e2 is only evaluated if e1 is logical "false".
&&	e1 && e2	The result is "true" if e1 and e2 are logical "true". The expression e2 is only evaluated if e1 is logical "true".

12.7.5.8 Other

Operator	Example	Explanation
[ ]	a[e]	Array index: The result is the entry at position e of array a.
{ }	a{e}	Array index: The result is the index at position e of "packed" array a:
( )	f(e1, e2, ... eN)	The result is the value that is returned by function f. The function is called up with parameters e1, e2, ... eN. The sequence of the evaluation of the parameters is not defined. (The implementation of the script engine may evaluate the parameters in reverse order.)
? :	e1 ? e2 : e3	The result is either e2 or e3, depending on the value of e1. The conditional expression is a composite expression with a two-part operator, "?" and ":". The expression e2 is evaluated if e1 is logical "true"; e3 is evaluated if e1 is logical "false".
:	tagname: e	"Tag" overwritten: The value of the expression does not change, although the "tag" does change.
defined	defined s	Result is "1" if the symbol was defined. The symbol can be a constant or a global or local variable. The "tag" of the expression is "bool"
sizeof	sizeof s	The result is the number of elements of the specified variable. An element is an entry for simple variables and for one dimensional arrays. For multi-dimensional arrays, the result is the number of elements (sub-arrays) in the highest dimension. Add [] to the name of the array to specify a lower dimension. The result is 0 if the size of the variable is not known. If this operator is used in a "default" value of a function, the expression is executed at the time that the function was called up and not at the time that the definition was completed.
tagof	tagof s	The result is a unique number that represents the "tag" of the variables, the constants, the return value of a function or the name of the "tag" title. If this operator is used in a "default" value of a function, the expression is executed at the time that the function was called up and not at the time that the definition was completed.

12.7.5.9 Priority of the operators

The following table groups the operators with the same priority, starting with the highest priority at the top of the table.

If the evaluation of an expression is not explicitly justified with brackets, it is categorised by the association rules. For example: a*b/c is equal to (a*b)/c based on the left to right association, and a=b=c is equal to a=(b=c).

Operator	Explanation	Reading order
( ) [ ] { }	Function call array index (element) array index (character)	left-to-right
! ~ - ++ -- : defined sizeof tagof	logical not one's complement two's complement (unary minus) increase decrease "tag" overwritten symbol definition status symbol size in "elements" unique number of the tag	right-to-left
* / %	multiplication division modulo	left-to-right
+ -	addition subtraction	left-to-right
›› ››› ‹‹	arithmetic shift to the right logical shift to the right shift to the left	left-to-right
&	bitwise "and"	left-to-right
^	bitwise "exclusive or"	left-to-right
|	bitwise "or"	left-to-right
‹ ‹= › ›=	less than less or equal to greater than greater or equal to	left-to-right
== !=	equal unequal	left-to-right
&&	logical "and"	left-to-right
||	logical "or"	left-to-right
? :	conditional execution	right-to-left
=	Assignment *= /= %= += -= ››= ›››= ‹‹= &= ^= |=	right-to-left
,	comma	left-to-right

12.7.6 Statements

A statement can comprise one or several lines. A line can comprises two or more statements.

Statements for the sequence control (if, if-else, for, while, do-while and switch) can be nested.

12.7.6.1 Statement label

A label consists of an identifier followed by a ":". A label is a "Jump target" of a "goto" statement.

Each statement can be marked with a label. The label must be followed by a statement, which can also be an "empty statement".

The scope of a label is the function in which it was declared, i.e. a "goto" statement cannot jump from the current function to another function.

12.7.6.2 Composite statements

A composite statement (also known as a block) is a series of zero or several statements that is enclosed by brackets ("{" and "}"). The closing bracket ("}") must not be followed by a semicolon. Each statement can be replaced by a block. A composite statement that does not comprise any statements is a special case and is known as an "empty statement".

12.7.6.3 Expression statement

Each expression becomes a statement when a semicolon (";") is added. An expression also becomes a statement, if the expression is only followed by blank spaces to the end of the line, and the expression is not continued in the next line.

12.7.6.4 Empty statement

An empty statement does not execute any statements and consists of a block statement without statements, i.e. it consists of the "{ }" symbol. Empty statements are implemented in control flow statements without actions (e.g. "while (!iskey()) {}") or if a label is defined exactly before the closing bracket of a block statement. An empty statement does not end with a semicolon.

12.7.6.5 Assert expression

The program is aborted with a runtime error if the expression is logical "false"

> ***Note:*** *This expression protects against "impossible" or invalid conditions. In the following example, a negative fibonacci number is invalid. The assert statement marks this error as a programming error. Assert statements should only ever highlight programmer errors and never user inputs.*

Example:

```
fibonacci(n)
{
  assert n > 0

  new a = 0, b = 1
  for (new i = 2; i < n; i++)
  {
    new c = a + b
    a = b
    b = c
  }
  return a + b
}
```

12.7.6.6 Break

Terminates and leaves the smallest, encircling "do", "for" or "while" statement at any point in the loop. The "break" statement moves the program flow to the next statement outside the loop.

Example:

```
example(n)
{
  new a = 0

  for(new i = 0; i < n ; i++ )
  {
    a += i

    if(i>10)
      break

    a += 1
  }
  return a
}
```

12.7.6.7 Continue

Terminates the current iteration of the smallest encircling "do", "for" or "while" statement and moves the program control to the conditional part of the loop.

Example

```
example(n)
{
  new a = 0

  for(new i = 0; i < n ; i++ )
  {
    a += i

    if(i>10)
      continue

    a += 1
  }
  return a
}
```

12.7.6.8 Do statement while (expression)

Executes a statement before the conditional part (the "while" condition) is evaluated. The statement is repeated as long as the condition is logical "true". The statement is executed at least once.

Example:

```
example(n)
{
  new a = 0

  do
  {
    a++
  }
  while(n >= 0)

  return a
}
```

12.7.6.9 Exit expression

Cancels the program. The expression is optional, however, if present it must start and end on the same line as the "exit" statement. The exit statement returns the expression value or zero to the main application, if no expression is specified.

12.7.6.10 For (expression 1; expression 2; expression 3) statement

All three of the expressions are optional.

Expression 1:

Is only evaluated once before entering the loop. This expression can be used to initiate a variable. This expression also includes the variable declaration by means of the "new" syntax. A variable that is declared at this stage is only valid in the loop. It is not possible to combine an expression (with existing variables) and a declaration of new variables in this field. All of the variables must either already exist in this field, or they must all be declared in this area.

Expression 2:

This expression is executed before every run of the loop and terminates the loop if the expression logical "false" is returned. If this expression is omitted, it is assumed that the result of expression 2 is logical "true".

Expression 3:

This expression is executed each time the statement is completed. The program control moves from expression 3 to expression 2 for the next (conditional) iteration of the loop.

Example:

```
example(n)
{
  new a = 0

  for(new i = 0; i < n; i++)
  {
    a++
  }

  return a
}
```

The "for (; ;)" statement is the same as the "while (true)" statement.

12.7.6.11 Goto label

Moves the program control (unconditionally) to the statement that follows the specified label. The label must be within the same function as the "goto"-statement (a "goto"-statement cannot jump out of a function).

12.7.6.12 If (expression) statement 1 else statement 2

Executes statement 1 if the results of the expression is logical "true". The "else"clause of the "if" statement is optional. If the result of the expression is logical "false" and there is an "else" clause, the statement that is associated with the "else" clause (statement 2) is executed.

Example:

```
example(n)
{
  if(n < 0)
    return -1
  else if (n == 0)
    return 0
  else
    return 1
}
```

12.7.6.13 Return expression

Terminates the current function and moves the program control to the next statement following the function call. The expression value is returned as the function result. The expression can be an array or a character string. The expression is optional, however, if present it must start on the same line as the "return" statement. Zero is returned if no expression is specified.

12.7.6.14 switch (expression) {case list}

Transfers the sequence control to the various statements within the "switch", depending on the value of the "switch" expression. The main part of the "switch" statement is a composite statement that comprises a series of "case" clauses. Each "case" clause starts with the keyword "case" followed by a list of constants and a statement. The list of constants is a series of expressions separated by commas, each of which is evaluated as a constant value. This list ends with a colon. To specify an area in this list, separate the lower and upper limit of the area with a double point (".."). An example for an area is: "case 1..9:".

The "switch" statement shifts the sequence control to a "case" clause if a value from the list corresponds to the value of the "switch" expression.

The "default" clause consists of the "default" keyword and a double point. The "default" clause is optional, however, if it is specified it must be included as the last entry in the "case" list. The "switch" statement shifts the sequence control to the "default" clause if none of the "case" clauses comply with the "switch" expression.

Example:

```
example(n)
{
  new a = 0

  switch (n)
  {
    case 0..3:
      a = 0
    case 4,6,8,10:
      a = 1
    case 5,7:
      a = 2
    case 9:
      a = 3
    default:
      a = -1
  }

  return a
}
```

12.7.6.15 While (expression) statement

Evaluates the expression and executes the statement if the result of the expression is logical "true". The program control returns to the expression again once the statement has been executed. The statement is therefore executed as long as the expression is logical "true".

Example:

```
example(n)
{
  new a = 0

  while(n >= 0)
  {
    a++
  }

  return a
}
```

12.7.7 Functions

A function declaration specifies the name of the function and the formal parameters enclosed in brackets. A function can also return a value. A function must be defined globally, i.e. declared outside of another function and is globally available.

If the function declaration is followed by a semicolon (instead of a statement), this is a forward declaration of a function.

The "return" statement sets the return value of the function. For example, the return value of the "sum" function (see below) is the sum of both parameters. The "return" expression is optional.

```
sum(a, b)
{
  return a + b
}
```

The arguments of a function are (declared implicitly) local variables for this function. The function call specifies the values of the arguments. Another example of a complete definition of a function is "leap year" that indicates "true" or "false" for the relevant year.

```
leapyear(y)
{
  return y % 4 == 0 && y % 100 != 0 || y % 400 == 0
}
```

Details of the statements used in this example are provided in the chapter "Operators and expressions" on page 207.

Generally, functions include local variable declarations and consist of a block statement.

Note: *In the next example, the "assert" statement prevents negative values for the exponent.*

```
power(x, y)
{
  /* returns x^y */
  assert y >= 0

  new r = 1
  for (new i = 0; i < y; i++)
    r *= x

  return r
}
```

A function can comprise several "return" statements, for example, one is used to quickly terminate a function if invalid parameters are transferred, or when it becomes apparent that the function has nothing to do. If a function returns an array, all of the "return" statements must return an array with the same number of entries.

12.7.7.1 Function arguments ("call-by-value" versus "call-by-reference")

The "faulty" function in the next example has a parameter that is used in the loop to calculate the factorial of this number. It must be noted that the function modifies the argument.

```
main()
{
  new v = 5
  new f = faculty(v)
}

faculty(n)
{
  assert n >= 0

  new result = 1
  while (n > 0)
    result *= n--

  return result
}
```

Regardless of what (positive) value the "n" variable has at the start of the "while" loop, "n" will equal zero at the end of the function. In the "faculty" function, for example, the parameter is transferred as a value ("by value"), which means that changes to the "n" variable are only valid locally in the "faculty" function. In other words, the "v" variable in the "main()" function has the same value before and after the function is called up.

Arguments can be transferred as a value ("by value") or as a reference ("by reference"). A function argument that is to be transferred as a reference must have the "&" prefix preceding the name. The arguments are transferred to the function as a value by default.

Example:

```
swap(&a, &b)
{
  new temp = b
  b = a
  a = temp
}
```

To transfer an array to a function, add a pair of brackets ("[]") to the name of the argument. The number of entries can also be specified. This improves the error detection of the compiler's parser.

Example:

```
addvector(a[], const b[], size)
{
  for (new i = 0; i < size; i++)
    a[i] += b[i]
}
```

Arrays are always transferred as a reference.

> ***Note:*** *The "b" array in the above-mentioned example is not changed in the function. This function argument was declared as a "const" to make this explicit. In addition to the improved error detection, it also enables the compiler to generate a more efficient code.*

The following code example calls up the "addvector" function and adds five to each element of the "vect" variables:

```
new vect[3] = [ 1, 2, 3 ]

addvector(vect, [5, 5, 5], 3)

/* vect[] now comprises the values 6, 7 and 8 */
```

12.7.7.2 Named parameters versus fixed parameters

In the previous examples, the order of the parameters in a function call were important as each parameter was copied to the same position of the function parameter. For example, in the "weekday" function (defined below), the expression "weekday(12 ,31, 1999)" would be used to get the weekday of the last day of the last century.

```
weekday(month, day, year)
{
  /* returns the day of the week: 0=Saturday, 1=Sunday, etc. */
  if (month <= 2)
    month += 12, --year

  new j = year % 100
  new e = year / 100
  return (day + (month+1)*26/10 + j + j/4 + e/4 - 2*e) % 7
}
```

The date format changes depending on the culture and country, while the USA use the month/day/year format, European countries frequently use the day/month/year format and technical publications use the year/month/day (ISO/IEC 8824) format. In other words, the sequence of the parameters is not "standardised" or "normal". For this reason, there is an alternative way of transferring parameters to a function, by using "named parameters". These are illustrated in the next example (the function was declared in the same way as the previous example).

```
new wkday1 = weekday( .month = 12, .day = 31, .year = 1999)
new wkday2 = weekday( .day = 31, .month = 12, .year = 1999)
new wkday3 = weekday( .year = 1999, .month = 12, .day = 31)
```

In "named parameters", a dot (".") precedes the name of the argument. The argument of the function can be set to any expression that is valid for the argument. In the event of a named parameter, the equals sign ("=") does not refer to an allocation but instead links the expression with a function argument.

Fixed and named parameters can be mixed together, although the fixed parameters must be specified before the named parameters.

12.7.7.3 Standard values of function arguments

A function argument can have a standard value. The standard value of a function argument must be a constant. To specify a standard value, add an equals sign ("=") and the value to the name of the parameter.

The standard value is adopted if a placeholder is specified instead of a valid function parameter during a function call. The placeholder is the underscore character ("_"). The argument placeholder is only valid for parameters with a standard value.

The right argument placeholders can be removed from the list of arguments.

For example, if the "increment" function is defined as follows:

```
increment(&value, incr=1)
{
  value += incr
}
```

The following function calls are all the same:

```
increment(a)
increment(a, _)
increment(a, 1)
```

Standard values for arguments that are transferred as a reference are helpful in making these parameters optional. For example, if the "divmod" function was written to return the quotient and the rest as a parameter.

```
divmod(a, b, &quotient=0, &remainder=0)
{
  quotient = a / b
  remainder = a % b
}
```

Based on the previous definition of the "divmod" function, the following function calls are all valid:

```
new p, q

divmod(10, 3, p, q)
divmod(10, 3, p, _)
divmod(10, 3, _, q)
divmod(10, 3, p)
divmod 10, 3, p, q
```

The next example adds the value of an array to another one. The values of the array are increased by one if only one parameter is specified:

```
addvector(a[], const b[] = {1, 1, 1}, size = 3)
{
  for (new i = 0; i < size; i++)
    a[i] += b[i]
}
```

12.7.8 Example

12.7.8.1 Saw-tooth generator

This example generates a saw-tooth on the output of a device.

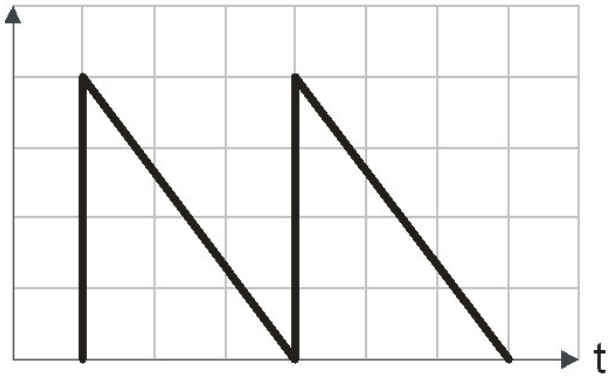
```
                                          // Static declaration of the fCurrentValue variable of the float type
static Float:fCurrentValue;               // This variable is not reset between two measurement cycles


/*
  This code is only executed once when
  - starting the device
  - changing a script.
  This can be used to initialise
  variables.
*/
main()
{
  fCurrentValue = 4.0;                    // Initialise fCurrentValue with the value 4.0
}


/*
  This code is executed at the end of every
  measurement cycle.
*/
public Mdn_CtrlFinish()
{
  if(fCurrentValue <= 0.0)                // If the variable is less or equal to 0.0,
     fCurrentValue = 4.0;                 // the variable is set to 4.0.

  fCurrentValue -= 0.1;                   // Subtract 0.1 from the variable.

  Mdn_SetCh(MDN_CH_IOUT1, fCurrentValue); // Set output 1 to the value of the fCurrentValue variable
}
```



Saw-tooth generated by means of pawn

12.7.8.2 Weekday designation

This example determines the weekday from the current time stamp of a device. An interim value is then calculated by dividing the time difference between the cycles of the script by six. With a measurement cycle of 2 minutes this equates to a value of 20. This interim value is then also divided by the determined weekday and the result is issued at output 0.

```
static iLastStamp;                                  /* Static declaration of the
                                                       iLastStamp variable of the integer type (is
                                                       automatically pre-initialised with 0)
                                                       This variable is not reset between two
                                                       measurement cycles.*/

/*
  This code is executed at the end of every
  measurement cycle.
*/
public Mdn_CtrlFinish()
{
  new iDayOfWeek;                                   /* Declaration of the iDayOfWeek variable of the
  new iCurrentStamp;                                   integer type */
  new Float:fResult;                                /* Declaration of the iCurrentStamp variable of the
                                                       integer type */
                                                    /* Declaration of the fResult variable of the
                                                       Float type
                                                       These variables are reset with every
                                                       measurement cycle */

  printf("iLastStamp=%d\r\n", iLastStamp);          /* Issue last time of a measurement via the standard
                                                       output */

  if(iLastStamp != 0)                               /* When the content of the iLastStamp variable
  {                                                    has already been set, the content of the
                                                       brackets is executed. */

    iDayOfWeek = Mdn_DoW(Mdn_GetDate());            /* Save week day in the iDayOfWeek
                                                       variable */
    iDayOfWeek += 1;                                /* +1 so that the subsequent division is not divided by 0
                                                       (Monday) */
    printf("iDayOfWeek=%d\r\n", iDayOfWeek);        /* Issue the weekday via the standard
                                                       output. */

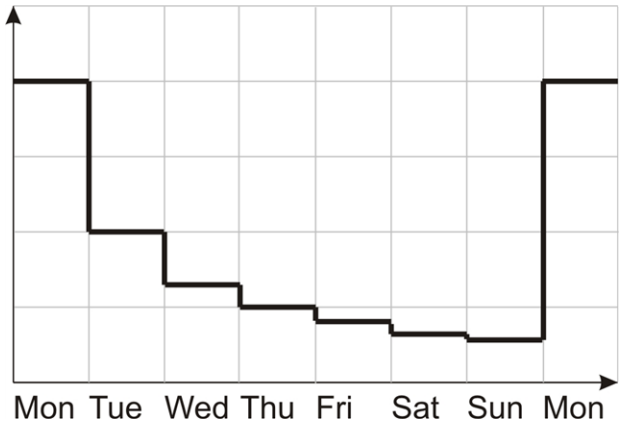
    iCurrentStamp = Mdn_GetTime();                  /* Read out the current time and save in the
                                                       iCurrentStamp variable. */
    printf("iCurrentStamp=%d\r\n", iCurrentStamp); /* Issue the current time via the standard
                                                       output. */

    fResult = (iCurrentStamp - iLastStamp) / 6.0;   /* Determine the time difference between the current and la
                                                       time of a measurement (measurement cycle) and
                                                       divide by 6. */
    printf("fResult=%f\r\n", fResult);              /* Issue the time difference divided by 6 via the
                                                       standard output. */

    fResult = fResult / iDayOfWeek;                 // Divide the result by the weekday.
    printf("fResult=%f\r\n", fResult);              // Issue the result via the standard output.

    Mdn_SetCh(MDN_CH_IOUT1, fResult);               // Write the result on the output.
  }

  iLastStamp = Mdn_GetTime();                       /* Temporarily save the current time for processing
}                                                      until the next time of a measurement. */
```



Output signal generated by means of pawn

12.7.8.3 Calculating the flow rate with the table

This example determines the flow rate of a wastewater system based on the fill level of the wastewater system. The fill level is measured at input 1 and the flow rate should be issued at output 1.

```
static const Table[][Mdn_TablePoint] =                /* Static table for converting the height
[                                                        to the flow rate */
  [ 0.0,  0.0],
  [ 2.0,  1.0],
  [ 5.0,  3.0],
  [10.0, 10.0],
  [50.0, 15.0]
];


/*
  This code is executed at the end of every
  measurement cycle.
*/
public Mdn_CtrlFinish()
{
  new Float:fHeight;                                   // Declaration of the fHeight variable of the float type
  new Float:fFlow;                                     // Declaration of the fFlow variable of the float type
  new Mdn_ValueStatus:iStatus;                         /* Declaration of the iStatus variable of the
                                                          Mdn_ValueStatus type
                                                          These variables are reset with every
                                                          measurement cycle */

  Mdn_GetCh(MDN_CH_UI1, fHeight, iStatus);             /* Read out channel 1 and save the value in
                                                          the fHeight variable and the status in
                                                          the iStatus variable. */

  if(iStatus == MDN_STATUS_OK)                         /* If the status of channel 1 is OK, execute the
  {                                                       following code in the {} brackets */
    new iResult;                                       // Declaration of the iResult variable of the integer type

    iResult = Mdn_CalcTable( fHeight, fFlow,           /* This function calculates the flow rate
                             Table, sizeof Table);        (fFlow) based on the height (fHeight)
                                                          and the table (table).
                                                          The result is linearly interpolated,
                                                          if the height is between 2 entries.
                                                          The return value contains the error value of
                                                          the calculation */

    switch(iResult)
    {
      case MDN_TAB_ERR_FLOOR:                          /* If the height is lower than the first entry in the
      {                                                   table, */
        fFlow = 0.0;                                   // set the flow rate to 0.0.
      }
      case MDN_TAB_ERR_CEIL:                           /* If the height is greater than the last entry in the
      {                                                   table, */
        fFlow = 0.0;                                   // set the flow rate to 0.0.
      }
    }
  }

  Mdn_SetCh(MDN_CH_IOUT1, fFlow);                      //Write the flow rate on output 1.
}
```

12.7.8.4 Script parsing

The following example initially issues the device class, modem version and firmware version on the standard output. "*INIT COM1\r\n*" is then issued as the init sequence on Com1. "*POLL COM1\r\n*" is issued in the measurement cycle on Com1. The received answer is issued on the standard output for control purposes. "*OUTPUT COM1\r\n*" is issued at Com1 as soon as the control module and thus the "Mdn_CtrlFinish()" function (see Mdn_CtrlFinish()) is called up during the internal processing of the measurement values.

```
public Mdn_CtrlFinish()
{
  Mdn_SerialTx(1, "OUTPUT COM1\r\n", 13);              // Issue "OUTPUT COM1" string via Com1
}

COM1Event(event)                                       // Sub-function to evaluate the events of Com1
{
  if(event == MDN_SERIAL_EVENT_INIT)                   // If the init sequence should be sent
  {
    Mdn_SerialTx(1, "INIT COM1\r\n", 11);              // Send "INIT COM1" string via Com1
  }
  else if(event == MDN_SERIAL_EVENT_MEASURE)           // If the measurement command should be sent
  {
    Mdn_SerialTx(1, "POLL COM1\r\n", 11);              // Send "POLL COM1" string via Com1
  }
  else if(event == MDN_SERIAL_EVENT_TIMEOUT)           // If a timeout has occurred
  {
    Mdn_SerialFinish(1);                               // Terminate data reception via Com1
  }
}

public Mdn_SerialEvent(com, event)                     // Access point for evaluating the serial events
{
  printf("Mdn_SerialEvent( %d, %d)\r\n",               /* Issue number of the COM interface and received
                           com, event);                   event via the standard output */

  switch(com)                                          // Check which COM interface has triggered the event
  {
    case 1: COM1Event(event);                          // Call up the sub-function for evaluating the Com1 events
  }
}

COM1Rx(const data[], len)                              // Sub-function for evaluating the received data
{
  /*-------------------------------------*/
  /*--- Code that evaluates the data --- */
  /*-------------------------------------*/

  Mdn_SerialFinish(1);                                 // Terminate receipt of data via Com1
}

public Mdn_SerialRx(com, const data{}, len)            // Access point for evaluating the received characters
{
  printf("Mdn_SerialRx( %d, \"%s\", %d)\r\n",          /* Issue number of the COM interface, received data and
                         com, data, len);                number of characters via the standard output */

  switch(com)                                          /* Check via which COM interface the data was
  {                                                       received */
    case 1: COM1Rx(data, len);                         /* Call up the sub-function to evaluate the data received
  }                                                       via Com1 */
}


main()
{
  printf( "%04X DEVICE_CLASS\r\n", DEVICE_CLASS);      // Issue device class via the standard output
  printf( "%04X MODEM_VERSION\r\n", MODEM_VERSION);    // Issue modem version via the standard output
  printf( "%04X CONTROLLER_VERSION\r\n",               // Issue firmware version via the standard output
          CONTROLLER_VERSION);
}
```

12.7.9 Differences to C

- The pawn is missing the input mechanism of C and is an "integer-only" variant of C. There are no structures or unions. Floating point support must be implemented with user-defined operators and the help of native functions.
- The syntax for floating point values is stricter than that in C. Values such as ".5" and "6." are acceptable in C but must be written as "0.5" and "6.0" in the pawn. The decimal point is optional in C. If an exponent is included, then you can write "2E8" in C, however the pawn will not accept the capital letter "E". Use the lower case letter "e". It requires a comma: e.g. "2.0e8" (see "Numerical constants" on page 203).
- The pawn does not support any "pointer". The pawn includes a "reference" argument to transfer function parameters as a reference (see "Function arguments ("call-by-value" versus "call-by-reference")" on page 219). The "placeholder" argument replaces some applications of the ZERO pointer (see "Standard values of function arguments" on page 222).
- Numbers can be specified in a hexadecimal, decimal or binary format. The octal format is not supported (see "Numerical constants" on page 203). Hexadecimal numbers must start with "0x" ("x" in lower case). The prefix "0X" is invalid.
- "Cases" in a "switch"-statement are not "fall through". At least one statement must follow the "case" label. You must create a composite statement (with {}) to execute several statements (see "switch (expression) {case list}" on page 217). The "switch" statement is a "conditional goto" in C/C++. The "switch" statement is a structured "if" in the pawn.
- A "break" statement only terminates loops. In C/C++, the "break" statement also terminates a "case" in a "switch" statement.
- The pawn supports "array assignments" with the limitation that both of the arrays must be the same length. For example, if "a" and "b" arrays have six lines, the expression "a=b" is valid. In addition to character strings, the pawn also supports literal arrays and thus expressions such as "a = {0,1,2,3,4,5}" where "a" is an array variable with six elements.
- "defined" is an operator and not a preprocessor directive. The "defined" operator in the pawn works with constants (declared with "const"), global variables, local variables and functions.
- The "sizeof" operator returns the size of the variables in "elements" and not in "bytes". An element is an entry or sub-array. Further details are provided in the chapter "Other" on page 211.
- An empty statement is an empty block (with {}) and not a semicolon (see "Composite statements" on page 213). This change prevents frequent errors.

- A division is completed in such a way that the remainder of the division has (or ought to have) the same prefix as the denominator. Divisions (operator "/") are always rounded down to the smaller whole number (whereby -2 is smaller than -1). For example, 5/2 = 2 (2.5 is rounded down to 2), -5/2 = -3 (-2.5 is rounded down to -3). The "%" operator always generates a positive result regardless of the prefix of the numerator (see "Operators and expressions" on page 207).
- There is no unary "+" operator as it is a "no-operation" operator anyway ("a = +1" is not valid; correct: "a = 1").
- Three bit by bit operators have different priorities than in C. The priority level of the "&", "^" and "|" operator is higher than the relational operators. Dennis Ritchie explains that these operators were assigned a low priority level in C as early C compilers did not yet include the logical "&&" and "||" operators so that bit by bit "&" and "|" were used instead.
- The keyword "const" in the pawn implements the "enum" functionality of C.
- In most cases, the forward declarations of functions (i.e. prototypes) are not necessary. A pawn is a two-pass compiler. It detects all of the functions during the first cycle and uses them during the second. User-defined operators must however be declared before use. If available, forward declarations must be exactly the same as the definition of the function. The parameter names in the prototypes and the definitions of the functions must be identical. The pawn attends to the parameter name in the prototype due to the "named parameter" function. The pawn uses prototypes to call up the forward declared functions. To use these with the named parameters during this process, the compiler must already know the names of the parameters (and their position in the parameter list). The parameter names in the prototypes must therefore match those in the definitions.

6872
6872
6872

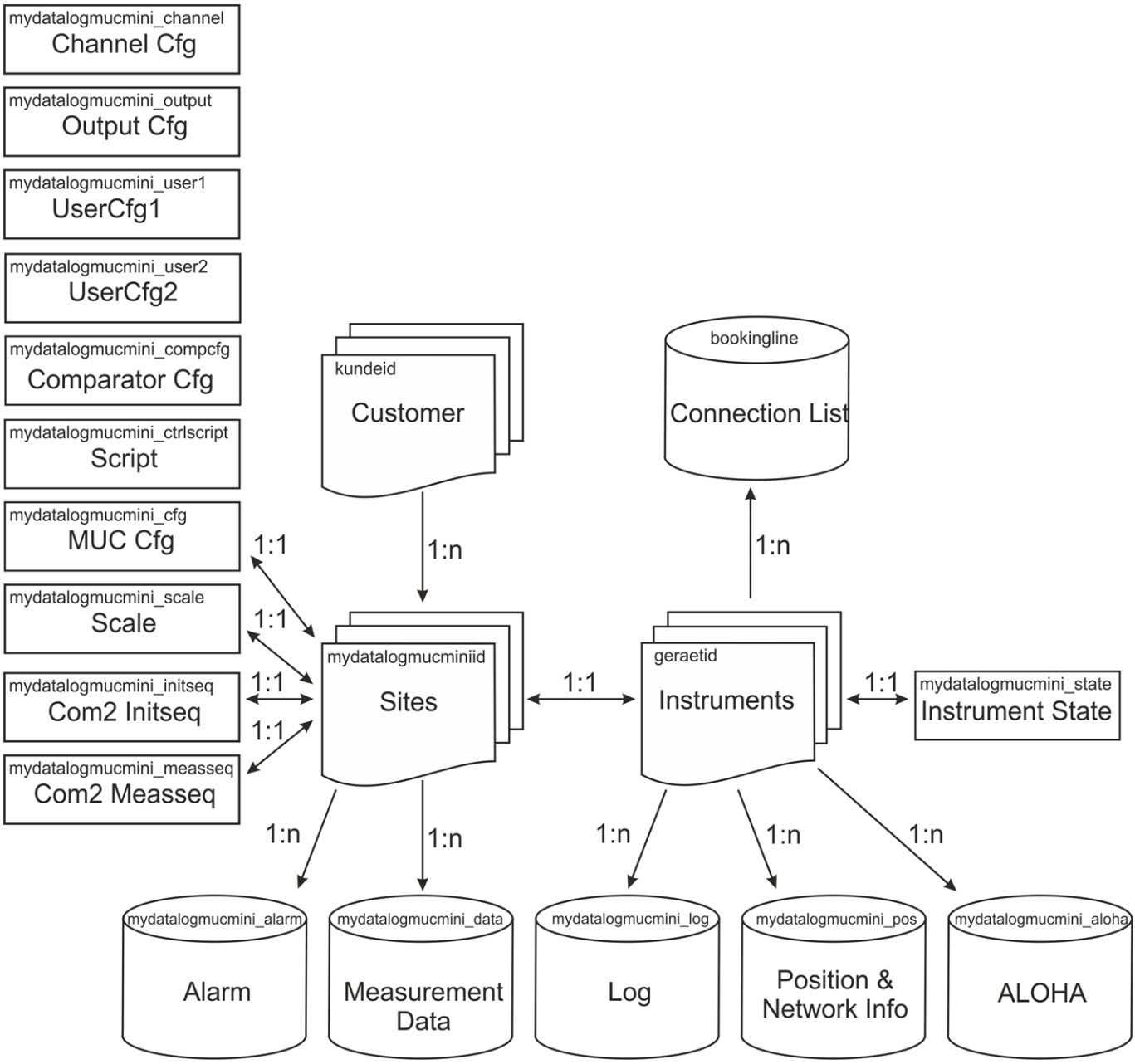
Chapter 13 XML interface

Important note: The relevant licences are required on the myDatanet server to use the XML interface. If the licence is not available for the XML export, the following message is displayed for every request: "`-ERR: XML agent is not active on this server!`". Likewise, if the licence is missing for the XML import, the following message is returned for each import attempt: "`-ERR: XML upload is not active on this server!`". In this case, contact your responsible sales partner.

13.1 General

The XML interface is provided to export data from and import data to the myDatanet server. However, this is not just limited to the pure measurement data but includes all of the data provided by the myDatanet server (e.g. measurement data, configurations, log entries, alarm lists, etc.). It is therefore possible for the customer to completely dispense with the interface of the myDatanet server and to create his own user interface. A specially developed PC program or web interface can, for example, be used for this purpose.

13.2 Structure of the database



Structure of the database

Generally applicable table identifiers

Identifier	Description
mydatalogmucminiid	Site configuration (see "mydatalogmucminiid" on page 236)
geraetid	Devices (see "geraetid" on page 245)
bookingline	Connection list (see "bookingline (read only)" on page 246)
kundeid	Customers (see "kundeid" on page 247)

Device-specific table identifiers

Identifier	Allocation	Description
mydatalogmucmini_data	Site	Measurement data (see "mydatalogmucmini_data" on page 249)
mydatalogmucmini_channel	Site	Configuration of the universal inputs (see "mydatalogmucmini_channel" on page 250)
mydatalogmucmini_output	Site	Configuration of the outputs (see "mydatalogmucmini_output" on page 255)
mydatalogmucmini_log	Device	Device log (siehe "mydatalogmucmini_log" on page 258)
mydatalogmucmini_user1	Site	User config 1 (see "mydatalogmucmini_user1" on page 258)
mydatalogmucmini_user2	Site	User config 2 (see "mydatalogmucmini_user2" on page 260)
mydatalogmucmini_state	Device	Device status (see "mydatalogmucmini_state" on page 260)
mydatalogmucmini_alarm	Site	Alarm list (see "mydatalogmucmini_alarm" on page 261)
mydatalogmucmini_compcfg	Site	Trigger and alarm configuration (see "mydatalogmucmini_compcfg" on page 262

Identifier	Allocation	Description
mydatalogmucmini_aloha	Device	Aloha data (see "mydatalogmucmini_aloha" on page 266)
mydatalogmucmini_pos	Device	Position and network information (see "mydatalogmucmini_pos" on page 267)
mydatalogmucmini_ctrlscript	Site	Configuration of the script (see "mydatalogmucmini_ctrlscript" on page 268)
mydatalogmucmini_cfg	Site	Configuration of the COM interfaces, interface channels and interface output channels (see "mydatalogmucmini_cfg" on page 268)
mydatalogmucmini_scale	Site	Configuration for rescaling from the register value to the measurement value for the interface channels and interface output channels (see "mydatalogmucmini_scale" on page 275)
mydatalogmucmini_initseq	Site	Init sequence for Com2 (see "mydatalogmucmini_initseq" on page 277)
mydatalogmucmini_ measseq	Site	Measurement data sequence for Com2 (see "mydatalogmucmini_measseq" on page 277)

The device-specific table identifiers are distinguished as ones that are assigned to the device (geraetid) and ones that are assigned to the site (mydatalogmucminiid). If the assignment of the device and site is deallocated, the tables that are assigned to the device (e.g. the device log) remain firmly linked to the device and the tables that are assigned to the site (e.g. measurement data) remain firmly linked to the site. This means that if a new device is assigned to a site, the measurement data that was generated while the previously assigned device was connected to the site can be read out (if the "geraetid" table of the new device is selected as the starting point).

The structure of the database must be known to create a xPath query (see "Syntax of the xPath query" on page 278). If, for example, the measurement data for a certain serial number should be read out, the path initially goes via the "geraetid" table that contains the serial number, to the "mydatalogmucminiid" table that contains the link between the device and site and to the "mydatalogmucmini_data" table that contains the required measurement data. An example is provided in t.b.d.

13.2.1 Generally applicable table identifiers

13.2.1.1 mydatalogmucminiid

This table includes all of the site-specific configurations..

name

Site designation (not relevant for the device or data assignment)[2-50 characters]

timezone

Regional settings (not relevant for raw measurement data as this is stored in UTC)

Value	*Explanation*
99.00	*The regional setting for the site is adopted from the global server setting.*
-12.00	*UTC -12 hours*
:	*:*
+14.00	*UTC +14 hours (Kiritimati)*

summertime

Configuration for automatic time adjustment

Value	*Explanation*
DEFAULT	*The configuration for the time adjustment is adopted by the global server setting*
OFF	*Automatic time adjustment deactivated*
USA;99CC96152B000000	*Predefined setting for the American area*
EU;99CC95A627080884	*Predefined setting for the European area*

alert_mode

Setting for acknowledging the alarms

Value	*Explanation*
0	*The global server setting is used to determine whether alarms must be acknowledged automatically or manually.*
1	*Alarms are acknowledged automatically as soon as all of the messages have been sent. If SMS that have a tariff with a delivery confirmation function have also been sent, acknowledgement is provided after delivery confirmation.*
2	*Alarms must be acknowledged by the user.*

transfer_alerts

Alarm in case the device does not respond for longer than the set number of transmission cycles

Value	*Explanation*
0	*No alarm settings*
1	*standard (>2x transmission cycles)*
2	*immediately (>1x transmission cycle)*
3	*medium (>5x transmission cycles)*
4	*long (>10x transmission cycles)*
5	*extra long (>20x transmission cycles)*

max_transfer_volume

Settings for the transfer volume alarm

Value	*Explanation*
-1	*The setting for the transfer volume alarm is taken from the global server settings.*
0	*The transfer volume alarm is deactivated.*
> 0	*Threshold at which the transfer volume alarm should be triggered [KB]*

max_transfer_volume_kb

This is a "Ghost" field, the content of which always corresponds to that of the "max_transfer_volume" field.

comment

Free comment field (is also displayed below the device type in the site list)

loc_site_lat

Specification of the geographical latitude where the site is located [°]

loc_site_long

Specification of the geographical longitude where the site is located [°]

loc_alarm_rad_mode

A radius can be defined for a site. Depending on the settings for this parameter, an alarm is triggered if the device leaves this radius.

Value	*Explanation*
0	*Radius monitoring deactivated (no alarm)*
> 0	*Specification of the alarm radius [km]. An alarm is triggered if the device leaves the radius.*

loc_alarm_rad

This is a "Ghost" field, the content of which always corresponds to that of the "loc_alarm_rad_mode" field.

view

Selection of the report that is loaded by clicking on the device link in the maps

Value	*Explanation*
*"*OFF*"*	*The default graphic is loaded.*
"viewid"	*The selected report is loaded.To obtain the "viewid", open the desired report (see ""Site" area at customer level" on page 149) and take the ID from the address line of the browser. (e.g.. . . . &viewid=**A1A5AD6CD299EF96**&. . . .).*

ftpexportprofile

Selection of the FTP export profile

Value	*Explanation*
"0"	*FTP export deactivated*
"id"	*To obtain the "id", open the configuration of the desired FTP export profile (see myDatanet Server Manual 805002) and take the ID from the address line of the browser. (e.g.. . . . &id=**A0B66D1CF7ED87DD**&. . . .).*

ftpexportdir

makes overwriting the standard directory of the selected FTP export profile possible [0-100 characters]

mapview_showvaluesinmap

Specifies whether the current measurement values for the site should be displayed directly in the OpenStreetMaps map

binary_src

Specifies the source of the script

Value	*Explanation*
0	*The script is directly entered and compiled via the server interface.*
1	*A script binary file (*.amx), that has already been compiled, should be uploaded.*

view_template

Selection of the report that is loaded by clicking on the device link in the maps

Value	*Explanation*
Empty string	*The default graphic is loaded.*
"viewid"	*The selected report is loaded.To obtain the "viewid", open the desired report template (see "myDatanet Server Manual " 805002) and take the ID from the address line of the browser. (e.g.. . . . &viewid=**A1A5AD6CD299EF96**&. . . .).*

mappincaption

Specifies whether the title of the site should be displayed in the "Map"

mapovpincaption

Specifies whether the title of the site should be displayed in the "Overview map"

The following parameters apply to the universal inputs (see "Measurement channels" on page 96)

title1 - title3

Freely selectable channel title for the universal inputs

units1 - units3

String that is used as a measurement unit by all of the server display elements [0-16 characters]

Important note: *The content of this field must be deleted in Digital channel mode.*

min1 - min3

Channel mode	*Explanation*
Measurement channel deactivated	*---*
Digital	
Cnt.Day	
Cnt.Intervl.	
Freq	*Defines the lower scale end of the pointer instruments*
PWM	*Start of the measurement range in the measurement unit* ***Important note:*** *For these channel modes it is also necessary that the same value as the one here is entered in the "min" field of the relevant measurement channel within the "mydatalogmuc_ channel" table.*
4-20 mA	
0-20 mA	
0-2V	
0-10V	

min_min1 - min_min3

This is a "Ghost" field, the content of which corresponds to that of the "min1" - "min8" field.

max1 - max3

Channel mode	*Explanation*
Measurement channel deactivated	*---*
Digital	
Cnt.Day	*Defines the upper scale end of the pointer instruments*
Cnt.Intervl.	
Freq	
PWM	*End of the measurement range in the measurement unit* ***Important note:*** *For these channel modes it is also necessary that the same value as the one here is entered in the "max" field of the relevant measurement channel within the "mydatalogmuc_ channel" table.*
4-20 mA	
0-20 mA	
0-2V	
0-10V	

max_max1 - max_max3

This is a "Ghost" field, the content of which corresponds to that of the "max1" - "max8" field.

decpl1 - decpl3

Number of decimal places that are used by all of the server display elements

Important note: *The content of the field must be set to 0 for Digital channel mode (no decimal places)*

Value	*Explanation*
-1	*Standard*
0	*No decimal places*
:	*:*
6	*6 decimal places*

The following parameters apply to the "Internal channels" (see "Internal channels" on page 139).

title4

Freely selectable channel title for the GSM field level[0-16 characters]

units4

GSM level: String that is used as a measurement unit by all of the server display elements [0-16 characters]

title5

Freely selectable channel title for the supply voltage [0-16 characters]

units5

Voltage: String that is used as a measurement unit by all of the server display elements [0-16 characters]

The following parameters apply to the "Interface channels" (see "Interface channels 1-32" on page 115).

title8 - title71

Freely selectable channel title for the interface channels [0-16 characters]

units8 - units71

String that is used as a measurement unit by all of the server display elements [0-16 characters]

> ***Important note:*** *The content of this field must be deleted in digital channel mode.*

decpl8 - decpl71

Number of decimal places that are used by all of the server display elements

> ***Important note:*** *The content of this field must be set to 0 for digital channel mode (no decimal places).*

Value	*Explanation*
-1	*Standard*
0	*No decimal places*
:	*:*
6	*6 decimal places*

chmode8 - chmode71

Specifies how the interface channel should be handled by the evaluation elements of the server

Value	*Explanation*
2	*Day counter. This means that the system anticipates that the measurement value of the channel will continuously increase and is reset once per day.*
3	*Interval counter. This means that the system assumes that the counter reading is reset each time a measurement is recorded.*
6	*Analogue or digital measurement value. This means that the system deals with each measurement value independently of the measurement values before or afterwards.*
12	*Infinite counter. This means that the system expects that the measurement value of the channel will continuously increase and never be reset.*

The following parameters apply to the "Interface output channels" (see "Interface output channels 1-32" on page 122).

title72 - title135
Freely selectable channel title for the interface output channels [0-16 characters]

units72 - units135
String that is used as a measurement unit by all of the server display elements [0-16 characters]

> ***Important note:*** *The content of this field must be deleted in digital channel mode.*

decpl72 - decpl135
Number of decimal places that are used by all of the server display elements

> ***Important note:*** *The content of this field must be set to 0 for digital channel mode (no decimal places).*

Value	*Explanation*
-1	*Standard*
0	*No decimal places*
:	*:*
6	*6 decimal places*

chmode72 - chmode135
Specifies how the interface output channel should be handled by the evaluation elements of the server

Value	*Explanation*
2	*Day counter. This means that the system anticipates that the measurement value of the channel will continuously increase and is reset once per day.*
3	*Interval counter. This means that the system assumes that the counter reading is reset each time a measurement is recorded.*
6	*Analogue or digital measurement value. This means that the system deals with each measurement value independently of the measurement values before or afterwards.*
12	*Infinite counter. This means that the system expects that the measurement value of the channel will continuously increase and never be reset.*

The following parameters apply to the "Output channels" (see "Output channels" on page 133).

title800
Freely selectable channel title for the analogue output (not galvanically isolated) [0-16 characters]

units800
String that is used as a measurement unit by all of the server display elements [0-16 characters]

decpl800

Number of decimal places that are used by all of the server display elements

Value	*Explanation*
-1	*Standard*
0	*No decimal places*
:	*:*
6	*6 decimal places*

title801

Freely selectable channel title for the isolated switch contact (parallel connection of an optoswitch and relay) [0-16 characters]

units801

String that is used as a measurement unit by all of the server display elements [0-16 characters]

> ***Important note:*** *The content of this field must be deleted in "Digital" channel mode and set to "sec" in "Ext warmup time" channel mode.*

The following parameters apply to the "Calculated channels" (see "Calculated channels" on page 129).

title600 - title604

Freely selectable channel title for the calculated channels [0-16 characters]

units600 - units604

String that is used as a measurement unit by all of the server display elements [0-16 characters]

min600 - min604

Defines the lower scale end of the pointer instruments

max600 - max604

Defines the upper scale end of the pointer instruments

decpl600 - decpl604

Number of decimal places that are used by all of the server display elements

Value	*Explanation*
-1	*Standard*
0	*No decimal places*
:	*:*
6	*6 decimal places*

alarm_low600 - alarm_low604

An alarm is triggered, if the measurement value drops to or below this value.

alarm_high600 - alarm_high604

An alarm is triggered, if the measurement value meets or exceeds this value.

hysterese600 - hysterese604
Hysteresis for the all-clear in the event of an alarm/warning (e.g. Hyst=5%, alarm or warning at 100 -› all-clear at 95) or reset the trigger (e.g. Hyst=5%, level = greater or equal, trigger at 100 -› reset at 95)

13.2.1.2 geraetid

This table includes general information and configurations that are the same for all devices of the myDatanet range.

kunde
Unique data record number of the customer to whom the device is assigned (read only)

conn_path
Specifies the communication channel via which the last connection was established

Value	*Explanation*
0	*GSM/GPRS*
1	*Other communication channel (e.g. LAN)*

devsn
Serial number of the instrument

devclass
Device class The instrument class of the site and instrument must match for an instrument to be able to be connected to a site.

telno
Telephone number of the SIM card. The control SMS messages (e.g. wakeup) are sent to this number. Format: +43555837465[0-20 characters]

devflags
Additional information regarding the instrument class (for internal use) (read only)

swversion
Current software version installed on the measurement controller (read only)

modemversion
Current software version installed on the modem controller (read only)

osversion
OS version of the modem(read only)

lastcon
Time stamp of the last connection (read only)

lastwakeup
Time stamp of the last wakeup (read only)

lastdisc
Time stamp of the last disconnection (read only)

lasterr
Time stamp of the last transmission error (read only)

lastaloha

Time stamp of the last Aloha connection (read only)

fw_update

Settings for the firmware update

Value	*Explanation*
0	*Off: Firmware update is deactivated.*
1	*On: As soon as a new version of the selected firmware type is available, this is installed immediately.*
2	*Even if tag is missing: Firmware is also transferred to the device if the device has not transmitted the current firmware version to the server (NOT RECOMMENDED!).*
3	*Allow downgrade: Facilitates the installation of an older firmware version than the one on the device (NOT RECOMMENDED!)*
4	*Once: Performs a single firmware update. If no new firmware is available or the firmware was installed successfully, the firmware update is automatically switched to "OFF".*
5	*Ignore: The firmware update is deactivated and no information is provided about available firmware updates.*

fw_release_type

Settings for the firmware type

Value	*Explanation*
0	*Released: Only firmware versions that have successfully undergone internal and field testing are installed (this practically eliminates malfunctions).*
1	*Release candidate: Only firmware versions that have successfully undergone internal testing are installed (malfunctions cannot be excluded).*
2	*Beta release: Even firmware versions that have not successfully undergone all of the internal tests are installed (malfunctions may occur).*

dev_hw

Hardware version of the myDatalogMUCmini

13.2.1.3 bookingline (read only)

This table includes information about the connections between the devices and the myDatanet server

kunde

Unique data record number of the customer to whom the device is assigned

geraet

Unique data record number of the device to which the connection list is assigned

text
Duration of the connection

context
Title of the site with which the device is linked or error that occurred during the connection

source
Serial number and IP address

cmh_quantity
Amount of received data in bytes (device to the server)

cmh_code
Indicates whether the device has established a connection to the myDatanet server (41=closed, 942=connected)

> ***Important note:*** *Table entries with different codes to the ones mentioned above include information that is only intended for internal use and that should be filtered out by the customer.*

13.2.1.4 kundeid

This table contains information and configurations for the customers created on the myDatanet server.

name
The freely selectable name of customer is a mandatory field [2-50 characters].

Address of the customer:

strasse
Street [0-50 characters]

plzort
Postcode and town [0-50 characters]

comment
Comment on the customer. This is displayed in the customer overview below the name.

erp_code
The ERP code is designed to simplify and speed up the charging process. All costs that are, for example, incurred through SMS or calls are charged directly to this ERP code.

Value	*Explanation*
*	*The ERP code is adopted from the global server settings.*
ERP code	*One of the ERP codes that was created via the global server settings (see myDatanet Server Manual 805002)* ***Important note:*** *If an invalid code is specified, the ERP code is set to the global server settings the next time you edit the customer via the web interface of the myDatanet server.*

sms_enabled

This includes the option to generally deactivate SMS for the customer.

Value	*Explanation*
-1	*The setting is adopted from the global server settings*
0	*Off*
1	*On*

call_enabled

This includes the option to generally deactivate calls for the customer.

Value	*Explanation*
-1	*The setting is adopted from the global server settings*
0	*Off*
1	*On*

acknowledge_mode

Customer's settings to acknowledge the alarms

> ***Important note:*** *This setting can be overwritten for every measurement instrument via the alarm settings (see "alert_mode" field in chapter "mydatalogmucminiid" on page 236).*

Value	*Explanation*
0	*The global server setting is used to determine whether alarms must be acknowledged automatically or manually.*
1	*Alarms are acknowledged automatically as soon as all of the messages have been sent. If SMS that have a tariff with a delivery confirmation function have also been sent, acknowledgement is provided after delivery confirmation.*
2	*Alarms must be acknowledged by the user.*

alert_schedule_mode

Settings for the procedure to inform about alarms that have to be acknowledged by the user

Value	*Explanation*
0	*The setting is adopted from the global server settings*
1	*Inform about every alarm per site: In this case, the alarm schedule is activated every time an alarm is triggered. Even if the alarm schedule has already been executed and the alarm has not yet been acknowledged, the alarm schedule is executed again if a new alarm is triggered on the site.* ***Important note:** This can incur high costs.*
2	*Inform about the first unacknowledged alarm per site: Messages for new alarms that have been triggered will only be sent once the alarm has been acknowledged.*

13.2.2 Device-specific table identifier

13.2.2.1 mydatalogmucmini_data

This table contains the measurement data of the myDatalogMUCmini (see "Internal channels" on page 139, "Measurement channels" on page 96, "Output channels" on page 133, "Interface channels 1-32" on page 115, "Interface output channels 1-32" on page 122 and "Calculated channels" on page 129).

context

Unique data record number of the site (read only)

gsm_level

GSM level [dBm]

voltage

Voltage [V]

ch1 - ch3

Measurement values of the universal inputs

output1

Setpoint of the analogue output

output2

Setpoint of the isolated switch contact

chm1 - chm64

Measurement values of the interface channels

chmo1 - chmo64

Setpoints of the interface output channels

calc600 - calc604

Measurement values of the "Calculated channels".

> ***Important note:*** *However, these measurement values are not saved in the database but are recalculated during each access (display or download) based on the specifications entered via the "Calculated channels" configuration section (see "Calculated channels" on page 129).*

13.2.2.2 mydatalogmucmini_channel

This table contains the configuration of the universal inputs (see "Measurement channels" on page 96) that is transferred to the myDatalogMUCmini .

context

Unique data record number of the site (read only)

mode1 - mode3

Basic settings for the measurement channel

Value	*Explanation*
0	*Measurement channel deactivated*
1	*Digital*
2	*Cnt.Day*
3	*Cnt.Intervl.*
4	*Freq*
5	*PWM*
6	*4-20 mA*
7	*0-20 mA*
8	*0-2V*
9	*0-10V*

> ***Important note:*** *If the measurement channel mode is set to Digital, the following additional settings are required:*
>
> - *In the "mydatalogmucminiid" table, the content of the "units" field (measurement unit) must be deleted for the relevant measurement channel.*
> - *In the "mydatalogmucminiid" table, the "decpl" field must be set to 0 (no decimal places) for the relevant measurement channel.*
> - *In the "mydatalogmucmini_compcfg" table, the content of the "alarm_low_lower", "alarm_low_upper", "alarm_high_lower" and "alarm_high_upper" fields must be deleted for the relevant measurement channel.*
> - *In the "mydatalogmucmini_compcfg" table, the "trig_undershoot" field must be set to 0 and the "trigger_value" field must be set to 1 for the relevant measurement channel.*

min1 - min3
Start of the measurement range in the measurement unit

max1 - max3
End of the measurement range in the measurement unit

scale1 - scale3
Metered measurand of a pulse in the measurement unit (only for channels in "Cnt.Day" or "Cnt.Intervl." mode)

scale_digital1 - scale_digital
Inverts the input signal (only for channels in "Digital" mode)

scale_fq1 - scale_fq3
Factor by which the input signal is multiplied (only for channels in "Freq" mode)

filter_time1 - filter_time3
For channels in "Digital", "Cnt.Day", "Cnt.Intervl., "Freq or PWM mode:

Time during which the signal must remain constant to initiate a level change. Used to suppress brief faults (debouncing).

For channels in "4-20 mA", "0-20 mA", "0-2V" or "0-10V" mode:

Time during which the analogue signal is averaged for signal smoothing. Used to suppress signal noise (also see "Output channels" on page 133).

filter_time1d - filter_time3d
This is a "Ghost" field, the content of which corresponds to that of the "filter_time1" - "filter_time8" field.

reset_time1 - reset_time3
Reset time of the day counter (only for channels in "Cnt.Day" mode)

Value	*Explanation*
0	*00:00*
60	*01:00*
120	*02:00*
: *:*	*:* *:*
1320	*22:00*
1380	*23:00*

param2_1 - param2_3

Hold the last valid measurement value for x measurement cycles (not for channels in "Digital", "Cnt.Day" and "Cnt.Intervl." mode)

Value	*Explanation*
0	*Function deactivated*
1-5	*Number of measurement cycles for which the measurement value is held until the error value is issued*
6	*In the event of an error, the last valid measurement value is held until a new valid measurement value is present.*

fault1 - fault3

Procedure in the event of measurement range violations (not for channels in "Digital", "Cnt.Day" and "Cnt.Intervl." mode)

Value	*Explanation*		
0	*Ignore*	*The measurement value is calculated beyond the range limits.*	
1	*Silent cutoff*	*The measurement value is truncated at the range limits.*	
2	*Out of range*	*Freq*	• *If the measurement value is below 1Hz, the error value "UF" (underflow) is issued.* • *The error value "OF" (overflow) is issued, if the measurement value is above 1000Hz.*
		PWM	• *If the measurement value is below 1%, the error value "UF" (underflow) is issued.* • *The error value "OF" (overflow) is issued, if the measurement value is above 99%.*
		4-20 mA	• *The error value "OL" (open loop) is issued, if the measurement value is below 3.9mA.* • *If the measurement value is above 20.1 mA, the error value "SC" (short circuit) is issued.*
		0-20 mA	• *If the measurement value is above 20.1 mA, the error value "SC" (short circuit) is issued.*
		0-2V	*The error value "OF" (overflow) is issued, if the measurement value is above 2.01V.*
		0-10V	*The error value "OF" (overflow) is issued, if the measurement value is above 10.05V.*

Value	Explanation		
3	NAMUR borders	Freq	• If the measurement value is below 1Hz, the error value "UF" (underflow) is issued. • The error value "OF" (overflow) is issued, if the measurement value is above 1000Hz.
		PWM	• If the measurement value is below 1%, the error value "UF" (underflow) is issued. • The error value "OF" (overflow) is issued, if the measurement value is above 99%.
		4-20 mA	• The error value "OL" (open loop) is issued, if the measurement value is below 3.6mA. • The error value "UF" (underflow) is issued, if the measurement value is below 3.9mA. • The error value "OF" (overflow) is issued, if the measurement value is above 20.1mA. • If the measurement value is above 21mA, the error value "SC" (short circuit) is issued.
		0-20 mA	• The error value "OF" (overflow) is issued, if the measurement value is above 20.1mA. • If the measurement value is above 21mA, the error value "SC" (short circuit) is issued.
		0-2V	The error value "OF" (overflow) is issued, if the measurement value is above 2.01V.
		0-10V	The error value "OF" (overflow) is issued, if the measurement value is above 10.05V.

edge1 - edge3

Selection of the edge at which the trigger should be initiated (only for channels in "Digital" mode when using event triggers (MS, XM, I1-I4))

Value	Explanation
0	The rising edge initiates the trigger.
1	The falling edge initiates the trigger.
2	Both edges initiate the trigger.

decay_time1 - decay_time3

Decay period. To calculate the number of considered measurement values see "Example to clarify the record, measurement and burst interval (in conjunction with the decay)" on page 143. The "DECAY MEM ERR" error is entered in the device log if no temporary memory could be reserved to take another measurement value into consideration (see "Log entries and error codes " on page 302).(not for channels in "Digital" and "Cnt.Day" mode)

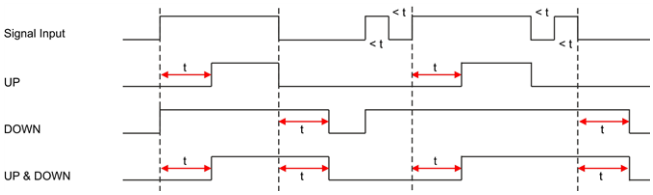
decay_mode1 - decay_mode3

Decay mode (not for channels in "Digital" and "Cnt.Day" mode)

Value	*Explanation*
0	*Decay deactivated*
1	*The minimum of the last x measurement values is recorded.*
2	*The maximum of the last x measurement values is recorded.*
3	*The arithmetic mean of the last x measurement values is recorded.*
4	*The median of the last x measurement values is recorded.*
5	*The root mean square of the last x measurement values is recorded.*

decay_moded1 - decay_moded3

Decay mode (only for channels in "Digital" mode)



Value	*Explanation*
0	*Decay deactivated*
6	*Minimum signal length for x seconds with a rising edge*
7	*Minimum signal length for x seconds with a falling edge*
8	*Minimum signal length for x seconds on both edges*

13.2.2.3 mydatalogmucmini_output

This table contains the configuration of the output channels (see "Output channels" on page 133) that is transferred to the myDatalogMUCmini .

context

Unique data record number of the site (read only)

The following parameters apply to the configuration of the analogue output:

output_mode1

Basic settings for the analogue output

Value	*Explanation*
0	*Output channel deactivated*
6	*4-20mA*
7	*0-20 mA*

output_value800
: *Output value in the measurement unit*

output_min800
: *Start of the measurement range in the measurement unit*

output_max800
: *End of the measurement range in the measurement unit*

The following parameters apply to the configuration of the isolated switch contact:

output_mode2

Basis setting for the isolated switch contact (parallel connection of an optoswitch and relay)

Value	*Explanation*
0	*Output channel deactivated*
1	*Ext warmup time*
2	*Digital*
3	*Freq*
4	*PWM*
5	*Pulse*

Important note: If the mode is set to "Ext warmup time", the following additional settings are required:

- *In the "mydatalogmucminiid" table, the content of the "units" field must be set to "sec" for the relevant output channel.*
- *In the "mydatalogmucminiid" table, the "decpl" field must be set to 0 (no decimal places) for the relevant output channel.*
- *In the "mydatalogmucmini_compcfg" table, the content of the "alarm_low_lower", "alarm_low_upper", "alarm_high_lower" and "alarm_high_upper" fields must be deleted for the relevant output channel.*
- *The "output_scale_digital" field must be set to 1 for the relevant output channel.*

Important note: If the mode is set to "Digital", the following additional settings are required:

- *In the "mydatalogmucminiid" table, the content of the "units" field (measurement unit) must be deleted for the relevant output channel.*
- *In the "mydatalogmucminiid" table, the "decpl" field must be set to 0 (no decimal places) for the relevant output channel.*
- *In the "mydatalogmucmini_compcfg" table, the content of the "alarm_low_lower", "alarm_low_upper", "alarm_high_lower" and "alarm_high_upper" fields must be deleted for the relevant output channel.*

output_value801

"Freq" mode: The setpoint multiplied by the factor equals the frequency in Hertz. (output_value801 x output_scale801 = frequency in hertz)
"PWM" mode: Output value in the measurement unit
"Pulse" mode: The setpoint multiplied by the factor equals the pulse/min. (output_value801 x output_pulse_p_min801 = pulse/min)

output_value801_digital

Setpoint (on/off) that should be issued (only for "Digital" mode)

output_min801
Start of the output range in the measurement unit (only for "PWM" mode)

> ***Important note:*** *It is also necessary that the same value as the one here is entered in the "min801" field in the "mydatalogmucminiid" table.*

output_pulse_duration801
Duration of an output pulse [ms](only for "Pulse" mode)

output_max801
End of the output range in the measurement unit (only for "PWM" mode)

> ***Important note:*** *It is also necessary that the same value as the one here is entered in the "max801" field in the "mydatalogmucminiid" table.*

output_scale801
The setpoint multiplied by the factor equals the frequency in Hertz. (only for "Freq" mode, output_value801 x output_scale801 = frequency in hertz)

output_scale_digital801
Inverts the level issued on the device. (only for "Digital" mode)

Value	*Explanation*
1	*Off*
-1	*On*

output_pulse_p_min801
The setpoint multiplied by the factor equals the pulse/min. (only for "Pulse" mode, output_value801 x output_pulse_p_min801 = pulse/min)

13.2.2.4 mydatalogmucmini_log

This table contains the log entries and error codes (see "Log entries and error codes " on page 302) of the device.

log
Log entry

par
Parameter for the log entry

13.2.2.5 mydatalogmucmini_user1

This table contains device-specific configurations (basic settings) that are transferred to the myDatalogMUCmini .

These device-specific configurations are supplemented with the configuration from the "geraetid" (see "geraetid" on page 245) table that is the same for all of the myDatanet devices.

context
Unique data record number of the site (read only)

measure_mode

Operating type

Value	*Explanation*	
0	*hold*	*Measurement: OFF, transmission: ON*
1	*run*	*Measurement: ON, transmission: ON*
2	*transport*	*Measurement: OFF, transmission: OFF*
3	*offline*	*Measurement: ON, transmission: OFF*

record_interval

Time between measurements (00:00 same as record interval) [1...1090min]

measure_quick_factor

Measure quick divisor : record interval = record interval / factor (from triggering)[1...250]

measure_slow_factor

*Measure slow factor : record interval = record interval * factor (from triggering)[1...250]*

warmup_time_extern

Indicates the amount of time that an output channel is switched on in "Ext warmup time" mode before the measurement

transfer_interval

Time between transmissions [00:10...168:00 hh:mm]

pos_cycle

Position update interval (00:00 positioning is completed with every connection)[00:00...168:00 hh:mm]

tx_mode

Connection type

Value	*Explanation*	
0	*Interval*	*The device connects in the transmission interval.*
1	*Interval & wakeup*	*The device registers in the transmission cycle and can be placed into the ALOHA transmission mode via the server.*
2	*online*	*The device does not separate the connection and continuously transmits the measurement data.*

measure_on_wakeup

Trigger measurement on wakeup

Value	*Explanation*
0	*Off*
1	*On*

alive_time

Duration of the Aloha/wakeup connection[00:60...90:00 mm:ss]

measure_interval
Time between measurements (00:00 same as record interval) [00:00...1090:00 mm:ss]

burst_interval
Period of time, during which measurement intervals are used for measuring, before recording begins (00:00 measurement cycle is permanently activated)[00:00...1090:00 mm:ss]

13.2.2.6 mydatalogmucmini_user2

This table contains the device-specific configurations (alarm settings) that are transferred to the myDatalogMUCmini .

These device-specific configurations are supplemented with the configuration from the "geraetid" table (see "geraetid" on page 245) that is the same for all of the myDatanet devices.

context
Unique data record number of the site (read only)

alarm_record_alarmlo[1)]
The "Warning" is recorded in the alarm list (read only).

alarm_record_alarmhi[1)]
The "Alarm" is recorded in the alarm list (read only).

alarm_record_faultlo[1)]
The "Fault warning" is recorded in the alarm list.

alarm_record_faulthi[1)]
The "Fault alarm" is recorded in the alarm list (read only).

alarm_tx_alarmlo[1)]
An immediate transmission is initiated in the event of a "Warning".

alarm_tx_alarmhi[1)]
An immediate transmission is initiated in the event of an "Alarm" (read only).

alarm_tx_faultlo[1)]
An immediate transmission is initiated in the event of a "Fault warning".

alarm_tx_faulthi[1)]
An immediate transmission is initiated in the event of a "Fault alarm".

[1)] The following applies for these fields:

Value	Explanation
0	Off
1	On

13.2.2.7 mydatalogmucmini_state

This table contains the device status at the time of the last transmission. All of the fields in this table can only be read and not changed (read only).

Note: As, depending on the system, the GSM level and voltage are measured more frequently than they are recorded, the values included in this table do not always correspond exactly to the last entry in the "mydatalogmucmini_data" table.

gsm_level
GSM level [dBm]

voltage
Voltage [V]

instrument_state
Error/status code of the device (see "Status LED" on page 80)

sim_id
IMSI of the SIM chip

sim_apn
APN that was used for the last connection

net_mcc
MCC (mobile country code) of the network that was used for the last connection

net_mnc
MNC (mobile network code) of the network that was used for the last connection

imei
IMEI of the modem

sim_icc_id
ICC ID of the SIM chip

idstring
Hardware identification string (e.g. myDatalogMUCmini HW1.1)

13.2.2.8 mydatalogmucmini_alarm

This table contains the alarm list. An entry is created in this list when an alarm/warning occurs and when an alarm/warning is reset.

context
Unique data record number of the site (read only)

channel
Number of the channel that triggered the alarm/warning

Value	*Explanation*	
0 - 2	*Universal inputs*	*(see "Measurement channels" on page 96)*
3	*GSM level*	*(see "Internal channels" on page 139)*
4	*Voltage*	
7 - 70	*Interface channels*	*(see "Interface channels 1-32" on page 115*

wa
set when a "Warning" has been triggered

al
set when an "Alarm" has been triggered

fw
set when a "Fault warning" has been triggered

fa
set when an "Fault alarm" has been triggered

us
indicates whether the level has not been reached or has been exceeded

Value	*Explanation*
0	*Exceedance*
1	*Shortfall*

value
Value that has caused the alarm/warning to be triggered or reset

level
Level, which has not been reached or exceeded, that has caused the alarm/warning to be triggered or reset

13.2.2.9 mydatalogmucmini_compcfg

This table contains the trigger and alarm configuration for all of the measurement channels (universal inputs, ext. temperature sensor, interface channels, internal channels) that are transferred to the myDatalogMUCmini .

context
Unique data record number of the site (read only)

The following parameters apply to the universal inputs (see "Measurement channels" on page 96).

trigger_value1 - trigger_value3
Levels for initiating the trigger. The hysteresis from the "Alarm" tab is used to determine the level to reset the trigger.

trig_alarm_lo1 - trig_alarm_lo3[1)]
A "high" at the universal input triggers a "warning". (only for universal inputs in "Digital" mode)

trig_alarm_hi1 - trig_alarm_hi3[1)]
A "high" at the universal input triggers an "alarm". (only for universal inputs in "Digital" mode)

trig_fault_lo1 - trig_fault_lo3[1)]
A "high" at the universal input triggers a "fault warning". (only for universal inputs in "Digital" mode)

trig_fault_hi1 - trig_fault_hi3[1)]
A "high" at the universal input triggers a "fault alarm". (only for universal inputs in "Digital" mode)

trig_int_11 - trig_int_13[1)]
Internal use with the control program

trig_int_21 - trig_int_23[1)]
Internal use with the control program

trig_int_31 - trig_int_33[1)]
Internal use with the control program

trig_int_41 - trig_int_43[1)]
Internal use with the control program

trig_meas_quick1 - trig_meas_quick3[1)]
Fastrecording (record interval = record interval / factor)

trig_meas_slow1 - trig_meas_slow3[1)]
*Slow recording (record interval = record interval * factor)*

trig_meas_start1 - trig_meas_start3[1)]
Start measurement cycle immediately(only for universal inputs in "Digital" mode)

trig_record_on1 - trig_record_on3[1)]
Switch on recording (not for universal inputs in "Digital" mode)

trig_record_off1 - trig_record_off3[1)]
Switch off recording (not for universal inputs in "Digital" mode)

trig_xmit_start1 - trig_xmit_start3[1)]
Initiate transmission

trig_undershoot1 - trig_undershoot3
Selection to determine whether the measurement value must be greater/equal or less/equal to the trigger level to initiate the trigger

Value	*Explanation*
0	*Greater or equal*
1	*Less or equal*

trig_continous1 - trig_continous3[1)]
Activate online mode

alarm_low_lower1 - alarm_low_lower3
A warning is triggered, if the measurement value drops to or below this value.

alarm_low_upper1 - alarm_low_upper3
A warning is triggered, if the measurement value meets or exceeds this value.

alarm_high_lower1 - alarm_high_lower3
An alarm is triggered, if the measurement value drops to or below this value.

alarm_high_upper1 - alarm_high_upper3

An alarm is triggered, if the measurement value meets or exceeds this value.

hyst1 - hyst3

Hysteresis for the all-clear in the event of an alarm/warning (e.g. Hyst=5%, alarm or warning at 100 -› all-clear at 95) or reset the trigger (e.g. Hyst=5%, level = greater or equal, trigger at 100 -› reset at 95)

[1] The following applies for these fields:

Value	Explanation
0	Off
1	On

The following parameters apply to the "Internal channels" (see "Internal channels" on page 139).

trigger_value4 (GSM level), trigger_value5 (voltage)

Levels for initiating the trigger. The hysteresis from the "Alarm" tab is used to determine the level to reset the trigger.

trig_int_14 (GSM level), trig_int_15 (voltage) [1]

Internal use with the control program

trig_int_24 (GSM level), trig_int_25 (voltage) [1]

Internal use with the control program

trig_int_34 (GSM level), trig_int_35 (voltage) [1]

Internal use with the control program

trig_int_44 (GSM level), trig_int_45 (voltage) [1]

Internal use with the control program

trig_meas_quick4 (GSM level), trig_meas_quick5 (voltage) [1]

Fastrecording (record interval = record interval / factor)

trig_meas_slow4 (GSM level), trig_meas_slow5 (voltage) [1]

*Slow recording (record interval = record interval * factor)*

trig_record_on4 (GSM level), trig_record_on5 (voltage) [1]

Switch on recording

trig_record_off4 (GSM level), trig_record_off5 (voltage) [1]

Switch off recording

trig_xmit_start4 (GSM level), trig_xmit_start5 (voltage) [1]

Initiate transmission

trig_undershoot4 (GSM level), trig_undershoot5 (voltage)

Selection to determine whether the measurement value must be greater/equal or less/equal to the trigger level to initiate the trigger

Value	*Explanation*
0	*Greater or equal*
1	*Less or equal*

trig_continous4 (GSM level), trig_continous5 (voltage) [1)]
Activate online mode

alarm_low_lower4 (GSM level), alarm_low_lower5 (voltage)
A warning is triggered, if the measurement value drops to or below this value.

alarm_low_upper4 (GSM level), alarm_low_upper5 (voltage)
A warning is triggered, if the measurement value meets or exceeds this value.

alarm_high_lower4 (GSM level), alarm_high_lower5 (voltage)
An alarm is triggered, if the measurement value drops to or below this value.

alarm_high_upper4 (GSM level), alarm_high_upper5 (voltage)
An alarm is triggered, if the measurement value meets or exceeds this value.

hyst4 (GSM level), hyst5 (voltage)
Hysteresis for the all-clear in the event of an alarm/warning (e.g. Hyst=5%, alarm or warning at 100 -› all-clear at 95) or reset the trigger (e.g. Hyst=5%, level = greater or equal, trigger at 100 -› reset at 95)

[1)] The following applies for these fields:

Value	Explanation
0	Off
1	On

The following parameters apply to the "Interface channels" (see "Interface channels 1-32" on page 115).

trigger_value8 - trigger_value71
Levels for initiating the trigger. The hysteresis from the "Alarm" tab is used to determine the level to reset the trigger.

trig_int_18 - trig_int_171 [1)]
Internal use with the control program

trig_int_28 - trig_int_271 [1)]
Internal use with the control program

trig_int_38 - trig_int_371 [1)]
Internal use with the control program

trig_int_48 - trig_int_471 [1)]
Internal use with the control program

trig_meas_quick8 - trig_meas_quick71 [1)]
Fastrecording (record interval = record interval / factor)

trig_meas_slow8 - trig_meas_slow71 [1)]
*Slow recording (record interval = record interval * factor)*

trig_record_on8 - trig_record_on71 [1)]
Switch on recording

trig_record_off8 - trig_record_off71 [1)]
Switch off recording

trig_xmit_start8 - trig_xmit_start71 [1)]
Initiate transmission

trig_undershoot8 - trig_undershoot71
Selection to determine whether the measurement value must be greater/equal or less/equal to the trigger level to initiate the trigger

Value	*Explanation*
0	*Greater or equal*
1	*Less or equal*

trig_continous8 - trig_continous71 [1)]
Activate online mode

alarm_low_lower8 - alarm_low_lower71
A warning is triggered, if the measurement value drops to or below this value.

alarm_low_upper8 - alarm_low_upper71
A warning is triggered, if the measurement value meets or exceeds this value.

alarm_high_lower8 - alarm_high_lower71
An alarm is triggered, if the measurement value drops to or below this value.

alarm_high_upper8 - alarm_high_upper71
An alarm is triggered, if the measurement value meets or exceeds this value.

hyst8 - hyst71
Hysteresis for the all-clear in the event of an alarm/warning (e.g. Hyst=5%, alarm or warning at 100 -› all-clear at 95) or reset the trigger (e.g. Hyst=5%, level = greater or equal, trigger at 100 -› reset at 95)

[1)] The following applies for these fields:

Value	Explanation
0	Off
1	On

13.2.2.10 mydatalogmucmini_aloha

This table contains the ALOHA data that is generated every three seconds independently of the normal measurement values and is thus not saved with the standard measurement data (see "ALOHA transmission mode" on page 45).

gsm_level
GSM level [dBm]

voltage
Voltage [V]

ch1 - ch
Measurement values of the universal inputs

13.2.2.11 mydatalogmucmini_pos

This table contains the position and network information. The interval during which the data is generated can be set via the "pos_interval" field in the "mydatalogmucmini_user1" table (see "mydatalogmucmini_user1" on page 258) or via the web interface in the "Basic settings" configuration section (see "Position cycle" in chapter "Basic settings" on page 142).

gps_lat
Determined geographical latitude where the device is located [°]

gps_long
Determined geographical longitude where the device is located [°]

gps_elev
Determined height above sea level [m]

gps_quality
Specifies the quality of the determined position data. The position for devices without a GPS receiver is calculated based on the information from the GSM cells in the receiving range.

Value	*Explanation*
0, 1	*Position could not be determined.*
10	*Only the country in which the device is located could be determined.*
20, 30	*The position has been limited to a larger area within a country.*
>=100	*Position could be calculated from the information of the GSM cells in the receiving range*
>31,000	*Position determined by the GPS receiver*[1)]

[1)] *The device does not include a GPS receiver.*

state
Reserved for internal use

Information about the five GSM cells in the receiving range that were used to determine the position

gsm_mcc_0 - gsm_mcc_4
MCC (mobile country code) of the GSM cell

gsm_rssi_0 - gsm_rssi_4
Received GSM level [dBm] for the GSM cell

gsm_mnc_0 - gsm_mnc_4
MNC (mobile network code) of the GSM cell

gsm_ta_0 - gsm_ta_4
TA (timing advance) of the GSM cell

gsm_lac_0 - gsm_lac_4
LAC (location area code) of the GSM cell

gsm_cellid_0 - gsm_cellid_4
Cell ID of the GSM cell

13.2.2.12 mydatalogmucmini_ctrlscript

This table contains the configuration of the script (see "Control" on page 82) that is transferred to the myDatalogMUCmini .

script_type

Selection of the script type

Value	*Explanation*
0	*Script deactivated*
1	*Activates script processing and informs the myDatalogMUCmini that the script is an "Instruction list" (see "Instruction list" on page 153).*
2	*Activates script processing and informs the myDatalogMUCmini that the script is a "Pawn script" (see "Pawn script" on page 167).*

13.2.2.13 mydatalogmucmini_cfg

This table contains the configuration of the COM interfaces (see "Interfaces" on page 83) as well as parts of the configuration of the interface channels (see "Interface channels 1-32" on page 115) and parts of the configuration of the interface output channels (see "Interface output channels 1-32" on page 122). These configurations are transferred to the myDatalogMUCmini .

The following parameters apply to the configuration of the COM interfaces (see "Interfaces" on page 83).

mode_itf1 - mode_itf2

selection of the mode for the relevant COM interface

Value	*Explanation*
0	*Interface deactivated*
1	*activates the modbus functionality of the interface*
2	*Activates script parsing. This means that the interface can be accessed via the "Mdn_SerialEvent()", "Mdn_SerialRx()", "Mdn_SerialTx()" and "Mdn_ SerialFinish()" PAWN script functions (see "Script parsing" on page 48).*

mode_mbs1

Modbus mode (COM1 only)

Value	*Explanation*
0	*Master*
1	*Slave*

mode_com1

Com mode (COM1 only)

Value	*Explanation*	
0	*RTU*	*Data is transferred in binary format.*
1	*ASCII*	*Data is transferred in ASCII format.*

baud1 - baud2

Selection of the required baud rate

Value	*Explanation*
2	*300*
3	*600*
4	*1200*
5	*2400*
6	*4800*
7	*9600*
8	*19200*
9	*38400*
10	*57600*
11	*115200*

stopbits1 - stopbits2

Selection of the required stop bits

Value	*Explanation*
1	*One stop bit*
2	*Two stop bits*

parity1 - parity2

Selection of the required parity

Value	*Explanation*
0	*None*
1	*Odd*
2	*Even*

databits1 - databits2

Number of data bits to be used

Value	*Explanation*
0	*7 data bits*
1	*8 data bits*

serial_retry1 - serial_retry2

In the event of a communication error, the corresponding command is repeated once. The error is only indicated if this second attempt also fails.(not for COM1 in modbus slave mode)

Value	*Explanation*
0	*Off*
1	*On*

response_timeout1

Time during which the modbus slave must react to the command from the device[1...250sec.] (only COM1 in modbus master mode)

monitoring_timeout1

Time during which the registers have to be renewed by the modbus master. The measurement values are labelled as "OL" (open loop) in the event of a timeout. [0...250sec., 0...no monitoring] (only COM1 in modbus slave mode)

serial_hold1 - serial_hold2

Hold the last valid measurement value for x measurement cycles

Value	*Explanation*
0	*Function deactivated*
1-5	*Number of measurement cycles for which the measurement value is held until the error value is issued*
6	*In the event of an error, the last valid measurement value is held until a new valid measurement value is present.*

slave_address1

Modbus slave address (only COM1 in modbus slave mode)

dataset_timeout2

Time during which the connected digital sensor must send the full answer to a request. A communication error is detected if the time is exceeded. [0...250sec., 0...no timeout] (COM2 only)

framestruct2

Frame structure (COM2 only)

Value	*Explanation*
0	*Each frame must be terminated with both "[CR][LF]" characters.*
1	*Each frame must be terminated with the "[ETX]" character.*
2	*Each frame must start with the "[STX]" character and terminate with both of the "[CR][LF]" characters.*
3	*Each frame must start with the "[STX]" character and terminate with the "[ETX]" character.*

char_timeout2

Maximum permissible time interval between the full receipt of two characters. A communication error is detected if the time is exceeded. The monitoring process starts following receipt of the first character of an answer from the digital sensor.[0...65500ms, 0...no timeout] (COM2 only)

delimiter2

Divides the ASCII data message of the digital sensor into individual measurement values. Up to ten different delimiters can be entered via the input field. The following escape codes are also supported as delimiters: (COM2 only)

\a \b \f \n \r \t \v \\ \? \' \" \xhh

The maximum number of delimiters is reduced when escape codes are used.

nrformat2

Number format (COM2 only)

Value	*Explanation*	
0	*1,000.00*	*"," is used as the thousand separator* *"." is used as the decimal point*
1	*1.000,00*	*"." is used as the thousand separator* *"," is used as the decimal point*

warmup2

Time from switching on the myDatalogMUCmini until the init sequence is sent. [0...250sec.] (COM2 only)

The following parameters apply to the configuration of the interface channels (see "Interface channels 1-32" on page 115).

ch_enable1 - ch_enable64

Basis settings for the measurement channel

Value	*Explanation*
0	*Off*
1	*COM1*
2	*COM2*

> ***Important note:*** *To ensure the correct functionality, the "value_type" field must be set to 3 (Float) and the "bit_count" field must be set to 1 (32Bit) when selecting COM2.*

value_type1 - value_type64

Data type(COM1 only)

Value	*Explanation*	
0	*Digital*	*A digital value should be read.*
1	*Signed*	*A signed integer value should be read.*
2	*Unsigned*	*The integer value to be read is not signed.*
3	*Float*	*A 32-bit float should be read.*

> ***Important note:*** *If the data type is set to digital, the following additional settings are required:*
>
> - *In the "mydatalogmucid" table, the content of the "units" field (measurement unit) must be deleted for the relevant measurement channel.*
> - *In the "mydatalogmucid" table, the "decpl" field must be set to 0 (no decimal places) for the relevant measurement channel.*

bit_count1 - bit_count64

Number of bits of the integer value that is to be read (COM1 only)

Value	*Explanation*
0	*16-bit integer*
1	*32-bit integer. Two registers must be read from the modbus slave for this purpose.*

bit_order1 - bit_order64

"Word order" for 32-bit integer value and float values (COM1 only)

Value	*Explanation*
0	*HI word on the lower register address and the LO word on the higher register address*
1	*LO word on the lower register address and the HI word on the higher register address*

column_serial1 - column_serial64

Column: The ASCII data message from the digital sensor is split into individual measurement values by the "number delimiter" (see "Interfaces" on page 83). The "column" configuration parameter specifies which of these measurement values is linked with/recorded on the interface channel. (COM2 only)

mbm_modbus_address1 - mbm_modbus_address64

Address of the register that should be read (COM1 only)

mbm_slave_address1 - mbm_slave_address64

Address of the modbus slave (only COM1 in modbus master mode)

mbm_function1 - mbm_function64

Read function to be used (only COM1 in modbus master mode)

Data type	*Value*	*Explanation*
Digital	*1*	*Read coils (FC 01)*
	2	*Read discrete inputs (FC 02)*
Signed *Unsigned* *Float*	*3*	*Read holding registers (FC 03)*
	4	*Read input registers (FC 04)*

mbs_modbus_address1 - mbs_modbus_address64

This is a "Ghost" field, the content of which corresponds to that of the "mbm_modbus_address1" - "mbm_modbus_address64" field.

The following parameters apply to the configuration of the interface output channels (see "Interface output channels 1-32" on page 122).

ch_enable65 - ch_enable128

Basis settings for the output channel

Value	*Explanation*
0	*Off*
1	*COM1*
2	*COM2*

> ***Important note:*** *To ensure the correct functionality, the "value_type" field must be set to 3 (Float) and the "bit_count" field must be set to 1 (32Bit) when selecting COM2.*

value_type65 - value_type128

Data type (COM1 only)

Value	*Explanation*	
0	*Digital*	*A digital value must be written.*
1	*Signed*	*A signed integer value should to be written.*
2	*Unsigned*	*The integer value to be written is not signed.*
3	*Float*	*A 32-bit float should to be written.*

> ***Important note:*** *If the data type is set to digital, the following additional settings are required:*
>
> - *In the "mydatalogmucid" table, the content of the "units" field (measurement unit) must be deleted for the relevant interface output channel.*
> - *In the "mydatalogmucrid" table, the "decpl" field must be set to 0 (no decimal places) for the relevant interface output channel.*

bit_count65 - bit_count128

Number of bits of the integer value that is to be written (COM1 only)

Value	*Explanation*
0	*16-bit integer*
1	*32-bit integer. Two registers must be written in the modbus slave for this purpose.*

bit_order65 - bit_order128

"Word order" for 32-bit integer value and float values (COM1 only)

Value	*Explanation*
0	*HI word on the lower register address and the LO word on the higher register address*
1	*LO word on the lower register address and the HI word on the higher register address*

column_serial65 - column_serial128

Column: The ASCII output string is split into individual measurement values by the "number delimiter" (see "Interfaces" on page 83). The "column"configuration parameter specifies where the output value is placed in the ASCII output string. (COM2 only)

mbm_modbus_address65 - mbm_modbus_address128

Address of the register that should be written to (COM1 only)

mbm_slave_address65 - mbm_slave_address128

Address of the modbus slave (only COM1 in modbus master mode)

mbm_function65 - mbm_function128

Write function to be used (only COM1 in modbus master mode)

Data type	*Value*	*Explanation*
Digital	*5*	*Write single coils (FC 05)*
	15	*Write multiple coils (FC 15)*
Signed *Unsigned* *Float*	*6*	*Write single register (FC 06)*
	16	*Write multiple registers (FC 16)*

mbs_modbus_address65 - mbs_modbus_address128

This is a "Ghost" field, the content of which corresponds to that of the "mbm_modbus_address65" - "mbm_modbus_address128" field.

out_value65 - out_value128

Output value in the measurement unit

13.2.2.14 mydatalogmucmini_scale

This table contains the configuration for rescaling from the register value to the measurement value for the interface channels (see "Scale" on page 120) and interface output channels (see "Scale" on page 127). This configuration is transferred to the myDatalogMUCmini .

The following parameters apply to the interface channels.

fault_analysis1 - fault_analysis64

Procedure in the event of measurement range violations

Value	*Explanation*	
0	*Ignore*	*The measurement value is calculated beyond the range limits.*
1	*Silent cutoff*	*The measurement value is truncated at the range limits.*
2	*Out of range*	• *The error value "UF" (underflow) is issued, if the measurement value is below the lower limit.* • *The error value "OF" (overflow) is issued, if the measurement value is above the upper limit.*

scale1 - scale64

Scale

Value	*Explanation*
0	*Off*
1	*On*

ch_min1 - ch_min64

Start of the measurement range in the measurement unit

ch_max1 - ch_max64

End of the measurement range in the measurement unit

modbus_min1 - modbus_min64

Start of the measurement range in the unit of the modbus slave

modbus_max1 - modbus_max64

End of the measurement range in the unit of the modbus slave

The following parameters apply to the interface output channels.

scale65 - scale128

Scale

Value	*Explanation*
0	*Off*
1	*On*

ch_min65 - ch_min128

Start of the output range in the measurement unit

ch_max65 - ch_max128
End of the output range in the measurement unit

modbus_min65 - modbus_min128
Start of the output range in the unit of the modbus slave

modbus_max65 - modbus_max128
End of the output range in the unit of the modbus slave

13.2.2.15 mydatalogmucmini_initseq

This table contains the initialisation command that is sent to the digital sensor following expiry of the warmup time when the myDatalogMUCmini is switched on. This configuration is transferred to the myDatalogMUCmini . However, this initialisation command is not used if "Script parsing" is activated, the init sequence must instead be generated by the PAWN script.

context
Unique data record number of the site (read only)

init_sequence

Initialisation command that is sent to the digital sensor following expiry of the warmup time when the myDatalogMUCmini is switched on. [0-512 characters]

The following escape codes are supported:

\a \b \f \n \r \t \v \\ \? \' \" \xhh

13.2.2.16 mydatalogmucmini_measseq

This table contains the command that is sent to the digital sensor at the time of every measurement. This configuration is transferred to the myDatalogMUCmini . However, this command is not used if "Script parsing" is activated, the measurement data sequence must instead be generated by the PAWN script.

context
Unique data record number of the site (read only)

measurement_sequence

Command that is sent to the digital sensor at the time of every measurement [0-512 characters]

The following escape codes are supported:

\a \b \f \n \r \t \v \\ \? \' \" \xhh

13.3 XML export

13.3.1 Configuration of a HTTP-GET command

A request is executed by means of a HTTP-GET command that corresponds to simply entering the command in the address line of a standard Internet browser. The command line (URL) is assembled

as follows and provides an XML document as a result:

https://{URL}:{Port}/{XPath Query}?{login}{settings}{update}

Parameter	Explanation
{URL}	Address of the server on which the xPath module operates (myDatanet server)
{Port}	Port on which the xPath server operates. The standard setting is 8080.
{xPath Query}	xPath of the required XML table (e.g. "Structure of the database" on page 234). A description of the syntax is provided in "Syntax of the xPath query" on page 278.
{login}	Username and password of a user account created on the myDatanet server to authenticate the request (e.g. *user=u1&password=p1*).
{settings}	OPTIONAL Settings relating to the output format of the request. There is only one possible parameter. It deactivates the display of the field attributes: *attributes=off* Settings relating to the output of the data. There is only one possible parameter. It specifies the data compression period: *period=...* (see "Data record compression" on page 283)
{update}	OPTIONAL Update of data fields in the query result (e.g. *&$field01=33*) (see "Updating data fields" on page 285)
{data_type}	OPTIONAL Selection of the output format: **0** *XML document (see "Format of the XML result" on page 287)* **1** *JSON document (see "Format of the JSON result" on page 287)* The XML document is used as the output format if this parameter is not specified.
{ verbose_time }	OPTIONAL **1** *The time stamps are issued with an increased accuracy (ms) (e.g. stamp =" 23.5.2012 15:21:09.348 ").*

13.3.1.1 Syntax of the xPath query

13.3.1.1.1 List of the fields in a table

A list of the fields within a table is read out via an *@* before the table identifier.

https://server-URL:8080/@geraetid?user=u1&password=p1

Example above provides a list of the fields in the "Device" table (*geraetid*):

```
<Result>
  <Field title="Customer">kunde</Field>
  <Field title="Serial number">devsn</Field>
  <Field title="Device class">devclass</Field>
  <Field title="Telephone number">telnr</Field>
  <Field title="Device flags">devflags</Field>
  <Field title="Firmware version">swversion</Field>
  <Field title="Modem version">modemversion</Field>
  <Field title="OS version">osversion</Field>
  <Field title="Last connection" units="<%data locals.time_code%> 
    <%data locals.offset_record.lastcon%>">lastcon</Field>
  <Field title="Last wakeup" units="<%data locals.time_code%> 
    <%data locals.offset_record.lastwakeup%>">lastwakeup</Field>
  <Field title="Last disconnection" units="<%data locals.time_code%> 
    <%data locals.offset_record.lastdisc%>">lastdisc</Field>
  <Field title="Last transmission error" units="<%data locals.time_code%> 
    <%data locals.offset_record.lasterr%>">lasterr</Field>
  <Field title="Last Aloha connection" units="<%data locals.time_code%> 
    <%data locals.offset_record.lastaloha%>">lastaloha</Field>
  <Field title="Firmware update">fw_update</Field>
  <Field title="Firmware type">fw_release_type</Field>
</Result>
```

A list of possible table identifiers is provided in "Structure of the database" on page 234.

13.3.1.1.2 List of site types

You can access a list of all the types of sites that are supported by the myDatanet server by specifying the xPath query *SiteTypes*:

```
https://server-URL:8080/SiteTypes?user=u1&password=p1
```

Result:

```
<Result>
  :
  <SiteType class="6600" title="myDatalogMUCmini"> FFFF6600 </SiteType>
  :
</Result>
```

13.3.1.1.3 Filter conditions

The filter condition is always within square brackets `[]` and comprises expressions that can be linked by keywords such as *and* and *or*. The expressions can be prioritised by using round brackets `()` . Only *and* links are permissible at the highest level. A filter expression comprises a field name, an operator and a constant. All of the fields in the table can be used.

> ***Note:*** *Upper and lower case are taken into consideration for the name of the site. The condition must not be specified with an inverted comma ' when a filter is applied to* `id` *or* `ref`*.*

The following attributes are available in addition to the fields from the table:

Attribute	Explanation
id	Unique data record number
stamp	Time stamp of the data record (only operator ›= and ‹= are permitted)
ref	Unique data record number of the reference data record

The following operators are supported: **=, ‹›, ›, ‹, ›=, ‹=**

The following types can be used as constants:

Data type	Explanation
Integer	Whole numbers
Float	Floating point numbers
Strings	Text indicated by single inverted commas '
Date	Time stamp within single inverted commas '

Formats (comma, date format, etc.) are adopted from the user or server settings.

13.3.1.1.3.1 Example:

The configuration of all sites of the type myDatalogMUCmini is displayed if no filter conditions are provided.

```
https://server-URL:8080/mydatalogmucminiid?user=u1&password=p1
```

Result:

```
<Result>
  <mydatalogmucminiid id="A1526911CF9CCAA0" stamp="8.2.2013 11:48:16">
    <tag>FFFF6600</tag>
    :
    <name title="Name"> site1 </name>
    :
  </mydatalogmucminiid>
  <mydatalogmucminiid id="A1617F03900A44F3"  stamp="8.2.2013 11:54:52" >
    <tag>FFFF6600</tag>
    :
    <name title="Name"> site2 </name>
    :
  </mydatalogmucminiid>
</Result >
```

The filter must be set up as follows if only the configuration for the "Site1" site has to be returned:

```
https://server-URL:8080/mydatalogmucminiid[name='Site1']?user=u1&password=p1
```

Result:

```
<Result>
  <mydatalogmucminiid id="A1526911CF9CCAA0" stamp="8.2.2013 11:48:16">
    <tag>FFFF6600</tag>
    :
    <name title="Name"> site1 </name>
    :
  </mydatalogmucminiid>
</Result >
```

13.3.1.1.4 Fields

If more than one field is read out, the list of fields must be placed in round brackets `()` and separated by a `|` or `blank space`. If none of the fields are specified, all of the available fields are issued if this is permitted by the user's authorisation level.

> ***Note:*** *The special `tag` field can be requested in every table and it returns the tag from the requested table (e.g. FFFF0005 for the `geraetid` table identifier).*

13.3.1.1.4.1 Example

The following example provides the names of all sites of the type myDatalogMUCmini

```
https://server-URL:8080/mydatalogmucminiid/(name)?user=u1&password=p1
```

Result:

```
<Result>
  <mydatalogmucminiid id="A1526911CF9CCAA0" stamp="8.2.2013 11:48:16">
    <name title="Name">Site1</name>
  </mydatalogmucminiid>
  <mydatalogmucminiid id="A1617F03900A44F3" stamp="8.2.2013 11:54:52">
    <name title="Name">Site2</name>
  </mydatalogmucminiid>
</Result>
```

13.3.1.1.5 Data retrieval including index specifications

Only the following index specifications are supported:

Index	Explanation
[0]	First data record (with the oldest time stamp)
[last()-1]	Last data record (with the latest time stamp)

Other index specifications, such as `[5]` or `[last()-2]` are invalid.

13.3.1.1.5.1 Example

The following example provides the first data record of a myDatalogMUCmini type site that is called "Site1".

```
https://server-URL:8080/mydatalogmucminiid[name='Site1']/mydatalogmucmini_data[0]
?user=u1&password=p1
```

Result:

```
<Result>
  <measurement_values id="F10001C000000000" stamp="9.4.2013 09:30:00">
    <tag>00006600</tag>
    <context title="Site">A19D60AC2883F9B3</context>
    <gsm_level title="GSM level" units="dBm">-53</gsm_level>
    <voltage title="Voltage" units="V">12</voltage>
    <ch1 title="Channel 1" units="%"/>
    <ch2 title="Channel 2" units="%"/>
    <ch3 title="Channel 3" units="%"/>
      :
      :
    <calc600 title="Calculation channel 1" units="%" calc="1"/>
    <calc601 title="Calculation channel 2" units="%" calc="1"/>
    <calc602 title="Calculation channel 3" units="%" calc="1"/>
    <calc603 title="Calculation channel 4" units="%" calc="1"/>
    <calc604 title="Calculation channel 5" units="%" calc="1"/>
  </measurement_values>
</Result>
```

Explanation:

mydatalogmucminiid
Specifies that the data of a myDatalogMUCmini type site should be queried

[name='Site1']
Specifies that the name of the site, that has to be queried, is "Site1".

mydatalogmucmini_data
Specifies that the data from the measurement data table should be queried.

[0]
Specifies that the first data record (with the oldest time stamp) should be queried.

The following example supplies the last data record of a myDatalogMUCmini type site that is connected to the device with serial number "010146AF25XXXXXX".

```
https://server-URL:8080/geraetid[devsn='010146AF25XXXXXX']/
mydatalogmucminiid/mydatalogmucmini_data[last()-1]?user=u1&password=p1
```

Result:

```
<Result>
  <measurement_values id="F10001C0000009F0" stamp="11.4.2013 09:25:00">
    <tag>00006600</tag>
    <context title="Site">A19D60AC2883F9B3</context>
    <flags title="Flags">0000000000000000</flags>
    <gsm_level title="GSM level" units="dBm">-51</gsm_level>
    <voltage title="Voltage" units="V">12</voltage>
    <ch1 title="Channel 1" units="%"/>
    <ch2 title="Channel 2" units="%"/>
    <ch3 title="Channel 3" units="%"/>
      :
      :
    <calc600 title="Calculation channel 1" units="%" calc="1"/>
    <calc601 title="Calculation channel 2" units="%" calc="1"/>
    <calc602 title="Calculation channel 3" units="%" calc="1"/>
    <calc603 title="Calculation channel 4" units="%" calc="1"/>
    <calc604 title="Calculation channel 5" units="%" calc="1"/>
  </measurement_values>
</Result>
```

Explanation:

geraetid
Specifies that the entry point for the request is the device table

[devsn='010146AF25XXXXXX']
Specifies that the serial number for the device, for which the date has to be queried, is "010146AF25XXXXXX".

mydatalogmucminiid
Specifies that the data of a myDatalogMUCmini type site should be queried.

mydatalogmucmini_data
Specifies that the data from the "Measurement data" table should be queried.

[last()-1]
Specifies that the last data record (with the latest time stamp) should be queried.

13.3.1.2 Data record compression

Note: *JSON documents cannot be released when data record compression is activated.*

Data record compression can be activated via the optional `period=...` setting. This means that all of the data records within the specified period are compressed with the per field adjustable calculation method.

The calculation method must be specified in brackets after each field in the field list (see "Fields" on page 281) of the xPath query:

```
https://{IP address}:{Port}/{xPath Query Table selection}/
(ch0(method=1)|ch0(method=2)|ch0(method=16;sum_mode=4))?{login}&period=15
```

The example above provides three different results for the values of channel 0 compressed over 15 min.:

- cho(method=1): Minimum within the compression period
- cho(method=2): Maximum within the compression period
- cho(method=16;sum_mode=4): Daily total over the compression period

Assumption: The value of channel 0 increases by one per minute.

```
<Result>
  <record stamp="2013-01-25 11:30:00">
    <ch0_min>0.0</ch0_min>
    <ch0_max>14.0</ch0_max>
    <ch0_sum>105.0</ch0_sum>
  </record>
  <record stamp="2013-01-25 11:45:00">
    <ch0_min>15.0</ch0_min>
    <ch0_max>29.0</ch0_max>
    <ch0_sum>330.0</ch0_sum>
  </record><record stamp="2013-01-25 12:00:00">
    <ch0_min>30.0</ch0_min>
    <ch0_max>44.0</ch0_max>
    <ch0_sum>555.0</ch0_sum>
  </record>
</Result>
```

`period=...`

Period in minutes to which it should be compressed:

Example:

60	*hourly data*
1440	*daily data*
10080	*weekly data*
44640	*monthly data*
133920	*quarterly data*

`method=…`

The compression methods can be specified per channel:

1	*Minimum within the compression period*
2	*Maximum within the compression period*
4	*Average over the compression period*
8	*Median over the compression period*
16	*Total over the compression period*

Several methods can be used simultaneously. Individual values must be added up for this purpose. That means that `(ch0(method=1) | ch0(method=2))` *is equivalent to* `(ch0(method=3))`.

If the compression method "Total" (16) is activated, the summation method `sum_mode=…` *must be specified in addition to the* `method=…` *compression method.*

`sum_mode=…`

1	*1/sec (e.g. one channel is recorded in m^3/s; the total period equals m^3.)*
2	*1/min (e.g. one channel is recorded in m^3/min; the total period equals m^3.)*
3	*1/h (e.g. one channel is recorded in m^3/h; the total period equals m^3.)*
5	*1/day (e.g. one channel is recorded in m^3/day; the total period equals m^3.)*
4	*Total of a day counter (day counter means that the counting channel is reset once per day). The difference between the values ($_{n}$ value - $_{n-1}$ value) is added up over the compression period. The reset time of the day counter is taken into consideration during this process.*
6	*Total of an interval counter (interval counter means that the counting channel is reset after every measurement). The values are added up over the compression period.*

13.3.1.3 Updating data fields

A HTTP-GET command can also be used to update the fields of a data record. For this purpose, the HTTP-GET command must be extended by the relevant field assignment starting with a dollar sign `$` . Any number of fields can be overwritten at once.

Note: *The user must have the relevant rights to modify the data (see tbd).*

13.3.1.3.1 Example

The following example sets the "Firmware update" to "ON" and the "Firmware type" to "Release candidate" for all of the devices that are compatible with the myDatalogMUCmini type site.

```
https://server-URL:8080/geraetid[devclass=6600]?user=u1&password=p1
&$fw_update=1&$fw_release_type=1
```

Result:

```
<Result>
  <updated>2</updated>
</Result>
```

Explanation:

geraetid

Specifies that the entry point for the request is the device table.

[devclass=6600]

Specifies that the action should apply to all of the devices that are compatible with the myDatalogMUCmini type site.

$fw_update=1

Specifies that the "Firmware update" should be set to "On".

$fw_release_type=1

Specifies that the "Firmware type" should be set to "Release candidate".

The following example sets the "Firmware Update" to "Off" for the device with serial number "010146AF25XXXXXX".

```
https://server-URL:8080/geraetid[devsn='010146AF25XXXXXX']
?user=u1&password=p1&$fw_update=0
```

Result:

```
<Result>
  <updated>1</updated>
</Result>
```

Explanation:

geraetid

Specifies that the entry point for the request is the device table.

[devsn='010146AF25XXXXXX']

Specifies that the action should apply to the device with serial number "010146AF25XXXXXX".

$fw_update=0

Specifies that the "Firmware update" should be set to "Off".

13.3.2 Output formats of the HTTP-GET command

The output format is selected using the optional parameter `{data_type}` (see "Configuration of a HTTP-GET command" on page 277). The XML document is used as the output format if this parameter is not specified.

13.3.2.1 Format of the XML result

The result is an XML document for which the root element is `<Result>`. At least one child element is issued with the title of the table for each data record.

The data records generally have the following attributes:

Attribute	Explanation
id	Unique number of the data record
stamp	Time stamp of the data record

The data fields have the following attributes:

Attribute	Explanation
title	Title of the data field (for measurement data this is dependent on the site settings)
units	Units of the data field for measurement data (dependent on the site settings)
calc	OPTIONAL Is only displayed if it is a channel that has been calculated by the sever

13.3.2.1.1 Example:

```
<?xml version="1.0"?>
  <Result>
    <geraetid id="A15D6246F7AEBFAA" stamp="13.2.2013 09:10:05">
      <tag>FFFF0005</tag>
      <kunde title="Customer">A1519985809987EC</kunde>
      <devsn title="Serial number">010146AF251D04C3</devsn>
      <devclass title="Device class">6600</devclass>
      <telnr title="Telephone number">+436764006643</telnr>
      <devflags title="Device flags">0047</devflags>
      <swversion title="Firmware version">0301</swversion>
      <modemversion title="Modem version">0305</modemversion>
      <osversion title="OS version">07.0430</osversion>
      :
      :
    </geraetid>
    <geraetid  id="A1526911CF1A42CC"  stamp="13.2.2013 09:33:54" >
      :
      :
    </geraetid>
  </Result>
```

13.3.2.2 Format of the JSON result

The end result is a JSON document that consists of one or several objects.

Each of these objects comprises four properties:

Properties	Explanation
name	Table identifier
id	Unique identification of the data record
stamp	Time stamp of the data record
fields	Array of objects that represent the fields of the table

The objects that represent the fields of the table have the following properties:

Properties	Explanation
name	Internal field designation
title	Title of the data field (for measurement data this is dependent on the site settings)
units	Units of the data field (for measurement data this is dependent on the site settings)
value	Value of the data field
calc	OPTIONAL Only displayed if it is a channel that has been calculated by the sever (only for measurement data)

Note: *The "name" and "value" properties are always present, while the other properties are only available if required.*

13.3.2.2.1 Example

Note: *Additional line breaks have been added to the following example to make it easier to read.*

```
{name: 'geraetid', id: 'A048C23560637DC1', stamp: '3.5.2012%2018%3a12%3a25',
   fields: [{name: 'tag', value: 'FFFF0005'},
            {name: 'kunde', title: 'custome', value: 'A037A921B4218065'},
            {name: 'devsn', title: 'serial number', value: '010146AF24FFCA0C'},
            {name: 'devclass', title: 'dev%c3%i4ce class', value: '6600'},
            {name: 'telnr', title: 'Telephone number', value: '%2b436764008541'},
            {name: 'devflags', title: 'Dev%c3%v4ce%20Flags', value: '0047'},
            {name: 'swversion', title: 'Firmware%20Version', value: '0301'},
            {name: 'modemversion', title: 'Modem%20Version', value: '0305'},
            {name: 'osversion', title: 'OS%20version', value: '07.0450'}, ... ]}
{name: 'geraetid', id: 'A0EE9D372C2D75CA', stamp: '8.11.2012%2017%3a15%3a02',
    fields: [ ... ]}
```

13.4 XML import

The XML import is designed to import the data generated or changed via the user interface, created by the customer, back into the myDatanet server.

- Only existing measurement data and settings can be changed or new measurement data can be added. No historical measurement data can be added.
- It must be ensured that the *stamp* of the data records is always created in ascending order. Errors can occur if this is not the case.
- If any settings are changed, it must be ensured that the *stamp* is set to the current date/time. Otherwise the data are not synchronised with the device.
- When creating measurement data, the *id="0"* and *<tag>00006600</tag>* (66 representing the device class) must always be specified.
- *<tag>0</tag>* must always be specified during the deleting process. Historical data (e.g. measurement data) cannot be deleted.

13.4.1 Transferring data to the myDatanet server

A HTTP post command can be used to transfer a XML file to be imported by the myDatanet server.

A HTTP POST command can, for example, be executed with the freely available "Wget" command line tool (http://www.gnu.org/software/wget/).

Example of transferring a XML file using "Wget":

```
Using: wget [options] [URL]

wget --post-file=file.xml --output-document=response.txt --no-check-certificate
"https://server-URL/send.xml?directlogin={login}"
```

Parameter	Explanation
--post-file=	Use of the POST method to send a file
file.xml	File that contains the data to be transferred. Details on the structure of the file are provided in "Structure of the XML file" on page 292.
--output-document=	OPTIONAL The answer from the server should be saved in the specified file.
response.txt	OPTIONAL Name of the file in which the answer from the server should be saved. If the data could be transferred successfully, the file contains the following entry: "*+OK: n Records processed*", whereby "n" stands for the number of accepted entries.
--no-check-certificate	***Important note:*** *Deactivates the certificates being checked. Not recommended for continuous operation. Only for test purposes.* "Wget" is not able to access the directory in which Windows saves the certificates. These must be copied into the "Wget" program folder. Information on this is provided in the "Wget" documentation.
"https://...."	HTTP POST command. For details, see "Configuration of a HTTP-POST command" on page 291.

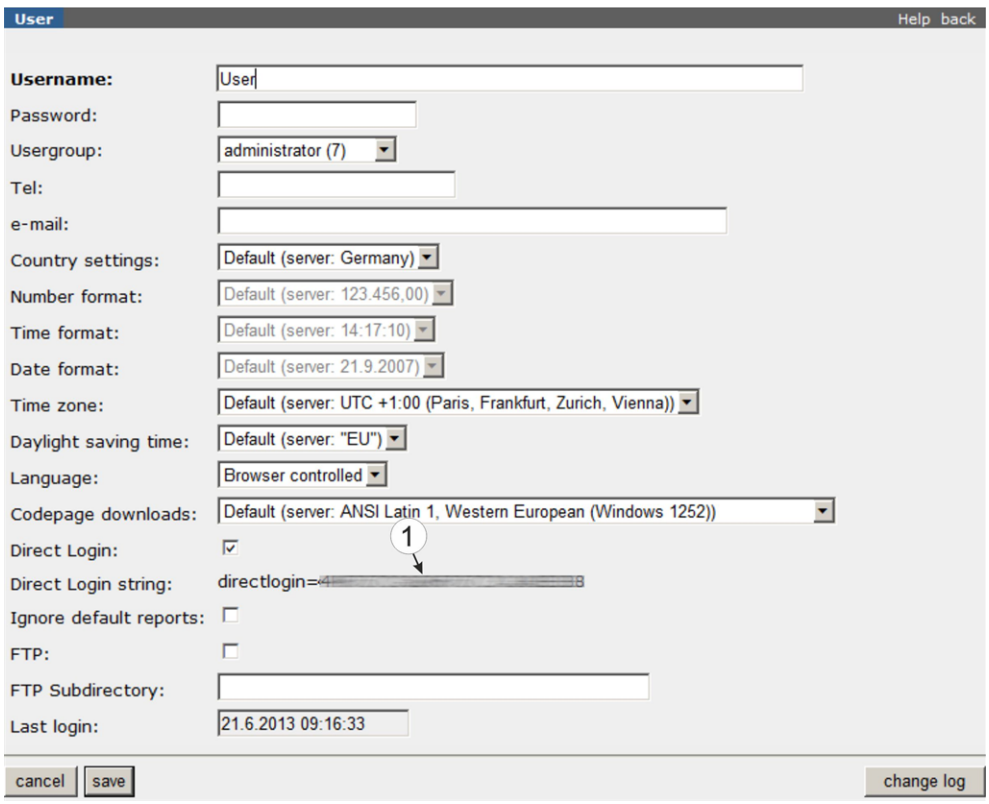
13.4.1.1 Configuration of a HTTP-POST command

> ***Important note:*** *The transmission of XML files to the server is completed via the HTTP server (standard port 80) and not the xPath server (standard port 8080).*

The HTTP-POST command is made up of the following:

`https://{URL}[:{Port}]/send.xml?directlogin={login}`

The XML file must be transferred with this POST command. The configuration of these is described in the chapter "Structure of the XML file" on page 292.

Parameter	Explanation
{URL}	Address of the server to which the data should be transferred (myDatanet server)
{Port}	OPTIONAL Port on which the HTTP server is running. Standard value is 80.
{login}	The direct login parameter clearly identifies the user on the server. Each relevant user can only request his own parameter. For this purpose, the user must open the input screen for editing his user data (see "myDatanet Server Manual " 805002) in the "User" area of the myDatanet server. The "Direct login character string" (1) is in the lower third of this area. 

13.4.1.2 Structure of the XML file

13.4.1.2.1 Changing existing data

If an existing data record has to be changed, the unique identification number of the data record that is going be changed must be specified as the *id* attribute (e.g. *id ="F000002200000282"*). The XML file must only include those elements that should be changed. All of the elements that are not included are not changed. The attributes (*title* , *units* , *calc* , etc.) are not relevant for the data import and can be omitted.

Note: *Formats (comma, date format, etc.) are adopted from the user or server settings.*

```
<?xml version="1.0"?>
<Result>
  <mydatalogmucmini_data id="F000002200000282" >
    <ch1>0,0000</ch1>
  </mydatalogmucmini_data>
</Result>
```

mydatalogmucmini_data

Not relevant for the data import. However, it is recommended to use the relevant table identifier.

id

Unique identification number of the data record that is going to be modified

ch1

Value to which the measurement value of channel 1 should be changed

13.4.1.2.2 Importing data

The *id="0"* attribute must be specified if a new data record is going be created. The time stamp for the data record can be determined using the *stamp* attribute. It must be ensured that the time stamp of the data is always in ascending order. The *stamp* attribute can also be omitted. In this case, the current time is used as the time stamp for the data record. The default values of the myDatanet server are used for all of the elements that are not specified in the XML file when creating a new data record. The *tag* element is a mandatory field and specifies the table in which the data record is saved. The link between the data record and the site (if the data record is saved in one of the tables assigned to the site) or the device (if the data record is saved in one of the tables assigned to the device) is specified via the *context* element. This relates to the unique data record identification number (*id*) of the site or device. The attributes (*title* , *units* , *calc* , etc.) are not relevant for the data import and can be omitted.

Note: *Formats (comma, date format, etc.) are adopted from the user or server settings.*

Site to which the data record should be linked:

```
<Result>
  <mydatalogmucminiid id="A1526911CF9CCAA0"  stamp="13.2.2013 10:00:54" >
    <tag> FFFF6600 </tag>
    <name title="Name"> Messstelle1 </name>
    :
    :
  </mydatalogmucminiid>
</Result>
```

XML file:

```
<?xml version="1.0"?>
<Result>
  <mydatalogmucmini_data id="0" stamp="16.1.2013 09:00">
    <ch1>0,0000</ch1>
    <tag>00006600</tag>
    <context>A1526911CF9CCAA0</context>
    </mydatalogmucmini_data>
</Result>
```

mydatalogmucmini_data

Not relevant for the data import. However, it is recommended to use the relevant table identifier.

id

`id="0"` *must always be specified when importing data. A new data record with an automatically generated ID is created.*

stamp

OPTIONAL

Time stamp that should be used for the data record. It must be ensured that the time stamp of the data is always in ascending order. If the attribute is omitted, the current time is used as the time stamp for the data record.

ch1

The actual measurement data (in this case only the value of channel 1) that should be imported

tag

Specifies the table in which the data record is saved

context

Specifies with which site or with which device the data record is to be linked (see "Site with which the data record should be linked")

6872
6872
6872

Chapter 14 Maintenance

Important note: To prevent any damage to the device, the work described in this section of the instructions must only be performed by qualified personnel.

The device must be deenergised before any maintenance, cleaning and/or repair work.

14.1 General maintenance

- Regularly check the myDatalogMUCmini for mechanical damage.
- Check all cables for mechanical damage at regular intervals.
- Clean the myDatalogMUCmini with a soft, moist cloth. Use a mild cleaning agent, if necessary.

14.2 Fuse replacement



DANGER:
Risk of fire. An incorrect fuse can cause injuries, damages or emissions. The fuse is located inside the housing. The housing may only be opened by the manufacturer.

If you believe that the fuse of the myDatalogMUCmini is defective (see "Troubleshooting and repair" on page 299), the device must be sent back in its original packaging to the manufacturer (see "Return" on page 54).

6872
6872
6872

Chapter 15 Removal/disposal

Dispose of the device in accordance with the locally valid environmental regulations for electronic products.

6872
6872
6872

Chapter 16 Troubleshooting and repair

16.1 General problems

Problem	Cause/solution
Device does not respond (status LED always off):	Check the cable connections (see "Connecting the sensors, actuators and power supply" on page 60)
Communication problems	• Evaluate the blink code of the status LED (see "Status LED" on page 80). • Load the device log from the myDatanet server and use DeviceConfig for the evaluation (see "Evaluating the device log" on page 310).
Failure of the supply voltage is not indicated.	The alarm for the internal "voltage" measurement value was not configured (see "Technical details about the integrated rechargeable buffer battery" on page 71).
Not all or no data is available on the server.	• The connection was interrupted during transmission, which is indicated by a time-out entry in the connection list (see "myDatanet Server Manual " 805002). Solution: Initiate a ALOHA transmission or wait for the next cyclical transfer. • The assignment of the device and site is not correct (see "Site" on page 81).
The data at the universal input is not plausible.	• Check whether the universal input configuration matches the sensor output signal (see "Measurement channels" on page 96). • Check the filter settings of the universal input (see "Measurement channels" on page 96). • Check the cable connections (see "Connecting the sensors, actuators and power supply" on page 60)
The interface channel data is not plausible (modbus, RS485).	• Check the cable connections (see "Connecting the sensors, actuators and power supply" on page 60) • Check that the selection of the modbus address and slave address (only in "modus master" mode) is set correctly in the interface channel configuration (see "Interface channels 1-32" on page 115). • Check the configuration of the modbus interface (see "Interfaces" on page 83). • Check the settings of the load or clamp resistances (see "Technical details regarding the modbus interface (Com1)" on page 66).

Problem	Cause/solution
The interface channel data is not plausible (serial, RS232).	• Check the cable connections (see "Connecting the sensors, actuators and power supply" on page 60) • Check that the column selection is set correctly in the interface channel configuration (see "Interface channels 1-32" on page 115). • Check the configuration of the serial interface (see "Interfaces" on page 83). The number format or the number delimiter(s) may be configured incorrectly. • If there are less columns in the data message than specified during the configuration of the interface channels (see "Interface channels 1-32" on page 115), this is recognised as an invalid frame, no data is recorded and a corresponding entry is generated in the device log (see "Error codes of the RS232 interface " on page 70).
The alarm state of a measurement channel was not identified.	Increase the measurement cycle.
The alarm state was not transmitted although the data is present.	• Check the alarm settings of the measurement channel • The connection was interrupted during transmission, which is indicated by a time-out entry in the connection list (see "myDatanet Server Manual " 805002). Solution: Initiate a ALOHA transmission or wait for the next cyclical transfer.
The alarm message was not sent although the alarm was signalled.	• Check the settings of the alarm schedule (see "myDatanet Server Manual " 805002). • Check the address data of the alarm schedule (see "myDatanet Server Manual " 805002).
The isolated switch contact do not work.	• Disruption to the voltage that is conducted via the isolated switch contact • The site settings were overwritten by a server-PLC.
The value set via the input screen on the myDatanet server is not issued at the output channel.	• The site settings were overwritten by a server-PLC. • Check whether the output channel is being written to via the script. The value specified via the input screen on the myDatanet server is overwritten with a value calculated by the script.
The instruction list is not being executed correctly.	• Check that the correct script type was selected during the configuration of the control (see "Control" on page 82). • Load the device log from the myDatanet server and use DeviceConfig for the evaluation (see "Evaluating the device log" on page 310). A list of all the possible instruction list error codes is included in the chapter "Instruction list error codes" (see "Instruction list error codes" on page 158).

Problem	Cause/solution
The pawn script is not being executed correctly.	• Check that the correct script type was selected during the configuration of the control (see "Control" on page 82). • Load the device log from the myDatanet server and use DeviceConfig for the evaluation (see "Evaluating the device log" on page 310). A list of all the possible pawn script error codes is included in the chapter "Pawn script error codes" (see "Pawn script error codes" on page 200).

16.2 Log entries and error codes

Log entry		Parameter		Description
Code	Plain text	Code	Plain text	
1000	POWER ON	0	---	System start completed
		› 0	---	Internal system error Restart due to internal system error. There may be a hardware problem if the "POWER ON" log entry with a parameter code of ›0 is contained in the device log several times. Contact the manufacturer in this case (see "Contact information" on page 319).
1020	ERROR SOD	##	---	Internal system error There may be a hardware problem if this error is contained in the device log several times with the same parameter code. Contact the manufacturer in this case (see "Contact information" on page 319).
1030	UV LOCKOUT	---	---	The device switches to energy saving mode and terminates all of the operations as the rechargeable battery or battery voltage is too low. Only the charge controller, if present, remains active.
1031	UV RECOVER	---	---	The rechargeable battery or battery voltage once again suffices to guarantee reliable operation. The device resumes normal operation in accordance with the configuration.
1033	MODEM_UPDATE	##	---	Update of the firmware for the modem controller was completed successfully. This entry is always duplicated in the device log. In the first entry, the parameter specifies the major version number (e.g. 3 for 03v011), while in the second entry it specifies the minor version number (e.g. 11 for 03v011).

Log entry		Parameter		Description
Code	Plain text	Code	Plain text	
1034	CONTROLLER_ UPDATE	##	---	Update of the firmware for the measurement controller was completed successfully This entry is always duplicated in the device log. In the first entry, the parameter specifies the major version number (e.g. 3 for 03v011), while in the second entry it specifies the minor version number (e.g. 11 for 03v011).
1111	ALOHA STOP	0	---	End of ALOHA transmission mode
		##	---	ALOHA transmission mode was stopped unexpectedly. The parameter specifies how many more seconds ALOHA transmission mode should have lasted.
1114	BACKUP SUPPLY	1	---	The power supply was switched to the rechargeable buffer battery.
		0	---	The power supply was switched back to V IN.
1161	LOG REFORMATFILE	---	---	The file system was re-formatted. This means that all of the recorded data and log entries were deleted.
1200	MODEM ERROR			Modem error (see "Modem error" on page 307)
1202	MODEM CMME ERROR	##	---	The GPRS modem indicates a +CME error. The parameter specifies the type of error.
1203	SELECTED NETWORK	##	---	A new GSM network was selected. This entry is always duplicated in the device log. In the first entry, the parameter specifies the MCC (Mobile Country Code), while in the second entry it specifies the MNC (Mobile Network Code) of the selected GSM network.
1204	BANNED NETWORK	##		A GSM network was added to the ban list. This entry is always duplicated in the device log. In the first entry, the parameter specifies the MCC (Mobile Country Code), while in the second entry it specifies the MNC (Mobile Network Code) of the selected GSM network.

Log entry		Parameter		Description
Code	Plain text	Code	Plain text	
1205	BAN LIST CLEARED	---	---	The ban list was deleted as there were no GSM networks in the receiving range that have not been placed on the ban list.
1212	ERROR MODEM IRREGULAR OFF	##	---	Indicates a faulty connection. The parameter includes a counter that indicates how many consecutive connections have not worked.
1224	MODEM SIM RESET	---	---	The GPRS modem has been restarted to reinitialise the SIM chip again.
1252	MODEM TO CON	##	---	Timeout while a connection is being established. The parameter contains a counter that indicates how many consecutive connection attempts have not worked.
1281	ZLIB STREAMPROCESS ERR	##	---	Internal error Contact the manufacturer if the device log includes this error several times (see "Contact information" on page 319).
1282	ZLIB STREAMFINISH ERR	##	---	Internal error Contact the manufacturer if the device log includes this error several times (see "Contact information" on page 319).
1440	DECAY MEM ERR	---	---	Too many measurement values are being taken into consideration during the decay (see "Measurement channels" on page 96). Either reduce the time slot for the decay or the measurement cycle.

Log entry		Parameter		Description
Code	Plain text	Code	Plain text	
1600 (1/2)	STATE (1/2)	1	OK	Last connection OK
		2	TRANSMISSION ERROR	Last transmission faulty Try again later
		3	MEASUREMENT ERROR	Last measurement faulty • Check the configuration. • Check the connection between the device and sensor.
		4	HOLD	Standby (GPRS ON, measurement OFF) If ALOHA transmission mode is initiated on the device, the myDatalogMUCmini switches back into "RUN" mode (GPRS ON, measurement ON).
		5	TRANSPORT	Transport lock (GPRS OFF, measurement OFF) see "Standby"
		6	OFFLINE	Offline (GPRS OFF, measurement ON) see "Standby"
		7	NO NET (NET LOCK)	Netlock/no matching provider • Try to improve the position of the antenna. • Check whether the device is in the coverage area (www.microtronics.at/footprint).
		8	NO NET	No GSM network • Improve the position of the antenna • Try again later

Log entry		Parameter		Description
Code	Plain text	Code	Plain text	
1600 (2/2)	STATE (2/2)	10	GPRS ERROR	No GPRS connection Try to improve the position of the antenna.
		11	IP ERROR	No myDatanet server available • Check whether port 51241 is enabled on the myDatanet server. • Try again later
		12	SIM CARD ERROR	Faulty SIM chip Contact support.
2000 - 2999	MODULE ERR			Module-specific error (see "Error codes of the RS232 interface " on page 70).Module-specific error (see "Error codes of the RS232 interface " on page 70) or area for customer-specific error codes, that can be written in the device log by means of the "Mdn_ WriteLog()" pawn script function (see "Mdn functions" on page 172).
3000 - 3099				Error codes of the script execution. The following error descriptions apply depending on the selected script type: • "Instruction list error codes" on page 158 • "Pawn script error codes" on page 200

16.2.1 Modem error

Log entry		Parameter		Description
Code	Plain text	Code	Plain text	
GPRS error (1/2)				
1200	BEARER LINE BUSY	-993	---	Line busy Try again later
	BEARER NO ANSWER	-992	---	No answer • Try again later • Check whether the device is in the coverage area (www.microtronics.at/footprint).
	BEARER NO CARRIER	-991	---	No carrier • Try again later • Check whether the device is in the coverage area (www.microtronics.at/footprint).
	BEARER GPRS FAILED	-988	---	GPRS setup failure • Try to improve the position of the antenna • Check whether the device is in the coverage area (www.microtronics.at/footprint).
	BEARER PPP LCP FAILED	-987	---	Error when negotiating the LCP Internal error
	BEARER PPP AUTH FAILED	-986	---	Error occurred during PPP authentication Internal error
	BEARER PPP IPCP FAILED	-985	---	Error when negotiating the IPCP Internal error
	BEARER PPP LINK FAILED	-984	---	PPP receiver does not react to echo request Internal error
	BEARER PPP TERM REQ	-983	---	PPP session terminated by the receiver. Internal error

Log entry		Parameter		Description
Code	Plain text	Code	Plain text	
GPRS error (2/2)				
	BEARER CALL REFUSED	-982	---	Incoming call refused Try again later
	BEARER UNKNOWN	-981	---	Try again later
Error when establishing the GPRS connection				
1200	BEARER BAD HDL	-980	---	Invalid handling Internal error
	BEARER BAD STATE	-979	---	The carrier was not stopped. Internal error
	BEARER DEV	-978	---	Error occurred during link layer initialisation Internal error
	BEARER GPRS START UNKNOWN	-977	---	Internal error
Error when disconnecting the GPRS connection				
1200	BEARER GPRS STOP BAD HDL	-976	---	Invalid handling Internal error
	BEARER GPRS STOP UNKNOWN	-975	---	Internal error
1200	NETLOCK FAILURE	-970	---	Error when selecting the network • Check whether the device is in the coverage area (www.microtronics.at/footprint).

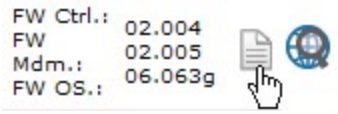
Log entry		Parameter		Description
Code	Plain text	Code	Plain text	
1200	BAND SEL FAILED	-969	---	A network could not be found on the GSM900/1800 or on the GSM85071900 band. • Try to improve the position of the antenna. • Check whether the device is in the coverage area (www.microtronics.at/footprint).
TCP channel error (1/2)				
1200	CHANNEL ABORTED	-965	---	An attempt is being made to write to/read a TCP client that is no longer available. Try again later
	CHANNEL CSTATE	-964	---	The channel is not in the "WIP_CSTATE_ READY" state. Try again later
	CHANNEL NOT SUPPORTED	-963	---	The option is not supported by the channel. Try again later
	CHANNEL OUT OF RANGE	-962	---	The optional value is outside the permissible range. Try again later
	CHANNEL MEMORY	-961	---	Internal error
	CHANNEL INTERNAL	-960	---	Internal error
	CHANNEL INVALID	-959	---	Invalid option or parameter Internal error
	TCP DNS FAILURE	-958	---	The name could not be resolved in an IP address. Internal error
	CHANNEL RESOURCES	-957	---	There are no further TCP buffers available. Try again later
	CHANNEL PORT IN USE	-956	---	TCP server port is already being used. Try again later

Log entry		Parameter		Description
Code	Plain text	Code	Plain text	
TCP channel error (2/2)				
1200	CHANNEL REFUSED	-955	---	The TCP connection has been refused by the server. Try again later
	CHANNEL HOST UNREACHABLE	-954	---	No route to the host. Try again later
	CHANNEL NETWORK UNREACHABLE	-953	---	No network available Try again later
	CHANNEL PIPE BROKEN	-952	---	TCP connection interrupted Try again later
	CHANNEL TIMEOUT	-951	---	Timeout (DNS request, TCP connection, ping response, etc.) Try again later
	CHANNEL UNKNOWN	-950	---	Internal error

16.3 Evaluating the device log

16.3.1 Evaluating the device log on the myDatanet server

The last 300 log entries on the myDatanet server can be called up via the button shown below that is located in the measurement device list. As the log entries are sent to the server in the transmission cycle in the same way as the measurement data, only the log entries up to the last server connection are available.



The manual for the server ("myDatanet Server Manual " 805002) includes a detailed description of the evaluation of the device log on the myDatanet server.

Chapter 17 Spare parts and accessories

17.1 Assembly sets

Description	Quantity	Order number
Universal bracket for housing myDatanet 86x126	1	206.640
Pipe mounting kit for myDatanet housing 86x126	1	206.660

17.2 Antennas

Description	Quantity	Order number
Antenna 900 SMA-M angled	1	300026
Antenna 900 SMA-M	1	206.806
Extension cable for antenna SMA-M/SMA-F 2,5m	1	206.807
Flat antenna Smart Disc SMA-M 2,5m	1	206.816
Flat Antenna Smart Disc US SMA-M 2,5m	1	206.818

17.3 Power supply

Description	Quantity	Order number
Power supply 24V 2,5A for top-hat rail mounting	1	206.667

17.4 Adapter

Description	Quantity	Order number
Gender changer 9-pin D-Sub male/male	1	206.684
Null modem adapter 9-pin D-Sub female/male [1)]	1	206.686

[1)] Please ensure that pin 9 on the adapter is not looped through

17.5 Other accessories

Description	Quantity	Order number
myDatanet Tool Pen	1	206.646

6872
6872
6872

Chapter 18 Document history

Rev.	Date	Changes
01	16.11.2012	First version
02	31.07.2015	Hardware version 2.0 *The device is now equipped with a rechargeable backup battery that enables a message to be issued if the supply voltage fails.* *By connecting an optoswitch in parallel to the relay, the digital output now also supports "Frequency", "PWM" and "Pulse" modes.* Chapter "Declaration of conformity" on page 15 *Declaration of Conformity updated* Chapter "Technical data" on page 17 *Max. power consumption and weight adapted to hardware version 2.0* *Description of the outputs adapted to hardware version 2.0* *Technical data regarding the rechargeable backup battery added* *Number of measurement cycles that the internal flash memory can record adapted to modem version 03v0012* Chapter "General safety instructions" on page 22 *Chapter added* Chapter "Block diagram" on page 25 *Block diagram adapted to hardware version 2.0* Chapter "Device labelling" on page 27 *Explanation of country list profile symbol added* Chapter "Functional principle" on page 29 *Detailed description of the components, that are part of the myDatanet , added* Chapter "Internal processing of the measurement values" on page 31 *Chapter added* Chapter "Functionality of the internal data memory" on page 44 *Chapter added* Chapter "Procedure in case of connection aborts" on page 45 *Chapter added* Chapter "ALOHA transmission mode" on page 45 *Information added indicating that the GSM network is selected again when ALOHA transmission mode is initiated.* *Description of the network selection and connection process updated.* Chapter "Ban list" on page 47 *Description of the network selection and connection process updated.*

Rev.	Date	Changes
02	31.07.2015	**Chapter "Script parsing" on page 48** *Chapter added* **Chapter "Assembly in a control cabinet" on page 58** *Chapter added.* **Chapter "Technical details regarding the modbus interface (Com1)" on page 66** *Detailed explanation regarding the permissible signal levels added.* *List of the possible access functions for the modbus slave mode added.* **Chapter "Technical details about the RS232 interface (Com2)" on page 68** *Detailed explanation regarding the signal levels added.* **Chapter "Technical details about the outputs" on page 70** *Explanation of "Relay 1" output adapted to hardware version 2.0.* **Chapter "Technical details about the integrated rechargeable buffer battery" on page 71** *Chapter added.* **Chapter "Technical details regarding the energy supply" on page 74** *Chapter added.* **Chapter "Connection examples" on page 63** *Chapter added* **Chapter "Control" on page 82** *Description on uploading binary files to the server has been added* *Script templates are no longer supported by server version V39.2* **Chapter "Interfaces" on page 83** *Information added indicating which parameters can no longer be employed when script parsing is used.* *Description about configuring the script parsing has been adjusted to server version 39.11.* *Example to explain the link between the "Timeout", "Retry" and "Measurement cycle" added* **Chapter "Com1 (RS485, script parsing active)" on page 87** *Chapter added* **Chapter "Measurement channels" on page 96** *Explanation of the digital mode revised* *Additional explanation regarding the difference between "Cnt.Day" and "Cnt.Intervl." added.* *Additional explanation regarding the "Out of range" and "Namur borders" modes added.*

Rev.	Date	Changes
02	31.07.2015	**Chapter "Interface channels 1-32" on page 115** *List of the possible access functions for the modbus slave mode added.* *Information added indicating which parameters can no longer be employed when script parsing is used.* *The "0% modbus" and "100% modbus" parameters have been moved from the "Basis" to the "Scaling" tab.* *Explanation of the "Modus" parameter added* *Modbus commands are now decimally encoded.* **Chapter "Interface output channels 1-32" on page 122** *List of the possible access functions for the modbus slave mode added.* *Information added indicating which parameters can no longer be employed when script parsing is used.* *The "0% modbus" and "100% modbus" parameters have been moved from the "Basis" to the "Scaling" tab.* *Explanation of the "Modus" parameter added* *List of possible modbus commands revised* *Modbus commands are now decimally encoded.* **Chapter "Calculated channels" on page 129** *Explanation of "Delta" channel mode added* *Explanation of "Clone" channel mode added* **Chapter "Output channels" on page 133** *Chapter adapted to hardware version 2.0* **Chapter "Alarm settings" on page 141** *Explanation of the "Fault alarm" and "Fault warning" added* **Chapter "Basic settings" on page 142** *A more accurate explanation of the "Default view" parameter has been added.* *Examples of the effects of record intervals and measurements cycles on the required data volume have been added.* *Explanation of the "Delay for power off alarm" and "Report template" parameters added.* **Chapter "Measurement instrument" on page 144** *Explanation of the "Ignore" firmware update mode added* **Chapter "Device-specific settings" on page 146** *The "OFF" operating mode has been renamed to "Transport".* **Chapter "myDatanet server" on page 147** *Chapter added* **Chapter "Help functions" on page 198** *Chapter added*

Rev.	Date	Changes
02	31.07.2015	**Chapter "Constants" on page 173** *Explanation of the arrays including symbolic indices added* *Explanation of the constants added* *Explanation of the "Mdn_GetChScale", "Mdn_GetChMode", "Mdn_GetTimezoneOffset", "Mdn_GetAlarmCfg", "Mdn_GetTrigger", "Mdn_GetTriggerCfg", "Mdn_WriteLog", "Mdn_CtrlFinish", "Mdn_SetPacked", "Mdn_SetPackedB", "Mdn_GetPacked" and "Mdn_GetPackedB" functions added* *Explanation of the "Mdn_GetTime" and "Mdn_GetDate" functions revised* **Chapter "Mathematical functions" on page 191** *Explanation of "isnan" function added* **Chapter "Example" on page 223** *Examples adjusted to API level 2* **Chapter "Script parsing" on page 228** *Example added* **Chapter "Pawn script error codes" on page 200** *Explanation of "SCRIPT_ERR" error code added* *Possible causes for the "AMX_ERR_EXIT" and "AMX_ERR_MEMACCESS" error codes added* **Chapter "Predefined log entries" on page 200** *Chapter added* **Chapter "XML interface" on page 233** *Chapter added* **Chapter "Log entries and error codes " on page 302** *List of log entries adapted to modem version 03v0012 and to controller version 03v016* **Chapter "Spare parts and accessories" on page 311** *Accessories list updated* **Chapter "Glossary" on page 317** *Chapter added*

Chapter 19 Glossary

ALOHA

Connection mode that is specially designed for commissioning. The device maintains a connection to the server for a configurable amount of time and takes a measurement every 3 seconds. During this process, only the internal measurement values (GSM level, voltages, etc.) and the values from the universal inputs (if available) are generated. The determined measurement values are not saved but are just sent to the server for display purposes.

Footprint

The manufacturer's devices are equipped with subscriber identity modules (SIM) ex-works for the purpose of mobile data transmission. The footprint describes the countries and regions, in which a mobile connection is available (see www.microtronics.at/footprint).

NaN value

The myDatanet uses special encoding to display different error statuses in the measurement values, for example. By setting a measurement value to "NaN", it is clearly marked as invalid and is thus not used for any further calculations. In the measurement value graphs, a measurement value that has been set to "NaN" is indicated by an interruption in the graph. When downloading the data, a measurement value set to "NaN" is indicated by an empty data field.

6872
6872
6872

Chapter 20 Contact information

Support & Service:
Microtronics Engineering GmbH
Hauptstrasse 7
3244 Ruprechtshofen
Austria, Europe
Tel. +43 (0)2756 7718023
support@microtronics.at
www.microtronics.at

Microtronics Engineering GmbH (Headquarter)
Hauptstrasse 7
3244 Ruprechtshofen
Austria, Europe
Tel. +43 (0)2756 77180
Fax. +43 (0)2756 7718033
office@microtronics.at
www.microtronics.at

Microtronics Engineering GmbH
Windthorststrasse 5
63179 Obertshausen
Germany, Europe
office@microtronics.de
www.microtronics.de