

Benutzerhandbuch myDataLogMUC xG/4G

Gültig ab:

- Firmware Version: 03v002
- Modem Version: 05v000
- Server Version: 49v011
- Hardware Version: 3.0



Kapitel 1 Inhaltsverzeichnis

Deckblatt	1
Kapitel 1 Inhaltsverzeichnis	3
Kapitel 2 Konformitätserklärung	13
2.1 myDatalogMUC 2G/4G EU	13
Kapitel 3 Technische Daten	15
Kapitel 4 Allgemeine Angaben	19
4.1 Übersetzung	19
4.2 Copyright	19
4.3 Gebrauchsnamen	19
4.4 Sicherheitshinweise	19
4.4.1 Verwendung der Gefahrenhinweise	20
4.4.2 Allgemeine Sicherheitshinweise	20
4.4.3 Sicherheits-/Vorsichtsmaßnahmen im Umgang mit GSM/GPRS-Modems	20
4.4.3.1 Sicherheits-/Vorsichtsmaßnahmen für den GSM/GPRS-Modemeinbau	21
4.4.3.2 Sicherheitsmaßnahmen für den Antenneneinbau	21
4.5 Übersicht	22
4.5.1 Blockschaltbild	22
4.6 Bestimmungsgemäße Verwendung	24
4.7 Allgemeine Produktinformationen	24
4.8 Gerätekenzeichnung	25
4.9 Einbau von Ersatz- und Verschleißteilen	26
4.10 Aufbewahrung des Produkts	26
4.11 Gewährleistung	26
4.12 Haftungsausschluss	27
4.13 Pflichten des Betreibers	27
4.14 Anforderungen an das Personal	28
Kapitel 5 Funktionsprinzip	29
5.1 Interne Verarbeitung der Messwerte	31
5.1.1 Filter-Modul	32
5.1.2 Overflow-Modul	33
5.1.3 Scale-Modul (Eingänge)	34

5.1.4 Decay-Modul.....	35
5.1.5 Hold-Modul.....	36
5.1.6 Alarm/Trigger-Modul.....	36
5.1.7 Control-Modul.....	36
5.1.8 Record-Modul.....	37
5.1.9 Setpoint-Modul.....	39
5.1.10 Scale-Modul (Ausgänge).....	40
5.2 Funktionsweise des internen Datenspeichers.....	40
5.3 Vorgehensweise bei Verbindungsabbrüchen.....	41
5.4 Aloha-Übertragungsmodus.....	41
5.5 Automatische Auswahl des GSM-Netzes.....	42
5.6 Device Logic-Abarbeitung.....	42
5.6.1 Device Logic Parsing.....	43
Kapitel 6 Lagerung, Lieferung und Transport.....	47
6.1 Eingangskontrolle.....	47
6.2 Lieferumfang.....	47
6.3 Lagerung.....	47
6.4 Transport.....	47
6.5 Rücksendung.....	48
Kapitel 7 Installation.....	49
7.1 Abmessungen.....	49
7.2 Montage des myDatalogMUC xG/4G.....	49
7.2.1 Hutschienenmontage.....	50
7.2.2 Montage in einem Schaltschrank.....	51
7.3 Sicherheitshinweise zur Verkabelung.....	52
7.3.1 Hinweise zur Vermeidung elektrostatischer Entladungen (ESD).....	52
7.4 Elektrische Installation.....	52
7.4.1 Anschluss der Sensoren, der Aktoren und der Versorgung.....	52
7.4.1.1 Anschlussbeispiele.....	56
7.4.2 Anschluss der GSM-Antenne.....	56
7.4.3 Technische Details zu den Universaleingängen.....	57
7.4.3.1 0/4...20mA Modus.....	57
7.4.3.2 0...2V Modus.....	57

7.4.3.3 0...10V Modus.....	57
7.4.3.4 Standard Digitalmodi (PWM, Frequenz, Digital, Tageszähler, Impulszähler).....	57
7.4.4 Technische Details zur PT100/1000-Schnittstelle.....	58
7.4.5 Technische Details zu den Modbus-Schnittstellen (Com1, Com3).....	58
7.4.5.1 Modbus-Slave Modus.....	60
7.4.6 Technische Details zur RS232-Schnittstelle (Com2).....	60
7.4.6.1 Fehlercodes der RS232-Schnittstelle.....	61
7.4.7 Technische Details zur USB-Schnittstelle.....	62
7.4.8 Technische Details zu den Ausgängen.....	62
7.4.8.1 Analogausgang 1-2 (OUT, GND).....	62
7.4.8.2 Relais 1-6.....	63
7.4.9 Technische Details zum integrierten Pufferakku.....	63
7.4.10 Technische Details zur Energieversorgung.....	66
Kapitel 8 Inbetriebnahme.....	67
8.1 Hinweise an den Benutzer.....	67
8.2 Mitgeltende Unterlagen.....	67
8.3 Allgemeine Grundsätze.....	67
8.4 Inbetriebnahme des Systems.....	67
8.5 Kommunikation mit dem Gerät testen.....	68
Kapitel 9 Benutzerschnittstellen.....	71
9.1 Benutzerschnittstelle am myDatalogMUC xG/4G.....	71
9.1.1 Bedienelemente.....	71
9.1.1.1 Taste zum Auslösen des Aloha-Übertragungsmodus.....	72
9.1.1.2 Status-LED.....	72
9.1.1.3 Statusanzeige: Selbsttests.....	73
9.1.1.4 Statusanzeige: Pufferakku aktiv.....	74
9.1.1.5 Taste zum Auslösen des Selbsttests.....	74
9.1.1.6 Statusanzeige: Schaltzustände der Relais.....	74
9.2 Benutzerschnittstelle am myDatatnet-Server.....	74
9.2.1 Messstellenkonfiguration.....	74
9.2.1.1 Site.....	75
9.2.1.2 Kommentar.....	75
9.2.1.3 Steuerung.....	75

9.2.1.4 Schnittstellen.....	76
9.2.1.4.1 Basis.....	76
9.2.1.4.1.1 Com1 und Com3 (Modbus-Master, RS485, Device Logic Parsing inaktiv).....	76
9.2.1.4.1.2 Com1 und Com3 (Modbus-Slave, RS485, Device Logic Parsing inaktiv).....	78
9.2.1.4.1.3 Com1 und Com3 (RS485, Device Logic Parsing aktiv).....	79
9.2.1.4.1.4 Com2 (Seriell, RS232).....	82
9.2.1.4.2 Konfig.....	85
9.2.1.4.2.1 Com1 und Com3.....	85
9.2.1.4.2.2 Com2.....	86
9.2.1.5 Sequenzen für COM2.....	86
9.2.1.6 Messkanäle.....	87
9.2.1.6.1 Basis.....	87
9.2.1.6.2 Konfig.....	90
9.2.1.6.3 Alarmer.....	104
9.2.1.6.4 Trigger.....	105
9.2.1.7 Interface Kanäle 1-32.....	107
9.2.1.7.1 Basis.....	108
9.2.1.7.2 Konfig.....	109
9.2.1.7.2.1 Com1 und Com3 (Modbus-Master, RS485).....	110
9.2.1.7.2.2 Com1 und Com3 (Modbus-Slave, RS485).....	111
9.2.1.7.2.3 Com2 (Seriell, RS232).....	112
9.2.1.7.3 Skalierung.....	112
9.2.1.7.3.1 Com1 und Com3 (Modbus, RS485).....	112
9.2.1.7.3.2 Com2 (Seriell, RS232).....	113
9.2.1.7.4 Alarmer.....	113
9.2.1.7.5 Trigger.....	114
9.2.1.8 Interface Kanäle 33-64.....	115
9.2.1.8.1 Basis.....	115
9.2.1.9 Interface Ausgabekanäle 1-32.....	115
9.2.1.9.1 Basis.....	116
9.2.1.9.2 Konfig.....	117
9.2.1.9.2.1 Com1 und Com3 (Modbus-Master, RS485).....	117
9.2.1.9.2.2 Com1 und Com3 (Modbus-Slave, RS485).....	119

9.2.1.9.2.3 Com2 (Seriell, RS232).....	120
9.2.1.9.2.4 Device Logic.....	120
9.2.1.9.3 Skalierung.....	120
9.2.1.9.3.1 Com1 und Com3 (Modbus, RS485).....	120
9.2.1.9.3.2 Com2 (Seriell, RS232).....	121
9.2.1.9.3.3 Device Logic.....	121
9.2.1.10 Interface Ausgabekanäle 33-64.....	121
9.2.1.11 Berechnete Kanäle.....	121
9.2.1.11.1 Basis.....	121
9.2.1.11.2 Berechnung.....	123
9.2.1.11.3 Alarme.....	125
9.2.1.12 Ausgabekanäle.....	125
9.2.1.12.1 Basis.....	125
9.2.1.13 Interne Kanäle.....	130
9.2.1.13.1 Basis.....	130
9.2.1.13.2 Alarme.....	130
9.2.1.13.3 Trigger.....	131
9.2.1.14 Alarmierung.....	132
9.2.1.15 Grundeinstellung.....	133
9.2.1.16 FTP-Export Einstellungen.....	135
9.2.2 Gerätekonfiguration.....	135
9.2.2.1 Kommentar.....	135
9.2.2.2 Messgerät.....	136
9.2.2.3 Gerätespezifische Einstellungen.....	137
9.2.2.4 GPRS.....	137
Kapitel 10 myDatenet-Server.....	139
10.1 Übersicht.....	139
10.1.1 Erklärung der Symbole.....	139
10.2 Bereich "Kunden".....	140
10.3 Bereich "Messstellen" auf Kundenebene.....	142
10.3.1 Auswertungen.....	143
10.3.2 Kartendarstellung.....	143
10.4 Empfohlene Vorgehensweise.....	143

10.4.1 Anlegen der Messstelle.....	143
Kapitel 11 Device Logic (Pawn).....	147
11.1 Allgemein.....	147
11.1.1 Direkte Eingabe einer Pawn Device Logic.....	147
11.1.2 Hochladen eines Binary-Files.....	147
11.2 Compiler-Optionen.....	147
11.3 Device API.....	148
11.3.1 Konstanten.....	148
11.3.2 System.....	148
11.3.3 Datum & Zeit.....	148
11.3.4 Encoding.....	150
11.3.5 Messkanäle.....	152
11.3.5.1 Konstanten.....	152
11.3.5.2 Funktionen.....	154
11.3.6 Serielle Schnittstellen.....	158
11.3.6.1 Konstanten.....	158
11.3.6.2 Callback Funktionen.....	159
11.3.6.3 Funktionen.....	159
11.3.7 Alarm & Trigger.....	160
11.3.7.1 Arrays mit symbolischen Indizes.....	160
11.3.7.2 Konstanten.....	161
11.3.7.3 Funktionen.....	162
11.3.8 Mathematik.....	164
11.3.9 Char & String.....	167
11.3.10 Verschiedene Funktionen.....	172
11.3.10.1 Arrays mit symbolischen Indizes.....	172
11.3.10.2 Konstanten.....	172
11.3.10.3 Funktionen.....	173
11.3.11 Consolen Funktionen.....	176
11.4 Vordefinierte Log-Einträge.....	177
11.5 Device Logic Fehlercodes.....	178
11.6 Syntax.....	179
11.6.1 Allgemeine Syntax.....	179

11.6.1.1	Format	179
11.6.1.2	Optionale Semikolons	179
11.6.1.3	Kommentare	180
11.6.1.4	Bezeichner	180
11.6.1.5	Reservierte Schlüsselworte	180
11.6.1.6	Numerische Konstanten	180
11.6.1.6.1	Numerische Integer-Konstanten	180
11.6.1.6.2	Numerische Gleitkomma-Konstanten	181
11.6.2	Variablen	181
11.6.2.1	Deklaration	181
11.6.2.2	Lokale Deklaration	181
11.6.2.3	Globale Deklaration	181
11.6.2.4	Statische lokale Deklaration	181
11.6.2.5	Statische globale Deklaration	182
11.6.2.6	Gleitkommawerte	182
11.6.3	Konstante Variablen	182
11.6.4	Array Variablen	182
11.6.4.1	Eindimensionales Array	182
11.6.4.2	Initialisierung	182
11.6.4.3	Progressive Initialisierung für Arrays	183
11.6.4.4	Mehrdimensionale Arrays	183
11.6.4.5	Arrays und der "sizeof"-Operator	184
11.6.5	Operatoren und Ausdrücke	184
11.6.5.1	Zeichenerklärung	184
11.6.5.2	Ausdrücke	185
11.6.5.3	Arithmetik	185
11.6.5.4	Bit-Manipulation	185
11.6.5.5	Zuweisung	186
11.6.5.6	Vergleichsoperatoren	186
11.6.5.7	Boolean	187
11.6.5.8	Sonstiges	187
11.6.5.9	Priorität der Operatoren	188
11.6.6	Anweisungen	189

11.6.6.1 Statement-Etikett	189
11.6.6.2 Zusammengesetzte Anweisungen.....	189
11.6.6.3 Ausdrucksanweisung.....	189
11.6.6.4 Leeres Statement	189
11.6.6.5 assert Ausdruck	190
11.6.6.6 break	190
11.6.6.7 continue.....	191
11.6.6.8 do Statement while (Ausdruck).....	191
11.6.6.9 exit Ausdruck	191
11.6.6.10 for (Ausdruck 1 ; Ausdruck 2 ; Ausdruck 3) Statement	192
11.6.6.11 goto Etikett	192
11.6.6.12 if (Ausdruck) Statement 1 else Statement 2.....	193
11.6.6.13 return Ausdruck	193
11.6.6.14 switch (Ausdruck) { case Liste }.....	193
11.6.6.15 while (Ausdruck) Statement	194
11.6.7 Funktionen.....	194
11.6.7.1 Funktionsargumente ("call-by-value" versus "call-by-reference").....	195
11.6.7.2 Benannte Parameter versus positionsgebundene Parameter.....	197
11.6.7.3 Standardwerte von Funktionsargumenten.....	197
11.7 Beispiele.....	199
11.7.1 Sägezahngenerator.....	199
11.7.2 Wochentagbestimmung.....	200
11.7.3 Durchflussberechnung mit Tabelle.....	202
11.7.4 Script Parsing.....	203
11.8 Unterschiede zu C.....	204
Kapitel 12 API.....	207
12.1 Allgemein.....	207
12.2 rapidM2M Playground.....	207
12.2.1 Übersicht.....	208
Kapitel 13 Wartung.....	209
13.1 Allgemeine Wartung.....	209
13.2 Sicherungswechsel.....	209
Kapitel 14 Demontage/Entsorgung.....	211

Kapitel 15 Fehlersuche und Behebung	213
15.1 Allgemeine Probleme.....	213
15.2 Log-Einträge und Fehlercodes.....	216
15.2.1 Modemfehler.....	221
15.3 Auswerten des Gerätelogs.....	224
15.3.1 Auswerten des Gerätelogs am myDatenet-Server.....	224
15.3.2 Auswerten des Gerätelogs mittels DeviceConfig.....	224
Kapitel 16 Ersatzteile und Zubehör	225
16.1 Antennen.....	225
16.2 Versorgung.....	225
16.3 Adapter.....	225
16.4 Sonstiges Zubehör.....	225
Kapitel 17 Dokumentenhistorie	227
Kapitel 18 Glossar	229
Kapitel 19 Kontaktinformationen	231

Kapitel 2 Konformitätserklärung

2.1 myDatalogMUC 2G/4G EU

EU-Konformitätserklärung

EU Declaration of Conformity / Déclaration de conformité UE

Produktbezeichnung: Stationäres Instrument zur Aufzeichnung und Übertragung von
Product: Messdaten und Ausführung verschiedenster Steuer- und
Désignation du produit: Regelaufgaben

Type : myDatalogMUC 2G/4G EU
Type code:
Type:



Hersteller: Microtronics Engineering GmbH
Manufacturer : Hauptstrasse 7
Fabricant: A-3244 Ruprechtshofen

Das bezeichnete Produkt stimmt mit den folgenden Europäischen Richtlinien überein.
The designated product is in conformity with the following european directives.
Le produit décrit est conforme aux directives européennes suivantes.

		Europäische Norm	Ausgabedatum
(2014/30/EU)	EMC Directive	ÖVE/ÖNORM EN61326-1	2013-08
(2014/35/EU)	LVD Directive	ÖVE/ÖNORM EN61010-1	2011-08
(2014/53/EU)	RED Directive	Safety & Health 3.1a	ÖVE/ÖNORM EN60950-1 2008-11 ÖVE/ÖNORM EN62311 2016-06 ÖVE/ÖNORM EN62368-1
		EMC 3.1b	ÖVE/ÖNORM EN301489-1 V2.1.1 2017-04 Draft ÖVE/ÖNORM EN301489-52 2017-01 V1.1.0
		Radio spectrum efficiency 3.2	ÖVE/ÖNORM EN301511 V12.5.1 2017-06 ÖVE/ÖNORM EN301908-1 V11.1.1 2016-10 ÖVE/ÖNORM EN301908-13 V11.1.2 2017-10
(2015/863/EU)	RoHS Directive	Prevention 4.1	ÖVE EN IEC 63000 2019-06

Ruprechtshofen, den 28.08.2020

Ort und Datum der Ausstellung
Place and date of issue
Lieu et date d'établissement

Hans-Peter Buber, Managing Director
Unterschrift
name and signature of authorised person
Nom et signature de la personne autorisée

Kapitel 3 Technische Daten

Spannungsversorgung	12...30VDC (+/-10%) Zusätzliche Informationen finden Sie unter "Technische Details zur Energieversorgung" auf Seite 66.
Leistungsaufnahme	typ. 1W (ohne Sensoren) max. 3W (ohne Sensoren)
Integrierter Pufferakku	LiPo-Akku mit 300mAh zum Absetzen einer Meldung beim Ausfall der Versorgung. Zusätzliche Informationen finden Sie unter "Technische Details zum integrierten Pufferakku" auf Seite 63.
Gehäuse	Material: Lexan/Noryl, UL94-V0 Gewicht: 320g Schutzart: IP20 Abmessungen (BHT): 157 x 88 x 64mm (ohne Antenne)
Betriebstemperatur	-20...+60°C
Luftfeuchtigkeit	15...90%rH nicht kondensierend
Lagertemperatur	-20...+60°C
Ladetemperatur (Pufferakku)	0...+40°C
Anzeige	LED zur Signalisierung des Betriebszustandes und der Fehlercodes LED zur Anzeige des Ergebnisses des Selbsttests LED zur Signalisierung, dass das Gerät vom Pufferakku versorgt wird LEDs zur Anzeige der Schaltzustände der Relais
Bedienung	Taste zum Auslösen des Aloha-Übertragungsmodus Taste zum Auslösen des Selbsttests Taste zum Auslösen eines Resets
Antennenanschluss	2 x SMA
Universaleingänge	8 x analog oder digital Modi: <ul style="list-style-type: none"> • 0...20mA: Auflösung 6,3µA, max. 23,7mA, Bürde 96Ω • 4...20mA: Auflösung 6,3µA, max. 23,7mA, Bürde 96Ω • 0...2V: Auflösung 610µV, max. 2,5V, Bürde 10k086 • 0...10V: Auflösung 7,97mV, max. 32V, Bürde 4k7 • PWM: 1...99%, max. 100Hz, Impulslänge min. 1ms, max. 32V, Bürde 4k7 • Frequenz: 1...1000Hz, max. 32V, 4k7 • Digital: low <1,31V, high >2,61V, max. 32V, Bürde 4k7 • Tageszähler: Impulslänge min. 1ms, max. 32V, Bürde 4k7 • Intervallzähler: Impulslänge min. 1ms, max. 32V, Bürde 4k7 Zusätzliche Informationen finden Sie unter "Technische Details zu den Universaleingängen" auf Seite 57.

<p>Externer Temperatursensor</p>	<p>1 x PT100/1000 (mit Autoerkennung)</p> <p>Zusätzliche Informationen finden Sie unter "Technische Details zur PT100/1000-Schnittstelle" auf Seite 58.</p>
<p>Modbus</p>	<p>2 x RS485 umschaltbar zwischen Modbus-Master und Modbus-Slave</p> <ul style="list-style-type: none"> • Baudrate: 300-115200 • Stopbits: 1, 2 • Parität: N, E, O • Datenbits: 7, 8 <p>Modi:</p> <ul style="list-style-type: none"> • RTU • ASCII <p>Anzahl der verarbeitbaren Interface Kanäle</p> <ul style="list-style-type: none"> • Eingänge: Modbuskanäle + serielle Kanäle = 64 • Ausgänge: Modbuskanäle + serielle Kanäle = 64 <p>Zusätzliche Informationen finden Sie unter "Technische Details zu den Modbus-Schnittstellen (Com1, Com3)" auf Seite 58.</p>
<p>Serielle Schnittstelle</p>	<p>1 x RS232 für den Anschluss eines digitalen Sensors</p> <ul style="list-style-type: none"> • Baudrate: 300-115200 • Stopbits: 1, 2 • Parität: N, E, O • Datenbits: 7, 8 <p>Modi:</p> <ul style="list-style-type: none"> • ASCII <p>Anzahl der verarbeitbaren Interface Kanäle</p> <ul style="list-style-type: none"> • Eingänge: Modbuskanäle + serielle Kanäle = 64 • Ausgänge: Modbuskanäle + serielle Kanäle = 64 <p>Zusätzliche Informationen finden Sie unter "Technische Details zur RS232-Schnittstelle (Com2)" auf Seite 60.</p>

Ausgänge	<p>6 x Relais (2 Gruppen für unterschiedliche Potentiale)</p> <ul style="list-style-type: none"> • U: 240VAC • U_{\max}: 400VAC • I_{\max}: 3A <p>2 x Analogausgang (keine galvanische Trennung)</p> <ul style="list-style-type: none"> • Bürdenspannung: Entspricht der Versorgungsspannung (12...30VDC , +/- 10%) • max. Bürde (32V, 20mA): 1200Ω • $I_{\text{out max}}$: 20mA <p>Modi:</p> <ul style="list-style-type: none"> • 4-20mA • 0-20mA <p>Zusätzliche Informationen finden Sie unter "Technische Details zu den Ausgängen" auf Seite 62.</p>
USB-Schnittstelle	<p>1 x Micro-B USB 2.0 Slave für die Verbindung mit einem PC. Für die Kommunikation mit dem myDatalogMUC xG/4G muss am PC das Konfigurationsprogramm DeviceConfig installiert sein.</p> <p>Zusätzliche Informationen finden Sie unter "Technische Details zur USB-Schnittstelle" auf Seite 62.</p>
Datenspeicher	<p>Interner Flash-Speicher für bis zu 4.470 Messzyklen</p> <p>Zusätzliche Informationen finden Sie unter "Funktionsweise des internen Datenspeichers" auf Seite 40.</p>
Datentyp	<p>f32 (32Bit floating point)</p> <p>Ausnahme: Modbus Register (siehe Konfigurationsparameter "Format" im Kapitel "Interface Kanäle 1-32" auf Seite 107)</p>
Datenübertragung	<p>2G/4G Europa (myDatalogMUC 2G/4G EU)</p> <ul style="list-style-type: none"> • 2G GPRS 900MHz / 1800MHz • LTE CAT1 B3, B7, B20 <p>3G/4G US (myDatalogMUC 3G/4G US)</p> <ul style="list-style-type: none"> • UMTS B2, B5 • LTE CAT1 B2, B4, B5, B12
SIM	<p>Das myDatalogMUC xG/4G ist mit einem integrierten SIM-Chip versehen.</p>
Monatliches Datenvolumen	<p>2,0MB bei 2min. Messintervall und 120min. Übertragungsintervall (nur die 8 Universaleingänge sind aktiv)</p> <p>Zusätzliche Informationen finden Sie unter "Grundeinstellung" auf Seite 133.</p>

Kapitel 4 Allgemeine Angaben

Die Informationen dieses Handbuchs wurden sorgfältig geprüft und nach bestem Wissen zusammengestellt. Der Hersteller übernimmt dennoch keine Verantwortung für möglicherweise in diesem Handbuch enthaltene falsche Angaben. Der Hersteller ist nicht verantwortlich für direkte, indirekte, versehentliche oder Folgeschäden, die aus Fehlern oder Unterlassungen in diesem Handbuch entstanden, selbst wenn auf die Möglichkeit solcher Schäden hingewiesen wurde. Im Interesse der fortlaufenden Produktentwicklung behält sich der Hersteller jederzeit und ohne vorherige Ankündigung oder Verpflichtung das Recht auf Verbesserungen an diesem Handbuch und der hierin beschriebenen Produkte vor.

***Hinweis:** Die Angaben dieses Handbuches sind ab den auf der Titelseite angeführten Versionsständen gültig. Überarbeitete Ausgaben dieses Handbuchs sowie Software und Treiber-Updates sind im Servicebereich des myDatanet-Servers erhältlich.*

4.1 Übersetzung

Bei Lieferungen in die Länder des europäischen Wirtschaftsraumes ist das Handbuch in die Sprache des Verwenderlandes zu übersetzen. Sollten im übersetzten Text Unstimmigkeiten auftreten, ist das Original-Handbuch (deutsch) zur Klärung heranzuziehen oder der Hersteller zu kontaktieren.

4.2 Copyright

Weitergabe, Vervielfältigung dieses Dokuments sowie Verwertung und Mitteilung seines Inhalts sind verboten, soweit nicht ausdrücklich gestattet. Zuwiderhandlungen verpflichten zu Schadenersatz. Alle Rechte vorbehalten.

4.3 Gebrauchsnamen

Die Wiedergabe von Gebrauchsnamen, Handelsnamen, Warenbezeichnungen und dgl. in diesem Handbuch berechtigen nicht zu der Annahme, dass solche Namen ohne weiteres von jedermann benutzt werden dürfen; oft handelt es sich um gesetzlich geschützte eingetragene Warenzeichen, auch wenn sie nicht als solche gekennzeichnet sind.

4.4 Sicherheitshinweise

Für Anschluss, Inbetriebnahme und Betrieb des myDatalogMUC xG/4G sind die nachfolgenden Informationen und übergeordneten gesetzlichen Bestimmungen des Landes (z.B. ÖVE), wie gültigen Ex-Vorschriften sowie die für den jeweiligen Einzelfall geltenden Sicherheits- und Unfallverhütungsvorschriften zu beachten.

Lesen Sie dieses Handbuch komplett durch, bevor Sie dieses Gerät auspacken, aufstellen oder bedienen. Beachten Sie alle Gefahren-, Warn- und Vorsichtshinweise. Nichtbeachtung kann zu schweren Verletzungen des Bedieners oder Schäden am Gerät führen.

Stellen Sie sicher, dass die Sicherheitseinrichtung dieses Messgerätes nicht beeinträchtigt wird. Verwenden bzw. installieren Sie das Messsystem nur auf solche Art und Weise, wie sie in diesem Handbuch beschrieben wird.

Wichtiger Hinweis: Das Produkt ist nicht zur Nutzung im Freien freigegeben, da es keinen Schutz gegen das Eindringen von Feuchtigkeit und nur sehr geringen Schutz gegen das Eindringen von Staub besitzt.

4.4.1 Verwendung der Gefahrenhinweise

GEFAHR:



Kennzeichnet eine mögliche oder drohende Gefahrensituation, die den Tod oder eine ernsthafte Verletzung zur Folge haben wird, wenn sie nicht vermieden wird.

WARNUNG:



Kennzeichnet eine mögliche oder drohende Gefahrensituation, die den Tod oder eine ernsthafte Verletzung zur Folge haben kann, wenn sie nicht vermieden wird.

VORSICHT:



Kennzeichnet eine mögliche Gefahrensituation, die leichte oder mittelschwere Verletzungen oder Schäden an diesem Instrument zur Folge haben kann.

Wichtiger Hinweis: Kennzeichnet eine Situation, die Schäden an diesem Instrument zur Folge haben kann, wenn sie nicht vermieden wird. Informationen, die besonders hervorgehoben werden müssen.

Hinweis: Kennzeichnet eine Situation, die keine Personenschäden zur Folge hat.

Hinweis: Informationen, die Angaben im Haupttext ergänzen.

4.4.2 Allgemeine Sicherheitshinweise

WARNUNG:



Gefährliche elektrische Spannung kann zu elektrischem Schlag und Verbrennungen führen. Schalten Sie immer alle verwendeten Spannungsversorgungen für das Gerät ab, bevor Sie es installieren, Wartungsarbeiten durchführen oder Störungen beheben.

WARNUNG:



Verwenden Sie dieses Gerät nie in Bereichen, in denen der Betrieb von Funkeinrichtungen untersagt ist. Das Gerät darf nicht in Krankenhäusern und/oder in der Nähe von medizinischen Geräten, wie etwa Herzschrittmachern oder Hörgeräten, betrieben werden, da deren Funktionsweise durch das im Gerät enthaltene GSM/GPRS-Modem beeinträchtigt werden kann.

WARNUNG:



Verwenden Sie dieses Gerät nie in explosionsgefährdeten Bereichen sowie in der Nähe von hochbrennbaren Bereichen (Tankstellen, Brennstofflagerstätten, Chemiewerken und Sprengstätten) oder in der Nähe von brennbaren Gasen, Dämpfen oder Staub.

4.4.3 Sicherheits-/Vorsichtsmaßnahmen im Umgang mit GSM/GPRS-Modems

Die folgenden Sicherheits-/Vorsichtsmaßnahmen sind bei allen Phasen des Einbaus, des Betriebs, der Wartung oder der Reparatur eines GSM/GPRS-Modems zu beachten. Der Hersteller haftet nicht, wenn der Kunde diese Vorsichtsmaßnahmen außer Acht lässt.

**VORSICHT:**

Die GSM/GPRS-Modemverbindung darf nicht in gefährlichen Umgebungen verwendet werden.

Der Hersteller und seine Lieferanten übernehmen weder ausdrückliche noch indirekte Garantie für die Verwendung bei Hochrisikoaktivitäten.

Zusätzlich zu den folgenden Sicherheitsbetrachtungen sind alle Richtlinien des Landes zu befolgen, in dem das Gerät installiert wird.

Wichtiger Hinweis: Für die Verbindung mittels GSM/GPRS-Modem, bei dessen Verwendung Funksignale und -netzwerke zum Einsatz kommen, wird zu keiner Zeit und unter keinen Umständen gehaftet. Das GSM/GPRS-Modem muss eingeschaltet sein und in einem Gebiet betrieben werden, in dem eine ausreichende Signalstärke vorhanden ist.

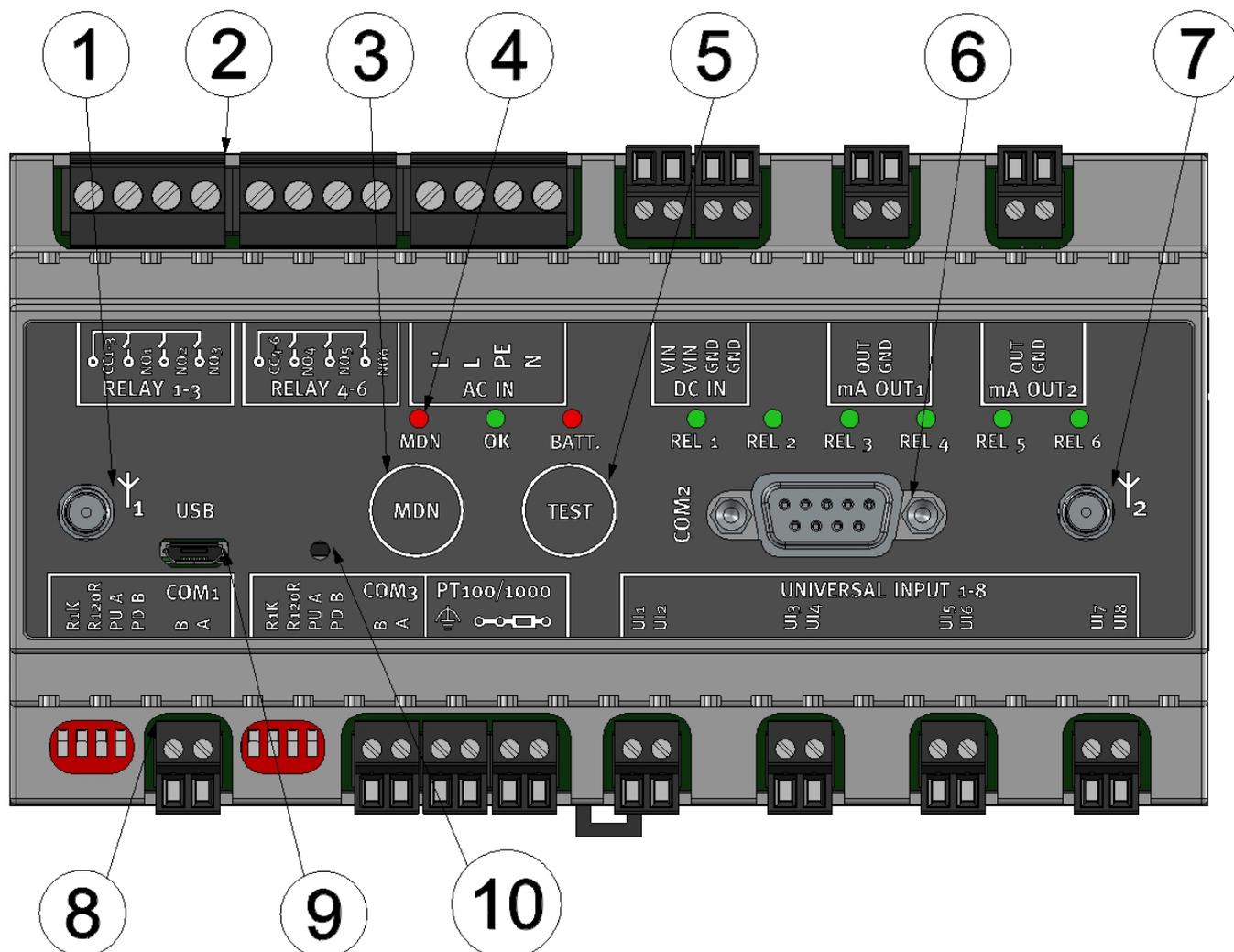
4.4.3.1 Sicherheits-/Vorsichtsmaßnahmen für den GSM/GPRS-Modemeinbau

- Dieses Gerät darf nur durch einen geschulten Techniker eingebaut werden, der anerkannte Einbaupraktiken für Funkfrequenzsender anwendet, einschließlich der korrekten Erdung von externen Antennen.
- Das Gerät darf nicht in Krankenhäusern und/oder in der Nähe von medizinischen Geräten, wie etwa Herzschrittmachern oder Hörgeräten, betrieben werden.
- Das Gerät darf nicht in der Nähe von hochbrennbaren Bereichen, wie etwa Tankstellen, Brennstofflagerstätten, Chemiewerken und Sprengstätten betrieben werden.
- Das Gerät darf nicht in der Nähe von brennbaren Gasen, Dämpfen oder Staub betrieben werden.
- Das Gerät darf weder starken Vibrationen noch Stößen ausgesetzt werden.
- Das GSM/GPRS-Modem kann Störungen verursachen, wenn es sich in der Nähe von Fernsehgeräten, Radios oder Computern befindet.
- Das GSM/GPRS-Modem nicht öffnen. Eine Änderung des Geräts ist unzulässig und führt zum Verlust der Betriebsgenehmigung.
- Die Nutzung von GSM-Diensten (SMS-Nachrichten, Datenkommunikation, GPRS, etc.) führt unter Umständen zu zusätzlichen Kosten. Der Benutzer ist allein verantwortlich für hierdurch erfolgte Schäden und Kosten.
- Bauen Sie das Gerät nicht anders ein, als in der Bedienungsanleitung angegeben. Eine fehlerhafte Verwendung führt zum Erlöschen der Garantie.

4.4.3.2 Sicherheitsmaßnahmen für den Antenneneinbau

- Nur Antennen verwenden, die vom Hersteller empfohlen oder geliefert werden.
- Die Antenne muss mindestens im Abstand von 20cm zu Personen aufgestellt werden.
- Die Antenne darf nicht außerhalb von geschützten Gebäuden aufsteigen und muss gegen Blitzschläge geschützt werden!
- Die Spannungsversorgung muss abgestellt werden, bevor eine Antenne ausgetauscht wird.

4.5 Übersicht

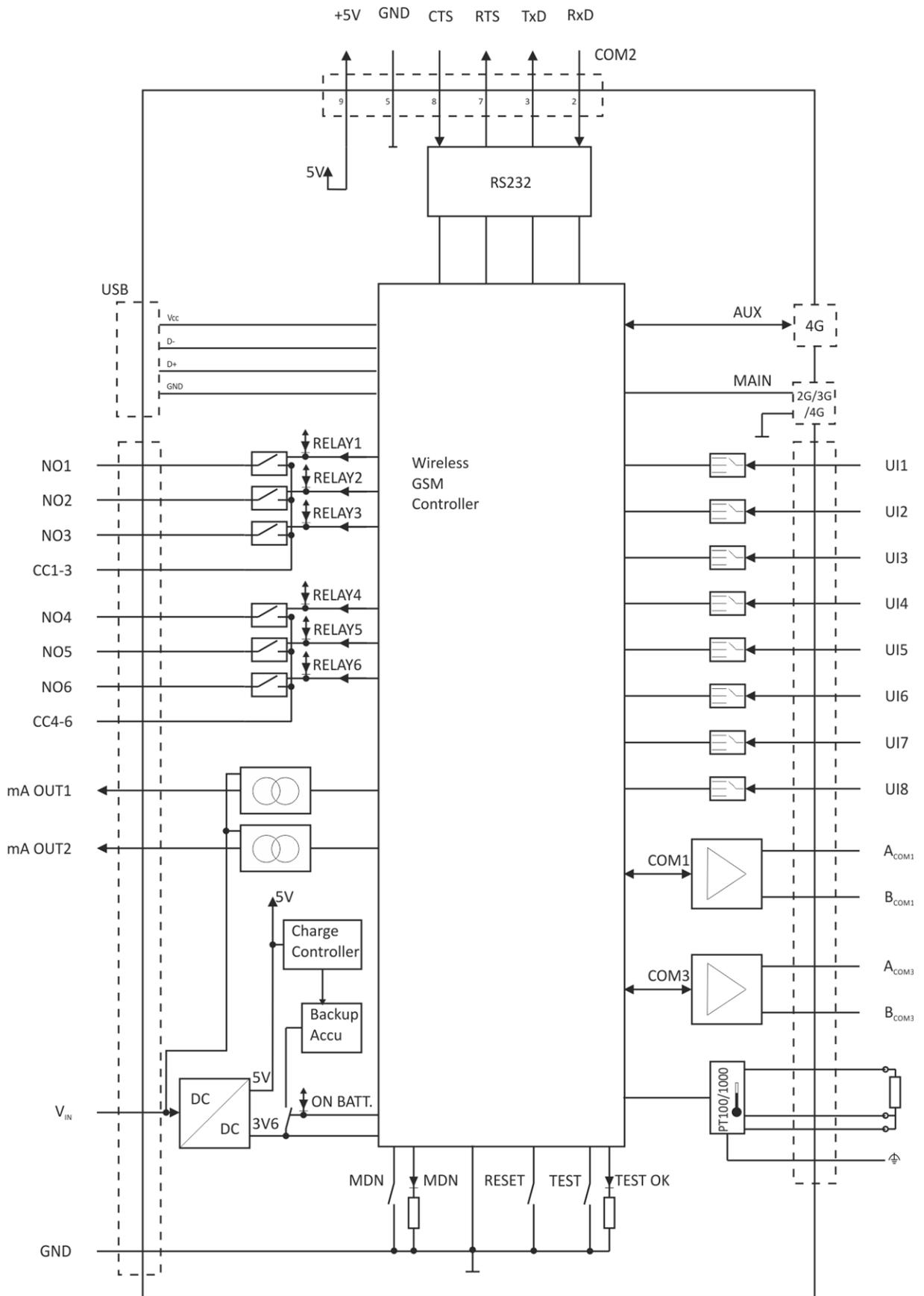


Übersicht myDatalogMUC xG/4G

1 Anschluss für die primäre Mobilfunkantenne (MAIN)	6 Sub-D Buchse der seriellen Schnittstelle (Com 2)
2 Obere Anschlussleiste	7 Anschluss für die sekundäre Mobilfunkantenne (AUX)
3 Taste zum Auslösen des Aloha-Übertragungsmodus	8 Untere Anschlussleiste
4 Signalisierungs-LEDs	9 Micro-B USB (nur für Debug)
5 Taste zum Auslösen des Selbsttests	10 Taste zum Auslösen eines Resets

4.5.1 Blockschaltbild

Wichtiger Hinweis: Das Hardware-Handshake der RS232-Schnittstelle (CTS, RTS) wird von der aktuellen Firmware nicht unterstützt.



Blockschaltbild des myDatalogMUC xG/4G

4.6 Bestimmungsgemäße Verwendung

Das stationäre Messgerät bietet eine universelle Lösung für eine Vielzahl von Signalerfassungs- und Steueraufgaben. Das Gerät benötigt eine permanente Energieversorgung. Die Speicherung der gemessenen und erfassten Daten erfolgt auf einem nicht flüchtigen Speichermedium. Diese gespeicherten Daten werden über das Mobilfunknetz an einen zentralen Server zur Weiterverarbeitung gesendet. Dazu ist das Gerät mit einem integrierten SIM-Chip versehen. Es sind die zulässigen maximalen Grenzwerte, aufgeführt im Kapitel "Technische Daten" auf Seite 15, unbedingt zu beachten. Sämtliche von diesen Grenzwerten abweichende Einsatzfälle, die nicht vom Hersteller in schriftlicher Form freigegeben sind, entfallen aus der Haftung des Herstellers.

Hinweis: Das Gerät ist ausschließlich zum vorab angeführten Zweck bestimmt. Eine andere, darüber hinausgehende Benutzung oder ein Umbau des Geräts ohne schriftliche Absprache mit dem Hersteller gilt als nicht bestimmungsgemäß. Für jegliche hieraus resultierende Schäden und daraus resultierende Folgeschäden haftet der Hersteller nicht. Das Risiko trägt allein der Betreiber.

Hinweis: Der Hersteller haftet nicht für Datenverluste jeglicher Art.

Hinweis: Der integrierte SIM-Chip gewährleistet eine Mobilfunkverbindung über eine Vielzahl internationaler Serviceprovider. Um alle Funktionen des Geräts nutzen zu können, muss gewährleistet sein, dass es sich im Versorgungsbereich eines dieser Anbieter befindet. Eine Liste aller unterstützten Länder und dazugehörige Serviceprovider finden Sie unter www.microtronics.com/footprint. Für die Nutzung der mobilen Datenübertragung ist ein "Managed Service"-Vertrag mit der Firma Microtronics Engineering GmbH erforderlich (siehe www.microtronics.com/managedservice). Dieser beinhaltet die Bereitstellung der Mobilfunkverbindung über die Netze der in der oben genannten Liste enthaltenen Serviceprovider.

4.7 Allgemeine Produktinformationen

Es handelt sich um ein stationäres Gerät, das neben der Aufzeichnung und Übertragung von Messdaten aus unterschiedlichsten Quellen, auch noch für verschiedenste Steuer- und Regelaufgaben verwendet werden kann.

Für die Messdatenerfassung stehen folgende Schnittstellen zur Verfügung:

- 8 Universaleingänge, die in verschiedenen Analog- oder Digitalmodi betrieben werden können
- Eine Schnittstelle für den Anschluss eines PT100 oder PT1000 mit automatischer Erkennung, welcher der beiden Typen verwendet wird
- 2 RS485 Schnittstellen, die entweder als Modbus-Master oder Modbus-Slave konfiguriert werden können
- Eine RS232 Schnittstelle, an die ein digitaler Sensor angeschlossen werden kann, der seine Daten blockweise durch ein Trennzeichen separiert im ASCII-Format liefert

Es gilt zu beachten, dass für die Aufzeichnung der von den 3 seriellen Schnittstellen (2 x RS485, 1 x RS232) gelesenen Messdaten nur insgesamt 64 Kanäle zur Verfügung stehen. Über diese 3 seriellen Schnittstellen ist es auch möglich Daten zurück in die Sensoren oder Aktoren zu liefern. Dazu stehen insgesamt 64 Kanäle zur Verfügung. Die auszugebenden Daten können entweder über die Eingabemaske des myDatanet-Servers festgelegt werden oder mittels Device Logic (siehe "Device Logic (Pawn)" auf Seite 147) vom myDatalogMUC xG/4G nach jedem Messzyklus berechnet werden.

Des Weiteren stehen für die Ausgabe von Regel- und Steuerbefehlen neben 2 0/4-20mA Analogausgängen, auch 6 Relais zur Verfügung, von denen je 3 eine gemeinsame Wurzel (siehe "Technische Details zu den

Ausgängen" auf Seite 62) besitzen. Der Ausgabewert der Analogausgänge kann drahtlos von einer zentralen Stelle aus vorgegeben oder mittels Device Logic (siehe "Device Logic (Pawn)" auf Seite 147) vom myDatalogMUC xG/4G selbst berechnet werden. Die Relais können so konfiguriert werden, dass sie vom Gerät selbst jeweils vor einer Messung geschaltet werden (zum Versorgen eines Sensors) oder, dass sie drahtlos von einer zentralen Stelle aus umgeschaltet werden können. Es ist auch möglich, die Stellwerte für die Relais vom myDatalogMUC xG/4G selbst mittels Device Logic ermitteln zu lassen.

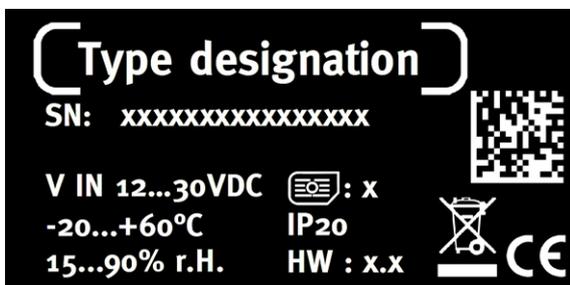
Die Messdaten der Eingänge werden zusammen mit den Zuständen der Ausgänge in einem internen Datenspeicher zwischengespeichert und in einem frei wählbaren Intervall drahtlos an eine zentrale Stelle übermittelt. Über dieselbe Verbindung erfolgt auch die Konfiguration des Geräts. Das Gerät ist mit einem integrierten SIM-Chip versehen.

4.8 Gerätekenzeichnung

Die Angaben in diesem Handbuch gelten ausschließlich für den Gerätetyp myDatalogMUC xG/4G . Das Typenschild befindet sich auf der rechten Seite des Geräts und beinhaltet folgende Angaben:

- Typenbezeichnung
- Seriennummer
- Angaben zur Spannungsversorgung
- Umgebungsbedingungen im Betrieb
- Länderlisten-Profil des SIM-Chips
- Schutzart
- Hardwarerevision
- Logo zur WEEE-Direktive der EU
- CE-Kennzeichnung

Wichtig für alle Rückfragen und Ersatzteilbestellungen ist die richtige Angabe der Typenbezeichnung und der Seriennummer. Nur so ist eine einwandfreie und schnelle Bearbeitung möglich.



Typenschild myDatalogMUC xG/4G



Hinweis: Dieses Symbol gibt das Länderlisten-Profil (siehe www.microtronics.com/footprint) des im Gerät verbauten SIM-Chips an.

Hinweis: Diese Betriebsanleitung ist Bestandteil des Gerätes und muss für den Benutzer jederzeit zur Verfügung stehen. Die darin enthaltenen Sicherheitshinweise sind zu beachten.



WARNUNG:

Es ist strengstens untersagt, die Sicherheitseinrichtungen außer Kraft zu setzen oder in ihrer Wirkungsweise zu verändern.

4.9 Einbau von Ersatz- und Verschleißteilen

Es wird an dieser Stelle ausdrücklich darauf aufmerksam gemacht, dass Ersatz- und Zubehörteile, die nicht vom Hersteller geliefert wurden, auch nicht vom Hersteller geprüft und freigegeben wurden. Der Einbau und/oder die Verwendung solcher Produkte können u. U. konstruktiv vorgegebene Eigenschaften des Geräts negativ verändern. Für sämtliche Schäden, die durch die Verwendung von Nicht-Originalteilen und Nicht-Original-Zubehörteilen entstehen, ist die Haftung des Herstellers ausgeschlossen.

4.10 Aufbewahrung des Produkts

Zur Aufbewahrung des myDatalogMUC xG/4G stellen Sie sicher, dass alle relevanten Daten zum myDatanet-Server übertragen wurden. Gegebenenfalls lösen Sie mittels Taste den Aloha-Übertragungsmodus (siehe "Aloha-Übertragungsmodus" auf Seite 41) aus und kontrollieren Sie anschließend erneut, ob nun alle relevanten Daten übertragen wurden. Als nächstes trennen Sie das Gerät von der Versorgungsspannung. Wenn möglich, sollte dabei die Versorgungsspannung zuerst abgeschaltet werden bevor Sie die Kabel an den Klemmen V_{IN} und GND (siehe "Anschluss der Sensoren, der Aktoren und der Versorgung" auf Seite 52) lösen. Anschließend können die restliche Verkabelung und die Antenne entfernt werden. Bewahren Sie das myDatalogMUC xG/4G in der Originalverpackung auf.

Die Konfiguration und die zuletzt ermittelten Daten bleiben erhalten. Die Systemzeit läuft allerdings ohne Energieversorgung nicht weiter. Das bedeutet, dass nach der Wiederinbetriebnahme eine Zeitsynchronisation erforderlich ist, damit die Zeitstempel der Mess- und Log-Daten wieder korrekt sind. Diese erfolgt automatisch bei der ersten Verbindung zum myDatanet-Server.

4.11 Gewährleistung

Das Gerät wurde vor Auslieferung funktional geprüft. Bei bestimmungsgemäßer Verwendung (siehe "Bestimmungsgemäße Verwendung" auf Seite 24) und Beachtung der Bedienungsanleitung, der mitgeltenden Unterlagen (siehe "Mitgeltende Unterlagen" auf Seite 67) und der darin enthaltenen Sicherheitshinweise und Anweisungen sind keine funktionalen Einschränkungen zu erwarten und ein einwandfreier Betrieb sollte möglich sein.

Hinweis: Beachten Sie hierzu auch das nachfolgende Kapitel "Haftungsausschluss" auf Seite 27.

Hinweis: Einschränkung der Gewährleistung

Bei Nichtbeachtung der Sicherheitshinweise und Anweisungen in dieser Unterlage behalten sich der Hersteller eine Einschränkung der Gewährleistung vor.

4.12 Haftungsausschluss

Der Hersteller übernimmt keine Haftung

- für Folgeschäden, die auf **eine Änderung** dieses Dokumentes zurückzuführen sind. Der Hersteller behält sich das Recht vor, den Inhalt des Dokuments einschließlich dieses Haftungsausschlusses unangekündigt zu ändern.
- für Personen- oder Sachschäden, die auf eine **Missachtung** der gültigen Vorschriften zurückzuführen sind. Für Anschluss, Inbetriebnahme und Betrieb der Geräte/Sensoren sind alle Informationen und übergeordneten gesetzlichen Bestimmungen des Landes (in Österreich z. B. die ÖVE-Richtlinien), wie gültige Ex-Vorschriften sowie die für den jeweiligen Einzelfall geltenden Sicherheits- und Unfallverhütungsvorschriften zu beachten.
- für Personen- oder Sachschäden, die auf **unsachgemäße Handhabung** zurückzuführen sind. Sämtliche Handhabungen am Gerät, welche über die montage- und anschlussbedingten Maßnahmen hinausgehen, dürfen aus Sicherheits- und Gewährleistungsgründen prinzipiell nur von Microtronics - Personal bzw. durch Microtronics autorisierte Personen oder Firmen vorgenommen werden.
- für Personen- oder Sachschäden, die auf den Betrieb des Geräts in technisch **nicht einwandfreiem** Zustand zurückzuführen sind.
- für Personen- oder Sachschäden, die auf eine **nicht bestimmungsgemäße Verwendung** zurückzuführen sind.
- für Personen- oder Sachschäden, die auf eine **Missachtung** der **Sicherheitshinweise** in dieser Anleitung zurückzuführen sind.
- für fehlende oder falsche Messwerte, die auf **unsachgemäße Installation** zurückzuführen sind und für die daraus resultierenden Folgeschäden.

4.13 Pflichten des Betreibers



WARNUNG:

Im EWR (Europäischer Wirtschaftsraum) sind die nationale Umsetzung der Rahmenrichtlinie (89/391/EWG) sowie die dazugehörigen Einzelrichtlinien und davon besonders die Richtlinie (2009/104/EG) über die Mindestvorschriften für Sicherheit und Gesundheitsschutz bei Benutzung von Arbeitsmitteln durch Arbeitnehmer bei der Arbeit, jeweils in der gültigen Fassung, zu beachten und einzuhalten.

Der Betreiber muss die örtliche Betriebserlaubnis einholen und die damit verbundenen Auflagen beachten.

Zusätzlich muss er die örtlichen gesetzlichen Bestimmungen für

- die Sicherheit des Personals (Unfallverhütungsvorschriften),
- die Sicherheit der Arbeitsmittel (Schutzausrüstung und Wartung),
- die Produktentsorgung (Abfallgesetz),
- die Materialentsorgung (Abfallgesetz),
- die Reinigung (Reinigungsmittel und Entsorgung) und
- die Umweltschutzauflagen einhalten.

Vor dem Betreiben des Messgeräts ist vom Betreiber sicherzustellen, dass bei der Montage und Inbetriebnahme, wenn diese vom Betreiber selbst durchgeführt werden, die örtlichen Vorschriften beachtet werden.

4.14 Anforderungen an das Personal

Die Installation, Inbetriebnahme und Wartung dürfen nur durch Personal durchgeführt werden, das die folgenden Bedingungen erfüllt:

- Qualifiziertes Fachpersonal mit entsprechender Ausbildung
- Autorisierung durch den Anlagenbetreiber

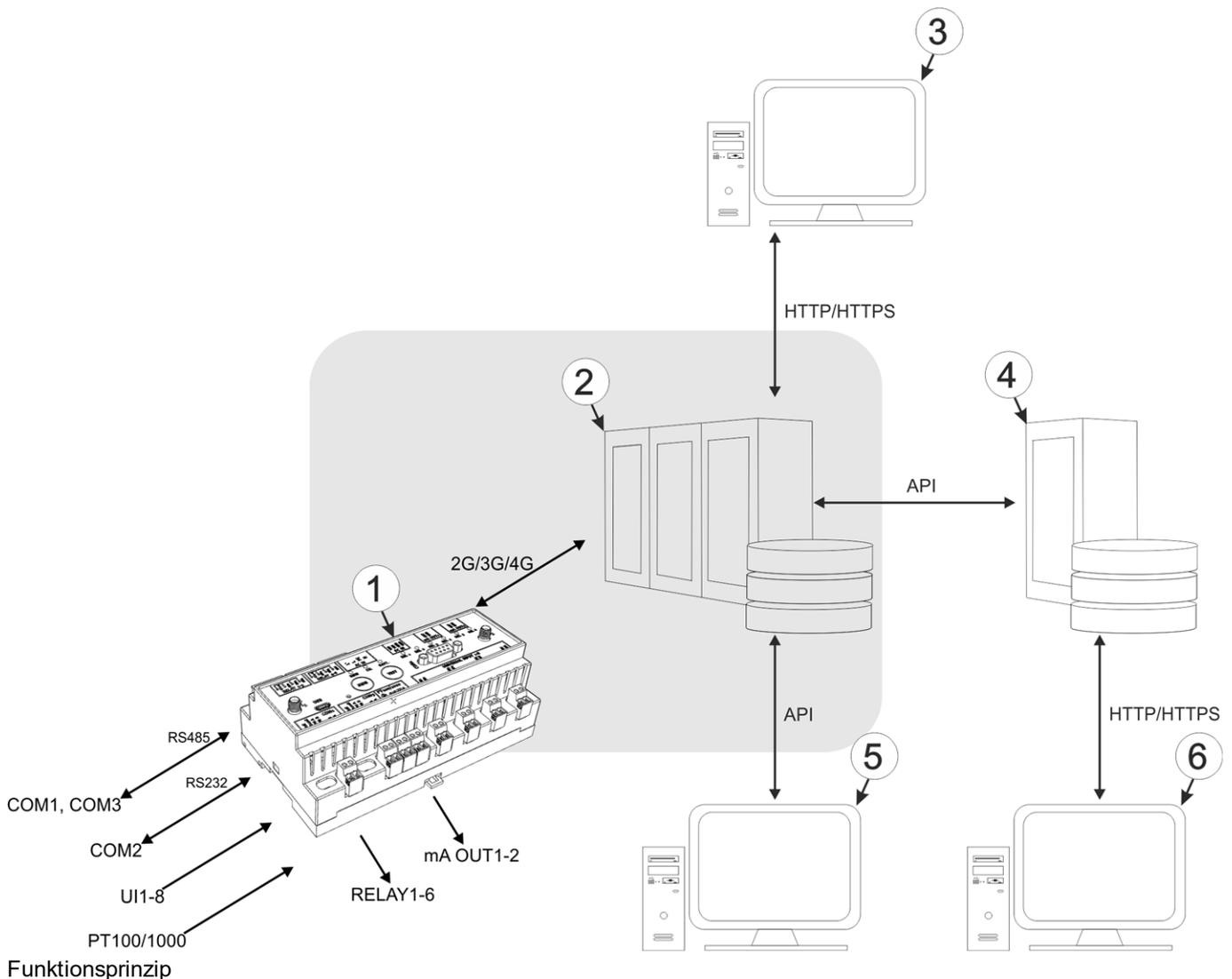
Hinweis: Qualifiziertes Fachpersonal

Im Sinne dieser Anleitung bzw. Warnhinweise auf dem Produkt selbst sind dies Personen, die mit Aufstellung, Montage, Inbetriebnahme und Betrieb des Produktes vertraut sind und über die ihrer Tätigkeit entsprechenden Qualifikationen verfügen, wie z.B.

- *Ausbildung und Unterweisung bzw. Berechtigung, Stromkreise und Geräte/Systeme gemäß den Standards der Sicherheitstechnik ein- und auszuschalten, zu erden und zu kennzeichnen*
- *Ausbildung oder Unterweisungen gemäß den Standards der Sicherheitstechnik in Pflege und Gebrauch angemessener Sicherheitsausrüstung*
- *Schulung in Erster Hilfe*

Kapitel 5 Funktionsprinzip

In der unten abgebildeten Grafik sind alle Komponenten, die Teil des myDatanet sind, grau hinterlegt. Alle anderen Komponenten müssen vom Kunden bereitgestellt/erstellt werden.



1	myDatalogMUC xG/4G mit integriertem Managed Service SIM-Chip (Datenübertragung inkludiert)
2	myDatanet-Server, zu dem die Daten übertragen werden
3	Client, der mittels Web-Browser auf die Oberfläche des myDatanet-Servers zugreift
4	kundenspezifischer Server, der den Clients eine eigene Oberfläche zur Verfügung stellt. Die Daten bezieht der kundenspezifische Server über die API des myDatanet-Servers (siehe "API" auf Seite 207).
5	Client, auf dem ein PC-Programm läuft, das seine Daten über die API des myDatanet-Servers (siehe "API" auf Seite 207) bezieht
6	Client, der mittels Web-Browser auf die Oberfläche des kundenspezifischen Servers zugreift

Funktionen und Komponenten, die durch myDatenet bereitgestellt werden:

- myDatalogMUC xG/4G

Das myDatalogMUC xG/4G ist ein stationäres Gerät für die Anbindung von Sensoren (UI1-8, PT100/1000, COM1-3) und Aktoren (RELAY1-6, mA OUT1-2, COM1-3) an den myDatenet-Server (2G/3G/4G).

- Managed Service

Das Managed Service ist die Basis für den Betrieb Ihrer Geräte und bietet Ihnen eine breite Palette an Services. Managed Service inkludiert Updates für Geräte-Firmware, mobile Datenübertragung auf globaler Ebene sowie gebührenfreien Support - ein Ansprechpartner für die gesamte Lösung.

- myDatenet-Server

Datenbank für die Speicherung der Messdaten und Konfigurationen. Der Zugriff auf die Daten erfolgt entweder über die API des Servers (siehe "API" auf Seite 207) oder über die Web-Oberfläche des Servers.

Funktionen und Komponenten, die durch den Kunden bereitgestellt werden

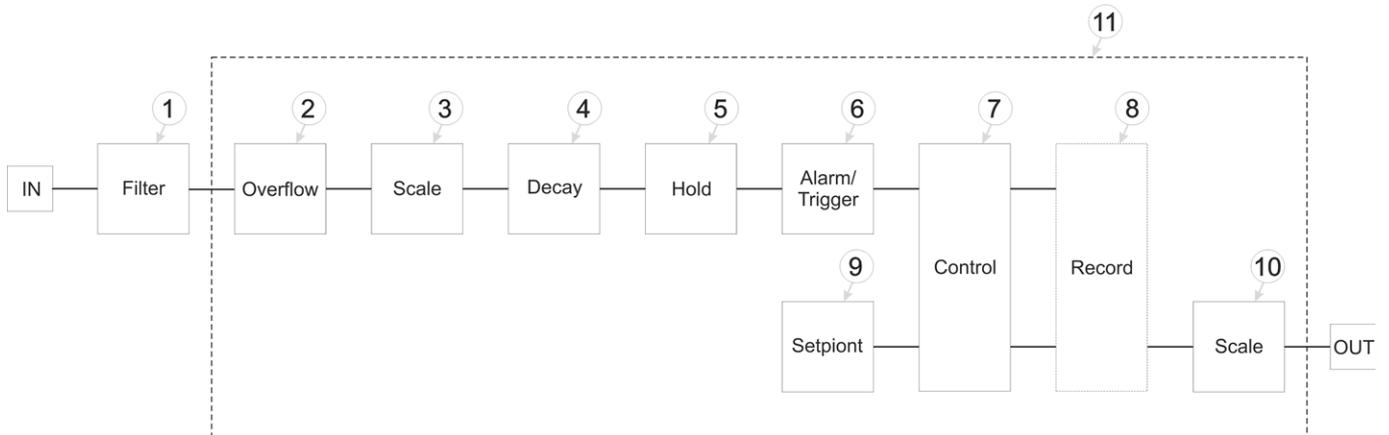
- Sensoren und Aktoren

Sensoren und Aktoren die über Schnittstellen verfügen, die mit dem im Kapitel "Technische Daten" (siehe "Technische Daten" auf Seite 15) aufgelisteten Spezifikationen kompatibel sind.

- Kundenspezifischer Server mit Web-Oberfläche für die Clients (optional)

Dadurch ist es möglich eine eigene Web-Oberfläche für die Clients zu erstellen. Die Daten werden dabei vom kundenspezifischen Server über die API (siehe "API" auf Seite 207) vom myDatenet-Server gelesen.

5.1 Interne Verarbeitung der Messwerte



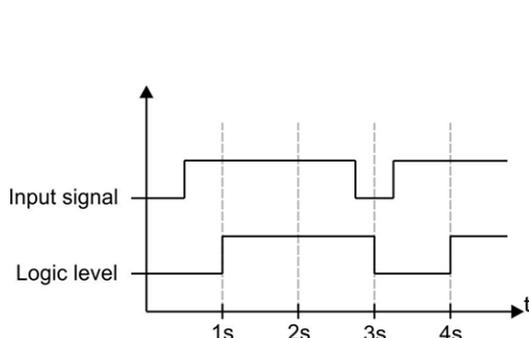
Schematische Darstellung der internen Verarbeitung der Messwerte

<p>1 Filter zur Kompensation kurzzeitiger Signalschwankungen (siehe "Filter-Modul" auf Seite 32). Das Filter-Modul wird permanent durchlaufen.</p>	<p>7 Ermittlung der Stellwerte, Device Logic-Abarbeitung (siehe "Control-Modul" auf Seite 36)</p>
<p>2 Überwachung der Messbereichsgrenzen (siehe "Overflow-Modul" auf Seite 33)</p>	<p>8 Da das Aufzeichnungsintervall und das Messintervall unterschiedlich gewählt werden können, erfolgt nicht zu jedem Messzeitpunkt auch die Aufzeichnung der Messwerte und Stellwerte der Ausgänge (siehe "Record-Modul" auf Seite 37)</p>
<p>3 Umskalierung von Rohwert zu Messwert (siehe "Scale-Modul (Eingänge)" auf Seite 34)</p>	<p>9 Vorgabe der Stellwerte für die Ausgänge (siehe "Setpoint-Modul" auf Seite 39)</p>
<p>4 Dämpfungsmodul zum Zusammenfassen mehrerer Messwerte (siehe "Decay-Modul" auf Seite 35)</p>	<p>10 Umskalierung von Stellwert auf physikalische Größe für die Ausgabe (siehe "Scale-Modul (Ausgänge)" auf Seite 40)</p>
<p>5 Modul zum Halten des letzten gültigen Wertes bei ungültigen Messwerten (siehe "Hold-Modul" auf Seite 36)</p>	<p>11 Diese Modulkette wird zu jedem Messzeitpunkt gestartet und einmal durchlaufen. Für Universaleingänge, die im Modus Digital betrieben werden, wird diese Modulkette (mit Ausnahme des Control-Modul) zusätzlich im Sekundenintervall durchlaufen, um möglichst zeitnahe auf Pegelwechsel reagieren zu können.</p>
<p>6 Überwachung der Alarmgrenzen und Triggerschwellen (siehe "Alarm/Trigger-Modul" auf Seite 36)</p>	

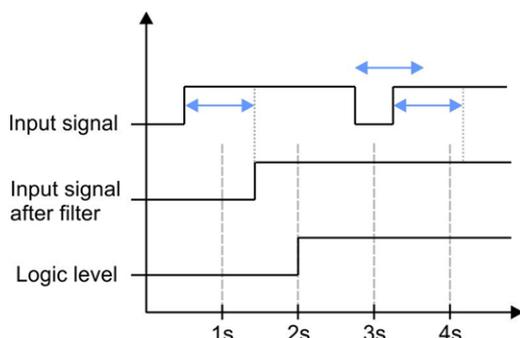
Hinweis: Ergänzende Erklärung zu Universaleingängen, die im Modus Digital betrieben werden.

Messkanäle -> Basis		Modus	Digital
Messkanäle -> Konfig.	↔	Filter Zeit	800ms

Input signal	Eingangssignal am Universaleingang
Input signal after filter	Eingangssignal unter Berücksichtigung der "Filter Zeit"
Logic level	Einmal pro sec. wird das Eingangssignal am Universaleingang analysiert.



Filter-Modul zur Kompensation kurzzeitiger Signalschwankungen nicht aktiv ("Filter Zeit" = 0)



Filter-Modul zur Kompensation kurzzeitiger Signalschwankungen aktiv ("Filter Zeit" ungleich 0)

Erklärung: Um möglichst zeitnahe auf Pegelwechsel an den Universaleingängen reagieren zu können, wird im Betriebsmodus Digital die unter "Interne Verarbeitung der Messwerte" auf Seite 31 beschriebene Modulkette (mit Ausnahme des Control-Moduls) für den entsprechenden Universaleingang ein mal pro sec. durchlaufen. Dadurch lassen sich Alarmer und Trigger auch asynchron zum Messintervall erkennen. Um einen Impuls sicher erkennen zu können, muss das Signal zumindest 1sec. anliegen. Mit Hilfe der "Filter Zeit" lassen sich zusätzlich kurzzeitige Störungen unterdrücken.

5.1.1 Filter-Modul

Das Filter-Modul dient der Kompensation von kurzzeitigen Schwankungen des Eingangssignals. Dieses Modul ist nur für die 8 Universaleingänge verfügbar.

Die folgende Tabelle gibt die für das Modul relevanten Parameter an:

Konfigurationsabschnitt	Modus	Parameter	Erklärung
Messkanäle -> Konfig	Digital	Filter Zeit	Zeit in [ms], für die ein Signal konstant anliegen muss, um einen Pegelwechsel auszulösen. Dient zur Unterdrückung von kurzzeitigen Störungen (Entprellung).
	Cnt.Day		
	Cnt.Intrvl.		
	Freq		
	PWM	Filter Zeit	Zeit in [ms], über die das Analogsignal zwecks Signalglättung gemittelt wird. Dient zur Unterdrückung von Signalrauschen (siehe auch "Beispiel zur Erklärung der Filter Zeit in Kombination mit der ext. Aufwärmzeit" auf Seite 129).
	4-20mA		
	0-20mA		
	0-2V		
0-10V			

5.1.2 Overflow-Modul

Dieses Modul überwacht die Messbereichsgrenzen des Rohwertes. Wurde z.B. ein Universaleingang in den Modus "4-20mA" geschaltet, führt ein Rohwert von 2mA zu einer Messbereichsverletzung. Das Overflow-Modul ist nur für die 8 Universaleingänge in den Kanalmodi "Freq", "PWM", "4-20mA", "0-20mA", "0-2V" und "0-10V" sowie für Interface Kanäle (nur Com1 und Com3) mit aktivierter Skalierung verfügbar.

Die folgende Tabelle gibt die für das Modul relevanten Parameter an:

Konfigurationsabschnitt	Modus/Interface	Parameter	Erklärung
Messkanäle -> Konfig	Digital	---	Handling bei Messbereichsverletzungen
	Cnt.Day		
	Cnt.Intrvl.		
	Freq	Überlauf	
	PWM		
	4-20mA		
	0-20mA		
	0-2V		
	0-10V		
Interface Kanäle 1-64 -> Skalierung	Com1	Überlauf	Handling bei Messbereichsverletzungen
	Com3		

5.1.3 Scale-Modul (Eingänge)

Dieses Modul kümmert sich um die Umskalierung von Rohwert (z.B. mA) in gewünschten Messwert (z.B. mm). Das Scale-Modul ist nur für die 8 Universaleingänge sowie für die Interface Kanäle (nur Com1 und Com3) verfügbar.

Die folgende Tabelle gibt die für das Modul relevanten Parameter an:

Konfigurationsabschnitt	Modus/Interface	Parameter	Erklärung
Messkanäle -> Basis	Digital	Invertieren	invertiert das Eingangssignal
	Cnt.Day	Impuls	Zählwert eines Impulses in der Messeinheit
	Cnt.Intrvl.		
	Freq	Faktor	Faktor mit dem das Eingangssignal multipliziert wird
	PWM 4-20mA 0-20mA 0-2V 0-10V	0%	Start des Messbereichs in der Messeinheit
		100%	Ende des Messbereichs in der Messeinheit
Interface Kanäle 1-64 -> Skalierung	Com1 Com3	0%	Start des Messbereichs in der Einheit des Modbus-Slaves
		100%	Ende des Messbereichs in der Einheit des Modbus-Slaves
		0% Modbus	Start des Messbereichs in der Messeinheit
		100% Modbus	Ende des Messbereichs in der Messeinheit

5.1.4 Decay-Modul

Das Dämpfungsmodul dient dazu, mehrere Messwerte zusammen zu fassen. Es kann z.B. der Mittelwert über ein gewünschtes Zeitfenster oder der Minimalwert innerhalb des gewünschten Zeitfensters ermittelt werden. Das Dämpfungsmodul ist nur für die 8 Universaleingänge mit Ausnahme des Kanalmodus "Cnt.Day" verfügbar.

Die folgende Tabelle gibt die für das Modul relevanten Parameter an:

Konfigurationsabschnitt	Modus	Parameter	Erklärung	
Messkanäle -> Konfig	Digital	Dämpfung	zeitliche Funktion im Messintervall	
		Zeit	Zeit x, die bei den Dämpfungsmodi "up", "down" und "up&down" eingesetzt wird	
	Cnt.Day	---	---	
	Cnt.Intervl. Freq PWM 4-20mA 0-20mA 0-2V 0-10V	Dämpfung		zeitliche Funktion im Messintervall
			Zeit	Zeitfenster für die Dämpfung. Für die Berechnung der Anzahl der berücksichtigten Messwerte siehe "Beispiel zur Erklärung Aufzeichnungs-, Mess-, Burstintervall in Verbindung mit der Dämpfung" auf Seite 134. Konnte kein temporärer Speicher mehr für die Berücksichtigung eines weiteren Messwerts reserviert werden, wird der Fehler "DECAY MEM ERR" ins Gerätelogs eingetragen (siehe "Log-Einträge und Fehlercodes" auf Seite 216).

5.1.5 Hold-Modul

Mit Hilfe des Hold-Moduls kann festgelegt werden, wie bei ungültigen Messwerten verfahren werden soll. Es ermöglicht den letzten gültigen Messwert so lange zu halten, bis ein neuer gültiger Messwert vorliegt oder den Fehler nach einer bestimmten Anzahl an ungültigen Messungen weiter zu geben. Das Hold-Modul ist nur für die Universaleingänge in den Kanalmodi "Freq", "PWM", "4-20mA", "0-20mA", "0-2V" und "0-10V" sowie für die Interface Kanäle verfügbar.

Die folgende Tabelle gibt die für das Modul relevanten Parameter an:

Konfigurationsabschnitt	Modus/Interface	Parameter	Erklärung
Schnittstellen -> Basis	Com1	Halten	Halten des letztgültigen Messwerts für x Messzyklen
	Com2		
	Com3		
Messkanäle -> Konfig	Digital	---	---
	Cnt.Day		
	Cnt.Intrvl.		
	Freq	Halten	Halten des letztgültigen Messwerts für x Messzyklen
	PWM		
	4-20mA		
	0-20mA		
	0-2V		
	0-10V		

5.1.6 Alarm/Trigger-Modul

Die Überwachung der Alarmgrenzen und Triggerschwellen wird von diesem Modul durchgeführt. Falls erforderlich, wird ein Eintrag in der Alarmliste erstellt. Ebenso werden die globalen Trigger gesetzt (siehe "Alarm Flags" bzw. "Trigger Flags" im Kapitel "Konstanten" auf Seite 161). Das Alarm/Trigger-Modul ist nur für Eingänge (Universaleingänge, ext. Temperatursensor, Interface Kanäle, interne Kanäle) verfügbar.

Für dieses Modul sind alle Parameter der Tabs "Alarmer" und "Trigger" der Konfigurationsabschnitte "Messkanäle" (siehe "Messkanäle" auf Seite 87), "Interface Kanäle 1-32" (siehe "Alarmer" auf Seite 113 und "Trigger" auf Seite 114) und "Interne Kanäle" (siehe "Interne Kanäle" auf Seite 130) relevant. Zusätzlich sind auch noch die Parameter der folgenden Tabellen relevant:

Konfigurationsabschnitt	Parameter	Erklärung
Alarmierung	Bei Alarm Bei Warnung Bei Störung-Alarm Bei Störung-Warnung	A Der Alarm wird in der Alarmliste aufgezeichnet.

5.1.7 Control-Modul

Das Control-Modul führt die Device Logic-Abarbeitung durch und ermittelt die Stellwerte für die Ausgänge. Bei der Ermittlung der Stellwerte für die Ausgänge gilt, dass ein mittels Device Logic berechneter Wert den

über die Eingabemaske am myDatenet-Server festgelegten Wert überschreibt. Ebenso können die Werte eines Messkanals durch die Device Logic vor der Aufzeichnung durch das Record-Modul noch verändert werden. Wurde als Device Logic Typ "Pawn" gewählt, kann mittels der Funktion "Mdn_SetAlarm()" (siehe Mdn_SetAlarm()) der Alarmstatus eines Kanals verändert werden. Ein durch das Alarm/Trigger-Modul erstellter Eintrag in der Alarmliste kann dabei nicht verändert sondern nur ein weiterer Eintrag für den entsprechenden Kanal erstellt werden.

Wurde als Device Logic Typ "IL" (Instruction List) gewählt, führt das Control-Modul die komplette Device Logic aus. Falls der Device Logic Typ "Pawn" gewählt wurde, wird die Funktion "Mdn_CtrlFinish()" (siehe Mdn_CtrlFinish()) vom Control-Modul aufgerufen.

Für dieses Modul sind neben den Parametern der folgenden Tabelle auch alle Parameter des Konfigurationsabschnitts "Steuerung" (siehe "Steuerung" auf Seite 75) relevant:

Konfigurationsabschnitt	Modus/Interface	Parameter	Erklärung
Ausgabekanäle	aus	---	---
	0-20mA		
	4-20mA		
	Ext. Aufwärmzeit	Ext Aufwärmzeit	Der Ausgabekanal wird "Ext. Aufwärmzeit" Sekunden vor der Messung eingeschaltet. Ist der Wert "0", wird der Ausgabekanal nicht eingeschaltet.
	Digital	---	----

5.1.8 Record-Modul

Die Aufzeichnung der Messwerte erfolgt durch das Record-Modul. Da das Aufzeichnungsintervall und das Messintervall unterschiedlich gewählt werden können, erfolgt nicht zu jedem Messzeitpunkt auch eine Aufzeichnung. Abhängig von den durch das Alarm/Trigger-Modul gesetzten globalen Triggern (siehe "Alarm Flags" bzw. "Trigger Flags" im Kapitel "Konstanten" auf Seite 161) wird falls erforderlich das Aufzeichnungsintervall modifiziert, die Übertragung ausgelöst oder eine neue Messung getriggert.

Die folgenden Tabellen geben die für das Modul relevanten Parameter an:

Konfigurationsabschnitt	Modus/Interface	Parameter	Erklärung	
Messkanäle -> Trigger	Digital	QU	schnelle Aufzeichnung (Aufzeichnungsintervall = Aufzeichnungsintervall / Faktor)	
		SL	langsame Aufzeichnung (Aufzeichnungsintervall = Aufzeichnungsintervall * Faktor)	
		MS	Messzyklus sofort starten	
		XM	Übertragung auslösen	
		ON	Online-Modus aktivieren	
	Cnt.Day Cnt.Intervl. Freq PWM 4-20mA 0-20mA 0-2V 0-10V ext. Temperatursensor	QU	schnelle Aufzeichnung (Aufzeichnungsintervall = Aufzeichnungsintervall / Faktor)	
		SL	langsame Aufzeichnung (Aufzeichnungsintervall = Aufzeichnungsintervall * Faktor)	
		RO	Aufzeichnung einschalten	
		RF	Aufzeichnung ausschalten	
		XM	Übertragung auslösen	
		ON	Online-Modus aktivieren	
	Interface Kanäle 1-64 -> Trigger	Com1 Com2 Com3 Device Logic	QU	schnelle Aufzeichnung (Aufzeichnungsintervall = Aufzeichnungsintervall / Faktor)
			SL	langsame Aufzeichnung (Aufzeichnungsintervall = Aufzeichnungsintervall * Faktor)
RO			Aufzeichnung einschalten	
RF			Aufzeichnung ausschalten	
XM			Übertragung auslösen	
ON			Online-Modus aktivieren	
Interne Kanäle -> Trigger	---	QU	schnelle Aufzeichnung (Aufzeichnungsintervall = Aufzeichnungsintervall / Faktor)	
		SL	langsame Aufzeichnung (Aufzeichnungsintervall = Aufzeichnungsintervall * Faktor)	
		RO	Aufzeichnung einschalten	
		RF	Aufzeichnung ausschalten	
		XM	Übertragung auslösen	
		ON	Online-Modus aktivieren	

Konfigurationsabschnitt	Parameter	Erklärung
Alarmierung	Bei Alarm Bei Warnung Bei Störung-Alarm Bei Störung-Warnung	Ü Eine sofortige Übertragung wird ausgelöst.

Konfigurationsabschnitt	Parameter	Erklärung
Grundeinstellungen	Aufzeichnungsintervall	zeitlicher Abstand der Messdatenaufzeichnungen
	Divisor schnelle Aufzeichnung	Aufzeichnungsintervall = Aufzeichnungsintervall / Faktor (ab Triggerung)
	Faktor langsame Aufzeichnung	Aufzeichnungsintervall = Aufzeichnungsintervall * Faktor (ab Triggerung)

5.1.9 Setpoint-Modul

Dieses Modul übernimmt die über die Konfigurationsoberfläche des myDatenet-Servers eingegebenen Stellwerte für die Ausgänge.

Die folgende Tabelle gibt die für das Modul relevanten Parameter an:

Konfigurationsabschnitt	Modus/Interface	Parameter	Erklärung
Interface Ausgabekanäle 1-64 -> Basis	aus	---	---
	Com1	Stellwert	Ausgabewert in der Messeinheit
	Com2		
	Com3		
	Device Logic		
Ausgabekanäle	aus	---	--
	0-20mA ¹⁾	Stellwert	Ausgabewert in der Messeinheit
	4-20mA ¹⁾		
	Ext Aufwärmzeit ²⁾³⁾	---	---
	Digital ²⁾	Stellwert	Stellwert (ein/aus), der ausgegeben werden soll

1) Dieser Modus ist nur für die Analogausgänge verfügbar.

2) Dieser Modus ist nur für die Relais verfügbar.

3) Im Modus "Ext Aufwärmzeit" wird der Ausgang vom Gerät selbst angesteuert um beispielsweise die Versorgung eines Sensors passend zur Messung durchzuschalten (siehe "Ausgabekanäle" auf Seite 125).

5.1.10 Scale-Modul (Ausgänge)

Dieses Modul kümmert sich um die Umskalierung von Stellwert (z.B. mm) in die gewünschten physikalische Größe (z.B. mA) für die Ausgabe.

Die folgende Tabelle gibt die für das Modul relevanten Parameter an:

Konfigurationsabschnitt	Modus/Interface	Parameter	Erklärung
Interface Ausgabekanäle 1-64 -> Skalierung	Com1 Com3	0% Modbus	Start des Ausgabebereichs in der Einheit des Modbus-Slaves
		100% Modbus	Ende des Ausgabebereichs in der Einheit des Modbus-Slaves
		0%	Start des Ausgabebereichs in der Messeinheit
		100%	Ende des Ausgabebereichs in der Messeinheit
Ausgabekanäle	aus	---	---
	0-20mA ¹⁾	0%	Start des Ausgabebereichs in der Messeinheit
		100%	Ende des Ausgabebereichs in der Messeinheit
	4-20mA ¹⁾	0%	Start des Ausgabebereichs in der Messeinheit
		100%	Ende des Ausgabebereichs in der Messeinheit
	Ext Aufwärmzeit ²⁾³⁾	---	---
	Digital ²⁾	Invertieren	invertiert den am Gerät ausgegebenen Pegel

¹⁾ Dieser Modus ist nur für die Analogausgänge verfügbar.

²⁾ Dieser Modus ist nur für die Relais verfügbar.

³⁾ Im Modus "Ext Aufwärmzeit" wird der Ausgang vom Gerät selbst angesteuert um beispielsweise die Versorgung eines Sensors passend zur Messung durchzuschalten (siehe "Ausgabekanäle" auf Seite 125).

5.2 Funktionsweise des internen Datenspeichers

Struktur	Ringspeicher
Gesamtgröße	4.470 Messzyklen
Anzahl der Sektoren	5
Sektorgröße	894 Messzyklen

Der interne Datenspeicher des myDatalogMUC xG/4G ist als Ringspeicher mit 5 Sektoren aufgebaut. Wurde die Anzahl der maximal möglichen Datensätze (4.470) erreicht, wird der Sektor mit den ältesten Daten

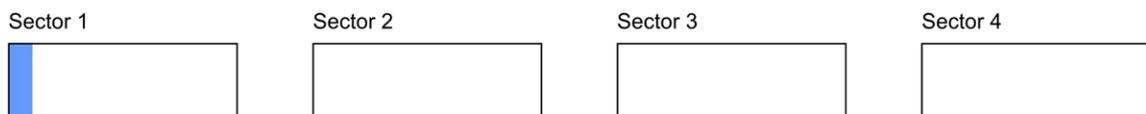
vollständig gelöscht bevor wieder neue Daten in diesem Sektor gespeichert werden können. D.h. der interne Datenspeicher enthält zumindest die Messwerte der letzten 3.576 Zyklen, maximal aber die Messwerte der letzten 4.470 Zyklen.

Aus diesem Grund empfiehlt es sich Übertragungsintervall und Aufzeichnungsintervall so aufeinander abzustimmen, dass zwischen zwei Übertragungen maximal 3.576 Messzyklen aufgezeichnet werden müssen. Beachten Sie, dass wenn das Messintervall kürzer als das Aufzeichnungsintervall ist, dennoch das Aufzeichnungsintervall für die Berechnung herangezogen werden muss. Der Grund dafür ist, dass in diesem Fall die Messung zwar im Messintervall erfolgt, aber die ermittelten Daten im Aufzeichnungsintervall im internen Datenspeicher abgelegt werden. Ist zu erwarten, dass aufgrund einer schlechten Netzabdeckung einzelne Übertragungen ausfallen oder mittels Trigger der Divisor für schnelle Aufzeichnung aktiviert wird, muss auch dies bei der Berechnung der zu speichernden Messzyklen berücksichtigt werden.

Hinweis:

Ergänzende Erklärung zur Funktionsweise des Ringspeichers

Datenspeicher nach dem ersten Messzyklus:



Datenspeicher nach 4.470 Messzyklen:



Datenspeicher nach 4.470 + 1 Messzyklen:



5.3 Vorgehensweise bei Verbindungsabbrüchen

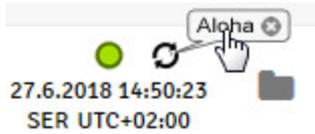
Für alle Verbindungen bis auf jene, die durch Auslösen des Aloha-Übertragungsmodus zustande kamen gilt, dass bei einem Abbruch der Verbindung nach 2min erneut versucht wird die Verbindung herzustellen. Der erneute Verbindungsaufbau erfolgt bis zu 2 mal, vorausgesetzt der vorhergehende erneute Verbindungsaufbau war erfolgreich. D.h. reißt die Verbindung ab und beim ersten Retry nach 2min ist es nicht möglich die Verbindung herzustellen, erfolgt nach weiteren 2min kein zweiter Retry.

Dieses Verfahren gilt auch, wenn es während des Online-Modus zu einem Abbruch der Verbindung kommt.

5.4 Aloha-Übertragungsmodus

Beim Aloha-Übertragungsmodus handelt es sich um einen speziellen Verbindungsmodus bei dem das myDatalogMUC xG/4G für eine über den Konfigurationsabschnitt „Grundeinstellungen“ konfigurierbare Zeit (siehe „Aloha/Wakeup Dauer“ im Kapitel "Grundeinstellung" auf Seite 133) eine Verbindung zum myDatanet-Server aufbaut. Ausgelöst wird der Aloha-Übertragungsmodus direkt am Gerät mittels Taste (siehe "Taste zum Auslösen des Aloha-Übertragungsmodus" auf Seite 72).

Dass sich ein Gerät im Aloha-Übertragungsmodus befindet, wird in der Messgeräteleiste (siehe "Benutzerhandbuch für myDatanet-Server " 206.886) durch eine Sprechblase mit der Beschriftung „Aloha“ angezeigt.



Durch Klicken auf die Sprechblase mit der Beschriftung „Aloha“ öffnen Sie das Aloha-Datenfenster (siehe "Benutzerhandbuch für myDatanet-Server " 206.886). Es beinhaltet die internen Messwerte „Spannung“ und „GSM Stärke“ sowie die Messwerte der 8 Universaleingänge. Im Aloha-Datenfenster werden immer die Messwerte der letzten 30min., unter Umständen somit auch jene einer vorangegangenen Aloha-Übertragung, angezeigt. Die Aloha-Daten werden unabhängig von der normalen Messwerterzeugung alle 3sec. generiert und somit auch nicht zusammen mit den Standardmessdaten abgespeichert.

Wird während des Aloha-Übertragungsmodus ein Messdatensatz aufgezeichnet, wird dieser sofort zum myDatanet-Server übertragen und in den Standardmessdaten abgespeichert.

Ist auf der Serveroberfläche ein Anzeigeelement sichtbar, das einen der Messwerte des Geräts verwendet, werden die Daten auch nach jeder Messwerterzeugung an den myDatanet-Server übertragen. Erfolgt die Messung öfters als die Aufzeichnung („Messintervall“ ungleich 0 und/oder „Burst Intervall“ ungleich 0) (siehe "Beispiel zur Erklärung Aufzeichnungs-, Mess-, Burstintervall in Verbindung mit der Dämpfung" auf Seite 134) kommt es dadurch zu einem erhöhten Datentransfer. Diese zusätzlichen Daten werden aber weder am Server noch im Gerät dauerhaft gespeichert, sondern am Server nur zur Aktualisierung der Anzeigeelemente (siehe "Benutzerhandbuch für myDatanet-Server " 206.886) verwendet.

5.5 Automatische Auswahl des GSM-Netzes

Da das myDatalogMUC xG/4G mit einem SIM-Chip ausgestattet ist, der eine Mobilfunkverbindung über eine Vielzahl internationaler Serviceprovider gewährleistet (siehe www.microtronics.com/footprint), ist eine Auswahl des GSM-Netzes, in das sich das Gerät einbuchen soll, erforderlich. Diese erfolgt automatisch vom Gerät.

5.6 Device Logic-Abarbeitung

Die PAWN-Device Logic muss folgende 2 Funktionen enthalten:

main();

Diese Funktion wird beim PowerOn und beim Austausch der Device Logic ausgeführt. Sie sollte alle Initialisierungen enthalten, die nur ein einziges Mal beim Programmstart durchgeführt werden müssen.

Mdn_CtrlFinish();

Diese Funktion ist der Einstiegspunkt für die Ausführung der Device Logic und wird zu jedem Messzeitpunkt, nachdem alle Messwerte erzeugt wurden und bevor die Ausgänge gesetzt wurden, aufgerufen. Sie sollte alle Berechnungen und Funktionen enthalten, die zyklisch durchgeführt werden sollen.

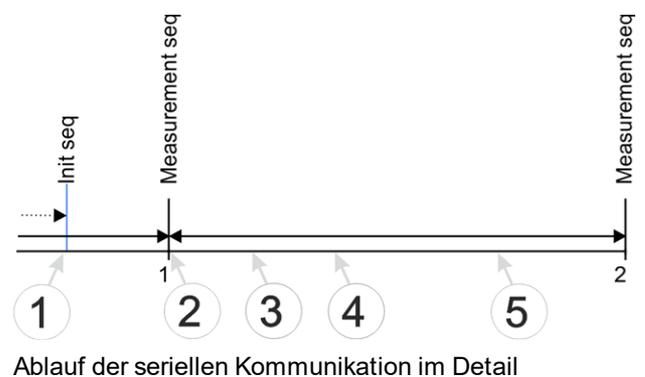
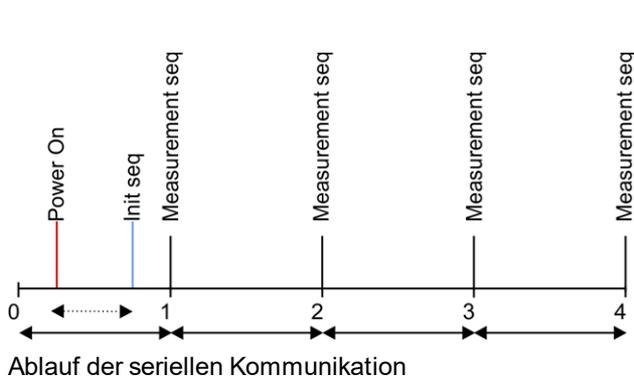
Hinweis: Wird mittels Device Logic direkt auf einen Ausgang geschrieben, überschreibt der ermittelte Wert den über die Eingabemaske am myDatanet-Server eingegebenen Stellwert.

Sollte bei der Device Logic-Ausführung ein Fehler auftreten, wird die Device Logic-Ausführung gestoppt und deaktiviert. Zudem wird in das Gerätelog der entsprechende Fehlercode eingetragen (siehe "Device Logic Fehlercodes" auf Seite 178).

5.6.1 Device Logic Parsing

Wurde das Device Logic Parsing für eine der Com-Schnittstellen aktiviert, kann mittels der Device Logic Funktionen "Mdn_SerialEvent()", "Mdn_SerialRx()", "Mdn_SerialTx()" und "Mdn_SerialFinish()" auf die betreffende Schnittstelle zugegriffen werden. Die Konfiguration der Baudrate, Stoppbits, Parität, Datenbits, des Datensatz Timeout sowie das Aktivieren des Retry erfolgt nach wie vor über den Konfigurationsabschnitt "Schnittstelle" (siehe "Schnittstellen" auf Seite 76). Die beiden folgenden Grafiken zeigen zu welchen Zeitpunkten die Callback Funktionen "Mdn_SerialEvent()" (siehe Mdn_SerialEvent()) und "Mdn_SerialRx()" (siehe Mdn_SerialRx()) aufgerufen werden und welche Parameter dabei übergeben werden.

Grundeinstellung		Messintervall	60sec.
Schnittstellen -> Konfig		Aufwärmzeit	30sec.



1 Funktion: Mdn_SerialEvent() Event: MDN_SERIAL_EVENT_INIT	2 Funktion: Mdn_SerialEvent() Event: MDN_SERIAL_EVENT_MEASURE
3 Funktion: Mdn_SerialEvent() Event: MDN_SERIAL_EVENT_TIMEOUT Event: MDN_SERIAL_EVENT_MEASURE oder Mdn_SerialRx()	4 Nur wenn "Retry" aktiv ist Funktion: Mdn_SerialEvent() Event: MDN_SERIAL_EVENT_TIMEOUT oder Mdn_SerialRx()
5 Funktion: Mdn_CtrlFinish()	

Erklärung zu 1

Über die Aufwärmzeit lässt sich eine Verzögerung zwischen PowerOn und dem Senden der Init Sequenz (Aufruf der Callback Funktion "Mdn_SerialEvent()" durch das System mit dem Event "MDN_SERIAL_EVENT_INIT") einstellen. Allerdings ist über die Konfigurationsoberfläche die Eingabe der Aufwärmzeit nur für die Com2 möglich. Der entsprechende Parameter befindet sich im Tab "Konfig" der Eingabemaske zur Konfiguration der Schnittstelle (siehe "Konfig" auf Seite 85). Für die Schnittstellen Com1 und Com3 ist die Aufwärmzeit immer 0sec. Auf das Event "MDN_SERIAL_EVENT_INIT" sollte der Anwender reagieren indem er die Funktion "Mdn_SerialTx()" benutzt, um die Init Sequenz über die Schnittstelle zu versenden.

Erklärung zu 2

Ist der Zeitpunkt zum Auslösen einer Messung gekommen, wird die Callback Funktion "Mdn_SerialEvent()" durch das System mit dem Event "MDN_SERIAL_EVENT_MEASURE" aufgerufen. Zu dem Zeitpunkt startet auch die interne Verarbeitung der Messwerte (siehe "Interne Verarbeitung der Messwerte" auf Seite 31). Auf das Event "MDN_SERIAL_EVENT_MEASURE" muss der Anwender reagieren indem er die Funktion "Mdn_SerialTx()" benutzt, um die Measurement Sequenz über die Schnittstelle zu versenden.

Hinweis: Der erste Messzeitpunkt nach dem PowerOn erfolgt nicht genau nach der über das Messintervall festgelegten Zeit nach dem PowerOn, sondern wird berechnet. Bei einem Messintervall von 1min. wird der erste Messzeitpunkt so gewählt, dass er zur vollen Minute erfolgt. D.h. Wäre der PowerOn um 12:05:34, erfolgt die erste Messung um 12:06:00 also 26sec. nach dem PowerOn.

Erklärung zu 3

Dieser Punkt wird erreicht, wenn entweder Daten empfangen wurden oder ein Timeout aufgetreten ist. Die Zeit für den Timeout wird über den Tab "Basis" der Eingabemaske zur Konfiguration der Schnittstellen (siehe "Konfig" auf Seite 85) eingetragen.

Ist der Retry aktiviert, wird durch das System im Falle eines Timeouts die Callback Funktion "Mdn_SerialEvent()" mit dem Event "MDN_SERIAL_EVENT_MEASURE" aufgerufen. Der Anwender muss auf dieses Event reagieren indem er die Funktion "Mdn_SerialTx()" benutzt, um die Measurement Sequenz erneut zu versenden.

Wurde der Retry nicht aktiviert, wird durch das System im Falle eines Timeout die Callback Funktion "Mdn_SerialEvent()" mit dem Event "MDN_SERIAL_EVENT_TIMEOUT" aufgerufen. In diesem Fall muss der Datenempfang mit der Funktion "Mdn_SerialFinish()" beendet werden.

Sollten jedoch gültige Daten empfangen worden sein, wird durch das System die Callback Funktion "Mdn_SerialRx()" aufgerufen. Sie liefert die empfangenen Daten als Array. Nach der Auswertung der Daten muss der Anwender die gewünschten Messwerte mittels der Funktion "Mdn_SetCh()" (siehe Mdn_SetCh()) in den Interface Kanal kopieren.

Wichtiger Hinweis: Es sollten hier nur Interface Kanäle verwendet werden, die mit einer Schnittstelle verbunden sind bei der das Device Logic Parsing aktiviert wurde. Andernfalls werden die durch das System erzeugten Daten überschrieben.

Wurden alle benötigten Daten empfangen, muss der Datenempfang mit der Funktion "Mdn_SerialFinish()" beendet werden. Andernfalls kann die interne Verarbeitung der Messwerte nicht fortgesetzt werden. Nach dem Aufruf der Funktion "Mdn_SerialFinish()" wird die interne Verarbeitung der Messwerte ab dem Decay-Modul fortgesetzt (siehe "Interne Verarbeitung der Messwerte" auf Seite 31). D.h. die Schwellen für Alarme oder Trigger können wie gewohnt über die Tabs "Alarme" und "Trigger" der Eingabemaske zur Konfiguration der Interface Kanäle 1-64 (siehe "Alarme" auf Seite 113 bzw. "Trigger" auf Seite 114) editiert werden.

Erklärung zu 4 (nur wenn "Retry" aktiv ist)

Im Falle eines Timeout wird durch das System die Callback Funktion "Mdn_SerialEvent()" mit dem Event "MDN_SERIAL_EVENT_TIMEOUT" aufgerufen. In diesem Fall muss der Datenempfang mit der Funktion "Mdn_SerialFinish()" beendet werden. Sollten jedoch gültige Daten empfangen worden sein, wird durch das System die Callback Funktion "Mdn_SerialRx()" aufgerufen. Sie liefert die empfangenen Daten als Array. Nach der Auswertung der Daten muss der Anwender die gewünschten Messwerte mittels der Funktion "Mdn_SetCh()" (siehe Mdn_SetCh()) in den Interface Kanal kopieren.

Wichtiger Hinweis: Es sollten hier nur Interface Kanäle verwendet werden, die mit einer Schnittstelle verbunden sind bei der das Device Logic Parsing aktiviert wurde. Andernfalls werden die durch das System erzeugten Daten überschrieben.

Wurden alle benötigten Daten empfangen, muss der Datenempfang mit der Funktion "Mdn_SerialFinish()" beendet werden. Andernfalls kann die interne Verarbeitung der Messwerte nicht fortgesetzt werden. Nach dem Aufruf der Funktion "Mdn_SerialFinish()" wird die interne Verarbeitung der Messwerte ab dem Decay-Modul fortgesetzt (siehe "Interne Verarbeitung der Messwerte" auf Seite 31). D.h. die Schwellen für Alarme oder Trigger können wie gewohnt über die Tabs "Alarme" und "Trigger" der Eingabemaske zur Konfiguration der Interface Kanäle 1-64 (siehe "Alarme" auf Seite 113 bzw. "Trigger" auf Seite 114) editiert werden.

Erklärung zu 5

Ist die interne Verarbeitung der Messwerte zum Control-Modul gelangt, wird die Callbackfunktion "Mdn_CtrlFinish()" (siehe Mdn_CtrlFinish()) aufgerufen. Hier ist es möglich, weitere Kommandos mittels der Funktion "Mdn_SerialTx()" über die Schnittstelle zu versenden, um z.B. Aktoren zu steuern.

Kapitel 6 Lagerung, Lieferung und Transport

6.1 Eingangskontrolle

Kontrollieren Sie den Lieferumfang sofort nach Eingang auf Vollständigkeit und augenscheinliche Unversehrtheit. Melden Sie eventuell festgestellte Transportschäden unverzüglich an den anliefernden Frachtführer. Senden Sie ebenfalls unverzüglich eine schriftliche Meldung an Microtronics Engineering GmbH. Unvollständigkeiten der Lieferung müssen innerhalb von 2 Wochen schriftlich an Ihre zuständige Vertretung oder direkt an die Firmenzentrale des Herstellers (siehe "Kontaktinformationen" auf Seite 231) gerichtet werden.

Hinweis: Später eingehende Reklamationen werden nicht anerkannt!

6.2 Lieferumfang

Zum Standardlieferumfang des myDatalogMUC 2G/4G EU (301065) gehören:

- myDatalogMUC 2G/4G EU
- myDatanet Tool Pen (206.646)
- Typenschild-Aufkleber zum Anbringen am Schaltschrank

Zum Standardlieferumfang des myDatalogMUC 3G/4G US (301072) gehören:

- myDatalogMUC 3G/4G US
- myDatanet Tool Pen (206.646)
- Typenschild-Aufkleber zum Anbringen am Schaltschrank

Weiteres Zubehör wie Montagesets, Antennen, Netzteile, Sensoren usw. je nach Bestellung. Diese bitte anhand des Lieferscheins prüfen.

6.3 Lagerung

Folgende Lagerbedingungen sind unbedingt einzuhalten:

myDatalogMUC xG/4G	Lagertemperatur	-20...+60°C
	Feuchte	15...90%rH

Die Messtechnik ist vor korrosiven und organischen Lösungsmitteldämpfen, radioaktiver Strahlung sowie starker elektromagnetischer Strahlung geschützt aufzubewahren.

6.4 Transport

Das myDatalogMUC xG/4G sollte keinen starken Stößen, Schlägen, Erschütterungen oder Vibrationen ausgesetzt werden. Der Transport muss in der Originalverpackung erfolgen.

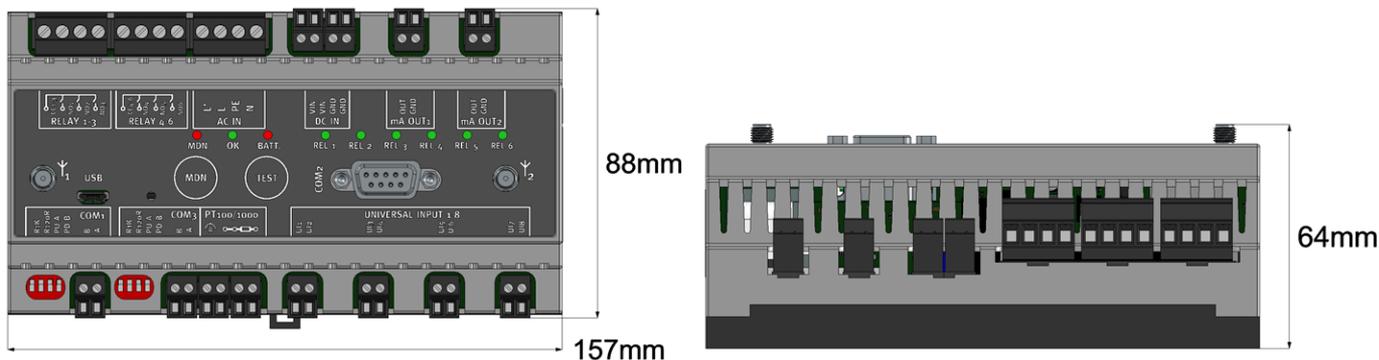
6.5 Rücksendung

Jeder Rücksendung muss ein vollständig ausgefülltes Retourenformular, welches im Servicebereich des myDatanet-Servers erhältlich ist, beigelegt werden. Die unbedingt erforderliche "RMA Nr" erhalten Sie vom Support & Service-Center (siehe "Kontaktinformationen" auf Seite 231). Die Rücksendung des myDatalogMUC xG/4G muss in der Originalverpackung frachtfrei zu Microtronics Engineering GmbH (siehe "Kontaktinformationen" auf Seite 231) erfolgen. Nicht ausreichend frei gemachte Sendungen werden nicht angenommen!

Kapitel 7 Installation

Wichtiger Hinweis: Um Schäden am Gerät zu vermeiden, dürfen die in diesem Abschnitt der Anleitung beschriebenen Arbeiten nur von qualifiziertem Personal ausgeführt werden.

7.1 Abmessungen



Abmessungen: Breite und Höhe

Abmessungen: Tiefe

7.2 Montage des myDatalogMUC xG/4G

Wichtiger Hinweis:

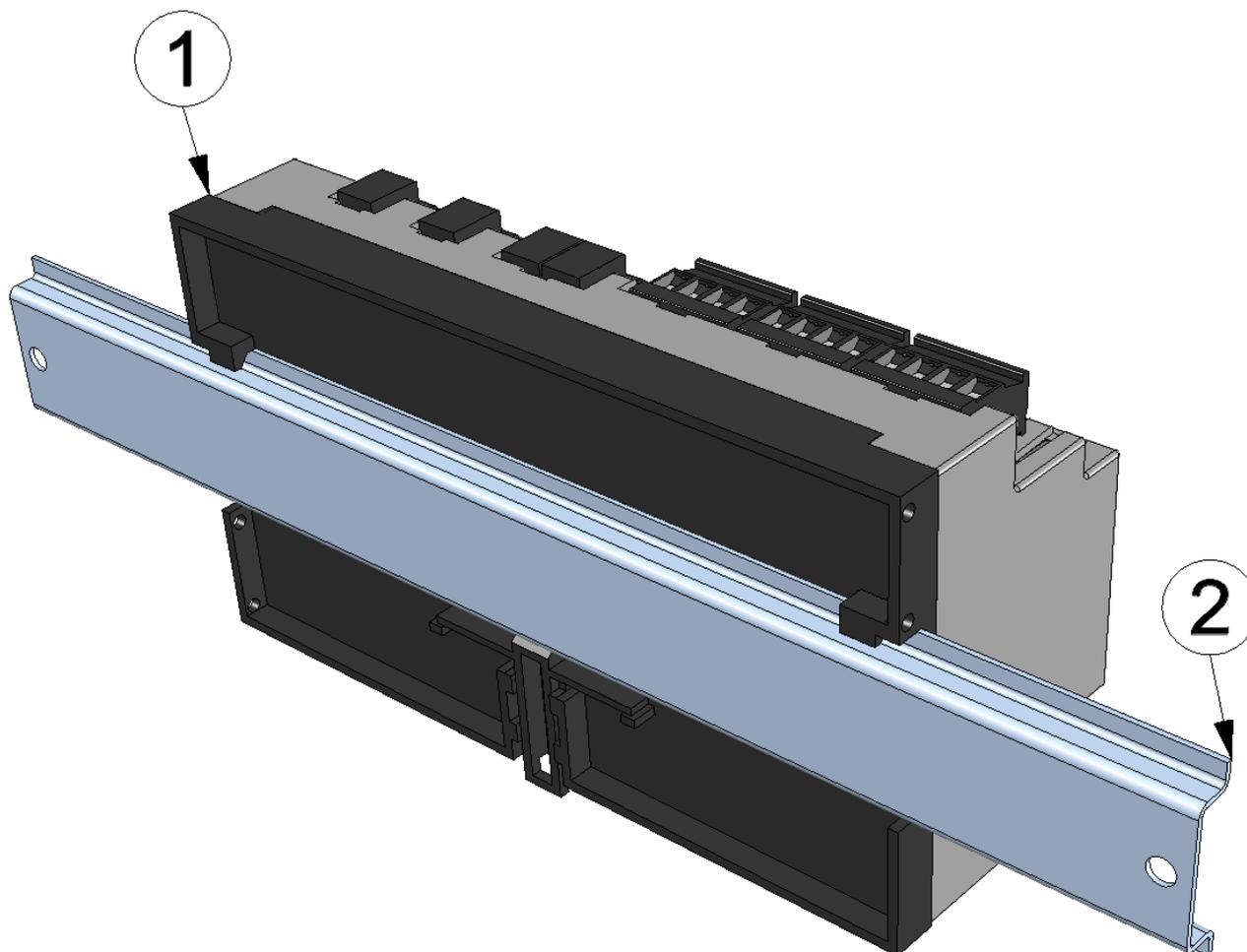
- Achten Sie auf eine sachgemäße Montage!
- Befolgen Sie bestehende gesetzliche bzw. betriebliche Richtlinien!
- Unsachgemäße Handhabung kann zu Verletzungen und/oder Beschädigungen an den Instrumenten führen!

Der Platz für die Montage muss nach bestimmten Kriterien ausgewählt werden. Vermeiden Sie unbedingt die folgenden Gegebenheiten:

- direkte Sonneneinstrahlung
- direkte Witterungseinflüsse (Regen, Schnee, ...)
- Gegenstände, die starke Hitze ausstrahlen (maximale Umgebungstemperatur: -20...+60°C)
- Objekte mit starkem elektromagnetischem Feld (Frequenzumrichter o.ä.)
- korrodierende Chemikalien oder Gase
- mechanische Stöße
- direkte Installation an Geh- oder Fahrwegen
- Vibrationen
- radioaktive Strahlung

Hinweis: Unter und über dem Gerät sollten Sie ca. 2-5cm Abstand für die Kabelanschlüsse vorsehen. Die Antennenanschlüsse befindet sich auf der Vorderseite des Geräts. Der benötigte Platz richtet sich nach den verwendeten Antennen. Weitere Informationen zu den Abmessungen für die Montage entnehmen Sie dem jeweiligen Unterkapitel.

7.2.1 Hutschienenmontage



Hutschienenmontage

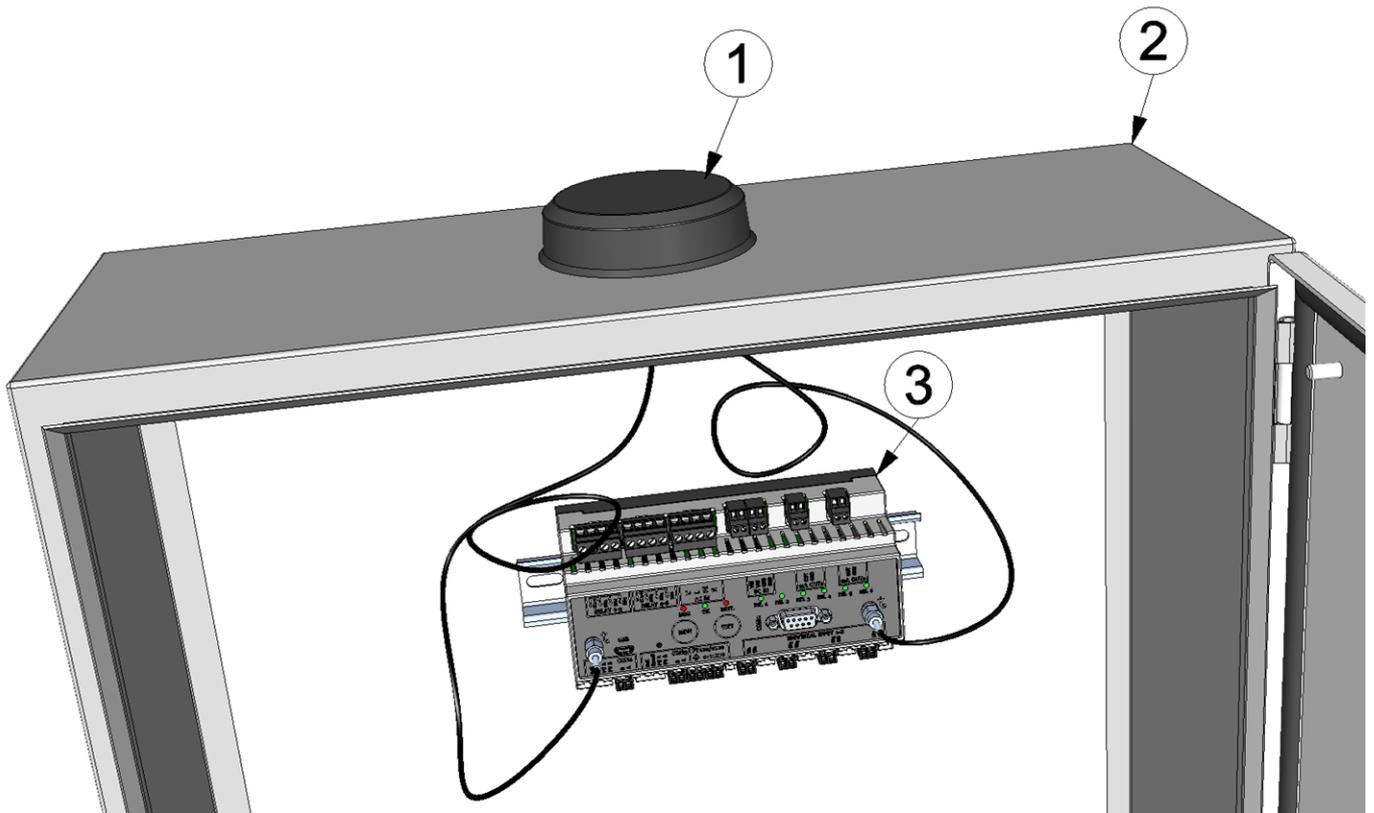
1 myDatalogMUC xG/4G

2 Hutschiene

1. Setzen Sie das myDatalogMUC xG/4G auf der Oberkante der Hutschiene auf. Durch eine leichte Drehung um die Horizontalachse rastet das myDatalogMUC xG/4G auf der Hutschiene ein (siehe Abbildung "Hutschienenmontage" auf Seite 50).

7.2.2 Montage in einem Schaltschrank

Direkt am myDatalogMUC xG/4G montierte Antennen eignen sich nicht für die Montage innerhalb eines Schaltschranks, da das GSM-Signal durch das Metall des Schrankes abgeschirmt wird. Der Hersteller empfiehlt in diesem Fall die Verwendung von als Zubehör erhältlichen Flachantennen.



Schaltschrank mit montierter Flachantenne

1 Flachantenne	3 myDatalogMUC xG/4G
2 Schaltschrank	

7.3 Sicherheitshinweise zur Verkabelung

Wichtiger Hinweis: Um Schäden zu vermeiden, stellen Sie stets die Spannungsversorgung am Gerät ab, wenn elektrische Anschlüsse durchgeführt werden.

Wenn Anschlüsse an das myDatalogMUC xG/4G gelegt werden, müssen die folgenden Warnungen und Hinweise ebenso beachtet werden, wie Warnungen und Hinweise, die in den einzelnen Kapiteln zum Einbau zu finden sind. Weitere Sicherheitsinformationen finden Sie unter "Sicherheitshinweise" auf Seite 19.

7.3.1 Hinweise zur Vermeidung elektrostatischer Entladungen (ESD)

Wichtiger Hinweis: Um Gefahren und ESD-Risiken zu minimieren, sollten Wartungsprozeduren, für die keine Stromversorgung des Geräts erforderlich ist, nur nach Trennung vom Stromnetz ausgeführt werden.

Die empfindlichen elektronischen Komponenten im Geräteinneren können durch statische Elektrizität beschädigt werden, was zur Beeinträchtigung der Geräteleistung bis hin zum Ausfall des Geräts führen kann. Der Hersteller empfiehlt die folgenden Schritte zur Vermeidung von Beschädigungen des Geräts durch elektrostatische Entladungen:

- Leiten Sie eventuell auf Ihrem Körper vorhandene statische Elektrizität ab, bevor Sie elektronische Komponenten des Geräts (wie z.B. Leiterplatten und die Komponenten darauf) berühren. Hierzu können Sie eine geerdete metallische Oberfläche berühren, wie etwa den Gehäuserahmen eines Geräts oder ein Metallrohr.
- Vermeiden Sie unnötige Bewegungen, um den Aufbau statischer Ladungen zu vermindern.
- Transportieren Sie statisch-empfindliche Komponenten in antistatischen Behältnissen oder Verpackungen.
- Tragen Sie ein Antistatikarmband, das über ein Kabel geerdet ist, um Ihren Körper zu entladen und von statischer Elektrizität freizuhalten.
- Fassen Sie Komponenten, die gegen Aufladungen empfindlich sind, nur in einem Antistatik-Arbeitsbereich an. Verwenden Sie, falls möglich, antistatische Fußbodenbeläge und Arbeitsunterlagen.

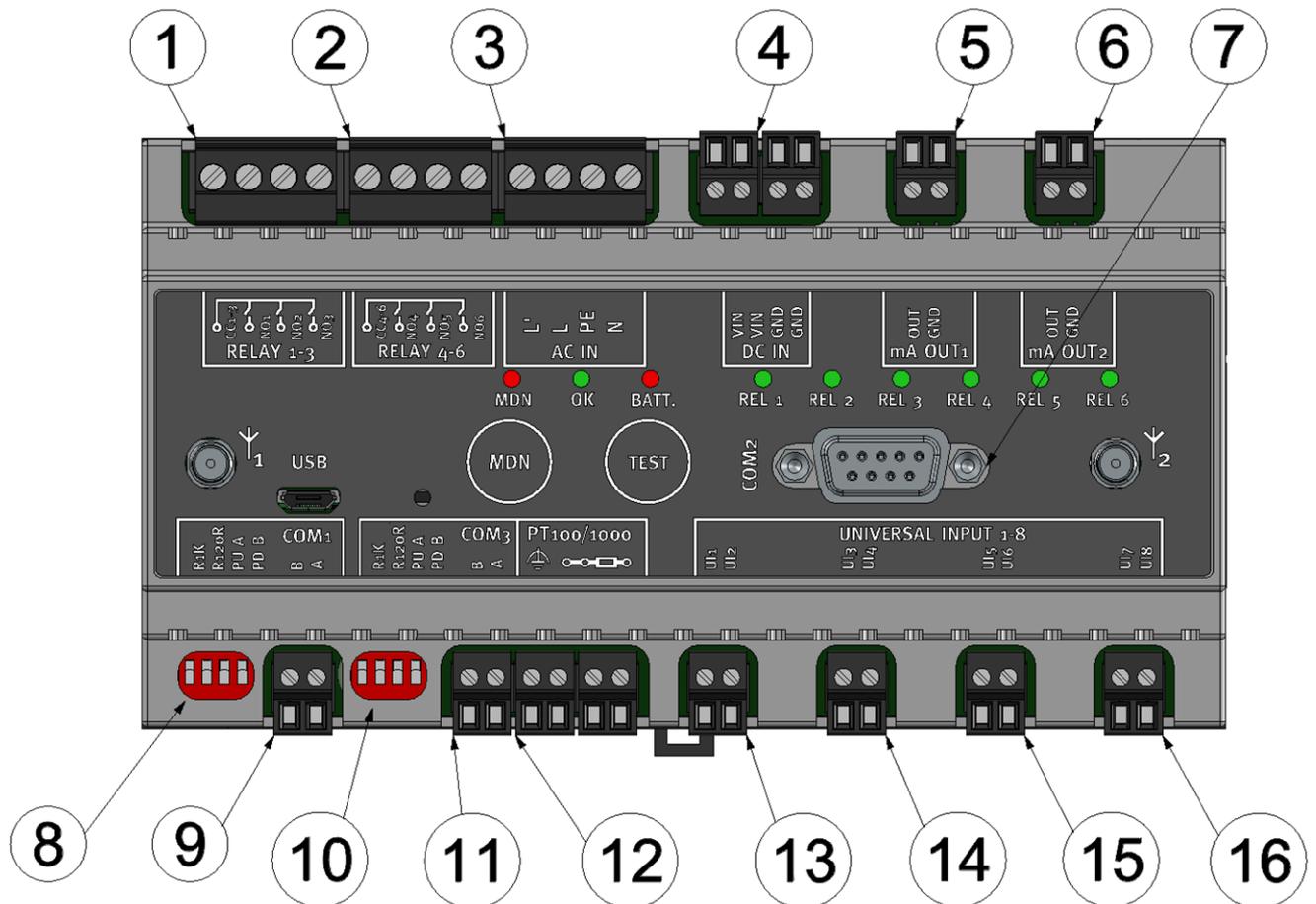
7.4 Elektrische Installation

Wichtiger Hinweis: Um Schäden am Gerät zu vermeiden, darf nur qualifiziertes Personal die in diesem Kapitel der Bedienungsanleitung beschriebene Installation durchführen.

7.4.1 Anschluss der Sensoren, der Aktoren und der Versorgung

Wichtiger Hinweis:

- Alle Verkabelungsarbeiten sollten im stromlosen Zustand erfolgen!
- Achten Sie auf eine sachgemäße Montage!
- Befolgen Sie bestehende gesetzliche bzw. betriebliche Richtlinien!
- Unsachgemäße Handhabung kann zu Verletzungen und/oder Beschädigungen an den Instrumenten führen!
- Verlegen Sie alle Daten- und Stromkabel so, dass sie keine Stolpergefahr darstellen und die Kabel keine scharfen Krümmungen aufweisen.



Anschluss der Sensoren und der Versorgung

1 Relais 1-3	9 Com1 (RS485)
2 Relais 4-6	10 Dip-Switch (Dip-Switch 2) zum Ein/Ausschalten der Abschluss- bzw. Klemmwiderstände für Com3 (RS485)
3 reserviert für Erweiterungen	11 Com3 (RS485)
4 Versorgung (V IN, GND)	12 Anschluss für den externen Temperaturfühler (PT100/1000)
5 mA Out1	13 Universaleingang 1-2
6 mA Out2	14 Universaleingang 3-4
7 Com2 (RS232)	15 Universaleingang 5-6
8 Dip-Switch (Dip-Switch 1) zum Ein/Ausschalten der Abschluss- bzw. Klemmwiderstände für Com1 (RS485)	16 Universaleingang 7-8

RELAY 1-3

CC1-3	Gemeinsame Wurzel für Relais 1-3
NO1	Arbeitskontakt des Relais 1 (Normally Open)
NO2	Arbeitskontakt des Relais 2 (Normally Open)
NO3	Arbeitskontakt des Relais 3 (Normally Open)

RELAY 4-6

CC4-6	Gemeinsame Wurzel für Relais 4-6
NO4	Arbeitskontakt des Relais 4 (Normally Open)
NO5	Arbeitskontakt des Relais 5 (Normally Open)
NO6	Arbeitskontakt des Relais 6 (Normally Open)

DC IN

GND	Masse
V IN	Versorgungsspannung: 12...30VDC (+/-10%), max. 3W

mA OUT1

GND	Masse
OUT	0/4-20mA Ausgangssignal

mA OUT2

GND	Masse
OUT	0/4-20mA Ausgangssignal

COM1

R 1k	1k Abschlusswiderstand zw. RS485 A und B
R 120 R	120Ω Abschlusswiderstand zw. RS485 A und B
PU A	1k Pull up auf RS485 A
PD B	1k Pull down auf RS485 B
B	RS485 B
A	RS485 A

COM3

R 1k	1k Abschlusswiderstand zw. RS485 A und B
R 120 R	120Ω Abschlusswiderstand zw. RS485 A und B
PU A	1k Pull up auf RS485 A
PD B	1k Pull down auf RS485 B
B	RS485 B
A	RS485 B

PT 100/1000

	Klemme für den Kabelschirm des externen Temperatursensors
	Klemmen für den externen Temperatursensor (2-Leiter oder 3-Leiter)

UNIVERSAL INPUTS 1-8

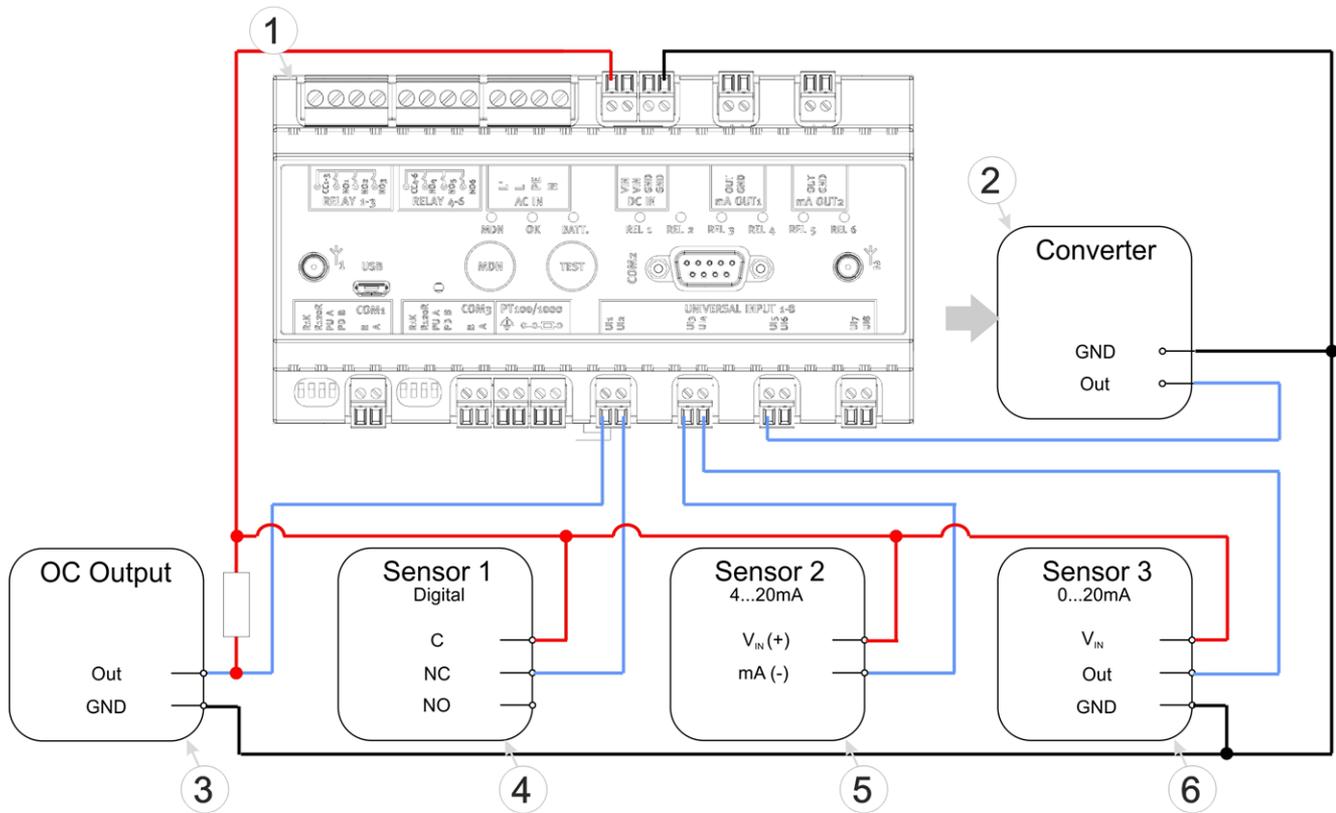
UI 1	Universaleingang 1
UI 2	Universaleingang 2
UI 3	Universaleingang 3
UI 4	Universaleingang 4
UI 5	Universaleingang 5
UI 6	Universaleingang 6
UI 7	Universaleingang 7
UI 8	Universaleingang 8

1. Verbinden Sie Ihre Sensoren und Aktoren mit den Eingängen und Ausgängen. Achten Sie dabei auf Stromlosigkeit! Die Kabel zur Versorgung des myDatalogMUC xG/4G sollten im stromlosen Zustand mit den Versorgungsklemmen verbunden werden.
2. Schließen Sie die Antenne an (siehe "Anschluss der GSM-Antenne" auf Seite 56).
3. Schalten Sie die 12...30VDC Versorgungsspannung des myDatalogMUC xG/4G ein. Daraufhin sollte das Status-LED zu flackern beginnen (siehe "Status-LED" auf Seite 72), um den Verbindungsaufbau zu signalisieren.

Der folgende Schritt ist nicht zwingend erforderlich.

4. Überprüfen Sie, ob die Verbindung zum myDatanet korrekt funktioniert hat (siehe "Kommunikation mit dem Gerät testen" auf Seite 68).

7.4.1.1 Anschlussbeispiele



Anschlussbeispiele

1 myDatalogMUC xG/4G	4 potentialfreier Relaiskontakt
2 Signalwandler, Trennwandler	5 2-Leiter mA-Sensor
3 Sensor mit Open Collector Ausgang	6 3-Leiter mA-Sensor

7.4.2 Anschluss der GSM-Antenne

Wichtiger Hinweis: Um eine korrekte Funktion zu gewährleisten, benutzen Sie nur Antennen, die vom Hersteller geliefert werden.

Die Standardantennen werden direkt an den Antennensteckern (siehe "Übersicht" auf Seite 22) des myDatalogMUC xG/4G angebracht. Im Falle einer niedrigen Funksignalstärke können Sie die Flachantenne Smart Disc Multi Band 2xSMA-M 2m (301090) verwenden.

Wenn die Entfernung zwischen der Lage der Antenne und dem myDatalogMUC xG/4G zu groß ist, können Sie eine 2,5m Antennenverlängerung SMA-M/SMA-F 2,5m (206.807) verwenden.

1. Stellen Sie sicher, dass das myDatalogMUC xG/4G spannungslos ist.
2. Verbinden Sie, falls Sie eine Antennenverlängerung benötigen, diese zuerst mit der Antenne.
3. Verbinden Sie die Antennenverlängerung bzw. die Antenne direkt mit dem Antennenanschluss des myDatalogMUC xG/4G (siehe "Übersicht" auf Seite 22).

Wichtiger Hinweis: Vermeiden Sie zu starke Krafteinwirkung beim Festziehen der Antenne. Benutzen Sie kein Werkzeug zum Festziehen der Antenne bzw. der Antennenverlängerung, sondern ziehen Sie die Antenne mit der Hand fest.

4. Stellen Sie die Spannungsversorgung des myDatalogMUC xG/4G wieder her.

Der folgende Schritt ist nicht zwingend erforderlich.

5. Überprüfen Sie, ob die Verbindung zum myDatenet korrekt funktioniert hat (siehe "Kommunikation mit dem Gerät testen" auf Seite 68).

7.4.3 Technische Details zu den Universaleingängen

Hinweis: Die Universaleingänge sind galvanisch nicht getrennt.

7.4.3.1 0/4...20mA Modus

Hinweis: Über 23,7mA wird der betroffene Eingang hochohmig (Sicherheitsabschaltung, um Schäden am Universaleingang zu vermeiden).

Auflösung	6,3 μ A
I _{max}	23,7mA
Bürde	96 Ω

7.4.3.2 0...2V Modus

Auflösung	610 μ V
U _{max}	2,5V
Bürde	10k086

7.4.3.3 0...10V Modus

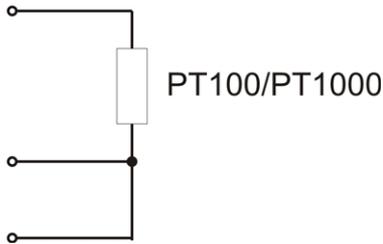
Auflösung	7,97mV
U _{max}	32V
Bürde	4k7

7.4.3.4 Standard Digitalmodi (PWM, Frequenz, Digital, Tageszähler, Impulszähler)

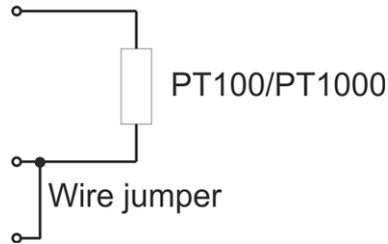
Allgemein	U _{max}	32V
	Low	<1,31V
	High	>2,61V
	Bürde	4k7
PWM	Messbereich	1...99%
	f _{max}	100Hz
	Impulslänge min.	1ms
Frequenz	Messbereich	1...1000Hz
Tages- und Impulszähler	Impulslänge min.	1ms

7.4.4 Technische Details zur PT100/1000-Schnittstelle

Die Schnittstelle für den externen Temperatursensor erkennt automatisch, ob ein PT100 oder PT1000 verwendet wird. Ebenso ist es möglich sowohl 3-Leiter als auch 2-Leiter Sensoren zu verwenden. Bei 2-Leiter Sensoren ist ein zusätzlicher Bügel (siehe "PT100/PT1000 2-Leiter" auf Seite 58) erforderlich.



PT100/PT1000 3-Leiter



PT100/PT1000 2-Leiter

7.4.5 Technische Details zu den Modbus-Schnittstellen (Com1, Com3)

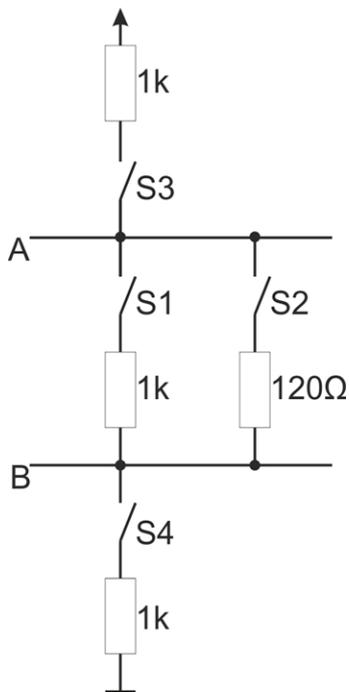
Hinweis: Die Modbus-Schnittstellen Com1 und Com3 entsprechen der Norm EIA-485.

Die Modbus-Schnittstellen Com1 und Com3 verfügen über einen Eingangsgleichtaktbereich der den gesamten für die RS485 spezifizierten Bereich (-7V...+12V) abdeckt. Höhere Spannungen führen zur Beschädigung der Schnittstelle. Signale mit einem Differenz-Spannungspegel von mehr als +/- 200mV innerhalb des spezifizierten Eingangsgleichtaktbereiches werden korrekt erkannt. Im Sendemodus liegt das Ausgangssignal im Bereich von 1,5...3,3V .

Baudrate	300-115200
Stopbits	1, 2
Parität	N, E, O
Datenbits	7, 8
Modi	RTU ASCII

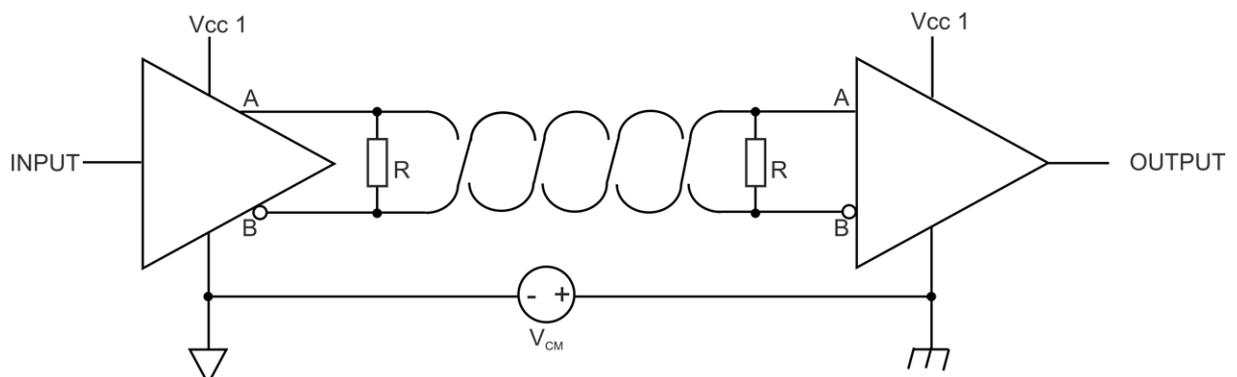
Bei Com1 und Com3 handelt es sich um RS485 Schnittstellen, die sowohl als Modbus-Master als auch als Modbus-Slave konfiguriert werden können (siehe "Schnittstellen" auf Seite 76). Im Modbus-Master Modus ist es möglich einen oder mehrere Slaves mit dem myDatalogMUC xG/4G zu verbinden. Über den Dip-Switch 1 (Com1) bzw. Dip-Switch 2 (Com3) (siehe "Anschluss der Sensoren, der Aktoren und der Versorgung" auf Seite 52) können die Abschluss- bzw. Klemmwidstände zugeschaltet werden.

S1	1k Abschlusswiderstand zw. RS485 A und B
S2	120Ω Abschlusswiderstand zw. RS485 A und B
S3	1k Pull up auf RS485 A
S4	1k Pull down auf RS485 B



Prinzipschaltbild zu den zuschaltbaren Widerständen

Hinweis: Ergänzende Erklärung zur Verbindung zweier RS485 Busteilnehmer



Prinzipschaltbild: Verbindung zweier RS485 Busteilnehmer

Ein Problem entsteht, wenn keine Verbindung zwischen den GND-Potentialen von Sender und Empfänger besteht. In diesem Fall entsteht eine Gleichtaktspannung (V_{CM}). Der GND-Potentialunterschied darf max. +/- 7V betragen. Bei höheren Spannungen kommt es zur Beschädigung der Schnittstelle. Kurzzeitige Überspannungen (ESD, EFT und Surge) werden jedoch durch Schutzschaltungen abgefangen.

Anmerkung: Der für RS485 spezifizierte Gleichakteingangsspannungsbereich von -7V...+12V ergibt sich aus dem max. zulässigen GND-Potentialunterschied (+/- 7V) und dem für RS485 max. zulässigen Ausgangsspannungsbereich von 0...5V.

7.4.5.1 Modbus-Slave Modus

Die folgende Tabelle gibt die möglichen Zugriffsfunktionen in Abhängigkeit vom Datentyp des Interface Kanals an:

Modbus Ad.	Datentyp	Read Funktion	Write Funktion
0x0000 : 0x003F	Digital	Read Coils (FC 01)	Write Single Coil (FC 05) Write Multiple Coils (FC 15)
0x0000 : 0x007F	Signed 16/32Bit Unsigned 16/32Bit Float	Read Holding Registers (FC 03)	Write Single Register (FC 06) Write Multiple Registers (FC 16)

Die folgende Tabelle gibt die möglichen Zugriffsfunktionen in Abhängigkeit vom Datentyp des Interface Ausgabekanals an:

Modbus Ad.	Datentyp	Read Funktion	Write Funktion
0x0800 : 0x083F	Digital	Read Discrete Inputs (FC 02)	---
0x0800 : 0x087F	Signed 16/32Bit Unsigned 16/32Bit Float	Read Input Registers (FC 04)	---

7.4.6 Technische Details zur RS232-Schnittstelle (Com2)

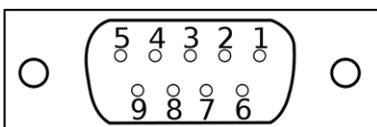
Hinweis: Die Modbus-Schnittstellen Com2 ist kompatibel zur Norm TIA/EIA-232-F.

Wichtiger Hinweis: Die RS232-Schnittstelle des myDatalogMUC xG/4G unterstützt kein Hardware-Handshake.

Die Ausgangstreiber sind gegen Überlastung geschützt und werden durch einen Kurzschluss auf GND oder +/-15V nicht beschädigt. Die Eingänge sind mit einem 5kΩ Abschlusswiderstand versehen.

Baudrate	300-115200
Stopbits	1, 2
Parität	N, E, O
Datenbits	7, 8
Modi	ASCII

Die Richtung der Signale entspricht jener eines DCE (z.B. Modem).



9pol. Sub-D(f)

Belegung des Sub-D Steckers

Pin	Signal	Typ
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) Das Hardware-Handshake wird von der aktuellen Firmware nicht unterstützt.

2) Versorgungsspannung (reserviert für Erweiterungen)

Sollte sich an Ihrem Sensor ebenfalls eine SUB-D(f) Buchse befinden, können Sie den als Zubehör erhältlichen Gender changer 9pol. D-Sub male/male (206.684) verwenden. Für den Fall, dass die Anschlüsseigenschaften (Übertragungsrichtung der einzelnen Signalleitungen) Ihres Sensors ebenfalls jener eines DCE (z.B. Modem) entsprechen, können Sie den als Zubehör erhältlichen Nullmodemadapter 9pol. D-Sub female/male (206.686) verwenden.

7.4.6.1 Fehlercodes der RS232-Schnittstelle

Log-Eintrag		Parameter		Beschreibung
Code	Klartext	Code	Klartext	
2000	MODULE ERR	0	CHAR TIMEOUT	Der zeitliche Abstand zwischen zwei empfangenen Zeichen ist größer als der über die Eingabemaske zur Konfiguration der Schnittstelle (siehe "Konfig" auf Seite 85) eingegebene Zeichen-Timeout.
		1	FRAME TIMEOUT	Innerhalb des Datensatz-Timeouts wurde nicht die komplette Antwort empfangen. Diese Zeitspanne wird über die Eingabemaske zur Konfiguration der Schnittstelle (siehe "Com2 (Seriell, RS232)" auf Seite 82) eingegebenen.
		2	INVALID FRAME	Der Antwort-Frame des Sensors ist ungültig, beispielsweise wenn die Spaltenanzahl des Sensors nicht mit der über die Konfigurationsoberfläche eingegebenen übereinstimmt (siehe "Spalte" im Kapitel "Konfig." auf Seite 109).

7.4.7 Technische Details zur USB-Schnittstelle

Über die USB-Slave-Schnittstelle wird die Verbindung zu einem PC hergestellt. Dabei ist ausschließlich die Kommunikation mit dem Konfigurationsprogramm DeviceConfig vorgesehen. Eine detaillierte Beschreibung des Konfigurationsprogramms DeviceConfig finden Sie im Handbuch zum DeviceConfig ("Benutzerhandbuch für DeviceConfig" 206.887). Das Konfigurationsprogramm DeviceConfig steht unter folgender Adresse gratis zum Download bereit:

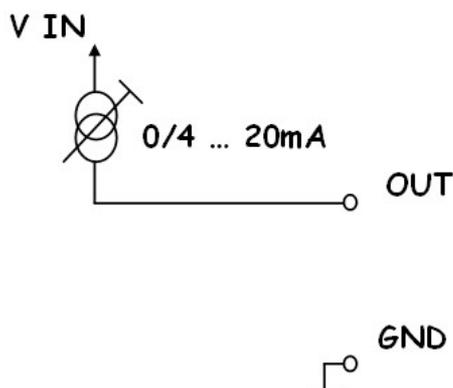
www.microtronics.com/deviceconfig

Wichtiger Hinweis: Sollte die Antenne des Geräts geerdet sein oder mit dem Massepotential eines anderen Objektes verbunden sein (z.B. Montage an einem Schaltschrank), entfernen Sie die Antennen bevor Sie das Gerät mit der USB-Schnittstelle eines PCs verbinden. Andernfalls kann es zu einer Potenzialverschiebung zwischen der Masse der Antenne und der Masse des PCs kommen, wodurch die USB-Schnittstelle des Geräts beschädigt werden kann.

7.4.8 Technische Details zu den Ausgängen

7.4.8.1 Analogausgang 1-2 (OUT, GND)

Hinweis: Bei den Analogausgängen des myDatalogMUC xG/4G handelt es sich um aktive, nicht galvanisch getrennte Stromausgänge.

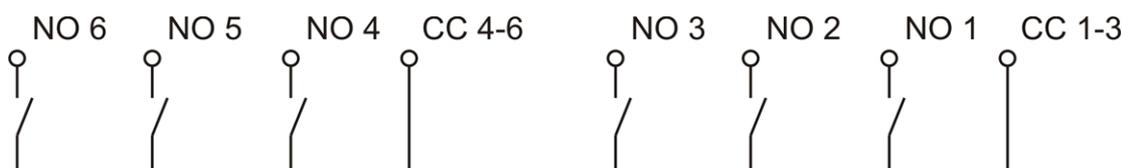


Ersatzschaltbild für den Analogausgang

Bürdenspannung	entspricht der Versorgungsspannung (12...30VDC)
max. Bürde (32V, 20mA)	1200 Ω
$I_{out max}$	20mA

7.4.8.2 Relais 1-6

Je 3 der Relais sind zu einer Gruppe mit gemeinsamer Wurzel zusammengefasst. Im Ruhezustand sind die Arbeitskontakte aller Relais geöffnet (Normally Open).



Ersatzschaltbild für die Relais

U	240VAC
U _{max}	400VAC
I _{max}	3A ¹⁾

¹⁾gilt pro Relais. D.h. über CC fließen bis zu 9A.

7.4.9 Technische Details zum integrierten Pufferakku

Der integrierte Pufferakku ermöglicht es bei Ausfall der Versorgungsspannung eine Meldung abzusetzen. Dazu muss allerdings der Alarm für den internen Messwert "Spannung" (siehe "Interne Kanäle" auf Seite 130) konfiguriert werden. Welcher Wert dabei für die Alarmschwelle "Wert niedrig" eingegeben wird, ist egal, da sobald die Versorgungsspannung unter 9,5V fällt, der interne Messwert "Spannung" auf 0 gesetzt und die Versorgung auf den Pufferakku umgeschaltet wird. Dadurch wird das Absetzen der Ausfallmeldung angestoßen - sofern nicht über den Parameter "Verzögerung für Ausfallsalarm", der sich im Konfigurationsabschnitt "Grundeinstellungen" (siehe "Grundeinstellung" auf Seite 133) befindet, eine Verzögerungszeit fürs Absetzen der Ausfallmeldung eingegeben wurde. Auf jeden Fall wird das Umschalten auf den Pufferakku durch den Log-Eintrag "BACKUP SUPPLY, 1" in das Gerätelog eingetragen. Wurde eine Verzögerungszeit konfiguriert, arbeitet das myDatalogMUC xG/4G, abgesehen von den Baugruppen, die bei Versorgung über den Pufferakku ausfallen, bis zum Ablauf der Verzögerungszeit normal weiter. Steigt während der Verzögerungszeit die Versorgungsspannung wieder über 10,5V, wird wieder vom Pufferakku auf die Versorgungsspannung umgeschaltet. Dies wird durch den Log-Eintrag "BACKUP SUPPLY, 0" in das Gerätelog eingetragen. Liegt die Versorgungsspannung nach Ablauf der Verzögerungszeit unter 10,5V, wird das Absetzen der Ausfallmeldung angestoßen.

Sobald das myDatalogMUC xG/4G ausschließlich vom Pufferakku versorgt wird, fallen folgende Baugruppen aus:

- Analogausgang 1-2
- LEDs zur Anzeige der Schaltzustände der Relais
- Ansteuerung der Relais (d.h. die Arbeitskontakte gehen in den Ruhezustand "NO" über)
- 5V Versorgung an COM2

Während das Gerät die Verbindung zum Server herstellt und aufrecht erhält, um die Ausfallmeldung abzusetzen, leuchtet die LED zur Signalisierung, dass das Gerät vom Pufferakku versorgt wird. Dabei werden auch alle noch ausstehenden Daten zum Server übertragen. Danach wird der Pufferakku deaktiviert und die entsprechende LED geht aus. Sollte die Versorgungsspannung zu dem Zeitpunkt noch nicht vollständig zusammengebrochen sein, läuft am Gerät nur mehr die Überwachung der Versorgungsspannung. Alle anderen Operationen werden eingestellt. Dieser Zustand wird solange beibehalten, bis die Versorgung vollständig ausfällt oder wieder über 10,5V steigt. Über 10,5V nimmt das myDatalogMUC xG/4G den normalen Betrieb wieder auf.

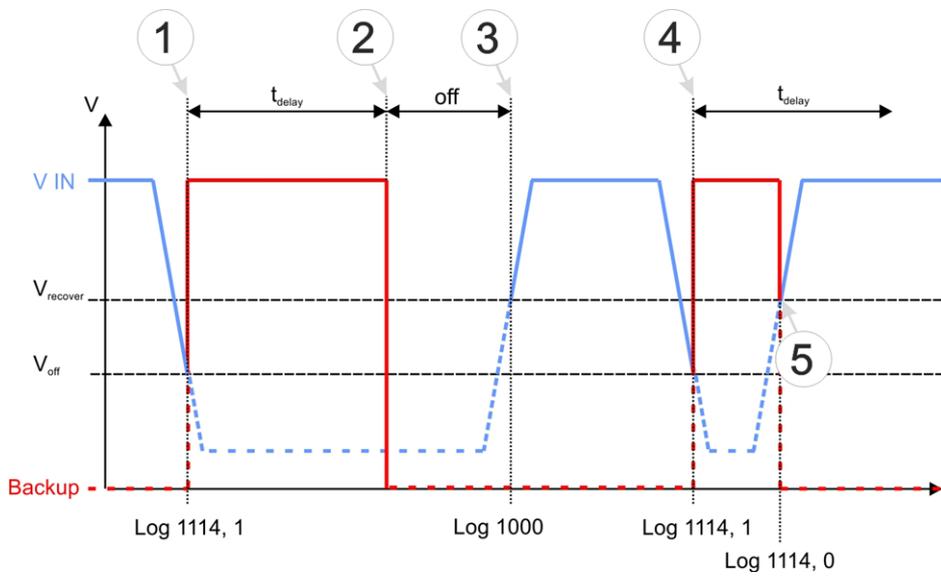
Geräte, die den myDatenet-Server über den Ausfall ihrer Versorgungsspannung informiert haben, sind am unten abgebildetem Symbol in der Messstellenliste erkennbar.



19.7.2012 15:02:41
SER UTC+2,00

Ein durch Hardware realisierter Regler sorgt dafür, dass der Pufferakku nur geladen wird, wenn die Umgebungstemperatur den zulässigen Bereich (0...+40°C) nicht verletzt.

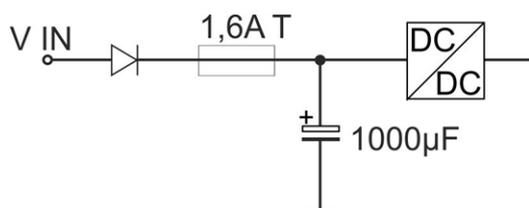
V _{IN}	Versorgungsspannung: 12...30VDC (+/-10%)
V _{recover}	Schwelle für das Umschalten auf den Normalbetrieb: 10,5V
V _{off}	Schwelle für das Umschalten auf die Versorgung über den Pufferakku: 9,5V
t _{delay}	Verzögerungszeit fürs Absetzen der Ausfallmeldung. Wird über den Parameter "Verzögerung für Ausfallsalarm"; der sich im Konfigurationsabschnitt "Grundeinstellungen" (siehe "Grundeinstellung" auf Seite 133) befindet, konfiguriert.
Backup	Spannung des Pufferakku



Versorgung des myDatalogMUC xG/4G

1	<ul style="list-style-type: none"> Die Versorgung wird auf den Pufferakku umgeschaltet. Der Log-Eintrag "BACKUP SUPPLY, 1" wird erstellt. Das Gerät arbeitet normal weiter (abgesehen von den Baugruppen, die bei Versorgung über den Pufferakku ausfallen).
2	<ul style="list-style-type: none"> Die Verzögerungszeit für das Absetzen der Ausfallmeldung ist abgelaufen. Die Ausfallmeldung wird abgesetzt. Danach wird der Pufferakku deaktiviert. In Folge dessen ist das myDatalogMUC xG/4G ausgeschaltet.
3	<ul style="list-style-type: none"> Die Versorgungsspannung überschreitet wieder die Schwelle für das Umschalten auf den Normalbetrieb. Der Log-Eintrag "POWER ON" wird erstellt. Das myDatalogMUC xG/4G nimmt den normalen Betrieb wieder auf.
4	<ul style="list-style-type: none"> Die Versorgung wird auf den Pufferakku umgeschaltet. Der Log-Eintrag "BACKUP SUPPLY, 1" wird erstellt. Das Gerät arbeitet normal weiter (abgesehen von den Baugruppen, die bei Versorgung über den Pufferakku ausfallen).
5	<ul style="list-style-type: none"> Die Versorgungsspannung überschreitet wieder die Schwelle für das Umschalten auf den Normalbetrieb. Die Verzögerungszeit für das Absetzen der Ausfallmeldung ist noch nicht abgelaufen. Der Log-Eintrag "BACKUP SUPPLY, 0" wird erstellt. Es wird KEINE Ausfallmeldung abgesetzt.

7.4.10 Technische Details zur Energieversorgung



Prinzipschaltbild der Energieversorgung

V IN	12...30VDC (+/-10%)
Leistungsaufnahme ¹⁾ (ohne Sensoren)	typ. 1W max. 3W
Eingangskapazität	1000µF
Sicherung	1,6A T
Verpolungsschutz	Ja

¹⁾ gilt für den laufenden Betrieb. Durch die Eingangskapazität tritt zum Einschaltzeitpunkt eine Stromspitze auf.

Um im Falle eines Versorgungsspannungsausfalls zuverlässig auf den integrierten Pufferakku umschalten zu können, ist das myDatalogMUC xG/4G mit einer relativ großen Eingangskapazität (1000µF) ausgestattet. Bitte beachten Sie bei der Auswahl des Netzteils, dass dieses in der Lage ist, den nötigen Anlaufstrom zu liefern. Eine Auswahl an kompatiblen Netzteilen finden Sie im Kapitel "Versorgung" auf Seite 225. Der Versorgungsspannungseingang ist des Weiteren mit einer Diode zum Schutz vor Verpolung sowie einer 1,6A T Sicherung ausgestattet.

Kapitel 8 Inbetriebnahme

8.1 Hinweise an den Benutzer

Bevor Sie das myDatalogMUC xG/4G anschließen und in Betrieb nehmen, sind die folgenden Benutzerhinweise unbedingt zu beachten!

Dieses Handbuch enthält alle Informationen, die zum Gebrauch des Gerätes erforderlich sind.

Es wendet sich an technisch qualifiziertes Personal, welches über einschlägiges Wissen im Bereich der Messtechnik verfügt.

Um die einwandfreie Funktion des myDatalogMUC xG/4G zu gewährleisten, muss dieses Handbuch sorgfältig gelesen werden.

Bei eventuellen Unklarheiten oder Schwierigkeiten in Bezug auf Montage, Anschluss oder Konfiguration wenden Sie sich an Microtronics Engineering GmbH (siehe "Kontaktinformationen" auf Seite 231).

8.2 Mitgeltende Unterlagen

Für die Installation, Inbetriebnahme und den Betrieb des Gesamtsystems werden neben dieser Bedienungsanleitung möglicherweise zusätzliche Anleitungen oder technische Beschreibungen benötigt.

Diese Anleitungen liegen den jeweiligen Zusatzgeräten oder Sensoren bei bzw. stehen auf der Microtronics - Webseite zum Download bereit.

8.3 Allgemeine Grundsätze

Die Inbetriebnahme des gesamten Messsystems darf erst nach Fertigstellung und Prüfung der Installation erfolgen. Vor der Inbetriebnahme ist das Studium des Handbuches erforderlich, um fehlerhafte oder falsche Konfiguration auszuschließen.

Machen Sie sich mit Hilfe des Handbuches mit der Bedienung des myDatalogMUC xG/4G und den Eingabemasken des myDatamet-Servers vertraut, bevor Sie mit der Konfiguration beginnen.

8.4 Inbetriebnahme des Systems

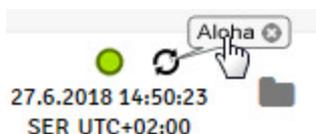
***Hinweis:** Es empfiehlt sich, das myDatalogMUC xG/4G zuerst im Büro in Betrieb zu nehmen bevor Sie das Gerät am Einsatzort fix montieren. Dabei sollten Sie gleich eine Messstelle für den späteren Betrieb am myDatamet-Server anlegen (siehe "Benutzerhandbuch für myDatamet-Server" 206.886) und eine Messstellenkonfiguration festlegen (siehe "Messstellenkonfiguration" auf Seite 74). Nutzen Sie die Gelegenheit sich in geordneter Umgebung mit den Funktionen des Geräts vertraut zu machen. Sie können auch geeignete Testsignale zum Simulieren der Sensoren verwenden, um die Konfiguration des myDatalogMUC xG/4G bereits vor der eigentlichen Inbetriebnahme optimal fest zu legen. Dadurch reduzieren Sie den Zeitaufwand bei der Installation vor Ort auf das Minimum.*

1. Haben Sie alle im Kapitel "Anschluss der Sensoren, der Aktoren und der Versorgung" auf Seite 52 beschriebenen Schritte durchgeführt, ist das myDatalogMUC xG/4G bereits betriebsbereit und sollte bereits die erste Verbindung zum myDatamet-Server durchgeführt haben.

2. Legen Sie eine Messstelle für den Betrieb am myDatanet-Server an (siehe "Benutzerhandbuch für myDatanet-Server " 206.886).
3. Konfigurieren Sie die erstellte Messstelle entsprechend Ihren Anforderungen (siehe "Messstellenkonfiguration" auf Seite 74).
4. Verknüpfen Sie das myDatalogMUC xG/4G mit der erstellten Messstelle (siehe "Site" auf Seite 75).
5. Lösen Sie den Aloha-Übertragungsmodus (siehe "Aloha-Übertragungsmodus" auf Seite 41) aus, damit die Konfiguration der Messstelle zum myDatalogMUC xG/4G übertragen wird.

8.5 Kommunikation mit dem Gerät testen

1. Legen Sie eine Messstelle für den Betrieb am myDatanet-Server an (siehe "Benutzerhandbuch für myDatanet-Server " 206.886).
2. Konfigurieren Sie die erstellte Messstelle entsprechend Ihren Anforderungen (siehe "Messstellenkonfiguration" auf Seite 74).
3. Verknüpfen Sie das myDatalogMUC xG/4G mit der erstellten Messstelle (siehe "Site" auf Seite 75).
4. Lösen Sie den Aloha-Übertragungsmodus aus (siehe "Aloha-Übertragungsmodus" auf Seite 41), damit die Konfiguration der Messstelle zum myDatalogMUC xG/4G übertragen wird.
5. Warten Sie bis in der Messgeräteleiste angezeigt wird, dass sich das Gerät im Aloha-Übertragungsmodus befindet. Angezeigt wird dieser durch eine Sprechblase mit der Beschriftung "Aloha".

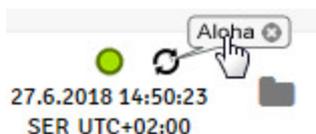


Die folgenden Schritte sind nur erforderlich, wenn Sie auch gleich die Messwerterfassung und die Datenübertragung testen wollen.

6. Beenden Sie den Aloha-Übertragungsmodus durch Klicken auf das Kreuz in der Sprechblase mit der Beschriftung "Aloha" oder warten Sie die Dauer des Aloha-Übertragungsmodus ab. Diese Dauer kann in den Grundeinstellungen (siehe "Grundeinstellung" auf Seite 133) der Messstellenkonfiguration festgelegt werden. Die Standardeinstellung ist 10min.
7. Verdrahten Sie anschließend die Sensoren (siehe "Anschluss der Sensoren, der Aktoren und der Versorgung" auf Seite 52) und starten Sie danach erneut den Aloha-Übertragungsmodus.

Wichtiger Hinweis: Alle Verkabelungsarbeiten sollten im stromlosen Zustand erfolgen!

8. Prüfen Sie die ankommenden Daten im Aloha-Datenfenster des myDatanet-Servers, welches Sie durch einen Klick auf die Sprechblase mit der Beschriftung "Aloha" erreichen (siehe "Benutzerhandbuch für myDatanet-Server " 206.886). Spezielles Augenmerk sollten Sie auf die internen Messwerte "GSM Stärke" und "Spannung" legen.



Hinweis: Ergänzende Erklärung zur Bewertung der "GSM Stärke":

"GSM Stärke"	
> -64dBm	
-64...-73dBm	
-74...-83dBm	
-84...-93dBm	
-94...-107dBm	
<= -108dBm	

Hinweis: Ergänzende Erklärung zur Bewertung der "Spannung":

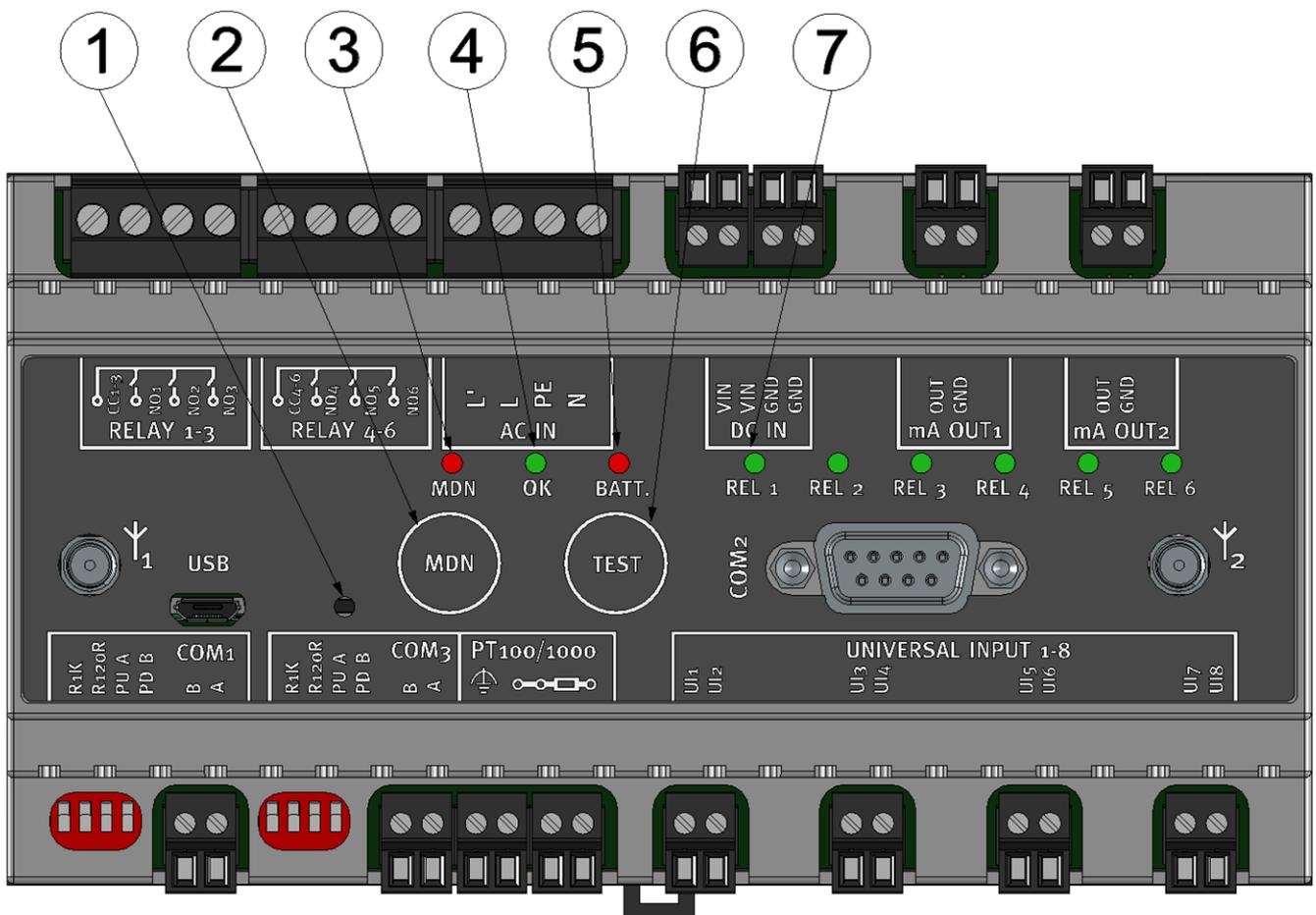
Die angezeigte Spannung sollte nur geringfügig von der an das Gerät angelegten Versorgungsspannung abweichen.

Kapitel 9 Benutzerschnittstellen

Die Konfiguration des myDatalogMUC xG/4G erfolgt über das Web-Interface am myDatenet-Server (siehe "Benutzerschnittstelle am myDatenet-Server" auf Seite 74), dessen Web-Adresse Sie von Ihrem zuständigen Vertriebspartner erhalten.

9.1 Benutzerschnittstelle am myDatalogMUC xG/4G

9.1.1 Bedienelemente



Bedienelemente

1	Taste zum Auslösen eines Resets	5	Statusanzeige: Pufferakku aktiv
2	Taste zum Auslösen des Aloha-Übertragungsmodus	6	Taste zum Auslösen des Selbsttests
3	Status-LED	7	Statusanzeige: Schaltzustände der Relais
4	Statusanzeige: Selbsttests		

9.1.1.1 Taste zum Auslösen des Aloha-Übertragungsmodus

Die Taste kann dazu verwendet werden, den Aloha-Übertragungsmodus auszulösen, oder das myDatalogMUC xG/4G anzuweisen, den Fehler/Status-Code sofort auszugeben.

Useraktion	Reaktion des Geräts	Operation nach Loslassen der Taste
kurz drücken ca. 1 sec.	Status-LED geht an	Ausgabe des Fehler/Status-Codes (siehe "Status-LED" auf Seite 72)
drücken und 5 sec. halten	Status-LED blinkt 3x und bleibt dann weiter an	Aloha-Übertragungsmodus

9.1.1.2 Status-LED

Die Status-LED dient sowohl der Anzeige der Fehler-/Status-Codes als auch der Signalisierung des aktuellen Betriebszustandes. Wurde der Aloha-Übertragungsmodus aktiviert oder die Energieversorgung hergestellt (PowerOn), zeigt die Status-LED für 10min. den aktuellen Betriebszustand an. In diesen 10min. werden die Fehler-/Status-Codes alle 3sec. ausgegeben sofern nicht eine aktive GPRS-Verbindung besteht.

Fehler-/Status-Codes

Blinkcode	Beschreibung	Lösung/Ursache
0x	Transportsperre (GPRS aus, Messung aus)	Wird der Aloha-Übertragungsmodus mittels Taste ausgelöst, schaltet der myDatalogMUC xG/4G wieder in den Modus "RUN" (GPRS an, Messung an).
1x	letzte Verbindung o.k.	---
2x	letzte Übertragung fehlerhaft	später erneut versuchen
4x	Standby (GPRS an, Messung aus)	siehe "Transportsperre"
6x	Offline (GPRS aus, Messung an)	siehe "Transportsperre"
7x	Netzsperrung/kein passender Provider	<ul style="list-style-type: none"> • Antennenpositionierung verbessern • überprüfen, ob sich das Gerät im Versorgungsbereich eines der Serviceprovider, die vom integrierten SIM-Chip unterstützt werden, befindet (www.microtronics.com/footprint)
8x	kein GSM-Netzwerk	<ul style="list-style-type: none"> • später erneut versuchen • Antennenpositionierung verbessern
10x	keine GPRS-Verbindung	Antennenpositionierung verbessern
11x	kein myDatatnet-Server erreichbar	<ul style="list-style-type: none"> • überprüfen, ob am myDatatnet-Server der Port 51241 frei geschaltet ist • später erneut versuchen
12x	fehlerhafter SIM-Chip	Support kontaktieren

Betriebszustände

Status-LED	Beschreibung
flackert	Verbindungsaufbau
leuchtet	GPRS-Verbindung hergestellt oder Taste gedrückt
aus	normaler Messbetrieb laut Konfiguration bis zur nächsten Übertragung

9.1.1.3 Statusanzeige: Selbsttests

Diese Statusanzeige dient sowohl der Signalisierung eines laufenden Selbsttests als auch der Anzeige des Ergebnisses des Tests. Bei Herstellen der Energieversorgung (PowerOn) wird der Selbsttest automatisch durchgeführt. Er kann aber auch jederzeit durch Drücken der Taste mit der Beschriftung "TEST" (siehe "Taste zum Auslösen des Selbsttests" auf Seite 74) vom Benutzer ausgelöst werden. Überprüft werden dabei die Analogausgänge und die Universaleingänge. Die Verkabelung der Sensoren muss nicht entfernt werden.

TEST/OK	Beschreibung
blinkt	Selbsttest läuft
leuchtet	Selbsttest erfolgreich, kein Hardwaredefekt festgestellt
aus	Hardwaredefekt festgestellt

Wurde ein Hardwaredefekt gefunden, wird im Gerätelog ein entsprechender Eintrag erzeugt. Mit Hilfe des Parameters, der zum Fehlercode gespeichert wurde, lässt sich die Ursache des Problems genauer eingrenzen. Eine Anleitung zum Auswerten des Gerätelogs finden Sie im Kapitel „Auswerten des Gerätelogs“ (siehe "Auswerten des Gerätelogs" auf Seite 224).

Log-Eintrag		Parameter		Beschreibung
Code	Klartext	Code	Klartext	
1020	ERROR SOD	## ¹⁾	---	Hardwaredefekt gefunden

¹⁾ Die folgende Tabelle gibt die Codierung des Parameters an:

Bit 0	Analogausgang 1	Bit 4	Universaleingang 1
Bit 1	Analogausgang 2	Bit 5	Universaleingang 2
Bit 2	Externer Temperatursensor	Bit 6	Universaleingang 3
Bit 3	Referenzspannung der Universaleingänge	Bit 7	Universaleingang 4
		Bit 8	Universaleingang 5
		Bit 9	Universaleingang 6
		Bit 10	Universaleingang 7
		Bit 11	Universaleingang 8

Hinweis: Beispiel zur Identifizierung der Ursache eines Hardwaredefekts:

Annahme: Das Gerätelog enthält:

Log-Eintrag	Parameter
1020	4088

Der Parameter muss von der Dezimaldarstellung (4088) in die Binärdarstellung (1111 1111 1000) umgewandelt werden. Nun ist ersichtlich, dass Bit3 bis Bit11 gesetzt sind. D.h. in diesem Beispiel wären die Referenzspannung und alle Universaleingänge defekt.

9.1.1.4 Statusanzeige: Pufferakku aktiv

Wichtiger Hinweis: Bitte beachten Sie, dass wenn das Gerät seine Energie aus dem Pufferakku bezieht, nicht mehr alle Hardwarekomponenten versorgt werden (siehe "Technische Details zum integrierten Pufferakku" auf Seite 63).

ON BATT.	Beschreibung
leuchtet	Das myDatalogMUC xG/4G bezieht seine Energie aus dem Pufferakku.
aus	Das myDatalogMUC xG/4G wird über die Klemmen V IN und GND gespeist.

9.1.1.5 Taste zum Auslösen des Selbsttests

Die Taste ermöglicht es dem Benutzer den Selbsttest zu jedem beliebigen Zeitpunkt auszulösen.

Useraktion	Reaktion des Geräts
drücken und 5sec. halten	LED "TEST/OK" beginnt zu blinken und der Selbsttest wird durchgeführt.

Nähre Details zum Selbsttest und der Diagnose bei Problemen finden Sie unter "Statusanzeige: Selbsttests" auf Seite 73.

9.1.1.6 Statusanzeige: Schaltzustände der Relais

Das myDatalogMUC xG/4G ist mit LEDs ausgestattet, die den Schaltzustand eines jeden der 6 Relais anzeigen.

RELAY x	Beschreibung
leuchtet	Der Arbeitskontakt des betreffenden Relais ist geschlossen.
aus	Der Arbeitskontakt des betreffenden Relais ist geöffnet (Ruhezustand).

9.2 Benutzerschnittstelle am myDatanet-Server

9.2.1 Messstellenkonfiguration

Hinweis: Abhängig vom jeweiligen Benutzerlevel sind einige der in den folgenden Unterkapiteln erwähnten Konfigurationsfelder unter Umständen ausgeblendet. Wenden Sie sich in diesem Fall an den Administrator des myDatanet-Servers.

Die Eingabemaske zur Konfiguration der Messstelle erreichen Sie durch Klicken auf den Messstellennamen in der Messstellenliste (siehe "Benutzerhandbuch für myDatanet-Server " 206.886).

9.2.1.1 Site

Kunde

gibt an, welchem Kunden die Messstelle zugeordnet ist



-Symbol

Messstelle einem anderen Kunden zuweisen

Name

Messstellenbezeichnung (nicht relevant für die Geräte- oder Datenzuordnung) [2-50 Zeichen]

Gerät S/N

Seriennummer des Geräts, das mit der Messstelle verknüpft ist (Gerätezuordnung!)

Applikations-Vorlage

Name der Applikations-Vorlage aus der die Messstelle erstellt wurde

Tags

Liste der Tags, die der Messstelle bereits zugewiesen sind. Durch einen Klick auf das Kreuz neben der Bezeichnung des Tags kann diese Zuweisung wieder aufgehoben werden. Die Eingabemaske zur Zuweisung der Tags kann durch Klicken auf das Plus-Symbol geöffnet werden.

9.2.1.2 Kommentar

Kommentar

freies Kommentarfeld (wird auch unterhalb des Gerätetyps in der Messstellenliste angezeigt)

9.2.1.3 Steuerung

Device Logic Typ

Auswahl des Device Logic Typs

<i>aus</i>	<i>Device Logic deaktiviert</i>		
<i>IL</i>	<i>aktiviert die Device Logic-Abarbeitung und teilt dem myDatalogMUC xG/4G mit, dass es sich bei der Device Logic um eine „Instruction List“ handelt</i>		
	<i>Device Logic</i>	<i>Eingabefenster zum Editieren der Device Logic, die in das Gerät geladen wird</i>	
<i>Pawn</i>	<i>aktiviert die Device Logic-Abarbeitung und teilt dem myDatalogMUC xG/4G mit, dass es sich bei der Device Logic um ein „Pawn Script“ handelt (siehe "Device Logic (Pawn)" auf Seite 147)</i>		
	<i>Device Logic Quelle</i>	<i>Pawn source code</i>	<i>Die Device Logic wird direkt über die Serveroberfläche eingegeben und kompiliert.</i>
		<i>Device Logic</i>	<i>Eingabefenster zum Editieren der Device Logic, die in das Gerät geladen wird</i>
	<i>Hochladen einer kompilierten Device Logic (Pawn, *.amx)</i>	<i>Ein bereits kompiliertes Device Logic Binary-File (*.amx) soll hochgeladen werden.</i>	
<i>Datei hochladen</i>		<i>Auswahl des Device Logic Binary-Files (*.amx), das auf den myDatanet-Server hochgeladen und bei der nächsten Verbindung in das myDatalogMUC xG/4G geladen wird. Der Dateipfad wird nur solange angezeigt, solange die Eingabemaske zur Konfiguration der Messstelle nicht geschlossen wurde.</i>	

9.2.1.4 Schnittstellen

9.2.1.4.1 Basis

Hinweis: Die Auswahl des Modbus-Modus (Master/Slave) für die Schnittstellen Com1 und Com3 erfolgt im Tab "Konfig" (siehe "Konfig" auf Seite 85).

9.2.1.4.1.1 Com1 und Com3 (Modbus-Master, RS485, Device Logic Parsing inaktiv)

Modus

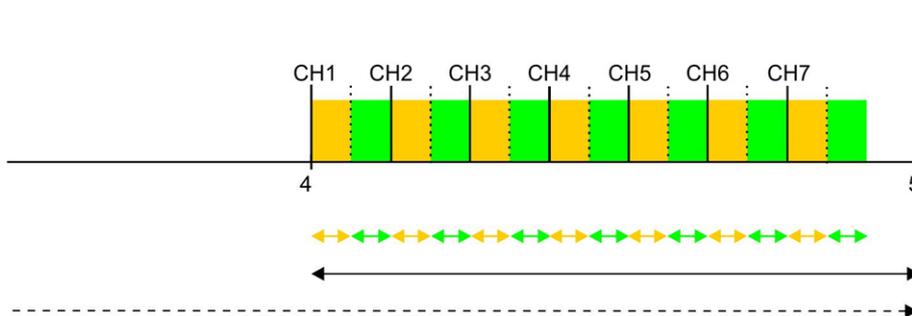
<i>aus</i>	<i>Schnittstelle deaktiviert</i>		
<i>Modbus</i>	<i>aktiviert die Modbus-Funktionalität der Schnittstelle</i>		
<i>(1/2)</i>	<i>Baudrate</i>	300	<i>Auswahl der benötigten Baudrate</i>
		600	
		1200	
		2400	
		4800	
		9600	
		19200	
		38400	
		57600	
		115200	
<i>Stoppbits</i>	1	<i>Auswahl der Anzahl der benötigten Stoppbits</i>	
	2		
<i>Parität</i>	<i>keine</i>	<i>Auswahl der benötigten Parität</i>	
	<i>ungerade</i>		
	<i>gerade</i>		
<i>Datenbits</i>	7	<i>Anzahl der zu verwendenden Datenbits</i>	
	8		
<i>Antwort timeout</i>	<i>Zeit innerhalb der der Modbus-Slave auf das Kommando des Geräts reagieren muss</i>		
<i>Halten</i>	<i>Halten des letztgültigen Messwerts für x Messzyklen</i>		
	<i>aus</i>	<i>Funktion deaktiviert</i>	
	1-5	<i>Anzahl der Messzyklen, für die der Messwert gehalten wird, bevor der Fehlerwert ausgegeben wird</i>	
	<i>ein</i>	<i>Im Fehlerfall wird der zuletzt gültige Messwert so lange gehalten, bis wieder ein neuer gültiger Messwert vorliegt.</i>	

Modbus (2/2)	Retry	Bei einem Kommunikationsfehler wird das entsprechende Kommando ein Mal wiederholt. Der Fehler wird erst aufgezeigt, wenn auch dieser zweite Versuch fehlschlägt.
Device Logic Parsing	<p>Aktiviert das Device Logic Parsing. Dadurch kann auf die Schnittstelle mittels der PAWN-Device Logic Funktionen "Mdn_SerialEvent()", "Mdn_SerialRx()", "Mdn_SerialTx()" und "Mdn_SerialFinish()" zugegriffen werden (siehe "Device Logic Parsing" auf Seite 43). Die Modbus-Funktionalität der Schnittstelle wird dadurch deaktiviert.</p> <p>Details zur Konfiguration des "Device Logic Parsing"-Modus finden Sie unter "Com1 und Com3 (RS485, Device Logic Parsing aktiv)" auf Seite 79.</p>	

Wichtiger Hinweis:

Beispiel zur Erklärung des Zusammenhangs zwischen "Antwort timeout", "Retry" und "Messintervall"

Grundeinstellung		Aufzeichnungsintervall	5min.
		Messintervall	1min.
Schnittstellen -> Basis	1ter Versuch	Antwort timeout	4sec.
	Retry		
		Retry	aktiv



Erklärung: In diesem Beispiel sind 7 Kanäle aktiviert, der Modbus-Slave antwortet jedoch nicht. Zum Messzeitpunkt wird zuerst versucht die Daten für Interface Kanal 1 zu lesen. Da der "Retry" aktiviert ist, wird nach Ablauf des "Antwort timeout" ein weiteres Mal versucht die Daten für Interface Kanal 1 vom Modbus-Slave zu lesen. Nach erneutem Ablauf des "Antwort timeout" wird für Interface Kanal 1 der Fehlerwert "OL" (Open Loop) gesetzt und der erste Versuch, die Daten für Interface Kanal 2 zu lesen, gestartet.

Daher müssen die Anzahl der aktivierten Kanäle (Interface Kanäle + Interface Ausgabekanäle), das "Antwort timeout", "Retry" und das Messintervall folgendermaßen gewählt werden:

Retry nicht aktiv: "Antwort timeout" * Anzahl aktiver Kanäle < "Messintervall"

Retry aktiv: "Antwort timeout" * 2 * Anzahl aktiver Kanäle < "Messintervall"

9.2.1.4.1.2 Com1 und Com3 (Modbus-Slave, RS485, Device Logic Parsing inaktiv)

Modus

<i>aus</i>	<i>Schnittstelle deaktiviert</i>	
<i>Modbus</i>	<i>aktiviert die Modbus-Funktionalität der Schnittstelle</i>	
<i>Baudrate</i>	<i>300</i>	<i>Auswahl der benötigten Baudrate</i>
	<i>600</i>	
	<i>1200</i>	
	<i>2400</i>	
	<i>4800</i>	
	<i>9600</i>	
	<i>19200</i>	
	<i>38400</i>	
	<i>57600</i>	
	<i>115200</i>	
<i>Stoppbits</i>	<i>1</i>	<i>Auswahl der Anzahl der benötigten Stoppbits</i>
	<i>2</i>	
<i>Parität</i>	<i>keine</i>	<i>Auswahl der benötigten Parität</i>
	<i>ungerade</i>	
	<i>gerade</i>	
<i>Datenbits</i>	<i>7</i>	<i>Anzahl der zu verwendenden Datenbits</i>
	<i>8</i>	
<i>Überwachung Timeout</i>	<i>0...keine Überwachung</i> <i>Zeit innerhalb der die Register durch den Modbus-Master erneuert werden müssen. Bei einem Timeout werden die Messwerte als "OL" (Open Loop) gekennzeichnet.</i>	
<i>Halten</i>	<i>Halten des letztgültigen Messwerts für x Messzyklen</i>	
	<i>aus</i>	<i>Funktion deaktiviert</i>
	<i>1-5</i>	<i>Anzahl der Messzyklen, für die der Messwert gehalten wird, bevor der Fehlerwert ausgegeben wird</i>
	<i>ein</i>	<i>Im Fehlerfall wird der zuletzt gültige Messwert so lange gehalten, bis wieder ein neuer gültiger Messwert vorliegt.</i>
<i>Device Logic Parsing</i>	<i>Aktiviert das Device Logic Parsing. Dadurch kann auf die Schnittstelle mittels der PAWN-Device Logic Funktionen "Mdn_SerialEvent()", "Mdn_SerialRx()", "Mdn_SerialTx()" und "Mdn_SerialFinish()" zugegriffen werden (siehe "Device Logic Parsing" auf Seite 43). Die Modbus-Funktionalität der Schnittstelle wird dadurch deaktiviert.</i> <i>Details zur Konfiguration des "Device Logic Parsing"-Modus finden Sie unter "Com1 und Com3 (RS485, Device Logic Parsing aktiv)" auf Seite 79.</i>	

9.2.1.4.1.3 Com1 und Com3 (RS485, Device Logic Parsing aktiv)

Modus

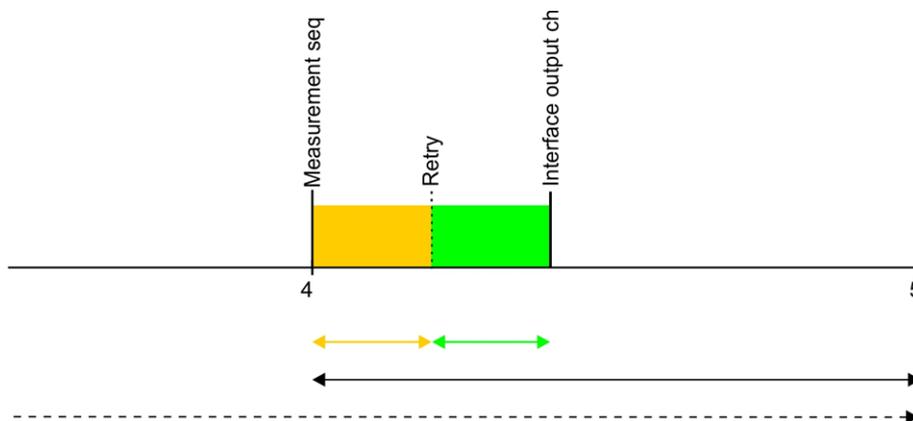
<i>aus</i>	<i>Schnittstelle deaktiviert</i>	
<i>Modbus</i>	<i>aktiviert die Modbus-Funktionalität der Schnittstelle</i> <i>Details zur Konfiguration des "Modbus"-Modus finden Sie unter "Basis" auf Seite 76 bzw. "Com1 und Com3 (Modbus-Slave, RS485, Device Logic Parsing inaktiv)" auf Seite 78.</i>	
<i>Device Logic Parsing (1/2)</i>	<i>Aktiviert das Device Logic Parsing. Dadurch kann auf die Schnittstelle mittels der PAWN-Device Logic Funktionen "Mdn_SerialEvent()", "Mdn_SerialRx()", "Mdn_SerialTx()" und "Mdn_SerialFinish()" zugegriffen werden (siehe "Device Logic Parsing" auf Seite 43). Die Modbus-Funktionalität der Schnittstelle wird dadurch deaktiviert.</i>	
<i>Baudrate</i>	300 600 1200 2400 4800 9600 19200 38400 57600 115200	<i>Auswahl der benötigten Baudrate</i>
<i>Stoppbits</i>	1 2	<i>Auswahl der Anzahl der benötigten Stoppbits</i>
<i>Parität</i>	keine ungerade gerade	<i>Auswahl der benötigten Parität</i>
<i>Datenbits</i>	7 8	<i>Anzahl der zu verwendenden Datenbits</i>
<i>Datensatz Timeout</i>	<i>0...Kein Timeout</i> <i>Zeit, innerhalb der der angeschlossene digitale Sensor die komplette Antwort auf eine Anfrage senden muss. Bei Überschreiten der Zeit wird ein Kommunikationsfehler erkannt.</i>	

<i>Device Logic Parsing</i> (2/2)	<i>Halten</i>	<i>Halten des letztgültigen Messwerts für x Messzyklen</i>	
		<i>aus</i>	<i>Funktion deaktiviert</i>
		<i>1-5</i>	<i>Anzahl der Messzyklen, für die der Messwert gehalten wird, bevor der Fehlerwert ausgegeben wird</i>
	<i>ein</i>	<i>Im Fehlerfall wird der zuletzt gültige Messwert so lange gehalten, bis wieder ein neuer gültiger Messwert vorliegt.</i>	
	<i>Retry</i>	<i>Bei einem Kommunikationsfehler wird das entsprechende Kommando ein Mal wiederholt. Der Fehler wird erst aufgezeigt, wenn auch dieser zweite Versuch fehlschlägt.</i>	

Wichtiger Hinweis:

Beispiel zur Erklärung des Zusammenhangs zwischen "Datensatz Timeout", "Retry" und "Messintervall"

Grundeinstellung		Aufzeichnungsintervall	5min.
		Messintervall	1min.
Schnittstellen -> Basis	1ter Versuch	Datensatz Timeout	12sec.
	Retry		
		Retry	aktiv



Erklärung: Zum Messzeitpunkt wird die Messdaten-Sequenz mittels der Funktion "Mdn_SerialTx()" (siehe "Device Logic Parsing" auf Seite 43) versendet, der Sensor antwortet jedoch nicht. Da der "Retry" aktiviert ist, wird nach Ablauf des "Datensatz Timeout" die Messdaten-Sequenz erneut gesendet. Nach erneutem Ablauf des "Datensatz Timeout" wird für alle Interface Kanäle, die ihre Daten über die entsprechenden Com beziehen, mittels der Funktion "Mdn_SetCh()" (siehe Mdn_SetCh()), der Fehlerwert "OL" (Open Loop) gesetzt. Danach wird der Datensatz, der die Stellwerte der Interface Ausgabekanäle enthält, mittels der Funktion "Mdn_SerialTx()" über die Schnittstelle versendet.

Daher müssen "Datensatz Timeout", "Retry" und das Messintervall folgendermaßen gewählt werden:

Retry nicht aktiv: "Antwort timeout" < "Messintervall"

Retry aktiv: "Antwort timeout" * 2 < "Messintervall"

9.2.1.4.1.4 Com2 (Seriell, RS232)

Modus

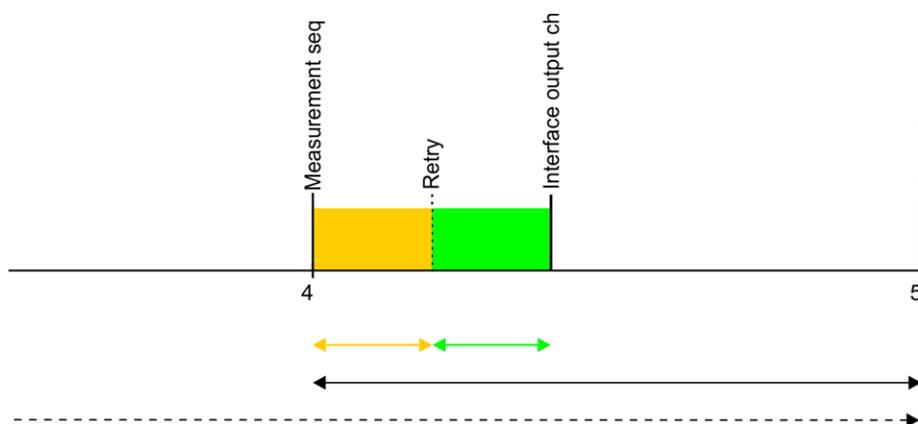
<i>aus</i>	<i>Schnittstelle deaktiviert</i>		
<i>ASCII</i>	<i>Baudrate</i>	<i>300</i>	<i>Auswahl der benötigten Baudrate</i>
		<i>600</i>	
		<i>1200</i>	
		<i>2400</i>	
		<i>4800</i>	
		<i>9600</i>	
		<i>19200</i>	
		<i>38400</i>	
		<i>57600</i>	
		<i>115200</i>	
<i>Stoppbits</i>	<i>1</i>	<i>2</i>	<i>Auswahl der Anzahl der benötigten Stoppbits</i>
<i>Parität</i>	<i>keine</i>	<i>Auswahl der benötigten Parität</i>	
	<i>ungerade</i>		
	<i>gerade</i>		
<i>Datenbits</i>	<i>7</i>	<i>8</i>	<i>Anzahl der zu verwendenden Datenbits</i>
<i>Datensatz Timeout</i>	<i>0...Kein Timeout</i> <i>Zeit, innerhalb der der angeschlossene digitale Sensor die komplette Antwort auf eine Anfrage senden muss. Bei Überschreiten der Zeit wird ein Kommunikationsfehler erkannt.</i>		
<i>Halten</i>	<i>Halten des letztgültigen Messwerts für x Messzyklen</i>		
	<i>aus</i>	<i>Funktion deaktiviert</i>	
	<i>1-5</i>	<i>Anzahl der Messzyklen, für die der Messwert gehalten wird, bevor der Fehlerwert ausgegeben wird</i>	
	<i>ein</i>	<i>Im Fehlerfall wird der zuletzt gültige Messwert so lange gehalten, bis wieder ein neuer gültiger Messwert vorliegt.</i>	
<i>Retry</i>	<i>Bei einem Kommunikationsfehler wird das entsprechende Kommando ein Mal wiederholt. Der Fehler wird erst aufgezeigt, wenn auch dieser zweite Versuch fehlschlägt.</i>		

Device Logic Parsing	<i>Aktiviert das Device Logic Parsing. Dadurch kann auf die Schnittstelle mittels der PAWN-Device Logic Funktionen "Mdn_SerialEvent()", "Mdn_SerialRx()", "Mdn_SerialTx()" und "Mdn_SerialFinish()" zugegriffen werden (siehe "Device Logic Parsing" auf Seite 43). Die Möglichkeit über den Tab "Konfig" des Konfigurationsabschnitts "Schnittstellen" festzulegen wie die empfangenen Zeichen auszuwerten sind, wird dadurch deaktiviert.</i>	
Baudrate	300 600 1200 2400 4800 9600 19200 38400 57600 115200	<i>Auswahl der benötigten Baudrate</i>
Stoppbits	1 2	<i>Auswahl der Anzahl der benötigten Stoppbits</i>
Parität	keine ungerade gerade	<i>Auswahl der benötigten Parität</i>
Datenbits	7 8	<i>Anzahl der zu verwendenden Datenbits</i>
Datensatz Timeout	<i>0...Kein Timeout</i> <i>Zeit, innerhalb der der angeschlossene digitale Sensor die komplette Antwort auf eine Anfrage senden muss. Bei Überschreiten der Zeit wird ein Kommunikationsfehler erkannt.</i>	
Halten	aus 1-5 ein	<i>Funktion deaktiviert</i> <i>Anzahl der Messzyklen, für die der Messwert gehalten wird, bevor der Fehlerwert ausgegeben wird</i> <i>Im Fehlerfall wird der zuletzt gültige Messwert so lange gehalten, bis wieder ein neuer gültiger Messwert vorliegt.</i>
Retry	<i>Bei einem Kommunikationsfehler wird das entsprechende Kommando ein Mal wiederholt. Der Fehler wird erst aufgezeigt, wenn auch dieser zweite Versuch fehlschlägt.</i>	

Wichtiger Hinweis:

Beispiel zur Erklärung des Zusammenhangs zwischen "Datensatz Timeout", "Retry" und "Messintervall"

Grundeinstellung		Aufzeichnungsintervall	5min.
		Messintervall	1min.
Schnittstellen -> Basis	 1ter Versuch	Datensatz Timeout	12sec.
	 Retry		
		Retry	aktiv



Erklärung: Zum Messzeitpunkt wird die Messdaten-Sequenz (siehe "Sequenzen für COM2" auf Seite 86) versendet, der Sensor antwortet jedoch nicht. Da der "Retry" aktiviert ist, wird nach Ablauf des "Datensatz Timeout" die Messdaten-Sequenz erneut gesendet. Nach erneutem Ablauf des "Datensatz Timeout" wird für alle Interface Kanäle, die ihre Daten über die Com2 beziehen, der Fehlerwert "OL" (Open Loop) gesetzt. Danach wird der Datensatz, der die Stellwerte der Interface Ausgabekanäle enthält, versendet.

Daher müssen "Datensatz Timeout", "Retry" und das Messintervall folgendermaßen gewählt werden:

Retry nicht aktiv: "Antwort timeout" < "Messintervall"

Retry aktiv: "Antwort timeout" * 2 < "Messintervall"

9.2.1.4.2 Konfig

9.2.1.4.2.1 Com1 und Com3

Modus

<i>aus</i>	<i>Schnittstelle deaktiviert</i>				
<i>Modbus</i>	<i>Modbus Modus</i>	<i>Master</i>	<i>Com Modus</i>	<i>RTU</i>	<i>Daten werden in binärer Form übertragen.</i>
				<i>ASCII</i>	<i>Daten werden im ASCII-Format übertragen.</i>
	<i>Slave</i>	<i>Com Modus</i>	<i>Com Modus</i>	<i>RTU</i>	<i>Daten werden in binärer Form übertragen.</i>
				<i>ASCII</i>	<i>Daten werden im ASCII-Format übertragen.</i>
			<i>Slave Ad.</i>	<i>Modbus Slave Adresse</i>	
<i>Device Logic Parsing</i>	---				

9.2.1.4.2.2 Com2

Modus

aus	Schnittstelle deaktiviert		
ASCII	Aufwärmzeit	Zeit vom Einschalten des myDatalogMUC xG/4G bis zum Senden der Init Sequenz.	
	Datensatz Struktur	...[CR][LF]	Jeder Frame muss mit den beiden Zeichen „[CR][LF]“ beendet werden.
		...[ETX]	Jeder Frame muss mit dem Zeichen „[ETX]“ beendet werden.
		[STX]...[CR][LF]	Jeder Frame muss mit dem Zeichen „[STX]“ beginnen und mit den beiden Zeichen „[CR][LF]“ beendet werden.
		[STX]...[ETX]	Jeder Frame muss mit dem Zeichen „[STX]“ beginnen und mit dem Zeichen „[ETX]“ beendet werden.
	Zeichen Timeout	0...Kein Timeout Maximal zulässiger Zeitabstand zwischen dem vollständigen Empfang zweier Zeichen. Bei Überschreiten der Zeit wird ein Kommunikationsfehler erkannt. Die Überwachung beginnt nach dem Empfang des ersten Zeichens einer Antwort des digitalen Sensors auf eine Anfrage.	
	Zahlen Trennzeichen	Unterteilt die ASCII-Datenmessage des digitalen Sensors in einzelne Messwerte. Über das Eingabefeld können bis zu 10 verschiedene Trennzeichen eingegeben werden. Als Trennzeichen werden auch folgende Escape Codes unterstützt: \a \b \f \n \r \t \v \ \? \' \" \xhh Bei der Verwendung von Escape Codes reduziert sich die max. Anzahl an Trennzeichen.	
Zahlenformat	1,000.00	"," wird als Tausendertrennzeichen verwendet "." wird als Dezimaltrennzeichen verwendet	
	1.000,00	"." wird als Tausendertrennzeichen verwendet "," wird als Dezimaltrennzeichen verwendet	
Device Logic Parsing	Aufwärmzeit	Zeit vom Einschalten des myDatalogMUC xG/4G bis zum Senden der Init Sequenz.	

9.2.1.5 Sequenzen für COM2

Hinweis: Dieser Konfigurationsabschnitt ist nur sichtbar, wenn im Konfigurationsabschnitt „Schnittstellen“ (siehe "Basis" auf Seite 76) für die Schnittstelle COM2 der Modus "ASCII" aktiviert ist. Ist der Modus "Device Logic Parsing" aktiv, müssen die Init Sequenz und die Messdaten-Sequenz durch die PAWN-Device Logic erzeugt werden.

Init Sequenz

Initialisierungsbefehl, der nach dem Einschalten des myDatalogMUC xG/4G nach Ablauf der Aufwärmzeit an den digitalen Sensor gesendet wird

Folgende Escape Codes werden unterstützt:

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

Messdaten Sequenz

Befehl, der zu jedem Messzeitpunkt an den digitalen Sensor gesendet wird [0-512 Zeichen]

Folgende Escape Codes werden unterstützt:

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

9.2.1.6 Messkanäle

9.2.1.6.1 Basis

Bezeichnung 1-8

frei wählbare Kanalbezeichnung für die Universaleingänge

Bezeichnung PT 100/1000

frei wählbare Kanalbezeichnung für den externen Temperatursensor

Modus

Basiseinstellung für den Messkanal

<i>Universaleingänge (Digitalmodi)</i>	<i>aus</i>	<i>---</i>	<i>Messkanal deaktiviert</i>	
	<i>Digital</i>	<i>Invertieren</i>	<i>invertiert das Eingangssignal</i>	
	<i>Cnt.Day</i>	<i>Impuls</i>		<i>Zählwert eines Impulses in der Messeinheit</i>
		<i>Max</i>		<i>definiert das obere Skalenende der Zeigerinstrumente</i>
		<i>Einheit</i>		<i>String, der als Messwerteinheit von allen Anzeigeelementen des Servers verwendet wird [0-16 Zeichen]</i>
		<i>Nachkomma</i>		<i>Anzahl der Nachkommastellen, die von allen Anzeigeelementen des Servers verwendet wird</i>
	<i>Cnt.Intrvl.</i>	<i>Impuls</i>		<i>Zählwert eines Impulses in der Messeinheit</i>
		<i>Max</i>		<i>definiert das obere Skalenende der Zeigerinstrumente</i>
		<i>Einheit</i>		<i>String, der als Messwerteinheit von allen Anzeigeelementen des Servers verwendet wird [0-16 Zeichen]</i>
		<i>Nachkomma</i>		<i>Anzahl der Nachkommastellen, die von allen Anzeigeelementen des Servers verwendet wird</i>
	<i>Freq</i>	<i>Faktor</i>		<i>Faktor mit dem das Eingangssignal multipliziert wird</i>
		<i>Min</i>		<i>definiert das untere Skalenende der Zeigerinstrumente</i>
		<i>Max</i>		<i>definiert das obere Skalenende der Zeigerinstrumente</i>
		<i>Einheit</i>		<i>String, der als Messwerteinheit von allen Anzeigeelementen des Servers verwendet wird [0-16 Zeichen]</i>
		<i>Nachkomma</i>		<i>Anzahl der Nachkommastellen, die von allen Anzeigeelementen des Servers verwendet wird</i>
	<i>PWM</i>	<i>0%</i>		<i>Start des Messbereichs in der Messeinheit</i>
		<i>100%</i>		<i>Ende des Messbereichs in der Messeinheit</i>
		<i>Einheit</i>		<i>String, der als Messwerteinheit von allen Anzeigeelementen des Servers verwendet wird [0-16 Zeichen]</i>
		<i>Nachkomma</i>		<i>Anzahl der Nachkommastellen, die von allen Anzeigeelementen des Servers verwendet wird</i>

<i>Universaleingänge (Analogmodi)</i>	<i>0-20mA</i>	<i>0%</i>	<i>Start des Messbereichs in der Messeinheit</i>
		<i>100%</i>	<i>Ende des Messbereichs in der Messeinheit</i>
		<i>Einheit</i>	<i>String, der als Messwerteinheit von allen Anzeigeelementen des Servers verwendet wird [0-16 Zeichen]</i>
		<i>Nachkomma</i>	<i>Anzahl der Nachkommastellen, die von allen Anzeigeelementen des Servers verwendet wird</i>
	<i>4-20mA</i>	<i>0%</i>	<i>Start des Messbereichs in der Messeinheit</i>
		<i>100%</i>	<i>Ende des Messbereichs in der Messeinheit</i>
		<i>Einheit</i>	<i>String, der als Messwerteinheit von allen Anzeigeelementen des Servers verwendet wird [0-16 Zeichen]</i>
		<i>Nachkomma</i>	<i>Anzahl der Nachkommastellen, die von allen Anzeigeelementen des Servers verwendet wird</i>
	<i>0-2V</i>	<i>0%</i>	<i>Start des Messbereichs in der Messeinheit</i>
		<i>100%</i>	<i>Ende des Messbereichs in der Messeinheit</i>
		<i>Einheit</i>	<i>String, der als Messwerteinheit von allen Anzeigeelementen des Servers verwendet wird [0-16 Zeichen]</i>
		<i>Nachkomma</i>	<i>Anzahl der Nachkommastellen, die von allen Anzeigeelementen des Servers verwendet wird</i>
	<i>0-10V</i>	<i>0%</i>	<i>Start des Messbereichs in der Messeinheit</i>
		<i>100%</i>	<i>Ende des Messbereichs in der Messeinheit</i>
		<i>Einheit</i>	<i>String, der als Messwerteinheit von allen Anzeigeelementen des Servers verwendet wird [0-16 Zeichen]</i>
		<i>Nachkomma</i>	<i>Anzahl der Nachkommastellen, die von allen Anzeigeelementen des Servers verwendet wird</i>
<i>ext. Temperatursensor</i>	<i>aus</i>	<i>---</i>	<i>Messkanal deaktiviert</i>
	<i>ein</i>	<i>Min</i>	<i>definiert das untere Skalenende der Zeigerinstrumente</i>
		<i>Max</i>	<i>definiert das obere Skalenende der Zeigerinstrumente</i>
		<i>Einheit</i>	<i>Auswahl der Temperatureinheit, die von allen Anzeigeelementen des Servers verwendet wird</i>
		<i>Nachkomma</i>	<i>Anzahl der Nachkommastellen, die von allen Anzeigeelementen des Servers verwendet wird</i>

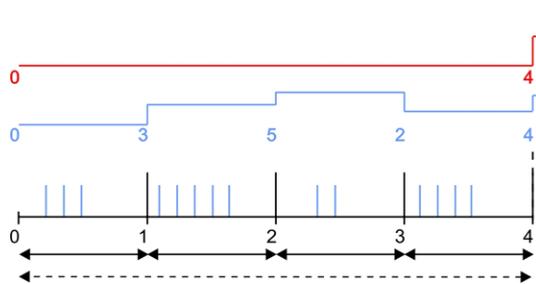
9.2.1.6.2 Konfig

Universaleingänge (Digitalmodi)	aus	---	---	
	Digital	Filter Zeit	Zeit in [ms], für die ein Signal konstant anliegen muss, um einen Pegelwechsel auszulösen. Dient zur Unterdrückung von kurzzeitigen Störungen (Entprellung).	
		Dämpfung	zeitliche Funktion im Messintervall	
			aus	Dämpfung deaktiviert
			up	Mindestsignallänge für x sec. bei steigender Flanke
			down	Mindestsignallänge für x sec. bei fallender Flanke
			up&down	Mindestsignallänge für x sec. bei beiden Flanken
Zeit	Zeit x, die bei den Dämpfungsmodi "up", "down" und "up&down" eingesetzt wird			

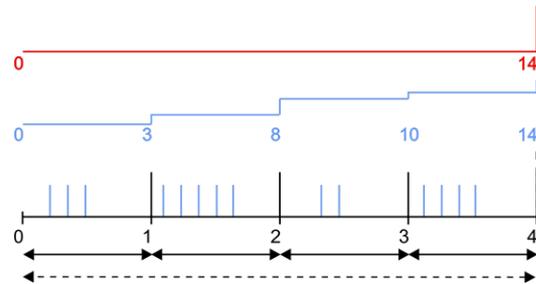
Universaleingänge (Countermodi)	Cnt.Day	Filter Zeit	Zeit in [ms], für die ein Signal konstant anliegen muss, um einen Pegelwechsel auszulösen. Dient zur Unterdrückung von kurzzeitigen Störungen (Entprellung).		
		Rücksetzen	Rücksetzeitpunkt des Tageszählers		
	Cnt.Intervl.	Filter Zeit	Zeit in [ms], für die ein Signal konstant anliegen muss, um einen Pegelwechsel auszulösen. Dient zur Unterdrückung von kurzzeitigen Störungen (Entprellung).		
		Dämpfung	zeitliche Funktion im Messintervall		
			aus	Dämpfung deaktiviert	
			min	Das Minimum der letzten x Messwerte wird aufgezeichnet.	
			max	Das Maximum der letzten x Messwerte wird aufgezeichnet.	
			avg	Das arithmetische Mittel der letzten x Messwerte wird aufgezeichnet.	
			med	Der Median der letzten x Messwerte wird aufgezeichnet.	
			rms	Das quadratische Mittel der letzten x Messwerte wird aufgezeichnet.	
Zeit	Zeitfenster für die Dämpfung. Für die Berechnung der Anzahl der berücksichtigten Messwerte siehe "Beispiel zur Erklärung Aufzeichnungs-, Mess-, Burstintervall in Verbindung mit der Dämpfung" auf Seite 134. Konnte kein temporärer Speicher mehr für die Berücksichtigung eines weiteren Messwerts reserviert werden, wird der Fehler "DECAY MEM ERR" ins Gerätelog eingetragen (siehe "Log-Einträge und Fehlercodes " auf Seite 216).				

Hinweis: Ergänzende Erklärung zum Unterschied zwischen "Cnt.Day" und "Cnt.Intervl."

Grundeinstellungen	←-----→	Aufzeichnungsintervall	4min.	Aufgezeichneter Wert	rote Line
	↔	Messintervall	1min.	Messwert	blaue Linie



Modus "Cnt.Intervl.": Die Impulse werden aufaddiert und nach jeder Messwerterzeugung zurückgesetzt.



Modus "Cnt.Day": Alle Impulse bis zum Rücksetzzeitpunkt werden aufaddiert.

Anmerkung zum Modus "Cnt.Intervl.": Ist das Aufzeichnungsintervall größer als das Messintervall, wird nur die Anzahl der bei der letzten Messwerterzeugung registrierten Impulse aufgezeichnet. In diesem Fall lassen sich die einzelnen Messwerte mittels Dämpfung zusammenfassen.

Universaleingänge (Frequenzmodus 1/2)	Freq (1/2)	Filter Zeit	Zeit in [ms], für die ein Signal konstant anliegen muss, um einen Pegelwechsel auszulösen. Dient zur Unterdrückung von kurzzeitigen Störungen (Entprellung).		
		Dämpfung	zeitliche Funktion im Messintervall		
			aus	Dämpfung deaktiviert	
			min	Das Minimum der letzten x Messwerte wird aufgezeichnet.	
			max	Das Maximum der letzten x Messwerte wird aufgezeichnet.	
			avg	Das arithmetische Mittel der letzten x Messwerte wird aufgezeichnet.	
			med	Der Median der letzten x Messwerte wird aufgezeichnet.	
			rms	Das quadratische Mittel der letzten x Messwerte wird aufgezeichnet.	
		Zeit	Zeitfenster für die Dämpfung. Für die Berechnung der Anzahl der berücksichtigten Messwerte siehe "Beispiel zur Erklärung Aufzeichnungs-, Mess-, Burstintervall in Verbindung mit der Dämpfung" auf Seite 134. Konnte kein temporärer Speicher mehr für die Berücksichtigung eines weiteren Messwerts reserviert werden, wird der Fehler "DECAY MEM ERR" ins Gerätelog eingetragen (siehe "Log-Einträge und Fehlercodes " auf Seite 216).		
		Halten	Halten des letztgültigen Messwerts für x Messzyklen		
			aus	Funktion deaktiviert	
			1-5	Anzahl der Messzyklen, für die der Messwert gehalten wird, bevor der Fehlerwert ausgegeben wird	
			ein	Im Fehlerfall wird der zuletzt gültige Messwert so lange gehalten, bis wieder ein neuer gültiger Messwert vorliegt.	

Universaleingänge (Frequenzmodus 2/2)	Freq (2/2)	Überlauf	Handling bei Messbereichsverletzungen	
			Ignorieren	Der Messwert wird über die Bereichsgrenzen hinaus berechnet.
			Abschneiden	Der Messwert wird bei den Bereichsgrenzen abgeschnitten.
			Überlauf	<ul style="list-style-type: none"> Ist der Messwert unter 1Hz, wird der Fehlerwert „UF“ (Under Flow) ausgegeben. Ist der Messwert über 1000Hz, wird der Fehlerwert „OF“ (Over Flow) ausgegeben.
			NAMUR Grenzen	<ul style="list-style-type: none"> Ist der Messwert unter 1Hz, wird der Fehlerwert „UF“ (Under Flow) ausgegeben. Ist der Messwert über 1000Hz, wird der Fehlerwert „OF“ (Over Flow) ausgegeben.

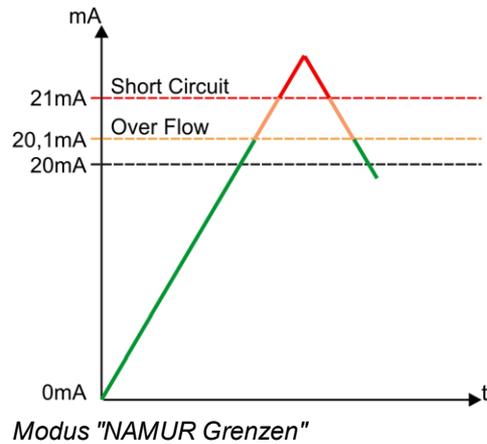
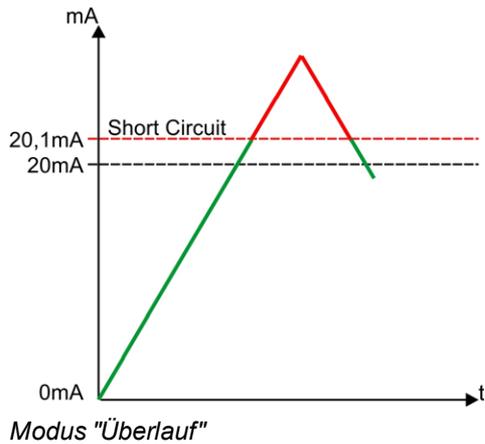
Universaleingänge (PWM-Modus 1/2)	PWM (1/2)	Filter Zeit	Zeit in [ms], für die ein Signal konstant anliegen muss, um einen Pegelwechsel auszulösen. Dient zur Unterdrückung von kurzzeitigen Störungen (Entprellung).		
		Dämpfung	zeitliche Funktion im Messintervall		
			aus	Dämpfung deaktiviert	
			min	Das Minimum der letzten x Messwerte wird aufgezeichnet.	
			max	Das Maximum der letzten x Messwerte wird aufgezeichnet.	
			avg	Das arithmetische Mittel der letzten x Messwerte wird aufgezeichnet.	
			med	Der Median der letzten x Messwerte wird aufgezeichnet.	
			rms	Das quadratische Mittel der letzten x Messwerte wird aufgezeichnet.	
		Zeit	Zeitfenster für die Dämpfung. Für die Berechnung der Anzahl der berücksichtigten Messwerte siehe "Beispiel zur Erklärung Aufzeichnungs-, Mess-, Burstintervall in Verbindung mit der Dämpfung" auf Seite 134. Konnte kein temporärer Speicher mehr für die Berücksichtigung eines weiteren Messwerts reserviert werden, wird der Fehler "DECAY MEM ERR" ins Gerätelog eingetragen (siehe "Log-Einträge und Fehlercodes " auf Seite 216).		
		Halten	Halten des letztgültigen Messwerts für x Messzyklen		
			aus	Funktion deaktiviert	
			1-5	Anzahl der Messzyklen, für die der Messwert gehalten wird, bevor der Fehlerwert ausgegeben wird	
ein	Im Fehlerfall wird der zuletzt gültige Messwert so lange gehalten, bis wieder ein neuer gültiger Messwert vorliegt.				

Universaleingänge (PWM-Modus 2/2)	PWM (2/2)	Überlauf	Handling bei Messbereichsverletzungen	
			Ignorieren	Der Messwert wird über die Bereichsgrenzen hinaus berechnet.
			Abschneiden	Der Messwert wird bei den Bereichsgrenzen abgeschnitten.
			Überlauf	<ul style="list-style-type: none"> Ist der Messwert unter 1%, wird der Fehlerwert „UF“ (Under Flow) ausgegeben. Ist der Messwert über 99%, wird der Fehlerwert „OF“ (Over Flow) ausgegeben.
			NAMUR Grenzen	<ul style="list-style-type: none"> Ist der Messwert unter 1%, wird der Fehlerwert „UF“ (Under Flow) ausgegeben. Ist der Messwert über 99%, wird der Fehlerwert „OF“ (Over Flow) ausgegeben.

Universaleingänge (0-20mA-Modus 1/2)	0-20mA (1/2)	Filter Zeit	Zeit in [ms], über die das Analogsignal zwecks Signalglättung gemittelt wird. Dient zur Unterdrückung von Signalrauschen (siehe auch "Beispiel zur Erklärung der Filter Zeit in Kombination mit der ext. Aufwärmzeit" auf Seite 129).		
		Dämpfung	zeitliche Funktion im Messintervall		
			aus	Dämpfung deaktiviert	
			min	Das Minimum der letzten x Messwerte wird aufgezeichnet.	
			max	Das Maximum der letzten x Messwerte wird aufgezeichnet.	
			avg	Das arithmetische Mittel der letzten x Messwerte wird aufgezeichnet.	
			med	Der Median der letzten x Messwerte wird aufgezeichnet.	
			rms	Das quadratische Mittel der letzten x Messwerte wird aufgezeichnet.	
		Zeit	Zeitfenster für die Dämpfung. Für die Berechnung der Anzahl der berücksichtigten Messwerte siehe "Beispiel zur Erklärung Aufzeichnungs-, Mess-, Burstintervall in Verbindung mit der Dämpfung" auf Seite 134. Konnte kein temporärer Speicher mehr für die Berücksichtigung eines weiteren Messwerts reserviert werden, wird der Fehler "DECAY MEM ERR" ins Gerätelog eingetragen (siehe "Log-Einträge und Fehlercodes " auf Seite 216).		
		Halten	Halten des letztgültigen Messwerts für x Messzyklen		
aus	Funktion deaktiviert				
1-5	Anzahl der Messzyklen, für die der Messwert gehalten wird, bevor der Fehlerwert ausgegeben wird				
ein	Im Fehlerfall wird der zuletzt gültige Messwert so lange gehalten, bis wieder ein neuer gültiger Messwert vorliegt.				

Universaleingänge (0-20mA-Modus 2/2)	0-20mA (2/2)	Überlauf	Handling bei Messbereichsverletzungen	
			Ignorieren	Der Messwert wird über die Bereichsgrenzen hinaus berechnet.
			Abschneiden	Der Messwert wird bei den Bereichsgrenzen abgeschnitten.
			Überlauf	<ul style="list-style-type: none"> Ist der Messwert über 20,1mA, wird der Fehlerwert "SC" (Short Circuit) ausgegeben.
		NAMUR Grenzen	<ul style="list-style-type: none"> Ist der Messwert zwischen 20,1mA und 21mA, wird der Fehlerwert "OF" (Over Flow) ausgegeben. Ist der Messwert über 21mA, wird der Fehlerwert "SC" (Short Circuit) ausgegeben. 	

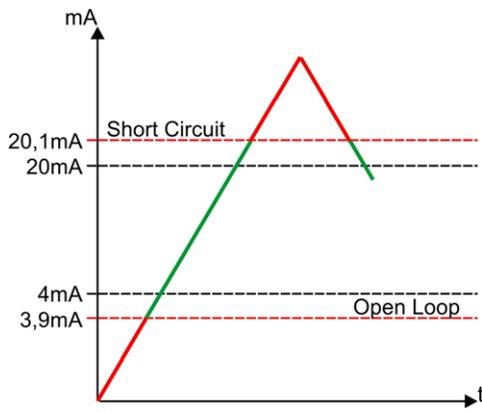
Hinweis: Ergänzende Erklärung zu den Überlaufmodi "Überlauf" und "NAMUR Grenzen"



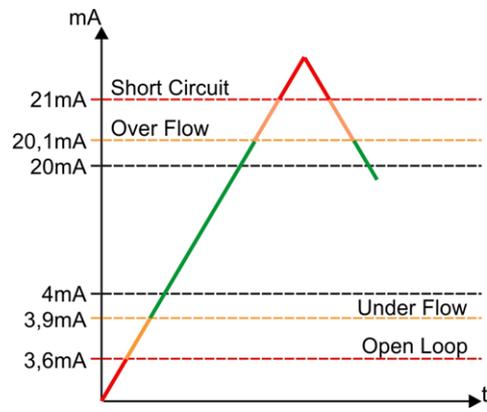
Universaleingänge (4-20mA-Modus 1/2)	4-20mA (1/2)	Filter Zeit	Zeit in [ms], über die das Analogsignal zwecks Signalglättung gemittelt wird. Dient zur Unterdrückung von Signalrauschen (siehe auch "Beispiel zur Erklärung der Filter Zeit in Kombination mit der ext. Aufwärmzeit" auf Seite 129).		
		Dämpfung	zeitliche Funktion im Messintervall		
			aus	Dämpfung deaktiviert	
			min	Das Minimum der letzten x Messwerte wird aufgezeichnet.	
			max	Das Maximum der letzten x Messwerte wird aufgezeichnet.	
			avg	Das arithmetische Mittel der letzten x Messwerte wird aufgezeichnet.	
			med	Der Median der letzten x Messwerte wird aufgezeichnet.	
			rms	Das quadratische Mittel der letzten x Messwerte wird aufgezeichnet.	
		Zeit	Zeitfenster für die Dämpfung. Für die Berechnung der Anzahl der berücksichtigten Messwerte siehe "Beispiel zur Erklärung Aufzeichnungs-, Mess-, Burstintervall in Verbindung mit der Dämpfung" auf Seite 134. Konnte kein temporärer Speicher mehr für die Berücksichtigung eines weiteren Messwerts reserviert werden, wird der Fehler "DECAY MEM ERR" ins Gerätelog eingetragen (siehe "Log-Einträge und Fehlercodes " auf Seite 216).		
		Halten	Halten des letztgültigen Messwerts für x Messzyklen		
			aus	Funktion deaktiviert	
			1-5	Anzahl der Messzyklen, für die der Messwert gehalten wird, bevor der Fehlerwert ausgegeben wird	
			ein	Im Fehlerfall wird der zuletzt gültige Messwert so lange gehalten, bis wieder ein neuer gültiger Messwert vorliegt.	

Universaleingänge (4-20mA-Modus 2/2)	4-20mA (2/2)	Überlauf	Handling bei Messbereichsverletzungen	
			Ignorieren	Der Messwert wird über die Bereichsgrenzen hinaus berechnet.
			Abschneiden	Der Messwert wird bei den Bereichsgrenzen abgeschnitten.
			Überlauf	<ul style="list-style-type: none"> Ist der Messwert unter 3,9mA, wird der Fehlerwert "OL" (Open Loop) ausgegeben. Ist der Messwert über 20,1mA, wird der Fehlerwert "SC" (Short Circuit) ausgegeben.
		NAMUR Grenzen	<ul style="list-style-type: none"> Ist der Messwert unter 3,6mA, wird der Fehlerwert "OL" (Open Loop) ausgegeben. Ist der Messwert zwischen 3,6mA und 3,9mA, wird der Fehlerwert "UF" (Under Flow) ausgegeben. Ist der Messwert zwischen 20,1mA und 21mA, wird der Fehlerwert "OF" (Over Flow) ausgegeben. Ist der Messwert über 21mA, wird der Fehlerwert "SC" (Short Circuit) ausgegeben. 	

Hinweis: Ergänzende Erklärung zu den Überlaufmodi "Überlauf" und "NAMUR Grenzen"



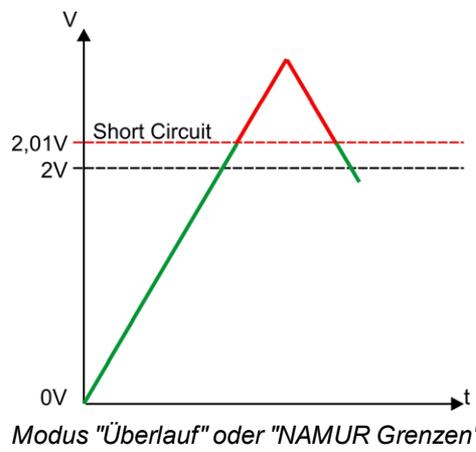
Modus "Überlauf"



Modus "NAMUR Grenzen"

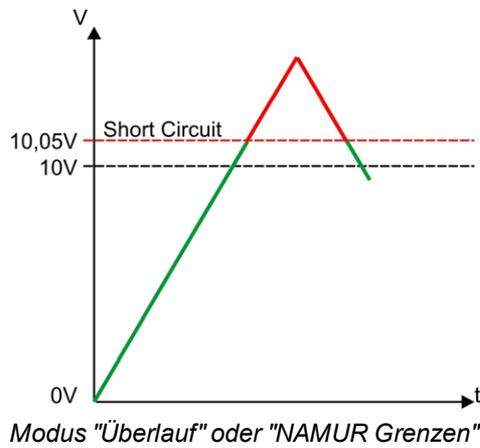
Universaleingänge (0-2V-Modus)	0-2V	Filter Zeit	Zeit in [ms], über die das Analogsignal zwecks Signalglättung gemittelt wird. Dient zur Unterdrückung von Signalrauschen (siehe auch "Beispiel zur Erklärung der Filter Zeit in Kombination mit der ext. Aufwärmzeit" auf Seite 129).		
		Dämpfung	zeitliche Funktion im Messintervall		
			aus	Dämpfung deaktiviert	
			min	Das Minimum der letzten x Messwerte wird aufgezeichnet.	
			max	Das Maximum der letzten x Messwerte wird aufgezeichnet.	
			avg	Das arithmetische Mittel der letzten x Messwerte wird aufgezeichnet.	
			med	Der Median der letzten x Messwerte wird aufgezeichnet.	
			rms	Das quadratische Mittel der letzten x Messwerte wird aufgezeichnet.	
		Zeit	Zeitfenster für die Dämpfung. Für die Berechnung der Anzahl der berücksichtigten Messwerte siehe "Beispiel zur Erklärung Aufzeichnungs-, Mess-, Burstintervall in Verbindung mit der Dämpfung" auf Seite 134. Konnte kein temporärer Speicher mehr für die Berücksichtigung eines weiteren Messwerts reserviert werden, wird der Fehler "DECAY MEM ERR" ins Gerätelog eingetragen (siehe "Log-Einträge und Fehlercodes " auf Seite 216).		
		Halten	Halten des letztgültigen Messwerts für x Messzyklen		
			aus	Funktion deaktiviert	
			1-5	Anzahl der Messzyklen, für die der Messwert gehalten wird, bevor der Fehlerwert ausgegeben wird	
			ein	Im Fehlerfall wird der zuletzt gültige Messwert so lange gehalten, bis wieder ein neuer gültiger Messwert vorliegt.	
		Überlauf	Handling bei Messbereichsverletzungen		
			Ignorieren	Der Messwert wird über die Bereichsgrenzen hinaus berechnet.	
			Abschneiden	Der Messwert wird bei den Bereichsgrenzen abgeschnitten.	
			Überlauf	Ist der Messwert über 2,01V, wird der Fehlerwert "OF" (Over Flow) ausgegeben.	
NAMUR Grenzen	Ist der Messwert über 2,01V, wird der Fehlerwert "OF" (Over Flow) ausgegeben.				

Hinweis: Ergänzende Erklärung zu den Überlaufmodi "Überlauf" und "NAMUR Grenzen"



Universaleingänge (0-10V-Modus)	0-10V	Filter Zeit	Zeit in [ms], über die das Analogsignal zwecks Signalglättung gemittelt wird. Dient zur Unterdrückung von Signalrauschen (siehe auch "Beispiel zur Erklärung der Filter Zeit in Kombination mit der ext. Aufwärmzeit" auf Seite 129).		
		Dämpfung	zeitliche Funktion im Messintervall		
			aus	Dämpfung deaktiviert	
			min	Das Minimum der letzten x Messwerte wird aufgezeichnet.	
			max	Das Maximum der letzten x Messwerte wird aufgezeichnet.	
			avg	Das arithmetische Mittel der letzten x Messwerte wird aufgezeichnet.	
			med	Der Median der letzten x Messwerte wird aufgezeichnet.	
			rms	Das quadratische Mittel der letzten x Messwerte wird aufgezeichnet.	
		Zeit	Zeitfenster für die Dämpfung. Für die Berechnung der Anzahl der berücksichtigten Messwerte siehe "Beispiel zur Erklärung Aufzeichnungs-, Mess-, Burstintervall in Verbindung mit der Dämpfung" auf Seite 134. Konnte kein temporärer Speicher mehr für die Berücksichtigung eines weiteren Messwerts reserviert werden, wird der Fehler "DECAY MEM ERR" ins Gerätelog eingetragen (siehe "Log-Einträge und Fehlercodes " auf Seite 216).		
		Halten	Halten des letztgültigen Messwerts für x Messzyklen		
			aus	Funktion deaktiviert	
			1-5	Anzahl der Messzyklen, für die der Messwert gehalten wird, bevor der Fehlerwert ausgegeben wird	
			ein	Im Fehlerfall wird der zuletzt gültige Messwert so lange gehalten, bis wieder ein neuer gültiger Messwert vorliegt.	
		Überlauf	Handling bei Messbereichsverletzungen		
			Ignorieren	Der Messwert wird über die Bereichsgrenzen hinaus berechnet.	
			Abschneiden	Der Messwert wird bei den Bereichsgrenzen abgeschnitten.	
Überlauf	Ist der Messwert über 10,05V, wird der Fehlerwert "OF" (Over Flow) ausgegeben.				
NAMUR Grenzen	Ist der Messwert über 10,05V, wird der Fehlerwert "OF" (Over Flow) ausgegeben.				

Hinweis: Ergänzende Erklärung zu den Überlaufmodi "Überlauf" und "NAMUR Grenzen"



ext. Temperatursensor	aus	---	Messkanal deaktiviert	
	ein	Halten	Halten des letztgültigen Messwerts für x Messzyklen	
			aus	Funktion deaktiviert
			1-5	Anzahl der Messzyklen, für die der Messwert gehalten wird, bevor der Fehlerwert ausgegeben wird
			ein	Im Fehlerfall wird der zuletzt gültige Messwert so lange gehalten, bis wieder ein neuer gültiger Messwert vorliegt.

9.2.1.6.3 Alarme

"Digital"- Modus	WA	Ein "High" am Universaleingang löst eine "Warnung" aus.	
	AL	Ein "High" am Universaleingang löst einen "Alarm" aus.	
	SW	Ein "High" am Universaleingang löst eine "Störung Warnung" aus.	
	SA	Ein "High" am Universaleingang löst einen "Störung Alarm" aus.	
Alle anderen Modi	Warnung	Wert niedrig	Wenn der Messwert auf oder unter diesen Wert fällt, wird eine Warnung ausgelöst.
		Wert hoch	Wenn der Messwert auf oder über diesen Wert steigt, wird eine Warnung ausgelöst.
	Alarm	Wert niedrig	Wenn der Messwert auf oder unter diesen Wert fällt, wird ein Alarm ausgelöst.
		Wert hoch	Wenn der Messwert auf oder über diesen Wert steigt, wird ein Alarm ausgelöst.
	Hyst %	Hysterese für Entwarnung bei Alarm/Warnung (z.B. Hyst=5%, Alarm od. Warnung bei 100 -> Entwarnung bei 95) bzw. Zurücknehmen des Triggers (z.B. Hyst=5%, Schwelle = größer gleich, Trigger bei 100 -> zurücknehmen bei 95)	

9.2.1.6.4 Trigger

Wird ein Universaleingang im Digital-Modus betrieben, wird zwischen zwei Arten von Triggern unterschieden:

- Eventtrigger (MS, XM, I1-I4)

Anders als bei den Leveltriggern wird die entsprechende Operation (z.B. Übertragung auslösen) beim Auftreten des Triggerereignisses nur ein einziges Mal ausgeführt. Mit Hilfe des Konfigurationsparameters "Flanke" wird angegeben, ob die steigende, fallende oder beide Flanken das Triggerereignis auslösen sollen.

- Leveltrigger (QU, SL, ON)

Ein "High" am Universaleingang löst den Trigger aus. Durch ein "Low" am Universaleingang wird der Trigger wieder zurückgenommen. Solange der Trigger aktiv ist, wird die entsprechende Operation (z.B. Online-Modus aktivieren) ausgeführt. Die über den Konfigurationsparameter "Flanke" getroffene Auswahl ist für die Leveltrigger nicht relevant. Ist es erforderlich, dass der Trigger durch ein "Low" am Universaleingang ausgelöst und durch ein "High" wieder zurückgenommen wird, muss das Eingangssignal mit Hilfe des Konfigurationsparameters "Invertieren", der sich im Tab "Basis" befindet, invertiert werden.

Auch bei den anderen Modi der Universaleingänge wird zwischen diesen beiden Arten von Triggern unterschieden:

- Eventtrigger (XM)

Die entsprechende Operation (z.B. Übertragung auslösen) wird beim Auftreten des Triggerereignisses nur ein einziges Mal ausgeführt.

- Leveltrigger (QU, SL, RO, RF, ON, I1-I4)

Solange der Trigger aktiv ist, wird die entsprechende Operation (z.B. Online-Modus aktivieren) ausgeführt.

"Digital"- Modus	Eventtrigger	MS	Messzyklus sofort starten		
		XM	Übertragung auslösen		
		I1	intern zur Verwendung mit Steuerprogramm		
		I2	intern zur Verwendung mit Steuerprogramm		
		I3	intern zur Verwendung mit Steuerprogramm		
		I4	intern zur Verwendung mit Steuerprogramm		
		Flanke	Auswahl der Flanke, bei der der Trigger ausgelöst werden soll		
			steigende	Die steigende Flanke löst den Trigger aus.	
			fallende	Die fallende Flanke löst den Trigger aus.	
			beide	Beide Flanken lösen den Trigger aus.	
	Leveltrigger	QU	schnelle Aufzeichnung (Aufzeichnungsintervall = Aufzeichnungsintervall / Faktor)		
		SL	langsame Aufzeichnung (Aufzeichnungsintervall = Aufzeichnungsintervall * Faktor)		
		ON	Online-Modus aktivieren		
		Flanke	nicht relevant für Leveltrigger		

Alle anderen Modi	Eventtrigger	XM	Übertragung auslösen	
		Schwelle	Schwellen für das Auslösen des Triggers. Für die Ermittlung der Schwelle zum Zurücksetzen des Triggers wird die Hysterese aus dem Tab "Alarme" verwendet.	
			größer gleich	Wenn der Messwert auf oder über diesen Wert steigt, wird der Trigger ausgelöst.
			kleiner gleich	Wenn der Messwert auf oder unter diesen Wert fällt, wird der Trigger ausgelöst
	Leveltrigger	QU	schnelle Aufzeichnung (Aufzeichnungsintervall = Aufzeichnungsintervall / Faktor)	
		SL	langsame Aufzeichnung (Aufzeichnungsintervall = Aufzeichnungsintervall * Faktor)	
		RO	Aufzeichnung einschalten	
		RF	Aufzeichnung ausschalten	
		ON	Online-Modus aktivieren	
		I1	intern zur Verwendung mit Steuerprogramm	
		I2	intern zur Verwendung mit Steuerprogramm	
		I3	intern zur Verwendung mit Steuerprogramm	
		I4	intern zur Verwendung mit Steuerprogramm	
		Schwelle	Schwellen für das Auslösen des Triggers. Für die Ermittlung der Schwelle zum Zurücksetzen des Triggers wird die Hysterese aus dem Tab "Alarme" verwendet.	
			größer gleich	Der Trigger ist aktiv, solange der Messwert größer/gleich der Schwelle ist.
			kleiner gleich	Der Trigger ist aktiv, solange der Messwert kleiner/gleich der Schwelle ist.

9.2.1.7 Interface Kanäle 1-32

Hinweis: Einige Konfigurationsparameter dieses Konfigurationsabschnitts wirken sich auch auf Parameter aus, die sich nicht im gerade geöffneten Tab befinden. Im folgenden Abschnitt wird dies durch das „>“-Symbol dargestellt. z.B. „Basis > Min“ bedeutet, dass der Konfigurationsparameter „Min“ sich im Tab „Basis“ befindet.

Ein Interface Kanal dient der Aufnahme eines einzelnen Messwerts, der von einer der 3 seriellen Schnittstellen (2 x RS485, 1 x RS232) gelesen wird. Für jeden Interface Kanal kann der zu verwendende Datentyp und die Schnittstelle, von der die Daten gelesen werden, unabhängig von den restlichen Kanälen festgelegt werden. In den folgenden Kapiteln erhalten Sie eine Beschreibung zur Konfiguration der Interface Kanäle.

9.2.1.7.1 Basis

Bezeichnung 1-32

frei wählbare Kanalbezeichnung für die Interface Kanäle [0-16 Zeichen]

Schnittstelle

Basiseinstellungen für den Messkanal

<i>aus</i>	---	<i>Messkanal deaktiviert</i>		
<i>Com1</i>	<i>Skalierung > Skalierung</i>	<i>aus</i>	<i>Min</i>	<i>definiert das untere Skalenende der Zeigerinstrumente</i>
			<i>Max</i>	<i>definiert das obere Skalenende der Zeigerinstrumente</i>
<i>Com3</i>	<i>(siehe "Skalierung" auf Seite 112)</i>	<i>ein</i>	---	
	<i>Konfig > Format</i> <i>(siehe "Konfig." auf Seite 109)</i>	<i>Digital</i>		---
		<i>Signed</i>	<i>Einheit</i>	<i>String, der als Messwerteinheit von allen Anzeigeelementen des Servers verwendet wird [0-16 Zeichen]. Dieser hat keinen direkten Einfluss auf die Werte.</i>
			<i>Nachkomma</i>	<i>Anzahl der Nachkommastellen, die von allen Anzeigeelementen des Servers verwendet wird</i>
		<i>Unsigned</i>	<i>Einheit</i>	<i>String, der als Messwerteinheit von allen Anzeigeelementen des Servers verwendet wird [0-16 Zeichen]. Dieser hat keinen direkten Einfluss auf die Werte.</i>
			<i>Nachkomma</i>	<i>Anzahl der Nachkommastellen, die von allen Anzeigeelementen des Servers verwendet wird</i>
		<i>Float</i>	<i>Einheit</i>	<i>String, der als Messwerteinheit von allen Anzeigeelementen des Servers verwendet wird [0-16 Zeichen]. Dieser hat keinen direkten Einfluss auf die Werte.</i>
			<i>Nachkomma</i>	<i>Anzahl der Nachkommastellen, die von allen Anzeigeelementen des Servers verwendet wird</i>

Com2	Min	definiert das untere Skalenende der Zeigerinstrumente
	Max	definiert das obere Skalenende der Zeigerinstrumente
	Einheit	String, der als Messwerteinheit von allen Anzeigeelementen des Servers verwendet wird [0-16 Zeichen]. Dieser hat keinen direkten Einfluss auf die Werte.
	Nachkomma	Anzahl der Nachkommastellen, die von allen Anzeigeelementen des Servers verwendet wird
Device Logic ¹⁾	Min	definiert das untere Skalenende der Zeigerinstrumente
	Max	definiert das obere Skalenende der Zeigerinstrumente
	Einheit	String, der als Messwerteinheit von allen Anzeigeelementen des Servers verwendet wird [0-16 Zeichen]. Dieser hat keinen direkten Einfluss auf die Werte.
	Nachkomma	Anzahl der Nachkommastellen, die von allen Anzeigeelementen des Servers verwendet wird

¹⁾ Der Messwert wird zwar nicht über eine der Schnittstellen gelesen, kann aber durch die Device Logic gesetzt werden (siehe "Mdn_SetCh()" im Kapitel "Messkanäle" auf Seite 152).

Modus

gibt an, wie der Interface Kanal von den Auswerteelementen des Servers zu behandeln ist

analog/digital	Analoger oder digitaler Messwert. D.h. das System betrachtet jeden Messwert unabhängig von den Messwerten davor oder danach.
Cnt.Day	Tageszähler. D.h. das System rechnet damit, dass der Messwert des Kanals kontinuierlich ansteigt und einmal pro Tag zurückgesetzt wird.
Cnt.Intrvl.	Intervallzähler. D.h. das System geht davon aus, dass der Zählerstand nach jeder Messwertaufzeichnung zurückgesetzt wird.
Cnt.Inf.	Endloszähler. D.h. das System erwartet, dass der Messwert des Kanals kontinuierlich ansteigt und niemals zurückgesetzt wird.

9.2.1.7.2 Konfig.

Hinweis: Für einen Interface Kanal, der mit einer Schnittstelle verbunden ist für die "Device Logic Parsing" aktiviert wurde (siehe "Basis" auf Seite 76), sind die folgenden Parameter nicht verfügbar.

9.2.1.7.2.1 Com1 und Com3 (Modbus-Master, RS485)

Slave Ad.

Adresse des Modbus-Slaves

Modbus Ad.

Adresse des Registers, das gelesen werden soll

Format

Datentyp

<i>Digital</i>	<i>Es soll ein Digitalwert gelesen werden.</i>				
	<i>Funktion</i>	<i>Read Coils (FC 01)</i>			
		<i>Read Discrete Inputs (FC 02)</i>			
<i>Signed</i>	<i>Es soll ein vorzeichenbehafteter Integer-Wert gelesen werden.</i>				
	<i>Bit</i>	16	<i>16-Bit Integer</i>		
		32	<i>32-Bit Integer. Dazu müssen 2 Register aus dem Modbus-Slave gelesen werden.</i>		
			<i>Word order</i>	<i>HI-LO</i>	<i>HI-Word auf der niedrigeren Registeradresse, LO-Word auf der höheren Registeradresse</i>
			<i>LO-HI</i>	<i>LO-Word auf der niedrigeren Registeradresse, HI-Word auf der höheren Registeradresse</i>	
	<i>Funktion</i>	<i>Read Holding Registers (FC 03)</i>			
<i>Read Input Registers (FC 04)</i>					
<i>Unsigned</i>	<i>Der zu lesende Integer-Wert ist nicht vorzeichenbehaftet.</i>				
	<i>Bit</i>	16	<i>16-Bit Integer</i>		
		32	<i>32-Bit Integer. Dazu müssen 2 Register aus dem Modbus-Slave gelesen werden.</i>		
			<i>Word order</i>	<i>HI-LO</i>	<i>HI-Word auf der niedrigeren Registeradresse, LO-Word auf der höheren Registeradresse</i>
			<i>LO-HI</i>	<i>LO-Word auf der niedrigeren Registeradresse, HI-Word auf der höheren Registeradresse</i>	
	<i>Funktion</i>	<i>Read Holding Registers (FC 03)</i>			
<i>Read Input Registers (FC 04)</i>					
<i>Float</i>	<i>Es soll ein 32Bit Float gelesen werden.</i>				
	<i>Word order</i>	<i>HI-LO</i>	<i>HI-Word auf der niedrigeren Registeradresse, LO-Word auf der höheren Registeradresse</i>		
		<i>LO-HI</i>	<i>LO-Word auf der niedrigeren Registeradresse, HI-Word auf der höheren Registeradresse</i>		
	<i>Funktion</i>	<i>Read Holding Registers (FC 03)</i>			
		<i>Read Input Registers (FC 04)</i>			

9.2.1.7.2.2 Com1 und Com3 (Modbus-Slave, RS485)

Die folgende Tabelle gibt die möglichen Zugriffsfunktionen in Abhängigkeit vom Datentyp des Interface Kanals an:

Modbus Ad.	Datentyp	Read Funktion	Write Funktion
0x0000 : 0x003F	Digital	Read Coils (FC 01)	Write Single Coil (FC 05) Write Multiple Coils (FC 15)
0x0000 : 0x007F	Signed 16/32Bit Unsigned 16/32Bit Float	Read Holding Registers (FC 03)	Write Single Register (FC 06) Write Multiple Registers (FC 16)

Modbus Ad.

Adresse des Registers, das gelesen werden soll

Format

Datentyp

<i>Digital</i>	<i>Es soll ein Digitalwert gelesen werden.</i>		
<i>Signed</i>	<i>Es soll ein vorzeichenbehafteter Integer-Wert gelesen werden.</i>		
	<i>Bit</i>	16	<i>16-Bit Integer</i>
		32	<i>32-Bit Integer. Dazu müssen 2 Register aus dem Modbus-Slave gelesen werden.</i>
		<i>Word order</i>	<i>HI-LO</i>
		<i>LO-HI</i>	<i>LO-Word auf der niedrigeren Registeradresse, HI-Word auf der höheren Registeradresse</i>
<i>Unsigned</i>	<i>Der zu lesende Integer-Wert ist nicht vorzeichenbehaftet.</i>		
	<i>Bit</i>	16	<i>16-Bit Integer</i>
		32	<i>32-Bit Integer. Dazu müssen 2 Register aus dem Modbus-Slave gelesen werden.</i>
		<i>Word order</i>	<i>HI-LO</i>
		<i>LO-HI</i>	<i>LO-Word auf der niedrigeren Registeradresse, HI-Word auf der höheren Registeradresse</i>
<i>Float</i>	<i>Es soll ein 32Bit Float gelesen werden.</i>		
	<i>Word order</i>	<i>HI-LO</i>	<i>HI-Word auf der niedrigeren Registeradresse, LO-Word auf der höheren Registeradresse</i>
		<i>LO-HI</i>	<i>LO-Word auf der niedrigeren Registeradresse, HI-Word auf der höheren Registeradresse</i>

9.2.1.7.2.3 Com2 (Seriell, RS232)

Spalte

Durch das "Zahlen Trennzeichen" (siehe "Schnittstellen" auf Seite 76) wird die ASCII-Datenmessage des digitalen Sensors in einzelne Messwerte unterteilt. Der Konfigurationsparameter „Spalte“ gibt an, welcher dieser Messwerte mit dem Interface Kanal verknüpft/aufgezeichnet wird.

9.2.1.7.3 Skalierung

Hinweis: Für einen Interface Kanal, der mit einer Schnittstelle verbunden ist für die "Device Logic Parsing" aktiviert wurde (siehe "Basis" auf Seite 76), sind die folgenden Parameter nicht verfügbar.

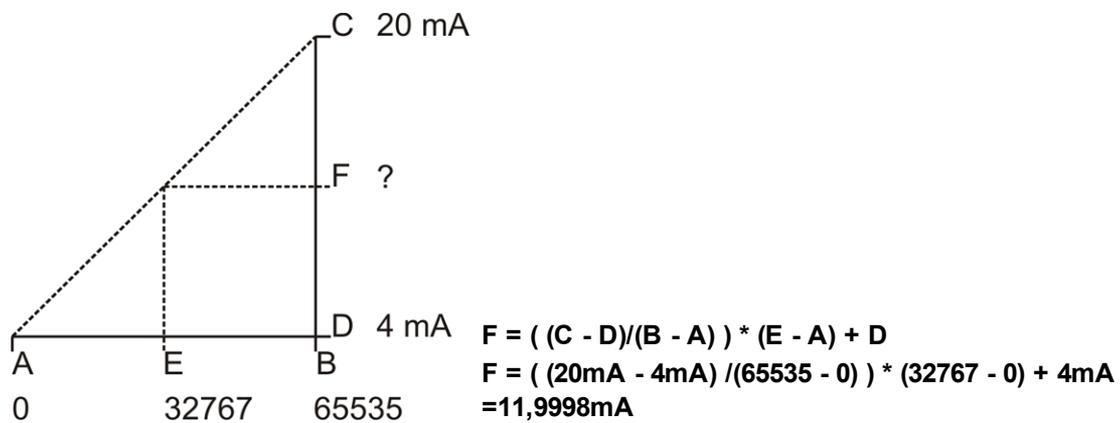
9.2.1.7.3.1 Com1 und Com3 (Modbus, RS485)

Skalierung	aus	---		
	ein	0% Modbus	Start des Messbereichs in der Einheit des Modbus-Slaves	
		100% Modbus	Ende des Messbereichs in der Einheit des Modbus-Slaves	
		0%	Start des Messbereichs in der Messeinheit	
		100%	Ende des Messbereichs in der Messeinheit	
		Überlauf	Handling bei Messbereichsverletzungen	
			Ignorieren	Der Messwert wird über die Bereichsgrenzen hinaus berechnet.
			Abschneiden	Der Messwert wird bei den Bereichsgrenzen abgeschnitten.
			Überlauf	<ul style="list-style-type: none"> Ist der Messwert unterhalb der unteren Bereichsgrenze, wird der Fehlerwert „UF“ (Under Flow) ausgegeben. Ist der Messwert oberhalb der oberen Bereichsgrenze, wird der Fehlerwert „OF“ (Over Flow) ausgegeben.

Hinweis: Berechnung des Messwerts bei aktiver Skalierung:

Basis > Min	4mA	D
Basis > Max	20mA	C
Min Modbus	0	A
Max Modbus	65535	B

Messwert in der Einheit des Modbus-Slaves	32767	E
Skalierter Messwert	11,9998mA	F



9.2.1.7.3.2 Com2 (Seriell, RS232)

Die Skalierung ist für die serielle Schnittstelle nicht verfügbar.

9.2.1.7.4 Alarmer

Hinweis: Wird "Device Logic" als Basiseinstellung für den Messkanal verwendet, können Alarmer nicht automatisch durch das System ausgelöst werden, da das Alarm/Trigger-Modul vor dem Control-Modul, in dem die Device Logic-Arbeitung erfolgt, ausgeführt wird (siehe "Interne Verarbeitung der Messwerte" auf Seite 31). Allerdings kann per Device Logic auf die über diesen Tab einstellbare Alarm-Konfiguration zugegriffen werden und ein Alarm mittels Device Logic ausgelöst werden (siehe "Mdn_GetAlarmCfg()" und "Mdn_SetAlarm()" im Kapitel "Funktionen" auf Seite 162).

Warnung	Wert niedrig	Wenn der Messwert auf oder unter diesen Wert fällt, wird eine Warnung ausgelöst.
	Wert hoch	Wenn der Messwert auf oder über diesen Wert steigt, wird eine Warnung ausgelöst.
Alarm	Wert niedrig	Wenn der Messwert auf oder unter diesen Wert fällt, wird ein Alarm ausgelöst.
	Wert hoch	Wenn der Messwert auf oder über diesen Wert steigt, wird ein Alarm ausgelöst.
Hyst %	Hysterese für Entwarnung bei Alarm/Warnung (z.B. Hyst = 5%, Alarm od. Warnung bei 100 -> Entwarnung bei 95) bzw. zurücknehmen des Triggers (z.B. Hyst = 5%, Schwelle = größer gleich, Trigger bei 100 -> zurücknehmen bei 95)	

9.2.1.7.5 Trigger

Hinweis: Wird "Device Logic" als Basiseinstellung für den Messkanal verwendet, können Trigger nicht automatisch durch das System ausgelöst werden, da das Alarm/Trigger-Modul vor dem Control-Modul, in dem die Device Logic-Abarbeitung erfolgt, ausgeführt wird (siehe "Interne Verarbeitung der Messwerte" auf Seite 31). Allerdings kann per Device Logic auf die über diesen Tab eingebbare Trigger-Konfiguration zugegriffen werden und ein Trigger mittels Device Logic ausgelöst werden (siehe "Mdn_GetTriggerCfg()" und "Mdn_SetTrigger()" im Kapitel "Funktionen" auf Seite 162).

Es wird zwischen den folgenden beiden Arten von Triggern unterschieden:

- Eventtrigger (XM)

Die entsprechende Operation (z.B. Übertragung auslösen) wird beim Auftreten des Triggerereignisses nur ein einziges Mal ausgeführt.

- Leveltrigger (QU, SL, RO, RF, ON, I1-I4)

Solange der Trigger aktiv ist, wird die entsprechende Operation (z.B. Online-Modus aktivieren) ausgeführt.

Eventtrigger	XM	Übertragung auslösen	
	Schwelle	Schwellen für das Auslösen des Triggers. Für die Ermittlung der Schwelle zum Zurücksetzen des Triggers wird die Hysterese aus dem Tab "Alarmer" verwendet.	
		größer gleich	Wenn der Messwert auf oder über diesen Wert steigt, wird der Trigger ausgelöst.
		kleiner gleich	Wenn der Messwert auf oder unter diesen Wert fällt, wird der Trigger ausgelöst
Leveltrigger	QU	schnelle Aufzeichnung (Aufzeichnungsintervall = Aufzeichnungsintervall / Faktor)	
	SL	langsame Aufzeichnung (Aufzeichnungsintervall = Aufzeichnungsintervall * Faktor)	
	RO	Aufzeichnung einschalten	
	RF	Aufzeichnung ausschalten	
	ON	Online-Modus aktivieren	
	I1	intern zur Verwendung mit Steuerprogramm	
	I2	intern zur Verwendung mit Steuerprogramm	
	I3	intern zur Verwendung mit Steuerprogramm	
	I4	intern zur Verwendung mit Steuerprogramm	
	Schwelle	Schwellen für das Auslösen des Triggers. Für die Ermittlung der Schwelle zum Zurücksetzen des Triggers wird die Hysterese aus dem Tab "Alarmer" verwendet.	
		größer gleich	Der Trigger ist aktiv, solange der Messwert größer/gleich der Schwelle ist.
kleiner gleich		Der Trigger ist aktiv, solange der Messwert kleiner/gleich der Schwelle ist.	

9.2.1.8 Interface Kanäle 33-64

9.2.1.8.1 Basis

Bezeichnung 33-64

frei wählbare Kanalbezeichnung für die Interface Kanäle

Die Bedeutung der restlichen Konfigurationsparameter in diesem Konfigurationsabschnitt entspricht dem Konfigurationsabschnitt "Interface Kanäle 1-32" (siehe "Interface Kanäle 1-32" auf Seite 107).

9.2.1.9 Interface Ausgabekanäle 1-32

Hinweis: *Einige Konfigurationsparameter dieses Konfigurationsabschnitts wirken sich auch auf Parameter aus, die sich nicht im gerade geöffneten Tab befinden. Im folgenden Abschnitt wird dies durch das „>“-Symbol dargestellt. Z.B. „Basis > Min“ bedeutet, dass sich der Konfigurationsparameter „Min“ im Tab „Basis“ befindet.*

Ein Interface Ausgabekanal dient der Aufnahme eines einzelnen Stellwerts, der über eine der 3 seriellen Schnittstellen (2 x RS485, 1 x RS232), an einen Sensor oder Aktor übermittelt wird. Für jeden Interface Ausgabekanal kann der zu verwendende Datentyp und die Schnittstelle, über die die Daten übermittelt werden sollen, unabhängig von den restlichen Kanälen festgelegt werden. In den folgenden Kapiteln erhalten Sie eine Beschreibung zur Konfiguration der Interface Ausgabekanäle.

9.2.1.9.1 Basis

Bezeichnung 1-32

frei wählbare Kanalbezeichnung für die Interface Ausgabekanäle [0-16 Zeichen]

Schnittstelle

Basiseinstellungen für den Ausgabekanal:

<i>aus</i>	---	<i>Ausgabekanal deaktiviert</i>				
<i>Com1</i>	<i>Skalierung > Skalierung</i>	<i>aus</i>	<i>Min</i>	<i>definiert das untere Skalenende der Zeigerinstrumente</i>		
			<i>Max</i>	<i>definiert das obere Skalenende der Zeigerinstrumente</i>		
<i>Com3</i>	<i>(siehe "Skalierung" auf Seite 120)</i>	<i>ein</i>	---			
	<i>Konfig > Format</i> <i>(siehe "Konfig" auf Seite 117)</i>	<i>Digital</i>	---			
		<i>Signed</i>	<i>Einheit</i>	<i>String, der als Messwerteinheit von allen Anzeigeelementen des Servers verwendet wird [0-16 Zeichen]. Dieser hat keinen direkten Einfluss auf die Werte.</i>		
			<i>Nachkomma</i>	<i>Anzahl der Nachkommastellen, die von allen Anzeigeelementen des Servers verwendet wird</i>		
		<i>Unsigned</i>	<i>Einheit</i>	<i>String, der als Messwerteinheit von allen Anzeigeelementen des Servers verwendet wird [0-16 Zeichen]. Dieser hat keinen direkten Einfluss auf die Werte.</i>		
			<i>Nachkomma</i>	<i>Anzahl der Nachkommastellen, die von allen Anzeigeelementen des Servers verwendet wird</i>		
		<i>Float</i>	<i>Einheit</i>	<i>String, der als Messwerteinheit von allen Anzeigeelementen des Servers verwendet wird [0-16 Zeichen]. Dieser hat keinen direkten Einfluss auf die Werte.</i>		
			<i>Nachkomma</i>	<i>Anzahl der Nachkommastellen, die von allen Anzeigeelementen des Servers verwendet wird</i>		
		<i>Stellwert</i>	<i>Ausgabewert in der Messeinheit</i>			
		<i>Com2</i>	<i>Min</i>	<i>definiert das untere Skalenende der Zeigerinstrumente</i>		
			<i>Max</i>	<i>definiert das obere Skalenende der Zeigerinstrumente</i>		
<i>Stellwert</i>	<i>Ausgabewert in der Messeinheit</i>					

Device Logic 1)	Min	definiert das untere Skalenende der Zeigerinstrumente
	Max	definiert das obere Skalenende der Zeigerinstrumente
	Einheit	String, der als Messwerteinheit von allen Anzeigeelementen des Servers verwendet wird [0-16 Zeichen]. Dieser hat keinen direkten Einfluss auf die Werte.
	Nachkomma	Anzahl der Nachkommastellen, die von allen Anzeigeelementen des Servers verwendet wird
	Stellwert	Ausgabewert in der Messeinheit

1) Der Stellwert wird zwar nicht über eine der Schnittstellen ausgegeben, kann jedoch von einer Device Logic verwendet werden. Eine mögliche Anwendung ist den Stellwert mittels Server-PLC vorzugeben und durch eine Device Logic am Gerät weiter zu verarbeiten.

Modus

gibt an, wie der Interface Ausgabekanal von den Auswerteelementen des Servers zu behandeln ist

analog/digital	Analoger oder digitaler Messwert. D.h. das System betrachtet jeden Messwert unabhängig von den Messwerten davor oder danach.
Cnt.Day	Tageszähler. D.h. das System rechnet damit, dass der Messwert des Kanals kontinuierlich ansteigt und einmal pro Tag zurückgesetzt wird.
Cnt.Intrvl.	Intervallzähler. D.h. das System geht davon aus, dass der Zählerstand nach jeder Messwertaufzeichnung zurückgesetzt wird.
Cnt.Inf.	Endloszähler. D.h. das System erwartet, dass der Messwert des Kanals kontinuierlich ansteigt und niemals zurückgesetzt wird.

9.2.1.9.2 Konfig

Hinweis: Für einen Interface Ausgabekanal, der mit einer Schnittstelle verbunden ist für die "Device Logic Parsing" aktiviert wurde (siehe "Basis" auf Seite 76), sind die folgenden Parameter nicht verfügbar.

9.2.1.9.2.1 Com1 und Com3 (Modbus-Master, RS485)

Slave Ad.

Adresse des Modbus-Slaves

Modbus Ad.

Adresse des Registers, in das geschrieben werden soll

Format

Datentyp

<i>Digital</i>	<i>Es soll ein Digitalwert geschrieben werden.</i>				
	<i>Funktion</i>	<i>Write Single Coils (FC 05)</i>			
		<i>Write Multiple Coils (FC 15)</i>			
<i>Signed</i>	<i>Es soll ein vorzeichenbehafteter Integer-Wert geschrieben werden.</i>				
	<i>Bit</i>	16	<i>16-Bit Integer</i>		
			<i>Funktion</i>	<i>Write Multiple Registers (FC 16)</i>	
				<i>Write Single Register (FC 06)</i>	
	32	<i>32-Bit Integer. Dazu müssen 2 Register in den Modbus-Slave geschrieben werden.</i>			
		<i>Funktion</i>	<i>Write Multiple Registers (FC 16)</i>		
		<i>Word order</i>	<i>HI-LO</i>	<i>HI-Word auf der niedrigeren Registeradresse, LO-Word auf der höheren Registeradresse</i>	
			<i>LO-HI</i>	<i>LO-Word auf der niedrigeren Registeradresse, HI-Word auf der höheren Registeradresse</i>	
<i>Unsigned</i>	<i>Der zu schreibende Integer-Wert ist nicht vorzeichenbehaftet.</i>				
	<i>Bit</i>	16	<i>16-Bit Integer</i>		
			<i>Funktion</i>	<i>Write Multiple Registers (FC 16)</i>	
				<i>Write Single Register (FC 06)</i>	
	32	<i>32-Bit Integer. Dazu müssen 2 Register in den Modbus-Slave geschrieben werden.</i>			
		<i>Funktion</i>	<i>Write Multiple Registers (FC 16)</i>		
		<i>Word order</i>	<i>HI-LO</i>	<i>HI-Word auf der niedrigeren Registeradresse, LO-Word auf der höheren Registeradresse</i>	
			<i>LO-HI</i>	<i>LO-Word auf der niedrigeren Registeradresse, HI-Word auf der höheren Registeradresse</i>	
<i>Float</i>	<i>Es soll ein 32Bit Float geschrieben werden.</i>				
	<i>Word order</i>	<i>HI-LO</i>	<i>HI-Word auf der niedrigeren Registeradresse, LO-Word auf der höheren Registeradresse</i>		
		<i>LO-HI</i>	<i>LO-Word auf der niedrigeren Registeradresse, HI-Word auf der höheren Registeradresse</i>		
	<i>Funktion</i>	<i>Write Multiple Registers (FC 16)</i>			

9.2.1.9.2.2 Com1 und Com3 (Modbus-Slave, RS485)

Die folgende Tabelle gibt die möglichen Zugriffsfunktionen in Abhängigkeit vom Datentyp des Interface Ausgabekanals an:

Modbus Ad.	Datentyp	Read Funktion	Write Funktion
0x0800 : 0x083F	Digital	Read Discrete Inputs (FC 02)	---
0x0800 : 0x087F	Signed 16/32Bit Unsigned 16/32Bit Float	Read Input Registers (FC 04)	---

Modbus Ad.

Adresse des Registers, in das geschrieben werden soll

Format

Datentyp

<i>Digital</i>	<i>Es soll ein Digitalwert geschrieben werden.</i>			
<i>Signed</i>	<i>Es soll ein vorzeichenbehafteter Integer-Wert geschrieben werden.</i>			
	<i>Bit</i>	16	<i>16-Bit Integer</i>	
		32	<i>32-Bit Integer. Dazu müssen 2 Register in den Modbus-Slave geschrieben werden.</i>	
		<i>Word order</i>	<i>HI-LO</i>	<i>HI-Word auf der niedrigeren Registeradresse, LO-Word auf der höheren Registeradresse</i>
		<i>LO-HI</i>	<i>LO-Word auf der niedrigeren Registeradresse, HI-Word auf der höheren Registeradresse</i>	
<i>Unsigned</i>	<i>Der zu schreibende Integer-Wert ist nicht vorzeichenbehaftet.</i>			
	<i>Bit</i>	16	<i>16-Bit Integer</i>	
		32	<i>32-Bit Integer. Dazu müssen 2 Register in den Modbus-Slave geschrieben werden.</i>	
		<i>Word order</i>	<i>HI-LO</i>	<i>HI-Word auf der niedrigeren Registeradresse, LO-Word auf der höheren Registeradresse</i>
		<i>LO-HI</i>	<i>LO-Word auf der niedrigeren Registeradresse, HI-Word auf der höheren Registeradresse</i>	
<i>Float</i>	<i>Es soll ein 32Bit Float geschrieben werden.</i>			
	<i>Word order</i>	<i>HI-LO</i>	<i>HI-Word auf der niedrigeren Registeradresse, LO-Word auf der höheren Registeradresse</i>	
		<i>LO-HI</i>	<i>LO-Word auf der niedrigeren Registeradresse, HI-Word auf der höheren Registeradresse</i>	

9.2.1.9.2.3 Com2 (Seriell, RS232)

Spalte

Durch das "Zahlen Trennzeichen" (siehe "Schnittstellen" auf Seite 76) wird der ASCII-Ausgabestring in einzelne Ausgabewerte unterteilt. Der Konfigurationsparameter „Spalte“ gibt an, an welche Position der Ausgabewert im ASCII-Ausgabestring gesetzt wird.

9.2.1.9.2.4 Device Logic

Für den Modus "Device Logic" ist die Konfiguration nicht verfügbar.

9.2.1.9.3 Skalierung

Hinweis: Für einen Interface Ausgabekanal, der mit einer Schnittstelle verbunden ist für die "Device Logic Parsing" aktiviert wurde (siehe "Basis" auf Seite 76), sind die folgenden Parameter nicht verfügbar.

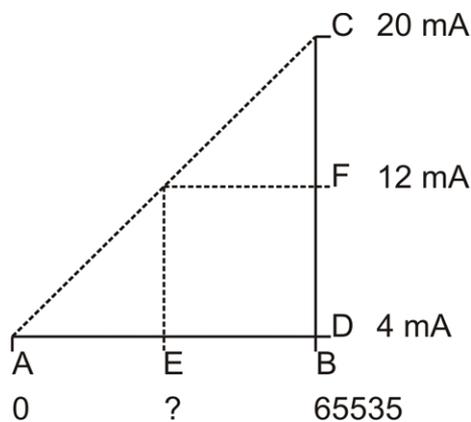
9.2.1.9.3.1 Com1 und Com3 (Modbus, RS485)

Skalierung	aus	--	
	ein	0% Modbus	Start des Ausgabebereichs in der Einheit des Modbus-Slaves
		100% Modbus	Ende des Ausgabebereichs in der Einheit des Modbus-Slaves
		0%	Start des Ausgabebereichs in der Messeinheit
		100%	Ende des Ausgabebereichs in der Messeinheit

Hinweis: Berechnung des Ausgabewerts bei aktiver Skalierung:

Basis > Min	4mA	D
Basis > Max	20mA	C
Min Modbus	0	A
Max Modbus	65535	B

Basis > Stellwert	12mA	F
Skalierter Ausgabewert	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 (Seriell, RS232)

Die Skalierung ist für die serielle Schnittstelle nicht verfügbar.

9.2.1.9.3.3 Device Logic

Für den Modus "Device Logic" ist die Skalierung nicht verfügbar.

9.2.1.10 Interface Ausgabekanäle 33-64

Bezeichnung 33-64

frei wählbare Kanalbezeichnung für die Interface Ausgabekanäle

Die Bedeutung der restlichen Konfigurationsparameter in diesem Konfigurationsabschnitt entspricht dem Konfigurationsabschnitt "Interface Ausgabekanäle 1-32" (siehe "Interface Ausgabekanäle 1-32" auf Seite 115).

9.2.1.11 Berechnete Kanäle

Hinweis: Die Werte der berechneten Kanäle werden jedes Mal direkt bei der Datenausgabe (Anzeige am myDatanet-Server oder Download vom myDatanet-Server) berechnet. Sie sind nicht in der Datenbank des Servers gespeichert.

9.2.1.11.1 Basis

Bezeichnung 1-5

frei wählbare Kanalbezeichnung für die berechneten Kanäle [0-16 Zeichen]

Modus

mögliche Berechnungsmodi für die berechneten Kanäle

<i>aus</i>	---	<i>berechneter Kanal deaktiviert</i>
<i>Tabelle</i>	<i>Min</i>	<i>definiert das untere Skalenende der Zeigerinstrumente</i>
	<i>Max</i>	<i>definiert das obere Skalenende der Zeigerinstrumente</i>
	<i>Einheit</i>	<i>String, der als Messwerteinheit von allen Anzeigeelementen des Servers verwendet wird [0-16 Zeichen]. Dieser hat keinen direkten Einfluss auf die Werte.</i>
	<i>Nachkomma</i>	<i>Anzahl der Nachkommastellen, die von allen Anzeigeelementen des Servers verwendet wird</i>
<i>digital</i>	<i>Invertieren</i>	<i>invertiert das Eingangssignal</i>
<i>+, -, x, /</i>	<i>Min</i>	<i>definiert das untere Skalenende der Zeigerinstrumente</i>
	<i>Max</i>	<i>definiert das obere Skalenende der Zeigerinstrumente</i>
	<i>Einheit</i>	<i>String, der als Messwerteinheit von allen Anzeigeelementen des Servers verwendet wird [0-16 Zeichen]. Dieser hat keinen direkten Einfluss auf die Werte.</i>
	<i>Nachkomma</i>	<i>Anzahl der Nachkommastellen, die von allen Anzeigeelementen des Servers verwendet wird</i>
<i>Delta</i>	<i>ermittelt die Differenz zwischen zwei Messwerten und dividiert das Ergebnis durch die Zeitdifferenz der Zeitstempel der Messwerte. Die Zeiteinheit (Wert/sec., Wert/min., ...) für das Ergebnis lässt sich über den Parameter "Zeitbasis", der sich im Tab "Berechnung" befindet, auswählen. Damit ist es z.B. möglich, den Zählerstand (m³) des Quellkanals in einen Durchfluss (m³/min.) umzurechnen (siehe "Ergänzende Erklärung: Modus Delta" auf Seite 124).</i>	
	<i>Min</i>	<i>definiert das untere Skalenende der Zeigerinstrumente</i>
	<i>Max</i>	<i>definiert das obere Skalenende der Zeigerinstrumente</i>
	<i>Einheit</i>	<i>String, der als Messwerteinheit von allen Anzeigeelementen des Servers verwendet wird [0-16 Zeichen]. Dieser hat keinen direkten Einfluss auf die Werte.</i>
	<i>Nachkomma</i>	<i>Anzahl der Nachkommastellen, die von allen Anzeigeelementen des Servers verwendet wird</i>
<i>klonen</i>	<i>erstellt einen Klon eines Messkanals. Dadurch ist es möglich, eine neue Kanalbezeichnung zu vergeben, andere Skalenenden für die Zeigerinstrumente zu wählen, einen neuen String als Messwerteinheit festzulegen sowie die Anzahl der Nachkommastellen anzupassen. Die Messwerte (Zahlenwert ohne Einheit) entsprechen exakt jenen des Quellkanals.</i>	
	<i>Min</i>	<i>definiert das untere Skalenende der Zeigerinstrumente</i>
	<i>Max</i>	<i>definiert das obere Skalenende der Zeigerinstrumente</i>
	<i>Einheit</i>	<i>String, der als Messwerteinheit von allen Anzeigeelementen des Servers verwendet wird [0-16 Zeichen]. Dieser hat keinen direkten Einfluss auf die Werte.</i>
	<i>Nachkomma</i>	<i>Anzahl der Nachkommastellen, die von allen Anzeigeelementen des Servers verwendet wird</i>



Element nach unten verschieben



Element nach oben verschieben

9.2.1.11.2 Berechnung

aus	---	berechneter Kanal deaktiviert
Tabelle	Quellkanal	Auswahl des Kanals, aus dem die Eingangsdaten herangezogen werden
		öffnet die Maske zur Eingabe der Wertetabelle (zwischen den Tabellenzeilen wird linear interpoliert, für Werte außerhalb der definierten Tabelle wird linear extrapoliert)
Digital	Quellkanal	Auswahl des Kanals, aus dem die Eingangsdaten herangezogen werden
	High Level	Schwelle für Signalerkennung
+, -, x, /	Quellkanal	Auswahl des Kanals, aus dem die Eingangsdaten herangezogen werden
	+, -, x, /	
	Quellkanal	Auswahl des Kanals, aus dem die Eingangsdaten herangezogen werden
Delta	Quellkanal	Auswahl des Kanals, aus dem die Eingangsdaten herangezogen werden
	Zeitbasis	gibt die gewünschte Zeiteinheit (Wert/sec., Wert/min., ...) für das Ergebnis an
	Offset	Offset, der nach der Multiplikation mit dem Parameter "Faktor" addiert wird. Vor der Multiplikation mit dem Parameter "Faktor" wird das Ergebnis in die gewünschte Zeiteinheit umgewandelt.
	Faktor	Faktor, mit dem das Ergebnis nach der Umwandlung in die gewünschte Zeiteinheit multipliziert wird. Danach wird der Parameter "Offset" addiert.
Klonen	Quellkanal	Auswahl des Kanals, der geklont werden soll

Hinweis:

Ergänzende Erklärung: Modus Delta

Annahme: Der Quellkanal enthält den Zählerstand eines Endloszählers in m^3 . Der berechnete Kanal 1 soll den Durchfluss in m^3/s enthalten und der berechnete Kanal 2 den Durchfluss in l/h .

Erforderliche Konfiguration

Parameter	Wert Kanal 1	Wert Kanal 2
Basis -> Modus	Delta	Delta
Basis -> Einheit	m^3/s	l/h
Berechnung -> Zeitbasis	Sekunden	Stunden
Berechnung -> Offset	0	0
Berechnung -> Faktor	1	1000

Quellkanal		Berechneter Kanal 1	Berechneter Kanal 2
Datum/Zeit	Endloszähler [m^3]	Durchfluss [m^3/s]	Durchfluss [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⁾Berechnung nicht möglich, da kein Messwert vor 12:50 vorhanden ist.

Erklärung: Für den Messzeitpunkt 12:50 können keine Werte für die berechneten Kanäle ermittelt werden, da kein Vorgängerwert vorhanden ist und somit die Differenz des Zählerstandes nicht ermittelt werden kann. Für den Messzeitpunkt 12:51 beträgt die Differenz des Zählerstandes $60m^3$ und die Zeitdifferenz 60sec.

Ergebnis = { (Wertdifferenz / Zeitdifferenz [sec.]) * Zeitbasis [sec] * Faktor } + Offset

Für den berechneten Kanal 1 (Zeitbasis "Sekunden", Offset "0" und Faktor "1") errechnet sich das Ergebnis wie folgt:

$$\text{Kanal 1} = \{ (60m^3 / 60sec.) * 1 * 1 \} + 0 = 1m^3/s$$

Für den berechneten Kanal 2 (Zeitbasis "Stunden", Offset "0" und Faktor "1000") errechnet sich das Ergebnis wie folgt:

$$\text{Kanal 2} = \{ (60m^3 / 60sec.) * 3600 * 1000 \} + 0 = 3.600.000l/h$$

9.2.1.11.3 Alarme

Hinweis: Die Überprüfung der Alarmschwellen kann bei berechneten Kanälen erst erfolgen, wenn das Gerät die Messdaten an den myDatenet-Server übermittelt hat.

Alarm niedrig	Wenn der Messwert auf oder unter diesen Wert fällt, wird ein Alarm ausgelöst.
Alarm hoch	Wenn der Messwert auf oder über diesen Wert steigt, wird ein Alarm ausgelöst.
Hyst %	Hysterese für Entwarnung bei Alarm/Warnung (z.B. Hyst = 5%, Alarm od. Warnung bei 100 -> Entwarnung bei 95) bzw. zurücknehmen des Triggers (z.B. Hyst = 5%, Schwelle = größer gleich, Trigger bei 100 -> zurücknehmen bei 95)

9.2.1.12 Ausgabekanäle

9.2.1.12.1 Basis

Ext. Aufwärmzeit

Gibt die Zeit an, die ein Ausgabekanal, der in den Modus "Ext. Aufwärmzeit" geschaltet ist, vor der Messung eingeschaltet wird

Bezeichnung mA OUT 1-2

frei wählbare Kanalbezeichnung für die Analogausgänge (keine galvanische Trennung) [0-16 Zeichen]

Bezeichnung Relais 1-6

frei wählbare Kanalbezeichnung für die Relais [0-16 Zeichen] (Je 3 Relais mit gemeinsamer Wurzel)

Modus

Basiseinstellung für die Analogausgänge

<i>aus</i>	---	<i>Ausgabekanal deaktiviert</i>
<i>0-20mA</i>	<i>0%</i>	<i>Start des Ausgabebereichs in der Messeinheit</i>
	<i>100%</i>	<i>Ende des Ausgabebereichs in der Messeinheit</i>
	<i>Stellwert</i>	<i>Ausgabewert in der Messeinheit</i>
	<i>Einheit</i>	<i>String, der als Messwerteinheit von allen Anzeigeelementen des Servers verwendet wird [0-16 Zeichen]</i>
	<i>Nachkomma</i>	<i>Anzahl der Nachkommastellen, die von allen Anzeigeelementen des Servers verwendet wird</i>
<i>4-20mA</i>	<i>0%</i>	<i>Start des Ausgabebereichs in der Messeinheit</i>
	<i>100%</i>	<i>Ende des Ausgabebereichs in der Messeinheit</i>
	<i>Stellwert</i>	<i>Ausgabewert in der Messeinheit</i>
	<i>Einheit</i>	<i>String, der als Messwerteinheit von allen Anzeigeelementen des Servers verwendet wird [0-16 Zeichen]</i>
	<i>Nachkomma</i>	<i>Anzahl der Nachkommastellen, die von allen Anzeigeelementen des Servers verwendet wird</i>

Basiseinstellung für die Relais

<i>aus</i>	---	<i>Ausgabekanal deaktiviert</i>
<i>Ext. Aufwärmzeit</i>	<i>Der Ausgabekanal wird "Ext. Aufwärmzeit" Sekunden vor der Messung eingeschaltet. Ist der Wert "0", wird der Ausgabekanal nicht eingeschaltet.</i>	
<i>Digital</i>	<i>Invertieren</i>	<i>invertiert den am Gerät ausgegebenen Pegel (siehe "Ergänzende Erklärung Modus "Digital"" auf Seite 126)</i>
	<i>Stellwert</i>	<i>Stellwert (ein/aus), der ausgegeben werden soll (siehe "Ergänzende Erklärung Modus "Digital"" auf Seite 126)</i>

Hinweis:

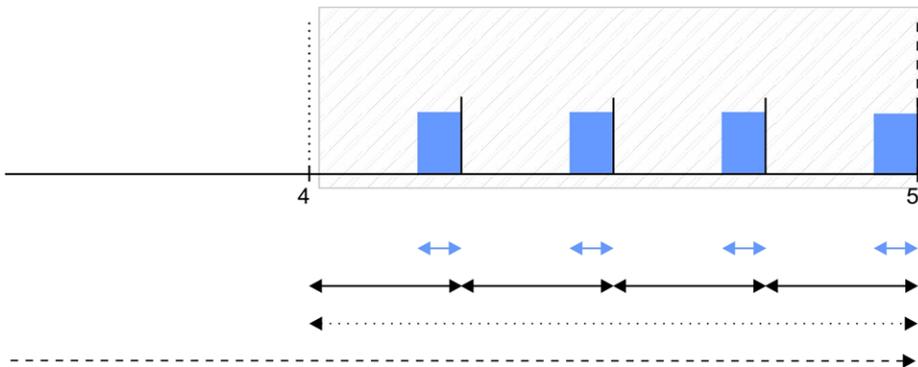
Ergänzende Erklärung Modus "Digital"

Invertieren	Stellwert		Ausgang am Gerät
<i>aus</i>	<i>aus</i>	=	<i>aus (Low)</i>
<i>aus</i>	<i>ein</i>	=	<i>ein (High)</i>
<i>ein</i>	<i>aus</i>	=	<i>ein (High)</i>
<i>ein</i>	<i>ein</i>	=	<i>aus (Low)</i>

Hinweis:

Beispiel zur Erklärung des Burstintervalls in Kombination mit der ext. Aufwärmzeit (Ext. Aufwärmzeit < Messintervall):

Grundeinstellung		Aufzeichnungsintervall	5min.
		Burst Intervall	60sec.
		Messintervall	15sec.
Ausgabekanäle		Ext. Aufwärmzeit	5sec.
Messkanäle -> Konfig.		Dämpfung	med
		Zeit	60sec.
Ausgang am Gerät		Sensorversorgung	

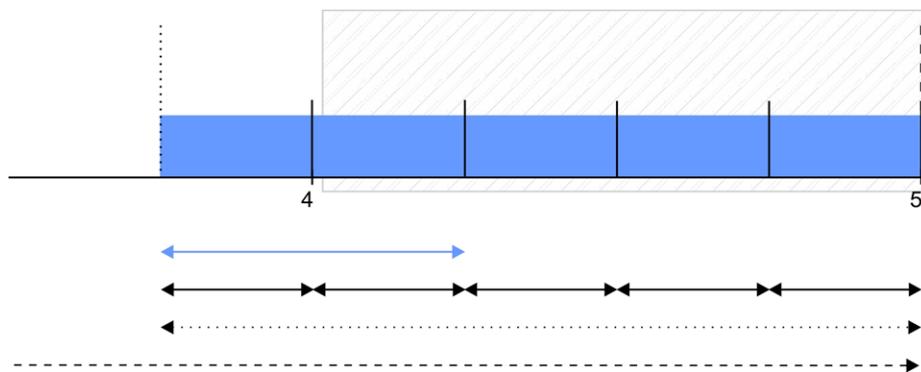


Messwertbildung: Die Sensorversorgung wird jeweils 5sec. vor Ablauf des Messintervalls aktiviert. Dadurch ergeben sich 4 gültige Messungen, die für die Median-Bildung herangezogen und als Messwert aufgezeichnet werden.

Hinweis:

Beispiel zur Erklärung des Burstintervalls in Kombination mit der ext. Aufwärmzeit (Ext. Aufwärmzeit > Messintervall):

Grundeinstellung		Aufzeichnungsintervall	5min.
		Burst Intervall	75sec.
		Messintervall	15sec.
Ausgabekanäle		Ext. Aufwärmzeit	30sec.
Messkanäle -> Konfig.		Dämpfung	med
		Zeit	60sec.
Ausgang am Gerät		Sensorversorgung	



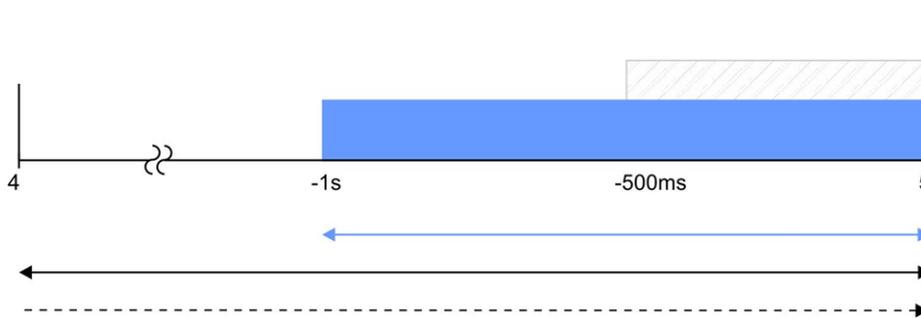
Erklärung: Da in diesem Beispiel die ext. Aufwärmzeit größer als das Messintervall ist, wird die Sensorversorgung gleich zu Beginn des Burst Intervalls aktiviert und erst bei der Aufzeichnung wieder deaktiviert. Die erste Messung wird 15sec. (Messintervall) nach Beginn des Burst Intervalls durchgeführt. Zu diesem Zeitpunkt war die Sensorversorgung noch nicht für die Dauer der ext. Aufwärmzeit aktiv. D.h. der Messwert des angeschlossenen Sensors ist wahrscheinlich noch nicht stabil und die Messung somit ungültig. In diesem Beispiel ist erst die zweite Messung nach Beginn des Burst Intervalls gültig. Um die ungültigen Messungen bei der Messwertbildung auszuschließen, muss das Zeitfenster für die Dämpfung entsprechend kleiner als das Burst Intervall gewählt werden. Um genau wie beim vorangegangenen Beispiel (Ext. Aufwärmzeit < Messintervall) 4 gültige Messungen zu erhalten, muss das Burst Intervall von 60sec. auf 75sec. erhöht werden.

Messwertbildung: Die Sensorversorgung wird zu Beginn des Burst Intervalls aktiviert. Die erste Messung wird 15sec. später durchgeführt. Es ergeben sich somit 5 Messungen. Die erste Messung ist ungültig (ext. Aufwärmzeit ist noch nicht abgelaufen). Da das Zeitfenster für die Dämpfung nur 60sec. beträgt, werden nur die 2-5te Messung für die Median-Bildung herangezogen und als Messwert aufgezeichnet. Die erste ungültige Messung wird somit ignoriert.

Hinweis:

Beispiel zur Erklärung der Filter Zeit in Kombination mit der ext. Aufwärmzeit

Grundeinstellungen		Aufzeichnungsintervall	5min.
		Messintervall	1min.
Ausgabekanäle		Ext. Aufwärmzeit	1sec.
Messkanäle -> Konfig.		Filter Zeit	500ms
Ausgang am Gerät		Sensorversorgung	



Erklärung: Die Sensorversorgung wird jeweils 1sec. vor Ablauf des Messintervalls aktiviert. Die Filter Zeit beginnt 500ms vor Ablauf des Messintervalls, wodurch zum Messzeitpunkt ein gültiger Wert vorliegt. Dies bedeutet aber auch, dass bei der Auswahl der Ext. Aufwärmzeit die Filter Zeit berücksichtigt werden muss. Im aktuellen Beispiel müsste der mit dem Eingang verbundene Sensor spätestens 500ms nach Aktivierung der Sensorversorgung stabile Werte liefern. Andernfalls würden bei der Mittelwertbildung während der Filter Zeit ungültige Werte berücksichtigt und so der Messwert verfälscht werden.

9.2.1.13 Interne Kanäle

9.2.1.13.1 Basis

Bezeichnung GSM Stärke	frei wählbare Kanalbezeichnung für die GSM-Feldstärke [0-16 Zeichen]	
	Einheit	String, der als Messwerteinheit von allen Anzeigeelementen des Servers verwendet wird [0-16 Zeichen]. Dieser hat keinen direkten Einfluss auf die Werte.
Bezeichnung Spannung	frei wählbare Kanalbezeichnung für die Versorgungsspannung [0-16 Zeichen]	
	Einheit	String, der als Messwerteinheit von allen Anzeigeelementen des Servers verwendet wird [0-16 Zeichen]. Dieser hat keinen direkten Einfluss auf die Werte.

9.2.1.13.2 Alarme

Warnung	Wert niedrig	Wenn der Messwert auf oder unter diesen Wert fällt, wird eine Warnung ausgelöst.
	Wert hoch	Wenn der Messwert auf oder über diesen Wert steigt, wird eine Warnung ausgelöst.
Alarm	Wert niedrig	Wenn der Messwert auf oder unter diesen Wert fällt, wird ein Alarm ausgelöst.
	Wert hoch	Wenn der Messwert auf oder über diesen Wert steigt, wird ein Alarm ausgelöst.
Hyst %	Hysterese für Entwarnung bei Alarm/Warnung (z.B. Hyst = 5%, Alarm od. Warnung bei 100 -> Entwarnung bei 95) bzw. zurücknehmen des Triggers (z.B. Hyst = 5%, Schwelle = größer gleich, Trigger bei 100 -> zurücknehmen bei 95)	

9.2.1.13.3 Trigger

Es wird zwischen den folgenden beiden Arten von Triggern unterschieden:

- Eventtrigger (XM)

Die entsprechende Operation (z.B. Übertragung auslösen) wird beim Auftreten des Triggerereignisses nur ein einziges Mal ausgeführt.
- Leveltrigger (QU, SL, RO, RF, ON, I1-I4)

Solange der Trigger aktiv ist, wird die entsprechende Operation (z.B. Online-Modus aktivieren) ausgeführt.

Eventtrigger	XM	Übertragung auslösen	
	Schwelle	Schwellen für das Auslösen des Triggers. Für die Ermittlung der Schwelle zum Zurücksetzen des Triggers wird die Hysterese aus dem Tab "Alarme" verwendet.	
		größer gleich	Wenn der Messwert auf oder über diesen Wert steigt, wird der Trigger ausgelöst.
		kleiner gleich	Wenn der Messwert auf oder unter diesen Wert fällt, wird der Trigger ausgelöst
Leveltrigger	QU	schnelle Aufzeichnung (Aufzeichnungsintervall = Aufzeichnungsintervall / Faktor)	
	SL	langsame Aufzeichnung (Aufzeichnungsintervall = Aufzeichnungsintervall * Faktor)	
	RO	Aufzeichnung einschalten	
	RF	Aufzeichnung ausschalten	
	ON	Online-Modus aktivieren	
	I1	intern zur Verwendung mit Steuerprogramm	
	I2	intern zur Verwendung mit Steuerprogramm	
	I3	intern zur Verwendung mit Steuerprogramm	
	I4	intern zur Verwendung mit Steuerprogramm	
	Schwelle	Schwellen für das Auslösen des Triggers. Für die Ermittlung der Schwelle zum Zurücksetzen des Triggers wird die Hysterese aus dem Tab "Alarme" verwendet.	
		größer gleich	Der Trigger ist aktiv, solange der Messwert größer/gleich der Schwelle ist.
kleiner gleich		Der Trigger ist aktiv, solange der Messwert kleiner/gleich der Schwelle ist.	

9.2.1.14 Alarmierung

Quittierung	Standard	Für die Entscheidung, ob die Alarme automatisch oder manuell quittiert werden müssen, wird die globale Servereinstellung herangezogen (siehe "Benutzerhandbuch für myDatenet-Server " 206.886).
	automatisch	Alarme werden automatisch quittiert, sobald alle Benachrichtigungen versendet wurden. Wurden auch SMS versendet, die einen Tarif mit Sendebestätigungsfunktion haben, so wird mit der Quittierung auf die Sendebestätigung gewartet.
	manuell	Alarme müssen durch den Anwender quittiert werden.
Übertragungsausfall Alarm	Alarmierung, falls sich das Instrument länger als die eingestellte Anzahl von Übertragungszyklen nicht meldet. Pro Übertragungszyklus wird eine zusätzliche Toleranz von 10min. eingeräumt, um etwaige Retrys beim Verbindungsaufbau zu berücksichtigen. Bsp.: Übertragungsintervall: 60min; 3x Übertragungsintervall -> Alarm nach > 03:30	
Transfervolumen	Standard	Die Einstellung für den Transfervolumenalarm wird von der globalen Servereinstellung übernommen (siehe "Benutzerhandbuch für myDatenet-Server " 206.886).
	aus	Der Transfervolumenalarm ist deaktiviert.
	individuell	Die Schwelle, bei der der Transfervolumenalarm ausgelöst werden soll, kann in das nebenstehende Feld in KiB eingegeben werden.
Bei Alarm	A	Der Alarm wird in der Alarmliste aufgezeichnet.
	Ü	Eine sofortige Übertragung wird ausgelöst.
Bei Warnung	A	Die Warnung wird in der Alarmliste aufgezeichnet.
	Ü	Eine sofortige Übertragung wird ausgelöst.
Bei Störung-Alarm ¹⁾	A	Der Störung-Alarm wird in der Alarmliste aufgezeichnet.
	Ü	Eine sofortige Übertragung wird ausgelöst.
Bei Störung-Warnung ²⁾	A	Die Störung-Warnung wird in der Alarmliste aufgezeichnet.
	Ü	Eine sofortige Übertragung wird ausgelöst.

¹⁾ Ein "Störung-Alarm" wird ausgelöst, wenn ein Messkanal so konfiguriert ist, dass bei einer Messbereichsverletzung einer der Fehlerwerte NAN, OL (Open Loop), UF (Under Flow), OF (Over Flow) oder SC (Short Circuit) ausgegeben wird und das Halten des letzten Messwert nicht aktiviert ist oder bereits die maximale Anzahl fürs Halten des letztgültigen Messwerts erreicht wurde.

²⁾ Eine "Störung-Warnung" wird ausgelöst, wenn ein Messkanal so konfiguriert ist, dass bei einer Messbereichsverletzung einer der Fehlerwerte NAN, OL (Open Loop), UF (Under Flow), OF (Over Flow) oder SC (Short Circuit) ausgegeben wird und die maximale Anzahl fürs Halten des letztgültigen Messwerts noch nicht erreicht wurde.

9.2.1.15 Grundeinstellung

Verbindungsart	Intervall	Das Gerät meldet sich im Übertragungsintervall.
	online	Das Gerät trennt die Verbindung nicht und übermittelt kontinuierlich die Messdaten. Alle 7 Tage wird die Verbindung jedoch kurzzeitig zwecks Überprüfung der Serverzuordnung unterbrochen.
Aloha/Wakeup Dauer	Dauer der Aloha/Wakeup Verbindung	
Übertragungsintervall	zeitlicher Abstand der Übertragungen	
Aufzeichnungsintervall	zeitlicher Abstand der Messdatenaufzeichnungen	
Divisor schnelle Aufzeichnung	Aufzeichnungsintervall = Aufzeichnungsintervall / Faktor (ab Triggerung)	
Faktor langsame Aufzeichnung	Aufzeichnungsintervall = Aufzeichnungsintervall * Faktor (ab Triggerung)	
Messintervall	zeitlicher Abstand der Messungen (00:00 gleich wie Aufzeichnungsintervall)	
Burst Intervall	Zeitspanne, in der im Messintervall gemessen wird, bevor aufgezeichnet wird (00:00 Messintervall ist durchgehend aktiviert)	
Zeitzone	Regionseinstellungen (nicht relevant für Rohmessdaten, da diese in UTC gespeichert werden)	
Verzögerung für Ausfallsalarm	Erst wenn die Versorgungsspannung länger als die hier konfigurierte Zeit ausfällt, wird ein entsprechender Alarm erzeugt. Dadurch kann das Generieren von Alarmen bei kurzzeitigen Versorgungsspannungsausfällen verhindert werden. Ein Eintrag im Gerätelog wird aber in jedem Fall erstellt.	
Sommerzeit	Konfiguration für die automatische Zeitumstellung	
	standard	Die Konfiguration für die Zeitumstellung wird von der globalen Servereinstellung übernommen(siehe "Benutzerhandbuch für myDatanet-Server " 206.886).
	aus	automatische Zeitumstellung deaktiviert
	USA	vordefinierte Einstellung für den amerikanischen Raum
	EU	vordefinierte Einstellung für den europäischen Raum
Positionsintervall	Intervall der Positionsaktualisierung (00:00 Positionsbestimmung wird bei jeder Verbindung durchgeführt)	
Standard Auswertung	Auswahl der Auswertung, die durch einen Klick auf den Gerätelink in den Karten geladen wird	
	aus	Die Standardgrafik wird geladen.
	"Name einer Auswertung"	Die ausgewählte Auswertung wird geladen.

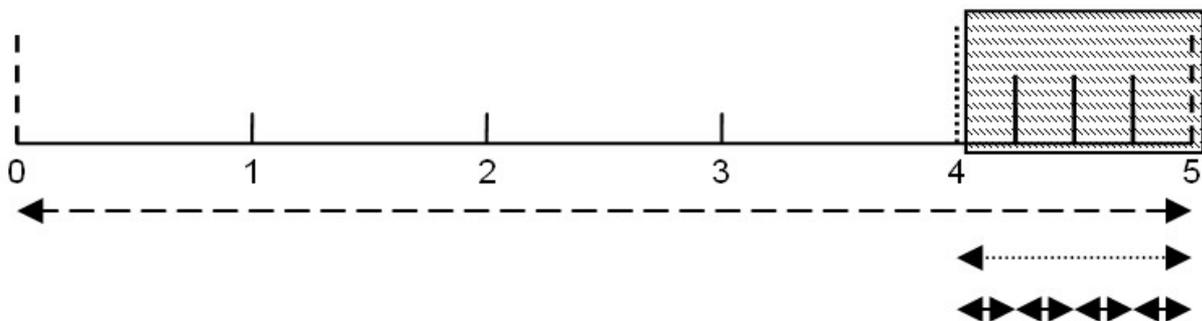
Auswertungs-Vorlage	Auswahl, ob beim Klicken auf das Symbol zur Anzeige der Messdaten, das sich in der Messstellen-/Applikationsliste befindet, die standardmäßige Grafik oder eine Auswertungs-Vorlage zur Darstellung der Daten verwendet wird. In der Dropdown-Liste werden nur jene Auswertungs-Vorlagen angezeigt, bei denen der Messstellen-/Applikationstyp der ersten Wildcard kompatibel zur Messstelle/Applikation ist, die aktuell bearbeitet wird.	
	(nicht zugeordnet)	Die standardmäßige Grafik wird für die Anzeige der Messdaten verwendet.
	"Name einer Auswertungs-Vorlage"	Name der Auswertungs-Vorlage, die zur Darstellung der Messdaten verwendet wird

Hinweis:

Beispiel zur Erklärung Aufzeichnungs-, Mess-, Burstintervall in Verbindung mit der Dämpfung

Da nur die Universaleingänge über ein Dämpfungsmodul (siehe "Interne Verarbeitung der Messwerte" auf Seite 31) verfügen, ist die Verwendung des Burstintervalls nur für diese Messkanäle sinnvoll. Eine Erklärung der Abläufe, wenn zusätzlich noch eine ext. Aufwärmzeit verwendet werden soll, finden Sie im Kapitel "Ausgabekanäle" auf Seite 125.

Grundeinstellung		Aufzeichnungsintervall	5min.
		Burst Intervall	1min.
		Messintervall	15sec.
Messkanäle -> Konfig.		Dämpfung	med
		Zeit	60sec.



Messwertbildung: Die letzten 4 Messwerte werden für die Median Bildung herangezogen und als Messwert aufgezeichnet.

Hinweis: Ergänzende Erklärung zu den Verbindungsarten

Verbindungsart	Energieverbrauch	Datenvolumen	Reaktionszeit
online			
Intervall & Wakeup			
Intervall			

Hinweis: Ergänzende Erklärung zu den Auswirkungen des Messintervalls und des Übertragungsintervalls auf das monatliche Datenvolumen

Messintervall/Übertragungsintervall	Datenvolumen pro Monat
<i>Nur die 8 Universaleingänge sind aktiv (nur geringe Änderung des Messsignals)</i>	
2min/10min	13,0MB
2min/2h	2,0MB
1min/2h	2,5MB
1min/4h	2,0MB
5min/2h	1,2MB
5min/online	4,0MB
<i>Nur die 64 Interface Kanäle sind aktiv (stark ändernde Werte (Zufallswerte))</i>	
1min/4h	18,6MB
1min/2h	19,0MB

9.2.1.16 FTP-Export Einstellungen

Hinweis: Dieser Konfigurationsabschnitt ist nur sichtbar, wenn die Lizenz "FTP Agent Extended" für den myDatenet-Server freigeschaltet wurde.

FTP Export Profil	aus	FTP Export deaktiviert
	"Name eines FTP Export Profils"	Liste mit den FTP-Export-Profilen, die am Server angelegt wurden (zum Anlegen eines FTP-Export-Profiles siehe "Benutzerhandbuch für myDatenet-Server " 206.886).
Einstellungen des gewählten Profils	zeigt eine Übersicht der wichtigsten Parameter des ausgewählten FTP-Export-Profiles an	
FTP Verzeichnis	ermöglicht es, das Standardverzeichnis des ausgewählten FTP-Export-Profiles zu überschreiben [0-100 Zeichen]	
letzter Export	Zeitstempel des letzten FTP Exportes	

9.2.2 Gerätekonfiguration

Hinweis: Abhängig vom jeweiligen Benutzerlevel sind einige der in den folgenden Unterkapiteln erwähnten Konfigurationsfelder unter Umständen ausgeblendet. Wenden Sie sich in diesem Fall an den Administrator des myDatenet-Servers.

Die Eingabemaske zur Konfiguration des Geräts erreichen Sie durch Klicken auf die Seriennummer in der Messstellenliste (siehe "Benutzerhandbuch für myDatenet-Server " 206.886) oder durch Klicken auf den Gerätenamen in der Messgeräteliste (siehe "Benutzerhandbuch für myDatenet-Server " 206.886).

9.2.2.1 Kommentar

Kommentar

freies Kommentarfeld (wird auch unterhalb des Messstellennamens in der Messgeräteliste angezeigt)

9.2.2.2 Messgerät

Kunde	Name des Kunden, dem das Messgerät zugeordnet ist	
Tags	Liste der Tags, die dem Messgerät bereits zugewiesen sind. Durch einen Klick auf das Kreuz neben der Bezeichnung des Tags kann diese Zuweisung wieder aufgehoben werden. Durch Klicken auf das Plus-Symbol wird die Eingabemaske zur Zuweisung der Tags geöffnet. Diese ermöglicht sowohl die Zuweisung vorhandener als auch die Erstellung neuer Tags.	
Seriennummer	Seriennummer des Geräts	
Geräteklasse	Damit ein Gerät mit einer Messstelle verbunden werden kann, müssen die Geräteklasse der Messstelle und die des Geräts übereinstimmen. Die Geräteklasse kann nach dem Anlegen des Geräts über die Serveroberfläche nur bis zur ersten Verbindung des Geräts mit dem Server verändert werden. Sollte beim Anlegen des Geräts eine Geräteklasse eingestellt werden, die nicht mit der tatsächlichen Geräteklasse des Geräts übereinstimmt, wird diese bei der ersten Verbindung automatisch korrigiert.	
Telefonnummer	Telefonnummer der SIM-Karte. An diese Nummer werden die Steuer-SMS (z.B. Wakeup) gesendet. Format: +43555837465	
Geräte Flags	zusätzliche Information zur Geräteklasse (für interne Verwendung)	
Firmware Version	aktuell installierte Softwareversion des Messcontrollers	
Modem Version	aktuell installierte Softwareversion des Modemcontrollers	
OS Version	OS Version des Modems	
Letzter Verbindungsaufbau	jeweils der letzte Zeitstempel der betreffenden Operation	
Letzter Wakeup		
Letzter Verbindungsabbau		
Letzter Übertragungsfehler		
Letzte Aloha Verbindung		
Wakeup SMS Anzahl	Anzahl der seit der letzten Verbindung an dieses Gerät gesendeten Wakeup-SMS. Bei jeder erfolgreich hergestellten Verbindung wird dieser Zähler zurückgesetzt.	
Device Logic Sync	Produktiv	Stimmen die im Gerät installierte und die am Server gespeicherte Device Logic nicht überein, wird die am Server gespeicherte Device Logic in das Gerät geladen.
	Entwicklung (sync)	Es erfolgt eine Synchronisation der Device Logic zwischen dem Gerät und dem Server. Dabei wird jenes mit dem aktuellsten Zeitstempel zur jeweils anderen Stelle übertragen.
	Entwicklung (no sync)	Es erfolgt keine Synchronisation der Device Logic zwischen dem Gerät und dem Server

Firmware Update	aus	Firmware Update ist deaktiviert
	ein	Sobald eine neue Version des ausgewählten Firmware-Typs vorhanden ist, wird diese sofort installiert.
	auch wenn tag nicht vorhanden	Firmware wird auch ans Gerät übertragen, wenn das Gerät den aktuellen Firmwarestand nicht an den Server übermittelt hat (NICHT EMPFOHLEN!).
	Downgrade erlauben	ermöglicht es, eine ältere Firmwareversion als die im Gerät vorhandene zu installieren (NICHT EMPFOHLEN!)
	einmalig	Führt einmalig ein Firmware Update durch. Ist keine neue Firmware verfügbar oder wurde die Firmware erfolgreich installiert, wird das Firmware Update automatisch auf "aus" geschaltet.
	ignorieren	Das Firmware Update ist deaktiviert und auf verfügbare Firmware Updates wird nicht hingewiesen.
Firmware Typ	Released	Nur Firmwareversionen bei denen sowohl interner Test als auch Feldtest erfolgreich waren, werden installiert (Fehlfunktionen nahezu ausgeschlossen).
	Release Candidate	Nur Firmwareversionen bei denen der interne Test erfolgreich war, werden installiert (Fehlfunktionen nicht ausgeschlossen).
	Beta Release	Auch Firmwareversionen bei denen noch nicht alle internen Tests erfolgreich abgeschlossen sind, werden installiert (Fehlfunktionen durchaus möglich).
Hardware Version	Hardwareversion des myDatalogMUC xG/4G	

9.2.2.3 Gerätespezifische Einstellungen

Betriebsart	hold	Messung: aus, Übertragung: ein
	run	Messung: ein, Übertragung: ein
	transport	Messung: aus, Übertragung: aus
	offline	Messung: ein, Übertragung: aus

9.2.2.4 GPRS

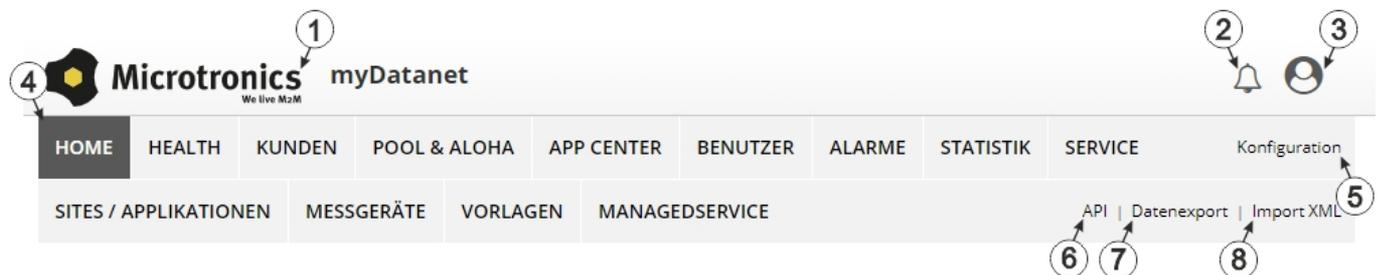
SIM Tarif

ausgewählter SIM-Tarif

Kapitel 10 myDatenet-Server

Hinweis: Alle Screenshots zeigen den myDatenet-Server in der Version 49v011 unter Verwendung des Standard-Farbschemas. Bei neueren Versionen können geringfügige Änderungen am Erscheinungsbild des Servers vorgenommen worden sein.

10.1 Übersicht



Übersicht myDatenet-Server

1 frei wählbares Logo	5 öffnet die Maske zur Eingabe der globalen Einstellungen für den Server
2 öffnet das Fenster in dem die für den aktuell eingeloggtten Benutzer bestimmten, vom System erstellten Benachrichtigungen zusammengefasst sind	6 öffnet den rapidM2M Playground
3 blendet das Menü zum Anpassen der Benutzereinstellungen und zum Ausloggen des aktuell aktiven Benutzers ein	7 wechselt in den Bereich "Datenexports" zur Konfiguration des Datenexports. Diese Schaltfläche ist nur sichtbar, wenn zumindest die Lizenz für eine Exportvariante vorhanden ist.
4 Schaltflächen zum Wechseln zwischen den einzelnen Serverbereichen	8 öffnet die Eingabemaske zum Upload einer XML-Datei. Diese Schaltfläche ist nur sichtbar, wenn die Lizenz für den XML-Import vorhanden ist.

10.1.1 Erklärung der Symbole



Fügt zur aktuellen Liste (Auswertungen, Messstelle, Benutzer, ...) einen neuen Eintrag hinzu



Löscht das nebenstehende Element (Auswertung, Messstelle, Benutzer, ...) aus der Liste



Ruft die Eingabemaske zum Editieren des nebenstehenden Elements (Auswertung, Messstelle, Benutzer, ...) auf

10.2 Bereich "Kunden"

HOME	KUNDEN	POOL & ALOHA	APP CENTER	BENUTZER	ALARME	STATISTIK	SERVICE	Konfiguration
1	MESS / APPLIKATIONEN	MESSGERÄTE	VORLAGEN	SCRIPTS	API Tracking Datenexport			



2 Kunden 3 10 11

2015 Austr Training 8

Suche... Q

Seiten: 1 (Gesamt 2)

! Training
Kommentar 4 5 6 7

1234 Ort Musterstrasse 1 9

Übersicht des Bereichs "Kunden"

1 Bereich, in dem eine Bilddatei als "Karte" und/oder die OpenStreetMaps Karte eingebildet werden kann

Auf der als "Karte" verwendeten Bilddatei lassen sich die Messstellen manuell platzieren.

In der OpenStreetMaps Karte werden die Messstellen erst angezeigt, wenn der Messstelle GPS-Koordinaten zugewiesen wurden.

2 fügt einen neuen Kunden hinzu

<p>3 Liste der Tags, die mindestens einem der in der Kundenliste angezeigten Kunden zugewiesen sind. Wurde die Kundenliste mittels Suchfeld oder Auswahl eines Tags beschränkt, wird dies bei der Erstellung der Liste der Tags berücksichtigt. Sobald die Kundenliste durch Auswahl eines Tags eingeschränkt wurde, erscheint am Ende der Liste der Tags ein Kreuz. Durch Klicken auf dieses Kreuz wird die Auswahl aller Tags zurückgesetzt und die Einschränkung aufgehoben.</p> <p>Durch Klicken mit der linken Maustaste auf einen der Tags werden in der Kundenliste nur mehr jene Kunden angezeigt, denen der entsprechende Tag zugewiesen ist und der gewählte Tag ist farblich hinterlegt.</p> <p>Durch Klicken mit der rechten Maustaste auf einen der Tags werden alle Kunden, denen der entsprechende Tag zugewiesen ist, ausgeblendet, der gewählte Tag ist farblich hinterlegt und die Bezeichnung des Tags durchgestrichen.</p> <p>Erneutes Klicken mit derselben Maustaste hebt die Einschränkung wieder auf.</p>
<p>4 öffnet die Eingabemaske zur Konfiguration des Kunden</p>
<p>5 löscht den Kunden</p>
<p>6 Kommentar, der in der Konfiguration des Kunden eingegeben werden kann</p>
<p>7 Wurde eine Standardauswertung definiert, gelangen Sie durch Klicken auf den Namen des Kunden zur Standardauswertung. Andernfalls wird durch Klicken auf den Namen des Kunden der Bereich "Messstellen" auf Kundenebene geöffnet (siehe "Bereich "Messstellen" auf Kundenebene" auf Seite 142 bzw. "Auswertungen" auf Seite 143).</p>
<p>8 Suchfeld zum Filtern der Kundenliste</p>
<p>9 Adresse des Kunden, die über die Eingabemaske zur Konfiguration des Kunden eingegeben werden kann</p>
<p>10 Symbol, über das sich eine OpenStreetMaps Karte laden lässt, auf der die Messstellen dargestellt werden. (siehe "Kartendarstellung" auf Seite 143)</p>
<p>11 Symbol, über das sich eine Bilddatei als "Übersichts-Karte" auf den Server laden lässt</p> <p>Um die "Karte" wieder zu entfernen, öffnen Sie den Upload-Dialog erneut und klicken Sie auf "senden" ohne zuvor eine Bilddatei auszuwählen.</p>

10.3 Bereich "Messstellen" auf Kundenebene

SITES / APPLIKATIONEN
MESSGERÄTE & ALOHA
BENUTZER
ALARME
STATISTIK
SERVICE

SITES / APPLIKATIONEN TAGS
MESSGERÄTE TAGS
API | Datenexport

+ Auswertungen

Seiten: 1 (Gesamt 1)

Auswertung 1

Auswertung 1

Kanal 1
Messstelle 1
-0,3

Kanal 2
Messstelle 1
-0,3

Int. Temp
Messstelle 1
23,6 °C

+ Sites / Applikationen

VERBINDUNG APP.

Filter: aus + aus
Sortierung: Name
Seitenlänge: 12

Austria

Messste

Messstelle 1
Device Name: 047394065Dxxxxxx (9.9.2020 - 9.9.2020)

Messstelle 2
Device Name: 04F027065Cxxxxxx (9.9.2020 - 9.9.2020)

● 9.9.2020 16:45:11
SER UTC+02:00

● 9.9.2020 16:43:54
SER UTC+02:00

Übersicht des Bereichs "Messstellen" auf Kundenebene

- 1** Bereich, in dem eine Bilddatei als "Karte" und/oder die OpenStreetMaps Karte eingeblendet werden kann

Auf der als "Karte" verwendeten Bilddatei lassen sich die Messstellen manuell platzieren.

In der OpenStreetMaps Karte werden die Messstellen erst angezeigt, wenn der Messstelle GPS-Koordinaten zugewiesen wurden.
- 2** Liste der Auswertungen (siehe "Auswertungen" auf Seite 143)
- 3** Liste der Sites / Applikationen (siehe "Site" auf Seite 75)

4 Symbol, das eine Messstelle auf der "Karte" repräsentiert
5 Symbol, über das sich eine OpenStreetMaps Karte laden lässt, auf der die Messstellen dargestellt werden. (siehe "Kartendarstellung" auf Seite 143)
6 Symbol, über das sich eine Bilddatei als "Karte" auf den Server laden lässt Um die "Karte" wieder zu entfernen, öffnen Sie den Upload-Dialog erneut und klicken Sie auf "senden" ohne zuvor eine Bilddatei auszuwählen.

10.3.1 Auswertungen

Die Auswertungen bieten eine Vielzahl an Möglichkeiten zur grafischen Darstellung der Daten auf der Web-Oberfläche des myDatanet-Server bzw. dem Download der Daten vom myDatanet-Servers. Eine detailliertere Anleitung zum Erstellen und dem Umgang mit den Auswertungen finden Sie im Benutzerhandbuch für myDatanet-Server (206.886).

10.3.2 Kartendarstellung

Die Kartendarstellung dient dazu, einen Überblick über die geografische Position der Messstellen zu geben. Eine detailliertere Anleitung zur Bedienung und Konfiguration der Kartendarstellung finden Sie im Benutzerhandbuch für myDatanet-Server (206.886).

10.4 Empfohlene Vorgehensweise

10.4.1 Anlegen der Messstelle

Hinweis: Abhängig vom jeweiligen Benutzerlevel sind einige der in den folgenden Kapiteln erwähnten Felder unter Umständen ausgeblendet. Wenden Sie sich in diesem Fall an den Administrator des myDatanet-Servers.

Eine detailliertere Anleitung zum Anlegen einer neuen Messstelle finden Sie im Benutzerhandbuch für myDatanet-Server (206.886).

1. Loggen Sie sich über das Web-Interface am myDatanet-Server ein. Die Web-Adresse erhalten Sie von Ihrem zuständigen Vertriebspartner.



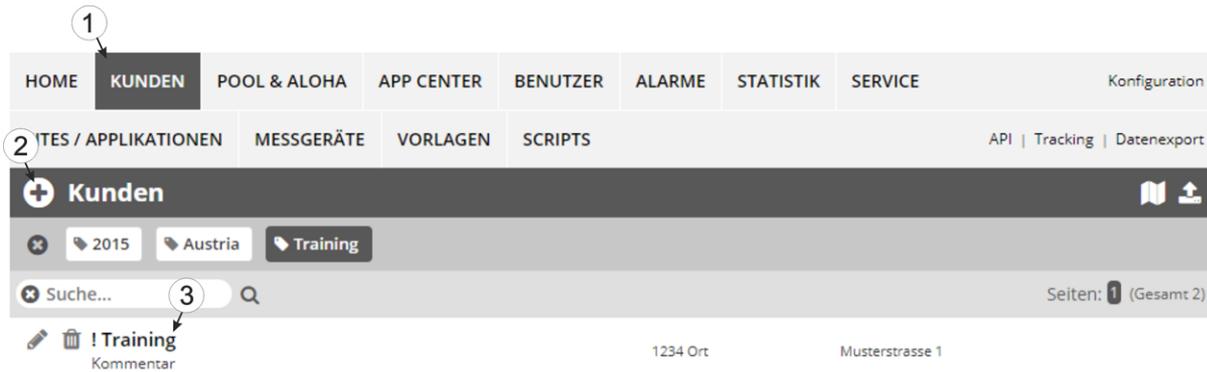
Benutzername

Kennwort

ANMELDEN

Login Formular des myDatanet-Servers

2. Klicken Sie auf den Menüpunkt "Kunde" des myDatanet-Servers um die Liste der verfügbaren Kunden aufzurufen. Wählen Sie einen bestehenden Kunden aus oder legen Sie einen neuen Kunden an.

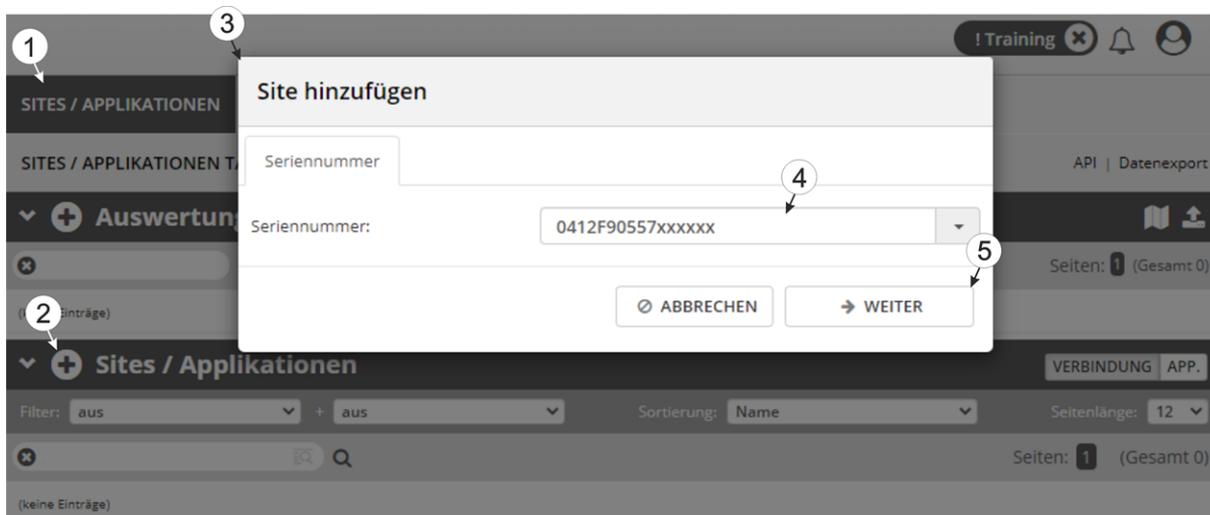


Auswählen des Kunden

1 Menüpunkt zum Aufrufen der Kundenliste	3 Liste der verfügbaren Kunden
2 Anlegen eines neuen Kunden	

3. Klicken Sie auf den Menüpunkt "Sites / Applikationen" des myDatanet-Servers, um die Liste der bestehenden Sites / Applikationen aufzurufen. Öffnen Sie das Eingabefenster zum Anlegen einer neuen Site durch Klicken auf das Symbol "Neue Site / Applikation hinzufügen", geben Sie die Seriennummer Ihres Geräts in das entsprechende Feld ein und klicken Sie anschließend auf den "Weiter" Button.

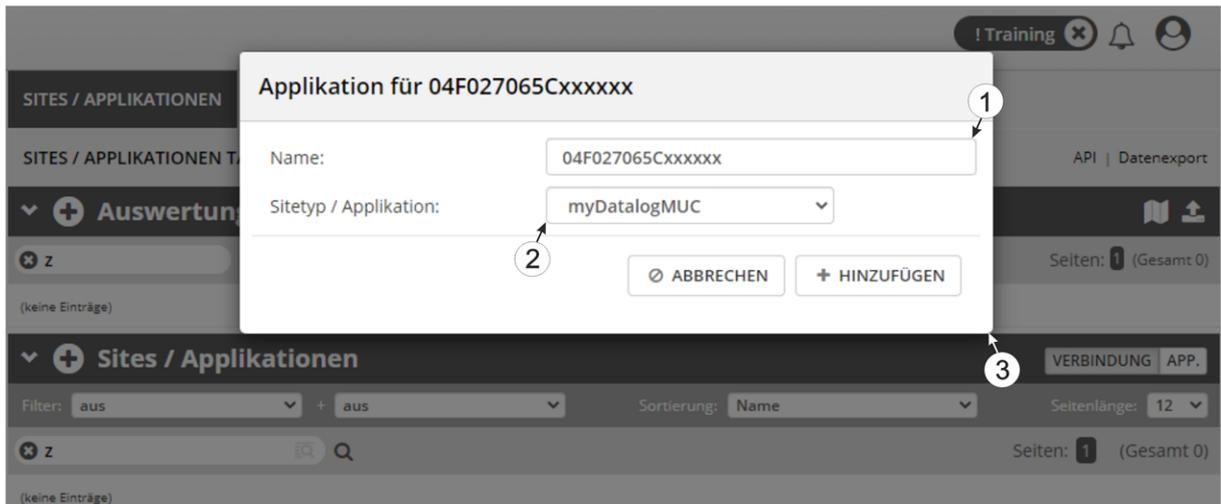
Hinweis: Die Seriennummer finden Sie auf dem Typenschild des Geräts (siehe "Gerätezeichnung" auf Seite 25)



Anlegen der Messstelle

1 Menüpunkt zum Aufrufen der Liste der bestehenden Sites / Applikationen	4 Feld zur Eingabe der Seriennummer
2 Symbol "Neue Site / Applikation hinzufügen"	5 Button "Weiter"
3 Eingabefenster für das Anlegen einer neuen Site	

4. Ändern Sie, falls erforderlich, den vorgeschlagenen Namen der Site, wählen Sie den gewünschten Sitetyp bzw. die gewünschte Applikation aus der Dropdown-Liste aus und Klicken Sie anschließend auf den "Hinzufügen" Button.



Anlegen der Messstelle abschließen

1 Name der Site (frei wählbare)	3 Button "Hinzufügen"
2 Dropdown-Liste der verfügbaren Applikationen, Vorlagen und Site Typen	

Kapitel 11 Device Logic (Pawn)

11.1 Allgemein

Das folgende Kapitel beschreibt die Funktionalität der Pawn Device Logic. PAWN (vormals SMALL) ist eine C-ähnliche Skriptsprache, welche auf embedded Systemen läuft.

Zusätzliche detaillierte Informationen finden Sie auf der Website der Entwickler:

<http://www.compuphase.com/pawn/pawn.htm>

Es gibt zwei Möglichkeiten, um ein Pawn Device Logic für das myDatalogMUC xG/4G zu erstellen:

- Direkte Eingabe in das Eingabefenster „Device Logic“ im Konfigurationsabschnitt „Steuerung“
- Verwenden eines Device Logic-Templates, das am myDatanet-Server angelegt wurde

11.1.1 Direkte Eingabe einer Pawn Device Logic

Die Eingabe der Pawn Device Logic erfolgt über den Konfigurationsabschnitt „Steuerung“ (siehe "Steuerung" auf Seite 75) der Eingabemaske zur Konfiguration der Messstelle. Als „Device Logic Type“ muss „Pawn“ ausgewählt werden, damit das myDatalogMUC xG/4G die unter „Device Logic“ eingegebenen Befehle als Pawn Device Logic interpretiert.

11.1.2 Hochladen eines Binary-Files

Wurde über die Listenauswahl "Device Logic Quelle" im Konfigurationsabschnitt „Steuerung“ (siehe "Steuerung" auf Seite 75) der Eingabemaske zur Konfiguration der Messstelle der Eintrag "Hochladen einer kompilierten Device Logic (Pawn, *.amx)" ausgewählt, kann ein zuvor erstelltes Binary-File auf den myDatanet-Server hochgeladen werden. Dieses wird dann bei der nächsten Verbindung in das myDatalogMUC xG/4G geladen. Als „Device Logic Type“ muss auch bei dieser Methode „Pawn“ ausgewählt werden, damit das myDatalogMUC xG/4G die Befehle als Pawn Device Logic interpretiert.

Hinweis: Wird mittels Pawn Device Logic direkt auf einen Ausgang geschrieben, überschreibt der ermittelte Wert den über die Eingabemaske am myDatanet-Server eingegebenen Stellwert.

11.2 Compiler-Optionen

Komprimierung des Pawn Programmcodes

```
// Über den Parameter wird angegeben welche der Sektionen komprimiert werden
// sollen
// 0: keine Komprimierung (default)
// 1: DATA
// 2: DATA und CODE
// 3: DATA, CODE und TABLES

#pragma amxcompress <0-3>
```

11.3 Device API

11.3.1 Konstanten

Returncodes für allgemeine Zwecke

OK = 0
ERROR = -1

11.3.2 System

main();

Diese Funktion wird beim PowerOn und beim Austausch der Device Logic ausgeführt. Sie sollte alle Initialisierungen enthalten, die nur ein einziges Mal beim Programmstart durchgeführt werden müssen.

forward public Mdn_CtrlFinish();

Diese Funktion ist der Einstiegspunkt für die Ausführung der Device Logic und wird zu jedem Messzeitpunkt, nachdem alle Messwerte erzeugt wurden und bevor die Ausgänge gesetzt wurden, aufgerufen. Sie sollte alle Berechnungen und Funktionen enthalten, die zyklisch durchgeführt werden sollen.

11.3.3 Datum & Zeit

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

Wurde kein Timestamp übergeben (timestamp=0), wird die aktuelle Systemzeit (in Local Time) in Stunden / Minuten / Sekunden konvertiert. Andernfalls wird der übergebene Timestamp in Stunden / Minuten / Sekunden konvertiert.

Parameter	Erklärung
<i>hour</i>	<i>Variable zur Aufnahme der Stunden - OPTIONAL</i>
<i>minute</i>	<i>Variable zur Aufnahme der Minuten - OPTIONAL</i>
<i>second</i>	<i>Variable zur Aufnahme der Sekunden - OPTIONAL</i>
<i>timestamp</i>	<i>Zeitstempel, der konvertiert werden soll</i> <i>= 0: Es wird die aktuelle Systemzeit (in Local Time) konvertiert.</i> <i>> 0: Es wird der übergebene Zeitstempel konvertiert.</i> <i>(Der Zeitstempel muss in Sekunden seit 31.12.1999 angegeben werden.)</i>

	Erklärung
<i>Rückgabewert</i>	<ul style="list-style-type: none"><i>timestamp = 0: Sekunden seit 31.12.1999 (aktuelle Systemzeit in Local Time)</i><i>timestamp > 0: Der übergebene Zeitstempel wird zurückgegeben.</i>

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

Wurde kein Timestamp übergeben (timestamp=0), wird für die aktuelle Systemzeit (in Local Time) das Datum (Jahr, Monat, Tag) ermittelt. Andernfalls wird für den übergebenen Timestamp das Datum (Jahr, Monat, Tag) ermittelt.

Parameter	Erklärung
year	Variable zur Aufnahme des Jahres - OPTIONAL Hinweis: Die Angabe des Jahres erfolgt relativ zum Jahr 2000, d.h. für das Jahr 2014 wird der Wert 14 retourniert.
month	Variable zur Aufnahme des Monats - OPTIONAL
day	Variable zur Aufnahme des Tages - OPTIONAL
timestamp	Zeitstempel für den das Datum ermittelt werden soll = 0: Es wird das Datum für die aktuelle Systemzeit (in Local Time) ermittelt. > 0: Es wird das Datum für den übergebene Zeitstempel ermittelt. (Der Zeitstempel muss in Sekunden seit 31.12.1999 angegeben werden.)

	Erklärung
Rückgabewert	<ul style="list-style-type: none"> timestamp = 0: Sekunden seit 31.12.1999 (aktuelle Systemzeit in Local Time) timestamp > 0: Der übergebene Zeitstempel wird zurückgegeben.

native Mdn_GetTimezoneOffset();

liefert die Differenz (in Sekunden) zwischen UTC und der für die Messstelle am myDatanet-Server konfigurierten lokalen Zeit. Dadurch kann im Skript die UTC bestimmt werden, indem diese Differenz von der Systemzeit (Local Time) subtrahiert wird. Der Offsetwert wird vom myDatanet-Server entsprechend der eingestellten Zeitzone (inkl. Sommer-/Winterzeit) gebildet und bei jeder Verbindung mit dem Gerät synchronisiert.

Bsp.: Für die Messstelle wird die mitteleuropäische Zeit (MEZ = UTC+1) verwendet -> Offset = 3600sec.

	Erklärung
Rückgabewert	Offsetwert in Sekunden

native Mdn_DoW(timestamp);

berechnet den Wochentag aus einem gegebenen Timestamp

Parameter	Erklärung
timestamp	Zeitstempel des zu berechnenden Tages

	Erklärung
Rückgabewert	Wochentag, 0=Montag ... 6=Sonntag

11.3.4 Encoding

`native Mdn_SetPacked(data{}, pos, &{Float,Fixed,_}:value, size=4, bool:bigendian=false);`
schreibt den übergebenen Wert an die angegebene Position in ein Array

<i>Parameter</i>	<i>Erklärung</i>
<i>data</i>	<i>Array, das die Daten aufnehmen soll</i>
<i>pos</i>	<i>Byteoffset innerhalb des Arrays zur Bestimmung der Position, an die der Wert geschrieben werden soll</i>
<i>value</i>	<i>Wert, der in das Array geschrieben werden soll</i>
<i>size</i>	<i>Anzahl der Bytes, die für den zu schreibenden Wert verwendet werden sollen</i>
<i>bigendian</i>	<i>Einstellung, für die zu verwendende Byte-Reihenfolge beim Schreiben des Werts:</i> <i>true: "Big Endian" wird verwendet</i> <i>false: "Little Endian" wird verwendet</i>

	<i>Erklärung</i>
<i>Rückgabewert</i>	<ul style="list-style-type: none"> • OK, wenn erfolgreich • ERROR

Hinweis: Ergänzende Erklärung zur Byte-Reihenfolge:

Im folgenden Beispiel wird die Ganzzahl 439.041.101 als 32-Bit-Integer-Wert ab Speicheradresse 10000 gespeichert.

<i>Adressen</i>	<i>Big Endian</i>			<i>Little Endian</i>		
	<i>Hex</i>	<i>Dez</i>	<i>Binär</i>	<i>Hex</i>	<i>Dez</i>	<i>Binär</i>
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);

schreibt den übergebenen Datenblock an die angegebene Position in ein Array

Parameter	Erklärung
<i>data</i>	<i>Array, das die Daten aufnehmen soll</i>
<i>pos</i>	<i>Byteoffset innerhalb des Arrays zur Bestimmung der Position, an die der Datenblock geschrieben werden soll</i>
<i>block</i>	<i>Datenblock, der in das Array geschrieben werden soll</i>
<i>size</i>	<i>Anzahl der Bytes, die vom Datenblock in das Array geschrieben werden sollen</i>

	Erklärung
<i>Rückgabewert</i>	<ul style="list-style-type: none"> • OK, wenn erfolgreich • ERROR

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

liefert den Wert, der sich an der angegebenen Position in einem Array befindet

Parameter	Erklärung
<i>data</i>	<i>Array, das die Daten enthält</i>
<i>pos</i>	<i>Byteoffset innerhalb des Arrays zur Bestimmung der Position, von der die Daten gelesen werden sollen</i>
<i>value</i>	<i>Variable zur Aufnahme der zu lesenden Daten</i>
<i>size</i>	<i>Anzahl der Bytes, die zu lesen sind</i>
<i>bigendian</i>	<i>Gibt an, wie die gepackten Daten zu interpretieren sind:</i> <i>true: Die Daten sind im "Big Endian"-Format im Array gespeichert.</i> <i>false: Die Daten sind im "Little Endian"-Format im Array gespeichert.</i>

	Erklärung
<i>Rückgabewert</i>	<ul style="list-style-type: none"> • OK, wenn erfolgreich • ERROR

Hinweis: Ergänzende Erklärung zur Byte-Reihenfolge:

Im folgenden Beispiel wird die Ganzzahl 439.041.101 als 32-Bit-Integer-Wert ab Speicheradresse 10000 gespeichert.

Adressen	Big Endian			Little Endian		
	Hex	Dez	Binär	Hex	Dez	Binär
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);

liest einen Datenblock, der sich an der angegebenen Position in einem Array befindet

Parameter	Erklärung
<i>data</i>	<i>Array, das die Daten enthält</i>
<i>pos</i>	<i>Byteoffset innerhalb des Arrays zur Bestimmung der Position, von der die Daten gelesen werden sollen</i>
<i>block</i>	<i>Array zur Aufnahme der zu lesenden Daten</i>
<i>size</i>	<i>Anzahl der Bytes, die zu lesen sind</i>

	Erklärung
<i>Rückgabewert</i>	<ul style="list-style-type: none">• OK, wenn erfolgreich• ERROR

11.3.5 Messkanäle

11.3.5.1 Konstanten

Nummern der Kanäle

```
MDN_CH_UI1           // Universaleingang 1
:
MDN_CH_UI8           // Universaleingang 8
MDN_CH_EXTTEMP      // ext. Temperatursensor
MDN_CH_GSM          // Interner Messkanal "GSM Stärke"
MDN_CH_VIN          // Interner Messkanal "Spannung"
MDN_CH_IOUT1        // Analogausgang 1
MDN_CH_IOUT2        // Analogausgang 2
MDN_CH_REL1         // Relais 1
:
MDN_CH_REL6         // Relais 6
MDN_CH_IN1          // Interface Kanal 1
:
MDN_CH_IN64         // Interface Kanal 64
MDN_CH_OUT1         // Interface Ausgabekanal 1
:
MDN_CH_OUT64        // Interface Ausgabekanal 64

//Erweiterungsmodul 1
MDN_CH_MUCE1_UI1    // Universaleingang 1
:
MDN_CH_MUCE1_UI8    // Universaleingang 8
MDN_CH_MUCE1_EXTTEMP // ext. Temperatursensor
MDN_CH_MUCE1_IOUT1  // Analogausgang 1
MDN_CH_MUCE1_IOUT2  // Analogausgang 2
MDN_CH_MUCE1_REL1   // Relais 1
:
MDN_CH_MUCE1_REL6   // Relais 6
```

```

//Erweiterungsmodul 2
MDN_CH_MUCE2_UI1      // Universaleingang 1
:
MDN_CH_MUCE2_UI8      // Universaleingang 8
MDN_CH_MUCE2_EXTTEMP // ext. Temperatursensor
MDN_CH_MUCE2_IOUT1    // Analogausgang 1
MDN_CH_MUCE2_IOUT2    // Analogausgang 2
MDN_CH_MUCE2_REL1     // Relais 1
:
MDN_CH_MUCE2_REL6     // Relais 6

//Erweiterungsmodul 3
MDN_CH_MUCE3_UI1      // Universaleingang 1
:
MDN_CH_MUCE3_UI8      // Universaleingang 8
MDN_CH_MUCE3_EXTTEMP // ext. Temperatursensor
MDN_CH_MUCE3_IOUT1    // Analogausgang 1
MDN_CH_MUCE3_IOUT2    // Analogausgang 2
MDN_CH_MUCE3_REL1     // Relais 1
:
MDN_CH_MUCE3_REL6     // Relais 6

//Angaben zum Status der Kommunikation mit dem jeweiligen Erweiterungsmodul
MDN_CH_MUCE1_COM_STATE
MDN_CH_MUCE2_COM_STATE
MDN_CH_MUCE3_COM_STATE

//Anzahl der Kanäle
MDN_NUM_IN_CHANNELS   // Gesamtanzahl der Interface Kanäle
MDN_NUM_OUT_CHANNELS  // Gesamtanzahl der Interface Ausgabekanäle
MDN_NUM_CHANNELS      // Gesamtanzahl aller Kanäle (alle Eingänge und Ausgänge)

```

Status des Messwertes/Messkanals

Kodierung, um verschiedene Fehlerzustände zu signalisieren

```

MDN_STATUS_OK = 0, // Wert ist in Ordnung.
MDN_STATUS_NAN = 1, // ungültiger Messwert / undefinierter Fehler
MDN_STATUS_OF = 2, // Der Messwert liegt über der Bereichsgrenze (Overflow).
MDN_STATUS_UF = 3, // Der Messwert liegt unter der Bereichsgrenze
// (Underflow).
MDN_STATUS_OL = 4, // Kabelbruch wurde erkannt bzw. kein Sensor
// angeschlossen.
MDN_STATUS_SC = 5, // Kurzschluss wurde erkannt.

```

Kanalmodi der Universaleingänge

```
MDN_MODE_IN_NONE      = 0,          // Kanal deaktiviert
MDN_MODE_IN_DIGITAL   = 1,          // digital
MDN_MODE_IN_DCTRDAY    = 2,          // Tageszähler
MDN_MODE_IN_DCTRCONT   = 3,          // Intervallzähler
MDN_MODE_IN_DFREQ      = 4,          // Frequenz
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
```

Kanalmodi der Ausgabekanäle

```
MDN_MODE_OUT_NONE     = 100,         // Kanal deaktiviert
MDN_MODE_OUT_EXTWARMUP = 101,         // Ext. Aufwärmzeit
MDN_MODE_OUT_DIGITAL   = 102,         // Digitalausgang
MDN_MODE_OUT_DFREQ     = 103,         // Frequenzausgang
MDN_MODE_OUT_DPWM      = 104,         // PWM
MDN_MODE_OUT_DIMPULS   = 105,         // Impulsausgang
MDN_MODE_OUT_A420MA    = 106,         // 4-20mA
MDN_MODE_OUT_A020MA    = 107,         // 0-20mA
```

11.3.5.2 Funktionen

native Mdn_GetCh(ch, &Float:value, &Mdn_ValueStatus:status = MDN_STATUS_OK);
liest den aktuellen Wert und Status eines Kanals aus

Parameter	Erklärung
<i>ch</i>	Nummer des Kanals (siehe "Nummern der Kanäle" im Kapitel "Konstanten" auf Seite 152)
<i>value</i>	Wert des Kanals
<i>status</i>	Status des Kanals (siehe "Status des Messwertes/Messkanals" im Kapitel "Konstanten" auf Seite 152) - OPTIONAL

	Erklärung
Rückgabewert	<ul style="list-style-type: none">• OK, wenn erfolgreich• ERROR, wenn keine gültige Kanalnummer übergeben wurde

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

liest die Parameter der Skalierung des Kanals und speichert diese in die übergebenen Variablen

Parameter	Erklärung		
<i>ch</i>	<i>Nummer des Kanals (siehe "Nummern der Kanäle" im Kapitel "Konstanten" auf Seite 152)</i>		
<i>min</i>			
	Kanaltyp	Modus	Erklärung
	<i>Universaleingänge</i>	<i>Digital Cnt.Day Cnt.Intervl. Freq</i>	<i>nicht relevant</i>
		<i>PWM 4-20mA 0-20mA 0-2V 0-10V</i>	<i>Start des Messbereichs in der Messeinheit</i>
	<i>ext. Temperatursensor</i>	<i>---</i>	<i>---</i>
	<i>Interner Messkanal</i>	<i>---</i>	<i>---</i>
	<i>Analogausgänge</i>	<i>alle Modi</i>	<i>Start des Ausgabebereichs in der Messeinheit</i>
	<i>Relais</i>	<i>---</i>	<i>---</i>
	<i>Interface Kanäle</i>	<i>wenn Skalierung "ein"</i>	<i>Start des Messbereichs in der Messeinheit</i>
<i>Interface Ausgabekanäle</i>	<i>wenn Skalierung "ein"</i>	<i>Start des Ausgabebereichs in der Messeinheit</i>	

Parameter	Erklärung		
<i>max</i>			
	Kanaltyp	Modus	Erklärung
	<i>Universaleingänge</i>	<i>Digital Cnt.Day Cnt.Intervl. Freq</i>	<i>nicht relevant</i>
		<i>PWM 4-20mA 0-20mA 0-2V 0-10V</i>	<i>Ende des Messbereichs in der Messeinheit</i>
	<i>ext. Temperatursensor</i>	<i>---</i>	<i>---</i>
	<i>Interner Messkanal</i>	<i>---</i>	<i>---</i>
	<i>Analogausgänge</i>	<i>alle Modi</i>	<i>Ende des Ausgabebereichs in der Messeinheit</i>
<i>Relais</i>	<i>---</i>	<i>---</i>	
<i>Interface Kanäle</i>	<i>wenn Skalierung "ein"</i>	<i>Ende des Messbereichs in der Messeinheit</i>	
<i>Interface Ausgabekanäle</i>	<i>wenn Skalierung "ein"</i>	<i>Ende des Ausgabebereichs in der Messeinheit</i>	

Parameter	Erklärung		
<i>scale</i>	Kanaltyp	Modus	Erklärung
	<i>Universaleingänge</i>	<i>Digital</i>	1: Invertieren "aus" -1: Invertieren "ein"
		<i>Cnt.Day Cnt.Intervl.</i>	Zählwert eines Impulses in der Messeinheit
		<i>Freq</i>	Faktor, mit dem das Eingangssignal multipliziert wird
		<i>PWM 4-20mA 0-20mA 0-2V 0-10V</i>	nicht relevant
	<i>ext. Temperatursensor</i>	---	---
	<i>Interner Messkanal</i>	---	---
	<i>Analogausgänge</i>	<i>alle Modi</i>	nicht relevant
	<i>Relais</i>	---	---
	<i>Interface Kanäle</i>	---	nicht relevant
<i>Interface Ausgabekanäle</i>	---	nicht relevant	

	Erklärung
<i>Rückgabewert</i>	<ul style="list-style-type: none"> • OK, wenn erfolgreich • ERROR, wenn keine gültige Kanalnummer übergeben wurde

native Mdn_GetChMode(ch);

liefert den eingestellten Modus des Kanals

Parameter	Erklärung
<i>ch</i>	Nummer des Kanals (siehe "Nummern der Kanäle" im Kapitel "Konstanten" auf Seite 152)

	Erklärung
<i>Rückgabewert</i>	<ul style="list-style-type: none"> • eingestellter Kanalmodus (siehe "Kanalmodi der Universaleingänge" bzw. "Kanalmodi der Ausgabekanäle" im Kapitel "Konstanten" auf Seite 152) • ERROR, wenn keine gültige Kanalnummer übergeben wurde

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

setzt den Wert und den Status eines Kanals. Mit dieser Funktion ist es auch möglich den Wert eines Eingangs zu verändern, bevor er gespeichert wird. Dies ist möglich, da die Abarbeitung des Scripts nach der Messwerterfassung und vor der Aufzeichnung erfolgt (siehe "Interne Verarbeitung der Messwerte" auf Seite 31).

Parameter	Erklärung
<i>ch</i>	Nummer des Kanals (siehe "Nummern der Kanäle" im Kapitel "Konstanten" auf Seite 152)
<i>value</i>	Wert des Kanals
<i>status</i>	Status des Kanals (siehe "Status des Messwertes/Messkanals" im Kapitel "Konstanten" auf Seite 152) - OPTIONAL

	Erklärung
Rückgabewert	<ul style="list-style-type: none">• OK, wenn erfolgreich• ERROR, wenn keine gültige Kanalnummer übergeben wurde

11.3.6 Serielle Schnittstellen

11.3.6.1 Konstanten

Nummern der Interfaces

```
MUC_ITF_COM1 = 1,      // COM1 (RS485)
MUC_ITF_COM2,         // COM2 (RS232)
MUC_ITF_COM3,         // COM3 (RS485)
```

Serial Events

Events, die durch die Funktion Mdn_SerialEvent() behandelt werden müssen

```
MDN_SERIAL_EVENT_INIT      = 0,      // Script soll die Init Sequenz senden
                               // (z.B. nach Einschalten des Sensors)
MDN_SERIAL_EVENT_MEASURE = 1,      // Script soll das Messkommando senden
MDN_SERIAL_EVENT_TIMEOUT = 2,      // Timeout beim Empfang der Antwort auf das
                               // Messkommando
```

11.3.6.2 Callback Funktionen

forward public Mdn_SerialEvent(com, event);

Diese Funktion ist der Einstiegspunkt für die Auswertung der seriellen Events. Falls das Script Parsing verwendet werden soll, muss diese Funktion vom User bereitgestellt werden.

Parameter	Erklärung
<i>com</i>	Nummer des Interface (siehe "Nummer der Interfaces" im Kapitel "Konstanten" auf Seite 158)
<i>event</i>	Event, das zum Aufruf dieser Funktion geführt hat <ul style="list-style-type: none"> • MDN_SERIAL_EVENT_INIT (Script soll die Init Sequenz senden) • MDN_SERIAL_EVENT_MEASURE (Script soll das Messkommando senden) • MDN_SERIAL_EVENT_TIMEOUT (Timeout beim Empfang der Antwort auf das Messkommando)

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

Diese Funktion ist der Einstiegspunkt für die Auswertung der empfangenen Zeichen. Falls das Script Parsing verwendet werden soll, muss diese Funktion vom User bereitgestellt werden.

Parameter	Erklärung
<i>com</i>	Nummer des Interface (siehe "Nummer der Interfaces" im Kapitel "Konstanten" auf Seite 158)
<i>data</i>	Array das die empfangenen Zeichen enthält
<i>len</i>	Anzahl der Zeichen, die empfangen wurden (max. 256)

11.3.6.3 Funktionen

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

versendet den übergebenen Datenblock über das angegebene Interface

Parameter	Erklärung
<i>com</i>	Nummer des Interface (siehe "Nummer der Interfaces" im Kapitel "Konstanten" auf Seite 158)
<i>data</i>	Array, das die zu sendenden Daten enthält
<i>len</i>	Anzahl der Bytes, die versendet werden sollen

	Erklärung
Rückgabewert	<ul style="list-style-type: none"> • OK, wenn erfolgreich • ERROR

native Mdn_SerialFinish(com);

beendet den Datenempfang über das angegebene Interface. Diese Funktion muss immer aufgerufen werden nachdem alle benötigten Daten empfangen wurden oder das Timeout Event aufgetreten ist.

Parameter	Erklärung
com	Nummer des Interface (siehe "Nummer der Interfaces" im Kapitel "Konstanten" auf Seite 158)

	Erklärung
Rückgabewert	<ul style="list-style-type: none">• OK, wenn erfolgreich• ERROR

11.3.7 Alarm & Trigger

11.3.7.1 Arrays mit symbolischen Indizes

Mdn_AlarmCfg

Alarmkonfiguration eines Messkanals

```
//Relevant für alle Kanalmodi bis auf "Digital"
// WarnValueLow   Wenn der Messwert auf oder unter diesen Wert fällt,
//                wird eine Warnung ausgelöst.
// WarnValueHigh  Wenn der Messwert auf oder über diesen Wert steigt,
//                wird eine Warnung ausgelöst.
// AlarmValueLow  Wenn der Messwert auf oder unter diesen Wert fällt,
//                wird ein Alarm ausgelöst.
// AlarmValueHigh Wenn der Messwert auf oder über diesen Wert steigt,
//                wird ein Alarm ausgelöst.
// Hyst           Hysterese für Entwarnung bei Alarm/Warnung

//Relevant für den Kanalmodus "Digital"
// Flags         MDN_FLG_WARNING:  "High" löst eine "Warnung" aus.
//                MDN_FLG_ALARM:    "High" löst einen "Alarm" aus.
//                MDN_FLG_FAULTLOW: "High" löst eine "Störung Warnung" aus.
//                MDN_FLG_FAULTHIGH: "High" löst einen "Störung Alarm" aus.

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

Mdn_TriggerCfg*Trigger-Konfiguration eines Messkanals*

```
// Mode      Trigger-Modus (siehe "Trigger-Modus" im Kapitel
//           "Konstanten" auf Seite 161)
// Flags     gibt an, welche Aktionen ausgelöst werden sollen, wenn der Trigger
//           aktiv ist (siehe "Trigger Flags" im Kapitel
//           "Konstanten" auf Seite 161)
// Value     Schwellen für das Auslösen des Tiggers

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

11.3.7.2 Konstanten**Alarm Flags**

Flags zum Spezifizieren des zu setzenden Alarmtyps für die Funktion Mdn_SetAlarm() bzw. zur Ermittlung des durch die Funktion Mdn_GetAlarmCfg() zurückgelieferten Alarmtyps.

```
MDN_FLG_WARNING    = 0b000000001, // Warnung
MDN_FLG_ALARM      = 0b000000010, // Alarm
MDN_FLG_FAULTLOW   = 0b000000100, // technischer Fehler niedriger Priorität
MDN_FLG_FAULTHIGH  = 0b000001000, // technischer Fehler hoher Priorität
MDN_FLG_UNDERFLOW  = 0b100000000, // Bit gesetzt: Alarm/Warnung, da
//                               Wert <= Schwelle
//                               // Bit gelöscht: Alarm/Warnung, da
//                               //                               Wert >= Schwelle
```

Trigger Flags

Trigger-Flags für die Funktion Mdn_SetTrigger(), zur Auswertung des Rückgabewerts der Funktion Mdn_GetTrigger() und zur Auswertung der durch die Funktion Mdn_GetTriggerCfg() gelesenen Trigger-Konfiguration eines Kanals

```
MDN_TRG_INTERN1    = 0b00000000000010000, // gerätespezifisch
MDN_TRG_INTERN2    = 0b00000000000100000, // gerätespezifisch
MDN_TRG_INTERN3    = 0b00000000001000000, // gerätespezifisch
MDN_TRG_INTERN4    = 0b00000000010000000, // gerätespezifisch
MDN_TRG_RECQUICK   = 0b0000000100000000, // Schnelle Aufzeichnung anfordern.
MDN_TRG_RECSLOW    = 0b0000000100000000, // Langsame Aufzeichnung anfordern.
MDN_TRG_MEASSTART  = 0b0000001000000000, // Startet eine neue Messung
MDN_TRG_RECORD_ON  = 0b0000010000000000, // Starte Aufzeichnung der aktuellen
//                               // Messung
MDN_TRG_RECORD_OFF = 0b0001000000000000, // Verhindere Aufzeichnung der
//                               // aktuellen Messung
MDN_TRG_TXSTART    = 0b0010000000000000, // Fordere eine einzelne Verbindung mit
//                               // dem Server an
MDN_TRG_CONTINUOUS = 0b1000000000000000, // Fordere eine Online Verbindung mit
//                               // dem Server an
```

Trigger-Modus

zur Auswertung der durch die Funktion `Mdn_GetTriggerCfg()` gelesenen Trigger-Konfiguration eines Kanals

```
MDN_TRG_MODE_NONE           = 0, // Kanal deaktiviert, kein Trigger

//Relevant für alle Kanalmodi bis auf "Digital"
MDN_TRG_MODE_LESS_OR_EQUAL  = 1, // Auslösen des Triggers, wenn
                                // Wert <= Schwelle
MDN_TRG_MODE_GREATER_OR_EQUAL = 2, // Auslösen des Triggers, wenn
                                // Wert >= Schwelle

//Relevant für den Kanalmodus "Digital"
MDN_TRG_MODE_DI_RISING_EDGE  = 3, // Steigende Flanke löst den Trigger aus.
MDN_TRG_MODE_DI_FALLING_EDGE = 4, // Fallende Flanke löst den Trigger aus.
MDN_TRG_MODE_DI_BOTH_EDGES   = 8, // Beide Flanken lösen den Trigger aus.
```

11.3.7.3 Funktionen

native `Mdn_GetAlarm(ch)`;

gibt den Alarmstatus eines Kanals zurück

Parameter	Erklärung
<code>ch</code>	Nummer des Kanals (siehe "Konstanten" auf Seite 152)

	Erklärung
Rückgabewert	positiv, wenn erfolgreich (siehe "Alarm Flags" im Kapitel "Konstanten" auf Seite 161), <code>ERROR</code> , falls kein gültiger Kanal angegeben wurde

native `Mdn_GetAlarmCfg(ch, Config[Mdn_AlarmCfg])`;

liefert die Alarmkonfiguration eines Kanals

Parameter	Erklärung
<code>ch</code>	Nummer des Kanals (siehe "Konstanten" auf Seite 152)
<code>Config</code>	Struktur zur Aufnahme der Alarmkonfiguration (siehe " <code>Mdn_AlarmCfg</code> " im Kapitel "Arrays mit symbolischen Indizes" auf Seite 160)

	Erklärung
Rückgabewert	<ul style="list-style-type: none">• <code>OK</code>, wenn erfolgreich• <code>ERROR</code>, wenn keine gültige Kanalnummer übergeben wurde

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

setzt den Alarmstatus eines Kanals. Um anzuzeigen, dass es sich um einen Alarm/Warnung aufgrund einer Unterschreitung des Alarm/Warnungs Levels handelt, muss das Alarmflag "MDN_FLG_UNDERFLOW" gesetzt werden. Für die Anzeige von Alarm/Warnung aufgrund einer Überschreitung des Alarm/Warnungs Levels bleibt das Alarmflag "MDN_FLG_UNDERFLOW" gelöscht. Die Alarmerkennung des Systems erfolgt vor der Abarbeitung des Scripts (siehe "Interne Verarbeitung der Messwerte" auf Seite 31).

Parameter	Erklärung
<i>ch</i>	Nummer des Kanals (siehe "Konstanten" auf Seite 152)
<i>alarm</i>	Alarmflags (siehe "Alarm Flags" im Kapitel "Konstanten" auf Seite 161). Der Wert "0" löscht den Alarmstatus des Kanals.
<i>value</i>	Wert des Kanals, der für die Alarmberechnung herangezogen wurde
<i>level</i>	enthält den Schwellenwert des Alarms - OPTIONAL

	Erklärung
Rückgabewert	<ul style="list-style-type: none"> • OK, wenn erfolgreich • ERROR, wenn keine gültige Kanalnummer übergeben wurde

native Mdn_SetTrigger(trigger);

setzt globale Triggerereignisse

Parameter	Erklärung
<i>trigger</i>	Trigger-Flags (siehe "Trigger Flags" im Kapitel "Konstanten" auf Seite 161)

	Erklärung
Rückgabewert	OK, wenn erfolgreich

native Mdn_GetTrigger();

liefert die globalen Triggerflags

	Erklärung
Rückgabewert	globale Triggerflags (siehe "Trigger Flags" im Kapitel "Konstanten" auf Seite 161)

native Mdn_GetTriggerCfg(ch, Config[Mdn_TriggerCfg]);

liefert die Trigger-Konfiguration eines Kanals

Parameter	Erklärung
<i>ch</i>	<i>Nummer des Kanals (siehe "Konstanten" auf Seite 152)</i>
<i>Config</i>	<i>Struktur zur Aufnahme der Trigger-Konfiguration (siehe "Mdn_TriggerCfg" im Kapitel "Arrays mit symbolischen Indizes" auf Seite 160)</i>

	Erklärung
<i>Rückgabewert</i>	<ul style="list-style-type: none"> • <i>OK, wenn erfolgreich</i> • <i>ERROR, wenn keine gültige Kanalnummer übergeben wurde</i>

11.3.8 Mathematik

Hilfreiche Konstanten

Definintition	Wert	Beschreibung
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	$\pi/2$
M_PI_4	0.78539816339744830962	$\pi/4$
M_1_PI	0.31830988618379067154	$1/\pi$
M_2_PI	0.63661977236758134308	$2/\pi$
M_2_SQRTPI	1.12837916709551257390	$2/\sqrt{\pi}$
M_SQRT2	1.41421356237309504880	$\sqrt{2}$
M_SQRT1_2	0.70710678118654752440	$1/\sqrt{2}$

native fround(Float:x);

führt kaufmännisches Runden des übergebenen Floats durch

Parameter	Erklärung
<i>x</i>	<i>Float, der gerundet werden soll</i>

	Erklärung
<i>Rückgabewert</i>	<i>kaufmännisch gerundeter ganzzahliger Wert</i>

native min(value1, value2);*liefert den kleineren der beiden übergebenen Werte*

Parameter	Erklärung
value1	zwei Werte, von denen der kleinere ermittelt werden soll
value2	

	Erklärung
Rückgabewert	der kleinere der beiden übergebenen Werte

native max(value1, value2);*liefert den größeren der beiden übergebenen Werte*

Parameter	Erklärung
value1	zwei Werte, von denen der größere ermittelt werden soll
value1	

	Erklärung
Rückgabewert	der größere der beiden übergebenen Werte

native clamp(value, min=cellmin, max=cellmax);*prüft, ob der übergebene Wert zwischen "min" und "max" liegt*

Parameter	Erklärung
value	Wert, der geprüft werden soll
min	untere Bereichsgrenze
max	obere Bereichsgrenze

	Erklärung
Rückgabewert	<ul style="list-style-type: none"> • "value", wenn der Wert zwischen "min" und "max" liegt • "min", wenn der Wert kleiner "min" ist • "max", wenn der Wert größer "max" ist

native swapchars(c);
vertauscht die Reihenfolge der Bytes

Parameter	Erklärung
<i>c</i>	<i>Wert für den die Bytes vertauscht werden sollen</i>

	Erklärung
<i>Rückgabewert</i>	<i>Wert, bei dem die Bytes im Parameter "c" vertauscht sind (das niedrigste Byte wird das höchste Byte)</i>

Die Arbeitsweise der folgenden Funktionen entspricht jener der Standard ANSI-C Implementierung:

native Float:sin(Float:x);
Sinus von x

native Float:cos(Float:x);
Kosinus von x

native Float:tan(Float:x);
Tangens von x

native Float:asin(Float:x);
arcsin(x) im Bereich $[-\pi/2, \pi/2]$, x Element von $[-1, 1]$

native Float:acos(Float:x);
arccos(x) im Bereich $[0, \pi]$, x Element von $[-1, 1]$

native Float:atan(Float:x);
arctan(x) im Bereich $[-\pi/2, \pi/2]$

native Float:atan2(Float:y, Float:x);
arctan(y/x) im Bereich $[-\pi, \pi]$

native Float:sinh(Float:x);
Sinus Hyperbolicus von x

native Float:cosh(Float:x);
Cosinus Hyperbolicus von x

native Float:tanh(Float:x);
Tangens Hyperbolicus von x

native Float:exp(Float:x);
Exponentialfunktion e^x

native Float:log(Float:x);
natürlicher Logarithmus $\ln(x)$, $x > 0$

native Float:log10(Float:x);
Logarithmus zur Basis 10 $\log_{10}(x)$, $x > 0$

native Float:pow(Float:x, Float:y);
 x^y . Ein Argumentenfehler liegt vor bei $x = 0$ und $y \leq 0$, oder bei $x < 0$ und y ist nicht ganzzahlig.

native Float:sqrt(Float:x);
Wurzel x, $x \geq 0$

native Float:ceil(Float:x);
kleinster ganzzahliger Wert, der nicht kleiner als x ist

native Float:floor(Float:x);

größter ganzzahliger Wert, der nicht größer als x ist

native Float:fabs(Float:x);

absoluter Wert | x |

native Float:ldexp(Float:x, n);

$x \cdot 2^n$

native Float:frexp(Float:x, &n);

zerlegt x in eine normalisierte Mantisse im Bereich [1/2, 1], die als Resultat geliefert wird, und eine Potenz von 2, die in n abgelegt wird. Ist x null, sind beide Teile des Resultats null.

native Float:modf(Float:x, &Float:ip);

zerlegt x in einen ganzzahligen Teil und einen Rest, die beide das gleiche Vorzeichen wie x besitzen. Der ganzzahlige Teil wird bei ip abgelegt, der Rest ist das Resultat.

native Float:fmod(Float:x, Float:y);

Gleitpunktrest von x/y, mit dem gleichen Vorzeichen wie x. Wenn y null ist, hängt das Resultat von der Implementierung ab.

native isnan(Float:x);

liefert einen Wert ungleich Null, wenn x "not a number" ist

11.3.9 Char & String

Hinweis: Um die Funktionen dieses Kapitels verwenden zu können, benötigen Sie folgendes Include-File: `#include <string>`

Die Arbeitsweise der folgenden Funktionen entspricht im Wesentlichen jener der Standard ANSI-C Implementierung:

native strlen(const string[]);

liefert die Länge von string (ohne '\0')

Parameter	Erklärung
<i>string</i>	<i>Zeichenkette, deren Länge bestimmt werden soll</i>

	Erklärung
<i>Rückgabewert</i>	<i>Anzahl der Zeichen ohne der abschließenden '\0'</i>

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

speichert den übergebenen Format-String in dem Array dest. Die Arbeitsweise der Funktionen entspricht der Funktion "sprintf" der Standard ANSI-C Implementierung

Parameter	Erklärung
<i>dest</i>	Array zur Aufnahme des formatierten Ergebnisses
<i>maxlength</i>	maximale Zeichenanzahl, die das dest Array aufnehmen kann
<i>format</i>	die zu verwendende Format-Zeichenkette

	Erklärung
<i>Rückgabewert</i>	<ul style="list-style-type: none">• -1 im Fehlerfall• Anzahl der Zeichen, die geschrieben worden wäre, wenn das Array dest lang genug gewesen wäre (ohne '\0'). <p>Das Array dest erhält in jedem Fall ein abschließendes Nullzeichen. In keinem Fall wird über die Länge des Arrays dest hinausgeschrieben.</p>

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

kopiert die Zeichenkette source in das Array dest (inklusive '\0').

Parameter	Erklärung
<i>dest</i>	Array zur Aufnahme der zu kopierenden Zeichenkette
<i>source</i>	zu kopierende Zeichenkette
<i>maxlength</i>	Größe (in Cells) des Arrays zur Aufnahme der zu kopierenden Zeichenkette - OPTIONAL

	Erklärung
<i>Rückgabewert</i>	Anzahl der kopierten Zeichen

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

fügt die Zeichenkette source an die Zeichenkette dest an (inklusive '\0')

Parameter	Erklärung
<i>dest</i>	Array zur Aufnahme des Ergebnisses. Dieses Array enthält bereits eine Zeichenkette an die die Zeichenkette source angefügt werden soll.
<i>source</i>	Zeichenkette, die an die im Array dest enthaltene Zeichenkette angefügt werden soll
<i>maxlength</i>	Größe (in Cells) des Arrays zur Aufnahme des Ergebnisses - OPTIONAL

	Erklärung
<i>Rückgabewert</i>	Anzahl der angefügten Zeichen

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

vergleicht die Zeichenketten string1 und string2

Parameter	Erklärung
<i>string1</i>	<i>die beiden Zeichenketten, die verglichen werden sollen</i>
<i>string2</i>	
<i>length</i>	<i>Die maximale Anzahl von Zeichen, die beim Vergleich berücksichtigt werden sollen - OPTIONAL</i>

	Erklärung
<i>Rückgabewert</i>	<ul style="list-style-type: none"> • 1: <i>string1</i> > <i>string 2</i> • 0: die beiden Zeichenketten sind gleich (zumindest die berücksichtigte Länge) • -1: <i>string1</i> < <i>string 2</i>

native strchr(const string[], char);

sucht ein Zeichen (erstes Vorkommen) in einer Zeichenkette

Parameter	Erklärung
<i>string</i>	<i>Zeichenkette, die durchsucht werden soll</i>
<i>char</i>	<i>Zeichen, das gesucht werden soll</i>

	Erklärung
<i>Rückgabewert</i>	<ul style="list-style-type: none"> • -1, wenn das gesuchte Zeichen nicht in der Zeichenkette enthalten ist • Array-Index des gesuchten Zeichens (erstes in der Zeichenkette vorkommendes Zeichen)

native strrchr(const string[], char);

sucht ein Zeichen (letztes vorkommen) in einer Zeichenkette

Parameter	Erklärung
<i>string</i>	<i>Zeichenkette, die durchsucht werden soll</i>
<i>char</i>	<i>Zeichen, das gesucht werden soll</i>

	Erklärung
<i>Rückgabewert</i>	<ul style="list-style-type: none"> • -1, wenn das gesuchte Zeichen nicht in der Zeichenkette enthalten ist • Array-Index des gesuchten Zeichens (letztes in der Zeichenkette vorkommendes Zeichen)

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

sucht die Position des ersten Zeichens in string1, das **nicht** in der Zeichenkette erlaubter Zeichen (string2) enthalten ist

Parameter	Erklärung
string1	Zeichenkette, die durchsucht werden soll
string2	Zeichenkette erlaubter Zeichen

	Erklärung
Rückgabewert	<ul style="list-style-type: none">• Länge von string1, wenn keine unerlaubten Zeichen gefunden wurden• Position des ersten Zeichens in der zu durchsuchenden Zeichenkette, das nicht in der Zeichenkette der erlaubten Zeichen enthalten ist

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

sucht die Position des ersten Zeichens in string1, das auch in der Zeichenkette erlaubter Zeichen (string2) enthalten ist

Parameter	Erklärung
string1	Zeichenkette, die durchsucht werden soll
string2	Zeichenkette erlaubter Zeichen

	Erklärung
Rückgabewert	<ul style="list-style-type: none">• Länge von string1, wenn kein erlaubtes Zeichen gefunden wurde• Position des ersten Zeichens in der zu durchsuchenden Zeichenkette, das auch in der Zeichenkette der erlaubten Zeichen enthalten ist

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

sucht den Array-Index des ersten Zeichens, das auch in der Zeichenkette erlaubter Zeichen enthalten ist

Parameter	Erklärung
string1	Zeichenkette, die durchsucht werden soll
string2	Zeichenkette erlaubter Zeichen

	Erklärung
Rückgabewert	<ul style="list-style-type: none">• -1, wenn das gesuchte Zeichen nicht in der Zeichenkette enthalten ist• Array-Index des ersten Zeichens in der zu durchsuchenden Zeichenkette, das auch in der Zeichenkette der erlaubten Zeichen enthalten ist

native strstr(const string1[], const string2[]);*sucht die Zeichenkette string2 in der Zeichenkette string1*

Parameter	Erklärung
<i>string1</i>	<i>Zeichenkette, die durchsucht werden soll</i>
<i>string2</i>	<i>zu suchende Zeichenkette</i>

	Erklärung
<i>Rückgabewert</i>	<ul style="list-style-type: none"> • <i>-1, wenn die zu suchende Zeichenkette string2 nicht in string1 enthalten ist</i> • <i>Array-Index an der die zu suchende Zeichenkette string2 im string1 beginnt</i>

native strtol(const string[], base);*wandelt eine Zeichenkette in einen Wert um*

Parameter	Erklärung
<i>string</i>	<i>umzuwandelnde Zeichenkette</i>
<i>base</i>	<i>gibt die Basis an, die für die Umwandlung verwendet werden soll</i> <i>2-36: Die angegebene Basis wird verwendet</i> <i>0: Als Basis wird 8, 10 oder 16 verwendet, abhängig von der umzuwandelnden Zeichenkette</i> <i>Basis 8: bei einer führenden 0</i> <i>Basis 16: bei 0x oder 0X</i>

	Erklärung
<i>Rückgabewert</i>	<i>Wert, der der Zeichenkette entspricht</i>

native Float: atof(const string[]);*wandelt eine Zeichenkette in einen Float um*

Parameter	Erklärung
<i>string</i>	<i>umzuwandelnde Zeichenkette</i>

	Erklärung
<i>Rückgabewert</i>	<i>Float, dessen Zahlenwert der Zeichenkette entspricht</i>

native tolower(c);

wandelt ein Zeichen in einen Kleinbuchstaben um

Parameter	Erklärung
<i>c</i>	<i>Zeichen, das in einen Kleinbuchstaben umgewandelt werden soll</i>

	Erklärung
<i>Rückgabewert</i>	<i>Die Kleinbuchstaben-Variante des übergebenen Zeichens, falls vorhanden, oder der unveränderte Zeichencode von "c", wenn der Buchstabe "c" kein Kleinbuchstaben-Äquivalent hat.</i>

native toupper(c);

wandelt ein Zeichen in einen Großbuchstaben um

Parameter	Erklärung
<i>c</i>	<i>Zeichen, das in einen Großbuchstaben umgewandelt werden soll</i>

	Erklärung
<i>Rückgabewert</i>	<i>Die Großbuchstaben-Variante des übergebenen Zeichens, falls vorhanden, oder der unveränderte Zeichencode von "c", wenn der Buchstabe "c" kein Großbuchstaben-Äquivalent hat.</i>

11.3.10 Verschiedene Funktionen

11.3.10.1 Arrays mit symbolischen Indizes

Mdn_TablePoint

zweispaltige Stützpunkttabelle, Datentyp Float

```
// key    Spalte, die durchsucht wird
// value  Spalte mit den zurückzuliefernden Ergebniswerten

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

11.3.10.2 Konstanten

Fehlercodes der Funktionen "Mdn_TablePoint"

```
MDN_TAB_ERR_FLOOR = -1, // gesuchter Wert kleiner als der erste Tabelleneintrag
MDN_TAB_ERR_CEIL  = -2, // gesuchter Wert größer als der letzte Tabelleneintrag
```

11.3.10.3 Funktionen

native `getapilevel()`;

gibt das implementierte API-Level der Skript-Engine aus

	Erklärung
Rückgabewert	implementiertes API-Level der Skript-Engine

native `CRC16(data{}, len)`;

liefert die berechnete Modbus CRC16 der übergebenen Daten

Parameter	Erklärung
<i>data</i>	<i>Array, das die Daten enthält für die die CRC16 berechnet werden soll</i>
<i>len</i>	<i>Anzahl der Bytes, die bei der Berechnung berücksichtigt werden sollen</i>

	Erklärung
Rückgabewert	berechnete CRC16

native `CRC32(data{}, len)`;

liefert die berechnete Ethernet CRC32 der übergebenen Daten

Parameter	Erklärung
<i>data</i>	<i>Array, das die Daten enthält für die die CRC32 berechnet werden soll</i>
<i>len</i>	<i>Anzahl der Bytes, die bei der Berechnung berücksichtigt werden sollen</i>

	Erklärung
Rückgabewert	berechnete CRC32

native `LRC(data{}, len)`;

liefert die berechnete Modbus LRC der übergebenen Daten

Parameter	Erklärung
<i>data</i>	<i>Array, das die Daten enthält, für die die LRC berechnet werden soll</i>
<i>len</i>	<i>Anzahl der Bytes, die bei der Berechnung berücksichtigt werden sollen</i>

	Erklärung
Rückgabewert	berechnete LRC

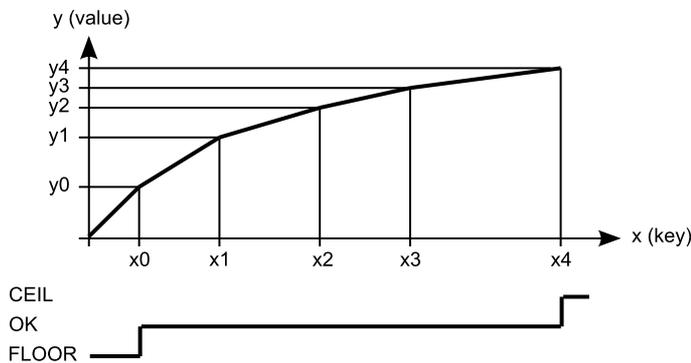
native Mdn_CalcTable(Float:key, &Float:value, const table[][Mdn_TablePoint], size = sizeof table);
*sucht einen bestimmten Wert in der "key"-Spalte der übergebenen Stützpunkttabelle und liefert den entsprechenden Wert der "value"-Spalte der Tabelle. Liegt der gesuchte Wert zwischen zwei Stützpunkten, wird der Rückgabewert zwischen den zwei angrenzenden "value"-Spaltenwerten linear interpoliert (Geradengleichung: $y = k*x + d$). Mit dieser Funktion können nicht lineare Kennlinien (z.B. Zusammenhang ADC-Wert -> Temperatur) nachgebildet werden.*

Parameter	Erklärung
<i>key</i>	<i>Wert, der für die Suche herangezogen wird</i>
<i>value</i>	<i>enthält das Ergebnis der Berechnung durch die Funktion</i>
<i>table</i>	<i>Die Tabelle, die durchsucht wird, muss vom Typ "Mdn_TablePoint" sein.</i>
<i>size</i>	<i>Anzahl der Elemente in der Tabelle</i>

	Erklärung
<i>Rückgabewert</i>	<ul style="list-style-type: none"> • OK, wenn der entsprechende Wert gefunden wurde • MDN_TAB_ERR_FLOOR, wenn der Schlüssel kleiner als der erste Tabelleneintrag ist. "value" beinhaltet den ersten Tabelleneintrag. • MDN_TAB_ERR_CEIL, wenn der Schlüssel größer als der letzte Tabelleneintrag ist. "value" beinhaltet den letzten Tabelleneintrag.

Hinweis: Ergänzende Erklärung zur Stützpunkttabelle "*table*"

Die Tabellenzeilen können in einem x/y-Koordinatensystem dargestellt werden. Die Werte der "key"-Spalte werden dabei auf der x-Achse aufgetragen, die dazugehörigen Werte der "value"-Spalte auf der y-Achse.



Darstellung der Stützpunkttabelle als x/y-Koordinatensystem

native Mdn_WriteLog(log, param);
erzeugt einen Eintrag im Geräte-log

Parameter	Erklärung
log	zu erzeugender Log-Eintrag (gültiger Bereich 0...999). Im Kapitel "Vordefinierte Log-Einträge" auf Seite 177 sind bereits einige Fehlercodes vordefiniert. Falls möglich, sollten Sie diese für die beschriebenen Fehlerbilder verwenden. Wichtiger Hinweis: Die Log-Einträge werden für die Anzeige am Server in den Bereich "MODULE ERR" (2000-2999) gemappt.
param	zusätzlicher Parameter zur genaueren Spezifizierung des Log-Eintrags (gültiger Bereich -32768 .. 32767)

	Erklärung
Rückgabewert	OK

native heapSpace();
liefert den freien Speicherplatz auf dem Heap

	Erklärung
Rückgabewert	Der freie Speicherplatz auf dem Heap. Der Stack und der Heap besetzen einen gemeinsamen Speicherbereich, so dass dieser Wert die Anzahl der Bytes angibt, die entweder für den Stack oder den Heap übrig sind.

native funcidx(const name[]);
liefert den Index einer öffentlichen Funktion

Parameter	Erklärung
name	Name der öffentlichen Funktion

	Erklärung
Rückgabewert	<ul style="list-style-type: none"> -1, wenn keine Funktion mit dem übergebenen Namen existiert Index der öffentlichen Funktion

native numargs();
liefert die Anzahl der an eine Funktion übergebenen Argumente

	Erklärung
Rückgabewert	Die Anzahl der Argumente, die an eine Funktion übergeben wurden. numargs ist nützlich innerhalb von Funktionen mit einer variablen Argumentenliste.

native getarg(arg, index=0);

liefert den Wert des Arguments

Parameter	Erklärung
<i>arg</i>	<i>Die Sequenznummer des Arguments. Verwenden Sie 0 für das erste Argument.</i>
<i>index</i>	<i>Index, falls sich "arg" auf ein Array bezieht</i>

	Erklärung
<i>Rückgabewert</i>	<i>Diese Funktion liefert ein Argument aus einer variablen Argumentenliste. Wenn das Argument ein Array ist, gibt "index" den Index des gewünschten Arrayelements an. Der Rückgabewert ist der Wert des Arguments.</i>

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

setzt den Wert des Arguments

Parameter	Erklärung
<i>arg</i>	<i>Die Sequenznummer des Arguments. Verwenden Sie 0 für das erste Argument.</i>
<i>index</i>	<i>Index, falls sich "arg" auf ein Array bezieht</i>
<i>value</i>	<i>Wert auf den das Argument gesetzt werden soll</i>

	Erklärung
<i>Rückgabewert</i>	<ul style="list-style-type: none">• <i>true</i>, wenn der Wert gesetzt werden konnte• <i>false</i>, wenn das Argument oder der Index ungültig sind <p><i>Diese Funktion setzt ein Argument in einer variablen Argumentenliste. Wenn das Argument ein Array ist, gibt "index" den Index des gewünschten Arrayelements an.</i></p>

11.3.11 Consolen Funktionen

Wichtiger Hinweis: Beim myDatalogMUC xG/4G wird als Standardausgabe die Schnittstelle Com2 (115200 8N1) benutzt, falls diese nicht anderweitig in Verwendung ist.

native print(const string[]);

druckt den angegebenen String auf die Standardausgabe

Parameter	Erklärung
<i>string</i>	<i>die auszugebende Zeichenfolge. Diese darf auch Escape-Sequenzen enthalten.</i>

	Erklärung
<i>Rückgabewert</i>	<i>OK</i>

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

druckt den übergebenen Format-String auf die Standardausgabe. Die Arbeitsweise der Funktionen entspricht jener der Standard ANSI-C Implementierung.

Parameter	Erklärung
format[]	die zu verwendende Format-Zeichenkette

	Erklärung
Rückgabewert	<ul style="list-style-type: none"> • Anzahl der gedruckten Zeichen • ERROR, falls nicht erfolgreich

Wichtiger Hinweis: Die folgende Funktion wird zwar weiterhin zugunsten der Abwärtskompatibilität unterstützt, sollte aber bei Geräten mit Modemversion 03v006 und höher nicht mehr verwendet werden.

native printi(value);

druckt den angegebenen Wert auf die Standardausgabe

Parameter	Erklärung
value	der auszugebende Wert vom Typ Integer

11.4 Vordefinierte Log-Einträge

Wichtiger Hinweis: Die Log-Einträge werden für die Anzeige am Server in den Bereich "MODULE ERR" (2000-2999) gemappt.

Log-Eintrag		Parameter		Beschreibung
Code	Klartext	Code	Klartext	
0	LOG_COM2_ERROR	0	CHAR TIMEOUT	Com2: Der zeitliche Abstand zwischen zwei empfangenen Zeichen ist größer als der erlaubte Zeichen-Timeout.
		1	FRAME TIMEOUT	Com2: Innerhalb des Datensatz-Timeouts wurde nicht die komplette Antwort empfangen.
		2	INVALID FRAME	Com2: Der Antwort-Frame des Sensors ist ungültig.
1	LOG_CHAR_TIMEOUT	##	---	Der zeitliche Abstand zwischen zwei empfangenen Zeichen ist größer als der erlaubte Zeichen-Timeout.
2	LOG_FRAME_TIMEOUT	##	---	Innerhalb des Datensatz-Timeouts wurde nicht die komplette Antwort empfangen.
3	LOG_FRAME_INVALID	##	---	Der Antwort-Frame des Sensors ist ungültig.

11.5 Device Logic Fehlercodes

Zu jedem Messzeitpunkt, nachdem alle Messwerte erzeugt wurden, wird das PAWN-Script durchlaufen. Sollte dabei ein Fehler auftreten, wird die Scriptausführung gestoppt und deaktiviert. Zudem wird in das Gerätelog der entsprechende Fehlercode eingetragen und eine Verbindung zum Server aufgebaut. Bei allen Log-Einträgen bis auf "SCRIPT_ERR" enthält der Parameter den 32-Bit Instruction Pointer der PAWN abstract machine (AMX). Da im Parameter eines Log-Eintrags nur 16-Bit Werte gespeichert werden können, werden jeweils 2 Einträge im Gerätelog erzeugt. Der erste Eintrag enthält Bit31-Bit16 und der zweite Eintrag enthält Bit15-Bit0 des 32-Bit Instruction Pointers. Eine Anleitung zum Auswerten des Gerätelogs finden Sie im Kapitel „Auswerten des Gerätelogs“ (siehe "Auswerten des Gerätelogs" auf Seite 224).

Log-Eintrag		Parameter		Beschreibung
Code	Klartext	Code	Klartext	
3000	SCRIPT_ERR	0	NO SCRIPT	keine gültige Device Logic vorhanden
		1	SCRIPT UPDATE	neue Device Logic erhalten
3001	AMX_ERR_EXIT	##	---	Abbruch z.B. Max. Anzahl der PAWN-Befehle (100.000) pro Durchlauf erreicht
3002	AMX_ERR_ASSERT	##	---	Assertion fehlgeschlagen
3003	AMX_ERR_STACKERR	##	---	Stack / Heap Kollision (unzureichende Stack-Größe)
3004	AMX_ERR_BOUNDS	##	---	Array-Index außerhalb des gültigen Bereichs
3005	AMX_ERR_MEMACCESS	##	---	ungültiger Speicherzugriff z.B. Verwechslung zwischen Cell (32Bit Element) Zugriff [] und Byte Zugriff {}
3006	AMX_ERR_INVINSTR	##	---	ungültige Anweisung
3007	AMX_ERR_STACKLOW	##	---	Stack-Unterlauf
3008	AMX_ERR_HEAPLOW	##	---	Heap Unterlauf
3009	AMX_ERR_CALLBACK	##	---	keine (ungültige) native Callback-Funktion
3010	AMX_ERR_NATIVE	##	---	Native-Funktion ist fehlgeschlagen
3011	AMX_ERR_DIVIDE	##	---	Division durch Null
3012	AMX_ERR_SLEEP	##	---	Sleep-Modus
3013	AMX_ERR_INVSTATE	##	---	ungültiger Zustand
3014	reserviert			

Log-Eintrag		Parameter		Beschreibung
Code	Klartext	Code	Klartext	
3015	reserviert			
3016	AMX_ERR_MEMORY	##	---	out of memory
3017	AMX_ERR_FORMAT	##	---	ungültiges/nicht unterstütztes P-Code Dateiformat
3018	AMX_ERR_VERSION	##	---	Datei ist für eine neuere Version des AMX
3019	AMX_ERR_NOTFOUND	##	---	Datei oder Funktion nicht gefunden
3020	AMX_ERR_INDEX	##	---	ungültiger Index-Parameter (ungültiger Einstiegspunkt)
3021	AMX_ERR_DEBUG	##	---	Debugger kann nicht ausgeführt werden
3022	AMX_ERR_INIT	##	---	AMX nicht initialisiert (oder doppelt initialisiert)
3023	AMX_ERR_USERDATA	##	---	Benutzer-Datenfeld kann nicht gesetzt werden (Tabelle voll)
3024	AMX_ERR_INIT_JIT	##	---	JIT kann nicht initialisiert werden.
3025	AMX_ERR_PARAMS	##	---	fehlerhafter Parameter
3026	AMX_ERR_DOMAIN	##	---	Domain-Fehler. Das Ergebnis des Ausdrucks ist nicht im gültigen Bereich.
3027	AMX_ERR_GENERAL	##	---	allgemeiner Fehler (unbekannter oder nicht spezifizierter Fehler)
3028	AMX_ERR_OVERLAY	##	---	Overlays werden nicht unterstützt (JIT) oder sind nicht initialisiert.

11.6 Syntax

11.6.1 Allgemeine Syntax

11.6.1.1 Format

Bezeichner, Zahlen und Zeichen werden durch Leerzeichen, Tabulatoren, Zeilenumbrüche und "Form Feed" getrennt. Eine Serie von einer oder mehreren dieser Separatoren wird als Leerraum erkannt.

11.6.1.2 Optionale Semikolons

Semikolons (um ein Statement zu beenden) sind optional, wenn sie am Ende einer Zeile auftreten. Semikolons sind notwendig, um mehrere Statements in einer Zeile zu trennen. Ein Ausdruck kann auf mehrere Zeilen aufgeteilt werden, jedoch müssen Postfix-Operatoren in derselben Zeile wie der Operand stehen.

11.6.1.3 Kommentare

Text zwischen den Symbolen `/*` und `*/` (beide Symbole können auf derselben oder auf unterschiedlichen Zeilen stehen) und Text nach `//` (bis zum Ende einer Zeile) sind Kommentare. Kommentare dürfen nicht verschachtelt werden. Der Compiler betrachtet Kommentare als Leerzeichen. Ein Kommentar, der mit `/**` (zwei Sterne und ein Leerzeichen nach dem zweiten Stern) beginnt und mit einem `*/` endet, ist ein Dokumentationskommentar. Ein Kommentar, der mit `///` (drei Schrägstriche und ein Leerzeichen nach dem dritten Schrägstrich) beginnt, ist ebenfalls ein Dokumentationskommentar. Der Parser kann den Dokumentationskommentar in unterschiedlicher Weise unterstützen, zum Beispiel könnte er eine Online-Hilfe daraus generieren.

11.6.1.4 Bezeichner

Namen von Variablen, Funktionen und Konstanten. Bezeichner bestehen aus den Zeichen `a...z`, `A...Z`, `0...9`, `_` oder `@`. Das erste Zeichen darf keine Ziffer sein. Die Zeichen `@` und `_` alleine sind keine gültigen Bezeichner, z.B. `"_Up"` ist ein gültiger Bezeichner, aber `"_"` ist es nicht. Pawn unterscheidet zwischen Groß- und Kleinschreibung. Der Parser schneidet Bezeichner ab einer bestimmten Länge ab. Es werden standardmäßig nur die ersten 16 Zeichen für die Unterscheidung herangezogen.

11.6.1.5 Reservierte Schlüsselworte

Statements	Operator	Direktiven	Andere
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			

11.6.1.6 Numerische Konstanten

11.6.1.6.1 Numerische Integer-Konstanten

Binär

Ob gefolgt von einer Serie von 0 und 1

Dezimal

eine Serie von Ziffern zwischen 0 und 9

Hexadezimal

0x gefolgt von einer Serie von Ziffern zwischen 0 und 9 und den Buchstaben a bis f

11.6.1.6.2 Numerische Gleitkomma-Konstanten

Eine Gleitkommazahl ist eine Zahl mit einem Nachkommateil. Eine Gleitkommazahl beginnt mit einer oder mehreren Ziffern, beinhaltet einen Dezimaltrennpunkt und hat zumindest eine Ziffer nach dem Dezimaltrennpunkt. z.B. "12.0" und "0.75" sind gültige Gleitkommazahlen. Optional kann noch ein Exponent angehängt werden. Die Notation ist der Buchstabe "e" (Kleinbuchstabe), gefolgt von einer ganzzahligen numerischen Konstante. Z.B. "3.12e4" oder "12.3e-3" sind gültige Gleitkommazahlen mit Exponent.

11.6.2 Variablen

11.6.2.1 Deklaration

Das Schlüsselwort "new" deklariert eine neue Variable. Für spezielle Deklarationen wird das Schlüsselwort "new" durch "static" ersetzt (siehe "Statische lokale Deklaration" auf Seite 181). Sofern sie nicht explizit initialisiert wird, ist der Wert der neuen Variablen Null.

Eine Variablendeklaration kann auftreten

- an jeder Position, an der ein Ausdruck gültig ist - lokale Variable
- an jeder Position, an der eine Funktionsdeklaration oder eine Implementation der Funktion gültig ist - globale Variablen;
- im ersten Ausdruck einer "for" Schleife (siehe "for (Ausdruck 1 ; Ausdruck 2 ; Ausdruck 3) Statement " auf Seite 192) - lokale Variable

Beispiel:

```
new a;          // ohne Initialisierung (Wert ist 0)
new b = 3;     // mit Initialisierung (Wert ist 3)
```

11.6.2.2 Lokale Deklaration

Eine lokale Deklaration erscheint innerhalb eines Anweisungs-Blocks. Auf eine Variable kann nur innerhalb dieses Blocks und der darin enthaltenen Blöcke zugegriffen werden. Eine Deklaration innerhalb des ersten Ausdrucks einer Schleifenanweisung ist ebenfalls eine lokale Deklaration.

11.6.2.3 Globale Deklaration

Eine globale Deklaration erscheint außerhalb einer Funktion und eine globale Variable kann in jeder Funktion verwendet werden. Globale Variablen können nur mit konstanten Ausdrücken initialisiert werden.

11.6.2.4 Statische lokale Deklaration

Eine lokale Variable wird zerstört, wenn die Ausführung den Block verlässt, in dem die Variable geschaffen wurde. Lokalen Variablen in einer Funktion existieren nur während der Laufzeit der genannten Funktion. Jede neuer Aufruf der Funktion erstellt und initialisiert neue lokale Variablen. Wenn eine lokale Variable mit dem Schlüsselwort "static" anstatt "new" deklariert ist, bleibt die Variable auch nach dem Ende einer Funktion im Speicher. Dies bedeutet, dass statische lokale Variablen eine private, dauerhafte Speicherung bereitstellen, die nur in einer einzigen Funktion (oder einem Block) zugänglich sind. Wie globale Variablen, können statische lokale Variablen nur mit konstanten Ausdrücken initialisiert werden.

11.6.2.5 Statische globale Deklaration

Eine statische globale Variable verhält sich wie eine globale Variable, mit dem Unterschied, dass die Variable nur in der Datei gültig ist, in der sie deklariert wurde. Um eine globale Variable statisch zu deklarieren, ersetzen Sie das Schlüsselwort "new" mit "static".

11.6.2.6 Gleitkommawerte

Pawn unterstützt Gleitkommawerte. Diese können an jeder Stelle eingesetzt werden, an der eine Variablendeklaration gültig ist.

Beispiel:

```
new Float:a;           // ohne Initialisierung (Wert ist 0.0)
new Float:b = 3.0;    // mit Initialisierung (Wert ist 3.0)
```

11.6.3 Konstante Variablen

Es ist manchmal notwendig eine Variable zu erstellen, die einmal initialisiert wird und dann nicht mehr verändert werden soll. Eine solche Variable verhält sich ähnlich wie eine symbolische Konstante, aber sie ist dennoch eine Variable. Um eine konstante Variable zu deklarieren, legen Sie das Schlüsselwort "const" zwischen das Schlüsselwort, das die Variablendeklaration ("new", "static") startet und den Namen der Variablen.

Beispiel:

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

Typische Situationen, in denen man eine konstante Variable nutzen könnte, sind:

- Um eine "array"-Konstante zu erstellen. Auf symbolische Konstanten kann nicht per Index zugegriffen werden.
- Ein besonderer Fall ist, wenn die Array-Argumente in einer Funktion als "const" markiert werden. Array-Argumente werden immer per Referenz übergeben. Wenn sie als "const" deklariert werden, schützt sie das vor ungewollten Änderungen. Siehe Beispiele von "const-Funktionsargumenten" im Kapitel "Funktionsargumente ("call-by-value" versus "call-by-reference")" auf Seite 195.

11.6.4 Array Variablen

11.6.4.1 Eindimensionales Array

Die Syntax `name[constant]` deklariert "name" als ein Array aus "constant" Elementen, wobei jedes Element ein Eintrag ist. "name" ist ein Platzhalter für den Namen der Variable und "constant" ist ein positiver Wert ungleich Null. "constant" ist optional und kann weggelassen werden. Wenn kein Wert zwischen den Klammern steht, ist die Anzahl von Elementen gleich der Anzahl der Initialwerte. Der Array-Index-Bereich ist "Null-basierend", das bedeutet, dass das erste Element "name[0]" und das letzte Element "name[constant-1]" ist.

11.6.4.2 Initialisierung

Datenobjekte können bei ihrer Deklaration initialisiert werden. Der initialisierte Wert von globalen Datenobjekten muss ein konstanter Wert sein. Arrays, global oder lokal, müssen ebenfalls mit konstanten Werten initialisiert werden. Nicht initialisierte Daten sind standardmäßig Null.

Beispiele:

Auflistung: gültige Deklaration

```
new i = 1
new j           /* j ist 0 */
new k = 'a'     /* k hat den Zeichencode von 'a' */
new a[] = [1,4,9,16,25] /* a hat 5 Elemente */
new s1[20] = ['a','b'] /* die restlichen 18 Elemente sind 0 */
new s2[] = ''Hello world...'' /* ein unpacked string */
```

Auflistung: ungültige Deklaration

```
new c[3] = 4           /* Ein Array kann nicht auf einen einzelnen
                       Wert gesetzt werden */
new i = "Good-bye"   /* Nur ein Array kann einen String halten. */
new q[]              /* Unbekannte Größe für ein Array */
new p[2] = { i + j, k - 3 } /* Arrayinitialisierer müssen Konstanten sein. */
```

11.6.4.3 Progressive Initialisierung für Arrays

Der Punkte-Operator führt die Initialisierung des Arrays aufgrund der letzten beiden initialisierten Werte weiter. Der Punkte-Operator (drei Punkte, "...") initialisiert das Array bis zur Arraygrenze.

Beispiel: Auflistung: Arrayinitialisierer

```
new a[10] = { 1, ... }           // setzt alle Elemente auf 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
```

11.6.4.4 Mehrdimensionale Arrays

(Es werden nur Arrays mit bis zu 3 Dimensionen unterstützt)

Mehrdimensionale Arrays sind Arrays, die Referenzen zu weiteren Sub-Arrays enthalten. Zum Beispiel ist ein zweidimensionales Array ein "Array auf Eindimensionale Arrays".

Beispiele für die Deklaration von zweidimensionalen 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'' ]
```

Wie die beiden letzten Deklarationen (Variablen "e" und "f") zeigen, hat die letzte Dimension eine nicht spezifizierte Länge. In diesem Fall wird die Länge des Sub-Arrays aus dem dazugehörigen Initialisierer erkannt. Jedes Sub-Array hat eine unterschiedliche Länge. In diesem speziellen Beispiel enthält "e[1][5]" den Buchstaben "l" des Wortes "Cancel", aber "e[0][5]" ist ungültig, da das Sub-Array e[0] nur drei Einträge (die Buchstaben "O", "K", und den Null-Terminator) beinhaltet. Der Unterschied zwischen den Deklarationen der

Arrays "e" und "f" ist, dass wir bei "f" den Compiler die Anzahl der höheren Dimension ermitteln lassen. "sizeof f" ist 2 genauso wie "sizeof e" (siehe "Arrays und der "sizeof"-Operator" auf Seite 184).

11.6.4.5 Arrays und der "sizeof"-Operator

Der "sizeof"-Operator gibt die Anzahl der Elemente einer Variablen zurück. Für eine einfache (nicht Array) Variable ist das Ergebnis von "sizeof" immer 1.

Ein Array mit einer Dimension enthält eine Anzahl von Elementen und der "sizeof"-Operator gibt diese Anzahl zurück. Der Codeausschnitt unterhalb würde deshalb "5" ausgeben, da das Array 4 Zeichen und den Null-Terminator enthält.

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

Der "sizeof"-Operator gibt immer die Anzahl der Einträge, auch für ein "packed" Array, zurück. Der Codeausschnitt unterhalb gibt auch "5" aus, da die Variable 5 Einträge enthält, auch wenn diese im Speicher weniger Platz benötigt.

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

Bei mehrdimensionalen Arrays kann der "sizeof"-Operator die Anzahl der Elemente jeder Dimension zurückgeben. Für die letzte (niedrigste) Dimension ist ein Element ein einzelner Eintrag, jedoch für die höchste Dimension ist es ein Sub-Array. Beachten Sie, dass im nachfolgenden Codeausschnitt die Syntax "sizeof matrix" die Anzahl der Elemente der höheren Dimension zurückgibt, und dass die Syntax "sizeof matrix[]" die niedrigere Dimension des zweidimensionalen Arrays ausgibt. Der Codeausschnitt gibt 3 (höhere Dimension) und 2 (niedrigere Dimension) aus.

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

Die Anwendung des "sizeof"-Operators auf mehrdimensionale Arrays ist besonders praktisch, wenn er als Standardwert für Funktionsargumente verwendet wird.

11.6.5 Operatoren und Ausdrücke

11.6.5.1 Zeichenerklärung

Die Anwendung von einigen Operatoren hängt von der jeweiligen Art des Operanden ab. Aus diesem Grund wird in diesem Kapitel folgende Notation angewendet:

- e** *beliebiger Ausdruck (eng. expression)*
- v** *beliebiger Ausdruck, dem ein Wert zugewiesen werden kann ("lvalue" Ausdruck - Variable)*
- a** *ein Array*
- f** *eine Funktion*
- s** *ein Symbol - dies kann eine Variable, eine Konstante oder eine Funktion sein*

11.6.5.2 Ausdrücke

Ein Ausdruck besteht aus ein oder mehreren Operanden mit einem Operator. Der Operand kann eine Variable, eine Konstante oder ein anderer Ausdruck sein. Ein Ausdruck gefolgt von einem Semikolon ist ein Statement.

Beispiele für Ausdrücke:

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

11.6.5.3 Arithmetik

Operator	Beispiel	Erklärung
+	e1 + e2	Ergebnis der Addition von e1 und e2
-	e1 - e2	Ergebnis der Subtraktion von e1 und e2
	-e	Ergebnis der arithmetischen Negation von e (Zweierkomplement)
*	e1 * e2	Ergebnis der Multiplikation von e1 und e2
/	e1 / e2	Ergebnis der Division e1 durch e2. Das Ergebnis wird zum nächstgelegenen ganzzahligen Wert, der kleiner oder gleich dem Quotienten ist, abgeschnitten. Sowohl positive als auch negative Werte werden abgerundet (Richtung unendlich).
%	e1 % e2	Ergebnis ist der Rest der Division e1 durch e2. Das Vorzeichen ist dasselbe wie bei e2
++	v++	erhöht v um 1. Das Ergebnis des Ausdrucks ist der Wert vor der Erhöhung.
	++v	erhöht v um 1. Das Ergebnis des Ausdrucks ist der Wert nach der Erhöhung.
--	v--	verringert v um 1. Das Ergebnis des Ausdrucks ist der Wert vor der Verringerung.
	--v	verringert v um 1. Das Ergebnis des Ausdrucks ist der Wert nach der Verringerung.

Hinweis: Das unäre + ist in Pawn nicht definiert. Die Operatoren ++ und -- ändern den Operanden. Der Operand muss ein "lvalue" sein.

11.6.5.4 Bit-Manipulation

Operator	Beispiel	Erklärung
~	~e	Ergebnis ist das Einerkomplement von e.
>>	e1 >> e2	Ergebnis der arithmetischen Verschiebung nach rechts von e1 durch e2 Bits. Die Verschiebung ist vorzeichenbehaftet: Das Bit ganz links wird auf die freien Bits des Ergebnisses kopiert.
>>>	e1 >>> e2	Ergebnis der logischen Verschiebung nach rechts von e1 durch e2 Bits. Die Verschiebung ist vorzeichenlos. Die freien Bits des Ergebnisses werden mit 0 aufgefüllt.
<<	e1 << e2	Ergebnis: Verschiebung nach links von e1 durch e2 Bits. Die freien Bits des Ergebnisses werden mit 0 aufgefüllt. Es gibt keinen Unterschied zwischen einer arithmetischen und einer logischen Verschiebung nach links.
&	e1 & e2	Ergebnis ist das bitweise logische "und" von e1 und e2.
	e1 e2	Ergebnis ist das bitweise logische "oder" von e1 und e2.
^	e1 ^ e2	Ergebnis ist das bitweise "exklusiv oder" von e1 und e2.

11.6.5.5 Zuweisung

Das Ergebnis eines Zuweisungsausdrucks ist der Wert des Operanden nach der Zuweisung.

Operator	Beispiel	Erklärung
=	$v = e$	weist den Wert von e der Variable v zu
	$v = a$	weist das Array a der Variable v zu. v muss ein Array mit derselben Größe und denselben Dimensionen sein wie a . a kann eine Zeichenkette oder ein Array sein.

Hinweis: Die folgenden Operatoren kombinieren eine Zuweisung mit einer arithmetischen oder bitweisen Operation. Das Ergebnis des Ausdrucks ist der Wert des linken Operanden nach der arithmetischen oder bitweisen Operation.

Operator	Beispiel	Erklärung
$+=$	$v += e$	erhöht v um e
$-=$	$v -= e$	vermindert v um e
$*=$	$v *= e$	multipliziert v mit e
$/=$	$v /= e$	dividiert v mit e
$\%=$	$v \% = e$	weist v den Rest der Division von v und e zu
$>>=$	$v >> = e$	verschiebt v arithmetisch um e Bits nach rechts
$>>>=$	$v >>> = e$	verschiebt v logisch um e Bits nach rechts
$<<=$	$v << = e$	verschiebt v um e Bits nach links
$\&=$	$v \& = e$	führt ein bitweises "und" von v und e aus und weist das Ergebnis v zu
$ =$	$v = e$	führt ein bitweises "oder" von v und e aus und weist das Ergebnis v zu
$\wedge=$	$v \wedge = e$	führt ein bitweises "exklusiv oder" von v und e aus und weist das Ergebnis v zu

11.6.5.6 Vergleichsoperatoren

Ein logisches "false" wird durch einen Integer-Wert von 0 repräsentiert; ein logisches "true" durch einen Wert, der nicht 0 ist. Ergebnisse eines Vergleichs-Ausdrucks sind entweder 0 oder 1 und ihr "tag" wird auf "bool" gesetzt.

Operator	Beispiel	Erklärung
$==$	$e1 == e2$	Das Ergebnis ist "true", wenn $e1$ und $e2$ gleich sind.
$!=$	$e1 != e2$	Das Ergebnis ist "true", wenn $e1$ nicht gleich $e2$ ist.

Hinweis: Die folgenden Operatoren können verkettet werden, wie im Ausdruck " $e1 \leq e2 \leq e3$ ". Dies bedeutet, dass das Ergebnis "1" ist, wenn jeder einzelne Vergleich zutrifft und "0", wenn zumindest ein Vergleich nicht zutrifft.

Operator	Beispiel	Erklärung
$<$	$e1 < e2$	Das Ergebnis ist ein logisches "true", wenn $e1$ kleiner ist als $e2$.
\leq	$e1 \leq e2$	Das Ergebnis ist ein logisches "true", wenn $e1$ kleiner oder gleich $e2$ ist.
$>$	$e1 > e2$	Das Ergebnis ist ein logisches "true", wenn $e1$ größer ist als $e2$.
\geq	$e1 \geq e2$	Das Ergebnis ist ein logisches "true", wenn $e1$ größer oder gleich $e2$ ist.

11.6.5.7 Boolean

Ein logisches "false" wird durch einen Integer-Wert von 0 repräsentiert, ein logisches "true" durch einen Wert, der nicht 0 ist. Ergebnisse eines Vergleichs-Ausdrucks sind entweder 0 oder 1 und ihr "tag" wird auf "bool" gesetzt.

Operator	Beispiel	Erklärung
!	!e	Das Ergebnis ist ein logisches "true", wenn e logisch "false" ist.
	e1 e2	Das Ergebnis ist "true", wenn entweder e1 oder e2 (oder beide) logisch "true" sind. Der Ausdruck e2 wird nur ausgewertet, wenn e1 logisch "false" ist.
&&	e1 && e2	Das Ergebnis ist "true", wenn e1 und e2 logisch "true" sind. Der Ausdruck e2 wird nur ausgewertet, wenn e1 logisch "true" ist.

11.6.5.8 Sonstiges

Operator	Beispiel	Erklärung
[]	a[e]	Array Index: Das Ergebnis ist der Eintrag an der Position e des Arrays a.
{}	a{e}	Array Index: Das Ergebnis ist das Zeichen an der Position e des "packed" Arrays a.
()	f(e1, e2, ... eN)	Das Ergebnis ist der Wert, der von der Funktion f zurückgegeben wird. Die Funktion wird mit den Parametern e1, e2, ... eN aufgerufen. Die Reihenfolge der Auswertung der Parameter ist nicht definiert. (Die Implementation der Script-Engine wertet die Parameter möglicherweise in umgekehrter Reihenfolge aus.)
?:	e1 ? e2 : e3	Das Ergebnis ist entweder e2 oder e3, abhängig vom Wert von e1. Der bedingte Ausdruck ist ein zusammengesetzter Ausdruck mit einem zweiseitigen Operator, "?" und ":". Der Ausdruck e2 wird ausgewertet, wenn e1 logisch "true" ist, e3 wird ausgewertet, wenn e1 logisch "false" ist.
:	tagname: e	"tag" Überschreibung: Der Wert des Ausdrucks ändert sich nicht, jedoch ändert sich der "tag".
defined	defined s	Ergebnis ist "1", wenn das Symbol definiert wurde. Das Symbol kann eine Konstante oder eine globale oder lokale Variable sein. Der "tag" des Ausdrucks ist "bool".
sizeof	sizeof s	Das Ergebnis ist die Anzahl der Elemente der angegebenen Variable. Für einfache Variablen und für eindimensionale Arrays ist ein Element ein Eintrag. Für mehrdimensionale Arrays ist das Ergebnis die Anzahl der Elemente (Sub Arrays) in der höchsten Dimension. Fügen Sie [] zum Namen des Arrays hinzu, um eine niedrigere Dimension anzugeben. Wenn die Größe der Variable nicht bekannt ist, dann ist das Ergebnis 0. Wenn dieser Operator in einem "default"-Wert einer Funktion verwendet wird, dann wird der Ausdruck zum Zeitpunkt des Aufrufs der Funktion und nicht zum Zeitpunkt der Definition ausgeführt.
tagof	tagof s	Das Ergebnis ist eine eindeutige Zahl, die den "tag" der Variablen, der Konstanten, des Rückgabewerts einer Funktion oder des Names der "tag"-Bezeichnung repräsentiert. Wenn dieser Operator in einem "default"-Wert einer Funktion verwendet wird, dann wird der Ausdruck zum Zeitpunkt des Aufrufs der Funktion und nicht zum Zeitpunkt der Definition ausgeführt.

11.6.5.9 Priorität der Operatoren

Die nachfolgende Tabelle gruppiert Operatoren mit derselben Priorität, beginnend mit der höchsten Priorität.

Wenn die Auswertung eines Ausdrucks nicht explizit durch Klammern begründet wird, wird sie von den Assoziationsregeln bestimmt. Zum Beispiel: $a*b/c$ ist gleich $(a*b)/c$ auf Grund der links zu rechts Assoziation, und $a=b=c$ ist gleichzusetzen mit $a=(b=c)$.

Operator	Erklärung	Lesefolge
() [] { }	Funktionsaufruf Array Index (Element) Array Index (Zeichen)	links-nach-rechts
! ~ - ++ -- : defined sizeof tagof	logisches Nicht Einernkomplement Zweierkomplement (unäres Minus) Erhöhung Verringerung "tag"-Überschreibung Symbol Definitions-Status Symbol Größe in "Elementen" eindeutige Zahl des "tags"	rechts-nach-links
* / %	Multiplikation Division Modulo	links-nach-rechts
+ -	Addition Subtraktion	links-nach-rechts
>> >>> <<	arithmetische Verschiebung nach rechts logische Verschiebung nach rechts Verschiebung nach links	links-nach-rechts
&	bitweises "und"	links-nach-rechts
^	bitweises "exklusiv oder"	links-nach-rechts
	bitweises "oder"	links-nach-rechts
< <= > >=	kleiner als kleiner oder gleich als größer als größer oder gleich als	links-nach-rechts
== !=	gleich ungleich	links-nach-rechts
&&	logisches "und"	links-nach-rechts
	logisches "oder"	links-nach-rechts
? :	bedingte Ausführung	rechts-nach-links
=	Zuweisung *= /= %= += -= >>= >>>= <<= &= ^= =	rechts-nach-links
,	Komma	links-nach-rechts

11.6.6 Anweisungen

Ein Statement kann aus einer oder mehreren Zeilen bestehen. Eine Zeile kann zwei oder mehrere Statements enthalten.

Statements zur Ablaufsteuerung (if, if-else, for, while, do-while und switch) können geschachtelt werden.

11.6.6.1 Statement-Etikett

Ein Etikett besteht aus einem Identifizierer gefolgt von einem ":". Ein Etikett ist ein "Sprung-Ziel" eines "goto" Statements.

Jede Anweisung kann mit einem Etikett versehen werden. Es muss dem Etikett ein Statement folgen, dies kann auch ein "leeres Statement" sein.

Der Gültigkeitsbereich eines Etiketts ist die Funktion in der es deklariert wurde d.h. ein "goto"-Statement kann nicht aus der aktuellen Funktion in eine andere Funktion springen.

11.6.6.2 Zusammengesetzte Anweisungen

Eine zusammengesetzte Anweisung (auch Block genannt) ist eine Serie von Null oder mehreren Anweisungen, welche durch Klammern ("{" und "}") umgeben ist. Die schließende Klammer ("}") darf nicht mit einem Semikolon abgeschlossen werden. Jede Anweisung kann durch einen Block ersetzt werden. Eine zusammengesetzte Anweisung, die keine Anweisungen enthält, ist ein Spezialfall und wird "leeres Statement" genannt.

11.6.6.3 Ausdrucksanweisung

Jeder Ausdruck wird zu einem Statement, wenn ein Semikolon (";") angehängt wird. Ein Ausdruck wird auch zu einem Statement, wenn dem Ausdruck bis zum Ende der Zeile nur Leerzeichen folgen und der Ausdruck nicht in der nächsten Zeile weitergeführt werden kann.

11.6.6.4 Leeres Statement

Eine leeres Statement führt keine Anweisungen aus und besteht aus einem Block-Statement ohne Anweisungen, d.h. es besteht aus dem Symbol "{}". Leere Statements werden in Kontrollflussanweisungen ohne Aktionen eingesetzt (z.B. "while (!iskey()) {}") oder, wenn ein Etikett genau vor einer schließenden Klammer eines Block-Statements definiert wird. Ein leeres Statement endet nicht mit einem Semikolon.

11.6.6.5 assert Ausdruck

bricht das Programm mit einem Laufzeitfehler ab, wenn der Ausdruck logisch "false" ergibt

Hinweis: Dieser Ausdruck schützt vor "unmöglich" oder ungültigen Bedingungen. Im folgenden Beispiel ist eine negative Fibonacci-Zahl ungültig. Die assert-Anweisung markiert diesen Fehler als Programmierer-Fehler. assert-Anweisungen sollten nur Programmierer-Fehler kennzeichnen und niemals Benutzereingaben.

Beispiel:

```
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
}
```

11.6.6.6 break

beendet und verlässt das kleinste, umschließende "do"-, "for"- oder "while"-Statement an jedem beliebigen Punkt in der Schleife. Das "break"-Statement bewegt den Programmfluss zum nächsten Statement außerhalb der Schleife.

Beispiel:

```
example(n)
{
    new a = 0

    for(new i = 0; i < n ; i++ )
    {
        a += i

        if(i>10)
            break

        a += 1
    }
    return a
}
```

11.6.6.7 continue

beendet die aktuelle Iteration der kleinsten umschließenden "do"-, "for"- oder "while"-Anweisung und bewegt die Programmsteuerung an den Bedingungsteil der Schleife.

Beispiel

```
example(n)
{
    new a = 0

    for(new i = 0; i < n ; i++ )
    {
        a += i

        if(i>10)
            continue

        a += 1
    }
    return a
}
```

11.6.6.8 do Statement while (Ausdruck)

führt ein Statement aus, bevor der Bedingungsteil (die "while"-Bedingung) evaluiert wird. Das Statement wird wiederholt, solange die Bedingung logisch "true" ist. Das Statement wird zumindest einmal ausgeführt.

Beispiel:

```
example(n)
{
    new a = 0

    do
    {
        a++
    }
    while(n >= 0)

    return a
}
```

11.6.6.9 exit Ausdruck

bricht das Programm ab. Der Ausdruck ist optional, aber wenn er vorhanden ist, muss er in der selben Zeile wie das "exit"-Statement beginnen und enden. Die "exit"-Anweisung gibt den Wert des Ausdrucks zurück an die Hauptanwendung oder gibt Null zurück, wenn kein Ausdruck angegeben wird.

11.6.6.10 for (Ausdruck 1 ; Ausdruck 2 ; Ausdruck 3) Statement

Alle drei Ausdrücke sind optional.

Ausdruck 1:

wird nur einmal ausgewertet, vor Eintritt in die Schleife. Dieser Ausdruck kann zum Initialisieren einer Variablen genutzt werden. Dieser Ausdruck hält auch die Variablendeklaration mittels der "new"-Syntax. Eine Variable, die an dieser Stelle deklariert wird, ist nur innerhalb der Schleife gültig. Es ist nicht möglich einen Ausdruck (mit bereits vorhandenen Variablen) und eine Deklaration von neuen Variablen in diesem Feld zu kombinieren. Es müssen entweder alle Variablen in diesem Feld bereits vorhanden sein, oder sie müssen alle in diesem Bereich deklariert werden.

Ausdruck 2:

Dieser Ausdruck wird vor jedem Durchlauf der Schleife ausgeführt und beendet die Schleife, wenn der Ausdruck logisch "false" zurückgibt. Wenn dieser Ausdruck weggelassen wird, wird das Ergebnis des Ausdrucks 2 als logisch "true" angenommen.

Ausdruck 3:

Dieser Ausdruck wird nach jeder Ausführung des Statements ausgeführt. Die Programmsteuerung bewegt sich von Ausdruck 3 zum Ausdruck 2 für die nächste (bedingte) Iteration der Schleife.

Beispiel:

```
example (n)
{
    new a = 0

    for(new i = 0; i < n; i++)
    {
        a++
    }

    return a
}
```

Das Statement "for (; ;)" ist gleich dem Statement "while (true)".

11.6.6.11 goto Etikett

bewegt die Programmsteuerung (unbedingt) zu der Anweisung, die dem angegebenen Etikett folgt. Das Etikett muss innerhalb der gleichen Funktion wie die "goto"-Anweisung sein. Eine "goto"-Anweisung kann nicht aus einer Funktion springen.

11.6.6.12 if (Ausdruck) Statement 1 else Statement 2

führt das Statement 1 aus, wenn das Ergebnis des Ausdrucks logisch "true" ergibt. Die "else"-Klausel des "if"-Statements ist optional. Wenn das Ergebnis des Ausdruck logisch "false" ergibt und eine "else"-Klausel existiert, dann wird das Statement, das mit der "else"-Klausel assoziiert ist, (Statement 2) ausgeführt.

Beispiel:

```
example(n)
{
    if(n < 0)
        return -1
    else if (n == 0)
        return 0
    else
        return 1
}
```

11.6.6.13 return Ausdruck

beendet die aktuelle Funktion und bewegt die Programmsteuerung zum nächsten Statement nach dem Funktionsaufruf. Der Wert des Ausdrucks wird als Funktionsergebnis zurückgeliefert. Der Ausdruck kann ein Array oder eine Zeichenfolge sein. Der Ausdruck ist optional, wenn er jedoch vorhanden ist, muss er an der selben Zeile beginnen, wie das "return"-Statement. Wenn kein Ausdruck angegeben ist, dann wird Null zurückgeliefert.

11.6.6.14 switch (Ausdruck) { case Liste }

überträgt die Ablaufsteuerung an unterschiedliche Statements innerhalb des "switch" in Abhängigkeit vom Wert des "switch"-Ausdrucks. Der Hauptteil der "switch"-Anweisung ist eine zusammengesetzte Anweisung, die eine Reihe von "case"-Klauseln enthält. Jede "case"-Klausel beginnt mit dem Schlüsselwort "case", gefolgt von einer Liste von Konstanten und einem Statement. Die Liste der Konstanten ist eine Serie von Ausdrücken, getrennt durch Kommas, die jeweils zu einem konstanten Wert ausgewertet werden. Diese Liste endet mit einem Doppelpunkt. Um einen Bereich in dieser Liste anzugeben, trennen Sie die untere und obere Grenze des Bereichs mit einem doppelten Punkt (".."). Ein Beispiel für einen Bereich ist: "case 1..9:".

Das "switch"-Statement bewegt die Ablaufsteuerung zu einer "case"-Klausel, wenn ein Wert der Liste dem Wert des "switch"-Ausdrucks entspricht.

Die "default"-Klausel besteht aus dem Schlüsselwort "default" und einem Doppelpunkt. Die "default"-Klausel ist optional, aber wenn sie angegeben wird, muss sie als letzter Eintrag in der "case"-Liste eingetragen sein. Das "switch"-Statement bewegt die Ablaufsteuerung zur "default"-Klausel, wenn keine der "case"-Klauseln mit dem "switch"-Ausdruck übereinstimmt.

Beispiel:

```
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
}
```

11.6.6.15 while (Ausdruck) Statement

wertet den Ausdruck aus und führt das Statement aus, wenn das Ergebnis des Ausdrucks logisch "true" ergibt. Nachdem die Anweisung ausgeführt wurde, kehrt die Programmsteuerung erneut zu dem Ausdruck zurück. Das Statement wird daher ausgeführt, solange der Ausdruck logisch "true" ist.

Beispiel:

```
example(n)
{
    new a = 0

    while(n >= 0)
    {
        a++
    }

    return a
}
```

11.6.7 Funktionen

Eine Funktionsdeklaration spezifiziert den Namen der Funktion und die formalen Parameter, die in Klammern eingeschlossen sind. Eine Funktion kann auch einen Wert zurückliefern. Eine Funktion muss global definiert, d.h. außerhalb einer anderen Funktion deklariert werden und ist global verfügbar.

Wenn ein Semikolon der Funktionsdeklaration folgt (anstatt einer Anweisung), dann ist dies eine Vorwärtsdeklaration einer Funktion.

Die "return"-Anweisung setzt den Rückgabewert der Funktion. Zum Beispiel, hat die Funktion "sum" (siehe unten) als Rückgabewert die Summe der beiden Parameter. Der "return"-Ausdruck ist optional.

```
sum(a, b)
{
    return a + b
}
```

Argumente einer Funktion sind (implizit deklarierte) lokale Variablen für diese Funktion. Der Funktionsaufruf bestimmt die Werte der Argumente. Ein weiteres Beispiel für eine vollständige Definition einer Funktion ist "leapyear", die "true" für ein Schaltjahr und "false" für kein Schaltjahr zurückgibt.

```
leapyear(y)
{
    return y % 4 == 0 && y % 100 != 0 || y % 400 == 0
}
```

Die Anweisungen, die in diesem Beispiel verwendet wurden, werden im Kapitel "Operatoren und Ausdrücke" auf Seite 184 behandelt.

Normalerweise beinhalten Funktionen lokale Variablendeklarationen und bestehen aus einer Block-Anweisung.

Hinweis: Im nächsten Beispiel verhindert die "assert"-Anweisung negative Werte für den Exponenten

```
power(x, y)
{
    /* gibt x hoch y zurück*/
    assert y >= 0

    new r = 1
    for (new i = 0; i < y; i++)
        r *= x

    return r
}
```

Eine Funktion kann mehrere "return"-Anweisungen enthalten, eine wird z.B. benutzt um schnell eine Funktion zu beenden, wenn ungültige Parameter übergeben werden, oder wenn sich herausstellt, dass die Funktion nichts zu tun hat. Wenn eine Funktion ein Array zurückgibt, müssen alle "return" Anweisungen ein Array mit derselben Anzahl von Einträgen zurückgeben.

11.6.7.1 Funktionsargumente ("call-by-value" versus "call-by-reference")

Die "faculty"-Funktion im nächsten Beispiel hat einen Parameter, der in der Schleife benutzt wird um die Fakultät dieser Zahl zu berechnen. Was Aufmerksamkeit verdient ist, dass die Funktion das Argument modifiziert.

```
main()
{
    new v = 5
    new f = faculty(v)
}
```

```
faculty(n)
{
    assert n >= 0

    new result = 1
    while (n > 0)
        result *= n--

    return result
}
```

Egal welchen (positiven) Wert die Variable "n" am Beginn der "while"-Schleife hatte, am Ende der Funktion wird "n" Null sein. Am Beispiel der Funktion "faculty" wird der Parameter als Wert ("by value") übergeben, damit ist eine Änderung der Variable "n" nur lokal in der Funktion "faculty" gültig. Mit anderen Worten, die Variable "v" in der Funktion "main()" hat vor und nach dem Aufruf der Funktion denselben Wert.

Argumente können als Wert ("by value") oder als Referenz ("by reference") übergeben werden. Einem Funktionsargument, das als Referenz übergeben werden soll, muss als Präfix "&" dem Namen vorangestellt werden. Als Standard werden der Funktion die Argumente als Wert übergeben.

Beispiel:

```
swap(&a, &b)
{
    new temp = b
    b = a
    a = temp
}
```

Um eine Array einer Funktion zu übergeben, fügen Sie ein Klammernpaar ("[]") dem Namen des Arguments an. Es kann auch zusätzlich die Anzahl der Einträge angegeben werden. Dies verbessert die Fehlererkennung des Parsers des Compilers.

Beispiel:

```
addvector(a[], const b[], size)
{
    for (new i = 0; i < size; i++)
        a[i] += b[i]
}
```

Arrays werden immer als Referenz übergeben.

Hinweis: Das Array "b" im oben gezeigten Beispiel wird in der Funktion nicht verändert. Dieses Funktionsargument wurde als "const" deklariert, um dies explizit zu machen. Zusätzlich zur verbesserten Fehlererkennung, erlaubt es dem Compiler einen effizienteren Code zu generieren.

Das folgende Codebeispiel ruft die Funktion "addvector" auf und addiert zu jedem Element der Variablen "vect" den Wert 5:

```
new vect[3] = [ 1, 2, 3 ]

addvector(vect, [5, 5, 5], 3)

/* vect[] beinhaltet nun die Werte 6, 7 und 8 */
```

11.6.7.2 Benannte Parameter versus positionsgebundene Parameter

In den vorangegangenen Beispielen war die Reihenfolge der Parameter bei einem Funktionsaufruf wichtig, da jeder Parameter zu dem Funktionsparameter mit derselben Position kopiert wurde. Zum Beispiel bei der "weekday" Funktion, die unterhalb definiert wird, würde der Aufruf "weekday(12, 31, 1999)" lauten, um den Wochentag des letzten Tages des letzten Jahrhunderts zu erhalten.

```
weekday(month, day, year)
{
  /* gibt den Tag der Woche zurück: 0=Samstag, 1=Sonntag, 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
}
```

Das Datumsformat unterscheidet sich je nach Kultur und Nation, während in den Vereinten Staaten von Amerika das Format Monat/Tag/Jahr verbreitet ist, verwenden europäische Staaten oft das Format Tag/Monat/Jahr und in technischen Publikationen wird Jahr/Monat/Tag (ISO/IEC 8824) verwendet. Mit anderen Worten, keine Reihenfolge der Parameter ist "standardisiert" oder "normal". Aus diesem Grund gibt es eine alternative Möglichkeit, um Parameter an eine Funktion zu übergeben: die "benannten Parameter". Diese wird im nächsten Beispiel gezeigt (die Funktion wurde genau wie im vorherigen Beispiel deklariert).

```
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)
```

Bei "benannten Parametern" wird ein Punkt (".") dem Namen des Arguments vorangestellt. Das Argument der Funktion kann auf einen beliebigen Ausdruck gesetzt werden, der gültig für das Argument ist. Das Gleichheitszeichen ("=") hat im Falle eines benannten Parameters nicht die Bedeutung einer Zuordnung, sondern verknüpft den Ausdruck mit einem Funktionsargument.

Es können positionsgebundene und benannte Parameter vermischt werden, jedoch müssen die positionsgebundenen vor den benannten Parametern angegeben werden.

11.6.7.3 Standardwerte von Funktionsargumenten

Ein Funktionsargument kann einen Standardwert haben. Der Standardwert eines Funktionsarguments muss eine Konstante sein. Um einen Standardwert anzugeben, fügen Sie an den Namen des Parameters ein Gleichheitszeichen ("=") und den Wert an.

Wenn bei einem Funktionsaufruf ein Platzhalter anstelle eines gültigen Funktionsparameters angegeben wird, wird der Standardwert übernommen. Der Platzhalter ist das Unterstrichzeichen ("_"). Der Argumentplatzhalter ist nur für Parameter mit einem Standardwert gültig.

Die rechten Argumentplatzhalter können von der Argumentenliste entfernt werden.

Zum Beispiel, wenn die Funktion "increment" wie folgt definiert ist:

```
increment(&value, incr=1)
{
    value += incr
}
```

sind die folgenden Funktionsaufrufe alle gleich:

```
increment(a)
increment(a, _)
increment(a, 1)
```

Standardwerte für Argumente, die als Referenz übergeben werden, sind hilfreich um diese Parameter optional zu machen. Zum Beispiel, wenn die Funktion "divmod" geschrieben wurde, um sowohl den Quotienten als auch den Rest als Parameter zu übergeben.

```
divmod(a, b, &quotient=0, &remainder=0)
{
    quotient = a / b
    remainder = a % b
}
```

Mit der vorangegangenen Definition der Funktion "divmod" sind die folgenden Funktionsaufrufe alle gültig:

```
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
```

Das nächste Beispiel addiert die Werte von einem Array zu einem anderen. Wenn nur ein Parameter angegeben wird, dann werden die Werte des Arrays um 1 erhöht:

```
addvector(a[], const b[] = {1, 1, 1}, size = 3)
{
    for (new i = 0; i < size; i++)
        a[i] += b[i]
}
```

11.7 Beispiele

11.7.1 Sägezahngenerator

Dieses Beispiel erzeugt einen Sägezahn am Ausgang eines Geräts.

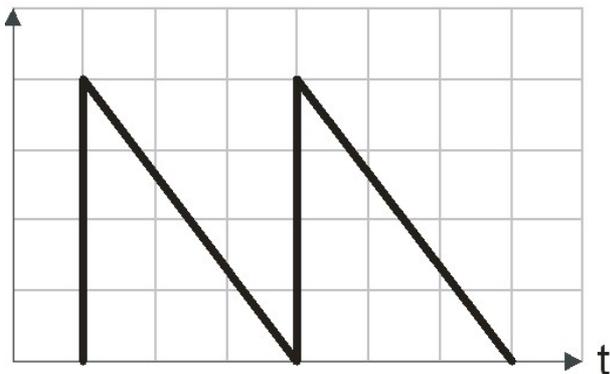
```
static Float:fCurrentValue;           // Statische Deklaration der Variable fCurrentValue vom
                                     // Typ Float. Diese Variable wird zwischen 2 Messzyklen
                                     // nicht zurückgesetzt

/*
  Dieser Code wird nur einmal beim
  - Starten des Gerätes
  - Ändern eines Scriptes
  ausgeführt.
  Dies kann zum Initialisieren von
  Variablen verwendet werden.
*/
main()
{
  fCurrentValue = 4.0;                // fCurrentValue mit dem Wert 4.0 initialisieren
}

/*
  Dieser Code wird am Ende jedes
  Messzyklus ausgeführt.
*/
public Mdn_CtrlFinish()
{
  if(fCurrentValue <= 0.0)           // Wenn die Variable kleiner oder gleich 0.0 ist,
    fCurrentValue = 4.0;              // setze die Variable auf 4.0.

  fCurrentValue -= 0.1;               // Ziehe 0.1 von der Variable ab.

  Mdn_SetCh(MDN_CH_IOUT1, fCurrentValue); // Setze den Ausgang 1 auf den Wert der Variable
                                     // fCurrentValue
}
```



Mittels Pawn erzeugter Sägezahn

11.7.2 Wochentagbestimmung

Dieses Beispiel ermittelt den Wochentag aus dem aktuellen Zeitstempel eines Geräts. Danach wird ein Zwischenwert berechnet, indem die Zeitdifferenz zwischen den Durchläufen des Scripts durch 6 dividiert wird. Bei 2min Messintervall ergibt sich so ein Wert von 20. Dieser Zwischenwert wird dann noch durch den ermittelten Wochentag dividiert und das Ergebnis dann am Ausgang 0 ausgegeben.

```
static iLastStamp; // Statische Deklaration der Variable
// iLastStampe vom Typ Integer (wird
// automatisch mit 0 vorinitialisiert)
// Diese Variable wird zwischen 2 Messzyklen
// nicht zurückgesetzt.

/* Dieser Code wird am Ende jedes
Messzyklus ausgeführt. */
public Mdn_CtrlFinish()
{
    new iDayOfWeek; // Deklaration der Variable iDayOfWeek vom
// Typ Integer
    new iCurrentStamp; // Deklaration der Variable iCurrentStamp vom
// Typ Integer
    new Float:fResult; // Deklaration der Variable fResult vom
// Typ Float
// Diese Variablen werden mit jedem Messzyklus
// zurückgesetzt

    printf("iLastStamp=%d\r\n", iLastStamp); // letzten Messzeitpunkt über die
// Standardausgabe ausgeben

    if(iLastStamp != 0) // Wenn der Inhalt der Variable iLastStamp
// bereits gesetzt wurde, wird der Inhalt der
// Klammern ausgeführt.

        iDayOfWeek = Mdn_DoW(Mdn_GetDate()); // Wochentag in die Variable iDayOfWeek
// speichern

        iDayOfWeek += 1; // +1 damit bei der späteren Division nicht
// durch 0 (Montag) dividiert wird

        printf("iDayOfWeek=%d\r\n", iDayOfWeek); // Den Wochentag über die Standardausgabe
// ausgeben.

        iCurrentStamp = Mdn_GetTime(); // Die aktuelle Zeit auslesen und in die
// Variable iCurrentStamp ablegen.

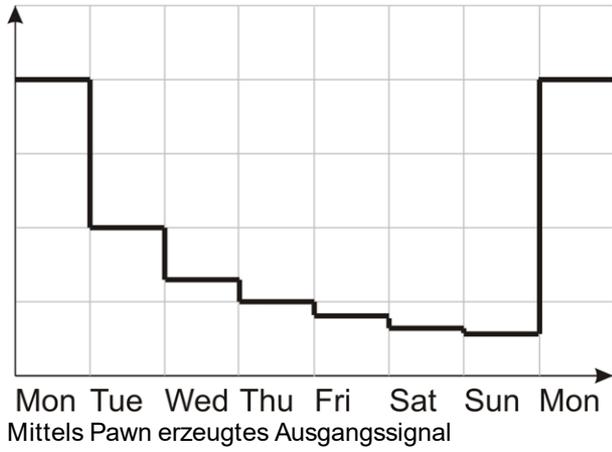
        printf("iCurrentStamp=%d\r\n", iCurrentStamp); // Die aktuelle Zeit über die Standardausgabe
// ausgeben.

        fResult = (iCurrentStamp - iLastStamp) / 6.0; // Zeitdifferenz zwischen aktuellem und
// letztem Messzeitpunkt bilden
// (Messintervall) und durch 6 dividieren.

        printf("fResult=%f\r\n", fResult); // Zeitdifferenz dividiert durch 6 über die
// Standardausgabe ausgeben.

        fResult = fResult / iDayOfWeek; // Ergebnis durch den Wochentag dividieren.
        printf("fResult=%f\r\n", fResult); // Ergebnis über die Standardausgabe ausgeben.

        Mdn_SetCh(MDN_CH_IOUT1, fResult); // Ergebnis auf den Ausgang schreiben.
    }
    iLastStamp = Mdn_GetTime(); // Akt. Zeitpunkt für die Verarbeitung zum
// nächsten Messzeitpunkt zwischenspeichern.
}
```



11.7.3 Durchflussberechnung mit Tabelle

Dieses Beispiel ermittelt den Durchfluss eines Abwasserkanals aufgrund des Füllstandes des Abwasserkanals. An Eingang 1 wird der Füllstand gemessen und an Ausgang 1 soll der Durchfluss ausgegeben werden.

```
static const Table[][Mdn_TablePoint] = // Statische Tabelle für die Umrechnung von
[ // Höhe zu Durchfluss
  [ 0.0, 0.0],
  [ 2.0, 1.0],
  [ 5.0, 3.0],
  [10.0, 10.0],
  [50.0, 15.0]
];

/* Dieser Code wird am Ende jedes
   Messzyklus ausgeführt. */
public Mdn_CtrlFinish()
{
  new Float:fHeight; // Deklaration der Variable fHeight vom
                    // Typ Float
  new Float:fFlow; // Deklaration der Variable fFlow vom
                  // Typ Float
  new Mdn_ValueStatus:iStatus; // Deklaration der Variable iStatus vom
                              // Typ Mdn_ValueStatus
                              // Diese Variablen werden mit jedem Messzyklus
                              // zurückgesetzt

  Mdn_GetCh(MDN_CH_UI1, fHeight, iStatus); // Lese den Kanal 1 aus und speichere den Wert
                                          // in die Variable fHeight und den Status in
                                          // die Variable iStatus.

  if(iStatus == MDN_STATUS_OK) // Wenn der Status des Kanals 1 OK ist, führe
  { // den folgenden Code in den {}-Klammern aus

    new iResult; // Deklaration der Variable iResult vom
                // Typ Integer

    iResult = Mdn_CalcTable( fHeight, fFlow, // Diese Funktion berechnet den Durchfluss
                            Table, sizeof Table); // (fFlow) aufgrund der Höhe (fHeight)
                                                  // und der Tabelle (Table).
                                                  // Sollte sich die Höhe zwischen 2 Einträgen
                                                  // befinden, wird das Ergebnis linear
                                                  // interpoliert. Der Rückgabewert enthält den
                                                  // Fehlerwert der Berechnung

    switch(iResult)
    {
      case MDN_TAB_ERR_FLOOR: // wenn die Höhe kleiner als der erste Eintrag
      { // in der Tabelle ist ->
        fFlow = 0.0; // Setze den Durchfluss auf 0.0.
      }
      case MDN_TAB_ERR_CEIL: // wenn die Höhe größer als der letzte Eintrag
      { // in der Tabelle ist ->
        fFlow = 0.0; // Setze den Durchfluss auf 0.0.
      }
    }
  }

  Mdn_SetCh(MDN_CH_IOUT1, fFlow); // Schreibe den Durchfluss auf den Ausgang 1.
}
```

11.7.4 Script Parsing

Das folgende Beispiel gibt zunächst Geräteklasse, Modem Version und Firmware Version auf der Standardausgabe aus. Als Init Sequenz wird anschließend "INIT COM1\r\n" auf der Com1 ausgegeben. Im Messintervall wird "POLL COM1\r\n" auf der Com1 ausgegeben. Die empfangene Antwort wird zur Kontrolle auf der Standardausgabe ausgegeben. Wenn bei der internen Verarbeitung der Messwerte das Control-Modul und somit die Funktion "Mdn_CtrlFinish()" (siehe Mdn_CtrlFinish()) aufgerufen wird, wird auf der Com1 "OUTPUT COM1\r\n" ausgegeben.

```

public Mdn_CtrlFinish()
{
    Mdn_SerialTx(1, "OUTPUT COM1\r\n", 13);           // String "OUTPUT COM1" über die Com1
}                                                       // ausgeben

COM1Event(event)                                     // Unterfunktion zum Auswerten der Events von
{                                                       // Com1

    if(event == MDN_SERIAL_EVENT_INIT)                // Wenn die Init Sequenz gesendet werden soll
    {
        Mdn_SerialTx(1, "INIT COM1\r\n", 11);        // String "INIT COM1" über die Com1 senden
    }
    else if(event == MDN_SERIAL_EVENT_MEASURE)        // Wenn das Messkommando gesendet werden soll
    {
        Mdn_SerialTx(1, "POLL COM1\r\n", 11);        // String "POLL COM1" über die Com1 senden
    }
    else if(event == MDN_SERIAL_EVENT_TIMEOUT)        // Wenn ein Timeout aufgetreten ist
    {
        Mdn_SerialFinish(1);                          // Datenempfang über Com1 beenden
    }
}

public Mdn_SerialEvent(com, event)                   // Einstiegspunkt für die Auswertung der
{                                                       // seriellen Events

    printf("Mdn_SerialEvent( %d, %d)\r\n",            // Nummer der Com-Schnittstelle und
           com, event);                               // empfangenes Event über die Standardausgabe
}                                                       // ausgeben

    switch(com)                                       // Prüfen, welche Com-Schnittstelle das Event
    {                                                 // ausgelöst hat

        case 1: COM1Event(event);                    // Unterfunktion zum Auswerten der Events von
    }                                                 // Com1 aufrufen
}

COM1Rx(const data{}, len)                             // Unterfunktion zum Auswerten der
{                                                       // empfangenen Daten

    /*-----*/
    /*--- Code der die Daten auswertet --- */
    /*-----*/

    Mdn_SerialFinish(1);                             // Datenempfang über Com1 beenden
}

```

```

public Mdn_SerialRx(com, const data{}, len)           // Einstiegspunkt für die Auswertung der
{                                                     // empfangenen Zeichen

    printf("Mdn_SerialRx( %d, \"%s\", %d)\r\n",       // Nummer der Com-Schnittstelle, empfangene
            com, data, len);                          // Daten und Zeichenanzahl über die
                                                    // Standardausgabe ausgeben

    switch(com)                                       // Prüfen, über welche Com-Schnittstelle die
    {                                                 // Daten empfangen wurden

        case 1: COM1Rx(data, len);                   // Unterfunktion zum Auswerten der über Com1
    }                                                 // empfangenen Daten aufrufen
}

main()
{
    printf( "%04X DEVICE_CLASS\r\n", DEVICE_CLASS); // Geräteklasse über die Standardausgabe
                                                    // ausgeben

    printf( "%04X MODEM_VERSION\r\n", MODEM_VERSION); // Modem Version über die Standardausgabe
                                                    // ausgeben

    printf( "%04X CONTROLLER_VERSION\r\n",          // Firmware Version über die Standardausgabe
            CONTROLLER_VERSION);                    // ausgeben
}

```

11.8 Unterschiede zu C

- Pawn fehlt der Eingabe-Mechanismus von C. Pawn ist eine "integer-only" Variante von C. Es gibt keine Strukturen oder Unions. Floating Point-Unterstützung muss mit benutzerdefinierten Operatoren und der Hilfe von nativen Funktionen implementiert werden.
- Die Syntax für Gleitkommawerte ist strenger als die in C. Werte wie ".5" und "6." sind in C akzeptabel, aber im Pawn muss man "0.5" und "6.0" schreiben. In C ist der Dezimalpunkt optional, wenn ein Exponent enthalten ist, so kann man in C "2E8" schreiben; Pawn akzeptiert den Großbuchstaben "E" nicht. Verwenden Sie den Kleinbuchstaben "e". Es erfordert das Komma: z.B. "2.0e8" (siehe "Numerische Konstanten" auf Seite 180).
- Pawn unterstützt keine "Zeiger". Für die Übergabe von Funktionsparametern als Referenz bietet Pawn ein "Referenz"-Argument (siehe "Funktionsargumente ("call-by-value" versus "call-by-reference")" auf Seite 195). Das "Platzhalter"-Argument ersetzt einige Verwendungen des NULL-Zeigers (siehe "Standardwerte von Funktionsargumenten" auf Seite 197).
- Zahlen können mit Hexadezimal-, Dezimal-, oder Binärbasis angegeben werden. Die Oktale Basis wird nicht unterstützt (siehe "Numerische Konstanten" auf Seite 180). Hexadezimale Zahlen müssen mit "0x" ("x" in Kleinbuchstaben) beginnen. Das Präfix "0X" ist ungültig.
- "Cases" in einem "switch"-Statement sind nicht "durchfallend". Es muss dem "case"-Label zumindest eine Anweisung folgen. Um mehrere Anweisungen auszuführen, müssen Sie ein zusammengesetztes Statement (mit { }) erstellen (siehe "switch (Ausdruck) { case Liste }" auf Seite 193). In C/C++ ist die "switch"-Anweisung ein "bedingtes goto". In Pawn ist die "switch"-Anweisung ein strukturiertes "if".
- Eine "break"-Anweisung beendet nur Schleifen. In C/C++ beendet die "break"-Anweisung auch ein "case" in einer "switch"-Anweisung.

- Pawn unterstützt "array Zuweisungen", mit der Limitation, dass beide Arrays die gleiche Länge haben müssen. Zum Beispiel, wenn "a" und "b" Arrays mit 6 Zeilen sind, dann ist der Ausdruck "a=b" gültig. Neben Zeichenketten, unterstützt Pawn auch literale Arrays und somit Ausdrücke wie "a = {0,1,2,3,4,5}", wobei "a" eine Array Variable mit 6 Elementen ist.
- "defined" ist ein Operator und keine Präprozessor-Direktive. Der "defined" Operator in Pawn arbeitet mit Konstanten (deklariert mit "const"), globalen Variablen, lokalen Variablen und Funktionen.
- Der "sizeof"-Operator gibt die Größe von Variablen in "Elementen" zurück und nicht in "Bytes". Ein Element ist ein Eintrag oder ein Sub-Array. Weitere Details finden Sie im Kapitel "Sonstiges" auf Seite 187.
- Eine leere Anweisung ist ein leerer Block (mit {}), nicht ein Semikolon (siehe "Zusammengesetzte Anweisungen" auf Seite 189). Diese Änderung verhindert häufige Fehler.
- Eine Division erfolgt in der Weise, dass der Rest der Division das gleiche Vorzeichen hat (oder hätte) wie der Nenner. Bei der Division (Operator "/") erfolgt die Rundung immer zum kleineren ganzzahligen Wert (wobei -2 kleiner ist als -1). D.h. $5/2=2$ (2,5 wird zu 2 abgerundet), $-5/2=-3$ (-2,5 wird zu -3 abgerundet). Der "%" -Operator ergibt immer ein positives Ergebnis unabhängig vom Vorzeichen des Zählers (siehe "Operatoren und Ausdrücke" auf Seite 184).
- Es gibt keinen unären Operator "+", da dieser sowieso ein "no-operation"-Operator ist ("a = +1" ist nicht gültig; korrekt: "a = 1").
- Drei der bitweisen Operatoren haben andere Prioritäten als in C. Die Prioritätsstufe des "&", "^" und "|" Operators ist höher als die relationalen Operatoren. Dennis Ritchie erklärte, dass diese Operatoren in C ihre niedrigen Prioritätsstufen bekamen, weil frühe C-Compiler noch nicht über die logischen Operatoren "&&" und "||" verfügten, so dass stattdessen bitweise "&" und "|" verwendet wurden.
- Das Schlüsselwort "const" in Pawn implementiert die "enum" Funktionalität von C.
- In den meisten Fällen sind Vorwärts-Deklarationen von Funktionen (d.h. Prototypen) nicht notwendig. Pawn ist ein 2-Pass-Compiler. Er erkennt alle Funktionen beim ersten Durchlauf und verwendet diese beim zweiten Durchlauf. Benutzerdefinierte Operatoren müssen jedoch vor der Benutzung deklariert werden. Falls vorhanden, müssen Vorwärts-Deklarationen genau mit der Definition der Funktion übereinstimmen. Die Parameternamen in den Prototypen und den Definitionen der Funktionen müssen ident sein. Pawn kümmert sich um Parameter-Namen im Prototyp auf Grund der "benannte Parameter"-Funktion. Pawn verwendet Prototypen, um vorwärts deklarierte Funktionen aufzurufen. Um diese dabei mit benannten Parametern zu verwenden, muss der Compiler bereits die Namen der Parameter (und ihre Position in der Parameterliste) kennen. Aus diesem Grund müssen die Parameternamen in den Prototypen mit jenen in den Definitionen übereinstimmen.

Kapitel 12 API

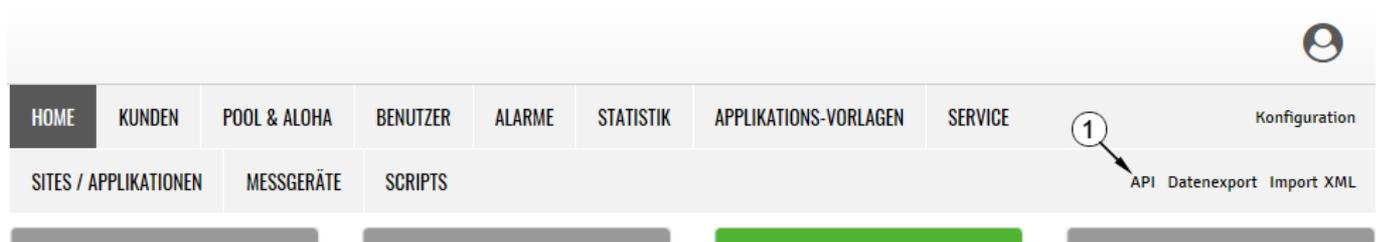
Wichtiger Hinweis: Für die Verwendung der API (Application Programming Interface) sind die entsprechenden Lizenzen am myDatanet-Server erforderlich. Für nähere Informationen wenden Sie sich an Ihren zuständigen Vertriebspartner.

12.1 Allgemein

Die API dient dazu, Daten aus dem myDatanet-Server zu exportieren sowie Daten in den myDatanet-Server zu importieren. Dies beschränkt sich für Geräte des Typs "myDatalogMUC xG/4G" jedoch auf die Messdaten, die berechneten Kanäle, die Stellwerte für die Ausgabekanäle, die Stellwerte für die Interface Ausgabekanäle, das Übertragungsintervall und das Aufzeichnungsintervall.

12.2 rapidM2M Playground

Der rapidM2M Playground ermöglicht es Ihnen, sich mit der API des myDatanet-Servers vertraut zu machen und die bereitgestellten Funktionen zu testen. Durch einen Klick auf die Schaltfläche "API" gelangen Sie zum rapidM2M Playground .



1 öffnet den rapidM2M Playground

12.2.1 Übersicht

The screenshot shows the rapidM2M Playground interface. At the top, there are input fields for 'Username' (1) and 'Password' (2). Below them is a list of HTTP methods (3) and a dropdown for the resource path (4). A 'GET' button (5) is used to execute the request. The response shows a '200' status code (11) and a JSON body (14) with a 'Copy' button (12). A documentation window (13) is open, showing details for the 'customer_id' parameter. At the bottom, a list of recent requests (15) shows the executed 'GET /1/customers/! training'.

rapidM2M Playground

1	Eingabefeld für den Benutzernamen
2	Eingabefeld für das Passwort
3	Auflistung der zur Verfügung stehenden HTTP-Kommandos. Die HTTP-Kommandos sind entsprechend ihrer Anwendungsgebiete gruppiert.
4	Abhängig vom gewählten HTTP-Kommando werden hier die Dropdown-Listen für die Auswahl des Kunden, des Benutzers und der Messstelle eingeblendet, die die entsprechende Wildcard ("\${CID}"...Kunde , "\${UID}"...Benutzer, "\${SID}"...Messstelle) im Ressource-Pfad des HTTP-Kommandos ersetzen sollen.
5	Button zum Ausführen des HTTP-Kommandos
6	öffnet die Webseite "http://rapidm2m.com/", die zusätzliche Informationen für Entwickler enthält
7	öffnet den login-Dialog des mit dem rapidM2M Playground verknüpften myDatatnet-Servers
8	öffnet die Kurzanleitung für die API
9	Button zum Wechsel des Farbschemas des rapidM2M Playground
10	Fenster, in dem das gewählte HTTP-Kommando angezeigt wird
11	Response-Code, der vom myDatatnet-Server als Antwort auf das HTTP-Kommando gesendet wurde
12	kopiert das JSON-Objekt, das als Antwort auf das HTTP-Kommando erzeugt wurde, in die Zwischenablage
13	Fenster, in dem die Dokumentation für das ausgewählte HTTP-Kommando angezeigt wird. Diese enthält abhängig vom ausgewählten Kommando eine Beschreibung der Aktion, die durchgeführt wird, Hinweise, die beachtet werden müssen und eine Beschreibung des Request Bodys sowie des Response Bodys.
14	Fenster, in dem das JSON-Objekt angezeigt wird, das als Antwort auf das HTTP-Kommando erzeugt wird
15	Fenster, in dem die zuletzt ausgeführten HTTP-Kommandos angezeigt werden

Kapitel 13 Wartung

Wichtiger Hinweis: Um Schäden am Instrument zu vermeiden, dürfen die in diesem Abschnitt der Anleitung beschriebenen Arbeiten nur von qualifiziertem Personal ausgeführt werden.

Vor Wartungs-, Reinigungs- und/oder Reparaturarbeiten ist das Gerät unbedingt spannungsfrei zu machen.

13.1 Allgemeine Wartung

- Überprüfen Sie das myDatalogMUC xG/4G regelmäßig auf mechanische Beschädigungen.
- Überprüfen Sie regelmäßig alle Kabel auf mechanische Beschädigungen.
- Reinigen Sie das myDatalogMUC xG/4G mit einem weichen, feuchten Tuch. Verwenden Sie ein mildes Reinigungsmittel, falls nötig.

13.2 Sicherungswechsel



GEFAHR:

Feuergefahr. Eine falsche Sicherung kann Verletzungen, Schäden oder Immissionen verursachen. Die Sicherung befindet sich im Inneren des Gehäuses, das nur vom Hersteller geöffnet werden darf.

Sollte der Verdacht bestehen, dass die Sicherung des myDatalogMUC xG/4G defekt ist (siehe "Fehlersuche und Behebung" auf Seite 213), muss das Gerät in der Originalverpackung an den Hersteller zurückgesendet werden (siehe "Rücksendung" auf Seite 48).

Kapitel 14 Demontage/Entsorgung

Durch falsche Entsorgung können Gefahren für die Umwelt entstehen.

Entsorgen Sie Gerätekomponenten und Verpackungsmaterialien entsprechend den gültigen örtlichen Umweltvorschriften für Elektroprodukte.

1. Trennen Sie die eventuell verwendete Ladespannung.
2. Lösen Sie eventuell angeschlossene Kabel mit geeignetem Werkzeug.



Logo zur WEEE-Direktive der EU

Dieses Symbol weist darauf hin, dass bei der Verschrottung des Gerätes die Anforderungen der Richtlinie 2012/19/EU über Elektro- und Elektronik-Altgeräte zu beachten sind. Die Microtronics Engineering GmbH unterstützt und fördert das Recycling bzw. die umweltgerechte, getrennte Sammlung/Entsorgung von Elektro- und Elektronik-Altgeräten zum Schutz der Umwelt und der menschlichen Gesundheit. Beachten Sie die örtlichen Entsorgungsvorschriften und Gesetze.

Die Microtronics Engineering GmbH entpflichtet in Österreich in den Verkehr gebrachte Waren über die ERA, daher können in Österreich Sammelstellen, welche mit der ERA Elektro Recycling Austria GmbH (<https://www.era-gmbh.at/>) kooperieren, für die Entsorgung genutzt werden.

Das Gerät enthält eine Batterie bzw. einen Akku (Lithium), welcher separat zu entsorgen ist.

Kapitel 15 Fehlersuche und Behebung

15.1 Allgemeine Probleme

Problem	Ursache/Lösung
Das Gerät zeigt keine Reaktion (Status-LED immer aus).	<ul style="list-style-type: none"> • Kabelverbindungen überprüfen (siehe "Anschluss der Sensoren, der Aktoren und der Versorgung" auf Seite 52)
Kommunikationsprobleme	<ul style="list-style-type: none"> • Werten Sie den Blinkcode der Status-LED aus (siehe "Status-LED" auf Seite 72). • Laden Sie das Gerätelogs vom myDatanet-Server und benutzen Sie DeviceConfig für die Auswertung (siehe "Auswerten des Gerätelogs" auf Seite 224). • Laden Sie das Gerätelogs mit Hilfe des DeviceConfig vom myDatalogMUC xG/4G (siehe "Benutzerhandbuch für DeviceConfig" 206.887). Eine Auflistung aller möglichen Fehlercodes finden Sie im Kapitel "Log-Einträge und Fehlercodes" (siehe "Log-Einträge und Fehlercodes" auf Seite 216).
Das Melden eines Versorgungsspannungsausfalls funktioniert nicht.	<ul style="list-style-type: none"> • Der Alarm für den internen Messwert "Spannung" wurde nicht konfiguriert (siehe "Technische Details zum integrierten Pufferakku" auf Seite 63).
Es sind nicht alle/keine Daten am Server vorhanden.	<ul style="list-style-type: none"> • Es kam zu einem Verbindungsabbruch während der Übertragung, erkennbar an einem Timeout-Eintrag in der Verbindungsliste (siehe "Benutzerhandbuch für myDatanet-Server" 206.886). Lösung: Aloha-Übertragung auslösen oder auf die nächste zyklische Übertragung warten. • Die Zuweisung von Gerät und Messstelle ist nicht korrekt (siehe "Site" auf Seite 75).
Die Daten am Universaleingang sind nicht plausibel.	<ul style="list-style-type: none"> • Kabelverbindungen überprüfen (siehe "Anschluss der Sensoren, der Aktoren und der Versorgung" auf Seite 52) • Prüfen Sie, ob das Ausgabesignal des von Ihnen verwendeten Sensors kompatibel mit den elektrischen Kenndaten der Universaleingänge ist (siehe "Technische Details zu den Universaleingängen" auf Seite 57). • Prüfen Sie, ob die Konfiguration des Universaleingangs zum Ausgabesignal des Sensors passt (siehe "Messkanäle" auf Seite 87). • Die Filtereinstellungen des Universaleingangs überprüfen (siehe "Messkanäle" auf Seite 87). • Prüfen Sie, ob die ext. Aufwärmzeit passend zu dem von Ihnen verwendeten Sensor gewählt wurde (siehe "Ausgabekanäle" auf Seite 125).

Problem	Ursache/Lösung
Die Daten der Interface Kanäle sind nicht plausibel (Modbus, RS485).	<ul style="list-style-type: none"> • Kabelverbindungen überprüfen (siehe "Anschluss der Sensoren, der Aktoren und der Versorgung" auf Seite 52) • Prüfen Sie, ob bei der Konfiguration der Interface Kanäle die Auswahl der Modbus-Adresse und die Slave-Adresse (nur im Modus "Modbus-Master) korrekt sind (siehe "Interface Kanäle 1-32" auf Seite 107). • Überprüfen Sie die Konfiguration der Modbus-Schnittstelle (siehe "Schnittstellen" auf Seite 76). • Überprüfen Sie die Einstellung der Abschluss- bzw. Klemmwidestände (siehe "Technische Details zu den Modbus-Schnittstellen (Com1, Com3)" auf Seite 58).
Die Daten der Interface Kanäle sind nicht plausibel (seriell, RS232).	<ul style="list-style-type: none"> • Kabelverbindungen überprüfen (siehe "Anschluss der Sensoren, der Aktoren und der Versorgung" auf Seite 52) • Prüfen, ob bei der Konfiguration der Interface Kanäle die Spaltenauswahl korrekt ist (siehe "Interface Kanäle 1-32" auf Seite 107). • Überprüfen Sie die Konfiguration der seriellen Schnittstelle (siehe "Schnittstellen" auf Seite 76). Es könnten das Zahlenformat oder das/die Zahlentrennzeichen falsch konfiguriert sein. • Sind weniger Spalten in der Datenmessage vorhanden als bei der Konfiguration der Interface Kanäle (siehe "Interface Kanäle 1-32" auf Seite 107) angegeben, wird das als ungültiger Frame erkannt, die Daten werden nicht aufgezeichnet und ein entsprechender Eintrag im Gerätelogs erzeugt (siehe "Fehlercodes der RS232-Schnittstelle " auf Seite 61).
Der Alarmzustand eines Messkanals wurde nicht erkannt.	<ul style="list-style-type: none"> • Messintervall erhöhen
Der Alarmzustand wurde nicht übertragen, obwohl die Daten vorhanden sind.	<ul style="list-style-type: none"> • Alarmeinstellungen des Messkanals überprüfen • Es kam zu einem Verbindungsabbruch während der Übertragung, erkennbar an einem Timeout-Eintrag in der Verbindungsliste (siehe "Benutzerhandbuch für myDatanet-Server " 206.886). Lösung: Aloha-Übertragung auslösen oder auf die nächste zyklische Übertragung warten.
Die Alarmnachricht wurde nicht zugestellt, obwohl der Alarm signalisiert wurde.	<ul style="list-style-type: none"> • Einstellungen des Alarmrufplans prüfen (siehe "Benutzerhandbuch für myDatanet-Server " 206.886). • Adressdaten des Alarmrufplans prüfen (siehe "Benutzerhandbuch für myDatanet-Server " 206.886).
Die Relais funktionieren nicht.	<ul style="list-style-type: none"> • Ausfall der Spannung, die über die Relais geführt wird • Die Messstelleneinstellungen wurden von einer Server-PLC überschrieben.

Problem	Ursache/Lösung
Am Ausgabekanal wird nicht der über die Eingabemaske am myDatanet-Server festgelegte Wert ausgegeben.	<ul style="list-style-type: none"> • Die Messstelleneinstellungen wurden von einer Server-PLC überschrieben. • Prüfen Sie, ob mittels Device Logic auf den Ausgabekanal geschrieben wird. Ein von einer Device Logic ermittelter Wert überschreibt den über die Eingabemaske am myDatanet-Server festgelegten Wert.
Die Instruction List wird nicht korrekt ausgeführt.	<ul style="list-style-type: none"> • Prüfen Sie, ob bei der Konfiguration der Steuerung der korrekte Device Logic Typ ausgewählt wurde (siehe "Steuerung" auf Seite 75). • Laden Sie das Gerätelog vom myDatanet-Server und benutzen Sie DeviceConfig für die Auswertung (siehe "Auswerten des Gerätelogs" auf Seite 224).
Die Pawn Device Logic wird nicht korrekt ausgeführt.	<ul style="list-style-type: none"> • Prüfen Sie, ob bei der Konfiguration der Steuerung der korrekte Device Logic Typ ausgewählt wurde (siehe "Steuerung" auf Seite 75). • Laden Sie das Gerätelog vom myDatanet-Server und benutzen Sie DeviceConfig für die Auswertung (siehe "Auswerten des Gerätelogs" auf Seite 224). Eine Auflistung aller möglichen Fehlercodes der Device Logic finden Sie im Kapitel "Device Logic Fehlercodes" auf Seite 178. • Laden Sie das Gerätelog mit Hilfe des DeviceConfig vom myDatalogMUC xG/4G (siehe "Benutzerhandbuch für DeviceConfig" 206.887). Eine Auflistung aller möglichen Fehlercodes der Device Logic finden Sie im Kapitel "Device Logic Fehlercodes" auf Seite 178.

15.2 Log-Einträge und Fehlercodes

Log-Eintrag		Parameter		Beschreibung
Code	Klartext	Code	Klartext	
1000	POWER ON	0	---	Systemstart abgeschlossen
		> 0	---	Interner System-Fehler Neustart aufgrund eines internen System-Fehlers. Sollte der "POWER ON" Log-Eintrag mehrmals mit einem Parameter-Code > 0 im Geräte-log enthalten sein, liegt unter Umständen ein Hardwareproblem vor. Kontaktieren Sie in diesem Fall den Hersteller (siehe "Kontaktinformationen" auf Seite 231).
1020	ERROR SOD	##	---	Interner System-Fehler Sollte dieser Fehler mehrmals mit demselben Parameter-Code im Geräte-log enthalten sein, liegt unter Umständen ein Hardwareproblem vor. Kontaktieren Sie in diesem Fall den Hersteller (siehe "Kontaktinformationen" auf Seite 231).
1030	UV LOCKOUT	---	---	Das Gerät schaltet aufgrund einer zu niedrigen Akku-/Batteriespannung in den Energiesparmodus und stellt alle Operationen ein. Nur die Laderegulierung, falls vorhanden, bleibt aktiv.
1031	UV RECOVER	---	---	Die Akku-/Batteriespannung reicht wieder aus, um einen sicheren Betrieb zu gewährleisten. Das Gerät nimmt nun den normalen Betrieb entsprechend der Konfiguration wieder auf.
1033	MODEM_UPDATE	##	---	Update der Firmware des Modemcontrollers wurde erfolgreich durchgeführt. Dieser Eintrag ist immer doppelt im Geräte-log enthalten. Beim ersten Eintrag gibt der Parameter die Hauptversionsnummer (z.B. 3 bei 03v011) und beim zweiten Eintrag die Nebenversionsnummer (z.B. 11 bei 03v011) an.
1034	CONTROLLER_UPDATE	##	---	Update der Firmware des Messcontrollers wurde erfolgreich durchgeführt. Dieser Eintrag ist immer doppelt im Geräte-log enthalten. Beim ersten Eintrag gibt der Parameter die Hauptversionsnummer (z.B. 3 bei 03v011) und beim zweiten Eintrag die Nebenversionsnummer (z.B. 11 bei 03v011) an.

Log-Eintrag		Parameter		Beschreibung
Code	Klartext	Code	Klartext	
1111	ALOHA STOP	0	---	Ende des Aloha-Übertragungsmodus
		##	---	Der Aloha-Übertragungsmodus wurde unerwartet beendet. Der Parameter gibt an wie viele Sekunden der Aloha-Übertragungsmodus noch andauern hätte sollen.
1114	BACKUP SUPPLY	1	---	Die Versorgung wurde auf den Pufferakku umgeschaltet.
		0	---	Die Versorgung wurde wieder auf V IN umgeschaltet.
1161	LOG REFORMATFILE	---	---	Das Filesystem wurde neu formatiert. D.h. alle bisher aufgezeichneten Daten sowie Log-Einträge wurden gelöscht.
1200	MODEM ERROR			Modemfehler (siehe "Modemfehler" auf Seite 221)
1202	MODEM CMME ERROR	##	---	Das GPRS-Modem meldet einen +CME-Fehler. Der Parameter gibt an, um welchen Fehler es sich handelt.
1203	SELECTED NETWORK	##	---	Ein neues GSM-Netzwerk wurde gewählt. Dieser Eintrag ist immer doppelt im Gerätelog enthalten. Beim ersten Eintrag gibt der Parameter den MCC (Mobile Country Code) und beim zweiten Eintrag den MNC (Mobile Network Code) des gewählten GSM-Netzwerks an.
1204	BANNED NETWORK	##		Ein GSM-Netzwerk wurde auf die Banlist gesetzt. Dieser Eintrag ist immer doppelt im Gerätelog enthalten. Beim ersten Eintrag gibt der Parameter den MCC (Mobile Country Code) und beim zweiten Eintrag den MNC (Mobile Network Code) des gewählten GSM-Netzwerks an.
1205	BAN LIST CLEARED	---	---	Die Banlist wurde gelöscht, da sich keine GSM-Netze mehr im Empfangsbereich befinden, die nicht auf die Banlist gesetzt wurden.
1212	ERROR MODEM IRREGULAR OFF	##	---	Zeigt eine fehlerhafte Verbindung an. Der Parameter enthält dabei einen Zähler, der angibt wie viele Verbindungen hintereinander nicht funktioniert haben.
1224	MODEM SIM RESET	---	---	Das GRPS-Modem wurde neu gestartet, um den SIM-Chip neu zu initialisieren.

Log-Eintrag		Parameter		Beschreibung
Code	Klartext	Code	Klartext	
1252	MODEM TO CON	##	---	Timeout während des Verbindungsaufbaus. Der Parameter enthält dabei einen Zähler, der angibt wie viele Verbindungsaufbauversuche hintereinander nicht funktioniert haben.
1281	ZLIB STREAMPROCESS ERR	##	---	interner Fehler Sollte dieser Fehler mehrmals im Geräte-log enthalten sein, kontaktieren Sie den Hersteller (siehe "Kontaktinformationen" auf Seite 231).
1282	ZLIB STREAMFINISH ERR	##	---	interner Fehler Sollte dieser Fehler mehrmals im Geräte-log enthalten sein, kontaktieren Sie den Hersteller (siehe "Kontaktinformationen" auf Seite 231).
1440	DECAY MEM ERR	---	---	Die Anzahl der Messwerte, die bei der Dämpfung berücksichtigt werden, ist zu groß (siehe "Messkanäle" auf Seite 87). Reduzieren Sie entweder das Zeitfenster für die Dämpfung oder das Messintervall.

Log-Eintrag		Parameter		Beschreibung
Code	Klartext	Code	Klartext	
1600	STATE	1	OK	letzte Verbindung o.k.
		2	TRANSMISSION ERROR	letzte Übertragung fehlerhaft später erneut versuchen
		3	MEASUREMENT ERROR	letzte Messung fehlerhaft <ul style="list-style-type: none"> Überprüfen Sie die Konfiguration. Überprüfen Sie die Verbindung zwischen Gerät und Sensor.
		4	HOLD	Standby (GPRS an, Messung aus) Wird der Aloha-Übertragungsmodus am Gerät ausgelöst, schaltet der myDatalogMUC xG/4G wieder in den Modus "RUN" (GPRS an, Messung an).
		5	TRANSPORT	Transportsperre (GPRS aus, Messung aus) siehe "Standby"
		6	OFFLINE	Offline (GPRS aus, Messung an) siehe "Standby"
		7	NO NET (NET LOCK)	Netzsperrung/kein passender Provider <ul style="list-style-type: none"> Versuchen Sie, die Antennenposition zu verbessern. Überprüfen Sie, ob sich das Gerät im Versorgungsbereich befindet (www.microtronics.com/footprint).
		8	NO NET	kein GSM Netzwerk <ul style="list-style-type: none"> Antennenpositionierung verbessern später erneut versuchen
		10	GPRS ERROR	keine GPRS Verbindung Versuchen Sie, die Antennenposition zu verbessern.
		11	IP ERROR	kein myDatanet-Server erreichbar <ul style="list-style-type: none"> Überprüfen Sie, ob am myDatanet-Server der Port 51241 frei geschaltet ist. später erneut versuchen
		12	SIM CARD ERROR	fehlerhafter SIM-Chip Kontaktieren Sie den Support.

Log-Eintrag		Parameter		Beschreibung
Code	Klartext	Code	Klartext	
2000 - 2999	MODULE ERR			Modul-spezifischer Fehler (siehe "Fehlercodes der RS232-Schnittstelle " auf Seite 61) bzw. Bereich für die kundenspezifischen Fehlercodes, die mittels der Pawn Script Funktion "Mdn_WriteLog()" in das Gerätelog geschrieben werden können (siehe "Verschiedene Funktionen" auf Seite 172).
3000 - 3099				Fehlercodes der Scriptausführung. Abhängig vom ausgewählten Script Typ gelten folgende Fehlerbeschreibungen: <ul style="list-style-type: none"> • "Device Logic Fehlercodes" auf Seite 178

15.2.1 Modemfehler

Log-Eintrag		Parameter		Beschreibung
Code	Klartext	Code	Klartext	
GPRS-Fehler				
1200	BEARER LINE BUSY	-993	---	Leitung belegt später erneut versuchen
	BEARER NO ANSWER	-992	---	Keine Antwort <ul style="list-style-type: none"> • später erneut versuchen • Überprüfen Sie, ob sich das Gerät im Versorgungsbereich befindet (www.microtronics.com/footprint).
	BEARER NO CARRIER	-991	---	Kein Träger <ul style="list-style-type: none"> • später erneut versuchen • Überprüfen Sie, ob sich das Gerät im Versorgungsbereich befindet (www.microtronics.com/footprint).
	BEARER GPRS FAILED	-988	---	GPRS Setup-Fehler <ul style="list-style-type: none"> • Versuchen Sie, die Antennenposition zu verbessern. • Überprüfen Sie, ob sich das Gerät im Versorgungsbereich befindet (www.microtronics.com/footprint).
	BEARER PPP LCP FAILED	-987	---	Fehler beim Aushandeln des LCP. interner Fehler
	BEARER PPP AUTH FAILED	-986	---	Fehler bei der PPP Authentifizierung. interner Fehler
	BEARER PPP IPCP FAILED	-985	---	Fehler beim Aushandeln des IPCP. interner Fehler
	BEARER PPP LINK FAILED	-984	---	PPP Gegenstelle reagiert nicht auf Echo-Request. interner Fehler
	BEARER PPP TERM REQ	-983	---	PPP Session wurde von der Gegenstelle beendet. interner Fehler
	BEARER CALL REFUSED	-982	---	eingehender Anruf abgelehnt später erneut versuchen
	BEARER UNKNOWN	-981	---	später erneut versuchen

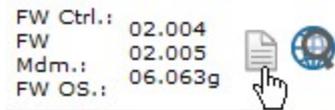
Log-Eintrag		Parameter		Beschreibung
Code	Klartext	Code	Klartext	
Fehler beim Aufbauen der GPRS-Verbindung				
1200	BEARER BAD HDL	-980	---	invalid handle interner Fehler
	BEARER BAD STATE	-979	---	Der Träger wurde nicht gestoppt. interner Fehler
	BEARER DEV	-978	---	Fehler bei der Link-Layer Initialisierung interner Fehler
	BEARER GPRS START UNKNOWN	-977	---	interner Fehler
Fehler beim Abbauen der GPRS-Verbindung				
1200	BEARER GPRS STOP BAD HDL	-976	---	invalid handle interner Fehler
	BEARER GPRS STOP UNKNOWN	-975	---	interner Fehler
1200	NETLOCK FAILURE	-970	---	Fehler bei der Netzauswahl <ul style="list-style-type: none"> Überprüfen Sie, ob sich das Gerät im Versorgungsbereich befindet (www.microtronics.com/footprint).
1200	BAND SEL FAILED	-969	---	Es konnte weder auf dem GSM900/1800 noch auf dem GSM850/1900 Band ein Netzwerk gefunden werden. <ul style="list-style-type: none"> Versuchen Sie, die Antennenposition zu verbessern. Überprüfen Sie, ob sich das Gerät im Versorgungsbereich befindet (www.microtronics.com/footprint).
TCP Channel Fehler (1/2)				
1200	CHANNEL ABORTED	-965	---	Es wird versucht auf einen/von einem nicht mehr verfügbaren TCP-Client zu schreiben/lesen. später erneut versuchen
	CHANNEL CSTATE	-964	---	Der Channel ist nicht im "WIP_CSTATE_READY" Zustand. später erneut versuchen

Log-Eintrag		Parameter		Beschreibung
Code	Klartext	Code	Klartext	
TCP Channel Fehler (2/2)				
1200	CHANNEL NOT SUPPORTED	-963	---	Die Option wird vom Channel nicht unterstützt. später erneut versuchen
	CHANNEL OUT OF RANGE	-962	---	Der Options-Wert liegt außerhalb des zulässigen Bereichs. später erneut versuchen
	CHANNEL MEMORY	-961	---	interner Fehler
	CHANNEL INTERNAL	-960	---	interner Fehler
	CHANNEL INVALID	-959	---	ungültige Option oder Parameter interner Fehler
	TCP DNS FAILURE	-958	---	Der Name konnte nicht in eine IP-Adresse aufgelöst werden. interner Fehler
	CHANNEL RESOURCES	-957	---	Es stehen keine weiteren TCP-Puffer mehr zur Verfügung. später erneut versuchen
	CHANNEL PORT IN USE	-956	---	TCP-Server-Port wird bereits verwendet. später erneut versuchen
	CHANNEL REFUSED	-955	---	Die TCP-Verbindung wurde vom Server abgelehnt. später erneut versuchen
	CHANNEL HOST UNREACHABLE	-954	---	keine Route zum Host später erneut versuchen
	CHANNEL NETWORK UNREACHABLE	-953	---	kein Netz erreichbar später erneut versuchen
	CHANNEL PIPE BROKEN	-952	---	TCP-Verbindung unterbrochen später erneut versuchen
	CHANNEL TIMEOUT	-951	---	Timeout (DNS-Request, TCP-Verbindung, Ping-Response,...) später erneut versuchen
	CHANNEL UNKNOWN	-950	---	interner Fehler

15.3 Auswerten des Gerätelogs

15.3.1 Auswerten des Gerätelogs am myDatanet-Server

Am myDatanet-Server sind die letzten 300 Log-Einträge über den unten abgebildeten Button, der sich in der Messgeräteleiste befindet, abrufbar. Da die Log-Einträge genau wie die Messdaten im Übertragungsintervall zum Server gesendet werden, sind immer nur die Log-Einträge bis zur letzten Serververbindung verfügbar.



Eine genauere Beschreibung zur Auswertung des Gerätelogs am myDatanet-Server finden Sie im Handbuch des Servers ("Benutzerhandbuch für myDatanet-Server " 206.886).

15.3.2 Auswerten des Gerätelogs mittels DeviceConfig

Mit Hilfe des Programms DeviceConfig können alle gespeicherten Logeinträge, auch jene, die noch nicht zum myDatanet-Server übertragen wurden, direkt über die USB-Schnittstelle oder die Bluetooth-Schnittstelle aus des myDatalogMUC xG/4G gelesen werden.

Eine genauere Beschreibung zur Auswertung des Gerätelogs mittels DeviceConfig finden Sie im Handbuch zum DeviceConfig ("Benutzerhandbuch für DeviceConfig" 206.887).

Kapitel 16 Ersatzteile und Zubehör

16.1 Antennen

Beschreibung	Menge	Bestellnummer
Antennenverlängerung SMA-M/SMA-F 2,5m	1	206.807
Winkeladapter SMA-M/SMA-F	1	300318
Flachantenne Smart Disc Multi Band 2xSMA-M 2m	1	301090

16.2 Versorgung

Beschreibung	Menge	Bestellnummer
Netzgerät 24V 2,5A für Hutschienenmontage	1	206.667

16.3 Adapter

Beschreibung	Menge	Bestellnummer
Gender changer 9pol. D-Sub male/male	1	206.684
Nullmodemadapter 9pol. D-Sub female/male ¹⁾	1	206.686

¹⁾ Bitte beachten Sie, dass der Pin 9 des Adapters nicht durchgeschliffen wird.

16.4 Sonstiges Zubehör

Beschreibung	Menge	Bestellnummer
myDatanet Tool Pen	1	206.646

Kapitel 17 Dokumentenhistorie

Rev.	Datum	Änderungen
01	12.10.2020	Erste Version

Kapitel 18 Glossar

Aloha

Verbindungsmodus, der speziell für die Inbetriebnahme gedacht ist. Das Gerät hält für eine konfigurierbare Zeit eine Verbindung zum Server aufrecht und führt alle 3sec. eine Messung durch. Dabei werden nur die internen Messwerte (GSM-Feldstärke, Spannungen, ...) und die Werte der Universaleingänge (falls vorhanden) erzeugt. Die ermittelten Messwerte werden nicht gespeichert, sondern nur zur Anzeige an den Server gesendet.

Footprint

Die Geräte des Herstellers sind ab Werk mit Subscriber Identity Modules (SIM) zur mobilen Übertragung der Daten ausgestattet. Der Footprint bezeichnet jene Länder und Regionen, in denen eine Mobilfunkverbindung zur Verfügung steht(siehe www.microtronics.com/footprint).

NaN-Wert

Beim myDatenet werden spezielle Kodierungen verwendet, um verschiedene Fehlerzustände in z.B. den Messwerten anzuzeigen. Durch das Setzen eines Messwerts auf "NaN" wird dieser eindeutig als ungültig gekennzeichnet und somit nicht mehr für weitere Berechnungen verwendet. In den Messwertgrafiken wird ein auf "NaN" gesetzter Messwert durch eine Unterbrechung in der Ganglinie angezeigt. Beim Download der Daten wird ein auf "NaN" gesetzter Messwert durch ein leeres Datenfeld signalisiert.

Kapitel 19 Kontaktinformationen

Support & Service:

Microtronics Engineering GmbH
Hauptstrasse 7
3244 Ruprechtshofen
Austria, Europe
Tel. +43 (0)2756 7718023
support@microtronics.com
www.microtronics.com

Microtronics Engineering GmbH (Headquarters)

Hauptstrasse 7
3244 Ruprechtshofen
Austria, Europe
Tel. +43 (0)2756 77180
Fax. +43 (0)2756 7718033
office@microtronics.com
www.microtronics.com



WE LIVE M2M

Zertifiziert durch TÜV AUSTRIA: EN ISO 9001:2015, EN ISO 14001:2015, EN ISO 50001:2011 für myDatenet | TÜV SÜD: ATEX Richtlinie 2014/34/EU

© Microtronics Engineering GmbH. All rights reserved. Photos: Microtronics