

**UMB-Protokoll 1.0**  
**Universal-Measurement-Bus**  
Kommunikationsprotokoll  
für meteorologische Sensoren



Stand 03.05.2021  
Protokoll-Version 1.0  
Dokument-Version 1.7



# Inhaltsverzeichnis

<b>1</b>	<b>Versionsgeschichte .....</b>	<b>4</b>
<b>2</b>	<b>Vorbemerkung.....</b>	<b>6</b>
2.1	Einschränkung der Garantie.....	6
2.2	Nomenklatur.....	6
2.3	Datenformat und Byteorder im Kommunikationsprotokoll: .....	6
2.4	Physikalische Anbindung und Hardwarestruktur.....	6
2.5	Softwareprotokoll .....	6
<b>3</b>	<b>UMB-Binär-Protokoll (Version 1.0).....</b>	<b>7</b>
3.1	Protokoll-Stack (Framing).....	7
3.1.1	(7) Application Layer .....	7
3.1.2	(6) Presentation Layer.....	7
3.1.3	(5) Session Layer .....	7
3.1.4	(4) Transport Layer .....	7
3.1.5	(3) Network Layer.....	7
3.1.6	(2b) Data-Link Layer / Logical Link Control.....	8
3.1.7	(2a) Data-Link Layer / Media Access Control .....	8
3.1.8	(1) Physical Layer .....	9
3.1.9	Zusammenfassung.....	9
3.2	Topologie .....	10
3.3	Klassen- und Geräte-ID.....	11
3.3.1	Beispiele für die Bildung von Adressen .....	11
3.4	CRC-Prüfsumme.....	12
3.5	Datentypen.....	12
3.6	Messwerttypen .....	12
3.7	Status- und Errorcodes .....	13
3.8	Kommandos (Datagramme) .....	15
3.8.1	Übersicht der Kommandos.....	15
3.8.2	Hard- und Softwareversion (20h).....	16
3.8.3	Geräteinformationen (2Dh).....	16
3.8.4	EEPROM auslesen (21h) .....	17
3.8.5	EEPROM programmieren (22h) .....	17
3.8.6	EEPROM programmieren mit PIN (F0h).....	18
3.8.7	Onlinedatenabfrage (23h) .....	18
3.8.8	Onlinedatenabfrage mehrere Kanäle (2Fh) .....	19
3.8.9	Offlinedatenabfrage (24h) .....	20
3.8.10	Reset / Reset auf Default (25h) .....	21
3.8.11	Reset mit Verzögerung (2Eh) .....	21
3.8.12	Statusabfrage (26h).....	21
3.8.13	Letzte Fehlermeldung (2Ch).....	22
3.8.14	Uhrzeit / Datum setzen (27h).....	22
3.8.15	Uhrzeit / Datum auslesen (28h) .....	22
3.8.16	Test- / Abgleichbefehl (29h) .....	23
3.8.17	Monitor (2Ah) .....	23
3.8.18	Protokollwechsel (2Bh).....	24
3.8.19	neue Geräte-ID setzen (30h).....	24
3.8.20	UMB-Tunnel (36h).....	25
3.8.21	Firmware übertragen (37h).....	26
3.8.22	Binärdaten übertragen (38h) .....	27
3.9	Kanalbelegungen .....	29
3.9.1	Kanalbelegung allgemeine Zuordnung .....	29

3.9.2	TLS-Kanalbelegung.....	29
3.10	Einheitenliste.....	30
3.10.1	Temperatur .....	30
3.10.2	Feuchte .....	30
3.10.3	Längen .....	30
3.10.4	Geschwindigkeiten .....	30
3.10.5	Elektrische Größen .....	30
3.10.6	Frequenz.....	30
3.10.7	Druck .....	31
3.10.8	Volumen.....	31
3.10.9	Zeit.....	31
3.10.10	Niederschlag.....	31
3.10.11	Sonstige .....	32
3.11	Beispiel einer Binärprotokoll-Abfrage .....	33
3.12	Bemerkungen zu Broadcast .....	34
<b>4</b>	<b>Anhang.....</b>	<b>35</b>
4.1	CRC-Berechnung.....	35
4.1.1	Beispiel einer CRC16-MCRF4XX -Berechnung in C.....	35
4.2	Automatisches auslesen eines Netzwerkes.....	37
4.2.1	Hintergrund .....	37
4.2.2	Notwendige ID-Konfiguration der Sensoren .....	37
4.2.3	Scannen des Netzwerkes.....	37
4.3	Daten-Typen in UMB-Produkten nach TLS2002 FG3.....	38
4.3.1	Beispiel einer TLS-Messwertabfrage.....	38
4.3.2	unterstützte TLS-DE-Typen FG3.....	38
4.3.3	Abgeleitete Größen .....	39
4.3.4	DE-Typ 70 „Zustand der Fahrbahnoberfläche“ (FBZ).....	39
4.3.5	DE-Typ 71 „Niederschlagsart“ (NS).....	40
4.3.6	DE-Typ 140 „Türkontakt“ (TK).....	41
4.3.7	DE-Typ 140 „Türkontakt“ (TK) invertiert.....	41
4.4	Zeichenketten und Codierung .....	42
4.4.1	Zeichenketten.....	42
4.4.2	ASCII-Codierung.....	42

# 1 Versionsgeschichte

Dokument-Version	Datum	Bearbeiter	Änderungsbeschreibung
0.0	24.11.2004	SR	Erstellung
0.1	14.06.2005	EES	erste Ausgabe
0.2	22.12.2005	EES	Aktualisierung Schnittstelle Hardware 2-draht
0.3	07.02.2006	EES	Umbenennung auf UMB, Ergänzung „Automatisches auslesen eines Netzwerks“ und Geräteinformation
0.4	07.03.2006	EES	Nomenklatur ergänzt, Kommandoübersicht BC ergänzt
0.5	09.03.2006	EES	- Kommando Geräteinformation um E2-Größe und die Angabe der Art der Info in der Antwort erweitert - Kanalbelegung um relative Messwerte erweitert
0.6	04.04.2006	EES	Einheitenliste ergänzt
1.0	12.04.2006	EES	- Messwerttypen ergänzt - Kommando Geräteinformation um Messwerttyp erweitert <b>erste freigegebene Version!</b>
1.1	19.05.2006	EES	- in die Antwort von Kommando Uhrzeit / Datum auslesen (28h) Statusbyte eingefügt
		SR	- Status 29h Unterspannung definiert.
		EES	- Kanalbelegung je Geräteklasse ergänzt - In der Kanalbelegung TLS-Kodierungen und im Anhang die Liste der unterstützten DE-Typen nach FG3 ergänzt - Logo eingefügt
1.2	18.07.2006	EES	- Status 2Ah Hardwarefehler definiert. - Status 2Bh Fehler in der Messung definiert. - Einheitenliste ergänzt (l/m <sup>2</sup> ) - Zeichensatz ASCII ergänzt - Messwerttyp ‚Summe‘ 14h definiert
		BEL	- R2S integriert
		EES	- Status 52h Channel Overrange definiert - Status 53h Channel Underrange definiert - Einheitenliste ergänzt (hPa Hektopascal)
		BEL	- bei R2S Kanal 700 die Niederschlagsarten in Anlehnung an die <b>WMO definiert</b>
		EES	- allg. Kanalbelegung 10500 für Impulse und 2000 für weitere TLS-Kanäle definiert - Messwerttyp 15h für ‚vektorieller Mittelwert‘ definiert - TLS-Kanäle für ANACON ergänzt (LD und 2. Kanal) - Status-Codes für Kalibrierung ergänzt - Änderung der Antwort auf die TLS-Kanal-Abfrage (Anpassung an bestehende Implementierung!)
1.3	08.08.2006	EES	- Antwortzeit des Kalibrier-Kommandos (29h) von kurz auf lang geändert - Status-Code 36h in ‚Kanal deaktiviert‘ geändert - Datentyp der TLS-Kanäle in der allg. Kanalbelegung der Geräteklassen angepasst - Kapitel Kanalbelegung nach hinten verschoben - auf Grund von rechenintensiven Kanälen (z.B. ANACON) wird die Antwortzeit für Onlinedatenabfrage (23h) auf ‚lang‘ geändert - Hinweis bei ‚Onlinedatenabfrage mehrere Kanäle‘ (2Fh) in Bezug auf lange Antwortzeiten ergänzt - ANACON TLS-Kanal DE-Typ 66 TPT Taupunkt ergänzt - diverse Kanalbelegungen ergänzt - Kommando 29h in ‚Testbefehl‘ umbenannt, da dieses nicht nur zum Abgleich, sondern auch zum Test diverser Gerätefunktionen verwendet wird

1.4	12.09.2006	EES	- Bemerkung ergänzt, dass im ASCII-Protokoll keine TLS-Kanäle zur Verfügung stehen
		SR	- Kanalbelegung Straßensensor. - Längeneinheit mil hinzugefügt
1.5	10.12.2007	EES	- Kommando ‚set_new_id‘ vers 1.1 erweitert - Status-Code 28h um info erweitert - Einheit Knoten ergänzt
		BEL	Beim R2S Kanäle erweitert um die nicht-metrischen Einheiten inch und mil bzw. inch/h und mil/h (Kapitel 3.9.3 und 3.10.3 sowie 3.10.9)
		EES	- Kanalliste für ANACON-UMB ergänzt (abs. Feuchte und Druck) - Geräteklasse und Kanalliste für Kompaktwetterstation ergänzt - Produktliste ergänzt - Kanalbelegung Geräteklasse 6 universelle Messtransmitter für Niederschlag und Impulse geändert
		EES	- Produktbezeichnungen für die Kompaktwetterstationen geändert - Error-Code INIT_ERROR (2Ch) = Fehler bei der Geräteinitialisierung und OS_ERROR (2Dh) = Fehler im Betriebssystem ergänzt - Kanalbezeichnungen für abs. und rel. Luftdruck geändert
1.6	17.12.2010	EES	- Produktliste ergänzt - Kanalbelegung Klasse 7 für Wind angepasst und Datentyp für abs. Niederschlag in float geändert - Fehlercodes für FLASH ergänzt (0x60; 0x61; 0x62)
		BR	- zusätzlicher ASCII Fehlercode 65525 - Artikelnummer ARS31-UMB eingefügt - Kanalliste ‚aktiver Straßensensor‘ ergänzt
		EES	- Klassen- und Geräte-ID überarbeitet (4 Bit Reserve ergänzt)
		EES	- NIRS31-UMB ergänzt
1.7	03.05.2021	RZ	- IRS31Pro-UMB ergänzt
		KRM	- Firmware Update Kommando (37h) hinzugefügt - MARWIS-UMB unterstützte Befehle hinzugefügt
		RZF	- UMB-Tunnel Kommando (36h) hinzugefügt - Layer-Definition korrigiert - Beschreibung für Onlinedatenabfrage (2Fh) korrigiert - Beschreibung für Broadcast korrigiert - UMB-ASCII entfernt - Definition für UMB-ASCII 2.0 in getrenntem Dokument
		KRM	- Reset Subkommando 0x14 spezifiziert
		RZF	- UMB-Binärdaten Kommando (38h) hinzugefügt
		KRM	- UMB-Fehler CALC_ERROR dokumentiert
		KRM	- UMB-Kommando zum Auslesen gerätespezifischer Versionsinformationen dokumentiert
		RZF	- <status> in Antwort-Beschreibungen kenntlich gemacht
		BR	Weitere TLS-Kanäle ergänzt

## 2 Vorbemerkung

Das hier beschriebene Protokoll wurde für meteorologische Sensoren entwickelt und ermöglicht eine vereinheitlichte Kommunikation mit unterschiedlichen Geräten wie z.B. IRS31-UMB, VS20-UMB und R2S-UMB.

### 2.1 Einschränkung der Garantie

**Die in diesem Dokument beschriebene Methoden und Einstellungen erlauben eine über den Standard (ConfigTool.NET) hinausgehende Konfigurierbarkeit der Geräte. Bei der Wahl von unzulässigen Einstellungen kann es zum Verlust der spezifizierten Messgenauigkeit bis hin zum Geräteversagen kommen. OTT HydroMet behält sich vor, bei Anwendung der hier beschriebenen Verfahren die Garantie auf die Produkte einzuschränken.**

### 2.2 Nomenklatur

Gerät: Der Begriff Gerät wird in diesem Dokument als Synonym für die Gerätefamilie der meteorologischen Sensoren wie z.B. IRS31, VS20 und R2S verwendet.

Hexadezimalwerte werden durch das Postfix ‚h‘ gekennzeichnet. Dezimalwerte werden durch das Postfix ‚d‘ gekennzeichnet

Mit ‚BC‘ sind Kommandos gekennzeichnet, welche als Broadcast abgesetzt werden können. ‚NBC‘ steht für ‚nicht broadcast-fähig‘ (siehe auch Bemerkungen zu Broadcast Seite 34).

### 2.3 Datenformat und Byteorder im Kommunikationsprotokoll:

LONG: LowLowByte LowHighByte HighLowByte HighHighByte

INT: LowByte HighByte

FLOAT: Nach IEEE Format (4 Bytes)

DOUBLE: Nach IEEE Format (8 Bytes)

### 2.4 Physikalische Anbindung und Hardwarestruktur

Üblicherweise erfolgt die Ansteuerung der Geräte in einem Netzwerk über eine halbduplexe RS485-2-Draht-Schnittstelle. ISOCON besitzt zusätzlich eine RS232-Schnittstelle. Die Baudrate beträgt in der Werkseinstellung 19200 Baud mit 8 Datenbits, einem Stoppbit und keiner Parität (8N1). Alternativ können die Telegramme auch über TCP-Sockets oder vergleichbar versendet werden. Die Anbindung an den Sensor erfolgt dann über LAN-Seriell-Adapter oder, falls der Sensor es unterstützt, direkt per Ethernet oder WLAN. Die verwendeten Port-Nummern sind der jeweiligen Gerätebeschreibung zu entnehmen.

### 2.5 Softwareprotokoll

Die Konfiguration und Abfrage der Geräte erfolgt im Binärprotokoll. Da das System ohne Kollisionserkennung arbeitet, wird nach strengem Controller-Device-Prinzip gearbeitet, d.h. eine Transaktion wird **immer** vom Controller eingeleitet.

Alternativ steht mit UMB-ASCII 2.0 ein menschenlesbares Protokoll zur Verfügung (**wird derzeit aber noch nicht von allen Produkten unterstützt**). Siehe Beschreibung zu UMB-ASCII 2.0.

### 3 UMB-Binär-Protokoll (Version 1.0)

Um die logischen Abschnitte des Header-Aufbaus zu abstrahieren, kann das OSI-Referenzmodell (Open Systems Interconnection) der International Standards Organisation (ISO) verwendet werden.

Die Datengramme durchlaufen die einzelnen Schichten (Layer) des Protokoll-Stacks und werden somit nach und nach mit den Header-Daten versehen. Auf diese Weise entsteht ein Frame, dessen maximale Länge auf 255 Bytes begrenzt ist. Für Nutzdaten stehen 210 Bytes zur Verfügung.

#### 3.1 Protokoll-Stack (Framing)

##### 3.1.1 (7) Application Layer

Die Anwendungsschicht ist nicht direkt Teil des UMB-Protokolls und entspricht den Programmen oder Funktionen, die sich des UMB-Protokolls bedienen, bzw. dessen Befehle ausführen. Als Beispiel sind hier für einen Controller das Config-Tool (PC) und für ein Device die Funktionen für die Abfrage eines Sensorwertes zu nennen.

##### 3.1.2 (6) Presentation Layer

Ein Kommando besteht aus mindestens zwei Zeichen: Dem Kommando <cmd> und der Versionsnummer des Kommandos <verc>. Die optionale Payload kann bis zu 210 Zeichen beinhalten. Der Wertebereich eines Bytes der Payload beträgt 0 bis 255. Bei einem Antwort-Telegramm ist das Status-Byte in der Payload enthalten.

1	2	3 - 4	5 - 6	7	8	9	10	11 ... (8 + len) optional	9 + len	10 + len 11 + len	12 + len
						<cmd>	<verc>	<payload>			

##### 3.1.3 (5) Session Layer

Die Sitzungsschicht ist nicht Teil des UMB-Protokolls und damit transparent.

##### 3.1.4 (4) Transport Layer

Die Transportschicht ist nicht Teil des UMB-Protokolls und damit transparent.

##### 3.1.5 (3) Network Layer

Die Adressierung erfolgt über eine 16-Bit Adresse. Den von der übergeordneten Schicht erstellten Nutzdaten werden in der Netzwerkschicht (Network Layer) die Empfänger- (<to>) und Absender-Adresse (<from>) hinzugefügt.

1	2	3 - 4	5 - 6	7	8	9	10	11 ... (8 + len) optional	9 + len	10 + len 11 + len	12 + len
		<to>	<from>								

### 3.1.6 (2b) Data-Link Layer / Logical Link Control

Zur Kennzeichnung des Datenrahmens werden 2 Steuerzeichen verwendet (SOH, EOT). Das Versionsfeld (<ver>) bestimmt die weitere Struktur des Telegramms und ist die Versionsnummer des Binärprotokolls (hier Version 1.0). Dabei stellt das obere Nibble die Versionsnummer und das untere Nibble die Revisionsnummer dar.

Beispiel: Version 1.0 → <ver> = 10h = 16d

Die Anzahl der Bytes zwischen den Steuerzeichen für die Nutzdaten (STX, ETX) wird durch das Längenfeld (<len>) angegeben. Um vor Übertragungsfehlern zu schützen wird das Telegramm durch eine Checksumme (<cs>) verifiziert.

1	2	3 - 4	5 - 6	7	8	9	10	11 ... (8 + len) optional	9 + len	10 + len 11 + len	12 + len
SOH	<ver>			<len>	STX				ETX	<cs>	EOT

### 3.1.7 (2a) Data-Link Layer / Media Access Control

Eine Zugriffskontrolle auf das Übertragungsmedium wird durch Timeouts geregelt.

#### 3.1.7.1 Timing Controller

Für die Implementierung des Protokolls im Controller werden folgende Punkte festgelegt:

- Es muss ein strenges **Controller-Device-Prinzip** eingehalten werden. (pro System darf es nur einen Controller geben!)
- nach dem Empfang einer Device-Nachricht muss der Controller eine Mindestpause von 3 Zeichen einhalten.
- nach Absetzen eines Broadcast darf der nächste Befehl frühestens nach 500 ms erfolgen.
- Bei speziellen Sensoren kann ein Retry notwendig sein. Die empfohlene Anzahl der Retries ist 3. Der Abstand der Nachrichten muss mindestens 500 ms betragen, jedoch in Summe nicht länger als 3 s.

Nach Ablauf dieser Zeit kann der Controller vom Verlust der Nachricht ausgehen und mit den Retries beginnen. Bei einer direkten RS232-Verbindung betragen die Timeoutzeiten für

- Kommandos mit Standard Antwortzeit: 60 ms
- Kommandos mit Antwortzeit lang: 510 ms

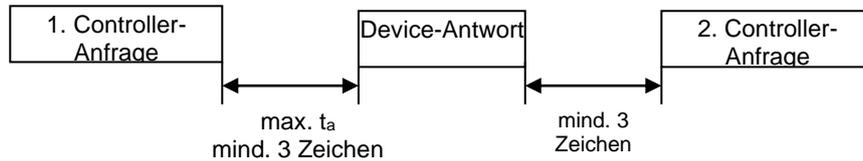
**ACHTUNG!!** Erfolgt die Abfrage z.B. über Ethernet oder GPRS muss die Timeoutzeit entsprechend der Laufzeit des verwendeten Mediums angepasst werden. Hierzu sollen die Timeoutzeiten in der Konfiguration der Controller-Software einstellbar sein.

#### 3.1.7.2 Timing Sensor

Für die Implementierung des Protokolls im Sensor werden folgende Punkte festgelegt:

- Bei kritischen Messungen kann der Empfangsinterrupt gesperrt werden.
- Ein Sensor (Device) muss bei empfangenem Kommando spätestens nach der Zeit  $t_a$  mit dem Senden der Antwort beginnen, jedoch nicht früher als die Mindestpause von 3 Zeichen. Dauert bei der Onlineabfrage die Messung länger, wird der zuletzt gespeicherte Messwert ausgegeben und nach der Abfrage gemessen.
- die Antwortzeiten ( $t_a$ ) sind in 2 Klassen aufgeteilt:
  - o Standard: die Antwortzeit  $t_a$  ist maximal 50 ms
  - o Lang: bei bestimmten Kommandos ist eine längere Verarbeitungszeit notwendig. Die maximale Antwortzeit  $t_a$  beträgt hier 500 ms. Diese Kommandos sind entsprechen gekennzeichnet.

- Überschreitet ein Kommando die zulässige Antwortzeit, so muss das Device den Versand der Antwort unterdrücken. Der Controller erhält keine Antwort, kann jedoch daraus nicht ableiten, ob das Kommando ausgeführt wurde oder nicht, da die zulässige Antwortzeit mit Abschluss der Ausführung abgelaufen sein könnte. Das Kommando muss vom Controller also auf jeden Fall durch den Controller wiederholt werden. (Bei Vorgängen, die Sequenzen aus mehreren Kommandos beinhalten, kann das bedeuten, dass der Vorgang abgebrochen und neu gestartet werden muss).



### 3.1.8 (1) Physical Layer

Das physikalische Medium ist nicht Teil des UMB-Protokolls und damit prinzipiell frei wählbar. Üblicherweise wird eine 2-Draht RS485, Standard-Baudrate: 19200 Baud, 8 Datenbits, 1 Stopbit, keine Parität verwendet.

Des Weiteren ist es möglich das UMB-Telegramm über andere Übertragungsmedien wie Ethernet, WLAN oder Bluetooth zu tunneln.

### 3.1.9 Zusammenfassung

Zur Übersicht ist hier ein komplettes Anfrage-Frame abgebildet:

1	2	3 - 4	5 - 6	7	8	9	10	11 ... (8 + len) optional	9 + len	10 + len 11 + len	12 + len
SOH	<ver>	<to>	<from>	<len>	STX	<cmd>	<verc>	<payload>	ETX	<cs>	EOT

Antwort-Frame:

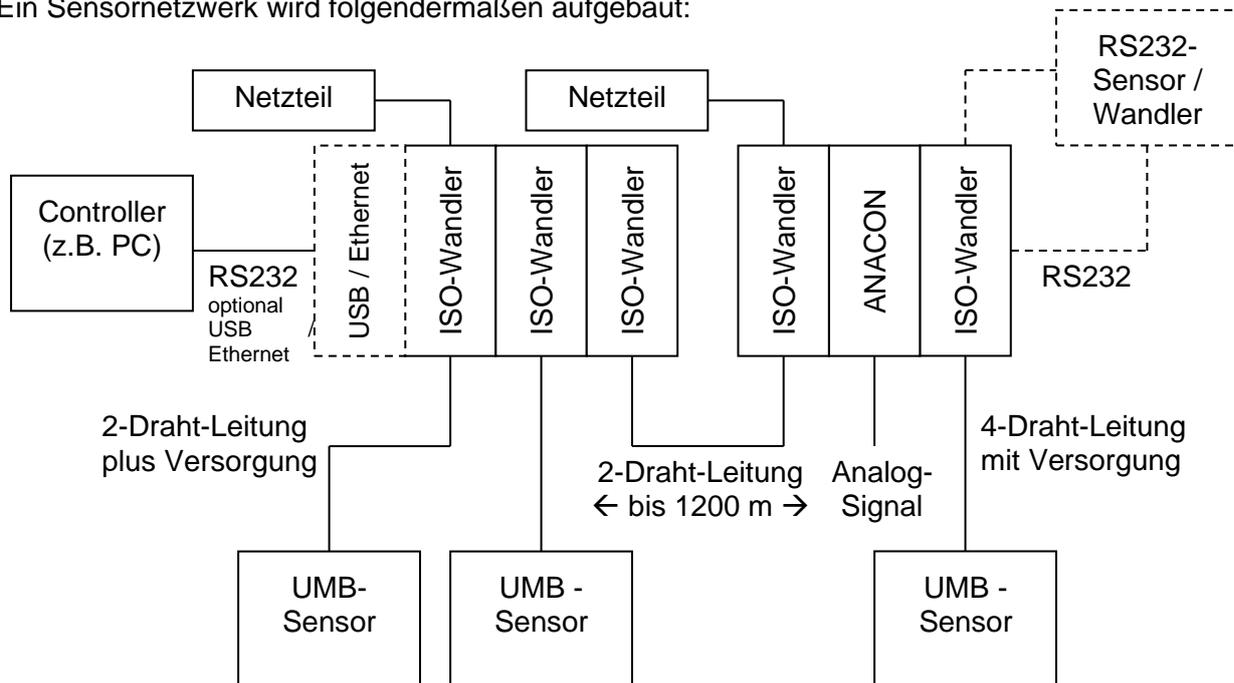
1	2	3 - 4	5 - 6	7	8	9	10	11	12 ... (9 + len) optional	10 + len	11 + len 12 + len	13 + len
SOH	<ver>	<to>	<from>	<len>	STX	<cmd>	<verc>	<status>	<payload>	ETX	<cs>	EOT

- SOH Steuerzeichen für den Start eines Frames (01h) 1 Byte
- <ver> Header-Versionsnummer, Bsp.: V 1.0 → <ver> = 10h = 16d; 1 Byte
- <to> Empfänger-Adresse, 2 Bytes
- <from> Absender-Adresse, 2 Bytes
- <len> Anzahl der Datenbytes zwischen STX und ETX; 1 Byte
- STX Steuerzeichen für den Start der Nutz-Datenübertragung (02h); 1 Byte
- <cmd> Befehl; 1 Byte
- <verc> Versionsnummer des Befehls; 1 Byte
- <status> Status des Befehls (bei Antwort-Telegrammen); 1 Byte
- <payload> Datenbytes; 0 – 210 Byte
- ETX Steuerzeichen für das Ende der Nutz-Datenübertragung (03h); 1 Byte
- <cs> Checksumme, 16 Bit CRC; 2 Byte
- EOT Steuerzeichen für das Ende des Frames (04h); 1 Byte

Steuerzeichen: SOH (01h), STX (02h), ETX (03h), EOT (04h).

### 3.2 Topologie

Ein Sensornetzwerk wird folgendermaßen aufgebaut:



**Controller:** der Controller wird mit der im ISO-Wandler integrierten RS232-Schnittstelle verbunden. Optional gibt es noch ein Schnittstellen-Modul für USB und Ethernet via Virtual-COM-Port.

**Sensor:** die Sensoren werden über eine 2-Draht-Verbindung plus Versorgung mit jeweils einem ISO-Wandler verbunden; weiter stellt der Wandler auch die Versorgungsspannung für den Sensor zur Verfügung

**ISO-Wandler:** die Wandler sind untereinander über anreihbare Steckverbinder gekoppelt. Um größere Distanzen zwischen ISO-Wandlern zu schaffen, können diese über die RS485-Schnittstelle miteinander verbunden werden.

#### Sensoren anderer Hersteller

Vorausgesetzt ein Sensor eines anderen Herstellers arbeitet ebenfalls nach dem Controller-Device-Prinzip, kann dieser an die RS232-Schnittstelle eines ISO-Wandlers angeschlossen werden; weiter ist denkbar, dass über einen entsprechenden Wandler (z.B. Phoenix) auch Sensoren mit RS422/485/2-Draht/4-Draht so mit dem Messnetz verbunden werden können.



#### Messmodule

Für Sensoren ohne Daten-Schnittstelle (z.B. 0 – 1V oder 4 – 20mA) gibt es intelligente Messmodule, welche die Analog-Signale auf dem Bus zur Verfügung stellen.

### 3.3 Klassen- und Geräte-ID

Die Adressierung erfolgt über eine 16-Bit Adresse. Diese gliedert sich in eine Sensorklassen-ID und eine Geräte-ID.

Adresse (2 Bytes = 16 Bit)				
Bit 15 – 12 (obere 4 Bit)		Bit 11 – 8 (mittlere 4 Bit)	Bit 7 – 0 (untere 8 Bit)	
Klassen-ID (0 bis 11)		Reserve	Geräte-ID (0 – 255)	
0	Broadcast	Für internen Gebrauch, normalerweise 0	0	Broadcast
1	Straßensensor IRS31-UMB		1 - 255	verfügbar
2	Regensensor R2S-UMB			
3	Sichtweitemesser VSx-UMB			
4	aktiver Straßensensor ARS31-UMB			
5	berührungsloser Straßensensor NIRS31-UMB			
6	universelle Messtransmitter ANACON-UMB			
7	Kompakt-Wetterstation WSx-UMB			
8	Windmesser VENTUS / V200A			
9	Straßensensor IRS31Pro-UMB			
10	berührungsloser mobiler Straßensensor MARWIS-UMB / STARWIS-UMB			
11	Schneehöhensensor SHM31-UMB			
Klassen-ID (12 bis 14)		Subklassen-ID		
12	Wolkenhöhsimulator	0		
	Reserviert für Erweiterungen	t.b.d.		
13	Diverse Protokoll-Konverter	t.b.d.		
14	Reserviert für Erweiterungen	t.b.d.		
Klassen-ID (15)		Reserve		
15	Controller bzw. Steuergeräte	n. def.		

Bei Klassen und Geräten ist jeweils die ID = 0 als Broadcast vorgesehen. So ist es möglich, ein Broadcast auf eine bestimmte Klasse zu senden. Wird ein Broadcast an die Klassen-ID 0 gesendet, ist auch die Geräte-ID auf Broadcast (= 0) zu setzen.

#### 3.3.1 Beispiele für die Bildung von Adressen

Soll z.B. ein Straßensensor mit der Geräte-ID 3 adressiert werden, geschieht das wie folgt:

Klassen-ID für Straßensensor ist 1 = 1h

Geräte-ID ist 3 = 3h

Setzt man die Klassen- und Geräte-ID zusammen ergibt sich eine Adresse 1003h = 4099d.

weitere Beispiele:

Klassen-ID	Geräte-ID	Adresse	Erklärung
3h	1h (1d)	3001h (12289d)	Sichtweiten-Sensor mit Geräte-ID 1
0h	000h (0000d)	0000h (0d)	Broadcast an alle Geräte bzw. Sensoren.
1h	000h (0000d)	1000h (4096d)	Broadcast an alle Straßensensoren.

### 3.4 CRC-Prüfsumme

Die CRC16-MCRF4XX-Prüfsumme wird mit folgendem Polynom gebildet:

$$x^{16} + x^{12} + x^5 + 1 \text{ (LSB first-Mode; Startwert FFFFh)}$$

Die Prüfsumme wird über alle Bytes vor der Checksumme gebildet (1 ... 9 + len), also von SOH bis ETX jeweils einschließlich.

Für die Prüfsumme gilt die Bytefolge Little Endian.

Empfängt ein Gerät ein Frame mit falscher CRC-Prüfsumme erfolgt auf dieses Kommando keine Reaktion.

Ein Programmbeispiel einer CRC-Berechnung befindet sich im Anhang.

### 3.5 Datentypen

In diesem Protokoll werden z.B. für die Messwertabfrage folgende Datentypen verwendet:

<type>	Type Name	Define	Bytes	Range
10h (16d)	unsigned char	UNSIGNED_CHAR	1	0 ... 255
11h (17d)	signed char	SIGNED_CHAR	1	-128 ... 127
12h (18d)	unsigned short	UNSIGNED_SHORT	2	0 ... 65.535
13h (19d)	signed short	SIGNED_SHORT	2	-32.768 ... 32.767
14h (20d)	unsigned long	UNSIGNED_LONG	4	0 ... 4.294.967.295
15h (21d)	signed long	SIGNED_LONG	4	-2.147.483.648 ... 2.147.483.647
16h (22d)	float	FLOAT	4	±1.18E-38 ... ±3.39E+38 (7 digits)
17h (23d)	double	DOUBLE	8	±2.23E-308 ... ±1.79E+308 (15 digits)

Bemerkung: float und double in IEEE Format

### 3.6 Messwerttypen

In diesem Protokoll werden für die Messwertabfrage folgende Messwerttypen verwendet:

<type>	Type Name	Define	Beschreibung
10h (16d)	current	MWT_CURRENT	aktueller Messwert
11h (17d)	min	MWT_MIN	Minimalwert
12h (18d)	max	MWT_MAX	Maximalwert
13h (19d)	avg	MWT_AVG	Mittelwert
14h (20d)	sum	MWT_SUM	Summe
15h (21d)	vct	MWT_VCT	vektorieller Mittelwert

### 3.7 Status- und Errorcodes

Jedes Antwort-Telegramm enthält ein Status-Byte. Dieses gibt Aufschluss über den Erfolg oder Misserfolg des Kommandos. Bei bestimmten Errorcodes werden weitere Informationen übermittelt, welche eine exakte Fehleranalyse ermöglichen.

Wurde ein Kommando nicht erfolgreich verarbeitet wird eine Fehlermeldung ausgegeben:

**<cmd>\_<verc><status>, [<info><sup>n</sup>]**

Gibt es zu einem Status keine weitere Information entfällt <info><sup>n</sup>. Damit die Frame-Steuerzeichen nicht zu häufig auftreten, wird bei diesen Codes auf 01h bis 0Ah verzichtet.

#### Codes:

<status>	<info>	Define	Beschreibung
00h (0d)		OK	Kommando erfolgreich; kein Fehler; alles i.O.
10h (16d)		UNKNOWN_CMD	unbekanntes Kommando; wird von diesem Gerät nicht unterstützt
11h (17d)		INVALID_PARAM	ungültige Parameter
12h (18d)		INVALID_HEADER	ungültige Header-Version
13h (19d)		INVALID_VERC	ungültige Version des Befehls
14h (20d)		INVALID_PW	Passwort für Kommando falsch
15h (21d)		INVALID_WERT	ungültiger Wert
20h (32d)		READ_ERR	Lesefehler
21h (33d)		WRITE_ERR	Schreibfehler
22h (34d)	<maxlength>	TOO_LONG	Länge zu groß; max. zulässige Länge wird in <maxlength> angegeben
23h (35d)		INVALID_ADDRESS	ungültige Adresse / Speicherstelle
24h (36d)	<channel> <sup>2 1)</sup>	INVALID_CHANNEL	ungültiger Kanal
25h (37d)		INVALID_CMD	Kommando in diesem Modus nicht möglich
26h (38d)		UNKNOWN_CAL_CMD	unbekanntes Test-/Abgleich-Kommando
27h (39d)		CAL_ERROR	Fehler bei der Kalibrierung
28h (40d)	<channel> <sup>2 (2)</sup>	BUSY	Gerät nicht bereit; z.B. Initialisierung / Kalibrierung läuft
29h (41d)		LOW_VOLTAGE	Unterspannung
2Ah (42d)		HW_ERROR	Hardwarefehler
2Bh (43d)		MEAS_ERROR	Fehler in der Messung
2Ch (44d)		INIT_ERROR	Fehler bei der Geräteinitialisierung
2Dh (45d)		OS_ERROR	Fehler im Betriebssystem
2Eh (46d)		COM_ERROR	Kommunikationsfehler auf einer internen Schnittstelle
2Fh (47d)		HW_SW_MISMATCH	Hardware und Software Version passen nicht
30h (48d)		E2_DEFAULT_KONF	Fehler in der Konfiguration, Default-Konfiguration wurde geladen
31h (49d)		E2_CAL_ERROR	Fehler im Abgleich / der Abgleich ist ungültig, Messung nicht möglich
32h (50d)		E2_CRC_KONF_ERR	CRC-Fehler beim Laden der Konfiguration; Default-Konfiguration wurde geladen
33h (51d)		E2_CRC_KAL_ERR	CRC-Fehler beim Laden der Abgleich-Daten; Messung nicht möglich
34h (52d)		ADJ_STEP1	Abgleich Step 1
35h (53d)		ADJ_OK	Abgleich OK
36h (54d)		CHANNEL_OFF	Kanal deaktiviert

<sup>1)</sup> info <channel><sup>2</sup> erfolgt nur bei Kommando Online-Messwertabfrage

<sup>2)</sup> info <channel><sup>2</sup> erfolgt nur bei Kommando Online-Messwertabfrage

37h (55d)		SERVICE_MODE	Service Modus aktiv
38h (56d)		RESET_PENDING	Ein Geräte-Reset steht unmittelbar bevor
39h (57d)		ADJ_DEFAULT	Abgleichwerte ungültig
50h (80d)	<channel> <sup>2</sup>	VALUE_OVERFLOW	Messgröße (+Offset) liegt außerhalb des eingestellten Darstellungsbereichs
51h (81d)	<channel> <sup>2</sup>	VALUE_UNDERFLOW	
52h (82d)	<channel> <sup>2</sup>	CHANNEL_OVERRANGE	Messwert (physikalisch) liegt außerhalb des Messbereichs (z.B. ADC-Ovrange)
53h (83d)	<channel> <sup>2</sup>	CHANNEL_UNDERRANGE	
54h (84d)	<channel> <sup>2</sup>	DATA_ERROR	Datenfehler in den Messdaten oder keine gültigen Daten vorhanden
55h (85d)	<channel> <sup>2</sup>	MEAS_UNABLE	Gerät / Sensor kann auf Grund der Umgebungsbedingungen keine gültige Messung durchführen
56h (86d)	<channel> <sup>2</sup>	CALC_ERROR	Interner Rechenfehler (NaN, Division durch 0)
60h (96d)		FLASH_CRC_ERR	CRC-Fehler in den Flash-Daten
61h (97d)		FLASH_WRITE_ERR	Fehler beim Schreiben ins Flash; z.B. Speicherstelle nicht gelöscht
62h (98d)		FLASH_FLOAT_ERR	Flash enthält ungültige Float-Werte
63h (99d)		FLASH_ERR	Flash defekt, Initialisierungsfehler
64h (100)		CONFIG_ERR	Fehlerhafte Konfiguration
FFh (255d)		UNKNOWN_ERR	unbekannter Fehler

### 3.8 Kommandos (Datagramme)

Aus Gründen der Übersichtlichkeit beschränkt sich die folgende Darstellung der Kommandos auf die Darstellungsschicht (Presentation Layer). Zur besseren Darstellung wird folgende Kurzform verwendet:

**<cmd><sub><verc></sub>[<payload0><sup>n</sup>, <payload1><sup>n</sup>, ...]**

**<cmd><sub><verc></sub><status>[<payload0><sup>n</sup>, <payload1><sup>n</sup>, ...]**

Hexadezimalwerte werden durch das Postfix ‚h‘ gekennzeichnet. Zeichenketten stehen in einfachen Anführungszeichen und werden mit Leerzeichen (20h) auf die festgelegte Länge aufgefüllt. Beim Übertragen von Wörtern gilt die Bytefolge Little-Endian (Intel, lowbyte first). Platzhalter für syntaktische Einheiten sind durch spitze Klammern gekennzeichnet. Ist die Länge der Variable größer als 1 Byte, wird diese im Index ‚n‘ angegeben.

#### 3.8.1 Übersicht der Kommandos

Sortiert nach <cmd>:

<cmd>	Beschreibung	BC	AZ
20h	Hard-Softwareversion und		k
21h	EEPROM auslesen		l
22h	EEPROM programmieren		l
23h	Onlinedatenabfrage		l
24h	Offlinedatenabfrage		k
25h	Reset / Default	●	k/l*
26h	Statusabfrage		k
27h	Uhrzeit / Datum setzen	●	k
28h	Uhrzeit / Datum auslesen		k
29h	Testbefehl		l
2Ah	Monitor		l
2Bh	Protokollwechsel	●	k
2Ch	letzte Fehlermeldung		k
2Dh	Geräteinformation		k
2Eh	Reset mit Verzögerung	●	k

2Fh	Onlinedatenabfrage mehrere Kanäle		l
30h	neue Geräte-ID dauerhaft setzen (verc 1.0)	●	k
30h	neue Geräte-ID temporär setzen (verc 1.1)	●	k
36h	UMB-Tunnel		l
37h	Firmware übertragen		l
38h	Binärdaten übertragen		l
40h – 7Fh	reserviert für gerätespezifische Kommandos (siehe Geräte-beschreibung)		
80h – 8Fh	reserviert für Entwicklung		
F0h	EEPROM programmieren mit PIN		l

AZ = Antwortzeit; k = kurz; l = lang

BC = broadcast-fähiges Kommando

\* Je nach Sub-CMD, siehe Befehlsspezifikation.

\*\*Anfrage wird mit ‚unbekanntes Kommando‘ beantwortet: <cmd><sub>10h</sub>[10h]

**ACHTUNG!!** Ein Gerät (Sensor) akzeptiert ein Kommando nur, wenn es **von einem Controller** gesendet wurde (Einhaltung Controller-Device-Prinzip).

### 3.8.2 Hard- und Softwareversion (20h)

**Kommando <cmd>:** 20h (NBC)

**Kommandoversion <verc>:** 1.0

**Daten <payload>:** keine

**Beschreibung:** Mit dem Kommando wird vom angesprochenen Gerät die Hard- und Softwareversion angefordert.

**Aufruf:** 20h<sub>10h</sub>[ ]

**Antwort:** 20h<sub>10h</sub><status>[<hardware>, <software>]

**Antwortzeit:** kurz

**Beispiel:** SW-Version 2.3 →<software> = 17h = 23d  
 HW-Version 6 →<hardware> = 06h = 6d

### 3.8.3 Geräteinformationen (2Dh)

**Kommando <cmd>:** 2Dh (NBC)

**Kommandoversion <verc>:** 1.0

**Daten <payload>:** <info>, <option><sup>n</sup>

**Beschreibung:** dieses Kommando gibt folgende Geräteinformationen zurück:

**Aufruf:** 2Dh<sub>10h</sub>[<info>, <option><sup>n</sup>]

<info> Art der gewünschten Information

<option><sup>n</sup> optional sind weitere Angaben notwendig

**Antwort:** 2Dh<sub>10h</sub><status>[<info>, <answer>]

**Antwortzeit:** kurz

<info>	<option>	Beschreibung	<Antwort>
10h	keine	Gerätebezeichnung	<bez> <sup>40</sup> z.B. ,Visibility-Sensor VS20'
11h	keine	Gerätebeschreibung	<beschr> <sup>40</sup> z.B. ,Sichtweite A92 West'
12h	keine	Hard- und Softwareversion	<hardware>, <software> Version 2.3 = 17h = 23d
13h	keine	erweiterte Versions-Info	<Lfd.-Nr> <sup>2</sup> , <MMJJ> <sup>2</sup> , <Projekt> <sup>2</sup> , <Stüli>, <SPlan>, <hardware>, <software>, <e2version>, <geräteversion> <sup>2</sup>
14h	keine	Größe des EEPROM	<e2_size> <sup>2</sup>
15h	keine	Anzahl verfügbare Kanäle	<channels> <sup>2</sup> , <blocks>
16h	<block>	Nummern der Kanäle	<block>, <channels>, [<channel> <sup>2</sup> ] <sup>channels</sup>
17h	keine	Anzahl der gerätespezifischen Versionsinformation-Slots auslesen	<num-slot-ids>
18h	<slot-id>	Gerätespezifische Versionsinformationen auslesen	<name> <sup>40</sup> , <major-version>, <minor-version>, <unixtime-date> <sup>4</sup> , <rc-version>
20h	<channel> <sup>2</sup>	Messgröße des Kanals	<channel> <sup>2</sup> , <messgröße> <sup>20</sup> z.B. ,visibility'
21h	<channel> <sup>2</sup>	Messbereich des Kanals	<channel> <sup>2</sup> , <min> <sup>n</sup> , <max> <sup>n</sup> n *)
22h	<channel> <sup>2</sup>	Messeinheit des Kanals	<channel> <sup>2</sup> , <einheit> <sup>15</sup> z.B. ,m'
23h	<channel> <sup>2</sup>	Datentyp des Kanals	<channel> <sup>2</sup> , <date_typ> z.B. 16h für float
24h	<channel> <sup>2</sup>	Messwerttyp	<channel> <sup>2</sup> , <mw_typ> z.B. 13h für Mittelwert
30h	<channel> <sup>2</sup>	komplette Kanalinfo	<channel> <sup>2</sup> , <messgröße> <sup>20</sup> , <einheit> <sup>15</sup> , <mw_typ>, <date_typ>, <min> <sup>n</sup> , <max> <sup>n</sup>
40h	Keine	Anzahl der IP-Interfaces	<interfaces>
41h	<interface>	IP-Information	<interface>, <name> <sup>20</sup> , <ip> <sup>4</sup> , <netmask> <sup>4</sup> , <gateway> <sup>4</sup> , <dns> <sup>4</sup> , <link>, <tcp_port> <sup>2</sup> n *)

n \*) Der Datentyp von Min und Max ist derselbe wie der des Messwerts, bei float32 float32, bei uint8 uint8 usw.

**Bemerkung:** Nicht alle Geräte stellen alle Informationen zur Verfügung. Ist eine Information nicht verfügbar, so wird der Status auf ‚Ungültiger Parameter‘ (11h) gesetzt und die Felder <info> und <answer> der Antwort entfallen.

Alle Zeichenketten sind mit Leerzeichen aufgefüllt und nicht mit ‘\0’ terminiert.

Bei der Abfrage der Nummern der Kanäle (16h) werden bis zu 100 Kanäle zu einem Block (beginnend mit Block 0) zusammengefasst. Hat ein Sensor mehr als z.B. 100 Kanäle, so gibt es entsprechend mehrere Blöcke. Die Anzahl der Blöcke wird bei der Abfrage der Anzahl der verfügbaren Kanäle (15h) angegeben.

Die gerätespezifischen Versionsinformationen (18h) geben Auskunft über die Stände einzelner Software-Komponenten. Ist <rc-version> = 0, dann handelt es sich um ein Public-Release, anderenfalls ein Release-Kandidat. Per Konvention ist die Information mit <slot-id> = 0 der Versionsstand der Systemsoftware. Die Build-Zeit wird als Unix-Timestamp in UTC-Zeit ausgegeben.

Bei der Abfrage des Messbereichs des Kanals (21h) oder der kompletten Kanalinfo (30h) hängt die Länge n von min und max vom Datentyp ab. (Siehe 3.5 Datentypen)

IP-Adressen, wie bei Info 41h, werden als 32-Bit unsigned Wert (little endian) übertragen.

### 3.8.4 EEPROM auslesen (21h)

**Kommando <cmd>:** 21h (NBC)

**Kommandoversion <verc>:** 1.0

**Daten <payload>:** <start><sup>2</sup>, <length>

**Beschreibung:** Mit dem Kommando wird die Übertragung von <length> Bytes ab der Speicherstelle <start> aus dem EEPROM begonnen.

**Aufruf:** 21h<sub>10h</sub>[<start><sup>2</sup>, <length>]

**Antwort:** 21h<sub>10h</sub><status>[<start><sup>2</sup>, <length>, <data><sup><length></sup>]

**Antwortzeit:** lang

**Bemerkung:** Die maximale Anzahl von Bytes (<length>) beträgt 200!

Im Fehlerfall entfallen die Felder <start>, <length> und <data>. Ist der Status 22h („TOO\_LONG“), so folgt ein Byte mit der zulässigen maximalen Länge der zu übertragenden Daten.

### 3.8.5 EEPROM programmieren (22h)

**Kommando <cmd>:** 22h (NBC)

**Kommandoversion <verc>:** 1.0

**Daten <payload>:** <start><sup>2</sup>, <length>, <data><sup><length></sup>

**Beschreibung:** Mit dem Kommando wird ein Datenblock <data> der Länge <length> Bytes an den Empfänger übertragen. Dieser wird ab der Adresse <start> in das EEPROM geschrieben. Wenn alle Bytes programmiert sind, wird der Absender des Kommandos durch eine Quittung über den Erfolg der Aktion informiert.

**Aufruf:** 22h<sub>10h</sub>[<start><sup>2</sup>, <length>, <data><sup><length></sup>]

**Antwort:** 22h<sub>10h</sub><status>[<max\_length>]

**Antwortzeit:** lang

**Bemerkung:** Da die maximale Zeit bis zur Antwort des Gerätes auf 500ms begrenzt ist, ist die maximale Anzahl von Bytes (<length>) vom Gerät abhängig, da es unterschiedlich lange dauern kann bis das EEPROM programmiert ist. Wird eine Anzahl von Bytes angegeben, welche von dem Gerät nicht verarbeitet werden kann, wird die maximale Anzahl nach dem Statuscode zurückgegeben (<max\_length>).

Im EEPROM gibt es schreibgeschützte Speicherstellen, welche nicht beschrieben werden können.

---

### 3.8.6 EEPROM programmieren mit PIN (F0h)

**Kommando <cmd>:** F0h (NBC)

**Kommandoversion <verc>:** 1.0

**Daten <payload>:** <pin><sup>2</sup>, <start><sup>2</sup>, <length>, <data><sup><length></sup>

**Beschreibung:** wie Kommando 22h; ermöglicht jedoch auch das Beschreiben geschützter E2-Adressen.

**Aufruf:** F0h<sub>10h</sub>[<pin><sup>2</sup>, <start><sup>2</sup>, <length>, <data><sup><length></sup>]

**Antwort:** F0h<sub>10h</sub><status>[max\_length]

**Antwortzeit:** lang

**ACHTUNG!!** Dieses Kommando ist ausschließlich für den internen Gebrauch bei OTT HydroMet vorgesehen! Bei unsachgemäßem Gebrauch kann das Gerät unbrauchbar gemacht oder beschädigt werden! (siehe Seite 6 - Einschränkung der Garantie!!) Dieses Kommando ist nicht für den Endanwender bestimmt. Alle notwendigen Einstellungen des Geräts können mit Hilfe der PC-Software von OTT HydroMet erledigt werden.

---

### 3.8.7 Onlinedatenabfrage (23h)

**Kommando <cmd>:** 23h (NBC)

**Kommandoversion <verc>:** 1.0

**Daten <payload>:** <channel><sup>2</sup>

**Beschreibung:** Mit dem Kommando wird ein Messwert eines bestimmten Kanals abgefragt.

**Aufruf:** 23h<sub>10h</sub>[<channel><sup>2</sup>]

**Antwort:** 23h<sub>10h</sub><status>[<channel><sup>2</sup>, <type>, <value><sup>n</sup>]

**Antwortzeit:** lang

<channel><sup>2</sup> gibt die Kanalnummer an

<type> gibt den Datentyp der Ausgabe an; davon hängt die Länge von <value> ab (siehe Seite 12 - Datentypen); entfällt, wenn <status> nicht OK ist

<value><sup>n</sup> abgefragter Wert; entfällt, wenn <status> nicht OK ist

**Bemerkung:** In der Gerätebeschreibung sind Kanal, Messwert und Format spezifiziert.

### 3.8.8 Onlinedatenabfrage mehrere Kanäle (2Fh)

**Kommando <cmd>:** 2Fh (NBC)

**Kommandoversion <verc>:** 1.0

**Daten <payload>:** <anzahl>, <channel><sup>2x<anzahl></sup>

**Beschreibung:** Mit dem Kommando können mehrere Kanäle mit einem Aufruf abgefragt werden. Für jeden Kanal wird ein Sub-Telegramm ausgegeben.

**Aufruf:** 2Fh<sub>10h</sub> [<anzahl>, <channel><sup>2 x <anzahl></sup>]

<anzahl> Anzahl der abgefragten Kanäle

<channel><sup>2</sup> gibt die Kanalnummern an

**Antwort:** 2Fh<sub>10h</sub> <status> [<anzahl>, {<sub-len>, <ch-status>, <channel><sup>2</sup> [<type>, <value><sup>n</sup>]} <anzahl>]

**Antwortzeit:** lang

<sub-len> gibt an, wie viele Bytes in diesem Sub-Telegramm noch folgen; wenn das nachfolgende Status-Byte z.B. ‚Value Overflow‘ anzeigt, dann entfallen <type> und <value><sup>n</sup> und der nächste Kanal folgt

<type> gibt den Datentyp der Ausgabe an; davon hängt die Länge von <value> ab (siehe Seite 12 - Datentypen)

<value><sup>n</sup> abgefragter Wert

**Bemerkung:** Bei Erstellung der Abfrage ist darauf zu achten, dass die Länge der Antwort nicht die maximale Framelänge überschreitet. Das <sub-len> Feld sollte in jedem Fall ausgewertet werden, auch wenn der Kanal-Status nicht ‚0‘ ist, um den korrekten Offset zum nächsten Kanal sicherzustellen.

**Kommando <cmd>:** 2Fh (NBC)

**Kommandoversion <verc>:** 1.1

**Daten <payload> (1. Aufruf):** <anzahl>, <channel><sup>2x<anzahl></sup>

**Daten <payload> (ab 2. Aufruf):** 00h

**Beschreibung:** Mit dem Kommando können mehrere Kanäle mit einem Aufruf abgefragt werden. Für jeden Kanal wird ein Sub-Telegramm ausgegeben. Neu ist, dass es für den Aufruf zwei verschiedene Anfrage-Telegramme gibt: Mit dem einen (1. Aufruf) werden die auszugebenden Kanäle festgelegt und intern gespeichert, mit dem andern (ab 2. Aufruf) werden die beim 1. Aufruf angegebenen Kanäle erneut ausgegeben. Das Antwortformat ist bei beiden Aufrufen identisch. Der 2. Aufruf wird empfohlen, wenn mit hoher Abtastrate viele, aber immer gleiche Messwerte ausgegeben werden sollen.

**1. Aufruf:** 2Fh<sub>11h</sub> [<anzahl>, <channel><sup>2 x <anzahl></sup>]

<anzahl> Anzahl der abgefragten Kanäle

<channel><sup>2</sup> gibt die Kanalnummern an

**Ab 2. Aufruf:** 2Fh<sub>11h</sub> [00h]

**Antwort:** 2Fh<sub>11h</sub> <status> [<anzahl>, {<sub-len>, <ch-status>, <channel><sup>2</sup>, <type>, <value><sup>n</sup>} <anzahl>]

**Antwortzeit:** lang

- 
- <anzahl> Anzahl der beim 1. Aufruf abgefragten Kanäle
- <sub-len> gibt an, wie viele Bytes in diesem Sub-Telegramm noch folgen; wenn das nachfolgende Status-Byte z.B. ‚Value Overflow‘ anzeigt, dann entfallen <type> und <value><sup>n</sup> und der nächste Kanal folgt
- <type> gibt den Datentyp der Ausgabe an; davon hängt die Länge von <value> ab (siehe Seite 12 - Datentypen)
- <value><sup>n</sup> abgefragter Wert

**Bemerkung:** In der Gerätebeschreibung werden Kanal und Format für jeden Messwert spezifiziert. Das <sub-len> Feld sollte in jedem Fall ausgewertet werden, auch wenn der Kanal-Status nicht 00h ist, um den korrekten Offset zum nächsten Kanal sicherzustellen.

**ACHTUNG!!** Es können maximal 20 Kanäle abgefragt werden, sofern die Länge der Antwort 210 Byte nicht überschreitet. Beim Datentyp Double können also höchstens 16 Kanäle in einer Abfrage bearbeitet werden.

**ACHTUNG!!** Bei rechenintensiven Kanälen, wie z.B. die vektorielle Mittelwertbildung für Wind im ANACON, kann die Antwortzeit ‚lang‘ unter Umständen nicht für die Ausgabe von mehreren Kanälen ausreichen. Antwortet der Sensor nicht auf die Anfrage, muss die Anzahl der Kanäle oder die Anzahl der Werte in der Mittelwert-bildung reduziert werden!!

---

### 3.8.9 Offlinedatenabfrage (24h)

Kommando <cmd>: 24h (NBC)

Kommandoversion <verc>: 1.0

derzeit nicht spezifiziert!!

---

---

### 3.8.10 Reset / Reset auf Default (25h)

**Kommando <cmd>:** 25h (BC)

**Kommandoversion <verc>:** 1.0

**Daten <payload>:** <reset>

**Beschreibung:** Mit dem Kommando wird ein Software-Reset ausgelöst. Alternativ kann vor dem Reset ein bestimmter Zustand wiederhergestellt werden.

**Aufruf:** 25h<sub>10h</sub>[<reset>]

<reset> 10h Gerät neu starten. Bei Linux-basierten Systemen wird nur die Anwendung neu gestartet (vgl. Reset Kommando 14h).  
11h Auslieferungszustand wiederherstellen + Reset 10h  
12h Geräte-ID auf Auslieferungszustand wiederherstellen + Reset 10h  
13h Gerätespezifisches Reset Kommando (Details sind in der jeweiligen Betriebsanleitung zu finden)  
14h Bei Linux-basierten Systemen wird das komplette System neu gestartet.

**Antwort:** 25h<sub>10h</sub><status>

**Antwortzeit:** bis zu 4 x lang!

**Bemerkung:** Die Antwort erfolgt unmittelbar vor dem Reset.

---

### 3.8.11 Reset mit Verzögerung (2Eh)

**Kommando <cmd>:** 2Eh (BC)

**Kommandoversion <verc>:** 1.0

**Daten <payload>:** <delay>

**Beschreibung:** Mit dem Kommando wird ein Software-Reset nach Ablauf der Verzögerungszeit <delay> ausgelöst (z.B. für Firmware-Update).

**Aufruf:** 2Eh<sub>10h</sub>[<delay>]

<delay> Verzögerungszeit in Sekunden (max. 255)

**Antwort:** 2Eh<sub>10h</sub><status>

**Bemerkung:** Die Antwort erfolgt zu Beginn der Verzögerungszeit.

---

### 3.8.12 Statusabfrage (26h)

**Kommando <cmd>:** 26h (NBC)

**Kommandoversion <verc>:** 1.0

**Daten <payload>:** keine

**Beschreibung:** Auslesen des aktuellen Status- bzw. Errorcodes; damit lässt sich das Gerät abfragen, ob es fehlerfrei funktioniert.

**Aufruf:** 26h<sub>10h</sub>[ ]

**Antwort:** 26h<sub>10h</sub><status>[<device-status>]

---

---

### 3.8.13 Letzte Fehlermeldung (2Ch)

**Kommando <cmd>:** 2Ch (NBC)

**Kommandoversion <verc>:** 1.0

**Daten <payload>:** keine

**Beschreibung:** gibt den Errorcode der letzten Antwort des Gerätes in Bezug auf die Kommunikation aus. z.B. ungültige Parameter

**Aufruf:** 2Ch<sub>10h</sub>[ ]

**Antwort:** 2Ch<sub>10h</sub><status>[<error>]

---

### 3.8.14 Uhrzeit / Datum setzen (27h)

**Kommando <cmd>:** 27h (BC)

**Kommandoversion <verc>:** 1.0

**Daten <payload>:** <unixtime><sup>4</sup>

**Beschreibung:** setzt Datum und Uhrzeit des angesprochenen Gerätes.

**Aufruf:** 27h<sub>10h</sub>[<unixtime><sup>4</sup>]

**Antwort:** 27h<sub>10h</sub><status>

---

**Bemerkung:** Unixtime ist die 4-Byte Hexadezimalzahl mit dem niederwertigsten Byte (LSB) zuerst, die den Sekunden seit dem 1.1.1970 0:00 UTC entspricht.

---

### 3.8.15 Uhrzeit / Datum auslesen (28h)

**Kommando <cmd>:** 28h (NBC)

**Kommandoversion <verc>:** 1.0

**Daten <payload>:** keine

**Beschreibung:** auslesen von Datum und Uhrzeit des angesprochenen Gerätes.

**Aufruf:** 28h<sub>10h</sub>[ ]

**Antwort:** 28h<sub>10h</sub><status>[<unixtime><sup>4</sup>]

---

**Bemerkung:** Unixtime ist die 4-Byte Hexadezimalzahl mit dem niederwertigsten Byte (LSB) zuerst, die den Sekunden seit dem 1.1.1970 0:00 UTC entspricht.

---

---

### 3.8.16 Test- / Abgleichbefehl (29h)

**Kommando <cmd>:** 29h (NBC)

**Kommandoversion <verc>:** 1.0

**Daten <payload>:** <pin><sup>2</sup>, <function>, <data><sup>n</sup>

**Beschreibung:** dieses Kommando dient zum Abgleich und zum Test des Geräts

**Aufruf:** 29h<sub>10h</sub>[<pin><sup>2</sup>, <function>, <data><sup>n</sup>]

**Antwort:** 29h<sub>10h</sub><status>[..., ...] (Gerätespezifisch!!)

**Antwortzeit:** bis zu 4 x lang!

**ACHTUNG!!** Dieses Kommando ist ausschließlich für den internen Gebrauch bei OTT HydroMet für die Werksprüfung vorgesehen! Bei unsachgemäßem Gebrauch kann das Gerät unbrauchbar gemacht werden! (siehe auch Seite 6 Einschränkung der Garantie!!) Die Testfunktionen sind in der Gerätebeschreibung enthalten.

---

### 3.8.17 Monitor (2Ah)

**Kommando <cmd>:** 2Ah (NBC)

**Kommandoversion <verc>:** 1.0

**Daten <payload>:** <monitor><sup>n</sup>

**Beschreibung:** Mit Hilfe von Monitorbefehlen lassen sich mit der PC-Software gerätespezifische Funktionen ausführen (siehe jeweilige Gerätebeschreibung).

**Aufruf:** 2Ah<sub>10h</sub>[<monitor-befehl><sup>n</sup>]

**Antwort:** 2Ah<sub>10h</sub><status>[<antwort><sup>n</sup>]

**Antwortzeit:** lang

**ACHTUNG!!** Dieses Kommando ist ausschließlich für den internen Gebrauch bei OTT HydroMet vorgesehen! Bei unsachgemäßem Gebrauch kann das Gerät unbrauchbar gemacht oder zerstört werden. Dieses Kommando ist in der Gerätebeschreibung spezifiziert.

---

### 3.8.18 Protokollwechsel (2Bh)

**Kommando <cmd>:** 2Bh (BC)

**Kommandoversion <verc>:** 1.0

**Daten <payload>:** <type>

**Beschreibung:** schaltet das Gerät temporär auf ein anderes Protokoll um.

**Aufruf:** 2Bh<sub>10h</sub>[<type>]

<type>	00h	UMB-Binär
	01h	UMB-ASCII
	02h	Terminal Protokoll
	03h	SDI-12
	05h	MODBUS-RTU
	06h	MODBUS-ASCII
	07h	XDR (NMEA)
	09h	UMB-ASCII 2.0
	0Ah	Wippen-Simulation (nur interner Gebrauch)
	0Bh	MODBUS-TCP
	0Ch	High-Speed-Modus (nur interner Gebrauch)
	10h	UMB-ASCII (aus Kompatibilitätsgründen)

**Antwort:** 2Bh<sub>10h</sub><status>

**ACHTUNG!!** Unmittelbar nach der Antwort kann das Gerät nur noch im neuen Protokoll angesprochen werden. Soll das Gerät wieder z.B. im Binär-Mode arbeiten, muss mit dem entsprechenden Kommando für Protokollwechsel in den Binär-Mode geschaltet werden!

Die Protokollumschaltung ist **temporär!!** Nach einem Reset oder einem gerätespezifischen Timeout kommuniziert das Gerät wieder in dem zuvor eingestellten Mode! Soll das Gerät dauerhaft im z.B. XDR-Mode betrieben werden, muss die Gerätekonfiguration im EEPROM geändert werden!

Es werden nicht alle genannten Protokolle von allen Geräten unterstützt. Genaue Informationen sind der Gerätebeschreibung zu entnehmen.

### 3.8.19 neue Geräte-ID setzen (30h)

**Kommando <cmd>:** 30h (BC)

**Kommandoversion <verc>:** 1.0

**Daten <payload>:** <ID><sup>2</sup>

**Beschreibung:** gibt dem Gerät **dauerhaft** eine neue ID.

**Aufruf:** 30h<sub>10h</sub>[<ID><sup>2</sup>]

<ID><sup>2</sup> neue Geräte-ID (1 – 255)

**Antwort:** 30h<sub>10h</sub><status>

**ACHTUNG!!** Unmittelbar nach der Antwort erfolgt ein Reset und danach kann das Gerät nur noch mit der neuen ID angesprochen werden. Achtung! Dieses Kommando ist broadcast-fähig!! Dadurch lassen sich Geräte, deren ID unbekannt ist, mit einer neuen ID versehen. Dies macht jedoch nur Sinn, wenn maximal ein Gerät am Bus angeschlossen ist.

---

**Kommandoversion <verc>:** 1.1

**Daten <payload>:** <ID><sup>2</sup>

**Beschreibung:** gibt dem Gerät **temporär** bis zum nächste Reset eine neue ID.

**Aufruf:** 30h<sub>11h</sub>[<ID><sup>2</sup>]

<ID><sup>2</sup> neue Geräte-ID (1 – 255)

**Antwort:** 30h<sub>11h</sub><status>

**ACHTUNG!!** danach kann das Gerät nur noch mit der neuen ID angesprochen werden, bis zum nächsten Reset. Achtung! Dieses Kommando ist broadcast-fähig!! Dadurch lassen sich Geräte, deren ID unbekannt ist, temporär mit einer neuen ID versehen. Dies macht jedoch nur Sinn, wenn maximal ein Gerät am Bus angeschlossen ist.

---

### 3.8.20 UMB-Tunnel (36h)

**Kommando <cmd>:** 36h (NBC)

**Kommandoversion <verc>:** 1.0

**Daten <payload>:** <int\_bus>,<data><sup>n</sup>

**Beschreibung:** Tunnelt eine Anfrage an einen internen Bus (siehe jeweilige Gerätebeschreibung). Das eigentliche Telegramm des Empfänger-Protokolls wird in<data> versendet. Die Antwort enthält das originale Telegramm in<answer>.

**Aufruf:** 36h<sub>10h</sub>[<int\_bus>,<data><sup>n</sup>]

<int\_bus> geräteinterne Bus-Nummer (0 – 255)

**Antwort:** 36h<sub>10h</sub><status>[<answer><sup>n</sup>]

**Antwortzeit: bis zu 4x lang!**

**ACHTUNG!!** Dieses Kommando ist ausschließlich für den internen Gebrauch bei OTT HydroMet vorgesehen! Bei unsachgemäßem Gebrauch kann das Gerät unbrauchbar gemacht oder zerstört werden.

Dieses Kommando ist in der jeweiligen Gerätebeschreibung spezifiziert.

---

### 3.8.21 Firmware übertragen (37h)

**Kommando <cmd>:** 37h (NBC)

**Kommandoversion <verc>:** 1.0

**Daten <payload>:** <sub-command><parameter><sup>n</sup>

**Beschreibung:** Firmware-Übertragung initiieren bzw. Firmware übertragen. Die Payload Daten sind abhängig vom jeweiligen Subkommando <sub-command>

**Aufruf:** 37h<sub>10h</sub>[<sub-command>, <parameter><sup>n</sup>]

**Antwort:** 37h<sub>10h</sub><status>[<sub-command>, <answer><sup>n</sup>]

**Antwortzeit:** bis zu 4 x lang!

<sub-command>	<parameter>	Beschreibung	<answer>
01h	<FW-size> <sup>4</sup> <FW-version><FW-time> <sup>4</sup> <crc> <sup>2</sup> <reboot><force>	Firmware-Übertragung initiieren	<blocklen> <sup>2</sup>
02h	Keine	Update-Status abfragen	<update-state><error-status><progress>
03h	Keine	Max. Baudrate	<maxBaudrate> <sup>4</sup>
04h	<useHS>	HighSpeed benutzen	Keine
10h	<blocklen> <sup>2</sup> <data-offset> <sup>4</sup> <data> <sup>&lt;blocklen=1...N&gt;</sup>	Blöcke übertragen	Keine

**Bemerkung:** Neue Firmware (Update) installieren durch Einspielen der Binärdaten. Die Firmware wird an das Gerät übertragen und beim nächsten Neustart vom Bootloader installiert.

Zunächst wird durch Übertragen des Subkommandos **01<sub>h</sub>** der Firmware-Übertragungsmodus initiiert. Ist dies erfolgreich, wird das durch den Status OK angezeigt. Das Gerät teilt daraufhin die gewünschte Blocklänge <blocklen><sup>2</sup> mit. Über das <reboot>-Flag kann das Gerät angewiesen werden, nach erfolgreicher Übertragung automatisch einen Reboot durchzuführen. Ein gesetztes <force>-Flag bewirkt, dass die übertragene Firmware in jedem Fall neu aufgespielt wird, unabhängig davon ob die gleiche Firmware-Version bereits installiert ist.

Anschließend werden die gesamten Binärdaten in je <blocklen><sup>2</sup>-Byte langen Blöcken (Subkommando **10<sub>h</sub>**) übertragen bis mit dem letzten Übertragungsblock, dessen Datenfeld natürlich auch kleiner sein darf als <blocklen><sup>2</sup> -Byte, die gesamte Länge der Firmware übertragen wurde. Jedes Frame beinhaltet auch die dazugehörige Offset-Adresse <data-offset> der Daten, um Paketverluste abhandeln zu können. Jeder einzelne Block wird vom Gerät mit Status OK quittiert.

Die Übertragung kann jederzeit durch erneutes Senden des Subkommandos **01<sub>h</sub>** neu begonnen werden. Eine fehlerhafte Übertragung der Daten setzt die Übertragung zurück.

Die Übertragung kann in einer höheren Baudrate stattfinden. Sofern vom Gerät unterstützt, liefert das Subkommando **03<sub>h</sub>** die maximal zulässige Baudrate als uint32 Wert. Mit Subkommando **04<sub>h</sub>** dann der HighSpeed-Modus an- oder abgeschaltet werden. Nach der Übertragung sollte der HighSpeed-Modus wieder abgeschaltet werden. Alternativ wird er vom Gerät nach 5 Sekunden zurückgesetzt, wenn kein gültiges Telegramm empfangen wurde. Der HighSpeed-Modus sollte **vor** dem Senden des Subkommandos **01<sub>h</sub>** aktiviert werden.

Der eigentliche Update-Prozess auf dem Gerät, nach erfolgreichem Upload der neuen Firmware, kann mehr oder weniger Zeit in Anspruch nehmen. Um Details zu dem internen Fortschritt zu erhalten, kann das Subkommando **02<sub>h</sub>** verwendet werden.

<b>&lt;update-state&gt;</b>	00h	Idle
	01h	Uploading
	02h	Flashing. <percentage> gibt den aktuellen Fortschritt an
	FEh	Finished
	FFh	Failed. Der Grund für das fehlgeschlagene Update lassen sich über <error-status> ermitteln (UMB-Fehlerstatus)
<b>&lt;error-status&gt;</b>		Gibt im Fehler-Fall genauere Informationen über den Grund des Fehlers an
<b>&lt;percentage&gt;</b>		Fortschritt während des Flash-Vorgangs

### 3.8.22 Binärdaten übertragen (38h)

**Kommando <cmd>:** 38h (NBC)

**Kommandoversion <verc>:** 1.0

**Daten <payload>:** <sub-command><parameter><sup>n</sup>

**Beschreibung:** Übertragung von Binärdaten, z.B. Grafik-Dateien. Die Payload Daten sind abhängig vom jeweiligen Subkommando <sub-command>

**Aufruf:** 38h<sub>10h</sub>[<sub-command>, <parameter><sup>n</sup>]

**Antwort:** 38h<sub>10h</sub><status>[<sub-command>, <answer><sup>n</sup>]

**Antwortzeit:** bis zu 4 x lang!

<sub-command>	<parameter>	Beschreibung	<answer>
10h	keine	Blocklänge	<blockLen>
11h	<objectNr>	Objekt Information	<locked><objectSize> <sup>4</sup> <maxSize> <sup>4</sup> <checksum> <sup>4</sup> <name> <sup>40</sup> <mimetype> <sup>100</sup>
12h	<objectNr>	Objekt sperren	keine
13h	<objectNr>	Objekt freigeben	keine
14h	<objectNr>	CRC berechnen	keine
20h	<objectNr><data-offset> <sup>4</sup> <length>	Objekt von Sensor übertragen	<length><data> <sup>&lt;length&gt;</sup>
30h	<objectNr><data-offset> <sup>4</sup> <length><data> <sup>&lt;length&gt;</sup>	Objekt an Sensor übertragen	keine

**Bemerkung:** Die Bedeutung der Daten an <objectNr> ist der Gerätebeschreibung zu entnehmen.

Durch Absenden des Subkommandos **10h** wird die maximale Blocklänge abgefragt, die der Sensor verarbeiten kann (max. 204 Bytes). Danach können mit Subkommando **11h** die Informationen des gewünschten Datenobjekts, z.B. die Größe in Bytes ermittelt werden. Ist die Checksumme auf 0 gesetzt, so bedeutet dies, dass keine CRC Berechnung verfügbar ist. Mit dem Subkommando **12h** kann das Datenobjekt vor Veränderung geschützt werden, um während der Übertragung konsistente Daten zu gewährleisten.

Die eigentliche Übertragung der Daten von Sensor zum Controller findet mit Subkommando **20h** statt. Der Controller kann durch Hochzählen des <data-offset><sup>4</sup>-Parameters die Daten in Blöcken abfragen, wobei <data-offset> + <length> die <objectSize> nicht überschreiten sollten. Die fehlerhafte Übertragung eines Blockes kann durch erneute Abfrage mit entsprechendem Offset korrigiert werden.

Mit Subkommando **30<sub>h</sub>** können auf ähnlich Weise Daten an den Sensor übertragen werden. Dieses Subkommando wird nicht bei allen Objekten unterstützt.

Optional kann mit Subkommando **14<sub>h</sub>** die Berechnung der CRC-32 Checksumme des Datenobjekts angestoßen werden, die über die Objekt-Informationen **11<sub>h</sub>** ausgelesen werden kann. Da diese Berechnung u.U. einige Zeit benötigt liefert das Subkommando **11<sub>h</sub>** währenddessen den Status *Busy* (**28<sub>h</sub>**).

**ACHTUNG!!** Nach der Übertragung muss das Datenobjekt unbedingt wieder mit Subkommando **13<sub>h</sub>** entsperrt werden! Andernfalls kann die korrekte Funktion des Sensors nicht sichergestellt werden.

### 3.9 Kanalbelegungen

Es sind maximal 65535 Messkanäle adressierbar. Die Kanalbelegung gilt für die Onlinedatenabfrage im Binärprotokoll. Die Kanalbelegung der einzelnen UMB-Geräte ist der entsprechenden Betriebsanleitung zu entnehmen.

#### 3.9.1 Kanalbelegung allgemeine Zuordnung

Um die Messgröße leichter zu identifizieren, wird folgende Zuordnung der Kanäle empfohlen:

Kanal	Messgröße
0 – 99	reserviert
100 – 199	Temperatur
200 – 299	Feuchte
300 – 399	Druck (z.B. Luft)
400 – 499	Geschwindigkeit (z.B. Wind, Strömung)
500 – 599	Richtung (z.B. Wind)
600 – 699	metrische Werte (z.B. Wasserfilmhöhe in mm, Sichtweite in m)
700 – 799	logische Zustände (z.B. Türkontakt 0 / 1 = auf / zu)
800 – 899	relative Messwerte (z.B. Salzkonzentration)
900 – 999	z.B. Fahrbahnzustand
1000 – 1999	TLS-Kodierungen (siehe auch Seite 38 Kapitel 0)
2000 – 2999	TLS-Kodierungen für 2. Kanal (z.B. ANACON)
4000 – 4999	Diagnose und Service (für Kunde)
10000 – 10099	Spannung
10100 – 10199	Strom
10200 – 10299	Widerstand
10300 – 10399	Frequenz
10400 – 10499	Kapazität
10500 – 10599	Impulse
20000 – 29999	gerätespezifisch
65535	reserviert

#### 3.9.2 TLS-Kanalbelegung

Diese Kanäle sind für die Ausgabe von Daten entsprechend der TLS-Typen nach TLS2002 DE-FG3 (Wetter- und Umfeld-Daten) vorgesehen. Die Kanalnummern entsprechen mit einem Offset von 1000 den DE-Typen FG3 (siehe auch Daten-Typen in UMB-Produkten nach TLS2002 FG3, Seite 38).

z.B. Sichtweite:

FG3 DE-Typ 60      Ergebnismeldung Sichtweite SW      Kanal 1060

### 3.10 Einheitenliste

Für alle UMB-Produkte werden folgende Einheiten für Messwerte verwendet. Diese werden u.a. bei der Ausgabe der Geräteinformation ausgegeben.

#### 3.10.1 Temperatur

Einheit	Beschreibung	Bemerkung
°C	Grad Celsius	
°F	Grad Fahrenheit	
K	Kelvin	

#### 3.10.2 Feuchte

Einheit	Beschreibung	Bemerkung
%rH	relative Feuchte	
g/kg	absolute Feuchte	
g/m <sup>3</sup>	absolute Feuchte	

#### 3.10.3 Längen

Einheit	Beschreibung	Bemerkung
µm	Mikrometer	
mm	Millimeter	
cm	Zentimeter	
dm	Dezimeter	
m	Meter	
km	Kilometer	
in	Zoll (Inch)	1 Zoll = 25,4 mm
mil	Milli-Inch	1 mil = 1/1000 inch = 0,0254 mm
ft	Fuß (foot, feet)	1 foot = 0,3048 Meter
mi	Meile	1 Statute Mile der USA = 1,609344 km

#### 3.10.4 Geschwindigkeiten

Einheit	Beschreibung	Bemerkung
m/s	Meter pro Sekunde	
km/h	Kilometer pro Stunde	
mph	miles per hour	Eine mph entspricht 1,609344 km/h oder 0,44704 m/s
kts	Knoten	1 Knoten = 1 Seemeile/Stunde = 1,852 km/h = 0,51444 m/s

#### 3.10.5 Elektrische Größen

Einheit	Beschreibung	Bemerkung
µV	Microvolt	
mV	Millivolt	
V	Volt	
nA	Nanoampere	
mA	Milliampere	
A	Ampere	
Ohm	Ohm	

#### 3.10.6 Frequenz

Einheit	Beschreibung	Bemerkung
Hz	Hertz	
kHz	Kilohertz	
MHz	Megahertz	
GHz	Gigahertz	

### 3.10.7 Druck

Einheit	Beschreibung	Bemerkung
bar	Bar	
mbar	Millibar	
Pa	Pascal	
mPa	Millipascal	
hPa	Hektopascal	
inHg	Inch Quecksilbersäule	

### 3.10.8 Volumen

Einheit	Beschreibung	Bemerkung
m <sup>3</sup>	Kubikmeter	
µl	Mikroliter	
l	Liter	

### 3.10.9 Zeit

Einheit	Beschreibung	Bemerkung
h	Stunden	
s	Sekunden	
ms	Millisekunden	
µs	Mikrosekunden	

### 3.10.10 Niederschlag

Einheit	Beschreibung	Bemerkung
mm/m	Millimeter pro Minute	Niederschlagsintensität
mm/h	Millimeter pro Stunde	Niederschlagsintensität
mm/d	Millimeter pro Tag	
l/m <sup>2</sup>	Liter pro Quadratmeter	Niederschlagsmenge
l/m <sup>2</sup> /h	Liter pro Quadratmeter pro Stunde	Niederschlagsintensität
in/m	Zoll (Inch) pro Minute	Niederschlagsintensität
in/h	Zoll (Inch) pro Stunde	Niederschlagsintensität
in/d	Zoll (Inch) pro Tag	Niederschlagsintensität
mil/h	milli-Inch pro Stunde	Niederschlagsintensität

### 3.10.11 Sonstige

Einheit	Beschreibung	Bemerkung
%	Prozent	relativer Anteil
°	Grad	Winkelangabe
kJ/kg	Kilojoule pro Kilogramm	
kg/m <sup>3</sup>	Kilogramm pro Kubikmeter	
Byte	Byte	
kByte	Kilobyte	
MByte	Megabyte	
GByte	Gigabyte	
logic		Logischer Zustand z.B. Straßenzustand, Niederschlagsart
events	Anzahl Events	z. B. measurement counter
norm value	normierter Wert	0 ... 65520
raw	Rohwerte	
digits		
count	Zählerwert	
rpm	Umdrehungen pro Minute	Lüfter-Drehzahl
lx	Lux	
klx	Kilolux	
W/m <sup>2</sup>	Watt pro Quadratmeter	Globalstrahlung
dB	Dezibel (Pegel)	
dBm	Leistungspegel	Bluetooth, WLAN
IPv4	IPv4 Adresse	IP-Adresse als 4 Byte Rohwert

Zeichensatz nach ISO-8859-1 (Latin-1) codiert. Siehe auch Seite 42.

### 3.11 Beispiel einer Binärprotokoll-Abfrage

Soll z.B. ein Wettersensor mit der Geräte-ID 1 nach der aktuellen Temperatur (Kanal 100) von einem PC abgefragt werden, geschieht das wie folgt:

**Sensor:**

Klassen-ID für **Wettersensor** ist 7d = 0 7h

Geräte-ID ist 1d = 01h

Setzt man die Klassen- und Geräte-ID zusammen ergibt sich eine Ziel-Adresse 7001h

**PC:**

Klassen-ID für **PC (Controller)** ist 15d = 0 Fh

PC-ID ist 1 = 01h

Setzt man die Klassen- und PC-ID zusammen ergibt sich eine Absender-Adresse F001h

Die Länge <len> beträgt für den Befehl Onlinedatenabfrage ist 4d = 04h, da der Befehl nur aus 4 Byte besteht

Das Kommando für Onlinedatenabfrage ist 23h

Die Versionsnummer des Befehls ist 1.0 = 10h

Der Befehl hat die Kanalnummer (100d) als Payload: 0064h

Die CRC beträgt 61D9h

**Die komplette Anfrage an das Gerät:**

SOH	<ver>	<to>		<from>		<len>	STX	<cmd>	<verc>	<channel>		ETX	<cs>		EOT
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
01h	10h	01h	70h	01h	F0h	04h	02h	23h	10h	64h	00h	03h	61h	D9h	04h

**Die komplette Antwort des Gerätes:**

SOH	<ver>	<to>		<from>		<len>	STX	<cmd>	<verc>	<status>
1	2	3	4	5	6	7	8	9	10	11
01h	10h	01h	F0h	01h	70h	0Ah	02h	23h	10h	00h

<channel>		<type>	<value>				ETX	<cs>		EOT
12	13	14	15	16	17	18	19	20	21	22
64h	00h	16h	F5h	54h	E1h	41h	03h	90h	86h	04h

Die aktuelle Temperatur ist also 28,16 °C und wird als IEEE-Float übertragen.

Mit Hilfe der Checksumme (8690h) kann die korrekte Datenübertragung überprüft werden.

**ACHTUNG!!** Bei der Übertragung von Word-Variablen wie z.B. der Geräteadressen gilt Little Endian (Intel, lowbyte first). Das bedeutet erst das LowByte und dann das HighByte.

### 3.12 Bemerkungen zu Broadcast

Wird ein Gerät direkt mit Klassen- und Geräte-ID angesprochen, wird die im Kommando beschriebene Antwort zurückgegeben.

Wird ein Gerät mit Broadcast (Geräte-ID ,0' oder Klassen- und Geräte-ID ,0') angesprochen, wird auf das Kommando **NICHT** geantwortet, da bei Broadcast davon ausgegangen werden muss, dass mehrere Geräte gleichzeitig angesprochen werden und es sonst zu Kollisionen kommen würde.

Nicht alle Kommandos sind broadcastfähig, da es z.B. keinen Sinn macht eine Messwertabfrage an alle Geräte zu schicken da diese bei Broadcast nicht antworten. Ob ein Kommando broadcastfähig ist, wird durch ,BC' gekennzeichnet. ,NBC' steht für nicht broadcastfähig.

Eine sinnvolle Anwendung von Broadcast-Kommandos ist z.B. das Stellen von Datum / Uhrzeit. Somit kann mit einem Telegramm das komplette Netz aktualisiert werden.

## 4 Anhang

### 4.1 CRC-Berechnung

Berechnung der CRC erfolgt nach den folgenden Regeln:

Norm: CRC16-MCRF4XX  
 Polynom:  $1021h = x^{16} + x^{12} + x^5 + 1$  (LSB-first Mode)  
 Startwert: FFFFh

**ACHTUNG!!** Im Gegensatz zu früheren Luft-Protokollen ist hier der Startwert für die CRC-Berechnungen nicht 0h, sondern FFFFh nach CCITT!

#### 4.1.1 Beispiel einer CRC16-MCRF4XX -Berechnung in C

Soll die CRC-Berechnung für mehrere Bytes erfolgen, muss die bisher berechnete CRC in einer unsigned short-Variable (die zu Beginn einer Prüfreihe auf FFFFh initialisiert werden muss) zwischengespeichert werden.

```

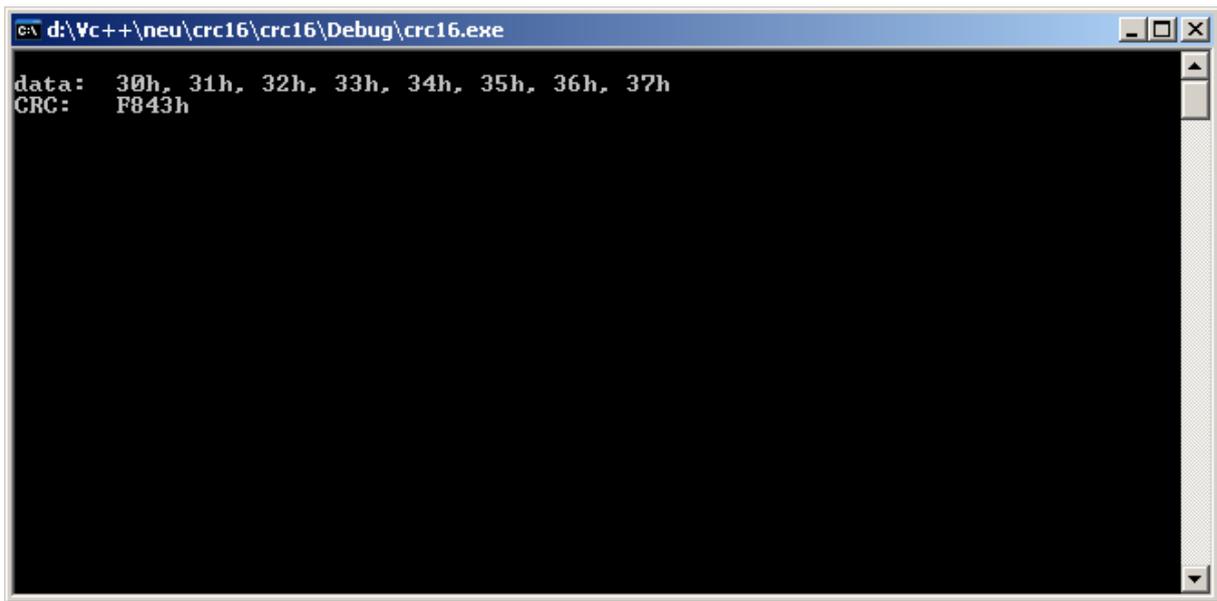
/*****
* DESCRIPTION:      UMB Crc (CRC16_MCRF4XX) function
*                  polynomial:      X^16+X^12+X^5+1
*                                  0x8408
*                  Initial Value:   suggested 0xFFFF
*                  Final Xor Value: 0x0 (not applied)!
*                  Example:         See http://www.sunshine2k.de/coding
*                                  /javascript/crc/crc_js.html
*                                  "123456789" --> 0x6F91
*                                  "CRC16_MCRF4XX"
* PARAMETERS:      crc              16-bit start value (recursive CRC value)
*                  *pData           Pointer to data buffer (source buffer)
*                  length           Length of data [bytes]
*****/
uint16 calcCrc16(uint16 crc, const uint8* pData, uint32 length)
{
    while(length)
    {
        uint8 byte = *pData;
        for(uint8 b = 0; b < 8; b++)
        {
            crc ^= (byte & 0x01u);
            crc = (crc & 0x01u) ? ((crc >> 1) ^ 0x8408u) : (crc >> 1);
            byte >>= 1;
        }

        pData++;
        length--;
    }
    return crc;
}

```

```
// ***** MAIN *****
void main(void)
{
    // example: CRC for 8 Bytes
    unsigned char values[8] =
        {0x30, 0x31, 0x32, 0x33, 0x34, 0x35, 0x36, 0x37};
    // calculation with start value FFFFh
    unsigned short crc = calcCrc16(0xFFFF, values, sizeof(values));
    // output
    printf("\ndata:  30h, 31h, 32h, 33h, 34h, 35h, 36h, 37h");
    printf("\nCRC:   %04Xh\n", crc);
}
```

Output:



## 4.2 Automatisches auslesen eines Netzwerks

In diesem Abschnitt wird ein Mechanismus beschrieben, der es ermöglicht ein bestehendes Netzwerk zu analysieren, um damit die Controller-Software zu konfigurieren.

### 4.2.1 Hintergrund

Da es sich um ein halbduplexes Netz auf RS485-Basis ohne Kollisionserkennung handelt, muss das Controller-Device-Prinzip eingehalten werden. Um ein Netzwerk zu scannen, müsste der Controller den gesamten Adressraum abfragen, was bei über 30000 möglichen Adressen zu lang dauern würde.

Stattdessen wird das System wie folgt Weise konfiguriert, damit die Controller-Software in kurzer Zeit das Netz scannen kann.

### 4.2.2 Notwendige ID-Konfiguration der Sensoren

Die Sensoren werden je Netzwerk und Geräteklasse mit bei 1 beginnenden Geräte-IDs versehen. Das entspricht auch dem Auslieferungszustand. Weitere Sensoren einer Geräteklasse werden mit aufsteigenden IDs (2, 3, 4, 5 ...) versehen.

Beispiel:

Sensoren	Klassen-ID	empfohlene Geräte-ID
1. Straßensensor	1	1
2. Straßensensor	1	2
1. Regensensor	2	1
2. Regensensor	2	2
1. Sichtweite	3	1
2. Sichtweite	3	2
1. Temperatur-Feuchte	4	1
2. Temperatur-Feuchte	4	2
3. Temperatur-Feuchte	4	3
1. NIRS	5	1

Da die unterschiedlichen Sensoren unterschiedliche Klassen IDs haben und sich die Adresse aus Klassen-ID und Geräte-ID zusammensetzt, hat jeder Teilnehmer eine eigene Adresse.

### 4.2.3 Scannen des Netzwerkes

Beim Scannen beginnt der Controller die Sensoren mit jeweils aufsteigender Klassen- und Geräte-ID abzufragen. Dazu verwendet man ein Kommando, welches von jedem Sensor verstanden wird, z.B. Statusabfrage (26h).

Die Geräte-ID wird so lange erhöht, bis auf die Statusabfrage keine Antwort mehr empfangen wird. Dann wird die Klassen-ID erhöht und wieder mit Geräte-ID 1 begonnen.

### 4.3 Daten-Typen in UMB-Produkten nach TLS2002 FG3

Die Ausgabe von TLS-Daten orientiert sich an der DE-Block-Struktur. Die TLS-Ausgabe beschränkt sich auf den TLS-konforme Daten-Normierung nach FG3. Die Antwort auf eine Messwertabfrage von TLS-Daten enthält den UMB-Kanal und den Messwert. Die UMB-Kanäle erhalten einen Offset von +1000 gegenüber dem DE-Typ FG3. Bei mehrkanaligen Geräten erhöht sich der Offset um jeweils 1000.

8-Bit-Messwert:

Position	Bezeichnung	Erläuterung
Byte 1	Messwert	

16-Bit-Messwert:

Position	Bezeichnung	Erläuterung
Byte 1	Messwert	low Byte
Byte 2	Messwert	high Byte

#### 4.3.1 Beispiel einer TLS-Messwertabfrage

Es soll z.B. ein Sichtweitenmessgerät die Sichtweite nach TLS ausgeben (16-Bit-Messwert). DE-Typ 60 (SW) ergibt UMB-Kanal 1060 = 0424h

**Aufruf:** 23h<sub>10h</sub>[<channel><sup>2</sup>]  
23h<sub>10h</sub>[24h, 04h]

**Antwort:** 23h<sub>10h</sub><status>[<channel><sup>2</sup>, <low byte>, <high byte>]  
23h<sub>10h</sub><status>[24h, 04h, E8h, 03h]

High Byte = 03h; Low Byte = E8h; ergibt 03E8h = 1000d = 1000 Meter Sichtweite

#### 4.3.2 unterstützte TLS-DE-Typen FG3

Die folgenden Beispiele illustrieren die Zuordnung der TLS Typen zu UMB Kanälen. In den Gerätehandbüchern ist beschrieben welche Typen die einzelnen Geräte unterstützen.

DE-Typ	UMB-Kanal	Bedeutung	Format	Bereich	Auflösung	Codierung
49	1049	Ergebnismeldung Fahrbahnoberflächen- temperatur FBT	16 Bit	-30...+80°C	0,1°C	80,0 = 800d = 0320h 0,0 = 0d = 0000h -0,1 = -1d = FFFFh -30,0 = -300d = FED4h
52	1052	Ergebnismeldung Restsalz RS	8 Bit	0%...100%	1%	0% = 0d = 00h 100% = 100d = 64h FFh = nicht bestimmbar
65	1065	Ergebnismeldung Gefrier- temperatur GT	16 Bit	-30...0 °C	0,1°C	0,0 = 0d = 0000h -0,1 = -1d = FFFFh -30,0 = -300d = FED4h

### 4.3.3 Abgeleitete Größen

1153	float	Niederschlagsintensität in Inch/h Abgeleitet aus Kanal 1053 (TLS-Code DE Typ 53 FG3)	0.....7.874 Inch/h
1253	float	Niederschlagsintensität in mil/h Abgeleitet aus Kanal 1053 (TLS-Code DE Typ 53 FG3)	0.....7 874 mil/h

### 4.3.4 DE-Typ 70 „Zustand der Fahrbahnoberfläche“ (FBZ)

Inhalt, bzw. Ausprägung	Definition
0	Fahrbahn ist vollkommen trocken (< ca. 30ml/m <sup>2</sup> = 0,03 mm), schnee- und eisfrei
1	Fahrbahn ist deucht, bzw. nass, oder schnee- oder eisbedeckt. Die Benetzung, bzw. Bedeckung übersteigt ca. 30 ml/m <sup>2</sup> = 0,03 mm. Nähere Differenzierung der Bedeckungsart nicht möglich.
2 ... 31	Frei für Erweiterungen
32	Fahrbahn ist benetzt mit flüssigem Wasser, bzw. wässriger Lösung. Die Menge übersteigt ca. 30 ml/m <sup>2</sup> = 0,03mm. Nähere Differenzierung nicht möglich.
33 ... 63	Frei für Erweiterungen
64	Fahrbahn ist bedeck mit gefrorenem Wasser, bzw. wässriger Lösung in festem Zustand. Weitere Differenzierung nicht möglich.
65	Fahrbahn ist bedeck mit Schnee oder Schneematsch. Gemisch von flüssigem und gefrorenem Wasser, bzw. wässriger Lösung.
66	Fahrbahn ist bedeckt mit Eis (festes, gefrorenes Wasser, bzw. gefrorene wässrige Lösung)
67	Fahrbahn ist bedeckt mit Raureif. Aus der Luft sublimierte Eiskristalle ohne deckende Eisfläche. Die Taupunkttemperatur liegt nahe der Fahrbahnoberflächen-Temperatur und liegt unter der Gefriertemperatur.
68 ... 127	Frei für Erweiterungen
128 ... 254	Frei für hersteller- oder anwendungsspezifische Codes
255	Sensorik kann auf Grund der herrschenden Bedingungen Zustand nicht bestimmen

Es ist zu beachten, dass in Relation zur Strecke die Messung immer nur punktförmig sein kann und daher die Ergebnisse entsprechend interpretiert werden müssen.

Die Benetzung bzw. Bedeckung wird immer bezogen auf eine glatte, ebene Fläche. Die Einschätzung der Gefährlichkeit einer Benetzung oder Bedeckung in Bezug auf die Beschaffenheit des Fahrbahnbelags (Rauheit, etc.) und den besonderen Bedingungen der betreffenden Straßendecke muss in der Zentrale vorgenommen werden.

Es ist nicht erforderlich, dass die Sensortechnik alle Zustände direkt als solches detektieren kann. Es genügt vielmehr, wenn die zur Bildung der Zustände beitragenden Einflüsse gemessen werden können, so dass der betreffende Zustand als wahrscheinlich angenommen werden kann.

Von einer Fahrbahnmessstelle müssen nicht alle Ausprägungen unterstützt werden.

Die Menge der Benetzung, bzw. Bedeckung, soweit bestimmbar, wird durch die „Wasserfilmdicke“ (mm, bzw. l/m<sup>2</sup>) angegeben.

### 4.3.5 DE-Typ 71 „Niederschlagsart“ (NS)

Inhalt, bzw. Ausprägung	Definition
Alle	In der Atmosphäre fallender Niederschlag. Wird gleichzeitig Niederschlagsintensität ermittelt, ist diese an der gleichen Stelle vorzunehmen. Es werden die Klassifikation und die Codes nach WMO Tabelle 4680 verwendet.
0	Kein Niederschlag
1 ... 39	Nicht benutzen
40	Niederschlag aller Art Nicht näher klassifizier- und quantifizierbar oder Sensorik nicht dafür ausgelegt
41	Leichter oder mittlerer Niederschlag aller Art (< 50 Partikel/Minute)
42	Starker Niederschlag aller Art (> 50 Partikel/Minute)
43 ... 49	Frei für Erweiterungen
50	Sprühregen (keine weitere Klassifikation von Regen möglich)
51 ... 59	Weitere Klassifikation von Sprühregen n. WMO
60	Regen, bzw. flüssiger Niederschlag (keine weitere Klassifikation von Regen möglich)
61 ... 69	Weitere Klassifikation von Regen n. WMO
70	Schnee, bzw. gefrorener Niederschlag (keine weitere Klassifikation von gefrorenem Niederschlag möglich)
71 ... 73	Weitere Klassifikation von Schnee n. WMO
74 ... 76	Weitere Klassifikation von Graupel n. WMO
77 ... 79	Weitere Klassifikation von Hagel n. WMO
80 ... 127	Frei für Erweiterungen
128 ... 254	Frei für hersteller- und anwendungsspezifische Erweiterungen
255	Sensorik kann auf Grund der herrschenden Bedingungen Zustand nicht bestimmen

Es wird darauf hingewiesen, dass von der Streckenstation, bzw. der Sensorik nicht alle Ausprägungen unterstützt werden müssen. Welcher Differenzierungsgrad notwendig und sinnvoll ist, hängt von der Verwendung ab. Bei einfachen Anwendungen können die Ausprägungen 0 und 40, bei normalen Anforderungen 0, 60 und 70 (entspricht den bisher verwendeten Ausprägungen 00, 01, 02 des Typs 63) ausreichen.

### 4.3.6 DE-Typ 140 „Türkontakt“ (TK)

Das sich dieses System auf FG3-Daten beschränkt, der Türkontakt aber in FG6 als DE-Typ 48 definiert ist, wird DE-Typ 140 als Türkontakt-Meldung in FG3 verwendet, wenn diese Meldung als einzige Betriebsmeldung vorhanden sein soll.

DE-Typ	UMB-Kanal	Bedeutung	Format	Bereich	Auflös.	Codierung
140	1140 2140	Betriebsmeldung Türkontakt TK	8 Bit	0 ... 1	1	00 = 0d = 00h 01 = 1d = 01h

Inhalt	Definition	Schalter-Kontakt
0	Tür geschlossen	offen
1	Tür geöffnet	geschlossen

### 4.3.7 DE-Typ 140 „Türkontakt“ (TK) invertiert

DE-Typ	UMB-Kanal	Bedeutung	Format	Bereich	Auflös.	Codierung
140	1145 2145	Betriebsmeldung Türkontakt TK invertiert	8 Bit	0 ... 1	1	00 = 0d = 00h 01 = 1d = 01h

Inhalt	Definition	Schalter-Kontakt
0	Tür geschlossen	geschlossen
1	Tür geöffnet	offen

## 4.4 Zeichenketten und Codierung

### 4.4.1 Zeichenketten

Strings innerhalb eines UMB-Frames werden nicht mit \0 terminiert, da sie als Zeichen-Feld mit fester Länge behandelt werden. Alle Zeichenketten sind mit Leerzeichen aufgefüllt.

### 4.4.2 ASCII-Codierung

Die Ausgabe der Zeichen für Textausgaben aller Art erfolgt nach der ASCII-Codetabelle mit der Erweiterung nach ISO-8859-1 (Latin-1), bzw. Windows-1252:

Code	...0	...1	...2	...3	...4	...5	...6	...7	...8	...9	...A	...B	...C	...D	...E	...F
0...	<i>NUL</i>	<i>SOH</i>	<i>STX</i>	<i>ETX</i>	<i>EOT</i>	<i>ENQ</i>	<i>ACK</i>	<i>BEL</i>	<i>BS</i>	<i>HT</i>	<i>LF</i>	<i>VT</i>	<i>FF</i>	<i>CR</i>	<i>SO</i>	<i>SI</i>
1...	<i>DLE</i>	<i>DC1</i>	<i>DC2</i>	<i>DC3</i>	<i>DC4</i>	<i>NAK</i>	<i>SYN</i>	<i>ETB</i>	<i>CAN</i>	<i>EM</i>	<i>SUB</i>	<i>ESC</i>	<i>FS</i>	<i>GS</i>	<i>RS</i>	<i>US</i>
2...	SP	!	"	#	\$	%	&	'	(	)	*	+	,	-	.	/
3...	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
4...	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
5...	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_
6...	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
7...	p	q	r	s	t	u	v	w	x	y	z	{		}	~	<i>DEL</i>
8...	<i>PAD</i>	<i>HOP</i>	<i>BPH</i>	<i>NBH</i>	<i>IND</i>	<i>NEL</i>	<i>SSA</i>	<i>ESA</i>	<i>HTS</i>	<i>HTJ</i>	<i>VTS</i>	<i>PLD</i>	<i>PLU</i>	<i>RI</i>	<i>SS2</i>	<i>SS3</i>
9...	<i>DCS</i>	<i>PU1</i>	<i>PU2</i>	<i>STS</i>	<i>CCH</i>	<i>MW</i>	<i>SPA</i>	<i>EPA</i>	<i>SOS</i>	<i>SGCI</i>	<i>SCI</i>	<i>CSI</i>	<i>ST</i>	<i>OSC</i>	<i>PM</i>	<i>APC</i>
A...	NBSP	ı	ø	£	¤	¥	ı	§	¨	©	ª	«	¬	SHY	®	ˆ
B...	°	±	²	³	´	µ	¶	·	¸	¹	º	»	¼	½	¾	¿
C...	À	Á	Â	Ã	Ä	Å	Æ	Ç	È	É	Ê	Ë	Ì	Í	Î	Ï
D...	Ð	Ñ	Ò	Ó	Ô	Õ	Ö	×	Ø	Ù	Ú	Û	Ü	Ý	Þ	ß
E...	à	á	â	ã	ä	å	æ	ç	è	é	ê	ë	ì	í	î	ï
F...	ð	ñ	ò	ó	ô	õ	ö	÷	ø	ù	ú	û	ü	ý	þ	ÿ