

## 234 Feel ModBus Protokoll

### Inhalt

1	Modbus RTU .....	3
1.1	Geräte Adresse.....	3
1.2	RS485 Settings.....	3
1.3	Telegrammaufbau .....	3
1.3.1	Register lesen.....	3
1.3.2	Antwort.....	3
1.3.3	Antwort im Fehlerfall .....	4
1.4	Datentyp .....	4
2	Function Codes .....	5
2.1	FC1 (0x01) Read Coils.....	6
2.1.1	Anfrage.....	6
2.1.2	Antwort.....	6
2.1.3	Fehlerantwort .....	6
2.1.4	Beispiel.....	6
2.2	FC3(0x03) Read Holding Registers .....	7
2.2.1	Anfrage.....	7
2.2.2	Antwort.....	7
2.2.3	Fehlerantwort .....	7
2.2.4	Beispiel.....	7
2.3	FC4(0x04) Read Input Registers .....	8
2.3.1	Anfrage.....	8
2.3.2	Antwort.....	8
2.3.3	Fehlerantwort .....	8
2.3.4	Beispiel.....	8
2.4	FC5 (0x05) Write Single Coil.....	9
2.4.1	Anfrage.....	9
2.4.2	Antwort.....	9
2.4.3	Fehlerantwort .....	9
2.4.4	Beispiel.....	9
2.5	FC6 (0x06) Write Single Register .....	10
2.5.1	Anfrage.....	10
2.5.2	Antwort.....	10
2.5.3	Fehlerantwort .....	10
2.5.4	Beispiel.....	10
2.6	FC15 (0x0F) Write Multiple Coils .....	11
2.6.1	Anfrage.....	11
2.6.2	Antwort.....	11
2.6.3	Fehlerantwort .....	11
2.6.4	Beispiel.....	11
2.7	FC16 (0x10) Write Multiple registers .....	12
2.7.1	Anfrage.....	12
2.7.2	Antwort.....	12
2.7.3	Fehlerantwort .....	12

2.7.4 Beispiel..... 12

3 Register Maps ..... 13

3.1 Output Coil ..... 13

3.2 Input Registers: ..... 15

3.3 Holding Registers: ..... 18

4 CRC berechnen ..... 26

5 String aufbau ..... 27

5.1 ASCII Table ..... 27

# 1 Modbus RTU

## 1.1 Geräte Adresse

Mit den Geräteadressen wird das Gerät in einem Bussystem beschrieben und unterschieden. Bei jedem Lese – oder Schreibbefehl muss die Geräteadresse mitgesendet werden.

Adresstypen	Adresse
Standard	0x01
Einstellbar	0x01...0xF7
Broadcast	0x00

### Standard

Die Standardadresse ist vom Hersteller eingestellt und sollte auf eine freie Busadresse geändert werden.

### Einstellbar

Neben der Standardadresse (0x01) sind Adressen von 1 bis 247 (0x01...0xF7) frei wählbar.

### Broadcast

Das Gerät reagiert immer auf die Broadcastadresse 0. Mit mehreren angeschlossenen Busteilnehmern dürfen Lesebefehle an die Broadcastadresse nicht genutzt werden.

Ein Schreibbefehl an diese Adresse wird von allen angeschlossenen Geräten übernommen. In diesem Fall wird keine Antwort gesendet. Auf diese Weise geschriebene Register können nur durch Rücklesen überprüft werden.

## 1.2 RS485 Settings

Default Settings:

Baudrate	19200
Daten Bits	8
Parität	Even
Stopp Bits	1

## 1.3 Telegrammaufbau

Ein Telegramm besteht immer aus einem Adressteil, dem Befehl (FC = Funktion Code), dem Datenteil und der Checksumme. Telegrammelemente fixer Länge sind grau hinterlegt. Alle ausgelesenen Daten müssen in der MSB First Formatierung ausgewertet werden.

### 1.3.1 Register lesen

Adresse	Funktion Code	Register Start Adresse		Registerlänge		Checksumme	
		1 Byte	1 Byte	1 Byte	1 Byte	1 Byte	1 Byte
		MSB	LSB	MSB	LSB	LSB	MSB

### 1.3.2 Antwort

Adresse	Funktion Code	Datenlänge	Erstes Datenregister		...		Letztes Datenregister		Checksumme	
			1 Byte	1 Byte	1 Byte	1 Byte	1 Byte	1 Byte	1 Byte	1 Byte
	FC	Byteanzahl	MSB	LSB	MSB	LSB	MSB	LSB	LSB	MSB

### 1.3.3 Antwort im Fehlerfall

Im Fehlerfall wird zum Funktion Code 128 (0x80) dazu addiert und im Datenteil der Fehlercode übertragen.  
Antwort auf Register lesen mit FC3 im Fehlerfall, Telegrammaufbau Slave → Master

Adresse	Funktion Code	Datenlänge	Fehlercode	Checksumme	
				1 Byte	1 Byte
1 Byte	1 Byte	1 Byte	1 Byte	1 Byte	1 Byte
	FC + 0x80	0x01	01, 02, 03 oder 04	LSB	MSB

## 1.4 Datentyp

Bei der Kommunikation mit ModBus RTU müssen beide gegenstellen immer wissen in welchem Datentyp die angefragten oder geschriebenen Daten angehören. Nur so können die Daten richtig interpretiert werden. Außerdem muss man wissen wie lange die Daten sind.

Dabei können eigentlich alle Datentypen verwendet werden. (int8, uint8, int16, uint16, float, [string](#) usw.)

## 2 Function Codes

Die Funktion Codes teilen dem Gerät mit welche Aktion es ausführen soll. Zum Beispiel lesen eines einzelnen Registers.

Die verschiedenen Registertypen, teilen dem Gerät die Registergröße mit. Zudem wird definiert ob der Wert im Register über das ModBus Protokoll geändert werden darf oder nicht.

Function Code [FC]	Registertyp	Zugriff	Größe
<a href="#">1 / 0x01</a>	Singel Output Coil	Lesen	1Bit
<a href="#">3 / 0x03</a>	Singel Holding Register	Lesen	16Bits
<a href="#">4 / 0x04</a>	Mehrere Input Register	Lesen	16Bits
<a href="#">5 / 0x05</a>	Singel Output Coil	Schreiben	1Bit
<a href="#">6 / 0x06</a>	Single Holding Register	Schreiben	16Bits
<a href="#">15 / 0x0F</a>	Mehrere Output Coils	Schreiben	1Bit
<a href="#">16 / 0x10</a>	Mehrere Holding Register	Schreiben	16Bits

## 2.1 FC1 (0x01) Read Coils

Dieser Funktionscode wird verwendet um Output Coils zu lesen.

Es wird dabei immer in Bytes geantwortet. Wenn nur ein Coil ausgelesen wird, antwortet der Slave mit einem Byte. Der Status des ausgelesenen Coils steht im LSB des Bytes. Werden mehrere Coils gleichzeitig gelesen, füllen diese Coils das Byte auf. Wobei das LSB der Registerstartadresse entspricht und das MSB der Startadresse + 7. Werden mehr als 8 Coils gelesen werden die darauffolgenden Bytes nach dem gleichen Schema aufgefüllt.

### 2.1.1 Anfrage

Protokoll Daten	Datenlänge	Wert
Funktionscode	1 Byte	0x01
Startadresse	2 Bytes	0x0000...0xFFFF
Registeranzahl	2 Bytes	0x0001...0x07D0

### 2.1.2 Antwort

Protokoll Daten	Datenlänge	Wert
Funktionscode	1 Byte	0x01
Byteanzahl	1 Byte	N = Anzahl an gesendeten Bytes
Coil Status	n Bytes	n = N oder N+1

### 2.1.3 Fehlerantwort

Protokoll Daten	Datenlänge	Wert	
Funktionscode	1 Byte	0x81	
Fehlercode	1 Byte	Funktionscode nicht unterstützt	0x01
		Startadresse nicht erlaubt	0x02
		Unzulässige Registeranzahl	0x03
		Lesefehler Output Coil	0x04

### 2.1.4 Beispiel

Es sollen die Register 0x03...0x0E gelesen werden.

Anfrage		Antwort	
Protokoll Daten	Daten	Protokoll Daten	Daten
Funktionscode	0x01	Funktionscode	0x01
Startadresse MSB	0x00	Byteanzahl	0x02
Startadresse LSB	0x03	Daten Byte 1 (0x0B...0x03)	0xCF
Registeranzahl MSB	0x00	Daten Byte 2 (0x0E...0x0C)	0x04
Registeranzahl LSB	0x0B		

Für die Register 0x0B bis 0x03 ergibt sich dann eine Bitfolge von 1100 1111 wobei die letzte 1 Register 0x03 ist. Für die Register 0x0E bis 0x0C ergibt sich dann folgendes Bitmuster 0000 0100.

## 2.2 FC3(0x03) Read Holding Registers

Dieser Funktionscode wird verwendet um Holdingregister auszulesen. Dabei entspricht eine Registeradresse immer 2 Byte also 16 Bit. Es werden also auch mindestens 2 Byte als Antwort gesendet. Dabei entspricht das erste Byte dem MSB und das zweite Byte dem LSB.

Werden Daten mit einem kleineren Datentyp als 16 Bit in ein solches Register gespeichert werden trotzdem zwei Byte gesendet. Das erste Byte ist dabei dann leer.

### 2.2.1 Anfrage

Protokoll Daten	Datenlänge	Wert
Funktionscode	1 Byte	0x03
Startadresse	2 Bytes	0x0000...0xFFFF
Registeranzahl	2 Bytes	0x0001...0x007D (125)

### 2.2.2 Antwort

Protokoll Daten	Datenlänge	Wert
Funktionscode	1 Byte	0x03
Byteanzahl	1 Byte	2 x N
Registerinhalt	N x 2 Bytes	

### 2.2.3 Fehlerantwort

Protokoll Daten	Datenlänge	Wert	
Funktionscode	1 Byte	0x83	
Fehlercode	1 Byte	Funktionscode nicht unterstützt	0x01
		Startadresse nicht erlaubt	0x02
		Unzulässige Registeranzahl	0x03
		Lesefehler Holding Register	0x04

### 2.2.4 Beispiel

Es sollen die Register 0x20...0x22 gelesen werden.

Anfrage		Antwort	
Protokoll Daten	Daten	Protokoll Daten	Daten
Funktionscode	0x03	Funktionscode	0x03
Startadresse MSB	0x00	Byteanzahl	0x06
Startadresse LSB	0x20	Registerinhalt MSB 0x20	0x46
Registeranzahl MSB	0x00	Registerinhalt LSB 0x20	0x53
Registeranzahl LSB	0x03	Registerinhalt MSB 0x21	0x4D
		Registerinhalt LSB 0x21	0x20
		Registerinhalt MSB 0x22	0x41
		Registerinhalt LSB 0x22	0x47

So können dann auch ganze Strings gelesen werden. Dabei sind dann in einem Register gleich zwei Ascii-Zeichen gespeichert. In diesem Fall dann jetzt „FSM AG“. Das „F“ und „S“ stehen dabei in Register 0x20 das „M“ und „Leerzeichen“ in Register 0x21 und das „A“ und „G“ in Register 0x22.

## 2.3 FC4(0x04) Read Input Registers

Dieser Funktionscode wird verwendet um Inputregister auszulesen. Dabei entspricht eine Registeradresse immer 2 Byte also 16 Bit. Es werden also auch mindestens 2 Byte als Antwort gesendet. Dabei entspricht das erste Byte dem MSB und das zweite Byte dem LSB.

Werden Daten mit einem kleineren Datentyp als 16 Bit in ein solches Register gespeichert werden trotzdem zwei Byte gesendet. Das erste Byte ist dabei dann leer.

### 2.3.1 Anfrage

Protokoll Daten	Datenlänge	Wert
Funktionscode	1 Byte	0x04
Startadresse	2 Bytes	0x0000...0xFFFF
Registeranzahl	2 Bytes	0x0001...0x007D (125)

### 2.3.2 Antwort

Protokoll Daten	Datenlänge	Wert
Funktionscode	1 Byte	0x04
Byteanzahl	1 Byte	2 x N
Registerinhalt	N x 2 Bytes	

### 2.3.3 Fehlerantwort

Protokoll Daten	Datenlänge	Wert	
Funktionscode	1 Byte	0x84	
Fehlercode	1 Byte	Funktionscode nicht unterstützt	0x01
		Startadresse nicht erlaubt	0x02
		Unzulässige Registeranzahl	0x03
		Lesefehler Input Register	0x04

### 2.3.4 Beispiel

Es soll das Register 16 gelesen werden. Der Inhalt ist als uint16 definiert.

Anfrage		Antwort	
Protokoll Daten	Daten	Protokoll Daten	Daten
Funktionscode	0x04	Funktionscode	0x04
Startadresse MSB	0x00	Byteanzahl	0x02
Startadresse LSB	0x10	Registerinhalt MSB 0x10	0x65
Registeranzahl MSB	0x00	Registerinhalt LSB 0x10	0x53
Registeranzahl LSB	0x01		

Als Antwort bekommen wir 0x65 im ersten Byte des Registers und 0x53 im zweiten Byte. Das muss jetzt zu einem uint16 zusammengeführt werden. Somit erhält man 0x6553 was Dezimal 25939 entspricht.

## 2.4 FC5 (0x05) Write Single Coil

Dieser Funktionscode wird verwendet um einzelne Output Coils zu schreiben.

Es werden dabei immer zwei Bytes gesendet. Es gibt aber nur zwei gültige Daten die gesendet werden können. 0x0000 (OFF/0) und 0xFF00 (ON/1). Andere Daten sind ungültig und werden als Fehler behandelt. Die normale Antwort auf ein erfolgreiches beschreiben des Registers ist ein Echo der gesendeten Anfrage.

### 2.4.1 Anfrage

Protokoll Daten	Datenlänge	Wert
Funktionscode	1 Byte	0x05
Register-Adresse	2 Bytes	0x0000...0xFFFF
Coil Status	2 Bytes	0x0000 oder 0xFF00

### 2.4.2 Antwort

Protokoll Daten	Datenlänge	Wert
Funktionscode	1 Byte	0x05
Register-Adresse	2 Bytes	0x0000...0xFFFF
Coil Status	2 Bytes	0x0000 oder 0xFF00

### 2.4.3 Fehlerantwort

Protokoll Daten	Datenlänge	Wert	
Funktionscode	1 Byte	0x85	
Fehlercode	1 Byte	Funktionscode nicht unterstützt	0x01
		Startadresse nicht erlaubt	0x02
		Unzulässiger Registerinhalt	0x03
		Schreibfehler Output Coil	0x04

### 2.4.4 Beispiel

In das Register 5 soll eine 1 geschrieben bzw. der Status auf ON gesetzt werden.

Anfrage		Antwort	
Protokoll Daten	Daten	Protokoll Daten	Daten
Funktionscode	0x05	Funktionscode	0x05
Register-Adresse MSB	0x00	Register-Adresse MSB	0x00
Register -Adresse LSB	0x04	Register-Adresse LSB	0x04
Coil Status MSB	0xFF	Coil Status MSB	0xFF
Coil Status LSB	0x00	Coil Status LSB	0x00

## 2.5 FC6 (0x06) Write Single Register

Dieser Funktionscode wird verwendet um Holdingregister zu schreiben. Dabei entspricht eine Registeradresse immer 2 Byte also 16 Bit. Es müssen also auch mindestens 2 Byte gesendet werden. Dabei entspricht das erste Byte dem MSB und das zweite Byte dem LSB.

Werden Daten mit einem kleineren Datentyp als 16 Bit in ein solches Register gespeichert werden trotzdem zwei Byte gesendet. Das erste Byte ist dabei dann leer. Die normale Antwort auf ein erfolgreiches beschreiben des Registers ist ein Echo der gesendeten Anfrage.

### 2.5.1 Anfrage

Protokoll Daten	Datenlänge	Wert
Funktionscode	1 Byte	0x06
Startadresse	2 Bytes	0x0000...0xFFFF
Registerinhalt	2 Bytes	0x0000...0xFFFF

### 2.5.2 Antwort

Protokoll Daten	Datenlänge	Wert
Funktionscode	1 Byte	0x06
Registeradresse	2 Bytes	0x0000...0xFFFF
Registerinhalt	2 Bytes	0x0000...0xFFFF

### 2.5.3 Fehlerantwort

Protokoll Daten	Datenlänge	Wert	
Funktionscode	1 Byte	0x86	
Fehlercode	1 Byte	Funktionscode nicht unterstützt	0x01
		Startadresse nicht erlaubt	0x02
		Unzulässiger Registerinhalt	0x03
		Schreibfehler Holding Register	0x04

### 2.5.4 Beispiel

Das Register 4 soll mit dem string „OK“ beschrieben werden. Dafür muss man für das „O“ und das „K“ die Hex-Werte aus einer Ascii-Tabelle ermitteln und die Bytes dann damit beschreiben. Für „O“ und „K“ wären das 0x4F und 0x4B.

Anfrage		Antwort	
Protokoll Daten	Daten	Protokoll Daten	Daten
Funktionscode	0x06	Funktionscode	0x06
Registeradresse MSB	0x00	Registeradresse MSB	0x00
Registeradresse LSB	0x04	Registeradresse LSB	0x04
Registerinhalt MSB	0x4F	Registerinhalt MSB	0x4F
Registerinhalt LSB	0x4B	Registerinhalt LSB	0x4B

## 2.6 FC15 (0x0F) Write Multiple Coils

Dieser Funktionscode wird verwendet um mehrere Output Coils zu schreiben.

Es werden dabei immer ganze Bytes geschrieben. Wie auch schon beim Lesen von Output Coils wird der Status des niedrigsten Registers im LSB des ersten Bytes geschrieben. Danach wird das Byte mit den nächsten 7 Registern aufgefüllt. Ist diese Byte voll wird das nächste Register wieder in das LSB des zweiten Bytes geschrieben.

### 2.6.1 Anfrage

Protokoll Daten	Datenlänge	Wert
Funktionscode	1 Byte	0x0F
Startadresse	2 Bytes	0x0000...0xFFFF
Registeranzahl	2 Bytes	0x0001...0x07B0
Byteanzahl	1 Byte	N
Coil Status	N x 1 Byte	

### 2.6.2 Antwort

Protokoll Daten	Datenlänge	Wert
Funktionscode	1 Byte	0x0F
Startadresse	2 Byte	0x0000...0xFFFF
Registeranzahl	n Bytes	0x0001...0x07B0

### 2.6.3 Fehlerantwort

Protokoll Daten	Datenlänge	Wert	
Funktionscode	1 Byte	0x8F	
Fehlercode	1 Byte	Funktionscode nicht unterstützt	0x01
		Startadresse nicht erlaubt	0x02
		Unzulässige Registeranzahl	0x03
		Schreibfehler mehrerer Output Coils	0x04

### 2.6.4 Beispiel

Es sollen die Register 0x13...0x1E geschrieben werden.

Anfrage		Antwort	
Protokoll Daten	Daten	Protokoll Daten	Daten
Funktionscode	0x0F	Funktionscode	0x0F
Startadresse MSB	0x00	Byteanzahl	0x02
Startadresse LSB	0x13	Daten Byte 1 (0x0B...0x03)	0xCF
Registeranzahl MSB	0x00	Daten Byte 2 (0x0E...0x0C)	0x04
Registeranzahl LSB	0x0B		

Für die Register 0x1B bis 0x13 ergibt sich dann eine Bitfolge von 1100 1111 wobei die letzte 1 Register 0x13 ist. Für die Register 0x1E bis 0x1C ergibt sich dann folgendes Bitmuster 0000 0100.

## 2.7 FC16 (0x10) Write Multiple registers

Dieser Funktionscode wird verwendet um mehrere Holding Register zu schreiben. Das wird besonders für Werte benötigt, deren Datentyp größer als 16bit ist. z.B.: ein int32. Dabei entspricht eine Registeradresse immer 2 Byte also 16 Bit. Es müssen also auch mindestens 2 Byte gesendet werden. Dabei entspricht das erste Byte dem MSB und das zweite Byte dem LSB. Bei erfolgreichem schreiben aller Register wird mit der Startadresse und die Anzahl der beschriebenen Register geantwortet.

### 2.7.1 Anfrage

Protokoll Daten	Datenlänge	Wert
Funktionscode	1 Byte	0x10
Startadresse	2 Bytes	0x0000...0xFFFF
Registeranzahl	2 Bytes	0x0001...0x007B
Byteanzahl	1 Byte	2 x N
Coil Status	N x 2 Bytes	wert

### 2.7.2 Antwort

Protokoll Daten	Datenlänge	Wert
Funktionscode	1 Byte	0x10
Startadresse	2 Byte	0x0000...0xFFFF
Registeranzahl	2 Bytes	0x0001...0x007B

### 2.7.3 Fehlerantwort

Protokoll Daten	Datenlänge	Wert	
Funktionscode	1 Byte	0x90	
Fehlercode	1 Byte	Funktionscode nicht unterstützt	0x01
		Startadresse nicht erlaubt	0x02
		Unzulässige Registeranzahl	0x03
		Schreibfehler mehrere Input Register	0x04

### 2.7.4 Beispiel

Es sollen die Register 5 und 6 geschrieben werden.

Anfrage		Antwort	
Protokoll Daten	Daten	Protokoll Daten	Daten
Funktionscode	0x10	Funktionscode	0x10
Startadresse MSB	0x00	Startadresse MSB	0x00
Startadresse LSB	0x05	Startadresse LSB	0x05
Registeranzahl MSB	0x00	Registeranzahl MSB	0x00
Registeranzahl LSB	0x04	Registeranzahl LSB	0x02
Byteanzahl	0x08		
Registerinhalt MSB 5	0x8D		
Registerinhalt LSB 5	0xFF		
Registerinhalt MSB 6	0x89		
Registerinhalt LSB 6	0x98		

Hier wurden jetzt zwei int32 Werte gesendet. In Register 5 und 6 wurde somit 0x8DFF8998 (-1912632936) geschrieben.

### 3 Register Maps

#### 3.1 Output Coil

Lesen mittels [FC1](#) und schreiben mittels [FC5](#) und [FC15](#).

Temperatursensor		
Register Adresse	Bedeutung/Inhalt	Typ
0	<b>Temperatursensor Abgleich @negativer Amplitude</b> Löst einen Abgleich der negativer Amplitude aus und ändert somit die Steigung der Linearisierungsgeraden zwischen Nullpunkt und negativer Amplitude	Bit
1	<b>Temperatursensor @ 0°C</b> Löst eine Korrektur des Temperatursensors aus um den Offset zu bereinigen „1“ → starten des Vorgangs  Zur Kontrolle ob der Vorgang beendet wurde kann das Register gelesen werden. „0“ → Vorgang abgeschlossen „1“ → Vorgang nicht abgeschlossen	Bit
2	<b>Temperatursensor Abgleich @Positiver Amplitude</b> Löst einen Abgleich der positiven Amplitude aus und ändert somit die Steigung der Linearisierungsgeraden zwischen Nullpunkt und positiver Amplitude	Bit

Feuchtesensor		
Register Adresse	Bedeutung/Inhalt	Typ
3	<b>Feuchtesensor Abgleich @11,3%rH</b> Löst einen Abgleich bei 11,3%rH aus und ändert somit die Steigung der Linearisierungsgeraden	Bit
4	<b>Feuchtesensor Abgleich @75,3%rH</b> Löst einen Abgleich bei 75,3%rH aus und ändert somit die Steigung der Linearisierungsgeraden	Bit

Drucksensor		
Register Adresse	Bedeutung/Inhalt	Typ
5	<b>Drucksensor Abgleich @negativer Amplitude</b> Löst einen Abgleich der negativer Amplitude aus und ändert somit die Steigung der Linearisierungsgeraden zwischen Nullpunkt und negativer Amplitude	Bit
6	<b>Drucksensor nullieren</b> Löst eine Nullpunktkorrektur des Drucksensors aus um den Nullpunktoffset zu bereinigen. „1“ → starten des Nulliervorgangs  Zur Kontrolle ob der Nulliervorgang beendet wurde kann das Register gelesen werden. „0“ → Nulliervorgang abgeschlossen „1“ → Nulliervorgang nicht abgeschlossen	Bit

7	<b>Drucksensor Abgleich @Positiver Amplitude</b> Löst einen Abgleich der positiven Amplitude aus und ändert somit die Steigung der Linearisierungsgeraden zwischen Nullpunkt und positiver Amplitude	Bit
8	<b>Drucksensor automatische Nullierung</b> "0" → Automatische Nullierung aus "1" → Automatische Nullierung ein	Bit

CO2-Sensor		
Register Adresse	Bedeutung/Inhalt	Typ
9	<b>CO2-Sensor Abgleich @Amplitude</b> Löst einen Abgleich der Amplitude aus und ändert somit die Steigung der Linearisierungsgeraden	Bit
10	<b>CO2-Sensor Abgleich @500ppm</b> Löst einen Abgleich der Amplitude aus und ändert somit die Steigung der Linearisierungsgeraden	Bit

Grenzkontakte		
Register Adresse	Bedeutung/Inhalt	Typ
11	<b>Grenzkontakt 1 Aktiv/nicht Aktiv</b> "0" → Grenzkontakt nicht Aktiv "1" → Grenzkontakt Aktiv	Bit
12	<b>Grenzkontakt 2 Aktiv/nicht Aktiv</b> "0" → Grenzkontakt nicht Aktiv "1" → Grenzkontakt Aktiv	Bit
13	<b>Grenzkontakt 1 betätigt/nicht betätigt</b> "0" → Grenzkontakt nicht betätigt "1" → Grenzkontakt betätigt	Bit
14	<b>Grenzkontakt 2 betätigt/nicht betätigt</b> "0" → Grenzkontakt nicht betätigt "1" → Grenzkontakt betätigt	Bit

### 3.2 Input Registers:

Nur lesen mit [FC4](#).

Allgemeine Informationen		
Register Adresse	Bedeutung/Inhalt	Typ
0 -> 5	<b>Firmware Version</b> Gibt die Firmware Version des Sensor im <a href="#">String</a> -Format zurück.	12 x Uint8
6 -> 10	<b>Hardware Version</b> Gibt die Hardware Version des Sensor im <a href="#">String</a> -Format zurück.	10 x Uint8
11 -> 20	<b>Herstellereerienummer</b> Gibt die Herstellereerienummer des Sensor im <a href="#">String</a> -Format zurück.	20 x Uint8

VOC-Sensor		
Register Adresse	Bedeutung/Inhalt	Typ
30 -> 39	<b>VOC-Sensor Typ</b> Gibt den Typ des VOC-Sensors im <a href="#">String</a> -Format zurück.	20 x Uint8
40	<b>Status VOC-Sensor</b> "0" → Innerhalb des zulässigen Messbereichs "1" → unter dem minimalen VOC-Index "2" → über dem maximalen VOC-Index	Uint8
41	<b>Roh-Messwert VOC-Sensor</b> Gibt den Roh-Wert vom VOC-Sensor zurück	int16
42	<b>Prozentualer Messwert VOC-Sensor</b> Gibt den Prozentualen Messwert von der eingestellten Range für den VOC-Sensor zurück. Ausgabe beinhaltet 2 Nachkommastellen z.B.: 4573 = 45,73%	int16
43 -> 44	<b>Realer Messwert VOC-Sensor</b> Gibt den Messwert in der eingestellten Einheit zurück ohne Nachkommastellen	Int32
45 -> 49	<b>Einheit VOC-Sensor</b> Liest die Einheit des Sensors im <a href="#">String</a> -Format.	10 x Uint8

Temperatursensor		
Register Adresse	Bedeutung/Inhalt	Typ
50 -> 59	<b>Temperatursensor Typ</b> Gibt den Typ des Temperatursensors im <a href="#">String</a> -Format zurück.	20 x Uint8
60	<b>Status Temperatursensor</b> "0" → Innerhalb des eingestellten Messbereichs "1" → unter der minimalen Temperatur "2" → über der maximalen Temperatur	Uint8
61	<b>Roh-Messwert Temperatursensor</b> Gibt den Roh-Wert vom Temperatursensor zurück	int16
62	<b>Prozentualer Messwert Temperatursensor</b> Gibt den Prozentualen Messwert von der eingestellten Range für den Temperatursensor zurück. Ausgabe beinhaltet 2 Nachkommastellen z.B.: 8523 = 85,23%	int16

63 -> 64	<b>Realer Messwert Temperatursensor</b> Gibt den Messwert in der eingestellten Einheit zurück. Ausgabe beinhaltet dabei 2 Nachkommastellen z.B. 2568 = 25,68°C	Int32
65 -> 69	<b>Einheit Temperatursensor</b> Liest die Einheit des Sensors im <a href="#">String</a> -Format.	10 x Uint8

Feuchtesensor		
Register Adresse	Bedeutung/Inhalt	Typ
70 -> 79	<b>Feuchtesensor Typ</b> Gibt den Typ des Feuchtesensors im <a href="#">String</a> -Format zurück.	20 x Uint8
80	<b>Status Feuchtesensor</b> "0" → Innerhalb des eingestellten Messbereichs "1" → unter der minimalen Feuchte "2" → über der maximalen Feuchte	Uint8
81	<b>Roh-Messwert Feuchtesensor</b> Gibt den Roh-Wert vom Feuchtesensor zurück	int16
82	<b>Prozentualer-Messwert Feuchtesensor</b> Gibt den Prozentualen Messwert von der eingestellten Range für den Feuchtesensor zurück. Ausgabe beinhaltet 2 Nachkommastellen z.B.: 8523 = 85,23%	int16
83 -> 84	<b>Realer-Messwert Feuchtesensor</b> Gibt den Messwert in der eingestellten Einheit zurück. Ausgabe beinhaltet dabei 2 Nachkommastellen z.B. 5127 = 51,27%	Int32
85 -> 89	<b>Einheit Feuchtesensor</b> Liest die Einheit des Sensors im <a href="#">String</a> -Format.	10 x Uint8

Drucksensor		
Register Adresse	Bedeutung/Inhalt	Typ
90 -> 99	<b>Drucksensor Typ</b> Gibt den Typ des Drucksensors im <a href="#">String</a> -Format zurück.	20 x Uint8
100	<b>Status Drucksensor</b> "0" → Innerhalb des eingestellten Messbereichs "1" → unter dem minimalen Druck "2" → über dem maximalen Druck	Uint8
101	<b>Roh-Messwert Drucksensor</b> Gibt den Roh-Wert vom Drucksensor zurück	int16
102	<b>Prozentualer-Messwert Drucksensor</b> Gibt den Prozentualen Messwert von der eingestellten Range für den Drucksensor zurück. Ausgabe beinhaltet 2 Nachkommastellen z.B.: 8523 = 85,23%	int16
103 -> 104	<b>Realer-Messwert Drucksensor</b> Gibt den Messwert in der eingestellten Einheit zurück. Ausgabe beinhaltet dabei 4 Nachkommastellen z.B. 25445 = 2,5445Pa	Int32
105 -> 109	<b>Einheit Drucksensor</b> Liest die Einheit des Sensors im <a href="#">String</a> -Format.	10 x Uint8

CO2-Sensor		
Register Adresse	Bedeutung/Inhalt	Typ
110 -> 119	<b>CO2-Sensor Typ</b> Gibt den Typ des CO2-Sensors im <a href="#">String</a> -Format zurück.	20 x Uint8
120	<b>Status CO2-Sensor</b> "0" → Innerhalb des eingestellten Messbereichs "1" → unter dem minimalen Druck "2" → über dem maximalen Druck	Uint8
121	<b>Roh-Messwert CO2-Sensor</b> Gibt den Roh-Wert vom CO2-Sensor zurück	int16
122	<b>Prozentualer-Messwert CO2-Sensor</b> Gibt den Prozentualen Messwert von der eingestellten Range für den CO2-Sensor zurück. Ausgabe beinhaltet 2 Nachkommastellen z.B.: 8523 = 85,23%	int16
123 -> 124	<b>Realer-Messwert CO2-Sensor</b> Gibt den Messwert in der eingestellten Einheit zurück ohne Nachkommastellen z.B. 951 = 951 ppm	Int32
125 -> 129	<b>Einheit CO2-Sensor</b> Liest die Einheit des Sensors im <a href="#">String</a> -Format.	10 x Uint8

Flow-Sensor		
Register Adresse	Bedeutung/Inhalt	Typ
130 -> 139	<b>Flow-Sensor Typ</b> Gibt den Typ des Flow-Sensors im <a href="#">String</a> -Format zurück.	20 x Uint8
140	<b>Status Flow-Sensor</b> "0" → Innerhalb des eingestellten Messbereichs "1" → unter dem minimalen Messbereichs "2" → über dem maximalen Messbereichs	Uint8
141	<b>Roh-Messwert Flow-Sensor</b> Gibt den Roh-Wert vom Flow-Sensor zurück	int16
142	<b>Prozentualer-Messwert Flow-Sensor</b> Gibt den Prozentualen Messwert von der eingestellten Range für den Flow-Sensor zurück. Ausgabe beinhaltet 2 Nachkommastellen z.B.: 8523 = 85,23%	int16
143 -> 144	<b>Realer-Messwert Flow-Sensor</b> Gibt den Messwert in der eingestellten Einheit zurück. Ausgabe beinhaltet dabei 4 Nachkommastellen z.B. 563471 = 56,3471m <sup>3</sup> /h	Int32
145 -> 149	<b>Einheit Flow-Sensor</b> Liest die Einheit des Sensors im <a href="#">String</a> -Format.	10 x Uint8

### 3.3 Holding Registers:

Lesen mit [FC3](#), schreiben mit [FC6](#) (eines Registers) oder mit [FC16](#) (mehrerer Register).

Allgemeine		
Register Adresse	Bedeutung/Inhalt	Typ
0 -> 9	<b>Seriennummer Kunde</b> Speicher für die Seriennummer des Kunden im <a href="#">String</a> -Format.	20 x Uint8
10 -> 11	<b>Baudrate</b> Stellt die Baudrate der RS485 Schnittstelle um. Werte von 3000...200.000 Baud sind möglich	Uint32
12	<b>ModBus Adresse</b> Stellt die ModBus Adresse um. Adresse 0x00 gilt als Broadcastadresse und kann nicht verwendet werden Standardadresse ist 0x01 Adressen von 0x01 bis 0xF7 sind möglich	Uint8

VOC-Sensor		
Register Adresse	Bedeutung/Inhalt	Typ
20	<b>Einheit VOC-Sensor</b> Einstellen der Einheit des Sensors. „0“ → „VOC“	Uint8
21 -> 22	<b>Unterer Messbereich VOC-Sensor</b> Lesen bzw. Schreiben des unteren Messbereichs in der eingestellten Einheit des VOC-Sensors ohne Nachkommastellen. z.B. 35 = 35	Int32
23 -> 24	<b>Oberer Messbereich VOC-Sensor</b> Lesen bzw. Schreiben des oberen Messbereichs in der eingestellten Einheit des VOC-Sensors ohne Nachkommastellen. z.B. 85 = 85	Int32
25	<b>Schaltmodus Grenzkontakt 1 des VOC-Sensors</b> “0“ → Sensor hat keine Auswirkung auf Grenzkontakt “1“ → eine Schaltschwelle: Obere Schaltschwelle > Messwert = Grenzkontakt aus “2“ → eine Schaltschwelle: Obere Schaltschwelle > Messwert = Grenzkontakt ein “3“ → zwei Schaltschwellen: Untere Schaltschwelle < Messwert > obere Schaltschwelle = Grenzkontakt ein “4“ → zwei Schwellen: Untere Schaltschwelle < Messwert > obere Schaltschwelle = Grenzkontakt ein	Uint8
26	<b>Untere Schaltschwelle Grenzkontakt 1 des VOC-Sensors</b> Stellt die untere Schaltschwelle des Grenzkontakt 1 für den Sensor ein. Der Wert wird prozentual mit 2 Nachkommastellen abgegeben. z.B.: 1000 = 10,00%	Int16

27	<b>Obere Schaltschwelle Grenzkontakt 1 des VOC-Sensors</b> Stellt die untere Schaltschwelle des Grenzkontakt 1 für den Sensor ein. Der Wert wird prozentual mit 2 Nachkommastellen abgegeben. z.B.: 9000 = 90,00%	Int16
28	<b>Schaltmodus Grenzkontakt 2 des VOC-Sensors</b> "0" → Sensor hat keine Auswirkung auf Grenzkontakt "1" → eine Schaltschwelle: Obere Schaltschwelle > Messwert = Grenzkontakt aus "2" → eine Schaltschwelle: Obere Schaltschwelle > Messwert = Grenzkontakt ein "3" → zwei Schaltschwellen: Untere Schaltschwelle < Messwert > obere Schaltschwelle = Grenzkontakt ein "4" → zwei Schwellen: Untere Schaltschwelle < Messwert > obere Schaltschwelle = Grenzkontakt ein	Uint8
29	<b>Untere Schaltschwelle Grenzkontakt 2 des VOC-Sensors</b> Stellt die untere Schaltschwelle des Grenzkontakt 2 für den Sensor ein. Der Wert wird prozentual mit 2 Nachkommastellen abgegeben. z.B.: 1000 = 10,00%	Int16
30	<b>Obere Schaltschwelle Grenzkontakt 2 des VOC-Sensors</b> Stellt die untere Schaltschwelle des Grenzkontakt 2 für den Sensor ein. Der Wert wird prozentual mit 2 Nachkommastellen abgegeben. z.B.: 9000 = 90,00%	Int16

Temperatursensor		
Register Adresse	Bedeutung/Inhalt	Typ
40	<b>Einheit Temperatursensor</b> Einstellen der Einheit des Sensors. „0“ → „degK“ „1“ → „degC“ „2“ → „degF“	Uint8
41 -> 42	<b>Unterer Messbereich Temperatursensor</b> Lesen bzw. Schreiben des unteren Messbereichs in der eingestellten Einheit des Temperatursensors mit 2 Nachkommastellen. z.B. -4000 = -40.00 °C	Int32
43 -> 44	<b>Oberer Messbereich Temperatursensor</b> Lesen bzw. Schreiben des oberen Messbereichs in der eingestellten Einheit des Temperatursensors mit 2 Nachkommastellen. z.B. 8000 = 80.00 °C	Int32
45	<b>Schaltmodus Grenzkontakt 1 des Temperatursensors</b> "0" → Sensor hat keine Auswirkung auf Grenzkontakt "1" → eine Schaltschwelle: Obere Schaltschwelle > Messwert = Grenzkontakt aus "2" → eine Schaltschwelle: Obere Schaltschwelle > Messwert = Grenzkontakt ein "3" → zwei Schaltschwellen: Untere Schaltschwelle < Messwert > obere Schaltschwelle = Grenzkontakt ein	Uint8

	<p>“4” → zwei Schwellen: Untere Schaltschwelle &lt; Messwert &gt; obere Schaltschwelle = Grenzkontakt ein</p>	
46	<p><b>Untere Schaltschwelle Grenzkontakt 1 des Temperatursensors</b> Stellt die untere Schaltschwelle des Grenzkontakt 1 für den Sensor ein. Der Wert wird prozentual mit 2 Nachkommastellen abgegeben. z.B.: 1000 = 10,00%</p>	Int16
47	<p><b>Obere Schaltschwelle Grenzkontakt 1 des Temperatursensors</b> Stellt die untere Schaltschwelle des Grenzkontakt 1 für den Sensor ein. Der Wert wird prozentual mit 2 Nachkommastellen abgegeben. z.B.: 9000 = 90,00%</p>	Int16
48	<p><b>Schaltmodus Grenzkontakt 2 des Temperatursensors</b> “0” → Sensor hat keine Auswirkung auf Grenzkontakt “1” → eine Schaltschwelle: Obere Schaltschwelle &gt; Messwert = Grenzkontakt aus “2” → eine Schaltschwelle: Obere Schaltschwelle &gt; Messwert = Grenzkontakt ein “3” → zwei Schaltschwellen: Untere Schaltschwelle &lt; Messwert &gt; obere Schaltschwelle = Grenzkontakt ein “4” → zwei Schwellen: Untere Schaltschwelle &lt; Messwert &gt; obere Schaltschwelle = Grenzkontakt ein</p>	UInt8
49	<p><b>Untere Schaltschwelle Grenzkontakt 2 des Temperatursensors</b> Stellt die untere Schaltschwelle des Grenzkontakt 2 für den Sensor ein. Der Wert wird prozentual mit 2 Nachkommastellen abgegeben. z.B.: 1000 = 10,00%</p>	Int16
50	<p><b>Obere Schaltschwelle Grenzkontakt 2 des Temperatursensors</b> Stellt die untere Schaltschwelle des Grenzkontakt 2 für den Sensor ein. Der Wert wird prozentual mit 2 Nachkommastellen abgegeben. z.B.: 9000 = 90,00%</p>	Int16

Feuchtesensor		
Register Adresse	Bedeutung/Inhalt	Typ
60	<p><b>Einheit Feuchtesensor</b> Einstellen der Einheit des Sensors. „0” → „%rH“ „1” → „g/m<sup>3</sup>“</p>	UInt8
61 -> 62	<p><b>Unterer Messbereich Feuchtesensor</b> Lesen bzw. Schreiben des unteren Messbereichs in der eingestellten Einheit des Feuchtesensors mit 1 Nachkommastellen. z.B. 205 = 20.5 %rH</p>	Int32
63 -> 64	<p><b>Oberer Messbereich Feuchtesensor</b> Lesen bzw. Schreiben des oberen Messbereichs in der eingestellten Einheit des Feuchtesensors mit 1 Nachkommastellen. z.B. 753 = 75.3 %rH</p>	Int32

65	<p><b>Schaltmodus Grenzkontakt 1 des Feuchtesensors</b></p> <p>“0” → Sensor hat keine Auswirkung auf Grenzkontakt</p> <p>“1” → eine Schaltschwelle: Obere Schaltschwelle &gt; Messwert = Grenzkontakt aus</p> <p>“2” → eine Schaltschwelle: Obere Schaltschwelle &gt; Messwert = Grenzkontakt ein</p> <p>“3” → zwei Schaltschwellen: Untere Schaltschwelle &lt; Messwert &gt; obere Schaltschwelle = Grenzkontakt ein</p> <p>“4” → zwei Schwellen: Untere Schaltschwelle &lt; Messwert &gt; obere Schaltschwelle = Grenzkontakt ein</p>	Uint8
66	<p><b>Untere Schaltschwelle Grenzkontakt 1 des Feuchtesensors</b></p> <p>Stellt die untere Schaltschwelle des Grenzkontakt 1 für den Sensor ein. Der Wert wird prozentual mit 2 Nachkommastellen abgegeben. z.B.: 1000 = 10,00%</p>	Int16
67	<p><b>Obere Schaltschwelle Grenzkontakt 1 des Feuchtesensors</b></p> <p>Stellt die untere Schaltschwelle des Grenzkontakt 1 für den Sensor ein. Der Wert wird prozentual mit 2 Nachkommastellen abgegeben. z.B.: 9000 = 90,00%</p>	Int16
68	<p><b>Schaltmodus Grenzkontakt 2 des Feuchtesensors</b></p> <p>“0” → Sensor hat keine Auswirkung auf Grenzkontakt</p> <p>“1” → eine Schaltschwelle: Obere Schaltschwelle &gt; Messwert = Grenzkontakt aus</p> <p>“2” → eine Schaltschwelle: Obere Schaltschwelle &gt; Messwert = Grenzkontakt ein</p> <p>“3” → zwei Schaltschwellen: Untere Schaltschwelle &lt; Messwert &gt; obere Schaltschwelle = Grenzkontakt ein</p> <p>“4” → zwei Schwellen: Untere Schaltschwelle &lt; Messwert &gt; obere Schaltschwelle = Grenzkontakt ein</p>	Uint8
69	<p><b>Untere Schaltschwelle Grenzkontakt 2 des Feuchtesensors</b></p> <p>Stellt die untere Schaltschwelle des Grenzkontakt 2 für den Sensor ein. Der Wert wird prozentual mit 2 Nachkommastellen abgegeben. z.B.: 1000 = 10,00%</p>	Int16
70	<p><b>Obere Schaltschwelle Grenzkontakt 2 des Feuchtesensors</b></p> <p>Stellt die untere Schaltschwelle des Grenzkontakt 2 für den Sensor ein. Der Wert wird prozentual mit 2 Nachkommastellen abgegeben. z.B.: 9000 = 90,00%</p>	Int16

Drucksensor		
Register Adresse	Bedeutung/Inhalt	Typ
80	<b>Einheit Drucksensor</b> Einstellen der Einheit des Sensors. „0“ → „Pa“ „1“ → „hPa“ „2“ → „kPa“ „3“ → „mbar“ „4“ → „bar“ „5“ → „psi“ „6“ → „mmH2O“	Uint8
81 -> 82	<b>Unterer Messbereich Drucksensor</b> Lesen bzw. Schreiben des unteren Messbereichs in der eingestellten Einheit des Drucksensors mit 4 Nachkommastellen. z.B. -1000000 = -100.0000 Pa	Int32
83 -> 84	<b>Oberer Messbereich Drucksensor</b> Lesen bzw. Schreiben des oberen Messbereichs in der eingestellten Einheit des Drucksensors mit 4 Nachkommastellen. z.B. 1000000 = 100.0000 Pa	Int32
85	<b>Schaltmodus Grenzkontakt 1 des Drucksensors</b> „0“ → Sensor hat keine Auswirkung auf Grenzkontakt „1“ → eine Schaltschwelle: Obere Schaltschwelle > Messwert = Grenzkontakt aus „2“ → eine Schaltschwelle: Obere Schaltschwelle > Messwert = Grenzkontakt ein „3“ → zwei Schaltschwellen: Untere Schaltschwelle < Messwert > obere Schaltschwelle = Grenzkontakt ein „4“ → zwei Schwellen: Untere Schaltschwelle < Messwert > obere Schaltschwelle = Grenzkontakt ein	Uint8
86	<b>Untere Schaltschwelle Grenzkontakt 1 des Drucksensors</b> Stellt die untere Schaltschwelle des Grenzkontakt 1 für den Sensor ein. Der Wert wird prozentual mit 2 Nachkommastellen abgegeben. z.B.: 1000 = 10,00%	Int16
87	<b>Obere Schaltschwelle Grenzkontakt 1 des Drucksensors</b> Stellt die untere Schaltschwelle des Grenzkontakt 1 für den Sensor ein. Der Wert wird prozentual mit 2 Nachkommastellen abgegeben. z.B.: 9000 = 90,00%	Int16
88	<b>Schaltmodus Grenzkontakt 2 des Drucksensors</b> „0“ → Sensor hat keine Auswirkung auf Grenzkontakt „1“ → eine Schaltschwelle: Obere Schaltschwelle > Messwert = Grenzkontakt aus „2“ → eine Schaltschwelle: Obere Schaltschwelle > Messwert = Grenzkontakt ein „3“ → zwei Schaltschwellen: Untere Schaltschwelle < Messwert > obere Schaltschwelle = Grenzkontakt ein	Uint8

	<p>“4” → zwei Schwellen:  Untere Schaltschwelle &lt; Messwert &gt; obere Schaltschwelle =  Grenzkontakt ein</p>	
89	<p><b>Untere Schaltschwelle Grenzkontakt 2 des Drucksensors</b>  Stellt die untere Schaltschwelle des Grenzkontakt 2 für den Sensor ein.  Der Wert wird prozentual mit 2 Nachkommastellen abgegeben.  z.B.: 1000 = 10,00%</p>	Int16
90	<p><b>Obere Schaltschwelle Grenzkontakt 2 des Drucksensors</b>  Stellt die untere Schaltschwelle des Grenzkontakt 2 für den Sensor ein.  Der Wert wird prozentual mit 2 Nachkommastellen abgegeben.  z.B.: 9000 = 90,00%</p>	Int16

CO2-Sensor		
Register Adresse	Bedeutung/Inhalt	Typ
100	<p><b>Einheit CO2-Sensor</b>  Einstellen der Einheit des Sensors.  „0“ → „ppm“</p>	Uint8
101 -> 102	<p><b>Unterer Messbereich CO2-Sensor</b>  Lesen bzw. Schreiben des unteren Messbereichs in der eingestellten Einheit des CO2-Sensors mit ohne Nachkommastellen.  z.B. 500 = 500 ppm</p>	Int32
103 -> 104	<p><b>Oberer Messbereich CO2-Sensor</b>  Lesen bzw. Schreiben des oberen Messbereichs in der eingestellten Einheit des CO2-Sensors mit ohne Nachkommastellen.  z.B. 2000 = 2000 ppm</p>	Int32
105	<p><b>Schaltmodus Grenzkontakt 1 des CO2-Sensors</b>  “0” → Sensor hat keine Auswirkung auf Grenzkontakt  “1” → eine Schaltschwelle:  Obere Schaltschwelle &gt; Messwert = Grenzkontakt aus  “2” → eine Schaltschwelle:  Obere Schaltschwelle &gt; Messwert = Grenzkontakt ein  “3” → zwei Schaltschwellen:  Untere Schaltschwelle &lt; Messwert &gt; obere Schaltschwelle =  Grenzkontakt ein  “4” → zwei Schwellen:  Untere Schaltschwelle &lt; Messwert &gt; obere Schaltschwelle =  Grenzkontakt ein</p>	Uint8
106	<p><b>Untere Schaltschwelle Grenzkontakt 1 des CO2-Sensors</b>  Stellt die untere Schaltschwelle des Grenzkontakt 1 für den Sensor ein</p>	Int16
107	<p><b>Obere Schaltschwelle Grenzkontakt 1 des CO2-Sensors</b>  Stellt die untere Schaltschwelle des Grenzkontakt 1 für den Sensor ein</p>	Int16
108	<p><b>Schaltmodus Grenzkontakt 2 des CO2-Sensor</b>  “0” → Sensor hat keine Auswirkung auf Grenzkontakt  “1” → eine Schaltschwelle:  Obere Schaltschwelle &gt; Messwert = Grenzkontakt aus  “2” → eine Schaltschwelle:  Obere Schaltschwelle &gt; Messwert = Grenzkontakt ein  “3” → zwei Schaltschwellen:</p>	Uint8

	<p>Untere Schaltschwelle &lt; Messwert &gt; obere Schaltschwelle = Grenzkontakt ein "4" → zwei Schwellen: Untere Schaltschwelle &lt; Messwert &gt; obere Schaltschwelle = Grenzkontakt ein</p>	
109	<p><b>Untere Schaltschwelle Grenzkontakt 2 des CO2-Sensors</b> Stellt die untere Schaltschwelle des Grenzkontakt 2 für den Sensor ein. Der Wert wird prozentual mit 2 Nachkommastellen abgegeben. z.B.: 1000 = 10,00%</p>	Int16
110	<p><b>Obere Schaltschwelle Grenzkontakt 2 des CO2-Sensors</b> Stellt die obere Schaltschwelle des Grenzkontakt 2 für den Sensor ein. Der Wert wird prozentual mit 2 Nachkommastellen abgegeben. z.B.: 9000 = 90,00%</p>	Int16

Flow-Sensor		
Register Adresse	Bedeutung/Inhalt	Typ
120	<p><b>Einheit Flow-Sensor</b> Einstellen der Einheit des Sensors. „0“ → „m<sup>3</sup>/h“ „1“ → „m<sup>3</sup>/min“ „2“ → „m<sup>3</sup>/s“ „3“ → „l/h“ „4“ → „l/min“ „5“ → „l/s“</p>	UInt8
121 -> 122	<p><b>Unterer Messbereich Flow-Sensor</b> Lesen bzw. Schreiben des unteren Messbereichs in der eingestellten Einheit des Drucksensors mit 4 Nachkommastellen. z.B. -1000000 = -100.0000 m<sup>3</sup>/h</p>	Int32
123 -> 124	<p><b>Oberer Messbereich Flow-Sensor</b> Lesen bzw. Schreiben des oberen Messbereichs in der eingestellten Einheit des Drucksensors mit 4 Nachkommastellen. z.B. 1000000 = 100.0000 m<sup>3</sup>/h</p>	Int32
125	<p><b>Schaltmodus Grenzkontakt 1 des Flow-Sensor</b> "0" → Sensor hat keine Auswirkung auf Grenzkontakt "1" → eine Schaltschwelle: Obere Schaltschwelle &gt; Messwert = Grenzkontakt aus "2" → eine Schaltschwelle: Obere Schaltschwelle &gt; Messwert = Grenzkontakt ein "3" → zwei Schaltschwellen: Untere Schaltschwelle &lt; Messwert &gt; obere Schaltschwelle = Grenzkontakt ein "4" → zwei Schwellen: Untere Schaltschwelle &lt; Messwert &gt; obere Schaltschwelle = Grenzkontakt ein</p>	UInt8
126	<p><b>Untere Schaltschwelle Grenzkontakt 1 des Flow-Sensor</b> Stellt die untere Schaltschwelle des Grenzkontakt 1 für den Sensor ein. Der Wert wird prozentual mit 2 Nachkommastellen abgegeben. z.B.: 1000 = 10,00%</p>	Int16

127	<b>Obere Schaltschwelle Grenzkontakt 1 des Flow-Sensor</b> Stellt die untere Schaltschwelle des Grenzkontakt 1 für den Sensor ein. Der Wert wird prozentual mit 2 Nachkommastellen abgegeben. z.B.: 9000 = 90,00%	Int16
128	<b>Schaltmodus Grenzkontakt 2 des Flow-Sensor</b> "0" → Sensor hat keine Auswirkung auf Grenzkontakt "1" → eine Schaltschwelle: Obere Schaltschwelle > Messwert = Grenzkontakt aus "2" → eine Schaltschwelle: Obere Schaltschwelle > Messwert = Grenzkontakt ein "3" → zwei Schaltschwellen: Untere Schaltschwelle < Messwert > obere Schaltschwelle = Grenzkontakt ein "4" → zwei Schwellen: Untere Schaltschwelle < Messwert > obere Schaltschwelle = Grenzkontakt ein	Uint8
129	<b>Untere Schaltschwelle Grenzkontakt 2 des Flow-Sensor</b> Stellt die untere Schaltschwelle des Grenzkontakt 2 für den Sensor ein. Der Wert wird prozentual mit 2 Nachkommastellen abgegeben. z.B.: 1000 = 10,00%	Int16
130	<b>Obere Schaltschwelle Grenzkontakt 2 des Flow-Sensor</b> Stellt die untere Schaltschwelle des Grenzkontakt 2 für den Sensor ein. Der Wert wird prozentual mit 2 Nachkommastellen abgegeben. z.B.: 9000 = 90,00%	Int16

## 4 CRC berechnen

Zur berechnung der Checksumme kann folgende Funktion genutzt werden:

```
uint16_t CalculateCRC(uint8_t ByteCount, uint8_t *CRCDData)
{
    unsigned char CRCHi = 0xFF;
    unsigned char CRCLo = 0xFF;
    unsigned Index;
    unsigned short Bytecount = ByteCount - 2;

    while (Bytecount--)
    {
        Index = CRCLo ^ *CRCDData++;
        CRCLo = CRCHi ^ CRCHighByte[Index] ;
        CRCHi = CRCLowByte[Index];
    }
    return (CRCHi << 8 | CRCLo);
}
```

## 5 String aufbau

Strings werden im ASCII-Format dargestellt und mit 0x00 Terminiert.

Form in Ansi-C Standard: z.B. "V01.00"

Form in ASCII Standard: "V01.00[NULL NULL]"

Form in Hex : 0x 5630 312E 3030 00

### 5.1 ASCII Table

# ASCII TABLE

Decimal	Hex	Char	Decimal	Hex	Char	Decimal	Hex	Char	Decimal	Hex	Char
0	0	[NULL]	32	20	[SPACE]	64	40	@	96	60	`
1	1	[START OF HEADING]	33	21	!	65	41	A	97	61	a
2	2	[START OF TEXT]	34	22	"	66	42	B	98	62	b
3	3	[END OF TEXT]	35	23	#	67	43	C	99	63	c
4	4	[END OF TRANSMISSION]	36	24	\$	68	44	D	100	64	d
5	5	[ENQUIRY]	37	25	%	69	45	E	101	65	e
6	6	[ACKNOWLEDGE]	38	26	&	70	46	F	102	66	f
7	7	[BELL]	39	27	'	71	47	G	103	67	g
8	8	[BACKSPACE]	40	28	(	72	48	H	104	68	h
9	9	[HORIZONTAL TAB]	41	29	)	73	49	I	105	69	i
10	A	[LINE FEED]	42	2A	*	74	4A	J	106	6A	j
11	B	[VERTICAL TAB]	43	2B	+	75	4B	K	107	6B	k
12	C	[FORM FEED]	44	2C	,	76	4C	L	108	6C	l
13	D	[CARRIAGE RETURN]	45	2D	-	77	4D	M	109	6D	m
14	E	[SHIFT OUT]	46	2E	.	78	4E	N	110	6E	n
15	F	[SHIFT IN]	47	2F	/	79	4F	O	111	6F	o
16	10	[DATA LINK ESCAPE]	48	30	0	80	50	P	112	70	p
17	11	[DEVICE CONTROL 1]	49	31	1	81	51	Q	113	71	q
18	12	[DEVICE CONTROL 2]	50	32	2	82	52	R	114	72	r
19	13	[DEVICE CONTROL 3]	51	33	3	83	53	S	115	73	s
20	14	[DEVICE CONTROL 4]	52	34	4	84	54	T	116	74	t
21	15	[NEGATIVE ACKNOWLEDGE]	53	35	5	85	55	U	117	75	u
22	16	[SYNCHRONOUS IDLE]	54	36	6	86	56	V	118	76	v
23	17	[ENG OF TRANS. BLOCK]	55	37	7	87	57	W	119	77	w
24	18	[CANCEL]	56	38	8	88	58	X	120	78	x
25	19	[END OF MEDIUM]	57	39	9	89	59	Y	121	79	y
26	1A	[SUBSTITUTE]	58	3A	:	90	5A	Z	122	7A	z
27	1B	[ESCAPE]	59	3B	;	91	5B	[	123	7B	{
28	1C	[FILE SEPARATOR]	60	3C	<	92	5C	\	124	7C	
29	1D	[GROUP SEPARATOR]	61	3D	=	93	5D	]	125	7D	}
30	1E	[RECORD SEPARATOR]	62	3E	>	94	5E	^	126	7E	~
31	1F	[UNIT SEPARATOR]	63	3F	?	95	5F	_	127	7F	[DEL]