

## ***ÄNDERUNGEN/SOFTWAREUPDATES***

20.12.2020 Schreib.-/ Lese-Register fehlerhaft; Stefan Müller

## BUS-Protokoll für das RA-GAS-Modbus-System

Verwendete Abkürzungen:

Rreg	(read) Lese-Register (kann nicht beschrieben werden)
RWreg	(read/write) Lese- und Schreibregister (kann auch beschrieben werden, teilweise mit Speicherung)
Fcode	Funktionscode
Adr	Adresse
Reg	Register

### 1 Registerbelegung

**Tabelle der Lese(Read)-Register**

<b>Rreg Nr.</b> (Fcode 0x04)	<b>Wertebereich</b>	<b>Zugeordnete Größe und teilw. Einheit</b>	<b>Messwerteigenschaft</b>
00	0 ... 65535		Geräteerkennung Kunden
01	0 ... 65535		Arbeitsweise (Sensor)
02	0 .. 10000	0 .. 10000 ppm	Gaskonzentration im ppm
03	0 ... 2500	0 .. 25.00 mA	Berechneter Ausgangsstrom in mA (mit zwei Kommastellen)
04	-200 ... 600	-20,0 .. 60,0 °C	Interne Leiterplattentemperatur in °C (mit Kommastelle)
05	0 ... 0xffff		Fehlererkennung (Bit's werden gesetzt)
40	0 ... 16384		AD-Wert der Temperaturmessung
41	0 ... 16384		AD-Wert des Potentiometers
42	0 ... 16384		AD-Wert des Sensors
43	50 ... 200	0,50 .. 2,00	Verstärkungsfaktor durch Poti (100 = 1,00)
44	50 ... 200	0,50 .. 2,00	Verstärkungsfaktor durch Temperaturkennlinie (100 = 1,00)
45	0 ... 16384		Korrigierter AD-Wert des Sensors
46	0 .. 10000	0 .. 10000 ppm	berechnete Gaskonzentration im ppm
49	0 .. 31129		Softwaredatum bis 31.12.2029

**Hinweis:** der angegebene Wertebereich widerspiegelt nicht gleichzeitig den Messbereich. Dieser ist abhängig vom Sensor und der Kalibrierung.

#### **Zusätzliche Erläuterungen zu einigen Rreg-Registern:**

Rreg\_00: hier befindet sich der Kundencode welcher vom Kunden in Register RWreg\_00 geschrieben und gespeichert wurde. (kann z.B. zur Raumnummerierung verwendet werden.)

Rreg\_01: Arbeitscode (Sensor)  
10 = CO-Sensor (1000)  
12 = CO-Sensor (300)  
20 = NO-Sensor (250)  
30 = NO2 (20)  
40 = NH3 (1000)  
42 = NH3 (100)  
50 = CL2 (10)  
60 = H2S (100)

Rreg\_02: Berechneter ppm-Wert aus Rreg\_46 (mit Nullpunktberuhigung)

Rreg\_03: der berechnete Strom für den analogen Stromausgang 4..20mA

Rreg\_05: Fehlerregister:

Bit\_0 = 1: (Sensorspannung[Endwert – Nullwert]) < 2 digit/ppm

Bit\_1 = 1: (Sensorspannung Endwert od. Nullwert) < 50 od. > 16000 digit

Bit\_2 = 1: (ungünstiger Kalibrierwert) Berechnung läuft zu < 50 und > 16000 digit

Bit\_3 = 1: Sensor-AD-Wert < 50 bzw. > 16000 digit

Bit\_4 = 1: Ausgangsstrom < 390 bzw. > 2100

Der Fehlercode wird auch durch die rote LED dargestellt. (Anzahl der Unterbrechungen entspricht dem gesetzten Bit)

Rreg\_45: entspricht  $Rreg\_42 * Rreg\_43 * Rreg\_44$   
Rreg\_46: Berechneter ppm-Wert aus dem linearen Zusammenhang RWreg\_10..\_13

**Tabelle der Lese/Schreib(Read/Write)-Register**

<b>Rwreg Nr.</b> (Fcode: 0x03, 0x06)	<b>Wertebereich</b>	<b>Zugeordnete Größe und Einheit</b>	<b>Messwerteigenschaft</b>
00	0 .. 65535 [0]		Kundencode: zur freien Belegung z.B. Raumcode *
02	0 ... 10000 [11111]	0 ... 10000 ppm	Messwertvorgabe für Testzwecke
03	0 ... 2500 [11111]	0 ... 25,00 mA	Ausgangsstrom vorgeben für Testzwecke
04	-200 ... 600 [11111]	-20,0 ... 60,0 °C	Temperatur vorgeben für Testzwecke
10	0 ... 16383		Sensorspannung im Nullpunkt *
11	0	0	Sensorwert Nullpunkt = 0 *
12	0 ... 16383		Sensorspannung im Kalibrierpunkt ( bei Endwert) *
13	0 ... 10000	0 ... 10000 ppm	Sensorwert im Kalibrierpunkt (bei Endwert) *
15	0 ... 10000 [0]	0 ... 10000 ppm [0 ppm]	Messwert unten für Ausgangsstrom unten *
16	0 ... 2500 [400]	0 ... 25,00 mA [4 mA]	Ausgangsstrom im unteren Punkt *
17	0 ... 10000 [1000]	0 ... 10000 ppm [1000ppm]	Messwert oben für Ausgangsstrom oben *
18	0 ... 2500 [2000]	0 ... 25,00 mA [20 mA]	Ausgangsstrom im oberen Punkt *
20	0 / 1		Status (Auswerte IC) (keine Eingabemöglichkeit)
21	0 / 1		Lock (Auswerte IC) *
22	0 ... 7		TIA_GAIN (Auswerte IC) *
23	0 ... 3		RLOAD (Auswerte IC) *
24	0 / 1		REF_Source (Auswerte IC) *
25	0 ... 3		INT_Z (Auswerte IC) *
26	0 / 1		BIAS_Sign (Auswerte IC) *
27	0 ... 13		BIAS (Auswerte IC) *
28	0 / 1		FET_Short (Auswerte IC) *
29	0 ... 7		OP_Mode (Auswerte IC) *
30	50 ... 200	0,50 ... 2,00	Kennlinie vom Sensorhersteller bei -20°C *
31	50 ... 200	0,50 ... 2,00	Kennlinie vom Sensorhersteller bei 0°C *
32	50 ... 200	0,50 ... 2,00	Kennlinie vom Sensorhersteller bei 10°C *
33	50 ... 200	0,50 ... 2,00	Kennlinie vom Sensorhersteller bei 20°C *
34	50 ... 200	0,50 ... 2,00	Kennlinie vom Sensorhersteller bei 30°C *
35	50 ... 200	0,50 ... 2,00	Kennlinie vom Sensorhersteller bei 40°C *
36	50 ... 200	0,50 ... 2,00	Kennlinie vom Sensorhersteller bei 60°C *
42	0 ... 16000 [11111]		Sensor AD-Wert vorgeben für Testzwecke
49	0 ... 65535		Neustart / Grunddaten / entsichern
50	1 ... 247 [1]		Modbus-Geräteadresse *
51	0 ... 3 [1]		Modbus Baudrate *
52	0 ... 4 [0]		Modbus Mode *
53	10 .. 1000 [180]		Kalibrierwert Ausgangsstrom 4mA *
54	10 ... 1000 [900]		Kalibrierwert Ausgangsstrom 20mA *
95	0, 129 ... 256 [90]		Sensornummer für MCS4000 - Mode
96	0 ... 65535		Einschaltzähler *
97	0 ... 65535		Betriebsstunden *

98	0 ... 65535		Geräteerkennung vom Werk	*
99	0 ... 65535		Arbeitsweise vom Werk	*

*Hinweise:*

- \* (**fett**) eingetragene Werte werden auch gespeichert  
(Achtung: nicht kontinuierlich beschreiben!)
- [x] Wert nach dem Einschalten bzw. bei Voreinstellung (Werkseinstellung)
- Grau hinterlegte Werte nur nach Entsicherung veränderbar

**Zusätzliche Erläuterungen zu einigen RWreg-Registern:**

RWreg\_00: hier kann ein Kundencode für eine Gerätezuordnung eingetragen werden z.B. zur Raumnummerierung

RWreg\_02, 03, 04, 46:

für Testzwecke Vorgabe verschiedener Werte. Mit 11111 ist dies inaktiv.

RWreg\_10, 11, 12, 13:

Kennlinienpaar für die Umwandlung des Sensor AD\_Wertes in die Gaskonzentration (lineare Interpolation mit diesen Werten)

RWreg\_15, 16, 17, 18:

Kennlinienpaar für die Umwandlung der Gaskonzentration in den Ausgangsstrom (lineare Interpolation mit diesen Werten)

RWreg\_20 bis \_29:

Spezifische Einstellungen für den Auswerte IC

RWreg\_30 bis \_36:

Korrekturwerte entsprechend Vorgabe des Sensorherstellers bzgl. der Temperatur zwischen den Temperaturwerten wird linear interpoliert

RWreg\_49: bei 10 = Reset

bei 20 = Systemneueinstellung und Reset

bei 9876 = entsichert

RWreg\_51: bei 0 = 2400 baud

bei 1 = 9600 baud

bei 2 = 19200 baud

bei 3 = 38400 baud

RWreg\_52: bei 0 = 8 Datenbits, 1 Stoppbit; keine parität

bei 1 = 8 Datenbits, 1 Stoppbit; ungerade Parität

bei 2 = 8 Datenbits, 2 Stoppbit; ungerade Parität

bei 3 = 8 Datenbits, 1 Stoppbit; gerade Parität

bei 4 = 8 Datenbits, 2 Stoppbit; gerade Parität

RWreg\_53: zur Kalibrierung des Ausgangsstroms

Beispiel: in RWreg\_3 = 400 eintragen; Veränderung des analogen Ausgangstrom (gemessen mit Multimeter) über Werteänderung von RWreg\_53 bis 4 mA anliegen

RWreg\_54: zur Kalibrierung des Ausgangsstroms

Beispiel: in RWreg\_3 = 2000 eintragen; Veränderung des analogen Ausgangstrom (gemessen mit Multimeter) über Werteänderung von RWreg\_54 bis 20 mA anliegen.

RWreg\_94: Eingabe einer Zahl 129 ... 256 erzeugt ein Umschalten der RS485 Schnittstelle vom MODBUS-Mode in den MCS4000-Mode mit der entsprechenden Sensornummer

Achtung: nur über den Systemstecker ist eine Reaktivierung möglich (automatisch MODBUS auf Systemstecker und Zugriff auf die Register)

## 2 Modbus-Übertragungsaufbau

Start	Slave Adresse	Funktion	Daten	Checksumme	Ende
3.5* Zeichenzeit	8 Bit	8 Bit	N* 8 Bit	16 Bit	3.5* Zeichenzeit

### *Start/Ende:*

Befinden sich auf dem Modbus keine Daten bzw. gibt es eine Datenpause von 3,5 \* der Zeichenzeit, so wird die Datenerfassung zurückgesetzt.

Ein jetzt neues Zeichen auf dem Bus wird damit als erstes Zeichen (Adresse) erkannt und ausgewertet.

*Beispiel:* 9600 baud, keine Parität, ein Stoppbit  
0,93 ms/Zeichen => ca. 3,3 ms für die Starterkennung

### *Slave Adresse (8 Bit = 1 Byte):*

Die Slave-Adresse (spezifische Geräteadresse) ist im RWreg\_50 abgelegt

Diese darf nur einmal im Modbusstrang verwendet werden.

Wird die Slave Adresse '0' gesendet nehmen alle Geräte welche auf 1 bis 247 geschaltet sind den Befehl an (Boardcast; es gibt jedoch keine Rückantwort!)

### *Funktionscode (8 Bit = 1 Byte):*

Folgende Funktionscodes aus dem allgemeinen Modbus-Protokoll sind implementiert.

Code 03: Registerinhalt (16 Bit) lesen (eines Lese- und Schreib-Registers)

Code 04: Registerinhalt (16 Bit) lesen (eines nur Lese-Registers)

Code 06: Register beschreiben (16 Bit) – ein Register

Code 16: Register beschreiben (16 Bit) – mehrere nacheinander folgende Register (max. 10)

### *Register (16 Bit = 2 Byte):*

Beschreibung siehe Kapitel Registeraufbau

### *Registeranzahl (16 Bit = 2 Byte):*

Für eine Begrenzung der Übertragungszeit/Zeichenketten ist die Registeranzahl auf maximal 10 begrenzt [0x0001 bis 0x000a]

### *Checksumme (16 Bit = 2 Byte):*

Die Ermittlung der Checksumme erfolgt nach den Richtlinien eines Modbus-Protokolls.

Dabei entsteht ein 16 Bit Wert, der mit dem LO- und HI-Byte der Zeichenkette angehängen wird.

## 2.1 Funktionscode 03 lesen von Lese/Schreib(Read/Write)-Registern (16 Bit)

Anfrage:

Slave Adresse	0x00 ... 0xff
<b>Funktionscode</b>	<b>0x03</b>
Start Register	Register HI
Start Register	Register LO
Registeranzahl	Registeranzahl HI
Registeranzahl	Registeranzahl LO
Checksumme	Check LO
Checksumme	Check HI

Rückantwort:

Slave Adresse	0x00 ... 0xff
<b>Funktionscode</b>	<b>0x03</b>
Anzahl der Bytes	Anzahl [n] der Registerwerte (Bytes = n * 2)
1. Registerwert	Wert HI
1. Registerwert	Wert LO
n. Registerwert	Wert HI
n. Registerwert	Wert LO
Checksumme	Check LO
Checksumme	Check HI

Bei fehlerhaftem Register (siehe Registerbelegung)

Slave Adresse	0x00 ... 0xff
<b>Funktionscode</b>	<b>0x83</b>
<b>Fehlercode</b>	<b>0x02</b>
Checksumme	Check LO
Checksumme	Check HI

Bei fehlerhafter Registeranzahl ( $\geq 0x000a$ ) [max. 10\*]

Slave Adresse	0x00 ... 0xff
<b>Funktionscode</b>	<b>0x83</b>
<b>Fehlercode</b>	<b>0x03</b>
Checksumme	Check LO
Checksumme	Check HI

## 2.2 Funktionscode 04 lesen von nur Lese(Read)-Registern (16 Bit)

Anfrage:

Slave Adresse	0x00 ... 0xff
<b>Funktionscode</b>	<b>0x04</b>
Start Register	Register HI
Start Register	Register LO
Registeranzahl	Registeranzahl HI
Registeranzahl	Registeranzahl LO
Checksumme	Check LO
Checksumme	Check HI

Rückantwort:

Slave Adresse	0x00 ... 0xff
<b>Funktionscode</b>	<b>0x04</b>
Anzahl der Bytes	Anzahl [n] der Registerwerte (Bytes = n * 2)
1. Registerwert	Wert HI
1. Registerwert	Wert LO
n. Registerwert	Wert HI
n. Registerwert	Wert LO
Checksumme	Check LO
Checksumme	Check HI

Bei fehlerhaftem Register (siehe Registerbelegung)

Slave Adresse	0x00 ... 0xff
<b>Funktionscode</b>	<b>0x84</b>
<b>Fehlercode</b>	<b>0x02</b>
Checksumme	Check LO
Checksumme	Check HI

Bei fehlerhafter Registeranzahl ( $\geq 0x000a$ ) [max. 10\*]

Slave Adresse	0x00 ... 0xff
<b>Funktionscode</b>	<b>0x84</b>
<b>Fehlercode</b>	<b>0x03</b>
Checksumme	Check LO
Checksumme	Check HI



## 2.3 Funktionscode 06 schreiben einfach Register (16 Bit)

Anfrage:

Slave Adresse	0x00 ... 0xff
<b>Funktionscode</b>	<b>0x06</b>
Register	Register HI
Register	Register LO
Register Wert	Wert HI
Register Wert	Wert LO
Checksumme	Check LO
Checksumme	Check HI

Rückantwort:

Slave Adresse	0x00 ... 0xff
<b>Funktionscode</b>	<b>0x06</b>
Register	Register HI
Register	Register LO
Register Wert	Wert HI
Register Wert	Wert LO
Checksumme	Check LO
Checksumme	Check HI

Bei fehlerhaftem Register (siehe Registerbelegung)

Slave Adresse	0x00 ... 0xff
<b>Funktionscode</b>	<b>0x86</b>
<b>Fehlercode</b>	<b>0x02</b>
Checksumme	Check LO
Checksumme	Check HI

Bei fehlerhaftem Wertebereich

Slave Adresse	0x00 ... 0xff
<b>Funktionscode</b>	<b>0x84</b>
<b>Fehlercode</b>	<b>0x03</b>
Checksumme	Check LO
Checksumme	Check HI

Werden Werte übertragen, die außerhalb des Messbereiches liegen, werden diese auf den Messbereich begrenzt und verwendet. Es wird dennoch die Fehlermeldung (Fehlercode 0x03) gesendet.

## 2.4 Funktionscode 16 schreiben mehrfach Register (16 Bit)

Anfrage:

Slave Adresse	0x00 ... 0xff
<b>Funktionscode</b>	<b>0x10</b>
Start Register	Register HI
Start Register	Register LO
Registeranzahl	Registeranzahl HI
Registeranzahl	Registeranzahl LO
Anzahl der Bytes	Anzahl der Register (n) mal 2
1. Registerwert	Wert HI
1. Registerwert	Wert LO
n. Registerwert	Wert HI
n. Registerwert	Wert LO
Checksumme	Check LO
Checksumme	Check HI

Rückantwort:

Slave Adresse	0x00 ... 0xff
<b>Funktionscode</b>	<b>0x10</b>
Start Register	Register HI
Start Register	Register LO
Registeranzahl	Registeranzahl HI
Registeranzahl	Registeranzahl LO
Checksumme	Check LO
Checksumme	Check HI

Bei fehlerhaftem Register (siehe Registerbelegung)

Slave Adresse	0x00 ... 0xff
<b>Funktionscode</b>	<b>0x90</b>
<b>Fehlercode</b>	<b>0x02</b>
Checksumme	Check LO
Checksumme	Check HI

Bei fehlerhafter Registeranzahl ( $\geq 0x000a$ ) [max. 10\*] oder fehlerhaften Wertebereich

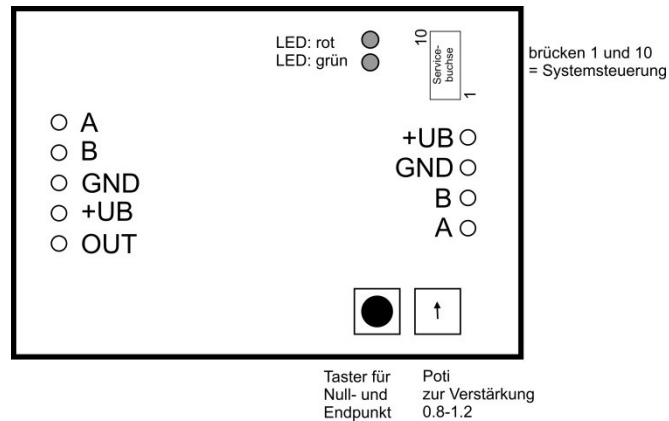
Slave Adresse	0x00 ... 0xff
<b>Funktionscode</b>	<b>0x90</b>
<b>Fehlercode</b>	<b>0x03</b>
Checksumme	Check LO
Checksumme	Check HI

Werden Werte übertragen, die außerhalb des Messbereiches liegen, werden diese auf den Messbereich begrenzt und verwendet. Es wird dennoch die Fehlermeldung (Fehlercode 0x03) gesendet.

*Hinweis:* Die Rückantwortzeit nach erfolgter richtiger Anfrage liegt unter 250 ms (meist kleiner 50 ms).

### 3 Anschlussinformation / Bedienelemente

Das Bild zeigt die wichtigsten Elemente zum Anschluss der Platine.



Versorgungsspannung: +UB = 10 – 36V DC

A & B: sind Leitungen einer RS485

OUT: Stromausgang 4 – 20mA

Achtung: max. Lastwiderstand abhängig von +UB

$$R_{last-max} = (+UB - 10V) / 0,02A$$

Wird beim Systemstecker PIN\_1 (GND) mit Pin\_10 verbunden so schaltet das System in einen vorgegebenen Modus.

- Feste Adresse = 247
- Baudrate = 9600
- MODBUS
- Abschalten des A/B Stranges vom System

Taster:

- zur Nullpunktkalibrierung: drücken => rote LED blinkt, nach ca. 3sec loslassen
- zur Endpunktkalibrierung: drücken => rote LED blinkt, nach ca. 6sec (beim zweiten Dauerlicht) loslassen

LED grün: blinkt (blitzt) wenn Datenaustausch ok ist

LED rot: blitzt bei Fehler im Datenaustausch z.B. Register falsch beschrieben