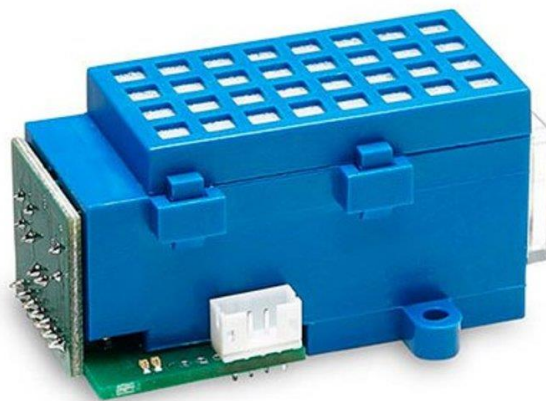


# BASIC<sup>EVO</sup>

Modul- und  
Kommunikationsbeschreibung  
ab Firmware-Version 5.51



## Inhalt

1	Allgemeines .....	4
1.1	Zu Ihrer Sicherheit .....	4
1.2	Bestimmungsgemäße Verwendung .....	4
1.3	Implementierungshinweise .....	5
1.4	Gewährleistungsverlust / Haftung / rechtl. Hinweise .....	5
2	Montage / Einbaulage .....	6
3	Belegung und Eigenschaften der Anschlusspins BASIC <sup>EVO</sup> .....	7
3.1	Stromaufnahme .....	7
4	LED-Statusanzeige .....	8
5	Datenschnittstellen .....	8
5.1	Funktion des Signals COM (Kommunikation) .....	8
5.2	Datenaustausch zwischen Master und BASIC <sup>EVO</sup> (Slave) .....	8
6	Modbus – Kommunikation .....	9
6.1	Automatische Erkennung von Baudrate, Framingformat und Modbus-Dialekt .....	10
6.2	Aufbau von Modbus – Datentelegrammen .....	10
6.3	Modbus Kommunikationsautomat .....	11
6.4	Modbus – Slave-ID .....	11
6.5	Modbus – Steuerbefehle .....	13
6.5.1	Steuerbefehl 0x03 → Read Holding Register .....	13
6.5.2	Steuerbefehl 0x06 → Write Single Register .....	14
6.6	Berechnung der Prüfsumme .....	16
7	Registerübersicht .....	17
7.1	Bedeutung der einzelnen Bits in der Statusbitleiste (SYS_Status): .....	18
7.2	Beschreibung des Einheitenkenners: .....	19
8	Hinweise zur Inbetriebnahme und Bedienung .....	19
8.1	Selbsttest + Warmup .....	19
8.2	Setzen des Nullpunkts .....	20
8.3	Setzen des Endpunktes .....	20
8.4	Berechnung des Korrekturwertes für den Endpunkt .....	20
8.5	Wiederherstellen der Kalibrierparameter auf Werkseinstellung .....	21
9	Anhang .....	22
9.1	Mechanische Abmessungen [mm] .....	22
9.2	Betrieb des BASIC <sup>EVO</sup> an einem Mikrocontroller .....	23

9.3	Betrieb des BASIC <sup>EVO</sup> an einem PC.....	24
9.4	Einbau des BASIC <sup>EVO</sup> in einer Modulbox mit Gasadapter .....	24
10	Impressum.....	25

## 1 Allgemeines

Das BASIC<sup>EVO</sup> ist eine in vielen Punkten verbesserte Weiterentwicklung des bewährten smartMODUL<sup>BASIC</sup>. Es kann durch seine komfortablen Schnittstellen einfach und schnell in bestehende Mess- und Regelsysteme integriert werden.

Die wichtigsten Änderungen des BASIC<sup>EVO</sup> im Vergleich zum smartMODUL<sup>BASIC</sup>:

- erweiterter Betriebsspannungsbereich 3,3 V – 6,0 V DC (+/- 5%)
- Statusanzeige durch 2 LED`s (Rot/Grün)
- erweiterte Software, leistungsfähiger Prozessor
- verbesserter mechanischer Aufbau, verbesserte Optik
- Unterstützung der MODBUS – Protokolle ASCII (Standard und smartGAS-spezifisch) und RTU

Das BASIC<sup>EVO</sup> basiert auf dem physikalischen Messverfahren der Infrarotabsorption und bietet neben der Selektivität die besten Voraussetzungen für zuverlässige und präzise Messungen.

Die kompakte Bauweise und der geringe Wartungsaufwand prädestinieren es für den Einsatz auch unter schwierigen Bedingungen.

### 1.1 Zu Ihrer Sicherheit

#### Bedeutung der Warnzeichen

Die folgenden Warnzeichen werden in diesem Dokument verwendet, um die zugehörigen Warntexte zu kennzeichnen.



#### VORSICHT!

Hinweis auf eine potenzielle Gefahrensituation. Wird diese nicht vermieden, können Verletzungen oder Schädigungen am Produkt oder der Umwelt eintreten.

Ist auch Warnung vor unsachgemäßem Gebrauch.



#### HINWEIS

Information zum Einsatz des Produktes

Vor Anschluss und Gebrauch des BASIC<sup>EVO</sup> ist diese Anleitung vollständig zu lesen und zu verstehen, bei Fragen oder Unklarheiten bitte unseren Service kontaktieren. Wichtige Informationen sind Warnzeichen kennzeichnen.

Diese Anleitung aufbewahren, ggf. dem Betreiber des Gerätes zur Aufbewahrung übergeben; bei Verkauf des Gerätes ist die Anleitung an den Käufer zu übergeben. Bei der Installation und beim Betrieb des Gerätes sind die gesetzlichen Vorschriften und Richtlinien, die dieses Produkt betreffen, unbedingt zu beachten!

### 1.2 Bestimmungsgemäße Verwendung

Das BASIC<sup>EVO</sup> ist eine autark messfähige Gasmesszelle und dient zur Bestimmung von Gaskonzentrationen gemäß seiner Spezifikation. Für andere Mess-, Prüf- und sonstige Zwecke ist es nicht geeignet.



## **VORSICHT!**

Der Sensor darf nicht in explosionsgefährdeter Umgebung sowie unter rauen Umweltbedingungen (z.B. bei hoher, kondensierender Luftfeuchtigkeit, starker Luftströmung, unter aggressiven Atmosphären, im Freien ohne Gehäuse) betrieben werden.

### **1.3 Implementierungshinweise**

- Wir empfehlen den Einbau des BASIC<sup>EVO</sup> in einem geschlossenen Gasraum z.B. die Verwendung mit einer Modulbox (s. Kapitel 9.4).
- Regelmäßige Nullpunktkontrolle und -justage – erfordert entsprechendes Nullgas
- Regelmäßige Endpunktjustage – erfordert entsprechendes Prüfgas
- Lassen Sie den Sensor vor jeder Kalibrierung mindestens 30 Minuten unter stabilen Umgebungsbedingungen in Betrieb laufen
- Datenkommunikation per UART (B3-Serie) oder RS485 (mit Connect-Interface) über Modbus RTU

### **1.4 Gewährleistungsverlust / Haftung / rechtl. Hinweise**



## **VORSICHT!**

Das Öffnen des Sensors sowie Manipulationen oder Beschädigungen am Gerät führen zum Erlöschen der Gewährleistung!

Gewährleistungsverlust droht außerdem bei Verwendung von aggressiven Chemikalien, Verschmutzungen sowie bei ins Gerät eingedrungenen Flüssigkeiten und bei Nichtbeachten der Hinweise in dieser Modul- und Kommunikationsbeschreibung!

Für Folgeschäden und bei Sach- und Personenschäden, die durch Nichtbeachtung der Modul- und Kommunikationsbeschreibung verursacht werden, übernimmt die smartGAS Mikrosensorik GmbH keine Haftung.

## 2 Montage / Einbaulage

Das BASIC<sup>EVO</sup> verfügt über zwei Befestigungslaschen (Bild 1) an den Seiten. Mit diesen Befestigungslaschen kann das Gerät mittels M3 Schrauben montiert werden. Bitte verwenden Sie hierfür nur Schrauben mit ebener Auflagefläche, z.B. Zylinderschrauben DIN 84, Linsenkopfschrauben DIN 7985 o.ä., jedoch keinesfalls Senkkopfschrauben

Die smartGAS Sensoren ermöglichen den Einbau in die Kundengeräte in verschiedenen Lagen. Da die Werkskalibrierung nicht alle Einbausituationen und Umgebungsbedingungen abdecken kann, ist der Null- und Endpunkt nach Einbau zu überprüfen und ggf. erneut zu kalibrieren. In jedem Fall empfehlen wir eine Funktionsprüfung aller Geräte nach finaler Montage in der Kundenapplikation im Rahmen der Inbetriebnahme.

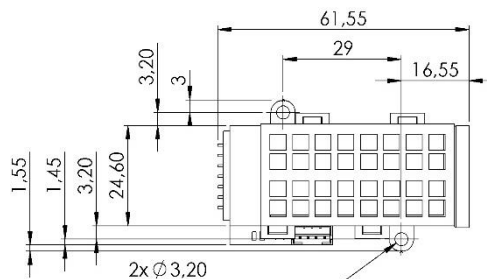


Bild 1: Befestigungslaschen - Ø 3,2mm



### VORSICHT!

*Infrarot-Gassensoren sind optische Messsysteme. Aus diesem Grund sind starke Stöße und Erschütterungen zu vermeiden!*



### HINWEIS

Es ist unbedingt darauf zu achten, dass der Sensor am Montageort nicht folgenden Einflüssen ausgesetzt wird: hohe Luftfeuchtigkeit (kondensierend), extremen Temperaturschwankungen, mechanischen Belastungen (z.B. Vibrationen), Staubeinwirkung, Verschmutzung. Es dürfen keine aggressiven Reinigungsmittel oder Klebstoffe verwendet werden. Modifikationen am BASIC<sup>EVO</sup>, egal welcher Art, sind unbedingt zu vermeiden. Keinesfalls dürfen Flüssigkeiten oder Verschmutzungen ins Innere des Gerätes gelangen!

### 3 Belegung und Eigenschaften der Anschlusspins BASIC<sup>EVO</sup>

Die elektrischen Anschlüsse des BASIC<sup>EVO</sup> erfolgen über eine 4-Pin Steckerleiste mit dem Rastermaß 2mm. Die passende Steckbuchse ist **nicht** im Lieferumfang des BASIC<sup>EVO</sup> enthalten, kann aber gesondert bestellt werden.

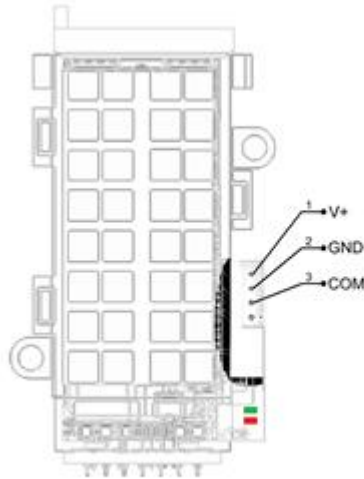


Bild 2: Pinbelegung der Steckerleiste am BASIC<sup>EVO</sup>

#### Stecker ST1

Pin	Belegung
1	+Vcc 3,3 – 6,0 V DC (+/- 5%)
2	GND
3	COM
4	NC (Do not connect)

Tabelle 1: ST1 Anschlussbelegung

#### 3.1 Stromaufnahme

Die folgende Tabelle zeigt eine Übersicht der Strom- bzw. Leistungsaufnahme. Um Störungen durch Spannungseinbrüche zu vermeiden, wird dringend empfohlen, nur ausreichend dimensionierte und spannungsstabilisierte Versorgungsspannungen zu verwenden.

Bei langen Zuleitungen müssen entsprechende Kabelquerschnitte verwendet werden, um übermäßige Spannungsabfälle über die Leitungen zu vermeiden!

Versorgungsspannung	Stromaufnahme
3,3 V	500 mA
6,0 V	240 mA

Tabelle 2: Spannungsabhängige Stromaufnahme



#### HINWEIS

Die Stromaufnahme kann beim Einschalten des BASIC<sup>EVO</sup> kurzzeitig höher sein.

## 4 LED-Statusanzeige

Neben der Steckerleiste befinden sich zwei Leuchtdioden (grün/rot), mit denen der aktuelle Gerätestatus nach Tabelle 3 angezeigt wird:

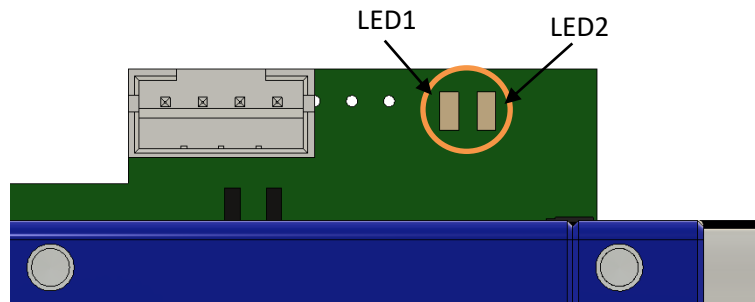


Bild 3: Position der Status LEDs

LD1 (Grün) <span style="color: green;">●</span>	LD2 (Rot) <span style="color: red;">●</span>	Gerätestatus
blinkt	-	Startphase/Selbsttest (ca. 40 Sekunden)
leuchtet	-	Normalbetrieb
-	blinkt	Messbereichsüber- oder Unterschreitung (OUT_OF_RANGE)
-	leuchtet	Gerätefehler/Service verständigen

Tabelle 3: LED-Statusanzeige



### HINWEIS

Messbereichsüber- oder Unterschreitung (OUT\_OF\_RANGE) kann ein und ausgeschaltet werden.

## 5 Datenschnittstellen

### 5.1 Funktion des Signals COM (Kommunikation)

Das BASIC<sup>EVO</sup> verfügt über eine halbduplex UART Datenschnittstelle, die nichtinvertierte UART-Signale liefert bzw. auswertet. Aufgrund des Halbduplex-Betriebs ist auch nur ein Kommunikationssignal (COM) erforderlich. Der Pegel auf der COM-Leitung liegt zwischen 0 V und +3,3 V. Daher kann es notwendig sein, je nach angeschlossenem Kommunikationspartner (Master) eine Pegelanpassung vorzusehen.



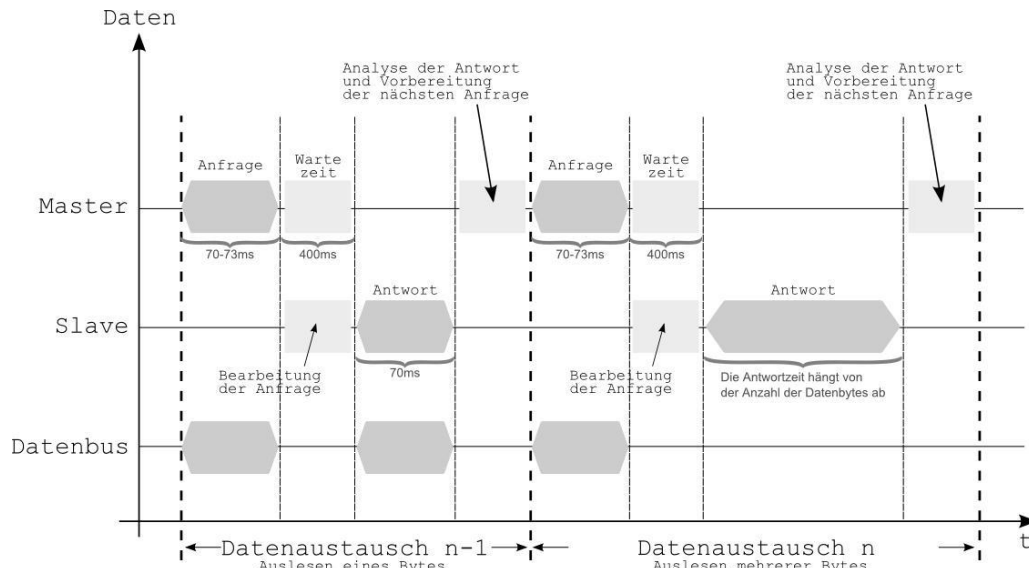
### VORSICHT!

Der Anschluss COM ist als Open-Collector Anschluss mit internem Pullup-Widerstand von 10 kOhm an 3,3 V ausgelegt. Er schaltet gegen GND und darf mit maximal 30 mA belastet werden. Die Spannung darf in keinem Fall 7 VDC überschreiten. Der Einsatz von Schutzwiderständen, EMV-Filtern, galvanischer Trennung und anderen elektrischen oder elektronischen Maßnahmen kann zu Problemen in der Kommunikation führen und muss daher von fachkundigem Personal abgewogen werden.

### 5.2 Datenaustausch zwischen Master und BASIC<sup>EVO</sup> (Slave)

In Bild 4 wird ein mögliches Szenario zwischen Master und BASIC<sup>EVO</sup> (= Slave) dargestellt. Die folgenden Zeiten beziehen sich auf MODBUS-ASCII und einer Baudrate von 2400Bd.





**Bild 4: Zeitdiagramm – Datenaustausch zwischen Master und BASIC<sup>EVO</sup> (Slave)**

Die Dauer eines Anfragestrings beträgt 70 – 73 ms. Danach ist es möglich, dass eine kurze Pause folgt (max. 400 ms). Anschließend folgt die Modulantwort. Diese ist abhängig davon wie viele Bytes ausgelesen werden. Wird nur ein Byte ausgelesen beträgt die Modulantwort ca. 70 ms. Beim Auslesen mehrerer Bytes verlängert sich die Antwortphase entsprechend.

Grundsätzlich kann gesagt werden, dass der BASIC<sup>EVO</sup>- Sensor auf eine Anfrage innerhalb von 400 ms reagiert. Anschließend wird die Zeichenkette direkt ohne Antwortpause gesendet.



### VORSICHT!

Bei höheren Baudraten (> 2400 Bd) kann mit wesentlich schnelleren Antwortzeiten gerechnet werden.



### HINWEIS

Das verwendete Kabel für die Verdrahtung sollte geschirmt und verdreht sein (Twisted-Pair-Kabel).

## 6 Modbus – Kommunikation

Das BASIC<sup>EVO</sup> unterstützt durch seine serielle halbduplexe Schnittstelle das MODBUS-Protokoll im ASCII sowie im RTU-Modus. Im ASCII-Modus existiert neben der Standard-Variante ein smartGAS-spezifisches Derivat, das sich vom Standard in der Berechnung der Prüfsumme unterscheidet.

Grundsätzlich funktioniert die Modbus-Kommunikation auf Basis eines Anfrage-Antwort-Mechanismus. Der Master richtet die Anfrage an einen von möglicherweise mehreren Slaves (Teilnehmern). Jeder angeschlossene Teilnehmer bekommt deshalb eine im Verbund eindeutige Teilnehmeradresse. Nur derjenige Teilnehmer, der seine Adresse in der Anfrage des Masters gefunden hat, wird antworten.

Die Art der Anfrage wird über einen Steuerbefehl (Function Code) bestimmt. So kann es zum Beispiel um ein Schreiben von Daten oder um ein Lesen von Daten zum / vom Teilnehmer gehen. Abhängig vom Steuerbefehl gibt es einen Datenteil, sowohl für die Anfrage als auch für die Antwort.

Jede Anfrage und jede Antwort müssen von ihrem Beginn als auch von ihrem Ende klar gekennzeichnet sein. Die Verwendung eines Prüffeldes (=Prüfwort/CRC) ist im Protokoll vorgesehen, damit eventuelle Kommunikationsfehler erkannt werden können. Modbus-Derivate realisieren dies auf unterschiedliche Art.

Detaillierte Informationen über das Modbus-Protokoll kann man beziehen unter [www.modbus.org](http://www.modbus.org)

## 6.1 Automatische Erkennung von Baudrate, Framingformat und Modbus-Dialekt

Die BASIC<sup>EVO</sup> Firmware ist mit einer automatischen Konfigurationserkennung ausgestattet. Das bedeutet, dass das der Sensor nach dem ersten Einschalten und erste Interaktion auf der Busleitung im System automatisch die verwendete Baudrate, das Framing sowie den MODBUS-Dialekt detektiert. Die in Tabelle 4 aufgeführten Framings und MODBUS-Baudraten harmonisieren miteinander und sind untereinander frei kombinierbar.

Framingformate und MODBUS – Baudraten			
Datenbits↓	Parität↓	Stopbit↓	Baudraten ↓
7	E	1	2400Bd
7	E	2	4800Bd
7	O	1	9600Bd
7	O	2	9200Bd
7	N	2	38400Bd
8	E	1	57600Bd
8	N	1	115200Bd
8	N	2	
8	O	1	

Tabelle 4: Frei kombinierbare Framingformate/Baudraten



### HINWEIS

Bei der Kommunikation über MODBUS-RTU ist es zwingend erforderlich ein Framingformat von 8 Datenbits zu verwenden.

## 6.2 Aufbau von Modbus – Datentelegrammen

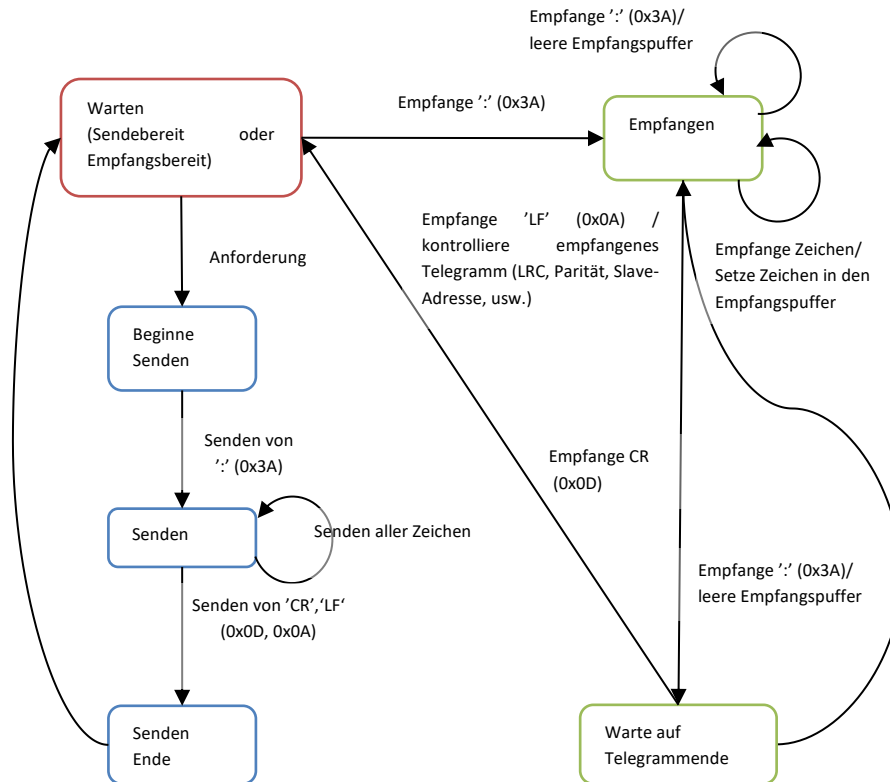
Wie schon an andere Stelle erwähnt empfiehlt die smartGAS Mikrosensorik den Aufbau von eine Datentelegramm mit Modbus-RTU. Es ist möglich den Aufbau des Datentelegrammes auch mit Modbus-ASCII möglich jedoch wird dies hier nicht näher erläutert. Die nachfolgenden Tabellen zeigen den grundsätzlichen Aufbau eines RTU-Datentelegramms:

Dialekt	Start	Slave ID	Function	Data	CRC	Ende
<b>Modbus-RTU</b>		1 Byte z.B.:0xA0	1 Byte z.B.:0x03	0 bis 1x252 Bytes z.B.:0x00,0x05, 0x00,0x02	2 Bytes z.B.:0xA4, 0xD3	
<b>Kommunikation</b>	Pause 3,5 Zeichen	0xA0	0x03	0x00,0x05,0x00,0x02	0xA4,0xD3	Pause 3,5 Zeichen

Im RTU-Modus wird jedes Byte unverändert durchgereicht. Daraus ergibt sich zwangsläufig, dass im RTU-Modus UART-Frames mit 8 Datenbits verwendet werden müssen. Der Vorteil des RTU-Modus liegt in der effektiveren Ausnutzung der Schnittstelle, weil nur etwa die Hälfte der Datenmenge im Vergleich zum ASCII-Modus übertragen werden muss.

## 6.3 Modbus Kommunikationsautomat

In Bild 5 wird, unabhängig ob Master oder Slave, das prinzipielle Zustandsdiagramm des Sende- und Empfangsautomaten dargestellt:



**Bild 5: Zustandsdiagramm eines Modbus-Teilnehmers (Betriebsart ASCII)**

Wird dem BASIC<sup>EVO</sup> eine unvollständige Anfrage gesendet, so wird von diesem keine Antwort zurückgegeben. Genauso verhält sich das Modul, wenn mindestens ein Register im angesprochenen Registerbereich nicht existiert. Fehlerfreie Telegramme werden verarbeitet. Fehlerhafte Telegramme werden nicht beantwortet.

## 6.4 Modbus – Slave-ID

Beim BASIC<sup>EVO</sup>- Sensor entspricht die Geräteadresse (Slave-ID) bei Auslieferung den letzten beiden Ziffern der Seriennummer auf dem Typenschild.

### Im Einzelbetrieb:

Ist nur ein Gerät mit dem Modbus-Master verbunden, so kann das Sensormodul über die globale Slave-ID 248 angesprochen werden.

### Im Mehrfachbetrieb:

Sind mehrere Geräte mit dem Modbus-Master verbunden, so müssen die Sensormodule über ihre Slave-ID angesprochen werden. Die Sensormodule können nicht über die globale Slave-ID 248 angesprochen werden.



**HINWEIS**

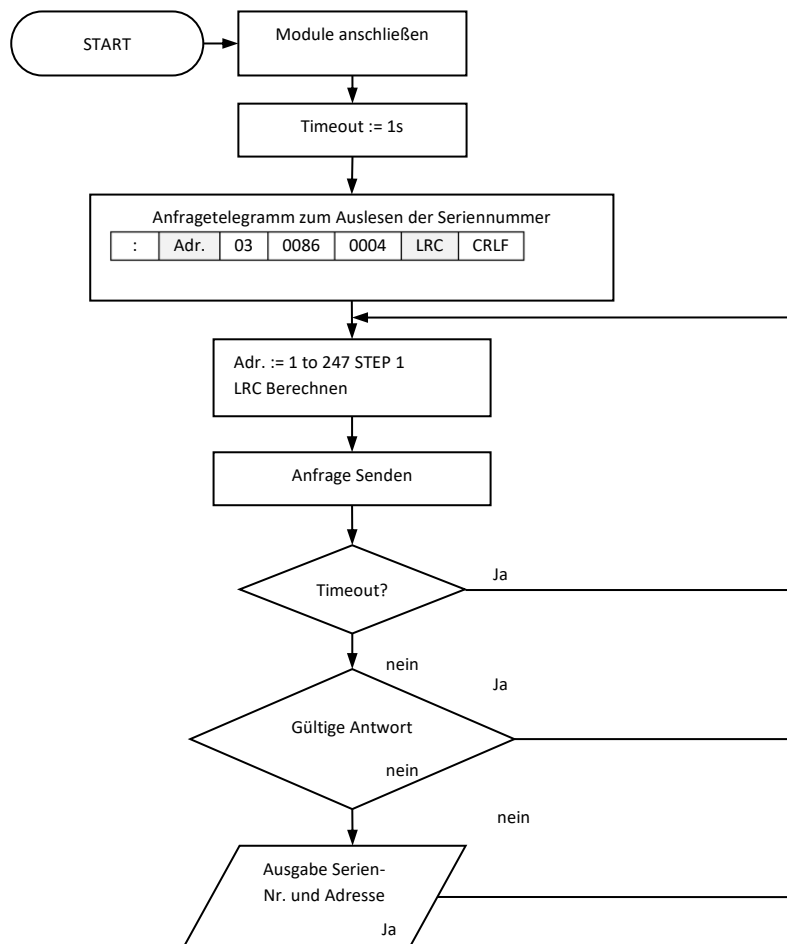
**Beispiel für die Ermittlung der Modbus – Adresse:**

Geräteadresse = #35 Dezimal → 0x23 Hex

Wenn die Seriennummer mit "00" endet, ist die Adresse immer #100 Dezimal = 0x64 Hexadezimal.

Die Adresse „0“ darf niemals verwendet werden!

Bild 6 zeigt in einem Flussdiagramm, wie unbekannte Modbus-Adressen von Modulen ermittelt werden können. Es kann ein beliebiges Register (z.B. Seriennummer) über alle Moduladressen (1-247) mit einem Timeout von einer Sekunde abgefragt werden. Wird ein Modul mit der richtigen Adresse angesprochen, reagiert dies, indem es eine Antwort sendet. In dieser Antwort wird die Moduladresse mitgesendet. Somit kann am Ende des Suchzyklus anhand der Modulantworten analysiert werden, welche Moduladressen gerade am Bussystem angeschlossen sind. Im Falle der Abfrage der Seriennummer, kann man dann rückschließen, welche Adresse welchem Modul zugeordnet ist. Der zulässige Adressbereich für BASIC<sup>EVO</sup> liegt zwischen 1 und 247. Laut der Modbus-Spezifikation sind die Adressen 248-255 reserviert. Die Adresse 0 steht für Broadcast und darf nicht verwendet werden!



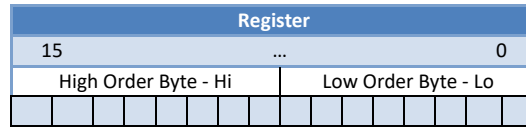
**Bild 6: Flussdiagramm Ermittlung von Moduladressen**

## 6.5 Modbus – Steuerbefehle

Für die Kommunikation mit dem BASIC<sup>EVO</sup>- Sensor unterstützt lediglich zwei Function-Codes:

- **0x03 – Read Holding Registers (multiple)**
- **0x06 – Write Single Register (single)**

Dabei ist ein Register 16 Bit breit, es besteht folglich aus 2 Bytes:



Alle für den Benutzer zugänglichen Daten des BASIC<sup>EVO</sup> sind auf Register mit jeweils 16-Bit Breite abgebildet.

### 6.5.1 Steuerbefehl 0x03 → Read Holding Register

Mit diesem Steuerbefehl ist es möglich, aus dem BASIC<sup>EVO</sup>- Sensor Werte auszulesen. Wesentlich ist, dass nur in dieser Anleitung beschriebene Register gelesen werden können. Daher ist dies besonders bei Abfrage mehrerer Register zu überprüfen.

Anfrage		Antwort		Bedeutung der Daten (nach ASCII – Tabelle)
Feld	(Hex)	Feld	(Hex)	
Modbusadresse	0xXX	Modbusadresse	0xXX	
Function	0x03	Function	0x03	
Start Register Hi	0x00	Byte count	0x08	
Start Register Lo	0x80	Register value Hi (128)	0x53	'S'
Register count Hi	0x00	Register value Lo (128)	0x4D	'M'
Register count Lo	0x04	Register value Hi (129)	0x46	' ' = Leerzeichen
Checksum Lo	0xXX	Register value Lo (129)	0x43	'C'
Checksum Hi	0xXX	Register value Hi (130)	0x4F	'O'
		Register value Lo (130)	0x32	'2'
		Register value Hi (131)	0x20	' ' = Leerzeichen
		Register value Lo (131)	0x20	' ' = Leerzeichen
		Checksum Lo	0xXX	
		Checksum Hi	0xXX	

**Beispiel 1: Auslesen der 4 Register für „Device Type“**

Hier wurden vier Register des BASIC<sup>EVO</sup>- Sensors ab Register-Startadresse 0x0080 (dezimal 128) abgefragt. Die Antwort bestand aus 8 Bytes Nutzdaten, die mit Hilfe der ASCII – Tabelle aufgelöst werden können. Beispiel: Antwort HEX 53 → nach ASCII – Tabelle → Buchstabe S

Die Antwort lautet nun: **"SM CO2"**, es handelt sich also um einen BASIC<sup>EVO</sup>- Sensor (**SM**) für das Messgas Kohlenstoffdioxid (**CO2**).

Anfrage		Antwort		Bedeutung der Daten
Feld	(Hex)	Feld	(Hex)	
Modbusadresse	0xXX	Modbusadresse	0xXX	
Function	0x03	Function	0x03	
Start Register Hi	0x00	Byte count	0x02	
Start Register Lo	0x0A	Register value Hi (14)	0x01	456
Register count Hi	0x00	Register value Lo (14)	0xC8	
Register count Lo	0x01	Checksum Lo	0xXX	
Checksum Lo	0xXX	Checksum Hi	0xXX	
Checksum Hi	0xXX			

**Beispiel 2: Auslesen des Registers „Konz“ (zur Anzeige der Gaskonzentration)**

Hier wurde ein Register ab Register-Startadresse 0x0A (dezimal 10) abgefragt. Die zwei Datenbytes wurden zusammengefasst als Hexadezimalwert übertragen. Wird dieser Wert (01C8) in eine Dezimalzahl umgewandelt, ergibt sich ein Konzentrationswert von 456.

Anfrage		Antwort		Bedeutung der Daten
Feld	(Hex)	Feld	(Hex)	
Modbusadresse	0xXX	Modbusadresse	0xXX	
Function	0x03	Function	0x03	
Start Register Hi	0x00	Byte count	0x02	
Start Register Lo	0x4F	Register value Hi (14)	0x00	3, bedeutet ppm x 1
Register count Hi	0x00	Register value Lo (14)	0x03	
Register count Lo	0x01	Checksum Lo	0xXX	
Checksum Lo	0xXX	Checksum Hi	0xXX	
Checksum Hi	0xXX			

**Beispiel 3: Auslesen des Registers „Einheit“**

Hier wurde ein Register ab Register-Startadresse 0x004F (dezimal 79) abgefragt. Die zwei Datenbytes wurden zusammengefasst als Hexadezimalwert übertragen. Wird dieser Wert (0x0003) in eine Dezimalzahl umgewandelt, ergibt sich „3“. Dies steht für die Einheit ppm mit der Skalierung x 1. Zusammengefasst mit den Daten aus den Beispielen 1 und 2 hat der ausgelesene BASIC<sup>EVO</sup>- Sensor also eine Gaskonzentration von 456 ppm CO<sub>2</sub> gemessen.

## 6.5.2 Steuerbefehl 0x06 → Write Single Register

Dieser Befehl ermöglicht es, einen neuen Wert gezielt in ein adressiertes Register zu schreiben. Dabei können jedoch nur diejenigen Register beschrieben werden, die dafür vorgesehen sind.

Anfrage		Antwort		Bedeutung der Daten
Feld	(Hex)	Feld	(Hex)	
Modbusadresse	0xXX	Modbusadresse	0xXX	
Function	0x06	Function	0x06	
Start Register Hi	0x00	Start Register Hi	0x00	
Start Register Lo	0xC0	Start Register Lo	0xC0	
Register count Hi	0x00	Register count Hi	0x00	Die neue Adresse des Moduls (160)
Register count Lo	0xA0	Register count Lo	0xA0	
Checksum Lo	0xXX	Checksum Lo	0xXX	
Checksum Hi	0xXX	Checksum Hi	0xXX	

**Beispiel 4: Schreiben in das Register „Modbus\_address“**

In diesem Beispiel wurde dem BASIC<sup>EVO</sup>- Sensor eine neue Modbus-Adresse vergeben 0xA0 (Hex) = 160 Dez. Nach Ablauf der Kommunikationssequenz ist das Gerät nur noch unter dieser neuen Adresse ansprechbar!



### HINWEIS

Die Adresse 0 sowie Adressen >247 dürfen nicht vergeben werden!

Anfrage		Antwort		Bedeutung der Daten
Feld	(Hex)	Feld	(Hex)	
Modbusadresse	0xXX	Modbusadresse	0xXX	
Function	0x06	Function	0x06	
Start Register Hi	0x00	Start Register Hi	0x00	
Start Register Lo	0x47	Start Register Lo	0x47	
Register count Hi	0x00	Register count Hi	0x00	Der Nullpunkt wurde neu gesetzt
Register count Lo	0x01	Register count Lo	0x01	
Checksum Lo	0xXX	Checksum Lo	0xXX	
Checksum Hi	0xXX	Checksum Hi	0xXX	

**Beispiel 5: Schreiben in das Register IR\_4tagneu (Setzen des Nullpunkts)**

Hier wurde bei dem BASIC<sup>EVO</sup>- Sensor der Nullpunkt neu gesetzt. Dies geschah durch Schreiben des Wertes 1 ins Register 0x0047(dezimal 71). Daraufhin hat das Gerät intern den aktuellen Korrekturwert für den Nullpunkt berechnet und gespeichert. Ein Auslesen des gleichen Registers zeigt dann den Wert der Korrektur.



### HINWEIS

Das Setzen des Nullpunkts darf nur unter Beaufschlagung mit Nullgas und anschließend stabilem Konzentrationswert erfolgen.

Anfrage		Antwort		Bedeutung der Daten
Feld	(Hex)	Feld	(Hex)	
Modbusadresse	0xXX	Modbusadresse	0xXX	
Function	0x06	Function	0x06	
Start Register Hi	0x00	Start Register Hi	0x00	
Start Register Lo	0x54	Start Register Lo	0x54	
Register count Hi	0x27	Register count Hi	0x27	Korrekturwert wurde auf 10000 gesetzt
Register count Lo	0x10	Register count Lo	0x10	
Checksum Lo	0xXX	Checksum Lo	0xXX	
Checksum Hi	0xXX	Checksum Hi	0xXX	

**Beispiel 6: Schreiben in das Register SPAN (Abgleich Endpunktkorrektur)**

Hier wurde dem BASIC<sup>EVO</sup>- Sensor eine neue Endpunktkorrektur gesetzt. Ein Wert von 2710 (Hex) = 10000 (Dezimal). Dies ist auch der Auslieferungszustand. Ein Wert von 11000 würde z.B. bedeuten, dass der Konzentrationswert um 10% höher angezeigt wird als intern gemessen. Mit diesem Register ist es also möglich, Abweichungen des BASIC<sup>EVO</sup>- Sensors bei der Konzentrationsanzeige zu korrigieren.



### HINWEIS

Das Setzen des Endpunkts auf diese Weise darf nur unter Beaufschlagung mit einem adäquaten Prüfgas und anschließend stabilem Konzentrationswert erfolgen!

Vor dem Setzen des Endpunktes ist es erforderlich, dass vorher der Nullpunkt korrekt gesetzt wurde.

## 6.6 Berechnung der Prüfsumme

Anhand eines Beispiels wird nun die Berechnung der Prüfsumme CRC speziell für die Betriebsart RTU erklärt. Wie die Berechnung der Prüfsummen LRC bei ASCII Standard funktioniert, ist in den Dokumenten des Modbus-Standards eingehend beschrieben.

Die Prüfsumme wird über die Slave-ID, die Function und die zugehörigen Daten (Start register und register count) berechnet. Wir erzeugen als Beispiel eine Anfrage zum Auslesen des Registers Konz aus dem BASIC<sup>EVO</sup>- Sensor mit der Adresse 14 (Dezimal) = 0E (Hex.)

Anfrage	
Feld	(Hex)
Modbusadresse	0x0E
Function	0x03
Start Register Hi	0x00
Start Register Lo	0x0A
Register count Hi	0x00
Register count Lo	0x01
Checksum Lo	0xFF
Checksum Hi	0xFF

Die sich ergebende Bytefolge ist in hexadezimaler Darstellung 0x0E, 0x03, 0x00, 0x0A, 0x00, 0x01.. Jetzt erfolgt die Bildung der Prüfsumme hier ein Beispielcode für Berechnung der CRC Prüfsumme:

```
C# example to calculate modbus RTU checksum:
    /// <summary>
    /// Calculates the checksum of an modbus RTU message and adds it to the end
    (last 2 bytes).
    /// </summary>
    /// <param name="Databytes"></param>
    /// <returns></returns>
    private void Calculate_CRC(ref byte[] Databytes)
    {
        UInt16 v_CRC = 0xFFFF;

        for (int x = 0; x < Databytes.Length - 2; x++)
        {
            v_CRC ^= (UInt16)Databytes[x];           // XOR byte into least sig.
            byte of crc

            for (int y = 8; y != 0; y--)
            {
                // Loop over each bit
                if ((v_CRC & 0x0001) != 0)
                {
                    // If the LSB is set
                    v_CRC >>= 1;                       // Shift right and XOR 0xA001
                    v_CRC ^= 0xA001;
                }
                else
                {
                    // Else LSB is not set
                    // Just shift right
                    v_CRC >>= 1;
                }
            }
        }
    }
}
```

**Bild 7: Code-Beispiel für CRC-Prüfsummen bilden**

Nach der berechneten der Prüfsumme und des Endekenners würde dann also folgender Datenstring gesendet werden: **0xF7A4**



Anfrage	
Feld	(Hex)
Modbusadresse	0x0E
Function	0x03
Start Register Hi	0x00
Start Register Lo	0x0A
Register count Hi	0x00
Register count Lo	0x01
Checksum Lo	0xA4
Checksum Hi	0xF7

Die Prüfsumme wird jedes Mal beim Senden von Daten mit übertragen und anschließend vom Empfänger nachgerechnet. Sollte der Datensatz korrupt oder verfälscht sein, dann würde die berechnete Prüfsumme beim Empfänger von der mitgesendeten abweichen. Der Datensatz wäre also nicht verwendbar.

## 7 Registerübersicht

Alle die unten aufgezählten Register sind Holdingregister.

Adresse	Name	R/W	Datentyp	Funktion / Beschreibung
0x0003	T_m	R/---	signed number	Messwert interne Temperatur (x0,1°C)
0x0009	Sys_status	R/---	number	Statusbitleiste, Erläuterung siehe Seite 18
0x000A	Konz	R/---	signed number	Messwert Gaskonzentration in ppm, Vol.-% oder %UEG (Einheitenkenner beachten!)
0x0047	IR_4tagneu	R/W	number	Nullpunkt-Bezugswert.
0x004F	Einheit	R/---	number	Einheitenkenner und Skalierungsfaktor für die Konzentration. Details und Beispielrechnung weiter unten.
0x0051	Konz_fs	R/---	number	Messbereichsendwert (Full Scale) der Konzentration.
0x0054	Span	R/W	number	Endpunkt-Bezugswert. Wert muss zwischen 5000 und 15000 sein, ansonsten wird dieser auf 10000 zurückgesetzt.
0x0059	fab_zero_value	R/---	signed number	Korrekturfaktor für Nullpunktkalibrierung ab Werk
0x005A	fab_span_value	R/---	signed number	Korrekturfaktor für Spankalibrierung ab Werk
0x0080-0x0083	DeviceType	R/---	string	Gibt den Typ des angeschlossenen Gerätes an
0x0084-0x0085	SW-Version	R/---	string	Firmwareversion des angeschlossenen Gerätes
0x0086-0x0089	SerialNr	R/---	string	Seriennummer des angeschlossenen Gerätes
0x00C0	Modbus_address	R/W	number	MODBUS-Adresse des BASIC <sup>EVO</sup> . Nach Änderung dieser Adresse kann mit dem BASIC <sup>EVO</sup> nur noch über die neue Adresse kommuniziert werden.

Tabelle 5: Modbus-Registertabelle

- **R – Read Holding Registers (multiple)**
- **W – Write Single Register (single)**



## HINWEIS

Alle weiteren hier nicht beschriebenen Register dürfen keinesfalls verändert werden.

### 7.1 Bedeutung der einzelnen Bits in der Statusbitleiste (SYS\_Status):

Störungen und Fehlermeldungen können nachfolgender Tabelle mit Hilfe des Registers SYS\_Status identifiziert werden,

Bit	Name	Wert → Meldung
00	----	ohne Funktion (reserviert)
01	WARMUP	1 → BASIC <sup>EVO</sup> befindet sich in der Aufwärmphase (ca. 10 s)
02	SYS_ERR	1 → Systemfehler
03	----	ohne Funktion (reserviert)
04	----	ohne Funktion (reserviert)
05	STARTUP	1 → BASIC <sup>EVO</sup> befindet sich in der Hochlaufphase (ca. 40 s)
06	KORR	1 → Korrektur aktiv (immer)
07	MW_ok	1 → Nullpunkt wurde gesetzt
08	----	ohne Funktion (reserviert)
09	----	ohne Funktion (reserviert)
10	----	ohne Funktion (reserviert)
11	MW_aktiv	1 → Mittelwertbildung zur Driftkorrektur aktiv
12	EEP_ERR	1 → EEprom-Fehler
13	WDG_WRN	1 → nach Watchdog-Reset
14	POWER_ON	1 → Versorgungsspannung eingeschaltet
15	OUT_OF_RANGE	1 → wenn „0x000A Konz“ < -10% FS oder „0x000A Konz“ > 110% FS

Tabelle 6: Zuordnung der Fehlermeldungen in der Statusbitleiste SYS\_Status

FS= (Full Scale) Messbereichsendwert



## HINWEIS

Der Wert 0 steht immer für den (fehlerfreien) Normalzustand.

Die beiden Bits 6 (KORR) und 7 (MW\_ok) sind interne Flags, die im Fertigungsprozess der einzelnen BASIC<sup>EVO</sup> gesetzt werden. Sie dienen auch der Qualitätskontrolle und werden auf den Wert „1“ gesetzt, wenn das jeweilige BASIC<sup>EVO</sup> temperaturkompensiert wurde und die Kalibrierung durchlaufen hat.

## 7.2 Beschreibung des Einheitenkenners:

Das Register Konz (0x000A) liefert einen Zahlenwert, der je nach Ausführung des BASIC<sup>EVO</sup> von seiner Skalierung und Einheit her variiert. Mithilfe des Registers Einheit (0x004F) lässt sich der Konzentrationswert richtig berechnen. Folgend wird daher die Bedeutung des Zahlenwertes im Register Einheit (0x004F) dargestellt:

Registerwert	→	Einheit / Skalierung
0	→	unbelegt, für Sonderanwendungen
1	→	ppm x 0,01
2	→	ppm x 0,1
3	→	ppm
4	→	Vol.-% x 0,001
5	→	Vol.-% x 0,01
6	→	Vol.-% x 0,1
7	→	UEG x 0,01 %
8	→	UEG x 0,1 %

Tabelle 7: Zuordnung Registerwert zur Messeinheit und Multiplikator



### HINWEIS

Teilmengen von <1 Vol.-% werden meist als ppm – Wert angegeben, die folgende Tabelle zeigt das Verhältnis von Vol.-% zu ppm:

Vol.-%	ppm
100	1.000.000
10	100.000
1	10.000
0,1	1000
0,01	100
0,001	10
0,0001	1

Tabelle 8: Verhältnis von Vol.-% zu ppm

## 8 Hinweise zur Inbetriebnahme und Bedienung



### HINWEIS

Wir empfehlen für das Setzen von Nullpunkt und Endwert das smartGAS Calibration Tool. Dieses kann von der smartGAS Homepage kostenlos heruntergeladen werden.

### 8.1 Selbsttest + Warmup

Nach dem Einschalten des BASIC<sup>EVO</sup>- Sensors erfolgt ein interner Selbsttest, die grüne LED blinkt. Danach liefert der Sensor schon Messwerte und Systemfehler werden ausgewertet. Nach 40 Sekunden Aufwärmen gibt der BASIC<sup>EVO</sup> Sensor korrekte Messdaten aus.



### HINWEIS

Während des Selbsttestphase werden keine korrekten Messwerte ausgegeben.

## 8.2 Setzen des Nullpunkts

Das Setzen des Nullpunkts wird empfohlen

- nach Neuinstallation des Sensors bzw. Messsystems
- in regelmäßigen Intervallen (muss der Anwendung angepasst werden)
- nach Instandsetzungen/Wartungsarbeiten am Sensor oder des Messsystems



### HINWEIS

Vor dem Abgleich des Nullpunktes muss der Sensor mindestens 30 Minuten in Betrieb sein, der Sensor muss mit einem Nullgas (z.B. N<sub>2</sub> – 100 Vol.-%) so lange durchströmt werden, bis die Anzeige für die Gaskonzentration einen stabilen Wert erreicht hat.

Sind die o.g. Voraussetzungen erfüllt, wird in das Register IR\_4tagneu (0x0047) der Wert 1 geschrieben und so der Nullpunkt neu gesetzt.

## 8.3 Setzen des Endpunktes

Das Setzen des Endpunktes (auch Endwert- oder Span- Kalibrierung genannt) bedingt die Verwendung eines Prüfgases, welches mit möglichst hoher Genauigkeit dem Messbereichs- Endwert des zu kalibrierenden Sensors entspricht.

Es gelten hier die gleichen Vorbedingungen wie beim Setzen des Nullpunktes: der Sensor muss mindestens 30 Minuten in Betrieb sein und solange mit dem Prüfgas durchströmt werden, bis sich im Register Konz (0x000A) ein stabiler Wert eingestellt hat.

Sind alle Voraussetzungen erfüllt, wird in das Register Span (0x0054) der Korrekturwert für den jeweiligen Messkanal geschrieben.

## 8.4 Berechnung des Korrekturwertes für den Endpunkt

Angenommen, ein Sensor zeigt bei Beaufschlagung mit einem Prüfgas, welches den Wert von 1003 ppm hat (hier „Konz\_Kal“ genannt), eine Konzentration von nur 978 ppm (hier „Konz\_Alt“ genannt) an.

Das Auslesen des Register Span ergibt den Wert 9985 (hier „Span\_Alt“ genannt).

Die Neuberechnung des Korrekturwertes für das Register Span erfolgt dann folgendermaßen:

$$\text{Span\_Neu} = \text{Konz\_Kal} \times \text{Span\_Alt} / \text{Konz\_Alt}$$

$$\text{Span\_Neu} = 1003 \times 9985 / 978 = \mathbf{10240}$$

Nun wird der neue Wert von **10240** in das Register Span (0x0054) geschrieben, und der Vorgang ist beendet!



### HINWEIS

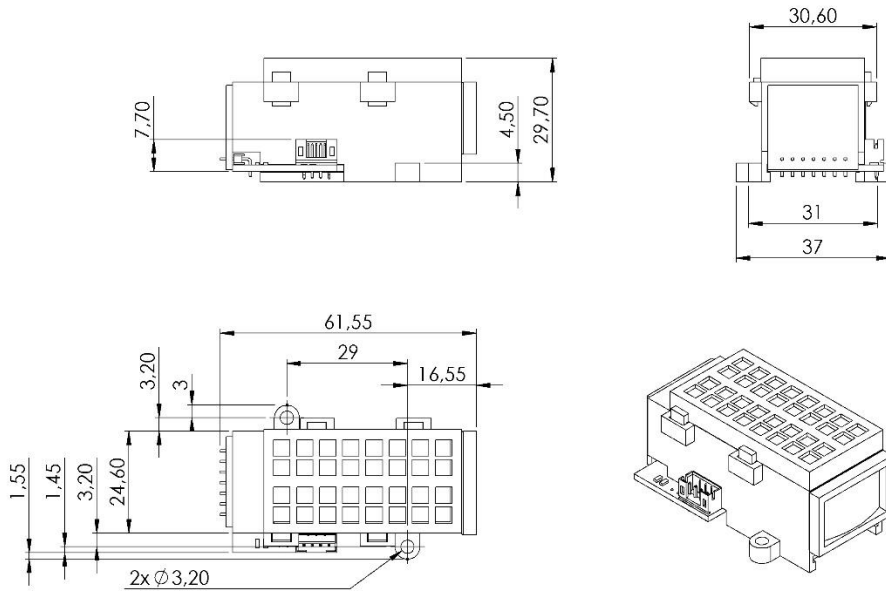
Führen Sie immer zuerst eine Nullpunktjustierung mit Ihrem smartGAS Sensor durch und danach eine Endpunktjustierung.

## **8.5 Wiederherstellen der Kalibrierparameter auf Werkseinstellung**

Zur Wiederherstellung der Kalibrierparameter auf Werkseinstellung können die Register „IR\_4tagneu“ und „Span“ neu beschrieben werden. Hierfür muss der Registerwert aus „fab\_zero\_value“ in „IR\_4tagneu“ und „fab\_span\_value“ in „Span“ geschrieben werden.

## 9 Anhang

### 9.1 Mechanische Abmessungen [mm]



## 9.2 Betrieb des BASIC<sup>EVO</sup> an einem Mikrocontroller

Zur Kommunikation des BASIC<sup>EVO</sup> mit einem Mikrocontroller müssen die Signale am COM-Pin des Moduls vom Pegel her an den Mikrocontroller angepasst werden. Dies geschieht am einfachsten unter Benutzung eines RS485-Schnittstellenbausteins, der gemäß Spannungsversorgung des Mikrocontrollers ausgewählt werden muss.

Am Mikrocontroller müssen die UART-Signale TXD (Transmit Data), RXD (Receive Data) und ein Signal für das Aktivieren des Senders TXEN (Transmitter Enable) bereitgestellt werden. Das folgende Bild zeigt die Schaltung:

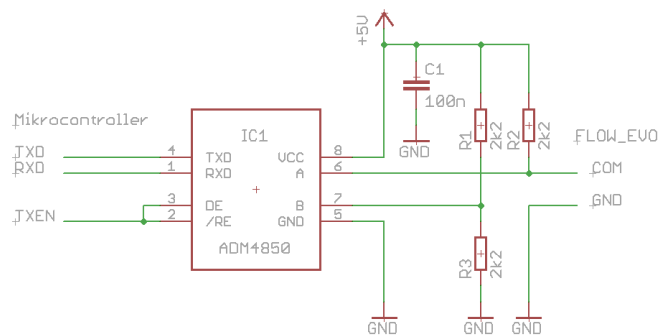


Abbildung 1: BASIC<sup>EVO</sup> an einem Mikrocontroller

Die Schaltung ist für einen Mikrocontroller mit 5 V Betriebsspannung ausgelegt. Bei Betrieb mit 3,3 V ist anstatt des ADM4850 (bzw. Vergleichstypen) ein ADM3075 (bzw. Vergleichstypen) zu verwenden. Alle anderen Bauteile sind nicht betroffen.

Es können bis zu 16 BASIC<sup>EVO</sup> parallel mit dieser Schaltung betrieben werden. Voraussetzung sind unterschiedliche Modbus-Adressen der Geräte.



### HINWEIS

Es ist zu beachten, dass BASIC<sup>EVO</sup> zusätzlich zum oben abgebildeten Kommunikations-Anschluss an eine Stromversorgung angeschlossen werden muss.

### 9.3 Betrieb des BASIC<sup>EVO</sup> an einem PC

Zum Betrieb eines BASIC<sup>EVO</sup> an einem PC kann bei smartGAS ein speziell dafür vorgesehener USB-Adapter inkl. Software als Zubehör bezogen werden (Artikel-Nr. Z6-000025). Die Speisung des BASIC<sup>EVO</sup> erfolgt über die USB-Schnittstelle, eine zusätzliche Stromversorgung ist nicht erforderlich. Das folgende Bild zeigt einen solchen Adapter:



Abbildung 2: USB-Adapter für Betrieb BASIC<sup>EVO</sup> am PC

### 9.4 Einbau des BASIC<sup>EVO</sup> in einer Modulbox mit Gasadapter

Zur Adaptierung des BASIC<sup>EVO</sup> mit den Kalibriergasen kann die unten abgebildete Modulbox und der passende Gasadapter bei uns bezogen werden:



BASIC<sup>EVO</sup> Gasadapter (Z7-000001) und Modulbox (Z7-000004)



## 10 Impressum

Die in dieser Beschreibung Bilder und Zeichnungen können vom Original abweichen, sie dienen lediglich der Illustration.

Alle Angaben – auch technische Spezifikationen – können ohne Vorankündigung geändert werden.

Alle Bilder und Grafiken in diesem Handbuch: © 2021 smartGAS Mikrosensorik GmbH, Heilbronn.



©2021 smartGAS Mikrosensorik GmbH

smartGAS Mikrosensorik GmbH | Hündlerstr. 1 | 74080 Heilbronn | Germany

phone: +49 7131/797553-0 | fax: +49 7131/797553-10 | [www.smartgas.eu](http://www.smartgas.eu) | [mail@smartgas.eu](mailto:mail@smartgas.eu)

Edition 05/11\_22