Modbus-Handbuch
für eviateg-Geräte

Handbuch Version 1.1

Firmwareversion 5.90

Norderstedt, 20.02.2025

eviateg GmbH

Kleistring 8

D-24558 Henstedt-Ulzburg

Internet: https://www.eviateg.de

E-Mail: info@eviateg.de

Inhaltsverzeichnis

1 Modbus-Unterstützung durch eviateg-Geräte

Ab der Firmware-Version 5.84 bieten eine Großzahl von eviateg-Störmeldegeräten eine Modbus-Schnittstelle in den Betriebsarten

- Modbus/TCP Server
- Modbus/RTU Server
- Modbus/ASCII Server
- Modbus/TCP Client
- Modbus/RTU Client
- Modbus/ASCII Client

Die unterstützten Betriebsarten richten sich nach der jeweiligen Geräte-Hardware und -Ausstattung.

1.1 Modbus-Server

In der Server-Betriebsart wird das eviateg-Störmeldegerät **durch einen Client angesteuert** (typischerweise eine SPS), um z.B.

- SMS und Sprachnachrichten
- Push-Nachrichten
- Threema-Nachrichten
- VdS-Nachrichten

zu verschicken,

- SMS zu empfangen

oder

- 230V-Relais zu schalten
- 230V-Eingänge abzufragen

Außerdem bieten die eviateg-Geräte in den Server-Betriebsarten eine **Watchdog-Funktion**, so dass beim Ausbleiben einer Ansteuerung durch die SPS automatisch Störmeldungen verschickt werden können.

1.2 Modbus-Client

In den Client-Betriebsarten überwacht das eviateg-Störmeldegerät mit bis zu **20 Modbus-Wächtern** zyklisch ausgewählte Register in einem oder mehreren Geräten mit Modbus-Unterstützung auf

- Abweichung von einem Sollwert
- Erreichen eines Sollwertes
- Überschreiten eines Sollwertes
- Unterschreiten eines Sollwertes

und führt die jeweils zugeordnete Reaktion aus (z.B. zum Senden einer Sprachnachricht).

Damit kann in einigen Fällen eine SPS mit Interface-Karten eingespart werden.

1.3 Unterstützte Geräte

Gerät	Modbus/TCP	Modbus/RTU	Modbus/ASCII	Bemerkung
IT35I	-	√	√	
IT35G	-	√	√	
IT70	-	√	√	mit RS232-Option
IT190	√	-	-	mit LAN-Board
IT191	√	-	-	mit LAN-Board
IT251I	-	√	√	
IT251GI	-	√	√	
IT251LI	√	√	√	

1.4 Modbus/TCP

Modbus/TCP ist auf eviateg-Geräten mit LAN-Schnittstelle ab Firmware-Version 5.84 verfügbar.

Geräte der Familien IT190 und IT191 können bei Bedarf mit der Option "LAN" (Artikelnummer 480016) ab Werk geliefert werden.

Der für Modbus/TCP verwendete IP-Port kann mit der Konfigurationssoftware "CONNY" auf der Eigenschaftenseite "Modbus" eingestellt werden. Als Werkseinstellung wird Port 502 verwendet.

1.5 Modbus über RS232

Geräte der Familien IT35 und IT251 sind von Haus aus mit RS232-Schnittstellen ausgestattet; Geräte der IT70-Familie können mit der Option "RS232" ab Werk geliefert werden.

Die RS232-Schnittstellen sind jeweils als DCE (Data Communication Equipment) ausgeführt.

Belegung der DB9-Buchse:

Pin	Name	Bezeichnung	Signal-Richtung	Verwendung bei Modbus
2	RxD	Empfangsdaten	eviateg-Gerät → SPS	Ja
3	TxD	Sendedaten	SPS → eviateg-Gerät	Ja
4	DTR	Betriebsbereitschaft	SPS → eviateg-Gerät	Optional
7	RTS	Sendeanforderung	SPS → eviateg-Gerät	Optional
5	GND	Signalerde		Ja

Als Baudrates sind 300, 600, 1200, 2400, 4800, 9600, 19200, 38400, 57600 und 115200 Baud einstellbar.

Als Parity kann zwischen None, Even und Odd gewählt werden.

Wenn "No Parity" eingestellt ist, werden automatisch 2 Stopbits verwendet, anderenfalls nur 1 Stopbit.

Modbus/ASCII verwendet eine Zeichenlänge von 7 Bits, Modbus/RTU von 8 Bits.

Die seriellen Parameter können mit der Konfugurationssoftware "CONNY" auf der Eigenschaftenseite "Modbus" eingestellt werden.

Hinweis:

Da die seriellen Modbus-Parameter auf der Eigenschaftenseite "Modbus" eingestellt werden, sollte die Eigenschaftenseite "RS232-Zusatzfunktion" **NICHT verwendet** werden.

1.5.1 Besonderheit bei IT251-Geräten

IT251 sind intern mit einem Multiplexer ausgestattet, der den Empfänger-Eingang (TxD, Sendedaten der SPS) **automatisch** zwischen der DB9-Buchse und der USB-Buchse umschaltet.

Damit Modbus-Daten über die DB9-Buchse empfangen werden können, **muss die DTR-Leitung (Pin 4) aktiv sein**. Wenn die SPS die DTR-Leitung nicht setzt, kann im Anschlusstecker die DSR-Leitung (Pin 6) auf Pin 4 gebrückt werden.
Achten Sie in diesem Fall darauf, dass die DTR-Leitung zwischen der SPS und diesem modifizierten Anschlusstecker unterbrochen wird, damit keine Ausgänge gegeneinander treiben !

Bei Bedarf kann ein passendes Kabel über die eviateg GmbH bezogen werden.

1.6 Modbus über RS485

Modbus über RS485-Schnittstellen (Zweidraht bzw. Vierdraht) werden zur Zeit noch nicht unterstützt.

2 Modbus aktivieren / deaktivieren

2.1 Modbus über RS232

Modbus-Server und -Client müssen (**speziell** für Modbus/RTU und Modbus/ASCII **über** eine **RS232-Schnittstelle**) aktiviert bzw. deaktiviert werden.

Bei **Anschluss über eine RS232-Schnittstelle** wird diese typischerweise auch zum Programmieren des Gerätes mit CONNY verwendet. Daher muss das Gerät zwischen Programmier- und Modbus-Betrieb unterscheiden können.

Wenn das angeschlossene Gerät (z.B. eine SPS) die **RTS-Leitung nicht setzt**, kann mit CONNY auf der Eigenschaftenseite "Modbus" die automatische Aktivierung freigegeben werden. Alternativ auch ein Anschlusskabel verwendet werden, bei dem die RTS-Leitung nicht zum eviateg-Gerät durchverbunden ist.

Bei Bedarf kann ein passendes Kabel über die eviateg GmbH bezogen werden.

Wenn das angeschlossene Gerät die RTS-Leitung setzt, sollte die automatische Aktivierung nicht freigegeben werden. Statt dessen sollten Eingangs-Reaktionen verwendet werden, die die Aktionen "**Modbus Enable**" bzw. "**Modbus Disable**" ausführen.

2.2 Modbus/TCP

Bei Modbus/TCP werden Verbindungen auf dem eingestellten Port (typ. 502) nur angenommen, wenn Modbus aktiviert ist. Daher sollte mit CONNY auf der Eigenschaftenseite "Modbus" die automatische Aktivierung freigegeben werden.

3 Inputs, Coils und Register

Als "**Discrete Inputs**" werden bei Modbus digitale Eingänge und Werte bezeichnet, deren Zustand (0 oder 1) nur gelesen werden kann.

Als "**Coils**" werden digitale Ausgänge bezeichnet, deren Zustand (0 oder 1) sowohl gelesen, als auch gesetzt werden kann.

Mit "**Input Register**" werden 16-Bit-Werte bezeichnet, deren Inhalt nur gelesen werden kann. Häufig werden mehrere Input-Register zusammengefasst, um z.B. 32-Bit-Werte oder Fließkommzahlen verarbeiten zu können (siehe unten).

Mit "**Holding Register**" werden 16-Bit-Werte bezeichnet, deren Inhalt gelesen und geschrieben werden kann. Auch Holding-Register werden häufig zusammengefasst, um z.B. 32-Bit-Werte oder Fließkommzahlen verarbeiten zu können (siehe unten).

3.1 Unterstützte Datentypen

Folgende Datentypen werden von den eviateg-Geräten sowohl in der Server- als auch in der Client-Betriebsart unterstützt:

- 16 Bit ("UINT16")
- 32 Bit ("UINT32"), hierbei werden zwei Register zusammengefasst
- 64 Bit ("UINT64"), hierbei werden vier Register zusammengefasst
- einfache Fließkommazahl ("FLOAT" bzw. "FLOAT32"), hierbei werden zwei Register zusammengefasst
- Fließkommazahl mit doppelter Genauigkeit ("DOUBLE" bzw. "FLOAT64"), hierbei werden vier Register zusammengefasst

3.2 Unterscheidung Big Endian / Little Endian ("WordSwapping")

Modbus verwendet laut Definition den "Big Endian Mode", bei dem die höherwertigen 16 Bit eines 32-Bit-Wertes zuerst übertragen werden:

Beispiel:

Register 50000: 4711
Register 50001: 0815

Im Normalfall "BigEndian" werden diese Bytes in der Reihenfolge 47110815 übertragen. Im "Little Endian Mode" werden diese Bytes in der Reihenfolge 08154711 übertragen.

Die Unterscheidung zwischen beiden Verfahren kann mit CONNY auf der Eigenschaftenseite "Modbus" getroffen werden.

3.3 Byte-Reihenfolge ("ByteSwapping")

In Sonderfällen muss auch die Reihenfolge der Bytes eines 16-Bit-Registers getauscht werden. In diesem Fall werden die Bytes aus dem obenstehenden Beispiel in der Reihenfolge 11471508 übertragen (und um die Verwirrung komplett zu machen, in der Reihenfolge 15081147 mit ByteSwapping und WordSwapping).

ByteSwapping kann mit CONNY auf der Eigenschaftenseite "Modbus" gewählt werden.

4 Abfrage- und Steuermöglichkeiten in der Server-Betriebsart

4.1 Implementierte Funktionscodes

In der Server-Betriebsart können die eviateg-Geräte mit folgenden Funktionscodes angesprochen werden:

Code (dez)	Code (hex)	Bits	Bezeichnung	Verwendung
01	01	1	Read Coils	Lesen des digitalen Zustands von Ausgängen
02	02	1	Read Discrete Inputs	Lesen des digitalen Zustands von Eingängen, Temperatur-Wächtern, Spannungssensoren, Spannungs-Wächtern und Analogwert-Wächtern (soweit auf dem Gerät vorhanden)
03	03	n*16	Read Holding Registers	Lesen eines Schreib-/Lese-Registers
04	04	16	Read Input Register	Lesen eines Lese-Registers
05	05	1	Write Single Coil	Setzen/Zurücksetzen eines Ausgangs
06	06	16	Write Single Register	Schreiben eines Schreib-/Lese-Registers
15	0F	n*1	Write Multiple Coils	Setzen/Zurücksetzen mehrerer Ausgänge
16	10	n*16	Write Multiple Registers	Schreiben mehrerer Schreib-/Lese-Register
17	11		Report Server ID	Abfrage der Gerätebezeichnung und der Firmware-Version

4.2 Discrete Inputs

In dieser Kategorie sind per Funktioncode 02 abfragbar:

- Zustand der digitalen Eingänge
- Zustand von Temperaturwächtern *)
- Zustand von Spannungs-Sensoren *)
- Zustand von Spannungs-Wächtern *)
- Zustand von Analogwert-Wächtern *)
- Einbuchzustand des GSM-Modems *)
- GPRS-Einbuchzustand *)

*) sofern vom jeweiligen Gerät unterstützt

Eine Auflistung der Input-Nummern und ihrer Bezeichnung kann mit dem Befehl `Modbus Structure` abgefragt werden (z.B. mit CONNY auf der Eigenschaftenseite "Modbus" und der Schaltfläche "Modbus-Struktur anzeigen").

4.3 Coils

In dieser Kategorie sind die digitalen Ausgänge (Relais) des Gerätes per Funktionscode 01 (Read Coils) abfragbar sowie per Funktionscode 05 (Write Single Coil) und 15 (Write Multiple Coils) steuerbar

Eine Auflistung der Coil-Nummern und ihrer Bezeichnung kann mit dem Befehl `Modbus Structure` abgefragt werden (z.B. mit CONNY auf der Eigenschaftenseite "Modbus" und der Schaltfläche "Modbus-Struktur anzeigen").

4.4 Input Register

In dieser Kategorie sind folgende statische 16-Bit-Register per Funktionscode 04 abfragbar (soweit vom jeweiligen Gerät unterstützt):

- das GSM-Netzwerk
- die GSM-Empfangsqualität
- die Seriennummer des Gerätes
- die Spannungen bzw. Ströme von Analog-Eingängen
- die Temperaturen von Temperatursensoren
- die Spannungen von Spannungssensoren
- die GSM-Empfangsfeldstärke in dBm
- das Guthaben der Prepaid-Karte
- die Einbuchstatistik des GSM-Modems
- der Gerätename
- die Firmware-Version
- Datum und Uhrzeit der Firmware
- die GSM-Rufnummer
- die Anzahl empfangener SMS
- die Anzahl ungelesener SMS

Eine Auflistung der InputRegister-Nummern und ihrer Bezeichnung können mit dem Befehl `Modbus Structure` abgefragt werden (z.B. mit CONNY auf der Eigenschaftenseite "Modbus" und der Schaltfläche "Modbus-Struktur anzeigen").

Einige dieser Register sind zu 32-Bit-Werten zusammengefasst (siehe Abschnitt 4.6 "Datentypen und Registerbereiche", Seite 12).

4.5 Holding Register

In dieser Kategorie sind 16-Bit-Register per Funktionscode 03 abfragbar und per Funktionscode 06 (Write Single Register) bzw. 16 (Write Multiple Register) beschreibbar.

Anwendungsfälle sind z.B.

- Befehlslänge für Befehle an das eviateg-Gerät
- Rückmeldungslänge für Rückmeldungen vom eviateg-Gerät

- das Watchdog-Register (siehe Abschnitt 6, Seite 16)
- Datum und Uhrzeit
- Buffer für Befehle und Rückmeldungen

Eine Auflistung der HoldingRegister-Nummern und ihrer Bezeichnung können mit dem Befehl `Modbus Structure` abgefragt werden (z.B. mit CONNY auf der Eigenschaftenseite "Modbus" und der Schaltfläche "Modbus-Struktur anzeigen").

Einige dieser Register sind zu 32-Bit-Werten zusammengefasst (siehe Abschnitt 4.6 "Datentypen und Registerbereiche", Seite 12).

4.6 Datentypen und Register-Bereiche

4.6.1 16 Bit ("UINT16")

Dieser Datentyp wird z.B. für die Darstellung des GSM-Netzes und der GSM-Empfangsqualität sowie für Befehls- und Rückmeldungslänge verwendet.

Register-Bereiche: Input-Register 0 bis 99 und Holding-Register 1000 bis 1099

4.6.2 32 Bit ("UINT32")

Bei diesem Datentyp werden zwei aufeinanderfolgende 16-Bit-Register zu einem 32-Bit-Wert zusammengefasst. Er wird z.B. für die Seriennummer des Gerätes verwendet.

Register-Bereiche: Input-Register 100 bis 199 und Holding-Register 1100 bis 1199

4.6.3 64 Bit ("UINT64")

Bei diesem Datentyp werden vier aufeinanderfolgende 16-Bit-Register zu einem 64-Bit-Wert zusammengefasst. Er wird z.Zt. noch nicht verwendet.

Register-Bereiche: Input-Register 200 bis 299 und Holding-Register 1200 bis 1299

4.6.4 Einfache Fließkommazahl ("FLOAT")

Dieser Datentyp wird in zwei Registern abgelegt und z.B. für Temperaturen, Analog-Messwerte, die GSM-Feldstärke und das Prepaid-Guthaben verwendet. Die Kodierung erfolgt nach IEEE 754.

Register-Bereiche: Input-Register 300 bis 399 und Holding-Register 1300 bis 1399

4.6.5 Fließkommazahl mit doppelter Genauigkeit ("DOUBLE")

Bei diesem Datentyp werden vier aufeinanderfolgende 16-Bit-Register zu einer 64-Bit-Fließkommazahl zusammengefasst. Er wird z.Zt. noch nicht verwendet.

Register-Bereiche: Input-Register 400 bis 499 und Holding-Register 1400 bis 1499

4.6.6 16-Bit-Arrays

Mit diesem Datentyp werden mehrere 16-Bit-Register zu einer logischen Einheit zusammengefasst, z.B. für die GSM-Einbuchstatistik (die aus sieben 16-Bit-Werten für die

Einbuchvorgänge des Gerätes besteht) oder Datum/Uhrzeit (aus sechs 16-Bit-Werten).

Register-Bereiche: Input-Register 500 bis 599 und Holding-Register 1500 bis 1599

4.6.7 Text

Dieser Datentyp ähnelt den 16-Bit-Arrays, nur dass die beteiligten Register im unteren Byte ein ASCII-Zeichen beinhalten. Er wird z.B. für den Gerätenamen, die Firmware-Version und die GSM-Rufnummer verwendet.

Register-Bereiche: Input-Register 600 bis 699 und Holding-Register 1600 bis 1699

4.6.8 Hinweise zu den Werten

Das GSM-Netz wird als 2 (2G), 3 (UMTS) oder 4 (LTE) angegeben.

Die **GSM-Empfangsqualität** wird mit Werten von 0 (schlecht) bis 31 (ausgezeichnet) angegeben.

Die **GSM-Empfangsfeldstärke** wird in dBm als Fließkommzahl angegeben (z.B. -83).

Das **Guthaben der Prepaid-Karte** wird in € angeben (z.B. 3,35).

Die **GSM-Einbuchstatistik** enthält im ersten Register (z.B. 500) die Einbuchvorgänge von vor sechs Tagen und im letzten Register (z.B. 506) des laufenden Tages.

Datum und Uhrzeit enthalten im

- ersten Register (z.B. 1500) das Jahr (vierstellig, z.B. 2024),
- zweiten Register den Monat
- dritten Register den Tag
- vierten Register die Stunde
- fünften Register die Minute
- sechsten Register die Sekunde

Wenn die Register aufeinanderfolgend beschrieben werden, wird die interne Uhrzeit des Gerätes in dem Moment gestellt, in dem das Sekunden-Register beschrieben wird.

Die Anzahl der empfangenen SMS gibt an, wie viele der zehn Speicherplätze für empfangene SMS belegt sind, die Anzahl der ungelesenen SMS hingegen, wie viele empfangenen SMS noch nicht abgefragt worden sind (siehe Abschnitt 7, Seite 17).

5 Übertragen von Befehlen an das Gerät per Modbus

Mit dem nachstend beschriebenen Verfahren können vom Client Befehle an das eviateg-Gerät übermittelt werden, z.B. zum

- Versenden von SMS und Sprachnachrichten
- Ändern von Meldungstexten
- Ändern von Zielrufnummern

5.1 Verwendete Register

Die Übertragung von Befehlen nutzt folgende Register:

- Register 1000 für die Befehlslänge ("Command Length")
- Register 2000 bis 2511 als Befehlspuffer ("Command Buffer")
- Register 1001 für die Rückmeldungslänge ("Response Length")
- Register 3000 bis 27999 als Rückmeldungspuffer ("Response Buffer")

Hinweis: die hier genannten Registernummern können sich ggf. bei späteren Firmware-Versionen ändern. Maßgebend sind immer die Registernummern, die mit dem Befehl `Modbus Structure` angezeigt werden.

5.2 Ablauf der Befehlsübertragung

Die Befehlsübertragung läuft in mehreren Schritten ab:

1. Der Client trägt mit der Funktion "Write Multiple Registers" in die Register 2000ff den gewünschten Befehl ein, z.B. "Macro 1".
   Die maximale Befehlslänge beträgt 512 Byte.
   Der Befehl kann mit dem CR-Zeichen (0x0D bzw. 13) abgeschlossen werden.
   Dabei werden nur die unteren acht Bit der jeweiligen 16Bit-Register genutzt.
2. Der Client trägt mit der Funktion "Write Single Register" in das Register 1000 die Länge des Befehls ein (in diesem Beispiel 8).
3. Das eviateg-Gerät setzt das Register 1001 auf den Wert 0, führt den Befehl aus und trägt in die Register 3000ff die Antwort ein.
   Auch hierbei werden nur die unteren acht Bit der jeweiligen 16Bit-Register genutzt.
4. Das Gerät trägt die Länge der Antwort in Register 1001 ein (in diesem Beispiel 6).
5. Der Client fragt nach einer plausiblen Zeit (z.B. eine halbe Sekunde) mit der Funktion "Read Holding Register" das Register 1001 ab und wiederholt diesen Schritt, bis das Register einen Wert größer 0 enthält.
6. Anschließend liest der Client mit der Funktion "Read Multiple Registers" die entsprechende Anzahl Bytes der Antwort aus den Registern 3000ff aus (in diesem Beispiel <CR><LF>OK<CR><LF>).
7. Zum Abschluss setzt der Client das Register 1001 wieder auf 0. Dadurch werden die im Gerät verwendeten Ressourcen wieder freigegeben.

Eine Wireshark-Aufzeichnung dieses Beispiels ist unter
https://www.eviateg.de/Modbus/Macro 1.pcapng

abrufbar. Es empfiehlt sich, in der Filterzeile oben im WireShark-Fenster "modbus" als Filter zu setzen.

5.3 Ablauf der Befehlsübertragung bei langen Antworten

Da die Anzahl der Register, die mit der Funktion "Read Multiple Registers" ausgelesen werden können, auf 125 begrenzt ist, muss der Client eine entsprechende Anzahl von "Read Multiple Register"-Funktionen ausführen, um die Rückmeldung aus Register 3125, 3250 etc. abzuholen.

Ein Beispiel für den Befehl "gsm ?" mit allen Informationen zum GSM-Betrieb ist unter https://www.eviateg.de/Modbus/GSM-Informationen.pcapng abrufbar.

5.4 Ablauf der Befehlsübertragung bei langen Befehlen

Wenn die Länge eines Befehls (z.B. Sprachalarmierungen) 125 Bytes übersteigt, muss der Client die Bytes in Register 2125, 2250 etc. eintragen, bevor er die Länge im Register 1000 eingeträgt.

5.5 Anregungen für Sprachalarmierungen

Wenn im eviateg-Gerät mit der CONNY-Funktion "Wortliste erzeugen" auf der Eigenschaftenseite "Sprachalarmierung" ALLE in Frage kommenden Wörter als individuelle Sprachblöcke gespeichert werden, kann per Modbus-Befehl eine flexible, der Sitution angepasste Sprachalarmierung gesendet werden.

Beispiel:

```
SendVoice DTMF 123 Meyer;Müller;Schulze 5 Pumpe drei Strom zu hoch
```

Für dieses Beispiel müssen also mindestens die fünf Sprachblöcke "Pumpe", "drei", "Strom", "zu" und "hoch" im Gerät gespeichert sein.

6 Watchdog-Funktion

Bei Bedarf kann das eviateg-Gerät in der Server-Betriebsart eine angeschlossene SPS mit Hilfe einer Watchdog-Funktion überwachen.

Dazu muss die SPS innerhalb eines einstellbaren Intervalls (Werkseinstellung: eine Minute) das Register 1002 mit einem jeweils wechselnden Wert beschreiben.

Wenn der Wert 0 in Register 1002 geschrieben wird, wird der Watchdog abgeschaltet.

Wenn die Watchdog-Zeit abgelaufen ist, wird das Ereignis "Modbus Watchdog" ausgelöst und die zugehörige Reaktion ausgeführt (z.B. Versenden einer SMS).

Das Intervall und die Reaktion können mit CONNY auf der Eigenschaftenseite "Modbus" eingestellt werden.

7 Abfrage von empfangenen SMS

Sobald das eviateg-Gerät eine SMS empfangen hat, die nicht mit einem entsprechenden Passwort als Steuer-SMS gekennzeichnet ist, wird diese SMS in einem von zehn SMS-Speicherplätzen abgelegt. Wenn alle zehn Plätze belegt sind, wird die SMS verworfen.

7.1 Lesen neuer SMS

Das Lesen neuer SMS ist mit dem Befehl "sms read new" entsprechend Abschnitt 5 möglich:

```
sms read new
Msg#: 0
Sent: 2025-02-20 12:08:30
Rcvd: 2025-02-20 12:08:22 MEZ
From: +494060848790
Text: Test 1

OK
```

Die Nachrichtennummer in der ersten Zeile gibt den Index des Speicherplatzes an.

7.2 Löschen von SMS

Eine einzelne SMS kann durch z.B. "sms erase 0" gelöscht werden.

```
sms erase 0
OK
```

Der gesamte SMS-Speicher kann mit "sms erase *" gelöscht werden.

```
sms erase *
SMS storage initialized for 10 messages

OK
```

7.3 Auflisten aller SMS

Eine Liste aller SMS kann mit "sms list" abgefragt werden.

```
sms list
0: !2025-02-20 12:35:18 - +494060848790 - Test 3
1: !2025-02-20 12:35:36 - +494060848790 - Test 4
OK
```

Ein Ausrufungszeichen vor dem Empfangsdatum zeigt an, dass die jeweilige SMS noch nicht gelesen worden ist.

Bei einem leeren SMS-Speicher wird "No message stored" ausgegeben.

```
sms list
No message stored
OK
```

8 Client-Betriebsart und Modbus-Wächter

In der Client-Betriebsart stellen die eviateg-Geräte 20 Wächter-Funktionen zur Verfügung, mit denen zyklisch Discrete Inputs, Coils, Input Register und Holding-Register überwacht werden können.

Das Abfrage-Intervall ist für jeden Modbus-Wächter individuell von einer Sekunde bis zu zehn Minuten einstellbar.

Als Pendant zur Entprellung von Eingängen kann auch für jeden Modbus-Wächter angegeben werden, ab der wievielten Abfrage ein Wert als geändert betrachtet werden soll.

Jedem Modbus-Wächter kann ein Name zugewiesen werden, der z.B. mit Textmakros in SMS verwendet werden kann.

Die Parametrierung der Modbus-Wächter erfolgt mit CONNY auf der **Eigenschaftenseite "Modbus-Wächter"**, die über die Eigenschaftenseite "Modbus" aufgerufen werden kann.

In der Betriebsart "Modbus/TCP Client" können bei Bedarf auch mehrere Modbus-Server abgefragt werden. Wenn der als Nächstes eingetaktete Modbus-Wächter die gleiche IP-Adresse verwendet, bleibt nach einer Abfrage die Verbindung bestehen; anderenfalls wird sie abgebaut.

8.1 Triggerung

Modbus-Wächter können triggern bei

- Änderung des Wertes
- Erreichen eines Sollwertes
- Verlassen eines Sollwertes
- Überschreiten eines Sollwertes
- Unterschreiten eines Sollwertes

Für den Fall, dass nicht auf kurzfristige Änderungen reagiert werden soll, kann eine Art "Tiefpass" angepasst werden, ab der wievielten Abfrage der Wächter triggern soll.

8.2 Wächter-Typen

Folgende Datentypen können überwacht werden:

- Discrete Inputs
- Coils
- 16-Bit-Input-Register bzw. Holding-Register
- 32-Bit-Werte (in zwei Input-Registern bzw. Holding-Registern)
- 64-Bit-Werte (in vier Input-Registern bzw. Holding-Registern)
- Fließkommazahlen (in zwei Input-Registern bzw. Holding-Registern)
- Fließkommazahlen mit doppelter Genauigkeit (in vier Input-Registern bzw. Holding-Registern)

- Register-Arrays (mit Prüfsummenbildung und -Vergleich über bis zu 125 Input- bzw. Holding-Register)

Das Format des Sollwertes richtet sich nach dem Wächter-Typ. Numerische Werte für Register können sowohl dezimal, als auch hexadezimal (mit dem Präfix "0x") eingegeben werden. Das zuletzt verwendete Format wird auch für die Anzeige des Sollwertes verwendet.

8.3 Modbus-Wächter-Reaktionen

Jedem Modbus-Wächter ist eine Reaktion zugeordnet, mit der z.B. eine SMS oder Sprachnachricht versendet werden kann.

Weiterhin gibt es eine Reaktion, die ausgelöst wird, wenn einer der Modbus-Wächter wieder in den Ausgangszustand zurückkehrt, z.B. wenn der Sollwert nicht mehr überschritten wird.

8.4 Textmakros für Modbus-Wächter

Das Textmakro "**&GN**" wird mit dem **Namen** des auslösenden Modbus-Wächters ersetzt (gilt für das Ereignis beim Triggern eines Modbus-Wächters und für das Ereignis bei der Rückkehr in den Ausgangszustand).

Das Textmakro "**&MC**" wird mit dem **zuletzt abgefragten Wert** des auslösenden Modbus-Wächters ersetzt (gilt für das Ereignis beim Triggern eines Modbus-Wächters und für das Ereignis bei der Rückkehr in den Ausgangszustand).

Ein Textmakro aus dem Bereich "**&MN0**" bis "**&MN19**" wird mit dem **Namen** des jeweiligen Modbus-Wächters ersetzt.

Ein Textmakro aus dem Bereich "**&MS0**" bis "**&MS19**" wird mit dem **Zustand** des jeweiligen Modbus-Wächters ("Ausgeloest" bzw. "Nicht ausgeloest") ersetzt.

Ein Textmakro aus dem Bereich "**&MV0**" bis "**&MVS19**" wird mit dem **Wert** des jeweiligen Modbus-Wächters ersetzt.

9 Beispiele für Modbus-Strukturen

Auf der Eigenschaftenseite "Modbus" kann mit der Konfigurationssoftware CONNY die Struktur des Modbus-Servers abgerufen werden (Schaltfläche "Modbus-Struktur anzeigen").

Die abgefragte Struktur wird im Unterverzeichnis "Modbus" des CONNY-Datenverzeichnisses (z.B. C:\ProgramData\eviateg GmbH\CONNY\Modbus) unter Verwendung des Gerätetyps, der Seriennummer und der jeweiligen Firmware abgelegt.

9.1 Struktur beim IT35G

```
Modbus structure of IT35G SN 208C0F FW 5.90
-------------------------------------------
Discrete inputs (ReadOnly):
00: In00
01: In01
02: GSM Registration
03: GPRS Registration
-------------------------------------------
Coils (ReadWrite):
00: Out00
-------------------------------------------
Input Registers (ReadOnly):
00000: GSM Network (16 Bit)
00001: GSM Signal Quality (16 Bit)
00002: Received SMS (16 Bit)
00003: Unread SMS (16 Bit)
00100..00101: Serial Number (32 Bit)
00300..00301: GSM Receive Level (Float 32 Bit)
00302..00303: Prepaid Credit (Float 32 Bit)
00500..00506: GSM Registration Statistic (Array of 16 Bit)
00600..00631: Device Name (Text)
00632..00635: Firmware Version (Text)
00636..00651: Firmware Build (Text)
00652..00671: GSM Number (Text)
-------------------------------------------
Holding Registers (ReadWrite):
01000: Command Length (16 Bit)
01001: Response Length (16 Bit)
01002: Watchdog (16 Bit)
01500..01505: Date & Time (Array of 16 Bit)
02000..02511: Command Buffer (Text)
03000..27999: Response Buffer (Text)
```

9.2 Struktur beim IT70

```
Modbus structure of IT70 SN 20B54D FW 5.90
-------------------------------------------
Discrete inputs (ReadOnly): 29
00: In00
01: In01
02: In02
03: In03
04: Pwr00
05: Pwr01
06: TMPGrd00
07: TMPGrd01
08: TMPGrd02
09: TMPGrd03
10: TMPGrd04
11: TMPGrd05
```

```
12: TMPGrd06
13: TMPGrd07
14: TMPGrd08
15: ADCGrd00
16: ADCGrd01
17: ADCGrd02
18: ADCGrd03
19: ADCGrd04
20: ADCGrd05
21: ADCGrd06
22: ADCGrd07
23: ADCGrd08
24: ADCGrd09
25: ADCGrd10
26: ADCGrd11
27: GSM Registration
28: GPRS Registration
----------------------------------------
Coils (ReadWrite): 2
00: Out00
01: Out01
----------------------------------------
Input Registers (ReadOnly): 21
00000: GSM Network (16 Bit)
00001: GSM Signal Quality (16 Bit)
00002: Received SMS (16 Bit)
00003: Unread SMS (16 Bit)
00100..00101: Serial Number (32 Bit)
00300..00301: InA (Float 32 Bit)
00302..00303: InB (Float 32 Bit)
00304..00305: InC (Float 32 Bit)
00306..00307: InD (Float 32 Bit)
00308..00309: T0 (Float 32 Bit)
00310..00311: T1 (Float 32 Bit)
00312..00313: Vmain (Float 32 Bit)
00314..00315: Vbatt (Float 32 Bit)
00316..00317: Ibatt (Float 32 Bit)
00318..00319: GSM Receive Level (Float 32 Bit)
00320..00321: Prepaid Credit (Float 32 Bit)
00500..00506: GSM Registration Statistic (Array of 16 Bit)
00600..00631: Device Name (Text)
00632..00635: Firmware Version (Text)
00636..00651: Firmware Build (Text)
00652..00671: GSM Number (Text)
----------------------------------------
Holding Registers (ReadWrite): 4+512+25000
01000: Command Length (16 Bit)
01001: Response Length (16 Bit)
01002: Watchdog (16 Bit)
01500..01505: Date & Time (Array of 16 Bit)
02000..02511: Command Buffer (Text)
03000..27999: Response Buffer (Text)
```

9.3 Struktur beim IT190 / IT191

```
Modbus structure of IT191 SN 30BA64 FW 5.90
----------------------------------------
Discrete inputs (ReadOnly): 20
00: In00
01: In01
02: In02
03: In03
04: In04
05: In05
06: In06
07: Pwr00
```

```
08: Pwr01
09: TMPGrd00
10: TMPGrd01
11: TMPGrd02
12: ADCGrd00
13: ADCGrd01
14: ADCGrd02
15: ADCGrd03
16: ADCGrd04
17: ADCGrd05
18: GSM Registration
19: GPRS Registration
-----------------------------------------
Coils (ReadWrite): 2
00: Out00
01: Out01
-----------------------------------------
Input Registers (ReadOnly): 17
00000: GSM Network (16 Bit)
00001: GSM Signal Quality (16 Bit)
00002: Received SMS (16 Bit)
00003: Unread SMS (16 Bit)
00100..00101: Serial Number (32 Bit)
00300..00301: InA (Float 32 Bit)
00302..00303: InB (Float 32 Bit)
00304..00305: T0 (Float 32 Bit)
00306..00307: Vmain (Float 32 Bit)
00308..00309: Vbatt (Float 32 Bit)
00310..00311: GSM Receive Level (Float 32 Bit)
00312..00313: Prepaid Credit (Float 32 Bit)
00500..00506: GSM Registration Statistic (Array of 16 Bit)
00600..00631: Device Name (Text)
00632..00635: Firmware Version (Text)
00636..00651: Firmware Build (Text)
00652..00671: GSM Number (Text)
-----------------------------------------
Holding Registers (ReadWrite): 4+512+25000
01000: Command Length (16 Bit)
01001: Response Length (16 Bit)
01002: Watchdog (16 Bit)
01500..01505: Date & Time (Array of 16 Bit)
02000..02511: Command Buffer (Text)
03000..27999: Response Buffer (Text)
```

9.4 Struktur beim IT251

```
Modbus structure of IT251GI SN 20A60F FW 5.90
-----------------------------------------
Discrete inputs (ReadOnly): 36
00: In00
01: In01
02: In02
03: In03
04: In04
05: In05
06: In06
07: In07
08: In08
09: In09
10: In10
11: In11
12: In12
13: In13
14: In14
15: In15
16: In16
```

```
17: In17
18: In18
19: In19
20: Pwr00
21: Pwr01
22: Pwr02
23: Pwr03
24: Pwr04
25: TMPGrd00
26: TMPGrd01
27: TMPGrd02
28: TMPGrd03
29: TMPGrd04
30: TMPGrd05
31: TMPGrd06
32: TMPGrd07
33: TMPGrd08
34: GSM Registration
35: GPRS Registration
-------------------------------------------
Coils (ReadWrite): 6
00: Out00
01: Out01
02: Out02
03: Out03
04: Out04
05: Out05
-------------------------------------------
Input Registers (ReadOnly): 18
00000: GSM Network (16 Bit)
00001: GSM Signal Quality (16 Bit)
00002: Received SMS (16 Bit)
00003: Unread SMS (16 Bit)
00100..00101: Serial Number (32 Bit)
00300..00301: Tmp0 (Float 32 Bit)
00302..00303: Tmp1 (Float 32 Bit)
00304..00305: Tmp2 (Float 32 Bit)
00306..00307: Vmain (Float 32 Bit)
00308..00309: Vbatt (Float 32 Bit)
00310..00311: Ibatt (Float 32 Bit)
00312..00313: GSM Receive Level (Float 32 Bit)
00314..00315: Prepaid Credit (Float 32 Bit)
00500..00506: GSM Registration Statistic (Array of 16 Bit)
00600..00631: Device Name (Text)
00632..00635: Firmware Version (Text)
00636..00651: Firmware Build (Text)
00652..00671: GSM Number (Text)
-------------------------------------------
Holding Registers (ReadWrite): 4+512+25000
01000: Command Length (16 Bit)
01001: Response Length (16 Bit)
01002: Watchdog (16 Bit)
01500..01505: Date & Time (Array of 16 Bit)
02000..02511: Command Buffer (Text)
03000..27999: Response Buffer (Text)
-------------------------------------------
```

10 Historie

Version	Datum	Änderungen
0.1	29.08.2024	Erste Version
0.2	13.09.2024	• Bemerkungen zur Tabelle "Unterstützte Geräte" • weitere Erläuterungen zu Modbus/TCP, Modbus/RTU und Modbus/ASCII • weitere Erläuterungen zu Modbus über RS232, insbesondere zum IT251 • Überarbeitung der Abschnitte "Input Register" und "Holding Register" • neuer Abschnitt "Unterstützte Datentypen" • Abschnitt "Übertragung von Befehlen" komplett überarbeitet • neue Abschnitte "Watchdog-Funktion" "Beispiele für Modbus-Strukturen" • Stichwortverzeichnis
1.0	18.10.2024	Beschreibung der Betriebsart "Modbus-Client", der Modbus-Wächter und der Textmakros
1.1	20-02-2025	Neu: Input-Register zum Abfragen empfangener SMS, neu: Abschnitt 7 Abschnitt 9 aktualisiert

Stichwortverzeichnis