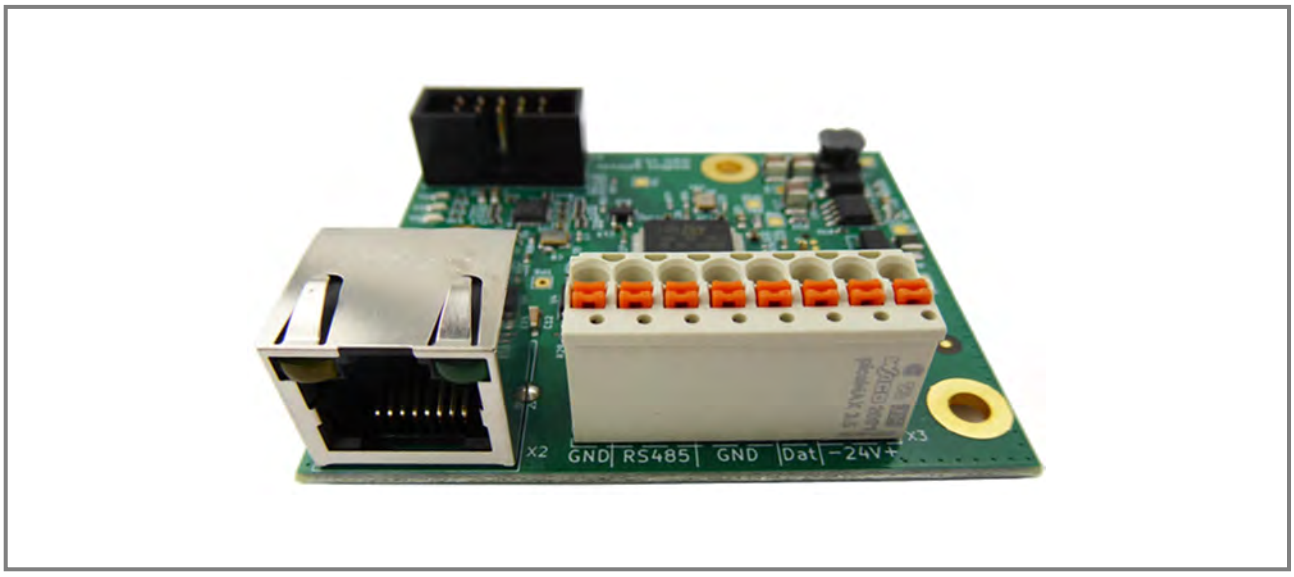
Dokumentation

Modbus-Modul

Anleitung für die Anwendungsentwicklung



Vor Beginn aller Arbeiten bitte Anleitung lesen
Stand: 11/2026
Art.-Nr.: 648980

Inhalt

Inhalt **2**
1. Allgemeines **3**
1.1 Dokumenthistorie 3
1.2 Firmware-Versionsprotokoll 3
1.3 Begriffe und Abkürzungen 4
2. Modbus TCP-Schnittstelle **5**
3. Modbus RTU-Schnittstelle **6**
4. Unterstützte Modbus-Funktionen **7**
5. Register-Map **8**
5.1 Allgemeines 9
5.2 Gerätespezifische Angaben 11
6. Einstieg **14**
6.1 Modbus Poll 14
6.2 Python-Beispiel 14
7. Firmware-Update **16**
8. Zurücksetzung auf Werkseinstellungen **18**

1. Allgemeines

1.1 Dokumenthistorie

Datum	Änderungen	Autor
2026-02-11	Erstversion, enthält Register-Map v1	Steinkamp
2026-04-21	• Register-Map v2 • Register für die Baudraten- und Paritätseinstellungen der RTU hinzugefügt • Beschreibung zur Register-Map-Übersicht hinzugefügt • Dokumentation unterstützter Modbus-Funktionen	Steinkamp
2026-07-07	• Release Firmware 0.6.1 • Hinzufügen von Anmerkungen zu TCP/IP	Steinkamp
2026-08-05	• Release Firmware 0.6.3	Steinkamp
2026-10-30	• Release Firmware 0.6.4 • Dokumentation RS-485 GND pin	Steinkamp

1.2 Firmware-Versionsprotokoll

Version	Änderungen
0.4.0	• Unterstützung der von „TFÜ“-Geräten übermittelten Luftfeuchtigkeits- und Temperaturwerte • Einführung Register-Map v1: o Vorbereitung zur Einbindung der MCU Core-Temperatur in den Info-Bereich o Kürzen des Geräte-Startadressen-Arrays auf max. 32 Geräte
0.5.0	• Seriennummern-Register beschreibbar gemacht. Wenn auf null gesetzt, wird die über den seriellen Geräteausgang empfangene Seriennummer verwendet.
0.5.1	• Fehlerbehebung Trockenfilter-Warnung
0.6.0	• Register-Map v2: Modbus-RTU-Einstellungen für Parität und Baudrate hinzugefügt • Konfiguration von RTU-Parität und -Baudrate implementiert • Unterstützung für die Modbus-Funktionen „Read coils“ und „Read discrete inputs“ hinzugefügt
0.6.1	• IPv4 Default Gateway hinzugefügt

0.6.3	• Zusammenfassung der Pump-/Nachfüll-Bits zu einem Coil • Unterstützung Modbus-Funktionscode 0x10 „Write multiple registers“
0,6.4	• Widerstandsfähigere RTU-Kommunikation • Sicherheitskorrekturen für TCP

1.3 Begriffe und Abkürzungen

LSB	Niedrigstwertiges Bit (oder Byte)
MSB	Höchstwertiges Bit (oder Byte)
Wort	Zwei Byte

2. Modbus TCP-Schnittstelle

Geräte mit einer TCP/IP-Schnittstelle sind mit dem in gängiger Netzwerkinfrastruktur verwendeten RJ45-Standardanschluss (8P8C) ausgestattet.

Standardmäßig ist das Modul so konfiguriert, dass es über DHCP dynamisch eine IP-Adresse zugewiesen bekommen kann.

Aktuell wird nur IPv4 unterstützt.

3. Modbus RTU-Schnittstelle

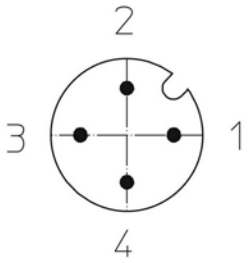
Mit einer RTU-Schnittstelle ausgestattete Geräte unterstützen die BASIC implementation class über einen M12-Anschluss.

Modus	RTU
Elektrische Schnittstelle	RS-485 (2W)
Baudraten	300, 600, 1200, 2400, 9600, 19200, 56000, 115200
Parität	keine, gerade, ungerade
RTU-Standardadresse	1
Anschlusstyp	A-codiert M12 (Stecker)
Abschluss	Es sind keine integrierten Abschlusswiderstände vorhanden; je nach Installation müssen externe Abschlusswiderstände hinzugefügt werden.

Die Standardwerte sind in obiger Auflistung unterstrichen.

Wie vom Modbus-Standard vorgeschrieben, beträgt die Stop-Bit-Länge sowohl bei ungerader als auch bei gerader Parität eins. Wenn die Parität auf „Keine“ gesetzt ist, ist das Stop-Bit zwei Bit lang.

Pinbelegung:



1: Nicht belegt
2: RS-485 A
3: GND
4: RS-485 B

4. Unterstützte Modbus-Funktionen

Das Modbus-Modul unterstützt folgende Modbus Function Codes:

0x01	Read coils
0x02	Read discrete inputs
0x03	Read holding registers
0x04	Read input registers
0x06	Write single register
0x10	Write multiple registers

Beim Empfang eines nicht unterstützten Modbus-Funktions-Codes antwortet das Modul mit einer Exception PDU.

Intern verwenden sowohl Coil als auch Register dieselbe Datenstruktur. Eine Coil-Adresse lässt sich wie folgt aus der Registeradresse und einem Bit-Index ableiten:

$$COIL_{address} = 16 \cdot REGISTER_{address} + BIT_{index}$$

5. Register-Map

Dieser Abschnitt enthält die Register-Map, die sowohl für TCP/IP als auch für RTU gilt.

Die Register-Map besteht aus einem Info-Bereich, einem beschreibbaren Konfigurationsbereich sowie mindestens einem Gerätebereich:

Info Section 0x0000	Config Section 0x011b	Device #0	Device #1	...	Device #16

In den meisten Anwendungsfällen ist nur ein Gerätebereich vorhanden. Bei Mehrkanal-Leckdetektoren können bis zu 17 Geräteteile vorhanden sein.

Im Info-Bereich können die Offset-Adressen der verfügbaren Gerätebereiche in den Registern „Geräte-Startadressen“ ausgelesen werden.

Die nachstehenden Registerzuordnungstabellen folgen diesen Konventionen:

- Die Länge (Spalte „L“) wird in Wörtern (= 2 Byte) angegeben, die die kleinste adressierbare Einheit darstellen.
- Berechtigungen in Spalte „P“:
  R steht für ‚schreibgeschützt‘ (Read-Only), RW für ‚Lese-/Schreibberechtigung‘ (Read/Write).
- Negative Werte werden im Zweierkomplement dargestellt.
- Bei Werten, die aus mehreren Wörtern bestehen, wird das MSB an der niedrigeren Adresse gespeichert und zuerst übertragen („Big-Endian“).

5.1 Allgemeines

5.1.1 Info-Bereich

Adresse	Name	L	P	Beschreibung
0	Gateway-Typ	1	R	0x0001: Dedizierte Hardware für Modbus-Modul 0x0002: Lemon (Display)
1	Anzahl Eingangs-kanäle	1	R	Maximale Anzahl an Eingangskanälen. Nur das LSB wird verwendet.
2	Firmware-Version des Moduls	2	R	0xMMmmPPRR MM: Major mm: Minor PP: Patch RR: 0x00 Release 0x11 Release-Kandidat 0x22 Beta 0x33 Alpha
4	Version Register-Map	1	R	0x0000: Erstmaliges Release 0x0001: FW >= v0.4 0x0002: FW >= v0.6
5	Geräteanzahl	1	R	Anzahl der SGB-Geräte, von denen gültige Daten empfangen werden. Nur das LSB wird verwendet.
6	Startadressen des Geräts	32	R	Inhaltsverzeichnis. Enthält so viele Startadressen, wie im Register für die Geräteanzahl angegeben sind. 0x0000 für leere Slots
38	Effektive IPv4-Adresse	2	R	Aktuell genutzte IPv4-Adresse. Entweder manuell oder über DHCP zugewiesen 0: Nicht zugewiesen, DHCP-Client aktiv oder ungültige Adresse konfiguriert
40	Effektive IPv4-Netmask	2	R	Aktuell genutzte IPv4-Netmask. Entweder manuell oder über DHCP zugewiesen 0: Nicht zugewiesen, DHCP-Client aktiv oder ungültige Adresse konfiguriert
42	Effektive IPv6-Adresse	8	R	Reserviert, nicht implementiert
50		16		Reserviert
64	Core-Temperatur	1	R	Nicht implementiert: Core-Temperatur des Geräte-Controllers (geringe Genauigkeit) Zweierkomplement der aktuellen Temperatur in °C
65		218		Reserviert

5.1.2 Konfigurationsbereich

Adresse	Name	L	P	Beschreibung
283	IPv4-Adresse	2	RW	Standardwert: 0xc0a8000a (192.168.0.10)
285	IPv4-Net-mask	2	RW	Standardwert: 0xffffff00 (255.255.255.0)
287	IPv4 Default Gateway	2	RW	Standardwert: 0xc0a80001 (192.168.0.1)
289	IPv4-Port	1	RW	Standardwert: 0x01f6 (dez: 502)
290	IPv6-Adresse	8	RW	Reserviert, nicht implementiert
298		8		Reserviert
306	IPv6-Port	1	RW	Reserviert, nicht implementiert Standardwert: 0x01f6 (dez: 502)
307	IP-Zuwei-sung	1	RW	0x0000: DHCP-Client aktiv, IPv4 (Standardwert) 0x0001: Statische IPv4-Adressen Rückgriff auf statischen Wert, wenn DHCP-Anforde-rung fehlschlägt. Erneuter Versuch nach dem Zurück-setzen, der Inhalt dieses Registers wird nicht verän-dert.
308	RTU-Adresse	1	RW	Server-Adresse für die RS-485-Schnittstelle. Nur das LSB wird verwendet. Bereich: 0x0001 .. 0x00FF (0x0000 reserviert für Broadcast). Standardwert: 0x0001
309	RTU-Baud-rate	1	RW	0x0003: 300 Baud 0x0006: 600 Baud 0x000C: 1200 Baud 0x0018: 2400 Baud 0x0060: 9600 Baud 0x00C0: 19200 Baud 0x0230: 56000 Baud 0x0480: 115200 Baud Standardwert: 0x0060 (9600 Baud)
310	RTU-Parität	1	RW	0x0000: Keine Parität (zwei Stop-Bits) 0x0001: Gerade Parität (ein Stop-Bit) 0x0002: Ungerade Parität (ein Stop-Bit) Standardwert: 0x0001 (Gerade Parität)

5.2 Gerätespezifische Angaben

In der ersten Spalte ist die absolute Adresse für einkanalige Modbus-Module angegeben. Die angegebenen Coil-Adressen können nur im Einkanalbetrieb verwendet werden.

Bei Mehrkanalgeräten (LDU, VLXE-SAB T../P..) muss zu der im Info-Abschnitt angegebenen Startadresse des Geräts der in der zweiten Spalte angegebenen Offset addiert werden. Die Coil-Adressen müssen anhand der oben angegebenen Formel berechnet werden.

Adresse	Offset	Name	L	P	Beschreibung
311	0	Eingangs-kanal	1	R	Der Kanal, über den das Modbus-Modul das SGB-Gerät empfängt
312	1	Serien-nummer des Ge-räts	2	RW	Seriennummer des Geräts. Kann bei Geräten überschrieben werden, die keine gültige Seriennummer übermitteln. Wenn die Seriennummer überschrieben wurde, bleibt sie erhalten, auch wenn von einem Gerät eine gültige Seriennummer empfangen wird. Dieses Verhalten lässt sich durch das Schreiben einer Null wiederherstellen.
314	3	Gerätetyp	1	R	0x0001: Druck-Leckanzeiger 0x0002: Vakuum-Leckanzeiger 0x0003: Flüssigkeitssensor
315	4	Geräte-anwen-dung	1	R	0x0001: Tank 0x0002: Rohr 0x00FF: Sonstige **Hinweis:** Wert ist je nach Typ des Leckanzeigers und der Firmware möglicherweise nicht zuverlässig.
316	5	Geräte-merkmale	1	R	**Bit** / **Coil** / **Beschreibung** 0 / 5056 / Feuchtigkeitssensor Trockenfilter 1 / 5057 / Pumpe vorhanden 2 / 5058 / Flüssigkeitsunterscheidung
317	6	Aktueller Druck	2	R	Zweierkomplement des Drucks in mbar
319	8	Alarm-druck	2	R	Gleiche Darstellung wie bei aktuellem Druck
323	12	Dicht-heits-anzeige	1	R	Dichtheitsanzeige (0–10). Nur LSB wird verwendet.

Adresse	Offset	Name	L	P	Beschreibung		
324	13	Alarm-status	1	R	**Bit**	**Coil**	**Beschreibung**
					0	5184	Alarm anliegend (‚High' bei anliegendem Alarm)
					1	5185	Akustischer Alarm (‚Low', wenn ein akustischer Alarm am Gerät bestätigt/stummgeschaltet wurde)
					2	5186	Druckalarm
					3	5187	Druckanstiegsalarm (nur VLXE und VL(R) (100–240V))
					4	5188	Magnetventilalarm
					5	5189	Flüssigkeitsalarm
					6	5190	Produktalarm (nur bei selektiven Flüssigkeitssensoren)
					7	5191	Wasseralarm (nur bei selektiven Flüssigkeitssensoren)
					14	5198	Allgemeiner externer Alarm
325	14	Wartungs-status	1	R	**Bit**	**Coil**	**Beschreibung**
					0	5200	Wartungs-Flag (‚High', wenn eines der folgenden Bits auf ‚High' steht)
					1	5201	Kontrolle fällig (nur Geräte mit „S")
					2	5202	Trockenfilterwarnung (nur Geräte mit „FC")
					3	5203	Trockenfilter-Sensoranschluss-Warnung (nur Geräte mit „FC")
					4	5204	Dichtheitswarnung

Adresse	Offset	Name	L	P	Beschreibung		
326	15	Betriebs-zustand	1	R	**Bit**	**Coil**	**Beschreibung**
					0	5216	Fehler
					2	5218	Befüllmodus (nur bei DLR-G und DLG)
					3	5219	Inbetriebnahmemodus (nur bei VLR (230V))
					8	5224	Nachfüllen (Pumpe in Betrieb oder Nachfüllventil geöffnet)
329	18	Firmware-Version	1	R	Firmware-Version des Leckanzeigers. Nur LSB wird verwendet.		
330	19	Protokoll-version	1	R	Protokollversion		

6. Einstieg

Dies ist die allgemeine Vorgehensweise zum Auslesen von Daten aus einem SGB-Gerät:

1. Startadresse des Geräteabschnitts ermitteln:
   Das Register „Gerätestartadressen“ mit einer Länge von 32 Wörtern auslesen. Alle Werte, die nicht Null sind, sind gültige Startadressen.
   Bei Einzel-Leckanzeigern gibt es nur eine Startadresse. Daher reicht es in den meisten Fällen aus, nur das erste Wort auszulesen.
2. Gerätedaten auslesen:
   Zum Auslesen des Gerätebereichs die zuvor ermittelte Startadresse zum Offset-Wert addieren, der in der Tabelle des Gerätebereichs angegeben ist.

Sobald die Startadresse bekannt ist, kann der erste Schritt übersprungen werden. Bei einkanaligen Geräten können der Einfachheit halber die oben aufgeführten absoluten Adressen verwendet werden.

6.1 Modbus Poll

Modbus Poll ist eine praktische Option für einen schnellen Einstieg. Um Ihnen den Einstieg noch weiter zu erleichtern, bieten wir auf unserer Website Konfigurationsdateien für Modbus Poll zum Herunterladen an:

sgb.de > Downloads > Modbus > Entwicklungstools

6.2 Python-Beispiel

Hier wird ein einfaches Beispiel mit Python und der Bibliothek `pymodbus` gezeigt, das den Zugriff auf Register und Coils veranschaulicht. Der Quellcode, einschließlich der importierten Datei mit den Konstanten, kann ebenfalls auf unserer Website heruntergeladen werden:

sgb.de > Downloads > Modbus > Entwicklungstools

Die importierte Datei mbgw_constants.py enthält die Adressen und Längen, die in den Registerzuordnungstabellen aufgeführt sind.

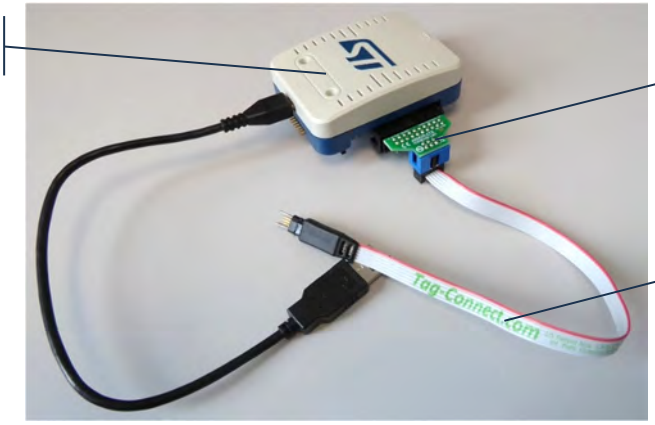
Nach dem Auslesen der Druck-Register wird die `pymodbus`-Funktion „`convert_from_registers()`“ verwendet, um aus den Rohdaten einen vorzeichenbehafteten Wert zu ermitteln.

Hierbei ist zu beachten, dass in diesem Beispiel nur der „Happy Flow“ ohne Ausnahmebehandlung gezeigt wird. Das Beispiel lässt sich mithilfe der Klasse `ModbusTcpClient` problemlos an TCP anpassen.

```
import mbgw_constants
from pymodbus.client import ModbusSerialClient

PORT = "COM3"
DEVICE_ID = 1

client = ModbusSerialClient(port=PORT, baudrate=9600, parity="E", stopbits=1)

# Fetch address offset of device (channel 0: application without mux)
# Since we only want the first device, we could set count=1 to read only a single register
response = client.read_holding_registers(address=mbgw_constants.REG_OFFSET_DEVICE_START_ADDRESSES,\
                                         count=mbgw_constants.REG_LEN_DEVICE_START_ADDRESSES,\
                                         device_id=DEVICE_ID)
device_offset = response.registers[0]
print("Device offset:", device_offset)

# Read current pressure
pressure_address = device_offset + mbgw_constants.REG_OFFSET_DEV_CURRENT_PRESSURE
response = client.read_holding_registers(address=pressure_address,\
                                         count=mbgw_constants.REG_LEN_DEV_CURRENT_PRESSURE,\
                                         device_id=DEVICE_ID)
pressure = ModbusSerialClient.convert_from_registers(response.registers,\
                                                     data_type=client.DATATYPE.INT32)
print("Pressure:", pressure, "mbar")

# Read pressure alarm coil
alarm_coil_address = 16 * (device_offset + mbgw_constants.REG_OFFSET_DEV_ALARM_STATUS) +\
    mbgw_constants.BIT_ALARM_STATUS_PRESSURE_ALARM
response = client.read_coils(address=alarm_coil_address, count=1, device_id=DEVICE_ID)
print("Alarm:", response.bits[0])

client.close()
```

7. Firmware-Update

Für ein Firmware-Update des Modbus-Moduls sind folgende Tools erforderlich:

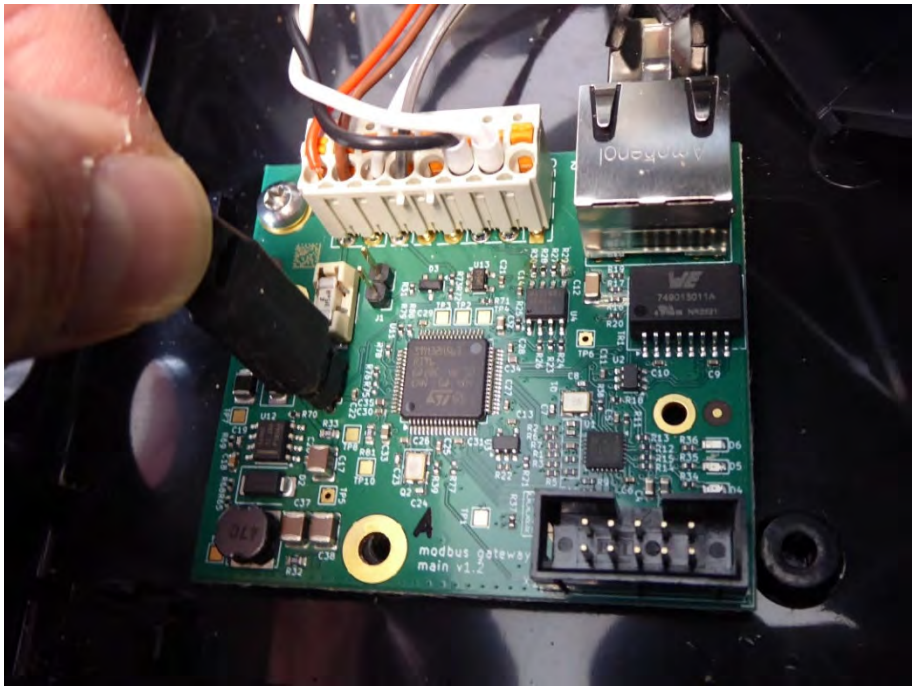


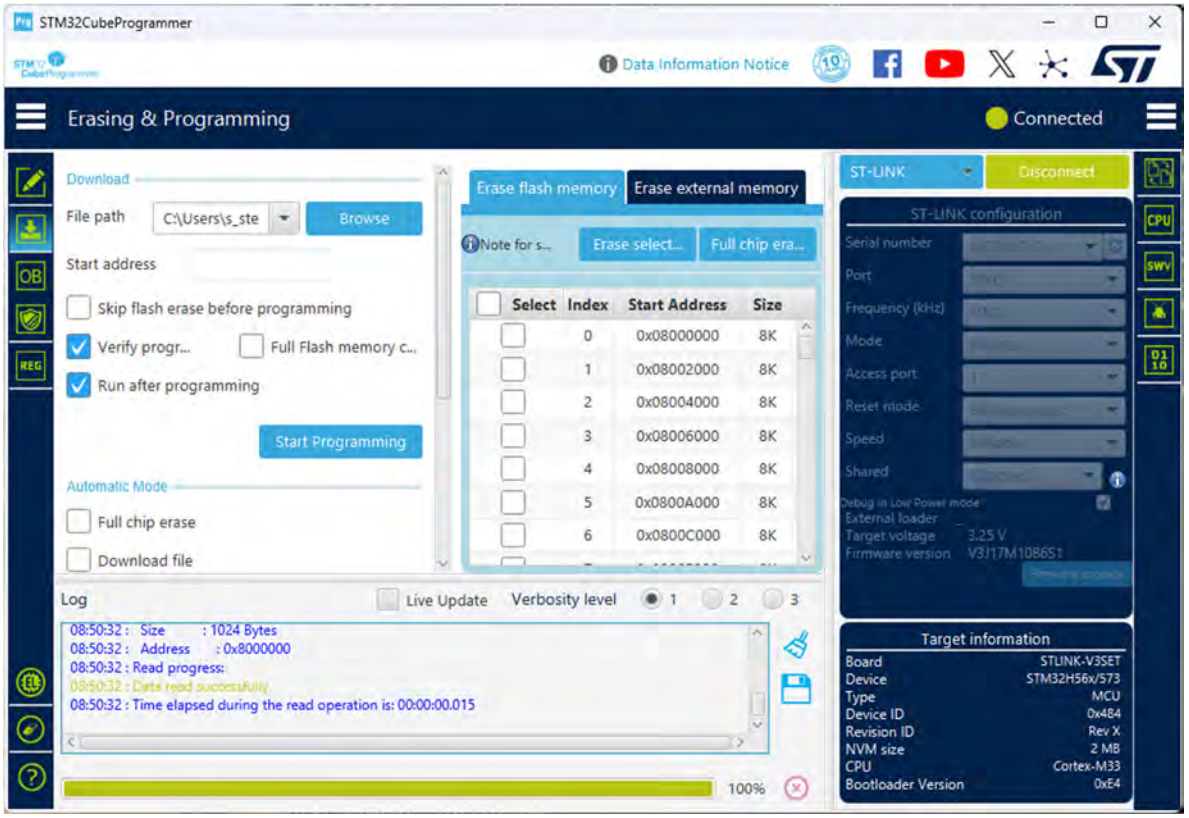
Programmierer	STLINK-V3SET	https://www.st.com/en/development-tools/stlink-v3set.html
Verbindungskabel 6-polig	Tag-Connect TC2030-IDC-NL	https://www.tag-connect.com/product/tc2030-idc-nl
Adapter für ST-Link	Tag-Connect Adapter ARM20-CTX 20-polig auf TC2030-IDC für Cortex	https://www.tag-connect.com/product/arm20-ctx-20-pin-to-tc2030-idc-adapter-for-cortex
STM32Cube-Programmer	Software	Kann kostenlos von der STMicro-Website heruntergeladen werden: https://www.st.com/en/development-tools/stm32cubeprog.html#get-software
Firmware-Binärdatei	modbus-module.elf	

Vorgehensweise:

1. Gerät spannungsfrei setzen.
2. Modbus-Modul PCB mit 12–24 V versorgen.
3. Das Programmierkabel an die Modbus-Modul-Leiterplattenbaugruppe anschließen.
4. Starten Sie STM32CubeProgrammer und wählen Sie links die Registerkarte „Erasing & Programming“ (Pfeil-nach-unten-Symbol) aus.
5. Klicken Sie auf „Durchsuchen“ und wählen Sie die Firmware-Binärdatei (*.elf) aus.
6. Die Kontrollkästchen “Verify programming” und “Run after programming” auswählen.

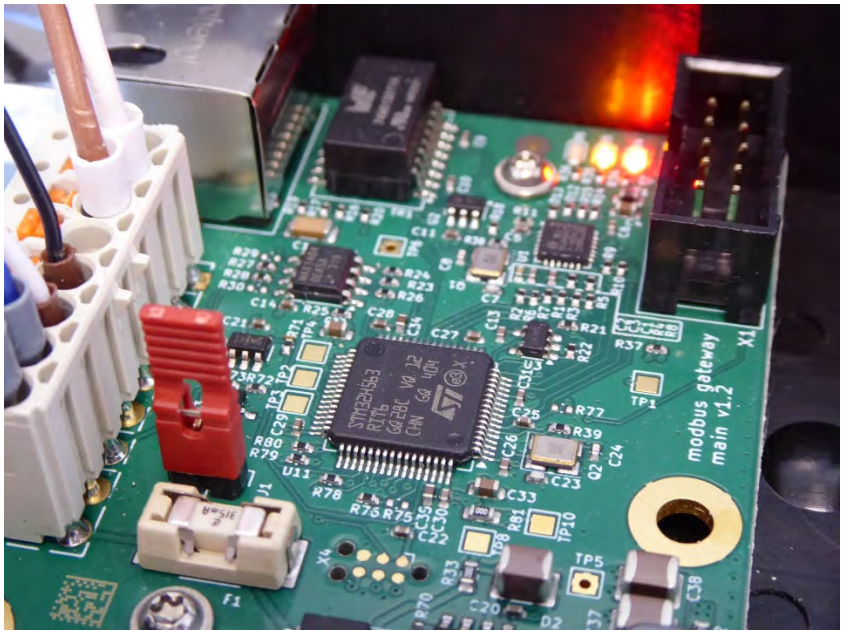
7. Oben rechts in der Ecke auf “Connect” klicken.
8. Auf “Start Programming” klicken.
9. Nach erfolgtem Programmiervorgang das Spannungsversorgungskabel trennen und den ursprünglichen Klemmenblock wieder verbinden.
10. Spannungsversorgung des Gerätes wiederherstellen.





8. Zurücksetzung auf Werkseinstellungen

Bei Bedarf kann das Modbus-Modul mit einem 2,54-mm-Kurzschluss-Jumper auf die Werkseinstellungen zurückgesetzt werden.



1. Gerät spannungsfrei setzen.
2. Die Stiftleiste wie oben abgebildet mit einem Jumper (wie er z.B. auf Computer-Mainboards zu finden ist) kurzschließen. Berührung mit anderen Teilen der Baugruppe vermeiden.
3. Spannungsversorgung wiederherstellen. Warten, bis die LED auf der Baugruppe nicht mehr blinkt. Das Modbus-Modul wurde mit den Standardeinstellungen neu gestartet.
4. Gerät erneut spannungsfrei setzen.
5. Jumper wieder entfernen.
6. Spannungsversorgung wiederherstellen.

Impressum

SGB GmbH
Hofstr. 10
57076 Siegen
Deutschland

T +49 271 48964-0
E sgb@sgb.de
W sgb.de | shop.sgb.de

Fotos und Skizzen sind unverbindlich für den Lieferumfang. Änderungen vorbehalten. © SGB GmbH, 11/2026