

---

## **RS485-NB – RS485/UART-zu-NB-IoT-Konverter Benutzerhandbuch**

Zuletzt geändert von Mengting Qiu

am 26.08.2024 um 16:31 Uhr

---

# Inhaltsverzeichnis

1. Einführung.....	5
1.1 Was ist ein RS485/UART-zu-NB-IoT-Konverter? .....	5
1.2 Funktionen.....	6
1.3 Spezifikationen .....	7
1.4 Anwendungen.....	7
1.5 Schlafmodus und Arbeitsmodus .....	8
1.6 Tasten und LEDs .....	8
1.7 BLE-Verbindung.....	8
1.8 Pin-Definitionen, Schalter und SIM-Richtung .....	9
1.8.1 Jumper JP2.....	9
1.8.2 BOOT-MODUS / SW1.....	9
1.8.3 Reset-Taste .....	9
1.8.4 Ausrichtung der SIM-Karte.....	10
1.8.5 SW2-Jumper (Defte UART-Pegel zu externem Sensor)	10
2. Verwenden Sie RS485-NB für die Kommunikation mit dem IoT-Server.....	10
2.1 Daten über NB-IoT-Netzwerk an IoT-Server senden.....	10
2.2 Gerät zum Lesen von RS485-Sensoren konfigurieren.....	11
2.2.1 Methode 1 – über das RS485-Konfigurationstool .....	11
2.2.2 Methode 2 – über AT-Befehle .....	12
2.2.3 Sensoren konfigurieren.....	15
2.2.4 Lesebefehle für jede Abtastung konfigurieren .....	15
2.2.5 Uplink, wenn der Sensor nicht reagiert .....	19
2.2.6 Uplink bei Unterbrechung .....	19
2.3 Nutzlasttypen.....	20
2.3.1 Allgemeines JSON-Format (Typ=5).....	20
2.3.2 HEX-Format Nutzlast (Typ=0) .....	21
2.4 Uplink testen und Aktualisierungsintervall ändern .....	22
2.5 Schneller Befehl zur Verarbeitung von MODBUS-Geräten .....	23
2.6 Zeitüberschreitung bei RS485-Befehl .....	23
2.7 RS485-Befehl löschen .....	23
2.8 Nutzlastversion festlegen .....	23
2.9 Steuerung der Ausgangsleistungsdauer .....	24
2.10 Sensorwert abrufen .....	24
2.11 +3V3-Ausgang .....	24
2.12 +5V Ausgang .....	24
2.13 Schalter Jumper .....	25
2.14 Taktprotokollierung (seit Firmware-Version v1.0.7) .....	25
2.15 Beispielabfrage gespeicherter historischer Datensätze .....	26
2.16 Abfrage des Uplink-Protokolls .....	27
2.17 Geplante Auflösung von Domänennamen .....	28
2.18 QoS-Stufe festlegen .....	28
3. RS485-NB konfigurieren.....	29
3.1 Methoden konfigurieren .....	29
3.2 Passwort für seriellen Zugriff.....	29
3.3 AT-Befehllssatz .....	31
4. Batterie und Stromverbrauch .....	32
5. Firmware-Update .....	33
6. Häufig gestellte Fragen .....	33
6.1 Wie kann ich auf die BC660K-GL AT-Befehle zugreifen? .....	33
6.2 Wie viele RS485-Slaves kann RS485-NB verbinden? .....	33
6.3 Wie konfiguriert man das Gerät über die MQTT-Abonnementfunktion? .....	33
6.4 Wie kann man RS485-NB verwenden, um eine Verbindung zu RS232-Geräten herzustellen? .....	34
6.5 Wie kann man feststellen, ob ein Problem mit dem eingestellten COMMAND vorliegt? .....	34
6.5.1 Einführung: .....	34

Benutzerhandbuch für LoRaWAN/NB-IoT-Endknoten – RS485-NB  
– RS485/UART-zu-NB-IoT-Konverter Benutzerhandbuch

6.5.2 PC für die Überwachung des RS485-Netzwerks mit dem Serial Tool einrichten .....	35
6.5.3 Mit ModRSsim2: .....	36
6.5.4 Beispiel – Testen des Befehls CFGDEV .....	39
6.5.5 Beispiel – Testen von CMD-Befehlssätzen .....	40
6.5.6 Testen mit PC .....	41
7. Bestellinformationen .....	43
8. Verpackungsinformationen .....	43
9. Support .....	44



## Inhaltsverzeichnis:

- 1.
  - [1.1 Was ist ein RS485/UART-zu-NB-IoT-Konverter?](#)
  - [1.2 Funktionen](#)
  - [1.3 Spezifikationen](#)
  - [1.4 Anwendungen](#)
  - [1.5 Schlafmodus und Arbeitsmodus](#)
  - [1.6 Tasten und LEDs](#)
  - [1.7 BLE-Verbindung](#)
  - [1.8 Pin-Definitionen, Schalter und SIM-Ausrichtung](#)
    - [1.8.1 Jumper JP2](#)
    - [1.8.2 BOOT-MODUS / SW1](#)
    - [1.8.3 Reset-Taste](#)
    - [1.8.4 Ausrichtung der SIM-Karte](#)
    - [1.8.5 Jumper SW2 \(Definiert UART-Pegel für externen Sensor\)](#)

- [2. Verwenden Sie RS485-NB für die Kommunikation mit dem IoT-Server.](#)
  - [2.1 Daten über das NB-IoT-Netzwerk an den IoT-Server senden](#)
  - [2.2 Gerät zum Lesen von RS485-Sensoren konfigurieren](#)
    - [2.2.1 Methode 1 – über das RS485-Konfigurationstool](#)
    - [2.2.2 Methode 2 – über AT-Befehle](#)
      - [2.2.2.1 Konfigurieren der UART-Einstellungen für RS485- oder TTL-Kommunikation](#)
    - [2.2.3 Konfigurieren der Sensoren](#)
    - [2.2.4 Konfigurieren der Lesebefehle für jede Abtastung](#)
    - [2.2.5 Uplink, wenn der Sensor nicht reagiert](#)
    - [2.2.6 Uplink bei Unterbrechung](#)
  - [2.3 Nutzlasttypen](#)
    - [2.3.1 Allgemeines JSON-Format \(Typ=5\)](#)
    - [2.3.2 HEX-Format Nutzlast \(Typ=0\)](#)
  - [2.4 Test-Uplink und Aktualisierungsintervall ändern](#)
  - [2.5 Schnellbefehl zur Steuerung von MODBUS-Geräten](#)
  - [2.6 RS485-Befehlszeitüberschreitung](#)
  - [2.7 RS485-Befehl löschen](#)
  - [2.8 Nutzlastversion einstellen](#)
  - [2.9 Dauer der Ausgangsleistung steuern](#)
  - [2.10 Sensorwert abrufen](#)
  - [2.11 +3V3-Ausgang](#)
  - [2.12 +5V-Ausgang](#)
  - [2.13 Jumper umschalten](#)
  - [2.14 Uhrzeitprotokollierung \(seit Firmware-Version v1.0.7\)](#)
  - [2.15 Beispielabfrage gespeicherter historischer Datensätze](#)
  - [2.16 Abfrage des Uplink-Protokolls](#)
  - [2.17 Geplante Auflösung von Domänennamen](#)
  - [2.18 Festlegen der QoS-Stufe](#)
  - [3. Konfiguration von RS485-NB](#)
    - [3.1 Konfigurationsmethoden](#)
    - [3.2 Passwort für seriellen Zugriff](#)
    - [3.3 AT-Befehlssatz](#)
  - [4. Akku und Stromverbrauch](#)
  - [5. Firmware-Update](#)
  - [6. Häufig gestellte Fragen](#)
    - [6.1 Wie kann ich auf die AT-Befehle des BC660K-GL zugreifen?](#)
    - [6.2 Wie viele RS485-Slaves kann RS485-NB verbinden?](#)
    - [6.3 Wie konfiguriere ich das Gerät über die MQTT-Abonnementfunktion?](#)
    - [6.4 Wie kann ich RS485-NB verwenden, um eine Verbindung zu RS232-Geräten herzustellen?](#)
    - [6.5 Wie kann ich feststellen, ob ein Problem mit dem eingestellten Befehl vorliegt?](#)
      - [6.5.1 Einführung:](#)
      - [6.5.2 PC für die Überwachung des RS485-Netzwerks mit dem Serial Tool einrichten](#)
      - [6.5.3 Mit ModRSsim2:](#)
      - [6.5.4 Beispiel – Testen des Befehls CFGDEV](#)
      - [6.5.5 Beispiel – Testen der CMD-Befehlssätze.](#)
      - [6.5.6 Testen mit dem PC](#)
  - [7. Bestellinformationen](#)
  - [8. Verpackungsinformationen](#)
  - [9. Support](#)

# 1. Einführung

## 1.1 Was ist ein RS485/UART-zu-NB-IoT-Konverter?

Der Dragino RS485-NB ist ein **RS485/UART-zu-NB-IoT-Konverter** für Internet-of-Things-Lösungen. Der Benutzer kann einen RS485- oder UART-Sensor an den RS485-NB-Konverter anschließen und den RS485-NB so konfigurieren, dass er regelmäßig Sensordaten ausliest und über das NB-IoT-Netzwerk an den IoT-Server hochlädt.

RS485-NB kann an einen RS485-Sensor, einen 3,3 V/5 V UART-Sensor oder einen Interrupt-Sensor angeschlossen werden. RS485-NB bietet **einen 3,3-V-Ausgang** und **einen 5-V-Ausgang** zur Stromversorgung externer Sensoren. Beide Ausgangsspannungen sind regelbar, um den Gesamtstromverbrauch des Systems zu minimieren.

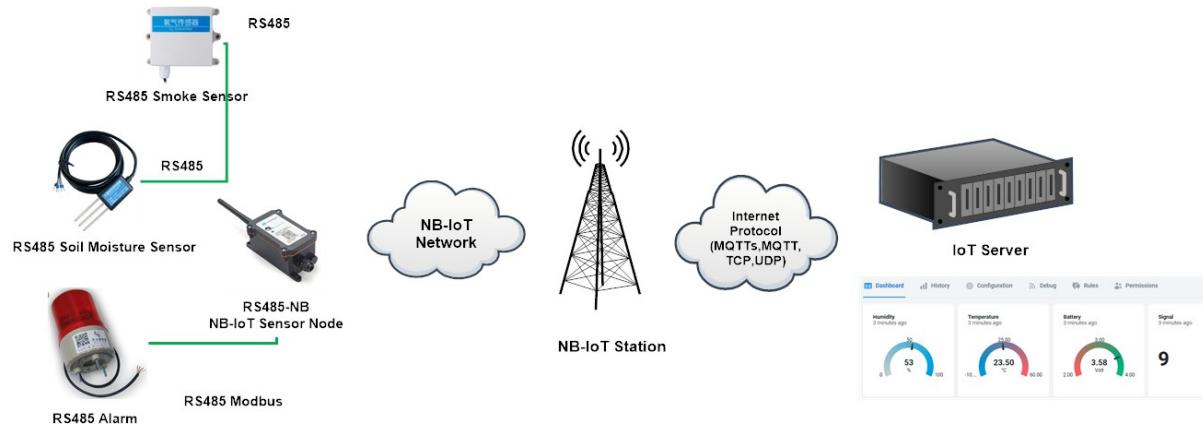
RS485-NB unterstützt verschiedene Uplink-Methoden, darunter **MQTT, MQTTs, UDP und TCP** für unterschiedliche Anwendungsanforderungen, sowie Uplinks zu verschiedenen IoT-Servern.

RS485-NB unterstützt **BLE-Konfiguration und OTA-Updates**, was die Bedienung für den Benutzer vereinfacht.

RS485-NB ist **wasserdicht** gemäß IP67 und wird mit **einem 8500-mAh-Li-SOCl2-Akku** betrieben. Es ist für den langfristigen Einsatz von bis zu mehreren Jahren ausgelegt.

RS485-NB verfügt über eine optionale integrierte SIM-Karte und eine Standardversion für die Verbindung mit dem IoT-Server. Dadurch ist es mit einer einfachen Konfiguration einsatzbereit.

#### RS485-NB in a NB-IoT Network



## 1.2 Funktionen

- NB-IoT-Bänder: B1/B2/B3/B4/B5/B8/B12/B13/B17/B18/B19/B20/B25/B28/B66/B70/B85 @H-FDD
- Extrem geringer Stromverbrauch
- Unterstützt mehrere RS485-Geräte durch flexible Regeln
- Unterstützung des Modbus-Protokolls
- Unterstützt Interrupt-Uplink
- Unterstützt den Anschluss von UART-Sensoren mit 3,3 V oder 5 V
- Mehrfachabtastung und ein Uplink
- Unterstützt Bluetooth v5.1 für die Fernkonfiguration und Aktualisierung der Firmware
- Uplink über MQTT, MQTTs, TCP oder UDP
- Regelmäßige Uplink-Verbindung
- Downlink zum Ändern der Konfiguration
- 8500-mAh-Akku für langfristigen Einsatz
- Nano-SIM-Kartensteckplatz für NB-IoT-SIM

## 1.3 Spezifikation

### Allgemeine Gleichstrom-Eigenschaften:

- Versorgungsspannung: 2,5 V bis 3,6 V
- Betriebstemperatur: -40 bis 85 °C

### E/A-Schnittstelle:

- Batterie-steuerbarer Ausgang (2,6 V bis 3,6 V, abhängig von der Batterie)
- +5 V steuerbarer Ausgang
- 1 x RS485-Schnittstelle
- 1 x UART-Schnittstelle, 3,3 V oder 5 V
- 1 x Interrupt- oder Digital-IN/OUT-Pins
- 1 x I2C-Schnittstelle
- 1 x Ein-Draht-Schnittstelle

### NB-IoT-Spezifikation:

#### NB-IoT-Modul: BC660K-GL

### Unterstützte Frequenzbänder:

- B1 @H-FDD: 2100 MHz
- B2 @H-FDD: 1900 MHz
- B3 @H-FDD: 1800 MHz
- B4 @H-FDD: 2100 MHz
- B5 @H-FDD: 860 MHz
- B8 @H-FDD: 900 MHz
- B12 @H-FDD: 720 MHz
- B13 @H-FDD: 740 MHz
- B17 @H-FDD: 730 MHz
- B18 @H-FDD: 870 MHz
- B19 @H-FDD: 870 MHz
- B20 @H-FDD: 790 MHz
- B25 @H-FDD: 1900 MHz
- B28 @H-FDD: 750 MHz
- B66 @H-FDD: 2000 MHz
- B70 @H-FDD: 2000 MHz
- B85 @H-FDD: 700 MHz

### Akku:

- Li/SOCl2 nicht wiederaufladbarer Akku
- Kapazität: 8500 mAh
- Selbstentladung: <1 % / Jahr bei 25 °C
- Maximaler Dauerstrom: 130 mA
- Maximaler Boost-Strom: 2 A, 1 Sekunde

### Leistungsaufnahme

- STOP-Modus: 10 uA bei 3,3 V
- Maximale Sendeleistung: 350mA@3.3v

## 1.4 Anwendungen

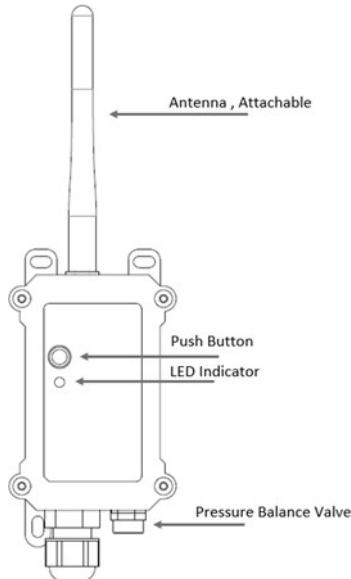
- Intelligente Gebäude und Hausautomation
- Logistik und Lieferkettenmanagement
- Intelligente Messung
- Intelligente Landwirtschaft
- Intelligente Städte
- Intelligente Fabrik

## 1.5 Schlafmodus und Arbeitsmodus

**Tiefschlafmodus:** Der Sensor hat kein NB-IoT aktiviert. Dieser Modus wird für die Lagerung und den Versand verwendet, um die Batterielebensdauer zu verlängern.

**Arbeitsmodus:** In diesem Modus arbeitet der Sensor als NB-IoT-Sensor, um sich mit dem NB-IoT-Netzwerk zu verbinden und Sensordaten an den Server zu senden. Zwischen den einzelnen periodischen Abtastungen/Sende-/Empfangsvorgängen befindet sich der Sensor im IDLE-Modus. Im IDLE-Modus hat der Sensor denselben Stromverbrauch wie im Deep-Sleep-Modus.

## 1.6 Tasten und LEDs



Verhalten bei ACT	Funktion	Aktion
Drücken von ACT zwischen 1 s < Zeit < 3 s der Sensor	Uplink senden	Wenn der Sensor bereits mit dem NB-IoT-Netzwerk verbunden ist, wird ein Uplink-Paket gesendet. Die <b>blaue LED</b> blinkt einmal. Währenddessen ist das BLE-Modul aktiv und der Benutzer kann über BLE eine Verbindung herstellen, um das Gerät zu konfigurieren.
Drücken Sie ACT länger als 3 Sekunden Sekunden in <a href="#">den OTA-Modus</a>	Aktives Gerät	Die <b>grüne LED</b> blinkt fünfmal schnell, das Gerät wechselt für 3 Sekunden. Anschließend beginnt es, sich mit dem NB-IoT-Netzwerk zu verbinden. Die <b>grüne LED</b> leuchtet nach dem Beitritt zum Netzwerk 5 Sekunden lang dauerhaft. Sobald der Sensor aktiv ist, ist das BLE-Modul aktiv und der Benutzer kann sich über BLE verbinden, um das Gerät zu konfigurieren, unabhängig davon, ob das Gerät mit dem NB-IoT-Netzwerk verbunden ist oder nicht.
Drücken Sie schnell 5 Mal auf ACT. Gerät im Tiefen	Gerät deaktivieren	Die <b>rote LED</b> leuchtet 5 Sekunden lang. Das bedeutet, dass sich das Gerät im Tiefen befindet.

**Hinweis:** Wenn das Gerät ein Programm ausführt, sind die Tasten möglicherweise nicht funktionsfähig. Drücken Sie die Tasten am besten, nachdem das Gerät die Programmausführung abgeschlossen hat.

## 1.7 BLE-Verbindung

RS485-NB unterstützt die Fernkonfiguration und Firmware-Aktualisierung über BLE.

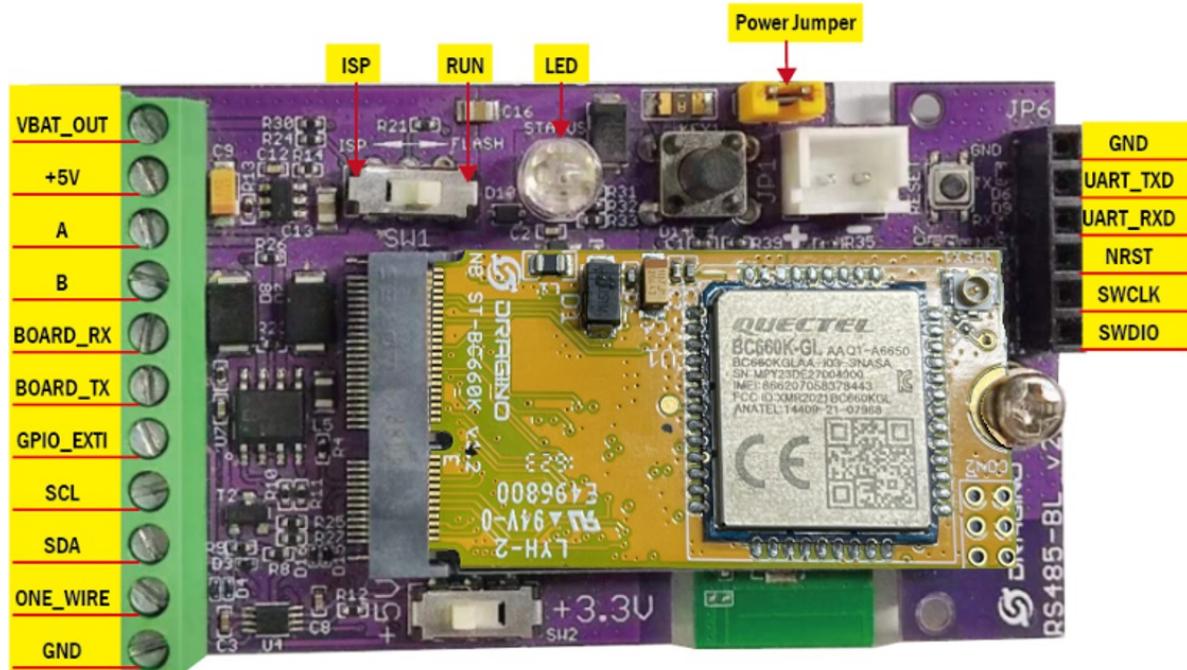
BLE kann verwendet werden, um die Parameter des Sensors zu konfigurieren oder die Konsolenausgabe des Sensors anzuzeigen. BLE wird nur in den folgenden Fällen aktiviert:

- Drücken Sie die Taste, um eine Uplink-Verbindung herzustellen.
- Drücken Sie die Taste, um das Gerät zu aktivieren.
- Gerät einschalten oder zurücksetzen.

Wenn innerhalb von 60 Sekunden keine Aktivitätsverbindung über BLE hergestellt wird, schaltet der Sensor das BLE-Modul aus, um in den Energiesparmodus zu wechseln.

## 1.8 Pin-Definitionen, Schalter- und SIM-Ausrichtung

RS485-NB verwendet die unten abgebildete Hauptplatine.



### 1.8.1 Jumper JP2

Das Gerät wird eingeschaltet, wenn dieser Jumper gesetzt ist.

### 1.8.2 BOOT-MODUS / SW1

1) **ISP**: Upgrade-Modus, das Gerät empfängt in diesem Modus kein Signal, ist jedoch für ein Firmware-Upgrade bereit. Die LED leuchtet nicht. Die Firmware wird nicht ausgeführt.

2) **Flash**: Arbeitsmodus, Gerät beginnt zu arbeiten und sendet Konsolenausgabe für weitere Fehlerbehebung

### 1.8.3 Reset-Taste

Drücken Sie diese Taste, um das Gerät neu zu starten.

#### 1.8.4 Ausrichtung der SIM-Karte

Siehe diesen Link. [So legen Sie die SIM-Karte ein.](#)

#### 1.8.5 SW2-Jumper (UART-Pegel für externen Sensor festlegen)

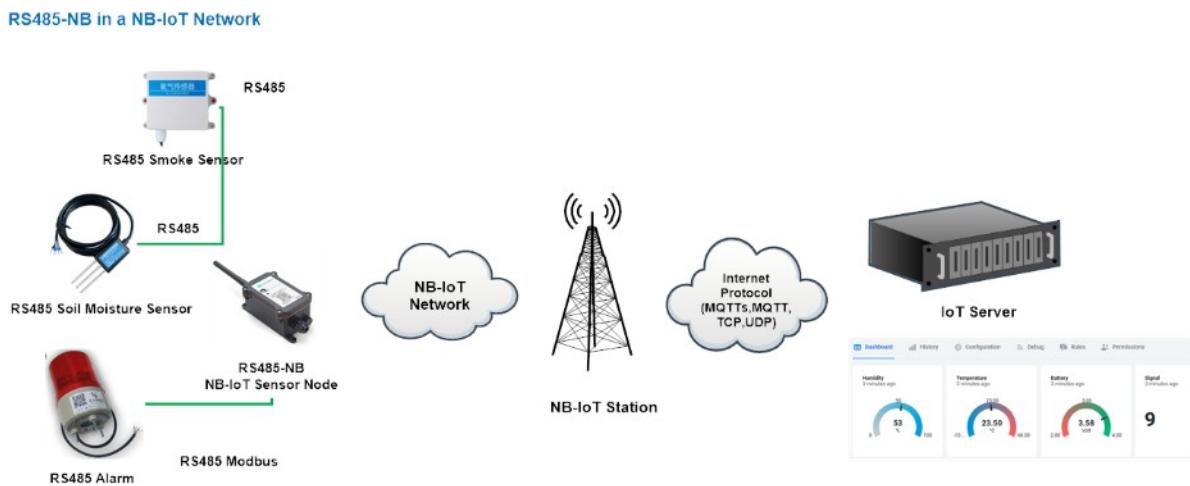
SW2 definiert den Spannungspegel der Pins BOARD\_RX und BOARD\_TX. Er sollte mit dem Spannungspegel des externen Sensors übereinstimmen.

## 2. Verwenden Sie RS485-NB für die Kommunikation mit dem IoT-Server.

### 2.1 Daten über das NB-IoT-Netzwerk an den IoT-Server senden

Der RS485-NB ist mit einem NB-IoT-Modul ausgestattet. Die vorinstallierte Firmware im RS485-NB ruft Umgebungsdaten von Sensoren ab und sendet die Werte über das NB-IoT-Modul an das lokale NB-IoT-Netzwerk. Das NB-IoT-Netzwerk leitet diese Werte über das vom RS485-NB definierte Protokoll an den IoT-Server weiter.

Nachfolgend ist die Netzwerkstruktur dargestellt:



Es gibt zwei Versionen: die **-GE-** und die **-1D-**Version des RS485-NB.

**GE-Version:** Diese Version enthält keine SIM-Karte und verweist nicht auf einen IoT-Server. Der Benutzer muss die folgenden zwei Schritte mit AT-Befehlen konfigurieren, um RS485-NB für die Datenübertragung an den IoT-Server einzurichten.

- Installieren Sie die NB-IoT-SIM-Karte und konfigurieren Sie den APN. Siehe Anleitung unter [„Netzwerk verbinden“](#).
- Richten Sie den Sensor so ein, dass er auf den IoT-Server verweist. Siehe Anleitung zum [Konfigurieren der Verbindung zu verschiedenen Servern](#). Nachfolgend finden Sie eine Übersicht über die Ergebnisse verschiedener Server.

Server	Dashboard	Kommentare

[Node-Red](#)



[DataCake](#)



[Allgemeines UDP](#)

Rohdaten-Nutzlast. Entwickler für die Gestaltung des Dashboards gesucht

[Allgemeines](#)

[MQTT](#)

Rohdaten. Entwickler für die Gestaltung des Dashboards gesucht

**1D-Version:** Diese Version verfügt über eine vorinstallierte 1NCE-SIM-Karte und ist so konfiguriert, dass sie Werte an DataCake sendet. Der Benutzer muss lediglich den Sensortyp in DataCake auswählen und RS485-NB aktivieren, um die Daten in DataCake anzuzeigen. [Die Konfigurationsanweisungen für DataCake](#) finden Sie hier.

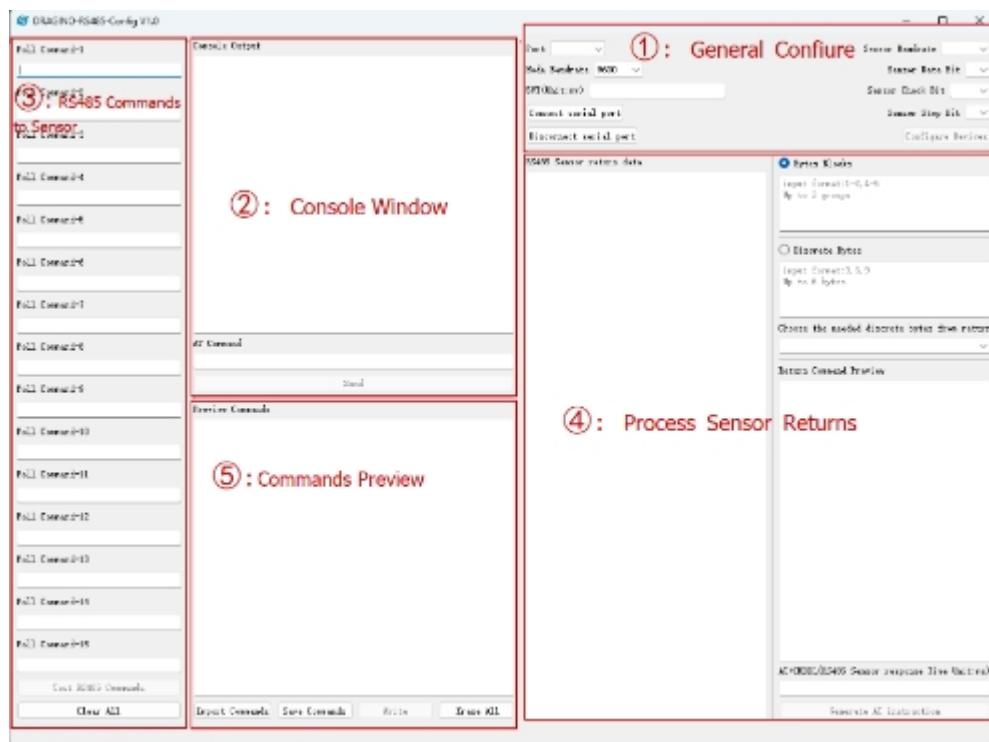
## 2.2 Gerät zum Lesen von RS485-Sensoren konfigurieren

Es gibt zahlreiche RS485- und TTL-Geräte auf dem Markt, und jedes Gerät verfügt über unterschiedliche Befehle zum Auslesen der gültigen Daten. Um diese Geräte möglichst flexibel zu unterstützen, bietet RS485-NB einen flexiblen Befehlssatz. Der Benutzer kann [Dragino RS485 Tool](#), [AT-Befehle](#) oder [LoRaWAN Downlink](#)-Befehle verwenden, um zu konfigurieren, wie RS485-NB den Sensor lesen soll und wie die Rückmeldung von RS485- oder TTL-Sensoren behandelt werden soll.

### 2.2.1 Methode 1 – über das RS485-Konfigurationstool

Die Verwendung des RS485-Konfigurationstools ist die empfohlene Methode. Bitte beachten Sie die Anleitung zur Verwendung des Tools:

- [Anleitung zum RS485-Konfigurationstool](#)



## 2.2.2 Methode 2 – über AT-Befehle

### 2.2.2.1 Konfigurieren Sie die UART-Einstellungen für die RS485- oder TTL-Kommunikation

RS485-NB kann entweder mit RS485-Sensoren oder TTL-Sensoren verbunden werden. Der Benutzer muss angeben, welcher Sensortyp angeschlossen werden soll.

#### 1. RS485-MODBUS-Modus:

**AT+MOD=1** // Unterstützt Sensoren vom Typ RS485-MODBUS. Der Benutzer kann mehrere RS485- und Modbus-Sensoren an die A/B-Pins anschließen.

#### 2. TTL-Modus:

**AT+MOD=2** // Unterstützt TTL-Level-Sensoren. Der Benutzer kann einen TTL-Sensor an die TXD/RXD/GND-Pins anschließen.

Die Standard-UART-Einstellungen für RS485-NB sind **9600, keine Parität, Stoppbit 1, Datenbit 8**. Wenn der Sensor andere Einstellungen hat, kann der Benutzer die RS485-NB-Einstellung entsprechend ändern.

AT-Befehle	Beschreibung	Beispiel
AT+BAUDR	Legt die Baudrate fest. Der Standardwert ist: 9600.	AT+BAUDR=9600 Optionen: (1200, 2400, 4800, 14400, 19200, 115200)
AT+PARITY	Stellen Sie die UART-Parität ein. Der Standardwert ist: keine Parität.	AT+PARITY=0 Option: 0: keine Parität, 1: ungerade Parität, 2: gerade Parität
AT+STOPBIT	Serielles Stoppbit festlegen Standardwert ist: 1 Bit.	AT+STOPBIT=0 für 1 Bit AT+STOPBIT=1 für 1,5 Bits AT+STOPBIT=2 für 2 Bits
AT+DATABIT	Serielles Datenbit einstellen.	AT+DATABIT=7 für 7 Bits

Der Standardwert ist: 8 Bits.

AT+DATABIT=8 für 8 Bits

Beispiel (Dreiparameter-Bodendetektor):

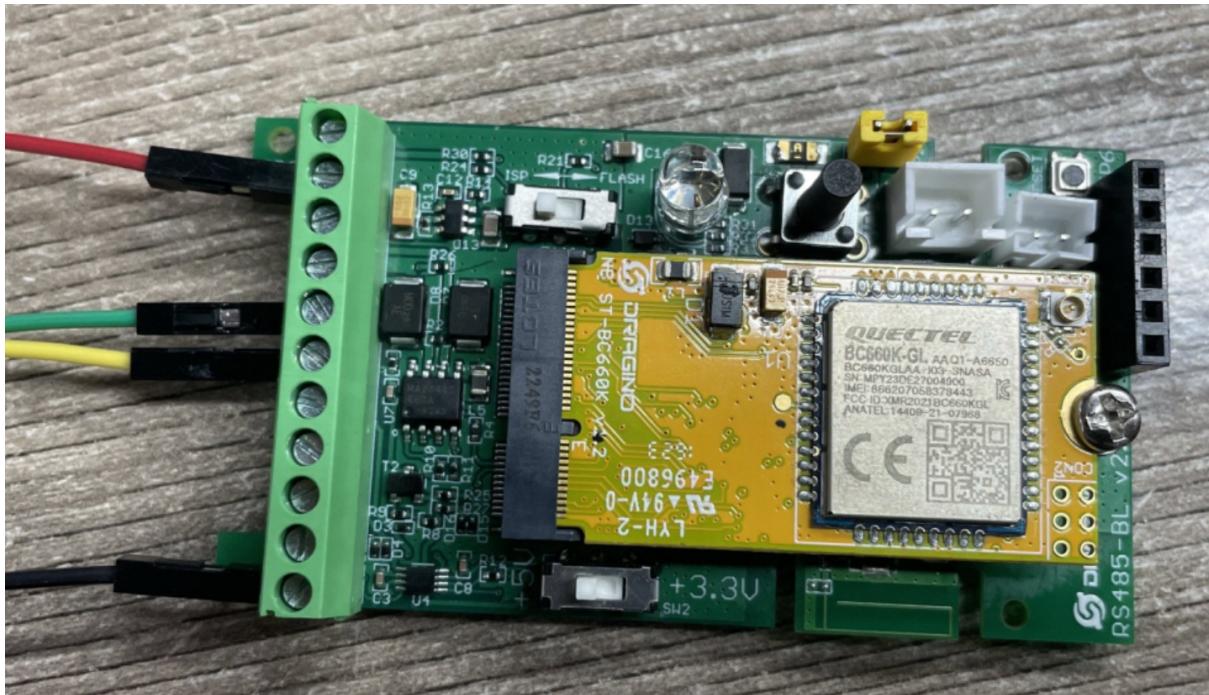
#### Verkabelung des UART-

**Sensors** GND < -----> GND

TX <-----> RX

RX <-----> TX

VCC <-----> 3,3/5 V



**Stellen Sie die richtige Konfiguration ein:**

**AT+BAUDR=9600 AT+PARITY=0**

**AT+STOPBIT=1 AT+DATABIT=8**

Wenn der Sensor 5 V benötigt, muss der Schalter auf 5 V gestellt und dann der Befehl **AT+5VT=30000** verwendet werden.

**Konfigurations-Lese-Befehl: AT+CFGDEV=FE**

**03 00 00 00 03 11 C4,0**

**FE:** Stationsadresse

**03:** Funktionscode

**00 00:** Startadresse des Registers

**00 03:** Anzahl der Register **11 04:**

Prüfcode

AT+CFGDEV=FE 03 00 00 00 03 11 C4, 0

AT+CFGDEV=fe 03 00 00 00 03 11 c4 , 0

RETURN DATA:

fe 03 06 00 00 09 49 00 00 b6 cb

OK

Verwenden Sie AT+COMMAND1, um es als Befehl festzulegen, und verwenden Sie AT+DATACUT1, um die benötigten Bytes abzufangen

AT+PAYVER=1  
AT+MBFUN=0

AT+COMMAND1=fe 03 00 00 00 03 11 c4 , 0	AT+SEARCH1=0,0	AT+DATACUT1=11,2,4~9	AT+CMDDL1=0
AT+COMMAND2=0,0	AT+SEARCH2=0,0	AT+DATACUT2=0,0,0	AT+CMDDL2=0
AT+COMMAND3=0,0	AT+SEARCH3=0,0	AT+DATACUT3=0,0,0	AT+CMDDL3=0
AT+COMMAND4=0,0	AT+SEARCH4=0,0	AT+DATACUT4=0,0,0	AT+CMDDL4=0
AT+COMMAND5=0,0	AT+SEARCH5=0,0	AT+DATACUT5=0,0,0	AT+CMDDL5=0
AT+COMMAND6=0,0	AT+SEARCH6=0,0	AT+DATACUT6=0,0,0	AT+CMDDL6=0
AT+COMMAND7=0,0	AT+SEARCH7=0,0	AT+DATACUT7=0,0,0	AT+CMDDL7=0
AT+COMMAND8=0,0	AT+SEARCH8=0,0	AT+DATACUT8=0,0,0	AT+CMDDL8=0
AT+COMMAND9=0,0	AT+SEARCH9=0,0	AT+DATACUT9=0,0,0	AT+CMDDL9=0
AT+COMMANDA=0,0	AT+SEARCHA=0,0	AT+DATACUTA=0,0,0	AT+CMDDL4A=0
AT+COMMANDB=0,0	AT+SEARCHB=0,0	AT+DATACUTB=0,0,0	AT+CMDDL4B=0
AT+COMMANDC=0,0	AT+SEARCHC=0,0	AT+DATACUTC=0,0,0	AT+CMDDL4C=0
AT+COMMANDD=0,0	AT+SEARCHD=0,0	AT+DATACUTD=0,0,0	AT+CMDDL4D=0
AT+COMMANDE=0,0	AT+SEARCHE=0,0	AT+DATACUTE=0,0,0	AT+CMDDL4E=0
AT+COMMANDF=0,0	AT+SEARCHF=0,0	AT+DATACUTF=0,0,0	AT+CMDDL4F=0

OK

**Nutzlast hochladen:**

[41270]\*\*\*\*\*Upload start:0\*\*\*\*\*

CMD1 = fe 03 00 00 00 03 11 c4  
RETURN1 = fe 03 06 00 00 09 49 00 00 b6 cb  
Payload = 01 00 00 09 49 00 00

[41928]BAT:3.584 V

[45458]Open a Socket Service successfully

[52516]Datagram is sent by RF

[53550]Send complete

### 2.2.3 Sensoren konfigurieren

Einige Sensoren müssen möglicherweise vor dem normalen Betrieb konfiguriert werden. Der Benutzer kann solche Sensoren über den PC oder über RS485-NB AT-Befehle **AT+CFGDEV** konfigurieren.

Wenn der Benutzer einen AT+CFGDEV-Befehl ausgibt, entspricht jeder **AT+CFGDEV** dem Senden eines Befehls an die RS485- oder TTL-Sensoren. Dieser Befehl wird nur ausgeführt, wenn der Benutzer ihn eingibt, und wird nicht bei jeder Abtastung ausgeführt.

AT-Befehle	Beschreibung	Beispiel
AT+CFGDEV	Dieser Befehl wird verwendet, um die RS485/TTL-Geräte zu konfigurieren. AT+CFGDEV=xx xx xx xx xx xx xx xx xx xx xx,m Geräte zu konfigurieren; diese werden während der Abtastung nicht verwendet. AT+CFGDEV=xx xx xx, mm: 0: kein CRC, 1: CRC-16/MODBUS am Ende dieses Befehls hinzufügen	AT+CFGDEV=xx xx xx xx xx xx xx xx xx xx xx xx,m

### 2.2.4 Konfigurieren Sie Lesebefehle für jede Abtastung

**Hinweis: Bitte aktualisieren Sie die Firmware auf Version 1.0.3 oder höher. Andernfalls wird das Gerät in einen nicht funktionsfähigen Zustand versetzt, wenn im AT+COMMANDx-CRC-Prüfmodus ein Datenfehler auftritt.**

RS485-NB ist ein batteriebetriebenes Gerät, das die meiste Zeit im Ruhezustand ist. Es wird in bestimmten Zeitabständen aktiviert, liest die RS485/TTL-Sensordaten und überträgt sie.

Bei jeder Probenahme müssen wir bestätigen, welche Befehle wir an die Sensoren senden müssen, um Daten auszulesen. Nachdem die RS485/TTL-Sensoren den Wert zurückgesendet haben, enthält dieser normalerweise mehrere Bytes, von denen wir nur einige wenige für eine verkürzte Nutzlast benötigen.

Um die Bandbreite des LoRaWAN-Netzwerks zu schonen, müssen wir möglicherweise Daten von verschiedenen Sensoren auslesen und deren gültige Werte zu einer kurzen Nutzlast zusammenfassen.

In diesem Abschnitt wird beschrieben, wie die oben genannten Ziele erreicht werden können.

Bei jeder Abtastung kann der RS485-NB 15 Befehle zum Auslesen von Sensoren unterstützen. Und die Rückmeldung zu einer oder mehreren Uplink-Nutzlasten kombinieren.

#### Befehl vom RS485-NB an den Sensor:

Der RS485-NB kann über **AT+COMMAND1**, **ATCOMMAND2**,..., bis zu maximal 15 voreingestellte Zeichenfolgen an **AT+COMMANDF** senden. Alle Befehle haben die gleiche Grammatik.

### Verarbeitung der Rückmeldung von Sensoren an RS485-NB:

Nachdem der RS485-NB eine Zeichenfolge an den Sensor gesendet hat, wartet der RS485-NB auf die Rückmeldung vom RS485 oder TTL-Sensor. Der Benutzer kann mit den Befehlen **AT+DATACUT** oder **AT+SEARCH** festlegen, wie die Rückmeldung verarbeitet werden soll.

- **AT+DATACUT**

Wenn der Rückgabewert vom Sensor eine ftx-Länge hat und wir wissen, an welcher Position wir den gültigen Wert erhalten sollten, können wir den Befehl AT+DATACUT verwenden.

- **AT+SEARCH**

Wenn der Rückgabewert vom Sensor eine dynamische Länge hat und wir nicht sicher sind, welche Bytes die gültigen Daten sind, wissen wir stattdessen, welcher Wert der gültige Wert ist. Wir können AT+SEARCH verwenden, um den gültigen Wert in der Rückgabekette zu suchen.

#### Wartezeit festlegen:

Einige RS485-Geräte können eine längere Verzögerung bei der Antwort haben, daher kann der Benutzer AT+CMDDL verwenden, um die Zeitüberschreitung für den Erhalt einer Antwort nach dem Senden des RS485-Befehls festzulegen. Beispiel: AT+CMDDL1=1000, um die Öffnungszeit auf 1000 ms zu setzen

#### Beispiele:

Nachfolgend finden Sie Beispiele für die Funktionsweise der oben genannten AT-Befehle.

**AT+COMMANDx:** Dieser Befehl wird bei jeder Abtastung an RS485/TTL-Geräte gesendet. Die maximale Befehslänge beträgt 14 Byte. Die Grammatik lautet:

```
AT+COMMANDx=xx xx xx, m  
xx xx: Der zu sendende RS485-Befehl m: 0: kein  
CRC, 1: CRC-16/MODBUS am Ende dieses Befehls hinzufügen
```

Nehmen wir beispielsweise einen RS485-Sensor. Der Befehl zum Abrufen des Sensorwerts lautet: 01 03 0B B8 00 02 46 0A. Dabei ist 01 03 0B B8 00 02 der Modbus-Befehl zum Auslesen des Registers 0B B8, in dem der Sensorwert gespeichert ist. 46 0A ist der CRC-16/MODBUS, der manuell berechnet wird.

Im RS485-NB sollten wir dafür den Befehl AT+COMMAND1=01 03 0B B8 00 02,1 verwenden.

Wenn ein einzelner Befehl 14 Byte überschreitet, können Sie die Befehlsverknüpfungsfunktion verwenden.

Wenn AT+CMDDLx=1, werden die Befehle von AT+COMMANDx und AT+COMMAND(x+1) zusammengefügt.

**Beispiele:** Um die Daten 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F zu senden, sollte Folgendes konfiguriert werden:

```
AT+COMMAND1=00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D,0  
AT+COMMAND1=1  
AT+COMMAND2=0E 0F,0
```

**AT+SEARCHx:** Dieser Befehl legt fest, wie die Rückgabe von AT+COMMANDx behandelt werden soll.

```
AT+SEARCHx=aa,xx xx xx xx xx  
• aa: 1: Präfix-Übereinstimmungsmodus; 2: Präfix- und Suffix-Übereinstimmungsmodus  
• xx xx xx xx xx: Übereinstimmungszeichenfolge. Maximal 5 Byte für Präfix und 5 Byte für Suffix
```

#### Beispiele:

1) Für eine Rückgabezeichenfolge von AT+COMMAND1: 16 0c 1e 56 34 2e 30 58 5f 36 41 30 31 00 49

Wenn wir AT+SEARCH1=1,1E 56 34 einstellen.(max. 5 Bytes für preftx)

Die gültigen Daten sind alle Bytes nach 1E 56 34, also **2e 30 58 5f 36 41 30 31 00 49**

```
CMD1    = 11 01 1e d0
SEARCH1  = 1e 56 34
RETURN1  = 2e 30 58 5f 36 41 30 31 00 49
Payload   = 8d 2d 01 2e 30 58 5f 36 41 30 31 00 49
```

## 2) Für eine Rückgabezeichenfolge von AT+COMMAND1: 16 0c 1e 56 34 2e 30 58 5f 36 41 30 31 00 49

Wenn wir AT+SEARCH1=2 setzen, 1E 56 34+31 00 49

Das Gerät sucht nach den Bytes zwischen 1E 56 34 und 31 00 49. Es handelt sich also um **2e 30 58 5f 36 41 30**.

```
CMD1    = 11 01 1e d0
SEARCH1  = 1e 56 34
RETURN1  = 2e 30 58 5f 36 41 30 31 00 49
Payload   = 8d 2d 01 2e 30 58 5f 36 41 30 31 00 49
```

**AT+DATACUTx:** Dieser Befehl legt fest, wie die Rückgabe von AT+COMMANDx behandelt wird. Die maximale Rückgabellänge beträgt 100 Byte.

**AT+DATACUTx=a,b,c**

- a: Länge für die Rückgabe von AT+COMMAND
- b:1: Gültigen Wert byteweise erfassen, maximal 6 Bytes. 2: Gültigen Wert in Byte-Abschnitten erfassen, maximal 3 Abschnitte.
- c: Position für gültigen Wert definieren.

Beispiele:

- Bytes erfassen:

```
AT+PAYVER=1
AT+COMMAND1=01 03 0b b8 00 02 ,1      AT+DATACUT1=10,1,9+4+6+8+1+3
AT+COMMAND2=0,0      AT+DATACUT2=0,0,0
AT+COMMAND3=0,0      AT+DATACUT3=0,0,0
AT+COMMAND4=0,0      AT+DATACUT4=0,0,0
AT+COMMAND5=0,0      AT+DATACUT5=0,0,0
' AT+DATACUT1=10,1,9+4+6+8+1+3
' a=10, return total 10 bytes (20 20 20 20 2d 30 2e 32 20 75)
' b=1 grab byte.
' c=9+4+6+8+1+3 (grap the 9th , 4th, 6th, 8th, 1th, 3rd byte and link them together by grab sequence
' so command1 valid value is 20 20 30 32 20 20
AT+COMMAND6=0,0      AT+DATACUT6=0,0,0
AT+COMMAND7=0,0      AT+DATACUT7=0,0,0
AT+COMMAND8=0,0      AT+DATACUT8=0,0,0
AT+CHS=0
```

OK

CMD1	= 01 03 0b b8 00 02 46 0a
RETURN1	= 20 20 20 20 2d 30 2e 32 20 75
Payload	= 0c fc 01 20 20 30 32 20 20

- Einen Abschnitt erfassen.

```
AT+PAYVER=1
AT+COMMAND1=01 03 0b b8 00 02 ,1      AT+DATAUT1=8,2,4~8
AT+COMMAND2=0,0      AT+DATAUT2=0,0,0
AT+COMMAND3=0,0      AT+DATAUT3=0,0,0
AT+COMMAND4=0,0      AT+DATAUT4=0,0,0
AT+COMMAND5=0,0      AT+DATAUT5=0,0,0
AT+COMMAND6=0,0      AT+DATAUT6=0,0,0
AT+COMMAND7=0,0      AT+DATAUT7=0,0,0
AT+COMMAND8=0,0      AT+DATAUT8=0,0,0
AT+CC a=8, return total 8 bytes (20 20 20 20 2d 30 2e 00)
AT+CC b=2
AT+CC c=4~8 (grap the 4th ~ 8th bytes from return, so command1 valid value is 20 2d 30 2e 00)
AT+COMMAND1=0,0      AT+DATAUT1=0,0,0
AT+CHS=0
```

OK

CMD1	= 01 03 0b b8 00 02 46 0a
RETURN1	= 20 20 20 20 2d 30 2e 00
Payload	= 0c fc 01 20 2d 30 2e 00

- Verschiedene Abschnitte erfassen.

```
AT+COMMAND1=01 03 0b b8 00 02 ,1      AT+DATAUT1=13,2,1~2+4~7+10~11
AT+COMMAND2=0,0      AT+DATAUT2=0,0,0
AT+COMMAND3=0,0      AT+DATAUT3=0,0,0
AT+COMMAND4=0,0      AT+DATAUT4=0,0,0
AT+COMMAND5=0,0      AT+DATAUT5=0,0,0
AT+COMMAND6=0,0      AT+DATAUT6=0,0,0
AT+DATAUT1=13,2,1~2+4~7+10~11
a=13, return total 13 bytes (90 02 6a 82 1a 04 20 2d 30 2e dd 9b 00)
b=2
c=1~2+4~7+10~11 (grap the 1 ~ 2 bytes + 4~7 bytes + 10~11 bytes
so command1 valid value is 90 02 82 1a 04 20 2e dd
AT+COMMAND1=0,0      AT+DATAUT1=0,0,0
AT+COMMANDF=0,0      AT+DATAUTF=0,0,0
AT+CHS=0
```

OK

CMD1	= 01 03 0b b8 00 02 46 0a
RETURN1	= 90 02 6a 82 1a 04 20 2d 30 2e dd 9b 00
Payload	= 0c fc 01 90 02 82 1a 04 20 2e dd

#### Hinweis:

AT+SEARCHx und AT+DATAUTx können zusammen verwendet werden. Wenn beide Befehle eingestellt sind, verarbeitet RS485-NB zuerst AT+SEARCHx in der Rückgabzeichenfolge und erhält eine temporäre Zeichenfolge. Anschließend verarbeitet es AT+DATAUTx in dieser temporären Zeichenfolge, um die endgültige Nutzlast zu erhalten. In diesem Fall muss AT+DATAUTx auf das Format AT+DATAUTx=0,xx,xx gesetzt werden, wobei die Rückgabebits auf 0 gesetzt werden.

#### Beispiel:

AT+COMMAND1=11 01 1E D0,0

AT+SEARCH1=1,1E 56 34 AT+DATAUT1=0,2,1~5

Rückgabestring von AT+COMMAND1: 16 0c 1e 56 34 2e 30 58 5f 36 41 30 31 00 49

Zeichenfolge nach dem Befehl SEARCH: 2e 30 58 5f 36 41 30 31 00 49

Gültige Nutzlast nach dem Befehl DataCUT: 2e 30 58 5f 36

```
CMD1 = 11 01 1e d0
SEARCH1 = 1e 56 34
RETURN1 = 2e 30 58 5f 36 41 30 31 00 49
Payload = 8d 2d 01 2e 30 58 5f 36
```

## 2.2.5 Uplink, wenn der Sensor nicht reagiert

- 1) Wenn **AT+MOD=1** ist und die von **AT+DATAUT** oder **AT+MBFUN** abgefangenen Daten leer sind, wird **NULL** angezeigt und die Nutzlast wird mit **n FFs** aufgefüllt.

```
[146157]Signal Strength:23
[149091]*****Upload start:4*****
CMD1 = fe 03 00 00 00 03 11 c4
RETURN1 = NULL
Payload = 01 ff ff ff ff ff ff

[149645]BAT:3.594 V
[156683]Datagram is sent by RF
[157718]Send complete
[158743]*****End of upload*****
```

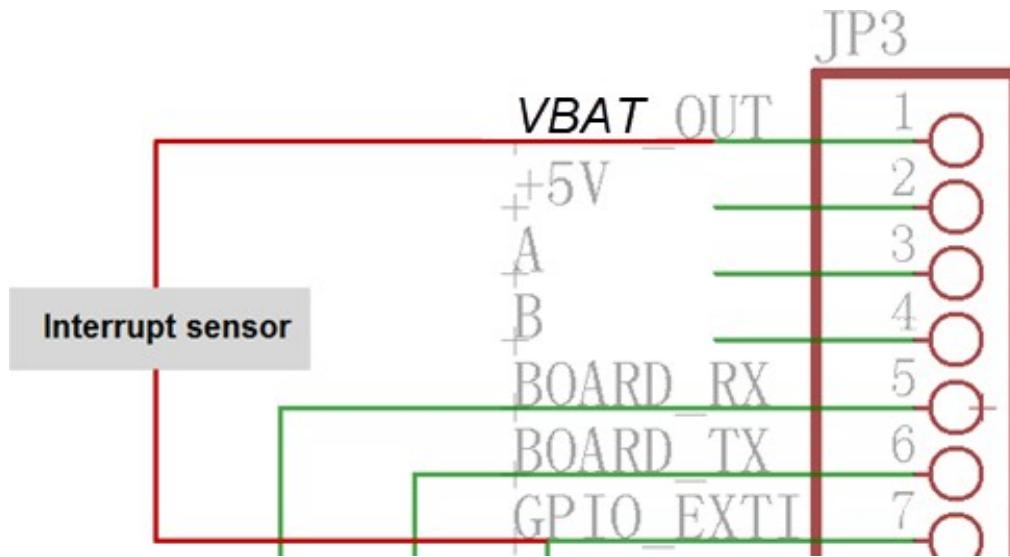
- 2) Wenn **AT+MOD=2**, und die von **AT+DATAUT** oder **AT+MBFUN** abgefangenen Daten leer sind, wird **NULL** angezeigt, und die Nutzlast wird mit **n 00s** gefüllt.

```
[172119]Signal Strength:23
[175053]*****Upload start:5*****
CMD1 = fe 03 00 00 00 03 11 c4
RETURN1 = NULL
Payload = 01 00 00 00 00 00 00

[176587]BAT:3.591 V
[183623]Datagram is sent by RF
[184658]Send complete
[185683]*****End of upload*****
```

## 2.2.6 Uplink bei Unterbrechung

Setzen Sie den Interrupt-Sensor zwischen 3,3 V\_out und GPIO ext.



**AT+INTMOD=0** Interrupt deaktivieren (Standardwert) **AT+INTMOD=1**

Interrupt-Auslösung durch steigende oder fallende Flanke. **AT+INTMOD=2**

Interrupt-Auslösung durch fallende Flanke.

**AT+INTMOD=3** Interrupt-Auslösung durch steigende Flanke.

## 2.3 Nutzlasttypen

Um unterschiedlichen Serveranforderungen gerecht zu werden, unterstützt RS485-NB verschiedene Nutzlasttypen.

Dazu gehören:

- [Allgemeine Nutzlast im JSON-Format](#). (Typ=5)
- [Nutzlast im HEX-Format](#). (Typ=0)

Der Benutzer kann den Nutzlasttyp bei der Auswahl des Verbindungsprotokolls angeben. Beispiel:

**AT+PRO=2,0** // UDP-Verbindung und Hex-Nutzlast verwenden

**AT+PRO=2,5** // UDP-Verbindung und JSON-Nutzlast

verwenden **AT+PRO=3,0** // MQTT-Verbindung und Hex-Nutzlast

verwenden **AT+PRO=3,5** // MQTT-Verbindung und JSON-Nutzlast

verwenden **AT+PRO=4,0** // TCP-Verbindung und Hex-Nutzlast verwenden

**AT+PRO=4,5** // TCP-Verbindung und JSON-Nutzlast

verwenden

### 2.3.1 Allgemeines JSON-Format (Typ=5)

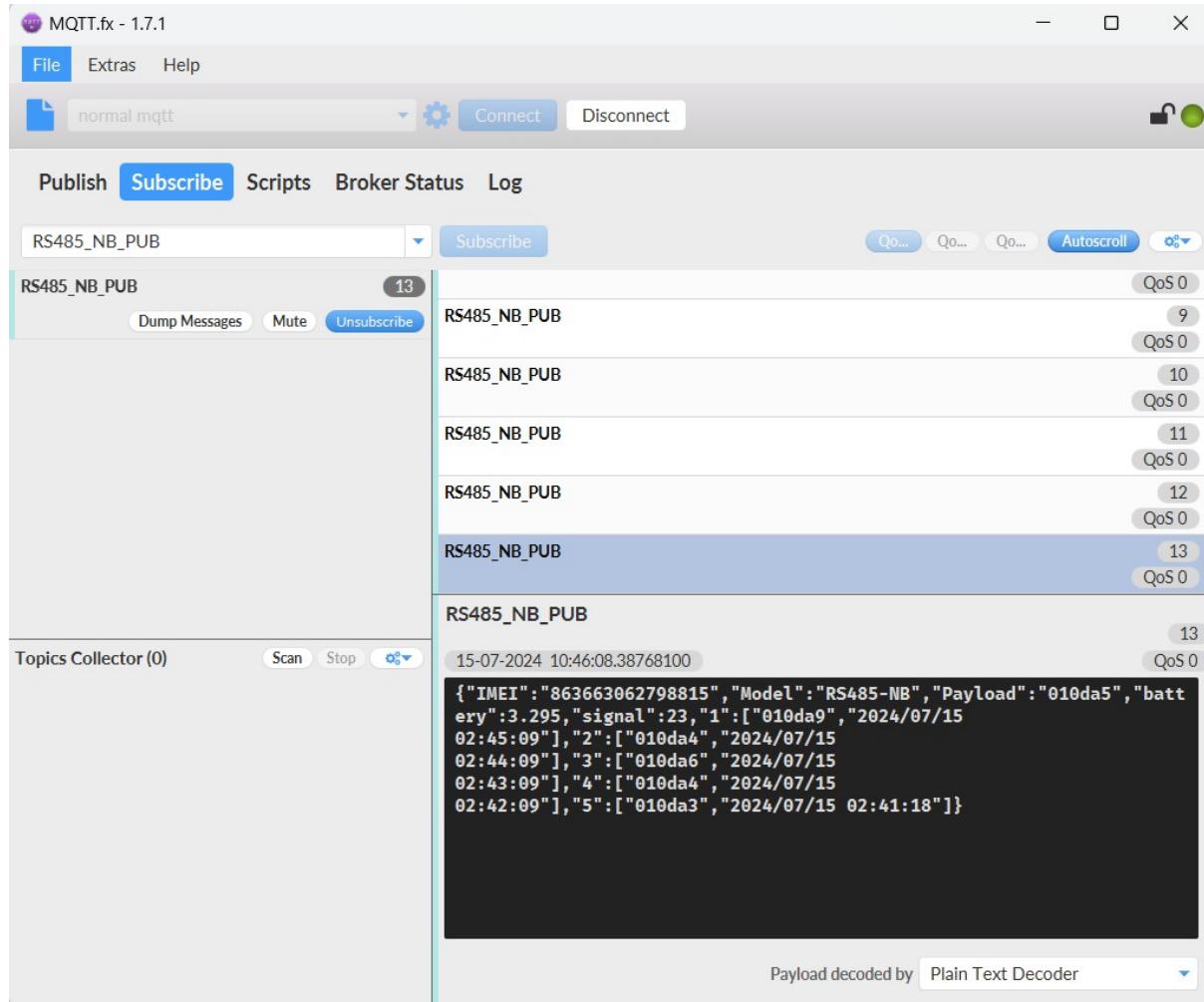
Dies ist das allgemeine JSON-Format. Wie unten angegeben:

```
{"IMEI":"866207053462705","Model":"RS485-  
NB","Payload":01111122223333,"battery":3.581,"signal":23}
```

**Hinweis: Die maximale Anzahl von Bytes in der Nutzlast beträgt 1 Nutzlastversionsnummer plus 687 RS485-Daten.**

AT+CLOCKLOG=1,65535,1,5:

```
{"IMEI":"863663062798815","Modell":"RS485-NB","Nutzlast":"010da5","Batterie":3.295,"Signal":23,"1":["010da9","2024/07/15 02:45:09"],"2":["010da4","2024/07/15 02:44:09"],"3":["010da6","2024/07/15 02:43:09"],"4":["010da4","2024/07/15 02:42:09"],"5":["010da3","2024/07/15 02:41:18"]}
```



### 2.3.2 HEX-Format Nutzlast (Typ=0)

Dies ist das HEX-Format. Wie unten gezeigt:

**f863663062798815 176e 0cdd 1a 00 00 66948e7b 01 0da4**

HEX Format for RS485-NB		176e	0cdd	1a	00	00	66948e7b	01	0da4
f863663062798815		Version	BAT	Signal	GPIO_EXIT Level	GPIO_EXIT Flag	Timestamp	Payload_version	RS485_sensor_data
f+IMEI	8 Bytes						12+n Bytes(The maximum n value is 710)		

#### Version:

Diese Bytes enthalten die Hardware- und Softwareversion.

**Höheres Byte:** Sensor-Modell angeben: 0x17 für RS485-NB

**Unteres Byte:** Geben Sie die Softwareversion an: 0x6e=110 bedeutet Firmware-Version 1.1.0

#### BAT (Batterieinfo):

Beispiel: 0x0cd = 3293 mV

#### **Signalstärke:**

NB-IoT-Netzwerksignalstärke.

#### **Beispiel 1: 0x1a = 26**

- 0** -113 dBm oder weniger
- 1** -111 dBm
- 2...30** -109 dBm... -53 dBm
- 31** -51 dBm oder mehr
- 99** Unbekannt oder nicht nachweisbar

#### **GPIO\_EXIT-Pegel:**

GPIO\_EXTI wird als Interrupt-Pin verwendet.

#### **Beispiel:**

01 (H): GPIO\_EXTI-Pin ist auf hohem Pegel.

00 (L): GPIO\_EXTI-Pin ist auf niedrigem

Pegel.

#### **GPIO\_EXIT-Flag:**

Dieses Datenfeld zeigt an, ob dieses Paket vom **Interrupt-Pin** generiert wurde oder nicht. Hinweis: Der Interrupt-Pin ist ein separater Pin in der Schraubklemme.

#### **Beispiel:**

0x00: Normales Uplink-Paket.

0x01: Interrupt-Uplink-Paket.

#### **Zeitstempel:**

Einheit Zeitstempel Beispiel: 66948E7B(H) = 1721011835(D)

Geben Sie den Dezimalwert in diesen Link (<https://www.epochconverter.com/>) ein, um die Zeit zu erhalten.

#### **Payload\_version:**

Die Payload-Versionsnummer wird zum Parsen verschiedener Dekodierungen verwendet.

#### **RS485\_sensor\_data:**

Die vom RS485-Sensor oder UART-Sensor zurückgegebenen Daten müssen mit den Befehlen AT+DATAUTx und AT+MBFUN oder AT+SEARCHx herausgeschnitten werden.

## **2.4 Uplink testen und Aktualisierungsintervall ändern**

Standardmäßig sendet der Sensor **alle 2 Stunden** Uplinks

Der Benutzer kann die folgenden Befehle verwenden, um das **Uplink-Intervall** zu ändern. **AT+TDC=7200** // Aktualisierungsintervall auf 7200 s einstellen

Der Benutzer kann auch die Taste länger als 1 Sekunde gedrückt halten, um einen Uplink zu aktivieren.

## 2.5 Schnellbefehl zur Verwaltung von MODBUS-Geräten

Der Befehl dient zum schnellen Konfigurieren des Lesens von Modbus-Geräten. Er ist nur für Geräte gültig, die dem [MODBUS-RTU-Protokoll](#) folgen.

### AT+MBFUN hat nur zwei Werte:

- **AT+MBFUN=1**: Modbus-Lesen aktivieren. Die Antwort erfolgt auf Basis der MODBUS-Rückgabe.

AT+MBFUN=1: Das Gerät kann automatisch die Modbus-Funktionscodes 01, 02, 03 oder 04 lesen. AT+MBFUN hat eine niedrigere Priorität als der Befehl AT+DATAUT. Wenn der Befehl AT+DATAUT konfiguriert ist, wird AT+MBFUN ignoriert.

- **AT+MBFUN=0**: Deaktiviert das schnelle Lesen von Modbus.

### Beispiel:

- AT+MBFUN=1 und AT+DATAUT1/AT+DATAUT2 sind nicht konfiguriert (0,0,0).
- AT+COMMAND1= 01 03 00 10 00 08,1 --> Slave-Adresse 01 lesen, Funktionscode 03, Startadresse 00 01, Anzahl der Register 00 08.
- AT+COMMAND2= 01 02 00 40 00 10,1 --> Slave-Adresse 01 lesen, Funktionscode 02, Startadresse 00 40, Anzahl der Eingänge 00 10.

## 2.6 RS485-Befehlszeitüberschreitung

Einige Modbus-Geräte reagieren langsam beim Senden von Antworten. Dieser Befehl wird verwendet, um den RS485-NB so zu konfigurieren, dass er länger auf ihre Reaktion wartet.

Standardwert: 0, Bereich: 0 bis 10 Sekunden

- **AT-Befehl:**

**AT+CMDDLaa=hex(bb cc)**

### Beispiel:

**AT+CMDDL1=1000**, um die Öffnungszeit auf 1000 ms zu setzen

## 2.7 RS485-Befehl löschen

Die Einstellungen AT+COMMANDx und AT+DATAUTx, AT+SEARCHx und AT+CMDDLx werden an einem speziellen Speicherort gespeichert. Der Benutzer kann sie mit dem folgenden Befehl löschen.

- **AT-Befehl:**

**AT+CMDEAR=mm,nn** mm: Startposition des Löschtorgangs, nn: Endposition des Löschtorgangs usw. AT+CMDEAR=1,10 bedeutet Löschen von AT+COMMAND1/AT+DATAUT1/AT+SEARCH1/AT+CMDDL1 bis AT+COMMANDA/AT+DATAUTA/AT+SEARCHA/AT+CMDDLA.

## 2.8 Nutzlastversion festlegen

Dies ist das erste Byte der Uplink-Nutzlast. RS485-NB kann an verschiedene Sensoren angeschlossen werden. Der Benutzer kann das Feld PAYVER einstellen, um dem Server mitzuteilen, wie die aktuelle Nutzlast decodiert werden soll.

- **AT-Befehl:**

**AT+PAYVER:** PAYVER-Feld = 1 setzen

## 2.9 Steuerung der Ausgangsleistungsdauer

Der Benutzer kann die Dauer der Ausgangsleistung vor jeder Abtastung einstellen.

- **AT-Befehl:**

**Beispiel:**

**AT+3V3T=1000** // Die 3V3-Ausgangsleistung wird 1 Sekunde vor jeder Abtastung eingeschaltet.

**AT+3V3T=0** // Normalerweise offene 3V3-Stromversorgung.

**AT+3V3T=65535** // Normalerweise geschlossene 3V3-Stromversorgung.

**AT+5VT=1000** // Die +5-V-Ausgangsleistung wird 1 Sekunde vor jeder Abtastung geöffnet.

**AT+5VT=0** // Normalerweise geschlossene +5-V-Stromversorgung.

**AT+5VT=65535** // Normalerweise offene +5V-Stromversorgung.

## 2.10 Sensorwert abrufen

- **AT-Befehl:**

**AT+GETSENSORVALUE=0** // Die serielle Schnittstelle ruft den aktuellen Sensorwert ab

**AT+GETSENSORVALUE=1** // Der serielle Port ruft den aktuellen Sensorwert ab und lädt ihn hoch.

## 2.11 +3V3-Ausgang

RS485-NB verfügt über einen steuerbaren +3V3-Ausgang, über den der Benutzer externe Sensoren mit Strom versorgen kann.

Der +3V3-Ausgang ist bei jeder Abtastung gültig. RS485-NB aktiviert den +3V3-Ausgang vor jeder Abtastung und deaktiviert ihn nach jeder Abtastung.

Die +3V3-Ausgangszeit kann über einen AT-Befehl gesteuert werden.

### AT+3V3T=1000

Bedeutet, dass die gültige Zeit für +3V3 auf 1000 ms eingestellt wird. Der tatsächliche +3V3-Ausgang hat also 1000 ms + Abtastzeit für andere Sensoren.

Standardmäßig ist AT+3V3T=0. Dies ist ein Sonderfall, der bedeutet, dass der +3V3-Ausgang jederzeit aktiv ist.

## 2.12 +5V-Ausgang

RS485-NB verfügt über einen steuerbaren +5V-Ausgang, über den der Benutzer externe Sensoren mit Strom versorgen kann.

Der +5-V-Ausgang ist bei jeder Abtastung gültig. RS485-NB aktiviert den +5-V-Ausgang vor jeder Abtastung und deaktiviert ihn nach jeder Abtastung.

Die Dauer des 5-V-Ausgangs kann über einen AT-Befehl gesteuert werden.

### AT+5VT=1000

Bedeutet, dass die Gültigkeitsdauer des 5-V-Ausgangs auf 1000 ms eingestellt wird. Der tatsächliche 5-V-Ausgang hat also eine Dauer von 1000 ms + Abtastzeit für andere Sensoren.

Standardmäßig ist AT+5VT=0. Wenn der externe Sensor 5 V benötigt und mehr Zeit benötigt, um einen stabilen Zustand zu erreichen, kann der Benutzer diesen Befehl verwenden, um die Einschaltzeit für diesen Sensor zu verlängern.

## 2.13 Jumper-Schalter

Jumper-Schalter	Funktion
SW1	ISP-Position: Firmware-Upgrade über UART Flash-Position: Gerät konfigurieren, Betriebsstatus überprüfen.
SW2	5V-Position: Auf Kompatibilität mit 5V-I/O einstellen. 3,3-V-Position: Auf Kompatibilität mit 3,3-V-E/A einstellen.

**+3,3 V:** immer eingeschaltet

**+5V:** Nur vor jeder Abtastung öffnen. Die Zeit ist standardmäßig **AT+5VT=0**.

**Hinweis:** Wenn SW2 auf +5V steht und AT+5VT=0 ist, kann der Arbeitsmodus 2 keine Daten senden.

## 2.14 Taktprotokollierung (seit Firmware-Version v1.0.7)

Manchmal, wenn wir viele Endknoten im Feld einsetzen, möchten wir, dass alle Sensoren gleichzeitig Daten erfassen und diese Daten zur Analyse gemeinsam hochladen. In einem solchen Fall können wir die Uhrzeitprotokollierungsfunktion verwenden.

Mit diesem Befehl können wir die Startzeit der Datenaufzeichnung und das Zeitintervall so einstellen, dass sie den Anforderungen der spezifischen Datenerfassungszeit entsprechen.

- **AT-Befehl:** **AT+CLOCKLOG=a,b,c,d**

**a:** 0: Uhrzeitprotokollierung deaktivieren. 1: Uhrzeitprotokollierung aktivieren

**b:** Geben Sie den ersten Abtaststart an. Sekunde: Bereich (0 bis 3599, 65535) // **Hinweis:** Wenn Parameter b auf 65535 gesetzt ist, beginnt die Protokollierungsperiode, nachdem der Knoten auf das Netzwerk zugegriffen und Pakete gesendet hat.

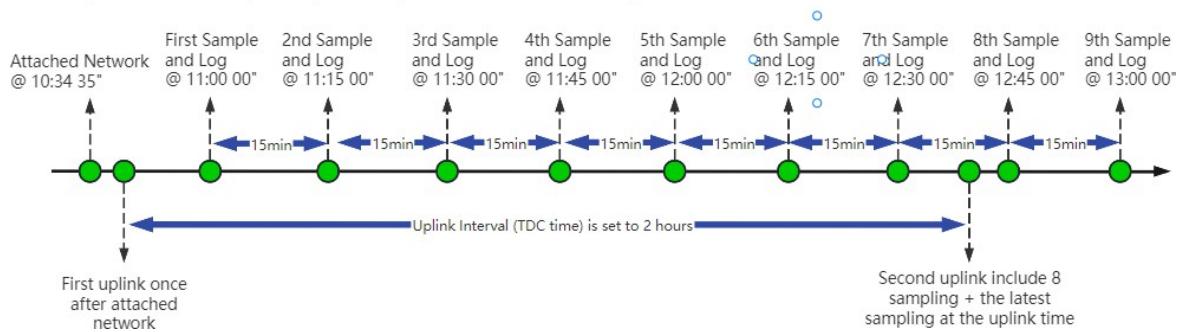
**c:** Sampling-Intervall angeben: Bereich (0 bis 255 Minuten)

**d:** Wie viele Einträge sollen bei jedem TDC hochgeladen werden (**max. 32**)

**Hinweis:** Um die Uhrzeitaufzeichnung zu deaktivieren, setzen Sie die folgenden Parameter: **AT+CLOCKLOG=1,65535,0,0**

Example: AT+CLOCKLOG=1,0,15,8

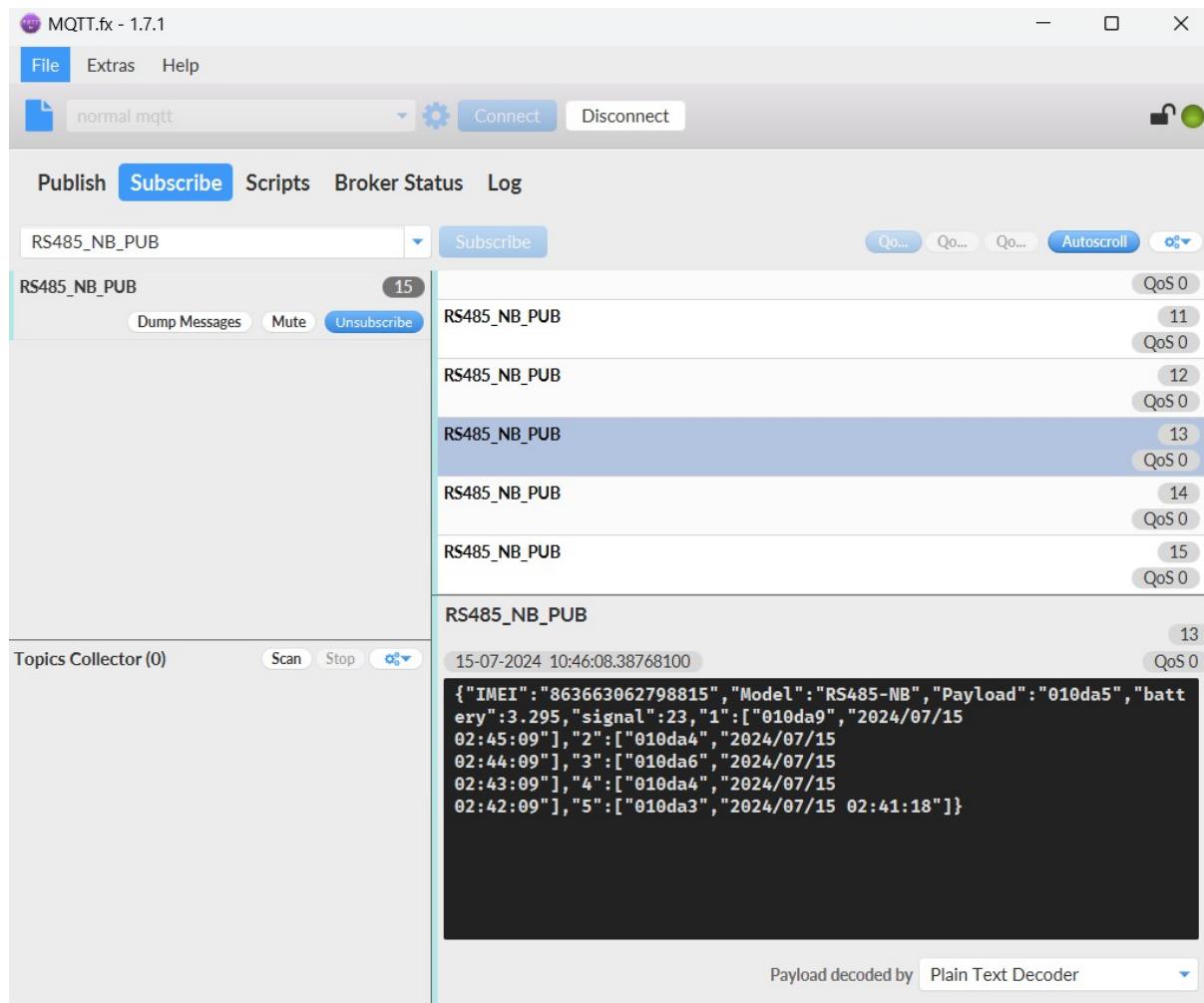
Device will log data to memory start from the 0" second (11:00 00" of first hour and then sampling and log every 15 minutes.Every TDC uplink, the uplink payload will consist: Battery info + last 8 memory record with timestamp + the latest sample at uplink time) . See below for the example.



**Beispiel:**

**AT+CLOCKLOG=1,65535,1,5**

Nachdem der Knoten das erste Paket gesendet hat, werden die Daten in Intervallen von einer Minute im Speicher aufgezeichnet. Für jeden TDC-Uplink umfasst die Uplink-Last: Batterieinformationen + die letzten 5 Speicheraufzeichnungen (Nutzlast + Zeitstempel).



**Hinweis:** Benutzer müssen die Serverzeit synchronisieren, bevor sie diesen Befehl konfigurieren. Wenn die Serverzeit vor der Konfiguration dieses Befehls nicht synchronisiert wird, wird der Befehl erst nach einem Neustart des Knotens wirksam.

- **Downlink-Befehl: 0x0A**

Format: Befehlscode (0x0A) gefolgt von 5 Bytes.

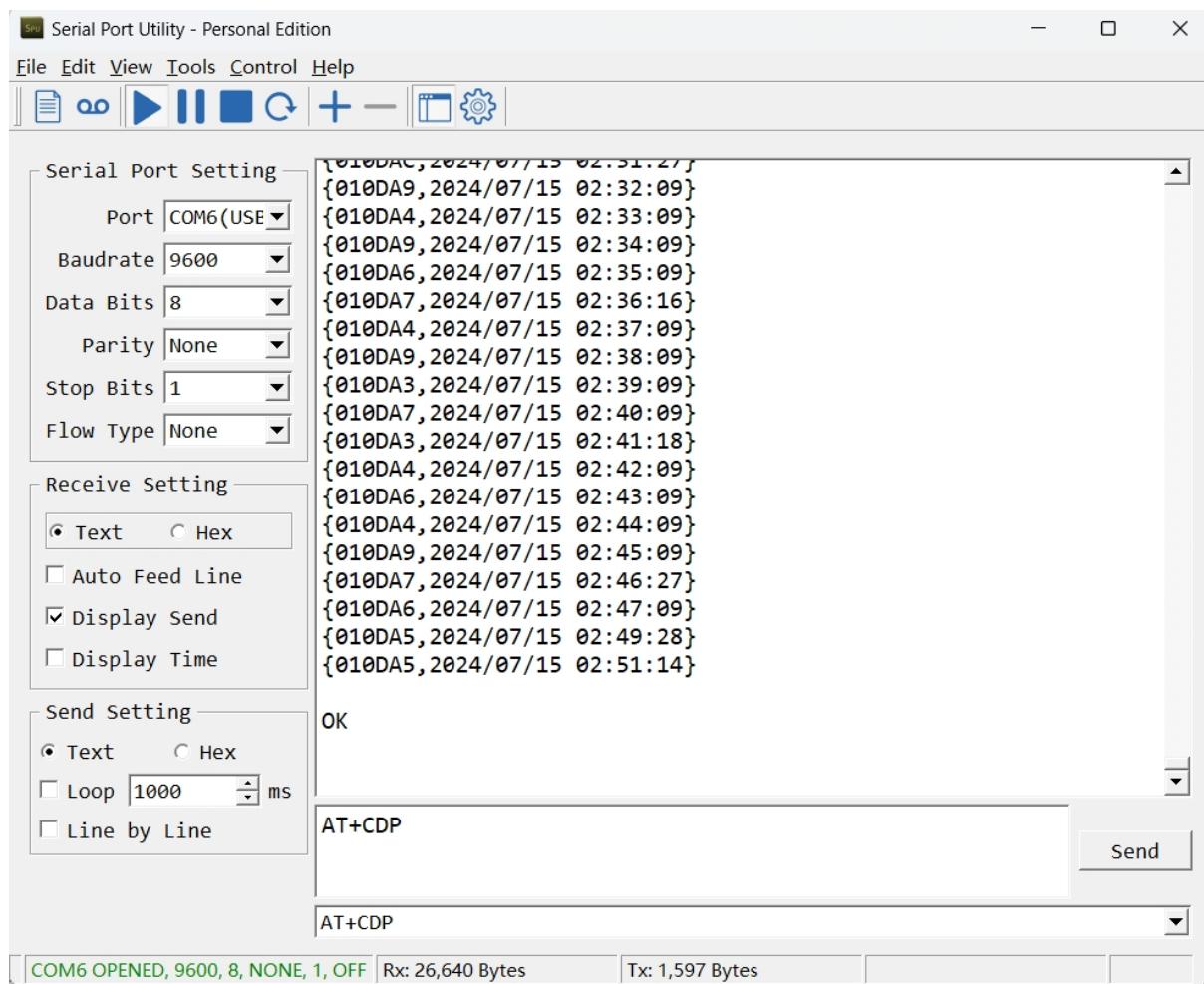
- **Beispiel 1:** Downlink-Nutzlast: **0A01FFFF0F08** // SHT-Aufzeichnungszeit einstellen: AT +CLOCKLOG=1,65535,15,8
- **Beispiel 1:** Downlink-Nutzlast: **0A0104B00F08** // SHT-Aufzeichnungszeit einstellen: AT +CLOCKLOG=1,1200,15,8

**Hinweis:** Bei der Eingabe der Downlink-Nutzlast dürfen keine Leerzeichen zwischen den Bytes stehen.

## 2.15 Beispielabfrage gespeicherter historischer Datensätze

- **AT-Befehl: AT+CDP**

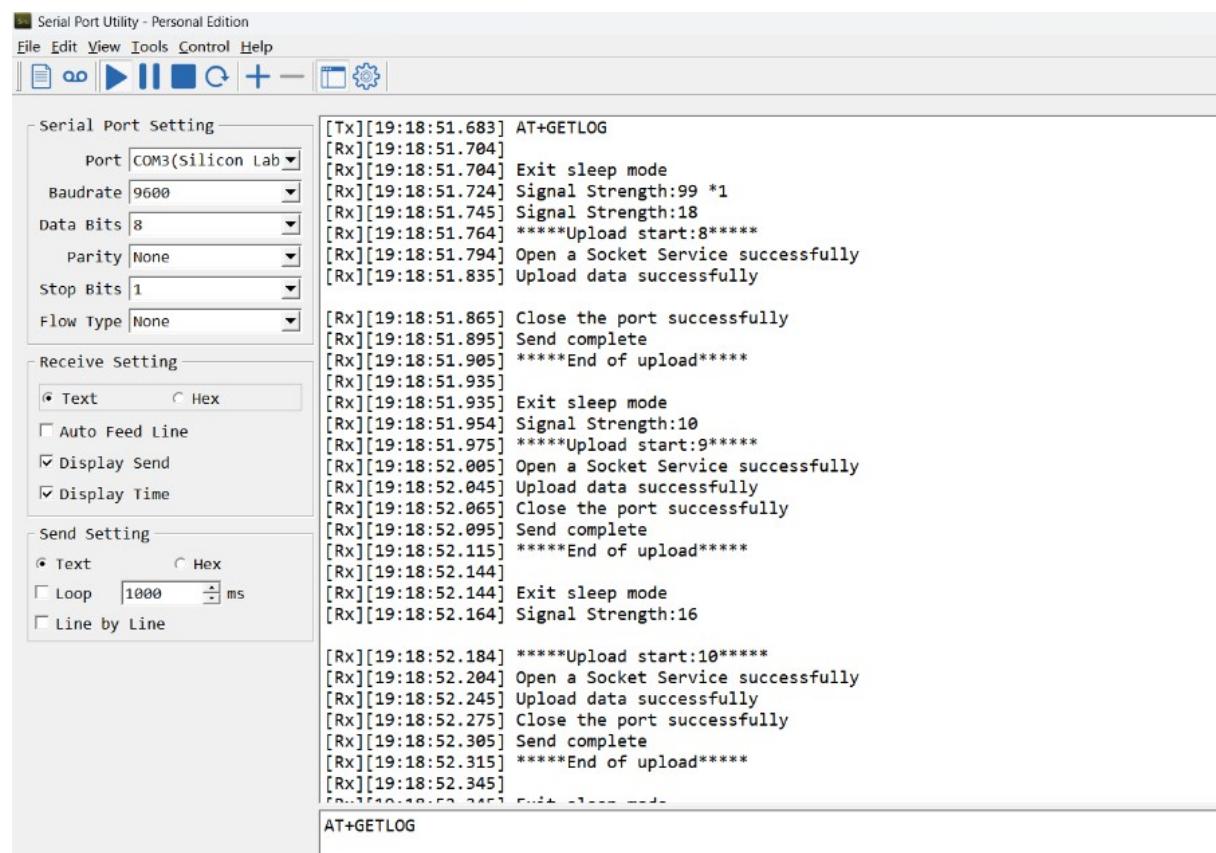
Mit diesem Befehl können Sie den gespeicherten Verlauf durchsuchen, wobei bis zu 32 Datengruppen aufgezeichnet werden können, wobei jede Gruppe historischer Daten maximal 100 Bytes enthält.



## 2.16 Abfrage des Uplink-Protokolls

- **AT-Befehl: AT+GETLOG**

Mit diesem Befehl können Sie Upstream-Protokolle von Datenpaketen abfragen.



## 2.17 Geplante Auflösung von Domänennamen

Dieser Befehl wird verwendet, um die geplante Auflösung von Domänennamen einzurichten.

### AT-Befehl:

- **AT+DNSTIMER** // Einheit: Stunde

Nach der Einstellung dieses Befehls wird die Auflösung von Domänennamen regelmäßig durchgeführt.

## 2.18 QoS-Stufe festlegen

Dieser Befehl dient zum Einstellen der QoS-Stufe von MQTT.

### AT-Befehl:

- **AT+MQOS** // 0~2

### Downlink-Befehl: 0x07

Format: Befehlscode (0x07) gefolgt von 1 Byte. **Beispiel 1:**

Downlink-Nutzlast: **0x0700** //AT+MQOS=0 **Beispiel 2:**

Downlink-Nutzlast: **0x0701** //AT+MQOS=1

## 3. RS485-NB konfigurieren

### 3.1 Konfigurationsmethoden

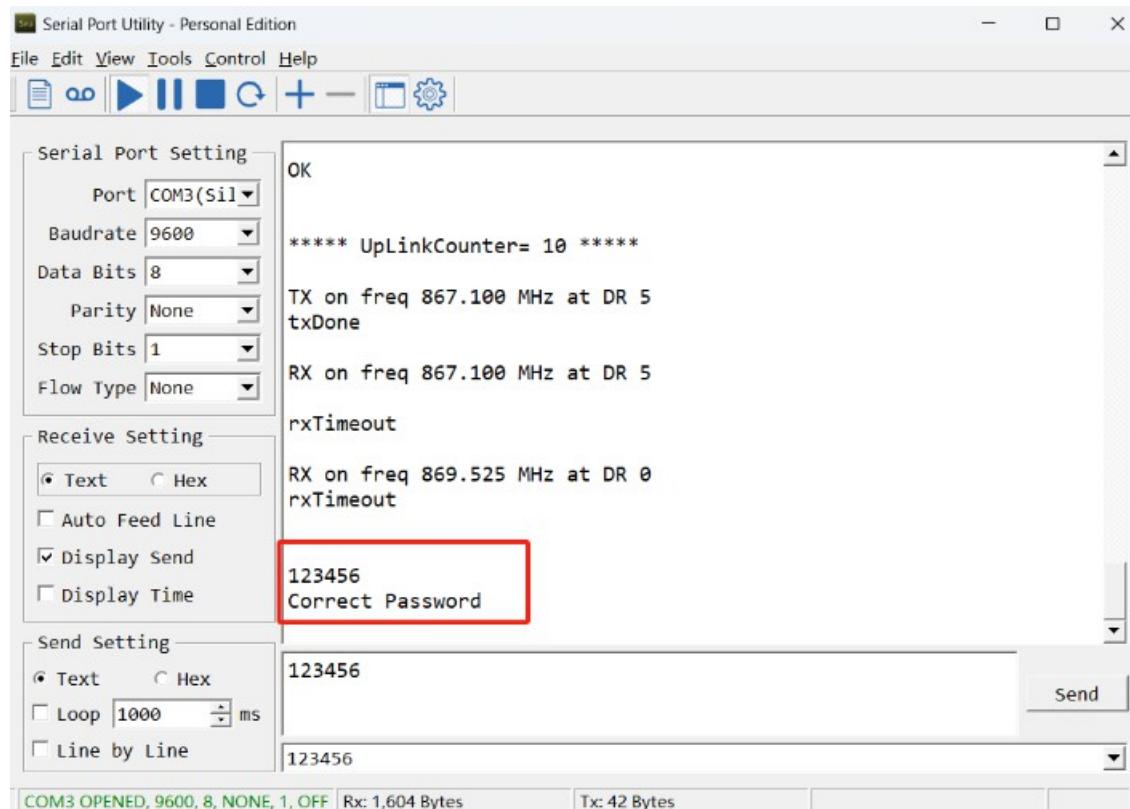
RS485-NB unterstützt die folgenden Konfigurationsmethoden:

- AT-Befehl über Bluetooth-Verbindung (**empfohlen**): [BLE-Konfigurationsanweisung](#).
- AT-Befehl über UART-Verbindung: Siehe [UART-Verbindung](#).

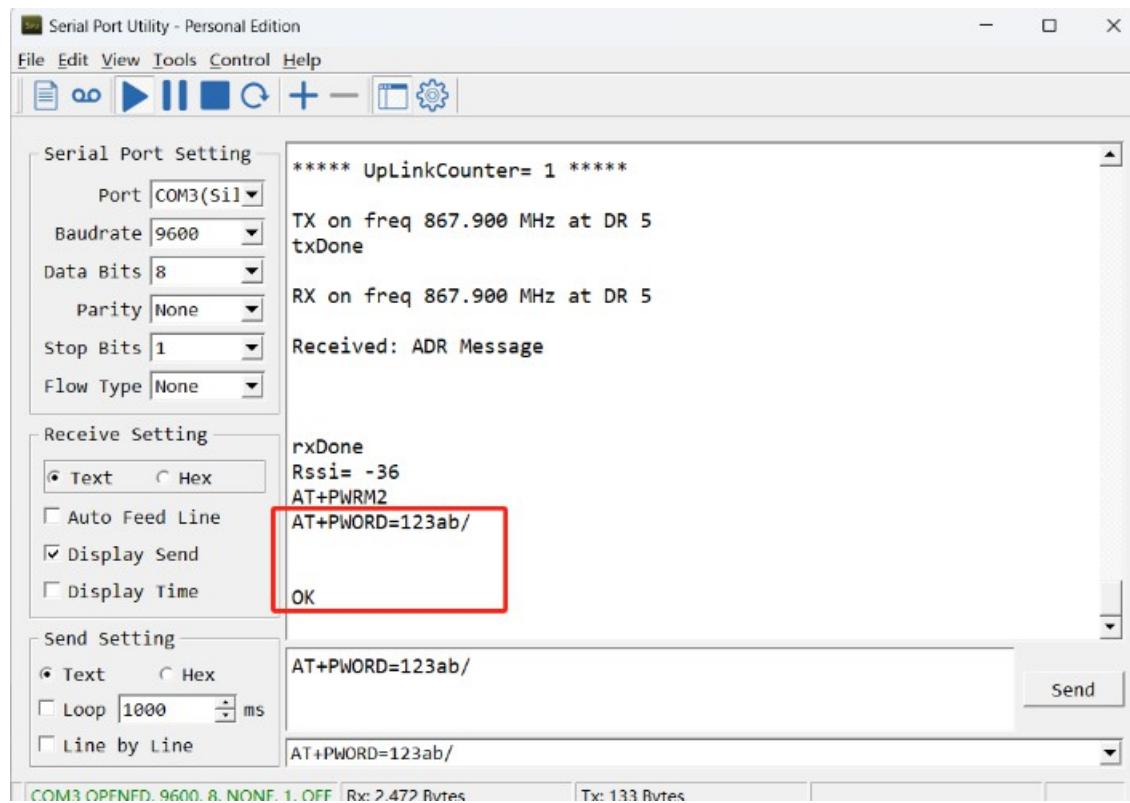
### 3.2 Passwort für den seriellen Zugriff

Nachdem die Bluetooth- oder UART-Verbindung erfolgreich hergestellt wurde, verwenden Sie das Passwort für den seriellen Zugriff, um das AT-Befehlsfenster zu öffnen.

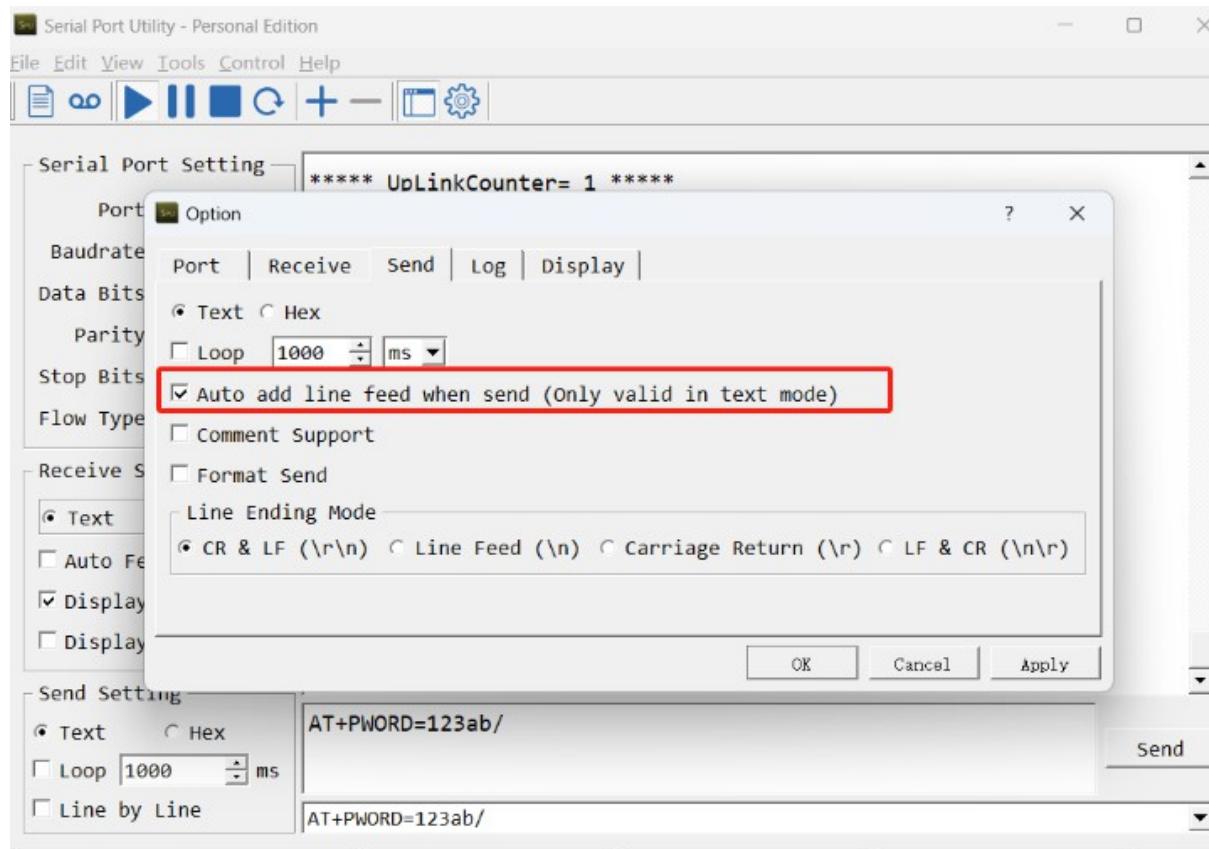
Das Etikett auf der Verpackung des Knotens enthält das ursprüngliche Passwort: **AT+PIN=xxxxxx**. Verwenden Sie dieses sechsstellige Passwort, um direkt auf das AT-Befehlsfenster zuzugreifen.



Wenn Sie das Passwort ändern müssen, verwenden Sie AT+PWORD=xxxxxx (6 Zeichen). NB-Knoten unterstützen nur Kleinbuchstaben.



**Hinweis: Nach Eingabe des Befehls müssen Sie einen Zeilenumbruch einfügen. Sie können auch automatische Zeilenumbrüche im Bluetooth-Tool oder im UART-Verbindungstool einstellen.**



### 3.3 AT-Befehlssatz

AT+<CMD>? : Hilfe zu <CMD>

AT+<CMD> : <CMD>

ausführen

AT+<CMD>=<Wert> : Wert festlegen

AT+<CMD>=? : Wert abrufen

#### Allgemeine Befehle

AT : Achtung

AT? : Kurze Hilfe

ATZ : MCU zurücksetzen

AT+TDC : Intervall für die Übertragung von Anwendungsdaten

AT+CFG : Alle Konfigurationen ausdrucken

AT+CFGMOD : Auswahl des Arbeitsmodus

AT+DEUI : Gerät-ID abrufen oder  
festlegen AT+PRO : Vereinbarung auswählen

AT+RXDL : Sende- und Empfangszeit verlängern

AT+DNSCFG: DNS-Server abrufen oder festlegen

AT+GETSENSORVALUE: Gibt den aktuellen Sensorwert zurück AT+SERVADDR: Serveradresse

AT+MOD: Arbeitsmodus abrufen oder festlegen

AT+5VT : Zeit für 5-V-Stromversorgung abrufen oder

einstellen AT+3V3T : Zeit für 3V3-

Stromversorgung abrufen oder einstellen

AT+INTMOD : Abrufen oder Einstellen des Trigger-Interrupt-Modus (0:Eingang, 1:fallend oder steigend,

2:fallend, 3:steigend) AT+BAUDR: Abrufen oder Einstellen der Baudrate von UART

AT+DATABIT: Abrufen oder Festlegen des Datenbits (7: 7 Bit, 8: 8 Bit)

von UART AT+PARITY: Parität (0: keine, 1: ungerade, 2: gerade) von

UART abrufen oder festlegen AT+STOPBIT: Stopbit (0: 1 Bit,

1: 1,5 Bit, 2: 2 Bit) des UART abrufen oder einstellen

AT+CMDEAR: Befehl zum Löschen (Anfangsnummer bis Endnummer)

AT+PAYVER: Nutzlastversion abrufen oder festlegen

AT+MBFUN: Modbus-Funktionscode abrufen oder festlegen (0: keine, 1: 01 oder 02, 2: 03 oder 04) für automatischen Schnitt AT+COMMAND: Senden Sie Daten von Befehl 1-15 an RS485-Gerät für Nutzlast

AT+SEARCH: Ruft das Hexadezimalzeichen ab, das in Befehl 1-15 erscheint AT+DATAUT:

Empfangsdaten nach Verwendung von Befehl 1-15 abschneiden

AT+CMDDL: Verzögerungszeit nach dem Senden von RS485-Befehl 1-

15 einstellen AT+CFGDEV: Daten an RS485-Gerät senden

AT+DNSTIMER: Domänennamen regelmäßig auflösen

#### **MQTT-Verwaltung**

AT+CLIENT : MQTT-Client abrufen oder festlegen

AT+UNAME : MQTT-Benutzernamen abrufen oder

festlegen AT+PWD : MQTT-Passwort abrufen oder festlegen

AT+PUBTOPIC : MQTT-Veröffentlichungsthema abrufen

oder festlegen AT+SUBTOPIC

: MQTT-Abonnementthema abrufen oder

festlegen

#### **Informationen**

AT+FDR : Zurücksetzen auf

Werkseinstellungen AT+PWD : Passwort

für seriellen Zugriff AT+LDATA

: Letzte hochgeladene Daten

abrufen

## **4. Akku und Stromverbrauch**

RS485-NB verwendet einen ER26500 + SPC1520-Akku. Unter dem folgenden Link finden Sie detaillierte Informationen zum Akku und zum Austausch.

[Batterieinformationen und Analyse des Stromverbrauchs.](#)

## 5. Firmware-Update

Der Benutzer kann die Geräte-Firmware ändern, um:

- Aktualisierung mit neuen Funktionen.
- Fehler beheben.

Die Firmware und das Änderungsprotokoll können heruntergeladen werden unter: [Link zum Herunterladen der Firmware](#)

Methoden zum Aktualisieren der Firmware:

- (Empfohlene Methode) OTA-Firmware-Update über BLE: [Anleitung](#).
- Aktualisierung über die UART-TTL-Schnittstelle: [Anleitung](#).

## 6. Häufig gestellte Fragen

### 6.1 Wie kann ich auf die BC660K-GL AT-Befehle zugreifen?

Der Benutzer kann direkt auf den BC660K-GL zugreifen und AT-Befehle senden.

[Siehe BC660K-GL AT-Befehlssatz](#)

### 6.2 Wie viele RS485-Slaves kann RS485-NB verbinden?

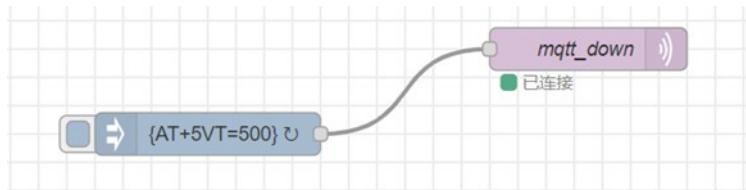
Der RS485-NB kann maximal 32 RS485-Geräte unterstützen. Jeder Uplink-Befehl des RS485-NB kann maximal 16 verschiedene RS485-Befehle unterstützen. Somit kann der RS485-NB maximal 16 RS485-Geräte unterstützen, die im Gerät für den Uplink vorprogrammiert sind. Für andere Geräte, die nicht vorprogrammiert sind, kann der Benutzer die Downlink-Nachricht (Typcode 0xA8) verwenden, um deren Informationen abzufragen.

### 6.3 Wie konfiguriert man das Gerät über die MQTT-Abonnementfunktion?

Abonnementinhalt: {AT COMMAND}

**Beispiel:**

Die Einstellung von AT+5VT=500 über Node-RED erfordert, dass MQTT den Inhalt {AT+5VT=500} sendet.



Die serielle Schnittstelle zeigt Folgendes an:

```
CMD1    = fe 03 00 00 00 03 11 c4
RETURN1 = fe 03 06 11 11 22 22 33 33 65 54
Payload  = 01 11 11 22 22 33 33
```

```
[42203]BAT:3.584 V
[47629]Opened the MQTT client network successfully
[51187]Successfully connected to the server
AT+PWRM2
[54905]Upload data successfully
[58443]Received downlink data:AT+5VT=500
[59330]Subscribe to topic successfully
[62877]Close the port successfully
[63916]Send complete
[64940]*****End of upload*****
```

## 6.4 Wie verwendet man RS485-NB, um eine Verbindung zu RS232-Geräten herzustellen?

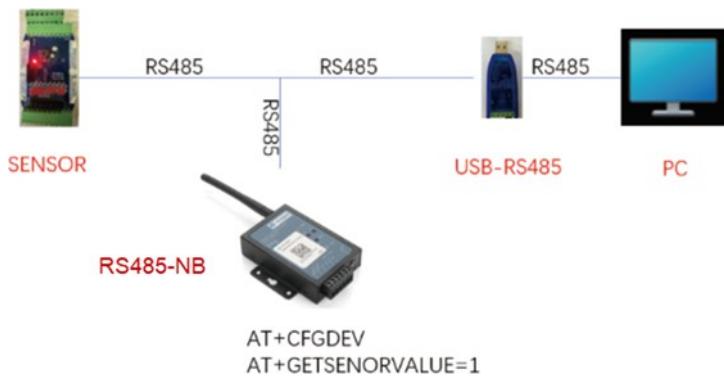
[Verwenden Sie RS485-NB, um eine Verbindung zu RS232-Geräten herzustellen. – DRAGINO](#)

## 6.5 Wie kann man feststellen, ob es ein Problem mit dem eingestellten BEFEHL gibt?

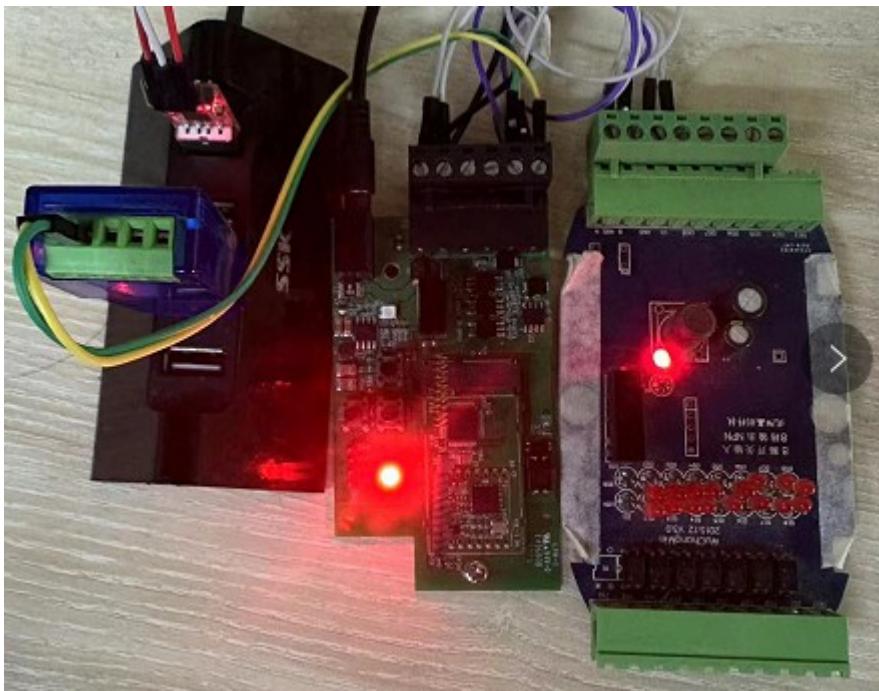
### 6.5.1 Einführung:

Benutzer können die folgende Struktur verwenden, um die Kommunikation zwischen RS485-NB schnell zu debuggen. Das Prinzip besteht darin, den PC in das RS485-Netzwerk einzubinden und die Pakete zwischen Modbus MTU und RS485-NB zu sniffen. **Auf diese Weise können wir:**

1. Testen, ob Modbus-MTU mit PC-Befehlen funktioniert.
2. Überprüfen, ob RS485-NB den erwarteten Befehl an Modbus-MTU gesendet hat.
3. Überprüfen, ob Modbus-MTU das erwartete Ergebnis an RS485-NB zurückgibt.
4. Wenn sowohl b) als auch c) Probleme bereiten, können wir die Ausgabe des PCs und die Ausgabe des RS485-NB vergleichen.

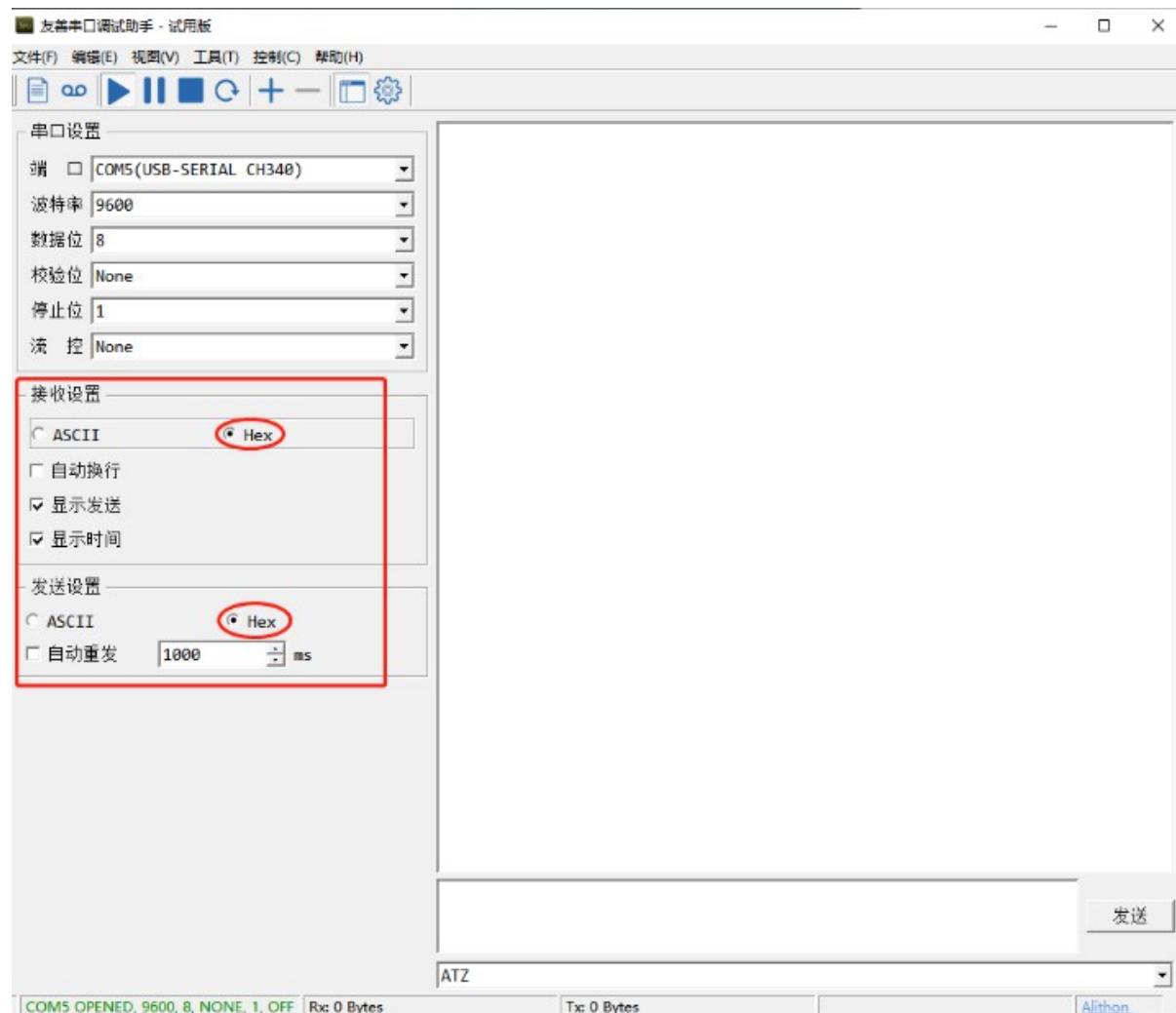


**Beispiel für eine Verbindung:**



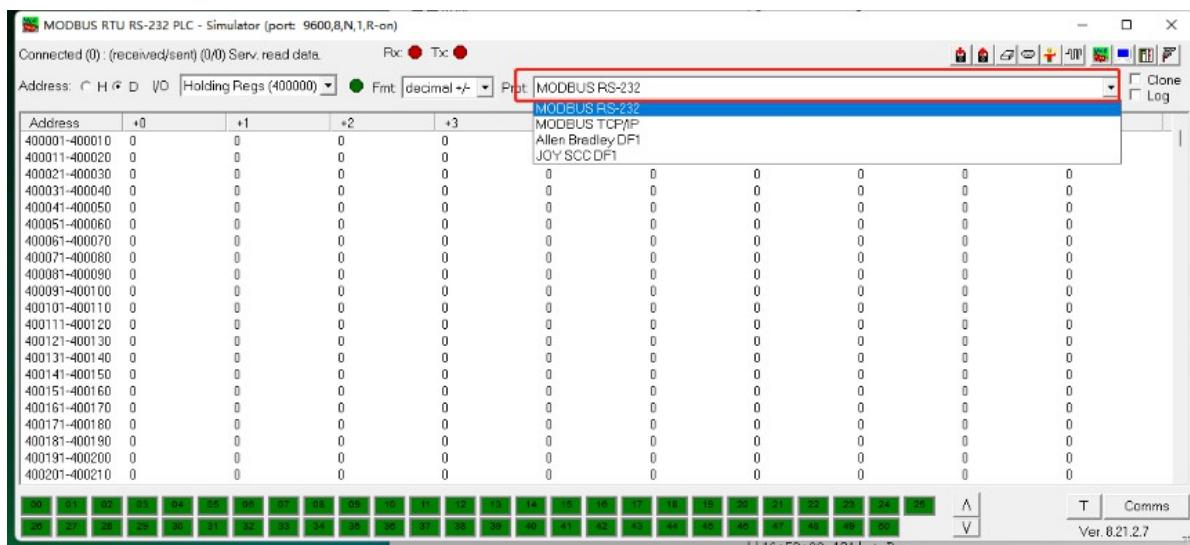
### 6.5.2 PC mit Serial Tool für die Überwachung des RS485-Netzwerks einrichten

**Hinweis: Empfang und Sendung auf Hex-Modus einstellen**

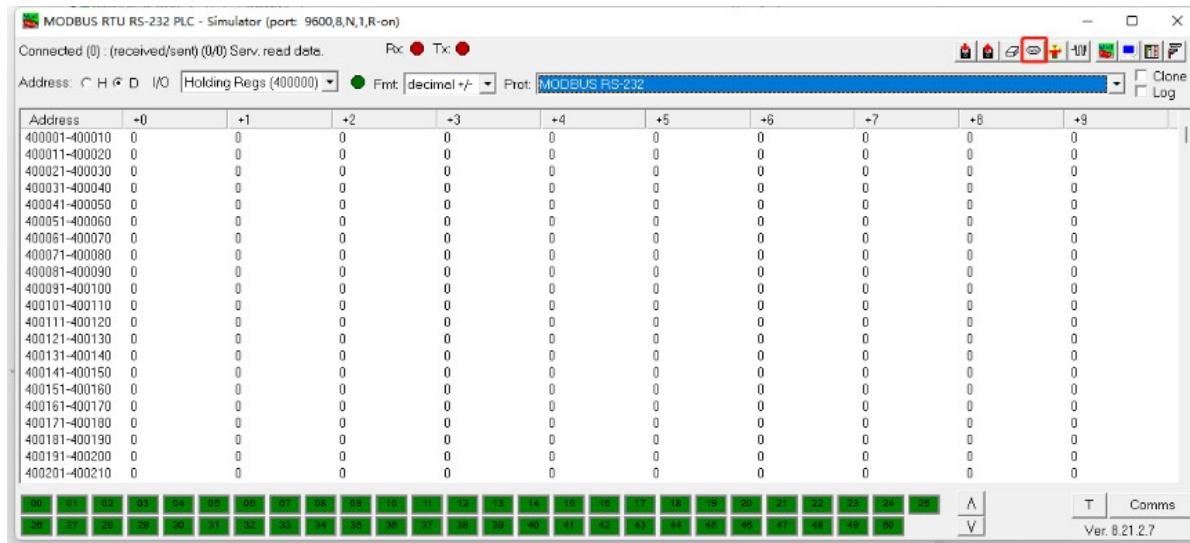


### 6.5.3 Mit ModRSsim2:

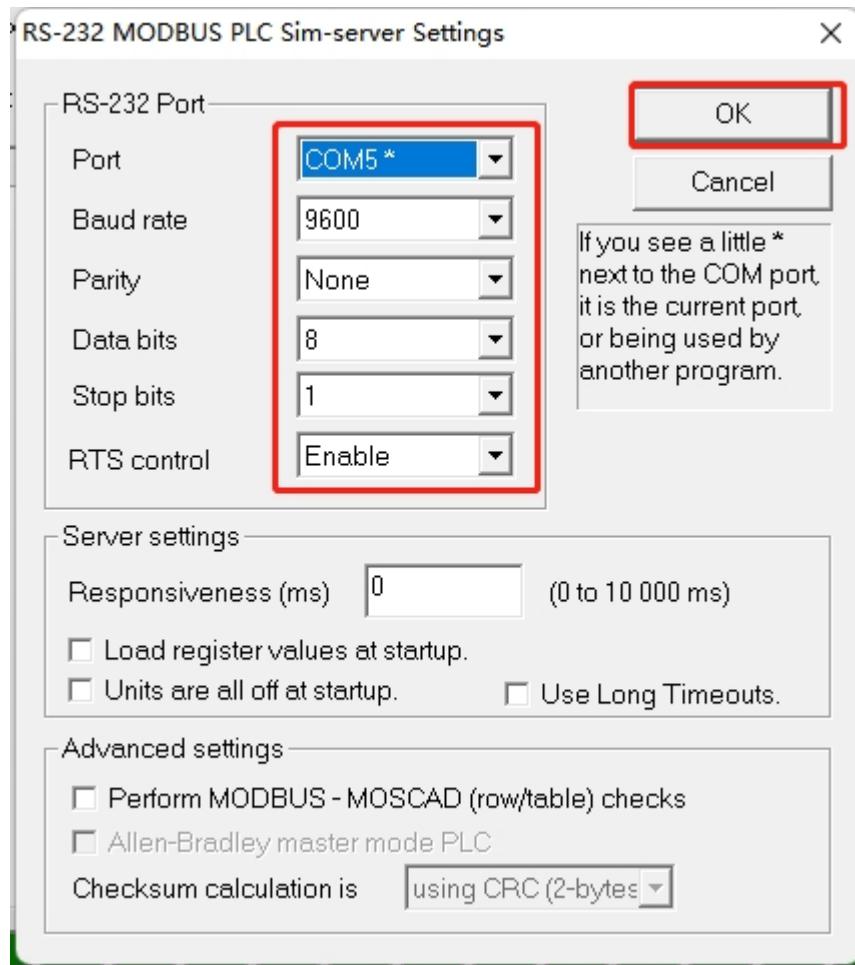
#### (1) Serielle Schnittstelle MODBUS RS-232 auswählen



**(2) Klicken Sie auf das Symbol für die serielle Schnittstelle**



**(3) Nachdem Sie den richtigen seriellen Anschluss und die richtige Baudrate ausgewählt haben, klicken Sie auf „OK“.**



**(4) Klicken Sie auf „Comms“.**



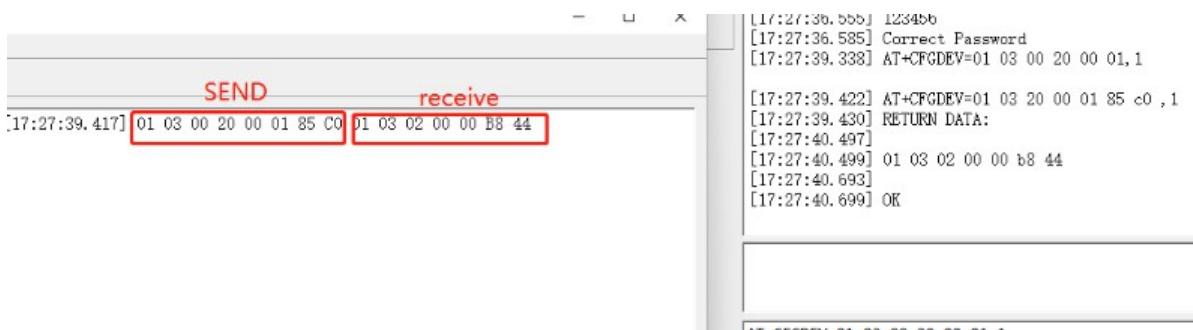
Führen Sie den Befehl RS485-NB aus und überprüfen Sie, ob er korrekt ist.

#### 6.5.4 Beispiel – Testen Sie den Befehl CFGDEV

RS485-LN hat den folgenden Befehl gesendet:

**AT+CFGDEV=01 03 20 00 01 85 c0,1** an das RS485-Netzwerk, und der PC kann diesen Befehl empfangen und Befehle von der MTU zurückgeben, die im seriellen Tool angezeigt werden.

Wir können die Ausgabe aus dem seriellen Port-Tool zur Analyse anzeigen. Und überprüfen, ob sie das erwartete Ergebnis sind.



Wir können auch ModRSsim2 verwenden, um die Ausgabe anzuzeigen.

```
17:09:26.391 Starting server(s)...
17:09:26.392 Starting comms emulation : MODBUS RS-232
17:09:26.393 Port COM5 opened.
17:09:26.393 Port I/O buffers configured.
17:09:26.394 Port configured 9600,8,N,1
17:09:26.394 Timeout configured (100ms/500ms)
17:09:26.394 Modem status: [CTS, 1|DSR, 1|RING, ]
7:18:32.517 RX:01 03 00 20 00 01
7:18:32.518 RX:05 C0 01 03 02 00
7:18:32.518 m_PDUSize is 256
7:18:32.518 Read Holding Regs from 32 for 1.
7:18:32.518 TX:01 03 02 00 00 b8 44
```

00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99	100
自动重发		1000	ms																																																																																																	

```
[17:17:58.821] 123456
[17:17:58.851] Correct Password
[17:18:32.440] AT+CFGDEV=01 03 00 20 00 01, 1
[17:18:32.523] AT+CFGDEV=01 03 20 00 01 85 c0 , 1
[17:18:32.527] RETURN DATA:
[17:18:33.601] 01 03 02 00 00 b8 44 01 03 02 00 00 b8 44
[17:18:33.848]
[17:18:33.850] OK
```

```
AT+CFGDEV=01 03 00 20 00 01, 1
```

COM12 OPENED 9600 8 NONE 1 OFF Rx: 16 297 Bytes Tx: 200 Bytes Althon

### 6.5.5 Beispiel – Testen Sie die CMD-Befehlssätze.

Führen Sie **AT+SENSORVALUE=1** aus, um die CMD-Befehlssätze in RS485-LN zu testen.

**Serielles Port-Tool:**

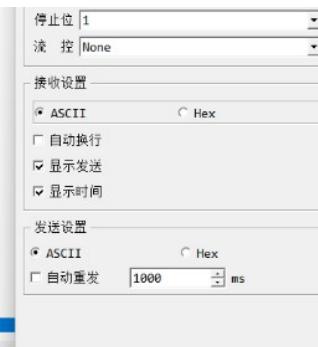
[17:27:39.417] 01 03 00 20 00 01 85 C0 01 03 02 00 00 B8 44 01 03 00 20 00 01 85 C0 01 03 02 00 00 B8 44 01 03 00 22 00 01 24 00 01 03 02 00 20 B9 9C 01 03 00 23 00 01 75 C0 01 03 02 00 00 B8 44 01 03 00 24 00 01 C4 01 01 83 01 80 F0	[17:27:40.699] OK [17:30:01.230] AT+GETSENSORVALUE=1
	[17:30:01.260]
	[17:30:01.261] OK
	[17:30:03.349]
	[17:30:03.351] CMD1 = 01 03 00 20 00 01 85 c0
	[17:30:03.401]
	[17:30:03.402] RETURN1 = 01 03 02 00 00 b8 44
	[17:30:03.440]
	[17:30:03.441] CMD2 = 01 03 00 21 00 01 d4 00
	[17:30:03.487]
	[17:30:03.491] RETURN2 = 01 03 02 00 00 b8 44
	[17:30:03.532] CMD3 = 01 03 00 22 00 01 24 00
	[17:30:03.571] RETURN3 = 01 03 02 00 20 b9 9c
	[17:30:03.611]
	[17:30:03.611] CMD4 = 01 03 00 23 00 01 75 c0
	[17:30:03.661] RETURN4 = 01 03 02 00 00 b8 44
	[17:30:03.700]
	[17:30:03.702] CMD5 = 01 03 00 24 00 01 c4 01
	[17:30:03.751]
	[17:30:03.751] RETURN5 = 01 83 01 80 f0 00 00
	[17:30:03.779]

### ModRSsim2:

```

17:24:36.080 m_PDUsize is 256
17:24:36.080 Read Holding Regs from 05 for 1.
17:24:36.080 TX:01 03 02 00 00 B8 44
17:24:36.931 RX:01 03 00 23 00 01
17:24:36.931 m_PDUsize is 256
17:24:36.931 Read Holding Regs from 05 for 1.
17:24:36.931 TX:01 03 02 00 00 B8 44
17:24:37.801 RX:01 03 00 24 00 01
17:24:37.801 RX:C4 01 01 83 01 80
17:24:37.902 m_PDUsize is 256
17:24:37.902 Read Holding Regs from 06 for 1.
17:24:37.902 TX:01 03 02 00 00 B8 44
17:24:38.572 RX:01 03 00 24 00 01
17:24:38.572 RX:C4 01 01 83 01 80
17:24:38.572 m_PDUsize is 256
17:24:38.572 Read Holding Regs from 06 for 1.
17:24:38.572 TX:01 03 02 00 00 B8 44
17:24:39.480 RX:01 03 00 24 00 01
17:24:39.490 RX:C4 01 01 83 01 80
17:24:39.490 m_PDUsize is 256
17:24:39.490 Read Holding Regs from 06 for 1.
17:24:39.490 TX:01 03 02 00 00 B8 44

```



```

[17:24:27.387]
[17:24:27.387] OK
[17:24:40.362]
[17:24:40.395]
[17:24:40.397] CMD1 = 01 03 00 20 00 01 85 c0
[17:24:40.446]
[17:24:40.447] RETURN1 = 01 03 02 00 00 b8 44
[17:24:40.487] CMD2 = 01 03 00 21 00 01 d4 00
[17:24:40.537] RETURN2 = 01 03 02 00 00 b8 44
[17:24:40.575]
[17:24:40.577] CMD3 = 01 03 00 22 00 01 24 00
[17:24:40.617] RETURN3 = 01 03 02 00 20 b9 9c
[17:24:40.663]
[17:24:40.667] CMD4 = 01 03 00 23 00 01 75 c0
[17:24:40.704]
[17:24:40.707] RETURN4 = 01 03 02 00 00 b8 44
[17:24:40.745]
[17:24:40.751] CMD5 = 01 03 00 24 00 01 c4 01
[17:24:40.794]
[17:24:40.797] RETURN5 = 01 83 01 80 f0 01 03

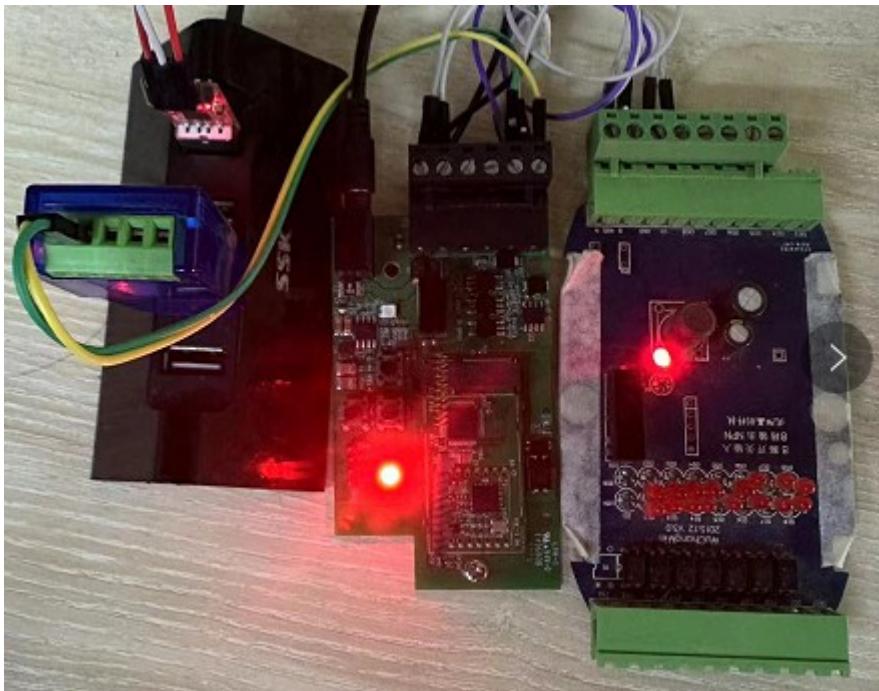
```

### 6.5.6 Test mit PC

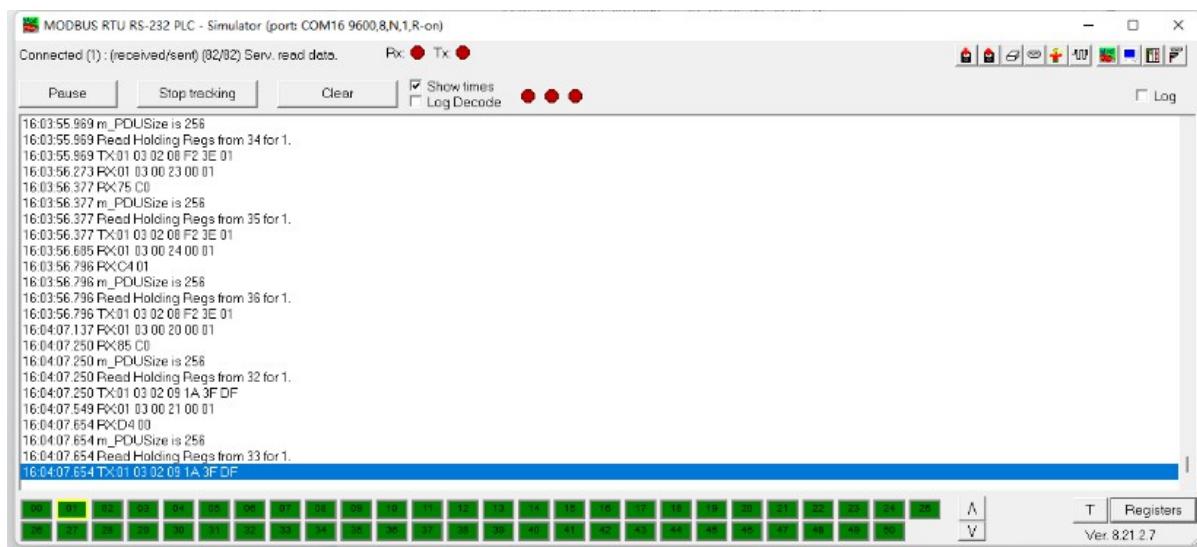
Wenn es weiterhin Probleme bei der korrekten Einrichtung der Befehle zwischen RS485-LN und MTU gibt, kann der Benutzer den korrekten RS485-Befehlssatz auf dem PC testen und mit dem über RS485-LN gesendeten RS485-Befehl vergleichen. Solange beide Befehle identisch sind, sollte die MTU das richtige Ergebnis zurückgeben.

Alternativ kann der Benutzer die im seriellen PC-Tool festgelegten Arbeitsbefehle an den Dragino-Support senden, um zu überprüfen, was in RS485-LN konfiguriert werden muss.

Verbindungs methode:



## Verbindungssituation:



## 7. Bestellinformationen

Teilenummer: **RS485-NB-XX XX**:

- **GE**: Allgemeine Version (ohne SIM-Karte)
- **1D**: mit 1NCE\* 10 Jahre 500 MB SIM-Karte und vorkonfiguriert für DataCake-Server

**YY: Größe der Anschlussbuchse**

- **M12**: M12-Loch
- **M16**: M16-Loch

## 8. Verpackungsinformationen

### Lieferumfang

- RS485-NB NB-IoT-Sensorknoten x 1
- Externe Antenne x 1

### Abmessungen und Gewicht:

- Gerätgröße: 13,0 x 5 x 4,5 cm
- Gerätgewicht: 150 g
- Verpackungsgröße/Stück: 14,0 x 8 x 5 cm
- Gewicht/Stück: 180 g

## 9. Support

- Der Support ist montags bis freitags von 09:00 bis 18:00 Uhr GMT+8 verfügbar. Aufgrund unterschiedlicher Zeitzonen können wir keinen Live-Support anbieten. Ihre Fragen werden jedoch so schnell wie möglich innerhalb der oben genannten Zeiten beantwortet.
- Geben Sie so viele Informationen wie möglich zu Ihrer Anfrage an (Produktmodelle, genaue Beschreibung Ihres Problems und Schritte zur Reproduktion usw.) und senden Sie eine E-Mail an [Support@dragino.cc](mailto:Support@dragino.cc).