

Datenblatt Universal Modbus Client/Master SL



Mit den einfach zu nutzenden Modbus Funktionsbausteinen können Werte über Modbus TCP, Modbus UDP oder Modbus RTU gelesen und geschrieben werden. Zudem werden die Protokolle Modbus RTU über TCP und Modbus RTU über UDP unterstützt.

Ohne Lizenzwerb (nur Download) arbeitet die Universal Modbus Client Master SL für 30 Minuten ohne funktionale Einschränkung (Demo).

Produktbeschreibung

Mit den Funktionsbausteinen kann die SPS als Modbus TCP/UDC Client bzw. als Modbus RTU Master Daten mit anderen Geräten wie Energiezählern, Reglern oder Sensoren austauschen. Alle relevanten Eigenschaften des Modbus-Protokolls sind implementiert:

- Unterstützte Funktionscodes: FC01, FC02, FC03, FC04, FC05, FC06, FC15, FC16
- Wahlweise lesend, schreibend oder bidirektionaler Zugriff mit automatischer Änderungserkennung
- Automatische Umwandlung der Byte-Order
- Alle IEC Datentypen sind nutzbar

Funktionsbeschreibung

Modbus ist ein einfaches, offenes Kommunikationsprotokoll, mit dem eine Master-Slave oder Client-Server-Kommunikation zwischen den am Netzwerk angeschlossenen Geräten realisiert werden kann. Das Grundprinzip von Modbus ist: Ein Master sendet eine Anfrage und ein Slave antwortet darauf. Bei Modbus TCP wird der Master als Client, ein Slave als Server bezeichnet. Die Funktion ist dieselbe. In weiterer Folge werden für die Beschreibungen der Funktionen des Protokolls unabhängig von den Varianten RTU und TCP nur die gebräuchlicheren Namen Master und Slave verwendet.

Unterstützte Funktionscodes

FC01 (0x01)	Read Coils
FC02 (0x02)	Read Discrete Inputs
FC03 (0x03)	Read Holding Registers
FC04 (0x04)	Read Input Registers
FC05 (0x05)	Write Single Coil
FC06 (0x06)	Write Single Register
FC15 (0x0F)	Write Multiple Coils
FC16 (0x10)	Write Multiple registers

Alle IEC Datentypen

Der Eingang **Value** des Funktionsblock ModbusValue ist vom Typ *Any*. Somit können alle Datentypen, wie BOOL, DINT, REAL, etc. übergeben werden. Auch Strukturen und Arrays sind möglich. Zu beachten ist jedoch, dass in diesem Fall keine automatische Anpassung der Byteorder erfolgt.

Automatische Anpassung der Byte Order

In der Regel stellen Modbus-Geräte ihre Daten in Form der Big-Endian-Byteorder (auch Motorola Byteorder genannt) zur Verfügung. Sofern es sich bei dem Eingang **Value** um einen Standarddatentyp handelt, wird die Anpassung der Byteorder automatisch vorgenommen. Dieses

Verhalten kann mittels des Eingangs **xUseBigEndian** deaktiviert werden.

Bidirektionaler Datenaustausch

Ist der Eingang **eReadWriteType** auf den Wert *Modbus_Bidir* gesetzt, erkennt der Funktionsblock automatisch Änderungen durch das SPS-Programm und sendet einen entsprechenden Schreib-Befehl zum Endgerät. Wurde der Wert nicht geändert, so wird der Wert vom Zielgerät gelesen.

(Screenshot Beispiel)

Je ein Funktionsblock für jede Verbindung

Je ein Funktionsblock für jeden zu lesenden Wert, Struktur oder Array

The screenshot shows the 'FUNCTION_BLOCK Janitza_UMG' definition with the following code:

```

1 FUNCTION_BLOCK Janitza_UMG
2 VAR_INPUT
3   byUnitId      : BYTE;
4 END_VAR
5 VAR_OUTPUT
6   rFrequency    : REAL;
7   rRealEnergy   : REAL;
8 END_VAR
9 VAR
10  connection    : ModbusClient_RTU;
11  read_Frequency : ModbusValue;
12  read_RealEnergy : ModbusValue;
13 END_VAR
14

```

Below the code, three instances of the function block are shown:

- Instance 1 (connection):** A 'ModbusClient_RTU' block with inputs: xUseBigEndian, sPort, byStopBits, byParity, ulBaudrate. It has an output xConnected.
- Instance 2 (read_Frequency):** A 'ModbusValue' block with inputs: rFrequency, byUnitId, byFunctionCode (3), wRegisterAddress (19050), Modbus_Read, connection. It has outputs: Value, eError, xBusy, xChanged.
- Instance 3 (read_RealEnergy):** A 'ModbusValue' block with inputs: rFrequency, byUnitId, byFunctionCode (3), wRegisterAddress (19060), Modbus_Read, connection. It has outputs: Value, eError, xBusy, xChanged.

Beispiel für eine Anbindung eines Energiezählers in FUB

Allgemeine Informationen



Hersteller	Mirasoft GmbH & Co. KG Steingraben 13 97788 Neuendorf Deutschland
Support	Tel: +49 9351 9793320 support@mirasoft.de
Artikel	Universal Modbus Client or Master
Artikelnummer	2312000008
Vertrieb	https://www.mirasoft.de https://www.anyviz.de
Lieferumfang	▪ Package Universal Modbus Client or Master

Systemvoraussetzungen und Einschränkungen

Programmiersystem	CODESYS Development System V3.5.8.0
Laufzeitsystem	CODESYS Control V3.5.8.0
Unterstützte Plattformen / Geräte	Hinweis: Verwenden Sie das Projekt "Device Reader", um die von der Steuerung unterstützten Funktionen zu ermitteln. „Device Reader“ ist kostenlos im CODESYS Store erhältlich.
Zusätzliche Anforderungen	
Einschränkungen	
Lizenzierung	Lizenzaktivierung optional auf CODESYS Runtime Key oder CODESYS Soft Key (kostenloser Bestandteil der CODESYS Controls)
Erforderliches Zubehör	