

Inhaltsverzeichnis

Vorwort.....	1
1. Allgemeine Beschreibung.....	2
1.1 Aufbau allgemein.....	3
2. Mögliche Funktionscodes für unit+ V1/V2 und econ4:	5
2.1 Funktionscodes Beschreibung allgemein.....	5
3. Modbus Basis Format Templates für unit+ und econ4, Aufbau und Syntax *.....	6
4. econ unit+ V1/V2 und econ4.....	7
4.1 econ unit+ V1	7
4.2 econ unit+ V2	8
4.3 econ4.....	9
5. Fehlercodes allgemein	10
6. Detaillierte Erläuterung zur Benennung.....	11
7. Anbindungsmöglichkeiten unit+ V1/V2, econ4 und Portangaben.....	12
8. Unterschiede bei der Byteorder unit+ V1/ unit+ V2/ econ4.....	12
9. Modbusinfo allgemein (Slaveadressen, Schreibweise).....	12
10. Mitgelieferte Templates	13

* **Hinweis:** Die Vorlagen der Basis-Format Templates sind in der unit+ V2 (ab V2.0.9) enthalten. Sie können diese Vorlagen auch über unseren econ Download Center herunterladen:
<https://www.econ-solutions.de/de/downloads?name=modbus>
Inhalt der Datei Templatepack_modbus_basic-format.zip:
modbus_basic_format_template.json / modbus_basic_format_template_LC.json

Vorwort

Diese Anleitung soll Ihnen die Erstellung eines Templates für Ihre Geräte ermöglichen, so dass Sie diese an eine unit+ oder an die econ4 selbst anbinden können.

Damit Sie sofort starten können, werden mit dieser Anleitung die in Kapitel 3 aufgeführten Basis-Format Templates schon mitgeliefert.

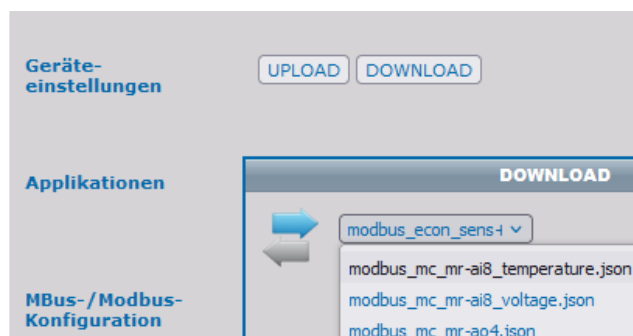
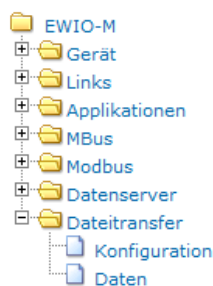
Falls Sie für die Erstellung oder den Support von Templates Unterstützung benötigen, können Sie unter auftragsmanagement@econ-solutions.de direkt ein Angebot für diese Dienstleistungen (Berechnung nach Aufwand) anfordern. Wir setzen uns gerne nach schriftlicher Bestellung schnellstmöglich mit Ihnen in Verbindung, vielen Dank.

Ihr econ Team

1. Allgemeine Beschreibung

Die Templates für Modbusgeräte/Modbusanbindung, als auch der Up- und Download, sind über die Weboberfläche bei den units unter dem Menüpunkt *Dateitransfer/Konfiguration/Geräteeinstellungen (V1)*, *Zähler/Modbus Einstell. (V2)* und in econ4 unter dem Menüpunkt *Unit- und Messstellenverwaltung/Andere Importer/angelegte Importquelle* zu finden.

unit+ V1



unit+ V2

Vorlagen für Modbus-Zähler

Sie können hier Vorlagen für Modbus-Zähler herunterladen oder löschen. Bitte beachten Sie, dass Vorlagen, die von eingerichteten Zählern werden).

Bitte wählen Sie eine oder mehrere Dateien aus (mehrere ausgewählte Dateien werden zum Herunterladen in ein Archiv (.tar.bz2) verpackt).

Alle auswählen

ecsEM72_100 (modbus_es_ecsEM72_100.json, 8 kB)
ecsEM76MID_100 (modbus_es_ecsEM76MID_100.json, 8 kB)
ecsPM66MID_300 (modbus_es_ecsPM66MID_300.json, 10 kB)
ecsPM69MID_300 (modbus_es_ecsPM69MID_300.json, 10 kB)
ecs_300_LC (modbus_es_ecs_300_LC.json, 10 kB)
janitza_umg508 (modbus_es_janitza_umg508_arbeit.json, 1 kB)
kbr_multimes-4F96-LCD (modbus_es_kbr_multimes_4F96-LCD.json, 3 kB)
econ_sens+ (modbus_es_sens+.json, 9 kB)
econ_sens3 (modbus_es_sens3_G2.json, 10 kB)
siemens SENTRON PAC3200 4200 (modbus_es_siemens_sentrion_pac3200_4200.json, 2 kB)



Herunterladen

Vorlage für Modbus-Zähler hochladen (.json):

Durchsuchen...

Hochladen

econ4

Konfigurationsdatei:	<div>Bitte wählen</div>	 Datei herunterladen
Neue Modbus JSON Konfiguration hochladen:	<div>Durchsuchen... Keine Datei ausgewählt.</div>	 Datei hochladen

In den jeweiligen Menüpunkten können die Templates hoch- bzw. heruntergeladen werden.

1.1 Aufbau allgemein

Die Benennung des Templates muss nach diesem Schema erfolgen, da diese sonst nicht als solche erkannt und verarbeitet werden:

modbus_Hersteller_Geraetetyp.json (z.B. modbus_econ_sens_plus.json)

Es dürfen keine Sonderzeichen (\ / : . ? * < > | etc.) Leerzeichen und Umlaute verwendet werden.

Die Form, Art und Syntax sind für alle Templates gleich. Unterschiede zwischen den units und econ4 gibt es lediglich in der Handhabung: die Funktion eines Templates ist u.a. abhängig von den unterstützten Datentypen, den Funktionscodes und den technischen Voraussetzungen/Ausstattung. So kann z.B. an der unit+ V1 kein ModbusTCP Gerät angebunden werden, sondern die Anbindung ist nur über ModbusRTU möglich. Grundsätzlich funktioniert ein Template auf den units und auf econ4 gleichermaßen.

Manche Syntaxfehler in der JSON werden nicht unbedingt als Fehler erkannt und gemeldet, sondern die Auswahl der im Template hinterlegten Datenpunkte wird zwar angezeigt, aber es erfolgt z.B. keine Aufzeichnung. Es muss dann das Template nochmals auf Syntaxfehler jeglicher Art überprüft werden. Das Template ist im JSON-Format aufgebaut, die korrekte Syntax ist also unbedingt einzuhalten.

Unter TX sind die Funktionscodes, das Startregister und die Anzahl der folgenden Adressen für die Sende-Frames abgelegt. Wenn z.B. eine neue Registeradresse in einen der bereits vorhandenen Bereiche fällt (in korrekter Folge), so ist hier nur die Länge zu erweitern, andernfalls muss ein neuer Bereich mit entsprechender Startadresse angelegt werden.

Unter RX sind dann die einzelnen Register aufgeführt.

Hier sind neue Register mit ihrer Adresse an der passenden Stelle einzufügen.

Für das Erstellen und Bearbeiten von Templates wird ein Programmtexteditor empfohlen (z.B. Notepad++), da die JSON eine Programmdatei ist und im entsprechenden Format abgespeichert werden muss, damit das Template funktioniert.

Um sicherzustellen, dass eine fehlende Aufzeichnung auf das Template zurückzuführen ist, sollte über ein Modbus-Test-Programm (z.B. Modbus Poll), das einzelne Gerät auf Erreichbarkeit, Datengleichheit, etc. überprüft werden.

Info zum LC-Block:

Dieser wird nur für die unit+ V2 benötigt (ab FW1.5.23), für Geräte, welche beim Livecheck ein „rotes Kreuz“ anzeigen, obwohl diese ansonsten (Aufzeichnung) funktionieren. Templates mit LC-Block können auch in der unit+ V1 ab FW 2.19ES und in econ4 ab FW 4.0.22.0 verwendet werden, der LC-Block wird ignoriert.

Empfehlung: Es ist am einfachsten, das erste in RX angelegte Register auszuwählen, da damit kein Wert, sondern nur die Erreichbarkeit abgefragt wird.

WICHTIG: Nach dem Upload eines Templates auf eine unit+ ist ein Neustart/Reboot erforderlich, erst dann wird das Template korrekt erkannt und funktioniert.

Auf den nachfolgenden Seiten sind noch weiter zu beachtende Infos und der grundsätzliche Aufbau des Templates mit entsprechender Syntax zu finden. Die aufgeführten Registertypen sind nur beispielhaft, welche Registertypen/Datentypen möglich sind, finden Sie im entsprechenden Kapitel zu den units, bzw. econ4.

Das Abspeichern der Templates muss in UTF-8 erfolgen und Format Zeilenende muss UNIX (LF) sein. Des Weiteren sind die im Template enthaltenen Leerzeichen und Tabulatoren zu beachten, da diese z.T. für die Funktion benötigt werden. Sollte der TX- und RX-Block aus mehr wie einer Zeile bestehen, so sind diese mit einem Komma abzuschließen, bei der letzten Zeile ist kein Komma zu setzen, d.h. z.B., wenn nur 1 Zeile vorhanden ist, so muss diese ohne Komma sein.

WICHTIG: Sollte sich das Datenformat/Datentyp des Registers ändern (z.B. vorhergehend FLOAT32 HL, nachfolgend INT64 HL), so sollte unter TX ein neuer Block eingetragen werden.

Als Hilfe für die Syntaxprüfung kann z.B. das Template über das Online-Tool <https://jsonlint.com/> auf Standardfehler geprüft werden (es werden nicht alle Fehler angezeigt, nur die für ein Standard-Json)

Beispiel mit Darstellung der Tabulatoren und dem Zeilenende (aus Notepad++):

```

1 {
2   "device":
3   {
4     "Address":0,
5     "AddressBase":0,
6     "Type": "manufacturer_type",
7     "TX":
8     {
9       { "function":3, "start":0, "length":10,
10        { "function":3, "start":10, "length":20,
11         { "function":3, "start":20, "length":30}
12      },
13      "RX":
14      [
15        { "register":0, "format": "INT32.HL", "unit": "unit", "description": "description", "select": ""},
16        { "register":10, "format": "INT32.HL", "unit": "unit", "description": "description", "select": ""},
17        { "register":20, "format": "INT32.HL", "unit": "unit", "description": "description", "select": ""}
18      ]
19    }
20 }
21

```

2. Mögliche Funktionscodes für unit+ V1/V2 und econ4:

- 01 (0x01) Read Coils (nur unit+ V2)
- 02 (0x02) Read Discrete Inputs (nur unit+ V2)
- 03 (Read Holding Registers)
- 04 (Read Input Registers)

2.1 Funktionscodes Beschreibung allgemein

Modbus Functions

- 01 (0x01) Read Coils
- 02 (0x02) Read Discrete Inputs
- 03 (0x03) Read Holding Registers
- 04 (0x04) Read Input Registers
- 05 (0x05) Write Single Coil
- 06 (0x06) Write Single Register
- 08 (0x08) Diagnostics (Serial Line only)
- 11 (0x0B) Get Comm Event Counter (Serial Line only)
- 15 (0x0F) Write Multiple Coils
- 16 (0x10) Write Multiple Registers
- 17 (0x11) Report Server ID (Serial Line only)
- 22 (0x16) Mask Write Register
- 23 (0x17) Read/Write Multiple Registers
- 43 / 14 (0x2B / 0x0E) Read Device Identification

3. Modbus Basis Format Templates für unit+ und econ4, Aufbau und Syntax *

Basis-Format Template:

```
{
  "device":
  {
    "Address": 0,
    "AddressBase": 0,
    "Type": "manufacturer_type",
    "TX":
    [
      {"function": 3, "start": 0, "length": 2},
      {"function": 3, "start": 10, "length": 2},
      {"function": 3, "start": 20, "length": 2}
    ],
    "RX":
    [
      {"register": 0, "format": "INT32 HL", "unit": "unit", "description": "description", "select":""},
      {"register": 10, "format": "INT32 HL", "unit": "unit", "description": "description", "select":""},
      {"register": 20, "format": "INT32 HL", "unit": "unit", "description": "description", "select":""}
    ]
  }
}
```

Basis-Format Template mit LC-Block:

(wird nur für unit+ V2 benötigt (ab FW1.5.23), für Geräte, welche beim Livecheck ein „rotes Kreuz“ anzeigen, obwohl diese ansonsten funktionieren. Templates mit LC-Block können auch in der unit+ V1 ab FW 2.19ES und in econ4 ab FW 4.0.22.0 verwendet werden, der LC-Block wird ignoriert)

```
{
  "device":
  {
    "Address": 0,
    "AddressBase": 0,
    "Type": "manufacturer_type_LC",
    "LC":
    [
      {"function": 3, "start": 0, "length": 1}
    ],
    "TX":
    [
      {"function": 3, "start": 0, "length": 2},
      {"function": 3, "start": 10, "length": 2},
      {"function": 3, "start": 20, "length": 2}
    ],
    "RX":
    [
      {"register": 0, "format": "INT32 HL", "unit": "unit", "description": "description", "select":""},
      {"register": 10, "format": "INT32 HL", "unit": "unit", "description": "description", "select":""},
      {"register": 20, "format": "INT32 HL", "unit": "unit", "description": "description", "select":""}
    ]
  }
}
```

4. econ unit+ V1/V2 und econ4

4.1 econ unit+ V1

Die einzelnen Elemente sind:

Address	Platzhalter für Slave-Adresse des Geräts, auf 0 belassen
AddressBase	Startadresse des Adressbereichs (0 oder 1)
Type	Bezeichnung des Geräts (beliebiger Text-String)
TX []	Array mit Sende- Elementen Jedes Element enthält die Bestandteile zur Bestimmung eines Sende-Frames.
function	Modbus-Funktion (z.B.3 für "Read Holding Register")
start	Startadresse (Dezimal)
length	Anzahl Bytes (Dezimal)
RX []	Array mit Empfangselementen Jedes Element enthält die Bestandteile zur Bestimmung eines Datenpunktes. Die Angaben erscheinen auf der Weboberfläche als auszuwählende Datenpunkte.
register format	Startadresse des Values Typ des Values, möglich sind beispielhaft: INT8, UINT8, INT16 HL, INT16 LH, UINT16 HL, UINT16 LH, INT32 HL, INT32 LH, UINT32 HL, UINT32 LH, INT32 B0123 (beliebige Reihenfolge) INT48 HL, INT48 LH, UINT48 HL, UINT48 LH, INT48 B012345 (beliebige Reihenfolge) INT64 HL, INT64 LH, UINT64 HL, UINT64 LH, INT64 B01234567 (beliebige Reihenfolge) FLOAT32 HL, FLOAT32 LH, FLOAT64 HL, FLOAT64 LH, HEX8, HEX16 HL, HEX16 LH, HEX32 HL, HEX32 LH, HEX48 HL, HEX48 LH, HEX64 HL, HEX64 LH
unit description select	Einheit als Text-String Beschreibung des Datenpunktes als Text-String Platzhalter für den Select-Button, nicht ändern!

4.2 econ unit+ V2

Die einzelnen Elemente sind:

Address	Platzhalter für Slave-Adresse des Geräts, auf 0 belassen
AddressBase	Startadresse des Adressbereichs (0 oder 1)
Type	Bezeichnung des Geräts (beliebiger Text-String)
TX []	Array mit Sende- Elementen. Jedes Element enthält die Bestandteile zur Bestimmung eines Sende-Frames.
function	Modbus-Funktion (z.B.3 für "Read Holding Register"), unterstützt werden die Modbus-Funktionen 1 ("Read Coils"), 2 ("Read Discrete Inputs"), 3 ("Read Holding Register") und 4 ("Read Input Register")
start	Startadresse (Dezimal)
length	Anzahl Bytes (Dezimal)
RX []	Array mit Empfangselementen. Jedes Element enthält die Bestandteile zur Bestimmung eines Datenpunktes. Die Angaben erscheinen auf der Weboberfläche als auszuwählende Datenpunkte.
register	Startadresse des Values
format	<p>Typ des Values, möglich sind:</p> <p>(HL: High-Byte - Low-Byte; LH: Low-Byte - High-Byte; B... (beliebige Byte-Reihenfolge mit Angabe der Byte-Positionen), hierbei steht 0 für das niederwertigste Byte, die Angabe INT32 B0123 ist gleichbedeutend mit INT32 LH, die Angabe INT32 B3210 ist gleichbedeutend mit INT32 HL)</p> <p>INT8, UINT8, INT16 HL, INT16 LH, UINT16 HL, UINT16 LH, INT32 HL, INT32 LH, UINT32 HL, UINT32 LH, INT32 B0123 (beliebige Byte-Reihenfolge), UINT32 B0123 (beliebige Byte-Reihenfolge), INT48 HL, INT48 LH, UINT48 HL, UINT48 LH, INT48 B012345 (beliebige Byte-Reihenfolge), UINT48 B012345 (beliebige Byte-Reihenfolge), INT64 HL, INT64 LH, UINT64 HL, UINT64 LH, INT64 B01234567 (beliebige Byte-Reihenfolge), UINT64 B01234567 (beliebige Byte-Reihenfolge), FLOAT32 HL, FLOAT32 LH, FLOAT32 B0123 (beliebige Byte-Reihenfolge), FLOAT64 HL, FLOAT64 LH, FLOAT64 B01234567 (beliebige Byte-Reihenfolge), HEX8, HEX16 HL, HEX16 LH, HEX32 HL, HEX32 LH, HEX48 HL, HEX48 LH, HEX64 HL, HEX64 LH</p>
unit	Einheit als Text-String
description	Beschreibung des Datenpunktes als Text-String
select	Platzhalter für den Select-Button, nicht ändern!
LC []	Optionales einelementiges Array mit Sende-Elementen für den Live-Check. Das Array-Element beschreibt den für den Live-Check zu verwendenden Sende-Frame.
function	Modbus-Funktion (z.B.3 für "Read Holding Register")
start	Startadresse (Dezimal)
length	Anzahl Bytes (Dezimal)

4.3 econ4

Die einzelnen Elemente sind:

Address	Platzhalter für Slave-Adresse des Geräts, auf 0 belassen
AddressBase	Startadresse des Adressbereichs (0 oder 1)
Type	Bezeichnung des Geräts (beliebiger Text-String)
TX []	Array mit Sende- Elementen. Jedes Element enthält die Bestandteile zur Bestimmung eines Sende-Frames.
function	Modbus-Funktion (z.B.3 für "Read Holding Register")
start	Startadresse (Dezimal)
length	Anzahl Bytes (Dezimal)
RX []	Array mit Empfangselementen. Jedes Element enthält die Bestandteile zur Bestimmung eines Datenpunktes. Die Angaben erscheinen auf der Weboberfläche als auszuwählende Datenpunkte.
register	Startadresse des Values
format	Typ des Values, möglich sind: (HL: High-Byte - Low-Byte; LH: Low-Byte - High-Byte)
	INT16, UINT16 INT32 HL, INT32 LH, UINT32 HL, UINT32 LH, INT32 B0123, INT32 B2301, INT32 B3210, INT32 B1032, UINT32 B0123, UINT32 B2301, UINT32 B3210, UINT32 B1032, INT64 HL, INT64 LH, UINT64 HL, UINT64 LH, INT64 B01234567, INT64 B67452301, UINT64 B01234567, UINT64 B67452301 FLOAT32 HL, FLOAT32 LH, FLOAT32 B0123, FLOAT32 B2301, FLOAT32 B3210, FLOAT32 B1032 FLOAT64 HL, FLOAT64 B01234567, FLOAT64 B67452301
unit	Einheit als Text-String
description	Beschreibung des Datenpunktes als Text-String
factor	Optionaler Default-Wert für die Faktor-Eigenschaft des Datenpunktes (Dezimal)
select	Platzhalter für den Select-Button, nicht ändern!

5. Fehlercodes allgemein

Error Code	Name	Description
01 (0x01)	Illegal function	The function code received in the request is not an allowable action for the slave.
02 (0x02)	Illegal data address	The data address received by the slave is not an allowable address for the slave.
03 (0x03)	Illegal data value	The value in the request data field is not an allowable value for the slave.
04 (0x04)	Slave device failure	The slave fails to perform a requested action because of an unrecoverable error.
05 (0x05)	Acknowledge	The slave accepts the request but needs a long time to process it.
06 (0x06)	Slave device busy	The slave is busy processing another command. The master must send the request once the slave is available.
07 (0x07)	Negative acknowledgment	The slave cannot perform the programming request sent by the master.
08 (0x08)	Memory parity error	The slave detects a parity error in the memory when attempting to read extended memory.
10 (0x0A)	Gateway path unavailable	The gateway is overloaded or not correctly configured.
11 (0x0B)	Gateway target device failed to respond	The slave is not present on the network.

Fehler Code	Name	Beschreibung
01 (0x01)	Illegal function	Der in der Anfrage empfangene Funktionscode ist kein gültiger Code für den Slave.
02 (0x02)	Illegal data address	Die vom Slave empfangene Datenadresse ist keine gültige Adresse des Slaves.
03 (0x03)	Illegal data value	Der Wert im angeforderten Datenfeld ist kein gültiger Wert für den Slave.
04 (0x04)	Slave device failure	Der Slave führt aufgrund eines nicht behebbaren Fehlers eine angeforderte Anfrage nicht aus.
05 (0x05)	Acknowledge	Der Slave nimmt die Anfrage an, benötigt aber viel Zeit für die Bearbeitung.
06 (0x06)	Slave device busy	Der Slave ist mit der Bearbeitung eines anderen Befehls beschäftigt.
07 (0x07)	Negative acknowledgment	Der Slave kann die vom Master gesendete Programmieranforderung nicht ausführen.
08 (0x08)	Memory parity error	Beim Versuch, den Erweiterungsspeicher zu lesen, erkennt der Slave einen Paritätsfehler im Speicher.
10 (0x0A)	Gateway path unavailable	Das Gateway ist überlastet oder nicht richtig konfiguriert.
11 (0x0B)	Gateway target device failed to respond	Der Slave ist im Netzwerk nicht erreichbar.

6. Detaillierte Erläuterung zur Benennung

Empfohlene Benennung Templates:

Damit die Templates erkannt werden und funktionieren, ist folgende Benennung der Datei, als auch der Geräte-Name (Typenamen) im Template selbst, zu beachten.

Für Templates mit LC-Block, ist das Kürzel LC anzuhängen, so dass Templates mit LC-Block eindeutig erkennbar sind. Sind die Templates (Register) von einem bestimmten Firmware-Stand abhängig, so sollte dieser auch mit angegeben werden.

Um eine durchgängige Struktur/Erkennung/Funktionalität einzuhalten, ist der Name der Datei als Geräte-Name ("Type") im Template einzutragen (ohne modbus_ Voranstellung/Präfix).

Im Dateinamen/Gerätenamen/Typenamen keine Sonderzeichen, Leerzeichen und den "." verwenden, um Fehlinterpretationen zu vermeiden und Funktionalität zu gewährleisten.

modbus_Herstellername (-kürzel)_Gerätename/Typ.json

Info zum LC-Block im Template: dieser wird nur für die unit+ V2 benötigt, und zwar bei Geräten, welche nicht mit dem LiveCheck in der V2 zurechtkommen und dort ein "rotes Kreuz" im LiveCheck liefern.

Damit das Template dann auf allen Geräten (unit+ V1 / unit+ V2 / econ4) funktioniert, sind diese auf folgende FW-Stände upzudaten: V1: 2.20ES / V2: ab 1.5.39 / econ4: 4.0.30.0

Werden bestehende Templates angepasst/geändert/ergänzt, so sollten diese für eine bessere Unterscheidung mit einer Nummer am Ende des Namens versehen werden.

Nachfolgend einige Beispiele:

Dateiname

modbus_cs_fa500.json / modbus_janitza_prodata2_LC.json

Im Template Eintragung "Type"

```
"Address": 0,  
"AddressBase": 0,  
"Type": "cs_fa500",  
"TX":
```

```
"Address": 0,  
"AddressBase": 0,  
"Type": "janitza_prodata2_LC",  
"LC":
```

Es sollten auch im Template immer die Einheiten (ohne Faktor) verwendet werden, welche die Hersteller in der Beschreibung angeben, bzw. im Registerwert stehen (Rohdaten).

Empfohlene Dokumentation:

Handbuch/Manual des Gerätes, sowie die Modbus und Registerbeschreibung (falls separat vorhanden), die json-Datei und die Info, wofür das Template gebraucht und erstellt wurde (unit+ V1(v1)/unit+ V2(v2)/econ4(e4)). Ggf. im Dateinamen und Typenamen dafür die Kürzel v1, v2, e4 mit Einfügen (sollte für Templates gemacht werden, welche nur mit dem jeweiligen Gerät funktionieren -> siehe auch Kap.9). Eine Excelregisterliste mit eventuellen Besonderheiten bzw. was speziell bei dem Gerät zu beachten ist (z.B. kann nur Baudrate 9600, also Baudrate nicht programmierbar).

7. Anbindungsmöglichkeiten unit+ V1/V2, econ4 und Portangaben

Gerät	Modus	Port (default)	Anschluss	Port änderbar (Abfrage/Anbindung)
unit+ V1	Modbus RTU	502	A-/B+	nein
unit+ V2	Modbus TCP-Port intern (inkl. I/O's) und angeschlossene Module, Server aktiv	502	nur intern/direkt	ja
	Modbus TCP-Port / Erweiterungs-Bus (Brückenstecker) und Module direkt	5021	A-/B+	nein
	Modbus TCP-Port / RTU	5022	A`-/B`+	nein
econ4	Modbus TCP-Port	502		ja

8. Unterschiede bei der Byteorder unit+ V1/ unit+ V2/ econ4

Aufgrund der unterschiedlichen verwendeten Versionen der jeweiligen Modbus-Bibliotheken, kommt es vor allem bei der Byteorder (Verwendung des Datatypes beginnend mit **BXXXX**) zu unterschiedlicher Schreibweise und Interpretation. D.h. z.B.: ist in econ4 die Schreibweise **FLOAT32 B2301** erforderlich, so muss in der unit+ V2 **FLOAT32 B1032** verwendet werden. Dementsprechend wird in der Benennung/Typename auch das Kürzel e4 und v2 entsprechend geführt, z.B. (modbus_kelectric_krypton_e10_e4.json), um deutlich zu machen, dass das Template nur im entsprechenden Gerät/Server funktioniert. Die Datatypes der Register und Bytefolge, welche im Template hinterlegt sind, müssen grundsätzlich den Einstellungen im Gerät und den Herstellerangaben entsprechen, da das Template sonst nicht funktioniert.

9. Modbusinfo allgemein (Slaveadressen, Schreibweise)

Bei der Erfassung und Verbindung von Geräten zu einer unit+ oder zu econ4 werden nur die beiden Modbusbetriebsarten *Modbus/TCP* und *Modbus/RTU* verwendet (siehe auch Kap.8). *Modbus/ASCII* findet keine Verwendung.

Anschlusshinweis/Klemmenbeschriftung: bei Modbus ist lediglich die Polarität (+/-) bindend und zu beachten, die Buchstabenzuordnung ist nicht genormt.

Schreibweise der Kommunikationseinstellungen: 19200 8N1 heißt z.B. ausführlich:
19200 baud / 8 bits / keine Parität / 1 Stopbit

Diese Einstellungen müssen an Modbus/RTU angeschlossenen Geräten **alle identisch** sein, auch am Master.

Je nach Gerät sind unterschiedliche Baudraten möglich, Bits sind i.d.R. gleich (also immer 8), bei der Parität sind *none/even/odd* (keine/gerade/ungerade) möglich, bei den Stopbits 1 oder 2.

Die Einstellmöglichkeiten sind der Anleitung/Datenblatt des jeweiligen Herstellers des Gerätes zu entnehmen. Des Weiteren muss jedes Gerät eine andere Slaveadresse/ID/Busadresse besitzen, einschließlich des Masters auf der RTU-Seite.

Ein Modbus-Slave antwortet nur auf Telegramme, die seine eigene Adresse enthalten. Die Adresse 0 ist keine gültige Slaveadresse, sondern wird für Broadcast-Telegramme an alle Slaves verwendet, die nicht beantwortet werden. Die Adressen 248 bis 255 sind reserviert.

Mögliche Adressen: 1-247, wobei max. 32 Teilnehmer am Bus möglich sind.

10. Mitgelieferte Templates

(diese sind schreibgeschützt – bei Bedarf Schreibschutz aufheben)

modbus_basic_format_template.json
modbus_basic_format_template_LC.json

modbus_device_DvModbusIR.json
modbus_device_DvModbusLIR_LC.json
modbus_es_cs_fa5xx.json
modbus_es_cs_iac500_LC.json
modbus_es_cs_va5xx.json
modbus_es_cs_va5xx_p.json
modbus_es_ecsEM64_100.json
modbus_es_ecsEM68MID_100.json
modbus_es_ecsEM72_100.json
modbus_es_ecsEM76MID_100.json
modbus_es_ecsEM113_200.json
modbus_es_ecsEM114MID_200.json
modbus_es_ecsEM226_200.json
modbus_es_ecsEM227MID_200.json
modbus_es_ecsPM66MID_300.json
modbus_es_ecsPM69MID_300.json
modbus_es_sens+.json
modbus_es_sens3.json
modbus_janitza_umg96_pq_pa_LC.json
modbus_janitza_umg96_rm_LC.json

Weitere Templates auf Anfrage bzw. nach Beauftragung.