

# Vitogate 300



[www.modbus.org](http://www.modbus.org)

## Inhalt

Inhalt.....	2
1. Eingetragene Warenzeichen .....	3
2. Copyright .....	3
3. Einleitung.....	4
4. Master/Slave Protokoll.....	4
5. Slave Adresse.....	4
6. MODBUS/TCP .....	4
7. Datenzugriff.....	5
Binäre Register .....	5
Analoge Register.....	5
Unterstützte Funktionscodes .....	6

## 1. Eingetragene Warenzeichen

In diesem Buch werden Warenzeichen und Produktbezeichnungen verschiedener Firmen verwendet. Die folgenden Bezeichnungen sind eingetragene Warenzeichen der jeweiligen Hersteller und werden in diesem Buch nicht gesondert aufgeführt:

- Microsoft, Windows und MS-DOS sind eingetragene Warenzeichen der Microsoft Corporation
- BACnet und ASHRAE sind eingetragene Warenzeichen der American Society of Heating, Refrigerating and Air-Conditioning Engineers, INC. (ASHRAE)
- IBM-PC und IBM-AT sind eingetragene Warenzeichen der International Business Machines Corporation (IBM)
- LONTalk ist eingetragenes Warenzeichen der Echelon, Inc.

## 2. Copyright

©2014 Viessmann Elektronik GmbH

Alle Rechte vorbehalten. Kein Teil dieses Buches darf in irgendeiner Form (Druck, Fotokopie, oder in einem anderen Verfahren) ohne schriftliche Genehmigung der Viessmann Elektronik GmbH reproduziert oder unter Verwendung elektronischer Systeme verarbeitet, vervielfältigt oder verbreitet werden.

### **3. Einleitung**

MODBUS ist ein weltweit verbreitetes Protokoll, welches allen Anwendern offen gelegt ist und von vielen Herstellern unterstützt wird. MODBUS wurde 1979 von der Firma Modicon (Frankreich) entwickelt. Ziel war eine einfache serielle SPS-Kommunikation über die RS232 und RS485 Schnittstelle.

Darauf aufsetzend wurde für den Einsatz in modernen Netzwerken MODBUS/TCP entwickelt. Dieses Protokoll ist heute ein offener Internet-Draft-Standard, der in die für Internet-Standardisierung zuständige Organisation IETF (Internet Engineering Task Force) eingebracht wurde. Seit 2007 ist MODBUS/TCP Teil der Norm IEC 61158.

Die MODBUS Nutzerorganisation veröffentlicht die Spezifikationen, so dass jeder Hersteller und Anwender dieses Protokoll implementieren kann – eine Möglichkeit, die bereits viele führende Hersteller genutzt haben.

Die Übertragung erfolgt als

- ASCII (7 oder 8 Datenbits)
- RTU (Remote Terminal Unit, Zeichen werden binär kodiert)

ASCII-Übertragung ist eher selten und kann einfach ‚mitgelesen‘ werden.

### **4. Master/Slave Protokoll**

Die Kommunikation findet (SPS-typisch) nur im Abfrage-Modus (Polling) statt. Ein Master ist der aktive Teil der Kommunikation, er darf senden. Dieser fragt im Normalfall zyklisch die Slaves ab.

Der Slave ist passiv und darf nur antworten, wenn er von einem Master angesprochen wird.

In seriellen MODBUS Netzwerken gibt es nur einen einzelnen Master.

### **5. Slave Adresse**

Jeder Slave muss per Projektierung / Konfiguration eine eindeutige Adresse bekommen.

Die Slaves werden im Bereich von 1-247 adressiert, 248-255 sind reserviert. Die Adresse 0 ist für Broadcast reserviert und darf nicht verwendet werden

### **6. MODBUS/TCP**

MODBUS über IP verwendet das TCP über den Standard-Port 502.

## 7. Datenzugriff

Die Daten können über einzelne Coils oder Register über eine Nummer angesprochen werden. Die Daten werden je nach Typ in Tabellen gruppiert. Hier wird in binären (Coil) und analogen (Register) Daten unterschieden.

### Binäre Register

- **coils**  
*„discrete output coils“*  
(read-write boolean, 1 Bit)  
Tabelle 1 – 9999
  
- **status**  
*„discrete input coils“*  
(read-only boolean, 1 Bit)  
Tabelle 10001 - 19999

### Analoge Register

- **input registers**  
*analog input register*  
(read-only integer, 16 Bit)  
Tabelle 30001 – 39999
  
- **holding registers**  
*analog output holding register*  
(read-write integer, 16 Bit)  
Tabelle 40001 - 49999

Die Zuordnung, welche Bits oder Register welche Bedeutung haben, kann nur in Form einer Datenpunktliste festgelegt werden. Zusätzliche Informationen wie z.B. der Name der Information, Maßeinheit, Grenzwerte oder Min-/Max-Informationen können nicht festgelegt werden.

Die 16-Bit Daten-Register müssen ggf. skaliert oder umgerechnet werden. Vitogate verwendet die Formate:

- Ganzzahlig vorzeichenlos und vorzeichenbehaftet (u – unsigned, s - signed)
- Faktor 10 (eine Nachkommastelle) (ut, unsigned tenth, st – signed tenth)
- Faktor 100 (zwei Nachkommastellen) (uh, unsigned hundredths, sh - signed hundredths)
- Faktor 1000 (drei Nachkommastellen) (um, unsigned mega, sm - signed mega)

Die Umrechnung muss auf der Master-Seite (beim Client) erfolgen.

## **Unterstützte Funktionscodes**

Zum Auslesen bzw. Schreiben der Daten werden vom Vitogate folgenden MODBUS Funktions-Codes unterstützt:

- 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