

Kemro K2

Modbus Interface

House Automation Systems

Documentation V 1.10

Status: in Arbeit / fertig für Freigabe / freigegeben

Änderungsverzeichnis

Version	Datum	geänderte Seiten	Beschreibung	geändert von
V1.0	23.7.2014		Neu erstellt	adk
V1.1	26.9.2014		+ Änderung der Zugriffe auf Register (read & write multiple registers) + unterstützte Modbus Kommandos + Statusbits für Diagnose	adk

Verfasser: kz/kz

Dokument-Version: V 1.01

Dateiname:

D:\Kemro\Views\adk_KeEnergy.ModBusDev_int\KAppIFrameWork\KeEnergy.ModBus\Doc\ModBusServer\ModbusServer_Documentation.doc

Versionsdatum: 28.10.2014

gedruckt: 28.10.2014 09:22

© KEBA 2012

Änderungen im Sinne der technischen Weiterentwicklung vorbehalten. Angaben erfolgen ohne Gewähr.
Wir wahren unsere Rechte.

A: KEBA AG, Gewerbepark Urfahr, A-4041 Linz, Telefon: +43 732 7090-0, Fax: +43 732 7309-10, E-Mail: keba@keba.com

D: KEBA GmbH Automation, Leonhard-Weiss-Straße 40, D-73037 Göppingen, Telefon: +49 7161 9741-0,
Fax: +49 7161 9741-40, E-Mail: keba@keba.com

US: KEBA Corp., 100 West Big Beaver Road, Troy, MI 48084, US, Telefon: +1 248 526-0561, Fax: +1 248 526-0562,
E-Mail: usa@keba.com

CN: KEBA Ltd., Unit 2107 Air China Plaza, Xiaoyun Road 36 #, Chaoyang District, Beijing, 100027, China,
Telefon: +86 10 8447-5861, Fax: +86 10 8447-5871, E-Mail: china@keba.com

www.keba.com

Inhalt

Änderungsverzeichnis 2

Inhalt..... 3

1 Allgemeines..... 5

 1.1 Kurzbeschreibung 5

 1.2 Schnittstelle 5

 1.3 Protokolle..... 5

 1.3.1 Funktionen 5

2 Modbus Server 6

 2.1 Ablaufdiagramm Registerzugriff 7

 2.2 Zwei zusammengehörige Register 7

3 Register Konfigurationsdatei 8

4 Konfigurationsdatei 9

5 Errorcodes..... 9

 5.1 Statuskommando 10

6 Bibliotheken 10

7 Literaturverzeichnis 10

1 Allgemeines

1.1 Kurzbeschreibung

Der Modbus Server bietet die Möglichkeit über eine Modbus Schnittstelle bestimmte Variablen der KeEnergy Heizungssteuerung zu lesen und zu schreiben. Die Variablen können mittels Parameterdatei angegeben werden. Weiters können in dieser Datei die Zugriffsrechte und der Umrechnungsfaktor einer Variable festgelegt werden.

1.2 Schnittstelle

Folgende Schnittstellen werden unterstützt:

- RS485
- RS232
- Ethernet

1.3 Protokolle

Folgende Protokolle werden unterstützt:

- ModBus/TCP
- ModBus/RTU

1.3.1 Funktionen

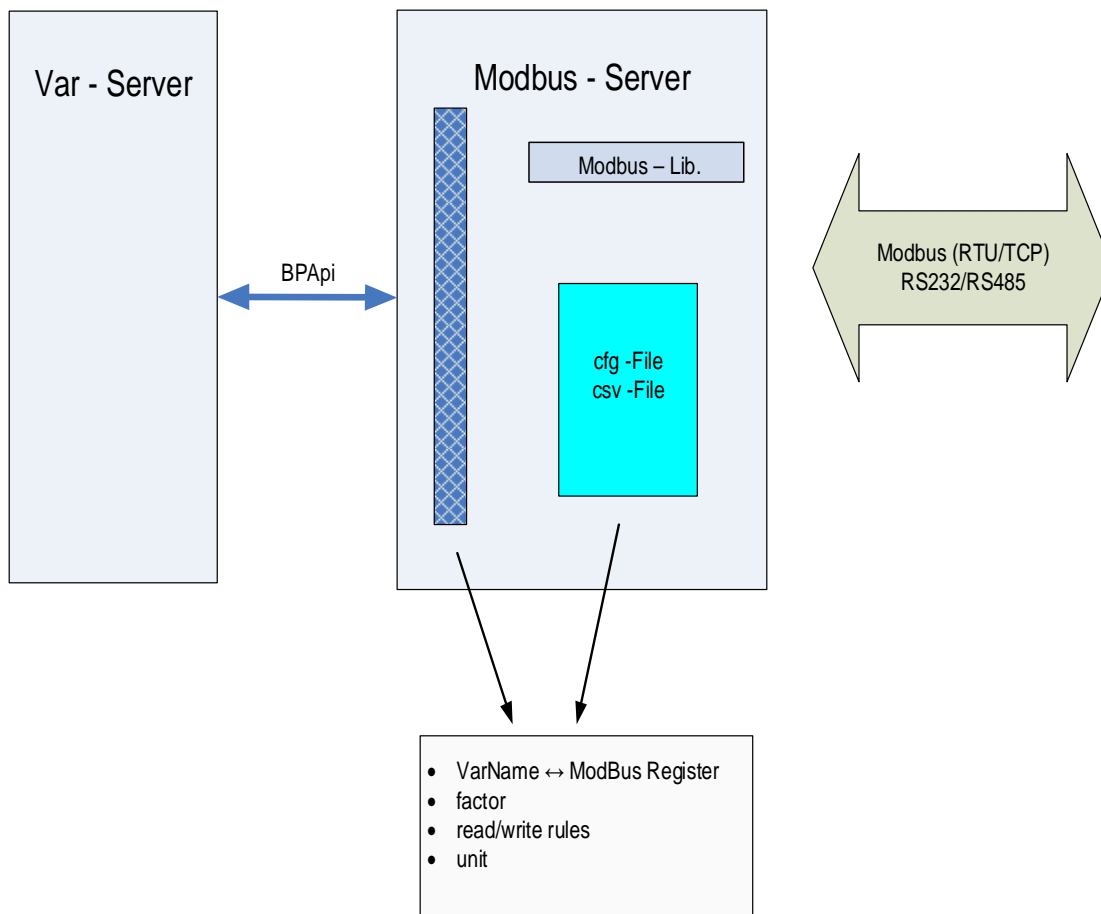
Function code	Name
0x01	Read Coils
0x03	Read Holding Registers
0x06	Write Single Register
0x10	Write Multiple Registers

Die oberhalb erwähnten Funktionen werden vom Modbusserver unterstützt. Alle anderen Funktionsaufrufe werden mit folgender Fehlermeldung MODBUS_EXCEPTION_ILLEGAL_FUNCTION beantwortet.

2 Modbus Server

Der Modbus Server ist eine ausführbare Datei, die unabhängig von der Heizungssteuerung gestartet werden kann. Je nach Konfiguration fungiert der Modbus Server als Slave oder Server und bearbeitet die Anfragen von Master oder Client.

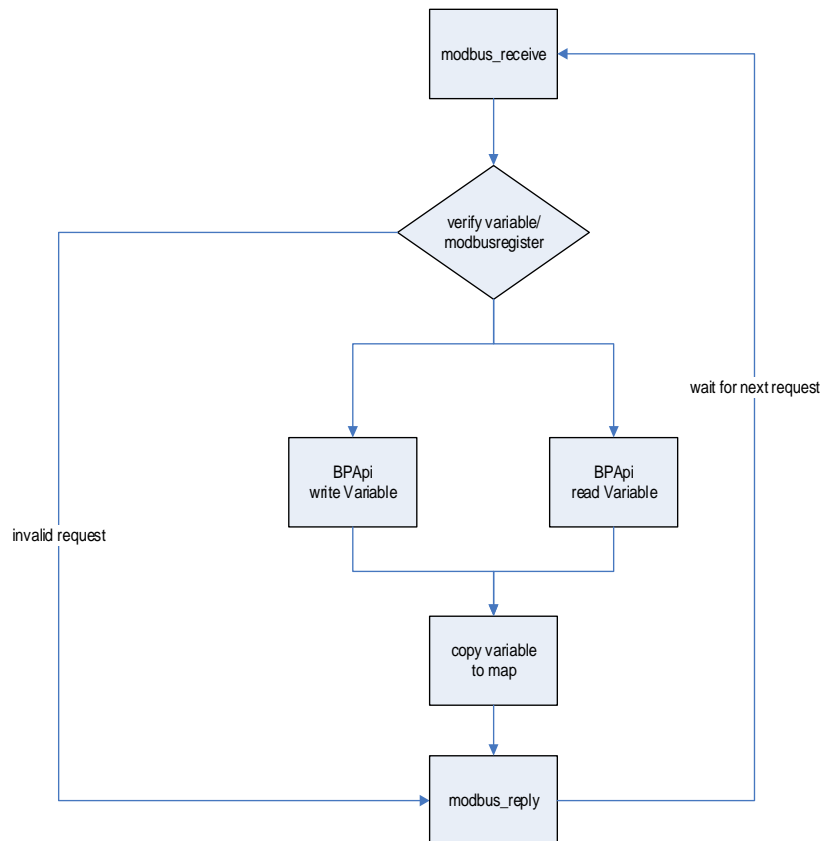
Im Modbus Server wird anhand der [Parameterdatei](#) die Variable überprüft und danach mittels der BPApi-Schnittstelle gelesen beziehungsweise geschrieben.



2.1 Ablaufdiagramm Registerzugriff

Mittels den Befehlen (Read Holding Registers + Write Multiple Registers) kann auch auf mehrere Register mittels einem Befehl zugegriffen werden.

Sollte ein Mehrfachzugriff ungültige Registeradressen oder Zugriffsrechte beinhalten wird der Fehlercode MODBUS_EXCEPTION_ILLEGAL_DATA_ADDRESS returniert und kein Zugriff auf die Register durchgeführt.



Bsp.:

Read Holding Registers (0x03):

Sent Data : 00 00 00 00 00 06 01 03 00 00 00 01 | // read reg 0 (address); qty 1
Received Data : 00 00 00 00 00 05 01 03 02 00 14 | // response two bytes + value

2.2 Zwei zusammengehörige Register

Variablen, die als 32 Bit Werte bearbeitet werden möchten (siehe [Kapitel 3](#)), müssen mit den Kommandos *Read Holding Registers* und *Write Multiple Registers* behandelt werden.

Sollte nur ein Register dieser Variable bearbeitet werden, wird ein Fehler returniert (siehe [Kapitel 5](#)).

3 Register Konfigurationsdatei

Die Register Konfigurationsdatei wird dem Anwender als csv.-Datei zur Verfügung gestellt und muss wie folgt aufgebaut sein.

Mod. add.	Rechte	Factor	Variablenname	Description
0	R	0	APPL.CtrlAppl.sParam.heatpump[0].TempHeatFlow.values.actValue	
1	R	0	APPL.CtrlAppl.sParam.heatpump[0].TempHeatFlow.values.actValue	
2	R	1	APPL.CtrlAppl.sParam.heatpump[0].TempHeatFlow.state.msgId	
3	R/W	100	APPL.CtrlAppl.sParam.heatCircuit[0].param.normalSetTemp	
....				

Variablen, die in der Parameterdatei zwei Register beanspruchen, werden als 32 Bit dem Anforderer zurückgesendet, wobei die niedrigere Adresse die zwei höheren Bytes abbildet.

Variablen die nur ein Register beschreiben werden als 16 Bit Werte bearbeitet. Auf der Client/Slave-Seite muss dieser Wert mittels des definierten Faktors umgerechnet werden. Falls kein Faktor angegeben ist, wird standardmäßig 1 verwendet.

*Bsp.: Raumtemperatur:
Übertragener Wert 2000 (Faktor 100) -> 20°C*

Variablen die Ist-Werte repräsentieren müssen auf Gültigkeit geprüft werden.

*Bsp.: Mod. add. 2
Ist die msgId ungleich 0, ist der gelesene Variablenwert ungültig.*

Aus dem csv. file wird für die Steuerung die Register Konfigurationsdatei (ModBusServerRegisters.cfg) erzeugt. Die Datei liegt unter /etc/default/.

NOTE: Wird die **Steuerung** gestoppt und wieder gestartet wird die Register Konfigurationsdatei neu eingelesen.

Beispiel Register-Konfigurationsdatei:

```
[ModBusReg]
MaximumModBusAddress      = 2

[ModbusReg.Adr:0]
Factor                    = 10
Variablenname             = APPL.CtrlAppl.sParam.heatCircuit[0].tempRoom.values.actValue
WritePermission           = false

[ModbusReg.Adr:1]
Factor                    = 1
Variablenname             =
APPL.CtrlAppl.sParam.heatCircuit[0].humidityRoom.values.actValue
WritePermission           = false
```


4 Konfigurationsdatei

In dieser Datei (ModbusServer.cfg) wird die Schnittstelle definiert. Sie liegt unter /etc/default/ auf der Steuerung. Es wird immer zuerst das RTU Protokoll ausgewertet. Sollten beide Protokolle nicht aktiviert sein startet der Modbus Server nicht.

Beim TCP Protokoll bekommt der Modbusserver die IP-Adresse der AP420.

```
[Modbus.RTU]
ENABLE      = 0
SIO_NAME    = "/dev/ttyS1"
BAUDRATE    = 19200
PARITY      = "N"           // N = none; E = even; O = odd
DATABITS    = 8            // 5, 6, 7, 8
NRSTOPBITS  = 2           // 1 or 2
SERIAL_MODE = 0           // RS232 = 0; RS485 = 1

[Modbus.TCP]
ENABLE      = 1
IP          = "localhost"  // IP-Address of AP420
PORT        = 502
```

5 Errorcodes

Modbus Server

Errorname	Number	Description
eModBusServer_ErrorConfiguration	-1	Interfaceconfiguration failed. Modbus Server is terminated
eModBusServer_Crashed	-2	Creation of modbus map during operation failed after reloading modbus register configuration. Modbus Server is terminated
eModBusServer_Error_UpdateTask	-3	Creation of update task failed. Modbus Server will be restarted.
eModBusServer_Error_ObserverTask	-4	Creation of observer task failed. Modbus Server will be restarted.
eModBusServer_Error_Region	-5	Creation of region failed. Modbus Server will be restarted.

Modbuslibrary

Errorname	Number	Description
MODBUS_EXCEPTION_ILLEGAL_FUNCTION	1	Modbusfunction not supported
MODBUS_EXCEPTION_ILLEGAL_DATA_ADDRESS	2	<ul style="list-style-type: none">- Register address doesn't exist- Multiple access (write or read) across the register addresses' borders- No write permission for register- No valid access to related registers (e.g.: only one register of a related variable is read)
MODBUS_EXCEPTION_ILLEGAL_DATA_VALUE	3	Maximum quantity for registers via command <code>write_multiple_registers</code> exceeds
MODBUS_EXCEPTION_SLAVE_OR_SERVER_FAILURE	4	<ul style="list-style-type: none">- Modbus Server has read invalid register configuration file- Control isn't not running

5.1 Statuskommando

Mittels dem Modbuskommando *Read_Coils* kann der Status des Modbus Servers abgefragt. Der Modbus Server besitzt 2 Statusbits die im ersten Byte (Adresse 0) abgebildet sind.

- Bit0: Status Steuerung: 0 ... Steuerung läuft nicht,
 1 ... Steuerung läuft
- Bit1: Status Modbus Server Initialisierung: 0 ... MB-Server nicht initialisiert
 1 ... MB-Server initialisiert

Werden mehr als zwei Bits gelesen wird der Fehler `MODBUS_EXCEPTION_ILLEGAL_DATA_ADDRESS` returniert.

Bsp.:

Sent Data : 00 00 00 00 00 06 01 01 00 00 00 02 | (Anzahl der *coils*)
Received Data : 00 00 00 00 00 04 01 01 01 03 | (Steuerung läuft, MB-Server ist initialisiert)

6 Bibliotheken

Schnittstelle zum Variablen Server: *BPApi* (KeEnergy.Package_1.12RC1)
Modbusbibliothek: *libmodbus-3.0.6*

7 Literaturverzeichnis

- <http://www.modbus.org/tech.php>
- Modbus_Messaging_Implementation_Guide_V1_0b.pdf
- Modbus_Application_Protocol_V1_1b3.pdf