

ETAtouch Modbus/TCP interface

Version 1.0

ETA Heiztechnik GmbH

February 25, 2014

Contents

1	Introduction	3
2	Configuration	3
2.1	Network setup	3
2.2	Enabling the service	4
2.3	Register assignment	4
2.4	Exporting	6
3	Types of variables	7
3.1	Simple numeric values	7
3.2	Text variables	8
3.3	Time slot variables	8
4	Function codes	9
4.1	FC03 – Read holding registers	9
4.2	FC16 – Write multiple registers	9

1 Introduction

The Modbus/TCP¹ interface on the ETAtouch allows users to read and write values of an ETA boiler. Since Modbus/TCP is a standard interface in building and automation control systems it is well suited to integrate the ETA heating system into building visualizations (see figure 1).

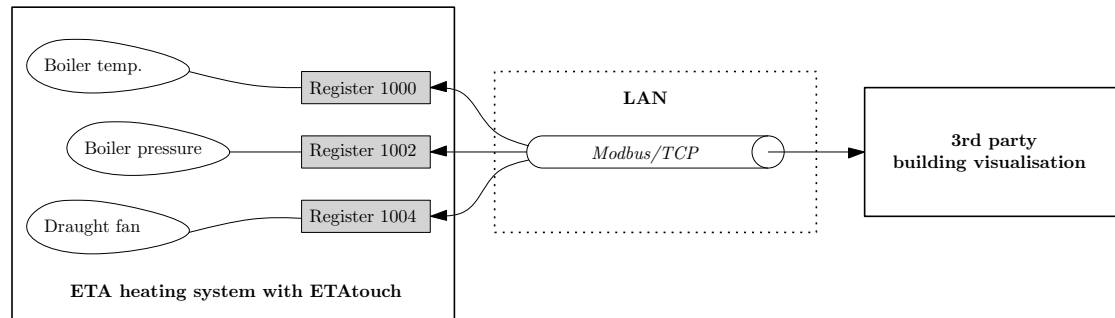


Figure 1: Integration of an ETA heating system into a third-party building visualisation over the LAN.

2 Configuration

Please note that this interface is only available in software version $\geq x.28.0$. All steps covered in this section require an authorisation level of Service which can be enabled by using the authorisation code 135.

2.1 Network setup

In order to be able to communicate with the boiler it must be available in the local area network.

Your Modbus/TCP master device must be connected via a standard Ethernet link to the ETAtouch device. Please do not mess up with Modbus/ASCII and Modbus/RTU!

The network settings dialog allows you to choose two options:

- Let the boiler's network interface to be configured automatically by using a DHCP server (recommended).
- Set the IP address, netmask, gateway and DNS server manually according to your local environment.

¹<http://www.modbus.org>

A successful integration of the boiler into the local area network is mandatory to use the Modbus/TCP interface.

The Modbus/TCP service listens on the default port 502 for incoming connections.

2.2 Enabling the service

The Modbus/TCP service must be enabled in the system configuration dialog (see figure 2).

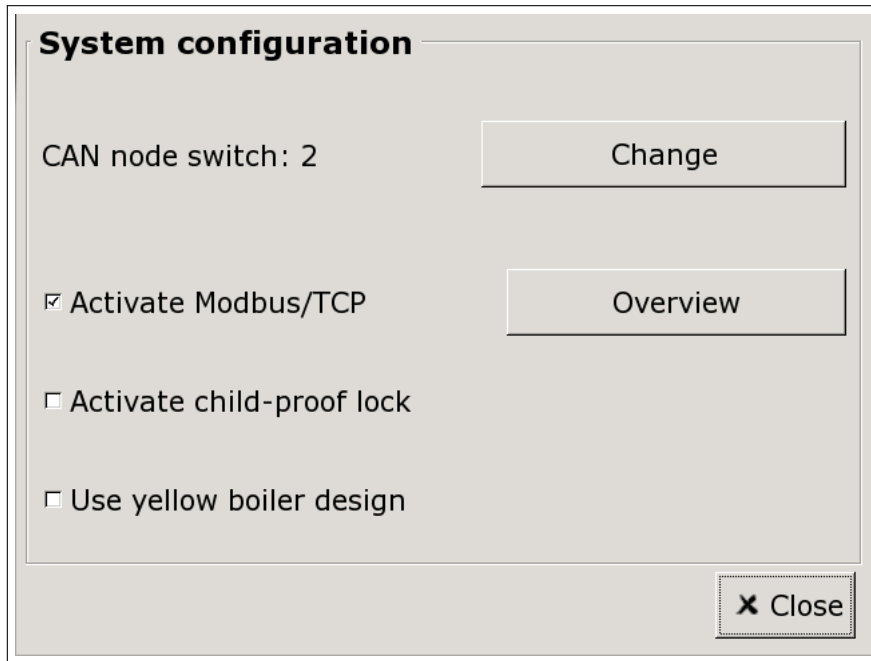


Figure 2: Enable the Modbus/TCP interface on the ETAtouch.

2.3 Register assignment

The assignment of an ETA variable to a unique Modbus register number must be configured by double-tapping on the corresponding line in the text menu (which shows the current parameters) on the ETAtouch device. If the value can be modified, the dialog in figure 3 is opened. In order to switch to the Modbus/TCP register assignment dialog as seen in figure 4 please tap the green arrow button in the left bottom corner of the dialog. If the value is read-only, you will see immediately the Modbus/TCP register assignment dialog.

Please note, that you can only configure parameters which have User permission!

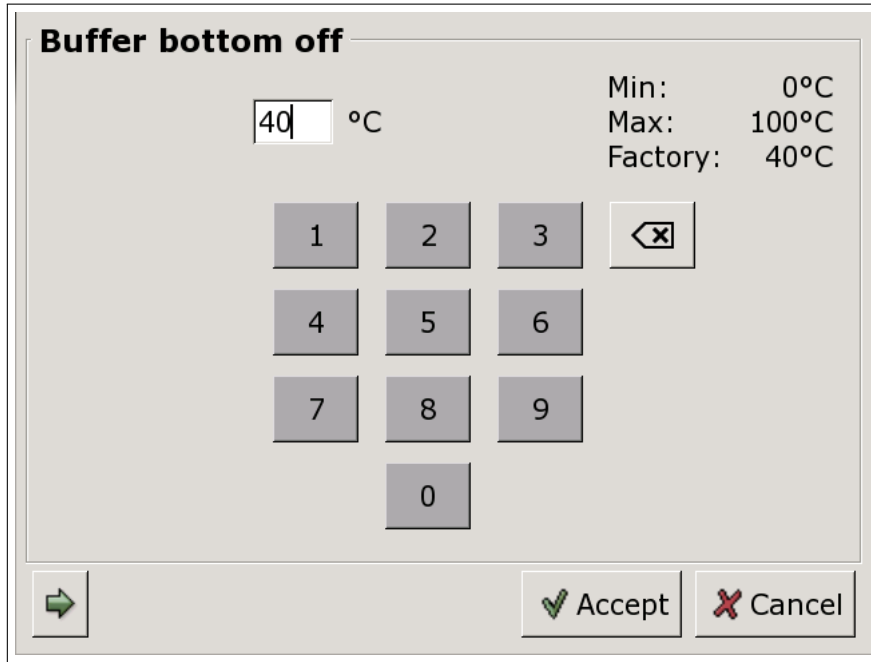


Figure 3: The standard dialog to modify a parameter. In order to change to the Modbus/TCP register assignment dialog tap the green arrow button in the left bottom corner.

In the Modbus/TCP register assignment dialog (see figure 4) you will find next to the configured register some variable meta data:

- the variable's minimum, maximum and default value (if it is writable)
- the variable's scale factor
- the variable's mapping from an integer value to a user readable string (if it is a text variable)

The register number R which can be assigned to a variable must be an even number in the range of $1000 \leq R \leq 65000$. The number of registers which can be assigned is limited to 100.

A single ETA variable will be mapped to two Modbus registers because the variable has a 32-bit data type (see figure 5). As a consequence, every read and write access must comprise at least two registers. The higher 16 bits of the variable's value are stored in register number R , the lower 16 bits are stored in register number $R + 1$ (see figure 5). When you map an ETA variable to a Modbus register number you only need to specify the first register number R , the consecutive register number $R + 1$ will then be available automatically. Due to this mechanism you are allowed to choose only even register numbers R .

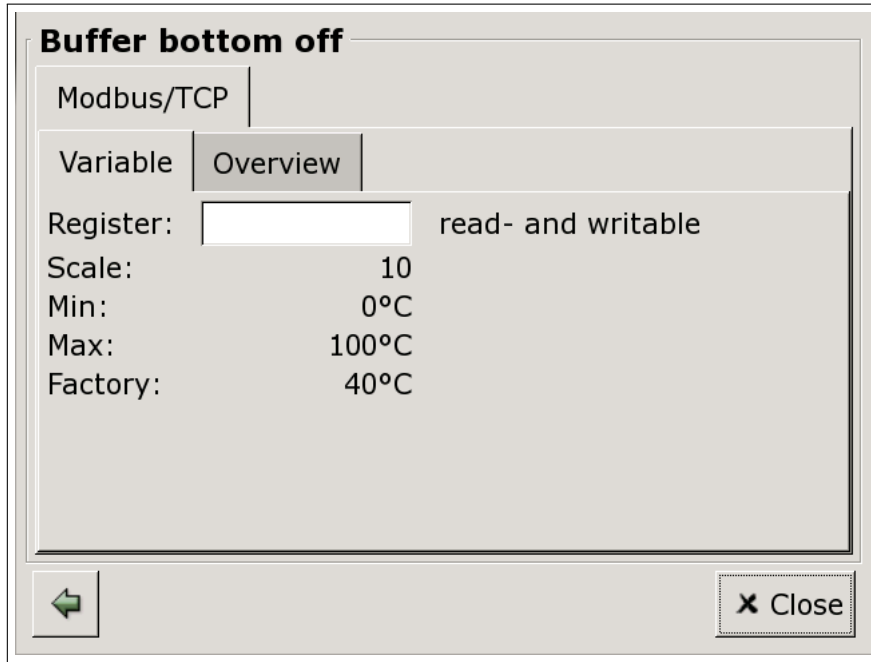


Figure 4: Register configuration screen on the ETAtouch.

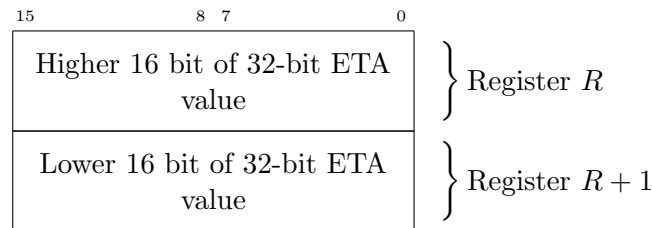


Figure 5: A single ETA variable value is mapped to two Modbus registers.

2.4 Exporting

Once you have configured the Modbus/TCP interface by assigning ETA variables to Modbus registers, you can export this configuration in the dialog *Export protocols* (see figure 6). This can be useful if you want to print you register assignments for instance. The following listing shows an example for an exported configuration file.

```

1 <?xml version="1.0"?>
2 <modbusTcp version="1">
3   <register id="1000">
4     <variable name="GM-C/PE-E_0:_Buffer_-_Buffer_bottom_off" nodeId="112" fubId="
      10251" fktId="0" ioId="0" varId="12212" scale="10" unit="°C" min="0" def="
      400" max="1000" />
5   </register>
6   <register id="1002">
7     <variable name="GM-C/PE-E_0:_Boiler_-_Boiler_pressure" nodeId="112" fubId="10021
      " fktId="0" ioId="0" varId="12180" scale="100" unit="bar" min="" def="" max=
      "" />

```

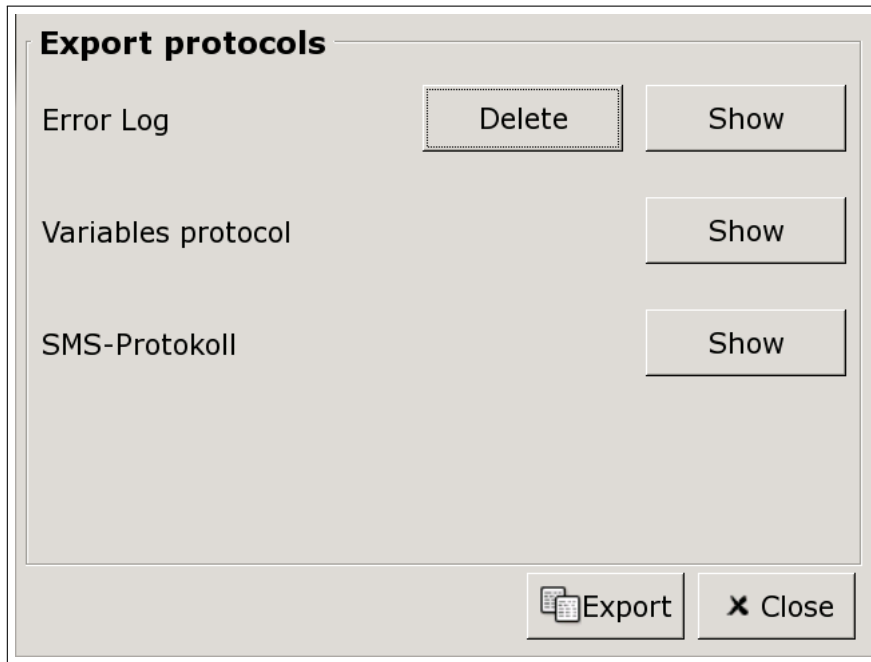


Figure 6: Export the Modbus/TCP configuration to an USB flash drive just by tapping the button *Export*.

```

8   </register>
9   <register id="1004">
10  <variable name="GM-C/PE-E_0:_HK_1_-_-Set-back" nodeId="112" fubId="10101" fktId
    ="0" ioId="0" varId="12107" scale="10" unit="°C" min="0" def="30" max="300
    " />
11  </register>
12 </modbusTcp>

```

3 Types of variables

Depending on the type of the ETA variable to be read or written you must interpret or provide the data in the two corresponding Modbus registers correctly. The following sections will give you an overview of the possible types of variables.

3.1 Simple numeric values

Simple numeric values are provided as integer values with a scale factor. If a variable of this type should be read, the received value must be divided by the scale factor in order to get the real value. If a variable of this type should be written, you must provide the unscaled integral value (see figure 7).

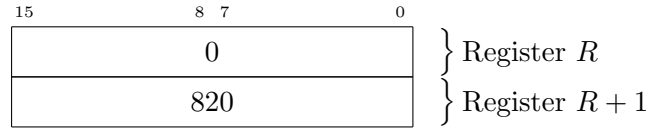


Figure 7: Example of reading/writing a temperature variable with a scale factor of 10. Reading this value means that the variable has a value of 82 centigrade. Writing this value means that you want to set the temperature to 82 centigrade by setting the register to 820.

3.2 Text variables

A text variable is stored as simple integer value which can be represented by a human readable string. The mapping which is available for a text variable can be found in the register assignment dialog. If a text variable should be set to a specific value, the integer value must be provided (see figure 8).

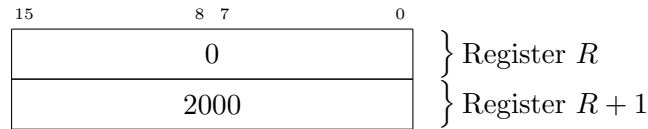


Figure 8: Example of reading a text variable. If the variable represents the boiler's state, then receiving a value of 2000 means the boiler is in state *Switched off*.

3.3 Time slot variables

Three informations are packed in a time slot variable (see figure 9):

- A temperature value (16 bit value)
- The start time of the time slot (8 bit value; in integral multiples of 15 minutes since midnight)
- The end time of the time slot (8 bit value; in integral multiples of 15 minutes since midnight)

Figure 10 shows an example of reading a time slot variable.

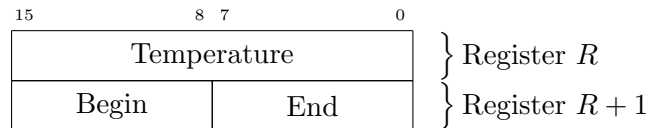


Figure 9: Register contents of a time slot variable.

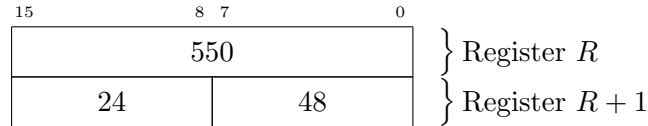


Figure 10: Example of reading a time slot variable. The time slot starts at 6 o'clock in the morning and ends at noon. The temperature during this window has a value of 55 centigrade.

4 Function codes

Because this interface is compliant to the standard Modbus/TCP implementation the details of this protocol are omitted. Following you will only find specific constraints to read/write values from/to the ETA boiler.

4.1 FC03 – Read holding registers

Using this function code you can read the mapped ETA variables. The constraints for this request are

- The start address to be read from must be even.
- The number of registers to be read must be even.

Figure 11 and 12 shows an example of reading two ETA variables, boiler temperature and hot water charging time slot 1 for Monday. The boiler temperature variable is mapped to register 1000 and the time slot variable to register 1002.

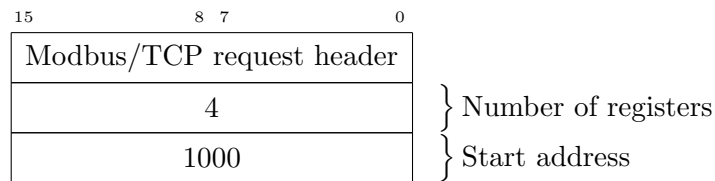


Figure 11: Modbus/TCP request for FC03 to read 4 registers (two ETA values) starting at register 1000.

4.2 FC16 – Write multiple registers

Using this function code you can set mapped ETA variables to specific values. The constraints for this request are

- The start address to be written to must be even.
- The number of registers to be written to must be even.
- Every value to be written must be a 32-bit value, where the higher 16 bits have to be provided in register R and the lower 16 bits in register $R + 1$.

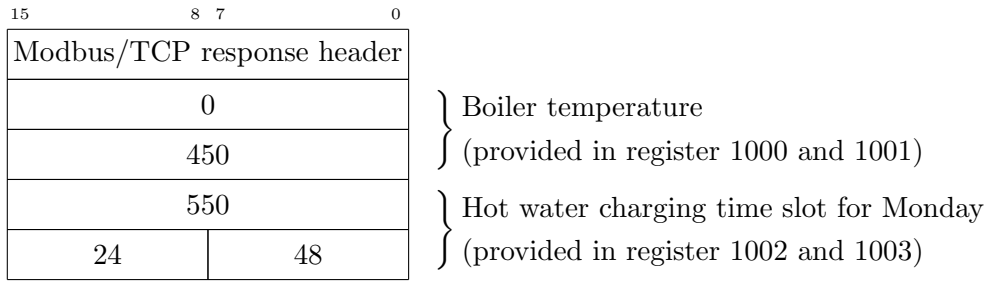


Figure 12: Modbus/TCP response for FC03 to read 4 registers (two ETA values) starting at register 1000.

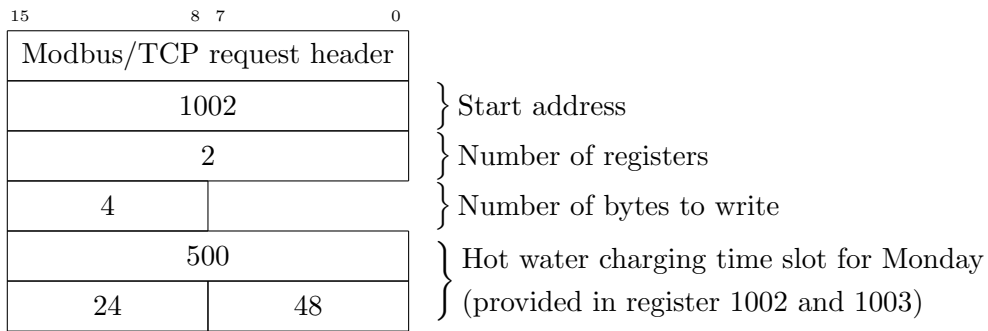


Figure 13: Modbus/TCP request for FC16 to write 2 registers (one ETA value) starting at register 1002.

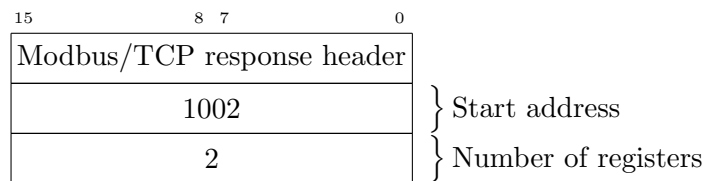


Figure 14: Modbus/TCP response for FC16 to write 2 registers (one ETA value) starting at register 1002.