

# **ETAtouch Modbus/TCP interface**

**Version 1.2**

ETA Heiztechnik GmbH

March 23, 2022

# Contents

<b>1</b>	<b>Introduction</b>	<b>4</b>
<b>2</b>	<b>Prerequisites</b>	<b>4</b>
<b>3</b>	<b>Register assignment</b>	<b>4</b>
3.1	General informations . . . . .	4
3.2	Types of variables . . . . .	5
3.2.1	Fixed point variables (DEFAULT) . . . . .	5
3.2.2	Text variables (TEXT) . . . . .	5
3.2.3	Floating point variables (IEEE-754) . . . . .	6
3.2.4	Time slot variables (TIMESLOT) . . . . .	6
3.2.5	Time point variables (TIMEPOINT) . . . . .	7
<b>4</b>	<b>Supported function codes</b>	<b>7</b>
4.1	FC03 - Read holding registers . . . . .	7
4.2	FC16 - Write multiple registers . . . . .	8
<b>5</b>	<b>Exporting</b>	<b>9</b>

## Changelog

BE-P-Version	Description
$\geq$ x.56.0	Floating point variables added
$\geq$ x.44.0	Time point variables added
x.28.0	Intitial implementation

# 1 Introduction

The Modbus/TCP<sup>1</sup> 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).

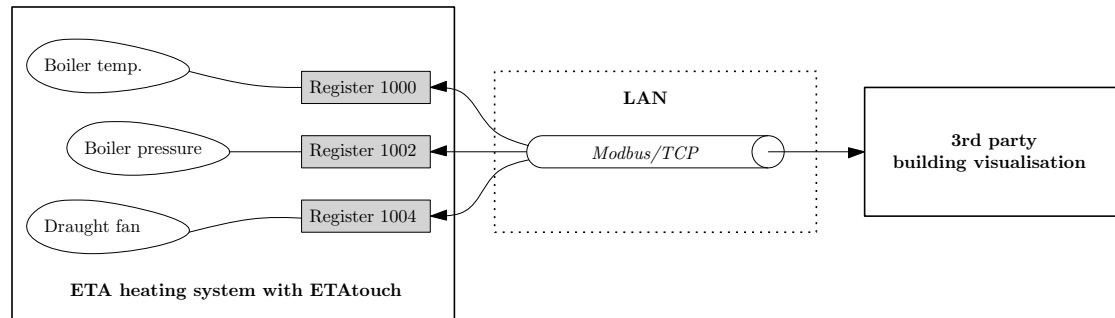


Figure 1: Integration of an ETA heating system into a third-party building visualisation over the LAN.

## 2 Prerequisites

In order to be able to communicate with the boiler it must be connected to your local area network. Please refer to the operating instructions of your boiler on

- how to setup the LAN interface and
- how to start the Modbus/TCP service

*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 Modbus/TCP service will then listen on the default port *502* for incoming connections.

## 3 Register assignment

### 3.1 General informations

The assignment of an ETA variable to a unique Modbus register number must be configured via the text menu. The register number *R* which can be assigned to a variable must be an even number. A single ETA variable will be mapped to two Modbus registers

---

<sup>1</sup><http://www.modbus.org>

because the variable has a 32-bit data type (see figure 2). 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 2). 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  $R$ .

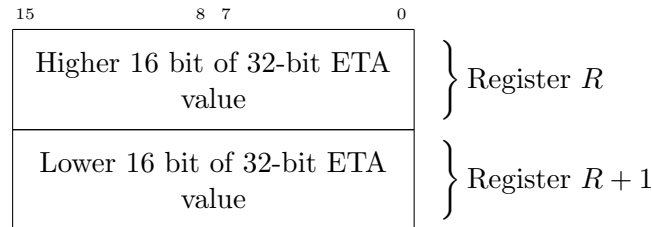


Figure 2: A single ETA variable value is mapped to two Modbus registers.

## 3.2 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.

### 3.2.1 Fixed point variables (DEFAULT)

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 3).

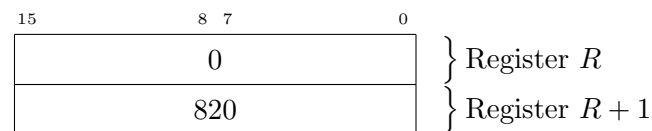


Figure 3: Example of reading/writing a temperature variable with a scale factor of 10. Reading this value means that the variable has a value of 82 centigrade. Writing this value means that you want to set the temperature to 82 centigrade by setting the register to 820.

### 3.2.2 Text variables (TEXT)

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 user interface. If a text variable should be set to a specific value, the integer value must be provided (see figure 4).

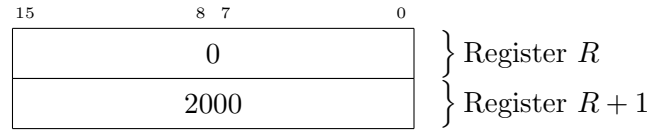


Figure 4: Example of reading a text variable. If the variable represents the boiler's state, then receiving a value of 2000 means the boiler is in state *Switched off*.

### 3.2.3 Floating point variables (IEEE-754)

The variable contains a 32 bit floating point value with single precision according to IEEE-754 floating point format, the byte order is big endian. Figure 6 shows an example of reading a floating point variable.

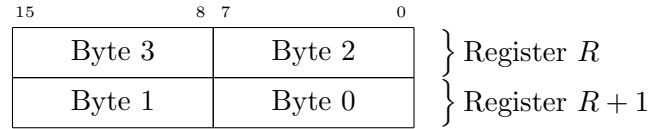


Figure 5: Register contents of a floating point variable.

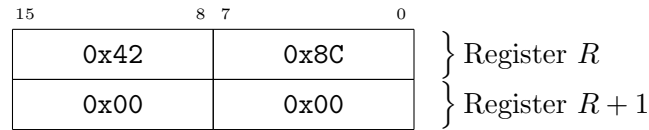


Figure 6: Example of reading a floating point variable. All 4 bytes must be combined to a single 32bit value according to IEEE-754. This results in a value of 70.0.

### 3.2.4 Time slot variables (TIMESLOT)

Three informations are packed in a time slot variable (see figure 7):

- 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 8 shows an example of reading a time slot variable.

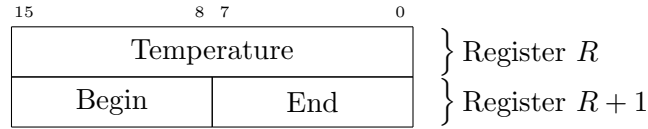


Figure 7: Register contents of a time slot variable.

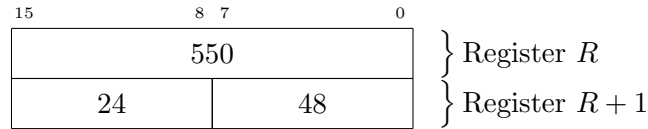


Figure 8: Example of reading a time slot variable. The time slot starts at 6 o'clock in the morning and ends at noon. The temperature during this window has a value of 55 centigrade.

### 3.2.5 Time point variables (TIMEPOINT)

Two informations are packed in a time point variable (see figure 9):

- An arbitrary 16 bit value
- The number of minutes since midnight

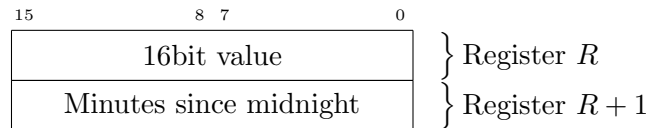


Figure 9: Register contents of a time point variable.

## 4 Supported 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 10 and 11 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.

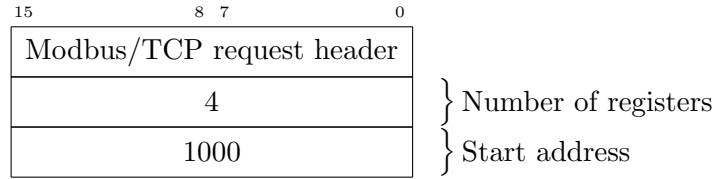


Figure 10: Modbus/TCP request for FC03 to read 4 registers (two ETA values) starting at register 1000.

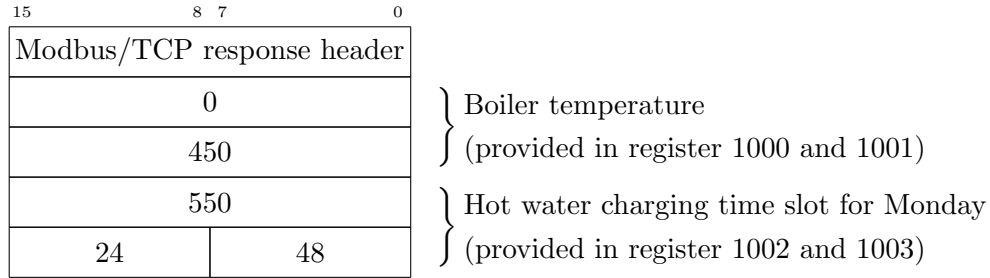


Figure 11: Modbus/TCP response for FC03 to read 4 registers (two ETA values) starting at register 1000.

## 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  $R + 1$ .

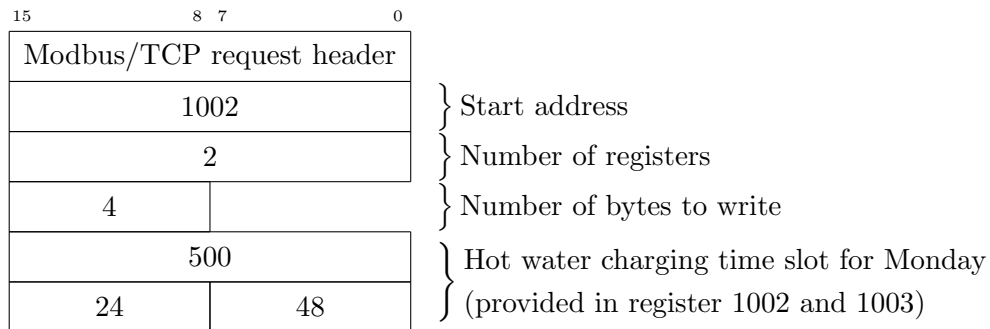


Figure 12: Modbus/TCP request for FC16 to write 2 registers (one ETA value) starting at register 1002.



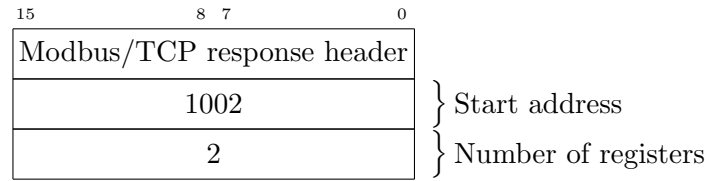


Figure 13: Modbus/TCP response for FC16 to write 2 registers (one ETA value) starting at register 1002.

## 5 Exporting

By exporting the protocols to an USB flash drive, the Modbus/TCP configuration will be exported, too. This can be useful if you want to print your register assignments. You will also find detailed informations on the variables you have configured.