OVUM
WÄRMEPUMPEN

# MIRA Modbus TCP/RTU interface

## Handbook for System Integrators

from software version V1.1.3 onwards

ovum.at

Technical documentation English

Lofererstraße 79
6322 Kirchbichl
Tel: +43 5332 81238 0
E-Mail: office@ovum.at
www.ovum.at



Version: V260630_en

SUPPORT

# TABLE OF CONTENTS


**1. INTRODUCTION ..... 4**
1.1. RELATED DOCUMENTS ..... 4

**2. ARCHITECTURE OF THE COMPLEX ..... 5**
2.1. LSM DISPLAY UNIT ..... 5
2.2. HSM HYDRAULIC UNIT ..... 5
2.3. WPM HEAT PUMP UNIT ..... 6

**3. SLAVE ID / UNIT ID – OVERVIEW ..... 7**

**4. MODBUS TCP ..... 8**
4.1. ENABLING THE MODBUS TCP SERVER ..... 8
4.2. REGISTERS AND FUNCTION CODES USED ..... 9
4.3. BYTE AND WORD ORDER (BIG-ENDIAN) ..... 9

**5. MODBUS RTU ..... 10**

**6. LOGIN FOR MODBUS COMMUNICATION ..... 11**
6.1. ADDRESS FOR REGISTRATION AND STATUS ENQUIRIES ..... 11
6.2. REGISTRATION PROCESS ..... 11
6.3. NOTE ON LOGIN SECURITY ..... 12

**7. MODBUS LEVEL – LICENCES AND FEATURES ..... 13**
7.1. OVERVIEW OF THE LEVELS ..... 13
7.2. PURCHASING AND ENTERING THE ACTIVATION CODE ..... 13
7.3. OVERVIEW OF FEATURES BY LEVEL ..... 14
7.3.1. HEATING CIRCUIT 1 / HEATING CIRCUIT 2 ..... 14
7.3.2. HEATING CIRCUIT 3 / HEATING CIRCUIT 4 ..... 15
7.3.3. HOT WATER ..... 15
7.3.4. HEATING BUFFER ..... 15
7.3.5. COOLING BUFFER ..... 16
7.3.6. SOLAR PANEL SYSTEM ..... 16
7.3.7. HEAT PUMP ..... 16
7.3.8. POWER CONTROL ..... 16

**8. DATA TYPES REFERENCE ..... 17**
8.1. PROCESS VALUES ..... 17
8.1.1. NOTES ON MULTI-WORD PROCESS VALUES ..... 17
8.1.2. NOTES ON BOOL ..... 17
8.2. PARAMETERS ..... 17
8.2.1. P_WAHL – SELECTION PARAMETERS ..... 18
8.2.2. P_INT AND P_FLOAT – NUMERIC PARAMETERS ..... 18
8.2.3. P_TEXT AND P_IP_ADR – TEXT PARAMETERS ..... 19
8.2.4. P_W_PROG_Q – WEEKLY PROGRAMME ..... 19

**9. STRUCTURE OF THE MODBUS LIST ..... 20**
9.1. COLUMN DESCRIPTION ..... 20

**10. STRUCTURE OF THE WORKSHEETS (SHEETS) ..... 21**
10.1. PROCEDURE FOR SYSTEM INTEGRATORS ..... 21

# 1. INTRODUCTION

OVUM heat pumps fitted with the MIRA controller have a built-in Modbus interface. There are two ways to communicate:

- **Modbus TCP:** Via the LAN network interface (RJ45) on the LSM display unit
- **Modbus RTU:** via the wired RS485 interface ('MODBUS 2' connection) directly to the HSM and WPM

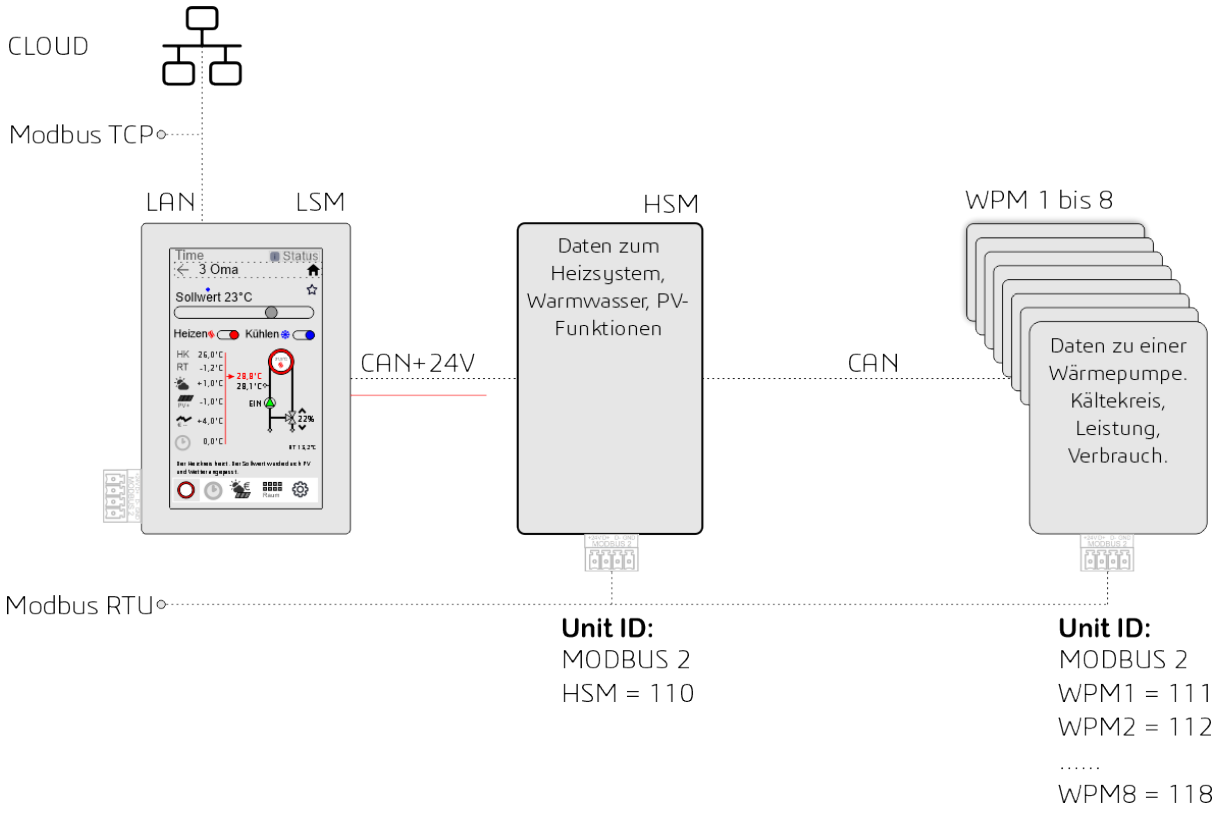
This document explains the system architecture, the slave IDs, the login procedure, the licensing model and the data types used in the Modbus register list. It is intended for system integrators who connect the device to higher-level control systems (BMS, EMS), their own visualisation systems or cloud connections.

## 1.1. RELATED DOCUMENTS

| Document | Description |
|---|---|
| MIRA_ModbusDocumentation_V1.1.x.xlsx | Complete Modbus list (addresses, data types, permissions). Stand-alone supplementary document. From version V1.1.3 onwards |

# 2. ARCHITECTURE OF THE COMPLEX

A heat pump system usually consists of several controllers (circuit boards) that communicate with one another via an internal CAN network. Depending on the chosen communication protocol (Modbus TCP or Modbus RTU), communication takes place with different devices.




## 2.1. LSM DISPLAY UNIT

The LSM display unit consists of a touchscreen and a Linux processor unit. The LSM communicates via the CAN network with the processor boards in the heat pump and/or the hydraulic unit. The LSM also handles all communication via the LAN interface (RJ45).

For Modbus TCP, the LSM collects the data points from all devices via the CAN network and makes them available on the network via the integrated Modbus TCP server. The master controls which device is actually addressed via the slave ID (unit ID / device ID).

With Modbus RTU, communication does not take place via the LSM — the master communicates directly with the HSM and WPM devices. For this reason, with Modbus RTU, a cable must be run to each device.

## 2.2. HSM HYDRAULIC UNIT

The HSM processor board is installed in the hydraulic unit (indoor unit) of the heat pump system (MPlus wall-mounted unit or Cube storage unit).

All controllers relating to the hydraulics and the heating system take place in the HSM, such as hot water production, cooling buffer, heating buffer, heating circuits, SG-Ready, PV control and others. To address a register in the HSM, use Unit ID 110. In the Modbus list, you will find the entry 'HSM' in the 'Slave ID' column.

## 2.3. WPM HEAT PUMP UNIT

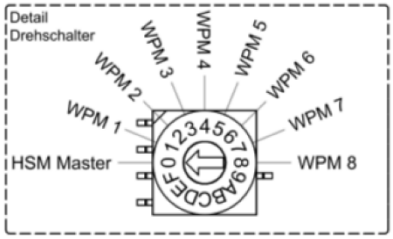
A WPM processor board is installed in every heat pump. The rotary selector switch on the board is used to specify which heat pump is involved (heat pump 1 = rotary selector position 1, heat pumps 2 to 8 accordingly). The heat pumps are addressed via the following unit IDs:

| Heat pump | Unit ID / Device ID |
|---|---|
| WPM 1 | 111 |
| WPM 2 | 112 |
| WPM 3 | 113 |
| WPM 4 | 114 |
| WPM 5 | 115 |
| WPM 6 | 116 |
| WPM 7 | 117 |
| WPM 8 | 118 |

**IMPORTANT**

Each system must be fitted with a heat pump numbered 1 (WPM 1). The serial number of the WPM 1 is used to verify the authorisation keys (see chapter 8).

# 3. SLAVE ID / UNIT ID – OVERVIEW




To find out which slave ID to use for a specific data point, refer to the 'Slave ID' column in the Modbus list. The following table summarises the meaning of the entries.

| Entry in the list | Unit ID | Meaning |
|---|---|---|
| HSM | 110 | This value can be accessed via the hydraulic unit (HSM). |
| WPM | 111 … 118 | This value can be accessed via heat pump modules WPM 1 to WPM 8 – depending on which WPM is to be addressed. |
| WPM, HSM | both | The value can be accessed via both the WPM IDs and the HSM ID. |
| NONE | – | The value is not tied to a specific unit (e.g. global network parameters). |

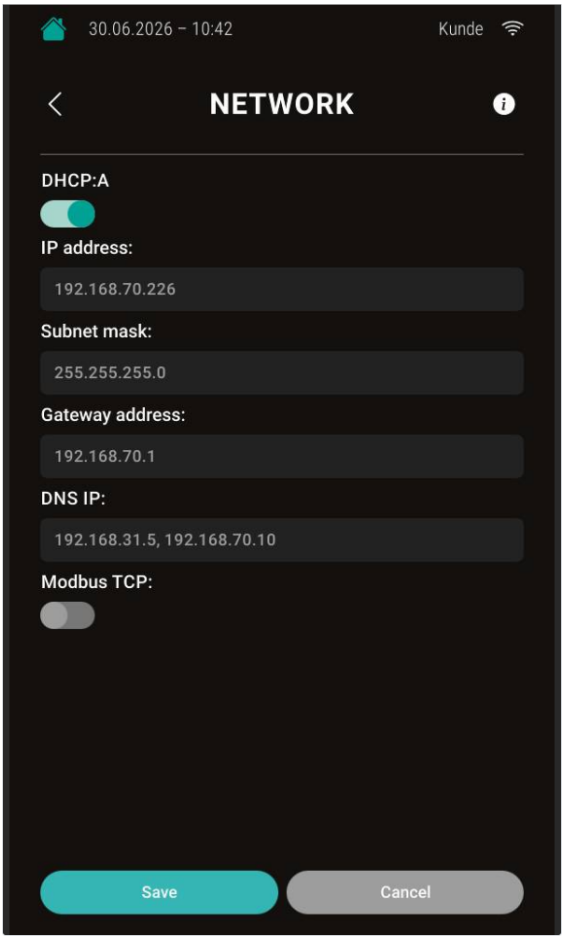
**TIP**

You can check which WPM IDs are actually active in your system by looking at the heat pump's display under*'Settings / Information'*.In a system with just one heat pump, this is usually accessible via Slave ID 111.

# 4. MODBUS TCP


If the system is fitted with an LSM display unit, it can be integrated into a network via the LAN port on the rear of the display. The master communicates with the heat pump via this channel using Modbus TCP on the standard port 502. Several masters can be connected at the same time.

## 4.1. ENABLING THE MODBUS TCP SERVER




The Modbus TCP service is disabled by default on the heat pump and must be enabled manually if required. You can find this setting on the display under*'Settings / Network'*.

This default setting follows the 'security by default' principle enshrined in the EU Radio Equipment Directive (RED Delegated Act 2022/30, which comes into force on 1 August 2025) and in the Cyber Resilience Act (Regulation (EU) 2024/2847): Network interfaces that are not strictly necessary for the system's intended operation remain disabled by default in order to minimise the system's attack surface.

**NOTICE**

**Data security**

Due to the nature of the protocol, Modbus TCP does not support either authentication or encryption. It should therefore only be enabled on a permanent basis in segmented networks or behind a firewall.

## 4.2. REGISTERS AND FUNCTION CODES USED

All values are represented as holding registers (range 4xxxx). Coils and discrete inputs are not used. The following Modbus function codes are used:

| FC | Description | Usage |
|---|---|---|
| 03 | Read Holding Registers | Reading from one or more registers (including across multiple registers, e.g. float, s32). |
| 06 | Write to a Single Register | Writing to a single 16-bit register (e.g. P_WAHL, P_INT, bool). |
| 16 | Write to Multiple Registers | Writing multi-word values in a single operation (float, s32, P_FLOAT, login code, P_TEXT, P_IP_ADR). |

**NOTE**

Multi-word values (e.g. float or s32) must be transferred using FC16 in a single write operation. If only a partial register is written to, the recipient may discard the value or observe an inconsistent state in the meantime.

## 4.3. BYTE AND WORD ORDER (BIG-ENDIAN)

All values that occupy more than one register are stored in big-endian format. The lowest-order register contains the most significant bytes, and the next register contains the least significant bytes. Big-endian order (most significant byte first, least significant byte last) also applies within a register.

Example of a 32-bit value (e.g. float or s32) stored from address N onwards:

| Index | Contents | Bytes |
|---|---|---|
| N | High-Word | Byte 3 (MSB), Byte 2 |
| N+1 | Low-Word | Byte 1, Byte 0 (LSB) |

A specific numerical example — the 32-bit value 0x764AA63A (equivalent to 1,984,603,706 in decimal) is stored as follows:

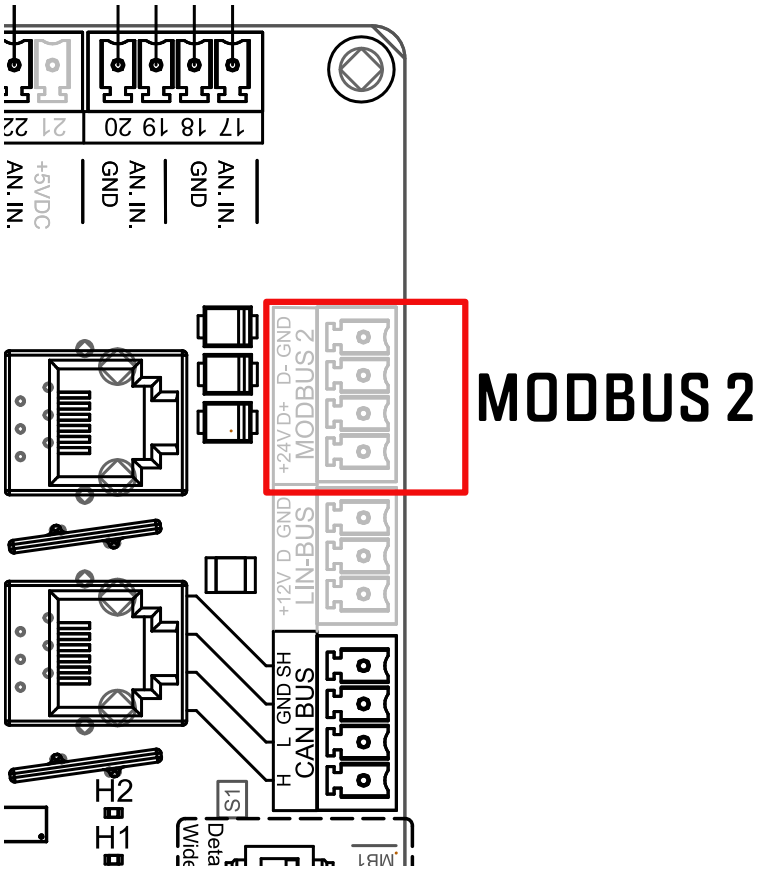
| Index | Hex | Decimal |
|---|---|---|
| N (high-word) | 0x764A | 30.282 |
| N+1 (low-word) | 0xA63A | 42.554 |

**TIP**

Some Modbus clients expect the word order to be reversed ("word swap"). If the 32-bit values you send are not accepted by your client (login fails, float values appear nonsensical), check your client's endianness setting and set it to big-endian (AB CD).

# 5. MODBUS RTU

Communication via the Modbus RTU wired interface takes place directly – without the involvement of the LSM display unit – with the individual processor boards in the hydraulic unit and the heat pump module. To do this, use the MODBUS 2 connection on each device.




**IMPORTANT**

**Cabling**

As the master communicates with the devices individually, the RS-485 bus cable must be connected to each device to be addressed (HSM and/or WPM). Use the 'MODBUS 2' connection on the relevant device.

The slave IDs are the same as for Modbus TCP (HSM = 110, WPM 1 … 8 = 111 … 118). The registration process (Chapter 6) is also identical – however, it must be carried out individually on each device being addressed.

# 6. LOGIN FOR MODBUS COMMUNICATION

In order for addresses to be accessed, a 32-bit login code (big-endian) must be written to address 101 of the relevant unit. If this code matches the Modbus login code stored under*'Settings / Network'*or in the wizard, the remaining registers are enabled. The authorisation status can be queried at any time via address 100 (1 = code accepted).

If a unit is operated without an LSM (display), you can log in to the heat pump via Modbus RTU using the code "287454020".

**NOTE**
Once the LSM has been connected to the WPM or HSM devices, the login code in those devices is permanently replaced by that of the LSM.

## 6.1. ADDRESS FOR REGISTRATION AND STATUS ENQUIRIES

| Address | Index | Unit / Slave ID | Name | Type | Access |
|---|---|---|---|---|---|
| 101 | 102 | HSM, WPM | Modbus login code | s32 | W |
| 100 | 101 | HSM, WPM | Login status (1 = Login successful) | bool | RO |

**NOTICE**
The login code is stored as an s32 value starting at address 101 (register 102), thereby occupying two 16-bit registers (addresses 101 and 102, registers 102 and 103). Big-endian: the higher-order word is located at the lower address.

## 6.2. REGISTRATION PROCESS

The login code must be transferred to both registers in a single write operation (FC16, Write Multiple Registers). Transferring the code to the registers separately will result in the login being rejected.

Write sequence using the registration code 0x764AA63A as an example:

```
FC16 Write Multiple Registers
    Unit ID          : 110   (or the WPM ID to be addressed 111 ... 118)
    Starting address : 101   (= Register 102)
    Quantity         : 2     (or: Number of registers)
    Data             : 0x764A 0xA63A ; High Word before Low Word


FC03 Read Holding Registers ; Check status
    Unit ID          : 110
    Starting address : 100   (= Register 101)
    Quantity         : 1
    Expected Value   : 1     ; 1 = Login successful
```

**IMPORTANT**

**Significance of the login code — Confirmation of familiarity with the documentation**

In order for a user to be able to send the login code to Register 101 correctly in the first place, they must have read this documentation and understood the addressing, byte order and restrictions on parameter data types.

**By successfully submitting or deactivating the login code, the user confirms** in a binding manner that they have read the documentation and, in particular, have taken note of the controller for the write behaviour of parameter types (P_...) — see chapter 8.2.

The LOGIN code can be deactivated by setting it to '1' on the heat pump's display.

**NOTE**

**Synchronisation between the LSM and the processor boards**

The login code set under*'Settings / Network'*(LSM parameter N2004) is synchronised via the CAN network to the processor boards (HSM, WPM, parameter S034) and stored locally in a register there. When communicating via Modbus RTU with a single device, the login code must therefore also be sent to its register 101.

## 6.3. NOTE ON LOGIN SECURITY

The login is not a security feature that keeps skilled attackers at bay. It merely ensures that the user has read the documentation and configured the byte order of their application correctly. It is insufficient as a protective mechanism for two reasons:

– **Brute force:** In principle, the 32-bit key can be systematically tested by trying every possible combination. The device does not lock out users after a certain number of failed attempts. A response time of 1 second has been defined as rate limiting. There is no two-factor authentication.

– **Interception of communications:** Modbus TCP transmits data in plain text. Anyone with access to the network (man-in-the-middle attacks, Wi-Fi sniffing, compromised routers) can intercept the key in plain text.

You should therefore operate the system exclusively within a segregated network segment or secure access via a VPN tunnel.

# 7. MODBUS LEVEL — LICENCES AND FEATURES

The number of data points accessible via Modbus depends on the enabled access level. By default, the authorised data points are limited to non-critical values (level 'Start'). This default setting once again follows the 'security by default' principle in accordance with the EU Radio Equipment Directive (RED Delegated Act 2022/30) and the Cyber Resilience Act (Regulation (EU) 2024/2847).

To unlock additional data points, an authorisation key is required, which can be purchased individually for a specific heat pump (serial number). Only one authorisation key is required per system and level to access all heat pumps.

## 7.1. OVERVIEW OF THE LEVELS

| Level | Description | Note |
|---|---|---|
| Start Level | Generally available | Always accessible without the need for an additional licence. This is the default setting as supplied. |
| Level Plus | User Level B | Extended scope for end users / plant operators. Activation code required. |
| Level BMS | Building Management System | Extended functionality for controlling the heat pump via a building management system. Activation code required. |

## 7.2. PURCHASING AND ENTERING THE ACTIVATION CODE

To generate the activation code, you will need the serial number of heat pump 1 (WPM 1). You can find this on the display under*'Settings / Activation codes / Serial:'*.

SNR of the WPM (1)

Please send an order to an authorisation level (see 7.3) together with the serial number to your OVUM partner.

You will then receive an alphanumeric activation code (example: "1S12X3T68512KZMCKX4P9Z8Z3M85"). Enter this on the display under*"Settings / Activation codes"*.This will activate the additional registers and functions.

**NOTE**

**Managing levels at runtime**

Levels are managed by the LSM display unit in conjunction with the serial number of the WPM 1. Once a level has been enabled, the LSM transmits the runtime information to all processor boards. The boards store this information in volatile memory. Following a restart of a processor board, the connection to the LSM must therefore be re-established before the Modbus feature bits are enabled again on that board.

## 7.3. OVERVIEW OF FEATURES BY LEVEL

The tables below provide an overview of the additional functions and data points available at each level. For a detailed description of the data points, including type, address and register, please refer to the Modbus list.

The column logic in the tables below is additive: whatever is included in the 'Start' level is also available in the 'Plus' and 'BMS' levels. It is also possible to unlock several levels simultaneously.

### 7.3.1. HEATING CIRCUIT 1 / HEATING CIRCUIT 2

| Group | Start Level | Level Plus | Level BMS |
|---|---|---|---|
| **Heating circuit 1**<br>**Heating circuit 2** | Heating circuit type (RO)<br>PV increase setpoint (R/W)<br>PV reduction setpoint (R/W)<br>Operating mode (R/W)<br>Room setpoint for heating (R/W)<br>Current room temperature<br>Current HK setpoint (RO)<br>Current HK actual value (RO) | Current room actual value (RO)<br>Room actual value vs. target (from 2027)<br>Room setpoint for cooling (R/W)<br>Holiday details (R/W)<br>Holiday VLSoll HZ (R/W)<br>Holiday VLSoll KUE (R/W)<br>Setpoint mode (R/W)<br>Fixed-setpoint heating (R/W)<br>Fixed Setpoint Cooling (R/W)<br>Heating limit (R/W) | – |

## 7.3.2. HEATING CIRCUIT 3 / HEATING CIRCUIT 4

| Group | Start Level | Level Plus | Level BMS |
|---|---|---|---|
| **Heating circuit 3**<br>**Heating circuit 4** | – | Heating circuit type (RO)<br>PV increase (R/W)<br>PV reduction (R/W)<br>Operating mode (R/W)<br>Room setpoint for heating (R/W)<br>Current HK setpoint (RO)<br>Current HK actual value (RO)<br>Current room actual value (RO)<br>Room setpoint for cooling (R/W)<br>Holiday details (R/W)<br>Holiday VLSoll HZ (R/W)<br>Holiday VLSoll KUE (R/W)<br>Setpoint mode (R/W)<br>Fixed-setpoint heating (R/W)<br>Fixed Setpoint Cooling (R/W)<br>Heating limit (R/W) | – |

## 7.3.3. HOT WATER

| Group | Start Level | Level Plus | Level BMS |
|---|---|---|---|
| **Hot water** | Installed (RO)<br>Main switch (R/W)<br>WW setpoint (R/W)<br>PV setpoint (R/W)<br>Actual value: Upper memory (RO)<br>Actual value: Lower memory (RO)<br>Current setpoint (RO | Fuel pump setpoint FWS (R/W)<br>FWS (RO) dispensing status<br>Request status (RO)<br>Holiday to follow (RW)<br>Circulation status (RO)<br>Actual circulation value (RO) | – |

## 7.3.4. HEATING BUFFER

| Group | Start Level | Level Plus | Level BMS |
|---|---|---|---|
| **Heating buffer** | Installed (RO)<br>PV setpoint (R/W)<br>Current setpoint (RO)<br>Actual value: Upper memory (RO)<br>Actual value: Lower memory (RO) | Heating/Cooling Status (RO)<br>Charge status (RO) | – |

### 7.3.5. COOLING BUFFER

| Group | Start Level | Level Plus | Level BMS |
|---|---|---|---|
| **Cooling buffer** | – | Installed (RO)<br>Current setpoint (RO)<br>Actual value, memory (RO)<br>Charge status (RO) | – |

### 7.3.6. SOLAR PANEL SYSTEM

| Group | Start Level | Level Plus | Level BMS |
|---|---|---|---|
| **Solar panel system** | PV_Status_Source (R/W)<br>PV_Data_Source (R/W)<br>PV_Status_Default (R/W)<br>SOC_Target (R/W)<br>NETZ_Default (R/W)<br>PV Production_Target (R/W) | PV status WP WW<br>PV Status WP HZ<br>PV status 2ST WW<br>PV status 2ST HZ | – |

### 7.3.7. HEAT PUMP

| Group | Start Level | Level Plus | Level BMS |
|---|---|---|---|
| **Heat pump** | Heating capacity (RO)<br>Requirement % (RO)<br>Power consumption (RO)<br>WP status (RO)<br>WP Admission (RO)<br>Withdrawal from the WP (RO)<br>ON-TIME (RO) | Serial number (RO)<br>RPS (RO)<br>Active WWAnf<br>Active HZStart<br>Active KueAnf | – |

### 7.3.8. POWER CONTROL

| Group | Start Level | Level Plus | Level BMS |
|---|---|---|---|
| **Power control** | Outside temperature (RO) | – | Performance target per work package (R/W)<br>Temperature setpoint control per heat pump (R/W)<br>Central performance target (R/W)<br>Central temperature setpoint via HK1 (R/W)<br>Comprehensive audit data (RO)<br>Charging circuit status (RO)<br>Fault code per heat pump (RO)<br>Requirements for WW/HZ/KUE (RO) |

# 8. DATA TYPES REFERENCE

The Modbus list uses two groups of data types: process values (measured values and states, which are normally read-only) and parameters (settings with value lists, min/max limits or text). The data type for each address is shown in the 'Type' column of the Modbus list.

## 8.1. PROCESS VALUES

| Type | Index | Range of values | Description |
|---|---|---|---|
| bool | 1 | 0 / not equal to 0 | Boolean value. 0 = false; any other value = true. |
| s16 | 1 | -32,768 … +32,767 | Signed 16-bit value (two's complement). |
| U16 | 1 | 0 … 65.535 | Unsigned 16-bit value. |
| s32 | 2 | -2,147,483,648 … +2,147,483,647 | Signed 32-bit value. Two registers, big-endian. |
| u32 | 2 | 0 … 4.294.967.295 | Unsigned 32-bit value. Two registers, big-endian. |
| s64 | 4 | $-9.2\times10^{18}$ … $+9.2\times10^{18}$ | Signed 64-bit value. Four registers, big-endian. |
| u64 | 4 | 0 … $1.8\times10^{19}$ | Unsigned 64-bit value. Four registers, big-endian. |
| float | 2 | IEEE 754, single precision | 32-bit floating-point in accordance with IEEE 754. Two registers, big-endian. |

### 8.1.1. NOTES ON MULTI-WORD PROCESS VALUES

Values of type s32, u32, s64, u64 and float occupy several registers. They must be read in one go (FC03 with the appropriate number of registers) so that the bytes can be assembled consistently. Example: float (2 registers) from address N:

```
Register N     = 0x4248     ; High Word
Register N+1   = 0xF5C3     ; Low Word
Combined Value = 0x4248F5C3 ; IEEE 754 -> 50.24
```

### 8.1.2. NOTES ON BOOL

When reading, any value other than 0 may be interpreted as true. When writing, it is recommended to use 0 for false and 1 for true. Other values are technically permitted, but may be treated more restrictively in future versions.

## 8.2. PARAMETERS

Parameters are configurable settings. They have their own type prefix (P_), which identifies the type of input value. The permissible values, limits or units are listed in the 'Type Information' column of the Modbus list.

**IMPORTANT**

**IMPORTANT — Parameters are stored in flash memory**

All data types with the prefix P_ (P_WAHL, P_INT, P_FLOAT, P_TEXT, P_IP_ADR, P_W_PROG_Q) are persistent parameters and are written to the controller's flash memory every time a write operation takes place. Flash memory has a limited number of write cycles.

**Cyclical or frequent overwriting of these parameters will damage the heat pump's flash memory and result in an irreparable fault in the controller.**

P_-types must only be written when there is an actual change in value — i.e. once per setting operation by the user or when changing operating modes. Never in a loop, a polling loop or for regular updates. For values written cyclically (e.g. setpoint values from an EMS), use the non-persistent process values (bool, s16, u16, s32, u32, float).

| Type | Index | Description |
|---|---|---|
| P_ELECTION | 1 | Selection parameters. A u16 value representing an entry from an enumeration. Possible values are listed in the 'Type Information' column. |
| P_INT | 1 | Integer parameters. An s16 value with minimum and maximum limits and a unit (e.g. °C, K, W). Values outside these limits are rejected by the device. |
| P_FLOAT | 2 | Floating-point parameters. IEEE 754 float (32-bit) using two registers, big-endian. The minimum and maximum limits and the unit are specified in 'Type Information'. |
| P_TEXT | 10 | Text parameters. Up to 20 ASCII characters, 0-terminated, stored in 10 consecutive registers (2 characters per register, high byte first). |
| P_IP_ADR | 10 | IP address as a text field. Layout identical to P_TEXT (20 ASCII characters, null-terminated). |
| P_W_PROG_Q | 84 | Weekly programme. A complex parameter comprising 84 registers, representing a complete timetable (e.g. quiet hours). Format available on request. |

### 8.2.1. P_WAHL – SELECTION PARAMETERS

For P_WAHL, the 'Type Information' column contains a list in square brackets of all possible values and their meanings. Example for the parameter HK1_MODE:

```
[
    0 = HK_OFF,
    1 = HK_AUTOMATIC,
    2 = HK_WINTER,
    3 = HK_SUMMER
]
```

The numeric index (e.g. 2 for HK_WINTER) is written as u16 using FC06.

### 8.2.2. P_INT AND P_FLOAT – NUMERIC PARAMETERS

The 'Type Information' column contains the unit as well as the minimum and maximum limits:

```
Unit: "°C", Decimal places: 1,  Min: -50,    Max: 50    ; Example P_INT
Unit: "kW", Decimal places: 1,  Min: 0,      Max: 1000  ; Example P_FLOAT
```

When writing to a P_FLOAT, both registers must be transferred together using FC16. For P_INT, FC06 is sufficient. Values outside the specified limits will be rejected by the device — recommended procedure: read back the value after writing and check it.

A float value resulting from a calculation may 'vary cyclically' in its final decimal places. Round the value before sending it to the MIRA Modbus value to prevent cyclical writing.

### 8.2.3. P_TEXT AND P_IP_ADR – TEXT PARAMETERS

Texts are stored in 10 consecutive registers. Each register contains two ASCII characters, with the first character in the high byte. Strings are null-terminated; unused positions should also be filled with 0x00 when writing.

Example of the text "192.168.1.10":

```
Register N     = 0x3139    ; '1'  '9'
Register N+1   = 0x322E    ; '2'  '.'
Register N+2   = 0x3136    ; '1'  '6'
Register N+3   = 0x382E    ; '8'  '.'
Register N+4   = 0x312E    ; '1'  '.'
Register N+5   = 0x3130    ; '1'  '0'
Register N+6   = 0x0000    ; Terminator + Padding
Register N+7   = 0x0000
Register N+8   = 0x0000
Register N+9   = 0x0000
```

### 8.2.4. P_W_PROG_Q – WEEKLY PROGRAMME

This data type occupies 84 registers and represents a complete weekly programme (e.g. the heat pump's silent periods). This parameter is currently unavailable.

# 9. STRUCTURE OF THE MODBUS LIST

The individual values (registers, addresses, types) are listed in the attached Modbus list. Each sheet in the list has the same column structure. The following table explains the meaning of the individual columns.

## 9.1. COLUMN DESCRIPTION

| Column | Contents | Explanation |
|---|---|---|
| Address | Numerical address | Modbus address (zero-based), as transmitted on the bus. |
| Index | Registration number | Register name (1-based), as displayed in many Modbus clients. Register = address + 1. |
| Index | Number of registers | Number of registers occupied by the value. s32 / u32 / float / P_FLOAT → 2; s64 / u64 → 4; P_TEXT / P_IP_ADR → 10; P_W_PROG_Q → 84. |
| Name | Internal name | Unique identifier for the register (e.g. WPM_HD). Used in the device's log file and in internal diagnostics. |
| Type | Data type | Data type as defined in Chapter 8 (e.g. float, s16, P_WAHL, P_INT, ...). |
| Read/Write | Type of access | RO = read-only, RW = read and write. |
| Slave ID | Unit allocation | Which unit the value can be accessed via: HSM, WPM, 'WPM, HSM' or NONE. See Chapter 3. |
| Type information | Model details | For P_WAHL: List of permitted values. For P_INT / P_FLOAT: Unit, Min, Max. For float / s16 / s32: Unit, if applicable. Otherwise empty. |
| Name-UI | Display name | Short display name, as it appears on the heat pump's display. |
| Short description | Brief description | Optional brief description (often left blank). |
| Description | Detailed description | Functional description of the value (meaning, purpose, context). |
| Tip | Note | Further guidance on Commissioning or diagnostics (optional). |

# 10. STRUCTURE OF THE WORKSHEETS (SHEETS)

The Excel file contains an 'Explanation' sheet and several data sheets. The data sheets represent the respective access levels — anyone who has imported a specific licence (e.g. BMS) will find all the additional tabs available to them in the corresponding sheet.

| Sheet | Contents |
|---|---|
| Information | Overview of data types, slave IDs and login. |
| Initial Values | Values that are generally available without a licence (corresponds to the 'Start' level). |
| Added Value | Values that are additionally available with the 'Plus' licence level. |
| BMS Values | Values that are additionally available with the 'BMS' licence level. |

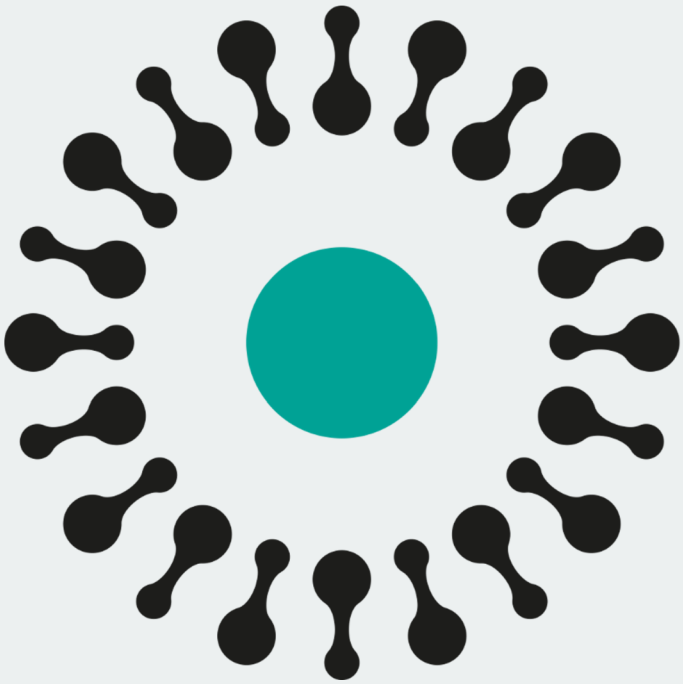
**NOTE**

The 'Read/Write' column within a sheet indicates whether the value in the relevant licence may only be read (RO) or may also be written to (RW). A value may appear in several sheets if it has been authorised for multiple levels.

## 10.1. PROCEDURE FOR SYSTEM INTEGRATORS

1. Enable the Modbus TCP server on the display (Settings / Network).
2. Read the login code from the display (Settings / Network / 'Modbus TCP Login Code').
3. Check licence — what level of access has been activated? (Settings / Activation codes).
4. Open the relevant sheet — Start Values, Plus Values or BMS Values
5. Determine the Slave ID for each value — see the 'Slave ID' column in Chapter 3.
6. Establish a connection — Modbus-TCP on port 502, write the login code to register 101 (FC16, s32, big-endian), check the status in register 100 (1 = OK).
7. Reading/writing values — FC03 for reading, FC06 for individual registers, FC16 for multi-word values. Refer to the Modbus list for data types and value limits.

If you have any questions regarding specific registers, weekly programmes (P_W_PROG_Q), activation codes or licensing issues, please contact OVUM Heiztechnik.



Einfach genial – genial einfach!

**OVUM Heiztechnik GmbH**

Lofererstraße 79

A-6322 Kirchbichl

Tel: +43 5332 812380

E-Mail: office@ovum.at

www.ovum.at

Version: 260630_en - Subject to technical changes, printing and typesetting errors