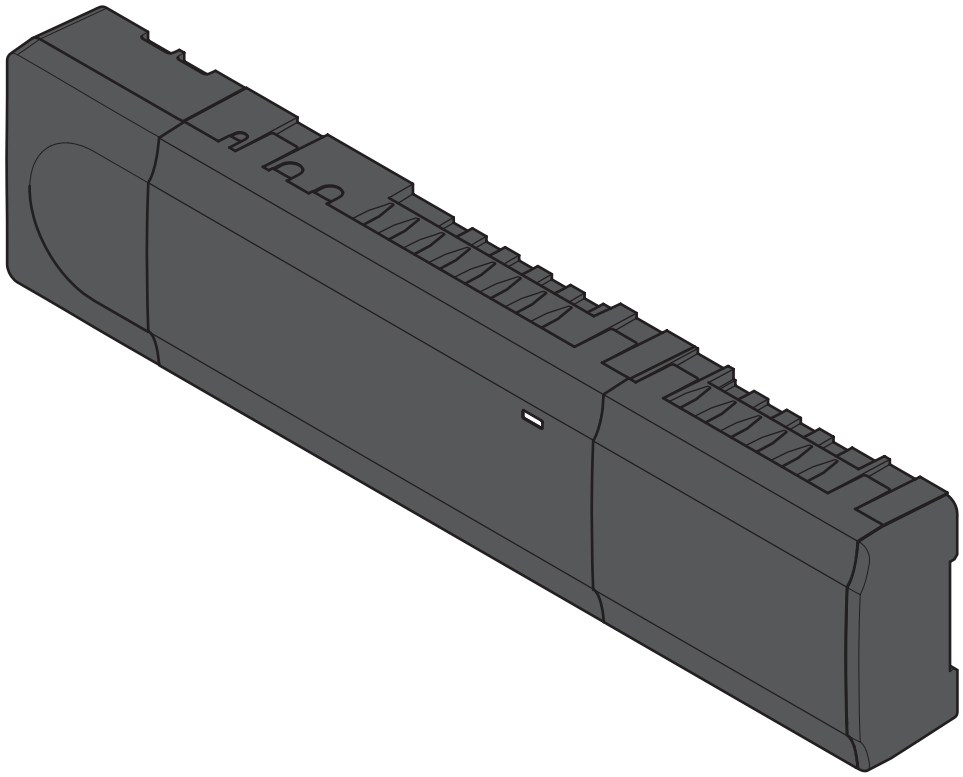
# Uponor Smatrix Base PRO Controller X-147
# Modbus RTU interface

EN Installation manual



# Table of Contents



# 1 Installation

## 1.1 General

With a Modbus RTU software update, the Base PRO controller can be connected and integrated to a building management system (BMS) through a Modbus-RTU connection over RS-485.

**Note**

Basic knowledge and training on BMS and Modbus RTU is required for installation and setup of Base PRO with Modbus RTU.

The BMS gets access to the following in the Base PRO system.

Read:

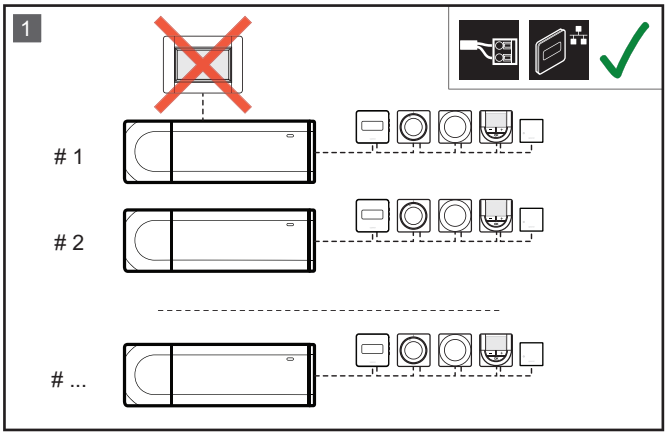
- Outdoor temperature
- Room temperature
- Floor temperature
- Humidity level
- Actuator status
- Pump or boiler status
- General purpose input (GPI) status
- Loss of thermostat connection
- Dynamic heat curve offset in integrated heat pump*

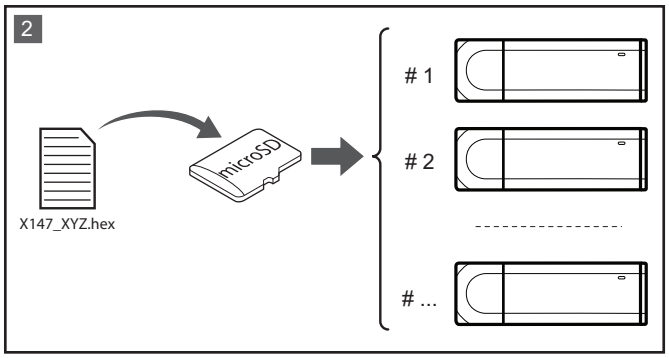
Read and write:

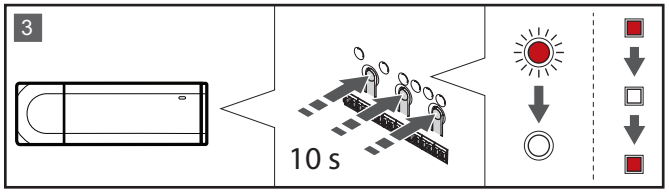
- Room setpoint
- Min/max levels for setpoint
- Min/max levels for floor temperature
- Heating/Cooling state
- Heating/Cooling offset
- Comfort/ECO
- Autobalancing on/off
- Cooling not allowed for a room
- Comfort setting
- Integrated heat pump defrost state*

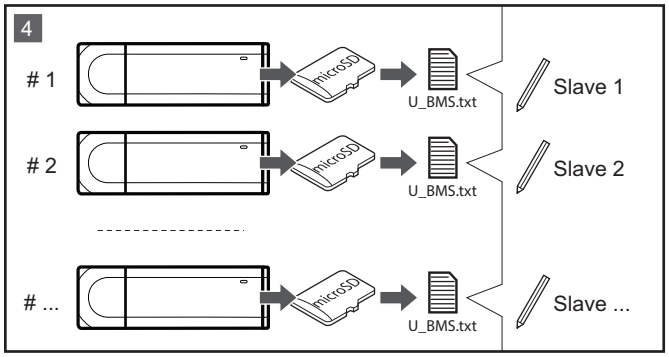
* *Requires **Heat Pump Integration via BMS** to be activated in U_BMS.txt.*

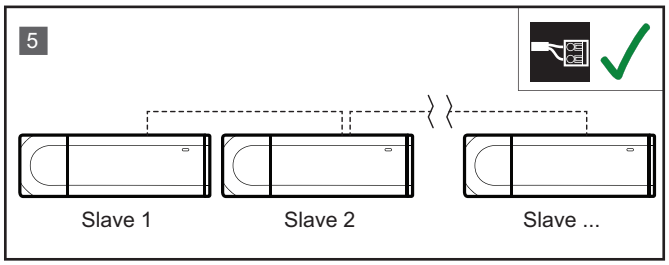
## 1.2 Preparation

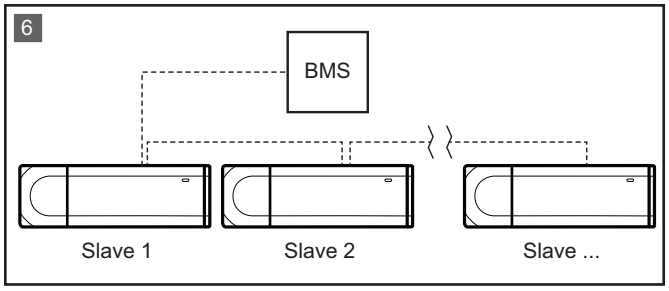





To connect a BMS to the Uponor Smatrix Base PRO controller some preparations must be made.

1. Connect and register thermostats, and system devices, to the controllers.

**Caution!**

If an Uponor Smatrix Base PRO Interface I-147 is connected, it must be unregistered and disconnected.

**Note**

Make sure that the thermostats are registered to consecutive channels so the BMS can read and write all channels in an efficient way. That is, addressing multiple channels in one message.

*See Uponor Smatrix Base PRO installation and operation manual, or quick guide, for information registering devices to the controller.*

2. Update the software of the controllers with the Modbus RTU compatible software (available on the Uponor website).

   *See separate documentation for information on how to update the software of a controller.*

3. Press the **<**, **OK**, and **>** buttons simultaneously (for about 10 seconds) until the power LED flashes, and all channel LEDs turn off. The controller has been reset to factory default.

4. Edit the U_BMS.txt file, on the controller microSD card, to setup the controller Modbus RTU interface.

   Available settings (default in **bold**):

   - BMS Slave address: **1** to 247
   - BMS Baud rate: **19200** or 9600
   - BMS Parity: **even**, odd, or no
   - Heat pump integration via BMS: **off** or on
     *on = enables dynamic heat curve offset in integrated heat pump, and integrated heat pump defrost state.*
   - Exclude zone from heat curve offset calculation:
     **0**, or 1 through 12 (0 = no channel)
     *Select master channels of the zones (lowest numbered channel in the zone) to exclude.*
     *The zone subchannels follow the state of the master channel.*
     *Example: 3,5,12 = exludes channels with master channels 3, 5 and 12.*
   - Bypass zone: **0**, or 1 to 12 (0 = no channel, example: 3,12)
     *Select master channels of the zones (lowest numbered channel in the zone) to bypass, MAXIMUM 2 channels.*
     *The zone subchannels follow the state of the master channel.*
     *Example: 4,12 = bypasses zones with master channels 4 and 12.*
   - Ceiling cooling channel: **0**, or 1 to 12 (0 = no channel, example: 1,4,10)
     *Select the channels which controls ceiling cooling in the zones. The zone subchannels will not follow the state of the master channel.*
     *Example: 1,4,10 = set ceiling cooling to channels 1, 4 and 10.*

   Repeat for each Base PRO controller in the system.

**Caution!**

Do not remove the microSD card while the controller is powered on.

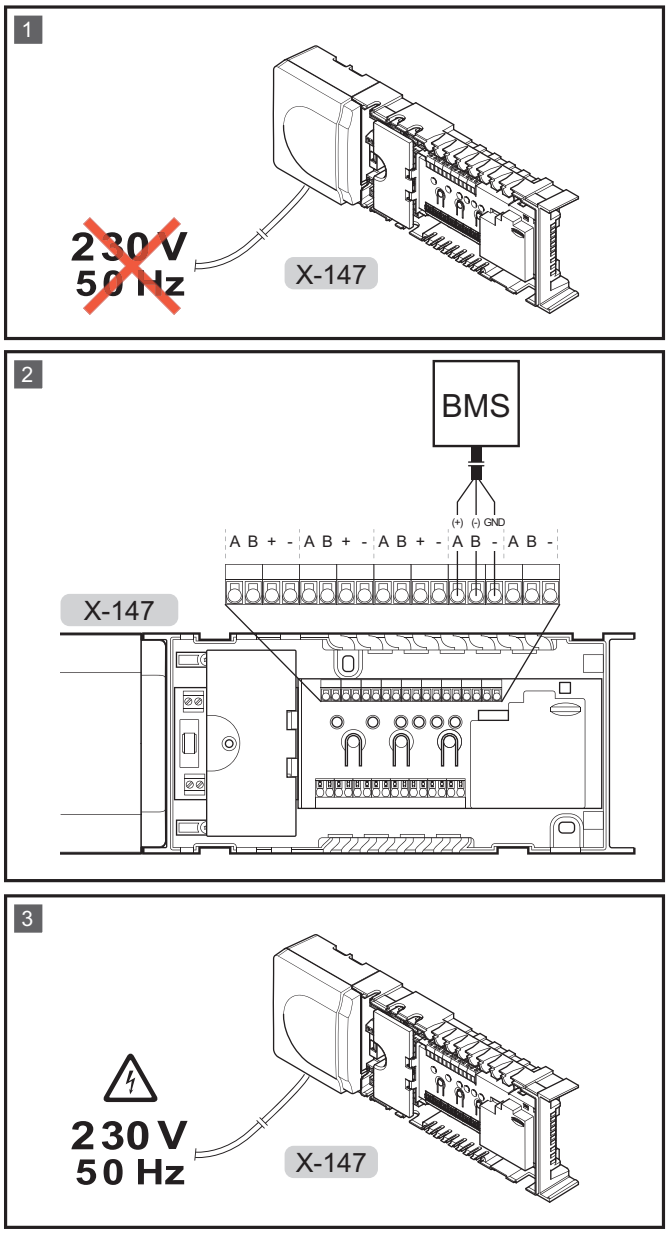
**Note**

Make sure the correct slave address is set (between 1 and 247, unique for each controller in the system), and that the other setting match the Modbus RTU settings in the BMS.

5. Connect the controllers to each other using the Base PRO controller system device bus.
6. Connect the Base PRO system to the BMS.

## 1.3 Connecting the BMS cable

The illustration below shows how to connect a BMS to the Modbus RTU interface on the controller.




To connect a BMS to the controller:

1. Ensure that the power is disconnected from the controller.
2. Connect the BMS cable to the Base PRO controller system device bus.

   Study the wiring diagrams of the controller to locate the connector position.

**Caution!**

Depending of EMC conditions in the installation, and/or distance between the devices, the - connection on the Base PRO controller might be needed.

3. Connect power to the controller again.

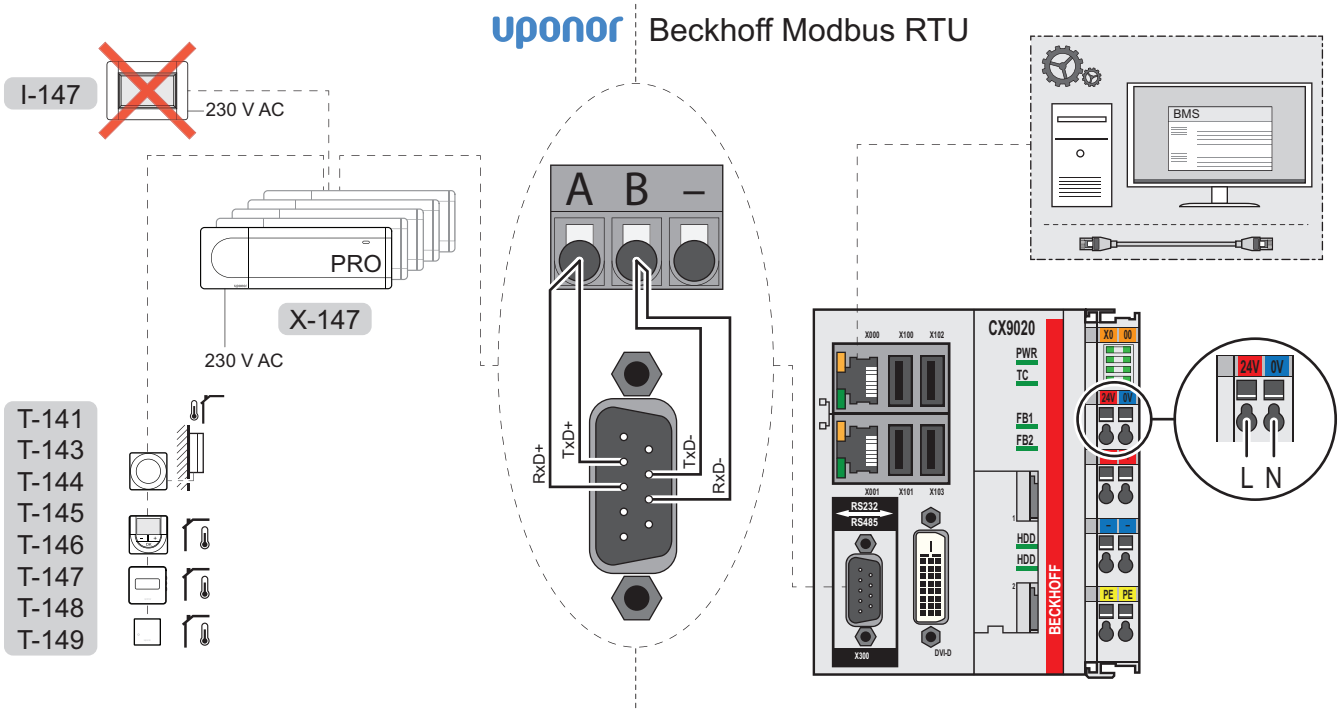
*See documentation from BMS supplier for more information about BMS integration.*

## 1.4 Modbus RTU settings

Configure the Modbus RTU interface in the BMS to match the controller settings:

- Bitrate: 19200 or 9600 bps
- Data bits: 8
- Stop bits: 1
- Parity bit: Even, odd, or no
- Flow control: No

## 1.5 Installation example, Beckhoff Modbus RTU




**Caution!**

If an Uponor Smatrix Base PRO Interface I-147 is connected, it must be unregistered and disconnected.

**Caution!**

Depending of EMC conditions in the installation, and/or distance between the devices, the - connection on the Base PRO controller might be needed.

# 2 Supported Modbus-RTU functions

The Base PRO controller supports the following functions for Modbus-RTU communication.

| Description | Function code (hex) |
|---|---|
| Read Coil | 0x01 |
| Read Discrete Input | 0x02 |
| Read Holding Registers | 0x03 |
| Read Input Registers | 0x04 |
| Write Single Coil | 0x05 |
| Write Single Holding Register | 0x06 |
| Write Multiple Coils | 0x0F |
| Write Multiple Holding Registers | 0x10 |

- Discrete Inputs (read only) and Coils (read and write) are function codes consisting of 1 bit of data. Most often on or off.
- Input Registers (read only) and Holding Registers (read and write) are function codes consisting of 2 bytes (16 bit) of data. Most often temperature data or settings.

## 2.1 Frame format: Read Coil

| Request | | | | |
|---|---|---|---|---|
| Slave address (0x01 – 0xF7) | Function code (0x01) | Starting register address (0x0000 – 0xFFFF) | Number of coils to read (1 – 2000) | CRC |
| 1 byte | 1 byte | 2 bytes | 2 bytes | 2 bytes |

| Response | | | | |
|---|---|---|---|---|
| Slave address (0x01 – 0xF7) | Function code (0x01) | Byte count (N) | Coil status (n = N, or n=N+1 if the number of coils to read can be divided by 8 with a reminder separated from 0) | CRC |
| 1 byte | 1 byte | 1 byte | n bytes | 2 bytes |

| Response in case of error | | | |
|---|---|---|---|
| Slave address (0x01 – 0xF7) | Error code (0x81) | Exception code (01, 02, 03, 04, or 06) | CRC |
| 1 byte | 1 byte | 1 byte | 2 bytes |

### Example:

| Request example: Read Heating/Cooling and Comfort/ECO states | | | | | | |
|---|---|---|---|---|---|---|
| Slave address (0x01) | Function code (0x01) | Starting register address HI (0x00) | Starting register address LO (0x00) | Number of coils to read HI (0x00) | Number of coils to read LO (0x02) | CRC |
| 1 byte | 1 byte | 1 byte | 1 byte | 1 byte | 1 bytes | 2 bytes |

| Response example: Read Heating/Cooling and Comfort/ECO states | | | | |
|---|---|---|---|---|
| Slave address (0x01) | Function code (0x01) | Byte count (0x01) | Coil status 1 to 8 (0x02) | CRC |
| 1 byte | 1 byte | 1 byte | 1 byte | 2 bytes |

## 2.2 Frame format: Read Discrete Input

| Request | | | | |
|---|---|---|---|---|
| Slave address (0x01 – 0xF7) | Function code (0x02) | Starting register address (0x0000 – 0xFFFF) | Number of coils to read (1 – 2000) | CRC |
| 1 byte | 1 byte | 2 bytes | 2 bytes | 2 bytes |

| Response | | | | |
|---|---|---|---|---|
| Slave address (0x01 – 0xF7) | Function code (0x02) | Byte count (N) | Coil status (n = N, or n=N+1 if the number of coils to read can be divided by 8 with a reminder separated from 0) | CRC |
| 1 byte | 1 byte | 1 byte | n bytes | 2 bytes |

| Response in case of error | | | |
|---|---|---|---|
| Slave address (0x01 – 0xF7) | Error code (0x82) | Exception code (01, 02, 03, 04, or 06) | CRC |
| 1 byte | 1 byte | 1 byte | 2 bytes |

### Examples:

| Request example:Read all discrete inputs | | | | | | |
|---|---|---|---|---|---|---|
| Slave address (0x01) | Function code (0x02) | Starting register address HI (0x00) | Starting register address LO (0x00) | Number of coils to read HI (0x00) | Number of coils to read LO 14 bits (0x0E) | CRC |
| 1 byte | 1 byte | 1 byte | 1 byte | 1 byte | 1 byte | 2 bytes |

| Response example: Read all discrete inputs | | | | | |
|---|---|---|---|---|---|
| Slave address (0x01) | Function code (0x02) | Byte count (0x02) | Coil status 1 to 8 (0xAA) | Coil status 9 to 16 (0x0A) | CRC |
| 1 byte | 1 byte | 2 byte | 1 bytes | 1 bytes | 2 bytes |

| Request example:Read actuator status, channel 12 | | | | | | |
|---|---|---|---|---|---|---|
| Slave address (0x01) | Function code (0x02) | Starting register address HI (0x00) | Starting register address LO Register 12 (0x0B) | Number of coils to read HI (0x00) | Number of coils to read LO 1 bit (0x01) | CRC |
| 1 byte | 1 byte | 1 byte | 1 byte | 1 byte | 1 byte | 2 bytes |

| Response example: Read actuator status, channel 12 | | | | |
|---|---|---|---|---|
| Slave address (0x01) | Function code (0x02) | Byte count (0x01) | Coil status channel 12 ON (0x01) | CRC |
| 1 byte | 1 byte | 1 byte | 1 bytes | 2 bytes |

| Request example:Read actuator status, all channels | | | | | | |
|---|---|---|---|---|---|---|
| Slave address (0x01) | Function code (0x02) | Starting register address HI (0x00) | Starting register address LO (0x00) | Number of coils to read HI (0x00) | Number of coils to read LO 12 bits (0x0C) | CRC |
| 1 byte | 1 byte | 1 byte | 1 byte | 1 byte | 1 byte | 2 bytes |

| Response example: Read actuator status, all channels | | | | | |
|---|---|---|---|---|---|
| Slave address (0x01) | Function code (0x02) | Byte count (0x02) | Coil status 1 to 8 (0xAA) | Coil status 9 to 12 (0x0A) | CRC |
| 1 byte | 1 byte | 2 byte | 1 bytes | 1 bytes | 2 bytes |

## 2.3 Frame format: Read Holding Registers

| Request | | | | |
|---|---|---|---|---|
| Slave address<br>(0x01 – 0xF7) | Function code<br>(0x03) | Starting register address<br>(0x0000 – 0xFFFF) | Quantity of holding registers to read<br>(0x0001 – 0x007D) | CRC |
| 1 byte | 1 byte | 2 bytes | 2 bytes | 2 bytes |

| Response | | | | |
|---|---|---|---|---|
| Slave address<br>(0x01 – 0xF7) | Function code<br>(0x03) | Byte count<br>(2*N)<br>N = quantity of registers | Holding register | CRC |
| 1 byte | 1 byte | 1 byte | 2*N bytes | 2 bytes |

| Response in case of error | | | |
|---|---|---|---|
| Slave address<br>(0x01 – 0xF7) | Error code<br>(0x83) | Exception code<br>(01, 02, 03, 04, or 06) | CRC |
| 1 byte | 1 byte | 1 byte | 2 bytes |

### Examples:

| Request example: Read setpoints (21.0 ˚C / 69.8 ˚F = value 689, 0x02BA),12 channels | | | | | | |
|---|---|---|---|---|---|---|
| Slave address<br>(0x01) | Function code<br>(0x03) | Starting register address HI<br>(0x00) | Starting register address LO<br>Start at 1 (0x00) | Quantity of holding registers to read HI<br>(0x00) | Quantity of holding registers to read LO<br>12 channels (0x0C) | CRC |
| 1 byte | 1 byte | 1 byte | 1 byte | 1 byte | 1 byte | 2 bytes |

| Response example: Read setpoints (21.0 ˚C / 69.8 ˚F = value 689, 0x02BA),12 channels | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Slave address<br>(0x01) | Function code<br>(0x03) | Byte count<br>(0x18) | Holding register HI (channel 1)<br>(0x02) | Holding register LO (channel 1)<br>(0xBA) | ... | Holding register HI (channel 12)<br>(0x02) | Holding register LO (channel 12)<br>(0xBA) | CRC |
| 1 byte | 1 byte | 1 byte | 1 byte | 1 byte | ... | 1 byte | 1 byte | 2 bytes |

| Request example: Read setpoint (21.0 ˚C / 69.8 ˚F = value 689, 0x02BA), channel 10 | | | | | | |
|---|---|---|---|---|---|---|
| Slave address<br>(0x01) | Function code<br>(0x03) | Starting register address HI<br>(0x00) | Starting register address LO<br>Start at 10 (0x09) | Quantity of holding registers to read HI<br>(0x00) | Quantity of holding registers to read LO<br>1 channel (0x01) | CRC |
| 1 byte | 1 byte | 1 byte | 1 byte | 1 byte | 1 byte | 2 bytes |

| Response example: Read setpoint (21.0 ˚C / 69.8 ˚F = value 689, 0x02BA), channel 10 | | | | | |
|---|---|---|---|---|---|
| Slave address<br>(0x01) | Function code<br>(0x03) | Byte count<br>(0x02) | Holding register HI<br>(0x02) | Holding register LO<br>(0xBA) | CRC |
| 1 byte | 1 byte | 1 byte | 1 bytes | 1 bytes | 2 bytes |

| Request example: Read comfort setting, channel 12 | | | | | | |
|---|---|---|---|---|---|---|
| Slave address (0x01) | Function code (0x03) | Starting register address HI (0x00) | Starting register address LO Start at 72 (0x47) | Quantity of holding registers to read HI (0x00) | Quantity of holding registers to read LO 1 channel (0x01) | CRC |
| 1 byte | 1 byte | 1 byte | 1 byte | 1 byte | 1 byte | 2 bytes |

| Response example: Read comfort setting, channel 12 | | | | | |
|---|---|---|---|---|---|
| Slave address (0x01) | Function code (0x03) | Byte count (0x02) | Holding register HI (0x00) | Holding register LO Off (0x00) | CRC |
| 1 byte | 1 byte | 1 byte | 1 bytes | 1 bytes | 2 bytes |

| Request example: Read heating/cooling offset temperature | | | | | | |
|---|---|---|---|---|---|---|
| Slave address (0x01) | Function code (0x03) | Starting register address HI (0x00) | Starting register address LO Start at 73 (0x48) | Quantity of holding registers to read HI (0x00) | Quantity of holding registers to read LO (0x01) | CRC |
| 1 byte | 1 byte | 1 byte | 1 byte | 1 byte | 1 byte | 2 bytes |

| Response example: Read heating/cooling offset temperature | | | | | |
|---|---|---|---|---|---|
| Slave address (0x01) | Function code (0x03) | Byte count (0x02) | Holding register HI (0x00) | Holding register LO 2 °C (0x24) | CRC |
| 1 byte | 1 byte | 1 byte | 1 bytes | 1 bytes | 2 bytes |

## 2.4 Frame format: Read Input Registers

| Request | | | | |
|---|---|---|---|---|
| Slave address (0x01 – 0xF7) | Function code (0x04) | Starting register address (0x0000 – 0xFFFF) | Quantity of input registers to read (0x0001 – 0x007D) | CRC |
| 1 byte | 1 byte | 2 bytes | 2 bytes | 2 bytes |

| Response | | | | |
|---|---|---|---|---|
| Slave address (0x01 – 0xF7) | Function code (0x04) | Byte count (2*N) | Input register | CRC |
| 1 byte | 1 byte | 1 byte | 2*N bytes | 2 bytes |

| Response in case of error | | | |
|---|---|---|---|
| Slave address (0x01 – 0xF7) | Error code (0x84) | Exception code (01, 02, 03, 04, or 06) | CRC |
| 1 byte | 1 byte | 1 byte | 2 bytes |

### Examples:

| Request example: Read room temperature data (21.0 ˚C / 69.8 ˚F = value 689, 0x02BA) for all zones | | | | | | |
|---|---|---|---|---|---|---|
| Slave address (0x01) | Function code (0x04) | Starting register address HI (0x00) | Starting register address LO Start at 2 (0x01) | Quantity of input registers to read HI (0x00) | Quantity of input registers to read LO 12 channels (0x0C) | CRC |
| 1 byte | 1 byte | 1 byte | 1 byte | 1 byte | 1 byte | 2 bytes |

| Response example: Read room temperature data (21.0 ˚C / 69.8 ˚F = value 689, 0x02BA) for all zones | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Slave address (0x01) | Function code (0x04) | Byte count (0x18) | Input register HI (channel 1) (0x02) | Input register LO (channel 1) (0xBA) | ... | Input register HI (channel 12) (0x02) | Input register LO (channel 12) (0xBA) | CRC |
| 1 byte | 1 byte | 1 byte | 1 bytes | 1 bytes | ... | 1 bytes | 1 bytes | 2 bytes |

| Request example: Read floor temperature data (21.0 ˚C / 69.8 ˚F = value 689, 0x02BA) for all zones | | | | | | |
|---|---|---|---|---|---|---|
| Slave address (0x01) | Function code (0x04) | Starting register address HI (0x00) | Starting register address LO Start at 26 (0x19) | Quantity of input registers to read HI (0x00) | Quantity of input registers to read LO 12 channels (0x0C) | CRC |
| 1 byte | 1 byte | 1 byte | 1 byte | 1 byte | 1 byte | 2 bytes |

| Response example: Read floor temperature data (21.0 ˚C / 69.8 ˚F = value 689, 0x02BA) for all zones | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Slave address (0x01) | Function code (0x04) | Byte count (0x18) | Input register HI (channel 1) (0x02) | Input register LO (channel 1) (0xBA) | ... | Input register HI (channel 12) (0x02) | Input register LO (channel 12) (0xBA) | CRC |
| 1 byte | 1 byte | 1 byte | 1 bytes | 1 bytes | ... | 1 bytes | 1 bytes | 2 bytes |

| Request example: Read humidity value (85% RH) for channel 12 | | | | | | |
|---|---|---|---|---|---|---|
| Slave address (0x01) | Function code (0x04) | Starting register address HI (0x00) | Starting register address LO Start at 25 (0x18) | Quantity of input registers to read HI (0x00) | Quantity of input registers to read LO 1 channel (0x01) | CRC |
| 1 byte | 1 byte | 1 byte | 1 byte | 1 byte | 1 byte | 2 bytes |

| Response example: Read humidity value (85% RH) for channel 12 | | | | | |
|---|---|---|---|---|---|
| Slave address (0x01) | Function code (0x04) | Byte count (0x02) | Input register HI (0x00) | Input register LO (0x55) | CRC |
| 1 byte | 1 byte | 1 byte | 1 bytes | 1 bytes | 2 bytes |

| Request example: Read thermostat loss alarm | | | | | | |
|---|---|---|---|---|---|---|
| Slave address (0x01) | Function code (0x04) | Starting register address HI (0x00) | Starting register address LO Start at 38 (0x25) | Quantity of input registers to read HI (0x00) | Quantity of input registers to read LO (0x01) | CRC |
| 1 byte | 1 byte | 1 byte | 1 byte | 1 byte | 1 byte | 2 bytes |

| Response example: Read thermostat loss alarm | | | | | |
|---|---|---|---|---|---|
| Slave address (0x01) | Function code (0x04) | Byte count (0x02) | Input register HI (0x08) | Input register LO (0x01) | CRC |
| 1 byte | 1 byte | 1 byte | 1 bytes | 1 bytes | 2 bytes |

| Request example: Offset heat pump heat curve with -10 degrees | | | | | | |
|---|---|---|---|---|---|---|
| Slave address (0x01) | Function code (0x04) | Starting register address HI (0x00) | Starting register address LO Start at 39 (0x26) | Quantity of input registers to read HI (0x00) | Quantity of input registers to read LO (0x01) | CRC |
| 1 byte | 1 byte | 1 byte | 1 byte | 1 byte | 1 byte | 2 bytes |

| Response example: Offset heat pump heat curve with -10 degrees | | | | | |
|---|---|---|---|---|---|
| Slave address (0x01) | Function code (0x04) | Byte count (0x02) | Input register HI (0x00) | Input register LO (0xF6) | CRC |
| 1 byte | 1 byte | 1 byte | 1 bytes | 1 bytes | 2 bytes |

## 2.5 Frame format: Write Single Coil

| Request | | | | |
|---|---|---|---|---|
| Slave address (0x01 – 0xF7) | Function code (0x05) | Starting register address (0x0000 – 0xFFFF) | Output value to write (0xFF00 = 1, 0x0000 = 0) | CRC |
| 1 byte | 1 byte | 2 bytes | 2 bytes | 2 bytes |

| Response | | | | |
|---|---|---|---|---|
| Slave address (0x01 – 0xF7) | Function code (0x05) | Output address to write (0x0000 – 0xFFFF) | Output value written (0xFF00 = 1, 0x0000 = 0) | CRC |
| 1 byte | 1 byte | 2 bytes | 2 bytes | 2 bytes |

| Response in case of error | | | |
|---|---|---|---|
| Slave address (0x01 – 0xF7) | Error code (0x85) | Exception code (01, 02, 03, 04, or 06) | CRC |
| 1 byte | 1 byte | 1 byte | 2 bytes |

### Example:

| Request example: Change state from ECO to Comfort | | | | | | |
|---|---|---|---|---|---|---|
| Slave address (0x01) | Function code (0x05) | Starting register address HI (0x00) | Starting register address LO Register 2 (0x01) | Output value to write HI (0x00) | Output value to write LO (0x00) | CRC |
| 1 byte | 1 byte | 1 byte | 1 byte | 1 byte | 1 byte | 2 bytes |

| Response example: Change state from ECO to Comfort | | | | | | |
|---|---|---|---|---|---|---|
| Slave address (0x01) | Function code (0x05) | Output address to write HI (0x00) | Output address to write LO Register 2 (0x01) | Output value written HI (0x00) | Output value written LO (0x00) | CRC |
| 1 byte | 1 byte | 1 byte | 1 byte | 1 byte | 1 byte | 2 bytes |

## 2.6 Frame format: Write Single Holding Register

| Request | | | | |
|---|---|---|---|---|
| Slave address (0x01 – 0xF7) | Function code (0x06) | Starting register address (0x0000 – 0xFFFF) | Register value (0x0000 – 0xFFFF) | CRC |
| 1 byte | 1 byte | 2 bytes | 2 bytes | 2 bytes |

| Response | | | | |
|---|---|---|---|---|
| Slave address (0x01 – 0xF7) | Function code (0x06) | Output address to write (0x0000 – 0xFFFF) | Output value written (0x0000 – 0xFFFF) | CRC |
| 1 byte | 1 byte | 2 bytes | 2 bytes | 2 bytes |

| Response in case of error | | | |
|---|---|---|---|
| Slave address (0x01 – 0xF7) | Error code (0x86) | Exception code (01, 02, 03, 04, or 06) | CRC |
| 1 byte | 1 byte | 1 byte | 2 bytes |

### Examples:

| Request example: Write 23 ˚C (73.4 ˚F = value 734, 0x02DE) setpoint, channel 5 | | | | | | |
|---|---|---|---|---|---|---|
| Slave address (0x01) | Function code (0x06) | Starting register address HI (0x00) | Starting register address LO Start at 5 (0x04) | Register value HI (0x02) | Register value LO (0xDE) | CRC |
| 1 byte | 1 byte | 1 byte | 1 byte | 1 byte | 1 byte | 2 bytes |

| Response example: Write 23 ˚C (73.4 ˚F = value 734, 0x02DE) setpoint, channel 5 | | | | | | |
|---|---|---|---|---|---|---|
| Slave address (0x01) | Function code (0x06) | Output address to write HI (0x00) | Output address to write LO Start at 5 (0x04) | Output value written HI (0x02) | Output value written LO (0xDE) | CRC |
| 1 byte | 1 byte | 1 byte | 1 byte | 1 byte | 1 byte | 2 bytes |

| Request example: Write 25 ˚C (77.0 ˚F = value 770, 0x0302) maximum floor temperature limit, channel 12 | | | | | | |
|---|---|---|---|---|---|---|
| Slave address (0x01) | Function code (0x06) | Starting register address HI (0x00) | Starting register address LO Start at 59 (0x3A) | Register value HI (0x03) | Register value LO (0x02) | CRC |
| 1 byte | 1 byte | 1 byte | 1 byte | 1 byte | 1 byte | 2 bytes |

| Response example: Write 25 ˚C (77.0 ˚F = value 770, 0x0302) maximum floor temperature limit, channel 12 | | | | | | |
|---|---|---|---|---|---|---|
| Slave address (0x01) | Function code (0x06) | Output address to write HI (0x00) | Output address to write LO Start at 59 (0x3A) | Output value written HI (0x03) | Output value written LO (0x02) | CRC |
| 1 byte | 1 byte | 1 byte | 1 byte | 1 byte | 1 byte | 2 bytes |

| Request example: Write 12 % comfort setting, channel 12 | | | | | | |
|---|---|---|---|---|---|---|
| Slave address (0x01) | Function code (0x06) | Starting register address HI (0x00) | Starting register address LO Start at 72 (0x47) | Register value HI (0x00) | Register value LO (0x0C) | CRC |
| 1 byte | 1 byte | 1 byte | 1 byte | 1 byte | 1 byte | 2 bytes |

| Response example: Write 12 % comfort setting, channel 12 | | | | | | |
|---|---|---|---|---|---|---|
| Slave address (0x01) | Function code (0x06) | Output address to write HI (0x00) | Output address to write LO Start at 72 (0x47) | Output value written HI (0x00) | Output value written LO (0x0C) | CRC |
| 1 byte | 1 byte | 1 byte | 1 byte | 1 byte | 1 byte | 2 bytes |

| Request example: Write 5 ˚C (41.0 ˚F = value 410, 0x005A) heating/cooling offset temperature | | | | | | |
|---|---|---|---|---|---|---|
| Slave address (0x01) | Function code (0x06) | Starting register address HI (0x00) | Starting register address LO Start at 73 (0x48) | Register value HI (0x00) | Register value LO (0x5A) | CRC |
| 1 byte | 1 byte | 1 byte | 1 byte | 1 byte | 1 byte | 2 bytes |

| Response example: Write 5 ˚C (41.0 ˚F = value 410, 0x005A) heating/cooling offset temperature | | | | | | |
|---|---|---|---|---|---|---|
| Slave address (0x01) | Function code (0x06) | Output address to write HI (0x00) | Output address to write LO Start at 73 (0x48) | Output value written HI (0x00) | Output value written LO (0x5A) | CRC |
| 1 byte | 1 byte | 1 byte | 1 byte | 1 byte | 1 byte | 2 bytes |

## 2.7 Frame format: Write Multiple Coils

| Request | | | | | | |
|---|---|---|---|---|---|---|
| Slave address (0x01 – 0xF7) | Function code (0x0F) | Starting register address (0x0000 – 0xFFFF) | Quantity of outputs (0x0001 – 0x07B0) | Byte count (N) | Output value to write | CRC |
| 1 byte | 1 byte | 2 bytes | 2 bytes | 1 byte | n bytes | 2 bytes |

| Response | | | | |
|---|---|---|---|---|
| Slave address (0x01 – 0xF7) | Function code (0x0F) | Starting register address (0x0000 – 0xFFFF) | Quantity of outputs (0x0001 – 0x07B0) | CRC |
| 1 byte | 1 byte | 2 bytes | 2 bytes | 2 bytes |

| Response in case of error | | | |
|---|---|---|---|
| Slave address (0x01 – 0xF7) | Error code (0x8F) | Exception code (01, 02, 03, 04, or 06) | CRC |
| 1 byte | 1 byte | 1 byte | 2 bytes |

### Example:

| Request example: Changes states to Cooling and ECO | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Slave address (0x01) | Function code (0x0F) | Starting register address HI (0x00) | Starting register address LO (0x00) | Quantity of outputs HI (0x00) | Quantity of outputs LO (0x02) | Byte count (0x01) | Output value to write (0x03) | CRC |
| 1 byte | 1 byte | 1 byte | 1 byte | 1 byte | 1 byte | 1 byte | ? bytes | 2 bytes |

| Response example: Changes states to Cooling and ECO | | | | | | |
|---|---|---|---|---|---|---|
| Slave address (0x01) | Function code (0x0F) | Starting register address HI (0x00) | Starting register address LO (0x00) | Quantity of outputs HI (0x00) | Quantity of outputs LO (0x02) | CRC |
| 1 byte | 1 byte | 1 byte | 1 byte | 1 byte | 1 byte | 2 bytes |

## 2.8 Frame format: Write Multiple Holding Registers

| Request | | | | | | |
|---|---|---|---|---|---|---|
| Slave address (0x01 – 0xF7) | Function code (0x10) | Starting register address (0x0000 – 0xFFFF) | Quantity of registers (0x0000 – 0x007B) | Byte count (2*N) N = quantity of registers | Register value | CRC |
| 1 byte | 1 byte | 2 bytes | 2 bytes | 1 byte | 2*N bytes | 2 bytes |

| Response | | | | |
|---|---|---|---|---|
| Slave address (0x01 – 0xF7) | Function code (0x10) | Starting register address (0x0000 – 0xFFFF) | Quantity of registers written (0x0000 – 0x007B) | CRC |
| 1 byte | 1 byte | 2 bytes | 2 bytes | 2 bytes |

| Response in case of error | | | |
|---|---|---|---|
| Slave address (0x01 – 0xF7) | Error code (0x90) | Exception code (01, 02, 03, 04, or 06) | CRC |
| 1 byte | 1 byte | 1 byte | 2 bytes |

### Example:

| Request example: Write 23 ˚C (73.4 ˚F = value 734, 0x02DE) maximum setpoint limit and 18 ˚C (64.4 ˚F = value 644, 0x0284) minimum setpoint limit, channel 5 | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Slave address (0x01) | Function code (0x10) | Starting register address HI (0x00) | Starting register address LO Start at 21 (0x14) | Quantity of registers HI (0x00) | Quantity of registers LO (0x02) | Byte count (0x04) | Register 21 value HI (0x02) | Register 21 value LO (0xDE) | Register 22 value HI (0x02) | Register 22 value LO (0x84) | CRC |
| 1 byte | 1 byte | 1 byte | 1 byte | 1 byte | 1 byte | 1 byte | 1 byte | 1 byte | 1 byte | 1 byte | 2 bytes |

| Response example: Write 23 ˚C (73.4 ˚F = value 734, 0x02DE) maximum setpoint limit and 18 ˚C (64.4 ˚F = value 644, 0x0284) minimum setpoint limit, channel 5 | | | | | | |
|---|---|---|---|---|---|---|
| Slave address (0x01) | Function code (0x10) | Starting register address HI (0x00) | Starting register address LO Start at 21 (0x14) | Quantity of registers written HI (0x00) | Quantity of registers written LO (0x02) | CRC |
| 1 byte | 1 byte | 1 byte | 1 byte | 1 byte | 1 byte | 2 bytes |

# 3 Available variables

These variables are available when communicating with the Base PRO controller (with the Modbus RTU compatible software installed).

## 3.1 Temperature conversion

The Uponor Smatrix Base PRO controller uses Fahrenheit to read and calculate temperatures.

To convert the data value into celcius, use the following mathematical formula.

Celcius = ((data value – 320)/1.8)/10

**Note**

If no data value is availabe the celcius value is set to 1802.6.

### Temperature (absolute value)

| Celsius (˚C) | Fahrenheit (˚F) | Data value | Hex value |
|---|---|---|---|
| -40.0 | -40.0 | -400 | 0xFE70 |
| -30.0 | -22.0 | -220 | 0xFF24 |
| -20.0 | -4.0 | -40 | 0xFFD8 |
| -17.8 | 0.0 | 0 | 0x0 |
| -10.0 | 14.0 | 140 | 0x8C |
| -5.0 | 23.0 | 230 | 0xE6 |
| 0.0 | 32.0 | 320 | 0x140 |
| 1.0 | 33.8 | 338 | 0x152 |
| 5.0 | 41.0 | 410 | 0x19A |
| 10.0 | 50.0 | 500 | 0x1F4 |
| 15.0 | 59.0 | 590 | 0x24E |
| 18.0 | 64.4 | 644 | 0x284 |
| 20.0 | 68.0 | 680 | 0x2A8 |
| 21.0 | 69.8 | 698 | 0x2BA |
| 25.0 | 77.0 | 770 | 0x302 |
| 30.0 | 86.0 | 860 | 0x35C |
| 35.0 | 95.0 | 950 | 0x3B6 |
| 40.0 | 104.0 | 1040 | 0x410 |

### Temperature (relative value)

| Celsius (˚C) | Data value | Hex value |
|---|---|---|
| 0.1 | 2 | 0x02 |
| 1.0 | 18 | 0x13 |
| 4.0 | 72 | 0x48 |
| 5.0 | 90 | 0x5A |

### Humidity (absolute value)

| Relative humidity (%) | Hex value |
|---|---|
| 0 | 0x0000 |
| 5 | 0x0005 |
| 10 | 0x000A |
| 15 | 0x000F |
| 20 | 0x0014 |
| 25 | 0x0019 |
| 30 | 0x001E |
| 35 | 0x0023 |
| 40 | 0x0028 |
| 45 | 0x002D |
| 50 | 0x0032 |
| 55 | 0x0037 |
| 60 | 0x003C |
| 65 | 0x0041 |
| 70 | 0x0046 |
| 75 | 0x004B |
| 80 | 0x0050 |
| 85 | 0x0055 |
| 90 | 0x005A |
| 95 | 0x005F |
| 100 | 0x0064 |

# 3.2 Controller limitations

## Cooling state

An ordinary Base PRO system (with touch screen, not connected to a BMS) uses an offset temperature to adjust the setpoints when switching between heating and cooling. This setting is only available via the touch screen interface (I-167), which is not installed when connected to a BMS.

When connected to a BMS, and the heating/cooling state is set to cooling, the cooling offset is set to 0 ˚C (0 °F), leaving the zone setpoints to be changed by the BMS.

## ECO state

If the Base PRO system is set to ECO state the controller applies an ECO setback value of 4 ˚C (4 °F) to the setpoints (which is shown in the thermostats).

When the BMS reads the setpoint, when in ECO state, it will recieve the actual setpoint (without the ECO setback value applied).

| Example | |
|---|---|
| **Setpoint written/read from BMS** | 21 ˚C (69.8 °F) |
| **Comfort/ECO state** | 1 (ECO) |
| **Setpoint shown in thermostat** | 17 ˚C (62.6 °F) |

## Zones

The Base PRO controller can control a maximum of 6 actuator channels, 12 if a slave module is installed. These channels are then divided into zones which are linked to a thermostat.

A zone can consist of 1 to 12 channels, and not all channels must be connected to an actuator and linked to a thermostat.

When writing to a channel, address the lowest channel number (master channel) in the zone (if more chan one channel in the zone) in order to change the value/state for the whole zone.

## Comfort setting

The Base PRO can set a basic level of comfort for a zone when there is no demand for heating. The value set is the percentage of time the actuators are opened.

It will shorten the heat up time for the room, which is useful in rooms where other heating sources, e.g. a fireplace, is present.

When the BMS reads the comfort setting it will recieve the actual percentage.

## Max/min limitations

If the BMS writes a setpoint to a zone, the Base PRO controller applies maximum and minimum limitations before checking if there is a demand for heating or cooling. The limited setpoint is also shown on the thermostat.

The maximum and minumlimitations are also affected whether the controller is set to Comfort or ECO state.

When the BMS reads the setpoint it will recieve the actual setpoint (without limitation applied).

| Example | |
|---|---|
| **Setpoint written/read from BMS** | 28 ˚C (82.4 °F) |
| **Room setpoint max temperature** | 25 ˚C (77 °F) |
| **Room setpoint min temperature** | 15 ˚C (59 °F) |
| **Setpoint in room thermostat** | 25 ˚C (77 °F) |
| **Comfort/ECO state** | 0 (Comfort) |
| **Room temperature** | 20 ˚C (68 °F) |
| **Heating demand** | YES |

## Dynamic heat curve offset in integrated heat pump

The Base PRO system can dynamically adjust the heat curve offset in a BMS integrated heat pump. The BMS reads the value from the controller and sends it to the heat pump.

The heat curve can be offset between -10 ˚C and +10 ˚C (-10 ˚F and +10 ˚F).

When the BMS reads the heat curve offset from the controller it will recieve a absolute degree value to offset the heat curve in the heat pump.

**Caution!**

The heat pump must be connected to the BMS.

Do not use the heat pump integration connectors on the Base PRO controller.

**Note**

This function requires **Heat pump integration via BMS** set to **on** on the microSD card to be activated.

## 3.3 Coil data points

These datapoints can be both read or written, and contains the status of different controller settings (binary, on/off).

**Note**

1 = 0xFF00,
0 = 0x0000

| Display name | Register address (decimal) | Function code (Hex) | Type | Analogue/ Binary | Coding |
|---|---|---|---|---|---|
| Heating/Cooling state | 00001 | 0x01<br>0x05<br>0x0F | Read<br>Write<br>Write multiple | Binary | Unsigned.<br>1 = Cooling, 0 = Heating (default) |
| Comfort/ECO state | 00002 | 0x01<br>0x05<br>0x0F | Read<br>Write<br>Write multiple | Binary | Unsigned.<br>1 = ECO, 0 = Comfort (default) |
| Auto balancing on/off | 00003 | 0x01<br>0x05<br>0x0F | Read<br>Write<br>Write multiple | Binary | Unsigned.<br>1 = ON (default), 0 = Off |
| Cooling not allowed channel 1 | 00004 | 0x01<br>0x05<br>0x0F | Read<br>Write<br>Write multiple | Binary | Unsigned.<br>0 = Cooling not allowed<br>1 = Cooling allowed (default) |
| Cooling not allowed channel 2 | 00005 | 0x01<br>0x05<br>0x0F | Read<br>Write<br>Write multiple | Binary | Unsigned.<br>0 = Cooling not allowed<br>1 = Cooling allowed (default) |
| Cooling not allowed channel 3 | 00006 | 0x01<br>0x05<br>0x0F | Read<br>Write<br>Write multiple | Binary | Unsigned.<br>0 = Cooling not allowed<br>1 = Cooling allowed (default) |
| Cooling not allowed channel 4 | 00007 | 0x01<br>0x05<br>0x0F | Read<br>Write<br>Write multiple | Binary | Unsigned.<br>0 = Cooling not allowed<br>1 = Cooling allowed (default) |
| Cooling not allowed channel 5 | 00008 | 0x01<br>0x05<br>0x0F | Read<br>Write<br>Write multiple | Binary | Unsigned.<br>0 = Cooling not allowed<br>1 = Cooling allowed (default) |
| Cooling not allowed channel 6 | 00009 | 0x01<br>0x05<br>0x0F | Read<br>Write<br>Write multiple | Binary | Unsigned.<br>0 = Cooling not allowed<br>1 = Cooling allowed (default) |
| Cooling not allowed channel 7 | 00010 | 0x01<br>0x05<br>0x0F | Read<br>Write<br>Write multiple | Binary | Unsigned.<br>0 = Cooling not allowed<br>1 = Cooling allowed (default) |
| Cooling not allowed channel 8 | 00011 | 0x01<br>0x05<br>0x0F | Read<br>Write<br>Write multiple | Binary | Unsigned.<br>0 = Cooling not allowed<br>1 = Cooling allowed (default) |
| Cooling not allowed channel 9 | 00012 | 0x01<br>0x05<br>0x0F | Read<br>Write<br>Write multiple | Binary | Unsigned.<br>0 = Cooling not allowed<br>1 = Cooling allowed (default) |

| Display name | Register address (decimal) | Function code (Hex) | Type | Analogue/ Binary | Coding |
|---|---|---|---|---|---|
| Cooling not allowed channel 10 | 00013 | 0x01 | Read | Binary | Unsigned.<br>0 = Cooling not allowed<br>1 = Cooling allowed (default) |
| | | 0x05 | Write | | |
| | | 0x0F | Write multiple | | |
| Cooling not allowed channel 11 | 00014 | 0x01 | Read | Binary | Unsigned.<br>0 = Cooling not allowed<br>1 = Cooling allowed (default) |
| | | 0x05 | Write | | |
| | | 0x0F | Write multiple | | |
| Cooling not allowed channel 12 | 00015 | 0x01 | Read | Binary | Unsigned.<br>0 = Cooling not allowed<br>1 = Cooling allowed (default) |
| | | 0x05 | Write | | |
| | | 0x0F | Write multiple | | |
| Integrated heat pump defrost state* | 00016 | 0x01 | Read | Binary | Unsigned.<br>0 = Heat pump defrost off/stopped (default)<br>1 = Heat pump defrost started |
| | | 0x05 | Write | | |
| | | 0x0F | Write multiple | | |

* *Requires **Heat Pump Integration via BMS** to be activated in U_BMS.txt. It takes about 2 minutes for the actuators in the Base PRO system to be fully opened.*

## 3.4 Discrete Input data points

These datapoints are read only and show the actuator, pump/boiler, and GPI status on the controller (binary, on/off).

| Display name | Register address (decimal) | Function code (Hex) | Type | Analogue/ Binary | Coding |
|---|---|---|---|---|---|
| Actuator status/channel 1 (bit 0) | 10001 | 0x02 | Read | Binary | Unsigned. 1 = ON, 0 = Off. |
| Actuator status/channel 2 (bit 1) | 10002 | 0x02 | Read | Binary | Unsigned. 1 = ON, 0 = Off. |
| Actuator status/channel 3 (bit 2) | 10003 | 0x02 | Read | Binary | Unsigned. 1 = ON, 0 = Off. |
| Actuator status/channel 4 (bit 3) | 10004 | 0x02 | Read | Binary | Unsigned. 1 = ON, 0 = Off. |
| Actuator status/channel 5 (bit 4) | 10005 | 0x02 | Read | Binary | Unsigned. 1 = ON, 0 = Off. |
| Actuator status/channel 6 (bit 5) | 10006 | 0x02 | Read | Binary | Unsigned. 1 = ON, 0 = Off. |
| Actuator status/channel 7 (bit 6) | 10007 | 0x02 | Read | Binary | Unsigned. 1 = ON, 0 = Off. |
| Actuator status/channel 8 (bit 7) | 10008 | 0x02 | Read | Binary | Unsigned. 1 = ON, 0 = Off. |
| Actuator status/channel 9 (bit 8) | 10009 | 0x02 | Read | Binary | Unsigned. 1 = ON, 0 = Off. |
| Actuator status/channel 10 (bit 9) | 10010 | 0x02 | Read | Binary | Unsigned. 1 = ON, 0 = Off. |
| Actuator status/channel 11 (bit 10) | 10011 | 0x02 | Read | Binary | Unsigned. 1 = ON, 0 = Off. |
| Actuator status/channel 12 (bit 11) | 10012 | 0x02 | Read | Binary | Unsigned. 1 = ON, 0 = Off. |
| Pump/Boiler (bit 12) | 10013 | 0x02 | Read | Binary | Unsigned. 1 = ON, 0 = Off. |
| GPI status (bit 13) | 10014 | 0x02 | Read | Binary | Unsigned. 1 = ON, 0 = Off. |
| | 10015 | | | | |
| | 10016 | | | | |

## 3.5 Input Register data points

These datapoints are read only and show the current sensor value (temperature or humidity) for each channel (1 to 12) on the controller.

**Note**

All temperature data is sent in Fahrenheit.

**Note**

0x7FFF is returned if no data is available for the channel.

| Display name | Register address (decimal) | Function code (Hex) | Type | Analogue/ Binary | Coding |
|---|---|---|---|---|---|
| Controller outdoor temperature | 30001 | 0x04 | Read | Analogue | Signed 16 bit integer. |
| Room temperature data channel 1 | 30002 | 0x04 | Read | Analogue | Signed 16 bit integer. |
| Room temperature data channel 2 | 30003 | 0x04 | Read | Analogue | Signed 16 bit integer. |
| Room temperature data channel 3 | 30004 | 0x04 | Read | Analogue | Signed 16 bit integer. |
| Room temperature data channel 4 | 30005 | 0x04 | Read | Analogue | Signed 16 bit integer. |
| Room temperature data channel 5 | 30006 | 0x04 | Read | Analogue | Signed 16 bit integer. |
| Room temperature data channel 6 | 30007 | 0x04 | Read | Analogue | Signed 16 bit integer. |
| Room temperature data channel 7 | 30008 | 0x04 | Read | Analogue | Signed 16 bit integer. |
| Room temperature data channel 8 | 30009 | 0x04 | Read | Analogue | Signed 16 bit integer. |
| Room temperature data channel 9 | 30010 | 0x04 | Read | Analogue | Signed 16 bit integer. |
| Room temperature data channel 10 | 30011 | 0x04 | Read | Analogue | Signed 16 bit integer. |
| Room temperature data channel 11 | 30012 | 0x04 | Read | Analogue | Signed 16 bit integer. |
| Room temperature data channel 12 | 30013 | 0x04 | Read | Analogue | Signed 16 bit integer. |
| Humidity data channel 1 | 30014 | 0x04 | Read | Analogue | Unsigned 16 bit integer. 0-100 % relative humidity |
| Humidity data channel 2 | 30015 | 0x04 | Read | Analogue | Unsigned 16 bit integer. 0-100 % relative humidity |
| Humidity data channel 3 | 30016 | 0x04 | Read | Analogue | Unsigned 16 bit integer. 0-100 % relative humidity |
| Humidity data channel 4 | 30017 | 0x04 | Read | Analogue | Unsigned 16 bit integer. 0-100 % relative humidity |
| Humidity data channel 5 | 30018 | 0x04 | Read | Analogue | Unsigned 16 bit integer. 0-100 % relative humidity |
| Humidity data channel 6 | 30019 | 0x04 | Read | Analogue | Unsigned 16 bit integer. 0-100 % relative humidity |
| Humidity data channel 7 | 30020 | 0x04 | Read | Analogue | Unsigned 16 bit integer. 0-100 % relative humidity |
| Humidity data channel 8 | 30021 | 0x04 | Read | Analogue | Unsigned 16 bit integer. 0-100 % relative humidity |
| Humidity data channel 9 | 30022 | 0x04 | Read | Analogue | Unsigned 16 bit integer. 0-100 % relative humidity |
| Humidity data channel 10 | 30023 | 0x04 | Read | Analogue | Unsigned 16 bit integer. 0-100 % relative humidity |
| Humidity data channel 11 | 30024 | 0x04 | Read | Analogue | Unsigned 16 bit integer. 0-100 % relative humidity |
| Humidity data channel 12 | 30025 | 0x04 | Read | Analogue | Unsigned 16 bit integer. 0-100 % relative humidity |

| Display name | Register address (decimal) | Function code (Hex) | Type | Analogue/ Binary | Coding |
|---|---|---|---|---|---|
| Floor temperature data channel 1 | 30026 | 0x04 | Read | Analogue | Signed 16 bit integer. |
| Floor temperature data channel 2 | 30027 | 0x04 | Read | Analogue | Signed 16 bit integer. |
| Floor temperature data channel 3 | 30028 | 0x04 | Read | Analogue | Signed 16 bit integer. |
| Floor temperature data channel 4 | 30029 | 0x04 | Read | Analogue | Signed 16 bit integer. |
| Floor temperature data channel 5 | 30030 | 0x04 | Read | Analogue | Signed 16 bit integer. |
| Floor temperature data channel 6 | 30031 | 0x04 | Read | Analogue | Signed 16 bit integer. |
| Floor temperature data channel 7 | 30032 | 0x04 | Read | Analogue | Signed 16 bit integer. |
| Floor temperature data channel 8 | 30033 | 0x04 | Read | Analogue | Signed 16 bit integer. |
| Floor temperature data channel 9 | 30034 | 0x04 | Read | Analogue | Signed 16 bit integer. |
| Floor temperature data channel 10 | 30035 | 0x04 | Read | Analogue | Signed 16 bit integer. |
| Floor temperature data channel 11 | 30036 | 0x04 | Read | Analogue | Signed 16 bit integer. |
| Floor temperature data channel 12 | 30037 | 0x04 | Read | Analogue | Signed 16 bit integer. |
| Thermostat loss | 30038 | 0x04 | Read | Binary | Unsigned.<br>1 = Thermostat loss, 0 = No alarm. |
| Dynamic heat curve offset in integrated heat pump* | 30039 | 0x04 | Read | Analogue | Signed 16 bit integer.<br>Absolute degree value (-10 to +10)<br>+10 = 000A<br>+1 = 0001<br>0 = 0000<br>-1 = 00FF<br>-10 = 00F6 |

* *Requires **Heat Pump Integration via BMS** to be activated in U_BMS.txt.*

# 3.6 Holding Register data points

These datapoints can be both read or written, and contain temperature setpoint and max/min temperature limits for each channel (1 to 12) on the controller.

**Note**

All temperature data is sent in Fahrenheit.

**Note**

0x7FFF is returned if no data is available for the channel.

| Display name | Register address (decimal) | Function code (Hex) | Type | Analogue/ Binary | Coding |
|---|---|---|---|---|---|
| Room setpoint temperature channel 1 | 40001 | 0x03 | Read | Analogue | Signed 16 bit integer. Range: min - max temperature, default: 21 °C (69.8 °F). |
| | | 0x06 | Write | | |
| | | 0x10 | Write multiple | | |
| Room setpoint temperature channel 2 | 40002 | 0x03 | Read | Analogue | Signed 16 bit integer. Range: min - max temperature, default: 21 °C (69.8 °F). |
| | | 0x06 | Write | | |
| | | 0x10 | Write multiple | | |
| Room setpoint temperature channel 3 | 40003 | 0x03 | Read | Analogue | Signed 16 bit integer. Range: min - max temperature, default: 21 °C (69.8 °F). |
| | | 0x06 | Write | | |
| | | 0x10 | Write multiple | | |
| Room setpoint temperature channel 4 | 40004 | 0x03 | Read | Analogue | Signed 16 bit integer. Range: min - max temperature, default: 21 °C (69.8 °F). |
| | | 0x06 | Write | | |
| | | 0x10 | Write multiple | | |
| Room setpoint temperature channel 5 | 40005 | 0x03 | Read | Analogue | Signed 16 bit integer. Range: min - max temperature, default: 21 °C (69.8 °F). |
| | | 0x06 | Write | | |
| | | 0x10 | Write multiple | | |
| Room setpoint temperature channel 6 | 40006 | 0x03 | Read | Analogue | Signed 16 bit integer. Range: min - max temperature, default: 21 °C (69.8 °F). |
| | | 0x06 | Write | | |
| | | 0x10 | Write multiple | | |
| Room setpoint temperature channel 7 | 40007 | 0x03 | Read | Analogue | Signed 16 bit integer. Range: min - max temperature, default: 21 °C (69.8 °F). |
| | | 0x06 | Write | | |
| | | 0x10 | Write multiple | | |
| Room setpoint temperature channel 8 | 40008 | 0x03 | Read | Analogue | Signed 16 bit integer. Range: min - max temperature, default: 21 °C (69.8 °F). |
| | | 0x06 | Write | | |
| | | 0x10 | Write multiple | | |
| Room setpoint temperature channel 9 | 40009 | 0x03 | Read | Analogue | Signed 16 bit integer. Range: min - max temperature, default: 21 °C (69.8 °F). |
| | | 0x06 | Write | | |
| | | 0x10 | Write multiple | | |
| Room setpoint temperature channel 10 | 40010 | 0x03 | Read | Analogue | Signed 16 bit integer. Range: min - max temperature, default: 21 °C (69.8 °F). |
| | | 0x06 | Write | | |
| | | 0x10 | Write multiple | | |
| Room setpoint temperature channel 11 | 40011 | 0x03 | Read | Analogue | Signed 16 bit integer. Range: min - max temperature, default: 21 °C (69.8 °F). |
| | | 0x06 | Write | | |
| | | 0x10 | Write multiple | | |
| Room setpoint temperature channel 12 | 40012 | 0x03 | Read | Analogue | Signed 16 bit integer. Range: min - max temperature, default: 21 °C (69.8 °F). |
| | | 0x06 | Write | | |
| | | 0x10 | Write multiple | | |

| Display name | Register address (decimal) | Function code (Hex) | Type | Analogue/ Binary | Coding |
|---|---|---|---|---|---|
| Room setpoint max temperature channel 1 | 40013 | 0x06 | Write | Analogue | Signed 16 bit integer. Range: min temperature - 35, default: 35 °C (95 °F). |
| | | 0x10 | Write multiple | | |
| Room setpoint min temperature channel 1 | 40014 | 0x06 | Write | Analogue | Signed 16 bit integer. Range: 5 - max temperature, default: 5 °C (41 °F). |
| | | 0x10 | Write multiple | | |
| Room setpoint max temperature channel 2 | 40015 | 0x06 | Write | Analogue | Signed 16 bit integer. Range: min temperature - 35, default: 35 °C (95 °F). |
| | | 0x10 | Write multiple | | |
| Room setpoint min temperature channel 2 | 40016 | 0x06 | Write | Analogue | Signed 16 bit integer. Range: 5 - max temperature, default: 5 °C (41 °F). |
| | | 0x10 | Write multiple | | |
| Room setpoint max temperature channel 3 | 40017 | 0x06 | Write | Analogue | Signed 16 bit integer. Range: min temperature - 35, default: 35 °C (95 °F). |
| | | 0x10 | Write multiple | | |
| Room setpoint min temperature channel 3 | 40018 | 0x06 | Write | Analogue | Signed 16 bit integer. Range: 5 - max temperature, default: 5 °C (41 °F). |
| | | 0x10 | Write multiple | | |
| Room setpoint max temperature channel 4 | 40019 | 0x06 | Write | Analogue | Signed 16 bit integer. Range: min temperature - 35, default: 35 °C (95 °F). |
| | | 0x10 | Write multiple | | |
| Room setpoint min temperature channel 4 | 40020 | 0x06 | Write | Analogue | Signed 16 bit integer. Range: 5 - max temperature, default: 5 °C (41 °F). |
| | | 0x10 | Write multiple | | |
| Room setpoint max temperature channel 5 | 40021 | 0x06 | Write | Analogue | Signed 16 bit integer. Range: min temperature - 35, default: 35 °C (95 °F). |
| | | 0x10 | Write multiple | | |
| Room setpoint min temperature channel 5 | 40022 | 0x06 | Write | Analogue | Signed 16 bit integer. Range: 5 - max temperature, default: 5 °C (41 °F). |
| | | 0x10 | Write multiple | | |
| Room setpoint max temperature channel 6 | 40023 | 0x06 | Write | Analogue | Signed 16 bit integer. Range: min temperature - 35, default: 35 °C (95 °F). |
| | | 0x10 | Write multiple | | |
| Room setpoint min temperature channel 6 | 40024 | 0x06 | Write | Analogue | Signed 16 bit integer. Range: 5 - max temperature, default: 5 °C (41 °F). |
| | | 0x10 | Write multiple | | |
| Room setpoint max temperature channel 7 | 40025 | 0x06 | Write | Analogue | Signed 16 bit integer. Range: min temperature - 35, default: 35 °C (95 °F). |
| | | 0x10 | Write multiple | | |
| Room setpoint min temperature channel 7 | 40026 | 0x06 | Write | Analogue | Signed 16 bit integer. Range: 5 - max temperature, default: 5 °C (41 °F). |
| | | 0x10 | Write multiple | | |
| Room setpoint max temperature channel 8 | 40027 | 0x06 | Write | Analogue | Signed 16 bit integer. Range: min temperature - 35, default: 35 °C (95 °F). |
| | | 0x10 | Write multiple | | |
| Room setpoint min temperature channel 8 | 40028 | 0x06 | Write | Analogue | Signed 16 bit integer. Range: 5 - max temperature, default: 5 °C (41 °F). |
| | | 0x10 | Write multiple | | |
| Room setpoint max temperature channel 9 | 40029 | 0x06 | Write | Analogue | Signed 16 bit integer. Range: min temperature - 35, default: 35 °C (95 °F). |
| | | 0x10 | Write multiple | | |
| Room setpoint min temperature channel 9 | 40030 | 0x06 | Write | Analogue | Signed 16 bit integer. Range: 5 - max temperature, default: 5 °C (41 °F). |
| | | 0x10 | Write multiple | | |
| Room setpoint max temperature channel 10 | 40031 | 0x06 | Write | Analogue | Signed 16 bit integer. Range: min temperature - 35, default: 35 °C (95 °F). |
| | | 0x10 | Write multiple | | |
| Room setpoint min temperature channel 10 | 40032 | 0x06 | Write | Analogue | Signed 16 bit integer. Range: 5 - max temperature, default: 5 °C (41 °F). |
| | | 0x10 | Write multiple | | |

| Display name | Register address (decimal) | Function code (Hex) | Type | Analogue/ Binary | Coding |
|---|---|---|---|---|---|
| Room setpoint max temperature channel 11 | 40033 | 0x06 | Write | Analogue | Signed 16 bit integer. Range: min temperature - 35, default: 35 °C (95 °F). |
| | | 0x10 | Write multiple | | |
| Room setpoint min temperature channel 11 | 40034 | 0x06 | Write | Analogue | Signed 16 bit integer. Range: 5 - max temperature, default: 5 °C (41 °F). |
| | | 0x10 | Write multiple | | |
| Room setpoint max temperature channel 12 | 40035 | 0x06 | Write | Analogue | Signed 16 bit integer. Range: min temperature - 35, default: 35 °C (95 °F). |
| | | 0x10 | Write multiple | | |
| Room setpoint min temperature channel 12 | 40036 | 0x06 | Write | Analogue | Signed 16 bit integer. Range: 5 - max temperature, default: 5 °C (41 °F). |
| | | 0x10 | Write multiple | | |
| Room floor max limit temperature channel 1 | 40037 | 0x06 | Write | Analogue | Signed 16 bit integer. Range: min temperature or 20 - 35, default: 26 °C (78.8 °F). |
| | | 0x10 | Write multiple | | |
| Room floor min limit temperature channel 1 | 40038 | 0x06 | Write | Analogue | Signed 16 bit integer. Range: 10 - 30 or max temperature, default: 20 °C (68 °F). |
| | | 0x10 | Write multiple | | |
| Room floor max limit temperature channel 2 | 40039 | 0x06 | Write | Analogue | Signed 16 bit integer. Range: min temperature or 20 - 35, default: 26 °C (78.8 °F). |
| | | 0x10 | Write multiple | | |
| Room floor min limit temperature channel 2 | 40040 | 0x06 | Write | Analogue | Signed 16 bit integer. Range: 10 - 30 or max temperature, default: 20 °C (68 °F). |
| | | 0x10 | Write multiple | | |
| Room floor max limit temperature channel 3 | 40041 | 0x06 | Write | Analogue | Signed 16 bit integer. Range: min temperature or 20 - 35, default: 26 °C (78.8 °F). |
| | | 0x10 | Write multiple | | |
| Room floor min limit temperature channel 3 | 40042 | 0x06 | Write | Analogue | Signed 16 bit integer. Range: 10 - 30 or max temperature, default: 20 °C (68 °F). |
| | | 0x10 | Write multiple | | |
| Room floor max limit temperature channel 4 | 40043 | 0x06 | Write | Analogue | Signed 16 bit integer. Range: min temperature or 20 - 35, default: 26 °C (78.8 °F). |
| | | 0x10 | Write multiple | | |
| Room floor min limit temperature channel 4 | 40044 | 0x06 | Write | Analogue | Signed 16 bit integer. Range: 10 - 30 or max temperature, default: 20 °C (68 °F). |
| | | 0x10 | Write multiple | | |
| Room floor max limit temperature channel 5 | 40045 | 0x06 | Write | Analogue | Signed 16 bit integer. Range: min temperature or 20 - 35, default: 26 °C (78.8 °F). |
| | | 0x10 | Write multiple | | |
| Room floor min limit temperature channel 5 | 40046 | 0x06 | Write | Analogue | Signed 16 bit integer. Range: 10 - 30 or max temperature, default: 20 °C (68 °F). |
| | | 0x10 | Write multiple | | |
| Room floor max limit temperature channel 6 | 40047 | 0x06 | Write | Analogue | Signed 16 bit integer. Range: min temperature or 20 - 35, default: 26 °C (78.8 °F). |
| | | 0x10 | Write multiple | | |
| Room floor min limit temperature channel 6 | 40048 | 0x06 | Write | Analogue | Signed 16 bit integer. Range: 10 - 30 or max temperature, default: 20 °C (68 °F). |
| | | 0x10 | Write multiple | | |
| Room floor max limit temperature channel 7 | 40049 | 0x06 | Write | Analogue | Signed 16 bit integer. Range: min temperature or 20 - 35, default: 26 °C (78.8 °F). |
| | | 0x10 | Write multiple | | |
| Room floor min limit temperature channel 7 | 40050 | 0x06 | Write | Analogue | Signed 16 bit integer. Range: 10 - 30 or max temperature, default: 20 °C (68 °F). |
| | | 0x10 | Write multiple | | |
| Room floor max limit temperature channel 8 | 40051 | 0x06 | Write | Analogue | Signed 16 bit integer. Range: min temperature or 20 - 35, default: 26 °C (78.8 °F). |
| | | 0x10 | Write multiple | | |
| Room floor min limit temperature channel 8 | 40052 | 0x06 | Write | Analogue | Signed 16 bit integer. Range: 10 - 30 or max temperature, default: 20 °C (68 °F). |
| | | 0x10 | Write multiple | | |

| Display name | Register address (decimal) | Function code (Hex) | Type | Analogue/ Binary | Coding |
|---|---|---|---|---|---|
| Room floor max limit temperature channel 9 | 40053 | 0x06 | Write | Analogue | Signed 16 bit integer. Range: min temperature or 20 - 35, default: 26 °C (78.8 °F). |
| | | 0x10 | Write multiple | | |
| Room floor min limit temperature channel 9 | 40054 | 0x06 | Write | Analogue | Signed 16 bit integer. Range: 10 - 30 or max temperature, default: 20 °C (68 °F). |
| | | 0x10 | Write multiple | | |
| Room floor max limit temperature channel 10 | 40055 | 0x06 | Write | Analogue | Signed 16 bit integer. Range: min temperature or 20 - 35, default: 26 °C (78.8 °F). |
| | | 0x10 | Write multiple | | |
| Room floor min limit temperature channel 10 | 40056 | 0x06 | Write | Analogue | Signed 16 bit integer. Range: 10 - 30 or max temperature, default: 20 °C (68 °F). |
| | | 0x10 | Write multiple | | |
| Room floor max limit temperature channel 11 | 40057 | 0x06 | Write | Analogue | Signed 16 bit integer. Range: min temperature or 20 - 35, default: 26 °C (78.8 °F). |
| | | 0x10 | Write multiple | | |
| Room floor min limit temperature channel 11 | 40058 | 0x06 | Write | Analogue | Signed 16 bit integer. Range: 10 - 30 or max temperature, default: 20 °C (68 °F). |
| | | 0x10 | Write multiple | | |
| Room floor max limit temperature channel 12 | 40059 | 0x06 | Write | Analogue | Signed 16 bit integer. Range: min temperature or 20 - 35, default: 26 °C (78.8 °F). |
| | | 0x10 | Write multiple | | |
| Room floor min limit temperature channel 12 | 40060 | 0x06 | Write | Analogue | Signed 16 bit integer. Range: 10 - 30 or max temperature, default: 20 °C (68 °F). |
| | | 0x10 | Write multiple | | |
| Comfort setting channel 1 | 40061 | 0x03 | Read | Analogue | Signed 16 bit integer. Range: 0 = off, 5 - 12 = 5 - 12 % default: 0. |
| | | 0x06 | Write | | |
| | | 0x10 | Write multiple | | |
| Comfort setting channel 2 | 40062 | 0x03 | Read | Analogue | Signed 16 bit integer. Range: 0 = off, 5 - 12 = 5 - 12 % default: 0. |
| | | 0x06 | Write | | |
| | | 0x10 | Write multiple | | |
| Comfort setting channel 3 | 40063 | 0x03 | Read | Analogue | Signed 16 bit integer. Range: 0 = off, 5 - 12 = 5 - 12 % default: 0. |
| | | 0x06 | Write | | |
| | | 0x10 | Write multiple | | |
| Comfort setting channel 4 | 40064 | 0x03 | Read | Analogue | Signed 16 bit integer. Range: 0 = off, 5 - 12 = 5 - 12 % default: 0. |
| | | 0x06 | Write | | |
| | | 0x10 | Write multiple | | |
| Comfort setting channel 5 | 40065 | 0x03 | Read | Analogue | Signed 16 bit integer. Range: 0 = off, 5 - 12 = 5 - 12 % default: 0. |
| | | 0x06 | Write | | |
| | | 0x10 | Write multiple | | |
| Comfort setting channel 6 | 40066 | 0x03 | Read | Analogue | Signed 16 bit integer. Range: 0 = off, 5 - 12 = 5 - 12 % default: 0. |
| | | 0x06 | Write | | |
| | | 0x10 | Write multiple | | |
| Comfort setting channel 7 | 40067 | 0x03 | Read | Analogue | Signed 16 bit integer. Range: 0 = off, 5 - 12 = 5 - 12 % default: 0. |
| | | 0x06 | Write | | |
| | | 0x10 | Write multiple | | |
| Comfort setting channel 8 | 40068 | 0x03 | Read | Analogue | Signed 16 bit integer. Range: 0 = off, 5 - 12 = 5 - 12 % default: 0. |
| | | 0x06 | Write | | |
| | | 0x10 | Write multiple | | |
| Comfort setting channel 9 | 40069 | 0x03 | Read | Analogue | Signed 16 bit integer. Range: 0 = off, 5 - 12 = 5 - 12 % default: 0. |
| | | 0x06 | Write | | |
| | | 0x10 | Write multiple | | |
| Comfort setting channel 10 | 40070 | 0x03 | Read | Analogue | Signed 16 bit integer. Range: 0 = off, 5 - 12 = 5 - 12 % default: 0. |
| | | 0x06 | Write | | |
| | | 0x10 | Write multiple | | |

| Display name | Register address (decimal) | Function code (Hex) | Type | Analogue/ Binary | Coding |
|---|---|---|---|---|---|
| Comfort setting channel 11 | 40071 | 0x03 | Read | Analogue | Signed 16 bit integer.<br>Range: 0 = off, 5 - 12 = 5 - 12 %<br>default: 0. |
| | | 0x06 | Write | | |
| | | 0x10 | Write multiple | | |
| Comfort setting channel 12 | 40072 | 0x03 | Read | Analogue | Signed 16 bit integer.<br>Range: 0 = off, 5 - 12 = 5 - 12 %<br>default: 0. |
| | | 0x06 | Write | | |
| | | 0x10 | Write multiple | | |
| Heating/cooling offset temperature | 40073 | 0x03 | Read | Analogue | Signed 16 bit integer.<br>Range: 0 - 5 °C<br>default: 2 °C (2 °F). |
| | | 0x06 | Write | | |
| | | 0x10 | Write multiple | | |

# 4 Troubleshooting

In case of problems and alarms in the Uponor Smatrix Base PRO system, check the table below.

*Otherwise see Uponor Smatrix Base PRO installation and operation manual section 16 for more information.*

| Problem | Indication | Probable cause | Solutions |
|---|---|---|---|
| No communication between controller and BMS | Changes done in BMS are not carried out in the controller and/or thermostats | Missing U_BMS.txt file | Download the file from the Uponor website and copy it onto the microSD card |
| | | U_BMS.txt file not configured correctly | Configure the U_BMS.txt file correctly<br>*See section 1.2 Preparation step 3 for more information* |
| | | Incorrect software version in the controller | Install the correct software in the controller<br>*See section 1.2 Preparation step 2 for more information* |
| | | Disconnected communication cable | Check all wiring to make sure all communication cables are connected correctly |
| Slow communication, or high latency, between Uponor system and BMS | It takes a long time for Uponor parameters to change in the after being sent from the BMS | The Uponor Smatrix Base PRO Interface I-147 is installed and registered in the system | Unregister and disconnect the interface from the Uponor system |
| Wrong parameters changed in Uponor system compared to BMS settings | Wrong parameters are changed in the Uponor system after new values are sent from the BMS | | |

**Uponor GmbH**

Industriestraße 56,
D-97437 Hassfurt, Germany

1090255 03_2018
Production: Uponor / MRY



**www.uponor.com**