

UPSI Generation 2 — Modbus RTU Specification

Version 2.0 · Bicker Elektronik GmbH · 13.07.2026

Table of Contents

- 1. Revision List
- 2. Availability
- 3. Connection Parameters
- 4. Register Addressing
- 5. Timing — read this before you start
- 6. Function Codes
- 7. Exception Codes and Silence
- 8. Byte Order
- 9. Register Map
- 10. Appendix A — Known Limitations
- 11. Appendix B — Changes since V1.1

1. Revision List

Rev.	Description
V1.0	Creating the document (20.06.2024)
V1.1	Register 72, function 6 corrected
V2.0	13.07.2026 — Register map corrected against the firmware. Added the chapters a Modbus master actually needs: register addressing, timing, function codes, exception codes, byte order, limits and device availability. See Appendix B.

2. Availability

Modbus RTU is available **only** on the following series, and **only** when an RS485 communication module is fitted:

Series	Modbus RTU	Firmware
UPSI-2415	yes (with RS485 module)	≥ v1.4.0

Series	Modbus RTU	Firmware
UPSI-1208R2 / UPSI-2406R2	yes (with RS485 module)	≥ v1.7.0
UPSI-1208 / UPSI-2406 (R1)	no — no RS485 interface	—
UPSI-2412	no — no RS485 interface	—

To check whether a device has an RS485 module, read command **0x7B** of the UPSGen binary protocol. It returns **1** for RS485.

The Modbus-related configuration parameters (slave address, termination, byte order) also exist on UPSI-1208/2406 (R1) and UPSI-2412. They can be written and read back, but **they have no effect** on those devices — there is no bus.

3. Connection Parameters

RS485 half duplex

Parameter	Factory default	Range
Baud rate	38400 bps	2400 · 4800 · 9600 · 19200 · 38400 · 57600 · 115200
Data length	8	8
Stop bits	1	1
Parity	none	none
Slave address	11	1 ... 127

Baud rates 57600 and 115200 are available from firmware UPSI-2415 v1.4.0 and UPSI-1208R2/2406R2 v1.7.0 onwards.

Note: raising the baud rate does **not** increase the achievable polling rate — see chapter 5.

Changing the settings

Baud rate, slave address, bus termination and byte order are **not** configurable over Modbus. They are set with the UPSGen binary protocol (over USB CDC) via the following parameters:

Parameter	Setting
0x0E	Baud rate

Parameter	Setting
0x0F	Slave address (takes effect immediately)
0x10	120 Ω bus termination (switched by the Is Active flag)
0x11	Byte order (takes effect after a restart)

See the *UPSI Generation 2 — Communication Protocol* document.

Broadcast

Broadcast (slave address 0) is not supported. A broadcast frame is ignored completely — it is neither executed nor answered.

4. Register Addressing

⚠ The register numbers in this document are 1-based

The **address on the wire is 0-based**. To read register 1, the master must send address 0x0000.

wire address = register number - 1

Register (this document)	Wire address
1 — Modbus RTU Version	0x0000
66 — CPU Core Temperature	0x0041
72 — Output Enabled (writable)	0x0047
75 — Shutdown Flag (writable)	0x004A
99 — Battery Time to Empty	0x0062

This is the most common source of "the device does not answer" — many Modbus masters display 1-based register numbers but transmit them 0-based, others do not. Check what your master actually puts on the wire.

5. Timing — read this before you start

⚠ The device answers no earlier than 32 ms after the request

This delay is **fixed and independent of the baud rate**. It gives the master time to switch its RS485 driver.

Set the master's response timeout to at least 100 ms (200 ms is safer). A master configured with a 20 ms timeout — a common default — will **never** see an answer.

Property	Value
Response delay	≥ 32 ms after the last request byte (plus a small jitter from the device's main loop)
Recommended master response timeout	≥ 100 ms
Maximum polling rate	approx. 25–30 requests/s — independent of the baud rate
Driver-enable hold time after the answer	16 ... 35 bit times (0.3 ms at 115200 up to 6.7 ms at 2400). The master must not transmit during this window
Frame stall reset	250 ms

No t3.5 inter-frame timeout

The device does **not** detect frame boundaries by bus silence. It derives the frame length from the function code (8 bytes for every supported request).

Consequence: the master must implement retries. On a multi-slave bus, a data or CRC byte of a *foreign* frame can coincidentally match this device's slave address. The parser then starts a phantom frame, collects seven more bytes, fails the CRC check and falls silent. This is self-healing, but it can consume the first bytes of the *next* request addressed to this device — that request is then lost and must be repeated.

6. Function Codes

FC	Name	Supported	Notes
03	Read Holding Registers	yes	
04	Read Input Registers	yes	Identical to FC 03. There is no separate input register bank — both function codes address the same 99 registers

FC	Name	Supported	Notes
06	Write Single Register	yes	Only registers 72 and 75 are writable
16	Write Multiple Registers	no	⚠ The device does not answer at all — see the warning below
01, 02, 05, 08, 15, 23	—	no	Exception 0x01

Limits for FC 03 / 04

Condition	Result
<code>quantity = 0</code> or <code>quantity > 125</code>	Exception 0x03
<code>start address + quantity > 99</code>	Exception 0x02
Maximum readable in one request	99 registers (address 0x0000 , quantity 99 → 198 data bytes)

⚠ Function code 16 produces no reply

Writing multiple registers is **not supported**, and the device answers with **silence**, not with exception **0x01**. The master will run into its timeout.

Write registers 72 and 75 **individually with FC 06**.

7. Exception Codes and Silence

The device sends exactly **three** exception codes:

Code	Name	Cause
0x01	ILLEGAL_FUNCTION	Unsupported function code
0x02	ILLEGAL_DATA_ADDRESS	Reading beyond register 99 · Writing to a read-only register
0x03	ILLEGAL_DATA_VALUE	<code>quantity = 0</code> or <code>quantity > 125</code>

Exception frame: `[slave ID] [FC | 0x80] [code] [CRC lo] [CRC hi]` — 5 bytes.

When the device stays silent

Silence is **not** an error on the device's side in these cases — it is the specified behaviour:

Situation	Reaction
CRC mismatch	no answer (as per Modbus specification)
Frame addressed to a different slave	no answer
Function code 16	no answer (see chapter 6)

CRC: CRC-16/Modbus (polynomial 0xA001 reflected, initial value 0xFFFF, no final XOR), transmitted **low byte first**.

8. Byte Order

The byte order of the register data is configurable through **parameter 0x11** of the UPSGen binary protocol.

Value	Byte order	Meaning
0	Big Endian	Standard Modbus — most significant byte first. Factory default on UPSI-2415 and UPSI-1208R2/2406R2
1	Little Endian	Numeric registers are byte-swapped. Deliberate, but not compliant with the Modbus specification

With the factory default (Big Endian) **no special handling is required** — the device behaves like any standard-compliant Modbus slave.

Strings are always transmitted MSB first

Character registers (device name, serial number, manufacturer, versions, battery name) are **always** sent most-significant-byte first, **regardless of the byte order parameter**. ASCII therefore always appears in reading order.

Only the **numeric** registers are affected by the parameter.

A change of the byte order takes effect only after a restart of the device. The Modbus stack is not re-initialised at runtime.

9. Register Map

Register numbers below are 1-based — see chapter 4.

Function column: **3 / 4** = readable, **6** = writable.

Register	Length	Data type	Description	Unit / values	Function
1 – 5	5	Char[]	Modbus RTU Version	ASCII, fixed string "1.0.0"	3 / 4

Register	Length	Data type	Description	Unit / values	Function
6 – 10	5	Char[]	UPS Core Version	ASCII	3 / 4
11 – 26	16	Char[]	Device Name	ASCII	3 / 4
27 – 39	13	Char[]	Serial Number	ASCII	3 / 4
40 – 55	16	Char[]	Manufacturer	ASCII	3 / 4
56 – 60	5	Char[]	Firmware Version	ASCII	3 / 4
61 – 65	5	Char[]	PCB Version	ASCII	3 / 4
66	1	Int16	CPU Core Temperature	°C (–128 ... +127)	3 / 4
67	1	Uint16	UPS Error Code (LED flash code)	see table below	3 / 4
68	1	Uint16 (bit field)	UPS Status Register	see table below	3 / 4
69	1	Uint16	Input Enabled	0 = disabled, 1 = enabled	3 / 4
70	1	Uint16	Input Voltage	mV	3 / 4
71	1	Uint16	Input Current	mA	3 / 4
72	1	Uint16	Output Enabled	see below — read and write differ	3 / 4 / 6
73	1	Uint16	Output Voltage	mV	3 / 4
74	1	Uint16	Output Current	mA	3 / 4
75	1	Uint16	Shutdown Flag	see below	3 / 4 / 6
76 – 91	16	Char[]	Battery Name	ASCII	3 / 4
92	1	Uint16	Battery Voltage	mV	3 / 4
93	1	Int16	Battery Current	mA — negative = discharging	3 / 4

Register	Length	Data type	Description	Unit / values	Function
94	1	UInt16	Battery State of Charge	% (0 ... 100)	3 / 4
95	1	UInt16	Battery State of Health	not implemented — always returns 255	3 / 4
96	1	UInt16	Battery Cycle Count	cycles	3 / 4
97	1	Int16	Battery Temperature	°C · -128 = no sensor connected	3 / 4
98	1	UInt16	Battery Time to Full	SUC = seconds, other packs = minutes · 65535 = not charging	3 / 4
99	1	UInt16	Battery Time to Empty	SUC = seconds, other packs = minutes · 65535 = not discharging	3 / 4

Register 99 is the last register. Reading beyond it returns exception **0x02**.

Character blocks are **null-padded** up to their declared length. A device name shorter than 32 characters is filled with **0x00** bytes.

Register 67 — UPS Error Code

Value	Meaning
0	Power off
1	UPS powered up from battery
2	Charging the energy storage before enabling the output
3	UPS in shutdown state
4	UPS in restart state
11	Input power fail — UPS in backup mode
12	Output over current or internal error
21	No energy storage detected, or wrong type
22	Energy storage over voltage

Value	Meaning
23	Energy storage over current
24	Energy storage temperature too low or too high
25	Battery start refused — state of charge too low
99	Device OK

Value 12 is ambiguous. Over current and internal error return the same code and cannot be distinguished through this register. Use bit 5 of the status register (68) to identify an over-current condition.

Register 68 — UPS Status Register

Bit	Mask	Meaning
0	0x0001	Charging
1	0x0002	Discharging
2	0x0004	Input power present
3	0x0008	Energy storage present
4	0x0010	Shutdown received
5	0x0020	Over current
6 – 15	—	reserved

Register 72 — Output Enabled

⚠ Reading and writing have different meanings

Reading returns a simple on/off state. Writing performs a **power cycle with timing** — it is not a plain on/off bit.

Reading (FC 3 / 4):

Value	Meaning
0	Output disabled
1	Output enabled

Writing (FC 6) — the 16-bit value is split into two bytes:

Byte	Meaning
High byte	Pre-delay in seconds — the device waits this long before switching the output off
Low byte	<code>0x00</code> = switch the output off permanently · <code>0xFF</code> = switch the output on permanently · <code>0x01–0xFE</code> = switch the output off and back on after this many seconds

Examples:

Value	Effect
<code>0x0000</code>	Output off immediately, permanently
<code>0x00FF</code>	Output on immediately, permanently
<code>0x000A</code>	Output off immediately, back on after 10 s
<code>0x0A05</code>	Wait 10 s, then output off, back on after 5 s

Register 75 — Shutdown Flag

Used by an external program that initiates the shutdown of the operating system. The program informs the UPS by setting this flag.

Value	Meaning
0	not set
1	set

⚠ The write is only accepted while the UPS is on battery

If the UPS is running on mains power, the firmware forces the flag to **0** — but still answers with a **positive echo**. The master receives a confirmation for a command that was discarded.

Always read the register back to learn the actual state.

Appendix A — Known Limitations

#	Limitation	Consequence for the master
1	Response delay of 32 ms , independent of the baud rate	Master response timeout must be ≥ 100 ms. Polling rate is capped at ~25–30 requests/s
2	No t3.5 inter-frame detection	The master must implement retries
3	FC 16 answers with silence , not with exception <code>0x01</code>	Use FC 06 for the two writable registers
4	No broadcast (address 0)	Address each device individually
5	FC 03 and FC 04 are identical	There is no separate input register bank
6	Register 95 (State of Health) always returns 255	The value is not implemented
7	Register 75 confirms a discarded write	Read the register back
8	Byte order change requires a restart	Set it once during commissioning

Appendix B — Changes since V1.1

Corrections to the register map

Item	Before	Now
Connection parameters, baud rate	standard 19200 bps , range garbled	standard 38400 bps , range 2400 ... 115200
Register 67	value 25 missing ; value 12 described as over current only	25 added ("battery start refused, SOC too low"); 12 documented as over current or internal error
Register 97	Int16, °C	added: -128 = no sensor connected
Register 95	"255 indicates not available"	not implemented — always returns 255

Item	Before	Now
Register 75	"0 = not set, 1 = set"	added: the write is only accepted on battery , and a discarded write still returns a positive echo
Register 72	write format hard to read	reformatted; the asymmetric read/write semantics are now stated explicitly, with examples
Register 66	Int16	effective range -128 ... +127 °C
Register 1 – 5	no value given	returns the fixed string "1.0.0"
Spelling	"Modus RTU Version", "Batter State of Charge/Health", "temperature to low"	corrected

New chapters

Everything below was missing from V1.1 and is required to implement a working master:

- **Chapter 2** — which devices have Modbus at all.
- **Chapter 4** — 1-based register numbers vs. 0-based wire addresses.
- **Chapter 5** — the **32 ms response delay**, the recommended master timeout, the achievable polling rate, and the absence of t3.5 framing.
- **Chapter 6** — which function codes are supported, that FC 03 and FC 04 are identical, and that **FC 16 answers with silence**.
- **Chapter 7** — the three exception codes, and the cases in which the device stays silent. CRC specification.
- **Chapter 8** — the byte order parameter, its series-dependent default, and the fact that strings are always MSB first.
- Quantity limits, slave address range and the absence of broadcast (chapters 3 and 6).