

Valtoris.

Integration Guide: Polling Growatt Inverters via Modbus RTU

Bypassing Cloud Latency and Resolving the 32-Bit Integer Byte-Swapping Trap

Document No:	AN-2026-07
Product Series:	Valtoris VT-DTU500 / Edge Gateways
Target Devices:	Growatt MIN, MAC, MAX Series Inverters
Date:	June 2026 (Rev. 1.1)

1. Executive Summary

Growatt solar inverters are highly reliable, but their default data logging relies entirely on external cloud servers via the ShineWiFi dongle. For industrial SCADA systems, zero-export controllers, or advanced home automation, engineers require real-time, local, sub-second access to the inverter's metrics without exposing the OT network to the internet.

Beneath the dongle port, almost all Growatt models expose a standard RS485 physical layer executing Modbus RTU. This document outlines the required wiring and provides solutions for the common U32 data mapping issues.

2. Universal RS485 Pinout

Depending on your inverter model, the COM port is either an RJ45 jack or a circular connector. To establish a serial link, connect three wires to your RS485 Master device.

Growatt COM Port (RJ45 / SYS)	RS485 Master Terminal	Signal Function
Pin 3 (or RS485A)	A+ (Data+)	Positive Data Line
Pin 4 (or RS485B)	B- (Data-)	Negative Data Line
Pin 5 (GND)	GND	Signal Ground (Crucial for preventing CRC noise)

The 32-Bit Integer (U32) Trap

Notice that Power and Energy values are 32-bit unsigned integers (U32). Because Modbus RTU is inherently a 16-bit protocol, a single U32 metric spans across two consecutive registers. If read sequentially without proper byte-swapping (Endianness recombination), the resulting decimals will be completely erroneous.

3. Modbus Register Map (Key Metrics)

Configure your polling node to **Baud Rate: 9600, 8 Data Bits, 1 Stop Bit, Parity: None**. Growatt's default Slave ID is typically 1. Polling requires **Input Registers (Function Code 0x04)**.

Register (Dec)	Hex Address	Data Format	Description	Multiplier
30001	0x0000	U16	Inverter Status (0=Wait, 1=Normal, 3=Fault)	N/A
30002	0x0001	U32 (2 Regs)	Input Power (PV Power)	0.1 W
30004	0x0003	U16	PV1 Voltage	0.1 V
30036	0x0023	U32 (2 Regs)	Current Active Power (Grid Output)	0.1 W
30054	0x0035	U32 (2 Regs)	Total Energy Generated (Lifetime)	0.1 kWh

4. Deep Dive: Decoding the U32 Hex Payload

Let us analyze a raw Modbus request targeting **Current Active Power (Register 30036 / 0x0023)**. Assume the inverter is currently generating 10,000 Watts.

```
Master Request (Tx): 01 04 00 23 00 02 81 C0
```

Breakdown:

01 : Slave ID

04 : Function Code (Read Input Registers)

00 23 : Starting Address (35 in Dec)

00 02 : Quantity to read (2 registers = 4 bytes)

```
Inverter Response (Rx): 01 04 04 00 01 86 A0 XX XX
```

Data Payload: 00 01 86 A0 (Hex)

When combined using Big-Endian structure, 0x000186A0 converts to 100000 in decimal. Applying the 0.1 multiplier yields **10,000 W**.

5. Integration Architecture: Software vs. Hardware

Deploying this logic locally requires a choice between maintaining custom parsing scripts or utilizing normalized edge hardware.

A. Software Parsing (Python)

When using an IPC or Raspberry Pi with a generic USB-RS485 adapter, your code must handle the U32 concatenation manually.

B. Hardware Normalization (Edge Gateway)

For commercial deployments, utilizing a dedicated protocol bridge removes the scripting layer.

- **Native U32 Support:** The Valtoris gateway firmware natively processes the Big-Endian U32 recombination and applies the 0.1 scaling factor automatically.

```
import struct

# Example: Growatt returns High Word
# first
high_word = 0x0001
low_word = 0x86A0

# Pack as two unsigned shorts (H),
# unpack as a 32-bit unsigned int (I)
val = struct.unpack('>I',
    struct.pack('>HH', high_word,
    low_word))[0]

power_watts = val * 0.1
print(f"Grid Power: {power_watts} W")
```

Note: OS-level serial polling across multiple inverter daisy-chains frequently encounters timing collisions requiring robust retry logic.

- **Cloud Publishing:** Clean JSON payloads (e.g., {"ActivePower_W": 10000.0}) are published directly to an MQTT broker via Ethernet or LTE.
- **Reliability:** Hardware watchdogs ensure continuous 24/7 telemetry streams without script maintenance.

Scaling Your Solar Deployments

To eliminate Python maintenance and secure deterministic Modbus polling across commercial sites, view the specifications for the [Valtoris Edge Gateway Series](#).