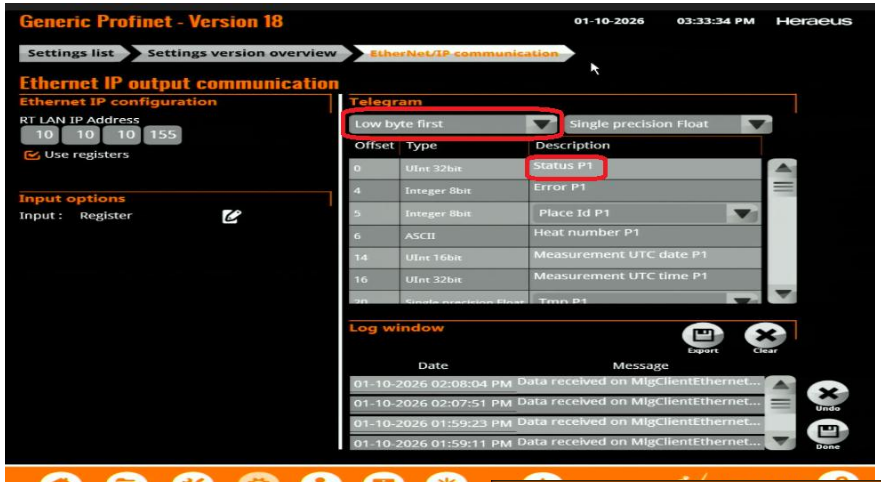
SensorLab Status Word Endianness Inconsistency

Endianness inconsistency:

- **Output Registers:** status words are always big-endian, regardless of the “byte-order” setting.
  - Status always comes through high-byte first.
- **Custom Telegrams:** status words are always little-endian (low-byte-first); there’s no option to make them big-endian.
  - Status always comes through low-byte first.
- **Result:** the two paths handle status-word byte order in opposite, non-configurable ways.
  - Depending on you selection, Status must be handled differently in the PLC.



Generic Profinet - Version 18 01-10-2026 03:33:34 PM Heraeus

Settings list > Settings version overview > EtherNet/IP communication

Ethernet IP output communication

Ethernet IP configuration

RT LAN IP Address: 10 10 10 155

☑ Use registers

Input options

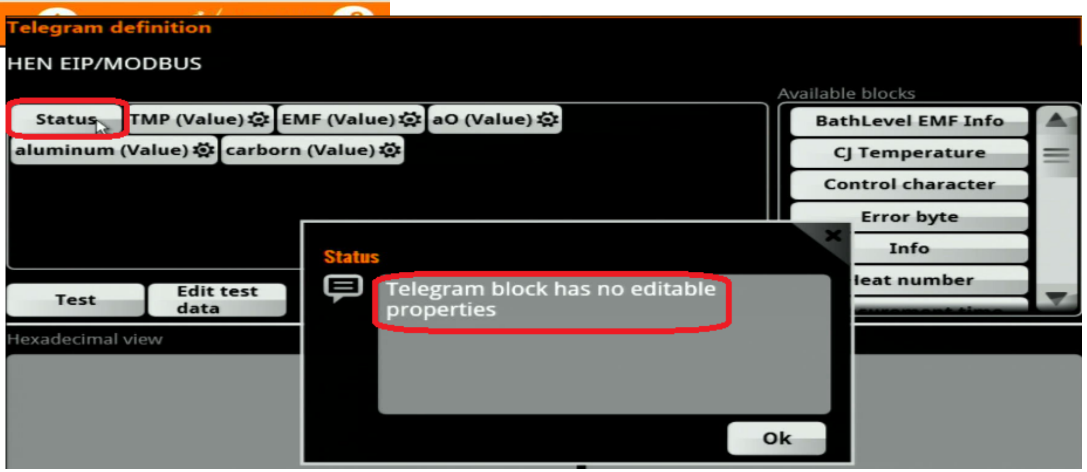
Input : Register

Telegram: Low byte first | Single precision Float

Offset	Type	Description
0	UInt 32bit	Status P1
4	Integer 8bit	Error P1
5	Integer 8bit	Place Id P1
6	ASCII	Heat number P1
14	UInt 16bit	Measurement UTC date P1
16	UInt 32bit	Measurement UTC time P1
20	Single precision Float	Tmp P1

Log window

```
Date                   Message
01-10-2026 02:08:04 PM Data received on MlgClientEthernet...
01-10-2026 02:07:51 PM Data received on MlgClientEthernet...
01-10-2026 01:59:23 PM Data received on MlgClientEthernet...
01-10-2026 01:59:11 PM Data received on MlgClientEthernet...
```



Telegram definition

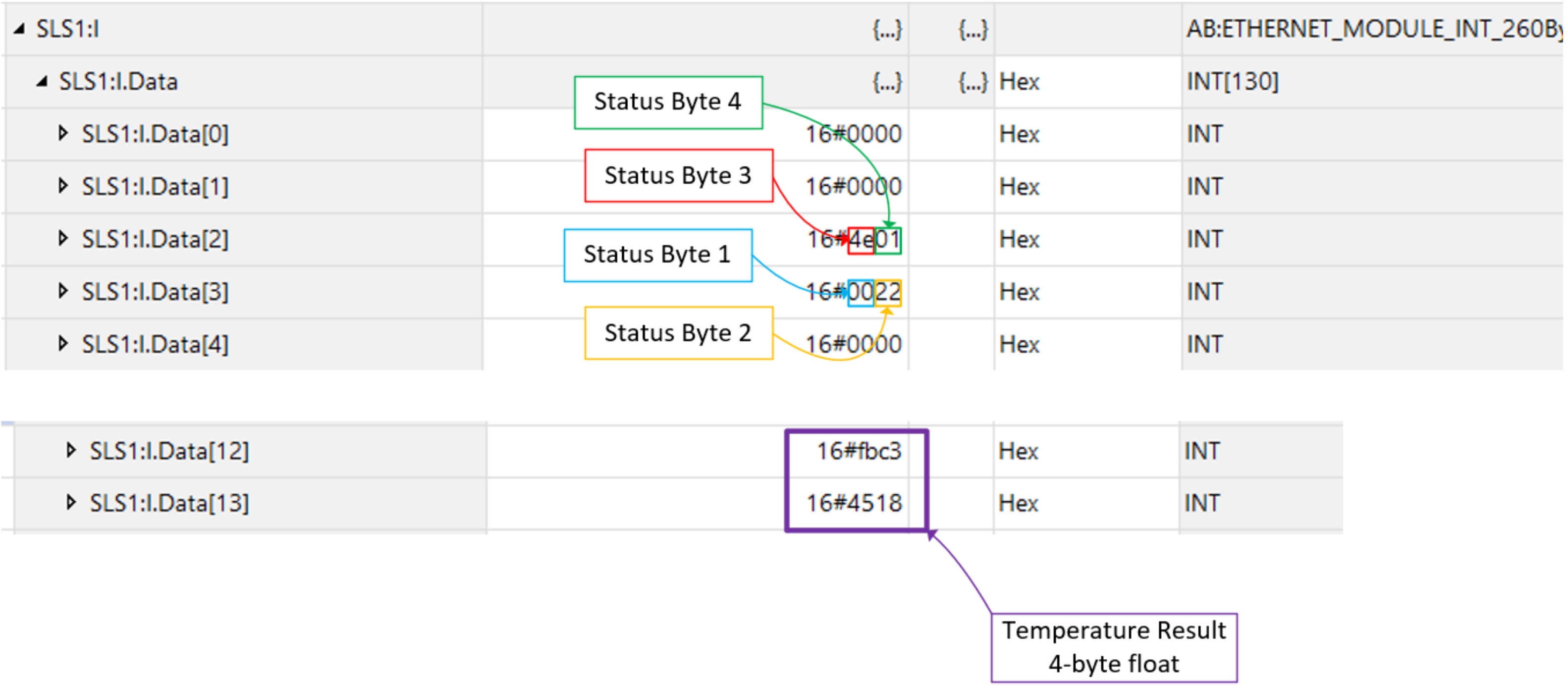
HEN EIP/MODBUS

Status | TMP (Value) | EMF (Value) | aO (Value) | aluminum (Value) | carborn (Value)

Status: Telegram block has no editable properties

PLC Data – SLS Output Register via EIP Generic Module Connection (configured Low-byte first)

- The result fields (e.g., Temperature 4-byte float, etc) decode correctly in the PLC and the raw words show little-endian byte order.
- However, the status word/bytes **still arrive high-byte-first**, i.e., their byte order doesn't follow the low-byte-first setting.

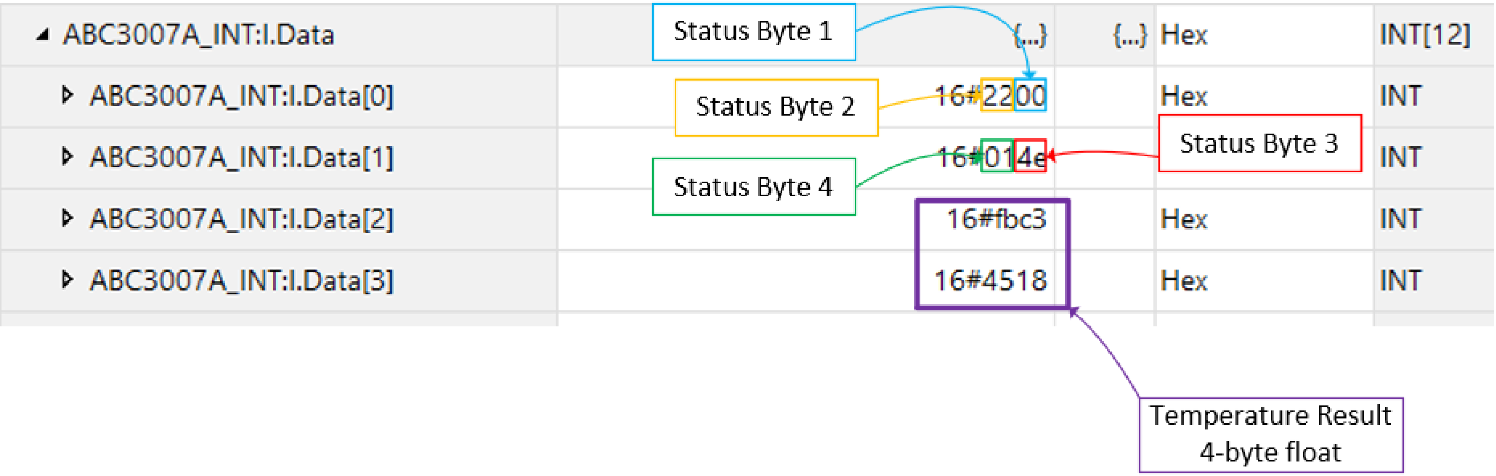


◢ SLS1:I	{...}	{...}		AB:ETHERNET_MODULE_INT_260By
◢ SLS1:I.Data	{...}	{...}	Hex	INT[130]
▷ SLS1:I.Data[0]	16#0000		Hex	INT
▷ SLS1:I.Data[1]	16#0000		Hex	INT
▷ SLS1:I.Data[2]	16#4e01		Hex	INT
▷ SLS1:I.Data[3]	16#0022		Hex	INT
▷ SLS1:I.Data[4]	16#0000		Hex	INT

▷ SLS1:I.Data[12]	16#fbc3	Hex	INT
▷ SLS1:I.Data[13]	16#4518	Hex	INT

PLC Data - SLS Custom Telegram via serial → EIP (Anybus ABC3007-A)

- Both the result fields (e.g., Temperature 4-byte floats, etc) and the status bytes decode correctly in the PLC as low-byte-first; the raw words show little-endian order.
- Note: the telegram editor **does not offer a big-endian option for status words**—only result fields allow endianness selection.



◢ ABC3007A_INT:I.Data	{...}	{...}	Hex	INT[12]
▷ ABC3007A_INT:I.Data[0]	16#2200		Hex	INT
▷ ABC3007A_INT:I.Data[1]	16#014e		Hex	INT
▷ ABC3007A_INT:I.Data[2]	16#fbc3		Hex	INT
▷ ABC3007A_INT:I.Data[3]	16#4518		Hex	INT

Status Byte 1

Status Byte 2

Status Byte 3

Status Byte 4

Temperature Result
4-byte float

Summary

- Endianness rules seem to be inconsistent for Status vs Result fields across SLS comms methods (this could also apply to Error words).
- This introduces an extra ‘gotcha’ during commissioning:
  - Status must be swapped/handled differently than the floats, increasing the chance of integration errors.

- If this inconsistency can’t or shouldn’t be corrected in the current instrument, it should be documented it as a known issue and to ensure the next SensorLab version applies endianness consistently across all fields (or provides an explicit configuration option for Status/Error).