

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 your selection, Status must be handled differently in the PLC.



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_260B
▲ SLS1:I.Data	{...}	{...}	Hex	INT[130]
▸ SLS1:I.Data[0]	Status Byte 4	16#0000	Hex	INT
▸ SLS1:I.Data[1]	Status Byte 3	16#0000	Hex	INT
▸ SLS1:I.Data[2]	Status Byte 1	16#4e01	Hex	INT
▸ SLS1:I.Data[3]		16#0022	Hex	INT
▸ SLS1:I.Data[4]	Status Byte 2	16#0000	Hex	INT
▸ SLS1:I.Data[12]		16#fbc3	Hex	INT
▸ SLS1:I.Data[13]		16#4518	Hex	INT

Temperature Result
4-byte float

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	Status Byte 1	{...}	{...}	Hex	INT[12]
▶ ABC3007A_INT:I.Data[0]	Status Byte 2	16#2200		Hex	INT
▶ ABC3007A_INT:I.Data[1]	Status Byte 4	16#014e	Status Byte 3		INT
▶ ABC3007A_INT:I.Data[2]		16#fbc3		Hex	INT
▶ ABC3007A_INT:I.Data[3]		16#4518		Hex	INT

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 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).