

Allocation and purpose of the extra bytes that get setup when using the Generic Modbus DTM

Goals and Symptoms

Why do we add 2 unused bytes in the input or output when we add a io scanner line in the DTM browser/device list for M580 when using a Generic Modbus device?

Facts and Changes

M580, BMENOC, M340

Causes and Fixes

The behavior is due to the fact that it was incorrectly defined in M340 and also current M580 Version of master DTMs. (The definition of M340 was carried to M580!).

M340 and M580 both have memory constraint of 32 bit alignment. Due to this, the algorithm for DDT/DDDT for M340/M580 was written to make sure all the structures are Modulo 4.

le: All structures created from master are made multiples of 4.

In the below example although 1 register is configured and expected is 2 bytes structure, because of above algorithm remaining 2 bytes are packed with bytes which leads to 4 bytes (with 2 bytes padding at end).

The screenshot shows the configuration of a Modbus DTM. The 'Request Setting' tab is active, displaying a table with the following data:

Repetitive Rate(ms)	RD Address	RD Length	Last Value	WR Address	WR Length
60	0	1	Hold Value	0	1

The 'Data Properties: Modbus_Device_2' window shows the following attributes:

Name	Value
Name	Modbus_Device_2
Comment	
Address	
384 Address	
RW program	<input checked="" type="checkbox"/>
Alias of	
Global data	NO
Owner	Modbus_Device_2
Type	T_Modbus_Device_2
Category	<Struct>
Save	
Constant	
Used	0
Custom	
Descriptor 1	
Size	12
HMI variable	

The 'Variables' window shows the following structure:

Name	Type
Modbus_Device_2	T_Modbus_Device_2
Freshness	BOOL
Freshness_1	BOOL
Inputs	T_Modbus_Device_2_IN
Free0	ARRAY0_31 OF BYTE
Free0[0]	BYTE
Free0[1]	BYTE
Free0[2]	BYTE
Free0[3]	BYTE
Outputs	T_Modbus_Device_2_OUT
Free1	ARRAY0_31 OF BYTE
Free1[0]	BYTE
Free1[1]	BYTE
Free1[2]	BYTE
Free1[3]	BYTE

Annotations in the Variables window:

- Red arrow pointing to Free0[0] and Free0[1]: 2 hidden padding bytes are here. These insure that if the first inputs are 32bit, it meets the alignment rule.
- Red boxes around Free0[2] and Free0[3]: 2 visible but unused padding bytes are here. These insure that if the first Output is 32bit, it meets the alignment rule.
- Red boxes around Free1[0] and Free1[1]: 2 visible but unused padding bytes are here. These insure that if the next Device DDT Variable meets the alignment rule.