DTE E4 ProfiNet Notes

FOR **TESTING ONLY…NOT RESPONSIBLE FOR ERRORS OR OMISSIONS**.

WAS PUT TOGETHER AS A RUSH EXAMPLE OF HOW TO CONNECT ONLY

DTE Setup Params - Profinet

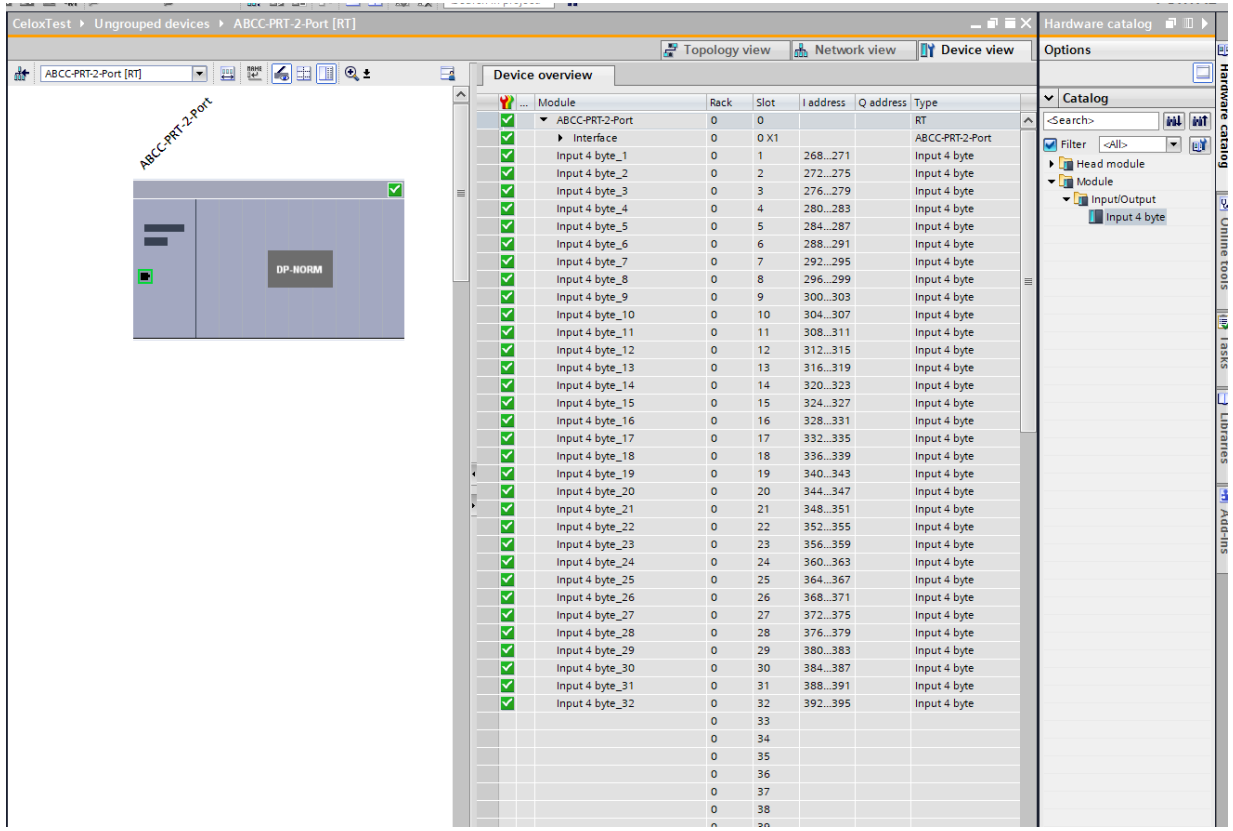
- Suggested setup parameters for DTE E4 with Profinet Comms

Table 4: List of parameters

ID	Function	Detail
5.7	Ethernet/IP or Modbus TCP or Profinet	ProfiNet
5.7.1	Telegram Bus	7 – All inclusive telegram (7 for DTE/ 6 for Celox)
5.7.2	Status Bus	Yes - The status byte is transmitted over PROFINET whenever the status changes and is placed at the beginning of the telegram payload.
5.7.3	Test Meas. Bus	Yes - Send data over Profinet when a test measurement is performed with a Checkmate instrument.
5.7.4	Enable DHCP (only for Ethernet/IP and Modbus TCP)	NA re Profinet
5.7.5	Slave IP (only for Ethernet/IP and Modbus TCP)	NA re Profinet
5.7.6	Slave Subnet Mask (only for Ethernet/IP and Modbus TCP)	NA re Profinet

Device Config Example

- For the PROFINET configuration, please add 32 instances of the 4-byte Input module



Status Bytes Data

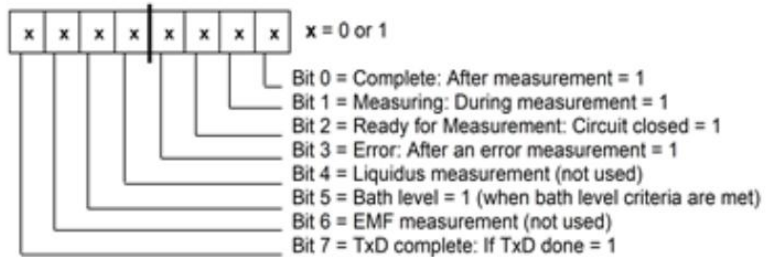
- Status Bytes Data

4.1.2 Com1 Status

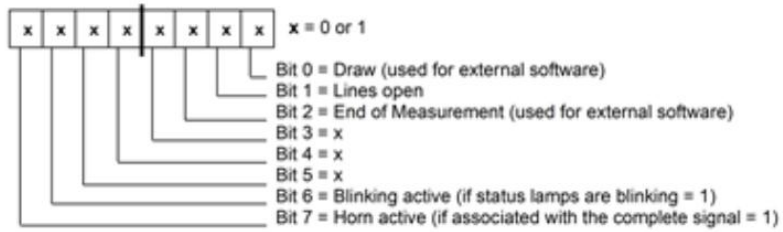
Sends the status of the instrument over the communication port. With every status change of the instrument, and when the Send status parameter is set to Yes, the data telegram begins with a four-byte status.

Configuration of the four byte data telegram:

Byte 1: status bits part 1



Byte 2: status bits part 2



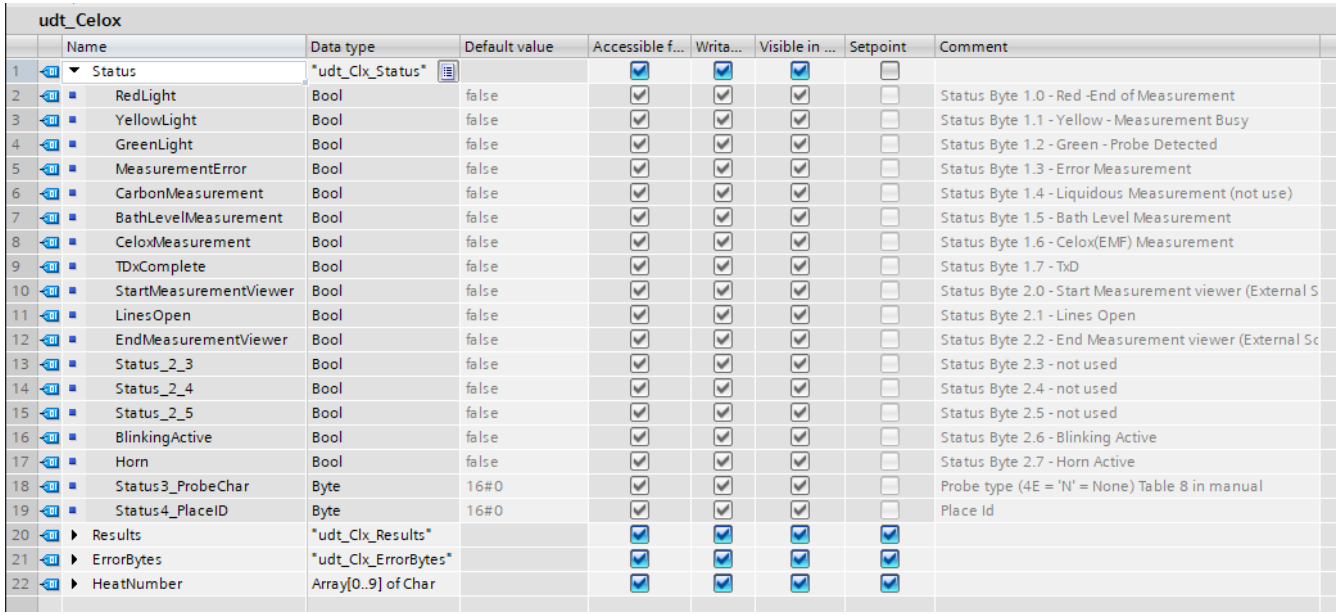
Byte 3: probe character

- N (4Ehex) = No probe attached (lines open)
- T (54hex) = Temperature probe

Byte 4: measurement place number (00 - 99)

Status Bytes Data

- Status Bytes Data
- I chose to build the Status portion on my structre like this.



udt_Celox

	Name	Data type	Default value	Accessible f...	Writa...	Visible in ...	Setpoint	Comment
1	Status	"udt_Clx_Status"		☑	☑	☑	☐	
2	RedLight	Bool	false	☑	☑	☑	☐	Status Byte 1.0 - Red -End of Measurement
3	YellowLight	Bool	false	☑	☑	☑	☐	Status Byte 1.1 - Yellow - Measurement Busy
4	GreenLight	Bool	false	☑	☑	☑	☐	Status Byte 1.2 - Green - Probe Detected
5	MeasurementError	Bool	false	☑	☑	☑	☐	Status Byte 1.3 - Error Measurement
6	CarbonMeasurement	Bool	false	☑	☑	☑	☐	Status Byte 1.4 - Liquidous Measurement (not use)
7	BathLevelMeasurement	Bool	false	☑	☑	☑	☐	Status Byte 1.5 - Bath Level Measurement
8	CeloxMeasurement	Bool	false	☑	☑	☑	☐	Status Byte 1.6 - Celox(EMF) Measurement
9	TDxComplete	Bool	false	☑	☑	☑	☐	Status Byte 1.7 - TxD
10	StartMeasurementViewer	Bool	false	☑	☑	☑	☐	Status Byte 2.0 - Start Measurement viewer (External S
11	LinesOpen	Bool	false	☑	☑	☑	☐	Status Byte 2.1 - Lines Open
12	EndMeasurementViewer	Bool	false	☑	☑	☑	☐	Status Byte 2.2 - End Measurement viewer (External Sc
13	Status_2_3	Bool	false	☑	☑	☑	☐	Status Byte 2.3 - not used
14	Status_2_4	Bool	false	☑	☑	☑	☐	Status Byte 2.4 - not used
15	Status_2_5	Bool	false	☑	☑	☑	☐	Status Byte 2.5 - not used
16	BlinkingActive	Bool	false	☑	☑	☑	☐	Status Byte 2.6 - Blinking Active
17	Horn	Bool	false	☑	☑	☑	☐	Status Byte 2.7 - Horn Active
18	Status3_ProbeChar	Byte	16#0	☑	☑	☑	☐	Probe type (4E = 'N' = None) Table 8 in manual
19	Status4_PlaceID	Byte	16#0	☑	☑	☑	☐	Place Id
20	Results	"udt_Clx_Results"		☑	☑	☑	☑	
21	ErrorBytes	"udt_Clx_ErrorBytes"		☑	☑	☑	☑	
22	HeatNumber	Array[0..9] of Char		☑	☑	☑	☑	

Error Bytes Data

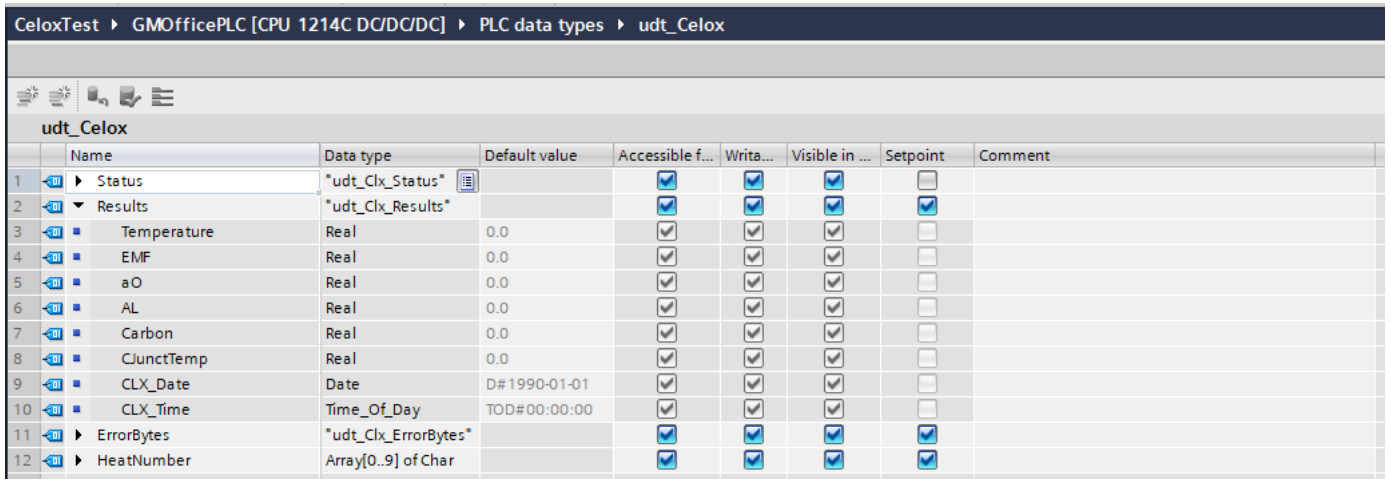
- Refer to the manual section 10.1.2
  - NOTE that you will need Manual v3.1.0 or higher or that section will not be included.
- I chose to build the Error portion of my structure like this

udt_Celox

	Name	Data type	Default value	Accessible f...	Writa...	Visible in ...	Setpoint	Comment
1	Status	"udt_Clx_Status"		☑	☑	☑	☐	
2	Results	"udt_Clx_Results"		☑	☑	☑	☑	
3	ErrorBytes	"udt_Clx_ErrorBytes"		☑	☑	☑	☑	
4	ERR_NoCJTemp	Bool	false	☑	☑	☑	☐	Error Byte 1.0 - No Cold Junction
5	ERR_TCBreak	Bool	false	☑	☑	☑	☐	Error Byte 1.1 - TC Break
6	ERR_MeasurementOutOfR...	Bool	false	☑	☑	☑	☐	Error Byte 1.2 - Measurement out of range
7	ERR_Spare_1_3	Bool	false	☑	☑	☑	☐	Error Byte 1.3 - Spare
8	ERR_RFLossWhileMeasuring	Bool	false	☑	☑	☑	☐	Error Byte 1.4 - RF Link Wireless broken during measure
9	ERR_Spare_1_5	Bool	false	☑	☑	☑	☐	Error Byte 1.5 - Spare
10	ERR_NoEvaluation	Bool	false	☑	☑	☑	☐	Error Byte 1.6 - No evaluation
11	ERR_GeneralError	Bool	false	☑	☑	☑	☐	Error Byte 1.7 - General Error
12	Application_Error	Byte	16#0	☑	☑	☑	☐	Error Byte 2 - Application error byte definition
13	ERR_WrongProduct	Bool	false	☑	☑	☑	☐	Error Byte 3.0 - Wrong Product
14	ERR_ProtocolError	Bool	false	☑	☑	☑	☐	Error Byte 3.1 - Protocol Error
15	ERR_BatteryLow	Bool	false	☑	☑	☑	☐	Error Byte 3.2 - Low Battery
16	ERR_NoRFLink	Bool	false	☑	☑	☑	☐	Error Byte 3.3 - No RF Link
17	ERR_NoWirelessACK	Bool	false	☑	☑	☑	☐	Error Byte 3.4 - No Wireless ACK
18	ERR_LostSample	Bool	false	☑	☑	☑	☐	Error Byte 3.5 - Lost Sample
19	ERR_Spare_3_6	Bool	false	☑	☑	☑	☐	Error Byte 3.6 - Spare
20	ERR_Spare_3_7	Bool	false	☑	☑	☑	☐	Error Byte 3.7 - Spare
21	Error_4	Byte	16#0	☑	☑	☑	☐	Error Byte 4 - Spare
22	HeatNumber	Array[0..9] of Char		☑	☑	☑	☑	

Results Data

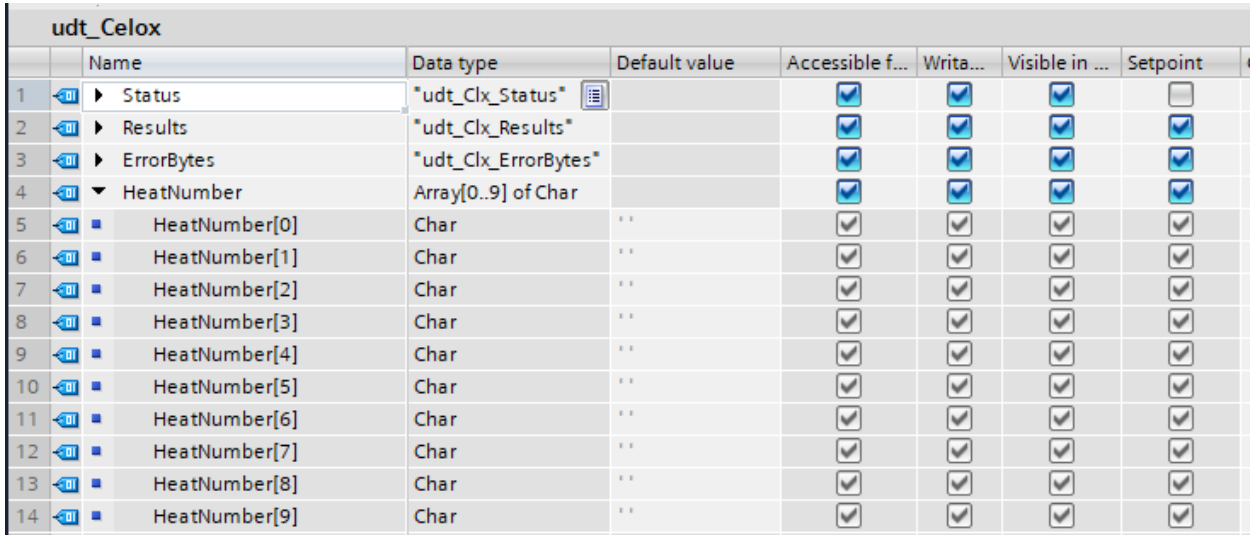
- I chose to build the Results portion of my structure like this.
  - Note that the Date and Time do not work as programmed and I did not have time to pursue this further.
  - The byte offset did seem to be correct, but I did not have time to check all of the error bytes to be sure, so **please defer to the manual and a better programmer for you final rollout.**
    THIS is for testing only



	Name	Data type	Default value	Accessible f...	Writa...	Visible in ...	Setpoint	Comment
1	Status	"udt_Clx_Status"		☑	☑	☑	☐	
2	Results	"udt_Clx_Results"		☑	☑	☑	☑	
3	Temperature	Real	0.0	☑	☑	☑	☐	
4	EMF	Real	0.0	☑	☑	☑	☐	
5	aO	Real	0.0	☑	☑	☑	☐	
6	AL	Real	0.0	☑	☑	☑	☐	
7	Carbon	Real	0.0	☑	☑	☑	☐	
8	CJunctTemp	Real	0.0	☑	☑	☑	☐	
9	CLX_Date	Date	D#1990-01-01	☑	☑	☑	☐	
10	CLX_Time	Time_Of_Day	TOD#00:00:00	☑	☑	☑	☐	
11	ErrorBytes	"udt_Clx_ErrorBytes"		☑	☑	☑	☑	
12	HeatNumber	Array[0..9] of Char		☑	☑	☑	☑	

Heat Number Data

- I chose to build the Heat Number portion of my structure like this.
  - Char array as strings don't seem to be supported in udts

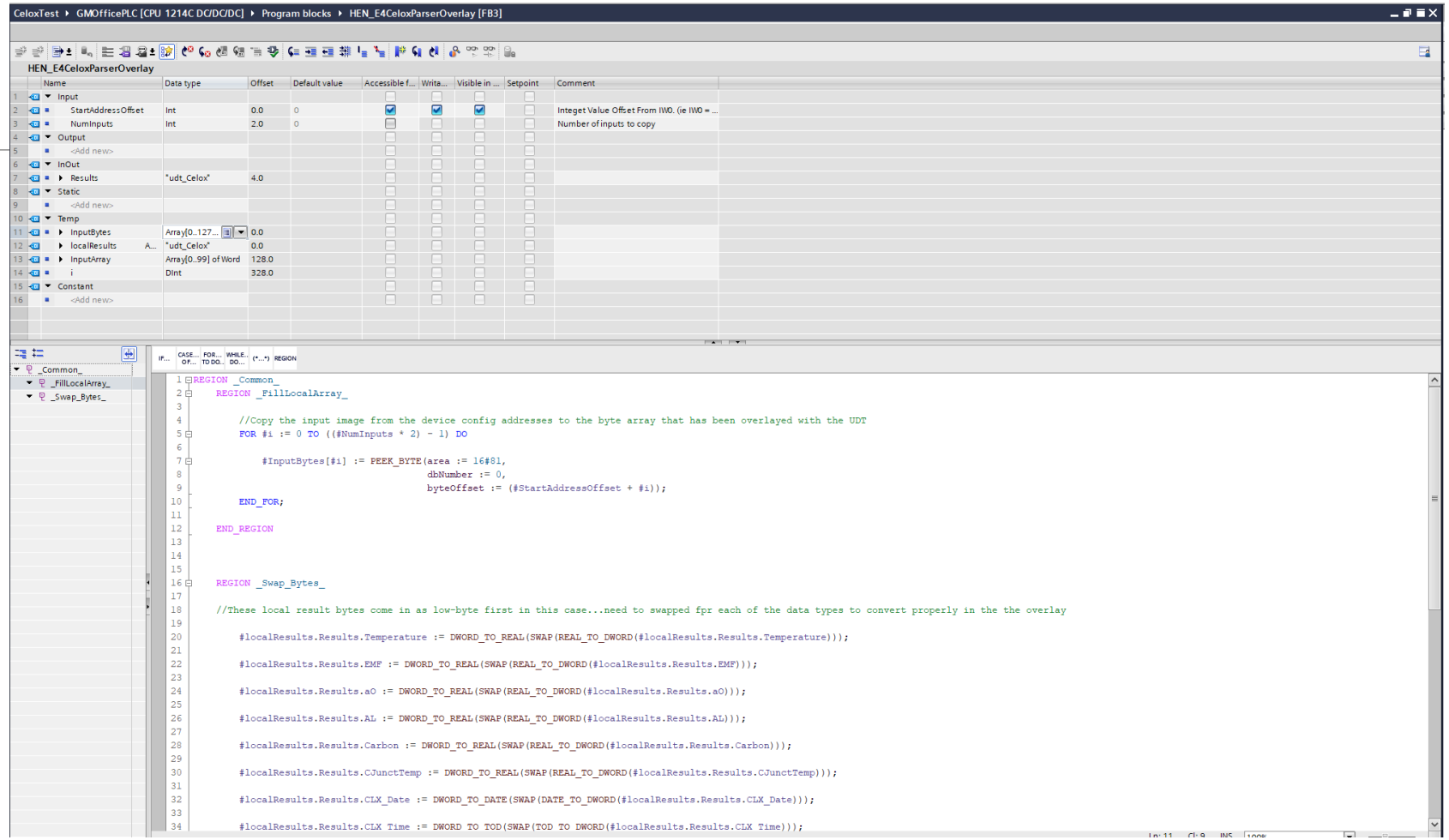


udt_Celox

	Name	Data type	Default value	Accessible f...	Writa...	Visible in ...	Setpoint
1	Status	"udt_Clx_Status"		☑	☑	☑	☐
2	Results	"udt_Clx_Results"		☑	☑	☑	☑
3	ErrorBytes	"udt_Clx_ErrorBytes"		☑	☑	☑	☑
4	HeatNumber	Array[0..9] of Char		☑	☑	☑	☑
5	HeatNumber[0]	Char	' '	☑	☑	☑	☑
6	HeatNumber[1]	Char	' '	☑	☑	☑	☑
7	HeatNumber[2]	Char	' '	☑	☑	☑	☑
8	HeatNumber[3]	Char	' '	☑	☑	☑	☑
9	HeatNumber[4]	Char	' '	☑	☑	☑	☑
10	HeatNumber[5]	Char	' '	☑	☑	☑	☑
11	HeatNumber[6]	Char	' '	☑	☑	☑	☑
12	HeatNumber[7]	Char	' '	☑	☑	☑	☑
13	HeatNumber[8]	Char	' '	☑	☑	☑	☑
14	HeatNumber[9]	Char	' '	☑	☑	☑	☑

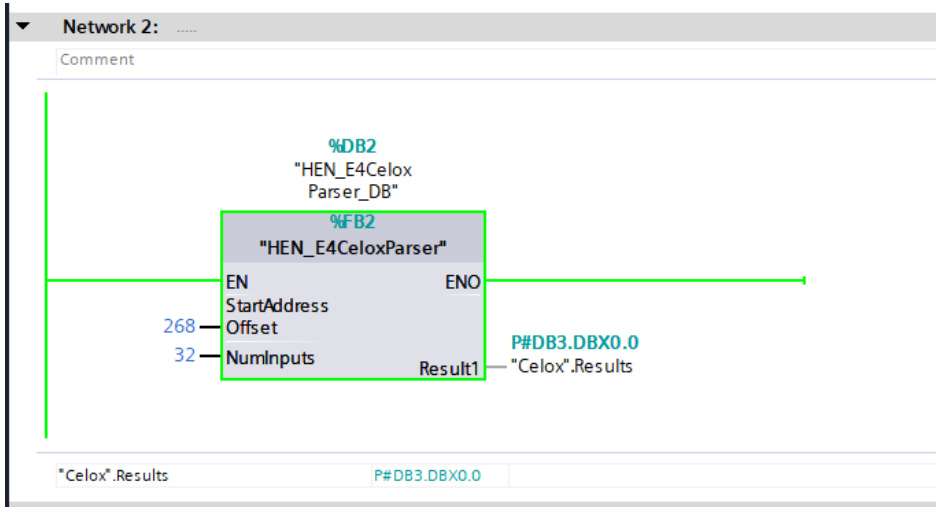
FB Logic

- I built an FB that takes the starting input image address offset and the number of inputs.
- It uses these to populate a local byte array
- That byte array is overlaid with a structure that matches the incoming data (see previous slides).
- The next step byte-swaps the floating-point fields because the data arrives in little-endian format.
- Finally, the FB copies the overlaid structure into the Results structure provided by the caller.



FB Logic

- The FB in Ladder



FB Logic

- The monitored results

Celox

	Name	Data type	Offset	Start value	Monitor value	Retain	Accessible f...	Writa...	Visible in ...	Setpoint	Comment
4	Celox	"udt_Celox"	672.0				☑	☑	☑		
5	Status	"udt_Clx_Status"	672.0				☑	☑	☑		
6	RedLight	Bool	672.0	false	FALSE		☑	☑	☑		Status Byte 1.0 - Red -End of Measurement
7	YellowLight	Bool	672.1	false	FALSE		☑	☑	☑		Status Byte 1.1 - Yellow - Measurement Busy
8	GreenLight	Bool	672.2	false	FALSE		☑	☑	☑		Status Byte 1.2 - Green - Probe Detected
9	MeasurementError	Bool	672.3	false	FALSE		☑	☑	☑		Status Byte 1.3 - Error Measurement
10	CarbonMeasurement	Bool	672.4	false	FALSE		☑	☑	☑		Status Byte 1.4 - Liquidous Measurement (not
11	BathLevelMeasurement	Bool	672.5	false	FALSE		☑	☑	☑		Status Byte 1.5 - Bath Level Measurement
12	CeloxMeasurement	Bool	672.6	false	FALSE		☑	☑	☑		Status Byte 1.6 - Celox(EMF) Measurement
13	TDxComplete	Bool	672.7	false	FALSE		☑	☑	☑		Status Byte 1.7 - TxD
14	StartMeasurementViewer	Bool	673.0	false	FALSE		☑	☑	☑		Status Byte 2.0 - Start Measurement viewer (Ex
15	LinesOpen	Bool	673.1	false	TRUE		☑	☑	☑		Status Byte 2.1 - Lines Open
16	EndMeasurementViewer	Bool	673.2	false	FALSE		☑	☑	☑		Status Byte 2.2 - End Measurement viewer (Ext
17	Status_2_3	Bool	673.3	false	FALSE		☑	☑	☑		Status Byte 2.3 - not used
18	Status_2_4	Bool	673.4	false	FALSE		☑	☑	☑		Status Byte 2.4 - not used
19	Status_2_5	Bool	673.5	false	FALSE		☑	☑	☑		Status Byte 2.5 - not used
20	BlinkingActive	Bool	673.6	false	FALSE		☑	☑	☑		Status Byte 2.6 - Blinking Active
21	Horn	Bool	673.7	false	FALSE		☑	☑	☑		Status Byte 2.7 - Horn Active
22	Status3_ProbeChar	Byte	674.0	16#0	16#4E		☑	☑	☑		Probe type (4E = 'N' = None) Table 8 in manual
23	Status4_PlaceID	Byte	675.0	16#0	16#01		☑	☑	☑		Place Id
24	Results	"udt_Clx_Results"	676.0				☑	☑	☑		
25	Temperature	Real	676.0	0.0	2447.78		☑	☑	☑		
26	EMF	Real	680.0	0.0	-180.005		☑	☑	☑		
27	aO	Real	684.0	0.0	0.478366		☑	☑	☑		
28	AL	Real	688.0	0.0	0.003875276		☑	☑	☑		
29	Carbon	Real	692.0	0.0	16#FFFF_FFFF		☑	☑	☑		
30	CJunctTemp	Real	696.0	0.0	72.68		☑	☑	☑		
31	CLX_Date	Date	700.0	D#1990-01-01	D#1990-01-01		☑	☑	☑		
32	CLX_Time	Time_Of_Day	702.0	TOD#00:00:00	16#8E88_0000		☑	☑	☑		
33	ErrorBytes	"udt_Clx_ErrorBytes"	706.0				☑	☑	☑		
34	ERR_NoCJTemp	Bool	706.0	false	FALSE		☑	☑	☑		Error Byte 1.0 - No Cold Junction
35	ERR_TCBreak	Bool	706.1	false	FALSE		☑	☑	☑		Error Byte 1.1 - TC Break
36	ERR_MeasurementOutOfRange	Bool	706.2	false	FALSE		☑	☑	☑		Error Byte 1.2 - Measurement out of range
37	ERR_Spare_1_3	Bool	706.3	false	TRUE		☑	☑	☑		Error Byte 1.3 - Spare
38	ERR_RFLossWhileMeasuring	Bool	706.4	false	TRUE		☑	☑	☑		Error Byte 1.4 - RF Link Wireless broken during r
39	ERR_Spare_1_5	Bool	706.5	false	FALSE		☑	☑	☑		Error Byte 1.5 - Spare
40	ERR_NoEvaluation	Bool	706.6	false	FALSE		☑	☑	☑		Error Byte 1.6 - No evaluation
41	ERR_GeneralError	Bool	706.7	false	FALSE		☑	☑	☑		Error Byte 1.7 - General Error
42	Application_Error	Byte	707.0	16#0	16#03		☑	☑	☑		Error Byte 2 - Application error byte definition
43	ERR_WrongProduct	Bool	708.0	false	FALSE		☑	☑	☑		Error Byte 3.0 - Wrong Product
44	ERR_ProtocolError	Bool	708.1	false	FALSE		☑	☑	☑		Error Byte 3.1 - Protocol Error
45	ERR_BatteryLow	Bool	708.2	false	FALSE		☑	☑	☑		Error Byte 3.2 - Low Battery
46	ERR_NoRFLink	Bool	708.3	false	FALSE		☑	☑	☑		Error Byte 3.3 - No RF Link
47	ERR_NoWirelessACK	Bool	708.4	false	FALSE		☑	☑	☑		Error Byte 3.4 - No Wireless ACK
48	ERR_LostSample	Bool	708.5	false	FALSE		☑	☑	☑		Error Byte 3.5 - Lost Sample
49	ERR_Spare_3_6	Bool	708.6	false	FALSE		☑	☑	☑		Error Byte 3.6 - Spare
50	ERR_Spare_3_7	Bool	708.7	false	FALSE		☑	☑	☑		Error Byte 3.7 - Spare
51	Error_4	Byte	709.0	16#0	16#00		☑	☑	☑		Error Byte 4 - Spare
52	HeatNumber	Array[0..9] of Char	710.0				☑	☑	☑		
53	HeatNumber[0]	Char	710.0	' '	'$00'		☑	☑	☑		
54	HeatNumber[1]	Char	711.0	' '	'$00'		☑	☑	☑		
55	HeatNumber[2]	Char	712.0	' '	'9'		☑	☑	☑		
56	HeatNumber[3]	Char	713.0	' '	'9'		☑	☑	☑		
57	HeatNumber[4]	Char	714.0	' '	'9'		☑	☑	☑		
58	HeatNumber[5]	Char	715.0	' '	'9'		☑	☑	☑		
59	HeatNumber[6]	Char	716.0	' '	'9'		☑	☑	☑		

Finally

- Hopefully these illustrations answer your connectivity questions. Parsing/programming the data is best handled by your in-house experts…that's their specialty.