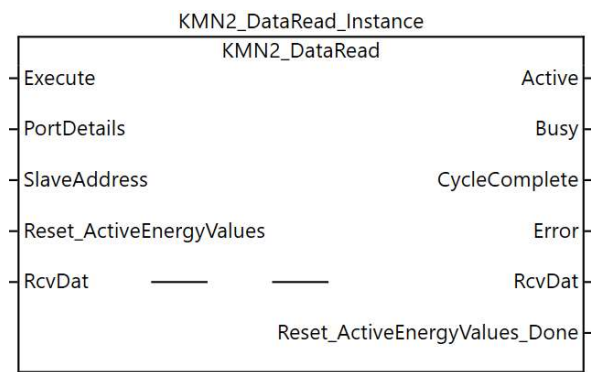


KMN2_DataRead

Function block to:

- 1. Read data from KM-N2 energy monitor
- 2. Perform reset of any active energy values on demand

Via Modbus RTU commands with NX-CIF105 / NX1W-CIF11/12.



Revision History

Revision	Date	Details
0.0.2	2021-01-29	'Busy' and 'CycleComplete' added, action of 'Active' changed
0.0.1	2021-01-08	Original version

Operation

- 1. While input variable 'Execute' is TRUE, reads KM-N2 energy readings into variable of structure 's_KMN2'
- 2. On rising edge of input variable 'Reset_ActiveEnergyValues' sends instruction code 03 to Modbus address 0xFFFF to reset any active energy values that can be reset.
- 3. 'CycleComplete' is TRUE for 1 cycle after all Modbus registers have been read (and 'Reset_ActiveEnergyValues' has been actioned if requested – see point 2). Purpose of this is to signal to the calling program that the communication cycle is complete, so next cycle a different 'SlaveAddress' can be specified in the case of multiple KM-N2s or multiple circuits on a single KM-N2.

Structure	Member	Data type	Units	KM-N2 Modbus address
s_KMN2	Voltage1_V	REAL	V	0x0000
	Voltage2_V	REAL	V	0x0002
	Voltage3_V	REAL	V	0x0004
	Current1_A	REAL	A	0x0006
	Current2_A	REAL	A	0x0008
	Current3_A	REAL	A	0x000A
	PowerFactor	REAL	N/A	0x000C
	Frequency_Hz	REAL	Hz	0x000E
	ActivePower_W	REAL	W	0x0010
	ReactivePower_Var	REAL	Var	0x0012
	VoltageV1toV2_V	REAL	V	0x0014
	VoltageV1toV3_V	REAL	V	0x0016
	VoltageV2toV3_V	REAL	V	0x0018
	ActiveEnergy_Wh	UDINT	Wh	0x0200
	ReactiveEnergy_Varh	UDINT	Varh	0x0204
	CumulativeTotalReactivePower_Varh	UDINT	Varh	0x0208
	ActiveEnergy_kWh	UDINT	kWh	0x0220
	ReactiveEnergy_kVarh	UDINT	kVarh	0x0224
	CumulativeTotalReactivePower_kVarh	UDINT	kVarh	0x0228
	ActiveEnergy_Resettable_Wh	UDINT	Wh	0x0240
	ReactiveEnergy_Resettable_Varh	UDINT	Varh	0x0244
	CumulativeTotalReactivePower_Resettable_Varh	UDINT	Varh	0x0248
	ActiveEnergy_Resettable_kWh	UDINT	kWh	0x0260
	ReactiveEnergy_Resettable_kVarh	UDINT	kVarh	0x0264
	CumulativeTotalReactivePower_Resettable_kVarh	UDINT	kVarh	0x0268

Example data trace (9.6kbps, 8, 1, e)

