KNX INTERFACE FOR HEAT RECOVERY UNITS

User manual

Description of the product

The Sabiana KNX interface card allows the connection of a Sabiana RVU appliance to a KNX installation.

Hardware description

Mechanically, the interface card is fitted on a plastic support inside the enclosure of the existing control board.

Power is supplied to the board by the KNX bus (SELV power); the communication lines on the RVU connector are optically isolated.

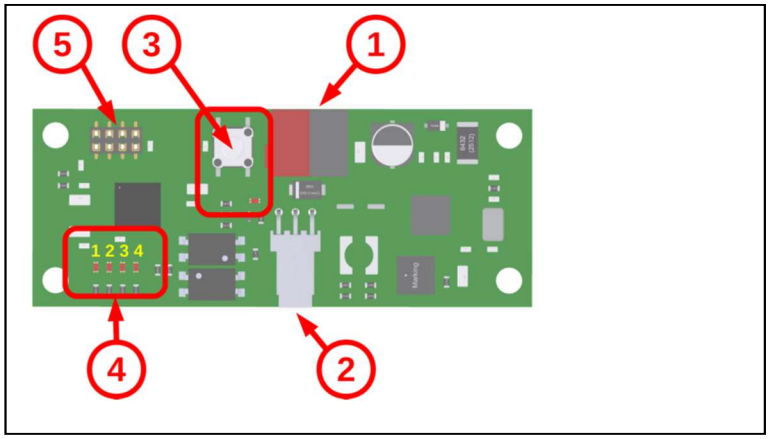
The communication between the board and the RVU unit is achieved through a serial Modbus RTU protocol, with a transmission speed of 1200 bps.

The main components on the board are:

1. the standard KNX bus connector
2. the 3-pin, 0.1" pitch Molex connector that will be linked to the corresponding port on the main control board through a pin-to-pin 3-pole cable (supplied with the interface)
3. the KNX programming pushbutton with the corresponding (red) LED indicator
4. a cluster of 4 LED indicators for diagnostics
5. a programming connector for the TI MSP430-F5341 MCU on board (intended for factory service only).

The meaning of the diagnostic LEDs is as follows:

- L1 - flashes when a KNX telegram is received
- L2 - flashes when a KNX telegram is sent
- L3 - flashes when a Modbus operation is performed
- L4 - pulses (~2s) if a Modbus communication error occurs



Information available through the KNX bus

The KNX devices connected on the same installation can access a number of data objects on the RVU (CO, *Communication Objects*) that basically match those seen on the standard Modbus port of the control board (although in a few cases the data format or range is slightly different).

For a complete list of available data objects and their specifics, please refer to the communication object tables listed later in this manual.

Interface programming

In order to setup the interface for operation, like for any KNX device, the user is required to:

- select which communication objects are required by making them addressable through a *Group Address*;
- Verify and possibly change the interface configuration parameters (see corresponding section below).

These operations can be performed by means of the standard KNX ETS software tool; a user qualified to operate on a KNX installation is supposed to be knowledgeable about this tool and its usage. Further details can however be found in the appropriate KNX documentation.

An application file (APRVUKNX_v*XXX*.knxprod) is supplied and available for free download from the manufacturer's website; this file must be loaded into ETS in order to describe the capabilities of the interface card.

Once the configuration is established, it can be downloaded to the interface by pressing the programming button and starting the download from ETS.

Interface configuration parameters

Through the `.knxprod` application file, ETS is made aware of the Communication Object set that the interface can handle, but also of the parameters that can be changed to affect its behaviour.

This interface has mainly two kind of parameter types, both related to how the data objects are transmitted on the KNX bus.

Before describing them, it is probably helpful to recall that, in a KNX system, a data object is normally transmitted on the bus:

- when its value changes;
- when responding to a request from another device.

This behaviour can be modified to better suit particular operating requirements. Specifically, in our case, the transmission of several data objects can be affected in one of the following ways:

- ***Periodic transmission*** **-** force transmission at regular intervals, regardless whether the value has changed[1].
  In this case, a transmission interval can be specified as parameter, together with an initial delay value after the device startup (i.e. the value is sent the first time after <T1> seconds/minutes/etc, then every <T2> seconds/minutes/etc.)
- ***Conditional transmission*** **-** limit transmission rate by setting a condition on the variation of the value (i.e. send the value only if it changes at least by <n> units) and/or on the emission frequency (i.e. send another value not earlier than <T> seconds/minutes/etc after last transmission).

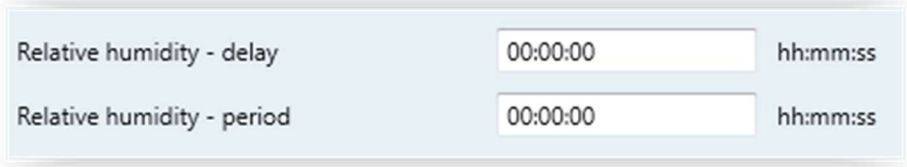
For a given Communication Object, one or both (or none) of above mechanisms will be available, according to the nature of the associated data; for specific information, please refer to the CO table in the appendix.

[1] The value is still transmitted if it changes between scheduled transmission intervals.

Periodic transmission

This condition is normally useful for steady or very slow-changing values; for instance, it helps in making sure that a value is correctly synced also to devices that might have started "listening" after last variation, or that a relevant condition (e.g. alarm) is detected within a certain timeframe even if a single transimission has been missed for whatever reason. This mechanism intends to effect an *increase* in transmissions with respect to the default behaviour.

Parameters for this mechanism have the following appearance:



If the period value amounts to 0, periodic transmission is not active.

Conditional transmission

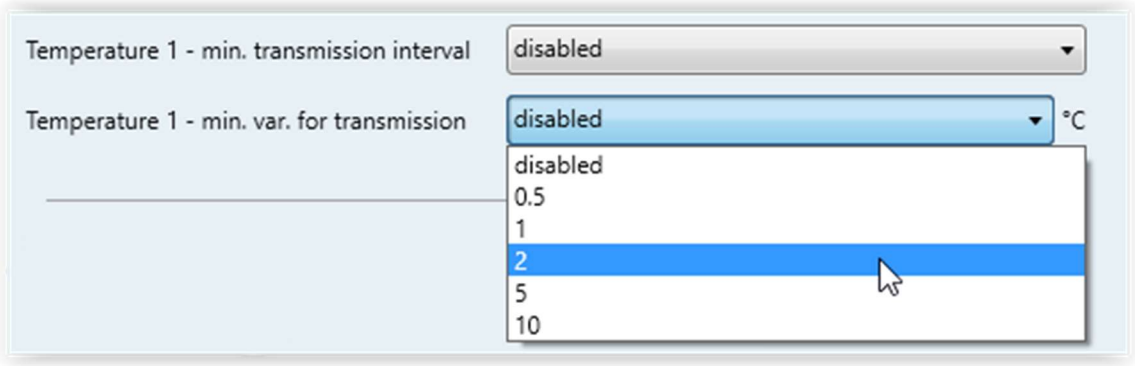
A condition can used to prevent an excessive transmission rate for either:

- values that may be too rapidly changing (and would "flood" the bus causing operational issues)
- values for which only variations beyond a certain threshold are deemed meaningful (e.g. a temperature varying in steps of 0.1 °C, but for which a variation of less than 1°C is not interesting for our purposes).

It must be pointed out that those above are two different conditions that could be set and work independently.

These conditions intend to *limit* transmissions with respect to the default behaviour.

Parameters for this mechanism have the following appearance:



Values are selectable from a predefined set through a drop-down box; if value "disabled" is selected, the corresponding limit is not active.

Caveats

- If both periodic AND conditional transmission can be applied, a conflict might ensue between them, also depending on the parameters chosen. For instance, one could configure a periodic transmission every 60s, but also a minimum spacing of 90s between transmissions. In this case, a conservative approach is taken, and the limit has priority over the forced transmission; in the example above, after a transmission has occurred, the next transmission (60s later) will be blocked. The next following transmission request will occur 120s later, which is beyond the 90s limit, and is therefore regularly carried out.
- For conditional transmission, if both conditions are active on the same value, the most restrictive is applied.

  For instance, assuming a value with min. interval = 30s and min. variation = 10 units:

  - value varies +15 after 20s -> not transmitted (too soon)
  - when the 30s expire: value is now +15 from last sent -> **transmitted**
  - 40s later, value varies +5 -> not transmitted (too little variation)

Data handling - Interfaces

Information available on the KNX side and on the Modbus side is almost completely coincident; however, there are slight differences in format and assignment to data objects, mostly because KNX data must adhere to the KNX standard requirements.

Just a few examples:

- temperature values are available in Modbus as integers, multiplied by 10 to account for decimals, whereas in KNX they are available as floating point values with a resolution of 0.01°C
- status or configuration flags are grouped into status words in Modbus, whereas on the KNX side they are available as individual binary objects
- some objects are made available in more that one form as required by the KNX standard, like e.g. fan speeds that must be presented in both RPM and percent form
- some objects must have different values or ranges (e.g. the values for operating mode selection), again because of KNX restrictions or requirements over data representation

KNX also has specific requirements for the way data is read or written; for instance, a few data objects enabled for write must have a corresponding "feedback" object enabled for read, in order to detect whether the value was actually modified by a write operation.

For every detail about data available through the KNX interface and differences with the data available through Modbus, please refer to the attached lists.

Data handling - Internal data flows

The data flows are handled by the device as follows.

An intermediate storage array exists as buffer for transfers in both directions. Transfers in each direction are composed of two "legs", from source to intermediate storage and from intermediate storage do destination, for a total of four legs; these four legs are cyclically executed in a continuous loop.

From Modbus to KNX - At each cycle, a block of Modbus register is read into the intermediate storage; values that have changed are marked as updated. In the next leg, intermediate objects are scanned and the next one found as updated is transmitted on the KNX bus[2]. The scan is then interrupted until the next cycle; this way, internal processing can continue and further updated objects will be transmitted in the next cycles (the KNX bus inherently has a limited transmission rate anyway).

In the process of distributing Modbus data to the intermediate storage, format conversions and data disassembly (e.g. word-grouped flags) are performed as appropriate.

Since the serial communication speed on the Modbus interface is 1200 bps, with the total number of Modbus registers to be read, a complete refresh cycle for the internally stored data is expected to take on average something less than 2s. No data is prioritized for faster refresh, although there are provisions to do so if required.

From KNX to Modbus - Whenever a KNX data object is received for writing, its value (if different) is written to the intermediate storage and marked as updated. In the next leg, for all intermediate objects updated from KNX, a new Modbus write request is issued (and inserted in a queue, should other requests be pending). This request is taken care of in the part of the cycle that also reads data from Modbus, and has priority over read requests; read requests are delayed until all write requests are cleared. One single Modbus operation (either block-read or single write) is performed for every cycle.

Appropriate format conversions and data assembly (e.g. word-grouped flags) are performed right before executing the Modbus write operation.

KNX read requests are completely handled by the KNX stack and satisfied using the internal stack data buffer, which however is kept up-to-date at all times.

An illustration of the internal data and operation structure is provided as attachment.

[2] In some cases (e.g. constraints on minimum transmission interval or minimum value change for transmission), the transmission request itself is not issued, but the KNX data object accessible from the bus is nonetheless updated.

Technical information

Power supply: ..... 30V DC SELV from KNX bus

Supply current: ..... max 20 mA

Comm speed on Modbus port:..... 1200 bps

EMC Compliance:..... EN 50491-5-1/2010; EN 50491-5-2/2010

Size (mm): ..... 76 x 30 x 15

List of objects

Sabiana KNX-RVU Interface
List of KNX Comm. Objects vs. RVU Modbus data
Rel. 1.00
04/10/2018

Source (Modbus) data						**KNX data**							
Src Addr	Src Type	Src Attr.	Size byte	Bit#	Values	CO Index	KNX DPT	KNX size (bits)	KNX Flags	CYC	Var name	Description	Notes
0x0000	char	RW	20	-		178	16.000	14*8	CRWTU		SerialNo	Serial Number	14 chars, not 20
0x000A	uns16	R	2	-		179	7.Model	16	CR-T-		CtrlModel	Controller model	
0x000B	uns16	R	2	-	*Low = minor, High = major*	180	7.FWRel	16	CR-T-		FW_Release	Firmware Release	
0x000C	uns16	R	2	-	*Low = minor, High = major*	181	7.FWRel	16	CR-T-		Protocol_Release	Protocol Release	
0x000D	uns16	R	2	-	*Low = minor, High = major*	182	7.FWRel	16	CR-T-		TEP_FW_Release	T-EP Firmware Release	
0x0100	Sig16	R	2	-	°Cx10	39	9.001	16f	CR-T-	X	TempProbe1	Probe 1 temperature	°C, not °Cx10 (Float, resolution: 0.01 °C)
0x0101	Sig16	R	2	-	°Cx10	40	9.001	16f	CR-T-	X	TempProbe2	Probe 2 temperature	°C, not °Cx10 (Float, resolution: 0.01 °C)
0x0102	Sig16	R	2	-	°Cx10	41	9.001	16f	CR-T-	X	TempProbe3	Probe 3 temperature	°C, not °Cx10 (Float, resolution: 0.01 °C)
0x0103	Sig16	R	2	-	°Cx10	42	9.001	16f	CR-T-	X	TempProbe4	Probe 4 temperature	°C, not °Cx10 (Float, resolution: 0.01 °C)
0x0104	Uns16	R	0.1	0	0/1	131	1.002	1	CR-T-		CFG_Inverted	Inverted configuration	
	Uns16	R	0.1	1	0/1	132	1.002	1	CR-T-		CFG_Preheating	Pre-heating present	
	Uns16	R	0.1	2	0/1	133	1.002	1	CR-T-		CFG_PreheatingH2O	Preheating with water or HE	
	Uns16	R	0.1	3	0/1	134	1.002	1	CR-T-		CFG_PostTreatment	Post treatment ON	
	Uns16	R	0.1	4	0/1	135	1.002	1	CR-T-		CFG_PostTrtSummer	Post treatment also summer	
	Uns16	R	0.1	5	0/1	136	1.002	1	CR-T-		CFG_RL5_IAQ	Relay 5 IAQ or Fault signaling	
	Uns16	R	0.1	6	0/1	137	1.002	1	CR-T-		CFG_PreTreatment	Pre treatment	
	Uns16	R	0.1	7	0/1	138	1.002	1	CR-T-		CFG_BoilerPBoost	Boiler pressure booster	
	Uns16	R	0.1	8	0/1	139	1.002	1	CR-T-		CFG_PostTrtExtHE	Post treatment external HE	
	Uns16	R	0.1	9	0/1	140	1.002	1	CR-T-		CFG_UART_HiSpd	UART High speed mode	
	Uns16	R	0.1	9	0/1	141	1.002	1	CR-T-		CFG_PostTrtOnT2	Post treatment T3/T2	

0x0105	Uns16	R	0.1	0	0/1	62	1.011	1	CR-T-	X	STM_RemoteOff	Remote OFF active	
	Uns16	R	0.1	1	0/1	63	1.011	1	CR-T-	X	STM_Bypass	Bypass active	
	Uns16	R	0.1	2	0/1	64	1.011	1	CR-T-	X	STM_ElecPreHeat	Electric Pre Heater active	
	Uns16	R	0.1	3	0/1	65	1.011	1	CR-T-	X	STM_WaterPreHeat	Water pre-heating active	
	Uns16	R	0.1	4	0/1	66	1.011	1	CR-T-	X	STM_Boost	Boost active	
	Uns16	R	0.1	5	0/1	67	1.011	1	CR-T-	X	STM_DefrostCycle	Defrost cycle active	
	Uns16	R	0.1	7	0/1	68	1.011	1	CR-T-	X	STM_PartyMode	Party Mode active	
	Uns16	R	0.1	8	0/1	69	1.011	1	CR-T-	X	STM_StateOn	On status	
	Uns16	R	0.1	11	0/1	70	1.002	1	CR-T-	X	STM_Winter	Winter setting	
0x0106	Uns16	R	2	-	RH% x10 (40.0% = 400)	43	9.007	16f	CR-T-	X	HumiditySet	Humidity setpoint	%, not %x10 (Float, resolution: 0.01 %)
0x0107	Uns16	R	2	-	x 15min	44	7.007	16	CR-T-	X	FilterCounter	Filter counter	Hours, not min/15
0x0108	Byte	R	0.1	2	0/1	71	1.001	1	CR-T-	X	OUT_Damper_CW	Damper CW output	
	Byte	R	0.1	3	0/1	72	1.001	1	CR-T-	X	OUT_Damper_CCW	Damper CCW output	
0x0109	Byte	R	0.1	0	0/1	73	1.011	1	CR-T-	X	RLY_Fault_IAQ	IAQ fault relay on	
	Byte	R	0.1	1	0/1	74	1.011	1	CR-T-	X	RLY_Preheat	Preheat relay on	
	Byte	R	0.1	2	0/1	75	1.011	1	CR-T-	X	RLY_PostHeat	Postheat relay on	
	Byte	R	0.1	3	0/1	76	1.011	1	CR-T-	X	RLY_Fans	Fans relay on	
	Byte	R	0.1	4	0/1	77	1.011	1	CR-T-	X	RLY_PostCoolHeat	Postcool/heat2 relay on	
0x010A	Byte	R	0.1	1	0/1	78	1.011	1	CR-T-	X	DIN_C1	Input C1 status	
	Byte	R	0.1	2	0/1	79	1.011	1	CR-T-	X	DIN_C2	Input C2 status	
	Byte	R	0.1	3	0/1	80	1.011	1	CR-T-	X	DIN_C3	Input C3 status	
	Byte	R	0.1	4	0/1	81	1.011	1	CR-T-	X	DIN_C4	Input C4 status	
0x010B	Uns16	R	2	-	rpm	45	8.RPM	16s	CR-T-	X	S_Fan1SpeedActual	Fan 1 actual Speed	
[0x010B]					**rpm**	46	5.001	8	CR-T-	X	Fan1SpeedActualPerc	**Fan 1 actual Speed %**	Percent version
0x010C	Uns16	R	2	-	rpm	47	8.RPM	16s	CR-T-	X	S_Fan2SpeedActual	Fan 2 actual Speed	
[0x010C]					**rpm**	48	5.001	8	CR-T-	X	Fan2SpeedActualPerc	**Fan 2 actual Speed %**	Percent version
0x010D	Uns16	R	2	-	0-1000 PWM Fan1 %	49	8.010	16s	CR-T-	X	Fan1DutyCycle	Duty Cycle Fan1	%, not %x10 (Float, resolution: 0.01 %)
0x010E	Uns16	R	2	-	0-1000 PWM Fan2 %	50	8.010	16s	CR-T-	X	Fan2DutyCycle	Duty Cycle Fan2	%, not %x10 (Float, resolution: 0.01 %)
0x010F	Uns16	R	2	-	0-100%	51	5.004	8	CR-T-	X	PreheaterDutyCycle	Duty Cycle El. Preheater	%, resolution: 1%

[0x0110]					0/1	147	1.005	1	CR-T-	X	ALM_General	General Alarm	
0x0110	Uns16	R	0.1	0	0/1	148	1.005	1	CR-T-	X	ALM_T1_Fail	T1 probe failure alarm	
	Uns16	R	0.1	1	0/1	149	1.005	1	CR-T-	X	ALM_T2_Fail	T2 probe failure alarm	
	Uns16	R	0.1	2	0/1	150	1.005	1	CR-T-	X	ALM_T3_Fail	T3 probe failure alarm	
	Uns16	R	0.1	3	0/1	151	1.005	1	CR-T-	X	ALM_T4_Fail	T4 probe failure alarm	
	Uns16	R	0.1	4	0/1	152	1.005	1	CR-T-	X	ALM_Timekeeper	Timekeeper failure alarm	
	Uns16	R	0.1	5	0/1	153	1.005	1	CR-T-	X	ALM_Frost	Frost alarm	
	Uns16	R	0.1	6	0/1	154	1.005	1	CR-T-	X	ALM_Frost_T2	Frost alarm (from T2 probe)	
	Uns16	R	0.1	7	0/1	155	1.005	1	CR-T-	X	ALM_Fireplace	Fireplace Alarm	
	Uns16	R	0.1	8	0/1	156	1.005	1	CR-T-	X	ALM_PressTxFail	Pressure transducer failure alarm	
	Uns16	R	0.1	9	0/1	157	1.005	1	CR-T-	X	ALM_Filter	Filter alarm	
	Uns16	R	0.1	10	0/1	158	1.005	1	CR-T-	X	ALM_FansFail	Fans failure	
	Uns16	R	0.1	11	0/1	159	1.005	1	CR-T-	X	ALM_RH_CO2_TxFail	RH or CO2 sensor failure alarm	
	Uns16	R	0.1	12	0/1	160	1.005	1	CR-T-	X	ALM_FanThermalInput	Fan thermic input alarm	
	Uns16	R	0.1	14	0/1	161	1.005	1	CR-T-	X	ALM_PreHeating	Pre Heating alarm	
	Uns16	R	0.1	15	0/1	162	1.005	1	CR-T-	X	ALM_PreFrost_T2	Pre frost alarm (T2)	
0x0111	Sig16	R	2	-	Pascal x10	52	9.006	16f	CR-T-	X	PressDiffSensor1	Diff. Pressure Sensor 1	Pa, not Pa/10 (Float, resolution: 0.01 Pa)
0x0112	Sig16	R	2	-	Pascal x10	53	9.006	16f	CR-T-	X	PressDiffSensor2	Diff. Pressure Sensor 2	Pa, not Pa/10 (Float, resolution: 0.01 Pa)
0x0113	Uns16	R	2	-	ppm	54	9.008	16f	CR-T-	X	CO2Value	CO2 reading	Float, resolution 0.01 ppm
0x0114	Uns16	R	2	-	RH% x10 (50,0%=500)	55	9.007	16f	CR-T-	X	RelHumValue	RH reading	%, not %x10 (Float, resolution: 0.01 %)
0x0115	Float32	R	4	-	V1 In (kg/m3) [0,5..2]	56	14.017	32f	CR-T-	X	Rho1	Rho1	Float, resolution: 1 kg/m3
0x0117	Float32	R	4	-	V1 Out (kg/m3) [0,5..2]	57	14.017	32f	CR-T-	X	Rho2	Rho2	Float, resolution: 1 kg/m3
0x0119	Float32	R	4	-	V2 In (kg/m3) [0,5..2]	58	14.017	32f	CR-T-	X	Rho3	Rho3	Float, resolution: 1 kg/m3
0x011B	Float32	R	4	-	V2 Out (kg/m3) [0,5..2]	59	14.017	32f	CR-T-	X	Rho4	Rho4	Float, resolution: 1 kg/m3
0x011D	Uns16	R	2	-	Speed coeff 1 (x 0x4000)	60	5.005	8	CR-T-	X	SpeedCoeff1	Speed coeff. 1	Ratio 0..255
0x011E	Uns16	R	2	-	Speed coeff 2 (x 0x4000)	61	5.005	8	CR-T-	X	SpeedCoeff2	Speed coeff. 2	Ratio 0..255
0x011F	Uns16	R	0.1	8	0/1	168	1.002	1	CR-T-		OPT_IAQ	IAQ Used	
	Uns16	R	0.1	9	0/1	169	1.002	1	CR-T-		OPT_PostTreatment	Post treatment used	
	Uns16	R	0.1	10	0/1	170	1.002	1	CR-T-		OPT_HE	HE used	
	Uns16	R	0.1	11	0/1	171	1.002	1	CR-T-		OPT_BoilerBoost	Boiler boost mode used	
	Uns16	R	0.1	12	0/1	172	1.002	1	CR-T-		OPT_CO2Sensor	CO2 sensor present	
	Uns16	R	0.1	13	0/1	173	1.002	1	CR-T-		OPT_DiffPressSensor	Differential pressure sensor present	
	Uns16	R	0.1	14	0/1	174	1.002	1	CR-T-		OPT_RHSensor	RH sensor present	
	Uns16	R	0.1	15	0/1	175	1.002	1	CR-T-		OPT_ReverseMount	Reverse mounting	

0x0120	Uns32	R	4	-	hours	82	13.100	32	CR-T-		FanOnHrs	Fans ON hours counter	seconds (in 3600-count steps)
0x0200	Sig16	RW	0.1	1	0/1	18	1.002	1	C-W-U		PRM_StopFanOn	Stop Mode Fan command	
	Sig16	RW	0.1	1	0/1	26	1.002	1	CR-T-		S_PRM_StopFanOn	Stop Mode Fan status	
	Sig16	RW	0.1	2	0/1	34	1.002	1	C-W-U		PRM_FlushFanOn	Flush Fan command	
	Sig16	RW	0.1	2	0/1	35	1.002	1	CR-T-		S_PRM_FlushFanOn	Flush Fan status	
0x0201	Sig16	RW	2	-	°C x10 (-40...+40)	87	9.001	16f	CRWTU		TempProbe1Ofst	Temp. probe 1 offset	°C, not °Cx10 (Float, resolution: 0.01 °C)
0x0202	Sig16	RW	2	-	°C x10 (-40...+40)	88	9.001	16f	CRWTU		TempProbe2Ofst	Temp. probe 2 offset	°C, not °Cx10 (Float, resolution: 0.01 °C)
0x0203	Sig16	RW	2	-	°C x10 (-40...+40)	89	9.001	16f	CRWTU		TempProbe3Ofst	Temp. probe 3 offset	°C, not °Cx10 (Float, resolution: 0.01 °C)
0x0204	Sig16	RW	2	-	°C x10 (-40...+40)	90	9.001	16f	CRWTU		TempProbe4Ofst	Temp. probe 4 offset	°C, not °Cx10 (Float, resolution: 0.01 °C)
0x0205	Sig16	RW	2	-	V x100 (100...1000)	91	9.020	16f	CRWTU		FanVoltageMin	Fan min voltage	Vx1000, not Vx100 (Float, mV, resolution: 0.01 mV)
0x0206	Sig16	RW	2	-	V x100 (100...1000)	92	9.020	16f	CRWTU		FanVoltageMax	Fan max voltage	Vx1000, not Vx100 (Float, mV, resolution: 0.01 mV)
0x0207	Sig16	RW	2	-	V x100 (100...1000)	93	9.020	16f	CRWTU		Fan1VoltageNom	Fan 1 Nominal V drive	Vx1000, not Vx100 (Float, mV, resolution: 0.01 mV)
0x0208	Sig16	RW	2	-	V x100 (100...1000)	94	9.020	16f	CRWTU		Fan2VoltageNom	Fan 2 Nominal V drive	Vx1000, not Vx100 (Float, mV, resolution: 0.01 mV)
0x0209	Sig16	RW	2	-	rpm (200...4000)	95	8.RPM	16s	CRWTU		FanMinSpeed	Fan min speed	
0x020A	Sig16	RW	2	-	rpm (1000...4000)	96	8.RPM	16s	CRWTU		FanMaxSpeed	Fan max Speed	
0x020B	Sig16	RW	2	-	rpm (1000...4000)	97	8.RPM	16s	CRWTU		Fan1SpeedSet	Fan 1 nominal speed	
[0x020B]					rpm (%)	98	5.001	8	CRWTU		Fan1SpeedSetPerc	**Fan 1 nominal speed %**	
0x020C	Sig16	RW	2	-	rpm (1000...4000)	99	8.RPM	16s	CRWTU		Fan2SpeedSet	Fan 2 nominal speed	
[0x020C]					rpm (%)	100	5.001	8	CRWTU		Fan2SpeedSetPerc	**Fan 2 nominal speed %**	
0x020D	Sig16	RW	2	-	rpm (1000...4000)	101	8.RPM	16s	CRWTU		Fan1SpeedStd	Fan 1 installation speed	
[0x020D]					rpm (%)	102	5.001	8	CRWTU		Fan1SpeedStdPerc	**Fan 1 installation speed %**	
0x020E	Sig16	RW	2	-	x100 (1000...9000)	103	8.001	16	CRWTU		KCoeff1	K coefficient 1	
0x020F	Sig16	RW	2	-	x100 (1000...9000)	104	8.001	16	CRWTU		KCoeff2	K coefficient 2	
0x0210	Sig16	RW	2	-	Q: m3/h (30...500)	105	9.009	16f	CRWTU	X	AirFlow1	Air flow 1	Resolution: 1 m3/h
0x0211	Sig16	RW	2	-	Q: m3/h (30...500)	106	9.009	16f	CRWTU	X	AirFlow2	Air flow 2	Resolution: 1 m3/h
0x0212	Sig16	RW	2		*1-4 (Speed 1-Speed 4)*	12	5.SpdSel	8	C-W-U		ManSpeedNrSet	Manual Speed command	(Write) - Different values (SEE NOTES)
0x0212	Sig16	RW	2		*1-4 (Speed 1-Speed 4)*	13	5.SpdSel	8	CR-T-	X	ManSpeedNrActual	Manual Speed status	(Read) - Different values (SEE NOTES)
0x0213	Sig16	RW	2	-	(0...35)	107	5.004	8	CRWTU		Speed1Pct	Speed 1 %	Resolution: 1%
0x0214	Sig16	RW	2	-	(35...70)	108	5.004	8	CRWTU		Speed2Pct	Speed 2 %	Resolution: 1%
0x0215	Sig16	RW	2	-	(45...100)	109	5.004	8	CRWTU		Speed3Pct	Speed 3 %	Resolution: 1%

0x0216	Sig16	RW	2	-	(100...110)	110	5.004	8	CRWTU		Speed4Pct	Speed 4 %	Resolution: 1%
0x0217	Sig16	RW	2	-	(110...130)	111	5.004	8	CRWTU		SpeedBoostPct	Boost Speed %	Resolution: 1%
0x0218	Sig16	RW	2	-	°C x10 (100..300)	112	9.001	16f	CRWTU		TempSummerSet	Summer T Setpoint	°C, not °Cx10 (Float, resolution: 0.01 °C)
0x0219	Sig16	RW	2	-	°C x10 (100..300)	113	9.001	16f	CRWTU		TempWinterSet	Winter T setpoint	°C, not °Cx10 (Float, resolution: 0.01 °C)
0x021A	Sig16	RW	2	-	minutes (1...15)	114	7.006	16	CRWTU		AirCoeffRecalcIntv	Air coefficients recalc. interval	Resolution: 1 min
0x021B	Sig16	RW	2	-	°C x10 (100...300)	115	9.001	16f	CRWTU		TempFreeCooling	Temp for free cooling	°C, not °Cx10 (Float, resolution: 0.01 °C)
0x021C	Sig16	RW	2	-	°C x10 (100.300)	116	9.001	16f	CRWTU		TempFreeHeating	Temp for free heating	°C, not °Cx10 (Float, resolution: 0.01 °C)
0x021D	Sig16	RW	2	-	(-20...+20)	117	6.001	8s	CRWTU		Fan2UnbalancePct	Fan 2 unbalance %	Resolution: 1%
0x021E	Sig16	RW	2	-	(1...96)	118	5.010	8	CRWTU		HumNSamples	Humidity samples for setpoint	
0x021F	Sig16	RW	2	-	minutes (60...240)	119	7.006	16	CRWTU		BoostTime	Boost time	Resolution: 1 min
0x0220	Sig16	RW	2	-	(5...50)	128	5.005	8	CRWTU		HumRegPConst	P constant for humidity regulator	Ratio 0..255
0x0221	Sig16	RW	2	-	Days (30...400)	120	7.007	16	CRWTU		FilterLife	Filter life	Hours, not Days
0x0222	Sig16	RW	2	-	ppm (100...30000)	121	9.008	16f	CRWTU		CO2MinSet	CO2 ppm min	Float, ppm, resolution: 0.01 ppm
0x0223	Sig16	RW	2	-	ppm (100...30000)	122	9.008	16f	CRWTU		CO2NomSet	CO2 ppm nom	Float, ppm, resolution: 0.01 ppm
0x0224	Sig16	RW	2	-	ppm (100...30000)	123	9.008	16f	CRWTU		CO2MaxSet	CO2 ppm max	Float, ppm, resolution: 0.01 ppm
0x0225	Sig16	RW	2	-	(10...40)	129	5.005	8	CRWTU		CO2PropConst	CO2 ppm prop constant	Ratio 0..255
0x0226	Sig16	RW	0.1	0	0/1	19	1.003	1	C-W-U		ENA_Manual	Manual mode lock	(Write)
	Sig16	RW	0.1	1	0/1	20	1.003	1	C-W-U		ENA_Party	Party mode lock	(Write)
	Sig16	RW	0.1	2	0/1	21	1.003	1	C-W-U		ENA_Holiday	Holiday mode lock	(Write)
	Sig16	RW	0.1	3	0/1	22	1.003	1	C-W-U		ENA_Auto	Auto mode lock	(Write)
	Sig16	RW	0.1	4	0/1	23	1.003	1	C-W-U		ENA_Weekly	Weekly Prog mode lock	(Write)
	Sig16	RW	0.1	5	0/1	24	1.003	1	C-W-U		ENA_TimeDayChange	Time/day change lock	(Write)
	Sig16	RW	0.1	6	0/1	25	1.003	1	C-W-U		ENA_OffCmd	Off command lock	(Write)
	Sig16	RW	0.1	0	0/1	27	1.003	1	CR-T-		S_ENA_Manual	Manual mode lock status	(Read)
	Sig16	RW	0.1	1	0/1	28	1.003	1	CR-T-		S_ENA_Party	Party mode lock status	(Read)
	Sig16	RW	0.1	2	0/1	29	1.003	1	CR-T-		S_ENA_Holiday	Holiday mode lock status	(Read)
	Sig16	RW	0.1	3	0/1	30	1.003	1	CR-T-		S_ENA_Auto	Auto mode lock status	(Read)
	Sig16	RW	0.1	4	0/1	31	1.003	1	CR-T-		S_ENA_Weekly	Weekly Prog mode lock status	(Read)
	Sig16	RW	0.1	5	0/1	32	1.003	1	CR-T-		S_ENA_TimeDayChange	Time/day change lock status	(Read)
	Sig16	RW	0.1	6	0/1	33	1.003	1	CR-T-		S_ENA_OffCmd	Off command lock status	(Read)

0x0227	Sig16	RW	2	-	ppm (100...30000)	124	9.008	16f	CRWTU		CO2SensorRange	CO2 Sensor PPM Range	Float, ppm, resolution: 0.01 ppm
0x0228	Sig16	RW	2	-	Minutes (5...20)	130	7.006	16	CRWTU		BoilerBoostTime	Boiler boost time	Resolution: 1 min
0x0229	Sig16	RW	2	-	RH% x10 (200...300)	125	9.007	16f	CRWTU		RelHumLowSet	RH Low value	%, not %x10 (Float, resolution: 0.01 %)
0x022A	Sig16	RW	2	-	RH% x10 (400...500)	126	9.007	16f	CRWTU		RelHumStdSet	RH Standard value	%, not %x10 (Float, resolution: 0.01 %)
0x022B	Sig16	RW	2	-	RH% x10 (600...800)	127	9.007	16f	CRWTU		RelHumHiSet	RH Hi value	%, not %x10 (Float, resolution: 0.01 %)
0x0300	Uns16	RW	2	0	1=ON 0=OFF	0	1.001	1	C-W-U		CMD_OnOff	On/Off Command	(Write)
	Uns16	RW	2	0	1=ON 0=OFF	6	1.011	1	CR-T-	X	S_CMD_OnOff	On/Off status	(Read)
0x0301	Uns16	RW	2	0	1=Manual	1	1.001	1	C-W-U		CMD_Manual	Manual mode command	(Write)
	Uns16	RW	2	0	1=Manual	7	1.011	1	CR-T-	X	S_CMD_Manual	Manual mode status	(Read)
0x0302	Uns16	RW	2	0	1=Holiday	2	1.001	1	C-W-U		CMD_Holiday	Holiday mode command	(Write)
	Uns16	RW	2	0	1=Holiday	8	1.011	1	CR-T-	X	S_CMD_Holiday	Holiday mode status	(Read)
0x0303	Uns16	RW	2	0	1=Party	3	1.001	1	C-W-U		CMD_Party	Party mode command	(Write)
	Uns16	RW	2	0	1=Party	9	1.011	1	CR-T-	X	S_CMD_Party	Party mode status	(Read)
0x0304	Uns16	RW	2	0	1=Auto	4	1.001	1	C-W-U		CMD_Auto	Auto mode command	(Write)
	Uns16	RW	2	0	1=Auto	10	1.011	1	CR-T-	X	S_CMD_Auto	Auto mode status	(Read)
0x0305	Uns16	RW	2	0	1=Program	5	1.001	1	C-W-U		CMD_Program	Program mode command	(Write)
	Uns16	RW	2	0	1=Program	11	1.011	1	CR-T-	X	S_CMD_Program	Program mode status	(Read)
0x0306	Uns16	RW	2		*1-8=P1-P8*	14	18.001	8	C-W-U		TimerProgNrSet	Timer program selection	(Write) - Different values (SEE NOTES)
	Uns16	RW	2		*1-8=P1-P8*	15	17.001	8	CR-T-	X	TimerProgNrActual	Timer program current	(Read) - Different values (SEE NOTES)
0x0307	Uns16	RW	2		*0..4*	16	5.ModeSel	8	C-W-U		ModeNrSet	Mode Selection command	(Write) - Different values (SEE NOTES)
	Uns16	RW	2		*0..4*	17	5.ModeSel	8	CR-T-	X	ModeNrSetActual	Mode Selection status	(Read) - Different values (SEE NOTES)
[0x0308]	Uns16	W	2	-	Magic=0x005A	183	1.003	1	C-W-U		CMD_ResetEnable	Parameter reset guard	Enable reset through object #184
[0x0308]	Uns16	W	2	-	Magic=0x005A	184	1.015	1	C-W-U		CMD_FactoryReset	Parameter reset	Reset - requires CO #183 to be '1' (resets both). Writes "magic" value 0x005A in RVU register 0x308
0x030B	Uns16	W	2	-	ppm	185	9.008	16f	C-W-U		CO2SensExt	CO2 external sensor reading	ppm
0x030A	Uns16	W	2	-	RH% x10	186	9.007	16f	C-W-U		RelHumSensExt	RH external sensor reading	RH% x10 (50,0%=500)
0x0800+ 0x0801	Uns16+ Uns16	RW	4	-	Time + Day	176	10.001	24	CRWTU		Time_Day	Time	MSB = hour (0...23) LSB = minutes (0...59) Day 1 = Mon ... Day 7 = Sun

Values in red are Custom DPTs; in ETS they are currently implemented as "joker" version of the same-sized standard DPT, as hinted in the listed name (e.g. 5.ModeSel -> 5.xxx)

Value details:

	Modbus values	KNX values
0x0212	*Manual Speed*	1-4 (Speed 1- Speed 4) (Range no longer 0..3 as in Modbus data)
0x0306	*Timer prog selection*	1-8=P1-P8 (Range no longer 0..7 as in Modbus data)
0x0307	*Mode Selection:* *0:Holiday* *1:Auto* *2:Program* *3:Manual* *4:Party*	0:Manual (not Holiday) 1:Auto 2:Program 3:Holiday(not Manual) 4:Party