

RCT Power Serial Communication Protocol

Document version 1.14

Table of changes

Version	Date	Author	Details
1.6	29.01.2019	DPO	Command modificator and frame for modified commands
1.7	07.02.2019	DPO	Table of inverter faults is actualized
1.8	25.11.2019	DPO	Instead of modified and not modified frames the standard and plant communication frames are introduced. ID's table is completed. Detailed description of BMS faults added.
1.9	07.04.2019	DPO	In Plant Communication Frame by CRC-16 description the missing Address value added.
1.10	29.04.2020	DPO	New faults added (see Table 1, positions 122..126)
1.11	16.06.2020	DPO	New commands for periodical requests are added (command 0x07 in table 4, command 0x47 in table 5). Description is in the Command Byte chapter.
1.12	25.06.2020	DPO	Table 8 (table of ID's) is actualized
1.13	07.07.2020	DPO	The following sentence added: The default Port number used for the TCP/IP connection is 8899. READ PERIODICALLY command code 0x07 changed to 0x08 (see Tables 4 and 5).
1.14	11.02.2022	DPO	Plant Communication Frame description is extended with more details.

This communication protocol also used for communication via WiFi board with external clients (e.g., Android App). The default Port number used for the TCP/IP connection is 8899.

Standard Communication Frame (commands 0x01...0x06)

This frame is used for communication with one device.

	Bytes	Description
Start	1	START Byte '+' (0x2B)
Command	1	COMMAND Byte [Read, Write]
Length	1	4 + n Bytes [ID, Data]

	(2 for long commands)	
ID	4	
Data	N	
CRC	2	CRC16-CCITT [Command, Length, ID, Data]

Table 1

Note 1: all values in MSBF format

Note 2: frame can be preceded by one zero byte

Plant Communication Frame (commands 0x41...0x46)

This frame is used for communication with any "slave" device in the plants network. Plants network consists of one master and at least one slave (up to 31 slaves are supported).

	Bytes	Description
Start	1	START Byte '+' (0x2B)
Command	1	COMMAND Byte [Read, Write]
Length	1 (2 for long commands)	8 + n Bytes [Address, ID, Data]
Address	4	Network ID of the slave device
ID	4	
Data	N	
CRC	2	CRC16-CCITT [Command, Length, Address, ID, Data]

Table 2

Note 1: all values in MSBF format

Note 2: frame can be preceded by one zero byte

The list of all available slaves could be requested via "net.slave_data" variable (ID 0xC0A7074F). The response to this variable request is the following structure (total size 104 bytes):

Offset	Data type	Bytes	Content
0	uint_32	4	Network ID of the slave device
4	char	24	Device name, max 23 symbols + 0-terminator
28	float	4	AC power of the inverter + external power (S0 – signal) (positive = feed-in) [W]
32	float	4	Battery power (positive = discharge) [W]
36	float	4	Battery SoC [0..1]
40	uint_16	2	Actual fault index (see FAULT_ENUM enumeration)
42	uint_8	1	Bit coded info. Bit 0: battery supported, bit 1: battery connected, bit 2: DC (PV-inputs) supported, bit 3: external power over S0
43	uint_8	1	Inverter state (same as prim_sm.state variable)
44	float	4	External power (S0 - signal) [W]
48	char	16	Software version

64	char	16	Serial Number
80	uint_32	4	Completeness indicator
84	uint_32	4	Software version of BMS (if applicable)
88	char	16	Reserved

Table 3

Note 1: all values in MSBF format

Note 2: float in IEEE 754 standard for floating point

Each time this structure requested, it contains information about only one slave device. To get the information about all available slave devices (inverters), this structure should be requested multiple times appropriately.

CRC calculation

CRC-16 CCITT with polinom 0x1021 and seed 0xFFFF. If the number of bytes [Command, Length, ID, Data] for CRC calculation is odd, then the one padding zero byte at the end should be added.

For example:

command: **0x02**

length: **0x05**

id: **0x3e 0x21 0x5f 0x90**

data: **0x55**

resulting stream for CRC calculation: **0x02 0x05 0x3e 0x21 0x5f 0x90 0x55**

7 bytes is odd, so the stream for CRC calculation should be padded as follow: **0x02 0x05 0x3e 0x21 0x5f 0x90 0x55 0x00**

Here is an example of CRC calculation function (Java). It takes as an input a byte from the stream. The function should be called consecutively for all bytes of the stream. The resulting CRC-16 is found in "crc" variable, wich must be initialized by **0xFFFF** before start of CRC calculation.

```
int crc = 0xFFFF;

void calc_crc(int b) {
    for (int i = 0; i < 8; i++) {
        boolean bit = ((b >> (7 - i) & 1) == 1);
        boolean c15 = (((crc >> 15) & 1) == 1);
        crc <<= 1;
    }
}
```

```

        if (c15 ^ bit) crc ^= 0x1021;
    }

    crc &= 0xffff;
}

```

Control Bytes

START	'+'	0x2B
STOP (escape)	'_'	0x2D

Table 4

Bytes Stream Interpreting Rules

- '-' => replace by '-' and interpret as ordinal '-' character and only one '-' symbol will be used for CRC calculation
- '+> replace by '+' and interpret as ordinal '+' character and only one '+' symbol will be used for CRC calculation
- '+' => interpret as "Start Byte" for a new frame
- '_' => interpret as "Stop Byte" for escaping '-' or '+' symbols and will not be used for CRC calculation

Command Byte

READ	0x01
WRITE	0x02
LONG WRITE	0x03
Reserved	0x04
RESPONSE	0x05
LONG RESPONSE	0x06
Reserved	0x07
READ PERIODICALLY	0x08
EXTENSION	0x3C

Table 5

To request something the **READ** command should be used. The Inverter answers with one of the **RESPONSE** commands. The **LONG** commands are used for strings, arrays and data logging. All variables with length smaller or equal 251 bytes will be answered with ordinal **RESPONSE** command. The **READ PERIODICALLY** is similar to the **READ** command, but used in case when some parameter should be read periodically (e.g. each 30 seconds). Once this request has been sent, the response will come periodically forever. The very first response will come immediately. The period of this responses (in seconds) could be changed by setting the variable *pas.period* (unsigned 32 bit integer, ID 0x9C8FE559), the default value after device restart is 30 seconds. The **READ PERIODICALLY** command should be sent only once per parameter (the later requests will be ignored). No more than 64 parameters can be serviced simultaneously. For disabling of all periodical sendings the value of the *pas.period* variable should be set to 0 (the list of all periodical requests will be cleared). To enable this again, the period should be set to non zero value and the appropriate parameters should be requested again.

Command modifier (for plant communication)

Bit 6 (0x40) served for modification of any command, the modified commands are:

READ M	0x41
WRITE M	0x42
LONG WRITE M	0x43
Reserved	0x44
RESPONSE M	0x45
LONG RESPONSE M	0x46
Reserved	0x47
READ PERIODICALLY M	0x48

Table 6

This modified commands are used for networked plants communication. The Master device should not interpret this commands, but forward them. This allows communication with slave devices over the master. The slave device should interpret this commands only when its network ID (address) is selected in the commands frame.

Example

	Start	Command	Length	ID	Data	CRC
Read	START	READ	4	id	-	crc16
Write	START	WRITE	4 + n	id	data[n]	crc16
Write	START	LONG WRITE	4 + n	id	data[n]	crc16

Table 7

Request battery power (positive by discharge), ID = 0x400F015B, 4 bytes floating point format:

0x2B 0x01 0x04 0x40 0x0F 0x01 0x5B 0x58 0xB4

Request inverters AC power, ID = 0xDB2D69AE:

0x2B 0x01 0x04 0xDB 0x2D 0x2D 0x69 0xAE 0x55 0xAB

Service variable "com_service" (ID: 0x8FC89B10)

This is a special variable. It is regularly polled. If it changes - it will be interpreted as later described. The following table presents the meaning of different possible values for the „com_service“ variable (ID 0x8FC89B10). Note that the inverter take actions only by changing of the „com_service“ variable, so to „execute“ the same command again the value of the „com_service“ must be reseted to 0 first.

Value	Description
0	No meaning – default state
1	Reserved for internal usage
2	Reserved for internal usage
3	Reserved for internal usage
4	Reserved for internal usage
5	FLASH parameters
6	Erase parameters FLASH
7	Reserved for internal usage
8	Reserved for internal usage

9	Write Wi-Fi settings to the Wi-Fi board
10	Read Wi-Fi settings from the Wi-Fi board
11	Bulk erase of the whole Datalog (external FLASH)
12	Tune current sensors
13	Start battery booster test
14	Stop battery booster test
15	Start battery commission (extension)
16	Stop BMS test, stop battery commission (extension)
17	Reserved for internal usage
18	Start BMS test
19	Reserved for internal usage
20	Reserved for internal usage

Table 8

Note: before changing the „com_service“ it’s a good practice to reset it to 0

Bootloader (update)

If the internal bootloader is active (during update) - the normal communication protocol (COM protocol) is not valid. All requests should be stopped. The bootloader sends periodically (every 500ms) by communication troubles during update the "magic number": **0x50F705AB** (MSBF). Stop all request as soon as possible by detecting the magic number sequence.

Special extensions for data logging

For requesting data logger records the appropriate variables should be used, see RCT Power Datalog Interface description.

Table of ID's

0x104EB6A, // 0: rb485.f_grid[2] Grid phase 3 frequency [Hz]	t_float
0x11F41DB, // 1: power_mng.schedule[0]	t_string

RCT Power GmbH, Line-Eid-Str. 1, 78467 Konstanz, Germany

Geschäftsführer: Thomas Hauser; Eingetragen im Handelsregister Amtsgericht Freiburg i. Br. HRB 713720

power_mng.schedule[0]		
0x16109E1, // 2: grid_mon[0].u_over.time Max. voltage switch-off time level 1 [s]	t_float	
0x1676FA6, // 3: battery.cells_stat[3] battery.cells_stat[3]	t_string	
0x19C0B60, // 4: cs_neg[2] Multiply value of the current sensor 2 by	t_float	
0x2247588, // 5: battery_placeholder[0].cells_stat[2].u_min.value battery_placeholder[0].cells_stat[2].u_min.value	t_float	
0x31A6110, // 6: energy.e_ext_month External month energy [Wh]	t_float	
0x35E64EA, // 7: battery_placeholder[0].module_sn[5] Module 5 Serial Number	t_string	
0x39BDE11, // 8: hw_test.state hw_test.state	t_uint8	
0x3A39CA2, // 9: g_sync.p_ac_load[0] Load household phase 1 [W]	t_float	
0x3D9C51F, // 10: battery.cells_stat[0].u_max.value battery.cells_stat[0].u_max.value	t_float	
0x40385DB, // 11: common_control_bits Bit coded functions	t_uint32	
0x48C9D69, // 12: battery_placeholder[0].cells_stat[1].u_min.value battery_placeholder[0].cells_stat[1].u_min.value	t_float	
0x4EAAA98, // 13: nsm.f_low_entry Entry frequency for P(f) under-frequency mode [Hz]	t_float	
0x528D1D8, // 14: frt.u_min[2] Point 3 voltage [V]	t_float	
0x56162CA, // 15: battery.cells_stat[4].u_min.time battery.cells_stat[4].u_min.time	t_uint32	
0x56417DF, // 16: battery.cells_stat[3].t_max.index battery.cells_stat[3].t_max.index	t_uint8	
0x58F1759, // 17: hw_test.bt_power[6] hw_test.bt_power[6]	t_float	
0x5C7CFB1, // 18: logger.day_egrid_load_log_ts logger.day_egrid_load_log_ts	t_int32	
0x64A60FE, // 19: battery.cells_stat[4].t_max.index battery.cells_stat[4].t_max.index	t_uint8	
0x64E4340, // 20: logger.minutes_ubat_log_ts logger.minutes_ubat_log_ts	t_int32	
0x6A9FFA2, // 21: battery.charged_amp_hours Total charge flow into battery [Ah]	t_float	
0x6E03755, // 22: wifi.ip IP Address	t_string	
0x71B5514, // 23: battery_placeholder[0].cells_stat[3].t_max.index battery_placeholder[0].cells_stat[3].t_max.index	t_uint8	
0x7367B64, // 24: rb485.phase_marker Next phase after phase 1 in Power Switch	t_int16	
0x73C7E5D, // 25: battery_placeholder[0].max_cell_temperature battery_placeholder[0].max_cell_temperature	t_float	
0x74B1EF5, // 26: battery_placeholder[0].cells_stat[3].u_max.index battery_placeholder[0].cells_stat[3].u_max.index	t_uint8	
0x77692DE, // 27: battery.cells_stat[4].u_max.index battery.cells_stat[4].u_max.index	t_uint8	

0x7C61FAD, // 28: adc.u_ref_1_5v[0] Reference voltage 1 [V]	t_uint16
0x8679611, // 29: net.id net.id	t_uint32
0x86C75B0, // 30: battery.stack_software_version[3] Software version stack 3	t_uint32
0x875C906, // 31: hw_test.bt_time[2] hw_test.bt_time[2]	t_float
0x8E81725, // 32: battery_placeholder[0].cells_stat[0].t_max.value battery_placeholder[0].cells_stat[0].t_max.value	t_float
0x95AFAA8, // 33: logger.minutes_ul3_log_ts logger.minutes_ul3_log_ts	t_int32
0x9648B47, // 34: battery_placeholder[0].minimum_discharge_voltage_constant_u t_float Min. discharge voltage [V]	t_float
0x9923C1E, // 35: battery.cells_stat[3].t_min.index battery.cells_stat[3].t_min.index	t_uint8
0xA04CA7F, // 36: g_sync.u_zk_n_avg Negative buffer capacitor voltage [V]	t_float
0xAA372CE, // 37: p_rec_req[1] Required battery to grid power [W]	t_float
0xAFDD6CF, // 38: acc_conv.i_acc_lp_fast Battery current [A]	t_float
0xB94A673, // 39: battery_placeholder[0].cells_stat[6].t_min.time battery_placeholder[0].cells_stat[6].t_min.time	t_uint32
0xBA16A10, // 40: wifi.sockb_protocol Network mode	t_enum
0xC2A7286, // 41: battery_placeholder[0].cells_resist[0] battery_placeholder[0].cells_resist[0]	t_string
0xC3815C2, // 42: net.load_reduction net.load_reduction	t_float
0xC588B75, // 43: energy.e_ext_day_sum energy.e_ext_day_sum	t_float
0xCB5D21B, // 44: dc_conv.dc_conv_struct[1].p_dc_lp Solar generator B power [W]	t_float
0xCBA34B9, // 45: nsm.u_q_u[3] High voltage max. point [V]	t_float
0xCC4BDAA, // 46: detect_phase_shift_enable Enable active island detection	t_bool
0xCFA8BC4, // 47: battery.stack_cycles[1] battery.stack_cycles[1]	t_uint16
0xD658831, // 48: i_bottom_max i_bottom_max	t_float
0xDACF21B, // 49: battery.cells_stat[4] battery.cells_stat[4]	t_string
0xDBD5E77, // 50: battery_placeholder[0].cells_stat[6].u_min.index battery_placeholder[0].cells_stat[6].u_min.index	t_uint8
0xDE3D20D, // 51: battery.status2 Battery extra status	t_int32
0xDF164DE, // 52: logger.day_eb_log_ts logger.day_eb_log_ts	t_int32
0xDF45696, // 53: io_board.io1_polarity Inverted signal on input I/O 1	t_bool
0xE0505B4, // 54: flash_rtc.time_stamp_set	t_uint32


Set date/time		
0xE4AA301, // 55: battery_placeholder[0].cells_stat[6].u_max.index battery_placeholder[0].cells_stat[6].u_max.index	t_uint8	
0xE799A56, // 56: io_board.rse_table[0] K4..K1: 0000	t_float	
0xEC64BA7, // 57: battery_placeholder[0].stack_software_version[3] Software version stack 3	t_uint32	
0xEF60C7E, // 58: battery.cells_stat[3].u_max.value battery.cells_stat[3].u_max.value	t_float	
0xF28E2E1, // 59: energy.e_ext_total_sum energy.e_ext_total_sum	t_float	
0xFA29566, // 60: logger.minutes_ub_log_ts logger.minutes_ub_log_ts	t_int32	
0xFB40090, // 61: io_board.check_rs485_result io_board.check_rs485_result	t_uint8	
0x1025B491, // 62: battery_placeholder[0].maximum_discharge_current Max. discharge current [A]	t_float	
0x10842019, // 63: nsm.cos_phi_p[3][1] Point 4 [cos(ϕ)] (positive = overexcited)	t_float	
0x1089ACA9, // 64: nsm.u_q_u[0] Low voltage min. point [V]	t_float	
0x108FC93D, // 65: max_phase_shift Max. phase shift from 120° position [degrees]	t_float	
0x10970E9D, // 66: energy.e_ac_month Month energy [Wh]	t_float	
0x1156DFD0, // 67: power_mng.battery_power Battery discharge power [W]	t_float	
0x120EC3B4, // 68: battery.cells_stat[4].u_min.index battery.cells_stat[4].u_min.index	t_uint8	
0x126ABC86, // 69: energy.e_grid_load_month Month energy grid load [Wh]	t_float	
0x132AA71E, // 70: logger.minutes_temp2_log_ts logger.minutes_temp2_log_ts	t_int32	
0x1348AB07, // 71: battery.cells[4] battery.cells[4]	t_string	
0x147E8E26, // 72: g_sync.p_ac[1] AC2	t_float	
0x14C0E627, // 73: wifi.password WiFi password	t_string	
0x14FCA232, // 74: nsm.rpm_lock_out_power Reactive Power Mode lock-out power [P/Pn]	t_float	
0x15AB1A61, // 75: power_mng.schedule[2] power_mng.schedule[2]	t_string	
0x162491E8, // 76: battery.module_sn[5] Module 5 Serial Number	t_string	
0x1639B2D8, // 77: battery_placeholder[0].cells_stat[4].u_max.index battery_placeholder[0].cells_stat[4].u_max.index	t_uint8	
0x16A1F844, // 78: battery.bms_sn BMS Serial Number	t_string	
0x16AF2A92, // 79: db.power_board.Current_Mean db.power_board.Current_Mean	t_float	
0x16B28CCA, // 80: adc.u_ref_1_5v[1] Reference voltage 2 [V]	t_uint16	

0x16ED8F8F, // 81: partition[1].last_id partition[1].last_id	t_int32
0x173D81E4, // 82: rb485.version_boot Power Switch bootloader version	t_uint32
0x1781CD31, // 83: battery_placeholder[0].soh SOH (State of Health)	t_float
0x17E3AF97, // 84: db.power_board.adc_p9V_meas db.power_board.adc_p9V_meas	t_float
0x18469762, // 85: battery_placeholder[0].cells_stat[0].u_max.value battery_placeholder[0].cells_stat[0].u_max.value	t_float
0x18BD807D, // 86: battery_placeholder[0].cells_stat[4].t_min.index battery_placeholder[0].cells_stat[4].t_min.index	t_uint8
0x18D1E9E0, // 87: battery.cells_stat[5].u_max.index battery.cells_stat[5].u_max.index	t_uint8
0x18F98B6D, // 88: battery.cells_stat[3].u_min.value battery.cells_stat[3].u_min.value	t_float
0x19608C98, // 89: partition[3].last_id partition[3].last_id	t_int32
0x19B814F2, // 90: logger.year_egrid_feed_log_ts logger.year_egrid_feed_log_ts	t_int32
0x1ABA3EE8, // 91: p_rec_req[0] Required compensation power [W]	t_float
0x1AC87AA0, // 92: g_sync.p_ac_load_sum_lp Load household - external Power[W]	t_float
0x1B39A3A3, // 93: battery.bms_power_version Software version BMS Power	t_uint32
0x1B5445C4, // 94: io_board.check_rse_result io_board.check_rse_result	t_uint16
0x1BFA5A33, // 95: energy.e_grid_load_total_sum energy.e_grid_load_total_sum	t_float
0x1C4A665F, // 96: grid_pll[0].f Grid frequency [Hz]	t_float
0x1D0623D6, // 97: wifi.dns_address DNS address	t_string
0x1D2994EA, // 98: power_mng.soc_charge_power Maintenance charge power [W]	t_float
0x1D49380A, // 99: logger.minutes_eb_log_ts logger.minutes_eb_log_ts	t_int32
0x1D83D2A5, // 100: battery_placeholder[0].cells[4] battery_placeholder[0].cells[4]	t_string
0x1E0EB397, // 101: battery_placeholder[0].cells_stat[6].u_max.value battery_placeholder[0].cells_stat[6].u_max.value	t_float
0x1E5FCA70, // 102: battery.maximum_charge_current Max. charge current [A]	t_float
0x1F44C23A, // 103: battery_placeholder[0].cells_stat[1].t_min.index battery_placeholder[0].cells_stat[1].t_min.index	t_uint8
0x1F73B6A4, // 104: battery.cells_stat[3].t_max.time battery.cells_stat[3].t_max.time	t_uint32
0x1F9CBBF2, // 105: db.power_board.Calibr_Value_Mean db.power_board.Calibr_Value_Mean	t_float
0x1FA192E3, // 106: battery_placeholder[0].cells_resist[4] battery_placeholder[0].cells_resist[4]	t_string
0x1FB3A602, // 107: battery_placeholder[0].cells_stat[2].t_max.value battery_placeholder[0].cells_stat[2].t_max.value	t_float

battery_placeholder[0].cells_stat[2].t_max.value		
0x1FEB2F67, // 108: switch_on_cond.u_min Min. voltage	t_float	
0x2082BFB6, // 109: hw_test.bt_time[9] hw_test.bt_time[9]	t_float	
0x20A3A91F, // 110: battery_placeholder[0].module_sn[4] Module 4 Serial Number	t_string	
0x20FD4419, // 111: prim_sm.island_next_repeat_timeout Next island trial timeout [s]	t_float	
0x21879805, // 112: logger.minutes_eac1_log_ts logger.minutes_eac1_log_ts	t_int32	
0x21961B58, // 113: battery.current Battery current [A]	t_float	
0x21E1A802, // 114: energy.e_dc_month_sum[1] energy.e_dc_month_sum[1]	t_float	
0x21EE7CBB, // 115: rb485.u_l_grid[2] Grid phase 3 voltage [V]	t_float	
0x2266DCB8, // 116: flash_rtc.rtc_mcc_quartz_max_diff Maximum allowed quartz frequency difference between RTC and Microcontroller [ppm]	t_float	
0x226A23A4, // 117: dc_conv.dc_conv_struct[0].u_target MPP on input A [V]	t_float	
0x2295401F, // 118: battery_placeholder[0].cells_stat[3].u_max.time battery_placeholder[0].cells_stat[3].u_max.time	t_uint32	
0x22CC80C6, // 119: frt.u_min_end FRT end undervoltage threshold [V]	t_float	
0x234B4736, // 120: fault[1].flt Error bit field 2	t_uint32	
0x234DD4DF, // 121: switch_on_cond.f_min Min. frequency	t_float	
0x235E0DF5, // 122: battery_placeholder[0].stack_software_version[1] Software version stack 1	t_uint32	
0x236D2178, // 123: frt.t_min[1] Point 2 time [s]	t_float	
0x23D4A386, // 124: battery_placeholder[0].cells_stat[0] battery_placeholder[0].cells_stat[0]	t_string	
0x23E55DA0, // 125: battery.cells_stat[5] battery.cells_stat[5]	t_string	
0x23F525DE, // 126: net.command net.command	t_uint16	
0x24150B85, // 127: g_sync.u_zk_sum_mov_avg Actual DC link voltage [V]	t_float	
0x241CFA0A, // 128: battery_placeholder[0].min_cell_temperature battery_placeholder[0].min_cell_temperature	t_float	
0x241F1F98, // 129: energy.e_dc_day_sum[1] energy.e_dc_day_sum[1]	t_float	
0x24AC4CBB, // 130: battery_placeholder[0].cells_resist[6] battery_placeholder[0].cells_resist[6]	t_string	
0x2545E22D, // 131: g_sync.u_l_rms[2] AC voltage phase 3 [V]	t_float	
0x257B5945, // 132: battery.cells_stat[2].u_min.index battery.cells_stat[2].u_min.index	t_uint8	
0x257B7612, // 133: battery.module_sn[3] Module 3 Serial Number	t_string	

0x26260419, // 134: nsm.cos_phi_p[1][0] Point 2 [P/Pn]	t_float
0x26363AAE, // 135: battery.cells_stat[1].t_max.index battery.cells_stat[1].t_max.index	t_uint8
0x265EACF6, // 136: battery.cells_stat[2].t_max.time battery.cells_stat[2].t_max.time	t_uint32
0x26EFFC2F, // 137: energy.e_grid_feed_year Year energy grid feed-in [Wh]	t_float
0x2703A771, // 138: cs_struct.is_tuned Current sensors are tuned	t_bool
0x27116260, // 139: battery_placeholder[0].cells_stat[5].u_min.value battery_placeholder[0].cells_stat[5].u_min.value	t_float
0x27650FE2, // 140: rb485.version_main Power Switch software version	t_uint32
0x2788928C, // 141: g_sync.p_ac_load[1] Load household phase 2 [W]	t_float
0x27BE51D9, // 142: g_sync.p_ac_sc[0] Grid power phase 1 [W]	t_float
0x27C39CEA, // 143: battery.stack_cycles[6] battery.stack_cycles[6]	t_uint16
0x27C828F4, // 144: energy.e_grid_feed_total_sum energy.e_grid_feed_total_sum	t_float
0x27EC8487, // 145: performance_free[0] performance_free[0]	t_uint32
0x2848A1EE, // 146: grid_offset grid_offset	t_float
0x29BDA75F, // 147: display_struct.brightness Display brightness	t_uint8
0x29CA60F8, // 148: io_board.rse_table[10] K4..K1: 1010	t_float
0x2A30A97E, // 149: battery.stack_cycles[5] battery.stack_cycles[5]	t_uint16
0x2A449E89, // 150: logger.year_log_ts logger.year_log_ts	t_int32
0x2AACCAA7, // 151: battery.max_cell_voltage battery.max_cell_voltage	t_float
0x2AE703F2, // 152: energy.e_dc_day[0] Solar generator A day energy [Wh]	t_float
0x2BC1E72B, // 153: battery.discharged_amp_hours Total charge flow from battery [Ah]	t_float
0x2E06172D, // 154: net.net_tunnel_id net.net_tunnel_id	t_uint32
0x2E0C6220, // 155: io_board.home_relay_sw_off_delay Switching off delay [s]	t_float
0x2E9F3C50, // 156: battery_placeholder[0].cells_stat[0].t_max.index battery_placeholder[0].cells_stat[0].t_max.index	t_uint8
0x2ED89924, // 157: db.power_board.afi_t300 AFI 300 mA switching off time [s]	t_float
0x2ED8A639, // 158: battery_placeholder[0].cells_stat[2].u_min.time battery_placeholder[0].cells_stat[2].u_min.time	t_uint32
0x2F0A6B15, // 159: logger.month_ea_log_ts logger.month_ea_log_ts	t_int32
0x2F3C1D7D, // 160: energy.e_load_day	t_float

Household day energy [Wh]	
0x2F84A0A9, // 161: battery_placeholder[0].cells[2] battery_placeholder[0].cells[2]	t_string
0x3044195F, // 162: grid_mon[1].u_under.time Min. voltage switch-off time level 2 [s]	t_float
0x31413485, // 163: battery_placeholder[0].cells_stat[5].u_min.index battery_placeholder[0].cells_stat[5].u_min.index	t_uint8
0x314C13EB, // 164: battery_placeholder[0].cells_stat[5].u_max.value battery_placeholder[0].cells_stat[5].u_max.value	t_float
0x315D1490, // 165: power_mng.bat_empty_full Bit 0 - battery was empty, bit 1 - battery was full	t_uint8
0x31ED1B75, // 166: modbus.mode RS485 working mode	t_enum
0x32CD0DB3, // 167: nsm.cos_phi_p[0][1] Point 1 [$\cos(\phi)$] (positive = overexcited)	t_float
0x32DCA605, // 168: frt.u_max[0] Point 1 voltage [V]	t_float
0x331D0689, // 169: battery.cells_stat[2].t_max.value battery.cells_stat[2].t_max.value	t_float
0x336415EA, // 170: battery.cells_stat[0].t_max.time battery.cells_stat[0].t_max.time	t_uint32
0x3390CC2F, // 171: switch_on_cond.test_time_fault Switching on time after any grid fault [s]	t_float
0x33F76B78, // 172: nsm.p_u[0][1] Point 1 voltage [V]	t_float
0x34A164E7, // 173: battery.cells_stat[0] battery.cells_stat[0]	t_string
0x34E33726, // 174: battery.cells_stat[2].u_max.index battery.cells_stat[2].u_max.index	t_uint8
0x34ECA9CA, // 175: logger.year_eb_log_ts logger.year_eb_log_ts	t_int32
0x3500F1E8, // 176: net.index net.index	t_int8
0x3503B92D, // 177: battery.cells_stat[3].u_max.time battery.cells_stat[3].u_max.time	t_uint32
0x3515F4A0, // 178: nsm.p_u[3][1] Point 4 voltage [V]	t_float
0x360BDE8A, // 179: nsm.startup_grad Startup gradient [P/(Pn*s)]	t_float
0x36214C57, // 180: net.prev_k net.prev_k	t_float
0x362346D4, // 181: switch_on_cond.max_rnd_test_time_fault Max additional random switching on time after any grid fault [s]	t_float
0x3623D82A, // 182: prim_sm.island_flag Grid-separated	t_uint16
0x365D12DA, // 183: p_rec_req[2] Required Pac [W]	t_float
0x36A9E9A6, // 184: power_mng.use_grid_power_enable Utilize external Inverter energy	t_bool
0x374B5DD6, // 185: battery_placeholder[0].cells_stat[6].u_min.time battery_placeholder[0].cells_stat[6].u_min.time	t_uint32
0x37F9D5CA, // 186: fault[0].flt Error bit field 1	t_uint32

0x381B8BF9, // 187: battery.soh SOH (State of Health)	t_float
0x383A3614, // 188: db.power_board.afi_i60 AFI 60 mA threshold [A]	t_float
0x38789061, // 189: nsm.f_low_rise_grad_storage Power rise gradient for P(f) under-frequency mode with battery [1/Pn*Hz]	t_float
0x3903A5E9, // 190: flash_rtc.flag_time_auto_switch Automatically adjust clock for daylight saving time	t_bool
0x3906A1D0, // 191: logger.minutes_eext_log_ts logger.minutes_eext_log_ts	t_int32
0x392D1BEE, // 192: wifi.connect_to_server wifi.connect_to_server	t_uint8
0x39AD4639, // 193: battery_placeholder[0].cells_stat[5].u_min.time battery_placeholder[0].cells_stat[5].u_min.time	t_uint32
0x3A0EA5BE, // 194: power_spring_up power_spring_up	t_float
0x3A3050E6, // 195: grid_lt.threshold Max. voltage [V]	t_float
0x3A35D491, // 196: battery_placeholder[0].cells_stat[2].u_max.value battery_placeholder[0].cells_stat[2].u_max.value	t_float
0x3A444FC6, // 197: g_sync.s_ac_lp[0] Apparent power phase 1 [VA]	t_float
0x3A7D5F53, // 198: battery.cells_stat[1].u_max.value battery.cells_stat[1].u_max.value	t_float
0x3A873343, // 199: energy.e_ac_day_sum energy.e_ac_day_sum	t_float
0x3A9D2680, // 200: energy.e_ext_year_sum energy.e_ext_year_sum	t_float
0x3AA565FC, // 201: net.package net.package	t_string
0x3AFEF139, // 202: prim_sm.is_thin_layer Thin-film solar module	t_bool
0x3B0C6A53, // 203: bat_mng_struct.profile_pdc_max bat_mng_struct.profile_pdc_max	t_string
0x3B5F6B9D, // 204: rb485.f_wr[0] Power Storage phase 1 frequency [Hz]	t_float
0x3B7FCD47, // 205: fault[2].flt Error bit field 3	t_uint32
0x3BA1B77B, // 206: battery.cells_stat[3].t_min.value battery.cells_stat[3].t_min.value	t_float
0x3C24F3E8, // 207: inv_struct.cosinus_phi cos 	t_float
0x3C705F61, // 208: io_board.rse_table[8] K4..K1: 1000	t_float
0x3C87C4F5, // 209: energy.e_grid_feed_day Day energy grid feed-in [Wh]	t_float
0x3CA8E8D0, // 210: hw_test.bt_time[0] hw_test.bt_time[0]	t_float
0x3CB1EF01, // 211: grid_mon[0].u_under.threshold Min. voltage level 1 [V]	t_float
0x3D789979, // 212: hw_test.bt_power[7] hw_test.bt_power[7]	t_float
0x3DBCC6B4, // 213: io_board.rse_table[6]	t_float

K4..K1: 0110		
0x3E25C391, // 214: bat_mng_struct.bat_calib_soc_thresh Part of max historical SOC for battery calibration in advance	t_float	
0x3E722B43, // 215: grid_mon[1].f_under.threshold Min. frequency level 2 [Hz]	t_float	
0x3E728842, // 216: power_spring_bat power_spring_bat	t_float	
0x3EFEB931, // 217: db.power_board.relays_state db.power_board.relays_state	t_uint16	
0x3F98F58A, // 218: battery.cells_stat[5].t_max.index battery.cells_stat[5].t_max.index	t_uint8	
0x400F015B, // 219: g_sync.p_acc_lp Battery power [W]	t_float	
0x4077335D, // 220: g_sync.s_ac_lp[1] Apparent power phase 2 [VA]	t_float	
0x40B07CA4, // 221: power_mng.schedule[6] power_mng.schedule[6]	t_string	
0x40FF01B7, // 222: battery.cells[6] battery.cells[6]	t_string	
0x41744E11, // 223: frt.u_min[0] Point 1 voltage [V]	t_float	
0x41B11ECF, // 224: battery.cells_stat[3].u_min.index battery.cells_stat[3].u_min.index	t_uint8	
0x428CCF46, // 225: battery.cells_stat[5].u_min.value battery.cells_stat[5].u_min.value	t_float	
0x431509D1, // 226: logger.month_eoload_log_ts logger.month_eoload_log_ts	t_int32	
0x43257820, // 227: g_sync.p_ac[0] AC1	t_float	
0x437B8122, // 228: rb485.available Power Switch is available	t_bool	
0x4397D078, // 229: nsm.cos_phi_p[1][1] Point 2 [$\cos(\phi)$] (positive = overexcited)	t_float	
0x43CD0B6F, // 230: nsm.pf_delay Delay time after P(f) [s]	t_float	
0x43F16F7E, // 231: flash_state Flash state	t_uint16	
0x43FF47C3, // 232: db.power_board.afi_t60 AFI 60 mA switching off time [s]	t_float	
0x442A3409, // 233: battery.cells_stat[4].t_min.time battery.cells_stat[4].t_min.time	t_uint32	
0x4443C661, // 234: battery.cells_stat[0].t_max.index battery.cells_stat[0].t_max.index	t_uint8	
0x44D4C533, // 235: energy.e_grid_feed_total Total energy grid feed-in [Wh]	t_float	
0x4539A6D4, // 236: can_bus.bms_update_response[0] can_bus.bms_update_response[0]	t_uint32	
0x465DDB50, // 237: battery_placeholder[0].cells_stat[2].t_min.value battery_placeholder[0].cells_stat[2].t_min.value	t_float	
0x46635546, // 238: net.n_descendants Number of descendant slaves	t_int8	
0x4686E044, // 239: battery_placeholder[0].cells_stat[1].u_min.index battery_placeholder[0].cells_stat[1].u_min.index	t_uint8	

0x46892579, // 240: flash_param.write_cycles Write cycles of flash parameters	t_uint32
0x46C3625D, // 241: battery_placeholder[0].cells_stat[2] battery_placeholder[0].cells_stat[2]	t_string
0x474F80D5, // 242: iso_struct.Rn Insulation resistance on negative DC input [Ohm]	t_float
0x4764F9EE, // 243: battery_placeholder[0].cells_stat[3].t_max.value battery_placeholder[0].cells_stat[3].t_max.value	t_float
0x47A1DACA, // 244: power_mng.schedule[8] power_mng.schedule[8]	t_string
0x485AD749, // 245: g_sync.u_ptp_rms[1] Phase to phase voltage 2 [V]	t_float
0x488052BA, // 246: logger.minutes_ul2_log_ts logger.minutes_ul2_log_ts	t_int32
0x48D73FA5, // 247: g_sync.i_dr_lp[2] Current phase 3 (average) [A]	t_float
0x494FE156, // 248: power_spring_offset power_spring_offset	t_float
0x495BF0B6, // 249: energy.e_dc_year_sum[0] energy.e_dc_year_sum[0]	t_float
0x4992E65A, // 250: update_is_allowed_id update_is_allowed_id	t_uint8
0x4A61BAEE, // 251: nsm.p_u[3][0] Point 4 P/Pn	t_float
0x4AAEB0D2, // 252: battery_placeholder[0].cells_stat[1] battery_placeholder[0].cells_stat[1]	t_string
0x4AE96C12, // 253: dc_conv.dc_conv_struct[1].mpp.mpp_step MPP search step on input B [V]	t_float
0x4B51A539, // 254: battery.prog_sn battery.prog_sn	t_string
0x4BC0F974, // 255: buf_v_control.power_reduction_max_solar Solar plant peak power [Wp]	t_float
0x4BE02BB7, // 256: energy.e_load_day_sum energy.e_load_day_sum	t_float
0x4C12C4C7, // 257: cs_neg[1] Multiply value of the current sensor 1 by	t_float
0x4C14CC7C, // 258: logger.year_ea_log_ts logger.year_ea_log_ts	t_int32
0x4C2A7CDC, // 259: nsm.cos_phi_p[2][1] Point 3 [$\cos(\phi)$] (positive = overexcited)	t_float
0x4C374958, // 260: nsm.startup_grad_after_fault Startup gradient after fault [P/(Pn*s)]	t_float
0x4CB7C0DC, // 261: battery.min_cell_voltage battery.min_cell_voltage	t_float
0x4D684EF2, // 262: battery_placeholder[0].cells[0] battery_placeholder[0].cells[0]	t_string
0x4D985F33, // 263: battery.cells_stat[5].u_max.value battery.cells_stat[5].u_max.value	t_float
0x4DB1B91E, // 264: switch_on_cond.f_max Max. frequency	t_float
0x4DC372A0, // 265: battery_placeholder[0].cells_stat[4].u_max.value battery_placeholder[0].cells_stat[4].u_max.value	t_float
0x4E04DD55, // 266: battery.soc_update_since	t_float

battery.soc_update_since		
0x4E0C56F2, // 267: flash_rtc.rtc_mcc_quartz_ppm_difference Quartz frequency difference between RTC and <u>Microcontroller</u> [ppm]	t_float	
0x4E2B42A4, // 268: hw_test.bt_power[0] hw_test.bt_power[0]	t_float	
0x4E3CB7F8, // 269: phase_3_mode 3-phase feed in	t_bool	
0x4E49AEC5, // 270: g_sync.p_ac_sum Real power [W]	t_float	
0x4E699086, // 271: battery.module_sn[4] Module 4 Serial Number	t_string	
0x4E77B2CE, // 272: hw_test.bt_cycle hw_test.bt_cycle	t_uint8	
0x4E9D95A6, // 273: logger.year_eext_log_ts logger.year_eext_log_ts	t_int32	
0x4EE8DB78, // 274: energy.e_load_year_sum energy.e_load_year_sum	t_float	
0x4F330E08, // 275: io_board.io2_usage Digital I/O 2 usage	t_enum	
0x4F735D10, // 276: db.temp2 Heat sink (battery actuator) temperature [°C]	t_float	
0x4FC53F19, // 277: battery_placeholder[0].module_sn[3] Module 3 Serial Number	t_string	
0x4FEDC1BE, // 278: battery_placeholder[0].cells_stat[5].t_min.value battery_placeholder[0].cells_stat[5].t_min.value	t_float	
0x4FF8CCE2, // 279: battery_placeholder[0].stack_software_version[5] Software version stack 5	t_uint32	
0x501A162D, // 280: battery.cells_resist[5] battery.cells_resist[5]	t_string	
0x50514732, // 281: battery.cells_stat[6].u_min.index battery.cells_stat[6].u_min.index	t_uint8	
0x508FCE78, // 282: adc.u_ref_1_5v[3] Reference voltage 4 [V]	t_uint16	
0x50B441C1, // 283: logger.minutes_ea_log_ts logger.minutes_ea_log_ts	t_int32	
0x5151D84C, // 284: prim_sm.island_reset_retrials_counter_time Reset island trials counter in [min] (by 0 not used)	t_float	
0x518C7BBE, // 285: battery.cells_stat[5].u_min.time battery.cells_stat[5].u_min.time	t_uint32	
0x51E5377D, // 286: battery_placeholder[0].stack_cycles[1] battery_placeholder[0].stack_cycles[1]	t_uint16	
0x5293B668, // 287: logger.minutes_soc_log_ts logger.minutes_soc_log_ts	t_int32	
0x53656F42, // 288: battery_placeholder[0].cells_stat[2].u_max.index battery_placeholder[0].cells_stat[2].u_max.index	t_uint8	
0x537C719F, // 289: battery.cells_stat[0].t_max.value battery.cells_stat[0].t_max.value	t_float	
0x53886C09, // 290: wifi.connect_to_service wifi.connect_to_service	t_uint8	
0x53EF7649, // 291: nsm.p_u[0][0] Point 1 P/Pn	t_float	
0x5411CE1B, // 292: logger.minutes_ul1_log_ts logger.minutes_ul1_log_ts	t_int32	

0x5438B68E, // 293: grid_mon[1].u_over.threshold Max. voltage level 2 [V]	t_float
0x54829753, // 294: p_rec_lim[1] Max. battery to grid power [W]	t_float
0x54B4684E, // 295: g_sync.u_l_rms[1] AC voltage phase 2 [V]	t_float
0x54DBC202, // 296: io_board.rse_table[12] K4..K1: 1100	t_float
0x554D8FEE, // 297: logger.minutes_eac2_log_ts logger.minutes_eac2_log_ts	t_int32
0x5570401B, // 298: battery.stored_energy Total energy flow into battery [Wh]	t_float
0x55C22966, // 299: g_sync.s_ac[2] Apparent power phase 3 [VA]	t_float
0x55DDF7BA, // 300: battery.max_cell_temperature battery.max_cell_temperature	t_float
0x5673D737, // 301: wifi.connect_to_wifi wifi.connect_to_wifi	t_bool
0x57429627, // 302: wifi.authentication_method WiFi authentication method	t_string
0x576D2A08, // 303: battery_placeholder[0].cells_stat[3].t_min.time battery_placeholder[0].cells_stat[3].t_min.time	t_uint32
0x57945EE4, // 304: battery_placeholder[0].maximum_charge_current Max. charge current [A]	t_float
0x58378BD0, // 305: hw_test.bt_time[3] hw_test.bt_time[3]	t_float
0x5847E59E, // 306: battery.maximum_charge_voltage_constant_u Max. charge voltage [V]	t_float
0x5867B3BE, // 307: io_board.rse_table[2] K4..K1: 0010	t_float
0x58C1A946, // 308: io_board.check_state io_board.check_state	t_uint8
0x592B13DF, // 309: power_mng.schedule[4] power_mng.schedule[4]	t_string
0x59358EB2, // 310: power_mng.maximum_charge_voltage Max. battery charge voltage [V]	t_float
0x5939EC5D, // 311: battery.module_sn[6] Module 6 Serial Number	t_string
0x5952E5E6, // 312: wifi.mask Netmask	t_string
0x5A120CE4, // 313: battery.cells_stat[1].t_max.time battery.cells_stat[1].t_max.time	t_uint32
0x5A316247, // 314: wifi.mode WiFi mode	t_string
0x5A9EFFF0, // 315: battery.stack_cycles[4] battery.stack_cycles[4]	t_uint16
0x5AF50FD7, // 316: battery.cells_stat[4].t_min.value battery.cells_stat[4].t_min.value	t_float
0x5B10CE81, // 317: power_mng.is_heiphoss HeiPhoss mode	t_uint8
0x5BA122A5, // 318: battery.stack_cycles[2] battery.stack_cycles[2]	t_uint16
0x5BB8075A, // 319: dc_conv.dc_conv_struct[1].u_sg_lp	t_float

Solar generator B voltage [V]	
0x5BD2DB45, // 320: io_board.io1_s0_imp_per_kwh Number of impulses per kWh for S0 signal on I/O 1	t_int16
0x5C93093B, // 321: battery_placeholder[0].status2 Battery extra status	t_int32
0x5CD75669, // 322: db.power_board.afi_t150 AFI 150 mA switching off time [s]	t_float
0x5D0CDCF0, // 323: p_rec_available[2] Available Pac [W]	t_float
0x5D1B0835, // 324: net.use_network_filter net.use_network_filter	t_bool
0x5D34D09D, // 325: logger.month_egrid_load_log_ts logger.month_egrid_load_log_ts	t_int32
0x5E540FB2, // 326: net.update_slaves Activate aut. update slaves	t_bool
0x5E942C62, // 327: dc_conv.dc_conv_struct[1].mpp.fixed_voltage Fixed voltage Solar generator B [V]	t_float
0x5EE03C45, // 328: io_board.alarm_home_relay_mode Multifunctional relay usage	t_enum
0x5EF54372, // 329: battery_placeholder[0].cells_stat[0].u_max.index battery_placeholder[0].cells_stat[0].u_max.index	t_uint8
0x5F33284E, // 330: prim_sm.state Inverter status	t_uint8
0x6002891F, // 331: g_sync.p_ac_sc_sum Grid power (ext. sensors) [W]	t_float
0x60435F1C, // 332: battery_placeholder[0].cells[6] battery_placeholder[0].cells[6]	t_string
0x60749E5E, // 333: battery.cells_stat[6].u_min.time battery.cells_stat[6].u_min.time	t_uint32
0x60A9A532, // 334: logger.day_eext_log_ts logger.day_eext_log_ts	t_int32
0x612F7EAB, // 335: g_sync.s_ac[1] Apparent power phase 2 [VA]	t_float
0x61EAC702, // 336: battery.cells_stat[0].t_min.value battery.cells_stat[0].t_min.value	t_float
0x6213589B, // 337: battery.cells_stat[6].u_min.value battery.cells_stat[6].u_min.value	t_float
0x6279F2A3, // 338: db.power_board.version_boot PIC bootloader software version	t_uint32
0x62B8940B, // 339: dc_conv.start_voltage Inverter DC-voltage start value [V]	t_float
0x62D645D9, // 340: battery.cells[5] battery.cells[5]	t_string
0x62FBE7DC, // 341: energy.e_grid_load_total Total energy grid load [Wh]	t_float
0x63476DBE, // 342: g_sync.u_ptp_rms[0] Phase to phase voltage 1 [V]	t_float
0x6383DEA9, // 343: battery_placeholder[0].cells_stat[1].t_max.value battery_placeholder[0].cells_stat[1].t_max.value	t_float
0x6388556C, // 344: battery.stack_software_version[0] Software version stack 0	t_uint32
0x6445D856, // 345: battery.cells_stat[1].u_min.index battery.cells_stat[1].u_min.index	t_uint8

0x6476A836, // 346: dc_conv.dc_conv_struct[0].mpp.enable_scan Enable rescan for global MPP on solar generator A	t_bool
0x649B10DA, // 347: battery.cells_resist[0] battery.cells_resist[0]	t_string
0x650C1ED7, // 348: g_sync.i_dr_eff[1] Current phase 2 [A]	t_float
0x652B7536, // 349: battery_placeholder[0].cells_stat[3].t_min.index battery_placeholder[0].cells_stat[3].t_min.index	t_uint8
0x6599E3D3, // 350: power_mng.schedule[3] power_mng.schedule[3]	t_string
0x65A44A98, // 351: flash_mem flash_mem	t_string
0x65B624AB, // 352: energy.e_grid_feed_month Month energy grid feed-in [Wh]	t_float
0x65EED11B, // 353: battery.voltage Battery voltage [V]	t_float
0x663F1452, // 354: power_mng.n_batteries power_mng.n_batteries	t_uint8
0x664A1326, // 355: io_board.rse_table[14] K4..K1: 1110	t_float
0x669D02FE, // 356: logger.minutes_eac_log_ts logger.minutes_eac_log_ts	t_int32
0x6709A2F4, // 357: energy.e_ac_year_sum energy.e_ac_year_sum	t_float
0x672552DC, // 358: power_mng.bat_calib_days_in_advance Battery calibration days in advance	t_uint8
0x6743CCCE, // 359: battery_placeholder[0].cells_stat[6].t_max.index battery_placeholder[0].cells_stat[6].t_max.index	t_uint8
0x675776B1, // 360: dc_conv.dc_conv_struct[1].u_target MPP on input B [V]	t_float
0x67BF3003, // 361: display_struct.display_dir Rotate display	t_bool
0x67C0A2F5, // 362: net.slave_p_total net.slave_p_total	t_float
0x682CDDA1, // 363: power_mng.battery_type Battery type	t_enum
0x6830F6E4, // 364: io_board.rse_table[9] K4..K1: 1001	t_float
0x68BA92E1, // 365: io_board.io2_s0_imp_per_kwh Number of impulses per kWh for S0 signal on I/O 2	t_int16
0x68BC034D, // 366: parameter_file Norm	t_string
0x68EEFD3D, // 367: energy.e_dc_total[1] Solar generator B total energy [Wh]	t_float
0x690C32D2, // 368: battery_placeholder[0].module_sn[0] Module 0 Serial Number	t_string
0x6974798A, // 369: battery.stack_software_version[6] Software version stack 6	t_uint32
0x69AA598A, // 370: can_bus.requested_id can_bus.requested_id	t_int32
0x69B8FF28, // 371: battery.cells[2] battery.cells[2]	t_string
0x6B5A56C2, // 372: logger.month_eb_log_ts	t_int32

logger.month_eb_log_ts		
0x6BA10831, // 373: db.power_board.afi_i30 AFI 30 mA threshold [A]	t_float	
0x6BBDC7C8, // 374: line_mon.u_max Max line voltage [V]	t_float	
0x6BFF1AF4, // 375: hw_test.bt_power[2] hw_test.bt_power[2]	t_float	
0x6C03F5ED, // 376: battery_placeholder[0].bms_power_version Software version BMS Power	t_uint32	
0x6C10E96A, // 377: battery_placeholder[0].cells_stat[0].u_min.time battery_placeholder[0].cells_stat[0].u_min.time	t_uint32	
0x6C243F71, // 378: modbus.address RS485 address	t_uint8	
0x6C2D00E4, // 379: io_board.rse_table[1] K4..K1: 0001	t_float	
0x6C44F721, // 380: i_dc_max Max. DC-component of I _{ac} [A]	t_float	
0x6CFCD774, // 381: energy.e_dc_year_sum[1] energy.e_dc_year_sum[1]	t_float	
0x6D5318C8, // 382: cs_map[1] Associate current sensor 1 with phase L	t_uint8	
0x6D639C25, // 383: battery_placeholder[0].cells_stat[0].t_min.value battery_placeholder[0].cells_stat[0].t_min.value	t_float	
0x6D7C0BF4, // 384: wifi.sockb_port Port	t_int32	
0x6DB1FDDC, // 385: battery.cells_stat[4].u_min.value battery.cells_stat[4].u_min.value	t_float	
0x6DCC4097, // 386: net.master_timeout net.master_timeout	t_float	
0x6E1C5B78, // 387: g_sync.p_ac_lp[1] AC power phase 2 [W]	t_float	
0x6E24632E, // 388: battery.cells_stat[5].u_max.time battery.cells_stat[5].u_max.time	t_uint32	
0x6E3336A8, // 389: battery_placeholder[0].cells_stat[5].t_max.index battery_placeholder[0].cells_stat[5].t_max.index	t_uint8	
0x6E491B50, // 390: battery.maximum_charge_voltage Max. charge voltage [V]	t_float	
0x6F3876BC, // 391: logger.error_log_time_stamp Time stamp for error log reading	t_int32	
0x6FB2E2BF, // 392: db.power_board.afi_i150 AFI 150 mA threshold [A]	t_float	
0x6FD36B32, // 393: rb485.f_wr[1] Power Storage phase 2 frequency [Hz]	t_float	
0x6FF4BD55, // 394: energy.e_ext_month_sum energy.e_ext_month_sum	t_float	
0x701A0482, // 395: dc_conv.dc_conv_struct[0].enabled Solar generator A connected	t_bool	
0x70349444, // 396: battery.cells_stat[1].t_min.index battery.cells_stat[1].t_min.index	t_uint8	
0x70A2AF4F, // 397: battery.bat_status battery.bat_status	t_int32	
0x70BD7C46, // 398: logger.year_eac_log_ts logger.year_eac_log_ts	t_int32	

0x70E28322, // 399: grid_mon[0].f_under.time Min. frequency switch-off time level 1 [s]	t_float
0x71196579, // 400: battery.cells_stat[5].t_min.index battery.cells_stat[5].t_min.index	t_uint8
0x71277E71, // 401: frt.u_min_begin FRT begin undervoltage threshold [V]	t_float
0x71465EAF, // 402: nsm.cos_phi_ts Time const for filter [s]	t_float
0x715C84A1, // 403: adc.u_ref_1_5v[2] Reference voltage 3 [V]	t_uint16
0x71765BD8, // 404: battery.status Battery status	t_int32
0x71B70DCE, // 405: hw_test.bt_power[4] hw_test.bt_power[4]	t_float
0x71CB0B57, // 406: battery.cells_resist[1] battery.cells_resist[1]	t_string
0x71E10B51, // 407: g_sync.p_ac_lp[0] AC power phase 1 [W]	t_float
0x7232F7AF, // 408: nsm.apm nsm.apm	t_enum
0x7268CE4D, // 409: battery.inv_cmd battery.inv_cmd	t_uint32
0x72ACC0BF, // 410: logger.minutes_ua_log_ts logger.minutes_ua_log_ts	t_int32
0x7301A5A7, // 411: flash_rtc.time_stamp_factory Production date	t_uint32
0x73489528, // 412: battery.module_sn[2] Module 2 Serial Number	t_string
0x73E3ED49, // 413: prim_sm.island_max_trials Max island trials	t_uint16
0x742966A6, // 414: db.power_board.afi_i300 AFI 300 mA threshold [A]	t_float
0x74FD4609, // 415: battery.cells_stat[2] battery.cells_stat[2]	t_string
0x751E80CA, // 416: prim_sm.island_reset_retrials_operation_time Reset island trials counter if island OK in [s]	t_float
0x75898A45, // 417: battery_placeholder[0].cells_stat[5].t_max.time battery_placeholder[0].cells_stat[5].t_max.time	t_uint32
0x75AE19ED, // 418: hw_test.hw_switch_time hw_test.hw_switch_time	t_float
0x7689BE6A, // 419: io_board.home_relay_sw_on_delay Switching on delay [s]	t_float
0x76C9A0BD, // 420: logger.minutes_soc_targ_log_ts logger.minutes_soc_targ_log_ts	t_int32
0x76CAA9BF, // 421: wifi.encryption_algorithm wifi.encryption_algorithm	t_string
0x770A6E7C, // 422: battery.cells_stat[0].u_max.index battery.cells_stat[0].u_max.index	t_uint8
0x777DC0EB, // 423: iso_struct.r_min Minimum allowed insulation resistance [Ohm]	t_float
0x77A9480F, // 424: battery_placeholder[0].minimum_discharge_voltage Min. discharge voltage [V]	t_float
0x77DD4364, // 425: hw_test.bt_time[5]	t_float

hw_test.bt_time[5]		
0x77E5CEF1, // 426: battery_placeholder[0].stack_software_version[0]	Software version stack 0	t_uint32
0x78228507, // 427: battery_placeholder[0].stack_cycles[6]	battery_placeholder[0].stack_cycles[6]	t_uint16
0x7839EBCB, // 428: battery_placeholder[0].cells_stat[3].u_min.time	battery_placeholder[0].cells_stat[3].u_min.time	t_uint32
0x7924ABD9, // 429: inverter_sn	Serial number	t_string
0x792897C9, // 430: battery_placeholder[0].cells_stat[4].t_min.time	battery_placeholder[0].cells_stat[4].t_min.time	t_uint32
0x792A7B79, // 431: io_board.s0_direction	S0 inputs single or bidirectional	t_enum
0x7940547B, // 432: inv_struct.force_dh	inv_struct.force_dh	t_bool
0x7946D888, // 433: i_dc_slow_time	Time for slow DC-component of Iac [s]	t_float
0x79C0A724, // 434: energy.e_ac_total_sum	energy.e_ac_total_sum	t_float
0x79D7D617, // 435: battery_placeholder[0].current	Battery current [A]	t_float
0x79E66CDF, // 436: battery_placeholder[0].cells_stat[6].t_min.index	battery_placeholder[0].cells_stat[6].t_min.index	t_uint8
0x7A5C91F8, // 437: nsm.p_u[1][0]	Point 2 P/Pn	t_float
0x7A67E33B, // 438: can_bus.bms_update_response[1]	can_bus.bms_update_response[1]	t_uint32
0x7A9091EA, // 439: rb485.u_l_grid[1]	Grid phase 2 voltage [V]	t_float
0x7AB9B045, // 440: energy.e_dc_month[1]	Solar generator B month energy [Wh]	t_float
0x7AE87E39, // 441: partition[2].last_id	partition[2].last_id	t_int32
0x7AF0AD03, // 442: power_mng.schedule[9]	power_mng.schedule[9]	t_string
0x7AF779C1, // 443: nsm.pu_mode	P(U) mode 0: Pn 1: Pload	t_bool
0x7B1F7FBE, // 444: wifi.gateway	Gateway	t_string
0x7B8E811E, // 445: battery_placeholder[0].cells_stat[6]	battery_placeholder[0].cells_stat[6]	t_string
0x7BF3886B, // 446: battery_placeholder[0].stack_cycles[2]	battery_placeholder[0].stack_cycles[2]	t_uint16
0x7C0827C5, // 447: partition[5].last_id	partition[5].last_id	t_int32
0x7C556C7A, // 448: io_board.io2_polarity	Inverted signal on input I/O 2	t_bool
0x7C78CBAC, // 449: g_sync.q_ac_sum_lp	Reactive power [var]	t_float
0x7C863EDB, // 450: battery_placeholder[0].cells[3]	battery_placeholder[0].cells[3]	t_string
0x7D839AE6, // 451: battery_placeholder[0].cells_resist[2]	battery_placeholder[0].cells_resist[2]	t_string

0x7DA7D8B6, // 452: db.power_board.version_main PIC software version	t_uint32
0x7DDE352B, // 453: wifi.sockb_ip wifi.sockb_ip	t_string
0x7E096024, // 454: energy.e_load_total_sum energy.e_load_total_sum	t_float
0x7E590128, // 455: battery.cells_stat[0].u_max.time battery.cells_stat[0].u_max.time	t_uint32
0x7E75B17A, // 456: nsm.q_u_max_u_high_rel Q _{max} at upper voltage level relative to S _{max} (positive = overexcited)	t_float
0x7F42BB82, // 457: battery.cells_stat[6].u_max.index battery.cells_stat[6].u_max.index	t_uint8
0x7F813D73, // 458: fault[3].flt Error bit field 4	t_uint32
0x7FF6252C, // 459: battery.cells_stat[5].t_max.time battery.cells_stat[5].t_max.time	t_uint32
0x804A3266, // 460: battery.cells_stat[6].u_max.value battery.cells_stat[6].u_max.value	t_float
0x80835476, // 461: db.power_board.adc_p5V_W_meas db.power_board.adc_p5V_W_meas	t_float
0x8128228D, // 462: battery_placeholder[0].cells_stat[1].u_max.value battery_placeholder[0].cells_stat[1].u_max.value	t_float
0x812E5ADD, // 463: energy.e_dc_total_sum[1] energy.e_dc_total_sum[1]	t_float
0x8160539D, // 464: battery.cells_stat[4].t_max.value battery.cells_stat[4].t_max.value	t_float
0x81AE960B, // 465: energy.e_dc_month[0] Solar generator A month energy [Wh]	t_float
0x81AF854E, // 466: nsm.pu_use P(U) active	t_bool
0x82258C01, // 467: cs_neg[0] Multiply value of the current sensor 0 by	t_float
0x82CD1525, // 468: grid_mon[1].u_under.threshold Min. voltage level 2 [V]	t_float
0x82E3C121, // 469: g_sync.q_ac[1] Reactive power phase 2 [var]	t_float
0x8320B84C, // 470: io_board.rse_data_delay Delay for new K4..K1 data [s]	t_float
0x8352F9DD, // 471: battery_placeholder[0].cells_stat[4].t_min.value battery_placeholder[0].cells_stat[4].t_min.value	t_float
0x83A5333A, // 472: nsm.cos_phi_p[0][0] Point 1 [P/Pn]	t_float
0x83BBEF0B, // 473: frt.u_max_begin FRT begin overvoltage threshold [V]	t_float
0x84ABE3D8, // 474: energy.e_grid_feed_year_sum energy.e_grid_feed_year_sum	t_float
0x85886E2E, // 475: p_rec_lim[0] Max. compensation power [W]	t_float
0x8594D11E, // 476: battery_placeholder[0].module_sn[6] Module 6 Serial Number	t_string
0x86782D58, // 477: hw_test.bt_power[9] hw_test.bt_power[9]	t_float
0x867DEF7D, // 478: energy.e_grid_load_day	t_float

Day energy grid load [Wh]		
0x872F380B, // 479: io_board.load_set Dummy household load [W]	t_float	
0x87E4387A, // 480: current_sensor_max Power Sensor current range [A]	t_float	
0x8822EF35, // 481: battery_placeholder[0].stack_software_version[2] Software version stack 2	t_uint32	
0x883DE9AB, // 482: g_sync.s_ac_lp[2] Apparent power phase 3 [VA]	t_float	
0x885BB57E, // 483: battery.cells_stat[6].t_min.value battery.cells_stat[6].t_min.value	t_float	
0x887D43C4, // 484: g_sync.i_dr_lp[0] Current phase 1 (average) [A]	t_float	
0x889DC27F, // 485: battery.cells_stat[0].u_min.value battery.cells_stat[0].u_min.value	t_float	
0x88BBF8CB, // 486: battery.cells_stat[5].t_min.value battery.cells_stat[5].t_min.value	t_float	
0x88C9707B, // 487: io_board.rse_table[15] K4..K1: 1111	t_float	
0x88DEBCFE, // 488: nsm.q_u_max_u_high Qmax at upper voltage level [var] (positive = overexcited)	t_float	
0x88DFDE8B, // 489: frt.u_max_end FRT end overvoltage threshold [V]	t_float	
0x88F36D45, // 490: io_board.rse_data Actual K4..K1 data	t_uint8	
0x89B21223, // 491: frt.t_max[0] Point 1 time [s]	t_float	
0x89B25F4B, // 492: battery.stack_cycles[3] battery.stack_cycles[3]	t_uint16	
0x89EE3EB5, // 493: g_sync.i_dr_eff[0] Current phase 1 [A]	t_float	
0x8A18539B, // 494: g_sync.u_zk_sum_avg DC link voltage [V]	t_float	
0x8AFD1410, // 495: battery_placeholder[0].stack_cycles[4] battery_placeholder[0].stack_cycles[4]	t_uint16	
0x8B4BE168, // 496: battery_placeholder[0].soc SOC (State of charge)	t_float	
0x8B9FF008, // 497: battery.soc_target Target SOC	t_float	
0x8BB08839, // 498: battery.cells_stat[6].t_min.time battery.cells_stat[6].t_min.time	t_uint32	
0x8C6E28E4, // 499: battery_placeholder[0].cells_stat[2].t_max.time battery_placeholder[0].cells_stat[2].t_max.time	t_uint32	
0x8CA00014, // 500: wifi.result WiFi result	t_int8	
0x8D33B6BC, // 501: nsm.f_low_exit Exit frequency for P(f) under-frequency mode [Hz]	t_float	
0x8D8E19F7, // 502: line_mon.u_min Min line voltage [V]	t_float	
0x8DD1C728, // 503: dc_conv.dc_conv_struct[1].mpp.enable_scan Enable rescan for global MPP on solar generator B	t_bool	
0x8DFFDD33, // 504: battery.cells_stat[3].u_min.time battery.cells_stat[3].u_min.time	t_uint32	

0x8E41FC47, // 505: iso_struct.Rp Insulation resistance on positive DC input [Ohm]	t_float
0x8EBF9574, // 506: power_mng.soc_min_island Min SOC target (island)	t_float
0x8EC23427, // 507: battery.cells_stat[4].u_max.time battery.cells_stat[4].u_max.time	t_uint32
0x8EC4116E, // 508: display_struct.blink Display blinking enable	t_bool
0x8EF6FBBD, // 509: battery.cells[1] battery.cells[1]	t_string
0x8EF9C9B8, // 510: battery.cells_stat[6].t_max.time battery.cells_stat[6].t_max.time	t_uint32
0x8F0FF9F3, // 511: p_rec_available[1] Available battery to grid power [W]	t_float
0x8FC89B10, // 512: com_service COM service	t_enum
0x902AFAFB, // 513: battery.temperature Battery temperature [°C]	t_float
0x903FE89E, // 514: hw_test.bt_time[8] hw_test.bt_time[8]	t_float
0x905F707B, // 515: rb485.f_wr[2] Power Storage phase 3 frequency [Hz]	t_float
0x9061EA7B, // 516: grid_lt.granularity Resolution	t_float
0x907CD1DF, // 517: wifi.connect_service_max_duration Service connection max duration [s]	t_int32
0x90832471, // 518: battery.cells_stat[1].u_max.time battery.cells_stat[1].u_max.time	t_uint32
0x9095FD74, // 519: battery_placeholder[0].cells[5] battery_placeholder[0].cells[5]	t_string
0x90B53336, // 520: temperature.sink_temp_power_reduction Heat sink temperature target [°C]	t_float
0x90C2AC13, // 521: battery_placeholder[0].stack_cycles[3] battery_placeholder[0].stack_cycles[3]	t_uint16
0x90F123FA, // 522: io_board.io1_usage Digital I/O 1 usage	t_enum
0x915CD4A4, // 523: grid_mon[1].f_over.threshold Max. frequency level 2 [Hz]	t_float
0x91617C58, // 524: g_sync.p_ac_grid_sum_lp Total grid power [W]	t_float
0x917E3622, // 525: energy.e_ext_year External year energy [Wh]	t_float
0x91C325D9, // 526: battery.cells_stat[0].t_min.time battery.cells_stat[0].t_min.time	t_uint32
0x91FB68CD, // 527: battery.cells_stat[6].t_max.value battery.cells_stat[6].t_max.value	t_float
0x920AFF34, // 528: battery_placeholder[0].cells_stat[1].t_max.index battery_placeholder[0].cells_stat[1].t_max.index	t_uint8
0x9214A00C, // 529: hw_test.booster_test_index hw_test.booster_test_index	t_uint8
0x921997EE, // 530: logger.month_egrid_feed_log_ts logger.month_egrid_feed_log_ts	t_int32
0x9247DB99, // 531: logger.minutes_egrid_load_log_ts	t_int32

logger.minutes_egridd_load_log_ts	
0x929394B7, // 532: svnversion_last_known svnversion_last_known	t_string
0x92BC682B, // 533: g_sync.i_dr_eff[2] Current phase 3 [A]	t_float
0x933F9A24, // 534: grid_mon[0].f_over.time Max. frequency switch-off time level 1 [s]	t_float
0x934E64E9, // 535: switch_on_cond.u_max Max. voltage	t_float
0x9350FE02, // 536: frt.u_max[2] Point 3 voltage [V]	t_float
0x93971C36, // 537: frt.t_max[2] Point 3 time [s]	t_float
0x93C0C2E2, // 538: power_mng.bat_calib_regularity Battery calibration interval [days]	t_uint32
0x93E6918D, // 539: nsm.f_exit Exit frequency for P(f) over-frequency mode [Hz]	t_float
0x93F976AB, // 540: rb485.u_l_grid[0] Grid phase 1 voltage [V]	t_float
0x940569AC, // 541: hw_test.bt_time[6] hw_test.bt_time[6]	t_float
0x947DDC38, // 542: battery_placeholder[0].cells_stat[0].t_min.index battery_placeholder[0].cells_stat[0].t_min.index	t_uint8
0x9486134F, // 543: battery_placeholder[0].cells_stat[1].t_max.time battery_placeholder[0].cells_stat[1].t_max.time	t_uint32
0x9558AD8A, // 544: rb485.f_grid[0] Grid phase1 frequency [Hz]	t_float
0x959930BF, // 545: battery.soc SOC (State of charge)	t_float
0x95E1E844, // 546: battery_placeholder[0].cells_stat[2].t_min.time battery_placeholder[0].cells_stat[2].t_min.time	t_uint32
0x961C8261, // 547: battery_placeholder[0].cells_stat[4].u_max.time battery_placeholder[0].cells_stat[4].u_max.time	t_uint32
0x96629BB9, // 548: can_bus.bms_update_state can_bus.bms_update_state	t_uint8
0x9680077F, // 549: nsm.cos_phi_p[2][0] Point 3 [P/Pn]	t_float
0x96E32D11, // 550: flash_param.erase_cycles Erase cycles of flash parameter	t_uint32
0x972B3029, // 551: power_mng.stop_discharge_voltage_buffer Stop discharge voltage buffer [V]	t_float
0x97997C93, // 552: power_mng.soc_max Max SOC target	t_float
0x97DC2ECB, // 553: battery_placeholder[0].cells[1] battery_placeholder[0].cells[1]	t_string
0x97E203F9, // 554: power_mng.is_grid power_mng.is_grid	t_bool
0x97E3A6F2, // 555: power_mng.u_acc_lp Battery voltage (inverter) [V]	t_float
0x980C5525, // 556: battery_placeholder[0].max_cell_voltage battery_placeholder[0].max_cell_voltage	t_float
0x98ACC1B8, // 557: io_board.rse_table[4] K4..K1: 0100	t_float

0x99396810, // 558: battery.module_sn[1] Module 1 Serial Number	t_string
0x993C06F6, // 559: battery.cells_resist[3] battery.cells_resist[3]	t_string
0x9981F1AC, // 560: db.power_board.adc_m9V_meas db.power_board.adc_m9V_meas	t_float
0x99EE89CB, // 561: power_mng.power_lim_src_index Power limit source	t_enum
0x9A33F9B7, // 562: power_mng.schedule[5] power_mng.schedule[5]	t_string
0x9A51A23B, // 563: logger.log_rate Data log resolution [s]	t_uint16
0x9A67600D, // 564: p_rec_lim[2] Pac max. [W]	t_float
0x9AAA9CAA, // 565: battery_placeholder[0].stack_cycles[5] battery_placeholder[0].stack_cycles[5]	t_uint16
0x9B92023F, // 566: io_board.rse_table[7] K4..K1: 0111	t_float
0x9C75BD89, // 567: frt.t_min[0] Point 1 time [s]	t_float
0x9C8FE559, // 568: pas.period pas.period	t_uint32
0x9D785E8C, // 569: battery.bms_software_version Software version BMS Master	t_uint32
0x9DC927AA, // 570: bat_mng_struct.profile_load bat_mng_struct.profile_load	t_string
0x9E1A88F5, // 571: dc_conv.dc_conv_struct[0].mpp.fixed_voltage Fixed voltage Solar generator A [V]	t_float
0x9E314430, // 572: battery.cells_stat[2].u_max.time battery.cells_stat[2].u_max.time	t_uint32
0x9F52F968, // 573: power_mng.feed_asymmetrical Allow asymmetrical feed	t_bool
0xA10D9A4B, // 574: battery.min_cell_temperature battery.min_cell_temperature	t_float
0xA1266D6B, // 575: line_mon.time_lim Switch off time line voltage [s]	t_float
0xA12BE39C, // 576: energy.e_load_month_sum energy.e_load_month_sum	t_float
0xA12E9B43, // 577: phase_marker Next phase after phase 1	t_int16
0xA1D2B565, // 578: wifi.service_port wifi.service_port	t_int32
0xA23FE8B9, // 579: battery_placeholder[0].cells_stat[6].t_min.value battery_placeholder[0].cells_stat[6].t_min.value	t_float
0xA2F87161, // 580: battery_placeholder[0].cells_stat[0].u_max.time battery_placeholder[0].cells_stat[0].u_max.time	t_uint32
0xA305214D, // 581: logger.buffer logger.buffer	t_string
0xA3393749, // 582: io_board.check_start io_board.check_start	t_uint8
0xA33D0954, // 583: nsm.q_u_hysteresis Curve with hysteresis	t_bool
0xA3E48B21, // 584: battery.cells_stat[2].t_min.value	t_float

battery.cells_stat[2].t_min.value	
0xA40906BF, // 585: battery.stack_software_version[4] Software version stack 4	t_uint32
0xA5044DCD, // 586: nsm.p_u[2][0] Point 3 P/Pn	t_float
0xA5341F4A, // 587: energy.e_grid_feed_month_sum energy.e_grid_feed_month_sum	t_float
0xA54C4685, // 588: battery.stack_software_version[1] Software version stack 1	t_uint32
0xA59C8428, // 589: energy.e_ext_total External total energy [Wh]	t_float
0xA60082A9, // 590: logger.minutes_egrid_feed_log_ts logger.minutes_egrid_feed_log_ts	t_int32
0xA616B022, // 591: battery.soc_target_low SOC target low	t_float
0xA6271C2E, // 592: grid_mon[0].u_over.threshold Max. voltage level 1 [V]	t_float
0xA6871A4D, // 593: battery.cells_stat[4].t_min.index battery.cells_stat[4].t_min.index	t_uint8
0xA6C4FD4A, // 594: battery.stack_cycles[0] battery.stack_cycles[0]	t_uint16
0xA7447FC4, // 595: temperature.bat_temp_power_reduction Battery actuator temperature target [°C]	t_float
0xA76AE9CA, // 596: relays.bits_real relays.bits_real	t_uint16
0xA7C708EB, // 597: logger.minutes_eload_log_ts logger.minutes_eload_log_ts	t_int32
0xA7DBD28C, // 598: battery.cells_stat[2].t_max.index battery.cells_stat[2].t_max.index	t_uint8
0xA7F4123B, // 599: battery_placeholder[0].stack_software_version[6] Software version stack 6	t_uint32
0xA7FA5C5D, // 600: power_mng.u_acc_mix_lp Battery voltage [V]	t_float
0xA7FE5C0C, // 601: battery.cells_stat[2].t_min.index battery.cells_stat[2].t_min.index	t_uint8
0xA81176D0, // 602: battery_placeholder[0].cells_stat[1].u_min.time battery_placeholder[0].cells_stat[1].u_min.time	t_uint32
0xA83F291F, // 603: battery_placeholder[0].cells_stat[6].u_min.value battery_placeholder[0].cells_stat[6].u_min.value	t_float
0xA8FEAEB9, // 604: battery_placeholder[0].cells_resist[5] battery_placeholder[0].cells_resist[5]	t_string
0xA9033880, // 605: battery.used_energy Total energy flow from battery [Wh]	t_float
0xA95AD038, // 606: grid_mon[0].f_under.threshold Min. frequency level 1 [Hz]	t_float
0xA95EE214, // 607: power_mng.model.bat_power_change power_mng.model.bat_power_change	t_float
0xA9CF517D, // 608: power_spring_down power_spring_down	t_float
0xAA911BEE, // 609: battery_placeholder[0].cells_stat[4].t_max.value battery_placeholder[0].cells_stat[4].t_max.value	t_float
0xAA9AA253, // 610: dc_conv.dc_conv_struct[1].p_dc Solar generator B power [W]	t_float

0xAACAC898, // 611: battery.cells_stat[4].t_max.time battery.cells_stat[4].t_max.time	t_uint32
0xAACE057A, // 612: io_board.io1_s0_min_duration Minimum S0 signal duration on I/O 1 [s]	t_float
0xABA015FC, // 613: battery_placeholder[0].module_sn[1] Module 1 Serial Number	t_string
0xAC2E2A56, // 614: io_board.rse_table[5] K4..K1: 0101	t_float
0xACF7666B, // 615: battery.efficiency Battery efficiency (used energy / stored energy)	t_float
0xAE99F87A, // 616: battery_placeholder[0].cells_stat[5].t_min.time battery_placeholder[0].cells_stat[5].t_min.time	t_uint32
0xAEF76FA1, // 617: power_mng.minimum_discharge_voltage Min. battery discharge voltage [V]	t_float
0xAF64D0FE, // 618: energy.e_dc_year[0] Solar generator A year energy [Wh]	t_float
0xB0041187, // 619: g_sync.u_sg_avg[1] Solar generator B voltage [V]	t_float
0xB0307591, // 620: db.power_board.status Power board status	t_uint16
0xB082C4D7, // 621: hw_test.bt_power[5] hw_test.bt_power[5]	t_float
0xB0EBE75A, // 622: battery.minimum_discharge_voltage Min. discharge voltage [V]	t_float
0xB0FA4D23, // 623: acc_conv.i_charge_max Max. battery converter charge current [A]	t_float
0xB130B8D6, // 624: battery_placeholder[0].cells_stat[1].t_min.time battery_placeholder[0].cells_stat[1].t_min.time	t_uint32
0xB1D1BE71, // 625: osci_struct.cmd_response_time osci_struct.cmd_response_time	t_float
0xB1D465C7, // 626: battery_placeholder[0].cells_stat[4].u_min.value battery_placeholder[0].cells_stat[4].u_min.value	t_float
0xB1EF67CE, // 627: energy.e_ac_total Total energy [Wh]	t_float
0xB20D1AD6, // 628: logger.day_egrid_feed_log_ts logger.day_egrid_feed_log_ts	t_int32
0xB221BCFA, // 629: g_sync.p_ac_sc[2] Grid power phase 3 [W]	t_float
0xB228EC94, // 630: battery_placeholder[0].cells_stat[3].t_max.time battery_placeholder[0].cells_stat[3].t_max.time	t_uint32
0xB238942F, // 631: last_successfull_flash_op last_successfull_flash_op	t_int16
0xB298395D, // 632: dc_conv.dc_conv_struct[0].u_sg_lp Solar generator A voltage [V]	t_float
0xB2FB9A90, // 633: bat_mng_struct.k_trust How fast the actual prediction can be trusted	t_float
0xB399B5B3, // 634: battery_placeholder[0].cells_stat[4].u_min.index battery_placeholder[0].cells_stat[4].u_min.index	t_uint8
0xB403A7E6, // 635: battery_placeholder[0].soc_update_since battery_placeholder[0].soc_update_since	t_float
0xB408E40A, // 636: acc_conv.i_acc_lp_slow acc_conv.i_acc_lp_slow	t_float
0xB4222BDE, // 637: wifi.state	t_uint8

wifi.state		
0xB45FE275, // 638: p_rec_available[0]	t_float	Available compensation power [W]
0xB4E053D4, // 639: battery.cells_stat[1].u_min.value	t_float	battery.cells_stat[1].u_min.value
0xB5317B78, // 640: dc_conv.dc_conv_struct[0].p_dc	t_float	Solar generator A power [W]
0xB55BA2CE, // 641: g_sync.u_sg_avg[0]	t_float	Solar generator A voltage [V]
0xB57B59BD, // 642: battery.ah_capacity	t_float	Battery capacity [Ah]
0xB5EDA8EC, // 643: battery_placeholder[0].cells_stat[3].u_max.value	t_float	battery_placeholder[0].cells_stat[3].u_max.value
0xB6623608, // 644: power_mng.bat_next_calib_date	t_uint32	Next battery calibration
0xB69171C4, // 645: db.power_board.Current_AC_RMS	t_float	db.power_board.Current_AC_RMS
0xB70D1703, // 646: battery_placeholder[0].cells_stat[5].u_max.index	t_uint8	battery_placeholder[0].cells_stat[5].u_max.index
0xB76E2B4C, // 647: nsm.cos_phi_const	t_float	Cos_phi constant value (positive = overexcited)
0xB7B2967F, // 648: energy.e_dc_total_sum[0]	t_float	energy.e_dc_total_sum[0]
0xB7C85C51, // 649: wifi.use_ethernet	t_bool	wifi.use_ethernet
0xB7FEA209, // 650: wifi.connect_service_timestamp	t_int32	Service auto disconnect time
0xB81FB399, // 651: battery.cells_stat[2].u_min.time	t_uint32	battery.cells_stat[2].u_min.time
0xB836B50C, // 652: dc_conv.dc_conv_struct[1].rescan_correction	t_float	Last global rescan MPP correction on input B [V]
0xB84A38AB, // 653: battery.soc_target_high	t_float	SOC target high
0xB84FDCF9, // 654: adc.u_acc	t_float	Battery voltage (inverter) [V]
0xB851FA70, // 655: io_board.rse_table[11]	t_float	K4..K1: 1011
0xB98C8194, // 656: nsm.min_cos_phi	t_float	Minimum allowed cos(phi) [0..1]
0xB9928C51, // 657: g_sync.p_ac_lp[2]	t_float	AC power phase 3 [W]
0xB9A026F9, // 658: energy.e_ext_day	t_float	External day energy [Wh]
0xB9E09F78, // 659: battery.cells_stat[5].u_min.index	t_uint8	battery.cells_stat[5].u_min.index
0xBA046C03, // 660: battery_placeholder[0].cells_stat[5].t_max.value	t_float	battery_placeholder[0].cells_stat[5].t_max.value
0xBA8B8515, // 661: dc_conv.dc_conv_struct[0].mpp.mpp_step	t_float	MPP search step on input A [V]
0xBB302278, // 662: battery.cells_stat[1].t_min.time	t_uint32	battery.cells_stat[1].t_min.time
0xBB617E51, // 663: nsm.u_q_u[1]	t_float	Low voltage max. point [V]

0xBBE6B9DF, // 664: io_board.p_rse_rise_grad Power rise gradient [P/Pn/s]	t_float
0xBCA77559, // 665: g_sync.q_ac[2] Reactive power phase 3 [var]	t_float
0xBCC6F92F, // 666: io_board.home_relay_threshold Switching on threshold [W]	t_float
0xBD008E29, // 667: power_mng.battery_power_extern Battery target power [W] (positive = discharge)	t_float
0xBD3A23C3, // 668: power_mng.soc_charge SOC min maintenance charge	t_float
0xBD4147B0, // 669: can_bus.set_cell_resist can_bus.set_cell_resist	t_uint32
0xBD55905F, // 670: energy.e_ac_day Day energy [Wh]	t_float
0xBD55D796, // 671: energy.e_dc_year[1] Solar generator B year energy [Wh]	t_float
0xBD95C46C, // 672: battery_placeholder[0].ah_capacity Battery capacity [Ah]	t_float
0xBDE3BF0A, // 673: battery.cells_stat[6].t_max.index battery.cells_stat[6].t_max.index	t_uint8
0xBDFE5547, // 674: io_board.rse_table[3] K4..K1: 0011	t_float
0xBF9B6042, // 675: svnversion_factory Control software factory version	t_string
0xBFFF3CAD, // 676: net.n_slaves net.n_slaves	t_uint8
0xC03462F6, // 677: g_sync.p_ac[2] AC3	t_float
0xC04A5F3A, // 678: battery_placeholder[0].bms_software_version Software version BMS Master	t_uint32
0xC0680302, // 679: battery.cells_stat[2].t_min.time battery.cells_stat[2].t_min.time	t_uint32
0xC07E02CE, // 680: nsm.q_u_sel Voltage selection	t_enum
0xC0A7074F, // 681: net.slave_data net.slave_data	t_string
0xC0B7C4D2, // 682: db.power_board.afi_t30 AFI 30 mA switching off time [s]	t_float
0xC0CC81B6, // 683: energy.e_ac_year Year energy [Wh]	t_float
0xC0DF2978, // 684: battery.cycles Battery charge / discharge cycles	t_int32
0xC198B25B, // 685: g_sync.u_zk_p_avg Positive buffer capacitor voltage [V]	t_float
0xC1C82889, // 686: hw_test.bt_power[1] hw_test.bt_power[1]	t_float
0xC1D051EC, // 687: display_struct.variate_contrast display_struct.variate_contrast	t_uint8
0xC24E85D0, // 688: db.core_temp Core temperature [°C]	t_float
0xC3352B17, // 689: nsm.rpm nsm.rpm	t_enum
0xC36675D4, // 690: i_ac_max_set	t_float

Maximum AC throttle current [A]		
0xC3A3F070, // 691: i_ac_extern_connected Current sensors detected	t_bool	
0xC3C7325E, // 692: hw_test.bt_time[4] hw_test.bt_time[4]	t_float	
0xC3DD7850, // 693: partition[6].last_id partition[6].last_id	t_int32	
0xC40D5688, // 694: prim_sm.state_source prim_sm.state_source	t_uint32	
0xC42F5807, // 695: battery.cells_stat[1].u_max.index battery.cells_stat[1].u_max.index	t_uint8	
0xC46E9CA4, // 696: nsm.u_lock_out Cos phi(P) lock out voltage [V]	t_float	
0xC4D87E96, // 697: prim_sm.island_retrials Island trials counter	t_uint16	
0xC4FA4E33, // 698: frt.u_min[1] Point 2 voltage [V]	t_float	
0xC55EF32E, // 699: logger.year_egrid_load_log_ts logger.year_egrid_load_log_ts	t_int32	
0xC56A1346, // 700: battery_placeholder[0].cells_stat[4].t_max.index battery_placeholder[0].cells_stat[4].t_max.index	t_uint8	
0xC642B9D6, // 701: acc_conv.i_discharge_max Max. battery converter discharge current [A]	t_float	
0xC66665E8, // 702: battery_placeholder[0].temperature Battery temperature [°C]	t_float	
0xC66A522B, // 703: hw_test.bt_time[1] hw_test.bt_time[1]	t_float	
0xC6DA81A0, // 704: battery.cells_stat[6].u_max.time battery.cells_stat[6].u_max.time	t_uint32	
0xC707102E, // 705: hw_test.bt_power[3] hw_test.bt_power[3]	t_float	
0xC71155B5, // 706: battery_placeholder[0].cells_stat[2].t_min.index battery_placeholder[0].cells_stat[2].t_min.index	t_uint8	
0xC717D1FB, // 707: iso_struct.Riso Total insulation resistance [Ohm]	t_float	
0xC7459513, // 708: power_mng.force_inv_class Change inverter class	t_enum	
0xC7605E16, // 709: io_board.s0_sum io_board.s0_sum	t_float	
0xC7D3B479, // 710: energy.e_load_year Household year energy [Wh]	t_float	
0xC7E85F32, // 711: battery_placeholder[0].cells_stat[4].t_max.time battery_placeholder[0].cells_stat[4].t_max.time	t_uint32	
0xC8609C8E, // 712: battery.cells[3] battery.cells[3]	t_string	
0xC88EB032, // 713: battery.cells_stat[0].u_min.time battery.cells_stat[0].u_min.time	t_uint32	
0xC8BA1729, // 714: battery.stack_software_version[2] Software version stack 2	t_uint32	
0xC8E56803, // 715: battery_placeholder[0].maximum_charge_voltage Max. charge voltage [V]	t_float	
0xC937D38D, // 716: battery_placeholder[0].stack_cycles[0] battery_placeholder[0].stack_cycles[0]	t_uint16	

0xC9900716, // 717: power_mng.is_island_only Island without power switch support	t_bool
0xC9D76279, // 718: energy.e_dc_day_sum[0] energy.e_dc_day_sum[0]	t_float
0xCA4E0C03, // 719: battery_placeholder[0].cells_stat[5].u_max.time battery_placeholder[0].cells_stat[5].u_max.time	t_uint32
0xCA6D6472, // 720: logger.day_eoload_log_ts logger.day_eoload_log_ts	t_int32
0xCABC44CA, // 721: g_sync.s_ac[0] Apparent power phase 1 [VA]	t_float
0xCB1B3B10, // 722: io_board.io2_s0_min_duration Minimum S0 signal duration on I/O 2 [s]	t_float
0xCB78F611, // 723: frt.t_max[1] Point 2 time [s]	t_float
0xCB85C397, // 724: battery_placeholder[0].cells_stat[3].u_min.value battery_placeholder[0].cells_stat[3].u_min.value	t_float
0xCB9E1E6C, // 725: nsm.Q_const Constant reactive power [var] (positive = overexcited)	t_float
0xCBBEEB21, // 726: battery_placeholder[0].cells_stat[2].u_max.time battery_placeholder[0].cells_stat[2].u_max.time	t_uint32
0xCBDAD315, // 727: logger.minutes_ebat_log_ts logger.minutes_ebat_log_ts	t_int32
0xCBEC8200, // 728: hw_test.timer2 hw_test.timer2	t_float
0xCCB51399, // 729: nsm.q_u_max_u_low Qmax at lower voltage level [var] (positive = overexcited)	t_float
0xCD8EDAD3, // 730: battery_placeholder[0].cells_stat[3].t_min.value battery_placeholder[0].cells_stat[3].t_min.value	t_float
0xCE266F0F, // 731: power_mng.soc_min Min SOC target	t_float
0xCE49EB86, // 732: battery_placeholder[0].cells_stat[2].t_max.index battery_placeholder[0].cells_stat[2].t_max.index	t_uint8
0xCF005C54, // 733: prim_sm.phase_3_mode prim_sm.phase_3_mode	t_bool
0xCF053085, // 734: g_sync.u_l_rms[0] AC voltage phase 1 [V]	t_float
0xCF096A6B, // 735: battery_placeholder[0].stack_software_version[4] Software version stack 4	t_uint32
0xD0C47326, // 736: battery.cells_stat[1].t_min.value battery.cells_stat[1].t_min.value	t_float
0xD143A391, // 737: can_bus.set_cell_v_t can_bus.set_cell_v_t	t_uint32
0xD166D94D, // 738: flash_rtc.time_stamp Actual date/time	t_uint32
0xD197CBE0, // 739: power_mng.stop_charge_current Stop charge current [A]	t_float
0xD1DFC969, // 740: power_mng.soc_target_set Force SOC target	t_float
0xD1F9D017, // 741: battery_placeholder[0].cells_stat[4].u_min.time battery_placeholder[0].cells_stat[4].u_min.time	t_uint32
0xD2DEA4B1, // 742: battery_placeholder[0].cells_stat[5].t_min.index battery_placeholder[0].cells_stat[5].t_min.index	t_uint8
0xD3085D80, // 743: net.soc_av	t_float

net.soc_av		
0xD3E94E6B, // 744: logger.minutes_temp_bat_log_ts logger.minutes_temp_bat_log_ts	t_int32	
0xD3F492EB, // 745: battery_placeholder[0].cells_stat[0].t_max.time battery_placeholder[0].cells_stat[0].t_max.time	t_uint32	
0xD451EF88, // 746: cs_map[2] Associate current sensor 2 with phase L	t_uint8	
0xD45913EC, // 747: io_board.rse_table[13] K4..K1: 1101	t_float	
0xD4C4A941, // 748: hw_test.bt_time[7] hw_test.bt_time[7]	t_float	
0xD5205A45, // 749: net.slave_timeout net.slave_timeout	t_float	
0xD536E7E9, // 750: frt.u_max[1] Point 2 voltage [V]	t_float	
0xD5567470, // 751: partition[4].last_id partition[4].last_id	t_int32	
0xD5790CE1, // 752: wifi.use_wifi Enable Wi-Fi Access Point	t_bool	
0xD580567B, // 753: nsm.u_lock_in Cos phi(P) lock in voltage [V]	t_float	
0xD60E7A2F, // 754: battery.cells_stat[1].u_min.time battery.cells_stat[1].u_min.time	t_uint32	
0xD81471DF, // 755: battery_placeholder[0].cells_stat[6].t_max.value battery_placeholder[0].cells_stat[6].t_max.value	t_float	
0xD82F2D0B, // 756: battery_placeholder[0].cells_stat[3].u_min.index battery_placeholder[0].cells_stat[3].u_min.index	t_uint8	
0xD83DC6AC, // 757: wifi.server_port wifi.server_port	t_int32	
0xD876A4AC, // 758: battery_placeholder[0].cells_stat[0].u_min.index battery_placeholder[0].cells_stat[0].u_min.index	t_uint8	
0xD884AF95, // 759: nsm.pf_desc_grad Power decrease gradient for P(f) mode [P/(Pn*s)]	t_float	
0xD9D66B76, // 760: energy.e_grid_load_year_sum energy.e_grid_load_year_sum	t_float	
0xD9E721A5, // 761: grid_lt.timeframe Timeframe	t_float	
0xD9F9F35B, // 762: acc_conv.state_slow acc_conv.state_slow	t_uint8	
0xDA207111, // 763: energy.e_grid_load_month_sum energy.e_grid_load_month_sum	t_float	
0xDABD323E, // 764: osci_struct.error Communication error	t_int16	
0xDAC7DD86, // 765: io_board.p_rse_desc_grad Power descent gradient [P/Pn/s]	t_float	
0xDB11855B, // 766: dc_conv.dc_conv_struct[0].p_dc_lp Solar generator A power [W]	t_float	
0xDB2D69AE, // 767: g_sync.p_ac_sum_lp AC power [W]	t_float	
0xDB45ABD0, // 768: dc_conv.dc_conv_struct[0].rescan_correction Last global rescan MPP correction on input A [V]	t_float	
0xDB62DCB7, // 769: net.n_devices net.n_devices	t_uint8	

0xDC667958, // 770: power_mng.state Battery state machine	t_uint8
0xDCA1CF26, // 771: g_sync.s_ac_sum_lp Apparent power [VA]	t_float
0xDCAC0EA9, // 772: g_sync.i_dr_lp[1] Current phase 2 (average) [A]	t_float
0xDD5930A2, // 773: battery.cells_stat[0].t_min.index battery.cells_stat[0].t_min.index	t_uint8
0xDD90A328, // 774: flash_rtc.time_stamp_update Last update date	t_uint32
0xDDD1C2D0, // 775: <u>svnversion</u> Control software version	t_string
0xDE17F021, // 776: energy.e_grid_load_year Year energy grid load [Wh]	t_float
0xDE68F62D, // 777: bat_mng_struct.profile_pext bat_mng_struct.profile_pext	t_string
0xDE9CBCB0, // 778: battery.cells_stat[5].t_max.value battery.cells_stat[5].t_max.value	t_float
0xDEE1957F, // 779: battery.cells_resist[4] battery.cells_resist[4]	t_string
0xDF0A735C, // 780: battery.maximum_discharge_current Max. discharge current [A]	t_float
0xDF6EA121, // 781: bat_mng_struct.profile_pdc bat_mng_struct.profile_pdc	t_string
0xDFB53AF3, // 782: db.power_board.Current_Mean_Mean_AC db.power_board.Current_Mean_Mean_AC	t_float
0xDFF966E3, // 783: battery.cells_stat[6].t_min.index battery.cells_stat[6].t_min.index	t_uint8
0xE04C3900, // 784: logger.day_eac_log_ts logger.day_eac_log_ts	t_int32
0xE0E16E63, // 785: cs_map[0] Associate current sensor 0 with phase L	t_uint8
0xE14B8679, // 786: i_dc_slow_max Max. slow DC-component of I _{ac} [A]	t_float
0xE14F1CBA, // 787: battery_placeholder[0].cells_stat[4] battery_placeholder[0].cells_stat[4]	t_string
0xE19C8B79, // 788: battery_placeholder[0].cells_resist[1] battery_placeholder[0].cells_resist[1]	t_string
0xE1F49459, // 789: frt.t_min[2] Point 3 time [s]	t_float
0xE24B00BD, // 790: power_mng.schedule[1] power_mng.schedule[1]	t_string
0xE271C6D2, // 791: nsm.u_q_u[2] High voltage min. point [V]	t_float
0xE29C24EB, // 792: logger.minutes_eac3_log_ts logger.minutes_eac3_log_ts	t_int32
0xE31F8B17, // 793: prim_sm.Uzk_pump_grad[0] start power [W]	t_float
0xE3F4D1DF, // 794: acc_conv.i_max Max. battery converter current [A]	t_float
0xE49BE3ED, // 795: nsm.pf_rise_grad Power increase gradient after P(f) restriction [P/(Pn*s)]	t_float
0xE4DC040A, // 796: logger.month_eext_log_ts	t_int32

logger.month_eext_log_ts	
0xE51B041C, // 797: battery_placeholder[0].maximum_charge_voltage_constant_u t_float	Max. charge voltage [V]
0xE52B89FA, // 798: io_board.home_relay_off_threshold t_float	Switching off threshold [W]
0xE5FBCC6F, // 799: logger.year_e-load_log_ts t_int32	logger.year_e-load_log_ts
0xE6248312, // 800: hw_test.bt_power[8] t_float	hw_test.bt_power[8]
0xE635A6C4, // 801: battery_placeholder[0].module_sn[2] t_string	Module 2 Serial Number
0xE63A3529, // 802: flash_result t_uint16	Flash result
0xE6F1CB83, // 803: nsm.pu_ts t_float	Time const for filter [s]
0xE7177DEE, // 804: battery.cells_stat[2].u_max.value t_float	battery.cells_stat[2].u_max.value
0xE7B0E692, // 805: battery.bat_impedance.impedance_fine t_float	Battery circuit impedance
0xE87B1F4B, // 806: battery_placeholder[0].cells_stat[0].u_min.value t_float	battery_placeholder[0].cells_stat[0].u_min.value
0xE94C2EFC, // 807: g_sync.q_ac[0] t_float	Reactive power phase 1 [var]
0xE952FF2D, // 808: nsm.q_u_max_u_low_rel t_float	Q _{max} at lower voltage level relative to S _{max} (positive = overexcited)
0xE96F1844, // 809: io_board.s0_external_power t_float	io_board.s0_external_power
0xE9BBF6E4, // 810: power_mng.amp_hours_measured t_float	Measured battery capacity [Ah]
0xEA399EA8, // 811: battery_placeholder[0].min_cell_voltage t_float	battery_placeholder[0].min_cell_voltage
0xEA77252E, // 812: battery.minimum_discharge_voltage_constant_u t_float	Min. discharge voltage [V]
0xEAEEB3CA, // 813: energy.e_dc_month_sum[0] t_float	energy.e_dc_month_sum[0]
0xEB4C2597, // 814: battery.cells_resist[6] t_string	battery.cells_resist[6]
0xEB7773BF, // 815: nsm.p_u[1][1] t_float	Point 2 voltage [V]
0xEB7BCB93, // 816: battery_placeholder[0].bms_sn t_string	BMS Serial Number
0xEBC62737, // 817: android_description t_string	Device name
0xEBF7A4E8, // 818: grid_mon[0].f_over.threshold t_float	Max. frequency level 1 [Hz]
0xECABB6CF, // 819: switch_on_cond.test_time t_float	Test time
0xEE049B1F, // 820: nsm.pf_hysteresis t_bool	Hysteresis mode
0xEEA3F59B, // 821: battery.stack_software_version[5] t_uint32	Software version stack 5
0xEEC44AA0, // 822: battery_placeholder[0].cells_stat[2].u_min.index t_uint8	battery_placeholder[0].cells_stat[2].u_min.index

0xEECDFEFC, // 823: battery.cells_stat[2].u_min.value battery.cells_stat[2].u_min.value	t_float
0xEF89568B, // 824: grid_mon[0].u_under.time Min. voltage switch-off time level 1 [s]	t_float
0xEFD3EC8A, // 825: battery.cells_stat[5].t_min.time battery.cells_stat[5].t_min.time	t_uint32
0xEFF4B537, // 826: energy.e_load_total Household total energy [Wh]	t_float
0xF03133E2, // 827: partition[0].last_id partition[0].last_id	t_int32
0xF044EDA0, // 828: battery.cells_stat[3].t_max.value battery.cells_stat[3].t_max.value	t_float
0xF0527539, // 829: db.power_board.adc_p3V3_meas db.power_board.adc_p3V3_meas	t_float
0xF09CC4A2, // 830: grid_mon[1].u_over.time Max. voltage switch-off time level 2 [s]	t_float
0xF0A03A20, // 831: bat_mng_struct.k Forecast correction	t_float
0xF0B436DD, // 832: g_sync.p_ac_load[2] Load household phase 3 [W]	t_float
0xF0BE6429, // 833: energy.e_load_month Household month energy [Wh]	t_float
0xF1342795, // 834: power_mng.stop_discharge_current Stop discharge current [A]	t_float
0xF168B748, // 835: power_mng.soc_strategy SOC target selection	t_enum
0xF1DE6E99, // 836: battery_placeholder[0].cells_resist[3] battery_placeholder[0].cells_resist[3]	t_string
0xF1FA5BB9, // 837: grid_mon[1].f_under.time Min. frequency switch-off time level 2 [s]	t_float
0xF23D4595, // 838: battery_placeholder[0].cells_stat[1].t_min.value battery_placeholder[0].cells_stat[1].t_min.value	t_float
0xF2405AC6, // 839: nsm.p_limit Max. grid power [W]	t_float
0xF247BB16, // 840: display_struct.contrast Display contrast	t_uint8
0xF25591AA, // 841: nsm.cos_phi_p[3][0] Point 4 [P/Pn]	t_float
0xF257D342, // 842: battery.cells_stat[1].t_max.value battery.cells_stat[1].t_max.value	t_float
0xF25C339B, // 843: g_sync.u_ptp_rms[2] Phase to phase voltage 3 [V]	t_float
0xF28341E2, // 844: logger.month_eac_log_ts logger.month_eac_log_ts	t_int32
0xF2BE0C9C, // 845: p_buf_available Available buffer power [W]	t_float
0xF393B7B0, // 846: power_mng.calib_charge_power Calibration charge power [W]	t_float
0xF3FD6C4C, // 847: nsm.pf_use_p_max By over-frequency in P(f) use Pmax instead of Pmom (instant P).	t_bool
0xF3FD8CE6, // 848: battery.cells_resist[2] battery.cells_resist[2]	t_string
0xF42D4DD0, // 849: io_board.alarm_home_value	t_enum

Evaluated value	
0xF451E935, // 850: battery_placeholder[0].cells_stat[0].t_min.time battery_placeholder[0].cells_stat[0].t_min.time	t_uint32
0xF473BC5E, // 851: buf_v_control.power_reduction_max_solar_grid Max. allowed grid feed-in power [W]	t_float
0xF49F58F2, // 852: nsm.p_u[2][1] Point 3 voltage [V]	t_float
0xF52C0B50, // 853: power_mng.schedule[7] power_mng.schedule[7]	t_string
0xF54BC06D, // 854: battery.cells_stat[4].u_max.value battery.cells_stat[4].u_max.value	t_float
0xF5584F90, // 855: g_sync.p_ac_sc[1] Grid power phase 2 [W]	t_float
0xF644DCA7, // 856: bat_mng_struct.k_reserve Main reservation coefficient [0..2]	t_float
0xF677D737, // 857: battery_placeholder[0].cells_stat[6].u_max.time battery_placeholder[0].cells_stat[6].u_max.time	t_uint32
0xF68ECC1F, // 858: battery_placeholder[0].cells_stat[1].u_max.time battery_placeholder[0].cells_stat[1].u_max.time	t_uint32
0xF6A85818, // 859: nsm.f_entry Entry frequency for P(f) over-frequency mode [Hz]	t_float
0xF742C6BA, // 860: battery_placeholder[0].cells_stat[1].u_max.index battery_placeholder[0].cells_stat[1].u_max.index	t_uint8
0xF76DE445, // 861: logger.minutes_temp_log_ts logger.minutes_temp_log_ts	t_int32
0xF79D41D9, // 862: db.temp1 Heat sink temperature [°C]	t_float
0xF87A2A1E, // 863: dc_conv.last_rescan Last global rescan	t_uint32
0xF8C0D255, // 864: battery.cells[0] battery.cells[0]	t_string
0xF8DECCE6, // 865: wifi.connected_ap_ssid WiFi associated AP	t_string
0xF99E8CC8, // 866: battery.cells_stat[6] battery.cells_stat[6]	t_string
0xF9FD0D61, // 867: wifi.service_ip wifi.service_ip	t_string
0xFA3276DC, // 868: battery.cells_stat[3].t_min.time battery.cells_stat[3].t_min.time	t_uint32
0xFA7DB323, // 869: io_board.check_s0_result io_board.check_s0_result	t_uint16
0xFAA837C8, // 870: nsm.f_low_rise_grad Power rise gradient for P(f) under-frequency mode without battery [1/Pn*Hz]	t_float
0xFAE429C5, // 871: rb485.f_grid[1] Grid phase 2 frequency [Hz]	t_float
0xFB57BA65, // 872: bat_mng_struct.count bat_mng_struct.count	t_string
0xFB5DE9C5, // 873: prim_sm.minigrid_flag Minigrid support	t_bool
0xFB796780, // 874: battery.cells_stat[1] battery.cells_stat[1]	t_string
0xFBD94C1F, // 875: power_mng.amp_hours Battery energy [Ah]	t_float

0xFBF3CE97, // 876: energy.e_dc_day[1] Solar generator B day energy [Wh]	t_float
0xFBF6D834, // 877: battery.module_sn[0] Module 0 Serial Number	t_string
0xFBF8D63C, // 878: energy.e_grid_load_day_sum energy.e_grid_load_day_sum	t_float
0xFC1C614E, // 879: energy.e_ac_month_sum energy.e_ac_month_sum	t_float
0xFC1F8C65, // 880: battery_placeholder[0].cells_stat[6].t_max.time battery_placeholder[0].cells_stat[6].t_max.time	t_uint32
0xFC5AA529, // 881: bat_mng_struct.bat_calib_soc_threshold SOC threshold for battery calibration in advance	t_float
0xFC724A9E, // 882: energy.e_dc_total[0] Solar generator A total energy [Wh]	t_float
0xFCA1CBB5, // 883: battery_placeholder[0].voltage Battery voltage [V]	t_float
0xFCC39293, // 884: nsm.rpm_lock_in_power Reactive Power Mode lock-in power [P/Pn]	t_float
0xFCF4E78D, // 885: logger.day_ea_log_ts logger.day_ea_log_ts	t_int32
0xFD4F17C4, // 886: grid_mon[1].f_over.time Max. frequency switch-off time level 2 [s]	t_float
0xFD72CC0D, // 887: frt.enabled Enable FRT	t_bool
0xFDB81124, // 888: energy.e_grid_feed_day_sum energy.e_grid_feed_day_sum	t_float
0xFDBD9EE9, // 889: battery.cells_stat[3].u_max.index battery.cells_stat[3].u_max.index	t_uint8
0xFE1AA500, // 890: buf_v_control.power_reduction External power reduction based on solar plant peak power [0..1]	t_float
0xFE38B227, // 891: battery_placeholder[0].cells_stat[5] battery_placeholder[0].cells_stat[5]	t_string
0xFE44BA26, // 892: battery.cells_stat[0].u_min.index battery.cells_stat[0].u_min.index	t_uint8
0xFED51BD2, // 893: dc_conv.dc_conv_struct[1].enabled Solar generator B connected	t_bool
0xFF2A258B, // 894: wifi.server_ip wifi.server_ip	t_string
0xFF5B8A54 // 895: battery_placeholder[0].cells_stat[3] battery_placeholder[0].cells_stat[3]	t_string

Table 9

Faults

There are two classes of the faults

- Inverter faults
- Battery (BMS) faults

The faults are bit-coded in four 32-bit variables, where each bit is reserved for one fault, so there are 4 x 32 faults at all. Bit 0 of the „fault[0].flt“ variable corresponds fault index 0.

Variable	Fault indexes
fault[0].flt	0..31
fault[1].flt	32..63
fault[2].flt	64..95
fault[3].flt	96..127

Table 10

The following table presents all available faults

Fault index	Description
0	TRAP occurred
1	RTC can't be configured
2	RTC 1Hz signal timeout
3	Hardware Stop by 3.3V fault
4	Hardware Stop by PWM Logic
5	Hardware Stop by Uzk overvoltage
6	Uzk+ is over limit
7	Uzk- is over limit
8	Throttle Phase L1 overcurrent
9	Throttle Phase L2 overcurrent
10	Throttle Phase L3 overcurrent
11	Buffer capacitor voltage
12	Quartz fault
13	Grid under_voltage phase 1
14	Grid under_voltage phase 2
15	Grid under_voltage phase 3

16	Battery overcurrent
17	Relays Test failed
18	Board Over Temperature
19	Core Over Temperature
20	Sink 1 Over Temperature
21	Sink 2 Over Temperature
22	Error by I2C communication with Power Board
23	Power Board Error
24	PWM output ports defect
25	Insulation is too small or not plausible
26	I DC Component Max (1 A)
27	I DC Component Max Slow (47 mA)
28	One of the DSD channels possibly defect (too big current offset)
29	Error by RS485 communication with Relays Box IGBT L1 BH defect
30	Phase to phase over voltage
31	IGBT L1 BH defect
32	IGBT L1 BL defect
33	IGBT L2 BH defect
34	IGBT L2 BL defect
35	IGBT L3 BH defect
36	IGBT L3 BL defect
37	Long Term over voltage phase 1
38	Long Term over voltage phase 2
39	Long Term over voltage phase 3

40	Over voltage phase 1, level 1
41	Over voltage phase 1, level 2
42	Over voltage phase 2, level 1
43	Over voltage phase 2, level 2
44	Over voltage phase 3, level 1
45	Over voltage phase 3, level 2
46	Over frequency, level 1
47	Over frequency, level 2
48	Under voltage phase 1, level 1
49	Under voltage phase 1, level 2
50	Under voltage phase 2, level 1
51	Under voltage phase 2, level 2
52	Under voltage phase 3, level 1
53	Under voltage phase 3, level 2
54	Under frequency, level 1
55	Under frequency, level 2
56	CPU Exception NMI
57	CPU Exception HardFault
58	CPU Exception MemManage
59	CPU Exception BusFault
60	CPU Exception UsageFault
61	RTC Power on reset
62	RTC Oscillation stops
63	RTC Supply voltage drop

64	Jump of RCD current DC + AC > 30mA was noticed
65	Jump of RCD current DC > 60mA was noticed
66	Jump of RCD current AC > 150mA was noticed
67	RCD current > 300mA was noticed
68	incorrect 5V was noticed
69	incorrect -9V was noticed
70	incorrect 9V was noticed
71	incorrect 3V3 was noticed
72	failure of RDC calibration was noticed
73	failure of I2C was noticed
74	afi frequency generator failure
75	sink temperature too high
76	Uzk is over limit
77	Usg A is over limit
78	Usg B is over limit
79	Switching On Conditions Umin phase 1
80	Switching On Conditions Umax phase 1
81	Switching On Conditions Fmin phase 1
82	Switching On Conditions Fmax phase 1
83	Switching On Conditions Umin phase 2
84	Switching On Conditions Umax phase 2
85	Battery current sensor defect
86	Battery booster damaged
87	Switching On Conditions Umin phase 3

88	Switching On Conditions Umax phase 3
89	Voltage surge or average offset is too big on AC-terminals (phase failure detected)
90	Inverter is disconnected from the household grid
91	Difference of the measured +9V between DSP and PIC is too big
92	1.5V error
93	2.5V error
94	1.5V measurement difference
95	2.5V measurement difference
96	The battery voltage is outside of the expected range
97	Unable to start the main PIC software
98	PIC bootloader detected unexpectedly
99	Phase position error (not 120° as expected)
100	Battery overvoltage
101	Throttle current is unstable
102	Difference between internal and external measured grid voltage is too big in phase 1
103	Difference between internal and external measured grid voltage is too big in phase 2
104	Difference between internal and external measured grid voltage is too big in phase 3
105	External emergency turn off signal is active
106	Battery is empty, not more energy for standby
107	CAN communication timeout with battery
108	Timing problem
109	Battery IGBT's Heat Sink Over Temperature
110	Battery heat sink temperature too high

111	Internal Relays Box error
112	Relays Box PE off error
113	Relays Box PE on error
114	Internal battery error
115	Parameter changed
116	3 attempts of island building are failed
117	Phase to phase under voltage
118	System reset detected
119	Update detected
120	FRT over-voltage
121	FRT under-voltage
122	IGBT L1 free wheeling diode defect
123	IGBT L2 free wheeling diode defect
124	IGBT L3 free wheeling diode defect
125	1 phase mode is activated but not allowed for this device class (e.g. 10K)
126	Island detected

Table 11

The fault “Internal battery error” (index 114) is a BMS class fault and must be interpreted as follow:

Error Code

32-Bit Error Code.

Bit-Mask

0xF8000000 Subsystem

0x00000001 active

Subsystem: Master (16), UI/Power(15)

Bit-Mask

0x07FFFE00 Code
 0x000001E0 Information
 0x0000001E Severity

Subsystem: Stack/Slave (0..14)

Bit-Mask

0x07FFC000 Code
 0x00003C00 Information
 0x000003C0 Severity
 0x0000003E Cell

Severity

Severity	Description	Action
1	Info	-
2	Warning	Continue normal operation, LED is flashing.
3	Error Class 1	Wait until error is resolved. Shutdown after 2 hours. Shutdown after 5 minutes if cell voltage is low.
4	Error Class 2	Shutdown after 5 minutes. Try to resolve after restart.
5	Error Class 3	Shutdown after 5 minutes. Can not be resolved without technician.

Master

Type	Severity	Limit	Description
0x0100	3	-	BMSERRORMASTERVCCADC_TIMEOUT
0x0210	3	11.4V (5%)	BMSERRORMASTERVCC12V_MIN
0x0220	3	12.6V (5%)	BMSERRORMASTERVCC12V_MAX
0x0310	3	3.135V (5%)	BMSERRORMASTERVCC3V3_MIN
0x0320	3	3.465V (5%)	BMSERRORMASTERVCC3V3_MAX
0x1100	4	610V	BMSERRORMASTERUMAXSYSTEM
0x1200	4	6	BMSERRORMASTERINCORRECTNROFSLAVES
0x1300	4	1h	BMSERRORMASTERDEVICENOT_AVAILABLE
0x1400	4	2h	BMSERRORMASTERDEVICENOT_CONNECTED
0x2100	4		BMSERRORMASTERCONNECTLSOPENCLAMP
0x2200	4		BMSERRORMASTERCONNECTHSOPENCLAMP

0x2300	4		BMSERRORMASTERCONNECTLS_STUCK
0x2400	4		BMSERRORMASTERCONNECTHS_STUCK
0x2500	4		BMSERRORMASTERCONNECTLSSTUCKOPEN
0x2600	4		BMSERRORMASTERCONNECTHSSTUCKOPEN
0x2710	4	Ubat -5%	BMSERRORMASTERCONNECTVOLTAGE_LOW
0x2720	4	Ubat +5%	BMSERRORMASTERCONNECTVOLTAGE_HIGH
0x2800	4	1h	BMSERRORMASTERCONNECTTIMEOUT
0x3100	3	10s	BMSERRORMASTEREXTERNCAN_ERROR

UI/Power

Type	Severity	Limit	Description
0x0100	5	40A	BMSERRORUI_CURRENT
0x0110	3 >4 >5	*>25A 40A	BMSERRORUICURRENTDISCHARGE
0x0120	3 >4 >5	*>25A 40A	BMSERRORUICURRENTCHARGE
0x0400	4	1A	BMSERRORUICURRENTOFFSET
0x0500	3		BMSERRORUIRELAYSOPEN
0x0600	3		BMSERRORUI_SLAVE
0x0700	3	5%	BMSERRORUISUPPLYVOLTAGE
0x0800	4	1s	BMSERRORUI_CAN
0x1100	3	5%	BMSERRORUICURRENTDIFF
0x1200	3	5%	BMSERRORUIVOLTAGEDIFF

* Temperature dependent characteristic I(T).

Slave

Type	Severity	Limit	Description
0x0100	5	2.8V / 3.6V	BMSERRORSLAVE_CELLVOLTAGE
0x0110	3 >4 >5	2.975V 2.9V 2.8V	BMSERRORSLAVECELLVOLTAGEMIN
0x0120	4 5	3.55V 3.6V	BMSERRORSLAVECELLVOLTAGEMAX
0x0200	5	-25°C / 65°C	BMSERRORSLAVE_CELLTEMP
0x0210	5	-25°C	BMSERRORSLAVECELLTEMPMIN
0x0220	5	65°C	BMSERRORSLAVECELLTEMPMAX
0x0300	4	-20°C / 60°C	BMSERRORSLAVE_BOARDTEMP
0x0400	4		BMSERRORSLAVE_HARDWARE
0x0500	3	1x	BMSERRORSLAVE_TIMING

0x0600	3	1x	BMSERRORSLAVE_SPI
0x0700	3	10%	BMSERRORSLAVE_SUPPLYVOLTAGE
0x0800	4	1s	BMSERRORSLAVE_CAN

Faults storing in the internal flash memory

All faults can occur simultaneously. After one of the faults is gone – the appropriate record will be immediately written into internal flash memory. This record contains the following information about the fault:

- Index
- Begin time stamp
- End time stamp
- Extra information bytes

All records are sorted by the faults end time. Requesting of the faults includes the time stamp (e.g. Treq), the answer will contain all the faults whose end time is equals or smaller as Treq. For example the request time stamp is 12:00 08.11.2018, then the answer list can include the faults like this (the latest faults are first):

```
11:33:07 08.11.2018  ERR_UNSTABLE_CURRENT
9:01:15 08.11.2018  ERR_UZK_OVERVOLTAGE
3:45:50 08.11.2018  ERR_UZK_OVERVOLTAGE
23:59:00 07.11.2018  ERR_RTC
20:44:05 07.11.2018  ERR_OVERCURRENT_L1
```

Table of inverter states

The state number is in the variable prim_sm.state (see "Control Reference" column in ID's table).

State #	Description
0	Standby
1	Initialization
2	Standby
3	Efficiency (debug state for development)

	purposes)
4	Insulation check
5	Island check (decision where to go - grid connected or island)
6	Power check (decision if enough energy to start or not)
7	Symmetry (DC-link alignment)
8	Relays test
9	Grid Passive (inverter get power from grid without bridge clocking)
10	Prepare Bat Passive
11	Battery Passive (inverter disconnected from grid and get power from battery)
12	H/W check (prepare to start)
13	Feed in

Table 12

Serial Number (German)

Seriennummer-Aufbau

Stelle / Position	Inhalt / Content
1	Länder-Code
2	Geräte-Code
3	Familien-Code
4	
5	Revision/Variante
6	Baureihe-Code
7	
8	Lieferant-Code
9	Lfd. Nr.
10	
11	
12	
13	

14	
----	--

Beispiele

0066A2400000 Internationaler Inverter Power Inverter Revision A 4.0 Vorserie Lfd. Nr.0000

0181A1900000 Internationale Batterie Power Battery stack Revision A 1.9 Vorserie Lfd. Nr.0000

**Achtung: Stelle 9 und 10 werden aktuell nur für Power Battery Stack und Power Battery Master (Familien Code 81 und 80) verwendet.
Alle anderen Produkte arbeiten mit 12-stelliger Seriennummer**

Länder Code (1.Stelle) / Country Code (1.Postion)

0	international
1	Deutschland
2	Spanien
3	Italien
4	Frankreich
5	Americas

Geräte Code (2.Stelle) / Device code (2.Positon)

0	Inverter
1	Batterie
2	Zubehör

Familiencode (3. und 4.Stelle) / Family code (3 & 4 Postion)

33	sunways NT Serie (850V)
44	sunways NT Serie (900V)
55	sunways AT Serie
77	sunways PT Serie
66	Power Inverter
65	Power Storage DC
64	Power Storage AC
80	Power Battery Master
81	Power Battery Stack
90	Power Sensor
91	Power Switch Board

Revision (5.Stelle / 5.Postion) (neue Datenblattangaben)

A	1. Entwicklungsrevision
B	2. Entwicklungsrevision
C	3. Entwicklungsrevision
D	4. Entwicklungsrevision

Achtung: wird teilweise als "demo" verwendet

Baureihe Code (6. und 7.Stelle) / Model series (6 & 7 Postion)

01	NT2500
02	NT3700
03	NT4200
04	NT5000
06	NT10000
07	NT11000
08	NT12000
09	NT3000
13	AT2700
14	AT3000
15	AT3600
16	AT4500
17	AT5000
18	
19	Power Battery Stack 1.9
20	
24	Power Inverter 4.0
25	Power Inverter 5.0
26	Power Inverter 6.0
34	Power Storage AC 4.0
35	Power Storage AC 5.0
36	Power Storage AC 6.0
44	Power Storage DC 4.0
45	Power Storage DC 5.0
46	Power Storage DC 6.0

48	Power Storage DC 8.0
49	Power Storage DC 9.0
40	Power Storage DC 10.0
50	Power Sensor 50
51	Power Sensor 100
71	Power Battery Master 71

Lieferant Code (8.Stelle) / Supplier code (8.Postion)

0	Vorserie
1	
2	
3	Lieferant RPG
4	Lieferant RCT Jiangzhu
5	
6	
7	
8	
9	

Laufende Nummer (9. bis 14 (12).Stelle)

000000.....999999	Laufende Nr.
-------------------	--------------

eigene Zählung je Baureihecode (d.h. pro Leistungsklasse und Ländervariante)