

Standard Network Variable Types

Network Variables Standard 00:00:00:00:00:00:00-0. This file is the standard type file, containing Standard Network Variable Types (SNVTs), Standard Configuration Property Types (SCPTs), and the enumeration types that support them. This file was created and is maintained by Echelon Corporation, USA. Contact us at +1-408-938-5200, at <http://www.echelon.com>, or at 550 Meridian Ave, San Jose CA 95126, USA.

Standard Network Variable Types (SNVTs) facilitate interoperability by providing a well-defined interface for communication between devices made by different manufacturers. A device may be installed in a network and logically connected to other devices via network variables as long as the data types match.

This document provides information on all available SNVTs. A SNVT index is defined for each network variable that is used when defining self-identification for network variables. The SNVT names are provided for use with network and development tools.

By Name		By Index
SNVT_abs_humid	1	SNVT_amp
SNVT_address	2	SNVT_amp_mil
SNVT_alarm	3	SNVT_angle
SNVT_alarm_2	4	SNVT_angle_vel
SNVT_amp	5	SNVT_btu_kilo
SNVT_amp_ac	6	SNVT_btu_mega
SNVT_amp_f	7	SNVT_char_ascii
SNVT_amp_mil	8	SNVT_count
SNVT_angle	9	SNVT_count_inc
SNVT_angle_deg	10	SNVT_date_cal
SNVT_angle_f	11	SNVT_date_day
SNVT_angle_vel	12	SNVT_date_time
SNVT_angle_vel_f	13	SNVT_elec_kwh
SNVT_area	14	SNVT_elec_whr
SNVT_btu_f	15	SNVT_flow
SNVT_btu_kilo	16	SNVT_flow_mil
SNVT_btu_mega	17	SNVT_length
SNVT_char_ascii	18	SNVT_length_kilo
SNVT_chlr_status	19	SNVT_length_micr
SNVT_clothes_w_a	20	SNVT_length_mil
SNVT_clothes_w_c	21	SNVT_lev_cont
SNVT_clothes_w_m	22	SNVT_lev_disc
SNVT_clothes_w_s	23	SNVT_mass
SNVT_color	24	SNVT_mass_kilo
SNVT_color_2	25	SNVT_mass_mega
SNVT_config_src	26	SNVT_mass_mil
SNVT_count	27	SNVT_power
SNVT_count_32	28	SNVT_power_kilo
SNVT_count_f	29	SNVT_ppm
SNVT_count_inc	30	SNVT_press
SNVT_count_inc_f	31	SNVT_res
SNVT_ctrl_req	32	SNVT_res_kilo
SNVT_ctrl_resp	33	SNVT_sound_db
SNVT_currency	34	SNVT_speed
SNVT_date_cal	35	SNVT_speed_mil
SNVT_date_day	36	SNVT_str_asc

SNVT_date_event	37	SNVT_str_int
SNVT_date_time	38	SNVT_telcom
SNVT_defr_mode	39	SNVT_temp
SNVT_defr_state	40	SNVT_time_passed
SNVT_defr_term	41	SNVT_vol
SNVT_density	42	SNVT_vol_kilo
SNVT_density_f	43	SNVT_vol_mil
SNVT_dev_c_mode	44	SNVT_volt
SNVT_dev_fault	45	SNVT_volt_dbmv
SNVT_dev_maint	46	SNVT_volt_kilo
SNVT_dev_status	47	SNVT_volt_mil
SNVT_earth_pos	48	SNVT_amp_f
SNVT_elapsed_tm	49	SNVT_angle_f
SNVT_elec_kwh	50	SNVT_angle_vel_f
SNVT_elec_kwh_l	51	SNVT_count_f
SNVT_elec_whr	52	SNVT_count_inc_f
SNVT_elec_whr_f	53	SNVT_flow_f
SNVT_ent_opmode	54	SNVT_length_f
SNVT_ent_state	55	SNVT_lev_cont_f
SNVT_ent_status	56	SNVT_mass_f
SNVT_enthalpy	57	SNVT_power_f
SNVT_evap_state	58	SNVT_ppm_f
SNVT_ex_control	59	SNVT_press_f
SNVT_file_pos	60	SNVT_res_f
SNVT_file_req	61	SNVT_sound_db_f
SNVT_file_status	62	SNVT_speed_f
SNVT_fire_indcte	63	SNVT_temp_f
SNVT_fire_init	64	SNVT_time_f
SNVT_fire_test	65	SNVT_vol_f
SNVT_flow	66	SNVT_volt_f
SNVT_flow_dir	67	SNVT_btu_f
SNVT_flow_f	68	SNVT_elec_whr_f
SNVT_flow_mil	69	SNVT_config_src
SNVT_flow_p	70	SNVT_color
SNVT_freq_f	71	SNVT_grammage
SNVT_freq_hz	72	SNVT_grammage_f
SNVT_freq_kilohz	73	SNVT_file_req
SNVT_freq_milhz	74	SNVT_file_status
SNVT_gfci_status	75	SNVT_freq_f
SNVT_grammage	76	SNVT_freq_hz
SNVT_grammage_f	77	SNVT_freq_kilohz
SNVT_hvac_emerg	78	SNVT_freq_milhz
SNVT_hvac_mode	79	SNVT_lux
SNVT_hvac_overid	80	SNVT_ISO_7811
SNVT_hvac_satsts	81	SNVT_lev_percent
SNVT_hvac_status	82	SNVT_multiplier
SNVT_hvac_type	83	SNVT_state
SNVT_ISO_7811	84	SNVT_time_stamp
SNVT_length	85	SNVT_zerospans
SNVT_length_f	86	SNVT_magcard
SNVT_length_kilo	87	SNVT_elapsed_tm
SNVT_length_micr	88	SNVT_alarm
SNVT_length_mil	89	SNVT_currency
SNVT_lev_cont	90	SNVT_file_pos
SNVT_lev_cont_f	91	SNVT_muldiv
SNVT_lev_disc	92	SNVT_obj_request
SNVT_lev_percent	93	SNVT_obj_status

SNVT_log_fx_request	94	SNVT_preset
SNVT_log_fx_status	95	SNVT_switch
SNVT_log_request	96	SNVT_trans_table
SNVT_log_status	97	SNVT_override
SNVT_lux	98	SNVT_pwr_fact
SNVT_magcard	99	SNVT_pwr_fact_f
SNVT_mass	100	SNVT_density
SNVT_mass_f	101	SNVT_density_f
SNVT_mass_kilo	102	SNVT_rpm
SNVT_mass_mega	103	SNVT_hvac_emerg
SNVT_mass_mil	104	SNVT_angle_deg
SNVT_motor_state	105	SNVT_temp_p
SNVT_muldiv	106	SNVT_temp_setpt
SNVT_multiplier	107	SNVT_time_sec
SNVT_multiplier_s	108	SNVT_hvac_mode
SNVT_nv_type	109	SNVT_occupancy
SNVT_obj_request	110	SNVT_area
SNVT_obj_status	111	SNVT_hvac_overid
SNVT_occupancy	112	SNVT_hvac_status
SNVT_override	113	SNVT_press_p
SNVT_ph	114	SNVT_address
SNVT_ph_f	115	SNVT_scene
SNVT_pos_ctrl	116	SNVT_scene_cfg
SNVT_power	117	SNVT_setting
SNVT_power_f	118	SNVT_evap_state
SNVT_power_kilo	119	SNVT_therm_mode
SNVT_ppm	120	SNVT_defr_mode
SNVT_ppm_f	121	SNVT_defr_term
SNVT_preset	122	SNVT_defr_state
SNVT_press	123	SNVT_time_min
SNVT_press_f	124	SNVT_time_hour
SNVT_press_p	125	SNVT_ph
SNVT_privacyzone	126	SNVT_ph_f
SNVT_ptz	127	SNVT_chlr_status
SNVT_pump_sensor	128	SNVT_tod_event
SNVT_pumpset_mn	129	SNVT_smo_obscur
SNVT_pumpset_sn	130	SNVT_fire_test
SNVT_pwr_fact	131	SNVT_temp_ror
SNVT_pwr_fact_f	132	SNVT_fire_init
SNVT_rac_ctrl	133	SNVT_fire_indcte
SNVT_rac_req	134	SNVT_time_zone
SNVT_reg_val	135	SNVT_earth_pos
SNVT_reg_val_ts	136	SNVT_reg_val
SNVT_res	137	SNVT_reg_val_ts
SNVT_res_f	138	SNVT_volt_ac
SNVT_res_kilo	139	SNVT_amp_ac
SNVT_rpm	143	SNVT_turbidity
SNVT_sblnd_state	144	SNVT_turbidity_f
SNVT_scene	145	SNVT_hvac_type
SNVT_scene_cfg	146	SNVT_elec_kwh_l
SNVT_sched_val	147	SNVT_temp_diff_p
SNVT_sec_state	148	SNVT_ctrl_req
SNVT_sec_status	149	SNVT_ctrl_resp
SNVT_setting	150	SNVT_ptz
SNVT_smo_obscur	151	SNVT_privacyzone
SNVT_sound_db	152	SNVT_pos_ctrl
SNVT_sound_db_f	153	SNVT_enthalpy

SNVT_speed	154	SNVT_gfci_status
SNVT_speed_f	155	SNVT_motor_state
SNVT_speed_mil	156	SNVT_pumpset_mn
SNVT_state	157	SNVT_ex_control
SNVT_state_64	158	SNVT_pumpset_sn
SNVT_str_asc	159	SNVT_pump_sensor
SNVT_str_int	160	SNVT_abs_humid
SNVT_switch	161	SNVT_flow_p
SNVT_switch_2	162	SNVT_dev_c_mode
SNVT_telcom	163	SNVT_valve_mode
SNVT_temp	164	SNVT_alarm_2
SNVT_temp_diff_p	165	SNVT_state_64
SNVT_temp_f	166	SNVT_nv_type
SNVT_temp_p	168	SNVT_ent_opmode
SNVT_temp_ror	169	SNVT_ent_state
SNVT_temp_setpt	170	SNVT_ent_status
SNVT_therm_mode	171	SNVT_flow_dir
SNVT_time_f	172	SNVT_hvac_satsts
SNVT_time_hour	173	SNVT_dev_status
SNVT_time_min	174	SNVT_dev_fault
SNVT_time_passed	175	SNVT_dev_maint
SNVT_time_sec	176	SNVT_date_event
SNVT_time_stamp	177	SNVT_sched_val
SNVT_time_stamp_p	178	SNVT_sec_state
SNVT_time_zone	179	SNVT_sec_status
SNVT_tod_event	180	SNVT_sblnd_state
SNVT_trans_table	181	SNVT_rac_ctrl
SNVT_turbidity	182	SNVT_rac_req
SNVT_turbidity_f	183	SNVT_count_32
SNVT_valve_mode	184	SNVT_clothes_w_c
SNVT_vol	185	SNVT_clothes_w_m
SNVT_vol_f	186	SNVT_clothes_w_s
SNVT_vol_kilo	187	SNVT_clothes_w_a
SNVT_vol_mil	188	SNVT_multiplier_s
SNVT_volt	189	SNVT_switch_2
SNVT_volt_ac	190	SNVT_color_2
SNVT_volt_dbmv	191	SNVT_log_status
SNVT_volt_f	192	SNVT_time_stamp_p
SNVT_volt_kilo	193	SNVT_log_fx_request
SNVT_volt_mil	194	SNVT_log_fx_status
SNVT_zerospan	195	SNVT_log_request

Resource File Set Version Information

Data Version: 13.02

Format Version: 5.00

Timestamp: 01 January 2008

SNVT_ISO_7811

Overview:

ISO 7811 (38 hexadecimal digits) . This SNVT is obsolete. Use SNVT_magcard instead.

Obsolete. Use SNVT_magcard.

Details:

Standard: *yes*

Resource Set: *Standard 00:00:00:00:00:00:00:00-0*

Index: *80*

Obsolete: *yes*

Size: *19*

Programmatic Name: *SNVT_ISO_7811*

Neuron C Type:

Structure	
digit1	Digit 1 (hexadecimal digit) .
	bitfield Signed: <i>no</i> Width: <i>4</i> Offset: <i>0</i> Minimum: <i>0</i> Maximum: <i>15</i> Scaling <i>1, 0, 0</i> (A,B,C): Scaled <i>$1 * 10^0 * (Raw + 0)$</i> value: Resolution: <i>1</i>
digit2	Digit 2 (hexadecimal digit) .
	bitfield Signed: <i>no</i> Width: <i>4</i> Offset: <i>4</i> Minimum: <i>0</i> Maximum: <i>15</i> Scaling <i>1, 0, 0</i> (A,B,C): Scaled <i>$1 * 10^0 * (Raw + 0)$</i> value: Resolution: <i>1</i>
digit3	Digit 3 (hexadecimal digit) .

	bitfield Signed: <i>no</i> Width: <i>4</i> Offset: <i>0</i> Minimum: <i>0</i> Maximum: <i>15</i> Scaling <i>1, 0, 0</i> (A,B,C): Scaled <i>1 * 10⁰ *(Raw+0)</i> value: Resolution: <i>1</i>
digit4	Digit 4 (hexadecimal digit) .
	bitfield Signed: <i>no</i> Width: <i>4</i> Offset: <i>4</i> Minimum: <i>0</i> Maximum: <i>15</i> Scaling <i>1, 0, 0</i> (A,B,C): Scaled <i>1 * 10⁰ *(Raw+0)</i> value: Resolution: <i>1</i>
digit5	Digit 5 (hexadecimal digit) .
	bitfield Signed: <i>no</i> Width: <i>4</i> Offset: <i>0</i> Minimum: <i>0</i> Maximum: <i>15</i> Scaling <i>1, 0, 0</i> (A,B,C): Scaled <i>1 * 10⁰ *(Raw+0)</i> value: Resolution: <i>1</i>
digit6	Digit 6 (hexadecimal digit) .
	bitfield Signed: <i>no</i> Width: <i>4</i> Offset: <i>4</i> Minimum: <i>0</i> Maximum: <i>15</i> Scaling <i>1, 0, 0</i> (A,B,C): Scaled <i>1 * 10⁰ *(Raw+0)</i> value:

	Resolution: <i>1</i>
digit7	Digit 7 (hexadecimal digit) .
	bitfield Signed: <i>no</i> Width: <i>4</i> Offset: <i>0</i> Minimum: <i>0</i> Maximum: <i>15</i> Scaling <i>1, 0, 0</i> (A,B,C): Scaled $1 * 10^0 * (Raw + 0)$ value: Resolution: <i>1</i>
digit8	Digit 8 (hexadecimal digit) .
	bitfield Signed: <i>no</i> Width: <i>4</i> Offset: <i>4</i> Minimum: <i>0</i> Maximum: <i>15</i> Scaling <i>1, 0, 0</i> (A,B,C): Scaled $1 * 10^0 * (Raw + 0)$ value: Resolution: <i>1</i>
digit9	Digit 9 (hexadecimal digit) .
	bitfield Signed: <i>no</i> Width: <i>4</i> Offset: <i>0</i> Minimum: <i>0</i> Maximum: <i>15</i> Scaling <i>1, 0, 0</i> (A,B,C): Scaled $1 * 10^0 * (Raw + 0)$ value: Resolution: <i>1</i>
digit10	Digit 10 (hexadecimal digit) .
	bitfield Signed: <i>no</i> Width: <i>4</i> Offset: <i>4</i> Minimum: <i>0</i> Maximum: <i>15</i> Scaling <i>1, 0, 0</i>

	(A,B,C): Scaled $1 * 10^0 * (Raw+0)$ value: Resolution: 1
digit11	Digit 11 (hexadecimal digit) .
	bitfield Signed: <i>no</i> Width: 4 Offset: 0 Minimum: 0 Maximum: 15 Scaling $1, 0, 0$ (A,B,C): Scaled $1 * 10^0 * (Raw+0)$ value: Resolution: 1
digit12	Digit 12 (hexadecimal digit) .
	bitfield Signed: <i>no</i> Width: 4 Offset: 4 Minimum: 0 Maximum: 15 Scaling $1, 0, 0$ (A,B,C): Scaled $1 * 10^0 * (Raw+0)$ value: Resolution: 1
digit13	Digit 13 (hexadecimal digit) .
	bitfield Signed: <i>no</i> Width: 4 Offset: 0 Minimum: 0 Maximum: 15 Scaling $1, 0, 0$ (A,B,C): Scaled $1 * 10^0 * (Raw+0)$ value: Resolution: 1
digit14	Digit 14 (hexadecimal digit) .
	bitfield Signed: <i>no</i> Width: 4 Offset: 4

	Minimum: <i>0</i> Maximum: <i>15</i> Scaling <i>1, 0, 0</i> (A,B,C): Scaled <i>1 * 10⁰ *(Raw+0)</i> value: Resolution: <i>1</i>
digit15	Digit 15 (hexadecimal digit) .
	bitfield Signed: <i>no</i> Width: <i>4</i> Offset: <i>0</i> Minimum: <i>0</i> Maximum: <i>15</i> Scaling <i>1, 0, 0</i> (A,B,C): Scaled <i>1 * 10⁰ *(Raw+0)</i> value: Resolution: <i>1</i>
digit16	Digit 16 (hexadecimal digit) .
	bitfield Signed: <i>no</i> Width: <i>4</i> Offset: <i>4</i> Minimum: <i>0</i> Maximum: <i>15</i> Scaling <i>1, 0, 0</i> (A,B,C): Scaled <i>1 * 10⁰ *(Raw+0)</i> value: Resolution: <i>1</i>
digit17	Digit 17 (hexadecimal digit) .
	bitfield Signed: <i>no</i> Width: <i>4</i> Offset: <i>0</i> Minimum: <i>0</i> Maximum: <i>15</i> Scaling <i>1, 0, 0</i> (A,B,C): Scaled <i>1 * 10⁰ *(Raw+0)</i> value: Resolution: <i>1</i>
digit18	Digit 18 (hexadecimal digit) .
	bitfield

	Signed: <i>no</i> Width: <i>4</i> Offset: <i>4</i> Minimum: <i>0</i> Maximum: <i>15</i> Scaling <i>1, 0, 0</i> (A,B,C): Scaled <i>1 * 10⁰ * (Raw+0)</i> value: Resolution: <i>1</i>
digit19	Digit 19 (hexadecimal digit) .
	bitfield Signed: <i>no</i> Width: <i>4</i> Offset: <i>0</i> Minimum: <i>0</i> Maximum: <i>15</i> Scaling <i>1, 0, 0</i> (A,B,C): Scaled <i>1 * 10⁰ * (Raw+0)</i> value: Resolution: <i>1</i>
digit20	Digit 20 (hexadecimal digit) .
	bitfield Signed: <i>no</i> Width: <i>4</i> Offset: <i>4</i> Minimum: <i>0</i> Maximum: <i>15</i> Scaling <i>1, 0, 0</i> (A,B,C): Scaled <i>1 * 10⁰ * (Raw+0)</i> value: Resolution: <i>1</i>
digit21	Digit 21 (hexadecimal digit) .
	bitfield Signed: <i>no</i> Width: <i>4</i> Offset: <i>0</i> Minimum: <i>0</i> Maximum: <i>15</i> Scaling <i>1, 0, 0</i> (A,B,C): Scaled <i>1 * 10⁰ * (Raw+0)</i> value: Resolution: <i>1</i>

digit22	Digit 22 (hexadecimal digit) .
	bitfield Signed: <i>no</i> Width: <i>4</i> Offset: <i>4</i> Minimum: <i>0</i> Maximum: <i>15</i> Scaling <i>1, 0, 0</i> (A,B,C): Scaled $1 * 10^0 * (Raw + 0)$ value: Resolution: <i>1</i>
digit23	Digit 23 (hexadecimal digit) .
	bitfield Signed: <i>no</i> Width: <i>4</i> Offset: <i>0</i> Minimum: <i>0</i> Maximum: <i>15</i> Scaling <i>1, 0, 0</i> (A,B,C): Scaled $1 * 10^0 * (Raw + 0)$ value: Resolution: <i>1</i>
digit24	Digit 24 (hexadecimal digit) .
	bitfield Signed: <i>no</i> Width: <i>4</i> Offset: <i>4</i> Minimum: <i>0</i> Maximum: <i>15</i> Scaling <i>1, 0, 0</i> (A,B,C): Scaled $1 * 10^0 * (Raw + 0)$ value: Resolution: <i>1</i>
digit25	Digit 25 (hexadecimal digit) .
	bitfield Signed: <i>no</i> Width: <i>4</i> Offset: <i>0</i> Minimum: <i>0</i> Maximum: <i>15</i> Scaling <i>1, 0, 0</i> (A,B,C): Scaled

	value: $1 * 10^0 * (Raw + 0)$ Resolution: 1
digit26	Digit 26 (hexadecimal digit) .
	bitfield Signed: <i>no</i> Width: 4 Offset: 4 Minimum: 0 Maximum: 15 Scaling $1, 0, 0$ (A,B,C): Scaled $1 * 10^0 * (Raw + 0)$ value: Resolution: 1
digit27	Digit 27 (hexadecimal digit) .
	bitfield Signed: <i>no</i> Width: 4 Offset: 0 Minimum: 0 Maximum: 15 Scaling $1, 0, 0$ (A,B,C): Scaled $1 * 10^0 * (Raw + 0)$ value: Resolution: 1
digit28	Digit 28 (hexadecimal digit) .
	bitfield Signed: <i>no</i> Width: 4 Offset: 4 Minimum: 0 Maximum: 15 Scaling $1, 0, 0$ (A,B,C): Scaled $1 * 10^0 * (Raw + 0)$ value: Resolution: 1
digit29	Digit 29 (hexadecimal digit) .
	bitfield Signed: <i>no</i> Width: 4 Offset: 0 Minimum: 0 Maximum: 15

	Scaling <i>1, 0, 0</i> (A,B,C): Scaled $1 * 10^0 * (Raw + 0)$ value: Resolution: <i>1</i>
digit30	Digit 30 (hexadecimal digit) .
	bitfield Signed: <i>no</i> Width: <i>4</i> Offset: <i>4</i> Minimum: <i>0</i> Maximum: <i>15</i> Scaling <i>1, 0, 0</i> (A,B,C): Scaled $1 * 10^0 * (Raw + 0)$ value: Resolution: <i>1</i>
digit31	Digit 31 (hexadecimal digit) .
	bitfield Signed: <i>no</i> Width: <i>4</i> Offset: <i>0</i> Minimum: <i>0</i> Maximum: <i>15</i> Scaling <i>1, 0, 0</i> (A,B,C): Scaled $1 * 10^0 * (Raw + 0)$ value: Resolution: <i>1</i>
digit32	Digit 32 (hexadecimal digit) .
	bitfield Signed: <i>no</i> Width: <i>4</i> Offset: <i>4</i> Minimum: <i>0</i> Maximum: <i>15</i> Scaling <i>1, 0, 0</i> (A,B,C): Scaled $1 * 10^0 * (Raw + 0)$ value: Resolution: <i>1</i>
digit33	Digit 33 (hexadecimal digit) .
	bitfield Signed: <i>no</i> Width: <i>4</i>

	Offset: 0 Minimum: 0 Maximum: 15 Scaling 1, 0, 0 (A,B,C): Scaled value: $1 * 10^0 * (Raw + 0)$ Resolution: 1
digit34	Digit 34 (hexadecimal digit) .
	bitfield Signed: no Width: 4 Offset: 4 Minimum: 0 Maximum: 15 Scaling 1, 0, 0 (A,B,C): Scaled value: $1 * 10^0 * (Raw + 0)$ Resolution: 1
digit35	Digit 35 (hexadecimal digit) .
	bitfield Signed: no Width: 4 Offset: 0 Minimum: 0 Maximum: 15 Scaling 1, 0, 0 (A,B,C): Scaled value: $1 * 10^0 * (Raw + 0)$ Resolution: 1
digit36	Digit 36 (hexadecimal digit) .
	bitfield Signed: no Width: 4 Offset: 4 Minimum: 0 Maximum: 15 Scaling 1, 0, 0 (A,B,C): Scaled value: $1 * 10^0 * (Raw + 0)$ Resolution: 1
digit37	Digit 37 (hexadecimal digit) .

measures relative humidity, it may calculate Absolute Humidity from this using the following:
 $\text{Absolute_Humidity [g/kg]} = \text{maximal_absolute_Humidity [g/kg]} \times \text{relative_Humidity}$
 Relative Humidity is specified by SNVT_lev_percent.

See also SNVT_enthalpy.

Details:

Standard:	<i>yes</i>
Resource Set:	<i>Standard 00:00:00:00:00:00:00:00-0</i>
Index:	<i>160</i>
Obsolete:	<i>no</i>
Size:	<i>2</i>
Programmatic Name:	<i>SNVT_abs_humid</i>
Neuron C Type:	unsigned long
	Minimum: <i>0</i>
	Maximum: <i>65535</i>
	Invalid: <i>65535</i>
	Scaling (A,B,C): <i>1, -2, 0</i>
	Scaled value: <i>$1 * 10^{-2} * (Raw + 0)$</i>
	Resolution: <i>0.01</i>
Formats:	<i>SNVT_abs_humid#SI: text("%f")</i>
	<i>SNVT_abs_humid#US: text("%f", *6.99983+0(0:1461))</i>

SNVT_address

Overview:

Neuron address (16-bit address value) .

Details:

Standard:	<i>yes</i>
Resource Set:	<i>Standard 00:00:00:00:00:00:00:00-0</i>
Index:	<i>114</i>
Obsolete:	<i>no</i>
Size:	<i>2</i>
Programmatic Name:	<i>SNVT_address</i>
Neuron C Type:	unsigned long
	Minimum: <i>16384</i>
	Maximum: <i>64767</i>
	Invalid: <i>64768</i>

Scaling (A,B,C): $1, 0, 0$ Scaled value: $1 * 10^0 * (Raw + 0)$ Resolution: 1 Formats: *SNVT_address: text("%x")*Used by: *SFPTnodeObject*

SNVT_alarm

Overview:

Alarm status.

This is used as an output network variable within a Node Object to report alarm status. **For new designs, SNVT_alarm_2 should be used instead of SNVT_alarm.**

See SNVT_alarm_2.

Details:

Standard: *yes*Resource Set: *Standard 00:00:00:00:00:00:00:00-0*Index: *88*Obsolete: *no*Size: *29*Programmatic Name: *SNVT_alarm*

Neuron C Type:

Structure	
location	Location (array of 6 bytes) . Location code for the node
	unsigned short [6] Minimum: 0 Maximum: 255 Scaling (A,B,C): $1, 0, 0$ Scaled value: $1 * 10^0 * (Raw + 0)$ Resolution: 1
object_id	Object ID (object index) . ID of object within node
	unsigned long Minimum: 0 Maximum: 65535 Scaling (A,B,C): $1, 0, 0$ Scaled

	value: $1 * 10^0 * (Raw + 0)$ Resolution: 1
alarm_type	Alarm type (alarm type names) .
	<i>alarm_type_t</i>
priority_level	Priority level (priority level names) .
	<i>priority_level_t</i>
index_to_SNVT	Index of NV (index of NV causing alarm) .
	unsigned long Minimum: 0 Maximum: 65535 Scaling 1, 0, 0 (A,B,C): Scaled $1 * 10^0 * (Raw + 0)$ value: Resolution: 1
value	Value (array of 4 bytes) . The type of this field is dependent on the NV causing the alarm condition.
	unsigned short [4] Minimum: 0 Maximum: 255 Scaling 1, 0, 0 (A,B,C): Scaled $1 * 10^0 * (Raw + 0)$ value: Resolution: 1
year	Year (years) . Zero (0) means year not specified.
	unsigned long Minimum: 0 Maximum: 3000 Scaling 1, 0, 0 (A,B,C): Scaled $1 * 10^0 * (Raw + 0)$ value: Resolution: 1
month	Month (months) . Zero (0) means month not specified.
	unsigned short Minimum: 0 Maximum: 12 Scaling 1, 0, 0 (A,B,C): Scaled $1 * 10^0 * (Raw + 0)$ value: Resolution: 1
day	Day (days) . Zero (0) means day not specified.

	unsigned short Minimum: 0 Maximum: 31 Scaling 1, 0, 0 (A,B,C): Scaled value: $1 * 10^0 * (Raw + 0)$ Resolution: 1
hour	Hour (hours) . This field uses a 24-hour value.
	unsigned short Minimum: 0 Maximum: 23 Scaling 1, 0, 0 (A,B,C): Scaled value: $1 * 10^0 * (Raw + 0)$ Resolution: 1
minute	Minute (minutes) .
	unsigned short Minimum: 0 Maximum: 59 Scaling 1, 0, 0 (A,B,C): Scaled value: $1 * 10^0 * (Raw + 0)$ Resolution: 1
second	Second (seconds) .
	unsigned short Minimum: 0 Maximum: 59 Scaling 1, 0, 0 (A,B,C): Scaled value: $1 * 10^0 * (Raw + 0)$ Resolution: 1
millisecond	Millisecond (milliseconds) .
	unsigned long Minimum: 0 Maximum: 999 Scaling 1, 0, 0 (A,B,C): Scaled value: $1 * 10^0 * (Raw + 0)$ Resolution: 1
alarm_limit	Alarm limit (array of 4 bytes) . The type of this field is dependent on the NV causing the alarm condition.

	unsigned short [4]
	Minimum: 0
	Maximum: 255
	Scaling 1, 0, 0
	(A,B,C):
	Scaled value: $1 * 10^0 * (Raw + 0)$
	Resolution: 1

Formats:

SNVT_alarm: text("%x %x %x %x %x %x %d %m %m %d <%x %x %x %x> %d/%d/%d/%d:%d:%d:%d <%x %x %x %x>", location[0], location[1], location[2], location[3], location[4], location[5], object_id, alarm_type, priority_level, index_to_SNVT, value[0], value[1], value[2], value[3], year, month, day, hour, minute, second, millisecond, alarm_limit[0], alarm_limit[1], alarm_limit[2], alarm_limit[3])

SNVT_alarm#LO: text("%x %x %x %x %x %x %d %m %m %d <%x %x %x %x> ",location[0], location[1], location[2], location[3],location[4], location[5], object_id, alarm_type,priority_level, index_to_SNVT, value[0], value[1],value[2], value[3]),date(year, month, day),(" "),time(hour, minute, second, millisecond),(" <%x %x %x %x>", alarm_limit[0], alarm_limit[1], alarm_limit[2], alarm_limit[3]))

Used by:

SFPTnodeObject

SNVT_alarm_2

Overview:

Alarm status 2. Used to report alarm status for a functional block or device. Replaces SNVT_alarm

This is used as an output network variable within a Node Object functional block to report alarm status. **This type should be used instead of SNVT_alarm for all new designs.**

Following are the summary requirements for this type:

- Alarm values shall specify the alarm type, priority, time, and description.
- Alarm descriptions shall support sufficient text description to be useful for a human interface like a summary log, historical log, Web page, email message, or SMS message.
- Alarm descriptions should allow manufacturers to embed their own error codes.
- Alarm descriptions may include references to strings defined in LonMark resource files.
- Alarm time stamps shall support a resolution of up to 1 millisecond.
- A receiving device shall be able to detect a missed update.
- A receiving device shall be able to determine when all alarm updates from an alarm producer have been sent.
- Multiple alarms may be reported for a single functional block. Receiving devices or applications must be able to request an update of all alarms for a functional block, determine when an update sequence has started, and determine when an update sequence has ended. This requirement will require support in the Node Object functional block.
- Alarm processing should be simple enough to be implemented in a Neuron® Chip -hosted device or other embedded controller.

- Alarm types should be compatible between SNVT_alarm and SNVT_alarm_2 to simplify handling of both types of alarms.

Details:

Standard: *yes*
 Resource Set: *Standard 00:00:00:00:00:00:00:00-0*
 Index: *164*
 Obsolete: *no*
 Size: *31*
 Programmatic Name: *SNVT_alarm_2*
 Neuron C Type:

Structure	
alarm_type	Alarm type (alarm type names) . Alarm condition reported by this update
	<i>alarm_type_t</i>
priority_level	Priority level (priority level names) . Priority level of the alarm reported by this update
	<i>priority_level_t</i>
alarm_time	Alarm time (seconds) . Alarm time in seconds since 2000-01-01T00:00:00Z (the 0 hour of 1 January 2000, Coordinated Universal Time)
	unsigned quad Minimum: <i>0</i> Maximum: <i>4294967295</i> Invalid: <i>4294967295</i> Scaling <i>1, 0, 0</i> (A,B,C): Scaled value: <i>$1 * 10^0 * (Raw + 0)$</i> Resolution: <i>1</i>
milliseconds	Milliseconds (milliseconds) . Alarm time in milliseconds since the second specified by the alarm_time field
	signed long Minimum: <i>-1</i> Maximum: <i>999</i> Invalid: <i>-1</i> Scaling <i>1, 0, 0</i> (A,B,C): Scaled value: <i>$1 * 10^0 * (Raw + 0)$</i> Resolution: <i>1</i>
sequence_number	Sequence number (count) . Sequence number for this update. Incremented by one for each update from an alarm source. Wraps to zero after reaching 255. An alarm receiver can use the sequence number to detect

	missed alarm messages.
	unsigned short Minimum: 0 Maximum: 255 Scaling 1, 0, 0 (A,B,C): Scaled value: $1 * 10^0 * (Raw + 0)$ Resolution: 1
	<p>Description (array of 22 characters) . Alarm description with NUL terminator. The terminator is not required if the description requires 22 characters. May include a reference to a language string, delimited by a 0x80 value.</p> <p>Alarm description with NUL terminator. The NUL terminator is not required if the description requires all 22 characters of the array. The alarm description is manufacturer-defined, but may include the alarm location, an identification of the object in alarm, the current value of the data point in alarm, and the limit that caused the alarm. The alarm description may include a reference to a string defined in a language file. The 0x80 value is reserved to delimit a language string reference. See <i>Language File String Reference</i> below.</p> <h2>Language File String Reference</h2> <p>The description field of a SNVT_alarm_2 value can reference a language string using the reserved 0x80 value (represented as an “\x80” ASCII string). This string must be contained in one or more LONMARK language files (one for each supported language), and may optionally also be contained in an XML file derived from the language file by a development tool.</p> <p>The syntax for a string reference is as follows: <code>\x80[programIDTemplate-][scopeSpecifier:] languageStringIndex;</code></p> <p>The components of a string reference are the following:</p> <p>A byte containing the value 0x80, represented by the “\x80” string.</p> <ul style="list-style-type: none"> • <i>programIDTemplate</i> is a hex byte string of up to 16 characters specifying the program ID template of the resource file set containing the language string. Unused characters within the program ID template may be omitted, shortening the string at all scope levels other

<div>description</div>	<div> <p>than 6. For example, the program ID template for a scope 3 resource file may consist of six characters representing the hex values of the format and manufacturer ID fields of the program ID template. If not included, the program ID template equals the program ID template of the device reporting the alarm.</p> <ul style="list-style-type: none"> • A hyphen (“-”) follows the program ID template. The hyphen is not included if the program ID template is not included, otherwise it is mandatory. • <i>scopeSpecifier</i> may be a “3”, “4”, “5”, or “6” to specify a scope 3, 4, 5, or 6 resource. If not included, the scope is 0. • A colon (“:”) following the scope specifier. The colon is not included if the scope specifier is not included, otherwise it is mandatory. • <i>languageStringIndex</i> is the index of the language string within the language file. This index ranges from 1 to 16’777’216. • A semicolon (“;”) terminates the string reference. <p>EXAMPLES</p> <p>The following string reference specifies language string index 522 within the standard resource file set.</p> <p style="text-align: center;">"\x80522;"</p> <p>The following string reference specifies language string index 100 within a user resource file set at scope 3, with a program ID template matching the program ID of the device that originated the alarm.</p> <p style="text-align: center;">"\x803:100;"</p> <p>The following string reference specifies language string index 200 within a user resource file set with a program ID template of 80:00:9F:00:00:00:00 at scope 3.</p> <p style="text-align: center;">"\x8080009F-3:200;"</p> </div>
	<div> <p>unsigned char [22]</p> <p>Minimum: 32</p> <p>Maximum: 126</p> <p>Scaling 1, 0, 0</p> <p>(A,B,C):</p> <p>Scaled 1 *10⁰ *(Raw+0)</p> </div>

	value:
	Resolution: 1

Formats: *SNVT_alarm_2: text("%m %m %d %d %d %s", alarm_type, priority_level, alarm_time, milliseconds, sequence_number, description)*

Used by: *SFPTdataLogger SFPTnodeObject*

SNVT_amp

Overview:

Electric current (Amperes) .

Details:

Standard: *yes*

Resource Set: *Standard 00:00:00:00:00:00:00:00-0*

Index: *1*

Obsolete: *no*

Size: *2*

Programmatic Name: *SNVT_amp*

Neuron C Type: *signed long*

Minimum: *-32768*

Maximum: *32767*

Scaling (A,B,C): *1, -1, 0*

Scaled value: *$1 * 10^{-1} * (Raw + 0)$*

Resolution: *0.1*

Formats: *SNVT_amp: text("%f")*

Used by: *SFPTvariableSpeedMotorDrive SNVT_pump_sensor*

SNVT_amp_ac

Overview:

Amperage in alternating current (amperes AC) .

Details:

Standard: *yes*

Resource Set: *Standard 00:00:00:00:00:00:00:00-0*

Index:	<i>139</i>
Obsolete:	<i>no</i>
Size:	<i>2</i>
Programmatic Name:	<i>SNVT_amp_ac</i>
Neuron C Type:	unsigned long
Minimum:	<i>0</i>
Maximum:	<i>65535</i>
Invalid:	<i>65535</i>
Scaling (A,B,C):	<i>1, 0, 0</i>
Scaled value:	<i>$1 * 10^0 * (Raw + 0)$</i>
Resolution:	<i>1</i>
Formats:	<i>SNVT_amp_ac: text("%f")</i>
Used by:	<i>SFPTautomaticTransferSwitch SFPTgeneratorSet</i>

SNVT_amp_f

Overview:

Electric current (Amperes) .

Details:

Standard:	<i>yes</i>
Resource Set:	<i>Standard 00:00:00:00:00:00:00:00-0</i>
Index:	<i>48</i>
Obsolete:	<i>no</i>
Size:	<i>4</i>
Programmatic Name:	<i>SNVT_amp_f</i>
Neuron C Type:	float
Minimum:	<i>-3.4028234663853E+038</i>
Maximum:	<i>3.4028234663853E+038</i>
Formats:	<i>SNVT_amp_f: text("%f")</i>

SNVT_amp_mil

Overview:

Electric current (milliAmperes) .

Details:

Standard: *yes*
 Resource Set: *Standard 00:00:00:00:00:00:00:00-0*
 Index: *2*
 Obsolete: *no*
 Size: *2*
 Programmatic Name: *SNVT_amp_mil*
 Neuron C Type: *signed long*
 Minimum: *-32768*
 Maximum: *32767*
 Scaling (A,B,C): *1, -1, 0*
 Scaled value: *$1 * 10^{-1} * (Raw + 0)$*
 Resolution: *0.1*
 Formats: *SNVT_amp_mil: text("%f")*

SNVT_angle**Overview:**

Angular distance (radians) .

Details:

Standard: *yes*
 Resource Set: *Standard 00:00:00:00:00:00:00:00-0*
 Index: *3*
 Obsolete: *no*
 Size: *2*
 Programmatic Name: *SNVT_angle*
 Neuron C Type: *unsigned long*
 Minimum: *0*
 Maximum: *65535*
 Scaling (A,B,C): *1, -3, 0*
 Scaled value: *$1 * 10^{-3} * (Raw + 0)$*
 Resolution: *0.001*
 Formats: *SNVT_angle: text("%f")*

SNVT_angle_deg

Overview:

Angular distance (degrees) .

Details:

Standard:	<i>yes</i>
Resource Set:	<i>Standard 00:00:00:00:00:00:00:00-0</i>
Index:	<i>104</i>
Obsolete:	<i>no</i>
Size:	<i>2</i>
Programmatic Name:	<i>SNVT_angle_deg</i>
Neuron C Type:	signed long
	Minimum: <i>-17999</i>
	Maximum: <i>18000</i>
	Invalid: <i>32767</i>
	Scaling (A,B,C): <i>2, -2, 0</i>
	Scaled value: <i>$2 * 10^{-2} * (Raw + 0)$</i>
	Resolution: <i>0.02</i>
Formats:	<i>SNVT_angle_deg: text("%f")</i>
Used by:	<i>SCPTminDeltaAngl SCPTnomAngle SCPTorientation</i> <i>SFPTdamperActuator SFPTisiSunblindActuator</i> <i>SFPTsunblindController SNVT_pos_ctrl</i>

SNVT_angle_f

Overview:

Angular distance (radians) .

Details:

Standard:	<i>yes</i>
Resource Set:	<i>Standard 00:00:00:00:00:00:00:00-0</i>
Index:	<i>49</i>
Obsolete:	<i>no</i>
Size:	<i>4</i>
Programmatic Name:	<i>SNVT_angle_f</i>
Neuron C Type:	float
	Minimum: <i>-3.4028234663853E+038</i>
	Maximum: <i>3.4028234663853E+038</i>

Formats: *SNVT_angle_f: text("%f")*

SNVT_angle_vel

Overview:

Angular velocity (radians/second) .

Details:

Standard:	<i>yes</i>
Resource Set:	<i>Standard 00:00:00:00:00:00:00:00-0</i>
Index:	<i>4</i>
Obsolete:	<i>no</i>
Size:	<i>2</i>
Programmatic Name:	<i>SNVT_angle_vel</i>
Neuron C Type:	signed long
	Minimum: <i>-32768</i>
	Maximum: <i>32767</i>
	Scaling (A,B,C): <i>1, -1, 0</i>
	Scaled value: <i>1 *10⁻¹ *(Raw+0)</i>
	Resolution: <i>0.1</i>
Formats:	<i>SNVT_angle_vel#SI: text("%f")</i>
	<i>SNVT_angle_vel#US: text("%f", *9.5493+0(0:970))</i>

SNVT_angle_vel_f

Overview:

Angular velocity (radians/second) .

Details:

Standard:	<i>yes</i>
Resource Set:	<i>Standard 00:00:00:00:00:00:00:00-0</i>
Index:	<i>50</i>
Obsolete:	<i>no</i>
Size:	<i>4</i>

Programmatic Name: *SNVT_angle_vel_f*
 Neuron C Type: *float*
 Minimum: *-3.4028234663853E+038*
 Maximum: *3.4028234663853E+038*
 Formats: *SNVT_angle_vel_f#SI: text("%f")*
 *SNVT_angle_vel_f#US: text("%f", *9.5493+0(0:970))*

SNVT_area

Overview:

Area (square meters) .

Details:

Standard: *yes*
 Resource Set: *Standard 00:00:00:00:00:00:00:00-0*
 Index: *110*
 Obsolete: *no*
 Size: *2*
 Programmatic Name: *SNVT_area*
 Neuron C Type: *unsigned long*
 Minimum: *0*
 Maximum: *65535*
 Invalid: *65535*
 Scaling (A,B,C): *2, -4, 0*
 Scaled value: *2 *10⁻⁴ *(Raw+0)*
 Resolution: *0.0002*
 Formats: *SNVT_area#SI: text("%f")*
 *SNVT_area#US: text("%f", *10.7639+0(0:884))*
 Used by: *SCPTareaDuctHeat SCPTductArea*

SNVT_btu_f

Overview:

Thermal energy (Btus) .

Details:

Standard: *yes*
 Resource Set: *Standard 00:00:00:00:00:00:00:00-0*
 Index: *67*
 Obsolete: *no*
 Size: *4*
 Programmatic Name: *SNVT_btu_f*
 Neuron C Type: *float*
 Minimum: *0*
 Maximum: *3.4028234663853E+038*
 Formats: *SNVT_btu_f: text("%f")*

SNVT_btu_kilo**Overview:**

Thermal energy (kilo-Btus) .

Details:

Standard: *yes*
 Resource Set: *Standard 00:00:00:00:00:00:00:00-0*
 Index: *5*
 Obsolete: *no*
 Size: *2*
 Programmatic Name: *SNVT_btu_kilo*
 Neuron C Type: *unsigned long*
 Minimum: *0*
 Maximum: *65535*
 Scaling (A,B,C): *1, 0, 0*
 Scaled value: *$1 * 10^0 * (Raw + 0)$*
 Resolution: *1*
 Formats: *SNVT_btu_kilo: text("%d")*

SNVT_btu_mega**Overview:**

Thermal energy (mega-Btus) .

Details:

Standard: *yes*
 Resource Set: *Standard 00:00:00:00:00:00:00:00-0*
 Index: *6*
 Obsolete: *no*
 Size: *2*
 Programmatic Name: *SNVT_btu_mega*
 Neuron C Type: unsigned long
 Minimum: *0*
 Maximum: *65535*
 Scaling (A,B,C): *1, 0, 0*
 Scaled value: $1 * 10^0 * (Raw + 0)$
 Resolution: *1*
 Formats: *SNVT_btu_mega: text("%d")*

SNVT_char_ascii

Overview:

ASCII character (8-bit ASCII character) .

Details:

Standard: *yes*
 Resource Set: *Standard 00:00:00:00:00:00:00:00-0*
 Index: *7*
 Obsolete: *no*
 Size: *1*
 Programmatic Name: *SNVT_char_ascii*
 Neuron C Type: unsigned char
 Minimum: *0*
 Maximum: *255*
 Scaling (A,B,C): *1, 0, 0*
 Scaled value: $1 * 10^0 * (Raw + 0)$
 Resolution: *1*
 Formats: *SNVT_char_ascii: text("%c")*
 Used by: *SFPTtelephoneDirectory*

SNVT_chlr_status

Overview:

Chiller status (run mode, op mode, state bits) .

```
typedef struct {  
  chiller_t chlr_run_mode;  
  hvac_t chlr_op_mode;  
  struct {  
    unsigned in_alarm :1; // offset 0  
    unsigned run_enabled :1; // offset 1  
    unsigned local :1; // offset 2  
    unsigned limited :1; // offset 3  
    unsigned chw_flow :1; // offset 4  
    unsigned condw_flow :1; // offset 5  
    /* The last two bits (offset 6) are not defined */  
  } chlr_state;  
} SNVT_chlr_status;
```

Details:

Standard: *yes*
Resource Set: *Standard 00:00:00:00:00:00:00:00-0*
Index: *127*
Obsolete: *no*
Size: *3*
Programmatic Name: *SNVT_chlr_status*

Neuron C Type:

Structure	
chlr_run_mode	Chiller run mode (chiller run mode names) .
	<i>chiller_t</i>
chlr_op_mode	Chiller operating mode (HVAC mode names) .
	<i>hvac_t</i>
chlr_state	Chiller state flags (alarm, enabled, local, limited, chiller water flow, condenser water flow) .
Structure	
in_alarm	Alarm flag (boolean) .
	bitfield Signed: <i>no</i> Width: <i>1</i> Offset: <i>0</i> Minimum: <i>0</i> Maximum: <i>1</i> Scaling (A,B,C): <i>1, 0, 0</i> Scaled value: <i>$1 * 10^0 * (Raw + 0)$</i>

	Resolution: <i>1</i>
run_enabled	Run-enabled flag (boolean) .
	bitfield Signed: <i>no</i> Width: <i>1</i> Offset: <i>1</i> Minimum: <i>0</i> Maximum: <i>1</i> Scaling <i>1, 0, 0</i> (A,B,C): Scaled <i>1 * 10⁰ *(Raw+0)</i> value: Resolution: <i>1</i>
local	Locally-controlled flag (boolean) .
	bitfield Signed: <i>no</i> Width: <i>1</i> Offset: <i>2</i> Minimum: <i>0</i> Maximum: <i>1</i> Scaling <i>1, 0, 0</i> (A,B,C): Scaled <i>1 * 10⁰ *(Raw+0)</i> value: Resolution: <i>1</i>
limited	Limited-condition flag (boolean) . Conditions may exist that prevent reaching the setpoint
	bitfield Signed: <i>no</i> Width: <i>1</i> Offset: <i>3</i> Minimum: <i>0</i> Maximum: <i>1</i> Scaling <i>1, 0, 0</i> (A,B,C): Scaled <i>1 * 10⁰ *(Raw+0)</i> value: Resolution: <i>1</i>
chw_flow	Chiller-water-flow flag (boolean) .
	bitfield Signed: <i>no</i> Width: <i>1</i> Offset: <i>4</i> Minimum: <i>0</i>

		Maximum: <i>1</i> Scaling <i>1, 0, 0</i> (A,B,C): Scaled <i>1 * 10⁰ *(Raw+0)</i> value: Resolution: <i>1</i>
	condw_flow	Condenser-water-flow flag (boolean) .
		bitfield Signed: <i>no</i> Width: <i>1</i> Offset: <i>5</i> Minimum: <i>0</i> Maximum: <i>1</i> Scaling <i>1, 0, 0</i> (A,B,C): Scaled <i>1 * 10⁰ *(Raw+0)</i> value: Resolution: <i>1</i>

Formats: *SNVT_chlr_status: text("%m,%m,%d,%d,%d,%d,%d,%d",
chlr_run_mode, chlr_op_mode, chlr_state.in_alarm,
chlr_state.run_enabled, chlr_state.local, chlr_state.limited,
chlr_state.chw_flow, chlr_state.condw_flow)*

*SNVT_chlr_status#LO: text("%m|m|%d|%d|%d|%d|%d|%d",
chlr_run_mode, chlr_op_mode, chlr_state.in_alarm,
chlr_state.run_enabled, chlr_state.local, chlr_state.limited,
chlr_state.chw_flow, chlr_state.condw_flow)*

Used by: *SFPTchiller*

SNVT_clothes_w_a

Overview:

Clothes Washer Alarm. Used to provide alarm status for a clothes washer

Details:

Standard: *yes*

Resource Set: *Standard 00:00:00:00:00:00:00:00-0*

Index: *187*

Obsolete: *no*

Size: *6*

Programmatic Name: *SNVT_clothes_w_a*

Neuron C Type:

Structure	
alarm	.
	Structure
	alarm_reset
	.
	bitfield
	Signed: <i>no</i>
	Width: <i>1</i>
	Offset: <i>0</i>
	Minimum: <i>0</i>
	Maximum: <i>1</i>
	Scaling <i>1, 0, 0</i> (A,B,C):
	Scaled <i>1 * 10⁰ *</i> value: <i>(Raw+0)</i>
	Resolution: <i>1</i>
	war_water_supply
	.
	bitfield
	Signed: <i>no</i>
	Width: <i>1</i>
	Offset: <i>1</i>
	Minimum: <i>0</i>
	Maximum: <i>1</i>
	Scaling <i>1, 0, 0</i> (A,B,C):
	Scaled <i>1 * 10⁰ *</i> value: <i>(Raw+0)</i>
	Resolution: <i>1</i>
	war_drain_slow
	.
	bitfield
	Signed: <i>no</i>
	Width: <i>1</i>
	Offset: <i>2</i>
	Minimum: <i>0</i>
	Maximum: <i>1</i>
	Scaling <i>1, 0, 0</i> (A,B,C):
	Scaled <i>1 * 10⁰ *</i> value: <i>(Raw+0)</i>
	Resolution: <i>1</i>
	war_door_open
	.
	bitfield
	Signed: <i>no</i>
	Width: <i>1</i>
	Offset: <i>3</i>

	Minimum: <i>0</i> Maximum: <i>1</i> Scaling <i>1, 0, 0</i> (A,B,C): Scaled <i>1 * 10⁰ *</i> value: <i>(Raw+0)</i> Resolution: <i>1</i>
war_load_unbalanced	.
	bitfield Signed: <i>no</i> Width: <i>1</i> Offset: <i>4</i> Minimum: <i>0</i> Maximum: <i>1</i> Scaling <i>1, 0, 0</i> (A,B,C): Scaled <i>1 * 10⁰ *</i> value: <i>(Raw+0)</i> Resolution: <i>1</i>
war_filter_cleaning	.
	bitfield Signed: <i>no</i> Width: <i>1</i> Offset: <i>5</i> Minimum: <i>0</i> Maximum: <i>1</i> Scaling <i>1, 0, 0</i> (A,B,C): Scaled <i>1 * 10⁰ *</i> value: <i>(Raw+0)</i> Resolution: <i>1</i>
war_hoses_reversed	.
	bitfield Signed: <i>no</i> Width: <i>1</i> Offset: <i>6</i> Minimum: <i>0</i> Maximum: <i>1</i> Scaling <i>1, 0, 0</i> (A,B,C): Scaled <i>1 * 10⁰ *</i> value: <i>(Raw+0)</i> Resolution: <i>1</i>
war_voltage_low	.

	bitfield Signed: <i>no</i> Width: <i>1</i> Offset: <i>7</i> Minimum: <i>0</i> Maximum: <i>1</i> Scaling <i>1, 0, 0</i> (A,B,C): Scaled <i>1 * 10⁰ *</i> value: <i>(Raw+0)</i> Resolution: <i>1</i>
war_power_failure	.
	bitfield Signed: <i>no</i> Width: <i>1</i> Offset: <i>0</i> Minimum: <i>0</i> Maximum: <i>1</i> Scaling <i>1, 0, 0</i> (A,B,C): Scaled <i>1 * 10⁰ *</i> value: <i>(Raw+0)</i> Resolution: <i>1</i>
war_drain_open	.
	bitfield Signed: <i>no</i> Width: <i>1</i> Offset: <i>1</i> Minimum: <i>0</i> Maximum: <i>1</i> Scaling <i>1, 0, 0</i> (A,B,C): Scaled <i>1 * 10⁰ *</i> value: <i>(Raw+0)</i> Resolution: <i>1</i>
war_execute_fail	.
	bitfield Signed: <i>no</i> Width: <i>1</i> Offset: <i>2</i> Minimum: <i>0</i> Maximum: <i>1</i> Scaling <i>1, 0, 0</i> (A,B,C): Scaled <i>1 * 10⁰ *</i>

	value: $(Raw+0)$ Resolution: 1
war_door_locked	.
	bitfield Signed: <i>no</i> Width: 1 Offset: 3 Minimum: 0 Maximum: 1 Scaling $1, 0, 0$ (A,B,C): Scaled $1 * 10^0 *$ value: $(Raw+0)$ Resolution: 1
war_service	.
	bitfield Signed: <i>no</i> Width: 1 Offset: 4 Minimum: 0 Maximum: 1 Scaling $1, 0, 0$ (A,B,C): Scaled $1 * 10^0 *$ value: $(Raw+0)$ Resolution: 1
rsrvd5	.
	bitfield Signed: <i>no</i> Width: 1 Offset: 5 Minimum: 0 Maximum: 1 Scaling $1, 0, 0$ (A,B,C): Scaled $1 * 10^0 *$ value: $(Raw+0)$ Resolution: 1
rsrvd6	.
	bitfield Signed: <i>no</i> Width: 1 Offset: 6 Minimum: 0

	Maximum: <i>1</i> Scaling <i>1, 0, 0</i> (A,B,C): Scaled <i>1 *10⁰ *</i> value: <i>(Raw+0)</i> Resolution: <i>1</i>
rsrvd7	.
	bitfield Signed: <i>no</i> Width: <i>1</i> Offset: <i>7</i> Minimum: <i>0</i> Maximum: <i>1</i> Scaling <i>1, 0, 0</i> (A,B,C): Scaled <i>1 *10⁰ *</i> value: <i>(Raw+0)</i> Resolution: <i>1</i>
err_motor_stall	.
	bitfield Signed: <i>no</i> Width: <i>1</i> Offset: <i>0</i> Minimum: <i>0</i> Maximum: <i>1</i> Scaling <i>1, 0, 0</i> (A,B,C): Scaled <i>1 *10⁰ *</i> value: <i>(Raw+0)</i> Resolution: <i>1</i>
err_water_temp	.
	bitfield Signed: <i>no</i> Width: <i>1</i> Offset: <i>1</i> Minimum: <i>0</i> Maximum: <i>1</i> Scaling <i>1, 0, 0</i> (A,B,C): Scaled <i>1 *10⁰ *</i> value: <i>(Raw+0)</i> Resolution: <i>1</i>
err_pressure	.
	bitfield

	Signed: <i>no</i> Width: <i>1</i> Offset: <i>2</i> Minimum: <i>0</i> Maximum: <i>1</i> Scaling <i>1, 0, 0</i> (A,B,C): Scaled <i>1 * 10⁰ *</i> value: <i>(Raw+0)</i> Resolution: <i>1</i>
err_overflow	.
	bitfield Signed: <i>no</i> Width: <i>1</i> Offset: <i>3</i> Minimum: <i>0</i> Maximum: <i>1</i> Scaling <i>1, 0, 0</i> (A,B,C): Scaled <i>1 * 10⁰ *</i> value: <i>(Raw+0)</i> Resolution: <i>1</i>
err_water_heat	.
	bitfield Signed: <i>no</i> Width: <i>1</i> Offset: <i>4</i> Minimum: <i>0</i> Maximum: <i>1</i> Scaling <i>1, 0, 0</i> (A,B,C): Scaled <i>1 * 10⁰ *</i> value: <i>(Raw+0)</i> Resolution: <i>1</i>
err_water_leak	.
	bitfield Signed: <i>no</i> Width: <i>1</i> Offset: <i>5</i> Minimum: <i>0</i> Maximum: <i>1</i> Scaling <i>1, 0, 0</i> (A,B,C): Scaled <i>1 * 10⁰ *</i> value: <i>(Raw+0)</i>

	Resolution: <i>1</i>
err_motor_speed	.
	bitfield Signed: <i>no</i> Width: <i>1</i> Offset: <i>6</i> Minimum: <i>0</i> Maximum: <i>1</i> Scaling (A,B,C): <i>1, 0, 0</i> Scaled value: $1 * 10^0 * (Raw + 0)$ Resolution: <i>1</i>
err_wash_thermistor	.
	bitfield Signed: <i>no</i> Width: <i>1</i> Offset: <i>7</i> Minimum: <i>0</i> Maximum: <i>1</i> Scaling (A,B,C): <i>1, 0, 0</i> Scaled value: $1 * 10^0 * (Raw + 0)$ Resolution: <i>1</i>
err_dry_thermistor	.
	bitfield Signed: <i>no</i> Width: <i>1</i> Offset: <i>0</i> Minimum: <i>0</i> Maximum: <i>1</i> Scaling (A,B,C): <i>1, 0, 0</i> Scaled value: $1 * 10^0 * (Raw + 0)$ Resolution: <i>1</i>
err_dry_overheat	.
	bitfield Signed: <i>no</i> Width: <i>1</i> Offset: <i>1</i> Minimum: <i>0</i> Maximum: <i>1</i>

	Scaling <i>1, 0, 0</i> (A,B,C): Scaled <i>1 * 10⁰ *</i> value: <i>(Raw+0)</i> Resolution: <i>1</i>
err_dry_heating	.
	bitfield Signed: <i>no</i> Width: <i>1</i> Offset: <i>2</i> Minimum: <i>0</i> Maximum: <i>1</i> Scaling <i>1, 0, 0</i> (A,B,C): Scaled <i>1 * 10⁰ *</i> value: <i>(Raw+0)</i> Resolution: <i>1</i>
err_dry_fan	.
	bitfield Signed: <i>no</i> Width: <i>1</i> Offset: <i>3</i> Minimum: <i>0</i> Maximum: <i>1</i> Scaling <i>1, 0, 0</i> (A,B,C): Scaled <i>1 * 10⁰ *</i> value: <i>(Raw+0)</i> Resolution: <i>1</i>
err_rsrvd4	.
	bitfield Signed: <i>no</i> Width: <i>1</i> Offset: <i>4</i> Minimum: <i>0</i> Maximum: <i>1</i> Scaling <i>1, 0, 0</i> (A,B,C): Scaled <i>1 * 10⁰ *</i> value: <i>(Raw+0)</i> Resolution: <i>1</i>
err_rsrvd5	.
	bitfield Signed: <i>no</i>

	Width: <i>1</i> Offset: <i>5</i> Minimum: <i>0</i> Maximum: <i>1</i> Scaling <i>1, 0, 0</i> (A,B,C): Scaled <i>1 * 10⁰ *</i> value: <i>(Raw+0)</i> Resolution: <i>1</i>
err_rsrvd6	.
	bitfield Signed: <i>no</i> Width: <i>1</i> Offset: <i>6</i> Minimum: <i>0</i> Maximum: <i>1</i> Scaling <i>1, 0, 0</i> (A,B,C): Scaled <i>1 * 10⁰ *</i> value: <i>(Raw+0)</i> Resolution: <i>1</i>
err_rsrvd7	.
	bitfield Signed: <i>no</i> Width: <i>1</i> Offset: <i>7</i> Minimum: <i>0</i> Maximum: <i>1</i> Scaling <i>1, 0, 0</i> (A,B,C): Scaled <i>1 * 10⁰ *</i> value: <i>(Raw+0)</i> Resolution: <i>1</i>
err_rsrvd0_7	.
	bitfield Signed: <i>no</i> Width: <i>8</i> Offset: <i>0</i> Minimum: <i>0</i> Maximum: <i>1</i> Scaling <i>1, 0, 0</i> (A,B,C): Scaled <i>1 * 10⁰ *</i> value: <i>(Raw+0)</i> Resolution: <i>1</i>

[illegible]

Used by: *SFPTclothesWasherDomestic*

Overview:

Clothes Washer Command. Used to program and start a clothes washer

Details:

Standard:	<i>yes</i>
Resource Set:	<i>Standard 00:00:00:00:00:00:00:00-0</i>
Index:	<i>184</i>
Obsolete:	<i>no</i>
Size:	<i>21</i>
Programmatic Name:	<i>SNVT_clothes_w_c</i>

Neuron C Type:

Structure	
cycle	.
	<i>appl_cwc_t</i>

subcycle	.														
	<i>appl_cws_t</i>														
rervd	.														
	unsigned short Minimum: 0 Maximum: 255 Scaling 1, 0, 0 (A,B,C): Scaled value: $1 * 10^0 * (Raw+0)$ Resolution: 1														
action	.														
	<table> <tr> <th colspan="2">Structure</th></tr> <tr> <td>power_on</td><td>.</td></tr> <tr> <td></td><td> bitfield Signed: no Width: 1 Offset: 0 Minimum: 0 Maximum: 1 Scaling 1, 0, 0 (A,B,C): Scaled value: $1 * 10^0 * (Raw+0)$ Resolution: 1 </td></tr> <tr> <td>run_mode</td><td>.</td></tr> <tr> <td></td><td> bitfield Signed: no Width: 1 Offset: 1 Minimum: 0 Maximum: 1 Scaling 1, 0, 0 (A,B,C): Scaled value: $1 * 10^0 * (Raw+0)$ Resolution: 1 </td></tr> <tr> <td>rsrvd2_7</td><td>.</td></tr> <tr> <td></td><td> bitfield Signed: no Width: 6 Offset: 2 Minimum: 0 Maximum: 63 </td></tr> </table>	Structure		power_on	.		bitfield Signed: no Width: 1 Offset: 0 Minimum: 0 Maximum: 1 Scaling 1, 0, 0 (A,B,C): Scaled value: $1 * 10^0 * (Raw+0)$ Resolution: 1	run_mode	.		bitfield Signed: no Width: 1 Offset: 1 Minimum: 0 Maximum: 1 Scaling 1, 0, 0 (A,B,C): Scaled value: $1 * 10^0 * (Raw+0)$ Resolution: 1	rsrvd2_7	.		bitfield Signed: no Width: 6 Offset: 2 Minimum: 0 Maximum: 63
Structure															
power_on	.														
	bitfield Signed: no Width: 1 Offset: 0 Minimum: 0 Maximum: 1 Scaling 1, 0, 0 (A,B,C): Scaled value: $1 * 10^0 * (Raw+0)$ Resolution: 1														
run_mode	.														
	bitfield Signed: no Width: 1 Offset: 1 Minimum: 0 Maximum: 1 Scaling 1, 0, 0 (A,B,C): Scaled value: $1 * 10^0 * (Raw+0)$ Resolution: 1														
rsrvd2_7	.														
	bitfield Signed: no Width: 6 Offset: 2 Minimum: 0 Maximum: 63														

		Scaling $1, 0, 0$ (A,B,C): Scaled $1 * 10^0 * (Raw+0)$ value: Resolution: 1
function	.	
	Structure	
	program	.
		<i>appl_cwp_t</i>
	wash	.
		Structure load_level . discrete_levels_t temp . unsigned short Minimum: 0 Maximum: 255 Scaling $1, 0, 0$ (A,B,C): Scaled $1 * 10^0 *$ value: $(Raw+0)$ Resolution: 1 time . unsigned short Minimum: 0 Maximum: 255 Scaling $1, 0, 0$ (A,B,C): Scaled $1 * 10^0 *$ value: $(Raw+0)$ Resolution: 1 prewash . boolean_t
	rinse	.
		Structure temp . unsigned short Minimum: 0 Maximum: 255 Scaling $1, 0, 0$ (A,B,C): Scaled $1 * 10^0 *$

		value: $(Raw+0)$ Resolution: 1
	repeat	.
		unsigned short Minimum: 0 Maximum: 9 Scaling $1, 0, 0$ (A,B,C): Scaled $1 * 10^0 *$ value: $(Raw+0)$ Resolution: 1
	option	.
		<i>appl_rin_t</i>
spin		.
		Structure
	speed	.
		<i>SNVT_rpm</i>
	time	.
		unsigned short Minimum: 0 Maximum: 255 Scaling $1, 0, 0$ (A,B,C): Scaled $1 * 10^0 *$ value: $(Raw+0)$ Resolution: 1
	hold	.
		<i>boolean_t</i>
dry		.
		Structure
	temp	.
		unsigned short Minimum: 0 Maximum: 1 Scaling $1, 0, 0$ (A,B,C): Scaled $1 * 10^0 * (Raw+0)$ value: Resolution: 1
	duration	.
		Structure

			time	.
				unsigned short Minimum: 0 Maximum: 255 Scaling 1, 0, 0 (A,B,C): Scaled 1 *10 ⁰ * value: (Raw+0) Resolution: 1
			dryness	.
				discrete_levels_t
time_remaining				.
				SNVT_time_min

Formats: *SNVT_clothes_w_c: text("%m %m %d %d %d %d %m %m %d %d %m %d %d %m %d %d %m %d %d %m %d", cycle, subcycle, rervd, action.power_on, action.run_mode, action.rsrvd2_7, function.program, function.wash.load_level, function.wash.temp, function.wash.time, function.wash.prewash, function.rinse.temp, function.rinse.repeat, function.rinse.option, function.spin.speed, function.spin.time, function.spin.hold, function.dry.temp, function.dry.duration.time, function.dry.duration.dryness, time_remaining)*

Used by: *SFPTclothesWasherDomestic SNVT_clothes_w_s*

SNVT_clothes_w_m

Overview:

Clothes Washer-Management Status. Provides status of door/lid and drain

Details:

Standard: *yes*

Resource Set: *Standard 00:00:00:00:00:00:00:00-0*

Index: *185*

Obsolete: *no*

Size: *1*

Programmatic Name: *SNVT_clothes_w_m*

Neuron C Type:

Structure	
door_ajar	Door/Lid Ajar. The door/lid of the washer is not fully closed
	bitfield

	Signed: <i>no</i> Width: <i>1</i> Offset: <i>0</i> Minimum: <i>0</i> Maximum: <i>1</i> Scaling (A,B,C): <i>1, 0, 0</i> Scaled value: $1 * 10^0 * (Raw + 0)$ Resolution: <i>1</i>
drain_on	Drain On. The drain is on
	bitfield Signed: <i>no</i> Width: <i>1</i> Offset: <i>1</i> Minimum: <i>0</i> Maximum: <i>1</i> Scaling (A,B,C): <i>1, 0, 0</i> Scaled value: $1 * 10^0 * (Raw + 0)$ Resolution: <i>1</i>
reserved	This field is reserved.. This field is reserved.
	bitfield Signed: <i>no</i> Width: <i>6</i> Offset: <i>2</i> Minimum: <i>0</i> Maximum: <i>63</i> Scaling (A,B,C): <i>1, 0, 0</i> Scaled value: $1 * 10^0 * (Raw + 0)$ Resolution: <i>1</i>

Formats: *SNVT_clothes_w_m: text("%d %d %d", door_ajar, drain_on, reserved)*
Used by: *SFPTclothesWasherDomestic*

SNVT_clothes_w_s

Overview:

Clothes Washer Status. Used to provide present status from a clothes washer, including command and alarm information

Details:

Standard: *yes*

Resource Set: *Standard 00:00:00:00:00:00:00:00-0*

Index: *186*

Obsolete: *no*

Size: *31*

Programmatic Name: *SNVT_clothes_w_s*

Neuron C Type:

Structure	
cycle	.
	<i>appl_cwc_t</i>
subcycle	.
	<i>appl_cws_t</i>
washer_command_data	.
	<i>SNVT_clothes_w_c</i>
time_remaining	.
	<i>SNVT_time_min</i>
alarm	.
	Structure
	alarm_reset
	.
	bitfield
	Signed: <i>no</i>
	Width: <i>1</i>
	Offset: <i>0</i>
	Minimum: <i>0</i>
	Maximum: <i>1</i>
	Scaling (A,B,C): <i>1, 0, 0</i>
	Scaled value: <i>1 * 10⁰ * (Raw+0)</i>
	Resolution: <i>1</i>
	war_water_supply
	.
	bitfield
	Signed: <i>no</i>
	Width: <i>1</i>
	Offset: <i>1</i>
	Minimum: <i>0</i>
	Maximum: <i>1</i>
	Scaling (A,B,C): <i>1, 0, 0</i>
	Scaled value: <i>1 * 10⁰ * (Raw+0)</i>

	Resolution: <i>1</i>
war_drain_slow	.
	bitfield Signed: <i>no</i> Width: <i>1</i> Offset: <i>2</i> Minimum: <i>0</i> Maximum: <i>1</i> Scaling <i>1, 0, 0</i> (A,B,C): Scaled <i>1 * 10⁰ *</i> value: <i>(Raw+0)</i> Resolution: <i>1</i>
war_door_open	.
	bitfield Signed: <i>no</i> Width: <i>1</i> Offset: <i>3</i> Minimum: <i>0</i> Maximum: <i>1</i> Scaling <i>1, 0, 0</i> (A,B,C): Scaled <i>1 * 10⁰ *</i> value: <i>(Raw+0)</i> Resolution: <i>1</i>
war_load_unbalanced	.
	bitfield Signed: <i>no</i> Width: <i>1</i> Offset: <i>4</i> Minimum: <i>0</i> Maximum: <i>1</i> Scaling <i>1, 0, 0</i> (A,B,C): Scaled <i>1 * 10⁰ *</i> value: <i>(Raw+0)</i> Resolution: <i>1</i>
war_filter_cleaning	.
	bitfield Signed: <i>no</i> Width: <i>1</i> Offset: <i>5</i> Minimum: <i>0</i> Maximum: <i>1</i>

	Scaling <i>1, 0, 0</i> (A,B,C): Scaled <i>1 *10⁰ *</i> value: <i>(Raw+0)</i> Resolution: <i>1</i>
war_hoses_reversed	.
	bitfield Signed: <i>no</i> Width: <i>1</i> Offset: <i>6</i> Minimum: <i>0</i> Maximum: <i>1</i> Scaling <i>1, 0, 0</i> (A,B,C): Scaled <i>1 *10⁰ *</i> value: <i>(Raw+0)</i> Resolution: <i>1</i>
war_voltage_low	.
	bitfield Signed: <i>no</i> Width: <i>1</i> Offset: <i>7</i> Minimum: <i>0</i> Maximum: <i>1</i> Scaling <i>1, 0, 0</i> (A,B,C): Scaled <i>1 *10⁰ *</i> value: <i>(Raw+0)</i> Resolution: <i>1</i>
war_power_failure	.
	bitfield Signed: <i>no</i> Width: <i>1</i> Offset: <i>0</i> Minimum: <i>0</i> Maximum: <i>1</i> Scaling <i>1, 0, 0</i> (A,B,C): Scaled <i>1 *10⁰ *</i> value: <i>(Raw+0)</i> Resolution: <i>1</i>
war_drain_open	.
	bitfield Signed: <i>no</i>

	Width: <i>1</i> Offset: <i>1</i> Minimum: <i>0</i> Maximum: <i>1</i> Scaling (A,B,C): <i>1, 0, 0</i> Scaled value: $1 * 10^0 * (Raw+0)$ Resolution: <i>1</i>
war_execute_fail	.
	bitfield Signed: <i>no</i> Width: <i>1</i> Offset: <i>2</i> Minimum: <i>0</i> Maximum: <i>1</i> Scaling (A,B,C): <i>1, 0, 0</i> Scaled value: $1 * 10^0 * (Raw+0)$ Resolution: <i>1</i>
war_door_locked	.
	bitfield Signed: <i>no</i> Width: <i>1</i> Offset: <i>3</i> Minimum: <i>0</i> Maximum: <i>1</i> Scaling (A,B,C): <i>1, 0, 0</i> Scaled value: $1 * 10^0 * (Raw+0)$ Resolution: <i>1</i>
war_service	.
	bitfield Signed: <i>no</i> Width: <i>1</i> Offset: <i>4</i> Minimum: <i>0</i> Maximum: <i>1</i> Scaling (A,B,C): <i>1, 0, 0</i> Scaled value: $1 * 10^0 * (Raw+0)$ Resolution: <i>1</i>

war_rsrvd5	.
	bitfield Signed: <i>no</i> Width: <i>1</i> Offset: <i>5</i> Minimum: <i>0</i> Maximum: <i>1</i> Scaling (A,B,C): <i>1, 0, 0</i> Scaled value: $1 * 10^0 * (Raw + 0)$ Resolution: <i>1</i>
war_rsrvd6	.
	bitfield Signed: <i>no</i> Width: <i>1</i> Offset: <i>6</i> Minimum: <i>0</i> Maximum: <i>1</i> Scaling (A,B,C): <i>1, 0, 0</i> Scaled value: $1 * 10^0 * (Raw + 0)$ Resolution: <i>1</i>
war_rsrvd7	.
	bitfield Signed: <i>no</i> Width: <i>1</i> Offset: <i>7</i> Minimum: <i>0</i> Maximum: <i>1</i> Scaling (A,B,C): <i>1, 0, 0</i> Scaled value: $1 * 10^0 * (Raw + 0)$ Resolution: <i>1</i>
err_motor_stall	.
	bitfield Signed: <i>no</i> Width: <i>1</i> Offset: <i>0</i> Minimum: <i>0</i> Maximum: <i>1</i> Scaling (A,B,C): <i>1, 0, 0</i>

	Scaled value: $1 * 10^0 * (Raw+0)$ Resolution: 1
err_water_temp	.
	bitfield Signed: <i>no</i> Width: 1 Offset: 1 Minimum: 0 Maximum: 1 Scaling (A,B,C): 1, 0, 0 Scaled value: $1 * 10^0 * (Raw+0)$ Resolution: 1
err_pressure	.
	bitfield Signed: <i>no</i> Width: 1 Offset: 2 Minimum: 0 Maximum: 1 Scaling (A,B,C): 1, 0, 0 Scaled value: $1 * 10^0 * (Raw+0)$ Resolution: 1
err_overflow	.
	bitfield Signed: <i>no</i> Width: 1 Offset: 3 Minimum: 0 Maximum: 1 Scaling (A,B,C): 1, 0, 0 Scaled value: $1 * 10^0 * (Raw+0)$ Resolution: 1
err_water_heat	.
	bitfield Signed: <i>no</i> Width: 1 Offset: 4

	Minimum: <i>0</i> Maximum: <i>1</i> Scaling (A,B,C): <i>1, 0, 0</i> Scaled value: $1 * 10^0 * (Raw+0)$ Resolution: <i>1</i>
err_water_leak	.
	bitfield Signed: <i>no</i> Width: <i>1</i> Offset: <i>5</i> Minimum: <i>0</i> Maximum: <i>1</i> Scaling (A,B,C): <i>1, 0, 0</i> Scaled value: $1 * 10^0 * (Raw+0)$ Resolution: <i>1</i>
err_motor_speed	.
	bitfield Signed: <i>no</i> Width: <i>1</i> Offset: <i>6</i> Minimum: <i>0</i> Maximum: <i>1</i> Scaling (A,B,C): <i>1, 0, 0</i> Scaled value: $1 * 10^0 * (Raw+0)$ Resolution: <i>1</i>
err_wash_thermistor	.
	bitfield Signed: <i>no</i> Width: <i>1</i> Offset: <i>7</i> Minimum: <i>0</i> Maximum: <i>1</i> Scaling (A,B,C): <i>1, 0, 0</i> Scaled value: $1 * 10^0 * (Raw+0)$ Resolution: <i>1</i>
err_dry_thermistor	.

	bitfield Signed: <i>no</i> Width: <i>1</i> Offset: <i>0</i> Minimum: <i>0</i> Maximum: <i>1</i> Scaling (A,B,C): <i>1, 0, 0</i> Scaled value: $1 * 10^0 * (Raw + 0)$ Resolution: <i>1</i>
err_dry_overheat	.
	bitfield Signed: <i>no</i> Width: <i>1</i> Offset: <i>1</i> Minimum: <i>0</i> Maximum: <i>1</i> Scaling (A,B,C): <i>1, 0, 0</i> Scaled value: $1 * 10^0 * (Raw + 0)$ Resolution: <i>1</i>
err_dry_heating	.
	bitfield Signed: <i>no</i> Width: <i>1</i> Offset: <i>2</i> Minimum: <i>0</i> Maximum: <i>1</i> Scaling (A,B,C): <i>1, 0, 0</i> Scaled value: $1 * 10^0 * (Raw + 0)$ Resolution: <i>1</i>
err_dry_fan	.
	bitfield Signed: <i>no</i> Width: <i>1</i> Offset: <i>3</i> Minimum: <i>0</i> Maximum: <i>1</i> Scaling (A,B,C): <i>1, 0, 0</i> Scaled value: $1 * 10^0 *$

	value: $(Raw+0)$ Resolution: 1
err_rsrvd4	.
	bitfield Signed: <i>no</i> Width: 1 Offset: 4 Minimum: 0 Maximum: 1 Scaling $1, 0, 0$ (A,B,C): Scaled $1 * 10^0 *$ value: $(Raw+0)$ Resolution: 1
err_rsrvd5	.
	bitfield Signed: <i>no</i> Width: 1 Offset: 5 Minimum: 0 Maximum: 1 Scaling $1, 0, 0$ (A,B,C): Scaled $1 * 10^0 *$ value: $(Raw+0)$ Resolution: 1
err_rsrvd6	.
	bitfield Signed: <i>no</i> Width: 1 Offset: 6 Minimum: 0 Maximum: 1 Scaling $1, 0, 0$ (A,B,C): Scaled $1 * 10^0 *$ value: $(Raw+0)$ Resolution: 1
err_rsrvd7	.
	bitfield Signed: <i>no</i> Width: 1 Offset: 7 Minimum: 0

*\$NVT_clothes_w_s\$: text("%m %m %d %d %m %d
%d %m %d %d %m %d %d %m %d %d %d %d %d %d
%d %d %d %d %d %d %d %d %d %d %d %d %d %d
%d %d %d %d %d %d %d %d %d", cycle, subcycle,
washer_command_data.cycle, washer_command_data.subcycle,
washer_command_data.rervd, washer_command_data.action.power_on,
washer_command_data.action.run_mode,
washer_command_data.action.rsrvd2_7,
washer_command_data.function.program,
washer_command_data.function.wash.load_level,
washer_command_data.function.wash.temp,
washer_command_data.function.wash.time,
washer_command_data.function.wash.prewash,
washer_command_data.function.rinse.temp,
washer_command_data.function.rinse.repeat,
washer_command_data.function.rinse.option,
washer_command_data.function.spin.speed,
washer_command_data.function.spin.time,
washer_command_data.function.spin.hold,
washer_command_data.function.dry.temp,
washer_command_data.function.dry.duration.time,
washer_command_data.function.dry.duration.dryness,*

washer_command_data.time_remaining, time_remaining,
alarm.alarm_reset, alarm.war_water_supply, alarm.war_drain_slow,
alarm.war_door_open, alarm.war_load_unbalanced,
alarm.war_filter_cleaning, alarm.war_hoses_reversed,
alarm.war_voltage_low, alarm.war_power_failure,
alarm.war_drain_open, alarm.war_execute_fail, alarm.war_door_locked,
alarm.war_service, alarm.war_rsrvd5, alarm.war_rsrvd6,
alarm.war_rsrvd7, alarm.err_motor_stall, alarm.err_water_temp,
alarm.err_pressure, alarm.err_overflow, alarm.err_water_heat,
alarm.err_water_leak, alarm.err_motor_speed,
alarm.err_wash_thermistor, alarm.err_dry_thermistor,
alarm.err_dry_overheat, alarm.err_dry_heating, alarm.err_dry_fan,
alarm.err_rsrvd4, alarm.err_rsrvd5, alarm.err_rsrvd6, alarm.err_rsrvd7,
alarm.err_rsrvd0_7, alarm.manuf_code)

Used by: *SFPTclothesWasherDomestic*

SNVT_color

Overview:

CIELAB color (L*,a*,b*) .

Details:

Standard: *yes*
 Resource Set: *Standard 00:00:00:00:00:00:00:00-0*
 Index: *70*
 Obsolete: *no*
 Size: *6*
 Programmatic Name: *SNVT_color*

Neuron C Type:

Structure	
L_star	L*.
	unsigned long Minimum: 0 Maximum: 1000 Scaling 1, -1, 0 (A,B,C): Scaled value: $1 * 10^{-1} * (Raw + 0)$ Resolution: 0.1
a_star	a*.
	signed long Minimum: -2000 Maximum: 2000

	Scaling $1, -1, 0$ (A,B,C): Scaled $1 * 10^{-1} * (Raw + 0)$ value: Resolution: 0.1
b_star	b*.
	signed long Minimum: -2000 Maximum: 2000 Scaling $1, -1, 0$ (A,B,C): Scaled $1 * 10^{-1} * (Raw + 0)$ value: Resolution: 0.1

Formats: *SNVT_color: text("%f,%f,%f", L_star, a_star, b_star)*

SNVT_color#LO: text("%f|%f|%f", L_star, a_star, b_star)

SNVT_color_2

Overview:

Color.

Details:

Standard:	<i>yes</i>
Resource Set:	<i>Standard 00:00:00:00:00:00:00:00-0</i>
Index:	<i>190</i>
Obsolete:	<i>no</i>
Size:	<i>5</i>
Programmatic Name:	<i>SNVT_color_2</i>
Neuron C Type:	Structure

Structure	
encoding	Color encoding.. Color encoding specified by the color_value union; additional encodings may be added
	<i>color_encoding_t</i>
color_value	Color value.. Color value encoded as specified by the encoding field
Union	
CIE31_lumen	CIE31 color space with lumen.. CIE31 color space with Y output in lumen

		Structure	
		x	CIE31 x value (% of full level) . CIE31 x-axis color value
			unsigned short Minimum: 0 Maximum: 200 Invalid: 255 Scaling 5, -1, 0 (A,B,C): Scaled $5 * 10^{-1}$ value: * (Raw+0) Resolution: 0.5
		y	CIE31 y value (% of full level) . CIE31 y-axis color value
			unsigned short Minimum: 0 Maximum: 200 Invalid: 255 Scaling 1, 0, 0 (A,B,C): Scaled $1 * 10^0$ * value: (Raw+0) Resolution: 1
		absolute_Y	Absolute luminance (lumen) . Absolute luminance
			unsigned long Minimum: 0 Maximum: 65534 Invalid: 65535 Scaling 1, 1, 0 (A,B,C): Scaled $1 * 10^1$ * value: (Raw+0) Resolution: 10
		CIE31_percent	CIE31 color space with percent.. CIE31 color space with Y output in percent of maximum lumen output of the lamp
		Structure	
		x	CIE31 x value (% of full level) . CIE31 x-axis color value

		unsigned short Minimum: 0 Maximum: 200 Invalid: 255 Scaling 5, -1, 0 (A,B,C): Scaled value: $5 * 10^{-1} * (Raw+0)$ Resolution: 0.5
	y	CIE31 y value (% of full level) . CIE31 y-axis color value
		unsigned short Minimum: 0 Maximum: 200 Invalid: 255 Scaling 1, 0, 0 (A,B,C): Scaled value: $1 * 10^0 * (Raw+0)$ Resolution: 1
	percent_Y	Luminance (% of full level) . Y output in percent of maximum lumen output of the lamp
		unsigned long Minimum: 0 Maximum: 10000 Invalid: 65535 Scaling 1, -2, 0 (A,B,C): Scaled value: $1 * 10^{-2} * (Raw+0)$ Resolution: 0.01
RGB	RGB color value..	
	Structure	
	red	Red component.. Red component for RGB color
		unsigned short Minimum: 0 Maximum: 255 Scaling 1, 0, 0 (A,B,C):

			Scaled value: $1 * 10^0 * (Raw+0)$ Resolution: 1
		green	Green component.. Green component for RGB color
			unsigned short Minimum: 0 Maximum: 255 Scaling 1, 0, 0 (A,B,C): Scaled value: $1 * 10^0 * (Raw+0)$ Resolution: 1
		blue	Blue component.. Blue component for RGB color
			unsigned short Minimum: 0 Maximum: 255 Scaling 1, 0, 0 (A,B,C): Scaled value: $1 * 10^0 * (Raw+0)$ Resolution: 1
		color_temperature	Color temperature (degrees Kelvin) .
			unsigned short Minimum: 56 Maximum: 150 Invalid: 255 Scaling 50, 0, 0 (A,B,C): Scaled value: $50 * 10^0 * (Raw+0)$ Resolution: 50

Used by: *SCPTsceneColor SFPTisiLampActuator*

SNVT_config_src

Overview:

Configuration source (configuration source names) .

Details:

Standard:	<i>yes</i>
Resource Set:	<i>Standard 00:00:00:00:00:00:00:00-0</i>
Index:	<i>69</i>
Obsolete:	<i>no</i>
Size:	<i>1</i>
Programmatic Name:	<i>SNVT_config_src</i>
Neuron C Type:	<i>config_source_t</i>
Formats:	<i>SNVT_config_src: text("%m")</i>
Used by:	<i>SCPTnwrkCnfg</i>

SNVT_count

Overview:

Absolute count (units) .

Details:

Standard:	<i>yes</i>
Resource Set:	<i>Standard 00:00:00:00:00:00:00:00-0</i>
Index:	<i>8</i>
Obsolete:	<i>no</i>
Size:	<i>2</i>
Programmatic Name:	<i>SNVT_count</i>
Neuron C Type:	<i>unsigned long</i>
Minimum:	<i>0</i>
Maximum:	<i>65535</i>
Scaling (A,B,C):	<i>1, 0, 0</i>
Scaled value:	<i>$1 * 10^0 * (Raw + 0)$</i>
Resolution:	<i>1</i>
Formats:	<i>SNVT_count: text("%d")</i>
Used by:	<i>SCPTnumDampers SCPTnumDigits SCPTnumValves SFPTclosedLoopSensor SFPTdeviceMonitor SFPTelevatorsArrivalGong SFPTelevatorsHallLantern SFPTelevatorsPositionIndicator SFPTelevatorsVoiceAnnouncer SFPTgeneratorSet SFPTthvacValvePositioner SFPTopenLoopSensor SFPTutilityDataLoggerRegister</i>

SNVT_count_32

Overview:

Absolute count. A 32-bit counter

Details:

Standard:	<i>yes</i>
Resource Set:	<i>Standard 00:00:00:00:00:00:00:00-0</i>
Index:	<i>183</i>
Obsolete:	<i>no</i>
Size:	<i>4</i>
Programmatic Name:	<i>SNVT_count_32</i>
Neuron C Type:	unsigned quad
Minimum:	<i>0</i>
Maximum:	<i>4294967295</i>
Invalid:	<i>4294967295</i>
Scaling (A,B,C):	<i>1, 0, 0</i>
Scaled value:	<i>$1 * 10^0 * (Raw + 0)$</i>
Resolution:	<i>1</i>
Formats:	<i>SNVT_count_32: text("%d")</i>
Used by:	<i>SFPTchannelMonitor</i>

SNVT_count_f

Overview:

Absolute count (units) .

Details:

Standard:	<i>yes</i>
Resource Set:	<i>Standard 00:00:00:00:00:00:00:00-0</i>
Index:	<i>51</i>
Obsolete:	<i>no</i>
Size:	<i>4</i>
Programmatic Name:	<i>SNVT_count_f</i>
Neuron C Type:	float

Minimum: 0
Maximum: 3.4028234663853E+038

Formats: *SNVT_count_f: text("%f")*

SNVT_count_inc

Overview:

Increment count (units (delta)) .

Details:

Standard: *yes*
Resource Set: *Standard 00:00:00:00:00:00:00:00-0*
Index: *9*
Obsolete: *no*
Size: *2*
Programmatic Name: *SNVT_count_inc*
Neuron C Type: *signed long*
Minimum: *-32768*
Maximum: *32767*
Scaling (A,B,C): *1, 0, 0*
Scaled value: *$1 * 10^0 * (Raw + 0)$*
Resolution: *1*
Formats: *SNVT_count_inc: text("%d")*

SNVT_count_inc_f

Overview:

Increment count (units (delta)) .

Details:

Standard: *yes*
Resource Set: *Standard 00:00:00:00:00:00:00:00-0*
Index: *52*
Obsolete: *no*
Size: *4*

Programmatic Name: *SNVT_count_inc_f*
 Neuron C Type: *float*
 Minimum: *-3.4028234663853E+038*
 Maximum: *3.4028234663853E+038*
 Formats: *SNVT_count_inc_f: text("%f")*

SNVT_ctrl_req

Overview:

Control request (receiver ID, sender ID, sender priority) .

The network variable of this type is used in a controlling device, to request control permission of a controllable device, addressed by the logical receiver_id number. A controlling device may be an operator keyboard; a PC based control system or a remote control unit. A controllable device may a camera telemetry receiver, a video matrix switch, etc.

The controlling device must use the nciDeviceId assigned, as the value of the SNVT_ctrl_req sender_id. The SNVT_ctrl_req sender_prio must use either the optional nciPriority value assigned, or a priority value supplied from the application.

‘Low priority values’ equals “high priority,” and ‘high priority values’ equals “low priority,” such that ‘1’ is of higher priority than ‘50.’

Standard control devices (e.g., keyboards) in normal control mode must have the priority assigned in the range 1 to 50. The range 51 to 200 is used in alarm control mode. The priority value zero (0) is used to release control.

EXAMPLES

```

control.receiver_id = 1
control.sender_id   = nciDeviceId
control.sender_prio = nciGroupPriority
Physical value      = Request controllable device #1, from this controlling device, having the
                      configurable id, nciDeviceId, and configurable priority, nciGroupPriority
  
```

Details:

Standard: *yes*
 Resource Set: *Standard 00:00:00:00:00:00:00:00-0*
 Index: *148*
 Obsolete: *no*
 Size: *5*
 Programmatic Name: *SNVT_ctrl_req*
 Neuron C Type:

Structure	
receiver_id	Receiver ID (ID number) .

	unsigned long Minimum: <i>1</i> Maximum: <i>65535</i> Invalid: <i>0</i> Scaling <i>1, 0, 0</i> (A,B,C): Scaled <i>1 * 10⁰ *(Raw+0)</i> value: Resolution: <i>1</i>
sender_id	Sender ID (ID number) .
	unsigned long Minimum: <i>1</i> Maximum: <i>65535</i> Invalid: <i>65535</i> Scaling <i>1, 0, 0</i> (A,B,C): Scaled <i>1 * 10⁰ *(Raw+0)</i> value: Resolution: <i>1</i>
sender_prio	Sender priority (priority value) .
	unsigned short Minimum: <i>0</i> Maximum: <i>200</i> Scaling <i>1, 0, 0</i> (A,B,C): Scaled <i>1 * 10⁰ *(Raw+0)</i> value: Resolution: <i>1</i>

Formats: *SNVT_ctrl_req: text("%d %d %d", receiver_id, sender_id, sender_prio)*

SNVT_ctrl_resp

Overview:

Control response (status, sender, controller ID) .

The network variable of this type is used to send the status on a control request, from a controlling device (SNVT_pos_ctrl).

Valid status responses are:

CTRLR_NO, CTRLR_PEND, CTRLR_REL, CTRLR_QUERY, CTRLR_RES, CTRLR_ERR

The status response CTRLR_NO, is sent on a control request to signal a control permission has been granted to the controlling device number (no.) sent as controller_id. Depending on the priority of the requesting device and the priority of the controlling device having permission, the

controller_id may be equal to the requesting device or the one currently having permission.

The status response CTRLR_QUERY is sent to the controller currently having permission, whenever a controlling device with lower priority is requesting permission to control. This status response requires a retransmitted request from the current controller to maintain permission.

If the current controller fails to retransmit within some timeout determined by the controllable device, the permission is given to the requesting controller with the lower priority.

The response is sent to verify if the controlling device is online—e.g. to resolve control status if the controlling device has been reset, or having a power failure.

The status response CTRLR_PEND, is used when control permission is switched from one controlling device to another, due to equal priority of the controllers, optionally by polling the operator of the controlling device with a programmable timeout.

The status response CTRLR_REL is sent as response if a controlling device request to release the present control, e.g. when a keyboard selects another camera, or if the keyboard has been reset.

The status response CTRLR_RES is sent from the controllable device if it is reset, to relinquish control in a controlling device.

The status response CTRLR_ERR is sent if a bad function in the controllable device has occurred.

A union holds the logical id of the controllable device. For camera telemetry receivers this value is a fixed value configured prior to use.

For matrices, this value holds the currently selected monitor, by the specified controller in the controller_id field. The matrix is logically assigned by monitor range during configuration; thus, these values must be transmitted when the status is CTRLR_RES.

e.g., A matrix having the monitor range 1 to 16 must on reset send:

sender.range.lower = 1

sender.range.upper = 16

EXAMPLES

ctrl.status = CTRLR_NO

ctrl.sender.id = 1

ctrl.controller_id = 2

Physical value = Control permission for controllable device #1, is granted to controlling device #2

Details:

Standard: *yes*

Resource Set: *Standard 00:00:00:00:00:00:00:00-0*

Index: *149*

Obsolete: *no*

Size: *7*

Programmatic Name: *SNVT_ctrl_resp*

Neuron C Type:

Structure

status	Control response type (control response type names) .	
	<i>control_resp_t</i>	
sender	Sender ID.	
	Union	
	id	Sender ID (ID number) .
		unsigned long Minimum: <i>1</i> Maximum: <i>65535</i> Invalid: <i>65535</i> Scaling <i>1, 0, 0</i> (A,B,C): Scaled <i>1 * 10⁰ * (Raw+0)</i> value: Resolution: <i>1</i>
	range	Sender ID range (lower, upper) .
		Structure
		lower
		Sender range lower ID (ID number) .
		unsigned long Minimum: <i>1</i> Maximum: <i>65535</i> Invalid: <i>65535</i> Scaling <i>1, 0, 0</i> (A,B,C): Scaled <i>1 * 10⁰ *</i> value: <i>(Raw+0)</i> Resolution: <i>1</i>
		upper
		Sender range upper ID (ID number) .
		unsigned long Minimum: <i>1</i> Maximum: <i>65535</i> Invalid: <i>65535</i> Scaling <i>1, 0, 0</i> (A,B,C): Scaled <i>1 * 10⁰ *</i> value: <i>(Raw+0)</i> Resolution: <i>1</i>
controller_id	Controller ID (ID number) .	
	unsigned long Minimum: <i>1</i> Maximum: <i>65535</i> Invalid: <i>65535</i>	

	Scaling (A,B,C):	$1, 0, 0$
	Scaled value:	$1 * 10^0 * (Raw + 0)$
	Resolution:	1

Formats: *SNVT_ctrl_resp: text("%m %d %d %d %d", status, sender.id, sender.range.upper, sender.range.lower, controller_id)*

SNVT_currency

Overview:

Currency (unit, magnitude, value) .

Used to represent a monetary value in a specified currency. The value field is a 32-bit signed value compatible with the Neuron C Extended Arithmetic s32_type type. Positive values correspond to credits, negative values to debits.

EXAMPLES

The power_of_10 field scales the value field, so that for example, USD(\$) -1.23 is represented as the following:

{CU_UNITED_STATES_DOLLAR, -2, {-1, -1, -1, -123}} or
{CU_UNITED_STATES_DOLLAR, -2, {0xFF, 0xFF, 0xFF, 0x85}}

EUR(€) 45.67 is represented as the following:

{CU_EUROPEAN_CURRENCY_UNIT, -2, {0, 0, 17, 215}} or
{CU_EUROPEAN_CURRENCY_UNIT, -2, {0, 0, 0x11, 0xD7}}

JPY(¥) 1000000 is represented as the following:

{CU_JAPAN_YEN, 0, {0, 15, 66, 64}} or
{CU_JAPAN_YEN, 0, {0, 0x0F, 0x42, 0x40}}

Details:

Standard: *yes*
Resource Set: *Standard 00:00:00:00:00:00:00:00-0*
Index: *89*
Obsolete: *no*
Size: *6*
Programmatic Name: *SNVT_currency*

Neuron C Type:

Structure	
currency	Currency (currency names) .
	<i>currency_t</i>
power_of_10	Magnitude (power of 10) .

	signed short Minimum: -128 Maximum: 127 Scaling (A,B,C): $1, 0, 0$ Scaled value: $1 * 10^0 * (Raw + 0)$ Resolution: 1
value	Value (currency value) . Credit is positive, debit is negative.
	s32_type Minimum: -2147483648 Maximum: 2147483647 Scaling (A,B,C): $1, 0, 0$ Scaled value: $1 * 10^0 * (Raw + 0)$ Resolution: 1

Formats: *SNVT_currency: text("%m,%d,E%d", currency, value, power_of_10)*

SNVT_currency#LO: text("%m|%d|E%d", currency, value, power_of_10)

SNVT_date_cal

Overview:

Date (year, month, day) . This SNVT is obsolete. Use SNVT_time_stamp instead.

This SNVT is obsolete. Use SNVT_time_stamp instead.

Details:

Standard:	<i>yes</i>
Resource Set:	<i>Standard 00:00:00:00:00:00:00:00-0</i>
Index:	<i>10</i>
Obsolete:	<i>yes</i>
Size:	<i>4</i>
Programmatic Name:	<i>SNVT_date_cal</i>
Neuron C Type:	

Structure	
year	Year (years) . Zero (0) means year not specified. Minus one (-1) represents NULL date.
	unsigned long Minimum: 0

	Maximum: <i>3000</i> Invalid: <i>65535</i> Scaling (A,B,C): <i>1, 0, 0</i> Scaled value: <i>$1 * 10^0 * (Raw + 0)$</i> Resolution: <i>1</i>
month	Month (months) .
	unsigned short Minimum: <i>0</i> Maximum: <i>12</i> Scaling (A,B,C): <i>1, 0, 0</i> Scaled value: <i>$1 * 10^0 * (Raw + 0)$</i> Resolution: <i>1</i>
day	Day (days) .
	unsigned short Minimum: <i>0</i> Maximum: <i>31</i> Scaling (A,B,C): <i>1, 0, 0</i> Scaled value: <i>$1 * 10^0 * (Raw + 0)$</i> Resolution: <i>1</i>

Formats: *SNVT_date_cal: text("%d/%d/%d", year, month, day)*

SNVT_date_cal#LO: text(date(year, month, day))

SNVT_date_day

Overview:

Day of week (day names) .

This is an enumerated list of the days of the week.

Details:

Standard: *yes*

Resource Set: *Standard 00:00:00:00:00:00:00:00-0*

Index: *11*

Obsolete: *no*

Size: *1*
 Programmatic Name: *SNVT_date_day*
 Neuron C Type: *days_of_week_t*
 Formats: *SNVT_date_day: text("%m")*

SNVT_date_event

Overview:

Date event. Reports the status of a schedule

This network variable type is used to report the status of a schedule.

Details:

Standard: *yes*
 Resource Set: *Standard 00:00:00:00:00:00:00:00-0*
 Index: *176*
 Obsolete: *no*
 Size: *26*
 Programmatic Name: *SNVT_date_event*
 Neuron C Type:

Structure	
days_to_active	Days to active (days) . Number of days until this schedule will be active. Positive if a schedule is inactive; zero or negative if a schedule is active
	signed long Minimum: <i>-32768</i> Maximum: <i>32767</i> Invalid: <i>32767</i> Scaling <i>1, 0, 0</i> (A,B,C): Scaled value: <i>$1 * 10^0 * (Raw + 0)$</i> Resolution: <i>1</i>
days_to_inactive	Days to inactive (days) . Number of days until this schedule will be inactive. Positive if a schedule is active; zero or negative if a schedule is inactive
	signed long Minimum: <i>-32768</i> Maximum: <i>32767</i> Invalid: <i>-32768</i> Scaling <i>1, 0, 0</i> (A,B,C):

	Scaled value: $1 * 10^0 * (Raw + 0)$ Resolution: 1
name	Schedule name (array of 22 characters) . Nul-terminated schedule name. The nul terminator is not required if the name is 22 characters.
	unsigned char [22] Minimum: 32 Maximum: 126 Scaling (A,B,C): 1, 0, 0 Scaled value: $1 * 10^0 * (Raw + 0)$ Resolution: 1

Formats: *SNVT_date_event: text("%d %d %s", days_to_active, days_to_inactive, name)*

Used by: *SFPTcalendar SFPTnodeObject*

SNVT_date_time

Overview:

Time of Day (hour, minute, second) . This SNVT is obsolete. Use SNVT_time_stamp instead.

Details:

Standard: *yes*

Resource Set: *Standard 00:00:00:00:00:00:00:00-0*

Index: *12*

Obsolete: *yes*

Size: *3*

Programmatic Name: *SNVT_date_time*

Neuron C Type:

Structure	
hour	Hour (hours) . This field uses a 24-hour value.
	unsigned short Minimum: 0 Maximum: 23 Scaling (A,B,C): 1, 0, 0 Scaled value: $1 * 10^0 * (Raw + 0)$

	Resolution: <i>1</i>
minute	Minute (minutes) .
	unsigned short Minimum: <i>0</i> Maximum: <i>59</i> Scaling <i>1, 0, 0</i> (A,B,C): Scaled <i>1 * 10⁰ *(Raw+0)</i> value: Resolution: <i>1</i>
second	Second (seconds) .
	unsigned short Minimum: <i>0</i> Maximum: <i>59</i> Scaling <i>1, 0, 0</i> (A,B,C): Scaled <i>1 * 10⁰ *(Raw+0)</i> value: Resolution: <i>1</i>

Formats: *SNVT_date_time: text("%d:%d:%d", hour, minute, second)*

SNVT_date_time#LO: text(time(hour, minute, second))

SNVT_defr_mode

Overview:

Defrost mode (defrost mode names) .

Details:

Standard: *yes*
 Resource Set: *Standard 00:00:00:00:00:00:00:00-0*
 Index: *120*
 Obsolete: *no*
 Size: *1*
 Programmatic Name: *SNVT_defr_mode*
 Neuron C Type: *defrost_mode_t*
 Formats: *SNVT_defr_mode: text("%m")*
 Used by: *SCPTdefrostMode*

SNVT_defr_state

Overview:

Defrost state (defrost state names) .

Details:

Standard:	<i>yes</i>
Resource Set:	<i>Standard 00:00:00:00:00:00:00:00-0</i>
Index:	<i>122</i>
Obsolete:	<i>no</i>
Size:	<i>1</i>
Programmatic Name:	<i>SNVT_defr_state</i>
Neuron C Type:	<i>defrost_state_t</i>
Formats:	<i>SNVT_defr_state: text("%m")</i>
Used by:	<i>SFPTrefrigDisplayCaseControllerDefrost</i> <i>SFPTrefrigDisplayCaseControllerEvaporator</i> <i>SFPTrefrigDisplayCaseControllerThermostat</i>

SNVT_defr_term

Overview:

Defrost termination (defrost termination names) .

Details:

Standard:	<i>yes</i>
Resource Set:	<i>Standard 00:00:00:00:00:00:00:00-0</i>
Index:	<i>121</i>
Obsolete:	<i>no</i>
Size:	<i>1</i>
Programmatic Name:	<i>SNVT_defr_term</i>
Neuron C Type:	<i>defrost_term_t</i>
Formats:	<i>SNVT_defr_term: text("%m")</i>
Used by:	<i>SCPTtermTimeTemp</i>

SNVT_density

Overview:

Density (kilograms/cubic meter) .

Details:

Standard:	<i>yes</i>
Resource Set:	<i>Standard 00:00:00:00:00:00:00:00-0</i>
Index:	<i>100</i>
Obsolete:	<i>no</i>
Size:	<i>2</i>
Programmatic Name:	<i>SNVT_density</i>
Neuron C Type:	unsigned long
	Minimum: <i>0</i>
	Maximum: <i>65535</i>
	Scaling (A,B,C): <i>5, -1, 0</i>
	Scaled value: <i>$5 * 10^{-1} * (Raw + 0)$</i>
	Resolution: <i>0.5</i>
Formats:	<i>SNVT_density#SI: text("%f", *1+0(0:880))</i>
	<i>SNVT_density#US: text("%f", *0.0624+0(0:865))</i>

SNVT_density_f

Overview:

Density (kilograms/cubic meter) .

Details:

Standard:	<i>yes</i>
Resource Set:	<i>Standard 00:00:00:00:00:00:00:00-0</i>
Index:	<i>101</i>
Obsolete:	<i>no</i>
Size:	<i>4</i>
Programmatic Name:	<i>SNVT_density_f</i>
Neuron C Type:	float
	Minimum: <i>0</i>
	Maximum: <i>$3.4028234663853E+038$</i>

Formats: *SNVT_density_f#SI: text("%f", *I+0(0:880))*

*SNVT_density_f#US: text("%f", *0.0624+0(0:865))*

SNVT_dev_c_mode

Overview:

Device control mode (device control mode names) .

A SNVT_dev_c_mode network variable is to used for heating, ventilation, and air-conditioning applications. This network variable defines and indicates the control mode of devices like pumps, fans and other actuator-based devices.

Details:

Standard: *yes*

Resource Set: *Standard 00:00:00:00:00:00:00:00-0*

Index: *162*

Obsolete: *no*

Size: *1*

Programmatic Name: *SNVT_dev_c_mode*

Neuron C Type: *device_c_mode_t*

Formats: *SNVT_dev_c_mode: text("%m")*

Used by: *SCPTactuatorCharacteristic SCPTcombFlowCharacteristic*
SCPTdeviceControlMode SCPTvalveFlowCharacteristic
SCPTvalveType SFPTpumpController

SNVT_dev_fault

Overview:

Device fault states. Fault information for the device

This network variable type is used to report fault information for a Pump Controller or Valve Position

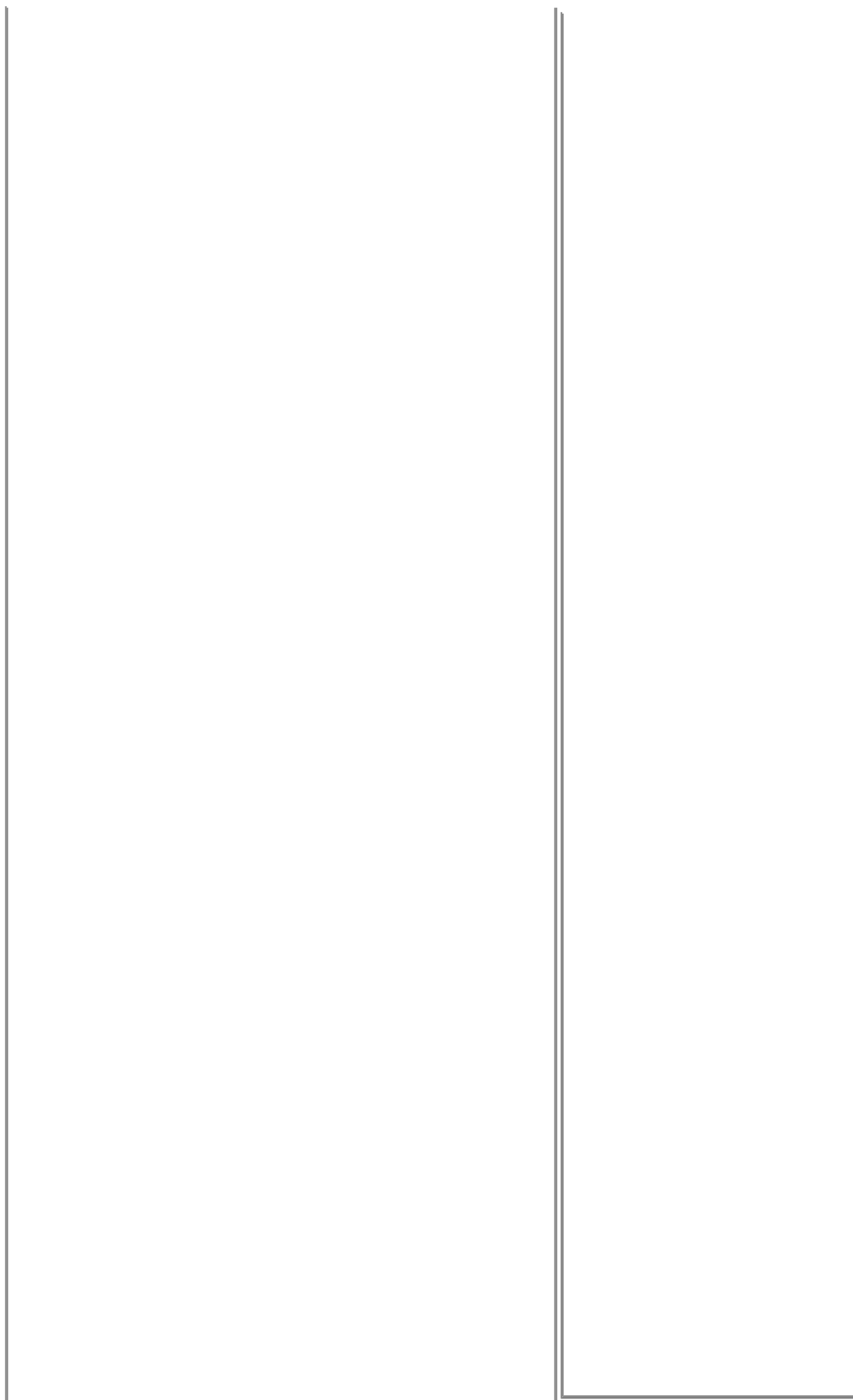
Details:

Standard: *yes*

Resource Set: *Standard 00:00:00:00:00:00:00:00-0*

Index: *174*

valve_pos



Formats:

```
SNVT_dev_fault: text(("DS=%m ",device_select),((device_select == 0) ?("SF: [VL=%d
dev_type.pump_ctrl.sf_voltage_low,dev_type.pump_ctrl.sf_voltage_high,dev_type.pum
SF: [VOR=%d ET=$d FR=%d {$d}] GF=%d",
dev_type.valve_pos.df_valve_blocked,dev_type.valve_pos.df_blocked_direction_open,
("UNKNOWN"))))
```

Used by: *SFPThvacValvePositioner* *SFPTpumpController*

SNVT_dev_maint

Overview:

Device maintenance. Device-maintenance states

This network variable type is used to report device-maintenance states of a Pump Controller or Valve

Details:

Standard: *yes*
Resource Set: *Standard 00:00:00:00:00:00:00:00-0*
Index: *175*
Obsolete: *no*
Size: *4*
Programmatic Name: *SNVT_dev_maint*

Neuron C
Type:

Structure	
device_select	Device selection (device sele
	<i>device_select_t</i>
dev_type	Union of device maintenance
	Union
	pump_ctrl

valve_pos



Formats: *SNVT_dev_maint: text(("DS=%m ",device_select),((device_select == 0) ? ("RQ=%d 1
((device_select == 1) ? ("MM=%d PC=%d EC=%d PC=%d LC=%d RC=%d BC=%d
d",dev_type.valve_pos.motor_maint,dev_type.valve_pos.packing_change,dev_type.val
("UNKNOWN"))))*

Used by: *SFPT_hvacValvePositioner SFPT_pumpController*

SNVT_dev_status

Overview:

Device status. Status of the device

This network variable type is used to report the status of a Pump Controller or Valve Positioner device

Details:

Standard: *yes*

Resource Set: *Standard 00:00:00:00:00:00:00:00-0*

Index: *173*

Obsolete: *no*

Size: *4*

Programmatic Name: *SNVT_dev_status*

Neuron C Type:	Structure	
	device_select	Device select
		<i>device_selec</i>
	dev_type	Union of dev
		Union pump_ctrl

valve_pos

Used by: *SFPThvacValvePositioner* *SFPThvacController*

Overview:

Details:

Standard: *yes*

Resource Set: *Standard 00:00:00:00:00:00:00:00-0*Index: *135*Obsolete: *no*Size: *11*Programmatic Name: *SNVT_earth_pos*

Neuron C Type:

Structure	
latitude_direction	Direction of latitude (direction (S/N)) . 0 = South latitude, 1 = North latitude
	bitfield Signed: <i>no</i> Width: <i>1</i> Offset: <i>0</i> Minimum: <i>0</i> Maximum: <i>1</i> Scaling <i>1, 0, 0</i> (A,B,C): Scaled <i>1 * 10⁰ *(Raw+0)</i> value: Resolution: <i>1</i>
longitude_direction	Direction of longitude (direction (E/W)) . 0 = East longitude, 1 = West longitude
	bitfield Signed: <i>no</i> Width: <i>1</i> Offset: <i>1</i> Minimum: <i>0</i> Maximum: <i>1</i> Scaling <i>1, 0, 0</i> (A,B,C): Scaled <i>1 * 10⁰ *(Raw+0)</i> value: Resolution: <i>1</i>
latitude_deg	Latitude degrees (degrees) .
	unsigned short Minimum: <i>0</i> Maximum: <i>90</i> Invalid: <i>255</i> Scaling <i>1, 0, 0</i> (A,B,C): Scaled <i>1 * 10⁰ *(Raw+0)</i> value: Resolution: <i>1</i>
latitude_min	Latitude minutes (minutes) .
	unsigned long Minimum: <i>0</i>

	Maximum: 59999 Invalid: 65535 Scaling 1, -3, 0 (A,B,C): Scaled value: $1 * 10^{-3} * (Raw + 0)$ Resolution: 0.001
longitude_deg	Longitude degrees (degrees) .
	unsigned short Minimum: 0 Maximum: 180 Invalid: 255 Scaling 1, 0, 0 (A,B,C): Scaled value: $1 * 10^0 * (Raw + 0)$ Resolution: 1
longitude_min	Longitude minutes (minutes) .
	unsigned long Minimum: 0 Maximum: 59999 Invalid: 65535 Scaling 1, -3, 0 (A,B,C): Scaled value: $1 * 10^{-3} * (Raw + 0)$ Resolution: 0.001
height_above_sea	Height above sea level (meters) .
	float Minimum: -3.4028234663853E+038 Maximum: 3.4028234663853E+038

Formats:

SNVT_earth_pos#SI: text(("%d %d ", latitude_direction, longitude_direction), (latitude_direction == 0) ? ("S") : ("N")), ("%d %d ", latitude_deg, latitude_min), (longitude_direction == 0) ? ("E") : ("W")), ("%d %d %f", longitude_deg, longitude_min, height_above_sea))

*SNVT_earth_pos#US: text(("%d %d ", latitude_direction, longitude_direction), (latitude_direction == 0) ? ("S") : ("N")), ("%d %d ", latitude_deg, latitude_min), (longitude_direction == 0) ? ("E") : ("W")), ("%d %d %f", longitude_deg, longitude_min, height_above_sea * 3.28084 + 0(0:840)))*

SNVT_elapsed_tm

Overview:

Elapsed time (day, hour, minute, second, millisecond) .

Details:

Standard: *yes*

Resource Set: *Standard 00:00:00:00:00:00:00:00-0*

Index: *87*

Obsolete: *no*

Size: *7*

Programmatic Name: *SNVT_elapsed_tm*

Neuron C Type:

Structure	
day	Days (days) . The value 65535 represents NULL or unknown elapsed time.
	unsigned long Minimum: <i>0</i> Maximum: <i>65535</i> Invalid: <i>65535</i> Scaling (A,B,C): <i>1, 0, 0</i> Scaled value: <i>$1 * 10^0 * (Raw + 0)$</i> Resolution: <i>1</i>
hour	Hours (hours) . This field uses a 24-hour value.
	unsigned short Minimum: <i>0</i> Maximum: <i>23</i> Scaling (A,B,C): <i>1, 0, 0</i> Scaled value: <i>$1 * 10^0 * (Raw + 0)$</i> Resolution: <i>1</i>
minute	Minutes (minutes) .
	unsigned short Minimum: <i>0</i> Maximum: <i>59</i> Scaling (A,B,C): <i>1, 0, 0</i> Scaled value: <i>$1 * 10^0 * (Raw + 0)$</i> Resolution: <i>1</i>
second	Seconds (seconds) .
	unsigned short Minimum: <i>0</i>

	Maximum: 59 Scaling 1, 0, 0 (A,B,C): Scaled value: $1 * 10^0 * (Raw + 0)$ Resolution: 1
millisecond	Milliseconds (milliseconds) .
	unsigned long Minimum: 0 Maximum: 999 Scaling 1, 0, 0 (A,B,C): Scaled value: $1 * 10^0 * (Raw + 0)$ Resolution: 1

Formats: *SNVT_elapsed_tm: text("%d %d:%d:%d:%d", day, hour, minute, second, millisecond)*

SNVT_elapsed_tm#LO: text("%d ", day), time(hour, minute, second, millisecond))

Used by: *SCPTactFbDly SCPTalrmClrT1 SCPTalrmClrT2 SCPTalrmIhbT
SCPTalrmSetT1 SCPTalrmSetT2 SCPTdriveT SCPTinFbDly
SCPTmaxRcvT SCPTmaxSndT SCPTminSndT SCPTmonInterval
SCPToffDely SCPTpwmPeriod SCPTrunHrAlarm SCPTrunHrInit
SCPTrunTimeAlarm SCPTscanTime SFPTchannelMonitor
SFPTthvacValvePositioner SFPTisiKeypad SFPTisiLampActuator
SFPTisiSunblindActuator SFPTlampActuator SFPTnodeObject
SFPTutilityDataLoggerRegister*

SNVT_elec_kwh

Overview:

Electric energy (kiloWatt-hours) .

For new designs, SNVT_elec_kwh_1 should be used instead of SNVT_elec_kwh.

Details:

Standard: *yes*
Resource Set: *Standard 00:00:00:00:00:00:00:00-0*
Index: *13*
Obsolete: *no*
Size: *2*
Programmatic Name: *SNVT_elec_kwh*

Neuron C Type:	unsigned long
Minimum:	0
Maximum:	65535
Scaling (A,B,C):	1, 0, 0
Scaled value:	$1 * 10^0 * (Raw + 0)$
Resolution:	1
Formats:	<i>SNVT_elec_kwh: text("%d")</i>
Used by:	<i>SCPTenergyCntInit SFPTisiLampActuator SFPTlampActuator SFPTpumpController</i>

SNVT_elec_kwh_l

Overview:

Electric energy (kiloWatt-hours) .

Used to communicate electricity-metering data to data-logging or energy management devices.

For new designs, SNVT_elec_kwh_l should be used instead of SNVT_elec_kwh.

Details:

Standard:	<i>yes</i>
Resource Set:	<i>Standard 00:00:00:00:00:00:00:00-0</i>
Index:	<i>146</i>
Obsolete:	<i>no</i>
Size:	<i>4</i>
Programmatic Name:	<i>SNVT_elec_kwh_l</i>
Neuron C Type:	s32_type
Minimum:	-2147483648
Maximum:	2147483647
Invalid:	2147483647
Scaling (A,B,C):	1, -1, 0
Scaled value:	$1 * 10^{-1} * (Raw + 0)$
Resolution:	0.1
Formats:	<i>SNVT_elec_kwh_l: text("%f")</i>

SNVT_elec_whr

Overview:

Electric energy (Watt-hours) .

Details:

Standard: *yes*
 Resource Set: *Standard 00:00:00:00:00:00:00:00-0*
 Index: *14*
 Obsolete: *no*
 Size: *2*
 Programmatic Name: *SNVT_elec_whr*
 Neuron C Type: unsigned long
 Minimum: *0*
 Maximum: *65535*
 Scaling (A,B,C): *1, -1, 0*
 Scaled value: *$1 * 10^{-1} * (Raw + 0)$*
 Resolution: *0.1*
 Formats: *SNVT_elec_whr: text("%f")*

SNVT_elec_whr_f

Overview:

Electric energy (Watt-hours) .

Details:

Standard: *yes*
 Resource Set: *Standard 00:00:00:00:00:00:00:00-0*
 Index: *68*
 Obsolete: *no*
 Size: *4*
 Programmatic Name: *SNVT_elec_whr_f*
 Neuron C Type: float
 Minimum: *0*
 Maximum: *3.4028234663853E+038*
 Formats: *SNVT_elec_whr_f: text("%f")*
 Used by: *SFPTgeneratorSet*

SNVT_ent_opmode

Overview:

Entry operation mode (entry operation mode names) . used to send operation-mode information to an entry object, e.g., a door, lock, sluice, or something which allows/prohibits entry to an area

Used to send operation-mode information to an entry object such as a door, lock, sluice, or something that allows or prohibits entry to an area.

Details:

Standard:	<i>yes</i>
Resource Set:	<i>Standard 00:00:00:00:00:00:00:00-0</i>
Index:	<i>168</i>
Obsolete:	<i>no</i>
Size:	<i>1</i>
Programmatic Name:	<i>SNVT_ent_opmode</i>
Neuron C Type:	<i>ent_opmode_cmd_t</i>
Formats:	<i>SNVT_ent_opmode: text("%m")</i>
Used by:	<i>SFPTentryExit</i>

SNVT_ent_state

Overview:

Entry state (entry state names) . Desired state for an entry object, e.g., a door, lock, sluice, or something that controls entry of an area

Desired state for an entry object such as a door, lock, sluice, or something that controls entry of an area.

Details:

Standard:	<i>yes</i>
Resource Set:	<i>Standard 00:00:00:00:00:00:00:00-0</i>
Index:	<i>169</i>
Obsolete:	<i>no</i>
Size:	<i>1</i>
Programmatic Name:	<i>SNVT_ent_state</i>
Neuron C Type:	<i>ent_cmd_t</i>
Formats:	<i>SNVT_ent_state: text("%m")</i>

Used by: *SFPTentryExit*

SNVT_ent_status

Overview:

Entry status. Status information from an entry object, e.g., a door, lock, sluice, or something that allows/prohibits entry into an area

Status information from an entry object such as a door, lock, sluice, or something that allows or prohibits entry into an area.

The last five bits (offset 3, after unknown_state field) are undefined.

Details:

Standard: *yes*

Resource Set: *Standard 00:00:00:00:00:00:00:00-0*

Index: *170*

Obsolete: *no*

Size: *5*

Programmatic Name: *SNVT_ent_status*

Neuron C Type:

Structure	
unlocked	Unlocked device (boolean) . Device is in unlocked position
	bitfield Signed: <i>no</i> Width: <i>1</i> Offset: <i>0</i> Minimum: <i>0</i> Maximum: <i>1</i> Scaling (A,B,C): <i>1, 0, 0</i> Scaled value: <i>$1 * 10^0 * (Raw + 0)$</i> Resolution: <i>1</i>
locked	Locked device (boolean) . Device is in locked position
	bitfield Signed: <i>no</i> Width: <i>1</i> Offset: <i>1</i> Minimum: <i>0</i>

	Maximum: 1 Scaling $1, 0, 0$ (A,B,C): Scaled $1 * 10^0 * (Raw + 0)$ value: Resolution: 1
security_locked	Security locked (boolean) . Device is in a security-driven locked position
	bitfield Signed: <i>no</i> Width: 1 Offset: 2 Minimum: 0 Maximum: 1 Scaling $1, 0, 0$ (A,B,C): Scaled $1 * 10^0 * (Raw + 0)$ value: Resolution: 1
closed	Closed device (boolean) . Device is in a closed position
	bitfield Signed: <i>no</i> Width: 1 Offset: 3 Minimum: 0 Maximum: 1 Scaling $1, 0, 0$ (A,B,C): Scaled $1 * 10^0 * (Raw + 0)$ value: Resolution: 1
open	Open device (boolean) . Device is in an open position
	bitfield Signed: <i>no</i> Width: 1 Offset: 4 Minimum: 0 Maximum: 1 Scaling $1, 0, 0$ (A,B,C): Scaled $1 * 10^0 * (Raw + 0)$ value: Resolution: 1
in_alarm	In alarm state (boolean) . The device is in the alarm

	state.
	bitfield Signed: <i>no</i> Width: <i>1</i> Offset: <i>5</i> Minimum: <i>0</i> Maximum: <i>1</i> Scaling <i>1, 0, 0</i> (A,B,C): Scaled value: $1 * 10^0 * (Raw + 0)$ Resolution: <i>1</i>
in_error_cond	In error condition (boolean) . Device has an error condition
	bitfield Signed: <i>no</i> Width: <i>1</i> Offset: <i>6</i> Minimum: <i>0</i> Maximum: <i>1</i> Scaling <i>1, 0, 0</i> (A,B,C): Scaled value: $1 * 10^0 * (Raw + 0)$ Resolution: <i>1</i>
open_pre_alarm	Open device, pre-alarm (boolean) . Device is open, and in warning state
	bitfield Signed: <i>no</i> Width: <i>1</i> Offset: <i>7</i> Minimum: <i>0</i> Maximum: <i>1</i> Scaling <i>1, 0, 0</i> (A,B,C): Scaled value: $1 * 10^0 * (Raw + 0)$ Resolution: <i>1</i>
open_alarm	Open Device, alarm state (boolean) . Device is open, and in not-closed alarm state
	bitfield Signed: <i>no</i> Width: <i>1</i> Offset: <i>0</i> Minimum: <i>0</i> Maximum: <i>1</i>

	Scaling $1, 0, 0$ (A,B,C): Scaled $1 * 10^0 * (Raw + 0)$ value: Resolution: 1
service_alarm	Service alarm (boolean) . Device needs service
	bitfield Signed: <i>no</i> Width: 1 Offset: 1 Minimum: 0 Maximum: 1 Scaling $1, 0, 0$ (A,B,C): Scaled $1 * 10^0 * (Raw + 0)$ value: Resolution: 1
tamper	Tamper mode (boolean) . Device has detected tamper
	bitfield Signed: <i>no</i> Width: 1 Offset: 2 Minimum: 0 Maximum: 1 Scaling $1, 0, 0$ (A,B,C): Scaled $1 * 10^0 * (Raw + 0)$ value: Resolution: 1
entry_req	Entry request pending (boolean) . Device has a pending entry request
	bitfield Signed: <i>no</i> Width: 1 Offset: 3 Minimum: 0 Maximum: 1 Scaling $1, 0, 0$ (A,B,C): Scaled $1 * 10^0 * (Raw + 0)$ value: Resolution: 1
exit_req	Exit request pending (boolean) . Device has a pending exit request
	bitfield

	Signed: <i>no</i> Width: <i>1</i> Offset: <i>4</i> Minimum: <i>0</i> Maximum: <i>1</i> Scaling (A,B,C): <i>1, 0, 0</i> Scaled value: $1 * 10^0 * (Raw + 0)$ Resolution: <i>1</i>
key_req	Key request pending (boolean) . Device has a pending key request
	bitfield Signed: <i>no</i> Width: <i>1</i> Offset: <i>5</i> Minimum: <i>0</i> Maximum: <i>1</i> Scaling (A,B,C): <i>1, 0, 0</i> Scaled value: $1 * 10^0 * (Raw + 0)$ Resolution: <i>1</i>
safety_ext_req	Safety-exit request pending (boolean) . Device has a pending safety-exit request
	bitfield Signed: <i>no</i> Width: <i>1</i> Offset: <i>6</i> Minimum: <i>0</i> Maximum: <i>1</i> Scaling (A,B,C): <i>1, 0, 0</i> Scaled value: $1 * 10^0 * (Raw + 0)$ Resolution: <i>1</i>
emergency_req	Emergency-exit request pending (boolean) . Device has a pending emergency-exit request
	bitfield Signed: <i>no</i> Width: <i>1</i> Offset: <i>7</i> Minimum: <i>0</i> Maximum: <i>1</i> Scaling (A,B,C): <i>1, 0, 0</i>

	Scaled value: $1 * 10^0 * (Raw + 0)$ Resolution: 1
unable_lock	Unable to lock (boolean) . Device is unable to close and/or lock
	bitfield Signed: no Width: 1 Offset: 0 Minimum: 0 Maximum: 1 Scaling (A,B,C): 1, 0, 0 Scaled value: $1 * 10^0 * (Raw + 0)$ Resolution: 1
unable_unlock	Unable to unlock (boolean) . Device is unable to open and/or unlock
	bitfield Signed: no Width: 1 Offset: 1 Minimum: 0 Maximum: 1 Scaling (A,B,C): 1, 0, 0 Scaled value: $1 * 10^0 * (Raw + 0)$ Resolution: 1
stuck	Device is stuck (boolean) . Device is unable to move
	bitfield Signed: no Width: 1 Offset: 2 Minimum: 0 Maximum: 1 Scaling (A,B,C): 1, 0, 0 Scaled value: $1 * 10^0 * (Raw + 0)$ Resolution: 1
forced_open	Forced-open Device (boolean) . Device is/was forced to go to an open position
	bitfield Signed: no

	Width: <i>1</i> Offset: <i>3</i> Minimum: <i>0</i> Maximum: <i>1</i> Scaling <i>1, 0, 0</i> (A,B,C): Scaled value: $1 * 10^0 * (Raw + 0)$ Resolution: <i>1</i>
forced_close	Forced-closed Device (boolean) . Device is/was forced to go to a closed position
	bitfield Signed: <i>no</i> Width: <i>1</i> Offset: <i>4</i> Minimum: <i>0</i> Maximum: <i>1</i> Scaling <i>1, 0, 0</i> (A,B,C): Scaled value: $1 * 10^0 * (Raw + 0)$ Resolution: <i>1</i>
opening	Device is opening (boolean) . Device is currently opening from a closed position
	bitfield Signed: <i>no</i> Width: <i>1</i> Offset: <i>5</i> Minimum: <i>0</i> Maximum: <i>1</i> Scaling <i>1, 0, 0</i> (A,B,C): Scaled value: $1 * 10^0 * (Raw + 0)$ Resolution: <i>1</i>
closing	Device is closing (boolean) . Device is currently closing from an open position
	bitfield Signed: <i>no</i> Width: <i>1</i> Offset: <i>6</i> Minimum: <i>0</i> Maximum: <i>1</i> Scaling <i>1, 0, 0</i> (A,B,C): Scaled value: $1 * 10^0 * (Raw + 0)$

	value: Resolution: <i>1</i>
moving	Device is in motion (boolean) . Device is currently changing position
	bitfield Signed: <i>no</i> Width: <i>1</i> Offset: <i>7</i> Minimum: <i>0</i> Maximum: <i>1</i> Scaling <i>1, 0, 0</i> (A,B,C): Scaled <i>1 * 10⁰ *(Raw+0)</i> value: Resolution: <i>1</i>
stopped	Device Stopped (boolean) . The device is stopped and can be moved manually
	bitfield Signed: <i>no</i> Width: <i>1</i> Offset: <i>0</i> Minimum: <i>0</i> Maximum: <i>1</i> Scaling <i>1, 0, 0</i> (A,B,C): Scaled <i>1 * 10⁰ *(Raw+0)</i> value: Resolution: <i>1</i>
safety_alarm	Safety-alarm (boolean) . Device is in a safety-alarm state
	bitfield Signed: <i>no</i> Width: <i>1</i> Offset: <i>1</i> Minimum: <i>0</i> Maximum: <i>1</i> Scaling <i>1, 0, 0</i> (A,B,C): Scaled <i>1 * 10⁰ *(Raw+0)</i> value: Resolution: <i>1</i>
unknown_state	Unknown state (boolean) . The state of the device is currently unknown
	bitfield Signed: <i>no</i> Width: <i>1</i>

Formats:

```
SNVT_ent_status: text("%d %d %d %d %d %d %d %d %d %d %d %d %d %d %d %d %d %d %d %d %d %d %m", unlocked, locked, security_locked, closed, open, in_alarm, in_error_cond, open_pre_alarm, open_alarm, service_alarm, tamper, entry_req, exit_req, key_req, safety_ext_req, emergency_req, unable_lock, unable_unlock, stuck, forced_open, forced_close, opening, closing, moving, stopped, safety_alarm, unknown_state, cmd_fb)
```

Used by: *SFPTentryExit*

SNVT_enthalpy

Overview:

Enthalpy (kiloJoules/kg) .

A quantity of heat per unit of mass, a SNVT_enthalpy network variable may be used to communicate enthalpy values in air conditioning and other process applications.

EXAMPLES

The enthalpy of the outdoor air is transmitted by an enthalpy sensor as 54 kJ/kg. An air-handling unit, with an economizer controller, transmits the difference in enthalpy between indoor and outdoor air as -17 kJ/kg.

Details:

Standard:	<i>yes</i>
Resource Set:	<i>Standard 00:00:00:00:00:00:00:00-0</i>
Index:	<i>153</i>
Obsolete:	<i>no</i>
Size:	<i>2</i>
Programmatic Name:	<i>SNVT_enthalpy</i>
Neuron C Type:	<i>signed long</i>

Minimum: -32768
 Maximum: 32767
 Invalid: 32767
 Scaling (A,B,C): 1, -2, 0
 Scaled value: $1 * 10^{-2} * (Raw + 0)$
 Resolution: 0.01

Formats: *SNVT_enthalpy#SI: text("%f", *1+0(0:1409))*

*SNVT_enthalpy#US: text("%f", *0.429923+0(0:1400))*

Used by: *SCPToutdoorAirEnthalpySetpoint SFPTdischargeAirController*

SNVT_evap_state

Overview:

Evaporator state (evaporator state names) .

Details:

Standard: *yes*
 Resource Set: *Standard 00:00:00:00:00:00:00:00-0*
 Index: *118*
 Obsolete: *no*
 Size: *1*
 Programmatic Name: *SNVT_evap_state*
 Neuron C Type: *evap_t*
 Formats: *SNVT_evap_state: text("%m")*
 Used by: *SFPTrefrigDisplayCaseControllerEvaporator*

SNVT_ex_control

Overview:

Exclusive control (status, address) .

This SNVT contains a definition of what device has exclusive control of an item.

Each time this SNVT is used, the documentation must specify to what item the exclusive control applies. Any functional block using this SNVT should provide a timeout or other means to prevent lockout of the item.

If the value of control_status is unknown or not applicable, it will have the value EX_CONTROL_NUL. If control_status is not EX_CONTROL_THIS_ADDR then control_device_addr does not contain valid data.

Details:

Standard: *yes*

Resource Set: *Standard 00:00:00:00:00:00:00:00-0*

Index: *157*

Obsolete: *no*

Size: *10*

Programmatic Name: *SNVT_ex_control*

Neuron C Type:

Structure	
control_status	Control type (control type names) .
	<i>ex_control_t</i>
control_device_addr	Control device address (LonWorks subnet-node address) .
Structure	
domain_id	Domain ID (array of 6 bytes) . ANSI/CEA-709.1 domain ID
	unsigned short [6] Minimum: 0 Maximum: 255 Scaling 1, 0, 0 (A,B,C): Scaled $1 * 10^0 * (Raw + 0)$ value: Resolution: 1
domain_length	Domain length (ANSI/CEA-709.1 domain length) . Valid domain lengths are 0, 1, 3, and 6.
	unsigned short Minimum: 0 Maximum: 6 Scaling 1, 0, 0 (A,B,C): Scaled $1 * 10^0 * (Raw + 0)$ value: Resolution: 1
subnet	Subnet (subnet number) . There can be 255 subnets (1-255) in a domain.
	unsigned short Minimum: 0 Maximum: 255

		Invalid: 0 Scaling (A,B,C): $1, 0, 0$ Scaled value: $1 * 10^0 * (Raw + 0)$ Resolution: 1
	node	Node (node number) . There can be 127 nodes (1-127) in a subnet.
		unsigned short Minimum: 0 Maximum: 127 Invalid: 0 Scaling (A,B,C): $1, 0, 0$ Scaled value: $1 * 10^0 * (Raw + 0)$ Resolution: 1

Formats:

*SNVT_ex_control: text("%m %d <%x %x %x %x %x %x> %d %d",
control_status, control_device_addr.domain_length,
control_device_addr.domain_id[0], control_device_addr.domain_id[1],
control_device_addr.domain_id[2], control_device_addr.domain_id[3],
control_device_addr.domain_id[4], control_device_addr.domain_id[5],
control_device_addr.subnet, control_device_addr.node)*

SNVT_file_pos

Overview:

File position (pointer, length) .

Used to control the position of the read/write pointer in a file used for random access, as well as to specify the length of the next file transfer.

Details:

Standard: *yes*
Resource Set: *Standard 00:00:00:00:00:00:00:00-0*
Index: *90*
Obsolete: *no*
Size: *6*
Programmatic Name: *SNVT_file_pos*
Neuron C Type:

Structure	
rw_ptr	Read/Write pointer (file byte address) .

	s32_type Minimum: 0 Maximum: 2147483647 Scaling 1, 0, 0 (A,B,C): Scaled $1 * 10^0 * (Raw+0)$ value: Resolution: 1
rw_length	Read/Write length (number of bytes) .
	unsigned long Minimum: 0 Maximum: 65535 Scaling 1, 0, 0 (A,B,C): Scaled $1 * 10^0 * (Raw+0)$ value: Resolution: 1

Formats: *SNVT_file_pos: text("%d %d",rw_ptr, rw_length)*

Used by: *SFPTnodeObject*

SNVT_file_req

Overview:

File request.

The sn and gp structures are compatible with the snode_struct and group_struct structures defined in ADDRDEFS.H.

Details:

Standard: *yes*
Resource Set: *Standard 00:00:00:00:00:00:00:00-0*
Index: *73*
Obsolete: *no*
Size: *12*
Programmatic Name: *SNVT_file_req*

Neuron C Type:

Structure	
request	Request (file request names) .
	<i>file_request_t</i>
index	Index (file index) .

	unsigned long Minimum: 0 Maximum: 65535 Scaling 1, 0, 0 (A,B,C): Scaled value: $1 * 10^0 * (Raw + 0)$ Resolution: 1																
receive_timeout	Receive timeout (milliseconds) .																
	unsigned long Minimum: 0 Maximum: 65535 Scaling 1, 0, 0 (A,B,C): Scaled value: $1 * 10^0 * (Raw + 0)$ Resolution: 1																
dest_address	Destination address (LonWorks address) .																
	<table border="1"> <tr> <th colspan="2">Union</th></tr> <tr> <td>addrt</td><td>Address table entry (Address table entry) . ANSI/CEA-709.1 address in device's internal address table entry</td></tr> <tr> <td></td><td> <table border="1"> <tr> <th colspan="2">Structure</th></tr> <tr> <td>type</td><td>Address type (8-bit unsigned value) . The address-table address type is 33 (0x21).</td></tr> <tr> <td></td><td> unsigned short Minimum: 33 Maximum: 33 Scaling 1, 0, 0 (A,B,C): Scaled value: $1 * 10^0 * (Raw + 0)$ Resolution: 1 </td></tr> <tr> <td>index</td><td>Address table index (16-bit unsigned value) .</td></tr> <tr> <td></td><td> unsigned long Minimum: 0 Maximum: 65535 Scaling 1, 0, 0 (A,B,C): Scaled value: $1 * 10^0 * (Raw + 0)$ Resolution: 1 </td></tr> </table> </td></tr> </table>	Union		addrt	Address table entry (Address table entry) . ANSI/CEA-709.1 address in device's internal address table entry		<table border="1"> <tr> <th colspan="2">Structure</th></tr> <tr> <td>type</td><td>Address type (8-bit unsigned value) . The address-table address type is 33 (0x21).</td></tr> <tr> <td></td><td> unsigned short Minimum: 33 Maximum: 33 Scaling 1, 0, 0 (A,B,C): Scaled value: $1 * 10^0 * (Raw + 0)$ Resolution: 1 </td></tr> <tr> <td>index</td><td>Address table index (16-bit unsigned value) .</td></tr> <tr> <td></td><td> unsigned long Minimum: 0 Maximum: 65535 Scaling 1, 0, 0 (A,B,C): Scaled value: $1 * 10^0 * (Raw + 0)$ Resolution: 1 </td></tr> </table>	Structure		type	Address type (8-bit unsigned value) . The address-table address type is 33 (0x21).		unsigned short Minimum: 33 Maximum: 33 Scaling 1, 0, 0 (A,B,C): Scaled value: $1 * 10^0 * (Raw + 0)$ Resolution: 1	index	Address table index (16-bit unsigned value) .		unsigned long Minimum: 0 Maximum: 65535 Scaling 1, 0, 0 (A,B,C): Scaled value: $1 * 10^0 * (Raw + 0)$ Resolution: 1
Union																	
addrt	Address table entry (Address table entry) . ANSI/CEA-709.1 address in device's internal address table entry																
	<table border="1"> <tr> <th colspan="2">Structure</th></tr> <tr> <td>type</td><td>Address type (8-bit unsigned value) . The address-table address type is 33 (0x21).</td></tr> <tr> <td></td><td> unsigned short Minimum: 33 Maximum: 33 Scaling 1, 0, 0 (A,B,C): Scaled value: $1 * 10^0 * (Raw + 0)$ Resolution: 1 </td></tr> <tr> <td>index</td><td>Address table index (16-bit unsigned value) .</td></tr> <tr> <td></td><td> unsigned long Minimum: 0 Maximum: 65535 Scaling 1, 0, 0 (A,B,C): Scaled value: $1 * 10^0 * (Raw + 0)$ Resolution: 1 </td></tr> </table>	Structure		type	Address type (8-bit unsigned value) . The address-table address type is 33 (0x21).		unsigned short Minimum: 33 Maximum: 33 Scaling 1, 0, 0 (A,B,C): Scaled value: $1 * 10^0 * (Raw + 0)$ Resolution: 1	index	Address table index (16-bit unsigned value) .		unsigned long Minimum: 0 Maximum: 65535 Scaling 1, 0, 0 (A,B,C): Scaled value: $1 * 10^0 * (Raw + 0)$ Resolution: 1						
Structure																	
type	Address type (8-bit unsigned value) . The address-table address type is 33 (0x21).																
	unsigned short Minimum: 33 Maximum: 33 Scaling 1, 0, 0 (A,B,C): Scaled value: $1 * 10^0 * (Raw + 0)$ Resolution: 1																
index	Address table index (16-bit unsigned value) .																
	unsigned long Minimum: 0 Maximum: 65535 Scaling 1, 0, 0 (A,B,C): Scaled value: $1 * 10^0 * (Raw + 0)$ Resolution: 1																

sn	Subnet-node address (LonWorks subnet-node address) .	
	Structure	
	type	Address type (8-bit unsigned value) . The subnet-node address type is 1.
		unsigned short Minimum: 1 Maximum: 1 Scaling 1, 0, 0 (A,B,C): Scaled $1 * 10^0 *$ value: $(Raw+0)$ Resolution: 1
	domain	Domain (LonWorks domain index) .
		bitfield Signed: no Width: 1 Offset: 0 Minimum: 0 Maximum: 1 Scaling 1, 0, 0 (A,B,C): Scaled $1 * 10^0 *$ value: $(Raw+0)$ Resolution: 1
	node	Node (node number) . There can be 127 nodes (1-127) in a subnet.
		bitfield Signed: no Width: 7 Offset: 1 Minimum: 0 Maximum: 127 Scaling 1, 0, 0 (A,B,C): Scaled $1 * 10^0 *$ value: $(Raw+0)$ Resolution: 1
	retry	Retry count (number of retries) .
		bitfield

		Signed: <i>no</i> Width: <i>4</i> Offset: <i>4</i> Minimum: <i>0</i> Maximum: <i>15</i> Scaling <i>1, 0, 0</i> (A,B,C): Scaled <i>1 * 10⁰ *</i> value: <i>(Raw+0)</i> Resolution: <i>1</i>
	tx_timer	Transaction timer (timer code value) .
		bitfield Signed: <i>no</i> Width: <i>4</i> Offset: <i>4</i> Minimum: <i>0</i> Maximum: <i>15</i> Scaling <i>1, 0, 0</i> (A,B,C): Scaled <i>1 * 10⁰ *</i> value: <i>(Raw+0)</i> Resolution: <i>1</i>
	subnet	Subnet (subnet number) . There can be 255 subnets (1-255) in a domain.
		unsigned short Minimum: <i>0</i> Maximum: <i>255</i> Invalid: <i>0</i> Scaling <i>1, 0, 0</i> (A,B,C): Scaled <i>1 * 10⁰ *</i> value: <i>(Raw+0)</i> Resolution: <i>1</i>
gp		Group address (LonWorks group address) .
		Structure
	type	Address type (boolean) . The group address type is 1.
		bitfield Signed: <i>no</i> Width: <i>1</i> Offset: <i>0</i>

	Minimum: <i>0</i> Maximum: <i>1</i> Scaling <i>1, 0, 0</i> (A,B,C): Scaled <i>1 * 10⁰ *</i> value: <i>(Raw+0)</i> Resolution: <i>1</i>
size	Size (LonWorks group size) . An acknowledged group can have from 0-64 addressees, plus the sender.
	bitfield Signed: <i>no</i> Width: <i>7</i> Offset: <i>1</i> Minimum: <i>0</i> Maximum: <i>65</i> Scaling <i>1, 0, 0</i> (A,B,C): Scaled <i>1 * 10⁰ *</i> value: <i>(Raw+0)</i> Resolution: <i>1</i>
domain	Domain (LonWorks domain index) .
	bitfield Signed: <i>no</i> Width: <i>1</i> Offset: <i>0</i> Minimum: <i>0</i> Maximum: <i>1</i> Scaling <i>1, 0, 0</i> (A,B,C): Scaled <i>1 * 10⁰ *</i> value: <i>(Raw+0)</i> Resolution: <i>1</i>
unused	Unused field. This field is reserved.
	bitfield Signed: <i>no</i> Width: <i>7</i> Offset: <i>1</i> Minimum: <i>0</i> Maximum: <i>0</i> Scaling <i>1, 0, 0</i> (A,B,C):

			<p>Scaled value: $1 * 10^0 * (Raw+0)$</p> <p>Resolution: 1</p>
		retry	<p>Retry count (number of retries) .</p>
			<p>bitfield</p> <p>Signed: no</p> <p>Width: 4</p> <p>Offset: 4</p> <p>Minimum: 0</p> <p>Maximum: 15</p> <p>Scaling (A,B,C): 1, 0, 0</p> <p>Scaled value: $1 * 10^0 * (Raw+0)$</p> <p>Resolution: 1</p>
		tx_timer	<p>Transaction timer (timer code value) .</p>
			<p>bitfield</p> <p>Signed: no</p> <p>Width: 4</p> <p>Offset: 4</p> <p>Minimum: 0</p> <p>Maximum: 15</p> <p>Scaling (A,B,C): 1, 0, 0</p> <p>Scaled value: $1 * 10^0 * (Raw+0)$</p> <p>Resolution: 1</p>
		group	<p>Group. There can be 256 groups (0-255) in a domain.</p>
			<p>unsigned short</p> <p>Minimum: 0</p> <p>Maximum: 255</p> <p>Scaling (A,B,C): 1, 0, 0</p> <p>Scaled value: $1 * 10^0 * (Raw+0)$</p> <p>Resolution: 1</p>
auth_on	<p>Authentication on (boolean) . This field specifies whether the message requires authentication.</p>		
	<p>unsigned short</p> <p>Minimum: 0</p> <p>Maximum: 1</p>		

	Scaling <i>1, 0, 0</i> (A,B,C): Scaled <i>1 * 10⁰ *(Raw+0)</i> value: Resolution: <i>1</i>
prio_on	Priority on (boolean) . This field specifies whether the message is to be sent with priority.
	unsigned short Minimum: <i>0</i> Maximum: <i>1</i> Scaling <i>1, 0, 0</i> (A,B,C): Scaled <i>1 * 10⁰ *(Raw+0)</i> value: Resolution: <i>1</i>

Formats:

SNVT_file_req: text(("%m,%d %d %d %d %d %d %d %d %d %d ",request, index, receive_timeout, dest_address.sn.domain,dest_address.sn.retry, dest_address.sn.tx_timer,auth_on, prio_on, dest_address.gp.type), ((dest_address.gp.type == 1) ?("GP %d", dest_address.gp.group) : ((dest_address.sn.type == 1) ?("S/N %d/%d",dest_address.sn.subnet, dest_address.sn.node) :("BCAST or ADDRT or UNBOUND"))))

SNVT_file_req#LO: text(("%m\%d %d %d %d %d %d %d %d %d %d ",request, index, receive_timeout, dest_address.sn.domain,dest_address.sn.retry, dest_address.sn.tx_timer,auth_on, prio_on, dest_address.gp.type), ((dest_address.gp.type == 1) ?("GP %d", dest_address.gp.group) : ((dest_address.sn.type == 1) ?("S/N %d/%d",dest_address.sn.subnet, dest_address.sn.node) :("BCAST or ADDRT or UNBOUND"))))

Used by:

SFPTnodeObject

SNVT_file_status

Overview:

File status.

Part of the LONWORKS File-Transfer Protocol (LW-FTP).

Details:

Standard:	<i>yes</i>
Resource Set:	<i>Standard 00:00:00:00:00:00:00:00-0</i>
Index:	<i>74</i>
Obsolete:	<i>no</i>
Size:	<i>27</i>

Programmatic Name: *SNVT_file_status*

Neuron C Type:

Structure																	
status	Status (file status names) .																
	<i>file_status_t</i>																
number_of_files	Number of files (count) .																
	unsigned long Minimum: 0 Maximum: 65535 Scaling 1, 0, 0 (A,B,C): Scaled $1 * 10^0 * (Raw+0)$ value: Resolution: 1																
selected_file	Selected file (file index) .																
	unsigned long Minimum: 0 Maximum: 65535 Scaling 1, 0, 0 (A,B,C): Scaled $1 * 10^0 * (Raw+0)$ value: Resolution: 1																
adr	Address.																
	<table> <tr> <th colspan="2">Union</th></tr> <tr> <th>descriptor</th><th>Descriptor.</th></tr> <tr> <td></td><td> <table> <tr> <th colspan="2">Structure</th></tr> <tr> <th>file_info</th><th>File info (array of 16 characters) .</th></tr> <tr> <td></td><td> signed char [16] Minimum: -128 Maximum: 127 Scaling 1, 0, 0 (A,B,C): Scaled $1 * 10^0 *$ value: $(Raw+0)$ Resolution: 1 </td></tr> <tr> <th>size</th><th>Size (bytes) .</th></tr> <tr> <td></td><td> s32_type Minimum: 0 Maximum: 2147483647 Scaling 1, 0, 0 (A,B,C): Scaled $1 * 10^0 *$ value: $(Raw+0)$ </td></tr> </table> </td></tr> </table>	Union		descriptor	Descriptor.		<table> <tr> <th colspan="2">Structure</th></tr> <tr> <th>file_info</th><th>File info (array of 16 characters) .</th></tr> <tr> <td></td><td> signed char [16] Minimum: -128 Maximum: 127 Scaling 1, 0, 0 (A,B,C): Scaled $1 * 10^0 *$ value: $(Raw+0)$ Resolution: 1 </td></tr> <tr> <th>size</th><th>Size (bytes) .</th></tr> <tr> <td></td><td> s32_type Minimum: 0 Maximum: 2147483647 Scaling 1, 0, 0 (A,B,C): Scaled $1 * 10^0 *$ value: $(Raw+0)$ </td></tr> </table>	Structure		file_info	File info (array of 16 characters) .		signed char [16] Minimum: -128 Maximum: 127 Scaling 1, 0, 0 (A,B,C): Scaled $1 * 10^0 *$ value: $(Raw+0)$ Resolution: 1	size	Size (bytes) .		s32_type Minimum: 0 Maximum: 2147483647 Scaling 1, 0, 0 (A,B,C): Scaled $1 * 10^0 *$ value: $(Raw+0)$
Union																	
descriptor	Descriptor.																
	<table> <tr> <th colspan="2">Structure</th></tr> <tr> <th>file_info</th><th>File info (array of 16 characters) .</th></tr> <tr> <td></td><td> signed char [16] Minimum: -128 Maximum: 127 Scaling 1, 0, 0 (A,B,C): Scaled $1 * 10^0 *$ value: $(Raw+0)$ Resolution: 1 </td></tr> <tr> <th>size</th><th>Size (bytes) .</th></tr> <tr> <td></td><td> s32_type Minimum: 0 Maximum: 2147483647 Scaling 1, 0, 0 (A,B,C): Scaled $1 * 10^0 *$ value: $(Raw+0)$ </td></tr> </table>	Structure		file_info	File info (array of 16 characters) .		signed char [16] Minimum: -128 Maximum: 127 Scaling 1, 0, 0 (A,B,C): Scaled $1 * 10^0 *$ value: $(Raw+0)$ Resolution: 1	size	Size (bytes) .		s32_type Minimum: 0 Maximum: 2147483647 Scaling 1, 0, 0 (A,B,C): Scaled $1 * 10^0 *$ value: $(Raw+0)$						
Structure																	
file_info	File info (array of 16 characters) .																
	signed char [16] Minimum: -128 Maximum: 127 Scaling 1, 0, 0 (A,B,C): Scaled $1 * 10^0 *$ value: $(Raw+0)$ Resolution: 1																
size	Size (bytes) .																
	s32_type Minimum: 0 Maximum: 2147483647 Scaling 1, 0, 0 (A,B,C): Scaled $1 * 10^0 *$ value: $(Raw+0)$																

		Resolution: <i>1</i>
type	Type.	
		unsigned long Minimum: <i>0</i> Maximum: <i>65535</i> Scaling <i>1, 0, 0</i> (A,B,C): Scaled <i>1 *10⁰ *</i> value: <i>(Raw+0)</i> Resolution: <i>1</i>
address	Address.	
	Structure	
	domain_id	Domain ID (array of 6 bytes) . ANSI/CEA- 709.1 domain ID
		unsigned short [6] Minimum: <i>0</i> Maximum: <i>255</i> Scaling <i>1, 0, 0</i> (A,B,C): Scaled <i>1 *10⁰ *</i> value: <i>(Raw+0)</i> Resolution: <i>1</i>
	domain_length	Domain length (ANSI/CEA-709.1 domain length) . Valid domain lengths are 0, 1, 3, and 6.
		unsigned short Minimum: <i>0</i> Maximum: <i>6</i> Scaling <i>1, 0, 0</i> (A,B,C): Scaled <i>1 *10⁰ *</i> value: <i>(Raw+0)</i> Resolution: <i>1</i>
	subnet	Subnet (subnet number) . There can be 255 subnets (1-255) in a domain.
		unsigned short Minimum: <i>0</i> Maximum: <i>255</i> Invalid: <i>0</i> Scaling <i>1, 0, 0</i>

			(A,B,C): Scaled $1 * 10^0 *$ value: $(Raw+0)$ Resolution: 1
		node	Node (node number) . There can be 127 nodes (1-127) in a subnet.
			unsigned short Minimum: 0 Maximum: 127 Invalid: 0 Scaling $1, 0, 0$ (A,B,C): Scaled $1 * 10^0 *$ value: $(Raw+0)$ Resolution: 1

Formats:

```

SNVT_file_status: text( ("%m,%d %d ",status, number_of_files,
selected_file ),( (status == 1) ?("%d %d %s",adr.descriptor.size,
adr.descriptor.type,adr.descriptor.file_info ) :("%d <%x %x %x %x %x %
x> %d %d", adr.address.domain_length, adr.address.domain_id[0],
adr.address.domain_id[1], adr.address.domain_id[2],
adr.address.domain_id[3], adr.address.domain_id[4],
adr.address.domain_id[5], adr.address.subnet, adr.address.node) ) )

```

```

SNVT_file_status#LO: text( ("%m%d %d ",status, number_of_files,
selected_file ),( (status == 1) ?("%d %d %s",adr.descriptor.size,
adr.descriptor.type,adr.descriptor.file_info ) :("%d <%x %x %x %x %x %
x> %d %d", adr.address.domain_length, adr.address.domain_id[0],
adr.address.domain_id[1], adr.address.domain_id[2],
adr.address.domain_id[3], adr.address.domain_id[4],
adr.address.domain_id[5], adr.address.subnet, adr.address.node) ) )

```

Used by:

SFPTnodeObject

SNVT_fire_indcte

Overview:

Fire indicator type (fire indicator type names) .

Details:

Standard:

yes

Resource Set:

Standard 00:00:00:00:00:00:00:00-0

Index:

133

Obsolete:	<i>no</i>
Size:	<i>1</i>
Programmatic Name:	<i>SNVT_fire_indcte</i>
Neuron C Type:	<i>fire_indicator_t</i>
Formats:	<i>SNVT_fire_indcte: text("%m")</i>
Used by:	<i>SCPTfireIndicate</i>

SNVT_fire_init

Overview:

Fire initiator type (fire initiator type names) .

Details:

Standard:	<i>yes</i>
Resource Set:	<i>Standard 00:00:00:00:00:00:00:00-0</i>
Index:	<i>132</i>
Obsolete:	<i>no</i>
Size:	<i>1</i>
Programmatic Name:	<i>SNVT_fire_init</i>
Neuron C Type:	<i>fire_initiator_t</i>
Formats:	<i>SNVT_fire_init: text("%m")</i>
Used by:	<i>SCPTfireInitType</i>

SNVT_fire_test

Overview:

Fire test request (fire test names) .

Details:

Standard:	<i>yes</i>
Resource Set:	<i>Standard 00:00:00:00:00:00:00:00-0</i>
Index:	<i>130</i>
Obsolete:	<i>no</i>
Size:	<i>1</i>

Programmatic Name: *SNVT_fire_test*
 Neuron C Type: *fire_test_t*
 Formats: *SNVT_fire_test: text("%m")*
 Used by: *SFPTpullStationFireInitiator SFPTsmokeFireInitiatorConvent*
SFPTsmokeFireInitiatorIntelli SFPTthermalFireInitiator
SFPTuniversalFireInitiator

SNVT_flow

Overview:

Flow volume (liters/second) .

Details:

Standard: *yes*
 Resource Set: *Standard 00:00:00:00:00:00:00:00-0*
 Index: *15*
 Obsolete: *no*
 Size: *2*
 Programmatic Name: *SNVT_flow*
 Neuron C Type: *unsigned long*
 Minimum: *0*
 Maximum: *65535*
 Invalid: *65535*
 Scaling (A,B,C): *1, 0, 0*
 Scaled value: *1 * 10⁰ *(Raw+0)*
 Resolution: *1*
 Formats: *SNVT_flow#SI: text("%d")*
 *SNVT_flow#US: text("%d", *2.1189+0(0:966))*
 *SNVT_flow#US_liq: text("%d", *0.26418+0(0:837))*
 Used by: *SCPTmaxFlow SCPTmaxFlowHeat SCPTmaxFlowUnit*
SCPTminDeltaFlow SCPTminFlow SCPTminFlowHeat
SCPTminFlowHeatStby SCPTminFlowStby SCPTminFlowUnit
SCPTminFlowUnitHeat SCPTminFlowUnitStby
SCPTminOutdoorAirFlowSetpoint SCPTnomAirFlow
SCPTnomAirFlowHeat SFPTairVelocitySensor SFPTdamperActuator
SFPTdischargeAirController SFPTsccAHU SFPTsccChilledCeiling
SFPTsccFanCoil SFPTsccHeatPump SFPTsccRadiator
SFPTsccRooftop SFPTsccSelfContained SFPTsccUnitVentilator
SFPTsccVAV SFPTspaceComfortController SFPTvariableAirVolume

SNVT_flow_dir

Overview:

Flow direction (flow direction names) . Direction of allowable flow, or direction of present flow.

Desired state for an entry object such as, a door, lock, sluice, or something that controls entry of an area.

Details:

Standard:	<i>yes</i>
Resource Set:	<i>Standard 00:00:00:00:00:00:00:00-0</i>
Index:	<i>171</i>
Obsolete:	<i>no</i>
Size:	<i>1</i>
Programmatic Name:	<i>SNVT_flow_dir</i>
Neuron C Type:	<i>flow_direction_t</i>
Formats:	<i>SNVT_flow_dir: text("%m")</i>
Used by:	<i>SFPTentryExit</i>

SNVT_flow_f

Overview:

Flow volume (liters/second) .

Details:

Standard:	<i>yes</i>
Resource Set:	<i>Standard 00:00:00:00:00:00:00:00-0</i>
Index:	<i>53</i>
Obsolete:	<i>no</i>
Size:	<i>4</i>
Programmatic Name:	<i>SNVT_flow_f</i>
Neuron C Type:	<i>float</i>
	<i>Minimum: -3.4028234663853E+038</i>
	<i>Maximum: 3.4028234663853E+038</i>

Formats: *SNVT_flow_f#SI: text("%f")*

*SNVT_flow_f#US: text("%f", *2.1189+0(0:966))*

*SNVT_flow_f#US_liq: text("%f", *0.26418+0(0:837))*

Used by: *SCPToffsetFlow SFP TairVelocitySensor*

SNVT_flow_mil

Overview:

Flow volume (milliliters/second) .

Details:

Standard: *yes*

Resource Set: *Standard 00:00:00:00:00:00:00:00-0*

Index: *16*

Obsolete: *no*

Size: *2*

Programmatic Name: *SNVT_flow_mil*

Neuron C Type: *unsigned long*

Minimum: *0*

Maximum: *65535*

Scaling (A,B,C): *1, 0, 0*

Scaled value: *1 *10⁰ *(Raw+0)*

Resolution: *1*

Formats: *SNVT_flow_mil#SI: text("%d", *1+0(0:838))*

*SNVT_flow_mil#US: text("%d", *0.0338+0(0:839))*

Used by: *SNVT_pumpset_sn*

SNVT_flow_p

Overview:

Flow volume (cubic meters/hour) .

Used for heating, ventilation, and air conditioning applications. The typical flow in this area is 0.01 to 650 m³/h.

Details:

Standard:	<i>yes</i>
Resource Set:	<i>Standard 00:00:00:00:00:00:00:00-0</i>
Index:	<i>161</i>
Obsolete:	<i>no</i>
Size:	<i>2</i>
Programmatic Name:	<i>SNVT_flow_p</i>
Neuron C Type:	unsigned long
	Minimum: <i>0</i>
	Maximum: <i>65535</i>
	Invalid: <i>65535</i>
	Scaling (A,B,C): <i>1, -2, 0</i>
	Scaled value: <i>1 * 10⁻² * (Raw + 0)</i>
	Resolution: <i>0.01</i>
Formats:	<i>SNVT_flow_p#SI: text("%f", *1+0(0:1408))</i>
	<i>SNVT_flow_p#US: text("%f", *0.588578+0(0:1405))</i>
Used by:	<i>SCPTmaxFlowSetpoint SCPTmaxRemoteFlowSetpoint</i> <i>SCPTminFlowSetpoint SCPTminRemoteFlowSetpoint</i> <i>SCPTpumpCharacteristic SCPTvalveKvs SFPTpumpController</i>

SNVT_freq_f

Overview:

Frequency (Hertz) .

Details:

Standard:	<i>yes</i>
Resource Set:	<i>Standard 00:00:00:00:00:00:00:00-0</i>
Index:	<i>75</i>
Obsolete:	<i>no</i>
Size:	<i>4</i>
Programmatic Name:	<i>SNVT_freq_f</i>
Neuron C Type:	float
	Minimum: <i>0</i>
	Maximum: <i>3.4028234663853E+038</i>
Formats:	<i>SNVT_freq_f: text("%f")</i>

SNVT_freq_hz

Overview:

Frequency (Hertz) .

Details:

Standard:	<i>yes</i>
Resource Set:	<i>Standard 00:00:00:00:00:00:00:00-0</i>
Index:	<i>76</i>
Obsolete:	<i>no</i>
Size:	<i>2</i>
Programmatic Name:	<i>SNVT_freq_hz</i>
Neuron C Type:	unsigned long
	Minimum: <i>0</i>
	Maximum: <i>65535</i>
	Scaling (A,B,C): <i>1, -1, 0</i>
	Scaled value: <i>1 * 10⁻¹ * (Raw+0)</i>
	Resolution: <i>0.1</i>
Formats:	<i>SNVT_freq_hz: text("%f")</i>
Used by:	<i>SCPTflashFreq SCPTnomFreq SCPTnormalRotationalSpeed SCPTstandbyRotationalSpeed SFPTautomaticTransferSwitch SFPTgeneratorSet SNVT_pump_sensor</i>

SNVT_freq_kilohz

Overview:

Frequency (kiloHertz) .

Details:

Standard:	<i>yes</i>
Resource Set:	<i>Standard 00:00:00:00:00:00:00:00-0</i>
Index:	<i>77</i>
Obsolete:	<i>no</i>
Size:	<i>2</i>
Programmatic Name:	<i>SNVT_freq_kilohz</i>
Neuron C Type:	unsigned long

Minimum: 0
Maximum: 65535
Scaling (A,B,C): 1, -1, 0
Scaled value: $1 * 10^{-1} * (Raw + 0)$
Resolution: 0.1

Formats: *SNVT_freq_kilohz: text("%f")*

SNVT_freq_milhz

Overview:

Frequency (Hertz) .

Details:

Standard: *yes*
Resource Set: *Standard 00:00:00:00:00:00:00:00-0*
Index: *78*
Obsolete: *no*
Size: *2*
Programmatic Name: *SNVT_freq_milhz*
Neuron C Type: unsigned long
Minimum: 0
Maximum: 65535
Scaling (A,B,C): 1, -4, 0
Scaled value: $1 * 10^{-4} * (Raw + 0)$
Resolution: 0.0001

Formats: *SNVT_freq_milhz: text("%f")*

SNVT_gfci_status

Overview:

GFCI status type (GFCI status type names) .

Used to communicate the status of the Ground-Fault Circuit-Interrupter (GFCI) device.

Details:

Standard: *yes*
 Resource Set: *Standard 00:00:00:00:00:00:00:00-0*
 Index: *154*
 Obsolete: *no*
 Size: *1*
 Programmatic Name: *SNVT_gfci_status*
 Neuron C Type: *gfci_status_t*
 Formats: *SNVT_gfci_status: text("%m")*

SNVT_grammage

Overview:

Grammage (grams/sq meter) .

Details:

Standard: *yes*
 Resource Set: *Standard 00:00:00:00:00:00:00:00-0*
 Index: *71*
 Obsolete: *no*
 Size: *2*
 Programmatic Name: *SNVT_grammage*
 Neuron C Type: *unsigned long*
 Minimum: *0*
 Maximum: *65535*
 Scaling (A,B,C): *1, -1, 0*
 Scaled value: *$1 * 10^{-1} * (Raw + 0)$*
 Resolution: *0.1*
 Formats: *SNVT_grammage: text("%f")*

SNVT_grammage_f

Overview:

Grammage (grams/sq meter) .

Details:

Standard: *yes*
 Resource Set: *Standard 00:00:00:00:00:00:00:00-0*
 Index: *72*
 Obsolete: *no*
 Size: *4*
 Programmatic Name: *SNVT_grammage_f*
 Neuron C Type: *float*
 Minimum: *0*
 Maximum: *3.4028234663853E+038*
 Formats: *SNVT_grammage_f: text("%f")*

SNVT_hvac_emerg

Overview:

HVAC emergency mode (emergency mode names) .

Used for heating, ventilation, and air-conditioning applications.

Details:

Standard: *yes*
 Resource Set: *Standard 00:00:00:00:00:00:00:00-0*
 Index: *103*
 Obsolete: *no*
 Size: *1*
 Programmatic Name: *SNVT_hvac_emerg*
 Neuron C Type: *emerg_t*
 Formats: *SNVT_hvac_emerg: text("%m")*
 Used by: *SFPTdamperActuator SFPTdischargeAirController*
 SFPTfireSmokeDamperActuator SFPT HVAC Valve Positioner
 SFPTpullStationFireInitiator SFPTroofTopUnit SFPTsccAHU
 SFPTsccChilledCeiling SFPTsccFanCoil SFPTsccHeatPump
 SFPTsccRadiator SFPTsccRooftop SFPTsccSelfContained
 SFPTsccUnitVentilator SFPTsccVAV
 SFPTsmokeFireInitiatorConvent SFPTsmokeFireInitiatorIntelli
 SFPTspaceComfortController SFPTthermalFireInitiator
 SFPTunitVentilatorController SFPTvariableAirVolume

SNVT_hvac_mode

Overview:

HVAC mode (HVAC mode names) .

Used for heating, ventilation, and air-conditioning applications.

Details:

Standard:	<i>yes</i>
Resource Set:	<i>Standard 00:00:00:00:00:00:00:00-0</i>
Index:	<i>108</i>
Obsolete:	<i>no</i>
Size:	<i>1</i>
Programmatic Name:	<i>SNVT_hvac_mode</i>
Neuron C Type:	<i>hvac_t</i>
Formats:	<i>SNVT_hvac_mode: text("%m")</i>
Used by:	<i>SCPThvacMode SFPTboilerController SFPTchilledCeilingController SFPTchiller SFPTdischargeAirController SFPTfanCoilUnit SFPTheatPump SFPT HVAC Valve Positioner SFPTpumpController SFPTroofTopUnit SFPTsccAHU SFPTsccChilledCeiling SFPTsccCommandModule SFPTsccFanCoil SFPTsccHeatPump SFPTsccRadiator SFPTsccRooftop SFPTsccSelfContained SFPTsccUnitVentilator SFPTsccVAV SFPTspaceComfortController SFPTthermostat SFPTunitVentilatorController SFPTvariableAirVolume</i>

SNVT_hvac_overid

Overview:

HVAC override (state, pct, flow) .

Used for heating, ventilation, and air-conditioning applications.

Details:

Standard:	<i>yes</i>
Resource Set:	<i>Standard 00:00:00:00:00:00:00:00-0</i>
Index:	<i>111</i>
Obsolete:	<i>no</i>
Size:	<i>5</i>
Programmatic Name:	<i>SNVT_hvac_overid</i>
Neuron C Type:	Structure

--	--

state	HVAC override state (override state names) .
	<i>hvac_overid_t</i>
percent	Percent (% of full scale) . Position or flow override value
	signed long Minimum: -32768 Maximum: 32767 Invalid: 32767 Scaling 5, -3, 0 (A,B,C): Scaled value: $5 * 10^{-3} * (Raw + 0)$ Resolution: 0.005
flow	Flow (liters/second) .
	unsigned long Minimum: 0 Maximum: 65535 Invalid: 65535 Scaling 1, 0, 0 (A,B,C): Scaled value: $1 * 10^0 * (Raw + 0)$ Resolution: 1

Formats:

SNVT_hvac_overid#SI: text("%m,%f,%f", state, percent, flow)

SNVT_hvac_overid#SI_LO: text("%m|%f|%f", state, percent, flow)

*SNVT_hvac_overid#US: text("%m,%f,%f", state, percent, flow*2.1189+0(0:966))*

*SNVT_hvac_overid#US_LO: text("%m|%f|%f", state, percent, flow*2.1189+0(0:966))*

Used by:

*SFPTchilledCeilingController SFPTdamperActuator SFPTsccAHU
SFPTsccChilledCeiling SFPTsccFanCoil SFPTsccHeatPump
SFPTsccRadiator SFPTsccRooftop SFPTsccSelfContained
SFPTsccUnitVentilator SFPTsccVAV SFPTspaceComfortController
SFPTunitVentilatorController SFPTvariableAirVolume*

SNVT_hvac_satsts

Overview:

HVAC saturation status. A value of 0 in a field indicates that the resource associated with that field has not saturated or reached an end stop before attaining the required setpoint. A value of 1 indicates that the resource associated with that field has saturated or reached an end stop without

attaining the required setpoint.

A value of 0 indicates that the resource associated with that field has not saturated or reached an end stop before attaining the required setpoint. A value of 1 indicates that the resource associated with that field has saturated or reached an end stop without attaining the required setpoint.

Details:

Standard: *yes*
 Resource Set: *Standard 00:00:00:00:00:00:00:00-0*
 Index: *172*
 Obsolete: *no*
 Size: *2*
 Programmatic Name: *SNVT_hvac_satsts*
 Neuron C Type:

Structure	
pri_heat	Primary heating saturation status (boolean) . A value of 0 indicates primary heating is not saturated. A value of 1 indicates primary heating is saturated.
	bitfield Signed: <i>no</i> Width: <i>1</i> Offset: <i>0</i> Minimum: <i>0</i> Maximum: <i>1</i> Scaling (A,B,C): <i>1, 0, 0</i> Scaled value: $1 * 10^0 * (Raw + 0)$ Resolution: <i>1</i>
sec_heat	Secondary heating saturation status (boolean) . A value of 0 indicates secondary heating is not saturated. A value of 1 indicates secondary heating is saturated.
	bitfield Signed: <i>no</i> Width: <i>1</i> Offset: <i>1</i> Minimum: <i>0</i> Maximum: <i>1</i> Scaling (A,B,C): <i>1, 0, 0</i> Scaled value: $1 * 10^0 * (Raw + 0)$ Resolution: <i>1</i>
pri_cool	Primary cooling saturation status (boolean) . A value of 0 indicates primary cooling is not

	saturated. A value of 1 indicates primary cooling is saturated.
	bitfield Signed: <i>no</i> Width: <i>1</i> Offset: <i>2</i> Minimum: <i>0</i> Maximum: <i>1</i> Scaling <i>1, 0, 0</i> (A,B,C): Scaled value: $1 * 10^0 * (Raw + 0)$ Resolution: <i>1</i>
sec_cool	Secondary cooling saturation status (boolean) . A value of 0 indicates secondary cooling is not saturated. A value of 1 indicates secondary cooling is saturated.
	bitfield Signed: <i>no</i> Width: <i>1</i> Offset: <i>3</i> Minimum: <i>0</i> Maximum: <i>1</i> Scaling <i>1, 0, 0</i> (A,B,C): Scaled value: $1 * 10^0 * (Raw + 0)$ Resolution: <i>1</i>
pri_duct_starved	Primary duct saturation status (boolean) . A value of 0 indicates primary duct is not saturated (starved). A value of 1 indicates primary duct is saturated (starved).
	bitfield Signed: <i>no</i> Width: <i>1</i> Offset: <i>4</i> Minimum: <i>0</i> Maximum: <i>1</i> Scaling <i>1, 0, 0</i> (A,B,C): Scaled value: $1 * 10^0 * (Raw + 0)$ Resolution: <i>1</i>
sec_duct_starved	Secondary duct saturation status (boolean) . A value of 0 indicates secondary duct is not saturated (starved). A value of 1 indicates secondary duct is saturated (starved).

	bitfield Signed: <i>no</i> Width: <i>1</i> Offset: <i>5</i> Minimum: <i>0</i> Maximum: <i>1</i> Scaling <i>1, 0, 0</i> (A,B,C): Scaled value: $1 * 10^0 * (Raw + 0)$ Resolution: <i>1</i>
reserved	.
	bitfield Signed: <i>no</i> Width: <i>2</i> Offset: <i>6</i> Minimum: <i>0</i> Maximum: <i>3</i> Scaling <i>1, 0, 0</i> (A,B,C): Scaled value: $1 * 10^0 * (Raw + 0)$ Resolution: <i>1</i>
reserved1	.
	bitfield Signed: <i>no</i> Width: <i>4</i> Offset: <i>0</i> Minimum: <i>0</i> Maximum: <i>15</i> Scaling <i>1, 0, 0</i> (A,B,C): Scaled value: $1 * 10^0 * (Raw + 0)$ Resolution: <i>1</i>
manufacturer_defined	Manufacturer defined (boolean) . Four manufacturer-defined bits -- please see product documentation for proper interpretation of these bits
	bitfield Signed: <i>no</i> Width: <i>4</i> Offset: <i>4</i> Minimum: <i>0</i> Maximum: <i>15</i> Scaling <i>1, 0, 0</i>

	(A,B,C): Scaled value: $1 * 10^0 * (Raw + 0)$ Resolution: 1
--	---

Formats: *SNVT_hvac_satsts: text("%d %d %d %d %d %d %d", pri_heat, sec_heat, pri_cool, sec_cool, pri_duct_starved, sec_duct_starved, manufacturer_defined)*

Used by: *SFPTsccAHU SFPTsccChilledCeiling SFPTsccFanCoil
SFPTsccHeatPump SFPTsccRadiator SFPTsccRooftop
SFPTsccSelfContained SFPTsccUnitVentilator SFPTsccVAV
SFPTspaceComfortController*

SNVT_hvac_status

Overview:

HVAC status (mode, 5 percents, flag) .

Used for heating, ventilation, and air-conditioning applications.

Details:

Standard: *yes*
Resource Set: *Standard 00:00:00:00:00:00:00:00-0*
Index: *112*
Obsolete: *no*
Size: *12*
Programmatic Name: *SNVT_hvac_status*

Neuron C Type:

Structure	
mode	HVAC status mode (HVAC mode names) .
	<i>hvac_t</i>
heat_output_primary	Primary heat output (% of full scale) .
	signed long Minimum: -32768 Maximum: 32767 Invalid: 32767 Scaling 5, -3, 0 (A,B,C): Scaled value: $5 * 10^{-3} * (Raw + 0)$ Resolution: 0.005
heat_output_secondary	Secondary heat output (% of full scale) .

	signed long Minimum: -32768 Maximum: 32767 Invalid: 32767 Scaling 5, -3, 0 (A,B,C): Scaled value: $5 * 10^{-3} * (Raw + 0)$ Resolution: 0.005
cool_output	Cooling output (% of full scale) .
	signed long Minimum: -32768 Maximum: 32767 Invalid: 32767 Scaling 5, -3, 0 (A,B,C): Scaled value: $5 * 10^{-3} * (Raw + 0)$ Resolution: 0.005
econ_output	Economizer output (% of full scale) .
	signed long Minimum: -32768 Maximum: 32767 Invalid: 32767 Scaling 5, -3, 0 (A,B,C): Scaled value: $5 * 10^{-3} * (Raw + 0)$ Resolution: 0.005
fan_output	Fan output (% of full scale) .
	signed long Minimum: -32768 Maximum: 32767 Invalid: 32767 Scaling 5, -3, 0 (A,B,C): Scaled value: $5 * 10^{-3} * (Raw + 0)$ Resolution: 0.005
in_alarm	In alarm state (alarm value) . Non-zero value represents a manufacturer-specific alarm state
	unsigned short Minimum: 0 Maximum: 255 Scaling 1, 0, 0 (A,B,C):

	Scaled value: Resolution: 1
--	--------------------------------

Formats: *SNVT_hvac_status: text("%m,%f,%f,%f,%f,%f,%d", mode, heat_output_primary, heat_output_secondary, cool_output, econ_output, fan_output, in_alarm)*

SNVT_hvac_status#LO: text("%m|%f|%f|%f|%f|%f|%d", mode, heat_output_primary, heat_output_secondary, cool_output, econ_output, fan_output, in_alarm)

Used by: *SFPTchilledCeilingController SFPTdischargeAirController
SFPTfanCoilUnit SFPTheatPump SFPTroofTopUnit SFPTsccAHU
SFPTsccChilledCeiling SFPTsccCommandModule SFPTsccFanCoil
SFPTsccHeatPump SFPTsccRadiator SFPTsccRooftop
SFPTsccSelfContained SFPTsccUnitVentilator SFPTsccVAV
SFPTspaceComfortController SFPTthermostat
SFPTunitVentilatorController SFPTvariableAirVolume*

SNVT_hvac_type

Overview:

HVAC unit type (HVAC unit type names) .

Used to indicate the type of HVAC equipment that is being controlled.

Details:

Standard: *yes*
Resource Set: *Standard 00:00:00:00:00:00:00:00-0*
Index: *145*
Obsolete: *no*
Size: *1*
Programmatic Name: *SNVT_hvac_type*
Neuron C Type: *hvac_hvt_t*
Formats: *SNVT_hvac_type: text("%m")*
Used by: *SCPTThvacType*

SNVT_length

Overview:

Length (meters) .

Details:

Standard:	<i>yes</i>
Resource Set:	<i>Standard 00:00:00:00:00:00:00:00-0</i>
Index:	<i>17</i>
Obsolete:	<i>no</i>
Size:	<i>2</i>
Programmatic Name:	<i>SNVT_length</i>
Neuron C Type:	unsigned long
	Minimum: <i>0</i>
	Maximum: <i>65535</i>
	Scaling (A,B,C): <i>1, -1, 0</i>
	Scaled value: <i>$1 * 10^{-1} * (Raw + 0)$</i>
	Resolution: <i>0.1</i>
Formats:	<i>SNVT_length#SI: text("%f")</i>
	<i>SNVT_length#US: text("%f", *3.28084+0(0:840))</i>
Used by:	<i>SFPThvacValvePositioner</i>

SNVT_length_f

Overview:

Length (meters) .

Details:

Standard:	<i>yes</i>
Resource Set:	<i>Standard 00:00:00:00:00:00:00:00-0</i>
Index:	<i>54</i>
Obsolete:	<i>no</i>
Size:	<i>4</i>
Programmatic Name:	<i>SNVT_length_f</i>
Neuron C Type:	float
	Minimum: <i>0</i>
	Maximum: <i>3.4028234663853E+038</i>
Formats:	<i>SNVT_length_f#SI: text("%f")</i>
	<i>SNVT_length_f#US: text("%f", *3.28084+0(0:840))</i>

SNVT_length_kilo

Overview:

Length (kilometers) .

Details:

Standard:	<i>yes</i>
Resource Set:	<i>Standard 00:00:00:00:00:00:00:00-0</i>
Index:	<i>18</i>
Obsolete:	<i>no</i>
Size:	<i>2</i>
Programmatic Name:	<i>SNVT_length_kilo</i>
Neuron C Type:	unsigned long
	Minimum: <i>0</i>
	Maximum: <i>65535</i>
	Scaling (A,B,C): <i>1, -1, 0</i>
	Scaled value: <i>$1 * 10^{-1} * (Raw + 0)$</i>
	Resolution: <i>0.1</i>
Formats:	<i>SNVT_length_kilo#SI: text("%f")</i>
	<i>SNVT_length_kilo#US: text("%f", *0.62137+0(0:841))</i>

SNVT_length_micr

Overview:

Length (micrometers (microns)) .

Details:

Standard:	<i>yes</i>
Resource Set:	<i>Standard 00:00:00:00:00:00:00:00-0</i>
Index:	<i>19</i>
Obsolete:	<i>no</i>
Size:	<i>2</i>
Programmatic Name:	<i>SNVT_length_micr</i>
Neuron C Type:	unsigned long

Minimum: 0
 Maximum: 65535
 Scaling (A,B,C): 1, -1, 0
 Scaled value: $1 * 10^{-1} * (Raw + 0)$
 Resolution: 0.1

Formats: *SNVT_length_micr: text("%f", *1+0(0:843))*

SNVT_length_mil

Overview:

Length (millimeters) .

Details:

Standard: yes
 Resource Set: *Standard 00:00:00:00:00:00:00:00-0*
 Index: 20
 Obsolete: no
 Size: 2
 Programmatic Name: *SNVT_length_mil*
 Neuron C Type: unsigned long
 Minimum: 0
 Maximum: 65535
 Scaling (A,B,C): 1, -1, 0
 Scaled value: $1 * 10^{-1} * (Raw + 0)$
 Resolution: 0.1
 Formats: *SNVT_length_mil#SI: text("%f")*
 *SNVT_length_mil#US: text("%f", *0.03937+0(0:844))*
 Used by: *SCPTmaxStroke SCPTminStroke SCPTvalveNominalSize*
 SCPTvalveStroke SFPTHvacValvePositioner

SNVT_lev_cont

Overview:

Continuous level (% of full level) .

Details:

Standard:	<i>yes</i>
Resource Set:	<i>Standard 00:00:00:00:00:00:00:00-0</i>
Index:	<i>21</i>
Obsolete:	<i>no</i>
Size:	<i>1</i>
Programmatic Name:	<i>SNVT_lev_cont</i>
Neuron C Type:	unsigned short
	Minimum: <i>0</i>
	Maximum: <i>200</i>
	Scaling (A,B,C): <i>5, -1, 0</i>
	Scaled value: <i>$5 * 10^{-1} * (Raw + 0)$</i>
	Resolution: <i>0.5</i>
Formats:	<i>SNVT_lev_cont: text("%f")</i>
Used by:	<i>SCPTmaxOut SCPTminDeltaLevel SCPTonOffHysteresis SCPTstep SCPTstepValue SFPTchannelMonitor SFPT HVACRelativeHumiditySensor SFPTsmokeFireInitiatorIntelli</i>

SNVT_lev_cont_f

Overview:

Continuous level (% of full scale) .

Details:

Standard:	<i>yes</i>
Resource Set:	<i>Standard 00:00:00:00:00:00:00:00-0</i>
Index:	<i>55</i>
Obsolete:	<i>no</i>
Size:	<i>4</i>
Programmatic Name:	<i>SNVT_lev_cont_f</i>
Neuron C Type:	float
	Minimum: <i>0</i>
	Maximum: <i>100</i>
Formats:	<i>SNVT_lev_cont_f: text("%f")</i>
Used by:	<i>SFPT HVACRelativeHumiditySensor</i>

SNVT_lev_disc

Overview:

Discrete level (level names) . This SNVT is obsolete. Use SNVT_switch instead.

Obsolete. SNVT_switch or SNVT_switch2 should be used for communicating state with discrete devices as well as level with continuous devices.

Details:

Standard:	<i>yes</i>
Resource Set:	<i>Standard 00:00:00:00:00:00:00:00-0</i>
Index:	<i>22</i>
Obsolete:	<i>yes</i>
Size:	<i>1</i>
Programmatic Name:	<i>SNVT_lev_disc</i>
Neuron C Type:	<i>discrete_levels_t</i>
Formats:	<i>SNVT_lev_disc: text("%m")</i>
Used by:	<i>SCPTinvrtOut SFPtrefrigDisplayCaseControllerDefrost SFPtrefrigDisplayCaseControllerThermostat SFPTutilityDataLoggerRegister</i>

SNVT_lev_percent

Overview:

Percentage level (% of full level) . Level percent. SNVT_switch should be used instead of SNVT_lev_percent, with the exception of network variables that are used to communicate a percentage value and that require the additional resolution provided by SNVT_lev_percent; or for network variable members of functional profiles that are designed primarily for interfacing with SNVT_lev_percent members of other profiles.

SNVT_switch or SNVT_switch2 should be used instead of SNVT_lev_percent, with the exception of network variables that are used to communicate a percentage value and that require the additional resolution provided by SNVT_lev_percent. SNVT_lev_percent may also be used for network variable members of functional profiles that are designed primary for interfacing with network variable members of existing profiles that are defined as SNVT_lev_percent. SNVT_switch or SNVT_switch2 should be used for communicating state with discrete devices as well as level with continuous devices.

Note: The SNVT_lev_percent type is marked as obsolete in the Version 11.00 standard resource file set. This will be changed in the next update to the standard resource file set.

Details:

Standard:	<i>yes</i>
-----------	------------

Resource Set:	<i>Standard 00:00:00:00:00:00:00:00-0</i>
Index:	<i>81</i>
Obsolete:	<i>no</i>
Size:	<i>2</i>
Programmatic Name:	<i>SNVT_lev_percent</i>
Neuron C Type:	signed long
	Minimum: <i>-32768</i>
	Maximum: <i>32767</i>
	Invalid: <i>32767</i>
	Scaling (A,B,C): <i>5, -3, 0</i>
	Scaled value: <i>$5 * 10^{-3} * (Raw + 0)$</i>
	Resolution: <i>0.005</i>
Formats:	<i>SNVT_lev_percent: text("%f")</i>
Used by:	<i>SCPTairTemp1Alrm SCPTairTemp1Day SCPTairTemp1Night</i> <i>SCPTcontrolSignal SCPTcontrolTemperatureWeighting</i> <i>SCPTdefScale SCPTemergencyPosition SCPTexhaustEnablePosition</i> <i>SCPTfanDifferentialSetpoint SCPTfreeCoolPosition SCPTHumSetpt</i> <i>SCPTlimitChlrCap SCPTmaxReturnExhaustFanCapacity</i> <i>SCPTmaxSetpoint SCPTmaxSupplyFanCapacity SCPTminDeltaRH</i> <i>SCPTminReturnExhaustFanCapacity SCPTminSetpoint</i> <i>SCPTminSupplyFanCapacity SCPTnightPurgePosition</i> <i>SCPToffsetRH SCPTreflection SCPTspaceHumSetpoint</i> <i>SCPTstrutOpen SFPTanalogInput SFPTanalogOutput</i> <i>SFPTboilerController SFPTchilledCeilingController SFPTchiller</i> <i>SFPTdamperActuator SFPTdischargeAirController SFPTfanCoilUnit</i> <i>SFPT HVACRelativeHumiditySensor SFPT HVACValvePositioner</i> <i>SFPTpumpController SFPTrefrigDisplayCaseControllerDefrost</i> <i>SFPTrefrigDisplayCaseControllerEvaporator SFPTroofTopUnit</i> <i>SFPTsccAHU SFPTsccChilledCeiling SFPTsccCommandModule</i> <i>SFPTsccFanCoil SFPTsccHeatPump SFPTsccRadiator</i> <i>SFPTsccRooftop SFPTsccSelfContained SFPTsccUnitVentilator</i> <i>SFPTsccVAV SFPTspaceComfortController SFPTsunblindController</i> <i>SFPTthermostat SFPTunitVentilatorController</i> <i>SFPTvariableAirVolume SFPTvariableSpeedMotorDrive</i> <i>SNVT_pos_ctrl</i>

SNVT_log_fx_request

Overview:

Log file transfer request.. Requests a data log to be transferred via FTP. Must be followed by a stanard FTP request to get the data log file. Required on devices implementing the Data Logger functional profile that support data log transfer via FTP.

Details:

Standard: *yes*

Resource Set: *Standard 00:00:00:00:00:00:00:00-0*
 Index: *193*
 Obsolete: *no*
 Size: *16*
 Programmatic Name: *SNVT_log_fx_request*
 Neuron C Type:

Structure	
requested_log	Requested log number.. The log number of the data log to be transferred. Logs are numbered from 1 to number_of_logs.
	unsigned long Minimum: <i>1</i> Maximum: <i>65535</i> Invalid: <i>0</i> Scaling <i>1, 0, 0</i> (A,B,C): Scaled value: $1 * 10^0 * (Raw + 0)$ Resolution: <i>1</i>
record_count	Record count.. The maximum number of log records to be transferred.
	unsigned quad Minimum: <i>0</i> Maximum: <i>4294967295</i> Scaling <i>1, 0, 0</i> (A,B,C): Scaled value: $1 * 10^0 * (Raw + 0)$ Resolution: <i>1</i>
start_time	Start time. (seconds) . Timestamp of first record to be transferred. If no records exist with this timestamp, the first record with a timestamp after this timestamp is the starting record.
	<i>SNVT_time_stamp_p</i>
end_time	End time. (seconds) . Timestamp of last record to be transferred. If no records exist with this timestamp, the last record with a timestamp before this timestamp is the ending record.
	<i>SNVT_time_stamp_p</i>

Used by: *SFPTnodeObject*

SNVT_log_fx_status

Overview:

Log file transfer status.. Reports the status of a data log file transfer using FTP. Required on devices implementing the Data Logger functional profile that support data log transfer via FTP.

Details:

Standard: *yes*
 Resource Set: *Standard 00:00:00:00:00:00:00:00-0*
 Index: *194*
 Obsolete: *no*
 Size: *5*
 Programmatic Name: *SNVT_log_fx_status*
 Neuron C Type:

Structure	
requestor_subnet	Requestor subnet ID.. Subnet ID of the device that requested the current log file transfer. Invalid if a file transfer is not active.
	unsigned short Minimum: <i>1</i> Maximum: <i>255</i> Invalid: <i>0</i> Scaling <i>1, 0, 0</i> (A,B,C): Scaled value: $1 * 10^0 * (Raw + 0)$ Resolution: <i>1</i>
requestor_node	Requestor node ID.. Node ID of the device that requested the current log file transfer. Invalid if a file transfer is not active.
	unsigned short Minimum: <i>1</i> Maximum: <i>255</i> Invalid: <i>0</i> Scaling <i>1, 0, 0</i> (A,B,C): Scaled value: $1 * 10^0 * (Raw + 0)$ Resolution: <i>1</i>
log_number	Data log number.. Log number for the log file currently being transferred via FTP. Invalid if none.
	unsigned long Minimum: <i>1</i> Maximum: <i>65535</i> Invalid: <i>0</i> Scaling <i>1, 0, 0</i> (A,B,C): Scaled value: $1 * 10^0 * (Raw + 0)$

	Resolution: 1
complete	Data log file transfer percent complete.. Percent of the current data log file transfer that has been completed. Invalid if none.
	unsigned short Minimum: 0 Maximum: 200 Invalid: 255 Scaling (A,B,C): $5, -1, 0$ Scaled value: $5 * 10^{-1} * (Raw + 0)$ Resolution: 0.5

Used by: *SFPTnodeObject*

SNVT_log_request

Overview:

Log status request.. Requests the current status of a data log. Status is reported by a SNVT_log_status output.

Details:

Standard: *yes*
 Resource Set: *Standard 00:00:00:00:00:00:00:00-0*
 Index: *195*
 Obsolete: *no*
 Size: *2*
 Programmatic Name: *SNVT_log_request*
 Neuron C Type: unsigned long
 Minimum: 1
 Maximum: 65535
 Invalid: 0
 Scaling (A,B,C): $1, 0, 0$
 Scaled value: $1 * 10^0 * (Raw + 0)$
 Resolution: 1

Used by: *SFPTnodeObject*

SNVT_log_status

Overview:

Log status (hundredths of second) . Reports the current status of a data log. Updated based on the cpLogNotificationThreshold value. Reports status only; alarms reported via Node Object nvoAlarm2 output. Required if the Node Object does not include an nvoLogStat output.

Details:

Standard:	<i>yes</i>
Resource Set:	<i>Standard 00:00:00:00:00:00:00:00-0</i>
Index:	<i>191</i>
Obsolete:	<i>no</i>
Size:	<i>30</i>
Programmatic Name:	<i>SNVT_log_status</i>
Neuron C Type:	

Structure	
status	Log state.. State of the selected data log.
	<i>log_status_t</i>
log_number	Selected log number.. The log number of the reported data log. Logs are numbered from 1 to number_of_logs.
	unsigned long Minimum: <i>1</i> Maximum: <i>65535</i> Invalid: <i>0</i> Scaling <i>1, 0, 0</i> (A,B,C): Scaled value: $1 * 10^0 * (Raw + 0)$ Resolution: <i>1</i>
level	Log level (Percent) . The percent of maximum records in the selected data log.
	unsigned short Minimum: <i>0</i> Maximum: <i>200</i> Invalid: <i>255</i> Scaling <i>5, -1, 0</i> (A,B,C): Scaled value: $5 * 10^{-1} * (Raw + 0)$ Resolution: <i>0.5</i>
record_count	Record count. (records) . Number of records in the selected data log. A record is a logged value and any associated data such as a timestamp.
	s32_type

	Minimum: <i>-2147483648</i> Maximum: <i>2147483647</i> Scaling <i>1, 0, 0</i> (A,B,C): Scaled <i>1 * 10⁰ *(Raw+0)</i> value: Resolution: <i>1</i>
byte_count	Byte count. (bytes) . Number of bytes in the selected data log.
	s32_type Minimum: <i>-2147483648</i> Maximum: <i>2147483647</i> Scaling <i>1, 0, 0</i> (A,B,C): Scaled <i>1 * 10⁰ *(Raw+0)</i> value: Resolution: <i>1</i>
total_record_count	Total record count. (records) . Total records collected in the selected data log since the data log was created. Wraps to 0 on overflow.
	s32_type Minimum: <i>-2147483648</i> Maximum: <i>2147483647</i> Scaling <i>1, 0, 0</i> (A,B,C): Scaled <i>1 * 10⁰ *(Raw+0)</i> value: Resolution: <i>1</i>
records_since_notification	Records since notification. (records) . The number of log records collected since the last notification.
	s32_type Minimum: <i>-2147483648</i> Maximum: <i>2147483647</i> Scaling <i>1, 0, 0</i> (A,B,C): Scaled <i>1 * 10⁰ *(Raw+0)</i> value: Resolution: <i>1</i>
current_notify_time	Current notify time.. Timestamp of the most recently collected data point.
	<i>SNVT_time_stamp_p</i>
previous_notify_time	Previous notify time. (seconds) . Timestamp of the most recently collected data point in the previous update to the log status.
	<i>SNVT_time_stamp_p</i>

Used by: *SFPTdataLogger SFPTnodeObject*

SNVT_lux

Overview:

Illumination (lux) .

1 lux = 1 lumen/m²

EXAMPLES

As a comparison:

1 foot-candle = 1 lumen/ft².

1 foot-candle = 10.76 lux.

Details:

Standard:	<i>yes</i>
Resource Set:	<i>Standard 00:00:00:00:00:00:00:00-0</i>
Index:	<i>79</i>
Obsolete:	<i>no</i>
Size:	<i>2</i>
Programmatic Name:	<i>SNVT_lux</i>
Neuron C Type:	unsigned long
	Minimum: <i>0</i>
	Maximum: <i>65535</i>
	Scaling (A,B,C): <i>1, 0, 0</i>
	Scaled value: <i>1 * 10⁰ * (Raw + 0)</i>
	Resolution: <i>1</i>
Formats:	<i>SNVT_lux: text("%d")</i>
Used by:	<i>SCPTfieldCalib SCPTluxSetpoint SFPTconstantLightController SFPTlightSensor SFPTsunblindController</i>

SNVT_magcard

Overview:

ISO 7811 (40 hexadecimal digits) . This data item contains data according to the ISO 7811 standard for card stripes.

This data item contains data according to the ISO 7811 standard for card stripes. A card reader conforming to ISO 7811 will read standard financial transaction cards (credit cards and ATM cards). ISO 7811 is similar to the credit-card account numbering system given in ANSI Standard X4.13-1971. This data type is compatible with the “magcard” I/O model in Neuron C. SNVT_magcard is NOT compatible with ISO 3554 (the “magtrack1” I/O model in Neuron C). See the *Neuron C Reference Guide* for more details.

The start-sentinel (0xB) is always present in digit1. The “value not available” for SNVT_magcard is defined as a start-sentinel in digit1, and an end-sentinel (0xF) in digit2. Parity fields are not included in the structure, and are not part of the 4-bit digits.

The longitudinal-redundancy check (LRC) character is not required, and therefore characters after the end-sentinel character should be ignored by a receiving device. For a sending device, digit40 should always contain 0x0, as it would only contain an LRC character anyway, and is stored as 0x0 in a Neuron Chip-based device.

Details:

Standard:	<i>yes</i>
Resource Set:	<i>Standard 00:00:00:00:00:00:00:00-0</i>
Index:	<i>86</i>
Obsolete:	<i>no</i>
Size:	<i>20</i>
Programmatic Name:	<i>SNVT_magcard</i>
Neuron C Type:	

Structure	
digit1	Digit 1 (hexadecimal digit) .
	bitfield Signed: <i>no</i> Width: <i>4</i> Offset: <i>0</i> Minimum: <i>11</i> Maximum: <i>11</i> Scaling <i>1, 0, 0</i> (A,B,C): Scaled <i>$1 * 10^0 * (Raw + 0)$</i> value: Resolution: <i>1</i>
digit2	Digit 2 (hexadecimal digit) .
	bitfield Signed: <i>no</i> Width: <i>4</i> Offset: <i>4</i> Minimum: <i>0</i> Maximum: <i>15</i> Scaling <i>1, 0, 0</i> (A,B,C): Scaled <i>$1 * 10^0 * (Raw + 0)$</i> value: Resolution: <i>1</i>

digit3	Digit 3 (hexadecimal digit) .
	bitfield Signed: <i>no</i> Width: <i>4</i> Offset: <i>0</i> Minimum: <i>0</i> Maximum: <i>15</i> Scaling <i>1, 0, 0</i> (A,B,C): Scaled <i>1 * 10⁰ *(Raw+0)</i> value: Resolution: <i>1</i>
digit4	Digit 4 (hexadecimal digit) .
	bitfield Signed: <i>no</i> Width: <i>4</i> Offset: <i>4</i> Minimum: <i>0</i> Maximum: <i>15</i> Scaling <i>1, 0, 0</i> (A,B,C): Scaled <i>1 * 10⁰ *(Raw+0)</i> value: Resolution: <i>1</i>
digit5	Digit 5 (hexadecimal digit) .
	bitfield Signed: <i>no</i> Width: <i>4</i> Offset: <i>0</i> Minimum: <i>0</i> Maximum: <i>15</i> Scaling <i>1, 0, 0</i> (A,B,C): Scaled <i>1 * 10⁰ *(Raw+0)</i> value: Resolution: <i>1</i>
digit6	Digit 6 (hexadecimal digit) .
	bitfield Signed: <i>no</i> Width: <i>4</i> Offset: <i>4</i> Minimum: <i>0</i> Maximum: <i>15</i> Scaling <i>1, 0, 0</i> (A,B,C): Scaled

	value: $1 * 10^0 * (Raw + 0)$ Resolution: 1
digit7	Digit 7 (hexadecimal digit) .
	bitfield Signed: <i>no</i> Width: 4 Offset: 0 Minimum: 0 Maximum: 15 Scaling $1, 0, 0$ (A,B,C): Scaled $1 * 10^0 * (Raw + 0)$ value: Resolution: 1
digit8	Digit 8 (hexadecimal digit) .
	bitfield Signed: <i>no</i> Width: 4 Offset: 4 Minimum: 0 Maximum: 15 Scaling $1, 0, 0$ (A,B,C): Scaled $1 * 10^0 * (Raw + 0)$ value: Resolution: 1
digit9	Digit 9 (hexadecimal digit) .
	bitfield Signed: <i>no</i> Width: 4 Offset: 0 Minimum: 0 Maximum: 15 Scaling $1, 0, 0$ (A,B,C): Scaled $1 * 10^0 * (Raw + 0)$ value: Resolution: 1
digit10	Digit 10 (hexadecimal digit) .
	bitfield Signed: <i>no</i> Width: 4 Offset: 4 Minimum: 0 Maximum: 15

	Scaling <i>1, 0, 0</i> (A,B,C): Scaled $1 * 10^0 * (Raw + 0)$ value: Resolution: <i>1</i>
digit11	Digit 11 (hexadecimal digit) .
	bitfield Signed: <i>no</i> Width: <i>4</i> Offset: <i>0</i> Minimum: <i>0</i> Maximum: <i>15</i> Scaling <i>1, 0, 0</i> (A,B,C): Scaled $1 * 10^0 * (Raw + 0)$ value: Resolution: <i>1</i>
digit12	Digit 12 (hexadecimal digit) .
	bitfield Signed: <i>no</i> Width: <i>4</i> Offset: <i>4</i> Minimum: <i>0</i> Maximum: <i>15</i> Scaling <i>1, 0, 0</i> (A,B,C): Scaled $1 * 10^0 * (Raw + 0)$ value: Resolution: <i>1</i>
digit13	Digit 13 (hexadecimal digit) .
	bitfield Signed: <i>no</i> Width: <i>4</i> Offset: <i>0</i> Minimum: <i>0</i> Maximum: <i>15</i> Scaling <i>1, 0, 0</i> (A,B,C): Scaled $1 * 10^0 * (Raw + 0)$ value: Resolution: <i>1</i>
digit14	Digit 14 (hexadecimal digit) .
	bitfield Signed: <i>no</i> Width: <i>4</i>

	Offset: 4 Minimum: 0 Maximum: 15 Scaling 1, 0, 0 (A,B,C): Scaled value: $1 * 10^0 * (Raw + 0)$ Resolution: 1
digit15	Digit 15 (hexadecimal digit) .
	bitfield Signed: no Width: 4 Offset: 0 Minimum: 0 Maximum: 15 Scaling 1, 0, 0 (A,B,C): Scaled value: $1 * 10^0 * (Raw + 0)$ Resolution: 1
digit16	Digit 16 (hexadecimal digit) .
	bitfield Signed: no Width: 4 Offset: 4 Minimum: 0 Maximum: 15 Scaling 1, 0, 0 (A,B,C): Scaled value: $1 * 10^0 * (Raw + 0)$ Resolution: 1
digit17	Digit 17 (hexadecimal digit) .
	bitfield Signed: no Width: 4 Offset: 0 Minimum: 0 Maximum: 15 Scaling 1, 0, 0 (A,B,C): Scaled value: $1 * 10^0 * (Raw + 0)$ Resolution: 1
digit18	Digit 18 (hexadecimal digit) .

	bitfield Signed: <i>no</i> Width: <i>4</i> Offset: <i>4</i> Minimum: <i>0</i> Maximum: <i>15</i> Scaling <i>1, 0, 0</i> (A,B,C): Scaled $1 * 10^0 * (Raw + 0)$ value: Resolution: <i>1</i>
digit19	Digit 19 (hexadecimal digit) .
	bitfield Signed: <i>no</i> Width: <i>4</i> Offset: <i>0</i> Minimum: <i>0</i> Maximum: <i>15</i> Scaling <i>1, 0, 0</i> (A,B,C): Scaled $1 * 10^0 * (Raw + 0)$ value: Resolution: <i>1</i>
digit20	Digit 20 (hexadecimal digit) .
	bitfield Signed: <i>no</i> Width: <i>4</i> Offset: <i>4</i> Minimum: <i>0</i> Maximum: <i>15</i> Scaling <i>1, 0, 0</i> (A,B,C): Scaled $1 * 10^0 * (Raw + 0)$ value: Resolution: <i>1</i>
digit21	Digit 21 (hexadecimal digit) .
	bitfield Signed: <i>no</i> Width: <i>4</i> Offset: <i>0</i> Minimum: <i>0</i> Maximum: <i>15</i> Scaling <i>1, 0, 0</i> (A,B,C): Scaled $1 * 10^0 * (Raw + 0)$ value:

	Resolution: <i>1</i>
digit22	Digit 22 (hexadecimal digit) .
	bitfield Signed: <i>no</i> Width: <i>4</i> Offset: <i>4</i> Minimum: <i>0</i> Maximum: <i>15</i> Scaling <i>1, 0, 0</i> (A,B,C): Scaled <i>$1 * 10^0 * (Raw + 0)$</i> value: Resolution: <i>1</i>
digit23	Digit 23 (hexadecimal digit) .
	bitfield Signed: <i>no</i> Width: <i>4</i> Offset: <i>0</i> Minimum: <i>0</i> Maximum: <i>15</i> Scaling <i>1, 0, 0</i> (A,B,C): Scaled <i>$1 * 10^0 * (Raw + 0)$</i> value: Resolution: <i>1</i>
digit24	Digit 24 (hexadecimal digit) .
	bitfield Signed: <i>no</i> Width: <i>4</i> Offset: <i>4</i> Minimum: <i>0</i> Maximum: <i>15</i> Scaling <i>1, 0, 0</i> (A,B,C): Scaled <i>$1 * 10^0 * (Raw + 0)$</i> value: Resolution: <i>1</i>
digit25	Digit 25 (hexadecimal digit) .
	bitfield Signed: <i>no</i> Width: <i>4</i> Offset: <i>0</i> Minimum: <i>0</i> Maximum: <i>15</i> Scaling <i>1, 0, 0</i>

	(A,B,C): Scaled $1 * 10^0 * (Raw+0)$ value: Resolution: 1
digit26	Digit 26 (hexadecimal digit) .
	bitfield Signed: <i>no</i> Width: 4 Offset: 4 Minimum: 0 Maximum: 15 Scaling $1, 0, 0$ (A,B,C): Scaled $1 * 10^0 * (Raw+0)$ value: Resolution: 1
digit27	Digit 27 (hexadecimal digit) .
	bitfield Signed: <i>no</i> Width: 4 Offset: 0 Minimum: 0 Maximum: 15 Scaling $1, 0, 0$ (A,B,C): Scaled $1 * 10^0 * (Raw+0)$ value: Resolution: 1
digit28	Digit 28 (hexadecimal digit) .
	bitfield Signed: <i>no</i> Width: 4 Offset: 4 Minimum: 0 Maximum: 15 Scaling $1, 0, 0$ (A,B,C): Scaled $1 * 10^0 * (Raw+0)$ value: Resolution: 1
digit29	Digit 29 (hexadecimal digit) .
	bitfield Signed: <i>no</i> Width: 4 Offset: 0

	Minimum: <i>0</i> Maximum: <i>15</i> Scaling <i>1, 0, 0</i> (A,B,C): Scaled <i>1 * 10⁰ *(Raw+0)</i> value: Resolution: <i>1</i>
digit30	Digit 30 (hexadecimal digit) .
	bitfield Signed: <i>no</i> Width: <i>4</i> Offset: <i>4</i> Minimum: <i>0</i> Maximum: <i>15</i> Scaling <i>1, 0, 0</i> (A,B,C): Scaled <i>1 * 10⁰ *(Raw+0)</i> value: Resolution: <i>1</i>
digit31	Digit 31 (hexadecimal digit) .
	bitfield Signed: <i>no</i> Width: <i>4</i> Offset: <i>0</i> Minimum: <i>0</i> Maximum: <i>15</i> Scaling <i>1, 0, 0</i> (A,B,C): Scaled <i>1 * 10⁰ *(Raw+0)</i> value: Resolution: <i>1</i>
digit32	Digit 32 (hexadecimal digit) .
	bitfield Signed: <i>no</i> Width: <i>4</i> Offset: <i>4</i> Minimum: <i>0</i> Maximum: <i>15</i> Scaling <i>1, 0, 0</i> (A,B,C): Scaled <i>1 * 10⁰ *(Raw+0)</i> value: Resolution: <i>1</i>
digit33	Digit 33 (hexadecimal digit) .
	bitfield

	Signed: <i>no</i> Width: <i>4</i> Offset: <i>0</i> Minimum: <i>0</i> Maximum: <i>15</i> Scaling <i>1, 0, 0</i> (A,B,C): Scaled <i>1 * 10⁰ * (Raw+0)</i> value: Resolution: <i>1</i>
digit34	Digit 34 (hexadecimal digit) .
	bitfield Signed: <i>no</i> Width: <i>4</i> Offset: <i>4</i> Minimum: <i>0</i> Maximum: <i>15</i> Scaling <i>1, 0, 0</i> (A,B,C): Scaled <i>1 * 10⁰ * (Raw+0)</i> value: Resolution: <i>1</i>
digit35	Digit 35 (hexadecimal digit) .
	bitfield Signed: <i>no</i> Width: <i>4</i> Offset: <i>0</i> Minimum: <i>0</i> Maximum: <i>15</i> Scaling <i>1, 0, 0</i> (A,B,C): Scaled <i>1 * 10⁰ * (Raw+0)</i> value: Resolution: <i>1</i>
digit36	Digit 36 (hexadecimal digit) .
	bitfield Signed: <i>no</i> Width: <i>4</i> Offset: <i>4</i> Minimum: <i>0</i> Maximum: <i>15</i> Scaling <i>1, 0, 0</i> (A,B,C): Scaled <i>1 * 10⁰ * (Raw+0)</i> value: Resolution: <i>1</i>

digit37	Digit 37 (hexadecimal digit) .
	bitfield Signed: <i>no</i> Width: <i>4</i> Offset: <i>0</i> Minimum: <i>0</i> Maximum: <i>15</i> Scaling <i>1, 0, 0</i> (A,B,C): Scaled <i>1 * 10⁰ *(Raw+0)</i> value: Resolution: <i>1</i>
digit38	Digit 38 (hexadecimal digit) .
	bitfield Signed: <i>no</i> Width: <i>4</i> Offset: <i>4</i> Minimum: <i>0</i> Maximum: <i>15</i> Scaling <i>1, 0, 0</i> (A,B,C): Scaled <i>1 * 10⁰ *(Raw+0)</i> value: Resolution: <i>1</i>
digit39	Digit 39 (hexadecimal digit) .
	bitfield Signed: <i>no</i> Width: <i>4</i> Offset: <i>0</i> Minimum: <i>0</i> Maximum: <i>15</i> Scaling <i>1, 0, 0</i> (A,B,C): Scaled <i>1 * 10⁰ *(Raw+0)</i> value: Resolution: <i>1</i>
digit40	Digit 40 (hexadecimal digit) .
	bitfield Signed: <i>no</i> Width: <i>4</i> Offset: <i>4</i> Minimum: <i>0</i> Maximum: <i>15</i> Scaling <i>1, 0, 0</i> (A,B,C): Scaled

[illegible]

SNVT_magcard#LO: text("%d", digit1, digit2, digit3, digit4, digit5, digit6, digit7, digit8, digit9, digit10, digit11, digit12, digit13, digit14, digit15, digit16, digit17, digit18, digit19, digit20, digit21, digit22, digit23, digit24, digit25, digit26, digit27, digit28, digit29, digit30, digit31, digit32, digit33, digit34, digit35, digit36, digit37, digit38, digit39, digit40)

SFPIdentifierSensor

$$SNVT_mass\#US: \text{text}("%f", *0.03527+0(0.845))$$

SNVT_mass_f

Overview:

Mass (grams) .

Details:

Standard:	<i>yes</i>
Resource Set:	<i>Standard 00:00:00:00:00:00:00:00-0</i>
Index:	<i>56</i>
Obsolete:	<i>no</i>
Size:	<i>4</i>
Programmatic Name:	<i>SNVT_mass_f</i>
Neuron C Type:	float
	Minimum: <i>0</i>
	Maximum: <i>3.4028234663853E+038</i>
Formats:	<i>SNVT_mass_f#SI: text("%f")</i>
	<i>SNVT_mass_f#US: text("%f", *0.002205+0(0:846))</i>
	<i>SNVT_mass_f#US_oz: text("%f", *0.03527+0(0:845))</i>

SNVT_mass_kilo

Overview:

Mass (kilograms) .

Details:

Standard:	<i>yes</i>
Resource Set:	<i>Standard 00:00:00:00:00:00:00:00-0</i>
Index:	<i>24</i>
Obsolete:	<i>no</i>
Size:	<i>2</i>
Programmatic Name:	<i>SNVT_mass_kilo</i>
Neuron C Type:	unsigned long
	Minimum: <i>0</i>
	Maximum: <i>65535</i>
	Scaling (A,B,C): <i>1, -1, 0</i>

Scaled value: $1 * 10^{-1} * (Raw + 0)$ Resolution: 0.1 Formats: *SNVT_mass_kilo#SI: text("%f")**SNVT_mass_kilo#US: text("%f", *2.20462+0(0:846))*

SNVT_mass_mega

Overview:

Mass (metric tons) .

Details:

Standard: *yes*Resource Set: *Standard 00:00:00:00:00:00:00:00-0*Index: *25*Obsolete: *no*Size: *2*Programmatic Name: *SNVT_mass_mega*Neuron C Type: *unsigned long*Minimum: *0*Maximum: *65535*Scaling (A,B,C): *1, -1, 0*Scaled value: $1 * 10^{-1} * (Raw + 0)$ Resolution: 0.1 Formats: *SNVT_mass_mega#SI: text("%f")**SNVT_mass_mega#US: text("%f", *1.1023+0(0:849))*

SNVT_mass_mil

Overview:

Mass (milligrams) .

Details:

Standard: *yes*

Resource Set:	<i>Standard 00:00:00:00:00:00:00:00-0</i>
Index:	<i>26</i>
Obsolete:	<i>no</i>
Size:	<i>2</i>
Programmatic Name:	<i>SNVT_mass_mil</i>
Neuron C Type:	unsigned long
	Minimum: <i>0</i>
	Maximum: <i>65535</i>
	Scaling (A,B,C): <i>1, -1, 0</i>
	Scaled value: <i>1 * 10⁻¹ * (Raw + 0)</i>
	Resolution: <i>0.1</i>
Formats:	<i>SNVT_mass_mil#SI: text("%f")</i>
	<i>SNVT_mass_mil#US: text("%f", *3.527e-5+0(0:845))</i>

SNVT_motor_state

Overview:

Motor state (motor state names) .

Defines the state of a mechanical motor; whether pump, fan, or other. A network variable using this SNVT does not explicitly define whether the motor's drive is enabled, or whether any brake (decelerator) is actively slowing the motor.

Details:

Standard:	<i>yes</i>
Resource Set:	<i>Standard 00:00:00:00:00:00:00:00-0</i>
Index:	<i>155</i>
Obsolete:	<i>no</i>
Size:	<i>1</i>
Programmatic Name:	<i>SNVT_motor_state</i>
Neuron C Type:	<i>motor_state_t</i>
Formats:	<i>SNVT_motor_state: text("%m")</i>

SNVT_muldiv

Overview:

Multiply/Divide (multiplier, divisor) . Gain factor

May be used as a gain factor for fixed-point sensor objects. It is compatible with the Neuron C `muldiv()` function, which provides a 16 x 16 unsigned multiplication with a 32-bit intermediate result, followed by a 32/16 unsigned division with a 16-bit end result. For more details, see the *Neuron C Reference Guide*.

Details:

Standard:	<i>yes</i>
Resource Set:	<i>Standard 00:00:00:00:00:00:00:00-0</i>
Index:	<i>91</i>
Obsolete:	<i>no</i>
Size:	<i>4</i>
Programmatic Name:	<i>SNVT_muldiv</i>
Neuron C Type:	

Structure	
multiplier	Multiplier (16-bit unsigned value) .
	unsigned long Minimum: <i>0</i> Maximum: <i>65535</i> Scaling <i>1, 0, 0</i> (A,B,C): Scaled <i>$1 * 10^0 * (Raw + 0)$</i> value: Resolution: <i>1</i>
divisor	Divisor (16-bit unsigned value) .
	unsigned long Minimum: <i>1</i> Maximum: <i>65535</i> Invalid: <i>0</i> Scaling <i>1, 0, 0</i> (A,B,C): Scaled <i>$1 * 10^0 * (Raw + 0)$</i> value: Resolution: <i>1</i>

Formats:	<i>SNVT_muldiv: text("%d/%d", multiplier, divisor)</i>
Used by:	<i>SCPTgain SCPTpulseValue</i>

SNVT_multiplier

Overview:

Multiplier (16-bit unsigned value) .

Details:

Standard:	<i>yes</i>
Resource Set:	<i>Standard 00:00:00:00:00:00:00:00-0</i>
Index:	<i>82</i>
Obsolete:	<i>no</i>
Size:	<i>2</i>
Programmatic Name:	<i>SNVT_multiplier</i>
Neuron C Type:	unsigned long
	Minimum: <i>0</i>
	Maximum: <i>65535</i>
	Scaling (A,B,C): <i>5, -4, 0</i>
	Scaled value: $5 * 10^{-4} * (Raw + 0)$
	Resolution: <i>0.0005</i>
Formats:	<i>SNVT_multiplier: text("%f")</i>
Used by:	<i>SCPTgainVAV SCPTgainVAVHeat SCPTsensConstTmp SCPTsensConstVAV</i>

SNVT_multiplier_s

Overview:

Multiplier. Value multiplier

Details:

Standard:	<i>yes</i>
Resource Set:	<i>Standard 00:00:00:00:00:00:00:00-0</i>
Index:	<i>188</i>
Obsolete:	<i>no</i>
Size:	<i>1</i>
Programmatic Name:	<i>SNVT_multiplier_s</i>
Neuron C Type:	unsigned short
	Minimum: <i>0</i>
	Maximum: <i>254</i>
	Invalid: <i>255</i>
	Scaling (A,B,C): <i>1, -2, 0</i>
	Scaled value: $1 * 10^{-2} * (Raw + 0)$
	Resolution: <i>0.01</i>

Used by: *SFPTisiLampActuator SNVT_switch_2*

SNVT_nv_type

Overview:

Network variable type. Network variable type description for network variables that support changeable types

This SNVT specifies the type of a network variable. A SNVT_nv_type network variable can be used to create a configuration network variable that specifies the type of a second network variable. The configuration network variable must be declared as a SCPTnvType configuration property. A SCPTnvType configuration property may also be declared within a configuration file, not using the SNVT_nv_type type. See the SCPTnvType description for further details on using this SCPT.

The type_scope field contains the scope of the resource file that defines the specified type. The scope may be any of the following values:

- 0 = Standard – applies to all devices.
- 1 = Reserved for future use.
- 2 = Reserved for future use.
- 3 = Manufacturer – applies to all devices from the manufacturer specified in the program ID template.
- 4 = Manufacturer and Device Class – applies to all devices from the manufacturer with the device class specified in the program ID template.
- 5 = Manufacturer, Device Class, and Device Subclass – applies to all devices from the manufacturer with the device class and device subclass specified in the program ID template.
- 6 = Manufacturer, Device Class, Device Subclass, and Device Model – applies to all devices of the specified type and manufacturer specified in the program ID template.

Details:

Standard: *yes*

Resource Set: *Standard 00:00:00:00:00:00:00:00-0*

Index: *166*

Obsolete: *no*

Size: *19*

Programmatic Name: *SNVT_nv_type*

Neuron C Type:

Structure	
type_program_ID	Type program ID. Program ID template of the resource file containing the network variable type definition
	unsigned short [8] Minimum: 0 Maximum: 255

	Scaling $1, 0, 0$ (A,B,C): Scaled $1 * 10^0 * (Raw + 0)$ value: Resolution: 1
type_scope	Type scope (file scope) . Scope of the resource file containing the network variable type definition
	unsigned short Minimum: 0 Maximum: 6 Scaling $1, 0, 0$ (A,B,C): Scaled $1 * 10^0 * (Raw + 0)$ value: Resolution: 1
type_index	Type index (type index) . Index within the specified resource file of the network variable type definition
	unsigned long Minimum: 1 Maximum: 65535 Invalid: 0 Scaling $1, 0, 0$ (A,B,C): Scaled $1 * 10^0 * (Raw + 0)$ value: Resolution: 1
type_category	Type category (type category names) . Type category of the network variable type
	<i>nv_type_category_t</i>
type_length	Type length (bytes) . Length of the network variable type
	unsigned short Minimum: 1 Maximum: 31 Invalid: 0 Scaling $1, 0, 0$ (A,B,C): Scaled $1 * 10^0 * (Raw + 0)$ value: Resolution: 1
scaling_factor_a	Scaling factor a. Scaling multiplier 'a' where ScaledValue = $a * (10^{**b}) * (RawValue + c)$
	signed long Minimum: -32768 Maximum: 32767 Invalid: 32767

	Scaling <i>1, 0, 0</i> (A,B,C): Scaled $1 * 10^0 * (Raw+0)$ value: Resolution: <i>1</i>
scaling_factor_b	Scaling factor b. Exponent 'b' where ScaledValue = $a * (10^{**b}) * (RawValue+c)$
	signed long Minimum: <i>-32768</i> Maximum: <i>32767</i> Invalid: <i>32767</i> Scaling <i>1, 0, 0</i> (A,B,C): Scaled $1 * 10^0 * (Raw+0)$ value: Resolution: <i>1</i>
scaling_factor_c	Scaling Factor c. Offset 'c' where ScaledValue = $a * (10^{**b}) * (RawValue+c)$
	signed long Minimum: <i>-32768</i> Maximum: <i>32767</i> Invalid: <i>32767</i> Scaling <i>1, 0, 0</i> (A,B,C): Scaled $1 * 10^0 * (Raw+0)$ value: Resolution: <i>1</i>

Formats: *SNVT_nv_type: text("PID %x:%x:%x:%x:%x:%x:%x:%x, Scope %d, Index %d, %m, %d bytes, A=%d, B=%d, C=%d", type_program_ID[0], type_program_ID[1], type_program_ID[2], type_program_ID[3], type_program_ID[4], type_program_ID[5], type_program_ID[6], type_program_ID[7], type_scope, type_index, type_category, type_length, scaling_factor_a, scaling_factor_b, scaling_factor_c)*

Used by: *SCPTnvDynamicAssignment SCPTnvType*

SNVT_obj_request

Overview:

Object request (ID, request) .

Allows a functional block to be placed in one of several functional modes. For more details, see the definition of the Node Object (SFPTnodeObject) functional profile.

Additions found in version 3.3 and later:

The RQ_CLEAR_RESET request clears the reset_complete flag in SNVT_obj_status (nvoStatus in the Node Object) of the corresponding Object (object_id). Further requests have no effect, until the next Reset sequence has again been executed.

The RQ_RESET request initiates the Reset sequence in SNVT_obj_status (nvoStatus in the Node Object) of the corresponding object (object_id) every time that it is sent. The reset_complete flag (SNVT_obj_status) is set when the Reset sequence is complete, and the flag must be cleared by RQ_CLEAR_RESET (SNVT_obj_request).

The existing RQ_CLEAR_STATUS and RQ_CLEAR_ALARM functions (SNVT_obj_request) remain unchanged.

Details:

Standard:	<i>yes</i>
Resource Set:	<i>Standard 00:00:00:00:00:00:00:00-0</i>
Index:	<i>92</i>
Obsolete:	<i>no</i>
Size:	<i>3</i>
Programmatic Name:	<i>SNVT_obj_request</i>
Neuron C Type:	

Structure	
object_id	Object ID (object index) .
	unsigned long Minimum: 0 Maximum: 65535 Scaling 1, 0, 0 (A,B,C): Scaled $1 * 10^0 * (Raw + 0)$ value: Resolution: 1
object_request	Object request (object request names) .
	<i>object_request_t</i>

Formats: *SNVT_obj_request: text("%d,%m", object_id, object_request)*

SNVT_obj_request#LO: text("%d|m", object_id, object_request)

Used by: *SFPTnodeObject*

SNVT_obj_status

Overview:

Object status (ID, status flags) .

Used to indicate the status of the various objects within a node. For more details, see the definition

of the Node Object (SFPTnodeObject).

Addition found in version 3.3 and later:

The reset_complete field, indicates the execution of the Reset sequence of any object (object_id) within the device. After a Reset sequence, the reset_complete flag goes to TRUE (1) and it remains '1' until it is cleared (acknowledged) via SNVT_obj_request (nviRequest in the Node Object) on in the corresponding Object (object_id).

Note: The additional reset flag uses reserved1 of the previous SNVT_obj_status structure definition.

Details:

Standard: *yes*
 Resource Set: *Standard 00:00:00:00:00:00:00:00-0*
 Index: *93*
 Obsolete: *no*
 Size: *6*
 Programmatic Name: *SNVT_obj_status*

Neuron C Type:	Structure												
	<table> <tr> <th>object_id</th><th>Object ID (object index) .</th></tr> <tr> <td></td><td> unsigned long Minimum: <i>0</i> Maximum: <i>65535</i> Scaling <i>1, 0, 0</i> (A,B,C): Scaled <i>$1 * 10^0 * (Raw + 0)$</i> value: Resolution: <i>1</i> </td></tr> <tr> <th>invalid_id</th><th>Invalid-ID flag (boolean) .</th></tr> <tr> <td></td><td> bitfield Signed: <i>no</i> Width: <i>1</i> Offset: <i>0</i> Minimum: <i>0</i> Maximum: <i>1</i> Scaling <i>1, 0, 0</i> (A,B,C): Scaled <i>$1 * 10^0 * (Raw + 0)$</i> value: Resolution: <i>1</i> </td></tr> <tr> <th>invalid_request</th><th>Invalid-request flag (boolean) .</th></tr> <tr> <td></td><td> bitfield Signed: <i>no</i> Width: <i>1</i> Offset: <i>1</i> </td></tr> </table>	object_id	Object ID (object index) .		unsigned long Minimum: <i>0</i> Maximum: <i>65535</i> Scaling <i>1, 0, 0</i> (A,B,C): Scaled <i>$1 * 10^0 * (Raw + 0)$</i> value: Resolution: <i>1</i>	invalid_id	Invalid-ID flag (boolean) .		bitfield Signed: <i>no</i> Width: <i>1</i> Offset: <i>0</i> Minimum: <i>0</i> Maximum: <i>1</i> Scaling <i>1, 0, 0</i> (A,B,C): Scaled <i>$1 * 10^0 * (Raw + 0)$</i> value: Resolution: <i>1</i>	invalid_request	Invalid-request flag (boolean) .		bitfield Signed: <i>no</i> Width: <i>1</i> Offset: <i>1</i>
object_id	Object ID (object index) .												
	unsigned long Minimum: <i>0</i> Maximum: <i>65535</i> Scaling <i>1, 0, 0</i> (A,B,C): Scaled <i>$1 * 10^0 * (Raw + 0)$</i> value: Resolution: <i>1</i>												
invalid_id	Invalid-ID flag (boolean) .												
	bitfield Signed: <i>no</i> Width: <i>1</i> Offset: <i>0</i> Minimum: <i>0</i> Maximum: <i>1</i> Scaling <i>1, 0, 0</i> (A,B,C): Scaled <i>$1 * 10^0 * (Raw + 0)$</i> value: Resolution: <i>1</i>												
invalid_request	Invalid-request flag (boolean) .												
	bitfield Signed: <i>no</i> Width: <i>1</i> Offset: <i>1</i>												

	Minimum: 0 Maximum: 1 Scaling $1, 0, 0$ (A,B,C): Scaled value: $1 * 10^0 * (Raw + 0)$ Resolution: 1
disabled	Disabled flag (boolean) .
	bitfield Signed: no Width: 1 Offset: 2 Minimum: 0 Maximum: 1 Scaling $1, 0, 0$ (A,B,C): Scaled value: $1 * 10^0 * (Raw + 0)$ Resolution: 1
out_of_limits	Out-of-limits flag (boolean) .
	bitfield Signed: no Width: 1 Offset: 3 Minimum: 0 Maximum: 1 Scaling $1, 0, 0$ (A,B,C): Scaled value: $1 * 10^0 * (Raw + 0)$ Resolution: 1
open_circuit	Open-circuit flag (boolean) .
	bitfield Signed: no Width: 1 Offset: 4 Minimum: 0 Maximum: 1 Scaling $1, 0, 0$ (A,B,C): Scaled value: $1 * 10^0 * (Raw + 0)$ Resolution: 1
out_of_service	Out-of-service flag (boolean) .
	bitfield

	Signed: <i>no</i> Width: <i>1</i> Offset: <i>5</i> Minimum: <i>0</i> Maximum: <i>1</i> Scaling (A,B,C): <i>1, 0, 0</i> Scaled value: $1 * 10^0 * (Raw + 0)$ Resolution: <i>1</i>
mechanical_fault	Mechanical-fault flag (boolean) .
	bitfield Signed: <i>no</i> Width: <i>1</i> Offset: <i>6</i> Minimum: <i>0</i> Maximum: <i>1</i> Scaling (A,B,C): <i>1, 0, 0</i> Scaled value: $1 * 10^0 * (Raw + 0)$ Resolution: <i>1</i>
feedback_failure	Feedback-failure flag (boolean) .
	bitfield Signed: <i>no</i> Width: <i>1</i> Offset: <i>7</i> Minimum: <i>0</i> Maximum: <i>1</i> Scaling (A,B,C): <i>1, 0, 0</i> Scaled value: $1 * 10^0 * (Raw + 0)$ Resolution: <i>1</i>
over_range	Over-range flag (boolean) .
	bitfield Signed: <i>no</i> Width: <i>1</i> Offset: <i>0</i> Minimum: <i>0</i> Maximum: <i>1</i> Scaling (A,B,C): <i>1, 0, 0</i> Scaled value: $1 * 10^0 * (Raw + 0)$ Resolution: <i>1</i>

under_range	Under-range flag (boolean) .
	bitfield Signed: <i>no</i> Width: <i>1</i> Offset: <i>1</i> Minimum: <i>0</i> Maximum: <i>1</i> Scaling <i>1, 0, 0</i> (A,B,C): Scaled value: $1 * 10^0 * (Raw + 0)$ Resolution: <i>1</i>
electrical_fault	Electrical-fault flag (boolean) .
	bitfield Signed: <i>no</i> Width: <i>1</i> Offset: <i>2</i> Minimum: <i>0</i> Maximum: <i>1</i> Scaling <i>1, 0, 0</i> (A,B,C): Scaled value: $1 * 10^0 * (Raw + 0)$ Resolution: <i>1</i>
unable_to_measure	Unable-to-measure flag (boolean) .
	bitfield Signed: <i>no</i> Width: <i>1</i> Offset: <i>3</i> Minimum: <i>0</i> Maximum: <i>1</i> Scaling <i>1, 0, 0</i> (A,B,C): Scaled value: $1 * 10^0 * (Raw + 0)$ Resolution: <i>1</i>
comm_failure	Communications-failure flag (boolean) .
	bitfield Signed: <i>no</i> Width: <i>1</i> Offset: <i>4</i> Minimum: <i>0</i> Maximum: <i>1</i> Scaling <i>1, 0, 0</i> (A,B,C): Scaled

	value: $1 * 10^0 * (Raw + 0)$ Resolution: 1
fail_self_test	Failed-self-test flag (boolean) .
	bitfield Signed: <i>no</i> Width: 1 Offset: 5 Minimum: 0 Maximum: 1 Scaling $1, 0, 0$ (A,B,C): Scaled value: $1 * 10^0 * (Raw + 0)$ Resolution: 1
self_test_in_progress	Self-test-in-progress flag (boolean) .
	bitfield Signed: <i>no</i> Width: 1 Offset: 6 Minimum: 0 Maximum: 1 Scaling $1, 0, 0$ (A,B,C): Scaled value: $1 * 10^0 * (Raw + 0)$ Resolution: 1
locked_out	Locked-out flag (boolean) .
	bitfield Signed: <i>no</i> Width: 1 Offset: 7 Minimum: 0 Maximum: 1 Scaling $1, 0, 0$ (A,B,C): Scaled value: $1 * 10^0 * (Raw + 0)$ Resolution: 1
manual_control	Manual-control flag (boolean) .
	bitfield Signed: <i>no</i> Width: 1 Offset: 0 Minimum: 0 Maximum: 1

	Scaling $1, 0, 0$ (A,B,C): Scaled value: $1 * 10^0 * (Raw + 0)$ Resolution: 1
in_alarm	Input-alarm flag (boolean) .
	bitfield Signed: <i>no</i> Width: 1 Offset: 1 Minimum: 0 Maximum: 1 Scaling $1, 0, 0$ (A,B,C): Scaled value: $1 * 10^0 * (Raw + 0)$ Resolution: 1
in_override	Input-override flag (boolean) .
	bitfield Signed: <i>no</i> Width: 1 Offset: 2 Minimum: 0 Maximum: 1 Scaling $1, 0, 0$ (A,B,C): Scaled value: $1 * 10^0 * (Raw + 0)$ Resolution: 1
report_mask	Report-mask flag (boolean) .
	bitfield Signed: <i>no</i> Width: 1 Offset: 3 Minimum: 0 Maximum: 1 Scaling $1, 0, 0$ (A,B,C): Scaled value: $1 * 10^0 * (Raw + 0)$ Resolution: 1
programming_mode	Programming-mode flag (boolean) .
	bitfield Signed: <i>no</i> Width: 1

	Offset: 4 Minimum: 0 Maximum: 1 Scaling 1, 0, 0 (A,B,C): Scaled value: $1 * 10^0 * (Raw + 0)$ Resolution: 1
programming_fail	Programming-fail flag (boolean) .
	bitfield Signed: no Width: 1 Offset: 5 Minimum: 0 Maximum: 1 Scaling 1, 0, 0 (A,B,C): Scaled value: $1 * 10^0 * (Raw + 0)$ Resolution: 1
alarm_notify_disabled	Alarm-notify-disabled flag (boolean) .
	bitfield Signed: no Width: 1 Offset: 6 Minimum: 0 Maximum: 1 Scaling 1, 0, 0 (A,B,C): Scaled value: $1 * 10^0 * (Raw + 0)$ Resolution: 1
reset_complete	Reset (boolean) .
	bitfield Signed: no Width: 1 Offset: 7 Minimum: 0 Maximum: 1 Scaling 1, 0, 0 (A,B,C): Scaled value: $1 * 10^0 * (Raw + 0)$ Resolution: 1
reserved2	reserved2. This field is reserved.

SNVT_obj_status#LO: *text("%d %d %d %d %d %d %d %d %d %d %d %d
%d %d %d %d %d %d %d %d %d %d %d %d %d", object_id,
invalid_id, invalid_request, disabled, out_of_limits, open_circuit,
out_of_service, mechanical_fault, feedback_failure, over_range,
under_range, electrical_fault, unable_to_measure, comm_failure,
fail_self_test, self_test_in_progress, locked_out, manual_control,
in_alarm, in_override, report_mask, programming_mode,
programming_fail, alarm_notify_disabled, reset_complete)*

SNVT_occupancy

Overview:

Details:

Standard:	<i>yes</i>
Resource Set:	<i>Standard 00:00:00:00:00:00:00:00-0</i>
Index:	<i>109</i>
Obsolete:	<i>no</i>
Size:	<i>1</i>
Programmatic Name:	<i>SNVT_occupancy</i>
Neuron C Type:	<i>occup t</i>

Formats: *SNVT_occupancy: text("%m")*

Used by: *SFPTchilledCeilingController SFPTdischargeAirController
SFPTfanCoilUnit SFPTheatPump SFPTisiLampActuator
SFPTisiOccupancySensor SFPTisiSunblindActuator
SFPToccupancyController SFPToccupancySensor SFPTroofTopUnit
SFPTsccAHU SFPTsccChilledCeiling SFPTsccCommandModule
SFPTsccFanCoil SFPTsccHeatPump SFPTsccRadiator
SFPTsccRooftop SFPTsccSelfContained SFPTsccUnitVentilator
SFPTsccVAV SFPTspaceComfortController SFPTsunblindController
SFPTthermostat SFPTunitVentilatorController
SFPTvariableAirVolume*

SNVT_override

Overview:

Override code (override code names) .

Details:

Standard: *yes*

Resource Set: *Standard 00:00:00:00:00:00:00:00-0*

Index: *97*

Obsolete: *no*

Size: *1*

Programmatic Name: *SNVT_override*

Neuron C Type: *override_t*

Formats: *SNVT_override: text("%m")*

Used by: *SCPTovrBehave*

SNVT_ph

Overview:

Acidity (pH) . Ratio of concentration of ions

Ratio of concentration of ions.

Details:

Standard: *yes*

Resource Set: *Standard 00:00:00:00:00:00:00:00-0*
 Index: *125*
 Obsolete: *no*
 Size: *2*
 Programmatic Name: *SNVT_ph*
 Neuron C Type: signed long
 Minimum: *-32768*
 Maximum: *32767*
 Scaling (A,B,C): *1, -3, 0*
 Scaled value: *$1 * 10^{-3} * (Raw + 0)$*
 Resolution: *0.001*
 Formats: *SNVT_ph: text("%f")*

SNVT_ph_f

Overview:

Acidity (pH) . Ratio of concentration of ions

Ratio of concentration of ions.

Details:

Standard: *yes*
 Resource Set: *Standard 00:00:00:00:00:00:00:00-0*
 Index: *126*
 Obsolete: *no*
 Size: *4*
 Programmatic Name: *SNVT_ph_f*
 Neuron C Type: float
 Minimum: *-3.4028234663853E+038*
 Maximum: *3.4028234663853E+038*
 Formats: *SNVT_ph_f: text("%f")*

SNVT_pos_ctrl

Overview:

Position control (receiver, controller ID, controller priority, function, action, value) .

Used for the programmable-position camera-command interface.

The command interface consists of three different command functions:

- Programmable camera-position presets/prepositions (CMF_REL),
- Programmable tables of preposition tours (CMF_TOUR), and
- Absolute positions for pan and tilt, in degrees (CMF_ABS).

The three command functions each have a set of actions, and arguments.

The programmable table of prepositions, a preposition tour table, can be called by number. A network tool configures the preposition tour table. The programmable prepositions are used by:

- Saving a preposition by number (CMA_SAV), and
- Recalling (going to) a previously saved preposition by number (CMA_CALL).

The absolute position can be:

- Read for the current position (CMA_READ), or
- Invoked (set) by using the 'write' command.

Each command input may have a response output, giving the status of the command.

If any illegal command function has been used, the response status is CMF_NUL (cam_func_t).

If any legal command function has been used, with an illegal action, the response status is CMA_NUL (cam_action_t).

The SNVT_pos_ctrl may be used also in a control-arbitration scenario. The receiver_id is used to refer to one particular device in a group of equal, controllable devices.

The controller_id is used to identify the requesting device from a group of equal, controlling devices. The controller_prio is used for the control arbitration in the controllable device. Standard control devices (e.g., keyboards) in normal control mode must have the priority assigned in the range 1 to 50. The range 51 to 200 is used in alarm control mode. The priority value zero (0) is used to release control.

EXAMPLES

Example 1, call of a preposition

Physical value controller #2, having priority 20, requests camera telemetry receiver #1, to go to (recall), a relative position, preposition #4.

nvoPositionCtrl.controller_id	=	2
nvoPositionCtrl.controller_prio	=	20
nvoPositionCtrl.receiver_id	=	1
nvoPositionCtrl.action	=	CMA_CALL
nvoPositionCtrl.function	=	CMF_REL
nvoPositionCtrl.value.number	=	4

Example 2, call of a preposition tour table

Physical value controller #2, having priority 20, requests camera telemetry receiver #1, to start (recall) a preposition tour, preposition tour #2.

```
nvoPositionCtrl.controller_id      = 2
nvoPositionCtrl.controller_prio    = 20
nvoPositionCtrl.receiver_id        = 1
nvoPositionCtrl.action              = CMA_CALL
nvoPositionCtrl.function            = CMF_TOUR
nvoPositionCtrl.value.number        = 2
```

Example 3, read current position as absolute values

Physical value controller #2, having priority 20, requests camera telemetry receiver #1, to retrieve (read), absolute positions of pan, tilt, and zoom.

```
nvoPositionCtrl.controller_id      = 2
nvoPositionCtrl.controller_prio    = 20
nvoPositionCtrl.receiver_id        = 1
nvoPositionCtrl.action              = CMA_READ
nvoPositionCtrl.function            = CMF_ABS
```

Example 4, call of a position as absolute values

Physical value controller #2, having priority 20, requests camera telemetry receiver #1, to go to an absolute position, defined by values of pan, tilt, and zoom.

```
nvoPositionCtrl.controller_id      = 2
nvoPositionCtrl.controller_prio    = 20
nvoPositionCtrl.receiver_id        = 1
nvoPositionCtrl.action              = CMA_CALL
nvoPositionCtrl.function            = CMF_ABS
nvoPositionCtrl.value.abspos.pan    = _____
nvoPositionCtrl.value.abspos.tilt   = _____
nvoPositionCtrl.value.abspos.zoom   = _____
```

Details:

Standard: *yes*
 Resource Set: *Standard 00:00:00:00:00:00:00:00-0*
 Index: *152*
 Obsolete: *no*
 Size: *13*
 Programmatic Name: *SNVT_pos_ctrl*

Neuron C Type:

Structure	
receiver_id	Receiver ID (ID number) .
	unsigned long

	Minimum: 0 Maximum: 65535 Invalid: 0 Scaling 1, 0, 0 (A,B,C): Scaled value: $1 * 10^0 * (Raw + 0)$ Resolution: 1								
controller_id	Controller ID (ID number) .								
	unsigned long Minimum: 0 Maximum: 65535 Invalid: 0 Scaling 1, 0, 0 (A,B,C): Scaled value: $1 * 10^0 * (Raw + 0)$ Resolution: 1								
controller_prio	Controller priority (priority value) .								
	unsigned short Minimum: 0 Maximum: 100 Scaling 1, 0, 0 (A,B,C): Scaled value: $1 * 10^0 * (Raw + 0)$ Resolution: 1								
function	Camera function (camera function names) .								
	<i>cam_func_t</i>								
action	Camera action (camera action names) .								
	<i>cam_act_t</i>								
value	Function value.								
	<table border="1"> <tr> <th colspan="2">Union</th></tr> <tr> <td>number</td><td>Action number (action number) .</td></tr> <tr> <td></td><td> unsigned short Minimum: 0 Maximum: 255 Invalid: 0 Scaling 1, 0, 0 (A,B,C): Scaled value: $1 * 10^0 * (Raw + 0)$ Resolution: 1 </td></tr> <tr> <td>abspos</td><td>Function absolute values (pan, tilt,</td></tr> </table>	Union		number	Action number (action number) .		unsigned short Minimum: 0 Maximum: 255 Invalid: 0 Scaling 1, 0, 0 (A,B,C): Scaled value: $1 * 10^0 * (Raw + 0)$ Resolution: 1	abspos	Function absolute values (pan, tilt,
Union									
number	Action number (action number) .								
	unsigned short Minimum: 0 Maximum: 255 Invalid: 0 Scaling 1, 0, 0 (A,B,C): Scaled value: $1 * 10^0 * (Raw + 0)$ Resolution: 1								
abspos	Function absolute values (pan, tilt,								

		zoom) .
		Structure
pan	Pan position.	
	<i>SNVT_angle_deg</i>	
tilt	Tilt position.	
	<i>SNVT_angle_deg</i>	
zoom	Zoom position.	
	<i>SNVT_lev_percent</i>	

Formats: *SNVT_pos_ctrl: text("%d %d %d %m %m %d %f %f %f", receiver_id, controller_id, controller_prio, function, action, value.number, value.abspos.pan, value.abspos.tilt, value.abspos.zoom)*

SNVT_power

Overview:

Power (Watts) .

Details:

Standard: *yes*
Resource Set: *Standard 00:00:00:00:00:00:00:00-0*
Index: *27*
Obsolete: *no*
Size: *2*
Programmatic Name: *SNVT_power*
Neuron C Type: *unsigned long*
Minimum: *0*
Maximum: *65535*
Scaling (A,B,C): *1, -1, 0*
Scaled value: *1 * 10⁻¹ * (Raw + 0)*
Resolution: *0.1*

Formats: *SNVT_power#SI: text("%f")*

*SNVT_power#US: text("%f", *3.4144+0(0:968))*

Used by: *SCPTmaxPower SCPTpwrSendOnDelta
SFPTchilledCeilingController SFPTfanCoilUnit
SFPTisiLampActuator SFPTisiSunblindActuator
SFPTpumpController SFPTsccAHU SFPTsccChilledCeiling
SFPTsccFanCoil SFPTsccHeatPump SFPTsccRadiator
SFPTsccRooftop SFPTsccSelfContained SFPTsccUnitVentilator*

SNVT_power_f

Overview:

Power (Watts) .

Details:

Standard:	<i>yes</i>
Resource Set:	<i>Standard 00:00:00:00:00:00:00:00-0</i>
Index:	<i>57</i>
Obsolete:	<i>no</i>
Size:	<i>4</i>
Programmatic Name:	<i>SNVT_power_f</i>
Neuron C Type:	<i>float</i> <i>Minimum: -3.4028234663853E+038</i> <i>Maximum: 3.4028234663853E+038</i>
Formats:	<i>SNVT_power_f#SI: text("%f")</i> <i>SNVT_power_f#US: text("%f", *3.4144+0(0:968))</i> <i>SNVT_power_f#US_hp: text("%f", *0.001341+0(0:969))</i> <i>SNVT_power_f#US_refr: text("%f", *2.8435e-4+0(0:967))</i>
Used by:	<i>SFPTautomaticTransferSwitch SFPTgeneratorSet</i>

SNVT_power_kilo

Overview:

Power (kiloWatts) .

Details:

Standard:	<i>yes</i>
Resource Set:	<i>Standard 00:00:00:00:00:00:00:00-0</i>
Index:	<i>28</i>
Obsolete:	<i>no</i>

Size: 2
 Programmatic Name: *SNVT_power_kilo*
 Neuron C Type: unsigned long
 Minimum: 0
 Maximum: 65535
 Scaling (A,B,C): 1, -1, 0
 Scaled value: $1 * 10^{-1} * (Raw + 0)$
 Resolution: 0.1
 Formats: *SNVT_power_kilo: text("%f")*
 Used by: *SFPTpumpController SFPTsccAHU SFPTsccChilledCeiling*
 SFPTsccFanCoil SFPTsccHeatPump SFPTsccRadiator
 SFPTsccRooftop SFPTsccSelfContained SFPTsccUnitVentilator
 SFPTsccVAV SFPTspaceComfortController
 SFPTunitVentilatorController SFPTvariableSpeedMotorDrive
 SNVT_pump_sensor

SNVT_ppm

Overview:

Concentration (ppm) .

Details:

Standard: *yes*
 Resource Set: *Standard 00:00:00:00:00:00:00:00-0*
 Index: 29
 Obsolete: *no*
 Size: 2
 Programmatic Name: *SNVT_ppm*
 Neuron C Type: unsigned long
 Minimum: 0
 Maximum: 65535
 Scaling (A,B,C): 1, 0, 0
 Scaled value: $1 * 10^0 * (Raw + 0)$
 Resolution: 1
 Formats: *SNVT_ppm: text("%d")*
 Used by: *SCPTlimitCO2 SCPTminDeltaCO2 SCPToffsetCO2*
 SFPTco2Sensor SFPTroofTopUnit SFPTsccAHU
 SFPTsccChilledCeiling SFPTsccCommandModule SFPTsccFanCoil
 SFPTsccHeatPump SFPTsccRadiator SFPTsccRooftop
 SFPTsccSelfContained SFPTsccUnitVentilator SFPTsccVAV
 SFPTspaceComfortController SFPTunitVentilatorController

SNVT_ppm_f

Overview:

Concentration (ppm) .

Details:

Standard:	<i>yes</i>
Resource Set:	<i>Standard 00:00:00:00:00:00:00:00-0</i>
Index:	<i>58</i>
Obsolete:	<i>no</i>
Size:	<i>4</i>
Programmatic Name:	<i>SNVT_ppm_f</i>
Neuron C Type:	<i>float</i>
	Minimum: <i>0</i>
	Maximum: <i>3.4028234663853E+038</i>
Formats:	<i>SNVT_ppm_f: text("%f")</i>
Used by:	<i>SFPTco2Sensor</i>

SNVT_preset

Overview:

Preset (mode, data, time) .

Network variables of SNVT_preset type are used to allow a sensor or actuator functional block to control and adopt one of several programmable values and ramp rates, in addition to the normal control mode. For a usage example, see the definition for the Closed Loop Sensor (SFPTclosedLoopSensor).

To program a preset, the SNVT_preset output is transmitted from a sensor with updated values for SNVT_preset.value, SNVT_preset.selector, and the time-related fields. In addition, SNVT_preset.learn is set to LN_LEARN_VALUE —or alternatively set to LN_LEARN_CURRENT, which causes the receiving actuator to learn whatever its current value is. A pre-programmed preset can be selected by transmitting the SNVT_preset output with the relevant preset number set in SNVT_preset.selector, and with SNVT_preset.learn set to LN_RECALL.

The time-related fields specify the time period over which the actuator should progress from the current level to the newly selected preset level. A benefit of this mechanism is that any set of

actuators that are preset with a common rate value for a particular preset number, will all arrive at this new value at the same time, regardless of the individual preset values to which they ramp.

Details:

Standard: *yes*
 Resource Set: *Standard 00:00:00:00:00:00:00:00-0*
 Index: *94*
 Obsolete: *no*
 Size: *14*
 Programmatic Name: *SNVT_preset*
 Neuron C Type:

Structure	
learn	Learn mode (learn mode names) .
	<i>learn_mode_t</i>
selector	Selector (16-bit unsigned value) . The selector is used to choose which preset.
	unsigned long Minimum: <i>0</i> Maximum: <i>65535</i> Scaling <i>1, 0, 0</i> (A,B,C): Scaled <i>$1 * 10^0 * (Raw+0)$</i> value: Resolution: <i>1</i>
value	Value (array of 4 bytes) .
	unsigned short [4] Minimum: <i>0</i> Maximum: <i>255</i> Scaling <i>1, 0, 0</i> (A,B,C): Scaled <i>$1 * 10^0 * (Raw+0)$</i> value: Resolution: <i>1</i>
day	Days (days) . The value 65535 represents NULL or unknown elapsed time.
	unsigned long Minimum: <i>0</i> Maximum: <i>65535</i> Invalid: <i>65535</i> Scaling <i>1, 0, 0</i> (A,B,C): Scaled <i>$1 * 10^0 * (Raw+0)$</i> value: Resolution: <i>1</i>
hour	Hours (hours) . This field uses a 24-hour value.

	unsigned short Minimum: 0 Maximum: 23 Scaling 1, 0, 0 (A,B,C): Scaled value: $1 * 10^0 * (Raw + 0)$ Resolution: 1
minute	Minutes (minutes) .
	unsigned short Minimum: 0 Maximum: 59 Scaling 1, 0, 0 (A,B,C): Scaled value: $1 * 10^0 * (Raw + 0)$ Resolution: 1
second	Seconds (seconds) .
	unsigned short Minimum: 0 Maximum: 59 Scaling 1, 0, 0 (A,B,C): Scaled value: $1 * 10^0 * (Raw + 0)$ Resolution: 1
millisecond	Milliseconds (milliseconds) .
	unsigned long Minimum: 0 Maximum: 999 Scaling 1, 0, 0 (A,B,C): Scaled value: $1 * 10^0 * (Raw + 0)$ Resolution: 1

Formats:

*SNVT_preset: text("%m %d <%x %x %x %x> %d %d:%d:%d:%d",
learn, selector, value[0], value[1], value[2], value[3], day, hour, minute,
second, millisecond)*

*SNVT_preset#LO: text((" %m %d <%x %x %x %x> %d ", learn, selector,
value[0], value[1], value[2], value[3], day), time(hour, minute, second,
millisecond))*

Used by:

*SFPTclosedLoopActuator SFPTclosedLoopSensor
SFPTlightingPanelController SFPTopenLoopActuator
SFPTopenLoopSensor*

SNVT_press

Overview:

Pressure (gauge) (kiloPascals) .

Details:

Standard:	<i>yes</i>
Resource Set:	<i>Standard 00:00:00:00:00:00:00:00-0</i>
Index:	<i>30</i>
Obsolete:	<i>no</i>
Size:	<i>2</i>
Programmatic Name:	<i>SNVT_press</i>
Neuron C Type:	signed long
	Minimum: <i>-32768</i>
	Maximum: <i>32767</i>
	Invalid: <i>32767</i>
	Scaling (A,B,C): <i>1, -1, 0</i>
	Scaled value: <i>1 * 10⁻¹ * (Raw+0)</i>
	Resolution: <i>0.1</i>
Formats:	<i>SNVT_press#SI: text("%f")</i>
	<i>SNVT_press#US: text("%f", *4.0217+0(0:954))</i>
	<i>SNVT_press#US_psi: text("%f", *0.1450+0(0:875))</i>
Used by:	<i>SCPTmaxPressureSetpoint SCPTmaxRemotePressureSetpoint</i> <i>SCPTminPressureSetpoint SCPTminRemotePressureSetpoint</i> <i>SCPTpumpCharacteristic SFPTgeneratorSet SFPTpressureSensor</i> <i>SFPTpumpController SFPTrefrigDisplayCaseControllerEvaporator</i> <i>SNVT_pumpset_sn</i>

SNVT_press_f

Overview:

Pressure (gauge) (Pascals) .

Details:

Standard:	<i>yes</i>
Resource Set:	<i>Standard 00:00:00:00:00:00:00:00-0</i>

Index: 59
 Obsolete: no
 Size: 4
 Programmatic Name: *SNVT_press_f*
 Neuron C Type: float
 Minimum: $-3.4028234663853E+038$
 Maximum: $3.4028234663853E+038$
 Formats: *SNVT_press_f#SI: text("%f")*
 *SNVT_press_f#SI_kPa: text("%f", *0.001+0(0:973))*
 *SNVT_press_f#US: text("%f", *0.0040217+0(0:954))*
 *SNVT_press_f#US_psi: text("%f", *1.4504e-4+0(0:875))*
 Used by: *SFPTpressureSensor* *SNVT_pumpset_sn*

SNVT_press_p

Overview:

Pressure (gauge) (Pascals) .

Details:

Standard: yes
 Resource Set: *Standard 00:00:00:00:00:00:00:00-0*
 Index: 113
 Obsolete: no
 Size: 2
 Programmatic Name: *SNVT_press_p*
 Neuron C Type: signed long
 Minimum: -32768
 Maximum: 32767
 Invalid: 32767
 Scaling (A,B,C): 1, 0, 0
 Scaled value: $1 * 10^0 * (Raw + 0)$
 Resolution: 1
 Formats: *SNVT_press_p#SI: text("%f")*
 *SNVT_press_p#US: text("%f", *0.0040217+0(0:954))*
 *SNVT_press_p#US_psi: text("%f", *1.4504e-4+0(0:875))*
 Used by: *SCPTbuildingStaticPressureSetpoint* *SCPTductStaticPressureLimit*

SCPTductStaticPressureSetpoint SCPTmaxDuctStaticPressureSetpoint
SCPTminDuctStaticPressureSetpoint
SCPTreturnFanStaticPressureSetpoint SFPTdischargeAirController
SFPTpressureSensor

SNVT_privacyzone

Overview:

Privacy zone (action, zone number, camera ID) .

Used to support programmable privacy zones. The command consists of an action type, the number of the privacy zone, and the ID value of the camera telemetry receiver having this privacy zone.

EXAMPLES

Example 1, input

nvoPZset.action = PZ_UPPER_LEFT

nvoPZset.number = (unused)

Physical value = programming the upper left corner for privacy zone #1

Example 2, output

nvoPZset.action = PZ_ENTER

nvoPZset.number = 1

Physical value = the camera is inside privacy zone #1

Details:

Standard: *yes*

Resource Set: *Standard 00:00:00:00:00:00:00:00-0*

Index: *151*

Obsolete: *no*

Size: *4*

Programmatic Name: *SNVT_privacyzone*

Neuron C Type:

Structure	
action	Privacy zone action type (privacy zone action type names) .
	<i>privacyzone_t</i>
number	Zone number (zone number) .
	unsigned short
	Minimum: <i>0</i>
	Maximum: <i>255</i>

	Invalid: 0 Scaling (A,B,C): $1, 0, 0$ Scaled value: $1 * 10^0 * (Raw + 0)$ Resolution: 1
camera_id	Camera ID (ID number) .
	unsigned long Minimum: 0 Maximum: 65535 Invalid: 0 Scaling (A,B,C): $1, 0, 0$ Scaled value: $1 * 10^0 * (Raw + 0)$ Resolution: 1

Formats: *SNVT_privacyzone: text("%m %d %d", action, number, camera_id)*

SNVT_ptz

Overview:

Camera PTZ (pan, pan speed, tilt, tilt speed, zoom, zoom speed) .

The pan and tilt speed values, as well as the pan and tilt directions, have to be updated simultaneously, since in most applications, a joystick is used to control camera movements. The movement must be smooth in all directions, for all magnitudes of pan and tilt speed.

The network output variable of this type transmits the speed and direction for pan, tilt, and zoom, in each update, to meet the requirements described above. If the speed is zero, e.g., in of case of a network variable of SNVT_ptz from a non-joystick, a default speed will be used. This default speed may optionally be programmable, via a configuration property.

EXAMPLES

Example 1

tilt.direction = PAN_RIGHT
tilt.speed = 125 (0x7D)
Physical value = Pan to the right by 50.0%

Example 2

tilt.direction = TILT_UP
tilt.speed = 63 (0x3F)

Physical value = Tilt up by 25.2%

Details:

Standard: *yes*
 Resource Set: *Standard 00:00:00:00:00:00:00:00-0*
 Index: *150*
 Obsolete: *no*
 Size: *6*
 Programmatic Name: *SNVT_ptz*
 Neuron C Type:

Structure	
pan_dir	Pan direction (pan direction names) .
	<i>pan_dir_t</i>
pan_speed	Pan speed (% of full level) .
	unsigned short Minimum: <i>0</i> Maximum: <i>250</i> Scaling <i>4, -1, 0</i> (A,B,C): Scaled <i>$4 * 10^{-1} * (Raw+0)$</i> value: Resolution: <i>0.4</i>
tilt_dir	Tilt direction (tilt direction names) .
	<i>tilt_dir_t</i>
tilt_speed	Tilt speed (% of full level) .
	unsigned short Minimum: <i>0</i> Maximum: <i>250</i> Scaling <i>4, -1, 0</i> (A,B,C): Scaled <i>$4 * 10^{-1} * (Raw+0)$</i> value: Resolution: <i>0.4</i>
zoom	Zoom direction (zoom direction names) .
	<i>zoom_t</i>
zoom_speed	Zoom speed (% of full level) .
	unsigned short Minimum: <i>0</i> Maximum: <i>250</i> Scaling <i>4, -1, 0</i> (A,B,C): Scaled <i>$4 * 10^{-1} * (Raw+0)$</i> value:

Resolution: 0.4

Formats: *SNVT_ptz: text("%m %f%m %f%m %f", pan_dir, pan_speed, tilt_dir, tilt_speed, zoom, zoom_speed)*

Used by: *SCPTdefaultPanTiltZoomSpeeds*

SNVT_pump_sensor

Overview:

Pump sensor (speed, temperature, status) .

Contains readings from sensors on a mechanical vacuum pump.

Details:

Standard: *yes*

Resource Set: *Standard 00:00:00:00:00:00:00:00-0*

Index: *159*

Obsolete: *no*

Size: *19*

Programmatic Name: *SNVT_pump_sensor*

Neuron C Type:

Structure	
rotational_speed	Rotational speed.
	<i>SNVT_freq_hz</i>
body_temperature	Body temperature.
	<i>SNVT_temp</i>
motor_external_temperature	Motor external temp.
	<i>SNVT_temp</i>
motor_internal_temperature	Motor internal temp.
	<i>SNVT_temp</i>
motor_overloaded	Motor overloaded (boolean) .
	<i>boolean_t</i>
oil_level_low	Oil level low (boolean) .
	<i>boolean_t</i>
phase_imbalance_detected	Phase imbalance (boolean) .
	<i>boolean_t</i>
current_usage	Current usage.
	<i>SNVT_amp</i>

power_usage	Power usage.
	<i>SNVT_power_kilo</i>
temperature_control	Pump body temp control status (temperature control status names) .
	<i>unit_temp_t</i>
electromagnetic_brake_active	Electromagnetic brake active (boolean) .
	<i>boolean_t</i>
friction_brake_active	Friction brake active (boolean) .
	<i>boolean_t</i>
gas_brake_active	Gas brake active (boolean) .
	<i>boolean_t</i>

Formats:

*SNVT_pump_sensor#SI: text("%f%f%f%f%
%m %m %m %f%f%
%m %m", rotational_speed, body_temperature*1+0(0:854),
motor_external_temperature*1+0(0:854),
motor_internal_temperature*1+0(0:854), motor_overloaded,
oil_level_low, phase_imbalance_detected, current_usage, power_usage,
temperature_control, electromagnetic_brake_active,
friction_brake_active, gas_brake_active)*

*SNVT_pump_sensor#US: text("%f%f%f%f%
%m %m %m %f%f%
%m %m", rotational_speed, body_temperature*1.8+32(0:855),
motor_external_temperature*1.8+32(0:855),
motor_internal_temperature*1.8+32(0:855), motor_overloaded,
oil_level_low, phase_imbalance_detected, current_usage, power_usage,
temperature_control, electromagnetic_brake_active,
friction_brake_active, gas_brake_active)*

SNVT_pumpset_mn

Overview:

Pumpset (main, booster, priority, ready, emerg, main enabled, booster enabled, maint) .

This SNVT contains the status of the main features of a ‘pumpset’, which consists of one or more mechanical vacuum pumps operating in unison.

Details:

Standard:	<i>yes</i>
Resource Set:	<i>Standard 00:00:00:00:00:00:00:00-0</i>
Index:	<i>156</i>
Obsolete:	<i>no</i>
Size:	<i>8</i>

Programmatic Name: *SNVT_pumpset_mn*

Neuron C Type:

Structure	
main_pump	Main pump state (motor state names) .
	<i>motor_state_t</i>
booster_pump	Booster pump state (motor state names) .
	<i>motor_state_t</i>
priority_level	Priority level (priority level names) .
	<i>priority_level_t</i>
process_ready	Process ready (boolean) .
	<i>boolean_t</i>
emergency_stop_activated	Emergency stop (boolean) .
	<i>boolean_t</i>
main_pump_drive_enabled	Main pump enabled (boolean) .
	<i>boolean_t</i>
booster_pump_drive_enabled	Booster pump enabled (boolean) .
	<i>boolean_t</i>
maintenance_required	Maintenance required (boolean) .
	<i>boolean_t</i>

Formats:

*SNVT_pumpset_mn: text("%m %m %m %m %m %m %m %m",
main_pump, booster_pump, priority_level, process_ready,
emergency_stop_activated, main_pump_drive_enabled,
booster_pump_drive_enabled, maintenance_required)*

SNVT_pumpset_sn

Overview:

Pumpset sensor (dilution, exhaust, pressure, vacuum, ...) .

This SNVT contains readings from sensors in a ‘pumpset’, which consists of one or more pumps operating in unison.

Details:

Standard: *yes*

Resource Set: *Standard 00:00:00:00:00:00:00:00-0*

Index: *158*

Obsolete: *no*

Size: *23*

Programmatic Name: *SNVT_pumpset_sn*

Neuron C Type:

Structure	
total_dilution_flow	Dilution gas flow.
	<i>SNVT_flow_mil</i>
exhaust_temperature	Exhaust line external temperature.
	<i>SNVT_temp</i>
exhaust_pressure	Exhaust line pressure.
	<i>SNVT_press</i>
shaft_seal_purge_pressure	Shaft seal purge pressure.
	<i>SNVT_press</i>
inlet_vacuum	Process gas inlet pressure.
	<i>SNVT_press_f</i>
supply_voltage	Pumpset power supply voltage.
	<i>SNVT_volt</i>
coolant_flow	Total coolant flow.
	<i>SNVT_flow_mil</i>
coolant_flow_low	Coolant flow too low (boolean) .
	<i>boolean_t</i>
dilution_active	Dilution gas being used (boolean) .
	<i>boolean_t</i>
ballast_dilution_active	Dilution gas being used as ballast (boolean) .
	<i>boolean_t</i>
inlet_purge_dilution_active	Dilution gas being used to purge process gas (boolean) .
	<i>boolean_t</i>
exhaust_dilution_active	Dilution gas being used to dilute exhaust (boolean) .
	<i>boolean_t</i>
dilution_flow_out_of_range	Dilution gas flow outside normal range (boolean) .
	<i>boolean_t</i>
power_supply_on	Main power supply is on (boolean) .
	<i>boolean_t</i>

Formats:

SNVT_pumpset_sn#SI: *text*("%d %f%f%f%f%f%d %m %m %m %m %m %m %m", total_dilution_flow*1+0(0:838), exhaust_temperature*1+0(0:854), exhaust_pressure, shaft_seal_purge_pressure, inlet_vacuum, supply_voltage, coolant_flow*1+0(0:838), coolant_flow_low, dilution_active, ballast_dilution_active, inlet_purge_dilution_active, exhaust_dilution_active, dilution_flow_out_of_range, power_supply_on)

*SNVT_pumpset_sn#US: text("%d %f%f%f%f%f%f%d %m %m %m %m %m %m %m", total_dilution_flow*0.06102+0(0:1460), exhaust_temperature*1.8+32(0:855), exhaust_pressure*0.2953+0(0:1402), shaft_seal_purge_pressure*0.2953+0(0:1402), inlet_vacuum*0.2953+0(0:1402), supply_voltage, coolant_flow*0.0338+0(0:839), coolant_flow_low, dilution_active, ballast_dilution_active, inlet_purge_dilution_active, exhaust_dilution_active, dilution_flow_out_of_range, power_supply_on)*

SNVT_pwr_fact

Overview:

Power factor (multiplier) .

Details:

Standard:	<i>yes</i>
Resource Set:	<i>Standard 00:00:00:00:00:00:00:00-0</i>
Index:	<i>98</i>
Obsolete:	<i>no</i>
Size:	<i>2</i>
Programmatic Name:	<i>SNVT_pwr_fact</i>
Neuron C Type:	signed long
	Minimum: <i>-20000</i>
	Maximum: <i>20000</i>
	Scaling (A,B,C): <i>5, -5, 0</i>
	Scaled value: <i>5 *10⁻⁵ *(Raw+0)</i>
	Resolution: <i>5E-005</i>
Formats:	<i>SNVT_pwr_fact: text("%f")</i>
Used by:	<i>SFPTautomaticTransferSwitch SFPTgeneratorSet</i>

SNVT_pwr_fact_f

Overview:

Power factor (multiplier) .

Details:

Standard:	<i>yes</i>
-----------	------------

Resource Set: *Standard 00:00:00:00:00:00:00:00-0*
 Index: *99*
 Obsolete: *no*
 Size: *4*
 Programmatic Name: *SNVT_pwr_fact_f*
 Neuron C Type: *float*
 Minimum: *-1*
 Maximum: *1*
 Formats: *SNVT_pwr_fact_f: text("%f")*

SNVT_rac_ctrl

Overview:

Rail-Audio Controller Control. Invokes audio control for a given source

Details:

Standard: *yes*
 Resource Set: *Standard 00:00:00:00:00:00:00:00-0*
 Index: *181*
 Obsolete: *no*
 Size: *18*
 Programmatic Name: *SNVT_rac_ctrl*
 Neuron C Type:

Structure	
audio_line	.
	bitfield Signed: <i>no</i> Width: <i>3</i> Offset: <i>0</i> Minimum: <i>0</i> Maximum: <i>7</i> Scaling <i>1, 0, 0</i> (A,B,C): Scaled <i>$1 * 10^0 * (Raw + 0)$</i> value: Resolution: <i>1</i>
duplex_full	.
	bitfield Signed: <i>no</i> Width: <i>1</i>

	Offset: 3 Minimum: 0 Maximum: 1 Scaling (A,B,C): 1, 0, 0 Scaled value: $1 * 10^0 * (Raw + 0)$ Resolution: 1						
dest_p2p	.						
	bitfield Signed: no Width: 1 Offset: 4 Minimum: 0 Maximum: 1 Scaling (A,B,C): 1, 0, 0 Scaled value: $1 * 10^0 * (Raw + 0)$ Resolution: 1						
reserved	.						
	bitfield Signed: no Width: 3 Offset: 5 Minimum: 0 Maximum: 7 Scaling (A,B,C): 1, 0, 0 Scaled value: $1 * 10^0 * (Raw + 0)$ Resolution: 1						
audio_type	.						
	<i>rail_audio_type_t</i>						
addr_init	.						
	<table border="1"> <tr> <th colspan="2">Structure</th></tr> <tr> <th>unit_id</th><th>.</th></tr> <tr> <td></td><td> bitfield Signed: no Width: 4 Offset: 0 Minimum: 0 Maximum: 8 Scaling (A,B,C): 1, 0, 0 </td></tr> </table>	Structure		unit_id	.		bitfield Signed: no Width: 4 Offset: 0 Minimum: 0 Maximum: 8 Scaling (A,B,C): 1, 0, 0
Structure							
unit_id	.						
	bitfield Signed: no Width: 4 Offset: 0 Minimum: 0 Maximum: 8 Scaling (A,B,C): 1, 0, 0						

		Scaled value: $1 * 10^0 * (Raw + 0)$ Resolution: 1
	location	.
		bitfield Signed: no Width: 4 Offset: 4 Minimum: 0 Maximum: 15 Scaling (A,B,C): 1, 0, 0 Scaled value: $1 * 10^0 * (Raw + 0)$ Resolution: 1
	car_id	.
		bitfield Signed: no Width: 5 Offset: 0 Minimum: 0 Maximum: 31 Scaling (A,B,C): 1, 0, 0 Scaled value: $1 * 10^0 * (Raw + 0)$ Resolution: 1
	reserved	.
		bitfield Signed: no Width: 3 Offset: 5 Minimum: 0 Maximum: 7 Scaling (A,B,C): 1, 0, 0 Scaled value: $1 * 10^0 * (Raw + 0)$ Resolution: 1
	audio_sensor_type	.
		<i>rail_audio_sensor_type_t</i>
addr_talk		.
	Structure	
	unit_id	.

	bitfield Signed: <i>no</i> Width: <i>4</i> Offset: <i>0</i> Minimum: <i>0</i> Maximum: <i>15</i> Scaling <i>1, 0, 0</i> (A,B,C): Scaled value: $1 * 10^0 * (Raw+0)$ Resolution: <i>1</i>
location	.
	bitfield Signed: <i>no</i> Width: <i>4</i> Offset: <i>4</i> Minimum: <i>0</i> Maximum: <i>15</i> Scaling <i>1, 0, 0</i> (A,B,C): Scaled value: $1 * 10^0 * (Raw+0)$ Resolution: <i>1</i>
car_id	.
	bitfield Signed: <i>no</i> Width: <i>5</i> Offset: <i>0</i> Minimum: <i>0</i> Maximum: <i>31</i> Scaling <i>1, 0, 0</i> (A,B,C): Scaled value: $1 * 10^0 * (Raw+0)$ Resolution: <i>1</i>
reserved	.
	bitfield Signed: <i>no</i> Width: <i>3</i> Offset: <i>5</i> Minimum: <i>0</i> Maximum: <i>7</i> Scaling <i>1, 0, 0</i> (A,B,C): Scaled value: $1 * 10^0 * (Raw+0)$

		value: Resolution: <i>1</i>
	audio_sensor_type	.
		<i>rail_audio_sensor_type_t</i>
addr_dest		.
	Union	
	p2p	.
	Structure	
	unit_id	.
		bitfield Signed: <i>no</i> Width: <i>4</i> Offset: <i>0</i> Minimum: <i>0</i> Maximum: <i>15</i> Scaling <i>1, 0, 0</i> (A,B,C): Scaled <i>1 * 10⁰ *</i> value: <i>(Raw+0)</i> Resolution: <i>1</i>
	location	.
		bitfield Signed: <i>no</i> Width: <i>4</i> Offset: <i>4</i> Minimum: <i>0</i> Maximum: <i>15</i> Scaling <i>1, 0, 0</i> (A,B,C): Scaled <i>1 * 10⁰ *</i> value: <i>(Raw+0)</i> Resolution: <i>1</i>
	car_id	.
		bitfield Signed: <i>no</i> Width: <i>5</i> Offset: <i>0</i> Minimum: <i>0</i> Maximum: <i>31</i> Scaling <i>1, 0, 0</i> (A,B,C): Scaled <i>1 * 10⁰ *</i> value: <i>(Raw+0)</i>

		Resolution: <i>1</i>
reserved		.
		bitfield Signed: <i>no</i> Width: <i>3</i> Offset: <i>5</i> Minimum: <i>0</i> Maximum: <i>7</i> Scaling <i>1, 0, 0</i> (A,B,C): Scaled <i>1 * 10⁰ *</i> value: <i>(Raw+0)</i> Resolution: <i>1</i>
audio_sensor_type		.
		<i>rail_audio_sensor_type_t</i>
p2m		.
	Structure	
	mask_unit	.
		unsigned char Minimum: <i>0</i> Maximum: <i>255</i> Scaling <i>1, 0, 0</i> (A,B,C): Scaled <i>1 * 10⁰ *(Raw+0)</i> value: Resolution: <i>1</i>
	mask_car	.
		unsigned char [4] Minimum: <i>0</i> Maximum: <i>255</i> Scaling <i>1, 0, 0</i> (A,B,C): Scaled <i>1 * 10⁰ *(Raw+0)</i> value: Resolution: <i>1</i>
	mask_location	.
		unsigned char [2] Minimum: <i>0</i> Maximum: <i>255</i> Scaling <i>1, 0, 0</i> (A,B,C): Scaled <i>1 * 10⁰ *(Raw+0)</i> value:

			Resolution: 1
		mask_audio	.
			unsigned char [3] Minimum: 0 Maximum: 255 Scaling 1, 0, 0 (A,B,C): Scaled $1 * 10^0 * (Raw + 0)$ value: Resolution: 1

Formats: *SNVT_rac_ctrl: text("AL: %d, FD: %d, Dp2p: %d, (rsvd: %d), AT: %m, Init: %d {%d} %d (%d) AS: %m + Talk: %d {%d} %d (%d) AS: %m + P2P: [%d {%d} %d (%d) AS: %m], P2M: [%x - %x%x%x%x {%x%x} MA: %x%x%x]", audio_line, duplex_full, dest_p2p, reserved, audio_type, addr_init.unit_id, addr_init.location, addr_init.car_id, addr_init.reserved, addr_init.audio_sensor_type, addr_talk.unit_id, addr_talk.location, addr_talk.car_id, addr_talk.reserved, addr_talk.audio_sensor_type, addr_dest.p2p.unit_id, addr_dest.p2p.location, addr_dest.p2p.car_id, addr_dest.p2p.reserved, addr_dest.p2p.audio_sensor_type, addr_dest.p2m.mask_unit, addr_dest.p2m.mask_car[0], addr_dest.p2m.mask_car[1], addr_dest.p2m.mask_car[2], addr_dest.p2m.mask_car[3], addr_dest.p2m.mask_location[0], addr_dest.p2m.mask_location[1], addr_dest.p2m.mask_audio[0], addr_dest.p2m.mask_audio[1], addr_dest.p2m.mask_audio[2])*

Used by: *SFPTrailcarAudioController SFPTrailcarAudioSensor*

SNVT_rac_req

Overview:

Rail-Audio Controller Request. Requests audio control for a given source

Details:

Standard: *yes*
Resource Set: *Standard 00:00:00:00:00:00:00:00-0*
Index: *182*
Obsolete: *no*
Size: *15*
Programmatic Name: *SNVT_rac_req*
Neuron C Type:

Structure	
dest_def	.
	bitfield

	Signed: <i>no</i> Width: <i>1</i> Offset: <i>0</i> Minimum: <i>0</i> Maximum: <i>1</i> Scaling (A,B,C): <i>1, 0, 0</i> Scaled value: $1 * 10^0 * (Raw + 0)$ Resolution: <i>1</i>						
dest_p2p	.						
	bitfield Signed: <i>no</i> Width: <i>1</i> Offset: <i>1</i> Minimum: <i>0</i> Maximum: <i>1</i> Scaling (A,B,C): <i>1, 0, 0</i> Scaled value: $1 * 10^0 * (Raw + 0)$ Resolution: <i>1</i>						
reserved	.						
	bitfield Signed: <i>no</i> Width: <i>6</i> Offset: <i>2</i> Minimum: <i>0</i> Maximum: <i>63</i> Scaling (A,B,C): <i>1, 0, 0</i> Scaled value: $1 * 10^0 * (Raw + 0)$ Resolution: <i>1</i>						
audio_type	.						
	<i>rail_audio_type_t</i>						
addr_init	.						
	<table> <tr> <th colspan="2">Structure</th></tr> <tr> <th>unit_id</th><th>.</th></tr> <tr> <td></td><td> bitfield Signed: <i>no</i> Width: <i>4</i> Offset: <i>0</i> Minimum: <i>0</i> </td></tr> </table>	Structure		unit_id	.		bitfield Signed: <i>no</i> Width: <i>4</i> Offset: <i>0</i> Minimum: <i>0</i>
Structure							
unit_id	.						
	bitfield Signed: <i>no</i> Width: <i>4</i> Offset: <i>0</i> Minimum: <i>0</i>						

	<p>Maximum: <i>15</i></p> <p>Scaling <i>1, 0, 0</i></p> <p>(A,B,C):</p> <p>Scaled <i>1 *10⁰ *(Raw+0)</i></p> <p>value:</p> <p>Resolution: <i>1</i></p>
location	.
	<p>bitfield</p> <p>Signed: <i>no</i></p> <p>Width: <i>4</i></p> <p>Offset: <i>4</i></p> <p>Minimum: <i>0</i></p> <p>Maximum: <i>15</i></p> <p>Scaling <i>1, 0, 0</i></p> <p>(A,B,C):</p> <p>Scaled <i>1 *10⁰ *(Raw+0)</i></p> <p>value:</p> <p>Resolution: <i>1</i></p>
car_id	.
	<p>bitfield</p> <p>Signed: <i>no</i></p> <p>Width: <i>5</i></p> <p>Offset: <i>0</i></p> <p>Minimum: <i>0</i></p> <p>Maximum: <i>31</i></p> <p>Scaling <i>1, 0, 0</i></p> <p>(A,B,C):</p> <p>Scaled <i>1 *10⁰ *(Raw+0)</i></p> <p>value:</p> <p>Resolution: <i>1</i></p>
reserved	.
	<p>bitfield</p> <p>Signed: <i>no</i></p> <p>Width: <i>3</i></p> <p>Offset: <i>5</i></p> <p>Minimum: <i>0</i></p> <p>Maximum: <i>7</i></p> <p>Scaling <i>1, 0, 0</i></p> <p>(A,B,C):</p> <p>Scaled <i>1 *10⁰ *(Raw+0)</i></p> <p>value:</p> <p>Resolution: <i>1</i></p>
audio_sensor_type	.
	<i>rail_audio_sensor_type_t</i>
addr_dest	.

Union	
p2p	.
	Structure
	unit_id
	bitfield Signed: <i>no</i> Width: <i>4</i> Offset: <i>0</i> Minimum: <i>0</i> Maximum: <i>15</i> Scaling <i>1, 0, 0</i> (A,B,C): Scaled <i>1 * 10⁰ *</i> value: <i>(Raw+0)</i> Resolution: <i>1</i>
	location
	bitfield Signed: <i>no</i> Width: <i>4</i> Offset: <i>4</i> Minimum: <i>0</i> Maximum: <i>15</i> Scaling <i>1, 0, 0</i> (A,B,C): Scaled <i>1 * 10⁰ *</i> value: <i>(Raw+0)</i> Resolution: <i>1</i>
	car_id
	bitfield Signed: <i>no</i> Width: <i>5</i> Offset: <i>0</i> Minimum: <i>0</i> Maximum: <i>31</i> Scaling <i>1, 0, 0</i> (A,B,C): Scaled <i>1 * 10⁰ *</i> value: <i>(Raw+0)</i> Resolution: <i>1</i>
	reserved
	bitfield Signed: <i>no</i> Width: <i>3</i> Offset: <i>5</i>

		Minimum: 0 Maximum: 7 Scaling 1, 0, 0 (A,B,C): Scaled value: $1 * 10^0 * (Raw+0)$ Resolution: 1
	audio_sensor_type	.
		<i>rail_audio_sensor_type_t</i>
p2m		.
	Structure	
	mask_unit	.
		unsigned char Minimum: 0 Maximum: 255 Scaling 1, 0, 0 (A,B,C): Scaled value: $1 * 10^0 * (Raw+0)$ Resolution: 1
	mask_car	.
		unsigned char [4] Minimum: 0 Maximum: 255 Scaling 1, 0, 0 (A,B,C): Scaled value: $1 * 10^0 * (Raw+0)$ Resolution: 1
	mask_location	.
		unsigned char [2] Minimum: 0 Maximum: 255 Scaling 1, 0, 0 (A,B,C): Scaled value: $1 * 10^0 * (Raw+0)$ Resolution: 1
	mask_audio	.
		unsigned char [3] Minimum: 0 Maximum: 255 Scaling 1, 0, 0 (A,B,C):

```
SNVT rac_req: text("DD: %d, Dp2p: %d, (rsvd: %d), AT: %m, Init: %d
{%d} %d (%d) AS: %m + P2P: [%d {%d} %d (%d) AS: %m], P2M: [%x
- %x%x%x%x {%x%x} MA: %x%x%x]", dest_def, dest_p2p, reserved,
audio_type, addr_init.unit_id, addr_init.location, addr_init.car_id,
addr_init.reserved, addr_init.audio_sensor_type, addr_dest.p2p.unit_id,
addr_dest.p2p.location, addr_dest.p2p.car_id, addr_dest.p2p.reserved,
addr_dest.p2p.audio_sensor_type, addr_dest.p2m.mask_unit,
addr_dest.p2m.mask_car[0], addr_dest.p2m.mask_car[1],
addr_dest.p2m.mask_car[2], addr_dest.p2m.mask_car[3],
addr_dest.p2m.mask_location[0], addr_dest.p2m.mask_location[1],
addr_dest.p2m.mask_audio[0], addr_dest.p2m.mask_audio[1],
addr_dest.p2m.mask_audio[2]))
```

SNVT_reg_val

Standard:	<i>yes</i>
Resource Set:	<i>Standard 00:00:00:00:00:00:00:00-0</i>
Index:	<i>136</i>
Obsolete:	<i>no</i>
Size:	<i>6</i>
Programmatic Name:	<i>SNVT_reg_val</i>
Neuron C Type:	Structure

raw	Raw value.
	s32_type Minimum: <i>-2147483648</i> Maximum: <i>2147483647</i> Scaling <i>1, 0, 0</i> (A,B,C): Scaled value: $1 * 10^0 * (Raw + 0)$ Resolution: <i>1</i>
unit	Unit code (defines unit of measure) .
	<i>reg_val_unit_t</i>
nr_decimals	Number of decimals (digits to right of decimal point) .
	bitfield Signed: <i>no</i> Width: <i>3</i> Offset: <i>0</i> Minimum: <i>0</i> Maximum: <i>7</i> Scaling <i>1, 0, 0</i> (A,B,C): Scaled value: $1 * 10^0 * (Raw + 0)$ Resolution: <i>1</i>

Formats: *SNVT_reg_val: text("%d %m %d", raw, unit, nr_decimals)*

Used by: *SCPTbaseValue SFPTutilityDataLoggerRegister*

SNVT_reg_val_ts

Overview:

Register value (raw value, unit code, number of decimals, status, state, timestamp) .

Used to send a value and appropriate unit of measure to a recording or displaying device that would use the unit field to properly handle the 32-bit value.

The status bitfield has the following two representations, and the functional profile or installation should specify the representation used:

Singular, enumerated	Mask, bitmapped	Description
0x1 (b0001)	0x1 (b0001)	Time has been changed during the measuring period and the change exceeds the error limit, or time is invalid time.
		Internal/External error during the measuring period (e.g., watchdog

0x2 (b0010)	0x2 (b0010)	reset, error in EEPROM operation, memory checksum error, Polling error from the external device, register overflow or underflow—with subtracting pulse inputs—during the measuring period).
0x3 (b0011)	0x4 (b0100)	Power failure during the measuring period.
0x4 (b0100)	0x8 (b1000)	Illegal value request (e.g., uninitiated after boot up, input values out of range).

EXAMPLES

Time has been changed during the measuring period and the change exceeds the error limit, or time is invalid time. Register is activated. Start time of measuring is January 31, 2000 at 23:45:00.

```

raw                =      1.23456 e5 = 123456
unit               =      RVU_KW
nr_decimals        =      2
status             =      0x1
reg_state          =      1
year               =      2000
month              =      1
day                =      31
hour               =      23
minute             =      45
second             =      00
Physical value     =      1234.56 kW

```

Details:

Standard: *yes*
Resource Set: *Standard 00:00:00:00:00:00:00:00-0*
Index: *137*
Obsolete: *no*
Size: *13*
Programmatic Name: *SNVT_reg_val_ts*
Neuron C Type:

Structure	
raw	Raw value.
	s32_type Minimum: <i>-2147483648</i> Maximum: <i>2147483647</i> Scaling <i>1, 0, 0</i> (A,B,C): Scaled value: <i>1 *10⁰ *(Raw+0)</i> Resolution: <i>1</i>
unit	Unit code (unit names) .

	<i>reg_val_unit_t</i>
nr_decimals	Number of decimals (digits to right of decimal point) .
	bitfield Signed: <i>no</i> Width: <i>3</i> Offset: <i>0</i> Minimum: <i>0</i> Maximum: <i>7</i> Scaling <i>1, 0, 0</i> (A,B,C): Scaled value: $1 * 10^0 * (Raw + 0)$ Resolution: <i>1</i>
status	Status (status or error during measuring period) .
	bitfield Signed: <i>no</i> Width: <i>4</i> Offset: <i>3</i> Minimum: <i>0</i> Maximum: <i>15</i> Scaling <i>1, 0, 0</i> (A,B,C): Scaled value: $1 * 10^0 * (Raw + 0)$ Resolution: <i>1</i>
reg_state	Activation state (activation state of register) .
	bitfield Signed: <i>no</i> Width: <i>1</i> Offset: <i>7</i> Minimum: <i>0</i> Maximum: <i>1</i> Scaling <i>1, 0, 0</i> (A,B,C): Scaled value: $1 * 10^0 * (Raw + 0)$ Resolution: <i>1</i>
year	Year (years) . Zero (0) means year not specified. Minus one (-1) represents NULL date.
	signed long Minimum: <i>-1</i> Maximum: <i>3000</i> Invalid: <i>-1</i> Scaling <i>1, 0, 0</i> (A,B,C):

	Scaled value: $1 * 10^0 * (Raw + 0)$ Resolution: 1
month	Month (months) . Zero (0) means month not specified.
	unsigned short Minimum: 0 Maximum: 12 Scaling (A,B,C): 1, 0, 0 Scaled value: $1 * 10^0 * (Raw + 0)$ Resolution: 1
day	Day (days) . Zero (0) means day not specified.
	unsigned short Minimum: 0 Maximum: 31 Scaling (A,B,C): 1, 0, 0 Scaled value: $1 * 10^0 * (Raw + 0)$ Resolution: 1
hour	Hour (hours) . This field uses a 24-hour value.
	unsigned short Minimum: 0 Maximum: 23 Scaling (A,B,C): 1, 0, 0 Scaled value: $1 * 10^0 * (Raw + 0)$ Resolution: 1
minute	Minute (minutes) .
	unsigned short Minimum: 0 Maximum: 59 Scaling (A,B,C): 1, 0, 0 Scaled value: $1 * 10^0 * (Raw + 0)$ Resolution: 1
second	Second (seconds) .
	unsigned short Minimum: 0 Maximum: 59 Scaling (A,B,C): 1, 0, 0

	(A,B,C):
Scaled value:	$1 * 10^0 * (Raw + 0)$
Resolution:	1

Formats: *SNVT_reg_val_ts: text("%d %m %d %d %d %d/%d/%d %d:%d:%d", raw, unit, nr_decimals, status, reg_state, year, month, day, hour, minute, second)*

SNVT_reg_val_ts#LO: text("%d %m %d %d %d ",raw, unit, nr_decimals, status, reg_state),date(year, month, day), (" "), time(hour, minute, second))

Used by: *SFPTutilityDataLoggerRegister SFPTutilityMeter*

SNVT_res

Overview:

Electric resistance (Ohms) .

Details:

Standard: *yes*
Resource Set: *Standard 00:00:00:00:00:00:00:00-0*
Index: *31*
Obsolete: *no*
Size: *2*
Programmatic Name: *SNVT_res*
Neuron C Type: *unsigned long*
Minimum: *0*
Maximum: *65535*
Scaling (A,B,C): *1, -1, 0*
Scaled value: $1 * 10^{-1} * (Raw + 0)$
Resolution: *0.1*
Formats: *SNVT_res: text("%f")*

SNVT_res_f

Overview:

Electric resistance (Ohms) .

Details:

Standard:	<i>yes</i>
Resource Set:	<i>Standard 00:00:00:00:00:00:00:00-0</i>
Index:	<i>60</i>
Obsolete:	<i>no</i>
Size:	<i>4</i>
Programmatic Name:	<i>SNVT_res_f</i>
Neuron C Type:	float
	Minimum: <i>0</i>
	Maximum: <i>3.4028234663853E+038</i>
Formats:	<i>SNVT_res_f: text("%f")</i>

SNVT_res_kilo

Overview:

Electric resistance (kiloOhms) .

Details:

Standard:	<i>yes</i>
Resource Set:	<i>Standard 00:00:00:00:00:00:00:00-0</i>
Index:	<i>32</i>
Obsolete:	<i>no</i>
Size:	<i>2</i>
Programmatic Name:	<i>SNVT_res_kilo</i>
Neuron C Type:	unsigned long
	Minimum: <i>0</i>
	Maximum: <i>65535</i>
	Scaling (A,B,C): <i>1, -1, 0</i>
	Scaled value: <i>1 * 10⁻¹ * (Raw + 0)</i>
	Resolution: <i>0.1</i>
Formats:	<i>SNVT_res_kilo: text("%f")</i>

SNVT_rpm

Overview:

Angular velocity (revolutions/minute (RPM)) .

Details:

Standard:	<i>yes</i>
Resource Set:	<i>Standard 00:00:00:00:00:00:00:00-0</i>
Index:	<i>102</i>
Obsolete:	<i>no</i>
Size:	<i>2</i>
Programmatic Name:	<i>SNVT_rpm</i>
Neuron C Type:	unsigned long
	Minimum: <i>0</i>
	Maximum: <i>65535</i>
	Invalid: <i>65535</i>
	Scaling (A,B,C): <i>1, 0, 0</i>
	Scaled value: <i>1 * 10⁰ * (Raw + 0)</i>
	Resolution: <i>1</i>
Formats:	<i>SNVT_rpm: text("%f")</i>
Used by:	<i>SCPTnomRPM SCPTpumpCharacteristic SFPTgeneratorSet SFPTpumpController SNVT_clothes_w_c</i>

SNVT_sblnd_state

Overview:

Sunblind State. Provides the present state of a sunblind

A network variable of type SNVT_sblnd_state may be used to feedback or forward the actual sunblind position, error messages and the cause of the latest change of the setpoint.

Details:

Standard:	<i>yes</i>
Resource Set:	<i>Standard 00:00:00:00:00:00:00:00-0</i>
Index:	<i>180</i>
Obsolete:	<i>no</i>
Size:	<i>6</i>
Programmatic Name:	<i>SNVT_sblnd_state</i>
Neuron C Type:	

Structure	
pos	.
	<i>SNVT_setting</i>

cmd_source	.
	<i>sblnd_cmd_source_t</i>
error_code	.
	<i>sblnd_error_t</i>

Formats: *SNVT_sblnd_state: text("%m %d %d %m %m", pos.function, pos.setting, pos.rotation, cmd_source, error_code)*

Used by: *SFPTsunblindActuator SFPTsunblindController*

SNVT_scene

Overview:

Scene control (function, scene number) .

Details:

Standard: *yes*
Resource Set: *Standard 00:00:00:00:00:00:00:00-0*
Index: *115*
Obsolete: *no*
Size: *2*
Programmatic Name: *SNVT_scene*

Neuron C Type:

Structure	
function	Scene control function (scene control function names) .
	<i>scene_t</i>
scene_number	Scene number.
	unsigned short Minimum: <i>1</i> Maximum: <i>255</i> Scaling <i>1, 0, 0</i> (A,B,C): Scaled <i>$1 * 10^0 * (Raw+0)$</i> value: Resolution: <i>1</i>

Formats: *SNVT_scene: text("%m %d",function, scene_number)*

Used by: *SFPTlightingPanelController SFPTpartitionWallController*
SFPTrealTimeBasedScheduler SFPTsceneController
SFPTscenePanel SFPTsunblindController

SNVT_scene_cfg

Overview:

Scene configuration (function, scene number, setting, rotation, fade, delay, priority) .

Used to tell an actuator functional block to save a specified setting as a scene, report the scene data for a specified scene, and manage scene storage space.

Details:

Standard: *yes*
 Resource Set: *Standard 00:00:00:00:00:00:00:00-0*
 Index: *116*
 Obsolete: *no*
 Size: *10*
 Programmatic Name: *SNVT_scene_cfg*

Neuron C Type:

Structure	
function	Scene configuration function (scene configuration function names) .
	<i>scene_config_t</i>
scene_number	Scene number.
	unsigned short Minimum: <i>1</i> Maximum: <i>255</i> Scaling <i>1, 0, 0</i> (A,B,C): Scaled value: $1 * 10^0 * (Raw + 0)$ Resolution: <i>1</i>
setting	Scene setting level (% of full level) .
	unsigned short Minimum: <i>0</i> Maximum: <i>200</i> Invalid: <i>255</i> Scaling <i>5, -1, 0</i> (A,B,C): Scaled value: $5 * 10^{-1} * (Raw + 0)$ Resolution: <i>0.5</i>
rotation	Scene rotation angle (degrees) .
	signed long Minimum: <i>-17999</i> Maximum: <i>18000</i>

	Invalid: 32767 Scaling 2, -2, 0 (A,B,C): Scaled value: $2 * 10^{-2} * (Raw+0)$ Resolution: 0.02
fade_time	Scene fade time (seconds) .
	unsigned long Minimum: 0 Maximum: 65535 Invalid: 65535 Scaling 1, -1, 0 (A,B,C): Scaled value: $1 * 10^{-1} * (Raw+0)$ Resolution: 0.1
delay_time	Scene delay time (seconds) .
	unsigned long Minimum: 0 Maximum: 65535 Invalid: 65535 Scaling 1, -1, 0 (A,B,C): Scaled value: $1 * 10^{-1} * (Raw+0)$ Resolution: 0.1
scene_priority	Scene priority (priority value) .
	unsigned short Minimum: 0 Maximum: 255 Scaling 1, 0, 0 (A,B,C): Scaled value: $1 * 10^0 * (Raw+0)$ Resolution: 1

Formats: *SNVT_scene_cfg: text("%m %d %d %d %d %d %d", function, scene_number, setting, rotation, fade_time, delay_time, scene_priority)*

Used by: *SFPTsceneController*

SNVT_sched_val

Overview:

Scheduler value. Index from scheduler that selects entry in SCPTvalueDefinition array, or is a direct value output

This network variable type is used to report the state of a schedule. The value is used as the output of a Scheduler, or is used as an index into a SCPTvalueDefinition array, where the corresponding SCPTvalueDefinition array value is used as the output value.

Details:

Standard:	<i>yes</i>
Resource Set:	<i>Standard 00:00:00:00:00:00:00:00-0</i>
Index:	<i>177</i>
Obsolete:	<i>no</i>
Size:	<i>1</i>
Programmatic Name:	<i>SNVT_sched_val</i>
Neuron C Type:	unsigned short
	Minimum: <i>0</i>
	Maximum: <i>255</i>
	Invalid: <i>255</i>
	Scaling (A,B,C): <i>1, 0, 0</i>
	Scaled value: <i>$1 * 10^0 * (Raw + 0)$</i>
	Resolution: <i>1</i>
Formats:	<i>SNVT_sched_val: text("%d")</i>
Used by:	<i>SCPTscheduleTimeValue</i>

SNVT_sec_state

Overview:

Security State. It is used to communicate with security devices and is used for control of devices

Details:

Standard:	<i>yes</i>
Resource Set:	<i>Standard 00:00:00:00:00:00:00:00-0</i>
Index:	<i>178</i>
Obsolete:	<i>yes</i>
Size:	<i>3</i>
Programmatic Name:	<i>SNVT_sec_state</i>
Neuron C Type:	

Structure	
state	.
	<i>sec_state_t</i>

identity	.
	unsigned long Minimum: 0 Maximum: 65535 Scaling 1, 0, 0 (A,B,C): Scaled $1 * 10^0 * (Raw + 0)$ value: Resolution: 1

Formats: *SNVT_sec_state: text("%m %d", state, identity)*

Used by: *SCPTdefaultState*

SNVT_sec_status

Overview:

Security Status. Status information from a security device.

Details:

Standard: *yes*

Resource Set: *Standard 00:00:00:00:00:00:00:00-0*

Index: *179*

Obsolete: *yes*

Size: *3*

Programmatic Name: *SNVT_sec_status*

Neuron C Type:

Structure	
status	.
	<i>sec_status_t</i>
identity	.
	unsigned long Minimum: 0 Maximum: 65535 Scaling 1, 0, 0 (A,B,C): Scaled $1 * 10^0 * (Raw + 0)$ value: Resolution: 1

Formats: *SNVT_sec_status: text("%m %d", status, identity)*

SNVT_setting

Overview:

Setting control (function, setting, rotation) .

Used to tell an actuator functional block to invoke a stored setting/scene.

Details:

Standard: *yes*
 Resource Set: *Standard 00:00:00:00:00:00:00:00-0*
 Index: *117*
 Obsolete: *no*
 Size: *4*
 Programmatic Name: *SNVT_setting*

Neuron C Type:

Structure	
function	Setting control function (setting control function names) .
	<i>setting_t</i>
setting	Scene setting level (% of full level) .
	unsigned short Minimum: <i>0</i> Maximum: <i>200</i> Invalid: <i>255</i> Scaling <i>5, -1, 0</i> (A,B,C): Scaled value: <i>$5 * 10^{-1} * (Raw+0)$</i> Resolution: <i>0.5</i>
rotation	Rotation angle (degrees) .
	signed long Minimum: <i>-17999</i> Maximum: <i>18000</i> Invalid: <i>32767</i> Scaling <i>2, -2, 0</i> (A,B,C): Scaled value: <i>$2 * 10^{-2} * (Raw+0)$</i> Resolution: <i>0.02</i>

Formats: *SNVT_setting: text("%m %f%f", function, setting, rotation)*

Used by: *SCPTdefaultSetting SCPTpowerupState SFPTconstantLightController
 SFPToccupancyController SFPTpartitionWallController
 SFPTrealTimeBasedScheduler SFPTsceneController SFPTscenePanel
 SFPTsunblindActuator SFPTsunblindController SFPTswitch*

SNVT_sblnd_state

SNVT_smo_obscur

Overview:

Smoke obscuration (percent obscuration) .

Details:

Standard:	<i>yes</i>
Resource Set:	<i>Standard 00:00:00:00:00:00:00:00-0</i>
Index:	<i>129</i>
Obsolete:	<i>no</i>
Size:	<i>2</i>
Programmatic Name:	<i>SNVT_smo_obscur</i>
Neuron C Type:	unsigned long
	Minimum: <i>0</i>
	Maximum: <i>5000</i>
	Invalid: <i>65535</i>
	Scaling (A,B,C): <i>1, -3, 0</i>
	Scaled value: <i>$1 * 10^{-3} * (Raw + 0)$</i>
	Resolution: <i>0.001</i>
Formats:	<i>SNVT_smo_obscur: text("%f")</i>
Used by:	<i>SCPTsmokeDayAlrmLim SCPTsmokeDayPreAlrmLim</i> <i>SCPTsmokeNightAlrmLim SCPTsmokeNightPreAlrmLim</i> <i>SCPTsmokeNomSens</i>

SNVT_sound_db

Overview:

Sound level (dB) .

Details:

Standard:	<i>yes</i>
Resource Set:	<i>Standard 00:00:00:00:00:00:00:00-0</i>
Index:	<i>33</i>

Obsolete:	<i>no</i>
Size:	<i>2</i>
Programmatic Name:	<i>SNVT_sound_db</i>
Neuron C Type:	signed long
Minimum:	<i>-32768</i>
Maximum:	<i>32767</i>
Scaling (A,B,C):	<i>1, -2, 0</i>
Scaled value:	<i>$1 * 10^{-2} * (Raw + 0)$</i>
Resolution:	<i>0.01</i>
Formats:	<i>SNVT_sound_db: text("%f")</i>
Used by:	<i>SCPTaudOutput</i>

SNVT_sound_db_f

Overview:

Sound level (dBspl) .

Details:

Standard:	<i>yes</i>
Resource Set:	<i>Standard 00:00:00:00:00:00:00:00-0</i>
Index:	<i>61</i>
Obsolete:	<i>no</i>
Size:	<i>4</i>
Programmatic Name:	<i>SNVT_sound_db_f</i>
Neuron C Type:	float
Minimum:	<i>-3.4028234663853E+038</i>
Maximum:	<i>3.4028234663853E+038</i>
Formats:	<i>SNVT_sound_db_f: text("%f")</i>

SNVT_speed

Overview:

Linear velocity (meters/second) .

Details:

Standard: *yes*
 Resource Set: *Standard 00:00:00:00:00:00:00:00-0*
 Index: *34*
 Obsolete: *no*
 Size: *2*
 Programmatic Name: *SNVT_speed*
 Neuron C Type: *unsigned long*
 Minimum: *0*
 Maximum: *65535*
 Scaling (A,B,C): *1, -1, 0*
 Scaled value: *$1 * 10^{-1} * (Raw + 0)$*
 Resolution: *0.1*
 Formats: *SNVT_speed#SI: text("%f")*
 *SNVT_speed#US: text("%f", *196.85+0(0:972))*
 *SNVT_speed#US_sec: text("%f", *3.2808+0(0:852))*
 Used by: *SFPTsunblindController*

SNVT_speed_f

Overview:

Linear velocity (meters/second) .

Details:

Standard: *yes*
 Resource Set: *Standard 00:00:00:00:00:00:00:00-0*
 Index: *62*
 Obsolete: *no*
 Size: *4*
 Programmatic Name: *SNVT_speed_f*
 Neuron C Type: *float*
 Minimum: *-3.4028234663853E+038*
 Maximum: *3.4028234663853E+038*
 Formats: *SNVT_speed_f#SI: text("%f")*
 *SNVT_speed_f#US: text("%f", *196.85+0(0:972))*
 *SNVT_speed_f#US_sec: text("%f", *3.2808+0(0:852))*

SNVT_speed_mil

Overview:

Linear velocity (meters/second) .

Details:

Standard:	<i>yes</i>
Resource Set:	<i>Standard 00:00:00:00:00:00:00:00-0</i>
Index:	<i>35</i>
Obsolete:	<i>no</i>
Size:	<i>2</i>
Programmatic Name:	<i>SNVT_speed_mil</i>
Neuron C Type:	unsigned long
	Minimum: <i>0</i>
	Maximum: <i>65535</i>
	Scaling (A,B,C): <i>1, -3, 0</i>
	Scaled value: <i>$1 * 10^{-3} * (Raw + 0)$</i>
	Resolution: <i>0.001</i>
Formats:	<i>SNVT_speed_mil#SI: text("%f")</i>
	<i>SNVT_speed_mil#US: text("%f", *3.2808+0(0:852))</i>
Used by:	<i>SFPTairVelocitySensor</i>

SNVT_state

Overview:

State vector (16 individual bit values) . Each state is a boolean single bit value. SNVT_state_64 is preferred.

Each state is a Boolean, single-bit value. SNVT_state_64 is preferred.

Details:

Standard:	<i>yes</i>
Resource Set:	<i>Standard 00:00:00:00:00:00:00:00-0</i>
Index:	<i>83</i>
Obsolete:	<i>no</i>
Size:	<i>2</i>

Programmatic Name: *SNVT_state*

Neuron C Type:

Structure	
bit0	State bit 0 (boolean) .
	bitfield Signed: <i>no</i> Width: <i>1</i> Offset: <i>0</i> Minimum: <i>0</i> Maximum: <i>1</i> Scaling <i>1, 0, 0</i> (A,B,C): Scaled value: $1 * 10^0 * (Raw + 0)$ Resolution: <i>1</i>
bit1	State bit 1 (boolean) .
	bitfield Signed: <i>no</i> Width: <i>1</i> Offset: <i>1</i> Minimum: <i>0</i> Maximum: <i>1</i> Scaling <i>1, 0, 0</i> (A,B,C): Scaled value: $1 * 10^0 * (Raw + 0)$ Resolution: <i>1</i>
bit2	State bit 2 (boolean) .
	bitfield Signed: <i>no</i> Width: <i>1</i> Offset: <i>2</i> Minimum: <i>0</i> Maximum: <i>1</i> Scaling <i>1, 0, 0</i> (A,B,C): Scaled value: $1 * 10^0 * (Raw + 0)$ Resolution: <i>1</i>
bit3	State bit 3 (boolean) .
	bitfield Signed: <i>no</i> Width: <i>1</i> Offset: <i>3</i> Minimum: <i>0</i> Maximum: <i>1</i>

	Scaling <i>1, 0, 0</i> (A,B,C): Scaled $1 * 10^0 * (Raw + 0)$ value: Resolution: <i>1</i>
bit4	State bit 4 (boolean) .
	bitfield Signed: <i>no</i> Width: <i>1</i> Offset: <i>4</i> Minimum: <i>0</i> Maximum: <i>1</i> Scaling <i>1, 0, 0</i> (A,B,C): Scaled $1 * 10^0 * (Raw + 0)$ value: Resolution: <i>1</i>
bit5	State bit 5 (boolean) .
	bitfield Signed: <i>no</i> Width: <i>1</i> Offset: <i>5</i> Minimum: <i>0</i> Maximum: <i>1</i> Scaling <i>1, 0, 0</i> (A,B,C): Scaled $1 * 10^0 * (Raw + 0)$ value: Resolution: <i>1</i>
bit6	State bit 6 (boolean) .
	bitfield Signed: <i>no</i> Width: <i>1</i> Offset: <i>6</i> Minimum: <i>0</i> Maximum: <i>1</i> Scaling <i>1, 0, 0</i> (A,B,C): Scaled $1 * 10^0 * (Raw + 0)$ value: Resolution: <i>1</i>
bit7	State bit 7 (boolean) .
	bitfield Signed: <i>no</i> Width: <i>1</i>

	Offset: 7 Minimum: 0 Maximum: 1 Scaling 1, 0, 0 (A,B,C): Scaled value: $1 * 10^0 * (Raw + 0)$ Resolution: 1
bit8	State bit 8 (boolean) .
	bitfield Signed: no Width: 1 Offset: 0 Minimum: 0 Maximum: 1 Scaling 1, 0, 0 (A,B,C): Scaled value: $1 * 10^0 * (Raw + 0)$ Resolution: 1
bit9	State bit 9 (boolean) .
	bitfield Signed: no Width: 1 Offset: 1 Minimum: 0 Maximum: 1 Scaling 1, 0, 0 (A,B,C): Scaled value: $1 * 10^0 * (Raw + 0)$ Resolution: 1
bit10	State bit 10 (boolean) .
	bitfield Signed: no Width: 1 Offset: 2 Minimum: 0 Maximum: 1 Scaling 1, 0, 0 (A,B,C): Scaled value: $1 * 10^0 * (Raw + 0)$ Resolution: 1
bit11	State bit 11 (boolean) .

	bitfield Signed: <i>no</i> Width: <i>1</i> Offset: <i>3</i> Minimum: <i>0</i> Maximum: <i>1</i> Scaling <i>1, 0, 0</i> (A,B,C): Scaled $1 * 10^0 * (Raw + 0)$ value: Resolution: <i>1</i>
bit12	State bit 12 (boolean) .
	bitfield Signed: <i>no</i> Width: <i>1</i> Offset: <i>4</i> Minimum: <i>0</i> Maximum: <i>1</i> Scaling <i>1, 0, 0</i> (A,B,C): Scaled $1 * 10^0 * (Raw + 0)$ value: Resolution: <i>1</i>
bit13	State bit 13 (boolean) .
	bitfield Signed: <i>no</i> Width: <i>1</i> Offset: <i>5</i> Minimum: <i>0</i> Maximum: <i>1</i> Scaling <i>1, 0, 0</i> (A,B,C): Scaled $1 * 10^0 * (Raw + 0)$ value: Resolution: <i>1</i>
bit14	State bit 14 (boolean) .
	bitfield Signed: <i>no</i> Width: <i>1</i> Offset: <i>6</i> Minimum: <i>0</i> Maximum: <i>1</i> Scaling <i>1, 0, 0</i> (A,B,C): Scaled $1 * 10^0 * (Raw + 0)$ value:

	Resolution: <i>1</i>
bit15	State bit 15 (boolean) .
	bitfield Signed: <i>no</i> Width: <i>1</i> Offset: <i>7</i> Minimum: <i>0</i> Maximum: <i>1</i> Scaling <i>1, 0, 0</i> (A,B,C): Scaled <i>1 * 10⁰ *(Raw+0)</i> value: Resolution: <i>1</i>

Formats: *SNVT_state: text("%d,%d,%d,%d,%d,%d,%d,%d,%d,%d,%d,%d,%d,%d,%d,%d,%d", bit0, bit1, bit2, bit3, bit4, bit5, bit6, bit7, bit8, bit9, bit10, bit11, bit12, bit13, bit14, bit15)*

SNVT_state#LO: text("%d%d%d%d%d%d%d%d%d%d%d%d%d%d%d", bit0, bit1, bit2, bit3, bit4, bit5, bit6, bit7, bit8, bit9, bit10, bit11, bit12, bit13, bit14, bit15)

Used by: *SCPTdayNightCntrl SCPTdirection SFPTgeneratorSet
SFPTrefrigDisplayCaseControllerEvaporator
SFPTrefrigDisplayCaseControllerThermostat*

SNVT_state_64

Overview:

State vector (64 individual bit values) . Each state is a boolean single-bit value.

Each state is a Boolean, single-bit value. SNVT_state_64 should not be used to represent multiple, unrelated fields.

Details:

Standard: *yes*
Resource Set: *Standard 00:00:00:00:00:00:00:00-0*
Index: *165*
Obsolete: *no*
Size: *8*
Programmatic Name: *SNVT_state_64*

Neuron C Type:

Structure	
bit0	State bit 0 (boolean) .

	bitfield Signed: <i>no</i> Width: <i>1</i> Offset: <i>0</i> Minimum: <i>0</i> Maximum: <i>1</i> Scaling <i>1, 0, 0</i> (A,B,C): Scaled $1 * 10^0 * (Raw + 0)$ value: Resolution: <i>1</i>
bit1	State bit 1 (boolean) .
	bitfield Signed: <i>no</i> Width: <i>1</i> Offset: <i>1</i> Minimum: <i>0</i> Maximum: <i>1</i> Scaling <i>1, 0, 0</i> (A,B,C): Scaled $1 * 10^0 * (Raw + 0)$ value: Resolution: <i>1</i>
bit2	State bit 2 (boolean) .
	bitfield Signed: <i>no</i> Width: <i>1</i> Offset: <i>2</i> Minimum: <i>0</i> Maximum: <i>1</i> Scaling <i>1, 0, 0</i> (A,B,C): Scaled $1 * 10^0 * (Raw + 0)$ value: Resolution: <i>1</i>
bit3	State bit 3 (boolean) .
	bitfield Signed: <i>no</i> Width: <i>1</i> Offset: <i>3</i> Minimum: <i>0</i> Maximum: <i>1</i> Scaling <i>1, 0, 0</i> (A,B,C): Scaled $1 * 10^0 * (Raw + 0)$ value:

	Resolution: <i>1</i>
bit4	State bit 4 (boolean) .
	bitfield Signed: <i>no</i> Width: <i>1</i> Offset: <i>4</i> Minimum: <i>0</i> Maximum: <i>1</i> Scaling <i>1, 0, 0</i> (A,B,C): Scaled value: $1 * 10^0 * (Raw + 0)$ Resolution: <i>1</i>
bit5	State bit 5 (boolean) .
	bitfield Signed: <i>no</i> Width: <i>1</i> Offset: <i>5</i> Minimum: <i>0</i> Maximum: <i>1</i> Scaling <i>1, 0, 0</i> (A,B,C): Scaled value: $1 * 10^0 * (Raw + 0)$ Resolution: <i>1</i>
bit6	State bit 6 (boolean) .
	bitfield Signed: <i>no</i> Width: <i>1</i> Offset: <i>6</i> Minimum: <i>0</i> Maximum: <i>1</i> Scaling <i>1, 0, 0</i> (A,B,C): Scaled value: $1 * 10^0 * (Raw + 0)$ Resolution: <i>1</i>
bit7	State bit 7 (boolean) .
	bitfield Signed: <i>no</i> Width: <i>1</i> Offset: <i>7</i> Minimum: <i>0</i> Maximum: <i>1</i> Scaling <i>1, 0, 0</i>

	(A,B,C): Scaled $1 * 10^0 * (Raw+0)$ value: Resolution: 1
bit8	State bit 8 (boolean) .
	bitfield Signed: <i>no</i> Width: 1 Offset: 0 Minimum: 0 Maximum: 1 Scaling $1, 0, 0$ (A,B,C): Scaled $1 * 10^0 * (Raw+0)$ value: Resolution: 1
bit9	State bit 9 (boolean) .
	bitfield Signed: <i>no</i> Width: 1 Offset: 1 Minimum: 0 Maximum: 1 Scaling $1, 0, 0$ (A,B,C): Scaled $1 * 10^0 * (Raw+0)$ value: Resolution: 1
bit10	State bit 10 (boolean) .
	bitfield Signed: <i>no</i> Width: 1 Offset: 2 Minimum: 0 Maximum: 1 Scaling $1, 0, 0$ (A,B,C): Scaled $1 * 10^0 * (Raw+0)$ value: Resolution: 1
bit11	State bit 11 (boolean) .
	bitfield Signed: <i>no</i> Width: 1 Offset: 3

	Minimum: <i>0</i> Maximum: <i>1</i> Scaling <i>1, 0, 0</i> (A,B,C): Scaled <i>$1 * 10^0 * (Raw+0)$</i> value: Resolution: <i>1</i>
bit12	State bit 12 (boolean) .
	bitfield Signed: <i>no</i> Width: <i>1</i> Offset: <i>4</i> Minimum: <i>0</i> Maximum: <i>1</i> Scaling <i>1, 0, 0</i> (A,B,C): Scaled <i>$1 * 10^0 * (Raw+0)$</i> value: Resolution: <i>1</i>
bit13	State bit 13 (boolean) .
	bitfield Signed: <i>no</i> Width: <i>1</i> Offset: <i>5</i> Minimum: <i>0</i> Maximum: <i>1</i> Scaling <i>1, 0, 0</i> (A,B,C): Scaled <i>$1 * 10^0 * (Raw+0)$</i> value: Resolution: <i>1</i>
bit14	State bit 14 (boolean) .
	bitfield Signed: <i>no</i> Width: <i>1</i> Offset: <i>6</i> Minimum: <i>0</i> Maximum: <i>1</i> Scaling <i>1, 0, 0</i> (A,B,C): Scaled <i>$1 * 10^0 * (Raw+0)$</i> value: Resolution: <i>1</i>
bit15	State bit 15 (boolean) .
	bitfield

	Signed: <i>no</i> Width: <i>1</i> Offset: <i>7</i> Minimum: <i>0</i> Maximum: <i>1</i> Scaling <i>1, 0, 0</i> (A,B,C): Scaled <i>1 * 10⁰ *(Raw+0)</i> value: Resolution: <i>1</i>
bit16	State bit 16 (boolean) .
	bitfield Signed: <i>no</i> Width: <i>1</i> Offset: <i>0</i> Minimum: <i>0</i> Maximum: <i>1</i> Scaling <i>1, 0, 0</i> (A,B,C): Scaled <i>1 * 10⁰ *(Raw+0)</i> value: Resolution: <i>1</i>
bit17	State bit 17 (boolean) .
	bitfield Signed: <i>no</i> Width: <i>1</i> Offset: <i>1</i> Minimum: <i>0</i> Maximum: <i>1</i> Scaling <i>1, 0, 0</i> (A,B,C): Scaled <i>1 * 10⁰ *(Raw+0)</i> value: Resolution: <i>1</i>
bit18	State bit 18 (boolean) .
	bitfield Signed: <i>no</i> Width: <i>1</i> Offset: <i>2</i> Minimum: <i>0</i> Maximum: <i>1</i> Scaling <i>1, 0, 0</i> (A,B,C): Scaled <i>1 * 10⁰ *(Raw+0)</i> value: Resolution: <i>1</i>

bit19	State bit 19 (boolean) .
	bitfield Signed: <i>no</i> Width: <i>1</i> Offset: <i>3</i> Minimum: <i>0</i> Maximum: <i>1</i> Scaling <i>1, 0, 0</i> (A,B,C): Scaled $1 * 10^0 * (Raw + 0)$ value: Resolution: <i>1</i>
bit20	State bit 20 (boolean) .
	bitfield Signed: <i>no</i> Width: <i>1</i> Offset: <i>4</i> Minimum: <i>0</i> Maximum: <i>1</i> Scaling <i>1, 0, 0</i> (A,B,C): Scaled $1 * 10^0 * (Raw + 0)$ value: Resolution: <i>1</i>
bit21	State bit 21 (boolean) .
	bitfield Signed: <i>no</i> Width: <i>1</i> Offset: <i>5</i> Minimum: <i>0</i> Maximum: <i>1</i> Scaling <i>1, 0, 0</i> (A,B,C): Scaled $1 * 10^0 * (Raw + 0)$ value: Resolution: <i>1</i>
bit22	State bit 22 (boolean) .
	bitfield Signed: <i>no</i> Width: <i>1</i> Offset: <i>6</i> Minimum: <i>0</i> Maximum: <i>1</i> Scaling <i>1, 0, 0</i> (A,B,C): Scaled

	value: $1 * 10^0 * (Raw + 0)$ Resolution: 1
bit23	State bit 23 (boolean) .
	bitfield Signed: <i>no</i> Width: 1 Offset: 7 Minimum: 0 Maximum: 1 Scaling $1, 0, 0$ (A,B,C): Scaled $1 * 10^0 * (Raw + 0)$ value: Resolution: 1
bit24	State bit 24 (boolean) .
	bitfield Signed: <i>no</i> Width: 1 Offset: 0 Minimum: 0 Maximum: 1 Scaling $1, 0, 0$ (A,B,C): Scaled $1 * 10^0 * (Raw + 0)$ value: Resolution: 1
bit25	State bit 25 (boolean) .
	bitfield Signed: <i>no</i> Width: 1 Offset: 1 Minimum: 0 Maximum: 1 Scaling $1, 0, 0$ (A,B,C): Scaled $1 * 10^0 * (Raw + 0)$ value: Resolution: 1
bit26	State bit 26 (boolean) .
	bitfield Signed: <i>no</i> Width: 1 Offset: 2 Minimum: 0 Maximum: 1

	Scaling <i>1, 0, 0</i> (A,B,C): Scaled $1 * 10^0 * (Raw + 0)$ value: Resolution: <i>1</i>
bit27	State bit 27 (boolean) .
	bitfield Signed: <i>no</i> Width: <i>1</i> Offset: <i>3</i> Minimum: <i>0</i> Maximum: <i>1</i> Scaling <i>1, 0, 0</i> (A,B,C): Scaled $1 * 10^0 * (Raw + 0)$ value: Resolution: <i>1</i>
bit28	State bit 28 (boolean) .
	bitfield Signed: <i>no</i> Width: <i>1</i> Offset: <i>4</i> Minimum: <i>0</i> Maximum: <i>1</i> Scaling <i>1, 0, 0</i> (A,B,C): Scaled $1 * 10^0 * (Raw + 0)$ value: Resolution: <i>1</i>
bit29	State bit 29 (boolean) .
	bitfield Signed: <i>no</i> Width: <i>1</i> Offset: <i>5</i> Minimum: <i>0</i> Maximum: <i>1</i> Scaling <i>1, 0, 0</i> (A,B,C): Scaled $1 * 10^0 * (Raw + 0)$ value: Resolution: <i>1</i>
bit30	State bit 30 (boolean) .
	bitfield Signed: <i>no</i> Width: <i>1</i>

	Offset: 6 Minimum: 0 Maximum: 1 Scaling 1, 0, 0 (A,B,C): Scaled value: $1 * 10^0 * (Raw + 0)$ Resolution: 1
bit31	State bit 31 (boolean) .
	bitfield Signed: no Width: 1 Offset: 7 Minimum: 0 Maximum: 1 Scaling 1, 0, 0 (A,B,C): Scaled value: $1 * 10^0 * (Raw + 0)$ Resolution: 1
bit32	State bit 32 (boolean) .
	bitfield Signed: no Width: 1 Offset: 0 Minimum: 0 Maximum: 1 Scaling 1, 0, 0 (A,B,C): Scaled value: $1 * 10^0 * (Raw + 0)$ Resolution: 1
bit33	State bit 33 (boolean) .
	bitfield Signed: no Width: 1 Offset: 1 Minimum: 0 Maximum: 1 Scaling 1, 0, 0 (A,B,C): Scaled value: $1 * 10^0 * (Raw + 0)$ Resolution: 1
bit34	State bit 34 (boolean) .

	bitfield Signed: <i>no</i> Width: <i>1</i> Offset: <i>2</i> Minimum: <i>0</i> Maximum: <i>1</i> Scaling <i>1, 0, 0</i> (A,B,C): Scaled $1 * 10^0 * (Raw + 0)$ value: Resolution: <i>1</i>
bit35	State bit 35 (boolean) .
	bitfield Signed: <i>no</i> Width: <i>1</i> Offset: <i>3</i> Minimum: <i>0</i> Maximum: <i>1</i> Scaling <i>1, 0, 0</i> (A,B,C): Scaled $1 * 10^0 * (Raw + 0)$ value: Resolution: <i>1</i>
bit36	State bit 36 (boolean) .
	bitfield Signed: <i>no</i> Width: <i>1</i> Offset: <i>4</i> Minimum: <i>0</i> Maximum: <i>1</i> Scaling <i>1, 0, 0</i> (A,B,C): Scaled $1 * 10^0 * (Raw + 0)$ value: Resolution: <i>1</i>
bit37	State bit 37 (boolean) .
	bitfield Signed: <i>no</i> Width: <i>1</i> Offset: <i>5</i> Minimum: <i>0</i> Maximum: <i>1</i> Scaling <i>1, 0, 0</i> (A,B,C): Scaled $1 * 10^0 * (Raw + 0)$ value:

	Resolution: <i>1</i>
bit38	State bit 38 (boolean) .
	bitfield Signed: <i>no</i> Width: <i>1</i> Offset: <i>6</i> Minimum: <i>0</i> Maximum: <i>1</i> Scaling <i>1, 0, 0</i> (A,B,C): Scaled value: $1 * 10^0 * (Raw + 0)$ Resolution: <i>1</i>
bit39	State bit 39 (boolean) .
	bitfield Signed: <i>no</i> Width: <i>1</i> Offset: <i>7</i> Minimum: <i>0</i> Maximum: <i>1</i> Scaling <i>1, 0, 0</i> (A,B,C): Scaled value: $1 * 10^0 * (Raw + 0)$ Resolution: <i>1</i>
bit40	State bit 40 (boolean) .
	bitfield Signed: <i>no</i> Width: <i>1</i> Offset: <i>0</i> Minimum: <i>0</i> Maximum: <i>1</i> Scaling <i>1, 0, 0</i> (A,B,C): Scaled value: $1 * 10^0 * (Raw + 0)$ Resolution: <i>1</i>
bit41	State bit 41 (boolean) .
	bitfield Signed: <i>no</i> Width: <i>1</i> Offset: <i>1</i> Minimum: <i>0</i> Maximum: <i>1</i> Scaling <i>1, 0, 0</i>

	(A,B,C): Scaled $1 * 10^0 * (Raw+0)$ value: Resolution: 1
bit42	State bit 42 (boolean) .
	bitfield Signed: <i>no</i> Width: 1 Offset: 2 Minimum: 0 Maximum: 1 Scaling $1, 0, 0$ (A,B,C): Scaled $1 * 10^0 * (Raw+0)$ value: Resolution: 1
bit43	State bit 43 (boolean) .
	bitfield Signed: <i>no</i> Width: 1 Offset: 3 Minimum: 0 Maximum: 1 Scaling $1, 0, 0$ (A,B,C): Scaled $1 * 10^0 * (Raw+0)$ value: Resolution: 1
bit44	State bit 44 (boolean) .
	bitfield Signed: <i>no</i> Width: 1 Offset: 4 Minimum: 0 Maximum: 1 Scaling $1, 0, 0$ (A,B,C): Scaled $1 * 10^0 * (Raw+0)$ value: Resolution: 1
bit45	State bit 45 (boolean) .
	bitfield Signed: <i>no</i> Width: 1 Offset: 5

	Minimum: <i>0</i> Maximum: <i>1</i> Scaling <i>1, 0, 0</i> (A,B,C): Scaled <i>$1 * 10^0 * (Raw+0)$</i> value: Resolution: <i>1</i>
bit46	State bit 46 (boolean) .
	bitfield Signed: <i>no</i> Width: <i>1</i> Offset: <i>6</i> Minimum: <i>0</i> Maximum: <i>1</i> Scaling <i>1, 0, 0</i> (A,B,C): Scaled <i>$1 * 10^0 * (Raw+0)$</i> value: Resolution: <i>1</i>
bit47	State bit 47 (boolean) .
	bitfield Signed: <i>no</i> Width: <i>1</i> Offset: <i>7</i> Minimum: <i>0</i> Maximum: <i>1</i> Scaling <i>1, 0, 0</i> (A,B,C): Scaled <i>$1 * 10^0 * (Raw+0)$</i> value: Resolution: <i>1</i>
bit48	State bit 48 (boolean) .
	bitfield Signed: <i>no</i> Width: <i>1</i> Offset: <i>0</i> Minimum: <i>0</i> Maximum: <i>1</i> Scaling <i>1, 0, 0</i> (A,B,C): Scaled <i>$1 * 10^0 * (Raw+0)$</i> value: Resolution: <i>1</i>
bit49	State bit 49 (boolean) .
	bitfield

	Signed: <i>no</i> Width: <i>1</i> Offset: <i>1</i> Minimum: <i>0</i> Maximum: <i>1</i> Scaling (A,B,C): <i>1, 0, 0</i> Scaled value: $1 * 10^0 * (Raw + 0)$ Resolution: <i>1</i>
bit50	State bit 50 (boolean) .
	bitfield Signed: <i>no</i> Width: <i>1</i> Offset: <i>2</i> Minimum: <i>0</i> Maximum: <i>1</i> Scaling (A,B,C): <i>1, 0, 0</i> Scaled value: $1 * 10^0 * (Raw + 0)$ Resolution: <i>1</i>
bit51	State bit 51 (boolean) .
	bitfield Signed: <i>no</i> Width: <i>1</i> Offset: <i>3</i> Minimum: <i>0</i> Maximum: <i>1</i> Scaling (A,B,C): <i>1, 0, 0</i> Scaled value: $1 * 10^0 * (Raw + 0)$ Resolution: <i>1</i>
bit52	State bit 52 (boolean) .
	bitfield Signed: <i>no</i> Width: <i>1</i> Offset: <i>4</i> Minimum: <i>0</i> Maximum: <i>1</i> Scaling (A,B,C): <i>1, 0, 0</i> Scaled value: $1 * 10^0 * (Raw + 0)$ Resolution: <i>1</i>

bit53	State bit 53 (boolean) .
	bitfield Signed: <i>no</i> Width: <i>1</i> Offset: <i>5</i> Minimum: <i>0</i> Maximum: <i>1</i> Scaling <i>1, 0, 0</i> (A,B,C): Scaled $1 * 10^0 * (Raw + 0)$ value: Resolution: <i>1</i>
bit54	State bit 54 (boolean) .
	bitfield Signed: <i>no</i> Width: <i>1</i> Offset: <i>6</i> Minimum: <i>0</i> Maximum: <i>1</i> Scaling <i>1, 0, 0</i> (A,B,C): Scaled $1 * 10^0 * (Raw + 0)$ value: Resolution: <i>1</i>
bit55	State bit 55 (boolean) .
	bitfield Signed: <i>no</i> Width: <i>1</i> Offset: <i>7</i> Minimum: <i>0</i> Maximum: <i>1</i> Scaling <i>1, 0, 0</i> (A,B,C): Scaled $1 * 10^0 * (Raw + 0)$ value: Resolution: <i>1</i>
bit56	State bit 56 (boolean) .
	bitfield Signed: <i>no</i> Width: <i>1</i> Offset: <i>0</i> Minimum: <i>0</i> Maximum: <i>1</i> Scaling <i>1, 0, 0</i> (A,B,C): Scaled

	value: $1 * 10^0 * (Raw + 0)$ Resolution: 1
bit57	State bit 57 (boolean) .
	bitfield Signed: <i>no</i> Width: 1 Offset: 1 Minimum: 0 Maximum: 1 Scaling $1, 0, 0$ (A,B,C): Scaled $1 * 10^0 * (Raw + 0)$ value: Resolution: 1
bit58	State bit 58 (boolean) .
	bitfield Signed: <i>no</i> Width: 1 Offset: 2 Minimum: 0 Maximum: 1 Scaling $1, 0, 0$ (A,B,C): Scaled $1 * 10^0 * (Raw + 0)$ value: Resolution: 1
bit59	State bit 59 (boolean) .
	bitfield Signed: <i>no</i> Width: 1 Offset: 3 Minimum: 0 Maximum: 1 Scaling $1, 0, 0$ (A,B,C): Scaled $1 * 10^0 * (Raw + 0)$ value: Resolution: 1
bit60	State bit 60 (boolean) .
	bitfield Signed: <i>no</i> Width: 1 Offset: 4 Minimum: 0 Maximum: 1

bit28, bit29, bit30, bit31, bit32, bit33, bit34, bit35, bit36, bit37, bit38, bit39, bit40, bit41, bit42, bit43, bit44, bit45, bit46, bit47, bit48, bit49, bit50, bit51, bit52, bit53, bit54, bit55, bit56, bit57, bit58, bit59, bit60, bit61, bit62, bit63)

Used by: *SFPTdeviceMonitor*

SNVT_str_asc

Overview:

Character string (30 characters max) (ASCII character string) .

An ASCII string of up to 30 characters plus a terminator (ASCII 0) that must be stored within the 'ascii' field, immediately after the last character of the string. The maximum string length is therefore 30 characters plus the terminator. SNVT_str_asc must only be used for communicating textual information to system integrators and operators. It must not be used for encoding commands or state information for other devices. Commands and state information must be encoded in appropriate scalar or structure network variable types.

Details:

Standard: *yes*
 Resource Set: *Standard 00:00:00:00:00:00:00:00-0*
 Index: *36*
 Obsolete: *no*
 Size: *31*
 Programmatic Name: *SNVT_str_asc*
 Neuron C Type:

Structure	
ascii	ASCII character string (array of 31 characters) .
	unsigned char [31] Minimum: 0 Maximum: 255 Scaling 1, 0, 0 (A,B,C): Scaled $1 * 10^0 * (Raw + 0)$ value: Resolution: 1

Formats: *SNVT_str_asc: text("%s",ascii)*

Used by: *SCPTactuatorType SCPTdevListDesc SCPTdialString SCPTfireTxt1 SCPTfireTxt2 SCPTfireTxt3 SCPTifaceDesc SCPTlocation SCPToemType SCPTpartNumber SCPTregName SCPTserialNumber SFPTchiller SFPTelevatorsPositionIndicator SFPTentryExit SFPTmodemController SFPTtelephoneDirectory*

SNVT_str_int

Overview:

Wide character string with locale code (15 characters max) (Wide character string) .

A wide-character string of up to 14 characters plus a terminator (ASCII 0) that must be stored within the wide_char field, immediately after the last character of the string. The maximum string length is therefore 14 characters plus the terminator. SNVT_str_asc must only be used for communicating textual information to system integrators and operators. It must not be used for encoding commands or state information for other devices. Commands and state information must be encoded in appropriate scalar or structure network variable types.

Details:

Standard: *yes*
 Resource Set: *Standard 00:00:00:00:00:00:00:00-0*
 Index: *37*
 Obsolete: *no*
 Size: *31*
 Programmatic Name: *SNVT_str_int*

Neuron C Type:

Structure	
char_set	Locale code (code value) .
	unsigned short Minimum: 0 Maximum: 255 Scaling 1, 0, 0 (A,B,C): Scaled value: $1 * 10^0 * (Raw + 0)$ Resolution: 1
wide_char	Wide character string (array of 15 wide characters) .
	unsigned long [15] Minimum: 0 Maximum: 65535 Scaling 1, 0, 0 (A,B,C): Scaled value: $1 * 10^0 * (Raw + 0)$ Resolution: 1

Formats: *SNVT_str_int: text("%d,%d,%d,%d,%d,%d,%d,%d,%d,%d,%d,%d,%d,%d,%d,%d,%d,%d", char_set, wide_char[0], wide_char[1], wide_char[2], wide_char[3], wide_char[4], wide_char[5], wide_char[6], wide_char[7],*

wide_char[8], wide_char[9], wide_char[10], wide_char[11], wide_char[12], wide_char[13], wide_char[14])

SNVT_str_int#LO: text("%d|%d|%d|%d|%d|%d|%d|%d|%d|%d|%d|%d|%d|%d|%d|%d|%d|%d|%d|%d", char_set, wide_char[0], wide_char[1], wide_char[2], wide_char[3], wide_char[4], wide_char[5], wide_char[6], wide_char[7], wide_char[8], wide_char[9], wide_char[10], wide_char[11], wide_char[12], wide_char[13], wide_char[14])

SNVT_switch

Overview:

Switch (value, state) .

A structure reporting a percentage level or load value and a discrete on/off state. Separate fields report the percentage value and state. This type should be used for both discrete (on/off) and analog control.

The value field is used to control the load's value, i.e. position, speed, or intensity, the state field being used to control whether the load is on or off (enabled or disabled).

When used as the output of a discrete sensor device, the OFF state is represented by a SNVT_switch network variable with state = FALSE and value = 0. The other discrete states are represented by state = TRUE and value > 0. When used as the output of a two-state sensor device, the ON state is represented by state = TRUE and value = 200 (meaning 100%).

When used as the input of a two-state discrete actuator, a SNVT_switch network variable with state = TRUE will be interpreted as the ON state if value > 0, and as the OFF state if value = 0. Additionally, a SNVT_switch input network variable with state = FALSE should be interpreted as the OFF state, whether or not value = 0. A state value of 0xFF indicates the switch value is undefined.

Input Network Variable

value (raw)	state	2-state interpretation
any	0	off (0; 0)
0	1	off (0; 1)
> 0	1	on (200; 1)
any	-1 (0xFF)	invalid (no action)

Output Network Variable

value (raw)	state	2-state interpretation
0	0	off
200 (0xC8)	1	on
0 .. 200 (0 .. 0xC8) (any valid value)	-1 (0xFF)	invalid (no action)

Details:

Standard: *yes*
 Resource Set: *Standard 00:00:00:00:00:00:00:00-0*
 Index: *95*
 Obsolete: *no*
 Size: *2*
 Programmatic Name: *SNVT_switch*
 Neuron C Type:

Structure	
value	Value (% of full level) .
	unsigned short Minimum: <i>0</i> Maximum: <i>200</i> Scaling <i>5, -1, 0</i> (A,B,C): Scaled <i>$5 * 10^{-1} * (Raw+0)$</i> value: Resolution: <i>0.5</i>
state	State (state code) . This field can either be -1 (NULL), 0 (OFF), or 1 (ON).
	signed short Minimum: <i>-1</i> Maximum: <i>1</i> Invalid: <i>-1</i> Scaling <i>1, 0, 0</i> (A,B,C): Scaled <i>$1 * 10^0 * (Raw+0)$</i> value: Resolution: <i>1</i>

Formats: *SNVT_switch: text("%f %d", value, state)*

Used by: *SCPTaudibleLevel SCPTbrightness SCPTdefltBehave
 SCPTlinkPowerDetectEnable SCPTprimeVal SCPTpwrUpState
 SCPTscrollSpeed SCPTsecondVal SFPTaudibleFireIndicator
 SFPTautomaticTransferSwitch SFPTboilerController SFPTcalendar
 SFPTchannelContinuityMonitor SFPTchannelMonitor
 SFPTchilledCeilingController SFPTchiller
 SFPTclothesWasherDomestic SFPTconstantLightController
 SFPTdamperActuator SFPTdataLogger SFPTdischargeAirController
 SFPTelevatorsArrivalGong SFPTelevatorsDirectionLantern
 SFPTelevatorsFireSystemsPort SFPTelevatorsHallLantern
 SFPTelevatorsPositionIndicator SFPTelevatorsVoiceAnnouncer
 SFPTentryExit SFPTfanCoilUnit SFPTfrostSensor
 SFPTgeneratorSet SFPTheatPump SFPT HVAC Valve Positioner
 SFPTlampActuator SFPTmodemController SFPTnodeObject
 SFPToccupancyController SFPTpartitionWallController
 SFPTpullStationFireInitiator SFPTpumpController SFPTtrainSensor
 SFPTrefrigDisplayCaseControllerEvaporator SFPTsccAHU
 SFPTsccChilledCeiling SFPTsccCommandModule SFPTsccFanCoil*

SFPTsccHeatPump SFPTsccRadiator SFPTsccRooftop
SFPTsccSelfContained SFPTsccUnitVentilator SFPTsccVAV
SFPTsceneController SFPTscheduler
SFPTsmokeFireInitiatorConvent SFPTsmokeFireInitiatorIntelli
SFPTspaceComfortController SFPTsunblindActuator
SFPTsunblindController SFPTswitch SFPTthermalFireInitiator
SFPTthermostat SFPTunitVentilatorController
SFPTuniversalFireIndicator SFPTuniversalFireInitiator
SFPTvariableAirVolume SFPTvariableSpeedMotorDrive
SFPTvisibleFireIndicator

SNVT_switch_2

Overview:

Switch with scene and setting control. An enhanced version of SNVT_switch with scene and setting controls similar to SNVT_scene and SNVT_setting.

Details:

Standard: *yes*
 Resource Set: *Standard 00:00:00:00:00:00:00:00-0*
 Index: *189*
 Obsolete: *no*
 Size: *3*
 Programmatic Name: *SNVT_switch_2*

Neuron C Type:

Structure	
state	Switch state. Switch state; maybe a state of the switch or other switch properties such as scene, occupancy state, and level multiplier
	<i>switch_state_t</i>
setting	Switch setting. Sets or reports the level, change, or angle for a switch
Union	
value	Value. Percent of full level when state is on. Reports last level for outputs when state is off.
	unsigned short Minimum: <i>0</i> Maximum: <i>200</i> Invalid: <i>255</i> Scaling <i>5, -1, 0</i> (A,B,C): Scaled <i>$5 * 10^{-1} * (Raw + 0)$</i>

	value: Resolution: 0.5
change	Percent change.. Percent change to level
	unsigned short Minimum: 0 Maximum: 200 Invalid: 255 Scaling 5, -1, 0 (A,B,C): Scaled $5 * 10^{-1} * (Raw + 0)$ value: Resolution: 0.5
delay	On or off delay (seconds) (seconds) . Time delay before changing state to on or off.
	unsigned short Minimum: 0 Maximum: 254 Invalid: 255 Scaling 1, 0, 0 (A,B,C): Scaled $1 * 10^0 * (Raw + 0)$ value: Resolution: 1
group_number	Group number. Group number that is enabled or disabled by the SW_ENABLE_GROUP and SW_DISABLE_GROUP states in the state field; if 0, all groups are enabled or disabled
	unsigned short Minimum: 0 Maximum: 63 Invalid: 255 Scaling 1, 0, 0 (A,B,C): Scaled $1 * 10^0 * (Raw + 0)$ value: Resolution: 1
multiplier	Factor (percent) . Multiplier for the level
	<i>SNVT_multiplier_s</i>
angle	Rotation angle (degrees) . Rotation angle for devices that support a rotation setting such as blinds.

	signed short Minimum: <i>-90</i> Maximum: <i>90</i> Invalid: <i>-128</i> Scaling (A,B,C): <i>2, 0, 0</i> Scaled value: $2 * 10^0 * (Raw + 0)$ Resolution: <i>2</i>
fan_level	Fan level. Percent of full level fan speed when state is on. Reports last fan speed for outputs when state is off. Positive values represent the down direction, and negative values represent the up direction.
	signed short Minimum: <i>-100</i> Maximum: <i>100</i> Invalid: <i>-128</i> Scaling (A,B,C): <i>1, 0, 0</i> Scaled value: $1 * 10^0 * (Raw + 0)$ Resolution: <i>1</i>
scene_number	Scene number. Scene number that is applied based on the function specified in the state field.
	unsigned short Minimum: <i>1</i> Maximum: <i>255</i> Invalid: <i>0</i> Scaling (A,B,C): <i>1, 0, 0</i> Scaled value: $1 * 10^0 * (Raw + 0)$ Resolution: <i>1</i>

Used by:

*SFPTisiKeypad SFPTisiLampActuator SFPTisiOccupancySensor
SFPTisiSunblindActuator*

SNVT_telcom

Overview:

Telecomm states (telecomm state names) .

Details:

Standard: *yes*
 Resource Set: *Standard 00:00:00:00:00:00:00:00-0*
 Index: *38*
 Obsolete: *no*
 Size: *1*
 Programmatic Name: *SNVT_telcom*
 Neuron C Type: *telcom_states_t*
 Formats: *SNVT_telcom: text("%m")*
 Used by: *SFPTmodemController*

SNVT_temp

Overview:

Temperature (degrees Celsius) .

SNVT_temp represents tenths of a degree Celsius above -274°C. To get SNVT_temp units define a constant: C_to_K equal to 2740, which is added to temperature, expressed in tenths of degrees C.

Details:

Standard: *yes*
 Resource Set: *Standard 00:00:00:00:00:00:00:00-0*
 Index: *39*
 Obsolete: *no*
 Size: *2*
 Programmatic Name: *SNVT_temp*
 Neuron C Type: *unsigned long*
 Minimum: *0*
 Maximum: *65535*
 Scaling (A,B,C): *1, -1, -2740*
 Scaled value: *$1 * 10^{-1} * (Raw + -2740)$*
 Resolution: *0.1*
 Formats: *SNVT_temp#SI: text("%f", *1+0(0:854))*
 *SNVT_temp#US: text("%f", *1.8+32(0:855))*
 *SNVT_temp#US_diff: text("%f", *1.8+0(0:855))*
 Used by: *SCPTrefrigGlide SCPTthermThreshold SFPTgeneratorSet*
 SFPT HVACTempSensor SNVT_pump_sensor SNVT_pumpset_sn

SNVT_temp_diff_p

Overview:

Temp difference (degrees Celsius) .

Used to communicate temperature differences. e.g., a unit has two temperature probes A and B. The value to be reported is A – B. This type would be used allow another device to format the data from a network variable using this SNVT correctly when converting from °C to °F, if needed. This conversion requires a 32°F offset converting to or from Fahrenheit temperature values, but no offset when converting to or from differential Fahrenheit temperature values. If there was no differential temperature SNVT, then the conversion can only be accomplished by the front-end system, or HMI, knowing the particular network variable.

Details:

Standard:	<i>yes</i>
Resource Set:	<i>Standard 00:00:00:00:00:00:00:00-0</i>
Index:	<i>147</i>
Obsolete:	<i>no</i>
Size:	<i>2</i>
Programmatic Name:	<i>SNVT_temp_diff_p</i>
Neuron C Type:	signed long
	Minimum: <i>-32768</i>
	Maximum: <i>32767</i>
	Invalid: <i>32767</i>
	Scaling (A,B,C): <i>1, -2, 0</i>
	Scaled value: <i>$1 * 10^{-2} * (Raw + 0)$</i>
	Resolution: <i>0.01</i>
Formats:	<i>SNVT_temp_diff_p#SI: text("%f", *1+0(0:854))</i>
	<i>SNVT_temp_diff_p#US: text("%f", *1.8+0(0:855))</i>
Used by:	<i>SCPTdefrostDetect SCPTdiffTempSetpoint</i> <i>SCPTtemperatureHysteresis SCPTtempOffset</i>

SNVT_temp_f

Overview:

Temperature (degrees Celsius) .

Details:

Standard:	<i>yes</i>
Resource Set:	<i>Standard 00:00:00:00:00:00:00:00-0</i>
Index:	<i>63</i>
Obsolete:	<i>no</i>
Size:	<i>4</i>
Programmatic Name:	<i>SNVT_temp_f</i>
Neuron C Type:	<i>float</i>
	<i>Minimum: -273.17001342773</i>
	<i>Maximum: 3.4028234663853E+038</i>
Formats:	<i>SNVT_temp_f#SI: text("%f", *1+0(0:854))</i>
	<i>SNVT_temp_f#US: text("%f", *1.8+32(0:855))</i>
	<i>SNVT_temp_f#US_diff: text("%f", *1.8+0(0:855))</i>
Used by:	<i>SFPThvacTempSensor</i>

SNVT_temp_p

Overview:

Temperature (degrees Celsius) .

To be used for heating, ventilation, and air-conditioning (HVAC) applications.

Details:

Standard:	<i>yes</i>
Resource Set:	<i>Standard 00:00:00:00:00:00:00:00-0</i>
Index:	<i>105</i>
Obsolete:	<i>no</i>
Size:	<i>2</i>
Programmatic Name:	<i>SNVT_temp_p</i>
Neuron C Type:	<i>signed long</i>
	<i>Minimum: -27317</i>
	<i>Maximum: 32767</i>
	<i>Invalid: 32767</i>
	<i>Scaling (A,B,C): 1, -2, 0</i>
	<i>Scaled value: $1 * 10^{-2} * (Raw + 0)$</i>
	<i>Resolution: 0.01</i>
Formats:	<i>SNVT_temp_p#SI: text("%f", *1+0(0:854))</i>

*SNVT_temp_p#US: text("%f", *1.8+32(0:855))*

*SNVT_temp_p#US_diff: text("%f", *1.8+0(0:855))*

Used by:

*SCPTcoolingLockout SCPTcoolLowerSP SCPTcoolSetpt
SCPTcoolUpperSP SCPTcutOutValue SCPTdeltaNight
SCPTdiffNight SCPTdiffValue SCPTdischargeAirCoolingSetpoint
SCPTdischargeAirDewpointSetpoint
SCPTdischargeAirHeatingSetpoint SCPTheatingLockout
SCPTheatLowerSP SCPTheatSetpt SCPTheatUpperSP
SCPThighLimTemp SCPTlowLimTemp SCPTmaxDefrstTemp
SCPTmaxDischargeAirCoolingSetpoint
SCPTmaxDischargeAirHeatingSetpoint SCPTmaxRemoteTempSetpoint
SCPTminDeltaTemp SCPTminDischargeAirCoolingSetpoint
SCPTminDischargeAirHeatingSetpoint SCPTminRemoteTempSetpoint
SCPTmixedAirLowLimitSetpoint SCPTmixedAirTempSetpoint
SCPToffsetTemp SCPToutdoorAirTempSetpoint SCPTsuperHtRefInit
SCPTsuperHtRefMax SCPTsuperHtRefMin SFPTboilerController
SFPTchilledCeilingController SFPTchiller SFPTdamperActuator
SFPTdischargeAirController SFPTfanCoilUnit SFPTheatPump
SFPT HVAC Temp Sensor SFPTpumpController
SFPTrefrigDisplayCaseControllerDefrost
SFPTrefrigDisplayCaseControllerEvaporator
SFPTrefrigDisplayCaseControllerThermostat SFPTroofTopUnit
SFPTsccAHU SFPTsccChilledCeiling SFPTsccCommandModule
SFPTsccFanCoil SFPTsccHeatPump SFPTsccRadiator
SFPTsccRooftop SFPTsccSelfContained SFPTsccUnitVentilator
SFPTsccVAV SFPTspaceComfortController SFPTsunblindController
SFPTthermostat SFPTunitVentilatorController
SFPTvariableAirVolume*

SNVT_temp_ror

Overview:

Temperature rate of change/rise (degrees Celsius/minute) .

Details:

Standard:	<i>yes</i>
Resource Set:	<i>Standard 00:00:00:00:00:00:00:00-0</i>
Index:	<i>131</i>
Obsolete:	<i>no</i>
Size:	<i>2</i>
Programmatic Name:	<i>SNVT_temp_ror</i>
Neuron C Type:	<i>signed long</i>
Minimum:	<i>-32768</i>
Maximum:	<i>32767</i>

Invalid: 32767
Scaling (A,B,C): 5, -1, 0
Scaled value: $5 * 10^{-1} * (Raw + 0)$
Resolution: 0.5

Formats: *SNVT_temp_ror#SI: text("%f", *1+0(0:1012))*
*SNVT_temp_ror#US: text("%f", *1.8+0(0:1011))*

Used by: *SCPTthermAlrmROR*

SNVT_temp_setpt

Overview:

Temperature (6 temperature values) .

Details:

Standard: *yes*
Resource Set: *Standard 00:00:00:00:00:00:00:00-0*
Index: *106*
Obsolete: *no*
Size: *12*
Programmatic Name: *SNVT_temp_setpt*

Neuron C Type:

Structure	
occupied_cool	Occupied cooling setpoint (degrees Celsius) . signed long Minimum: -27317 Maximum: 32767 Invalid: 32767 Scaling 1, -2, 0 (A,B,C): Scaled value: $1 * 10^{-2} * (Raw + 0)$ Resolution: 0.01
standby_cool	Standby cooling setpoint (degrees Celsius) . signed long Minimum: -27317 Maximum: 32767 Invalid: 32767 Scaling 1, -2, 0 (A,B,C): Scaled value: $1 * 10^{-2} * (Raw + 0)$

	value: Resolution: <i>0.01</i>
unoccupied_cool	Unoccupied cooling setpoint (degrees Celsius) .
	signed long Minimum: <i>-27317</i> Maximum: <i>32767</i> Invalid: <i>32767</i> Scaling <i>1, -2, 0</i> (A,B,C): Scaled <i>1 *10⁻² *(Raw+0)</i> value: Resolution: <i>0.01</i>
occupied_heat	Occupied heating setpoint (degrees Celsius) .
	signed long Minimum: <i>-27317</i> Maximum: <i>32767</i> Invalid: <i>32767</i> Scaling <i>1, -2, 0</i> (A,B,C): Scaled <i>1 *10⁻² *(Raw+0)</i> value: Resolution: <i>0.01</i>
standby_heat	Standby heating setpoint (degrees Celsius) .
	signed long Minimum: <i>-27317</i> Maximum: <i>32767</i> Invalid: <i>32767</i> Scaling <i>1, -2, 0</i> (A,B,C): Scaled <i>1 *10⁻² *(Raw+0)</i> value: Resolution: <i>0.01</i>
unoccupied_heat	Unoccupied heating setpoint (degrees Celsius) .
	signed long Minimum: <i>-27317</i> Maximum: <i>32767</i> Invalid: <i>32767</i> Scaling <i>1, -2, 0</i> (A,B,C): Scaled <i>1 *10⁻² *(Raw+0)</i> value: Resolution: <i>0.01</i>

Formats:

SNVT_temp_setpt#SI: text("%f,%f,%f,%f,%f,%f", occupied_cool, standby_cool, unoccupied_cool, occupied_heat, standby_heat, unoccupied_heat)

SNVT_temp_setpt#SI_LO: text("%f|%f|%f|%f|%f|%f", occupied_cool, standby_cool, unoccupied_cool, occupied_heat, standby_heat, unoccupied_heat)

*SNVT_temp_setpt#US: text("%f,%f,%f,%f,%f,%f", occupied_cool*1.8+32(0:855), standby_cool*1.8+32(0:855), unoccupied_cool*1.8+32(0:855), occupied_heat*1.8+32(0:855), standby_heat*1.8+32(0:855), unoccupied_heat*1.8+32(0:855))*

*SNVT_temp_setpt#US_diff: text("%f,%f,%f,%f,%f,%f", occupied_cool*1.8+0(0:855), standby_cool*1.8+0(0:855), unoccupied_cool*1.8+0(0:855), occupied_heat*1.8+0(0:855), standby_heat*1.8+0(0:855), unoccupied_heat*1.8+0(0:855))*

*SNVT_temp_setpt#US_LO: text("%f|%f|%f|%f|%f|%f", occupied_cool*1.8+32(0:855), standby_cool*1.8+32(0:855), unoccupied_cool*1.8+32(0:855), occupied_heat*1.8+32(0:855), standby_heat*1.8+32(0:855), unoccupied_heat*1.8+32(0:855))*

*SNVT_temp_setpt#US_LO_diff: text("%f|%f|%f|%f|%f|%f", occupied_cool*1.8+0(0:855), standby_cool*1.8+0(0:855), unoccupied_cool*1.8+0(0:855), occupied_heat*1.8+0(0:855), standby_heat*1.8+0(0:855), unoccupied_heat*1.8+0(0:855))*

Used by:

*SCPTsetPnts SFPTchilledCeilingController SFPTsccAHU
SFPTsccChilledCeiling SFPTsccCommandModule SFPTsccFanCoil
SFPTsccHeatPump SFPTsccRadiator SFPTsccRooftop
SFPTsccSelfContained SFPTsccUnitVentilator SFPTsccVAV
SFPTspaceComfortController SFPTunitVentilatorController*

SNVT_therm_mode

Overview:

Thermostat mode (thermostat mode names) .

Details:

Standard:	<i>yes</i>
Resource Set:	<i>Standard 00:00:00:00:00:00:00:00-0</i>
Index:	<i>119</i>
Obsolete:	<i>no</i>
Size:	<i>1</i>
Programmatic Name:	<i>SNVT_therm_mode</i>
Neuron C Type:	<i>therm_mode_t</i>
Formats:	<i>SNVT_therm_mode: text("%m")</i>
Used by:	<i>SCPTthermMode</i>

SNVT_time_f

Overview:

Elapsed time (seconds) .

Details:

Standard:	<i>yes</i>
Resource Set:	<i>Standard 00:00:00:00:00:00:00:00-0</i>
Index:	<i>64</i>
Obsolete:	<i>no</i>
Size:	<i>4</i>
Programmatic Name:	<i>SNVT_time_f</i>
Neuron C Type:	float
	Minimum: <i>0</i>
	Maximum: <i>3.4028234663853E+038</i>
Formats:	<i>SNVT_time_f: text("%f")</i>
Used by:	<i>SFPTgeneratorSet</i>

SNVT_time_hour

Overview:

Elapsed time (hours) .

Details:

Standard:	<i>yes</i>
Resource Set:	<i>Standard 00:00:00:00:00:00:00:00-0</i>
Index:	<i>124</i>
Obsolete:	<i>no</i>
Size:	<i>2</i>
Programmatic Name:	<i>SNVT_time_hour</i>
Neuron C Type:	unsigned long
	Minimum: <i>0</i>
	Maximum: <i>65535</i>
	Scaling (A,B,C): <i>1, 0, 0</i>
	Scaled value: <i>1 * 10⁰ * (Raw + 0)</i>

Resolution: *1*

Formats: *SNVT_time_hour: text("%d")*

Used by: *SCPTblockProtectionTime SFPTpumpController*
SFPTvariableSpeedMotorDrive

SNVT_time_min

Overview:

Elapsed time (minutes) .

Details:

Standard: *yes*

Resource Set: *Standard 00:00:00:00:00:00:00:00-0*

Index: *123*

Obsolete: *no*

Size: *2*

Programmatic Name: *SNVT_time_min*

Neuron C Type: *unsigned long*
 Minimum: *0*
 Maximum: *65535*
 Scaling (A,B,C): *1, 0, 0*
 Scaled value: *1 * 10⁰ * (Raw + 0)*
 Resolution: *1*

Formats: *SNVT_time_min: text("%d")*

Used by: *SCPTbypassTime SCPTdefrostFanDelay SCPTdefrostRecoveryTime*
SCPTmanOvrTime SCPTmaxDefrostTime SCPTminDefrostTime
SCPTsaturationDelay SFPTclothesWasherDomestic
SNVT_clothes_w_c SNVT_clothes_w_s

SNVT_time_passed

Overview:

Elapsed time (hours, minutes, secs, 10-msecs) . This SNVT is obsolete. Use SNVT_elapsed_tm instead.

This SNVT is obsolete. Use SNVT_time_stamp instead.

Details:

Standard: *yes*
Resource Set: *Standard 00:00:00:00:00:00:00:00-0*
Index: *40*
Obsolete: *yes*
Size: *4*
Programmatic Name: *SNVT_time_passed*
Neuron C Type:

Structure	
hours	Hours (hours) .
	unsigned short Minimum: 0 Maximum: 255 Scaling 1, 0, 0 (A,B,C): Scaled $1 * 10^0 * (Raw + 0)$ value: Resolution: 1
minutes	Minutes (minutes) .
	unsigned short Minimum: 0 Maximum: 59 Scaling 1, 0, 0 (A,B,C): Scaled $1 * 10^0 * (Raw + 0)$ value: Resolution: 1
seconds	Seconds (seconds) .
	unsigned short Minimum: 0 Maximum: 59 Scaling 1, 0, 0 (A,B,C): Scaled $1 * 10^0 * (Raw + 0)$ value: Resolution: 1
milliseconds	Milliseconds (milliseconds) .
	unsigned short Minimum: 0 Maximum: 99 Scaling 1, 1, 0 (A,B,C): Scaled $1 * 10^1 * (Raw + 0)$ value: Resolution: 10

Formats: *SNVT_time_passed: text("%d:%d:%d:%d", hours, minutes, seconds, milliseconds)*

SNVT_time_passed#LO: text(time(hours, minutes, seconds, milliseconds))

SNVT_time_sec

Overview:

Elapsed time (seconds) .

Details:

Standard: *yes*

Resource Set: *Standard 00:00:00:00:00:00:00:00-0*

Index: *107*

Obsolete: *no*

Size: *2*

Programmatic Name: *SNVT_time_sec*

Neuron C Type: *unsigned long*

Minimum: *0*

Maximum: *65535*

Invalid: *65535*

Scaling (A,B,C): *1, -1, 0*

Scaled value: *$1 * 10^{-1} * (Raw + 0)$*

Resolution: *0.1*

Formats: *SNVT_time_sec: text("%f")*

Used by: *SCPTclOffDelay SCPTclOnDelay SCPTdebounce SCPTdelayTime
SCPTdrainDelay SCPTdriveTime SCPTfadeTime
SCPThighLimDefrDly SCPThighLimDly SCPTholdTime
SCPTinjDelay SCPTlowLimDly SCPTmaxDefrstTime
SCPTmaxRcvTime SCPTmaxSendTime SCPTminSendTime
SCPTmodeHrtBt SCPTpumpDownDelay SCPTpwrUpDelay
SCPTtrampDownTm SCPTtrampUpTm SCPTstrtupDelay
SCPTtimeout SCPTupdateRate*

SNVT_time_stamp

Overview:

Time stamp (year, month, day, hour, minute, second) .

Details:

Standard: *yes*
Resource Set: *Standard 00:00:00:00:00:00:00:00-0*
Index: *84*
Obsolete: *no*
Size: *7*
Programmatic Name: *SNVT_time_stamp*
Neuron C Type:

Structure	
year	Year (years) . Zero (0) means year not specified. Minus one (-1) represents NULL date.
	signed long Minimum: <i>-1</i> Maximum: <i>3000</i> Invalid: <i>-1</i> Scaling <i>1, 0, 0</i> (A,B,C): Scaled <i>1 * 10⁰ * (Raw + 0)</i> value: Resolution: <i>1</i>
month	Month (months) . Zero (0) means month not specified.
	unsigned short Minimum: <i>0</i> Maximum: <i>12</i> Scaling <i>1, 0, 0</i> (A,B,C): Scaled <i>1 * 10⁰ * (Raw + 0)</i> value: Resolution: <i>1</i>
day	Day (days) . Zero (0) means day not specified.
	unsigned short Minimum: <i>0</i> Maximum: <i>31</i> Scaling <i>1, 0, 0</i> (A,B,C): Scaled <i>1 * 10⁰ * (Raw + 0)</i> value: Resolution: <i>1</i>
hour	Hour (hours) . This field uses a 24-hour value.
	unsigned short Minimum: <i>0</i> Maximum: <i>23</i> Scaling <i>1, 0, 0</i> (A,B,C):

	Scaled value: $1 * 10^0 * (Raw + 0)$ Resolution: 1
minute	Minute (minutes) .
	unsigned short Minimum: 0 Maximum: 59 Scaling (A,B,C): 1, 0, 0 Scaled value: $1 * 10^0 * (Raw + 0)$ Resolution: 1
second	Second (seconds) .
	unsigned short Minimum: 0 Maximum: 59 Scaling (A,B,C): 1, 0, 0 Scaled value: $1 * 10^0 * (Raw + 0)$ Resolution: 1

Formats: *SNVT_time_stamp: text("%d/%d/%d %d:%d:%d", year, month, day, hour, minute, second)*

SNVT_time_stamp#LO: text(date(year, month, day), (" "), time(hour, minute, second))

Used by: *SCPTdefrostStart SCPTinstallDate SCPTmaintDate SCPTmanfDate
 SCPTsummerTime SCPTwinterTime SFPTnodeObject
 SFPTrealTimeKeeper SFPTsccCommandModule
 SFPTutilityDataLoggerRegister SFPTutilityMeter*

SNVT_time_stamp_p

Overview:

Precision timestamp. (seconds) . Timestamp with hundredths of a second resolution.

Details:

Standard: *yes*
 Resource Set: *Standard 00:00:00:00:00:00:00:00-0*
 Index: *192*
 Obsolete: *no*

Size: 5

Programmatic Name: *SNVT_time_stamp_p*

Neuron C Type:

Structure	
second	Time in seconds (seconds) . Time in seconds since 2000-01-01T00:00:00Z (the 0 hour of 1 January 2000, Coordinated Universal Time)
	unsigned quad Minimum: 0 Maximum: 4294967295 Scaling 1, 0, 0 (A,B,C): Scaled $1 * 10^0 * (Raw+0)$ value: Resolution: 1
hundredths	Hundredths of a second. (seconds) . Hundredths portion of a the timestamp.
	unsigned short Minimum: 0 Maximum: 99 Invalid: 255 Scaling 1, -2, 0 (A,B,C): Scaled $1 * 10^{-2} * (Raw+0)$ value: Resolution: 0.01

Used by: *SCPTlogFileHeader SCPTlogRecord SCPTlogRequest*
SNVT_log_fx_request SNVT_log_status

SNVT_time_zone

Overview:

Time zone descriptor (offset, type, startDST, endDST) .

Time Zone with Daylight-Savings Time (DST).

EXAMPLES

Example 1

```
SNVT_time_zone TimeZone;

// Definition of time zone for Prague, Czech Republic, local time = GMT +
TimeZone.second_time_offset
s32_from_ascii("3600", &(TimeZone.second_time_offset));
```

```
// Definition of type of description of time zone
TimeZone.type_of_description = CAL_MEU;

// Definition of time of start and end of DST -> 02:00:00
TimeZone.hour_of_start_DST = 2;
TimeZone.minute_of_start_DST = 0;
TimeZone.second_of_start_DST = 0;
TimeZone.hour_of_end_DST = 2;
TimeZone.minute_of_end_DST = 0;
TimeZone.second_of_end_DST = 0;

// Definition of date of start DST - Sunday of 4 week in March
TimeZone.start_DST.M_start_DST.month_of_start_DST = 3;
TimeZone.start_DST.M_start_DST.week_of_start_DST = 4;
TimeZone.start_DST.M_start_DST.dateday_of_start_DST = 0;

// Definition of date of end DST - Sunday of 4 week in September
TimeZone.end_DST.M_end_DST.month_of_end_DST = 9;
TimeZone.end_DST.M_end_DST.week_of_end_DST = 4;
TimeZone.end_DST.M_end_DST.dateday_of_end_DST = 0;
```

Example 2

```
// Definition of time zone for country X, that has start DST in every year 30. Mar
end of DST 1. September

s32_from_ascii("-36000", &(TimeZone.second_time_offset));

// Definition of type of description of time zone
TimeZone.type_of_description = CAL_JUL ;

// Definition of time of start and end of DST -> 02:00:00
TimeZone.hour_of_start_DST = 2;
TimeZone.minute_of_start_DST = 0;
TimeZone.second_of_start_DST = 0;
TimeZone.hour_of_end_DST = 2;
TimeZone.minute_of_end_DST = 0;
TimeZone.second_of_end_DST = 0;

// Definition of day of start DST - 30. March
TimeZone.start_DST.J_day_of_start_DST = 89;

// Definition of day of end DST - 1. September
TimeZone.end_DST.J_day_of_end_DST = 244;
```

Example 3

```
// Definition of time zone for country Y, that has start DST in the not leap year
February and end of DST 1. September.

// In the leap year has start DST 20. February and end of DST 31. August
s32_from_ascii("-56000", &(TimeZone.second_time_offset));

// Definition of type of description of time zone
TimeZone.type_of_description = CAL_GREG ;

// Definition of time of start and end of DST -> 02:00:00
TimeZone.hour_of_start_DST = 2;
TimeZone.minute_of_start_DST = 0;
TimeZone.second_of_start_DST = 0;
TimeZone.hour_of_end_DST = 2;
TimeZone.minute_of_end_DST = 0;
TimeZone.second_of_end_DST = 0;

// Definition of day of start DST - 20. February
```

```
TimeZone.start_DST.G_day_of_start_DST = 50;
```

```
// Definition of day of end DST - 1. September - not leap year, leap year 31. Augu
TimeZone.end_DST.G_day_of_end_DST = 243;
```

Details:

Standard: *yes*

Resource Set: *Standard 00:00:00:00:00:00:00:00-0*

Index: *134*

Obsolete: *no*

Size: *15*

Programmatic *SNVT_time_zone*

Name:

Neuron C

Type:

Structure	
second_time_offset	Offset from GMT (seconds) . West direction is negative offset
	s32_type Minimum: <i>-86400</i> Maximum: <i>86400</i> Scaling (A,B,C): <i>1, 0, 0</i> Scaled value: <i>1 * 10⁰ *(Raw+0)</i> Resolution: <i>1</i>
type_of_description	Calendar type (calendar type names) .
	<i>calendar_type_t</i>
hour_of_start_DST	DST start hour (hours) .
	unsigned short Minimum: <i>0</i> Maximum: <i>23</i> Scaling (A,B,C): <i>1, 0, 0</i> Scaled value: <i>1 * 10⁰ *(Raw+0)</i> Resolution: <i>1</i>
minute_of_start_DST	DST start minute (minutes) .
	unsigned short Minimum: <i>0</i> Maximum: <i>59</i> Scaling (A,B,C): <i>1, 0, 0</i> Scaled value: <i>1 * 10⁰ *(Raw+0)</i> Resolution: <i>1</i>
second_of_start_DST	DST start second (seconds) .
	unsigned short Minimum: <i>0</i> Maximum: <i>59</i> Scaling (A,B,C): <i>1, 0, 0</i> Scaled value: <i>1 * 10⁰ *(Raw+0)</i>

	Resolution: 1								
start_DST	DST start day (day descriptor) . Daylight savings time start day								
	Union								
G_day_of_start_DST	Gregorian calendar day of start DST (days)								
	unsigned long Minimum: 0 Maximum: 365 Scaling 1, 0, 0 (A,B,C): Scaled $1 * 10^0 * (Raw + 0)$ value: Resolution: 1								
J_day_of_start_DST	Julian calendar day of start DST (days)								
	unsigned long Minimum: 1 Maximum: 365 Scaling 1, 0, 0 (A,B,C): Scaled $1 * 10^0 * (Raw + 0)$ value: Resolution: 1								
M_start_DST	Meu calendar day of start DST (month, dateday) .								
	Structure <table> <tr> <td>month_of_start_DST</td><td>Month of start DST (months) .</td></tr> <tr> <td></td><td> bitfield Signed: no Width: 4 Offset: 0 Minimum: 1 Maximum: 12 Scaling 1, 0 (A,B,C): Scaled $1 * 10^0 * (Raw + 0)$ value: Resolution: 1 </td></tr> <tr> <td>week_of_start_DST</td><td>Week of start DST (weeks) .</td></tr> <tr> <td></td><td> bitfield Signed: no Width: 3 Offset: 4 </td></tr> </table>	month_of_start_DST	Month of start DST (months) .		bitfield Signed: no Width: 4 Offset: 0 Minimum: 1 Maximum: 12 Scaling 1, 0 (A,B,C): Scaled $1 * 10^0 * (Raw + 0)$ value: Resolution: 1	week_of_start_DST	Week of start DST (weeks) .		bitfield Signed: no Width: 3 Offset: 4
month_of_start_DST	Month of start DST (months) .								
	bitfield Signed: no Width: 4 Offset: 0 Minimum: 1 Maximum: 12 Scaling 1, 0 (A,B,C): Scaled $1 * 10^0 * (Raw + 0)$ value: Resolution: 1								
week_of_start_DST	Week of start DST (weeks) .								
	bitfield Signed: no Width: 3 Offset: 4								

			Minimum: 1 Maximum: 5 Scaling (A,B,C): 1, 0, 0 Scaled value: $1 * 10^0 * (Raw + 0)$ Resolution: 1
		day_of_start_DST	Day of week (day names) .
			days_of_week_index
hour_of_end_DST	DST end hour (hours) .		
	unsigned short Minimum: 0 Maximum: 23 Scaling (A,B,C): 1, 0, 0 Scaled value: $1 * 10^0 * (Raw + 0)$ Resolution: 1		
minute_of_end_DST	DST end minute (minutes) .		
	unsigned short Minimum: 0 Maximum: 59 Scaling (A,B,C): 1, 0, 0 Scaled value: $1 * 10^0 * (Raw + 0)$ Resolution: 1		
second_of_end_DST	DST end second (seconds) .		
	unsigned short Minimum: 0 Maximum: 59 Scaling (A,B,C): 1, 0, 0 Scaled value: $1 * 10^0 * (Raw + 0)$ Resolution: 1		
end_DST	DST end day (day descriptor) . Daylight savings time end day		
	Union		
	G_day_of_end_DST	Gregorian calendar day of end DST (day of year)	
		unsigned long Minimum: 0 Maximum: 365 Scaling (A,B,C): 1, 0, 0 Scaled value: $1 * 10^0 * (Raw + 0)$ Resolution: 1	

	J_day_of_end_DST	Julian calendar day of end DST (days) .	
		unsigned long Minimum: 1 Maximum: 365 Scaling 1, 0, 0 (A,B,C): Scaled $1 * 10^0 * (Raw + 0)$ value: Resolution: 1	
	M_end_DST	Meu calendar day of end DST (month, weekday) .	
		Structure	
		month_of_end_DST	Month of end DST (months) .
			bitfield Signed: no Width: 4 Offset: 0 Minimum: 1 Maximum: 12 Scaling 1, 0, (A,B,C): Scaled $1 * 10^0 * (Raw + 0)$ value: Resolution: 1
		week_of_end_DST	Week of end DST (weeks) .
			bitfield Signed: no Width: 3 Offset: 4 Minimum: 1 Maximum: 5 Scaling 1, 0, (A,B,C): Scaled $1 * 10^0 * (Raw + 0)$ value: Resolution: 1
		dateday_of_end_DST	Day of week (day names) .
			days_of_week_t

Formats:

SNVT_time_zone: text(("%d %m %d:%d:%d ",second_time_offset,
 type_of_description,hour_of_start_DST, minute_of_start_DST,second_of_start_DST),
 ((type_of_description SNVT_time_zone#LO: text(("%d %m ", second_time_offset,
 type_of_description),time(hour_of_start_DST, minute_of_start_DST, second_of_start_

((type_of_description SNVT_tod_event: text("%m,%m,%f", current_state, next_state, time_to_next_state)

Used by: SCPTtimeZone

SNVT_tod_event

Overview:

Time of day event (current, next, time) . Occupancy scheduling event

Details:

Standard: yes

Resource Set: Standard 00:00:00:00:00:00:00:00-0

Index: 128

Obsolete: no

Size: 4

Programmatic Name: SNVT_tod_event

Neuron C Type:

Structure	
current_state	Occupancy, current (occupancy code names) .
	occup_t
next_state	Occupancy, next (occupancy code names) .
	occup_t
time_to_next_state	Time to next state (minutes) .
	unsigned long Minimum: 0 Maximum: 65535 Scaling 1, 0, 0 (A,B,C): Scaled $1 * 10^0 * (Raw + 0)$ value: Resolution: 1

Formats: SNVT_tod_event#LO: text("%m|%m|%f", current_state, next_state, time_to_next_state)

Used by: SFPTchilledCeilingController SFPTdischargeAirController
 SFPTsccAHU SFPTsccChilledCeiling SFPTsccFanCoil
 SFPTsccHeatPump SFPTsccRadiator SFPTsccRooftop
 SFPTsccSelfContained SFPTsccUnitVentilator SFPTsccVAV
 SFPTspaceComfortController SFPTunitVentilatorController

SNVT_trans_table

Overview:

Translation table (points, interpolation) .

A translation table is defined by two of sets network variables of type SNVT_trans_table; one for the X axis, and one for the Y axis.

The point seven-element array contains a single axis of translation values, represented as IEEE754 single-precision floating-point values (compatible with the Neuron C Extended Arithmetic float_type type). The point values in the network variable for the X axis must be monotonically increasing. The interp_pts_m_to_n fields specify the type of interpolation to be used between the indicated pair of point elements. The values in these fields are defined with the interp_t enumeration type, and may be IP_LINEAR or IP_CUBIC_SPLINE.

If more than one pair of network variables of type SNVT_trans_table are present in a functional block, the interp_pts_6_to_0 field specifies the type of interpolation to be used between point 6 of this table, and point 0 of the subsequent table (in order of X point value).

A SNVT_IP.H header file is available for Neuron C applications. There are no direct enumeration references to this file by any SNVT, including SNVT_trans_table. The interp_pts_m_to_n fields are bitfields, and therefore do not reference interp_t, though their interpretation is the same as of the values defined by interp_t.

Details:

Standard:	<i>yes</i>
Resource Set:	<i>Standard 00:00:00:00:00:00:00:00-0</i>
Index:	<i>96</i>
Obsolete:	<i>no</i>
Size:	<i>30</i>
Programmatic Name:	<i>SNVT_trans_table</i>

Neuron C Type:

Structure	
point	Points (array of 7 points) .
	float [7] Minimum: <i>-3.4028234663853E+038</i> Maximum: <i>3.4028234663853E+038</i>
interp_pts_0_to_1	Interpolation for point 0 to point 1 (interpolation method code) .
	bitfield Signed: <i>no</i> Width: <i>2</i> Offset: <i>0</i> Minimum: <i>0</i> Maximum: <i>1</i> Scaling <i>1, 0, 0</i>

	<p>(A,B,C):</p> <p>Scaled value: $1 * 10^0 * (Raw + 0)$</p> <p>Resolution: 1</p>
interp_pts_1_to_2	Interpolation for point 1 to point 2 (interpolation method code) .
	<p>bitfield</p> <p>Signed: no</p> <p>Width: 2</p> <p>Offset: 2</p> <p>Minimum: 0</p> <p>Maximum: 1</p> <p>Scaling 1, 0, 0</p> <p>(A,B,C):</p> <p>Scaled value: $1 * 10^0 * (Raw + 0)$</p> <p>Resolution: 1</p>
interp_pts_2_to_3	Interpolation for point 2 to point 3 (interpolation method code) .
	<p>bitfield</p> <p>Signed: no</p> <p>Width: 2</p> <p>Offset: 4</p> <p>Minimum: 0</p> <p>Maximum: 1</p> <p>Scaling 1, 0, 0</p> <p>(A,B,C):</p> <p>Scaled value: $1 * 10^0 * (Raw + 0)$</p> <p>Resolution: 1</p>
interp_pts_3_to_4	Interpolation for point 3 to point 4 (interpolation method code) .
	<p>bitfield</p> <p>Signed: no</p> <p>Width: 2</p> <p>Offset: 6</p> <p>Minimum: 0</p> <p>Maximum: 1</p> <p>Scaling 1, 0, 0</p> <p>(A,B,C):</p> <p>Scaled value: $1 * 10^0 * (Raw + 0)$</p> <p>Resolution: 1</p>
interp_pts_4_to_5	Interpolation for point 4 to point 5 (interpolation method code) .
	<p>bitfield</p>

	Signed: <i>no</i> Width: <i>2</i> Offset: <i>0</i> Minimum: <i>0</i> Maximum: <i>1</i> Scaling (A,B,C): <i>1, 0, 0</i> Scaled value: $1 * 10^0 * (Raw + 0)$ Resolution: <i>1</i>
interp_pts_5_to_6	Interpolation for point 5 to point 6 (interpolation method code) .
	bitfield Signed: <i>no</i> Width: <i>2</i> Offset: <i>2</i> Minimum: <i>0</i> Maximum: <i>1</i> Scaling (A,B,C): <i>1, 0, 0</i> Scaled value: $1 * 10^0 * (Raw + 0)$ Resolution: <i>1</i>
interp_pts_6_to_0	Interpolation for point 6 to point 0 (interpolation method code) . This field is used when multiple interpolation tables are linked.
	bitfield Signed: <i>no</i> Width: <i>2</i> Offset: <i>4</i> Minimum: <i>0</i> Maximum: <i>1</i> Scaling (A,B,C): <i>1, 0, 0</i> Scaled value: $1 * 10^0 * (Raw + 0)$ Resolution: <i>1</i>

Formats:

SNVT_trans_table: text("%f%f%f%f%f%f%f%f%d %d %d %d %d %d %d", point[0], point[1], point[2], point[3], point[4], point[5], point[6], interp_pts_0_to_1, interp_pts_1_to_2, interp_pts_2_to_3, interp_pts_3_to_4, interp_pts_4_to_5, interp_pts_5_to_6, interp_pts_6_to_0)

Used by:

*SCPTtrnsTblX SCPTtrnsTblX2 SCPTtrnsTblX3 SCPTtrnsTblY
SCPTtrnsTblY2 SCPTtrnsTblY3*

SNVT_turbidity

Overview:

Turbidity (nephelometric turbidity units) .

Used to communicate with water-clarity measuring devices (turbidimeters). SNVT_turbidity can be used to represent ranges associated with drinking water. Units of this variable are NTU (Nephelometric Turbidity Units), which are 1-to-1 equivalent to FNU (Formazin Nephelometric Units), and FTU (Formazin Turbidity Units).

Details:

Standard:	<i>yes</i>
Resource Set:	<i>Standard 00:00:00:00:00:00:00:00-0</i>
Index:	<i>143</i>
Obsolete:	<i>no</i>
Size:	<i>2</i>
Programmatic Name:	<i>SNVT_turbidity</i>
Neuron C Type:	unsigned long
Minimum:	<i>0</i>
Maximum:	<i>65535</i>
Invalid:	<i>65535</i>
Scaling (A,B,C):	<i>1, -3, 0</i>
Scaled value:	<i>$1 * 10^{-3} * (Raw + 0)$</i>
Resolution:	<i>0.001</i>
Formats:	<i>SNVT_turbidity: text("%f")</i>

SNVT_turbidity_f

Overview:

Turbidity (nephelometric turbidity units) .

Used to communicate with water-clarity measuring devices (turbidimeters). SNVT_turbidity_f can be used to represent ranges associated with industrial applications (e.g., typical measurements may be from 0 to 10000 NTU). Units of this variable are NTU (Nephelometric Turbidity Units), which are 1-to-1 equivalent to FNU (Formazin Nephelometric Units), and FTU (Formazin Turbidity Units).

Details:

Standard:	<i>yes</i>
-----------	------------

Resource Set:	<i>Standard 00:00:00:00:00:00:00:00-0</i>
Index:	<i>144</i>
Obsolete:	<i>no</i>
Size:	<i>4</i>
Programmatic Name:	<i>SNVT_turbidity_f</i>
Neuron C Type:	float
	Minimum: <i>0</i>
	Maximum: <i>3.4028234663853E+038</i>
Formats:	<i>SNVT_turbidity_f: text("%f")</i>

SNVT_valve_mode

Overview:

Valve mode (valve mode names) .

Used for heating, ventilation, and air-conditioning applications. This network variable type defines and indicates the working mode of valves.

Details:

Standard:	<i>yes</i>
Resource Set:	<i>Standard 00:00:00:00:00:00:00:00-0</i>
Index:	<i>163</i>
Obsolete:	<i>no</i>
Size:	<i>1</i>
Programmatic Name:	<i>SNVT_valve_mode</i>
Neuron C Type:	<i>valve_mode_t</i>
Formats:	<i>SNVT_valve_mode: text("%m")</i>
Used by:	<i>SCPTvalveOperatingMode SFPTHvacValvePositioner</i>

SNVT_vol

Overview:

Volume (liters) .

Details:

Standard: *yes*
 Resource Set: *Standard 00:00:00:00:00:00:00:00-0*
 Index: *41*
 Obsolete: *no*
 Size: *2*
 Programmatic Name: *SNVT_vol*
 Neuron C Type: *unsigned long*
 Minimum: *0*
 Maximum: *65535*
 Scaling (A,B,C): *1, -1, 0*
 Scaled value: *1 * 10⁻¹ * (Raw+0)*
 Resolution: *0.1*
 Formats: *SNVT_vol#SI: text("%f")*
 *SNVT_vol#US: text("%f", *0.035315+0(0:877))*
 *SNVT_vol#US_gal: text("%f", *0.26418+0(0:858))*

SNVT_vol_f

Overview:

Volume (liters) .

Details:

Standard: *yes*
 Resource Set: *Standard 00:00:00:00:00:00:00:00-0*
 Index: *65*
 Obsolete: *no*
 Size: *4*
 Programmatic Name: *SNVT_vol_f*
 Neuron C Type: *float*
 Minimum: *0*
 Maximum: *3.4028234663853E+038*
 Formats: *SNVT_vol_f#SI: text("%f")*
 *SNVT_vol_f#US: text("%f", *0.035315+0(0:877))*
 *SNVT_vol_f#US_gal: text("%f", *0.26418+0(0:858))*

SNVT_vol_kilo

Overview:

Volume (kiloliters) .

Details:

Standard:	<i>yes</i>
Resource Set:	<i>Standard 00:00:00:00:00:00:00:00-0</i>
Index:	<i>42</i>
Obsolete:	<i>no</i>
Size:	<i>2</i>
Programmatic Name:	<i>SNVT_vol_kilo</i>
Neuron C Type:	unsigned long
Minimum:	<i>0</i>
Maximum:	<i>65535</i>
Scaling (A,B,C):	<i>1, -1, 0</i>
Scaled value:	<i>$1 * 10^{-1} * (Raw + 0)$</i>
Resolution:	<i>0.1</i>
Formats:	<i>SNVT_vol_kilo#SI: text("%f")</i>
	<i>SNVT_vol_kilo#US: text("%f", *35.315+0(0:877))</i>
	<i>SNVT_vol_kilo#US_gal: text("%f", *264.179+0(0:858))</i>

SNVT_vol_mil

Overview:

Volume (milliliters) .

Details:

Standard:	<i>yes</i>
Resource Set:	<i>Standard 00:00:00:00:00:00:00:00-0</i>
Index:	<i>43</i>
Obsolete:	<i>no</i>
Size:	<i>2</i>
Programmatic Name:	<i>SNVT_vol_mil</i>
Neuron C Type:	unsigned long

Minimum: 0
 Maximum: 65535
 Scaling (A,B,C): 1, -1, 0
 Scaled value: $1 * 10^{-1} * (Raw + 0)$
 Resolution: 0.1

Formats: *SNVT_vol_mil#SI: text("%f")*
*SNVT_vol_mil#US: text("%f", *0.0338+0(0:861))*

SNVT_volt

Overview:

Electric voltage (Volts) .

Details:

Standard: *yes*
 Resource Set: *Standard 00:00:00:00:00:00:00:00-0*
 Index: *44*
 Obsolete: *no*
 Size: *2*
 Programmatic Name: *SNVT_volt*
 Neuron C Type: *signed long*
 Minimum: *-32768*
 Maximum: *32767*
 Scaling (A,B,C): *1, -1, 0*
 Scaled value: $1 * 10^{-1} * (Raw + 0)$
 Resolution: *0.1*
 Formats: *SNVT_volt: text("%f")*
 Used by: *SFPTgeneratorSet SFPTvariableSpeedMotorDrive SNVT_pumpset_sn*

SNVT_volt_ac

Overview:

Voltage in alternating current (volts AC) .

Used to represent voltages (absolute or differential) that need a large range, rather than a fine

resolution.

Details:

Standard:	<i>yes</i>
Resource Set:	<i>Standard 00:00:00:00:00:00:00:00-0</i>
Index:	<i>138</i>
Obsolete:	<i>no</i>
Size:	<i>2</i>
Programmatic Name:	<i>SNVT_volt_ac</i>
Neuron C Type:	unsigned long
	Minimum: <i>0</i>
	Maximum: <i>65535</i>
	Invalid: <i>65535</i>
	Scaling (A,B,C): <i>1, 0, 0</i>
	Scaled value: <i>$1 * 10^0 * (Raw + 0)$</i>
	Resolution: <i>1</i>
Formats:	<i>SNVT_volt_ac: text("%f")</i>
Used by:	<i>SFPTautomaticTransferSwitch SFPTgeneratorSet</i>

SNVT_volt_dbmv

Overview:

Electric voltage (dB microVolts) .

Details:

Standard:	<i>yes</i>
Resource Set:	<i>Standard 00:00:00:00:00:00:00:00-0</i>
Index:	<i>45</i>
Obsolete:	<i>no</i>
Size:	<i>2</i>
Programmatic Name:	<i>SNVT_volt_dbmv</i>
Neuron C Type:	signed long
	Minimum: <i>-32768</i>
	Maximum: <i>32767</i>
	Scaling (A,B,C): <i>1, -2, 0</i>
	Scaled value: <i>$1 * 10^{-2} * (Raw + 0)$</i>
	Resolution: <i>0.01</i>
Formats:	<i>SNVT_volt_dbmv: text("%f")</i>

SNVT_volt_f

Overview:

Electric voltage (Volts) .

Details:

Standard:	<i>yes</i>
Resource Set:	<i>Standard 00:00:00:00:00:00:00:00-0</i>
Index:	<i>66</i>
Obsolete:	<i>no</i>
Size:	<i>4</i>
Programmatic Name:	<i>SNVT_volt_f</i>
Neuron C Type:	float
	Minimum: <i>-3.4028234663853E+038</i>
	Maximum: <i>3.4028234663853E+038</i>
Formats:	<i>SNVT_volt_f: text("%f")</i>

SNVT_volt_kilo

Overview:

Electric voltage (kiloVolts) .

Details:

Standard:	<i>yes</i>
Resource Set:	<i>Standard 00:00:00:00:00:00:00:00-0</i>
Index:	<i>46</i>
Obsolete:	<i>no</i>
Size:	<i>2</i>
Programmatic Name:	<i>SNVT_volt_kilo</i>
Neuron C Type:	signed long
	Minimum: <i>-32768</i>
	Maximum: <i>32767</i>
	Scaling (A,B,C): <i>1, -1, 0</i>
	Scaled value: <i>1 *10⁻¹ *(Raw+0)</i>

Resolution: *0.1*

Formats: *SNVT_volt_kilo: text("%f")*

SNVT_volt_mil

Overview:

Electric voltage (milliVolts) .

Details:

Standard: *yes*

Resource Set: *Standard 00:00:00:00:00:00:00:00-0*

Index: *47*

Obsolete: *no*

Size: *2*

Programmatic Name: *SNVT_volt_mil*

Neuron C Type: *signed long*

Minimum: *-32768*

Maximum: *32767*

Scaling (A,B,C): *1, -1, 0*

Scaled value: *$1 * 10^{-1} * (Raw + 0)$*

Resolution: *0.1*

Formats: *SNVT_volt_mil: text("%f")*

SNVT_zerospan

Overview:

Zero and span (Zero, span) . Linear transformation parameters: multiply by the span-factor, then add the zero-term.

Used to represent a linear transformation on fixed-point data. Linear transformation parameters: multiply by the span-factor, then add the zero-term.

Details:

Standard: *yes*

Resource Set: *Standard 00:00:00:00:00:00:00:00-0*

Index: *85*

Obsolete: *no*

Size: *4*

Programmatic Name: *SNVT_zerospan*

Neuron C Type:

Structure	
zero	Zero-term (16-bit signed value) .
	signed long Minimum: -32768 Maximum: 32767 Scaling 5, -3, 0 (A,B,C): Scaled $5 * 10^{-3} * (Raw+0)$ value: Resolution: 0.005
span	Span-factor (16-bit unsigned value) .
	unsigned long Minimum: 0 Maximum: 65535 Scaling 5, -4, 0 (A,B,C): Scaled $5 * 10^{-4} * (Raw+0)$ value: Resolution: 0.0005

Formats: *SNVT_zerospan: text("%f,%f", zero, span)*

SNVT_zerospan#LO: text("%f|%f", zero, span)