Standard Network Variable Types

Network Variables Standard 00:00:00:00:00:00:00:00-0. 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
SNVT_date_time
SNVT_defr_mode
SNVT_defr_state
SNVT_defr_term
SNVT_density
SNVT_density_f
SNVT_dev_c_mode
SNVT_dev_fault
SNVT_dev_maint
SNVT_dev_status
SNVT_earth_pos
SNVT_elapsed_tm
SNVT_elec_kwh
SNVT_elec_kwh_l
SNVT_elec_whr
SNVT_elec_whr_f
SNVT_ent_opmode
SNVT_ent_state
SNVT_ent_status
SNVT_enthalpy
SNVT_evap_state
SNVT_ex_control
SNVT_file_pos
SNVT_file_req
SNVT_file_status
SNVT_fire_indcte
SNVT_fire_init
SNVT_fire_test
SNVT_flow
SNVT_flow_dir
SNVT_flow_f
SNVT_flow_mil
SNVT_flow_p
SNVT_freq_f
SNVT_freq_hz
SNVT_freq_kilohz
SNVT_freq_milhz
SNVT_gfci_status
SNVT_grammage
SNVT_grammage_f
SNVT_hvac_emerg
SNVT_hvac_mode
SNVT_hvac_overid
SNVT_hvac_satsts
SNVT_hvac_status
SNVT_hvac_type
SNVT_ISO_7811
SNVT_length
SNVT_length_f
SNVT_length_kilo
SNVT_length_micr
SNVT_length_mil
SNVT_lev_cont
SNVT_lev_cont_f
SNVT_lev_disc
SNVT_lev_percent

37 SNVT_str_int
38 SNVT_telcom
39 SNVT_temp
40 SNVT_time_passed
41 SNVT_vol
42 SNVT_vol_kilo
43 SNVT_vol_mil
44 SNVT_volt
45 SNVT_volt_dbmv
46 SNVT_volt_kilo
47 SNVT_volt_mil
48 SNVT_amp_f
49 SNVT_angle_f
50 SNVT_angle_vel_f
51 SNVT_count_f
52 SNVT_count_inc_f
53 SNVT_flow_f
54 SNVT_length_f
55 SNVT_lev_cont_f
56 SNVT_mass_f
57 SNVT_power_f
58 SNVT_ppm_f
59 SNVT_press_f
60 SNVT_res_f
61 SNVT_sound_db_f
62 SNVT_speed_f
63 SNVT_temp_f
64 SNVT_time_f
65 SNVT_vol_f
66 SNVT_volt_f
67 SNVT_btu_f
68 SNVT_elec_whr_f
69 SNVT_config_src
70 SNVT_color
71 SNVT_grammage
72 SNVT_grammage_f
73 SNVT_file_req
74 SNVT_file_status
75 SNVT_freq_f
76 SNVT_freq_hz
77 SNVT_freq_kilohz
78 SNVT_freq_milhz
79 SNVT_lux
80 SNVT_ISO_7811
81 SNVT_lev_percent
82 SNVT_multiplier
83 SNVT_state
84 SNVT_time_stamp
85 SNVT_zerospan
86 SNVT_magcard
87 SNVT_elapsed_tm
88 SNVT_alarm
89 SNVT_currency
90 SNVT_file_pos
91 SNVT_muldiv
92 SNVT_obj_request
93 SNVT_obj_status

SNVT_log_fx_request
SNVT_log_fx_status
SNVT_log_request
SNVT_log_status
SNVT_lux
SNVT_magcard
SNVT_mass
SNVT_mass_f
SNVT_mass_kilo
SNVT_mass_mega
SNVT_mass_mil
SNVT_motor_state
SNVT_muldiv
SNVT_multiplier
SNVT_multiplier_s
SNVT_nv_type
SNVT_obj_request
SNVT_obj_status
SNVT_occupancy
SNVT_override
SNVT_ph
SNVT_ph_f
SNVT_pos_ctrl
SNVT_power
SNVT_power_f
SNVT_power_kilo
SNVT_ppm
SNVT_ppm_f
SNVT_preset
SNVT_press
SNVT_press_f
SNVT_press_p
SNVT_privacyzone
SNVT_ptz
SNVT_pump_sensor
SNVT_pumpset_mn
SNVT_pumpset_sn
SNVT_pwr_fact
SNVT_pwr_fact_f
SNVT_rac_ctrl
SNVT_rac_req
SNVT_reg_val
SNVT_reg_val_ts
SNVT_res
SNVT_res_f
SNVT_res_kilo
SNVT_rpm
SNVT_sblnd_state
SNVT_scene
SNVT_scene_cfg
SNVT_sched_val
SNVT_sec_state
SNVT_sec_status
SNVT_setting
SNVT_smo_obscur
SNVT_sound_db
SNVT_sound_db_f

94	SNVT_preset
95	SNVT_switch
96	SNVT_trans_table
97	SNVT_override
98	SNVT_pwr_fact
99	SNVT_pwr_fact_f
100	SNVT_density
101	SNVT_density_f
102	SNVT_rpm
103	SNVT_hvac_emerg
104	SNVT_angle_deg
105	SNVT_temp_p
106	SNVT_temp_setpt
107	SNVT_time_sec
108	SNVT_hvac_mode
109	SNVT_occupancy
110	SNVT_area
111	SNVT_hvac_overid
112	SNVT_hvac_status
113	SNVT_press_p
114	SNVT_address
115	SNVT_scene
116	SNVT_scene_cfg
117	SNVT_setting
118	SNVT_evap_state
119	SNVT_therm_mode
120	SNVT_defr_mode
121	SNVT_defr_term
122	SNVT_defr_state
123	SNVT_time_min
124	SNVT_time_hour
125	SNVT_ph
126	SNVT_ph_f
127	SNVT_chlr_status
128	SNVT_tod_event
129	SNVT_smo_obscur
130	SNVT_fire_test
131	SNVT_temp_ror
132	SNVT_fire_init
133	SNVT_fire_indcte
134	SNVT_time_zone
135	SNVT_earth_pos
136	SNVT_reg_val
137	SNVT_reg_val_ts
138	SNVT_volt_ac
139	SNVT_amp_ac
143	SNVT_turbidity
144	SNVT_turbidity_f
145	SNVT_hvac_type
146	SNVT_elec_kwh_l
147	SNVT_temp_diff_p
148	SNVT_ctrl_req
149	SNVT_ctrl_resp
150	SNVT_ptz
151	SNVT_privacyzone
152	SNVT_pos_ctrl
153	SNVT_enthalpy

SNVT_speed
SNVT_speed_f
SNVT_speed_mil
SNVT_state
SNVT_state_64
SNVT_str_asc
SNVT_str_int
SNVT_switch
SNVT_switch_2
SNVT_telcom
SNVT_temp
SNVT_temp_diff_p
SNVT_temp_f
SNVT_temp_p
SNVT_temp_ror
SNVT_temp_setpt
SNVT_therm_mode
SNVT_time_f
SNVT_time_hour
SNVT_time_min
SNVT_time_passed
SNVT_time_sec
SNVT_time_stamp
SNVT_time_stamp_p
SNVT_time_zone
SNVT_tod_event
SNVT_trans_table
SNVT_turbidity
SNVT_turbidity_f
SNVT_valve_mode
SNVT_vol
SNVT_vol_f
SNVT_vol_kilo
SNVT_vol_mil
SNVT_volt
SNVT_volt_ac
SNVT_volt_dbmv
SNVT_volt_f
SNVT_volt_kilo
SNVT_volt_mil
SNVT_zerospan

154 SNVT_gfci_status
155 SNVT_motor_state
156 SNVT_pumpset_mn
157 SNVT_ex_control
158 SNVT_pumpset_sn
159 SNVT_pump_sensor
160 SNVT_abs_humid
161 SNVT_flow_p
162 SNVT_dev_c_mode
163 SNVT_valve_mode
164 SNVT_alarm_2
165 SNVT_state_64
166 SNVT_nv_type
168 SNVT_ent_opmode
169 SNVT_ent_state
170 SNVT_ent_status
171 SNVT_flow_dir
172 SNVT_hvac_satsts
173 SNVT_dev_status
174 SNVT_dev_fault
175 SNVT_dev_maint
176 SNVT_date_event
177 SNVT_sched_val
178 SNVT_sec_state
179 SNVT_sec_status
180 SNVT_sblnd_state
181 SNVT_rac_ctrl
182 SNVT_rac_req
183 SNVT_count_32
184 SNVT_clothes_w_c
185 SNVT_clothes_w_m
186 SNVT_clothes_w_s
187 SNVT_clothes_w_a
188 SNVT_multiplier_s
189 SNVT_switch_2
190 SNVT_color_2
191 SNVT_log_status
192 SNVT_time_stamp_p
193 SNVT_log_fx_request
194 SNVT_log_fx_status
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: *no* Width: *4* Offset: *0* Minimum: *0* Maximum: *15* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} *(Raw+0)* Resolution: *1*
digit2	Digit 2 (hexadecimal digit) .
	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*
digit3	Digit 3 (hexadecimal digit) .

	bitfield Signed: *no* Width: *4* Offset: *0* Minimum: *0* Maximum: *15* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} *(Raw+0)* Resolution: *1*
digit4	Digit 4 (hexadecimal digit) .
	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*
digit5	Digit 5 (hexadecimal digit) .
	bitfield Signed: *no* Width: *4* Offset: *0* Minimum: *0* Maximum: *15* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} *(Raw+0)* Resolution: *1*
digit6	Digit 6 (hexadecimal digit) .
	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*
digit7	Digit 7 (hexadecimal digit) .
	bitfield Signed: *no* Width: *4* Offset: *0* Minimum: *0* Maximum: *15* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} *(Raw+0)* Resolution: *1*
digit8	Digit 8 (hexadecimal digit) .
	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*
digit9	Digit 9 (hexadecimal digit) .
	bitfield Signed: *no* Width: *4* Offset: *0* Minimum: *0* Maximum: *15* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} *(Raw+0)* Resolution: *1*
digit10	Digit 10 (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*
digit11	Digit 11 (hexadecimal digit) .
	bitfield Signed: *no* Width: *4* Offset: *0* Minimum: *0* Maximum: *15* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} *(Raw+0)* Resolution: *1*
digit12	Digit 12 (hexadecimal digit) .
	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*
digit13	Digit 13 (hexadecimal digit) .
	bitfield Signed: *no* Width: *4* Offset: *0* Minimum: *0* Maximum: *15* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} *(Raw+0)* Resolution: *1*
digit14	Digit 14 (hexadecimal digit) .
	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*
digit15	Digit 15 (hexadecimal digit) .
	bitfield Signed: *no* Width: *4* Offset: *0* Minimum: *0* Maximum: *15* Scaling (A,B,C): *1, 0, 0* 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 (A,B,C): *1, 0, 0* 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 (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} *(Raw+0)* Resolution: *1*
digit18	Digit 18 (hexadecimal digit) .
	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*
digit19	Digit 19 (hexadecimal digit) .
	bitfield Signed: *no* Width: *4* Offset: *0* Minimum: *0* Maximum: *15* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} *(Raw+0)* Resolution: *1*
digit20	Digit 20 (hexadecimal digit) .
	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*
digit21	Digit 21 (hexadecimal digit) .
	bitfield Signed: *no* Width: *4* Offset: *0* Minimum: *0* Maximum: *15* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} *(Raw+0)* Resolution: *1*

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

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

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

	Offset: *0* Minimum: *0* Maximum: *15* Scaling (A,B,C): *1, 0, 0* 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 (A,B,C): *1, 0, 0* 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 (A,B,C): *1, 0, 0* 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 (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} *(Raw+0)* Resolution: *1*
digit37	Digit 37 (hexadecimal digit) .

	bitfield Signed: *no* Width: *4* Offset: *0* Minimum: *0* Maximum: *15* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} *(Raw+0)* Resolution: *1*
digit38	Digit 38 (hexadecimal digit) .
	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*

Formats:

SNVT_ISO_7811: 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,%d,%d,%d,%d,%d,%d,%d,%d,%d,%d,%d,%d,%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)

SNVT_ISO_7811#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|%d|%d|%d|%d|%d|%d|%d|%d|%d|%d|%d|%d|%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)

SNVT_abs_humid

Overview:

Absolute humidity (gram/kilogram) .

Used for optimal control in heating, ventilation, and air conditioning applications. If a device

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

See also SNVT_enthalpy.

Details:

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

SNVT_address

Overview:

Neuron address (16-bit address value) .

Details:

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

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) .
	alarm_type_t
priority_level	Priority level (priority level names) .
	priority_level_t
index_to_SNVT	Index of NV (index of NV causing alarm) .
	unsigned long Minimum: *0* Maximum: *65535* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} *(Raw+0)* 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 (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} *(Raw+0)* Resolution: *1*
year	Year (years) . Zero (0) means year not specified.
	unsigned long Minimum: *0* Maximum: *3000* Scaling (A,B,C): *1, 0, 0* 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* Scaled value: *1 *10 ^{0} *(Raw+0)* Resolution: *1*
millisecond	Millisecond (milliseconds) .
	unsigned long Minimum: *0* Maximum: *999* Scaling (A,B,C): *1, 0, 0* 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 (A,B,C): *1, 0, 0* 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
	alarm_type_t
priority_level	Priority level (priority level names) . Priority level of the alarm reported by this update
	priority_level_t
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: *0* Maximum: *4294967295* Invalid: *4294967295* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} *(Raw+0)* Resolution: *1*
milliseconds	Milliseconds (milliseconds) . Alarm time in milliseconds since the second specified by the alarm_time field
	signed long Minimum: *-1* Maximum: *999* Invalid: *-1* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} *(Raw+0)* Resolution: *1*
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 (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} *(Raw+0)* Resolution: *1*
	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. 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 *Language File String Reference* below. # Language File String Reference 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. The syntax for a string reference is as follows: `\x80[`*`programIDTemplate`*`-][`*`scopeSpecifier`*`:]` *`languageStringIndex`*`;` The components of a string reference are the following: A byte containing the value 0x80, represented by the "**\x80**" string. • *programIDTemplate*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

description	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. • A hyphen (“-”) follows the program ID template. The hyphen is not included if the program ID template is not included, otherwise it is mandatory. • *scopeSpecifier*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. • *languageStringIndex*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. **EXAMPLES** The following string reference specifies language string index 522 within the standard resource file set. "\x80522;" 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. "\x803:100;" The following string reference specifies language string index 200 within a user resource file set with a program ID template of 80:00:9F:00:00:00:00:00 at scope 3. "\x8080009F-3:200;"
	unsigned char [22] Minimum: *32* Maximum: *126* Scaling (A,B,C): *1, 0, 0* Scaled *1 *10 ^{0} *(Raw+0)*

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

SNVT_amp_f

Overview:

Electric current (Amperes) .

Details:

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

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

SNVT_angle_f

Overview:

Angular distance (radians) .

Details:

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

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

SNVT_angle_vel

Overview:

Angular velocity (radians/second) .

Details:

Standard: *yes*
Resource Set: *Standard 00:00:00:00:00:00:00:00-0*
Index: *4*
Obsolete: *no*
Size: *2*
Programmatic Name: *SNVT_angle_vel*
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_angle_vel#SI: text("%f")*

*SNVT_angle_vel#US: text("%f", *9.5493+0(0:970))*

SNVT_angle_vel_f

Overview:

Angular velocity (radians/second) .

Details:

Standard: *yes*
Resource Set: *Standard 00:00:00:00:00:00:00:00-0*
Index: *50*
Obsolete: *no*
Size: *4*

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) .
	chiller_t
chlr_op_mode	Chiller operating mode (HVAC mode names) .
	hvac_t
chlr_state	Chiller state flags (alarm, enabled, local, limited, chiller water flow, condenser water flow) .

Structure	
in_alarm	Alarm flag (boolean) .

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*
run_enabled	Run-enabled flag (boolean) .
	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*
local	Locally-controlled flag (boolean) .
	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*
limited	Limited-condition flag (boolean) . Conditions may exist that prevent reaching the setpoint
	bitfield Signed: *no* Width: *1* Offset: *3* Minimum: *0* Maximum: *1* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} *(Raw+0)* Resolution: *1*
chw_flow	Chiller-water-flow flag (boolean) .
	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*
	condw_flow	Condenser-water-flow flag (boolean) .
		bitfield Signed: *no* Width: *1* Offset: *5* Minimum: *0* Maximum: *1* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} *(Raw+0)* Resolution: *1*

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: *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*
	war_water_supply	.
		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*
	war_drain_slow	.
		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*
	war_door_open	.
		bitfield Signed: *no* Width: *1* Offset: *3*

	Minimum: *0* Maximum: *1* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} * (Raw+0)* Resolution: *1*
war_load_unbalanced	.
	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*
war_filter_cleaning	.
	bitfield Signed: *no* Width: *1* Offset: *5* Minimum: *0* Maximum: *1* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} * (Raw+0)* Resolution: *1*
war_hoses_reversed	.
	bitfield Signed: *no* Width: *1* Offset: *6* Minimum: *0* Maximum: *1* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} * (Raw+0)* Resolution: *1*
war_voltage_low	.

	bitfield Signed: *no* Width: *1* Offset: *7* Minimum: *0* Maximum: *1* Scaling (A,B,C): *1, 0, 0* Scaled value: $1 * 10^{0} * (Raw+0)$ Resolution: *1*
war_power_failure	.
	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*
war_drain_open	.
	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*
war_execute_fail	.
	bitfield Signed: *no* Width: *1* Offset: *2* Minimum: *0* Maximum: *1* Scaling (A,B,C): *1, 0, 0* Scaled $1 * 10^{0} *$

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

	Maximum: *1* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} * (Raw+0)* Resolution: *1*
rsrvd7	.
	bitfield Signed: *no* Width: *1* Offset: *7* Minimum: *0* Maximum: *1* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} * (Raw+0)* Resolution: *1*
err_motor_stall	.
	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*
err_water_temp	.
	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*
err_pressure	.
	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*
err_overflow	.
	bitfield Signed: *no* 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: *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*
err_water_leak	.
	bitfield Signed: *no* Width: *1* Offset: *5* Minimum: *0* Maximum: *1* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} * (Raw+0)*

	Resolution: *1*
err_motor_speed	.
	bitfield Signed: *no* Width: *1* Offset: *6* Minimum: *0* Maximum: *1* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} * (Raw+0)* Resolution: *1*
err_wash_thermistor	.
	bitfield Signed: *no* Width: *1* Offset: *7* Minimum: *0* Maximum: *1* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} * (Raw+0)* Resolution: *1*
err_dry_thermistor	.
	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*
err_dry_overheat	.
	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*
err_dry_heating	.
	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*
err_dry_fan	.
	bitfield Signed: *no* 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_rsrvd4	.
	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*
err_rsrvd5	.
	bitfield Signed: *no*

	Width: *1* Offset: *5* Minimum: *0* Maximum: *1* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} * (Raw+0)* Resolution: *1*
err_rsrvd6	.
	bitfield Signed: *no* Width: *1* Offset: *6* Minimum: *0* Maximum: *1* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} * (Raw+0)* Resolution: *1*
err_rsrvd7	.
	bitfield Signed: *no* Width: *1* Offset: *7* Minimum: *0* Maximum: *1* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} * (Raw+0)* Resolution: *1*
err_rsrvd0_7	.
	bitfield Signed: *no* Width: *8* Offset: *0* Minimum: *0* Maximum: *1* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} * (Raw+0)* Resolution: *1*

	manuf_code	.
		unsigned short Minimum: *0* Maximum: *255* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} * (Raw+0)* Resolution: *1*

Formats: *SNVT_clothes_w_a: 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 %d %d %d %d %d %d %d", 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.rsrvd5, alarm.rsrvd6, alarm.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_clothes_w_c

Overview:

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

Details:

Standard: *yes*

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

Index: *184*

Obsolete: *no*

Size: *21*

Programmatic Name: *SNVT_clothes_w_c*

Neuron C Type:

Structure	
cycle	.
	appl_cwc_t

subcycle	.
	appl_cws_t
rervd	.
	unsigned short Minimum: *0* Maximum: *255* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} *(Raw+0)* Resolution: *1*
action	.

Structure	
power_on	.
	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*
run_mode	.
	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*
rsrvd2_7	.
	bitfield Signed: *no* Width: *6* Offset: *2* Minimum: *0* Maximum: *63*

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

	value: *(Raw+0)* Resolution: *1*
repeat	.
	unsigned short Minimum: *0* Maximum: *9* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 0 * (Raw+0)* Resolution: *1*
option	.
	appl_rin_t

spin	.

Structure	
speed	.
	SNVT_rpm
time	.
	unsigned short Minimum: *0* Maximum: *255* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 0 * (Raw+0)* Resolution: *1*
hold	.
	boolean_t

dry	.

Structure	
temp	.
	unsigned short Minimum: *0* Maximum: *1* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 0 *(Raw+0)* Resolution: *1*
duration	.

Structure

			time	.
				unsigned short Minimum: *0* Maximum: *255* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} * (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: *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*
drain_on	Drain On. The drain is on
	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*
reserved	This field is reserved.. This field is reserved.
	bitfield Signed: *no* Width: *6* Offset: *2* Minimum: *0* Maximum: *63* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} *(Raw+0)* Resolution: *1*

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:

<table>
<tr><td colspan="3">Structure</td></tr>
<tr><td>cycle</td><td colspan="2">.</td></tr>
<tr><td></td><td colspan="2">appl_cwc_t</td></tr>
<tr><td>subcycle</td><td colspan="2">.</td></tr>
<tr><td></td><td colspan="2">appl_cws_t</td></tr>
<tr><td>washer_command_data</td><td colspan="2">.</td></tr>
<tr><td></td><td colspan="2">SNVT_clothes_w_c</td></tr>
<tr><td>time_remaining</td><td colspan="2">.</td></tr>
<tr><td></td><td colspan="2">SNVT_time_min</td></tr>
<tr><td>alarm</td><td colspan="2">.</td></tr>
<tr><td rowspan="6"></td><td colspan="2">Structure</td></tr>
<tr><td>alarm_reset</td><td>.</td></tr>
<tr><td></td><td>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</td></tr>
<tr><td>war_water_supply</td><td>.</td></tr>
<tr><td></td><td>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)</td></tr>
</table>

	Resolution: *1*
war_drain_slow	.
	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*
war_door_open	.
	bitfield Signed: *no* Width: *1* Offset: *3* Minimum: *0* Maximum: *1* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} * (Raw+0)* Resolution: *1*
war_load_unbalanced	.
	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*
war_filter_cleaning	.
	bitfield Signed: *no* Width: *1* Offset: *5* Minimum: *0* Maximum: *1*

	Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} * (Raw+0)* Resolution: *1*
war_hoses_reversed	.
	bitfield Signed: *no* Width: *1* Offset: *6* Minimum: *0* Maximum: *1* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} * (Raw+0)* Resolution: *1*
war_voltage_low	.
	bitfield Signed: *no* Width: *1* Offset: *7* Minimum: *0* Maximum: *1* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} * (Raw+0)* Resolution: *1*
war_power_failure	.
	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*
war_drain_open	.
	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*
war_execute_fail	.
	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*
war_door_locked	.
	bitfield Signed: *no* Width: *1* Offset: *3* Minimum: *0* Maximum: *1* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} * (Raw+0)* Resolution: *1*
war_service	.
	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*

war_rsrvd5	.
	bitfield Signed: *no* Width: *1* Offset: *5* Minimum: *0* Maximum: *1* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} * (Raw+0)* Resolution: *1*
war_rsrvd6	.
	bitfield Signed: *no* Width: *1* Offset: *6* Minimum: *0* Maximum: *1* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} * (Raw+0)* Resolution: *1*
war_rsrvd7	.
	bitfield Signed: *no* Width: *1* Offset: *7* Minimum: *0* Maximum: *1* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} * (Raw+0)* Resolution: *1*
err_motor_stall	.
	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*
err_water_temp	.
	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*
err_pressure	.
	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*
err_overflow	.
	bitfield Signed: *no* 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: *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*
err_water_leak	.
	bitfield Signed: *no* Width: *1* Offset: *5* Minimum: *0* Maximum: *1* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} * (Raw+0)* Resolution: *1*
err_motor_speed	.
	bitfield Signed: *no* Width: *1* Offset: *6* Minimum: *0* Maximum: *1* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} * (Raw+0)* Resolution: *1*
err_wash_thermistor	.
	bitfield Signed: *no* Width: *1* Offset: *7* Minimum: *0* Maximum: *1* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} * (Raw+0)* Resolution: *1*
err_dry_thermistor	.

	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*
err_dry_overheat	.
	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*
err_dry_heating	.
	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*
err_dry_fan	.
	bitfield Signed: *no* Width: *1* Offset: *3* Minimum: *0* Maximum: *1* Scaling (A,B,C): *1, 0, 0* Scaled *1 *10 ^{0} **

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

		Maximum: *1* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} * (Raw+0)* Resolution: *1*
	err_rsrvd0_7	.
		bitfield Signed: *no* Width: *8* Offset: *0* Minimum: *0* Maximum: *255* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} * (Raw+0)* Resolution: *1*
	manuf_code	.
		unsigned short Minimum: *0* Maximum: *255* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} * (Raw+0)* Resolution: *1*

Formats: *SNVT_clothes_w_s: text("%m %m %m %m %d %d %d %d %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 %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 (A,B,C): *1, -1, 0* Scaled value: $1 * 10^{-1} * (Raw+0)$ Resolution: *0.1*
a_star	a*.
	signed long Minimum: *-2000* Maximum: *2000*

	Scaling (A,B,C): *1, -1, 0* Scaled value: *1 *10 -1 *(Raw+0)* Resolution: *0.1*
b_star	b*.
	signed long Minimum: *-2000* Maximum: *2000* Scaling (A,B,C): *1, -1, 0* Scaled value: *1 *10 -1 *(Raw+0)* 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: *yes*
Resource Set: *Standard 00:00:00:00:00:00:00:00-0*
Index: *190*
Obsolete: *no*
Size: *5*
Programmatic Name: *SNVT_color_2*
Neuron C Type:

Structure	
encoding	Color encoding.. Color encoding specified by the color_value union; additional encodings may be added
	color_encoding_t
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 (A,B,C): *5, -1, 0* 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 (A,B,C): *1, 0, 0* Scaled value: $1 * 10^{0} * (Raw+0)$ Resolution: *1*
		absolute_Y	Absolute luminance (lumen) . Absolute luminance
			unsigned long Minimum: *0* Maximum: *65534* Invalid: *65535* Scaling (A,B,C): *1, 1, 0* Scaled value: $1 * 10^{1} * (Raw+0)$ 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 (A,B,C): *5, -1, 0* 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 (A,B,C): *1, 0, 0* 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 (A,B,C): *1, -2, 0* 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 (A,B,C): *1, 0, 0*

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

Used by: *SCPTsceneColor* *SFPTisiLampActuator*

SNVT_config_src

Overview:

Configuration source (configuration source names) .

Details:

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

SNVT_count

Overview:

Absolute count (units) .

Details:

Standard:	*yes*
Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*8*
Obsolete:	*no*
Size:	*2*
Programmatic Name:	*SNVT_count*
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_count: text("%d")*
Used by:	*SCPTnumDampers SCPTnumDigits SCPTnumValves SFPTclosedLoopSensor SFPTdeviceMonitor SFPTelevatorArrivalGong SFPTelevatorHallLantern SFPTelevatorPositionIndicator SFPTelevatorVoiceAnnouncer SFPTgeneratorSet SFPThvacValvePositioner SFPTopenLoopSensor SFPTutilityDataLoggerRegister*

SNVT_count_32

Overview:

Absolute count. A 32-bit counter

Details:

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

SNVT_count_f

Overview:

Absolute count (units) .

Details:

Standard:	*yes*
Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*51*
Obsolete:	*no*
Size:	*4*
Programmatic Name:	*SNVT_count_f*
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: *1* Maximum: *65535* Invalid: *0* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} *(Raw+0)* Resolution: *1*
sender_id	Sender ID (ID number) .
	unsigned long Minimum: *1* Maximum: *65535* Invalid: *65535* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} *(Raw+0)* Resolution: *1*
sender_prio	Sender priority (priority value) .
	unsigned short Minimum: *0* Maximum: *200* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} *(Raw+0)* Resolution: *1*

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) .
	control_resp_t
sender	Sender ID.

Union	
id	Sender ID (ID number) .
	unsigned long Minimum: *1* Maximum: *65535* Invalid: *65535* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} *(Raw+0)* Resolution: *1*
range	Sender ID range (lower, upper) .

Structure	
lower	Sender range lower ID (ID number) .
	unsigned long Minimum: *1* Maximum: *65535* Invalid: *65535* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} * (Raw+0)* Resolution: *1*
upper	Sender range upper ID (ID number) .
	unsigned long Minimum: *1* Maximum: *65535* Invalid: *65535* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} * (Raw+0)* Resolution: *1*

controller_id	Controller ID (ID number) .
	unsigned long Minimum: *1* Maximum: *65535* Invalid: *65535*

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) .
	currency_t
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: *yes*
Resource Set: *Standard 00:00:00:00:00:00:00:00-0*
Index: *10*
Obsolete: *yes*
Size: *4*
Programmatic Name: *SNVT_date_cal*
Neuron C Type:

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

	Maximum: *3000* Invalid: *65535* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} *(Raw+0)* Resolution: *1*
month	Month (months) .
	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) .
	unsigned short Minimum: *0* Maximum: *31* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} *(Raw+0)* Resolution: *1*

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

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

SNVT_defr_term

Overview:

Defrost termination (defrost termination names) .

Details:

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

SNVT_density

Overview:

Density (kilograms/cubic meter) .

Details:

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

SNVT_density_f

Overview:

Density (kilograms/cubic meter) .

Details:

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

Formats:	*SNVT_density_f#SI: text("%f", *1+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*

Obsolete: *no*

Size: *4*

Programmatic Name: *SNVT_dev_fault*

Neuron C Type:

Structure	
device_select	Device selection (device selec
	device_select_t
dev_type	Union of device fault structure

Union
pump_ctrl

valve_pos

Formats:	*SNVT_dev_fault: text(("DS=%m ",device_select),((device_select == 0) ?("SF: [VL=%* 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
	device_select_t
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 I ((device_select == 1) ?("MM=%d PC=%d EC=%d PC=%d LC=%d RC=%d BC=%d",dev_type.valve_pos.motor_maint,dev_type.valve_pos.packing_change,dev_type.valv ("UNKNOWN"))))*
Used by:	*SFPThvacValvePositioner SFPTpumpController*

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
	device_selec
dev_type	Union of dev
	Union
	pump_ctrl

valve_pos

Formats: *SNVT_dev_status: text(("DS=%m ",device_select),((device_select == 0) ?("DF=%d S dev_type.pump_ctrl.device_fault,dev_type.pump_ctrl.supply_fault,dev_type.pump_ctrl. ((device_select == 1) ?("R=%d A=%d I=%d LC=%d SOR=%d RCS=%d {%d} HW=*

Used by: *SFPThvacValvePositioner SFPTpumpController*

SNVT_earth_pos

Overview:

Earth position (lat & long direction, latitude deg & min, longitude deg & min, height) .

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: *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*
longitude_direction	Direction of longitude (direction (E/W)) . 0 = East longitude, 1 = West longitude
	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*
latitude_deg	Latitude degrees (degrees) .
	unsigned short Minimum: *0* Maximum: *90* Invalid: *255* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} *(Raw+0)* Resolution: *1*
latitude_min	Latitude minutes (minutes) .
	unsigned long Minimum: *0*

	Maximum: *59999* Invalid: *65535* Scaling (A,B,C): *1, -3, 0* Scaled value: *1 *10 $^{-3}$ *(Raw+0)* Resolution: *0.001*
longitude_deg	Longitude degrees (degrees) .
	unsigned short Minimum: *0* Maximum: *180* Invalid: *255* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} *(Raw+0)* Resolution: *1*
longitude_min	Longitude minutes (minutes) .
	unsigned long Minimum: *0* Maximum: *59999* Invalid: *65535* Scaling (A,B,C): *1, -3, 0* 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: *0* Maximum: *65535* Invalid: *65535* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} *(Raw+0)* Resolution: *1*
hour	Hours (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	Minutes (minutes) .
	unsigned short Minimum: *0* Maximum: *59* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} *(Raw+0)* Resolution: *1*
second	Seconds (seconds) .
	unsigned short Minimum: *0*

	Maximum: *59* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} *(Raw+0)* Resolution: *1*
millisecond	Milliseconds (milliseconds) .
	unsigned long Minimum: *0* Maximum: *999* Scaling (A,B,C): *1, 0, 0* 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 SFPThvacValvePositioner SFPTisiKeypad SFPTisiLampActuator SFPTisiSunblindActuator SFPTlampActuator SFPTnodeObject SFPTutilityDataLoggerRegister*

SNVT_elec_kwh

Overview:

Electric energy (kiloWatt-hours) .

For new designs, SNVT_elec_kwh_l 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:	*SNVT_elec_kwh: text("%d")*
Used by:	*SCPTenergyCntInit SFPTisiLampActuator SFPTlampActuator SFPTpumpController*

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:	*yes*
Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*146*
Obsolete:	*no*
Size:	*4*
Programmatic Name:	*SNVT_elec_kwh_l*
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:	*SNVT_elec_kwh_l: text("%f")*

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

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

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

	state.
	bitfield Signed: *no* Width: *1* Offset: *5* Minimum: *0* Maximum: *1* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} *(Raw+0)* Resolution: *1*
in_error_cond	In error condition (boolean) . Device has an error condition
	bitfield Signed: *no* Width: *1* Offset: *6* Minimum: *0* Maximum: *1* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} *(Raw+0)* Resolution: *1*
open_pre_alarm	Open device, pre-alarm (boolean) . Device is open, and in warning state
	bitfield Signed: *no* Width: *1* Offset: *7* Minimum: *0* Maximum: *1* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} *(Raw+0)* Resolution: *1*
open_alarm	Open Device, alarm state (boolean) . Device is open, and in not-closed alarm state
	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*
service_alarm	Service alarm (boolean) . Device needs service
	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*
tamper	Tamper mode (boolean) . Device has detected tamper
	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*
entry_req	Entry request pending (boolean) . Device has a pending entry request
	bitfield Signed: *no* Width: *1* Offset: *3* Minimum: *0* Maximum: *1* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} *(Raw+0)* Resolution: *1*
exit_req	Exit request pending (boolean) . Device has a pending exit request
	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*
key_req	Key request pending (boolean) . Device has a pending key request
	bitfield Signed: *no* Width: *1* Offset: *5* Minimum: *0* Maximum: *1* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} *(Raw+0)* Resolution: *1*
safety_ext_req	Safety-exit request pending (boolean) . Device has a pending safety-exit request
	bitfield Signed: *no* Width: *1* Offset: *6* Minimum: *0* Maximum: *1* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} *(Raw+0)* Resolution: *1*
emergency_req	Emergency-exit request pending (boolean) . Device has a pending emergency-exit request
	bitfield Signed: *no* Width: *1* Offset: *7* Minimum: *0* Maximum: *1* Scaling (A,B,C): *1, 0, 0*

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

	value: Resolution: *1*
moving	Device is in motion (boolean) . Device is currently changing position
	bitfield Signed: *no* Width: *1* Offset: *7* Minimum: *0* Maximum: *1* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} *(Raw+0)* Resolution: *1*
stopped	Device Stopped (boolean) . The device is stopped and can be moved manually
	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*
safety_alarm	Safety-alarm (boolean) . Device is in a safety-alarm state
	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*
unknown_state	Unknown state (boolean) . The state of the device is currently unknown
	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*
cmd_fb	Command feedback (entry command names) . Feedback of requested-operation-mode of device
	ent_opmode_cmd_t

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 %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: *yes*

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

Index: *153*

Obsolete: *no*

Size: *2*

Programmatic Name: *SNVT_enthalpy*

Neuron C Type: signed long

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:

<table>
<tr><td colspan="3">Structure</td></tr>
<tr><td>control_status</td><td colspan="2">Control type (control type names) .</td></tr>
<tr><td></td><td colspan="2">ex_control_t</td></tr>
<tr><td>control_device_addr</td><td colspan="2">Control device address (LonWorks subnet-node address) .</td></tr>
<tr><td></td><td colspan="2"><table>
<tr><td colspan="2">Structure</td></tr>
<tr><td>domain_id</td><td>Domain ID (array of 6 bytes) . ANSI/CEA-709.1 domain ID</td></tr>
<tr><td></td><td>unsigned short [6]
Minimum: 0
Maximum: 255
Scaling (A,B,C): 1, 0, 0
Scaled value: 1 *10 ⁰ *(Raw+0)
Resolution: 1</td></tr>
<tr><td>domain_length</td><td>Domain length (ANSI/CEA-709.1 domain length) . Valid domain lengths are 0, 1, 3, and 6.</td></tr>
<tr><td></td><td>unsigned short
Minimum: 0
Maximum: 6
Scaling (A,B,C): 1, 0, 0
Scaled value: 1 *10 ⁰ *(Raw+0)
Resolution: 1</td></tr>
<tr><td>subnet</td><td>Subnet (subnet number) . There can be 255 subnets (1-255) in a domain.</td></tr>
<tr><td></td><td>unsigned short
Minimum: 0
Maximum: 255</td></tr>
</table></td></tr>
</table>

		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 (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} *(Raw+0)* Resolution: *1*
rw_length	Read/Write length (number of bytes) .
	unsigned long Minimum: *0* Maximum: *65535* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} *(Raw+0)* 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) .
	file_request_t
index	Index (file index) .

	unsigned long Minimum: *0* Maximum: *65535* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} *(Raw+0)* Resolution: *1*
receive_timeout	Receive timeout (milliseconds) .
	unsigned long Minimum: *0* Maximum: *65535* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} *(Raw+0)* Resolution: *1*
dest_address	Destination address (LonWorks address) .

Union	
addrt	Address table entry (Address table entry) . ANSI/CEA-709.1 address in device's internal address table entry

Structure	
type	Address type (8-bit unsigned value) . The address-table address type is 33 (0x21).
	unsigned short Minimum: *33* Maximum: *33* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} * (Raw+0)* Resolution: *1*
index	Address table index (16-bit unsigned value) .
	unsigned long Minimum: *0* Maximum: *65535* Scaling (A,B,C): *1, 0, 0* 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 (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} * (Raw+0)* Resolution: *1*
domain	Domain (LonWorks domain index) .
	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*
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 (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} * (Raw+0)* Resolution: *1*
retry	Retry count (number of retries) .
	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*
		tx_timer	Transaction timer (timer code value) .
			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*
		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*
	gp	Group address (LonWorks group address) .	
		Structure	
		type	Address type (boolean) . The group address type is 1.
			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*
size	Size (LonWorks group size) . An acknowledged group can have from 0-64 addressees, plus the sender.
	bitfield Signed: *no* Width: *7* Offset: *1* Minimum: *0* Maximum: *65* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} * (Raw+0)* Resolution: *1*
domain	Domain (LonWorks domain index) .
	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*
unused	Unused field. This field is reserved.
	bitfield Signed: *no* Width: *7* Offset: *1* Minimum: *0* Maximum: *0* Scaling (A,B,C): *1, 0, 0*

			Scaled value: *1 *10 ^{0} * (Raw+0)* Resolution: *1*
		retry	Retry count (number of retries) .
			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*
		tx_timer	Transaction timer (timer code value) .
			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*
		group	Group. There can be 256 groups (0-255) in a domain.
			unsigned short Minimum: *0* Maximum: *255* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} * (Raw+0)* Resolution: *1*
auth_on	Authentication on (boolean) . This field specifies whether the message requires authentication.		
	unsigned short Minimum: *0* Maximum: *1*		

	Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} *(Raw+0)* Resolution: *1*
prio_on	Priority on (boolean) . This field specifies whether the message is to be sent with priority.
	unsigned short Minimum: *0* Maximum: *1* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} *(Raw+0)* Resolution: *1*

Formats: *SNVT_file_req: text(("%m,%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 ",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: *yes*
Resource Set: *Standard 00:00:00:00:00:00:00:00-0*
Index: *74*
Obsolete: *no*
Size: *27*

Programmatic Name: *SNVT_file_status*

Neuron C Type:

Structure	
status	Status (file status names) .
	file_status_t
number_of_files	Number of files (count) .
	unsigned long Minimum: *0* Maximum: *65535* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} *(Raw+0)* Resolution: *1*
selected_file	Selected file (file index) .
	unsigned long Minimum: *0* Maximum: *65535* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} *(Raw+0)* Resolution: *1*
adr	Address.

Union	
descriptor	Descriptor.

Structure	
file_info	File info (array of 16 characters) .
	signed char [16] Minimum: *-128* Maximum: *127* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} * (Raw+0)* Resolution: *1*
size	Size (bytes) .
	s32_type Minimum: *0* Maximum: *2147483647* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} * (Raw+0)*

	Resolution: *1*
type	Type.
	unsigned long Minimum: *0* Maximum: *65535* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} * (Raw+0)* Resolution: *1*
address	Address.
	Structure
domain_id	Domain ID (array of 6 bytes) . ANSI/CEA-709.1 domain ID
	unsigned short [6] Minimum: *0* Maximum: *255* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} * (Raw+0)* 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 (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} * (Raw+0)* 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 *1, 0, 0*

			(A,B,C): 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_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:	*no*
Size:	*1*
Programmatic Name:	*SNVT_fire_indcte*
Neuron C Type:	*fire_indicator_t*
Formats:	*SNVT_fire_indcte: text("%m")*
Used by:	*SCPTfireIndicate*

SNVT_fire_init

Overview:

Fire initiator type (fire initiator type names) .

Details:

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

SNVT_fire_test

Overview:

Fire test request (fire test names) .

Details:

Standard:	*yes*
Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*130*
Obsolete:	*no*
Size:	*1*

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 *10 ^{0} *(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:	*yes*
Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*171*
Obsolete:	*no*
Size:	*1*
Programmatic Name:	*SNVT_flow_dir*
Neuron C Type:	*flow_direction_t*
Formats:	*SNVT_flow_dir: text("%m")*
Used by:	*SFPTentryExit*

SNVT_flow_f

Overview:

Flow volume (liters/second) .

Details:

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

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 SFPTairVelocitySensor*

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

SNVT_freq_f

Overview:

Frequency (Hertz) .

Details:

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

SNVT_freq_hz

Overview:

Frequency (Hertz) .

Details:

Standard:	*yes*
Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*76*
Obsolete:	*no*
Size:	*2*
Programmatic Name:	*SNVT_freq_hz*
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_hz: text("%f")*
Used by:	*SCPTflashFreq SCPTnomFreq SCPTnormalRotationalSpeed SCPTstandbyRotationalSpeed SFPTautomaticTransferSwitch SFPTgeneratorSet SNVT_pump_sensor*

SNVT_freq_kilohz

Overview:

Frequency (kiloHertz) .

Details:

Standard:	*yes*
Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*77*
Obsolete:	*no*
Size:	*2*
Programmatic Name:	*SNVT_freq_kilohz*
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 SFPThvacValvePositioner 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:	*yes*
Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*108*
Obsolete:	*no*
Size:	*1*
Programmatic Name:	*SNVT_hvac_mode*
Neuron C Type:	*hvac_t*
Formats:	*SNVT_hvac_mode: text("%m")*
Used by:	*SCPThvacMode SFPTboilerController SFPTchilledCeilingController SFPTchiller SFPTdischargeAirController SFPTfanCoilUnit SFPTheatPump SFPThvacValvePositioner SFPTpumpController SFPTroofTopUnit SFPTsccAHU SFPTsccChilledCeiling SFPTsccCommandModule SFPTsccFanCoil SFPTsccHeatPump SFPTsccRadiator SFPTsccRooftop SFPTsccSelfContained SFPTsccUnitVentilator SFPTsccVAV SFPTspaceComfortController SFPTthermostat SFPTunitVentilatorController SFPTvariableAirVolume*

SNVT_hvac_overid

Overview:

HVAC override (state, pct, flow) .

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

Details:

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

state	HVAC override state (override state names) .
	hvac_overid_t
percent	Percent (% of full scale) . Position or flow override value
	signed long Minimum: *-32768* Maximum: *32767* Invalid: *32767* Scaling (A,B,C): *5, -3, 0* Scaled value: *5 *10 $^{-3}$ *(Raw+0)* Resolution: *0.005*
flow	Flow (liters/second) .
	unsigned long Minimum: *0* Maximum: *65535* Invalid: *65535* Scaling (A,B,C): *1, 0, 0* 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: *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*
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: *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*
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: *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*
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: *no* Width: *1* Offset: *3* Minimum: *0* Maximum: *1* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} *(Raw+0)* Resolution: *1*
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: *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*
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: *no* Width: *1* Offset: *5* 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: *2* Offset: *6* Minimum: *0* Maximum: *3* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} *(Raw+0)* Resolution: *1*
reserved1	.
	bitfield Signed: *no* Width: *4* Offset: *0* Minimum: *0* Maximum: *15* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} *(Raw+0)* Resolution: *1*
manufacturer_defined	Manufacturer defined (boolean) . Four manufacturer-defined bits -- please see product documentation for proper interpretation of these bits
	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*

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) .
	hvac_t
heat_output_primary	Primary heat output (% of full scale) .
	signed long Minimum: *-32768* Maximum: *32767* Invalid: *32767* Scaling (A,B,C): *5, -3, 0* 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 (A,B,C): *5, -3, 0* 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 (A,B,C): *5, -3, 0* 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 (A,B,C): *5, -3, 0* 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 (A,B,C): *5, -3, 0* 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 (A,B,C): *1, 0, 0*

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

SNVT_length

Overview:

Length (meters) .

Details:

Standard:	*yes*
Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*17*
Obsolete:	*no*
Size:	*2*
Programmatic Name:	*SNVT_length*
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#SI: text("%f")* *SNVT_length#US: text("%f", *3.28084+0(0:840))*
Used by:	*SFPThvacValvePositioner*

SNVT_length_f

Overview:

Length (meters) .

Details:

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

SNVT_length_kilo

Overview:

Length (kilometers) .

Details:

Standard:	*yes*
Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*18*
Obsolete:	*no*
Size:	*2*
Programmatic Name:	*SNVT_length_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_length_kilo#SI: text("%f")* *SNVT_length_kilo#US: text("%f", *0.62137+0(0:841))*

SNVT_length_micr

Overview:

Length (micrometers (microns)) .

Details:

Standard:	*yes*
Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*19*
Obsolete:	*no*
Size:	*2*
Programmatic Name:	*SNVT_length_micr*
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:	*yes*
Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*21*
Obsolete:	*no*
Size:	*1*
Programmatic Name:	*SNVT_lev_cont*
Neuron C Type:	unsigned short Minimum: *0* Maximum: *200* Scaling (A,B,C): *5, -1, 0* Scaled value: *5 *10 $^{-1}$ *(Raw+0)* Resolution: *0.5*
Formats:	*SNVT_lev_cont: text("%f")*
Used by:	*SCPTmaxOut SCPTminDeltaLevel SCPTonOffHysteresis SCPTstep SCPTstepValue SFPTchannelMonitor SFPThvacRelativeHumiditySensor SFPTsmokeFireInitiatorIntelli*

SNVT_lev_cont_f

Overview:

Continuous level (% of full scale) .

Details:

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

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:	*yes*
Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*22*
Obsolete:	*yes*
Size:	*1*
Programmatic Name:	*SNVT_lev_disc*
Neuron C Type:	*discrete_levels_t*
Formats:	*SNVT_lev_disc: text("%m")*
Used by:	*SCPTinvrtOut SFPTrefrigDisplayCaseControllerDefrost SFPTrefrigDisplayCaseControllerThermostat SFPTutilityDataLoggerRegister*

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:	*yes*

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

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: *1* Maximum: *65535* Invalid: *0* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} *(Raw+0)* Resolution: *1*
record_count	Record count.. The maximum number of log records to be transferred.
	unsigned quad Minimum: *0* Maximum: *4294967295* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} *(Raw+0)* Resolution: *1*
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.
	SNVT_time_stamp_p
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.
	SNVT_time_stamp_p

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: *1* Maximum: *255* Invalid: *0* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} *(Raw+0)* Resolution: *1*
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: *1* Maximum: *255* Invalid: *0* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} *(Raw+0)* Resolution: *1*
log_number	Data log number.. Log number for the log file currently being transferred via FTP. Invalid if none.
	unsigned long Minimum: *1* Maximum: *65535* Invalid: *0* Scaling (A,B,C): *1, 0, 0* 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: *yes*

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

Index: *191*

Obsolete: *no*

Size: *30*

Programmatic Name: *SNVT_log_status*

Neuron C Type:

Structure	
status	Log state.. State of the selected data log.
	log_status_t
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: *1* Maximum: *65535* Invalid: *0* Scaling (A,B,C): *1, 0, 0* Scaled value: $1 * 10^{0} * (Raw+0)$ Resolution: *1*
level	Log level (Percent) . The percent of maximum records in the selected data log.
	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*
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: *-2147483648* Maximum: *2147483647* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} *(Raw+0)* Resolution: *1*
byte_count	Byte count. (bytes) . Number of bytes in the selected data log.
	s32_type Minimum: *-2147483648* Maximum: *2147483647* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} *(Raw+0)* Resolution: *1*
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: *-2147483648* Maximum: *2147483647* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} *(Raw+0)* Resolution: *1*
records_since_notification	Records since notification. (records) . The number of log records collected since the last notificiation.
	s32_type Minimum: *-2147483648* Maximum: *2147483647* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} *(Raw+0)* Resolution: *1*
current_notify_time	Current notify time.. Timestamp of the most recently collected data point.
	SNVT_time_stamp_p
previous_notify_time	Previous notify time. (seconds) . Timestamp of the most recently collected data point in the previous update to the log status.
	SNVT_time_stamp_p

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:	*yes*
Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*79*
Obsolete:	*no*
Size:	*2*
Programmatic Name:	*SNVT_lux*
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_lux: text("%d")*
Used by:	*SCPTfieldCalib* *SCPTluxSetpoint* *SFPTconstantLightController* *SFPTlightSensor* *SFPTsunblindController*

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: *yes*

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

Index: *86*

Obsolete: *no*

Size: *20*

Programmatic Name: *SNVT_magcard*

Neuron C Type:

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

digit3	Digit 3 (hexadecimal digit) .
	bitfield Signed: *no* Width: *4* Offset: *0* Minimum: *0* Maximum: *15* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} *(Raw+0)* Resolution: *1*
digit4	Digit 4 (hexadecimal digit) .
	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*
digit5	Digit 5 (hexadecimal digit) .
	bitfield Signed: *no* Width: *4* Offset: *0* Minimum: *0* Maximum: *15* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} *(Raw+0)* Resolution: *1*
digit6	Digit 6 (hexadecimal digit) .
	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*
digit7	Digit 7 (hexadecimal digit) .
	bitfield Signed: *no* Width: *4* Offset: *0* Minimum: *0* Maximum: *15* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} *(Raw+0)* Resolution: *1*
digit8	Digit 8 (hexadecimal digit) .
	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*
digit9	Digit 9 (hexadecimal digit) .
	bitfield Signed: *no* Width: *4* Offset: *0* Minimum: *0* Maximum: *15* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} *(Raw+0)* Resolution: *1*
digit10	Digit 10 (hexadecimal digit) .
	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*
digit11	Digit 11 (hexadecimal digit) .
	bitfield Signed: *no* Width: *4* Offset: *0* Minimum: *0* Maximum: *15* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} *(Raw+0)* Resolution: *1*
digit12	Digit 12 (hexadecimal digit) .
	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*
digit13	Digit 13 (hexadecimal digit) .
	bitfield Signed: *no* Width: *4* Offset: *0* Minimum: *0* Maximum: *15* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} *(Raw+0)* Resolution: *1*
digit14	Digit 14 (hexadecimal digit) .
	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*
digit15	Digit 15 (hexadecimal digit) .
	bitfield Signed: *no* Width: *4* Offset: *0* Minimum: *0* Maximum: *15* Scaling (A,B,C): *1, 0, 0* 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 (A,B,C): *1, 0, 0* 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 (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} *(Raw+0)* Resolution: *1*
digit18	Digit 18 (hexadecimal digit) .

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

	Resolution: *1*
digit22	Digit 22 (hexadecimal digit) .
	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*
digit23	Digit 23 (hexadecimal digit) .
	bitfield Signed: *no* Width: *4* Offset: *0* Minimum: *0* Maximum: *15* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} *(Raw+0)* Resolution: *1*
digit24	Digit 24 (hexadecimal digit) .
	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*
digit25	Digit 25 (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*
digit26	Digit 26 (hexadecimal digit) .
	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*
digit27	Digit 27 (hexadecimal digit) .
	bitfield Signed: *no* Width: *4* Offset: *0* Minimum: *0* Maximum: *15* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} *(Raw+0)* Resolution: *1*
digit28	Digit 28 (hexadecimal digit) .
	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*
digit29	Digit 29 (hexadecimal digit) .
	bitfield Signed: *no* Width: *4* Offset: *0*

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

	Signed: *no* Width: *4* Offset: *0* Minimum: *0* Maximum: *15* Scaling (A,B,C): *1, 0, 0* 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 (A,B,C): *1, 0, 0* 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 (A,B,C): *1, 0, 0* 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 (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} *(Raw+0)* Resolution: *1*

digit37	Digit 37 (hexadecimal digit) .
	bitfield Signed: *no* Width: *4* Offset: *0* Minimum: *0* Maximum: *15* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} *(Raw+0)* Resolution: *1*
digit38	Digit 38 (hexadecimal digit) .
	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*
digit39	Digit 39 (hexadecimal digit) .
	bitfield Signed: *no* Width: *4* Offset: *0* Minimum: *0* Maximum: *15* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} *(Raw+0)* Resolution: *1*
digit40	Digit 40 (hexadecimal digit) .
	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*

Formats: *SNVT_magcard: 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,%d,%d,%d,%d,%d,%d,%d,%d,%d,%d,%d,%d,%d,%d,%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)*

SNVT_magcard#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|%d|%d|%d|%d|%d|%d|%d|%d|%d|%d|%d|%d|%d|%d|%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)

Used by: *SFPTidentifierSensor*

SNVT_mass

Overview:

Mass (grams) .

Details:

Standard:	*yes*
Resource Set:	*Standard 00:00:00:00:00:00:00-0*
Index:	*23*
Obsolete:	*no*
Size:	*2*
Programmatic Name:	*SNVT_mass*
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#SI: text("%f")* *SNVT_mass#US: text("%f", *0.03527+0(0:845))*

SNVT_mass_f

Overview:

Mass (grams) .

Details:

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

SNVT_mass_kilo

Overview:

Mass (kilograms) .

Details:

Standard:	*yes*
Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*24*
Obsolete:	*no*
Size:	*2*
Programmatic Name:	*SNVT_mass_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_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:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*26*
Obsolete:	*no*
Size:	*2*
Programmatic Name:	*SNVT_mass_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_mass_mil#SI: text("%f")* *SNVT_mass_mil#US: text("%f", *3.527e-5+0(0:845))*

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

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: *yes*

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

Index: *91*

Obsolete: *no*

Size: *4*

Programmatic Name: *SNVT_muldiv*

Neuron C Type:

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

Formats: *SNVT_muldiv: text("%d/%d", multiplier, divisor)*

Used by: *SCPTgain SCPTpulseValue*

SNVT_multiplier

Overview:

Multiplier (16-bit unsigned value) .

Details:

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

SNVT_multiplier_s

Overview:

Multiplier. Value multiplier

Details:

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

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 (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} *(Raw+0)* 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 (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} *(Raw+0)* 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 (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} *(Raw+0)* Resolution: *1*
type_category	Type category (type category names) . Type category of the network variable type
	nv_type_category_t
type_length	Type length (bytes) . Length of the network variable type
	unsigned short Minimum: *1* Maximum: *31* Invalid: *0* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} *(Raw+0)* 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 (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} *(Raw+0)* Resolution: *1*
scaling_factor_b	Scaling factor b. Exponent 'b' where ScaledValue = a*(10**b)*(RawValue+c)
	signed long Minimum: *-32768* Maximum: *32767* Invalid: *32767* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} *(Raw+0)* Resolution: *1*
scaling_factor_c	Scaling Factor c. Offset 'c' where ScaledValue = a* (10**b)*(RawValue+c)
	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_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: *yes*

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

Index: *92*

Obsolete: *no*

Size: *3*

Programmatic Name: *SNVT_obj_request*

Neuron C Type:

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

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	
object_id	Object ID (object index) .
	unsigned long Minimum: *0* Maximum: *65535* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} *(Raw+0)* Resolution: *1*
invalid_id	Invalid-ID flag (boolean) .
	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*
invalid_request	Invalid-request flag (boolean) .
	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*
disabled	Disabled flag (boolean) .
	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*
out_of_limits	Out-of-limits flag (boolean) .
	bitfield Signed: *no* Width: *1* Offset: *3* Minimum: *0* Maximum: *1* Scaling (A,B,C): *1, 0, 0* 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 (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} *(Raw+0)* Resolution: *1*
out_of_service	Out-of-service flag (boolean) .
	bitfield

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

under_range	Under-range flag (boolean) .
	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*
electrical_fault	Electrical-fault flag (boolean) .
	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*
unable_to_measure	Unable-to-measure flag (boolean) .
	bitfield Signed: *no* Width: *1* Offset: *3* Minimum: *0* Maximum: *1* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} *(Raw+0)* Resolution: *1*
comm_failure	Communications-failure flag (boolean) .
	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*
fail_self_test	Failed-self-test flag (boolean) .
	bitfield Signed: *no* Width: *1* Offset: *5* Minimum: *0* Maximum: *1* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} *(Raw+0)* Resolution: *1*
self_test_in_progress	Self-test-in-progress flag (boolean) .
	bitfield Signed: *no* Width: *1* Offset: *6* Minimum: *0* Maximum: *1* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} *(Raw+0)* Resolution: *1*
locked_out	Locked-out flag (boolean) .
	bitfield Signed: *no* Width: *1* Offset: *7* Minimum: *0* Maximum: *1* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} *(Raw+0)* Resolution: *1*
manual_control	Manual-control flag (boolean) .
	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*
in_alarm	Input-alarm flag (boolean) .
	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*
in_override	Input-override flag (boolean) .
	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*
report_mask	Report-mask flag (boolean) .
	bitfield Signed: *no* Width: *1* Offset: *3* Minimum: *0* Maximum: *1* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} *(Raw+0)* Resolution: *1*
programming_mode	Programming-mode flag (boolean) .
	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*
programming_fail	Programming-fail flag (boolean) .
	bitfield Signed: *no* Width: *1* Offset: *5* Minimum: *0* Maximum: *1* Scaling (A,B,C): *1, 0, 0* 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 (A,B,C): *1, 0, 0* 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 (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} *(Raw+0)* Resolution: *1*
reserved2	reserved2. This field is reserved.

	bitfield
	Signed: *no*
	Width: *8*
	Offset: *0*
	Minimum: *0*
	Maximum: *0*
	Scaling (A,B,C): *1, 0, 0*
	Scaled value: *1 *10 ^{0} *(Raw+0)*
	Resolution: *1*

Formats: *SNVT_obj_status: 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,%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_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|%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)

Used by: *SFPTnodeObject*

SNVT_occupancy

Overview:

Occupancy (occupancy code names) .

Details:

Standard: *yes*
Resource Set: *Standard 00:00:00:00:00:00:00:00-0*
Index: *109*
Obsolete: *no*
Size: *1*
Programmatic Name: *SNVT_occupancy*
Neuron C Type: *occup_t*

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 (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} *(Raw+0)* Resolution: *1*
controller_id	Controller 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*
controller_prio	Controller priority (priority value) .
	unsigned short Minimum: *0* Maximum: *100* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} *(Raw+0)* Resolution: *1*
function	Camera function (camera function names) .
	cam_func_t
action	Camera action (camera action names) .
	cam_act_t
value	Function value.
	Union
	number: Action number (action number) .
	unsigned short Minimum: *0* Maximum: *255* Invalid: *0* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} *(Raw+0)* Resolution: *1*
	abspos: Function absolute values (pan, tilt,

	zoom) .
	Structure
	pan: Pan position.
	SNVT_angle_deg
	tilt: Tilt position.
	SNVT_angle_deg
	zoom: Zoom position.
	SNVT_lev_percent

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*

SFPTsccVAV SFPTspaceComfortController

SNVT_power_f

Overview:

Power (Watts) .

Details:

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

SNVT_power_kilo

Overview:

Power (kiloWatts) .

Details:

Standard:	*yes*
Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*28*
Obsolete:	*no*

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*

SFPTvariableAirVolume

SNVT_ppm_f

Overview:

Concentration (ppm) .

Details:

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

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) .
	learn_mode_t
selector	Selector (16-bit unsigned value) . The selector is used to choose which preset.
	unsigned long Minimum: *0* Maximum: *65535* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} *(Raw+0)* Resolution: *1*
value	Value (array of 4 bytes) .
	unsigned short [4] Minimum: *0* Maximum: *255* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} *(Raw+0)* Resolution: *1*
day	Days (days) . The value 65535 represents NULL or unknown elapsed time.
	unsigned long Minimum: *0* Maximum: *65535* Invalid: *65535* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} *(Raw+0)* Resolution: *1*
hour	Hours (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	Minutes (minutes) .
	unsigned short Minimum: *0* Maximum: *59* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} *(Raw+0)* Resolution: *1*
second	Seconds (seconds) .
	unsigned short Minimum: *0* Maximum: *59* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} *(Raw+0)* Resolution: *1*
millisecond	Milliseconds (milliseconds) .
	unsigned long Minimum: *0* Maximum: *999* Scaling (A,B,C): *1, 0, 0* 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:	*yes*
Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*30*
Obsolete:	*no*
Size:	*2*
Programmatic Name:	*SNVT_press*
Neuron C Type:	signed long Minimum: *-32768* Maximum: *32767* Invalid: *32767* Scaling (A,B,C): *1, -1, 0* Scaled value: *1 *10 $^{-1}$ *(Raw+0)* Resolution: *0.1*
Formats:	*SNVT_press#SI: text("%f")* *SNVT_press#US: text("%f", *4.0217+0(0:954))* *SNVT_press#US_psi: text("%f", *0.1450+0(0:875))*
Used by:	*SCPTmaxPressureSetpoint SCPTmaxRemotePressureSetpoint SCPTminPressureSetpoint SCPTminRemotePressureSetpoint SCPTpumpCharacteristic SFPTgeneratorSet SFPTpressureSensor SFPTpumpController SFPTrefrigDisplayCaseControllerEvaporator SNVT_pumpset_sn*

SNVT_press_f

Overview:

Pressure (gauge) (Pascals) .

Details:

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

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) .
	privacyzone_t
number	Zone number (zone number) .
	unsigned short Minimum: *0* Maximum: *255*

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

	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.
	SNVT_freq_hz
body_temperature	Body temperature.
	SNVT_temp
motor_external_temperature	Motor external temp.
	SNVT_temp
motor_internal_temperature	Motor internal temp.
	SNVT_temp
motor_overloaded	Motor overloaded (boolean) .
	boolean_t
oil_level_low	Oil level low (boolean) .
	boolean_t
phase_imbalance_detected	Phase imbalance (boolean) .
	boolean_t
current_usage	Current usage.
	SNVT_amp

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

Formats: *SNVT_pump_sensor#SI: text("%f %f %f %f %m %m %m %f %f %m %m %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 %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: *yes*

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

Index: *156*

Obsolete: *no*

Size: *8*

Programmatic Name: *SNVT_pumpset_mn*

Neuron C Type:

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

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

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

SNVT_pwr_fact_f

Overview:

Power factor (multiplier) .

Details:

Standard:	*yes*

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: *no* Width: *3* Offset: *0* Minimum: *0* Maximum: *7* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} *(Raw+0)* Resolution: *1*
duplex_full	.
	bitfield Signed: *no* Width: *1*

	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	.
	rail_audio_type_t
addr_init	.

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	.
		rail_audio_sensor_type_t
addr_talk	.	
	Structure	
	unit_id	.

	bitfield Signed: *no* Width: *4* Offset: *0* Minimum: *0* Maximum: *15* 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 *1 *10 ^{0} *(Raw+0)*

		value: Resolution: *1*
	audio_sensor_type	.
		rail_audio_sensor_type_t
addr_dest	.	
	Union	
	p2p	.
		Structure
		unit_id .
		bitfield Signed: *no* Width: *4* Offset: *0* Minimum: *0* Maximum: *15* 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	.
	rail_audio_sensor_type_t

p2m	.

Structure	
mask_unit	.
	unsigned char Minimum: *0* Maximum: *255* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} *(Raw+0)* Resolution: *1*
mask_car	.
	unsigned char [4] Minimum: *0* Maximum: *255* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} *(Raw+0)* Resolution: *1*
mask_location	.
	unsigned char [2] Minimum: *0* Maximum: *255* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} *(Raw+0)*

	Resolution: *1*
mask_audio	.
	unsigned char [3] Minimum: *0* Maximum: *255* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} *(Raw+0)* 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: *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*
dest_p2p	.
	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*
reserved	.
	bitfield Signed: *no* Width: *6* Offset: *2* Minimum: *0* Maximum: *63* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} *(Raw+0)* Resolution: *1*
audio_type	.
	rail_audio_type_t
addr_init	.

Structure	
unit_id	.
	bitfield Signed: *no* Width: *4* Offset: *0* Minimum: *0*

		Maximum: *15* 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	.
		rail_audio_sensor_type_t
addr_dest	.	

Union	
p2p	.

Structure	
unit_id	.
	bitfield Signed: *no* Width: *4* Offset: *0* Minimum: *0* Maximum: *15* 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	.
	rail_audio_sensor_type_t

p2m	.

Structure	
mask_unit	.
	unsigned char Minimum: *0* Maximum: *255* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} *(Raw+0)* Resolution: *1*
mask_car	.
	unsigned char [4] Minimum: *0* Maximum: *255* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} *(Raw+0)* Resolution: *1*
mask_location	.
	unsigned char [2] Minimum: *0* Maximum: *255* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} *(Raw+0)* Resolution: *1*
mask_audio	.
	unsigned char [3] Minimum: *0* Maximum: *255* Scaling (A,B,C): *1, 0, 0*

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

Formats: *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])*

Used by: *SFPTrailcarAudioController SFPTrailcarAudioSensor*

SNVT_reg_val

Overview:

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

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 reg_val_unit_t enumeration is a fixed list and will not be modified nor extended due to its self-defining use in SNVT_reg_val, and SNVT_reg_val_ts.

EXAMPLES

raw	=	123456
unit	=	RVU_KWH
nr_decimals	=	2
Physical value	=	1234.56 kWh

Details:

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

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

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: *-2147483648* Maximum: *2147483647* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} *(Raw+0)* Resolution: *1*
unit	Unit code (unit names) .

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

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

SNVT_res_kilo

Overview:

Electric resistance (kiloOhms) .

Details:

Standard:	*yes*
Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*32*
Obsolete:	*no*
Size:	*2*
Programmatic Name:	*SNVT_res_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_res_kilo: text("%f")*

SNVT_rpm

Overview:

Angular velocity (revolutions/minute (RPM)) .

Details:

Standard:	*yes*
Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*102*
Obsolete:	*no*
Size:	*2*
Programmatic Name:	*SNVT_rpm*
Neuron C Type:	unsigned long Minimum: *0* Maximum: *65535* Invalid: *65535* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} *(Raw+0)* Resolution: *1*
Formats:	*SNVT_rpm: text("%f")*
Used by:	*SCPTnomRPM SCPTpumpCharacteristic SFPTgeneratorSet SFPTpumpController SNVT_clothes_w_c*

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:	*yes*
Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*180*
Obsolete:	*no*
Size:	*6*
Programmatic Name:	*SNVT_sblnd_state*

Neuron C Type:

Structure	
pos	.
	SNVT_setting

cmd_source	.
	sblnd_cmd_source_t
error_code	.
	sblnd_error_t

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) .
	scene_t
scene_number	Scene number.
	unsigned short Minimum: *1* Maximum: *255* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} *(Raw+0)* Resolution: *1*

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) .
	scene_config_t
scene_number	Scene number.
	unsigned short Minimum: *1* Maximum: *255* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} *(Raw+0)* Resolution: *1*
setting	Scene setting level (% of full level) .
	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*
rotation	Scene rotation angle (degrees) .
	signed long Minimum: *-17999* Maximum: *18000*

	Invalid: *32767* Scaling (A,B,C): *2, -2, 0* 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 (A,B,C): *1, -1, 0* 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 (A,B,C): *1, -1, 0* Scaled value: *1 *10 $^{-1}$ *(Raw+0)* Resolution: *0.1*
scene_priority	Scene priority (priority value) .
	unsigned short Minimum: *0* Maximum: *255* Scaling (A,B,C): *1, 0, 0* 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: *yes*

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

Index: *177*

Obsolete: *no*

Size: *1*

Programmatic Name: *SNVT_sched_val*

Neuron C Type: unsigned short

Minimum:	*0*
Maximum:	*255*
Invalid:	*255*
Scaling (A,B,C):	*1, 0, 0*
Scaled value:	*1 *10 ^{0} *(Raw+0)*
Resolution:	*1*

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

Used by: *SCPTscheduleTimeValue*

SNVT_sec_state

Overview:

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

Details:

Standard: *yes*

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

Index: *178*

Obsolete: *yes*

Size: *3*

Programmatic Name: *SNVT_sec_state*

Neuron C Type:

Structure	
state	.
	sec_state_t

identity	.
	unsigned long Minimum: *0* Maximum: *65535* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} *(Raw+0)* 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	.
	sec_status_t
identity	.
	unsigned long Minimum: *0* Maximum: *65535* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} *(Raw+0)* 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) .
	setting_t
setting	Scene setting level (% of full level) .
	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*
rotation	Rotation angle (degrees) .
	signed long Minimum: *-17999* Maximum: *18000* Invalid: *32767* Scaling (A,B,C): *2, -2, 0* Scaled value: $2 * 10^{-2} * (Raw+0)$ Resolution: *0.02*

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

SNVT_sound_db

Overview:

Sound level (dB) .

Details:

Standard:	*yes*
Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*33*

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

SNVT_sound_db_f

Overview:

Sound level (dBspl) .

Details:

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

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:	*yes*
Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*35*
Obsolete:	*no*
Size:	*2*
Programmatic Name:	*SNVT_speed_mil*
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_speed_mil#SI: text("%f")* *SNVT_speed_mil#US: text("%f", *3.2808+0(0:852))*
Used by:	*SFPTairVelocitySensor*

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:	*yes*
Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*83*
Obsolete:	*no*
Size:	*2*

Programmatic Name: *SNVT_state*

Neuron C Type:

Structure	
bit0	State bit 0 (boolean) .
	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*
bit1	State bit 1 (boolean) .
	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*
bit2	State bit 2 (boolean) .
	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*
bit3	State bit 3 (boolean) .
	bitfield Signed: *no* Width: *1* Offset: *3* Minimum: *0* Maximum: *1*

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

	Offset: *7* Minimum: *0* Maximum: *1* Scaling (A,B,C): *1, 0, 0* 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 (A,B,C): *1, 0, 0* 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 (A,B,C): *1, 0, 0* 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 (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} *(Raw+0)* Resolution: *1*
bit11	State bit 11 (boolean) .

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

	Resolution: *1*
bit15	State bit 15 (boolean) .
	bitfield Signed: *no* Width: *1* Offset: *7* Minimum: *0* Maximum: *1* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} *(Raw+0)* Resolution: *1*

Formats: *SNVT_state: text("%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|%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: *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*
bit1	State bit 1 (boolean) .
	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*
bit2	State bit 2 (boolean) .
	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*
bit3	State bit 3 (boolean) .
	bitfield Signed: *no* Width: *1* Offset: *3* Minimum: *0* Maximum: *1* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} *(Raw+0)*

	Resolution: *1*
bit4	State bit 4 (boolean) .
	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*
bit5	State bit 5 (boolean) .
	bitfield Signed: *no* Width: *1* Offset: *5* Minimum: *0* Maximum: *1* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} *(Raw+0)* Resolution: *1*
bit6	State bit 6 (boolean) .
	bitfield Signed: *no* Width: *1* Offset: *6* Minimum: *0* Maximum: *1* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} *(Raw+0)* Resolution: *1*
bit7	State bit 7 (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*
bit8	State bit 8 (boolean) .
	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*
bit9	State bit 9 (boolean) .
	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*
bit10	State bit 10 (boolean) .
	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*
bit11	State bit 11 (boolean) .
	bitfield Signed: *no* Width: *1* Offset: *3*

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

	Signed: *no* Width: *1* Offset: *7* Minimum: *0* Maximum: *1* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} *(Raw+0)* Resolution: *1*
bit16	State bit 16 (boolean) .
	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*
bit17	State bit 17 (boolean) .
	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*
bit18	State bit 18 (boolean) .
	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*

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

	value: *1 *10 ^{0} *(Raw+0)* Resolution: *1*
bit23	State bit 23 (boolean) .
	bitfield Signed: *no* Width: *1* Offset: *7* Minimum: *0* Maximum: *1* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} *(Raw+0)* Resolution: *1*
bit24	State bit 24 (boolean) .
	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*
bit25	State bit 25 (boolean) .
	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*
bit26	State bit 26 (boolean) .
	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*
bit27	State bit 27 (boolean) .
	bitfield Signed: *no* Width: *1* Offset: *3* Minimum: *0* Maximum: *1* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} *(Raw+0)* Resolution: *1*
bit28	State bit 28 (boolean) .
	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*
bit29	State bit 29 (boolean) .
	bitfield Signed: *no* Width: *1* Offset: *5* Minimum: *0* Maximum: *1* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} *(Raw+0)* Resolution: *1*
bit30	State bit 30 (boolean) .
	bitfield Signed: *no* Width: *1*

	Offset: *6* Minimum: *0* Maximum: *1* Scaling (A,B,C): *1, 0, 0* 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 (A,B,C): *1, 0, 0* 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 (A,B,C): *1, 0, 0* 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 (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} *(Raw+0)* Resolution: *1*
bit34	State bit 34 (boolean) .

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

	Resolution: *1*
bit38	State bit 38 (boolean) .
	bitfield Signed: *no* Width: *1* Offset: *6* Minimum: *0* Maximum: *1* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} *(Raw+0)* Resolution: *1*
bit39	State bit 39 (boolean) .
	bitfield Signed: *no* Width: *1* Offset: *7* Minimum: *0* Maximum: *1* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} *(Raw+0)* Resolution: *1*
bit40	State bit 40 (boolean) .
	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*
bit41	State bit 41 (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*
bit42	State bit 42 (boolean) .
	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*
bit43	State bit 43 (boolean) .
	bitfield Signed: *no* Width: *1* Offset: *3* Minimum: *0* Maximum: *1* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} *(Raw+0)* Resolution: *1*
bit44	State bit 44 (boolean) .
	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*
bit45	State bit 45 (boolean) .
	bitfield Signed: *no* Width: *1* Offset: *5*

	Minimum: *0* Maximum: *1* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} *(Raw+0)* Resolution: *1*
bit46	State bit 46 (boolean) .
	bitfield Signed: *no* Width: *1* Offset: *6* Minimum: *0* Maximum: *1* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} *(Raw+0)* Resolution: *1*
bit47	State bit 47 (boolean) .
	bitfield Signed: *no* Width: *1* Offset: *7* Minimum: *0* Maximum: *1* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} *(Raw+0)* Resolution: *1*
bit48	State bit 48 (boolean) .
	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*
bit49	State bit 49 (boolean) .
	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*
bit50	State bit 50 (boolean) .
	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*
bit51	State bit 51 (boolean) .
	bitfield Signed: *no* Width: *1* Offset: *3* Minimum: *0* Maximum: *1* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} *(Raw+0)* Resolution: *1*
bit52	State bit 52 (boolean) .
	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*

bit53	State bit 53 (boolean) .
	bitfield Signed: *no* Width: *1* Offset: *5* Minimum: *0* Maximum: *1* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} *(Raw+0)* Resolution: *1*
bit54	State bit 54 (boolean) .
	bitfield Signed: *no* Width: *1* Offset: *6* Minimum: *0* Maximum: *1* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} *(Raw+0)* Resolution: *1*
bit55	State bit 55 (boolean) .
	bitfield Signed: *no* Width: *1* Offset: *7* Minimum: *0* Maximum: *1* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} *(Raw+0)* Resolution: *1*
bit56	State bit 56 (boolean) .
	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*
bit57	State bit 57 (boolean) .
	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*
bit58	State bit 58 (boolean) .
	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*
bit59	State bit 59 (boolean) .
	bitfield Signed: *no* Width: *1* Offset: *3* Minimum: *0* Maximum: *1* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} *(Raw+0)* Resolution: *1*
bit60	State bit 60 (boolean) .
	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*
bit61	State bit 61 (boolean) .
	bitfield Signed: *no* Width: *1* Offset: *5* Minimum: *0* Maximum: *1* Scaling (A,B,C): *1, 0, 0* Scaled value: $1 *10^{0} *(Raw+0)$ Resolution: *1*
bit62	State bit 62 (boolean) .
	bitfield Signed: *no* Width: *1* Offset: *6* Minimum: *0* Maximum: *1* Scaling (A,B,C): *1, 0, 0* Scaled value: $1 *10^{0} *(Raw+0)$ Resolution: *1*
bit63	State bit 63 (boolean) .
	bitfield Signed: *no* Width: *1* Offset: *7* Minimum: *0* Maximum: *1* Scaling (A,B,C): *1, 0, 0* Scaled value: $1 *10^{0} *(Raw+0)$ Resolution: *1*

Formats: *SNVT_state_64: 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 %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 %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, bit16, bit17, bit18, bit19, bit20, bit21, bit22, bit23, bit24, bit25, bit26, bit27,*

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

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

Used by: *SCPTactuatorType SCPTdevListDesc SCPTdialString SCPTfireTxt1 SCPTfireTxt2 SCPTfireTxt3 SCPTifaceDesc SCPTlocation SCPToemType SCPTpartNumber SCPTregName SCPTserialNumber SFPTchiller SFPTelevatorPositionIndicator 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 (A,B,C): *1, 0, 0* 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 (A,B,C): *1, 0, 0* 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", 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", 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: *0* Maximum: *200* Scaling (A,B,C): *5, -1, 0* Scaled value: *5 *10 $^{-1}$ *(Raw+0)* Resolution: *0.5*
state	State (state code) . This field can either be -1 (NULL), 0 (OFF), or 1 (ON).
	signed short Minimum: *-1* Maximum: *1* Invalid: *-1* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} *(Raw+0)* Resolution: *1*

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 SFPTelevatorArrivalGong SFPTelevatorDirectionLantern SFPTelevatorFireSystemsPort SFPTelevatorHallLantern SFPTelevatorPositionIndicator SFPTelevatorVoiceAnnouncer SFPTentryExit SFPTfanCoilUnit SFPTfrostSensor SFPTgeneratorSet SFPTheatPump SFPThvacValvePositioner SFPTlampActuator SFPTmodemController SFPTnodeObject SFPToccupancyController SFPTpartitionWallController SFPTpullStationFireInitiator SFPTpumpController SFPTrainSensor 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
	switch_state_t
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: *0* Maximum: *200* Invalid: *255* Scaling (A,B,C): *5, -1, 0* Scaled *5 *10 $^{-1}$ *(Raw+0)*

	value: Resolution: *0.5*
change	Percent change.. Percent change to level
	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*
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 (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} *(Raw+0)* Resolution: *1*
group_number	Group number. Group number that is enabled or disabled by the SW_ENABLE_GROUP and SW_DISABLE_GROUIP states in the state field; if 0, all groups are enabled or disabled
	unsigned short Minimum: *0* Maximum: *63* Invalid: *255* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} *(Raw+0)* Resolution: *1*
multiplier	Factor (percent) . Multiplier for the level
	SNVT_multiplier_s
angle	Rotation angle (degrees) . Rotation angle for devices that support a rotation setting such as blinds.

		signed short Minimum: *-90* Maximum: *90* Invalid: *-128* Scaling (A,B,C): *2, 0, 0* Scaled value: *2 *10 ^{0} *(Raw+0)* Resolution: *2*
	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: *-100* Maximum: *100* Invalid: *-128* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} *(Raw+0)* Resolution: *1*
scene_number	Scene number. Scene number that is applied based on the function specified in the state field.	
	unsigned short Minimum: *1* Maximum: *255* Invalid: *0* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} *(Raw+0)* Resolution: *1*	

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 -274ºC. 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 SFPThvacTempSensor 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 ºC to ºF, if needed. This conversion requires a 32ºF 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:	*yes*
Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*147*
Obsolete:	*no*
Size:	*2*
Programmatic Name:	*SNVT_temp_diff_p*
Neuron C Type:	signed long 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_temp_diff_p#SI: text("%f", *1+0(0:854))* *SNVT_temp_diff_p#US: text("%f", *1.8+0(0:855))*
Used by:	*SCPTdefrostDetect SCPTdiffTempSetpoint SCPTtemperatureHysteresis SCPTtempOffset*

SNVT_temp_f

Overview:

Temperature (degrees Celsius) .

Details:

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

SNVT_temp_p

Overview:

Temperature (degrees Celsius) .

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

Details:

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

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

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 (A,B,C): *1, -2, 0* 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 (A,B,C): *1, -2, 0* Scaled $1 * 10^{-2} * (Raw+0)$

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

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

SNVT_time_f

Overview:

Elapsed time (seconds) .

Details:

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

SNVT_time_hour

Overview:

Elapsed time (hours) .

Details:

Standard:	*yes*
Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*124*
Obsolete:	*no*
Size:	*2*
Programmatic Name:	*SNVT_time_hour*
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_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 *10 ^{0} *(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 (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} *(Raw+0)* Resolution: *1*
minutes	Minutes (minutes) .
	unsigned short Minimum: *0* Maximum: *59* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} *(Raw+0)* Resolution: *1*
seconds	Seconds (seconds) .
	unsigned short Minimum: *0* Maximum: *59* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} *(Raw+0)* Resolution: *1*
milliseconds	Milliseconds (milliseconds) .
	unsigned short Minimum: *0* Maximum: *99* Scaling (A,B,C): *1, 1, 0* Scaled value: *1 *10 ^{1} *(Raw+0)* 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 SCPTrampDownTm SCPTrampUpTm 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: *-1* Maximum: *3000* Invalid: *-1* Scaling (A,B,C): *1, 0, 0* 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* 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 (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} *(Raw+0)* Resolution: *1*
hundredths	Hundredths of a second. (seconds) . Hundredths portion of a the timestamp.
	unsigned short Minimum: *0* Maximum: *99* Invalid: *255* Scaling (A,B,C): *1, -2, 0* Scaled value: *1 *10 $^{-2}$ *(Raw+0)* 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. Ma
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 Name: *SNVT_time_zone*

Neuron C Type:

Structure	
second_time_offset	Offset from GMT (seconds) . West direction is negative offset
	s32_type Minimum: *-86400* Maximum: *86400* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} *(Raw+0)* Resolution: *1*
type_of_description	Calendar type (calendar type names) .
	calendar_type_t
hour_of_start_DST	DST start 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_start_DST	DST start 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_start_DST	DST start second (seconds) .
	unsigned short Minimum: *0* Maximum: *59* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} *(Raw+0)*

	Resolution: *1*
start_DST	DST start day (day descriptor) . Daylight savings time start da

Union	
G_day_of_start_DST	Gregorian calendar day of start DST (da
	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_start_DST	Julian calendar day of start DST (days)
	unsigned long Minimum: *1* Maximum: *365* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} *(Raw+0)* Resolution: *1*
M_start_DST	Meu calendar day of start DST (month, dateday) .

Structure	
month_of_start_DST	Month of start D (months) .
	bitfield Signed: *no* Width: *4* Offset: *0* Minimum: *1* Maximum: *12* Scaling (A,B,C): *1, (* Scaled value: *1 * *(Ra* Resolution: *1*
week_of_start_DST	Week of start DS (weeks) .
	bitfield Signed: *no* Width: *3* Offset: *4*

			Minimum: *1* Maximum: *5* Scaling (A,B,C): *1,* Scaled value: *1 * (Ra* Resolution: *1*
		dateday_of_start_DST	Day of week (d names) .
			*days_of_week_*
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	
		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 (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} *(Raw+0)* Resolution: *1*
M_end_DST	Meu calendar day of end DST (month, w dateday) .
	Structure
	month_of_end_DST: Month of end DS (months) .
	bitfield Signed: *no* Width: *4* Offset: *0* Minimum: *1* Maximum: *12* Scaling (A,B,C): *1, 0,* Scaled value: *1 *1((Raw* Resolution: *1*
	week_of_end_DST: Week of end DST (weeks) .
	bitfield Signed: *no* Width: *3* Offset: *4* Minimum: *1* Maximum: *5* Scaling (A,B,C): *1, 0,* Scaled value: *1 *1((Raw* Resolution: *1*
	dateday_of_end_DST: Day of week (da 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 (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} *(Raw+0)* 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: *yes*

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

Index: *96*

Obsolete: *no*

Size: *30*

Programmatic Name: *SNVT_trans_table*

Neuron C Type:

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

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

	Signed: *no* Width: *2* Offset: *0* Minimum: *0* Maximum: *1* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} *(Raw+0)* Resolution: *1*
interp_pts_5_to_6	Interpolation for point 5 to point 6 (interpolation method code) .
	bitfield Signed: *no* Width: *2* Offset: *2* Minimum: *0* Maximum: *1* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} *(Raw+0)* Resolution: *1*
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: *no* Width: *2* Offset: *4* Minimum: *0* Maximum: *1* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} *(Raw+0)* Resolution: *1*

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

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:	*yes*

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

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

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:	*yes*
Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*42*
Obsolete:	*no*
Size:	*2*
Programmatic Name:	*SNVT_vol_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_vol_kilo#SI: text("%f")* *SNVT_vol_kilo#US: text("%f", *35.315+0(0:877))* *SNVT_vol_kilo#US_gal: text("%f", *264.179+0(0:858))*

SNVT_vol_mil

Overview:

Volume (milliliters) .

Details:

Standard:	*yes*
Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*43*
Obsolete:	*no*
Size:	*2*
Programmatic Name:	*SNVT_vol_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_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:	*yes*
Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*138*
Obsolete:	*no*
Size:	*2*
Programmatic Name:	*SNVT_volt_ac*
Neuron C Type:	unsigned long Minimum: *0* Maximum: *65535* Invalid: *65535* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} *(Raw+0)* Resolution: *1*
Formats:	*SNVT_volt_ac: text("%f")*
Used by:	*SFPTautomaticTransferSwitch SFPTgeneratorSet*

SNVT_volt_dbmv

Overview:

Electric voltage (dB microVolts) .

Details:

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

SNVT_volt_f

Overview:

Electric voltage (Volts) .

Details:

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

SNVT_volt_kilo

Overview:

Electric voltage (kiloVolts) .

Details:

Standard:	*yes*
Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*46*
Obsolete:	*no*
Size:	*2*
Programmatic Name:	*SNVT_volt_kilo*
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_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 (A,B,C): *5, -3, 0* Scaled value: *5 *10 $^{-3}$ *(Raw+0)* Resolution: *0.005*
span	Span-factor (16-bit unsigned value) .
	unsigned long Minimum: *0* Maximum: *65535* Scaling (A,B,C): *5, -4, 0* Scaled value: *5 *10 $^{-4}$ *(Raw+0)* Resolution: *0.0005*

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

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