

ezManager Library

사용자 설명서

버전 1.1

솔내시스템(주)



솔내시스템(주)

<https://www.sollae.co.kr>

목차

1	소개.....	- 8 -
1.1	소개.....	- 8 -
1.2	표기법	- 8 -
1.3	주의사항	- 9 -
2	라이브러리 시작과 종료	- 10 -
2.1	Start_Library.....	- 10 -
2.2	End_Library.....	- 10 -
3	제품(EZTCP) 검색	- 12 -
3.1	EzTCP_Search.....	- 12 -
3.2	EzTCP_Read	- 13 -
3.3	get_eztcp_count.....	- 16 -
3.4	get_eztcp_info_by_index.....	- 17 -
4	설정 값 읽기.....	- 20 -
4.1	네트워크	- 20 -
4.1.1	get_arp.....	- 20 -
4.1.2	get_dhcp.....	- 20 -
4.1.3	get_dhcp_dns.....	- 21 -
4.1.4	get_pppoe.....	- 22 -
4.1.5	get_ip4_local_address.....	- 22 -
4.1.6	get_ip4_subnet_mask.....	- 23 -
4.1.7	get_ip4_gateway_address.....	- 24 -
4.1.8	get_ip4_dns_address.....	- 25 -
4.1.9	get_pppoe_id.....	- 26 -
4.1.10	get_pppoe_pwd.....	- 27 -
4.1.11	get_ip6.....	- 28 -
4.1.12	get_ip6_gua_type.....	- 28 -
4.1.13	get_ip6_eui_type.....	- 29 -
4.1.14	get_ip6_local_address.....	- 31 -
4.1.15	get_ip6_subnet_prefix_length.....	- 32 -
4.1.16	get_ip6_gateway_address.....	- 33 -
4.1.17	get_ip6_dns_address.....	- 34 -
4.2	시리얼 포트	- 35 -
4.2.1	get_uart_count.....	- 35 -
4.2.2	get_uart_min_baudrate	- 35 -
4.2.3	get_uart_max_baudrate.....	- 36 -
4.2.4	get_uart_serial_type	- 37 -
4.2.5	get_uart_ttl	- 38 -
4.2.6	get_uart_baudrate	- 38 -
4.2.7	get_uart_parity	- 39 -
4.2.8	get_uart_databit	- 40 -
4.2.9	get_uart_stopbit.....	- 41 -
4.2.10	get_uart_flow_control.....	- 42 -
4.2.11	get_uart_dtrdsr	- 43 -
4.2.12	get_uart_tx_delay.....	- 43 -
4.3	TCP/IP	- 44 -
4.3.1	get_uart_communication_mode.....	- 44 -
4.3.2	get_uart_peer_address	- 45 -

4.3.3	get_uart_peer_port.....	- 46 -
4.3.4	get_uart_local_port.....	- 47 -
4.3.5	get_uart_cod_tcp_server.....	- 48 -
4.3.6	get_uart_watermark.....	- 49 -
4.3.7	get_uart_timeout.....	- 51 -
4.3.8	get_uart_data_frame_interval.....	- 52 -
4.3.9	get_uart_separator_length.....	- 53 -
4.3.10	get_uart_separator_type.....	- 54 -
4.3.11	get_uart_separator.....	- 54 -
4.3.12	get_uart_tcp_nodelay.....	- 56 -
4.3.13	get_uart_rfc2217.....	- 57 -
4.3.14	get_uart_protocol.....	- 57 -
4.4	CSC-HR2.....	- 58 -
4.4.1	get_csc_hr2_communication_mode.....	- 58 -
4.4.2	get_csc_hr2_id.....	- 59 -
4.4.3	get_csc_hr2_network_timeout.....	- 60 -
4.4.4	get_csc_hr2_network_threshold.....	- 61 -
4.4.5	get_csc_hr2_server_timeout.....	- 62 -
4.4.6	get_csc_hr2_server_threshold.....	- 63 -
4.4.7	get_csc_hr2_check_port.....	- 64 -
4.4.8	get_csc_hr2_first_server_address.....	- 65 -
4.4.9	get_csc_hr2_first_server_port.....	- 66 -
4.4.10	get_csc_hr2_second_server_address.....	- 67 -
4.4.11	get_csc_hr2_second_server_port.....	- 68 -
4.5	I/O 제품.....	- 69 -
4.5.1	get_io_http.....	- 69 -
4.5.2	get_io_http_port.....	- 70 -
4.5.3	get_io_html_size.....	- 71 -
4.5.4	get_io_modbus.....	- 72 -
4.5.5	get_io_input_notification.....	- 73 -
4.5.6	get_io_output_automatic_initialize.....	- 74 -
4.5.7	get_io_modbus_type.....	- 74 -
4.5.8	get_io_unit_id.....	- 75 -
4.5.9	get_io_input_address.....	- 76 -
4.5.10	get_io_output_address.....	- 77 -
4.5.11	get_io_poll_interval.....	- 78 -
4.5.12	get_io_slave_control_type.....	- 79 -
4.5.13	get_io_master_control_type.....	- 81 -
4.5.14	get_io_tcp_type.....	- 82 -
4.5.15	get_io_multi_connection_number.....	- 83 -
4.5.16	get_io_modbus_peer_address.....	- 84 -
4.5.17	get_io_modbus_peer_port.....	- 85 -
4.5.18	get_io_modbus_local_port.....	- 86 -
4.5.19	get_io_output_number.....	- 87 -
4.5.20	get_io_input_number.....	- 88 -
4.5.21	get_io_event_notification.....	- 88 -
4.5.22	get_event_notification_email.....	- 89 -
4.5.23	get_io_event_notification_port.....	- 90 -
4.5.24	get_io_input_valid_time.....	- 92 -
4.5.25	get_io_macro_type.....	- 93 -
4.5.26	get_io_macro.....	- 94 -
4.5.27	get_io_port_macro.....	- 95 -
4.5.28	get_io_macro_text.....	- 96 -
4.5.29	get_io_output_delay.....	- 98 -
4.5.30	get_io_output_initial_state.....	- 99 -
4.5.31	get_io_comment.....	- 100 -
4.6	무선랜.....	- 102 -

4.6.1	get_wlan_type.....	- 102 -
4.6.2	get_wlan_channel.....	- 103 -
4.6.3	get_wlan_ssid.....	- 104 -
4.6.4	get_wlan_antenna.....	- 105 -
4.6.5	get_wlan_phy_mode.....	- 106 -
4.6.6	get_wlan_short_preamble.....	- 108 -
4.6.7	get_wlan_short_slot.....	- 109 -
4.6.8	get_wlan_cts_protection.....	- 110 -
4.6.9	get_wlan_background_scan.....	- 111 -
4.6.10	get_wlan_encryption_type.....	- 112 -
4.6.11	get_wlan_authentication_type.....	- 114 -
4.6.12	get_wlan_wep_key_length.....	- 116 -
4.6.13	get_wlan_wep_key_index.....	- 117 -
4.6.14	get_wlan_wep_key_data_type.....	- 118 -
4.6.15	get_wlan_wep_key.....	- 119 -
4.6.16	get_wlan_max_wpa_passphrase_length.....	- 121 -
4.6.17	get_wlan_wpa_passphrase.....	- 121 -
4.6.18	get_wlan_shared_key.....	- 123 -
4.6.19	get_wlan_wpa_enterprise.....	- 124 -
4.6.20	get_wlan_wpa_enterprise_id.....	- 125 -
4.6.21	get_wlan_wpa_enterprise_pwd.....	- 127 -
4.7	부가기능.....	- 128 -
4.7.1	get_telnet.....	- 128 -
4.7.2	get_send_mac_address.....	- 129 -
4.7.3	get_ssl.....	- 129 -
4.7.4	get_ssh.....	- 130 -
4.7.5	get_ip4_address_search.....	- 131 -
4.7.6	get_remote_debug.....	- 131 -
4.7.7	get_tcp_multi_connection.....	- 132 -
4.7.8	get_power_management.....	- 133 -
4.7.9	get_comment.....	- 133 -
4.7.10	get_allowed_ezmanager.....	- 134 -
4.7.11	get_allowed_mac_address.....	- 135 -
4.7.12	get_allowed_ip4_address.....	- 136 -
4.7.13	get_allowed_ip4_network_mask_type.....	- 138 -
4.7.14	get_allowed_ip6_address.....	- 140 -
4.7.15	get_allowed_ip6_subnet_prefix_length.....	- 141 -
4.7.16	get_ip4_change_notification_type.....	- 142 -
4.7.17	get_ip4_change_notification_data_type.....	- 144 -
4.7.18	get_ip4_change_notification_interval.....	- 145 -
4.7.19	get_ip4_change_notification_peer_port.....	- 146 -
4.7.20	get_ip4_change_notification_peer_address.....	- 147 -
4.7.21	get_ip4_change_notification_ddns_id.....	- 149 -
4.7.22	get_ip4_change_notification_ddns_pwd.....	- 150 -
5	설정 값 변경.....	- 152 -
5.1	네트워크.....	- 152 -
5.1.1	set_arp.....	- 152 -
5.1.2	set_dhcp.....	- 152 -
5.1.3	set_dhcp_dns.....	- 153 -
5.1.4	set_pppoe.....	- 154 -
5.1.5	set_ip4_local_address.....	- 155 -
5.1.6	set_ip4_subnet_mask.....	- 156 -
5.1.7	set_ip4_gateway_address.....	- 157 -
5.1.8	set_ip4_dns_address.....	- 158 -
5.1.9	set_pppoe_id.....	- 159 -
5.1.10	set_pppoe_pwd.....	- 160 -

5.1.11	set_ip6.....	- 160 -
5.1.12	set_ip6_gua_type.....	- 161 -
5.1.13	set_ip6_eui_type.....	- 162 -
5.1.14	set_ip6_local_address.....	- 163 -
5.1.15	set_ip6_subnet_prefix_length.....	- 164 -
5.1.16	set_ip6_gateway_address.....	- 165 -
5.1.17	set_ip6_dns_address.....	- 166 -
5.2	시리얼 포트.....	- 167 -
5.2.1	set_uart_serial_type.....	- 167 -
5.2.2	set_uart_ttl.....	- 168 -
5.2.3	set_uart_baudrate.....	- 169 -
5.2.4	set_uart_parity.....	- 170 -
5.2.5	set_uart_databit.....	- 171 -
5.2.6	set_uart_stopbit.....	- 172 -
5.2.7	set_uart_flow_control.....	- 173 -
5.2.8	set_uart_dtrdsr.....	- 175 -
5.2.9	set_uart_tx_delay.....	- 176 -
5.3	TCP/IP.....	- 177 -
5.3.1	set_uart_communication_mode.....	- 177 -
5.3.2	set_uart_peer_address.....	- 178 -
5.3.3	set_uart_peer_port.....	- 180 -
5.3.4	set_uart_local_port.....	- 180 -
5.3.5	set_uart_cod_tcp_server.....	- 182 -
5.3.6	set_uart_watermark.....	- 183 -
5.3.7	set_uart_timeout.....	- 184 -
5.3.8	set_uart_data_frame_interval.....	- 185 -
5.3.9	set_uart_separator_length.....	- 186 -
5.3.10	set_uart_separator_type.....	- 187 -
5.3.11	set_uart_separator.....	- 188 -
5.3.12	set_uart_tcp_nodelay.....	- 189 -
5.3.13	set_uart_rfc2217.....	- 190 -
5.3.14	set_uart_protocol.....	- 192 -
5.4	CSC-HR2.....	- 193 -
5.4.1	set_csc_hr2_communication_mode.....	- 193 -
5.4.2	set_csc_hr2_id.....	- 194 -
5.4.3	set_csc_hr2_network_timeout.....	- 195 -
5.4.4	set_csc_hr2_network_threshold.....	- 195 -
5.4.5	set_csc_hr2_server_timeout.....	- 196 -
5.4.6	set_csc_hr2_server_threshold.....	- 197 -
5.4.7	set_csc_hr2_checkport.....	- 198 -
5.4.8	set_csc_hr2_first_server_address.....	- 199 -
5.4.9	set_csc_hr2_first_server_port.....	- 200 -
5.4.10	set_csc_hr2_second_server_address.....	- 201 -
5.4.11	set_csc_hr2_second_server_port.....	- 201 -
5.5	I/O 제품.....	- 202 -
5.5.1	set_io_http.....	- 202 -
5.5.2	set_io_http_port.....	- 203 -
5.5.3	set_io_html_size.....	- 204 -
5.5.4	set_io_modbus.....	- 205 -
5.5.5	set_io_input_notification.....	- 206 -
5.5.6	set_io_output_automatic_initialize.....	- 207 -
5.5.7	set_io_modbus_type.....	- 208 -
5.5.8	set_io_unit_id.....	- 209 -
5.5.9	set_io_input_address.....	- 210 -
5.5.10	set_io_output_address.....	- 211 -
5.5.11	set_io_poll_interval.....	- 212 -
5.5.12	set_io_slave_control_type.....	- 213 -

5.5.13	set_io_master_control_type	- 214 -
5.5.14	set_io_tcp_type	- 215 -
5.5.15	set_io_multi_connection_number	- 216 -
5.5.16	set_io_modbus_peer_address	- 217 -
5.5.17	set_io_modbus_peer_port	- 218 -
5.5.18	set_io_modbus_local_port	- 219 -
5.5.19	set_io_event_notification	- 220 -
5.5.20	set_event_notification_email	- 221 -
5.5.21	set_io_event_notification_port	- 222 -
5.5.22	set_io_input_valid_time	- 223 -
5.5.23	set_io_macro	- 224 -
5.5.24	set_io_port_macro	- 225 -
5.5.25	set_io_macro_text	- 226 -
5.5.26	set_io_output_delay	- 227 -
5.5.27	set_io_output_initial_state	- 228 -
5.5.28	set_io_comment	- 229 -
5.6	무선랜	- 230 -
5.6.1	set_wlan_type	- 230 -
5.6.2	set_wlan_channel	- 232 -
5.6.3	set_wlan_ssid	- 232 -
5.6.4	set_wlan_antenna	- 233 -
5.6.5	set_wlan_phy_mode	- 234 -
5.6.6	set_wlan_short_preamble	- 236 -
5.6.7	set_wlan_short_slot	- 237 -
5.6.8	set_wlan_cts_protection	- 238 -
5.6.9	set_wlan_background_scan	- 239 -
5.6.10	set_wlan_encryption_type	- 240 -
5.6.11	set_wlan_authentication_type	- 241 -
5.6.12	set_wlan_wep_key_length	- 243 -
5.6.13	set_wlan_wep_key_index	- 244 -
5.6.14	set_wlan_wep_key_data_type	- 245 -
5.6.15	set_wlan_wep_key	- 246 -
5.6.16	set_wlan_wpa_passphrase	- 247 -
5.6.17	set_wlan_shared_key	- 248 -
5.6.18	set_wlan_wpa_enterprise	- 249 -
5.6.19	set_wlan_wpa_enterprise_id	- 251 -
5.6.20	set_wlan_wpa_enterprise_pwd	- 252 -
5.7	부가기능	- 253 -
5.7.1	set_telnet	- 253 -
5.7.2	set_send_mac_address	- 253 -
5.7.3	set_ssl	- 254 -
5.7.4	set_ssh	- 256 -
5.7.5	set_ip4_address_search	- 257 -
5.7.6	set_remote_debug	- 258 -
5.7.7	set_tcp_multi_connection	- 259 -
5.7.8	set_power_management	- 260 -
5.7.9	set_comment	- 261 -
5.7.10	set_allowed_mac_address	- 261 -
5.7.11	set_allowed_ip4_address	- 262 -
5.7.12	set_allowed_ip4_network_mask_type	- 263 -
5.7.13	set_allowed_ezmanager	- 264 -
5.7.14	set_allowed_ip6_address	- 265 -
5.7.15	set_allowed_ip6_subnet_prefix_length	- 266 -
5.7.16	set_ip4_change_notification_type	- 267 -
5.7.17	set_ip4_change_notification_data_type	- 268 -
5.7.18	set_ip4_change_notification_interval	- 269 -
5.7.19	set_ip4_change_notification_peer_port	- 270 -

5.7.20	<i>set_ip4_change_notification_peer_address</i>	- 270 -
5.7.21	<i>set_ip4_change_notification_ddns_id</i>	- 271 -
5.7.22	<i>set_ip4_change_notification_ddns_pwd</i>	- 272 -
6	설정 값 저장	- 274 -
6.1	EzTCP_Write.....	- 274 -
6.2	EzTCP_Initialize.....	- 277 -
7	비밀번호 관리	- 280 -
7.1	EzTCP_ChangePassword	- 280 -
8	현재상태 보기	- 285 -
8.1	EzTCP_Status	- 285 -
8.2	get_status_string_length	- 286 -
8.3	get_status_string	- 286 -
8.4	get_tcpip4_session_count	- 288 -
8.5	get_tcpip6_session_count	- 289 -
8.6	get_tcpip4_session_info.....	- 291 -
8.7	get_tcpip6_session_info.....	- 293 -
8.8	EzTCP_CloseTcpIp4	- 294 -
8.9	EzTCP_CloseTcpIp6	- 297 -
9	인증서	- 300 -
9.1	EzTCP_ReadCert.....	- 300 -
9.2	get_certificates_string_length	- 301 -
9.3	get_certificates_string	- 301 -
9.4	EzTCP_WriteSelfSignedCertificates	- 303 -
9.5	EzTCP_WriteCertificates	- 305 -
10	기타	- 307 -
10.1	get_version.....	- 307 -
10.2	is_ip6	- 308 -
10.3	is_uart_rs232.....	- 308 -
10.4	is_uart_rs485.....	- 309 -
10.5	is_uart_rs422.....	- 310 -
10.6	is_uart_ttl.....	- 310 -
10.7	is_uart_8_databit_only.....	- 311 -
10.8	is_uart_7_and_8_databit_only	- 311 -
10.9	is_uart_one5_stopbit.....	- 312 -
10.10	is_uart_dtrdsr.....	- 313 -
10.11	is_uart_xonxoff.....	- 313 -
10.12	is_uart_tx_delay.....	- 314 -
10.13	is_uart_data_frame_interval	- 314 -
10.14	is_uart_separator.....	- 315 -
10.15	is_uart_tcp_nodelay	- 316 -
10.16	is_uart_rfc2217	- 316 -
10.17	is_uart_protocol	- 317 -
10.18	is_uart_tcp_server.....	- 318 -
10.19	is_uart_at_command.....	- 318 -
10.20	is_uart_tcp_client.....	- 319 -
10.21	is_uart_udp	- 320 -
10.22	is_uart_serial_modbus	- 320 -
10.23	is_csc_hr2	- 321 -
10.24	is_io	- 322 -
10.25	is_io_output_automatic_initialize	- 322 -

10.26	is_event_notification.....	- 323 -
10.27	is_wlan.....	- 323 -
10.28	is_wlan_soft_ap.....	- 324 -
10.29	is_wlan_antenna.....	- 325 -
10.30	is_wlan_phy_mode.....	- 325 -
10.31	is_wlan_background_scan.....	- 326 -
10.32	is_wlan_rsn.....	- 326 -
10.33	is_wlan_authentication_type.....	- 327 -
10.34	is_wlan_wpa_enterprise.....	- 328 -
10.35	is_send_mac_address.....	- 328 -
10.36	is_ssl.....	- 329 -
10.37	is_ssh.....	- 330 -
10.38	is_remote_debug.....	- 330 -
10.39	is_tcp_multi_connection.....	- 331 -
10.40	is_power_management.....	- 331 -
10.41	is_password.....	- 332 -
10.42	is_certificates.....	- 333 -
11	면책 고지 사항.....	- 334 -
11.1	면책 고지 사항.....	- 334 -
12	문서 변경 이력.....	- 335 -

1 소개

1.1 소개

- 제품(ezTCP) 설정 프로그램인 ezManager가 제공하는 주요 기능을 라이브러리 형태로 제공합니다. 따라서 이 라이브러리로 ezManager와 같은 프로그램을 만들 수 있습니다.
- ezManager 라이브러리는 정적 라이브러리(static library)와 동적 라이브러리(dynamic-link library) 형태로 제공합니다.

1.2 표기법

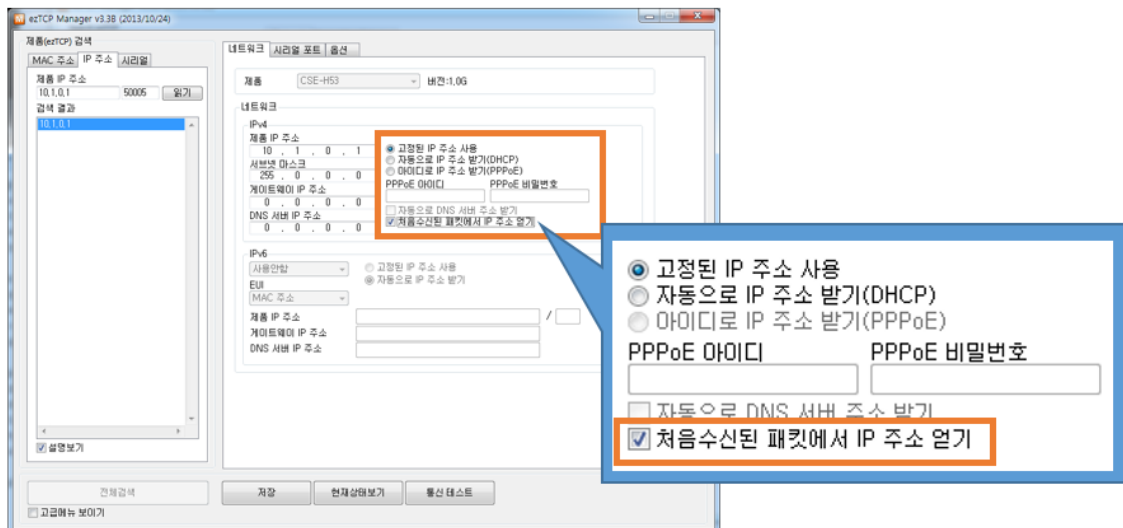
- 본 문서 전체에서 사용한 '['과 ']'으로 감싸진 굵은 글씨를 사용한 문구는 ezManager 프로그램에서 사용되는 용어들입니다.

4.1.1 get_arp

```
int get_arp(unsigned char *mac_address);
```

Description

get_arp는 **[처음 수신된 패킷에서 IP 주소 얻기]** 기능의 선택 여부를 반환합니다.



1.3 주의사항

- 비정상적인 라이브러리 사용으로 인한 피해는 당사에서 책임지지 않습니다.

2 라이브러리 시작과 종료

2.1 Start_Library

```
void Start_Library(int max_unit_number = 256);
```

Description

ezManager 라이브러리 사용을 위해 필요한 메모리를 동적 할당하고 라이브러리 함수들이 사용하는 변수들을 초기화 합니다.

Parameters

int max_unit_number

*EzTCP_Search*를 사용해서 검색할 수 있는 제품(ezTCP)의 최대 개수를 설정합니다. 이 값을 지정하지 않은 경우 기본값은 256입니다.

Return values

Examples

See also

End_Library

Remarks

ezManager 라이브러리 함수를 오류 없이 사용하기 위해 응용프로그램이 시작될 때 *Start_Library* 함수를 호출하십시오.

2.2 End_Library

```
void End_Library(void);
```

Description

Start_Library 함수 호출을 통해서 동적으로 할당한 메모리의 사용을 해제합니다.

Parameters

Return values

Examples

See also

Start_Library

Remarks

ezManager 라이브러리 사용을 종료하는 경우 *End_Library* 함수를 호출하여 라이브러리 내부에서 동적으로 할당하여 사용한 메모리를 해제하십시오. 그렇지 않은 경우 메모리 누수가 생길 수 있습니다.

3 제품(ezTCP) 검색

3.1 EzTCP_Search

```
int EzTCP_Search(int udp_port_number, int *error);
```

Description

로컬 네트워크에서 제품(ezTCP)을 검색합니다. 검색을 위해 암호화된 패킷을 UDP 50005번, 50007번 포트로 브로드캐스트 한 후 2초간 제품(ezTCP)의 응답을 기다립니다.

Parameters

int udp_port_number

제품(ezTCP)을 검색할 때 사용하는 포트 번호를 지정할 수 있습니다. 0을 입력하면 기본 포트번호인 50005번, 50007번을 사용합니다.

*in *error*

함수 실행이 실패한 경우 오류 값이 저장됩니다.

Return values

*EzTCP_Search*는 함수 호출이 성공하면 *EZTCP_SUCCESS*를 반환하고, 그렇지 않으면 *EZTCP_ERR* 을 반환하고 *error*에 시스템 오류 번호를 저장합니다.

Examples

```
int error = 0;
int res = 0;

start_library();

res = EzTCP_Search(0, &error);

if ( res == EZTCP_ERR )
{
    LPVOID lpMsgBuf;

    FormatMessage(FORMAT_MESSAGE_ALLOCATE_BUFFER | FORMAT_MESSAGE_FROM_SYSTEM |
        FORMAT_MESSAGE_IGNORE_INSERTS, NULL, error, MAKELANGID(LANG_NEUTRAL, SUBLANG_DEFAULT),
        (LPSTR)&lpMsgBuf, 0, NULL);

    printf("Error[%d] %s\r\n", error, (char*)lpMsgBuf);
}
else
{
    int i = 0;
```

```

    int eztcp_count = get_eztcp_count();

    for(i = 0; i < eztcp_count; i++)
    {
        struct eztcp_info out_eztcp_info;
        int ret = get_eztcp_info_by_index(i, &out_eztcp_info);

        if(ret == EZTCP_SUCCESS)
        {
            printf("MAC Addr. [%02x: %02x: %02x: %02x: %02x: %02x]\r\n",
                out_eztcp_info.mac_address[0], out_eztcp_info.mac_address[1],
                out_eztcp_info.mac_address[2], out_eztcp_info.mac_address[3],
                out_eztcp_info.mac_address[4], out_eztcp_info.mac_address[5]);
        }
    }
}

End_Library();

```

See also

EzTCP_Read, get_eztcp_count, get_eztcp_info_by_index

Remarks

*EzTCP_Search*는 제품(ezTCP)을 로컬 네트워크에서 검색하기 위해 UDP 브로드캐스트를 사용합니다. UDP 브로드캐스트는 사용자 컴퓨터에서 실행 중인 방화벽 등 보안 프로그램에 의해서 차단될 수 있습니다. 이런 경우 제품이 검색되지 않습니다.

3.2 EzTCP_Read

```

int EzTCP_Read(int search_method, unsigned char *mac_address, char *ip_addr,
               int udp_port_number, int *error);

```

Description

제품(ezTCP)의 MAC주소 또는 IP주소를 사용하여 네트워크에서 제품(ezTCP)을 검색합니다.

Parameters

int search_method

제품(ezTCP)을 검색할 때 MAC주소를 사용하는 경우 0을, IP주소를 사용하는 경우 1을 입력합니다

*unsigned char *mac_address*

제품의 MAC주소로 6바이트입니다.

*char *ip_addr*

제품의 IP주소 입니다.

int udp_port_number

제품(ezTCP)을 검색할 때 사용하는 포트 번호를 지정할 수 있습니다. 0을 입력하면 기본 포트번호인 50005번, 50007번을 사용합니다.

*int *error*

함수 실행이 실패한 경우 오류 값이 저장됩니다.

Return values

*EzTCP_Read*는 함수 호출이 성공하면 *EZTCP_SUCCESS*를 반환하고, 그렇지 않으면 *EZTCP_ERR* 을 반환하고 error에 시스템 오류 번호를 저장합니다.

Examples

```
int res = 0;
int error = 0;
unsigned char mac_address[6];
char *ip_addr = "10.1.0.1";

mac_address[0] = 0x00;
mac_address[1] = 0x30;
mac_address[2] = 0xf9;
mac_address[3] = 0x00;
mac_address[4] = 0x00;
mac_address[5] = 0x01;

Start_Library();

// MAC address
res = EzTCP_Read(0, mac_address, NULL, 0, &error);
if ( res == EZTCP_SUCCESS )
{
    int i = 0;
    int eztcp_count = get_eztcp_count();

    for(i = 0; i < eztcp_count; i++)
    {
        struct eztcp_info out_eztcp_info;
        int ret = get_eztcp_info_by_index(i, &out_eztcp_info);
```

```

        if(ret == EZTCP_SUCCESS)
        {
            printf("MAC Addr. [%02X: %02X: %02X: %02X: %02X: %02X]\r\n",
                out_eztcp_info.mac_address[0], out_eztcp_info.mac_address[1],
                out_eztcp_info.mac_address[2], out_eztcp_info.mac_address[3],
                out_eztcp_info.mac_address[4], out_eztcp_info.mac_address[5]);
        }
    }

// IP address
res = EzTCP_Read(1, NULL, ip_addr, 0, &error);
if ( res == EZTCP_SUCCESS )
{
    int i = 0;
    int eztcp_count = get_eztcp_count();

    for(i = 0; i < eztcp_count; i++)
    {
        struct eztcp_info out_eztcp_info;
        int ret = get_eztcp_info_by_index(i, &out_eztcp_info);

        if(ret == EZTCP_SUCCESS)
        {
            printf("IP Addr. [%s]\r\n", out_eztcp_info.hostname);
        }
    }

End_Library();

```

See also

EzTCP_Search, get_eztcp_count, get_eztcp_info_by_index

Remarks

*EzTCP_Read*는 *search method*가 0이면 제품(ezTCP)을 네트워크에서 검색하기 위해 UDP 브로드캐스트를 사용합니다. UDP 브로드캐스트는 사용자 컴퓨터에서 실행 중인 방화벽 등 보안 프로그램에 의해서 차단될 수 있습니다. 이런 경우 제품이 검색되지 않습니다.

3.3 get_eztcp_count

```
int get_eztcp_count(void);
```

Description

EzTCP_Search 또는 *EzTCP_Read*를 사용하여 검색된 제품(ezTCP)의 개수를 확인할 수 있습니다.

Parameters

Return values

EzTCP_Search 또는 *EzTCP_Read*를 사용하여 검색된 제품(ezTCP)의 총 개수를 반환합니다.

Examples

```
int error = 0;
int res = 0;

Start_Library();

res = EzTCP_Search(0, &error);

if ( res == EZTCP_ERR )
{
    LPVOID lpMsgBuf;

    FormatMessage(FORMAT_MESSAGE_ALLOCATE_BUFFER | FORMAT_MESSAGE_FROM_SYSTEM |
FORMAT_MESSAGE_IGNORE_INSERTS, NULL, error, MAKELANGID(LANG_NEUTRAL, SUBLANG_DEFAULT),
(LPSTR)&lpMsgBuf, 0, NULL);

    printf("Error[%d] %s\r\n", error, (char*)lpMsgBuf);
}
else
{
    int i = 0;
    int eztcp_count = get_eztcp_count();

    for(i = 0; i < eztcp_count; i++)
    {
        struct eztcp_info out_eztcp_info;
        int ret = get_eztcp_info_by_index(i, &out_eztcp_info);

        if(ret == EZTCP_SUCCESS)
        {
```

```

        printf("MAC Addr. [%02x: %02x: %02x: %02x: %02x: %02x]\r\n",
               out_eztcp_info.mac_address[0], out_eztcp_info.mac_address[1],
               out_eztcp_info.mac_address[2], out_eztcp_info.mac_address[3],
               out_eztcp_info.mac_address[4], out_eztcp_info.mac_address[5]);
    }
}

End_Library();

```

See also

EzTCP_Search, EzTCP_Read, get_eztcp_info_by_index

Remarks

3.4 get_eztcp_info_by_index

```
int get_eztcp_info_by_index(int index, struct eztcp_info *out_eztcp_info);
```

Description

검색된 제품(ezTCP)의 정보를 조회합니다.

Parameters

int index

검색된 제품(ezTCP) 중 정보를 조회할 제품의 순번으로 0부터 시작합니다.

*struct eztcp_info *out_eztcp_info*

제품(ezTCP) 정보를 저장할 eztcp_info 구조체의 포인터입니다.

Return values

index 값이 유효한 경우 *EZTCP_SUCCESS*을 반환하고 제품(ezTCP) 정보를 *out_eztcp_info*에 저장합니다. 그렇지 않은 경우 *EZTCP_ERR*을 반환합니다.

Examples

```

int error = 0;
int res = 0;

start_Library();

```

```

res = EzTCP_Search(0, &error);

if ( res == EZTCP_ERR )
{
    LPVOID lpMsgBuf;

    FormatMessage(FORMAT_MESSAGE_ALLOCATE_BUFFER | FORMAT_MESSAGE_FROM_SYSTEM |
FORMAT_MESSAGE_IGNORE_INSERTS, NULL, error, MAKELANGID(LANG_NEUTRAL, SUBLANG_DEFAULT),
(LPSTR)&lpMsgBuf, 0, NULL);

    printf("Error[%d] %s\r\n", error, (char*)lpMsgBuf);
}
else
{
    int i = 0;
    int eztcp_count = get_eztcp_count();

    for(i = 0; i < eztcp_count; i++)
    {
        struct eztcp_info out_eztcp_info;
        int ret = get_eztcp_info_by_index(i, &out_eztcp_info);

        if(ret == EZTCP_SUCCESS)
        {
            printf("MAC Addr. [%02x: %02x: %02x: %02x: %02x: %02x]\r\n",
out_eztcp_info.mac_address[0], out_eztcp_info.mac_address[1],
out_eztcp_info.mac_address[2], out_eztcp_info.mac_address[3],
out_eztcp_info.mac_address[4], out_eztcp_info.mac_address[5]);
        }
    }
}

End_Library();

```

See also

EzTCP_Search, EzTCP_Read, get_eztcp_count

Remarks

eztcp_info 구조체는 다음과 같은 구조를 가지고 있습니다.

```

struct eztcp_info
{
    unsigned char    mac_address[6];
    char             hostname[NI_MAXHOST];

```

```

unsigned char    comment[65];
int              env_status;
char             product_name[32];
int              search_method;
int              search_port;
}

```

mac_address

제품(ezTCP) MAC주소 입니다.

hostname

제품(ezTCP) IP주소 입니다.

comment

제품(ezTCP) 설명 입니다. *NI_MAXHOST*는 *ezManagerLibrary.h*에 1025로 정의되어 있습니다.

env_status

제품(ezTCP) 설정 값의 상태를 나타냅니다. 설정 값이 정상인 경우 1, CRC 검사 에러등의 문제가 있는 경우 0 입니다.

product_name

제품(ezTCP) 이름 입니다.

search_method

제품(ezTCP)을 검색한 방법입니다. *search_method*가 0인 경우 MAC 주소 검색, 1인 경우 IP 주소 검색입니다.

search_port

제품(ezTCP)을 검색한 UDP 포트번호 입니다.

4 설정 값 읽기

4.1 네트워크

4.1.1 get_arp

```
int get_arp(unsigned char *mac_address);
```

Description

*get_arp*는 [처음 수신된 패킷에서 IP 주소 얻기] 기능의 선택 여부를 반환합니다.

Parameters

*unsigned char *mac_address*

EzTCP_Search 또는 *EzTCP_Read*를 사용하여 검색된 제품(ezTCP)의 MAC주소 입니다.

Return values

*get_arp*는 [처음 수신된 패킷에서 IP 주소 얻기] 기능이 선택되어 있으면 1을 반환하고 그렇지 않은 경우 0을 반환합니다.

*mac_address*을 가진 제품(ezTCP)을 찾을 수 없는 경우에는 -1을 반환합니다.

Examples

See also

Remarks

4.1.2 get_dhcp

```
int get_dhcp(unsigned char *mac_address);
```

Description

*get_dhcp*는 [자동으로 IP 주소 받기(DHCP)] 기능의 선택 여부를 반환합니다.

Parameters

*unsigned char *mac_address*

EzTCP_Search 또는 *EzTCP_Read*를 사용하여 검색된 제품(ezTCP)의 MAC주소 입니다.

Return values

*get_dhcp*는 [자동으로 IP 주소 받기(DHCP)] 기능이 선택되어 있으면 1을 반환하고 그렇지 않은 경우 0을 반환합니다.

*mac_address*을 가진 제품(ezTCP)을 찾을 수 없는 경우에는 -1을 반환합니다.

Examples**See also**

get_dhcp_dns

Remarks**4.1.3 get_dhcp_dns**

```
int get_dhcp_dns(unsigned char *mac_address);
```

Description

*get_dhcp_dns*는 [자동으로 DNS 서버 주소 받기] 기능의 선택 여부를 반환합니다.

Parameters

*unsigned char *mac_address*

EzTCP_Search 또는 *EzTCP_Read*를 사용하여 검색된 제품(ezTCP)의 MAC주소 입니다.

Return values

*get_dhcp_dns*는 [자동으로 DNS 서버 주소 받기] 기능이 선택되어 있으면 1을 반환하고 그렇지 않은 경우 0을 반환합니다.

*mac_address*을 가진 제품(ezTCP)을 찾을 수 없는 경우에는 -1을 반환합니다.

Examples**See also**

get_dhcp

Remarks

[자동으로 DNS 서버 주소 받기]가 설정된 경우 제품(ezTCP)은 DHCP 서버로부터 할당 받은 DNS 서버 IP 주소를 사용합니다. 그렇지 않은 경우 사용자가 입력한 DNS 서버 IP 주소를 사용합니다.

4.1.4 get_pppoe

```
int get_pppoe(unsigned char *mac_address);
```

Description

*get_pppoe*는 [아이디로 IP 주소 받기(PPPoE)] 기능의 선택 여부를 반환합니다.

Parameters

*unsigned char *mac_address*

EzTCP_Search 또는 *EzTCP_Read*를 사용하여 검색된 제품(ezTCP)의 MAC주소 입니다.

Return values

*get_pppoe*는 [아이디로 IP 주소 받기(PPPoE)] 기능이 선택되어 있으면 1을 반환하고 그렇지 않은 경우 0을 반환합니다.

*mac_address*을 가진 제품(ezTCP)을 찾을 수 없는 경우에는 -1을 반환합니다.

Examples

See also

get_pppoe_id, *get_pppoe_pwd*

Remarks

4.1.5 get_ip4_local_address

```
int get_ip4_local_address(unsigned char *mac_address, char *out_address);
```

Description

*get_ip4_local_address*는 제품(ezTCP)의 IPv4 [제품 IP 주소]를 *out_addr*에 저장합니다.

Parameters

*unsigned char *mac_address*

EzTCP_Search 또는 *EzTCP_Read*를 사용하여 검색된 제품(ezTCP)의 MAC주소 입니다.

*char *out_address*

[제품 IP 주소]가 저장될 포인터입니다.

Return values

*get_ip4_local_address*는 *out_addr*에 [제품 IP 주소]를 저장하는데 성공하면 1을 반환하고 그렇지 않으면 0을 반환합니다.

*mac_address*을 가진 제품(ezTCP)을 찾을 수 없는 경우에는 -1을 반환합니다.

Examples

```
struct eztcp_info out_eztcp_info;
int ret = get_eztcp_info_by_index(0, &out_eztcp_info);

if(ret == EZTCP_SUCCESS)
{
    char buf[128];
    memset(buf, 0x00, 128);
    get_ip4_local_address(out_eztcp_info.mac_address, buf);
    printf("Local IPv4 : %s\r\n", buf);
}
```

See also

get_ip4_local_address, get_ip4_subnet_mask, get_ip4_gateway_address, get_ip4_dns_address

Remarks

*get_ip4_local_address*는 IPv4의 점으로 구분된 십진수 표기법으로 [제품 IP 주소]를 *out_address*에 저장합니다.

4.1.6 get_ip4_subnet_mask

```
int get_ip4_subnet_mask(unsigned char *mac_address, char *out_subnet_mask);
```

Description

*get_ip4_subnet_mask*는 제품(ezTCP)의 [서브넷 마스크]를 *out_addr*에 저장합니다.

Parameters

*unsigned char *mac_address*

EzTCP_Search 또는 *EzTCP_Read*를 사용하여 검색된 제품(ezTCP)의 MAC주소 입니다.

*char *out_subnet_mask*

[서브넷 마스크]가 저장될 포인터입니다.

Return values

*get_ip4_subnet_mask*는 *out_subnet_mask*에 [서브넷 마스크]를 저장하는데 성공하면 1을 반환하고 그렇지 않으면 0을 반환합니다.

*mac_address*을 가진 제품(ezTCP)을 찾을 수 없는 경우에는 -1을 반환합니다.

Examples

```
struct eztcp_info out_eztcp_info;
int ret = get_eztcp_info_by_index(0, &out_eztcp_info);

if(ret == EZTCP_SUCCESS)
{
    char buf[128];
    memset(buf, 0x00, 128);
    get_ip4_subnet_mask(out_eztcp_info.mac_address, buf);
    printf("Subnet mask : %s\r\n", buf);
}
```

See also

get_ip4_local_address, get_ip4_subnet_mask, get_ip4_gateway_address, get_ip4_dns_address

Remarks

*get_ip4_subnet_mask*는 IPv4의 점으로 구분된 십진수 표기법으로 [서브넷 마스크]를 *out_addr*에 저장합니다.

4.1.7 get_ip4_gateway_address

```
int get_ip4_gateway_address(unsigned char *mac_address, char *out_address);
```

Description

*get_ip4_gateway_address*는 제품(ezTCP)의 IPv4 [게이트웨이 IP 주소]를 *out_addr*에 저장합니다.

Parameters

*unsigned char *mac_address*

EzTCP_Search 또는 *EzTCP_Read*를 사용하여 검색된 제품(ezTCP)의 MAC주소 입니다.

*char *out_address*

[게이트웨이 IP 주소]가 저장될 포인터입니다.

Return values

*get_ip4_gateway_address*는 *out_address*에 [게이트웨이 IP 주소]를 저장하는데 성공하면 1을 반환하고 그렇지 않으면 0을 반환합니다.

*mac_address*을 가진 제품(ezTCP)을 찾을 수 없는 경우에는 -1을 반환합니다.

Examples

```
struct eztcp_info out_eztcp_info;
```

```
int ret = get_eztcp_info_by_index(0, &out_eztcp_info);

if(ret == EZTCP_SUCCESS)
{
    char buf[128];
    memset(buf, 0x00, 128);
    get_ip4_gateway_address(out_eztcp_info.mac_address, buf);
    printf("Gateway : %s\r\n", buf);
}
```

See also

get_ip4_local_address, get_ip4_subnet_mask, get_ip4_gateway_address, get_ip4_dns_address

Remarks

*get_ip4_gateway_address*는 IPv4의 점으로 구분된 십진수 표기법으로 [게이트웨이 IP 주소]를 *out_address*에 저장합니다.

4.1.8 get_ip4_dns_address

```
int get_ip4_dns_address(unsigned char *mac_address, char *out_address);
```

Description

*get_ip4_dns_address*는 제품(ezTCP)의 IPv4 [DNS 서버 IP 주소]를 *out_addr*에 저장합니다.

Parameters

*unsigned char *mac_address*

EzTCP_Search 또는 *EzTCP_Read*를 사용하여 검색된 제품(ezTCP)의 MAC주소 입니다.

*char *out_address*

[DNS 서버 IP 주소]가 저장될 포인터입니다.

Return values

*get_ip4_dns_address*는 *out_address*에 [DNS 서버 IP 주소]를 저장하는데 성공하면 1을 반환하고 그렇지 않으면 0을 반환합니다.

*mac_address*을 가진 제품(ezTCP)을 찾을 수 없는 경우에는 -1을 반환합니다.

Examples

```
struct eztcp_info out_eztcp_info;
int ret = get_eztcp_info_by_index(0, &out_eztcp_info);

if(ret == EZTCP_SUCCESS)
```

```
{
    char buf[128];
    memset(buf, 0x00, 128);
    get_ip4_dns_address(out_eztcp_info.mac_address, buf);
    printf("DNS : %s\r\n", buf);
}
```

See also

get_ip4_local_address, get_ip4_subnet_mask, get_ip4_gateway_address, get_ip4_dns_address.

Remarks

*get_ip4_dns_address*는 IPv4의 점으로 구분된 십진수 표기법으로 [DNS 서버 IP 주소]를 *out_address*에 저장합니다.

4.1.9 get_pppoe_id

```
int get_pppoe_id(unsigned char *mac_address, char *out_pppoe_id);
```

Description

*get_pppoe_id*는 제품(ezTCP)의 [PPPoE 아이디]를 *pppoe_id*에 저장합니다.

Parameters

*unsigned char *mac_address*

EzTCP_Search 또는 *EzTCP_Read*를 사용하여 검색된 제품(ezTCP)의 MAC주소 입니다.

*char *out_pppoe_id*

[PPPoE 아이디]가 저장될 포인터입니다.

Return values

*get_pppoe_id*는 *out_pppoe_id*에 [PPPoE 아이디]를 저장하는데 성공하면 1을 반환합니다.

*mac_address*을 가진 제품(ezTCP)을 찾을 수 없는 경우에는 -1을 반환합니다.

Examples

```
struct eztcp_info out_eztcp_info;
int ret = get_eztcp_info_by_index(0, &out_eztcp_info);

if(ret == EZTCP_SUCCESS)
{
    char buf[128];
    memset(buf, 0x00, 128);
```

```

    get_ppoe_id(out_eztcp_info.mac_address, buf);
    printf("PPPoE ID : %s\r\n", buf);
}

```

See also

get_pppoe, get_pppoe_pwd

Remarks

[PPPoE 아이디]는 NULL을 제외하고 최대 32바이트입니다. 따라서 *out_pppoe_id*는 최소 33바이트의 메모리 공간을 가지고 있어야 합니다.

4.1.10 get_pppoe_pwd

```
int get_pppoe_pwd(unsigned char *mac_address, char *out_pppoe_pwd);
```

Description

*get_pppoe_pwd*는 제품(ezTCP)의 [PPPoE 비밀번호]를 *out_pppoe_pwd*에 저장합니다.

Parameters

*unsigned char *mac_address*

EzTCP_Search 또는 *EzTCP_Read*를 사용하여 검색된 제품(ezTCP)의 MAC주소 입니다.

*char *out_pppoe_pwd*

[PPPoE 비밀번호]가 저장될 포인터입니다.

Return values

*get_pppoe_pwd*는 *out_pppoe_pwd*에 [PPPoE 비밀번호]를 저장하는데 성공하면 1을 반환합니다.

*mac_address*을 가진 제품(ezTCP)을 찾을 수 없는 경우에는 -1을 반환합니다.

Examples

```

struct eztcp_info out_eztcp_info;
int ret = get_eztcp_info_by_index(0, &out_eztcp_info);

if(ret == EZTCP_SUCCESS)
{
    char buf[128];
    memset(buf, 0x00, 128);
    get_pppoe_pwd(out_eztcp_info.mac_address, buf);
    printf("PPPoE password : %s\r\n", buf);
}

```

See also

get_pppoe, get_pppoe_id

Remarks

[PPPoE 비밀번호]는 NULL을 제외하고 최대 16바이트입니다. 따라서 *out_pppoe_pwd*는 최소 17바이트의 메모리 공간을 가지고 있어야 합니다.

4.1.11 get_ip6

```
int get_ip6(unsigned char *mac_address);
```

Description

*get_ip6*는 IPv6 기능의 선택 여부를 반환합니다.

Parameters

*unsigned char *mac_address*

EzTCP_Search 또는 *EzTCP_Read*를 사용하여 검색된 제품(ezTCP)의 MAC주소 입니다.

Return values

*get_ip6*는 IPv6 기능이 선택되어 있으면 1을 반환하고 그렇지 않은 경우 0을 반환합니다. *mac_address*을 가진 제품(ezTCP)을 찾을 수 없는 경우에는 -1을 반환합니다.

Examples**See also**

is_ip6, get_ip6_gua_type, get_ip6_eui_type, get_ip6_local_address, get_ip6_subnet_prefix_length, get_ip6_gateway_address, get_ip6_dns_address.

Remarks

IPv6에 대한 자세한 내용은 당사 홈페이지에서 IPv6 가이드 문서를 참조하십시오.

4.1.12 get_ip6_gua_type

```
int get_ip6_gua_type(unsigned char *mac_address);
```

Description

*get_ip6_gua_type*은 IPv6 [제품 IP 주소] 설정 방식을 반환합니다.

Parameters

*unsigned char *mac_address*

EzTCP_Search 또는 *EzTCP_Read*를 사용하여 검색된 제품(ezTCP)의 MAC주소 입니다.

Return values

*get_ip6_gua_type*은 IPv6 [제품 IP 주소] 설정 방식을 반환하며 그 값은 아래와 같습니다.

0

[자동으로 IP 주소 받기]

1

[고정된 IP 주소 사용]

*mac_address*을 가진 제품(ezTCP)을 찾을 수 없는 경우에는 -1을 반환합니다.

Examples

```
struct eztcp_info out_eztcp_info;
int ret = get_eztcp_info_by_index(0, &out_eztcp_info);

if(ret == EZTCP_SUCCESS)
{
    if(is_ip6(out_eztcp_info.mac_address) == 1)
    {
        int gua_type = get_ip6_gua_type(out_eztcp_info.mac_address);
        if(gua_type == 0)
            printf("[IPv6] Network Type : Obtain an IP automatically.\r\n");
        else if(gua_type == 1)
            printf("[IPv6] Network Type : Use static IP address.\r\n");
    }
}
```

See also

is_ip6, *get_ip6*, *get_ip6_eui_type*, *get_ip6_local_address*, *get_ip6_subnet_prefix_length*,
get_ip6_gateway_address, *get_ip6_dns_address*

Remarks

IPv6에 대한 자세한 내용은 당사 홈페이지에서 IPv6 가이드 문서를 참조하십시오.

4.1.13 get_ip6_eui_type

```
int get_ip6_eui_type(unsigned char *mac_address);
```

Description



*get_ip6_eui_type*은 IPv6 주소의 Interface ID 생성 방식을 반환합니다.

Parameters

*unsigned char *mac_address*

EzTCP_Search 또는 *EzTCP_Read*를 사용하여 검색된 제품(ezTCP)의 MAC주소 입니다.

Return values

*get_ip6_eui_type*은 Interface ID 생성 방식을 반환하며 그 값은 아래와 같습니다.

0

[MAC 주소]

1

[Random]

*mac_address*을 가진 제품(ezTCP)을 찾을 수 없는 경우에는 -1을 반환합니다.

Examples

```
struct eztcp_info out_eztcp_info;
int ret = get_eztcp_info_by_index(0, &out_eztcp_info);

if(ret == EZTCP_SUCCESS)
{
    if(is_ip6(out_eztcp_info.mac_address) == 1)
    {
        int eui_type = get_ip6_eui_type(out_eztcp_info.mac_address);
        if(gua_type == 0)
            printf("[IPv6] EUI : MAC address.\r\n");
        else if(gua_type == 1)
            printf("[IPv6] EUI : Random.\r\n");
    }
}
```

See also

is_ip6, *get_ip6*, *get_ip6_gua_type*, *get_ip6_local_address*, *get_ip6_subnet_prefix_length*,
get_ip6_gateway_address, *get_ip6_dns_address*

Remarks

IPv6에 대한 자세한 내용은 당사 홈페이지에서 IPv6 가이드 문서를 참조하십시오.

4.1.14 get_ip6_local_address

```
int get_ip6_local_address(unsigned char *mac_address, char *out_address);
```

Description

*get_ip6_local_address*는 제품(ezTCP)의 IPv6 [**제품 IP 주소**]를 *out_addr*에 저장합니다.

Parameters

*unsigned char *mac_address*

EzTCP_Search 또는 *EzTCP_Read*를 사용하여 검색된 제품(ezTCP)의 MAC주소 입니다.

*char *out_address*

IPv6 [**제품 IP 주소**]가 저장될 포인터입니다.

Return values

*get_ip6_local_address*는 *out_address*에 IPv6 [**제품 IP 주소**]를 저장하는데 성공하면 1을 반환합니다.

*mac_address*을 가진 제품(ezTCP)을 찾을 수 없는 경우에는 -1을 반환합니다.

Examples

```
struct eztcp_info out_eztcp_info;
int ret = get_eztcp_info_by_index(0, &out_eztcp_info);

if(ret == EZTCP_SUCCESS)
{
    unsigned char *mac_address = out_eztcp.mac_address;
    if(is_ip6(mac_address) == 1)
    {
        char buf[128];
        memset(buf, 0x00, 128);
        get_ip6_local_address(mac_address, buf);
        printf("Local IPv6 : %s\r\n", buf);
    }
}
```

See also

is_ip6, get_ip6, get_ip6_gua_type, get_ip6_eui_type, get_ip6_local_address_prefix,
get_ip6_gateway_address, get_ip6_dns_address

Remarks

IPv6에 대한 자세한 내용은 당사 홈페이지에서 IPv6 가이드 문서를 참조하십시오.

4.1.15 get_ip6_subnet_prefix_length

```
int get_ip6_subnet_prefix_length(unsigned char *mac_address,
                                char *out_subnet_prefix_length);
```

Description

*get_ip6_subnet_prefix_length*는 제품(ezTCP)의 IPv6 서브넷 접두사 길이를 *out_subnet_prefix_length*에 저장합니다.

Parameters

*unsigned char *mac_address*

EzTCP_Search 또는 *EzTCP_Read*를 사용하여 검색된 제품(ezTCP)의 MAC주소 입니다.

*char *out_subnet_prefix_length*

IPv6 서브넷 접두사 길이가 저장될 포인터입니다.

Return values

*get_ip6_subnet_prefix_length*는 *out_subnet_prefix_length*에 IPv6 서브넷 접두사 길이를 저장하는 데 성공하면 1을 반환합니다.

*mac_address*을 가진 제품(ezTCP)을 찾을 수 없는 경우에는 -1을 반환합니다.

Examples

```
struct eztcp_info out_eztcp_info;
int ret = get_eztcp_info_by_index(0, &out_eztcp_info);

if(ret == EZTCP_SUCCESS)
{
    unsigned char *mac_address = out_eztcp.mac_address;
    if(is_ip6(mac_address) == 1)
    {
        char buf[128];
        memset(buf, 0x00, 128);
        get_ip6_subnet_prefix_length(mac_address, buf);
        printf("Subnet prefix length : %s\r\n", buf);
    }
}
```

See also

is_ip6, get_ip6, get_ip6_gua_type, get_ip6_eui_type, get_ip6_local_address, get_ip6_gateway_address, get_ip6_dns_address

Remarks

IPv6에 대한 자세한 내용은 당사 홈페이지에서 IPv6 가이드 문서를 참조하십시오.

4.1.16 get_ip6_gateway_address

```
int get_ip6_gateway_address(unsigned char *mac_address, char *out_address);
```

Description

*get_ip6_gateway_address*는 제품(ezTCP)의 IPv6 [게이트웨이 IP 주소]를 *out_addr*에 저장합니다.

Parameters

*unsigned char *mac_address*

EzTCP_Search 또는 *EzTCP_Read*를 사용하여 검색된 제품(ezTCP)의 MAC주소 입니다.

*char *out_address*

IPv6 [게이트웨이 IP 주소]가 저장될 포인터입니다.

Return values

*get_ip6_gateway_address*는 *out_address*에 IPv6[제품 IP 주소]를 저장하는데 성공하면 1을 반환합니다.

*mac_address*을 가진 제품(ezTCP)을 찾을 수 없는 경우에는 -1을 반환합니다.

Examples

```
struct eztcp_info out_eztcp_info;
int ret = get_eztcp_info_by_index(0, &out_eztcp_info);

if(ret == EZTCP_SUCCESS)
{
    unsigned char *mac_address = out_eztcp.mac_address;
    if(is_ip6(mac_address) == 1)
    {
        char buf[128];
        memset(buf, 0x00, 128);
        get_ip6_gateway_address(mac_address, buf);
        printf("Gateway IPv6 : %s\r\n", buf);
    }
}
```

See also

is_ip6, get_ip6, get_ip6_gua_type, get_ip6_eui_type, get_ip6_local_address, get_ip6_subnet_prefix_length, get_ip6_dns_address

Remarks

IPv6에 대한 자세한 내용은 당사 홈페이지에서 IPv6 가이드 문서를 참조하십시오.

4.1.17 get_ip6_dns_address

```
int get_ip6_dns_address(unsigned char *mac_address, char *out_address);
```

Description

*get_ip6_dns_address*는 제품(ezTCP)의 IPv6 [DNS 서버 IP 주소]를 *out_addr*에 저장합니다.

Parameters

*unsigned char *mac_address*

EzTCP_Search 또는 *EzTCP_Read*를 사용하여 검색된 제품(ezTCP)의 MAC주소 입니다.

*char *out_address*

IPv6 [DNS 서버 IP 주소]가 저장될 포인터입니다.

Return values

*get_ip6_dns_address*는 *out_address*에 IPv6 [DNS 서버 IP 주소]를 저장하는데 성공하면 1을 반환합니다.

*mac_address*을 가진 제품(ezTCP)을 찾을 수 없는 경우에는 -1을 반환합니다.

Examples

```
struct eztcp_info out_eztcp_info;
int ret = get_eztcp_info_by_index(0, &out_eztcp_info);

if(ret == EZTCP_SUCCESS)
{
    if(is_ip6(out_eztcp_info.mac_address) == 1)
    {
        char buf[128];
        memset(buf, 0x00, 128);
        get_ip6_dns_address(out_eztcp_info.mac_address, buf);
        printf("DNS IPV6 : %s\r\n", buf);
    }
}
```

See also



is_ip6, get_ip6, get_ip6_gua_type, get_ip6_eui_type, get_ip6_local_address, get_ip6_subnet_prefix_length, get_ip6_gateway_address.

Remarks

IPv6에 대한 자세한 내용은 당사 홈페이지에서 IPv6 가이드 문서를 참조하십시오.

4.2 시리얼 포트

4.2.1 get_uart_count

```
int get_uart_count(unsigned char *mac_address);
```

Description

*get_uart_count*는 제품(ezTCP)이 지원하는 시리얼 포트 개수를 반환합니다.

Parameters

*unsigned char *mac_address*

EzTCP_Search 또는 *EzTCP_Read*를 사용하여 검색된 제품(ezTCP)의 MAC주소 입니다.

Return values

*get_uart_count*는 제품(ezTCP)이 지원하는 시리얼 포트 개수를 반환합니다.

*mac_address*을 가진 제품(ezTCP)을 찾을 수 없는 경우에는 -1을 반환합니다.

Examples

See also

get_uart_min_baudrate, get_uart_max_baudrate, get_uart_baudrate, get_uart_ttl, get_uart_telnet_type, get_uart_tx_delay, get_uart_parity, get_uart_databit, get_uart_stopbit, get_uart_flow_control, get_uart_dtrdsr.

Remarks

4.2.2 get_uart_min_baudrate

```
int get_uart_min_baudrate(unsigned char *mac_address);
```

Description

*get_uart_min_baudrate*는 제품(ezTCP)이 지원하는 최저 시리얼 통신속도를 반환합니다.

Parameters

*unsigned char *mac_address*

EzTCP_Search 또는 *EzTCP_Read*를 사용하여 검색된 제품(ezTCP)의 MAC주소 입니다.

Return values

*get_uart_min_baudrate*는 제품(ezTCP)이 지원하는 최저 시리얼 통신속도를 반환합니다.

*mac_address*을 가진 제품(ezTCP)을 찾을 수 없는 경우에는 -1을 반환합니다.

Examples**See also**

get_uart_count, *get_uart_max_baudrate*, *get_uart_baudrate*, *get_uart_serial_type*, *get_uart_ttl*,
get_uart_tx_delay, *get_uart_parity*, *get_uart_databit*, *get_uart_stopbit*, *get_uart_flow_control*,
get_uart_dtrdsr.

Remarks**4.2.3 get_uart_max_baudrate**

```
int get_uart_max_baudrate(unsigned char *mac_address, int uart_index);
```

Description

*get_uart_max_baudrate*는 제품(ezTCP)이 지원하는 최대 시리얼 통신속도를 반환합니다.

Parameters

*unsigned char *mac_address*

EzTCP_Search 또는 *EzTCP_Read*를 사용하여 검색된 제품(ezTCP)의 MAC주소 입니다.

int uart_index

제품(ezTCP)의 시리얼 포트 순번으로 0부터 시작합니다.

Return values

*get_uart_max_baudrate*는 제품(ezTCP)이 지원하는 최고 시리얼 통신속도를 반환합니다.

*mac_address*을 가진 제품(ezTCP)을 찾을 수 없는 경우에는 -1을 반환합니다.

Examples**See also**

get_uart_count, *get_uart_min_baudrate*, *get_uart_baudrate*, *get_uart_serial_type*, *get_uart_ttl*,
get_uart_tx_delay, *get_uart_parity*, *get_uart_databit*, *get_uart_stopbit*, *get_uart_flow_control*,

get_uart_dtrdsr.

Remarks

제품(ezTCP)이 지원하는 최대 시리얼 통신속도는 다른 설정 값에 따라 달라질 수 있습니다.

4.2.4 get_uart_serial_type

```
int get_uart_serial_type(unsigned char *mac_address, int uart_index);
```

Description

*get_uart_serial_type*은 *uart_index*로 지정된 시리얼 포트의 [시리얼 종류]를 반환합니다.

Parameters

*unsigned char *mac_address*

EzTCP_Search 또는 *EzTCP_Read*를 사용하여 검색된 제품(ezTCP)의 MAC주소 입니다.

int uart_index

제품(ezTCP)의 시리얼 포트 순번으로 0부터 시작합니다.

Return values

*get_uart_serial_type*은 *uart_index*로 지정된 시리얼 포트의 [시리얼 종류]를 반환하며 그 값은 아래와 같습니다.

0

RS-232

1

RS-485

2

RS-422

*mac_address*을 가진 제품(ezTCP)을 찾을 수 없는 경우에는 -1을 반환합니다.

Examples

See also

is_uart_rs232, *is_uart_rs485*, *is_uart_rs422*, *get_uart_count*, *get_uart_min_baudrate*,
get_uart_max_baudrate, *get_uart_baudrate*, *get_uart_ttl*, *get_uart_tx_delay*, *get_uart_parity*,
get_uart_databit, *get_uart_stopbit*, *get_uart_flow_control*, *get_uart_dtrdsr*.

Remarks

4.2.5 get_uart_ttl

```
int get_uart_ttl(unsigned char *mac_address, int uart_index);
```

Description

*get_uart_ttl*은 *uart_index*로 지정된 시리얼 포트의 [TTL] 출력 기능의 선택 여부를 반환합니다.

Parameters

*unsigned char *mac_address*

EzTCP_Search 또는 *EzTCP_Read*를 사용하여 검색된 제품(ezTCP)의 MAC주소 입니다.

int uart_index

제품(ezTCP)의 시리얼 포트 순번으로 0부터 시작합니다.

Return values

*get_uart_ttl*은 *uart_index*로 지정된 시리얼 포트가 [TTL] 출력 기능을 사용 중 이면 1을 반환하고 그렇지 않으면 0을 반환합니다.

*mac_address*을 가진 제품(ezTCP)을 찾을 수 없는 경우에는 -1을 반환합니다.

Examples

See also

get_uart_count, *get_uart_min_baudrate*, *get_uart_max_baudrate*, *get_uart_baudrate*, *get_uart_serial_type*, *is_uart_ttl*, *get_uart_tx_delay*, *get_uart_parity*, *get_uart_databit*, *get_uart_stopbit*, *get_uart_flow_control*, *get_uart_dtrdsr*

Remarks

4.2.6 get_uart_baudrate

```
int get_uart_baudrate(unsigned char *mac_address, int uart_index);
```

Description

*get_uart_baudrate*는 *uart_index*로 지정된 시리얼 포트의 [시리얼 통신속도]를 반환합니다.

Parameters

*unsigned char *mac_address*

EzTCP_Search 또는 *EzTCP_Read*를 사용하여 검색된 제품(ezTCP)의 MAC주소 입니다.

int uart_index

제품(ezTCP)의 시리얼 포트 순번으로 0부터 시작합니다.

Return values

*get_uart_baudrate*는 *uart_index*로 지정된 시리얼 포트의 [시리얼 통신속도]를 반환합니다.
*mac_address*을 가진 제품(ezTCP)을 찾을 수 없는 경우에는 -1을 반환합니다.

Examples

See also

get_uart_count, *get_uart_min_baudrate*, *get_uart_max_baudrate*, *get_uart_serial_type*, *get_uart_ttl*,
get_uart_tx_delay, *get_uart_parity*, *get_uart_databit*, *get_uart_stopbit*, *get_uart_flow_control*,
get_uart_dtrdsr,

Remarks

*uart_index*로 지정된 시리얼 포트의 [시리얼 통신속도]는 다른 설정 값에 따라 자동으로 변경될 수 있습니다.

4.2.7 get_uart_parity

```
int get_uart_parity(unsigned char *mac_address, int uart_index);
```

Description

*get_uart_parity*는 *uart_index*로 지정된 시리얼 포트의 [패리티]를 반환합니다.

Parameters

*unsigned char *mac_address*

EzTCP_Search 또는 *EzTCP_Read*를 사용하여 검색된 제품(ezTCP)의 MAC주소 입니다.

int uart_index

제품(ezTCP)의 시리얼 포트 순번으로 0부터 시작합니다.

Return values

*get_uart_parity*은 *uart_index*로 지정된 시리얼 포트의 [패리티]를 반환하며 그 값은 아래와 같습니다.

0

NONE

1

EVEN

2

ODD

3

MARK

4

SPACE

*mac_address*을 가진 제품(ezTCP)을 찾을 수 없는 경우에는 -1을 반환합니다.

Examples

See also

`get_uart_count`, `get_uart_min_baudrate`, `get_uart_max_baudrate`, `get_uart_baudrate`, `get_uart_ttl`,
`get_uart_tx_delay`, `get_uart_databit`, `get_uart_stopbit`, `get_uart_flow_control`, `get_uart_dtrdsr`,

Remarks

4.2.8 get_uart_databit

```
int get_uart_databit(unsigned char *mac_address, int uart_index);
```

Description

*get_uart_databit*는 *uart_index*로 지정된 시리얼 포트의 [데이터 비트]를 반환합니다.

Parameters

*unsigned char *mac_address*

EzTCP_Search 또는 *EzTCP_Read*를 사용하여 검색된 제품(ezTCP)의 MAC주소 입니다.

int uart_index

제품(ezTCP)의 시리얼 포트 순번으로 0부터 시작합니다.

Return values

*get_uart_databit*는 *uart_index*로 지정된 시리얼 포트의 [데이터 비트]를 반환하며 그 값은 아래와 같습니다.

0

5비트

1

6비트

2

7비트

3

8비트

*mac_address*을 가진 제품(ezTCP)을 찾을 수 없는 경우에는 -1을 반환합니다.

Examples

See also

get_uart_count, get_uart_min_baudrate, get_uart_max_baudrate, get_uart_baudrate, is_uart_ttl, get_uart_tx_delay, get_uart_parity, is_uart_8_databit_only, is_uart_7_and_8_databit_only, get_uart_stopbit, get_uart_flow_control, get_uart_dtrdsr

Remarks

is_uart_8_databit_only이 1을 반환하면 8비트 데이터 비트만 지원합니다.

is_uart_7_and_8_databit_only이 1을 반환하면 7, 8비트 데이터 비트만 지원합니다.

4.2.9 get_uart_stopbit

```
int get_uart_stopbit(unsigned char *mac_address, int uart_index);
```

Description

get_uart_stopbit는 uart_index로 지정된 시리얼 포트의 [정지 비트]를 반환합니다.

Parameters

unsigned char *mac_address

EzTCP_Search 또는 EzTCP_Read를 사용하여 검색된 제품(ezTCP)의 MAC주소 입니다.

int uart_index

제품(ezTCP)의 시리얼 포트 순번으로 0부터 시작합니다.

Return values

get_uart_stopbit은 uart_index로 지정된 시리얼 포트의 [정지 비트]를 반환하며 그 값은 아래와 같습니다.

0

1비트

1

1.5비트

2

2비트

mac_address을 가진 제품(ezTCP)을 찾을 수 없는 경우에는 -1을 반환합니다.

Examples

See also

get_uart_count, get_uart_min_baudrate, get_uart_max_baudrate, get_uart_baudrate, get_uart_ttl,

get_uart_tx_delay, get_uart_parity, get_uart_databit, is_uart_one5_stopbit, get_uart_flow_control, get_uart_dtrdsr,

Remarks

*is_uart_one5_stopbit*가 1을 반환하는 경우에만 1.5비트 [정지 비트]를 사용할 수 있습니다.

4.2.10 get_uart_flow_control

```
int get_uart_flow_control(unsigned char *mac_address, int uart_index);
```

Description

*get_uart_flow_control*는 *uart_index*로 지정된 시리얼 포트의 [흐름 제어]를 반환합니다.

Parameters

*unsigned char *mac_address*

EzTCP_Search 또는 *EzTCP_Read*를 사용하여 검색된 제품(ezTCP)의 MAC주소 입니다.

int uart_index

제품(ezTCP)의 시리얼 포트 순번으로 0부터 시작합니다.

Return values

*get_uart_flow_control*은 *uart_index*로 지정된 시리얼 포트의 [흐름 제어]를 반환하며 그 값은 아래와 같습니다.

0

NONE

1

RTS/CTS

2

XON/XOFF

*mac_address*을 가진 제품(ezTCP)을 찾을 수 없는 경우에는 -1을 반환합니다.

Examples

See also

get_uart_count, get_uart_min_baudrate, get_uart_max_baudrate, get_uart_baudrate, get_uart_ttl, get_uart_tx_delay, get_uart_parity, get_uart_databit, get_uart_stopbit, get_uart_dtrdsr.

Remarks

4.2.11 get_uart_dtrdsr

```
int get_uart_dtrdsr(unsigned char *mac_address, int uart_index);
```

Description

*get_uart_dtrdsr*은 *uart_index*로 지정된 시리얼 포트의 [DTR/DSR] 기능의 선택 여부를 반환합니다.

Parameters

*unsigned char *mac_address*

EzTCP_Search 또는 *EzTCP_Read*를 사용하여 검색된 제품(ezTCP)의 MAC주소 입니다.

int uart_index

제품(ezTCP)의 시리얼 포트 순번으로 0부터 시작합니다.

Return values

*get_uart_dtrdsr*은 *uart_index*로 지정된 시리얼 포트가 [DTR/DSR] 기능을 사용 중 이면 1을 반환하고 그렇지 않으면 0을 반환합니다.

*mac_address*을 가진 제품(ezTCP)을 찾을 수 없는 경우에는 -1을 반환합니다.

Examples

See also

get_uart_count, *get_uart_min_baudrate*, *get_uart_max_baudrate*, *get_uart_baudrate*, *get_uart_ttl*, *get_uart_tx_delay*, *get_uart_parity*, *get_uart_databit*, *get_uart_stopbit*, *get_uart_flow_control*, *is_uart_dtrdsr*

Remarks

4.2.12 get_uart_tx_delay

```
int get_uart_tx_delay(unsigned char *mac_address, int uart_index);
```

Description

*get_uart_tx_delay*는 *uart_index*로 지정된 시리얼 포트의 [데이터 전송 간격]을 반환합니다.

Parameters

*unsigned char *mac_address*

EzTCP_Search 또는 *EzTCP_Read*를 사용하여 검색된 제품(ezTCP)의 MAC주소 입니다.

int uart_index

제품(ezTCP)의 시리얼 포트 순번으로 0부터 시작합니다.

Return values

*get_uart_tx_delay*는 *uart_index*로 지정된 시리얼 포트의 [데이터 전송 간격]을 반환합니다.
*mac_address*을 가진 제품(ezTCP)을 찾을 수 없는 경우에는 -1을 반환합니다.

Examples**See also**

get_uart_count, *get_uart_min_baudrate*, *get_uart_max_baudrate*, *get_uart_baudrate*, *get_uart_ttl*,
is_uart_tx_delay, *get_uart_parity*, *get_uart_databit*, *get_uart_stopbit*, *get_uart_flow_control*, *get_uart_dtrdsr*

Remarks

4.3 TCP/IP

4.3.1 get_uart_communication_mode

```
int get_uart_communication_mode(unsigned char *mac_address, int uart_index);
```

Description

*get_uart_communication_mode*는 *uart_index*로 지정된 시리얼 포트의 [통신모드]를 반환합니다.

Parameters

*unsigned char *mac_address*

EzTCP_Search 또는 *EzTCP_Read*를 사용하여 검색된 제품(ezTCP)의 MAC주소 입니다.

int uart_index

제품(ezTCP)의 시리얼 포트 순번으로 0부터 시작합니다.

Return values

*get_uart_communication_mode*는 *uart_index*로 지정된 시리얼 포트의 [통신모드]를 반환하며 그 값은 아래와 같습니다.

0

T2S – TCP 서버

1

ATC – AT 명령

2

COD – TCP 클라이언트

3

U2S – UDP

4

시리얼 Modbus/TCP

*mac_address*을 가진 제품(ezTCP)을 찾을 수 없는 경우에는 -1을 반환합니다.

Examples

See also

is_uart_tcp_server, *is_uart_at_command*, *is_uart_tcp_client*, *is_uart_udp*, *is_uart_serial_modbus*,
get_uart_peer_address, *get_uart_peer_port*, *get_uart_local_port*, *get_uart_cod_tcp_server*,
get_uart_watermark, *get_uart_timeout*, *get_uart_data_frame_interval*, *get_uart_separator_length*,
get_uart_separator_type, *get_uart_separator*, *get_uart_tcp_nodelay*, *get_uart_rfc2217*, *get_uart_protocol*

Remarks

4.3.2 *get_uart_peer_address*

```
int get_uart_peer_address(unsigned char *mac_address, int uart_index,
                          char *out_address);
```

Description

*get_uart_peer_address*는 *uart_index*로 지정된 시리얼 포트의 [통신할 주소]를 *out_addr*에 저장합니다.

Parameters

*unsigned char *mac_address*

EzTCP_Search 또는 *EzTCP_Read*를 사용하여 검색된 제품(ezTCP)의 MAC주소 입니다.

int uart_index

제품(ezTCP)의 시리얼 포트 순번으로 0부터 시작합니다.

*char *out_address*

[통신할 주소]가 저장될 포인터입니다.

Return values

*get_uart_peer_address*는 *out_address*에 [통신할 주소]를 저장하는데 성공하면 1을 반환하고 그렇지 않으면 0을 반환합니다.

*mac_address*을 가진 제품(ezTCP)을 찾을 수 없는 경우에는 -1을 반환합니다.

Examples

```
struct eztcp_info out_eztcp_info;
int ret = get_eztcp_info_by_index(0, &out_eztcp_info);
```

```

if(ret == EZTCP_SUCCESS)
{
    int i = 0;
    char buf[65];
    int uart_count = get_uart_count(out_eztcp_info.mac_address);
    for(i=0;i<uart_count;i++)
    {
        memset(buf, 0x00, 65);
        get_uart_peer_address(out_eztcp_info.mac_address, i, buf);
        printf("UART[%d] Peer address : %s\r\n", i, buf);
    }
}

```

See also

get_uart_communication_mode, get_uart_peer_port, get_uart_local_port, get_uart_cod_tcp_server, get_uart_watermark, get_uart_timeout, get_uart_data_frame_interval, get_uart_separator_length, get_uart_separator_type, get_uart_separator, get_uart_tcp_nodelay, get_uart_rfc2217, get_uart_protocol

Remarks

[통신할 주소]는 NULL을 제외하고 최대 64바이트입니다. 따라서 *out_address*은 최소 65바이트의 메모리 공간을 가지고 있어야 합니다.

4.3.3 get_uart_peer_port

```
int get_uart_peer_port(unsigned char *mac_address, int uart_index, char *out_port);
```

Description

*get_uart_peer_port*는 *uart_index*로 지정된 시리얼 포트의 [통신할 포트]를 *out_port*에 저장합니다.

Parameters

*unsigned char *mac_address*

EzTCP_Search 또는 *EzTCP_Read*를 사용하여 검색된 제품(*ezTCP*)의 MAC주소 입니다.

int uart_index

제품(*ezTCP*)의 시리얼 포트 순번으로 0부터 시작합니다.

*char *out_port*

[통신할 포트]가 저장될 포인터입니다.

Return values

*get_uart_peer_port*는 *out_port*에 [통신할 포트]를 저장하는데 성공하면 1을 반환합니다.
*mac_address*을 가진 제품(ezTCP)을 찾을 수 없는 경우에는 -1을 반환합니다.

Examples

```
struct eztcp_info out_eztcp_info;
int ret = get_eztcp_info_by_index(0, &out_eztcp_info);

if(ret == EZTCP_SUCCESS)
{
    int i = 0;
    char buf[65];
    int uart_count = get_uart_count(out_eztcp_info.mac_address);
    for(i=0;i<uart_count;i++)
    {
        memset(buf, 0x00, 65);
        get_uart_peer_port(out_eztcp_info.mac_address, i, buf);
        printf("UART[%d] Peer port : %s\r\n", i, buf);
    }
}
```

See also

get_uart_communication_mode, *get_uart_peer_address*, *get_uart_local_port*, *get_uart_cod_tcp_server*,
get_uart_watermark, *get_uart_timeout*, *get_uart_data_frame_interval*, *get_uart_separator_length*,
get_uart_separator_type, *get_uart_separator*, *get_uart_tcp_nodelay*, *get_uart_rfc2217*, *get_uart_protocol*

Remarks

4.3.4 get_uart_local_port

```
int get_uart_local_port(unsigned char *mac_address, int uart_index, char *out_port);
```

Description

*get_uart_local_port*는 *uart_index*로 지정된 시리얼 포트의 [제품 로컬포트]를 *out_port*에 저장합니다.

Parameters

*unsigned char *mac_address*

EzTCP_Search 또는 *EzTCP_Read*를 사용하여 검색된 제품(ezTCP)의 MAC주소 입니다.

int uart_index

제품(ezTCP)의 시리얼 포트 순번으로 0부터 시작합니다.

*char *out_port*

[제품 로컬포트]가 저장될 포인터입니다.

Return values

*get_uart_local_port*는 *out_port*에 [제품 로컬포트]를 저장하는데 성공하면 1을 반환합니다.
*mac_address*을 가진 제품(ezTCP)을 찾을 수 없는 경우에는 -1을 반환합니다.

Examples

```
struct eztcp_info out_eztcp_info;
int ret = get_eztcp_info_by_index(0, &out_eztcp_info);

if(ret == EZTCP_SUCCESS)
{
    int i = 0;
    char buf[65];
    int uart_count = get_uart_count(out_eztcp_info.mac_address);
    for(i=0; i<uart_count; i++)
    {
        memset(buf, 0x00, 65);
        get_uart_local_port(out_eztcp_info.mac_address, i, buf);
        printf("UART[%d] Local port : %s\r\n", i, buf);
    }
}
```

See also

get_uart_communication_mode, *get_uart_peer_address*, *get_uart_peer_port*, *get_uart_cod_tcp_server*,
get_uart_watermark, *get_uart_timeout*, *get_uart_data_frame_interval*, *get_uart_separator_length*,
get_uart_separator_type, *get_uart_separator*, *get_uart_tcp_nodelay*, *get_uart_rfc2217*, *get_uart_protocol*

Remarks

4.3.5 get_uart_cod_tcp_server

```
int get_uart_cod_tcp_server(unsigned char *mac_address, int uart_index);
```

Description

*get_uart_cod_tcp_server*은 *uart_index*로 지정된 시리얼 포트의 [TCP 서버] 기능의 선택 여부를 반환합니다.

Parameters



*unsigned char *mac_address*

EzTCP_Search 또는 *EzTCP_Read*를 사용하여 검색된 제품(ezTCP)의 MAC주소 입니다.

int uart_index

제품(ezTCP)의 시리얼 포트 순번으로 0부터 시작합니다.

Return values

*get_uart_cod_tcp_server*은 *uart_index*로 지정된 시리얼 포트가 [TCP 서버] 기능을 사용 중 이면 1을 반환하고 그렇지 않으면 0을 반환합니다.

*mac_address*을 가진 제품(ezTCP)을 찾을 수 없는 경우에는 -1을 반환합니다.

Examples

See also

get_uart_communication_mode, *get_uart_peer_address*, *get_uart_peer_port*, *get_uart_local_port*, *get_uart_watermark*, *get_uart_timeout*, *get_uart_data_frame_interval*, *get_uart_separator_length*, *get_uart_separator_type*, *get_uart_separator*, *get_uart_tcp_nodelay*, *get_uart_rfc2217*, *get_uart_protocol*

Remarks

4.3.6 get_uart_watermark

```
int get_uart_watermark(unsigned char *mac_address, int uart_index, char *out_watermark);
```

Description

*get_uart_watermark*는 *uart_index*로 지정된 시리얼 포트의 [접속전 데이터 크기] 또는 [패킷 블록 설정(바이트)]를 반환합니다.

Parameters

*unsigned char *mac_address*

EzTCP_Search 또는 *EzTCP_Read*를 사용하여 검색된 제품(ezTCP)의 MAC주소 입니다.

int uart_index

제품(ezTCP)의 시리얼 포트 순번으로 0부터 시작합니다.

*char *out_watermark*

[접속전 데이터 크기] 또는 [패킷 블록 설정(바이트)]가 저장될 포인터입니다.

Return values

*get_uart_watermark*는 *out_watermark*에 [접속전 데이터 크기] 또는 [패킷 블록 설정(바이트)]를 저장하는데 성공하면 1을 반환합니다.

*mac_address*을 가진 제품(ezTCP)을 찾을 수 없는 경우에는 -1을 반환합니다.

Examples

```

struct eztcp_info out_eztcp_info;
int ret = get_eztcp_info_by_index(0, &out_eztcp_info);

if(ret == EZTCP_SUCCESS)
{
    int i = 0;
    char buf[65];
    int uart_count = get_uart_count(out_eztcp_info.mac_address);
    for(i=0;i<uart_count;i++)
    {
        memset(buf, 0x00, 65);
        int comm_mode = get_uart_communication_mode(out_eztcp_info.mac_address, i);
        if(comm_mode==MUX_TYPE_T2S || comm_mode==MUX_TYPE_ATC || comm_mode==MUX_TYPE_COD)
        {
            get_uart_watermark(out_eztcp_info.mac_address, i, buf);
            printf("UART[%d] Event byte : %s\r\n", i, buf);
        }
        else if(comm_mode==MUX_TYPE_U2S)
        {
            get_uart_watermark(out_eztcp_info.mac_address, i, buf);
            printf("UART[%d] Block size (byte) : %s\r\n", i, buf);
        }
        else
        {
            printf("Not available.\r\n");
        }
    }
}

```

See also

get_uart_communication_mode, get_uart_peer_address, get_uart_peer_port, get_uart_local_port, get_uart_cod_tcp_server, get_uart_timeout, get_uart_data_frame_interval, get_uart_separator_length, get_uart_separator_type, get_uart_separator, get_uart_tcp_nodelay, get_uart_rfc2217, get_uart_protocol

Remarks

[통신모드]가 0 (T2S - TCP 서버), 1 (ATC - AT 명령), 2 (COD - TCP 클라이언트)인 경우 [접속전 데이터 크기]로 사용됩니다.

[통신모드]가 3 (U2S - UDP)인 경우 [패킷 블록 설정 (바이트)]로 사용됩니다.

4.3.7 get_uart_timeout

```
int get_uart_timeout(unsigned char *mac_address, int uart_index, char *out_timeout);
```

Description

*get_uart_timeout*은 *uart_index*로 지정된 시리얼 포트의 [접속종료 대기시간(초)]를 반환합니다.

Parameters

*unsigned char *mac_address*

EzTCP_Search 또는 *EzTCP_Read*를 사용하여 검색된 제품(ezTCP)의 MAC주소 입니다.

int uart_index

제품(ezTCP)의 시리얼 포트 순번으로 0부터 시작합니다.

*char *out_timeout*

[접속종료 대기시간(초)]이 저장될 포인터입니다.

Return values

*get_uart_timeout*은 *out_timeout*에 [접속종료 대기시간(초)]를 저장하는데 성공하면 1을 반환합니다.

*mac_address*을 가진 제품(ezTCP)을 찾을 수 없는 경우에는 -1을 반환합니다.

Examples

```
struct eztcp_info out_eztcp_info;
int ret = get_eztcp_info_by_index(0, &out_eztcp_info);

if(ret == EZTCP_SUCCESS)
{
    int i = 0;
    char buf[65];
    int uart_count = get_uart_count(out_eztcp_info.mac_address);
    for(i=0; i<uart_count; i++)
    {
        memset(buf, 0x00, 65);
        get_uart_timeout(out_eztcp_info.mac_address, i, buf);
        printf("UART[%d] Timeout : %s\r\n", i, buf);
    }
}
```

See also

get_uart_communication_mode, *get_uart_peer_address*, *get_uart_peer_port*, *get_uart_local_port*,

get_uart_cod_tcp_server, get_uart_watermark, get_uart_data_frame_interval, get_uart_separator_length, get_uart_separator_type, get_uart_separator, get_uart_tcp_nodelay, get_uart_rfc2217, get_uart_protocol

Remarks

4.3.8 get_uart_data_frame_interval

```
int get_uart_data_frame_interval(unsigned char *mac_address, int uart_index,
                                char *out_data_frame_interval);
```

Description

*get_uart_data_frame_interval*은 *uart_index*로 지정된 시리얼 포트의 [데이터 프레임 간격(10ms)]를 반환합니다.

Parameters

*unsigned char *mac_address*

EzTCP_Search 또는 *EzTCP_Read*를 사용하여 검색된 제품(ezTCP)의 MAC주소 입니다.

int uart_index

제품(ezTCP)의 시리얼 포트 순번으로 0부터 시작합니다.

*char *out_data_frame_interval*

[데이터 프레임 간격(10ms)]이 저장될 포인터입니다.

Return values

*get_uart_data_frame_interval*은 *out_data_frame_interval*에 [데이터 프레임 간격(10ms)]를 저장하는데 성공하면 1을 반환합니다.

*mac_address*을 가진 제품(ezTCP)을 찾을 수 없는 경우에는 -1을 반환합니다.

Examples

```
struct eztcp_info out_eztcp_info;
int ret = get_eztcp_info_by_index(0, &out_eztcp_info);

if(ret == EZTCP_SUCCESS)
{
    int i = 0;
    char buf[65];
    int uart_count = get_uart_count(out_eztcp_info.mac_address);
    unsigned char *mac_address = out_eztcp_info.mac_address;
    for(i=0; i<uart_count; i++)
    {
        memset(buf, 0x00, 65);
```

```

        get_uart_data__frame_interval(mac_address, i, buf);
        printf("UART[%d] Data frame interval : %s\r\n", i, buf);
    }
}

```

See also

get_uart_communication_mode, get_uart_peer_address, get_uart_peer_port, get_uart_local_port,
get_uart_cod_tcp_server, get_uart_watermark, get_uart_timeout, is_uart_data_frame_interval,
get_uart_separator_length, get_uart_separator_type, get_uart_separator, get_uart_tcp_nodelay,
get_uart_rfc2217, get_uart_protocol

Remarks

4.3.9 get_uart_separator_length

```
int get_uart_separator_length(unsigned char *mac_address, int uart_index);
```

Description

*get_uart_separator_length*는 *uart_index*로 지정된 시리얼 포트의 [구분자 길이]를 반환합니다.

Parameters

*unsigned char *mac_address*

EzTCP_Search 또는 *EzTCP_Read*를 사용하여 검색된 제품(ezTCP)의 MAC주소 입니다.

int uart_index

제품(ezTCP)의 시리얼 포트 순번으로 0부터 시작합니다.

Return values

*get_uart_separator_length*는 [구분자 길이]를 반환하며 그 값은 0 ~ 4입니다.

*mac_address*을 가진 제품(ezTCP)을 찾을 수 없는 경우에는 -1을 반환합니다.

Examples

See also

get_uart_communication_mode, get_uart_peer_address, get_uart_peer_port, get_uart_local_port,
get_uart_cod_tcp_server, get_uart_watermark, get_uart_timeout, get_uart_data_frame_interval,
is_uart_separator, get_uart_separator_type, get_uart_separator, get_uart_tcp_nodelay, get_uart_rfc2217,
get_uart_protocol

Remarks

4.3.10 get_uart_separator_type

```
int get_uart_separator_type(unsigned char *mac_address, int uart_index);
```

Description

*get_uart_separator_type*은 *uart_index*로 지정된 시리얼 포트의 [구분자 동작방식]을 반환합니다.

Parameters

*unsigned char *mac_address*

EzTCP_Search 또는 *EzTCP_Read*를 사용하여 검색된 제품(ezTCP)의 MAC주소 입니다.

int uart_index

제품(ezTCP)의 시리얼 포트 순번으로 0부터 시작합니다.

Return values

*get_uart_separator_type*은 [구분자 동작방식]을 반환하며 그 값은 아래와 같습니다.

0

구분자까지 전송

1

구분자+1바이트 전송

2

구분자+2바이트 전송

*mac_address*을 가진 제품(ezTCP)을 찾을 수 없는 경우에는 -1을 반환합니다.

Examples

See also

get_uart_communication_mode, *get_uart_peer_address*, *get_uart_peer_port*, *get_uart_local_port*,
get_uart_cod_tcp_server, *get_uart_watermark*, *get_uart_timeout*, *get_uart_data_frame_interval*,
is_uart_separator, *get_uart_separator_length*, *get_uart_separator*, *get_uart_tcp_nodelay*, *get_uart_rfc2217*,
get_uart_protocol

Remarks

4.3.11 get_uart_separator

```
int get_uart_separator(unsigned char *mac_address, int uart_index, int separator_index,  
char *out_separator);
```

Description

*get_uart_separator*는 *uart_index*로 지정된 시리얼 포트의 [구분자(16진수)]를 반환합니다.

Parameters

*unsigned char *mac_address*

EzTCP_Search 또는 *EzTCP_Read*를 사용하여 검색된 제품(ezTCP)의 MAC주소 입니다.

int uart_index

제품(ezTCP)의 시리얼 포트 순번으로 0부터 시작합니다.

int separator_index

구분자의 순번으로 0부터 시작합니다.

*char *out_separator*

*separator_index*로 지정된 구분자가 16진수 문자열로 저장될 포인터입니다.

Return values

*get_uart_separator*는 *out_separator*에 *separator_index*로 지정된 구분자를 16진수 문자열로 저장하는데 성공하면 1을 반환합니다.

*separator_index*가 지정된 범위를 벗어나는 경우에는 -2를 반환합니다.

*mac_address*을 가진 제품(ezTCP)을 찾을 수 없는 경우에는 -1을 반환합니다.

Examples

```
struct eztcp_info out_eztcp_info;
int ret = get_eztcp_info_by_index(0, &out_eztcp_info);

if(ret == EZTCP_SUCCESS)
{
    int i = 0;
    int j = 0;
    char buf[65];
    int uart_count = get_uart_count(out_eztcp_info.mac_address);
    unsigned char *mac_address = out_eztcp_info.mac_address;
    for(i=0;i<uart_count;i++)
    {
        memset(buf, 0x00, 65);

        int sep_count = get_uart_separator_length(mac_address, i);
        for(j=0;j<sep_count;j++)
        {
            get_uart_separator(mac_address, i, j, buf);
            printf("UART[%d] Separator[%d] - %s\r\n", i, j, buf);
        }
    }
}
```



```
}
}
```

See also

get_uart_communication_mode, get_uart_peer_address, get_uart_peer_port, get_uart_local_port,
get_uart_cod_tcp_server, get_uart_watermark, get_uart_timeout, get_uart_data_frame_interval,
is_uart_separator, get_uart_separator_length, get_uart_separator_type, get_uart_tcp_nodelay,
get_uart_rfc2217, get_uart_protocol

Remarks

4.3.12 get_uart_tcp_nodelay

```
int get_uart_tcp_nodelay(unsigned char *mac_address, int uart_index);
```

Description

*get_uart_tcp_nodelay*는 *uart_index*로 지정된 시리얼 포트의 [전송지연 기능 사용안함] 기능의 선택 여부를 반환합니다.

Parameters

*unsigned char *mac_address*

EzTCP_Search 또는 *EzTCP_Read*를 사용하여 검색된 제품(ezTCP)의 MAC주소 입니다.

int uart_index

제품(ezTCP)의 시리얼 포트 순번으로 0부터 시작합니다.

Return values

*get_uart_tcp_nodelay*는 *uart_index*로 지정된 시리얼 포트가 [전송지연 기능 사용안함] 기능을 사용 중이면 1을 반환하고 그렇지 않으면 0을 반환합니다.

*mac_address*을 가진 제품(ezTCP)을 찾을 수 없는 경우에는 -1을 반환합니다.

Examples

See also

get_uart_communication_mode, get_uart_peer_address, get_uart_peer_port, get_uart_local_port,
get_uart_cod_tcp_server, get_uart_watermark, get_uart_timeout, get_uart_data_frame_interval,
is_uart_separator, get_uart_separator_type, get_uart_separator, is_uart_tcp_nodelay, get_uart_rfc2217,
get_uart_protocol

Remarks

4.3.13 get_uart_rfc2217

```
int get_uart_rfc2217(unsigned char *mac_address, int uart_index);
```

Description

*get_uart_rfc2217*은 *uart_index*로 지정된 시리얼 포트의 [시리얼 포트 설정/상태 전송 (RFC2217)] 기능의 선택 여부를 반환합니다.

Parameters

*unsigned char *mac_address*

EzTCP_Search 또는 *EzTCP_Read*를 사용하여 검색된 제품(ezTCP)의 MAC주소 입니다.

int uart_index

제품(ezTCP)의 시리얼 포트 순번으로 0부터 시작합니다.

Return values

*get_uart_rfc2217*은 *uart_index*로 지정된 시리얼 포트가 [시리얼 포트 설정/상태 전송 (RFC2217)] 기능을 사용 중 이면 1을 반환하고 그렇지 않으면 0을 반환합니다.

*mac_address*을 가진 제품(ezTCP)을 찾을 수 없는 경우에는 -1을 반환합니다.

Examples

See also

get_uart_communication_mode, *get_uart_peer_address*, *get_uart_peer_port*, *get_uart_local_port*, *get_uart_cod_tcp_server*, *get_uart_watermark*, *get_uart_timeout*, *get_uart_data_frame_interval*, *is_uart_separator*, *get_uart_separator_type*, *get_uart_separator*, *get_uart_tcp_nodelay*, *is_uart_rfc2217*, *get_uart_protocol*

Remarks

4.3.14 get_uart_protocol

```
int get_uart_protocol(unsigned char *mac_address, int uart_index);
```

Description

*get_uart_protocol*은 *uart_index*로 지정된 시리얼 포트가 사용 중인 [프로토콜]을 반환합니다.

Parameters

*unsigned char *mac_address*

EzTCP_Search 또는 *EzTCP_Read*를 사용하여 검색된 제품(ezTCP)의 MAC주소 입니다.

int uart_index

제품(ezTCP)의 시리얼 포트 순번으로 0부터 시작합니다.

Return values

*get_uart_protocol*은 *uart_index*로 지정된 시리얼 포트가 사용 중인 [프로토콜]을 반환하며 그 값은 아래와 같습니다.

0

TCP

1

TCP + TELNET

2

TCP + SSL

*mac_address*을 가진 제품(ezTCP)을 찾을 수 없는 경우에는 -1을 반환합니다.

Examples

See also

get_uart_communication_mode, *get_uart_peer_address*, *get_uart_peer_port*, *get_uart_local_port*, *get_uart_cod_tcp_server*, *get_uart_watermark*, *get_uart_timeout*, *get_uart_data_frame_interval*, *is_uart_separator*, *get_uart_separator_type*, *get_uart_separator*, *get_uart_tcp_nodelay*, *get_uart_rfc2217*, *is_uart_protocol*.

Remarks

현재 CSE-T16, CSE-T32, CSE-T48만 사용 가능하며 추후 지원가능 모델은 추가될 수 있습니다. 보다 자세한 내용은 해당 제품 사용설명서를 참조하시기 바랍니다.

4.4 CSC-HR2

4.4.1 get_csc_hr2_communication_mode

```
int get_csc_hr2_communication_mode (unsigned char *mac_address);
```

Description

*get_csc_hr2_communication_mode*는 CSC-HR2의 [통신모드]를 반환합니다.

Parameters

*unsigned char *mac_address*

EzTCP_Search 또는 *EzTCP_Read*를 사용하여 검색된 제품(ezTCP)의 MAC주소 입니다.

Return values

*get_csc_hr2_communication_mode*는 CSC-HR2의 [통신모드]를 반환하며 그 값은 아래와 같습니다.

0

자동

1

EZU-100 펌웨어 변경

*mac_address*을 가진 제품(ezTCP)을 찾을 수 없는 경우에는 -1을 반환합니다.

Examples**See also**

is_csc_hr2, *get_csc_hr2_id*, *get_csc_hr2_network_timeout*, *get_csc_hr2_network_threshold*,
get_csc_hr2_server_timeout, *get_csc_hr2_server_threshold*, *get_csc_hr2_check_port*,
get_csc_hr2_first_server_address, *get_csc_hr2_first_server_port*, *get_csc_hr2_second_server_address*,
get_csc_hr2_second_server_port.

Remarks**4.4.2 get_csc_hr2_id**

```
int get_csc_hr2_id(unsigned char *mac_address, char *out_id);
```

Description

*get_csc_hr2_id*는 CSC-HR2의 [CSC-HR2 아이디]를 반환합니다.

Parameters

*unsigned char *mac_address*

EzTCP_Search 또는 *EzTCP_Read*를 사용하여 검색된 제품(ezTCP)의 MAC주소 입니다.

*char *out_addr*

[CSC-HR2 아이디]가 저장될 포인터입니다.

Return values

*get_csc_hr2_id*는 *out_id*에 [CSC-HR2 아이디]를 저장하는데 성공하면 1을 반환합니다.

*mac_address*을 가진 제품(ezTCP)을 찾을 수 없는 경우에는 -1을 반환합니다.

Examples

```
struct eztcp_info out_eztcp_info;  
int ret = get_eztcp_info_by_index(0, &out_eztcp_info);
```

```

if(ret == EZTCP_SUCCESS)
{
    char csc_hr2_id[17];
    if(is_csc_hr2(out_eztcp_info.mac_address) == 1)
    {
        memset(csc_hr2_id, 0x00, 17);
        get_csc_hr2_id(out_eztcp_info.mac_address, csc_hr2_id);
        printf("ID of CSC-HR2 : %s\r\n", csc_hr2_id);
    }
}

```

See also

is_csc_hr2, get_csc_hr2_communication_mode, get_csc_hr2_network_timeout,
get_csc_hr2_network_threshold, get_csc_hr2_server_timeout, get_csc_hr2_server_threshold,
get_csc_hr2_check_port, get_csc_hr2_first_server_address, get_csc_hr2_first_server_port,
get_csc_hr2_second_server_address, get_csc_hr2_second_server_port.

Remarks

[CSC-HR2 아이디]는 NULL을 제외하고 최대 16바이트입니다. 따라서 *out_id*은 최소 17바이트의 메모리 공간을 가지고 있어야 합니다.

4.4.3 get_csc_hr2_network_timeout

```
int get_csc_hr2_network_timeout(unsigned char *mac_address, char *out_timeout);
```

Description

*get_csc_hr2_network_timeout*은 CSC-HR2의 [절체 타임아웃]을 반환합니다.

Parameters

*unsigned char *mac_address*

EzTCP_Search 또는 *EzTCP_Read*를 사용하여 검색된 제품(ezTCP)의 MAC주소 입니다.

*char *out_timeout*

[절체 타임아웃]이 저장될 포인터입니다.

Return values

*get_csc_hr2_network_timeout*은 *out_timeout*에 [절체 타임아웃]을 저장하는데 성공하면 1을 반환합니다.

*mac_address*을 가진 제품(ezTCP)을 찾을 수 없는 경우에는 -1을 반환합니다.

Examples

```
struct eztcp_info out_eztcp_info;
int ret = get_eztcp_info_by_index(0, &out_eztcp_info);

if(ret == EZTCP_SUCCESS)
{
    char buf[64];
    if(is_csc_hr2(out_eztcp_info.mac_address) == 1)
    {
        memset(buf, 0x00, 64);
        get_csc_hr2_network_timeout(out_eztcp_info.mac_address, buf);
        printf("[CSC-HR2] Network Timeout : %s\r\n", buf);
    }
}
```

See also

is_csc_hr2, get_csc_hr2_communication_mode, get_csc_hr2_id, get_csc_hr2_network_threshold,
get_csc_hr2_server_timeout, get_csc_hr2_server_threshold, get_csc_hr2_check_port,
get_csc_hr2_first_server_address, get_csc_hr2_first_server_port, get_csc_hr2_second_server_address,
get_csc_hr2_second_server_port.

Remarks

4.4.4 get_csc_hr2_network_threshold

```
int get_csc_hr2_network_threshold(unsigned char *mac_address, char *out_threshold);
```

Description

*get_csc_hr2_network_threshold*는 CSC-HR2의 [절체 바이트 수]를 반환합니다.

Parameters

*unsigned char *mac_address*

EzTCP_Search 또는 *EzTCP_Read*를 사용하여 검색된 제품(ezTCP)의 MAC주소 입니다.

*char *out_threshold*

[절체 바이트 수]가 저장될 포인터입니다.

Return values

*get_csc_hr2_network_threshold*는 *out_threshold*에 [절체 바이트 수]를 저장하는데 성공하면 1을 반환합니다.

*mac_address*을 가진 제품(ezTCP)을 찾을 수 없는 경우에는 -1을 반환합니다.

Examples

```
struct eztcp_info out_eztcp_info;
int ret = get_eztcp_info_by_index(0, &out_eztcp_info);

if(ret == EZTCP_SUCCESS)
{
    char buf[64];
    if(is_csc_hr2(out_eztcp_info.mac_address) == 1)
    {
        memset(buf, 0x00, 64);
        get_csc_hr2_network_threshold(out_eztcp_info.mac_address, buf);
        printf("[CSC-HR2] Network Threshold : %s\r\n", buf);
    }
}
```

See also

is_csc_hr2, get_csc_hr2_communication_mode, get_csc_hr2_id, get_csc_hr2_network_timeout,
get_csc_hr2_server_timeout, get_csc_hr2_server_threshold, get_csc_hr2_check_port,
get_csc_hr2_first_server_address, get_csc_hr2_first_server_port, get_csc_hr2_second_server_address,
get_csc_hr2_second_server_port.

Remarks

4.4.5 get_csc_hr2_server_timeout

```
int get_csc_hr2_server_timeout(unsigned char *mac_address, char *out_timeout);
```

Description

*get_csc_hr2_server_timeout*은 CSC-HR2의 [서버변경 타임아웃]을 반환합니다.

Parameters

*unsigned char *mac_address*

EzTCP_Search 또는 *EzTCP_Read*를 사용하여 검색된 제품(ezTCP)의 MAC주소 입니다.

*char *out_timeout*

[서버변경 타임아웃]이 저장될 포인터입니다.

Return values

*get_csc_hr2_server_timeout*은 *out_timeout*에 [서버변경 타임아웃]을 저장하는데 성공하면 1을 반환합니다.

*mac_address*을 가진 제품(ezTCP)을 찾을 수 없는 경우에는 -1을 반환합니다.

Examples

```
struct eztcp_info out_eztcp_info;
int ret = get_eztcp_info_by_index(0, &out_eztcp_info);

if(ret == EZTCP_SUCCESS)
{
    char buf[64];
    if(is_csc_hr2(out_eztcp_info.mac_address) == 1)
    {
        memset(buf, 0x00, 64);
        get_csc_hr2_server_timeout(out_eztcp_info.mac_address, buf);
        printf("[CSC-HR2] Server Timeout : %s\r\n", buf);
    }
}
```

See also

is_csc_hr2, get_csc_hr2_communication_mode, get_csc_hr2_id, get_csc_hr2_network_timeout,
get_csc_hr2_network_threshold, get_csc_hr2_server_threshold, get_csc_hr2_check_port,
get_csc_hr2_first_server_address, get_csc_hr2_first_server_port, get_csc_hr2_second_server_address,
get_csc_hr2_second_server_port.

Remarks

4.4.6 get_csc_hr2_server_threshold

```
int get_csc_hr2_server_threshold(unsigned char *mac_address, char *out_threshold);
```

Description

*get_csc_hr2_server_threshold*는 CSC-HR2의 [서버변경 바이트 수]를 반환합니다.

Parameters

*unsigned char *mac_address*

EzTCP_Search 또는 *EzTCP_Read*를 사용하여 검색된 제품(ezTCP)의 MAC주소 입니다.

*char *out_threshold*

[서버변경 바이트 수]가 저장될 포인터입니다.

Return values

*get_csc_hr2_server_threshold*는 *out_threshold*에 [서버변경 바이트 수]를 저장하는데 성공하면 1을 반환합니다.

*mac_address*을 가진 제품(ezTCP)을 찾을 수 없는 경우에는 -1을 반환합니다.

Examples

```
struct eztcp_info out_eztcp_info;
int ret = get_eztcp_info_by_index(0, &out_eztcp_info);

if(ret == EZTCP_SUCCESS)
{
    char buf[64];
    if(is_csc_hr2(out_eztcp_info.mac_address) == 1)
    {
        memset(buf, 0x00, 64);
        get_csc_hr2_server_threshold(out_eztcp_info.mac_address, buf);
        printf("[CSC-HR2] Server Threshold : %s\r\n", buf);
    }
}
```

See also

is_csc_hr2, get_csc_hr2_communication_mode, get_csc_hr2_id, get_csc_hr2_network_timeout,
get_csc_hr2_network_threshold, get_csc_hr2_server_timeout, get_csc_hr2_check_port,
get_csc_hr2_first_server_address, get_csc_hr2_first_server_port, get_csc_hr2_second_server_address,
get_csc_hr2_second_server_port.

Remarks

4.4.7 get_csc_hr2_check_port

```
int get_csc_hr2_check_port(unsigned char *mac_address, char *out_check_port);
```

Description

*get_csc_hr2_check_port*는 CSC-HR2의 [통신품질 점검포트]를 반환합니다.

Parameters

*unsigned char *mac_address*

EzTCP_Search 또는 *EzTCP_Read*를 사용하여 검색된 제품(ezTCP)의 MAC주소 입니다.

*char *out_threshold*

[통신품질 점검포트]가 저장될 포인터입니다.

Return values

*get_csc_hr2_check_port*는 *out_check_port*에 [통신품질 점검포트]를 저장하는데 성공하면 1을 반환합니다.

*mac_address*을 가진 제품(ezTCP)을 찾을 수 없는 경우에는 -1을 반환합니다.

Examples

```
struct eztcp_info out_eztcp_info;
int ret = get_eztcp_info_by_index(0, &out_eztcp_info);

if(ret == EZTCP_SUCCESS)
{
    char buf[64];
    if(is_csc_hr2(out_eztcp_info.mac_address) == 1)
    {
        memset(buf, 0x00, 64);
        get_csc_hr2_check_port(out_eztcp_info.mac_address, buf);
        printf("[CSC-HR2] Check Port : %s\r\n", buf);
    }
}
```

See also

is_csc_hr2, get_csc_hr2_communication_mode, get_csc_hr2_id, get_csc_hr2_network_timeout, get_csc_hr2_network_threshold, get_csc_hr2_server_timeout, get_csc_hr2_server_threshold, get_csc_hr2_first_server_address, get_csc_hr2_first_server_port, get_csc_hr2_second_server_address, get_csc_hr2_second_server_port.

Remarks

4.4.8 get_csc_hr2_first_server_address

```
int get_csc_hr2_first_server_address(unsigned char *mac_address, char *out_address);
```

Description

*get_csc_hr2_first_server_address*는 CSC-HR2의 [첫 번째 서버 주소]를 *out_addr*에 저장합니다.

Parameters

*unsigned char *mac_address*

EzTCP_Search 또는 *EzTCP_Read*를 사용하여 검색된 제품(ezTCP)의 MAC주소 입니다.

*char *out_address*

[첫 번째 서버 주소]가 저장될 포인터입니다.

Return values

*get_csc_hr2_first_server_address*는 *out_address*에 [첫 번째 서버 주소]를 저장하는데 성공하면 1을 반환하고 그렇지 않으면 0을 반환합니다.

mac_address을 가진 제품(ezTCP)을 찾을 수 없는 경우에는 -1을 반환합니다.

Examples

```
struct eztcp_info out_eztcp_info;
int ret = get_eztcp_info_by_index(0, &out_eztcp_info);

if(ret == EZTCP_SUCCESS)
{
    char buf[65];
    if(is_csc_hr2(out_eztcp_info.mac_address) == 1)
    {
        memset(buf, 0x00, 65);
        get_csc_hr2_first_server_address(out_eztcp_info.mac_address, buf);
        printf("[CSC-HR2] First Server Address : %s\r\n", buf);
    }
}
```

See also

is_csc_hr2, get_csc_hr2_communication_mode, get_csc_hr2_id, get_csc_hr2_network_timeout,
get_csc_hr2_network_threshold, get_csc_hr2_server_timeout, get_csc_hr2_server_threshold,
get_csc_hr2_check_port, get_csc_hr2_first_server_port, get_csc_hr2_second_server_address,
get_csc_hr2_second_server_port.

Remarks

[첫 번째 서버 주소]는 NULL을 제외하고 최대 64바이트입니다. 따라서 *out_address*은 최소 65바이트의 메모리 공간을 가지고 있어야 합니다.

4.4.9 get_csc_hr2_first_server_port

```
int get_csc_hr2_first_server_port(unsigned char *mac_address, char *out_port);
```

Description

*get_csc_hr2_first_server_port*는 CSC-HR2가 통신할 첫 번째 서버의 [포트번호]를 *out_port*에 저장합니다.

Parameters

*unsigned char *mac_address*

EzTCP_Search 또는 *EzTCP_Read*를 사용하여 검색된 제품(ezTCP)의 MAC주소 입니다.

*char *out_port*

[포트번호]가 저장될 포인터입니다.

Return values

*get_csc_hr2_first_server_port*는 *out_port*에 통신할 첫 번째 서버의 [포트번호]를 저장하는데 성공하면 1을 반환합니다.

*mac_address*을 가진 제품(ezTCP)을 찾을 수 없는 경우에는 -1을 반환합니다.

Examples

```
struct eztcp_info out_eztcp_info;
int ret = get_eztcp_info_by_index(0, &out_eztcp_info);

if(ret == EZTCP_SUCCESS)
{
    char buf[65];
    if(is_csc_hr2(out_eztcp_info.mac_address) == 1)
    {
        memset(buf, 0x00, 65);
        get_csc_hr2_first_server_port(out_eztcp_info.mac_address, buf);
        printf("[CSC-HR2] First Server Port Number : %s\r\n", buf);
    }
}
```

See also

is_csc_hr2, *get_csc_hr2_communication_mode*, *get_csc_hr2_id*, *get_csc_hr2_network_timeout*,
get_csc_hr2_network_threshold, *get_csc_hr2_server_timeout*, *get_csc_hr2_server_threshold*,
get_csc_hr2_check_port, *get_csc_hr2_first_server_address*, *get_csc_hr2_second_server_address*,
get_csc_hr2_second_server_port.

Remarks

4.4.10 get_csc_hr2_second_server_address

```
int get_csc_hr2_second_server_address(unsigned char *mac_address, char *out_address);
```

Description

*get_csc_hr2_second_server_address*는 CSC-HR2의 [두 번째 서버 주소]를 *out_address*에 저장합니다.

Parameters

*unsigned char *mac_address*

EzTCP_Search 또는 *EzTCP_Read*를 사용하여 검색된 제품(ezTCP)의 MAC주소 입니다.

*char *out_address*

[두 번째 서버 주소]가 저장될 포인터입니다.

Return values

*get_csc_hr2_second_server_address*는 *out_address*에 [두 번째 서버 주소]를 저장하는데 성공하면 1을 반환하고 그렇지 않으면 0을 반환합니다.

*mac_address*을 가진 제품(ezTCP)을 찾을 수 없는 경우에는 -1을 반환합니다.

Examples

```
struct eztcp_info out_eztcp_info;
int ret = get_eztcp_info_by_index(0, &out_eztcp_info);

if(ret == EZTCP_SUCCESS)
{
    char buf[65];
    if(is_csc_hr2(out_eztcp_info.mac_address) == 1)
    {
        memset(buf, 0x00, 65);
        get_csc_hr2_second_server_address(out_eztcp_info.mac_address, buf);
        printf("[CSC-HR2] Second Server Address : %s\r\n", buf);
    }
}
```

See also

is_csc_hr2, *get_csc_hr2_communication_mode*, *get_csc_hr2_id*, *get_csc_hr2_network_timeout*,
get_csc_hr2_network_threshold, *get_csc_hr2_server_timeout*, *get_csc_hr2_server_threshold*,
get_csc_hr2_check_port, *get_csc_hr2_first_server_port*, *get_csc_hr2_second_server_port*.

Remarks

[두 번째 서버 주소]는 NULL을 제외하고 최대 64바이트입니다. 따라서 *out_address*은 최소 65바이트의 메모리 공간을 가지고 있어야 합니다.

4.4.11 get_csc_hr2_second_server_port

```
int get_csc_hr2_second_server_port(unsigned char *mac_address, char *out_port);
```

Description

*get_csc_hr2_second_server_port*는 CSC-HR2가 통신할 두 번째 서버의 [포트번호]를 *out_port*에 저장합니다.

Parameters



*unsigned char *mac_address*

EzTCP_Search 또는 *EzTCP_Read*를 사용하여 검색된 제품(ezTCP)의 MAC주소 입니다.

*char *out_port*

[포트번호]가 저장될 포인터입니다.

Return values

*get_csc_hr2_second_server_port*는 *out_port*에 통신할 두 번째 서버의 [포트번호]를 저장하는데 성공하면 1을 반환합니다.

*mac_address*을 가진 제품(ezTCP)을 찾을 수 없는 경우에는 -1을 반환합니다.

Examples

```
struct eztcp_info out_eztcp_info;
int ret = get_eztcp_info_by_index(0, &out_eztcp_info);

if(ret == EZTCP_SUCCESS)
{
    char buf[65];
    if(is_csc_hr2(out_eztcp_info.mac_address) == 1)
    {
        memset(buf, 0x00, 65);
        get_csc_hr2_second_server_port(out_eztcp_info.mac_address, buf);
        printf("[CSC-HR2] Second Server Port Number : %s\r\n", buf);
    }
}
```

See also

is_csc_hr2, *get_csc_hr2_communication_mode*, *get_csc_hr2_id*, *get_csc_hr2_network_timeout*,
get_csc_hr2_network_threshold, *get_csc_hr2_server_timeout*, *get_csc_hr2_server_threshold*,
get_csc_hr2_check_port, *get_csc_hr2_first_server_address*, *get_csc_hr2_second_server_address*.

Remarks

4.5 I/O 제품

4.5.1 get_io_http

```
int get_io_http(unsigned char *mac_address);
```

Description



*get_io_http*는 I/O 제품 [웹(HTTP)] 제어방식 기능 선택 여부를 반환합니다.

Parameters

*unsigned char *mac_address*

EzTCP_Search 또는 *EzTCP_Read*를 사용하여 검색된 제품(ezTCP)의 MAC주소 입니다.

Return values

*get_io_http*는 [웹(HTTP)] 제어방식 기능이 선택되어 있으면 1을 반환하고 그렇지 않은 경우 0을 반환합니다.

*mac_address*을 가진 제품(ezTCP)을 찾을 수 없는 경우에는 -1을 반환합니다.

Examples

See also

is_io, *get_io_http*, *get_io_html_size*

Remarks

4.5.2 get_io_http_port

```
int get_io_http_port(unsigned char *mac_address, char *out_http_port);
```

Description

*get_io_http_port*는 I/O 제품 [웹(HTTP) 포트] 번호를 반환합니다.

Parameters

*unsigned char *mac_address*

EzTCP_Search 또는 *EzTCP_Read*를 사용하여 검색된 제품(ezTCP)의 MAC주소 입니다.

*char *out_http_port*

[웹(HTTP) 포트] 번호가 저장될 포인터입니다.

Return values

*get_io_http_port*는 *out_http_port*에 [웹(HTTP) 포트] 번호를 저장하는데 성공하면 1을 반환합니다.

*mac_address*을 가진 제품(ezTCP)을 찾을 수 없는 경우에는 -1을 반환합니다.

Examples

```
struct eztcp_info out_eztcp_info;
int ret = get_eztcp_info_by_index(0, &out_eztcp_info);
```

```

if(ret == EZTCP_SUCCESS)
{
    char buf[8];
    if(is_io(out_eztcp_info.mac_address) == 1)
    {
        memset(buf, 0x00, 8);
        get_io_http_port(out_eztcp_info.mac_address, buf);
        printf("[I/O] Web(HTTP) Port : %s\r\n", buf);
    }
}

```

See also

is_io, get_io_http, get_io_html_size

Remarks**4.5.3 get_io_html_size**

```
int get_io_html_size(unsigned char *mac_address);
```

Description

*get_io_html_size*는 I/O 제품 [웹(HTTP)페이지 크기]를 반환합니다.

Parameters

*unsigned char *mac_address*

EzTCP_Search 또는 *EzTCP_Read*를 사용하여 검색된 제품(ezTCP)의 MAC주소 입니다.

Return values

*get_io_html_size*는 [웹(HTTP)페이지 크기]를 반환하며 그 값은 아래와 같습니다.

0

80KB

1

96KB

2

112KB

*mac_address*을 가진 제품(ezTCP)을 찾을 수 없는 경우에는 -1을 반환합니다.

Examples


```

struct eztcp_info out_eztcp_info;
int ret = get_eztcp_info_by_index(0, &out_eztcp_info);

if(ret == EZTCP_SUCCESS)
{
    if(is_io(out_eztcp_info.mac_address) == 1)
    {
        int html_size = get_io_html_size(out_eztcp_info.mac_address);
        printf("[I/O] Size of web(HTTP) : ");
        switch(html_size)
        {
            case 0:
                printf("80KB\r\n");
                break;
            case 1:
                printf("96KB\r\n");
                break;
            case 2:
                printf("112KB\r\n");
                break;
        }
    }
}

```

See also

is_io, get_io_http, get_io_http_port.

Remarks**4.5.4 get_io_modbus**

```
int get_io_modbus(unsigned char *mac_address);
```

Description

*get_io_modbus*는 I/O 제품 [Modbus/TCP] 제어방식 기능 선택 여부를 반환합니다.

Parameters

*unsigned char *mac_address*

EzTCP_Search 또는 *EzTCP_Read*를 사용하여 검색된 제품(ezTCP)의 MAC주소 입니다.

Return values

*get_io_modbus*는 [Modbus/TCP] 제어방식 기능이 선택되어 있으면 1을 반환하고 그렇지 않은 경우 0을 반환합니다.

*mac_address*을 가진 제품(ezTCP)을 찾을 수 없는 경우에는 -1을 반환합니다.

Examples**See also**

is_io, *get_io_modbus*, *get_io_input_notification*, *get_io_output_automatic_initialize*, *get_io_modbus_type*, *get_io_unit_id*, *get_io_input_address*, *get_io_output_address*, *get_io_poll_interval*, *get_io_slave_control_type*, *get_io_master_control_type*, *get_io_tcp_type*, *get_io_multi_connection_number*, *get_io_modbus_peer_address*, *get_io_modbus_peer_port*, *get_io_modbus_local_port*.

Remarks**4.5.5 get_io_input_notification**

```
int get_io_input_notification(unsigned char *mac_address);
```

Description

*get_io_input_notification*은 I/O 제품의 [입력포트 변경 알림] 기능 선택 여부를 반환합니다.

Parameters

*unsigned char *mac_address*

EzTCP_Search 또는 *EzTCP_Read*를 사용하여 검색된 제품(ezTCP)의 MAC주소 입니다.

Return values

*get_io_input_notification*은 [입력포트 변경 알림] 기능이 선택되어 있으면 1을 반환하고 그렇지 않은 경우 0을 반환합니다.

*mac_address*을 가진 제품(ezTCP)을 찾을 수 없는 경우에는 -1을 반환합니다.

Examples**See also**

is_io, *get_io_modbus*, *get_io_input_notification*, *get_io_output_automatic_initialize*, *get_io_modbus_type*, *get_io_unit_id*, *get_io_input_address*, *get_io_output_address*, *get_io_poll_interval*, *get_io_slave_control_type*, *get_io_master_control_type*, *get_io_tcp_type*, *get_io_multi_connection_number*, *get_io_modbus_peer_address*, *get_io_modbus_peer_port*, *get_io_modbus_local_port*.

Remarks

4.5.6 get_io_output_automatic_initialize

```
int get_io_output_automatic_initialize(unsigned char *mac_address);
```

Description

*get_io_output_automatic_initialize*는 I/O 제품의 [출력포트 상태 초기화(Modbus/TCP 접속 종료 시 출력포트[초기상태] 값으로 상태 변경) 기능 선택 여부를 반환합니다.

Parameters

*unsigned char *mac_address*

EzTCP_Search 또는 *EzTCP_Read*를 사용하여 검색된 제품(ezTCP)의 MAC주소 입니다.

Return values

*get_io_output_automatic_initialize*은 [출력포트 상태 초기화(Modbus/TCP 접속 종료 시 출력포트[초기상태] 값으로 상태 변경) 기능이 선택되어 있으면 1을 반환하고 그렇지 않은 경우 0을 반환합니다.

*mac_address*을 가진 제품(ezTCP)을 찾을 수 없는 경우에는 -1을 반환합니다.

Examples

See also

is_io, *get_io_modbus*, *get_io_input_notification*, *is_io_output_automatic_initialize*,
get_io_output_automatic_initialize, *get_io_modbus_type*, *get_io_unit_id*, *get_io_input_address*,
get_io_output_address, *get_io_poll_interval*, *get_io_slave_control_type*, *get_io_master_control_type*,
get_io_tcp_type, *get_io_multi_connection_number*, *get_io_modbus_peer_address*,
get_io_modbus_peer_port, *get_io_modbus_local_port*.

Remarks

보다 자세한 내용은 해당 제품 사용설명서를 참조하시기 바랍니다.

4.5.7 get_io_modbus_type

```
int get_io_modbus_type(unsigned char *mac_address);
```

Description

*get_io_modbus_type*은 I/O 제품의 [마스터/슬레이브]를 반환합니다.

Parameters

*unsigned char *mac_address*

EzTCP_Search 또는 *EzTCP_Read*를 사용하여 검색된 제품(ezTCP)의 MAC주소 입니다.

Return values

*get_io_modbus_type*은 [마스터/슬레이브]를 반환하며 그 값은 아래와 같습니다.

0

슬레이브

1

마스터

*mac_address*을 가진 제품(ezTCP)을 찾을 수 없는 경우에는 -1을 반환합니다.

Examples

```
struct eztcp_info out_eztcp_info;
int ret = get_eztcp_info_by_index(0, &out_eztcp_info);

if(ret == EZTCP_SUCCESS)
{
    if(is_io(out_eztcp_info.mac_address) == 1)
    {
        int Modbus_type = get_io_modbus_type(out_eztcp_info.mac_address);
        switch(Modbus_type)
        {
            case 0:
                printf("[I/O] This is Modbus slave.\r\n");
                break;
            case 1:
                printf("[I/O] This is Modbus master.\r\n");
                break;
        }
    }
}
```

See also

is_io, *get_io_modbus*, *get_io_input_notification*, *get_io_output_automatic_initialize*, *get_io_modbus_type*, *get_io_unit_id*, *get_io_input_address*, *get_io_output_address*, *get_io_poll_interval*, *get_io_slave_control_type*, *get_io_master_control_type*, *get_io_tcp_type*, *get_io_multi_connection_number*, *get_io_modbus_peer_address*, *get_io_modbus_peer_port*, *get_io_modbus_local_port*.

Remarks

4.5.8 get_io_unit_id

```
int get_io_unit_id(unsigned char *mac_address, char *out_unit_id);
```

Description

*get_io_unit_id*는 I/O 제품의 [유니트 아이디]를 반환합니다.

Parameters

*unsigned char *mac_address*

EzTCP_Search 또는 *EzTCP_Read*를 사용하여 검색된 제품(ezTCP)의 MAC주소 입니다.

*char *out_unit_id*

[유니트 아이디]가 저장될 포인터입니다.

Return values

*get_io_unit_id*는 *out_unit_id*에 [유니트 아이디]를 저장하는데 성공하면 1을 반환합니다.

*mac_address*을 가진 제품(ezTCP)을 찾을 수 없는 경우에는 -1을 반환합니다.

Examples

```
struct eztcp_info out_eztcp_info;
int ret = get_eztcp_info_by_index(0, &out_eztcp_info);

if(ret == EZTCP_SUCCESS)
{
    char buf[8];
    if(is_io(out_eztcp_info.mac_address) == 1)
    {
        memset(buf, 0x00, 8);
        get_io_unit_id(out_eztcp_info.mac_address, buf);
        printf("[I/O] Unit ID : %s\r\n", buf);
    }
}
```

See also

is_io, *get_io_modbus*, *get_io_input_notification*, *get_io_output_automatic_initialize*, *get_io_modbus_type*, *get_io_unit_id*, *get_io_input_address*, *get_io_output_address*, *get_io_poll_interval*, *get_io_slave_control_type*, *get_io_master_control_type*, *get_io_tcp_type*, *get_io_multi_connection_number*, *get_io_modbus_peer_address*, *get_io_modbus_peer_port*, *get_io_modbus_local_port*.

Remarks

4.5.9 get_io_input_address

```
int get_io_input_address(unsigned char *mac_address, char *out_address);
```

Description

*get_io_input_address*는 I/O 제품의 [입력포트 주소]를 반환합니다.

Parameters

*unsigned char *mac_address*

EzTCP_Search 또는 *EzTCP_Read*를 사용하여 검색된 제품(ezTCP)의 MAC주소 입니다.

*char *out_address*

[입력포트 주소]가 저장될 포인터입니다.

Return values

*get_io_input_address*는 *out_address*에 [입력포트 주소]를 저장하는데 성공하면 1을 반환합니다.

*mac_address*을 가진 제품(ezTCP)을 찾을 수 없는 경우에는 -1을 반환합니다.

Examples

```
struct eztcp_info out_eztcp_info;
int ret = get_eztcp_info_by_index(0, &out_eztcp_info);

if(ret == EZTCP_SUCCESS)
{
    char buf[8];
    if(is_io(out_eztcp_info.mac_address) == 1)
    {
        memset(buf, 0x00, 8);
        get_io_input_address(out_eztcp_info.mac_address, buf);
        printf("[I/O] Input Port Base Address : %s\r\n", buf);
    }
}
```

See also

is_io, *get_io_modbus*, *get_io_input_notification*, *get_io_output_automatic_initialize*, *get_io_modbus_type*, *get_io_unit_id*, *get_io_input_address*, *get_io_output_address*, *get_io_poll_interval*, *get_io_slave_control_type*, *get_io_master_control_type*, *get_io_tcp_type*, *get_io_multi_connection_number*, *get_io_modbus_peer_address*, *get_io_modbus_peer_port*, *get_io_modbus_local_port*.

Remarks

4.5.10 get_io_output_address

```
int get_io_output_address(unsigned char *mac_address, char *out_address);
```

Description

*get_io_output_address*는 I/O 제품의 [출력포트 주소]를 반환합니다.

Parameters

*unsigned char *mac_address*

EzTCP_Search 또는 *EzTCP_Read*를 사용하여 검색된 제품(ezTCP)의 MAC주소 입니다.

*char *out_address*

[출력포트 주소]가 저장될 포인터입니다.

Return values

*get_io_output_address*는 *out_address*에 [출력포트 주소]를 저장하는데 성공하면 1을 반환합니다.

*mac_address*을 가진 제품(ezTCP)을 찾을 수 없는 경우에는 -1을 반환합니다.

Examples

```
struct eztcp_info out_eztcp_info;
int ret = get_eztcp_info_by_index(0, &out_eztcp_info);

if(ret == EZTCP_SUCCESS)
{
    char buf[8];
    if(is_io(out_eztcp_info.mac_address) == 1)
    {
        memset(buf, 0x00, 8);
        get_io_output_address(out_eztcp_info.mac_address, buf);
        printf("[I/O] Output Port Base Address : %s\r\n", buf);
    }
}
```

See also

is_io, *get_io_modbus*, *get_io_input_notification*, *get_io_output_automatic_initialize*, *get_io_modbus_type*, *get_io_unit_id*, *get_io_input_address*, *get_io_output_address*, *get_io_poll_interval*, *get_io_slave_control_type*, *get_io_master_control_type*, *get_io_tcp_type*, *get_io_multi_connection_number*, *get_io_modbus_peer_address*, *get_io_modbus_peer_port*, *get_io_modbus_local_port*.

Remarks

4.5.11 get_io_poll_interval

```
int get_io_poll_interval(unsigned char *mac_address, char *out_poll_interval);
```

Description

*get_io_poll_interval*은 I/O 제품의 마스터 [통신 주기]를 반환합니다.

Parameters

*unsigned char *mac_address*

EzTCP_Search 또는 *EzTCP_Read*를 사용하여 검색된 제품(ezTCP)의 MAC주소 입니다.

*char *out_poll_interval*

[통신 주기]가 저장될 포인터입니다.

Return values

*get_io_poll_interval*은 *out_poll_interval*에 마스터 [통신 주기]를 저장하는데 성공하면 1을 반환합니다.

*mac_address*을 가진 제품(ezTCP)을 찾을 수 없는 경우에는 -1을 반환합니다.

Examples

```
struct eztcp_info out_eztcp_info;
int ret = get_eztcp_info_by_index(0, &out_eztcp_info);

if(ret == EZTCP_SUCCESS)
{
    char buf[8];
    if(is_io(out_eztcp_info.mac_address) == 1)
    {
        memset(buf, 0x00, 8);
        get_io_poll_interval(out_eztcp_info.mac_address, buf);
        printf("[I/O] Poll Interval : %s\r\n", buf);
    }
}
```

See also

is_io, *get_io_modbus*, *get_io_input_notification*, *get_io_output_automatic_initialize*, *get_io_modbus_type*, *get_io_unit_id*, *get_io_input_address*, *get_io_output_address*, *get_io_poll_interval*, *get_io_slave_control_type*, *get_io_master_control_type*, *get_io_tcp_type*, *get_io_multi_connection_number*, *get_io_modbus_peer_address*, *get_io_modbus_peer_port*, *get_io_modbus_local_port*.

Remarks

4.5.12 get_io_slave_control_type

```
int get_io_slave_control_type(unsigned char *mac_address);
```


Description

*get_io_slave_control_type*은 I/O 제품의 [슬레이브 출력포트 제어방식]을 반환합니다.

Parameters

*unsigned char *mac_address*

EzTCP_Search 또는 *EzTCP_Read*를 사용하여 검색된 제품(ezTCP)의 MAC주소 입니다.

Return values

*get_io_slave_control_type*은 I/O 제품의 [슬레이브 출력포트 제어방식]을 반환하며 그 값은 아래와 같습니다.

0

FC 16 (동시제어)

1

FC 05 (개별제어)

*mac_address*을 가진 제품(ezTCP)을 찾을 수 없는 경우에는 -1을 반환합니다.

Examples

```
struct eztcp_info out_eztcp_info;
int ret = get_eztcp_info_by_index(0, &out_eztcp_info);

if(ret == EZTCP_SUCCESS)
{
    if(is_io(out_eztcp_info.mac_address) == 1)
    {
        int ctrl_type = get_io_slave_control_type(out_eztcp_info.mac_address);
        switch(ctrl_type)
        {
            case 0:
                printf("[I/O] FC 16(Multiple).\r\n");
                break;

            case 1:
                printf("[I/O] FC 05(Single).\r\n");
                break;

        }
    }
}
```

See also

is_io, *get_io_modbus*, *get_io_input_notification*, *get_io_output_automatic_initialize*, *get_io_modbus_type*,
get_io_unit_id, *get_io_input_address*, *get_io_output_address*, *get_io_poll_interval*,

`get_io_slave_control_type`, `get_io_master_control_type`, `get_io_tcp_type`, `get_io_multi_connection_number`, `get_io_modbus_peer_address`, `get_io_modbus_peer_port`, `get_io_modbus_local_port`.

Remarks

보다 자세한 내용은 제품 사용설명서를 참조하시기 바랍니다.

4.5.13 `get_io_master_control_type`

```
int get_io_master_control_type(unsigned char *mac_address);
```

Description

`get_io_master_control_type`은 I/O 제품의 [마스터 출력포트 제어방식]을 반환합니다.

Parameters

*unsigned char *mac_address*

EzTCP_Search 또는 *EzTCP_Read*를 사용하여 검색된 제품(ezTCP)의 MAC주소 입니다.

Return values

`get_io_master_control_type`은 I/O 제품의 [마스터 출력포트 제어방식]을 반환하며 그 값은 아래와 같습니다.

0

AND

1

OR

*mac_address*을 가진 제품(ezTCP)을 찾을 수 없는 경우에는 -1을 반환합니다.

Examples

```
struct eztcp_info out_eztcp_info;
int ret = get_eztcp_info_by_index(0, &out_eztcp_info);

if(ret == EZTCP_SUCCESS)
{
    if(is_io(out_eztcp_info.mac_address) == 1)
    {
        int ctrl_type = get_io_master_control_type(out_eztcp_info.mac_address);
        switch(ctrl_type)
        {
            case 0:
                printf("[I/O] AND.\r\n");
                break;
```

```

        case 1:
            printf("[I/O] OR.\r\n");
            break;
        }
    }
}

```

See also

is_io, get_io_modbus, get_io_input_notification, get_io_output_automatic_initialize, get_io_modbus_type, get_io_unit_id, get_io_input_address, get_io_output_address, get_io_poll_interval, get_io_slave_control_type, get_io_master_control_type, get_io_tcp_type, get_io_multi_connection_number, get_io_modbus_peer_address, get_io_modbus_peer_port, get_io_modbus_local_port.

Remarks

보다 자세한 내용은 제품 사용설명서를 참조하시기 바랍니다.

4.5.14 get_io_tcp_type

```
int get_io_tcp_type(unsigned char *mac_address);
```

Description

*get_io_tcp_type*은 I/O 제품의 Modbus/TCP 접속 방법을 반환합니다.

Parameters

*unsigned char *mac_address*

EzTCP_Search 또는 *EzTCP_Read*를 사용하여 검색된 제품(ezTCP)의 MAC주소 입니다.

Return values

*get_io_tcp_type*은 I/O 제품의 Modbus/TCP 접속 방법을 반환하며 그 값은 아래와 같습니다.

0

수동접속

1

능동접속

*mac_address*을 가진 제품(ezTCP)을 찾을 수 없는 경우에는 -1을 반환합니다.

Examples

```

struct eztcp_info out_eztcp_info;
int ret = get_eztcp_info_by_index(0, &out_eztcp_info);

if(ret == EZTCP_SUCCESS)

```

```

{
    if(is_io(out_eztcp_info.mac_address) == 1)
    {
        int ctrl_type = get_io_tcp_type(out_eztcp_info.mac_address);
        switch(ctrl_type)
        {
            case 0:
                printf("[I/O] Passive Connection.\r\n");
                break;
            case 1:
                printf("[I/O] Active Connection.\r\n");
                break;
        }
    }
}

```

See also

is_io, get_io_modbus, get_io_input_notification, get_io_output_automatic_initialize, get_io_modbus_type, get_io_unit_id, get_io_input_address, get_io_output_address, get_io_poll_interval, get_io_slave_control_type, get_io_master_control_type, get_io_tcp_type, get_io_multi_connection_number, get_io_modbus_peer_address, get_io_modbus_peer_port, get_io_modbus_local_port.

Remarks

보다 자세한 내용은 제품 사용설명서를 참조하시기 바랍니다.

4.5.15 get_io_multi_connection_number

```
int get_io_multi_connection_number(unsigned char *mac_address);
```

Description

*get_io_multi_connection_number*은 I/O 제품이 Modbus/TCP 수동접속인 경우 동시 접속 가능한 Modbus/TCP 클라이언트의 개수를 반환합니다.

Parameters

*unsigned char *mac_address*

EzTCP_Search 또는 *EzTCP_Read*를 사용하여 검색된 제품(ezTCP)의 MAC주소 입니다.

Return values

*get_io_multi_connection_number*은 I/O 제품이 Modbus/TCP 수동접속인 경우 동시 접속 가능한 Modbus/TCP 클라이언트의 개수를 반환하며 그 값은 1 ~ 8입니다.

*mac_address*을 가진 제품(ezTCP)을 찾을 수 없는 경우에는 -1을 반환합니다.

Examples

See also

is_io, get_io_modbus, get_io_input_notification, get_io_output_automatic_initialize, get_io_modbus_type, get_io_unit_id, get_io_input_address, get_io_output_address, get_io_poll_interval, get_io_slave_control_type, get_io_master_control_type, get_io_tcp_type, get_io_multi_connection_number, get_io_modbus_peer_address, get_io_modbus_peer_port, get_io_modbus_local_port.

Remarks

4.5.16 get_io_modbus_peer_address

```
int get_io_modbus_peer_address(unsigned char *mac_address, char *out_address);
```

Description

*get_io_modbus_peer_address*는 I/O 제품이 Modbus/TCP 능동접속인 경우 [통신할 주소]를 반환합니다.

Parameters

*unsigned char *mac_address*

EzTCP_Search 또는 *EzTCP_Read*를 사용하여 검색된 제품(ezTCP)의 MAC주소 입니다.

*char *out_address*

[통신할 주소]가 저장될 포인터입니다.

Return values

*get_io_modbus_peer_address*는 *out_address*에 [통신할 주소]를 저장하는데 성공하면 1을 반환하고 그렇지 않으면 0을 반환합니다.

*mac_address*을 가진 제품(ezTCP)을 찾을 수 없는 경우에는 -1을 반환합니다.

Examples

```
struct eztcp_info out_eztcp_info;
int ret = get_eztcp_info_by_index(0, &out_eztcp_info);

if(ret == EZTCP_SUCCESS)
{
    char buf[65];
    unsigned char *mac_address = out_eztcp_info.mac_address;
    if(is_io(mac_address) == 1 && get_io_tcp_type(mac_address) == 1)
    {
        memset(buf, 0x00, 65);
    }
}
```

```

        get_io_modbus_peer_address(out_eztcp_info.mac_address, buf);
        printf("[I/O] Peer Address : %s\r\n", buf);
    }
}

```

See also

is_io, get_io_modbus, get_io_input_notification, get_io_output_automatic_initialize, get_io_modbus_type, get_io_unit_id, get_io_input_address, get_io_output_address, get_io_poll_interval, get_io_slave_control_type, get_io_master_control_type, get_io_tcp_type, get_io_multi_connection_number, get_io_modbus_peer_address, get_io_modbus_peer_port, get_io_modbus_local_port.

Remarks

[통신할 주소]는 NULL을 제외하고 최대 64바이트입니다. 따라서 *out_address*은 최소 65바이트의 메모리 공간을 가지고 있어야 합니다.

4.5.17 get_io_modbus_peer_port

```
int get_io_modbus_peer_port(unsigned char *mac_address, char *out_port);
```

Description

*get io modbus peer port*는 I/O 제품이 Modbus/TCP 능동접속인 경우 [통신할 포트]를 *out_port*에 저장합니다.

Parameters

*unsigned char *mac_address*

EzTCP_Search 또는 *EzTCP_Read*를 사용하여 검색된 제품(ezTCP)의 MAC주소 입니다.

*char *out_port*

[통신할 포트]가 저장될 포인터입니다.

Return values

*get io modbus peer port*는 *out_port*에 [통신할 포트]를 저장하는데 성공하면 1을 반환합니다.

*mac_address*을 가진 제품(ezTCP)을 찾을 수 없는 경우에는 -1을 반환합니다.

Examples

```

struct eztcp_info out_eztcp_info;
int ret = get_eztcp_info_by_index(0, &out_eztcp_info);

if(ret == EZTCP_SUCCESS)
{
    char buf[65];

```

```

    unsigned char *mac_address = out_eztcp_info.mac_address;
    if(is_io(mac_address) == 1 && get_io_tcp_type(mac_address) == 1)
    {
        memset(buf, 0x00, 65);
        get_io_modbus_peer_port(out_eztcp_info.mac_address, buf);
        printf("[I/O] Peer Port : %s\r\n", buf);
    }
}

```

See also

is_io, get_io_modbus, get_io_input_notification, get_io_output_automatic_initialize, get_io_modbus_type, get_io_unit_id, get_io_input_address, get_io_output_address, get_io_poll_interval, get_io_slave_control_type, get_io_master_control_type, get_io_tcp_type, get_io_multi_connection_number, get_io_modbus_peer_address, get_io_modbus_peer_port, get_io_modbus_local_port.

Remarks

4.5.18 get_io_modbus_local_port

```
int get_io_modbus_local_port(unsigned char *mac_address, char *out_port);
```

Description

*get_io_modbus_local_port*는 I/O 제품이 Modbus/TCP 수동접속인 경우 Modbus/TCP [제품 로컬포트]를 out_port에 저장합니다.

Parameters

*unsigned char *mac_address*

EzTCP_Search 또는 EzTCP_Read를 사용하여 검색된 제품(ezTCP)의 MAC주소 입니다.

*char *out_port*

Modbus/TCP [제품 로컬포트]가 저장될 포인터입니다.

Return values

*get_io_modbus_local_port*는 out_port에 Modbus/TCP [제품 로컬포트]를 저장하는데 성공하면 1을 반환합니다.

mac_address을 가진 제품(ezTCP)을 찾을 수 없는 경우에는 -1을 반환합니다.

Examples

```

struct eztcp_info out_eztcp_info;
int ret = get_eztcp_info_by_index(0, &out_eztcp_info);

```

```

if(ret == EZTCP_SUCCESS)
{
    char buf[65];
    unsigned char *mac_address = out_eztcp_info.mac_address;
    if(is_io(mac_address) == 1 && get_io_tcp_type(mac_address) == 0)
    {
        memset(buf, 0x00, 65);
        get_io_modbus_local_port(out_eztcp_info.mac_address, buf);
        printf("[I/O] Local Port : %s\r\n", buf);
    }
}

```

See also

is_io, get_io_modbus, get_io_input_notification, get_io_output_automatic_initialize, get_io_modbus_type, get_io_unit_id, get_io_input_address, get_io_output_address, get_io_poll_interval, get_io_slave_control_type, get_io_master_control_type, get_io_tcp_type, get_io_multi_connection_number, get_io_modbus_peer_address, get_io_modbus_peer_port, get_io_modbus_local_port.

Remarks

4.5.19 get_io_output_number

```
int get_io_output_number(unsigned char *mac_address);
```

Description

*get_io_output_port_number*는 I/O 제품에 설치되어 있는 출력포트 개수를 반환합니다.

Parameters

*unsigned char *mac_address*

EzTCP_Search 또는 *EzTCP_Read*를 사용하여 검색된 제품(ezTCP)의 MAC주소 입니다.

Return values

*get_io_output_port_number*는 I/O 제품에 설치되어 있는 출력포트 개수를 반환합니다.

*mac_address*을 가진 제품(ezTCP)을 찾을 수 없는 경우에는 -1을 반환합니다.

Examples

See also

is_io, get_io_output_number, get_io_input_number, is_event_notification, get_io_event_notification, get_event_notification_email, get_io_event_notification_port, get_io_input_valid_time, get_io_macro_type,

`get_io_macro`, `get_io_port_macro`, `get_io_macro_text`, `get_io_output_delay`, `get_io_output_initial_state`, `get_io_comment`.

Remarks

4.5.20 `get_io_input_number`

```
int get_io_input_number(unsigned char *mac_address);
```

Description

`get_io_input_port_number`는 I/O 제품에 설치되어 있는 입력포트 개수를 반환합니다.

Parameters

*unsigned char *mac_address*

EzTCP_Search 또는 *EzTCP_Read*를 사용하여 검색된 제품(ezTCP)의 MAC주소 입니다.

Return values

`get_io_input_port_number`는 I/O 제품에 설치되어 있는 입력포트 개수를 반환합니다.

`mac_address`을 가진 제품(ezTCP)을 찾을 수 없는 경우에는 -1을 반환합니다.

Examples

See also

`is_io`, `get_io_output_number`, `get_io_input_number`, `is_event_notification`, `get_io_event_notification`, `get_event_notification_email`, `get_io_event_notification_port`, `get_io_input_valid_time`, `get_io_macro_type`, `get_io_macro`, `get_io_port_macro`, `get_io_macro_text`, `get_io_output_delay`, `get_io_output_initial_state`, `get_io_comment`.

Remarks

4.5.21 `get_io_event_notification`

```
int get_io_event_notification(unsigned char *mac_address);
```

Description

`get_io_event_notification`은 I/O 제품의 [입력 또는 출력포트 변경 알림(전자메일)] 기능 선택 여부를 반환합니다.

Parameters

*unsigned char *mac_address*

EzTCP_Search 또는 *EzTCP_Read*를 사용하여 검색된 제품(ezTCP)의 MAC주소 입니다.

Return values

*get_io_event_notification*는 [입력 또는 출력포트 변경 알림(전자메일)] 기능이 선택되어 있으면 1을 반환하고 그렇지 않은 경우 0을 반환합니다.

*mac_address*을 가진 제품(ezTCP)을 찾을 수 없는 경우에는 -1을 반환합니다.

Examples

See also

is_io, *get_io_output_number*, *get_io_input_number*, *is_event_notification*, *get_io_event_notification*, *get_event_notification_email*, *get_io_event_notification_port*, *get_io_input_valid_time*, *get_io_macro_type*, *get_io_macro*, *get_io_port_macro*, *get_io_macro_text*, *get_io_output_delay*, *get_io_output_initial_state*, *get_io_comment*.

Remarks

4.5.22 get_event_notification_email

```
int get_event_notification_email(unsigned char *mac_address, char *out_email);
```

Description

*get_event_notification_email*은 I/O 제품의 입력 또는 출력포트에 변경이 발생한 경우 알림을 발송할 [전자메일 주소]를 반환합니다.

Parameters

*unsigned char *mac_address*

EzTCP_Search 또는 *EzTCP_Read*를 사용하여 검색된 제품(ezTCP)의 MAC주소 입니다.

*char *out_email*

[전자메일 주소]가 저장될 포인터입니다.

Return values

*get_event_notification_email*은 *out_email*에 [전자메일 주소]를 저장하는데 성공하면 1을 반환합니다.

*mac_address*을 가진 제품(ezTCP)을 찾을 수 없는 경우에는 -1을 반환합니다.

Examples

```
struct eztcp_info out_eztcp_info;
int ret = get_eztcp_info_by_index(0, &out_eztcp_info);
```

```

if(ret == EZTCP_SUCCESS)
{
    char buf[65];
    unsigned char *mac_address = out_eztcp_info.mac_address;
    if(is_io(mac_address) == 1 && is_event_notification(mac_address) == 1)
    {
        if(get_io_event_notification(mac_address) == 1)
        {
            memset(buf, 0x00, 65);
            get_event_notification_email(out_eztcp_info.mac_address, buf);
            printf("[I/O] Email Address : %s\r\n", buf);
        }
    }
}

```

See also

is_io, get_io_output_number, get_io_input_number, is_event_notification, get_io_event_notification, get_event_notification_email, get_io_event_notification_port, get_io_input_valid_time, get_io_macro_type, get_io_macro, get_io_port_macro, get_io_macro_text, get_io_output_delay, get_io_output_initial_state, get_io_comment.

Remarks

[전자메일 주소]는 NULL을 포함하여 최대 64바이트입니다. 따라서 *out_email*은 최소 64바이트의 메모리 공간을 가지고 있어야 합니다.

4.5.23 get_io_event_notification_port

```

int get_io_event_notification_port(unsigned char *mac_address, int port_flag,
                                   int port_index);

```

Description

*get_io_event_notification_port*는 I/O 제품에 설치된 입력 또는 출력포트의 [입력 또는 출력 포트 변경 알림(전자메일)] 기능 선택 여부를 반환합니다.

Parameters

*unsigned char *mac_address*

EzTCP_Search 또는 *EzTCP_Read*를 사용하여 검색된 제품(ezTCP)의 MAC주소 입니다.

int port_flag

*port_flag*가 0이면 입력포트를 의미하고 1이면 출력포트를 의미합니다.

int port_index

입력 또는 출력포트의 순번으로 0부터 시작합니다.

Return values

*get_io_event_notification_port*는 *port flag*와 *port index*가 가리키는 입력 또는 출력포트에 [입력 또는 출력포트 변경 알림(전자메일)] 기능이 선택되어 있으면 1을 반환하고 그렇지 않으면 0을 반환합니다.

*mac_address*을 가진 제품(ezTCP)을 찾을 수 없는 경우에는 -1을 반환합니다.

Examples

```
struct eztcp_info out_eztcp_info;
int ret = get_eztcp_info_by_index(0, &out_eztcp_info);

if(ret == EZTCP_SUCCESS)
{
    int idx = 0;
    unsigned char *mac_address = out_eztcp_info.mac_address;
    if(is_io(mac_address) == 1 && is_event_notification(mac_address) == 1)
    {
        if(get_io_event_notification(mac_address) == 1)
        {
            int input_number = get_io_input_number(mac_address);
            int output_number = get_io_output_number(mac_address);

            for(idx = 0; idx < input_number; idx++)
            {
                if(get_io_event_notification_port(mac_address, 0, idx) == 1)
                {
                    printf("[I/O] Input port#%d is enabled\r\n", idx);
                }
                else
                {
                    printf("[I/O] Input port#%d is disabled\r\n", idx);
                }
            }

            for(idx = 0; idx < output_number; idx++)
            {
                if(get_io_event_notification_port(mac_address, 1, idx) == 1)
                {
                    printf("[I/O] Output port#%d is enabled\r\n", idx);
                }
            }
        }
    }
}
```

```

        else
        {
            printf("[I/O] output port#%d is disabled\r\n", idx);
        }
    }
}
}
}

```

See also

is_io, get_io_output_number, get_io_input_number, is_event_notification, get_io_event_notification, get_event_notification_email, get_io_event_notification_port, get_io_input_valid_time, get_io_macro_type, get_io_macro, get_io_port_macro, get_io_macro_text, get_io_output_delay, get_io_output_initial_state, get_io_comment.

Remarks

4.5.24 get_io_input_valid_time

```

int get_io_input_valid_time(unsigned char *mac_address, int port_index,
                           char *out_input_valid_time);

```

Description

*get_io_input_valid_time*은 *port_index*로 지정된 입력포트의 [신호유지 시간(밀리초)]를 반환합니다.

Parameters

*unsigned char *mac_address*

EzTCP_Search 또는 *EzTCP_Read*를 사용하여 검색된 제품(ezTCP)의 MAC주소 입니다.

int port_index

입력포트의 순번으로 0부터 시작합니다.

*char *out_input_valid_time*

*port_index*로 지정된 입력포트의 [신호유지 시간(밀리초)]가 저장될 포인터입니다.

Return values

*get_io_input_valid_time*은 *out_input_valid_time*에 [신호유지 시간(밀리초)]를 저장하는데 성공하면 1을 반환합니다.

*mac_address*을 가진 제품(ezTCP)을 찾을 수 없는 경우에는 -1을 반환합니다.

Examples

```

struct eztcp_info out_eztcp_info;
int ret = get_eztcp_info_by_index(0, &out_eztcp_info);

if(ret == EZTCP_SUCCESS)
{
    char buf[8];
    int idx = 0;
    int input_number = get_io_input_number(out_eztcp_info.mac_address);
    if(is_io(out_eztcp_info.mac_address) == 1)
    {
        for(idx = 0; idx < input_number; idx++)
        {
            memset(buf, 0x00, 8);
            get_io_input_valid_time(out_eztcp_info.mac_address, idx, buf);
            printf("[I/O] Input Port#%d Valid Time(ms) : %s\r\n", idx, buf);
        }
    }
}

```

See also

is_io, get_io_output_number, get_io_input_number, is_event_notification, get_io_event_notification, get_event_notification_email, get_io_event_notification_port, get_io_input_valid_time, get_io_macro_type, get_io_macro, get_io_port_macro, get_io_macro_text, get_io_output_delay, get_io_output_initial_state, get_io_comment.

Remarks

4.5.25 get_io_macro_type

```
int get_io_macro_type(unsigned char *mac_address);
```

Description

*get_io_macro_type*은 I/O 제품이 지원하는 매크로 기능의 종류를 반환합니다.

Parameters

*unsigned char *mac_address*

EzTCP_Search 또는 *EzTCP_Read*를 사용하여 검색된 제품(ezTCP)의 MAC주소 입니다.

Return values

*get_io_macro_type*은 I/O 제품에 설치된 출력포트에 매크로 기능을 각각 선택할 수 있으면

1을 반환하고 그렇지 않으면 0을 반환합니다.

*get_io_macro_type*이 0을 반환하는 I/O 제품은 매크로 기능을 선택하면 모든 출력포트가 매크로 기능을 사용하게 됩니다.

*mac_address*을 가진 제품(ezTCP)을 찾을 수 없는 경우에는 -1을 반환합니다.

Examples

See also

is_io, *get_io_output_number*, *get_io_input_number*, *is_event_notification*, *get_io_event_notification*, *get_event_notification_email*, *get_io_event_notification_port*, *get_io_input_valid_time*, *get_io_macro_type*, *get_io_macro*, *get_io_port_macro*, *get_io_macro_text*, *get_io_output_delay*, *get_io_output_initial_state*, *get_io_comment*.

Remarks

4.5.26 get_io_macro

```
int get_io_macro(unsigned char *mac_address);
```

Description

*get_io_macro*는 I/O 제품의 [매크로 기능] 선택 여부를 반환합니다. *get_io_macro_type*이 0을 반환하는 경우에만 사용할 수 있습니다.

Parameters

*unsigned char *mac_address*

EzTCP_Search 또는 *EzTCP_Read*를 사용하여 검색된 제품(ezTCP)의 MAC주소 입니다.

Return values

*get_io_macro*는 I/O 제품이 [매크로 기능]이 선택되어 있으면 1을 반환하고 그렇지 않은 경우 0을 반환합니다.

*mac_address*을 가진 제품(ezTCP)을 찾을 수 없는 경우에는 -1을 반환합니다.

Examples

```
struct eztcp_info out_eztcp_info;
int ret = get_eztcp_info_by_index(0, &out_eztcp_info);

if(ret == EZTCP_SUCCESS)
{
    if(is_io(out_eztcp_info.mac_address) == 1)
    {
        if(get_io_macro_type(out_eztcp_info.mac_address) == 0)
```

```

        {
            if(get_io_macro(out_eztcp_info.mac_address) == 1)
                printf("[I/O] Macro is enabled.\r\n");
            else
                printf("[I/O] Macro is disabled.\r\n");
        }
    }
}

```

See also

is_io, get_io_output_number, get_io_input_number, is_event_notification, get_io_event_notification, get_event_notification_email, get_io_event_notification_port, get_io_input_valid_time, get_io_macro_type, get_io_macro, get_io_port_macro, get_io_macro_text, get_io_output_delay, get_io_output_initial_state, get_io_comment.

Remarks

4.5.27 get_io_port_macro

```
int get_io_port_macro(unsigned char *mac_address, int port_index);
```

Description

*get_io_port_macro*는 *port_index*로 지정된 출력포트의 [매크로 기능] 선택 여부를 반환합니다. *get_io_macro_type*이 1을 반환하는 경우에만 사용할 수 있습니다.

Parameters

*unsigned char *mac_address*

EzTCP_Search 또는 *EzTCP_Read*를 사용하여 검색된 제품(ezTCP)의 MAC주소 입니다.

int port_index

출력포트의 순번으로 0부터 시작합니다.

Return values

*get_io_port_macro*는 *port_index*로 지정된 출력포트에 [매크로 기능]이 선택되어 있으면 1을 반환하고 그렇지 않은 경우 0을 반환합니다.

*mac_address*을 가진 제품(ezTCP)을 찾을 수 없는 경우에는 -1을 반환합니다.

Examples

```

struct eztcp_info out_eztcp_info;
int ret = get_eztcp_info_by_index(0, &out_eztcp_info);

```



```

if(ret == EZTCP_SUCCESS)
{
    int idx = 0;
    int output_number = 0;
    unsigned char *mac_address = out_eztcp_info.mac_address;
    if(is_io(mac_address) == 1 && get_io_macro_type(mac_address) == 1)
    {
        output_number = get_io_output_number(mac_address);
        for(idx = 0; idx < output_number; idx++)
        {
            if(get_io_port_macro(mac_address, idx) == 1)
                printf("[I/O] Output Port#%d Macro is enabled.\r\n", idx);
            else
                printf("[I/O] Output Port#%d Macro is disabled.\r\n", idx);
        }
    }
}

```

See also

is_io, get_io_output_number, get_io_input_number, is_event_notification, get_io_event_notification, get_event_notification_email, get_io_event_notification_port, get_io_input_valid_time, get_io_macro_type, get_io_macro, get_io_port_macro, get_io_macro_text, get_io_output_delay, get_io_output_initial_state, get_io_comment.

Remarks

4.5.28 get_io_macro_text

```
int get_io_macro_text(unsigned char *mac_address, int port_index, char *out_macro_text);
```

Description

*get_io_macro_text*는 *port_index*로 지정된 출력포트의 매크로 문자열을 반환합니다.

Parameters

*unsigned char *mac_address*

EzTCP_Search 또는 *EzTCP_Read*를 사용하여 검색된 제품(*ezTCP*)의 MAC주소 입니다.

int port_index

출력포트의 순번으로 0부터 시작합니다.

*char *out_macro_text*

*port_index*로 지정된 출력포트의 매크로 문자열이 저장될 포인터입니다.

Return values

*get_io_macro_text*는 *out_macro_text*에 매크로 문자열을 저장하는데 성공하면 1을 반환합니다.

*mac_address*을 가진 제품(ezTCP)을 찾을 수 없는 경우에는 -1을 반환합니다.

Examples

```
struct eztcp_info out_eztcp_info;
int ret = get_eztcp_info_by_index(0, &out_eztcp_info);

if(ret == EZTCP_SUCCESS)
{
    int idx = 0;
    int output_number = 0;
    char buf[IO_SCRIPT_LEN + 1];
    unsigned char *mac_address = out_eztcp_info.mac_address;
    if(is_io(mac_address) == 1)
    {
        output_number = get_io_output_number(mac_address);
        for(idx = 0; idx < output_number; idx++)
        {
            memset(buf, 0x00, IO_SCRIPT_LEN + 1);
            get_io_macro_text(mac_address, idx, buf);
            printf("[I/O] Output port#%d Macro : %s\r\n", idx, buf);
        }
    }
}
```

See also

is_io, *get_io_output_number*, *get_io_input_number*, *is_event_notification*, *get_io_event_notification*, *get_event_notification_email*, *get_io_event_notification_port*, *get_io_input_valid_time*, *get_io_macro_type*, *get_io_macro*, *get_io_port_macro*, *get_io_macro_text*, *get_io_output_delay*, *get_io_output_initial_state*, *get_io_comment*.

Remarks

매크로 문자열은 NULL을 제외하고 최대 32바이트입니다. 따라서 *out_macro_text*는 최소 33바이트의 메모리 공간을 가지고 있어야 합니다.

매크로 문자열 길이는 **IO_SCRIPT_LEN**에 정의되어 있습니다.

4.5.29 get_io_output_delay

```
int get_io_output_delay(unsigned char *mac_address, int port_index,
                        char *out_output_delay);
```

Description

*get_io_output_delay*는 *port_index*로 지정된 출력포트의 [출력지연(밀리초)]을 반환합니다.

Parameters

*unsigned char *mac_address*

EzTCP_Search 또는 *EzTCP_Read*를 사용하여 검색된 제품(ezTCP)의 MAC주소 입니다.

int port_index

출력포트의 순번으로 0부터 시작합니다.

*char *out_output_delay*

*port_index*로 지정된 출력포트의 [출력지연(밀리초)]가 저장될 포인터입니다.

Return values

*get_io_output_delay*는 *out_output_delay*에 [출력지연(밀리초)]를 저장하는데 성공하면 1을 반환합니다.

*mac_address*을 가진 제품(ezTCP)을 찾을 수 없는 경우에는 -1을 반환합니다.

Examples

```
struct eztcp_info out_eztcp_info;
int ret = get_eztcp_info_by_index(0, &out_eztcp_info);

if(ret == EZTCP_SUCCESS)
{
    char buf[8];
    int idx = 0;
    int output_number = get_io_output_number(out_eztcp_info.mac_address);
    if(is_io(out_eztcp_info.mac_address) == 1)
    {
        for(idx = 0; idx < output_number; idx++)
        {
            memset(buf, 0x00, 8);
            get_io_output_delay(out_eztcp_info.mac_address, idx, buf);
            printf("[I/O] Output Port#%d Delay(ms) : %s\r\n", idx, buf);
        }
    }
}
```

See also

is_io, get_io_output_number, get_io_input_number, is_event_notification, get_io_event_notification, get_event_notification_email, get_io_event_notification_port, get_io_input_valid_time, get_io_macro_type, get_io_macro, get_io_port_macro, get_io_macro_text, get_io_output_delay, get_io_output_initial_state, get_io_comment.

Remarks

4.5.30 get_io_output_initial_state

```
int get_io_output_initial_state(unsigned char *mac_address, int port_index);
```

Description

*get_io_output_initial_state*는 *port_index*로 지정된 출력포트의 [초기상태]를 반환합니다.

Parameters

*unsigned char *mac_address*

EzTCP_Search 또는 *EzTCP_Read*를 사용하여 검색된 제품(ezTCP)의 MAC주소 입니다.

int port_index

출력포트의 순번으로 0부터 시작합니다.

Return values

*get_io_output_initial_state*는 출력포트의 [초기상태]를 반환하며 그 값은 아래와 같습니다.

0

OFF

1

ON

*mac_address*을 가진 제품(ezTCP)을 찾을 수 없는 경우에는 -1을 반환합니다.

Examples

```
struct eztcp_info out_eztcp_info;
int ret = get_eztcp_info_by_index(0, &out_eztcp_info);

if(ret == EZTCP_SUCCESS)
{
    int idx = 0;
    unsigned char *mac_address = out_eztcp_info.mac_address;
    int output_number = get_io_output_number(mac_address);
    if(is_io(out_eztcp_info.mac_address) == 1)
```

```

    {
        for(idx = 0;idx < output_number;idx++)
        {
            if(get_io_output_initial_state(mac_address, idx) == 1)
                printf("[I/O] Output Port#%d initial state is ON\r\n");
            else
                printf("[I/O] Output Port#%d initial state is OFF\r\n");
        }
    }
}

```

See also

is_io, get_io_output_number, get_io_input_number, is_event_notification, get_io_event_notification, get_event_notification_email, get_io_event_notification_port, get_io_input_valid_time, get_io_macro_type, get_io_macro, get_io_port_macro, get_io_macro_text, get_io_output_delay, get_io_output_initial_state, get_io_comment.

Remarks

4.5.31 get_io_comment

```

int get_io_comment(unsigned char *mac_address, int port_flag, int port_index,
                  char *out_comment);

```

Description

*get_io_comment*는 I/O 제품에 설치된 입력 또는 출력포트의 **[I/O 포트 설명]**을 반환합니다.

Parameters

*unsigned char *mac_address*

EzTCP_Search 또는 *EzTCP_Read*를 사용하여 검색된 제품(ezTCP)의 MAC주소 입니다.

int port_flag

*port_flag*가 0이면 입력포트를 의미하고 1이면 출력포트를 의미합니다.

int port_index

입력 또는 출력포트의 순번으로 0부터 시작합니다.

*char *out_comment*

*port_index*로 지정된 입력 또는 출력포트의 **[I/O 포트 설명]**이 저장될 포인터입니다.

Return values

*get_io_comment*는 *out_comment*에 *port_flag*와 *port_index*가 가리키는 입력 또는 출력포트의 **[I/O 포트 설명]**을 저장하는데 성공하면 1을 반환합니다.

*mac_address*을 가진 제품(ezTCP)을 찾을 수 없는 경우에는 -1을 반환합니다.

Examples

```
struct eztcp_info out_eztcp_info;
int ret = get_eztcp_info_by_index(0, &out_eztcp_info);

if(ret == EZTCP_SUCCESS)
{
    int idx = 0;
    char buf[IO_COMMENT_LEN + 1];
    unsigned char *mac_address = out_eztcp_info.mac_address;
    if(is_io(mac_address) == 1 && is_event_notification(mac_address) == 1)
    {
        if(get_io_event_notification(mac_address) == 1)
        {
            int input_number = get_io_input_number(mac_address);
            int output_number = get_io_output_number(mac_address);

            for(idx = 0; idx < input_number; idx++)
            {
                memset(buf, 0x00, IO_COMMENT_LEN + 1);
                get_io_comment(mac_address, 0, idx, buf);
                printf("[I/O] Input port#%d Comment : %s\r\n", idx, buf);
            }

            for(idx = 0; idx < output_number; idx++)
            {
                memset(buf, 0x00, IO_COMMENT_LEN + 1);
                get_io_comment(mac_address, 1, idx, buf);
                printf("[I/O] Output port#%d Comment : %s\r\n", idx, buf);
            }
        }
    }
}
```

See also

is_io, get_io_output_number, get_io_input_number, is_event_notification, get_io_event_notification, get_event_notification_email, get_io_event_notification_port, get_io_input_valid_time, get_io_macro_type, get_io_macro, get_io_port_macro, get_io_macro_text, get_io_output_delay, get_io_output_initial_state, get_io_comment.

Remarks



[I/O 포트 설명]은 NULL을 제외하고 최대 16바이트입니다. 따라서 *out_comment*는 최소 17바이트의 메모리 공간을 가지고 있어야 합니다.

[I/O 포트 설명] 길이는 *IO_COMMENT_LEN*에 정의되어 있습니다.

4.6 무선랜

4.6.1 get_wlan_type

```
int get_wlan_type(unsigned char *mac_address);
```

Description

*get_wlan_type*은 무선랜 제품의 [무선랜 종류]를 반환합니다.

Parameters

*unsigned char *mac_address*

EzTCP_Search 또는 *EzTCP_Read*를 사용하여 검색된 제품(ezTCP)의 MAC주소 입니다.

Return values

*get_wlan_type*은 무선랜 제품의 [무선랜 종류]를 반환하며 그 값은 아래와 같습니다.

0

애드혹

1

인프라스트럭처

2

Soft AP

*mac_address*을 가진 제품(ezTCP)을 찾을 수 없는 경우에는 -1을 반환합니다.

Examples

```
struct eztcp_info out_eztcp_info;
int ret = get_eztcp_info_by_index(0, &out_eztcp_info);

if(ret == EZTCP_SUCCESS)
{
    unsigned char *mac_address = out_eztcp_info.mac_address;
    if(is_wlan(mac_address) == 1)
    {
        int wlan_type = get_wlan_type(mac_address);
        if(wlan_type == 0)
            printf("[WLAN] WLAN Topology : Ad-hoc\r\n");
    }
}
```

```

        else if(wlan_type == 1)
            printf("[WLAN] WLAN Topology : Infrastructure\r\n");
        else if(is_wlan_soft_ap(mac_address) == 1 && wlan_type == 2)
            printf("[WLAN] WLAN Topology : Soft AP\r\n");
    }
}

```

See also

is_wlan, is_wlan_soft_ap, get_wlan_type, get_wlan_channel, get_wlan_ssid, get_wlan_antenna, get_wlan_phy_mode, get_wlan_short_preamble, get_wlan_short_slog, get_wlan_cts_protection, get_wlan_background_scan, get_wlan_encryption_type, get_wlan_authentication_type, get_wlan_wep_key_length, get_wlan_wep_key_index, get_wlan_wep_key_data_type, get_wlan_wep_key, get_wlan_max_wpa_passphrase_length, get_wlan_wpa_passphrase, get_wlan_shared_key, get_wlan_wpa_enterprise, get_wlan_wpa_enterprise_id, get_wlan_wpa_enterprise_pwd.

Remarks

4.6.2 get_wlan_channel

```
int get_wlan_channel(unsigned char *mac_address);
```

Description

*get_wlan_channel*은 무선랜 제품의 [채널]을 반환합니다.

Parameters

*unsigned char *mac_address*

EzTCP_Search 또는 *EzTCP_Read*를 사용하여 검색된 제품(ezTCP)의 MAC주소 입니다.

Return values

*get_wlan_channel*은 무선랜 제품의 [채널]을 반환합니다.

*mac_address*을 가진 제품(ezTCP)을 찾을 수 없는 경우에는 -1을 반환합니다.

Examples

```

struct eztcp_info out_eztcp_info;
int ret = get_eztcp_info_by_index(0, &out_eztcp_info);

if(ret == EZTCP_SUCCESS)
{
    unsigned char *mac_address = out_eztcp_info.mac_address;
    if(is_wlan(mac_address) == 1)

```



```

{
    int wlan_type = get_wlan_type(mac_address);
    if(wlan_type == 0 || (is_wlan_soft_ap(mac_address) == 1 && wlan_type == 2))
        printf("[WLAN] Channel : %d\r\n", get_wlan_channel(mac_address));
}
}

```

See also

is_wlan, is_wlan_soft_ap, get_wlan_type, get_wlan_channel, get_wlan_ssid, get_wlan_antenna,
 get_wlan_phy_mode, get_wlan_short_preamble, get_wlan_short_slog, get_wlan_cts_protection,
 get_wlan_background_scan, get_wlan_encryption_type, get_wlan_authentication_type,
 get_wlan_wep_key_length, get_wlan_wep_key_index, get_wlan_wep_key_data_type,
 get_wlan_wep_key, get_wlan_max_wpa_passphrase_length, get_wlan_wpa_passphrase,
 get_wlan_shared_key, get_wlan_wpa_enterprise, get_wlan_wpa_enterprise_id,
 get_wlan_wpa_enterprise_pwd.

Remarks

4.6.3 get_wlan_ssid

```
int get_wlan_ssid(unsigned char *mac_address, char *out_ssid);
```

Description

*get_wlan_ssid*는 무선랜 제품의 [SSID]를 반환합니다.

Parameters

*unsigned char *mac_address*

EzTCP_Search 또는 *EzTCP_Read*를 사용하여 검색된 제품(ezTCP)의 MAC주소 입니다.

*char *out_ssid*

무선랜 제품의 [SSID]가 저장될 포인터입니다.

Return values

*get_wlan_ssid*는 *out_ssid*에 무선랜 제품의 [SSID]를 저장하는데 성공하면 1을 반환합니다.
*mac_address*을 가진 제품(ezTCP)을 찾을 수 없는 경우에는 -1을 반환합니다.

Examples

```

struct eztcp_info out_eztcp_info;
int ret = get_eztcp_info_by_index(0, &out_eztcp_info);

if(ret == EZTCP_SUCCESS)
{

```

```

char buf[33];
unsigned char *mac_address = out_eztcp_info.mac_address;
if(is_wlan(mac_address) == 1)
{
    memset(buf, 0x00, 33);
    get_wlan_ssid(mac_address, buf);
    printf("[WLAN] SSID : %s\r\n", buf);
}
}

```

See also

is_wlan, is_wlan_soft_ap, get_wlan_type, get_wlan_channel, get_wlan_ssid, get_wlan_antenna, get_wlan_phy_mode, get_wlan_short_preamble, get_wlan_short_slog, get_wlan_cts_protection, get_wlan_background_scan, get_wlan_encryption_type, get_wlan_authentication_type, get_wlan_wep_key_length, get_wlan_wep_key_index, get_wlan_wep_key_data_type, get_wlan_wep_key, get_wlan_max_wpa_passphrase_length, get_wlan_wpa_passphrase, get_wlan_shared_key, get_wlan_wpa_enterprise, get_wlan_wpa_enterprise_id, get_wlan_wpa_enterprise_pwd.

Remarks

[SSID]는 NULL을 제외하고 최대 32바이트입니다. 따라서 *out_ssid*는 최소 33바이트의 메모리 공간을 가지고 있어야 합니다.

4.6.4 get_wlan_antenna

```
int get_wlan_antenna(unsigned char *mac_address);
```

Description

*get_wlan_antenna*는 무선랜 제품의 [안테나]를 반환합니다.

Parameters

*unsigned char *mac_address*

EzTCP_Search 또는 *EzTCP_Read*를 사용하여 검색된 제품(ezTCP)의 MAC주소 입니다.

Return values

*get_wlan_antenna*는 무선랜 제품의 [안테나]를 반환하며 그 값은 아래와 같습니다.

0

내장안테나

1

외장안테나

*mac_address*을 가진 제품(ezTCP)을 찾을 수 없는 경우에는 -1을 반환합니다.

Examples

```
struct eztcp_info out_eztcp_info;
int ret = get_eztcp_info_by_index(0, &out_eztcp_info);

if(ret == EZTCP_SUCCESS)
{
    unsigned char *mac_address = out_eztcp_info.mac_address;
    if(is_wlan(mac_address) == 1 && is_wlan_antenna(mac_address) == 1)
    {
        if(get_wlan_antenna(mac_address) == 0)
            printf("[WLAN] Internal Antenna.\r\n");
        else
            printf("[WLAN] External Antenna.\r\n");
    }
}
```

See also

is_wlan, is_wlan_soft_ap, get_wlan_type, get_wlan_channel, get_wlan_ssid, is_wlan_antenna, get_wlan_antenna, get_wlan_phy_mode, get_wlan_short_preamble, get_wlan_short_slog, get_wlan_cts_protection, get_wlan_background_scan, get_wlan_encryption_type, get_wlan_authentication_type, get_wlan_wep_key_length, get_wlan_wep_key_index, get_wlan_wep_key_data_type, get_wlan_wep_key, get_wlan_max_wpa_passphrase_length, get_wlan_wpa_passphrase, get_wlan_shared_key, get_wlan_wpa_enterprise, get_wlan_wpa_enterprise_id, get_wlan_wpa_enterprise_pwd.

Remarks

4.6.5 get_wlan_phy_mode

```
int get_wlan_phy_mode(unsigned char *mac_address);
```

Description

*get_wlan_phy_mode*는 무선랜 제품의 무선 고급설정들 중에서 [Phy Mode]를 반환합니다.

Parameters

*unsigned char *mac_address*

EzTCP_Search 또는 *EzTCP_Read*를 사용하여 검색된 제품(ezTCP)의 MAC주소 입니다.

Return values

*get_wlan_phy_mode*는 무선랜 제품의 무선 고급설정들 중에서 [Phy Mode]를 반환하며 그 값은 아래와 같습니다.

1
802.11

2
802.11b

3
802.11b/g

*mac_address*을 가진 제품(ezTCP)을 찾을 수 없는 경우에는 -1을 반환합니다.

Examples

```
struct eztcp_info out_eztcp_info;
int ret = get_eztcp_info_by_index(0, &out_eztcp_info);

if(ret == EZTCP_SUCCESS)
{
    unsigned char *mac_address = out_eztcp_info.mac_address;
    if(is_wlan(mac_address) == 1 && is_wlan_phy_mode(mac_address) == 1)
    {
        int phy_mode = get_wlan_phy_mode(mac_address);

        if(phy_mode == 1 )
            printf("[WLAN] Phy mode : 802.11\r\n");
        else if(phy_mode == 2)
            printf("[WLAN] Phy mode : 802.11b\r\n");
        else if(phy_mode == 3)
            printf("[WLAN] Phy mode : 802.11b/g\r\n");
    }
}
```

See also

is_wlan, is_wlan_soft_ap, get_wlan_type, get_wlan_channel, get_wlan_ssid, is_wlan_antenna,
get_wlan_antenna, is_wlan_phy_mode, get_wlan_phy_mode, get_wlan_short_preamble,
get_wlan_short_slog, get_wlan_cts_protection, get_wlan_background_scan, get_wlan_encryption_type,
get_wlan_authentication_type, get_wlan_wep_key_length, get_wlan_wep_key_index,
get_wlan_wep_key_data_type, get_wlan_wep_key, get_wlan_max_wpa_passphrase_length,
get_wlan_wpa_passphrase, get_wlan_shared_key, get_wlan_wpa_enterprise, get_wlan_wpa_enterprise_id,
get_wlan_wpa_enterprise_pwd,

Remarks

보다 자세한 내용은 해당 제품 사용설명서를 참조하시기 바랍니다.

4.6.6 get_wlan_short_preamble

```
int get_wlan_short_preamble(unsigned char *mac_address);
```

Description

*get_wlan_short_preamble*은 무선랜 제품의 무선 고급설정들 중에서 [Short Preamble] 선택 여부를 반환합니다.

Parameters

*unsigned char *mac_address*

EzTCP_Search 또는 *EzTCP_Read*를 사용하여 검색된 제품(ezTCP)의 MAC주소 입니다.

Return values

*get_wlan_short_preamble*은 무선랜 제품의 무선 고급설정들 중에서 [Short Preamble]이 선택되어 있으면 1을 반환하고 그렇지 않으면 0을 반환합니다.

*mac_address*을 가진 제품(ezTCP)을 찾을 수 없는 경우에는 -1을 반환합니다.

Examples

```
struct eztcp_info out_eztcp_info;
int ret = get_eztcp_info_by_index(0, &out_eztcp_info);

if(ret == EZTCP_SUCCESS)
{
    unsigned char *mac_address = out_eztcp_info.mac_address;
    if(is_wlan(mac_address) == 1 && is_wlan_phy_mode(mac_address) == 1)
    {
        int phy_mode = get_wlan_phy_mode(mac_address);

        if(phy_mode == 2 || phy_mode == 3)
        {
            if(get_wlan_short_preamble(mac_address) == 1)
                printf("[WLAN] Short Preamble is enabled.\r\n");
            else
                printf("[WLAN] Short Preamble is disabled.\r\n");
        }
    }
}
```

See also

is_wlan, is_wlan_soft_ap, get_wlan_type, get_wlan_channel, get_wlan_ssid, is_wlan_antenna,
get_wlan_antenna, is_wlan_phy_mode, get_wlan_phy_mode, get_wlan_short_preamble,

get_wlan_short_slog, get_wlan_cts_protectionin, get_wlan_background_scan, get_wlan_encryption_type,
 get_wlan_authentication_type, get_wlan_wpa_enterprise, get_wlan_wpa_enterprise_id,
 get_wlan_wpa_enterprise_pwd, get_wlan_wep_key_length, get_wlan_wep_key_index,
 get_wlan_wep_key_data_type, get_wlan_wep_key, get_wlan_max_wpa_passphrase_length,
 get_wlan_wpa_passphrase, get_wlan_shared_key.

Remarks

보다 자세한 내용은 해당 제품 사용설명서를 참조하시기 바랍니다.

4.6.7 get_wlan_short_slot

```
int get_wlan_short_slot(unsigned char *mac_address);
```

Description

*get_wlan_short_slot*은 무선랜 제품의 무선 고급설정들 중에서 [Short Slot] 선택 여부를 반환합니다.

Parameters

*unsigned char *mac_address*

EzTCP_Search 또는 *EzTCP_Read*를 사용하여 검색된 제품(ezTCP)의 MAC주소 입니다.

Return values

*get_wlan_short_slot*은 무선랜 제품의 무선 고급설정들 중에서 [Short Slot]이 선택되어 있으면 1을 반환하고 그렇지 않으면 0을 반환합니다.

*mac_address*을 가진 제품(ezTCP)을 찾을 수 없는 경우에는 -1을 반환합니다.

Examples

```
struct eztcp_info out_eztcp_info;
int ret = get_eztcp_info_by_index(0, &out_eztcp_info);

if(ret == EZTCP_SUCCESS)
{
    unsigned char *mac_address = out_eztcp_info.mac_address;
    if(is_wlan(mac_address) == 1 && is_wlan_phy_mode(mac_address) == 1)
    {
        int phy_mode = get_wlan_phy_mode(mac_address);

        if(phy_mode == 3)
        {
            if(get_wlan_short_slot(mac_address) == 1)
                printf("[WLAN] Short Slot is enabled.\r\n");
        }
    }
}
```

```

        else
            printf("[WLAN] Short Slot is disabled.\r\n");
        }
    }
}

```

See also

is_wlan, is_wlan_soft_ap, get_wlan_type, get_wlan_channel, get_wlan_ssid, is_wlan_antenna, get_wlan_antenna, is_wlan_phy_mode, get_wlan_phy_mode, get_wlan_short_preamble, get_wlan_short_slog, get_wlan_cts_protection, get_wlan_background_scan, get_wlan_encryption_type, get_wlan_authentication_type, get_wlan_wpa_enterprise, get_wlan_wpa_enterprise_id, get_wlan_wpa_enterprise_pwd, get_wlan_wep_key_length, get_wlan_wep_key_index, get_wlan_wep_key_data_type, get_wlan_wep_key, get_wlan_max_wpa_passphrase_length, get_wlan_wpa_passphrase, get_wlan_shared_key.

Remarks

보다 자세한 내용은 해당 제품 사용설명서를 참조하시기 바랍니다.

4.6.8 get_wlan_cts_protection

```
int get_wlan_cts_protection(unsigned char *mac_address);
```

Description

*get_wlan_cts_protection*은 무선랜 제품의 무선 고급설정들 중에서 [CTS Protection] 선택 여부를 반환합니다.

Parameters

*unsigned char *mac_address*

EzTCP_Search 또는 *EzTCP_Read*를 사용하여 검색된 제품(ezTCP)의 MAC주소 입니다.

Return values

*get_wlan_cts_protection*은 무선랜 제품의 무선 고급설정들 중에서 [CTS Protection]이 선택 되어 있으면 1을 반환하고 그렇지 않으면 0을 반환합니다.

*mac_address*을 가진 제품(ezTCP)을 찾을 수 없는 경우에는 -1을 반환합니다.

Examples

```

struct eztcp_info out_eztcp_info;
int ret = get_eztcp_info_by_index(0, &out_eztcp_info);

if(ret == EZTCP_SUCCESS)
{

```

```

    unsigned char *mac_address = out_eztcp_info.mac_address;
    if(is_wlan(mac_address) == 1 && is_wlan_phy_mode(mac_address) == 1)
    {
        int phy_mode = get_wlan_phy_mode(mac_address);

        if(phy_mode == 3)
        {
            if(get_wlan_cts_protection(mac_address) == 1)
                printf("[WLAN] CTS Protection is enabled.\r\n");
            else
                printf("[WLAN] CTS Protection is disabled.\r\n");
        }
    }
}

```

See also

is_wlan, is_wlan_soft_ap, get_wlan_type, get_wlan_channel, get_wlan_ssid, is_wlan_antenna, get_wlan_antenna, is_wlan_phy_mode, get_wlan_phy_mode, get_wlan_short_preamble, get_wlan_short_slog, get_wlan_cts_protection, get_wlan_background_scan, get_wlan_encryption_type, get_wlan_authentication_type, get_wlan_wpa_enterprise, get_wlan_wpa_enterprise_id, get_wlan_wpa_enterprise_pwd, get_wlan_wep_key_length, get_wlan_wep_key_index, get_wlan_wep_key_data_type, get_wlan_wep_key, get_wlan_max_wpa_passphrase_length, get_wlan_wpa_passphrase, get_wlan_shared_key.

Remarks

보다 자세한 내용은 해당 제품 사용설명서를 참조하시기 바랍니다.

4.6.9 get_wlan_background_scan

```
int get_wlan_background_scan(unsigned char *mac_address);
```

Description

*get_wlan_background_scan*은 무선랜 제품의 무선 고급설정들 중에서 **[Background Scan]** 선택 여부를 반환합니다.

Parameters

*unsigned char *mac_address*

EzTCP_Search 또는 *EzTCP_Read*를 사용하여 검색된 제품(ezTCP)의 MAC주소 입니다.

Return values

*get_wlan_background_scan*은 무선랜 제품의 무선 고급설정들 중에서 **[Background Scan]**이

선택되어 있으면 1을 반환하고 그렇지 않으면 0을 반환합니다.

*mac_address*을 가진 제품(ezTCP)을 찾을 수 없는 경우에는 -1을 반환합니다.

Examples

```
struct eztcp_info out_eztcp_info;
int ret = get_eztcp_info_by_index(0, &out_eztcp_info);

if(ret == EZTCP_SUCCESS)
{
    unsigned char *mac_address = out_eztcp_info.mac_address;
    if(is_wlan(mac_address) == 1 && is_wlan_background_scan(mac_address) == 1)
    {
        if(get_wlan_background_scan(mac_address) == 1)
            printf("[WLAN] Background Scan is enabled.\r\n");
        else
            printf("[WLAN] Background Scan is disabled.\r\n");
    }
}
```

See also

is_wlan, is_wlan_soft_ap, get_wlan_type, get_wlan_channel, get_wlan_ssid, is_wlan_antenna,
get_wlan_antenna, is_wlan_phy_mode, get_wlan_phy_mode, get_wlan_short_preamble,
get_wlan_short_slog, get_wlan_cts_protection, is_wlan_background_scan, get_wlan_background_scan,
get_wlan_encryption_type, get_wlan_authentication_type, get_wlan_wpa_enterprise,
get_wlan_wpa_enterprise_id, get_wlan_wpa_enterprise_pwd, get_wlan_wep_key_length,
get_wlan_wep_key_index, get_wlan_wep_key_data_type, get_wlan_wep_key,
get_wlan_max_wpa_passphrase_length, get_wlan_wpa_passphrase, get_wlan_shared_key.

Remarks

보다 자세한 내용은 해당 제품 사용설명서를 참조하시기 바랍니다.

4.6.10 get_wlan_encryption_type

```
int get_wlan_encryption_type(unsigned char *mac_address);
```

Description

*get_wlan_encryption_type*은 무선랜 제품의 [암호화 방식]을 반환합니다.

Parameters

*unsigned char *mac_address*

EzTCP_Search 또는 *EzTCP_Read*를 사용하여 검색된 제품(ezTCP)의 MAC주소 입니다.

Return values

*get_wlan_encryption_type*은 무선랜 제품의 [암호화 방식]을 반환하며 그 값은 아래와 같습니다.

0

없음

1

WEP

2

WPA

*mac_address*을 가진 제품(ezTCP)을 찾을 수 없는 경우에는 -1을 반환합니다.

*is_wlan_rsn*이 1을 반환하는 경우에는 -2를 반환합니다.

Examples

```
struct eztcp_info out_eztcp_info;
int ret = get_eztcp_info_by_index(0, &out_eztcp_info);

if(ret == EZTCP_SUCCESS)
{
    unsigned char *mac_address = out_eztcp_info.mac_address;
    if(is_wlan(mac_address) == 1 && is_wlan_rsn(mac_address) == 0)
    {
        int enc_type = get_wlan_encryption_type(mac_address);
        if(enc_type == 0)
            printf("[WLAN] Encryption : None.\r\n");
        else if(enc_type == 1)
            printf("[WLAN] Encryption : WEP.\r\n");
        else if(enc_type == 2)
            printf("[WLAN] Encryption : WPA.\r\n");
    }
}
```

See also

is_wlan, *is_wlan_soft_ap*, *get_wlan_type*, *get_wlan_channel*, *get_wlan_ssid*, *is_wlan_antenna*,
get_wlan_antenna, *is_wlan_phy_mode*, *get_wlan_phy_mode*, *get_wlan_short_preamble*,
get_wlan_short_slog, *get_wlan_cts_protection*, *is_wlan_background_scan*, *get_wlan_background_scan*,
is_wlan_rsn, *get_wlan_encryption_type*, *get_wlan_authentication_type*, *get_wlan_wpa_enterprise*,
get_wlan_wpa_enterprise_id, *get_wlan_wpa_enterprise_pwd*, *get_wlan_wep_key_length*,
get_wlan_wep_key_index, *get_wlan_wep_key_data_type*, *get_wlan_wep_key*,
get_wlan_max_wpa_passphrase_length, *get_wlan_wpa_passphrase*, *get_wlan_shared_key*.

Remarks



4.6.11 get_wlan_authentication_type

```
int get_wlan_authentication_type(unsigned char *mac_address);
```

Description

*get_wlan_authentication_type*은 무선랜 제품의 [인증 방식] 또는 WPA[인증 방식/암호화 방법]을 반환합니다. *get_wlan_authentication_type*이 반환 값은 *get_wlan_encryption_type*의 반환 값에 따라서 의미가 달라집니다.

Parameters

*unsigned char *mac_address*

EzTCP_Search 또는 *EzTCP_Read*를 사용하여 검색된 제품(ezTCP)의 MAC주소 입니다.

Return values

*get_wlan_encryption_type*이 0(없음)을 반환하는 경우 *get_wlan_authentication_type*은 항상 1을 반환합니다.

1

개방 모드

*get_wlan_encryption_type*이 1(WEP)을 반환하는 경우 *get_wlan_authentication_type*은 WEP[인증 방식]을 반환하며 그 값은 아래와 같습니다.

1

개방 모드

2

공유 모드

3

자동 모드

*get_wlan_encryption_type*이 2(WPA)를 반환하는 경우 *get_wlan_authentication_type*은 WPA[인증 방식/암호화 방법]을 반환하며 그 값은 아래와 같습니다.

0

WPA PSK – TKIP

1

WPA PSK – AES

2

WPA PSK – TKIP/AES

3

WPA2 PSK – TKIP

4

WPA2 PSK – AES

WPA2 PSK- TKIP/AES

*mac_address*을 가진 제품(ezTCP)을 찾을 수 없는 경우에는 -1을 반환합니다.
*is_wlan_rsn*이 1을 반환하는 경우에는 -2를 반환합니다.

Examples

```

struct eztcp_info out_eztcp_info;
int ret = get_eztcp_info_by_index(0, &out_eztcp_info);

if(ret == EZTCP_SUCCESS)
{
    unsigned char *mac_address = out_eztcp_info.mac_address;
    if(is_wlan(mac_address) == 1 && is_wlan_rsn(mac_address) == 0)
    {
        int enc_type = get_wlan_encryption_type(mac_address);
        int auth_type = get_wlan_authentication_type(mac_address);
        if(enc_type == 0)
        {
            printf("[WLAN] Encyption : None.\r\n");
            if(auth_type == 1)
                printf("[WLAN] Authentication : Open System.\r\n");
        }
        else if(enc_type == 1)
        {
            printf("[WLAN] Encyption : WEP.\r\n");
            if(auth_type == 1)
                printf("[WLAN] Authentication : Open System.\r\n");
            else if(auth_type == 2)
                printf("[WLAN] Authentication : Shared Key.\r\n");
            else if(auth_type == 3)
                printf("[WLAN] Authentication : Auth.\r\n");
        }
        else if(enc_type == 2)
        {
            printf("[WLAN] Encyption : WPA.\r\n");
            if(auth_type == 0)
                printf("[WLAN] Authentication : WPA PSK - TKIP.\r\n");
            else if(auth_type == 1)
                printf("[WLAN] Authentication : WPA PSK - AES.\r\n");
            else if(auth_type == 2)

```

```

        printf("[WLAN] Authentication : WPA PSK - TKIP/AES.\r\n");
    else if(auth_type == 3)
        printf("[WLAN] Authentication : WPA2 PSK - TKIP.\r\n");
    else if(auth_type == 4)
        printf("[WLAN] Authentication : WPA2 PSK - AES.\r\n");
    else if(auth_type == 5)
        printf("[WLAN] Authentication : WPA2 PSK - TKIP/AES.\r\n");
    }
}
}

```

See also

is_wlan, is_wlan_soft_ap, get_wlan_type, get_wlan_channel, get_wlan_ssid, is_wlan_antenna, get_wlan_antenna, is_wlan_phy_mode, get_wlan_phy_mode, get_wlan_short_preamble, get_wlan_short_slog, get_wlan_cts_protection, is_wlan_background_scan, get_wlan_background_scan, is_wlan_rsn, get_wlan_encryption_type, get_wlan_authentication_type, get_wlan_wep_key_length, get_wlan_wep_key_index, get_wlan_wep_key_data_type, get_wlan_wep_key, get_wlan_max_wpa_passphrase_length, get_wlan_wpa_passphrase, get_wlan_wpa_enterprise, get_wlan_wpa_enterprise_id, get_wlan_wpa_enterprise_pwd, get_wlan_shared_key.

Remarks

4.6.12 get_wlan_wep_key_length

```
int get_wlan_wep_key_length(unsigned char *mac_address);
```

Description

*get_wlan_wep_key_length*은 무선랜 제품의 WEP 키 길이를 반환합니다.

Parameters

*unsigned char *mac_address*

EzTCP_Search 또는 *EzTCP_Read*를 사용하여 검색된 제품(ezTCP)의 MAC주소 입니다.

Return values

*get_wlan_wep_key_length*은 무선랜 제품의 WEP 키 길이를 반환하며 그 값은 아래와 같습니다.

1

64비트

2

128비트

*mac_address*을 가진 제품(ezTCP)을 찾을 수 없는 경우에는 -1을 반환합니다.

Examples

```
struct eztcp_info out_eztcp_info;
int ret = get_eztcp_info_by_index(0, &out_eztcp_info);

if(ret == EZTCP_SUCCESS)
{
    unsigned char *mac_address = out_eztcp_info.mac_address;
    if(is_wlan(mac_address) == 1)
    {
        int wep_type = get_wlan_wep_key_length(mac_address);
        if(wep_type == 1)
            printf("[WLAN] WEP Key : 64Bit.\r\n");
        else if(wep_type == 2)
            printf("[WLAN] WEP Key : 128Bit.\r\n");
    }
}
```

See also

is_wlan, is_wlan_soft_ap, get_wlan_type, get_wlan_channel, get_wlan_ssid, is_wlan_antenna, get_wlan_antenna, is_wlan_phy_mode, get_wlan_phy_mode, get_wlan_short_preamble, get_wlan_short_slog, get_wlan_cts_protection, is_wlan_background_scan, get_wlan_background_scan, get_wlan_encryption_type, get_wlan_authentication_type, get_wlan_wep_key_length, get_wlan_wep_key_index, get_wlan_wep_key_data_type, get_wlan_wep_key, get_wlan_max_wpa_passphrase_length, get_wlan_wpa_passphrase, get_wlan_wpa_enterprise, get_wlan_wpa_enterprise_id, get_wlan_wpa_enterprise_pwd, get_wlan_shared_key.

Remarks

4.6.13 get_wlan_wep_key_index

```
int get_wlan_wep_key_index(unsigned char *mac_address);
```

Description

*get_wlan_wep_key_index*은 무선랜 제품이 사용하는 WEP 키 순번을 반환합니다.

Parameters

*unsigned char *mac_address*

EzTCP_Search 또는 *EzTCP_Read*를 사용하여 검색된 제품(ezTCP)의 MAC주소 입니다.

Return values

*get_wlan_wep_key_index*는 무선랜 제품이 사용하는 WEP 키 순번을 반환합니다.

*mac_address*을 가진 제품(ezTCP)을 찾을 수 없는 경우에는 -1을 반환합니다.

Examples

```
struct eztcp_info out_eztcp_info;
int ret = get_eztcp_info_by_index(0, &out_eztcp_info);

if(ret == EZTCP_SUCCESS)
{
    unsigned char *mac_address = out_eztcp_info.mac_address;
    if(is_wlan(mac_address) == 1)
    {
        int key_index = get_wlan_wep_key_index(mac_address);
        printf("[WLAN] WEP Key index %d is using now.\r\n", key_index);
    }
}
```

See also

is_wlan, is_wlan_soft_ap, get_wlan_type, get_wlan_channel, get_wlan_ssid, is_wlan_antenna, get_wlan_antenna, is_wlan_phy_mode, get_wlan_phy_mode, get_wlan_short_preamble, get_wlan_short_slog, get_wlan_cts_protection, is_wlan_background_scan, get_wlan_background_scan, get_wlan_encryption_type, get_wlan_authentication_type, get_wlan_wep_key_length, get_wlan_wep_key_index, get_wlan_wep_key_data_type, get_wlan_wep_key, get_wlan_max_wpa_passphrase_length, get_wlan_wpa_passphrase, get_wlan_shared_key, get_wlan_wpa_enterprise, get_wlan_wpa_enterprise_id, get_wlan_wpa_enterprise_pwd.

Remarks

4.6.14 get_wlan_wep_key_data_type

```
int get_wlan_wep_key_data_type(unsigned char *mac_address, int wep_key_index);
```

Description

*get_wlan_wep_key_data_type*은 *wep_key_index*로 지정된 WEP 키가 저장된 데이터 형식을 반환합니다.

Parameters

*unsigned char *mac_address*

EzTCP_Search 또는 *EzTCP_Read*를 사용하여 검색된 제품(ezTCP)의 MAC주소 입니다.

int wep_key_index

WEP 키 순번으로 0 ~ 3입니다.

Return values

*get_wlan_wep_key_data_type*은 *wep_key_index*로 지정된 WEP 키가 저장된 데이터 형식을 반환하며 그 값은 아래와 같습니다.

0

16진수

1

ASCII

*mac_address*을 가진 제품(ezTCP)을 찾을 수 없는 경우에는 -1을 반환합니다.

Examples**See also**

is_wlan, *is_wlan_soft_ap*, *get_wlan_type*, *get_wlan_channel*, *get_wlan_ssid*, *is_wlan_antenna*,
get_wlan_antenna, *is_wlan_phy_mode*, *get_wlan_phy_mode*, *get_wlan_short_preamble*,
get_wlan_short_slog, *get_wlan_cts_protection*, *is_wlan_background_scan*, *get_wlan_background_scan*,
get_wlan_encryption_type, *get_wlan_authentication_type*, *get_wlan_wep_key_length*,
get_wlan_wep_key_index, *get_wlan_wep_key_data_type*, *get_wlan_wep_key*,
get_wlan_max_wpa_passphrase_length, *get_wlan_wpa_passphrase*, *get_wlan_shared_key*,
get_wlan_wpa_enterprise, *get_wlan_wpa_enterprise_id*, *get_wlan_wpa_enterprise_pwd*.

Remarks**4.6.15 get_wlan_wep_key**

```
int get_wlan_wep_key(unsigned char *mac_address, char *out_wep_key);
```

Description

*get_wlan_wep_key*는 현재 무선랜 제품이 사용 중인 WEP 키를 반환합니다.

Parameters

*unsigned char *mac_address*

EzTCP_Search 또는 *EzTCP_Read*를 사용하여 검색된 제품(ezTCP)의 MAC주소 입니다.

*char *out_wep_key*

WEP 키가 저장될 포인터입니다.

Return values

*get_wlan_wep_key*는 *out_wep_key*에 무선랜 제품이 사용 중인 WEP 키를 저장하는데 성공하면 1을 반환합니다.

*mac_address*을 가진 제품(ezTCP)을 찾을 수 없는 경우에는 -1을 반환합니다.

Examples

```

struct eztcp_info out_eztcp_info;
int ret = get_eztcp_info_by_index(0, &out_eztcp_info);

if(ret == EZTCP_SUCCESS)
{
    int wep_type;
    int key_index;
    int key_type;
    char buf[32];
    unsigned char *mac_address = out_eztcp_info.mac_address;
    if(is_wlan(mac_address) == 1)
    {
        wep_type = get_wlan_wep_key_length(mac_address);
        if(wep_type == 1)
            printf("[WLAN] WEP Key : 64Bit.\r\n");
        else if(wep_type == 2)
            printf("[WLAN] WEP Key : 128Bit.\r\n");

        key_index = get_wlan_wep_key_index(mac_address);

        key_type = get_wlan_wep_key_data_type(mac_address, key_index);
        if(key_type == 0)
            printf("[WLAN] WEP Key index %d is hex format\r\n", key_index);
        else if(key_type == 1)
            printf("[WLAN] WEP Key index %d is ASCII format\r\n", key_index);

        memset(buf, 0x00, 32);
        get_wlan_wep_key(mac_address, buf);
        printf("[WLAN] WEP Key index %d is %s\r\n", key_index, buf);
    }
}

```

See also

is_wlan, is_wlan_soft_ap, get_wlan_type, get_wlan_channel, get_wlan_ssid, is_wlan_antenna,
 get_wlan_antenna, is_wlan_phy_mode, get_wlan_phy_mode, get_wlan_short_preamble,
 get_wlan_short_slog, get_wlan_cts_protection, is_wlan_background_scan, get_wlan_background_scan,
 get_wlan_encryption_type, get_wlan_authentication_type, get_wlan_wep_key_length,
 get_wlan_wep_key_index, get_wlan_wep_key_data_type, get_wlan_wep_key,
 get_wlan_max_wpa_passphrase_length, get_wlan_wpa_passphrase, get_wlan_shared_key,
 get_wlan_wpa_enterprise, get_wlan_wpa_enterprise_id, get_wlan_wpa_enterprise_pwd.

Remarks

*out_wep_key*에는 WEP 키 길이가 128비트, 저장된 데이터 형식이 16진수인 경우 NULL을 제외하고 최대 26바이트가 저장됩니다. 따라서 *out_wep_key*는 최소 33바이트의 메모리 공간을 가지고 있어야 합니다.

4.6.16 get_wlan_max_wpa_passphrase_length

```
int get_wlan_max_wpa_passphrase_length(unsigned char *mac_address);
```

Description

*get_wlan_max_wpa_passphrase_length*는 무선랜 제품에 저장가능한 WPA 암호문 최대길이를 반환합니다.

Parameters

*unsigned char *mac_address*

EzTCP_Search 또는 *EzTCP_Read*를 사용하여 검색된 제품(ezTCP)의 MAC주소 입니다.

Return values

*get_wlan_max_wpa_passphrase_length*는 무선랜 제품에 저장가능한 WPA 암호문 최대길이를 반환합니다. 그 값은 32 또는 64입니다.

*mac_address*을 가진 제품(ezTCP)을 찾을 수 없는 경우에는 -1을 반환합니다.

Examples**See also**

is_wlan, *is_wlan_soft_ap*, *get_wlan_type*, *get_wlan_channel*, *get_wlan_ssid*, *is_wlan_antenna*, *get_wlan_antenna*, *is_wlan_phy_mode*, *get_wlan_phy_mode*, *get_wlan_short_preamble*, *get_wlan_short_slog*, *get_wlan_cts_protection*, *is_wlan_background_scan*, *get_wlan_background_scan*, *get_wlan_encryption_type*, *get_wlan_authentication_type*, *get_wlan_wep_key_length*, *get_wlan_wep_key_index*, *get_wlan_wep_key_data_type*, *get_wlan_wep_key*, *get_wlan_max_wpa_passphrase_length*, *get_wlan_wpa_passphrase*, *get_wlan_shared_key*, *get_wlan_wpa_enterprise*, *get_wlan_wpa_enterprise_id*, *get_wlan_wpa_enterprise_pwd*.

Remarks

*get_wlan_max_wpa_passphrase_length*의 반환 값은 NULL이 포함된 WPA 암호문의 최대 길이이므로 실제로 사용할 수 있는 WPA 암호문의 최대길이는 *get_wlan_max_wpa_passphrase_length*의 반환 값보다 1바이트 작습니다.

4.6.17 get_wlan_wpa_passphrase

```
int get_wlan_wpa_passphrase(unsigned char *mac_address, char *out_wpa_passphrase);
```

Description

*get_wlan_wpa_passphrase*는 무선랜 제품의 WPA 암호문을 반환합니다.

Parameters

*unsigned char *mac_address*

EzTCP_Search 또는 *EzTCP_Read*를 사용하여 검색된 제품(ezTCP)의 MAC주소 입니다.

*char *out_wpa_passphrase*

WPA 암호문이 저장될 포인터입니다.

Return values

*get_wlan_wpa_passphrase*는 *out_wpa_passphrase*에 무선랜 제품의 WPA 암호문을 저장하는데 성공하면 1을 반환합니다.

*mac_address*을 가진 제품(ezTCP)을 찾을 수 없는 경우에는 -1을 반환합니다.

*is_wlan_rsn*이 1을 반환하는 경우에는 -2를 반환합니다.

Examples

```
struct eztcp_info out_eztcp_info;
int ret = get_eztcp_info_by_index(0, &out_eztcp_info);

if(ret == EZTCP_SUCCESS)
{
    char buf[65];
    int max_len;
    unsigned char *mac_address = out_eztcp_info.mac_address;
    if(is_wlan(mac_address) == 1 && is_wlan_rsn(mac_address) == 0)
    {
        max_len = get_wlan_max_wpa_passphrase_length(mac_address);
        printf("[WLAN] Maximum WPA passphrase length : %d\r\n", max_len);

        memset(buf, 0x00, 65);
        get_wlan_wpa_passphrase(mac_address, buf);
        printf("[WLAN] WPA passphrase : %s.\r\n", buf);
    }
}
```

See also

is_wlan, *is_wlan_soft_ap*, *get_wlan_type*, *get_wlan_channel*, *get_wlan_ssid*, *is_wlan_antenna*,
get_wlan_antenna, *is_wlan_phy_mode*, *get_wlan_phy_mode*, *get_wlan_short_preamble*,
get_wlan_short_slog, *get_wlan_cts_protection*, *is_wlan_background_scan*, *get_wlan_background_scan*,
get_wlan_encryption_type, *get_wlan_authentication_type*, *get_wlan_wep_key_length*,
get_wlan_wep_key_index, *get_wlan_wep_key_data_type*, *get_wlan_wep_key*,

`get_wlan_max_wpa_passphrase_length`, `is_wlan_rsn`, `get_wlan_wpa_passphrase`, `get_wlan_shared_key`, `get_wlan_wpa_enterprise`, `get_wlan_wpa_enterprise_id`, `get_wlan_wpa_enterprise_pwd`.

Remarks

WPA 암호문은 NULL을 포함해서 최대 64바이트입니다. 따라서 `out_wpa_passphrase`는 최소 64바이트의 메모리 공간을 가지고 있어야 합니다.

4.6.18 get_wlan_shared_key

```
int get_wlan_shared_key(unsigned char *mac_address, char *out_shared_key);
```

Description

`get_wlan_shared_key`는 무선랜 제품의 보안 설정 중에서 [Shared Key]를 반환합니다.

Parameters

*unsigned char *mac_address*

`EzTCP_Search` 또는 `EzTCP_Read`를 사용하여 검색된 제품(ezTCP)의 MAC주소 입니다.

*char *out_shared_key*

[Shared Key]가 저장될 포인터입니다.

Return values

`get_wlan_shared_key`는 `out_shared_key`에 무선랜 제품의 [Shared Key]를 저장하는데 성공하면 1을 반환합니다.

`mac_address`을 가진 제품(ezTCP)을 찾을 수 없는 경우에는 -1을 반환합니다.

`is_wlan_rsn`이 1을 반환하지 않는 경우에는 -2를 반환합니다.

Examples

```
struct eztcp_info out_eztcp_info;
int ret = get_eztcp_info_by_index(0, &out_eztcp_info);

if(ret == EZTCP_SUCCESS)
{
    char buf[64];
    unsigned char *mac_address = out_eztcp_info.mac_address;
    if(is_wlan(mac_address) == 1 && is_wlan_rsn(mac_address) == 1)
    {
        memset(buf, 0x00, 64);
        get_wlan_shared_key(mac_address, buf);
        printf("[WLAN] Security Settings - Shared Key : %s\r\n", buf);
    }
}
```

}

See also

is_wlan, is_wlan_soft_ap, get_wlan_type, get_wlan_channel, get_wlan_ssid, is_wlan_antenna, get_wlan_antenna, is_wlan_phy_mode, get_wlan_phy_mode, get_wlan_short_preamble, get_wlan_short_slog, get_wlan_cts_protection, is_wlan_background_scan, get_wlan_background_scan, get_wlan_encryption_type, get_wlan_authentication_type, get_wlan_wep_key_length, get_wlan_wep_key_index, get_wlan_wep_key_data_type, get_wlan_wep_key, get_wlan_max_wpa_passphrase_length, is_wlan_rsn, get_wlan_wpa_passphrase, get_wlan_shared_key, get_wlan_wpa_enterprise, get_wlan_wpa_enterprise_id, get_wlan_wpa_enterprise_pwd.

Remarks

[Shared Key]는 NULL을 포함해서 최대 64바이트입니다. 따라서 *out_shared_key*는 최소 64바이트의 메모리 공간을 가지고 있어야 합니다.

4.6.19 get_wlan_wpa_enterprise

```
int get_wlan_wpa_enterprise(unsigned char *mac_address);
```

Description

*get_wlan_wpa_enterprise*는 무선랜 제품의 보안 설정 중에서 [802.1X] 설정 값을 반환합니다.

Parameters

*unsigned char *mac_address*

EzTCP_Search 또는 *EzTCP_Read*를 사용하여 검색된 제품(ezTCP)의 MAC주소 입니다.

Return values

*get_wlan_wpa_enterprise*는 무선랜 제품의 [802.1X] 설정 값을 반환하며 그 값은 아래와 같습니다.

0

사용안함

1

EAP TLS

2

EAP TTLS

3

PEAP

*mac_address*을 가진 제품(ezTCP)을 찾을 수 없는 경우에는 -1을 반환합니다.

*is_wpa_enterprise*가 1을 반환하지 않는 무선랜 제품인 경우에는 -2를 반환합니다.

Examples

```

struct eztcp_info out_eztcp_info;
int ret = get_eztcp_info_by_index(0, &out_eztcp_info);

if(ret == EZTCP_SUCCESS)
{
    int _802_1x;
    unsigned char *mac_address = out_eztcp_info.mac_address;
    if(is_wlan(mac_address) == 1 && is_wlan_rsn(mac_address) == 1)
    {
        if(is_wpa_enterprise(mac_address) == 1)
        {
            _802_1x = get_wlan_wpa_enterprise(mac_address);
            if(_802_1x == 0)
                printf("[WLAN] 802.1X : Disable.\r\n");
            else if(_802_1x == 1)
                printf("[WLAN] 802.1X : EAP TLS.\r\n");
            else if(_802_1x == 2)
                printf("[WLAN] 802.1X : EAP TTLS.\r\n");
            else if(_802_1x == 3)
                printf("[WLAN] 802.1X : PEAP.\r\n");
        }
    }
}

```

See also

is_wlan, is_wlan_soft_ap, get_wlan_type, get_wlan_channel, get_wlan_ssid, is_wlan_antenna, get_wlan_antenna, is_wlan_phy_mode, get_wlan_phy_mode, get_wlan_short_preamble, get_wlan_short_slog, get_wlan_cts_protection, is_wlan_background_scan, get_wlan_background_scan, is_wlan_rsn, get_wlan_encryption_type, get_wlan_authentication_type, get_wlan_wep_key_length, get_wlan_wep_key_index, get_wlan_wep_key_data_type, get_wlan_wep_key, get_wlan_max_wpa_passphrase_length, get_wlan_wpa_passphrase, get_wlan_shared_key, is_wpa_enterprise, get_wlan_wpa_enterprise, get_wlan_wpa_enterprise_id, get_wlan_wpa_enterprise_pwd.

Remarks

4.6.20 get_wlan_wpa_enterprise_id

```
int get_wlan_wpa_enterprise_id(unsigned char *mac_address, char *out_enterprise_id);
```

Description



*get_wlan_wpa_enterprise_id*는 무선랜 제품의 보안 설정 중에서 802.1X [아이디]를 반환합니다.

Parameters

*unsigned char *mac_address*

EzTCP_Search 또는 *EzTCP_Read*를 사용하여 검색된 제품(ezTCP)의 MAC주소 입니다.

*char *out_enterprise_id*

802.1X [아이디]가 저장될 포인터입니다.

Return values

*get_wlan_wpa_enterprise_id*는 *out_enterprise_id*에 802.1X [아이디]를 저장하는데 성공하면 1을 반환합니다.

*mac_address*을 가진 제품(ezTCP)을 찾을 수 없는 경우에는 -1을 반환합니다.

*is_wpa_enterprise*가 1을 반환하지 않는 무선랜 제품인 경우에는 -2를 반환합니다.

Examples

```
struct eztcp_info out_eztcp_info;
int ret = get_eztcp_info_by_index(0, &out_eztcp_info);

if(ret == EZTCP_SUCCESS)
{
    int _802_1x;
    char buf[33];
    unsigned char *mac_address = out_eztcp_info.mac_address;
    if(is_wlan(mac_address) == 1 && is_wlan_rsn(mac_address) == 1)
    {
        if(is_wpa_enterprise(mac_address) == 1)
        {
            _802_1x = get_wlan_wpa_enterprise(mac_address);
            if(_802_1x > 0)
            {
                memset(buf, 0x00, 33);
                get_wlan_wpa_enterprise_id(mac_address, buf);
                printf("[WLAN] 802.1X ID : %s.\r\n", buf);
            }
        }
    }
}
```

See also

is_wlan, *is_wlan_soft_ap*, *get_wlan_type*, *get_wlan_channel*, *get_wlan_ssid*, *is_wlan_antenna*,
get_wlan_antenna, *is_wlan_phy_mode*, *get_wlan_phy_mode*, *get_wlan_short_preamble*,

get_wlan_short_slog, get_wlan_cts_protection, is_wlan_background_scan, get_wlan_background_scan,
 is_wlan_rsn, get_wlan_encryption_type, get_wlan_authentication_type, get_wlan_wep_key_length,
 get_wlan_wep_key_index, get_wlan_wep_key_data_type, get_wlan_wep_key,
 get_wlan_max_wpa_passphrase_length, get_wlan_wpa_passphrase, get_wlan_shared_key,
 is_wpa_enterprise, get_wlan_wpa_enterprise, get_wlan_wpa_enterprise_id, get_wlan_wpa_enterprise_pwd.

Remarks

802.1X [아이디]는 NULL을 제외하고 최대 32바이트입니다. 따라서 *out_enterprise_id*는 최소 33바이트의 메모리 공간을 가지고 있어야 합니다.

4.6.21 get_wlan_wpa_enterprise_pwd

```
int get_wlan_wpa_enterprise_pwd(unsigned char *mac_address,
                                char *out_enterprise_pwd);
```

Description

*get_wlan_wpa_enterprise_pwd*는 무선랜 제품의 보안 설정 중에서 802.1X [비밀번호]를 반환합니다.

Parameters

*unsigned char *mac_address*

EzTCP_Search 또는 *EzTCP_Read*를 사용하여 검색된 제품(ezTCP)의 MAC주소 입니다.

*char *out_enterprise_pwd*

802.1X [비밀번호]가 저장될 포인터입니다.

Return values

*get_wlan_wpa_enterprise_pwd*는 *out_enterprise_pwd*에 802.1X [비밀번호]를 저장하는데 성공하면 1을 반환합니다.

*mac_address*을 가진 제품(ezTCP)을 찾을 수 없는 경우에는 -1을 반환합니다.

*is_wpa_enterprise*가 1을 반환하지 않는 무선랜 제품인 경우에는 -2를 반환합니다.

Examples

```
struct eztcp_info out_eztcp_info;
int ret = get_eztcp_info_by_index(0, &out_eztcp_info);

if(ret == EZTCP_SUCCESS)
{
    int _802_1x;
    char buf[33];
    unsigned char *mac_address = out_eztcp_info.mac_address;
    if(is_wlan(mac_address) == 1 && is_wlan_rsn(mac_address) == 1)
```



```

    {
        if(is_wpa_enterprise(mac_address) == 1)
        {
            _802_1x = get_wlan_wpa_enterprise(mac_address);
            if(_802_1x >= 2)
            {
                memset(buf, 0x00, 33);
                get_wlan_wpa_enterprise_pwd(mac_address, buf);
                printf("[WLAN] 802.1X Password : %s.\r\n", buf);
            }
        }
    }
}

```

See also

is_wlan, is_wlan_soft_ap, get_wlan_type, get_wlan_channel, get_wlan_ssid, is_wlan_antenna, get_wlan_antenna, is_wlan_phy_mode, get_wlan_phy_mode, get_wlan_short_preamble, get_wlan_short_slog, get_wlan_cts_protection, is_wlan_background_scan, get_wlan_background_scan, is_wlan_rsn, get_wlan_encryption_type, get_wlan_authentication_type, get_wlan_wep_key_length, get_wlan_wep_key_index, get_wlan_wep_key_data_type, get_wlan_wep_key, get_wlan_max_wpa_passphrase_length, get_wlan_wpa_passphrase, get_wlan_shared_key, is_wpa_enterprise, get_wlan_wpa_enterprise, get_wlan_wpa_enterprise_id, get_wlan_wpa_enterprise_pwd.

Remarks

802.1X [비밀번호]는 NULL을 제외하고 최대 32바이트입니다. 따라서 *out_enterprise_pwd*는 최소 33바이트의 메모리 공간을 가지고 있어야 합니다.

4.7 부가기능

4.7.1 get_telnet

```
int get_telnet(unsigned char *mac_address);
```

Description

*get_telnet*은 [텔넷] 기능의 선택 여부를 반환합니다.

Parameters

*unsigned char *mac_address*

EzTCP_Search 또는 *EzTCP_Read*를 사용하여 검색된 제품(*ezTCP*)의 MAC주소 입니다.

Return values

*get_telnet*은 [텔넷] 기능이 선택되어 있으면 1을 반환하고 그렇지 않은 경우 0을 반환합니다.

*mac_address*을 가진 제품(ezTCP)을 찾을 수 없는 경우에는 -1을 반환합니다.

Examples

See also

`get_telnet`, `get_send_mac_address`, `get_ssl`, `get_ssh`, `get_ip4_address_search`, `get_remote_debug`,
`get_tcp_multi_connection`, `get_power_management`, `get_comment`

Remarks

4.7.2 get_send_mac_address

```
int get_send_mac_address(unsigned char *mac_address);
```

Description

*get_send_mac_address*는 [MAC 주소 전송] 기능의 선택 여부를 반환합니다.

Parameters

*unsigned char *mac_address*

EzTCP_Search 또는 *EzTCP_Read*를 사용하여 검색된 제품(ezTCP)의 MAC주소 입니다.

Return values

*get_send_mac_address*는 [MAC 주소 전송] 기능이 선택되어 있으면 1을 반환하고 그렇지 않은 경우 0을 반환합니다.

*mac_address*을 가진 제품(ezTCP)을 찾을 수 없는 경우에는 -1을 반환합니다.

Examples

See also

`get_telnet`, `is_send_mac_address`, `get_send_mac_address`, `get_ssl`, `get_ssh`, `get_ip4_address_search`,
`get_remote_debug`, `get_tcp_multi_connection`, `get_power_management`, `get_comment`

Remarks

4.7.3 get_ssl

```
int get_ssl(unsigned char *mac_address);
```

Description

*get_ssl*은 [SSL 보안통신] 기능의 선택 여부를 반환합니다.

Parameters

*unsigned char *mac_address*

EzTCP_Search 또는 *EzTCP_Read*를 사용하여 검색된 제품(ezTCP)의 MAC주소 입니다.

Return values

*get_ssl*은 [SSL 보안통신] 기능이 선택되어 있으면 1을 반환하고 그렇지 않은 경우 0을 반환합니다.

*mac_address*을 가진 제품(ezTCP)을 찾을 수 없는 경우에는 -1을 반환합니다.

Examples

See also

get_telnet, *get_send_mac_address*, *is_ssl*, *get_ssl*, *get_ssh*, *get_ip4_address_search*, *get_remote_debug*, *get_tcp_multi_connection*, *get_power_management*, *get_comment*

Remarks

4.7.4 get_ssh

```
int get_ssh(unsigned char *mac_address);
```

Description

*get_ssh*는 [SSH 보안통신] 기능의 선택 여부를 반환합니다.

Parameters

*unsigned char *mac_address*

EzTCP_Search 또는 *EzTCP_Read*를 사용하여 검색된 제품(ezTCP)의 MAC주소 입니다.

Return values

*get_ssh*는 [SSH 보안통신] 기능이 선택되어 있으면 1을 반환하고 그렇지 않은 경우 0을 반환합니다.

*mac_address*을 가진 제품(ezTCP)을 찾을 수 없는 경우에는 -1을 반환합니다.

Examples

See also

get_telnet, *get_send_mac_address*, *get_ssl*, *is_ssh*, *get_ssh*, *get_ip4_address_search*, *get_remote_debug*,

get_tcp_multi_connection, get_power_management, get_comment

Remarks

4.7.5 get_ip4_address_search

```
int get_ip4_address_search(unsigned char *mac_address);
```

Description

*get_ip4_address_search*는 [IPv4 주소 검색] 기능의 선택 여부를 반환합니다.

Parameters

*unsigned char *mac_address*

EzTCP_Search 또는 *EzTCP_Read*를 사용하여 검색된 제품(ezTCP)의 MAC주소 입니다.

Return values

*get_ip4_address_search*는 [IPv4 주소 검색] 기능이 선택되어 있으면 1을 반환하고 그렇지 않은 경우 0을 반환합니다.

*mac_address*을 가진 제품(ezTCP)을 찾을 수 없는 경우에는 -1을 반환합니다.

Examples

See also

get_telnet, get_send_mac_address, get_ssl, get_ssh, get_ip4_address_search, get_remote_debug, get_tcp_multi_connection, get_power_management, get_comment

Remarks

4.7.6 get_remote_debug

```
int get_remote_debug(unsigned char *mac_address);
```

Description

*get_remote_debug*는 [디버깅 로그 보기] 기능의 선택 여부를 반환합니다.

Parameters

*unsigned char *mac_address*

EzTCP_Search 또는 *EzTCP_Read*를 사용하여 검색된 제품(ezTCP)의 MAC주소 입니다.

Return values

*get_remote_debug*는 [디버깅 로그 보기] 기능이 선택되어 있으면 1을 반환하고 그렇지 않은 경우 0을 반환합니다.

*mac_address*을 가진 제품(ezTCP)을 찾을 수 없는 경우에는 -1을 반환합니다.

Examples**See also**

get_telnet, *get_send_mac_address*, *get_ssl*, *get_ssh*, *get_ip4_address_search*, *is_remote_debug*, *get_remote_debug*, *get_tcp_multi_connection*, *get_power_management*, *get_comment*

Remarks**4.7.7 get_tcp_multi_connection**

```
int get_tcp_multi_connection(unsigned char *mac_address);
```

Description

*get_tcp_multi_connection*는 [다중 접속] 기능의 선택 여부를 반환합니다.

Parameters

*unsigned char *mac_address*

EzTCP_Search 또는 *EzTCP_Read*를 사용하여 검색된 제품(ezTCP)의 MAC주소 입니다.

Return values

*get_tcp_multi_connection*는 [다중 접속] 기능이 선택되어 있으면 1을 반환하고 그렇지 않은 경우 0을 반환합니다.

*mac_address*을 가진 제품(ezTCP)을 찾을 수 없는 경우에는 -1을 반환합니다.

Examples**See also**

get_telnet, *get_send_mac_address*, *get_ssl*, *get_ssh*, *get_ip4_address_search*, *get_remote_debug*, *is_tcp_multi_connection*, *get_tcp_multi_connection*, *get_power_management*, *get_comment*

Remarks

4.7.8 get_power_management

```
int get_power_management(unsigned char *mac_address);
```

Description

*get_power_management*는 [전원 관리] 기능의 선택 여부를 반환합니다.

Parameters

*unsigned char *mac_address*

EzTCP_Search 또는 *EzTCP_Read*를 사용하여 검색된 제품(ezTCP)의 MAC주소 입니다.

Return values

*get_power_management*는 [전원 관리] 기능이 선택되어 있으면 1을 반환하고 그렇지 않은 경우 0을 반환합니다.

*is_power_management*가 1을 반환하지 않는 경우 -2를 반환합니다.

*mac_address*을 가진 제품(ezTCP)을 찾을 수 없는 경우에는 -1을 반환합니다.

Examples

See also

get_telnet, *get_send_mac_address*, *get_ssl*, *get_ssh*, *get_ip4_address_search*, *get_remote_debug*,
get_tcp_multi_connection, *is_power_management*, *get_power_management*, *get_comment*

Remarks

4.7.9 get_comment

```
int get_comment(unsigned char *mac_address, char *out_comment);
```

Description

*get_comment*는 제품(ezTCP)의 [설명]을 반환합니다.

Parameters

*unsigned char *mac_address*

EzTCP_Search 또는 *EzTCP_Read*를 사용하여 검색된 제품(ezTCP)의 MAC주소 입니다.

*char *out_comment*

제품(ezTCP)의 [설명]이 저장될 포인터입니다.

Return values

*get_telnet*은 *out_comment*에 제품(ezTCP)의 [설명]을 저장하는데 성공하면 1을 반환합니다.
*mac_address*을 가진 제품(ezTCP)을 찾을 수 없는 경우에는 -1을 반환합니다.

Examples

```
struct eztcp_info out_eztcp_info;
int ret = get_eztcp_info_by_index(0, &out_eztcp_info);

if(ret == EZTCP_SUCCESS)
{
    char buf[65];
    memset(buf, 0x00, 65);
    get_comment(out_eztcp_info.mac_address, buf);
    printf("[ezTCP] Comment : %s.\r\n", buf);
}
```

See also

get_telnet, *get_send_mac_address*, *get_ssl*, *get_ssh*, *get_ip4_address_search*, *get_remote_debug*,
get_tcp_multi_connection, *get_power_management*, *get_comment*

Remarks

제품(ezTCP)의 [설명]은 NULL을 제외하고 최대 64바이트입니다. 따라서 *out_comment*는 최소 65바이트의 메모리 공간을 가지고 있어야 합니다.

4.7.10 get_allowed_ezmanager

```
int get_allowed_ezmanager(unsigned char *mac_address);
```

Description

*get_allowed_ezmanager*은 여러 가지 제품(ezTCP) 접근 제한 기능 중에서 [ezManager에도 적용]의 선택 여부를 반환합니다.

Parameters

*unsigned char *mac_address*

EzTCP_Search 또는 *EzTCP_Read*를 사용하여 검색된 제품(ezTCP)의 MAC주소 입니다.

Return values

*get_allowed_ezmanager*은 제품(ezTCP) 접근 제한 기능 중 [ezManager에도 적용]이 선택되어 있으면 1을 반환하고 그렇지 않으면 0을 반환합니다.

*mac_address*을 가진 제품(ezTCP)을 찾을 수 없는 경우에는 -1을 반환합니다.

Examples

```
struct eztcp_info out_eztcp_info;
int ret = get_eztcp_info_by_index(0, &out_eztcp_info);

if(ret == EZTCP_SUCCESS)
{
    unsigned char *mac_address = out_eztcp_info.mac_address;

    if(get_allowed_ezmanager(mac_address) == 1)
        printf("[ezTCP Firewall] Apply To ezManager is enabled.\r\n");
    else
        printf("[ezTCP Firewall] Apply To ezManager is disabled.\r\n");
}
```

See also

get_allowed_ezmanager, get_allowed_mac_address, get_allowed_ip4_address,
get_allowed_ip4_network_mask_type, get_allowed_ip6_address, get_allowed_ip6_subnet_prefix_length

Remarks

제품(ezTCP) 접근 제한에 대한 보다 자세한 내용은 해당 제품 사용설명서를 참조하시기 바랍니다.

4.7.11 get_allowed_mac_address

```
int get_allowed_mac_address(unsigned char *mac_address, int mac_index,
                           char *out_mac_address);
```

Description

*get allowed mac address*는 여러 가지 제품(ezTCP) 접근 제한 기능 중에서 [다음의 MAC 주소만 접근 가능]이 선택된 경우 입력되어있는 MAC 주소를 반환합니다.

Parameters

*unsigned char *mac_address*

EzTCP_Search 또는 *EzTCP_Read*를 사용하여 검색된 제품(ezTCP)의 MAC주소 입니다.

int mac_index

입력되어있는 MAC 주소 6바이트 중에서 읽고자 하는 MAC 주소의 순번으로 0 ~ 5입니다.

*char *out_mac_address*

*mac_index*로 지정된 MAC 주소가 저장될 포인터입니다.

Return values

*get_allowed_mac_address*은 제품(ezTCP) 접근 제한 기능 중 [다음의 MAC 주소만 접근 가능]에 입력되어있는 MAC 주소를 *out_mac*에 저장하는데 성공하면 1을 반환합니다.

*mac_index*가 지정된 범위를 벗어나면 -2를 반환합니다.

*mac_address*을 가진 제품(ezTCP)을 찾을 수 없는 경우에는 -1을 반환합니다.

Examples

```
struct eztcp_info out_eztcp_info;
int ret = get_eztcp_info_by_index(0, &out_eztcp_info);

if(ret == EZTCP_SUCCESS)
{
    int idx;
    char buf[3];
    unsigned char *mac_address = out_eztcp_info.mac_address;

    printf("[ezTCP Firewall] Allowed MAC Address : ");
    for(idx = 0; idx < 6; idx++)
    {
        memset(buf, 0x00, 3);
        get_allowed_mac_address(mac_address, idx, buf);
        printf("%s ", buf);
    }
    printf("\r\n");
}
```

See also

get_allowed_ezmanager, *get_allowed_mac_address*, *get_allowed_ip4_address*,
get_allowed_ip4_network_mask_type, *get_allowed_ip6_address*, *get_allowed_ip6_subnet_prefix_length*

Remarks

제품(ezTCP) 접근 제한에 대한 보다 자세한 내용은 해당 제품 사용설명서를 참조하시기 바랍니다.

4.7.12 get_allowed_ip4_address

```
int get_allowed_ip4_address(unsigned char *mac_address, char *out_address);
```

Description

*get_allowed_ip4_address*는 여러 가지 제품(ezTCP) 접근 제한 기능 중에서 [다음의 IP 주소 대역만 접근 가능]의 선택된 경우 입력되어있는 [IPv4 주소]를 반환합니다.

Parameters

*unsigned char *mac_address*

EzTCP_Search 또는 *EzTCP_Read*를 사용하여 검색된 제품(ezTCP)의 MAC주소 입니다.

*char *out_address*

IPv4 주소가 저장될 포인터입니다.

Return values

*get_allowed_ip4_address*은 [다음의 IP 주소 대역만 접근 가능]에 입력되어있는 [IPv4 주소]를 *out_address*에 저장하는데 성공하면 1을 반환하고 그렇지 않으면 0을 반환합니다.

*mac_address*을 가진 제품(ezTCP)을 찾을 수 없는 경우에는 -1을 반환합니다.

Examples

```
struct eztcp_info out_eztcp_info;
int ret = get_eztcp_info_by_index(0, &out_eztcp_info);

if(ret == EZTCP_SUCCESS)
{
    int idx;
    int networkmask_type;
    char buf[16];
    unsigned char *mac_address = out_eztcp_info.mac_address;
    printf("[ezTCP Firewall] Allowed IP Range\r\n");

    memset(buf, 0x00, 16);
    get_allowed_ip4_address(mac_address, buf);
    printf("IPv4 Address : %s\r\n", buf);

    networkmask_type = get_allowed_ip4_network_mask_type(mac_address);
    switch(networkmask_type)
    {
        case 0:
            printf("Network Mask : 255.255.255.255\r\n");
            break;
        case 1:
            printf("Network Mask : 255.255.255.0\r\n");
            break;
        case 2:
            printf("Network Mask : 255.255.0.0\r\n");
            break;
        case 3:
            printf("Network Mask : 255.0.0.0\r\n");
    }
}
```

```

        break;
    case 4:
        printf("Network Mask : 0.0.0.0\r\n");
        break;
    }
}

```

See also

get_allowed_ezmanager, get_allowed_mac_address, get_allowed_ip4_address,
get_allowed_ip4_network_mask_type, get_allowed_ip6_address, get_allowed_ip6_subnet_prefix_length

Remarks

제품(ezTCP) 접근 제한에 대한 보다 자세한 내용은 해당 제품 사용설명서를 참조하시기 바랍니다.

4.7.13 get_allowed_ip4_network_mask_type

```
int get_allowed_ip4_network_mask_type(unsigned char *mac_address);
```

Description

*get_allowed_ip4_network_mask_type*은 여러 가지 제품(ezTCP) 접근 제한 기능 중에서 [다음의 IP 주소 대역만 접근 가능]의 선택된 경우 입력되어있는 [넷 마스크(IPv4 주소 대역)]의 형태를 반환합니다.

Parameters

*unsigned char *mac_address*

EzTCP_Search 또는 *EzTCP_Read*를 사용하여 검색된 제품(ezTCP)의 MAC주소 입니다.

Return values

*get_allowed_ip4_network_mask_type*은 [다음의 IP 주소 대역만 접근 가능]에 입력되어있는 [넷 마스크(IPv4 주소 대역)]의 형태를 반환하며 그 값은 아래와 같습니다.

0	255.255.255.255
1	255.255.255.0
2	255.255.0.0
3	255.0.0.0
4	

0.0.0.0

*mac_address*을 가진 제품(ezTCP)을 찾을 수 없는 경우에는 -1을 반환합니다.

Examples

```
struct eztcp_info out_eztcp_info;
int ret = get_eztcp_info_by_index(0, &out_eztcp_info);

if(ret == EZTCP_SUCCESS)
{
    int network_mask_type;
    char buf[16];
    unsigned char *mac_address = out_eztcp_info.mac_address;

    printf("[ezTCP Firewall] Allowed IP Range\r\n");

    memset(buf, 0x00, 16);
    get_allowed_ip4_address(mac_address, buf);
    printf("IPv4 Address : %s\r\n", buf);

    network_mask_type = get_allowed_ip4_network_mask_type(mac_address);
    switch(networkmask_type)
    {
        case 0:
            printf("Network Mask : 255.255.255.255\r\n");
            break;
        case 1:
            printf("Network Mask : 255.255.255.0\r\n");
            break;
        case 2:
            printf("Network Mask : 255.255.0.0\r\n");
            break;
        case 3:
            printf("Network Mask : 255.0.0.0\r\n");
            break;
        case 4:
            printf("Network Mask : 0.0.0.0\r\n");
            break;
    }
}
```

See also



get_allowed_ezmanager, get_allowed_mac_address, get_allowed_ip4_address,
get_allowed_ip4_network_mask_type, get_allowed_ip6_address, get_allowed_ip6_subnet_prefix_length

Remarks

제품(ezTCP) 접근 제한에 대한 보다 자세한 내용은 해당 제품 사용설명서를 참조하시기 바랍니다.

4.7.14 get_allowed_ip6_address

```
int get_allowed_ip6_address(unsigned char *mac_address, char *out_address);
```

Description

*get_allowed_ip6_address*는 여러 가지 제품(ezTCP) 접근 제한 기능 중에서 [다음의 IP 주소 대역만 접근 가능]의 선택된 경우 입력되어있는 [IPv6 주소]를 반환합니다.

Parameters

*unsigned char *mac_address*

EzTCP_Search 또는 *EzTCP_Read*를 사용하여 검색된 제품(ezTCP)의 MAC주소 입니다.

*char *out_address*

IPv6 주소가 저장될 포인터입니다.

Return values

*get_allowed_ip6_address*은 [다음의 IP 주소 대역만 접근 가능]에 입력되어있는 [IPv6 주소]를 *out_address*에 저장하는데 성공하면 1을 반환합니다.

*mac_address*을 가진 제품(ezTCP)을 찾을 수 없는 경우에는 -1을 반환합니다.

Examples

```
struct eztcp_info out_eztcp_info;
int ret = get_eztcp_info_by_index(0, &out_eztcp_info);

if(ret == EZTCP_SUCCESS)
{
    char buf[128];
    unsigned char *mac_address = out_eztcp_info.mac_address;

    printf("[ezTCP Firewall] Allowed IP Range\r\n");

    memset(buf, 0x00, 128);
    get_allowed_ip6_address(mac_address, buf);
    printf("IPv6 Address : %s\r\n", buf);
}
```

```

memset(buf, 0x00, 128);
get_allowed_ip6_subnet_prefix_length(mac_address, buf);
printf("\tSubnet Prefix Length : %s\r\n", buf);
}

```

See also

get_allowed_ezmanager, get_allowed_mac_address, get_allowed_ip4_address,
get_allowed_ip4_network_mask_type, get_allowed_ip6_address, get_allowed_ip6_subnet_prefix_length

Remarks

제품(ezTCP) 접근 제한에 대한 보다 자세한 내용은 해당 제품 사용설명서를 참조하시기 바랍니다.

4.7.15 get_allowed_ip6_subnet_prefix_length

```

int get_allowed_ip6_subnet_prefix_length(unsigned char *mac_address,
                                         char *out_subnet_prefix_length);

```

Description

*get_allowed_ip6_subnet_prefix_length*는 여러 가지 제품(ezTCP) 접근 제한 기능 중에서 [다음의 IP 주소 대역만 접근 가능]이 선택된 경우 입력되어있는 [IPv6 주소]의 서브넷 접두사 길이를 반환합니다.

Parameters

*unsigned char *mac_address*

EzTCP_Search 또는 *EzTCP_Read*를 사용하여 검색된 제품(ezTCP)의 MAC주소 입니다.

*char *out_subnet_prefix_length*

서브넷 접두사 길이가 저장될 포인터입니다.

Return values

*get_allowed_ip6_subnet_prefix_length*는 [다음의 IP 주소 대역만 접근 가능]에 입력되어있는 [IPv6 주소]의 서브넷 접두사 길이를 *out_subnet_prefix_length*에 저장하는데 성공하면 1을 반환합니다.

*mac_address*을 가진 제품(ezTCP)을 찾을 수 없는 경우에는 -1을 반환합니다.

Examples

```

struct eztcp_info out_eztcp_info;
int ret = get_eztcp_info_by_index(0, &out_eztcp_info);

```

```

if(ret == EZTCP_SUCCESS)
{
    char buf[128];
    unsigned char *mac_address = out_eztcp_info.mac_address;

    printf("[ezTCP Firewall] Allowed IP Range\r\n");

    memset(buf, 0x00, 128);
    get_allowed_ip6_address(mac_address, buf);
    printf("IPv6 Address : %s\r\n", buf);

    memset(buf, 0x00, 128);
    get_allowed_ip6_subnet_prefix_length(mac_address, buf);
    printf("\tSubnet Prefix Length : %s\r\n", buf);
}

```

See also

get_allowed_ezmanager, get_allowed_mac_address, get_allowed_ip4_address,
get_allowed_ip4_network_mask_type, get_allowed_ip6_address, get_allowed_ip6_subnet_prefix_length

Remarks

제품(ezTCP) 접근 제한에 대한 보다 자세한 내용은 해당 제품 사용설명서를 참조하시기 바랍니다.

4.7.16 get_ip4_change_notification_type

```
int get_ip4_change_notification_type(unsigned char *mac_address);
```

Description

*get_ip4_change_notification_type*은 [IPv4 주소 통보]기능의 [프로토콜]을 반환합니다.

Parameters

*unsigned char *mac_address*

EzTCP_Search 또는 *EzTCP_Read*를 사용하여 검색된 제품(ezTCP)의 MAC주소 입니다.

Return values

*get_ip4_change_notification_type*은 [IPv4 주소 통보]기능의 [프로토콜]을 반환하며 그 값은 아래와 같습니다.

0

사용안함

1

DDNS(dyndns.org)

2

TCP

3

UDP

*mac_address*을 가진 제품(ezTCP)을 찾을 수 없는 경우에는 -1을 반환합니다.

Examples

```
struct eztcp_info out_eztcp_info;
int ret = get_eztcp_info_by_index(0, &out_eztcp_info);

if(ret == EZTCP_SUCCESS)
{
    unsigned char *mac_address = out_eztcp_info.mac_address;

    printf("[Notify IPv4 Change] Protocol : ");

    int protocol_type = get_ip4_change_notification_type(mac_address);
    switch(protocol_type)
    {
        case 0:
            printf("Disable\r\n");
            break;
        case 1:
            printf("DDNS(dyndns.org)\r\n");
            break;
        case 2:
            printf("TCP\r\n");
            break;
        case 3:
            printf("UDP\r\n");
            break;
    }
}
```

See also

get_ip4_change_notification_type,
get_ip4_change_notification_interval,
get_ip4_change_notification_peer_address,
get_ip4_change_notification_ddns_pwd.

get_ip4_change_notification_data_type,
get_ip4_change_notification_peer_port,
get_ip4_change_notification_ddns_id,

Remarks

[IPv4 주소 통보]기능에 대한 보다 자세한 내용은 해당 제품 사용설명서를 참조하시기 바랍니다.

4.7.17 get_ip4_change_notification_data_type

```
int get_ip4_change_notification_data_type(unsigned char *mac_address);
```

Description

*get_ip4_change_notification_data_type*은 [IPv4 주소 통보]기능의 [데이터 형식]을 반환합니다.

Parameters

*unsigned char *mac_address*

EzTCP_Search 또는 *EzTCP_Read*를 사용하여 검색된 제품(ezTCP)의 MAC주소 입니다.

Return values

*get_ip4_change_notification_data_type*은 [IPv4 주소 통보]기능의 [데이터 형식]을 반환하며 그 값은 아래와 같습니다.

0

ASCII

1

16진수

*mac_address*을 가진 제품(ezTCP)을 찾을 수 없는 경우에는 -1을 반환합니다.

Examples

```
struct eztcp_info out_eztcp_info;
int ret = get_eztcp_info_by_index(0, &out_eztcp_info);

if(ret == EZTCP_SUCCESS)
{
    unsigned char *mac_address = out_eztcp_info.mac_address;

    int data_type;
    int protocol_type = get_ip4_change_notification_type(mac_address);
    if(protocol_type == 2 || protocol_type == 3)
    {
        printf("[Notify IPv4 Change] Data Type : ");
        data_type = get_ip4_change_notification_data_type(mac_address);
        if(data_type == 0)
```

```

        printf("ASCII\r\n");
    else if(data_type == 1)
        printf("HEX\r\n");
    }
}

```

See also

get_ip4_change_notification_type,
get_ip4_change_notification_interval,
get_ip4_change_notification_peer_address,
get_ip4_change_notification_ddns_pwd.

get_ip4_change_notification_data_type,
get_ip4_change_notification_peer_port,
get_ip4_change_notification_ddns_id,

Remarks

[IPv4 주소 통보]기능에 대한 보다 자세한 내용은 해당 제품 사용설명서를 참조하시기 바랍니다.

4.7.18 get_ip4_change_notification_interval

```

int get_ip4_change_notification_interval(unsigned char *mac_address, char
*out_interval);

```

Description

*get_ip4_change_notification_interval*은 [IPv4 주소 통보]기능의 [통보 주기]를 반환합니다.

Parameters

*unsigned char *mac_address*

EzTCP_Search 또는 *EzTCP_Read*를 사용하여 검색된 제품(ezTCP)의 MAC주소 입니다.

*char *out_interval*

[통보 주기]가 저장될 포인터입니다.

Return values

*get_ip4_change_notification_interval*은 *out_interval*에 IPv4 주소 통보 기능의 [통보 주기]를 저장하는데 성공하면 1을 반환합니다.

*mac_address*을 가진 제품(ezTCP)을 찾을 수 없는 경우에는 -1을 반환합니다.

Examples

```

struct eztcp_info out_eztcp_info;
int ret = get_eztcp_info_by_index(0, &out_eztcp_info);

if(ret == EZTCP_SUCCESS)
{

```

```

    unsigned char *mac_address = out_eztcp_info.mac_address;

    char buf[8];
    int protocol_type = get_ip4_change_notification_type(mac_address);
    if(protocol_type == 2 || protocol_type == 3)
    {
        get_ip4_change_notification_interval(mac_address, buf);
        printf("[Notify IPv4 Change] Interval : %s\r\n", buf);
    }
}

```

See also

get_ip4_change_notification_type,	get_ip4_change_notification_data_type,
get_ip4_change_notification_interval,	get_ip4_change_notification_peer_port,
get_ip4_change_notification_peer_address,	get_ip4_change_notification_ddns_id,
get_ip4_change_notification_ddns_pwd.	

Remarks

[IPv4 주소 통보]기능에 대한 보다 자세한 내용은 해당 제품 사용설명서를 참조하시기 바랍니다.

4.7.19 get_ip4_change_notification_peer_port

```
int get_ip4_change_notification_peer_port(unsigned char *mac_address, char *out_port);
```

Description

*get_ip4_change_notification_peer_port*은 [IPv4 주소 통보]기능의 [포트]를 반환합니다.

Parameters

*unsigned char *mac_address*

EzTCP_Search 또는 *EzTCP_Read*를 사용하여 검색된 제품(ezTCP)의 MAC주소 입니다.

*char *out_port*

[포트]가 저장될 포인터입니다.

Return values

*get_ip4_change_notification_peer_port*은 *out_port*에 [IPv4 주소 통보]기능의 [포트]를 저장하는데 성공하면 1을 반환합니다.

*mac_address*을 가진 제품(ezTCP)을 찾을 수 없는 경우에는 -1을 반환합니다.

Examples

```

struct eztcp_info out_eztcp_info;
int ret = get_eztcp_info_by_index(0, &out_eztcp_info);

if(ret == EZTCP_SUCCESS)
{
    unsigned char *mac_address = out_eztcp_info.mac_address;

    char buf[8];
    int protocol_type = get_ip4_change_notification_type(mac_address);
    if(protocol_type == 2 || protocol_type == 3)
    {
        get_ip4_change_notification_peer_port(mac_address, buf);
        printf("[Notify IPv4 Change] Port Number : %s\r\n", buf);
    }
}

```

See also

get_ip4_change_notification_type,
get_ip4_change_notification_interval,
get_ip4_change_notification_peer_address,
get_ip4_change_notification_ddns_pwd.

get_ip4_change_notification_data_type,
get_ip4_change_notification_peer_port,
get_ip4_change_notification_ddns_id,

Remarks

[IPv4 주소 통보]기능에 대한 보다 자세한 내용은 해당 제품 사용설명서를 참조하시기 바랍니다.

4.7.20 get_ip4_change_notification_peer_address

```

int get_ip4_change_notification_peer_address(unsigned char *mac_address,
                                             char *out_address);

```

Description

*get_ip4_change_notification_peer_address*은 [IPv4 주소 통보]기능의 [호스트 이름] 또는 [통보할 주소]를 반환합니다.

Parameters

*unsigned char *mac_address*

EzTCP_Search 또는 *EzTCP_Read*를 사용하여 검색된 제품(ezTCP)의 MAC주소 입니다.

*char *out_address*

[호스트 이름] 또는 [통보할 주소]가 저장될 포인터입니다.

Return values

*get_ip4_change_notification_peer_address*는 *out_address*에 [호스트 이름] 또는 [통보할 주소]를 저장하는데 성공하면 1을 반환하고 그렇지 않으면 0을 반환합니다.

*mac_address*을 가진 제품(ezTCP)을 찾을 수 없는 경우에는 -1을 반환합니다.

Examples

```
struct eztcp_info out_eztcp_info;
int ret = get_eztcp_info_by_index(0, &out_eztcp_info);

if(ret == EZTCP_SUCCESS)
{
    unsigned char *mac_address = out_eztcp_info.mac_address;

    char buf[64];
    int protocol_type = get_ip4_change_notification_type(mac_address);
    memset(buf, 0x00, 64);
    get_ip4_change_notification_peer_address(mac_address, buf);

    if(protocol == 1)
    {
        printf("[Notify IPv4 Change] Host Name (dyndns) : %s\r\n", buf);
    }
    else if(protocol_type == 2 || protocol_type == 3)
    {
        printf("[Notify IPv4 Change] Host Name (custom) : %s\r\n", buf);
    }
}
```

See also

get_ip4_change_notification_type,
get_ip4_change_notification_interval,
get_ip4_change_notification_peer_address,
get_ip4_change_notification_ddns_pwd.

get_ip4_change_notification_data_type,
get_ip4_change_notification_peer_port,
get_ip4_change_notification_ddns_id,

Remarks

[IPv4 주소 통보]기능의 [프로토콜]이 1(DDNS(dyndns.org))인 경우 [호스트 이름]으로 사용되고 2(TCP) 또는 3(UDP)인 경우에만 [통보할 주소]로 사용됩니다.

[호스트 이름] 또는 [통보할 주소]는 NULL을 포함하여 최대 64바이트입니다. 따라서 *out_address*는 최소 64바이트의 메모리 공간을 가지고 있어야 합니다.

[IPv4 주소 통보]기능에 대한 보다 자세한 내용은 해당 제품 사용설명서를 참조하시기 바랍니다.

4.7.21 get_ip4_change_notification_ddns_id

```
int get_ip4_change_notification_ddns_id(unsigned char *mac_address, char *out_ddns_id);
```

Description

get_ip4_change_notification_ddns_id는 IPv4 주소 통보 기능의 [DDNS 아이디]를 반환합니다.

Parameters

*unsigned char *mac_address*

EzTCP_Search 또는 *EzTCP_Read*를 사용하여 검색된 제품(ezTCP)의 MAC주소 입니다.

*char *out_ddns_id*

[DDNS 아이디]가 저장될 포인터입니다.

Return values

*get_ip4_change_notification_ddns_id*는 *out_ddns_id*에 [DDNS 아이디]를 저장하는데 성공하면 1을 반환합니다.

*mac_address*을 가진 제품(ezTCP)을 찾을 수 없는 경우에는 -1을 반환합니다.

Examples

```
struct eztcp_info out_eztcp_info;
int ret = get_eztcp_info_by_index(0, &out_eztcp_info);

if(ret == EZTCP_SUCCESS)
{
    unsigned char *mac_address = out_eztcp_info.mac_address;

    char buf[33];
    int protocol_type = get_ip4_change_notification_type(mac_address);
    if(protocol == 1)
    {
        memset(buf, 0x00, 33);
        get_ip4_change_notification_ddns_id(mac_address, buf);

        printf("[Notify IPv4 Change] DDNS ID : %s\r\n", buf);
    }
}
```

See also

get_ip4_change_notification_type,

get_ip4_change_notification_data_type,

```
get_ip4_change_notification_interval,  
get_ip4_change_notification_peer_address,  
get_ip4_change_notification_ddns_pwd.
```

```
get_ip4_change_notification_peer_port,  
get_ip4_change_notification_ddns_id,
```

Remarks

[DDNS 아이디]는 NULL을 제외하고 최대 32바이트입니다. 따라서 *out_ddns_id*는 최소 33바이트의 메모리 공간을 가지고 있어야 합니다.

[IPv4 주소 통보]기능에 대한 보다 자세한 내용은 해당 제품 사용설명서를 참조하시기 바랍니다.

4.7.22 get_ip4_change_notification_ddns_pwd

```
int get_ip4_change_notification_ddns_pwd(unsigned char *mac_address,  
char *out_ddns_pwd);
```

Description

*get_ip4_change_notification_ddns_pwd*는 [IPv4 주소 통보]기능의 [DDNS 비밀번호]를 반환합니다.

Parameters

*unsigned char *mac_address*

EzTCP_Search 또는 *EzTCP_Read*를 사용하여 검색된 제품(ezTCP)의 MAC주소 입니다.

*char *out_ddns_pwd*

[DDNS 비밀번호]가 저장될 포인터입니다.

Return values

*get_ip4_change_notification_ddns_pwd*는 *out_ddns_pwd*에 [DDNS 비밀번호]를 저장하는데 성공하면 1을 반환합니다.

*mac_address*을 가진 제품(ezTCP)을 찾을 수 없는 경우에는 -1을 반환합니다.

Examples

```
struct eztcp_info out_eztcp_info;  
int ret = get_eztcp_info_by_index(0, &out_eztcp_info);  
  
if(ret == EZTCP_SUCCESS)  
{  
    unsigned char *mac_address = out_eztcp_info.mac_address;  
  
    char buf[17];  
    int protocol_type = get_ip4_change_notification_type(mac_address);  
    if(protocol == 1)
```

```

    {
        memset(buf, 0x00, 17);
        get_ip4_change_notification_ddns_pwd(mac_address, buf);

        printf("[Notify IPv4 Change] DDNS Password : %s\r\n", buf);
    }
}

```

See also

get_ip4_change_notification_type,
get_ip4_change_notification_interval,
get_ip4_change_notification_peer_address,
get_ip4_change_notification_ddns_pwd.

get_ip4_change_notification_data_type,
get_ip4_change_notification_peer_port,
get_ip4_change_notification_ddns_id,

Remarks

[**DDNS 비밀번호**]는 NULL을 제외하고 최대 16바이트입니다. 따라서 *out_ddns_pwd*는 최소 17바이트의 메모리 공간을 가지고 있어야 합니다.

[**IPv4 주소 통보**]기능에 대한 보다 자세한 내용은 해당 제품 사용설명서를 참조하시기 바랍니다.

5 설정 값 변경

5.1 네트워크

5.1.1 set_arp

```
int set_arp(unsigned char *mac_address, int onoff);
```

Description

*set_arp*는 [처음 수신된 패킷에서 IP 주소 얻기] 설정 값을 변경합니다.

Parameters

*unsigned char *mac_address*

EzTCP_Search 또는 *EzTCP_Read*를 사용하여 검색된 제품(ezTCP)의 MAC주소 입니다.

int onoff

해당 기능의 선택 여부를 결정하며 아래와 같은 값을 가질 수 있습니다.

0

기능 선택을 해제합니다.

1

기능을 선택합니다.

Return values

*set_arp*는 설정 값 변경에 성공하면 1을 반환합니다.

*mac_address*을 가진 제품(ezTCP)을 찾을 수 없는 경우에는 -1을 반환합니다.

Examples

See also

Remarks

5.1.2 set_dhcp

```
int set_dhcp(unsigned char *mac_address, int onoff);
```

Description

*get_dhcp*는 [자동으로 IP 주소 받기(DHCP)] 설정 값을 변경합니다.

Parameters*unsigned char *mac_address**EzTCP_Search* 또는 *EzTCP_Read*를 사용하여 검색된 제품(ezTCP)의 MAC주소 입니다.*int onoff*

해당 기능의 선택 여부를 결정하며 아래와 같은 값을 가질 수 있습니다.

0

기능 선택을 해제합니다.

1

기능을 선택합니다.

Return values*set_dhcp*는 설정 값 변경에 성공하면 1을 반환합니다.*mac_address*을 가진 제품(ezTCP)을 찾을 수 없는 경우에는 -1을 반환합니다.**Examples****See also***set_dhcp_dns*, *set_pppoe***Remarks**

[자동으로 IP 주소 받기(DHCP)]와 [아이디로 IP 주소 받기(PPPoE)]는 동시에 설정할 수 없습니다. 따라서 *set_dhcp*의 함수 입력값인 *onoff*에 1을 입력하는 경우 [아이디로 IP 주소 받기(PPPoE)]는 자동으로 선택이 해제됩니다.

5.1.3 set_dhcp_dns

```
int set_dhcp_dns(unsigned char *mac_address, int onoff);
```

Description*set_dhcp_dns*는 [자동으로 DNS 서버 주소 받기] 설정 값을 변경합니다.**Parameters***unsigned char *mac_address**EzTCP_Search* 또는 *EzTCP_Read*를 사용하여 검색된 제품(ezTCP)의 MAC주소 입니다.*int onoff*

해당 기능의 선택 여부를 결정하며 아래와 같은 값을 가질 수 있습니다.

0

기능 선택을 해제합니다.

1

기능을 선택합니다.

Return values*set_dhcp_dns*는 설정 값 변경에 성공하면 1을 반환합니다.*mac_address*을 가진 제품(ezTCP)을 찾을 수 없는 경우에는 -1을 반환합니다.**Examples****See also***set_dhcp***Remarks**

[**자동으로 DNS 서버 주소 받기**]가 설정된 경우 제품(ezTCP)은 DHCP 서버로부터 할당 받은 DNS 서버 IP 주소를 사용합니다. 그렇지 않은 경우 사용자가 입력한 DNS 서버 IP 주소를 사용합니다.

5.1.4 set_pppoe

```
int set_pppoe(unsigned char *mac_address, int onoff);
```

Description*set_pppoe*는 [**아이디로 IP 주소 받기(PPPoE)**] 설정 값을 변경합니다.**Parameters***unsigned char *mac_address**EzTCP_Search* 또는 *EzTCP_Read*를 사용하여 검색된 제품(ezTCP)의 MAC주소 입니다.*int onoff*

해당 기능의 선택 여부를 결정하며 아래와 같은 값을 가질 수 있습니다.

0

기능 선택을 해제합니다.

1

기능을 선택합니다.

Return values*set_pppoe*는 설정 값 변경에 성공하면 1을 반환합니다.*mac_address*을 가진 제품(ezTCP)을 찾을 수 없는 경우에는 -1을 반환합니다.**Examples**

See also

set_pppoe_id, set_pppoe_pwd, set_dhcp

Remarks

[아이디로 IP 주소 받기(PPPoE)]와 [자동으로 IP 주소 받기(DHCP)]는 동시에 설정할 수 없습니다. 따라서 *set_pppoe*의 입력값인 *onoff*에 1을 입력하는 경우 [자동으로 IP 주소 받기(DHCP)]는 자동으로 선택이 해제됩니다.

5.1.5 set_ip4_local_address

```
int set_ip4_local_address(unsigned char *mac_address, char *ip4_local_address);
```

Description

*set_ip4_local_address*는 제품(ezTCP)의 IPv4 [제품 IP 주소] 설정 값을 변경합니다.

Parameters

*unsigned char *mac_address*

EzTCP_Search 또는 *EzTCP_Read*를 사용하여 검색된 제품(ezTCP)의 MAC주소 입니다.

*char *ip4_local_address*

IPv4 [제품 IP 주소]를 변경할 IPv4 주소가 저장되어있는 포인터입니다.

Return values

*set_ip4_local_address*의 반환 값은 다음과 같습니다.

1

설정 값 변경에 성공하면 1을 반환합니다.

-1

*mac_address*을 가진 제품(ezTCP)을 찾을 수 없는 경우에는 -1을 반환합니다.

-2

제품(ezTCP)이 사용할 수 없는 IP 주소인 경우에는 -2를 반환합니다.

Examples

```
struct eztcp_info out_eztcp_info;
int ret = get_eztcp_info_by_index(0, &out_eztcp_info);

char *new_ip4_addr = "10.1.0.2";

if(ret == EZTCP_SUCCESS)
{
    int res = set_ip4_local_address(out_eztcp_info.mac_address, new_ip4_addr);
}
```

```

    if(res == 1)
        printf("Success.\r\n");
    else if(res == -1)
        printf("There is no product information of the requested MAC address\r\n");
    else if(res == -2)
        printf("You entered an incorrect IPv4 address.\r\n");
}

```

See also

set_ip4_local_address, set_ip4_subnet_mask, set_ip4_gateway_address, set_ip4_dns_address.

Remarks

set_ip4_local_address 입력 값인 *ip4_local_address*에는 IPv4의 점으로 구분된 십진수 표기법으로 된 IPv4 주소를 입력해야 합니다.

5.1.6 set_ip4_subnet_mask

```
int set_ip4_subnet_mask(unsigned char *mac_address, char *ip4_subnet_mask);
```

Description

*set_ip4_subnet_mask*는 제품(ezTCP)의 IPv4 [서브넷 마스크] 설정 값을 변경합니다.

Parameters

*unsigned char *mac_address*

EzTCP_Search 또는 *EzTCP_Read*를 사용하여 검색된 제품(ezTCP)의 MAC주소 입니다.

*char *ip4_subnet_mask*

IPv4 [서브넷 마스크]를 변경할 서브넷 마스크가 저장되어있는 포인터입니다.

Return values

*set_ip4_subnet_mask*의 반환 값은 다음과 같습니다.

1

설정 값 변경에 성공하면 1을 반환합니다.

-1

*mac_address*을 가진 제품(ezTCP)을 찾을 수 없는 경우에는 -1을 반환합니다.

-2

제품(ezTCP)이 사용할 수 없는 서브넷 마스크인 경우에는 -2를 반환합니다.

Examples

```
struct eztcp_info out_eztcp_info;
```

```

int ret = get_eztcp_info_by_index(0, &out_eztcp_info);

char *new_subnet_mask = "255.255.255.0";

if(ret == EZTCP_SUCCESS)
{
    int res = set_ip4_subnet_mask(out_eztcp_info.mac_address, new_subnet_mask);
    if(res == 1)
        printf("Success.\r\n");
    else if(res == -1)
        printf("There is no product information of the requested MAC address\r\n");
    else if(res == -2)
        printf("You entered an incorrect subnet mask.\r\n");
}

```

See also

set_ip4_local_address, set_ip4_subnet_mask, set_ip4_gateway_address, set_ip4_dns_address.

Remarks

set_ip4_subnet_mask 입력 값인 *ip4_subnet_mask*에는 IPv4의 점으로 구분된 십진수 표기법으로 된 서브넷마스크를 입력해야 합니다.

5.1.7 set_ip4_gateway_address

```
int set_ip4_gateway_address(unsigned char *mac_address, char *ip4_gateway_addr);
```

Description

*set_ip4_gateway_address*는 제품(ezTCP)의 IPv4 [게이트웨이 IP 주소] 설정 값을 변경합니다.

Parameters

*unsigned char *mac_address*

EzTCP_Search 또는 *EzTCP_Read*를 사용하여 검색된 제품(ezTCP)의 MAC주소 입니다.

*char *ip4_gateway_addr*

IPv4 [게이트웨이 IP 주소]를 변경할 게이트웨이 IPv4 주소가 저장되어있는 포인터입니다.

Return values

*set_ip4_gateway_address*는 설정 값 변경에 성공하면 1을 사용합니다.

*mac_address*을 가진 제품(ezTCP)을 찾을 수 없는 경우에는 -1을 반환합니다.

Examples

```
struct eztcp_info out_eztcp_info;
int ret = get_eztcp_info_by_index(0, &out_eztcp_info);

char *new_gateway = "10.1.0.254";

if(ret == EZTCP_SUCCESS)
{
    int res = set_ip4_gateway_address(out_eztcp_info.mac_address, new_gateway);
    if(res == 1)
        printf("Success.\r\n");
    else if(res == -1)
        printf("There is no product information of the requested MAC address\r\n");
}
```

See also

set_ip4_local_address, set_ip4_subnet_mask, set_ip4_gateway_address, set_ip4_dns_address.

Remarks

set_ip4_gateway_address 입력 값인 *ip4_gateway_addr*에는 IPv4의 점으로 구분된 십진수 표기법으로 된 게이트웨이 IPv4 주소를 입력해야 합니다. 사용할 수 없는 게이트웨이 IPv4 주소가 입력되면 [게이트웨이 IP 주소]는 0으로 변경됩니다.

5.1.8 set_ip4_dns_address

```
int set_ip4_dns_address(unsigned char *mac_address, char *ip4_dns_addr);
```

Description

*set_ip4_dns_address*는 제품(ezTCP)의 IPv4 [DNS 서버 IP 주소] 설정 값을 변경합니다.

Parameters

*unsigned char *mac_address*

EzTCP_Search 또는 *EzTCP_Read*를 사용하여 검색된 제품(ezTCP)의 MAC주소 입니다.

*char *ip4_dns_addr*

IPv4 [DNS 서버 IP 주소]를 변경할 DNS 서버 IPv4 주소가 저장되어있는 포인터입니다.

Return values

*set_ip4_dns_address*는 설정 값 변경에 성공하면 1을 사용합니다.

*mac_address*을 가진 제품(ezTCP)을 찾을 수 없는 경우에는 -1을 반환합니다.

Examples

```
struct eztcp_info out_eztcp_info;
int ret = get_eztcp_info_by_index(0, &out_eztcp_info);

char *new_dns = "10.1.0.200";

if(ret == EZTCP_SUCCESS)
{
    int res = set_ip4_dns_address(out_eztcp_info.mac_address, new_dns);
    if(res == 1)
        printf("Success.\r\n");
    else if(res == -1)
        printf("There is no product information of the requested MAC address\r\n");
}
```

See also

set_ip4_local_address, set_ip4_subnet_mask, set_ip4_gateway_address, set_ip4_dns_address.

Remarks

set_ip4_dns_address 입력 값인 *ip4 dns addr*에는 IPv4의 점으로 구분된 십진수 표기법으로 된 DNS 서버 IPv4 주소를 입력해야 합니다. 사용할 수 없는 DNS 서버 IPv4 주소가 입력되면 [DNS 서버 IP 주소]는 0으로 변경됩니다.

5.1.9 set_pppoe_id

```
int set_pppoe_id(unsigned char *mac_address, char *pppoe_id);
```

Description

*set_pppoe_id*는 제품(ezTCP)의 [PPPoE 아이디] 설정 값을 변경합니다.

Parameters

*unsigned char *mac_address*

EzTCP_Search 또는 *EzTCP_Read*를 사용하여 검색된 제품(ezTCP)의 MAC주소 입니다.

*char *pppoe_id*

[PPPoE 아이디]를 변경할 PPPoE 아이디가 저장되어있는 포인터입니다.

Return values

*set_pppoe_id*는 설정 값 변경에 성공하면 1을 반환합니다.

*mac_address*을 가진 제품(ezTCP)을 찾을 수 없는 경우에는 -1을 반환합니다.

Examples

See also

set_pppoe, set_pppoe_pwd

Remarks

[PPPoE 아이디]는 NULL을 제외하고 최대 32바이트입니다.

5.1.10 set_pppoe_pwd

```
int set_pppoe_pwd(unsigned char *mac_address, char *pppoe_pwd);
```

Description

*set_pppoe_pwd*는 제품(ezTCP)의 [PPPoE 비밀번호] 설정 값을 변경합니다.

Parameters

*unsigned char *mac_address*

EzTCP_Search 또는 *EzTCP_Read*를 사용하여 검색된 제품(ezTCP)의 MAC주소 입니다.

*char *pppoe_pwd*

[PPPoE 비밀번호]를 변경할 PPPoE 비밀번호가 저장되어있는 포인터입니다.

Return values

*set_pppoe_pwd*는 설정 값 변경에 성공하면 1을 반환합니다.

*mac_address*을 가진 제품(ezTCP)을 찾을 수 없는 경우에는 -1을 반환합니다.

Examples

See also

get_pppoe, get_pppoe_id

Remarks

[PPPoE 비밀번호]는 NULL을 제외하고 최대 16바이트입니다.

5.1.11 set_ip6

```
int set_ip6(unsigned char *mac_address, int onoff);
```

Description



set_ip6는 [IPv6] 기능의 선택 여부를 변경합니다.

Parameters

*unsigned char *mac_address*

EzTCP_Search 또는 *EzTCP_Read*를 사용하여 검색된 제품(ezTCP)의 MAC주소 입니다.

int onoff

해당 기능의 선택 여부를 결정하며 아래와 같은 값을 가질 수 있습니다.

0

기능 선택을 해제합니다.

1

기능을 선택합니다.

Return values

*set_ip6*의 반환 값은 다음과 같습니다.

1

설정 값 변경에 성공하면 1을 반환합니다.

-1

*mac_address*을 가진 제품(ezTCP)을 찾을 수 없는 경우에는 -1을 반환합니다.

-2

*is_ip6*가 1을 반환하지 않는 제품인 경우 -2를 반환합니다.

-3

*onoff*에 1을 입력하는 경우 [SSL 보안통신] 기능이 선택되어있으면 -3을 반환합니다.

Examples

See also

is_ip6, *set_ip6_gua_type*, *set_ip6_eui_type*, *set_ip6_local_address*, *set_ip6_subnet_prefix_length*, *set_ip6_gateway_address*, *set_ip6_dns_address*.

Remarks

IPv6에 대한 보다 자세한 내용은 당사 홈페이지에서 “IPv6 가이드” 문서를 참조하시기 바랍니다.

5.1.12 set_ip6_gua_type

```
int set_ip6_gua_type(unsigned char *mac_address, int gua_type);
```

Description

*set_ip6_eui_type*는 IPv6 [제품 IP 주소] 설정 방식을 변경합니다.

Parameters

*unsigned char *mac_address*

EzTCP_Search 또는 *EzTCP_Read*를 사용하여 검색된 제품(ezTCP)의 MAC주소 입니다.

int gua_type

IPv6 [제품 IP 주소] 설정 방식을 나타내며 아래와 같은 값을 가질 수 있습니다.

0

[자동으로 IP 주소 받기]

1

[고정된 IP 주소 사용]

Return values

*get_ip6_gua_type*의 반환 값은 다음과 같습니다.

1

설정 값 변경에 성공하면 1을 반환합니다.

-1

*mac_address*을 가진 제품(ezTCP)을 찾을 수 없는 경우에는 -1을 반환합니다.

-2

*is_ip6*가 1을 반환하지 않는 제품인 경우 -2를 반환합니다.

-3

*gua_type*에 유효하지 않은 값을 입력한 경우 -3을 반환합니다.

Examples

See also

is_ip6, *set_ip6_gua_type*, *set_ip6_eui_type*, *set_ip6_local_address*, *set_ip6_subnet_prefix_length*, *set_ip6_gateway_address*, *set_ip6_dns_address*.

Remarks

IPv6에 대한 보다 자세한 내용은 당사 홈페이지에서 “IPv6 가이드” 문서를 참조하시기 바랍니다.

5.1.13 set_ip6_eui_type

```
int set_ip6_eui_type(unsigned char *mac_address, int eui_type);
```

Description

*set_ip6_eui_type*은 Interface ID 생성 방식을 변경합니다. Interface ID 생성 방식은 IPv6 [제품 IP 주소] 설정 방식이 [자동으로 IP 주소 받기]인 경우에 사용됩니다.

Parameters

*unsigned char *mac_address*

EzTCP_Search 또는 *EzTCP_Read*를 사용하여 검색된 제품(ezTCP)의 MAC주소 입니다.

int eui_type

Interface ID 생성 방식을 나타내며 아래와 같은 값을 가질 수 있습니다.

0

[MAC 주소]

1

[Random]

Return values

*get_ip6_eui_type*의 반환 값은 다음과 같습니다.

1

설정 값 변경에 성공하면 1을 반환합니다.

-1

*mac_address*을 가진 제품(ezTCP)을 찾을 수 없는 경우에는 -1을 반환합니다.

-2

*is_ip6*가 1을 반환하지 않는 제품인 경우 -2를 반환합니다.

-3

*eui_type*에 유효하지 않은 값을 입력한 경우 -3을 반환합니다.

Examples**See also**

is_ip6, *set_ip6_gua_type*, *set_ip6_eui_type*, *set_ip6_local_address*, *set_ip6_subnet_prefix_length*,
set_ip6_gateway_address, *set_ip6_dns_address*.

Remarks

IPv6에 대한 보다 자세한 내용은 당사 홈페이지에서 “IPv6 가이드” 문서를 참조하시기 바랍니다.

5.1.14 set_ip6_local_address

```
int set_local_ip6(unsigned char *mac_address, char *ip6_local_address);
```

Description

*set_ip6_local_address*는 제품(ezTCP)의 IPv6 [**제품 IP 주소**]를 변경합니다.

Parameters

*unsigned char *mac_address*

EzTCP_Search 또는 *EzTCP_Read*를 사용하여 검색된 제품(ezTCP)의 MAC주소 입니다.

*char *ip6_local_address*

IPv6 [제품 IP 주소]를 변경할 IPv6 주소가 저장되어있는 포인터입니다.

Return values

*set_ip6_local_address*의 반환 값은 다음과 같습니다.

1

설정 값 변경에 성공하면 1을 반환합니다.

-1

*mac_address*을 가진 제품(ezTCP)을 찾을 수 없는 경우에는 -1을 반환합니다.

-2

*is_ip6*가 1을 반환하지 않는 제품인 경우 -2를 반환합니다.

-3

잘못된 IPv6 주소형식인 경우에는 -3을 반환합니다.

Examples

See also

is_ip6, *set_ip6_gua_type*, *set_ip6_eui_type*, *set_ip6_local_address*, *set_ip6_subnet_prefix_length*, *set_ip6_gateway_address*, *set_ip6_dns_address*.

Remarks

IPv6에 대한 보다 자세한 내용은 당사 홈페이지에서 “IPv6 가이드” 문서를 참조하시기 바랍니다.

5.1.15 set_ip6_subnet_prefix_length

```
int set_ip6_subnet_prefix_length(unsigned char *mac_address,
                                int ip6_subnet_prefix_length);
```

Description

*set_ip6_subnet_prefix_length*는 제품(ezTCP)의 IPv6 서브넷 접두사 길이를 변경합니다.

Parameters

*unsigned char *mac_address*

EzTCP_Search 또는 *EzTCP_Read*를 사용하여 검색된 제품(ezTCP)의 MAC주소 입니다.

int ip6_subnet_prefix_length

변경할 IPv6 서브넷 접두사 길이입니다. 사용 가능한 범위는 0 ~ 128입니다.

Return values

*set_ip6_subnet_prefix_length*의 반환 값은 다음과 같습니다.

1

설정 값 변경에 성공하면 1을 반환합니다.

-1

*mac_address*을 가진 제품(ezTCP)을 찾을 수 없는 경우에는 -1을 반환합니다.

-2

*is_ip6*가 1을 반환하지 않는 제품인 경우 -2를 반환합니다.

-3

입력 가능한 IPv6 서브넷 접두사 길이가 아닌 경우에는 -3을 반환합니다.

Examples

See also

is_ip6, *set_ip6_gua_type*, *set_ip6_eui_type*, *set_ip6_local_address*, *set_ip6_subnet_prefix_length*, *set_ip6_gateway_address*, *set_ip6_dns_address*.

Remarks

IPv6에 대한 보다 자세한 내용은 당사 홈페이지에서 “IPv6 가이드” 문서를 참조하시기 바랍니다.

5.1.16 set_ip6_gateway_address

```
int set_ip6_gateway_address(unsigned char *mac_address, char *ip6_gateway_address);
```

Description

*set_ip6_gateway_address*는 제품(ezTCP)의 IPv6 [게이트웨이 IP 주소]를 변경합니다.

Parameters

*unsigned char *mac_address*

EzTCP_Search 또는 *EzTCP_Read*를 사용하여 검색된 제품(ezTCP)의 MAC주소 입니다.

*char *ip6_gateway_address*

IPv6 [게이트웨이 IP 주소]를 변경할 IPv6 주소가 저장되어있는 포인터입니다.

Return values

*set_ip6_gateway_address*의 반환 값은 다음과 같습니다.

1

설정 값 변경에 성공하면 1을 반환합니다.

-1

*mac_address*을 가진 제품(ezTCP)을 찾을 수 없는 경우에는 -1을 반환합니다.

-2

*is_ip6*가 1을 반환하지 않는 제품인 경우 -2를 반환합니다.

-3

잘못된 IPv6 주소형식인 경우에는 -3를 반환합니다.

Examples

See also

is_ip6, *set_ip6_gua_type*, *set_ip6_eui_type*, *set_ip6_local_address*, *set_ip6_subnet_prefix_length*, *set_ip6_gateway_address*, *set_ip6_dns_address*.

Remarks

IPv6에 대한 보다 자세한 내용은 당사 홈페이지에서 “IPv6 가이드” 문서를 참조하시기 바랍니다.

5.1.17 set_ip6_dns_address

```
int set_ip6_dns_address(unsigned char *mac_address, char *ip6_dns_address);
```

Description

*set_ip6_dns_address*는 제품(ezTCP)의 IPv6 [DNS 서버 IP 주소]를 변경합니다.

Parameters

*unsigned char *mac_address*

EzTCP_Search 또는 *EzTCP_Read*를 사용하여 검색된 제품(ezTCP)의 MAC주소 입니다.

*char *ip6_dns_address*

IPv6 [DNS 서버 IP 주소]를 변경할 IPv6 주소가 저장되어있는 포인터입니다.

Return values

*set_ip6_dns_address*의 반환 값은 다음과 같습니다.

1

설정 값 변경에 성공하면 1을 반환합니다.

-1

*mac_address*을 가진 제품(ezTCP)을 찾을 수 없는 경우에는 -1을 반환합니다.

-2

*is_ip6*가 1을 반환하지 않는 제품인 경우 -2를 반환합니다.



-3

잘못된 IPv6 주소형식인 경우에는 -3를 반환합니다.

Examples

See also

is_ip6, set_ip6_gua_type, set_ip6_eui_type, set_ip6_local_address, set_ip6_subnet_prefix_length, set_ip6_gateway_address, set_ip6_dns_address.

Remarks

IPv6에 대한 보다 자세한 내용은 당사 홈페이지에서 “IPv6 가이드” 문서를 참조하시기 바랍니다.

5.2 시리얼 포트

5.2.1 set_uart_serial_type

```
int set_uart_serial_type(unsigned char *mac_address, int uart_index, int serial_type);
```

Description

set_uart_serial_type은 uart_index로 지정된 시리얼 포트의 [시리얼 종류]를 변경합니다.

Parameters

unsigned char *mac_address

EzTCP_Search 또는 EzTCP_Read를 사용하여 검색된 제품(ezTCP)의 MAC주소 입니다.

int uart_index

제품(ezTCP)의 시리얼 포트 순번으로 0부터 시작합니다.

int serial_type

[시리얼 종류]를 나타내며 아래와 같은 값을 가질 수 있습니다.

0	RS-232
1	RS-485
2	RS-422

Return values

set_uart_serial_type의 반환 값은 다음과 같습니다.

1

설정 값 변경에 성공하면 1을 반환합니다.

-1

*mac_address*을 가진 제품(ezTCP)을 찾을 수 없는 경우에는 -1을 반환합니다.

-2

제품(ezTCP)이 CSE-M53 또는 CSE-M53N인 경우 [시리얼 종류]를 RS-485 또는 RS-422로 변경할 때 제품(ezTCP)에 설정된 [시리얼 통신속도]가 230,400bps보다 큰 경우에는 -2를 반환합니다.

Examples

See also

is_uart_rs232, *is_uart_rs485*, *is_uart_rs422*, *set_uart_serial_type*, *is_uart_ttl*, *set_uart_ttl*, *get_uart_min_baudrate*, *get_uart_max_baudrate*, *set_uart_baudrate*, *set_uart_parity*, *set_uart_databit*, *set_uart_stopbit*, *set_uart_flow_control*, *set_uart_dtrdsr*, *set_uart_tx_delay*.

Remarks

*is_uart_rs232*가 1을 반환하는 경우에만 RS-232를 사용할 수 있습니다.

*is_uart_rs485*가 1을 반환하는 경우에만 RS-485를 사용할 수 있습니다.

*is_uart_rs422*이 1을 반환하는 경우에만 RS-422를 사용할 수 있습니다.

5.2.2 set_uart_ttl

```
int set_uart_ttl(unsigned char *mac_address, int uart_index, int onoff);
```

Description

*set_uart_ttl*은 *uart_index*로 지정된 시리얼 포트의 [TTL] 출력 기능의 선택 여부를 변경합니다.

Parameters

*unsigned char *mac_address*

EzTCP_Search 또는 *EzTCP_Read*를 사용하여 검색된 제품(ezTCP)의 MAC주소 입니다.

int uart_index

제품(ezTCP)의 시리얼 포트 순번으로 0부터 시작합니다.

int onoff

해당 기능의 선택 여부를 결정하며 아래와 같은 값을 가질 수 있습니다.

0

기능 선택을 해제합니다.

1

기능을 선택합니다.

Return values

*set_uart_ttl*의 반환 값은 다음과 같습니다.

1

설정 값 변경에 성공하면 1을 반환합니다.

-1

*mac_address*을 가진 제품(ezTCP)을 찾을 수 없는 경우에는 -1을 반환합니다.

-2

*is_uart_ttl*이 1을 반환하지 않는 제품인 경우 -2를 반환합니다.

Examples**See also**

is_uart_rs232, *is_uart_rs485*, *is_uart_rs422*, *set_uart_serial_type*, *is_uart_ttl*, *set_uart_ttl*, *get_uart_min_baudrate*, *get_uart_max_baudrate*, *set_uart_baudrate*, *set_uart_parity*, *set_uart_databit*, *set_uart_stopbit*, *set_uart_flow_control*, *set_uart_dtrdsr*, *set_uart_tx_delay*.

Remarks

*is_uart_ttl*이 1을 반환하는 경우에만 사용할 수 있습니다.

5.2.3 set_uart_baudrate

```
int set_uart_baudrate(unsigned char *mac_address, int uart_index, int baudrate);
```

Description

*set_uart_baudrate*는 *uart_index*로 지정된 시리얼 포트의 [시리얼 통신속도]를 변경합니다.

Parameters

*unsigned char *mac_address*

EzTCP_Search 또는 *EzTCP_Read*를 사용하여 검색된 제품(ezTCP)의 MAC주소 입니다.

int uart_index

제품(ezTCP)의 시리얼 포트 순번으로 0부터 시작합니다.

int baudrate

변경할 시리얼 통신속도 입니다.

Return values

*set_uart_baudrate*의 반환 값은 다음과 같습니다.

1

설정 값 변경에 성공하면 1을 반환합니다.

-1

*mac_address*을 가진 제품(ezTCP)을 찾을 수 없는 경우에는 -1을 반환합니다.

-2

[시리얼 통신속도]를 변경할 수 없는 경우 -2를 반환합니다.

-3

*baudrate*에 입력된 값의 범위가 *get_uart_min_baudrate*과 *get_uart_max_baudrate*의 반환값을 벗어나는 경우 -3을 반환합니다.

Examples

See also

is_uart_rs232, *is_uart_rs485*, *is_uart_rs422*, *set_uart_serial_type*, *is_uart_ttl*, *set_uart_ttl*, *get_uart_min_baudrate*, *get_uart_max_baudrate*, *set_uart_baudrate*, *set_uart_parity*, *set_uart_databit*, *set_uart_stopbit*, *set_uart_flow_control*, *set_uart_dtrdsr*, *set_uart_tx_delay*.

Remarks

*uart_index*로 지정된 시리얼 포트의 [시리얼 통신속도]는 다른 설정 값에 따라 자동으로 변경될 수 있습니다.

5.2.4 set_uart_parity

```
int set_uart_parity(unsigned char *mac_address, int uart_index, int parity);
```

Description

*set_uart_parity*는 *uart_index*로 지정된 시리얼 포트의 [패리티]를 변경합니다.

Parameters

*unsigned char *mac_address*

EzTCP_Search 또는 *EzTCP_Read*를 사용하여 검색된 제품(ezTCP)의 MAC주소 입니다.

int uart_index

제품(ezTCP)의 시리얼 포트 순번으로 0부터 시작합니다.

int parity

[패리티] 종류를 나타내며 아래와 같은 값을 가질 수 있습니다.

0

NONE

1

EVEN

2

ODD

3
MARK

4
SPACE

Return values

*set_uart_parity*의 반환 값은 다음과 같습니다.

1

설정 값 변경에 성공하면 1을 반환합니다.

-1

*mac_address*을 가진 제품(ezTCP)을 찾을 수 없는 경우에는 -1을 반환합니다.

-2

*is_uart_7_and_8_databit_only*가 1을 반환하는 제품(ezTCP)인 경우 [패리티]를 NONE으로 변경할 때 제품(ezTCP)에 설정된 [데이터 비트]가 7인 경우에는 -2를 반환합니다.

Examples

See also

is_uart_rs232, *is_uart_rs485*, *is_uart_rs422*, *set_uart_serial_type*, *is_uart_ttl*, *set_uart_ttl*,
get_uart_min_baudrate, *get_uart_max_baudrate*, *set_uart_baudrate*, *is_uart_8_databit_only*,
is_uart_7_and_8_databit_only, *set_uart_parity*, *set_uart_databit*, *set_uart_stopbit*, *set_uart_flow_control*,
set_uart_dtrdsr, *set_uart_tx_delay*.

Remarks

*is_uart_7_and_8_databit_only*가 1을 반환하는 제품(ezTCP)은 [데이터 비트]를 7비트로 사용하기 위해선 [패리티]를 NONE 이외의 값으로 설정해야 합니다.

5.2.5 set_uart_databit

```
int set_uart_databit(unsigned char *mac_address, int uart_index, int databit);
```

Description

*get_uart_databit*는 *uart_index*로 지정된 시리얼 포트의 [데이터 비트]를 변경합니다.

Parameters

*unsigned char *mac_address*

EzTCP_Search 또는 *EzTCP_Read*를 사용하여 검색된 제품(ezTCP)의 MAC주소 입니다.

int uart_index

제품(ezTCP)의 시리얼 포트 순번으로 0부터 시작합니다.

int databit

[데이터 비트] 종류를 나타내며 아래와 같은 값을 가질 수 있습니다.

0	5비트
1	6비트
2	7비트
3	8비트

Return values

*set_uart_databit*의 반환 값은 다음과 같습니다.

- 1
설정 값 변경에 성공하면 1을 반환합니다.
- 1
*mac_address*을 가진 제품(ezTCP)을 찾을 수 없는 경우에는 -1을 반환합니다.
- 2
*is_uart_7_and_8_databit_only*가 1을 반환하는 제품(ezTCP)인 경우 [데이터 비트]를 7비트로 변경할 때 제품(ezTCP)에 설정된 [패리티]가 NONE인 경우에는 -2를 반환합니다.

Examples

See also

is_uart_rs232, *is_uart_rs485*, *is_uart_rs422*, *set_uart_serial_type*, *is_uart_ttl*, *set_uart_ttl*, *get_uart_min_baudrate*, *get_uart_max_baudrate*, *set_uart_baudrate*, *is_uart_8_databit_only*, *is_uart_7_and_8_databit_only*, *set_uart_parity*, *set_uart_databit*, *set_uart_stopbit*, *set_uart_flow_control*, *set_uart_dtrdsr*, *set_uart_tx_delay*.

Remarks

*is_uart_7_and_8_databit_only*가 1을 반환하는 제품(ezTCP)은 [데이터 비트]를 7비트로 사용하기 위해선 [패리티]를 NONE 이외의 값으로 설정해야 합니다.

5.2.6 set_uart_stopbit

```
int set_uart_stopbit(unsigned char *mac_address, int uart_index, int stopbit);
```

Description

*set_uart_stopbit*는 *uart_index*로 지정된 시리얼 포트의 [정지 비트]를 변경합니다.

Parameters

*unsigned char *mac_address*

EzTCP_Search 또는 *EzTCP_Read*를 사용하여 검색된 제품(ezTCP)의 MAC주소 입니다.

int uart_index

제품(ezTCP)의 시리얼 포트 순번으로 0부터 시작합니다.

int stopbit

[정지 비트] 종류를 나타내며 아래와 같은 값을 가질 수 있습니다.

0

1비트

1

1.5비트

2

2비트

Return values

*set_uart_stopbit*의 반환 값은 다음과 같습니다.

1

설정 값 변경에 성공하면 1을 반환합니다.

-1

*mac_address*을 가진 제품(ezTCP)을 찾을 수 없는 경우에는 -1을 반환합니다.

-2

*is_uart_one5_stopbit*가 1을 반환하지 않는 제품(ezTCP)인 경우 [정지 비트]를 1.5비트로 변경하려는 경우에는 -2를 반환합니다.

Examples

See also

is_uart_rs232, *is_uart_rs485*, *is_uart_rs422*, *set_uart_serial_type*, *is_uart_ttl*, *set_uart_ttl*,
get_uart_min_baudrate, *get_uart_max_baudrate*, *set_uart_baudrate*, *is_uart_8_databit_only*,
is_uart_7_and_8_databit_only, *set_uart_parity*, *set_uart_databit*, *is_uart_one5_stopbit*, *set_uart_stopbit*,
set_uart_flow_control, *set_uart_dtrdsr*, *set_uart_tx_delay*.

Remarks

5.2.7 set_uart_flow_control

```
int set_uart_flow_control(unsigned char *mac_address, int uart_index, int flow_control);
```

Description



*set_uart_flow_control*는 *uart_index*로 지정된 시리얼 포트의 [흐름 제어]를 변경합니다.

Parameters

*unsigned char *mac_address*

EzTCP_Search 또는 *EzTCP_Read*를 사용하여 검색된 제품(ezTCP)의 MAC주소 입니다.

int uart_index

제품(ezTCP)의 시리얼 포트 순번으로 0부터 시작합니다.

int flow_control

[흐름 제어] 종류를 나타내며 아래와 같은 값을 가질 수 있습니다.

0	NONE
1	RTS/CTS
2	XON/XOFF

Return values

*set_uart_flow_control*의 반환 값은 다음과 같습니다.

1	설정 값 변경에 성공하면 1을 반환합니다.
-1	<i>mac_address</i> 을 가진 제품(ezTCP)을 찾을 수 없는 경우에는 -1을 반환합니다.
-2	제품(ezTCP)이 CSE-M53 또는 CSE-H53인 경우 [흐름 제어]를 XON/XOFF로 변경할 때 제품(ezTCP)에 설정된 [시리얼 통신속도]가 230,400bps보다 큰 경우에는 -2를 반환합니다.
-3	<i>is_uart_xonxoff</i> 가 1을 반환하지 않는 제품(ezTCP)인 경우 [흐름 제어]를 XON/XOFF로 변경하려는 경우에는 -3을 반환합니다.

Examples

See also

is_uart_rs232, *is_uart_rs485*, *is_uart_rs422*, *set_uart_serial_type*, *is_uart_ttl*, *set_uart_ttl*,
get_uart_min_baudrate, *get_uart_max_baudrate*, *set_uart_baudrate*, *is_uart_8_databit_only*,
is_uart_7_and_8_databit_only, *set_uart_parity*, *set_uart_databit*, *is_uart_one5_stopbit*, *set_uart_stopbit*,
is_uart_xonxoff, *set_uart_flow_control*, *set_uart_dtrdsr*, *set_uart_tx_delay*.

Remarks

5.2.8 set_uart_dtrdsr

```
int set_uart_dtrdsr(unsigned char *mac_address, int uart_index, int onoff);
```

Description

*set_uart_dtrdsr*은 *uart_index*로 지정된 시리얼 포트의 [DTR/DSR] 설정 값을 변경합니다.

Parameters

*unsigned char *mac_address*

EzTCP_Search 또는 *EzTCP_Read*를 사용하여 검색된 제품(ezTCP)의 MAC주소 입니다.

int uart_index

제품(ezTCP)의 시리얼 포트 순번으로 0부터 시작합니다.

int onoff

해당 기능의 선택 여부를 결정하며 아래와 같은 값을 가질 수 있습니다.

0

기능 선택을 해제합니다.

1

기능을 선택합니다.

Return values

*set_uart_dtrdsr*의 반환 값은 다음과 같습니다.

1

설정 값 변경에 성공하면 1을 반환합니다.

-1

*mac_address*을 가진 제품(ezTCP)을 찾을 수 없는 경우에는 -1을 반환합니다.

-2

*is_uart_dtrdsr*이 1을 반환하지 않는 제품인 경우 -2를 반환합니다.

Examples

See also

is_uart_rs232, *is_uart_rs485*, *is_uart_rs422*, *set_uart_serial_type*, *is_uart_ttl*, *set_uart_ttl*,
get_uart_min_baudrate, *get_uart_max_baudrate*, *set_uart_baudrate*, *is_uart_8_databit_only*,
is_uart_7_and_8_databit_only, *set_uart_parity*, *set_uart_databit*, *is_uart_one5_stopbit*, *set_uart_stopbit*,
set_uart_flow_control, *is_uart_dtrdsr*, *set_uart_dtrdsr*, *set_uart_tx_delay*.

Remarks

*is_uart_dtrdsr*이 1을 반환하는 경우에만 사용할 수 있습니다.

5.2.9 set_uart_tx_delay

```
int set_uart_tx_delay(unsigned char *mac_address, int uart_index, int tx_delay);
```

Description

*set_uart_tx_delay*는 *uart_index*로 지정된 시리얼 포트의 [데이터 전송 간격]을 변경합니다.

Parameters

*unsigned char *mac_address*

EzTCP_Search 또는 *EzTCP_Read*를 사용하여 검색된 제품(ezTCP)의 MAC주소 입니다.

int uart_index

제품(ezTCP)의 시리얼 포트 순번으로 0부터 시작합니다.

int tx_delay

변경할 데이터 전송 간격입니다.

Return values

*set_uart_tx_delay*의 반환 값은 다음과 같습니다.

1

설정 값 변경에 성공하면 1을 반환합니다.

-1

*mac_address*을 가진 제품(ezTCP)을 찾을 수 없는 경우에는 -1을 반환합니다.

-2

*is_uart_tx_delay*가 1을 반환하지 않는 제품인 경우 -2를 반환합니다.

-3

*tx_delay*에 입력된 값의 범위가 0 ~ 25가 아니면 -3을 반환합니다.

Examples

See also

is_uart_rs232, *is_uart_rs485*, *is_uart_rs422*, *set_uart_serial_type*, *is_uart_ttl*, *set_uart_ttl*,
get_uart_min_baudrate, *get_uart_max_baudrate*, *set_uart_baudrate*, *is_uart_8_databit_only*,
is_uart_7_and_8_databit_only, *set_uart_parity*, *set_uart_databit*, *is_uart_one5_stopbit*, *set_uart_stopbit*,
set_uart_flow_control, *is_uart_dtrdsr*, *set_uart_dtrdsr*, *is_uart_tx_delay*, *set_uart_tx_delay*.

Remarks

*is_uart_tx_delay*가 1을 반환하는 경우에만 사용할 수 있습니다.

5.3 TCP/IP

5.3.1 set_uart_communication_mode

```
int set_uart_communication_mode(unsigned char *mac_address, int uart_index,
                                int communication_mode);
```

Description

*set_uart_communication_mode*는 *uart_index*로 지정된 시리얼 포트의 [통신모드]를 변경합니다.

Parameters

*unsigned char *mac_address*

EzTCP_Search 또는 *EzTCP_Read*를 사용하여 검색된 제품(ezTCP)의 MAC주소 입니다.

int uart_index

제품(ezTCP)의 시리얼 포트 순번으로 0부터 시작합니다.

int communication_mode

[통신모드] 종류를 나타내며 아래와 같은 값을 가질 수 있습니다.

- | | |
|---|-----------------|
| 0 | T2S – TCP 서버 |
| 1 | ATC – AT 명령 |
| 2 | COD – TCP 클라이언트 |
| 3 | U2S – UDP |
| 4 | 시리얼 Modbus/TCP |

Return values

*set_uart_communication_mode*의 반환 값은 다음과 같습니다.

- | | |
|----|---|
| 1 | 설정 값 변경에 성공하면 1을 반환합니다. |
| -1 | <i>mac_address</i> 을 가진 제품(ezTCP)을 찾을 수 없는 경우에는 -1을 반환합니다. |
| -2 | [통신모드]를 T2S – TCP 서버가 아닌 값으로 변경할 때 제품(ezTCP)에 [다중 접속] 기능이 설정된 경우에는 -2를 반환합니다. |

-3

[통신모드]를 U2S – UDP로 변경할 때 제품(ezTCP)에 [SSL 보안통신] 기능이 설정된 경우에는 -3을 반환합니다.

-4

[통신모드]를 T2S – TCP 서버 이외의 값으로 변경할 때 제품(ezTCP)에 [SSH 보안통신] 기능이 설정된 경우에는 -4를 반환합니다.

Examples

See also

is_uart_tcp_server, is_uart_at_command, is_uart_tcp_client, is_uart_udp, is_uart_serial_modbus, set_uart_communication_mode, set_uart_peer_address, set_uart_peer_port, set_uart_local_port, set_uart_cod_tcp_server, set_uart_watermark, set_uart_timeout, set_uart_data_frame_interval, set_uart_separator_length, set_uart_separator_type, set_uart_separator, set_uart_tcp_nodelay, set_uart_rfc2217, set_uart_protocol.

Remarks

is_uart_tcp_server가 1을 반환하는 경우에만 T2S - TCP 서버를 사용할 수 있습니다.

is_uart_at_command가 1을 반환하는 경우에만 ATC – AT 명령을 사용할 수 있습니다.

is_uart_tcp_client가 1을 반환하는 경우에만 T2S - TCP 클라이언트를 사용할 수 있습니다.

is_uart_udp가 1을 반환하는 경우에만 U2S - UDP를 사용할 수 있습니다.

is_uart_serial_modbus가 1을 반환하는 경우에만 시리얼 Modbus/TCP를 사용할 수 있습니다.

5.3.2 set_uart_peer_address

```
int set_uart_peer_address(unsigned char *mac_address, int uart_index,
                          char *peer_address);
```

Description

set_uart_peer_address는 uart_index로 지정된 시리얼 포트의 [통신할 주소]를 변경합니다.

Parameters

*unsigned char *mac_address*

EzTCP_Search 또는 EzTCP_Read를 사용하여 검색된 제품(ezTCP)의 MAC주소 입니다.

int uart_index

제품(ezTCP)의 시리얼 포트 순번으로 0부터 시작합니다.

*char *peer_address*

통신할 주소가 저장되어있는 포인터입니다.

IPv6 기능이 선택되어있으면 peer_address에 IPv6 주소형식을 입력해야 하고 그렇지 않으면 IPv4의 점으로 구분된 십진수 표기법으로 된 IP 주소 또는 호스트 이름을 입력해야 합니다.

Return values

*set_uart_peer_address*의 반환 값은 다음과 같습니다.

1

설정 값 변경에 성공하면 1을 반환합니다.

-1

*mac_address*을 가진 제품(ezTCP)을 찾을 수 없는 경우에는 -1을 반환합니다.

-2

IPv6 기능이 선택되어있는 경우 *peer_address*에 잘못된 IPv6 주소형식을 입력하면 -2를 반환합니다.

Examples

```
char *ipv6_peer_address = "2001:DB8::9";
char *ipv4_peer_address = "192.168.0.5";

struct eztcp_info out_eztcp_info;
int ret = get_eztcp_info_by_index(0, &out_eztcp_info);

if(ret == EZTCP_SUCCESS)
{
    int res;
    unsigned char *mac_address = out_eztcp_info.mac_address;

    if(get_ip6(mac_address) == 1)
        res = set_uart_peer_address(mac_address, 0, ipv6_peer_address);
    else
        res = set_uart_peer_address(mac_address, 0, ipv4_peer_address);

    if(res == 1)
        printf("Success\r\n");
    else if(res == -1)
        printf("There is no product information of the requested MAC address\r\n");
    else if(res == -2)
        printf("You entered an incorrect IPV6 address.\r\n");
}
```

See also

is_uart_tcp_server, is_uart_at_command, is_uart_tcp_client, is_uart_udp, is_uart_serial_modbus, set_uart_communication_mode, set_uart_peer_address, set_uart_peer_port, set_uart_local_port, set_uart_cod_tcp_server, set_uart_watermark, set_uart_timeout, set_uart_data_frame_interval, set_uart_separator_length, set_uart_separator_type, set_uart_separator, set_uart_tcp_nodelay, set_uart_rfc2217, set_uart_protocol.

Remarks

[통신할 주소]는 NULL을 제외하고 최대 64바이트입니다.

5.3.3 set_uart_peer_port

```
int set_uart_peer_port(unsigned char *mac_address, int uart_index, int peer_port);
```

Description

*set_uart_peer_port*는 *uart_index*로 지정된 시리얼 포트의 [통신할 포트]를 변경합니다.

Parameters

*unsigned char *mac_address*

EzTCP_Search 또는 *EzTCP_Read*를 사용하여 검색된 제품(ezTCP)의 MAC주소 입니다.

int uart_index

제품(ezTCP)의 시리얼 포트 순번으로 0부터 시작합니다.

int peer_port

변경할 [통신할 포트]입니다.

Return values

*set_uart_peer_port*의 반환 값은 다음과 같습니다.

1

설정 값 변경에 성공하면 1을 반환합니다.

-1

*mac_address*을 가진 제품(ezTCP)을 찾을 수 없는 경우에는 -1을 반환합니다.

Examples**See also**

is_uart_tcp_server, *is_uart_at_command*, *is_uart_tcp_client*, *is_uart_udp*, *is_uart_serial_modbus*,
set_uart_communication_mode, *set_uart_peer_address*, *set_uart_peer_port*, *set_uart_local_port*,
set_uart_cod_tcp_server, *set_uart_watermark*, *set_uart_timeout*, *set_uart_data_frame_interval*,
set_uart_separator_length, *set_uart_separator_type*, *set_uart_separator*, *set_uart_tcp_nodelay*,
set_uart_rfc2217, *set_uart_protocol*.

Remarks**5.3.4 set_uart_local_port**

```
int set_uart_local_port(unsigned char *mac_address, int uart_index, int local_port);
```

Description

*set_uart_local_port*는 *uart_index*로 지정된 시리얼 포트의 [제품 로컬포트]를 변경합니다.

Parameters

*unsigned char *mac_address*

EzTCP_Search 또는 *EzTCP_Read*를 사용하여 검색된 제품(ezTCP)의 MAC주소 입니다.

int uart_index

제품(ezTCP)의 시리얼 포트 순번으로 0부터 시작합니다.

int local_port

변경할 [제품 로컬포트]입니다.

Return values

*set_uart_local_port*의 반환 값은 다음과 같습니다.

1

설정 값 변경에 성공하면 1을 반환합니다.

-1

*mac_address*을 가진 제품(ezTCP)을 찾을 수 없는 경우에는 -1을 반환합니다.

-2

*local_port*에 입력한 제품 로컬포트 번호가 이미 사용 중인 경우 -2를 반환합니다.

-3

[텔넷] 기능이 선택되어있을 때 *local_port*에 23번을 입력하는 경우 -3을 반환합니다.

-4

I/O 제품의 Modbus/TCP 접속 방법이 [수동 접속]이고 *local_port*에 입력한 제품 로컬포트 번호가 Modbus/TCP [제품 로컬포트]로 이미 사용 중인 경우 -4를 반환합니다.

-5

I/O 제품 [웹(HTTP)] 제어방식 기능이 선택되어있을 때 *local_port*에 입력한 제품 로컬포트 번호가 I/O 제품 [웹(HTTP) 포트] 번호로 이미 사용 중인 경우 -5를 반환합니다.

-6

*local_port*에 입력한 제품 로컬포트 번호가 50,005번이면 -6을 반환합니다.

-7

*local_port*에 입력한 제품 로컬포트 번호가 50,006번이면 -7을 반환합니다.

-8

*local_port*에 입력한 제품 로컬포트 번호가 50,007번이면 -8을 반환합니다.

Examples

See also

is_uart_tcp_server, *is_uart_at_command*, *is_uart_tcp_client*, *is_uart_udp*, *is_uart_serial_modbus*,
set_uart_communication_mode, *set_uart_peer_address*, *set_uart_peer_port*, *set_uart_local_port*,

set_uart_cod_tcp_server, set_uart_watermark, set_uart_timeout, set_uart_data_frame_interval,
set_uart_separator_length, set_uart_separator_type, set_uart_separator, set_uart_tcp_nodelay,
set_uart_rfc2217, set_uart_protocol.

Rermarks

50,005번, 50,006번 그리고 50,007번은 제품(ezTCP) 검색 등을 위해서 라이브러리 내부에서 사용하고 있는 포트번호 이므로 제품(ezTCP)의 설정 값으로 사용할 수 없습니다.

5.3.5 set_uart_cod_tcp_server

```
int set_uart_cod_tcp_server(unsigned char *mac_address, int uart_index, int onoff);
```

Description

*set_uart_cod_tcp_server*은 *uart_index*로 지정된 시리얼 포트의 [TCP 서버] 기능의 선택 여부를 변경합니다.

Parameters

*unsigned char *mac_address*

EzTCP_Search 또는 *EzTCP_Read*를 사용하여 검색된 제품(ezTCP)의 MAC주소 입니다.

int uart_index

제품(ezTCP)의 시리얼 포트 순번으로 0부터 시작합니다.

int onoff

해당 기능의 선택 여부를 결정하며 아래와 같은 값을 가질 수 있습니다.

0

기능 선택을 해제합니다.

1

기능을 선택합니다.

Return values

*set_uart_cod_tcp_server*의 반환 값은 다음과 같습니다.

1

설정 값 변경에 성공하면 1을 반환합니다.

-1

*mac_address*을 가진 제품(ezTCP)을 찾을 수 없는 경우에는 -1을 반환합니다.

Examples

See also

is_uart_tcp_server, is_uart_at_command, is_uart_tcp_client, is_uart_udp, is_uart_serial_modbus,

set_uart_communication_mode, set_uart_peer_address, set_uart_peer_port, set_uart_local_port,
 set_uart_cod_tcp_server, set_uart_watermark, set_uart_timeout, set_uart_data_frame_interval,
 set_uart_separator_length, set_uart_separator_type, set_uart_separator, set_uart_tcp_nodelay,
 set_uart_rfc2217, set_uart_protocol.

Remarks

*get_uart_communication_mode*이 2(COD - TCP 클라이언트)를 반환하는 경우에만 사용할 수 있습니다.

5.3.6 set_uart_watermark

```
int set_uart_watermark(unsigned char *mac_address, int uart_index, int watermark);
```

Description

*set_uart_watermark*는 *uart_index*로 지정된 시리얼 포트의 [접속전 데이터 크기] 또는 [패킷 블록 설정(바이트)]를 변경합니다.

Parameters

*unsigned char *mac_address*

EzTCP_Search 또는 *EzTCP_Read*를 사용하여 검색된 제품(ezTCP)의 MAC주소 입니다.

int uart_index

제품(ezTCP)의 시리얼 포트 순번으로 0부터 시작합니다.

int watermark

변경할 접속전 데이터 크기 또는 패킷 블록 설정(바이트) 값입니다.

Return values

*set_uart_watermark*의 반환 값은 다음과 같습니다.

1

설정 값 변경에 성공하면 1을 반환합니다.

-1

*mac_address*을 가진 제품(ezTCP)을 찾을 수 없는 경우에는 -1을 반환합니다.

Examples

See also

is_uart_tcp_server, is_uart_at_command, is_uart_tcp_client, is_uart_udp, is_uart_serial_modbus,
 set_uart_communication_mode, set_uart_peer_address, set_uart_peer_port, set_uart_local_port,
 set_uart_cod_tcp_server, set_uart_watermark, set_uart_timeout, set_uart_data_frame_interval,
 set_uart_separator_length, set_uart_separator_type, set_uart_separator, set_uart_tcp_nodelay,
 set_uart_rfc2217, set_uart_protocol.

Remarks

[통신모드]가 0 (T2S – TCP 서버), 1 (ATC – AT 명령), 2 (COD – TCP 클라이언트)인 경우 [접속전 데이터 크기]로 사용됩니다.

[통신모드]가 3 (U2S – UDP)인 경우 [패킷 블록 설정 (바이트)]로 사용됩니다.

[접속전 데이터 크기] 또는 [패킷 블록 설정(바이트)]는 다른 설정 값에 따라 자동으로 변경될 수 있습니다.

5.3.7 set_uart_timeout

```
int set_uart_timeout(unsigned char *mac_address, int uart_index, int timeout);
```

Description

*set_uart_timeout*은 *uart_index*로 지정된 시리얼 포트의 [접속종료 대기시간(초)]를 변경합니다.

Parameters

*unsigned char *mac_address*

EzTCP_Search 또는 *EzTCP_Read*를 사용하여 검색된 제품(ezTCP)의 MAC주소 입니다.

int uart_index

제품(ezTCP)의 시리얼 포트 순번으로 0부터 시작합니다.

int timeout

변경할 접속종료 대기시간(초)입니다.

Return values

*set_uart_timeout*의 반환 값은 다음과 같습니다.

1

설정 값 변경에 성공하면 1을 반환합니다.

-1

*mac_address*을 가진 제품(ezTCP)을 찾을 수 없는 경우에는 -1을 반환합니다.

Examples**See also**

is_uart_tcp_server, *is_uart_at_command*, *is_uart_tcp_client*, *is_uart_udp*, *is_uart_serial_modbus*, *set_uart_communication_mode*, *set_uart_peer_address*, *set_uart_peer_port*, *set_uart_local_port*, *set_uart_cod_tcp_server*, *set_uart_watermark*, *set_uart_timeout*, *set_uart_data_frame_interval*, *set_uart_separator_length*, *set_uart_separator_type*, *set_uart_separator*, *set_uart_tcp_nodelay*, *set_uart_rfc2217*, *set_uart_protocol*.

Remarks

[통신모드]가 0 (T2S – TCP 서버), 1 (ATC – AT 명령), 2 (COD – TCP 클라이언트)인 경우에만 사용할 수 있습니다.

[접속종료 대기시간(초)]는 다른 설정 값에 따라 자동으로 변경될 수 있습니다.

5.3.8 set_uart_data_frame_interval

```
int set_uart_data_frame_interval(unsigned char *mac_address, int uart_index,
                                int data_frame_interval);
```

Description

*set_uart_data_frame_interval*은 *uart_index*로 지정된 시리얼 포트의 [데이터 프레임 간격(10ms)]를 변경합니다.

Parameters

*unsigned char *mac_address*

EzTCP_Search 또는 *EzTCP_Read*를 사용하여 검색된 제품(ezTCP)의 MAC주소 입니다.

int uart_index

제품(ezTCP)의 시리얼 포트 순번으로 0부터 시작합니다.

int data_frame_interval

변경할 데이터 프레임 간격(10ms)입니다.

Return values

*set_uart_data_frame_interval*의 반환 값은 다음과 같습니다.

1

설정 값 변경에 성공하면 1을 반환합니다.

-1

*mac_address*을 가진 제품(ezTCP)을 찾을 수 없는 경우에는 -1을 반환합니다.

-2

*is_uart_data_frame_interval*이 1을 반환하지 않는 제품인 경우 -2를 반환합니다.

Examples

See also

is_uart_tcp_server, *is_uart_at_command*, *is_uart_tcp_client*, *is_uart_udp*, *is_uart_serial_modbus*, *set_uart_communication_mode*, *set_uart_peer_address*, *set_uart_peer_port*, *set_uart_local_port*, *set_uart_cod_tcp_server*, *set_uart_watermark*, *set_uart_timeout*, *is_uart_data_frame_interval*, *set_uart_data_frame_interval*, *set_uart_separator_length*, *set_uart_separator_type*, *set_uart_separator*, *set_uart_tcp_nodelay*, *set_uart_rfc2217*, *set_uart_protocol*.

Remarks



[통신모드]가 4 (시리얼 Modbus/TCP)인 경우에는 사용할 수 없습니다.

5.3.9 set_uart_separator_length

```
int set_uart_separator_length(unsigned char *mac_address, int uart_index,
                             int separator_length);
```

Description

*set_uart_separator_length*는 *uart_index*로 지정된 시리얼 포트의 [구분자 길이]를 변경합니다.

Parameters

*unsigned char *mac_address*

EzTCP_Search 또는 *EzTCP_Read*를 사용하여 검색된 제품(ezTCP)의 MAC주소 입니다.

int uart_index

제품(ezTCP)의 시리얼 포트 순번으로 0부터 시작합니다.

int separator_length

변경할 구분자 길이로 0 ~ 4입니다.

Return values

*set_uart_separator_length*의 반환 값은 다음과 같습니다.

1

설정 값 변경에 성공하면 1을 반환합니다.

-1

*mac_address*을 가진 제품(ezTCP)을 찾을 수 없는 경우에는 -1을 반환합니다.

-2

*is_uart_separator*가 1을 반환하지 않는 제품인 경우 -2를 반환합니다.

Examples

See also

is_uart_tcp_server, *is_uart_at_command*, *is_uart_tcp_client*, *is_uart_udp*, *is_uart_serial_modbus*,
set_uart_communication_mode, *set_uart_peer_address*, *set_uart_peer_port*, *set_uart_local_port*,
set_uart_cod_tcp_server, *set_uart_watermark*, *set_uart_timeout*, *set_uart_data_frame_interval*,
set_uart_separator_length, *is_uart_separator*, *set_uart_separator_type*, *set_uart_separator*,
set_uart_tcp_nodelay, *set_uart_rfc2217*, *set_uart_protocol*.

Remarks

5.3.10 set_uart_separator_type

```
int set_uart_separator_type(unsigned char *mac_address, int uart_index,
                           int separator_type);
```

Description

*set_uart_separator_type*은 *uart_index*로 지정된 시리얼 포트의 [구분자 동작방식]을 변경합니다.

Parameters

*unsigned char *mac_address*

EzTCP_Search 또는 *EzTCP_Read*를 사용하여 검색된 제품(ezTCP)의 MAC주소 입니다.

int uart_index

제품(ezTCP)의 시리얼 포트 순번으로 0부터 시작합니다.

int separator_type

[구분자 동작방식] 종류를 나타내며 아래와 같은 값을 가질 수 있습니다.

0

구분자까지 전송

1

구분자+1바이트 전송

2

구분자+2바이트 전송

Return values

*set_uart_separator_type*의 반환 값은 다음과 같습니다.

1

설정 값 변경에 성공하면 1을 반환합니다.

-1

*mac_address*을 가진 제품(ezTCP)을 찾을 수 없는 경우에는 -1을 반환합니다.

-2

*is_uart_separator*가 1을 반환하지 않는 제품인 경우 -2를 반환합니다.

-3

*separator_type*에 유효하지 않은 값을 입력한 경우 -3을 반환합니다.

Examples

See also

is_uart_tcp_server, *is_uart_at_command*, *is_uart_tcp_client*, *is_uart_udp*, *is_uart_serial_modbus*,
set_uart_communication_mode, *set_uart_peer_address*, *set_uart_peer_port*, *set_uart_local_port*,

set_uart_cod_tcp_server, set_uart_watermark, set_uart_timeout, set_uart_data_frame_interval,
 set_uart_separator_length, is_uart_separator, set_uart_separator_type, set_uart_separator,
 set_uart_tcp_nodelay, set_uart_rfc2217, set_uart_protocol.

Remarks

5.3.11 set_uart_separator

```
int set_uart_separator(unsigned char *mac_address, int uart_index, int separator_index,
                      char *separator);
```

Description

*set_uart_separator*는 *uart_index*로 지정된 시리얼 포트의 [구분자(16진수)]를 변경합니다.

Parameters

*unsigned char *mac_address*

EzTCP_Search 또는 *EzTCP_Read*를 사용하여 검색된 제품(ezTCP)의 MAC주소 입니다.

int uart_index

제품(ezTCP)의 시리얼 포트 순번으로 0부터 시작합니다.

int separator_index

구분자의 순번으로 0부터 시작합니다.

*char *separator*

*separator_index*로 지정된 구분자를 변경할 16진수 문자열입니다.

Return values

*set_uart_separator*의 반환 값은 다음과 같습니다.

1

설정 값 변경에 성공하면 1을 반환합니다.

-1

*mac_address*을 가진 제품(ezTCP)을 찾을 수 없는 경우에는 -1을 반환합니다.

-2

*is_uart_separator*가 1을 반환하지 않는 제품인 경우 -2를 반환합니다.

-3

*separator_index*에 유효하지 않은 값을 입력한 경우 -3을 반환합니다.

Examples

```
char *separator = "3A";
struct eztcp_info out_eztcp_info;
```

```

int ret = get_eztcp_info_by_index(0, &out_eztcp_info);

if(ret == EZTCP_SUCCESS)
{
    int res;
    unsigned char *mac_address = out_eztcp_info.mac_address;

    if(is_uart_separaot(mac_address) == 1)
    {
        res = set_uart_separator(mac_address, 0, 0, separator);

        if(res == 1)
            printf("Success\r\n");
        else if(res == -1)
            printf("There is no product information of the requested MAC
address\r\n");
        else if(res == -2)
            printf("This product can't use separator function.\r\n");
        else if(res == -3)
            printf("You entered an invalid separator.\r\n");
    }
}
}

```

See also

is_uart_tcp_server, is_uart_at_command, is_uart_tcp_client, is_uart_udp, is_uart_serial_modbus, set_uart_communication_mode, set_uart_peer_address, set_uart_peer_port, set_uart_local_port, set_uart_cod_tcp_server, set_uart_watermark, set_uart_timeout, set_uart_data_frame_interval, set_uart_separator_length, is_uart_separator, set_uart_separator_type, set_uart_separator, set_uart_tcp_nodelay, set_uart_rfc2217, set_uart_protocol.

Remarks

*separator*는 16진수 문자열로 최대 2자입니다.

5.3.12 set_uart_tcp_nodelay

```
int set_uart_tcp_nodelay(unsigned char *mac_address, int uart_index, int onoff);
```

Description

*set_uart_tcp_nodelay*는 *uart_index*로 지정된 시리얼 포트의 [전송지연 기능 사용안함] 기능의 선택 여부를 변경합니다.

Parameters

*unsigned char *mac_address*

EzTCP_Search 또는 *EzTCP_Read*를 사용하여 검색된 제품(ezTCP)의 MAC주소 입니다.

int uart_index

제품(ezTCP)의 시리얼 포트 순번으로 0부터 시작합니다.

int onoff

해당 기능의 선택 여부를 결정하며 아래와 같은 값을 가질 수 있습니다.

0

기능 선택을 해제합니다.

1

기능을 선택합니다.

Return values

*set_uart_tcp_nodelay*의 반환 값은 다음과 같습니다.

1

설정 값 변경에 성공하면 1을 반환합니다.

-1

*mac_address*을 가진 제품(ezTCP)을 찾을 수 없는 경우에는 -1을 반환합니다.

-2

*is_uart_tcp_nodelay*가 1을 반환하지 않는 경우 -2를 반환합니다.

Examples

See also

is_uart_tcp_server, *is_uart_at_command*, *is_uart_tcp_client*, *is_uart_udp*, *is_uart_serial_modbus*, *set_uart_communication_mode*, *set_uart_peer_address*, *set_uart_peer_port*, *set_uart_local_port*, *set_uart_cod_tcp_server*, *set_uart_watermark*, *set_uart_timeout*, *set_uart_data_frame_interval*, *set_uart_separator_length*, *set_uart_separator_type*, *set_uart_separator*, *is_uart_tcp_nodelay*, *set_uart_tcp_nodelay*, *set_uart_rfc2217*, *set_uart_protocol*.

Remarks

5.3.13 set_uart_rfc2217

```
int set_uart_rfc2217(unsigned char *mac_address, int uart_index, int onoff);
```

Description

*set_uart_rfc2217*은 *uart_index*로 지정된 시리얼 포트의 [시리얼 포트 설정/상태 전송 (RFC2217)] 기능의 선택 여부를 변경합니다.

Parameters

*unsigned char *mac_address*

EzTCP_Search 또는 *EzTCP_Read*를 사용하여 검색된 제품(ezTCP)의 MAC주소 입니다.

int uart_index

제품(ezTCP)의 시리얼 포트 순번으로 0부터 시작합니다.

int onoff

해당 기능의 선택 여부를 결정하며 아래와 같은 값을 가질 수 있습니다.

0

기능 선택을 해제합니다.

1

기능을 선택합니다.

Return values

*set_uart_rfc2217*의 반환 값은 다음과 같습니다.

1

설정 값 변경에 성공하면 1을 반환합니다.

-1

*mac_address*을 가진 제품(ezTCP)을 찾을 수 없는 경우에는 -1을 반환합니다.

-2

*is_uart_rfc2217*이 1을 반환하지 않는 경우 -2를 반환합니다.

-3

[다중 접속]이 선택되어 있을 때 *onoff*에 1을 입력하면 -3을 반환합니다.

-4

[SSL 보안통신] 또는 [SSH 보안통신]이 선택되어 있을 때 *onoff*에 1을 입력하면 -4를 반환합니다.

Examples

See also

is_uart_tcp_server, *is_uart_at_command*, *is_uart_tcp_client*, *is_uart_udp*, *is_uart_serial_modbus*, *set_uart_communication_mode*, *set_uart_peer_address*, *set_uart_peer_port*, *set_uart_local_port*, *set_uart_cod_tcp_server*, *set_uart_watermark*, *set_uart_timeout*, *set_uart_data_frame_interval*, *set_uart_separator_length*, *set_uart_separator_type*, *set_uart_separator*, *set_uart_tcp_nodelay*, *is_uart_rfc2217*, *set_uart_rfc2217*, *set_uart_protocol*.

Remarks

5.3.14 set_uart_protocol

```
int set_uart_protocol(unsigned char *mac_address, int uart_index, int uart_protocol);
```

Description

*set_uart_protocol*은 *uart_index*로 지정된 시리얼 포트가 사용 중인 [프로토콜]을 변경합니다.

Parameters

*unsigned char *mac_address*

EzTCP_Search 또는 *EzTCP_Read*를 사용하여 검색된 제품(ezTCP)의 MAC주소 입니다.

int uart_index

제품(ezTCP)의 시리얼 포트 순번으로 0부터 시작합니다.

int uart_protocol

[프로토콜] 종류를 나타내며 아래와 같은 값을 가질 수 있습니다.

0

TCP

1

TCP + TELNET

2

TCP + SSL

Return values

*set_uart_protocol*의 반환 값은 다음과 같습니다.

1

설정 값 변경에 성공하면 1을 반환합니다.

-1

*mac_address*을 가진 제품(ezTCP)을 찾을 수 없는 경우에는 -1을 반환합니다.

-2

*is_uart_protocol*이 1을 반환하지 않는 경우 -2를 반환합니다.

Examples

See also

is_uart_tcp_server, *is_uart_at_command*, *is_uart_tcp_client*, *is_uart_udp*, *is_uart_serial_modbus*, *set_uart_communication_mode*, *set_uart_peer_address*, *set_uart_peer_port*, *set_uart_local_port*, *set_uart_cod_tcp_server*, *set_uart_watermark*, *set_uart_timeout*, *set_uart_data_frame_interval*, *set_uart_separator_length*, *set_uart_separator_type*, *set_uart_separator*, *set_uart_tcp_nodelay*, *set_uart_rfc2217*, *is_uart_protocol*, *set_uart_protocol*.

Remarks

현재 CSE-T16, CSE-T32, CSE-T48만 사용 가능하며 추후 지원가능 모델은 추가 될 수 있습니다. 보다 자세한 내용은 해당 제품 사용설명서를 참조하시기 바랍니다.

5.4 CSC-HR2

5.4.1 set_csc_hr2_communication_mode

```
int set_csc_hr2_communication_mode(unsigned char *mac_address,
                                   int communication_mode);
```

Description

*set_csc_hr2_communication_mode*는 CSC-HR2의 [통신모드]를 변경합니다.

Parameters

*unsigned char *mac_address*

EzTCP_Search 또는 *EzTCP_Read*를 사용하여 검색된 제품(ezTCP)의 MAC주소 입니다.

int communication_mode

[통신모드] 종류를 나타내며 아래와 같은 값을 가질 수 있습니다.

0

자동

1

EZU-100 펌웨어 변경

Return values

*set_csc_hr2_communication_mode*의 반환 값은 다음과 같습니다.

1

설정 값 변경에 성공하면 1을 반환합니다.

-1

*mac_address*을 가진 제품(ezTCP)을 찾을 수 없는 경우에는 -1을 반환합니다.

-2

*is_csc_hr2*가 1을 반환하지 않는 경우 -2를 반환합니다.

Examples

See also

is_csc_hr2, *set_csc_hr2_communication_mode*, *set_csc_hr2_id*, *set_csc_hr2_network_timeout*,
set_csc_hr2_network_threshold, *set_csc_hr2_server_timeout*, *set_csc_hr2_server_threshold*,

```
set_csc_hr2_checkport,          set_csc_hr2_first_server_address,    set_csc_hr2_first_server_port,
set_csc_hr2_second_server_address, set_csc_hr2_second_server_port.
```

Remarks

보다 자세한 내용은 CSC-HR2 제품 사용설명서를 참조하시기 바랍니다.

5.4.2 set_csc_hr2_id

```
int set_csc_hr2_id(unsigned char *mac_address, char *csc_hr2_id);
```

Description

*set_csc_hr2_id*는 CSC-HR2의 [CSC-HR2 아이디]를 변경합니다.

Parameters

*unsigned char *mac_address*

EzTCP_Search 또는 *EzTCP_Read*를 사용하여 검색된 제품(ezTCP)의 MAC주소 입니다.

*char *csc_hr2_id*

변경할 CSC-HR2 아이디입니다.

Return values

*set_csc_hr2_id*의 반환 값은 다음과 같습니다.

1

설정 값 변경에 성공하면 1을 반환합니다.

-1

*mac_address*을 가진 제품(ezTCP)을 찾을 수 없는 경우에는 -1을 반환합니다.

-2

*is_csc_hr2*가 1을 반환하지 않는 경우 -2를 반환합니다.

Examples

See also

```
is_csc_hr2,      set_csc_hr2_communication_mode,    set_csc_hr2_id,      set_csc_hr2_network_timeout,
set_csc_hr2_network_threshold,    set_csc_hr2_server_timeout,    set_csc_hr2_server_threshold,
set_csc_hr2_checkport,    set_csc_hr2_first_server_address,    set_csc_hr2_first_server_port,
set_csc_hr2_second_server_address, set_csc_hr2_second_server_port.
```

Remarks

[CSC-HR2 아이디]는 NULL을 제외하고 최대 16바이트입니다. 보다 자세한 내용은 CSC-HR2 제품 사용설명서를 참조하시기 바랍니다.

5.4.3 set_csc_hr2_network_timeout

```
int set_csc_hr2_network_timeout(unsigned char *mac_address, int timeout);
```

Description

*set_csc_hr2_network_timeout*은 CSC-HR2의 [절체 타임아웃]을 변경합니다.

Parameters

*unsigned char *mac_address*

EzTCP_Search 또는 *EzTCP_Read*를 사용하여 검색된 제품(ezTCP)의 MAC주소 입니다.

int timeout

변경할 절체 타임아웃입니다.

Return values

*set_csc_hr2_network_timeout*의 반환 값은 다음과 같습니다.

1

설정 값 변경에 성공하면 1을 반환합니다.

-1

*mac_address*을 가진 제품(ezTCP)을 찾을 수 없는 경우에는 -1을 반환합니다.

-2

*is_csc_hr2*가 1을 반환하지 않는 경우 -2를 반환합니다.

Examples

See also

is_csc_hr2, *set_csc_hr2_communication_mode*, *set_csc_hr2_id*, *set_csc_hr2_network_timeout*,
set_csc_hr2_network_threshold, *set_csc_hr2_server_timeout*, *set_csc_hr2_server_threshold*,
set_csc_hr2_checkport, *set_csc_hr2_first_server_address*, *set_csc_hr2_first_server_port*,
set_csc_hr2_second_server_address, *set_csc_hr2_second_server_port*.

Remarks

보다 자세한 내용은 CSC-HR2 제품 사용설명서를 참조하시기 바랍니다.

5.4.4 set_csc_hr2_network_threshold

```
int set_csc_hr2_network_threshold(unsigned char *mac_address, int threshold);
```

Description

*set_csc_hr2_network_threshold*는 CSC-HR2의 [절체 바이트 수]를 변경합니다.

Parameters*unsigned char *mac_address**EzTCP_Search* 또는 *EzTCP_Read*를 사용하여 검색된 제품(ezTCP)의 MAC주소 입니다.*int threshold*

변경할 절체 바이트 수입니다.

Return values*set_csc_hr2_network_threshold*의 반환 값은 다음과 같습니다.

1

설정 값 변경에 성공하면 1을 반환합니다.

-1

*mac_address*을 가진 제품(ezTCP)을 찾을 수 없는 경우에는 -1을 반환합니다.

-2

*is_csc_hr2*가 1을 반환하지 않는 경우 -2를 반환합니다.**Examples****See also**

is_csc_hr2, set_csc_hr2_communication_mode, set_csc_hr2_id, set_csc_hr2_network_timeout,
 set_csc_hr2_network_threshold, set_csc_hr2_server_timeout, set_csc_hr2_server_threshold,
 set_csc_hr2_checkport, set_csc_hr2_first_server_address, set_csc_hr2_first_server_port,
 set_csc_hr2_second_server_address, set_csc_hr2_second_server_port.

Remarks

보다 자세한 내용은 CSC-HR2 제품 사용설명서를 참조하시기 바랍니다.

5.4.5 set_csc_hr2_server_timeout

```
int set_csc_hr2_server_timeout(unsigned char *mac_address, int timeout);
```

Description*set_csc_hr2_server_timeout*은 CSC-HR2의 [서버변경 타임아웃]을 변경합니다.**Parameters***unsigned char *mac_address**EzTCP_Search* 또는 *EzTCP_Read*를 사용하여 검색된 제품(ezTCP)의 MAC주소 입니다.*int timeout*

변경할 서버변경 타임아웃입니다.

Return values

*set_csc_hr2_server_timeout*의 반환 값은 다음과 같습니다.

1

설정 값 변경에 성공하면 1을 반환합니다.

-1

*mac_address*을 가진 제품(ezTCP)을 찾을 수 없는 경우에는 -1을 반환합니다.

-2

*is_csc_hr2*가 1을 반환하지 않는 경우 -2를 반환합니다.

Examples**See also**

is_csc_hr2, *set_csc_hr2_communication_mode*, *set_csc_hr2_id*, *set_csc_hr2_network_timeout*,
set_csc_hr2_network_threshold, *set_csc_hr2_server_timeout*, *set_csc_hr2_server_threshold*,
set_csc_hr2_checkport, *set_csc_hr2_first_server_address*, *set_csc_hr2_first_server_port*,
set_csc_hr2_second_server_address, *set_csc_hr2_second_server_port*.

Remarks

보다 자세한 내용은 CSC-HR2 제품 사용설명서를 참조하시기 바랍니다.

5.4.6 set_csc_hr2_server_threshold

```
int set_csc_hr2_server_threshold(unsigned char *mac_address, int threshold);
```

Description

*set_csc_hr2_server_threshold*는 CSC-HR2의 [서버변경 바이트 수]를 변경합니다.

Parameters

*unsigned char *mac_address*

EzTCP_Search 또는 *EzTCP_Read*를 사용하여 검색된 제품(ezTCP)의 MAC주소 입니다.

int threshold

변경할 서버변경 바이트 수입니다.

Return values

*set_csc_hr2_server_threshold*의 반환 값은 다음과 같습니다.

1

설정 값 변경에 성공하면 1을 반환합니다.

-1

*mac_address*을 가진 제품(ezTCP)을 찾을 수 없는 경우에는 -1을 반환합니다.

-2

*is_csc_hr2*가 1을 반환하지 않는 경우 -2를 반환합니다.

Examples

See also

is_csc_hr2, *set_csc_hr2_communication_mode*, *set_csc_hr2_id*, *set_csc_hr2_network_timeout*,
set_csc_hr2_network_threshold, *set_csc_hr2_server_timeout*, *set_csc_hr2_server_threshold*,
set_csc_hr2_checkport, *set_csc_hr2_first_server_address*, *set_csc_hr2_first_server_port*,
set_csc_hr2_second_server_address, *set_csc_hr2_second_server_port*.

Remarks

보다 자세한 내용은 CSC-HR2 제품 사용설명서를 참조하시기 바랍니다.

5.4.7 set_csc_hr2_checkport

```
int set_csc_hr2_checkport(unsigned char *mac_address, int check_port);
```

Description

*set_csc_hr2_check_port*는 CSC-HR2의 [통신품질 점검포트]를 변경합니다.

Parameters

*unsigned char *mac_address*

EzTCP_Search 또는 *EzTCP_Read*를 사용하여 검색된 제품(ezTCP)의 MAC주소 입니다.

int check_port

변경할 통신품질 점검포트 번호입니다.

Return values

*set_csc_hr2_checkport*의 반환 값은 다음과 같습니다.

1

설정 값 변경에 성공하면 1을 반환합니다.

-1

*mac_address*을 가진 제품(ezTCP)을 찾을 수 없는 경우에는 -1을 반환합니다.

-2

*is_csc_hr2*가 1을 반환하지 않는 경우 -2를 반환합니다.

Examples

See also

```
is_csc_hr2,      set_csc_hr2_communication_mode,  set_csc_hr2_id,      set_csc_hr2_network_timeout,
set_csc_hr2_network_threshold,  set_csc_hr2_server_timeout,  set_csc_hr2_server_threshold,
set_csc_hr2_checkport,      set_csc_hr2_first_server_address,  set_csc_hr2_first_server_port,
set_csc_hr2_second_server_address, set_csc_hr2_second_server_port.
```

Remarks

보다 자세한 내용은 CSC-HR2 제품 사용설명서를 참조하시기 바랍니다.

5.4.8 set_csc_hr2_first_server_address

```
int set_csc_hr2_first_server_address(unsigned char *mac_address, char *server_address);
```

Description

*set_csc_hr2_first_server_address*는 CSC-HR2의 [첫 번째 서버 주소]를 변경합니다.

Parameters

*unsigned char *mac_address*

EzTCP_Search 또는 *EzTCP_Read*를 사용하여 검색된 제품(ezTCP)의 MAC주소 입니다.

*char *server_address*

첫 번째 서버 주소가 저장되어있는 포인터입니다.

Return values

*set_csc_hr2_first_server_address*의 반환 값은 다음과 같습니다.

1

설정 값 변경에 성공하면 1을 반환합니다.

-1

*mac_address*을 가진 제품(ezTCP)을 찾을 수 없는 경우에는 -1을 반환합니다.

-2

*is_csc_hr2*가 1을 반환하지 않는 경우 -2를 반환합니다.

Examples

See also

```
is_csc_hr2,      set_csc_hr2_communication_mode,  set_csc_hr2_id,      set_csc_hr2_network_timeout,
set_csc_hr2_network_threshold,  set_csc_hr2_server_timeout,  set_csc_hr2_server_threshold,
set_csc_hr2_checkport,      set_csc_hr2_first_server_address,  set_csc_hr2_first_server_port,
set_csc_hr2_second_server_address, set_csc_hr2_second_server_port.
```

Remarks

[첫 번째 서버 주소]는 NULL을 제외하고 최대 64바이트입니다. 보다 자세한 내용은 CSC-

HR2 제품 사용설명서를 참조하시기 바랍니다.

5.4.9 set_csc_hr2_first_server_port

```
int set_csc_hr2_first_server_port(unsigned char *mac_address, int server_port);
```

Description

*set_csc_hr2_first_server_port*는 CSC-HR2가 통신할 첫 번째 서버의 [포트번호]를 변경합니다.

Parameters

*unsigned char *mac_address*

EzTCP_Search 또는 *EzTCP_Read*를 사용하여 검색된 제품(ezTCP)의 MAC주소 입니다.

int server_port

변경할 첫 번째 서버의 포트번호입니다.

Return values

*set_csc_hr2_first_server_port*의 반환 값은 다음과 같습니다.

1

설정 값 변경에 성공하면 1을 반환합니다.

-1

*mac_address*을 가진 제품(ezTCP)을 찾을 수 없는 경우에는 -1을 반환합니다.

-2

*is_csc_hr2*가 1을 반환하지 않는 경우 -2를 반환합니다.

Examples

See also

is_csc_hr2, *set_csc_hr2_communication_mode*, *set_csc_hr2_id*, *set_csc_hr2_network_timeout*,
set_csc_hr2_network_threshold, *set_csc_hr2_server_timeout*, *set_csc_hr2_server_threshold*,
set_csc_hr2_checkport, *set_csc_hr2_first_server_address*, *set_csc_hr2_first_server_port*,
set_csc_hr2_second_server_address, *set_csc_hr2_second_server_port*.

Remarks

*is_csc_hr2*가 1을 반환하는 경우에만 사용할 수 있습니다. 보다 자세한 내용은 CSC-HR2 제품 사용설명서를 참조하시기 바랍니다.

5.4.10 set_csc_hr2_second_server_address

```
int set_csc_hr2_second_server_address(unsigned char *mac_address,
                                       char *server_address);
```

Description

*set_csc_hr2_second_server_address*는 CSC-HR2의 [두 번째 서버 주소]를 변경합니다.

Parameters

*unsigned char *mac_address*

EzTCP_Search 또는 *EzTCP_Read*를 사용하여 검색된 제품(ezTCP)의 MAC주소 입니다.

*char *server_address*

두 번째 서버 주소가 저장되어있는 포인터입니다.

Return values

*set_csc_hr2_second_server_address*의 반환 값은 다음과 같습니다.

1

설정 값 변경에 성공하면 1을 반환합니다.

-1

*mac_address*을 가진 제품(ezTCP)을 찾을 수 없는 경우에는 -1을 반환합니다.

-2

*is_csc_hr2*가 1을 반환하지 않는 경우 -2를 반환합니다.

Examples

See also

<i>is_csc_hr2,</i>	<i>set_csc_hr2_communication_mode,</i>	<i>set_csc_hr2_id,</i>	<i>set_csc_hr2_network_timeout,</i>
<i>set_csc_hr2_network_threshold,</i>	<i>set_csc_hr2_server_timeout,</i>	<i>set_csc_hr2_server_threshold,</i>	
<i>set_csc_hr2_checkport,</i>	<i>set_csc_hr2_first_server_address,</i>	<i>set_csc_hr2_first_server_port,</i>	
<i>set_csc_hr2_second_server_address,</i>	<i>set_csc_hr2_second_server_port.</i>		

Remarks

[두 번째 서버 주소]는 NULL을 제외하고 최대 64바이트입니다. 보다 자세한 내용은 CSC-HR2 제품 사용설명서를 참조하시기 바랍니다.

5.4.11 set_csc_hr2_second_server_port

```
int set_csc_hr2_second_server_port(unsigned char *mac_address, int server_port);
```

Description

*set_csc_hr2_second_server_port*는 CSC-HR2가 통신할 두 번째 서버의 [포트번호]를 변경합니다.

Parameters

*unsigned char *mac_address*

EzTCP_Search 또는 *EzTCP_Read*를 사용하여 검색된 제품(ezTCP)의 MAC주소 입니다.

int server_port

변경할 두 번째 서버의 포트번호입니다.

Return values

*set_csc_hr2_second_server_port*의 반환 값은 다음과 같습니다.

1

설정 값 변경에 성공하면 1을 반환합니다.

-1

*mac_address*을 가진 제품(ezTCP)을 찾을 수 없는 경우에는 -1을 반환합니다.

-2

*is_csc_hr2*가 1을 반환하지 않는 경우 -2를 반환합니다.

Examples

See also

is_csc_hr2, *set_csc_hr2_communication_mode*, *set_csc_hr2_id*, *set_csc_hr2_network_timeout*,
set_csc_hr2_network_threshold, *set_csc_hr2_server_timeout*, *set_csc_hr2_server_threshold*,
set_csc_hr2_checkport, *set_csc_hr2_first_server_address*, *set_csc_hr2_first_server_port*,
set_csc_hr2_second_server_address, *set_csc_hr2_second_server_port*.

Remarks

보다 자세한 내용은 CSC-HR2 제품 사용설명서를 참조하시기 바랍니다.

5.5 I/O 제품

5.5.1 set_io_http

```
int set_io_http(unsigned char *mac_address, int onoff);
```

Description

*set_io_http*는 I/O 제품 [웹(HTTP)] 제어방식 기능 선택 여부를 변경합니다.

Parameters*unsigned char *mac_address**EzTCP_Search* 또는 *EzTCP_Read*를 사용하여 검색된 제품(ezTCP)의 MAC주소 입니다.*int onoff*

해당 기능의 선택 여부를 결정하며 아래와 같은 값을 가질 수 있습니다.

0

기능 선택을 해제합니다.

1

기능을 선택합니다.

Return values*set_io_http*의 반환 값은 다음과 같습니다.

1

설정 값 변경에 성공하면 1을 반환합니다.

-1

*mac_address*을 가진 제품(ezTCP)을 찾을 수 없는 경우에는 -1을 반환합니다.

-2

*is_io*가 1을 반환하지 않는 경우 -2를 반환합니다.**Examples****See also**

is_io, *set_io_http*, *set_io_http_port*, *set_io_html_size*, *set_io_modbus*, *set_io_input_notification*,
set_io_output_automatic_initialize, *set_io_modbus_type*, *set_io_unit_id*, *set_io_input_address*,
set_io_output_address, *set_io_poll_interval*, *set_io_slave_control_type*, *set_io_master_control_type*,
set_io_tcp_type, *set_io_multi_connection_number*, *set_io_modbus_peer_address*,
set_io_modbus_peer_port, *set_io_modbus_local_port*, *set_io_event_notification*,
set_event_notification_email, *set_io_event_notification_port*, *set_io_input_valid_time*, *set_io_macro*,
set_io_port_macro, *set_io_macro_text*, *set_io_output_delay*, *set_io_output_initial_state*, *set_io_comment*.

Remarks**5.5.2 set_io_http_port**

```
int set_io_http_port(unsigned char *mac_address, int http_port);
```

Description*set_io_http_port*는 I/O 제품 [웹(HTTP) 포트] 번호를 변경합니다.**Parameters**

*unsigned char *mac_address*

EzTCP_Search 또는 *EzTCP_Read*를 사용하여 검색된 제품(ezTCP)의 MAC주소 입니다.

int http_port

변경할 웹(HTTP) 포트번호입니다.

Return values

*set_io_http_port*의 반환 값은 다음과 같습니다.

1

설정 값 변경에 성공하면 1을 반환합니다.

-1

*mac_address*을 가진 제품(ezTCP)을 찾을 수 없는 경우에는 -1을 반환합니다.

-2

*is_io*가 1을 반환하지 않는 경우 -2를 반환합니다.

Examples

See also

is_io, *set_io_http*, *set_io_http_port*, *set_io_html_size*, *set_io_modbus*, *set_io_input_notification*, *set_io_output_automatic_initialize*, *set_io_modbus_type*, *set_io_unit_id*, *set_io_input_address*, *set_io_output_address*, *set_io_poll_interval*, *set_io_slave_control_type*, *set_io_master_control_type*, *set_io_tcp_type*, *set_io_multi_connection_number*, *set_io_modbus_peer_address*, *set_io_modbus_peer_port*, *set_io_modbus_local_port*, *set_io_event_notification*, *set_event_notification_email*, *set_io_event_notification_port*, *set_io_input_valid_time*, *set_io_macro*, *set_io_port_macro*, *set_io_macro_text*, *set_io_output_delay*, *set_io_output_initial_state*, *set_io_comment*.

Remarks

5.5.3 set_io_html_size

```
int set_io_html_size(unsigned char *mac_address, int html_size);
```

Description

*set_io_html_size*는 I/O 제품 [웹(HTTP)페이지 크기]를 변경합니다.

Parameters

*unsigned char *mac_address*

EzTCP_Search 또는 *EzTCP_Read*를 사용하여 검색된 제품(ezTCP)의 MAC주소 입니다.

int html_size

[웹(HTTP)페이지 크기]를 나타내며 아래와 같은 값을 가질 수 있습니다.

0

	80KB
1	
	96KB
2	
	112KB

Return values

*set_io_html_size*의 반환 값은 다음과 같습니다.

1

설정 값 변경에 성공하면 1을 반환합니다.

-1

*mac_address*을 가진 제품(ezTCP)을 찾을 수 없는 경우에는 -1을 반환합니다.

-2

*is_io*가 1을 반환하지 않는 제품인 경우 -2를 반환합니다.

-3

*html_size*에 유효하지 않은 값을 입력한 경우 -3을 반환합니다.

Examples

See also

is_io, *set_io_http*, *set_io_http_port*, *set_io_html_size*, *set_io_modbus*, *set_io_input_notification*, *set_io_output_automatic_initialize*, *set_io_modbus_type*, *set_io_unit_id*, *set_io_input_address*, *set_io_output_address*, *set_io_poll_interval*, *set_io_slave_control_type*, *set_io_master_control_type*, *set_io_tcp_type*, *set_io_multi_connection_number*, *set_io_modbus_peer_address*, *set_io_modbus_peer_port*, *set_io_modbus_local_port*, *set_io_event_notification*, *set_event_notification_email*, *set_io_event_notification_port*, *set_io_input_valid_time*, *set_io_macro*, *set_io_port_macro*, *set_io_macro_text*, *set_io_output_delay*, *set_io_output_initial_state*, *set_io_comment*.

Remarks

5.5.4 set_io_modbus

```
int set_io_modbus(unsigned char *mac_address, int onoff);
```

Description

*set_io_modbus*는 I/O 제품 [Modbus/TCP] 제어방식 기능 선택 여부를 변경합니다.

Parameters

*unsigned char *mac_address*

EzTCP_Search 또는 *EzTCP_Read*를 사용하여 검색된 제품(ezTCP)의 MAC주소 입니다.

int onoff

해당 기능의 선택 여부를 결정하며 아래와 같은 값을 가질 수 있습니다.

0

기능 선택을 해제합니다.

1

기능을 선택합니다.

Return values

*set_io_modbus*의 반환 값은 다음과 같습니다.

1

설정 값 변경에 성공하면 1을 반환합니다.

-1

*mac_address*을 가진 제품(ezTCP)을 찾을 수 없는 경우에는 -1을 반환합니다.

-2

*is_io*가 1을 반환하지 않는 경우 -2를 반환합니다.

Examples

See also

is_io, *set_io_http*, *set_io_http_port*, *set_io_html_size*, *set_io_modbus*, *set_io_input_notification*,
set_io_output_automatic_initialize, *set_io_modbus_type*, *set_io_unit_id*, *set_io_input_address*,
set_io_output_address, *set_io_poll_interval*, *set_io_slave_control_type*, *set_io_master_control_type*,
set_io_tcp_type, *set_io_multi_connection_number*, *set_io_modbus_peer_address*,
set_io_modbus_peer_port, *set_io_modbus_local_port*, *set_io_event_notification*,
set_event_notification_email, *set_io_event_notification_port*, *set_io_input_valid_time*, *set_io_macro*,
set_io_port_macro, *set_io_macro_text*, *set_io_output_delay*, *set_io_output_initial_state*, *set_io_comment*.

Remarks

5.5.5 *set_io_input_notification*

```
int set_io_input_notification(unsigned char *mac_address, int onoff);
```

Description

*set_io_input_notification*은 I/O 제품의 [입력포트 변경 알림] 기능 선택 여부를 변경합니다.

Parameters

*unsigned char *mac_address*

EzTCP_Search 또는 *EzTCP_Read*를 사용하여 검색된 제품(ezTCP)의 MAC주소 입니다.

int onoff

해당 기능의 선택 여부를 결정하며 아래와 같은 값을 가질 수 있습니다.

0

기능 선택을 해제합니다.

1

기능을 선택합니다.

Return values

*set_io_input_notification*의 반환 값은 다음과 같습니다.

1

설정 값 변경에 성공하면 1을 반환합니다.

-1

*mac_address*을 가진 제품(ezTCP)을 찾을 수 없는 경우에는 -1을 반환합니다.

-2

*is_io*가 1을 반환하지 않는 경우 -2를 반환합니다.

Examples

See also

is_io, *set_io_http*, *set_io_http_port*, *set_io_html_size*, *set_io_modbus*, *set_io_input_notification*, *set_io_output_automatic_initialize*, *set_io_modbus_type*, *set_io_unit_id*, *set_io_input_address*, *set_io_output_address*, *set_io_poll_interval*, *set_io_slave_control_type*, *set_io_master_control_type*, *set_io_tcp_type*, *set_io_multi_connection_number*, *set_io_modbus_peer_address*, *set_io_modbus_peer_port*, *set_io_modbus_local_port*, *set_io_event_notification*, *set_event_notification_email*, *set_io_event_notification_port*, *set_io_input_valid_time*, *set_io_macro*, *set_io_port_macro*, *set_io_macro_text*, *set_io_output_delay*, *set_io_output_initial_state*, *set_io_comment*.

Remarks

5.5.6 set_io_output_automatic_initialize

```
int set_io_output_automatic_initialize(unsigned char *mac_address, int onoff);
```

Description

*set_io_output_automatic_initialize*는 I/O 제품의 [출력포트 상태 초기화(Modbus/TCP 접속 종료 시 출력포트[초기상태] 값으로 상태 변경) 기능 선택 여부를 변경합니다.

Parameters

*unsigned char *mac_address*

EzTCP_Search 또는 *EzTCP_Read*를 사용하여 검색된 제품(ezTCP)의 MAC주소 입니다.

int onoff

해당 기능의 선택 여부를 결정하며 아래와 같은 값을 가질 수 있습니다.

0

기능 선택을 해제합니다.

1

기능을 선택합니다.

Return values

*set_io_output_automatic_initialize*의 반환 값은 다음과 같습니다.

1

설정 값 변경에 성공하면 1을 반환합니다.

-1

*mac_address*을 가진 제품(ezTCP)을 찾을 수 없는 경우에는 -1을 반환합니다.

-2

*is_io*가 1을 반환하지 않는 경우 -2를 반환합니다.

-3

*is_io_output_automatic_initialize*가 1을 반환하지 않는 경우 -3을 반환합니다.

Examples

See also

is_io, *set_io_http*, *set_io_http_port*, *set_io_html_size*, *set_io_modbus*, *set_io_input_notification*,
is_io_output_automatic_initialize, *set_io_output_automatic_initialize*, *set_io_modbus_type*, *set_io_unit_id*,
set_io_input_address, *set_io_output_address*, *set_io_poll_interval*, *set_io_slave_control_type*,
set_io_master_control_type, *set_io_tcp_type*, *set_io_multi_connection_number*,
set_io_modbus_peer_address, *set_io_modbus_peer_port*, *set_io_modbus_local_port*,
set_io_event_notification, *set_event_notification_email*, *set_io_event_notification_port*,
set_io_input_valid_time, *set_io_macro*, *set_io_port_macro*, *set_io_macro_text*, *set_io_output_delay*,
set_io_output_initial_state, *set_io_comment*.

Remarks

보다 자세한 내용은 해당 제품 사용설명서를 참조하시기 바랍니다.

5.5.7 set_io_modbus_type

```
int set_io_modbus_type(unsigned char *mac_address, int modbus_type);
```

Description

*set_io_modbus_type*은 I/O 제품의 [마스터/슬레이브]를 변경합니다.

Parameters

*unsigned char *mac_address*

EzTCP_Search 또는 *EzTCP_Read*를 사용하여 검색된 제품(ezTCP)의 MAC주소 입니다.

int modbus_type

[마스터/슬레이브]종류를 나타내며 아래와 같은 값을 가질 수 있습니다.

0
 슬레이브
1
 마스터

Return values

*set_io_modbus_type*의 반환 값은 다음과 같습니다.

1
 설정 값 변경에 성공하면 1을 반환합니다.
-1
 *mac_address*을 가진 제품(ezTCP)을 찾을 수 없는 경우에는 -1을 반환합니다.
-2
 *is_io*가 1을 반환하지 않는 경우 -2를 반환합니다.
-3
 *modbus_type*에 유효하지 않은 값을 입력한 경우 -3을 반환합니다.

Examples

See also

is_io, *set_io_http*, *set_io_http_port*, *set_io_html_size*, *set_io_modbus*, *set_io_input_notification*,
set_io_output_automatic_initialize, *set_io_modbus_type*, *set_io_unit_id*, *set_io_input_address*,
set_io_output_address, *set_io_poll_interval*, *set_io_slave_control_type*, *set_io_master_control_type*,
set_io_tcp_type, *set_io_multi_connection_number*, *set_io_modbus_peer_address*,
set_io_modbus_peer_port, *set_io_modbus_local_port*, *set_io_event_notification*,
set_event_notification_email, *set_io_event_notification_port*, *set_io_input_valid_time*, *set_io_macro*,
set_io_port_macro, *set_io_macro_text*, *set_io_output_delay*, *set_io_output_initial_state*, *set_io_comment*.

Remarks

5.5.8 set_io_unit_id

```
int set_io_unit_id(unsigned char *mac_address, int unit_id);
```

Description

*set_io_unit_id*는 I/O 제품의 [유니트 아이디]를 변경합니다.

Parameters*unsigned char *mac_address**EzTCP_Search* 또는 *EzTCP_Read*를 사용하여 검색된 제품(ezTCP)의 MAC주소 입니다.*int unit_id*

변경할 유닛 아이디입니다.

Return values*set_io_unit_id*의 반환 값은 다음과 같습니다.*1*

설정 값 변경에 성공하면 1을 반환합니다.

*-1**mac_address*을 가진 제품(ezTCP)을 찾을 수 없는 경우에는 -1을 반환합니다.*-2**is_io*가 1을 반환하지 않는 경우 -2를 반환합니다.**Examples****See also**

is_io, *set_io_http*, *set_io_http_port*, *set_io_html_size*, *set_io_modbus*, *set_io_input_notification*,
set_io_output_automatic_initialize, *set_io_modbus_type*, *set_io_unit_id*, *set_io_input_address*,
set_io_output_address, *set_io_poll_interval*, *set_io_slave_control_type*, *set_io_master_control_type*,
set_io_tcp_type, *set_io_multi_connection_number*, *set_io_modbus_peer_address*,
set_io_modbus_peer_port, *set_io_modbus_local_port*, *set_io_event_notification*,
set_event_notification_email, *set_io_event_notification_port*, *set_io_input_valid_time*, *set_io_macro*,
set_io_port_macro, *set_io_macro_text*, *set_io_output_delay*, *set_io_output_initial_state*, *set_io_comment*.

Remarks**5.5.9 set_io_input_address**

```
int set_io_input_address(unsigned char *mac_address, int input_address);
```

Description*set_io_input_address*는 I/O 제품의 [입력포트 주소]를 변경합니다.**Parameters***unsigned char *mac_address**EzTCP_Search* 또는 *EzTCP_Read*를 사용하여 검색된 제품(ezTCP)의 MAC주소 입니다.*int input_address*

변경할 입력포트 주소입니다.

Return values

*set_io_input_address*의 반환 값은 다음과 같습니다.

1

설정 값 변경에 성공하면 1을 반환합니다.

-1

*mac_address*을 가진 제품(ezTCP)을 찾을 수 없는 경우에는 -1을 반환합니다.

-2

*is_io*가 1을 반환하지 않는 경우 -2를 반환합니다.

-3

*input_address*에 유효하지 않은 값을 입력한 경우 -3을 반환합니다.

Examples

See also

is_io, *set_io_http*, *set_io_http_port*, *set_io_html_size*, *set_io_modbus*, *set_io_input_notification*,
set_io_output_automatic_initialize, *set_io_modbus_type*, *set_io_unit_id*, *set_io_input_address*,
set_io_output_address, *set_io_poll_interval*, *set_io_slave_control_type*, *set_io_master_control_type*,
set_io_tcp_type, *set_io_multi_connection_number*, *set_io_modbus_peer_address*,
set_io_modbus_peer_port, *set_io_modbus_local_port*, *set_io_event_notification*,
set_event_notification_email, *set_io_event_notification_port*, *set_io_input_valid_time*, *set_io_macro*,
set_io_port_macro, *set_io_macro_text*, *set_io_output_delay*, *set_io_output_initial_state*, *set_io_comment*.

Remarks

5.5.10 set_io_output_address

```
int set_io_output_address(unsigned char *mac_address, int output_address);
```

Description

*set_io_output_address*는 I/O 제품의 [출력포트 주소]를 변경합니다.

Parameters

*unsigned char *mac_address*

EzTCP_Search 또는 *EzTCP_Read*를 사용하여 검색된 제품(ezTCP)의 MAC주소 입니다.

int input_address

변경할 출력포트 주소입니다.

Return values

*set_io_output_address*의 반환 값은 다음과 같습니다.

1

설정 값 변경에 성공하면 1을 반환합니다.

-1

*mac_address*을 가진 제품(*ezTCP*)을 찾을 수 없는 경우에는 -1을 반환합니다.

-2

*is_io*가 1을 반환하지 않는 경우 -2를 반환합니다.

-3

*output_address*에 유효하지 않은 값을 입력한 경우 -3을 반환합니다.

Examples

See also

is_io, *set_io_http*, *set_io_http_port*, *set_io_html_size*, *set_io_modbus*, *set_io_input_notification*, *set_io_output_automatic_initialize*, *set_io_modbus_type*, *set_io_unit_id*, *set_io_input_address*, *set_io_output_address*, *set_io_poll_interval*, *set_io_slave_control_type*, *set_io_master_control_type*, *set_io_tcp_type*, *set_io_multi_connection_number*, *set_io_modbus_peer_address*, *set_io_modbus_peer_port*, *set_io_modbus_local_port*, *set_io_event_notification*, *set_event_notification_email*, *set_io_event_notification_port*, *set_io_input_valid_time*, *set_io_macro*, *set_io_port_macro*, *set_io_macro_text*, *set_io_output_delay*, *set_io_output_initial_state*, *set_io_comment*.

Remarks

5.5.11 *set_io_poll_interval*

```
int set_io_poll_interval(unsigned char *mac_address, int poll_interval);
```

Description

*set_io_poll_interval*은 I/O 제품의 마스터 [통신 주기]를 변경합니다.

Parameters

*unsigned char *mac_address*

EzTCP_Search 또는 *EzTCP_Read*를 사용하여 검색된 제품(*ezTCP*)의 MAC주소 입니다.

int poll_interval

변경할 통신 주기입니다.

Return values

*set_io_poll_interval*의 반환 값은 다음과 같습니다.

1

설정 값 변경에 성공하면 1을 반환합니다.

-1

*mac_address*을 가진 제품(ezTCP)을 찾을 수 없는 경우에는 -1을 반환합니다.

-2

*is_io*가 1을 반환하지 않는 경우 -2를 반환합니다.

Examples

See also

is_io, *set_io_http*, *set_io_http_port*, *set_io_html_size*, *set_io_modbus*, *set_io_input_notification*,
set_io_output_automatic_initialize, *set_io_modbus_type*, *set_io_unit_id*, *set_io_input_address*,
set_io_output_address, *set_io_poll_interval*, *set_io_slave_control_type*, *set_io_master_control_type*,
set_io_tcp_type, *set_io_multi_connection_number*, *set_io_modbus_peer_address*,
set_io_modbus_peer_port, *set_io_modbus_local_port*, *set_io_event_notification*,
set_event_notification_email, *set_io_event_notification_port*, *set_io_input_valid_time*, *set_io_macro*,
set_io_port_macro, *set_io_macro_text*, *set_io_output_delay*, *set_io_output_initial_state*, *set_io_comment*.

Remarks

5.5.12 *set_io_slave_control_type*

```
int set_io_slave_control_type(unsigned char *mac_address, int slave_control_type);
```

Description

*set_io_slave_control_type*은 I/O 제품의 [슬레이브 출력포트 제어방식]을 변경합니다.

Parameters

*unsigned char *mac_address*

EzTCP_Search 또는 *EzTCP_Read*를 사용하여 검색된 제품(ezTCP)의 MAC주소 입니다.

int slave_control_type

[슬레이브 출력포트 제어방식]종류를 나타내며 아래와 같은 값을 가질 수 있습니다.

0

FC 16 (동시제어)

1

FC 05 (개별제어)

Return values

*set_io_slave_control_type*의 반환 값은 다음과 같습니다.

1

설정 값 변경에 성공하면 1을 반환합니다.

-1

*mac_address*을 가진 제품(ezTCP)을 찾을 수 없는 경우에는 -1을 반환합니다.

-2

*is_io*가 1을 반환하지 않는 경우 -2를 반환합니다.

-3

*slave_control_type*에 유효하지 않은 값을 입력한 경우 -3을 반환합니다.

Examples

See also

is_io, *set_io_http*, *set_io_http_port*, *set_io_html_size*, *set_io_modbus*, *set_io_input_notification*,
set_io_output_automatic_initialize, *set_io_modbus_type*, *set_io_unit_id*, *set_io_input_address*,
set_io_output_address, *set_io_poll_interval*, *set_io_slave_control_type*, *set_io_master_control_type*,
set_io_tcp_type, *set_io_multi_connection_number*, *set_io_modbus_peer_address*,
set_io_modbus_peer_port, *set_io_modbus_local_port*, *set_io_event_notification*,
set_event_notification_email, *set_io_event_notification_port*, *set_io_input_valid_time*, *set_io_macro*,
set_io_port_macro, *set_io_macro_text*, *set_io_output_delay*, *set_io_output_initial_state*, *set_io_comment*.

Remarks

보다 자세한 내용은 제품 사용설명서를 참조하시기 바랍니다.

5.5.13 set_io_master_control_type

```
int set_io_master_control_type(unsigned char *mac_address, int master_control_type);
```

Description

*set_io_master_control_type*은 I/O 제품의 [마스터 출력포트 제어방식]을 반환합니다.

Parameters

*unsigned char *mac_address*

EzTCP_Search 또는 *EzTCP_Read*를 사용하여 검색된 제품(ezTCP)의 MAC주소 입니다.

int master_control_type

[마스터 출력포트 제어방식]종류를 나타내며 아래와 같은 값을 가질 수 있습니다.

0

AND

1

OR

Return values

*set_io_master_control_type*의 반환 값은 다음과 같습니다.

1

설정 값 변경에 성공하면 1을 반환합니다.

-1

*mac_address*을 가진 제품(ezTCP)을 찾을 수 없는 경우에는 -1을 반환합니다.

-2

*is_io*가 1을 반환하지 않는 경우 -2를 반환합니다.

-3

*master_control_type*에 유효하지 않은 값을 입력한 경우 -3을 반환합니다.

Examples

See also

is_io, *set_io_http*, *set_io_http_port*, *set_io_html_size*, *set_io_modbus*, *set_io_input_notification*, *set_io_output_automatic_initialize*, *set_io_modbus_type*, *set_io_unit_id*, *set_io_input_address*, *set_io_output_address*, *set_io_poll_interval*, *set_io_slave_control_type*, *set_io_master_control_type*, *set_io_tcp_type*, *set_io_multi_connection_number*, *set_io_modbus_peer_address*, *set_io_modbus_peer_port*, *set_io_modbus_local_port*, *set_io_event_notification*, *set_event_notification_email*, *set_io_event_notification_port*, *set_io_input_valid_time*, *set_io_macro*, *set_io_port_macro*, *set_io_macro_text*, *set_io_output_delay*, *set_io_output_initial_state*, *set_io_comment*.

Remarks

보다 자세한 내용은 제품 사용설명서를 참조하시기 바랍니다.

5.5.14 set_io_tcp_type

```
int set_io_tcp_type(unsigned char *mac_address, int tcp_type);
```

Description

*set_io_tcp_type*은 I/O 제품의 Modbus/TCP 접속 방법을 변경합니다.

Parameters

*unsigned char *mac_address*

EzTCP_Search 또는 *EzTCP_Read*를 사용하여 검색된 제품(ezTCP)의 MAC주소 입니다.

int master_control_type

[마스터 출력포트 제어방식]종류를 나타내며 아래와 같은 값을 가질 수 있습니다.

0

수동접속

1

능동접속

Return values

*set_io_tcp_type*의 반환 값은 다음과 같습니다.

1

설정 값 변경에 성공하면 1을 반환합니다.

-1

*mac_address*을 가진 제품(ezTCP)을 찾을 수 없는 경우에는 -1을 반환합니다.

-2

*is_io*가 1을 반환하지 않는 경우 -2를 반환합니다.

-3

*tcp_type*에 유효하지 않은 값을 입력한 경우 -3을 반환합니다.

Examples

See also

is_io, *set_io_http*, *set_io_http_port*, *set_io_html_size*, *set_io_modbus*, *set_io_input_notification*, *set_io_output_automatic_initialize*, *set_io_modbus_type*, *set_io_unit_id*, *set_io_input_address*, *set_io_output_address*, *set_io_poll_interval*, *set_io_slave_control_type*, *set_io_master_control_type*, *set_io_tcp_type*, *set_io_multi_connection_number*, *set_io_modbus_peer_address*, *set_io_modbus_peer_port*, *set_io_modbus_local_port*, *set_io_event_notification*, *set_event_notification_email*, *set_io_event_notification_port*, *set_io_input_valid_time*, *set_io_macro*, *set_io_port_macro*, *set_io_macro_text*, *set_io_output_delay*, *set_io_output_initial_state*, *set_io_comment*.

Remarks

보다 자세한 내용은 제품 사용설명서를 참조하시기 바랍니다.

5.5.15 *set_io_multi_connection_number*

```
int set_io_multi_connection_number(unsigned char *mac_address, int number);
```

Description

*set_io_multi_connection_number*은 I/O 제품이 Modbus/TCP 수동접속인 경우 동시 접속 가능한 Modbus/TCP 클라이언트의 개수를 변경합니다.

Parameters

*unsigned char *mac_address*

EzTCP_Search 또는 *EzTCP_Read*를 사용하여 검색된 제품(ezTCP)의 MAC주소 입니다.

int number

변경할 동시 접속 가능한 Modbus/TCP 클라이언트의 개수로 1 ~ 8입니다.

Return values

*set_io_multi_connection_number*의 반환 값은 다음과 같습니다.

1

설정 값 변경에 성공하면 1을 반환합니다.

-1

*mac_address*을 가진 제품(ezTCP)을 찾을 수 없는 경우에는 -1을 반환합니다.

-2

*is_io*가 1을 반환하지 않는 경우 -2를 반환합니다.

-3

*number*에 유효하지 않은 값을 입력한 경우 -3을 반환합니다.

Examples

See also

is_io, *set_io_http*, *set_io_http_port*, *set_io_html_size*, *set_io_modbus*, *set_io_input_notification*,
set_io_output_automatic_initialize, *set_io_modbus_type*, *set_io_unit_id*, *set_io_input_address*,
set_io_output_address, *set_io_poll_interval*, *set_io_slave_control_type*, *set_io_master_control_type*,
set_io_tcp_type, *set_io_multi_connection_number*, *set_io_modbus_peer_address*,
set_io_modbus_peer_port, *set_io_modbus_local_port*, *set_io_event_notification*,
set_event_notification_email, *set_io_event_notification_port*, *set_io_input_valid_time*, *set_io_macro*,
set_io_port_macro, *set_io_macro_text*, *set_io_output_delay*, *set_io_output_initial_state*, *set_io_comment*.

Remarks

5.5.16 *set_io_modbus_peer_address*

```
int set_io_modbus_peer_address(unsigned char *mac_address, char *peer_address);
```

Description

*set_io_modbus_peer_address*는 I/O 제품이 Modbus/TCP 능동접속인 경우 [통신할 주소]를 변경합니다.

Parameters

*unsigned char *mac_address*

EzTCP_Search 또는 *EzTCP_Read*를 사용하여 검색된 제품(ezTCP)의 MAC주소 입니다.

*char *peer_address*

통신할 주소가 저장되어있는 포인터입니다. 통신할 주소에는 IPv4의 점으로 구분된 십진 수 표기법으로 된 IP 주소 또는 호스트 이름을 입력해야 합니다.

Return values

*set_io_modbus_peer_address*의 반환 값은 다음과 같습니다.

1

설정 값 변경에 성공하면 1을 반환합니다.

-1

*mac_address*을 가진 제품(ezTCP)을 찾을 수 없는 경우에는 -1을 반환합니다.

-2

*is_io*가 1을 반환하지 않는 경우 -2를 반환합니다.

Examples

See also

is_io, *set_io_http*, *set_io_http_port*, *set_io_html_size*, *set_io_modbus*, *set_io_input_notification*, *set_io_output_automatic_initialize*, *set_io_modbus_type*, *set_io_unit_id*, *set_io_input_address*, *set_io_output_address*, *set_io_poll_interval*, *set_io_slave_control_type*, *set_io_master_control_type*, *set_io_tcp_type*, *set_io_multi_connection_number*, *set_io_modbus_peer_address*, *set_io_modbus_peer_port*, *set_io_modbus_local_port*, *set_io_event_notification*, *set_event_notification_email*, *set_io_event_notification_port*, *set_io_input_valid_time*, *set_io_macro*, *set_io_port_macro*, *set_io_macro_text*, *set_io_output_delay*, *set_io_output_initial_state*, *set_io_comment*.

Remarks

[통신할 주소]는 NULL을 제외하고 최대 64바이트입니다.

5.5.17 set_io_modbus_peer_port

```
int set_io_modbus_peer_port(unsigned char *mac_address, int peer_port);
```

Description

*set_io_modbus_peer_port*는 I/O 제품이 Modbus/TCP 능동접속인 경우 [통신할 포트]를 변경합니다.

Parameters

*unsigned char *mac_address*

EzTCP_Search 또는 *EzTCP_Read*를 사용하여 검색된 제품(ezTCP)의 MAC주소 입니다.

int peer_port

변경할 통신할 포트입니다.

Return values

*set_io_modbus_peer_port*의 반환 값은 다음과 같습니다.

1

설정 값 변경에 성공하면 1을 반환합니다.

-1

*mac_address*을 가진 제품(ezTCP)을 찾을 수 없는 경우에는 -1을 반환합니다.

-2

*is_io*가 1을 반환하지 않는 경우 -2를 반환합니다.

Examples

See also

is_io, *set_io_http*, *set_io_http_port*, *set_io_html_size*, *set_io_modbus*, *set_io_input_notification*, *set_io_output_automatic_initialize*, *set_io_modbus_type*, *set_io_unit_id*, *set_io_input_address*, *set_io_output_address*, *set_io_poll_interval*, *set_io_slave_control_type*, *set_io_master_control_type*, *set_io_tcp_type*, *set_io_multi_connection_number*, *set_io_modbus_peer_address*, *set_io_modbus_peer_port*, *set_io_modbus_local_port*, *set_io_event_notification*, *set_event_notification_email*, *set_io_event_notification_port*, *set_io_input_valid_time*, *set_io_macro*, *set_io_port_macro*, *set_io_macro_text*, *set_io_output_delay*, *set_io_output_initial_state*, *set_io_comment*.

Remarks

5.5.18 *set_io_modbus_local_port*

```
int set_io_modbus_local_port(unsigned char *mac_address, int local_port);
```

Description

*set_io_modbus_local_port*는 I/O 제품이 Modbus/TCP 수동접속인 경우 Modbus/TCP [제품 로컬포트]를 변경합니다.

Parameters

*unsigned char *mac_address*

EzTCP_Search 또는 *EzTCP_Read*를 사용하여 검색된 제품(ezTCP)의 MAC주소 입니다.

int peer_port

변경할 제품 로컬포트입니다.

Return values

*set_io_modbus_local_port*의 반환 값은 다음과 같습니다.

1

설정 값 변경에 성공하면 1을 반환합니다.

-1

*mac_address*을 가진 제품(ezTCP)을 찾을 수 없는 경우에는 -1을 반환합니다.

-2

*is_io*가 1을 반환하지 않는 경우 -2를 반환합니다.

Examples

See also

is_io, set_io_http, set_io_http_port, set_io_html_size, set_io_modbus, set_io_input_notification, set_io_output_automatic_initialize, set_io_modbus_type, set_io_unit_id, set_io_input_address, set_io_output_address, set_io_poll_interval, set_io_slave_control_type, set_io_master_control_type, set_io_tcp_type, set_io_multi_connection_number, set_io_modbus_peer_address, set_io_modbus_peer_port, set_io_modbus_local_port, set_io_event_notification, set_event_notification_email, set_io_event_notification_port, set_io_input_valid_time, set_io_macro, set_io_port_macro, set_io_macro_text, set_io_output_delay, set_io_output_initial_state, set_io_comment.

Remarks

5.5.19 set_io_event_notification

```
int set_io_event_notification(unsigned char *mac_address, int onoff)
```

Description

*set_io_event_notification*은 I/O 제품의 [입력 또는 출력포트 변경 알림(전자메일)] 기능 선택 여부를 변경합니다.

Parameters

*unsigned char *mac_address*

EzTCP_Search 또는 *EzTCP_Read*를 사용하여 검색된 제품(ezTCP)의 MAC주소 입니다.

int onoff

해당 기능의 선택 여부를 결정하며 아래와 같은 값을 가질 수 있습니다.

0

기능 선택을 해제합니다.

1

기능을 선택합니다.

Return values

*set_io_event_notification*의 반환 값은 다음과 같습니다.

1

설정 값 변경에 성공하면 1을 반환합니다.

-1

*mac_address*을 가진 제품(ezTCP)을 찾을 수 없는 경우에는 -1을 반환합니다.

-2

*is_io*가 1을 반환하지 않는 경우 -2를 반환합니다.

-3

*is_event_notification*이 1을 반환하지 않는 경우 -3을 반환합니다.

Examples

See also

is_io, set_io_http, set_io_http_port, set_io_html_size, set_io_modbus, set_io_input_notification, set_io_output_automatic_initialize, set_io_modbus_type, set_io_unit_id, set_io_input_address, set_io_output_address, set_io_poll_interval, set_io_slave_control_type, set_io_master_control_type, set_io_tcp_type, set_io_multi_connection_number, set_io_modbus_peer_address, set_io_modbus_peer_port, set_io_modbus_local_port, is_event_notification, set_io_event_notification, set_event_notification_email, set_io_event_notification_port, set_io_input_valid_time, set_io_macro, set_io_port_macro, set_io_macro_text, set_io_output_delay, set_io_output_initial_state, set_io_comment.

Remarks

5.5.20 set_event_notification_email

```
int set_event_notification_email(unsigned char *mac_address, char *email);
```

Description

*set_event_notification_email*은 I/O 제품의 입력 또는 출력포트에 변경이 발생한 경우 알림을 발송할 [전자메일 주소]를 변경합니다.

Parameters

*unsigned char *mac_address*

EzTCP_Search 또는 *EzTCP_Read*를 사용하여 검색된 제품(ezTCP)의 MAC주소 입니다.

*char *email*

전자메일 주소가 저장되어있는 포인터입니다.

Return values

*set_event_notification_email*의 반환 값은 다음과 같습니다.

1

설정 값 변경에 성공하면 1을 반환합니다.

-1

*mac_address*을 가진 제품(ezTCP)을 찾을 수 없는 경우에는 -1을 반환합니다.

-2

*is_event_notification*이 1을 반환하지 않는 경우 -2을 반환합니다.

Examples

See also

is_io, set_io_http, set_io_http_port, set_io_html_size, set_io_modbus, set_io_input_notification,

set_io_output_automatic_initialize, set_io_modbus_type, set_io_unit_id, set_io_input_address,
 set_io_output_address, set_io_poll_interval, set_io_slave_control_type, set_io_master_control_type,
 set_io_tcp_type, set_io_multi_connection_number, set_io_modbus_peer_address,
 set_io_modbus_peer_port, set_io_modbus_local_port, is_event_notification, set_io_event_notification,
 set_event_notification_email, set_io_event_notification_port, set_io_input_valid_time, set_io_macro,
 set_io_port_macro, set_io_macro_text, set_io_output_delay, set_io_output_initial_state, set_io_comment.

Remarks

[전자메일 주소]는 NULL을 포함하여 최대 64바이트입니다.

5.5.21 set_io_event_notification_port

```
int set_io_event_notification_port(unsigned char *mac_address, int port_flag,  

                                   int port_index, int onoff);
```

Description

*set_io_event_notification_port*는 I/O 제품에 설치된 입력 또는 출력포트의 [입력 또는 출력 포트 변경 알림(전자메일)] 기능 선택 여부를 변경합니다.

Parameters

*unsigned char *mac_address*

EzTCP_Search 또는 *EzTCP_Read*를 사용하여 검색된 제품(ezTCP)의 MAC주소 입니다.

int port_flag

*port_flag*가 0이면 입력포트를 의미하고 1이면 출력포트를 의미합니다.

int port_index

입력 또는 출력포트의 순번으로 0부터 시작합니다.

int onoff

해당 기능의 선택 여부를 결정하며 아래와 같은 값을 가질 수 있습니다.

0

기능 선택을 해제합니다.

1

기능을 선택합니다.

Return values

*set_io_event_notification_port*의 반환 값은 다음과 같습니다.

1

설정 값 변경에 성공하면 1을 반환합니다.

-1

*mac_address*을 가진 제품(ezTCP)을 찾을 수 없는 경우에는 -1을 반환합니다.

-2

*is_io*가 1을 반환하지 않는 경우 -2를 반환합니다.

-3

*is_event_notification*이 1을 반환하지 않는 경우 -3을 반환합니다.

Examples

See also

is_io, *set_io_http*, *set_io_http_port*, *set_io_html_size*, *set_io_modbus*, *set_io_input_notification*, *set_io_output_automatic_initialize*, *set_io_modbus_type*, *set_io_unit_id*, *set_io_input_address*, *set_io_output_address*, *set_io_poll_interval*, *set_io_slave_control_type*, *set_io_master_control_type*, *set_io_tcp_type*, *set_io_multi_connection_number*, *set_io_modbus_peer_address*, *set_io_modbus_peer_port*, *set_io_modbus_local_port*, *is_event_notification*, *set_io_event_notification*, *set_event_notification_email*, *set_io_event_notification_port*, *set_io_input_valid_time*, *set_io_macro*, *set_io_port_macro*, *set_io_macro_text*, *set_io_output_delay*, *set_io_output_initial_state*, *set_io_comment*.

Remarks

5.5.22 *set_io_input_valid_time*

```
int set_io_input_valid_time(unsigned char *mac_address, int port_index,
                           int input_valid_time);
```

Description

*set_io_input_valid_time*은 *port_index*로 지정된 입력포트의 [신호유지 시간(밀리초)]를 변경합니다.

Parameters

*unsigned char *mac_address*

EzTCP_Search 또는 *EzTCP_Read*를 사용하여 검색된 제품(ezTCP)의 MAC주소 입니다.

int port_index

입력포트의 순번으로 0부터 시작합니다.

int input_valid_time

*port_index*로 지정된 입력포트의 신호유지 시간(밀리초)입니다.

Return values

*set_io_input_valid_time*의 반환 값은 다음과 같습니다.

1

설정 값 변경에 성공하면 1을 반환합니다.

-1

*mac_address*을 가진 제품(ezTCP)을 찾을 수 없는 경우에는 -1을 반환합니다.

-2

*is_io*가 1을 반환하지 않는 경우 -2를 반환합니다.

-3

*input_valid_time*에 유효하지 않은 값을 입력한 경우 -3을 반환합니다.

Examples

See also

is_io, *set_io_http*, *set_io_http_port*, *set_io_html_size*, *set_io_modbus*, *set_io_input_notification*, *set_io_output_automatic_initialize*, *set_io_modbus_type*, *set_io_unit_id*, *set_io_input_address*, *set_io_output_address*, *set_io_poll_interval*, *set_io_slave_control_type*, *set_io_master_control_type*, *set_io_tcp_type*, *set_io_multi_connection_number*, *set_io_modbus_peer_address*, *set_io_modbus_peer_port*, *set_io_modbus_local_port*, *set_io_event_notification*, *set_event_notification_email*, *set_io_event_notification_port*, *set_io_input_valid_time*, *set_io_macro*, *set_io_port_macro*, *set_io_macro_text*, *set_io_output_delay*, *set_io_output_initial_state*, *set_io_comment*.

Remarks

5.5.23 set_io_macro

```
int set_io_macro(unsigned char *mac_address, int onoff);
```

Description

*set_io_macro*는 I/O 제품의 [매크로 기능] 선택 여부를 변경합니다. *get_io_macro_type*이 0을 반환하는 경우에만 사용할 수 있습니다.

Parameters

*unsigned char *mac_address*

EzTCP_Search 또는 *EzTCP_Read*를 사용하여 검색된 제품(ezTCP)의 MAC주소 입니다.

int onoff

해당 기능의 선택 여부를 결정하며 아래와 같은 값을 가질 수 있습니다.

0

기능 선택을 해제합니다.

1

기능을 선택합니다.

Return values

*set_io_macro*의 반환 값은 다음과 같습니다.

1

설정 값 변경에 성공하면 1을 반환합니다.

-1

*mac_address*을 가진 제품(*ezTCP*)을 찾을 수 없는 경우에는 -1을 반환합니다.

-2

*is_io*가 1을 반환하지 않는 경우 -2를 반환합니다.

-3

*get_io_macro_type*이 0을 반환하지 않는 경우 -3을 반환합니다.

Examples

See also

is_io, *set_io_http*, *set_io_http_port*, *set_io_html_size*, *set_io_modbus*, *set_io_input_notification*, *set_io_output_automatic_initialize*, *set_io_modbus_type*, *set_io_unit_id*, *set_io_input_address*, *set_io_output_address*, *set_io_poll_interval*, *set_io_slave_control_type*, *set_io_master_control_type*, *set_io_tcp_type*, *set_io_multi_connection_number*, *set_io_modbus_peer_address*, *set_io_modbus_peer_port*, *set_io_modbus_local_port*, *set_io_event_notification*, *set_event_notification_email*, *set_io_event_notification_port*, *set_io_input_valid_time*, *set_io_macro*, *set_io_port_macro*, *set_io_macro_text*, *set_io_output_delay*, *set_io_output_initial_state*, *set_io_comment*.

Remarks

5.5.24 set_io_port_macro

```
int set_io_port_macro(unsigned char *mac_address, int port_index, int onoff);
```

Description

*set_io_port_macro*는 *port_index*로 지정된 출력포트의 [매크로 기능] 선택 여부를 변경합니다. *get_io_macro_type*이 1을 반환하는 경우에만 사용할 수 있습니다.

Parameters

*unsigned char *mac_address*

EzTCP_Search 또는 *EzTCP_Read*를 사용하여 검색된 제품(*ezTCP*)의 MAC주소 입니다.

int port_index

출력포트의 순번으로 0부터 시작합니다.

int onoff

해당 기능의 선택 여부를 결정하며 아래와 같은 값을 가질 수 있습니다.

0

기능 선택을 해제합니다.

1

기능을 선택합니다.

Return values

*set_io_macro_port*의 반환 값은 다음과 같습니다.

1

설정 값 변경에 성공하면 1을 반환합니다.

-1

*mac_address*을 가진 제품(ezTCP)을 찾을 수 없는 경우에는 -1을 반환합니다.

-2

*is_io*가 1을 반환하지 않는 경우 -2를 반환합니다.

-3

*get_io_macro_type*이 1을 반환하지 않는 경우 -3을 반환합니다.

-4

*port_index*에 유효하지 않은 값을 입력한 경우 -4를 반환합니다.

Examples**See also**

is_io, *set_io_http*, *set_io_http_port*, *set_io_html_size*, *set_io_modbus*, *set_io_input_notification*,
set_io_output_automatic_initialize, *set_io_modbus_type*, *set_io_unit_id*, *set_io_input_address*,
set_io_output_address, *set_io_poll_interval*, *set_io_slave_control_type*, *set_io_master_control_type*,
set_io_tcp_type, *set_io_multi_connection_number*, *set_io_modbus_peer_address*,
set_io_modbus_peer_port, *set_io_modbus_local_port*, *set_io_event_notification*,
set_event_notification_email, *set_io_event_notification_port*, *set_io_input_valid_time*, *set_io_macro*,
set_io_port_macro, *set_io_macro_text*, *set_io_output_delay*, *set_io_output_initial_state*, *set_io_comment*.

Remarks**5.5.25 set_io_macro_text**

```
int set_io_macro_text(unsigned char *mac_address, int port_index, char *macro_text);
```

Description

*set_io_macro_text*는 *port_index*로 지정된 출력포트의 매크로 문자열을 변경합니다.

Parameters

*unsigned char *mac_address*

EzTCP_Search 또는 *EzTCP_Read*를 사용하여 검색된 제품(ezTCP)의 MAC주소 입니다.

int port_index

출력포트의 순번으로 0부터 시작합니다.

*char *macro_text*

*port_index*로 지정된 출력포트의 매크로 문자열입니다.

Return values

*set_io_macro_text*의 반환 값은 다음과 같습니다.

1

설정 값 변경에 성공하면 1을 반환합니다.

-1

*mac_address*을 가진 제품(ezTCP)을 찾을 수 없는 경우에는 -1을 반환합니다.

-2

*is_io*가 1을 반환하지 않는 경우 -2를 반환합니다.

-3

*macro_text*에 입력한 매크로 문자열이 문법에 맞지 않으면 -3을 반환합니다.

-4

*port_index*에 유효하지 않은 값을 입력한 경우 -4를 반환합니다.

Examples

See also

is_io, *set_io_http*, *set_io_http_port*, *set_io_html_size*, *set_io_modbus*, *set_io_input_notification*, *set_io_output_automatic_initialize*, *set_io_modbus_type*, *set_io_unit_id*, *set_io_input_address*, *set_io_output_address*, *set_io_poll_interval*, *set_io_slave_control_type*, *set_io_master_control_type*, *set_io_tcp_type*, *set_io_multi_connection_number*, *set_io_modbus_peer_address*, *set_io_modbus_peer_port*, *set_io_modbus_local_port*, *set_io_event_notification*, *set_event_notification_email*, *set_io_event_notification_port*, *set_io_input_valid_time*, *set_io_macro*, *set_io_port_macro*, *set_io_macro_text*, *set_io_output_delay*, *set_io_output_initial_state*, *set_io_comment*.

Remarks

매크로 문자열은 NULL을 제외하고 최대 32바이트입니다. 매크로 문자열 길이는 *IO_SCRIPT_LEN*에 정의되어 있습니다.

5.5.26 set_io_output_delay

```
int set_io_output_delay(unsigned char *mac_address, int port_index, int output_delay);
```

Description

*set_io_output_delay*는 *port_index*로 지정된 출력포트의 [출력지연(밀리초)]을 변경합니다.

Parameters

*unsigned char *mac_address*

EzTCP_Search 또는 *EzTCP_Read*를 사용하여 검색된 제품(ezTCP)의 MAC주소 입니다.

int port_index

출력포트의 순번으로 0부터 시작합니다.

int output_delay

*port_index*로 지정된 출력포트의 출력지연(밀리초)입니다.

Return values

*set_io_output_delay*의 반환 값은 다음과 같습니다.

1

설정 값 변경에 성공하면 1을 반환합니다.

-1

*mac_address*을 가진 제품(ezTCP)을 찾을 수 없는 경우에는 -1을 반환합니다.

-2

*is_io*가 1을 반환하지 않는 경우 -2를 반환합니다.

-3

*port_index*에 유효하지 않은 값을 입력한 경우 -3를 반환합니다.

Examples

See also

is_io, *set_io_http*, *set_io_http_port*, *set_io_html_size*, *set_io_modbus*, *set_io_input_notification*,
set_io_output_automatic_initialize, *set_io_modbus_type*, *set_io_unit_id*, *set_io_input_address*,
set_io_output_address, *set_io_poll_interval*, *set_io_slave_control_type*, *set_io_master_control_type*,
set_io_tcp_type, *set_io_multi_connection_number*, *set_io_modbus_peer_address*,
set_io_modbus_peer_port, *set_io_modbus_local_port*, *set_io_event_notification*,
set_event_notification_email, *set_io_event_notification_port*, *set_io_input_valid_time*, *set_io_macro*,
set_io_port_macro, *set_io_macro_text*, *set_io_output_delay*, *set_io_output_initial_state*, *set_io_comment*.

Remarks

5.5.27 set_io_output_initial_state

```
int set_io_output_initial_state(unsigned char *mac_address, int port_index, int onoff);
```

Description

*set_io_output_initial_state*는 *port_index*로 지정된 출력포트의 [초기상태]를 변경합니다.

Parameters

*unsigned char *mac_address*

EzTCP_Search 또는 *EzTCP_Read*를 사용하여 검색된 제품(ezTCP)의 MAC주소 입니다.

int port_index

출력포트의 순번으로 0부터 시작합니다.

int onoff

해당 기능의 선택 여부를 결정하며 아래와 같은 값을 가질 수 있습니다.

0

기능 선택을 해제합니다.

1

기능을 선택합니다.

Return values

*set_io_output_initial_state*의 반환 값은 다음과 같습니다.

1

설정 값 변경에 성공하면 1을 반환합니다.

-1

*mac_address*을 가진 제품(ezTCP)을 찾을 수 없는 경우에는 -1을 반환합니다.

-2

*is_io*가 1을 반환하지 않는 경우 -2를 반환합니다.

-3

*port_index*에 유효하지 않은 값을 입력한 경우 -3를 반환합니다.

Examples

See also

is_io, *set_io_http*, *set_io_http_port*, *set_io_html_size*, *set_io_modbus*, *set_io_input_notification*, *set_io_output_automatic_initialize*, *set_io_modbus_type*, *set_io_unit_id*, *set_io_input_address*, *set_io_output_address*, *set_io_poll_interval*, *set_io_slave_control_type*, *set_io_master_control_type*, *set_io_tcp_type*, *set_io_multi_connection_number*, *set_io_modbus_peer_address*, *set_io_modbus_peer_port*, *set_io_modbus_local_port*, *set_io_event_notification*, *set_event_notification_email*, *set_io_event_notification_port*, *set_io_input_valid_time*, *set_io_macro*, *set_io_port_macro*, *set_io_macro_text*, *set_io_output_delay*, *set_io_output_initial_state*, *set_io_comment*.

Remarks

5.5.28 set_io_comment

```
int set_io_comment(unsigned char *mac_address, int port_flag, int port_index,
                  char *comment);
```

Description

*set_io_comment*는 I/O 제품에 설치된 입력 또는 출력포트의 [I/O 포트 설명]을 변경합니다.

Parameters

*unsigned char *mac_address*



EzTCP_Search 또는 *EzTCP_Read*를 사용하여 검색된 제품(ezTCP)의 MAC주소 입니다.

int port_flag

*port_flag*가 0이면 입력포트를 의미하고 1이면 출력포트를 의미합니다.

int port_index

입력 또는 출력포트의 순번으로 0부터 시작합니다.

*char *comment*

*port_index*로 지정된 입력 또는 출력포트의 I/O 포트 설명이 저장된 포인터입니다.

Return values

*set_io_comment*의 반환 값은 다음과 같습니다.

1

설정 값 변경에 성공하면 1을 반환합니다.

-1

*mac_address*을 가진 제품(ezTCP)을 찾을 수 없는 경우에는 -1을 반환합니다.

-2

*is_io*가 1을 반환하지 않는 경우 -2를 반환합니다.

-3

*port_index*에 유효하지 않은 값을 입력한 경우 -3를 반환합니다.

Examples

See also

is_io, *set_io_http*, *set_io_http_port*, *set_io_html_size*, *set_io_modbus*, *set_io_input_notification*, *set_io_output_automatic_initialize*, *set_io_modbus_type*, *set_io_unit_id*, *set_io_input_address*, *set_io_output_address*, *set_io_poll_interval*, *set_io_slave_control_type*, *set_io_master_control_type*, *set_io_tcp_type*, *set_io_multi_connection_number*, *set_io_modbus_peer_address*, *set_io_modbus_peer_port*, *set_io_modbus_local_port*, *set_io_event_notification*, *set_event_notification_email*, *set_io_event_notification_port*, *set_io_input_valid_time*, *set_io_macro*, *set_io_port_macro*, *set_io_macro_text*, *set_io_output_delay*, *set_io_output_initial_state*, *set_io_comment*.

Remarks

[I/O 포트 설명]은 NULL을 제외하고 최대 16바이트입니다. [I/O 포트 설명]길이는 *IO_COMMENT_LEN*에 정의되어 있습니다.

5.6 무선랜

5.6.1 set_wlan_type

```
int set_wlan_type(unsigned char *mac_address, int wlan_type);
```

Description

*set_wlan_type*은 무선랜 제품의 [무선랜 종류]를 변경합니다.

Parameters

*unsigned char *mac_address*

EzTCP_Search 또는 *EzTCP_Read*를 사용하여 검색된 제품(ezTCP)의 MAC주소 입니다.

int wlan_type

[무선랜 종류]를 나타내며 아래와 같은 값을 가질 수 있습니다.

0	애드혹
1	인프라스트럭처
2	Soft AP

Return values

*set_wlan_type*의 반환 값은 다음과 같습니다.

- 1
설정 값 변경에 성공하면 1을 반환합니다.
- 1
*mac_address*을 가진 제품(ezTCP)을 찾을 수 없는 경우에는 -1을 반환합니다.
- 2
*is_wlan*이 1을 반환하지 않는 경우 -2를 반환합니다.
- 3
*wlan_type*에 유효하지 않은 값을 입력한 경우 -3를 반환합니다.
- 4
*wlan_type*에 Soft AP(2)를 입력한 경우 *is_wlan_soft_ap*가 1을 반환하지 않으면 -4를 반환합니다.

Examples

See also

is_wlan, *is_wlan_soft_ap*, *set_wlan_type*, *set_wlan_channel*, *set_wlan_ssid*, *set_wlan_antenna*, *set_wlan_phy_mode*, *set_wlan_short_preamble*, *set_wlan_short_slot*, *set_wlan_cts_protection*, *set_wlan_background_scan*, *set_wlan_encryption_type*, *set_wlan_authentication_type*, *set_wlan_wep_key_length*, *set_wlan_wep_key_index*, *set_wlan_wep_key_data_type*, *set_wlan_wep_key*, *set_wlan_wpa_passphrase*, *set_wlan_shared_key*, *set_wlan_wpa_enterprise*, *set_wlan_wpa_enterprise_id*, *set_wlan_wpa_enterprise_pwd*.

Remarks

5.6.2 set_wlan_channel

```
int set_wlan_channel(unsigned char *mac_address, int channel);
```

Description

*set_wlan_channel*은 무선랜 제품의 [채널]을 변경합니다.

Parameters

*unsigned char *mac_address*

EzTCP_Search 또는 *EzTCP_Read*를 사용하여 검색된 제품(ezTCP)의 MAC주소 입니다.

int channel

변경할 무선랜 채널번호입니다.

Return values

*set_wlan_channel*의 반환 값은 다음과 같습니다.

1

설정 값 변경에 성공하면 1을 반환합니다.

-1

*mac_address*을 가진 제품(ezTCP)을 찾을 수 없는 경우에는 -1을 반환합니다.

-2

*is_wlan*이 1을 반환하지 않는 경우 -2를 반환합니다.

Examples

See also

is_wlan, *set_wlan_type*, *set_wlan_channel*, *set_wlan_ssid*, *set_wlan_antenna*, *set_wlan_phy_mode*,
set_wlan_short_preamble, *set_wlan_short_slot*, *set_wlan_cts_protection*, *set_wlan_background_scan*,
set_wlan_encryption_type, *set_wlan_authentication_type*, *set_wlan_wep_key_length*,
set_wlan_wep_key_index, *set_wlan_wep_key_data_type*, *set_wlan_wep_key*, *set_wlan_wpa_passphrase*,
set_wlan_shared_key, *set_wlan_wpa_enterprise*, *set_wlan_wpa_enterprise_id*,
set_wlan_wpa_enterprise_pwd.

Remarks

무선랜 제품의 [채널]은 [무선랜 종류]가 애드혹 또는 Soft-AP인 경우에만 사용됩니다.

5.6.3 set_wlan_ssid

```
int set_wlan_ssid(unsigned char *mac_address, char *ssid);
```

Description



*set_wlan_ssid*는 무선랜 제품의 [SSID]를 변경합니다.

Parameters

*unsigned char *mac_address*

EzTCP_Search 또는 *EzTCP_Read*를 사용하여 검색된 제품(ezTCP)의 MAC주소 입니다.

*char *ssid*

무선랜 제품의 SSID가 저장되어 있는 포인터입니다.

Return values

*set_wlan_ssid*의 반환 값은 다음과 같습니다.

1

설정 값 변경에 성공하면 1을 반환합니다.

-1

*mac_address*을 가진 제품(ezTCP)을 찾을 수 없는 경우에는 -1을 반환합니다.

-2

*is_wlan*이 1을 반환하지 않는 경우 -2를 반환합니다.

Examples

See also

is_wlan, *set_wlan_type*, *set_wlan_channel*, *set_wlan_ssid*, *set_wlan_antenna*, *set_wlan_phy_mode*,
set_wlan_short_preamble, *set_wlan_short_slot*, *set_wlan_cts_protection*, *set_wlan_background_scan*,
set_wlan_encryption_type, *set_wlan_authentication_type*, *set_wlan_wep_key_length*,
set_wlan_wep_key_index, *set_wlan_wep_key_data_type*, *set_wlan_wep_key*, *set_wlan_wpa_passphrase*,
set_wlan_shared_key, *set_wlan_wpa_enterprise*, *set_wlan_wpa_enterprise_id*,
set_wlan_wpa_enterprise_pwd.

Remarks

[SSID]는 NULL을 제외하고 최대 32바이트입니다.

5.6.4 set_wlan_antenna

```
int set_wlan_antenna(unsigned char *mac_address, int antenna_type);
```

Description

*set_wlan_antenna*는 무선랜 제품의 [안테나]를 변경합니다.

Parameters

*unsigned char *mac_address*

EzTCP_Search 또는 *EzTCP_Read*를 사용하여 검색된 제품(ezTCP)의 MAC주소 입니다.

int antenna_type

무선랜 제품의 [안테나] 설정 값을 나타내며 아래와 같은 값을 가질 수 있습니다.

0

내장안테나

1

외장안테나

Return values

*set_wlan_antenna*의 반환 값은 다음과 같습니다.

1

설정 값 변경에 성공하면 1을 반환합니다.

-1

*mac_address*을 가진 제품(ezTCP)을 찾을 수 없는 경우에는 -1을 반환합니다.

-2

*is_wlan*이 1을 반환하지 않는 경우 -2를 반환합니다.

-3

*is_wlan_antenna*가 1을 반환하지 않는 경우 -3을 반환합니다.

Examples

See also

is_wlan, *set_wlan_type*, *set_wlan_channel*, *set_wlan_ssid*, *is_wlan_antenna*, *set_wlan_antenna*, *set_wlan_phy_mode*, *set_wlan_short_preamble*, *set_wlan_short_slot*, *set_wlan_cts_protection*, *set_wlan_background_scan*, *set_wlan_encryption_type*, *set_wlan_authentication_type*, *set_wlan_wep_key_length*, *set_wlan_wep_key_index*, *set_wlan_wep_key_data_type*, *set_wlan_wep_key*, *set_wlan_wpa_passphrase*, *set_wlan_shared_key*, *set_wlan_wpa_enterprise*, *set_wlan_wpa_enterprise_id*, *set_wlan_wpa_enterprise_pwd*.

Remarks

보다 자세한 내용은 해당 제품 사용설명서를 참조하시기 바랍니다.

5.6.5 set_wlan_phy_mode

```
int set_wlan_phy_mode(unsigned char *mac_address, int phy_mode);
```

Description

*set_wlan_phy_mode*는 무선랜 제품의 [무선 고급설정]들 중에서 [Phy Mode]를 변경합니다.

Parameters

*unsigned char *mac_address*

EzTCP_Search 또는 *EzTCP_Read*를 사용하여 검색된 제품(ezTCP)의 MAC주소 입니다.

int phy_mode

무선랜 제품의 [Phy Mode] 설정 값을 나타내며 아래와 같은 값을 가질 수 있습니다.

1	802.11
2	802.11b
3	802.11b/g

Return values

*set_wlan_phy_mode*의 반환 값은 다음과 같습니다.

- 1
설정 값 변경에 성공하면 1을 반환합니다.
- 1
*mac_address*을 가진 제품(ezTCP)을 찾을 수 없는 경우에는 -1을 반환합니다.
- 2
*is_wlan*이 1을 반환하지 않는 경우 -2를 반환합니다.
- 3
*is_wlan_phy_mode*가 1을 반환하지 않는 경우 -3를 반환합니다.
- 4
*phy_mode*에 유효하지 않은 값을 입력한 경우 -4를 반환합니다.

Examples

See also

is_wlan, *set_wlan_type*, *set_wlan_channel*, *set_wlan_ssid*, *set_wlan_antenna*, *is_wlan_phy_mode*, *set_wlan_phy_mode*, *set_wlan_short_preamble*, *set_wlan_short_slot*, *set_wlan_cts_protection*, *set_wlan_background_scan*, *set_wlan_encryption_type*, *set_wlan_authentication_type*, *set_wlan_wep_key_length*, *set_wlan_wep_key_index*, *set_wlan_wep_key_data_type*, *set_wlan_wep_key*, *set_wlan_wpa_passphrase*, *set_wlan_shared_key*, *set_wlan_wpa_enterprise*, *set_wlan_wpa_enterprise_id*, *set_wlan_wpa_enterprise_pwd*.

Remarks

보다 자세한 내용은 해당 제품 사용설명서를 참조하시기 바랍니다.

5.6.6 set_wlan_short_preamble

```
int set_wlan_short_preamble(unsigned char *mac_address, int onoff);
```

Description

*set_wlan_short_preamble*은 무선랜 제품의 [무선 고급설정]들 중에서 [Short Preamble] 선택 여부를 변경합니다.

Parameters

*unsigned char *mac_address*

EzTCP_Search 또는 *EzTCP_Read*를 사용하여 검색된 제품(ezTCP)의 MAC주소 입니다.

int onoff

해당 기능의 선택 여부를 결정하며 아래와 같은 값을 가질 수 있습니다.

0

기능 선택을 해제합니다.

1

기능을 선택합니다.

Return values

*set_wlan_short_preamble*의 반환 값은 다음과 같습니다.

1

설정 값 변경에 성공하면 1을 반환합니다.

-1

*mac_address*을 가진 제품(ezTCP)을 찾을 수 없는 경우에는 -1을 반환합니다.

-2

*is_wlan*이 1을 반환하지 않는 경우 -2를 반환합니다.

-3

*is_wlan_phy_mode*가 1을 반환하지 않는 경우 -3를 반환합니다.

Examples

See also

is_wlan, *set_wlan_type*, *set_wlan_channel*, *set_wlan_ssid*, *set_wlan_antenna*, *is_wlan_phy_mode*, *set_wlan_phy_mode*, *set_wlan_short_preamble*, *set_wlan_short_slot*, *set_wlan_cts_protection*, *set_wlan_background_scan*, *set_wlan_encryption_type*, *set_wlan_authentication_type*, *set_wlan_wep_key_length*, *set_wlan_wep_key_index*, *set_wlan_wep_key_data_type*, *set_wlan_wep_key*, *set_wlan_wpa_passphrase*, *set_wlan_shared_key*, *set_wlan_wpa_enterprise*, *set_wlan_wpa_enterprise_id*, *set_wlan_wpa_enterprise_pwd*.

Remarks



[무선 고급설정]들 중에서 [Phy Mode]가 2(802.11b) 또는 3(802.11b/g)인 경우에만 사용됩니다. 보다 자세한 내용은 해당 제품 사용설명서를 참조하시기 바랍니다.

5.6.7 set_wlan_short_slot

```
int set_wlan_short_slot(unsigned char *mac_address, int onoff);
```

Description

*set_wlan_short_slot*은 무선랜 제품의 [무선 고급설정]들 중에서 [Short Slot] 선택 여부를 변경합니다.

Parameters

*unsigned char *mac_address*

EzTCP_Search 또는 *EzTCP_Read*를 사용하여 검색된 제품(ezTCP)의 MAC주소 입니다.

int onoff

해당 기능의 선택 여부를 결정하며 아래와 같은 값을 가질 수 있습니다.

0

기능 선택을 해제합니다.

1

기능을 선택합니다.

Return values

*set_wlan_short_slot*의 반환 값은 다음과 같습니다.

1

설정 값 변경에 성공하면 1을 반환합니다.

-1

*mac_address*을 가진 제품(ezTCP)을 찾을 수 없는 경우에는 -1을 반환합니다.

-2

*is_wlan*이 1을 반환하지 않는 경우 -2를 반환합니다.

-3

*is_wlan_phy_mode*가 1을 반환하지 않는 경우 -3를 반환합니다.

Examples

See also

is_wlan, *set_wlan_type*, *set_wlan_channel*, *set_wlan_ssid*, *set_wlan_antenna*, *is_wlan_phy_mode*, *set_wlan_phy_mode*, *set_wlan_short_preamble*, *set_wlan_short_slot*, *set_wlan_cts_protection*, *set_wlan_background_scan*, *set_wlan_encryption_type*, *set_wlan_authentication_type*, *set_wlan_wep_key_length*, *set_wlan_wep_key_index*, *set_wlan_wep_key_data_type*, *set_wlan_wep_key*, *set_wlan_wpa_passphrase*, *set_wlan_shared_key*, *set_wlan_wpa_enterprise*, *set_wlan_wpa_enterprise_id*,

set_wlan_wpa_enterprise_pwd.

Remarks

[무선 고급설정]들 중에서 [Phy Mode]가 3(802.11b/g)인 경우에만 사용할 수 있습니다. 보다 자세한 내용은 해당 제품 사용설명서를 참조하시기 바랍니다.

5.6.8 set_wlan_cts_protection

```
int set_wlan_cts_protection(unsigned char *mac_address, int onoff);
```

Description

*set_wlan_cts_protection*은 무선랜 제품의 [무선 고급설정]들 중에서 [CTS Protection] 선택 여부를 변경합니다.

Parameters

*unsigned char *mac_address*

EzTCP_Search 또는 *EzTCP_Read*를 사용하여 검색된 제품(ezTCP)의 MAC주소 입니다.

int onoff

해당 기능의 선택 여부를 결정하며 아래와 같은 값을 가질 수 있습니다.

0

기능 선택을 해제합니다.

1

기능을 선택합니다.

Return values

*set_wlan_cts_protection*의 반환 값은 다음과 같습니다.

1

설정 값 변경에 성공하면 1을 반환합니다.

-1

*mac_address*을 가진 제품(ezTCP)을 찾을 수 없는 경우에는 -1을 반환합니다.

-2

*is_wlan*이 1을 반환하지 않는 경우 -2를 반환합니다.

-3

*is_wlan_phy_mode*가 1을 반환하지 않는 경우 -3를 반환합니다.

Examples

See also

is_wlan, *set_wlan_type*, *set_wlan_channel*, *set_wlan_ssid*, *set_wlan_antenna*, *is_wlan_phy_mode*,

set_wlan_phy_mode, set_wlan_short_preamble, set_wlan_short_slot, set_wlan_cts_protection,
 set_wlan_background_scan, set_wlan_encryption_type, set_wlan_authentication_type,
 set_wlan_wep_key_length, set_wlan_wep_key_index, set_wlan_wep_key_data_type, set_wlan_wep_key,
 set_wlan_wpa_passphrase, set_wlan_shared_key, set_wlan_wpa_enterprise, set_wlan_wpa_enterprise_id,
 set_wlan_wpa_enterprise_pwd.

Remarks

[무선 고급설정]들 중에서 [Phy Mode]가 3(802.11b/g)인 경우에만 사용할 수 있습니다. 보다 자세한 내용은 해당 제품 사용설명서를 참조하시기 바랍니다.

5.6.9 set_wlan_background_scan

```
int set_wlan_background_scan(unsigned char *mac_address, int onoff);
```

Description

*set_wlan_background_scan*은 무선랜 제품의 [무선 고급설정]들 중에서 [Background Scan] 선택 여부를 변경합니다.

Parameters

*unsigned char *mac_address*

EzTCP_Search 또는 *EzTCP_Read*를 사용하여 검색된 제품(ezTCP)의 MAC주소 입니다.

int onoff

해당 기능의 선택 여부를 결정하며 아래와 같은 값을 가질 수 있습니다.

0

기능 선택을 해제합니다.

1

기능을 선택합니다.

Return values

*set_wlan_background_scan*의 반환 값은 다음과 같습니다.

1

설정 값 변경에 성공하면 1을 반환합니다.

-1

*mac_address*을 가진 제품(ezTCP)을 찾을 수 없는 경우에는 -1을 반환합니다.

-2

*is_wlan*이 1을 반환하지 않는 경우 -2를 반환합니다.

-3

*is_wlan_background_scan*이 1을 반환하지 않는 경우 -3를 반환합니다.

Examples



See also

is_wlan, set_wlan_type, set_wlan_channel, set_wlan_ssid, set_wlan_antenna, set_wlan_phy_mode, set_wlan_short_preamble, set_wlan_short_slot, set_wlan_cts_protection, is_wlan_background_scan, set_wlan_background_scan, set_wlan_encryption_type, set_wlan_authentication_type, set_wlan_wep_key_length, set_wlan_wep_key_index, set_wlan_wep_key_data_type, set_wlan_wep_key, set_wlan_wpa_passphrase, set_wlan_shared_key, set_wlan_wpa_enterprise, set_wlan_wpa_enterprise_id, set_wlan_wpa_enterprise_pwd.

Remarks

보다 자세한 내용은 해당 제품 사용설명서를 참조하시기 바랍니다.

5.6.10 set_wlan_encryption_type

```
int set_wlan_encryption_type(unsigned char *mac_address, int encryption_type);
```

Description

*set_wlan_encryption_type*은 무선랜 제품의 [암호화 방식]을 변경합니다.

Parameters

*unsigned char *mac_address*

EzTCP_Search 또는 *EzTCP_Read*를 사용하여 검색된 제품(ezTCP)의 MAC주소 입니다.

int encryption_type

무선랜 제품의 [암호화 방식]을 나타내며 아래와 같은 값을 가질 수 있습니다.

0	없음
1	WEP
2	WPA

Return values

*set_wlan_encryption_type*의 반환 값은 다음과 같습니다.

1	설정 값 변경에 성공하면 1을 반환합니다.
-1	<i>mac_address</i> 을 가진 제품(ezTCP)을 찾을 수 없는 경우에는 -1을 반환합니다.
-2	<i>is_wlan</i> 이 1을 반환하지 않는 경우 -2를 반환합니다.
-3	

*is_wlan_rsn*이 0을 반환하지 않는 경우 -3을 반환합니다.

-4

*encryption_type*에 유효하지 않은 값을 입력한 경우 -4를 반환합니다.

-5

[무선랜 종류]가 애드훅(0)인 경우 *encryption_type*에 WPA(2)를 입력하면 -5를 반환합니다.

Examples

See also

is_wlan, *set_wlan_type*, *set_wlan_channel*, *set_wlan_ssid*, *set_wlan_antenna*, *set_wlan_phy_mode*, *set_wlan_short_preamble*, *set_wlan_short_slot*, *set_wlan_cts_protection*, *set_wlan_background_scan*, *is_wlan_rsn*, *set_wlan_encryption_type*, *set_wlan_authentication_type*, *set_wlan_wep_key_length*, *set_wlan_wep_key_index*, *set_wlan_wep_key_data_type*, *set_wlan_wep_key*, *set_wlan_wpa_passphrase*, *set_wlan_shared_key*, *set_wlan_wpa_enterprise*, *set_wlan_wpa_enterprise_id*, *set_wlan_wpa_enterprise_pwd*.

Remarks

5.6.11 *set_wlan_authentication_type*

```
int set_wlan_authentication_type(unsigned char *mac_address, int authentication_type);
```

Description

*set_wlan_authentication_type*은 무선랜 제품의 [인증 방식] 또는 WPA [인증 방식/암호화 방법]을 변경합니다.

Parameters

*unsigned char *mac_address*

EzTCP_Search 또는 *EzTCP_Read*를 사용하여 검색된 제품(ezTCP)의 MAC주소 입니다.

int encryption_type

무선랜 제품의 [인증 방식] 또는 WPA [인증 방식/암호화 방법]을 나타내며 *get_wlan_encryption_type*의 반환 값에 따라서 *authentication_type*에 입력 가능한 값과 의미는 달라집니다.

*get_wlan_encryption_type*이 0(없음)을 반환하는 경우에는 아래와 같은 값을 가질 수 있습니다.

1

개방 모드

*get_wlan_encryption_type*이 1(WEP)을 반환하는 경우에는 아래와 같은 값을 가질 수 있습니다.

- 1 개방 모드
- 2 공유 모드
- 3 자동 모드

*get_wlan_encryption_type*이 2(WPA)을 반환하는 경우에는 아래와 같은 값을 가질 수 있습니다.

- 0 WPA PSK – TKIP
- 1 WPA PSK – AES
- 2 WPA PSK – TKIP/AES
- 3 WPA2 PSK – TKIP
- 4 WPA2 PSK – AES
- 5 WPA2 PSK- TKIP/AES

Return values

*set_wlan_authentication_type*의 반환 값은 다음과 같습니다.

- 1 설정 값 변경에 성공하면 1을 반환합니다.
 - 1 *mac_address*을 가진 제품(ezTCP)을 찾을 수 없는 경우에는 -1을 반환합니다.
 - 2 *is_wlan*이 1을 반환하지 않는 경우 -2를 반환합니다.
 - 3 *is_wlan_rsn*이 0을 반환하지 않는 경우 -3을 반환합니다.
 - 4 무선랜 제품의 [암호화 방식]이 0(없음) 또는 1(WEP)일 때 *authentication_type*에 유효하지 않은 값을 입력한 경우 -4를 반환합니다.
 - 5 무선랜 제품의 [암호화 방식]이 2(WPA)일 때 *authentication_type*에 유효하지 않은 값을 입력한 경우 -5를 반환합니다.
- is_wlan_authentication_type*이 0을 반환하면 *authentication_type*에 0(WPA PSK – TKIP)

또는 4(WPA2 PSK – AES)만 입력할 수 있습니다.

Examples

See also

is_wlan, set_wlan_type, set_wlan_channel, set_wlan_ssid, set_wlan_antenna, set_wlan_phy_mode, set_wlan_short_preamble, set_wlan_short_slot, set_wlan_cts_protection, set_wlan_background_scan, is_wlan_rsn, set_wlan_encryption_type, is_wlan_authentication_type, set_wlan_authentication_type, set_wlan_wep_key_length, set_wlan_wep_key_index, set_wlan_wep_key_data_type, set_wlan_wep_key, set_wlan_wpa_passphrase, set_wlan_shared_key, set_wlan_wpa_enterprise, set_wlan_wpa_enterprise_id, set_wlan_wpa_enterprise_pwd.

Remarks

5.6.12 set_wlan_wep_key_length

```
int set_wlan_wep_key_length(unsigned char *mac_address, int wep_key_length);
```

Description

*set_wlan_wep_key_length*은 무선랜 제품의 WEP 키 길이를 변경합니다.

Parameters

*unsigned char *mac_address*

EzTCP_Search 또는 *EzTCP_Read*를 사용하여 검색된 제품(ezTCP)의 MAC주소 입니다.

int wep_key_length

무선랜 제품의 WEP 키 길이를 나타내며 아래와 같은 값을 가질 수 있습니다.

- | | |
|---|-------|
| 1 | 64비트 |
| 2 | 128비트 |

Return values

*set_wlan_wep_key_length*의 반환 값은 다음과 같습니다.

- | | |
|----|--|
| 1 | 설정 값 변경에 성공하면 1을 반환합니다. |
| -1 | <i>mac_address</i> 을 가진 제품(ezTCP)을 찾을 수 없는 경우에는 -1을 반환합니다. |
| -2 | <i>is_wlan</i> 이 1을 반환하지 않는 경우 -2를 반환합니다. |

-3

*wep_key_length*에 유효하지 않은 값을 입력한 경우 -3를 반환합니다.

Examples

See also

is_wlan, *set_wlan_type*, *set_wlan_channel*, *set_wlan_ssid*, *set_wlan_antenna*, *set_wlan_phy_mode*,
set_wlan_short_preamble, *set_wlan_short_slot*, *set_wlan_cts_protection*, *set_wlan_background_scan*,
set_wlan_encryption_type, *set_wlan_authentication_type*, *set_wlan_wep_key_length*,
set_wlan_wep_key_index, *set_wlan_wep_key_data_type*, *set_wlan_wep_key*, *set_wlan_wpa_passphrase*,
set_wlan_shared_key, *set_wlan_wpa_enterprise*, *set_wlan_wpa_enterprise_id*,
set_wlan_wpa_enterprise_pwd.

Remarks

5.6.13 *set_wlan_wep_key_index*

```
int set_wlan_wep_key_index(unsigned char *mac_address, int wep_key_index);
```

Description

*set_wlan_wep_key_index*은 무선랜 제품이 사용하는 WEP 키 순번을 변경합니다.

Parameters

*unsigned char *mac_address*

EzTCP_Search 또는 *EzTCP_Read*를 사용하여 검색된 제품(ezTCP)의 MAC주소 입니다.

int wep_key_index

변경할 WEP 키 순번으로 0~3입니다.

Return values

*set_wlan_wep_key_index*의 반환 값은 다음과 같습니다.

1

설정 값 변경에 성공하면 1을 반환합니다.

-1

*mac_address*을 가진 제품(ezTCP)을 찾을 수 없는 경우에는 -1을 반환합니다.

-2

*is_wlan*이 1을 반환하지 않는 경우 -2를 반환합니다.

-3

*wep_key_index*에 유효하지 않은 값을 입력한 경우 -3를 반환합니다.

Examples

See also

is_wlan, set_wlan_type, set_wlan_channel, set_wlan_ssid, set_wlan_antenna, set_wlan_phy_mode, set_wlan_short_preamble, set_wlan_short_slot, set_wlan_cts_protection, set_wlan_background_scan, set_wlan_encryption_type, set_wlan_authentication_type, set_wlan_wep_key_length, set_wlan_wep_key_index, set_wlan_wep_key_data_type, set_wlan_wep_key, set_wlan_wpa_passphrase, set_wlan_shared_key, set_wlan_wpa_enterprise, set_wlan_wpa_enterprise_id, set_wlan_wpa_enterprise_pwd.

Remarks

5.6.14 set_wlan_wep_key_data_type

```
int set_wlan_wep_key_data_type(unsigned char *mac_address, int wep_key_data_type);
```

Description

*set_wlan_wep_key_data_type*은 WEP 키가 저장될 데이터 형식을 변경합니다.

Parameters

*unsigned char *mac_address*

EzTCP_Search 또는 *EzTCP_Read*를 사용하여 검색된 제품(ezTCP)의 MAC주소 입니다.

int wep_key_data_type

WEP 키가 저장될 데이터 형식을 나타내며 아래와 같은 값을 가질 수 있습니다.

0
16진수
1
ASCII

Return values

*set_wlan_wep_key_data_type*의 반환 값은 다음과 같습니다.

1
설정 값 변경에 성공하면 1을 반환합니다.
-1
*mac_address*을 가진 제품(ezTCP)을 찾을 수 없는 경우에는 -1을 반환합니다.
-2
*is_wlan*이 1을 반환하지 않는 경우 -2를 반환합니다.
-3
*wep_key_data_type*에 유효하지 않은 값을 입력한 경우 -3을 반환합니다.

Examples

See also

is_wlan, set_wlan_type, set_wlan_channel, set_wlan_ssid, set_wlan_antenna, set_wlan_phy_mode,
 set_wlan_short_preamble, set_wlan_short_slot, set_wlan_cts_protection, set_wlan_background_scan,
 set_wlan_encryption_type, set_wlan_authentication_type, set_wlan_wep_key_length,
 set_wlan_wep_key_index, set_wlan_wep_key_data_type, set_wlan_wep_key, set_wlan_wpa_passphrase,
 set_wlan_shared_key, set_wlan_wpa_enterprise, set_wlan_wpa_enterprise_id,
 set_wlan_wpa_enterprise_pwd.

Remarks

5.6.15 set_wlan_wep_key

```
int set_wlan_wep_key(unsigned char *mac_address, char *wep_key);
```

Description

*set_wlan_wep_key*는 무선랜 제품이 사용할 WEP 키를 변경합니다.

Parameters

*unsigned char *mac_address*

EzTCP_Search 또는 *EzTCP_Read*를 사용하여 검색된 제품(ezTCP)의 MAC주소 입니다.

*char *wep_key*

무선랜 제품이 사용할 WEP 키가 저장되어 있는 포인터입니다.

Return values

*set_wlan_wep_key*의 반환 값은 다음과 같습니다.

1

설정 값 변경에 성공하면 1을 반환합니다.

-1

*mac_address*을 가진 제품(ezTCP)을 찾을 수 없는 경우에는 -1을 반환합니다.

-2

*is_wlan*이 1을 반환하지 않는 경우 -2를 반환합니다.

-3

*wep_key*에 입력한 WEP 키 문자열 길이가 유효하지 않은 경우 -3을 반환합니다.

-4

*wep_key*에 입력한 WEP 키 문자열 내용이 유효하지 않은 경우 -4를 반환합니다.

Examples

```

struct eztcp_info out_eztcp_info;
int ret = get_eztcp_info_by_index(0, &out_eztcp_info);

if(ret == EZTCP_SUCCESS)
{
    unsigned char *mac_address = out_eztcp_info.mac_address;

    int wep_key_length = 1; // 64bit
    int wep_key_index = 1; // index #1
    int wep_key_data_type = 1; // ASCII
    char *wep_key = "12345";

    if(is_wlan(mac_address) == 1)
    {
        set_wlan_wep_key_length(mac_address, wep_key_length);
        set_wlan_wep_key_index(mac_address, wep_key_index);
        set_wlan_wep_key_data_type(mac_address, wep_key_data_type);
        set_wlan_wep_key(mac_address, wep_key);
    }
}

```

See also

is_wlan, set_wlan_type, set_wlan_channel, set_wlan_ssid, set_wlan_antenna, set_wlan_phy_mode, set_wlan_short_preamble, set_wlan_short_slot, set_wlan_cts_protection, set_wlan_background_scan, set_wlan_encryption_type, set_wlan_authentication_type, set_wlan_wep_key_length, set_wlan_wep_key_index, set_wlan_wep_key_data_type, set_wlan_wep_key, set_wlan_wpa_passphrase, set_wlan_shared_key, set_wlan_wpa_enterprise, set_wlan_wpa_enterprise_id, set_wlan_wpa_enterprise_pwd.

Remarks

WEP 키 길이가 64비트이면 *wep_key*에 10자리의 16진수 코드(0~9, A~F, a~f) 또는 5자리의 ASCII코드를 입력해야합니다.

WEP 키 길이가 128비트이면 *wep_key*에 26자리의 16진수 코드(0~9, A~F, a~f) 또는 13자리의 ASCII코드를 입력해야합니다.

5.6.16 set_wlan_wpa_passphrase

```
int set_wlan_wpa_passphrase(unsigned char *mac_address, char *wpa_pp);
```

Description

*set_wlan_wpa_passphrase*는 무선랜 제품의 WPA 암호문을 변경합니다.

Parameters*unsigned char *mac_address**EzTCP_Search* 또는 *EzTCP_Read*를 사용하여 검색된 제품(ezTCP)의 MAC주소 입니다.*char *wpa_pp*

무선랜 제품이 사용할 WPA 암호문이 저장되어 있는 포인터입니다.

Return values*set_wlan_wpa_passphrase*의 반환 값은 다음과 같습니다.

1

설정 값 변경에 성공하면 1을 반환합니다.

-1

*mac_address*을 가진 제품(ezTCP)을 찾을 수 없는 경우에는 -1을 반환합니다.

-2

*is_wlan*이 1을 반환하지 않는 경우 -2를 반환합니다.

-3

*is_wlan_rsn*이 0을 반환하지 않는 경우 -3을 반환합니다.**Examples****See also**

is_wlan, *set_wlan_type*, *set_wlan_channel*, *set_wlan_ssid*, *set_wlan_antenna*, *set_wlan_phy_mode*,
set_wlan_short_preamble, *set_wlan_short_slot*, *set_wlan_cts_protection*, *set_wlan_background_scan*,
set_wlan_encryption_type, *set_wlan_authentication_type*, *set_wlan_wep_key_length*,
set_wlan_wep_key_index, *set_wlan_wep_key_data_type*, *set_wlan_wep_key*, *set_wlan_wpa_passphrase*,
set_wlan_shared_key, *set_wlan_wpa_enterprise*, *set_wlan_wpa_enterprise_id*,
set_wlan_wpa_enterprise_pwd.

Remarks

WPA 암호문은 NULL을 포함해서 최대 64바이트입니다.

5.6.17 set_wlan_shared_key

```
int set_wlan_shared_key(unsigned char *mac_address, char *shared_key);
```

Description*set_wlan_shared_key*는 무선랜 제품의 보안 설정 중에서 [Shared Key]를 변경합니다.**Parameters***unsigned char *mac_address*

EzTCP_Search 또는 *EzTCP_Read*를 사용하여 검색된 제품(ezTCP)의 MAC주소 입니다.
`char *shared_key`

Shared Key값이 저장되어 있는 포인터입니다.

Return values

`set_wlan_shared_key`의 반환 값은 다음과 같습니다.

1

설정 값 변경에 성공하면 1을 반환합니다.

-1

`mac_address`을 가진 제품(ezTCP)을 찾을 수 없는 경우에는 -1을 반환합니다.

-2

`is_wlan`이 1을 반환하지 않는 경우 -2를 반환합니다.

-3

`is_wlan_rsn`이 1을 반환하지 않는 경우 -3을 반환합니다.

-4

[무선랜 종류]가 애드혹(0) 또는 Soft AP(2)인 경우 입력한 Shared Key값을 WEP 키 값으로 사용할 수 없으면 -4를 반환합니다.

Examples

See also

`is_wlan`, `set_wlan_type`, `set_wlan_channel`, `set_wlan_ssid`, `set_wlan_antenna`, `set_wlan_phy_mode`,
`set_wlan_short_preamble`, `set_wlan_short_slot`, `set_wlan_cts_protection`, `set_wlan_background_scan`,
`set_wlan_encryption_type`, `set_wlan_authentication_type`, `set_wlan_wep_key_length`,
`set_wlan_wep_key_index`, `set_wlan_wep_key_data_type`, `set_wlan_wep_key`, `set_wlan_wpa_passphrase`,
`set_wlan_shared_key`, `set_wlan_wpa_enterprise`, `set_wlan_wpa_enterprise_id`,
`set_wlan_wpa_enterprise_pwd`.

Remarks

[Shared Key]는 NULL을 포함해서 최대 64바이트입니다. 보다 자세한 내용은 해당 제품 사용 설명서를 참조하시기 바랍니다.

5.6.18 set_wlan_wpa_enterprise

```
int set_wlan_wpa_enterprise(unsigned char *mac_address, int enterprise_type);
```

Description

`set_wlan_wpa_enterprise`는 무선랜 제품의 [보안 설정] 중에서 [802.1X] 설정 값을 변경합니다.

Parameters



*unsigned char *mac_address*

EzTCP_Search 또는 *EzTCP_Read*를 사용하여 검색된 제품(ezTCP)의 MAC주소 입니다.

int enterprise_type

[802.1X]종류를 나타내며 아래와 같은 값을 가질 수 있습니다.

0	사용안함
1	EAP TLS
2	EAP TTLS
3	PEAP

Return values

*set_wlan_wpa_enterprise*의 반환 값은 다음과 같습니다.

1	설정 값 변경에 성공하면 1을 반환합니다.
-1	<i>mac_address</i> 을 가진 제품(ezTCP)을 찾을 수 없는 경우에는 -1을 반환합니다.
-2	<i>is_wlan</i> 이 1을 반환하지 않는 경우 -2를 반환합니다.
-3	<i>is_wpa_enterprise</i> 가 1을 반환하지 않는 경우 -3을 반환합니다.

Examples

See also

is_wlan, *set_wlan_type*, *set_wlan_channel*, *set_wlan_ssid*, *set_wlan_antenna*, *set_wlan_phy_mode*,
set_wlan_short_preamble, *set_wlan_short_slot*, *set_wlan_cts_protection*, *set_wlan_background_scan*,
set_wlan_encryption_type, *set_wlan_authentication_type*, *set_wlan_wep_key_length*,
set_wlan_wep_key_index, *set_wlan_wep_key_data_type*, *set_wlan_wep_key*, *set_wlan_wpa_passphrase*,
set_wlan_shared_key, *is_wpa_enterprise*, *set_wlan_wpa_enterprise*, *set_wlan_wpa_enterprise_id*,
set_wlan_wpa_enterprise_pwd.

Remarks

보다 자세한 내용은 해당 제품 사용설명서를 참조하시기 바랍니다.

5.6.19 set_wlan_wpa_enterprise_id

```
int set_wlan_wpa_enterprise_id(unsigned char *mac_address, char *enterprise_id);
```

Description

*set_wlan_wpa_enterprise_id*는 무선랜 제품의 [보안 설정] 중에서 802.1X [아이디]를 변경합니다.

Parameters

*unsigned char *mac_address*

EzTCP_Search 또는 *EzTCP_Read*를 사용하여 검색된 제품(ezTCP)의 MAC주소 입니다.

*char *enterprise_id*

802.1X 아이디가 저장되어 있는 포인터입니다.

Return values

*set_wlan_wpa_enterprise_id*의 반환 값은 다음과 같습니다.

1

설정 값 변경에 성공하면 1을 반환합니다.

-1

*mac_address*을 가진 제품(ezTCP)을 찾을 수 없는 경우에는 -1을 반환합니다.

-2

*is_wlan*이 1을 반환하지 않는 경우 -2를 반환합니다.

-3

*is_wpa_enterprise*가 1을 반환하지 않는 경우 -3을 반환합니다.

Examples

See also

is_wlan, *set_wlan_type*, *set_wlan_channel*, *set_wlan_ssid*, *set_wlan_antenna*, *set_wlan_phy_mode*, *set_wlan_short_preamble*, *set_wlan_short_slot*, *set_wlan_cts_protection*, *set_wlan_background_scan*, *set_wlan_encryption_type*, *set_wlan_authentication_type*, *set_wlan_wep_key_length*, *set_wlan_wep_key_index*, *set_wlan_wep_key_data_type*, *set_wlan_wep_key*, *set_wlan_wpa_passphrase*, *set_wlan_shared_key*, *is_wpa_enterprise*, *set_wlan_wpa_enterprise*, *set_wlan_wpa_enterprise_id*, *set_wlan_wpa_enterprise_pwd*.

Remarks

802.1X [아이디]는 NULL을 제외하고 최대 32바이트입니다. 보다 자세한 내용은 해당 제품 사용설명서를 참조하시기 바랍니다.

5.6.20 set_wlan_wpa_enterprise_pwd

```
int set_wlan_wpa_enterprise_pwd(unsigned char *mac_address, char *enterprise_pwd);
```

Description

*set_wlan_wpa_enterprise_pwd*는 무선랜 제품의 [보안 설정] 중에서 802.1X [비밀번호]를 변경합니다.

Parameters

*unsigned char *mac_address*

EzTCP_Search 또는 *EzTCP_Read*를 사용하여 검색된 제품(ezTCP)의 MAC주소 입니다.

*char *enterprise_pwd*

802.1X 비밀번호가 저장되어 있는 포인터입니다.

Return values

*set_wlan_wpa_enterprise_pwd*의 반환 값은 다음과 같습니다.

1

설정 값 변경에 성공하면 1을 반환합니다.

-1

*mac_address*을 가진 제품(ezTCP)을 찾을 수 없는 경우에는 -1을 반환합니다.

-2

*is_wlan*이 1을 반환하지 않는 경우 -2를 반환합니다.

-3

*is_wpa_enterprise*가 1을 반환하지 않는 경우 -3을 반환합니다.

Examples

See also

is_wlan, *set_wlan_type*, *set_wlan_channel*, *set_wlan_ssid*, *set_wlan_antenna*, *set_wlan_phy_mode*, *set_wlan_short_preamble*, *set_wlan_short_slot*, *set_wlan_cts_protection*, *set_wlan_background_scan*, *set_wlan_encryption_type*, *set_wlan_authentication_type*, *set_wlan_wep_key_length*, *set_wlan_wep_key_index*, *set_wlan_wep_key_data_type*, *set_wlan_wep_key*, *set_wlan_wpa_passphrase*, *set_wlan_shared_key*, *is_wpa_enterprise*, *set_wlan_wpa_enterprise*, *set_wlan_wpa_enterprise_id*, *set_wlan_wpa_enterprise_pwd*.

Remarks

무선랜 제품의 [보안 설정] 중에서 [802.1X] 설정 값이 2(EAP TTLS) 또는 3(PEAP)를 반환하는 경우에만 사용됩니다.

802.1X [비밀번호]는 NULL을 제외하고 최대 32바이트입니다.

보다 자세한 내용은 해당 제품 사용설명서를 참조하시기 바랍니다.

5.7 부가기능

5.7.1 set_telnet

```
int set_telnet(unsigned char *mac_address, int onoff);
```

Description

*set_telnet*은 [텔넷] 기능의 선택 여부를 변경합니다.

Parameters

*unsigned char *mac_address*

EzTCP_Search 또는 *EzTCP_Read*를 사용하여 검색된 제품(ezTCP)의 MAC주소 입니다.

int onoff

해당 기능의 선택 여부를 결정하며 아래와 같은 값을 가질 수 있습니다.

0

기능 선택을 해제합니다.

1

기능을 선택합니다.

Return values

*set_telnet*의 반환 값은 다음과 같습니다.

1

설정 값 변경에 성공하면 1을 반환합니다.

-1

*mac_address*을 가진 제품(ezTCP)을 찾을 수 없는 경우에는 -1을 반환합니다.

Examples

See also

set_telnet, *set_send_mac_address*, *set_ssl*, *set_ssh*, *set_ip4_address_search*, *set_remote_debug*, *set_tcp_multi_connection*, *set_power_management*, *set_comment*.

Remarks

5.7.2 set_send_mac_address

```
int set_send_mac_address(unsigned char *mac_address, int onoff);
```

Description

*set_send_mac_address*는 [MAC 주소 전송] 기능의 선택 여부를 변경합니다.

Parameters

*unsigned char *mac_address*

EzTCP_Search 또는 *EzTCP_Read*를 사용하여 검색된 제품(ezTCP)의 MAC주소 입니다.

int onoff

해당 기능의 선택 여부를 결정하며 아래와 같은 값을 가질 수 있습니다.

0

기능 선택을 해제합니다.

1

기능을 선택합니다.

Return values

*set_send_mac_address*의 반환 값은 다음과 같습니다.

1

설정 값 변경에 성공하면 1을 반환합니다.

-1

*mac_address*을 가진 제품(ezTCP)을 찾을 수 없는 경우에는 -1을 반환합니다.

-2

*is_send_mac_address*가 1을 반환하지 않는 경우 -2를 반환합니다.

-3

[SSL 보안통신]과 [MAC 주소 전송]을 동시에 사용하지 못하는 제품(ezTCP)은 두 기능을 동시에 설정하는 경우 -3을 반환합니다.

-4

[SSH 보안통신]이 설정되어있는 경우 -4를 반환합니다.

Examples

See also

set_telnet, *is_send_mac_address*, *set_send_mac_address*, *set_ssl*, *set_ssh*, *set_ip4_address_search*, *set_remote_debug*, *set_tcp_multi_connection*, *set_power_management*, *set_comment*.

Remarks

5.7.3 set_ssl

```
int set_ssl(unsigned char *mac_address, int onoff);
```

Description

*set_ssl*은 [SSL 보안통신] 기능의 선택 여부를 변경합니다.

Parameters

*unsigned char *mac_address*

EzTCP_Search 또는 *EzTCP_Read*를 사용하여 검색된 제품(ezTCP)의 MAC주소 입니다.

int onoff

해당 기능의 선택 여부를 결정하며 아래와 같은 값을 가질 수 있습니다.

0

기능 선택을 해제합니다.

1

기능을 선택합니다.

Return values

*set_ssl*의 반환 값은 다음과 같습니다.

1

설정 값 변경에 성공하면 1을 반환합니다.

-1

*mac_address*을 가진 제품(ezTCP)을 찾을 수 없는 경우에는 -1을 반환합니다.

-2

*is_ssl*이 1을 반환하지 않는 경우 -2를 반환합니다.

-3

[SSH 보안통신]이 설정되어있는 경우 -3를 반환합니다.

-4

[SSL 보안통신]과 [MAC 주소 전송]을 동시에 사용하지 못하는 제품(ezTCP)은 두 기능을 동시에 설정하는 경우 -4을 반환합니다.

-5

RS-485 또는 RS-422과 [SSL 보안통신]을 동시에 사용하지 못하는 제품(ezTCP)인 경우 -5를 반환합니다.

-6

시리얼 포트의 [통신모드]가 U2S – UDP인 경우 -6를 반환합니다.

-7

IPv6 기능이 선택된 경우 -7을 반환합니다.

-8

시리얼 포트의 [시리얼 포트 설정/상태 전송(RFC2217)]가 선택된 경우 -8을 반환합니다.

Examples

See also

set_telnet, set_send_mac_address, is_ssl, set_ssl, set_ssh, set_ip4_address_search, set_remote_debug, set_tcp_multi_connection, set_power_management, set_comment.

Remarks

5.7.4 set_ssh

```
int set_ssh(unsigned char *mac_address, int onoff);
```

Description

*set_ssh*는 [SSH 보안통신] 기능의 선택 여부를 반환합니다.

Parameters

*unsigned char *mac_address*

EzTCP_Search 또는 *EzTCP_Read*를 사용하여 검색된 제품(ezTCP)의 MAC주소 입니다.

int onoff

해당 기능의 선택 여부를 결정하며 아래와 같은 값을 가질 수 있습니다.

0

기능 선택을 해제합니다.

1

기능을 선택합니다.

Return values

*set_ssh*의 반환 값은 다음과 같습니다.

1

설정 값 변경에 성공하면 1을 반환합니다.

-1

*mac_address*을 가진 제품(ezTCP)을 찾을 수 없는 경우에는 -1을 반환합니다.

-2

*is_ssh*가 1을 반환하지 않는 경우 -2를 반환합니다.

-3

[SSL 보안통신]이 설정되어있는 경우 -3를 반환합니다.

-4

시리얼 포트의 [시리얼 종류]가 RS-232가 아닌 경우 -4을 반환합니다.

-5

시리얼 포트의 [통신모드]가 T2S - TCP 서버가 아닌 경우 -5를 반환합니다.

-6

[MAC 주소 전송]이 선택된 경우 -6를 반환합니다.

-7

시리얼 포트의 [시리얼 포트 설정/상태 전송(RFC2217)]가 선택된 경우 -7을 반환합니다.

Examples

See also

set_telnet, set_send_mac_address, set_ssl, is_ssh, set_ssh, set_ip4_address_search, set_remote_debug, set_tcp_multi_connection, set_power_management, set_comment.

Remarks

5.7.5 set_ip4_address_search

```
int set_ip4_address_search(unsigned char *mac_address, int onoff);
```

Description

*set_ip4_address_search*는 [IPv4 주소 검색] 기능의 선택 여부를 변경합니다.

Parameters

*unsigned char *mac_address*

EzTCP_Search 또는 *EzTCP_Read*를 사용하여 검색된 제품(ezTCP)의 MAC주소 입니다.

int onoff

해당 기능의 선택 여부를 결정하며 아래와 같은 값을 가질 수 있습니다.

0

기능 선택을 해제합니다.

1

기능을 선택합니다.

Return values

set_ip4_address_search의 반환 값은 다음과 같습니다.

1

설정 값 변경에 성공하면 1을 반환합니다.

-1

*mac_address*을 가진 제품(ezTCP)을 찾을 수 없는 경우에는 -1을 반환합니다.

Examples

See also

set_telnet, set_send_mac_address, set_ssl, set_ssh, set_ip4_address_search, set_remote_debug,
set_tcp_multi_connection, set_power_management, set_comment.

Remarks

5.7.6 set_remote_debug

```
int set_remote_debug(unsigned char *mac_address, int onoff);
```

Description

*set_remote_debug*는 [디버깅 로그 보기] 기능의 선택 여부를 변경합니다.

Parameters

*unsigned char *mac_address*

EzTCP_Search 또는 *EzTCP_Read*를 사용하여 검색된 제품(ezTCP)의 MAC주소 입니다.

int onoff

해당 기능의 선택 여부를 결정하며 아래와 같은 값을 가질 수 있습니다.

0

기능 선택을 해제합니다.

1

기능을 선택합니다.

Return values

*set_remote_debug*의 반환 값은 다음과 같습니다.

1

설정 값 변경에 성공하면 1을 반환합니다.

-1

*mac_address*을 가진 제품(ezTCP)을 찾을 수 없는 경우에는 -1을 반환합니다.

-2

*is_remote_debug*가 1을 반환하지 않는 경우 -2를 반환합니다.

Examples

See also

set_telnet, set_send_mac_address, set_ssl, set_ssh, set_ip4_address_search, is_remote_debug,
set_remote_debug, set_tcp_multi_connection, set_power_management, set_comment.

Remarks

5.7.7 set_tcp_multi_connection

```
int set_tcp_multi_connection(unsigned char *mac_address, int onoff);
```

Description

*set_tcp_multi_connection*는 [다중 접속] 기능의 선택 여부를 변경합니다.

Parameters

*unsigned char *mac_address*

EzTCP_Search 또는 *EzTCP_Read*를 사용하여 검색된 제품(ezTCP)의 MAC주소 입니다.

int onoff

해당 기능의 선택 여부를 결정하며 아래와 같은 값을 가질 수 있습니다.

0

기능 선택을 해제합니다.

1

기능을 선택합니다.

Return values

*set_tcp_multi_connection*의 반환 값은 다음과 같습니다.

1

설정 값 변경에 성공하면 1을 반환합니다.

-1

*mac_address*을 가진 제품(ezTCP)을 찾을 수 없는 경우에는 -1을 반환합니다.

-2

*is_tcp_multi_connection*이 1을 반환하지 않는 경우 -2를 반환합니다.

-3

시리얼 포트의 [통신모드]가 T2S-TCP 서버가 아닌 경우 -3을 반환합니다.

-4

[MAC 주소 전송]이 선택된 경우 -4를 반환합니다.

-5

[SSL 보안통신]이 설정되어있는 경우 -5를 반환합니다.

-6

[SSH 보안통신]이 설정되어있는 경우 -6을 반환합니다.

Examples

See also

set_telnet, *set_send_mac_address*, *set_ssl*, *set_ssh*, *set_ip4_address_search*, *set_remote_debug*,

is_tcp_multi_connection, set_tcp_multi_connection, set_power_management, set_comment.

Remarks

5.7.8 set_power_management

```
int set_power_management(unsigned char *mac_address, int onoff);
```

Description

*set_power_management*는 [전원 관리] 기능의 선택 여부를 변경합니다.

Parameters

*unsigned char *mac_address*

EzTCP_Search 또는 *EzTCP_Read*를 사용하여 검색된 제품(ezTCP)의 MAC주소 입니다.

int onoff

해당 기능의 선택 여부를 결정하며 아래와 같은 값을 가질 수 있습니다.

0

기능 선택을 해제합니다.

1

기능을 선택합니다.

Return values

*set_power_management*의 반환 값은 다음과 같습니다.

1

설정 값 변경에 성공하면 1을 반환합니다.

-1

*mac_address*을 가진 제품(ezTCP)을 찾을 수 없는 경우에는 -1을 반환합니다.

-2

*is_power_management*가 1을 반환하지 않는 경우 -2를 반환합니다.

Examples

See also

set_telnet, set_send_mac_address, set_ssl, set_ssh, set_ip4_address_search, set_remote_debug, set_tcp_multi_connection, is_power_management, set_power_management, set_comment.

Remarks

5.7.9 set_comment

```
int set_comment(unsigned char *mac_address, char *comment);
```

Description

*set_comment*는 제품(ezTCP)의 [설명]을 변경합니다.

Parameters

*unsigned char *mac_address*

EzTCP_Search 또는 *EzTCP_Read*를 사용하여 검색된 제품(ezTCP)의 MAC주소 입니다.

*char *comment*

제품(ezTCP)의 [설명]이 저장되어 있는 포인터입니다.

Return values

*set_comment*의 반환 값은 다음과 같습니다.

1

설정 값 변경에 성공하면 1을 반환합니다.

-1

*mac_address*을 가진 제품(ezTCP)을 찾을 수 없는 경우에는 -1을 반환합니다.

Examples

See also

set_telnet, *set_send_mac_address*, *set_ssl*, *set_ssh*, *set_ip4_address_search*, *set_remote_debug*,
set_tcp_multi_connection, *set_power_management*, *set_comment*.

Remarks

제품(ezTCP)의 [설명]은 NULL을 제외하고 최대 64바이트입니다.

5.7.10 set_allowed_mac_address

```
int set_allowed_mac_address(unsigned char *mac_address, int index,  
                             char *in_mac_address);
```

Description

*set_allowed_mac_address*는 여러 가지 제품(ezTCP) 접근 제한 기능 중에서 [다음의 MAC 주소만 접근 가능]에 입력되어있는 MAC 주소를 변경합니다.

Parameters



*unsigned char *mac_address*

EzTCP_Search 또는 *EzTCP_Read*를 사용하여 검색된 제품(ezTCP)의 MAC주소 입니다.

int index

MAC 주소 6바이트 중에서 변경할 MAC 주소의 순번으로 0~5입니다.

Return values

*set_allowed_mac_address*의 반환 값은 다음과 같습니다.

1

설정 값 변경에 성공하면 1을 반환합니다.

-1

*mac_address*을 가진 제품(ezTCP)을 찾을 수 없는 경우에는 -1을 반환합니다.

-2

*index*에 유효하지 않은 값을 입력한 경우 -2를 반환합니다.

-3

*in_mac_address*에 유효하지 않은 값을 입력한 경우 -3을 반환합니다.

Examples

See also

set_allowed_mac_address, *set_allowed_ip4_address*, *set_allowed_ip4_network_mask_type*,
set_allowed_ezmanager, *set_allowed_ip6_address*, *set_allowed_ip6_subnet_prefix_length*.

Remarks

제품(ezTCP) 접근 제한에 대한 보다 자세한 내용은 해당 제품 사용설명서를 참조하십시오.

5.7.11 set_allowed_ip4_address

```
int set_allowed_ip4_address(unsigned char *mac_address, char *ip4_address);
```

Description

*set_allowed_ip4_address*는 여러 가지 제품(ezTCP) 접근 제한 기능 중에서 [다음의 IP 주소 대역만 접근 가능]에 입력되어있는 [IPv4 주소]를 변경합니다.

Parameters

*unsigned char *mac_address*

EzTCP_Search 또는 *EzTCP_Read*를 사용하여 검색된 제품(ezTCP)의 MAC주소 입니다.

*char *ip4_address*

IPv4 주소가 저장되어 있는 포인터입니다.

Return values

*set_allowed_ip4_address*의 반환 값은 다음과 같습니다.

1

설정 값 변경에 성공하면 1을 반환합니다.

-1

*mac_address*을 가진 제품(ezTCP)을 찾을 수 없는 경우에는 -1을 반환합니다.

Examples**See also**

set_allowed_mac_address, *set_allowed_ip4_address*, *set_allowed_ip4_network_mask_type*,
set_allowed_ezmanager, *set_allowed_ip6_address*, *set_allowed_ip6_subnet_prefix_length*.

Remarks

제품(ezTCP) 접근 제한에 대한 보다 자세한 내용은 해당 제품 사용설명서를 참조하십시오.

5.7.12 set_allowed_ip4_network_mask_type

```
int set_allowed_ip4_network_mask_type(unsigned char *mac_address,
                                       int network_mask_type);
```

Description

*set_allowed_ip4_network_mask_type*은 여러 가지 제품(ezTCP) 접근 제한 기능 중에서 [다음의 IP 주소 대역만 접근 가능]에 입력되어있는 [넷 마스크(IPv4 주소 대역)]의 형태를 변경합니다.

Parameters

*unsigned char *mac_address*

EzTCP_Search 또는 *EzTCP_Read*를 사용하여 검색된 제품(ezTCP)의 MAC주소 입니다.

int network_mask_type

[넷 마스크(IPv4 주소 대역)] 설정 값을 나타내며 아래와 같은 값을 가질 수 있습니다.

0

255.255.255.255

1

255.255.255.0

2

255.255.0.0

3

255.0.0.0

4

0.0.0.0

Return values

*set_allowed_ip4_network_mask_type*의 반환 값은 다음과 같습니다.

1

설정 값 변경에 성공하면 1을 반환합니다.

-1

*mac_address*을 가진 제품(ezTCP)을 찾을 수 없는 경우에는 -1을 반환합니다.

-2

*network_mask_type*에 유효하지 않는 값을 입력한 경우 -2를 반환합니다.

Examples**See also**

set_allowed_mac_address, *set_allowed_ip4_address*, *set_allowed_ip4_network_mask_type*,
set_allowed_ezmanager, *set_allowed_ip6_address*, *set_allowed_ip6_subnet_prefix_length*.

Remarks

제품(ezTCP) 접근 제한에 대한 보다 자세한 내용은 해당 제품 사용설명서를 참조하십시오.

5.7.13 set_allowed_ezmanager

```
int set_allowed_ezmanager(unsigned char *mac_address, int onoff);
```

Description

*set_allowed_ezmanager*은 여러 가지 제품(ezTCP) 접근 제한 기능 중에서 [ezManager에도 적용]의 선택 여부를 변경합니다.

Parameters

*unsigned char *mac_address*

EzTCP_Search 또는 *EzTCP_Read*를 사용하여 검색된 제품(ezTCP)의 MAC주소 입니다.

int onoff

해당 기능의 선택 여부를 결정하며 아래와 같은 값을 가질 수 있습니다.

0

기능 선택을 해제합니다.

1

기능을 선택합니다.

Return values

*set_allowed_ezmanager*의 반환 값은 다음과 같습니다.

1

설정 값 변경에 성공하면 1을 반환합니다.

-1

*mac_address*을 가진 제품(ezTCP)을 찾을 수 없는 경우에는 -1을 반환합니다.

Examples**See also**

set_allowed_mac_address, *set_allowed_ip4_address*, *set_allowed_ip4_network_mask_type*,
set_allowed_ezmanager, *set_allowed_ip6_address*, *set_allowed_ip6_subnet_prefix_length*.

Remarks

제품(ezTCP) 접근 제한에 대한 보다 자세한 내용은 해당 제품 사용설명서를 참조하십시오.

5.7.14 set_allowed_ip6_address

```
int set_allowed_ip6_address(unsigned char *mac_address, char *ip6_address);
```

Description

*set_allowed_ip6_address*는 여러 가지 제품(ezTCP) 접근 제한 기능 중에서 [다음의 IP 주소 대역만 접근 가능]에 입력되어있는 [IPv6 주소]를 변경합니다.

Parameters

*unsigned char *mac_address*

EzTCP_Search 또는 *EzTCP_Read*를 사용하여 검색된 제품(ezTCP)의 MAC주소 입니다.

*char *ip6_address*

IPv6 주소가 저장되어 있는 포인터입니다.

Return values

*set_allowed_ip6_address*의 반환 값은 다음과 같습니다.

1

설정 값 변경에 성공하면 1을 반환합니다.

-1

*mac_address*을 가진 제품(ezTCP)을 찾을 수 없는 경우에는 -1을 반환합니다.

-2

*ip6_address*에 잘못된 형식의 IPv6 주소를 입력한 경우에는 -2를 반환합니다.

Examples

See also

set_allowed_mac_address, set_allowed_ip4_address, set_allowed_ip4_network_mask_type,
set_allowed_ezmanager, set_allowed_ip6_address, set_allowed_ip6_subnet_prefix_length.

Remarks

제품(ezTCP) 접근 제한에 대한 보다 자세한 내용은 해당 제품 사용설명서를 참조하십시오.

5.7.15 set_allowed_ip6_subnet_prefix_length

```
int set_allowed_ip6_subnet_prefix_length(unsigned char *mac_address,  
                                         int subnet_prefix_length);
```

Description

*set_allowed_ip6_subnet_prefix_length*는 여러 가지 제품(ezTCP) 접근 제한 기능 중에서 [다음의 IP 주소 대역만 접근 가능]에 입력되어있는 [서브넷 접두사 길이]를 변경합니다.

Parameters

*unsigned char *mac_address*

EzTCP_Search 또는 *EzTCP_Read*를 사용하여 검색된 제품(ezTCP)의 MAC주소 입니다.

int subnet_prefix_length

변경할 서브넷 접두사 길이로 0 ~ 128입니다.

Return values

*set_allowed_ip6_subnet_prefix_length*의 반환 값은 다음과 같습니다.

1

설정 값 변경에 성공하면 1을 반환합니다.

-1

*mac_address*을 가진 제품(ezTCP)을 찾을 수 없는 경우에는 -1을 반환합니다.

-2

*subnet_prefix_length*에 유효하지 않는 값을 입력한 경우 -2를 반환합니다.

Examples

See also

set_allowed_mac_address, set_allowed_ip4_address, set_allowed_ip4_network_mask_type,
set_allowed_ezmanager, set_allowed_ip6_address, set_allowed_ip6_subnet_prefix_length.

Remarks

제품(ezTCP) 접근 제한에 대한 보다 자세한 내용은 해당 제품 사용설명서를 참조하십시오.

5.7.16 set_ip4_change_notification_type

```
int set_ip4_change_notification_type(unsigned char *mac_address, int notification_type);
```

Description

*set_ip4_change_notification_type*은 IPv4 주소 통보 기능의 [프로토콜]을 변경합니다.

Parameters

*unsigned char *mac_address*

EzTCP_Search 또는 *EzTCP_Read*를 사용하여 검색된 제품(ezTCP)의 MAC주소 입니다.

int notification_type

[프로토콜] 설정 값을 나타내며 아래와 같은 값을 가질 수 있습니다.

0

사용안함

1

DDNS(dyndns.org)

2

TCP

3

UDP

Return values

*set_ip4_change_notification_type*의 반환 값은 다음과 같습니다.

1

설정 값 변경에 성공하면 1을 반환합니다.

-1

*mac_address*을 가진 제품(ezTCP)을 찾을 수 없는 경우에는 -1을 반환합니다.

-2

*notification_type*에 유효하지 않는 값을 입력한 경우 -2를 반환합니다.

Examples

See also

set_ip4_change_notification_type,
set_ip4_change_notification_interval,

set_ip4_change_notification_data_type,
set_ip4_change_notification_peer_port,

set_ip4_change_notification_peer_address,
set_ip4_change_notification_ddns_pwd.

set_ip4_change_notification_ddns_id,

Remarks

IP 주소 통보기능에 대한 보다 자세한 내용은 해당 제품 사용설명서를 참조하십시오.

5.7.17 set_ip4_change_notification_data_type

```
int set_ip4_change_notification_data_type(unsigned char *mac_address, int data_type);
```

Description

set_ip4_change_notification_data_type은 IPv4 주소 통보 기능의 [데이터 형식]을 변경합니다.

Parameters

*unsigned char *mac_address*

EzTCP_Search 또는 EzTCP_Read를 사용하여 검색된 제품(ezTCP)의 MAC주소 입니다.

int notification_type

[데이터 형식] 설정 값을 나타내며 아래와 같은 값을 가질 수 있습니다.

0

ASCII

1

16진수

Return values

set_ip4_change_notification_data_type의 반환 값은 다음과 같습니다.

1

설정 값 변경에 성공하면 1을 반환합니다.

-1

mac_address을 가진 제품(ezTCP)을 찾을 수 없는 경우에는 -1을 반환합니다.

-2

data_type에 유효하지 않는 값을 입력한 경우 -2를 반환합니다.

Examples

See also

set_ip4_change_notification_type,
set_ip4_change_notification_interval,
set_ip4_change_notification_peer_address,

set_ip4_change_notification_data_type,
set_ip4_change_notification_peer_port,
set_ip4_change_notification_ddns_id,

set_ip4_change_notification_ddns_pwd.

Remarks

IPv4 주소 통보 기능의 [데이터 형식]은 [프로토콜]이 2(TCP) 또는 3(UDP)인 경우에만 사용됩니다.

IP 주소 통보기능에 대한 보다 자세한 내용은 해당 제품 사용설명서를 참조하십시오.

5.7.18 set_ip4_change_notification_interval

```
int set_ip4_change_notification_interval(unsigned char *mac_address, int interval);
```

Description

set_ip4_change_notification_interval은 IPv4 주소 통보 기능의 [통보 주기]를 변경합니다.

Parameters

*unsigned char *mac_address*

EzTCP_Search 또는 EzTCP_Read를 사용하여 검색된 제품(ezTCP)의 MAC주소 입니다.

int interval

변경할 통보 주기입니다.

Return values

set_ip4_change_notification_interval의 반환 값은 다음과 같습니다.

1

설정 값 변경에 성공하면 1을 반환합니다.

-1

mac_address을 가진 제품(ezTCP)을 찾을 수 없는 경우에는 -1을 반환합니다.

-2

[프로토콜]이 2(TCP) 또는 3(UDP)이 아니면 -2를 반환합니다.

Examples

See also

set_ip4_change_notification_type,
set_ip4_change_notification_interval,
set_ip4_change_notification_peer_address,
set_ip4_change_notification_ddns_pwd.

set_ip4_change_notification_data_type,
set_ip4_change_notification_peer_port,
set_ip4_change_notification_ddns_id,

Remarks

IP 주소 통보기능에 대한 보다 자세한 내용은 해당 제품 사용설명서를 참조하십시오.

5.7.19 set_ip4_change_notification_peer_port

```
int set_ip4_change_notification_peer_port(unsigned char *mac_address, int peer_port);
```

Description

*set_ip4_change_notification_peer_port*은 IPv4 주소 통보 기능의 [포트]를 변경합니다.

Parameters

*unsigned char *mac_address*

EzTCP_Search 또는 *EzTCP_Read*를 사용하여 검색된 제품(ezTCP)의 MAC주소 입니다.

int peer_port

변경할 IPv4 주소 통보 기능의 포트번호입니다.

Return values

*set_ip4_change_notification_peer_port*의 반환 값은 다음과 같습니다.

1

설정 값 변경에 성공하면 1을 반환합니다.

-1

*mac_address*을 가진 제품(ezTCP)을 찾을 수 없는 경우에는 -1을 반환합니다.

Examples

See also

set_ip4_change_notification_type,
set_ip4_change_notification_interval,
set_ip4_change_notification_peer_address,
set_ip4_change_notification_ddns_pwd.

set_ip4_change_notification_data_type,
set_ip4_change_notification_peer_port,
set_ip4_change_notification_ddns_id,

Remarks

IPv4 주소 통보 기능의 [포트]는 [프로토콜]이 2(TCP) 또는 3(UDP)인 경우에만 사용됩니다.

IP 주소 통보기능에 대한 보다 자세한 내용은 해당 제품 사용설명서를 참조하십시오.

5.7.20 set_ip4_change_notification_peer_address

```
int set_ip4_change_notification_peer_address(unsigned char *mac_address,  
                                              char *peer_address);
```

Description

*set_ip4_change_notification_peer_address*은 IPv4 주소 통보 기능의 [호스트 이름] 또는

[통보할 주소]를 변경합니다.

Parameters

*unsigned char *mac_address*

EzTCP_Search 또는 *EzTCP_Read*를 사용하여 검색된 제품(ezTCP)의 MAC주소 입니다.

*char *peer_address*

호스트 이름 또는 통보할 주소가 저장되어 있는 포인터입니다.

Return values

*set_ip4_change_notification_peer_address*의 반환 값은 다음과 같습니다.

1

설정 값 변경에 성공하면 1을 반환합니다.

-1

*mac_address*을 가진 제품(ezTCP)을 찾을 수 없는 경우에는 -1을 반환합니다.

Examples

See also

set_ip4_change_notification_type,
set_ip4_change_notification_interval,
set_ip4_change_notification_peer_address,
set_ip4_change_notification_ddns_pwd.

set_ip4_change_notification_data_type,
set_ip4_change_notification_peer_port,
set_ip4_change_notification_ddns_id,

Remarks

IPv4 주소 통보 기능의 [프로토콜]이 1(DDNS(dyndng.org))인 경우 [호스트 이름]을 변경합니다.

IPv4 주소 통보 기능의 [프로토콜]이 2(TCP) 또는 3(UDP)인 경우에는 [통보할 주소]를 변경합니다.

[호스트 이름] 또는 [통보할 주소]는 NULL을 포함하여 최대 64바이트입니다.

IP 주소 통보기능에 대한 보다 자세한 내용은 해당 제품 사용설명서를 참조하십시오.

5.7.21 set_ip4_change_notification_ddns_id

```
int set_ip4_change_notification_ddns_id(unsigned char *mac_address, char *ddns_id);
```

Description

*set_ip4_change_notification_ddns_id*는 IPv4 주소 통보 기능의 [DDNS 아이디]를 변경합니다.

Parameters

*unsigned char *mac_address*

EzTCP_Search 또는 *EzTCP_Read*를 사용하여 검색된 제품(ezTCP)의 MAC주소 입니다.
`char *ddns_id`

DDNS 아이디가 저장되어 있는 포인터입니다.

Return values

*set_ip4_change_notification_ddns_id*의 반환 값은 다음과 같습니다.

1

설정 값 변경에 성공하면 1을 반환합니다.

-1

*mac_address*을 가진 제품(ezTCP)을 찾을 수 없는 경우에는 -1을 반환합니다.

Examples

See also

set_ip4_change_notification_type,
set_ip4_change_notification_interval,
set_ip4_change_notification_peer_address,
set_ip4_change_notification_ddns_pwd.

set_ip4_change_notification_data_type,
set_ip4_change_notification_peer_port,
set_ip4_change_notification_ddns_id,

Remarks

IPv4 주소 통보 기능의 [DDNS 아이디]는 [프로토콜]이 1(DDNS(dyndns.org))인 경우에만 사용 됩니다.

[DDNS 아이디]는 NULL을 제외하고 최대 32바이트입니다.

IP 주소 통보기능에 대한 보다 자세한 내용은 해당 제품 사용설명서를 참조하십시오.

5.7.22 set_ip4_change_notification_ddns_pwd

```
int set_ip4_change_notification_ddns_pwd(unsigned char *mac_address,
                                         char *ddns_pwd);
```

Description

*set_ip4_change_notification_ddns_pwd*는 IPv4 주소 통보 기능의 [DDNS 비밀번호]를 변경 합니다.

Parameters

*unsigned char *mac_address*

EzTCP_Search 또는 *EzTCP_Read*를 사용하여 검색된 제품(ezTCP)의 MAC주소 입니다.

*char *ddns_pwd*

DDNS 비밀번호가 저장되어 있는 포인터입니다.

Return values

*set_ip4_change_notification_ddns_pwd*의 반환 값은 다음과 같습니다.

1

설정 값 변경에 성공하면 1을 반환합니다.

-1

*mac_address*을 가진 제품(ezTCP)을 찾을 수 없는 경우에는 -1을 반환합니다.

Examples**See also**

set_ip4_change_notification_type,
set_ip4_change_notification_interval,
set_ip4_change_notification_peer_address,
set_ip4_change_notification_ddns_pwd.

set_ip4_change_notification_data_type,
set_ip4_change_notification_peer_port,
set_ip4_change_notification_ddns_id,

Remarks

IPv4 주소 통보 기능의 [DDNS 비밀번호]는 [프로토콜]이 1(DDNS(dyndns.org))인 경우에만 사용됩니다.

[DDNS 비밀번호]는 NULL을 제외하고 최대 16바이트입니다.

IP 주소 통보기능에 대한 보다 자세한 내용은 해당 제품 사용설명서를 참조하십시오.

6 설정 값 저장

6.1 EzTCP_Write

```
int EzTCP_Write(unsigned char *mac_address, char *password, int udp_port_number,
                int *error);
```

Description

제품(ezTCP)의 MAC주소를 사용하여 설정 값을 제품(ezTCP)에 저장합니다.

Parameters

*unsigned char *mac_address*

EzTCP_Search 또는 *EzTCP_Read*를 사용하여 검색된 제품(ezTCP)의 MAC주소 입니다.

*char *password*

제품(ezTCP)에 비밀번호가 설정된 경우 현재 제품(ezTCP)에 설정되어있는 비밀번호를 입력해야 합니다.

int udp_port_number

EzTCP_Search 또는 *EzTCP_Read*로 제품(ezTCP)을 검색할 때 사용한 UDP 포트 번호 입니다. 0을 입력하면 기본값인 50005번, 50007번을 사용합니다.

*int *error*

함수 실행이 실패한 경우 오류 값이 저장됩니다.

Return values

*EzTCP_Write*의 반환 값은 다음과 같습니다.

EZTCP_SUCCESS

제품(ezTCP)에 설정 값을 성공적으로 저장하면 EZTCP_SUCCESS를 반환합니다.

EZTCP_ERR

EzTCP_Write 실행 중 오류가 발생한 경우 EZTCP_ERR을 반환합니다. 이 경우 *error*에 오류 번호를 저장합니다.

오류 번호가 0보다 작은 경우에는 아래와 같은 값을 가질 수 있습니다.

EZTCP_ERR_NO_INFO

*mac_address*을 가진 제품(ezTCP)을 찾을 수 없는 경우에는 EZTCP_ERR_NO_INFO를 *error*에 저장합니다.

EZTCP_ERR_PWD_LENGTH

*password*에 입력한 비밀번호의 문자열 길이가 유효하지 않으면 EZTCP_ERR_PWD_LENGTH를 *error*에 저장합니다. 비밀번호는 4 ~ 8자입니다.

EZTCP_ERR_LOCAL_IP

제품(ezTCP)에 사용할 수 없는 IP주소를 저장하려는 경우 EZTCP_ERR_LOCAL_IP를 *error*에 저장합니다.

EZTCP_ERR_LOCAL_PORT_0

제품 로컬포트 번호를 중복해서 저장하려는 경우 EZTCP_ERR_LOCAL_PORT_0를 *error*에 저장합니다.

EZTCP_ERR_LOCAL_PORT_1

[텔넷] 기능이 선택되어있을 때 제품 로컬포트 번호에 23번을 저장하려는 경우 EZTCP_ERR_LOCAL_PORT_1을 *error*에 저장합니다.

EZTCP_ERR_LOCAL_PORT_2

I/O 제품의 Modbus/TCP 접속 방법이 [수동 접속]이고 저장하려는 제품 로컬포트 번호가 Modbus/TCP [제품 로컬포트]로 이미 사용 중인 경우 EZTCP_ERR_LOCAL_PORT_2를 *error*에 저장합니다.

EZTCP_ERR_LOCAL_PORT_3

I/O 제품 [웹(HTTP)] 제어방식 기능이 선택되어있을 때 저장하려는 제품 로컬포트 번호가 I/O 제품 [웹(HTTP) 포트] 번호로 이미 사용 중인 경우 EZTCP_ERR_LOCAL_PORT_3를 *error*에 저장합니다.

EZTCP_ERR_LOCAL_PORT_4

저장하려는 제품 로컬포트 번호가 50005번이면 EZTCP_ERR_LOCAL_PORT_4을 *error*에 저장합니다.

EZTCP_ERR_LOCAL_PORT_5

저장하려는 제품 로컬포트 번호가 50006번이면 EZTCP_ERR_LOCAL_PORT_5을 *error*에 저장합니다.

EZTCP_ERR_LOCAL_PORT_6

저장하려는 제품 로컬포트 번호가 50007번이면 EZTCP_ERR_LOCAL_PORT_6을 *error*에 저장합니다.

EZTCP_ERR_IO_SCRIPT

I/O 제품에 문법에 맞지 않은 매크로를 저장하려는 경우 EZTCP_ERR_IO_SCRIPT을 *error*에 저장합니다.

EZTCP_ERR_IO_ADDR

I/O 제품의 [입력포트 주소] 또는 [출력포트 주소]에 유효하지 않은 값을 저장하려는 경우 EZTCP_ERR_IO_ADDR을 *error*에 저장합니다.

EZTCP_ERR_RES

제품(ezTCP)으로부터 응답을 수신하지 못하면 EZTCP_ERR_RES를 *error*에 저장합니다.

EZTCP_ERR_PWD

제품(ezTCP)에 설정된 비밀번호와 입력한 비밀번호가 일치하지 않으면 EZTCP_ERR_PWD를 *error*에 저장합니다.

EZTCP_ERR_PRODUCT_MISMATCH

제품종류가 일치하지 않는 경우 EZTCP_ERR_PRODUCT_MISMATCH를 *error*에 저장합니다.

EZTCP_ERR_FLASH_NOSPACE

제품(ezTCP)에 설정값을 저장할 충분한 메모리 공간이 없는 경우 EZTCP_ERR_FLASH_NOSPACE를 *error*에 저장합니다.

EZTCP_ERR_UNKNOWN

제품(ezTCP)으로부터 알 수 없는 오류번호를 수신한 경우 EZTCP_ERR_UNKNOWN을 *error*에 저장합니다.

오류번호가 0보다 큰 경우에는 함수 실행 중 사용자 시스템에서 발생한 오류번호입니다.

Examples

```
struct eztcp_info out_eztcp_info;
int ret = get_eztcp_info_by_index(0, &out_eztcp_info);

if(ret == EZTCP_SUCCESS)
{
    unsigned char *mac_address = out_eztcp_info.mac_address;

    // change some variables by using set_XXXXX functions.

    int err = 0;
    int res = EzTCP_write(mac_address, NULL, out_eztcp_info.search_port, &err);

    if(res == EZTCP_SUCCESS)
    {
        printf("The requested operation has been successfully completed!\r\n");
    }
    else if(res == EZTCP_ERR)
    {
        if(err < 0)
        {
            printf("Writing has failed.[Error : %d]\r\n", err);
        }
        else if(err > 0)
        {
            LPVOID lpMsgBuf;

            FormatMessage(FORMAT_MESSAGE_ALLOCATE_BUFFER | FORMAT_MESSAGE_FROM_SYSTEM |
                FORMAT_MESSAGE_IGNORE_INSERTS, NULL, err, MAKELANGID(LANG_NEUTRAL, SUBLANG_DEFAULT),
                (LPSTR)&lpMsgBuf, 0, NULL);

            printf("Error[%d] %s\r\n", err, (char*)lpMsgBuf);
        }
    }
}
}
```

See also

EzTCP_Write, EzTCP_Initialize.

Remarks

6.2 EzTCP_Initialize

```
int EzTCP_Initialize(unsigned char *mac_address, char *password, int udp_port_number,
                    int *error);
```

Description

제품(ezTCP)의 MAC주소를 사용하여 제품(ezTCP)의 설정 값을 초기화합니다.

Parameters

*unsigned char *mac_address*

EzTCP_Search 또는 *EzTCP_Read*를 사용하여 검색된 제품(ezTCP)의 MAC주소 입니다.

*char *password*

제품(ezTCP)에 비밀번호가 설정된 경우 현재 제품(ezTCP)에 설정되어있는 비밀번호를 입력해야 합니다.

int udp_port_number

EzTCP_Search 또는 *EzTCP_Read*로 제품(ezTCP)을 검색할 때 사용한 UDP 포트 번호 입니다. 0을 입력하면 기본값인 50005번, 50007번을 사용합니다.

*int *error*

함수 실행이 실패한 경우 오류 값이 저장됩니다.

Return values

*EzTCP_Initialize*의 반환 값은 다음과 같습니다.

EZTCP_SUCCESS

제품(ezTCP)의 설정 값을 성공적으로 초기화하면 EZTCP_SUCCESS를 반환합니다.

EZTCP_ERR

EzTCP_Initialize 실행 중 오류가 발생한 경우 EZTCP_ERR을 반환합니다. 이 경우 *error*에 오류_번호를 저장합니다.

오류 번호가 0보다 작은 경우에는 아래와 같은 값을 가질 수 있습니다.

EZTCP_ERR_NO_INFO

*mac_address*을 가진 제품(ezTCP)을 찾을 수 없는 경우에는 EZTCP_ERR_NO_INFO를 *error*에 저장합니다.

EZTCP_ERR_PWD_LENGTH

*password*에 입력한 비밀번호의 문자열 길이가 유효하지 않으면 EZTCP_ERR_PWD_LENGTH를 *error*에 저장합니다. 비밀번호는 4 ~ 8자입니다.

EZTCP_ERR_RES

제품(ezTCP)으로부터 응답을 수신하지 못하면 EZTCP_ERR_RES를 *error*에 저장합니다.

EZTCP_ERR_PWD

제품(ezTCP)에 설정된 비밀번호와 입력한 비밀번호가 일치하지 않으면 EZTCP_ERR_PWD를 *error*에 저장합니다.

EZTCP_ERR_PRODUCT_MISMATCH

제품종류가 일치하지 않는 경우 EZTCP_ERR_PRODUCT_MISMATCH를 *error*에 저장합니다.

EZTCP_ERR_FLASH_NOSPACE

제품(ezTCP)에 설정값을 저장할 충분한 메모리 공간이 없는 경우 EZTCP_ERR_FLASH_NOSPACE를 *error*에 저장합니다.

EZTCP_ERR_UNKNOWN

제품(ezTCP)으로부터 알 수 없는 오류번호를 수신한 경우 EZTCP_ERR_UNKNOWN을 *error*에 저장합니다.

오류번호가 0보다 큰 경우에는 함수 실행 중 사용자 시스템에서 발생한 오류번호입니다.

Examples

```
struct eztcp_info out_eztcp_info;
int ret = get_eztcp_info_by_index(0, &out_eztcp_info);

if(ret == EZTCP_SUCCESS)
{
    unsigned char *mac_address = out_eztcp_info.mac_address;

    int err = 0;
    int res = EzTCP_Initialize(mac_address, NULL, out_eztcp_info.search_port, &err);

    if(res == EZTCP_SUCCESS)
    {
        printf("The requested operation has been successfully completed!\r\n");
    }
    else if(res == EZTCP_ERR)
    {
        if(err < 0)
        {
            printf("Initializing has failed.[Error : %d]\r\n", err);
        }
        else if(err > 0)
        {
            LPVOID lpMsgBuf;
```

```
        FormatMessage(FORMAT_MESSAGE_ALLOCATE_BUFFER | FORMAT_MESSAGE_FROM_SYSTEM |  
        FORMAT_MESSAGE_IGNORE_INSERTS, NULL, err, MAKELANGID(LANG_NEUTRAL, SUBLANG_DEFAULT),  
        (LPSTR)&lpmgBuf, 0, NULL);  
  
        printf("Error[%d] %s\r\n", err, (char*)lpmgBuf);  
  
    }  
  
}
```

See also

EzTCP_Write, EzTCP_Initialize.

Remarks

7 비밀번호 관리

7.1 EzTCP_ChangePassword

```
int EzTCP_ChangePassword(unsigned char *mac_address, char *current_pwd,
                        char *new_pwd, int udp_port_number, int *error);
```

Description

제품(ezTCP)에 비밀번호를 설정, 변경 또는 삭제할 수 있습니다.

Parameters

*unsigned char *mac_address*

EzTCP_Search 또는 *EzTCP_Read*를 사용하여 검색된 제품(ezTCP)의 MAC주소 입니다.

*char *current_pwd*

제품(ezTCP)에 비밀번호가 설정된 경우 현재 제품(ezTCP)에 설정되어있는 비밀번호를 입력해야 합니다. *current_pwd*는 제품(ezTCP)의 비밀번호를 변경 또는 삭제할 때 사용됩니다.

현재 제품(ezTCP)에 비밀번호가 설정되어 있지 않으면 *current_pwd*는 NULL이거나 길이가 0이어야 합니다.

*char *new_pwd*

제품(ezTCP)에 저장할 비밀번호를 입력해야 합니다.

*new_pwd*는 제품(ezTCP)에 비밀번호를 처음 설정하거나 이미 설정된 비밀번호를 변경할 때 사용됩니다.

제품(ezTCP)에 설정된 비밀번호를 삭제할 때에는 *new_pwd*는 NULL이거나 길이가 0이어야 합니다.

int udp_port_number

EzTCP_Search 또는 *EzTCP_Read*로 제품(ezTCP)을 검색할 때 사용한 UDP 포트 번호 입니다. 0을 입력하면 기본값인 50005번, 50007번을 사용합니다.

*int *error*

함수 실행이 실패한 경우 오류 값이 저장됩니다.

Return values

*EzTCP_ChangePassword*의 반환 값은 다음과 같습니다.

EZTCP_SUCCESS

제품(ezTCP)의 설정 값을 성공적으로 초기화하면 EZTCP_SUCCESS를 반환합니다.

EZTCP_ERR

EzTCP_ChangePassword 실행 중 오류가 발생한 경우 EZTCP_ERR를 반환합니다. 이 경우 *error*에 오류 번호를 저장합니다.

오류 번호가 0보다 작은 경우에는 아래와 같은 값을 가질 수 있습니다.

EZTCP_ERR_NO_INFO

*mac_address*을 가진 제품(ezTCP)을 찾을 수 없는 경우에는 *EZTCP_ERR_NO_INFO*를 *error*에 저장합니다.

EZTCP_ERR_PWD_LENGTH

current_pwd 또는 *new_pwd*에 입력한 비밀번호의 문자열 길이가 유효하지 않으면 *EZTCP_ERR_PWD_LENGTH*를 *error*에 저장합니다. 비밀번호는 4~8자입니다.

EZTCP_ERR_RES

제품(ezTCP)으로부터 응답을 수신하지 못하면 *EZTCP_ERR_RES*를 *error*에 저장합니다.

EZTCP_ERR_PWD

제품(ezTCP)에 설정된 비밀번호와 입력한 비밀번호가 일치하지 않으면 *EZTCP_ERR_PWD*를 *error*에 저장합니다.

EZTCP_ERR_PRODUCT_MISMATCH

제품종류가 일치하지 않는 경우 *EZTCP_ERR_PRODUCT_MISMATCH*를 *error*에 저장합니다.

EZTCP_ERR_FLASH_NOSPACE

제품(ezTCP)에 설정값을 저장할 충분한 메모리 공간이 없는 경우 *EZTCP_ERR_FLASH_NOSPACE*를 *error*에 저장합니다.

EZTCP_ERR_UNKNOWN

제품(ezTCP)으로부터 알 수 없는 오류번호를 수신한 경우 *EZTCP_ERR_UNKNOWN*을 *error*에 저장합니다.

오류번호가 0보다 큰 경우에는 함수 실행 중 사용자 시스템에서 발생한 오류번호입니다.

Examples

```
//-----
// example of setting new password
//
struct eztcp_info out_eztcp_info;
int ret = get_eztcp_info_by_index(0, &out_eztcp_info);

if(ret == EZTCP_SUCCESS)
{
    char *new_pwd = "bbbb";
    unsigned char *mac_address = out_eztcp_info.mac_address;
    int err = 0;

    int res = EzTCP_ChangePassword(mac_address, NULL, new_pwd,
    out_eztcp_info.search_port, &err);
}
```

```

    if(res == EZTCP_SUCCESS)
    {
        printf("The requested operation has been successfully completed!\r\n");
    }
    else if(res == EZTCP_ERR)
    {
        if(err < 0)
        {
            printf("Changing the password has failed.[Error : %d]\r\n", err);
        }
        else if(err > 0)
        {
            LPVOID lpMsgBuf;

            FormatMessage(FORMAT_MESSAGE_ALLOCATE_BUFFER | FORMAT_MESSAGE_FROM_SYSTEM |
                FORMAT_MESSAGE_IGNORE_INSERTS, NULL, err, MAKELANGID(LANG_NEUTRAL, SUBLANG_DEFAULT),
                (LPSTR)&lpMsgBuf, 0, NULL);

            printf("Error[%d] %s\r\n", err, (char*)lpMsgBuf);
        }
    }
}

//-----

//-----
// example of changing current password
//
struct eztcp_info out_eztcp_info;
int ret = get_eztcp_info_by_index(0, &out_eztcp_info);

if(ret == EZTCP_SUCCESS)
{
    char *cur_pwd = "bbbb"
    char *new_pwd = "aaaa";
    unsigned char *mac_address = out_eztcp_info.mac_address;
    int err = 0;

    int res = EzTCP_ChangePassword(mac_address, cur_pwd, new_pwd,
        out_eztcp_info.search_port, &err);

    if(res == EZTCP_SUCCESS)
    {
        printf("The requested operation has been successfully completed!\r\n");
    }
    else if(res == EZTCP_ERR)

```

```

    {
        if(err < 0)
        {
            printf("Changing the password has failed.[Error : %d]\r\n", err);
        }
        else if(err > 0)
        {
            LPVOID lpMsgBuf;

            FormatMessage(FORMAT_MESSAGE_ALLOCATE_BUFFER | FORMAT_MESSAGE_FROM_SYSTEM |
                FORMAT_MESSAGE_IGNORE_INSERTS, NULL, err, MAKELANGID(LANG_NEUTRAL, SUBLANG_DEFAULT),
                (LPSTR)&lpMsgBuf, 0, NULL);

            printf("Error[%d] %s\r\n", err, (char*)lpMsgBuf);
        }
    }
}

//-----

//-----
// example of deleting current password
//
struct eztcp_info out_eztcp_info;
int ret = get_eztcp_info_by_index(0, &out_eztcp_info);

if(ret == EZTCP_SUCCESS)
{
    char *cur_pwd = "aaaa"
    unsigned char *mac_address = out_eztcp_info.mac_address;
    int err = 0;

    int res = EzTCP_ChangePassword(mac_address, cur_pwd, NULL,
        out_eztcp_info.search_port, &err);

    if(res == EZTCP_SUCCESS)
    {
        printf("The requested operation has been successfully completed!\r\n");
    }
    else if(res == EZTCP_ERR)
    {
        if(err < 0)
        {
            printf("Changing the password has failed.[Error : %d]\r\n", err);
        }
        else if(err > 0)

```

```
        {  
            LPVOID lpMsgBuf;  
            FormatMessage(FORMAT_MESSAGE_ALLOCATE_BUFFER | FORMAT_MESSAGE_FROM_SYSTEM |  
                FORMAT_MESSAGE_IGNORE_INSERTS, NULL, err, MAKELANGID(LANG_NEUTRAL, SUBLANG_DEFAULT),  
                (LPSTR)&lpMsgBuf, 0, NULL);  
            printf("Error[%d] %s\r\n", err, (char*)lpMsgBuf);  
        }  
    }  
}  
//-----
```

See also

Remarks

8 현재상태 보기

8.1 EzTCP_Status

```
int EzTCP_Status(unsigned char *mac_address, int udp_port_number, int *error);
```

Description

제품(ezTCP) 상태정보를 읽을 수 있습니다. 상태정보는 문자열로 되어있으며 그 길이는 제품의 종류에 따라서 다를 수 있습니다.

Parameters

*unsigned char *mac_address*

EzTCP_Search 또는 *EzTCP_Read*를 사용하여 검색된 제품(ezTCP)의 MAC주소 입니다.

int udp_port_number

EzTCP_Search 또는 *EzTCP_Read*로 제품(ezTCP)을 검색할 때 사용한 UDP 포트 번호 입니다. 0을 입력하면 기본값인 50005번, 50007번을 사용합니다.

*int *error*

함수 실행이 실패한 경우 오류 값이 저장됩니다.

Return values

*EzTCP_Status*의 반환 값은 다음과 같습니다.

EZTCP_SUCCESS

제품(ezTCP)의 상태정보를 성공적으로 읽으면 *EZTCP_SUCCESS*를 반환합니다.

EZTCP_ERR

EzTCP_Status 실행 중 오류가 발생한 경우 *EZTCP_ERR*를 반환합니다. 이 경우 *error*에 오류 번호를 저장합니다.

오류 번호가 0보다 작은 경우에는 아래와 같은 값을 가질 수 있습니다.

EZTCP_ERR_NO_INFO

*mac_address*을 가진 제품(ezTCP)을 찾을 수 없는 경우에는 *EZTCP_ERR_NO_INFO*를 *error*에 저장합니다.

EZTCP_ERR_RES

제품(ezTCP)으로부터 응답을 수신하지 못하면 *EZTCP_ERR_RES*를 *error*에 저장합니다.

EZTCP_ERR_UNKNOWN

제품(ezTCP)으로부터 알 수 없는 오류번호를 수신한 경우 *EZTCP_ERR_UNKNOWN*을 *error*에 저장합니다.

오류번호가 0보다 큰 경우에는 함수 실행 중 사용자 시스템에서 발생한 오류번호입니다.

Examples

See also

EzTCP_Status, get_status_string_length, get_status_string, get_tcpip4_session_count, get_tcpip6_session_count, get_tcpip4_session_info, get_tcpip6_session_info, EzTCP_CloseTcpIp4, EzTCP_CloseTcpIp6.

Remarks

8.2 get_status_string_length

```
int get_status_string_length(void);
```

Description

제품(ezTCP) 상태정보 문자열의 길이를 반환합니다. *EzTCP_Status* 실행이 성공한 경우에 사용하십시오.

Parameters

Return values

*get_status_string_length*는 제품(ezTCP) 상태정보 문자열의 길이를 반환합니다.

Examples

See also

EzTCP_Status, get_status_string_length, get_status_string, get_tcpip4_session_count, get_tcpip6_session_count, get_tcpip4_session_info, get_tcpip6_session_info, EzTCP_CloseTcpIp4, EzTCP_CloseTcpIp6.

Remarks

8.3 get_status_string

```
void get_status_string(char *buf);
```

Description

제품(ezTCP) 상태정보 문자열을 *buf*에 저장합니다. *EzTCP_Status* 실행이 성공한 경우에 사용하십시오.

Parameters

*char *buf*

제품(ezTCP) 상태정보가 저장될 포인터입니다.

Return values

Examples

```
struct eztcp_info out_eztcp_info;
int ret = get_eztcp_info_by_index(0, &out_eztcp_info);

if(ret == EZTCP_SUCCESS)
{
    char *buf;
    int err = 0;
    int status_len = 0;

    int res = EzTCP_Status(out_eztcp_info.mac_address, out_eztcp_info.search_port, &err);

    if(res == EZTCP_SUCCESS)
    {
        status_len = get_status_string_length();
        if(status_len > 0)
        {
            buf = (char*)malloc(status_len + 1);
            memset(buf, 0x00, status_len + 1);
            get_status_string(buf);
            printf("[Status]\r\n%s\r\n", buf);
            free(buf);
        }
    }
    else if(res == EZTCP_ERR)
    {
        if(err < 0)
        {
            printf("The requested operation has failed.[Error : %d]\r\n", err);
        }
        else if(err > 0)
        {
            LPVOID lpMsgBuf;

            FormatMessage(FORMAT_MESSAGE_ALLOCATE_BUFFER | FORMAT_MESSAGE_FROM_SYSTEM |
                FORMAT_MESSAGE_IGNORE_INSERTS, NULL, err, MAKELANGID(LANG_NEUTRAL, SUBLANG_DEFAULT),
                (LPSTR)&lpMsgBuf, 0, NULL);
        }
    }
}
```



```

        printf("Error[%d] %s\r\n", err, (char*)lpMsgBuf);
    }
}
}

```

See also

EzTCP_Status, get_status_string_length, get_status_string, get_tcpip4_session_count, get_tcpip6_session_count, get_tcpip4_session_info, get_tcpip6_session_info, EzTCP_CloseTcpIp4, EzTCP_CloseTcpIp6.

Remarks

*buf*는 *get_status_string_length*의 반환 값보다 큰 메모리 공간을 가지고 있어야 합니다.

8.4 get_tcpip4_session_count

```
int get_tcpip4_session_count(void);
```

Description

제품(ezTCP) 상태정보 중에서 제품이 가지고 있는 TCP/IP 버전4 접속 개 수를 반환합니다. *EzTCP_Status* 실행이 성공한 경우에 사용하십시오.

Parameters

Return values

*get_tcpip4_session_count*는 제품(ezTCP)이 가지고 있는 TCP/IP 버전4 접속 개 수를 반환합니다.

Examples

```

struct eztcp_info out_eztcp_info;
int ret = get_eztcp_info_by_index(0, &out_eztcp_info);

if(ret == EZTCP_SUCCESS)
{
    char buf[256];
    int err = 0;
    int idx;
    int tcpip4_count;

    int res = EzTCP_Status(out_eztcp_info.mac_address, out_eztcp_info.search_port, &err);

```

```

    if(res == EZTCP_SUCCESS)
    {
        tcpip4_count = get_tcpip4_session_count();
        for(idx = 0; idx < tcpip4_count; idx++)
        {
            if(get_tcpip4_session_info(idx, buf) == 1)
            {
                memset(buf, 0x00, 256);
                printf("[TCP/IP4] SESSION #d\r\n%s\r\n", idx, buf);
            }
        }
    }
    else if(res == EZTCP_ERR)
    {
        if(err < 0)
        {
            printf("The requested operation has failed.[Error : %d]\r\n", err);
        }
        else if(err > 0)
        {
            LPVOID lpMsgBuf;

            FormatMessage(FORMAT_MESSAGE_ALLOCATE_BUFFER | FORMAT_MESSAGE_FROM_SYSTEM |
                FORMAT_MESSAGE_IGNORE_INSERTS, NULL, err, MAKELANGID(LANG_NEUTRAL, SUBLANG_DEFAULT),
                (LPSTR)&lpMsgBuf, 0, NULL);

            printf("Error[%d] %s\r\n", err, (char*)lpMsgBuf);
        }
    }
}

```

See also

EzTCP_Status, get_status_string_length, get_status_string, get_tcpip4_session_count,
 get_tcpip6_session_count, get_tcpip4_session_info, get_tcpip6_session_info, EzTCP_CloseTcpIp4,
 EzTCP_CloseTcpIp6.

Remarks**8.5 get_tcpip6_session_count**

```
int get_tcpip6_session_count(void);
```

Description

제품(ezTCP) 상태정보 중에서 제품이 가지고 있는 TCP/IP 버전6 접속 개 수를 반환합니다.
EzTCP_Status 실행이 성공한 경우에 사용하십시오.

Parameters

Return values

*get_tcpip6_session_count*는 제품(ezTCP)이 가지고 있는 TCP/IP 버전6 접속 개 수를 반환합니다.

Examples

```
struct eztcp_info out_eztcp_info;
int ret = get_eztcp_info_by_index(0, &out_eztcp_info);

if(ret == EZTCP_SUCCESS)
{
    char buf[256];
    int err = 0;
    int idx;
    int tcpip6_count;

    int res = EzTCP_Status(out_eztcp_info.mac_address, out_eztcp_info.search_port, &err);

    if(res == EZTCP_SUCCESS)
    {
        tcpip6_count = get_tcpip6_session_count();

        for(idx = 0; idx < tcpip6_count; idx++)
        {
            if(get_tcpip6_session_info(idx, buf) == 1)
            {
                memset(buf, 0x00, 256);
                printf("[TCP/IP6] SESSION %d\r\n%s\r\n", idx, buf);
            }
        }
    }
    else if(res == EZTCP_ERR)
    {
        if(err < 0)
        {
            printf("The requested operation has failed.[Error : %d]\r\n", err);
        }
        else if(err > 0)
    }
```

```

        {
            LPVOID lpMsgBuf;

            FormatMessage(FORMAT_MESSAGE_ALLOCATE_BUFFER | FORMAT_MESSAGE_FROM_SYSTEM |
                FORMAT_MESSAGE_IGNORE_INSERTS, NULL, err, MAKELANGID(LANG_NEUTRAL, SUBLANG_DEFAULT),
                (LPSTR)&lpMsgBuf, 0, NULL);

            printf("Error[%d] %s\r\n", err, (char*)lpMsgBuf);

        }
    }
}

```

See also

EzTCP_Status, get_status_string_length, get_status_string, get_tcpip4_session_count,
get_tcpip6_session_count, get_tcpip4_session_info, get_tcpip6_session_info, EzTCP_CloseTcpIp4,
EzTCP_CloseTcpIp6.

Remarks

8.6 get_tcpip4_session_info

```
int get_tcpip4_session_info(int session_number, char *out_session_info);
```

Description

제품(ezTCP) 상태정보 중에서 *session number*가 가리키는 TCP/IP 버전4 접속정보를 *out_session_info*에 저장합니다. *EzTCP_Status* 실행이 성공한 경우에 사용하십시오.

Parameters

int session_number

TCP/IP 버전4 접속정보 순번으로 0부터 시작하며 *get_tcpip4_session_count*의 반환 값보다 크거나 같을 수 없습니다.

*char *out_session_info*

TCP/IP 버전4 접속정보가 저장될 포인터입니다. *out_session_info*의 크기는 256바이트 이상이어야 합니다.

Return values

*get_tcpip4_session_info*는 *out_session_info*에 TCP/IP4 버전4 접속정보를 저장하는데 성공하면 1을 반환하고 그렇지 않으면 0을 반환합니다.

Examples

```

struct eztcp_info out_eztcp_info;
int ret = get_eztcp_info_by_index(0, &out_eztcp_info);

```

```

if(ret == EZTCP_SUCCESS)
{
    char buf[256];
    int err = 0;
    int idx;
    int tcpip4_count;

    int res = EzTCP_Status(out_eztcp_info.mac_address, out_eztcp_info.search_port, &err);

    if(res == EZTCP_SUCCESS)
    {
        tcpip4_count = get_tcpip4_session_count();
        for(idx = 0; idx < tcpip4_count; idx++)
        {
            if(get_tcpip4_session_info(idx, buf) == 1)
            {
                memset(buf, 0x00, 256);
                printf("[TCP/IP4] SESSION #d\r\n%s\r\n", idx, buf);
            }
        }
    }
    else if(res == EZTCP_ERR)
    {
        if(err < 0)
        {
            printf("The requested operation has failed.[Error : %d]\r\n", err);
        }
        else if(err > 0)
        {
            LPVOID lpMsgBuf;

            FormatMessage(FORMAT_MESSAGE_ALLOCATE_BUFFER | FORMAT_MESSAGE_FROM_SYSTEM |
                FORMAT_MESSAGE_IGNORE_INSERTS, NULL, err, MAKELANGID(LANG_NEUTRAL, SUBLANG_DEFAULT),
                (LPSTR)&lpMsgBuf, 0, NULL);

            printf("Error[%d] %s\r\n", err, (char*)lpMsgBuf);
        }
    }
}

```

See also

EzTCP_Status, get_status_string_length, get_status_string, get_tcpip4_session_count,
 get_tcpip6_session_count, get_tcpip4_session_info, get_tcpip6_session_info, EzTCP_CloseTcpIp4,
 EzTCP_CloseTcpIp6.

Remarks

*out_session_info*의 크기는 256바이트 이상이 되어야합니다.

8.7 get_tcpip6_session_info

```
int get_tcpip6_session_info(int session_number, char *out_session_info);
```

Description

제품(ezTCP) 상태정보 중에서 *session_number*가 가리키는 TCP/IP 버전6 접속정보를 *out_session_info*에 저장합니다. *EzTCP_Status* 실행이 성공한 경우에 사용하십시오.

Parameters

int session_number

TCP/IP 버전6 접속정보 순번으로 0부터 시작하며 *get_tcpip6_session_count*의 반환 값보다 크거나 같을 수 없습니다.

*char *out_session_info*

TCP/IP 버전6 접속정보가 저장될 포인터입니다. *out_session_info*의 크기는 256바이트 이상이 되어야합니다.

Return values

*get_tcpip6_session_info*는 *out_session_info*에 TCP/IP4 버전6 접속정보를 저장하는데 성공하면 1을 반환하고 그렇지 않으면 0을 반환합니다.

Examples

```
struct eztcp_info out_eztcp_info;
int ret = get_eztcp_info_by_index(0, &out_eztcp_info);

if(ret == EZTCP_SUCCESS)
{
    char buf[256];
    int err = 0;
    int idx;
    int tcpip6_count;

    int res = EzTCP_Status(out_eztcp_info.mac_address, out_eztcp_info.search_port, &err);

    if(res == EZTCP_SUCCESS)
    {
        tcpip6_count = get_tcpip6_session_count();
    }
}
```

```

        for(idx = 0;idx < tcpip6_count;idx++)
        {
            if(get_tcpip6_session_info(idx, buf) == 1)
            {
                memset(buf, 0x00, 256);
                printf("[TCP/IP6] SESSION #d\r\n%s\r\n", idx, buf);
            }
        }
    }
    else if(res == EZTCP_ERR)
    {
        if(err < 0)
        {
            printf("The requested operation has failed.[Error : %d]\r\n", err);
        }
        else if(err > 0)
        {
            LPVOID lpMsgBuf;

            FormatMessage(FORMAT_MESSAGE_ALLOCATE_BUFFER | FORMAT_MESSAGE_FROM_SYSTEM |
            FORMAT_MESSAGE_IGNORE_INSERTS, NULL, err, MAKELANGID(LANG_NEUTRAL, SUBLANG_DEFAULT),
            (LPSTR)&lpMsgBuf, 0, NULL);

            printf("Error[%d] %s\r\n", err, (char*)lpMsgBuf);
        }
    }
}

```

See also

EzTCP_Status, get_status_string_length, get_status_string, get_tcpip4_session_count,
get_tcpip6_session_count, get_tcpip4_session_info, get_tcpip6_session_info, EzTCP_CloseTcpIp4,
EzTCP_CloseTcpIp6.

Remarks

*out_session_info*의 크기는 256바이트 이상이 되어야합니다.

8.8 EzTCP_CloseTcpIp4

```
int EzTCP_CloseTcpIp4(unsigned char *mac_address, int session_number,
                    char *password, int *error);
```

Description



제품(ezTCP)의 TCP/IP 버전4 접속 중에서 *session number*가 가리키는 접속의 상태가 접속완료인 경우 그 접속상태를 접속종료로 변경합니다. *EzTCP_Status* 실행이 성공한 경우에 사용하십시오.

Parameters

*unsigned char *mac_address*

EzTCP_Search 또는 *EzTCP_Read*를 사용하여 검색된 제품(ezTCP)의 MAC주소 입니다.

int session_number

TCP/IP 버전4 접속정보 순번으로 0부터 시작하며 *get_tcpip4_session_count*의 반환 값보다 크거나 같을 수 없습니다.

*char *password*

제품(ezTCP)에 비밀번호가 설정된 경우 현재 설정되어있는 비밀번호가 저장된 포인터입니다.

*int *error*

함수 실행이 실패한 경우 오류 값이 저장됩니다.

Return values

*EzTCP_CloseTcpIp4*의 반환 값은 다음과 같습니다.

EZTCP_SUCCESS

제품(ezTCP)에 설정 값을 성공적으로 저장하면 *EZTCP_SUCCESS*를 반환합니다.

EZTCP_ERR

EzTCP_CloseTcpIp4 실행 중 오류가 발생한 경우 *EZTCP_ERR*를 반환합니다. 이 경우 *error*에 오류 번호를 저장합니다.

오류 번호가 0보다 작은 경우에는 아래와 같은 값을 가질 수 있습니다.

EZTCP_ERR_NO_INFO

*mac_address*을 가진 제품(ezTCP)을 찾을 수 없는 경우에는 *EZTCP_ERR_NO_INFO*를 *error*에 저장합니다.

EZTCP_ERR_PWD_LENGTH

*password*에 입력한 비밀번호의 문자열 길이가 유효하지 않으면 *EZTCP_ERR_PWD_LENGTH*를 *error*에 저장합니다. 비밀번호는 4 ~ 8자입니다.

EZTCP_ERR_RES

제품(ezTCP)으로부터 응답을 수신하지 못하면 *EZTCP_ERR_RES*를 *error*에 저장합니다.

EZTCP_ERR_PWD

제품(ezTCP)에 설정된 비밀번호와 입력한 비밀번호가 일치하지 않으면 *EZTCP_ERR_PWD*를 *error*에 저장합니다.

EZTCP_ERR_UNKNOWN

제품(ezTCP)으로부터 알 수 없는 오류번호를 수신한 경우 *EZTCP_ERR_UNKNOWN*을 *error*에 저장합니다.

오류번호가 0보다 큰 경우에는 함수 실행 중 사용자 시스템에서 발생한 오류번호입니다.

Examples

```
struct eztcp_info out_eztcp_info;
int ret = get_eztcp_info_by_index(0, &out_eztcp_info);

if(ret == EZTCP_SUCCESS)
{
    char buf[256];
    int err = 0;
    int idx;
    int tcpip4_count;
    unsigned char *mac_address = out_eztcp_info.mac_address;

    int res = EzTCP_Status(mac_address, out_eztcp_info.search_port, &err);

    if(res == EZTCP_SUCCESS)
    {
        tcpip4_count = get_tcpip4_session_count();

        err = 0;

        res = EzTCP_CloseTcpIp4(mac_address, 0, NULL, &err);
        if(res == EZTCP_SUCCESS)
        {
            printf("Success.\r\n");
        }
        else if(res == EZTCP_ERR)
        {
            if(err < 0)
            {
                printf("The requested operation has failed.[Error : %d]\r\n", err);
            }
            else if(err > 0)
            {
                LPVOID lpMsgBuf;

                FormatMessage(FORMAT_MESSAGE_ALLOCATE_BUFFER |
                    FORMAT_MESSAGE_FROM_SYSTEM | FORMAT_MESSAGE_IGNORE_INSERTS, NULL, err,
                    MAKELANGID(LANG_NEUTRAL, SUBLANG_DEFAULT), (LPSTR)&lpMsgBuf, 0, NULL);
                printf("Error[%d] %s\r\n", err, (char*)lpMsgBuf);
            }
        }
    }
}
```

```
}
}
```

See also

EzTCP_Status, get_status_string_length, get_status_string, get_tcpip4_session_count, get_tcpip6_session_count, get_tcpip4_session_info, get_tcpip6_session_info, EzTCP_CloseTcpIp4, EzTCP_CloseTcpIp6.

Remarks

8.9 EzTCP_CloseTcpIp6

```
int EzTCP_CloseTcpIp6(unsigned char *mac_address, int session_number,
                      char *password, int *error);
```

Description

제품(ezTCP)의 TCP/IP 버전6 접속 중에서 *session number*가 가리키는 접속의 상태가 접속완료인 경우 그 접속상태를 접속종료로 변경합니다. *EzTCP_Status* 실행이 성공한 경우에 사용하십시오.

Parameters

*unsigned char *mac_address*

EzTCP_Search 또는 *EzTCP_Read*를 사용하여 검색된 제품(ezTCP)의 MAC주소 입니다.

int session_number

TCP/IP 버전6 접속정보 순번으로 0부터 시작하며 *get_tcpip6_session_count*의 반환 값보다 크거나 같을 수 없습니다.

*char *password*

제품(ezTCP)에 비밀번호가 설정된 경우 현재 설정되어있는 비밀번호가 저장된 포인터입니다.

*int *error*

함수 실행이 실패한 경우 오류 값이 저장됩니다.

Return values

*EzTCP_CloseTcpIp6*의 반환 값은 다음과 같습니다.

EZTCP_SUCCESS

제품(ezTCP)에 설정 값을 성공적으로 저장하면 *EZTCP_SUCCESS*를 반환합니다.

EZTCP_ERR

*EzTCP_CloseTcpIp6*실행 중 오류가 발생한 경우 *EZTCP_ERR*을 반환합니다. 이 경우

*error*에 오류 번호를 저장합니다.

오류 번호가 0보다 작은 경우에는 아래와 같은 값을 가질 수 있습니다.

EZTCP_ERR_NO_INFO

*mac_address*을 가진 제품(ezTCP)을 찾을 수 없는 경우에는 *EZTCP_ERR_NO_INFO*를 *error*에 저장합니다.

EZTCP_ERR_PWD_LENGTH

*password*에 입력한 비밀번호의 문자열 길이가 유효하지 않으면 *EZTCP_ERR_PWD_LENGTH*를 *error*에 저장합니다. 비밀번호는 4~8자입니다.

EZTCP_ERR_RES

제품(ezTCP)으로부터 응답을 수신하지 못하면 *EZTCP_ERR_RES*를 *error*에 저장합니다.

EZTCP_ERR_PWD

제품(ezTCP)에 설정된 비밀번호와 입력한 비밀번호가 일치하지 않으면 *EZTCP_ERR_PWD*를 *error*에 저장합니다.

EZTCP_ERR_UNKNOWN

제품(ezTCP)으로부터 알 수 없는 오류번호를 수신한 경우 *EZTCP_ERR_UNKNOWN*을 *error*에 저장합니다.

오류번호가 0보다 큰 경우에는 함수 실행 중 사용자 시스템에서 발생한 오류번호입니다.

Examples

```
struct eztcp_info out_eztcp_info;
int ret = get_eztcp_info_by_index(0, &out_eztcp_info);

if(ret == EZTCP_SUCCESS)
{
    char buf[256];
    int err = 0;
    int idx;
    int tcpip6_count;
    unsigned char *mac_address = out_eztcp_info.mac_address;

    int res = EzTCP_Status(mac_address, out_eztcp_info.search_port, &err);

    if(res == EZTCP_SUCCESS)
    {
        tcpip6_count = get_tcpip6_session_count();

        err = 0;

        res = EzTCP_CloseTcpIp6(mac_address, 0, NULL, &err);
        if(res == EZTCP_SUCCESS)
```

```

        {
            printf("Success.\r\n");
        }
        else if(res == EZTCP_ERR)
        {
            if(err < 0)
            {
                printf("The requested operation has failed.[Error : %d]\r\n", err);
            }
            else if(err > 0)
            {
                LPVOID lpMsgBuf;

                FormatMessage(FORMAT_MESSAGE_ALLOCATE_BUFFER |
                    FORMAT_MESSAGE_FROM_SYSTEM | FORMAT_MESSAGE_IGNORE_INSERTS, NULL, err,
                    MAKELANGID(LANG_NEUTRAL, SUBLANG_DEFAULT), (LPSTR)&lpMsgBuf, 0, NULL);

                printf("Error[%d] %s\r\n", err, (char*)lpMsgBuf);
            }
        }
    }
}

```

See also

EzTCP_Status, get_status_string_length, get_status_string, get_tcpip4_session_count,
 get_tcpip6_session_count, get_tcpip4_session_info, get_tcpip6_session_info, EzTCP_CloseTcpIp4,
 EzTCP_CloseTcpIp6.

Remarks

9 인증서

9.1 EzTCP_ReadCert

```
int EzTCP_ReadCert(unsigned char *mac_address, int udp_port_number, int *error);
```

Description

제품(ezTCP)에 저장된 인증서 정보를 읽을 수 있습니다. 인증서 정보는 문자열로 되어있으며 그 길이는 인증서에 따라서 다를 수 있습니다.

Parameters

*unsigned char *mac_address*

EzTCP_Search 또는 *EzTCP_Read*를 사용하여 검색된 제품(ezTCP)의 MAC주소 입니다.

int udp_port_number

EzTCP_Search 또는 *EzTCP_Read*로 제품(ezTCP)을 검색할 때 사용한 UDP 포트 번호 입니다. 0을 입력하면 기본값인 50005번, 50007번을 사용합니다.

*int *error*

함수 실행이 실패한 경우 오류 값이 저장됩니다.

Return values

*EzTCP_ReadCert*의 반환 값은 다음과 같습니다.

EZTCP_SUCCESS

제품(ezTCP)의 상태정보를 성공적으로 읽으면 EZTCP_SUCCESS를 반환합니다.

EZTCP_ERR

EzTCP_ReadCert 실행 중 오류가 발생한 경우 EZTCP_ERR를 반환합니다.

이 경우 *error*에 오류 번호를 저장합니다. 오류 번호가 0보다 작은 경우 아래와 같은 값을 가질 수 있습니다.

EZTCP_ERR_NO_INFO

*mac_address*을 가진 제품(ezTCP)을 찾을 수 없는 경우에는 EZTCP_ERR_NO_INFO를 *error*에 저장합니다.

EZTCP_ERR_RES

제품(ezTCP)으로부터 응답을 수신하지 못하면 EZTCP_ERR_RES를 *error*에 저장합니다.

EZTCP_ERR_UNKNOWN

제품(ezTCP)으로부터 알 수 없는 오류번호를 수신한 경우 EZTCP_ERR_UNKNOWN을 *error*에 저장합니다.

오류번호가 0보다 큰 경우에는 함수 실행 중 사용자 시스템에서 발생한 오류번호입니다.

Examples

See also

is_certificates, EzTCP_ReadCert, get_certificates_string_length, get_certificates_string,
EzTCP_WriteSelfSignedCertificates, EzTCP_WriteCertificates.

Remarks

9.2 get_certificates_string_length

```
int get_certificates_string_length(void);
```

Description

제품(ezTCP)에 저장된 인증서 정보 문자열의 길이를 반환합니다. *EzTCP_ReadCert* 실행이 성공한 경우에 사용하십시오.

Parameters

Return values

*get_certificates_string_length*는 제품(ezTCP) 인증서 정보 문자열의 길이를 반환합니다.

Examples

See also

is_certificates, EzTCP_ReadCert, get_certificates_string_length, get_certificates_string,
EzTCP_WriteSelfSignedCertificates, EzTCP_WriteCertificates.

Remarks

9.3 get_certificates_string

```
void get_certificates_string(char *buf);
```

Description

제품(ezTCP) 인증서 정보 문자열을 *buf*에 저장합니다. *EzTCP_ReadCert* 실행이 성공한 경우에 사용하십시오.

Parameters

*char *buf*

제품(ezTCP) 인증서 정보가 저장될 포인터입니다.

Return values

Examples

```
struct eztcp_info out_eztcp_info;
int ret = get_eztcp_info_by_index(0, &out_eztcp_info);

if(ret == EZTCP_SUCCESS)
{
    char *buf;
    int err = 0;
    int certificates_len = 0;

    int res = EzTCP_ReadCert(out_eztcp_info.mac_address, out_eztcp_info.search_port,
&err);

    if(res == EZTCP_SUCCESS)
    {
        certificates_len = get_certificates_string_length();
        if(certificates_len > 0)
        {
            buf = (char*)malloc(certificates_len + 1);
            memset(buf, 0x00, certificates_len + 1);
            get_certificates_string(buf);
            printf("[Certificates]\r\n%s\r\n", buf);
            free(buf);
        }
    }
    else if(res == EZTCP_ERR)
    {
        if(err < 0)
        {
            printf("The requested operation has failed.[Error : %d]\r\n", err);
        }
        else if(err > 0)
        {
            LPVOID lpMsgBuf;
```

```

        FormatMessage(FORMAT_MESSAGE_ALLOCATE_BUFFER | FORMAT_MESSAGE_FROM_SYSTEM |
FORMAT_MESSAGE_IGNORE_INSERTS, NULL, err, MAKELANGID(LANG_NEUTRAL, SUBLANG_DEFAULT),
(LPSTR)&lpmMsgBuf, 0, NULL);

        printf("Error[%d] %s\r\n", err, (char*)lpmMsgBuf);

    }

}
}

```

See also

is_certificates, EzTCP_ReadCert, get_certificates_string_length, get_certificates_string,
EzTCP_WriteSelfSignedCertificates, EzTCP_WriteCertificates.

Remarks

9.4 EzTCP_WriteSelfSignedCertificates

```

int EzTCP_WriteSelfSignedCertificates(unsigned char *mac_address, int rsa_key_length,
char *country_code, char *state_name, char *locality_name, char *organization_name,
char *organization_unit_name, char *common_name, char *email, int udp_port_number,
int *error);

```

Description

제품(ezTCP)에 인증서를 직접 만든 후 저장합니다.

Parameters

*unsigned char *mac_address*

EzTCP_Search 또는 *EzTCP_Read*를 사용하여 검색된 제품(ezTCP)의 MAC주소 입니다.

int rsa_key_length

RSA 키 길이입니다. 사용할 수 있는 값은 1024, 1536, 2048입니다.

*char *country_code*

2자리 코드의 국가 이름입니다.

*char *state_name*

주 이름입니다.

*char *locality_name*

지역명 또는 도시이름입니다.

*char *organization_name*

조직명입니다. 회사이름 같은 것을 사용합니다.

*char *organization_unit_name*

조직 단위 이름입니다. 부서명 같은 것을 사용합니다.

*char *common_name*

일반 이름입니다. 서버 주소 같은 것을 사용합니다.

*char *email*

메일주소입니다.

int udp_port_number

EzTCP_Search 또는 *EzTCP_Read*로 제품(ezTCP)을 검색할 때 사용한 UDP 포트 번호입니다. 0을 입력하면 기본값인 50005번, 50007번을 사용합니다.

*int *error*

함수 실행이 실패한 경우 오류 값이 저장됩니다.

Return values

*EzTCP_WriteSelfSignedCertificates*의 반환 값은 다음과 같습니다.

EZTCP_SUCCESS

제품(ezTCP)의 상태정보를 성공적으로 읽으면 *EZTCP_SUCCESS*를 반환합니다.

EZTCP_ERR

EzTCP_WriteSelfSignedCertificates 실행 중 오류가 발생한 경우 *EZTCP_ERR*을 반환합니다.

이 경우 *error*에 오류 번호를 저장합니다. 오류 번호가 0보다 작은 경우 아래와 같은 값을 가질 수 있습니다.

EZTCP_ERR_NO_INFO

*mac_address*을 가진 제품(ezTCP)을 찾을 수 없는 경우에는 *EZTCP_ERR_NO_INFO*를 *error*에 저장합니다.

EZTCP_ERR_RES

제품(ezTCP)으로부터 응답을 수신하지 못하면 *EZTCP_ERR_RES*를 *error*에 저장합니다.

EZTCP_ERR_UNKNOWN

제품(ezTCP)으로부터 알 수 없는 오류번호를 수신한 경우 *EZTCP_ERR_UNKNOWN*을 *error*에 저장합니다.

오류번호가 0보다 큰 경우에는 함수 실행 중 사용자 시스템에서 발생한 오류번호입니다.

Examples

See also

is_certificates, *EzTCP_ReadCert*, *get_certificates_string_length*, *get_certificates_string*, *EzTCP_WriteSelfSignedCertificates*, *EzTCP_WriteCertificates*.

Remarks

9.5 EzTCP_WriteCertificates

```
int EzTCP_WriteCertificates(unsigned char *mac_address, char *certificates_filepath,
char *certificates_password, char *rsa_key_filepath, char *rsa_key_password, int
udp_port_number, int *error);
```

Description

제품(ezTCP)에 타 인증기관에서 발급받은 인증서를 저장합니다.

Parameters

*unsigned char *mac_address*

EzTCP_Search 또는 *EzTCP_Read*를 사용하여 검색된 제품(ezTCP)의 MAC주소 입니다.

*char *certificates_filepath*

인증서 파일이 있는 경로입니다.

*char *certificates_password*

인증서 파일에 암호가 설정된 경우 그 암호를 입력하세요.

*char *ras_key_filepath*

인증서 파일에 키가 같이 포함 안 된 경우 키 파일이 있는 경로입니다. 필수 항목은 아닙니다.

*char *rsa_key_password*

키 파일에 암호가 설정된 경우 그 암호를 입력하세요.

int udp_port_number

EzTCP_Search 또는 *EzTCP_Read*로 제품(ezTCP)을 검색할 때 사용한 UDP 포트 번호 입니다. 0을 입력하면 기본값인 50005번, 50007번을 사용합니다.

*int *error*

함수 실행이 실패한 경우 오류 값이 저장됩니다.

Return values

*EzTCP_WriteCertificates*의 반환 값은 다음과 같습니다.

EZTCP_SUCCESS

제품(ezTCP)의 상태정보를 성공적으로 읽으면 *EZTCP_SUCCESS*를 반환합니다.

EZTCP_ERR

*EzTCP_WriteCertificates*실행 중 오류가 발생한 경우 *EZTCP_ERR*을 반환합니다.

이 경우 *error*에 오류 번호를 저장합니다. 오류 번호가 0보다 작은 경우 아래와 같은 값을 가질 수 있습니다.

EZTCP_ERR_NO_INFO

*mac_address*을 가진 제품(ezTCP)을 찾을 수 없는 경우에는 *EZTCP_ERR_NO_INFO*를 *error*에 저장합니다.

EZTCP_ERR_RES

제품(ezTCP)으로부터 응답을 수신하지 못하면 EZTCP_ERR_RES를 *error*에 저장합니다.

EZTCP_ERR_UNKNOWN

제품(ezTCP)으로부터 알 수 없는 오류번호를 수신한 경우 EZTCP_ERR_UNKNOWN을 *error*에 저장합니다.

EZTCP_ERR_FOPEN

파일을 찾을 수 없거나 읽을 수 없을 때 EZTCP_ERR_FOPEN을 *error*에 저장합니다.

EZTCP_ERR_READ_CERT

인증서를 분석할 수 없으면 EZTCP_ERR_READ_CERT를 *error*에 저장합니다.

EZTCP_ERR_NO_RSA_KEY

RSA 키를 찾을 수 없으면 EZTCP_ERR_NO_RSA_KEY를 *error*에 저장합니다.

EZTCP_ERR_NO_CERT

인증서를 찾을 수 없으면 EZTCP_ERR_NO_CERT를 *error*에 저장합니다.

EZTCP_ERR_READ_RSA_KEY

키를 분석할 수 없으면 EZTCP_ERR_READ_RAS_KEY를 *error*에 저장합니다.

오류번호가 0보다 큰 경우에는 함수 실행 중 사용자 시스템에서 발생한 오류번호입니다.

Examples**See also**

is_certificates, EzTCP_ReadCert, get_certificates_string_length, get_certificates_string,
EzTCP_WriteSelfSignedCertificates, EzTCP_WriteCertificates.

Remarks

10 기타

10.1 get_version

```
int get_version(unsigned char *mac_address, char *out_version);
```

Description

*get_version*은 제품(ezTCP)의 펌웨어 버전을 반환합니다.

Parameters

*unsigned char *mac_address*

EzTCP_Search 또는 *EzTCP_Read*를 사용하여 검색된 제품(ezTCP)의 MAC주소 입니다.

*char *out_version*

펌웨어 버전이 저장될 포인터입니다.

Return values

*get_version*은 *out_version*에 제품(ezTCP)의 펌웨어 버전을 저장하는데 성공하면 1을 반환합니다.

*mac_address*을 가진 제품(ezTCP)을 찾을 수 없는 경우에는 -1을 반환합니다.

Examples

```
struct eztcp_info out_eztcp_info;
int ret = get_eztcp_info_by_index(0, &out_eztcp_info);

if(ret == EZTCP_SUCCESS)
{
    char buf[8];
    memset(buf, 0x00, 8);
    if(get_version(out_eztcp_info.mac_address, buf) == 1)
    {
        printf("Firmware version : %s\r\n", buf);
    }
}
```

See also

Remarks

펌웨어 버전은 1.1A와 같이 숫자로 된 주 번호 1자리, 부 번호 1자리 그리고 영문으로 된 개정번호 1자리를 사용합니다.

10.2 is_ip6

```
int is_ip6(unsigned char *mac_address);
```

Description

*is_ip6*는 제품(ezTCP)이 TCP/IPv6 기능을 사용할 수 있는지 확인합니다.

Parameters

*unsigned char *mac_address*

EzTCP_Search 또는 *EzTCP_Read*를 사용하여 검색된 제품(ezTCP)의 MAC주소 입니다.

Return values

*is_ip6*는 제품(ezTCP)이 TCP/IPv6 기능을 사용할 수 있으면 1을 반환하고 그렇지 않으면 0을 반환합니다.

*mac_address*을 가진 제품(ezTCP)을 찾을 수 없는 경우에는 -1을 반환합니다.

Examples

See also

Remarks

*is_ip6*의 반환값은 다른 설정 값에 따라 달라질 수 있습니다.

10.3 is_uart_rs232

```
int is_uart_rs232(unsigned char *mac_address, int uart_index);
```

Description

*is_uart_rs232*는 *uart_index*로 지정된 시리얼 포트가 [시리얼 종류]로 RS-232를 사용할 수 있는지 확인합니다.

Parameters

*unsigned char *mac_address*

EzTCP_Search 또는 *EzTCP_Read*를 사용하여 검색된 제품(ezTCP)의 MAC주소 입니다.

int uart_index

제품(ezTCP)의 시리얼 포트 순번으로 0부터 시작합니다.

Return values

*is_uart_rs232*는 제품(ezTCP)이 RS-232를 사용할 수 있으면 1을 반환하고 그렇지 않으면 0을 반환합니다.

*mac_address*을 가진 제품(ezTCP)을 찾을 수 없는 경우에는 -1을 반환합니다.

Examples

See also

Remarks

10.4 is_uart_rs485

```
int is_uart_rs485(unsigned char *mac_address, int uart_index);
```

Description

*is_uart_rs485*는 *uart_index*로 지정된 시리얼 포트가 [시리얼 종류]로 RS-485를 사용할 수 있는지 확인합니다.

Parameters

*unsigned char *mac_address*

EzTCP_Search 또는 *EzTCP_Read*를 사용하여 검색된 제품(ezTCP)의 MAC주소 입니다.

int uart_index

제품(ezTCP)의 시리얼 포트 순번으로 0부터 시작합니다.

Return values

*is_uart_rs485*는 *uart_index*로 지정된 시리얼 포트가 RS-485를 사용할 수 있으면 1을 반환하고 그렇지 않으면 0을 반환합니다.

*mac_address*을 가진 제품(ezTCP)을 찾을 수 없는 경우에는 -1을 반환합니다.

Examples

See also

Remarks

*is_rs485*의 반환값은 다른 설정 값에 따라 달라질 수 있습니다.

10.5 is_uart_rs422

```
int is_uart_rs422(unsigned char *mac_address, int uart_index);
```

Description

*is_uart_rs422*는 *uart_index*로 지정된 시리얼 포트가 [시리얼 종류]로 RS-422을 사용할 수 있는지 확인합니다.

Parameters

*unsigned char *mac_address*

EzTCP_Search 또는 *EzTCP_Read*를 사용하여 검색된 제품(ezTCP)의 MAC주소 입니다.

int uart_index

제품(ezTCP)의 시리얼 포트 순번으로 0부터 시작합니다.

Return values

*is_uart_rs422*는 *uart_index*로 지정된 시리얼 포트가 RS-422를 사용할 수 있으면 1을 반환하고 그렇지 않으면 0을 반환합니다.

*mac_address*을 가진 제품(ezTCP)을 찾을 수 없는 경우에는 -1을 반환합니다.

Examples

See also

Remarks

*is_rs422*의 반환값은 다른 설정 값에 따라 달라질 수 있습니다.

10.6 is_uart_ttl

```
int is_uart_ttl(unsigned char *mac_address);
```

Description

*is_uart_ttl*은 제품(ezTCP) 시리얼 포트가 [TTL] 출력 기능을 사용할 수 있는지 확인합니다.

Parameters

*unsigned char *mac_address*

EzTCP_Search 또는 *EzTCP_Read*를 사용하여 검색된 제품(ezTCP)의 MAC주소 입니다.

Return values

*is_uart_ttl*은 제품(ezTCP) 시리얼 포트가 [TTL] 출력 기능을 사용할 수 있으면 1을 반환하고 그렇지 않으면 0을 반환합니다.

*mac_address*을 가진 제품(ezTCP)을 찾을 수 없는 경우에는 -1을 반환합니다.

Examples

See also

Remarks

10.7 is_uart_8_databit_only

```
int is_uart_8_databit_only(unsigned char *mac_address);
```

Description

*is_uart_8_databit_only*는 제품(ezTCP) 시리얼 포트가 [데이터 비트]로 8비트만 사용 가능한지 확인합니다.

Parameters

*unsigned char *mac_address*

EzTCP_Search 또는 *EzTCP_Read*를 사용하여 검색된 제품(ezTCP)의 MAC주소 입니다.

Return values

*is_uart_8_databit_only*는 제품(ezTCP)이 시리얼 포트의 데이터 비트로 8비트만 사용 가능할 경우 1을 반환하고 그렇지 않으면 0을 반환합니다.

*mac_address*을 가진 제품(ezTCP)을 찾을 수 없는 경우에는 -1을 반환합니다.

Examples

See also

Remarks

10.8 is_uart_7_and_8_databit_only

```
int is_uart_7_and_8_databit_only(unsigned char *mac_address);
```

Description



*is_uart_7_and_8_databit_only*는 제품(ezTCP)이 시리얼 포트의 [데이터 비트]로 7비트와 8비트만 사용 가능한지 확인합니다.

Parameters

*unsigned char *mac_address*

EzTCP_Search 또는 *EzTCP_Read*를 사용하여 검색된 제품(ezTCP)의 MAC주소 입니다.

Return values

*is_uart_7_and_8_databit_only*는 제품(ezTCP)이 시리얼 포트의 [데이터 비트]로 7비트와 8비트만 사용 가능할 경우 1을 반환하고 그렇지 않으면 0을 반환합니다.

*mac_address*을 가진 제품(ezTCP)을 찾을 수 없는 경우에는 -1을 반환합니다.

Examples

See also

Remarks

10.9 is_uart_one5_stopbit

```
int is_uart_one5_stopbit(unsigned char *mac_address);
```

Description

*is_uart_one5_stopbit*는 제품(ezTCP) 시리얼 포트가 [정지 비트]로 1.5비트를 사용할 수 있는지 확인합니다.

Parameters

*unsigned char *mac_address*

EzTCP_Search 또는 *EzTCP_Read*를 사용하여 검색된 제품(ezTCP)의 MAC주소 입니다.

Return values

*is_uart_one5_stopbit*는 제품(ezTCP)이 시리얼 포트의 [정지 비트]로 1.5비트를 사용할 수 있으면 1을 반환하고 그렇지 않으면 0을 반환합니다.

*mac_address*을 가진 제품(ezTCP)을 찾을 수 없는 경우에는 -1을 반환합니다.

Examples

See also

Remarks**10.10 is_uart_dtrdsr**

```
int is_uart_dtrdsr(unsigned char *mac_address);
```

Description

*is_uart_dtrdsr*은 제품(ezTCP) 시리얼 포트가 [DTR/DSR] 기능을 사용할 수 있는지 확인합니다.

Parameters

*unsigned char *mac_address*

EzTCP_Search 또는 *EzTCP_Read*를 사용하여 검색된 제품(ezTCP)의 MAC주소입니다.

Return values

*is_uart_dtrdsr*은 제품(ezTCP) 시리얼 포트가 [DTR/DSR] 기능을 사용할 수 있으면 1을 반환하고 그렇지 않으면 0을 반환합니다.

*mac_address*을 가진 제품(ezTCP)을 찾을 수 없는 경우에는 -1을 반환합니다.

Examples**See also****Remarks****10.11 is_uart_xonxoff**

```
int is_uart_xonxoff(unsigned char *mac_address);
```

Description

*is_uart_xonxoff*는 제품(ezTCP) 시리얼 포트가 [흐름 제어]로 XON/XOFF를 사용할 수 있는지 확인합니다.

Parameters

*unsigned char *mac_address*

EzTCP_Search 또는 *EzTCP_Read*를 사용하여 검색된 제품(ezTCP)의 MAC주소입니다.

Return values

*is_uart_xonxoff*는 제품(ezTCP) 시리얼 포트가 [흐름 제어]로 XON/XOFF를 사용할 수 있으면 1을 반환하고 그렇지 않으면 0을 반환합니다.

*mac_address*을 가진 제품(ezTCP)을 찾을 수 없는 경우에는 -1을 반환합니다.

Examples**See also****Remarks****10.12 is_uart_tx_delay**

```
int is_uart_tx_delay(unsigned char *mac_address);
```

Description

*is_uart_tx_delay*는 제품(ezTCP) 시리얼 포트가 [데이터 전송 간격] 기능을 사용할 수 있는지 확인합니다.

Parameters

*unsigned char *mac_address*

EzTCP_Search 또는 *EzTCP_Read*를 사용하여 검색된 제품(ezTCP)의 MAC주소입니다.

Return values

*is_uart_tx_delay*는 제품(ezTCP) 시리얼 포트가 [데이터 전송 간격]을 사용할 수 있으면 1을 반환하고 그렇지 않으면 0을 반환합니다.

*mac_address*을 가진 제품(ezTCP)을 찾을 수 없는 경우에는 -1을 반환합니다.

Examples**See also****Remarks****10.13 is_uart_data_frame_interval**

```
int is_uart_data_frame_interval(unsigned char *mac_address);
```

Description

*is_uart_data_frame_interval*은 제품(ezTCP) 시리얼 포트가 [데이터 프레임 간격(10ms)] 기능을 사용할 수 있는지 확인합니다.

Parameters

*unsigned char *mac_address*

EzTCP_Search 또는 *EzTCP_Read*를 사용하여 검색된 제품(ezTCP)의 MAC주소 입니다.

Return values

*is_uart_data_frame_interval*은 제품(ezTCP) 시리얼 포트가 [데이터 프레임 간격(10ms)] 기능을 사용할 수 있으면 1을 반환하고 그렇지 않으면 0을 반환합니다.

*mac_address*을 가진 제품(ezTCP)을 찾을 수 없는 경우에는 -1을 반환합니다.

Examples**See also****Remarks****10.14 is_uart_separator**

```
int is_uart_separator(unsigned char *mac_address);
```

Description

*is_uart_separator*는 제품(ezTCP) 시리얼 포트가 구분자 기능을 사용할 수 있는지 확인합니다.

Parameters

*unsigned char *mac_address*

EzTCP_Search 또는 *EzTCP_Read*를 사용하여 검색된 제품(ezTCP)의 MAC주소 입니다.

Return values

*is_uart_separator*는 제품(ezTCP) 시리얼 포트가 구분자 기능을 사용할 수 있으면 1을 반환하고 그렇지 않으면 0을 반환합니다.

*mac_address*을 가진 제품(ezTCP)을 찾을 수 없는 경우에는 -1을 반환합니다.

Examples**See also**

Remarks**10.15 is_uart_tcp_nodelay**

```
int is_uart_tcp_nodelay(unsigned char *mac_address, int uart_index);
```

Description

*is_uart_tcp_nodelay*는 제품(ezTCP) 시리얼 포트가 [전송지연 기능 사용안함] 기능을 사용할 수 있는지 확인합니다.

Parameters

*unsigned char *mac_address*

EzTCP_Search 또는 *EzTCP_Read*를 사용하여 검색된 제품(ezTCP)의 MAC주소 입니다.

Return values

*is_uart_tcp_nodelay*는 제품(ezTCP) 시리얼 포트가 [전송지연 기능 사용안함] 기능을 사용할 수 있으면 1을 반환하고 그렇지 않으면 0을 반환합니다.

*mac_address*을 가진 제품(ezTCP)을 찾을 수 없는 경우에는 -1을 반환합니다.

Examples**See also****Remarks**

[전송지연 기능 사용안함] 기능의 사용 가능 여부는 다른 설정 값에 따라 달라질 수 있습니다.

10.16 is_uart_rfc2217

```
int is_uart_rfc2217(unsigned char *mac_address, int uart_index);
```

Description

*is_uart_rfc2217*은 제품(ezTCP) 시리얼 포트가 [시리얼 포트 설정/상태 전송(RFC2217)] 기능을 사용할 수 있는지 확인합니다.

Parameters

*unsigned char *mac_address*

EzTCP_Search 또는 *EzTCP_Read*를 사용하여 검색된 제품(ezTCP)의 MAC주소 입니다.

Return values

*is_uart_rfc2217*은 제품(ezTCP) 시리얼 포트가 [시리얼 포트 설정/상태 전송(RFC2217)] 기능을 사용할 수 있으면 1을 반환하고 그렇지 않으면 0을 반환합니다.

*mac_address*을 가진 제품(ezTCP)을 찾을 수 없는 경우에는 -1을 반환합니다.

Examples

See also

Remarks

[시리얼 포트 설정/상태 전송(RFC2217)] 기능의 사용 가능 여부는 다른 설정 값에 따라 달라질 수 있습니다.

10.17 is_uart_protocol

```
int is_uart_protocol(unsigned char *mac_address);
```

Description

*is_uart_protocol*은 제품(ezTCP) 시리얼 포트가 [프로토콜] 기능을 사용할 수 있는지 확인합니다.

Parameters

*unsigned char *mac_address*

EzTCP_Search 또는 *EzTCP_Read*를 사용하여 검색된 제품(ezTCP)의 MAC주소 입니다.

Return values

*is_uart_protocol*은 제품(ezTCP) 시리얼 포트가 [프로토콜] 기능을 사용할 수 있으면 1을 반환하고 그렇지 않으면 0을 반환합니다.

*mac_address*을 가진 제품(ezTCP)을 찾을 수 없는 경우에는 -1을 반환합니다.

Examples

See also

Remarks

현재 CSE-T16, CSE-T32, CSE-T48만 사용 가능하며 추후 지원가능 모델은 추가 될 수 있습니다. 보다 자세한 내용은 해당 제품 사용설명서를 참조하시기 바랍니다.

10.18 is_uart_tcp_server

```
int is_uart_tcp_server(unsigned char *mac_address, int uart_index);
```

Description

*is_uart_tcp_server*는 *uart_index*로 지정된 시리얼 포트가 [통신모드]로 T2S - TCP 서버를 사용할 수 있는지 확인합니다.

Parameters

*unsigned char *mac_address*

EzTCP_Search 또는 *EzTCP_Read*를 사용하여 검색된 제품(ezTCP)의 MAC주소 입니다.

int uart_index

제품(ezTCP)의 시리얼 포트 순번으로 0부터 시작합니다.

Return values

*is_uart_tcp_server*는 *uart_index*로 지정된 시리얼 포트가 [통신모드]로 T2S - TCP 서버를 사용할 수 있으면 1을 반환하고 그렇지 않으면 0을 반환합니다.

*mac_address*을 가진 제품(ezTCP)을 찾을 수 없는 경우에는 -1을 반환합니다.

Examples

See also

Remarks

*is_uart_tcp_server*의 반환값은 다른 설정 값에 따라 달라질 수 있습니다.

10.19 is_uart_at_command

```
int is_uart_at_command(unsigned char *mac_address, int uart_index);
```

Description

*is_uart_at_command*는 *uart_index*로 지정된 시리얼 포트가 [통신모드]로 ATC - AT 명령을 사용할 수 있는지 확인합니다.

Parameters

*unsigned char *mac_address*

EzTCP_Search 또는 *EzTCP_Read*를 사용하여 검색된 제품(ezTCP)의 MAC주소 입니다.

int uart_index

제품(ezTCP)의 시리얼 포트 순번으로 0부터 시작합니다.

Return values

*is_uart_at_command*는 *uart_index*로 지정된 시리얼 포트가 [통신모드]로 ATC – AT 명령을 사용할 수 있으면 1을 반환하고 그렇지 않으면 0을 반환합니다.

*mac_address*을 가진 제품(ezTCP)을 찾을 수 없는 경우에는 -1을 반환합니다.

Examples

See also

Remarks

*is_uart_at_command*의 반환값은 다른 설정 값에 따라 달라질 수 있습니다.

10.20 is_uart_tcp_client

```
int is_uart_tcp_client(unsigned char *mac_address, int uart_index);
```

Description

*is_uart_tcp_client*는 *uart_index*로 지정된 시리얼 포트가 [통신모드]로 COD – TCP 클라이언트를 사용할 수 있는지 확인합니다.

Parameters

*unsigned char *mac_address*

EzTCP_Search 또는 *EzTCP_Read*를 사용하여 검색된 제품(ezTCP)의 MAC주소 입니다.

int uart_index

제품(ezTCP)의 시리얼 포트 순번으로 0부터 시작합니다.

Return values

*is_uart_tcp_client*는 *uart_index*로 지정된 시리얼 포트가 [통신모드]로 COD – TCP 클라이언트를 사용할 수 있으면 1을 반환하고 그렇지 않으면 0을 반환합니다.

*mac_address*을 가진 제품(ezTCP)을 찾을 수 없는 경우에는 -1을 반환합니다.

Examples

See also

Remarks

*is_uart_tcp_client*의 반환값은 다른 설정 값에 따라 달라질 수 있습니다.

10.21 is_uart_udp

```
int is_uart_udp(unsigned char *mac_address, int uart_index);
```

Description

*is_uart_udp*는 *uart_index*로 지정된 시리얼 포트가 [통신모드]로 U2S - UDP를 사용할 수 있는지 확인합니다.

Parameters

*unsigned char *mac_address*

EzTCP_Search 또는 *EzTCP_Read*를 사용하여 검색된 제품(ezTCP)의 MAC주소 입니다.

int uart_index

제품(ezTCP)의 시리얼 포트 순번으로 0부터 시작합니다.

Return values

*is_uart_udp*는 *uart_index*로 지정된 시리얼 포트가 [통신모드]로 U2S - UDP를 사용할 수 있으면 1을 반환하고 그렇지 않으면 0을 반환합니다.

*mac_address*을 가진 제품(ezTCP)을 찾을 수 없는 경우에는 -1을 반환합니다.

Examples

See also

Remarks

*is_uart_udp*의 반환값은 다른 설정 값에 따라 달라질 수 있습니다.

10.22 is_uart_serial_modbus

```
int is_uart_serial_modbus(unsigned char *mac_address, int uart_index);
```

Description

*is_uart_serial_modbus*는 *uart_index*로 지정된 시리얼 포트가 [통신모드]로 시리얼 Modbus/TCP 를 사용할 수 있는지 확인합니다.

Parameters

*unsigned char *mac_address*

EzTCP_Search 또는 *EzTCP_Read*를 사용하여 검색된 제품(ezTCP)의 MAC주소 입니다.

int uart_index

제품(ezTCP)의 시리얼 포트 순번으로 0부터 시작합니다.

Return values

*is_uart_serial_modbus*는 *uart_index*로 지정된 시리얼 포트가 [통신모드]로 시리얼 Modbus/TCP를 사용할 수 있으면 1을 반환하고 그렇지 않으면 0을 반환합니다.

*mac_address*을 가진 제품(ezTCP)을 찾을 수 없는 경우에는 -1을 반환합니다.

Examples

See also

Remarks

*is_uart_serial_modbus*의 반환값은 다른 설정 값에 따라 달라질 수 있습니다.

10.23 is_csc_hr2

```
int is_csc_hr2(unsigned char *mac_address);
```

Description

*is_csc_hr2*는 *mac_address*가 가리키는 제품(ezTCP)이 CSC-HR2인지 확인합니다.

Parameters

*unsigned char *mac_address*

EzTCP_Search 또는 *EzTCP_Read*를 사용하여 검색된 제품(ezTCP)의 MAC주소 입니다.

Return values

*is_csc_hr2*는 *mac_address*가 가리키는 제품(ezTCP)이 CSC-HR2이면 1을 반환하고 그렇지 않으면 0을 반환합니다.

*mac_address*을 가진 제품(ezTCP)을 찾을 수 없는 경우에는 -1을 반환합니다.

Examples

See also

Remarks

10.24 is_io

```
int is_io(unsigned char *mac_address);
```

Description

*is_io*는 *mac_address*가 가리키는 제품(ezTCP)이 디지털 입력과 출력 포트를 지원하는 제품인지 확인합니다.

Parameters

*unsigned char *mac_address*

EzTCP_Search 또는 *EzTCP_Read*를 사용하여 검색된 제품(ezTCP)의 MAC주소 입니다.

Return values

*is_io*는 *mac_address*가 가리키는 제품(ezTCP)이 디지털 입력과 출력 포트를 지원하면 1을 반환하고 그렇지 않으면 0을 반환합니다.

*mac_address*을 가진 제품(ezTCP)을 찾을 수 없는 경우에는 -1을 반환합니다.

Examples

See also

Remarks

10.25 is_io_output_automatic_initialize

```
int is_io_output_automatic_initialize(unsigned char *mac_address);
```

Description

*is_io_output_automatic_initialize*는 I/O 제품이 [출력포트 상태 초기화(Modbus/TCP 접속 종료 시 출력포트[초기상태] 값으로 상태 변경] 기능을 사용할 수 있는지 확인합니다.

Parameters

*unsigned char *mac_address*

EzTCP_Search 또는 *EzTCP_Read*를 사용하여 검색된 제품(ezTCP)의 MAC주소 입니다.

Return values

*is_io_output_automatic_initialize*는 제품(ezTCP)이 [출력포트 상태 초기화(Modbus/TCP 접속 종료 시 출력포트[초기상태] 값으로 상태 변경]기능을 사용할 수 있으면 1을 반환하고 그렇지 않으면 0을 반환합니다.

*mac_address*을 가진 제품(ezTCP)을 찾을 수 없는 경우에는 -1을 반환합니다.

Examples

See also

Remarks

10.26 is_event_notification

```
int is_event_notification(unsigned char *mac_address);
```

Description

*is_event_notification*은 I/O 제품이 [입력 또는 출력포트 변경 알림(전자메일)] 기능을 사용할 수 있는지 확인합니다.

Parameters

*unsigned char *mac_address*

EzTCP_Search 또는 *EzTCP_Read*를 사용하여 검색된 제품(ezTCP)의 MAC주소입니다.

Return values

*is_event_notification*은 제품(ezTCP)이 [입력 또는 출력포트 변경 알림(전자메일)] 기능을 사용할 수 있으면 1을 반환하고 그렇지 않으면 0을 반환합니다.

*mac_address*을 가진 제품(ezTCP)을 찾을 수 없는 경우에는 -1을 반환합니다.

Examples

See also

Remarks

10.27 is_wlan

```
int is_wlan(unsigned char *mac_address);
```

Description

*is_wlan*은 *mac_address*가 가리키는 제품(ezTCP)이 무선랜을 지원하는 제품인지 확인합니다.

Parameters*unsigned char *mac_address**EzTCP_Search* 또는 *EzTCP_Read*를 사용하여 검색된 제품(ezTCP)의 MAC주소 입니다.**Return values***is_wlna*은 *mac_address*가 가리키는 제품(ezTCP)이 무선랜을 지원하면 1을 반환하고 그렇지 않으면 0을 반환합니다.*mac_address*을 가진 제품(ezTCP)을 찾을 수 없는 경우에는 -1을 반환합니다.**Examples****See also****Remarks****10.28 is_wlan_soft_ap**

```
int is_wlan_soft_ap(unsigned char *mac_address);
```

Description*is_wlan_soft_ap*는 무선랜 제품이 [무선랜 종류]로 Soft AP를 사용할 수 있는지 확인합니다.**Parameters***unsigned char *mac_address**EzTCP_Search* 또는 *EzTCP_Read*를 사용하여 검색된 제품(ezTCP)의 MAC주소 입니다.**Return values***is_wlan_soft_ap*는 무선랜 제품이 [무선랜 종류]로 Soft AP를 사용할 수 있으면 1을 반환하고 그렇지 않으면 0을 반환합니다.*mac_address*을 가진 제품(ezTCP)을 찾을 수 없는 경우에는 -1을 반환합니다.**Examples****See also****Remarks**

10.29 is_wlan_antenna

```
int is_wlan_antenna(unsigned char *mac_address);
```

Description

*is_wlan_antenna*는 무선랜 제품이 [안테나] 설정 기능을 사용할 수 있는지 확인합니다.

Parameters

*unsigned char *mac_address*

EzTCP_Search 또는 *EzTCP_Read*를 사용하여 검색된 제품(ezTCP)의 MAC주소 입니다.

Return values

*is_wlan_antenna*는 무선랜 제품이 [안테나] 설정 기능을 사용할 수 있으면 1을 반환하고 그렇지 않으면 0을 반환합니다.

*mac_address*을 가진 제품(ezTCP)을 찾을 수 없는 경우에는 -1을 반환합니다.

Examples

See also

Remarks

10.30 is_wlan_phy_mode

```
int is_wlan_phy_mode(unsigned char *mac_address);
```

Description

*is_wlan_phy_mode*는 무선랜 제품이 [무선 고급설정]을 사용할 수 있는지 확인합니다.

Parameters

*unsigned char *mac_address*

EzTCP_Search 또는 *EzTCP_Read*를 사용하여 검색된 제품(ezTCP)의 MAC주소 입니다.

Return values

*is_wlan_phy_mode*는 무선랜 제품이 [무선 고급설정]을 사용할 수 있으면 1을 반환하고 그렇지 않으면 0을 반환합니다.

*mac_address*을 가진 제품(ezTCP)을 찾을 수 없는 경우에는 -1을 반환합니다.

Examples**See also****Remarks****10.31 is_wlan_background_scan**

```
int is_wlan_background_scan(unsigned char *mac_address);
```

Description

*is_wlan_background_scan*은 무선랜 제품이 [무선 고급설정]들 중에서 [Background Scan]을 사용할 수 있는지 확인합니다.

Parameters

*unsigned char *mac_address*

EzTCP_Search 또는 *EzTCP_Read*를 사용하여 검색된 제품(ezTCP)의 MAC주소 입니다.

Return values

*is_wlan_background_scan*은 무선랜 제품이 [무선 고급설정]들 중에서 [Background Scan]을 사용할 수 있으면 1을 반환하고 그렇지 않으면 0을 반환합니다.

*mac_address*을 가진 제품(ezTCP)을 찾을 수 없는 경우에는 -1을 반환합니다.

Examples**See also****Remarks****10.32 is_wlan_rsn**

```
int is_wlan_rsn(unsigned char *mac_address);
```

Description

*is_wlan_rsn*은 무선랜 제품이 RSN(Robust Security Network)을 사용할 수 있는지 확인합니다.

Parameters

*unsigned char *mac_address*

EzTCP_Search 또는 *EzTCP_Read*를 사용하여 검색된 제품(ezTCP)의 MAC주소 입니다.

Return values

*is_wlan_rsn*은 무선랜 제품이 **RSN(Robust Security Network)**을 사용할 수 있으면 1을 반환하고 그렇지 않으면 0을 반환합니다.

*mac_address*을 가진 제품(ezTCP)을 찾을 수 없는 경우에는 -1을 반환합니다.

Examples

See also

Remarks

RSN을 지원하는 무선랜 제품은 무선랜에 접속할 때 암호화 방식과 인증방식을 자동으로 선택합니다. 그러므로 RSN을 지원하는 제품과 그렇지 않은 제품은 암호화 방식과 인증방식에 관련된 설정 값을 읽고 변경하는 함수들이 따로 제공됩니다.

10.33 is_wlan_authentication_type

```
int is_wlan_authentication_type(unsigned char *mac_address);
```

Description

*is_wlan_authentication_type*은 **RSN을 지원하지 않는** 무선랜 제품이 사용 가능한 **WPA [인증 방식/암호화 방법]**을 확인합니다.

Parameters

*unsigned char *mac_address*

EzTCP_Search 또는 *EzTCP_Read*를 사용하여 검색된 제품(ezTCP)의 MAC주소 입니다.

Return values

*is_wlan_authentication_type*이 0을 반환하는 무선랜 제품은 **WPA [인증 방식/암호화 방법]**으로 WPA PSK – TKIP 와 WPA2 PSK – AES만 사용할 수 있습니다.

*is_wlan_authentication_type*이 1을 반환하는 무선랜 제품은 **WPA [인증 방식/암호화 방법]**으로 WPA PSK – TKIP, WPA PSK – AES, WPA PSK – TKIP/AES, WPA2 PSK – TKIP, WPA2 PSK – AES, WPA2 PSK – TKIP/AES를 사용할 수 있습니다.

*mac_address*을 가진 제품(ezTCP)을 찾을 수 없는 경우에는 -1을 반환합니다.

Examples

See also**Remarks****10.34 is_wlan_wpa_enterprise**

```
int is_wlan_wpa_enterprise(unsigned char *mac_address);
```

Description

*is_wlan_wpa_enterprise*는 무선랜 제품이 보안 설정 중에서 [802.1X]를 사용할 수 있는지 확인합니다.

Parameters

*unsigned char *mac_address*

EzTCP_Search 또는 *EzTCP_Read*를 사용하여 검색된 제품(ezTCP)의 MAC주소 입니다.

Return values

*is_wlan_wpa_enterprise*는 무선랜 제품이 보안 설정 중에서 [802.1X]를 사용할 수 있으면 1을 반환하고 그렇지 않으면 0을 반환합니다.

*mac_address*을 가진 제품(ezTCP)을 찾을 수 없는 경우에는 -1을 반환합니다.

Examples**See also****Remarks****10.35 is_send_mac_address**

```
int is_send_mac_address(unsigned char *mac_address);
```

Description

*is_send_mac_address*는 제품(ezTCP)이 [MAC 주소 전송] 기능을 사용할 수 있는지 확인합니다.

Parameters

*unsigned char *mac_address*

EzTCP_Search 또는 *EzTCP_Read*를 사용하여 검색된 제품(ezTCP)의 MAC주소 입니다.

Return values

*is_send_mac_address*는 제품(ezTCP)이 [MAC 주소 전송] 기능을 사용할 수 있으면 1을 반환하고 그렇지 않으면 0을 반환합니다.

*mac_address*을 가진 제품(ezTCP)을 찾을 수 없는 경우에는 -1을 반환합니다.

Examples

See also

Remarks

*is_send_mac_address*의 반환값은 다른 설정 값에 따라 달라질 수 있습니다.

10.36 is_ssl

```
int is_ssl(unsigned char *mac_address);
```

Description

*is_ssl*은 제품(ezTCP)이 [SSL 보안통신]을 사용할 수 있는지 확인합니다.

Parameters

*unsigned char *mac_address*

EzTCP_Search 또는 *EzTCP_Read*를 사용하여 검색된 제품(ezTCP)의 MAC주소 입니다.

Return values

*is_ssl*은 제품(ezTCP)이 [SSL 보안통신]을 사용할 수 있으면 1을 반환하고 그렇지 않으면 0을 반환합니다.

*mac_address*을 가진 제품(ezTCP)을 찾을 수 없는 경우에는 -1을 반환합니다.

Examples

See also

Remarks

*is_ssl*의 반환값은 다른 설정 값에 따라 달라질 수 있습니다.

10.37 is_ssh

```
int is_ssh(unsigned char *mac_address);
```

Description

*is_ssh*는 제품(ezTCP)이 [SSH 보안통신]을 사용할 수 있는지 확인합니다.

Parameters

*unsigned char *mac_address*

EzTCP_Search 또는 *EzTCP_Read*를 사용하여 검색된 제품(ezTCP)의 MAC주소 입니다.

Return values

*is_ssh*는 제품(ezTCP)이 [SSH 보안통신]을 사용할 수 있으면 1을 반환하고 그렇지 않으면 0을 반환합니다.

*mac_address*을 가진 제품(ezTCP)을 찾을 수 없는 경우에는 -1을 반환합니다.

Examples

See also

Remarks

*is_ssh*의 반환값은 다른 설정 값에 따라 달라질 수 있습니다.

10.38 is_remote_debug

```
int is_remote_debug(unsigned char *mac_address);
```

Description

*is_remote_debug*는 제품(ezTCP)이 [디버깅 로그 보기] 기능을 사용할 수 있는지 확인합니다.

Parameters

*unsigned char *mac_address*

EzTCP_Search 또는 *EzTCP_Read*를 사용하여 검색된 제품(ezTCP)의 MAC주소 입니다.

Return values

*is_remote_debug*는 제품(ezTCP)이 [디버깅 로그 보기] 기능을 사용할 수 있으면 1을 반환하고 그렇지 않으면 0을 반환합니다.

*mac_address*을 가진 제품(ezTCP)을 찾을 수 없는 경우에는 -1을 반환합니다.

Examples**See also****Remarks****10.39 is_tcp_multi_connection**

```
int is_tcp_multi_connection(unsigned char *mac_address);
```

Description

*is_tcp_multi_connection*은 제품(ezTCP)이 [다중 접속] 기능을 사용할 수 있는지 확인합니다.

Parameters

*unsigned char *mac_address*

EzTCP_Search 또는 *EzTCP_Read*를 사용하여 검색된 제품(ezTCP)의 MAC주소 입니다.

Return values

*is_tcp_multi_connection*은 제품(ezTCP)이 [다중 접속] 기능을 사용할 수 있으면 1을 반환하고 그렇지 않으면 0을 반환합니다.

*mac_address*을 가진 제품(ezTCP)을 찾을 수 없는 경우에는 -1을 반환합니다.

Examples**See also****Remarks**

*is_tcp_multi_connection*의 반환값은 다른 설정 값에 따라 달라질 수 있습니다.

10.40 is_power_management

```
int is_power_management(unsigned char *mac_address);
```

Description

*is_power_management*는 제품(ezTCP)이 [전원 관리] 기능을 사용할 수 있는지 확인합니다.

Parameters

*unsigned char *mac_address*

EzTCP_Search 또는 *EzTCP_Read*를 사용하여 검색된 제품(ezTCP)의 MAC주소 입니다.

Return values

*is_power_management*는 제품(ezTCP)이 [전원 관리] 기능을 사용할 수 있으면 1을 반환하고 그렇지 않으면 0을 반환합니다.

*mac_address*을 가진 제품(ezTCP)을 찾을 수 없는 경우에는 -1을 반환합니다.

Examples

See also

Remarks

10.41 is_password

```
int is_password(unsigned char *mac_address);
```

Description

*is_password*는 비밀번호가 제품(ezTCP)에 설정되어있는지 확인합니다.

Parameters

*unsigned char *mac_address*

EzTCP_Search 또는 *EzTCP_Read*를 사용하여 검색된 제품(ezTCP)의 MAC주소 입니다.

Return values

*is_password*는 제품(ezTCP)에 비밀번호가 설정되어있으면 1을 반환하고 그렇지 않으면 0을 반환합니다.

*mac_address*을 가진 제품(ezTCP)을 찾을 수 없는 경우에는 -1을 반환합니다.

Examples

See also

Remarks

10.42 is_certificates

```
int is_certificates(unsigned char *mac_address);
```

Description

*is_certificates*는 제품(ezTCP)이 [인증서 관리] 기능을 사용할 수 있는지 확인합니다.

Parameters

*unsigned char *mac_address*

EzTCP_Search 또는 *EzTCP_Read*를 사용하여 검색된 제품(ezTCP)의 MAC주소 입니다.

Return values

*is_certificates*는 제품(ezTCP)이 인증서 기능을 사용할 수 있으면 1을 반환하고 그렇지 않으면 0을 반환합니다.

*mac_address*을 가진 제품(ezTCP)을 찾을 수 없는 경우에는 -1을 반환합니다.

Examples

See also

Remarks

11 면책 고지 사항

11.1 면책 고지 사항

솔내시스템(주)과 그 대리점은 CIE-H14의 사용 또는 사용불능에 따른 손해 및 손실, 영업중지로 인한 비용, 정보 손실을 포함한 기타 고지 받은 어떠한 재정적 손해에 대해서도 책임지지 않습니다.

CIE-H14는 허락되지 않는 응용분야에서의 사용을 금지합니다. 허락되지 않은 응용분야라 함은 군사, 핵, 항공, 폭발물, 의학, 방범설비, 화재경보기, 엘리베이터를 수반한 용도 혹은 차량, 항공기, 트럭, 보트, 헬리콥터 및 이에 국한되지 않는 모든 교통수단을 포함합니다.

또한, 고장 및 실패로 인한 재정적 손실 및 기물파손, 신체 상해 혹은 사람이나 동물의 상해를 초래하는 실험, 개발 및 각종 응용분야에 사용할 수 없습니다. 구매자(혹은 업체)가 자발적 혹은 비자발적으로 이러한 허락되지 않는 응용분야에 사용할 시 솔내시스템(주)과 그 대리점에 손해배상을 포함한 어떠한 책임도 묻지 않을 것에 동의한 것으로 간주합니다.

구매한 제품의 환불 및 수리, 교환에 대한 배상 책임과 구매자(혹은 업체)의 단독 구제책은 솔내시스템(주)과 그 대리점의 선택사항입니다.

솔내시스템(주)과 그 대리점은 동반된 기술자료, 하드웨어, 펌웨어를 포함한 CIE-H14의 상업성이나 특정목적에 따른 적합성에 대한 모든 명시적 혹은 묵시적 보증 및 기타 이에 국한되지 않는 여타의 보증을 하지 않습니다.

12 문서 변경 이력

날짜	버전	변경내용	작성자
2013.10.30	1.0	○ 최초 작성.	김형준
2022.05.25	1.1	○ 인증서관련 함수 추가.	김형준