Visual C++을 위한 ezManager 프로그램 라이브러리

2012-11-16

Sollae Systems Co., Ltd.
http://www.sollae.co.kr

목차

1 소개 - 4 -

2 데이터 구조체 - 5 -

2.1 lib_env - 5 -
2.1.1 개요 - 5 -
2.1.2 변수 - 5 -
2.2 net_env - 6 -
2.2.1 개요 - 6 -
2.2.2 변수 - 7 -
2.2.3 주의 - 9 -
2.3 uart_env - 10 -
2.3.1 개요 - 10 -
2.3.2 변수 - 10 -
2.3.3 주의 - 11 -
2.4 uart_dev_env - 12 -
2.4.1 개요 - 12 -
2.4.2 변수 - 12 -
2.5 uart_var_env - 16 -
2.5.1 개요 - 16 -
2.5.2 변수 - 16 -
2.5.3 주의 - 18 -
2.6 opt_env - 19 -
2.6.1 개요 - 19 -
2.6.2 변수 - 19 -
2.7 etc_env - 23 -
2.7.1 개요 - 23 -
2.7.2 변수 - 23 -
2.7.3 주의 - 24 -
2.8 io_env - 25 -
2.8.1 개요 - 25 -
2.8.2 변수 - 25 -
2.9 io_var_env - 27 -
2.9.1 개요 - 27 -
2.9.2 변수 - 27 -
2.9.3 주의 - 31 -
2.10 ip_trap_env - 32 -
2.10.1 개요 - 32 -
2.10.2 변수 - 32 -
2.10.3 주의 - 32 -
2.11 port_map_env - 33 -
2.11.1 개요 - 33 -
2.11.2 변수 - 33 -
2.12 wlan_env - 34 -
2.12.1 개요 - 34 -
2.12.2 변수 - 34 -
2.13 wlan_opt_env - 35 -

2.13.1 개요 ..... - 35 -
2.13.2 변수 ..... - 35 -
2.14 wlan_var_env ..... - 38 -
2.14.1 개요 ..... - 38 -
2.14.2 변수 ..... - 38 -
2.15 csc_hr2_env ..... - 40 -
2.15.1 개요 ..... - 40 -
2.15.2 변수 ..... - 40 -
2.16 redundancy_var_env ..... - 42 -
2.16.1 개요 ..... - 42 -
2.16.2 변수 ..... - 42 -
2.17 tcp_status_env ..... - 44 -
2.17.1 개요 ..... - 44 -
2.17.2 변수 ..... - 44 -
2.18 tcp6_status_env ..... - 46 -
2.18.1 개요 ..... - 46 -
2.18.2 변수 ..... - 46 -

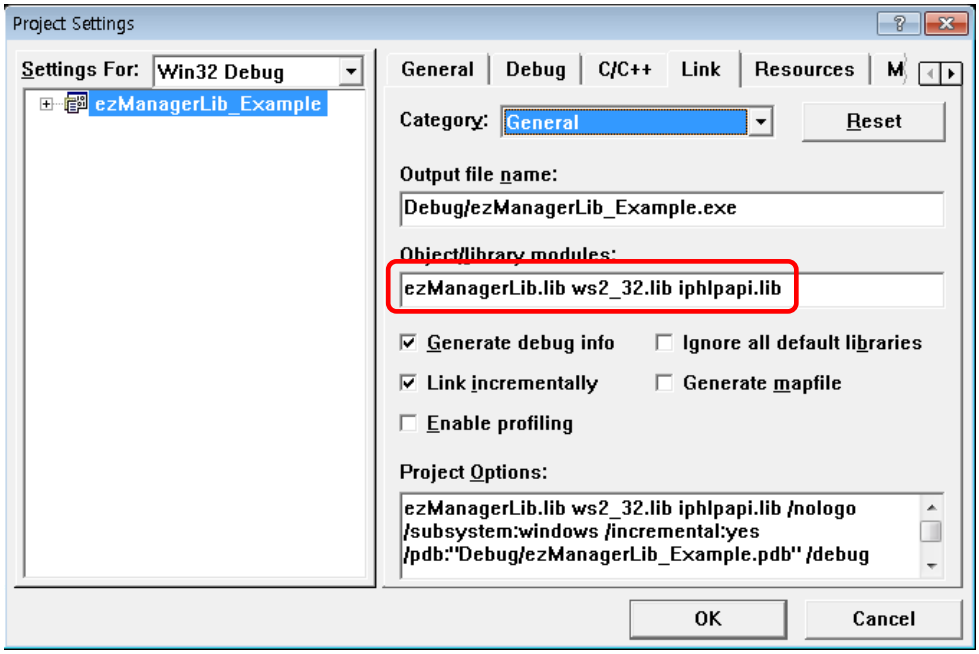
3 함수 ..... - 48 -

3.1 EzManager_Search ..... - 48 -
3.1.1 개요 ..... - 48 -
3.1.2 함수원형 ..... - 48 -
3.1.3 함수인자 ..... - 48 -
3.1.4 함수 반환 값 ..... - 49 -
3.1.5 주의 ..... - 49 -
3.2 EzManager_Read ..... - 50 -
3.2.1 개요 ..... - 50 -
3.2.2 함수원형 ..... - 50 -
3.2.3 함수인자 ..... - 50 -
3.2.4 함수 반환 값 ..... - 51 -
3.2.5 주의 ..... - 51 -
3.3 EzManager_Write ..... - 52 -
3.3.1 개요 ..... - 52 -
3.3.2 함수원형 ..... - 52 -
3.3.3 함수인자 ..... - 52 -
3.3.4 함수 반환 값 ..... - 53 -
3.3.5 주의 ..... - 54 -
3.4 EzManager_Status ..... - 55 -
3.4.1 개요 ..... - 55 -
3.4.2 함수원형 ..... - 55 -
3.4.3 함수인자 ..... - 55 -
3.4.4 함수 반환 값 ..... - 56 -
3.4.5 주의 ..... - 56 -
3.5 EzManager_ChangePwd ..... - 58 -
3.5.1 개요 ..... - 58 -
3.5.2 함수원형 ..... - 58 -
3.5.3 함수인자 ..... - 58 -
3.5.4 함수 반환 값 ..... - 59 -

3.5.5 주의 ..... - 59 -
3.6 EzManager_CloseTCP ..... - 60 -
3.6.1 개요 ..... - 60 -
3.6.2 함수원형 ..... - 60 -
3.6.3 함수인자 ..... - 60 -
3.6.4 함수 반환 값 ..... - 61 -
3.6.5 주의 ..... - 61 -
3.7 EzManager_RdbOnOff ..... - 62 -
3.7.1 개요 ..... - 62 -
3.7.2 함수원형 ..... - 62 -
3.7.3 함수인자 ..... - 62 -
3.7.4 함수 반환 값 ..... - 64 -
3.7.5 주의 ..... - 64 -
3.8 EzManager_ReBoot ..... - 65 -
3.8.1 개요 ..... - 65 -
3.8.2 함수원형 ..... - 65 -
3.8.3 함수인자 ..... - 65 -
3.8.4 함수 반환 값 ..... - 66 -
3.8.5 주의 ..... - 66 -
3.9 GetLibVer ..... - 67 -
3.9.1 개요 ..... - 67 -
3.9.2 함수원형 ..... - 67 -
3.9.3 함수인자 ..... - 67 -
3.9.4 함수 반환 값 ..... - 67 -
3.10 GetProductName ..... - 68 -
3.10.1 개요 ..... - 68 -
3.10.2 함수원형 ..... - 68 -
3.10.3 함수인자 ..... - 68 -
3.10.4 함수 반환 값 ..... - 68 -
3.11 Exit_Library ..... - 69 -
3.11.1 개요 ..... - 69 -
3.11.2 함수원형 ..... - 69 -

1 소개

- ezManager프로그램에서 제공하는 주요 기능을 라이브러리 형태로 제공하므로 사용자가 직접 ezManager와 같은 프로그램을 작성할 수 있습니다.
- ezManager프로그램 라이브러리를 사용하기 위해서는 작성중인 소스코드에 ezManagerLib.h 헤더파일을 포함하고 ws2_32.lib, iphlpapi.lib 와 ezManagerLib.lib 세 개의 라이브러리를 컴파일 시 포함시켜야 합니다.



주의사항:

- ezManager 라이브러리 구조체 변수 중 "**사용하지 않음**"으로 지정되어 있는 변수는 사용하지 마십시오.
- ezManager 라이브러리 구조체 변수 중 "**읽기 전용**"으로 지정되어 있는 변수는 값을 변경하지 마십시오.
- 환경변수를 저장하는 함수를 사용하기 전에 ezTCP의 MAC 주소 또는 IP 주소를 확인하십시오. 다른 제품의 환경변수를 잘 못 저장하는 경우 제품이 정상적으로 동작하지 않을 수 있습니다.
- **비정상적인 라이브러리 사용으로 인한 피해는 당사에서 책임지지 않습니다.**

2 데이터 구조체

2.1 lib_env

2.1.1 개요

- ezManager 라이브러리 함수들이 사용하는 기본이 되는 데이터 구조체 입니다.

```
#define MAX_COM_PORT        48

struct lib_env
{
        struct net_env          net_env;
        struct ip6_env          ip6_env;
        struct uart_env         uart_env[MAX_COM_PORT];
        struct opt_env          opt_env;
        struct etc_env          etc_env;
        struct io_env           io_env;
        struct ip_trap_env      ip_trap_env;
        struct port_map_env     port_map_env;
        struct wlan_env         wlan_env;
        struct csc_hr2_env      csc_hr2_env;
};
```

2.1.2 변수

- net_env
  [입력/출력] 네트워크관련 환경변수를 저장 할 구조체 입니다.
- ip6_env
  [입력/출력] IPv6관련 환경변수를 저장 할 구조체 입니다.
- uart_env
  [입력/출력] UART 설정에 관련된 환경변수를 저장 할 구조체 입니다.
- opt_env
  [입력/출력] 환경변수를 저장 할 구조체 입니다.
- etc_env
  [입력/출력] 환경변수를 저장 할 구조체 입니다.

- io_env
  [입력/출력] I/O 제품관련 환경변수를 저장 할 구조체 입니다.
- ip_trap_env
  [입력/출력] IP 주소 통보 기능에 대한 환경변수를 저장 할 구조체 입니다.
- port_map_env
  [입력/출력] 제품의 TCP포트 관련 환경변수를 저장 할 구조체 입니다.
- wlan_env
  [입력/출력] 무선랜 관련 환경변수를 저장 할 구조체 입니다.
- csc_hr2_env
  [입력/출력] CSC-HR2 관련 환경변수를 저장 할 구조체 입니다.

2.2 net_env

2.2.1 개요

- 네트워크관련 환경변수를 저장 할 구조체 입니다.

```
struct net_env
{
        _u8     mac_addr        [6];    // 읽기 전용. 값을 변경하지 마십시오.
        _u8     secure          [6];    // 읽기 전용. 값을 변경하지 마십시오.
        _u16    major;                  // 읽기 전용. 값을 변경하지 마십시오.
        _u16    minor;                  // 읽기 전용. 값을 변경하지 마십시오.
        _u32    local_ip;
        _u32    net_mask;
        _u32    gate_ip;
        _u8     pppoe_id        [32];
        _u8     pppoe_pwd       [16];
        _u32    socket_ip;              // 읽기 전용. 값을 변경하지 마십시오.
        _u32    dns_ip;
        _u16    product_id_old;         // 읽기 전용. 값을 변경하지 마십시오.
        _u16    product_id_new;         // 읽기 전용. 값을 변경하지 마십시오.
        _u8     ddns_id         [32];
        _u8     ddns_pwd        [16];
        _u8     ddns_host_name  [64];
        _u8     ssh_id          [16];
        _u8     ssh_pwd         [16];
};
```

2.2.2 변수

- mac_addr

  [출력] 제품의 MAC 주소 입니다.

  읽기 전용 변수입니다.

  예) MAC 주소 00:30:f9:12:34:56 는 다음과 같이 저장되어 있습니다.
  mac _addr[0] = 0x00, mac _addr[1] = 0x30, mac _addr[2] = 0xf9,
  mac _addr[3] = 0x12, mac _addr[4] = 0x34, mac _addr[5] = 0x56

- major

  [출력] 펌웨어 버전번호 중 메이저 번호 입니다.

- minor

  [출력] 펌웨어 버전번호 중 마이너와 리비전 번호 입니다.

7(MSB)	6	5	4	3	2	1	0(LSB)
마이너 번호				리비전 번호			

- local_ip

  [입력/출력] 제품 IP 주소 입니다.

  라이브러리에서 제공하는 아래의 매크로를 사용하여 IP 주소를 생성할 수 있습니다.

```
#define MK_IP(__b1,__b2,__b3,__b4) \
        (((unsigned long)(__b4) << 24) | ((unsigned long)(__b3) << 16) | \
        ((unsigned long)(__b2) << 8) | ((unsigned long)(__b1)) )
```

- net_mask

  [입력/출력] 서브넷 마스크 입니다.

- gate_ip;

  [입력/출력] 게이트웨이 IP 주소 입니다.

- poe_uid

  [입력/출력] PPPoE를 위한 로그인 아이디 입니다.

- poe_pwd

  [입력/출력] PPPoE를 위한 로그인 비밀번호 입니다.

- socket_ip

  [출력] 프로그램에서 사용하는 TCP/IP 소켓에서 읽은 IP 주소 정보 입니다.
  읽기 전용 변수 입니다. 이 IP 주소는 TCP/IP 소켓에서 읽은 패킷의 IP 주소 입니다. ezTCP가 유동아이피를 사용하는 경우 ezTCP의 IP 주소를 확인 하는 용도로 사용가능 합니다.

- dns_ip
  [입력/출력] DNS 서버 IP 주소 입니다.
- product_id_old
  [출력] ezTCP 제품 아이디 입니다.
  읽기 전용 변수 입니다. 이 변수 값을 변경하지 마십시오.

아이디	제품명
0x10	CIE-H10
0x11	CSE-H20 / CSE-H21
0x12	CSE-M32
0x14	CSE-M73
0x15	CSW-H80
0x21	CIE-M10
0x29	CSE-H25
0x2b	CSE-M53
0x2c	CSE-H53
0x2d	CSW-M83
0x2e	CSW-M85
0x2f	CSE-H55
0x30	CSC-HR2
0x34	CIE-H12
0x35	CSW-H85
0x36	CSE-T32
0x37	CSE-M53A
0x39	CSE-T16
0x3a	CSE-T48
0x3b	CSE-H53A
0x3c	CSW-M84
0x3d	CSE-M53N
0x3e	CSE-H53N
0x3f	CSE-H55N

- product_id_new
  [출력] ezTCP 제품 아이디 입니다.
  읽기 전용 변수 입니다. 이 변수 값을 변경하지 마십시오. product_id_new 값이 0 인 경우에는 product_id_old를 사용하십시오.

아이디	제품명
0x20	CIE-H10
0x21	CIE-M10
0x23	CSE-M32
0x24	CSE-H20
0x25	CSE-H21

0x26	CSE-M73
0x27	CSW-H80
0x29	CSE-H25
0x2b	CSE-M53
0x2c	CSE-H53
0x2d	CSW-M83
0x2e	CSW-M85
0x2f	CSE-H55
0x30	CSC-HR2
0x34	CIE-H12
0x35	CSW-H85
0x36	CSE-T32
0x37	CSE-M53A
0x39	CSE-T16
0x3a	CSE-T48
0x3b	CSE-H53A
0x3c	CSW-M84
0x3d	CSE-M53N
0x3e	CSE-H53N
0x3f	CSE-H55N

※ **product_id_new 변수는 제품 구분이 세분화 되면서 추가된 변수 입니다. 따라서 제품 아이디를 확인하는 경우에는 product_id_new 변수가 0 인 경우에는 product_id_old 변수를 사용하십시오.**

- ddns_id
  [입력/출력] DDNS 서비스를 위한 로그인 아이디 입니다.
- ddns_pwd
  [입력/출력] DDNS 서비스를 위한 로그인 비밀번호 입니다.
- ddns_host_name
  [입력/출력] DDNS 서비스를 위한 호스트 이름 입니다.
  ※ **DDNS에 관련된 자세한 내용은 제품 사용자 설명서를 참고하십시오.**
- ssh_id, ssh_pwd
  [입력/출력] SSH 서비스를 위한 아이디와 비밀번호 입니다.

2.2.3 주의

- 아래의 변수들을 제외한 모든 변수들은 리틀 엔디언(Host Byte Order)을 사용 합니다.
- 다음의 변수들은 빅 엔디언(Network Byte Order)을 사용 합니다.
  local_ip, net_mask, gate_ip, socket_ip, dns_ip

2.3 uart_env

2.3.1 개요

- UART 설정에 관련된 환경변수를 저장 할 구조체 입니다.

```
struct uart_env
{
        struct uart_dev_env     uart_dev_env;
        struct uart_var_env     uart_var_env;
        struct uart_del_env     uart_del_env;
        _u8                     host_name       [64];
        _u8                     comment         [32];
        _u32                    use_flag;       // 읽기 전용. 값을 변경하지 마십시오.
};
```

2.3.2 변수

- uart_dev_env
  [입력/출력] UART 하드웨어 설정에 관련된 환경변수를 저장 할 구조체 입니다.
- uart_var_env
  [입력/출력] UART 작동에 관련된 환경변수를 저장 할 구조체 입니다.
- uart_del_env
  [입력/출력] UART 작동에 관련된 환경변수를 저장 할 구조체 입니다.
  ※ **현재 지원하지 않습니다.**
- host_name
  [입력/출력] ezTCP가 접속할 호스트의 DNS 이름 입니다.
  이 변수는 ezTCP가 TCP 클라이언트로 작동되는 경우 사용 됩니다. ezTCP가 접속할 호스트 주소를 DNS 이름으로 사용하려면 이 변수에 DNS 이름을 입력하고 “uart_var_env” 구조체의 “peer_ip” 변수는 0으로 설정하여야 합니다.
- comment
  [입력/출력] UART 포트의 간략한 설명 입니다.
- use_flag
  [출력] UART 사용 여부를 나타냅니다.
  읽기 전용 변수 입니다. use_flag가 1인 경우에만 유효한 UART 입니다.

2.3.3 주의

- “uart_env” 변수는 “lib_env” 구조체에 다음과 같이 정의 되어 있습니다.

```
struct uart_env uart_env[MAX_COM_PORT]
```

이것은 유효한 UART 개 수가 제품에 따라서 다르다는 것을 나타냅니다. 따라서, “uart_env” 변수를 사용하기 전에 “use_flag” 변수를 확인하여 유효한 UART인지를 확인해야 합니다.

2.4 uart_dev_env

2.4.1 개요

- UART 하드웨어 설정에 관련된 환경변수를 저장 할 구조체 입니다.

```
struct uart_dev_env
{
        unsigned int    max_stype       : 2;    /* 사용하지 않음. 값을 사용하지 마십시오. */
        unsigned int    stype           : 2;
        unsigned int    databit         : 2;
        unsigned int    stopbit         : 2;
        unsigned int    parity          : 2;
        unsigned int    flowctrl        : 2;
        unsigned int    telcom          : 2;
        unsigned int    parity2         : 2;
        unsigned int    en_ttl          : 1;    // 읽기 전용. 값을 변경하지 마십시오.
        unsigned int    ttl             : 1;
        unsigned int    en_tx_delay     : 1;    // 읽기 전용. 값을 변경하지 마십시오.
        unsigned int    tx_delay        : 5;
        unsigned int    dtrdsr          : 1;
        unsigned int    pad0            : 1;
        unsigned int    pad2            : 6;    /* 사용하지 않음. 값을 사용하지 마십시오. */
        _u32            max_baud;               // 읽기 전용. 값을 변경하지 마십시오.
        _u32            sio_baud;
};
```

2.4.2 변수

- stype
  [입력/출력] 시리얼 종류.

stype	설명
0	RS-232
1	RS-485
2	RS-422

- databit
  [입력/출력] 데이터 비트.

databit	설명
0	5-비트
1	6-비트
2	7-비트
3	8-비트

- stopbit
  [입력/출력] 정지 비트.

stopbit	설명
0	1-비트
1	1.5-비트
2	2-비트

- parity
  [입력/출력] 패리티 비트.

parity	설명
0	사용안함
1	EVEN 패리티
2	ODD 패리티
3	parity2 변수를 사용

- flowctrl
  [입력/출력] 흐름제어.

flowctrl	설명
0	사용안함
1	RTS / CTS
2	Xon / Xoff

- telcom
  [입력/출력]
  시리얼 포트 설정/상태 전송(RFC2217) 옵션 사용여부를 설정합니다.
  자세한 설명은 제품 사용자 설명서를 참고 하십시오.

- parity2
  [입력/출력] 패리티 비트.

parity2	설명
0	Mark 패리티
1	Space 패리티

- en_ttl
  [출력]
  읽기 전용 변수 입니다. 제품의 TTL레벨 출력 지원여부를 나타냅니다.

  이 변수는 아래의 제품만 유효 합니다.

LAN 종류	제품명
유선랜	CSE-M73(H/W 버전 1.3이상, F/W 버전 1.4a이상)
무선랜	

- ttl
  [입력/출력]
  TTL레벨 출력 사용여부를 설정합니다.

  이 변수는 아래의 제품만 유효 합니다.

LAN 종류	제품명
유선랜	CSE-M73(H/W 버전 1.3이상, F/W 버전 1.4a이상)
무선랜	

- en_tx_delay
  [출력]
  읽기 전용 변수 입니다. 제품의 데이터 전송 간격 옵션 지원여부를 나타냅니다.

- tx_delay
  [입력/출력]
  데이터 전송 간격을 설정 합니다.
  자세한 설명은 제품 사용자 설명서를 참고 하십시오.
- dtrdsr
  [입력/출력] DTR/DSR 흐름제어.
- max_baud
  [출력] 최대 시리얼 통신 속도.
  읽기 전용 변수 입니다. 이 변수는 UART가 지원하는 최대 시리얼 통신 속도를 나타냅니다.
- sio_baud
  [입력/출력] 시리얼 통신 속도.
  UART의 통신 속도를 설정합니다.
  max_baud 보다 큰 값을 설정하는 경우 제품이 동작하지 않을 수 있으므로 시리얼 통신 속도 설정 시 주의하여 주십시오.

2.5 uart_var_env

2.5.1 개요

- UART 작동에 관련된 환경변수를 저장 할 구조체 입니다.

```
struct uart_var_env
{
        _u32    local_ip;                   /* 사용하지 않음. 값을 사용하지 마십시오. */
        _u32    peer_ip;
        _u16    local_port;
        _u16    peer_port;
        _u8     mux_type;
        _u8     no_delay        :1;
        _u8     cod_listen      :1;         /* 사용하지 않음. 값을 사용하지 마십시오. */
        _u8     secure          :2;
        _u8     pad             :4;         /* 사용하지 않음. 값을 사용하지 마십시오. */
        _u16    water_mark;
        _u16    time_mark;
        _u16    timeout;
};
```

2.5.2 변수

- peer_ip
  [입력/출력] 제품이 통신 할 서버의 IP 주소.
  “mux_type”이 COD(TCP 클라이언트)와 U2S(UDP)인 경우에만 사용 됩니다.
  IP 주소 대신 DNS 이름을 사용하려면 peer_ip 값을 0으로 설정하고 host_name에 DNS 이름을 저장하면 됩니다.
- local_port
  [입력/출력] 제품 로컬포트.
  “mux_type”이 COD(TCP 클라이언트)와 U2S(UDP)인 경우에만 사용 됩니다.
- peer_port
  [입력/출력] 제품이 통신할 서버의 포트번호.
  “mux_type”이 COD(TCP 클라이언트)와 U2S(UDP)인 경우에만 사용 됩니다.

- mux_type
  [입력/출력] 제품의 동작방식을 설정.

mux_type	동작	설명
0	T2S	[TCP 서버] 제품은 TCP/IP 접속을 기다립니다.
1	ATC	[AT 명령] AT명령어를 사용하여 TCP 서버 또는 TCP 클라이언트로 사용이 가능합니다.
2	COD	[TCP 클라이언트] 제품은 peer_ip와 peer_port를 사용하여 TCP/IP 접속을 시도합니다.
3	U2S	[UDP] UDP를 사용하여 통신을 합니다.

- no_delay
  [입력/출력]
  no_dealy 옵션을 설정하면 UART에서 수신한 데이터를 전송지연 기능 없이 네트워크로 전송합니다. 좀 더 빠른 데이터 전송을 원하는 경우에 사용하십시오.
- water_mark
  [입력/출력] TCP 접속 전 데이터 크기 [단위:바이트].
  제품의 시리얼포트로 “water_mark” 이상의 데이터가 수신되면 TCP 접속을 시도 하거나 네트워크로 데이터를 전송 합니다.
  “mux_type”이 COD(TCP 클라이언트)와 U2S(UDP)인 경우에만 사용 됩니다.
- time_mark
  [입력/출력] 데이터 프레임 간격 [단위:10ms]
  시리얼포트로 마지막 데이터를 수신 한 후 “time_mark”동안 시리얼포트로 데이터가 들어오지 않으면 이전에 수신된 데이터를 전송 합니다.
  최소값은 4(40ms)입니다.

- timeout
  [입력/출력] 접속동료 대기시간 [단위:초].
  “mux_type”이 T2S(TCP서버), ATC(AT명령), COD(TCP클라이언트)인 경우 timeout에 지정된 시간만큼 TCP/IP 데이터 통신이 없으면 제품이 TCP/IP 접속을 먼저 해제 합니다. 0이면 TCP/IP 접속을 계속 유지합니다.

2.5.3 주의

- 아래의 변수들을 제외한 모든 변수들은 리틀 엔디언(Host Byte Order)을 사용 합니다.
- 다음의 변수들은 빅 엔디언(Network Byte Order)을 사용 합니다.
  peer_ip

2.6 opt_env

2.6.1 개요

- 환경변수를 저장 할 구조체 입니다.

```
struct opt_env
{
        unsigned int    ezcfg_lock      : 1;
        unsigned int    rcfg            : 1;
        unsigned int    arp             : 1;
        unsigned int    dhcp            : 1;
        unsigned int    pppoe           : 1;
        unsigned int    auto_ns         : 1;
        unsigned int    pad0            : 2;    /* 사용하지 않음. 값을 사용하지 마십시오. */
        unsigned int    ip6             : 1;
        unsigned int    ip6_eui         : 2;
        unsigned int    ip6_gua         : 2;
        unsigned int    pad1            : 3;    /* 사용하지 않음. 값을 사용하지 마십시오. */
        unsigned int    debug           : 1;
        unsigned int    telnet          : 1;
        unsigned int    ssl             : 1;
        unsigned int    ssh             : 1;
        unsigned int    http            : 1;
        unsigned int    ddns            : 3;
        unsigned int    t2smc           : 1;
        unsigned int    secure          : 1;    // 읽기 전용. 값을 변경하지 마십시오.
        unsigned int    mac_id          : 1;
        unsigned int    ps              : 1;
        unsigned int    pd              : 1;
        unsigned int    pad3            : 3;    /* 사용하지 않음. 값을 사용하지 마십시오. */
};
```

2.6.2 변수

- ezcfg_lock
  [입력/출력]
  “ezcfg_lock”옵션을 설정하면 “etc_env” 구조체 변수들 중 “allow_mac” 또는 “allow_ip” 에 설정된 호스트에서 받은 패킷에만 응답을 보냅니다.
- rcfg
  [입력/출력]
  “rcfg”옵션을 설정하면 IP 주소 검색 기능이 활성화 됩니다.

- arp
  [입력/출력]
  arp옵션을 설정하면 ARP 패킷을 사용하여 제품 IP 주소를 임시로 변경할 수 있습니다.
- dhcp
  [입력/출력]
  dhcp옵션을 설정하면 DHCP 프로토콜이 활성화 됩니다.
- pppoe
  [입력/출력]
  pppoe옵션을 설정하면 PPPoE 프로토콜이 활성화 됩니다.
- auto_ns
  [입력/출력]
  auto_ns 옵션을 설정하면 제품이 DHCP 또는 PPPoE 프로토콜을 사용할 때 DNS 서버 IP 주소를 자동으로 설정합니다. auto_ns 옵션을 설정하지 않으면 “net_env” 구조체의 “dns_ip” 변수에 설정된 IP 주소를 DNS 서버로 사용합니다.
- ip6
  [입력/출력]
  ip6를 설정하면 IP6 주소를 사용합니다.
- ip6_eui
  [입력/출력]
  링크-로컬 IP6 주소를 만드는 방법을 설정합니다.

ip6_eui	설명
0	제품(ezTCP) MAC 주소를 사용.
1	임의로 생성.

- ip6_gua
  [입력/출력]
  IP6 주소 방식

ip6_gua	설명
0	자동으로 IP 주소 받기.
1	고정된 IP 주소 사용.

- debug
  [입력/출력]
  debug옵션을 설정하면 제품은 디버깅 정보를 담은 UDP 브로드캐스트 패킷을 UDP 포트 50006번으로 전송합니다.
- telnet
  [입력/출력]
  telnet옵션을 설정하면, 제품의 Telnet콘솔에 접속할 수 있습니다.
  이 옵션은 ssh 옵션과 같이 사용할 수 없습니다.
- ssl
  [입력/출력]
  ssl옵션을 설정하면, TCP/IP 데이터 통신 시 SSL프로토콜을 사용하여 데이터를 암호화 합니다.
- ssh
  [입력/출력]
  ssh옵션을 설정하면, TCP/IP 데이터 통신 시 SSH프로토콜을 사용하여 데이터를 암호화 합니다.
- http
  [입력/출력]
  HTTP프로토콜을 사용하여 제품을 모니터링 할 수 있습니다.
  이 변수는 아래의 제품만 유효 합니다.

LAN 종류	제품명
유선랜	CIE-H10, CIE-M10, CIE-H12
무선랜	

- ddns
  [입력/출력]
  IP 주소 통보 기능의 종류를 선택합니다.

ddns	설명
0	사용안함
1	DDNS (dyndns.org 서비스를 사용합니다)
2	TCP
3	UDP

※ 자세한 설명은 제품 사용자 설명서를 참고 하십시오.

- t2smc
  [입력/출력]
  "mux_type"을 T2S(TCP서버)로 사용하는 경우 다중접속 허용 여부를 선택합니다.
  이 변수는 아래의 제품만 유효 합니다.

LAN 종류	제품명
유선랜	CSE-M73, CSE-H25
무선랜	

- secure
  [출력]
  읽기 전용 변수 입니다. 제품이 SSL 또는 SSH 기능 지원여부를 나타냅니다.

- mac_id
  [입력/출력]
  mac_id 옵션을 설정하면, 제품은 TCP/IP 접속이 완료되면 원격지 호스트로 제품의 MAC 주소를 전송 합니다.

2.7 etc_env

2.7.1 개요

- 환경변수를 저장 할 구조체 입니다.

```
struct etc_env
{
        _u8     allow_mac       [6];
        _u8     pad0            [2];            /* 사용하지 않음. 값을 사용하지 마십시오. */
        _u32    allow_ip;
        _u32    allow_subnet;
        _u8     comment         [64];
        _u8     pwd_old         [PASSWD_SIZE]; /* 사용하지 않음. 값을 사용하지 마십시오. */
        _u8     pwd             [PASSWD_SIZE]; /* 사용하지 않음. 값을 사용하지 마십시오. */
        _u8     allow_ip6       [16];
        _u32    allow_ip6_prefix;
};
```

2.7.2 변수

- allow_mac
  [입력/출력]
  제품에 접속 가능한 호스트를 MAC 주소로 제한하고자 하는 경우 사용합니다.

- allow_ip
  allow_subnet
  [입력/출력]
  제품에 접속 가능한 호스트를 IP 주소와 서브넷 마스크를 사용하여 제한하고자 하는 경우 사용합니다. 사용 예는 다음과 같습니다.

allow_ip	allow_subnet	접속가능 한 호스트 범위
10.1.0.1	255.0.0.0	10.1.0.1 ~ 10.255.255.254
10.1.0.1	255.255.255.0	10.1.0.1 ~ 10.1.0.254
192.168.1.4	255.255.255.255	192.168.1.4

- comment
  [입력/출력]
  제품에 대한 간단한 설명을 입력할 수 있습니다.

- pwd_old, pwd
  [출력]
  사용이 금지된 변수 입니다..
  제품에 비밀번호를 설정할 때 사용하는 변수로 직접적인 사용은 금지되어 있습니다. 비밀번호를 설정 또는 변경하고자 하는 경우에는 라이브러리에서 제공하는 함수를 사용하십시오.
  pwd_old 변수에 "********" 이 설정되어 있는 경우 해당 제품에 비밀번호가 설정되어 있는 것입니다.
- allow_ip6
  [입력/출력]
  제품에 접속 가능한 호스트의 IP6 주소 입니다.
- allow_ip6_prefix
  [입력/출력]
  제품에 접속 가능한 호스트의 IP6 주소에 대한 서브넷 접두사 길이 입니다.

2.7.3 주의

- 다음의 변수들은 빅 엔디언(Network Byte Order)을 사용합니다.
  allow_ip, allow_subnet.

2.8 io_env

2.8.1 개요

- I/O 제품관련 환경변수를 저장 할 구조체 입니다.

```
#define MAX_DI                  8
#define MAX_DO                  8
#define IO_SCRIPT_LEN           32
#define IO_COMMENT_LEN          16

struct io_env
{
        _u8             di_num;                 // 읽기 전용. 값을 변경하지 마십시오.
        _u8             do_num;                 // 읽기 전용. 값을 변경하지 마십시오.
        _u16            html_size;
        _u8             script          [MAX_DO][IO_SCRIPT_LEN];
        _u8             host_name       [64];
        _u8             di_comment      [MAX_DI][IO_COMMENT_LEN];
        _u8             do_comment      [MAX_DO][IO_COMMENT_LEN];
        struct io_var_env io_var_env;
        _u32            use_flag;       // 읽기 전용. 값을 변경하지 마십시오.
};
```

2.8.2 변수

- di_num
  [출력] 사용 가능한 디지털 입력 포트 개수를 나타냅니다.
  읽기 전용 변수 입니다.
- do_num
  [출력] 사용 가능한 디지털 출력 포트 개수를 나타냅니다.
  읽기 전용 변수 입니다.
- html_size
  [입력/출력] HTML을 위한 메모리 사이즈 [단위: Kilobyte, KB].
  html_sizes에 설정 가능한 값은 80, 96, 112 중 하나 입니다. 기본값은 80입니다.
- script
  [입력/출력] 디지털 출력 포트를 위한 매크로.
  자세한 설명은 제품 사용자 설명서를 참고 하십시오.

- host_name
  [입력/출력] 제품이 접속할 호스트의 DNS 이름.
  이 변수는 ezTCP가 TCP 클라이언트로 작동되는 경우 사용 됩니다. ezTCP가 접속할 호스트 주소를 DNS 이름으로 사용하려면 이 변수에 DNS 이름을 입력하고 “io_var_env” 구조체의 “peer_ip” 변수는 0으로 설정하여야 합니다.
- di_comment
  [입력/출력]
  디지털 입력 포트에 대한 간단한 설명을 입력할 수 있습니다.
- do_comment
  [입력/출력]
  디지털 출력 포트에 대한 간단한 설명을 입력할 수 있습니다.
- io_var_env
  [입력/출력]
  디지털 입력/출력 포트에 관련된 환경변수를 저장 할 구조체 입니다.
- use_flag
  [출력]
  “io_env”구조체 사용 여부를 나타냅니다.
  읽기 전용 변수 입니다. 이 변수가 1인 경우에만 유효한 I/O 제품 입니다.

2.9 io_var_env

2.9.1 개요

- 디지털 입력/출력 포트에 관련된 환경변수를 저장 할 구조체 입니다.

```
struct io_var_env
{
        unsigned int        modbus              : 1;
        unsigned int        macro               : 1;
        unsigned int        master              : 1;
        unsigned int        active              : 1;
        unsigned int        notify              : 1;
        unsigned int        conns               : 3;
        unsigned int        emacro              : 8;
        unsigned int        query               : 1;
        unsigned int        ctrl                : 1;
        unsigned int        pad0                : 6;    /* 사용하지 않음. 값을 사용하지 마십시오. */
        unsigned int        pad1                : 8;    /* 사용하지 않음. 값을 사용하지 마십시오. */
        _u32                peer_ip;
        _u16                peer_port;
        _u16                slave_id;
        _u16                input_addr;
        _u16                output_addr;
        _u32                init_output;
        _u32                poll_interval;
        _u16                input_valid_time[8];
        _u16                output_delay[8];
};
```

2.9.2 변수

- modbus
  [입력/출력]
  modbus옵션을 설정하면 제품의 I/O포트를 Modbus/TCP 프로토콜을 사용하여 제어 합니다.
- macro
  [입력/출력]
  macro옵션을 설정하면 제품의 디지털 출력 포트를 매크로 기능을 사용하여 제어 합니다.

- master
  [입력/출력]

master	설명
0	Modbus/TCP 슬레이브
1	Modbus/TCP 마스터

- active
  [입력/출력]

active	설명
0	수동접속 (TCP 서버)
1	능동접속 (TCP 클라이언트)

- notify
  [입력/출력] 입력포트 변경 알림.
  notify_input옵션을 설정하면, 디지털 입력 포트에 입력 값이 변경되는 경우 입력 포트 데이터를 Modbus/TCP를 사용하여 마스터로 전송 합니다.
- conns
  [입력/출력] Modbus/TCP 다중접속 개 수
  Modbus/TCP를 사용하는 경우 한번에 접속 가능한 TCP/IP 접속 개수를 선택할 수 있습니다. 최대 8개까지 가능합니다.
  ※ **펌웨어 버전 1.3F 이상에서 지원합니다.**
- emacro
  [입력/출력]
  각 디지털 출력 포트 별로 매크로 기능 사용을 설정합니다.
  LSB부터 포트 0번이고 현재8비트만 사용합니다. 비트가 1이면 사용, 0이면 사용 안 함 입니다.

- query
  [입력/출력]
  master 변수를 1(Modbus/TCP 마스터)로 선택한 경우, 제품이 사용할 제어 명령어 종류를 선택합니다.

query	설명
0	**FC 16(동시제어)** Modbus/TCP 클래스0의 16번 함수인 write multiple registers를 이용해 출력포트를 일괄적으로 제어하고 03번 함수인 read multiple registers를 이용해 입력포트를 감시합니다.
1	**FC 05(개별제어)** 입력포트 감시는 02번 함수인 read input discretes를 사용하고 출력포트 제어는 FC 05번 함수인 write coil을 사용합니다. 이 write coil 함수는 각각의 출력포트를 개별적으로 제어할 수 있게 해 줍니다.

- ctrl
  [입력/출력]
  master 변수를 1(Modbus/TCP 마스터)로 선택한 경우, 제품 출력포트 제어 방식을 선택합니다.

ctrl	설명
0	논리곱
1	논리합

- peer_ip
  [입력/출력]
  active 변수를 1(능동접속)로 선택한 경우 제품이 통신 할 서버의 IP 주소를 설정합니다.

- peer_port
  [입력/출력]
  active 변수를 1(능동접속)로 선택한 경우 제품이 통신 할 서버의 포트번호를 설정합니다.
  active 변수를 0(수동접속)로 선택한 경우 제품의 로컬 포트번호로 사용됩니다.

- slave_id
  [입력/출력] 유니트 아이디
  master변수 따라서 다음의 의미를 가집니다.

master	설명
0 - Modbus/TCP 슬레이브	제품의 유니트 아이디
1 - Modbus/TCP 마스터	원격장비의 유니트 아이디

- input_addr
  [입력/출력] 입력 포트 주소
  master변수 따라서 다음의 의미를 가집니다.

master	설명
0 - Modbus/TCP 슬레이브	제품의 입력 포트 주소
1 - Modbus/TCP 마스터	슬레이브 장비의 입력 포트 주소

*** 입력 포트 주소와 출력 포트 주소는 8이상 차이가 나야 합니다.**

- output_addr
  [입력/출력] 출력 포트 주소
  master변수 따라서 다음의 의미를 가집니다.

master	설명
0 - Modbus/TCP 슬레이브	제품의 출력 포트 주소
1 - Modbus/TCP 마스터	슬레이브 장비의 출력 포트 주소.

*** 입력 포트 주소와 출력 포트 주소는 8이상 차이가 나야 합니다.**

- init_output
  [입력/출력] 출력 포트의 초기상태.
  디지털 출력 포트의 초기상태를 설정합니다. LSB부터 포트 0번이고 현재8 비트만 사용합니다. 비트가 1이면 켜지고 0이면 꺼집니다.

- poll_interval
  [입력/출력]
  슬레이브 장비로부터 데이터를 읽는 주기를 설정합니다. [단위: 밀리 초]

- input_valid_time
  [입력/출력]
  ※ 이 변수에 대한 자세한 설명은 제품 사용자 설명서를 참고 하십시오.
- output_delay
  [입력/출력]
  ※ 이 변수에 대한 자세한 설명은 제품 사용자 설명서를 참고 하십시오.

2.9.3 주의

- 다음의 변수들은 빅 엔디언(Network Byte Order)을 사용합니다.
  peer_ip

2.10 ip_trap_env

2.10.1 개요

- IP 주소 통보 기능에 관련된 환경변수를 저장 할 구조체 입니다.
  ※ **IP 주소 통보 기능에 대한 자세한 정보는 IP 주소 통보 기능 기술문서를 참고 하십시오.**

```
struct ip_trap_env
{
        unsigned int        level      : 3;
        unsigned int        pad0       : 13;     /* 사용하지 않음. 값을 사용하지 마십시오. */
        unsigned int        pad1       : 16;     /* 사용하지 않음. 값을 사용하지 마십시오. */
        _u32                peer_ip;
        _u16                peer_port;
        _u16                interval;
};
```

2.10.2 변수

- level
  [입력/출력]
  "opt_env" 구조체의 "ddns"변수가 2(TCP) 또는 3(UDP)인 경우 제품이 전송하는 IP 주소 관련 데이터의 형식을 지정합니다.

level	설명
0	ASCII
1	바이너리

- peer_ip
  [입력/출력] 제품의 IP 주소 정보를 수신 할 호스트의 IP 주소.
- peer_port
  [입력/출력] 제품의 IP 주소 정보를 수신 할 호스트의 포트번호.
- interval
  [입력/출력] 제품의 IP 주소 정보를 전송하는 주기 [단위 : 분].

2.10.3 주의

- 다음의 변수들은 빅 엔디언(Network Byte Order)을 사용합니다.
  peer_ip

2.11 port_map_env

2.11.1 개요

- 제품의 TCP포트 관련 환경변수를 저장 할 구조체 입니다.

```
struct port_map_env
{
        _u16    http_port;
        _u16    reserved1;      /* 사용하지 않음. 값을 사용하지 마십시오. */
        _u16    reserved2;      /* 사용하지 않음. 값을 사용하지 마십시오. */
        _u16    reserved3;      /* 사용하지 않음. 값을 사용하지 마십시오. */
        _u16    reserved4;      /* 사용하지 않음. 값을 사용하지 마십시오. */
};
```

2.11.2 변수

- http_port
  [입력/출력] HTTP 포트번호

  이 변수는 아래의 제품만 유효 합니다.

LAN 종류	제품명
유선랜	CIE-H10, CIE-M10, CIE-H12
무선랜	

2.12 wlan_env

2.12.1 개요

- 무선랜 관련 환경변수를 저장 할 구조체 입니다.

```
struct wlan_env
{
        struct wlan_opt     wlan_opt_env;
        struct wlan_var     wlan_var_env;
        _u32                use_flag;          // 읽기 전용. 값을 변경하지 마십시오.
};
```

2.12.2 변수

- wlan_opt_env
  [입력/출력] 무선랜 관련 환경변수를 저장 할 구조체 입니다.
- wlan_var_env
  [입력/출력] 무선랜 관련 환경변수를 저장 할 구조체 입니다.
- use_flag
  [출력] 무선랜 사용 여부를 나타냅니다.
  읽기 전용 변수 입니다. 이 변수가 1인 경우에만 유효한 무선랜 제품 입니다.

2.13 wlan_opt_env

2.13.1 개요

- 무선랜 관련 환경변수를 저장 할 구조체 입니다.

```
struct wlan_opt
{
        unsigned int    cctype          : 4;
        unsigned int    channel         : 4;
        unsigned int    wep             : 2;
        unsigned int    wep_id          : 2;
        unsigned int    pad0            : 1;
        unsigned int    bg_scan         : 1;
        unsigned int    auth            : 2;
        unsigned int    wpa             : 3;
        unsigned int    cipher          : 2;
        unsigned int    pad1            : 3;    /* 사용하지 않음. 값을 사용하지 마십시오. */
        unsigned int    antenna         : 1;
        unsigned int    phy             : 3;
        unsigned int    short_preamble  : 1;
        unsigned int    short_slot      : 1;
        unsigned int    cts_protection  : 1;
        unsigned int    pad2            : 1;    /* 사용하지 않음. 값을 사용하지 마십시오. */
};
```

2.13.2 변수

- cctype
  [입력/출력] 무선랜 작동 방식

cctype	설명
0	애드혹
1	인프라스트럭처

- channel
  [입력/출력] 무선랜 채널번호
  cctype이 0(애드혹)인 경우에만 유효합니다.

- wep
  [입력/출력] 무선랜 암호화 방식

wep	설명
0	사용안함
1	WEP – 64비트 키

2	WEP – 128비트 키

※ WEP(Wired Equivalent Privacy)

- wep_id
  [입력/출력] WEP키 아이디 (0, 1, 2, 3)
- auth
  [입력/출력] 무선랜 인증방식

auth	설명
0	사용안함
1	개방모드
2	공유모드
3	혼합모드

- wpa
  [입력/출력] WPA(Wi-Fi Protected Access) 인증방식

wpa	설명
0	사용안함
1	EAP TLS
2	WPA-PSK
3	EAP TTLS
4	WPA2-PSK
5	PEAP

- cipher
  [입력/출력] WPA 암호와 방법

cipher	설명
0	사용안함
1	TKIP (Temporal Key Integrity Protocol)
2	AES (Advanced Encryption Standard)
3	TKIP/AES

- antenna
  [입력/출력] 안테나를 선택합니다.

antenna	설명
0	내장 안테나
1	외장 안테나

※ 안테나 설정은 CSW-M85만 지원 합니다.

- passive, bg_scan, phy, short_preamble, short_slot, cts_protection
  [입력/출력]
  무선랜 작동에 관한 고급설정입니다. 자세한 내용은 해당제품 사용자 설명서를 참고하십시오.

2.14 wlan_var_env

2.14.1 개요

- 무선랜 관련 환경변수를 저장 할 구조체 입니다.

```
struct wlan_var
{
        _u8     ssid                [32];
        _u8     key40               [4][5];
        _u8     key104              [4][13];
        _u8     wpa_passphrase      [64];
        _u8     wpa_psk             [32];     // 읽기 전용. 값을 변경하지 마십시오.
        _u8     power_table         [16];     // 읽기 전용. 값을 변경하지 마십시오.
        _u8     pad0                [3];      /* 사용하지 않음. 값을 사용하지 마십시오. */
        _u8     key_flag;
        _u8     eap_id              [32];
        _u8     eap_pwd             [16];
};
```

2.14.2 변수

- ssid
  [입력/출력] 액세스 포인트(무선랜장비)의 SSID
  “ssid”는 0x00으로 끝나는 ASCII 데이터 입니다. 따라서, 최대 31바이트까지 사용 가능 합니다.
- key40
  [입력/출력]
  “wep” 변수가 1(WEP – 64비트 키)인 경우에 사용되는 키 입니다.
- key104
  [입력/출력]
  “wep” 변수가 2(WEP – 128비트 키)인 경우에 사용되는 키 입니다.

- wpa_passphrase
  [입력/출력] WPA(Wi-Fi Protected Access) 암호문
  "wpa_passphrase"는 0x00으로 끝나는 ASCII 데이터 입니다. 따라서, 최대 63바이트까지 사용 가능 합니다. "wpa_psk"를 생성하는데 사용되는 값으로 최소 8바이트 이상 입력해야 합니다. 허용되는 문자는 "A~Z", "a~z" 그리고 "0" ~ "9" 입니다.

제품	WPA 암호문 최대 길이
CSW-H80	31바이트
CSW-M83 / M85	63바이트

- wpa_psk
  [출력] WPA(Wi-Fi Protected Access) PSK(Pre-shared Key).
  "ssid"와 "wpa_passphrase"를 사용하여 계산됩니다.
  읽기 전용 변수 입니다.
  "EzManager_Write" 함수를 사용 하여 환경변수를 저장할 때 라이브러리가 "wpa_psk" 값을 자동으로 계산합니다.

- power_table
  [출력] 채널 별 무선랜 신호세기 설정
  각 무선랜 채널 별 신호세기를 설정할 수 있습니다.
  ※ 읽기 전용 변수 입니다. 현재 지원하지 않습니다.

- key_flag
  [출력] WEP 키 코드구분

key_flag	설명
0	16진수 코드
1	ASCII 코드

- eap_id, eap_pwd
  [입력/출력]
  무선랜 보안설정이 EAP TLS, EAP TTLS 또는 PEAP인 경우 사용할 사용자 아이디와 비밀번호 입니다.
  자세한 내용은 해당제품 사용자 설명서를 참고하십시오.

2.15 csc_hr2_env

2.15.1 개요

- CSC-HR2 관련 환경변수를 저장 할 구조체 입니다.

```
struct csc_hr2_env
{
        struct uart_dev_env         uart_dev_env;
        _u8                         mux_type;
        _u8                         pad                 [3];
        _u8                         csc_hr2_id          [16];
        _u8                         server_host_name1   [64];
        _u8                         server_host_name2   [64];
        struct redundancy_var_env   redundancy_var_env;
};
```

2.15.2 변수

- uart_dev_env
  [입력/출력]
  UART 하드웨어 설정에 관련된 환경변수를 저장 할 구조체 입니다. **2.4 uart_dev_env**를 참조 하십시오.
- mux_type
  [입력/출력] CSC-HR2 통신모드

mux_type	설명
0	- 자동모드 항상 TCP 클라이언트 모드로 동작하며, 설정된 서버로 접속을 시도 합니다.
1	- EZU-100 펌웨어 변경모드 3G 연결 모듈의 펌웨어를 변경하는 모드 입니다. 펌웨어 파일 전송은 RS232 포트로 이루어집니다.

- csc_hr2_id
  [입력/출력] CSC-HR2의 아이디
  서버에서 여러 대의 CSC-HR2를 구분할 수 있는 아이디를 설정할 수 있습니다.

- server_host_name1
  [입력/출력] 첫 번째 서버 주소
  CSC-HR2가 접속해야 할 첫 번째 서버 주소 입니다. 이 변수를 사용하려면 redundancy_var_env의 peer_ip[0]은 0(영)값을 설정해야 합니다.
- server_host_name2
  [입력/출력] 두 번째 서버 주소
  CSC-HR2가 접속해야 할 두 번째 서버 주소 입니다. 이 변수를 사용하려면 redundancy_var_env의 peer_ip[1]은 0(영)값을 설정해야 합니다.
- redundancy_var_env
  [입력/출력]
  CSC-HR2 이중화 관련 환경변수를 저장 할 구조체 입니다.

2.16 redundancy_var_env

2.16.1 개요

- CSC-HR2 이중화 관련 환경변수를 저장 할 구조체 입니다.

```
struct redundancy_var_env
{
        _u32    peer_ip         [2];
        _u16    peer_port       [2];
        _u16    timeout         [2];
        _u16    threshold       [2];
        _u16    check_port;
        _u16    pad0;
};
```

2.16.2 변수

- peer_ip[0]
  [입력/출력] 첫 번째 서버 IP주소
  CSC-HR2가 접속해야 할 첫 번째 서버 IP주소 입니다. 이 변수가 0(영)이 아니면 csc_hr2_env의 server_host_name1은 사용하지 않습니다.
- peer_ip[1]
  [입력/출력] 두 번째 서버 IP주소
  CSC-HR2가 접속해야 할 두 번째 서버 IP주소 입니다. 이 변수가 0(영)이 아니면 csc_hr2_env의 server_host_name2은 사용하지 않습니다.
- peer_port[0]
  [입력/출력] 첫 번째 서버 포트번호
- peer_port[1]
  [입력/출력] 두 번째 서버 포트번호
- timeout[0]
  [입력/출력] 절체 타임아웃 [단위: 초]
  유선랜에서 3G로 네트워크를 절체 할 때 기준이 되는 시간입니다.
- timeout[1]
  [입력/출력] 서버변경 타임아웃 [단위: 초]
  접속을 시도할 서버를 변경 할 때 기준이 되는 시간입니다.

- threshold[0]
  [입력/출력] 절체 바이트 수 [단위: 바이트]
  유선랜에서 3G로 네트워크를 절체 할 때 기준이 되는 데이터통신 량입니다.
- threshold[1]
  [입력/출력] 서버변경 바이트 수 [단위: 바이트]
  접속을 시도할 서버를 변경 할 때 기준이 되는 데이터통신 량 입니다.
  ※ 자세한 내용은 제품 사용자 설명서를 참고하십시오.
- check_port
  [입력/출력] CSC-HR2 통신품질 확인을 위한 TCP 포트번호 입니다.

2.17 tcp_status_env

2.17.1 개요

- 제품의 TCP/IP (IPv4) 세션 상태정보를 저장 할 구조체 입니다.

```
struct tcp_status_env
{
        _u8     stat;           // 읽기 전용. 값을 변경하지 마십시오.
        _u8     pad1;           /* 사용하지 않음. 값을 사용하지 마십시오. */
        _u16    winq_out;       /* 사용하지 않음. 값을 사용하지 마십시오. */
        char    name    [8];    // 읽기 전용. 값을 변경하지 마십시오.
        _u32    local_ip;       // 읽기 전용. 값을 변경하지 마십시오.
        _u32    peer_ip;        // 읽기 전용. 값을 변경하지 마십시오.
        _u16    local_port;     // 읽기 전용. 값을 변경하지 마십시오.
        _u16    peer_port;      // 읽기 전용. 값을 변경하지 마십시오.
};
```

2.17.2 변수

- stat

[출력] TCP/IP(IPv4) 접속 상태

stat	설명	
0	CLOSED	접속이 종료된 상태 입니다.
1	LISTEN	접속 대기중인 상태 입니다.
2	SYN_SENT	SYN 패킷을 전송한 상태입니다.
3	SYN_RCVD	SYN 패킷을 수신한 상태 입니다.
4	ESTABLISHED	접속된 상태 입니다.
5	FIN_WAIT1	FIN 패킷을 전송한 상태입니다.
6	FIN_WAIT2	FIN_WAIT1 상태에서 ACK 패킷을 수신한 상태 입니다.
7	CLOSE_WAIT	접속된 상태에서 FIN 패킷을 수신한 상태 입니다.
8	CLOSING	FIN_WAIT1 상태에서 FIN 패킷을 수신한 상태 입니다.
9	LAST_ACK	CLOSE_WAIT 상태에서 FIN 패킷을 전송한 상태 입니다.
10	TIME_WAIT	새로운 세션을 시작하기 위해서 대기 중 입니다.
11	END	접속이 종료된 상태 입니다.

- name
  [출력] TCP/IP 세션이름 입니다.
- local_ip
  [출력] 제품 IPv4 주소 입니다.
- peer_ip
  [출력] 원격호스트의 IPv4 주소 입니다.
- local_port
  [출력] 제품의 로컬 포트번호 입니다.
- peer_port
  [출력] 원격호스트의 포트번호 입니다.

2.18 tcp6_status_env

2.18.1 개요

- 제품의 TCP/IP(IPv6) 세션 상태정보를 저장 할 구조체 입니다.

```
struct tcp6_status_env
{
        _u8     stat;               // 읽기 전용. 값을 변경하지 마십시오.
        _u8     pad1;               /* 사용하지 않음. 값을 사용하지 마십시오. */
        _u16    winq_out;           /* 사용하지 않음. 값을 사용하지 마십시오. */
        char    name    [8];        // 읽기 전용. 값을 변경하지 마십시오.
        _u8     local_ip6  [16];    // 읽기 전용. 값을 변경하지 마십시오.
        _u8     peer_ip6   [16];    // 읽기 전용. 값을 변경하지 마십시오.
        _u16    local_port;         // 읽기 전용. 값을 변경하지 마십시오.
        _u16    peer_port;          // 읽기 전용. 값을 변경하지 마십시오.
};
```

2.18.2 변수

- stat

  [출력] TCP/IP(IPv6) 접속 상태

stat	설명	
0	CLOSED	접속이 종료된 상태 입니다.
1	LISTEN	접속 대기중인 상태 입니다.
2	SYN_SENT	SYN 패킷을 전송한 상태입니다.
3	SYN_RCVD	SYN 패킷을 수신한 상태 입니다.
4	ESTABLISHED	접속된 상태 입니다.
5	FIN_WAIT1	FIN 패킷을 전송한 상태입니다.
6	FIN_WAIT2	FIN_WAIT1 상태에서 ACK 패킷을 수신한 상태 입니다.
7	CLOSE_WAIT	접속된 상태에서 FIN 패킷을 수신한 상태 입니다.
8	CLOSING	FIN_WAIT1 상태에서 FIN 패킷을 수신한 상태 입니다.
9	LAST_ACK	CLOSE_WAIT 상태에서 FIN 패킷을 전송한 상태 입니다.
10	TIME_WAIT	새로운 세션을 시작하기 위해서 대기 중 입니다.
11	END	접속이 종료된 상태 입니다.

- name
  [출력] TCP/IP 세션이름 입니다.
- local_ip6
  [출력] 제품 IPv6 주소 입니다.
- peer_ip6
  [출력] 원격호스트의 IPv6 주소 입니다.
- local_port
  [출력] 제품의 로컬 포트번호 입니다.
- peer_port
  [출력] 원격호스트의 포트번호 입니다.

3 함수

3.1 EzManager_Search

3.1.1 개요

- 로컬 또는 원격지 네트워크에서 제품을 검색하는 함수 입니다.
- 검색된 제품은 MAC 주소 또는 IP 주소로 구분 됩니다.

3.1.2 함수원형

```
int EzManager_Search(
        int             mode,
        _u32            ip,
        struct lib_env  *lib_env,
        int             *nResultCount,
        int             *nErrNum,
        int             port =0,
        _32             bind_ip = 0
);
```

3.1.3 함수인자

- mode

  [입력]

mode	설명
0	로컬 네트워크에서 제품을 검색합니다.
1	원격 네트워크에서 제품을 검색합니다.

- ip

  [입력]

  “mode”가 1인 경우 검색할 제품의 IP 주소 입니다.

  빅 엔디언(Network Byte Order)을 사용합니다.

- lib_env

  [출력]

  검색된 제품의 환경변수를 저장할 “lib_env” 구조체 포인터 입니다.

- nResultCount
  [출력]
  검색된 제품의 개 수를 저장할 정수형 포인터 입니다.
- nErrNum
  [출력]
  함수 실행 중 에러가 발생한 경우 에러 번호를 저장할 정수형 포인터 입니다.
- port
  [입력]
  함수 실행 시 사용할 UDP 포트번호 입니다.
  UDP 포트번호를 지정하지 않으면 50005번과 50007번을 사용합니다.
- bind_ip
  [입력]
  함수 실행 시 소켓을 바인딩 할 IP주소 입니다. 바인딩 할 IP주소를 지정하지 않는 경우 운영체제에서 자동으로 할당합니다.
  빅 엔디언(Network Byte Order)을 사용합니다.

3.1.4 함수 반환 값

- 함수 실행이 정상적으로 완료되면 1을 반환 합니다.
- 함수 실행 중 에러가 발생한 경우 EZTCP_ERR(-1)을 반환하고 에러 번호를 nErrNum에 저장 합니다.

3.1.5 주의

- “EzManager_Search”함수를 실행 하기 전 “lib_env” 구조체 포인터 변수에 충분한 공간을 할당해야 합니다.
  예) lib_env = (struct lib_env *)malloc(sizeof(struct lib_env) * 256)
- **“EzManager_Search” 함수는 실행이 완료 될 때까지 2초가 소요됩니다.**
- **라이브러리를 사용하는 프로그램 종료 시 “Exit_Library” 함수를 호출하여 동적으로 할당된 메모리를 해제시켜야 합니다.**

3.2 EzManager_Read

3.2.1 개요

- 제품의 IP 주소 또는 MAC 주소를 사용하여 제품의 환경변수를 읽는 함수 입니다.

3.2.2 함수원형

```
int EzManager_Read(
        int             mode,
        _u8             *mac_addr,
        _u32            ip,
        struct lib_env  *lib_env,
        int             *nErrNum,
        int             port = 0,
        _u32            bind_ip = 0
);
```

3.2.3 함수인자

- mode

  [입력]

mode	설명
0	로컬 네트워크에서 제품의 환경변수를 읽습니다.
1	원격 네트워크에서 제품의 환경변수를 읽습니다.

- mac_addr

  [입력] 제품의 MAC 주소

  “mode”가 0인 경우 사용 됩니다.

- ip

  [입력] 제품의 IP 주소

  “mode”가 1인 경우 사용 됩니다.

  빅 엔디언(Network Byte Order)을 사용합니다.

- lib_env

  [출력]

  제품의 환경변수를 저장할 “lib_env” 구조체 포인터 입니다.

- nErrNum
  [출력]
  함수 실행 중 에러가 발생한 경우 에러 번호를 저장할 정수형 포인터 입니다.
- port
  [입력]
  함수 실행 시 사용할 UDP 포트번호 입니다.
  UDP 포트번호를 지정하지 않으면 50005번과 50007번을 사용합니다.
- bind_ip
  [입력]
  함수 실행 시 소켓을 바인딩 할 IP주소 입니다. 바인딩 할 IP주소를 지정하지 않는 경우 운영체제에서 자동으로 할당합니다.
  빅 엔디언(Network Byte Order)을 사용합니다.

3.2.4 함수 반환 값

- 함수 실행이 정상적으로 완료되면 1을 반환 합니다.
- 함수 실행 중 에러가 발생한 경우 EZTCP_ERR(-1)을 반환하고 에러 번호를 nErrNum에 저장 합니다.

3.2.5 주의

- “EzManager_Read”함수를 실행 하기 전 “lib_env” 구조체 포인터 변수에 충분한 공간을 할당해야 합니다.
  예) lib_env = (struct lib_env *)malloc(sizeof(struct lib_env) * 256)
- **“EzManager_Read” 함수는 실행이 완료 될 때까지 2초가 소요됩니다.**
- **라이브러리를 사용하는 프로그램 종료 시 “Exit_Library” 함수를 호출하여 동적으로 할당된 메모리를 해제시켜야 합니다.**

3.3 EzManager_Write

3.3.1 개요

- 제품에 환경변수를 저장하는 함수 입니다.

3.3.2 함수원형

```
int EzManager_Write(
        int                 mode,
        struct lib_env      *lib_env,
        _u8                 *cur_pwd,
        int                 *nErrNum,
        int                 port = 0,
        _u32                bind_ip = 0
);
```

3.3.3 함수인자

- mode
  [입력]

mode	설명
0	로컬 네트워크에서 환경변수를 제품에 저장 합니다.
1	원격 네트워크에서 환경변수를 제품에 저장 합니다.

- lib_env
  [입력]
  제품에 저장할 환경변수를 저장하고 있는 “lib_env” 구조체 포인터 입니다.
- cur_pwd
  [입력]
  제품에 비밀번호가 설정되어 있는 경우 현재 설정된 비밀번호가 저장되어 있는 버퍼에 대한 포인터 입니다.
- nErrNum
  [출력]
  함수 실행 중 에러가 발생한 경우 에러 번호를 저장할 정수형 포인터 입니다.

- port
  [입력]
  함수 실행 시 사용할 UDP 포트번호 입니다.
  UDP 포트번호를 지정하지 않으면 50005번과 50007번을 사용합니다.
- bind_ip
  [입력]
  함수 실행 시 소켓을 바인딩 할 IP주소 입니다. 바인딩 할 IP주소를 지정하지 않는 경우 운영체제에서 자동으로 할당합니다.
  빅 엔디언(Network Byte Order)을 사용합니다.

3.3.4 함수 반환 값

- 함수 실행이 정상적으로 완료되면 1을 반환 합니다.
- 함수 실행 중 에러가 발생한 경우 EZTCP_ERR(-1)을 반환하고 에러 번호를 nErrNum에 저장 합니다.

nErrNum	설명
EZTCP_ERR_PWD	비밀번호가 일치하지 않습니다.
EZTCP_ERR_RES	제품으로부터 응답이 없습니다.
EZTCP_ERR_LOCAL_IP	제품 IP 주소가 적절하지 않습니다.
EZTCP_ERR_CIE_H10_SCRIPT	[I/O제품] 매크로가 문법에 맞지 않습니다.
EZTCP_ERR_CIE_H10_IO_ADDR	[I/O제품] I/O 포트 주소가 적절하지 않습니다.
EZTCP_ERR_LOCAL_PORT_100	제품 로컬포트가 중복 됩니다.
EZTCP_ERR_LOCAL_PORT_101	제품 로컬포트로 23번은 사용할 수 없습니다.
EZTCP_ERR_LOCAL_PORT_102	제품 로컬포트와 Modbus/TCP 포트 번호가 중복됩니다.
EZTCP_ERR_LOCAL_PORT_103	제품 로컬포트로 80번은 사용할 수 없습니다.
EZTCP_ERR_LOCAL_PORT_104	제품 로컬포트로 50005번은 사용할 수 없습니다.

EZTCP_ERR_LOCAL_PORT_105	제품 로컬포트로 50006번은 사용할 수 없습니다.
EZTCP_ERR_PRODUCT_MISMATCH	제품 아이디가 일치하지 않습니다.
EZTCP_ERR_UNKNOWN	알 수 없는 에러가 발생했습니다.
EZTCP_ERR_NO_INFO	EzManager_Search함수를 호출하기 전에 임의로 EzManager_Write 함수를 실행하면 발생합니다.
EZTCP_ERR_NO_NETWORK	네트워크에서 데이터를 송,수신 할 수 없는 경우 발생합니다.

3.3.5 주의

- "EzManager_Write" 함수를 사용하기 전에 반듯이 "EzManager_Search" 함수를 호출하여 제품을 검색하십시오.
- "EzManager_Write" 함수는 제품으로부터의 응답을 3초간 기다립니다. 제품으로부터 응답이 없는 경우 EZTCP_ERR_RES 에러가 발생합니다.
- **라이브러리를 사용하는 프로그램 종료 시 "Exit_Library" 함수를 호출하여 동적으로 할당된 메모리를 해제시켜야 합니다.**

3.4 EzManager_Status

3.4.1 개요

- 제품 상태정보를 읽을 때 사용하는 함수 입니다. 제품 상태 정보는 일반 텍스트로 이루어져 있습니다.

3.4.2 함수원형

```
int EzManager_Status(
        int                     mode,
        struct lib_env          *lib_env,
        _u8                     *stat_buf,
        int                     stat_buf_len,
        struct tcp_status_env   *tcp_status_env,
        int                     *tcp_session_count,
        int                     *nErrNum,
        int                     port = 0,
        _u32                    bind_ip = 0
);
```

3.4.3 함수인자

- mode
  [입력]

mode	설명
0	로컬 네트워크에서 제품 상태정보를 읽습니다.
1	원격 네트워크에서 제품 상태정보를 읽습니다.

- lib_env
  [입력]
  상태정보를 읽을 제품의 환경변수가 저장되어 있는 “lib_env” 구조체 포인터 입니다.
- stat_buf
  [출력]
  제품의 상태정보를 저장할 버퍼에 대한 포인터 입니다.
- stat_buf_len
  [입력]
  제품의 상태정보를 저장할 버퍼의 사이즈 입니다.

- tcp_status_env
  [입력]
  제품의 TCP/IP 세션 상태정보를 저장할 “tcp_status_env” 구조체 포인터 입니다.
- tcp_session_count
  [출력]
  제품의 TCP/IP 세션 상태정보의 개 수 입니다.
- nErrNum
  [출력]
  함수 실행 중 에러가 발생한 경우 에러 번호를 저장할 정수형 포인터 입니다.
- port
  [입력]
  함수 실행 시 사용할 UDP 포트번호 입니다.
  UDP 포트번호를 지정하지 않으면 50005번과 50007번을 사용합니다.
- bind_ip
  [입력]
  함수 실행 시 소켓을 바인딩 할 IP주소 입니다. 바인딩 할 IP주소를 지정하지 않는 경우 운영체제에서 자동으로 할당합니다.
  빅 엔디언(Network Byte Order)을 사용합니다.

3.4.4 함수 반환 값

- 함수 실행이 정상적으로 완료되면 1을 반환 합니다.
- 함수 실행 중 에러가 발생한 경우 EZTCP_ERR(-1)을 반환하고 에러 번호를 nErrNum에 저장 합니다.

3.4.5 주의

- “EzManager_Status” 함수를 사용하기 전에 반듯이 “EzManager_Search” 함수를 호출하여 제품을 검색하십시오.

- “EzManager_Status”함수를 실행 하기 전 “tcp_status_env” 구조체 포인터 변수에 충분한 공간을 할당해야 합니다.

  예) tcp_status_env = (struct tcp_status_env *)malloc(sizeof(tcp_status_env) * MAX_TCP_SESSION)

- MAX_TCP_SESSION은 16으로 정의되어 있습니다.

- “EzManager_Status” 함수는 제품으로부터의 응답을 2초간 기다립니다. 제품으로부터 응답이 없는 경우 EZTCP_ERR_RES 에러가 발생합니다.

- **라이브러리를 사용하는 프로그램 종료 시 “Exit_Library” 함수를 호출하여 동적으로 할당된 메모리를 해제시켜야 합니다.**

3.5 EzManager_ChangePwd

3.5.1 개요

- 제품에 비밀번호를 설정하거나 삭제하는데 사용하는 함수 입니다.

3.5.2 함수원형

```
int EzManager_ChangePwd(
        int                 mode,
        struct lib_env      *lib_env,
        _u8                 *cur_pwd,
        _u8                 *new_pwd,
        int                 *nErrNum,
        int                 port = 0,
        _u32                bind_ip = 0
);
```

3.5.3 함수인자

- mode

  [입력]

mode	설명
0	로컬 네트워크에서 비밀번호를 제품에 저장 합니다.
1	원격 네트워크에서 비밀번호를 제품에 저장 합니다.

- lib_env

  [입력]

  비밀번호를 저장할 제품의 환경변수가 저장되어 있는 “lib_env” 구조체 포인터 입니다.

- cur_pwd

  [입력]

  제품에 비밀번호가 설정되어 있는 경우 현재 설정된 비밀번호가 저장되어 있는 버퍼에 대한 포인터 입니다.

- new_pwd

  [입력]

  제품에 설정할 비밀번호가 저장되어 있는 버퍼에 대한 포인터 입니다.

- nErrNum
  [출력]
  함수 실행 중 에러가 발생한 경우 에러 번호를 저장할 정수형 포인터 입니다.
- port
  [입력]
  함수 실행 시 사용할 UDP 포트번호 입니다.
  UDP 포트번호를 지정하지 않으면 50005번과 50007번을 사용합니다.
- bind_ip
  [입력]
  함수 실행 시 소켓을 바인딩 할 IP주소 입니다. 바인딩 할 IP주소를 지정하지 않는 경우 운영체제에서 자동으로 할당합니다.
  빅 엔디언(Network Byte Order)을 사용합니다.

3.5.4 함수 반환 값

- 함수 실행이 정상적으로 완료되면 1을 반환 합니다.
- 함수 실행 중 에러가 발생한 경우 EZTCP_ERR(-1)을 반환하고 에러 번호를 nErrNum에 저장 합니다.

3.5.5 주의

- “EzManager_ChangePwd”함수를 사용하기 전에 반듯이 “EzManager_Search” 함수를 호출하여 제품을 검색하십시오.
- “EzManager_ ChangePwd” 함수는 제품으로부터의 응답을 3초간 기다립니다. 제품으로부터 응답이 없는 경우 EZTCP_ERR_RES 에러가 발생합니다.
- **라이브러리를 사용하는 프로그램 종료 시 “Exit_Library” 함수를 호출하여 동적으로 할당된 메모리를 해제시켜야 합니다.**

3.6 EzManager_CloseTCP

3.6.1 개요

- 접속이 완료된 TCP/IP 세션을 종료 할 때 사용하는 함수 입니다.

3.6.2 함수원형

```
int EzManager_CloseTCP(
        int                     mode,
        struct lib_env          *lib_env,
        struct tcp_status_env   *tcp_status_env,
        _u8                     *cur_pwd,
        int                     *nErrNum,
        int                     port = 0,
        _u32                    bind_ip = 0
);
```

3.6.3 함수인자

- mode
  [입력]

mode	설명
0	로컬 네트워크에 있는 제품의 TCP/IP 세션을 종료 합니다.
1	원격 네트워크에 있는 제품의 TCP/IP 세션을 종료 합니다.

- lib_env
  [입력]
  TCP/IP 세션을 종료할 제품의 환경변수가 저장되어 있는 “lib_env” 구조체 포인터 입니다.
- tcp_status_env
  [입력]
  제품의 TCP/IP 세션정보가 저장되어 있는 “tcp_status_env” 구조체 포인터 입니다.
- cur_pwd
  [입력]
  제품에 비밀번호가 설정되어 있는 경우 현재 설정된 비밀번호가 저장되어 있는 버퍼에 대한 포인터 입니다.

- nErrNum
  [출력]
  함수 실행 중 에러가 발생한 경우 에러 번호를 저장할 정수형 포인터 입니다.
- port
  [입력]
  함수 실행 시 사용할 UDP 포트번호 입니다.
  UDP 포트번호를 지정하지 않으면 50005번과 50007번을 사용합니다.
- bind_ip
  [입력]
  함수 실행 시 소켓을 바인딩 할 IP주소 입니다. 바인딩 할 IP주소를 지정하지 않는 경우 운영체제에서 자동으로 할당합니다.
  빅 엔디언(Network Byte Order)을 사용합니다.

3.6.4 함수 반환 값

- 함수 실행이 정상적으로 완료되면 1을 반환 합니다.
- 함수 실행 중 에러가 발생한 경우 EZTCP_ERR(-1)을 반환하고 에러 번호를 nErrNum에 저장 합니다.

3.6.5 주의

- "EzManager_CloseTCP"함수를 사용하기 전 "EzManager_Status" 함수를 사용하여 제품의 TCP/IP 세션 정보를 알아내야 합니다.
- "EzManager_CloseTCP" 함수는 제품으로부터의 응답을 2초간 기다립니다. 제품으로부터 응답이 없는 경우 EZTCP_ERR_RES 에러가 발생합니다.
- **라이브러리를 사용하는 프로그램 종료 시 "Exit_Library" 함수를 호출하여 동적으로 할당된 메모리를 해제시켜야 합니다.**

3.7 EzManager_RdbOnOff

3.7.1 개요

- 디버깅 메시지 전송 기능을 시작/정지 할 때 사용하는 함수 입니다.
- ezTCP는 부팅 후 디버깅 메시지 전송 기능이 설정 되어 있으면 디버깅 메시지를 네트워크로 브로드캐스팅 합니다.
- EzManager_RdbOnOff 함수를 사용하여 브로드캐스팅 되는 디버깅 메시지를 특정 IP 주소로 전송하거나 중지 시킬 수 있습니다.
- 이 함수는 아래의 제품만 유효 합니다.

LAN 종류	제품명
유선랜	CSE-M53
무선랜	

3.7.2 함수원형

```
int EzManager_RdbOnOff(
        int         mode,
        _u8         *mac_addr,
        _u32        eztcp_ip,
        BOOL        onoff,
        _u32        rcvd_ip,
        int*        nErrNum,
        int         port = 0,
        _u32        bind_ip = 0
);
```

3.7.3 함수인자

- mode

  [입력]

mode	설명
0	로컬 네트워크에 있는 제품의 디버깅 메시지 전송 기능을 시작/정지 합니다.
1	원격 네트워크에 있는 제품의 디버깅 메시지 전송 기능을 시작/정지 합니다.

- mac_addr
  [입력]
  디버깅 메시지 전송 기능을 시작/정지 할 제품의 MAC 주소 입니다.
  “mode”가 0인 경우 사용 됩니다.
- eztcp_ip
  [입력]
  디버깅 메시지 전송 기능을 시작/정지 할 제품의 IP 주소 입니다.
  “mode”가 1인 경우 사용 됩니다.
  빅 엔디언(Network Byte Order)을 사용합니다.
- onoff
  [입력]

onoff	설명
0	디버깅 메시지 전송 기능을 정지 시킵니다.
1	디버깅 메시지 전송 기능을 시작 시킵니다.

- rcvd_ip
  [입력]
  디버깅 메시지를 수신할 PC의 IP 주소 입니다.
  빅 엔디언(Network Byte Order)을 사용합니다.
- nErrNum
  [출력]
  함수 실행 중 에러가 발생한 경우 에러 번호를 저장할 정수형 포인터 입니다.
- port
  [입력]
  함수 실행 시 사용할 UDP 포트번호 입니다.
  UDP 포트번호를 지정하지 않으면 50005번과 50007번을 사용합니다.
- bind_ip
  [입력]
  함수 실행 시 소켓을 바인딩 할 IP주소 입니다. 바인딩 할 IP주소를 지정하지 않는 경우 운영체제에서 자동으로 할당합니다.
  빅 엔디언(Network Byte Order)을 사용합니다.

3.7.4 함수 반환 값

- 함수 실행이 정상적으로 완료되면 1을 반환 합니다.
- 함수 실행 중 에러가 발생한 경우 EZTCP_ERR(-1)을 반환하고 에러 번호를 nErrNum에 저장 합니다.

3.7.5 주의

- “EzManager_RdbOnOff” 함수는 제품으로부터의 응답을 2초간 기다립니다. 제품으로부터 응답이 없는 경우 EZTCP_ERR_RES 에러가 발생합니다.
- **라이브러리를 사용하는 프로그램 종료 시 “Exit_Library” 함수를 호출하여 동적으로 할당된 메모리를 해제시켜야 합니다.**

3.8 EzManager_ReBoot

3.8.1 개요

- 제품을 리부팅 할 때 사용하는 함수 입니다.
- 이 함수는 아래의 제품만 유효 합니다.

LAN 종류	제품명
유선랜	CSE-M53
무선랜	

3.8.2 함수원형

```
int EzManager_ReBoot(
        int         mode,
        _u8         *mac_addr,
        _u32        eztcp_ip,
        int         *nErrNum,
        int         port = 0,
        _u32        bind_ip = 0
);
```

3.8.3 함수인자

- mode

  [입력]

mode	설명
0	로컬 네트워크에 있는 제품을 리부팅 시킵니다.
1	원격 네트워크에 있는 제품을 리부팅 시킵니다.

- mac_addr

  [입력]

  리부팅 시킬 제품의 MAC 주소 입니다.

  “mode”가 0인 경우 사용 됩니다.

- eztcp_ip

  [입력]

  리부팅 시킬 제품의 IP 주소 입니다.

  “mode”가 1인 경우 사용 됩니다.

  빅 엔디언(Network Byte Order)을 사용합니다.

- nErrNum
  [출력]
  함수 실행 중 에러가 발생한 경우 에러 번호를 저장할 정수형 포인터 입니다.
- port
  [입력]
  함수 실행 시 사용할 UDP 포트번호 입니다.
  UDP 포트번호를 지정하지 않으면 50005번과 50007번을 사용합니다.
- bind_ip
  [입력]
  함수 실행 시 소켓을 바인딩 할 IP주소 입니다. 바인딩 할 IP주소를 지정하지 않는 경우 운영체제에서 자동으로 할당합니다.
  빅 엔디언(Network Byte Order)을 사용합니다.

3.8.4 함수 반환 값

- 함수 실행이 정상적으로 완료되면 1을 반환 합니다.
- 함수 실행 중 에러가 발생한 경우 EZTCP_ERR(-1)을 반환하고 에러 번호를 nErrNum에 저장 합니다.

3.8.5 주의

- “EzManager_ReBoot” 함수는 제품으로부터의 응답을 2초간 기다립니다. 제품으로부터 응답이 없는 경우 EZTCP_ERR_RES 에러가 발생합니다.
- **라이브러리를 사용하는 프로그램 종료 시 “Exit_Library” 함수를 호출하여 동적으로 할당된 메모리를 해제시켜야 합니다.**

3.9 GetLibVer

3.9.1 개요

- 라이브러리 버전 정보를 확인하는 함수 입니다.

3.9.2 함수원형

```
char* GetLibVer(int* len);
```

3.9.3 함수인자

- len
  [출력]
  라이브러리 버전 정보 문자열 길이를 저장할 정수형 포인터 입니다.

3.9.4 함수 반환 값

- 라이브러리 버전 정보 문자열 포인터가 반환 됩니다.

3.10 GetProductName

3.10.1 개요

- 제품명을 조회하는 함수 입니다.

3.10.2 함수원형

```
char* GetProductName(int product_id, int* len);
```

3.10.3 함수인자

- product_id
  [입력] 제품 아이디
  “net_env” 구조체 “product_id_old” 또는 “product_id_new” 변수를 사용 합니다. “product_id_new” 변수가 0인 경우 “product_id_old” 변수를 사용하십시오.
- len
  [출력] The length of product name string
  제품명 문자열 길이를 저장할 정수형 포인터 입니다.

3.10.4 함수 반환 값

- 제품명 문자열 포인터가 반환 됩니다.

3.11 Exit_Library

3.11.1 개요

- ezManager 라이브러리는 데이터 저장을 위해서 동적으로 메모리를 할당하여 사용합니다. 따라서, 프로그램 종료 시 “Exit_Library” 함수를 호출하여 동적으로 할당된 메모리를 해제해야 합니다.

3.11.2 함수원형

```
void Exit_Library(void);
```