

ezTCP/Ethernet series

ezTCP コンフィグライブラリ

ezConfig VC++Library (Static Link)

ライブラリ仕様書

第 4 版

Alpha Project Co., Ltd.

<https://www.apnet.co.jp>

ご使用になる前に

このたびは、「ezConfig Library」をお使い頂きまして誠に有り難うございます。ソフトウェアをお役立て頂くために、このライブラリ仕様書を十分お読み下さいますようお願いいたします。今後共、弊社製品をご愛顧賜りますよう宜しくお願いいたします。

使用上の注意

- 本ソフトウェア^{※1}と仕様書^{※2}の利用または使用に関連して生じた一切の損害（逸失利益、機密情報若しくはその他の情報喪失、事業の中断を含む）について、当社ではその責任を負わず、また損害賠償責任を負いません。
- 本ソフトウェアと仕様書は、何等の保証なく提供されています。明示、黙示、または法律上の保証であると否とを問わず、本ソフトウェアと仕様書について、瑕疵がないこと、特定目的に対する適合性、ウィルスの不存在、応答的的確性、使用結果のいかんを含めて、一切の保証を行いません。
- 本ソフトウェアは、弊社取り扱いの製品「ezTCP シリーズ」とともに利用されるソフトウェアであり、その他の製品及びその他の目的のために利用することを禁止します。
- 本ソフトウェアを組み込んで作成されたアプリケーションプログラム等を弊社製品「ezTCP シリーズ」とともに利用する目的において再配布することを許可します。ただし、本ソフトウェア及びサンプルプログラム、仕様書の一部または全部を二次利用可能な形式（ライブラリ形式、ソースコード等）で再配布する事は禁止します。

※1 「ソフトウェア」: Sollae System 製「ezConfig Library」のことを示しています。

※2 「仕様書」: 本文書である ALPHA PROJECT 製「ライブラリ仕様書」のことを示しています。

動作確認

- Microsoft 社の VC++6.0[®]での動作を確認しています。

目 次

1. 製品概要	1
1.1 ライブラリの構成.....	1
1.2 製品区分	1
2. ライブラリをご使用になる前に	2
2.1 使用上の注意	2
2.2 導入に際して	2
3. eztcplib.h	3
3.1 ご使用に際して	3
3.2 spe_env 構造体	4
3.3 wlan_env.....	14
3.4 etc_opt.....	18
3.5 stat_env	19
4. eztcplib.lib	24
4.1 ProbeEzTCP.....	24
4.2 WriteEzTCP	26
4.3 StatusEzTCP	28
4.4 ChangePwdEzTCP.....	30
4.5 RemoteReadEzTCP.....	32
4.6 RemoteWriteEzTCP	34
4.7 RemoteStatusEzTCP	36
4.8 RemoteChangePwdEzTCP.....	38
4.9 GetLibVerEzTCP	41
5. その他	42

1. 製品概要

本ライブラリは EZL シリーズのコンフィギュレーションソフトウェアである「ezConfig」の機能をアプリケーションに組み込む為のスタティックライブラリです。本ライブラリを組み込むことにより、自作アプリケーションから EZL の設定を変更することが出来るようになります。

1.1 ライブラリの構成

本ライブラリは以下のファイルによって構成されています。

ファイル名	ファイルの説明
eztcp lib. h	ezConfig ライブラリヘッダファイル。構造体に関する記述があります。
eztcp lib. lib	ezConfig ライブラリファイル。関数に関する記述があります。

Table 1.1-1 ライブラリの構成

※リンク時には ws2_32. lib が必要となります。

使用法に関しては添付のサンプルプログラムを参考にしてください。

1.2 製品区分

本文中で述べている対応製品は次のように区別しています。

有線タイプ	
EZL-50R	EZL-50L
EZL-200A	EZL-200L
EZL-220	EZL-200F

無線タイプ	
EZL-300L	EZL-300S
EZL-80	EZL-80C
EZL-90	EZL-300W Lite

Table 1.2-1 説明文中の製品区分について

2. ライブラリをご使用になる前に

本ライブラリをご使用になる場合は次の点に注意してください。

2.1 使用上の注意

1. 「予約メンバ(reserved members)」、「未使用メンバ(not used members)」表記のある構造体は使用が認められていません。その為、これらのメンバはご使用にならないで下さい。
2. 誤った情報を書き込んだ場合、ezTCP は意図するとおりに動作しなくなります。そのため、ezTCP に対し設定を書き込む場合は、事前に ezTCP の MAC アドレスや、IP アドレスなどの設定を控えておくようお願い致します。
3. 使用によって生じた損害の保証、及び一切のサポートは行いません。

2.2 導入に際して

ご使用の際は、ソースコードに eztcp.h をインクルードし、ライブラリに ws2_32.lib と eztcp.lib を追加してください。

3. eztcplib.h

本文では、「eztcplib.h」に記述のある各構造体について説明しています。

3.1 ご使用に際して

本ヘッダファイルをご使用になる場合は次の点にご注意下さい。

○注意点

- ・本章で言及されていない値については、予約 (Reserved) されています。その為、型や、名称、言及されていない引数の命令などに関しては、変更を行わないでください。
- ・本章で説明している全ての引数は、次の例外を除き全てリトルエンディアンで格納されています。

例外 (ビッグエンディアンの引数) :

local_ip, net_mask, gate_ip, remote_ip

※枠内に示す引数に関しては、ビッグエンディアンで設定を行ってください。

List 3.1-1 引数の例外(ビッグエンディアン)

3.2 spe_env 構造体

3.2.1 概要

spe_env 構造体は機能の調査用です。状態表示や値の設定には ezCfg ライブラリを用います。

```

struct spe_env
{
    /* RESERVED SECTION. NEVER MODIFY */
    _u8 config[4];           // Reserved. Never modify.
    _u8 eth_addr[6];       // Ethernet address. Never modify.
    _u8 secure[18];        // Reserved. Never modify.
    unsigned int level;    // Reserved. Never modify.
    unsigned int eap_lock; // Reserved. Never modify.
    unsigned int en_xonoff; // Reserved. Never modify.
    unsigned int ezl50;    // Reserved. Never modify.
    unsigned int dhcp_lock; // Reserved. Never modify.
    unsigned int pppoe_lock; // Reserved. Never modify.
    unsigned int to_lock; // Reserved. Never modify.
    unsigned int lp_lock; // Reserved. Never modify.
    unsigned int rp_lock; // Reserved. Never modify.
    unsigned int wm_lock; // Reserved. Never modify.
    unsigned int rip_lock; // Reserved. Never modify.
    unsigned int en_databit; // Reserved. Never modify.
    unsigned int en_parity; // Reserved. Never modify.
    unsigned int major; // Reserved. Never modify.
    unsigned int minor; // Reserved. Never modify.

    /* network section */
    unsigned int dhcp;
    unsigned int bootp; // NOT USED. Never modify.
    unsigned int arp;
    unsigned int ssl; // Reserved. Never modify.
    unsigned int ezcfg;
    unsigned int nat; // NOT USED. Never modify.
    unsigned int pppoe;
    unsigned int stat; // Reserved. Never modify.
    unsigned int arp_probe; // NOT USED. Never modify.
    unsigned int max_mux; // Reserved. Never modify.
    unsigned int eapol; // Reserved. Never modify.
    unsigned int t2s_peer; // Reserved. Never modify.
    unsigned int pwr_down;
    unsigned int en_refg; // Read only. Never modify.
    unsigned int rcfg;
    unsigned int comment; // Read only. Never modify.
    unsigned int pad2a; // NOT USED. Never modify.
    unsigned int pad2b; // NOT USED. Never modify.
    _u32 local_ip;
    _u32 net_mask;
    _u32 gate_ip;
    _u8 poe_uid3[16]; // NOT USED. Never modify.
    _u8 poe_uid2[8]; // NOT USED. Never modify.
    _u8 passwd[12];

    /* mux/sio section */
    unsigned int parity;
    unsigned int databit;
    unsigned int rtscts;
    unsigned int xonoff;
    unsigned int telnet;

```

```
unsigned int http;           // NOT USED. Never modify.
unsigned int fcs;           // NOT USED. Never modify.
unsigned int seq_ack;       // NOT USED. Never modify.
unsigned int hdlc_emu;      // NOT USED. Never modify.
unsigned int logo;         // NOT USED. Never modify.
unsigned int oem;          // NOT USED. Never modify.
unsigned int wlancfg;      // Read only. Never modify.
unsigned int stype;
unsigned int stopbit;
unsigned int t2mc;
unsigned int pad3b;        // NOT USED. Never modify.
_u32 sio_baud;
_u16 timeout;
_u8 mux_type;
_u8 mux_pad;              // NOT USED. Never modify.
_u16 water_mark;
_u16 pad4;                // NOT USED. Never modify.
_u32 remote_ip;
_u16 remote_port;
_u16 local_port;
_u8 poe_uid[8];
_u8 poe_pwd[8];
_u32 id;                  // Reserved. Never modify.
};
```

List 3.2-1 spe_env 構造体

※List 3.2-1 で言及されない値は予約されています。

型、名前、言及されない引数の命令は変更しないでください。

不正な使用によって生じた損害の保証、及び一切のサポートは行いません。

3.2.2 引数

表の説明

①メンバー名	②型	③入出力	④対応モード	⑤対応製品
⑥内容				

※⑤対応製品の詳細については、「1.2 製品区分」を参照してください。

eth_addr				
unsigned char [6]	Read		T2S、ATC、COD	
<p>・イーサネットアドレス (MAC アドレス) を格納します。</p> <p>例：MAC アドレス 0030f9123456 は以下のように eth_addr 配列に格納されます。</p> <pre>eth_addr[0] = 0x00, eth_addr[1] = 0x30, eth_addr[2] = 0xf9, eth_addr[3] = 0x12, eth_addr[4] = 0x34, eth_addr[5] = 0x56,</pre>				

※値を変更してはいけません。また、不正な使用によって生じた損害の保証、および一切のサポートは行いません。

major				
unsigned int	Read		T2S、ATC、COD	
<p>・メジャーバージョン番号を格納します。</p>				

※値を変更してはいけません。また、不正な使用によって生じた損害の保証、および一切のサポートは行いません。

minor				
unsigned int	Read		T2S、ATC、COD	
<p>・マイナーバージョン番号を格納します。</p>				

※値を変更してはいけません。また、不正な使用によって生じた損害の保証、および一切のサポートは行いません。

dhcp			
unsigned int	Read/Write	ATC、COD	
<ul style="list-style-type: none"> この値が0以外の場合、dhcp プロトコルが有効です。 			

arp			
unsigned int	Read/Write	T2S、ATC、COD	
<ul style="list-style-type: none"> この値が0以外の場合、arp プロトコルが有効です。 			

ezcfg			
unsigned int	Read/Write	T2S、ATC、COD	
<ul style="list-style-type: none"> この値が0以外の場合、ezCFG による状態取得が有効です。本フラグがセットされていない場合、状態取得は ISP モードに限り可能です。 			

pppoe			
unsigned int	Read/Write	ATC、COD	
<ul style="list-style-type: none"> この値が0以外の場合、pppoe プロトコルが有効です。 			

pwr_down			
unsigned int	Read/Write		
<p>※この引数に対応しているのは無線タイプの製品です。</p> <ul style="list-style-type: none"> この値が0以外の場合、パワーダウンモードが有効です。 			

en_rcfg			
unsigned int	Read		
<ul style="list-style-type: none"> この値が0以外の場合、リモートコンフィグレーションをサポートしていることを示します。 ※リモートコンフィグレーションの詳細情報については、ezConfig program' s manual を参照して下さい。 			

rcfg							
unsigned int	Read/Write						
<p>・この値が0以外の場合、リモートコンフィグレーションが有効です。</p> <p>※このフラグは現在下記の製品にて対応しています。</p> <table border="1"> <thead> <tr> <th>LAN タイプ</th> <th>製品名</th> </tr> </thead> <tbody> <tr> <td>有線 LAN</td> <td>EZL-50L, EZL-200L, EZL-220, EZL-200F</td> </tr> <tr> <td>無線 LAN</td> <td>EZL-300L, EZL-300S EZL-80 / 80c / 90 / 300W Lite(ファーム名 : v1.3i もしくは、それ以上のバージョン)</td> </tr> </tbody> </table>		LAN タイプ	製品名	有線 LAN	EZL-50L, EZL-200L, EZL-220, EZL-200F	無線 LAN	EZL-300L, EZL-300S EZL-80 / 80c / 90 / 300W Lite(ファーム名 : v1.3i もしくは、それ以上のバージョン)
LAN タイプ	製品名						
有線 LAN	EZL-50L, EZL-200L, EZL-220, EZL-200F						
無線 LAN	EZL-300L, EZL-300S EZL-80 / 80c / 90 / 300W Lite(ファーム名 : v1.3i もしくは、それ以上のバージョン)						

comment							
unsigned int	Read						
<p>・この値が0以外の場合、製品がコメント機能をサポートしていることを示しています。</p> <p>※このフラグは現在下記の製品にて対応しています。</p> <table border="1"> <thead> <tr> <th>LAN タイプ</th> <th>製品名</th> </tr> </thead> <tbody> <tr> <td>有線 LAN</td> <td>EZL-50L, EZL-200L, EZL-220, EZL-200F</td> </tr> <tr> <td>無線 LAN</td> <td>EZL-300L, EZL-300S EZL-80 / 80c / 90 / 300W Lite(ファーム名 : v1.3i もしくは、それ以上のバージョン)</td> </tr> </tbody> </table>		LAN タイプ	製品名	有線 LAN	EZL-50L, EZL-200L, EZL-220, EZL-200F	無線 LAN	EZL-300L, EZL-300S EZL-80 / 80c / 90 / 300W Lite(ファーム名 : v1.3i もしくは、それ以上のバージョン)
LAN タイプ	製品名						
有線 LAN	EZL-50L, EZL-200L, EZL-220, EZL-200F						
無線 LAN	EZL-300L, EZL-300S EZL-80 / 80c / 90 / 300W Lite(ファーム名 : v1.3i もしくは、それ以上のバージョン)						

local_ip	
unsigned int	Read/Write T2S、ATC、COD
<p>・ローカル IP アドレスを格納します。IP アドレスはMK_IP マクロで設定できます。</p> <pre> /*MK_IP マクロ*/ #define MK_IP(__b1, __b2, __b3, __b4) \ (((unsigned long)(__b4) << 24) ((unsigned long)(__b3) << 16) \ ((unsigned long)(__b2) << 8) ((unsigned long)(__b1))) </pre>	

net_mask	
unsigned int	Read/Write T2S、ATC、COD
<p>・サブネットマスクを格納します。サブネットマスクはマクロで設定できます。(「3.2.2 引数」の「local_ip」参照)</p>	

gate_ip	
unsigned int	Read/Write T2S、ATC、COD
<p>・ゲートウェイ IP アドレスを格納します。ゲートウェイ IP はマクロで設定できます。(「3.2.2 引数」の「local_ip」参照)</p>	

password

unsigned char[12]

Read/Write

- ・ ezTCP にパスワードを設定する場合、設定用の関数に” password” 引数が必要になります。

parity

unsigned int

Read/Write

T2S、ATC、COD

EZL-200

- ・ シリアル通信のパリティビットを格納します。

parity	パリティ
0	なし
1	偶数パリティ
2	奇数パリティ

databit

unsigned int

Read/Write

T2S、ATC、COD

- ・ シリアル通信のデータビットを格納します。

rtscts

unsigned int

Read/Write

T2S、ATC、COD

- ・ この値が 0 以外の場合、フローコントロール (RTS/CTS) が有効です。

xonoff

unsigned int

Read/Write

T2S、ATC、COD

- ・ この値が 0 以外の場合、XON/XOFF が有効です。

※EZL-50、及びEZL-200では使用できません。

telnet							
unsigned int	Read/Write						
<ul style="list-style-type: none"> この値が0以外の場合、デバイス設定用の TELNET 接続が有効です。 <p>※このフラグは現在下記の製品にて対応しています。</p> <table border="1"> <thead> <tr> <th>LAN タイプ</th> <th>製品名</th> </tr> </thead> <tbody> <tr> <td>有線 LAN</td> <td>EZL-100, EZL-110, EZL-200A, EZL-220, EZL-200F</td> </tr> <tr> <td>無線 LAN</td> <td>EZL-300S</td> </tr> </tbody> </table>		LAN タイプ	製品名	有線 LAN	EZL-100, EZL-110, EZL-200A, EZL-220, EZL-200F	無線 LAN	EZL-300S
LAN タイプ	製品名						
有線 LAN	EZL-100, EZL-110, EZL-200A, EZL-220, EZL-200F						
無線 LAN	EZL-300S						

wlancfg	
unsigned int	Read
<ul style="list-style-type: none"> この値が0以外の場合、製品が無線 LAN の製品であることを示しています。 	

stype															
unsigned int	Read/Write														
<ul style="list-style-type: none"> シリアルポートのタイプを格納します。 <table border="1"> <thead> <tr> <th>stype</th> <th>説明</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>RS-232</td> </tr> <tr> <td>1</td> <td>RS-422</td> </tr> <tr> <td>2</td> <td>RS-485</td> </tr> </tbody> </table> <p>※このフラグは現在下記の製品にて対応しています。</p> <table border="1"> <thead> <tr> <th>LAN タイプ</th> <th>製品名</th> </tr> </thead> <tbody> <tr> <td>有線 LAN</td> <td>EZL-220, EZL-200F</td> </tr> <tr> <td>無線 LAN</td> <td>EZL-300S</td> </tr> </tbody> </table>		stype	説明	0	RS-232	1	RS-422	2	RS-485	LAN タイプ	製品名	有線 LAN	EZL-220, EZL-200F	無線 LAN	EZL-300S
stype	説明														
0	RS-232														
1	RS-422														
2	RS-485														
LAN タイプ	製品名														
有線 LAN	EZL-220, EZL-200F														
無線 LAN	EZL-300S														

stopbit							
unsigned int	Read/Write						
<ul style="list-style-type: none"> ストップビットの設定を格納します。 							
<table border="1"> <thead> <tr> <th>stopbit</th> <th>説明</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>1 stop bit</td> </tr> <tr> <td>1</td> <td>2 stop bit</td> </tr> </tbody> </table>		stopbit	説明	0	1 stop bit	1	2 stop bit
stopbit	説明						
0	1 stop bit						
1	2 stop bit						
※このフラグは現在下記の製品にて対応しています。							
LAN タイプ	製品名						
有線 LAN	EZL-220, EZL-200F						
無線 LAN	EZL-300S						

t2mc	
unsigned int	Read/Write
<ul style="list-style-type: none"> telnet フラグと、このフラグが 0 以外の場合、デバイス設定用の telnet の複数接続が有効になります。 	
※このフラグは現在下記の製品にて対応しています。	
LAN タイプ	製品名
有線 LAN	EZL-200F
無線 LAN	EZL-300S

sio_baud	
unsigned int	Read/Write
T2S、ATC、COD	
<ul style="list-style-type: none"> シリアル通信のボーレートを格納します。 	

timeout	
unsigned int	Read/Write
T2S、ATC、COD	
<ul style="list-style-type: none"> タイムアウト時間 (秒) を格納します。T2S, COD, ATC いずれかのモードにおいて、通信がない状態でタイムアウト時間が経過すると、ezTCP は通信を自動的に切断します。この値が 0 の場合、ezTCP は通信の切断を自動的には行いません。他の動作モード (T2S, COD, ATC 以外) の場合、この値は調整用に使えます。 	

mux_type																		
unsigned int	Read/Write	T2S、ATC、COD																
<p>・モードの設定を格納します。</p> <table border="1"> <thead> <tr> <th>mux_type</th> <th>モード</th> <th>説明</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>T2S</td> <td>T2S モードでは、サーバーとして動作します。この動作モードが選択されている場合、ezTCP はクライアントからの接続要求を待ちます。</td> </tr> <tr> <td>1</td> <td>ATC</td> <td>ATC モードでは AT コマンドによって動作を制御します。ezTCP は AT コマンドによって、サーバーとしても、クライアントとしても動作可能です。</td> </tr> <tr> <td>2</td> <td>COD</td> <td>COD モードでは、クライアントとして動作します。ezTCP はシリアルポートにある一定のデータ (water_mark で設定したバイト数) を受信した時点で、remote_ip と remote_port で指定されたサーバーに接続要求を行います。</td> </tr> <tr> <td>3</td> <td>U2S</td> <td>UDP モードでは、ezTCP は UDP を使えます。</td> </tr> </tbody> </table> <p>※EZL-50L, EZL-200, EZL-200L, EZL-220, EZL-200F, EZL-300S, EZL-300L, EZL-80, EZL-80C, EZL-90 は、この値を割り当てることができます。EZL-50、EZL-60 の場合は、HOTFLASH (ISP ソフトウェア) を使用してこの値を設定します。</p>				mux_type	モード	説明	0	T2S	T2S モードでは、サーバーとして動作します。この動作モードが選択されている場合、ezTCP はクライアントからの接続要求を待ちます。	1	ATC	ATC モードでは AT コマンドによって動作を制御します。ezTCP は AT コマンドによって、サーバーとしても、クライアントとしても動作可能です。	2	COD	COD モードでは、クライアントとして動作します。ezTCP はシリアルポートにある一定のデータ (water_mark で設定したバイト数) を受信した時点で、remote_ip と remote_port で指定されたサーバーに接続要求を行います。	3	U2S	UDP モードでは、ezTCP は UDP を使えます。
mux_type	モード	説明																
0	T2S	T2S モードでは、サーバーとして動作します。この動作モードが選択されている場合、ezTCP はクライアントからの接続要求を待ちます。																
1	ATC	ATC モードでは AT コマンドによって動作を制御します。ezTCP は AT コマンドによって、サーバーとしても、クライアントとしても動作可能です。																
2	COD	COD モードでは、クライアントとして動作します。ezTCP はシリアルポートにある一定のデータ (water_mark で設定したバイト数) を受信した時点で、remote_ip と remote_port で指定されたサーバーに接続要求を行います。																
3	U2S	UDP モードでは、ezTCP は UDP を使えます。																

water_mark			
unsigned int	Read/Write	COD、U2S	
<p>・接続開始時に許容するデータ量を格納します。この値は COD、U2S モードで使用します。</p>			

remote_ip			
unsigned int	Read/Write	COD、U2S	
<p>・ターゲットの IP アドレスを格納します。この値は、COD、U2S モードで使用します。</p>			

remote_port			
unsigned int	Read/Write	COD、U2S	
<p>・ターゲットのポート番号を格納します。この値は、COD、U2S モードで使用します。</p>			

local_port			
unsigned int	Read/Write	T2S	
<p>・ローカル IP アドレスを格納します。この値は T2S モードで使用します。</p>			

poe_uid			
unsigned char [8]	Read/Write		
・ PPPoE ログイン ID を格納します。			

poe_pwd			
unsigned char [8]	Read/Write		
・ PPPoE ログインパスワードを格納します。			

3.3 wlan_env

3.3.1 概要

wlan_env 構造体は無線タイプの製品で使用可能です。使用可能な製品を次に示します。

無線タイプ	
EZL-80	EZL-80C
EZL-90	EZL-300W Lite
EZL-300L	

Table 3.3-1 wlan_env 構造体で使用可能な製品一覧

wlan_env 構造体は機能の調査用です。状態表示や値の設定には ezCfg ライブラリを用います。

```

struct wlan_env
{
    unsigned int cctype;        /* Connection Control Type */
    unsigned int channel;      /* Channel */
    unsigned int wep;          /* 0 - NO WEP, 1- 64BIT, 2- 128BIT */
    unsigned int wep_id;       /* 0, 1, 2, 3 */
    unsigned int cfg_ahc;
    unsigned int cfg_ifs;
    unsigned int auth;         /* 0 - Open System, 1 - Shared Key */
    unsigned int pad1;         // NOT USED. Never modify.
    unsigned int pad2;         // NOT USED. Never modify.
    _u8 pad3[12];              // NOT USED. Never modify.
    _u8 target_ssid[32];
    _u8 new_ssid[32];
    _u8 key64[4][6];
    _u8 key128[4][14];
};

```

List 3.3-2 wlan_env 構造体

※List 3.3-2 で言及されない値は予約されています。
 型、名前、言及されない引数の命令は変更しないでください。
 不正な使用によって生じた損害の保証、及び一切のサポートは行いません。

3.3.2 引数

表の説明

①メンバー名	②型	③入出力	④対応モード	⑤対応製品
⑥内容				

※⑤対応製品の詳細については、「1.2 製品区分」を参照してください。

cctype										
unsigned int		Read/Write								
<ul style="list-style-type: none"> ・コネクションコントロールのタイプを格納します。 										
<table border="1"> <thead> <tr> <th>cctype</th> <th>type</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>アドホック</td> </tr> <tr> <td>1</td> <td>インフラストラクチャ</td> </tr> </tbody> </table>					cctype	type	0	アドホック	1	インフラストラクチャ
cctype	type									
0	アドホック									
1	インフラストラクチャ									

channel				
unsigned int		Read/Write		
<ul style="list-style-type: none"> ・無線 LAN のチャンネルを格納します。 				

wep												
unsigned int		Read/Write										
<ul style="list-style-type: none"> ・WEP(Wired Equivalent Privacy)の設定を格納します。 												
<table border="1"> <thead> <tr> <th>wep</th> <th>type</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>no WEP</td> </tr> <tr> <td>1</td> <td>64bit</td> </tr> <tr> <td>2</td> <td>128bit</td> </tr> </tbody> </table>					wep	type	0	no WEP	1	64bit	2	128bit
wep	type											
0	no WEP											
1	64bit											
2	128bit											

wep_id														
unsigned int		Read/Write												
<ul style="list-style-type: none"> ・WEP(Wired Equivalent Pravacy)のインデックスナンバーを格納します。 														
<table border="1"> <thead> <tr> <th>wep_id</th> <th>0</th> <th>1</th> <th>2</th> <th>3</th> </tr> </thead> <tbody> <tr> <td></td> <td></td> <td></td> <td></td> <td></td> </tr> </tbody> </table>					wep_id	0	1	2	3					
wep_id	0	1	2	3										

cfg_ahc

unsigned int

Read/Write

- ・ このフラグをセットした場合、ezTCP は起動後数秒の間、初期設定のアドホックネットワークに接続します。詳しくは、「Fig 3.3-3 起動モードの選択」を参照してください。

cfg_ifs

unsigned int

Read/Write

- ・ このフラグをセットした場合、ezTCP は起動後数秒の間、初期設定のインフラストラクチャネットワークに接続します。詳しくは、「Fig 3.3-3 起動モードの選択」を参照してください。

auth

unsigned int

Read/Write

- ・ インフラストラクチャ ネットワークで用いる認証モードを格納します。

auth	mode
0	オープンシステム認証
1	共有キー認証

target_ssid

Read/Write

- ・ インフラストラクチャ ネットワークで用いる SSID を格納します。

new_ssid

Read/Write

- ・ アドホック ネットワークで用いる SSID を格納します。

key64

Read/Write

- ・ 64bit の key 値を 4 セット格納します。

key128		
	Read/Write	
<p>・ 128bit の key 値を 4 セット格納します。</p>		

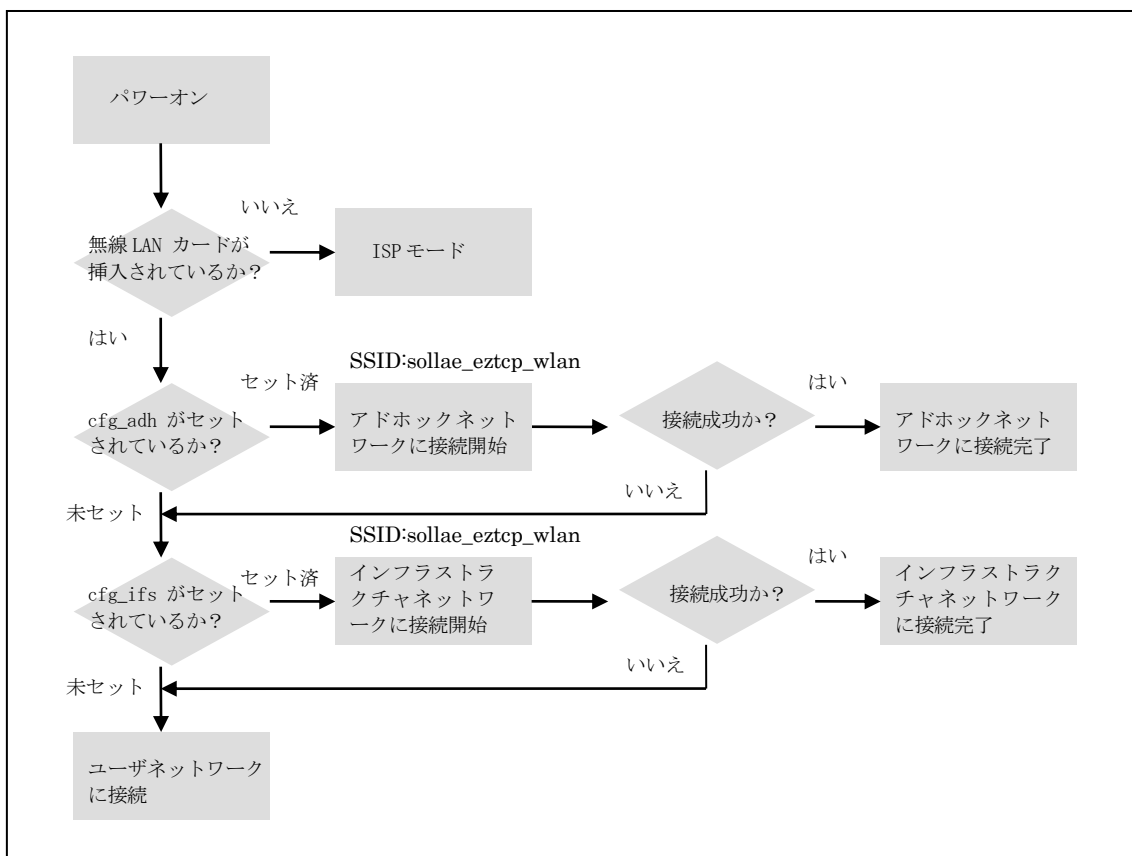


Fig 3.3-3 起動モードの選択

3.4 etc_opt

3.4.1 概要

etc_opt 構造体は、常用しないオプションに使用します。現在はコメント用オプションとして機能しています。
この関数は現在下記の製品にて対応しています。

LAN タイプ	製品名
有線 LAN	EZL-50L, EZL-200L, EZL-220, EZL-200F
無線 LAN	EZL-300L, EZL-300S EZL-80 / 80c / 90 / 300W Lite(ファーム名 : v1.3i もしくは、それ以上のバージョン)

List 3.4-1 対応製品一覧

```
struct etc_opt
{
    _u8 comment[32];
    _u8 ext_opt[32];           // NOT USED. Never modify.
    _u8 opt[20];              // NOT USED. Never modify.
};
```

List 3.4-2 etc_opt 構造体

※List 3.4-2 で言及されない値は予約されています。
型、名前、言及されない引数の命令は変更しないでください。
不正な使用によって生じた損害 の保証、及び一切のサポートは行いません。

3.4.2 引数

表の説明

①メンバー名	②型	③入出力	④対応モード	⑤対応製品
⑥内容				

※⑤対応製品の詳細については、「1.2 製品区分」を参照してください。

comment	Read/Write		
・異なる種類の ezTCP を用いる場合、それらの区別にお使いください。			

3.5 stat_env

3.5.1 概要

stat_env 構造体は、ezCfg ライブラリで使用する各関数の状態をステータス値として持っています。

```
struct stat_env
{
    _u8 ver[8];
    _u8 freq[6];
    _u8 boot_ver[8];
    _u8 mac_addr[6];
    _u8 ip_addr[4];
    _u8 sub_mask[4];
    _u8 gate_addr[4];
    unsigned int uptime_day;
    unsigned int uptime_hour;
    unsigned int uptime_minute;
    unsigned int uptime_second;
    long unsigned int sio_rx;
    long unsigned int sio_tx;
    long unsigned int eth_rx;
    long unsigned int eth_tx;
    unsigned int eth_crc;
    unsigned int eth_align;
    unsigned int eth_lost;
    char text[256];
};
```

List 3.5-1 stat_env 構造体

※List 3.5-1 で言及されない値は予約されています。

型、名前、言及されない引数の命令は変更しないでください。

不正な使用によって生じた損害 の保証、及び一切のサポートは行いません。

3.5.2 引数

表の説明

①メンバー名			
②型	③入出力	④対応モード	⑤対応製品
⑥内容			

※⑤対応製品の詳細については、「1.2 製品区分」を参照してください。

ver			
unsigned char[8]	Read	T2S、ATC、COD	
・メジャーバージョン番号を格納します。			

freq			
unsigned char[6]	Read	T2S、ATC、COD	
・ezTCPの周波数を格納します。			

boot_ver			
unsigned char[8]	Read	T2S、ATC、COD	
・ブートバージョンを格納します。 ※現在この機能を利用できるのは、EZL-50、EZL-60のみとなっております。			

mac_addr			
unsigned char[8]	Read	T2S、ATC、COD	
・MACアドレスを格納します。			

ip_addr			
unsigned char[4]	Read	T2S、ATC、COD	
・ローカルIPアドレスを格納します。			

sub_mask			
unsigned char[8]	Read	T2S、ATC、COD	EZL-50、EZL-200
<ul style="list-style-type: none"> ・サブネットマスクの値を格納します。 			

gate_addr			
unsigned char[4]	Read	T2S、ATC、COD	EZL-50、EZL-200
<ul style="list-style-type: none"> ・ゲートウェイ IP アドレスを格納します。 			

uptime_day			
unsigned int	Read	T2S、ATC、COD	EZL-50、EZL-200
<ul style="list-style-type: none"> ・ezTCP の起動からの稼働日数を格納します。この値は WriteEzTCP 関数を実行することにより初期化されます。 			

uptime_hour			
unsigned int	Read	T2S、ATC、COD	EZL-50、EZL-200
<ul style="list-style-type: none"> ・ezTCP の起動からの稼働時間（時）を格納します。この値は WriteEzTCP 関数を実行することにより初期化されます。 			

uptime_minute			
unsigned int	Read	T2S、ATC、COD	EZL-50、EZL-200
<ul style="list-style-type: none"> ・ezTCP の起動からの稼働時間（分）を格納します。この値は WriteEzTCP 関数を実行することにより初期化されます。 			

uptime_second			
unsigned int	Read	T2S、ATC、COD	EZL-50、EZL-200
<ul style="list-style-type: none"> ・ezTCP の起動からの稼働時間（秒）を格納します。この値は WriteEzTCP 関数を実行することにより初期化されます。 			

sio_rx			
long unsigned int	Read	T2S、ATC、COD	EZL-50、EZL-200
・シリアル通信で受信したバイト数を格納します。			

sio_tx			
long unsigned int	Read	T2S、ATC、COD	EZL-50、EZL-200
・シリアル通信で送信したバイト数を格納します。			

eth_rx			
long unsigned int	Read	T2S、ATC、COD	EZL-50、EZL-200
・イーサネット通信で受信したバイト数を格納します。			

eth_tx			
long unsigned int	Read	T2S、ATC、COD	EZL-50、EZL-200
・イーサネット通信で送信したバイト数を格納します。			

eth_crc			
unsigned int	Read	T2S、ATC、COD	EZL-50、EZL-200
・CRCエラーが起こったパケット数を格納します。			

eth_align			
unsigned int	Read	T2S、ATC、COD	EZL-50、EZL-200
・アライメントエラーが起こったパケット数を格納します。			

eth_lost			
unsigned int	Read	T2S、ATC、COD	EZL-50、EZL-200
・消失したパケットの数を格納します。			

text			
char	Read	T2S、ATC、COD	EZL-50、EZL-200
・予約された値です。			

4. eztcplib.lib

4.1 ProbeEzTCP

4.1.1 概要

ProbeEzTCP 関数はローカルネットワーク上の ezTCP を検索します。各 ezTCP の識別は MAC アドレスで行います。

4.1.2 プロトタイプ

```
int ProbeEzTCP(struct spe_env * eztcpenv,
              struct wlan_env * eztcpwenv,
              struct etc_opt * eztcpopt,
              int * nResultCount,
              int * nErrNum);
```

List 4.1-1 ProbeEzTCP 関数のプロトタイプ宣言

4.1.3 引数

型	名前	入出力	説明
struct spe_env *	eztcpenv	Read	ezTCP の各種設定が格納されている spe_env 構造体へのポインタです。
struct wlan_env *	eztcpwenv	Read	無線タイプの ezTCP の各種設定が格納されている wlan_env 構造体へのポインタです。
struct etc_opt *	eztcpopt	Read	各種設定が格納されている etc_opt 構造体へのポインタです。
int *	nResultCount	Read	検出された ezTCP の数が格納されている整数型へのポインタです。
int *	nErrNum	Read	エラーが発生した場合、エラー番号を格納する整数型へのポインタです。

Table 4.1-2 ProbeEzTCP 関数の引数

4.1.4 戻り値

型	値	説明
int	1	正常終了。
	-1	エラーが発生。戻り値は、EZTCP_ERR(-1)。エラー番号が eErrNum へ格納される。

Table 4.1-3 ProbeEzTCP 関数の戻り値

4.1.5 使用上の注意点

- 本関数を実行する前に、eztcpenv, eztcpwenv, eztcpopt 構造体のために十分なメモリ領域を確保する必要があります。

```
eg. env_base = (struct spe_env *)malloc(sizeof(struct spe_env) * 256);  
    wenv_base = (struct wlan_env *)malloc(sizeof(struct wlan_env) * 256);  
    etc_base = (struct etc_opt *)malloc(sizeof(struct etc_opt) * 256);
```

上記の例では env_base, wenv_base, etc_base はそれぞれ ezTCP の情報を 256 台分取得することができます。つまりこの場合、ローカルネットワーク上に存在する ezTCP を最大 256 台検索を行うことができます。

- ProbeEzTCP 関数を使用するには eztcplib.h ヘッダーファイルを指定し、ライブラリに ws2_32.lib と eztcplib.lib を加える必要があります。
- ProbeEzTCP 関数の実行には少なくとも数秒かかります。また、ローカルネットワーク上に多くの ezTCP が存在する場合、関数の実行に多くの時間かかる場合があります。

4.2 WriteEzTCP

4.2.1 概要

WriteEzTCP 関数はローカルネットワーク上の ezTCP の設定を行います。ezTCP の指定は、eztcpenv 引数で行います。

4.2.2 プロトタイプ

```
int WriteEzTCP(struct spe_env * eztcpenv,
              struct wlan_env * eztcpwenv,
              struct etc_opt * eztcpopt,
              int *nErrNum);
```

List 4.2-1 WriteEzTCP 関数のプロトタイプ宣言

4.2.3 引数

型	名前	入出力	説明
struct spe_env	eztcpenv	Read	変更する設定が含まれている spe_env 構造体へのポインタです。
struct wlan_env *	eztcpwenv	Read	無線タイプの変更する設定が含まれている spe_env 構造体へのポインタです。
struct etc_opt *	eztcpopt	Read	変更する設定が格納されている etc_opt 構造体へのポインタです。
int *	nErrNum	Read	エラーが発生した場合、エラー番号を格納する整数型へのポインタです。

Table 4.2-2 WriteEzTCP 関数の引数

4.2.4 戻り値

型	値	説明
int	1	正常終了。
	-1	エラーが発生。戻り値は、EZTCP_ERR(-1)。エラー番号が eErrNum へ格納される。

Table 4.2-3 WriteEzTCP 関数の戻り値

4.2.5 使用上の注意

- WriteEzTCP 関数を使用する前に、ProbeEzTCP 関数で ezTCP を検索し正しい MAC アドレスを取得しておく必要があります。
- WriteEzTCP 関数を使用するには eztcplib.h ヘッダーファイルを指定し、ライブラリに ws2_32.lib と eztcplib.lib を加える必要があります。
- WriteEzTCP 関数の実行には少なくとも数秒かかります。また、ローカルネットワーク上に多くの ezTCP が存在する場合、関数の実行に多くの時間かかる場合があります。

4.3 StatusEzTCP

4.3.1 概要

StatusEzTCP 関数は、eztcpenv 引数の中で定義されている MAC アドレスの ezTCP の設定情報を取得します。

4.3.2 プロトタイプ

```
int StatusEzTCP(struct spe_env * eztcpenv,
               struct stat_env * eztcpstat,
               int *nErrNum);
```

List 4.3-1 StatusEzTCP 関数のプロトタイプ宣言

4.3.3 引数

型	名前	入出力	説明
struct spe_env	eztcpenv	Write	ターゲットの MAC アドレスが格納されている spe_env 構造体へのポインタです。
struct stat_env *	eztcpstat	Read	ezTCP の状態が格納されている stat_env 構造体へのポインタです。
int *	nErrNum	Read	エラーが発生した場合、エラー番号を格納する整数型へのポインタです。

Table 4.3-2 StatusEzTCP 関数の引数

4.3.4 戻り値

型	値	説明
int	1	正常終了。
	-1	エラーが発生。戻り値は、EZTCP_ERR (-1)。エラー番号が eErrNum へ格納される。

Table 4.3-3 StatusEzTCP 関数の戻り値

4.3.5 使用上の注意

- StatusEzTCP 関数を使用する前に、ProbeEzTCP 関数で ezTCP を検索し正しい MAC アドレスを取得しておく必要があります。
- StatusEzTCP 関数を使用するには eztcplib.h ヘッダーファイルを指定し、ライブラリに ws2_32.lib と eztcplib.lib を加える必要があります。
- StatusEzTCP 関数の実行には少なくとも数秒かかります。また、ローカルネットワーク上に多くの ezTCP が存在する場合、関数の実行に多くの時間がかかる場合があります。

4.4 ChangePwdEzTCP

4.4.1 概要

ChangePwdEzTCP 関数は、ezTCP パスワードの変更または削除をします。

4.4.2 プロトタイプ

```
int ChangePwdEzTCP(struct spe_env * eztcpenv,
                  const char * szChangePwd,
                  int *nErrNum);
```

List 4.4-1 ChangePwdEzTCP 関数のプロトタイプ宣言

4.4.3 引数

型	名前	入出力	説明
struct spe_env *	eztcpenv	Write	ターゲットの MAC アドレスが格納されている spe_env 構造体へのポインタです。
const char *	szChangePwd	Write	新しいパスワード文字列へのポインタです。
int*	nErrNum	Read	エラーが発生した場合、エラー番号を格納する整数型へのポインタです。

Table 4.4-2 ChangePwdEzTCP 関数の引数

4.4.4 戻り値

型	値	説明
int	1	正常終了。
	-1	エラーが発生。戻り値は、EZTCP_ERR(-1)。エラー番号が eErrNum へ格納される。

Table 4.4-3 ChangePwdEzTCP 関数の戻り値

4.4.5 使用上の注意

- ChangePwdEzTCP 関数を使用する前に、ProbeEzTCP 関数で ezTCP を検索し正しい MAC アドレスを取得しておく必要があります。
- ChangePwdEzTCP 関数を使用するには eztcplib.h ヘッダーファイルを指定し、ライブラリに ws2_32.lib と eztcplib.lib を加える必要があります。
- ChangePwdEzTCP 関数の実行には少なくとも数秒かかります。また、ローカルネットワーク上に多くの ezTCP が存在する場合、関数の実行に多くの時間がかかる場合があります。

```
eg.

#define CURRENT_PWD "current_password"
#define CHANGE_PWD "new_password"

// eg. Change or erase password

memset(env_base->passwd, 0, 12);
memcpy(env_base->passwd, CURRENT_PWD, sizeof(CURRENT_PWD));
memset(szPWD, 0, 12);

// Remark this line to erase password.
memcpy(szPWD, CHANGE_PWD, sizeof(CHANGE_PWD));
nResult = ChangePwdEzTCP((env_base), szPWD, &nErr);
if (nResult == EZTCP_ERR)
{
    if (nErr == EZTCP_ERR_PWD)
        MessageBox(hWnd, "The password is mismatch!",
                    "Change password fail!", MB_OK);
    if (nErr == EZTCP_ERR_RES)
        MessageBox(hWnd, "There is no response of ezTCP!",
                    "Change password fail!", MB_OK);
}
else
{
    MessageBox(hWnd, "The password is changed!",
                "ezTCP notice!", MB_OK);
}
```

List 4.4-4 ChangePwdEzTCP 関数のサンプルプログラム

4.5 RemoteReadEzTCP

4.5.1 概要

RemoteReadEzTCP 関数は、ローカルまたはリモートネットワーク上の ezTCP を検索します。
各 ezTCP の識別は IP アドレスで行います。
この関数は現在下記の製品にて対応しています。

LAN タイプ	製品名
有線 LAN	EZL-50L, EZL-200L, EZL-220, EZL-200F
無線 LAN	EZL-300L, EZL-300S EZL-80 / 80c / 90 / 300W Lite(ファーム名 : v1.3i もしくは、それ以上のバージョン)

List 4.5-1 RemoteReadEzTCP 関数のプロトタイプ宣言

4.5.2 プロトタイプ

```
int RemoteReadEzTCP(DWORD ip,
    struct spe_env * eztcpenv,
    struct wlan_env * eztcpwenv,
    struct etc_opt * eztcpopt,
    int * nResultCount,
    int * nErrNum);
```

List 4.5-2 RemoteReadEzTCP 関数のプロトタイプ宣言

4.5.3 引数

型	名前	入出力	説明
DWORD	ip	Write	設定を読み出す ezTCP を、ローカル IP アドレスで指定します。 この引数はリトルエンディアンで値を渡してください。 (関数内部でこの値を用いる場合、ビッグエンディアンに変更されま す。)
struct spe_env*	eztcpenv	Read	ezTCP の各種設定が格納されている spe_env 構造体へのポインタです。
struct wlan_env	eztcpwenv	Read	無線タイプの ezTCP の各種設定が格納されている wlan_env 構造体への ポインタです。
struct etc_opt*	eztcpopt	Read	各種設定が格納されている etc_opt 構造体へのポインタです。
int *	nResultCount	Read	検出された ezTCP の数が格納されている整数型へのポインタです。
int *	nErrNum	Read	エラーが発生した場合、エラー番号を格納する整数型へのポインタで す。

Table 4.5-3 RemoteReadEzTCP 関数の引数

4.5.4 戻り値

型	値	説明
int	1	正常終了。
	-1	エラーが発生。戻り値は、EZTCP_ERR(-1)。エラー番号が eErrNum へ格納される。

Table 4.5-4 RemoteReadEzTCP 関数の戻り値

4.5.5 使用上の注意

- 本関数を実行する前に、eztcpenv, eztcpwenv, eztcpopt 構造体のために十分なメモリ領域を確保する必要があります。

```
eg. env_base = (struct spe_env *)malloc(sizeof(struct spe_env) * 256);
    wenv_base = (struct wlan_env *)malloc(sizeof(struct wlan_env) * 256);
    etc_base = (struct etc_opt *)malloc(sizeof(struct etc_opt) * 256);
```

上記の例では env_base, wenv_base, etc_base はそれぞれ ezTCP の情報を 256 台分取得することができます。つまりこの場合、ローカルネットワーク上に存在する ezTCP を最大 256 台検索を行うことができます。

- ProbeEzTCP 関数を使用するには eztcplib.h ヘッダーファイルを指定し、ライブラリに ws2_32.lib と eztcplib.lib を加える必要があります。
- ProbeEzTCP 関数の実行には少なくとも数秒かかります。また、ローカルネットワーク上に多くの ezTCP が存在する場合、関数の実行に多くの時間かかる場合があります。

4.6 RemoteWriteEzTCP

4.6.1 概要

RemoteWriteEzTCP 関数は、ローカルまたはリモートネットワーク上の ezTCP の設定を変更します。
ezTCP の指定は MAC アドレスで行います。

4.6.2 プロトタイプ

```
int RemoteWriteEzTCP(DWORD ip,
                    struct spe_env * eztcpenv,
                    struct wlan_env * eztcpwenv,
                    struct etc_opt * eztcpopt,
                    int *nErrNum);
```

List 4.6-1 RemoteWriteEzTCP 関数のプロトタイプ宣言

4.6.3 引数

型	名前	入出力	説明
DWORD	ip	Write	設定を書き込む ezTCP を、ローカル IP アドレスで指定します。 この引数はリトルエンディアンで値を渡してください。 (関数内部でこの値を用いる場合、ビッグエンディアンに変更 されます。)
struct spe_env*	eztcpenv	Write	変更する設定が含まれている spe_env 構造体へのポインタで す。
struct wlan_env*	eztcpwenv	Write	無線タイプの変更する設定が含まれている spe_env 構造体へ のポインタです。
struct etc_opt*	eztcpopt	Read	変更する設定が格納されている etc_opt 構造体へのポインタ です。
int *	nErrNum	Read	エラーが発生した場合、エラー番号を格納する整数型へのポイ ンタです。

Table 4.6-2 RemoteWriteEzTCP 関数の引数

4.6.4 戻り値

型	値	説明
int	1	正常終了。
	-1	エラーが発生。戻り値は、EZTCP_ERR(-1)。エラー番号が eErrNum へ格納される。

Table 4.6-4 RemoteWriteEzTCP 関数の戻り値

nErrNum について

nErrNum	説明
EZTCP_ERR_PWD	対象のezTCPにパスワードを設定している場合は、spe_env構造体のpassword引数にもパスワードを設定してください。
EZTCP_ERR_RES	環境値を書き込んでいる最中にエラーが起こった場合、EZTCP_ERR_RES が int nErrNum に格納されます。

Table 4.6-5 nErrNum について

4.6.5 使用上の注意

- 確実なデータ通信のため、必ず RemoteWriteEzTCP 関数を用いる前に、RemoteReadEzTCP 関数を呼び出すようにしてください。
- RemoteWriteEzTCP 関数を使用するには eztcplib.h ヘッダーファイルを指定し、ライブラリに ws2_32.lib と eztcplib.lib を加える必要があります。
- RemoteWriteEzTCP 関数の実行には少なくとも数秒かかります。また、ローカルネットワーク上に多くの ezTCP が存在する場合、関数の実行に多くの時間がかかる場合があります。

4.7 RemoteStatusEzTCP

4.7.1 概要

RemoteStatusEzTCP 関数は、ローカルまたはリモートネットワーク上の ezTCP の状態値を読み出します。
各 ezTCP の識別は IP アドレスで行います。
この関数は現在下記の製品にて対応しています。

LAN タイプ	製品名
有線 LAN	EZL-50L, EZL-200L, EZL-220, EZL-200F
無線 LAN	EZL-300L, EZL-300S EZL-80 / 80c / 90 / 300W Lite(ファーム名 : v1.3i もしくは、それ以上のバージョン)

List 4.7-1 RemoteReadEzTCP 関数のプロトタイプ宣言

4.7.2 プロトタイプ

```
int StatusEzTCP (DWORD ip,
                struct spe_env * eztcpenv,
                struct stat_env * eztcpstat,
                int *nErrNum);
```

List 4.7-2 RemoteStatusEzTCP 関数のプロトタイプ宣言

4.7.3 引数

型	名前	入出力	説明
DWORD	ip	Write	設定を読み出す ezTCP を、ローカル IP アドレスで指定します。 この引数はリトルエンディアンで値を渡してください。 (関数内部でこの値を用いる場合、ビッグエンディアンに変更されます。)
struct spe_env*	eztcpenv	Write	ターゲットの ezTCP の各種設定を格納する spe_env 構造体へのポインタです。
struct stat_env*	eztcpstat	Read	ezTCP の状態が格納する spe_env 構造体へのポインタです。
int *	nErrNum	Read	エラーが発生した場合、エラー番号を格納する整数型へのポインタです。

Table 4.7-3 RemoteStatusEzTCP 関数の引数

4.7.4 戻り値

型	値	説明
int	1	正常終了。
	-1	エラーが発生。戻り値は、EZTCP_ERR(-1)。エラー番号が eErrNum へ格納される。

Table 4.7-4 RemoteStatusEzTCP 関数の戻り値

4.7.5 使用上の注意

- 確実なデータ通信のため、必ず RemoteStatusEzTCP 関数を用いる前に、RemoteReadEzTCP 関数を呼び出すようにしてください。
- RemoteStatusEzTCP 関数を使用するには eztcp.h ヘッダーファイルを指定し、ライブラリに ws2_32.lib と eztcp.lib を加える必要があります。
- RemoteStatusEzTCP 関数の実行には少なくとも数秒かかります。また、ローカルネットワーク上に多くの ezTCP が存在する場合、関数の実行に多くの時間がかかる場合があります。

4.8 RemoteChangePwdEzTCP

4.8.1 概要

RemoteChangePwdEzTCP 関数は、ローカルまたはリモートネットワーク上の ezTCP のパスワードを変更、消去します。ezTCP の指定は IP アドレスで行います。この関数は現在下記の製品にて対応しています。

LAN タイプ	製品名
有線 LAN	EZL-50L, EZL-200L, EZL-220, EZL-200F
無線 LAN	EZL-300L, EZL-300S EZL-80 / 80c / 90 / 300W Lite(ファーム名 : v1.3i もしくは、それ以上のバージョン)

List 4.8-1 RemoteReadEzTCP 関数のプロトタイプ宣言

4.8.2 プロトタイプ

```
int RemoteChangePwdEzTCP(DWORD ip,
                          struct spe_env * eztcpenv,
                          const char * szChangePwd,
                          int *nErrNum);
```

List 4.8-2 RemoteChangeEzTCP 関数のプロトタイプ宣言

4.8.3 引数

型	名前	入出力	説明
DWORD	ip	Write	パスワードを変更する ezTCP を、ローカル IP アドレスで指定します。この引数はリトルエンディアンで値を渡してください。(関数内部でこの値を用いる場合、ビッグエンディアンに変更されます。)
struct spe_env *	eztcpenv	Write	変更する設定が含まれている spe_env 構造体へのポインタです。
const char *	szChangePwd	Write	新しいパスワード文字列へのポインタです。
int *	nErrNum	Read	エラーが発生した場合、エラー番号を格納する整数型へのポインタです。

Table 4.8-3 RemoteChangeEzTCP 関数の引数

4.8.4 戻り値

型	値	説明
int	1	正常終了。
	-1	エラーが発生。戻り値は、EZTCP_ERR(-1)。エラー番号が eErrNum へ格納される。

Table 4.8-4 RemoteChangeEzTCP 関数の戻り値

4.8.5 使用上の注意

- ・ 確実なデータ通信のため、必ず RemoteChangePwdEzTCP 関数を用いる前に、RemoteReadEzTCP 関数を呼び出すようにしてください。
- ・ RemoteChangePwdEzTCP 関数を使用するには ezteplib.h ヘッダーファイルを指定し、ライブラリに ws2_32.lib と ezteplib.lib を加える必要があります。
- ・ RemoteChangePwdEzTCP 関数の実行には少なくとも数秒かかります。また、ローカルネットワーク上に多くの ezTCP が存在する場合、関数の実行に多くの時間がかかる場合があります。

```
eg.  
  
#define CURRENT_PWD "current_password"  
#define CHANGE_PWD "new_password"  
  
// eg. Change or erase password  
  
DWORD ip;  
  
memset(env_base->passwd, 0, 12);  
memcpy(env_base->passwd, CURRENT_PWD, sizeof(CURRENT_PWD));  
memset(szPWD, 0, 12);  
  
// Remark this line to erase password.  
memcpy(szPWD, CHANGE_PWD, sizeof(CHANGE_PWD));  
  
IpAddressControl.GetAddress(ip);  
  
nResult = RemoteChangePwdEzTCP(ip, (env_base), szPWD, &nErr);  
if (nResult == EZTCP_ERR)  
{  
    if (nErr == EZTCP_ERR_PWD)  
        MessageBox(hWnd, "The password is mismatch!",  
                    "Change password fail!", MB_OK);  
  
    if (nErr == EZTCP_ERR_RES)  
        MessageBox(hWnd, "There is no response of ezTCP!",  
                    "Change password fail!", MB_OK);  
}  
else  
{  
    MessageBox(hWnd, "The password is changed!",  
                "ezTCP notice!", MB_OK);  
}
```

List 4.8-5 RemoteChangeEzTCP 関数のサンプルプログラム

4.9 GetLibVerEzTCP

4.9.1 概要

GetLibVerEzTCP 関数は、ライブラリのバージョン情報を取得します。
バージョン情報用の文字列の最大数は、64 キャラクタになります。

4.9.2 プロトタイプ

```
int GetLibVerEzTCP( char * szVersion, int len );
```

List 4.9-1 GetLibVerEzTCP 関数のプロトタイプ宣言

4.9.3 引数

型	名前	入出力	説明
char *	szVersion	Read	バージョン情報文字列へのポインタです。
int	len	Write	szVersion ポインタのサイズを格納します。

Table 4.9-2 GetLibVerEzTCP 関数の引数

4.9.4 戻り値

型	値	説明
int	1	正常終了。
	-1	エラーが発生。戻り値は、EZTCP_ERR(-1)。エラー番号が eErrNum へ格納される。

Table 4.9-3 GetLibVerEzTCP 関数の戻り値

4.9.5 使用上の注意

バージョン情報文字列のため、最低でも 70byte の空間を確保してください。

5. その他

改定履歴

版数	日付	改定内容
1 版	2002/09/13	新規
2 版	2006/08/24	バージョンアップに伴い、無線 LAN 及びリモートネットワーク上で使用する関数、及び構造体を追加
3 版	2008/02/15	英文マニュアルの更新に伴い、内容を追記。(2~4 章) 書式を統一する(全章)
4 版	2023/10/02	住所を更新

参考文献

SollaeSystem 社 「ezcflib_manual」

本文書で取り上げているソフトウェア「EzConfig Library」について

- ・ソフトウェアの著作権は SollaeSystems 社が保有しています。

本文書について

- ・本文書の著作権は株式会社アルファプロジェクトが保有します。
- ・本文書の内容を無断で転載することは一切禁止します。
- ・本文書の内容は、将来予告なしに変更されることがあります。
- ・本文書の内容については、万全を期して作成いたしましたが、万一ご不審な点、誤りなどお気付きの点がありましたら弊社までご連絡下さい。
- ・本文書の内容に基づき、アプリケーションを運用した結果、万一損害が発生しても、弊社では一切責任を負いませんのでご了承下さい。

商標について

- ・本書で使用されている会社名、製品名は、各社の登録商標または商標または商品名称です。



株式会社アルファプロジェクト

〒431-3114
静岡県浜松市中央区積志町8 3 4
<https://www.apnet.co.jp>
E-Mail : query@apnet.co.jp
