
KEIm SoM ETHERNET サンプル アプリケーションマニュアル

Ver.1.0.0



株式会社近藤電子工業



はじめに

この度は、KEIm 製品をお買い上げいただき誠にありがとうございます。

本製品をご使用になる前に、本マニュアル及び関連資料を十分ご確認ください、使用上の注意を守って正しくご使用ください。



取扱い上の注意

- 本書に記載されている内容は、将来予告なく変更されることがあります。本製品のご使用にあたっては、弊社窓口又は弊社ホームページなどで最新の情報をご確認ください。
- 本製品には一般電子機器用部品が使用されています。極めて高い信頼性を要求する装置（航空、宇宙機器、原子力制御機器、生命維持のための医療機器等）には使用しないでください。
- 本製品は国内使用を前提として開発及び製造を行っています。本製品又は本製品を組み込んだ製品を輸出される場合は、お客様の責任において「外国為替及び外国貿易法」及びその他輸出関連法令等を順守し、必要な手続きを行ってください。
- LAN、USB 以外のコネクタへのケーブルの抜き差しは、必ず電源を OFF にした状態で行ってください。
- 水、湿気、ほこり、油煙等の多い場所では使用しないでください。
- 本製品の関連資料の全部又は一部を弊社に無断で使用または複製することを禁止します。
- 本書及び関連資料で取り上げる会社名及び製品名等は、各メーカーの商標または登録商標です。

お問い合わせ先

- 製品に関するお問い合わせは、下記のメールアドレスよりお願いいたします。

keim-support@kd-group.co.jp

目次

1. 概要.....	5
1.1. 使用部品の詳細.....	6
2. 操作方法	6
3. ファイル構成	9
4. フローチャート	10
5. 各定義値、関数の説明.....	12
5.1. 定義式	12
5.2. 構造体	12
5.3. 変数	12
5.4. 関数.....	13
5.4.1. Global 関数.....	13
5.4.1.1. main 関数.....	13
5.4.1.2. EtherApl_Init 関数.....	13
5.4.1.3. EtherApl_Set_NetConfig 関数.....	13
5.4.1.4. EtherApl_Exec 関数	13
5.4.2. static 関数.....	14
5.4.2.1. ether_apl_rcv_tcp 関数	14
5.4.2.2. ether_apl_rcv_udp 関数	14
5.4.2.3. ether_apl_analyze_rcv_data 関数.....	14
5.4.2.4. ether_cmd_initialize 関数	15
5.4.2.5. ether_cmd_rtc_write 関数	15
5.4.2.6. ether_cmd_rtc_read 関数.....	15
5.4.2.7. ether_cmd_lcd_light 関数.....	16
5.4.2.8. ether_cmd_led_write 関数.....	16
5.4.2.9. ether_cmd_led_read 関数.....	16
5.4.2.10. ether_cmd_servo_ctl 関数.....	17
5.4.2.11. ether_cmd_step_ctl 関数.....	17
5.4.2.12. ether_cmd_auto_mode 関数.....	17
6. 更新履歴	18

図表目次

表 1.1 コマンド受信による各種制御	5
表 1.2 コマンドフォーマット	5
表 1.3 使用部品詳細	6
表 3.1 ファイル構成	9
表 5.1 定義式	12
表 5.2 構造体	12
表 5.3 変数	12
表 5.4 main 関数	13
表 5.5 EtherApl_Init 関数	13
表 5.6 EtherApl_Set_NetConfig 関数	13
表 5.7 EtherApl_Exec 関数	13
表 5.8 ether_apl_recv_tcp 関数	14
表 5.9 ether_apl_recv_udp 関数	14
表 5.10 ether_apl_analyze_recv_data 関数	14
表 5.11 ether_cmd_initialize 関数	15
表 5.12 ether_cmd_rtc_write 関数	15
表 5.13 ether_cmd_rtc_read 関数	15
表 5.14 ether_cmd_lcd_light 関数	16
表 5.15 ether_cmd_led_write 関数	16
表 5.16 ether_cmd_led_read 関数	16
表 5.17 ether_cmd_servo_ctl 関数	17
表 5.18 ether_cmd_step_ctl 関数	17
表 5.19 ether_cmd_auto_mode 関数	17
図 4.1 メイン フローチャート 1/2	10
図 4.2 メイン フローチャート 2/2	11

1. 概要

本サンプルアプリでは以下の制御を行っています。

Ethernet 経由で受信した以下のコマンドにて各種制御を実施します。

表 1.1 コマンド受信による各種制御

コマンド	データ	名称	説明
IT	0000	初期化	初期化、サーボをセンター位置、LED を全消灯の状態での停止
RW	XXYY	RTC ライト	XX で指定したカテゴリに対する値 YY を RTC に書き込み 00:Sec、01:Min、02:Hour、03:Day、04:Week、05:Month、06:Year
RR	XX00	RTC リード	XX で指定したカテゴリに対する値を RTC から読み込み 00:Sec、01:Min、02:Hour、03:Day、04:Week、05:Month、06:Year
BC	00YY	LCD 調光	非対応
LW	00YY	LED ライト	YY で指定された値を LED に設定
LR	0000	LED リード	LED の現在の状態を取得
SV	00YY	サーボ制御	YY で指定された値(0～180 度)にサーボを稼働
SP	00YY	ステッピングモータ制御	YY で指定された値(0～255)だけステッピングモータにパルス出力
AU	0099	自動	動作モードを指定

*:データ部の YY は 16 進数、99 は 10 進数となります。

フローチャートについては以下をご参照ください。

Ethernet 経由で送受信するコマンドのフォーマットを表 1.2 に示します。

表 1.2 コマンドフォーマット

名称	ヘッダ	コマンド	データ	デリミタ
送信	:	C	C D D D D	CR LF
レスポンス(Data 時)	:	C	C D D - -	CR LF
レスポンス(OK 時)	:	C	C O K - -	CR LF
レスポンス(NG 時)	:	C	C N G - -	CR LF

1.1. 使用部品の詳細

Ethernet サンプルアプリは以下の部品を使用し作成しております。

本商品に付属されておりませんので、別途ご用意ください。

表 1.3 使用部品詳細

品名	品番	メーカー
スタンダードサーボモータ	#900-00005	Parallax Inc.
http://akizukidenshi.com/catalog/g/gM-06837/		
ステッピングモータ	ST-42BYG0506H	MERCURY MOTOR.
http://akizukidenshi.com/catalog/g/gP-05374/		

2. 操作方法

①PC 側の LAN の設定

IP アドレス 192.168.0.10

サブネットマスク 255.255.255.0

デフォルトゲートウェイ 192.168.1.1

②KEIm SoM 開発キットの CN7 と PC を LAN ケーブルで接続します。

③KEIm SoM 開発キットの CN8 と PC を USB ケーブルで接続します。

④電源を ON し CN7 の LED が点滅していることを確認してください。

⑤TeraTerm(例)で USB 接続している COM を接続します。

⑥本アプリケーションプログラムを起動します。

TeraTerm に以下のメッセージが表示されます。

===== W5500 NET CONF : Static =====

MAC : 00:08:DC:AB:CD:EF

IP : 192.168.0.231

GW : 192.168.1.1

SN : 255.255.255.0

=====

TCP :LBTStart

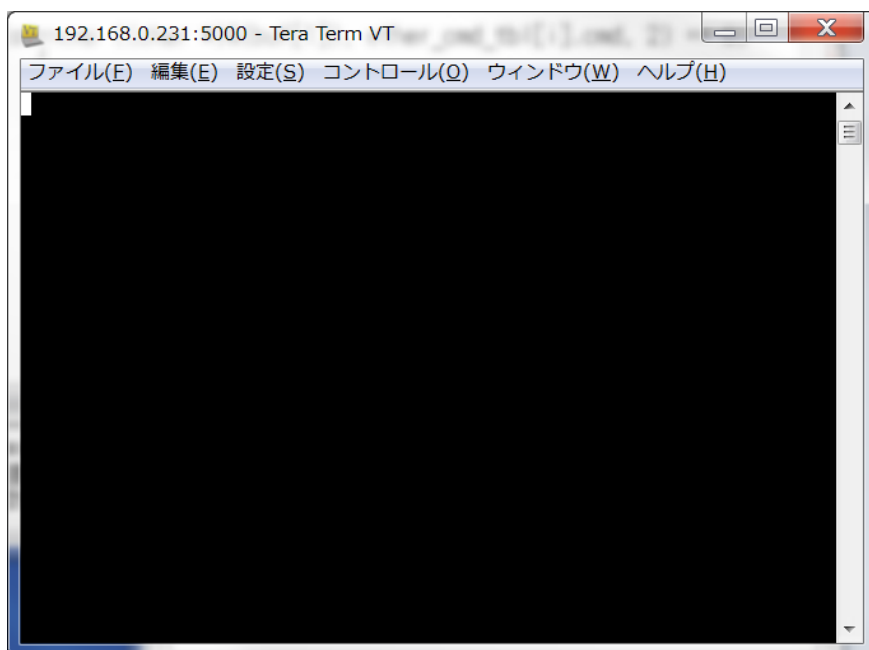
TCP :Opened

UDP :LBUStart

UDP :Opened, port [3000]

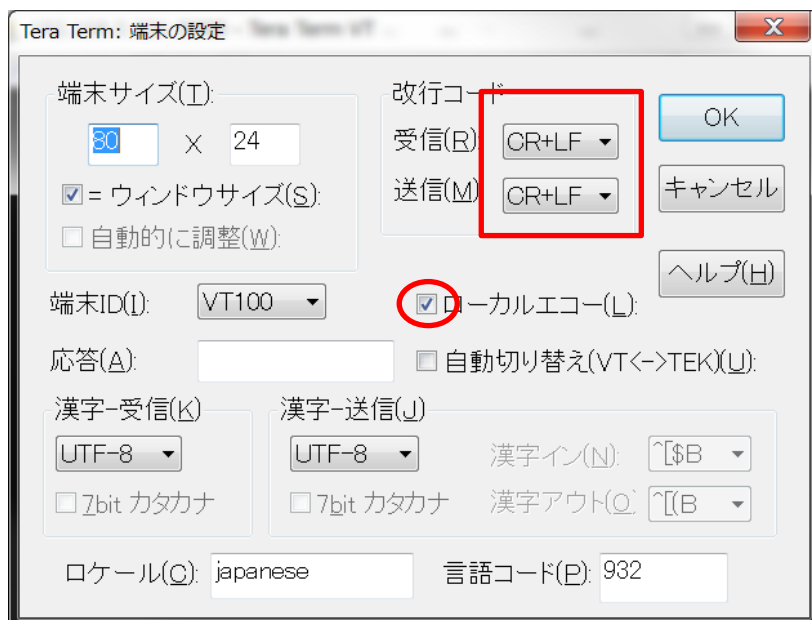
TCP :Listen, port [5000]

- ⑦もう一つ TeraTerm(例)を起動し、ファイル->新しい接続で TCP/IP をチェックし、
ホストを 192.168.0.231 に設定してください。
サービスはその他をチェック、TCP ポートは 5000 を設定、プロトコルは IPv4 を設定し OK をクリックします。



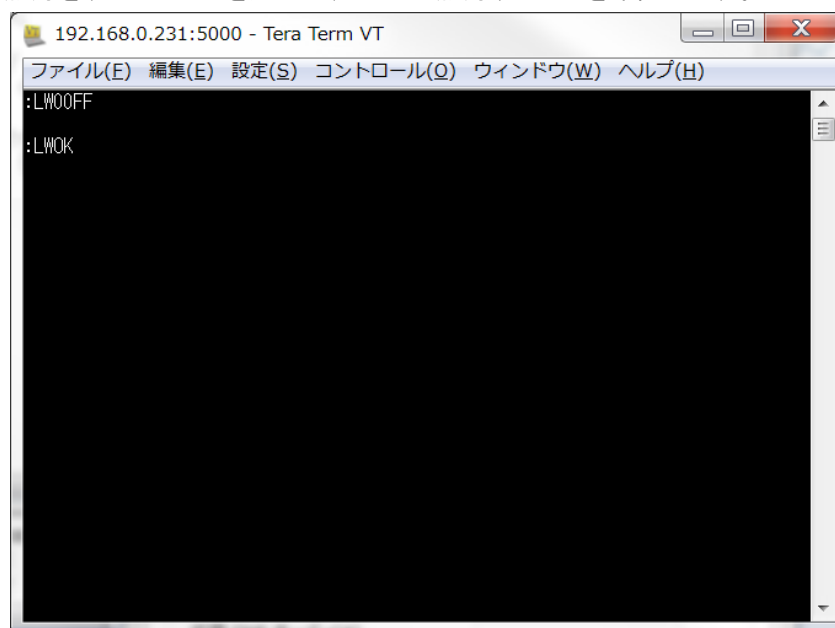
接続完了画面

- ⑧TeraTerm の設定->端末をクリックし改行コードとローカルエコーの設定を行います。
改行コード 受信/送信 CR+LF
ローカルエコーにチェックをいれます。



端末の設定画面

⑨例として LED の点灯をするコマンドを入力し、LED が点灯することを確認します。



3. ファイル構成

表 3.1 ファイル構成

ファイル名	内容
main.c	メイン関数
AdcDemoApl.h	ADC デモ関数定義
AdcDemoApl.c	ADC デモ関数
AdcSampleApl.h	ADC サンプル関数定義
AdcSampleApl.c	ADC サンプル関数
BuzzerDemoApl.h	Buzzer デモ関数定義
BuzzerDemoApl.c	Buzzer デモ関数
BuzzerSampleApl.h	Buzzer サンプル関数定義
BuzzerSampleApl.c	Buzzer サンプル関数
LedDemoApl.h	Led デモ関数定義
LedDemoApl.c	Led デモ関数
LedSampleApl.h	Led サンプル関数定義
LedSampleApl.c	Led サンプル関数
RtcDemoApl.h	Rtc デモ関数定義
RtcDemoApl.c	Rtc デモ関数
RtcSampleApl.h	Rtc サンプル関数定義
RtcSampleApl.c	Rtc サンプル関数
ServoDemoApl.h	ServoMotor デモ関数定義
ServoDemoApl.c	ServoMotor デモ関数
ServoSampleApl.h	ServoMotor サンプル関数定義
ServoSampleApl.c	ServoMotor サンプル関数
SteppingDemoApl.h	SteppingMotor デモ関数定義
SteppingDemoApl.c	SteppingMotor デモ関数
SteppingSampleApl.h	SteppingMotor サンプル関数定義
SteppingSampleApl.c	SteppingMotor サンプル関数
EtherSampleApl.h	Ethernet サンプル関数定義
EtherSampleApl.c	Ethernet サンプル関数
w5500_spi.h	w5500 初期化関数定義
w5500_spi.c	w5500 初期化関数
ioLibrary	w5500 関数ライブラリ

4. フローチャート

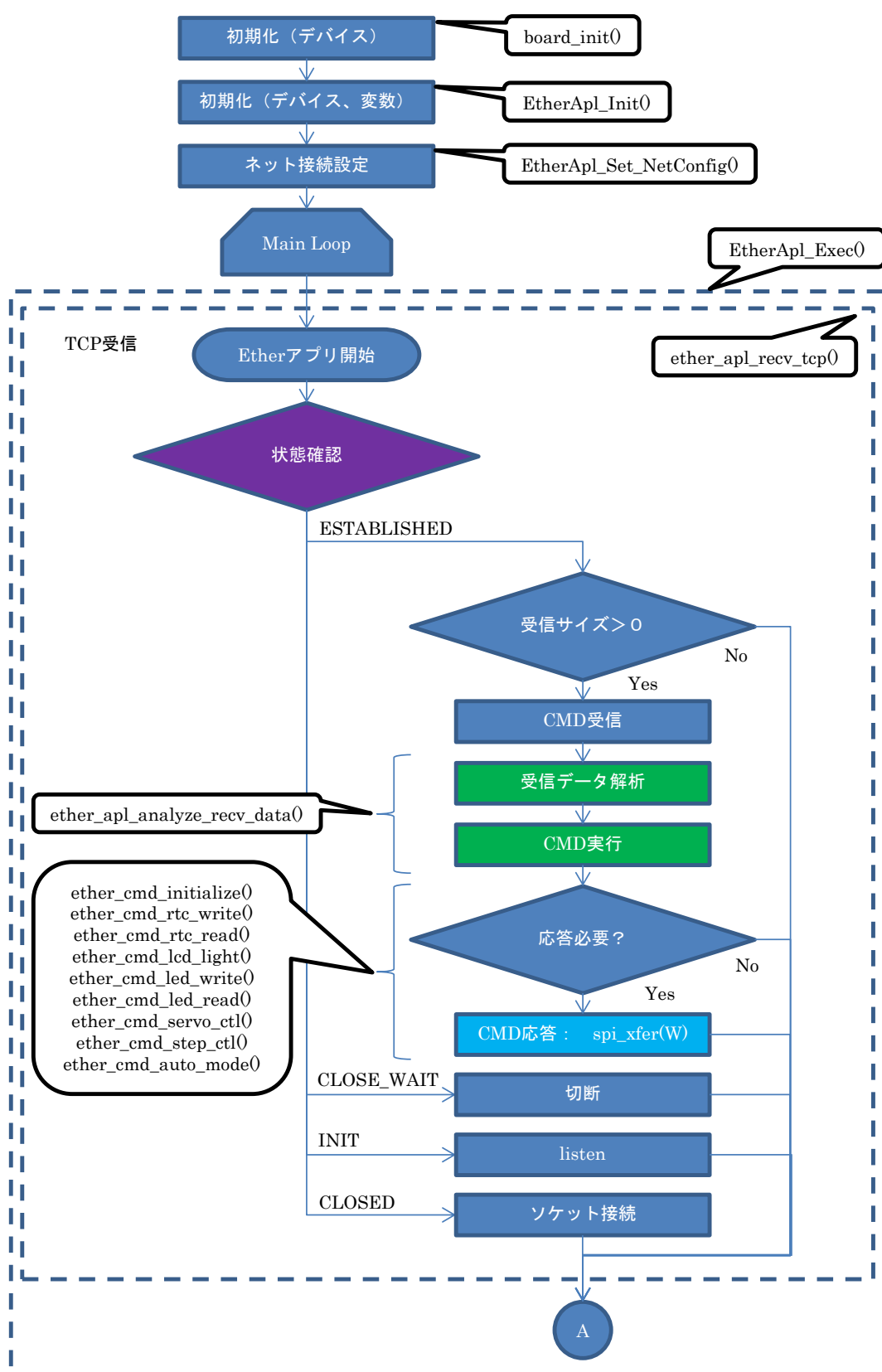


図 4.1 メイン フローチャート 1/2

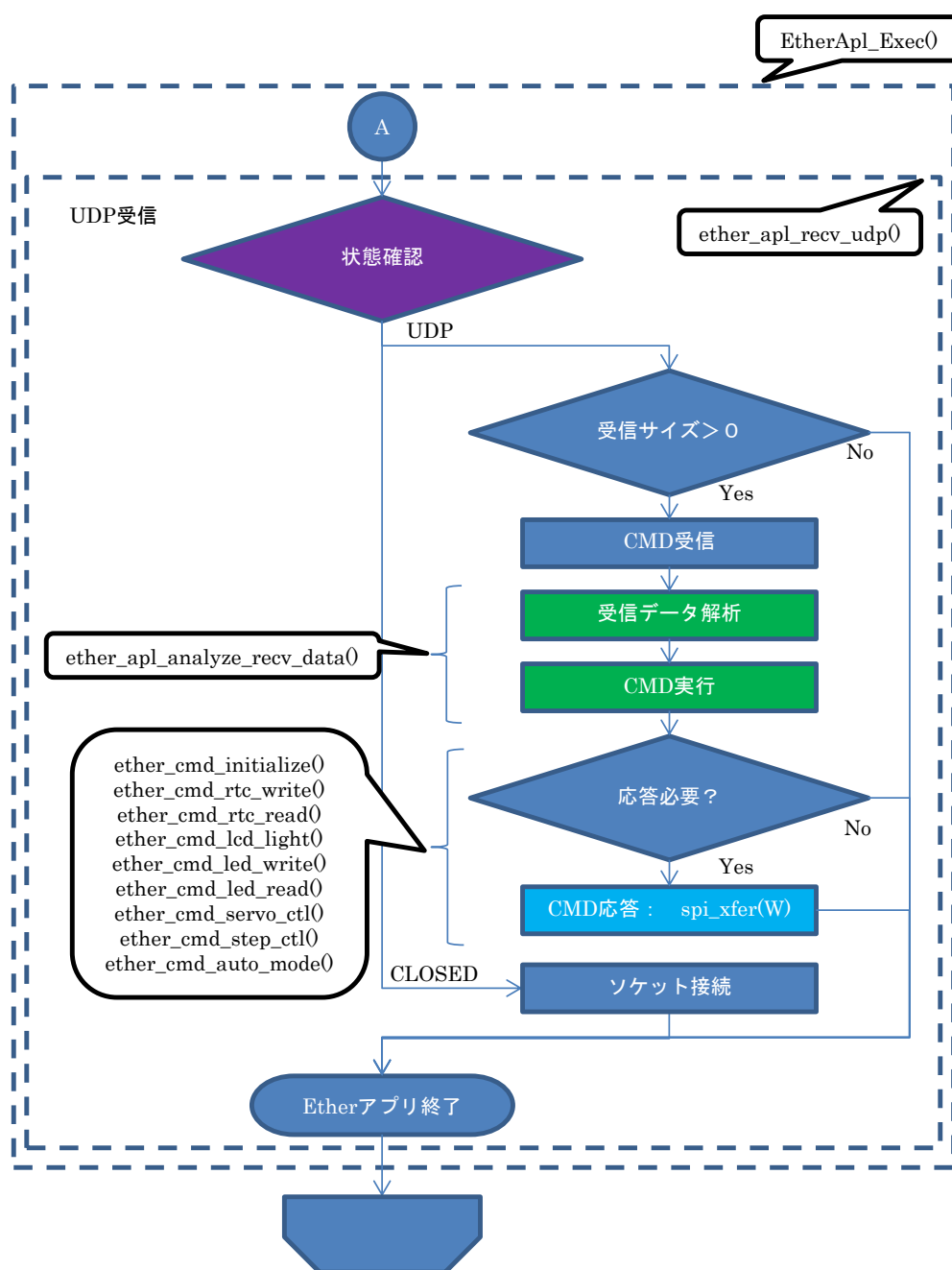


図 4.2 メイン フローチャート 2/2

5. 各定義値、関数の説明

5.1. 定義式

表 5.1 定義式

型	名前	値	説明
—	SOCK_TCPS	0	TCP SOCKET 番号
—	SOCK_UDPS	1	UDP SOCKET 番号
—	ETHER_MAX_BUF	2048	ETHER 用受信バッファサイズ

5.2. 構造体

表 5.2 構造体

型	メンバー		説明
ETHER_CMD_TBL	char	*cmd	受信 CMD 文字列
	void	(*func)(alt_u8 data0, alt_u8 data1, int type, alt_u8 * addr, alt_u16 port)	受信 CMD に対する実行関数

5.3. 変数

表 5.3 変数

型	名前	初期値	説明
wiz_NetInfo	ether_net_info.mac	0x00, 0x08, 0xdc, 0xab, 0xcd, 0xef	
	.ip	192, 168, 0, 231	
	.sn	255, 255, 255, 0	
	.gw	192, 168, 1, 1	
	.dns	0, 0, 0, 0	
	.dhcp	NETINFO_STATIC	
alt_u8	ether_buf[ETHER_MAX_BUF]	なし	
ETHER_CMD_TBL	ether_cmd_tbl[]	{"IT", ether_cmd_initialize }	
		{"RW", ether_cmd_rtc_write }	
		{"RR", ether_cmd_rtc_read }	
		{"BC", ether_cmd_lcd_light }	
		{"LW", ether_cmd_led_write }	
		{"LR", ether_cmd_led_read }	
		{"SV", ether_cmd_servo_ctl }	
		{"SP", ether_cmd_step_ctl }	
		{"AU", ether_cmd_auto_mode }	

5.4. 関数

5.4.1. Global 関数

5.4.1.1. main 関数

表 5.4 main 関数

関数名	main
引数	なし
戻り値	int
概要	メイン関数
処理詳細	ボードの初期化 Ether アプリの初期化 ネット接続設定 メインループ ・Ether メイン処理実行

5.4.1.2. EtherApl_Init 関数

表 5.5 EtherApl_Init 関数

関数名	EtherApl_Init
引数	なし
戻り値	なし
概要	Ether アプリの初期化
処理詳細	W5500 デバイスの初期化

5.4.1.3. EtherApl_Set_NetConfig 関数

表 5.6 EtherApl_Set_NetConfig 関数

関数名	EtherApl_Set_NetConfig
引数	なし
戻り値	なし
概要	Ether アプリのネット接続設定
処理詳細	ネット接続設定

5.4.1.4. EtherApl_Exec 関数

表 5.7 EtherApl_Exec 関数

関数名	EtherApl_Exec
引数	なし
戻り値	なし
概要	Ether アプリの実行
処理詳細	TCP 受信処理 UDP 受信処理

5.4.2. static 関数

5.4.2.1. ether_apl_rcv_tcp 関数

表 5.8 ether_apl_rcv_tcp 関数

関数名	ether_apl_rcv_tcp
引数	alt_u16 port : ポート番号
戻り値	Int
概要	TCP 受信処理
処理詳細	<p>TCP ソケットの各状態で以下を実行</p> <ul style="list-style-type: none"> •ESTABLISHED データ受信サイズを確認、0 以上ならデータを受信 受信データの解析と各 CMD 処理を実行 •CLOSE_WAIT 切断処理を実行 •INIT Listen 処理を実行 •CLOSED Socket 処理を実行

5.4.2.2. ether_apl_rcv_udp 関数

表 5.9 ether_apl_rcv_udp 関数

関数名	ether_apl_rcv_udp
引数	alt_u16 port : ポート番号
戻り値	Int
概要	UDP 受信処理
処理詳細	<p>UDP ソケットの各状態で以下を実行</p> <ul style="list-style-type: none"> •UDP データ受信サイズを確認し、0 以上ならデータを受信 受信データの解析と各 CMD 処理を実行 •CLOSED Socket 処理を実行

5.4.2.3. ether_apl_analyze_rcv_data 関数

表 5.10 ether_apl_analyze_rcv_data 関数

関数名	ether_apl_analyze_rcv_data
引数	alt_u8 *buf : 受信バッファ int type : TCP or UDP alt_u8 *addr : UDP 用 アドレス alt_u16 port : UDP 用 ポート
戻り値	なし
概要	CMD 解析処理
処理詳細	<p>先頭が「:」であることの確認 ether_cmd_tbl[]から、受信データの CMD を比較 一致した場合、Data 領域の解析を実行 該当する CMD の各関数呼び出しを実行</p>

5.4.2.4. ether_cmd_initialize 関数

表 5.11 ether_cmd_initialize 関数

関数名	ether_cmd_initialize
引数	alt_u8 data0 alt_u8 data1 int type alt_u8 *addr alt_u16 port
戻り値	なし
概要	各 CMD 実行処理 ("IT")
処理詳細	応答処理 自動モードをすべて停止 サーボをセンターに変更 LED を全消灯

5.4.2.5. ether_cmd_rtc_write 関数

表 5.12 ether_cmd_rtc_write 関数

関数名	ether_cmd_rtc_write
引数	alt_u8 data0 alt_u8 data1 int type alt_u8 *addr alt_u16 port
戻り値	なし
概要	各 CMD 実行処理 ("RW")
処理詳細	応答処理 ブザー音を鳴らす動作要求 指定 RTC の書き込み要求

5.4.2.6. ether_cmd_rtc_read 関数

表 5.13 ether_cmd_rtc_read 関数

関数名	ether_cmd_rtc_read
引数	alt_u8 data0 alt_u8 data1 int type alt_u8 *addr alt_u16 port
戻り値	なし
概要	各 CMD 実行処理 ("RR")
処理詳細	指定 RTC の読み込み要求 取得したデータを Ether 経由で出力 応答処理

5.4.2.7. ether_cmd_lcd_light 関数

表 5.14 ether_cmd_lcd_light 関数

関数名	ether_cmd_lcd_light
引数	alt_u8 data0 alt_u8 data1 int type alt_u8 *addr alt_u16 port
戻り値	なし
概要	各 CMD 実行処理 ("BC")
処理詳細	応答処理 (未使用)

5.4.2.8. ether_cmd_led_write 関数

表 5.15 ether_cmd_led_write 関数

関数名	ether_cmd_led_write
引数	alt_u8 data0 alt_u8 data1 int type alt_u8 *addr alt_u16 port
戻り値	なし
概要	各 CMD 実行処理 ("LW")
処理詳細	応答処理 ブザー音を鳴らす動作要求 任意点灯用パターン設定

5.4.2.9. ether_cmd_led_read 関数

表 5.16 ether_cmd_led_read 関数

関数名	ether_cmd_led_read
引数	alt_u8 data0 alt_u8 data1 int type alt_u8 *addr alt_u16 port
戻り値	なし
概要	各 CMD 実行処理 ("LR")
処理詳細	LED の出力状態を取得 Ether で結果を出力 応答処理

5.4.2.10. ether_cmd_servo_ctl 関数

表 5.17 ether_cmd_servo_ctl 関数

関数名	ether_cmd_servo_ctl
引数	alt_u8 data0 alt_u8 data1 int type alt_u8 *addr alt_u16 port
戻り値	なし
概要	各 CMD 実行処理 ("SV")
処理詳細	指定された角度でサーボを駆動 ・0 より小さい場合、NG を応答 ・0 以上の場合、OK を応答 応答処理

5.4.2.11. ether_cmd_step_ctl 関数

表 5.18 ether_cmd_step_ctl 関数

関数名	ether_cmd_step_ctl
引数	alt_u8 data0 alt_u8 data1 int type alt_u8 *addr alt_u16 port
戻り値	なし
概要	各 CMD 実行処理 ("SP")
処理詳細	CMD モードを開始 ・0 より小さい場合、NG を応答 ・0 以上の場合、OK を応答

5.4.2.12. ether_cmd_auto_mode 関数

表 5.19 ether_cmd_auto_mode 関数

関数名	ether_cmd_auto_mode
引数	alt_u8 data0 alt_u8 data1 int type alt_u8 *addr alt_u16 port
戻り値	なし
概要	各 CMD 実行処理 ("AU")
処理詳細	応答処理 ブザー音を鳴らす動作要求 Servo/Stepping/Led/Adc の各デモアプリにそれぞれのモード指定を設定

6. 更新履歴

Ver.	更新日付	内容
1.0.0	2015/09/14	新規作成