

**HI.CommunicationEngine**  
**MAILクライアント**  
**リファレンスマニュアル**  
**SMTP編**

## ご注意

1. 本製品(ソフトウェア製品及びその関連ソフトウェア製品を含む。以下、同じ。)の使用に際しては、「外国為替及び外国貿易法」等、技術輸出に関する日本及び関連諸国の関係法規の遵守が必要となります。
2. 弊社は、本製品の使用に際しては、弊社もしくは第三者の特許権、著作権、商標権、その他の知的所有権等の権利に関し、別途、個別の契約書等(マニュアルの記載を含む。以下、同じ。)にて弊社による明示的な許諾がある場合を除き、その保証または実施権の許諾を行うものではありません。また本製品を使用したことにより第三者の知的所有権等の権利に関わる問題が生じた場合、弊社はその責を負いませんので予めご了承ください。
3. 本製品およびその仕様、またはマニュアルに記載されている事柄については、将来、事前の予告なしに変更することがありますので、最終的な設計、ご購入、ご使用に際しましては、事前に最新の製品規格または仕様書(マニュアルを含む)をご確認ください。
4. 本製品の使用(マニュアル記載事項に基づくものも含む)により直接または間接に生ずるいかなる損害についても、弊社は一切の責任を負いません。また、本製品の配布に使用される搭載機器や媒体が原因の損害に対しましても、弊社は一切の責任を負いません。
5. 本製品を、宇宙、航空、原子力、燃焼制御、運輸、交通、各種安全装置、ライフサポート関連の医療機器等のように、特別な品質・信頼性が要求され、その故障や誤動作が直接人命を脅かしたり、人体に危害を及ぼす恐れのある用途向けには使用できません。お客様の用途がこれに該当するかどうか疑問のある場合には、事前に弊社営業担当迄ご相談をお願い致します。
6. 本製品を使用してお客様のシステム製品を設計される際には、通常予測される故障発生率、故障モードをご考慮の上、本製品の動作が原因での事故、その他の拡大損害を生じないようにフェールセーフ等の十分なシステム上の対策を講じて頂きますようお願い致します。
7. 本製品およびマニュアルの著作権は弊社が所有しております。お客様は、弊社から提供された本製品を、別途、個別の契約書等にて定める場合を除き、いかなる場合においても全体的または部分的に複写・解析・改変することはできないものとします。
8. お客様は、別途、個別の契約書等にて定める場合を除き、本製品のマニュアルの一部または全部を無断で使用、複製することはできません。
9. 弊社は、本製品を1台のコンピュータで使用する権利をお客様に対してのみ許諾します。よって、本製品を第三者へ譲渡、貸与、賃借することは許諾しないものとします。但し、別途、個別の契約書等にて定められる場合はその条件に従います。
10. 本製品をはじめ弊社製品およびその関連製品についてのお問い合わせ、ご相談は弊社営業担当迄お願い致します。

$\mu$ ITRON は、Micro Industrial TRON の略称です。TRON は、The Realtime Operating system Nucleus の略称です。

その他、本書で登場するシステム名、製品名は各社の登録商標または商標です。

---

## はじめに

---

このマニュアルは、HI.CommunicationEngine TCP/IPマネージャ上で動作するTCP/IPネットワークアプリケーション「SMTPクライアント」について説明します。

HI.CommunicationEngine SMTPクライアントは、ネットワーク上のメールサーバに接続してメールの送信を行う機能を提供します。

このリファレンスマニュアルではSMTPクライアントのサービスコールとその使い方および関連事項を説明します。TCP/IPマネージャについては関連マニュアルを参照してください。

### 【関連マニュアル】

- HI.CommunicationEngine TCP/IPマネージャ リファレンスマニュアル
- 使用する $\mu$ ITRON仕様OS のユーザーズマニュアル

---

# 目次

---

<b>1</b>	<b>概要</b> .....	<b>1</b>
1.1	機能 .....	1
1.2	関連するマネージャ .....	1
1.3	構成 .....	2
<b>2</b>	<b>SMTPの使用法</b> .....	<b>3</b>
2.1	SMTPクライアントの制限事項 .....	3
2.2	SMTPの使用法 .....	4
<b>3</b>	<b>サービスコール</b> .....	<b>5</b>
3.1	SMTPサービスコール .....	7
3.1.1	<i>SMTP_init</i> SMTP初期化 .....	7
3.1.2	<i>SMTP_strat</i> SMTPの起動 .....	8
3.1.3	<i>SMTP_stop</i> SMTPの停止 .....	9
3.1.4	<i>SMTP_createMailInfo</i> SMTP管理テーブルの初期化 .....	10
3.1.5	<i>SMTP_addRcpt</i> 宛先メールアドレスの登録 .....	13
3.1.6	<i>SMTP_addFile</i> 送信ファイル情報の登録 .....	14
3.1.7	<i>SMTP_addHeaderField</i> ヘッダフィールドの登録 .....	15
3.1.8	<i>SMTP_encode</i> 送信メールの生成 .....	17
3.1.9	<i>SMTP_sendMail</i> メール送信 .....	19
3.1.10	<i>SMTP_deleteMailInfo</i> SMTP管理テーブルの返却 .....	21
<b>4</b>	<b>SMTPクライアントの使用例</b> .....	<b>22</b>
4.1	メール送信 .....	22
4.1.1	メール送信の手順 .....	22
4.1.2	プログラム例 .....	23
4.1.3	プログラム例の解説 .....	25
4.1.4	<i>MailInfo</i> 構造体 .....	27

---

## 図表目次

---

図 1-1	SMTPクライアント使用時のプログラム構成例 .....	2
図 4-1	メール送信の手順 .....	22
図 4-2	メール送信時のMailInfoの内容 .....	27
表 2-1	SMTPクライアントの制限事項、限界値 .....	3
表 3-1	SMTPサービスルール一覧 .....	6



---

# 1 概要

---

## 1.1 機能

SMTPクライアントはSMTP (Simple Mail Transfer Protocol)、SMTP-AUTH (ユーザ認証機能)、MIME (Multipurpose Internet Mail Extension) を実現します。

- (1) SMTP (Simple Mail Transfer Protocol)  
メールの送信を行います。
- (2) SMTP-AUTH (SMTP Authentication)  
以下のユーザ認証方式に対応しています。
  - ・ PLAIN
  - ・ LOGIN
  - ・ CRAM-MD5
- (3) MIME (Multipurpose Internet Mail Extension)  
送信メールに添付されるファイルをエンコードします。

## 1.2 関連するマネージャ

SMTPクライアントは、TCP/IPマネージャを介してネットワーク上のSMTPサーバにアクセスします。TCP/IPマネージャとのインタフェースにはITRON TCP/IP APIを使用しています。SMTPクライアントを使用するためには、TCP/IPマネージャが動作していることが必要です。

### 1.3 構成

図 1-1に SMTP クライアント使用時のプログラム構成例を示します。

ユーザプログラムは SMTP クライアントに対し、SMTP サーバへ送信するメールデータを指定してサービスコールを発行します。

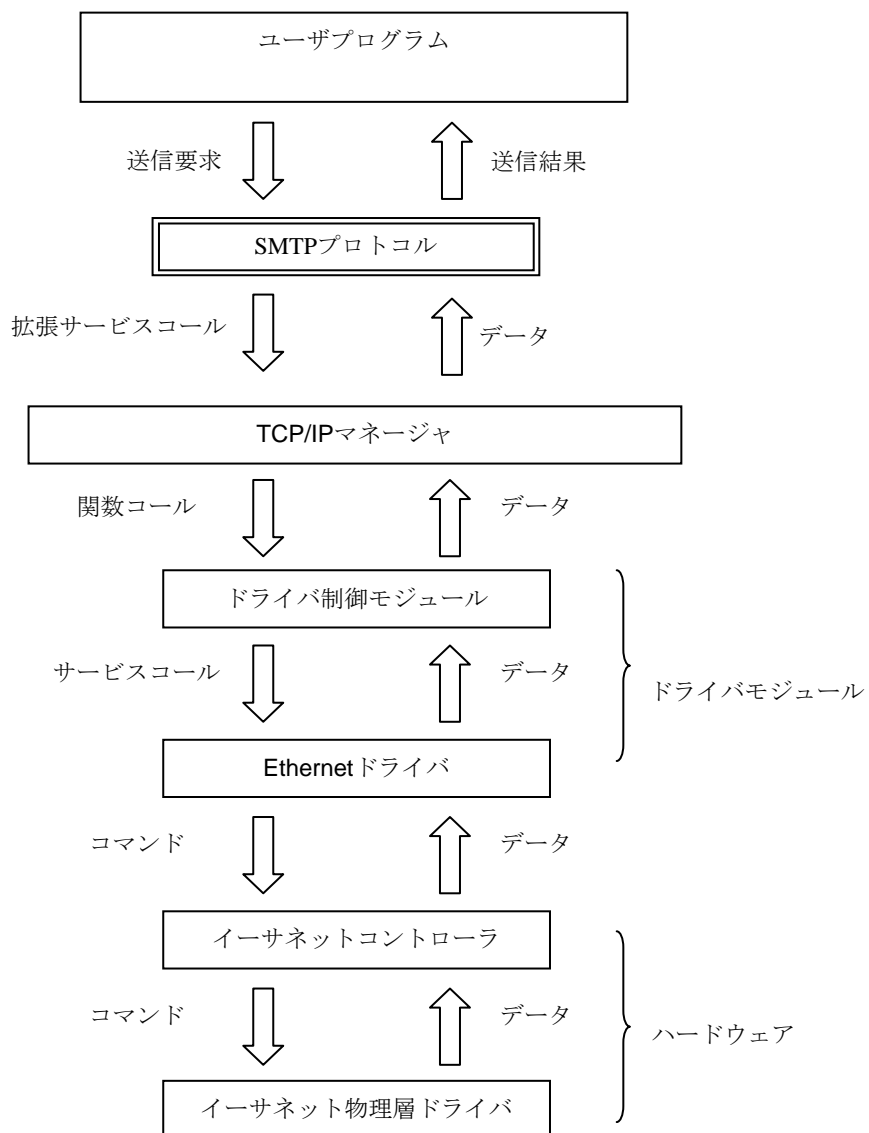


図 1-1 SMTPクライアント使用時のプログラム構成例



---

## 2 SMTP の使用方法

---

### 2.1 SMTP クライアントの制限事項

表 2-1に本 SMTP クライアントの制限事項、限界値を示します。

表 2-1 SMTPクライアントの制限事項、限界値

項番	項目	制限事項 / 限界値
1	OSリソース	可変長メモリプールを1つ使用します 拡張サービスコールを9個使用します*3
2	送信できるメール本体のサイズ	生成したメールの格納領域に依存*1
3	送信できる添付ファイルの数	5
4	送信できる添付ファイルのサイズ	生成したメールの格納領域に依存*1
5	送信できる添付ファイルの種類	テキスト/バイナリ
6	エンコード方式	Base64 エンコード (固定)
7	メールアドレス長 (ユーザ名、ドメイン名含む)	100 文字 (ヌルを含む)
8	指定できる送信先	(TO,CC,BCC) の合計で 100 まで
9	メールサーバのホスト名	100 文字 (ヌルを含む)
10	メッセージ本文	テキスト (ASCII/シフト JIS コード*2)
11	メッセージタイトル文 (件名)	テキスト (ASCII/シフト JIS コード*2)
12	SMTP サーバへのログインユーザ名	64 文字
13	SMTP サーバへのログインパスワード	64 文字

\*1 ユーザが確保する領域です。

\*2 シフト JIS コードは ISO-2022-JP に変換されます。その他の 2 バイトコードは未サポートです。

\*3 使用する OS に拡張 SVC 機能が存在しない場合は該当しません。

## 2.2 SMTP の使用方法

SMTP はメールの送信機能を実現します。

SMTP を使うためには、可変長メモリープール ID および可変長メモリープールサイズを SMTP\_start で登録しなければなりません。

また、POP3サービスコール内でget\_tim()を使用しているので、POP3サービスコール実行中はset\_tim()を発行しないようにしてください。

```
ER ercd = SMTP_start ( MailSmtPar *par );
```

```
typedef struct {
    ID      mplid;          SMTPが使用する可変長メモリープールID
    INT     mplsz;          SMTPが使用する可変長メモリープールサイズ
    UH      timeinterval;  μITRON仕様OSで使用するtmoutの1カウント当りの時間 (ミリ秒)
    H       perrno;         エラー詳細情報
} MailSmtPar ;
```

- (1) **mplid**                      SMTPメモリープールID  
SMTPで使用する可変長メモリープールIDを設定します。  
mplidに0を指定すると、空いている可変長メモリープールIDを探して生成します (オブジェクトの動的生成機能およびID番号自動割付け機能を有するOSの場合のみ)。
- (2) **mplsz**                      SMTPメモリープールサイズ  
SMTPで使用する可変長メモリープールサイズを設定します。  
可変長メモリープールからは以下の領域を確保します。

項番	サービスコール	確保する領域サイズ*(byte)	解放箇所
1	SMTP_createMailInfo	1784、および40	SMTP_deleteMailInfo
2	SMTP_addRcpt	宛先メールアドレス文字数+1、 および (受信者数+1) ×4	SMTP_deleteMailInfo
3	SMTP_addFile	104 × (添付ファイル数+1)	SMTP_deleteMailInfo
4	SMTP_addHeaderField	フィールド名文字数+フィールド・ ボディ文字数+1、および (添付ファイル数+1) ×4	SMTP_deleteMailInfo
5	SMTP_encode	メール本文の文字数×2+10	SMTP_encode
6	SMTP_sendMail	TCPウィンドウバッファ情報の送信 用ウィンドウバッファサイズ (sbufsz)、および受信用ウィンドウバ ッファサイズ(rbufsz)	SMTP_sendMail

※ μITRON仕様OSで使用する管理領域のサイズは含みません。また、4の倍数でない場合は4の倍数になるように切り上げます。

TCPウィンドウバッファは、TCPウィンドウバッファ情報にてウィンドウバッファの領域を指定しなかった場合のみメモリープールから確保されます。

- (3) **timeinterval**            μITRON仕様OSで使用するtmoutの1カウント当りの時間 (ミリ秒)  
以下の値を指定できます。  
【 1, 2, 4, 5, 10, 20 】

SMTPの機能を終了するには、SMTP\_stopを発行します。これによりSMTP\_stratで確保した可変長メモリープールを削除します (オブジェクトの動的削除機能を有するOSの場合のみ)。

### 3 サービスコール

サービスコールは、ユーザアプリケーションがSMTPクライアントを利用する場合のインタフェースを提供します。

本節では、サービスコールについての詳細な説明を以下の形式で行っています。

No.	サービスコール名	機能
		【発行可能なシステム状態*1】
	C言語インタフェース	
	サービスコール呼出し形式	
	パラメータ	
	型	パラメータ
	・	・
	・	・
	・	・
	リターンパラメータ*3	
	型	パラメータ
	・	・
	・	・
	パケットの構造	
	リターン値/エラーコード*3	
	リターン値またはニモニック	リターン値またはエラーコードの意味*2
	・	・
	・	・
	・	・
	解説*3	
	・・・・・・・・	

\*1 発行可能なシステム状態を以下のアルファベットで示します

- T：タスク実行状態
- D：ディスパッチ禁止状態
- L：CPUロック状態
- I：非タスク部実行状態

なお、各状態の詳細は使用するμITRON仕様OSのユーザーズマニュアルを参照してください。

発行可能なシステム状態以外の状態でサービスコールを発行した場合、システムの正常な動作は保証されません。

\*2 エラーコードE\_PARの理由として、アドレスが4の倍数以外、アドレスが奇数についてのエラーは、奇数アドレスからの16ビットや32ビットアクセスが可能なマイコン向けの製品では発生しません。

\*3 拡張SVC機能およびオブジェクトの動的な生成(ID番号自動割付け含む)・削除機能を有さないOSを使用する場合は、該当しない説明が含まれます。

表 3-1 にSMTPサービスコール一覧を示します。

**表 3-1 SMTPサービスコール一覧**

項番	SMTPサービスコール名	機能
1	SMTP_init	SMTPの初期化
2	SMTP_start	SMTPの起動
3	SMTP_stop	SMTPの停止
4	SMTP_createMailInfo	SMTP管理テーブルの初期化
5	SMTP_addRcpt	宛先メールアドレスの登録
6	SMTP_addFile	送信ファイル情報の登録
7	SMTP_addHeaderField	ヘッダフィールドの登録
8	SMTP_encode	送信メールの生成
9	SMTP_sendMail	メールの送信
10	SMTP_deleteMailInfo	SMTP管理テーブルの返却

## 3.1 SMTP サービスコール

### 3.1.1 SMTP\_init SMTP 初期化

【T/D/L/I】

C 言語インタフェース

```
void SMTP_init ( void );
```

パラメータ

無し

リターンパラメータ

無し

解 説

SMTPクライアントを初期化します。

SMTPの内部変数を初期化し、各サービスコールを $\mu$ ITRON仕様OSの拡張サービスコールに登録します。

SMTP\_init は、SMTPを使用する前に1回だけ必ず実行してください（初めてSMTP\_startを呼び出す前に1回だけ実行し、その後は実行しないでください）。

### 3.1.2 SMTP\_strat SMTP の起動

【T】

C 言語インタフェース

```
ER ercd = SMTP_start (MailSmtppar *par);
```

パラメータ

MailSmtppar	*par	SMTP管理情報の先頭アドレス
-------------	------	-----------------

リターンパラメータ

ER	ercd	リターン値
ID	par-> mplid	SMTPが使用する可変長メモリプールID (mplid=0指定の場合)

パケットの構造

```
MailSmtppar *par
typedef struct {
    ID          mplid;          SMTPが使用する可変長メモリプールID
    INT         mplsz;         SMTPが使用する可変長メモリプールサイズ
    UH         timeinterval;   μ ITRON仕様OSで使用するtmoutの1カウント当りの時間 (ミリ秒)
    H          perno;         エラー詳細情報
} MailSmtppar ;
```

リターン値/エラーコード

E_OK	正常終了
E_PAR	パラメータエラー (詳細エラーコード参照)
E_OBJ	オブジェクト状態不正 (すでに起動している)
E_SYS	メモリプールの生成に失敗した、または拡張SVCの登録に失敗した
E_RSFN	拡張SVCの登録に失敗した

詳細エラーコード (E\_PAR発生時)

EMAIL_MEMSIZ	1	mplszが0、4の倍数以外
EMAIL_ID	2	mplidが不正 (ID未登録)
EMAIL_ILTIME	5	timeintervalが不正

解 説

SMTPを起動します。

SMTPが使用するメモリプールを確保します。

parが0または4の倍数以外の場合は、エラーコードとしてE\_PARを返します。この場合はpernoに詳細エラーコードの設定は行いません。E\_PAR発生時に必ずpernoを参照する場合は、本サービスコールを呼び出す前にpernoに0を設定してください。

SMTP\_initにて拡張サービスコールの登録に失敗した場合は、エラーコードとしてE\_RSFNまたはE\_SYSを返します。

mplidにはSMTPで使用する可変長メモリプールIDを指定します。mplidに0を指定すると、空いている可変長メモリプールIDを探して生成します。

mplidに0以外の値を指定する場合は、構築時に登録した最大可変長メモリプールID以内を指定してください。範囲を超えた可変長メモリプールIDを指定された場合は、正常な動作を保証できません。

mplszにはSMTPが使用する可変長メモリプールサイズを指定します。

timeintervalには、以下の値を指定できます。

【 1, 2, 4, 5, 10, 20 】

### 3.1.3 SMTP\_stop

### SMTP の停止

【T】

C 言語インタフェース

```
ER ercd = SMTP_stop (void);
```

パラメータ

なし

リターンパラメータ

ER	ercd	リターン値
----	------	-------

リターン値/エラーコード

E_OK	正常終了
------	------

E_OBJ	オブジェクト状態不正 (すでに終了している)
-------	------------------------

解説

SMTPを停止します。

SMTPのサービスを終了し、SMTPが使用した可変長メモリプールを削除します。

他のSMTPサービスコールの処理中に本サービスコールは使用しないでください。使用した場合の動作の保証はできません。

### 3.1.4 SMTP\_createMailInfo

### SMTP 管理テーブルの初期化

【T】

C 言語インタフェース

```
ER ercd = SMTP_createMailInfo (MailInfo **p_pmi);
```

パラメータ

MailInfo \*\*p\_pmi 送信メールの管理情報を返す領域のアドレス

リターンパラメータ

ER ercd リターン値またはエラーコード  
MailInfo \*\*p\_pmi 送信メールの管理情報を格納した領域のアドレス

パケットの構造

```
typedef struct tagMailInfo{
    MailInfo_Reserved *pReserved;
    H nErr;
    H nPort;
    H numOfRcpt;
    B pszErrStr[MAX_PATH];
    B pszSubject[MAX_LENGTH];
    B pszMailAddress[MAX_LENGTH];
    B pszDomainName[MAX_LENGTH];
    B userLogin[MAX_SMTP_LENGTH+1];
    B passWord[MAX_SMTP_LENGTH+1];
    H nAuthType;
    UW smtpServer;
    MailWinBuf winBufInfo;
    UW myIpAddr;
    H myPortNo;
    ID cepId;
} MailInfo;

typedef struct tagMailInfo_Reserved{
    XfileInfo *aFileInfo;
    BOOL fAlternative;
    H nFile;
    H nField;
    H anRcpt[MAX_RCPTTYPE];
    B **aapList[MAX_RCPTTYPE];
    B **apField;
} MailInfo_Reserved;

typedef struct tagXFileInfo{
    H nMode;
    B pszFileName[MAX_LENGTH];
} XFileInfo;

typedef struct {
    UW *sbuf;
    INT sbufsz;
    UW *rbuf;
    INT rbufsz;
} MailWinBuf;
```

送信メールの管理情報

- MailInfo\_Reserved \*pReserved: ヘッダと宛先情報のアドレス
- H nErr: 発生したエラーの詳細を示す番号
- H nPort: SMTPポート番号(Default:25)
- H numOfRcpt: 送信するメールの宛先 (受信者) 数
- B pszErrStr[MAX\_PATH]: エラー内容を示す文字列 (MAX\_PATH = 256)
- B pszSubject[MAX\_LENGTH]: 件名 (MAX\_LENGTH = 100)
- B pszMailAddress[MAX\_LENGTH]: 送信者のメールアドレス
- B pszDomainName[MAX\_LENGTH]: クライアントのホスト名
- B userLogin[MAX\_SMTP\_LENGTH+1]: 認証ユーザ名 (MAX\_SMTP\_LENGTH = 64)
- B passWord[MAX\_SMTP\_LENGTH+1]: 認証パスワード
- H nAuthType: 認証タイプ
- UW smtpServer: SMTPサーバアドレス
- MailWinBuf winBufInfo: TCPウィンドウバッファ情報
- UW myIpAddr: 自IPアドレス (IP Address)
- H myPortNo: 自ポート番号 (Default:1024)
- ID cepId: TCP通信端点ID

ヘッダと宛先情報

- XfileInfo \*aFileInfo: 送信ファイル情報のアドレス
- BOOL fAlternative: メール内容 (Content-Type) 情報
- H nFile: 添付ファイルの数
- H nField: ヘッダフィールドの数
- H anRcpt[MAX\_RCPTTYPE]: 宛先 (TO/CC/BCC) 毎の受信者数 (MAX\_RCPTTYPE = 4)
- B \*\*aapList[MAX\_RCPTTYPE]: 宛先 (TO/CC/BCC) 毎のリスト
- B \*\*apField: ヘッダフィールドのリスト

送信ファイル情報

- H nMode: ファイル種別
- B pszFileName[MAX\_LENGTH]: ファイル名 (MAX\_LENGTH = 100)

TCPウィンドウバッファ情報

- UW \*sbuf: 送信用ウィンドウバッファの先頭アドレス
- INT sbufsz: 送信用ウィンドウバッファのサイズ
- UW \*rbuf: 受信用ウィンドウバッファの先頭アドレス
- INT rbufsz: 受信用ウィンドウバッファのサイズ

リターン値/エラーコード

E\_OK 正常終了



E_PAR	パラメータエラー (p_pmiが0、または4の倍数以外)
E_NOMEM	メモリ不足 (必要なメモリが確保できない)
E_OBJ	オブジェクト状態不正 (SMTPが起動されていない)

## 解 説

「送信メールの管理情報」と「ヘッダと宛先情報」のメモリを獲得し、初期化します。

本サービスコールが正常終了した後に、「送信メールの管理情報」の以下の情報を設定してください。

- SMTPポート番号 (nPort)
  - 本サービスコールにて、25に初期化します。
  - サブミッションポート (SMTP認証) を使用する場合は587に変更してください。
- 件名 (numOfRcpts)
  - メールの件名を設定してください。最大99文字 (MAX\_LENGTH - 1) まで指定できます。文字列の終端にはヌル(0)を設定してください。
- 送信者のメールアドレス (pszMailAddress)
  - 送信者のメールアドレスを設定してください。最大99文字 (MAX\_LENGTH - 1) まで指定できます。文字列の終端にはヌル(0)を設定してください。
- クライアントのホスト名 (pszDomainName)
  - クライアントのホスト名 (ドメイン名) を設定してください。最大99文字 (MAX\_LENGTH - 1) まで指定できます。文字列の終端にはヌル(0)を設定してください。
- 認証ユーザ名 (userLogin)
  - SMTP認証を使用する場合のみ設定が必要です。認証に使用するユーザ名を設定してください。最大64文字 (MAX\_SMTP\_LENGTH) まで指定できます。文字列の終端にはヌル(0)を設定してください。
- 認証パスワード (passWord)
  - SMTP認証を使用する場合のみ設定が必要です。認証に使用するパスワードを設定してください。最大64文字 (MAX\_SMTP\_LENGTH) まで指定できます。文字列の終端にはヌル(0)を設定してください。
- 認証タイプ (nAuthType)
  - 本サービスコールにて、AUTH\_NONE (0) 認証なしに初期化します。
  - 使用するSMTP認証タイプを設定してください。
  - AUTH\_PLAIN (1) 認証方式にPLAINを使用
  - AUTH\_LOGIN (2) 認証方式にLOGINを使用
  - AUTH\_CRAM\_MD5 (3) 認証方式にCRAM-MD5を使用
- SMTPサーバアドレス (smtpServer)
  - SMTPサーバのIPアドレスを設定してください。
- 送信用ウィンドウバッファの先頭アドレス (sbuf)
  - TCP送信ウィンドウバッファに使用するメモリ領域を設定してください。sbufszに設定されたサイズ分の領域が必要です。ヌル (0) を指定した場合は、sbufszに設定されたサイズの領域を可変長メモリプールから獲得、使用します。
- 送信用ウィンドウバッファのサイズ (sbufsz)
  - TCPの送信ウィンドウバッファに使用するメモリ領域のサイズを設定してください。2048以上を設定する必要があります。
- 受信用ウィンドウバッファの先頭アドレス (rbuf)
  - TCP受信ウィンドウバッファに使用するメモリ領域を設定してください。rbufszに設定されたサイズ分の領域が必要です。ヌル (0) を指定した場合は、rbufszに設定されたサイズの領域を可変長メモリプールから獲得、使用します。
- 受信用ウィンドウバッファのサイズ (rbufsz)
  - TCPの受信ウィンドウバッファに使用するメモリ領域のサイズを設定してください。2048以上を設定する必要があります。
- 自IPアドレス (myIpAddr)
  - 使用する自IPアドレス (TCP/IPマネージャに登録したIPアドレス) を設定してください。0を指定した場合は、TCP/IPマネージャにて動作を開始しているIPアドレスを使用します。
- 自ポート番号 (myPortNo)
  - 使用する自ポート番号を設定してください。デフォルトでは1024を使用します。

p\_pmiが0または4の倍数以外の場合は、エラーコードとしてE\_PARを返します。本サービスコールでE\_PARが発生した場合は、「送信メールの管理情報」の獲得前であるため、詳細エラーコードの設定は行いません。

可変長メモリプールから使用するメモリ、および $\mu$ ITRON仕様OSが使う管理領域（詳細は、使用する $\mu$ ITRON仕様OSのマニュアルを参照してください）の合計サイズが獲得できない場合は、エラーコードとしてE\_NOMEMを返します。

### 3.1.5 SMTP\_addRcpt

### 宛先メールアドレスの登録

【T】

C言語インタフェース

```
ER ercd = SMTP_addRcpt (MailInfo *pmi, UH rcpttype, B *rcptadr);
```

パラメータ

MailInfo	*pmi	送信メールの管理情報の先頭アドレス
UH	rcpttype	宛先種別 (TO/CC/BCC)
B	*rcptadr	宛先メールアドレスを格納した領域の先頭アドレス

リターンパラメータ

ER	ercd	リターン値またはエラーコード
----	------	----------------

パケットの構造

送信メールの管理情報の構造に関しては「3.1.4 SMTP管理テーブルの初期化」を参照してください。

リターン値/エラーコード

E_OK	正常終了
E_PAR	パラメータエラー
E_NOMEM	メモリ不足 (必要なメモリが確保できない)
E_OBJ	オブジェクト状態不正 (SMTPが起動されていない)
E_QOVR	宛先数オーバー (既に宛先数を100件登録した)

詳細エラーコード (E\_PAR発生時)

EMAIL_ILTYPE	3	rcpttypeが不正 (RCPT_TO (0) ~RCPT_BCC (2) 以外)
EMAIL_RCPTADR	4	rcptadrが0
EMAIL_STROVR	18	宛先メールアドレスが 99文字を超えた

解 説

「送信メールの管理情報」に宛先メールアドレスを登録します。  
宛先 (受信者) の数だけ本関数を呼び出す必要があります。

pmiが0または4の倍数以外の場合は、エラーコードとしてE\_PARを返します。この場合はnErrに詳細エラーコードの設定は行いません。E\_PAR発生時に必ずnErrを参照する場合は、本サービスコールを呼び出す前にnErrに0を設定してください。

rcpttypeには宛先種別として次の値を指定します。

RCPT\_TO (0) : 本来の受信者  
RCPT\_CC (1) : 複写されたメールの受信者 (carbon copy)  
RCPT\_BCC (2) : 上記の受信者 (TO/CC) に知られない、複写されたメールの受信者 (blind carbon copy)

rcptadrには宛先メールアドレスを最大99文字(MAX\_LENGTH - 1)まで指定できます。文字列の終端にはヌル (0) を設定してください。

可変長メモリプールから使用するメモリ、およびμITRON仕様OSが使う管理領域 (詳細は、使用するμITRON仕様OSのマニュアルを参照してください) の合計サイズが獲得できない場合は、エラーコードとしてE\_NOMEMを返します。

「ヘッダと宛先情報」の内容は変更しないでください。変更した場合は、正常な動作を保証できません。

なお、本サービスコールを呼び出す前には、メール送信の準備処理を行う必要があります。メール送信手順の詳細は、「4.1.1 メール送信の手順」で説明します。

### 3.1.6 SMTP\_addFile

### 送信ファイル情報の登録

【T】

C言語インタフェース

```
ER ercd = SMTP_addFile (MailInfo *pmi, B *attachname, UH filemode);
```

パラメータ

MailInfo	*pmi	送信メールの管理情報の先頭アドレス
B	*attachname	添付ファイル名称を格納した領域の先頭アドレス
UH	filemode	ファイルの種別

リターンパラメータ

ER	ercd	リターン値またはエラーコード
----	------	----------------

パケットの構造

送信メールの管理情報の構造に関しては「3.1.4 SMTP管理テーブルの初期化」を参照してください。

リターン値/エラーコード

E_OK	正常終了
E_PAR	パラメータエラー
E_NOMEM	メモリ不足 (必要なメモリが確保できない)
E_OBJ	オブジェクト状態不正 (SMTPが起動されていない)
E_QOVR	送信ファイル数オーバー (既に5つの添付ファイルを登録した)

詳細エラーコード (E\_PAR発生時)

EMAIL_ATNAMEADR	6	attachnameが0
EMAIL_ILMODE	8	filemodeが不正 (FMODE_BODY (0) またはFMODE_ATTACH (4) 以外)
EMAIL_STROVR	18	ファイル名称が 99文字を超えた

解 説

「送信ファイル情報」にファイル種別とファイル名を登録します。

pmiが0または4の倍数以外の場合は、エラーコードとしてE\_PARを返します。この場合はnErrに詳細エラーコードの設定は行いません。E\_PAR発生時に必ずnErrを参照する場合は、本サービスコールを呼び出す前にnErrに0を設定してください。

filemodeにはファイル種別として次の値を指定します。

FMODE_BODY (0)	メール本体
FMODE_ATTACH (4)	添付ファイル

attachnameにはファイル名を最大99文字(MAX\_LENGTH - 1)まで指定できます。文字列の終端にはヌル(0)を設定してください。ファイル名は添付ファイルだけではなく、メール本体にも必要です。メール本体のファイル名は自由ですが、拡張子は、"txt" にしてください。

可変長メモリプールから使用するメモリ、およびμITRON仕様OSが使う管理領域 (詳細は、使用するμITRON仕様OSのマニュアルを参照してください) の合計サイズが獲得できない場合は、エラーコードとしてE\_NOMEMを返します。

「ヘッダと宛先情報」の内容は変更しないでください。変更した場合は、正常な動作を保証できません。

なお、本サービスコールを呼び出す前には、メール送信の準備処理を行う必要があります。メール送信手順の詳細は、「4.1.1 メール送信の手順」で説明します。

### 3.1.7 SMTP\_addHeaderField

### ヘッダフィールドの登録

【T】

C言語インタフェース

```
ER ercd = SMTP_addHeaderField (MailInfo *pmi, B *fieldname, B *content);
```

パラメータ

MailInfo	*pmi	送信メールの管理情報の先頭アドレス
B	*fieldname	フィールド名を格納した領域の先頭アドレス
B	*content	フィールド・ボディを格納した領域の先頭アドレス

リターンパラメータ

ER	ercd	エラーコードまたはリターン値
----	------	----------------

パケットの構造

送信メールの管理情報の構造に関しては「3.1.4 SMTP管理テーブルの初期化」を参照してください。

リターン値/エラーコード

E_OK	正常終了
E_PAR	パラメータエラー
E_NOMEM	メモリ不足 (必要なメモリが確保できない)
E_OBJ	オブジェクト状態不正 (SMTPが起動されていない)
E_QOVR	ヘッダフィールド数オーバー (既に32767個のヘッダフィールドを登録した)

詳細エラーコード (E\_PAR発生時)

EMAIL_FIELDADR	9	fieldnameが0
EMAIL_CNTNTADR	10	contentが0
EMAIL_STROVR	18	フィールド名、またはフィールド・ボディが 99文字を超えた

解 説

「ヘッダと宛先情報」にフィールド名とフィールド・ボディを設定します。

メールヘッダ部は、通常”:(コロン)”までの「フィールド名」と、それ以降の「フィールド・ボディ」を1つの「ヘッダフィールド」とし、複数のヘッダフィールドで構成されます。ヘッダフィールドの記述順序は特に規定されていません。

ヘッダフィールドのうち、日付 (Date)、送信者 (From)、宛先 (To, cc, bcc)、件名 (Subject)、MIMEのバージョン (MIME-Version) フィールドに関しては、SMTP\_sendMail、またはSMTP\_encodeサービスコールにて自動的に設定します。これらのヘッダフィールド以外に必要なヘッダフィールド情報がある場合は、本サービスコールで設定してください。

ヘッダフィールドは、32767個まで登録できます。既に32767個のヘッダフィールドを登録した状態で、本サービスコールを呼び出した場合は、エラーコードとしてE\_QOVRを返します。

pmiが0または4の倍数以外の場合は、エラーコードとしてE\_PARを返します。この場合はnErrに詳細エラーコードの設定は行いません。E\_PAR発生時に必ずnErrを参照する場合は、本サービスコールを呼び出す前にnErrに0を設定してください。

fieldnameにはフィールド名を最大99文字(MAX\_LENGTH - 1)まで指定できます。文字列の終端にはヌル(0)を設定してください。

contentにはフィールド・ボディを最大99文字(MAX\_LENGTH - 1)まで指定できます。文字列の終端にはヌル(0)を設定してください。

可変長メモリプールから使用するメモリ、およびμITRON仕様OSが使う管理領域 (詳細は、使用するμITRON仕様OSのマニュアルを参照してください) の合計サイズが獲得できない場合、エラーコードとしてE\_NOMEMを返します。

「ヘッダと宛先情報」の内容は変更しないでください。変更した場合は、正常な動作を保証できません。

なお、本サービスコールを呼び出す前には、メール送信の準備処理を行う必要があります。メール送信

手順の詳細は、「4.1.1 メール送信の手順」で説明します。

一般的に使用されている主なヘッダフィールド一覧と、設定するサービスコールを以下に示します。

表 3-2 ヘッダフィールド一覧

分類	フィールド名	意味	自動設定するサービスコール
フォーマットおよびコントロール情報	MIME-Version:	使用されているMIMEのバージョン	SMTP_encode
送信者および受信者表示	From:	メール送信者のアドレス	SMTP_send Mail
	Sender:	実際のメール送信者、または代理人のアドレス (From:と異なる場合)	—
	To:	メール宛先のアドレス	SMTP_encode
	cc:	カーボンコピー宛先のアドレス	SMTP_encode
	bcc:	ブラインドコピー宛先のアドレス	SMTP_encode
	X-Mailer:	クライアントメールソフトに関する情報	—
応答コントロール	Reply-To:	メール返信先のアドレス	—
メッセージの識別と参照ヘッダ	Message-ID:	メールのユニークなID	—
	In-Reply-To:	メール返信時の返信元メールのMessage-ID	—
	References:	他の関連したメールのMessage-ID	—
その他のテキストヘッダ	Keywords:	データベース検索のための検索キー	—
	Subject:	メールの件名	SMTP_encode
	Comments:	メールのコメント	—
	Organization:	メール送信者が所属する組織名	—
	Date:	メールの作成日付	SMTP_send Mail
	Precedence:	メールの優先度	—
	Resent-From:	メール転送時のFrom:を含む	—
	Resent-Sender:	メール転送時のSender:を含む	—
	Resent-To:	メール転送時のTo:を含む	—
	Resent-cc:	メール転送時のcc:を含む	—
Resent-bcc:	メール転送時のbcc:を含む	—	

### 3.1.8 SMTP\_encode

### 送信メールの生成

【T】

#### C言語インタフェース

```
ER ercd = SMTP_encode (MailInfo *pmi, B *outbuff, W outbufflen,
                      MultiAttach *atchbuff, BodyInfo *bodybuff);
```

#### パラメータ

MailInfo	*pmi	送信メールの管理情報の先頭アドレス
B	*outbuff	生成したメールを格納する領域の先頭アドレス
W	outbufflen	生成したメールを格納する領域の長さ
MultiAttach	*atchbuff	添付ファイル情報の先頭アドレス
BodyInfo	*bodybuff	メール本体情報の先頭アドレス

#### リターンパラメータ

ER	ercd	エラーコードまたはリターン値
----	------	----------------

#### パケットの構造

送信メールの管理情報の構造に関しては「3.1.4 SMTP管理テーブルの初期化」を参照してください。

```
typedef struct tagBodyInfo{                               メール本体情報
    UW bodySize;                                         メール本体のサイズ
    B *bodyBuf;                                          メール本体の先頭アドレス
} BodyInfo;

typedef struct tagMultiAttach{                           添付ファイル情報
    B NoOfAttachment;                                   添付ファイル実体情報の数
    AttFileInfo AttInfo[MAX_ATTACH_CNT];               添付ファイル実体情報の先頭アドレス
                                                    (MAX_ATTACH_CNT = 5)
} MultiAttach;

typedef struct tagAttFileInfo{                           添付ファイル実体情報
    B attFileName[MAX_LENGTH];                         添付ファイルの名称
    UW attFileSize;                                    添付ファイルのサイズ
    B *fileBuf;                                        添付ファイルの先頭アドレス
} AttFileInfo;
```

#### リターン値/エラーコード

E_OK	正常終了
E_PAR	パラメータエラー
E_NOMEM	メモリ不足 (必要なメモリが確保できない)
E_OBJ	オブジェクト状態不正 (SMTPが起動されていない)

#### 詳細エラーコード (E\_PAR発生時)

EMAIL_ILTYPE	3	データタイプが不正 (シフトJISコード以外の2バイトコード)
EMAIL_ATCHADR	11	atchbuffが0、または4の倍数以外
EMAIL_BODYADR	12	bodybuffが0、または4の倍数以外
EMAIL_OUTBADR	13	outbuffが0
EMAIL_OUTBLEN	14	outbufflenが0、または負の値
EMAIL_NOCONTENTS	19	エンコードするファイルが存在しない
EMAIL_SUBJECTOVR	40	pszSubjectが99文字を超えた
EMAIL_FROMOVR	41	pszMailAddressが99文字を超えた

#### 詳細エラーコード (E\_NOMEM発生時)

EMAIL_VMPL	50	メモリプール領域不足
EMAIL_BUFOVR	51	バッファ領域不足

#### 解説

送信メールを生成し、「生成したメールを格納する領域(outbuff)」に格納します。

pmiが0または4の倍数以外の場合は、エラーコードとしてE\_PARを返します。この場合はnErrに詳細エラーコードの設定は行いません。E\_PAR発生時に必ずnErrを参照する場合は、本サービスコールを呼び出す前にnErrに0を設定してください。

「生成したメールを格納する領域(outbuff)」、「メール本体情報」および「添付ファイル情報」の領域はユーザアプリケーションにて確保してください。

「生成したメールを格納する領域の長さ(outbufflen)」は、最大で送信するメールの約4倍の長さが必要です。

メール生成中に「生成したメールを格納する領域(outbuff)」が不足した場合は、エラーコードとしてE\_NOMEMを返し、nErrに詳細エラーコードとしてEMAIL\_BUFOVRを設定します。この場合は、「生成したメールを格納する領域(outbuff)」を確保しなおして、再度本サービスコールを呼び出してください。

可変長メモリプールから使用するメモリ、および $\mu$ ITRON仕様OSが使う管理領域（詳細は、使用する $\mu$ ITRON仕様OSのマニュアルを参照してください）の合計サイズが獲得できない場合は、エラーコードとしてE\_NOMEMを返し、nErrに詳細エラーコードとしてEMAIL\_VMPLを設定します。

添付可能なファイルの最大数は5ファイルです。

添付ファイルが存在しない場合は、添付ファイル情報(MultiAttach)内の添付ファイル実体情報の数(NoOfAttachment)を0に設定してください。

「ヘッダと宛先情報」の内容は変更しないでください。変更した場合は、正常な動作を保証できません。

なお、本サービスコールを呼び出す前には、メール送信の準備処理を行う必要があります。

メール送信手順の詳細は、「4.1.1 メール送信の手順」で説明します。



### 3.1.9 SMTP\_sendMail

### メールの送信

【T】

C言語インタフェース

```
ER ercd = SMTP_sendMail (MailInfo *pmi, B *databuffer, B *timebuffer, UH waittime);
```

パラメータ

MailInfo	*pmi	送信メールの管理情報の先頭アドレス
B	*databuffer	送信メールを格納した領域の先頭アドレス
B	*timebuffer	日時の文字列を格納した領域の先頭アドレス
UH	waittime	SMTPサーバからの応答待ち時間 (秒)

リターンパラメータ

ER	ercd	エラーコードまたはリターン値
----	------	----------------

パケットの構造

送信メールの管理情報の構造に関しては「3.1.4 SMTP管理テーブルの初期化」を参照してください。

リターン値/エラーコード

E_OK	正常終了
E_PAR	パラメータエラー
E_NOMEM	メモリ不足 (必要なメモリが確保できない)
E_OBJ	オブジェクト状態不正 (SMTPが起動されていない)
E_TMOUT	タイムアウトエラー (指定時間以内にSMTPサーバからの応答がない)
EV_PROT	SMTPサーバからエラー応答を受信した

詳細エラーコード (E\_PAR発生時)

EMAIL_ILTIME	5	waittimeが0
EMAIL_NUMOFRcpt	15	numOfRcptが0
EMAIL_DATBADR	16	databufferが0
EMAIL_TIMBADR	17	timebufferが0
EMAIL_STROVR	18	日時の情報が 99文字を超えた
EMAIL_NOCONTENTS	19	送信するファイルが存在しない
EMAIL_WINBUF	20	sbufまたはrbufが奇数、sbufszが2048未満、rbufszが2048未満
EMAIL_DOMAINOVR	42	pszDomainNameが99文字を超えた
EMAIL_USNAMEOVR	43	userLoginが64文字を超えた
EMAIL_PASSWDOVR	44	passWordが64文字を超えた
EMAIL_AUTHTYPE	45	nAuthTypeが不正

詳細エラーコード (EV\_PROT発生時)

EMAIL_CONNECT	101	コネクション接続失敗
EMAIL_HELOCMD	102	HELOコマンド失敗
EMAIL_MAILCMD	103	MAILコマンド失敗
EMAIL_RCPTCMD	104	RCPTコマンド失敗
EMAIL_DATACMD	105	DATAコマンド失敗
EMAIL_QUITCMD	106	QUITコマンド失敗
EMAIL_AUTHCMD	107	AUTHコマンド失敗

解説

SMTPサーバとコネクション接続を行い、SMTPに関する各コマンド (HELO, MAIL, RCPT, DATA and QUIT) を発行しメールを送信します。

コネクション接続後は、SMTPサーバからの応答をwaittimeで指定された時間まで待ちます。

pmiが0または4の倍数以外の場合は、エラーコードとしてE\_PARを返します。この場合はnErrに詳細エラーコードの設定は行いません。E\_PAR発生時に必ずnErrを参照する場合は、本サービスコールを呼び出す前にnErrに0を設定してください。

databufferには、SMTP\_encodeサービスコールにて生成した送信メールを格納した領域の先頭アドレスを

設定してください。

`timebuffer`には日時の情報を設定してください。日時情報は次のように指定します。

(例) `Mon,27 May 02 11:24:05 + 0900`

上記のように、「曜日,日 月 年 時:分:秒 + 地域差時分」で指定します。

曜日の指定はアルファベット 3 文字で指定します。

<Mon / Tue / Wed / Thu / Fri / Sat / Sun>

また、月の指定もアルファベット 3 文字で指定します。

<Jan / Feb / Mar / Apr / May / Jun / Jul / Aug / Sep / Oct / Nov / Dec>

地域差時分は、標準時刻 (GMT) との差で日本では「+0900」を使用します。

日時情報の終端にはヌル (0) を設定してください。

SMTPサーバからエラー応答を受信した場合は、メールの送信処理を異常終了します。この場合は、エラーコードとしてEV\_PROTを返し、nErrに詳細エラーコードを設定します。

エラーコードがEV\_PROTの場合は、nErrに詳細エラーコードと、pszErrStrに「エラー内容を示す文字列」を必ず格納します。

可変長メモリプールからTCPウィンドウバッファの領域 (sbufsz+rbufsz)、および $\mu$ ITRON仕様OSが使う管理領域 (詳細は、使用する $\mu$ ITRON仕様OSのマニュアルを参照してください) の合計サイズが獲得できない場合は、エラーコードとしてE\_NOMEMを返します。

なお、本サービスコールを呼び出す前には、メール送信の準備処理を行う必要があります。メール送信手順の詳細は、「4.1.1 メール送信の手順」で説明します。

### 3.1.10 SMTP\_deleteMailInfo

### SMTP 管理テーブルの返却

【T】

C言語インタフェース

```
ER ercd = SMTP_deleteMailInfo (MailInfo *pmi);
```

パラメータ

MailInfo \*pmi 送信メールの管理情報の先頭アドレス

リターンパラメータ

ER ercd エラーコードまたはリターン値

パケットの構造

送信メールの管理情報の構造に関しては「3.1.4 SMTP管理テーブルの初期化」を参照してください。

リターン値/エラーコード

E_OK	正常終了
E_PAR	パラメータエラー (pmiが0、または4の倍数以外)
E_OBJ	オブジェクト状態不正 (SMTPが起動されていない)
E_SYS	メモリブロックの返却に失敗した (各情報のアドレスが不正)

解 説

「送信メールの管理情報」、「ヘッダと宛先情報」および「送信ファイル情報」のメモリを返却します。

pmiが0または4の倍数以外の場合は、エラーコードとしてE\_PARを返します。この場合はnErrに詳細エラーコードの設定は行いません。E\_PAR発生時に必ずnErrを参照する場合は、本サービスコールを呼び出す前にnErrに0を設定してください。

pmiがSMTP\_createMailInfoサービスコールで獲得した領域でない場合は、エラーコードとしてE\_SYSを返します。また、各情報のアドレスが獲得時のアドレスと異なる場合は、領域を返却できないため、エラーコードとしてE\_SYSを返します。

他のSMTPサービスコールの処理中に本サービスコールは使用しないでください。使用した場合の動作の保証はできません。

---

## 4 SMTP クライアントの使用例

---

### 4.1 メールの送信

#### 4.1.1 メール送信の手順



図 4-1 メール送信の手順



```

------(Step 5)-----
ercd = SMTP_addRcpt(pmi, RCPT_TO, "tcp@linux.denhon.com");
if(ercd != E_OK) {
    putstring("SMTP_addRcpt error¥r¥n");
    for(;;);
}

------(Step 6)-----
ercd = SMTP_addFile(pmi, "mailbody.txt", FMODE_BODY);
if(ercd != E_OK) {
    putstring("SMTP_addFile 1 error¥r¥n");
    for(;;);
}

------(Step 7)-----
multiAtt.AttInfo[0].fileBuf = example_msg;
multiAtt.AttInfo[0].attFileSize = 99;
strcpy(multiAtt.AttInfo[0].attFileName, "add.txt");
multiAtt.NoOfAttachment = 1;

------(Step 8)-----
ercd = SMTP_addFile(pmi, multiAtt.AttInfo[0].attFileName, FMODE_ATTACH);
if(ercd != E_OK) {
    putstring("SMTP_addFile 2 error¥r¥n");
    for(;;);
}

------(Step 9)-----
multiAtt.AttInfo[1].fileBuf = (B *)bin_dat;
multiAtt.AttInfo[1].attFileSize = 9;
strcpy(multiAtt.AttInfo[1].attFileName, "add2.bin");
multiAtt.NoOfAttachment = 2;

------(Step 10)-----
ercd = SMTP_addFile(pmi, multiAtt.AttInfo[1].attFileName, FMODE_ATTACH);
if(ercd != E_OK) {
    putstring("SMTP_addFile 3 error¥r¥n");
    for(;;);
}

------(Step 11)-----
ercd = SMTP_addHeaderField(pmi, "X-Mailer:", CMAIL_VERSTR);
if(ercd != E_OK) {
    putstring("SMTP_addHeaderField error¥r¥n");
    for(;;);
}

------(Step 12)-----
bodyInf.bodySize = strlen(example_msg);
bodyInf.bodyBuf = example_msg;

------(Step 13)-----
ercd = SMTP_encode(pmi, outbuff, 0x1000, &multiAtt, &bodyInf);
if(ercd != E_OK) {
    putstring("SMTP_encode error¥r¥n");
    for(;;);
}

```

```

------(Step 14)-----
ercd = SMTP_sendMail(pmi, outbuff, "Fri,28 Jun 02 18:00:00 + 0900", 10);
if(ercd != E_OK) {
    putstring ("SMTP_sendMail error¥r¥n");
    for(;;);
}

```

```

------(Step 15)-----
ercd = SMTP_deleteMailInfo(pmi);
if(ercd != E_OK) {
    putstring("SMTP_deleteMailInfo error¥r¥n");
    for(;;);
}

```

```

------(Step 16)-----
ercd = SMTP_stop();
if(ercd != E_OK) {
    putstring("SMTP_stop¥r¥n");
    for(;;);
}

```

#### 4.1.3 プログラム例の解説

- Step 1 SMTP\_init()でSMTPを初期化します。
- Step 2 SMTP\_start()でSMTPを起動します。  
 timeintervalには1ミリ秒を指定しています。  
 mplidは1を指定し、mplszにはH'200000バイトを指定しています。
- Step 3 SMTP\_createMailInfo()でSMTP管理テーブルを初期化します。
- Step 4 「送信メールの管理情報」を設定します。  
 送信者のメールアドレスには "hicom7704@linux.denhon.com" を設定しています。  
 クライアントのホスト名には "linux.denhon.com" を設定しています。  
 件名には "Sending example" を設定しています。  
 SMTPサーバアドレスにはH'D2020302 (210.2.3.2) を設定しています。  
 <SMTPのポート番号を25以外で使用する場合は変更してください。>
- Step 5 SMTP\_addRcpt()で宛先メールアドレスを登録します。  
 宛先種別にはTOを指定しています。  
 宛先メールアドレスには "tcp@linux.denhon.com" を指定しています。  
 <宛先の数だけSMTP\_addRcpt()をコールする必要があります。>
- Step 6 SMTP\_addFile()でメール本体を登録します。  
 filemodeにはファイル種別としてFMODE\_BODYを指定します。  
 ファイル名には "mailbody.txt" を指定しています。
- Step 7 「添付ファイル情報」を設定します。  
 この領域はユーザアプリケーション側で確保する必要があります。  
 添付ファイルの先頭アドレスにはexample\_msg[ ]を設定しています。  
 添付ファイルのサイズには99を設定しています。  
 添付ファイルの名称には "add.txt" を設定しています。  
 添付ファイル実体情報の数には1を設定しています。

- Step 8 SMTP\_addFile()で添付ファイルを登録します。  
filemodeにはファイル種別としてFMODE\_ATTACHを指定します。  
ファイル名にはStep 7で設定した「添付ファイル実体情報」の添付ファイル名称を指定しています。
- Step 9 「添付ファイル情報」を設定します。  
この領域はユーザアプリケーション側で確保する必要があります。  
添付ファイルの先頭アドレスにはbin\_dat[]を設定しています。  
添付ファイルのサイズには9を設定しています。  
添付ファイルの名称には "add2.bin" を設定しています。  
添付ファイル実体情報の数を2に更新しています。
- Step 10 SMTP\_addFile()で添付ファイルを登録します。  
filemodeにはファイル種別としてFMODE\_ATTACHを指定します。  
ファイル名にはStep 9で設定した「添付ファイル実体情報」の添付ファイル名称を指定しています。
- Step 11 SMTP\_addHeaderField()でヘッダフィールドを登録します。  
フィールド名には "X-Mailer:" を設定しています。  
フィールド・ボディにはCMAIL\_VERSTRを設定しています。CMAIL\_VERSTRは、smtp03.h内で以下の文字列として定義しています。
- ```
#define DESCRIPTION "HI.CommunicationEngine Mailer"
#define VERSION_STR " Version 3.0"
#define CMAIL_VERSTR DESCRIPTION VERSION_STR
```
- Step 12 「メール本体情報」を設定します。  
この領域はユーザアプリケーション側で確保する必要があります。  
メール本体のサイズにはexample\_msgのデータ数 (99) を設定しています。  
メール本体の先頭アドレスにはexample\_msg[]を設定しています。
- Step 13 SMTP\_encode()で送信メールを生成します。  
生成したメールを格納する領域の先頭アドレスとしてoutbuff[]を指定しています。  
生成したメールを格納する領域の長さとしてH'1000を指定しています。  
「添付ファイル情報」の先頭アドレスとしてmultiAttを指定しています。  
「メール本体情報」の先頭アドレスとしてbodyInfを指定しています。
- Step 14 SMTP\_sendMail()でメールを送信します。  
送信メールを格納した領域の先頭アドレスとしてoutbuff[]を指定しています。  
日時 of 文字列として "Fri,28 Jun 02 18:00:00 + 0900" を指定しています。  
SMTPサーバからの応答待ち時間として10秒を指定しています。
- Step 15 SMTP\_deleteMailInfo()でSMTP管理テーブルを返却します。
- Step 16 SMTP\_stop()でSMTPを停止します。



#### 4.1.4 MailInfo 構造体

例題で設定されるMailInfoの内容を以下に示します。

MailInfoの構造に関しては「3.1.4 SMTP管理テーブルの初期化」を参照してください。

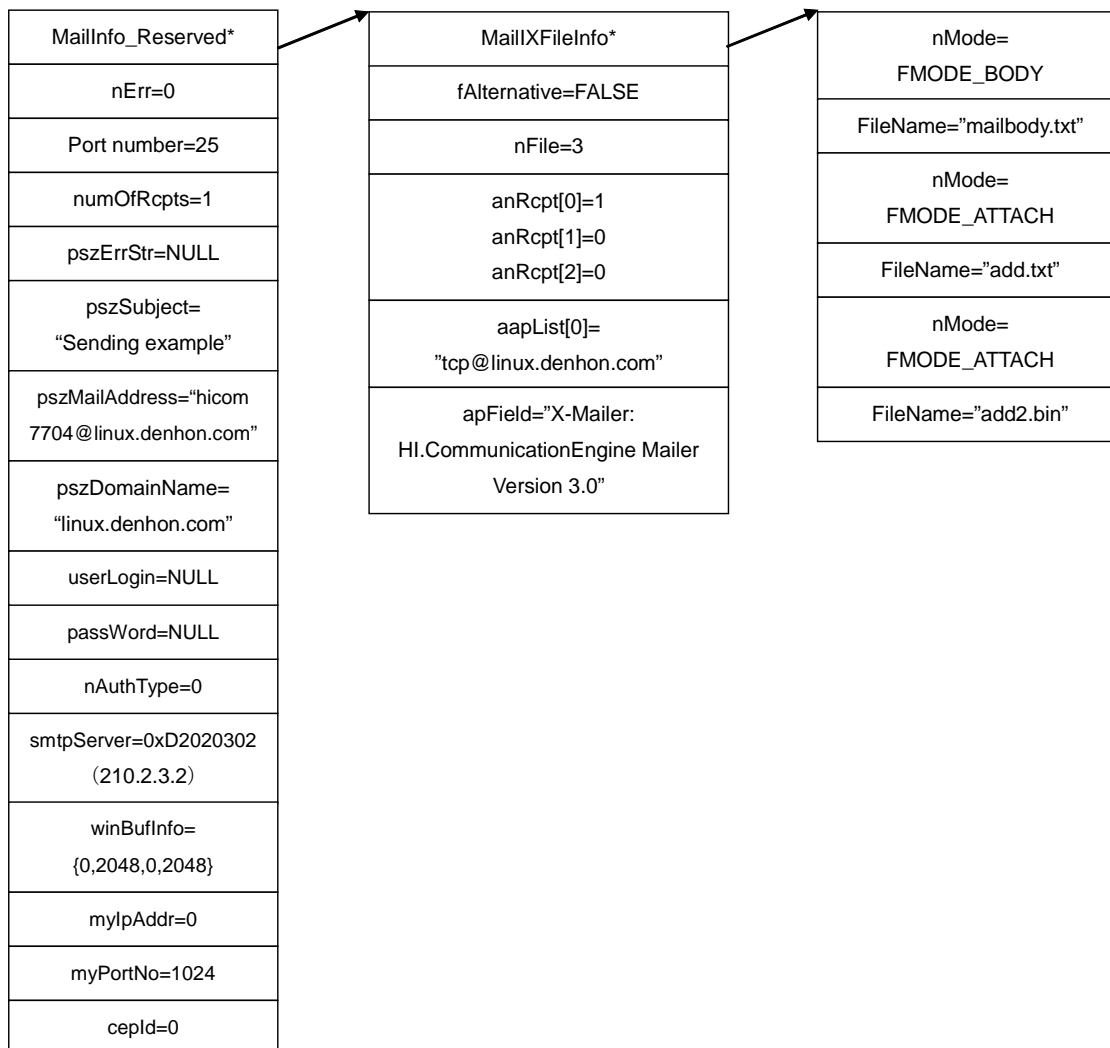


図 4-2 メール送信時のMailInfoの内容

HI.CommunicationEngine  
MAILクライアント リファレンスマニュアル SMTP編  
CE7000SMT03J-3

|      |                                  |    |     |
|------|----------------------------------|----|-----|
| 発行年月 | 2014年                            | 7月 | 第3版 |
| 発行   | ルネサスセミコンダクタパッケージ&テストソリューションズ株式会社 |    |     |
| 編集   | ルネサスセミコンダクタパッケージ&テストソリューションズ株式会社 |    |     |

©ルネサスセミコンダクタパッケージ&テストソリューションズ株式会社 2014