

CC-Link IE フィールドネットワーク

Basic サンプルコード ユーザーズマニュアル

スレーブ局編

1.01 版

改訂履歴

日付	番号	改訂内容
2016/08/01	1.00 版	作成
2016/11/01	1.01 版	誤記修正

目次

1 概要.....	6
2 用語.....	6
3 機能.....	6
4 仕様.....	7
5 アプリケーションの開発.....	10
5.1 開発環境.....	10
5.2 開発手順.....	10
5.3 サンプルコードのファイルリスト.....	11
5.4 サンプルコードの関数一覧.....	12
5.5 ユーザープログラムの作成.....	15
(1) CCIEF_BASIC_SLAVE.c.....	17
(2) SLMP_SERVER.c.....	22
(3) SOCKET.c.....	25
(4) TIMER.c.....	25
(5) USER_SAMPLE.c.....	27
5.6 関数詳細.....	29
5.6.1 戻り値の定義.....	29
5.6.2 SLMP_MakePacketStream.....	30
5.6.3 SLMP_GetSlmpInfo.....	31
5.6.4 local_itoa.....	31
5.6.5 local_atoi.....	31
5.6.6 SLMP_MakeErrorData.....	32
5.6.7 ccief_basic_slave_initialize.....	33
5.6.8 ccief_basic_slave_terminate.....	34
5.6.9 ccief_basic_slave_main.....	34
5.6.10 ccief_basic_slave_set_rx.....	35
5.6.11 ccief_basic_slave_get_ry.....	35
5.6.12 ccief_basic_slave_get_rww.....	36
5.6.13 ccief_basic_slave_set_rwr.....	36
5.6.14 ccief_basic_slave_get_pointer.....	37
5.6.15 ccief_basic_slave_set_unit_info.....	38
5.6.16 ccief_basic_slave_set_err_code.....	38
5.6.17 ccief_basic_slave_set_unit_data.....	39
5.6.18 ccief_basic_slave_get_master_info.....	39
5.6.19 ccief_basic_slave_recv_cyclic_data.....	40
5.6.20 ccief_basic_slave_send_cyclic_data.....	40
5.6.21 ccief_basic_slave_send_cyclic_data_error.....	41
5.6.22 ccief_basic_slave_disconnection.....	41
5.6.23 ccief_basic_slave_disconnection_timer_timeout.....	42

5.6.24 slmp_server_initialize.....	43
5.6.25 slmp_server_terminate.....	44
5.6.26 slmp_server_main.....	45
5.6.27 slmp_server_user_port.....	45
5.6.28 slmp_server_basic_port.....	46
5.6.29 slmp_server_paramset_port.....	46
5.6.30 slmp_server_set_status.....	47
5.6.31 slmp_server_slmp_send_response.....	48
5.6.32 slmp_server_slmp_send_err_response.....	48
5.6.33 slmp_server_service.....	49
5.6.34 slmp_server_memory_read.....	50
5.6.35 slmp_server_memory_write.....	50
5.6.36 slmp_server_node_search_basic.....	51
5.6.37 slmp_server_node_search_send_response_timeout.....	52
5.6.38 slmp_server_ip_address_set_basic.....	52
5.6.39 slmp_server_device_info_compare.....	53
5.6.40 slmp_server_parameter_get.....	53
5.6.41 slmp_server_parameter_set.....	54
5.6.42 slmp_server_parameter_set_start.....	54
5.6.43 slmp_server_parameter_set_end.....	55
5.6.44 slmp_server_parameter_set_cancel.....	55
5.6.45 slmp_server_communication_setting_get.....	56
5.6.46 slmp_server_read_type_name.....	56
5.6.47 slmp_server_remote_reset.....	57
5.6.48 socket_initialize.....	58
5.6.49 socket_terminate.....	58
5.6.50 socket_recv.....	59
5.6.51 socket_send.....	59
5.6.52 timer_initialize.....	60
5.6.53 timer_terminate.....	60
5.6.54 timer_main.....	60
5.6.55 timer_start.....	61
5.6.56 timer_stop.....	61
5.6.57 timer_get_time.....	62
5.6.58 timer_broadcast_send_wait_time.....	62
5.6.59 timer_analyze_time_data.....	63
5.6.60 main.....	64
5.6.61 user_callback_rcv_cyclic_data.....	64
5.6.62 user_callback_cyclic_disconnection.....	65
5.6.63 user_callback_set_ip_address_basic.....	65

5.6.64 user_callback_parameter_get.....	66
5.6.65 user_callback_parameter_set.....	66
5.6.66 user_callback_parameter_set_end.....	67
5.6.67 user_callback_remote_reset.....	67
5.6.68 user_parameter_file_read.....	68
5.6.69 user_parameter_file_write	69
5.6.70 user_display_cyclic_information	69
5.6.71 user_get_adapter_info	70
5.6.72 user_set_adapter_info	70
6 付録: サンプルコードのコンパイルから実行までの手順.....	71
6.1 仕様.....	71
6.2 アプリケーションの作成.....	71
6.4 アプリケーションの実行.....	72

関連資料

本書に関連する資料は、以下があります。

No	発行元	資料名称	資料番号
1	CC-Link 協会	CC-Link IE フィールドネットワーク Basic 仕様書(アプリケーション層プロトコル編)	BAP-C2010-004
2	CC-Link 協会	SLMP(Seamless Message Protocol)仕様書(概要編)	BAP-C2006-001
3	CC-Link 協会	SLMP(Seamless Message Protocol)仕様書(サービス編)	BAP-C2006-002
4	CC-Link 協会	SLMP(Seamless Message Protocol)仕様書(プロトコル編)	BAP-C2006-003

基数表記について

本書では、特に明記する場合を除き、下記に示す基数表記を使用しています。

No	基数	表記の内容	参考例
1	10 進数	数値の最後に基数を表す単位は付けません。	0
2	16 進数	数値の最初に hex を示す 0x を付けます。	0x00

データ型表記について

本書では、特に明記する場合を除き、下記に示すデータ型名を使用しています。

No	データ形名	符号	ビット数	バイト数	C 言語表記(32 ビット)
1	int8_t	有	8	1	char
2	int16_t	有	16	2	short
3	int32_t	有	32	4	int, long
4	int64_t	有	64	8	long long
5	uint8_t	無	8	1	unsigned char
6	uint16_t	無	16	2	unsigned short
7	uint32_t	無	32	4	unsigned int, unsigned long
8	uint64_t	無	64	8	unsigned long long

使用の注意事項について

本書で記述する内容についての注意事項を以下に示します。

- 添付するサンプルコードは使用例を示すものであり、その動作を保証するものではありません。
- CC-Link IE フィールドネットワーク Basic、および SLMP の用語と意味やトラブルシューティングなどについては記述していません。必要に応じて関連マニュアルを各製品ベンダーから入手し、参照してください。
- 本書およびサンプルコードの記載事項、仕様などは、お断りなしに変更することがありますので、ご了承ください。

1 概要

本書は、CC-Link IE フィールドネットワーク Basic のスレーブ局アプリケーションを開発する担当者向けのユーザーズマニュアルです。

2 用語

本マニュアルでは、特に明記する場合を除き、下記の総称・略称を使用して説明します。

総称・略称	内容
CCIEF-BASIC	CC-Link IE Field Network Basic の略称。
SLMP	Seamless Message Protocol の略称。
SLMP 情報	SLMP 通信パケットに関わる、以下の情報を含む構造体。 ネットワーク番号、ノード番号、プロセッサ番号、パケットのデータ長、コマンド、サブコマンド、データへのポインタ
マスタ局	CC-Link IE フィールドネットワーク Basic マスタ局の略称。
スレーブ局	CC-Link IE フィールドネットワーク Basic スレーブ局の略称。

3 機能

本サンプルコードは、以下の機能を提供します。

表 1 サンプルコードの機能

No.	名称	内容
1	CCIEF-BASIC スレーブ局	CCIEF-BASIC スレーブ局として、CCIEF-BASIC マスタ局とサイクリック伝送を実施します。
2	SLMP サーバー	SLMP サーバーとして、メモリ読み書き、接続機器の検出、パラメータ読み書き、リモートリセットの各サービスを実施します。

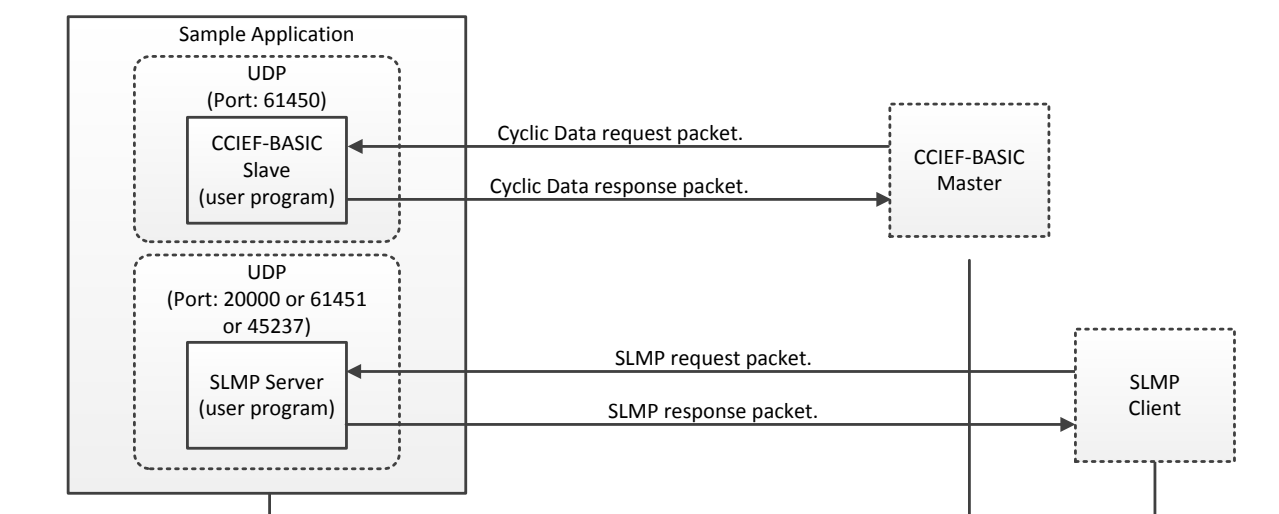


図 1. サンプルコードの機能

4 仕様

CCIEF-BASIC スレーブ局

本サンプルコードにおける CCIEF-BASIC スレーブ局の仕様を以下に示す。

表 2 CCIEF-BASIC スレーブ局の仕様

項目		内容
通信プロトコル		UDP
ポート番号		61450
IP アドレス		IPv4 クラス C: アドレス範囲 192.0.0.1～223.255.255.254 ネットワークアドレス長 24 ビット、ホストアドレス長 8 ビット (サブネットマスク 255.255.255.0)
伝文フォーマット		シリアル番号を付加しない SLMP(バイナリモード)
送信方式		ディレクテッド・ブロードキャスト(受信時)、ユニキャスト(送信時)
占有局数		1 局占有～16 局占有※1
機能		サイクリック伝送※2
サイクリックデータ	RX	64 bit(1 局占有)～最大 1024 bit(16 局占有)※1
	RY	64 bit(1 局占有)～最大 1024 bit(16 局占有)※1
	RWw	32 word(1 局占有)～最大 512 word(16 局占有)※1
	RWr	32 word(1 局占有)～最大 512 word(16 局占有)※1
パラメータ		CSV 形式(カンマ区切り)のテキストファイル。 詳細は、下記参照。

※1 パラメータの占有局数の設定値に準じる。

※2 本サンプルコードは、マスタ局から受信したサイクリックデータを折り返し送信する。

本サンプルコードは、CCIEF-BASIC スレーブ局のパラメータを以下の様に定義する。

パラメータは、CSV 形式(カンマ区切り)のテキストファイルで記述する。

表 3 本サンプルコードのパラメータ仕様

ID	パラメータ	設定範囲	内容
1	IP アドレス	192.0.0.1～223.255.255.254 (0: ネットワークアダプタの既定値)	スレーブ局の IP アドレスを設定。
2	サブネットマスク	255.255.255.0 (0: ネットワークアダプタの既定値)	スレーブ局のサブネットマスクを設定。
3	デフォルトゲート ウェイ IP アドレス	192.0.0.1～223.255.255.254 (0: ネットワークアダプタの既定値)	スレーブ局のデフォルトゲートウェイ IP アドレスを設定。
4	占有局数	1～16	スレーブ局の占有局数を設定。
5	応答遅延時間	1～4294967295 [ms] (0: 遅延時間無し)	サイクリック伝送で、スレーブ局からマスタ局への応答送信を遅延する時間を設定。

CCIEF-BASIC スレーブ局のパラメータ例を以下に示す。

IP アドレス=192.168.3.1、サブネットマスク=255.255.255.0、デフォルトゲートウェイ IP アドレス=192.168.3.254、占有局数=1、応答遅延時間=無しの場合の例

[SlaveParameter.csv]

```
''
CCIEF-BASIC Slave Sample Parameter,,
''
ID, DATA, COMMENT
1, 192. 168. 3. 1, IP Address
2, 255. 255. 255. 0, Subnet Mask
3, 192. 168. 3. 254, Default Gateway IP Address
4, 1, Occupied Station Number
5, 0, Cyclic Response Wait Time
```

IP アドレス、サブネットマスク、デフォルトゲートウェイ IP アドレスはネットワークアダプタの既定値で動作、占有局数=3、応答遅延時間 150ms の場合の例

[SlaveParameter.csv]

```
''
CCIEF-BASIC Slave Sample Parameter,,
''
ID, DATA, COMMENT
1, 0, IP Address
2, 0, Subnet Mask
3, 0, Default Gateway IP Address
4, 3, Occupied Station Number
5, 150, Cyclic Response Wait Time
```


SLMP サーバー

本サンプルコードにおける SLMP サーバーの仕様を以下に示す。

表 4 SLMP サーバーの仕様

項目	内容
通信プロトコル	UDP
ポート番号	20000※1、61451、45237
伝文フォーマット	SLMP(バイナリモード)
送信方式	リミテッド・ブロードキャスト(送受信)、またはユニキャスト(送受信)
サービス	下記参照。

※1 ユーザーの実装先の環境に応じて、変更可能。

本サンプルコードで提供する SLMP サーバーのサービスを以下に示す。

表 5 SLMP サーバーで提供するサービス

クラス名	サービス名	ポート 番号	コマンド	サブ コマンド	内容
Memory	Memory Read	20000	0613	0000	内部メモリ※1を読み出す。
	Memory Write	20000	1613	0000	内部メモリ※1に書き込む。
NodeConnect	NodeSearch	61451	0E30	0000	ネットワーク内のサーバーの検出に応答する。
	IPAddressSet	61451	0E31	0000	IP アドレスなどの通信に必要なネットワーク情報を設定する。※2
Parameter Setting	DeviceInfoCompare	20000	0E32	0000	相手機器を確認する。
	ParameterGet	20000	0E33	0000	パラメータ※3の値を読み出す。
	ParameterSet	20000	0E34	0000	パラメータ※3の値を書き込む。
	ParameterSetStart	20000	0E35	0000	パラメータの書き込み排他処理を開始する。
	ParameterSetEnd	20000	0E36	0000	パラメータの書き込み排他処理を終了する。
	ParameterSetCancel	20000	0E3A	0000	パラメータの書き込み排他処理をキャンセルする。
NodeMonitoring	CommunicationSettingGet	45237	0E45	0000	通信設定を読み出す。
Remote Control	Read Type Name	20000	0101	0000	ユニットの形名および形名コードを読み出す。
	Remote Reset	20000	1006	0000	アプリケーションの実行をリセットする。

※1 サンプルコードでは、10k ワードのメモリ空間をサポートする。

※2 サンプルコードでは、ネットワークアダプタの IP アドレス、およびサブネットマスクの設定をサポートする。

※3 サンプルコードで定義している CCIEF-BASIC スレーブ局のパラメータをサポートする。

5 アプリケーションの開発

5.1 開発環境

本書に付属するサンプルコードは「gcc(GNU Compiler Collection)」でコンパイルエラーが出ないことを確認している。6 章に、付録として gcc を使用したサンプルコードのコンパイルから実行までの手順を示している。

5.2 開発手順

添付のサンプルコードを使用してアプリケーションを開発する際の手順を示す。

サンプルコードは表 6 に示すプログラム部品で構成する。ユーザーは、SLMP ライブラリを実装先の環境に応じて変更する。また、用途に合わせてユーザープログラムの内容を変更する。

表 6 サンプルコード構成

No.	プログラム部品名	概要
1	SLMP ライブラリ	SLMP 通信パケットの生成と、パケットからの SLMP 情報取得を行う関数。実装先の環境に応じてプログラムを変更する。
2	ユーザープログラム	機器の機能を実装するためのアプリケーションプログラム。一例として Linux のソケット通信を使用して CC-Link IE フィールドネットワーク Basic のスレーブ局としてサイクリック通信を行うサンプルコード、および SLMP サーバーとして SLMP 通信パケットを送受信するサンプルコードを記述している。環境に応じてプログラムを変更する。

アプリケーション開発手順を以下に示す。

- (1) ユーザープログラムの作成(CCIEF_BASIC_SLAVE.c、CCIEF_BASIC_SLAVE.h、SLMP_SERVER.c、SLMP_SERVER.h、SOCKET.c、SOCKET.h、TIMER.c、TIMER.h、USER_SAMPLE.c、USER_SAMPLE.h)
ユーザーは、ユーザープログラムを作成する。詳細については、5.5 を参照。
- (2) SLMP ライブラリの作成(SLMP.c、SLMP.h)
添付のサンプルコードに含まれる SLMP ライブラリのソースコードをコンパイル後、ライブラリアンを実行し、ライブラリファイルを作成する。
- (3) ユーザープログラムとライブラリファイルの結合
ユーザープログラム、ライブラリファイルを結合し、ロードモジュールファイルを作成する。

5.3 サンプルコードのファイルリスト

サンプルコードのディレクトリ構成を以下に示す。

root.	+ library	+ include	…SLMP ライブラリヘッダファイル
		+ src	…SLMP ライブラリコードファイル
	+ sample	+ include	…ユーザープログラムヘッダファイル
		+ src	…ユーザープログラムコードファイル

サンプルコードのファイルリストを以下に示す。

No.	フォルダ名			ファイル名	説明
1	Root			version.txt	バージョン情報
2				readme.txt	ヘルプファイル
3		library	include	SLMP.h	SLMP ライブラリヘッダ
4			src	SLMP.c	SLMP ライブラリ関数
5		sample	include	CCIEF_BASIC_SLAVE.h	ユーザープログラムヘッダ(CCIEF-BASIC スレーブ局)
6				SLMP_SERVER.h	ユーザープログラムヘッダ(SLMP サーバー)
7				SOCKET.h	ユーザープログラムヘッダ(ソケット)
8				TIMER.h	ユーザープログラムヘッダ(タイマー)
9				USER_SAMPLE.h	ユーザープログラムヘッダ
10			src	CCIEF_BASIC_SLAVE.c	ユーザープログラム(CCIEF-BASIC スレーブ局)
11				SLMP_SERVER.c	ユーザープログラム(SLMP サーバー)
12				SOCKET.c	ユーザープログラム(ソケット)
13				TIMER.c	ユーザープログラム(タイマー)
14				USER_SAMPLE.c	ユーザープログラム

5.4 サンプルコードの関数一覧

サンプルコードに含まれる関数一覧を表 7 に示す。

表 7 サンプルコード関数一覧

No.	プログラム部品名	ファイル名	関数名	関数型	概要	公開/ 非公開
1	SLMP ライ ブラリ	SLMP.c	SLMP_MakePacketStream	int	SLMP 通信パケット作成	公開
2			SLMP_GetSlmpInfo	int	SLMP 情報取得	公開
3			local_itoa	uint8_t	数値から ASCII への変換	公開
4			local_atoi	uint8_t	ASCII から数値への変換	公開
5			SLMP_MakeErrorData	int	SLMP エラー応答データ作成	公開
6	ユーザ プログラ ム	CCIEF_BA SIC_SLAV E.c	ccief_basic_slave_initialize	int	CCIEF-BASIC スレーブ局初 期化	公開
7			ccief_basic_slave_terminate	void	CCIEF-BASIC スレーブ局終 了	公開
8			ccief_basic_slave_main	int	CCIEF-BASIC スレーブ局メ イン処理	公開
9			ccief_basic_slave_set_rx	int	RX データセット	公開
10			ccief_basic_slave_get_ry	int	RY データ取得	公開
11			ccief_basic_slave_get_rww	int	RWw データ取得	公開
12			ccief_basic_slave_set_rwr	int	RWr データセット	公開
13			ccief_basic_slave_get_pointer	uint16_t *	デバイス先頭ポインタ取得	公開
14			ccief_basic_slave_set_unit_info	void	自局ユニット情報セット	公開
15			ccief_basic_slave_set_err_code	void	エラーコードセット	公開
16			ccief_basic_slave_set_unit_data	void	自局管理情報セット	公開
17			ccief_basic_slave_get_master_info	void	マスタ局情報取得	公開
18			ccief_basic_slave_recv_cyclic_data	int	サイクリックデータ受信	非公開
19			ccief_basic_slave_send_cyclic_data	int	サイクリックデータ送信	非公開
20			ccief_basic_slave_send_cyclic_data_ error	int	サイクリックエラーデータ送 信	非公開
21			ccief_basic_slave_disconnection	void	解列処理	非公開
22			ccief_basic_slave_disconnection_tim er_timeout	void	解列検出時間タイムアウト (コールバック関数)	非公開
23		SLMP_SE RVER.c	slmp_server_initialize	int	SLMP サーバー初期化	公開
24			slmp_server_terminate	void	SLMP サーバー終了	公開
25			slmp_server_main	int	SLMP サーバーメイン処理	公開
26			slmp_server_user_port	int	SLMP サーバーユーザー指 定ポート受信処理	公開

27			slmp_server_basic_port	int	SLMP サーバー CCIEF-BASIC 用 NodeConnect ポート受信処理	公開
28			slmp_server_paramset_port	int	SLMP サーバーパラメータ設定用ポート受信処理	公開
29			slmp_server_set_status	void	SLMP サーバー状態セット	公開
30			slmp_server_slmp_send_response	int	SLMP 応答送信	非公開
31			slmp_server_slmp_send_err_response	int	SLMP エラー応答送信	非公開
32			slmp_server_service	int	SLMP サービス実行	公開
33			slmp_server_memory_read	int	メモリ読み出し	公開
34			slmp_server_memory_write	int	メモリ書き込み	公開
35			slmp_server_node_search_basic	int	自動検出(CCIEF-BASIC 用)	公開
36			slmp_server_node_search_send_response_timeout	void	自動検出応答送信待ちタイムアウト (コールバック関数)	非公開
37			slmp_server_ip_address_set_basic	int	通信設定(CCIEF-BASIC 用)	公開
38			slmp_server_device_info_compare	int	機器接続情報確認	公開
39			slmp_server_parameter_get	int	パラメータ読み出し	公開
40			slmp_server_parameter_set	int	パラメータ書き込み	公開
41			slmp_server_parameter_set_start	int	パラメータ書き込み排他処理開始	公開
42			slmp_server_parameter_set_end	int	パラメータ書き込み排他処理終了	公開
43			slmp_server_parameter_set_cancel	int	パラメータ書き込み排他処理キャンセル	公開
44			slmp_server_communication_setting_get	int	通信設定取得	公開
45			slmp_server_read_type_name	int	形名読み出し	公開
46			slmp_server_remote_reset	int	リモートリセット	公開
47		SOCKET.c	socket_initialize	int	ソケット初期化	公開
48			socket_terminate	void	ソケット終了	公開
49			socket_recv	int	パケット受信	公開
50			socket_send	int	パケット送信	公開
51		TIMER.c	timer_initialize	void	タイマー初期化	公開
52			timer_terminate	void	タイマー終了	公開
53			timer_main	void	タイマーメイン処理	公開
54			timer_start	int	タイマー開始	公開

55			timer_stop	void	タイマー停止	公開
56			timer_get_time	uint32_t	現在時間取得	公開
57			timer_broadcast_send_wait_time	uint32_t	ブロードキャスト送信待ち時間取得	公開
58			timer_analyze_time_data	void	時計情報解析	公開
59		USER_SA	main	void	メイン処理	公開
60		MPLE.c	user_callback_recv_cyclic_data	void	サイクリックデータ受信 (コールバック関数)	公開
61			user_callback_cyclic_disconnection	void	解列検出 (コールバック関数)	公開
62			user_callback_set_ip_address_basic	void	通信設定(CCIEF-BASIC 用) (コールバック関数)	公開
63			user_callback_parameter_get	int	パラメータ読み出し (コールバック関数)	公開
64			user_callback_parameter_set	int	パラメータ書き込み (コールバック関数)	公開
65			user_callback_parameter_set_end	int	パラメータ書き込み完了 (コールバック関数)	公開
66			user_callback_remote_reset	int	リモートリセット要求 (コールバック関数)	公開
67			user_parameter_file_read	int	パラメータファイル読み出し	非公開
68			user_parameter_file_write	int	パラメータファイル書き込み	非公開
69			user_display_cyclic_information	void	サイクリック情報表示	非公開
70			user_get_adapter_info	int	ネットワークアダプタ情報取得	非公開
71			user_set_adapter_info	int	ネットワークアダプタ情報設定	非公開

公開: 外部に公開する関数を示す。非公開: ローカルファイル内で使用する関数を示す。

5.5 ユーザープログラムの作成

ユーザーは実装先の環境に応じて、ユーザープログラムを作成する。

ユーザープログラムのファイル一覧を以下に示す。

表 8 ユーザープログラムのファイル一覧

No.	ファイル名	内容
1	CCIEF_BASIC_SLAVE.c	CCIEF-BASIC マスタ局とサイクリック通信を実施する。
2	SLMP_SERVER.c	SLMP クライアントからの SLMP 要求パケットを受信後、各サービス処理を実行し、SLMP 応答パケットを SLMP クライアントへ送信する。
3	SOCKET.c	ソケット処理を実施する為の関数群を提供する。
4	TIMER.c	タイマー処理を実施する為のライブラリを提供する。
5	USER_SAMPLE.c	CCIEF-BASIC スレーブ局、および SLMP サーバーの初期化とメイン処理の実行、およびパラメータファイルの読み書きを実施する。 コールバック関数を実装し、CCIEF-BASIC マスタ局とのサイクリックデータ処理、および SLMP サーバーの各サービス処理を実施する。

ユーザーの実装環境において、OS およびプロトコルスタックの相違によるプログラムの変更が必要な箇所を以下に示す。

表 9 OS およびプロトコルスタックの相違によるプログラムの変更が必要な箇所

No.	ファイル名	関数名	変更点
1	CCIEF_BASIC_SLAVE.c	ccief_basic_slave_main	ソケット関数、構造体の実装方法。
2		ccief_basic_slave_recv_cyclic_data	ソケット関数、構造体の実装方法。
3		ccief_basic_slave_send_cyclic_data	ソケット関数、構造体の実装方法。
4	SLMP_SERVER.c	slmp_server_user_port	ソケット関数、構造体の実装方法。
5		slmp_server_basic_port	ソケット関数、構造体の実装方法。
6		slmp_server_paramset_port	ソケット関数、構造体の実装方法。
7		slmp_server_slmp_send_response	ソケット関数、構造体の実装方法。
8		slmp_server_slmp_send_err_response	ソケット関数、構造体の実装方法。
9		slmp_server_service	ソケット関数、構造体の実装方法。
10		slmp_server_memory_read	ソケット関数、構造体の実装方法。
11		slmp_server_memory_write	ソケット関数、構造体の実装方法。
12		slmp_server_node_search_basic	ソケット関数、構造体の実装方法。
13		slmp_server_node_search_send_response_timeout	ソケット関数、構造体の実装方法。
14		slmp_server_ip_address_set_basic_basic	ソケット関数、構造体の実装方法。
15		slmp_server_device_info_compare	ソケット関数、構造体の実装方法。
16		slmp_server_parameter_get	ソケット関数、構造体の実装方法。
17		slmp_server_parameter_set	ソケット関数、構造体の実装方法。
18		slmp_server_parameter_set_start	ソケット関数、構造体の実装方法。
19		slmp_server_parameter_set_end	ソケット関数、構造体の実装方法。

20		slmp_server_parameter_set_cancel	ソケット関数、構造体の実装方法。
21		slmp_server_communication_setting_get	ソケット関数、構造体の実装方法。
22		slmp_server_remote_reset	ソケット関数、構造体の実装方法。
23	SOCKET.c	socket_initialize	ソケット関数、構造体の実装方法。※1
24		socket_terminate	ソケット関数、構造体の実装方法。
25		socket_recv	ソケット関数、構造体の実装方法。
26		socket_send	ソケット関数、構造体の実装方法。
27	TIMER.c	timer_get_time	システム時間の取得方法。
28		timer_broadcast_send_wait_time	システム時間の取得方法。
29	USER_SAMPLE.c	main	ソケット関数、構造体の実装方法。
30		user_callback_parameter_get	ソケット関数、構造体の実装方法。
31		user_callback_parameter_set	ソケット関数、構造体の実装方法。
32		user_parameter_file_read	ソケット関数、構造体の実装方法。
33		user_parameter_file_write	ソケット関数、構造体の実装方法。
34		user_get_adapter_info	ネットワークアダプタ情報の取得方法。
35		user_set_adapter_info	ネットワークアダプタ情報の設定方法。

※1 ノンブロッキングモード設定が必要。

ユーザープログラム作成におけるポイントについて、各ファイルごとに下記に説明する。

(1) CCIEF_BASIC_SLAVE.c

CCIEF-BASIC マスタ局とサイクリック通信を実施する。

① サイクリックデータ

サンプルコードでは、内部にサイクリックデータ(RX、RY、RWw、RWr)を定義する。ユーザープログラムはサイクリックデータに任意のタイミングでアクセスすることで、マスタ局とサイクリック通信を実現する。サイクリックデータのアクセス方法には、サイクリックデータのデバイスに直接アクセスする方法と、各デバイスの先頭ポインタを取得し、そのポインタでアクセスする方法を定義している。

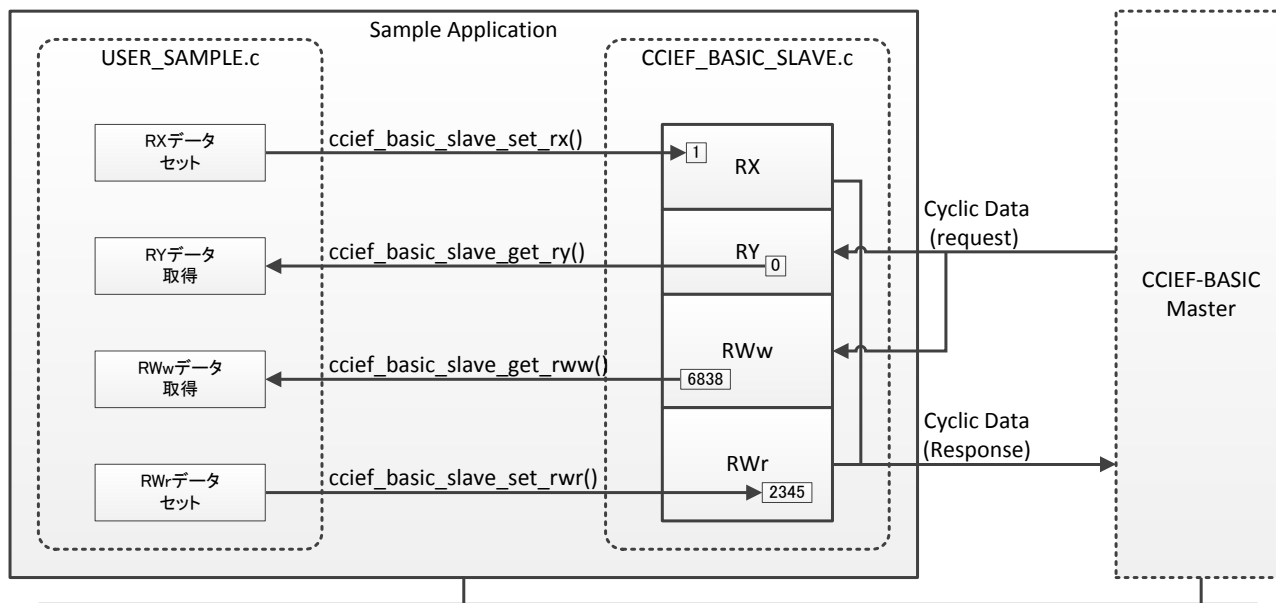


図 2. サイクリックデータのデバイスに直接アクセス

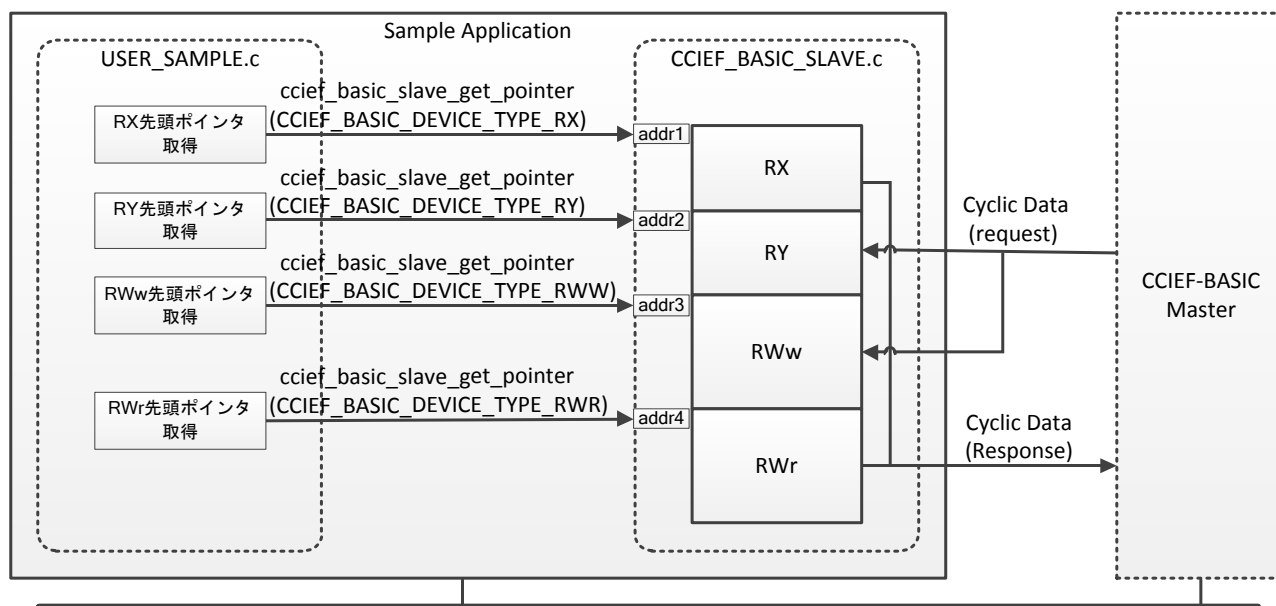


図 3. サイクリックデータの各デバイスの先頭ポインタを取得

② コールバック関数

サンプルコードは、コールバック関数を定義し、以下に示すタイミングで指定されたコールバック関数を実行する。

ユーザーはコールバック関数を実装することで、機能を簡単に開発できる。

表 10 コールバック関数一覧

No.	コールバック関数名	実行タイミング
1	CCIEF_BASIC_SLAVE_CALLBACK_RECV_CYCLIC_DATA	CCIEF-BASIC マスタ局からサイクリック要求データを受信した時。
2	CCIEF_BASIC_SLAVE_CALLBACK_CYCLIC_DISCONNECTION	CCIEF-BASIC マスタ局からの解列、およびサイクリック停止を検出した時。

[CCIEF_BASIC_SLAVE.h]

```
typedef void(*CCIEF_BASIC_SLAVE_CALLBACK_RECV_CYCLIC_DATA)(int iCyclicState, int iOccupiedStationNumber);
typedef void(*CCIEF_BASIC_SLAVE_CALLBACK_CYCLIC_DISCONNECTION)(void);
```

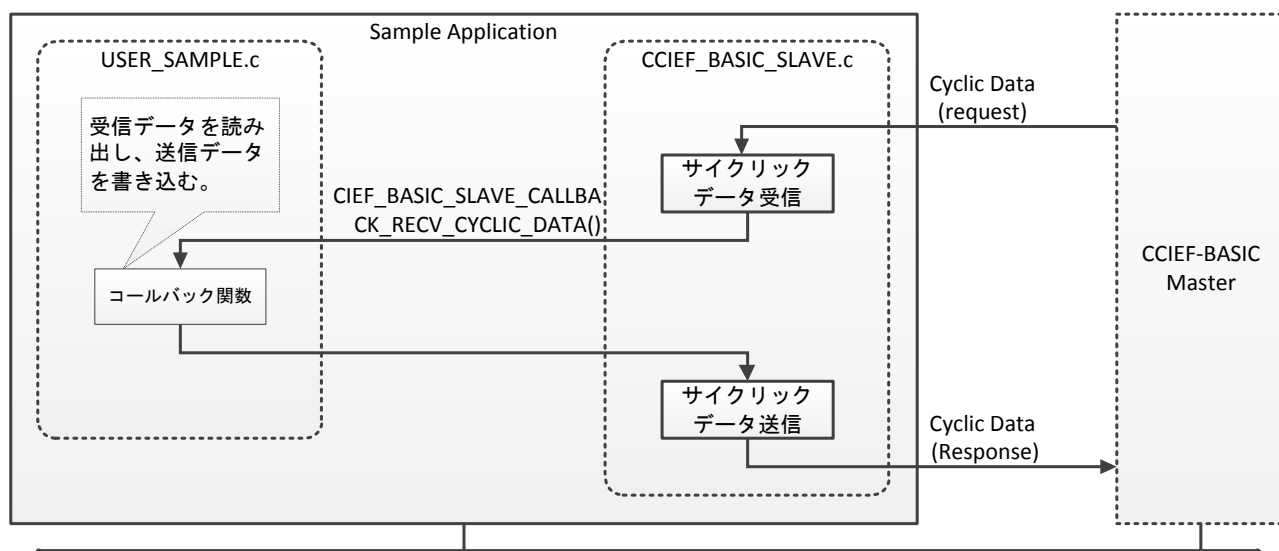


図 4. コールバック関数のイメージ

③ スレーブ局通知情報設定

サンプルコードは、サイクリック伝送の応答送信でマスタ局へ通知する情報をユーザープログラムで設定できる。

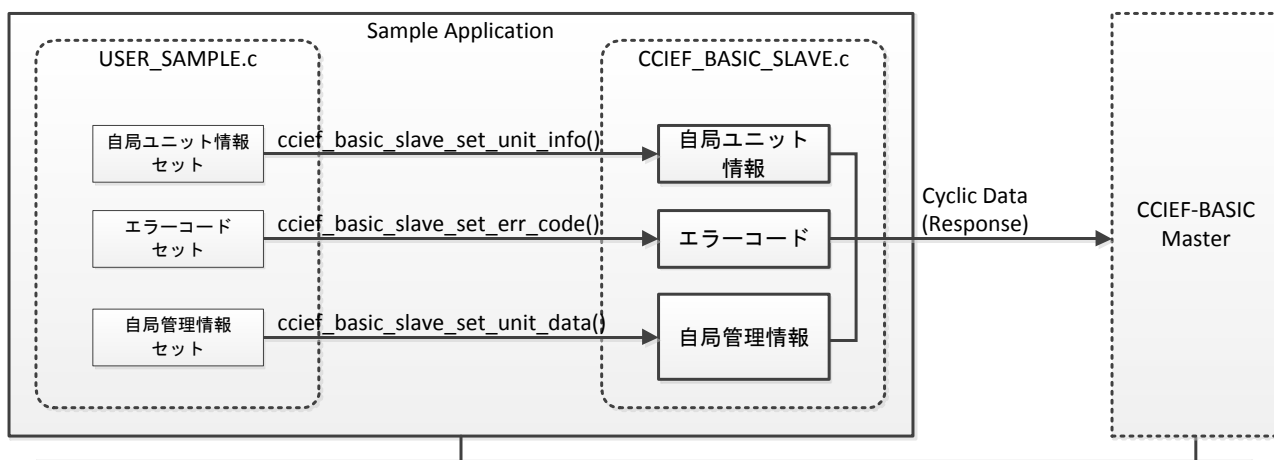


図 5. スレーブ局通知情報を設定

④ マスタ局情報取得

サンプルコードは、サイクリック伝送データで受信したマスタ局の情報をユーザープログラムで取得できる。

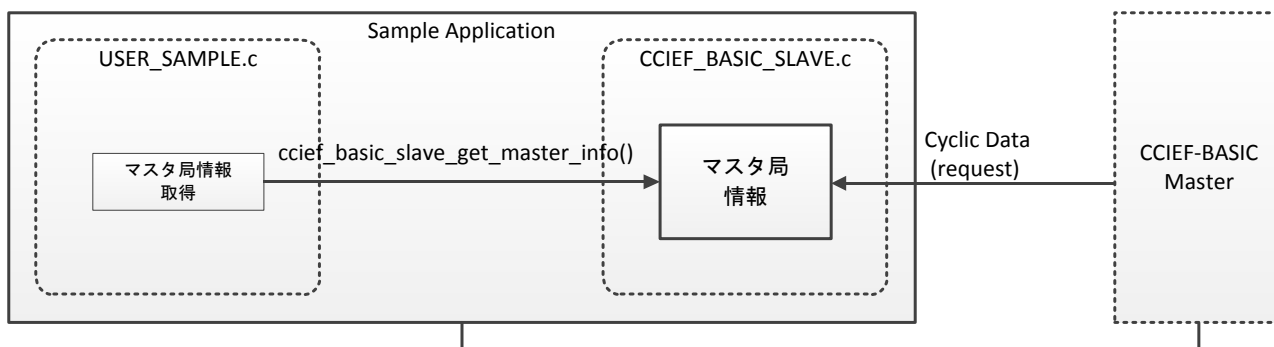


図 6. マスタ局情報を取得

⑤ フローチャート

サンプルプログラムのフローチャートを以下に示す。

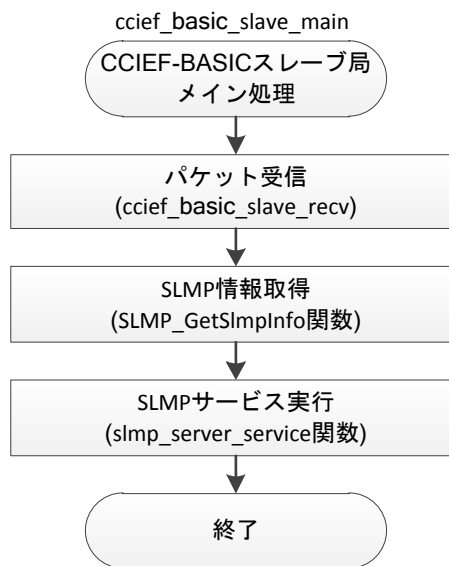


図 7. CCIEF_BASIC_SLAVE.c のフローチャート 1(サンプルプログラムの例)

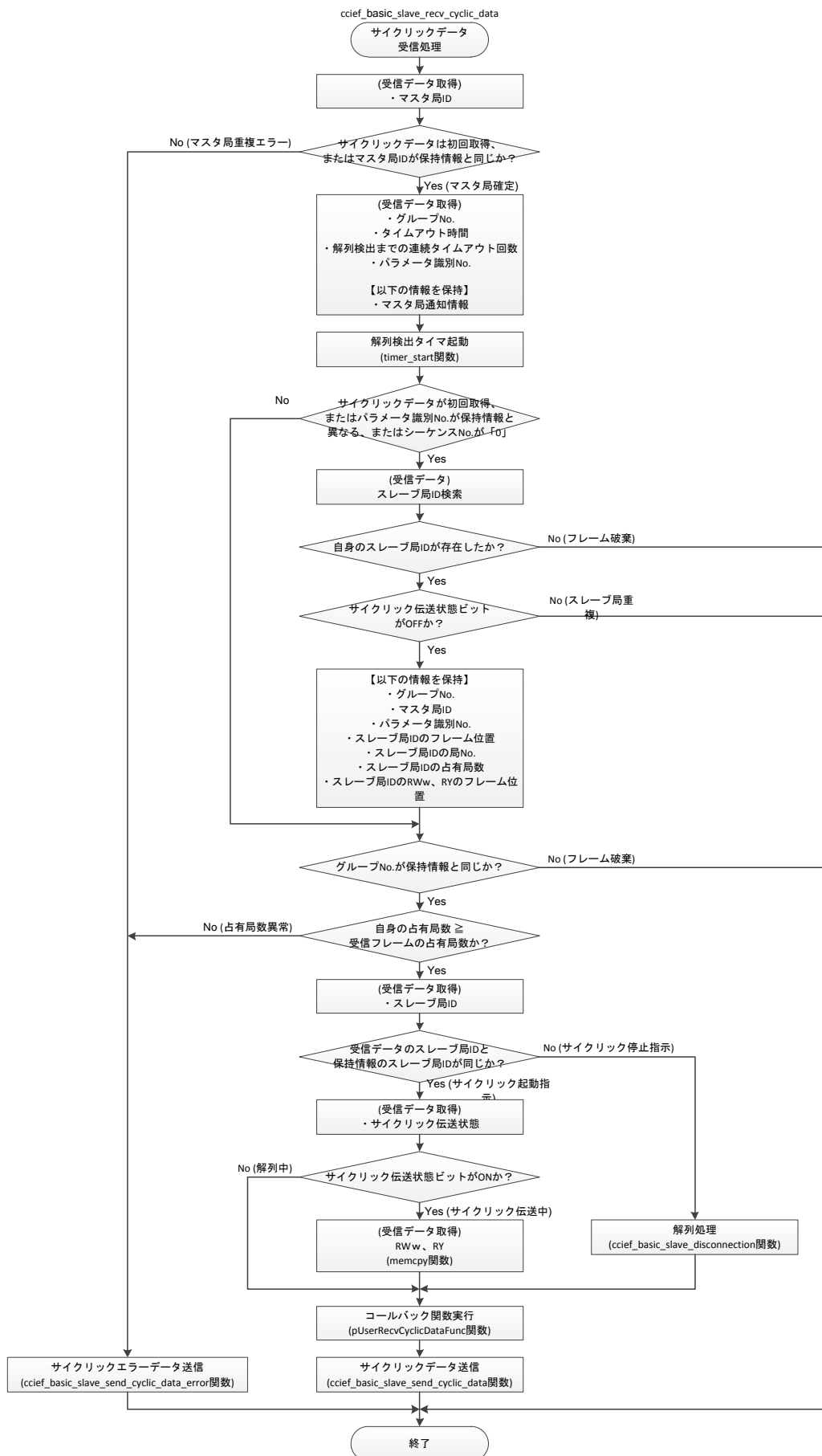


図 8. CCIEF_BASIC_SLAVE.c のフローチャート 2(サンプルプログラムの例)

(2) SLMP_SERVER.c

SLMP クライアントからの SLMP 要求パケットを受信後、各サービス処理を実行し、SLMP 応答パケットを SLMP クライアントへ送信する。

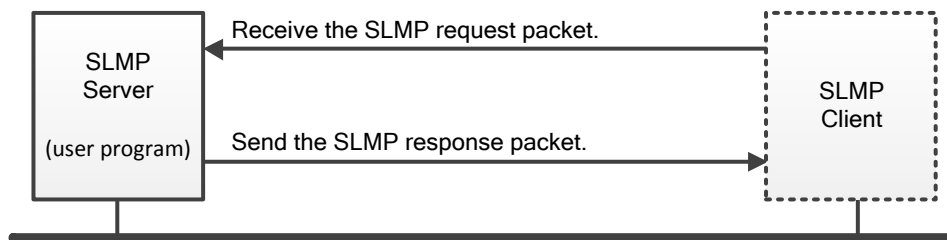


図 9. SLMP サーバー

① コールバック関数

サンプルコードでは、以下に示すコールバック関数を定義する。

ユーザーはコールバック関数を実装することで、機能を簡単に開発できる。

表 11 コールバック関数一覧

No.	コールバック関数名	実行タイミング
1	SLMP_SERVER_CALLBACK_IPADDRESS_SET_BASIC	SLMP サーバーが通信設定のコマンド要求(0E31)を受信した時。
2	SLMP_SERVER_CALLBACK_PARAMETER_GET	SLMP サーバーがパラメータ読み出しのコマンド要求(0E33)を受信した時。
3	SLMP_SERVER_CALLBACK_PARAMETER_SET	SLMP サーバーがパラメータ書き込みのコマンド要求(0E34)を受信した時。
4	SLMP_SERVER_CALLBACK_PARAMETER_SET_END	SLMP サーバーがパラメータ書き込み完了のコマンド要求(0E36)を受信した時。
5	SLMP_SERVER_CALLBACK_REMOTE_RESET	SLMP サーバーがリモートリセットのコマンド要求(1006)を受信した時。

[SLMP_SERVER.h]

```

typedef void(*SLMP_SERVER_CALLBACK_IPADDRESS_SET_BASIC)(uint32_t ulIpAddress, uint32_t ulSubnetMask );
typedef int(*SLMP_SERVER_CALLBACK_PARAMETER_GET)(uint16_t usId, uint16_t *pusSize, uint8_t **ppucData);
typedef int(*SLMP_SERVER_CALLBACK_PARAMETER_SET)(uint16_t usId, uint16_t usSize, uint8_t *pucData);
typedef int(*SLMP_SERVER_CALLBACK_PARAMETER_SET_END)(void);
typedef int(*SLMP_SERVER_CALLBACK_REMOTE_RESET)(void);
  
```

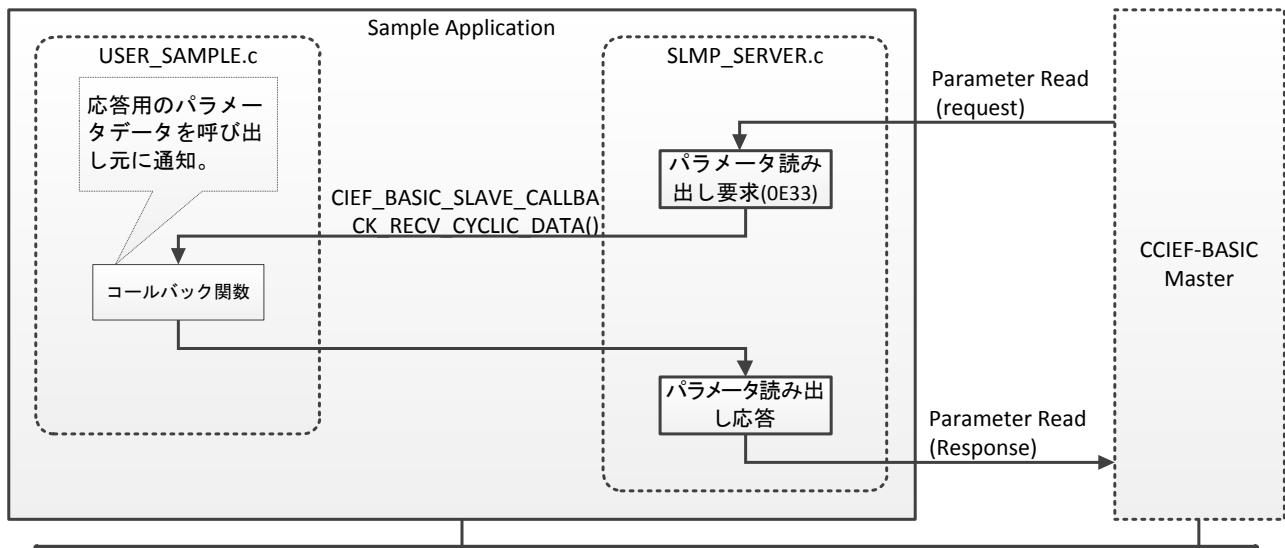


図 10. コールバック関数のイメージ

② フローチャート

サンプルプログラムのフローチャートを以下に示す。

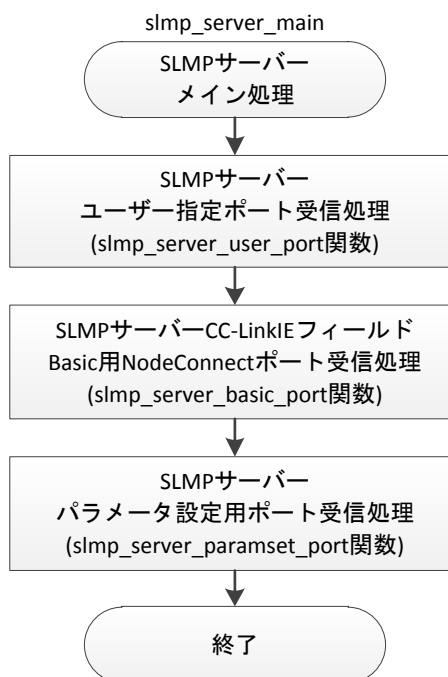


図 11. SLMP_SERVER.c のフローチャート 1(サンプルプログラムの例)

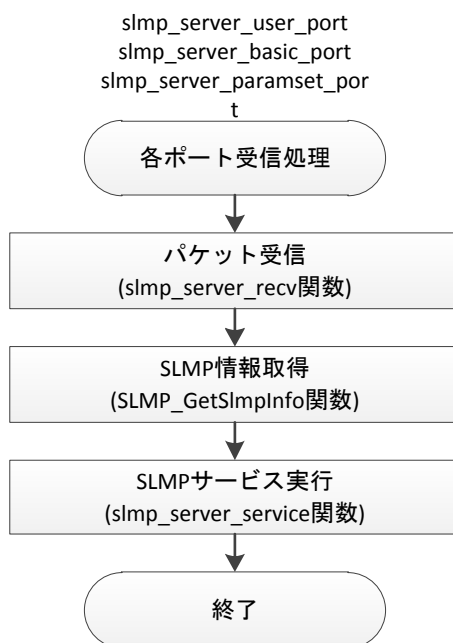


図 12. SLMP_SERVER.c のフローチャート 2(サンプルプログラムの例)

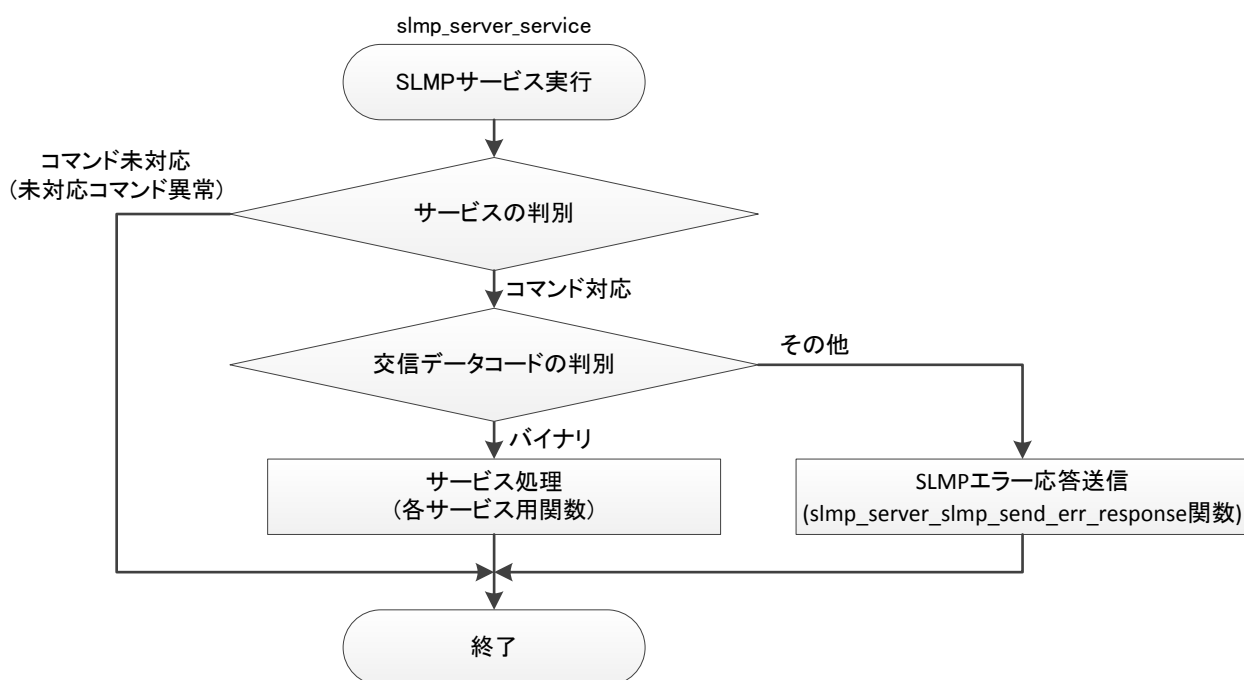


図 13. SLMP_SERVER.c のフローチャート 3(サンプルプログラムの例)

(3) SOCKET.c

ソケット処理を実施する為の関数群を提供する。

※実装先の環境に応じて変更する。

(4) TIMER.c

タイマー処理を実施する為のライブラリを提供する。

※経過時間(プロセッサ時間)の取得方法等、実装先の環境に応じて変更する。

① コールバック関数

サンプルコードでは、以下に示すコールバック関数を定義する。コールバック関数は、登録されたタイマーがタイムアウトした時に実行される。

[TIMER.h]

```
typedef void (*TIMER_CALLBACK)( int iId, void *pCallbackArg );
```

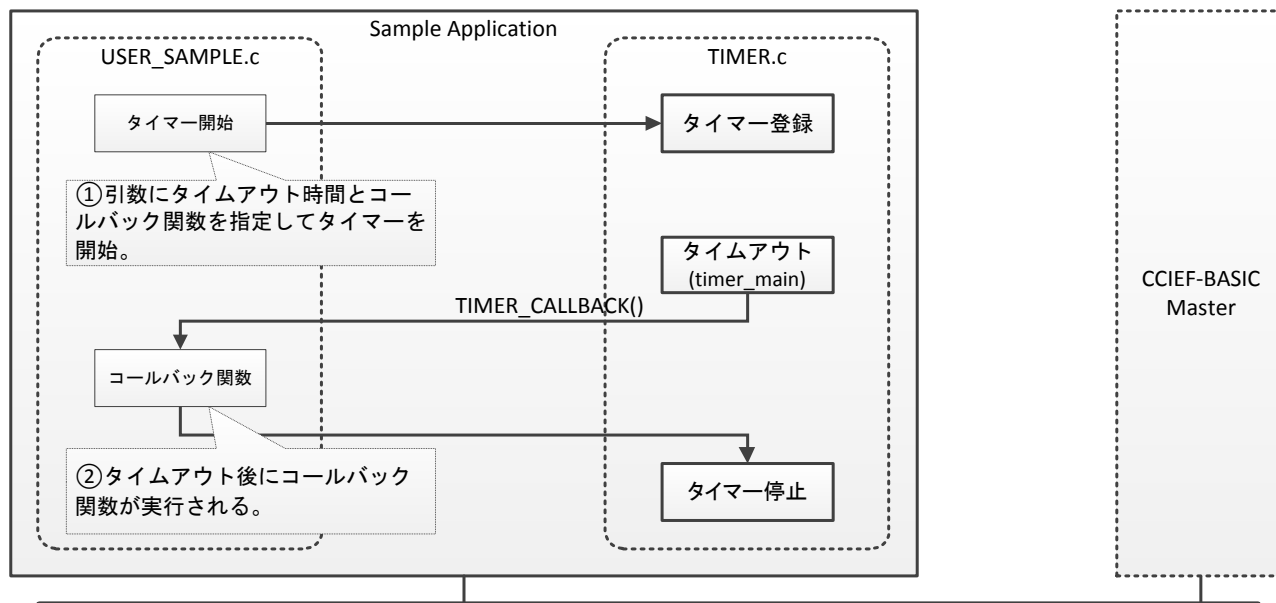


図 14. コールバック関数のイメージ

② フローチャート

サンプルプログラムのフローチャートを以下に示す。

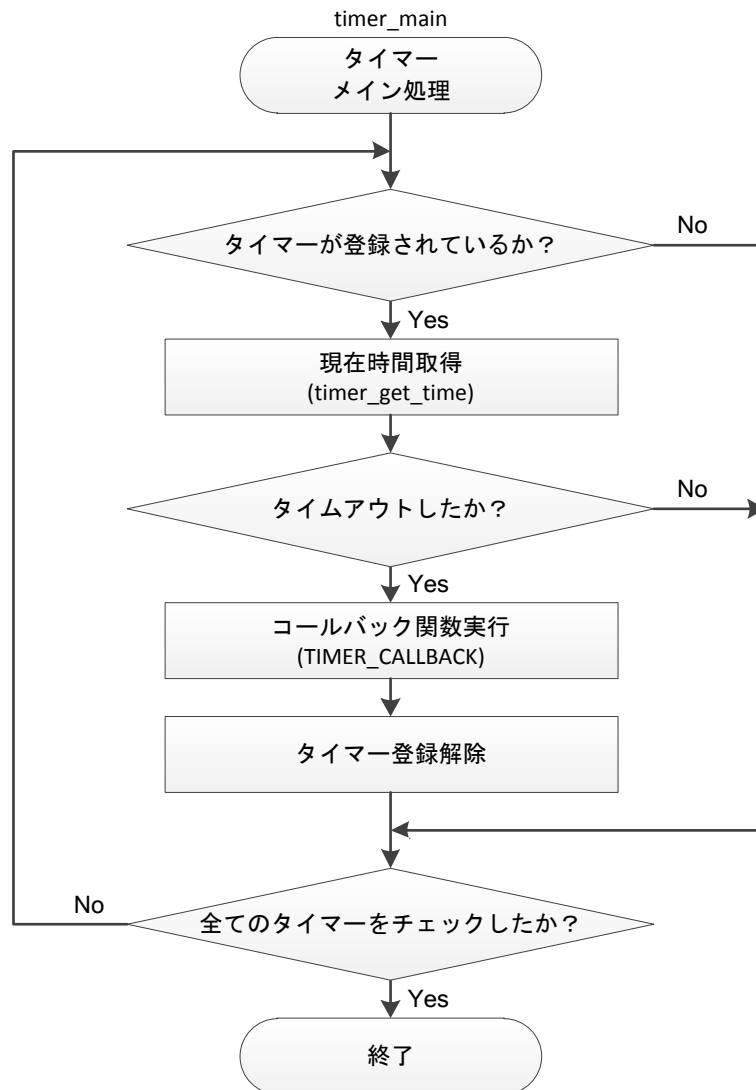


図 15. TIMER.c のフローチャート(サンプルプログラムの例)

(5) USER_SAMPLE.c

CCIEF-BASIC スレーブ局、および SLMP サーバーの初期化とメイン処理、およびパラメータファイルの読み書きを実施する。

コールバック関数を実装し、CCIEF-BASIC マスタ局とのサイクリックデータ処理、および SLMP サーバーの各サービス処理を実施する。

① プログラム実装

サンプルプログラムは、CCIEF_BASIC_SLAVE.c、および SLMP_SERVER.c が提供するコールバック関数を実装する。サンプルプログラムの実装内容を以下に示す。

※ユーザーの実装先の環境に応じて、変更する。

表 12 サンプルプログラムの実装内容

No.	プログラム名	実装内容	実装元コールバック関数名
1	user_callback_recv_cyclic_data	スレーブ局の RWw、RY のデータを、RWr、RX に折り返しセットする。	CCIEF_BASIC_SLAVE_CALLBACK_RECV_CYCLIC_DATA (表 10 参照)
2	user_callback_cyclic_disconnection	自局ユニット情報にアプリケーション停止中をセットする。	CCIEF_BASIC_SLAVE_CALLBACK_CYCLIC_DISCONNECTION (表 10 参照)
3	user_callback_set_ip_address_basic	通信設定を反映し、リセットを実行する。 (CCIEF-BASIC 用)	SLMP_SERVER_CALLBACK_IPADDRESS_SET_BASIC (表 11 参照)
4	user_callback_parameter_get	スレーブ局のパラメータを応答する。	SLMP_SERVER_CALLBACK_PARAMETER_GET (表 11 参照)
5	user_callback_parameter_set	スレーブ局のパラメータに反映する。	SLMP_SERVER_CALLBACK_PARAMETER_SET (表 11 参照)
6	user_callback_parameter_set_end	パラメータをファイルに書き込む。	SLMP_SERVER_CALLBACK_PARAMETER_SET_END (表 11 参照)
7	user_callback_remote_reset	リセットを実行する。	SLMP_SERVER_CALLBACK_REMOTE_RESET (表 11 参照)

② フローチャート

サンプルプログラムのフローチャートを以下に示す。

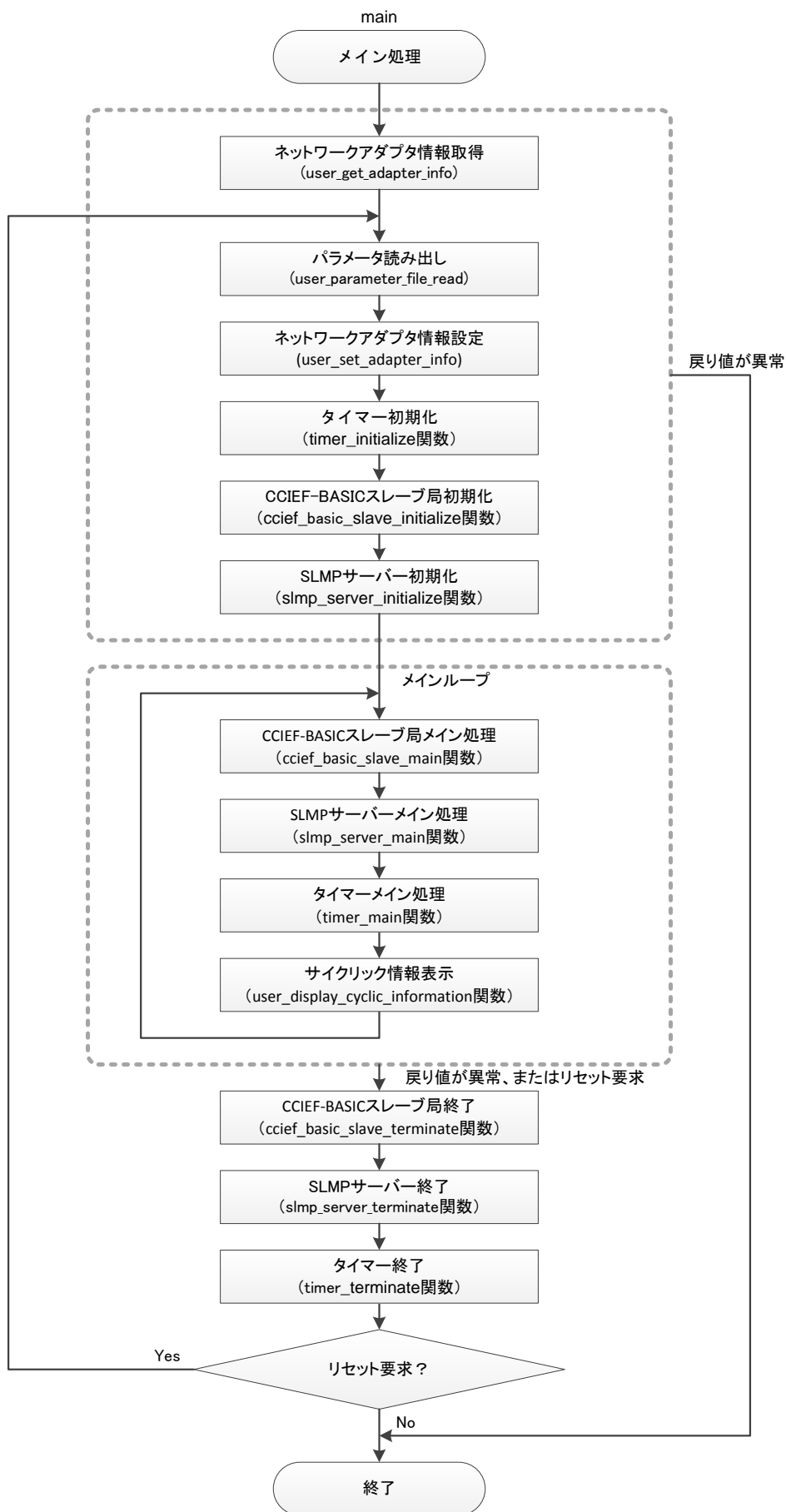


図 16. USER_SAMPLE.c のフローチャート(サンプルプログラムの例)

5.6 関数詳細

5.6.1 戻り値の定義

SLMP ライブラリ内の関数戻り値として返すエラーコード及び終了コードは以下を使用する。

[SLMP.h]

#define	SLMP_ERR_OK	0
#define	SLMP_ERR_NG	(-1)
#define	SLMP_ERR_COMMAND_SUBCOMMAND	(0xC059)
#define	SLMP_ERR_WRONG_DATA	(0xC05C)
#define	SLMP_ERR_DATA_LENGTH	(0xC061)
#define	SLMP_ERR_UNDER_EXECUTION	(0xC0EE0)
#define	SLMP_ERR_REQ_DATA_SIZE	(0xC0EE1)
#define	SLMP_ERR_RES_DATA_SIZE	(0xC0EE2)
#define	SLMP_ERR_NO_EXIST_SERVER_NO	(0xCF10)
#define	SLMP_ERR_CAN_NOT_COMMUNICATION_SETTING	(0xCF20)
#define	SLMP_ERR_NO_EXIST_PARAM_ID	(0xCF30)
#define	SLMP_ERR_CAN_NOT_PARAMETER_SET	(0xCF31)
#define	SLMP_END_DUPLICATE_MASTER	(0xCFE0)
#define	SLMP_END_INVALID_NUMBER_OF_OCCUPIED_STATIONS	(0xCFE1)
#define	SLMP_END_SLAVE	(0xCFF0)
#define	SLMP_END_DISCONNECTED_REQUEST	(0xCFFF)

ユーザープログラム内の関数戻り値として返すエラーコードは以下を使用する。

[CCIEF_BASIC_SLAVE.h]

#define	CCIEF_BASIC_SLAVE_ERR_OK	0
#define	CCIEF_BASIC_SLAVE_ERR_NG	(-1)
#define	CCIEF_BASIC_SLAVE_ERR_DEVICE_RANGE	(-100)

[SLMP_SERVER.h]

#define	SLMP_SERVER_ERR_OK	0
#define	SLMP_SERVER_ERR_NG	(-1)
#define	SLMP_SERVER_ERR_UNSUPPORT_SERVICE	(-2)

[SOCKET.h]

#define	SOCKET_ERR_OK	0
#define	SOCKET_ERR_SOCKET	(-100)
#define	SOCKET_ERR_RECV	(-103)
#define	SOCKET_ERR_SEND	(-104)
#define	SOCKET_ERR_NO_RECEIVABLE	(-200)

[TIMER.h]

#define	TIMER_OK	0
#define	TIMER_RESOURCE_NONE	(-1)

[USER_SAMPLE.h]

#define	USER_ERR_OK	0
#define	USER_ERR_NG	(-1)

5.6.2 SLMP_MakePacketStream

表 13 SLMP_MakePacketStream

機能	SLMP 通信パケット作成			
ファイル名	SLMP.c	公開/非公開	公開	
呼び出し形式	int SLMP_MakePacketStream (uint32_t ulFrameType, const SLMP_INFO *p, uint8_t *pucStream)			
引数	型名	変数名	内容	入出力
	uint32_t	ulFrameType	フレーム種別	入力
	const SLMP_INFO *	p	SLMP 情報	入力
	uint8_t *	pucStream	送信パケット	出力
戻り値	SLMP_ERR_OK 正常 SLMP_ERR_NG 異常			
説明	SLMP 通信パケット生成			

SLMP_INFO の構成を、サンプルコードを元に下記に示す。

[SLMP.h]

typedef struct	
{	
uint32_t ulFrameType;	/* Frame Type */
uint16_t usSerialNumber;	/* Serial Number */
uint16_t usNetNumber;	/* Network Number */
uint16_t usNodeNumber;	/* Node Number */
uint16_t usProcNumber;	/* Processor Number */
uint16_t usDataLength;	/* Data Length */
uint16_t usTimer;	/* Timer Value */
uint16_t usCommand;	/* Command */
uint16_t usSubCommand;	/* Sub Command */
uint16_t usEndCode;	/* End Code */
uint8_t * pucData;	/* Data */
}SLMP_INFO;	

5.6.3 SLMP_GetSmpInfo

表 14 SLMP_GetSmpInfo

機能	SLMP 情報取得			
ファイル名	SLMP.c	公開/非公開	公開	
呼び出し形式	int SLMP_GetSmpInfo (SLMP_INFO *p, const uint8_t *pucStream)			
引数	型名	変数名	内容	入出力
	SLMP_INFO *	p	SLMP 情報	出力
	uint8_t *	pucStream	受信パケット	入力
戻り値	SLMP_ERR_OK 正常 SLMP_ERR_NG 異常			
説明	SLMP 情報取得			

5.6.4 local_itoa

表 15 local_itoa

機能	数値から ASCII への変換			
ファイル名	SLMP.c	公開/非公開	公開	
呼び出し形式	uint8_t local_itoa (uint8_t ucInt)			
引数	型名	変数名	内容	入出力
	uint8_t	ucInt	数値	入力
戻り値	uint8_t ASCII コード			
説明	数値から ASCII への変換			

5.6.5 local_atoi

表 16 local_atoi

機能	ASCII から数値への変換			
ファイル名	SLMP.c	公開/非公開	公開	
呼び出し形式	uint8_t local_atoi (uint8_t ucInt)			
引数	型名	変数名	内容	入出力
	uint8_t	ucInt	数値	入力
戻り値	uint8_t 数値			
説明	ASCII から数値への変換			

5.6.6 SLMP_MakeErrorData

表 17 SLMP_MakeErrorData

機能	SLMP エラー応答データ作成			
ファイル名	SLMP.c	公開/非公開	公開	
呼び出し形式	int SLMP_MakeErrorData (const SLMP_INFO *p, uint8_t *pucStream , uint16_t *pusDataSize)			
引数	型名	変数名	内容	入出力
	const SLMP_INFO *	p	SLMP 情報	入力
	uint8_t *	pucStream	応答データ	出力
	uint16_t *	pusDataSize	応答データサイズ	出力
戻り値	SLMP_ERR_OK 正常 SLMP_ERR_NG 異常			
説明	SLMP エラー応答データ作成			

5.6.7 ccief_basic_slave_initialize

表 18 ccief_basic_slave_initialize

機能	CCIEF-BASIC スレーブ局初期化			
ファイル名	CCIEF_BASIC_SLAVE.c	公開/非公開	公開	
呼び出し形式	int ccief_basic_slave_initialize (CCIEF_BASIC_SLAVE_INFO *pSlave, CCIEF_BASIC_SLAVE_CALLBACK_RECV_CYCLIC_DATA pRecvCyclicDataFunc, CCIEF_BASIC_SLAVE_CALLBACK_CYCLIC_DISCONNECTION pCyclicDisconnectionFunc)			
引数	型名	変数名	内容	入出力
	CCIEF_BASIC_SLAVE_INFO *	pSlave	スレーブ局情報	入力
	CCIEF_BASIC_SLAVE_CALLBACK_RECV_CYCLIC_DATA	pRecvCyclicDataFunc	コールバック関数 (サイクリックデータ受信)	入力
	CCIEF_BASIC_SLAVE_CALLBACK_CYCLIC_DISCONNECTION	pCyclicDisconnectionFunc	コールバック関数 (解列検出)	入力
戻り値	CCIEF_BASIC_SLAVE_ERR_OK 正常 SOCKET_ERR_SOCKET ソケット異常			
説明	CCIEF-BASIC スレーブ局の初期化。 各変数の初期化。 ソケットの生成。			

CCIEF_BASIC_SLAVE_INFO の構成を、サンプルコードを元に下記に示す。

[CCIEF_BASIC_SLAVE.h]

typedef struct
{
uint16_t usVenderCode; /* Vender code */
uint32_t ulModelCode; /* Model code */
uint16_t usMachineVersion; /* Machine version */
uint32_t ullpAddress; /* Slave ip address */
int iOccupiedStationNumber; /* Number of occupied stations */
} CCIEF_BASIC_SLAVE_INFO;

5.6.8 ccief_basic_slave_terminate

表 19 ccief_basic_slave_terminate

機能	CCIEF-BASIC スレーブ局終了			
ファイル名	CCIEF_BASIC_SLAVE.c	公開/非公開	公開	
呼び出し形式	void ccief_basic_slave_terminate (void)			
引数	型名	変数名	内容	入出力
	-	-	-	-
戻り値	-			
説明	CCIEF-BASIC スレーブ局の終了。 ソケットをクローズ。			

5.6.9 ccief_basic_slave_main

表 20 ccief_basic_slave_main

機能	CCIEF-BASIC スレーブ局メイン処理			
ファイル名	CCIEF_BASIC_SLAVE.c	公開/非公開	公開	
呼び出し形式	int ccief_basic_slave_main (void)			
引数	型名	変数名	内容	入出力
	-	-	-	-
戻り値	CCIEF_BASIC_SLAVE_ERR_OK 正常 CCIEF_BASIC_SLAVE_ERR_NG 異常 SOCKET_ERR_RECV ソケット異常			
説明	パケット受信。 解列検出時間をチェック。			

5.6.10 ccief_basic_slave_set_rx

表 21 ccief_basic_slave_set_rx

機能	RX データセット			
ファイル名	CCIEF_BASIC_SLAVE.c	公開/非公開	公開	
呼び出し形式	int ccief_basic_slave_set_rx (int iNumber, int iValue)			
引数	型名	変数名	内容	入出力
	int	iNumber	デバイス番号	入力
	int	iValue	セット値 0 (ビット OFF) 1 (ビット ON)	入力
戻り値	CCIEF_BASIC_SLAVE_ERR_OK 正常 CCIEF_BASIC_SLAVE_ERR_DEVICE_RANGE デバイス番号異常			
説明	引数 iNumber で指定されたデバイス番号の RX に、引数 iValue の値をセット。			

5.6.11 ccief_basic_slave_get_ry

表 22 ccief_basic_slave_get_ry

機能	RY データ取得			
ファイル名	CCIEF_BASIC_SLAVE.c	公開/非公開	公開	
呼び出し形式	int ccief_basic_slave_get_ry (int iNumber, int *piValue)			
引数	型名	変数名	内容	入出力
	int	iNumber	デバイス番号	入力
	int *	piValue	データ格納先ポインタ 格納値: 0 (ビット OFF) 1 (ビット ON)	出力
戻り値	CCIEF_BASIC_SLAVE_ERR_OK 正常 CCIEF_BASIC_SLAVE_ERR_DEVICE_RANGE デバイス番号異常			
説明	引数 iNumber で指定されたデバイス番号の RY データを取得。			

5.6.12 ccief_basic_slave_get_rww

表 23 ccief_basic_slave_get_rww

機能	RWw データ取得			
ファイル名	CCIEF_BASIC_SLAVE.c	公開/非公開	公開	
呼び出し形式	int ccief_basic_slave_get_rww (int iNumber, uint16_t *pusValue)			
引数	型名	変数名	内容	入出力
	int	iNumber	デバイス番号	入力
	uint16_t *	pusValue	データ格納先ポインタ	出力
戻り値	CCIEF_BASIC_SLAVE_ERR_OK 正常 CCIEF_BASIC_SLAVE_ERR_DEVICE_RANGE デバイス番号異常			
説明	引数 iNumber で指定されたデバイス番号の RWw データを取得。			

5.6.13 ccief_basic_slave_set_rwr

表 24 ccief_basic_slave_set_rwr

機能	RWr データセット			
ファイル名	CCIEF_BASIC_SLAVE.c	公開/非公開	公開	
呼び出し形式	int ccief_basic_slave_set_rwr (int iNumber, uint16_t usValue)			
引数	型名	変数名	内容	入出力
	int	iNumber	デバイス番号	入力
	uint16_t	usValue	セット値	入力
戻り値	CCIEF_BASIC_SLAVE_ERR_OK 正常 CCIEF_BASIC_SLAVE_ERR_DEVICE_RANGE デバイス番号異常			
説明	引数 iNumber で指定されたデバイス番号の RWr に、引数 iValue の値をセット。			

5.6.14 ccief_basic_slave_get_pointer

表 25 ccief_basic_slave_get_pointer

機能	デバイス先頭ポインタ取得			
ファイル名	CCIEF_BASIC_SLAVE.c	公開/非公開	公開	
呼び出し形式	uint16_t *ccief_basic_slave_get_pointer (int iDeviceType)			
引数	型名	変数名	内容	入出力
	int	iDeviceType	デバイス種別	入力
戻り値	デバイス先頭ポインタ			
説明	デバイスの先頭ポインタを取得。			

デバイス種別の定義を、サンプルコードを元に下記に示す。

[CCIEF_BASIC_SLAVE.h]

#define	CCIEF_BASIC_DEVICE_TYPE_RX	1	/* Type of device for RX */
#define	CCIEF_BASIC_DEVICE_TYPE_RY	2	/* Type of device for RY */
#define	CCIEF_BASIC_DEVICE_TYPE_RWW	3	/* Type of device for RWW */
#define	CCIEF_BASIC_DEVICE_TYPE_RWR	4	/* Type of device for RWR */

5.6.15 ccief_basic_slave_set_unit_info

表 26 ccief_basic_slave_set_unit_info

機能	自局ユニット情報セット			
ファイル名	CCIEF_BASIC_SLAVE.c	公開/非公開	公開	
呼び出し形式	void ccief_basic_slave_set_unit_info (uint16_t usUnitInfo)			
引数	型名	変数名	内容	入出力
	uint16_t	usUnitInfo	ユニット情報	入力
戻り値	-			
説明	自局ユニット情報をセット。			

アプリケーション動作状態の定義を、サンプルコードを元に下記に示す。

[CCIEF_BASIC_SLAVE.h]

```
#define CCIEF_BASIC_UNIT_INFO_APPLICATION_STOP 0x0000 /* Stopping application for setting the unit info */
#define CCIEF_BASIC_UNIT_INFO_APPLICATION_RUNNING 0x0001 /* Running application for setting the unit info */
```

5.6.16 ccief_basic_slave_set_err_code

表 27 ccief_basic_slave_set_err_code

機能	エラーコードセット			
ファイル名	CCIEF_BASIC_SLAVE.c	公開/非公開	公開	
呼び出し形式	void ccief_basic_slave_set_err_code (uint16_t usErrCode)			
引数	型名	変数名	内容	入出力
	uint16_t	usErrCode	エラーコード	入力
戻り値	-			
説明	エラーコードをセット。			

5.6.17 ccief_basic_slave_set_unit_data

表 28 ccief_basic_slave_set_unit_data

機能	自局管理情報セット			
ファイル名	CCIEF_BASIC_SLAVE.c	公開/非公開	公開	
呼び出し形式	void ccief_basic_slave_set_unit_data (uint32_t ulUnitData)			
引数	型名	変数名	内容	入出力
	uint32_t	ulUnitData	自局管理情報	入力
戻り値	－			
説明	自局管理情報をセット。			

5.6.18 ccief_basic_slave_get_master_info

表 29 ccief_basic_slave_get_master_info

機能	マスタ局情報取得			
ファイル名	CCIEF_BASIC_SLAVE.c	公開/非公開	公開	
呼び出し形式	void ccief_basic_slave_get_master_info (CCIEF_BASIC_SLAVE_MASTER_INFO *pInfo)			
引数	型名	変数名	内容	入出力
	CCIEF_BASIC_SLAVE_MASTER_INFO *	pInfo	マスタ局情報格納先ポインタ	出力
戻り値	-			
説明	マスタ局情報を取得。			

CCIEF_BASIC_SLAVE_MASTER_INFO の構成を、サンプルコードを元に下記に示す。

[CCIEF_BASIC_SLAVE.h]

typedef struct				
{				
uint16_t	usUnitInfo;	/* Information of the unit */		
uint16_t	usReserve;	/* Reserve */		
uint8_t	aucTimeData[8];	/* Time of the master */		
} CCIEF_BASIC_MASTER_NOTIFY_INFO;				
typedef struct				
{				
uint32_t	ulId;	/* Id of the master */		
uint8_t	ucGroupNumber;	/* Group number */		
CCIEF_BASIC_MASTER_NOTIFY_INFO	NotifyInfo;	/* Notify information of the master */		
} CCIEF_BASIC_SLAVE_MASTER_INFO;				

5.6.19 ccief_basic_slave_recv_cyclic_data

表 30 ccief_basic_slave_recv_cyclic_data

機能	サイクリックデータ受信			
ファイル名	CCIEF_BASIC_SLAVE.c	公開/非公開	非公開	
呼び出し形式	int ccief_basic_slave_recv_cyclic_data (SOCKET sock, const SLMP_INFO *source, uint32_t ulRecvAddr, uint16_t usRecvPortNumber)			
引数	型名	変数名	内容	入出力
	SOCKET	sock	ソケットディスクリプタ	入力
	const SLMP_INFO *	source	受信パケット情報	入力
	uint32_t	ulRecvAddr	送信元 IP アドレス	入力
	uint16_t	usRecvPortNumber	送信元ポート番号	入力
戻り値	CCIEF_BASIC_SLAVE_ERR_OK 正常 SOCKET_ERR_SEND ソケット異常			
説明	受信データ解析。 マスタ局通知情報の取得。 RY、RWw に受信データを転送。			

5.6.20 ccief_basic_slave_send_cyclic_data

表 31 ccief_basic_slave_send_cyclic_data

機能	サイクリックデータ送信			
ファイル名	CCIEF_BASIC_SLAVE.c		公開/非公開	非公開
呼び出し形式	int ccief_basic_slave_send_cyclic_data (SOCKET sock, const SLMP_INFO *source, uint32_t ulSendAddr, uint16_t usSendPortNumber)			
引数	型名	変数名	内容	入出力
	SOCKET	sock	ソケットディスクリプタ	入力
	const SLMP_INFO *	source	受信パケット情報	入力
	uint32_t	ulSendAddr	送信先 IP アドレス	入力
	uint16_t	usSendPortNumber	送信先ポート番号	入力
戻り値	CCIEF_BASIC_SLAVE_ERR_OK 正常 SOCKET_ERR_SEND ソケット異常			
説明	RX、RWrを送信データに転送。 マスタ局へサイクリックデータを送信。			

5.6.21 ccief_basic_slave_send_cyclic_data_error

表 32 ccief_basic_slave_send_cyclic_data_error

機能	サイクリックエラーデータ送信			
ファイル名	CCIEF_BASIC_SLAVE.c	公開/非公開	非公開	
呼び出し形式	int ccief_basic_slave_send_cyclic_data_error (SOCKET sock, const SLMP_INFO *source, uint16_t usEndCode, uint32_t ulSendAddr, uint16_t usSendPortNumber)			
引数	型名	変数名	内容	入出力
	SOCKET	sock	ソケットディスクリプタ	入力
	const SLMP_INFO *	source	受信パケット情報	入力
	uint16_t	usEndCode	終了コード	入力
	uint32_t	ulSendAddr	送信先 IP アドレス	入力
	uint16_t	usSendPortNumber	送信先ポート番号	入力
戻り値	CCIEF_BASIC_SLAVE_ERR_OK 正常 SOCKET_ERR_SEND ソケット異常			
説明	マスタ局へサイクリックエラーデータを送信。			

5.6.22 ccief_basic_slave_disconnection

表 33 ccief_basic_slave_disconnection

機能	解列処理			
ファイル名	CCIEF_BASIC_SLAVE.c	公開/非公開	非公開	
呼び出し形式	void ccief_basic_slave_disconnection (void)			
引数	型名	変数名	内容	入出力
	-	-	-	-
戻り値	-			
説明	サイクリック通信の解列処理。 ユーザーが指定したコールバック関数(解列検出)を実行。			

5.6.23 ccief_basic_slave_disconnection_timer_timeout

表 34 ccief_basic_slave_disconnection_timer_timeout

機能	解列検出時間タイムアウト(コールバック関数)			
ファイル名	CCIEF_BASIC_SLAVE.c	公開/非公開	非公開	
呼び出し形式	void ccief_basic_slave_disconnection_timer_timeout (int iId, void *pCallbackArg)			
引数	型名	変数名	内容	入出力
	int	iId	タイマーID	入力
	void *	pCallbackArg	コールバック関数引数 (未使用)	入力
戻り値	-			
説明	サイクリック通信の解列検出時間がタイムアウトした際に、 タイマー機能からコールバック実行される。 サイクリック通信の解列処理を実行。			

5.6.24 slmp_server_initialize

表 35 slmp_server_initialize

機能	SLMP サーバー初期化			
ファイル名	SLMP_SERVER.c	公開/非公開	公開	
呼び出し形式	int slmp_server_initialize (SLMP_SERVER_INFO *pServerInfo, SLMP_SERVER_CALLBACK_IPADDRESS_SET_BASIC pIpAddrSetFunc, SLMP_SERVER_CALLBACK_PARAMETER_GET pParameterGetFunc, SLMP_SERVER_CALLBACK_PARAMETER_SET pParameterSetFunc, SLMP_SERVER_CALLBACK_PARAMETER_SET_END pParameterSetEndFunc, SLMP_SERVER_CALLBACK_REMOTE_RESET pRemoteResetFunc)			
引数	型名	変数名	内容	入出力
	SLMP_SERVER_INFO *	pServerInfo	サーバー情報	入力
	SLMP_SERVER_CALLBACK_IPADDRESS_SET_BASIC	pIpAddrSetFunc	コールバック関数 (通信設定)	入力
	SLMP_SERVER_CALLBACK_PARAMETER_GET	pParameterGetFunc	コールバック関数 (パラメータ読み出し)	入力
	SLMP_SERVER_CALLBACK_PARAMETER_SET	pParameterSetFunc	コールバック関数 (パラメータ書き込み)	入力
	SLMP_SERVER_CALLBACK_PARAMETER_SET_END	pParameterSetEndFunc	コールバック関数 (パラメータ書き込み完了)	入力
	SLMP_SERVER_CALLBACK_REMOTE_RESET	pRemoteResetFunc	コールバック関数 (ユニットリセット)	入力
戻り値	SLMP_SERVER_ERR_OK 正常 SOCKET_ERR_SOCKET ソケット異常			
説明	SLMP サーバーの初期化。 各変数の初期化。 ソケットの生成。			

SLMP_SERVER_INFO の構成を、サンプルコードを元に下記に示す。

[SLMP_SERVER.h]

```
typedef struct
{
    uint16_t    usVenderCode;           /* Vender code */
    uint32_t    ulModelCode;            /* Model code */
    uint16_t    usMachineVersion;       /* Machine version */
    uint8_t     aucMacAddress[6];       /* Mac Address */
    uint32_t    ulIpAddress;            /* Server ip address */
    uint32_t    ulSubnetMask;           /* Server subnet mask */
    uint32_t    ulDefaultGatewayIpAddress; /* Server default gateway ip address */
    uint16_t    usPortNumber;           /* Server port number */
    uint8_t     acHostname[64];        /* Hostname */
    uint16_t    usStatus;               /* Status */
    uint8_t     acTypeName[16];         /* Type name */
    uint16_t    usTypeNameCode;         /* Type name code */
    uint16_t    *pusMemory;             /* Pointer of the user memory */
    unsigned int uiMemorySize;          /* Size of the user memory */
} SLMP_SERVER_INFO;
```

5.6.25 slmp_server_terminate

表 36 slmp_server_terminate

機能	SLMP サーバー終了			
ファイル名	SLMP_SERVER.c	公開/非公開	公開	
呼び出し形式	void slmp_server_terminate (void)			
引数	型名	変数名	内容	入出力
	-	-	-	-
戻り値	-			
説明	SLMP サーバーの終了。 ソケットをクローズ。			

5.6.26 slmp_server_main

表 37 slmp_server_main

機能	SLMP サーバーメイン処理			
ファイル名	SLMP_SERVER.c	公開/非公開	公開	
呼び出し形式	int slmp_server_main (void)			
引数	型名	変数名	内容	入出力
	-	-	-	-
戻り値	SLMP_SERVER_ERR_OK 正常 SOCKET_ERR_RECV ソケット異常			
説明	各ソケットの処理呼び出し。			

5.6.27 slmp_server_user_port

表 38 slmp_server_user_port

機能	SLMP サーバーユーザー指定ポート受信処理			
ファイル名	SLMP_SERVER.c	公開/非公開	公開	
呼び出し形式	int slmp_server_user_port (void)			
引数	型名	変数名	内容	入出力
	-	-	-	-
戻り値	SLMP_SERVER_ERR_OK 正常 SOCKET_ERR_RECV ソケット異常			
説明	ユーザー指定ポートで受信したパケットに対して以下の処理を実施。 ・要求パケット受信 ・各サービス処理 ・応答パケット送信			

5.6.28 slmp_server_basic_port

表 39 slmp_server_basic_port

機能	SLMP サーバ—CCIEF—BASIC 用 NodeConnect ポート受信処理			
ファイル名	SLMP_SERVER.c	公開/非公開	公開	
呼び出し形式	int slmp_server_basic_port (void)			
引数	型名	変数名	内容	入出力
	—	—	—	—
戻り値	SLMP_SERVER_ERR_OK 正常 SOCKET_ERR_RECV ソケット異常			
説明	CCIEF—BASIC 用 NodeConnect ポート(61451)で受信したパケットに対して以下の処理を実施。 ・要求パケット受信 ・各サービス処理 ・応答パケット送信			

5.6.29 slmp_server_paramset_port

表 40 slmp_server_paramset_port

機能	SLMP サーバパラメータ設定用ポート受信処理			
ファイル名	SLMP_SERVER.c	公開/非公開	公開	
呼び出し形式	int slmp_server_paramset_port (void)			
引数	型名	変数名	内容	入出力
	-	-	-	-
戻り値	SLMP_SERVER_ERR_OK 正常 SOCKET_ERR_RECV ソケット異常			
説明	パラメータ設定用ポート(45237)で受信したパケットに対して以下の処理を実施。 ・要求パケット受信 ・各サービス処理 ・応答パケット送信			

5.6.30 slmp_server_set_status

表 41 slmp_server_set_status

機能	SLMP サーバー状態セット			
ファイル名	SLMP_SERVER.c	公開/非公開	公開	
呼び出し形式	void slmp_server_set_status (uint16_t usStatus)			
引数	型名	変数名	内容	入出力
	uint16_t	usStatus	サーバー状態	入力
戻り値	-			
説明	引数 usStatus で指定されたサーバー状態をセット。			

5.6.31 slmp_server_slmp_send_response

表 42 slmp_server_slmp_send_response

機能	SLMP 応答送信			
ファイル名	SLMP_SERVER.c		公開/非公開	非公開
呼び出し形式	int slmp_server_slmp_send_response (SOCKET sock, const SLMP_INFO *source, uint32_t ulSendAddr, uint16_t usSendPortNumber, uint8_t *pucSendData, uint16_t usSendDataSize)			
引数	型名	変数名	内容	入出力
	SOCKET	sock	ソケットディスクリプタ	入力
	const SLMP_INFO *	source	受信パケット情報	入力
	uint32_t	ulSendAddr	送信先 IP アドレス	入力
	uint16_t	usSendPortNumber	送信先ポート番号	入力
	uint8_t *	pucSendData	送信データ格納先ポインタ	入力
	uint16_t	usSendDataSize	送信データサイズ	入力
戻り値	SLMP_SERVER_ERR_OK 正常 SOCKET_ERR_SEND ソケット異常			
説明	SLMP 応答送信。			

5.6.32 slmp_server_slmp_send_err_response

表 43 slmp_server_slmp_send_err_response

機能	SLMP エラー応答送信			
ファイル名	SLMP_SERVER.c	公開/非公開	非公開	
呼び出し形式	int slmp_server_slmp_send_err_response (SOCKET sock, const SLMP_INFO *source, uint16_t usEndCode, uint32_t ulSendAddr, uint16_t usSendPortNumber)			
引数	型名	変数名	内容	入出力
	SOCKET	sock	ソケットディスクリプタ	入力
	const SLMP_INFO *	source	受信パケット情報	入力
	uint16_t	usEndCode	終了コード	入力
	uint32_t	ulSendAddr	送信先 IP アドレス	入力
	uint16_t	usSendPortNumber	送信先ポート番号	入力
戻り値	SLMP_SERVER_ERR_OK 正常 SOCKET_ERR_SEND ソケット異常			
説明	SLMP エラー応答送信。			

5.6.33 slmp_server_service

表 44 slmp_server_service

機能	SLMP サービス実行			
ファイル名	SLMP_SERVER.c		公開/非公開	公開
呼び出し形式	int slmp_server_service (SOCKET sock, const SLMP_INFO *source, uint32_t ulRecvAddr, uint16_t usRecvPortNumber, SLMP_SERVICE *pServiceTable, int iServiceTableNumber)			
引数	型名	変数名	内容	入出力
	SOCKET	sock	ソケットディスクリプタ	入力
	const SLMP_INFO *	source	受信パケット情報	入力
	uint32_t	ulRecvAddr	送信元 IP アドレス	入力
	uint16_t	usRecvPortNumber	送信元ポート番号	入力
	SLMP_SERVICE *	pServiceTable	SLMP サービステーブル	入力
	int	iServiceTableNumber	SLMP サービス数	入力
戻り値	SLMP_SERVER_ERR_OK 正常 SLMP_SERVER_ERR_UNsupport_SERVICE 未対応サービス異常 SOCKET_ERR_SEND ソケット異常			
説明	SLMP サービスを実行。			

SLMP_SERVICE の構成を、サンプルコードを元に下記に示す。

[SLMP_SERVER.h]

```

typedef struct
{
    uint16_t    usCommand;          /* Command */
    int         (*pFunc)( SOCKET sock, const SLMP_INFO *source, uint32_t ulRecvAddr,
                          uint16_t usRecvPortNumber ); /* Function */
} SLMP_SERVICE;

```

5.6.34 slmp_server_memory_read

表 45 slmp_server_memory_read

機能	メモリ読み出し			
ファイル名	SLMP_SERVER.c	公開/非公開	公開	
呼び出し形式	int slmp_server_memory_read (SOCKET sock, const SLMP_INFO *source, uint32_t ulRecvAddr, uint16_t usRecvPortNumber)			
引数	型名	変数名	内容	入出力
	SOCKET	sock	ソケットディスクリプタ	入力
	const SLMP_INFO *	source	受信パケット情報	入力
	uint32_t	ulRecvAddr	送信元 IP アドレス	入力
	uint16_t	usRecvPortNumber	送信元ポート番号	入力
戻り値	SLMP_SERVER_ERR_OK 正常 SOCKET_ERR_SEND ソケット異常			
説明	メモリ読み出しの要求データ受信。 応答データ作成、送信。			

5.6.35 slmp_server_memory_write

表 46 slmp_server_memory_write

機能	メモリ書き込み			
ファイル名	SLMP_SERVER.c		公開/非公開	公開
呼び出し形式	int slmp_server_memory_write (SOCKET sock, const SLMP_INFO *source, uint32_t ulRecvAddr, uint16_t usRecvPortNumber)			
引数	型名	変数名	内容	入出力
	SOCKET	sock	ソケットディスクリプタ	入力
	const SLMP_INFO *	source	受信パケット情報	入力
	uint32_t	ulRecvAddr	送信元 IP アドレス	入力
	uint16_t	usRecvPortNumber	送信元ポート番号	入力
戻り値	SLMP_SERVER_ERR_OK 正常 SOCKET_ERR_SEND ソケット異常			
説明	メモリ書き込みの要求データ受信。 応答データ作成、送信。			

5.6.36 slmp_server_node_search_basic

表 47 slmp_server_node_search_basic

機能	自動検出(CCIEF-BASIC 用)			
ファイル名	SLMP_SERVER.c	公開/非公開	公開	
呼び出し形式	int slmp_server_node_search_basic (SOCKET sock, const SLMP_INFO *source, uint32_t ulRecvAddr, uint16_t usRecvPortNumber)			
引数	型名	変数名	内容	入出力
	SOCKET	sock	ソケットディスクリプタ	入力
	const SLMP_INFO *	source	受信パケット情報	入力
	uint32_t	ulRecvAddr	送信元 IP アドレス	入力
	uint16_t	usRecvPortNumber	送信元ポート番号	入力
戻り値	SLMP_SERVER_ERR_OK 正常 SOCKET_ERR_SEND ソケット異常			
説明	自動検出の要求データ受信。 応答データ作成。 ブロードキャスト送信待ち時間取得。 自動検出応答送信待ちタイマー開始。			

自動検出の応答データの設定値を、下記に示す。

表 48 自動検出の応答内容

No.	項目	設定値	備考
1	サーバーMAC アドレス	既定値	ネットワークアダプタの設定値
2	サーバーIP アドレス	既定値	ネットワークアダプタの設定値
3	サーバサブネットマスク	既定値	ネットワークアダプタの設定値
4	サーバデフォルトゲートウェイ IP アドレス	既定値	ネットワークアダプタの設定値
5	サーバホスト名	"SlaveSample"	define 名 "USER_SERVER_HOSTNAME" で定義※1
6	サーバベンダーコード	0x1234	define 名 "USER_PROFILE_VENDOR_CODE" で定義※1
7	サーバ形名コード	0x00010001	define 名 "USER_PROFILE_MODEL_CODE" で定義※1
8	サーバ機器バージョン	0x0001	define 名 "USER_PROFILE_MACHINE_VERSION" で定義※1
9	交信相手ユニット IP アドレス	0xFFFFFFFF	固定値
10	交信相手ユニット通信ポート番号	0xFFFF	固定値
11	サーバステータス	0x0000 (初期値)	プログラムで適時変更※2
12	サーバ通信ポート番号	61451	固定値(CCIEF-BASIC 自動検出用ポート番号)
13	サーバ通信プロトコル設定	1	固定値(UDP)

※1 USER_SAMPLE.h で定義。

※2 slmp_server_set_status の設定値。

5.6.37 slmp_server_node_search_send_response_timeout

表 49 slmp_server_node_search_send_response_timeout

機能	自動検出応答送信待ちタイムアウト (コールバック関数)			
ファイル名	SLMP_SERVER.c	公開/非公開	非公開	
呼び出し形式	void slmp_server_node_search_send_response_timeout (int iId, void *pCallbackArg)			
引数	型名	変数名	内容	入出力
	int	iId	タイマーID（未使用）	入力
	void *	pCallbackArg	コールバック関数引数	入力
戻り値	-			
説明	自動検出応答送信待ちタイムアウトの際に、 タイマー機能からコールバック実行される。 応答データ送信。			

5.6.38 slmp_server_ip_address_set_basic

表 50 slmp_server_ip_address_set_basic

機能	通信設定(CCIEF-BASIC 用)			
ファイル名	SLMP_SERVER.c	公開/非公開	公開	
呼び出し形式	int slmp_server_ip_address_set_basic (SOCKET sock, const SLMP_INFO *source, uint32_t ulRecvAddr, uint16_t usRecvPortNumber)			
引数	型名	変数名	内容	入出力
	SOCKET	sock	ソケットディスクリプタ	入力
	const SLMP_INFO *	source	受信パケット情報	入力
	uint32_t	ulRecvAddr	送信元 IP アドレス	入力
	uint16_t	usRecvPortNumber	送信元ポート番号	入力
戻り値	SLMP_SERVER_ERR_OK 正常 SOCKET_ERR_SEND ソケット異常			
説明	通信設定の要求データ受信。 ユーザーが指定したコールバック関数(通信設定)を実行。 応答データ作成、送信。			

5.6.39 slmp_server_device_info_compare

表 51 slmp_server_device_info_compare

機能	機器接続情報確認			
ファイル名	SLMP_SERVER.c	公開/非公開	公開	
呼び出し形式	int slmp_server_device_info_compare (SOCKET sock, const SLMP_INFO *source, uint32_t ulRecvAddr, uint16_t usRecvPortNumber)			
引数	型名	変数名	内容	入出力
	SOCKET	sock	ソケットディスクリプタ	入力
	const SLMP_INFO *	source	受信パケット情報	入力
	uint32_t	ulRecvAddr	送信元 IP アドレス	入力
	uint16_t	usRecvPortNumber	送信元ポート番号	入力
戻り値	SLMP_SERVER_ERR_OK 正常 SOCKET_ERR_SEND ソケット異常			
説明	機器接続情報確認の要求データ受信。 応答データ作成、送信。			

5.6.40 slmp_server_parameter_get

表 52 slmp_server_parameter_get

機能	パラメータ読み出し			
ファイル名	SLMP_SERVER.c	公開/非公開	公開	
呼び出し形式	int slmp_server_parameter_get (SOCKET sock, const SLMP_INFO *source, uint32_t ulRecvAddr, uint16_t usRecvPortNumber)			
引数	型名	変数名	内容	入出力
	SOCKET	sock	ソケットディスクリプタ	入力
	const SLMP_INFO *	source	受信パケット情報	入力
	uint32_t	ulRecvAddr	送信元 IP アドレス	入力
	uint16_t	usRecvPortNumber	送信元ポート番号	入力
戻り値	SLMP_SERVER_ERR_OK 正常 SOCKET_ERR_SEND ソケット異常			
説明	パラメータ読み出しの要求データ受信。 ユーザーが指定したコールバック関数(パラメータ読み出し)を実行。 応答データ作成、送信。			

5.6.41 slmp_server_parameter_set

表 53 slmp_server_parameter_set

機能	パラメータ書き込み			
ファイル名	SLMP_SERVER.c	公開/非公開	公開	
呼び出し形式	int slmp_server_parameter_set (SOCKET sock, const SLMP_INFO *source, uint32_t ulRecvAddr, uint16_t usRecvPortNumber)			
引数	型名	変数名	内容	入出力
	SOCKET	sock	ソケットディスクリプタ	入力
	const SLMP_INFO *	source	受信パケット情報	入力
	uint32_t	ulRecvAddr	送信元 IP アドレス	入力
	uint16_t	usRecvPortNumber	送信元ポート番号	入力
戻り値	SLMP_SERVER_ERR_OK 正常 SOCKET_ERR_SEND ソケット異常			
説明	パラメータ書き込みの要求データ受信。 ユーザーが指定したコールバック関数(パラメータ書き込み)を実行。 応答データ作成、送信。			

5.6.42 slmp_server_parameter_set_start

表 54 slmp_server_parameter_set_start

機能	パラメータ書き込み排他処理開始			
ファイル名	SLMP_SERVER.c	公開/非公開	公開	
呼び出し形式	int slmp_server_parameter_set_start (SOCKET sock, const SLMP_INFO *source, uint32_t ulRecvAddr, uint16_t usRecvPortNumber)			
引数	型名	変数名	内容	入出力
	SOCKET	sock	ソケットディスクリプタ	入力
	const SLMP_INFO *	source	受信パケット情報	入力
	uint32_t	ulRecvAddr	送信元 IP アドレス	入力
	uint16_t	usRecvPortNumber	送信元ポート番号	入力
戻り値	SLMP_SERVER_ERR_OK 正常 SOCKET_ERR_SEND ソケット異常			
説明	パラメータ書き込み排他処理開始の要求データ受信。 応答データ作成、送信。			

5.6.43 slmp_server_parameter_set_end

表 55 slmp_server_parameter_set_end

機能	パラメータ書き込み排他処理終了			
ファイル名	SLMP_SERVER.c	公開/非公開	公開	
呼び出し形式	int slmp_server_parameter_set_end (SOCKET sock, const SLMP_INFO *source, uint32_t ulRecvAddr, uint16_t usRecvPortNumber)			
引数	型名	変数名	内容	入出力
	SOCKET	sock	ソケットディスクリプタ	入力
	const SLMP_INFO *	source	受信パケット情報	入力
	uint32_t	ulRecvAddr	送信元 IP アドレス	入力
	uint16_t	usRecvPortNumber	送信元ポート番号	入力
戻り値	SLMP_SERVER_ERR_OK 正常 SOCKET_ERR_SEND ソケット異常			
説明	パラメータ書き込み排他処理終了の要求データ受信。 ユーザーが指定したコールバック関数(パラメータ書き込み完了)を実行。 応答データ作成、送信。			

5.6.44 slmp_server_parameter_set_cancel

表 56 slmp_server_parameter_set_cancel

機能	パラメータ書き込み排他処理キャンセル			
ファイル名	SLMP_SERVER.c	公開/非公開	公開	
呼び出し形式	int slmp_server_parameter_set_cancel (SOCKET sock, const SLMP_INFO *source, uint32_t ulRecvAddr, uint16_t usRecvPortNumber)			
引数	型名	変数名	内容	入出力
	SOCKET	sock	ソケットディスクリプタ	入力
	const SLMP_INFO *	source	受信パケット情報	入力
	uint32_t	ulRecvAddr	送信元 IP アドレス	入力
	uint16_t	usRecvPortNumber	送信元ポート番号	入力
戻り値	SLMP_SERVER_ERR_OK 正常 SOCKET_ERR_SEND ソケット異常			
説明	パラメータ書き込み排他処理キャンセルの要求データ受信。 応答データ作成、送信。			

5.6.45 slmp_server_communication_setting_get

表 57 slmp_server_communication_setting_get

機能	通信設定取得			
ファイル名	SLMP_SERVER.c	公開/非公開	公開	
呼び出し形式	int slmp_server_communication_setting_get (SOCKET sock, const SLMP_INFO *source, uint32_t ulRecvAddr, uint16_t usRecvPortNumber)			
引数	型名	変数名	内容	入出力
	SOCKET	sock	ソケットディスクリプタ	入力
	const SLMP_INFO *	source	受信パケット情報	入力
	uint32_t	ulRecvAddr	送信元 IP アドレス	入力
	uint16_t	usRecvPortNumber	送信元ポート番号	入力
戻り値	SLMP_SERVER_ERR_OK 正常 SOCKET_ERR_SEND ソケット異常			
説明	通信設定取得の要求データ受信。 応答データ作成、送信。			

5.6.46 slmp_server_read_type_name

表 58 slmp_server_read_type_name

機能	形名読み出し			
ファイル名	SLMP_SERVER.c	公開/非公開	公開	
呼び出し形式	int slmp_server_read_type_name (SOCKET sock, const SLMP_INFO *source, uint32_t ulRecvAddr, uint16_t usRecvPortNumber)			
引数	型名	変数名	内容	入出力
	SOCKET	sock	ソケットディスクリプタ	入力
	const SLMP_INFO *	source	受信パケット情報	入力
	uint32_t	ulRecvAddr	送信元 IP アドレス	入力
	uint16_t	usRecvPortNumber	送信元ポート番号	入力
戻り値	SLMP_SERVER_ERR_OK 正常 SOCKET_ERR_SEND ソケット異常			
説明	形名読み出しの要求データ受信。 応答データ作成、送信。			

形名読み出しの応答データの設定値を、下記に示す。

表 59 形名読み出しの応答内容

No.	項目	設定値	備考
1	形名	"SampleCode "	define 名 "USER_TYPE_NAME" で定義※1
2	形名コード	0x1234	define 名 "USER_TYPE_NAME_CODE" で定義※1

※1 USER_SAMPLE.h で定義。

5.6.47 slmp_server_remote_reset

表 60 slmp_server_remote_reset

機能	リモートリセット			
ファイル名	SLMP_SERVER.c	公開/非公開	公開	
呼び出し形式	int slmp_server_remote_reset (SOCKET sock, const SLMP_INFO *source, uint32_t ulRecvAddr, uint16_t usRecvPortNumber)			
引数	型名	変数名	内容	入出力
	SOCKET	sock	ソケットディスクリプタ	入力
	const SLMP_INFO *	source	受信パケット情報	入力
	uint32_t	ulRecvAddr	送信元 IP アドレス	入力
	uint16_t	usRecvPortNumber	送信元ポート番号	入力
戻り値	SLMP_SERVER_ERR_OK 正常 SOCKET_ERR_SEND ソケット異常			
説明	リモートリセットの要求データ受信。 ユーザーが指定したコールバック関数(リモートリセット)を実行。 応答データ作成、送信。			

5.6.48 socket_initialize

表 61 socket_initialize

機能	ソケット初期化			
ファイル名	SOCKET.c	公開/非公開	公開	
呼び出し形式	int socket_initialize (SOCKET *sock, uint32_t ulIpAddress, uint16_t usPortNumber)			
引数	型名	変数名	内容	入出力
	SOCKET *	sock	ソケットディスクリプタ 格納先ポインタ	出力
	uint32_t	ulIpAddress	IP アドレス	入力
	uint16_t	usPortNumber	ポート番号	入力
戻り値	SOCKET_ERR_OK 正常 SOCKET_ERR_SOCKET ソケット生成異常			
説明	ソケットの初期化。 ソケットディスクリプタを返す。 ユーザーの実装先の環境に応じて書き換えること。			

5.6.49 socket_terminate

表 62 socket_terminate

機能	ソケット終了			
ファイル名	SOCKET.c		公開/非公開	公開
呼び出し形式	void socket_terminate (SOCKET sock)			
引数	型名	変数名	内容	入出力
	SOCKET	sock	ソケットディスクリプタ	入力
戻り値	-			
説明	ソケットの終了。 ユーザーの実装先の環境に応じて書き換えること。			

5.6.50 socket_recv

表 63 socket_recv

機能	パケット受信			
ファイル名	SOCKET.c		公開/非公開	公開
呼び出し形式	int socket_recv (SOCKET sock, uint8_t *pucStream, int iLength, uint32_t *pulRecvAddr, uint16_t *pusRecvPortNumber)			
引数	型名	変数名	内容	入出力
	SOCKET	sock	ソケットディスクリプタ	入力
	uint8_t *	pucStream	受信パケット	出力
	int	iLength	受信パケット長	入力
	uint32_t *	pulRecvAddr	送信元 IP アドレス	出力
	uint16_t *	pusRecvPortNumber	送信元ポート番号	出力
戻り値	SOCKET_ERR_OK 正常 SOCKET_ERR_NO_RECEIVABLE 受信データなし SOCKET_ERR_RECV ソケット受信異常			
説明	パケット受信。 ユーザーの実装先の環境に応じて書き換えること。 ※本関数は定期的に実行される必要がある。			

5.6.51 socket_send

表 64 socket_send

機能	パケット送信			
ファイル名	SOCKET.c	公開/非公開	公開	
呼び出し形式	int socket_send (SOCKET sock, uint8_t *pucStream, int iLength, uint32_t ulSendAddr, uint16_t usSendPortNumber)			
引数	型名	変数名	内容	入出力
	SOCKET	sock	ソケットディスクリプタ	入力
	uint8_t *	pucStream	送信パケット	入力
	int	iLength	送信パケット長	入力
	uint32_t	ulSendAddr	送信先 IP アドレス	入力
	uint16_t	usSendPortNumber	送信先ポート番号	入力
戻り値	SOCKET_ERR_OK 正常 SOCKET_ERR_SEND ソケット送信異常			
説明	パケット送信。 ユーザーの実装先の環境に応じて書き換えること。			

5.6.52 timer_initialize

表 65 timer_initialize

機能	タイマー初期化			
ファイル名	TIMER.c		公開/非公開	公開
呼び出し形式	void timer_initialize (void)			
引数	型名	変数名	内容	入出力
	-	-	-	-
戻り値	-			
説明	タイマー機能の初期化。 ※タイマー最大個数は、define 名 "TIMER_MAX" で定義。			

5.6.53 timer_terminate

表 66 timer_terminate

機能	タイマー終了			
ファイル名	TIMER.c		公開/非公開	公開
呼び出し形式	void timer_terminate (void)			
引数	型名	変数名	内容	入出力
	－	－	－	－
戻り値	－			
説明	タイマー機能の終了。			

5.6.54 timer_main

表 67 timer_main

機能	タイマーメイン処理			
ファイル名	TIMER.c	公開/非公開		公開
呼び出し形式	void timer_main (void)			
引数	型名	変数名	内容	入出力
	-	-	-	-
戻り値	-			
説明	タイマーメイン処理。 タイムアップ時に、ユーザーが指定したコールバック関数を実行。 ※本関数は定期的に実行される必要がある。			

5.6.55 timer_start

表 68 timer_start

機能	タイマー開始			
ファイル名	TIMER.c		公開/非公開	公開
呼び出し形式	int timer_start (uint32_t ulTime, int *pId, TIMER_CALLBACK pCallbackFunc, void *pCallbackArg)			
引数	型名	変数名	内容	入出力
	uint32_t	ulTime	タイムアウト時間[ms]	入力
	int *	pId	タイマーID 格納先ポインタ	出力
	TIMER_CALLBACK	pCallbackFunc	コールバック関数 格納元ポインタ	入力
	void *	pCallbackArg	コールバック関数の引数	入力
戻り値	TIMER_OK 正常 TIMER_RESOURCE_NONE タイマー枯渇			
説明	タイマーを開始。 引数 pId で指定された格納先へ、開始したタイマーID をセット。 引数 pCallbackFunc で指定されたコールバック関数を登録。 引数 pCallbackArg で指定されたコールバック引数を登録。			

5.6.56 timer_stop

表 69 timer_stop

機能	タイマー停止			
ファイル名	TIMER.c		公開/非公開	公開
呼び出し形式	void timer_main (int iId)			
引数	型名	変数名	内容	入出力
	int	iId	タイマーID	入力
戻り値	－			
説明	引数 piId で指定されたタイマーを停止。			

5.6.57 timer_get_time

表 70 timer_get_time

機能	現在時間取得			
ファイル名	TIMER.c		公開/非公開	公開
呼び出し形式	uint32_t timer_get_time (void)			
引数	型名	変数名	内容	入出力
	－	－	－	－
戻り値	経過時間(プロセッサ時間) [ms]			
説明	経過時間(プロセッサ時間)を返す。 ユーザーの実装先の環境に応じて書き換えること。			

5.6.58 timer_broadcast_send_wait_time

表 71 timer_broadcast_send_wait_time

機能	ブロードキャスト送信待ち時間取得			
ファイル名	TIMER.c	公開/非公開	公開	
呼び出し形式	uint32_t timer_broadcast_send_wait_time (uint32_t ulMaxWaitTime)			
引数	型名	変数名	内容	入出力
	uint32_t	ulMaxWaitTime	応答待ち時間の最大値	入力
戻り値	送信待ち時間[ms]			
説明	ブロードキャスト送信待ち時間取得。			

5.6.59 timer_analyze_time_data

表 72 timer_analyze_time_data

機能	時計情報解析			
ファイル名	TIMER.c		公開/非公開	公開
呼び出し形式	void timer_analyze_time_data (int64_t llTime, TIMER_TIME_DATA *pTimeData)			
引数	型名	変数名	内容	入出力
	int64_t	llTime	時計情報(UNIX 時間)	入力
	TIMER_TIME_DATA *	pTimeData	時計データの格納先ポインタ	出力
戻り値	-			
説明	引数 llTime の時間を解析。 引数 pTimeData に格納。			

TIMER_TIME_DATA の構成を、サンプルコードを元に下記に示す。

[TIMER.h]

```
typedef struct
{
    uint16_t  usYear;           /* Year */
    uint16_t  usMonth;         /* Month */
    uint16_t  usDay;           /* Day */
    uint16_t  usHour;          /* Hour */
    uint16_t  usMinute;        /* Minute */
    uint16_t  usSecond;        /* Second */
    uint16_t  usMilliseconds;  /* Milliseconds */
} TIMER_TIME_DATA;
```

5.6.60 main

表 73 main

機能	メイン処理			
ファイル名	USER_SAMPLE.c	公開/非公開	公開	
呼び出し形式	void main (int argc, char *argv[])			
引数	型名	変数名	内容	入出力
	int	argc	コマンドライン引数の総個数	入力
	char *	argv[]	コマンドライン引数	入力
戻り値	-			
説明	ネットワークアダプタ情報の取得、設定。 CCIEF-BASIC スレーブ局、および SLMP サーバーの初期化。 タイマー機能の初期化。 パラメータの読み書き。 CCIEF-BASIC スレーブ局、および SLMP サーバーのメイン処理の実行。(ループ処理) タイマー機能のメイン処理の実行。(ループ処理) リセット処理。 ユーザーの実装先の環境に応じて書き換えること。			

5.6.61 user_callback_recv_cyclic_data

表 74 user_callback_recv_cyclic_data

機能	サイクリックデータ受信(コールバック関数)			
ファイル名	USER_SAMPLE.c		公開/非公開	公開
呼び出し形式	void user_callback_recv_cyclic_data (int iCyclicState, int iOccupiedStationNumber)			
引数	型名	変数名	内容	入出力
	int	iCyclicState	サイクリック伝送状態 0 (サイクリックデータ無効) 1 (サイクリックデータ有効)	入力
	int	iOccupiedStationNumber	受信データの占有局数	入力
戻り値	-			
説明	CCIEF-BASIC マスタ局からのサイクリックデータ受信時に実行される。 ユーザーの実装先の環境に応じて書き換えること。 ※サンプルコードでは、以下の処理を実施。 RY、RWw のデータを、RX、RWrのデータに折り返し転送。 応答遅延時間待つ。 自局ユニット情報をセット。			

5.6.62 user_callback_cyclic_disconnection

表 75 user_callback_cyclic_disconnection

機能	解列検出(コールバック関数)			
ファイル名	USER_SAMPLE.c	公開/非公開	公開	
呼び出し形式	void user_callback_cyclic_disconnection (void)			
引数	型名	変数名	内容	入出力
	-	-	-	-
戻り値	-			
説明	CCIEF-BASIC マスタ局とのサイクリック通信が解列、または停止した場合に実行される。 ユーザーの実装先の環境に応じて書き換えること。 ※サンプルコードでは、以下の処理を実施。 自局ユニット情報をセット。			

5.6.63 user_callback_set_ip_address_basic

表 76 user_callback_set_ip_address_basic

機能	通信設定(CCIEF-BASIC 用)(コールバック関数)			
ファイル名	USER_SAMPLE.c	公開/非公開	公開	
呼び出し形式	void user_callback_set_ip_address_basic (uint32_t ulIpAddress, uint32_t ulSubnetMask)			
引数	型名	変数名	内容	入出力
	uint32_t	ulIpAddress	IP アドレス	入力
	uint32_t	ulSubnetMask	サブネットマスク	入力
戻り値	-			
説明	SLMP 通信設定コマンド(0E31)の要求データ受信時に実行される。 ユーザーの実装先の環境に応じて書き換えること。 ※サンプルコードでは、以下の処理を実施。 引数で要求された通信設定を、ネットワークアダプタ設定にセット。 ユニットをリセット。			

5.6.64 user_callback_parameter_get

表 77 user_callback_parameter_get

機能	パラメータ読み出し(コールバック関数)			
ファイル名	USER_SAMPLE.c		公開/非公開	公開
呼び出し形式	int user_callback_parameter_get (uint16_t usId, uint16_t *pusSize, uint8_t **ppucData)			
引数	型名	変数名	内容	入出力
	uint16_t	usId	パラメータ ID	入力
	uint16_t *	pusSize	パラメータ値のサイズ 格納先ポインタ	出力
	uint8_t **	ppucData	パラメータ値 格納先ポインタ	出力
戻り値	SLMP_ERR_OK 正常 SLMP_ERR_NO_EXIST_PARAM_ID パラメータ ID 異常			
説明	SLMP パラメータ読み出しコマンド(0E33)の要求データ受信時に実行される。 ユーザーの実装先の環境に応じて書き換えること。 ※サンプルコードでは、以下の処理を実施。 引数 usId で指定された CCIEF-BASIC スレーブ局パラメータの パラメータ ID の値とサイズを、第 2、3 引数に格納する。			

5.6.65 user_callback_parameter_set

表 78 user_callback_parameter_set

機能	パラメータ書き込み(コールバック関数)			
ファイル名	USER_SAMPLE.c		公開/非公開	公開
呼び出し形式	int user_callback_parameter_set (uint16_t usId, uint16_t usSize, uint8_t *pucData)			
引数	型名	変数名	内容	入出力
	uint16_t	usId	パラメータ ID	入力
	uint16_t	usSize	パラメータ値のサイズ	入力
	uint8_t *	pucData	パラメータ値	入力
戻り値	SLMP_ERR_OK 正常 SLMP_ERR_NO_EXIST_PARAM_ID パラメータ ID 異常			
説明	SLMP パラメータ書き込みコマンド(0E34)の要求データ受信時に実行される。 ユーザーの実装先の環境に応じて書き換えること。 ※サンプルコードでは、以下の処理を実施。 引数で指定されたパラメータ ID の値を、CCIEF-BASIC スレーブ局のパラメータに反映する。			

5.6.66 user_callback_parameter_set_end

表 79 user_callback_parameter_set_end

機能	パラメータ書き込み完了(コールバック関数)			
ファイル名	USER_SAMPLE.c	公開/非公開	公開	
呼び出し形式	int user_callback_parameter_set_end (void)			
引数	型名	変数名	内容	入出力
	-	-	-	-
戻り値	SLMP_ERR_OK 正常			
説明	SLMP パラメータ書き込み完了コマンド(0E36)の要求データ受信時に実行される。 ユーザーの実装先の環境に応じて書き換えること。 ※サンプルコードでは、以下の処理を実施。 CCIEF-BASIC スレーブ局のパラメータを、ファイルに書き込む。			

5.6.67 user_callback_remote_reset

表 80 user_callback_remote_reset

機能	リモートリセット要求(コールバック関数)			
ファイル名	USER_SAMPLE.c	公開/非公開	公開	
呼び出し形式	int user_callback_remote_reset (void)			
引数	型名	変数名	内容	入出力
	-	-	-	-
戻り値	SLMP_ERR_OK 正常			
説明	SLMP リモートリセットコマンド(1006)の要求データ受信時に実行される。 ユーザーの実装先の環境に応じて書き換えること。 ※サンプルコードでは、以下の処理を実施。 リセット処理を発行。			

5.6.68 user_parameter_file_read

表 81 user_parameter_file_read

機能	パラメータファイル読み出し			
ファイル名	USER_SAMPLE.c	公開/非公開	非公開	
呼び出し形式	int user_parameter_file_read (char *file_path, USER_SLAVE_PARAMETER *pParameter, USER_ADAPTER_INFO *pAdapterInfo)			
引数	型名	変数名	内容	入出力
	char *	file_path	ファイル	入力
	USER_SLAVE_PARAMETER *	pParameter	CCIEF-BASIC スレーブ局 パラメータ格納先ポインタ	出力
	USER_ADAPTER_INFO *	pAdapterInfo	ネットワークアダプタ情報	出力
戻り値	USER_ERR_OK 正常 USER_ERR_NG 異常			
説明	引数 file_path で指定されたファイルを読み込み、引数 pParameter に格納。 引数 pAdapterInfo に、パラメータの設定値を反映。 ユーザーの実装先の環境に応じて書き換えること。			

USER_SLAVE_PARAMETER の構成を、サンプルコードを元に下記に示す。

[USER_SAMPLE.h]

```
typedef struct
{
    uint32_t  ulIpAddress;           /* Slave ip address */
    uint32_t  ulSubnetMask;         /* Subnet Mask */
    uint32_t  ulDefaultGatewayIpAddress; /* Default Gateway IP Address */
    uint16_t  usOccupiedStationNumber; /* Number of occupied stations */
    uint32_t  ulCyclicResponseWaitTime; /* Wait for cyclic response time [ms] (0:Not wait) */
} USER_SLAVE_PARAMETER;
```

USER_ADAPTER_INFO の構成を、サンプルコードを元に下記に示す。

[USER_SAMPLE.h]

```
typedef struct
{
    uint8_t   aucMacAddress[6];     /* MAC Address */
    uint32_t  ulIpAddress;         /* IP Address */
    uint32_t  ulSubnetMask;         /* Subnet Mask */
    uint32_t  ulDefaultGatewayIpAddress; /* Default Gateway IP Address */
} USER_ADAPTER_INFO;
```

5.6.69 user_parameter_file_write

表 82 user_parameter_file_write

機能	パラメータファイル書き込み			
ファイル名	USER_SAMPLE.c	公開/非公開	非公開	
呼び出し形式	int user_parameter_file_write (char *file_path, USER_SLAVE_PARAMETER *pParameter)			
引数	型名	変数名	内容	入出力
	char *	file_path	ファイル	入力
	USER_SLAVE_PARAMETER *	pParameter	CCIEF-BASIC スレーブ局 パラメータ格納先ポインタ	入力
戻り値	USER_ERR_OK 正常 USER_ERR_NG 異常			
説明	引数 file_path で指定されたファイルに、引数 pParameter の値を書き込む。 ユーザーの実装先の環境に応じて書き換えること。			

5.6.70 user_display_cyclic_information

表 83 user_display_cyclic_information

機能	サイクリック情報表示			
ファイル名	USER_SAMPLE.c	公開/非公開	非公開	
呼び出し形式	void user_display_cyclic_information (void)			
引数	型名	変数名	内容	入出力
	-	-	-	-
戻り値	-			
説明	サイクリック情報を画面に表示。 ユーザーの実装先の環境に応じて書き換えること。			

5.6.71 user_get_adapter_info

表 84 user_get_adapter_info

機能	ネットワークアダプタ情報取得			
ファイル名	USER_SAMPLE.c	公開/非公開	非公開	
呼び出し形式	int user_get_adapter_info (USER_ADAPTER_INFO *pGetAdapterInfo)			
引数	型名	変数名	内容	入出力
	USER_ADAPTER_INFO *	pGetAdapterInfo	ネットワークアダプタ情報 格納先ポインタ	出力
戻り値	USER_ERR_OK 正常 USER_ERR_NG 異常			
説明	ネットワークアダプタ情報を取得。 ユーザーの環境に応じて書き換えること。 ※サンプルコードは、Linux 上のネットワークアダプタの取得例を記述している。			

5.6.72 user_set_adapter_info

表 85 user_set_adapter_info

機能	ネットワークアダプタ情報設定			
ファイル名	USER_SAMPLE.c	公開/非公開	非公開	
呼び出し形式	int user_set_adapter_info (USER_ADAPTER_INFO *pSetAdapterInfo)			
引数	型名	変数名	内容	入出力
	USER_ADAPTER_INFO *	pSetAdapterInfo	ネットワークアダプタ情報	入力
戻り値	USER_ERR_OK 正常 USER_ERR_NG 異常			
説明	ネットワークアダプタ情報を設定。 ユーザーの環境に応じて書き換えること。 ※サンプルコードは、Linux 上のネットワークアダプタの設定例を記述している。			

6 付録: サンプルコードのコンパイルから実行までの手順

「gcc(GNU Compiler Collection)」を使用する場合のサンプルコードのコンパイルから実行までの手順を下記に示します。

OS は Linux とし、ディストリビューションは CentOS とする。

6.1 仕様

サンプルコードの実行環境の例を、図 17 に示します。

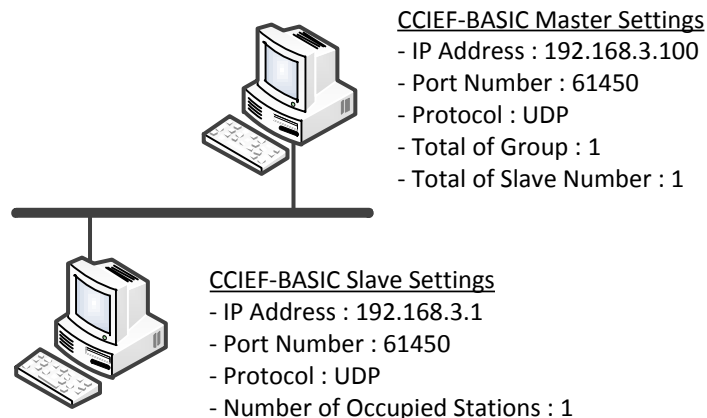


図 17. サンプルコードの実行環境

サンプルコードでは、CCIEF-BASIC マスタ局(サンプルアプリケーション※1)とのサイクリック通信を実行します。サイクリック通信の状況を、5 秒間隔で画面上に表示します。

コマンドライン引数で、ユーザーが設定したパラメータファイルを指定し、アプリケーションを起動します。(※2※3)

実行環境に複数のネットワークアダプタが装着されている場合は、装着されているネットワークアダプタの選択画面を表示し、ユーザーが選択したネットワークアダプタで、サンプルコードを起動します。

※1 詳細は、「GC-Link IE フィールドネットワーク Basic サンプルコード仕様書 マスタ局編」を参照ください。

※2 コマンドライン引数でパラメータファイルの指定が無い場合は、デフォルトのパラメータで起動します。

※3 パラメータで設定した CCIEF-BASIC スレーブ局の IP アドレスが、ネットワークアダプタの既定値と異なる場合は、パラメータの設定値でネットワークアダプタの設定が変更されます。

6.2 アプリケーションの作成

gcc を使用してアプリケーションを作成するまでの手順を示します。但しサンプルコードのディレクトリ構成は「/home/ユーザー名/CCIEF-BASIC_Slave」とする。

- (1) コンパイルを行う。「gcc」コマンドを持ちいてコンパイルを行う。
- (2) コンパイルを行うためのコマンドを図18に示す。

```

ユーザー@localhost:~
ファイル(F) 編集(E) 表示(V) 検索(S) 端末(T) ヘルプ(H)
[ユーザー@localhost ~]$ gcc -I /home/ユーザー/CCIEF-BASIC_Slave/library/include/ -c /home/ユーザー/CCIEF-BASIC_Slave/library/src/SLMP.c
[ユーザー@localhost ~]$ gcc -I /home/ユーザー/CCIEF-BASIC_Slave/sample/include/ -c /home/ユーザー/CCIEF-BASIC_Slave/sample/src/SOCKET.c
[ユーザー@localhost ~]$ gcc -I /home/ユーザー/CCIEF-BASIC_Slave/sample/include/ -c /home/ユーザー/CCIEF-BASIC_Slave/sample/src/TIMER.c
[ユーザー@localhost ~]$ gcc -I /home/ユーザー/CCIEF-BASIC_Slave/library/include/ -I /home/ユーザー/CCIEF-BASIC_Slave/sample/include/ -c /home/ユーザー/CCIEF-BASIC_Slave/sample/src/CCIEF_BASIC_SLAVE.c
[ユーザー@localhost ~]$ gcc -I /home/ユーザー/CCIEF-BASIC_Slave/library/include/ -I /home/ユーザー/CCIEF-BASIC_Slave/sample/include/ -c /home/ユーザー/CCIEF-BASIC_Slave/sample/src/SLMP_SERVER.c
[ユーザー@localhost ~]$ gcc -I /home/ユーザー/CCIEF-BASIC_Slave/library/include/ -I /home/ユーザー/CCIEF-BASIC_Slave/sample/include/ -c /home/ユーザー/CCIEF-BASIC_Slave/sample/src/USER_SAMPLE.c
[ユーザー@localhost ~]$ gcc SLMP.o CCIEF_BASIC_SLAVE.o SLMP_SERVER.o SOCKET.o TIMER.o USER_SAMPLE.o -o Slave_sample
[ユーザー@localhost ~]$

```

図18 コンパイルコマンド

6.3 アプリケーションの実行

gcc をアプリケーションを実行するまでの手順を示します。

- (1) CCIEF-BASIC スレーブ局のパラメータファイルを作成します。(詳細は、「4 仕様」を参照)

この時、パラメータファイルを変更する際に「Quoted field as text」のチェック欄にチェックが記されている場合チェックは外してください。

[SlaveParameter.csv]

```

,,
CCIEF-SIM Slave Sample Parameter,,
,,
ID, DATA, COMMENT
1, 0, IP Address
2, 0, Subnet Mask
3, 0, Default Gateway IP Address
4, 1, Occupied Station Number
5, 0, Cyclic Response Wait Time

```

- (2) ルートユーザでない場合はコマンドで「sudo ./Slave_sample SlaveParameter.csv」を実行する。この時に現在使用しているユーザのパスワードを入力する。ルートユーザの場合「./Slave_sample SlaveParameter.csv」を実行する。

※デフォルトゲートウェイが設定されていない場合はコマンドで「route add default gw 192.168.3.**」を実行し、デフォルトゲートウェイを追加する。

アプリケーションの実行例を、以下に示します。

CC-Link IE フィールドネットワーク Basic マスタ局とのサイクリック通信を実行した場合の出力は、次のようになります。

Start CC-Link IE Field Basic Slave!

IP Address: 192.168.3.1
Subnet mask: 255.255.255.0
Default GW IP address: 192.168.3.254
Number of Occupied Stations: 1
Cyclic response wait time: 0[ms] (0:Not wait)

Show the cyclic information at 5-second intervals.

Master | Cyclic Data | Link scan time | Time Data

Master - The control master station. (Master ID / Group Number)
Cyclic Data - Count of the receive Cyclic Data at intervals.
Link scan time - The link scan time at intervals. [ms]
Time Data - The time from the master information data.

CCIEF-BASIC: Master (Unknown) | Cyclic Data=0 | 0.000[ms] |
CCIEF-BASIC: Master (Unknown) | Cyclic Data=0 | 0.000[ms] |
CCIEF-BASIC: Master (Unknown) | Cyclic Data=0 | 0.000[ms] |
CCIEF-BASIC: Master (0xC0A80364/1) | Cyclic Data=4761 | 1.050[ms] | 2016-03-31 16:06:59
CCIEF-BASIC: Master (0xC0A80364/1) | Cyclic Data=4751 | 1.052[ms] | 2016-03-31 16:07:04
CCIEF-BASIC: Master (0xC0A80364/1) | Cyclic Data=4749 | 1.053[ms] | 2016-03-31 16:07:09
CCIEF-BASIC: Master (0xC0A80364/1) | Cyclic Data=4757 | 1.051[ms] | 2016-03-31 16:07:14
CCIEF-BASIC: Master (0xC0A80364/1) | Cyclic Data=4570 | 1.094[ms] | 2016-03-31 16:07:19