

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

マスタ局編

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_MASTER.c..... | 16 |
| (2) CCIEF_BASIC_SLAVES.c..... | 19 |
| (3) SOCKET.c..... | 19 |
| (4) TIMER.c..... | 19 |
| (5) USER_SAMPLE.c..... | 21 |
| 5.6 関数詳細..... | 24 |
| 5.6.1 戻り値の定義..... | 24 |
| 5.6.2 SLMP_MakePacketStream..... | 25 |
| 5.6.3 SLMP_GetSlmpInfo..... | 26 |
| 5.6.4 local_itoa..... | 26 |
| 5.6.5 local_atoi..... | 26 |
| 5.6.6 SLMP_MakeErrorData..... | 27 |
| 5.6.7 ccief_basic_master_initialize..... | 28 |
| 5.6.8 ccief_basic_master_terminate..... | 29 |
| 5.6.9 ccief_basic_master_main..... | 29 |
| 5.6.10 ccief_basic_master_start_cyclic..... | 30 |
| 5.6.11 ccief_basic_master_stop_cyclic..... | 30 |
| 5.6.12 ccief_basic_master_get_rx..... | 31 |
| 5.6.13 ccief_basic_master_set_ry..... | 31 |
| 5.6.14 ccief_basic_master_set_rww..... | 32 |
| 5.6.15 ccief_basic_master_get_rwr..... | 32 |
| 5.6.16 ccief_basic_master_get_pointer..... | 33 |
| 5.6.17 ccief_basic_master_set_unit_info..... | 33 |
| 5.6.18 ccief_basic_master_get_slave_info..... | 34 |
| 5.6.19 ccief_basic_master_get_group_info..... | 35 |
| 5.6.20 ccief_basic_master_check_parameter..... | 36 |
| 5.6.21 ccief_basic_master_recv..... | 36 |
| 5.6.22 ccief_basic_master_polling..... | 37 |
| 5.6.23 ccief_basic_master_execute_state..... | 37 |

| | |
|---|----|
| 5.6.24 ccief_basic_master_execute_state_wait_cyclic..... | 38 |
| 5.6.25 ccief_basic_master_execute_state_persuasion | 38 |
| 5.6.26 ccief_basic_master_execute_state_linkscan_end..... | 39 |
| 5.6.27 ccief_basic_master_execute_state_linkscan..... | 39 |
| 5.6.28 ccief_basic_master_persuasion_timer_timeout..... | 40 |
| 5.6.29 ccief_basic_master_cyclic_timer_timeout | 40 |
| 5.6.30 ccief_basic_master_make_cyclic_data..... | 41 |
| 5.6.31 ccief_basic_master_send_cyclic_data..... | 41 |
| 5.6.32 ccief_basic_master_rcv_cyclic_data_response | 42 |
| 5.6.33 ccief_basic_slaves_initialize | 43 |
| 5.6.34 ccief_basic_slaves_execute_state | 44 |
| 5.6.35 ccief_basic_slaves_execute_state_disconnect | 44 |
| 5.6.36 ccief_basic_slaves_execute_state_connecting..... | 45 |
| 5.6.37 ccief_basic_slaves_execute_state_cyclic_stop | 45 |
| 5.6.38 ccief_basic_slaves_execute_state_cyclic_end..... | 46 |
| 5.6.39 ccief_basic_slaves_execute_state_cyclic..... | 46 |
| 5.6.40 socket_initialize | 47 |
| 5.6.41 socket_terminate | 47 |
| 5.6.42 socket_rcv..... | 48 |
| 5.6.43 socket_send | 48 |
| 5.6.44 timer_initialize..... | 49 |
| 5.6.45 timer_terminate | 49 |
| 5.6.46 timer_main..... | 49 |
| 5.6.47 timer_start..... | 50 |
| 5.6.48 timer_stop | 50 |
| 5.6.49 timer_get_time | 51 |
| 5.6.50 timer_calculate_time_data..... | 51 |
| 5.6.51 timer_analyze_time_data..... | 52 |
| 5.6.52 timer_gettimeofday | 53 |
| 5.6.53 main..... | 54 |
| 5.6.54 user_callback_cyclic_link_scan_end..... | 54 |
| 5.6.55 user_parameter_file_read..... | 55 |
| 5.6.56 user_get_input_line..... | 55 |
| 5.6.57 user_show_menu_top | 56 |
| 5.6.58 user_input_check..... | 56 |
| 5.6.59 user_start_cyclic | 56 |
| 5.6.60 user_stop_cyclic | 57 |
| 5.6.61 user_start_application | 57 |
| 5.6.62 user_stop_application | 58 |
| 5.6.63 user_show_slave_info..... | 58 |

| | |
|------------------------------------|----|
| 5.6.64 user_show_master_info | 58 |
| 5.6.65 user_show_parameter | 59 |
| 5.6.66 user_get_adapter_info | 59 |
| 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 | 60 |
| 6.1 仕様 | 60 |
| 6.2 アプリケーションの作成 | 60 |
| 6.4 アプリケーションの実行 | 61 |

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 のマスタ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 | マスタ局 | CCIEF-BASIC マスタ局として、CCIEF-BASIC スレーブ局とサイクリック伝送を実施します。 |

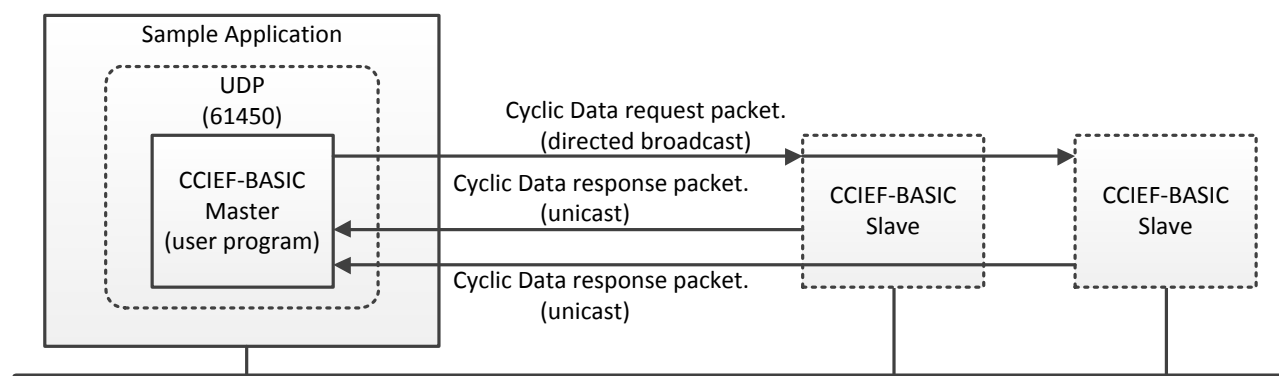


図 1. サンプルコードの機能

4 仕様

本サンプルコードにおける CCIEF-BASIC マスタ局の仕様を以下に示す。

表 2 CCIEF-BASIC マスタ局の仕様

| 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(送信時)、ユニキャスト(受信時) |
| 接続台数 | | マスタ局: 1 局 スレーブ局: 最大 64 局(1 台で複数局占有可能) |
| グループ数 | | 最大 64 グループ |
| 機能 | | サイクリック伝送 |
| サイクリックデータ | RX | 最大 4096 bit(全スレーブ局合計) |
| | RY | 最大 4096 bit(全スレーブ局合計) |
| | RWw | 最大 2048 word(全スレーブ局合計) |
| | RWr | 最大 2048 word(全スレーブ局合計) |
| 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。 |

本サンプルコードは、マスタ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

| 区分 | ID | パラメータ | 範囲 | 内容 |
|------------------------|----|-------------------------|---|----------------------------|
| Group (グループ 設定) | 1 | グループ設定個数 | 1～64 | グループの設定個数。 |
| | 2 | グループ番号 | 1～64 | グループ番号。 |
| | 3 | タイムアウト時間 | 1～65535 [ms] (0:500 [ms]) | リンクスキャンがタイムアウトする時間を指定。 |
| | 4 | 解列検出までの連続 タイムアウト回数 | 1～65535 回 (0:3 回) | 解列を検出するまでの連続タイムアウト回数を指定。 |
| | 5 | コンスタントリンク スキャンタイム | 1～2000 [ms] (0:コンスタントリンク スキャン未使用) | コンスタントリンクスキャンの時間を指定。 |
| | 6～ | ID2～5 を繰り返し設定 (可変個数) | 同左 | グループの設定個数分、ID2～5 を繰り返し設定。 |
| Slave (スレーブ局 設定) | 1 | スレーブ局設定個数 | 1～64 | スレーブ局の設定個数。 |
| | 2 | IP アドレス | 192.0.0.1～223.255.255.254 | スレーブ局の IP アドレスを設定。 |
| | 3 | 占有局数 | 1～16 | スレーブ局の占有局数を設定。 |
| | 4 | グループ番号 | 1～64 | スレーブ局のグループ番号を設定。 |
| | 5～ | ID2～4 を繰り返し設定 (可変個数) | 同左 | スレーブ局の設定個数分、ID2～4 を繰り返し設定。 |

パラメータ設定例を以下に示す。

[システム構成]

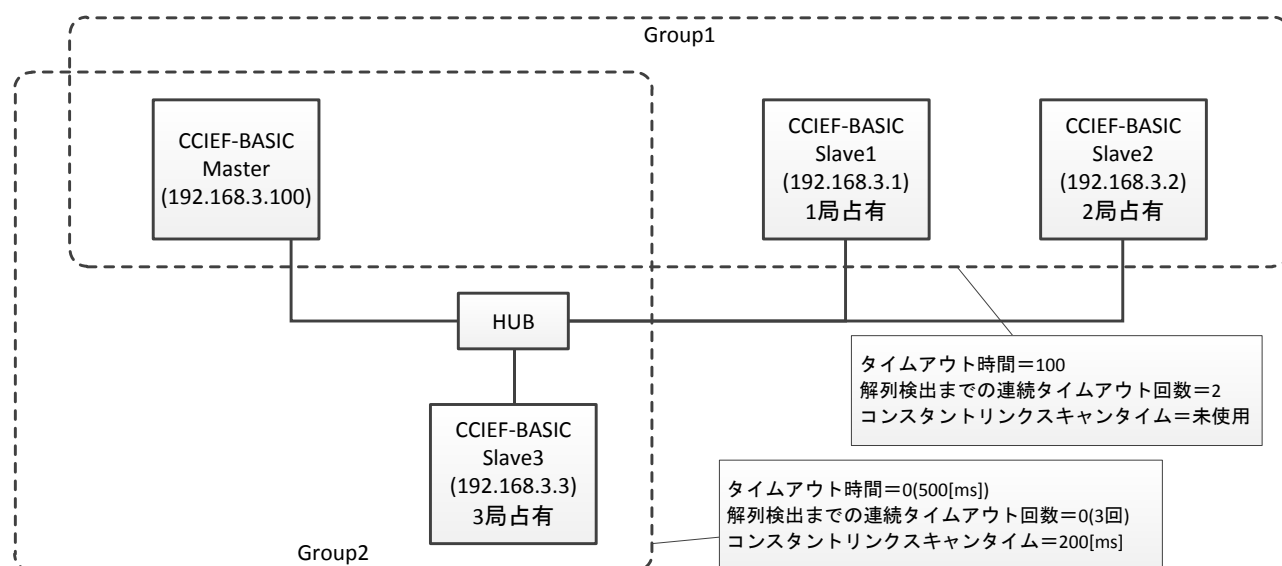


図 2. システム構成例

[MasterParameter.csv]

```
''
CCIEF-BASIC Master Sample Parameter,,
''
Group,,
ID, DATA, COMMENT
1, 2, Total number of group
2, 1, Number of group
3, 100, Group1 Cyclic transmission timeout
4, 2, Group1 Count of cyclic transmission timeout
5, 0, Group1 Constant link scan time
6, 2, Number of group
7, 0, Group2 Cyclic transmission timeout
8, 0, Group2 Count of cyclic transmission timeout
9, 200, Group2 Constant link scan time
''
Slave,,
ID, DATA, COMMENT
1, 3, Total number of slave
2, 192. 168. 3. 1, Slave1 IP address
3, 1, Slave1 Number of occupied stations
4, 1, Slave1 Number of group
5, 192. 168. 3. 2, Slave2 IP address
6, 2, Slave2 Number of occupied stations
7, 1, Slave2 Number of group
8, 192. 168. 3. 3, Slave3 IP address
9, 3, Slave3 Number of occupied stations
10, 2, Slave3 Number of group
```

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 4 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 4 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 のマスタ局としてサイクリック通信を行う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_MASTER.c、CCIEF_BASIC_MASTER.h、CCIEF_BASIC_SLAVES.c、CCIEF_BASIC_SLAVES.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_MASTER.h | ユーザープログラムヘッダ(マスタ局) |
| 6 | | | | CCIEF_BASIC_SLAVES.h | ユーザープログラムヘッダ(スレーブ局状態) |
| 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_MASTER.c | ユーザープログラム(マスタ局) |
| 11 | | | | CCIEF_BASIC_SLAVES.c | ユーザープログラム(スレーブ局状態) |
| 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 5 に示す。

表 5 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 ー プ ロ グ ラ ム | CCIEF_BA SIC_MAS TER.c | ccief_basic_master_initialize | int | CCIEF-BASIC マスタ局初期 化 | 公開 |
| 7 | | | ccief_basic_master_terminate | void | CCIEF-BASIC マスタ局終了 | 公開 |
| 8 | | | ccief_basic_master_main | int | CCIEF-BASIC マスタ局メイン 処理 | 公開 |
| 9 | | | ccief_basic_master_start_cyclic | int | サイクリック伝送開始 | 公開 |
| 10 | | | ccief_basic_master_stop_cyclic | int | サイクリック伝送停止 | 公開 |
| 11 | | | ccief_basic_master_get_rx | int | RX データ取得 | 公開 |
| 12 | | | ccief_basic_master_set_ry | int | RY データセット | 公開 |
| 13 | | | ccief_basic_master_set_rww | int | RWw データセット | 公開 |
| 14 | | | ccief_basic_master_get_rwr | int | RWr データ取得 | 公開 |
| 15 | | | ccief_basic_master_get_pointer | uint16_t * | デバイス先頭ポインタ取得 | 公開 |
| 16 | | | ccief_basic_master_set_unit_info | void | 自局ユニット情報セット | 公開 |
| 17 | | | ccief_basic_master_get_slave_info | int | スレーブ局受信情報取得 | 公開 |
| 18 | | | ccief_basic_master_get_group_info | int | グループ情報取得 | 公開 |
| 19 | | | ccief_basic_master_check_paramet er | int | パラメータチェック | 非公開 |
| 20 | | | ccief_basic_master_recv | Int | パケット受信 | 非公開 |
| 21 | | | ccief_basic_master_polling | Int | 定期実行 | 非公開 |
| 22 | | | ccief_basic_master_execute_state | void | マスタ局状態実行 | 非公開 |
| 23 | | | ccief_basic_master_execute_state_ wait_cyclic | void | マスタ局状態実行 (待機中) | 非公開 |
| 24 | | | ccief_basic_master_execute_state_p ersuasion | void | マスタ局状態実行 (マスタ局調停中) | 非公開 |
| 25 | | | ccief_basic_master_execute_state_li nkscan_end | void | マスタ局状態実行 (リンクスキャン完了) | 非公開 |
| 26 | | | ccief_basic_master_execute_state_li nkscan | void | マスタ局状態実行 (リンクスキャン中) | 非公開 |

| | | | | | | |
|----|--|----------------------|--|---------|--------------------------|-----|
| 27 | | | ccief_basic_master_persuasion_time_r_timeout | void | フレーム監視時間タイムアウト(コールバック関数) | 非公開 |
| 28 | | | ccief_basic_master_cyclic_timer_timeout | void | サイクリック伝送タイムアウト(コールバック関数) | 非公開 |
| 29 | | | ccief_basic_master_make_cyclic_data | int | サイクリック伝送データ作成 | 非公開 |
| 30 | | | ccief_basic_master_send_cyclic_data | int | サイクリック伝送データ送信 | 非公開 |
| 31 | | | ccief_basic_master_recv_cyclic_data_response | void | サイクリック伝送データ応答受信 | 非公開 |
| 32 | | CCIEF_BASIC_SLAVES.c | ccief_basic_slaves_initialize | void | スレーブ局状態初期化 | 公開 |
| 33 | | | ccief_basic_slaves_execute_state | void | スレーブ局状態実行 | 公開 |
| 34 | | | ccief_basic_slaves_execute_state_disconnect | void | スレーブ局状態実行(解列中) | 非公開 |
| 35 | | | ccief_basic_slaves_execute_state_connecting | void | スレーブ局状態実行(復列待ち) | 非公開 |
| 36 | | | ccief_basic_slaves_execute_state_cyclic_stop | void | スレーブ局状態実行(サイクリック停止中) | 非公開 |
| 37 | | | ccief_basic_slaves_execute_state_cyclic_end | void | スレーブ局状態実行(サイクリック伝送完了) | 非公開 |
| 38 | | | ccief_basic_slaves_execute_state_cyclic | void | スレーブ局状態実行(サイクリック伝送中) | 非公開 |
| 39 | | SOCKET.c | socket_initialize | int | ソケット初期化 | 公開 |
| 40 | | | socket_terminate | void | ソケット終了 | 公開 |
| 41 | | | socket_recv | int | パケット受信 | 公開 |
| 42 | | | socket_send | int | パケット送信 | 公開 |
| 43 | | TIMER.c | timer_initialize | void | タイマー初期化 | 公開 |
| 44 | | | timer_terminate | void | タイマー終了 | 公開 |
| 45 | | | timer_main | void | タイマーメイン処理 | 公開 |
| 46 | | | timer_start | int | タイマー開始 | 公開 |
| 47 | | | timer_stop | void | タイマー停止 | 公開 |
| 48 | | | timer_get_time | int64_t | 現在時間取得 | 公開 |
| 49 | | | timer_calculate_time_data | int64_t | 時計情報算出 | 公開 |
| 50 | | | timer_analyze_time_data | void | 時計情報解析 | 公開 |
| 51 | | | timer_gettimeofday | int | システム時間取得 | 公開 |
| 52 | | USER_SAMPLE.c | main | void | メイン処理 | 公開 |
| 53 | | | user_callback_cyclic_link_scan_end | void | リンクスキャン完了(コールバック関数) | 公開 |
| 54 | | | user_parameter_file_read | int | パラメータファイル読み出し | 非公開 |

| | | | | | | |
|----|--|--|------------------------|------|----------------|-----|
| 55 | | | user_get_input_line | void | 入力文字列取得 | 非公開 |
| 56 | | | user_show_menu_top | void | メニュー画面表示 | 非公開 |
| 57 | | | user_input_check | int | 入力チェック | 非公開 |
| 58 | | | user_start_cyclic | void | サイクリック伝送開始 | 非公開 |
| 59 | | | user_stop_cyclic | void | サイクリック伝送停止 | 非公開 |
| 60 | | | user_start_application | void | アプリケーション開始 | 非公開 |
| 61 | | | user_stop_application | void | アプリケーション停止 | 非公開 |
| 62 | | | user_show_slave_info | void | スレーブ局情報表示 | 非公開 |
| 63 | | | user_show_master_info | void | マスタ局情報表示 | 非公開 |
| 64 | | | user_show_parameter | void | パラメータ表示 | 非公開 |
| 65 | | | user_get_adapter_info | int | ネットワークアダプタ情報取得 | 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 6 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_MASTER.c | サイクリック通信を実施する。 |
| 2 | CCIEF_BASIC_SLAVES.c | スレーブ局ごとの個別状態の処理を実施する。 |
| 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 | マスタ局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込みを実施する。 コールバック関数を実装し、サイクリック伝送のリンクスキャン完了処理を実施する。 |

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 7 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_MASTER.c | ccief_basic_master_initialize | ソケット関数、構造体の実装方法。 |
| 2 | | ccief_basic_master_check_parameter | ソケット関数、構造体の実装方法。※1 |
| 3 | | ccief_basic_master_recv | ソケット関数、構造体の実装方法。 |
| 4 | | ccief_basic_master_execute_state_wait_cyclic | ソケット関数、構造体の実装方法。 |
| 5 | SOCKET.c | socket_initialize | ソケット関数、構造体の実装方法。※1 |
| 6 | | socket_terminate | ソケット関数、構造体の実装方法。 |
| 7 | | socket_recv | ソケット関数、構造体の実装方法。 |
| 8 | | socket_send | ソケット関数、構造体の実装方法。 |
| 9 | TIMER.c | timer_calculate_time_data | システム時間の取得方法。 |
| 10 | | timer_gettimeofday | システム時間の取得方法。 |
| 11 | USER_SAMPLE.c | main | ソケット関数、構造体の実装方法。 |
| 12 | | user_show_parameter | ソケット関数、構造体の実装方法。 |
| 13 | | user_get_adapter_info | ネットワークアダプタ情報の取得方法。 |

※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_MASTER.c

スレーブ局とサイクリック通信を実施する。マスタ局のグループ別の状態を処理する。

※詳細は、CC-Link IE フィールドネットワーク仕様書 (CC-Link IE フィールドネットワーク Basic-アプリケーション層プロトコル編)を参照。

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。

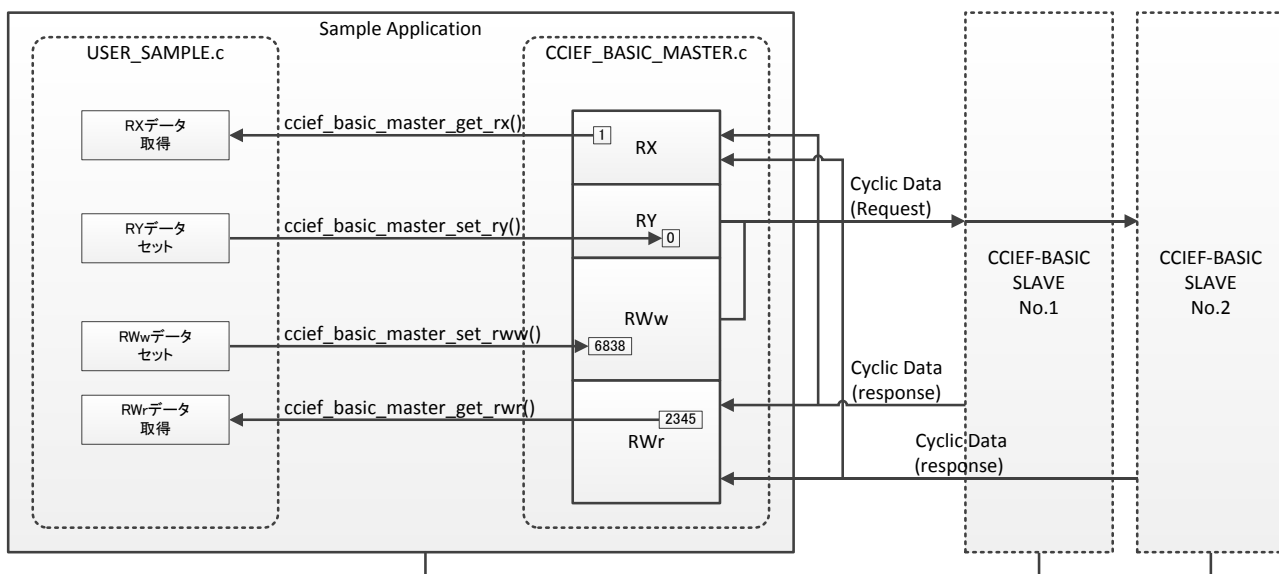


図 3. サイクリックデータのデバイスに直接アクセス

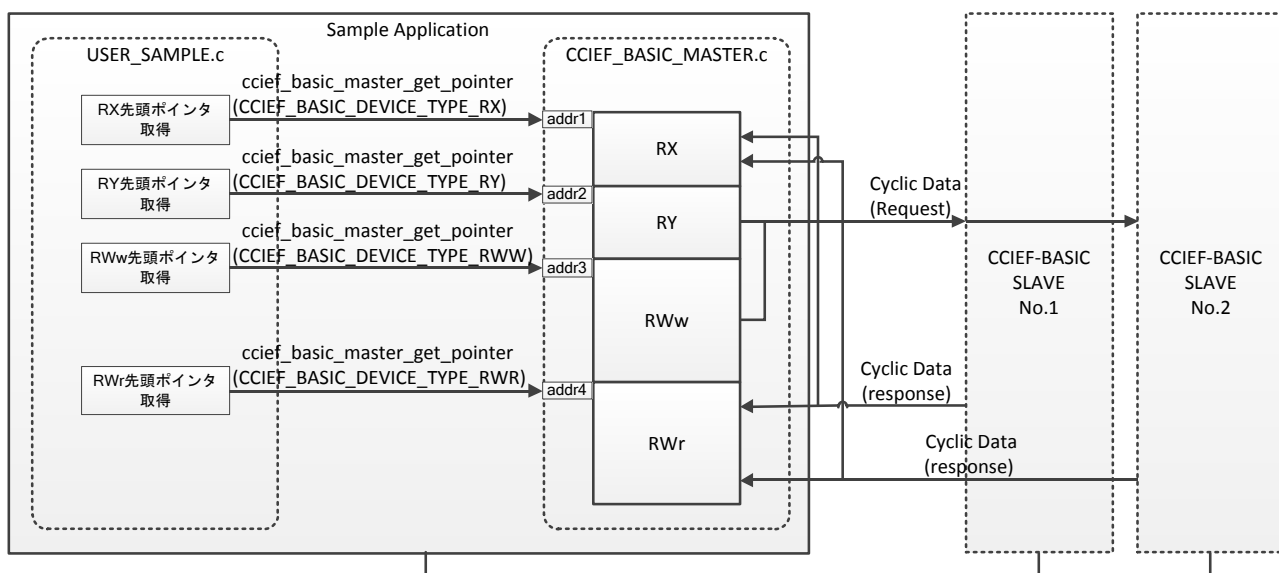


図 4. サイクリックデータの各デバイスの先頭ポインタを取得

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 8 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_MASTER_CALLBACK_CYC LIC_LINK_SCAN_END | マスタ局の状態がリンクスキャン完了に遷移する時に、実行される。 |

[CCIEF_BASIC_MASTER.h]

```
typedef void(*CCIEF_BASIC_MASTER_CALLBACK_CYCLIC_LINK_SCAN_END)(uint8_t ucGroupNumber);
```

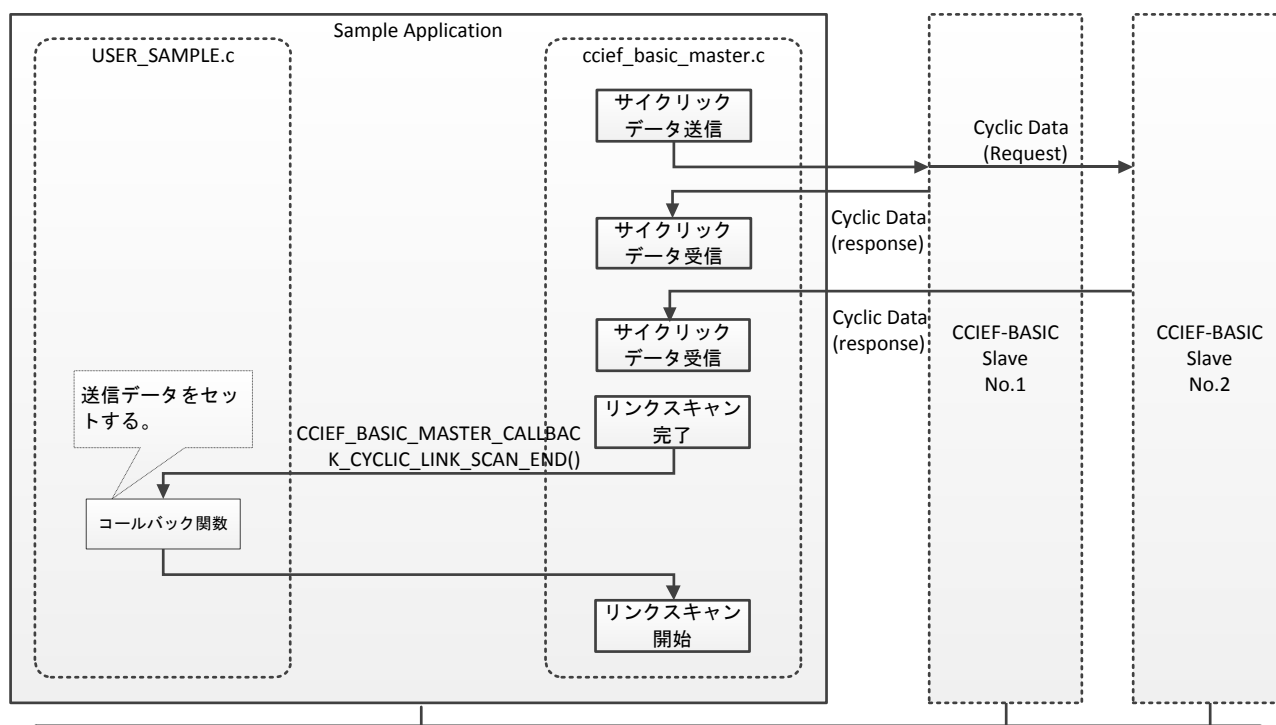


図 5. コールバック関数のイメージ

③ 自局ユニット情報

サンプルコードは、サイクリック伝送の要求データでスレーブ局へ通知する自局ユニット情報をユーザープログラムでセットできる。

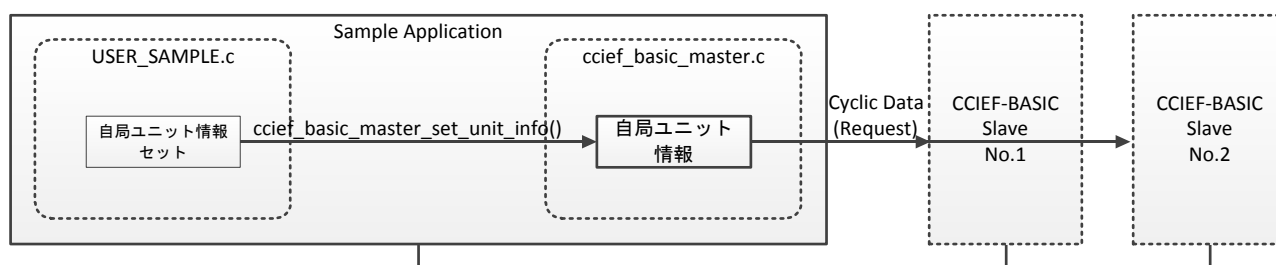


図 6. 自局ユニット情報を設定

④ スレーブ局受信情報取得

サンプルコードは、サイクリックデータで受信した各スレーブ局の情報をユーザープログラムで取得できる。

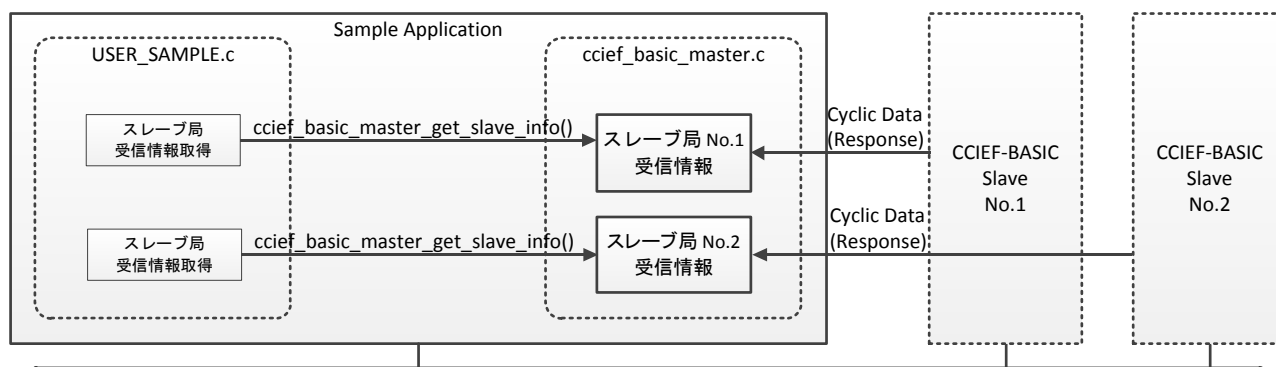


図 7. スレーブ局受信情報を取得

⑤ グループ情報取得

サンプルコードは、マスタ局の各グループ情報をユーザープログラムで取得できる。

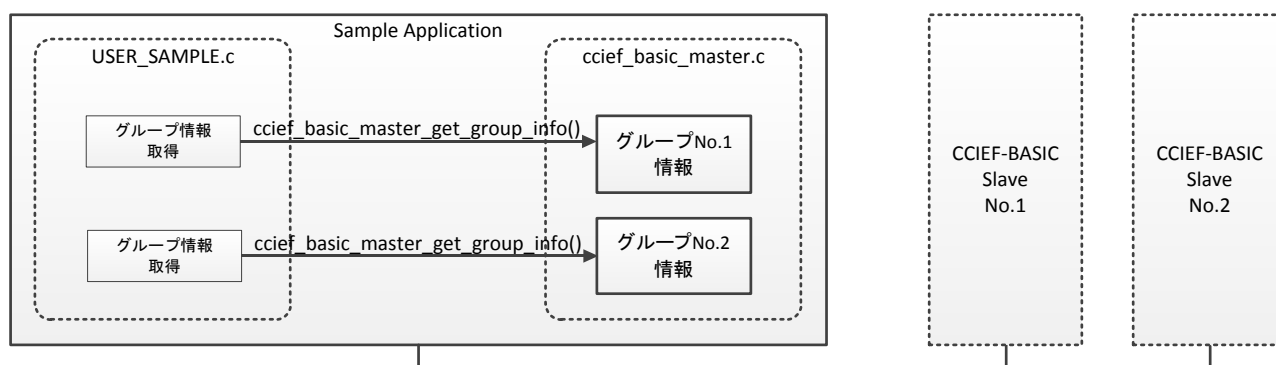


図 8. グループ情報を取得

(2) CCIEF_BASIC_SLAVES.c

スレーブ局ごとの個別状態の処理を実施する。

※詳細は、CC-Link IE フィールドネットワーク仕様書 (CC-Link IE フィールドネットワーク Basic-アプリケーション層プロトコル編)を参照。

(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 );
```

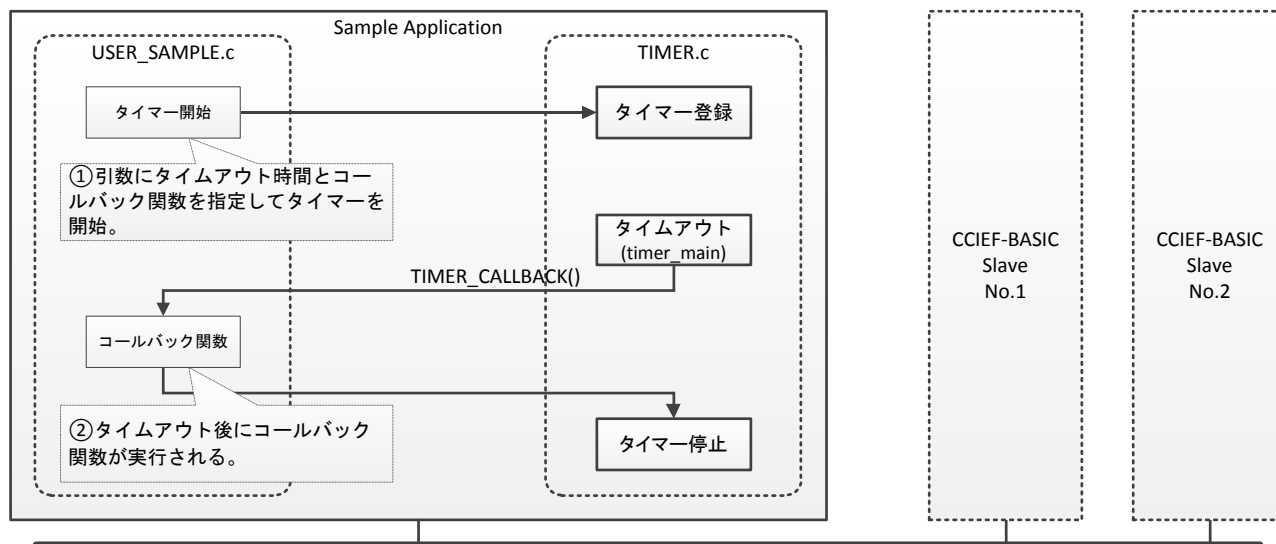


図 9. コールバック関数のイメージ

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。

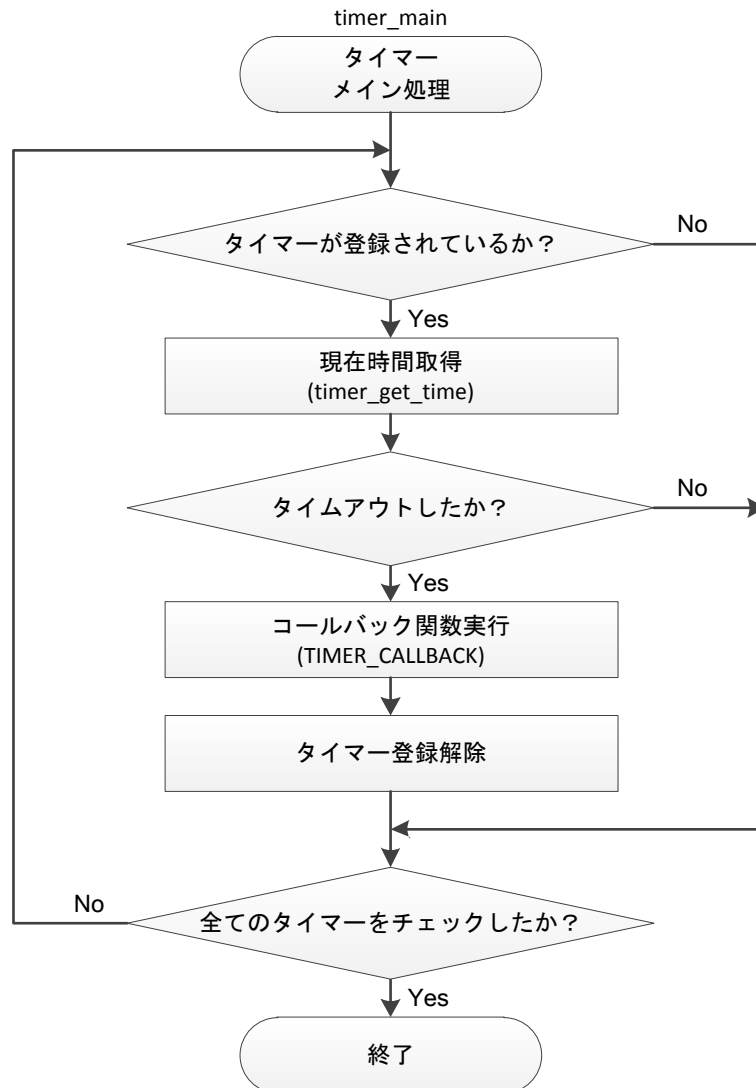


図 10. TIMER.c のフローチャート(サンプルプログラムの例)

(5) USER_SAMPLE.c

マスタ局、およびタイマー機能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出しを実施する。

コールバック関数を実装し、サイクリック伝送のリンクスキャン完了処理を実施する。

ユーザー操作により、サイクリック伝送の開始と停止、各種情報を画面に表示する。

① プログラム実装

サンプルプログラムは、CCIEF_BASIC_MASTER.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。

表 9 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_cyclic_link_scan_end | 指定されたグループ番号に属するスレーブ局の RY、および RWw に、送信データをセットする。(※1) | CCIEF_BASIC_MASTER_CALLBACK_CYCLIC_LINK_SCAN_END(表 8 参照) |

※1: ユーザーの実装先の環境に応じて、変更する。

② ユーザー操作

サンプルプログラムはアプリケーションの実行後、ユーザー操作(キー入力)で以下の処理を実行できる。

表 10 ユーザー操作

| No. | ユーザー操作 | 実行内容 |
|-----|----------|---|
| 1 | ‘1’ キー | スレーブ局のサイクリックを開始。※1 |
| 2 | ‘2’ キー | スレーブ局のサイクリックを停止。※1 |
| 3 | ‘3’ キー | アプリケーションを開始。 (マスタ局のサイクリックデバイス(RY、RWw)にデータをセット) |
| 4 | ‘4’ キー | アプリケーションを停止。 (マスタ局のサイクリックデバイス(RY、RWw)をクリア) |
| 5 | ‘5’ キー | スレーブ局情報を画面に表示。※1 |
| 6 | ‘6’ キー | マスタ局情報を画面に表示。※2 |
| 7 | ‘7’ キー | パラメータ設定内容を画面に表示。 |
| 8 | ‘Esc’ キー | アプリケーションの終了。 |

※1: 全てのスレーブ局が対象。

※2: 全てのグループが対象。

③ 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。

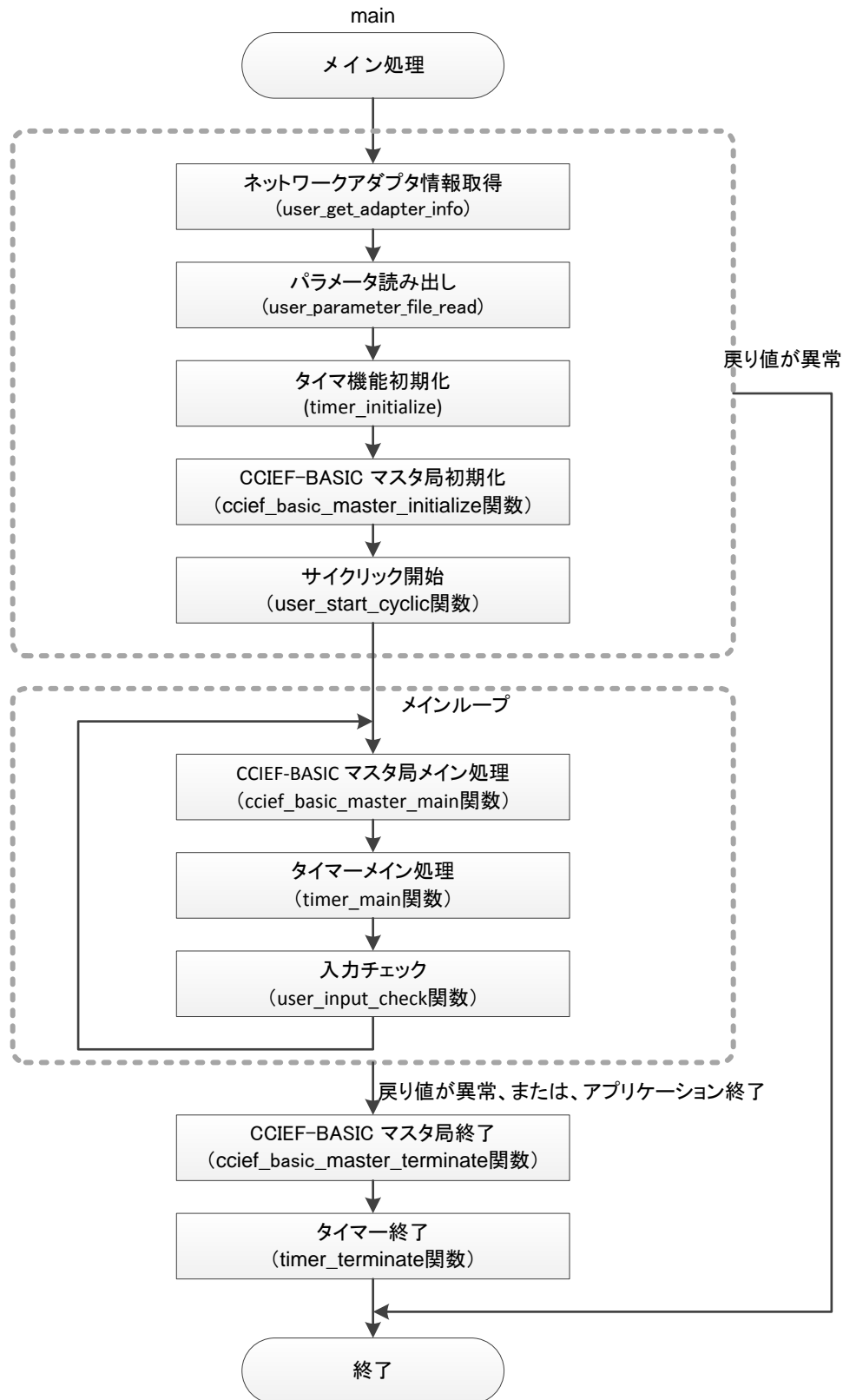


図 11. USER_SAMPLE.c のフローチャート(サンプルプログラムの例)

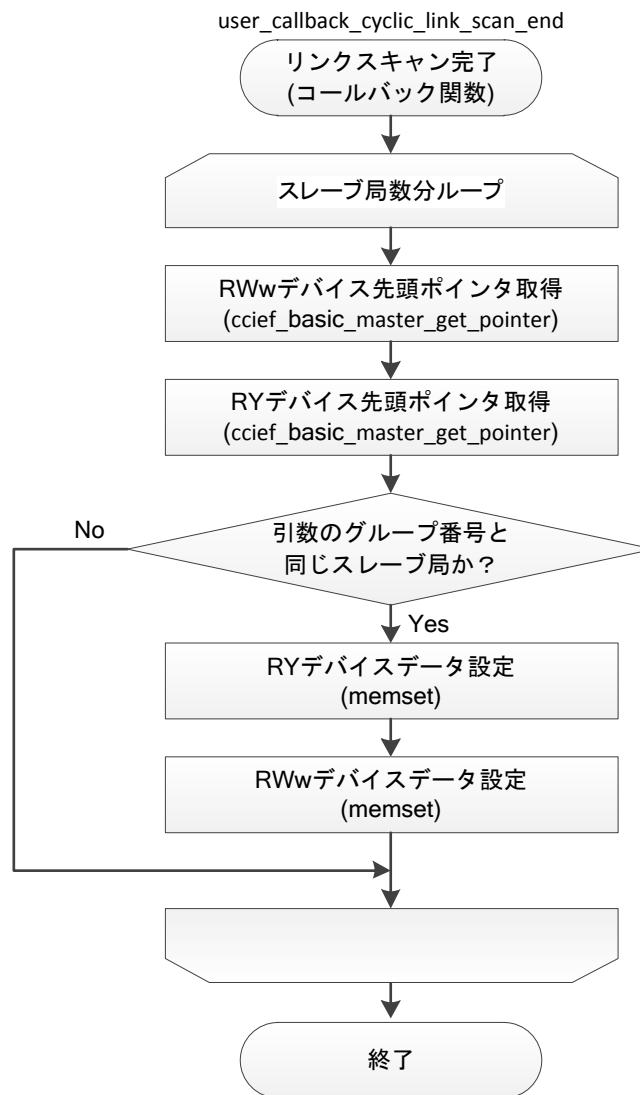


図 12. USER_SAMPLE.c のフローチャート 2(サンプルプログラムの例)

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 | (0xCCE0) |
| #define SLMP_ERR_REQ_DATA_SIZE | (0xCCE1) |
| #define SLMP_ERR_RES_DATA_SIZE | (0xCCE2) |
| #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_MASTER.h]

| | |
|---|--------|
| #define CCIEF_BASIC_MASTER_ERR_OK | 0 |
| #define CCIEF_BASIC_MASTER_ERR_NG | (-1) |
| #define CCIEF_BASIC_MASTER_ERR_DEVICE_RANGE | (-100) |
| #define CCIEF_BASIC_MASTER_ERR_MASTER_DUPLICATION | (-200) |
| #define CCIEF_BASIC_MASTER_ERR_SLAVE_DUPLICATION | (-300) |

[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) |
| #define USER_EXIT | 1 |

5.6.2 SLMP_MakePacketStream

表 11 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 12 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 13 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 14 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 15 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 |
| | 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_master_initialize

表 16 ccief_basic_master_initialize

| | | | | |
|--------|--|------------------------|-------------------------|-----|
| 機能 | CCIEF-BASIC マスタ局初期化 | | | |
| ファイル名 | CCIEF_BASIC_MASTER.c | | 公開/非公開 | 公開 |
| 呼び出し形式 | int ccief_basic_master_initialize (uint32_t ulIpAddress, uint32_t ulSubnetMask, CCIEF_BASIC_MASTER_PARAMETER *pParameter, CCIEF_BASIC_MASTER_CALLBACK_CYCLIC_LINK_SCAN_END pCyclicLinkScanEndFunc) | | | |
| 引数 | 型名 | 変数名 | 内容 | 入出力 |
| | uint32_t | ulIpAddress | マスタ局 IP アドレス | 入力 |
| | uint32_t | ulSubnetMask | マスタ局サブネットマスク | 入力 |
| | CCIEF_BASIC_MASTER_PARAMETER * | pParameter | マスタ局パラメータ | 入力 |
| | CCIEF_BASIC_MASTER_CALLBACK_CYCLIC_LINK_SCAN_END | pCyclicLinkScanEndFunc | コールバック関数 (リンクスキャン完了) | 入力 |
| 戻り値 | CCIEF_BASIC_MASTER_ERR_OK 正常 CCIEF_BASIC_MASTER_ERR_SOCKET ソケット異常 | | | |
| 説明 | マスタ局の初期化。 各変数の初期化。 ソケットの生成。 | | | |

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

[CCIEF_BASIC_MASTER.h]

```

typedef struct
{
    uint8_t    ucGroupNumber;                /* Group number */
    uint16_t   usCyclicTransmissionTimeout;  /* Cyclic transmission timeout */
    uint16_t   usCyclicTransmissionTimeoutCount; /* Count of cyclic transmission timeout */
    uint16_t   usConstantLinkScanTime;      /* Constant link scan time */
} CCIEF_BASIC_GROUP_PARAMETER;

typedef struct
{
    uint32_t   ulIpAddress;                  /* Slave ip address */
    uint16_t   usOccupiedStationNumber;      /* Number of occupied stations */
    uint8_t    ucGroupNumber;                /* Group number */
} CCIEF_BASIC_SLAVE_PARAMETER;

typedef struct
{
    int        iTotalGroupNumber;            /* Total number of the groups */
    CCIEF_BASIC_GROUP_PARAMETER Group[CCIEF_BASIC_MAX_GROUP_NUMBER]; /* Parameter of the groups */
    int        iTotalSlaveNumber;            /* Total number of the slaves */
    CCIEF_BASIC_SLAVE_PARAMETER Slave[CCIEF_BASIC_MAX_SLAVE_NUMBER]; /* Parameter of the slaves */
} CCIEF_BASIC_MASTER_PARAMETER;

```

5.6.8 ccief_basic_master_terminate

表 17 ccief_basic_master_terminate

| | | | | |
|--------|---|--------|----|-----|
| 機能 | CCIEF-BASIC マスタ局終了 | | | |
| ファイル名 | CCIEF_BASIC_MASTER.c | 公開/非公開 | 公開 | |
| 呼び出し形式 | void ccief_basic_master_terminate (void) | | | |
| 引数 | 型名 | 変数名 | 内容 | 入出力 |
| | - | - | - | - |
| 戻り値 | - | | | |
| 説明 | マスタ局の終了。 ソケットをクローズ。 | | | |

5.6.9 ccief_basic_master_main

表 18 ccief_basic_master_main

| | | | | |
|--------|--|--------|----|-----|
| 機能 | CCIEF-BASIC マスタ局メイン処理 | | | |
| ファイル名 | CCIEF_BASIC_MASTER.c | 公開/非公開 | 公開 | |
| 呼び出し形式 | int ccief_basic_master_main (void) | | | |
| 引数 | 型名 | 変数名 | 内容 | 入出力 |
| | - | - | - | - |
| 戻り値 | CCIEF_BASIC_MASTER_ERR_OK 正常 SOCKET_ERR_RECV ソケット異常 | | | |
| 説明 | パケット受信。 定期処理を実行。 | | | |

5.6.10 ccief_basic_master_start_cyclic

表 19 ccief_basic_master_start_cyclic

| | | | | |
|--------|--|--------------|----------|-----|
| 機能 | サイクリック伝送開始 | | | |
| ファイル名 | CCIEF_BASIC_MASTER.c | 公開/非公開 | 公開 | |
| 呼び出し形式 | int ccief_basic_master_start_cyclic (int iSlaveNumber) | | | |
| 引数 | 型名 | 変数名 | 内容 | 入出力 |
| | int | iSlaveNumber | スレーブ局 No | 入力 |
| 戻り値 | CCIEF_BASIC_MASTER_ERR_OK 正常 CCIEF_BASIC_MASTER_ERR_NG 異常 | | | |
| 説明 | 引数 iSlaveNumber で指定されたスレーブ局のサイクリック伝送を開始。 | | | |

5.6.11 ccief_basic_master_stop_cyclic

表 20 ccief_basic_master_stop_cyclic

| | | | | |
|--------|--|--------------|----------|-----|
| 機能 | サイクリック伝送停止 | | | |
| ファイル名 | CCIEF_BASIC_MASTER.c | 公開/非公開 | 公開 | |
| 呼び出し形式 | int ccief_basic_master_stop_cyclic (int iSlaveNumber) | | | |
| 引数 | 型名 | 変数名 | 内容 | 入出力 |
| | int | iSlaveNumber | スレーブ局 No | 入力 |
| 戻り値 | CCIEF_BASIC_MASTER_ERR_OK 正常 CCIEF_BASIC_MASTER_ERR_NG 異常 | | | |
| 説明 | 引数 iSlaveNumber で指定されたスレーブ局のサイクリック伝送を停止。 | | | |

5.6.12 ccief_basic_master_get_rx

表 21 ccief_basic_master_get_rx

| | | | | |
|--------|--|---------|---|-----|
| 機能 | RX データ取得 | | | |
| ファイル名 | CCIEF_BASIC_MASTER.c | 公開/非公開 | 公開 | |
| 呼び出し形式 | int ccief_basic_master_get_rx (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_MASTER_ERR_OK 正常 CCIEF_BASIC_MASTER_ERR_DEVICE_RANGE デバイス範囲異常 | | | |
| 説明 | 引数 iNumber で指定されたデバイス番号の RX データを取得。 | | | |

5.6.13 ccief_basic_master_set_ry

表 22 ccief_basic_master_set_ry

| | | | | |
|--------|--|---------|-----------------------------------|-----|
| 機能 | RY データセット | | | |
| ファイル名 | CCIEF_BASIC_MASTER.c | 公開/非公開 | 公開 | |
| 呼び出し形式 | int ccief_basic_master_set_ry (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_MASTER_ERR_OK 正常 CCIEF_BASIC_MASTER_ERR_NG 異常 CCIEF_BASIC_MASTER_ERR_DEVICE_RANGE デバイス範囲異常 | | | |
| 説明 | 引数 iNumber で指定されたデバイス番号の RY に、引数 iValue の値をセット。 | | | |

5.6.14 ccief_basic_master_set_rww

表 23 ccief_basic_master_set_rww

| | | | | |
|--------|--|---------|--------|-----|
| 機能 | RWw データセット | | | |
| ファイル名 | CCIEF_BASIC_MASTER.c | 公開/非公開 | 公開 | |
| 呼び出し形式 | int ccief_basic_master_set_rww (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_MASTER_ERR_OK 正常 CCIEF_BASIC_MASTER_ERR_DEVICE_RANGE デバイス範囲異常 | | | |
| 説明 | 引数 iNumber で指定されたデバイス番号の RWw に、引数 iValue の値をセット。 | | | |

5.6.15 ccief_basic_master_get_rwr

表 24 ccief_basic_master_get_rwr

| | | | | |
|--------|--|----------|------------|-----|
| 機能 | RWr データ取得 | | | |
| ファイル名 | CCIEF_BASIC_MASTER.c | 公開/非公開 | 公開 | |
| 呼び出し形式 | int ccief_basic_master_get_rwr (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_MASTER_ERR_OK 正常 CCIEF_BASIC_MASTER_ERR_DEVICE_RANGE デバイス範囲異常 | | | |
| 説明 | 引数 iNumber で指定されたデバイス番号の RWr データを取得。 | | | |

5.6.16 ccief_basic_master_get_pointer

表 25 ccief_basic_master_get_pointer

| | | | | |
|--------|---|-------------|--------|-----|
| 機能 | デバイス先頭ポインタ取得 | | | |
| ファイル名 | CCIEF_BASIC_MASTER.c | 公開/非公開 | 公開 | |
| 呼び出し形式 | uint16_t *ccief_basic_master_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_MASTER.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 */ |

5.6.17 ccief_basic_master_set_unit_info

表 26 ccief_basic_master_set_unit_info

| | | | | |
|--------|--|------------|--------------|-----|
| 機能 | 自局ユニット情報セット | | | |
| ファイル名 | CCIEF_BASIC_MASTER.c | | 公開/非公開 | 公開 |
| 呼び出し形式 | void ccief_basic_master_set_unit_info (uint16_t usUnitInfo) | | | |
| 引数 | 型名 | 変数名 | 内容 | 入出力 |
| | uint16_t | usUnitInfo | アプリケーション動作状態 | 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_MASTER.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 */ |

5.6.18 ccief_basic_master_get_slave_info

表 27 ccief_basic_master_get_slave_info

| | | | | |
|-------------|---|--------------|----------------------|-----|
| 機能 | スレーブ局受信情報取得 | | | |
| ファイル名 | CCIEF_BASIC_MASTER.c | 公開/非公開 | 公開 | |
| 呼 び 出 し 形 式 | int ccief_basic_master_get_slave_info (int iSlaveNumber, CCIEF_BASIC_SLAVE_INFO *pSlaveInfo) | | | |
| 引数 | 型名 | 変数名 | 内容 | 入出力 |
| | int | iSlaveNumber | スレーブ局 No | 入力 |
| | CCIEF_BASIC_SLAVE_INFO * | pSlaveInfo | スレーブ局受信情報 格納先ポインタ | 出力 |
| 戻り値 | CCIEF_BASIC_MASTER_ERR_OK 正常 CCIEF_BASIC_MASTER_ERR_NG 異常 | | | |
| 説明 | 引数 iSlaveNumber で指定されたスレーブ局の情報を、引数 pSlaveInfo に格納。 | | | |

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_MASTER.h]

```
typedef struct
{
    uint16_t                usVenderCode;                /* Vender code */
    uint16_t                usReserve1;                   /* Reserve */
    uint32_t                ulModelCode;                  /* Model code */
    uint16_t                usMachineVersion;             /* Machine version */
    uint16_t                usReserve2;                   /* Reserve */
    uint16_t                usUnitInfo;                   /* Information of the unit */
    uint16_t                usErrCode;                    /* Error code */
    uint32_t                ulUnitData;                   /* Data of the unit */
} CCIEF_BASIC_SLAVE_NOTIFY_INFO;

typedef struct
{
    uint16_t                usProtocolVersion;            /* Protocol version */
    uint16_t                usEndCode;                    /* Error code of the slave */
    uint32_t                ulId;                         /* Id of the slave */
    uint8_t                 ucGroupNumber;                /* Group number of the slave */
    uint16_t                usFrameSequenceNumber;       /* Frame sequence number */
    int                     usOccupiedStationNumber;     /* Number of occupied stations */
    int                     iState;                       /* State of the slave */
    CCIEF_BASIC_SLAVE_NOTIFY_INFO NotifyInfo;            /* Notify information from the slave */
    int                     iCyclicState;                 /* Cyclic state */
    int                     iStationNumber;               /* Number of stations */
} CCIEF_BASIC_SLAVE_INFO;
```

5.6.19 ccief_basic_master_get_group_info

表 28 ccief_basic_master_get_group_info

| | | | | |
|-------------|---|------------|-------------------|-----|
| 機能 | グループ情報取得 | | | |
| ファイル名 | CCIEF_BASIC_MASTER.c | 公開/非公開 | 公開 | |
| 呼 び 出 し 形 式 | int ccief_basic_master_get_group_info (int iGroupName, CCIEF_BASIC_GROUP_INFO *pGroupInfo) | | | |
| 引数 | 型名 | 変数名 | 内容 | 入出力 |
| | int | iGroupName | スレーブ局 No | 入力 |
| | CCIEF_BASIC_GROUP_INFO * | pGroupInfo | グループ情報 格納先ポインタ | 出力 |
| 戻り値 | CCIEF_BASIC_MASTER_ERR_OK 正常 CCIEF_BASIC_MASTER_ERR_NG 異常 | | | |
| 説明 | 引数 iSlaveNumber で指定されたマスタ局の情報を、引数 pGroupInfo に格納。 | | | |

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

[CCIEF_BASIC_MASTER.h]

| | | | |
|---------------------------|-------------------------------|---|--|
| typedef struct | | | |
| { | | | |
| uint16_t | usProtocolVersion; | /* Protocol version */ | |
| uint32_t | ulId; | /* Id of the master */ | |
| uint8_t | ucGroupName; | /* Group number of the slave */ | |
| int | iTotalSlaveNumber; | /* Total number of the slaves */ | |
| int | usTotalOccupiedStationNumber; | /* Total number of occupied stations */ | |
| int | iState; | /* State of Master */ | |
| uint16_t | usUnitInfo; | /* Information of the unit */ | |
| uint16_t | usFrameSequenceNumber; | /* Frame sequence number */ | |
| uint16_t | usParameterId; | /* Parameter id */ | |
| int64_t | llTimeData; | /* Data of time */ | |
| int64_t | llLinkScanTimeCurrent; | /* Current link scan time[us] */ | |
| int64_t | llLinkScanTimeMinimum; | /* Minimum link scan time[us] */ | |
| int64_t | llLinkScanTimeMaximum; | /* Maximum link scan time[us] */ | |
| } CCIEF_BASIC_GROUP_INFO; | | | |

5.6.20 ccief_basic_master_check_parameter

表 29 ccief_basic_master_check_parameter

| | | | | |
|--------|--|------------|-----------|-----|
| 機能 | パラメータチェック | | | |
| ファイル名 | CCIEF_BASIC_MASTER.c | 公開/非公開 | 非公開 | |
| 呼び出し形式 | int ccief_basic_master_check_parameter (CCIEF_BASIC_MASTER_PARAMETER *pParameter) | | | |
| 引数 | 型名 | 変数名 | 内容 | 入出力 |
| | CCIEF_BASIC_MASTER_PARAMETER * | pParameter | マスタ局パラメータ | 入力 |
| 戻り値 | CCIEF_BASIC_MASTER_ERR_OK 正常 CCIEF_BASIC_MASTER_ERR_NG 異常 | | | |
| 説明 | 引数 pParameter で指定されたマスタ局パラメータをチェック。 | | | |

5.6.21 ccief_basic_master_recv

表 30 ccief_basic_master_recv

| | | | | |
|--------|--|--------|-----|-----|
| 機能 | パケット受信 | | | |
| ファイル名 | CCIEF_BASIC_MASTER.c | 公開/非公開 | 非公開 | |
| 呼び出し形式 | int ccief_basic_master_recv (void) | | | |
| 引数 | 型名 | 変数名 | 内容 | 入出力 |
| | - | - | - | - |
| 戻り値 | CCIEF_BASIC_MASTER_ERR_OK 正常 SOCKET_ERR_RECV ソケット異常 CCIEF_BASIC_MASTER_ERR_MASTER_DUPLICATION マスタ局重複 | | | |
| 説明 | CCIEF-BASIC スレーブ局のパケット受信。 ユーザーの実装先の環境に応じてプログラムを書き換えること。 ※本関数は定期的に行われる必要がある。 | | | |

5.6.22 ccief_basic_master_polling

表 31 ccief_basic_master_polling

| | | | | |
|--------|--|--------|--------|-----|
| 機能 | 定期実行 | | | |
| ファイル名 | CCIEF_BASIC_MASTER.c | | 公開/非公開 | 非公開 |
| 呼び出し形式 | int ccief_basic_master_polling (CCIEF_BASIC_MASTER_GROUP_INFO *pGroup) | | | |
| 引数 | 型名 | 変数名 | 内容 | 入出力 |
| | CCIEF_BASIC_MASTER_GROUP_INFO | pGroup | グループ情報 | 入力 |
| | * | | | |
| 戻り値 | CCIEF_BASIC_MASTER_ERR_OK 正常 | | | |
| 説明 | 引数 pGroup で指定されたサイクリック伝送グループの定期処理。 リンクスキャン完了をチェック。 ※本関数は定期的に行われる必要がある。 | | | |

5.6.23 ccief_basic_master_execute_state

表 32 ccief_basic_master_execute_state

| | | | | |
|--------|---|--------|--------|-----|
| 機能 | マスタ局状態実行 | | | |
| ファイル名 | CCIEF_BASIC_MASTER.c | 公開/非公開 | 非公開 | |
| 呼び出し形式 | void ccief_basic_master_execute_state (CCIEF_BASIC_MASTER_GROUP_INFO *pGroup, int iEvent) | | | |
| 引数 | 型名 | 変数名 | 内容 | 入出力 |
| | CCIEF_BASIC_MASTER_GROUP_INFO * | pGroup | グループ情報 | 入力 |
| | int | iEvent | イベント | 入力 |
| 戻り値 | - | | | |
| 説明 | マスタ局状態を実行。 各状態別に処理を振り分ける。 ※詳細は、「CC-Link IE フィールドネットワーク仕様書 (CC-Link IE フィールドネットワーク Basic-アプリケーション層プロトコル編)」を参照。 | | | |

5.6.24 ccief_basic_master_execute_state_wait_cyclic

表 33 ccief_basic_master_execute_state_wait_cyclic

| | | | | |
|--------|--|--------|--------|-----|
| 機能 | マスタ局状態実行（待機中） | | | |
| ファイル名 | CCIEF_BASIC_MASTER.c | 公開/非公開 | 非公開 | |
| 呼び出し形式 | void ccief_basic_master_execute_state_wait_cyclic (CCIEF_BASIC_MASTER_GROUP_INFO *pGroup, int iEvent) | | | |
| 引数 | 型名 | 変数名 | 内容 | 入出力 |
| | CCIEF_BASIC_MASTER_GROUP_INFO * | pGroup | グループ情報 | 入力 |
| | int | iEvent | イベント | 入力 |
| 戻り値 | - | | | |
| 説明 | マスタ局状態が「待機中」の処理を実行。 ※詳細は、「CC-Link IE フィールドネットワーク仕様書 (CC-Link IE フィールドネットワーク Basic-アプリケーション層プロトコル編)」を参照。 | | | |

5.6.25 ccief_basic_master_execute_state_persuasion

表 34 ccief_basic_master_execute_state_persuasion

| | | | | |
|--------|--|--------|--------|-----|
| 機能 | マスタ局状態実行（マスタ局調停中） | | | |
| ファイル名 | CCIEF_BASIC_MASTER.c | 公開/非公開 | 非公開 | |
| 呼び出し形式 | void ccief_basic_master_execute_state_persuasion (CCIEF_BASIC_MASTER_GROUP_INFO *pGroup, int iEvent) | | | |
| 引数 | 型名 | 変数名 | 内容 | 入出力 |
| | CCIEF_BASIC_MASTER_GROUP_INFO * | pGroup | グループ情報 | 入力 |
| | int | iEvent | イベント | 入力 |
| 戻り値 | - | | | |
| 説明 | マスタ局状態が「マスタ局調停中」の処理を実行。 ※詳細は、「CC-Link IE フィールドネットワーク仕様書 (CC-Link IE フィールドネットワーク Basic-アプリケーション層プロトコル編)」を参照。 | | | |

5.6.26 ccief_basic_master_execute_state_linkscan_end

表 35 ccief_basic_master_execute_state_linkscan_end

| | | | | |
|--------|---|--------|--------|-----|
| 機能 | マスタ局状態実行（リンクスキャン完了） | | | |
| ファイル名 | CCIEF_BASIC_MASTER.c | 公開/非公開 | 非公開 | |
| 呼び出し形式 | void ccief_basic_master_execute_state_linkscan_end (CCIEF_BASIC_MASTER_GROUP_INFO *pGroup, int iEvent) | | | |
| 引数 | 型名 | 変数名 | 内容 | 入出力 |
| | CCIEF_BASIC_MASTER_GROUP_INFO * | pGroup | グループ情報 | 入力 |
| | int | iEvent | イベント | 入力 |
| 戻り値 | - | | | |
| 説明 | マスタ局状態が「リンクスキャン完了」の処理を実行。 ※詳細は、「CC-Link IE フィールドネットワーク仕様書（CC-Link IE フィールドネットワーク Basic-アプリケーション層プロトコル編）」を参照。 | | | |

5.6.27 ccief_basic_master_execute_state_linkscan

表 36 ccief_basic_master_execute_state_linkscan

| | | | | |
|--------|--|--------|--------|-----|
| 機能 | マスタ局状態実行（リンクスキャン中） | | | |
| ファイル名 | CCIEF_BASIC_MASTER.c | 公開/非公開 | 非公開 | |
| 呼び出し形式 | void ccief_basic_master_execute_state_linkscan (CCIEF_BASIC_MASTER_GROUP_INFO *pGroup, int iEvent) | | | |
| 引数 | 型名 | 変数名 | 内容 | 入出力 |
| | CCIEF_BASIC_MASTER_GROUP_INFO * | pGroup | グループ情報 | 入力 |
| | int | iEvent | イベント | 入力 |
| 戻り値 | － | | | |
| 説明 | マスタ局状態が「リンクスキャン中」の処理を実行。 ※詳細は、「CC-Link IE フィールドネットワーク仕様書（CC-Link IE フィールドネットワーク Basic-アプリケーション層プロトコル編）」を参照。 | | | |

5.6.28 ccief_basic_master_persuasion_timer_timeout

表 37 ccief_basic_master_persuasion_timer_timeout

| | | | | |
|--------|---|--------------|-------------------|-----|
| 機能 | フレーム監視時間タイムアウト(コールバック関数) | | | |
| ファイル名 | CCIEF_BASIC_MASTER.c | 公開/非公開 | 非公開 | |
| 呼び出し形式 | void ccief_basic_master_persuasion_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 | グループ情報 格納元ポインタ | 入力 |
| 戻り値 | - | | | |
| 説明 | フレーム監視時間のタイムアウト時に、タイマー機能からコールバック実行される。 「フレーム監視時間タイムアウト」イベントを発行。 | | | |

5.6.29 ccief_basic_master_cyclic_timer_timeout

表 38 ccief_basic_master_cyclic_timer_timeout

| | | | | |
|--------|---|--------------|-------------------|-----|
| 機能 | サイクリック伝送タイムアウト(コールバック関数) | | | |
| ファイル名 | CCIEF_BASIC_MASTER.c | | 公開/非公開 | 非公開 |
| 呼び出し形式 | void ccief_basic_master_cyclic_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 | グループ情報 格納元ポインタ | 入力 |
| 戻り値 | - | | | |
| 説明 | サイクリック伝送のタイムアウト時に、タイマー機能からコールバック実行される。 「リンクスキャン完了」イベントを発行。 | | | |

5.6.30 ccief_basic_master_make_cyclic_data

表 39 ccief_basic_master_make_cyclic_data

| | | | | |
|--------|--|--------|--------|-----|
| 機能 | サイクリック伝送データ作成 | | | |
| ファイル名 | CCIEF_BASIC_MASTER.c | 公開/非公開 | 非公開 | |
| 呼び出し形式 | int ccief_basic_master_make_cyclic_data (CCIEF_BASIC_MASTER_GROUP_INFO *pGroup) | | | |
| 引数 | 型名 | 変数名 | 内容 | 入出力 |
| | CCIEF_BASIC_MASTER_GROUP_INFO * | pGroup | グループ情報 | 入力 |
| 戻り値 | データサイズ(バイト) | | | |
| 説明 | サイクリック伝送の要求データを作成。 作成したデータサイズを戻り値にセット。 | | | |

5.6.31 ccief_basic_master_send_cyclic_data

表 40 ccief_basic_master_send_cyclic_data

| | | | | |
|--------|--|-------------|------------------|-----|
| 機能 | サイクリック伝送データ送信 | | | |
| ファイル名 | CCIEF_BASIC_MASTER.c | 公開/非公開 | 非公開 | |
| 呼び出し形式 | int ccief_basic_master_send_cyclic_data (uint32_t ullpAddress, uint8_t *pucData, int iDataSize) | | | |
| 引数 | 型名 | 変数名 | 内容 | 入出力 |
| | uint32_t | ullpAddress | グループ情報 | 入力 |
| | uint8_t * | pucData | 送信データ 格納元ポインタ | 入力 |
| | int | iDataSize | 送信データサイズ | 入力 |
| 戻り値 | CCIEF_BASIC_MASTER_ERR_OK 正常 | | | |
| 説明 | サイクリック伝送の要求データを送信。 | | | |

5.6.32 ccief_basic_master_recv_cyclic_data_response

表 41 ccief_basic_master_recv_cyclic_data_response

| | | | | |
|--------|---|---------|------------------|-----|
| 機能 | サイクリック伝送データ応答受信 | | | |
| ファイル名 | CCIEF_BASIC_MASTER.c | 公開/非公開 | 非公開 | |
| 呼び出し形式 | void ccief_basic_master_recv_cyclic_data_response (uint8_t *pucData) | | | |
| 引数 | 型名 | 変数名 | 内容 | 入出力 |
| | uint8_t * | pucData | 受信データ 格納元ポインタ | 入力 |
| 戻り値 | － | | | |
| 説明 | サイクリック伝送の応答データ受信。 | | | |

5.6.33 ccief_basic_slaves_initialize

表 42 ccief_basic_slaves_initialize

| | | | | |
|--------|---|--------|------------|-----|
| 機能 | スレーブ局状態初期化 | | | |
| ファイル名 | CCIEF_BASIC_SLAVES.c | 公開/非公開 | 公開 | |
| 呼び出し形式 | void ccief_basic_slaves_initialize (CCIEF_BASIC_SLAVES_CYCLIC_DATA_INFO *pSlave) | | | |
| 引数 | 型名 | 変数名 | 内容 | 入出力 |
| | CCIEF_BASIC_SLAVES_CYCLIC_DATA_INFO | pSlave | スレーブ局 | 入力 |
| | * | | サイクリック伝送情報 | |
| 戻り値 | - | | | |
| 説明 | スレーブ局状態を初期化。 | | | |

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

[CCIEF_BASIC_SLAVES.h]

```
typedef struct
{
    CCIEF_BASIC_SLAVE_PARAMETER *pParameter;          /* Parameter */
    int iNumber;                                       /* Slave Number */
    uint32_t ulId;                                    /* Id number */
    int iStationNumber;                               /* Number of stations */
    int iGroupStationNumber;                          /* Number of stations for the group */
    int iCyclicStart;                                 /* Start cyclic of the user operation */
    int iState;                                       /* State of Slave */
    int iCyclicState;                                 /* Cyclic state */
    int iCyclicStateSet;                             /* Setting of the cyclic state */
    int iReceiveComplete;                            /* State of response receive */
    int iDuplicateState;                              /* State of slave duplication */
    uint16_t *pusFrameSequenceNumber;                /* Frame sequence number of the master */
    uint16_t usProtocolVersion;                      /* Protocol version of the slave */
    uint16_t usEndCode;                              /* End code of the slave */
    uint16_t usFrameSequenceNumber;                  /* Frame sequence number of the slave */
    CCIEF_BASIC_SLAVE_NOTIFY_INFO NotifyInfo;        /* Notification information of the slave */
    uint16_t usCyclicTransmissionTimeoutCount;       /* Count of cyclic transmission timeout */
    uint16_t usTimeoutCount;                         /* Counter of timeout for the cyclic transmission timeout */
    uint16_t *pusRWw;                                /* Pointer of RWw for the packet */
    uint16_t *pusRY;                                 /* Pointer of RY for the packet */
    uint16_t *pusRWr;                                /* Pointer of RWr for the packet */
    uint16_t *pusRX;                                 /* Pointer of RX for the packet */
    uint16_t *pusSlaveRWr;                           /* Pointer of RWr for the slave */
    uint16_t *pusSlaveRX;                            /* Pointer of RX for the slave */
} CCIEF_BASIC_SLAVES_CYCLIC_DATA_INFO;
```

5.6.34 ccief_basic_slaves_execute_state

表 43 ccief_basic_slaves_execute_state

| | | | | |
|--------|--|--------|---------------------|-----|
| 機能 | スレーブ局状態実行 | | | |
| ファイル名 | CCIEF_BASIC_SLAVES.c | 公開/非公開 | 公開 | |
| 呼び出し形式 | void ccief_basic_slaves_execute_state (CCIEF_BASIC_SLAVES_CYCLIC_DATA_INFO *pSlave, int iEvent) | | | |
| 引数 | 型名 | 変数名 | 内容 | 入出力 |
| | CCIEF_BASIC_SLAVES_CYCLIC_DATA_INFO * | pSlave | スレーブ局 サイクリック伝送情報 | 入力 |
| | int | iEvent | イベント | 入力 |
| | | | | |
| 戻り値 | - | | | |
| 説明 | スレーブ局状態を実行。 各状態別に処理を振り分ける。 ※詳細は、「CC-Link IE フィールドネットワーク仕様書 (CC-Link IE フィールドネットワーク Basic-アプリケーション層プロトコル編)」を参照。 | | | |

5.6.35 ccief_basic_slaves_execute_state_disconnect

表 44 ccief_basic_slaves_execute_state_disconnect

| | | | | |
|--------|---|--------|------------|-----|
| 機能 | スレーブ局状態実行（解列中） | | | |
| ファイル名 | CCIEF_BASIC_SLAVES.c | 公開/非公開 | 非公開 | |
| 呼び出し形式 | void ccief_basic_slaves_execute_state_disconnect (CCIEF_BASIC_SLAVES_CYCLIC_DATA_INFO *pSlave, int iEvent) | | | |
| 引数 | 型名 | 変数名 | 内容 | 入出力 |
| | CCIEF_BASIC_SLAVES_CYCLIC_DATA_INFO | pSlave | スレーブ局 | 入力 |
| | * | | サイクリック伝送情報 | |
| | int | iEvent | イベント | 入力 |
| 戻り値 | － | | | |
| 説明 | スレーブ局状態が「解列中」の処理を実行。 ※詳細は、「CC-Link IE フィールドネットワーク仕様書（CC-Link IE フィールドネットワーク Basic-アプリケーション層プロトコル編）」を参照。 | | | |

5.6.36 ccief_basic_slaves_execute_state_connecting

表 45 ccief_basic_slaves_execute_state_connecting

| | | | | |
|--------|---|--------|---------------------|-----|
| 機能 | スレーブ局状態実行（復列待ち） | | | |
| ファイル名 | CCIEF_BASIC_SLAVES.c | 公開/非公開 | 非公開 | |
| 呼び出し形式 | void ccief_basic_slaves_execute_state_connecting (CCIEF_BASIC_SLAVES_CYCLIC_DATA_INFO *pSlave, int iEvent) | | | |
| 引数 | 型名 | 変数名 | 内容 | 入出力 |
| | CCIEF_BASIC_SLAVES_CYCLIC_DATA_INFO * | pSlave | スレーブ局 サイクリック伝送情報 | 入力 |
| | int | iEvent | イベント | 入力 |
| 戻り値 | － | | | |
| 説明 | スレーブ局状態が「復列待ち」の処理を実行。 ※詳細は、「CC-Link IE フィールドネットワーク仕様書（CC-Link IE フィールドネットワーク Basic-アプリケーション層プロトコル編）」を参照。 | | | |

5.6.37 ccief_basic_slaves_execute_state_cyclic_stop

表 46 ccief_basic_slaves_execute_state_cyclic_stop

| | | | | |
|--------|--|--------|---------------------|-----|
| 機能 | スレーブ局状態実行（サイクリック停止中） | | | |
| ファイル名 | CCIEF_BASIC_SLAVES.c | 公開/非公開 | 非公開 | |
| 呼び出し形式 | void ccief_basic_slaves_execute_state_cyclic_stop (CCIEF_BASIC_SLAVES_CYCLIC_DATA_INFO *pSlave, int iEvent) | | | |
| 引数 | 型名 | 変数名 | 内容 | 入出力 |
| | CCIEF_BASIC_SLAVES_CYCLIC_DATA_INFO * | pSlave | スレーブ局 サイクリック伝送情報 | 入力 |
| | int | iEvent | イベント | 入力 |
| 戻り値 | － | | | |
| 説明 | スレーブ局状態が「サイクリック停止中」の処理を実行。 ※詳細は、「CC-Link IE フィールドネットワーク仕様書（CC-Link IE フィールドネットワーク Basic-アプリケーション層プロトコル編）」を参照。 | | | |

5.6.38 ccief_basic_slaves_execute_state_cyclic_end

表 47 ccief_basic_slaves_execute_state_cyclic_end

| | | | | |
|--------|---|--------|------------|-----|
| 機能 | スレーブ局状態実行（サイクリック伝送完了） | | | |
| ファイル名 | CCIEF_BASIC_SLAVES.c | 公開/非公開 | 非公開 | |
| 呼び出し形式 | void ccief_basic_slaves_execute_state_cyclic_end (CCIEF_BASIC_SLAVES_CYCLIC_DATA_INFO *pSlave, int iEvent) | | | |
| 引数 | 型名 | 変数名 | 内容 | 入出力 |
| | CCIEF_BASIC_SLAVES_CYCLIC_DATA_INFO | pSlave | スレーブ局 | 入力 |
| | * | | サイクリック伝送情報 | |
| | int | iEvent | イベント | 入力 |
| 戻り値 | - | | | |
| 説明 | スレーブ局状態が「サイクリック伝送完了」の処理を実行。 ※詳細は、「CC-Link IE フィールドネットワーク仕様書（CC-Link IE フィールドネットワーク Basic-アプリケーション層プロトコル編）」を参照。 | | | |

5.6.39 ccief_basic_slaves_execute_state_cyclic

表 48 ccief_basic_slaves_execute_state_cyclic

| | | | | |
|--------|---|--------|---------------------|-----|
| 機能 | スレーブ局状態実行（サイクリック伝送中） | | | |
| ファイル名 | CCIEF_BASIC_SLAVES.c | 公開/非公開 | 非公開 | |
| 呼び出し形式 | void ccief_basic_slaves_execute_state_cyclic (CCIEF_BASIC_SLAVES_CYCLIC_DATA_INFO *pSlave, int iEvent) | | | |
| 引数 | 型名 | 変数名 | 内容 | 入出力 |
| | CCIEF_BASIC_SLAVES_CYCLIC_DATA_INFO * | pSlave | スレーブ局 サイクリック伝送情報 | 入力 |
| | int | iEvent | イベント | 入力 |
| 戻り値 | － | | | |
| 説明 | スレーブ局状態が「サイクリック伝送中」の処理を実行。 ※詳細は、「CC-Link IEフィールドネットワーク仕様書（CC-Link IE フィールドネットワーク Basic-アプリケーション層プロトコル編）」を参照。 | | | |

5.6.40 socket_initialize

表 49 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。 | | | |

5.6.41 socket_terminate

表 50 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。 | | | |

5.6.42 socket_recv

表 51 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。 | | | |

5.6.43 socket_send

表 52 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。 | | | |

5.6.44 timer_initialize

表 53 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。 | | | |

5.6.45 timer_terminate

表 54 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。 | | | |

5.6.46 timer_main

表 55 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。 | | | |

5.6.47 timer_start

表 56 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 (long lTime, int *piId, TIMER_CALLBACK pCallbackFunc, void *pCallbackArg) | | | |
| 引数 | 型名 | 変数名 | 内容 | 入出力 |
| | long | lTime | タイムアウト時間[ms] | 入力 |
| | int * | piId | 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 piId 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。 | | | |

5.6.48 timer_stop

表 57 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。 | | | |

5.6.49 timer_get_time

表 58 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 | int64_t timer_get_time (void) | | | |
| 引数 | 型名 | 変数名 | 内容 | 入出力 |
| | - | - | - | - |
| 戻り値 | 現在時間(システム時間)[us] | | | |
| 説明 | 現在のシステム時間を取得。 | | | |

5.6.50 timer_calculate_time_data

表 59 timer_calculate_time_data

| | | | | |
|--------|--|-----|--------|-----|
| 機能 | 時計情報算出 | | | |
| ファイル名 | TIMER.c | | 公開/非公開 | 公開 |
| 呼び出し形式 | int64_t timer_calculate_time_data (void) | | | |
| 引数 | 型名 | 変数名 | 内容 | 入出力 |
| | - | - | - | - |
| 戻り値 | 時計情報 | | | |
| 説明 | 現在のシステム時間から時計情報(UNIX 時間)を算出する。 ※時計情報は、現地時間から算出。 | | | |

5.6.51 timer_analyze_time_data

表 60 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 で指定された時計情報(UNIX 時間)を解析し、 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;
```

5.6.52 timer_gettimeofday

表 61 timer_gettimeofday

| | | | | |
|--------|--|-----|-----------------------|-----|
| 機能 | システム時間取得 | | | |
| ファイル名 | TIMER.c | | 公開/非公開 | 公開 |
| 呼び出し形式 | int timer_gettimeofday (struct timeval *tv, struct timezone *tz) | | | |
| 引数 | 型名 | 変数名 | 内容 | 入出力 |
| | struct timeval * | tv | システム時間[us] 格納先ポインタ | 出力 |
| | struct timezone * | tz | タイムゾーン 格納先ポインタ | 出力 |
| 戻り値 | TIMER_OK 正常 | | | |
| 説明 | システム時間[us]を取得する。 ユーザーの実装先の環境に応じて書き換えること。 ※サンプルコードは、POSIX 標準関数である gettimeofday の実装例を記述している。 | | | |

5.6.53 main

表 62 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。 パラメータの読み出し。 タイマー機能の初期化。 タイマー機能のメイン処理の実行。(ループ処理) ユーザーの実装先の環境に応じて書き換えること。 | | | |

5.6.54 user_callback_cyclic_link_scan_end

表 63 user_callback_cyclic_link_scan_end

| | | | | |
|--------|---|---------------|----------|-----|
| 機能 | リンクスキャン完了(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 | void user_callback_cyclic_link_scan_end (uint8_t ucGroupNumber) | | | |
| 引数 | 型名 | 変数名 | 内容 | 入出力 |
| | uint8_t | ucGroupNumber | グループ No. | 入力 |
| 戻り値 | - | | | |
| 説明 | マスタ局からのリンクスキャン完了時に実行される。 サイクリック伝送のグループが複数存在する場合は、 引数 ucGroupNumber のグループ番号毎に実行される。 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。 指定されたグループ番号に属するスレーブ局の RY、および RWw に スレーブ局 ID をセット。 | | | |

5.6.55 user_parameter_file_read

表 64 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, CCIEF_BASIC_MASTER_PARAMETER *pParameter) | | | |
| 引数 | 型名 | 変数名 | 内容 | 入出力 |
| | char * | file_path | ファイル | 入力 |
| | CCIEF_BASIC_MASTER_PARAMETER * | pParameter | マスタ局パラメータ 格納先ポインタ | 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。 ユーザーの実装先の環境に応じて書き換えること。 | | | |

5.6.56 user_get_input_line

表 65 user_get_input_line

| | | | | |
|--------|--|-------------|------------|-----|
| 機能 | 入力文字列取得 | | | |
| ファイル名 | USER_SAMPLE.c | 公開/非公開 | 非公開 | |
| 呼び出し形式 | void user_get_input_line (char *pcLine, int iLineLength) | | | |
| 引数 | 型名 | 変数名 | 内容 | 入出力 |
| | char * | pcLine | 文字列格納先ポインタ | 出力 |
| | int | iLineLength | 最大入力文字数 | 入力 |
| 戻り値 | - | | | |
| 説明 | ユーザーが入力した文字列を取得。 (Enter キー押下で入力終了) ユーザーの実装先の環境に応じて書き換えること。 | | | |

5.6.57 user_show_menu_top

表 66 user_show_menu_top

| | | | | |
|--------|---------------------------------------|--------|-----|-----|
| 機能 | メニュー画面表示 | | | |
| ファイル名 | USER_SAMPLE.c | 公開/非公開 | 非公開 | |
| 呼び出し形式 | void user_show_menu_top (void) | | | |
| 引数 | 型名 | 変数名 | 内容 | 入出力 |
| | - | - | - | - |
| 戻り値 | - | | | |
| 説明 | メニュー画面を表示。 ユーザーの実装先の環境に応じて書き換えること。 | | | |

5.6.58 user_input_check

表 67 user_input_check

| | | | | |
|--------|---|--------|-----|-----|
| 機能 | 入力チェック | | | |
| ファイル名 | USER_SAMPLE.c | 公開/非公開 | 非公開 | |
| 呼び出し形式 | int user_input_check (void) | | | |
| 引数 | 型名 | 変数名 | 内容 | 入出力 |
| | - | - | - | - |
| 戻り値 | USER_ERR_OK 正常 USER_EXIT 終了 | | | |
| 説明 | メニュー画面のユーザーのキー入力をチェック。 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。 | | | |

5.6.59 user_start_cyclic

表 68 user_start_cyclic

| | | | | |
|--------|---|--------|-----|-----|
| 機能 | サイクリック伝送開始 | | | |
| ファイル名 | USER_SAMPLE.c | 公開/非公開 | 非公開 | |
| 呼び出し形式 | void user_start_cyclic (void) | | | |
| 引数 | 型名 | 変数名 | 内容 | 入出力 |
| | － | － | － | － |
| 戻り値 | － | | | |
| 説明 | サイクリック伝送を開始。 (パラメータ設定されている、全てのスレーブ局のサイクリック伝送を開始。) ユーザーの実装先の環境に応じて書き換えること。 | | | |

5.6.60 user_stop_cyclic

表 69 user_stop_cyclic

| | | | | |
|--------|---|--------|-----|-----|
| 機能 | サイクリック伝送停止 | | | |
| ファイル名 | USER_SAMPLE.c | 公開/非公開 | 非公開 | |
| 呼び出し形式 | void user_stop_cyclic (void) | | | |
| 引数 | 型名 | 変数名 | 内容 | 入出力 |
| | － | － | － | － |
| 戻り値 | － | | | |
| 説明 | サイクリック伝送を停止。 （パラメータ設定されている、全てのスレーブ局のサイクリック伝送を停止。） 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_start_application

表 70 user_start_application

| | | | | |
|--------|---|--------|-----|-----|
| 機能 | アプリケーション開始 | | | |
| ファイル名 | USER_SAMPLE.c | 公開/非公開 | 非公開 | |
| 呼び出し形式 | void user_start_application (void) | | | |
| 引数 | 型名 | 変数名 | 内容 | 入出力 |
| | - | - | - | - |
| 戻り値 | - | | | |
| 説明 | アプリケーションを開始。 ユーザーの実装先の環境に応じて書き換えること。 | | | |

5.6.62 user_stop_application

表 71 user_stop_application

| | | | | |
|--------|---|--------|-----|-----|
| 機能 | アプリケーション停止 | | | |
| ファイル名 | USER_SAMPLE.c | 公開/非公開 | 非公開 | |
| 呼び出し形式 | void user_stop_application (void) | | | |
| 引数 | 型名 | 変数名 | 内容 | 入出力 |
| | - | - | - | - |
| 戻り値 | - | | | |
| 説明 | アプリケーションを停止。 ユーザーの実装先の環境に応じて書き換えること。 | | | |

5.6.63 user_show_slave_info

表 72 user_show_slave_info

| | | | | |
|--------|--|--------|----|-----|
| 機能 | スレーブ局情報表示 | | | |
| ファイル名 | USER_SAMPLE.c | 公開/非公開 | | 非公開 |
| 呼び出し形式 | void user_show_slave_info (void) | | | |
| 引数 | 型名 | 変数名 | 内容 | 入出力 |
| | - | - | - | - |
| 戻り値 | - | | | |
| 説明 | 画面にスレーブ局情報を表示。 (パラメータで設定されている、全てのスレーブ局の情報を表示。) ユーザーの実装先の環境に応じて書き換えること。 | | | |

5.6.64 user_show_master_info

表 73 user_show_master_info

| | | | | |
|--------|--|--------|-----|-----|
| 機能 | マスタ局情報表示 | | | |
| ファイル名 | USER_SAMPLE.c | 公開/非公開 | 非公開 | |
| 呼び出し形式 | void user_show_master_info (void) | | | |
| 引数 | 型名 | 変数名 | 内容 | 入出力 |
| | - | - | - | - |
| 戻り値 | - | | | |
| 説明 | 画面にマスタ局情報を表示。 ユーザーの実装先の環境に応じて書き換えること。 | | | |

5.6.65 user_show_parameter

表 74 user_show_parameter

| | | | | |
|--------|---|--------|-----|-----|
| 機能 | パラメータ表示 | | | |
| ファイル名 | USER_SAMPLE.c | 公開/非公開 | 非公開 | |
| 呼び出し形式 | void user_show_parameter (void) | | | |
| 引数 | 型名 | 変数名 | 内容 | 入出力 |
| | - | - | - | - |
| 戻り値 | - | | | |
| 説明 | 画面にマスタ局パラメータを表示。 ユーザーの実装先の環境に応じて書き換えること。 | | | |

5.6.66 user_get_adapter_info

表 75 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、WindowsOS 上のネットワークアダプタの取得例を記述している。 | | | |

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 11 に示します。

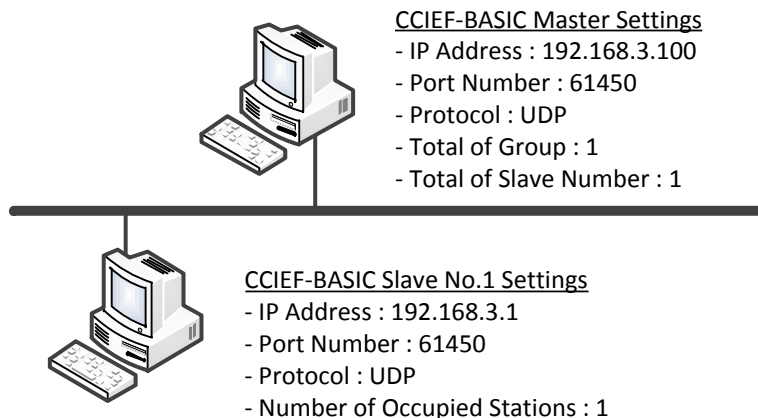


図 11. サンプルコードの実行環境

サンプルコードでは、CCIEF-BASIC スレーブ局(サンプルアプリケーション※1)とサイクリック通信を実行します。

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。

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 スレーブ局編」を参照ください。

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_Master」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12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_Master/library/include/ -c /home/ユーザー/CCIEF-BASIC_Master/library/src/SLMP.c  
[ユーザー@localhost ~]$ gcc -I /home/ユーザー/CCIEF-BASIC_Master/sample/include/ -c /home/ユーザー/CCIEF-BASIC_Master/sample/src/SOCKET.c  
[ユーザー@localhost ~]$ gcc -I /home/ユーザー/CCIEF-BASIC_Master/sample/include/ -c /home/ユーザー/CCIEF-BASIC_Master/sample/src/TIMER.c  
[ユーザー@localhost ~]$ gcc -I /home/ユーザー/CCIEF-BASIC_Master/library/include/ -I /home/ユーザー/CCIEF-BASIC_Master/sample/include/ -c  
/home/ユーザー/CCIEF-BASIC_Master/sample/src/CCIEF_BASIC_MASTER.c  
[ユーザー@localhost ~]$ gcc -I /home/ユーザー/CCIEF-BASIC_Master/library/include/ -I /home/ユーザー/CCIEF-BASIC_Master/sample/include/ -c  
/home/ユーザー/CCIEF-BASIC_Master/sample/src/CCIEF_BASIC_SLAVES.c  
[ユーザー@localhost ~]$ gcc -I /home/ユーザー/CCIEF-BASIC_Master/library/include/ -I /home/ユーザー/CCIEF-BASIC_Master/sample/include/ -c  
/home/ユーザー/CCIEF-BASIC_Master/sample/src/USER_SAMPLE.c  
[ユーザー@localhost ~]$ gcc SLMP.o CCIEF_BASIC_SLAVES.o CCIEF_BASIC_MASTER.o SOCKET.o TIMER.o USER_SAMPLE.o -o Master_sample  
[ユーザー@localhost ~]$
```

図12 コンパイルコマンド

6.3 アプリケーションの実行

アプリケーションを実行するまでの手順を示します。

- (1) マスタ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。

[MasterParameter.csv]

```
,,
CCIEF-BASIC Master Sample Parameter,,
,,
Group,,
ID, DATA, COMMENT
1, 1, Total number of group
2, 1, Number of group
3, 100, Group1 Cyclic transmission timeout
4, 2, Group1 Count of cyclic transmission timeout
5, 0, Group1 Constant link scan time
,,
Slave,,
ID, DATA, COMMENT
1, 1, Total number of slave
2, 192.168.3.1, Slave1 IP address
3, 1, Slave1 Number of occupied stations
4, 1, Slave1 Number of group
```

- (2) ルートユーザでない場合はコマンドで「sudo ./Master_sample MasterParameter.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「./Master_sample MasterParameter.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 スレーブ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。

- (3) アプリケーションが正常に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 Master Station!

Show master parameter!

Master:
  IP Address:          192.168.3.100 (Master ID:0xC0A80364)
  Subnet mask:         255.255.255.0
  Default GW IP address: 192.168.3.254
```

```
Total Number of Group:          1
Group No.1:
Disconnection Time[ms]:         100 (0:500[ms])
Disconnection Timeout Count:    2 (0:3)
Constant Link Scan Time[ms]:    Not use

Total Number of Slave:          1
Slave No.1:
IP Address:                     192.168.3.1 (Slave ID:0xC0A80301)
Occupied Station Number:        1
Group Number:                   1

Start cyclic of all the slaves!

Start the application!

Please input the followng key values if you want any action.

'1' - Start the cyclic.
'2' - Stop the cyclic.
'3' - Start the application.
'4' - Stop the application.
'5' - Show information of the slave.
'6' - Show information of the master.
'7' - Show the parameter.
'Esc' - Exit the application.
```

図 2. マスタ局起動画面

(4) アプリケーションのユーザー操作例を、以下に示します。

- ① '1' キーを押下。

```
Start cyclic of all the slaves!
```

図 3. '1' キーを押下

- ② '2' キーを押下。

```
Stop cyclic of all the slaves!
```

図 4. '2' キーを押下

- ③ '3' キーを押下。

```
Start the application!
```

図 5. '3' キーを押下

- ④ '4' キーを押下。

```
Stop the application!
```

図 6. '4' キーを押下

- ⑤ '5' キーを押下。

```
Show the state of all the slaves!

Slave No.1:
```

```

Slave ID: 0xC0A80301
Occupied Station Number: 1
Group No.: 1
State: 5 [CYCLIC]
Protocol Version: 0x0001
Error Code: 0x0000
Slave Notify Information:
  Vender Code: 0x1234
  Model Code: 0x00010001
  Machine Version: 0x0001
  Unit Info: 0x0001
  Error Code: 0x0000
  Unit Data: 0x00000000
Frame sequence number: 0x0385

```

| RX | F | E | D | C | B | A | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | data |
|------|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|--------|
| 0000 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | | 0x0386 |
| 0010 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | | 0x0386 |
| 0020 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | | 0x0386 |
| 0030 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | | 0x0386 |

| RY | F | E | D | C | B | A | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | data |
|------|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|--------|
| 0000 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | | 0x0387 |
| 0010 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | | 0x0387 |
| 0020 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | | 0x0387 |
| 0030 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | | 0x0387 |

| RWw | | +7 | +6 | +5 | +4 | +3 | +2 | +1 | +0 |
|------|--|--------|--------|--------|--------|--------|--------|--------|--------|
| 0000 | | 0x0387 | 0x0387 | 0x0387 | 0x0387 | 0x0387 | 0x0387 | 0x0387 | 0x0387 |
| 0008 | | 0x0387 | 0x0387 | 0x0387 | 0x0387 | 0x0387 | 0x0387 | 0x0387 | 0x0387 |
| 0010 | | 0x0387 | 0x0387 | 0x0387 | 0x0387 | 0x0387 | 0x0387 | 0x0387 | 0x0387 |
| 0018 | | 0x0387 | 0x0387 | 0x0387 | 0x0387 | 0x0387 | 0x0387 | 0x0387 | 0x0387 |

| RWr | | +7 | +6 | +5 | +4 | +3 | +2 | +1 | +0 |
|------|--|--------|--------|--------|--------|--------|--------|--------|--------|
| 0000 | | 0x0386 | 0x0386 | 0x0386 | 0x0386 | 0x0386 | 0x0386 | 0x0386 | 0x0386 |
| 0008 | | 0x0386 | 0x0386 | 0x0386 | 0x0386 | 0x0386 | 0x0386 | 0x0386 | 0x0386 |
| 0010 | | 0x0386 | 0x0386 | 0x0386 | 0x0386 | 0x0386 | 0x0386 | 0x0386 | 0x0386 |
| 0018 | | 0x0386 | 0x0386 | 0x0386 | 0x0386 | 0x0386 | 0x0386 | 0x0386 | 0x0386 |

図 7. '5' キーを押下

⑥ '6' キーを押下。

Show the state of master!

```

Master:
  Protocol Version: 0x0001
  Master ID: 0xC0A80364
  Unit Info: 0x0001
  Parameter ID: 0x6C32

Group No. 1:
  Total Number of Slave: 1
  Total Number of Occupied Station: 1
  State: 4 [LINK_SCAN_END]
  Time Data: 1459868384442 [2016-03-31 14:59:44.442]

```

```

Frame sequence number:      0x060E
Link scan time(Current):    1.133[ms]
Link scan time(Minimum):    0.964[ms]
Link scan time(Maximum):    3.046[ms]
Group:
  Master(ID:100)
    |--- Slave No.1 (ID:0xC0A80301 CyclicState:ON State:5 [CYCLIC])

```

図 8. '6' キーを押下

⑦ '7' キーを押下。

```

Show master parameter!

Master:
  IP Address:      192.168.3.100 (Master ID:0xC0A80364)
  Subnet mask:     255.255.255.0
  Default GW IP address: 192.168.3.254
  Total Number of Group: 1
  Group No.1:
    Disconnection Time[ms]: 100 (0:500[ms])
    Disconnection Timeout Count: 2 (0:3)
    Constant Link Scan Time[ms]: Not use
  Total number of slave: 1
  Slave No.1:
    IP Address:      192.168.3.1 (Slave ID:0xC0A80301)
    Occupied Station Number: 1
    Group Number:    1

```

図 9. '7' キーを押下

⑧ 'Esc' キーを押下。

```

Exit the application? (if you want exit, please press 'Y')

Application has exited. (please any press)

```

図 10. 'Esc' キーを押下