

## 第5部 ECHONET 共通下位通信インタフェース仕様



	変更部位 (目次項目)	追加・変更概要
7	4.1	・第2部状態遷移の改定にともない、以下のAPIを追加。 「ClcInitAll」、「ClcStop」、「ClcHalt」、 「ClcLowInitAll」、「ClcLowStop」、 「ClcLowHalt」
8	4.1	・第2部状態遷移の改定にともない、 「ClcReset」を「ClcStart」に「ClcLowReset」 を「ClcLowStart」名称変更。
9	4.2	・第2部状態遷移の改定にともない、APIの詳細説明を変更。

・Version2.11

2002年4月26日 コンソーシアム会員内公開。  
 変更のある項目は以下の通り

	変更部位 (目次項目)	追加・変更概要
1	4.2.2	使用する構造体のバッファ sbuf と rbuf の型を short から unsigned char* に変更
2	4.2.20	戻り値に CLC_ADAPTER_ERROR(4)を追記
3	4.2.21	戻り値に CLC_ADAPTER_ERROR(4)を追記
4	4.2.22	戻り値に CLC_ADAPTER_ERROR(4)を追記
5	4.2.24	構文の引数 node_id の型を unsigned char から unsigned char*に変更

• Version3.00Draft

2002年6月12日 コンソーシアム会員内公開。  
 変更のある項目は以下の通り

	変更部位 (目次項目)	追加・変更概要
1	1.2	図 1.1 新規伝送メディア追加
2	2.1	(22)下位通信ソフトウェアアドレステーブルデータ取得 (23)マスタルータ通知 (24)ハードウェアアドレスデータ取得を追記
3	2.2	(22)下位通信ソフトウェアアドレステーブルデータ取得 (23)マスタルータ通知 (24)ハードウェアアドレスデータ取得の説明を追記
4	3.1	表 3.1 に、下位通信ソフトウェアアドレステーブルデータ取得、マスタルータ通知、ハードウェアアドレスデータ取得を追記
5	3.2	下位通信ソフトウェアアドレステーブルデータ取得、マスタルータ通知、ハードウェアアドレスデータ取得の説明を追記
6	4.1	表 4.1 の記述修正
7	4.1	表 4.1 に ClcLowGetAddressTableData, ClcLowSetMasterRouterFlag, ClcLowGetHardwareAddress を追記
8	4.2.29	ClcLowGetAddressTableData の説明を追記
9	4.2.30	ClcLowSetMasterRouterFlag の説明を追記
10	4.2.31	ClcLowGetHardwareAddress の説明を追記

• Version3.00  
 • Version3.10Draft  
 • Version3.10  
 • Version3.11  
 • Version3.12

2002年8月29日 コンソーシアム会員内公開。  
 2002年11月8日 コンソーシアム会員内公開。  
 2002年12月18日 コンソーシアム会員内公開。  
 2003年3月7日 コンソーシアム会員内公開。  
 2003年5月22日 コンソーシアム会員内公開。  
 変更のある項目は以下の通り

	変更部位 (目次項目)	追加・変更概要
1	3.2	(24)ハードウェアアドレスデータ取得の出力データからハードウェアタイプを削除
2	4.2.32	誤記修正
3	2.1 3.1 3.2	下位通信ソフトウェアテーブルデータサイズ取得の追記

• Version3.20Draft

2003年10月17日 コンソーシアム会員内公開。  
変更のある項目は以下の通り

	変更部位 (目次項目)	追加・変更概要
1	2.1、2.2、3.1、3.2	インタフェースを追加
2	4.1	関数を追加
3	4.2.18	ClcLowGetProData の引数修正
4	4.2.31	マスターータ設定関数に名称変更
5	4.2.33	追加
6	4.2.34	追加
7	4.2.35	追加

• Version3.20

2004年 1月 8日 コンソーシアム会員内公開。  
変更のある項目は以下の通り

	変更部位 (目次項目)	追加・変更概要
1	3.1	表3.1 誤記修正
2	4.1	表4.1 誤記修正
3	4.2.8	誤記修正
4	4.2.18	誤記修正
5	4.2.31	誤記修正
6	4.2.33	NodeID リストをテーブル記載に変更 誤記修正
7	4.2.34	関数名「ClcGetMasterRouterInfo」に変更 名称を「マスターータ情報取得関数」に変更

• Version3.21

2004年 5月 26日 コンソーシアム会員内公開。  
変更のある項目は以下の通り

	変更部位 (目次項目)	追加・変更概要
1	4.2.20	関数 ClcSendData の引数 dnode_id の説明を修正

• Version3.21

2005年10月13日 一般公開。

- エコーネットコンソーシアムが発行している規格類は、工業所有権(特許, 実用新案など)に関する抵触の有無に関係なく制定されています。エコーネットコンソーシアムは、この規格類の内容に関する工業所有権に対して、一切の責任を負いません。
- 本規格発行者は有償・無償を問わず、いかなる第三者に対しても JAVA、IrDA、Bluetooth、HBS のライセンスを許諾する権限や免責を与える権限を有していません。JAVA、IrDA、Bluetooth、HBS を使用する場合、当該使用者は自己の責任と判断に基づき、上記規格について使用許可を得るなどの措置が必要です。
- この書面の使用による、いかなる損害も責任を負うものではありません。

## 目次

第1章 ECHONET 共通下位通信インタフェース仕様概要 .....	1-1
1.1 基本的な考え方 .....	1-1
1.2 通信レイヤ上の位置づけ .....	1-2
第2章 ECHONET 共通下位通信インタフェース機能仕様 .....	2-1
2.1 ECHONET 共通下位通信インタフェース一覧 .....	2-1
2.2 ECHONET 共通下位通信インタフェース機能詳細仕様 .....	2-2
第3章 レベル1 ECHONET 共通下位通信インタフェース仕様 .....	3-1
3.1 レベル1 ECHONET 共通下位通信インタフェースのサービス一覧 .....	3-1
3.2 レベル1 ECHONET 共通下位通信インタフェース詳細仕様 .....	3-3
第4章 レベル2 ECHONET 共通下位通信インタフェース仕様 .....	4-1
4.1 C 言語用レベル2 ECHONET 共通下位通信インタフェース関数一覧 .....	4-2
4.2 C 言語用レベル2 ECHONET 共通下位通信インタフェース詳細仕様 .....	4-4
4.2.1 ClcGetDevID .....	4-5
4.2.2 ClcInit .....	4-6
4.2.3 ClcRequestRun .....	4-7
4.2.4 ClcSetTrouble .....	4-8
4.2.5 ClcStart .....	4-9
4.2.6 ClcSuspend .....	4-10
4.2.7 ClcWakeUp .....	4-11
4.2.8 ClcGetProData .....	4-12
4.2.9 ClcGetStatus .....	4-13
4.2.10 ClcInitAll .....	4-14
4.2.11 ClcStop .....	4-15
4.2.12 ClcHalt .....	4-16
4.2.13 ClcLowInit .....	4-17
4.2.14 ClcLowRequestRun .....	4-19
4.2.15 ClcLowStart .....	4-20
4.2.16 ClcLowSuspend .....	4-21
4.2.17 ClcLowWakeUp .....	4-22
4.2.18 ClcGetLowProData .....	4-23
4.2.19 ClcGetLowStatus .....	4-25
4.2.20 ClcSendData .....	4-26
4.2.21 ClcGetSendResult .....	4-28
4.2.22 ClcSendCancel .....	4-29
4.2.23 ClcReceiveData .....	4-30

---

4 . 2 . 2 4	ClcGetNodeID .....	4-3 1
4 . 2 . 2 5	ClcSetNodeID.....	4-3 2
4 . 2 . 2 6	ClcLowInitAll .....	4-3 3
4 . 2 . 2 7	ClcLowStop.....	4-3 5
4 . 2 . 2 8	ClcLowHalt.....	4-3 6
4 . 2 . 2 9	ClcLowGetAddressTableDataSize .....	4-3 7
4 . 2 . 3 0	ClcLowGetAddressTableData.....	4-3 8
4 . 2 . 3 1	ClcLowSetMasterRouterFlag.....	4-3 9
4 . 2 . 3 2	ClcLowGetHardwareAddress .....	4-4 0
4 . 2 . 3 3	ClcGetNodeIDList .....	4-4 1
4 . 2 . 3 4	ClcGetMasterRouterInfo.....	4-4 3
4 . 2 . 3 5	ClcLowReqToHardwareAddress.....	4-4 4

## 第1章 ECHONET 共通下位通信インタフェース仕様概要

### 1.1 基本的な考え方

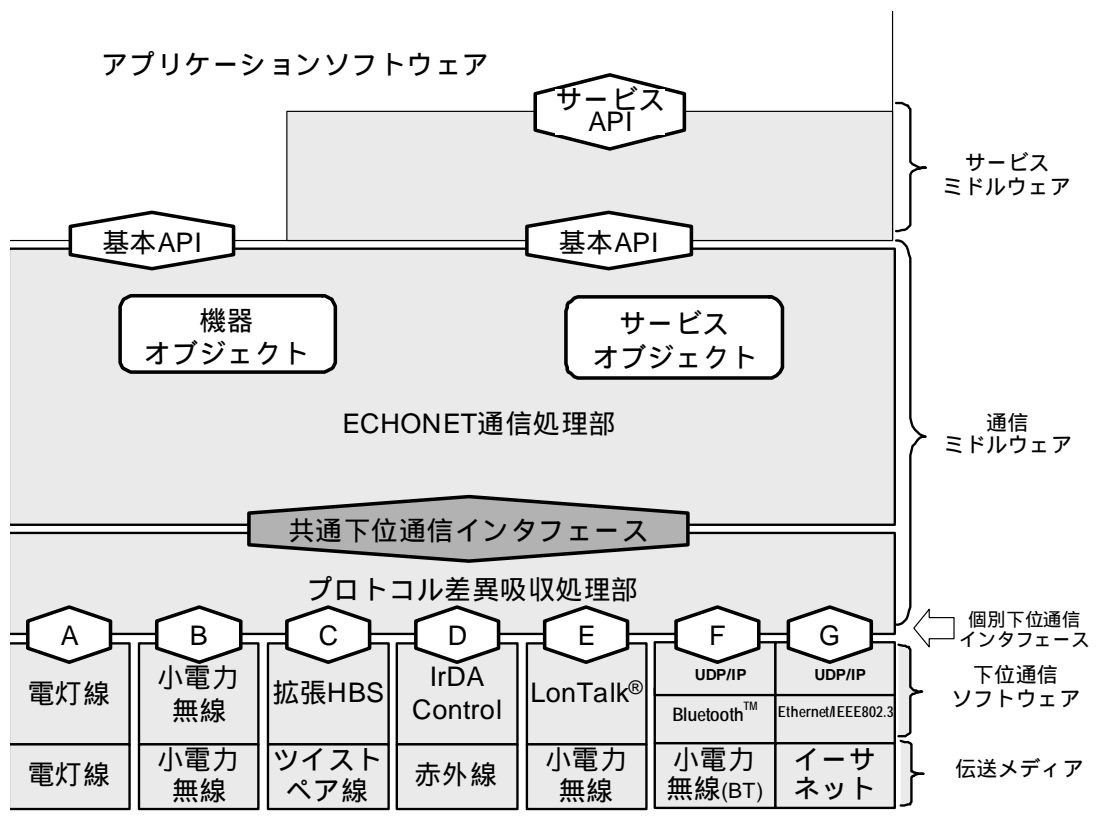
「第5部 ECHONET 共通下位通信インタフェース仕様」の規定は、次節の図1.1にて示す ECHONET 通信処理部とプロトコル差異吸収処理部との間の情報のやり取りを実現する為のソフトウェアインタフェースの仕様に関するものである。この共通下位通信インタフェースにより、ECHONET 通信処理部の処理仕様は、個々の下位通信ソフトウェアの仕様を意識することなく共通的な形で記述ができることとなる。この共通下位通信インタフェースの仕様では、プロトコル差異吸収処理部がサポートする想定でのインタフェースサービスについて、入出力データ項目、及び具体的な言語が指定された場合の関数規定を、それぞれ、共通下位通信インタフェース仕様レベル1, 2として規定する。共通下位通信インタフェース仕様レベル1, 2は、基本APIのレベル1, 2の考えに基づく。



## 1.2 通信レイヤ上の位置づけ

本章で規定するインタフェース仕様は、ECHONET 通信処理部が、下位通信ソフトウェアの違いを意識することなく、プロトコル差異吸収処理部以下を制御する為に下位通信ソフトウェアの違いを吸収する形で規定するものである。

下図1.1に、本書で規定する共通下位通信インタフェースの位置づけを網掛けで示す。共通下位通信インタフェースは、プロトコル差異吸収処理と ECHONET 通信処理部の間において、相互のやり取りを実現するものである。



LonTalk は、米国その他の国々での Echelon Corporation の登録商標です。  
 Bluetooth は Bluetooth SIG, Inc の登録商標です。  
 Ethernet は、Xerox Corporation の登録商標です。  
 その他すべての商標は、それぞれの所有者に属するものです。

図1.1 共通下位通信インタフェースの位置づけ

## 第2章 ECHONET 共通下位通信インタフェース機能仕様

### 2.1 ECHONET 共通下位通信インタフェース一覧

プロトコル差異吸収処理部がサポートする ECHONET 共通下位通信インタフェースの一覧を以下に示す。プロトコル差異吸収処理部は、これらのインタフェースを備えているものとする。インタフェースの機能仕様説明は、次節にて示す。

- (1) 下位通信ソフトウェア搭載情報要求
- (2) 初期化要求
- (3) 動作開始要求
- (4) 障害通知
- (5) ウォームスタート要求
- (6) 一時停止要求
- (7) 動作再開要求
- (8) プロトコル差異吸収処理部プロファイルデータ取得要求
- (9) 下位通信ソフトウェアプロファイルデータ取得要求
- (10) プロトコル差異吸収処理部ステータスデータ取得要求
- (11) 下位通信ソフトウェアステータスデータ取得要求
- (12) 電文送信要求
- (13) 送信結果取得
- (14) 送信中止要求
- (15) 受信電文要求
- (16) NodeID 取得要求
- (17) NodeID 設定要求
- (18) 完全初期化要求
- (19) 通信停止要求
- (20) 完全停止要求
- (21) 停止通知
- (22) 下位通信ソフトウェアアドレステーブルデータサイズ取得
- (23) 下位通信ソフトウェアアドレステーブルデータ取得
- (24) マスタルータ通知
- (25) ハードウェアアドレスデータ取得
- (26) NondeID リスト取得
- (27) マスタルータ情報取得
- (28) ハードウェアアドレス変換要求

## 2.2 ECHONET 共通下位通信インタフェース機能詳細仕様

プロトコル差異吸収処理部がサポートする ECHONET 共通下位通信インタフェースの機能の説明を以下に示す。なお、本節の説明の中で使用されているプロトコル差異吸収処理部、下位通信ソフトウェアの遷移状態に関しては、第3部の対応する下位通信ソフトウェアを参照されたい。

### (1) 下位通信ソフトウェア搭載情報要求

プロトコル差異吸収処理部に対し、下位通信ソフトウェアに関する情報(搭載している下位通信ソフトウェアの数と下位通信ソフトウェア ID)を要求する。

### (2) 初期化要求

プロトコル差異吸収処理部及び下位通信ソフトウェアに対し、コールドスタートによる初期化を実施後、通信停止状態に遷移することを要求する。このとき、下位通信ソフトウェアの保持している MAC アドレスは、破棄・更新される。

### (3) 動作開始要求

プロトコル差異吸収処理部及び下位通信ソフトウェアに対し、通信停止状態から通常動作状態に遷移することを要求する。

### (4) 障害通知

ECHONET 通信処理部より上位層の障害(異常)状態をプロトコル差異吸収処理部へ通知する。

### (5) ウォームスタート要求

プロトコル差異吸収処理部及び下位通信ソフトウェアに対し、ウォームスタートによる初期化を実施後、通信停止状態に遷移することを要求する。このとき、下位通信ソフトウェアが保持している MAC アドレスは変化しない。

### (6) 一時停止要求

プロトコル差異吸収処理部及び下位通信ソフトウェアに対し、通常動作状態から一時停止状態に遷移することを要求する。

### (7) 動作再開要求

プロトコル差異吸収処理部及び下位通信ソフトウェアに対し、一時停止状態を解除し、通常動作状態に遷移することを要求する。

- (8) プロトコル差異吸収処理部プロファイルデータ取得要求  
プロトコル差異吸収処理部に対して、プロトコル差異吸収処理部のプロファイルデータを要求する。本機能にて要求するプロファイルデータは、ソフトウェア開発メーカーコードやバージョンNo.等の静的情報とする。
- (9) 下位通信ソフトウェアプロファイルデータ取得要求  
プロトコル差異吸収処理部に対し、下位通信ソフトウェアのプロファイルデータを要求する。本機能にて要求するプロファイルデータは、下位通信ソフトウェアの開発メーカーコードやバージョンNo.等の静的情報とする。
- (10) プロトコル差異吸収処理部ステータスデータ取得要求  
プロトコル差異吸収処理部に対し、ステータスデータを要求する。本機能にて要求するステータスデータは、異常状態や処理状態といったプロトコル差異吸収処理部の動的情報とする。
- (11) 下位通信ソフトウェアステータスデータ取得要求  
プロトコル差異吸収処理部に対し、下位通信ソフトウェアのステータスデータを要求する。本機能にて要求するステータスデータは、異常状態や処理状態といった下位通信ソフトウェアの動的情報とする。
- (12) 電文送信要求  
プロトコル差異吸収処理部に対して、指定する ECHONET 電文の送信を要求する。
- (13) 送信結果取得  
プロトコル差異吸収処理部に対して、この要求の直前に要求した電文送信の処理のステータスを要求する。
- (14) 送信中止要求  
プロトコル差異吸収処理部に対して、この要求の直前の電文送信要求に応じて下位通信ソフトウェアが行っている電文送信処理の中止を要求する。
- (15) 受信電文要求  
プロトコル差異吸収処理部に対して、受信した電文の受け渡しを要求する。
- (16) NodeID 取得要求  
プロトコル差異吸収処理部にて保持している NodeID 情報を要求する。
- (17) NodeID 設定要求  
プロトコル差異吸収処理部に対して、NodeID 情報を設定する。

- (18) 完全初期化要求  
プロトコル差異吸収処理部に対して、下位通信ソフトウェアをコールドスタートさせた後、通信停止状態に遷移させることを要求する。このとき、ハウスコード情報、MAC アドレスは再取得される。
- (19) 通信停止要求  
プロトコル差異吸収処理部に対して、下位通信ソフトウェアを通信停止状態に遷移させることを要求する。
- (20) 完全停止要求  
プロトコル差異吸収処理部に対して、下位通信ソフトウェアを停止状態に遷移させることを要求する。
- (21) 停止通知  
プロトコル差異吸収処理部から ECHONET 通信処理部に対して、下位通信ソフトウェアが停止状態に遷移したことを通知する。
- (22) 下位通信ソフトウェアアドレステーブルデータサイズ取得  
下位通信ソフトウェアで保持している下位アドレステーブルデータの組数を取得する。
- (23) 下位通信ソフトウェアアドレステーブルデータ取得  
下位通信ソフトウェアで保持している下位アドレステーブルデータを取得する。
- (24) マスタルータ通知  
自ノードがマスタルータであるか否かを下位通信ソフトウェアに通知する。
- (25) ハードウェアアドレスデータ取得  
下位通信ソフトウェアに対して保持しているハードウェアアドレスデータを取得する。
- (26) NodeID リスト取得  
下位通信ソフトウェアが保持する NodeID リスト取得する。
- (27) マスタルータ情報取得  
下位通信ソフトウェアが保持するマスタルータ情報を取得する。
- (28) ハードウェアアドレス変換要求  
下位通信ソフトウェアに対して、渡した NodeID に対応するハードウェアアドレスを要求する。

### 第3章 レベル1 ECHONET 共通下位通信インタフェース仕様

#### 3.1 レベル1 ECHONET 共通下位通信インタフェースのサービス一覧

レベル1 ECHONET 共通下位通信インタフェースは、表3.1に示すサービス毎に、ECHONET 通信処理部とプロトコル差異吸収処理部間でやり取りされるデータについての規定を行う。本レベル1 ECHONET 共通下位通信インタフェースに準拠した実装とするためには、次節で規定する入出力データ項目を備えている必要がある。ただし、複数のサービスを1つのサービスに統合したり、1つのサービスを複数のサービスとして分割したりしてもかまわない。また、複数のデータ項目を1つのデータ項目として処理したり、或いは1つのデータ項目を複数のデータ項目として処理したりしてもかまわない。

表3.1 レベル1 ECHONET 共通下位通信インタフェースのサービス一覧 (1/2)

No.	サービス名称	機能概要	搭載規定
1	下位通信ソフトウェア搭載 情報要求	搭載している(アクセス可能な)下位通信ソフトウェアの数・下位通信ソフトウェアの種類を要求する。	Required
2	初期化要求	指定したプロトコル差異吸収処理部及び下位通信ソフトウェアのコールドスタートによる初期化を要求する。	Required
3	動作開始要求	指定したプロトコル差異吸収処理部及び下位通信ソフトウェアの動作開始を要求する。	Required
4	障害通知	ECHONET 通信処理部より上位層の障害(異常)状況をプロトコル差異吸収処理部へ通知する。	Optional
5	ウォームスタート要求	指定したプロトコル差異吸収処理部及び下位通信ソフトウェアに対して、ウォームスタートによる初期化を要求する。	Required
6	一時停止要求	指定したプロトコル差異吸収処理部及び下位通信ソフトウェアに対して、動作の一時停止を要求する。	Optional
7	動作再開要求	指定したプロトコル差異吸収処理部及び下位通信ソフトウェアに対して、動作の再開を要求する。	Optional
8	プロトコル差異吸収処理部 プロファイルデータ取得要求	プロトコル差異吸収処理部の静的情報を取得する。	Required
9	下位通信ソフトウェアプロ ファイルデータ取得要求	下位通信ソフトウェアの静的情報を取得する。	Required
10	プロトコル差異吸収処理部 ステータスデータ取得要求	プロトコル差異吸収処理部の動的状態(処理障害等)情報を取得する。	Optional
11	下位通信ソフトウェアステ ータスデータ取得要求	下位通信ソフトウェアの動的状態(処理障害、アドレス重複等)に関する情報を取得する。	Required
12	電文送信要求	プロトコル差異吸収処理部に対して、電文の送信を要求する。	Required

表3.1 ECHONET 共通下位通信インタフェース一覧表(2/2)

No.	サービス名称	機能概要	搭載規定
13	送信結果取得	プロトコル差異吸収処理部に対して、電文の送信結果を要求する。	Optional
14	送信中止要求	プロトコル差異吸収処理部に対して、電文の送信の中止を要求する。	Optional
15	受信電文要求	プロトコル差異吸収処理部に対して、受信電文を要求する。	Required
16	NodeID 取得要求	プロトコル差異吸収処理部で保持している NodeID を取得する。	Required
17	NodeID 設定要求	プロトコル差異吸収処理部に対して NodeID を設定する。	Optional
18	完全初期化要求	指定したプロトコル差異吸収処理部及び下位通信ソフトウェアに対して、コールドスタートによる初期化を要求する。このとき、ハウスコード情報は再取得される。	Optional
19	通信停止要求	指定したプロトコル差異吸収処理部及び下位通信ソフトウェアに対して、通信停止状態への遷移を要求する。	Optional
20	完全停止要求	指定したプロトコル差異吸収処理部及び下位通信ソフトウェアに対して、停止状態への遷移を要求する。	Optional
21	停止通知	プロトコル差異吸収処理部から ECHONET 通信処理部に対し、プロトコル差異吸収処理部及び下位通信ソフトウェアが停止状態に遷移した旨を通知する	Optional
22	下位通信ソフトウェアアドレステーブルデータサイズ取得	下位通信ソフトウェアで保持している下位アドレステーブルデータの組数の取得する。	Optional
23	下位通信ソフトウェアアドレステーブルデータ取得	下位通信ソフトウェアで保持している下位アドレステーブルデータを取得する。	Optional
24	マスターータ通知	自ノードがマスターータであるか否かを下位通信ソフトウェアに通知する。	Optional
25	ハードウェアアドレスデータ取得	下位通信ソフトウェアから保持しているハードウェアアドレスデータを取得する。	Optional
26	NodeID リスト取得	下位通信ソフトウェアが保持する NodeID リストを取得する。	Optional
27	マスターータ情報取得	下位通信ソフトウェアが保持するマスターータ情報を取得する。	Optional
28	ハードウェアアドレス変換要求	下位通信ソフトウェアに対して、渡した NodeID に対応するハードウェアアドレスを要求する。	Optional

### 3.2 レベル1 ECHONET 共通下位通信インタフェース詳細仕様

前節の表3.1で示した各サービスにより、入出力されるデータに関しての規定を行う。以下の表中で、データの入出力の方向は、ECHONET 通信処理部を基準とする。すなわち、「入力 (Input)」とは、ECHONET 通信処理部からプロトコル差異吸収処理部に対してデータを渡すことを示し、「出力 (Output)」とは、プロトコル差異吸収処理部からECHONET 通信処理部へデータを渡すことを示す。レベル1 ECHONET 共通下位通信インタフェースに準拠するためには、これらのデータの受け渡しが可能であればよく、受け渡し方法 (構造体を用いるとか、受け渡し用のバッファのポインタ情報を渡す等) については特に規定しない。

#### (1) 下位通信ソフトウェア搭載情報要求 (搭載必須機能)

搭載している (アクセス可能な) 下位通信ソフトウェアの数と種別 (電灯線、小電力無線等) を要求する。表3.2にデータ仕様を示す。

表3.2 下位通信ソフトウェア種別要求サービス入出力データ一覧

方向	データ名	内容・条件	備考
Input	-		
Output	Device_num	・ 搭載している下位通信ソフトウェアの数を示す。	Optional
Output	Device_id	・ 下位通信ソフトウェアの種類を示す。 ・ 電灯線下位通信ソフトウェア, 特定小電力無線下位通信ソフトウェア, 拡張 HBS 下位通信ソフトウェア, LonTalk <sup>®</sup> 依存下位通信ソフトウェア, IrDA 依存下位通信ソフトウェア等の識別が可能であること。 ・ 複数の下位通信ソフトウェアに対応可能な場合には、複数の応答が可能であること。	Required
Output	Return Value	TRUE : 正常時、FALSE : 異常時	Optional



## (2) 初期化要求 (搭載必須機能)

指定した下位通信ソフトウェアのコールドスタートによる初期化と、その後の通信停止状態への遷移、およびそれに対応するプロトコル差異吸収処理部の初期化を要求する。この要求に対応する一連の処理の過程で、MAC アドレス情報は再取得される。ハウスコード情報をもつ下位通信ソフトウェアの場合、ハウスコード情報は変化しない。表3.3に入出力仕様を示す。

表3.3 初期化要求サービス入出力データ一覧

方向	データ名	内容・条件	備考
Input	Device_id	<ul style="list-style-type: none"> <li>初期化対象下位通信ソフトウェアの指定を行う。</li> <li>全下位通信ソフトウェアの一斉初期化、あるいは個々の下位通信ソフトウェアの個別初期化が指定可能であること。</li> </ul>	Required
Input	p_init	<ul style="list-style-type: none"> <li>初期化パラメータを指定する。</li> <li>送信電文最長保持時間、受信電文最長保持時間等含むが初期化対象下位通信ソフトウェアにより具体的な内容は異なる。</li> </ul>	Required
Output	Return Value	TRUE : 初期化成功、FALSE : 初期化失敗	Optional

## (3) 動作開始要求 (搭載必須機能)

指定した下位通信ソフトウェアとそれに対応するプロトコル差異吸収処理部の動作開始を要求する。表3.4に入出力仕様を示す。

表3.4 動作開始要求サービス入出力データ一覧

方向	データ名	内容・条件	備考
Input	Device_id	<ul style="list-style-type: none"> <li>動作開始対象下位通信ソフトウェアの指定を行う。</li> <li>全下位通信ソフトウェアの一斉動作開始、あるいは個々の下位通信ソフトウェアの動作開始が指定可能であること。</li> </ul>	Optional
Output	Return Value	TRUE : 動作開始成功、FALSE : 動作開始失敗	Optional

## (4) 障害通知

ECHONET 通信処理部より上位層の障害（異常）状態をプロトコル差異吸収処理部へ通知する。表3.5に入出力仕様を示す。

表3.5 障害通知サービス入出力データ一覧

方向	データ名	内容・条件	備考
Input	trouble_no	・ 障害(異常)状態を示すトラブル番号を通知する。	Required
Output	Return Value	TRUE: 障害通知受理、FALSE: 障害通知不受理	Optional

## (5) ウォームスタート要求（搭載必須機能）

指定した下位通信ソフトウェア、およびそれに対応するプロトコル差異吸収処理部に対してウォームスタートによる初期化と、その後の通信停止状態への遷移を要求する。この要求に対応する一連の処理の過程でハウスコード情報、MAC アドレス情報は変化しない。表3.6に入出力仕様を示す。

表3.6 ウォームスタート要求サービス入出力データ一覧

方向	データ名	内容・条件	備考
Input	device_id	・ ウォームスタート対象下位通信ソフトウェアの指定を行う。 ・ 全下位通信ソフトウェアの一斉ウォームスタート、あるいは個々の下位通信ソフトウェアのウォームスタート指定が可能であること。	Required
Output	Return Value	TRUE: ウォームスタート受理、FALSE: 受理不可	Optional

## (6) 一時停止要求

指定した下位通信ソフトウェア、およびそれに対応するプロトコル差異吸収処理部に対して一時停止状態への遷移を要求する。表3.7に入出力仕様を示す。

表3.7 一時停止要求サービス入出力データ一覧

方向	データ名	内容・条件	備考
Input	device_id	・ 一時停止対象下位通信ソフトウェアの指定を行う。 ・ 全下位通信ソフトウェアの一斉一時停止、あるいは個々の下位通信ソフトウェアの初期化が指定可能であること。	Required
Output	Return Value	TRUE: 一時停止受理、FALSE: 受理不可	Optional

## (7) 動作再開要求

指定した下位通信ソフトウェア、およびそれに対応するプロトコル差異吸収処理部に対して一時停止状態を解除し、通常動作状態に遷移することを要求する。表3.8に入出力仕様を示す。

表3.8 動作再開要求サービス入出力データ一覧

方向	データ名	内容・条件	備考
Input	device_id	<ul style="list-style-type: none"> <li>動作再開対象下位通信ソフトウェアの指定を行う。</li> <li>全下位通信ソフトウェアの一斉動作再開、あるいは個々の下位通信ソフトウェアの動作再開が指定可能であること。</li> </ul>	Required
Output	Return Value	TRUE : 再開成功、FALSE : 再開不可(失敗含)	Optional

## (8) プロトコル差異吸収処理部プロファイルデータ取得要求(搭載必須機能)

指定した下位通信ソフトウェアに対応するプロトコル差異吸収処理部のプロファイルデータを要求する。本サービスにて要求するデータは、プロトコル差異吸収処理部のメーカーコードやバージョンNo.等の静的情報とする。表3.9に入出力仕様を示す。

表3.9 プロトコル差異吸収処理部プロファイルデータ取得要求サービス入出力データ一覧

方向	データ名	内容・条件	備考
Input	device_id	<ul style="list-style-type: none"> <li>取得対象とするプロトコル差異吸収処理部に関連する下位通信ソフトウェアの指定を行う。</li> <li>全下位通信ソフトウェア、あるいは個々の下位通信ソフトウェアの指定が可能であること。</li> </ul>	Required
Output	version_No	プロトコル差異吸収処理部のバージョン情報を示す。	Optional
Output	company_name	メーカーコード示す。	Optional
Output	Rwlen	バッファサイズ情報示す。	Optional
Output	Return Value	TRUE : 正常時、FALSE : 異常時	Optional

## (9) 下位通信ソフトウェアプロファイルデータ取得要求(搭載必須機能)

指定した下位通信ソフトウェアのプロファイルデータを要求する。本機能にて要求するプロファイルデータは、下位通信ソフトウェアのソフトウェア開発メーカーのメーカーコードやバージョンNo.等の静的情報とする。表3.10に入出力仕様を示す。

表3.10 下位通信ソフトウェアプロファイルデータ取得要求サービス入出力データ一覧

方向	データ名	内容・条件	備考
Input	device_id	<ul style="list-style-type: none"> <li>取得対象とする下位通信ソフトウェアの指定を行う。</li> <li>全下位通信ソフトウェア、あるいは個々の下位通信ソフトウェアの指定が可能であること。</li> </ul>	Required
Output	version_No	<ul style="list-style-type: none"> <li>下位通信ソフトウェアのバージョン情報示す。</li> </ul>	Optional
Output	company_name	<ul style="list-style-type: none"> <li>メーカー情報示す。</li> </ul>	Optional
Output	mac_address	<ul style="list-style-type: none"> <li>MACアドレス情報示す。</li> </ul>	Optional
Output	Rwlen	<ul style="list-style-type: none"> <li>バッファサイズ情報示す。</li> </ul>	Optional
Output	Broad	<ul style="list-style-type: none"> <li>同報機能の有無示す。</li> </ul>	Optional
Output	Baud	<ul style="list-style-type: none"> <li>伝送レート示す。</li> </ul>	Optional
Output	Return Value	TRUE : 正常時、FALSE : 異常時	Optional

## (10) プロトコル差異吸収処理部ステータスデータ取得要求

プロトコル差異吸収処理部に対して、プロトコル差異吸収処理部のステータスデータを要求する。本機能にて要求するステータスデータは、異常状態や処理状態といった動的情報とする。表3.11に入出力仕様を示す。

表3.11 プロトコル差異吸収処理部ステータスデータ取得要求サービス入出力データ一覧

方向	データ名	内容・条件	備考
Input	device_id	<ul style="list-style-type: none"> <li>取得対象とするプロトコル差異吸収処理部に関連する下位通信ソフトウェアの指定を行う。</li> <li>全下位通信ソフトウェア、あるいは個々の下位通信ソフトウェアの指定が可能であること。</li> </ul>	Required
Output	State	<ul style="list-style-type: none"> <li>遷移状態情報を示す。</li> </ul>	Required
Output	trouble_no	<ul style="list-style-type: none"> <li>プロトコル差異吸収処理部の障害情報示す。</li> </ul>	Optional
Output	upper_trouble	<ul style="list-style-type: none"> <li>上位層障害として認識している情報示す。</li> </ul>	Optional
Output	Low_trouble	<ul style="list-style-type: none"> <li>下位層障害として認識している情報示す。</li> </ul>	Optional
Output	Return Value	TRUE : 正常時、FALSE : 異常時	Optional

## (11) 下位通信ソフトウェアステータスデータ取得要求 (搭載必須機能)

プロトコル差異吸収処理部に対して、下位通信ソフトウェアのステータスデータを要求する。本機能にて要求するステータスデータは、異常状態や処理状態といった動的情報とする。表3.12に入出力仕様を示す。

表3.12 下位通信ソフトウェアステータスデータ取得要求サービス入出力データ一覧

方向	データ名	内容・条件	備考
Input	device_id	<ul style="list-style-type: none"> <li>取得対象とする下位通信ソフトウェアの指定を行う。</li> <li>全下位通信ソフトウェア、あるいは個々の下位通信ソフトウェアの指定が可能であること。</li> </ul>	Required
Output	State	<ul style="list-style-type: none"> <li>遷移状態情報。第3部規定の状態の認識が可能であること。</li> </ul>	Required
Output	trouble_no	<ul style="list-style-type: none"> <li>下位通信ソフトウェアの障害情報を示す。</li> </ul>	Optional
Output	upper_trouble	<ul style="list-style-type: none"> <li>上位層障害として下位通信ソフトウェアが、認識している情報。</li> </ul>	Optional
Output	Return Value	TRUE : 正常時、FALSE : 異常時	Optional

## (12) 電文送信要求 (搭載必須機能)

指定した下位通信ソフトウェアによる指定 ECHONET 電文の送信を要求する。表3.13に入出力仕様を示す。

表3.13 電文送信要求サービス入出力サービス一覧

方向	データ名	内容・条件	備考
Input	device_id	<ul style="list-style-type: none"> <li>送信対象とする下位通信ソフトウェアの指定を行う。</li> <li>個々の下位通信ソフトウェアの識別が可能であること。</li> </ul>	Required
Input	send_data	<ul style="list-style-type: none"> <li>ECHONET 電文形式の送信要求データを示す。</li> <li>形式は ECHONET 通信処理部間でやり取りされるもの (ECHONET フレーム) である。</li> </ul>	Required
Input	dnode_id	<ul style="list-style-type: none"> <li>サブネット内送信先 NodeID 情報と、同報/個別の指定情報を示す。</li> </ul>	Required
Output	Return Value	TRUE : 正常時、FALSE : 異常時	Optional

## (13) 送信結果取得

指定した下位通信ソフトウェアに対応するプロトコル差異吸収処理部に対して、「電文送信要求」で要求した電文の送信結果を要求する。表3.14に入出力仕様を示す。

表3.14 送信結果取得サービス入出力データ一覧

方向	データ名	内容・条件	備考
Input	device_id	<ul style="list-style-type: none"> <li>送信結果取得対象とする下位通信ソフトウェアの指定を行う。</li> <li>個々の下位通信ソフトウェアの識別が可能であること。</li> </ul>	Required
Output	result	<ul style="list-style-type: none"> <li>送信中、送信正常終了、送信エラー終了、或いは送信中止中の情報を示す。</li> </ul>	Required
Output	Return Value	TRUE : 正常時、FALSE : 異常時	Optional

## (14) 送信中止要求

指定した下位通信ソフトウェアが実行中の電文送信処理の中止を要求する。表3.15に入出力仕様を示す。

表3.15 送信中止要求サービス入出力データ一覧

方向	データ名	内容・条件	備考
Input	device_id	<ul style="list-style-type: none"> <li>送信中止対象とする下位通信ソフトウェアの指定を行う。</li> <li>個々の下位通信ソフトウェアの識別が可能であること。</li> </ul>	Required
Output	Return Value	TRUE : 中止成功、FALSE : 中止失敗(送信済)	Optional

## (15) 受信電文要求(搭載必須機能)

指定した下位通信ソフトウェアが受信している電文を要求する。表3.16に入出力仕様を示す。

表3.16 受信電文要求サービス入出力データ一覧

方向	データ名	内容・条件	備考
Input	device_id	<ul style="list-style-type: none"> <li>要求対象とする下位通信ソフトウェアの指定を行う。</li> <li>個々の下位通信ソフトウェアの識別が可能であること。</li> </ul>	Required
Output	receive_data	<ul style="list-style-type: none"> <li>ECHONET 電文形式の受信データを示す。</li> <li>形式は ECHONET 通信処理部間でやり取りされるもの (ECHONET フレーム) である。</li> </ul>	Required
Output	snode_id	<ul style="list-style-type: none"> <li>サブネット内送信元 NodeID 情報を示す。</li> </ul>	Required
Output	Return Value	TRUE : 正常時、FALSE : 異常時(受信データ無しの指示等異常指示コード)	Optional

## (16) NodeID 取得要求 (搭載必須機能)

指定した下位通信ソフトウェアが保持している MAC アドレスに対応する NodeID 情報を要求する。表 3.17 に入出力仕様を示す。

表 3.17 NodeID 取得要求サービス入出力データ一覧

方向	データ名	内容・条件	備考
Input	device_id	<ul style="list-style-type: none"> <li>NodeID 取得対象とする下位通信ソフトウェアの指定を行う。</li> <li>個々の下位通信ソフトウェアの指定が可能であること。</li> </ul>	Required
Output	nodeID	<ul style="list-style-type: none"> <li>NodeID 情報を示す。</li> </ul>	Required
Output	Return Value	TRUE : 正常時、FALSE : 異常時(nodeID 未設定や指定 device_id 異常等指示コード)	Optional

## (17) NodeID 設定要求

指定した下位通信ソフトウェアに対応するプロトコル差異吸収処理部に対して、NodeID 情報を設定する。下位通信ソフトウェアは、この設定に基づいて Mac アドレスを変更する。表 3.18 に入出力仕様を示す。

表 3.18 NodeID 設定要求サービス入出力データ一覧

方向	データ名	内容・条件	備考
Input	device_id	<ul style="list-style-type: none"> <li>NodeID 設定対象とする下位通信ソフトウェアの指定を行う。</li> <li>個々の下位通信ソフトウェアの指定が可能であること。</li> </ul>	Required
Input	nodeID	<ul style="list-style-type: none"> <li>NodeID 情報を示す。</li> </ul>	Required
Output	Return Value	TRUE : 正常時、FALSE : 異常時(設定不可等指示コード)	Optional

## (18) 完全初期化要求

指定した下位通信ソフトウェアとそれに対応するプロトコル差異吸収処理部のコールドスタートによる初期化と、その後の通信停止状態への遷移を要求する。この要求に対応する一連の処理の過程でハウスコード情報、MAC アドレス情報は再取得される。

表3.19 完全初期化要求サービス入出力データ一覧

方向	データ名	内容・条件	備考
Input	software_id	<ul style="list-style-type: none"> <li>完全初期化対象下位通信ソフトウェアの指定を行う。</li> <li>全下位通信ソフトウェアの一斉初完全期化、あるいは個々の下位通信ソフトウェアの個別完全初期化が指定可能であること。</li> </ul>	Required
Input	p_init	<ul style="list-style-type: none"> <li>初期化パラメータを指定する。 <ul style="list-style-type: none"> <li>送信電文最長保持時間 受信電文最長保持時間等含むが初期化対象下位通信ソフトウェアにより具体的な内容は異なる。</li> </ul> </li> </ul>	Required
Output	Return Value	TRUE：初期化成功、FALSE：初期化失敗	Optional

## (19) 通信停止要求

指定した下位通信ソフトウェア、およびそれに対応するプロトコル差異吸収処理部に対して通信停止状態への遷移を要求する。表3.20に入出力仕様を示す。

表3.20 通信停止要求サービス入出力データ一覧

方向	データ名	内容・条件	備考
Input	software_id	<ul style="list-style-type: none"> <li>通信停止対象下位通信ソフトウェアの指定を行う。</li> <li>全下位通信ソフトウェアの一斉通信停止、あるいは個々の下位通信ソフトウェアの個別通信停止が指定可能であること。</li> </ul>	Required
Output	Return Value	TRUE：一時停止受理、FALSE：受理不可	Optional

## (20) 完全停止要求

指定した下位通信ソフトウェア、およびそれに対応するプロトコル差異吸収処理部に対して停止状態への遷移を要求する。表3.21に入出力仕様を示す。

表3.21 完全停止要求サービス入出力データ一覧

方向	データ名	内容・条件	備考
Input	software_id	<ul style="list-style-type: none"> <li>完全停止対象下位通信ソフトウェアの指定を行う。</li> <li>全下位通信ソフトウェアの一斉完全停止、あるいは個々の下位通信ソフトウェアの個別完全停止が指定可能であること。</li> </ul>	Required
Output	Return Value	TRUE：一時停止受理、FALSE：受理不可	Optional



## (2.1) 停止通知

下位通信ソフトウェア、およびそれに対応するプロトコル差異吸収処理部が停止状態へ遷移したことを ECHONET 通信処理部に通知する。表3.2.2に入出力仕様を示す。

表3.2.2 停止通知サービス入出力データ一覧

方向	データ名	内容・条件	備考
Output	software_id	・ 停止状態に遷移した下位通信ソフトウェアを示す。	Required
Input	Return Value	TRUE : 通知受理、FALSE : 受理不可	Optional

## (2.2) 下位通信ソフトウェアアドレステーブルデータサイズ取得(Optional)

下位通信ソフトウェアが下位アドレステーブルで保持しているアドレスデータの組数の取得を行う。表3.2.3に入出力仕様を示す。

表3.2.3 下位通信ソフトウェアアドレステーブルデータサイズ取得  
サービス入出力データ一覧

方向	データ名	内容・条件	備考
Input	device_id	・ 下位通信ソフトウェア種別要求サービスにより取得した下位通信ソフトウェア ID を指定する。	Optional
Output	data_number	・ 下位アドレステーブルデータで保持しているアドレステーブルの組数を示す。	Required
Output	Return Value	TRUE : 正常時、FALSE : 異常時	Optional

## (23) 下位通信ソフトウェアアドレステーブルデータ取得 (Optional)

下位通信ソフトウェアで保持している下位アドレステーブルデータを取得する。データはデータ組数とデータ組から構成される。各データ組はハードウェアタイプ、ハードウェアアドレス、MAC アドレス、及びマスタールータであることを示すフラグよりなるデータである。

表3.24 下位通信ソフトウェアアドレステーブルデータ取得サービス  
入出力データ一覧

方向	データ名	内容・条件	備考
Input	device_id	・ 下位通信ソフトウェア種別要求サービスにより取得した下位通信ソフトウェア ID を指定する。	Optional
Output	data_number	・ 下位アドレステーブルデータで保持しているアドレステーブルの組数を示す。	Required
Output	hardwareaddress	・ 下位アドレステーブルデータで保持しているアドレステーブルに対応するハードウェアアドレスのリストを示す。	Required
Output	node_id	・ 下位アドレステーブルデータで保持しているアドレステーブルの NodeID のリストを示す。	Required
Output	masterrouter_flag	・ 下位アドレステーブルデータで保持しているアドレステーブルのノードがマスタールータであれば 1、そうでなければ 0 となる。	Required
Output	Return Value	TRUE : 正常時、FALSE : 異常時	Optional

## (24) マスタルータ通知 (Optional)

自ノードがマスタルータであるか否かを下位通信ソフトウェアに設定する。

表3.25 マスタルータ通知サービス入出力データ一覧

方向	データ名	内容・条件	備考
Input	device_id	・ 下位通信ソフトウェア種別要求サービスにより取得した下位通信ソフトウェア ID を指定する。	Optional
Input	masterRouter_Flag	・ マスタルータであれば 1、そうでなければ 0 を指定する。	Required
Output	Return Value	TRUE : 正常時、FALSE : 異常時	Optional

## (25) ハードウェアアドレスデータ取得 (Optional)

下位通信ソフトウェアが保持しているハードウェアアドレスデータを取得する。出力データはハードウェアアドレスである。

表3.26 ハードウェアアドレスデータ取得サービス入出力データ一覧

方向	データ名	内容・条件	備考
Input	device_id	・ 下位通信ソフトウェア種別要求サービスにより取得した下位通信ソフトウェア ID を指定する。	Optional
Output	hardwareaddress	・ 下位アドレステーブルデータで保持しているハードウェアアドレスを示す。	Required
Output	Return Value	TRUE : 正常時、FALSE : 異常時	Optional

## (26) NodeID リスト取得

下位通信ソフトウェアが保持する NodeID リスト取得する。

表3.27 NodeID リスト取得サービス入出力データ一覧

方向	データ名	内容・条件	備考
Input	device_id	・ 下位通信ソフトウェア種別要求サービスにより取得した下位通信ソフトウェア ID を指定する。	Optional
Output	node_id_list	・ 下位通信ソフトウェアが保持している NodeID リストを示す。	Required

## (27) マスタルータ情報取得

下位通信ソフトウェアが保持するマスタルータ情報を取得する。

表3.28 マスタルータ情報取得サービス入出力データ一覧

方向	データ名	内容・条件	備考
Input	device_id	・ 下位通信ソフトウェア種別要求サービスにより取得した下位通信ソフトウェア ID を指定する。	Optional
Output	result,	・ マスタルータの存在の有無	Required
Output	master_node_id	・ マスタルータの NodeID	Required

## (28) ハードウェアアドレス変換要求

下位通信ソフトウェアに対して、渡した NodeID に対応するハードウェアアドレスを要求する。

表3.29 ハードウェアアドレス変換要求サービス入出力データ一覧

方向	データ名	内容・条件	備考
Input	device_id	・ 下位通信ソフトウェア種別要求サービスにより取得した下位通信ソフトウェア ID を指定する。	Optional
Input	node_id	・ 変換対象の NodeID を示す。	Required
Output	hardwareaddress	・ ハードウェアアドレスへのポインタを示す。	Required
Output	hardwareaddress_len	・ ハードウェアアドレスのサイズへのポインタを示す。	Required

## 第4章 レベル2 ECHONET 共通下位通信インタフェース仕様

本章では、レベル2 ECHONET 共通下位通信インタフェースとして、本インタフェースを利用して開発するソフトウェアの互換性を考慮した API の詳細規定を示す。本章で示す規定は、プロトコル差異吸収処理部に API 処理を搭載する（ECHONET 通信処理部にてプロトコル差異吸収処理部の処理を呼び出す）形式とした。

レベル2 ECHONET 共通下位通信インタフェースについて、ECHONET 規格 Version 2.10 においては、ANSI 規格の C 言語を対象としたもの（以下「C 言語用」と呼ぶ）を規定する。

## 4.1 C 言語用レベル2 ECHONET 共通下位通信インタフェース関数一覧

C 言語用レベル2 ECHONET 共通下位通信インタフェースの関数として、以下の22個のものを規定する。レベル2の Optional とは、関数として実装しなくてもよい(例えば、プロトコル差異吸収処理部動作開始関数の実行で、個別下位通信ソフトウェアも同じに動作開始するような仕様になっているプロトコル差異吸収処理部においては、No.10の関数は実装していなくても良い)ことを示すが、その機能を実現している場合には、レベル2準拠とするには本項にて示す関数を実現するものとする。

表4.1 C 言語用レベル2 ECHONET 共通下位通信インタフェース関数一覧

No.	関数名	名称	備考
1	ClcGetDevID	下位通信ソフトウェア搭載情報要求関数	Required
2	ClcInit	初期化要求関数	Required
3	ClcRequestRun	動作開始要求関数	Required
4	ClcSetTrouble	上位層障害通知関数	Optional
5	ClcStart	ウォームスタート要求関数	Required
6	ClcSuspend	動作一時停止要求関数	Optional
7	ClcWakeUp	動作再開要求関数	Optional
8	ClcGetProData	プロトコル差異吸収処理部プロファイルデータ取得要求関数	Required
9	ClcGetStatus	プロトコル差異吸収処理部ステータスデータ取得要求関数	Optional
10	ClcInitAll	完全初期化要求関数	Optional
11	ClcStop	通信停止要求関数	Optional
12	ClcHalt	完全停止要求関数	Optional
13	ClcLowInit	下位通信ソフトウェア初期化要求関数	Optional
14	ClcLowRequestRun	下位通信ソフトウェア動作開始要求関数	Optional
15	ClcLowStart	下位通信ソフトウェアウォームスタート要求関数	Optional
16	ClcLowSuspend	下位通信ソフトウェア動作一時停止要求関数	Optional
17	ClcLowWakeUp	下位通信ソフトウェア動作再開要求関数	Optional
18	ClcGetLowProData	下位通信ソフトウェアプロファイルデータ取得要求関数	Required
19	ClcGetLowStatus	下位通信ソフトウェアステータスデータ取得要求関数	Required
20	ClcSendData	電文送信要求関数	Required
21	ClcGetSendResult	送信結果要求関数	Optional
22	ClcSendCancel	送信中止要求関数	Optional
23	ClcReceiveData	受信電文要求関数	Required
24	ClcGetNodeID	ノード ID 取得要求関数	Required
25	ClcSetNodeID	ノード ID 設定要求関数	Optional

26	ClcLowInitAll	下位通信ソフトウェア完全初期化要求関数	Optional
27	ClcLowStop	下位通信ソフトウェア通信停止要求関数	Optional
28	ClcLowHalt	下位通信ソフトウェア完全停止要求関数	Optional
29	ClcLowGetAddressT ableDataSize	下位通信ソフトウェアアドレステーブルデータサイ ズ取得関数	Optional
30	ClcLowGetAddressT ableData	下位通信ソフトウェアアドレステーブルデータ取得 関数	Optional
31	ClcLowSetMasterRo uterFlag	マスタールータ設定関数	Optional
32	ClcLowGetHardwar eAddress	ハードウェアアドレスデータ取得関数	Optional
33	ClcGetNodeIDList	NodeID リスト取得関数	Optional
34	ClcGetMasterRoute rInfo	マスタールータ情報取得関数	Optional
35	ClcLowReqToHard wareAddress	ハードウェアアドレス変換要求関数	Optional

## 4.2 C 言語用レベル2 ECHONET 共通下位通信インタフェース詳細仕様

本節では、表4.1にて示した各関数の詳細仕様を、以下の7つの項目について示す。

- (1) 名称  
関数名称を示す。
- (2) 機能  
機能を説明する。
- (3) 構文  
関数の構文を示す。
- (4) 説明  
引数や、変数の詳細仕様について説明する。
- (5) 戻り値  
戻り値を示す。
- (6) 使用する構造体  
構造体があれば、構造体の仕様を示す。
- (7) 注意事項・制限事項  
注意事項や制限事項があれば、示す。



## 4.2.1 ClcGetDevID

## (1) 名称

下位通信ソフトウェア搭載情報要求関数

## (2) 機能

プロトコル差異吸収処理部を介して操作が可能な下位通信ソフトウェアの数、および種類の識別を示す下位通信ソフトウェア ID 情報を要求する。

## (3) 構文

```

BOOL ClcGetDeviceID (
    unsigned char *device_num /* [OUT] 操作可能な下位通信ソフトウェア数*/
    unsigned char *device_idset /* [OUT] 操作可能な下位通信ソフトウェア ID */
)

```

## (4) 説明

\*device\_num : 操作可能な下位通信ソフト数へのポインタ

\*device\_idset : 操作可能な下位通信ソフトウェア ID 情報へのポインタ。ポインタの先には、device\_num で指定された数の情報が存在する。下位通信ソフトウェアの種類と対応する下位通信ソフトウェア ID の関係は、以下の通りである。

電灯線	0x11 ~ 0x1F
特定小電力無線	0x31 ~ 0x3F
拡張 HBS	0x41 ~ 0x4F
IrDA_Control	0x51 ~ 0x5F
LonTalk <sup>R</sup>	0x61 ~ 0x6F
Bluetooth <sup>TM</sup>	0x71 ~ 0x7F
Ethernet	0x81 ~ 0x8F

## (5) 戻り値

0 : 取得失敗  
1 : 取得成功

## (6) 使用する構造体

特に無し。

## (7) 注意事項・制限事項

この関数は、「初期化要求関数 : ClcInit」や「動作開始要求関数 : ClcRequestRun」より前に呼び出されることを前提とする。

## 4.2.2 ClcInit

## (1) 名称

初期化要求関数

## (2) 機能

存在する全下位通信ソフトウェア、および対応するプロトコル差異吸収処理部の MAC アドレス再取得を伴う初期化（コールドスタート）を要求する。この要求を受けたプロトコル差異吸収処理部は、コールドスタート可能な状態にある全ての下位通信ソフトウェアをコールドスタートし、通信停止状態に遷移させた後、初期化パラメータにて自身を初期化する。

## (3) 構文

```

BOOL ClcInit (
    CLC_INIT_DATA *init_data    /*[IN] 初期化パラメータへのポインタ */
)

```

## (4) 説明

\*init\_data : プロトコル差異吸収処理部の初期化パラメータへのポインタ。

## (5) 戻り値

0 : 初期化失敗  
1 : 初期化成功

## (6) 使用する構造体

```

typedef struct {
    short      sbuf_len;          /* 送信バッファサイズ */
    unsigned char *sbuf;        /* 送信バッファへのポインタ */
    short      rbuf_len;        /* 受信バッファサイズ */
    unsigned char *rbuf         /* 受信バッファへのポインタ */
    short      sholdtime,       /* プロトコル差異処理部送信電文最長保持時間情報 */
    short      rholdtime,       /* プロトコル差異処理部受信電文最長保持時間情報 */
    unsigned char clc_mode,     /* 動作モード指定
                                0x00 通常動作モード。
                                0x01 テスト・メンテナンスモード(詳細は、特に規定しない) */
} CLC_INIT_DATA

```

## (7) 注意事項・制限事項

本関数は、すでに全ての下位通信ソフトウェアがコールドスタート状態、またはウォームスタート状態にある場合は、初期化失敗を戻り値とする。

### 4.2.3 ClcRequestRun

(1) 名称

動作開始要求関数

(2) 機能

存在する全下位通信ソフトウェア、および対応するプロトコル差異吸収処理部の動作開始を要求する。この要求を受けたプロトコル差異吸収処理部は、全ての下位通信ソフトウェアを通常動作状態に遷移させる。

(3) 構文

BOOL ClcRequestRun ( void )

(4) 説明

特記無し。

(5) 戻り値

0 : 起動失敗  
1 : 起動成功

(6) 使用する構造体

無し。

(7) 注意事項・制限事項

本関数は、通信停止状態にある下位通信ソフトウェアのみを通常動作状態に遷移させる。1つでも遷移に成功した場合、戻り値は起動成功となる。

## 4.2.4 ClcSetTrouble

## (1) 名称

障害通知関数

## (2) 機能

アプリケーションソフトウェア及び ECHONET 通信処理部の障害 (異常) 状態をプロトコル差異吸収処理へ通知する。

## (3) 構文

```
BOOL ClcSetTrouble (  
    char htrouble_no      /*[IN] 上位レイヤトラブル番号 */  
)
```

## (4) 説明

htrouble\_no : トラブル番号。

- 1 トラブル解消
- 1 アプリケーションソフトウェア異常
- 2 ECHONET 通信処理部異常

## (5) 戻り値

0 : 通知受理失敗  
1 : 通知受理成功

## (6) 使用する構造体

無し。

## (7) 注意事項・制限事項

異常が通知されている間は、プロトコル差異吸収処理部では以下の動作を行う。

## ・電文受信時処理

下位通信ソフトウェアに対し、上位層の動作異常を通知後は、電文受信処理を行わないか、或いは受信電文を廃棄する。

## ・ECHOENT 通信制御処理部からの電文送信要求

エラーリターンする。

## 4.2.5 ClcStart

### (1) 名称

ウォームスタート要求関数

### (2) 機能

存在する全下位通信ソフトウェア、および対応するプロトコル差異吸収処理部の MAC アドレスを保持しての初期化(ウォームスタート)を要求する。この要求を受けたプロトコル差異吸収処理部は、ウォームスタート可能な状態にある全ての下位通信ソフトウェアをウォームスタートし、通信停止状態に遷移させる。

### (3) 構文

BOOL ClcStart ( void )

### (4) 説明

特記無し。

### (5) 戻り値

0 : 要求失敗  
1 : 要求成功

### (6) 使用する構造体

無し。

### (7) 注意事項・制限事項

本関数は、すでに全ての下位通信ソフトウェアがコールドスタート状態、またはウォームスタート状態にある場合は、要求失敗を戻り値とする。

本要求を受けた時、プロトコル差異吸収処理部は以下の処理を行う。

- ・送受信バッファのクリア
- ・上位レイヤト障害設定のリセット
- ・各種ステータス・ワーク領域のリセット

## 4.2.6 ClcSuspend

### (1) 名称

一次停止要求関数

### (2) 機能

存在する全下位通信ソフトウェア、および対応するプロトコル差異吸収処理部の一時停止を要求する。この要求を受けたプロトコル差異吸収処理部は、一時停止状態に遷移可能な全ての下位通信ソフトウェアを一時停止状態に遷移させる。

### (3) 構文

BOOL ClcSuspend (void)

### (4) 説明

特記無し。

### (5) 戻り値

0 : 一時停止失敗  
1 : 一時停止成功

### (6) 使用する構造体

無し。

### (7) 注意事項・制限事項

本関数は、全ての下位通信ソフトウェアが通常動作以外の状態にあるときは、一時停止失敗を戻り値とする。

下位通信ソフトウェア、プロトコル差異吸収処理部は、本要求を受けた時に電文送信途中であれば、一連の送信処理を終了し、一時停止状態に遷移する。また、電文の受信途中であれば、受信電文を廃棄し、処理を終了する。

一時停止中には以下の動作を行う。

- ・電文の受信  
行わない。
- ・ECHONET 通信制御処理部からの電文送信要求  
エラーリターンする。

## 4.2.7 ClcWakeUp

### (1) 名称

動作再開要求関数。

### (2) 機能

存在する全下位通信ソフトウェア、および対応するプロトコル差異吸収処理部の一時停止解除を要求する。この要求を受けたプロトコル差異吸収処理部は、全ての一時停止状態にある下位通信ソフトウェアを一時停止状態に遷移させる。

### (3) 構文

BOOL ClcWakeUp (void)

### (4) 説明

特記無し。

### (5) 戻り値

0 : 再開失敗

1 : 再開成功

### (6) 使用する構造体

無し。

### (7) 注意事項・制限事項

本関数は、全ての下位通信ソフトウェアが一時停止状態以外の状態にあるときは、再開失敗を戻り値とする。

## 4.2.8 ClcGetProData

## (1) 名称

プロトコル差異吸収処理部プロファイルデータ取得要求関数

## (2) 機能

プロトコル差異吸収処理部の、プロファイルデータを取得する。本関数にて要求するプロファイルデータは、第2部にて規定されているプロトコル差異吸収処理部プロファイルクラスのプロパティに関する情報の中の静的な情報に関するものである。

## (3) 構文

```
BOOL ClcGetProData (
    CLC_PRO_DATA *pro_data,    /*[OUT] プロファイルデータへのポインタ*/
)
```

## (4) 説明

\*pro\_data : プロトコル差異吸収処理部のプロファイルデータへのポインタ。

## (5) 戻り値

0 : 取得失敗  
1 : 取得成功

## (6) 使用する構造体

```
typedef struct {
    unsigned char    ver[3];        /* プロトコル差異吸収処理部バージョン No. */
    unsigned char    maker[3];     /* メーカーコード */
    short            slen;         /* 送信可能電文長 */
    short            rlen;         /* 受信可能電文長 */
} CLC_PRO_DATA
```

## (7) 注意事項・制限事項

特記無し。



## 4.2.9 ClcGetStatus

## (1) 名称

プロトコル差異吸収処理部ステータスデータ取得要求関数

## (2) 機能

プロトコル差異吸収処理部の、ステータスデータを要求する。本機能にて取得できるステータスデータは、保持している異常状態や動作モードといった動的情報である。

## (3) 構文

```

BOOL ClcGetStatus (
    CLC_STATUS *status /*[OUT] プロトコル差異吸収処理部のステータス */
)

```

## (4) 説明

**status** : プロトコル差異吸収処理部のステータスが返される。

## (5) 戻り値

0 : 取得失敗

1 : 取得成功

## (6) 使用する構造体

```

typedef struct {
    char upper_trouble; /* 上位レイヤ障害コード(0~127)
                        障害無し及びトラブル解消時は(0) */
    char clc_mode; /* 動作モードコード
                  通常動作時は(0)
                  メンテナンス等テストモード時は(1)
                  モニタモード時は(2) */
} CLC_STATUS;

```

## (7) 注意事項・制限事項

特記無し。

## 4.2.10 ClcInitAll

## (1) 名称

完全初期化要求関数

## (2) 機能

存在する全下位通信ソフトウェア、および対応するプロトコル差異吸収処理部のハウスコード情報、MAC アドレス再取得を伴う初期化（コールドスタート）を要求する。この要求を受けたプロトコル差異吸収処理部は、コールドスタート可能な状態にある全ての下位通信ソフトウェアをコールドスタートし、通信停止状態に遷移させた後、初期化パラメータにて自身を初期化する。

## (3) 構文

```

BOOL ClcInitAll (
    CLC_INIT_DATA *init_data    /*[IN] 初期化パラメータへのポインタ */
)

```

## (4) 説明

\*init\_data : プロトコル差異吸収処理部の初期化パラメータへのポインタ。

## (5) 戻り値

0 : 初期化失敗  
1 : 初期化成功

## (6) 使用する構造体

```

typedef struct {
    short      sbuf_len;          /* 送信バッファサイズ */
    short      *sbuf;            /* 送信バッファへのポインタ */
    short      rbuf_len;        /* 受信バッファサイズ */
    short      *rbuf;           /* 受信バッファへのポインタ */
    short      sholdtime,       /* プロトコル差異処理部送信電文最長保持時間情報 */
    short      rholdtime,       /* プロトコル差異処理部受信電文最長保持時間情報 */
    unsigned char clc_mode,     /* 動作モード指定
                                0x00 通常動作モード。
                                0x01 テスト・メンテナンスモード(詳細は、特に規定しない) */
} CLC_INIT_DATA

```

## (7) 注意事項・制限事項

本関数は、すでに全ての下位通信ソフトウェアがコールドスタート状態、またはウォームスタート状態、通信停止状態にある場合は、初期化失敗を戻り値とする。

ハウスコード情報を使用しない下位通信ソフトウェアには対しては、初期化要求と同じ処理となる。

## 4.2.1.1 ClcStop

### (1) 名称

通信停止要求関数

### (2) 機能

存在する全下位通信ソフトウェア、および対応するプロトコル差異吸収処理部の通信停止を要求する。この要求を受けたプロトコル差異吸収処理部は、通信停止状態に遷移可能な全ての下位通信ソフトウェアを通信停止状態に遷移させる。

### (3) 構文

BOOL ClcStop (void)

### (4) 説明

特記無し。

### (5) 戻り値

0 : 停止失敗  
1 : 停止成功

### (6) 使用する構造体

無し。

### (7) 注意事項・制限事項

本関数は、全ての下位通信ソフトウェアが通常動作以外の状態にあるときは、停止失敗を戻り値とする。

下位通信ソフトウェア、プロトコル差異吸収処理部は、本要求を受けた時に電文送信途中であれば、一連の送信処理を終了し、通信停止状態に遷移する。また、電文の受信途中であれば、受信電文を廃棄し、処理を終了する。

通信停止中には以下の動作を行う。

- ・電文の受信  
行わない。
- ・ECHONET 通信制御処理部からの送信要求  
エラーリターンする。

## 4.2.1.2 ClcHalt

### (1) 名称

完全停止要求関数

### (2) 機能

存在する全下位通信ソフトウェア、および対応するプロトコル差異吸収処理部の完全停止を要求する。この要求を受けたプロトコル差異吸収処理部は、停止状態に遷移可能な全ての下位通信ソフトウェアを停止状態に遷移させる。

### (3) 構文

BOOL ClcHalt (void)

### (4) 説明

特記無し。

### (5) 戻り値

0 : 停止失敗  
1 : 停止成功

### (6) 使用する構造体

無し。

### (7) 注意事項・制限事項

本関数は、全ての下位通信ソフトウェアがコールドスタート状態、ウォームスタート状態以外の状態にあるときは、停止失敗を戻り値とする。

下位通信ソフトウェア、プロトコル差異吸収処理部は、本要求を受けた時に電文送信途中であれば、一連の送信処理を終了し、通信停止状態に遷移する。また、電文の受信途中であれば、受信電文を廃棄し、処理を終了する。

通信停止中には以下の動作を行う。

- ・電文の受信  
行わない。
- ・ECHONET 通信制御処理部からの送信要求  
エラーリターンする。

## 4.2.1.3 ClcLowInit

## (1) 名称

下位通信ソフトウェア初期化要求関数

## (2) 機能

指定した下位通信ソフトウェア、および対応するプロトコル差異吸収処理部の MAC アドレス再取得を伴う初期化（コールドスタート）を要求する。この要求を受けたプロトコル差異吸収処理部は、指定された下位通信ソフトウェアをコールドスタートし、通信停止状態に遷移させた後、初期化パラメータにて自身を初期化する。

## (3) 構文

```

BOOL ClcLowInit (
    unsigned char    device_id,          /* [IN] 初期化対象下位通信ソフトウェアID */
    CLC_INIT_DATA *clcinit_data,       /* [IN] 初期化パラメータへのポインタ(1) */
    LOW_INIT_DATA *lowinit_data,      /* [IN] 初期化パラメータへのポインタ(2) */
    void            *low_init          /* [IN] 初期化パラメータへのポインタ(3) */
)

```

## (4) 説明

**device\_id** : 初期化対象下位通信ソフトウェア識別情報。

電灯線 0x11 ~ 0x1F

特定小電力無線 0x31 ~ 0x3F

拡張 HBS 0x41 ~ 0x4F

IrDA\_Control 0x51 ~ 0x5F

LonTalk<sup>R</sup> 0x61 ~ 0x6F

Bluetooth<sup>TM</sup> 0x71 ~ 0x7F

Ethernet 0x81 ~ 0x8F

**\*clcinit\_data** : プロトコル差異吸収処理部の初期化パラメータへのポインタ。

**\*lowinit\_data** : 下位通信ソフトウェア共通規定項目の初期化パラメータへのポインタ。

**\*low\_init** : 個々の下位通信ソフトウェア毎に異なる初期化パラメータへのポインタ。パラメータ内容は、個別下位通信ソフトウェア毎に規定する。

## (5) 戻り値

0 : 初期化失敗

1 : 初期化成功

## (6) 使用する構造体

```

typedef struct {
    short          sbuf_len;          /* 送信バッファサイズ */
    short          *sbuf;             /* 送信バッファへのポインタ */
    short          rbuf_len;         /* 受信バッファサイズ */
    short          *rbuf;            /* 受信バッファへのポインタ */
    short          sholdtime,        /* プロトコル差異 送信電文最長保持時間情報 */
    short          rholdtime,       /* プロトコル差異 受信電文最長保持時間情報 */
    unsigned char  clc_mode,        /* プロトコル差異 吸収処理部の動作モード指定 */
                                     0x00  通常動作モード
                                     0x01  テスト・メンテナンスモード
                                     (詳細は、特に規定しない。)
```

} CLC\_INIT\_DATA

```

typedef struct {
    short          sftoholdtime,     /* 下位通信ソフト 送信電文最長保持時間情報 */
    short          rftoholdtime,    /* 下位通信ソフト 受信電文最長保持時間情報 */
    unsigned char  low_mode,        /* 動作モード指定 */
    short          mac_len,         /* Mac アドレス長 */
    unsigned char  mac_ad[7],      /* Mac アドレス */
} LOW_INIT_DATA
```

mac\_ad[7]以外は、特に初期化データが無い場合には、NULLを設定する。

mac\_ad[7]は、mac\_len が NULL 設定の場合には、意味が無いものとする。(mac\_len が NULL の場合には、Mac アドレス設定無しとする。)

## (7) 注意事項・制限事項

本関数は、対象となる下位通信ソフトウェアがコールドスタート状態、またはウォームスタート状態にある場合は、初期化失敗を戻り値とする。

## 4.2.1.4 ClcLowRequestRun

## (1) 名称

下位通信ソフトウェア動作開始要求関数

## (2) 機能

指定する通信ソフトウェア、および対応するプロトコル差異吸収処理部の動作開始を要求する。この要求を受けたプロトコル差異吸収処理部は、指定された下位通信ソフトウェアを通常動作状態に遷移させる。

## (3) 構文

```
BOOL ClcLowRequestRun (  
    unsigned char device_id, /*[IN] 動作開始対象下位通信ソフトウェア ID */  
)
```

## (4) 説明

device\_id : 動作開始対象下位通信ソフトウェア識別情報。

電灯線	0x11 ~ 0x1F
特定小電力無線	0x31 ~ 0x3F
拡張 HBS	0x41 ~ 0x4F
IrDA_Control	0x51 ~ 0x5F
LonTalk <sup>R</sup>	0x61 ~ 0x6F
Bluetooth <sup>TM</sup>	0x71 ~ 0x7F
Ethernet	0x81 ~ 0x8F

## (5) 戻り値

0 : 起動失敗  
1 : 起動成功

## (6) 使用する構造体

無し。

## (7) 注意事項・制限事項

本関数は、指定した下位通信ソフトウェアが通信停止状態にない場合は、戻り値を起動失敗とする。

## 4.2.15 ClcLowStart

## (1) 名称

下位通信ソフトウェアウォームスタート要求関数

## (2) 機能

指定した下位通信ソフトウェアのMACアドレスを保持しての初期化(ウォームスタート)を要求する。この要求を受けたプロトコル差異吸収処理部は、指定された下位通信ソフトウェアをウォームスタートし、通信停止状態に遷移させる。

## (3) 構文

```

BOOL ClcLowStart (
    unsigned char    device_id,    /*[IN] ウォームスタート対象
                                   下位通信ソフトウェアID */
)

```

## (4) 説明

device\_id : ウォームスタート対象下位通信ソフトウェア識別情報。

電灯線	0x11 ~ 0x1F
特定小電力無線	0x31 ~ 0x3F
拡張HBS	0x41 ~ 0x4F
IrDA_Control	0x51 ~ 0x5F
LonTalk <sup>R</sup>	0x61 ~ 0x6F
Bluetooth <sup>TM</sup>	0x71 ~ 0x7F
Ethernet	0x81 ~ 0x8F

## (5) 戻り値

0 : 要求失敗  
1 : 要求成功

## (6) 使用する構造体

無し。

## (7) 注意事項・制限事項

本関数は、対象となる下位通信ソフトウェアがコールドスタート状態、またはウォームスタート状態にある場合は、要求失敗を戻り値とする。

本要求を受けた時、指定された下位通信ソフトウェアに対応するプロトコル差異吸収処理部は以下の処理を行う。

- ・送受信バッファのクリア
- ・上位レイヤト障害設定のリセット
- ・各種ステータス・ワーク領域のリセット



## 4.2.16 ClcLowSuspend

## (1) 名称

下位通信ソフトウェア一時停止要求関数

## (2) 機能

指定する下位通信ソフトウェア、および対応するプロトコル差異吸収処理部の一時停止を要求する。この要求を受けたプロトコル差異吸収処理部は、指定された下位通信ソフトウェアを一時停止状態に遷移させる。

## (3) 構文

```
BOOL ClcLowSuspend (
    unsigned char device_id /*[IN] 一時停止対象下位通信ソフトウェア ID */
)
```

## (4) 説明

device\_id : 一時停止対象下位通信ソフトウェア識別情報。

電灯線	0x11 ~ 0x1F
特定小電力無線	0x31 ~ 0x3F
拡張 HBS	0x41 ~ 0x4F
IrDA_Control	0x51 ~ 0x5F
LonTalk <sup>R</sup>	0x61 ~ 0x6F
Bluetooth <sup>TM</sup>	0x71 ~ 0x7F
Ethernet	0x81 ~ 0x8F

## (5) 戻り値

0 : 一時停止失敗  
1 : 一時停止成功

## (6) 使用する構造体

無し。

## (7) 注意事項・制限事項

本関数は、指定された下位通信ソフトウェアが通常動作以外の状態にあるときは、一時停止失敗を戻り値とする。

指定された下位通信ソフトウェア、プロトコル差異吸収処理部は、本要求を受けた時に電文送信途中であれば、一連の送信処理を終了し、一時停止状態に遷移する。また、電文の受信途中であれば、受信電文を廃棄し、処理を終了する。

一時停止中には以下の動作を行う。

- ・電文の受信  
行わない。
- ・ECHONET 通信制御処理部からの送信要求  
エラーリターンする。

## 4.2.17 ClcLowWakeUp

## (1) 名称

下位通信ソフトウェア動作再開要求関数。

## (2) 機能

指定する下位通信ソフトウェア、および対応するプロトコル差異吸収処理部の一時停止解除を要求する。この要求を受けたプロトコル差異吸収処理部は、指定された下位通信ソフトウェアを通常動作状態に遷移させる。

## (3) 構文

```
BOOL ClcLowWakeUp (
    unsigned char device_id /*[IN] 動作再開対象下位通信ソフトウェア ID */
)
```

## (4) 説明

**device\_id** : 動作再開対象下位通信ソフトウェア識別情報。

電灯線	0x11 ~ 0x1F
特定小電力無線	0x31 ~ 0x3F
拡張 HBS	0x41 ~ 0x4F
IrDA_Control	0x51 ~ 0x5F
LonTalk <sup>R</sup>	0x61 ~ 0x6F
Bluetooth <sup>TM</sup>	0x71 ~ 0x7F
Ethernet	0x81 ~ 0x8F

## (5) 戻り値

0 : 再開失敗  
1 : 再開成功

## (6) 使用する構造体

無し。

## (7) 注意事項・制限事項

本関数は、指定された下位通信ソフトウェアが一時停止状態以外の状態にあるときは、再開失敗を戻り値とする。

## 4.2.18 ClcGetLowProData

## (1) 名称

下位通信ソフトウェアプロファイルデータ取得要求関数

## (2) 機能

指定した下位通信ソフトウェアのプロファイルデータを取得する。本関数にて要求するプロファイルデータは、ソフトウェア開発メーカー名やバージョン No.等の下位通信ソフトウェアプロファイルクラスのプロパティ値情報である。

## (3) 構文

```

BOOL ClcGetLowProData (
    unsigned char    device_id,      /*[IN]  下位通信ソフトウェア ID */
    LOW_PRO_DATA *pro_data,        /*[OUT] プロファイルデータ*/
)

```

## (4) 説明

**device\_id** : プロファイルデータ取得対象下位通信ソフトウェア識別情報。

電灯線 0x11 ~ 0x1F

特定小電力無線 0x31 ~ 0x3F

拡張 HBS 0x41 ~ 0x4F

IrDA\_Control 0x51 ~ 0x5F

LonTalk<sup>R</sup> 0x61 ~ 0x6F

Bluetooth<sup>TM</sup> 0x71 ~ 0x7F

Ethernet 0x81 ~ 0x8F

**\*pro\_data** : 指定された下位通信ソフトウェアのプロファイルデータ構造体へのポインタ。

## (5) 戻り値

0 : 取得失敗

1 : 取得成功

## (6) 使用する構造体

```

typedef struct {
    unsigned char    kind;          /* 下位媒体種別
                                   電灯線 : 0x31
                                   小電力無線 : 0x33   拡張 HBS : 0x34
                                   IrDAControl : 0x35
                                   LonTalkR : 0x36 */
    unsigned char    ver[3];       /* 下位通信ソフトウェアバージョン No. */
    unsigned char    maker[3];     /* メーカーコード */
    short            mac_len;       /* MAC アドレス長 */
    unsigned char    mac_ad[7];    /* MAC アドレス */
    unsigned char    mac_mask[7];  /* MAC アドレスマスク値 */
    short            house_len;    /* ハウスコード長 */
    short            *housecode;   /* ハウスコード情報へのポインタ */
    short            slen;         /* 送信可能電文長 */
    short            rlen;         /* 受信可能電文長 */
    short            broad;        /* 同報機能の有無(0 : 機能無、1 : 機能有) */
    short            baud;         /* 伝送レート */
} LOW_PRO_DATA

```

( 7 ) 注意事項・制限事項

特記無し。

## 4.2.19 ClcGetLowStatus

## (1) 名称

下位通信ソフトウェアステータスデータ取得要求関数

## (2) 機能

指定した下位通信ソフトウェアの、ステータスデータを要求する。本関数にて取得できるステータスデータは、保持している異常状態や動作モードといった動的情報である。

## (3) 構文

```

BOOL ClcGetLowStatus (
    unsigned char device_id, /*[IN] 下位通信ソフトウェア ID */
    LOW_STATUS *status /*[OUT] 下位通信ソフトウェアのステータス */
)

```

## (4) 説明

device\_id : ステータス取得対象下位通信ソフトウェア識別情報。

電灯線 0x11 ~ 0x1F

特定小電力無線 0x31 ~ 0x3F

拡張HBS 0x41 ~ 0x4F

IrDA\_Control 0x51 ~ 0x5F

LonTalk<sup>R</sup> 0x61 ~ 0x6F

Bluetooth<sup>TM</sup> 0x71 ~ 0x7F

Ethernet 0x81 ~ 0x8F

\*status : 下位通信ソフトウェアステータスデータ構造体へのポインタ。

## (5) 戻り値

0 : 取得失敗

1 : 取得成功

## (6) 使用する構造体

```

typedef struct {
    char upper_trouble; /* 上位レイヤ障害コード(0~127)
                        障害無し及びトラブル解消時は(0) */
    char low_trouble; /* 下位通信ソフトウェア部障害コード(0~127)
                      障害無し及びトラブル解消時は(0) */
    char low_mode; /* 動作モードコード
                  通常動作時 : 0
                  メンテナンス等テストモード時 : 1
                  モニタモード時 : 2 */
    short state; /* 下位通信ソフトウェア部状態
                 LOW_STS_STOP : 0 停止状態
                 LOW_STS_INI : 1 コールドスタート状態
                 LOW_STS_RUN : 2 通常動作状態
                 LOW_STS_ESTOP : 3 エラー停止状態
                 LOW_STS_RST : 4 ウォームスタート状態
                 LOW_STS_CSTOP : 5 通信停止状態
                 LOW_STS_SPD : 6 一時停止状態
                 */
} LOW_STATUS;

```

## (7) 注意事項・制限事項

特記無し。

## 4.2.20 ClcSendData

## (1) 名称

電文送信関数

## (2) 機能

指定した下位通信ソフトウェアを使用して、ECHONET 電文を送信することを要求する。指定された下位通信ソフトウェアに対応するプロトコル差異吸収処理部は、下位通信ソフトウェアの仕様に従った電文を作成し、下位通信ソフトウェアに対してその電文送信を要求する。

## (3) 構文

```
short ClcSendData (
    unsigned char    device_id,      /*[IN]  下位通信ソフトウェア ID */
    const unsigned char *buf,        /*[IN]  送信データへのポインタ */
    short snd_sz,      /*[IN]  送信データサイズ */
    unsigned char dnode_id,         /*[IN]  送信先 NodeID */
    unsigned char broad,            /*[IN]  同報指定情報 */
)
```

## (4) 説明

**device\_id** : 電文送信対象下位通信ソフトウェア識別情報。

電灯線	0x11 ~ 0x1F
特定小電力無線	0x31 ~ 0x3F
拡張 HBS	0x41 ~ 0x4F
IrDA_Control	0x51 ~ 0x5F
LonTalk <sup>R</sup>	0x61 ~ 0x6F
Bluetooth <sup>TM</sup>	0x71 ~ 0x7F
Ethernet	0x81 ~ 0x8F

**\*buf** : 送信する ECHONET 電文へのポインタを指定する。ここで渡される ECHONET 電文は、第2部4.2節で規定されている ECHONET 通信処理部間でやり取りされるものである。

**snd\_sz** : 送信データサイズ (送信 ECHONET 電文のサイズ) を指定する。

**dnode\_id** : 自サブネット内送信先 NodeID を指定する。  
同報指定情報 (broad) が 0x00 の場合、デフォルトルータのノード ID、または適切な経路上にあるルータの NodeID を指定する。同報指定情報 (broad) が 0xFF の場合、0xFF もしくは該当する同報対象指定コードを指定する。

**broad** : 同報指定を行う。  
0x00 : 同報指定無し、あるいは指定サブネット内一斉同報  
0xFF : ドメイン内一斉同報、あるいは自サブネット内一斉指定

## (5) 戻り値

CLC_BUFFER_FULL	: 0	バッファフルエラー
CLC_NO_ERROR	: 1	送信受理
CLC_BUFFER_SIZE_ERROR	: 2	バッファサイズエラー
CLC_STATE_ERROR	: 3	下位通信ソフトウェア内部エラー
CLC_ADAPTER_ERROR	: 4	機器アダプタ処理失敗

(6) 使用する構造体

無し。

(7) 注意事項・制限事項

本関数は、指定された下位通信ソフトウェアが通常動作状態にない場合は、下位通信ソフトウェア内部エラーを戻り値とする。

## 4.2.2.1 ClcGetSendResult

## (1) 名称

送信結果取得関数

## (2) 機能

指定した下位通信ソフトウェアが、電文送信関数 (ClcSendData) に応じて最近に行った ECHONET 電文の送信結果を要求する。指定された下位通信ソフトウェアに対応するプロトコル差異吸収処理部は、下位通信ソフトウェアに送信結果を要求する。

## (3) 構文

```
short ClcGetSendResult (
    unsigned char device_id, /*[IN] 下位通信ソフトウェア ID */
    unsigned char *result /*[OUT] 送信結果へのポインタ */
)
```

## (4) 説明

**device\_id** : 送信結果取得対象下位通信ソフトウェア識別情報。

電灯線	0x11 ~ 0x1F
特定小電力無線	0x31 ~ 0x3F
拡張 HBS	0x41 ~ 0x4F
IrDA_Control	0x51 ~ 0x5F
LonTalk <sup>R</sup>	0x61 ~ 0x6F
Bluetooth <sup>TM</sup>	0x71 ~ 0x7F
Ethernet	0x81 ~ 0x8F

**\*result** : 送信結果へのポインタ

0x00	: 送信成功
0x01	: 送信失敗
0xFF	: 無応答

## (5) 戻り値

CLC_CANCEL	: 0	送信中止時
CLC_NO_ERROR	: 1	正常時
CLC_NO_SENDEND	: 2	送信中 (送信未完了)
CLC_INTERNAL_ERROR	: 3	下位通信ソフトウェア内部エラー
CLC_ADAPTER_ERROR	: 4	機器アダプタ処理失敗

## (6) 使用する構造体

無し。

## (7) 注意事項・制限事項

本関数は、指定された下位通信ソフトウェアが通常動作状態にない場合は、下位通信ソフトウェア内部エラーを戻り値とする。

**result** が意味を持つのは、戻り値が正常時 (NO\_ERROR) の場合のみである。



## 4.2.2.2 ClcSendCancel

## (1) 名称

送信中止要求関数

## (2) 機能

指定した下位通信ソフトウェアが、電文送信関数 (ClcSendData) に応じて行っている ECHONET 電文の送信中心を要求する。指定された下位通信ソフトウェアに対応するプロトコル差異吸収処理部は、下位通信ソフトウェアに送信中止を要求する。

## (3) 構文

```
short ClcSendCancel (
    unsigned char device_id /*[IN] 下位通信ソフトウェア ID */
)
```

## (4) 説明

device\_id : 送信中止要求対象下位通信ソフトウェア識別情報。

電灯線	0x11 ~ 0x1F
特定小電力無線	0x31 ~ 0x3F
拡張 HBS	0x41 ~ 0x4F
IrDA_Control	0x51 ~ 0x5F
LonTalk <sup>R</sup>	0x61 ~ 0x6F
Bluetooth <sup>TM</sup>	0x71 ~ 0x7F
Ethernet	0x81 ~ 0x8F

## (5) 戻り値

CLC_CANCEL	: 0	送信済みにより中止処理実行無し
CLC_NO_ERROR	: 1	正常時
CLC_INTERNAL_ERROR	: 3	下位通信ソフトウェア内部エラー
CLC_ADAPTER_ERROR	: 4	機器アダプタ処理失敗

## (6) 使用する構造体

無し。

## (7) 注意事項・制限事項

本関数は、指定された下位通信ソフトウェアが通常動作状態にない場合は、下位通信ソフトウェア内部エラーを戻り値とする。

## 4.2.2.3 ClcReceiveData

## (1) 名称

受信電文要求関数。

## (2) 機能

指定した下位通信ソフトウェアが保持している、受信 ECHONET 電文を要求する。指定された下位通信ソフトウェアに対応するプロトコル差異吸収処理部は、下位通信ソフトウェアに対して受信している電文を要求する。

## (3) 構文

```
short   ClcReceiveData (
  unsigned char  device_id, /*[IN]  下位通信ソフトウェア ID */
  unsigned char *buf,      /*[IN]  受信バッファへのポインタ */
  short  buf_sz,          /*[IN]  受信バッファサイズ */
  short  *rcv_cz,         /*[OUT] 受信データサイズ */
  unsigned char *snode_id /*[OUT] 送信元 NodeID */
)
```

## (4) 説明

**device\_id** : 受信電文要求対象下位通信ソフトウェア識別情報。

電灯線 0x11 ~ 0x1F

特定小電力無線 0x31 ~ 0x3F

拡張 HBS 0x41 ~ 0x4F

IrDA\_Control 0x51 ~ 0x5F

LonTalk<sup>R</sup> 0x61 ~ 0x6F

Bluetooth<sup>TM</sup> 0x71 ~ 0x7F

Ethernet 0x81 ~ 0x8F

**\*buf** : 受信バッファへのポインタを指定する。

**buf\_sz** : 受信バッファのサイズを指定する。

**rcv\_sz** : 受信データのサイズが返される。

**snode\_id** : 自サブネット内での送信元 NodeID が返される。他サブネットから送信された場合、ルータの NodeID となる。

## (5) 戻り値

CLC_NO_RECEIVE	: 0	受信データ無し
CLC_NO_ERROR	: 1	正常時 (受信データ有り)
CLC_BUFFER_SIZE_ERROR	: 2	バッファサイズエラー
CLC_INTERNAL_ERROR	: 3	下位通信ソフトウェア内部エラー

## (6) 使用する構造体

無し。

## (7) 注意事項・制限事項

本関数は、指定された下位通信ソフトウェアが通常動作状態にない場合は、下位通信ソフトウェア内部エラーを戻り値とする。

## 4.2.2.4 ClcGetNodeID

## (1) 名称

NodeID 取得要求関数

## (2) 機能

指定した下位通信ソフトウェアに対応するプロトコル差異吸収処理部が保持している NodeID 情報を要求する。

## (3) 構文

```

BOOL ClcGetNodeID (
    unsigned char device_id, /*[IN] 下位通信ソフトウェア ID */
    unsigned char *node_id, /*[OUT] NodeID */
)

```

## (4) 説明

**device\_id** : NodeID 情報取得対象下位通信ソフトウェア識別情報。

電灯線 0x11 ~ 0x1F

特定小電力無線 0x31 ~ 0x3F

拡張 HBS 0x41 ~ 0x4F

IrDA\_Control 0x51 ~ 0x5F

LonTalk<sup>R</sup> 0x61 ~ 0x6F

Bluetooth<sup>TM</sup> 0x71 ~ 0x7F

Ethernet 0x81 ~ 0x8F

**node\_id** : NodeID コード。

## (5) 戻り値

0 : NodeID 取得失敗

1 : NodeID 取得成功

## (6) 使用する構造体

無し。

## (7) 注意事項・制限事項

保持している NodeID が無い場合、本関数の戻り値は、NodeID 取得失敗となる。

## 4.2.2.5 ClcSetNodeID

## (1) 名称

NodeID 設定要求関数

## (2) 機能

指定した下位通信ソフトウェアに対応するプロトコル差異吸収処理部が保持している NodeID 情報を更新する。

## (3) 構文

```
short ClcSetNodeID (
    unsigned char device_id, /*[IN] 下位通信ソフトウェア ID */
    unsigned char node_id   /*[IN] NodeID 情報 */
)
```

## (4) 説明

プロトコル差異吸収処理部にて認識している下位通信ソフトウェアに対応する NodeID を設定する。

device\_id : NodeID 情報設定対象下位通信ソフトウェア識別情報。

電灯線 0x11 ~ 0x1F

特定小電力無線 0x31 ~ 0x3F

拡張 HBS 0x41 ~ 0x4F

IrDA\_Control 0x51 ~ 0x5F

LonTalk<sup>R</sup> 0x61 ~ 0x6F

Bluetooth<sup>TM</sup> 0x71 ~ 0x7F

Ethernet 0x81 ~ 0x8F

node\_id : NodeID コード。

## (5) 戻り値

CLC\_NO\_CHANGE : 0 ソフトウェアでは変更不可

CLC\_NO\_ERROR : 1 正常時

CLC\_INTERNAL\_ERROR : 3 下位通信ソフトウェア内部エラー

## (6) 使用する構造体

無し。

## (7) 注意事項・制限事項

NodeID を更新したプロトコル差異吸収処理部は、対応する下位通信ソフトウェアの MAC アドレスを更新した NodeID の値に応じて更新する。

## 4.2.2.6 ClcLowInitAll

## (1) 名称

下位通信ソフトウェア完全初期化要求関数

## (2) 機能

指定する下位通信ソフトウェア、および対応するプロトコル差異吸収処理部のハウスコード情報、MAC アドレス再取得を伴う初期化(コールドスタート)を要求する。この要求を受けたプロトコル差異吸収処理部は、指定された下位通信ソフトウェアをコールドスタートし、通信停止状態に遷移させた後、初期化パラメータにて自身を初期化する。

## (3) 構文

```

BOOL ClcLowInitAll (
    unsigned char    device_id,      /*[IN]  下位通信ソフトウェア ID */
    CLC_INIT_DATA   *init_data,     /*[IN]  初期化パラメータへのポインタ */
    LOW_INIT_DATA   *lowinit_data,  /*[IN]  初期化パラメータへのポインタ(2) */
    void            *low_init       /*[IN]  初期化パラメータへのポインタ(3) */
)

```

## (4) 説明

**device\_id** : 完全初期化要求対象下位通信ソフトウェア識別情報。

電灯線 0x11 ~ 0x1F

特定小電力無線 0x31 ~ 0x3F

拡張 HBS 0x41 ~ 0x4F

IrDA\_Control 0x51 ~ 0x5F

LonTalk<sup>R</sup> 0x61 ~ 0x6F

Bluetooth<sup>TM</sup> 0x71 ~ 0x7F

Ethernet 0x81 ~ 0x8F

**\*init\_data** : プロトコル差異吸収処理部の初期化パラメータへのポインタ。

**\*lowinit\_data** : 下位通信ソフトウェア共通規定項目の初期化パラメータへのポインタ。

**\*low\_init** : 個々の下位通信ソフトウェア毎に異なる初期化パラメータへのポインタ。パラメータ内容は、個別下位通信ソフトウェア毎に規定する。

## (5) 戻り値

0 : 初期化失敗

1 : 初期化成功

## (6) 使用する構造体

```

typedef struct {
    short          sbuf_len;      /* 送信バッファサイズ */
    short          *sbuf;        /* 送信バッファへのポインタ */
    short          rbuf_len;     /* 受信バッファサイズ */
    short          *rbuf;        /* 受信バッファへのポインタ */
    short          sholdtime,    /* プロトコル差異 送信電文最長保持時間情報 */
    short          rholdtime,    /* プロトコル差異 受信電文最長保持時間情報 */
    unsigned char  clc_mode,     /* プロトコル差異 吸収処理部の動作モード指定 */
                                     0x00  通常動作モード
                                     0x01  テスト・メンテナンスモード
                                     (詳細は、特に規定しない。)
```

} CLC\_INIT\_DATA

```

typedef struct {
    short          sfholdtime,   /* 下位通信ソフトウェア送信電文最長保持時間情報 */
    short          rfholdtime,   /* 下位通信ソフトウェア受信電文最長保持時間情報 */
    unsigned char  low_mode,     /* 動作モード指定 */
    short          mac_len,      /* MAC アドレス長 */
    unsigned char  mac_ad[7],    /* MAC アドレス */
} LOW_INIT_DATA
```

mac\_ad[7]以外は、特に初期化データが無い場合には、NULLを設定する。

mac\_ad[7]は、mac\_lenがNULL設定の場合には、意味が無いものとする。(mac\_lenがNULLの場合には、MACアドレス設定無しとする。)

## (7) 注意事項・制限事項

本関数は、指定された下位通信ソフトウェアがコールドスタート状態、またはウォームスタート状態、通信停止状態にある場合は、初期化失敗を戻り値とする。

ハウスコード情報を使用しない下位通信ソフトウェアには対しては、初期化要求と同じ処理となる。

## 4.2.27 ClcLowStop

## (1) 名称

下位通信ソフトウェア通信停止要求関数

## (2) 機能

指定する下位通信ソフトウェア、および対応するプロトコル差異吸収処理部の通信停止を要求する。この要求を受けたプロトコル差異吸収処理部は、指定された下位通信ソフトウェアを通信停止状態に遷移させる。

## (3) 構文

```
BOOL ClcLowStop (
    unsigned char device_id /*[IN]下位通信ソフトウェア ID */
)
```

## (4) 説明

device\_id : 通信停止対象下位通信ソフトウェア識別情報。

電灯線	0x11 ~ 0x1F
特定小電力無線	0x31 ~ 0x3F
拡張 HBS	0x41 ~ 0x4F
IrDA_Control	0x51 ~ 0x5F
LonTalk <sup>R</sup>	0x61 ~ 0x6F
Bluetooth <sup>TM</sup>	0x71 ~ 0x7F
Ethernet	0x81 ~ 0x8F

## (5) 戻り値

0 : 通信停止失敗  
1 : 通信停止成功

## (6) 使用する構造体

無し。

## (7) 注意事項・制限事項

本関数は、指定された下位通信ソフトウェアが通常動作以外の状態にあるときは、通信停止失敗を戻り値とする。

指定された下位通信ソフトウェア、プロトコル差異吸収処理部は、本要求を受けた時に電文送信途中であれば、一連の送信処理を終了し、通信停止状態に遷移する。また、電文の受信途中であれば、受信電文を廃棄し、処理を終了する。

一時停止中には以下の動作を行う。

- ・電文の受信  
行わない。
- ・ECHONET 通信制御処理部からの電文送信要求  
エラーリターンする。

## 4.2.28 ClcLowHalt

## (1) 名称

下位通信ソフトウェア完全停止要求関数

## (2) 機能

指定する下位通信ソフトウェア、および対応するプロトコル差異吸収処理部の完全停止を要求する。この要求を受けたプロトコル差異吸収処理部は、指定された下位通信ソフトウェアを停止状態に遷移させる。

## (3) 構文

```
BOOL ClcLowHalt (
    unsigned char    device_id    /*[IN]下位通信ソフトウェア ID */
)
```

## (4) 説明

device\_id : 完全停止対象下位通信ソフトウェア識別情報。

電灯線	0x11 ~ 0x1F
特定小電力無線	0x31 ~ 0x3F
拡張 HBS	0x41 ~ 0x4F
IrDA_Control	0x51 ~ 0x5F
LonTalk <sup>R</sup>	0x61 ~ 0x6F
Bluetooth <sup>TM</sup>	0x71 ~ 0x7F
Ethernet	0x81 ~ 0x8F

## (5) 戻り値

0 : 完全停止失敗  
1 : 完全停止成功

## (6) 使用する構造体

無し。

## (7) 注意事項・制限事項

本関数は、指定された下位通信ソフトウェアがコールドスタート状態、ウォームスタート状態にあるときは、停止失敗を戻り値とする。

下位通信ソフトウェア、プロトコル差異吸収処理部は、本要求を受けた時に電文送信途中であれば、一連の送信処理を終了し、停止状態に遷移する。また、電文の受信途中であれば、受信電文を廃棄し、処理を終了する。

停止中には以下の動作を行う。

- ・電文の受信  
行わない。
- ・ECHONET 通信制御処理部からの電文送信要求  
エラーリターンする。



## 4.2.29 ClcLowGetAddressTableDataSize

## (1) 名称

下位通信ソフトウェアアドレステーブルデータサイズ取得関数

## (2) 機能

下位通信ソフトウェアで保持している下位アドレステーブルデータの組数を取得する。

## (3) 構文

```

BOOL ClcLowGetAddressTableDataSize (
    unsigned char device_id,          /*[IN] 下位通信ソフトウェア ID */
    unsigned char* data_number,      /*[OUT] 宛先の組数 */
)

```

## (4) 説明

出力データはデータ組数のポインタからなる。

device\_id : 下位通信ソフトウェア識別情報。

電灯線 0x11 ~ 0x1F

特定小電力無線 0x31 ~ 0x3F

拡張 HBS 0x41 ~ 0x4F

IrDA\_Control 0x51 ~ 0x5F

LonTalk<sup>R</sup> 0x61 ~ 0x6FBluetooth<sup>TM</sup> 0x71 ~ 0x7F

Ethernet 0x81 ~ 0x8F

data\_number : 下位アドレステーブルデータで保持しているアドレステーブルの組数のポインタ。

## (5) 戻り値

0 : 異常時

1 : 正常時

## (6) 使用する構造体

無し。

## (7) 注意事項・制限事項

## 4.2.30 ClcLowGetAddressTableData

## (1) 名称

下位通信ソフトウェアアドレステーブルデータ取得関数

## (2) 機能

下位通信ソフトウェアで保持している下位アドレステーブルデータを取得する。

## (3) 構文

```

BOOL ClcLowGetAddressTableData (
    unsigned char device_id, /*[IN] 下位通信ソフトウェア ID */
    unsigned char* data_number, /*[IN/OUT] 宛先の組数 */
    ADDRESSTABLE* addresstable, /*[OUT] アドレステーブル構造体 */
)

```

## (4) 説明

入力データ(data\_number)は、ClcLowGetAddressTableDataSize で取得したアドレステーブルの組数のポインタ。出力データは、実際に格納したアドレステーブルの組数と、各データ組のハードウェアアドレス、NodeID、及びマスターータであることを示すフラグからなる構造体の配列データからなる。

device\_id : 下位通信ソフトウェア識別情報。

電灯線 0x11 ~ 0x1F

特定小電力無線 0x31 ~ 0x3F

拡張 HBS 0x41 ~ 0x4F

IrDA\_Control 0x51 ~ 0x5F

LonTalk<sup>R</sup> 0x61 ~ 0x6F

Bluetooth<sup>TM</sup> 0x71 ~ 0x7F

Ethernet 0x81 ~ 0x8F

data\_number : 下位アドレステーブルデータで保持しているアドレステーブルの組数のポインタを示す。

addresstable : 下位アドレステーブルデータで保持しているハードウェアアドレス、NodeID、マスターータであることを示すフラグを収容するアドレステーブル構造体の配列の先頭ポインタ。

## (5) 戻り値

0 : 異常時

1 : 正常時

## (6) 使用する構造体

```

typedef struct{
    unsigned char hardwareaddress_size; /* ハードウェアアドレスのデータサイズ */
    unsigned char hardwareaddress[8]; /* ハードウェアアドレス。下位バイト詰めで格納。 */
    unsigned char node_id; /* NodeID */
    unsigned char masterrouter_flag; /* 対応するノードがマスターータであるか否かを示す識別子。マスターータであれば1、そうでなければ0。 */
}ADDRESSTABLE

```

## (7) 注意事項・制限事項

無し。

## 4.2.3.1 ClcLowSetMasterRouterFlag

## (1) 名称

マスタールータ設定関数

## (2) 機能

自ノードがマスタールータであるか否かを下位通信ソフトウェアに通知する。

## (3) 構文

```

BOOL ClcLowSetMasterRouterFlag (
    unsigned char    device_id,          /*[IN] 下位通信ソフトウェア ID */
    unsigned char    masterRouter_Flag /*[IN] マスタールータ識別フラグ */
)

```

## (4) 説明

device\_id : 下位通信ソフトウェア識別情報。

電灯線	0x11 ~ 0x1F
特定小電力無線	0x31 ~ 0x3F
拡張 HBS	0x41 ~ 0x4F
IrDA_Control	0x51 ~ 0x5F
LonTalk <sup>R</sup>	0x61 ~ 0x6F
Bluetooth <sup>TM</sup>	0x71 ~ 0x7F
Ethernet	0x81 ~ 0x8F

masterRouter\_Flag : マスタールータであれば 1、そうでなければ 0 を指定する。

## (5) 戻り値

0 : 異常時  
1 : 正常時

## (6) 使用する構造体

無し。

## (7) 注意事項・制限事項

無し。

## 4.2.3.2 ClcLowGetHardwareAddress

## (1) 名称

ハードウェアアドレスデータ取得関数

## (2) 機能

下位通信ソフトウェアに対して保持しているハードウェアアドレスデータを取得する。

## (3) 構文

```

BOOL ClcLowGetHardwareAddress (
    unsigned char    device_id,          /*[IN]  下位通信ソフトウェア ID */
    unsigned char*   hardwareaddresssize, /*[OUT] ハードウェアアドレスのサイズ*/
    unsigned char*   hardwareaddress    /*[OUT] ハードウェアアドレス */
)

```

## (4) 説明

出力データはハードウェアアドレスである。

device\_id : 下位通信ソフトウェア識別情報。

電灯線 0x11 ~ 0x1F

特定小電力無線 0x31 ~ 0x3F

拡張 HBS 0x41 ~ 0x4F

IrDA\_Control 0x51 ~ 0x5F

LonTalk<sup>R</sup> 0x61 ~ 0x6FBluetooth<sup>TM</sup> 0x71 ~ 0x7F

Ethernet 0x81 ~ 0x8F

hardwareaddresssize : 下位アドレステーブルデータで保持しているハードウェアアドレスのサイズのポインタを示す。

hardwareaddress : 下位アドレステーブルデータで保持しているアドレステーブルに対応するハードウェアアドレスの配列のポインタを示す。

## (5) 戻り値

0 : 異常時

1 : 正常時

## (6) 使用する構造体

無し。

## (7) 注意事項・制限事項

無し。

## 4.2.3.3 ClcGetNodeIDList

## (1) 名称

NodeID リスト取得関数

## (2) 機能

下位通信ソフトウェアが保持する NodeID リストを取得する。

## (3) 構文

```

BOOL ClcGetNodeIDList (
    unsigned char device_id, /*[IN] 下位通信ソフトウェア ID */
    unsigned char *node_id_list /*[OUT] NodeID リスト */
)

```

## (4) 説明

device\_id : 下位通信ソフトウェア識別情報。

電灯線	0x11 ~ 0x1F
特定小電力無線	0x31 ~ 0x3F
拡張 HBS	0x41 ~ 0x4F
IrDA_Control	0x51 ~ 0x5F
LonTalk <sup>R</sup>	0x61 ~ 0x6F
Bluetooth <sup>TM</sup>	0x71 ~ 0x7F
Ethernet	0x81 ~ 0x8F

\*node\_id\_list : 下位通信ソフトウェアが保持している NodeID リストの配列(要素数 32)へのポインタである。以下のようなデータとする。

Node IDリストは、下図の32バイトのテーブルにおいて、存在するNode ID(16進表記)を示すビット位置に1をセットして1バイト目から順に列挙する。

	ビット0	ビット1	ビット2	ビット3	ビット4	ビット5	ビット6	ビット7
1バイト目	F8	F9	FA	FB	FC	FD	FE	FF
2バイト目	F0	F1	F2	F3	F4	F5	F6	F7
3バイト目	E8	E9	EA	EB	EC	ED	EE	EF
4バイト目	E0	E1	E2	E3	E4	E5	E6	E7
5バイト目	D8	D9	DA	DB	DC	DD	DE	DF
6バイト目	D0	D1	D2	D3	D4	D5	D6	D7
7バイト目	C8	C9	CA	CB	CC	CD	CE	CF
8バイト目	C0	C1	C2	C3	C4	C5	C6	C7
9バイト目	B8	B9	BA	BB	BC	BD	BE	BF
10バイト目	B0	B1	B2	B3	B4	B5	B6	B7
11バイト目	A8	A9	AA	AB	AC	AD	AE	AF
12バイト目	A0	A1	A2	A3	A4	A5	A6	A7
13バイト目	98	99	9A	9B	9C	9D	9E	9F
14バイト目	90	91	92	93	94	95	96	97
15バイト目	88	89	8A	8B	8C	8D	8E	8F
16バイト目	80	81	82	83	84	85	86	87
17バイト目	78	79	7A	7B	7C	7D	7E	7F
18バイト目	70	71	72	73	74	75	76	77
19バイト目	68	69	6A	6B	6C	6D	6E	6F
20バイト目	60	61	62	63	64	65	66	67
21バイト目	58	59	5A	5B	5C	5D	5E	5F
22バイト目	50	51	52	53	54	55	56	57
23バイト目	48	49	4A	4B	4C	4D	4E	4F
24バイト目	40	41	42	43	44	45	46	47
25バイト目	38	39	3A	3B	3C	3D	3E	3F
26バイト目	30	31	32	33	34	35	36	37
27バイト目	28	29	2A	2B	2C	2D	2E	2F
28バイト目	20	21	22	23	24	25	26	27
29バイト目	18	19	1A	1B	1C	1D	1E	1F
30バイト目	10	11	12	13	14	15	16	17
31バイト目	08	09	0A	0B	0C	0D	0E	0F
32バイト目	00	01	02	03	04	05	06	07

## (5) 戻り値

0 : 異常時

1 : 正常時

## (6) 使用する構造体

無し。

## (7) 注意事項・制限事項

無し。

## 4.2.3.4 ClcGetMasterRouterInfo

## (1) 名称

マスターータ情報取得関数

## (2) 機能

下位通信ソフトウェアが保持するマスターータ情報を取得する。

## (3) 構文

```

BOOL ClcGetMasterRouterInfo (
    unsigned char device_id, /*[IN] 下位通信ソフトウェア ID */
    unsigned char *result, /*[OUT] マスターータの存在の有無*/
    unsigned char *master_node_id /*[OUT] マスターータのNodeID */
)

```

## (4) 説明

**device\_id** : 下位通信ソフトウェア識別情報。

電灯線	0x11 ~ 0x1F
特定小電力無線	0x31 ~ 0x3F
拡張 HBS	0x41 ~ 0x4F
IrDA_Control	0x51 ~ 0x5F
LonTalk <sup>R</sup>	0x61 ~ 0x6F
Bluetooth <sup>TM</sup>	0x71 ~ 0x7F
Ethernet	0x81 ~ 0x8F

**\*result** : マスターータの存在の有無へのポインタを示す。

0x00 : 無し

0x01 : 有り

**\*master\_node\_id** : マスターータの NodeID の値へのポインタを示す。

## (5) 戻り値

0 : 異常時

1 : 正常時

## (6) 使用する構造体

無し。

## (7) 注意事項・制限事項

無し。

## 4.2.35 ClcLowReqToHardwareAddress

## (1) 名称

ハードウェアアドレス変換要求関数

## (2) 機能

下位通信ソフトウェアに対して渡した NodeID に対応するハードウェアアドレスを要求する。

## (3) 構文

BOOL ClcLowGetHardwareAddress (

unsigned char device\_id, /\*[IN] 下位通信ソフトウェア ID \*/

unsigned char node\_id, /\*[IN] 変換対象の NodeID \*/

unsigned char\* hardwareaddress, /\*[OUT] ハードウェアアドレスへのポインタ\*/

unsigned char\* hardwareaddress\_len /\*[OUT] ハードウェアアドレスのサイズへのポインタ\*/

)

## (4) 説明

device\_id : 下位通信ソフトウェア識別情報。

電灯線 0x11 ~ 0x1F

特定小電力無線 0x31 ~ 0x3F

拡張 HBS 0x41 ~ 0x4F

IrDA\_Control 0x51 ~ 0x5F

LonTalk<sup>R</sup> 0x61 ~ 0x6FBluetooth<sup>TM</sup> 0x71 ~ 0x7F

Ethernet 0x81 ~ 0x8F

node\_id : 変換対象の Node ID。

hardwareaddress : 変換後のハードウェアアドレスへのポインタが返る。

hardwareaddress\_len : 変換後のハードウェアアドレスのサイズのポインタが返る。

## (5) 戻り値

0 : 異常時

1 : 正常時

## (6) 使用する構造体

無し。

## (7) 注意事項・制限事項

無し。