

第3部 ECHONET Lite 通信装置仕様

改定履歴

- | | | |
|----------------------|-------------|-------------------|
| • Version 1.00 Draft | 2011年3月9日 | 制定, コンソーシアム会員内公開。 |
| • Version 1.00 | 2011年6月30日 | コンソーシアム会員内公開。 |
| | 2011年12月21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。
- 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 Lite通信装置仕様概要	1-1
1. 1 基本的考え方	1-1
1. 2 ECHONET Liteノードの通信装置仕様の概要	1-1
1. 3 ECHONET Liteゲートウェイの通信装置仕様の概要	1-1
1. 4 ECHONET Liteミドルウェアアダプタの通信装置仕様の概要	1-1
第2章 ECHONET Liteゲートウェイ	2-1
2. 1 基本的な考え方	2-1
第3章 ECHONET Liteミドルウェアアダプタ	3-1
3. 1 基本的な考え方	3-1
3. 1. 1 ECHONET Liteミドルウェアアダプタの想定構成 (解説)	3-4
3. 2 機能定義	3-7
3. 3 機械・物理特性	3-8
3. 3. 1 形状	3-8
3. 3. 2 表示部	3-9
3. 4 電気特性	3-9
3. 5 論理条件	3-9
3. 6 ECHONET Liteミドルウェアアダプタ通信ソフトウェア仕様	3-10
3. 6. 1 ECHONET Liteミドルウェアアダプタ通信インタフェースの概要	3-10
3. 6. 2 ECHONET Liteミドルウェアアダプタ通信インタフェース機械・物理特性	3-11
3. 6. 3 電気特性	3-20
3. 6. 4 論理条件	3-21
3. 6. 5 ECHONET Liteミドルウェアアダプタ通信ソフトウェアプロトコル	3-24
3. 7 機器インタフェース情報認識サービス	3-25
3. 7. 1 機器インタフェース情報認識サービス用フレーム構成	3-25
3. 7. 2 機器インタフェース情報認識サービス用コマンド	3-27
3. 7. 3 機器インタフェース情報認識サービスシーケンス	3-30
3. 7. 4 タイプ共通用状態遷移図	3-31
3. 7. 5 異常処理	3-33
3. 8 オブジェクト生成タイプ用通信プロトコル	3-34
3. 8. 1 オブジェクト生成タイプ用フレーム構成	3-35
3. 8. 2 アダプタ内部サービス	3-37
3. 8. 3 オブジェクト生成タイプ用状態遷移	3-42
3. 8. 4 オブジェクト生成タイプ用コマンド	3-45
3. 8. 5 オブジェクト生成タイプ用通信シーケンス	3-73
3. 8. 6 オブジェクト生成タイプの機械・物理特性	3-83
3. 9 Peer to Peerタイプ用通信プロトコル	3-84

3. 9. 1	プログラム選択形態.....	3-8 4
3. 9. 2	プログラムダウンロード形態.....	3-8 5
3. 9. 3	ECHONET Liteレディ機器からのプログラムダウンロードプロトコル.....	3-8 6
3. 10	Peer to Peer タイプのプログラムダウンロード形態における インタプリタ方式プログラム実行環境仕様 (推奨) ...	3-9 5
3. 10. 1	本推奨仕様の適用範囲.....	3-9 5
3. 10. 2	インタプリタ方式プログラム実行環境の概要.....	3-9 6
3. 10. 3	プログラム本体のフォーマット仕様.....	3-10 5
3. 10. 4	ダウンロードプログラム言語仕様.....	3-10 6
3. 10. 5	インタプリタ基本API仕様.....	3-10 8
3. 10. 6	インタプリタ ECHONET Lite API仕様.....	3-12 4
3. 10. 7	プログラム圧縮・伸張仕様.....	3-14 7
付録1	参考文献.....	i
付録2	インタプリタ方式サンプルプログラム.....	ii

第1章 ECHONET Lite通信装置仕様概要

1. 1 基本的考え方

第3部では、ECHONET Lite ノード、ECHONET Lite 機器アダプタ、ECHONET Lite ゲートウェイ、ECHONET Lite ミドルウェアアダプタの通信装置としての仕様について規定する。また、ECHONET Lite ミドルウェアアダプタと ECHONET Lite 機器のインタフェースの詳細や機能分担を合わせて規定する。

1. 2 ECHONET Liteノードの通信装置仕様の概要

ECHONET Lite ノードは、ECHONET Lite ネットワークを介して直接情報の交換が可能な通信端末の総称であり、1 通信端末を機能の区別をせずに示す際に用いるものである。ECHONET Lite ノードとしての要件を以下に示す。

- ・下位通信層
- ・ECHONET Lite 通信ミドルウェア

上記要件以外、ECHONET Lite ノードの通信装置としての仕様について特に規定しない。

1. 3 ECHONET Liteゲートウェイの通信装置仕様の概要

ECHONET Lite ドメインを外部のネットワークに接続する機能を有する ECHONET Lite ノード、言い換えると「サービスミドルウェアとしてゲートウェイ基本部の必須機能を持つ ECHONET Lite ノード」が ECHONET Lite ゲートウェイである。

ただし、ゲートウェイ機能を備える専用の ECHONET Lite ノードである必要はなく、他の機能を兼ねるものであってもよい。ECHONET Lite ゲートウェイを通信装置としてみた場合、ECHONET Lite ノードと変わるところはなく、したがって、ECHONET Lite ゲートウェイの通信装置としての仕様については特に規定しない。

ECHONET Liteゲートウェイに関しては第2章に述べる。

1. 4 ECHONET Liteミドルウェアアダプタの通信装置仕様の概要

ECHONET Lite ミドルウェアアダプタは、単独では ECHONET Lite ノードとなり得ない機器に、ECHONET Lite ノードとしての機能を付加するものである。

本規格においては、ECHONET Lite 通信処理部、下位通信層を持たない機器を ECHONET Lite ネットワークに接続するためのアダプタを ECHONET Lite ミドルウェアアダプタとして規定する。したがって、ECHONET Lite ミドルウェアアダプタは下記の要件を備える。

- ECHONET Lite 通信処理部
- 下位通信層
- ECHONET Lite ミドルウェアアダプタ通信ソフトウェア

ECHONET Liteミドルウェアアダプタ通信ソフトウェアは、ECHONET Liteミドルウェアアダプタと機器（ECHONET Liteレディ機器）の間の通信を行うもので、その仕様に関しては第3章にて述べる。

第2章 ECHONET Liteゲートウェイ

2. 1 基本的な考え方

ECHONET Lite プロトコルを使用して ECHONET Lite ドメイン内と、外部システムとを接続するアプリケーションソフトウェアがゲートウェイであり、これが搭載される機器がゲートウェイ装置である。しかしながら、ECHONET Lite では、現在のところ、特にアプリケーションがどのような処理をしなければならないかなどの規定については何ら定めていない。したがって、ECHONET Lite ドメインと外部システムとの接続は、アプリケーションソフトウェアの機能次第である。

ただし、通常の住居などにシステム導入する際には、ECHONET Lite ドメインのセキュリティを確保するため、ゲートウェイアプリケーションは、認証機能や、アクセス制御機能などのセキュリティ機能などを設けることを推奨する。なお、その際の機能的な定義については、第4部にて述べる。

第3章 ECHONET Lite ミドルウェアアダプタ

3. 1 基本的な考え方

本章では、下位通信層、ECHONET Lite 通信処理部を持たない機器を、ECHONET Lite ネットワークに接続するためのアダプタを ECHONET Lite ミドルウェアアダプタとして規定する。これを図3-1に示す。

図3-1に示すように、ECHONET Lite ミドルウェアアダプタを接続する機器を「ECHONET Lite レディ機器」と定義する。

ECHONET Lite ミドルウェアアダプタは、ECHONET Lite レディ機器側のネットワークに関する処理負担を最小限とし、ECHONET Lite レディ機器のコストアップ（ソフトウェア/ハードウェアのコストアップ）を抑えるように構成する。

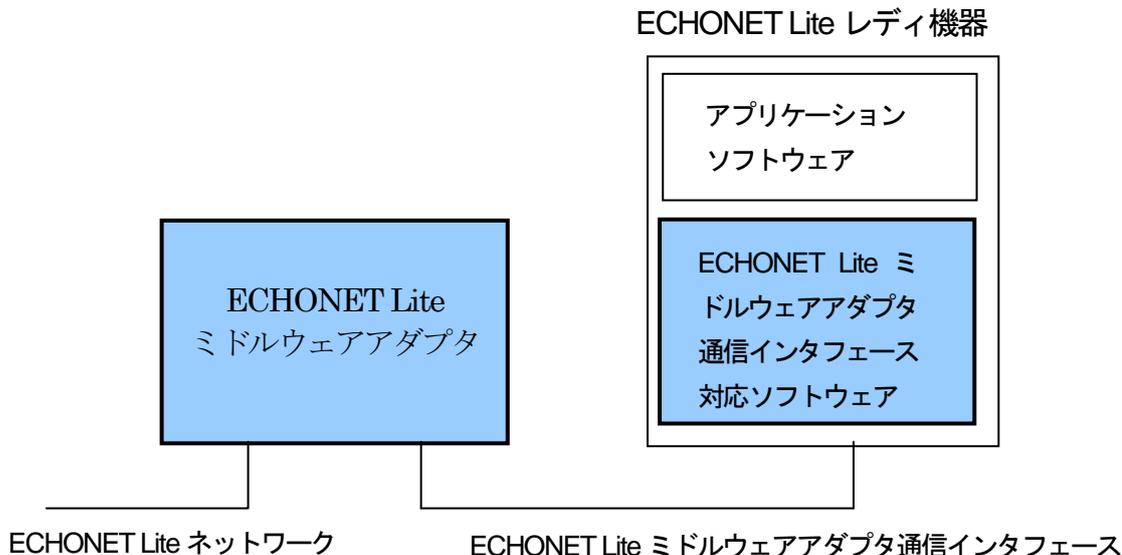


図3-1 ECHONET Lite ミドルウェアアダプタと ECHONET Lite レディ機器

ECHONET Lite ミドルウェアアダプタの装置仕様、ならびに電気、論理仕様は、3. 1～3. 5 節に規定する。ECHONET Lite ミドルウェアアダプタ通信インタフェースソフトウェア仕様の機械、物理、論理特性は、3. 6 節に規定する。

ECHONET Lite ミドルウェアアダプタ通信インタフェースソフトウェア仕様のプロトコル仕様は、ECHONET Lite レディ機器のネットワーク処理に関する負担を考慮し、本バージョンでは次のような複数のタイプの仕様を規定する。

- (1) オブジェクト生成タイプ
- (2) Peer to Peer タイプ

相互接続される ECHONET Lite ミドルウェアアダプタ、ECHONET Lite レディ機器の双方が、それぞれどのような仕様を実装しているかにより、両者の接続可否が発生する。よって、接続相手がどのようなタイプの仕様を実装しているかを識別するための「機器インタフェース情報認識サービス」を規定し、これを 3. 7 節に記述している。

機器 I/F 情報認識サービスによって識別後、ECHONET Lite ミドルウェアアダプタ通信インタフェースのタイプを選択する。

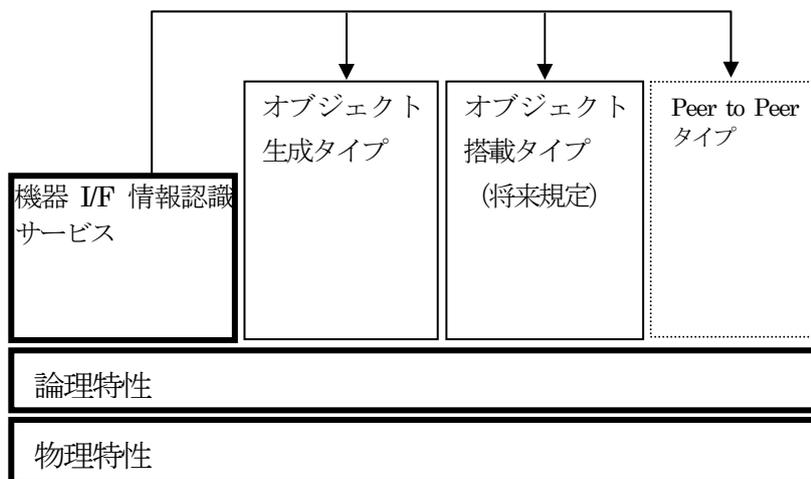


図3-2 ECHONET Lite ミドルウェアアダプタ通信ソフトウェア階層

ECHONET Lite ミドルウェアアダプタと、ECHONET Lite レディ機器がそれぞれ対応しているタイプ間の接続が可能な組み合わせ例は、下記のようになる。

表3-1 ECHONET Lite ミドルウェアアダプタと ECHONET Lite レディ機器のタイプ別接続可否

ECHONET Lite ミドルウェアアダプタ	ECHONET Lite レディ機器
オブジェクト生成タイプ	オブジェクト生成タイプ
Peer to Peer タイプ	Peer to Peer タイプ

オブジェクト搭載タイプで接続する場合は、ECHONET Lite ミドルウェアアダプタ内に、接続する ECHONET Lite レディ機器に対応した基本的なオブジェクト構築情報が準備されていなければならない。また、Peer to Peer タイプで接続する場合は、ECHONET Lite ミドルウェアアダプタ内に、接続する ECHONET Lite レディ機器に対応した ECHONET Lite ミドルウェアアダプタ通信ソフトウェアが準備されていなければならない。

3. 1. 1 ECHONET Liteミドルウェアアダプタの想定構成（解説）

想定される ECHONET Lite ミドルウェアアダプタの実装方法に対応するため、複数の ECHONET Lite ミドルウェアアダプタ通信ソフトウェアプロトコルを規定している。想定している ECHONET Lite ミドルウェアアダプタの構成を図3-3に示す。

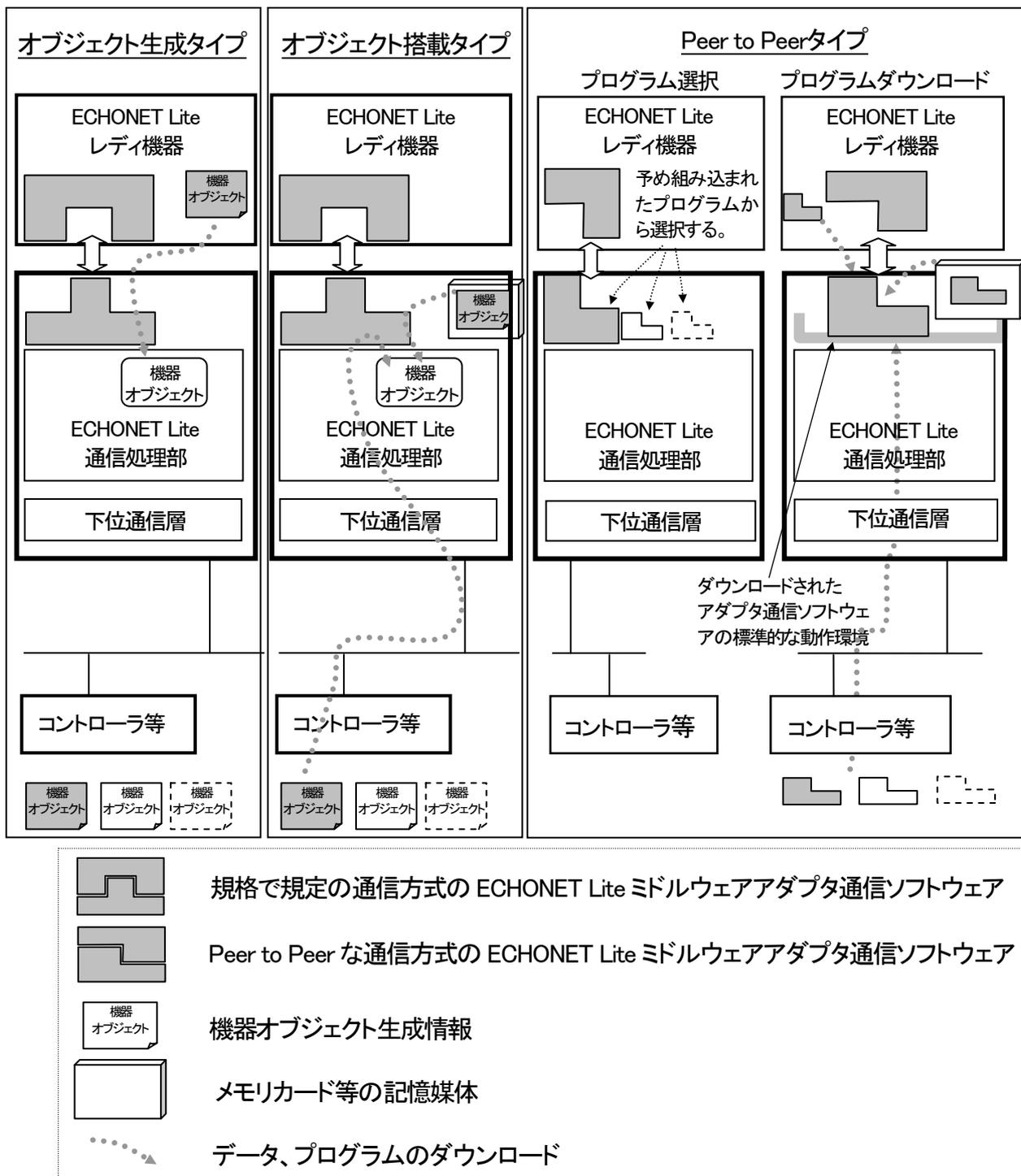


図3-3 ECHONET Lite ミドルウェアアダプタの想定構成

図3-3に示すように、ECHONET Lite ミドルウェアアダプタの実装方法として、次のような3種類が想定される。

(1)オブジェクト生成タイプ

ECHONET Lite ミドルウェアアダプタと ECHONET Lite レディ機器の間で、規格化された標準的な通信方式によって情報授受を行うタイプであり、ECHONET Lite レディ機器内にあらかじめ準備された少なくとも一つ以上のオブジェクト生成情報を規格化された手順により ECHONET Lite ミドルウェアアダプタへ設定するタイプ。

(2)オブジェクト搭載タイプ

ECHONET Lite ミドルウェアアダプタと ECHONET Lite レディ機器の間で、規格化された標準的な通信方式によって情報授受を行うタイプであり、オブジェクト生成情報は ECHONET Lite ミドルウェアアダプタ内に予め保持しているか又は外部からダウンロードすることによりオブジェクトが生成されることが想定される。オブジェクトの取得方法についてはここでは規定しない。

(3)Peer to Peer タイプ

ECHONET Lite ミドルウェアアダプタと ECHONET Lite レディ機器の間で、ユーザ定義の通信方式によって、情報授受を行うタイプ。接続された ECHONET Lite レディ機器に対応した通信方式 (ECHONET Lite ミドルウェアアダプタ通信ソフトウェア) を、あらかじめ ECHONET Lite ミドルウェアアダプタ内に備えている「プログラム選択」の形態と、外部からダウンロードして入手する「プログラムダウンロード」形態が想定される。

本規格では、家電機器の対応の経過からこれら複数の方法を想定し、ECHONET Lite ミドルウェアアダプタと ECHONET Lite レディ機器が正当な組み合わせで接続された場合には、これを識別し、接続、動作するようにするため、下記を規格化する。

(1)オブジェクト生成タイプを実現する、標準的な通信方式

オブジェクト生成タイプを実現するための標準的な通信方式を規定し、また、ECHONET Lite レディ機器からオブジェクト生成情報を取得し、ECHONET Lite ミドルウェアアダプタ内にオブジェクトを生成するための方法を規定する。

オブジェクト生成情報をダウンロードする場合については、本バージョンでは規定しない。

(2)Peer to Peer 方式プログラムダウンロード形態を実現する通信方式

ECHONET Lite レディ機器から、ECHONET Lite ミドルウェアアダプタ通信ソフトウェアをダウンロードする通信方式を規定する。ECHONET Lite レディ機器以外からダウンロードする通信方式は、本バージョンでは規定しない。

(3)ECHONET Lite レディ機器の通信方式を識別する方法 「機器インタフェース情報認

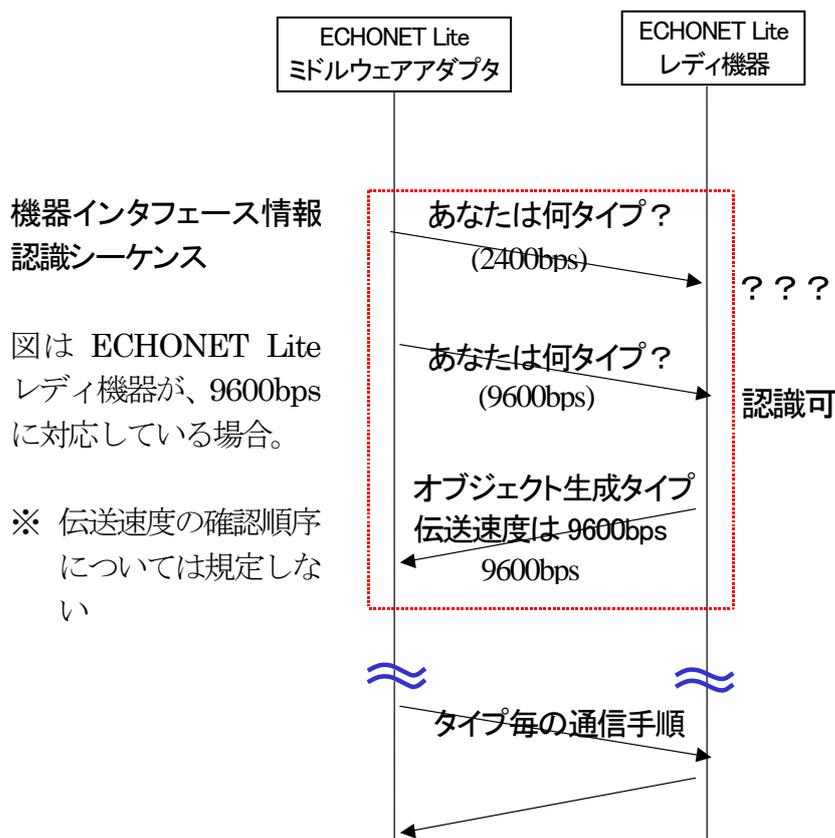
識サービス」

ECHONET Lite レディ機器の機器インタフェース情報を取得し、通信方式（オブジェクト生成方式、あるいはPeer to Peer 方式、あるいはその他の新通信方式）や、機器の情報を識別するためのサービス。

本サービスに従い、ECHONET Lite ミドルウェアアダプタから ECHONET Lite レディ機器に機器インタフェース情報の問い合わせを行う。問合せ後、ECHONET Lite ミドルウェアアダプタは取得した情報にもとづく ECHONET Lite ミドルウェアアダプタ通信ソフトウェアを実行する。

(4)上記共通の、通信インタフェースの論理、物理特性。

また、機器インタフェース情報認識サービスの実施イメージを図3-4に記載する。詳細は図3-1 3参照のこと。



3. 2 機能定義

ECHONET Lite ミドルウェアアダプタとして、必要な機能を以下のように規定する。

(1) 伝送メディアとの入出力機能

伝送メディアとの間で電文を入出力するための機能。本機能は、下位通信層が実行する。すなわち、下位通信層のプロトコルを扱うことができる1つのトランシーバが機能として必要となる。

(2) ECHONET Lite 通信処理機能

第2部の「第5章 ECHONET Lite 通信処理部処理仕様」に規定されている処理を行うための機能。本機能は、ECHONET Lite 通信処理部が実行する。

(3) ECHONET Lite ミドルウェアアダプタ通信インタフェース機能

本章の「3. 6 ECHONET Lite ミドルウェアアダプタ通信ソフトウェア仕様」に規定される機能であり、ECHONET Lite レディ機器アプリケーションソフトウェアと、ECHONET Lite 通信処理部間で、ECHONET Lite ミドルウェアアダプタ通信インタフェース仕様に基づくECHONET Lite ミドルウェアアダプタ、ECHONET Lite レディ機器間の通信処理を行う。本機能は、ECHONET Lite ミドルウェアアダプタ通信ソフトウェアが実行する。

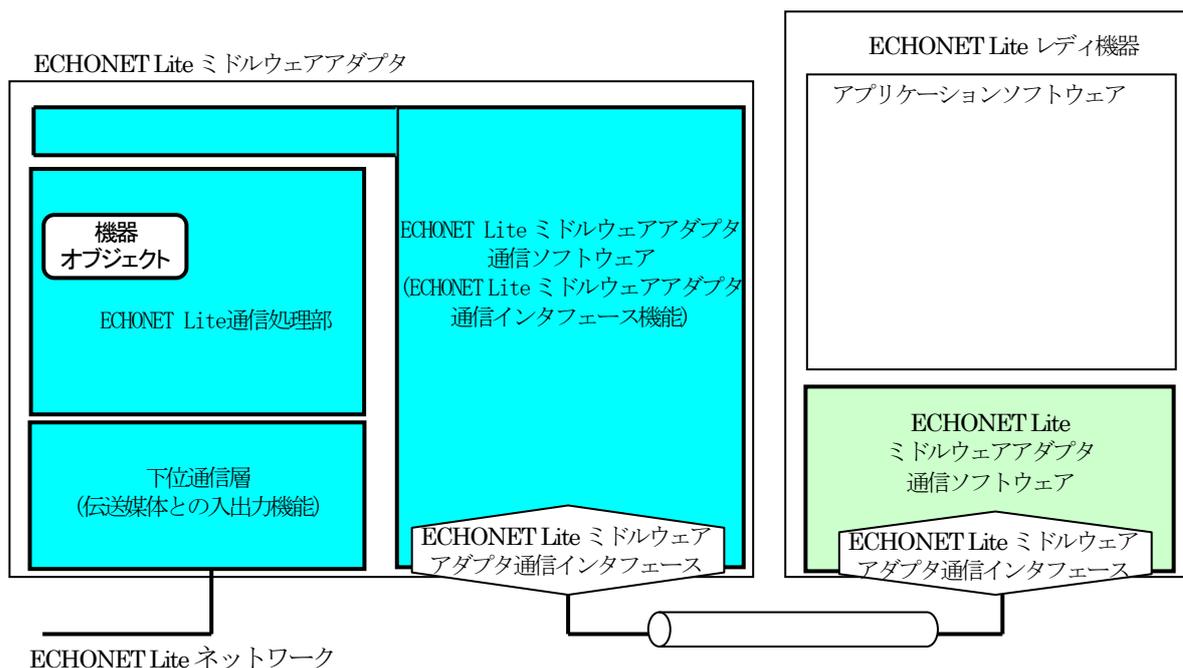


図3-5 ECHONET Lite ミドルウェアアダプタの機能

3. 3 機械・物理特性

伝送メディアとの接続仕様に関しては、個々の下位通信層に従う。また、ECHONET Lite レディ機器との接続仕様に関しては、第6節において規定する。ここでは、それ以外のECHONET Lite ミドルウェアアダプタとしての機械・物理特性に関して規定する。

3. 3. 1 形状

形状に関しては、ECHONET Lite ミドルウェアアダプタ通信インタフェースによるECHONET Lite レディ機器との接続部以外、特に規定は設けない。伝送メディアとの接続部の形状に関しては、使用する各下位通信層における規定に従うものとする。

3. 3. 2 表示部

ECHONET Lite ミドルウェアアダプタの動作状態を表示するためにLEDを具備する場合は、少なくとも下記仕様を満足することを推奨する。なお、ここで規定されていない手段による表示方法に関しては、個々の製品の独自規定とする。

- | | |
|-----------|---------------------|
| ① LED の数 | : 1 個 (運転状態提示用) |
| ② LED の色 | : 緑 |
| ③ 状態の表示方法 | 通常動作時 : 点灯 |
| | イニシャル処理時 : 点滅 (長周期) |
| | 異常時 : 点滅 (短周期) |
| | 非動作時 : 消灯 |
- ※長周期・・・約 2sec 点灯、約 0.5sec 消灯の繰り返し
※短周期・・・約 0.5sec 点灯、約 0.5sec 消灯の繰り返し

注) イニシャル処理とは、コールドスタート (完全リセットスタート)、およびウォームスタート (既に獲得したアドレスや、初期設定情報は保持したまま、ハードウェア的なリセット処理を行うスタート) のことである。

3. 4 電気特性

伝送メディアとの接続仕様に関しては、ECHONET Lite ミドルウェアアダプタが対応する個々の下位通信層の規定されたものに従う。また、ECHONET Lite レディ機器との接続仕様に関しては、第6節に規定する。

3. 5 論理条件

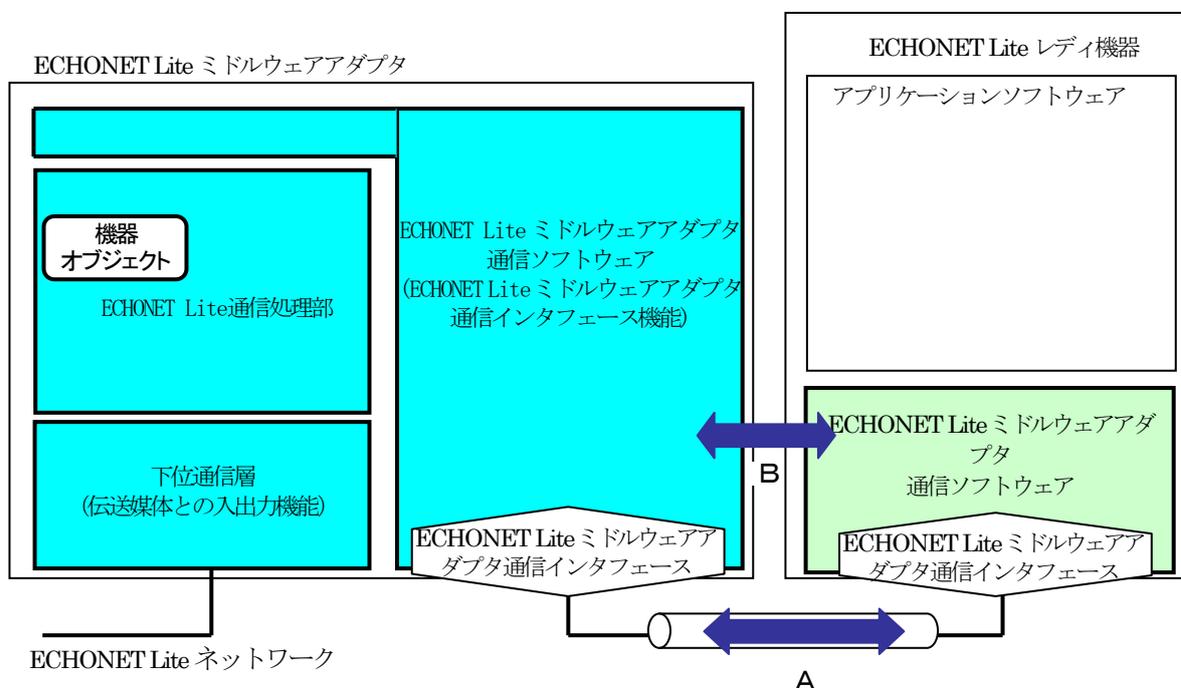
ECHONET Lite ミドルウェアアダプタ通信ソフトウェアの論理条件については、第6節において規定する。下位通信層に関する論理条件については、伝送に使用する下位通信層の仕様に基づくものとする。

3. 6 ECHONET Liteミドルウェアアダプタ通信ソフトウェア仕様

ECHONET Lite ミドルウェアアダプタ通信ソフトウェアとは、ECHONET Lite ミドルウェアアダプタ、及びECHONET Lite レディ機器上で動作するものであり、ECHONET Lite ミドルウェアアダプタ通信ソフトウェアプロトコルを扱う。ここでは、ECHONET Lite ミドルウェアアダプタ通信インタフェース、ECHONET Lite ミドルウェアアダプタ通信ソフトウェアプロトコル、及びその扱いについて規定する。

3. 6. 1 ECHONET Liteミドルウェアアダプタ通信インタフェースの概要

ECHONET Lite ミドルウェアアダプタ通信ソフトウェアによる ECHONET Lite ミドルウェアアダプタと機器との間のやり取りを、図3-6に示す。



- A : ECHONET Lite ミドルウェアアダプタ通信インタフェース機械・物理・電気特性
- B : ECHONET Lite ミドルウェアアダプタ通信ソフトウェアプロトコル

図3-6 ECHONET Lite ミドルウェアアダプタと機器のやりとり

3. 6. 2 ECHONET Liteミドルウェアアダプタ通信インタフェース機械・物理

特性

ECHONET Lite ミドルウェアアダプタ通信インタフェースの、機械・物理特性について規定する。

(1) 伝送メディア

ECHONET Lite ミドルウェアアダプタ通信インタフェースで使用する伝送メディアは、以下のように推奨する。

多芯ケーブル 8本 心線径は規定せず

(2) ケーブル長

オープンコレクタの場合、保証の範囲は最大2mまでとする。

(3) 接続形態

1つの ECHONET Lite ミドルウェアアダプタと対応する一つの ECHONET Lite レディ機器の、1対1接続形態とする。

(4) コネクタ形状

ECHONET Lite レディ機器側のコネクタはPHコネクタ8ピンを推奨とする。

ECHONET Lite ミドルウェアアダプタ側のコネクタについては特に規定しない。

特に、コンシューマが、ECHONET Lite ミドルウェアアダプタを取り付け、交換などを行うことを前提としてコネクタを実装する場合は、以下の検討項目に対してクリアにする必要がある。

- 誤挿入防止
- 活線挿抜対応
- 給電クラス

そこで、給電クラス1用に、これらの検討項目をクリアした推奨ミドルウェアアダプタ9Pコネクタ(MA9/MA9Bコネクタ)を、ECHONET Lite レディ機器側(MA9/MA9Bソケット)とECHONET Lite ミドルウェアアダプタ側(MA9/MA9Bプラグ)のコネクタとして推奨する。

- ※ 給電クラス1と給電クラス2, 3の異なる機器で使用すると、故障などの問題が発生するおそれがあるため、MA9/MA9Bコネクタは給電クラス1の機器のみに使用する。
- ※ 給電クラス1の範囲内であっても給電容量による違いで動作しないことを防ぐため、給電容量2000mVA未満をMA9コネクタ、給電容量2000mVA以上をMA9Bコネクタとする。

MA9/MA9B コネクタを採用した場合は、多芯ケーブルの本数は問わないものとする。
 メーカーがコネクタを採用する場合においては、厳密な相互互換を目指すためにこの
 MA9/MA9B コネクタの仕様に沿うことが望ましい。

MA9/MA9B コネクタ仕様

●物理仕様

項目	内容	備考
極数	9 極	ECHONET Lite ミドルウェアアダプタ規格推奨の 8 極に、将来規定として家電機器 WakeUP 機能を追加。(WakeUP 動作内容は要検討事項)
コネクタ端子間ピッチ	2mm	家電機器、アダプタの電力供給と信号線を考慮した設定
定格電流	0.5ADC	ECHONET Lite における家電機器の電力供給
定格電圧	15V DC	ECHONET Lite における家電機器の電力供給
使用温度範囲	-20°C~+85°C	屋外での使用も想定。 定格電圧・電流で連続して使用可能な周囲温度の範囲。
保存温度範囲	-40°C~+85°C	屋外での使用も想定。 無負荷の状態での保存可能な周囲温度の範囲。
耐熱性	<ul style="list-style-type: none"> ・接触抵抗 初期規格値の 2 倍以下 ・絶縁抵抗 初期規格値を満足すること ・耐電圧 初期規格値を満足すること ・外観 ひび割れ、変形などの異常がないこと 	85±2°C、500 時間放置後、常温常湿中に 0.5 時間放置し測定。
耐寒性	<ul style="list-style-type: none"> ・接触抵抗 初期規格値の 2 倍以下 ・絶縁抵抗 初期規格値を満足すること ・耐電圧 初期規格値を満足すること ・外観 ひび割れ、変形などの異常がないこと 	-40±2°C、500 時間放置後、常温常湿中に 0.5 時間放置し測定。
耐湿性	<ul style="list-style-type: none"> ・接触抵抗 初期規格値の 2 倍以下 ・絶縁抵抗 初期規格値を満足すること ・耐電圧 初期規格値を満足すること ・外観 ひび割れ、変形などの異常がないこと 	60±2°C、相対湿度 90~95%、500 時間放置後、常温常湿中に 0.5 時間放置し測定。

接触抵抗	10mΩ or 以下	適合コネクタを嵌合し、各端子間を測定する。 (但し、導体抵抗は除く) 測定周波数 1,000Hz、測定電流 100mA or 以下
絶縁抵抗	1000MΩ or 以上	導通相互間に DC500V 1分間印加後 測定
耐電圧	アーク、絶縁破壊などの異常がないこと	導通相互間に AC500V 1分間印加し 確認 遮断電流 2mA
絶縁	絶縁距離 2.5mm 以上 図3-7(b) *2部分を参照	MA9プラグの端子(充電部)とコネクタ開口部(外郭)との絶縁距離 2.5mm 以上を確保する。(補助絶縁対応)
材料	RoHS 指令適合 ハウジング UL94 V-0 以上 PBT ガラス入り コンタクト (導電部) 銅合金	
形状仕様	給電クラス1 の2000mVA未満 の用途に使用 MA9 ソケット (ECHONET Lite レディ機器側) MA9 プラグ (ECHONET Lite ミドルウェアアダプタ側) 嵌合部分 給電クラス1 の2000mVA以上 の用途に使用 MA9B ソケット (ECHONET Lite レディ機器側) MA9B プラグ (ECHONET Lite ミドルウェアアダプタ側) 嵌合部分	図3-7(a) の形状であること 図3-7(b) の形状であること 図3-7(c) の形状であること 図3-7(d) の形状であること 図3-7(e) の形状であること 図3-7(f) の形状であること
誤挿入防止機構	図3-7(a) *1 部分を参照	ユーザーによる誤挿入を想定し、防止する機構を設ける。
活線挿抜機構	図3-7(b) *1 部分を参照	挿抜時に1番ピンが最初に接触し、最後まで接触を保つタイミング機能により、機器とアダプタの安全性を確保
挿抜回数	500 回	
ロック機構	有り (ハーフロック) 図3-7(c) *1 部分を参照	アンロック強度 20~40N 程度とし、確実に挿入されたことが認識できること。
防水機構	Oリング用溝 図3-7(c) *2 部分を参照	防水、防沫のコネクタを作成できるように、Oリングを装着するための溝を設ける。

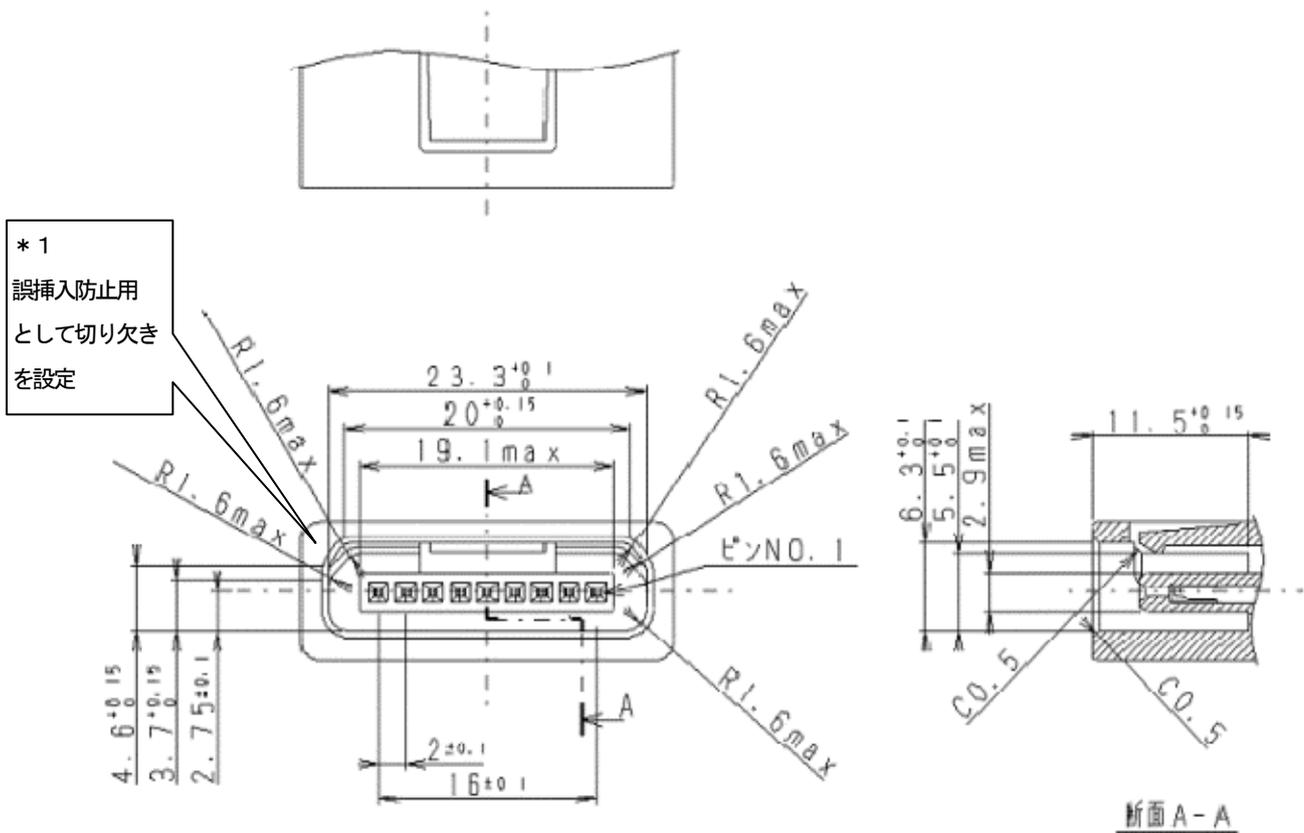


図3-7 (a) MA9 ソケット (ECHONET Lite レディ機器側)

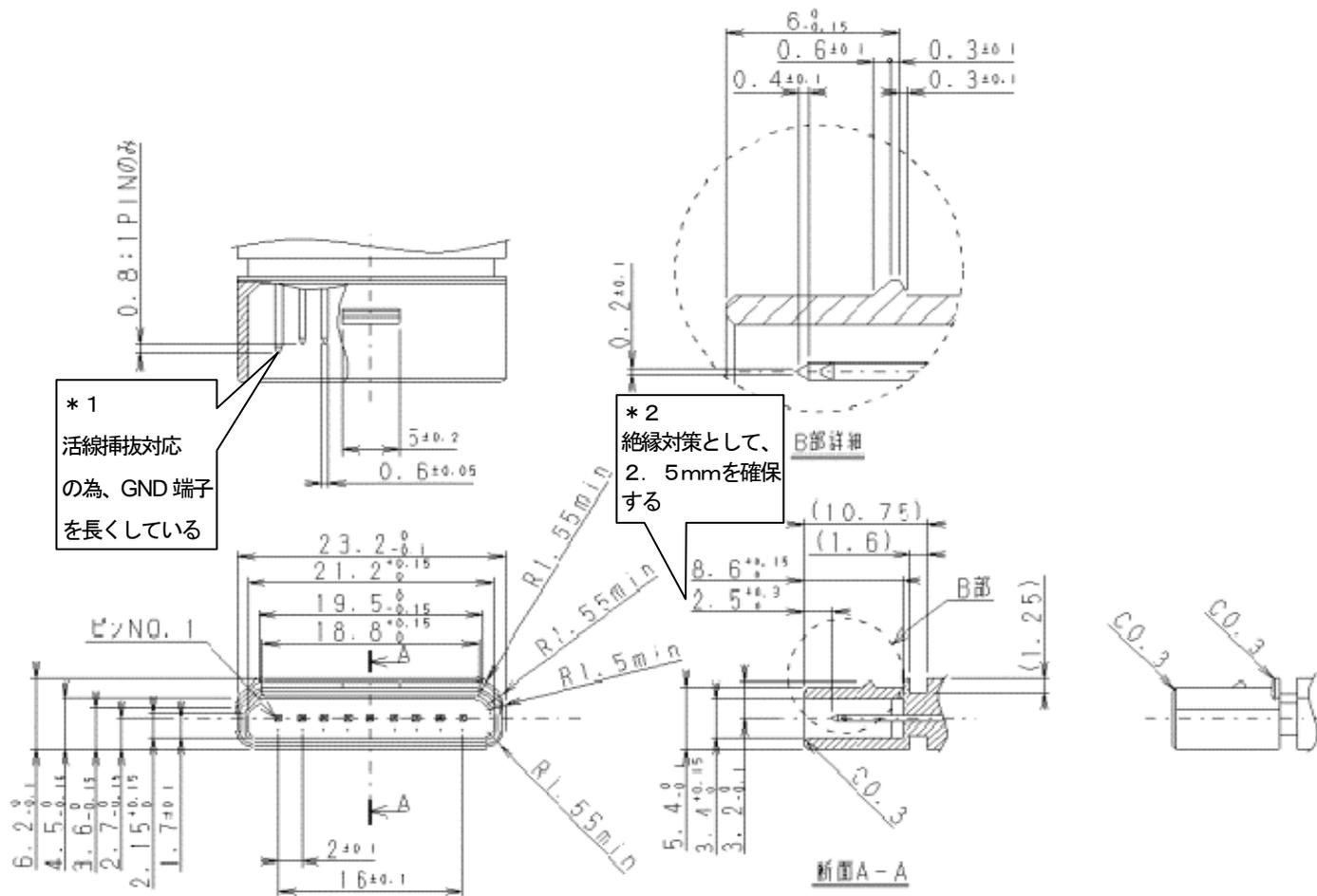


図3-7 (b) MA9プラグ (ECHONET Lite ミドルウェアアダプタ側)

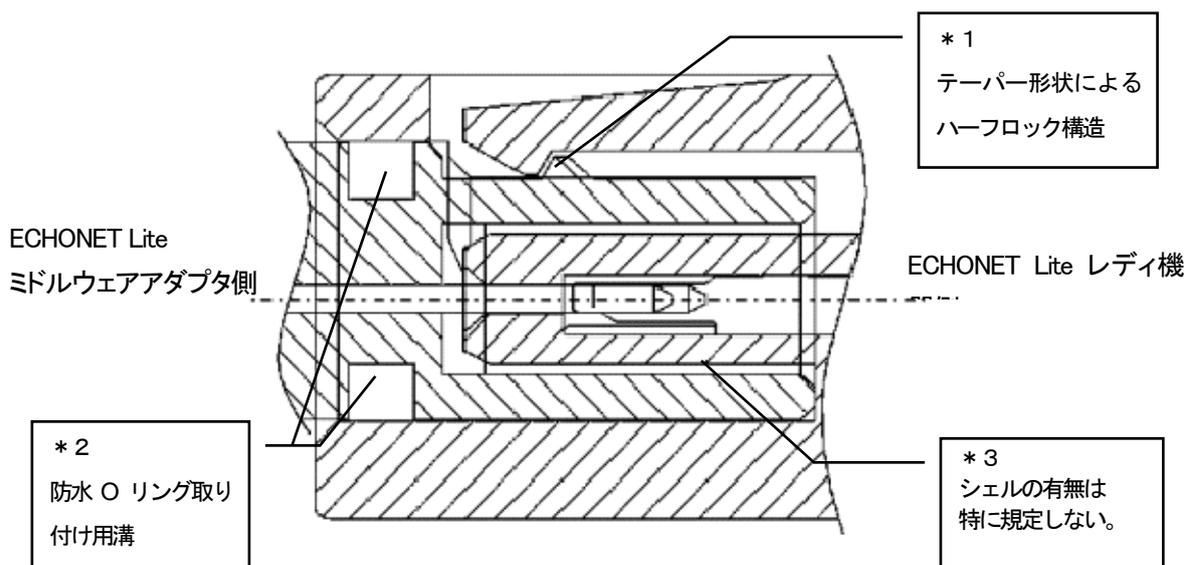
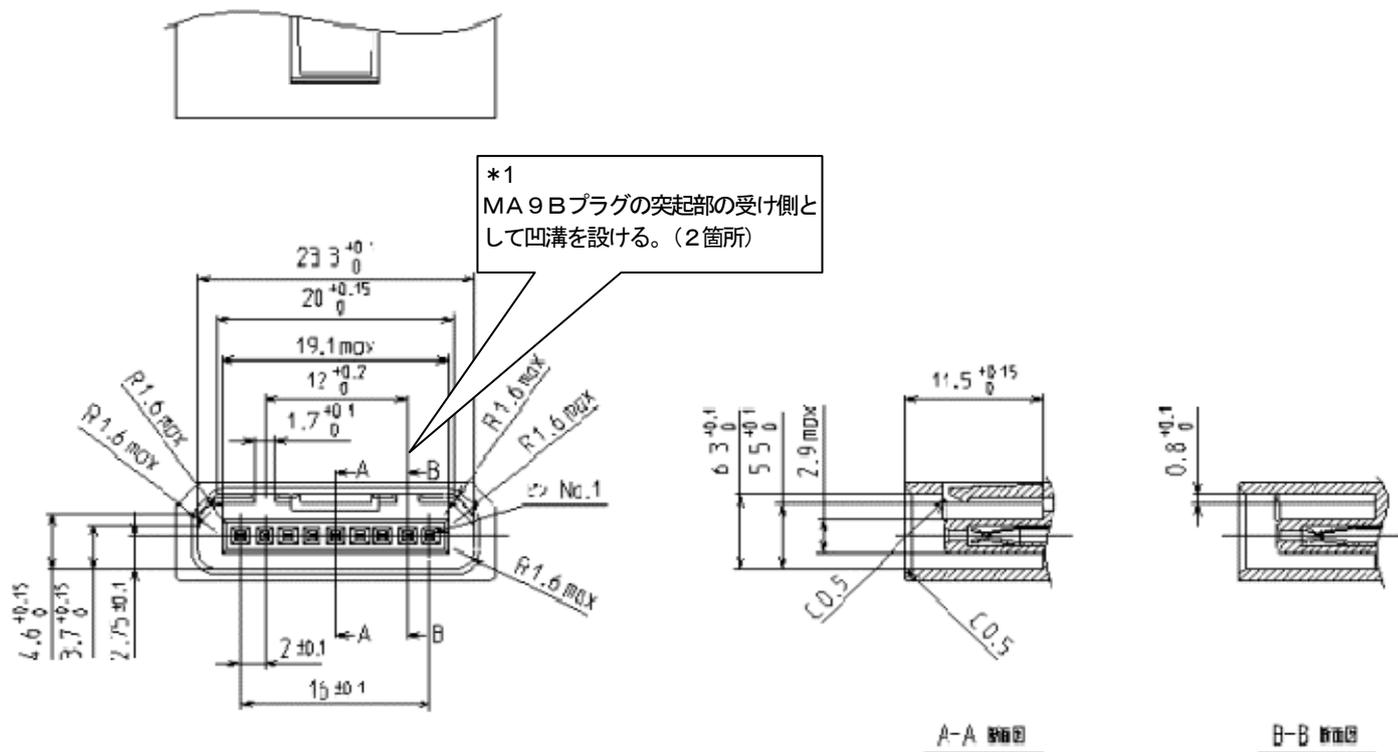


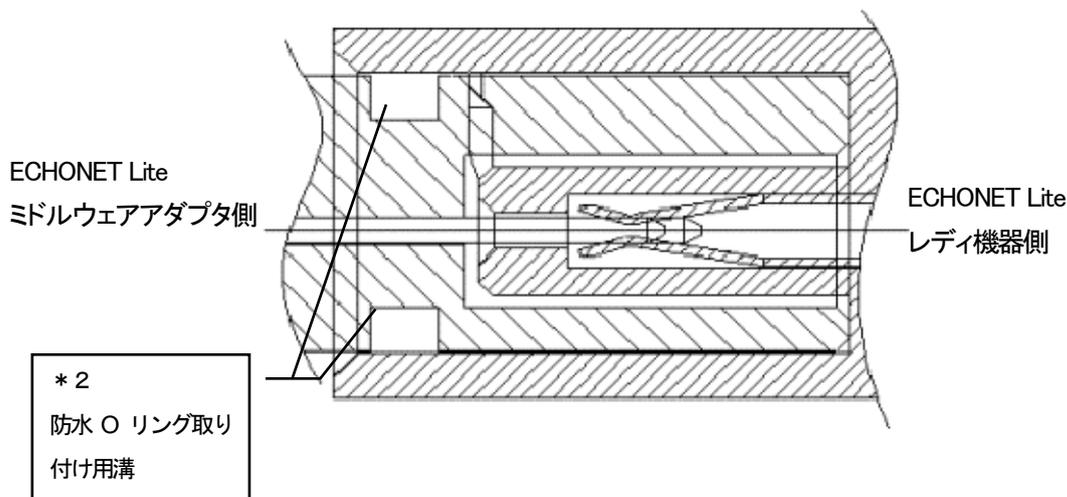
図3-7 (c) MA9コネクタ 嵌合図



(注)

MA9Bソケットは、断面B-Bの凹溝(2箇所)以外MA9ソケットと同一形状としている。従って、MA9 Bプラグだけでなく、MA9プラグと接続可能。

図3-7 (d) MA9B ソケット (ECHONET Lite レディ機器側)



(注)
 断面A-Aの嵌合図は、図3-7 (c) と同一のため記載しない。

図3-7 (f) MA9Bコネクタ 嵌合図 (断面B-B)

(5) コネクタと信号の対応

図3-8にPHコネクタのピン配列を示す (推奨)。また、MA9/MA9Bコネクタ使用時のピン配列も合わせて示す (推奨)。

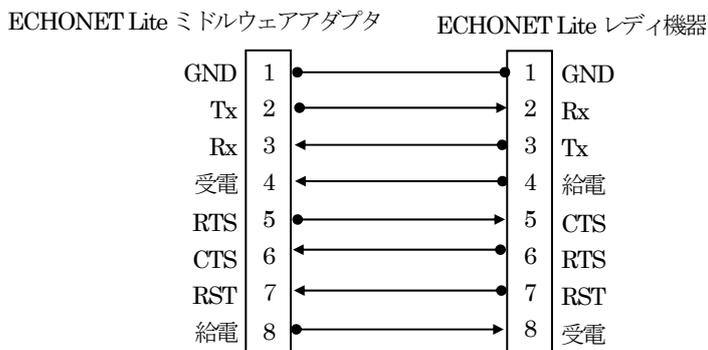


図3-8 (a) ECHONET Lite ミドルウェアアダプタ通信インタフェース ピン配置 (推奨)

●MA9/MA9Bコネクタ使用時

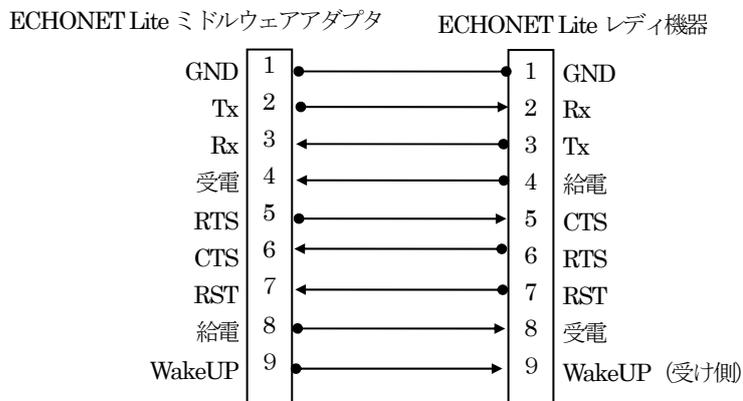


図3-8 (b) ECHONET Lite ミドルウェアアダプタ通信インタフェース ピン配置2 (推奨)

○受給電 3.6.3 (5)給電 参照
 ECHONET Lite レディ機器→ECHONET Lite ミドルウェアアダプタ
 ECHONET Lite レディ機器:オプション
 ECHONET Lite ミドルウェアアダプタ:オプション
 ECHONET Lite ミドルウェアアダプタ→ECHONET Lite レディ機器
 ECHONET Lite レディ機器:オプション
 ECHONET Lite ミドルウェアアダプタ:オプション

○RST(リセット)
 ECHONET Lite レディ機器から ECHONET Lite ミドルウェアアダプタへのリセット出力
 Low でアダプタのマイコン動作停止状態、Low→High でリセット起動
 ECHONET Lite レディ機器:オプション
 ECHONET Lite ミドルウェアアダプタ:必須

○RTS/CTS 3.6.4 (1)制御方式 参照
 ECHONET Lite レディ機器側:オプション
 ECHONET Lite ミドルウェアアダプタ側:必須

3. 6. 3 電気特性

ECHONET Lite ミドルウェアアダプタ通信インタフェースの、電気特性について規定する。

(1) ケーブルの特性インピーダンス
特に規定しない。

(2) 信号の伝送速度

ECHONET Lite ミドルウェアアダプタ通信インタフェースの機器インタフェース情報認識サービスで使用する信号の伝送速度としては、以下の 2 種類の速度を実装する。ECHONET Lite レディ機器は、下記2種類の速度のいずれか一方を搭載するものとする。

伝送速度 2400 b p s ± 2% / 9600 b p s ± 2%

(3) 信号の伝送方式、及び伝送波形

コネクタ端子をインタフェースのポイントとし、信号の伝送方式、及び伝送波形を、以下のように規定する。

- ① 伝送方式：ベースバンド伝送
- ② 伝送波形：単流NRZ方式
- ③ 論理

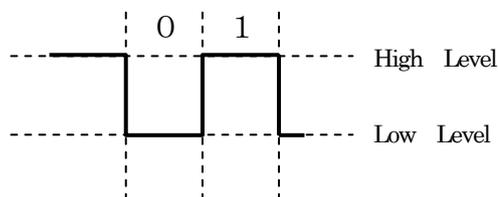


図3-9 論理レベル

(4) 出力端子仕様

タイプ1：オープンコレクタ

タイプ2：3.3V CMOS

(5) 給電仕様

表3-2 ECHONET Lite レディ機器給電仕様 (クラス1)

給電電圧	4.5V~15.0V
給電容量	1200mVA 以上

表3-3 ECHONET Lite レディ機器給電仕様 (クラス2)

給電電圧	4.5~5.5V
給電容量	300mA 以上

表3-4 ECHONET Lite レディ機器給電仕様 (クラス3)

給電電圧	3.0~4.5V
給電容量	300mA 以上

表3.5 ECHONET Lite ミドルウェアアダプタ給電仕様

給電電圧	3.0~5.5V
給電容量	100mA 以上

(6) リセット

ECHONET Lite レディ機器から ECHONET Lite ミドルウェアアダプタを電氣的にリセットするためのリセット端子 (RST) を規定する。ECHONET Lite レディ機器での実装はオプションとするが、ECHONET Lite ミドルウェアアダプタでは実装必須とする。RST端子がHigh 状態で通常動作とし、Low状態でECHONET Lite ミドルウェアアダプタ動作停止とし、Low → High でリセット起動とする。

3.6.4 論理条件

ECHONET Lite ミドルウェアアダプタ通信インタフェースの論理条件について規定する。

(1) 制御方式

RTS/CTSによる制御方式とする。RTSは、相手側への送信開始の通知、及び受信不可の状態を通知する為の信号とし、CTSは、相手側の受信可否状態の通知、及び相手からの通信開始の通知用の信号とする。電文送信時のRTS/CTSの制御手順は、以下とする。

- ①受信できない場合はRTSを「High Level」にし、受信可能な場合はRTSを「Low Level」にする。
- ②送信する前にCTSが「Low Level」であることを確認する。
(CTSが「High Level」の場合は送信しない)
- ③TXDにデータを出力する。

先にサービス要求を送信開始した方を優先するが、サービス要求の衝突が起きた場合は、ECHONET Lite ミドルウェアアダプタからのサービス要求の送信を優先する。また、フレーム送信中に、CTSの状態が変わっても、送信側は、当該フレームの送信は中断しなくてもよい。中断した場合には、当該フレームは無効として、その後の規定処理を行う。

(2) 同期方式

同期方式はキャラクタ毎の調歩同期とし、以下のように規定する。

- ① キャラクタ構成 (図3-10参照) 合計11ビット
 - スタートビット (ST) : 1ビット
 - データ : 8ビット
 - パリティ : 1ビット
 - ストップビット (STP) : 1ビット
- ② データ送信順序 : LSBファースト
- ③ スタートビット : 論理0
- ④ ストップビット : 論理1
- ⑤ パリティ : 偶数パリティ

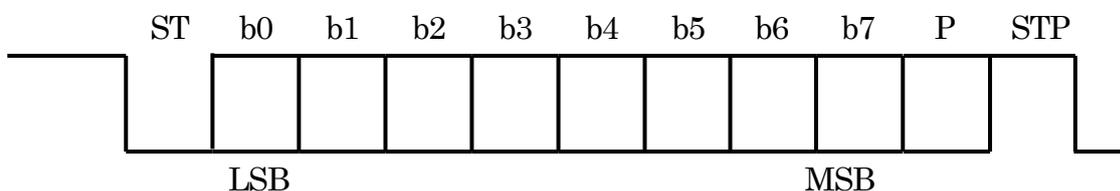


図3-10 キャラクタ構成

(3) 時間規定

機器I/F情報認識サービス実行時の、時間規定を図3-11、表3-6に示す。アダプタが要求フレームを送信し、ECHONET Lite レディ機器がこれを受信し応答フレームを送信する。

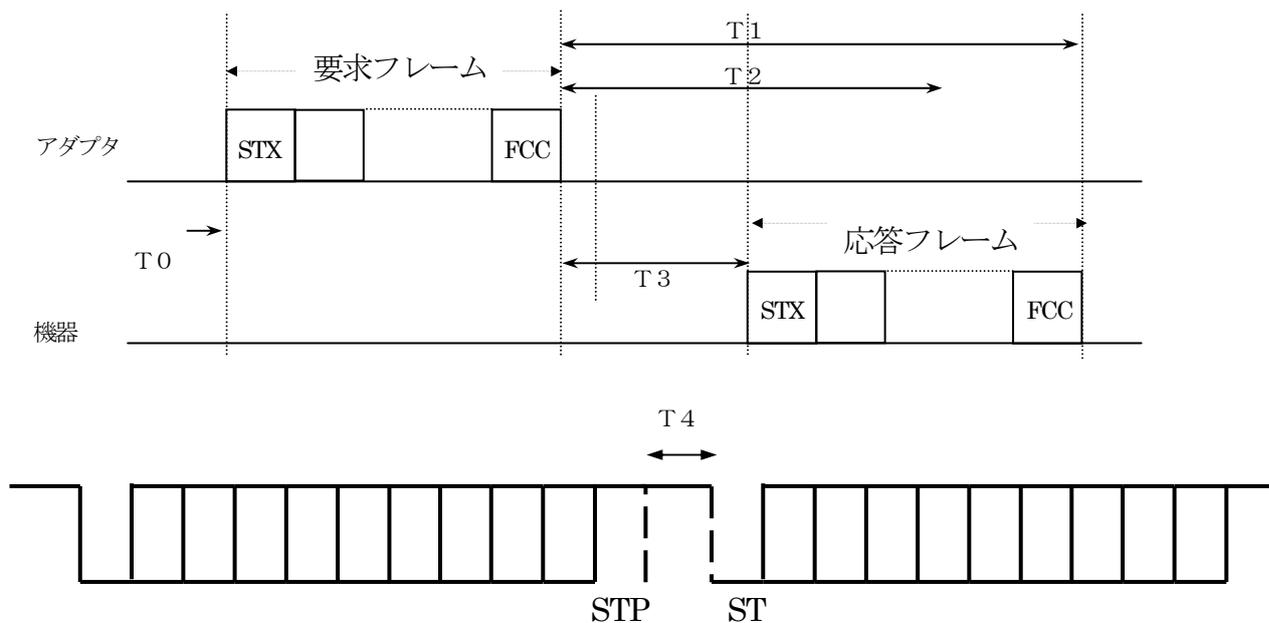


図3-11 時間規定

表3-6 時間規定

記号	該当側	時間名称	内容
T0	ECHONET Lite ミドルウェアアダプタ/ECHONET Lite レディ機器	受信側フレーム同期確認 (フレーム終了、先頭判定)	受信なしの状態を 9600bps 以下の場合は 10msec 以上 9600bps を超える場合は3キャラクタ構成成長以上
T1	ECHONET Lite ミドルウェアアダプタ	応答待ちタイムアウト	自送信フレームの終わりから 300msec
T2	ECHONET Lite ミドルウェアアダプタ	再送信禁止時間	自送信フレームの終わりから 300msec
T3	ECHONET Lite レディ機器	応答送信禁止時間 (T4 確認時間を意味する)	9600bps 以下の場合は 10msec 以上 9600bps を超える場合は3キャラクタ構成成長以上
T4	ECHONET Lite ミドルウェアアダプタ/ECHONET Lite レディ機器	送信側キャラクタ構成間隔	9600bps 以下の場合は 10msec 未満 9600bps を超える場合は3キャラクタ構成成長未満

(4) フィールド構成

ECHONET Lite ミドルウェアアダプタ通信インタフェースのフレームは、制御コード(STX)を先頭に、チェックコード(FCC)を最後部に配置し、これらの間にデータを挿入する。挿入するデータは、「3. 6. 5 ECHONET Lite ミドルウェアアダプタ通信ソフトウェアプロトコル」に示される各プロトコルにより異なる。

(5) 制御コード(STX)

先頭コード。0x02 固定とする。

(6) データ部(DATA)

データ部は、ECHONET Lite ミドルウェアアダプタ通信インタフェースのSTXの次からFCCの前までである。ECHONET Lite ミドルウェアアダプタ通信インタフェースのタイプにより、具体的な内容は異なる。(詳細は、「3. 6. 5 ECHONET Lite ミドルウェアアダプタ通信ソフトウェアプロトコル」参照。)

(7) チェックコード(FCC)

チェックコードは、フレーム伝送エラー検出のためのものであり、データ部のキャラクタの値を合計した値の2の補数とする。

(8) フレームの終了検出

ストップビット検出後、3フィールド長未満(ただし9600bps以下の場合は10msec)経過しても次のキャラクタのスタートビットが検出されない場合、フレームは終了したものと判断する。

(9) 誤り検出

受信端末は以下の誤り検出を行う

○バイト受信エラー

各バイトに、パリティビットを設ける。パリティは、偶数パリティ。

フレーム受信時、任意の1バイト受信においてパリティエラーを検出した場合、受信端末は受信中のフレームを無効とし、破棄する。

フレームの破棄は、フレーム受信完了またはフレーム受信タイムアウト時点とする。

○FCC エラー

1 フレーム受信完了時に FCC チェックを行う。FCC は、DATA部の総和の2の補数。

FCC コードが正常の場合は、フレームを有効とする。

FCC コードが不正の場合は、フレームを無効とし、破棄する。

(10) 誤り制御

ACK/NAKによる誤り制御は無し。

3. 6. 5 ECHONET Liteミドルウェアアダプタ通信ソフトウェアプロトコル

ECHONET Lite ミドルウェアアダプタ通信ソフトウェアプロトコルは、ECHONET Lite ミドルウェアアダプタのタイプにより異なる。本部において、以下の3つそれぞれについて、以降、節を分けて規定する。

- ① 機器インタフェース情報認識サービスソフトウェアプロトコル
- ② オブジェクト生成タイプ用通信ソフトウェアプロトコル
- ③ Peer to Peer タイプ用通信ソフトウェアプロトコル

3. 7 機器インタフェース情報認識サービス

ECHONET Lite ミドルウェアアダプタに接続した ECHONET Lite レディ機器の機器インタフェース情報を認識するためのサービス仕様を示す。

3. 7. 1 機器インタフェース情報認識サービス用フレーム構成

「機器インタフェース情報認識サービス」を実行する際のフレームの構成を図3-12のとおり規定する。フレームタイプコード (FT)、コマンド番号コード (CN)、フレーム番号コード (FN)、データ長コード (DL)、フレームデータ (FD) の部分が、ECHONET Lite ミドルウェアアダプタ通信インタフェースプロトコルのDATAとなる。

STX	FT	CN	FN	DL	FD	FCC
1byte	2byte	1byte	1byte	2byte	最大 16 byte	1byte

図3-12 機器インタフェース情報認識サービスソフトウェアプロトコル

(1) STX (制御コード)

先頭コード。機器インタフェース情報認識サービスソフトウェアプロトコルにおいては、0x02 固定とする。

(2) FT (Frame Type : フレームタイプ)

フレーム毎のタイプを示す。機器インタフェース情報認識用フレームは、0xFFFF 固定とする。

(3) CN (Command No. : コマンド番号コード)

コマンド番号コードは、ECHONET Lite ミドルウェアアダプタ通信インタフェースの機器インタフェース情報認識サービスソフトウェアプロトコルで規定されているサービスを指定するための1バイトのコードとする。

表3-7 機器インタフェース情報認識サービスのコマンドコード

		上位4ビット																
		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
下 位 4 ビ ット	0	機器インタフェース情報 要求								機器インタフェース情 報応答								
	1	機器インタフェース情報 確定通知								機器インタフェース情 報確定受理応答								
	2	-								-								
	3	-								-								
	4	-								-								
	5	-								-								
	6	-								-								
	7	-								-								
	8	-								-								
	9	-								-								
	A	-								-								
	B	-								-								
	C	-								-								
	D	-								-								
	E	-								-								
	F	-								-								

for future reserved

for future reserved

注) - : 機器インタフェース情報認識サービスコマンド for future reserved

(4) FN (Frame No : フレーム番号)

要求側が付与する番号。(0x01~0xFF) 応答フレームは対応する要求フレームと同じ番号にする必要がある。

なお、フレーム番号に対応できない ECHONET Lite レディ機器についてはこの番号を0x00 固定値とする。

(5) DL (Data Length : データ長コード)

データ長コードは、後に続くフレームデータ (FD) 部のサイズを示す2バイト長のコードとする。サイズは、バイト数とし、HEX表示する。本サービスにおいては、DLの取る最大値は、0x0010 とする。尚、本データの配置はビッグエンディアンとする。

(6) FD (Frame Data: フレームデータ)

フレームデータ部は、フレームタイプ (FT) 及びコマンド番号コード (CN) により規定されるデータのフィールドである。2バイト以上のデータの並びはビッグエンディアンとする。具体的な構成は、コマンド番号コード (CN) 毎に規定する。

(7) FCC (Frame Check Code: フレームチェックコード)

フレームチェックコードとして1バイトのチェックコードを規定する。

3. 7. 2 機器インタフェース情報認識サービス用コマンド

以下に、各ECHONET Lite ミドルウェアアダプタ通信インタフェースの機器インタフェース情報認識サービスで用いるコマンドの詳細を示す。

(1) 機器インタフェース情報要求/応答コマンド (Required)

ECHONET Lite ミドルウェアアダプタが、ECHONET Lite レディ機器が実装するECHONET Lite ミドルウェアアダプタ通信インタフェースの種類を取得する。

◇要求コマンドの方向

ECHONET Lite ミドルウェアアダプタ → ECHONET Lite レディ機器

◇要求コマンド

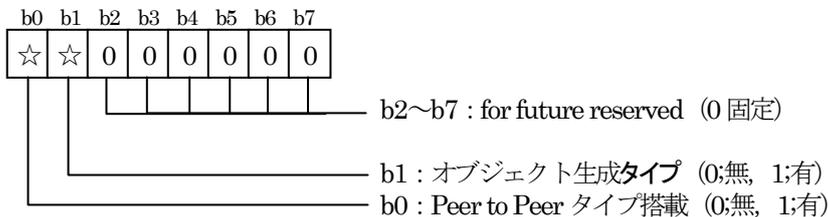
1バイト	2バイト	1バイト	1バイト	2バイト	1バイト
STX	FT	CN	FN	DL	FCC

STX : 0x02
 FT : 0xFFFF
 CN : 0x00
 FN : 0x**
 DL : 0x0000
 FCC : 0x**

◇応答コマンド

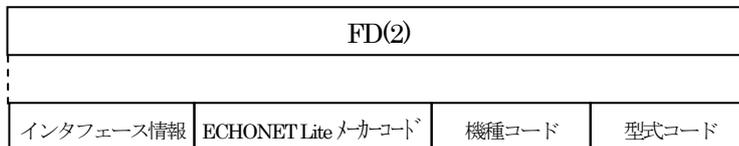
1バイト	2バイト	1バイト	1バイト	2バイト	1バイト	1バイト	nバイト	1バイト
STX	FT	CN	FN	DL	FD(0)	FD(1)	FD(2)	FCC

STX : 0x02
 FT : 0xFFFF
 CN : 0x80
 FN : 要求時の値 (機能を搭載していない場合は0x00)
 DL : n + 2
 FD(0) : ECHONET Lite レディ機器対応ミドルウェアアダプタ種別



FD(1) : 伝送速度
 0x00 : 2400bps
 0x01 : 4800bps
 0x02 : 9600bps
 0x03 : 19.2Kbps
 0x04 : 38.4Kbps
 0x05 : 57.6Kbps
 0x06 : 115Kbps
 0x07~0xFF : 未定義 (for future reserved)

- FD(2) : 方式別定義領域 (n byte)
 - オブジェクト生成タイプ: 無し (0Byte)
 - Peer to Peer タイプ



インタフェース情報 (1 バイト)

b0	b1	b2	b3	b4	b5	b6	b7
*	*	*	*	*	*	*	*

- b7,b6 : 通信シーケンス
- b7,b6=0,1 : アダプタポーリング 1,1 : 双方向
1,0 0,0 : 未定義
- b5 : フロー制御有無 0 : 無し 1 : 有り
- b4 : ACK/NAK応答有無 0 : 無し 1 : 有り
- b3 : ダウンロード有無 0 : 無し 1 : 有り
- b2~b0 : 未定義

ECHONET Lite メーカーコード (3 バイト)

ECHONET Lite 規定による

機種コード (2 バイト)

1 バイト目: クラスグループ 2 バイト目: クラスコード

型式コード (2 バイト)

メーカーコード*機種コード毎に独自規定

FCC : 0x**

(2) 機器インタフェース情報確定通知/確定受理応答コマンド (Required)

ECHONET Lite ミドルウェアアダプタは、ECHONET Lite レディ機器から指定された通信方式に対応可能か対応不可能かを、機器インタフェース情報確定通知により ECHONET Lite レディ機器に通知する。本通知は、機器インタフェース情報受信後、Ti=300msec 以内に行うものとする。対応可能とした場合は、機器インタフェース情報確定受理応答の待ち受けを行う。対応不可とした場合は、接続不可状態に遷移する。

ECHONET Lite レディ機器は、機器インタフェース情報確定通知を受信した場合、Result の値をチェックし、値が「対応可」である場合は、Ti=300msec 以内に機器インタフェース情報確定受理応答を ECHONET Lite ミドルウェアアダプタに送信後、未確認状態に遷移する。値が「対応不可」の場合は、機器インタフェース情報要求の待ち受けとなる。

ECHONET Lite ミドルウェアアダプタは、機器インタフェース情報確定通知送信後、300msec 以内に機器インタフェース情報確定受理応答を受信した場合は、未確認状態に遷移にする。300msec を経過しても受信しない場合は、機器インタフェース情報要求送信を再度行う。

◇要求コマンドの方向

ECHONET Lite ミドルウェアアダプタ → ECHONET Lite レディ機器

◇要求コマンド

1バイト	2バイト	1バイト	1バイト	2バイト	1バイト	1バイト
STX	FT	CN	FN	DL	FD(0)	FCC

STX : 0x02
 FT : 0xFFFF
 CN : 0x01
 FN : 0x**
 DL : 0x0001
 FD(0) : Result
 0x01 : 対応不可
 0x00 : 対応可
 0x02 : 現行速度対応可 (指定速度対応不可)
 0x11 : 複数指定時 Peer to Peer タイプ対応可
 0x12 : 複数指定時オブジェクト生成タイプ対応可
 FCC : 0x**

◇応答コマンド

1バイト	2バイト	1バイト	1バイト	2バイト	1バイト
STX	FT	CN	FN	DL	FCC

STX : 0x02
 FT : 0xFFFF
 CN : 0x81
 FN : 要求時の値 (機能を搭載していない場合は 0x00)
 DL : 0x0000
 FCC : 0x**

3. 7. 3 機器インタフェース情報認識サービスシーケンス

図3-13に本サービスの動作シーケンス、図3-14に、ECHONET Lite ミドルウェアアダプタ、ECHONET Lite レディ機器双方の状態遷移図を示す。両図に示すように、ECHONET Lite ミドルウェアアダプタ、ECHONET Lite レディ機器双方は、機器インタフェース情報認識サービスが正常に実行できるまでを「未認識」状態とし、可能な伝送速度で、順次機器インタフェース情報認識サービスの実行を試みる。機器インタフェース情報認識サービスによって ECHONET Lite レディ機器の通信方式を認識し、以降はその方式で通信を行う。

ECHONET Lite レディ機器、ECHONET Lite ミドルウェアアダプタ共に、起動後、またはいずれかの側がリセットなどされた場合を考慮し、各通信方式においては、同期ずれ等正常に通信できない状況(通信不良)を検出した場合は、適当な回復策(複数回試みるなど)を実施後、回復しない場合は図3-14に示すような状態遷移で、未認識状態に遷移する必要がある。

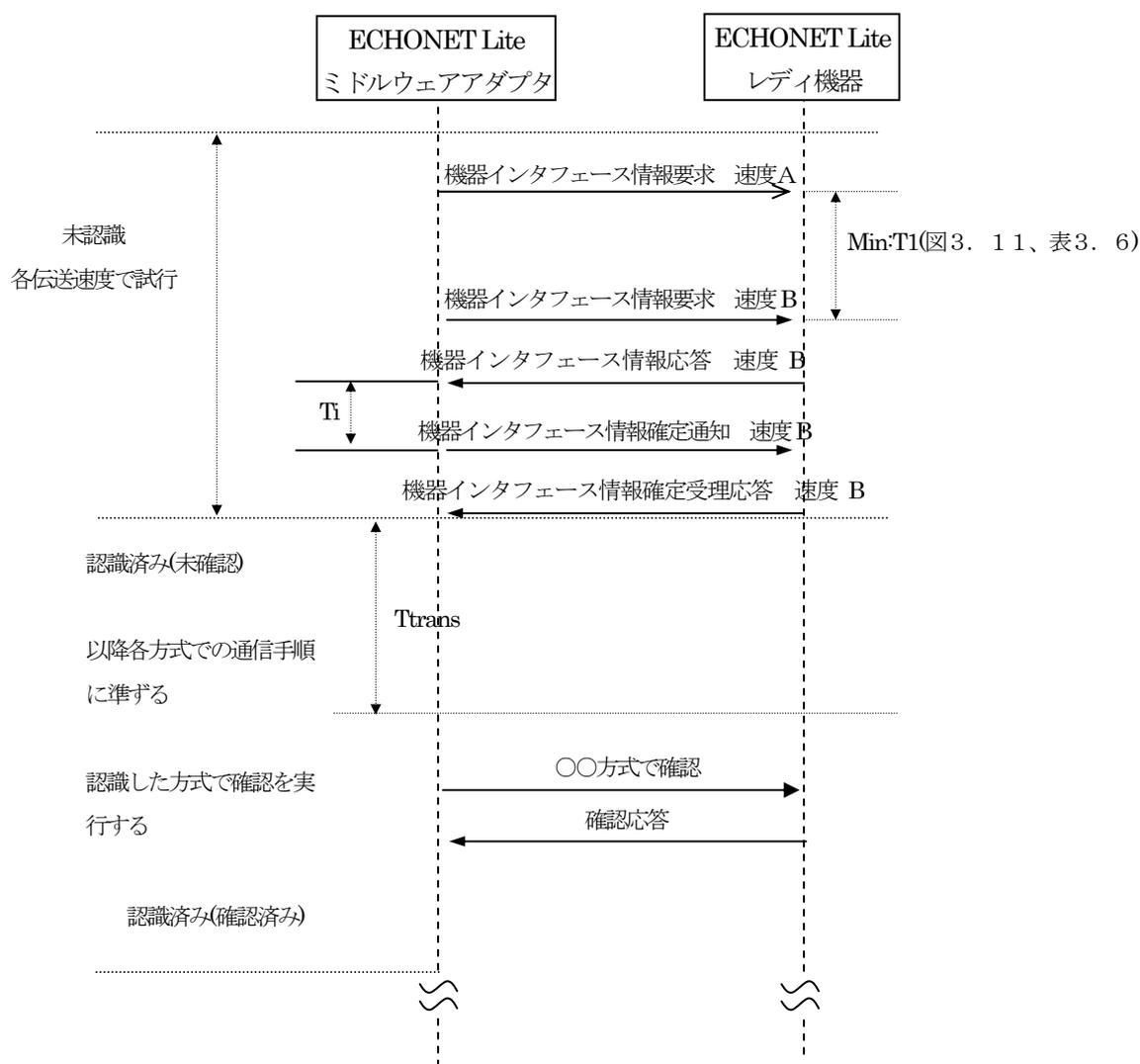


図3-13 動作シーケンス

T_{trans} : 各方式仕様への移行時間。500msec 以上あけること。

T_i : MAX.300msec

アダプタは、応答を正常に認識できない場合はT₁時間(図3-11)後再度認識サービスを発行する。

3. 7. 4 タイプ共通状態遷移図

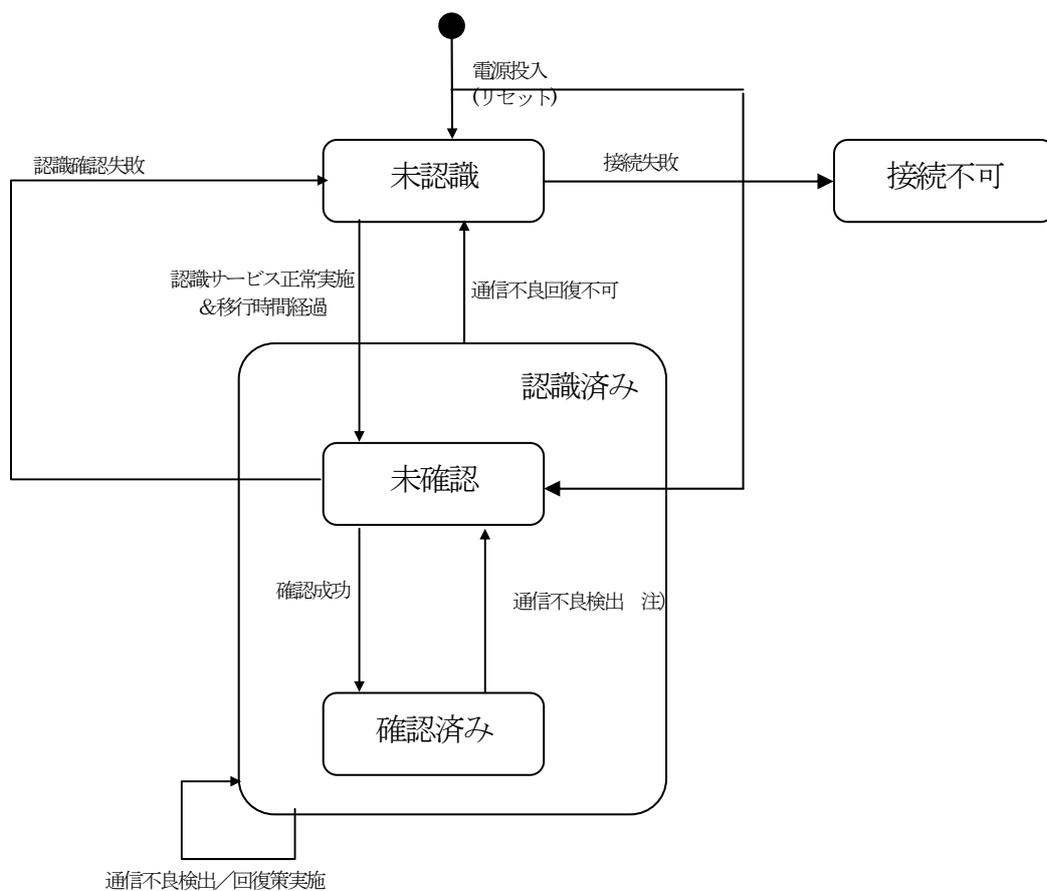


図3-14 状態遷移

通信不良：各方式の通信手順による。

注) 通信不良検出時に、ECHONET Lite ミドルウェアアダプタは再度確認処理を実施することも可。

電源投入時に、未認識状態から起動するか、認識済みから起動するか否かは、実装に委ねる。

表3-8 状態定義

状態名		状態定義	備考
未認識		相手との通信方式の認識を完了していない状態。 通信不良検出後。	
		ECHONET Lite レディ機器側： アダプタからの認識サービスの要求待ち状態。 ECHONET Lite ミドルウェアアダプタからの機器インタフェース情報確定通知が「対応不可」の場合は、機器インタフェース情報要求の待ち受けを行う。 「対応可」の場合は、機器インタフェース情報確定受理応答送信後、未確認状態に遷移する。 ECHONET Lite ミドルウェアアダプタ側： ECHONET Lite レディ機器へ認識サービスの要求試行状態。 認識が完了するまで、任意の間隔で適宜可能な認識サービスを試行する。複数の速度A、Bを適宜繰り返す。 機器インタフェース情報応答を受け取った場合、その内容での通信が可能ならば、「対応可」とした機器インタフェース情報確定通知をECHONET Lite レディ機器に送信し、その受理応答を受けた後に未確認状態に遷移する。受け取った情報での通信が不可能な場合は、「対応不可」とした機器インタフェース情報確定通知をECHONET Lite レディ機器に送信し、接続不可状態に遷移する。	速度を繰り返す順序、試行回数、間隔(Min : T1 図3-11)は規格外とする。各アダプタの性質に合わせて最適な方法を実行する。
接続不可状態		ECHONET Lite ミドルウェアアダプタ、ECHONET Lite レディ機器間で一致する通信方式が存在しない場合に遷移する状態。	通信不能の異常処理を行う。
認識済み	未確認	相手との通信方式の認識を完了し、方式毎の通信手順で通信をしている状態。	通信不良を検出後、適当な回復処理を実行後にも回復しない場合には「未認識」状態に移行する。
	確認済み		方式毎の通信手順で、正常に通信が確認できた状態。

3. 7. 5 異常処理

ECHONET Lite ミドルウェアアダプタは、以下のような状態になった場合、ノードプロファイルオブジェクト、あるいは機器オブジェクトに異常をセットして、通信が可能な場合、他のノードに異常をアナウンスする。

(a)通信不能

機器インタフェース情報認識サービスにより、ECHONET Lite レディ機器との通信が不可能であることが判明した場合。ノードプロファイルオブジェクト異常内容プロパティ (EPC=0x89) に 0x03E9 (ミドルウェアアダプタ認識異常) をセットした後、異常発生状態プロパティ (EPC=0x88) に 0x41 をセットする。

(b)設定異常

確認済み状態のときに行うパラメータ設定に失敗し、その続行が不可能と判断した場合。異常の要因に応じてノードプロファイルオブジェクト異常内容プロパティ (EPC=0x89) に下記の異常内容コードをセットし、異常発生状態プロパティ (EPC=0x88) に 0x41 をセットする。なお、具体的な詳細要因は、各通信方法に依存する。

異常内容コード

0x03EA : オブジェクト異常

0xE3EB : アダプタ初期化異常

0x03EC : その他設定異常

(c)通信異常

確認済み状態において、ECHONET Lite レディ機器との通信ができない場合。通信異常となった機器に対応する機器オブジェクト異常内容プロパティ (EPC=0x89) に 0x03E9 (通信異常) をセットした後、異常発生状態プロパティ (EPC=0x88) に 0x41 をセットする。また、通信異常と判定する条件については、各通信方法に依存する。

3. 8 オブジェクト生成タイプ用通信プロトコル

ECHONET Lite ミドルウェアアダプタと ECHONET Lite レディ機器間の通信インタフェースの種類が、オブジェクト生成タイプの場合の通信ソフトウェアプロトコルを規定する。

ミドルウェアアダプタのタイプとして、以下の2種類を規定する。

基本ミドルウェアアダプタ
拡張ミドルウェアアダプタ

拡張ミドルウェアアダプタとは、ミドルウェアアダプタインタフェース上において、ECHONET Lite ネットワーク上に存在する他のノードのアドレス情報を処理できるアダプタである。一方、基本ミドルウェアアダプタとは、ミドルウェアアダプタインタフェース上において、ECHONET Lite ネットワーク上に存在する他のノードのアドレス情報を処理できないアダプタである。また、拡張ミドルウェアアダプタは、配列プロパティをその要素に空きが存在するか、しないかを区別して処理できるアダプタである。拡張ミドルウェアアダプタは、基本ミドルウェアアダプタの機能を全て含むものとする。

なお、通信定義オブジェクトについては、拡張ミドルウェアアダプタについては全てをオプション機能で実装可能とする。

なお、拡張相当の処理要求を行う ECHONET Lite レディ機器が、基本ミドルウェアアダプタに接続された場合、機器は以下の処理を行わなければならない。

- ・他ノードの状態の参照、変更ができないこと表示する。
- ・相手先を指定しての通知は、一斉同報の通知で行う。
- ・他ノードからの要求に対する応答は、基本ミドルウェアアダプタ通信フレームで行う。

本バージョンにおいては、拡張ミドルウェアアダプタへの対応を考慮した上で基本ミドルウェアアダプタについて規定する。

基本ミドルウェアアダプタは、少なくとも3個の機器オブジェクトを内部に生成し、管理できなくてはならない。管理可能な機器オブジェクト数が3個の場合は、プロパティの値を保持するための領域として、最小1KBを確保する必要がある。

また、4個以上の機器オブジェクトを生成、管理するためには、オプション(Optional)コマンドを使用する。この場合は、プロパティの値を保持するための領域として、最小2KBを確保する必要がある。

基本ミドルウェアアダプタは、規格書第2部で搭載必須となっているノードプロファイルオブジェクト等のオブジェクトクラスは必須で搭載するものとする。

3. 8. 1 オブジェクト生成タイプ用フレーム構成

オブジェクト生成タイプソフトウェアプロトコル（フレーム）の構成を図3-15のとおり規定する。フレームタイプコード（FT）、コマンド番号コード（CN）、フレーム番号コード（FN）、データ長コード（DL）、フレームデータ（FD）の部分が、ECHONET Lite ミドルウェアアダプタ通信インタフェースプロトコルのDATAとなる。図3-15の構成は、機器インタフェース情報認識サービスと同じである。

STX	FT	CN	FN	DL	FD	FCC
1byte	2byte	1byte	1byte	2byte	nbyte	1byte

図3-15 オブジェクト生成タイプ ECHONET Lite ミドルウェアアダプタ通信ソフトウェアプロトコル

(1) STX (制御コード)

制御コード。 0x02 固定とする。

(2) FT (Frame Type : フレームタイプ)

フレーム毎のタイプを示す。

b15~b12 : Version

フレームのバージョン番号を示す。0000 とする。

b11~b8 : for future reserved (0000)

b7~b0 : Type

各種コマンドのフレームタイプを示す。

0x00 : 機器インタフェース情報確認フレーム

0x01 : アダプタ初期化用フレーム

0x02 : オブジェクト構築用フレーム

0x03 : 基本 ECHONET Lite 通常フレーム

0x04 : 拡張 ECHONET Lite 通常フレーム

0x05~0xDF : for future reserved

0xE0~0xFE : ユーザ定義領域

0xFF : エラー通知フレーム

ただし、以下の機器インタフェース情報認識フレームと同値を取らないこと

0xFFFF : 機器インタフェース情報認識用フレーム

(3) CN (Command No. : コマンド番号コード)

コマンド番号コードは、ECHONET Lite ミドルウェアアダプタ通信インタフェースのオブジェクト生成タイプソフトウェアプロトコルで規定するサービスを指定するための1バイトのコードとする。本規格においては、表3-9で示したコマンドを規定する。具体的な割り当ての無いコードについては、for future reserved とする。

表3-9 オブジェクト生成タイプインタフェースのコマンドコード

フレームタイプ (F T)	コマンド番号 コード (CN)	コマンド名	搭載
機器インタフェース確認モード 機器インタフェース情報確認フレーム(0x0000)を使用	0x00	機器インタフェース情報確認要求	Required
	0x80	機器インタフェース情報確認応答	Required
アダプタ初期化モード アダプタ初期化用フレーム(0x0001)を使用	0x01	アダプタ初期化設定要求	Required
	0x81	アダプタ初期化設定応答	Required
	0x02	アダプタ初期化完了通知	Required
	0x82	アダプタ初期化完了通知受理応答	Required
オブジェクト構築モード オブジェクト構築用フレーム(0x0002)を使用	0x00	機器問い合わせ要求	Required
	0x80	機器問い合わせ応答	Required
	0x01	機器問い合わせ完了通知	Required
	0x81	機器問い合わせ完了通知受理応答	Required
	0x02	アダプタ立ち上げ通知	Required
	0x82	アダプタ立ち上げ通知受理応答	Required
	0x03	オブジェクト指定機器問合せ要求	Optional*1
	0x83	オブジェクト指定機器問合せ応答	Optional*1
ECHONET Lite 通信モード 基本 ECHONET Lite 通常フレーム(0x0003)を使用	0x10	機器状態アクセス要求	Required
	0x90	機器状態アクセス応答	Required
	0x11	機器状態通知要求	Required*2
	0x91	機器状態通知応答	Required
	0x12	要素指定機器状態アクセス要求	Required
	0x92	要素指定機器状態アクセス応答	Required
	0x13	要素指定機器状態通知要求	Required*2
	0x93	要素指定機器状態通知応答	Required
	0x14	オブジェクトアクセス要求	Required*2
	0x94	オブジェクトアクセス応答	Required
	0x20	機器状態アクセス一括要求	Optional
	0xA0	機器状態アクセス一括応答	Optional
	0x21	機器状態アクセス一括 UP 要求	Optional
	0xA1	機器状態アクセス一括 UP 応答	Optional
	0x22	機器状態通知一括要求	Optional
	0xA2	機器状態通知一括応答	Optional
	0x23	オブジェクトアクセス一括要求	Optional
0xA3	オブジェクトアクセス一括応答	Optional	

注) *1: 生成する機器オブジェクト数4以上の場合には Required。

*2: 機器側は Optional。

(4) FN (Frame No: フレーム番号)

要求側が付与する番号。(0x01~0xFF) 要求側は順番に番号を付与する必要がある。応答フレームは対応する要求フレームと同じ番号にする必要がある。

なお、フレーム番号に対応できない ECHONET Lite レディ機器についてはこの番号を 0x00 固定値とする。

(5) DL (Data Length : データ長コード)

データ長コードは、後に続くフレームデータ (FD) 部のサイズを示す2バイト長のコードとする。サイズは、バイト数とし、HEX表示する。例えば、FD部が20バイトの場合には、DLは20バイトを示す0x0014 となる。尚、本データの配置はビッグエンディアンとする。

(6) FD (Frame Data: フレームデータ)

フレームデータ部は、フレームタイプ (FT) 及びコマンド番号コード (CN) により規定されるデータのフィールドである。2バイト以上のデータの並びはビッグエンディアンとする。具体的な構成は、コマンド番号コード (CN) 毎に規定する。

(7) FCC (Frame Check Code: フレームチェックコード)

フレームチェックコードとして1バイトのチェックコードを規定する。

3. 8. 2 アダプタ内部サービス

ミドルウェアアダプタは、自ノードプロパティとしてアダプタ内に値を保持するものと値を保持しないでアダプタを通過するものとをプロパティ個別に設定できる。従って、他のノードから Set、SetM (機器状態の変更)、あるいは Get、GetM (機器状態の参照) といったサービスを受信した場合に、そのサービスをアダプタがどのように処理するかをアダプタ内部サービスとして IASet、IASetup、IASetM、IASetMup、IAGet、IAGetup、IAGetM、IAGetMup の8種類を規定し、プロパティ毎に個々に設定可能とする。内部サービスとして、Set を受け付けた場合には IASet、IASetup のいずれかが、SetM を受け付けた場合には IASetM、IASetMup のいずれかが、Get を受け付けた場合には IAGet、IAGetup のいずれかが、GetM を受け付けた場合には IAGetM、IAGetMup のいずれかが行われる。

A) アダプタ内に値を保持するサービス ECHONET Lite レディ機器の通信負荷を軽くするためにアダプタで応答する。	IASet、IASetM、IAGet、IAGetM
B) アダプタ内に値を保持しないサービス リアルタイム性を要求されるプロパティに対しアダプタを通過させてECHONET Lite レディ機器で応答する。	IASetup、IASetMup、IAGetup、IAGetMup

上記 A) B) どちらのサービスを選択するかについては、プロパティ毎に設定可能で、オブジェクト構築コマンドにより ECHONET Lite レディ機器から機器間問い合わせデータとして取得する。

なお、IASetM、IAGetM については将来規定とし、本バージョンでは使用しないものと

する。

3.8.2.1 IASet/IASetM

IASet/IASetMとは、他のノードからSet/SetMを要求されたミドルウェアアダプタが、ミドルウェアアダプタ内部に存在する機器オブジェクトの該当する領域に指定された値を書き込むアダプタ内部サービスである。

他ノードに対する受理/不可応答は、ミドルウェアアダプタが値を書き込んだタイミングとなる。

本サービスでは、ECHONET Lite ミドルウェアアダプタは他ノードからの要求の値をECHONET Lite レディ機器へ通知することは行わない為、ECHONET Lite レディ機器は、定期的にミドルウェアアダプタの機器オブジェクトを参照することで、他ノードから状態変更の依頼を確認する。図にその動作を示す。

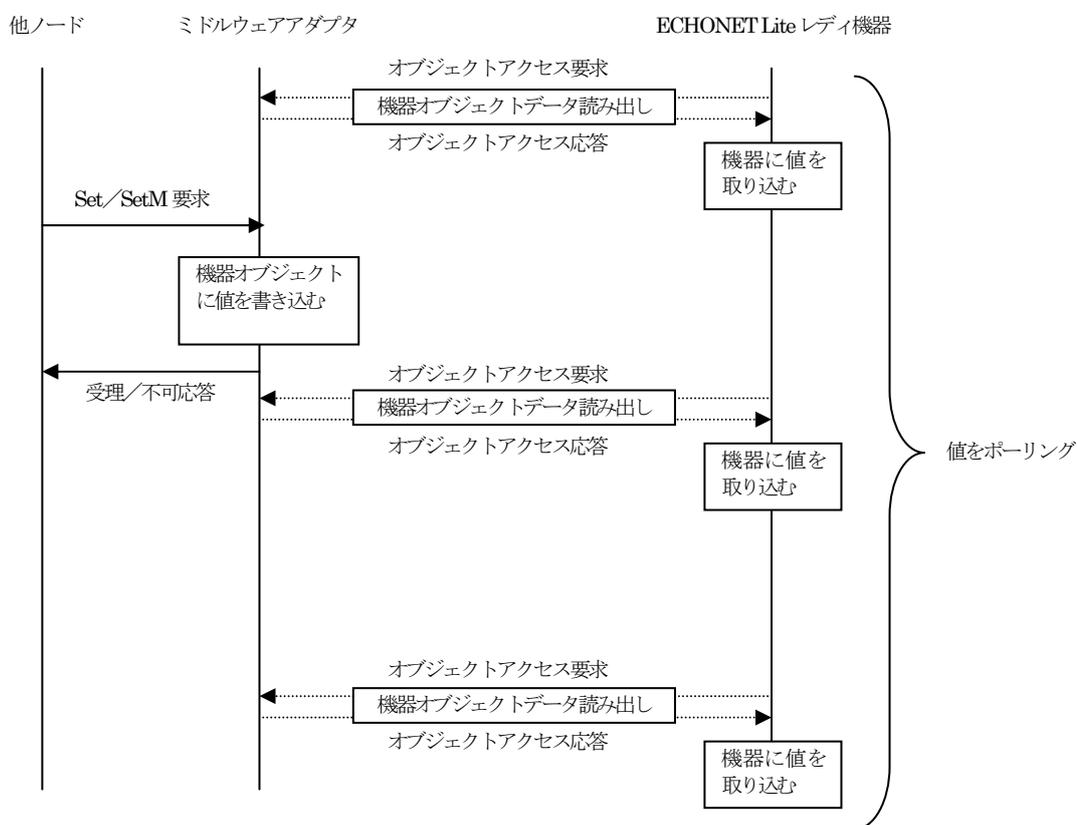


図 3-16 IASet (IASetM) 動作

3.8.2.2 IASetup/IASetMup

IASetup/IASetMup とは、他のノードから Set/SetM を要求されたミドルウェアアダプタが、その要求（状態取得要求）を ECHONET Lite レディ機器に伝えるアダプタ内部サービスである。

他ノードに対する受理応答は、ミドルウェアアダプタが ECHONET Lite レディ機器に IASetup の応答として ECHONET Lite レディ機器から機器状態アクセス応答を受け取ったタイミングとなる。図にその動作を示す。機器状態アクセス応答 (IASetMup の場合は要素指定機器状態アクセス応答) が返ってこなかった場合はタイムアウトして他ノードに不可応答を返す。

ただし、プロパティが存在しない場合は、機器状態アクセス要求を行わず他ノードに不可応答を返す。

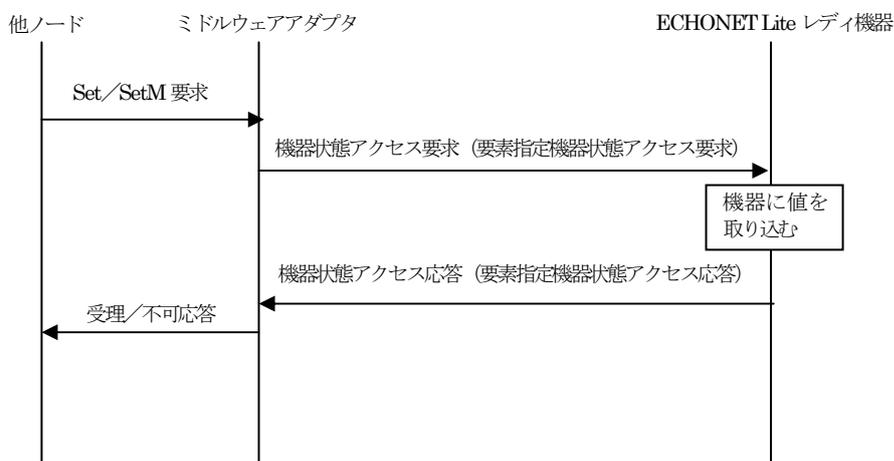


図3-17 IASetup (IASetMup) の動作

3.8.2.3 IAGet/IAGetM

IAGet/IAGetM とは、他のノードから Get/GetM を要求されたミドルウェアアダプタが、ミドルウェアアダプタ内部に存在する機器オブジェクトの該当する領域の指定された値を応答するアダプタ内部サービスである。

ECHONET Lite レディ機器は、自己の状態が変わるごとに、ミドルウェアアダプタの機器オブジェクトの対応するプロパティ値を変更する必要がある。

ただし、プロパティが存在しない場合は、他ノードに不可応答を返す。

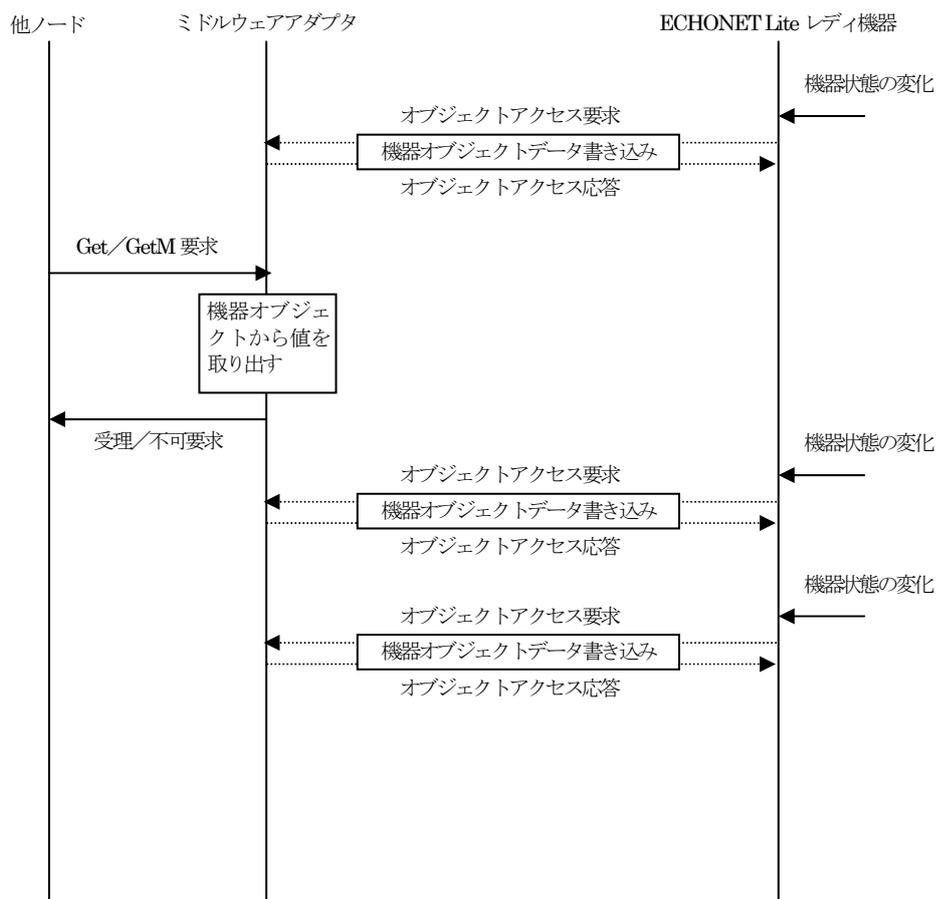


図 3-1 8 IAGet (IAGetM) の動作

3.8.2.4 IAGetup/IAGetMup

IAGetup/IAGetMup とは、他のノードから Get/GetM を要求されたミドルウェアアダプタが、その要求（状態変更要求）を ECHONET Lite レディ機器に伝えるアダプタ内部サービスである。

他ノードに対する値の応答は、ECHONET Lite レディ機器から対応するプロパティ値の応答が返ってきたタイミングとなる。

ただし、プロパティが存在しない場合は、機器状態アクセス要求を行わず他ノードに不可応答を返す。

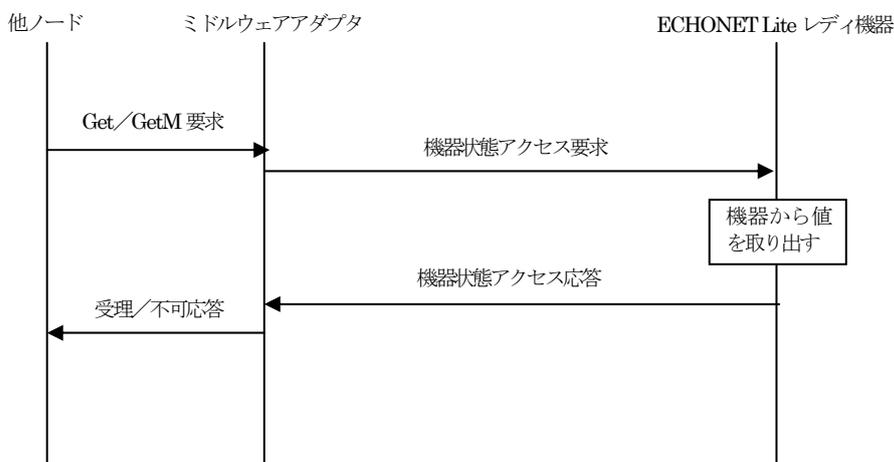


図3-19 IAGetup(IAGetMup)の動作

3.8.2.5 状態アナウンスとアダプタ内部サービス

状態アナウンスプロパティマッププロパティの対応するビットが1とされている（状態設定されている）プロパティは、アダプタ内部サービスが IAGet、IAGetup、IAGetM、IAGetMup となっている場合、その値が変化したタイミングでドメイン内一斉同報を行わなければならない。

IAGet、IAGetM に設定されている場合は、ミドルウェアアダプタがドメイン内一斉同報の処理を行う。IAGetup、IAGetMup に設定されている場合は、ECHONET Lite レディ機器がドメイン内一斉同報の処理を行う。

3. 8. 3 オブジェクト生成タイプ用状態遷移

本項では、各 ECHONET Lite ミドルウェアアダプタ通信インタフェースのオブジェクト生成タイプを実装する場合の、ECHONET Lite ミドルウェアアダプタの状態遷移を規定する。ECHONET Lite ミドルウェアアダプタは、内部でのオブジェクト生成が完了するまでは、ECHONET Lite への参入処理は行わないものとする。

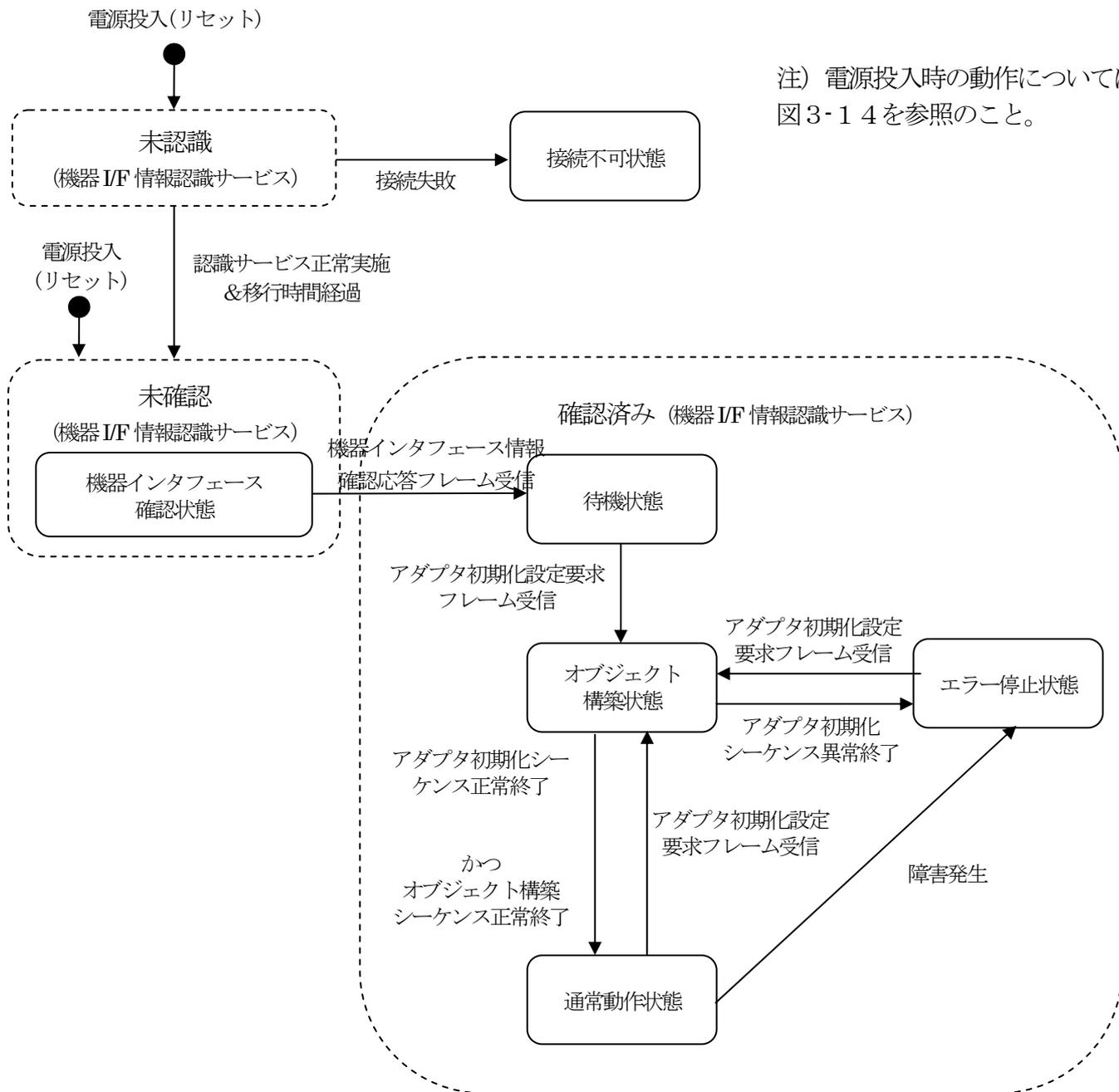


図3-20 ミドルウェアアダプタ状態遷移

3.8.3.1 機器インタフェース確認状態

電源投入、あるいは外部からのリセット入力直後、または機器インタフェース情報認識サービスの認識済みからの移行状態で、機器インタフェース情報認識サービスでの認識確認処理を行うまでの状態。機器インタフェース情報確認処理以外のフレームには状態不一致エラーを返す。また、ECHONET Lite ネットワーク上からの入力はすべて破棄する。機器インタフェース情報確認処理終了後、待機状態に遷移する。

機器インタフェース情報確認処理が規定回数異常終了した場合は、機器インタフェース情報の未認識状態へ遷移する。

3.8.3.2 待機状態

ECHONET Lite レディ機器とのインタフェース状態確認が終了、または機器オブジェクト構築が成功し、アダプタ初期化要求フレーム待ちの状態。アダプタ初期化設定要求フレーム以外のフレームには状態不一致エラーを返す。また、ECHONET Lite ネットワーク上からの入力はすべて破棄する。

3.8.3.3 オブジェクト構築状態

アダプタ初期化設定要求フレームの入力により、「待機状態」、「エラー停止状態」、「通常動作状態」から遷移する状態で、アダプタ側初期化とオブジェクト構築シーケンスを実行している状態である。この間、アダプタ初期化モードとオブジェクト構築モード関連フレーム以外のフレームには状態不一致エラーを返す。

ミドルウェアアダプタは、この期間に、通信ミドルウェアのコールドスタートを行う。同時に、オブジェクト構築シーケンスを実行する。

アダプタ側初期化シーケンスが正常に終了し、かつ、オブジェクト構築シーケンス一連の処理が正常終了した場合、通常動作状態に遷移する。異常終了した場合は、エラー停止状態に遷移する。

3.8.3.4 エラー停止状態

ECHONET Lite ミドルウェアアダプタは、ECHONET Lite ミドルウェアアダプタの初期化シーケンスの失敗、もしくは、通常動作状態での障害発生時にエラー停止状態へ遷移する。本状態に遷移した場合は、3.7.5項に記載されている設定異常の処理を行う。異常内容プロパティにどのような値をセットするかは、エラー停止状態に遷移することになった要因により変化する。要因と異常内容コードの関係を以下に示す。

オブジェクト構築に失敗した場合	: 0x03EA (オブジェクト異常)
アダプタ初期化シーケンスに失敗した場合	: 0x03EB (アダプタ初期化異常)
通常動作中障害発生	: 0x03EC (その他設定異常)

3.8.3.5 通常動作状態

オブジェクト構築状態におけるオブジェクト構築シーケンスが正常終了した場合に、本状態に遷移する。このとき初期値が必要なプロパティにおいては、ミドルウェアアダプタは、レディ機器に対して機器状態アクセス要求を発行することで初期値を取得する。本状態においてのみ、通常の ECHONET Lite 通信が可能となる。本状態においては、オブジェクト構築モード関連フレームは受け付けない。

3. 8. 4 オブジェクト生成タイプ用コマンド

本項では、各 ECHONET Lite ミドルウェアアダプタ通信インタフェースのオブジェクト生成タイプで用いるコマンドの詳細を規定する。

3.8.4.1 機器インタフェース確認モード

(1) 機器インタフェース情報確認要求/応答コマンド (Required)

ECHONET Lite ミドルウェアアダプタが、保持している機器アダプタインタフェースの通信方式とオブジェクトを確認するため ECHONET Lite レディ機器に問い合わせる。オブジェクトが存在しない場合は、オブジェクト数0で問い合わせる。

ECHONET Lite レディ機器は、ミドルウェアアダプタ種別とオブジェクトの一致を確認し、応答する。

◇要求コマンドの方向

ECHONET Lite ミドルウェアアダプタ → ECHONET Lite レディ機器

◇要求コマンド

1バイト	2バイト	1バイト	1バイト	2バイト	1バイト	1バイト	n+1バイト	1バイト
STX	FT	CN	FN	DL	FD(0)	FD(1)	FD(2)	FCC

- STX : 0x02
- FT : 0x0000
- CN : 0x00
- FN : 0x**
- DL : 0x****
- FD(0) : ミドルウェアアダプタ種別
 - 0x01 Peer to Peer 方式
 - 0x02 オブジェクト生成方式
 - 0x03 オブジェクト内蔵方式
 - 0x04~0xFF 未定義 (for future reserved)
- FD(1) : 伝送速度
 - 0x00 : 2400bps, 0x01 : 4800bps
 - 0x02 : 9600bps, 0x03 : 19.2Kbps
 - 0x04 : 38.4Kbps, 0x05 : 57.6Kbps
 - 0x06 : 115Kbps, 0x07~0xFF : for future reserved
- FD(2) : アダプタ保持オブジェクト定義領域 (0~n+1 バイト)
 - nは、18×<オブジェクト数>の値。
 - アダプタ内で保持しているオブジェクトがない場合は、FD(2)は0バイトとする。

<オブジェクト数><EOJ><メーカーコード><商品コード>
 オブジェクト数分を列挙 (最大3)

- EOJ アダプタ内で保持しているオブジェクトコード (3バイト)
- メーカーコード アダプタ内で保持しているメーカーコード (3バイト)
- 商品コード アダプタ内で保持している商品コード (12バイト)
- FCC : 0x**

◇応答コマンド

1バイト	2バイト	1バイト	1バイト	2バイト	2バイト	1バイト
STX	FT	CN	FN	DL	FD(0)	FCC

STX : 0x02
 FT : 0x0000
 CN : 0x80
 FN : 要求時の値 (機能を搭載していない場合は0x00)
 DL : 0x0002
 FD(0) : 処理結果

FD(0)の値	内容
0x0000	正常終了
0x0011	アダプタ種別不一致エラー
0x0012	オブジェクト不一致エラー
0x0021	機器インタフェース情報破棄
0xFFFF	その他のエラー

FCC : 0x**

3.8.4.2 アダプタ初期化モード

(1) アダプタ初期化設定要求/応答コマンド (Required)

ECHONET Lite レディ機器から基本ミドルウェアアダプタに対し、ECHONET Lite 通信ミドルウェア以下の初期化を要求する。

基本ミドルウェアアダプタは、アダプタ初期化設定要求を受け取ることで、待機状態からオブジェクト構築状態へと遷移する。

◇要求コマンドの方向

ECHONET Lite レディ機器 → ECHONET Lite ミドルウェアアダプタ

◇要求コマンド

1バイト	2バイト	1バイト	1バイト	2バイト	2バイト	1バイト
STX	FT	CN	FN	DL	FD(0)	FCC

STX : 0x02
 FT : 0x0001
 CN : 0x01
 FN : 0x** (機能を搭載していない場合は0x00)
 DL : 0x0002
 FD(0) : 初期化方法

FD(0)の値	内容
0x0001	機器情報保持スタート 基本ミドルウェアアダプタ内に既に機器オブジェクトが存在する場合は、それを保持したままで動作することを要求する。機器オブジェクトが存在しない場合は、機器問い合わせ要求が行われ、機器オブジェクト生成を行う。
0x0002	機器情報破棄スタート 基本ミドルウェアアダプタ内に既に機器オブジェクトが存在する場合は、それを破棄してスタート
0x0003 ~ 0xFFFF	for future reserved

FCC : 0x**

◇応答コマンド

1バイト	2バイト	2バイト	1バイト	2バイト	2バイト	1バイト	8バイト	1バイト
STX	FT	CN	FN	DL	FD(0)	FD(1)	FD(2)	FCC

STX : 0x02
 FT : 0x0001
 CN : 0x81
 FN : 要求時の値
 DL : 0x0002 / 0x000B
 FD(0) : 応答結果

FD(0)の値	内容
0x0000	初期化受理
0x0011	初期化拒否
0x0101	状態不一致エラー (機器インタフェース確認状態 (未確認))

0xFFFF	その他のエラー

FD(1) : 下位通信層ID

※第2部6. 11. 1項 ノードプロファイルクラス詳細規定のs 識別番号プロパティを参照。

FD(2) : 識別番号の固有番号フィールド

※第2部6. 11. 1項 ノードプロファイルクラス詳細規定の識別番号プロパティを参照。識別番号が8バイトより長い場合は、0x00を設定。

FCC : 0x**

(2) アダプタ初期化完了通知／通知受理応答コマンド (Required)

基本ミドルウェアアダプタは、初期化が終了した場合、アダプタ初期化完了通知を ECHONET Lite レディ機器に送らなければならない。

初期化が正常に終了した場合は、「初期化正常終了」を送る。

初期化に失敗した場合は、「初期化異常終了」を渡す。

ECHONET Lite レディ機器は、アダプタ初期化完了通知を受け取った場合、アダプタ初期化完了通知受理応答を返さなくてはならない。

◇要求コマンドの方向

ECHONET Lite ミドルウェアアダプタ → ECHONET Lite レディ機器

◇要求コマンド

1バイト	2バイト	1バイト	1バイト	2バイト	2バイト	1バイト
STX	FT	CN	FN	DL	FD(0)	FCC

STX : 0x02
 FT : 0x0001
 CN : 0x02
 FN : 0x**
 DL : 0x0002
 FD(0) : 初期化結果通知

FD(0)の値	内容
0x0000	初期化正常終了
0x0011	初期化異常終了

FCC : 0x**

◇応答コマンド

1バイト	2バイト	1バイト	1バイト	2バイト	2バイト	1バイト
STX	FT	CN	FN	DL	FD(0)	FCC

STX : 0x02
 FT : 0x0001
 CN : 0x82
 FN : 要求時の値 (機能を搭載していない場合は0x00)
 DL : 0x0002
 FD(0) : 応答結果

FD(0)の値	内容
0x0000	通知受理
0xFFFF	その他のエラー

FCC : 0x**

3.8.4.3 オブジェクト構築モード

(1) 機器問合せ要求/応答コマンド (Required)

基本ミドルウェアアダプタは、内部に機器オブジェクトが存在しない場合に、ECHONET Lite レディ機器に対し機器問い合わせ要求を行う。

この要求に対し、ECHONET Lite レディ機器からは機器の情報が機器問い合わせ応答として返される。

機器問い合わせ要求を受け取った ECHONET Lite レディ機器は、機器オブジェクトを基本ミドルウェアアダプタに構築するために必要な情報を、機器問い合わせ応答として基本ミドルウェアアダプタに渡す。構築可能最大機器オブジェクト数は3である。3つの機器に関する情報を1つの応答で送る、あるいは2つの応答、または3つの応答に分けて送ることができる。

◇要求コマンドの方向

ECHONET Lite ミドルウェアアダプタ → ECHONET Lite レディ機器

◇要求コマンド

1バイト	2バイト	1バイト	1バイト	2バイト	1バイト
STX	FT	CN	FN	DL	FCC

STX : 0x02
 FT : 0x0002
 CN : 0x00
 FN : 0x**
 DL : 0x0000
 FCC : 0x**

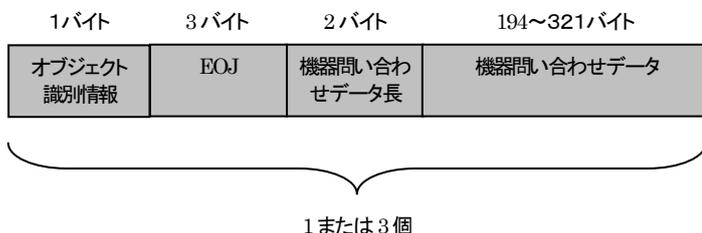
◇応答コマンド

1バイト	2バイト	1バイト	1バイト	2バイト	2バイト	1バイト	*バイト	1バイト
STX	FT	CN	FN	DL	FD(0)	FD(1)	FD(2)	FCC

STX : 0x02
 FT : 0x0002
 CN : 0x80
 FN : 要求時の値 (機能を搭載していない場合は0x00)
 DL : 0x****
 FD(0) : 応答結果
 0x0000 正常
 0xFFFF その他のエラー
 FD(1) : フレーム内送信オブジェクト数
 機器問合せ応答フレームに含まれるオブジェクトの数
 FD(2) : オブジェクトデータ
 オブジェクトデータフォーマットを参照
 FCC : 0x**

<オブジェクトデータフォーマット>

以下のフォーマットを1個または3個実装する。



名称	サイズ (byte)	説明
オブジェクト識別情報	1	アダプタが管理すべきオブジェクトの総数情報と各オブジェクトを管理するために識別番号を付与する。 b7~b4: 総オブジェクト数 (1:0001、2:0010、3:0011) b3~b0: オブジェクト識別番号 (1:0001、2:0010、3:0011)
EOJ	3	生成するオブジェクトの ECHONET オブジェクトコード。
機器間い合わせデータ長	2	機器間い合わせデータのサイズを示す。
機器間い合わせデータ	194~321	機器間い合わせデータフォーマットを参照。

<機器間い合わせデータフォーマット>



名称	サイズ (byte)	説明
有効/無効ビットマップ	2	以降に連続するデータの中で意味のあるものを示す。 b15: SetM プロパティマップ b14 : Set プロパティマップ b13 : GetM プロパティマップ b12: Get プロパティマップ b11: 状態アナウンスプロパティマップ b10: IASetup プロパティ動作マップ b9: IAGetup プロパティ動作マップ b8: IASetMup プロパティ動作マップ b7: IAGetMup プロパティ動作マップ b6: Version 情報 b5: メーカーコード b4: 事業所コード b3: 商品コード b2: 製造番号 b1: 製造年月日 b0: プロパティサイズマップ 1 がセットされる項目が有効である。
SetM プロパティマップ	17	SetM を受け付けるプロパティを示す。*1
Set プロパティマップ	17	Set を受け付けるプロパティを示す。*1
GetM プロパティマップ	17	GetM を受け付けるプロパティを示す。*1
Get プロパティマップ	17	Get を受け付けるプロパティを示す。*1
状態アナウンスプロパティマップ	17	状態アナウンスを行うプロパティを示す。*1。
IASetup プロパティ動作マップ	17	Set を受け付けるプロパティに対するアダプタ内部サービスを示す。プロパティへの処理は、IASetup サービス処理となる。*1
IAGetup プロパティ動作マップ	17	Get を受け付けるプロパティに対するアダプタ内部サービスを示す。プロパティへの処理は、IAGetup サービス処理となる。*1
IASetMup プロパティ動作マップ	17	SetM を受け付けるプロパティに対するアダプタ内部サービスを示す。プロパティへの処理は、IASetMup サービス処理となる。*1
IAGetMup プロパティ動作マップ	17	GetM を受け付けるプロパティに対するアダプタ内部サービスを示す。プロパティへの処理は、IAGetMup サービス処理となる。*1
Version 情報	4	使用する規格 Version を示す。書式は ECHONET Lite 規格に従う。
メーカーコード	3	ECHONET Lite レディ機器の製造メーカーを示す。書式は ECHONET Lite 規格に従う。
事業所コード	3	ECHONET Lite レディ機器を製造した事業所を示すベンダ依存のコードを示す。書式は ECHONET Lite 規格に従う。
商品コード	12	ベンダ依存の ECHONET Lite レディ機器の商品コードを示す。書式は ECHONET Lite 規格に従う。
製造番号	12	ベンダ依存の ECHONET Lite レディ機器の製造番号を示す。書式は ECHONET Lite 規格に従う。
製造年月日	4	ECHONET Lite レディ機器が製造された年月日を示す。書式は ECHONET Lite 規格に従う。
プロパティサイズマップ	1~128	規定の機器オブジェクトが持つ全てのプロパティに関して、個々のプロパティのサイズをバイト単位で示す。配列プロパティの場合は 1 つの配列要素のサイズと最大配列要素数の積となる。存在するプロパティについて、EPC の小さい順にサイズを列挙する。

注) *1: 書式は ECHONET Lite 規格第 2 部付録 2 記載のプロパティマップ記述形式 (2) での記述方法とする。(サイズは 17 バイト固定)

(2) 機器問合せ完了通知／通知受理応答コマンド (Required)

基本ミドルウェアアダプタは、全ての機器オブジェクトを構築するための情報を受け取った場合、機器問い合わせ完了通知を ECHONET Lite レディ機器に送る。

機器問い合わせ応答で受け取ったデータに異常がある場合は、機器問合せ処理結果として異常ありを返し、エラー停止状態に遷移する。

機器問い合わせ完了通知を受け取った ECHONET Lite レディ機器は、基本ミドルウェアアダプタに機器問い合わせ完了通知受理応答を返す。

機器問合せ処理結果が異常ありである機器問い合わせ完了通知を受け取った場合は、初期化シーケンスのやり直しを行わなければならない。

◇要求コマンドの方向

ECHONET Lite ミドルウェアアダプタ → ECHONET Lite レディ機器

◇要求コマンド

1バイト	2バイト	1バイト	1バイト	2バイト	2バイト	1バイト
STX	FT	CN	FN	DL	FD(0)	FCC

STX : 0x02
 FT : 0x0002
 CN : 0x01
 FN : 0x**
 DL : 0x0002
 FD(0) : 機器問合せ処理結果

FD(0)の値	内容
0x0000	異常なし
0x0011	異常あり

FCC : 0x**

◇応答コマンド

1バイト	2バイト	1バイト	1バイト	2バイト	2バイト	1バイト
STX	FT	CN	FN	DL	FD(0)	FCC

STX : 0x02
 FT : 0x0002
 CN : 0x81
 FN : 要求時の値 (機能を搭載していない場合は0x00)
 DL : 0x0002
 FD(0) : 応答結果

FD(0)の値	内容
0x0000	正常受理
0xFFFF	その他のエラー

FCC : 0x**

(3) アダプタ立ち上げ通知／通知受理応答コマンド (Required)

基本ミドルウェアアダプタは、初期化シーケンスが正常終了し、かつ、オブジェクト構築シーケンスが正常終了したときに、その内部に機器オブジェクトを保持している場合、あるいは機器オブジェクトの構築に成功した場合は、ECHONET Lite レディ機器に対しアダプタ立ち上げ通知を送信する。

なお、アダプタ立ち上げ通知を送る前に、第2部にしたがったノードプロファイルオブジェクトも構築しておかなければならない。

機器オブジェクトの構築に失敗した場合は、初期化処理結果を立ち上げ失敗としたアダプタ立ち上げ通知を送り、エラー停止状態に遷移する。

応答として応答結果が正常受理であるアダプタ立ち上げ通知応答を受け取った場合、通常動作状態に遷移する。

アダプタ立ち上げ通知を受け取った ECHONET Lite レディ機器は、基本ミドルウェアアダプタに対し、アダプタ立ち上げ通知受理応答を返す。

初期化処理結果が立ち上げ失敗であるアダプタ立ち上げ通知を受け取った場合は、初期化シーケンスのやり直しを行わなければならない。

◇要求コマンドの方向

ECHONET Lite ミドルウェアアダプタ → ECHONET Lite レディ機器

◇要求コマンド

1バイト	2バイト	1バイト	1バイト	2バイト	2バイト	1バイト
STX	FT	CN	FN	DL	FD(0)	FCC

STX : 0x02
 FT : 0x0002
 CN : 0x02
 FN : 0x**
 DL : 0x0002
 FD(0) : 初期化処理結果

FD(0)の値	内容
0x0000	立ち上げ成功
0x0011	立ち上げ失敗

FCC : 0x**

◇応答コマンド

1バイト	2バイト	1バイト	1バイト	2バイト	2バイト	1バイト
STX	FT	CN	FN	DL	FD(0)	FCC

STX : 0x02
 FT : 0x0002
 CN : 0x82
 FN : 要求時の値 (機能を搭載していない場合は0x00)
 DL : 0x0002
 FD(0) : 応答結果

FD(0)の値	内容
0x0000	正常受理
0xFFFF	その他のエラー

FCC : 0x**

(4) オブジェクト指定機器問合せ要求/応答コマンド (Optional)

基本ミドルウェアアダプタは、内部に機器オブジェクトが存在しない場合などに、ECHONET Lite レディ機器に対しオブジェクト指定機器問い合わせ要求を行う。この要求に対し、ECHONET Lite レディ機器からは機器の情報がオブジェクト指定機器問い合わせ応答として返される。

オブジェクト指定機器問い合わせ要求を受け取った ECHONET Lite レディ機器は、機器オブジェクトを基本ミドルウェアアダプタに構築するために必要な情報を、オブジェクト指定機器問い合わせ応答として基本ミドルウェアアダプタに渡す。オブジェクト情報の取得管理は ECHONET Lite ミドルウェアアダプタで行う。ECHONET Lite レディ機器は、生成したいオブジェクト数が最大値となるような 0x01 から始まる 1 バイトのオブジェクト識別番号を、オブジェクトコード (EOJ) に付与して管理し、ECHONET Lite ミドルウェアアダプタに回答する。ECHONET Lite ミドルウェアアダプタは、このオブジェクト識別番号を利用してオブジェクトを識別管理する。

注) * : 生成できる機器オブジェクト数が 4 以上のアダプタ、及び、生成要求する機器オブジェクトが 4 以上のレディ機器では Required とする。

◇要求コマンドの方向

ECHONET Lite ミドルウェアアダプタ → ECHONET Lite レディ機器

◇要求コマンド

1バイト	2バイト	1バイト	1バイト	2バイト	1バイト	1バイト
STX	FT	CN	FN	DL	FD(0)	FCC

STX : 0x02
 FT : 0x0002
 CN : 0x03
 FN : 0x**
 DL : 0x0001
 FD(0) : オブジェクト識別番号 (0x01 から順次使用する)
 FCC : 0x**

◇応答コマンド

1バイト	2バイト	2バイト	1バイト	2バイト	2バイト	1バイト	nバイト	1バイト
STX	FT	CN	FN	DL	FD(0)	FD(1)	FD(2)	FCC

STX : 0x02
 FT : 0x0002

CN : 0x83
 FN : 要求時の値 (機能を搭載していない場合は0x00)
 DL : 0x****
 FD(0) : 応答結果

名称	サイズ (byte)	説明
Result	2	応答結果。 0x0000: 受理応答 0x0001: 受理応答(ネットワーク停止中) 0x0011: 拒否応答 0x0012: 指定機器オブジェクト無し 0x0101: 状態不一致エラー(機器インタフェース確認状態(未確認)) 0x0103: 状態不一致エラー(待機状態) 0x0105: 状態不一致エラー(エラー停止状態) 0xFFFF: その他のエラー

FD(1) : フレーム内送信オブジェクト数
 0x01 固定。
 FD(2) : オブジェクトデータ
 オブジェクトデータフォーマットを参照
 FCC : 0x**

<オブジェクトデータフォーマット>

2バイト	3バイト	2バイト	194~289バイト
オブジェクト識別情報	EOJ	機器問い合わせデータ長	機器問い合わせデータ

名称	サイズ (byte)	説明
オブジェクト識別情報	2	1バイト目:オブジェクト識別番号 2バイト目:オブジェクト全数
EOJ	3	構築する機器オブジェクトの ECHONET オブジェクトコード。
機器問い合わせデータ長	2	機器問い合わせデータのサイズを示す。
機器問い合わせデータ	194~289	機器問い合わせデータフォーマットを参照。

<機器問い合わせデータフォーマット>

2バイト	17バイト	17バイト	17バイト	17バイト	17バイト	17バイト
有効無効ビットマップ	SetM プロパティマップ	Set プロパティマップ	GetM プロパティマップ	Get プロパティマップ	状態アナウンスプロパティマップ	IASetup プロパティ動作マップ
17バイト	17バイト	17バイト	4バイト	3バイト	3バイト	12バイト
IAGetup プロパティ動作マップ	IASetMup プロパティ動作マップ	IAGetMup プロパティ動作マップ	Version 情報	メーカーコード	事業所コード	商品コード
12バイト	4バイト	1~128バイト				
製造番号	製造年月日	プロパティサイズマップ				

名称	サイズ (byte)	説明
有効/無効ビットマップ	2	以降に連続するデータの中で意味のあるものを示す。 b15: SetM プロパティマップ b14 : Set プロパティマップ b13 : GetM プロパティマップ b12: Get プロパティマップ b11: 状態アナウンスプロパティマップ b10: IASetup プロパティ動作マップ b9: IAGetup プロパティ動作マップ b8: IASetMup プロパティ動作マップ b7: IAGetMup プロパティ動作マップ b6: Version 情報 b5: メーカーコード b4: 事業所コード b3: 商品コード b2: 製造番号 b1: 製造年月日 b0: プロパティサイズマップ 1 がセットされる項目が有効である。
SetM プロパティマップ	17	SetM を受け付けるプロパティを示す。*1
Set プロパティマップ	17	Set を受け付けるプロパティを示す。*1
GetM プロパティマップ	17	GetM を受け付けるプロパティを示す。*1
Get プロパティマップ	17	Get を受け付けるプロパティを示す。*1
状態アナウンスプロパティマップ	17	状態アナウンスを行うプロパティを示す。*1
IASetup プロパティ動作マップ	17	Set を受け付けるプロパティに対するアダプタ内部サービスを示す。プロパティへの処理は、IASetup サービス処理となる。*1
IAGetup プロパティ動作マップ	17	Get を受け付けるプロパティに対するアダプタ内部サービスを示す。プロパティへの処理は、IAGetup サービス処理となる。*1
IASetMup プロパティ動作マップ	17	SetM を受け付けるプロパティに対するアダプタ内部サービスを示す。プロパティへの処理は、IASetMup サービス処理となる。*1
IAGetMup プロパティ動作マップ	17	GetM を受け付けるプロパティに対するアダプタ内部サービスを示す。プロパティへの処理は、IAGetMup サービス処理となる。*1
Version 情報	4	使用する規格 Version を示す。書式は ECHONET Lite 規格に従う。
メーカーコード	3	ECHONET Lite レディ機器の製造メーカーを示す。書式は ECHONET Lite 規格に従う。
事業所コード	3	ECHONET Lite レディ機器を製造した事業所を示すベンダ依存のコードを示す。書式は ECHONET Lite 規格に従う。
商品コード	12	ベンダ依存の ECHONET Lite レディ機器の商品コードを示す。書式は ECHONET Lite 規格に従う。
製造番号	12	ベンダ依存の ECHONET Lite レディ機器の製造番号を示す。書式は ECHONET Lite 規格に従う。
製造年月日	4	ECHONET Lite レディ機器が製造された年月日を示す。書式は ECHONET Lite 規格に従う。
プロパティサイズマップ	1~128	規定の機器オブジェクトが持つ全てのプロパティに関して個々のプロパティのサイズをバイト単位で示す。配列プロパティの場合は 1 つの配列要素のサイズと最大配列要素数の積となる。存在するプロパティについて、EPC の小さい順にサイズを列挙する。

注) *1: 書式は ECHONET Lite 規格第 2 部付録 2 記載のプロパティマップ記述形式 (2) での記述方法とする。(サイズは 17 バイト固定)

3.8.4.4 ECHONET Lite通信モード

(1) 機器状態アクセス要求/応答コマンド (Required)

基本ミドルウェアアダプタは、アダプタ内部サービスが IASetup、IASetMup であるプロパティに対し Set が行われた場合、あるいはアダプタ内部サービスが IAGetup、IAGetMup であるプロパティに Get が行われた場合は、Set、SetM、あるいは Get、GetM の情報を機器状態アクセス要求により ECHONET Lite レディ機器に伝える。

機器状態アクセス要求を受け取った ECHONET Lite レディ機器は、それが IASetup、IASetMup の場合はそれに対する応答を、IAGetup、IAGetMup の場合は指定されたプロパティに対応する状態値を、機器状態アクセス応答により基本ミドルウェアアダプタに返す。

◇要求コマンドの方向

ECHONET Lite ミドルウェアアダプタ → ECHONET Lite レディ機器

◇要求コマンド

1バイト	2バイト	1バイト	1バイト	2バイト	6+nバイト	1バイト
STX	FT	CN	FN	DL	FD(0)	FCC

STX : 0x02
 FT : 0x0003
 CN : 0x10
 FN : 0x**
 DL : 6+n
 FD(0) : オブジェクト情報

名称	サイズ (byte)	説明
EOJ	3	参照、または変更対象機器の EOJ
Length	2	EPC+EDT のバイト数。これが 0x01 の場合は IAGetup (状態参照)、それ以外の場合は IASetup (状態変更)となる。
EPC	1	参照、または変更対象のプロパティの EPC
EDT	n	この値が存在する場合は、プロパティの変更となり、ECHONET Lite レディ機器の EPC に対応する状態が変更される (MAX 245byte)。この値が存在しない場合は、状態参照となる。

FCC : 0x**

◇応答コマンド

1バイト	2バイト	1バイト	1バイト	2バイト	8+nバイト	1バイト
STX	FT	CN	FN	DL	FD(0)	FCC

STX : 0x02
 FT : 0x0003
 CN : 0x90
 FN : 要求時の値 (機能を搭載していない場合は 0x00)
 DL : 8+n

FD(0) : オブジェクト情報

名称	サイズ (byte)	説明
EOJ	3	参照、または変更対象 ECHONET Lite レディ機器の EOJ。
Result	2	応答結果。 0x0000: 受理応答 0x0011: 拒否応答 0xFFFF: その他のエラー
Length	2	EPC+EDT のバイト数。これが 0x01 の場合は変更応答、それ以外の場合は参照応答となる。
EPC	1	参照、または変更対象のプロパティの EPC。
EDT	n	この値が存在する場合は、これが参照応答値となる (MAX 245byte)。

FCC : 0x**

(2) 機器状態通知要求/応答コマンド (Required)

基本ミドルウェアアダプタは、機器状態通知を受け取った場合、機器状態通知応答を ECHONET Lite レディ機器に返し、通知対象プロパティのアダプタ内部サービスが IAGet、IAGetM の場合は、内部に保持している機器オブジェクトの対象プロパティに通知する値を書き込むとともに、その値をドメイン内にアナウンスする。通知対象プロパティのアダプタ内部サービスが IAGetup、IAGetMup の場合は、機器状態通知応答を ECHONET Lite レディ機器に返し、通知するべき値をドメイン内にアナウンスする。

◇要求コマンドの方向

ECHONET Lite レディ機器 → ECHONET Lite ミドルウェアアダプタ

◇要求コマンド

1バイト	2バイト	1バイト	1バイト	2バイト	6+nバイト	1バイト
STX	FT	CN	FN	DL	FD(0)	FCC

STX : 0x02
 FT : 0x0003
 CN : 0x11
 FN : 0x** (機能を搭載していない場合は 0x00)
 DL : 6+n
 FD(0) : オブジェクト情報

名称	サイズ (byte)	説明
EOJ	3	通知対象 ECHONET Lite レディ機器の EOJ。
Length	2	EPC+EDT のバイト数。
EPC	1	通知対象のプロパティの EPC。
EDT	n	通知データ (MAX 245byte)。

FCC : 0x**

◇応答コマンド

1バイト	2バイト	1バイト	1バイト	2バイト	2バイト	3バイト	1バイト
STX	FT	CN	FN	DL	FD(0)	FD(1)	FCC

STX : 0x02
 FT : 0x0003
 CN : 0x91
 FN : 要求時の値
 DL : 0x0005
 FD(0) : 処理結果

名称	サイズ (byte)	説明
Result	2	応答結果。 0x0000: 受理応答 0x0011: 拒否応答(ネットワーク停止中) 0x0012: 拒否応答(その他) 0x0101: 状態不一致エラー(機器インタフェース確認状態(未確認)) 0x0103: 状態不一致エラー(待機状態) 0x0104: 状態不一致エラー(オブジェクト構築状態) 0x0105: 状態不一致エラー(エラー停止状態) 0xFFFF: その他のエラー(EOJ,EPC が存在しなかった場合等)

FD(1) : 通知対象 ECHONET Lite レディ機器のEOJ
 FCC : 0x**

(3) 要素指定機器状態アクセス要求/応答コマンド (Required)

基本ミドルウェアアダプタは、アダプタ内部サービスが IASetMup であるプロパティに対し SetM が行われた場合、あるいはアダプタ内部サービスが IAGetMup であるプロパティに GetM が行われた場合は、SetM、あるいは GetM の情報を要素指定機器状態アクセス要求により機器に伝える。

要素指定機器状態アクセス要求を受け取った ECHONET Lite レディ機器は、それが IASetMup の場合はそれに対する応答を、IAGetMup の場合は指定されたプロパティの要素番号に対応する状態値を、要素指定機器状態アクセス応答により基本ミドルウェアアダプタに返す。

◇要求コマンドの方向

ECHONET Lite ミドルウェアアダプタ → ECHONET Lite レディ機器

◇要求コマンド

1バイト	2バイト	1バイト	1バイト	2バイト	8+nバイト	1バイト
STX	FT	CN	FN	DL	FD(0)	FCC

STX : 0x02

FT : 0x0003
 CN : 0x12
 FN : 0x**
 DL : 8+n
 FD(0) : ミドルウェアアダプタ種別

名称	サイズ (byte)	説明
EOJ	3	参照、または変更対象 ECHONET Lite レディ機器の EOJ
Length	2	EPC+EDT+要素番号のバイト数。これが 0x03 の場合は参照、それ以外の場合は変更となる。
EPC	1	参照、または変更対象のプロパティの EPC
要素番号	2	参照、または変更対象の配列の要素番号。
EDT	n	この値が存在する場合は、プロパティの変更となり、ECHONET Lite レディ機器の EPC に対応する状態が変更される (MAX 245byte)。この値が存在しない場合は、状態参照となる。

FCC : 0x**

◇応答コマンド

1バイト	2バイト	1バイト	1バイト	2バイト	10+nバイト	1バイト
STX	FT	CN	FN	DL	FD(0)	FCC

STX : 0x02
 FT : 0x0003
 CN : 0x92
 FN : 要求時の値 (機能を搭載していない場合は 0x00)
 DL : 10+n
 FD(0) : オブジェクト情報

名称	サイズ (byte)	説明
EOJ	3	参照、または変更対象 ECHONET Lite レディ機器の EOJ。
Result	2	応答結果。 0x0000: 受理応答 0x0011: 拒否応答 0xFFFF: その他のエラー (EOJ, EPC が存在しなかった場合等)
Length	2	EPC+EDT+要素番号のバイト数。これが 0x01 の場合は変更応答、それ以外の場合は参照応答となる。
EPC	1	参照、または変更対象のプロパティの EPC。
要素番号	2	参照、または変更対象の配列の要素番号。
EDT	n	この値が存在する場合は、これが参照応答値となる。

FCC : 0x**

(4) 要素指定機器状態通知要求/応答コマンド (Required)

基本ミドルウェアアダプタは、要素指定機器状態通知を受け取った場合、要素指定機器状態通知応答を ECHONET Lite レディ機器に返し、通知対象プロパティのアダプタ内部サー

ビスが IAGetM の場合は、内部に保持している機器オブジェクトの対象プロパティの要素番号に通知する値を書き込むとともに、その値をドメイン内にアナウンスする。通知対象プロパティのアダプタ内部サービスが IAGetMup の場合は、要素指定機器状態通知応答を ECHONET Lite レディ機器に返し、プロパティの要素番号で指定される値をドメイン内にアナウンスする。

◇要求コマンドの方向

ECHONET Lite レディ機器 → ECHONET Lite ミドルウェアアダプタ

◇要求コマンド

1バイト	2バイト	1バイト	1バイト	2バイト	8+nバイト	1バイト
STX	FT	CN	FN	DL	FD(0)	FCC

STX : 0x02
 FT : 0x0003
 CN : 0x13
 FN : 0x** (機能を搭載していない場合は0x00)
 DL : 8+n
 FD(0) : オブジェクト情報

名称	サイズ (byte)	説明
EOJ	3	通知対象 ECHONET Lite レディ機器の EOJ。
Length	2	EPC+EDT+要素番号データのバイト数。
EPC	1	通知対象のプロパティの EPC。
要素番号	2	通知対象の要素番号。
EDT	n	通知データ(MAX 245byte)。

FCC : 0x**

◇応答コマンド

1バイト	2バイト	1バイト	1バイト	2バイト	2バイト	3バイト	1バイト
STX	FT	CN	FN	DL	FD(0)	FD(1)	FCC

STX : 0x02
 FT : 0x0003
 CN : 0x93
 FN : 要求時の値
 DL : 0x0005
 FD(0) : 応答結果

名称	サイズ (byte)	説明
Result	2	応答結果。 0x0000: 受理応答 0x0011: 拒否応答(ネットワーク停止中) 0x0012: 拒否応答(その他) 0x0101: 状態不一致エラー(機器インタフェース確認状)

		態(未確認) 0x0103: 状態不一致エラー(待機状態) 0x0104: 状態不一致エラー(オブジェクト構築状態) 0x0105: 状態不一致エラー(エラー停止状態) 0xFFFF: その他のエラー(EQJ,EPC が存在しなかった場合等)
--	--	---

FD(1) : 通知対象 ECHONET Lite レディ機器のEOJ
 FCC : 0x**

(5) オブジェクトアクセス要求/応答コマンド (Required)

アダプタ内部サービスが IASet 或いは IAGet となっているプロパティに対し、書き込み指定のオブジェクトアクセス要求が行われた場合、基本ミドルウェアアダプタは指定されたプロパティの値を対応するプロパティに書き込む。アダプタ内部サービスが IAGet 或いは IASet となっているプロパティに対し、読み出し指定のオブジェクトアクセス要求が行われた場合、基本ミドルウェアアダプタは指定されたプロパティの値をオブジェクトアクセス応答として ECHONET Lite レディ機器に返す。IASetup 且つ IAGetup が指定されているプロパティ、或いは、IASetMup 且つ IAGetMup が指定されているプロパティに対しオブジェクトアクセス要求が行われた場合は、Result を拒否応答としたオブジェクトアクセス応答を ECHONET Lite レディ機器に返す。

◇要求コマンドの方向

ECHONET Lite レディ機器 → ECHONET Lite ミドルウェアアダプタ

◇要求コマンド

1バイト	2バイト	1バイト	1バイト	2バイト	6+nバイト	1バイト
STX	FT	CN	FN	DL	FD(0)	FCC

STX : 0x02
 FT : 0x0003
 CN : 0x14
 FN : 0x** (機能を搭載していない場合は0x00)
 DL : 6+n
 FD(0) : オブジェクト情報

名称	サイズ (byte)	説明
EOJ	3	参照、または変更対象 ECHONET Lite レディ機器の EOJ
Length	2	EPC+EPC データのバイト数。これが 0x01 の場合は参照、それ以外の場合は変更となる。
EPC	1	参照、または変更対象のプロパティの EPC
EDT	n	この値が存在する場合は、プロパティの変更となり、ECHONET Lite レディ機器の EPC に対応する状態が変更される (MAX 245byte)。この値が存在しない場合は、状態参照となる。

FCC : 0x**

◇応答コマンド

1バイト	2バイト	1バイト	1バイト	2バイト	2バイト	6+nバイト	1バイト
STX	FT	CN	FN	DL	FD(0)	FD(0)	FCC

STX : 0x02
 FT : 0x0003

 CN : 0x94
 FN : 要求時の値
 DL : 8+n
 FD(0) : 応答結果

名称	サイズ (byte)	説明
Result	2	応答結果。 0x0000: 受理応答 0x0001: 受理応答(ネットワーク停止中) 0x0011: 拒否応答 0x0101: 状態不一致エラー(機器インタフェース確認状態(未確認)) 0x0103: 状態不一致エラー(待機状態) 0x0104: 状態不一致エラー(オブジェクト構築状態) 0x0105: 状態不一致エラー(エラー停止状態) 0xFFFF: その他のエラー(EOJ,EPC が存在しなかった場合等)

FD(1) : オブジェクト情報

名称	サイズ (byte)	説明
EOJ	3	参照、または変更対象 ECHONET Lite レディ機器のEOJ。
Length	2	EPC+EPC データのバイト数。これが 0x01 の場合は変更応答、それ以外の場合は参照応答となる。
EPC	1	参照、または変更対象のプロパティの EPC。
EDT	n	この値が存在する場合は、これが参照応答値となる(MAX 245byte)。

FCC : 0x**

(6) 機器状態アクセス一括要求/応答コマンド (Optional)

ECHONET Lite ミドルウェアアダプタが、以下の場合に、ECHONET Lite レディ機器への値の読み出しや値の設定を、それぞれのサービスが指定されているプロパティを一括して行う為のコマンド。

- IAGet、IAGetM サービスが指定されているプロパティに対して、ECHONET Lite レディ機器が保持するプロパティの値の読み出しを、一括で行う場合
- IASet、IASetM サービスが指定されているプロパティに対して、ECHONET Lite ミドルウェアアダプタが保持するプロパティの値の読み出しを、一括で行う場合
- IAGet、IAGetM サービスが指定されているプロパティに対して、ECHONET Lite ミドルウェアアダプタが保持するプロパティの値の機器への通知を、一括で行う場合
- IASet、IASetM サービスが指定されているプロパティに対して、ECHONET Lite ミドルウェアアダプタが保持するプロパティの値の機器への通知を、一括で行う場合

機器状態アクセス一括要求を受け取った ECHONET Lite レディ機器は、それが読み出しの要求なのか通知の要求なのかにより、応答電文を構成して、機器状態アクセス応答により基本ミドルウェアアダプタに返す。

◇要求コマンドの方向

ECHONET Lite ミドルウェアアダプタ → ECHONET Lite レディ機器

◇要求コマンド

1バイト	2バイト	1バイト	1バイト	2バイト	1バイト	3バイト	nバイト	1バイト
STX	FT	CN	FN	DL	FD(0)	FD(1)	FD(2)	FCC

- STX : 0x02
 FT : 0x0003
 CN : 0x20
 FN : 0x00~0xFF
 DL : FD(0)~FD(2)全体のサイズをHEX表示した値
 FD(0) : アクセス要求内容指定
 0x00 : IAGet、IAGetM サービス指定プロパティ読み出し要求
 0x01 : IAGet、IAGetM サービス指定プロパティ通知要求
 0x02 : IASet、IASetM サービス指定プロパティ読み出し要求
 0x03 : IASet、IASetM サービス指定プロパティ通知要求
 0x02~0xFF : for future reserved
 FD(1) : オブジェクトコード(EOJ)
 FD(2) : FD(0)が0x01及び0x03の時に、存在する。
 FD(0)が、0x01の時には、
 「Version 情報」「メーカーコード」「事業所コード」「商品コード」「製造番号」「製造年月日」プロパティ及び各種プロパティマッププロパティを除く、IAGet/IAGetMとしてミドルウェアアダプタにて保持しているプロパティ値を、プロパティコードの小さい順に列挙する。
 FD(0)が、0x03の時には、
 「Version 情報」「メーカーコード」「事業所コード」「商品コード」「製造番号」「製造年月日」プロパティ及び各種プロパティマッププロパティを除く、IASet/IASetMとしてミドルウェアアダプタにて保持しているプロパティ値を、プロパティコードの小さい順に列挙する。
 FCC : 0x**

◇応答コマンド

1バイト	2バイト	1バイト	1バイト	2バイト	2バイト	nバイト	1バイト
STX	FT	CN	FN	DL	FD(0)	FD(1)	FCC

- STX : 0x02
- FT : 0x0003
- CN : 0xA0
- FN : 要求時の値 (機能を搭載していない場合は0x00)
- DL : FD(0)~FD(1)全体のサイズをHEX表示した値
- FD(0) : 処理結果

FD(0)の値	内容	備考
0x0000	要求受理 (正常)	
0x0011	要求不受理 (要求指定対象オブジェクト無)	
0xFF FF	要求不受理 (上記以外の理由)	
上記以外	for future reserved	

FD(1) : 要求電文のFD(0)で、0x00及び0x02 (読み出し要求) が指定された時で、応答電文のFD(0)が要求受理 (0x0000) の時に、存在する。

要求電文のFD(0)が、0x00の時には、

「Version 情報」「メーカーコード」「事業所コード」「商品コード」「製造番号」「製造年月日」プロパティ及び各種プロパティマッププロパティを除く、IAGet/IAGetMとしてレディ機器にて保持しているプロパティ値を、プロパティコードの小さい順に列挙する。

要求電文のFD(0)が、0x02の時には、

「Version 情報」「メーカーコード」「事業所コード」「商品コード」「製造番号」「製造年月日」プロパティ及び各種プロパティマッププロパティを除く、IASet/IASetMとしてレディ機器にて保持しているプロパティ値を、プロパティコードの小さい順に列挙する。

FCC : 0x**

(7) 機器状態アクセス一括UP 要求/応答コマンド (Optional)

複合電文対応の機能を持つ ECHONET Lite ミドルウェアアダプタにおいて、以下の場合に、ECHONET Lite レディ機器への値の読み出しや値の設定を、複合電文で指定されたプロパティに対して一括して行う為のコマンド。

- ・IAGetup、IAGetMup サービスが指定されているプロパティに対して、他のノードから読み出し要求があり、ECHONET Lite レディ機器が保持する対象のプロパティの値の読み出しを行う場合

- ・IASetup サービスが指定されているプロパティに対して、他のノードから書き込み要求があり、ECHONET Lite レディ機器へ要求のあった値の通知を行う場合

機器状態アクセス一括UP 要求を受け取った ECHONET Lite レディ機器は、それが読み出しの要求なのか書き込みの要求なのかにより、応答電文を構成して、機器状態アクセス一括UP 応答により基本ミドルウェアアダプタに返す。

ミドルウェアアダプタは、本コマンドの応答としてコマンドエラーが指定された通信エラーコマンドを受信した場合には、機器状態アクセス要求コマンドを用いて複合電文で指定されたプロパティに関して個別にて要求を行うものとする。

◇要求コマンドの方向

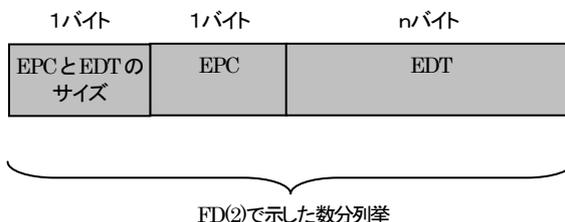
ECHONET Lite ミドルウェアアダプタ → ECHONET Lite レディ機器

◇要求コマンド

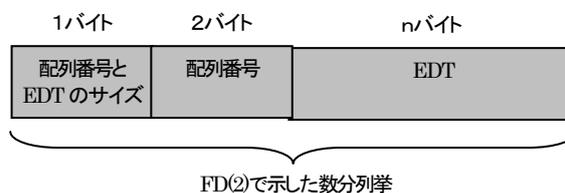
1バイト	2バイト	1バイト	1バイト	2バイト	1バイト	3バイト	2バイト	1~256バイト	1バイト
STX	FT	CN	FN	DL	FD(0)	FD(1)	FD(2)	FD(3)	FCC

- STX : 0x02
 FT : 0x0003
 CN : 0x21
 FN : 0x00~0xFF
 DL : FD(0)~FD(3)全体のサイズをHEX表示した値
 FD(0) : アクセス要求内容指定
 0x00 : IAGetup, IAGetMup サービス指定プロパティ値読み出し要求
 0x01 : IASetup, IASetMup サービス指定プロパティ値書き込み要求
 0x02~0xFF : for future reserved
 FD(1) : オブジェクトコード(EOJ)
 FD(2) : 2バイト。EPC数、EPCコード+配列要素数。
 指定プロパティが非配列型の場合
 ・1バイト目は、0x00固定。2バイト目でEPC数。
 指定プロパティが配列型の場合
 ・1バイト目は、EPCコード値(0x80~0xFF)。2バイト目は配列要素数。
 FD(3) : FD(0)が0x00の場合と0x01の場合で構成が異なる。
 FD(0)が0x00の場合で、FD(2)の1バイト目が0x00の場合には、IAGetup要求のあるEPCのコードを列挙する。列挙するEPCの数は、FD(2)の2バイト目で指定。
 FD(0)が0x00の場合で、FD(2)の1バイト目が0x00以外(指定プロパティが配列型)の場合には、IAGetup要求のある配列要素番号を列挙する。列挙する配列要素の数は、FD(2)の2バイト目で指定。
 FD(0)が0x01の場合には、指定プロパティが非配列か配列かにより、以下とする。

FD(0)が0x01の場合で、指定プロパティが非配列型の場合、FD(2)で示された数だけEPC毎の下記構成を列挙する。



FD(0)が0x01の場合で、指定プロパティが配列型の場合、FD(2)で示した配列要素数分、下記構成を列挙する。



FCC : 0x**

◇応答コマンド

1バイト	2バイト	1バイト	1バイト	2バイト	2バイト	2バイト	nバイト	1バイト
STX	FT	CN	FN	DL	FD(0)	FD(1)	FD(2)	FCC

- STX : 0x02
 FT : 0x0003
 CN : 0xA1
 FN : 要求時の値 (機能を搭載していない場合は0x00)
 DL : FD(0)~FD(2)全体のサイズをHEX表示した値

FD(0) : 処理結果

FD(0)の値	内容	備考
0x0000	要求受理 (正常)	
0x0011	要求不受理 (要求指定対象オブジェクト無)	
0xFF F F F	要求不受理 (上記以外の理由)	
上記以外	for future reserved	

FD(1) : 応答電文のFD(0)が要求受理 (0x0000) の時に存在する。
 2バイト。EPC 数, EPC コード+配列要素数。

指定プロパティが非配列型の場合

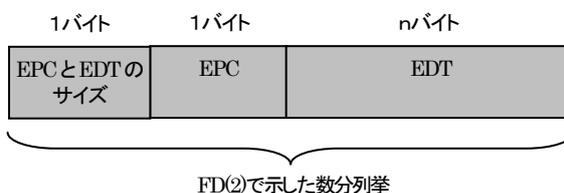
- ・1バイト目は、0x00 固定。2バイト目で EPC 数。

指定プロパティが配列型の場合

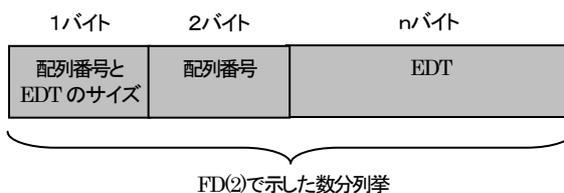
- ・1バイト目は、EPC コード値 (0x00 以外)。2バイト目は配列要素数。

FD(2) : 応答電文のFD(0)が要求受理 (0x0000) で、要求電文のFD(0)が 0x00 の場合に存在する。

指定プロパティが非配列型の場合、FD(2)で示された数だけ EPC 毎の下記構成を列挙する。



プロパティが配列型の場合、FD(2)で示した配列要素集分、下記構成を列挙する。



FCC : 0x**

(8) 機器状態通知一括要求/応答コマンド (Optional)

ECHONET Lite レディ機器が、以下の場合に、ECHONET Lite ミドルウェアアダプタへの値の通知を、それぞれのサービスが指定されているプロパティを一括して行う為のコマンド。

- ・IAGet、IAGetM サービスが指定されているプロパティの値の ECHONET Lite 上への通知指定を、一括で行う場合
- ・IASet、IASetM サービスが指定されているプロパティの値の ECHONET Lite 上への通知指定を、一括で行う場合
- ・IAGetup、IAGetMup サービスが指定されているプロパティの値の ECHONET Lite 上への通知指定を、一括で行う場合
- ・IASetup、IASetMup サービスが指定されているプロパティの値の ECHONET Lite 上への通知指定を、一括で行う場合

基本ミドルウェアアダプタは、機器状態通知一括要求を ECHONET Lite レディ機器から受け取った場合、機器状態通知一括応答を ECHONET Lite レディ機器に返し、通知対象プロパティをドメイン内にアナウンスする。この時、ECHONET Lite ミドルウェアアダプタでは、複合電文に対応した機能を持つ場合には、複合電文を形成して送信する。

ECHONET Lite ミドルウェアアダプタでは、IAGet、IAGet r M、IASet、IASetM 指

定のプロパティに関しては、受け取った値で保持している内容を更新する。

◇要求コマンドの方向

ECHONET Lite レディ機器 → ECHONET Lite ミドルウェアアダプタ

◇要求コマンド

1バイト	2バイト	1バイト	1バイト	2バイト	1バイト	3バイト	nバイト	1バイト	mバイト	1バイト
STX	FT	CN	FN	DL	FD(0)	FD(1)	FD(2)	FD(3)	FD(4)	FCC

- STX : 0x02
- FT : 0x0003
- CN : 0x22
- FN : 0x** (機能を搭載していない場合は0x00)
- DL : FD(0)~FD(4)全体のサイズをHEX表示した値
- FD(0) : アクセス要求内容指定
 - 0x00 : IAGet、IAGetM サービス指定プロパティ通知要求
 - 0x01 : IASet、IASetM サービス指定プロパティ通知要求
 - 0x02 : IAGetup、IAGetMup サービス指定プロパティ通知要求
 - 0x03 : IASetup、IASetMup サービス指定プロパティ通知要求
 - 0x04~0xFF : for future reserved
- FD(1) : オブジェクトコード(EOJ)
- FD(2) : 「Version 情報」「メーカーコード」「事業所コード」「商品コード」「製造番号」「製造年月日」プロパティ及び各種プロパティマッププロパティを除くFD(0)で指定された対象のプロパティの値を、通知の有無に関わらずプロパティコードの小さい順に列挙する。
- FD(3) : FD(2)で指定したプロパティの内、外部へ通知するプロパティの数 (FD(4)で列挙するプロパティ数) の値
- FD(4) : FD(2)で指定したプロパティの内、外部へ通知するプロパティコードを列挙する。
- FCC : 0x**

◇応答コマンド

1バイト	2バイト	1バイト	1バイト	2バイト	2バイト	1バイト
STX	FT	CN	FN	DL	FD(0)	FCC

- STX : 0x02
- FT : 0x0003
- CN : 0xA2
- FN : 要求時の値
- DL : 0x0002
- FD(0) : 処理結果 (Result)

名称	サイズ (byte)	説明
Result	2	応答結果。 0x0000: 受理応答 0x0011: 拒否応答(ネットワーク停止中) 0x0012: 拒否応答(その他) 0x0101: 状態不一致エラー(機器インタフェース確認状態(未確認)) 0x0103: 状態不一致エラー(待機状態) 0x0104: 状態不一致エラー(オブジェクト構築状態) 0x0105: 状態不一致エラー(エラー停止状態) 0xFFFF: その他のエラー

FCC : 0x**

(9) オブジェクトアクセス一括要求/応答コマンド (Optional)

ECHONET Lite レディ機器が、以下の場合に、ECHONET Lite ミドルウェアアダプタへの値の読み出しや値の設定を、それぞれのサービスが指定されているプロパティを一括して行う為のコマンド。

- ・ IASet、IASetM サービスが指定されている、ECHONET Lite ミドルウェアアダプタが保持するプロパティの値の読み出しを行う場合
- ・ IAGet、IAGetM サービスが指定されているプロパティに対して、ECHONET Lite レディ機器が ECHONET Lite ミドルウェアアダプタへプロパティの値の書き込みを行う場合

オブジェクトアクセス一括要求を受け取った ECHONET Lite ミドルウェアアダプタは、それが読み出しの要求なのか書き込みの要求なのかにより、応答電文を構成して、オブジェクトアクセス一括応答により ECHONET Lite レディ機器へ返す。

◇要求コマンドの方向

ECHONET Lite レディ機器 → ECHONET Lite ミドルウェアアダプタ

◇要求コマンド

1バイト	2バイト	1バイト	1バイト	2バイト	1バイト	3バイト	nバイト	1バイト
STX	FT	CN	FN	DL	FD(0)	FD(1)	FD(2)	FCC

- STX : 0x02
 FT : 0x0003
 CN : 0x23
 FN : 0x** (機能を搭載していない場合は0x00)
 DL : FD(0)~FD(2)全体のサイズをHEX表示した値
 FD(0) : アクセス要求内容指定
 0x00 : IASet, IASetM サービス指定プロパティ値読み出し要求
 0x01 : IAGet, IAGetM サービス指定プロパティ値書き込み要求
 0x02~0xFF : for future reserved
 FD(1) : オブジェクトコード(E0J)
 FD(2) : FD(0)が0x01の時に、存在する。
 「Version 情報」「メーカーコード」「事業所コード」「商品コード」「製造番号」「製造年月日」プロパティ及び各種プロパティマッププロパティを除く、IAGet/IAGetMとしてレディ機器にて保持しているプロパティ値を、プロパティコードの小さい順に列挙する。
 FCC : 0x**

◇応答コマンド

1バイト	2バイト	1バイト	1バイト	2バイト	2バイト	nバイト	1バイト	mバイト	1バイト
STX	FT	CN	FN	DL	FD(0)	FD(1)	FD(2)	FD(3)	FCC

- STX : 0x02
 FT : 0x0003
 CN : 0xA3
 FN : 要求時の値
 DL : FD(0)~FD(3)全体のサイズをHEX表示した値
 FD(0) : 処理結果

名称	サイズ (byte)	説明
Result	2	応答結果。 0x0000: 受理応答 0x0001: 受理応答(ネットワーク停止中) 0x0011: 拒否応答 0x0101: 状態不一致エラー(機器インタフェース確認状態(未確認)) 0x0103: 状態不一致エラー(待機状態) 0x0104: 状態不一致エラー(オブジェクト構築状態) 0x0105: 状態不一致エラー(エラー停止状態) 0xFFFF: その他のエラー

FD(1) : 応答電文のFD(0)が受理応答 (0x0000 或いは 0x0001) の場合で、要求電文のFD(0)で、0x00 が指定された時に、存在する。

「Version 情報」「メーカーコード」「事業所コード」「商品コード」「製造番号」「製造年月日」プロパティ及び各種プロパティマッププロパティを除く、IASet/IASetM としてレディ機器にて保持しているプロパティ値を、プロパティコードの小さい順に列挙する。

FD(2) : 応答電文のFD(1)が存在する時に存在する。FD(1)で指定したプロパティの内、FD(3)で列挙するプロパティの数を示す。

FD(3) : 応答電文のFD(1)が存在する時に存在する。FD(1)で指定したプロパティの内、前回アクセス要求されてから新たに外部から書き込みのあったプロパティコードを列挙する。

FCC : 0x**

3.8.4.5 通信エラー通知コマンド (Required)

ミドルウェアアダプタ、および ECHONET Lite レディ機器において、フレーム受信時に下表で定義されている通信エラーが発生した場合は、通信エラー通知フレームを応答データとして通知する必要がある。

◇要求コマンドの方向

ECHONET Lite ミドルウェアアダプタ ⇔ ECHONET Lite レディ機器

◇要求コマンド

1バイト	2バイト	1バイト	1バイト	2バイト	1バイト
STX	FT	CN	FN	DL	FCC

STX : 0x02
 FT : 0x00FF
 CN : エラー番号

エラー名称	Command No	エラー内容
FCC エラー	0x00	正常にフレームを受信したが、FCC が異常である。
コマンドエラー	0x01	該当する Command No が存在しない。
応答結果エラー	0x02	受信した応答フレームの応答結果が想定外の値である。
フレーム内エラー	0x03	正常にフレームを受信したが、フォーマットが異常である。
その他のエラー	0xFF	上記以外のフレーム受信エラー

FN : 0x**
 DL : 0x0000
 FCC : 0x**

3. 8. 5 オブジェクト生成タイプ用通信シーケンス

本項では、各 ECHONET Lite ミドルウェアアダプタ通信インタフェースのオブジェクト生成タイプで用いる通信シーケンスを規定する。以下通信シーケンス毎に詳細を示す。

表3-10 オブジェクト生成タイプ用通信シーケンス一覧

No.	シーケンス名	概要	備考
1	情報確認シーケンス	ECHONET Lite ミドルウェアアダプタが、ECHONET Lite レディ機器に対して、インタフェース方式およびオブジェクト内容を確認するためのシーケンス	
2	初期化シーケンス	ECHONET Lite レディ機器が、ECHONET Lite ミドルウェアアダプタに対して初期化を行うためのシーケンス	
3	オブジェクト構築シーケンス	ECHONET Lite ミドルウェアアダプタが ECHONET Lite レディ機器からオブジェクト情報を取得し、内部にオブジェクトを生成するためのシーケンス	
4	機器状態アクセス要求シーケンス	ECHONET Lite ミドルウェアアダプタが、ECHONET Lite レディ機器内にプロパティを持つものに対して Set/Get 要求を伝えるためのシーケンス	
5	機器状態通知要求シーケンス	ECHONET Lite レディ機器が、ECHONET Lite ミドルウェアアダプタ内部にプロパティを持つものに対して値の書き込みおよび外部への通知をするためのシーケンス	
6	要素指定機器状態アクセス要求シーケンス	ECHONET Lite ミドルウェアアダプタが、ECHONET Lite レディ機器内にプロパティを持つものに対して Set/Get 要求を伝えるためのシーケンス (要素指定)	
7	要素指定機器状態通知要求シーケンス	ECHONET Lite レディ機器が、ECHONET Lite ミドルウェアアダプタ内部にプロパティを持つものに対して値の書き込みおよび外部への通知をするためのシーケンス (要素指定)	
8	オブジェクトアクセス要求シーケンス	ECHONET Lite ミドルウェアアダプタが内部に持つプロパティへ ECHONET Lite レディ機器からアクセスを行うためのシーケンス	

3.8.5.1 機器インタフェース確認モード

(1) 情報確認シーケンス (Required)

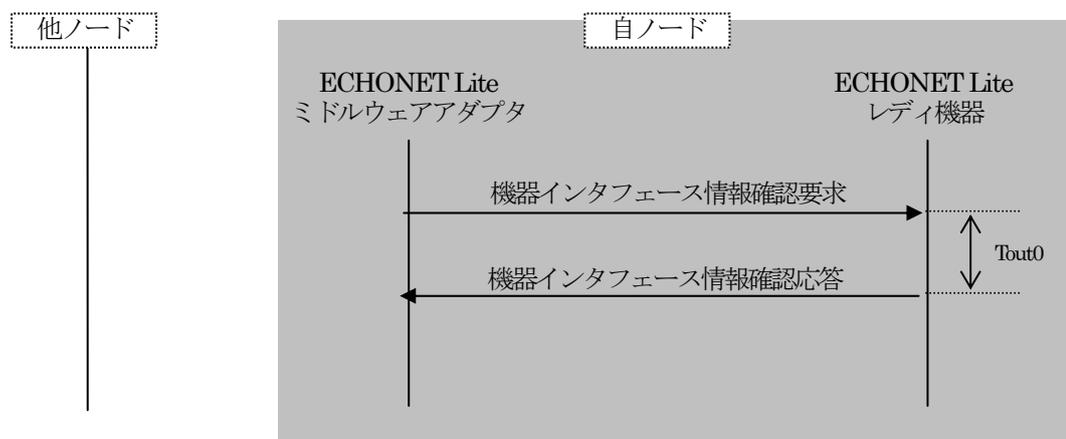


図3-21 情報確認シーケンス

【機器側機器インタフェース情報確認シーケンス】

ECHONET Lite レディ機器は機器インタフェース確認状態のとき、機器インタフェース情報確認フレーム以外を受け付けない。機器インタフェース情報確認要求を受け取った場合、アダプタ種別とオブジェクトを確認し、**Tout30** 以内に機器インタフェース情報確認応答を返す。アダプタ種別の不一致の場合、あるいは機器オブジェクト情報の不一致の場合はエラーを処理結果にセットし、回答する。

アダプタ種別の不一致の場合は、**Tout40** 経過後、アダプタへのリセット信号を出力して、機器インタフェース確認要求を待つ。3回繰り返して、アダプタ種別不一致のエラーが継続している場合は、機器インタフェース情報破棄を処理結果にセットし、機器インタフェース情報確認応答を返し、**Tout50** 経過後、機器インタフェースに関する情報を破棄し、リセット信号出力して、未認識状態に遷移する。

フレームエラーなど全ての通信エラーを無視し、電源投入後あるいはアダプタへのリセット信号出力後 **Tout50** 以上経過しても、機器インタフェース情報確認要求がこない場合は、アダプタへのリセット信号を出力して、機器インタフェース確認要求を待つ。3回繰り返して、要求無しのエラーが継続している場合は、機器インタフェース情報を破棄し、アダプタへのリセット信号を出力して、未認識状態に遷移する。

ECHONET Lite レディ機器が未認識状態のときは、機器インタフェース情報認識サービス仕様の動作にもとづく。

【アダプタ側機器インタフェース情報確認シーケンス】

基本ミドルウェアアダプタは、電源投入後、保持している機器アダプタインタフェースの通信方式とオブジェクトを確認するため ECHONET Lite レディ機器に機器インタフェース情報確認要求を送信する。オブジェクトが存在しない場合は、オブジェクト数0で問い合わせる。

機器インタフェース情報確認応答の処理結果により以下の処理を行う。

- ・ 正常終了で帰ってきた場合は、待機状態に遷移する。
- ・ アダプタ種別の不一致の場合は、待機状態に遷移する。
- ・ オブジェクト不一致の場合は、機器オブジェクトに関する情報を破棄した後、待機状態へ遷移する。
- ・ 機器インタフェース情報破棄の場合は、Tout50 以内に、機器インタフェースと機器オブジェクトに関する情報を破棄して、未認識状態に遷移する。

フレームエラーなど全ての通信エラーを無視して、Tout61 内に応答が帰ってこない場合は、再度、機器インタフェース情報確認要求を送信する。

応答が帰ってこない場合は、機器インタフェースと機器オブジェクトに関する情報を破棄して、未認識状態に遷移する。

3.8.5.2 アダプタ初期化モード

(1) 初期化シーケンス (Required)

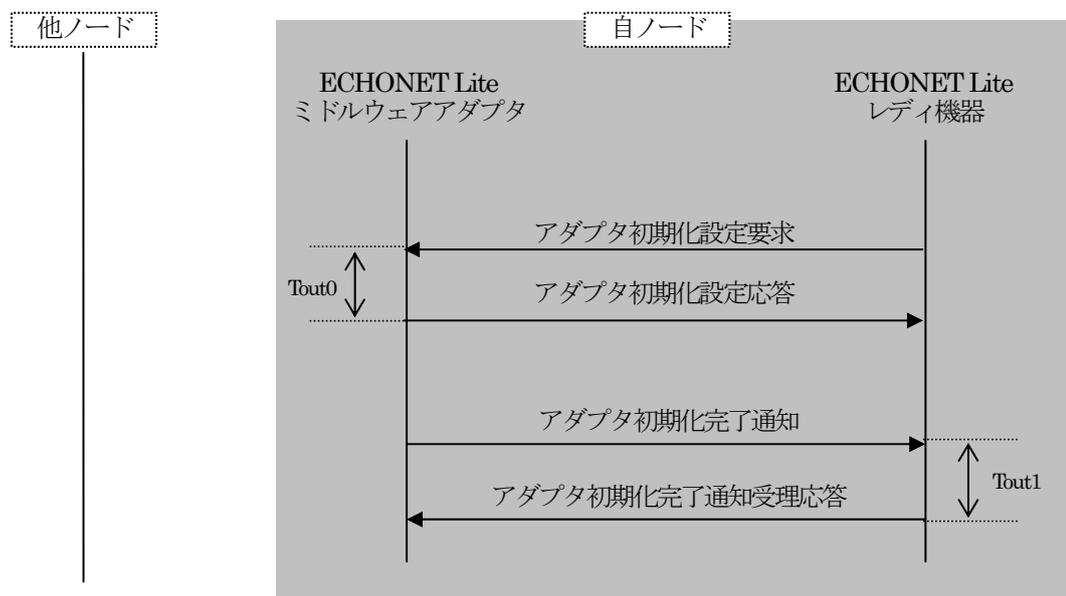


図3-22 初期化シーケンス

【機器側初期化シーケンス】

ECHONET Lite レディ機器はオブジェクト構築状態移行後、アダプタ初期化設定要求を基本ミドルウェアアダプタに要求する。

Tout0 内に初期化受理応答がない場合は、1 回のみ再送を行う。再送後も初期化設定応答がない場合は、ECHONET Lite レディ機器はスタンダロンモードで動作を行う。

初期化設定応答を受け取った ECHONET Lite レディ機器は、初期化設定完了通知を少なくとも Tout11 期間待つ。Tout11 経過後も初期化設定完了通知を受信しなかった場合は、初期化シーケンスを終了し、スタンダロン動作する。

初期化設定完了通知を受け取った場合は Tout1 内にアダプタ初期化設定完了通知受理応答を基本ミドルウェアアダプタに返す。

【アダプタ側初期化シーケンス】

基本ミドルウェアアダプタは、機器情報保持のアダプタ初期化設定要求を受信した場合、アダプタ初期化設定応答を Tout0 内に返信し、内部の初期化(下位通信層の指定されたスタート処理) を始める。

機器情報破棄のアダプタ初期化設定要求を受信した場合は、機器オブジェクトに関する情報を破棄した後、同様の処理を行う。

初期化が終了した後、アダプタ初期化完了通知を ECHONET Lite レディ機器に送る。

その後 Tout1 内に初期化正常終了アダプタ初期化完了通知受理応答がない場合は、1 回再送を行う。再送後も応答がない場合は、処理を中止し、アダプタ初期化設定要求を待つ。

3.8.5.3 オブジェクト構築モード

オブジェクト構築モードのシーケンスを機器オブジェクト保持の場合と、機器オブジェクト非保持の場合とで、それぞれ下図（図3-23、図3-24）に示す。

(1) オブジェクト構築シーケンス (Required)

a) 機器オブジェクト保持の場合

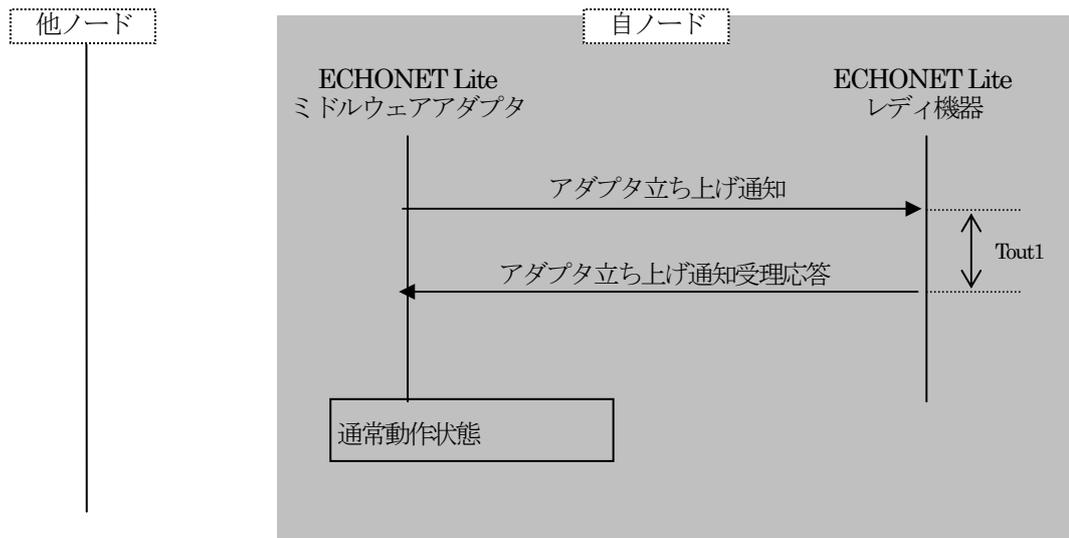


図3-23 オブジェクト動作シーケンス (1)

【アダプタ側オブジェクト構築シーケンス】

アダプタ立ち上げ通知をレディ機器へ送信した後、レディ機器から T_{out1} 以内にアダプタ立ち上げ通知受理応答を受信するのを確認する。その後、通常動作状態へ遷移する。

【機器側オブジェクト構築シーケンス】

アダプタ立ち上げ通知を受信した場合、 T_{out1} 以内にアダプタ立ち上げ通知受理応答を送信する。

b) 機器オブジェクト非保持の場合

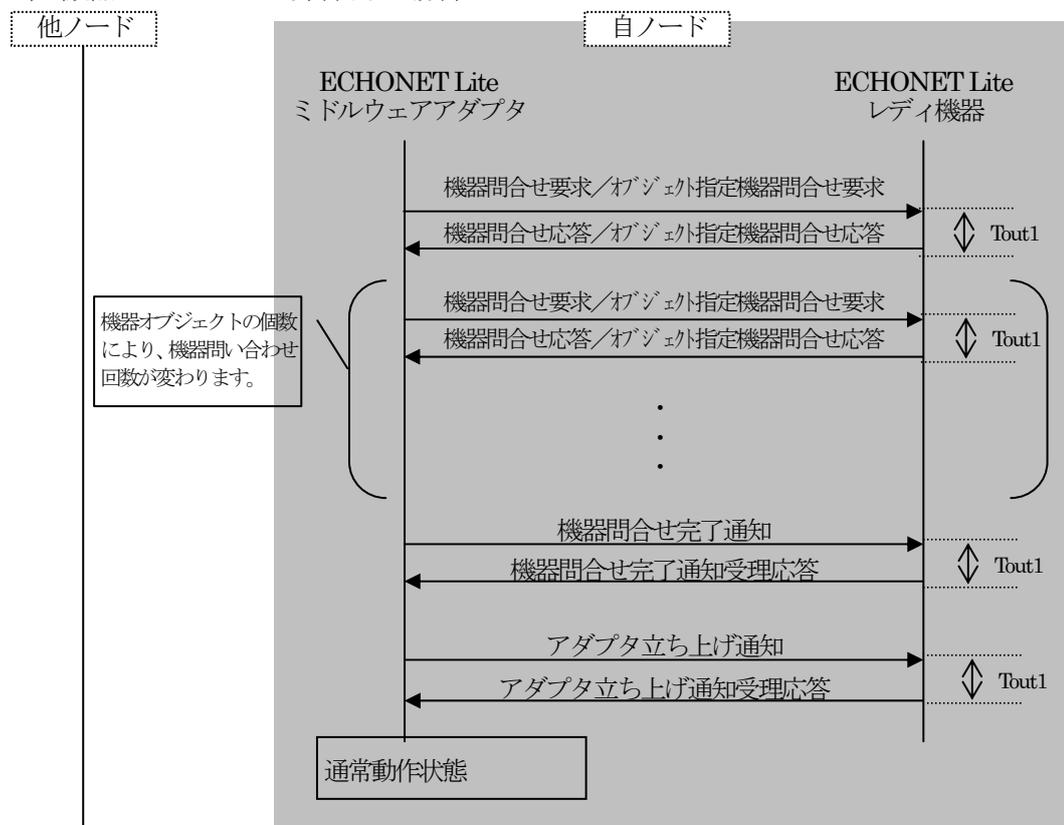


図3-24 オブジェクト動作シーケンス (2)

【アダプタ側オブジェクト構築シーケンス】

機器問い合わせ要求 (オブジェクト指定機器問い合わせ要求) をレディ機器へ送信した後、レディ機器から Tout1 以内に機器問い合わせ応答 (オブジェクト指定機器問い合わせ応答) を受信するのを確認する。オブジェクトの個数が複数の場合は、オブジェクト情報を取得完了するまで処理を繰り返す。

オブジェクト情報取得が終わると、機器問い合わせ完了通知をレディ機器へ送信する。レディ機器から Tout1 以内に機器問い合わせ完了通知受理応答を受信するのを確認する。

アダプタ立ち上げ通知をレディ機器へ送信した後、レディ機器から Tout1 以内にアダプタ立ち上げ通知受理応答を受信するのを確認する。その後、通常動作状態へ遷移する。

【機器側オブジェクト構築シーケンス】

機器問い合わせ要求 (オブジェクト指定機器問い合わせ要求) を受信した場合、Tout1 以内に機器問い合わせ応答 (オブジェクト指定機器問い合わせ応答) を送信する。搭載オブジェクト情報が複数の場合は、処理を繰り返し、機器問い合わせ完了通知を受信した場合、Tout1 以内に機器問い合わせ完了通知受理応答を送信する。

アダプタ立ち上げ通知を受信した場合、Tout1 以内にアダプタ立ち上げ通知受理応答を送信する。

3.8.5.4 ECHONET Lite通信モード

ECHONET Lite 通信モードでは、機器状態アクセス要求、機器状態通知要求、要素指定機器状態アクセス要求、要素指定機器状態通知要求、オブジェクトアクセス要求の5つのコマンドシーケンスを定義しており、それぞれ下図（図3-25、図3-26、図3-27、図3-28、図3-29）に示す。

(1) 機器状態アクセス要求シーケンス (Required)

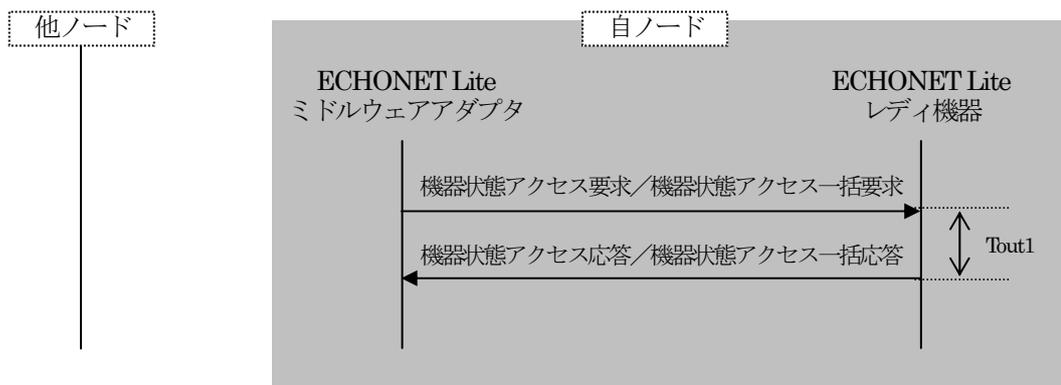


図3-25 機器状態アクセス要求シーケンス

【アダプタ側機器状態アクセス要求シーケンス】

機器状態アクセス要求（機器状態アクセス一括要求）をレディ機器へ送信した後、レディ機器から T_{out1} 以内に機器状態アクセス応答（機器状態アクセス一括応答）を受信するのを確認する。

【機器側機器状態アクセス要求シーケンス】

機器状態アクセス要求（機器状態アクセス一括要求）を受信した場合、 T_{out1} 以内に機器状態アクセス応答（機器状態アクセス一括応答）を送信する。

(2) 機器状態通知要求シーケンス (Required)

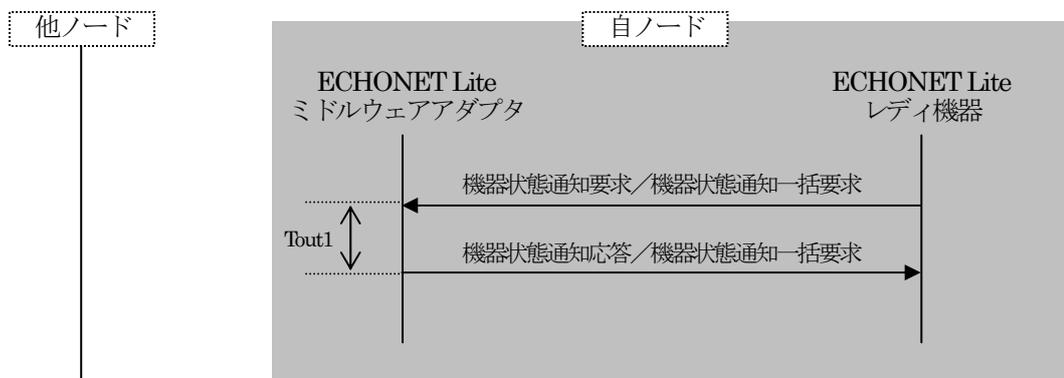


図3-26 機器状態通知要求シーケンス

【アダプタ側機器状態通知要求シーケンス】

レディ機器から機器状態通知要求（機器状態通知一括要求）を受信した場合、**Tout1** 以内に機器状態通知応答（機器状態通知一括要求）をレディ機器へ送信する。

【機器側機器状態通知要求シーケンス】

機器状態通知要求を送信した後、**Tout1** 以内に機器状態通知応答（機器状態通知一括応答）を受信するのを確認する。

(3) 要素指定機器状態アクセス要求シーケンス (Required)

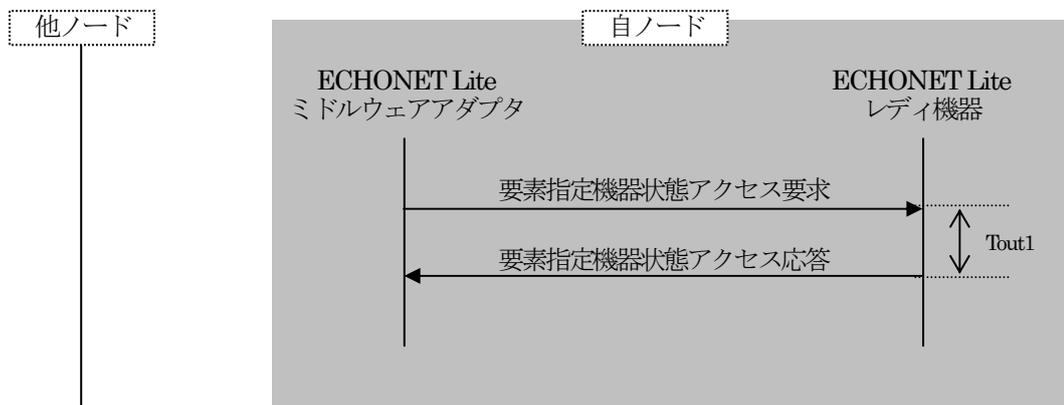


図3-27 要素指定機器状態アクセス要求シーケンス

【アダプタ側要素指定機器状態アクセス要求シーケンス】

要素指定機器状態アクセス要求をレディ機器へ送信した後、レディ機器から **Tout1** 以内に要素指定機器状態アクセス応答を受信するのを確認する。

【機器側要素指定機器状態アクセス要求シーケンス】

要素指定機器状態アクセス要求を受信した場合、**Tout1** 以内に要素指定機器状態アクセス応答を送信する。

(4) 要素指定機器状態通知要求シーケンス (Required)

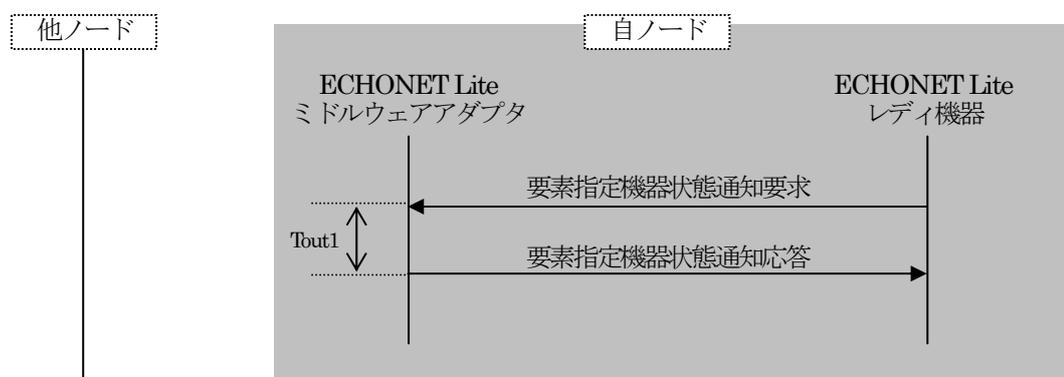


図3-28 要素指定機器状態通知要求シーケンス

【アダプタ側要素指定機器状態通知要求シーケンス】

レディ機器から要素指定機器状態通知要求を受信した場合、Tout1 以内に要素指定機器状態通知応答をレディ機器へ送信する。

【機器側要素指定機器状態通知要求シーケンス】

要素指定機器状態通知要求を送信した後、Tout1 以内に要素指定機器状態通知応答を受信するのを確認する。

(5) オブジェクトアクセス要求シーケンス (Required)

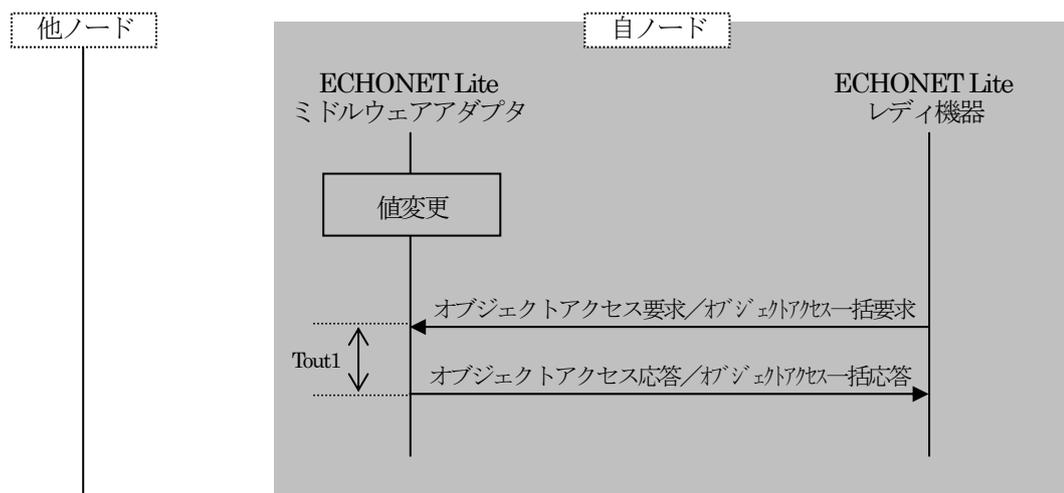


図3-29 オブジェクトアクセス要求シーケンス

【アダプタ側オブジェクトアクセス要求シーケンス】

レディ機器からオブジェクトアクセス要求 (オブジェクトアクセス一括要求) を受信した場合、Tout1 以内にオブジェクトアクセス応答 (オブジェクトアクセス一括応答) をレディ機器へ送信する。

【機器側オブジェクトアクセス要求シーケンス】

オブジェクトアクセス要求（オブジェクトアクセス一括要求）を送信した後、Tout1 以内にオブジェクトアクセス応答（オブジェクトアクセス一括応答）を受信するのを確認する。

3.8.5.5 機器側通信エラー処理シーケンス

ECHONET Lite レディ機器は、ミドルウェアアダプタからのフレーム受信時に通信エラーが発生した場合、受信フレームを破棄して通信エラー通知フレームを送信する。

ECHONET Lite レディ機器は、コマンドを発行した時にミドルウェアアダプタから応答データとして通信エラーフレームを受信した場合は、タイムアウト時処理と同様にコマンドが正常通知できなかつたと判断して、再度コマンドを発行する。

ただし、通信エラーが頻発する場合、エラー表示を行うことを推奨する。

3.8.5.6 アダプタ側通信エラー処理シーケンス

ミドルウェアアダプタは、ECHONET Lite レディ機器からのフレーム受信時に通信エラーが発生した場合、受信フレームを破棄して通信エラー通知フレームを送信する。

ミドルウェアアダプタは、コマンドを発行した時に ECHONET Lite レディ機器から応答データとして通信エラーフレームを受信した場合は、タイムアウト時処理と同様にコマンドが正常通知できなかつたと判断して、再度コマンドを発行する。

ただし、通信エラーが頻発する場合、異常内容プロパティにエラーコードをセットし、異常状態プロパティをエラー発生とし、3.7.5 項に記載する通信異常処理を行う。

3.8.5.7 同時発行可能フレーム数

基本ミドルウェアアダプタにおいて、ECHONET Lite ミドルウェアアダプタ、ECHONET Lite レディ機器は、要求フレーム、あるいは通知フレームは送信後、その応答フレームを受信するまで新たな要求フレーム、あるいは通知フレームを送信してはならない。応答フレームを返信する前、または要求に新たな要求フレーム、あるいは通知フレームを受信した場合、それは破棄されなくてはならない。

3.8.5.8 複合電文処理

基本ミドルウェアアダプタは、複合電文処理機能を有する場合、ECHONET Lite 上の他ノードからの複合電文による Get 要求受信時には、以下の処理を行う。

- (1) 複合にて指定されているプロパティが、IAGetup サービス指定のものを含まない場合には、基本ミドルウェアアダプタにて応答電文を構築し、返信を返す。
- (2) 複合にて指定されているプロパティが、IAGetup サービス指定のものを含む場合には、機器状態アクセス要求コマンド或いは機器状態アクセス一括 UP 要求コマンドにて機器からの値の取得後に、応答電文を構築し、返信を返す。

また、基本ミドルウェアアダプタが、複合電文処理機能を有する場合、ECHONET Lite 上の他のノードから複合電文による Set 要求受信時には、以下の処理を行う。

- (1) 複合にて指定されているプロパティが、IASetup サービス指定のものを含まない

場合には、基本ミドルウェアアダプタにて応答電文を構築し、返信を返す。

- (2) 複合にて指定されているプロパティが、IASetup サービス指定のものを含む場合には、機器状態アクセス要求コマンド或いは機器状態アクセス一括 UP 要求コマンドにて Set 要求を通知後に、応答電文を構築し、返信を返す。

3.8.5.9 タイムアウト時間

タイムアウト時間を、下表のように定める。

表3-11 タイムアウト時間一覧

記号	名称	時間	内容
Tout0、 Tout1	最大応答時間1	3sec	ミドルウェアアダプタ・ECHONET Lite レディ機器間で、要求受信から応答返信までの最大値
Tout10	最大初期化処理時間	5sec	初期化処理に要することが可能な最大時間
Tout11	最大初期化処理待ち時間	6sec	初期化処理待ち可能な最大時間
Tout2	最大応答時間2	5sec	ノード間で、要求受信から応答返信までの最大値
Tout30	最大機器インタフェース情報確認処理待ち時間 (ECHONET Lite レディ機器)	3sec	ECHONET Lite レディ機器がミドルウェアアダプタに対して機器インタフェース情報確認応答を返信するまでの最大値
Tout40	最小オブジェクト情報破棄待ち時間	6sec	ミドルウェアアダプタが機器インタフェース情報を破棄するのを ECHONET Lite レディ機器が待つ最大値
Tout50	最大機器インタフェース情報確認要求発行時間	3sec	ECHONET Lite レディ機器がミドルウェアアダプタに対して機器インタフェース情報確認応答を返信するまでの最大値
Tout61	最大機器インタフェース情報確認処理待ち時間 (ミドルウェアアダプタ)	5sec	ミドルウェアアダプタが機器インタフェース情報確認応答を待つ時間。

※注：Tout61>Tout30、Tout61>Tout50

3. 8. 6 オブジェクト生成タイプの機械・物理特性

通信インタフェースは、基本的に機器インタフェース情報認識サービスの機械・物理特性に準拠とするが、RST (リセット) 端子および伝送速度の取り扱いが以下ようになる。

○RST (リセット)

ECHONET Lite レディ機器からアダプタへのリセット出力

Low で動作停止状態、Low→High でリセット起動

ECHONET Lite レディ機器 : 必須とする

ミドルウェアアダプタ : 必須とする

○伝送速度： 2400bps / 4800bps / 9600bps / 19200bps / 38400bps / 57600bps / 115200bps

3. 9 Peer to Peer タイプ用通信プロトコル

ECHONET Lite ミドルウェアアダプタと ECHONET Lite レディ機器の間で、ユーザ定義の通信方式によって、情報授受を行う為のプロトコルである。Peer to Peer タイプを実現する形態には以下の2つがある。

- プログラム選択形態
- プログラムダウンロード形態

3. 9. 1 プログラム選択形態

プログラム選択形態は、Peer to Peer タイプの通信のうち、ECHONET Lite レディ機器に対応した通信方式（ECHONET Lite ミドルウェアアダプタ通信ソフトウェア）を、あらかじめ ECHONET Lite ミドルウェアアダプタ内に備えている形態である。本規格では特に規定しない。

3. 9. 2 プログラムダウンロード形態

プログラムダウンロード形態は、Peer to Peer タイプの通信のうち、ECHONET Lite レディ機器に対応した通信方式（ECHONET Lite ミドルウェアアダプタ通信ソフトウェア、以下プログラム）を、外部からダウンロードして入手する形態である。本バージョンでは、ECHONET Lite レディ機器からプログラムをダウンロードするプロトコルについて規定する。また、ダウンロードしたプログラムを ECHONET Lite ミドルウェアアダプタ内で実行するプログラム実行環境を推奨仕様として規定する。（図3-30）

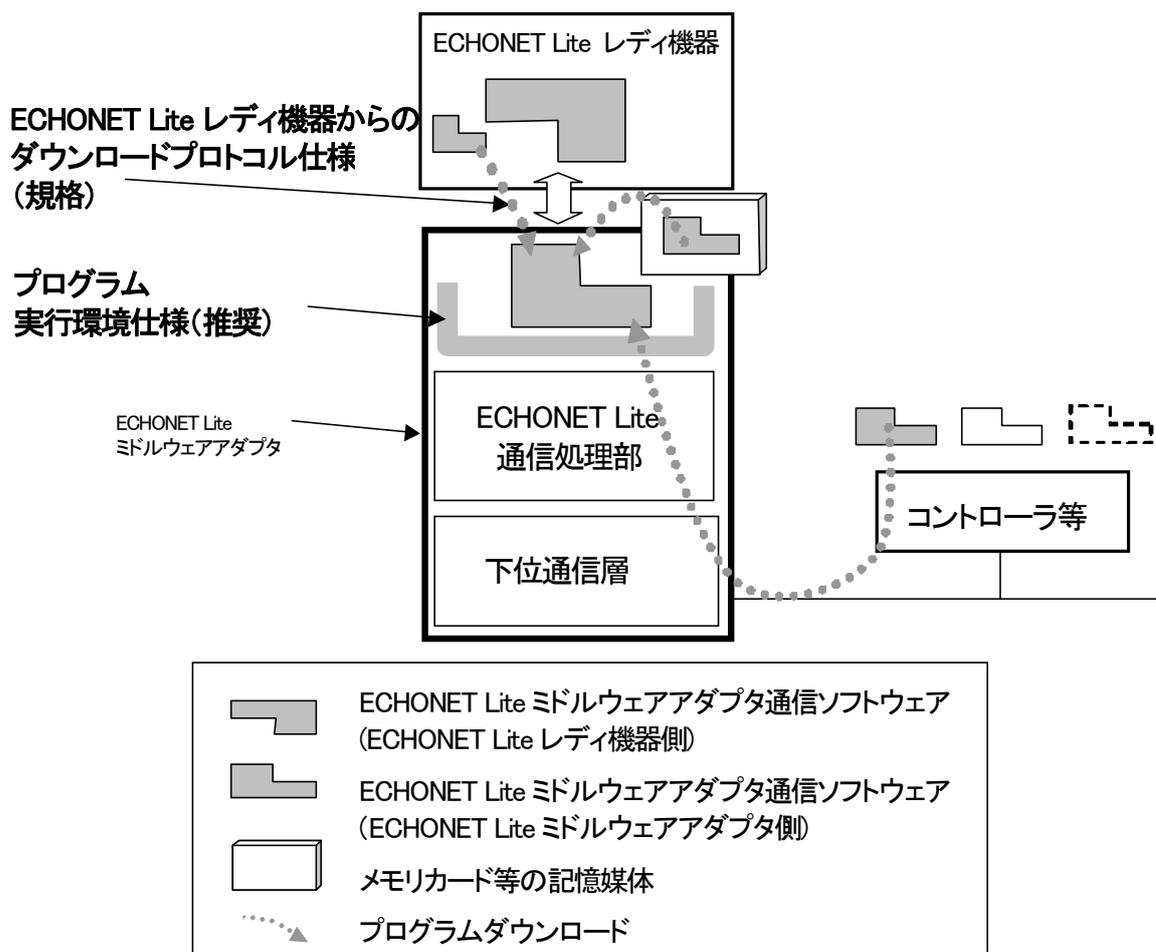


図3-30 プログラムダウンロード形態の仕様化範囲

3. 9. 3 ECHONET Liteレディ機器からのプログラムダウンロードプロトコル

ECHONET Lite レディ機器から、ECHONET Lite ミドルウェアアダプタ通信インタフェースを介してプログラムをダウンロードするプロトコルを示す。

3.9.3.1 プログラムダウンロード用フレーム構成

プログラムダウンロード形態のプロトコルで使用するフレームの構成を図3-31のとおり規定する。フレームタイプコード (FT)、コマンド番号コード (CN)、フレーム番号コード (FN)、データ長コード (DL)、フレームデータ (FD) の部分が、ECHONET Lite ミドルウェアアダプタ通信インタフェースプロトコルのDATAとなる。

STX	FT	CN	FN	DL	FD	FCC
1byte	2byte	1byte	1byte	2byte	nbyte	1byte

図3-31 プログラムダウンロード用フレーム構成

(1) STX (制御コード)

制御コード。 0x02 固定とする。

(2) FT (Frame Type : フレームタイプ)

フレーム毎のタイプを示す。ECHONET Lite ミドルウェアアダプタ通信インタフェースのプログラムダウンロードプロトコルにおいては、0x0100 固定とする。

(3) CN (Command No. : コマンド番号コード)

ECHONET Lite ミドルウェアアダプタ通信インタフェースのプログラムダウンロードプロトコルで規定するコマンドを指定するための1バイトのコードとする。本バージョンにおいては、表3-12で示したコマンドを規定する。具体的な割り当ての無いコードについては、for future reserved とする。

表3-12 プログラムダウンロードインタフェースのコマンドコード

コマンド名	コマンド番号コード (CN)	搭載
プログラムダウンロード要求	0x00	Required
プログラムダウンロード応答	0x80	Required
ダウンロード完了通知	0x01	Required
ダウンロード完了応答	0x81	Required

(4) FN (Frame No : フレーム番号)

要求側が付与する番号。(0x01~0xFF) 要求側はインクリメンタルに番号を付与する必

要がある。応答フレームは対応する要求フレームと同じ番号にする必要がある。

なお、応答フレームに要求フレームと同じ番号を付与することができないECHONET Lite レディ機器についてはこの番号を0x00固定値とする。

(5) DL (Data Length : データ長コード)

データ長コードは、後に続くフレームデータ (FD) 部のサイズを示す2バイト長のコードとする。サイズは、バイト数とし、HEX表示する。例えば、FD部が20バイトの場合には、DLは20バイトを示す0x0014 となる。尚、本データの配置はビッグエンディアンとする。

(6) FD (Frame Data: フレームデータ)

フレームデータ部は、フレームタイプ (FT) 及びコマンド番号コード (CN) により規定されるデータのフィールドである。2バイト以上のデータの並びはビッグエンディアンとする。具体的な構成は、コマンド番号コード (CN) 毎に規定する。

(7) FCC (Frame Check Code: フレームチェックコード)

フレームチェックコードとして1バイトのチェックコードを規定する。Frame Type 以降、Frame DATA までの合計値の2の補数とする。

3.9.3.2 プログラムダウンロード用コマンド

各 ECHONET Lite ミドルウェアアダプタ通信インタフェースのプログラムダウンロードプロトコルで用いるコマンドの詳細を規定する。

(1) プログラムダウンロード要求/応答コマンド (Required)

◇要求コマンドの方向

ECHONET Lite ミドルウェアアダプタ → ECHONET Lite レディ機器

◇要求コマンド

1バイト	2バイト	1バイト	1バイト	2バイト	2バイト	7バイト	9バイト	1バイト
STX	FT	CN	FN	DL	FD(0)	FD(1)	FD(2)	FCC

STX : 0x02
 FT : 0x0100
 CN : 0x00
 FN : 0x**
 DL : 0x0012
 FD(0) : 0x**** 送信要求シーケンス番号(2バイト)
 0x0000 : 送信対象情報の取得要求
 0x0001～ : 分割データシーケンス番号
 FD(1) : 対象機器情報(7バイト)
 ECHONET Lite メーカーコード(3バイト)
 機種コード(2バイト)
 型式コード(2バイト)
 ※機器インタフェース情報認識サービスで取得した情報で埋める
 FD(2) : ソフトウェア情報(9バイト)
 ECHONET Lite メーカーコード(3バイト)
 プログラム識別子(6バイト)
 ※既にダウンロードを実施し、ECHONET Lite ミドルウェアアダプタ内にプログラムを保持している場合は、その情報で埋める
 ※ダウンロードを実施しておらず、プログラムを保持していない場合は、全て 0x f f で埋める
 FCC : 0x**

◇応答コマンド

a) 要求コマンドの FD(0)=0x00 の場合、(送信対象情報の取得要求)

1バイト	2バイト	1バイト	1バイト	2バイト	1バイト	2バイト	9バイト	2バイト	1バイト
STX	FT	CN	FN	DL	FD(0)	FD(1)	FD(2)	FD(3)	FCC

- STX : 0x02
- FT : 0x0100
- CN : 0x80
- FN : 0x**
- DL : 0x000e
- FD(0) : データタイプ(プログラム 0x00/取得先 0x01)
 ※取得先とは、URL 情報や、ダウンロードデータのパス情報などを考慮した情報で、
 記述方法及び、取得先を元にした処理は規定しない。
- FD(1) : 電文分割数(2 バイト)
- FD(2) : ソフトウェア情報(9 バイト)
 ECHONET Lite メーカーコード(3 バイト)
 プログラム識別子(6 バイト)
 ※プログラム識別子は、ECHONET Lite レディ機器メーカー内でプログラムを一意に識
 別可能とするためのメーカー独自規定とする
 ※FD (0) =0x01(応答データが取得先)の場合は、全て 0x f f で埋める
- FD(3) : ダウンロードデータサイズ(2 バイト)
- FCC : 0x**

b) 要求コマンドの FD(0)≠0x00 の場合、(分割データシーケンス番号)

1バイト	2バイト	1バイト	1バイト	2バイト	2バイト	最大128バイト	1バイト
STX	FT	CN	FN	DL	FD(0)	FD(1)	FCC

- STX : 0x02
- FT : 0x0100
- CN : 0x80
- FN : 0x**
- DL : 0x****
- FD(0) : 分割データシーケンス番号
- FD(1) : ダウンロードデータ(n バイト、最大 128 バイト)
- FCC : 0x**

(2) ダウンロード完了通知/応答コマンド (Required)

◇通知コマンドの方向

ECHONET Lite ミドルウェアアダプタ → ECHONET Lite レディ機器

◇通知コマンド

1バイト	2バイト	1バイト	1バイト	2バイト	1バイト	1バイト
STX	FT	CN	FN	DL	FD(0)	FCC

STX : 0x02
FT : 0x0100
CN : 0x01
FN : 0x**
DL : 0x0001
FD(0) : ダウンロード結果
 0x00 : 失敗
 0x01 : 成功
FCC : 0x**

◇応答コマンド

1バイト	2バイト	1バイト	1バイト	2バイト	1バイト
STX	FT	CN	FN	DL	FCC

STX : 0x02
FT : 0x0100
CN : 0x81
FN : 0x**
DL : 0x0000
FCC : 0x**

3.9.3.3 ダウンロードデータフォーマット仕様

図3-3 2にダウンロードデータのデータフォーマットを規定する。1 バイトのプログラム実行環境識別子と、プログラム本体から構成し、プログラム実行環境識別子については、下記のように規定する。また、プログラム本体のフォーマットは、プログラム実行環境識別子毎に規定することとする。

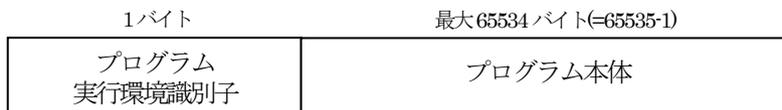


図3-3 2 ダウンロードデータのフォーマット

(1) プログラム実行環境識別子

プログラム本体を実行可能なプログラム実行環境を表す1バイトのコードを表す。ミドルウェアアダプタは、搭載されているプログラム実行環境と比較し、ダウンロードしたプログラム本体が実行可能かを判断する。本バージョンにおいては、表3-1 3で示したプログラム実行環境識別子を規定する。新たな実行環境については、このプログラム実行環境識別子を追加規定する。具体的な割り当てのないコードについては、for future reserved とする。

表3-1 3 プログラム実行環境識別子コード

実行環境名	プログラム実行環境識別子
インタプリタ方式 (3.10 にて定義)	0x00

3.9.3.4 プログラムダウンロード用通信シーケンス

各 ECHONET Lite ミドルウェアアダプタ通信インタフェースのプログラムダウンロードプロトコルで用いる通信シーケンスを規定する。

図3-33に、プログラムダウンロードプロトコルの通信シーケンスを示す。通信方式が Peer to Peer (機器インタフェース情報応答コマンドFD (0) : b0 が1)、かつ、ECHONET Lite レディ機器がダウンロード情報を保持する (機器インタフェース情報応答コマンドFD (2) のインタフェース情報 : b3 が1) 場合、ECHONET Lite ミドルウェアアダプタは、機器インタフェース情報認識サービスで規定された移行時間(T_{trans})経過後、プログラムダウンロード要求コマンドを送信する。

アダプタからのプログラムダウンロード要求の送信は、機器インタフェース情報認識サービスで認識した伝送速度で行う。機器インタフェース情報応答コマンドで送られた伝送速度に対応できない場合は、機器インタフェース情報確定通知で、「現行速度対応可 (指定速度対応不可)」を送信し、現行速度のままプログラムダウンロードを行う。

アダプタは、プログラムダウンロード応答コマンド (要求シーケンス番号=0) のソフトウェア情報と、現在保持しているソフトウェア情報が異なり、かつ、プログラムダウンロード応答コマンド (要求シーケンス番号=1) のプログラム実行環境識別子が搭載されているプログラム実行環境を示している場合に、要求シーケンス番号=2 以降のダウンロードを実行する。プログラムダウンロード応答コマンド (要求シーケンス番号=0) のソフトウェア情報と、現在保持しているソフトウェア情報が同じ、かつ、プログラムダウンロード応答コマンド (要求シーケンス番号=1) のプログラム実行環境識別子が搭載されているプログラム実行環境を示している場合は、ダウンロードを実行せずに、プログラムダウンロード完了通知をFD (0) =0x01 (成功) を送信する。

応答待ちタイムアウト時間($T1$)は、300ms とする。

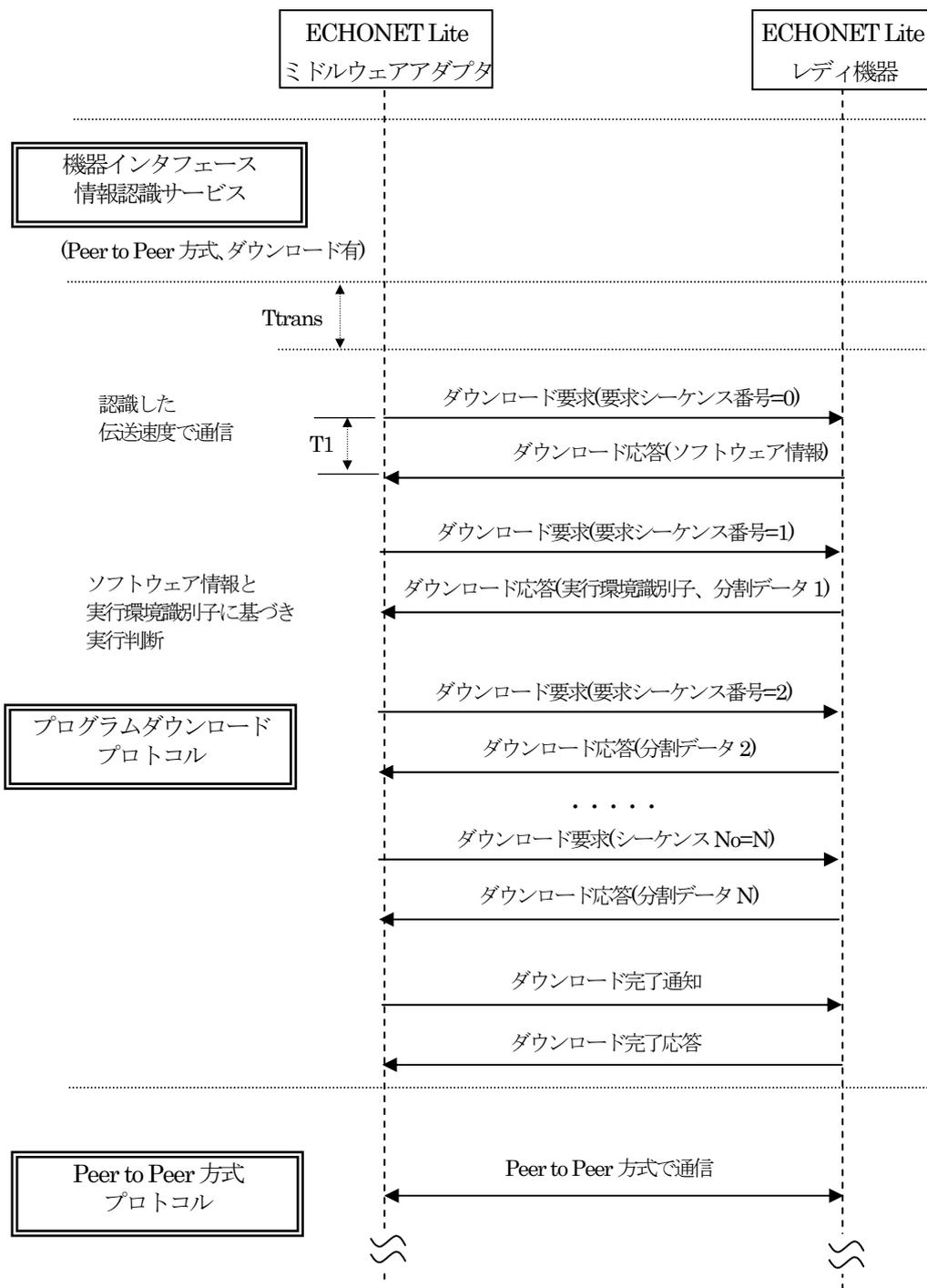


図3-33 プログラムダウンロード通信シーケンス

T_{trans} : 各方式仕様への移行時間。500msec 以上あけること。
 T_i : 受信タイムアウト時間。MAX.300msec

3.9.3.5 異常処理

プログラムダウンロード通信シーケンス中に発生した異常処理について以下のように規定する。

(1) 表示部

ミドルウェアアダプタは、ダウンロード中に発生した異常を通知するための表示部を搭載しなければならない。表示手段については特に規定しないが、表示するためにLEDを具備する場合は、ECHONET Lite 規格書第3部第3章3.3.2表示部を推奨とする。

(2) ダウンロードデータの受信失敗

プログラムダウンロード中に受信に失敗した場合（受信タイムアウト、FCCエラー、FrameNo不正^{*1}）は、3回まで再送する。再送を3回してもなお受信に失敗した場合は、プログラムダウンロード不可異常とし、表示部に異常表示する。

*1 FrameNo不正：受信した応答コマンドのFrameNoが、0でないかつ、要求コマンドのFrameNoと異なる場合。

(3) ダウンロードデータの大きさが受信範囲超過（ダウンロード応答失敗）

ダウンロード応答受信時に含まれるデータサイズがアダプタ内部で許容するダウンロードデータのサイズを越えている場合は、ダウンロードデータは破棄し、表示部に異常表示する。

(4) ダウンロードデータエラー

プログラムダウンロード応答コマンド中のECHONET Lite メーカーコードが要求したメーカーコードと異なる場合、又はプログラムダウンロード応答コマンド（要求シーケンス番号=1）のプログラム実行環境識別子が自己の実行環境と異なる場合は、ダウンロードデータを破棄し、表示部に異常表示する。

3. 10 Peer to Peer タイプのプログラムダウンロード形態におけるインタプリタ方式プログラム実行環境仕様 (推奨)

3. 10. 1 本推奨仕様の適用範囲

本推奨仕様では、Peer to Peer タイプのプログラムダウンロード形態におけるプログラム実行環境仕様のうち、ECHONET Lite ミドルウェアアダプタに搭載し、ダウンロードデータ中のプログラム本体を解釈・実行するインタプリタ方式の実行環境仕様について規定する。

- (1) プログラム本体フォーマット仕様
- (2) ダウンロードプログラム言語仕様
- (3) インタプリタAPI仕様
- (4) プログラム圧縮・伸張仕様

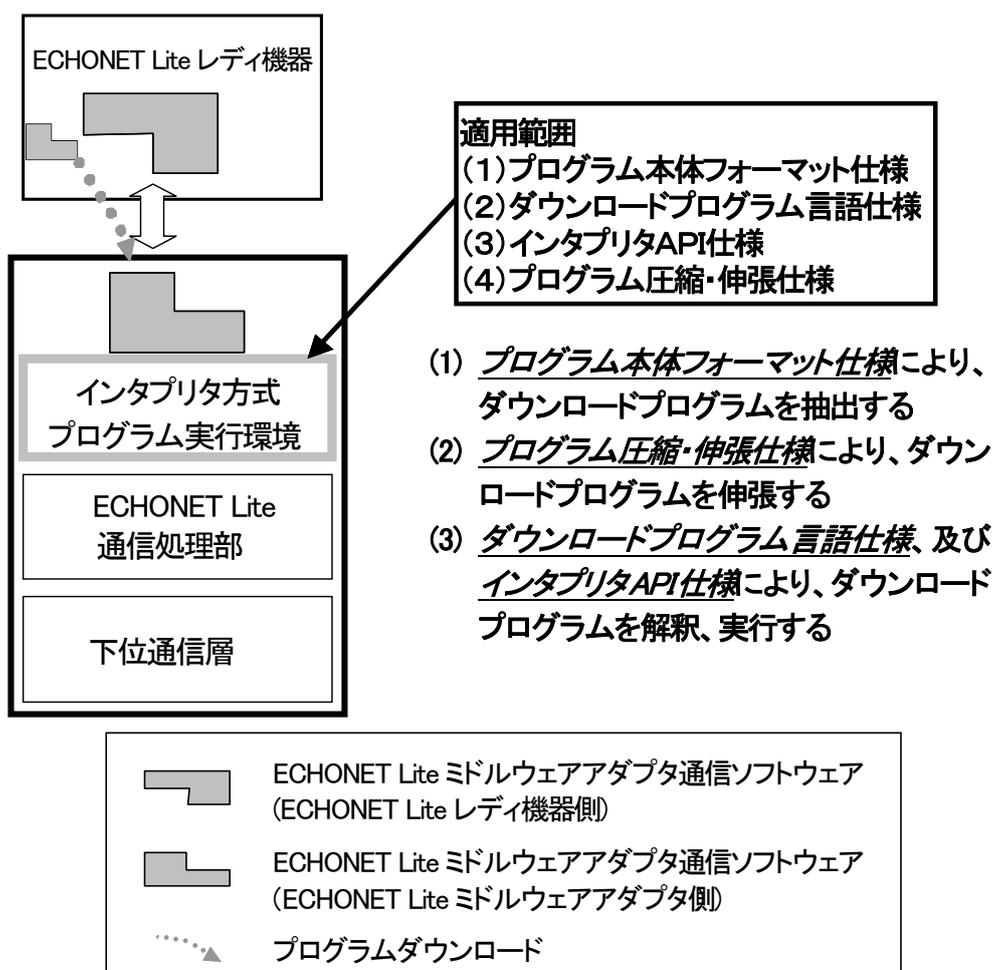


図3-34 適用範囲

3. 10. 2 インタプリタ方式プログラム実行環境の概要

3.10.2.1 ダウンロードプログラムとインタプリタの機能分担

図3-35に、ダウンロードプログラムとインタプリタとの間の機能分担を示す。方式毎にフォーマットが異なる各種ユーザ定義の通信方式のコマンドを統一的に扱えるように、中間オブジェクトと呼ぶ仮想的なオブジェクトの概念をインタプリタ API に導入する。

ダウンロードプログラムは、ECHONET Lite ミドルウェアアダプタ通信インタフェースの各種ユーザ定義の通信方式のコマンドを、中間オブジェクトを用いたコマンドに変換し、インタプリタ API にアクセスする。

インタプリタは、ダウンロードプログラムより予め設定された変換テーブルを用いて、中間オブジェクトから ECHONET オブジェクトの変換を行い、ECHONET Lite 通信処理部 API にアクセスする。

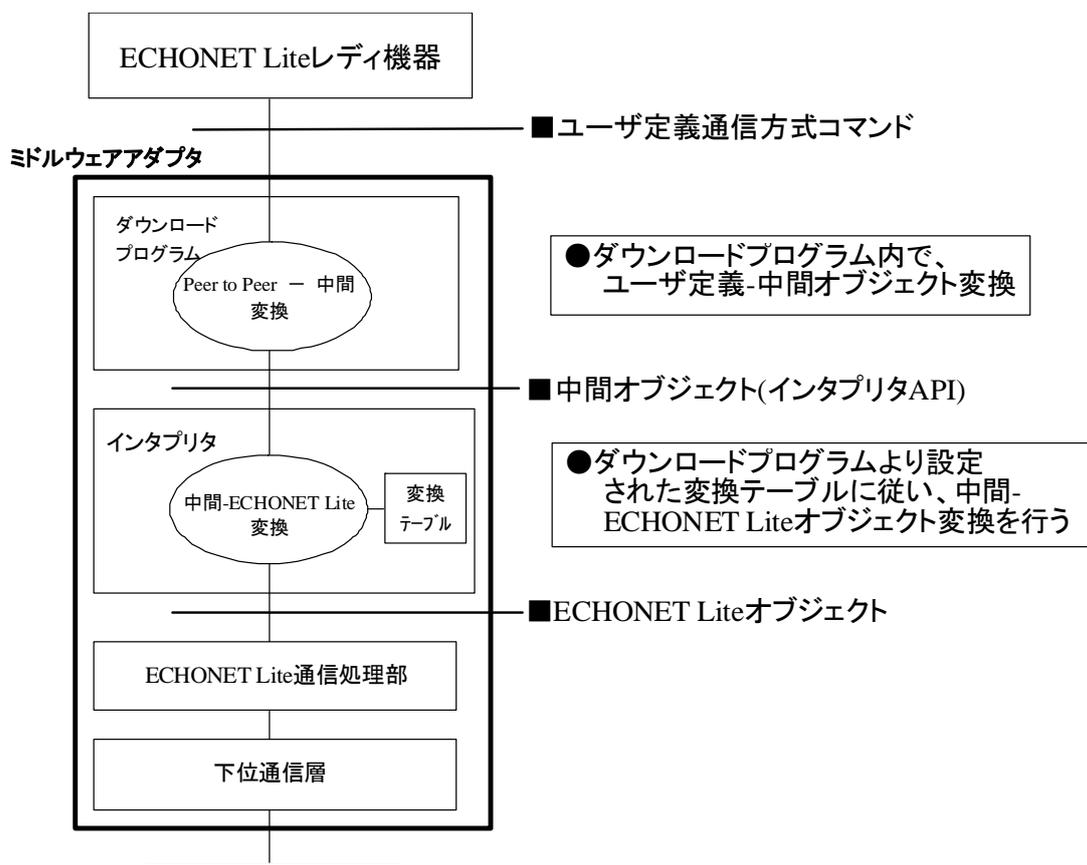
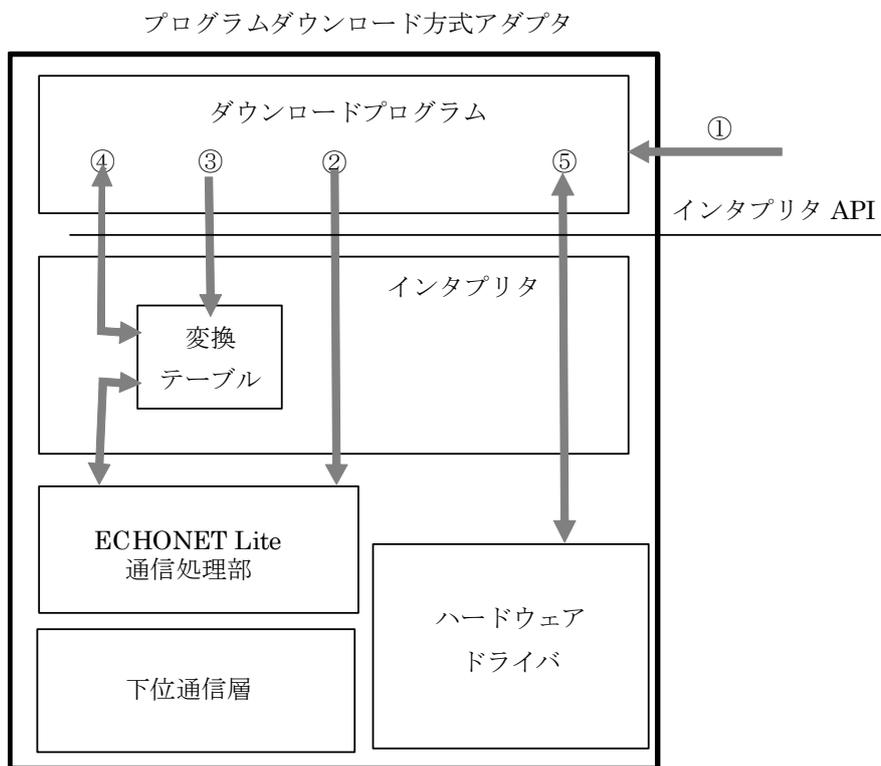


図3-35 機能分担

3.10.2.2 動作手順

図3-36に、インタプリタ方式プログラム実行環境を搭載した ECHONET Lite ミドルウェアアダプタの動作手順を示す。

- ① プログラムダウンロード
ECHONET Lite レディ機器よりプログラムをダウンロードする
- ② ECHONET オブジェクト生成
インタプリタがダウンロードプログラムを読み込み実行を開始し、ダウンロードプログラムから ECHONET オブジェクトを生成する。
自ノードオブジェクト及び、ノードプロファイルオブジェクトはインタプリタが生成する。
- ③ 変換テーブルの初期化
ダウンロードプログラムから、中間オブジェクト→ECHONET オブジェクトの相互変換を行う変換テーブルの初期化を行う。
- ④ ECHONET Lite 初期化
ダウンロードプログラムから ECHONET Lite 初期化 API を呼び出す。
- ⑤ ECHONET Lite コマンド送受信/Peer to Peer 方式コマンド送受信
ECHONET Lite を利用した通信は、変換テーブルを介して行う。
ECHONET Lite レディ機器との通信、データ解釈は、ダウンロードプログラムが直接行う。



3.10.2.3 変換テーブルモデル

(1)変換テーブルモデルの構成

図3-37に、本方式で使用する変換テーブルのモデルを示す。

- ECHONET Lite ノードは、複数(n 個)の中間オブジェクトを所有している。
- 中間オブジェクトはIDにより識別される。ECHONET Lite のクラスグループ、クラス、インスタンス情報を保持している。
- 中間オブジェクトは、複数(N 個)の中間オブジェクトプロパティと、複数(M 個)のECHONET プロパティを所有している。
- 中間オブジェクトプロパティと ECHONET プロパティは変換テーブルにて関連付けられている。
- プロパティ間の関連には、同一値型、マッピング型、関数型の3種類があり、それぞれの型に従った変換テーブルが生成される。
- 同一値型は、1対1のプロパティコードの関連を、マッピング型は、N対Mのプロパティコードと値の関連を、関数型は、N対Mのプロパティコードの関連を表す。
- アダプタは、上記テーブルセットをノードの個数分保持している。

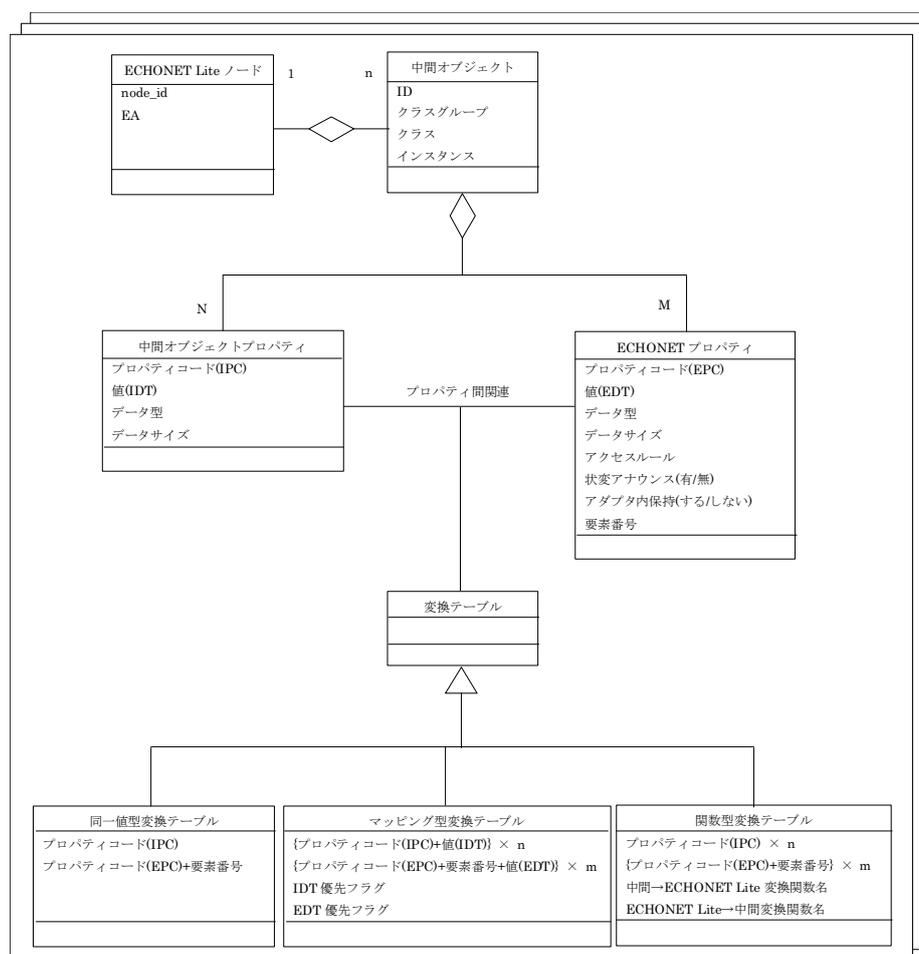


図3-37 変換テーブルモデルの概要

(2)ECHONET Lite ノード

◇概要

アダプタが所有する ECHONET Lite ノードの定義情報

◇レコード形式

ノードID	通信アドレス
-------	--------

※ノードID は、アダプタ内でノードを一意に識別するための ID である。

※通信アドレスは下位通信層の通信に使用されるアドレスである。

◇使用例

自ノードが、通信アドレス A、コントローラノードが通信アドレス B の場合
(ただし通信アドレス A、B には実際のアドレスの数値が入る)、

ノードID	通信アドレス
0	通信アドレスA
1	通信アドレスB

※自ノードは、ノードID=0 で登録される。

(3)中間オブジェクトモデル

◇概要

アダプタが所有する中間オブジェクトの定義情報

◇レコード形式

中間オブジェクトID	ECHONET		
	クラスグループ	クラス	インスタンス

◇使用例

家庭用エアコン、温度センサ、湿度センサを各1台ずつ所有している場合、

中間オブジェクトID	ECHONET		
	クラスグループ	クラス	インスタンス
1	0x01	0x30	0x01
2	0x00	0x11	0x01
3	0x00	0x12	0x01

(4)中間オブジェクトプロパティモデル

◇概要

中間オブジェクトが所有するプロパティの定義情報

◇レコード形式

中間オブジェクトID	プロパティ		
	プロパティコード(IPC)	型	サイズ

◇使用例

中間オブジェクト(ID=1)が、

- ・コード(IPC)=0x01 として、unsigned char 型を 2 個、
- ・コード(IPC)=0x02 として、signed long 型を 1 個

所有している場合、

中間オブジェクトID	プロパティ		
	プロパティコード(IPC)	型	サイズ
1	0x01	unsigned char	2
	0x02	signed long	4

(5)ECHONET プロパティモデル

◇概要

中間オブジェクトが所有する ECHONET プロパティの定義情報

◇レコード形式

中間オブジェクト ID	プロパティ					
	プロパティコード (EPC)	要素番号	型	アクセス ルール	状態 アナウンス	サイズ

◇使用例

中間オブジェクト(ID=1)が、

- ・コード(EPC)=0xB0 として、unsigned char 型を 1 個、
- ・コード(EPC)=0xC9、第 0 要素として、unsigned char 型を 1 個
 第 1 要素として、unsigned char 型を 1 個

を所有している場合、

中間オブジェクト ID	プロパティ					
	プロパティコード (EPC)	要素番号	型	アクセス ルール	状態 アナウンス	サイズ
1	0xB0	-	unsigned char	set get	無	1
	0xC9	0	unsigned char	set get	無	1
	0xC9	1	unsigned char	set get	無	1

(6)同一値型変換テーブルモデル

◇概要

変換が不要な同一値型プロパティの変換情報

◇レコード形式

中間オブジェクトID	IPC	EPC	要素番号
------------	-----	-----	------

◇使用例

中間オブジェクト ID=1 の

- ・IPC=0x10 を、EPC=0x20 と、
- ・IPC=0x11 を、EPC=0x30、要素番号3 と同一値変換する場合。

中間オブジェクトID	IPC	EPC	要素番号
1	0x10	0x20	-
	0x11	0x30	3

(7)マッピング型変換テーブルモデル

◇概要

マッピングテーブルによる値変換を行うマッピング型プロパティの変換情報

◇レコード形式

中間オブジェクトID	MapID	IPC0…IPCn	IFLG	EPC0:ELE0…EPCm:ELEm	EFLG
------------	-------	-----------	------	---------------------	------

- MapID : マッピング型変換テーブルの番号 (テーブル生成時に使用する)
- IPC0 … IPCn : n 組の中間オブジェクトプロパティコード(IPC)
- IFLG : IPC→EPC 変換時にマッピングが一意に定まらない場合の優先フラグ
- EPC0:ELE0…EPCm:ELEm : m 組の(ECHONET プロパティコード+要素番号)
要素番号は、配列の場合に指定する
- EFLG : EPC→IPC 変換時にマッピングが一意に定まらない場合の優先フラグ

◇使用例

IPC=0x22 {1, 2, 3, 4, 5, 6, 7, 8}と、EPC=0xAA {A, A, A, B, B, B, C, C}をマッピングする場合。

EDT=A の場合は、IDT=1、EDT=B の場合は、IDT=5、EDT=C の場合は、IDT=8となる。

(EFLG=1 のマッピング関係が優先される)

中間オブジェクトID	MapID	IPC=0x22	IFLG	EPC=0xAA	EFLG
1	1	1	1	A	1
		2	1	A	0
		3	1	A	0
		4	1	B	0
		5	1	B	1
		6	1	B	0
		7	1	C	0
		8	1	C	1

(8)関数型変換テーブルモデル

◇概要

関数テーブルによる値変換を行う関数型プロパティの変換情報

◇レコード形式

中間オブジェクトID	IPC0...IPCn	EPC0:ELE0...EPCm:ELEm	I2E_FUNC	E2I_FUNC
------------	-------------	-----------------------	----------	----------

- IPC0 ... IPCn : n 組の中間オブジェクトプロパティコード(IPC)
- EPC0:ELE0...EPCm:ELEm : m 組の(ECHONET プロパティコード+要素番号)
要素番号は、配列の場合に指定する
- I2E_FUNC : IPC → EPC 変換に使用するダウンロードプログラム中の関数名
- E2I_FUNC : EPC → IPC 変換に使用するダウンロードプログラム中の関数名

◇使用例

EPC=0xB3 {10, 11,12,13,14,15,16, 17,18,19,1A,1B,1C,1D,1E,1F} と、
 IPC=0x11 {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F} を、
 FUNC1(X) {return X - 0x10;}
 FUNC2(X) {return X + 0x10;}
 の2関数により変換する場合。

中間オブジェクトID	IPC0...IPCn	EPC0:ELE0...EPCm:ELEm	I2E_FUNC	E2I_FUNC
1	0x11	0xB3	FUNC2	FUNC1

3. 10. 3 プログラム本体のフォーマット仕様

図3-38にインタプリタ方式プログラム実行環境における、プログラム本体のフォーマットを示す。(図中ハッチング部)

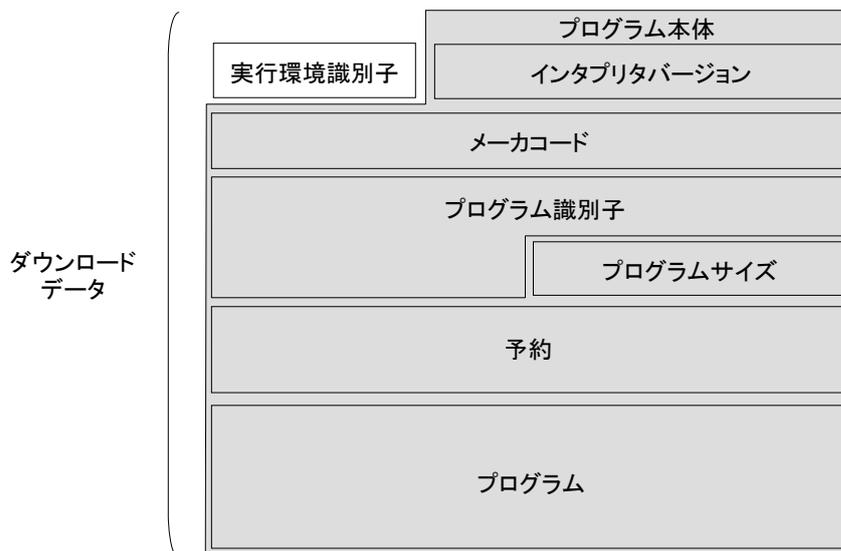


図3-38 プログラム本体のフォーマット

- (1)インタプリタバージョン (3バイト)
 - 1バイト目: 0x00 (固定)
 - 2バイト目: 仕様メジャーバージョン (バイナリ値)、本バージョンでは0x01
 - 3バイト目: 仕様マイナーバージョン (バイナリ値)、本バージョンでは0x00
- (2)メーカーコード (4バイト)
 - 1バイト目: 0x00 (固定)
 - 2~4バイト目: ECHONET Lite レディ機器のメーカーコード (ECHONET コンソーシアムで規定)
- (3)プログラム識別子 (6バイト)

プログラムダウンロード要求/応答コマンドと同じ。
- (4)プログラムサイズ(2バイト)
 - 1~2バイト: プログラムサイズ (単位: バイト)
- (5)予約領域(20バイト)
- (6)プログラム

ダウンロードプログラム言語仕様、インタプリタ API 仕様に基づき作成したプログラムを、プログラム圧縮・伸張仕様に基づき変換したプログラム。

3. 10. 4 ダウンロードプログラム言語仕様

3.10.4.1 概要

ダウンロードプログラムは、逆ポーランド記法を用いたスタック言語を用いる。

API の定義は、図 3-39 のように記述する。

```
FUNC1(arg1 arg2 arg3 <name> -- ret1 ret2 , コメント)
```

図 3-39 API定義

これはスタックダイアグラムと呼ばれ、"FUNC1"がAPI名、"--"の左側が入力(引数)、"--"の右側が出力(戻り値)を、","以降がコメントを示す。arg1 ~ arg3 を順にスタックに積み、FUNC1 を実行すると、実行後のスタックには、ret1, ret2 の順に積まれて戻ってくる。<name>は、FUNC1 実行後の次に積まれたスタックを使用することを意味する。なお、"--"の左右に何も記述がない場合は、入力、出力がないことを表す。

arg1 ~ arg3 を順にスタックに積み、FUNC1 を実行するプログラムは、図 3-40 のように記述する。

```
arg1 arg2 arg3 FUNC1 name
```

図 3-40 APIのプログラム記述方法

また、FUNC1API を実行する前後のスタックの変化を図 3-41 に示す。

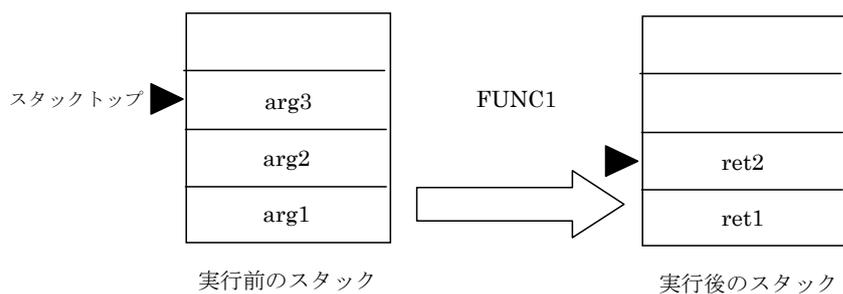


図 3-41 API実行前後のスタックの変化

3.10.4.2 字句構造

(1) 文字コード

ASCII コード (0x20~0x7E) を使用する。大文字、小文字の区別はしない。

(2) 空白

スペース (0x20) とする。

(3) 字句変換

ダウンロードプログラムは空白を区切りとしたトークンで構成される。

(4) トークン

トークンは、データ (3.10.4.3)、ユーザ定義名 (3.10.4.4)、インタプリタ基本API (3.10.5)、インタプリタ ECHONET Lite API (3.10.6) から構成される。

3.10.4.3 データ

16ビット値とする

32ビット値を扱う場合は、8ビット×4のバイト配列で表し、ビッグエンディアンにてデータを配置する。

3.10.4.4 ユーザ定義名

インタプリタ基本APIの変数定義(3.10.5.1)、配列定義(3.10.5.2)、定数定義(3.10.5.3)、関数定義(3.10.5.4)で定義されるユーザ定義名name。インタプリタ基本API、インタプリタ ECHONET Lite APIと重複して定義してはならない。

3. 1 0. 5 インタプリタ基本API仕様

3.10.5.1 変数定義(VARIABLE)

(1)機能 16ビット変数 `name` を定義する。

(2)スタックダイアグラム

VARIABLE (<name> --, 変数を定義する)

(3)記述例

アダプタの記述

```
VARIABLE n
```

C 言語の記述

```
int n;
```

3.10.5.2 配列(バイト列)定義(CREATE~ALLOT)

(1)機能 バイト配列 `name` を定義する。

(2)スタックダイアグラム

CREATE (<name> --, 配列名 `name` を定義する)
ALLOT (n --, n バイト確保する)

(3)記述例

アダプタの記述

```
CREATE buf 10 ALLOT
```

C 言語の記述

```
char buf[10];
```

3.10.5.3 定数定義(CONSTANT)

(1)機能 定数 name を定義する。

(2)スタックダイアグラム

CONSTANT (n <name> -- , 定数 name を定義する)

(3)記述例

アダプタの記述

```
128 CONSTANT MAX_CHARS
```

C 言語の記述

```
#define MAX_CHARS 128;
```

3.10.5.4 関数定義(: ~ ;)

(1)機能 関数 name を定義する

(2)スタックダイアグラム

:(<name> -- , 関数定義を開始する)

;(-- , 関数定義を終了する)

(3)記述例

アダプタの記述

```
:calc ( q -- ret )   ¥q は入力 (引数)、ret は出力 (戻り値)  
    128 + n!  
    n @  
;
```

C 言語の記述

```
int calc( int q)  
{  
    n = q + 128;  
    return n;  
}
```

3.10.5.5 変数代入(!)

- (1)機能 16ビット値を変数へ代入する。
- (2)スタックダイアグラム
!(x addr — , x を addr へ代入する)
- (3)記述例

アダプタの記述

```
34 n !
```

C言語の記述

```
n = 34;
```

3.10.5.6 変数取り出し(@)

- (1)機能 変数から16ビット値をスタックトップに取り出す。
- (2)スタックダイアグラム
@(addr — x, addr から値を取り出す)
- (3)記述例

アダプタの記述

```
n @
```

C言語の記述

```
n;
```

3.10.5.7 1バイト代入(C!)

- (1)機能 1バイトデータをバイト配列に代入する。
- (2)スタックダイアグラム
C! (byte addr -, byte を addr へ代入する)
- (3)記述例

アダプタの記述

```
34 buf 3 + C!
```

C言語の記述

```
buf[3] = 34;
```

3.10.5.8 1バイト取り出し(C@)

- (1)機能 1バイトデータをスタックトップに取り出す。
- (2)スタックダイアグラム
C@ (addr -- byte, addr から値を取り出す)
- (3)記述例

アダプタの記述

```
buf 3 + C@
```

C言語の記述

```
Buf[3];
```

3.10.5.9 四則演算(+ - * / MOD)

(1)機能 加算、減算、乗算、除算、剰余算を行う。結果をスタックトップに格納する。

(2)スタックダイアグラム

+ (n1 n2 — add , n1 + n2 を計算する)
- (n1 n2 — sub , n1 - n2 を計算する)
* (n1 n2 — mul , n1 * n2 を計算する)
/ (n1 n2 — div , n1 / n2 を計算する)
MOD (n1 n2 — rem , n1 % n2 を計算する)

(3)記述例

アダプタの記述

```
1 1 +  
1 1 -  
1 1 *  
1 1 /  
1 1 MOD
```

C 言語の記述

```
1 + 1;  
1 - 1;  
1 * 1;  
1 / 1;  
1 % 1;
```

3.10.5.10 ビットのシフト(LSHIFT RSHIFT)

(1)機能 左シフト、右シフトを行う。結果をスタックトップに格納する。

(2)スタックダイアグラム

LSHIFT (n1 s ← n2 , n1 を s ビット左シフトする)

RSHIFT (n1 s ← n2 , n1 を s ビット右シフトする)

(3)記述例

アダプタの記述

n @ 4 LSHIFT	¥ n の左シフト
n @ 4 RSHIFT	¥ n の論理右シフト (n=FFFF のとき、演算結果は0FFF)

C言語の記述

n << 4
n >> 4

※論理右シフト：符号付きの場合、0が左から入る

3.10.5.1.1 比較演算子(= > < >= <= <>)

(1)機能 2 値を比較し、真なら TRUE を、偽なら FALSE をスタックトップに格納する。

(2)スタックダイアグラム

= (n1 n2 — flg , n1 = n2 なら、flg = TRUE)
> (n1 n2 — flg , n1 > n2 なら、flg = TRUE)
< (n1 n2 — flg , n1 < n2 なら、flg = TRUE)
>= (n1 n2 — flg , n1 >= n2 なら、flg = TRUE)
<= (n1 n2 — flg , n1 <= n2 なら、flg = TRUE)
<> (n1 n2 — flg , n1 <> n2 なら、flg = TRUE)

(3)記述例

アダプタの記述

```
2 2 =  
2 2 >  
2 2 <  
2 2 >=  
2 2 <=  
2 2 <>
```

C言語の記述

```
2 == 2  
2 > 2  
2 < 2  
2 >= 2  
2 <= 2  
2 != 2
```

(4) 注意事項

2 値の比較は、符号付き (signed) データとして行う。
符号なし (unsigned) データの比較は以下のように行う。

【プログラム】

(unsigned で「a」と「b」を比較し、「a」の方が大きければ TRUE、
等しいか「a」のほうが小さければ FALSE を返す)

```
a @ 0 < b @ 0 < AND a @ 0 >= b @ 0 >= AND OR @ TRUE = IF  
a @ b @ >  
ELSE  
b @ a @ >  
THEN
```

【解説】

unsigned の比較の場合、先頭ビットが 1 (signed では負) の方が、先頭
ビットが 0 (signed では正) よりも大きい。従って、符号が異なる数値
の比較では、「signed」と「unsigned」の大小関係は逆転する。また、
同符号のときは、「signed」と「unsigned」の大小関係は一致す
るため、上記のようなプログラムとなる。

3.10.5.12 真偽値(TRUE FALSE)

(1) 機能 真偽値をスタックトップに格納する。

(2) スタックダイアグラム

TRUE (— -1 , TRUE の値、-1)

FALSE (— 0 , FALSE の値 0)

(3) 記述例

アダプタの記述

```
2 TRUE =  
2 FALSE =
```

C 言語の記述

```
2 = true  
2 = false
```

3.10.5.13 論理演算子(AND OR NOT XOR)

(1)機能 論理演算を行い、結果をスタックトップに格納する。

(2)スタックダイアグラム

AND (n1 n2 -- n3, ビット単位の論理積)

OR (n1 n2 -- n3, ビット単位の論理和)

NOT (n1 -- n2, 全ビット反転)

XOR (n1 n2 -- n3, ビット単位の排他的論理和)

(3)記述例

アダプタの記述

```
0 1 AND
0 1 OR
n@NOT
0 1 XOR
```

C言語の記述

```
0 & 1
0 | 1
~n
0 ^ 1
```

3.10.5.14 条件文(IF～ELSE～THEN)

(1)機能 条件により、処理を分岐する。

(2)スタックダイアグラム

```
IF (flg ー , flg が TRUE の場合、以降のコードを実行する)
ELSE ( ー , flg が FALSE の場合、以降のコードを実行する)
THEN ( ー , IF..ELSE..を終了する)
```

(3)記述例

アダプタの記述(A)

```
n @ 128 = IF
    256 q !
ELSE
    64 q !
THEN
```

C言語の記述(A)

```
if(n==128){
    q = 256;
} else{
    q = 64;
}
```

アダプタの記述(B)

```
: SAMPLE_IF ( X ー )
    DUP 1 = IF 11 SWAP THEN
    DUP 2 = IF 12 SWAP THEN
    DUP 3 = IF 13 SWAP THEN
    DROP
;
```

C言語の記述(B)

```
int sample_if(x) {
    if (x == 1) return 11;
    if (x == 2) return 12;
    if (x == 3) return 13;
}
```

3.10.5.15 ループ(BEGIN~WHILE~REPEAT)

(1)機能 繰り返し処理する。

(2)スタックダイアグラム

BEGIN (— , ループを開始する)

WHILE (flg — , flg が TRUE の間、ループの実行を続ける)

REPEAT (— , ループを終了する)

(3)記述例

アダプタの記述

```
VARIABLE q
VARIABLE n
0 q !
10 n !
BEGIN
n @ 0 > WHILE
    q @ n @ + q !
    n @ 1 -
    n !
REPEAT
```

C言語の記述

```
int q;
int n;
q=0;
n=10;
while( n>0 ){
    q = q + n;
    n--;
}
```

3.10.5.16 基数の変換(HEX DECIMAL BINARY)

(1)機能 基数を設定する。

(2)スタックダイアグラム

HEX (— , 以降のデータを16進数として扱う)
DECIMAL (— , 以降のデータを10進数として扱う)
BINARY (— , 以降のデータを2進数として扱う)

(3)記述例

アダプタの記述

HEX	¥ 16進数に
DECIMAL	¥ 10進数に
BINARY	¥ 2進数に

(4)注意事項

デフォルトは、DECIMAL(基数10)。
プログラム中の任意の場所で記述することが可能である。

3.10.5.17 スタック操作(DUP PICK DROP SWAP ROLL)

(1)機能 スタックを操作する。

(2)スタックダイアグラム

DUP (n ← n n , スタックトップを複製する)

PICK (xn ... x1 x0 n ← xn ... x1 x0 xn, n 番目のスタックをコピーする)

DROP (n ← , スタックトップを削除する)

SWAP (n1 n2 ← n2 n1 , 上から2つを入れ替える)

ROLL (xn ... x1 x0 n ← xn-1 ... x1 x0 xn, n 番目のスタックを回転する)

(3)記述例

アダプタの記述

2 DUP	¥ 2	→ 2 2
3 2 1 2 PICK	¥ 3 2 1	→ 3 2 1 3
2 DROP	¥ 2	→
2 1 SWAP	¥ 2 1	→ 1 2
3 2 1 2 ROLL	¥ 3 2 1	→ 2 1 3

3.10.5.18 コメント(¥ (...))

(1)機能 ソースコード上にコメントを記述する。インタプリタで解釈しない。

(2)スタックダイアグラム

¥ (ー , 以降コメントして扱う)

((ー , コメントを開始する)

) (ー , コメントを終了する)

(3)記述例

アダプタの記述

```
¥ この行はコメントです。  
2 3 ( ここもコメントです ) +
```

C言語の記述

```
// この行はコメントです。  
2 + /* ここもコメントです */ 3
```

3.10.5.19 LONG型データ演算(OP_LONG)

(1)機能 long 型(4バイトデータ)の演算を行う。

(2)スタックダイアグラム

OP_LONG (addr1 [addr2 | n] addr3 type — result)

(3)説明

addr1 : 演算対象のデータを格納するアドレス1
addr2 : 演算対象のデータを格納するアドレス2(NOT の場合は不要)
n : LSHIFT、RSHIFT 時のシフトビット数
addr3 : 演算結果のデータを格納するアドレス
type : 演算の種別

1	(+)
2	(-)
3	(*)
4	(/)
5	(MOD)
6	(LSHIFT)
7	(RSHIFT)
8	(AND)
9	(OR)
10	(XOR)
11	(NOT)
12	(=)
13	(<)
14	(<=)
15	(>)
16	(>=)
17	(<>)

result : 演算結果のデータを格納するアドレス(addr3 と同じ)

(4)注意事項

addr1、addr2、addr3、及びresult は、いずれも先頭から4バイト目までに、ビッグエンディアンで、long 型の値を格納する。
TRUE は、0xffffffff、FALSE は0x00000000 で表す。

3.10.5.20 LONG値の真偽値への変換(CONV_ADDR_TO_TF)

(1)機能 4バイトデータを2バイトのTRUEまたはFALSEの値に変換する。

(2)スタックダイアグラム

CONV_ADDR_TO_TF (addr — val)

(3)説明

addr	: 4バイト値を格納したアドレス
val	: 0 (FALSE: addrの先頭4バイトが全て0の場合)
	-1 (TRUE: addrの先頭4バイトの何れかが0でない場合)

3. 1 0. 6 インタプリタ ECHONET Lite API仕様

3.10.6.1 INIT_ECHONET

(1)機能 ECHONET Lite 通信処理部の初期化を行う

(2)スタックダイアグラム

INIT_ECHONET (mode nretry — result)

(3)説明

mode	:	スタートモード
		0 (ウォームスタート)
		1 (コールドスタート(1))
		2 (コールドスタート(2))
		3 (コールドスタート(3))
nretry	:	初期化失敗時のリトライ回数
result	:	結果
		0 (正常終了)
		-1 (異常終了)

(4)注意事項

- ・ECHONET Lite 通信処理部の初期化、動作開始処理を行う。
- ・本APIは、ECHONET オブジェクトの初期化、変換テーブルの初期化後に1度だけ呼ばれる。

3.10.6.2 SET_COM_PARAM

(1)機能 ECHONET Lite ミドルウェアアダプタ通信インタフェースのパラメータ設定を行う。本APIでは、エラー時のリトライ待機時間[ms]を設定する。

(2)スタックダイアグラム

SET_COM_PARAM (r_time — , 通信のパラメータ設定を行う)

(3)説明 r_time : エラー時のリトライ待機時間[ms] (デフォルト値1000ms)

(4)注意事項

3.10.6.3 SET_UART_RV_MODE

(1)機能 ECHONET Lite レディ機器からの受信電文の終了判定条件を設定する

(2)スタックダイアグラム

SET_UART_RV_MODE ((addr) (val | len) mode —)

(3)説明

(addr)	: mode=2 時の終了コードの配列アドレス mode=1 の場合は、省略可能
val	: mode=1 時の受信終了判定時間(ミリ秒)
len	: mode=2 時の終了コードのコード長
mode	: 終了判定モード 1 (時間による判定モード) 2 (終了コードによる判定モード)

(4)注意事項

- ・ (addr) は、mode=1 の場合は不要
- ・ (val | len) は、mode= 1 の場合は、val として、mode=2 の場合は、len として指定する。

3.10.6.4 CREATE_MNG_TABLES

(1)機能 オブジェクト情報、プロパティ情報、プロパティ関連情報を格納する管理テーブルを作成する

(2)スタックダイアグラム

CREATE_MNG_TABLES (n_obj n_ipc n_epc n_epcm n_irel n_mrel n_frel —)

(3)説明

n_obj	: 家電機器オブジェクトの数
n_ipc	: 家電機器側定義プロパティの数
n_epc	: 配列でない ECHONET プロパティの数
n_epcm	: 配列の ECHONET プロパティの数
n_irel	: 同一値型プロパティ関係の数
n_mrel	: マッピング型プロパティ関係の数
n_frel	: 関数型プロパティ関係の数

(4)注意事項

- ・ 本 API は、オブジェクト情報、プロパティ情報、プロパティ関連情報を登録する API (RGST_XXX) を実行する前に一度だけ呼ばれる。

3.10.6.5 RGST_NODE

- (1)機能 他ノードを登録する
- (2)スタックダイアグラム
RGST_NODE (node_id addr_buf addr_size —)
- (3)説明
- | | |
|-----------|--------------------------|
| node_id | : ノードID |
| addr_buf | : 通信アドレスが格納されているバッファアドレス |
| addr_size | : 通信アドレスのバイト数 |
- (4)注意事項
- ・引数 **node_id** は、ダウンロードプログラム内でノードを一意に識別するためのID。
 - ・自ノードは、インタプリタ起動時に、**node_id=0** にて自動生成される。

3.10.6.6 RGST_OBJ

- (1)機能 中間オブジェクトを登録する
- (2)スタックダイアグラム
RGST_OBJ (obj_id node_id obj_class_group obj_class instance —)
- (3)説明
- | | |
|-----------------|--------------------------|
| obj_id | : 中間オブジェクトID |
| node_id | : ECHONET Lite ノードID |
| obj_class_group | : ECHONET オブジェクトのクラスグループ |
| obj_class | : ECHONET オブジェクトのクラス |
| instance | : ECHONET オブジェクトのインスタンス |
- (4)注意事項
- ・自ノードへオブジェクトを登録する場合は、**node_id=0** を指定する。
 - ・ノードプロファイルオブジェクトは、インタプリタ起動時に、**obj_id=0** にて自ノード(**node_id=0**)へ自動生成される。
 - ・本API コール時にノードプロファイルオブジェクトの
 - ・自ノードインスタンスリスト **S**
 - ・自ノードクラスリスト **S**
 - ・自ノードインスタンス数
 - ・自ノードクラス数
 - ・自ノードインスタンスリスト
 - ・自ノードクラスリストの各プロパティを設定する。

3.10.6.7 RGST_EPC

(1)機能 配列でない ECHONET プロパティを登録する

(2)スタックダイアグラム

RGST_EPC (obj_id epc type rule anno keep_edt size —)

(3)説明

obj_id	: 中間オブジェクト ID
epc	: ECHONET プロパティコード(EPC)
type	: プロパティの型
	0 (signed char)
	1 (signed short)
	2 (signed long)
	3 (unsigned char)
	4 (unsigned short)
	5 (unsigned long)
	6 (データ型なし)
rule	: 処理可能なアクセスルール(以下を OR 指定する)
	0x0001 (Set)
	0x0002 (Get)
	0x0004 (Anno)
anno	: 状態時アナウンス有無
	1 (アナウンス有)
	0 (アナウンス無)
keep_edt	: プロパティ値保持フラグ
	1 (保持する)
	0 (保持しない)
size	: データエリアサイズ(バイト数)

(4)注意事項

- ・ ノードプロファイルオブジェクト(obj_id=0)の必須プロパティと、異常内容プロパティ(0x89)は、インタプリタ起動時に自動生成される。
- ・ 対象オブジェクトのプロパティマップの設定も行う。
- ・ rule と ESV の関係
 - Set : SetI(0x60)、SetC(0x61)
 - Get : Get(0x62)
 - Anno : INF_REQ(0x63)
- ・ 引数、keep_edt は、オブジェクト生成方式の、”アダプタ内に値を保持しないサービス”用に設定した。保持しない、に設定したプロパティは、ミドルウェアで折り返さないの、すべての要求が、CHK_RV_IPC にて上がる。保持する、に設定したプロパティは、Set 要求のみが、CHK_RV_IPC にて上がる。

3.10.6.8 RGST_EPCM

(1)機能 配列の ECHONET プロパティを登録する

(2)スタックダイアグラム

RGST_EPCM (obj_id epc type rule anno keep_edt size member_no —)

(3)説明

obj_id	: 中間オブジェクト ID
epc	: ECHONET プロパティコード(EPC)
type	: プロパティの型
	0 (signed char)
	1 (signed short)
	2 (signed long)
	3 (unsigned char)
	4 (unsigned short)
	5 (unsigned long)
	6 (データ型なし)
rule	: 処理可能なアクセスルール(以下を OR 指定する)
	0x0001 (要素指定 Set)
	0x0002 (要素指定 Get)
	0x0010 (要素指定存在確認要求)
	0x0040 (要素指定通知要求)
anno	: 状態時アナウンス有無
	1 (アナウンス有)
	0 (アナウンス無)
keep_edt	: プロパティ値保持フラグ
	1 (保持する)
	0 (保持しない)
size	: データエリアサイズ(バイト数)
member_no	: 登録する要素番号

(4)注意事項

- ・対象オブジェクトのプロパティマップの設定も行う。
- ・rule と ESV の関係

要素指定 Set	: SetMI (0x64)、SetMC (0x65)
要素指定 Get	: GetM (0x66)
要素指定存在確認要求	: CheckM (0x6c)
要素指定通知要求	: INFM_REQ (0x67)

3.10.6.9 ADD_EPC_MEMBER

(1)機能 RGST_EPCM で登録した配列の ECHONET プロパティに要素番号を指定

して配列要素を追加する

(2)スタックダイアグラム

ADD_EPC_MEMBER (obj_id epc member_no —)

(3)説明 obj_id : 中間オブジェクト ID
 epc : ECHONET プロパティコード(EPC)
 member_no : 追加する要素番号

(4)注意事項

3.10.6.10 RGST_IPC

(1)機能 中間オブジェクトプロパティを登録する

(2)スタックダイアグラム

RGST_IPC (obj_id ipc type size —)

(3)説明 obj_id : 中間オブジェクト ID
 ipc : 中間オブジェクトプロパティコード(IPC)
 type : プロパティの型
 0 (signed char)
 1 (signed short)
 2 (signed long)
 3 (unsigned char)
 4 (unsigned short)
 5 (unsigned long)
 6 (データ型なし)
 size : データエリアサイズ (バイト数)

(4)注意事項

3.10.6.1.1 RGST_IDENTICAL_PROP

(1)機能 同一値型変換テーブルを登録する

(2)スタックダイアグラム

RGST_IDENTICAL_PROP (ipc (ele) epc obj_id —)

(3)説明

ipc	: 中間オブジェクトプロパティコード(IPC)
(ele)	: ECHONET プロパティの要素番号 ECHONET プロパティが配列要素の場合のみ指定する
epc	: ECHONET プロパティコード(EPC)
obj_id	: 中間オブジェクト ID

(4)注意事項

3.10.6.12 RGST_MAP_PROP_REL

(1)機能 マッピング型変換テーブルを作成する

(2)スタックダイアグラム

RGST_MAP_PROP_REL (map_id ipc0...ipcn (ele0) epc0... (elem) epcm nmparel
 nipc nepc
 obj_id --)

(3)説明

map_id : 作成するマッピング関係 ID
 ipc0...ipcn : n組の中間オブジェクトプロパティコード
 (ele0)epc0... (elem) epcm : m組の ECHONET プロパティコードと要素番号
 要素番号は、ECHONET プロパティが配列型の場合のみ指定する
 nmparel : 作成するマッピングレコードの数
 nipc : 中間オブジェクトプロパティの数
 nepc : ECHONET プロパティの数
 obj_id : 中間オブジェクト ID

(4)注意事項

- ・本関数は、マッピング型変換テーブルを作成するのみである。
- ・テーブルデータの生成は、RGST_MAP_PROP_VAL, RGST_MAP_PROP_VAL_PR にて行う。

nipc		nepc			
中間オブジェクトID	MapID	IPC0...IPCn IDT0...IDTn	IFLG	EPC0:ELE0...EPCm:ELEm EDT0...EDTm	EFLG
					} nmaprel

3.10.6.13 RGST_MAP_PROP_VAL

(1)機能 マッピング型変換テーブルにプロパティの対応を登録する

(2)スタックダイアグラム

RGST_MAP_PROP_VAL ((val0...vall) idt0...idtn edt0...edtm map_id —)

(3)説明 (val0...vall) : エスケープ時のプロパティ値
 idt, edt がエスケープされたときの値。val=0x0000の場合、NO CARE を表す。val=0xFFFF の場合、idt 値、edt 値が0xFFFF であることを表す。
 (val0...vall) が並ぶ順番は、idt0...idtn edt0...edtm のうち、値が 0xFFFF であるものの順に対応している。
 エスケープコードを使用しない場合は、省略可能。
 idt0...idtn : n組の中間オブジェクトプロパティ値又は、プロパティ値を格納している配列のアドレス値が0xFFFFの時はエスケープコードとみなし、対応するエスケープ時のプロパティ値を参照する。
 edt0...edtm : m組の ECHONET プロパティコード値又は、プロパティを格納している配列のアドレス値が0xFFFFの時はエスケープコードとみなし、対応するエスケープ時のプロパティ値を参照する。
 map_id : 登録するマッピング型変換テーブルの ID
 RGST_MAP_PROP_REL で作成した ID を指定する

(4)注意事項 ・NO CARE の使い方
 下表のような場合は、EPC0=2 であれば、EPC1 の値には関係なく (NOCARE)、
 IPC0=3 とする。その他は、ECP0 と EPC1 の両方の値により、IPC0 の値
 を関連付ける。

中間オブジェクトID	MapID	IPC0	IFLG	EPC0	EPC1(val)	EFLG
1	1	1	1	1	1	1
		2	1	1	2	1
		3	1	2	0xFFFF(0x0000)	1
		4	1	3	1	1
		5	1	3	2	1

val=0xFFFF の場合は、EPC0=2、EPC1=0xFFFF が IPC0=3 に変換される。

中間オブジェクトID	MapID	IPC0	IFLG	EPC0	EPC1(val)	EFLG
1	1	1	1	1	1	1
		2	1	1	2	1
		3	1	2	0xFFFF(0xFFFF)	1
		4	1	3	1	1
		5	1	3	2	1

- 変換の条件が複数行にマッチする場合は、先に現れた方が優先される。
- unsigned char 型(1バイト)の IPC と、 unsigned long 型(4バイト)の EPC のマッピング関係を登録する場合

HEX

CREATE LONG_EPC 4 ALLOT

12 LONG_EPC 0 + C!

34 LONG_EPC 1 + C!

56 LONG_EPC 2 + C!

78 LONG_EPC 3 + C!

0 1 5 4 RGST_EPC ¥ unsigned long, 4 bytes

.....

41 LONG_EPC 1 RGST_MAP_PROP_VAL

.....

3.10.6.14 RGST_MAP_PROP_VAL_PR

(1)機能 対応が一意に決まらないとき、どの対応を優先させるかを示すフラグを付与して、マッピング変換テーブルにプロパティの対応を登録する。

(2)スタックダイアグラム

```
RGST_MAP_PROP_VAL_PR ( (val0...vall) i_pflg e_pflg idt0...idtn edt0...edtm  
map_id  
— )
```

(3)説明 (val0...vall) : エスケープ時のプロパティ値
idt, edt がエスケープされたときの値。val=0x0000
の場合、NO CARE を表す。val=0xFFFF の場合、
idt 値、edt 値が 0xFFFF であることを表す。
(val0...vall) が 並ぶ 順 番 は、 idt0...idtn
edt0...edtm のうち、値が 0xFFFF であるものの順
に対応している。
エスケープコードを使用しない場合は、省略可能。

i_pflg : IPC→EPC 変換時の優先フラグ
(複数の対応付けがある場合、1 を付与した値を優先
する)

e_pflg : EPC→IPC 変換時の優先フラグ
(複数の対応付けがある場合、1 を付与した値を優先
する)

idt0...idtn : n組の中間オブジェクトプロパティ値又は、プロパ
ティ値を格納している配列のアドレス値が 0 x FFFF
の時はエスケープコードとみなし、対応するエスケ
ープ時のプロパティ値を参照する。

edt0...edtm : m組の ECHONET プロパティコード値又は、プロ
パティ値を格納している配列のアドレス値が 0 x
FFFF の時はエスケープコードとみなし、対応する
エスケープ時のプロパティ値を参照する。

map_id : 登録するマッピング型変換テーブルの ID
RGST_MAP_PROP_REL で作成した ID を指定す
る

(4)注意事項

- ・ NO CARE の使い方
下表のような場合は、EPC0=2 であれば、EPC1 の値には関係なく
(NOCARE)、IPC0=3 とする。その他は、ECP0 と EPC1 の両方の値
により、IPC0 の値を関連付ける。

中間オブジェクトID	MapID	IPC0	IFLG	EPC0	EPC1(val)	EFLG
1	1	1	1	1	1	1
		2	1	1	2	1
		3	1	2	0xFFFF(0x0000)	1
		4	1	3	1	1
		5	1	3	2	1

val=0xFFFF の場合は、EPC0=2、EPC1=0xFFFF が IPC0=3 に変換される。

中間オブジェクトID	MapID	IPC0	IFLG	EPC0	EPC1(val)	EFLG
1	1	1	1	1	1	1
		2	1	1	2	1
		3	1	2	0xFFFF(0xFFFF)	1
		4	1	3	1	1
		5	1	3	2	1

・i_pflg, e_pflg の使い方

下表の場合、EDT=A に対応する IDT は、1, 2, 3 があるが、EFLG の立っている 1 が優先して対応付けされる。

中間オブジェクトID	MapID	0x22(=IPC)	IFLG	0xAA(=EPC)	EFLG
1	1	1	1	A	1
		2	1	A	0
		3	1	A	0
		4	1	B	0
		5	1	B	1
		6	1	B	0
		7	1	C	0
		8	1	C	1

・IF ELSE のような使い方

下表の場合は、EPC0=2 の時、EPC1=1 以外は、IPC0=5 となる。

中間オブジェクトID	MapID	IPC0	IFLG	EPC0	EPC1	EFLG
1	1	1	1	1	1	1
		2	1	1	2	1
		3	1	1	3	1
		4	1	2	1	1
		5	1	2	-	1
		6	1	3	1	1
		7	1	3	2	1
		8	1	3	3	1

・変換の条件が複数行にマッチする場合は、先に現れた方が優先される。

・unsigned char 型(1 バイト)の IPC と、unsigned long 型(4 バイト)の EPC のマッピング関係を登録する場合

HEX

CREATE LONG_EPC 4 ALLOT

12 LONG_EPC 0 + C!

34 LONG_EPC 1 + C!

56 LONG_EPC 2 + C!

78 LONG_EPC 3 + C!

0 1 5 4 RGST_EPC ¥ unsigned long, 4 bytes

.....

41 LONG_EPC 1 RGST_MAP_PROP_VAL_PR

.....

3.10.6.15 RGST_FUNC_PROP

(1)機能 関数型変換テーブルにプロパティの対応を登録する

(2)スタックダイアグラム

RGST_FUNC_PROP (idt2edt_func edt2idt_func ipc0...ipcn (ele0) epc0... (elem)
 epcm
 nipc nepc obj_id -)

(3)説明

idt2edt_func : 中間→ECHONET プロパティ値に変換する関数名
 edt2idt_func : ECHONET Lite→中間プロパティ値に変換する関
 数名
 ipc0...ipcn : n組の中間オブジェクトプロパティコード
 (ele0)epc0... (elem)epcm
 : m組の ECHONET プロパティコードと要素番号
 要素番号(ele)は、ECHONET プロパティが配列型
 の場合のみ指定する
 nipc : 中間オブジェクトプロパティの数
 nepc : ECHONET プロパティの数
 obj_id : 中間オブジェクト ID

(4)注意事項

- ・ idt2edt_func の記述方法(変換関数名が "I2E"の場合)

C" I2E" FIND DROP

- ・ edt2idt_func の記述方法(変換関数名が "E2I"の場合)

C" E2I" FIND DROP

- ・ nipc, nepc の意味

	nipc	nepc		
	┌──────────┴──────────┐			
中間オブジェクトID	IPC0...IPCn	EPC0:ELE0...EPCm:ELEm	I2E_FUNC	E2I_FUNC

3.10.6.16 SET_IPC

(1)機能 中間オブジェクトプロパティの値を指定して、対応する EPC に値を書き込む。状態変化通知処理設定のある場合、状態通知サービスを行う

(2)スタックダイアグラム

SET_IPC (idt ipc obj_id --)

(3)説明

idt	:	中間オブジェクトプロパティ値又は、プロパティ値が格納されている配列のアドレス
ipc	:	中間オブジェクトプロパティコード
obj_id	:	中間オブジェクト ID

(4)注意事項

- obj_id=0 を指定することにより、ノードプロファイルオブジェクトにアクセスが可能である。
- unsigned long 型の IPC に、値 0x12345678 を書き込む場合

```
HEX
CREATE LONG_IPC 4 ALLOT
12 LONG_IPC 0 + C!
34 LONG_IPC 1 + C!
56 LONG_IPC 2 + C!
78 LONG_IPC 3 + C!
0 1 5 4 RGST_IPC      ¥ unsigned long, 4 bytes
.....
LONG_IPC 1 0 SET_IPC¥ ipc=1 , obj_id=0
.....
```

3.10.6.17 SET_SEND_IPC

(1)機能 中間オブジェクトプロパティの値を指定して対応する ECHONET プロパティに値を書き込み、ECHONET Lite に送信する

(2)スタックダイアグラム

SET_SEND_IPC (idt ipc obj_id isv dst_id —)

(3)説明

idt : 中間オブジェクトプロパティ値又は、プロパティ値が格納されている配列のアドレス

ipc : 中間オブジェクトプロパティコード

obj_id : 中間オブジェクト ID

isv : 中間オブジェクトサービスコード(0x00**)
 上位 4 ビットと下位 4 ビットの組み合わせでサービスを指定する

上位 4 ビット : 0x1* (設定、応答不要)
 0x2* (設定、応答要)
 0x3* (取得)
 0x4* (通知)

下位 4 ビット : 0x*1 (要求)
 0x*2 (応答、応答不要)
 0x*3 (応答、応答要)
 0x*4 (不可応答)

dst_id : 送信先アドレスを表す ID または同報種別
 送信先アドレス ID : 0x0000 - 0x7fff
 同報種別 : 0x8000 - 0xffff

(4)注意事項

- obj_id=0 を指定することにより、ノードプロファイルオブジェクトにアクセスが可能である。
- isv と ESV の関係 (要素でない EPC の場合)

上位	下位			
	0x*1	0x*2	0x*3	0x*4
0x1*	SetI	-	-	SetI_SNA
0x2*	SetC	Set Res	-	SetC_SNA
0x3*	Get	Get Res	-	Get_SNA
0x4*	INF_REQ	INF	INFC	INF_SNA

・ isv と ESV の関係 (要素 EPC の場合)

上位	下位			
	0x*1	0x*2	0x*3	0x*4
0x1*	SetMI	-	-	SetMI_SNA
0x2*	SetMC	SetM_Res	-	SetMC_SNA
0x3*	GetM	GetM_Res	-	GetM_SNA
0x4*	INFM_REQ	INFM	INFC	INFM_SNA

- isv は、表中ハッチング部分を必須とする。
- dst_id で個別アドレスを指定する場合は、CHK_RV_IPC の src_id_buf を代入する。

同報アドレスは、ECHONET Lite の同報種別指定コード+同報対照指定コードの下位 15 ビットを用いる。

- unsigned long 型の IPC に、値 0x12345678 を書き込む場合

HEX

CREATE LONG_IPC 4 ALLOT

12 LONG_IPC 0 + C!

34 LONG_IPC 1 + C!

56 LONG_IPC 2 + C!

78 LONG_IPC 3 + C!

0 1 5 4 RGST_IPC ¥ unsigned long, 4 bytes

.....

LONG_IPC 1 0 SET_IPC ¥ ipc=1 , obj_id=0

.....

3.10.6.18 CHK_RV_IPC

(1)機能 ECHONET Lite からの受信により、値が変化した中間オブジェクトのプロパティを確認する

(2)スタックダイアグラム

CHK_RV_IPC (buf_num src_id_buf obj_id_buf ipc_buf rv_code_buf — nipc)

(3)説明

buf_num : バッファ最大要素数

src_id_buf : インタプリタ内部で管理する送信元アドレスを表す ID を格納するバッファアドレス

obj_id_buf : 受信のあったプロパティに対応する中間オブジェクト ID を格納するバッファアドレス

ipc_buf : 受信のあった中間オブジェクトプロパティコードを格納するバッファアドレス

rv_code_buf : 受信のあったサービスを格納するバッファアドレス
 中間オブジェクトサービスコード(0x00**)を格納。
 上位4ビットと下位4ビットの組み合わせでサービスを指定する

上位4ビット : 0x1* (設定、応答不要)
 0x2* (設定、応答要)
 0x3* (取得)
 0x4* (通知)

下位4ビット : 0x*1 (要求)
 0x*2 (応答、応答不要)
 0x*3 (応答、応答要)
 0x*4 (不可応答)

nipc : 値が変化した中間オブジェクトプロパティの数

(4)注意事項

- obj_id_buf, ipc_buf, rv_code_buf の各バッファは、ダウンロードプログラム上に確保する。本APIをコールすることで、上記バッファに値の変化した中間オブジェクトID、プロパティコードが格納される。
- rv_code_buf と ESV の関係
 - IPC が要素でない EPC の場合、

上位	下位			
	0x*1	0x*2	0x*3	0x*4
0x1*	SetI	-	-	SetI_SNA
0x2*	SetC	Set_Res	-	SetC_SNA
0x3*	Get	Get_Res	-	Get_SNA
0x4*	INF_REQ	INF	INFC	INF_SNA

- IPC が要素 EPC の場合、

上位	下位			
	0x*1	0x*2	0x*3	0x*4
0x1*	SetMI	—	—	SetMI_SNA
0x2*	SetMC	SetM_Res	—	SetMC_SNA
0x3*	GetM	GetM_Res	—	GetM_SNA
0x4*	INFM_REQ	INFM	INFMC	INFM_SNA

・rv_code_buf は、表中ハッチング部分を必須とする。

3.10.6.19 GET_IPC

(1)機能 中間オブジェクトプロパティコード(IPC)の値を、対応する ECHONET プロパティコード(EPC)の値から変換して読み込む

(2)スタックダイアグラム

GET_IPC ((idt_buf) ipc obj_id — idt)

(3)説明 (idt_buf) : 中間オブジェクトプロパティ値を格納するバッファのアドレス。
 読み込むプロパティが配列の場合のみ指定する。
 ipc : 中間オブジェクトプロパティコード
 obj_id : 中間オブジェクト ID
 idt : 中間オブジェクトプロパティ値。
 プロパティが配列の場合は、引数で指定したプロパティ値を格納するバッファのアドレス。

(4)注意事項

- ・配列の場合に使用する idt_buf バッファは、ダウンロードプログラム上に確保する。
 本APIをコールすることで、上記バッファにipcで指定したプロパティ値が格納される。
- ・obj_id=0を指定することにより、ノードプロファイルオブジェクトにアクセスが可能である。
- ・unsigned long 型の IPC に、値を読み込む場合

```

HEX
CREATE LONG_IPC 4 ALLOT
0 1 5 4 RGST_IPC      ¥ unsigned long, 4 bytes
.....
LONG_IPC 1 0 GET_IPC      ¥ ipc=1 , obj_id=0
.....
  
```

配列 LONG_IPC に最新値が更新される。

3.10.6.20 FROM_EQUIPMENT

(1)機能 家電機器インタフェースからデータを受信する

(2)スタックダイアグラム

FROM_EQUIPMENT (rv_buf buf_size time_out — rv_code)

(3)説明

rv_buf	: 受信データを格納するバッファアドレス
buf_size	: 受信データを格納するバッファのサイズ
time_out	: タイムアウト時間[ms] (0 を指定することでノンブロッキングモード)
rv_code	: 受信コード
-1	: (受信失敗)
0	: (受信データなし)
>0	: (受信データバイト数)

(4)注意事項

- ・rv_buf バッファは、ダウンロードプログラム上に確保する。本 API をコールすることで、上記バッファに家電機器インタフェースから受信したデータが格納される。

3.10.6.21 TO_EQUIPMENT

(1)機能 家電機器インタフェースへデータを送信する

(2)スタックダイアグラム

TO_EQUIPMENT (tr_buf dat_size b_flg —)

(3)説明

tr_buf	: 送信データが格納されているバッファアドレス
dat_size	: 送信データバイト数
b_flg	: ブロッキングモード指定フラグ
1	: (ブロッキングモード)
0	: (ノンブロッキングモード)

(4)注意事項

- ・tr_buf バッファは、ダウンロードプログラム上に確保する。本 API をコールすることで、上記バッファに格納されているデータが家電機器インタフェースから送信される。

3.10.6.22 SET_BUF

(1)機能 バッファへデータをセットする

(2)スタックダイアグラム

SET_BUF (dat0...datn tr_buf dat_size --)

(3)説明 dat0...datn : 格納するデータ列
tr_buf : データを格納するバッファのアドレス
dat_size : 格納するデータバイト数

(4)注意事項

3.10.6.23 SLEEP

(1)機能 指定時間処理を止めて待機する

(2)スタックダイアグラム

SLEEP (s_time --)

(3)説明 s_time : 待機時間[ms]

(4)注意事項

3.10.6.24 SET_TIMER

(1)機能 システムタイマに時間をセットする(0~327670ms)

(2)スタックダイアグラム

SET_TIMER (time --)

(3)説明 time : 時間[10ms]

(4)注意事項

3.10.6.25 GET_TIMER

(1)機能 システムタイマの値を取り出す(0~327670ms)

(2)スタックダイアグラム
GET_TIMER (-- time)

(3)説明 time : 時間[10ms]

(4)注意事項

3.10.6.26 INDICATE_STATUS

(1)機能 表示部に、エラー状態を表示する

(2)スタックダイアグラム
INDICATE_STATUS (status —)

(3)説明 status : 状態
1 : エラー発生
0 : エラー解除

(4)注意事項

3.10.6.27 STOP

(1)機能 ECHONET Lite ミドルウェアアダプタの動作を停止する

(2)スタックダイアグラム
STOP (—)

(3)説明 なし

(4)注意事項

3. 10. 7 プログラム圧縮・伸張仕様

3.10.7.1 圧縮プログラムの概要

プログラム圧縮は、テキスト形式のプログラムをバイナリ形式のプログラムに変換することにより、プログラムサイズを縮小するための処理である。ECHONET Lite レディ機器メーカーがプログラム開発時に圧縮処理を行う。圧縮されたプログラムはミドルウェアアダプタ内のインタプリタ内で伸張され、実行される。

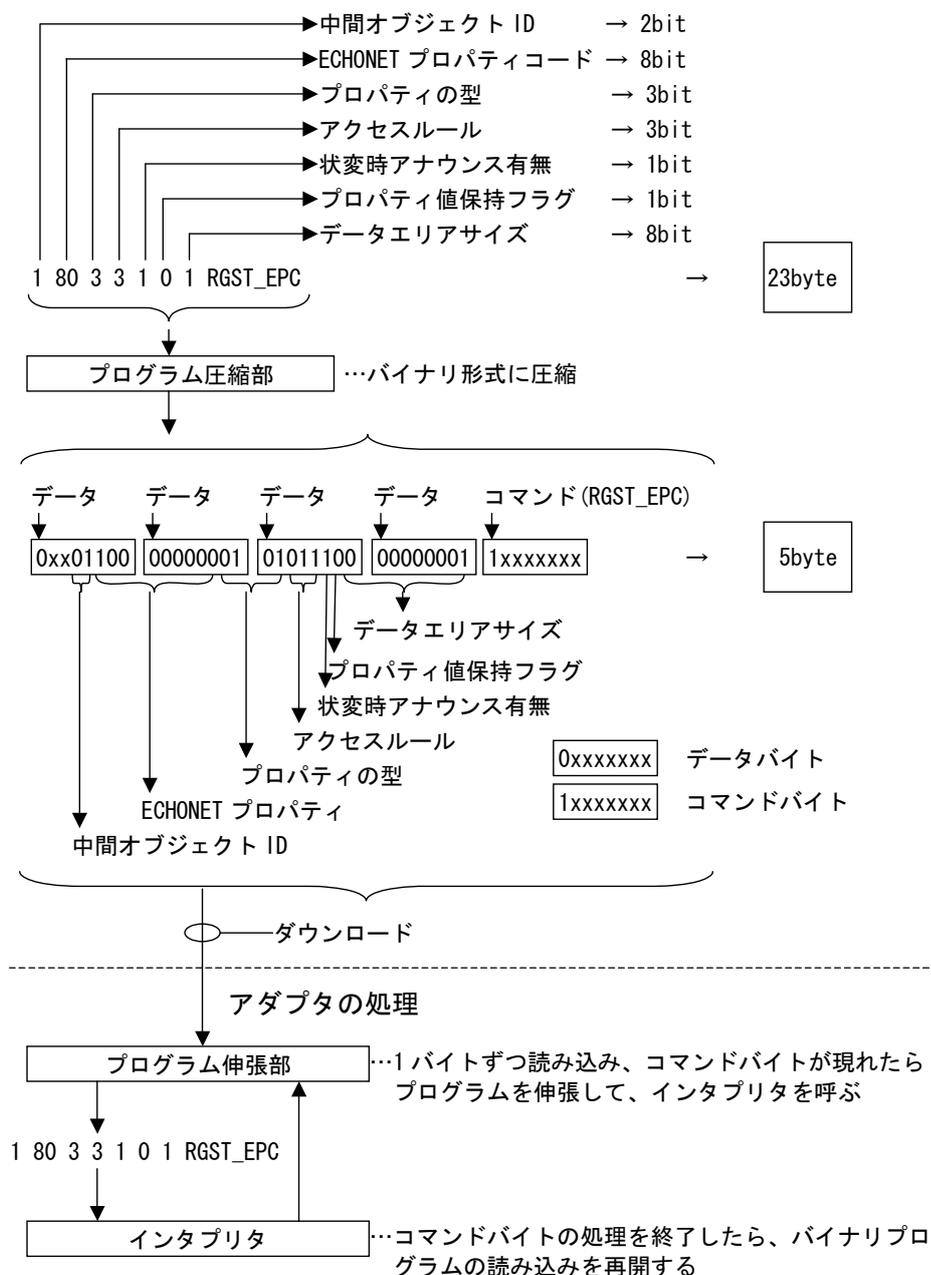


図3-42 プログラム圧縮処理の概要

図3-42は、RGST_EPC を例にした圧縮処理の概要を表した図である。RGST_EPC

の入力となるデータの情報は図に示す通りである(中間オブジェクト ID は、実際には4ビットであるが、図では2ビットとしている)。プログラム圧縮処理では、各入力データに、情報量に従ったビット数を割り当てて値を格納したデータを作成し、バイトの2ビット目以下に配置する。各バイトの先頭のビットは、データを表すバイト(データバイト)であるか、コマンドを表すバイト(コマンドバイト)であるかを判別するフラグとして用いる。圧縮したプログラムはスタックのトップからデコードできるように、データバイト中のビット列は、右詰めにする。

3.10.7.2 バイトの仕様

(1) コマンドバイト

コマンドバイトは、先頭ビットが1のバイトである。残りの7ビットで、コマンドの種類を表す。コマンドバイトには、固定コマンドバイトと、ユーザ定義コマンドバイトの2種類がある。

(a) 固定コマンドバイト

インタプリタが提供する API コマンドに対応して予め、コードを定義されたコマンドバイトである。

(b) ユーザ定義コマンドバイト

「VARIABLE」、「:」または「CREATE」によって、ユーザが新たな変数、関数、または配列を定義したときに、定義された文字列に対して割り当てられるコマンドバイトである。定義された順に、「0xCD」から、最大50個まで作成できる。

以下の表に、インタプリタが提供する API コマンドとコマンドバイトとの対応を示す。

表3-14 APIコマンドとコマンドバイトの対応表

API コマンド	コマンドバイト	API コマンド	コマンドバイト
VARIABLE	0x80	INIT_ECHONET	0xA7
:	0x81	SET_COM_PARAM	0xA8
;	0x82	SET_UART_RV_MODE	0xA9
CREATE	0x83	CREATE_MNG_TABLES	0xAA
ALLOT	0x84	RGST_OBJ	0xAB
!	0x85	RGST_EPC	0xAC
@	0x86	RGST_EPCM	0xAD
C!	0x87	ADD_EPC_MEMBER	0xAE
C@	0x88	RGST_IPC	0xAF
+	0x89	RGST_IDENTICAL_PROP	0xB0
-	0x8A	RGST_MAP_PROP_REL	0xB1
*	0x8B	RGST_MAP_PROP_VAL	0xB2
/	0x8C	RGST_MAP_PROP_VAL_PR	0xB3

MOD	0x8D	RGST_FUNC_PROP	0xB4
LSHIFT	0x8E	SET_IPC	0xB5
RSHIFT	0x8F	SET_SEND_IPC	0xB6
AND	0x90	CHK_RV_IPC	0xB7
OR	0x91	GET_IPC	0xB8
XOR	0x92	FROM_EQUIPMENT	0xB9
NOT	0x93	TO_EQUIPMENT	0xBA
TRUE	0x94	SLEEP	0xBB
FALSE	0x95	SET_BUF	0xBC
=	0x96	SET_TIMER	0xBD
<	0x97	GET_TIMER	0xBE
<=	0x98	RGST_NODE	0xBF
>	0x99	INDICATE_STATUS	0xC0
>=	0x9A	STOP	0xC1
◇	0x9B	RESET	0xC2
IF	0x9C	END_OF_CODE	0xC3
ELSE	0x9D	OP_LONG	0xC4
THEN	0x9E	CONV_ADDR_TO_TF	0xC5
BEGIN	0x9F		
WHILE	0xA0	(以降 0xD6 まで空き)	
REPEAT	0xA1		
DUP	0xA2	(0xD7 以降ユーザ定義)	
PICK	0xA3		
DROP	0xA4		
SWAP	0xA5		
ROLL	0xA6	EXTENSION	0xFF

なお、HEX、DECIMAL、BINARY、CONSTANT、C”、FIND は、圧縮処理の中で処理が完了するため、コマンドバイトは割り当てられない。

(2) データバイト

データバイトは、先頭ビットが0のバイトである。残りの7ビットがデータの内容を表す。データバイトの生成方法の詳細は、3. 10. 7. 3に記す。

3.10.7.3 プログラム圧縮方法

プログラム圧縮処理では、コマンドを表す文字列はコマンドバイトに変換され、データを表す文字列はデータバイト列に変換される。圧縮されたプログラムの最後には、プログラムの終了を表すコマンドバイト(END_OF_CODE(0xC3))が付与される。

データを表す文字列からデータバイト列への変換は、3.10.7.1で概説したとおり、以下の手順に従って行われる。(図3-42を参照)

- (i) データを表す文字列を数値に変換する。
- (ii) 上記(i)で変換した数値を、情報量(ビットサイズ)に従って切り詰めたビットパターンを作成する。

(図3-42の例では、中間オブジェクトIDは値が「1」で情報量が2bitであるので、「01」というビットパターンを、ECHONETプロパティコードは値が「80」で情報量が8bitであるので「10000000」というビットパターンを、プロパティの型は値が「3」で情報量が3bitであるので「011」というビットパターンを作成する。)

- (iii) コマンド(図3-42ではRGST_EPC)の直前のデータから順に、上記(ii)で生成したビットパターンを右詰でバイトの2ビット目以下に配置し、バイトの先頭に0を付したデータバイト列を生成する。

各データの情報は、3.10.7.4を参照のこと。3.10.7.4に指定されていないデータの情報は16ビットである。

以下に、圧縮プログラムの生成方法を示す。

テキスト形式プログラムの先頭から文字列を1つずつ読み込み、

- (1) 文字列がデータの時、読み飛ばす。
- (2) 文字列が「VARIABLE」、「:」、「CREATE」の時、
 - (2-1) 読み飛ばしたデータをデータバイト列に変換し、出力データに追加する。
 - (2-2) 「VARIABLE」、「:」、「CREATE」に対応するコマンドバイトを出力データに追加する。
 - (2-3) 次の文字列に、ユーザ定義コマンドバイトを割り当て、出力データに追加する。
- (3) 文字列が、上記(2)以外のコマンド(ユーザ定義の変数、配列、関数を含む)の時、
 - (3-1) 読み飛ばしたデータをデータバイト列に変換し出力データに追加する。
 - (3-2) コマンドに対応するコマンドバイトを出力データに追加する。

読み込みが終了したら、END_OF_CODE(0xC3)を出力データに付与し、出力データを出力して終了する。

ただし、上記手順に以下の2つの例外を設ける。

- (a) RGST_FUNC_PROP で、関数型プロパティ関係を登録するときに指定する「C"FUNC_NAME" FIND DROP」(FUNC_NAMEは関数名)は、FUNC_NAMEに対応するコマンドバイトに対して上記(3)の処理を行わず、コマンドデータの値を8bitのデータとして、圧縮処理を行う。

(b) SET_BUF の2つ前の文字列が CREATE で定義されたバッファ名の場合、バッファ名に対応するコマンドバイトに対して上記(3)の処理を行わず、コマンドバイトの値を 8bit のデータとして扱って圧縮処理を行う。

3.10.7.4 各情報のビットサイズ

データバイトの作成時に、デフォルトの 16bit 以外の情報量(ビットサイズ)を割り当てるデータの一覧を示す。

表3-15 データの情報量 (ビットサイズ)

データ	ビットサイズ	値の範囲	使用コマンド
リトライ回数	3	0~7	INIT_ECHONET
リトライ待機時間	15	0 ~ 32767(ミリ秒)	SET_COM_PRARM
終了判定モード	2	1~2	SET_UART_RV_MODE
受信終了判定時間	10	0~1023(ミリ秒)	SET_UART_RV_MODE
受信終了コード長	3	0~7(バイト)	SET_UART_RV_MODE
家電機器オブジェクトの数	4	0~15	CREATE_MNG_TABLES
家電機器側定義プロパティの数	7	0~127	CREATE_MNG_TABLES
配列でない ECHONET プロパティの数	7	0~127	CREATE_MNG_TABLES
配列の ECHOENT プロパティの数	7	0~127	CREATE_MNG_TABLES
同一値型プロパティ関係の数	7	0~127	CREATE_MNG_TABLES
マッピング型プロパティ関係の数	7	0~127	CREATE_MNG_TABLES
関数型プロパティ関係の数	7	0~127	CREATE_MNG_TABLES
中間オブジェクト ID	4	0~15	RGST_OBJ RGST_EPC RGST_EPCM ADD_EPC_MEMBER RGST_IPC RGST_IDENTICAL_PROP RGST_MAP_PROP_REL RGST_FUNC_PROP SET_IPC SET_SEND_IPC GET_IPC
ECHONET オブジェクトのクラスグループ	8	0x00~0xFF	RGST_OBJ

ECHONET オブジェクトのクラス	8	0x00~0xFF	RGST_OBJ
ECHONET オブジェクトのインスタンス	8	0x00~0xFF	RGST_OBJ
ECHONET プロパティコード (EPC)	8	0x00~0xFF	RGST_EPC RGST_EPCM ADD_EPC_MEMBER RGST_IDENTICAL_PROP RGST_MAP_PROP_REL RGST_FUNC_PROP
プロパティの型	3	0~7	RGST_EPC RGST_EPCM
アクセスルール(配列でない EPC)	3	0x00~0x03	RGST_EPC
アクセスルール(配列 EPC)	7	0x00~0x7F	RGST_EPCM
状態時アナウンス有無	1	0~1	RGST_EPC RGST_EPCM
プロパティ値保持フラグ	1	0~1	RGST_EPC RGST_EPCM
データエリアサイズ(バイト数)	8	0~255(バイト)	RGST_EPC RGST_EPCM
要素番号 (配列 EPC)	8	0~255	RGST_EPCM ADD_EPC_MEMBER RGST_IDENTICAL_PROP RGST_MAP_PROP_REL RGST_FUNC_PROP
中間オブジェクトのプロパティコード(IPC)	8	0~255	RGST_IPC RGST_IDENTICAL_PROP RGST_MAP_PROP_REL RGST_FUNC_PROP SET_IPC SET_SEND_IPC GET_IPC
中間オブジェクトのプロパティの型	3	1~7	RGST_IPC
データエリアサイズ(バイト数)	8	0~255	RGST_IPC
中間オブジェクトプロパティコードの数	3	0~7	RGST_MAP_PROP_REL RGST_FUNC_PROP
ECHONET プロパティコードの数	3	0~7	RGST_MAP_PROP_REL RGST_FUNC_PROP
マッピングレコードの数	8	0~255	RGST_MAP_PROP_REL
マッピング型変換テーブルの ID	7	0~127	RGST_MAP_PROP_REL RGST_MAP_PROP_VAL RGST_MAP_PROP_VAL_PR

中間オブジェクトプロパティ値(IDT)	16	0x0000 ~ 0xFFFF	RGST_MAP_PROP_VAL RGST_MAP_PROP_VAL_PR SET_IPC SET_SEND_IPC
ECHONET オブジェクトプロパティ値(EDT)	16	0x0000 ~ 0xFFFF	RGST_MAP_PROP_VAL RGST_MAP_PROP_VAL_PR
IPC→EPC 変換の優先フラグ	1	0~1	RGST_MAP_PROP_VAL_PR
EPC→IPC 変換の優先フラグ	1	0~1	RGST_MAP_PROP_VAL_PR
バッファ最大要素数	7	0~127	CHK_RV_IPC
(送受信)バッファのサイズ	7	0~127	FROM_EQUIPMENT TO_EQUIPMENT SET_BUF
タイムアウト時間	15	0 ~ 32768(ミリ秒)	FROM_EQUIPMENT
待機時間	15	0 ~ 32768(ミリ秒)	SLEEP
時間	15	0~32760(x10ミリ秒)	SET_TIMER
スタートモード	3	0~7	INIT_ECHONET
ノードID	8	0~255	RGST_NODE RGST_OBJ
ブロッキングモード指定フラグ	1	0~1	TO_EQUIPMENT
状態	1	0~1	INDICATE_STATUS
演算の種別	5	0~31	OP_LONG
1 バイトデータ	8	0x00~0xFF	SET_BUF
2 バイトデータ	16	0x0000 ~ 0xFFFF	(デフォルト)

3.10.7.5 変数・関数の使用の制限

一部のコマンドは、処理するデータを変数や関数の戻り値で指定できないものがある。この場合、直接値を指定するか、CONSTANT で定義した定数で指定する。

正しい記述方法(1)

```
HEX  
1 80 3 3 1 0 1RGST_EPC
```

正しい記述方法(2)

```
HEX  
3 CONSTANT RULE  
1 80 3 RULE 1 0 1RGST_EPC
```

正しくない記述方法(1)

```
HEX  
VARIABLE RULE  
RULE 3!  
1 80 3 RULE @ 1 0 1RGST_EPC
```

正しくない記述方法(2)

```
HEX  
: FUNC  
3  
;  
1 80 3 FUNC 1 0 1RGST_EPC
```

値、または CONSTANT で定義した定数を使用する必要がある API を以下に示す。

- (1)INIT_ECHONET が処理するデータ
- (2)CREATE_MNG_TABLES が処理するデータ
- (3)RGST_EPC が処理するデータ
- (4)RGST_EPCM が処理するデータ
- (5)ADD_EPC_MEMBER が処理するデータ
- (6)RGST_IPC が処理するデータ
- (7)RGST_IDENTICAL_PROP が処理するデータ
- (8)RGST_MAP_PROP_REL が処理するデータ
- (9)RGST_FUNC_PROP が処理するデータ(「C" FUNC_NAME" FIND DROP」
(FUNC_NAME は関数名))を除

付録1 参考文献

- (1) 「PH-CONNECTER」 日本圧着端子製造(株) 発行

付録2 インタプリタ方式サンプルプログラム

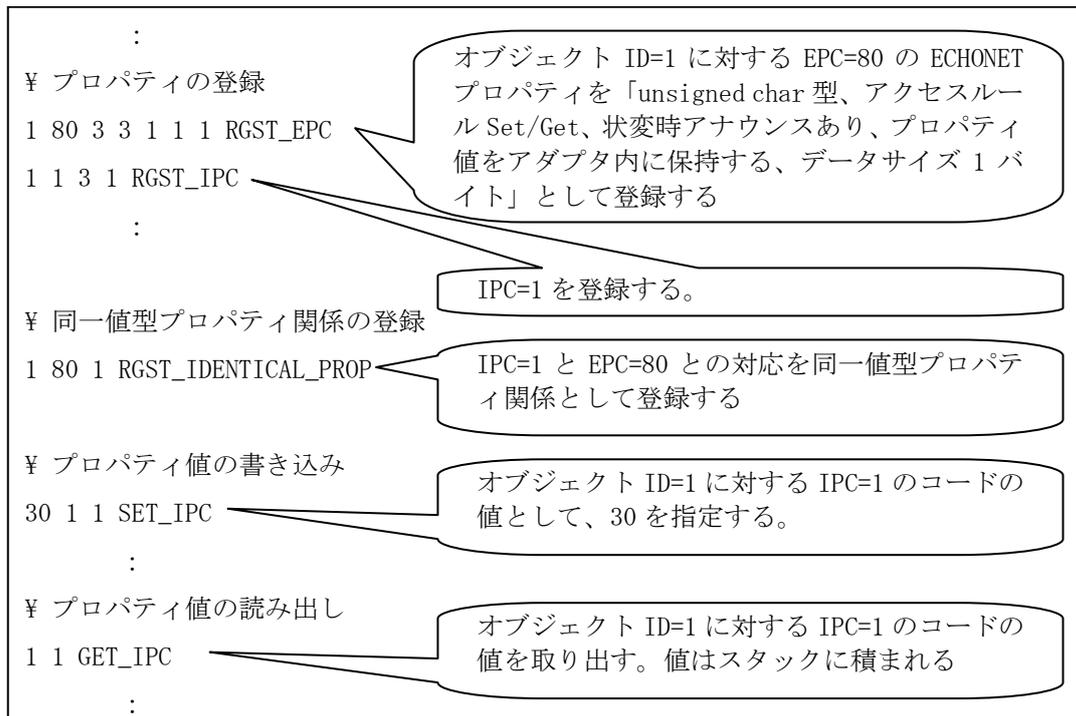
(1) プロパティ変換のサンプル

(1.1) 同一値型プロパティ変換

同一値型プロパティ関係の変換についてサンプルを示す。下記は、家庭用エアコンにおいて、中間オブジェクトプロパティの動作状態 (IPC=0x01) と、ECHONET プロパティとの動作状態 (EPC=0x01) が同一値型の関係にある場合のプロパティ関係の登録と、プロパティの読み出し/書き込みのコードのサンプルである。

動作状態 (IPC=0x01)	動作状態 (EPC=0x80)
0x31 (電源 OFF)	0x31 (電源 OFF)
0x30 (電源 ON)	0x30 (電源 ON)

プロパティ値の関係



プログラムコード

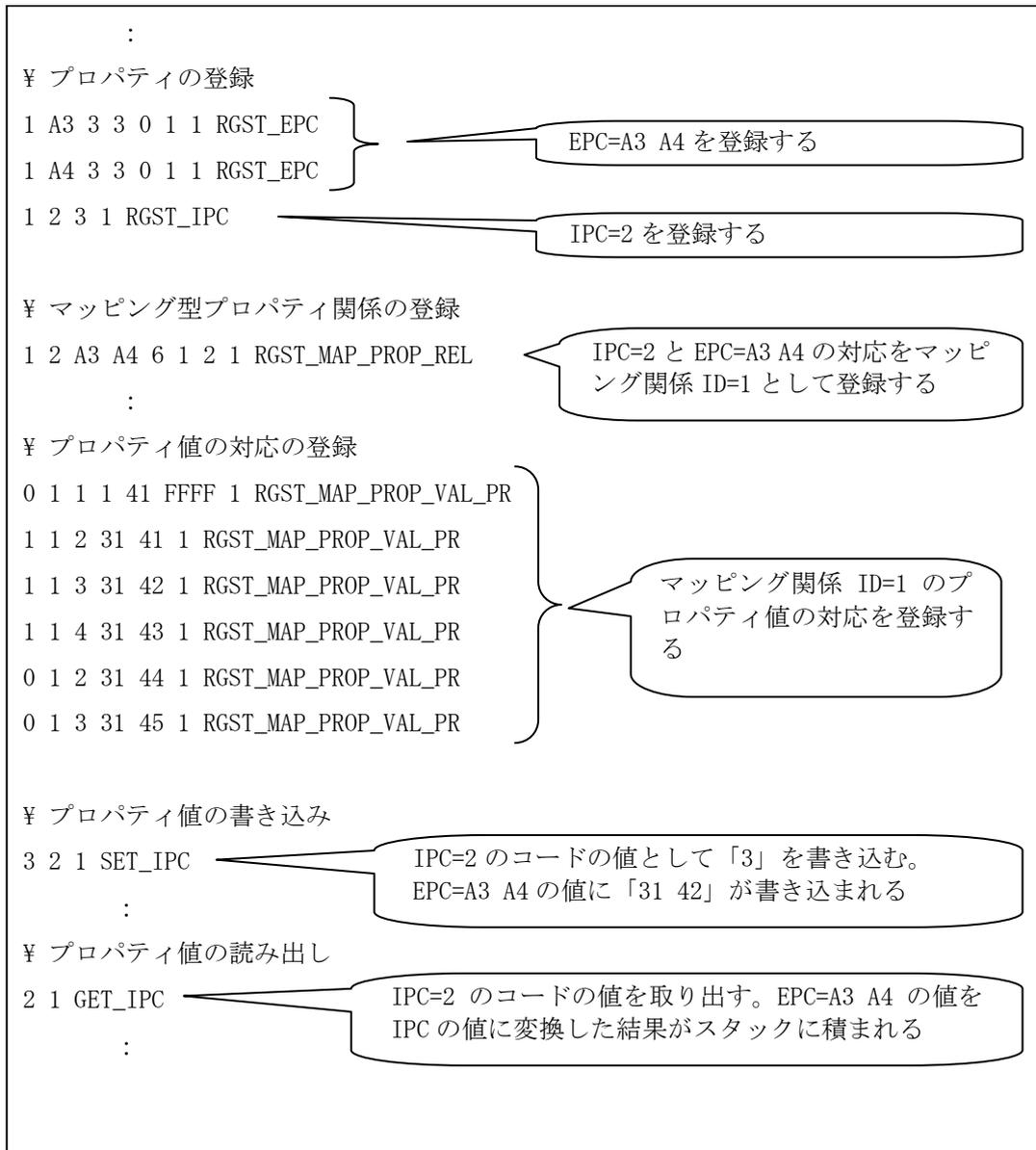
(1.2) マッピング型プロパティ変換

同一値型プロパティ関係の変換についてサンプルを示す。マッピング型プロパティ関係の例で、家庭用エアコンにおいて、中間オブジェクトプロパティの上下風向(IPC=0x02)が、ECHONET プロパティの風向スイング設定(EPC=0xA3)と風向き上下設定(EPC=0xA4)に対応しており、値の対応が下記の通りであるとする。下記の例では、IPC 値と EPC の値の組の対応に、1 対多のものがあるため、優先する対応に、(優)の印を付与している。

上下風向 (IPC=0x02)	風向スイング設定 (EPC=0xA3)	風向上下設定 (EPC=0xA4)
0x01(スイング)	0x41(上下)	(no care)
(優) 0x02(上)	0x31(OFF)	0x41(上)
(優) 0x03(下)	0x31(OFF)	0x42(下)
0x04(中央)	0x31(OFF)	0x43(中央)
0x02(上)	0x31(OFF)	0x44(上中)
0x03(下)	0x31(OFF)	0x45(下中)

プロパティ値の関係

上記のマッピング型関係にある場合のプロパティ変換コードのサンプルを以下に示す。



プログラムコード

(1.3) 関数型プロパティ変換

関数型プロパティ関係の変換についてサンプルを示す。下記は、家庭用エアコンにおいて、中間オブジェクトプロパティの設定温度の値が、ECHONET プロパティの設定温度の値より、常に10大きい関係にある場合の関数の定義である。参考までに、C言語の表現も示す。

¥ 「設定温度」の IPC→EPC 変換

```
: I2E  
10 -  
;
```

¥ 「設定温度」の EPC→IPC 変換

```
: E2I  
10 +  
;
```

プロパティ値の関係

```
unsigned char I2E (unsigned char x)
```

```
{  
    return (x - 0x10);  
}
```

```
unsigned char E2I (unsigned char x)
```

```
{  
    return (x + 0x10);  
}
```

プロパティ値の関係(参考:C言語表現)

上記の関数型関係にある場合のプロパティ変換コードのサンプルを下記に示す。

```
      :
¥ プロパティの登録
1 B3 3 3 0 0 1 RGST_EPC
1 3 3 1 RGST_IPC

¥ 「設定温度」の IPC→EPC 変換
: I2E
  10 -
;

¥ 「設定温度」の EPC→IPC 変換
: E2I
  10 +
;

¥ プロパティ値の対応の登録
C" I2E" FIND DROP C" E2I" FIND DROP 3 B3 1 1 1 RGST_FUNC_PROP

¥ プロパティ値の書き込み
30 3 1 SET_IPC

¥ プロパティ値の読み出し
3 1 GET_IPC
      :
```

EPC=B3 を登録する(データサイズ1バイト)

IPC=3 を登録する(データサイズ1バイト)

IPC=3 と EPC=B3 との対応を、関数名と共に、関数型プロパティ関係として登録する

IPC=2 の値が 48 に対応する EPC の値(32)を書き込む

オブジェクト ID 「1」、IPC コード 「3」 のときの IPC 値を、スタック上に積む

プログラムコード

(2) 全体処理のサンプル

以下に、動作状態(電源 ON/OFF)のプロパティ変換と通信処理を実行する全体プログラムのサンプルを示す。値の対応は下記のようなマッピング型とする。

動作状態 (IPC=0x01)	動作状態 (EPC=0x80)
0x01(電源 OFF)	0x31(電源 OFF)
0x02(電源 ON)	0x30(電源 ON)

HEX	¥ 16 進数に設定
400 SET_COM_PARAM	¥ リトライ待機時間 1024ms
10 1 SET_UART_RV_MODE	¥ 受信終了判定時間 16msec
CREATE RV_BUF 16 ALLOT	¥ 受信バッファ 22 バイト
CREATE TR_BUF 16 ALLOT	¥ 送信バッファ 22 バイト
CREATE SRC_BUF 5 ALLOT	¥ 送信元格納バッファ
CREATE OBJ_BUF 5 ALLOT	¥ 受信オブジェクト格納バッファ
CREATE IPC_BUF 5 ALLOT	¥ 受信 IPC 格納バッファ
CREATE RV_CD_BUF 5 ALLOT	¥ 受信コード格納バッファ
VARIABLE ERR_FLG	¥ エラーフラグ
0 ERR_FLG !	¥ エラーフラグに 0 を設定
VARIABLE RV_NUM	¥ 受信数
VARIABLE CNT	¥ ループカウンタ
VARIABLE CNT2	¥ ループカウンタ 2
VARIABLE IDT	¥ IDT
VARIABLE FCC	¥ FCC
1 1 1 0 0 1 0 CREATE_MNG_TABLES 1)	¥ 管理テーブルの作成(オブジェクト 1、EPC 1)
1 0 1 30 1 RGST_OBJ	¥ オブジェクト登録
1 80 3 3 1 1 1 RGST_EPC	¥ 動作状態(EPC=80)登録
1 1 3 1 RGST_IPC	¥ 動作状態(IPC=1)登録
1 1 80 2 1 1 1 RGST_MAP_PROP_REL 関	¥ IPC=1 と EPC=80 をマッピング型プロパティ 関
	¥ 係(ID=1)として対応付け
1 31 1 RGST_MAP_PROP_VAL	¥ 電源 OFF
2 30 1 RGST_MAP_PROP_VAL	¥ 電源 ON

```

: CALC_FCC                                     ¥ 送信時 FCC 計算 (TR_BUF[1]-[14]までの和)
1 CNT2 !
0 FCC !
BEGIN
  CNT2 @ 14 <
WHILE
  TR_BUF CNT2 @ + C@ FCC @ + FCC !      ¥ FCC += TR_BUF[CNT]
  CNT2 @ 1 + CNT2 !
REPEAT
FCC @ FF AND
;

: CHK_FCC                                     ¥ 受信時 FCC チェック (TR_BUF[1]-[14]までの和)
1 CNT2 !
0 FCC !
BEGIN
  CNT2 @ 14 <
WHILE
  RV_BUF CNT2 @ + C@ FCC @ + FCC !      ¥ FCC += RV_BUF[CNT]
  CNT2 @ 1 + CNT2 !
REPEAT
FCC @ FF AND 0 = IF
  0                                     ¥ OK (return 0)
ELSE
  1                                     ¥ NG (return 1)
THEN
;

: PR_ECHO                                     ¥ 関数定義 (ECHONET Lite からの受信に対する処理)
● ● . . . ● TR_BUF 16 SET_BUF          ¥ 送信バッファ設定 (操作要求)
¥ ● ● . . . ● の部分は家電機器通信の仕様に
¥ よって異なる
IPC_BUF CNT @ + C@ 1 = IF                ¥ IPC=1 の場合
1 1 GET_IPC IDT !                         ¥ IDT (IPC=1) を取得
IDT @ TR_BUF ● + C!                       ¥ IDT を送信バッファの ● バイト目に設定
CALC_FCC TR_BUF 15 + C!                   ¥ FCC 設定
TR_BUF 16 0 TO_EQUIPMENT                  ¥ 機器に送信

```

```

RV_BUF 16 100 FROM_EQUIPMENT          ¥ 機器から受信
2 = IF                                  ¥ 受信ありの場合
  CHK_FCC 0 = IF                        ¥ FCC OK の場合
    RV_BUF 5 + C@ 0 = IF                ¥ 正常応答の場合
      IDT @ 1 1 SET_IPC                 ¥ IPC セット
    THEN
  THEN
THEN
;

: PR_REG                                ¥ 関数定義(定常処理)
● ● . . . ● TR_BUF 16 SET_BUF          ¥ 送信バッファ設定(状態要求)
TR_BUF 16 0 TO_EQUIPMENT                ¥ 機器に送信
RV_BUF 16 100 FROM_EQUIPMENT           ¥ 機器から受信
2 = IF                                  ¥ 受信ありの場合
  CHK_FCC 0 = IF                        ¥ FCC OK の場合
    RV_BUF ● + C@ 1 1 SET_IPC          ¥ 送信バッファの●バイト目より IDT 設定
  THEN
THEN
;

1 3 INIT_ECHONET                        ¥ ECHONET Lite 初期化

PR_REG                                  ¥ 初期値設定(定常処理)

BEGIN
  TRUE
WHILE
  5 SRC_BUF OBJ_BUF IPC_BUF RV_CD_BUF CHK_RV_IPC
                                          ¥ ECHONET Lite から受信
RV_NUM !                                ¥ 受信数を設定
0 CNT !                                  ¥ カウンタを設定
BEGIN
  CNT @ RV_NUM @ <
WHILE
  PR_ECHO                                ¥ ECHONET Lite からの受信に対する処理
  CNT @ 1 + CNT !                        ¥ ループカウンタを1つ増やす
REPEAT
  PR_REG                                  ¥ 機器の状態確認(定常処理)
REPEAT

```