

## スマートメーターと HEMS

## 第 6 回 ECHONET Lite (2)

アンリツカスタマーサポート株式会社  
EMC センター

&lt;&lt;はじめに&gt;&gt;

前回は、ECHONET Lite のネットワークの構成、機器の分類、フレームフォーマットなどをお話しました。今回も ECHONET Lite を続けたいと思います。  
宜しくお願い致します。

ECHONET Lite は、エコーネットコンソーシアム<sup>[1]</sup>が開発・管理する通信プロトコルです。2012 年 2 月に、スマートハウス標準化検討会で、HEMS とスマートメーター B ルートの標準インターフェースに推奨されました。

今回も、「ECHONET Lite 規格書 Version 1.11」(文献 [1]) から、ECHONET Lite の仕様をご紹介します。

ECHONET Lite の通信レイヤは、アプリケーションソフトウェア、ECHONET Lite 通信ミドルウェア、下位通信層で構成されます。

文献 [1] は、このうち、ECHONET Lite 通信ミドルウェアを規定します。

ECHONET Lite 通信ミドルウェアは、アプリケーションソフトウェアと下位通信層の間にあって、ECHONET Lite 通信処理部と ECHONET オブジェクトで構成されています。

ECHONET Lite 通信処理部は、ECHONET オブジェクトを介してアプリケーションソフトウェアと情報をやり取りします。それに従って、ECHONET Lite フレームを作って下位通信層に渡したり、逆に、下位通信層から受け取ったフレームをばらして、その内容をアプリケーションソフトウェアに伝えたりします。

ECHONET オブジェクトは、機器の機能モデルです。機器オブジェクトとプロファイルオブジェクトの二つがあります。機器オブジェクトは、いわゆるエアコン、給湯器といった、機器本来の機能に関するモデルです。一方、プロファイルオブジェクトは、通信ノードとしてのモデルです。

## 1 ECHONET オブジェクトの構造

機器オブジェクトは、「エアコン」、「給湯器」という機能ごとに定義されています。それぞれの機能を「クラス」と言います。家庭用エアコンクラス、瞬間式給湯器クラス——という具合です。

クラスは、用途や目的に応じて、「クラスグループ」に分けられています。たとえば、家庭用エアコンクラスと瞬間式給湯器クラスは、それぞれ、空調関連機器クラスグループ、住宅・設備関連機器クラスグループに属しています。

また、特別なクラスとして、すべての機器に共通に適用する「スーパークラス」があります。

各クラスには、関連する機能・情報がリストアップされています。その項目を「プロパティ」と言います。プロパティには、それぞれ、制御値や値のフォーマット、あるいは「搭載は必須? 任意?」——といったことが規定されています。これによって、他の機器とのインターフェースを標準化します。

スーパークラスを除くクラスは、クラスグループコード (大分類) とクラスコード (小分類) で識別されます。

クラスコードの下には、さらに、「インスタンスコード」という追番が設けられています。インスタンスコードによって、たとえば、エアコン同士を区別できるようになります。

クラスグループコード、クラスコード、インスタンスコードを並べたものを、EOJ コードと言うことがあります。

ECHONET Lite 通信ミドルウェアでは、実際の機器は EOJ コードで区別されます。

機器オブジェクトの仕様は、文献 [1] とは別に、「Appendix ECHONET 機器オブジェクト詳細規定」(文献 [2]) にまとめられています。

現在、約 100 種類のオブジェクトが定義されています。

機器オブジェクトに対して、プロファイル オブジェクトは通信ノードとしての機能・情報のモデルです。

## スマートメーターと HEMS

## 第 6 回 ECHONET Lite (2)

プロファイルオブジェクトにも、機器オブジェクトと同じように、クラスグループコード、クラスコード、インスタンスコードが設けられています。

ただし、次の種類だけです。

- クラスグループコード 0x0E
- クラスコード 0xF0

クラスグループコード 0x0E は「プロファイルオブジェクトクラスグループ」、クラスコード 0xF0 は「ノードプロファイルクラス」を表します。

## 2 基本シーケンス

ECHONET Lite 機器が必ず搭載しなければならない、通信の手順を「基本シーケンス」と言います。

### 2.1 オブジェクト制御の基本シーケンス

ECHONET Lite のメッセージは、

- プロパティ値の「書き込み要求」、
- プロパティ値の「読み出し要求」、
- 要求に対する応答、
- プロパティ値の「通知」

の四つが基本になっています。

これらを基にして、前回 (第 5 回) 表 5.3-1、表 5.3-2、および表 5.3-3 に示したメッセージタイプ<sup>[2]</sup> が設けられています。(参考に、表 5.3-1、表 5.3-2、表 5.3-3 を、それぞれ、付表 1、付表 2、付表 3 に再掲します)

ECHONET Lite 通信ミドルウェアのやり取りは、これらのメッセージタイプに従って行われます。メッセージタイプは、ECHONET Lite フレームの ESV フィールドに ESV コードで指定します。ESV は、「ECHONET Lite サービス」の略号です。(文献 [1] 第 2 部 4.2)

#### 2.1.1 書き込み要求 (応答不要) SetI のシーケンス

プロパティ値の「書き込み要求 (応答不要)」(SetI) ESV 0x60 は、送信先の機器の設定値を変えたいときに使います。ユニキャストもブロードキャストもできます。送信先に応答メッセージを要求しないので、素早く軽い制御ができます。

シーケンスを図 2.1.1-1 に示します。メッセージの送信先ノードは、メッセージを受け付けるまでに、二つのチェックをします。

**A DEOJ OK?** DEOJ フィールドに指定された DEOJ コードが、自ノードに搭載されている ECHONET オブジェクトの EOJ コードと一致するか？

一致しない場合、メッセージを破棄します。(何の応答もしません)

**B EPC OK?** EPC フィールドに指定された EPC コードが、自ノードに搭載されている ECHONET オブジェクトの EPC コードと一致するか？ かつ要求を処理できるか？

No の場合、要求を受理せず、「不可応答」(SetI\_SNA) ESV 0x50 メッセージを返します。応答の範囲はユニキャストです。(どの ESV の不可応答も、不可応答はすべてユニキャストのみです)

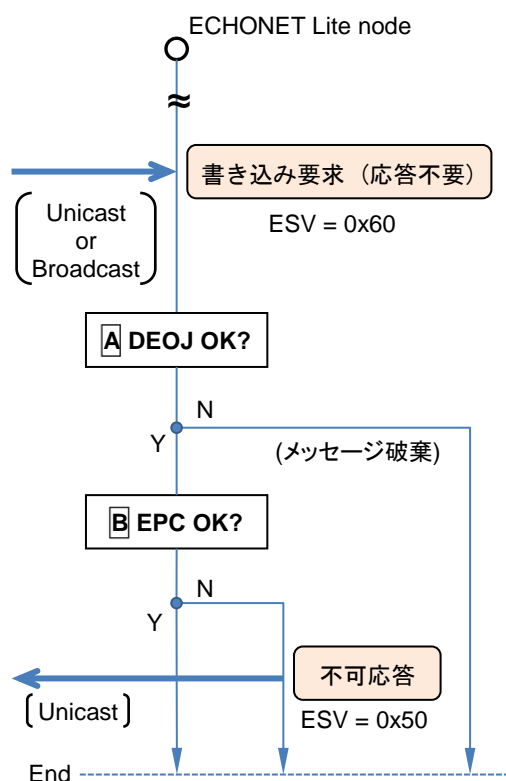


図 2.1.1-1: 書き込み要求 (応答不要) のシーケンス

SetI を使って、たとえば、コントローラが、エアコンをオンして、運転モードを暖房に設定するときのことを考えてみます。

このとき、関連するのは、エアコンの「動作状態」と「運転モード設定」のプロパティです。文献 [2]

## スマートメーターと HEMS

## 第 6 回 ECHONET Lite (2)

から、これらのパラメータを、それぞれ、表 2.1.1-1 と表 2.1.1-2 に示します。

表の中で、「アクセスルール」というのは、そのプロパティに可能な ESV をまとめたもので、表 2.1.1-3 のように決められています。「動作状態」「運転モード設定」、どちらのプロパティも、アクセスルール Set/ Get なので、SetI が可能です。

また、「状態時アナウンス」というのは、そのプロパティの状態が変わったときの通知のことです。これが「要」の場合、設定値が変わったら、それを必ず通知しなければいけません。

表 2.1.1-1: 家庭用エアコンの「動作状態」のプロパティ

クラスグループ	空調関連機器クラスグループ 0x01
クラス	家庭用エアコンクラス 0x30
EPC コード	0x80
値域	ON: 0x30 OFF: 0x31
データの型	unsigned char
サイズ	1 byte
アクセスルール	Set/ Get
搭載の必要性	必須
状態時アナウンス	要

表 2.1.1-2: 家庭用エアコンの「運転モード設定」のプロパティ

EPC コード	0xB0
値域	自動: 0x41 冷房: 0x42 暖房: 0x43 除湿: 0x44 送風: 0x45 その他: 0x40
データの型	unsigned char
サイズ	1 byte
アクセスルール	Set/ Get
搭載の必要性	必須
状態時アナウンス	要

表 2.1.1-3: アクセスルール

アクセスルール	ESV
Set	SetI, SetC, SetGet
Get	Get, SetGet, INF_REQ
Anno	INF, INFC

この例では、コントローラが送信するフレームは、図 2.1.1-2 のようになります。(参考に、ECHONET Lite のフレームフォーマットを付図 1 に示します)

**EHD 1** と **EHD 2** の値 0x10 81 は、おまじないのようなものです。ECHONET Lite メッセージは、必ず 0x10 81 で始まると思って間違いありません。

**TID** には、いくつを入れても構いません。

**SEOJ** には、送信元のコントローラの EOJ コード 0x05 FF 01 が入ります。

**DEOJ** には、送信先の家庭用エアコンの EOJ コード 0x01 30 01 が入ります。

**ESV** は、SetI の 0x60 です。

SetI の処理対象カウンタは、(OPCSet や OPCGet ではなく、) OPC です。

**OPC** には、このフレームで扱う プロパティが二つ(「動作状態」と「運転モード設定」)なので、2 を入れます。

**EPC 1** と **2** には、「動作状態」と「運転モード設定」の EPC、0x80 と 0xB0 を入れます。

**PDC 1** と **2** には、それぞれ、EDT 1 と EDT 2 のバイト数を入れます。

**EDT 1** と **2** には、それぞれ、動作状態 オン(0x30) と運転モード 暖房(0x43) の値を入れます。

<不可応答について>

SetI に対する不可応答 (SetI\_SNA) のフレームは、(a) 一致しない EPC がある場合、(b) 多過ぎる処理できない EPC がある場合、それぞれで、図 2.1.1-3、図 2.1.1-4 のように作ります。

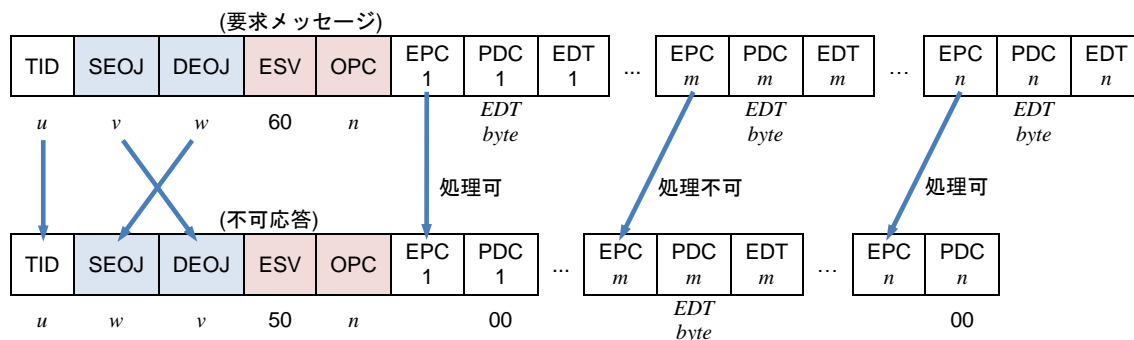
EHD 1	EHD 2	TID	SEOJ	DEOJ	ESV	OPC	EPC 1	PDC 1	EDT 1	EPC 2	PDC 2	EDT 2
10	81	xxxx	05 FF 01	01 30 01	60	02	80	01	30	B0	01	43

値は 16 進数です。

図 2.1.1-2: 家庭用エアコンの設定例

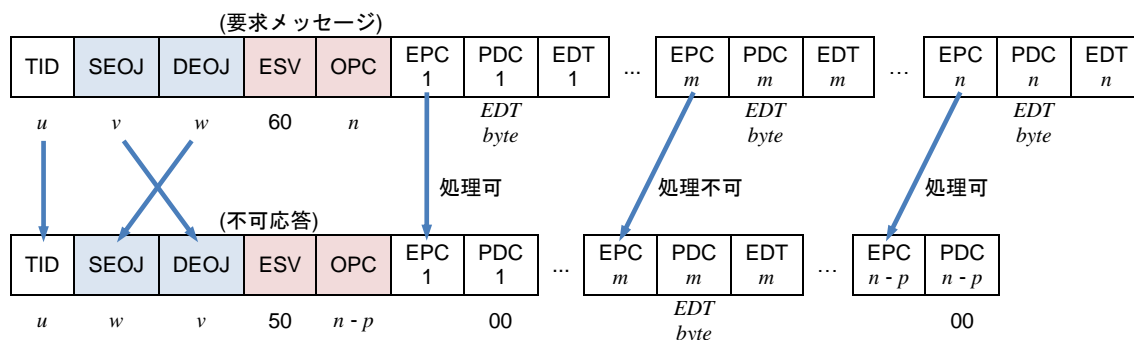
## スマートメーターと HEMS

## 第 6 回 ECHONET Lite (2)



値は 16 進数です。

図 2.1.1-3: SetI (0x60) への不可応答 (一致しない EPC がある場合)



値は 16 進数です。p は、オーバーフローして、処理できなかった EPC の数です。

図 2.1.1-4: SetI (0x60) への不可応答 (EPC が多すぎて処理できない場合)

(a) 一致しない EPC がある場合の不可応答は、次のようにします。

- 受信した EPC を列挙します。これは、要求メッセージの EPC と一致するはずですが。
- 一致した EPC は、EDT を付けません。PDC には 0 バイトを入れます。
- 一致しないで処理できなかった EPC は、EDT に要求メッセージの要求値をそのまま入れ、PDC にはそのバイト数を入れます。
- OPC (EPC の数) には、受信した EPC の数を入れます。これは、要求メッセージの OPC と一致するはずですが。
- TID は、要求メッセージと同じ値を入れます。
- SEOJ と DEOJ は、要求メッセージと逆になります。

(b) 処理できない EPC がある場合も、処理できた EPC は EDT を付けず、処理できなかった EPC には要求された EDT をそのまま付けます。

ただし、数が多くて、オーバーフローしてしまった EPC は、不可応答の発信元としては、よく分からないので、応答メッセージに入れることができません。このため、要求メッセージと EPC が必ずしも一致しないことが起こり得ます。

OPC にはオーバーフローせずに処理できた EPC の数が入ります。

### 2.1.2 書き込み要求 (応答要) SetC

#### 読み出し要求 Get

#### 書き込み・読み出し要求 SetGet のシーケンス

プロパティ値の「書き込み要求 (応答要)」(SetC) ESV 0x61 は、SetI と同じく、送信先の機器の設定値を変えたいときに使います。

## スマートメーターと HEMS

## 第 6 回 ECHONET Lite (2)

プロパティ値の「読み出し要求」(Get) ESV 0x62 は、送信先の機器の設定値を知りたいときに使います。

プロパティ値の「書き込み・読み出し要求」(SetGet) ESV 0x6E は、送信先の機器の設定値を変えるのと読み出すのを一緒にやりたい状況で使います。

ただし、SetGet は、搭載必須でなく、オプションです。対応していない機器もあり得ます。そのときは、不可応答 (SetGet\_SNA) ESV 0x5E メッセージが返ってきます。

SetC、Get、SetGet、どの ESV も、ユニキャスト、ブロードキャスト、両方できます。

これらのシーケンスを図 2.1.2-1 に示します。

SetI と同じように、送信先は、DEOJ フィールドに指定された DEOJ コードが、自ノードに搭載されている ECHONET オブジェクトの EOJ コードと一致しない場合は、メッセージを破棄します。

また、EPC フィールドに指定された EPC コードが、自ノードに搭載されている ECHONET オブジェクトの EPC コードと一致しない場合や、EPC の数が多くて要求を処理できない場合は、要求を受理せず、「不可応答」メッセージを返します。応答の範囲はユニキャストです。(どの ESV の不可応答も、不可応答はすべてユニキャストのみです)

図 2.1.2-1 で、要求と応答、不可応答の ESV コードは、表 2.1.2-1 のような関係になります。

表 2.1.2-1: 要求・応答・不可応答

要求 (記号)	応答 (記号)	不可応答 (記号)
0x61 (SetC)	0x71 (Set_Res)	0x51 (SetC_SNA)
0x62 (Get)	0x72 (Get_Res)	0x52 (Get_SNA)
0x6E (SetGet)	0x7E (SetGet_Res)	0x5E (SetGet_SNA)

### (1) 書き込み要求 (応答要) SetC

SetC への応答 (Set\_Res) は、図 2.1.2-2 のようにします。応答の範囲は、ユニキャストです。

- 受信した EPC を列挙します。これは、要求メッセージの EPC と一致するはずですが、EDT は付けません。このため、PDC には 0 バイトを入れます。

- OPC (EPC の数) には、受信した EPC の数を入れます。これは、要求メッセージの OPC と一致するはずですが。
- TID には、要求メッセージと同じ値を入れます。
- SEOJ と DEOJ は要求メッセージと逆になります。

<不可応答について>

SetC への不可応答 (SetC\_SNA) の扱いは、SetI への不可応答 (SetI\_SNA) と同じです。

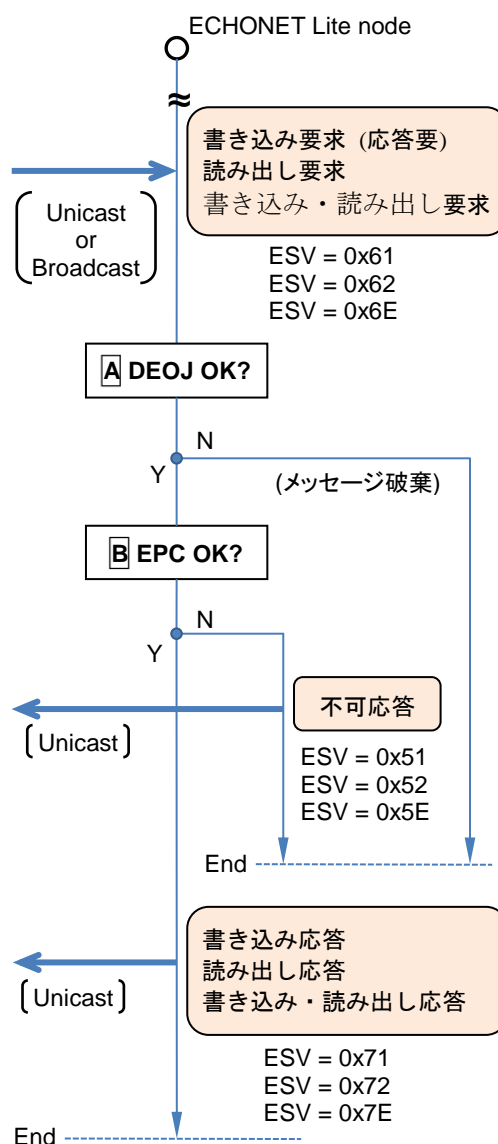


図 2.1.2-1: 書き込み要求 (応答要), 読み出し要求, 書き込み・読み出し要求のシーケンス

## スマートメーターと HEMS

## 第 6 回 ECHONET Lite (2)

**(2) 読み出し要求 Get**

Get のフレームとその応答 (Get\_Res) のフレームを、図 2.1.2-3 に示します。(EHD 1 と EHD 2 は省略しました)

Get のフレームは、読み出し要求する EPC を伝えればいいので、EDT は付きません。PDC には 0 バイトを入れます。

Get\_Res は次のようにします。応答の範囲は、ユニキャストです。

- 受信した EPC を列挙します。これは、要求メッセージの EPC と一致するはずでず。各 EDT に読み出したプロパティ値を入れ、PDC にはそのバイト数を入れます。
- OPC (EPC の数) には、受信した EPC の数を入れます。これは、要求メッセージの OPC と一致するはずでず。
- TID には、要求メッセージと同じ値を入れます。
- SEOJ と DEOJ を要求メッセージと逆にします。

## &lt;不可応答について&gt;

Get に対する不可応答 (Get\_SNA) は、一致しない EPC がある場合は、図 2.1.2-4 のようになります。

- 受信した EPC を列挙します。これは、要求メッセージの EPC と一致するはずでず。
- 一致した EPC には、EDT に読み出したプロパティ値を入れ、PDC にはそのバイト数を入れます。
- 一致しないで処理できなかった EPC には、EDT を付けません。このため PDC は 0 バイトになります。
- OPC (EPC の数) には、受信した EPC の数を入れます。これは、要求メッセージの OPC と一致するはずでず。
- TID は、要求メッセージと同じ値を入れます。
- SEOJ と DEOJ は、要求メッセージからひっくり返ります。

数が多くて、オーバーフローしてしまった EPC が一部ある場合も、処理できた EPC には読み出した EDT を付け、処理できなかった EPC は EDT を付けません。

ただし、オーバーフローしまった EPC については、SetI と同じです。

OPC にはオーバーフローせずに処理できた EPC の数が入ります。

**(3) 書き込み・読み出し要求 SetGet**

SetGet のフレームとその応答 (SetGet\_Res) のフレームを、図 2.1.2-5 に示します。(EHD 1 と EHD 2 は省略しています)

SetGet のフレームでは、OPC が、Set 用の OPCSet と Get 用の OPCGet の二つに分かれます。そして、最初に Set する EPC を指定し、その後に、Get する EPC を指定します。図 2.1.2-5 では、Set、Get を合わせた EPC の総数が  $n$  ケ、Set の EPC が  $m$  ケになっています。(参考に、ECHONET Lite の EDATA の構造を付図 2 に示します)

SetGet\_Res は、次のようにします。応答の範囲は、ユニキャストです。

- 受信した EPC を列挙します。これは、要求メッセージの EPC と一致するはずでず。
- Set の EPC には、PDC に 0 バイトを入れ、EDT を付けません。
- Get の EPC は、EDT に読み出したプロパティ値を入れ、PDC にはそのバイト数を入れます。
- OPCSet (Set の EPC の数) には、受信した Set の EPC の数を入れます。OPCGet (Get の EPC の数) には、受信した Get の EPC の数を入れます。これらは、要求メッセージの OPCSet、OPCGet と一致するはずでず。
- TID は、要求メッセージと同じ値を入れます。
- SEOJ と DEOJ は要求メッセージとひっくり返ります。

## &lt;不可応答について&gt;

SetGet の不可応答 (SetGet\_SNA) は、一致しない EPC がある場合は、図 2.1.2-6 のようにします。

- 受信した EPC を列挙します。これは、要求メッセージの EPC と一致するはずでず。
- Set の EPC では、一致した EPC には、PDC に 0 バイトを入れ、EDT を付けません。

## スマートメーターと HEMS

## 第 6 回 ECHONET Lite (2)

- 一致しない Set の EPC は、EDT に要求メッセージの要求値をそのまま入れ、PDC にはそのバイト数を入れます。
- Get の EPC では、一致した EPC には、EDT に読み出したプロパティ値を入れ、PDC にはそのバイト数を入れます。
- 一致しない Get の EPC は、PDC に 0 バイトを入れ、EDT を付けません。
- OPCSet には、受信した Set の EPC の数を入れます。OPCGet には、受信した Get の EPC の数を入れます。これらは、要求メッセージの OPCSet、OPCGet と一致するはずですが。
- TID は、要求メッセージと同じ値を入れます。
- SEOJ と DEOJ を要求メッセージとひっくり返します。  
数が多くて処理できない EPC がある場合も、Set の EPC では、処理できた EPC は EDT を付けず、処理できなかった EPC には要求された EDT を付けます。Get の EPC では、処理できた EPC には読み出した EDT を付け、処理できなかった EPC は EDT を付けません。  
ただし、オーバーフローした EPC については、他と同じです。  
OPCSet と OPCGet には、それぞれ、オーバーフローせずに処理できた EPC 数が入ります。

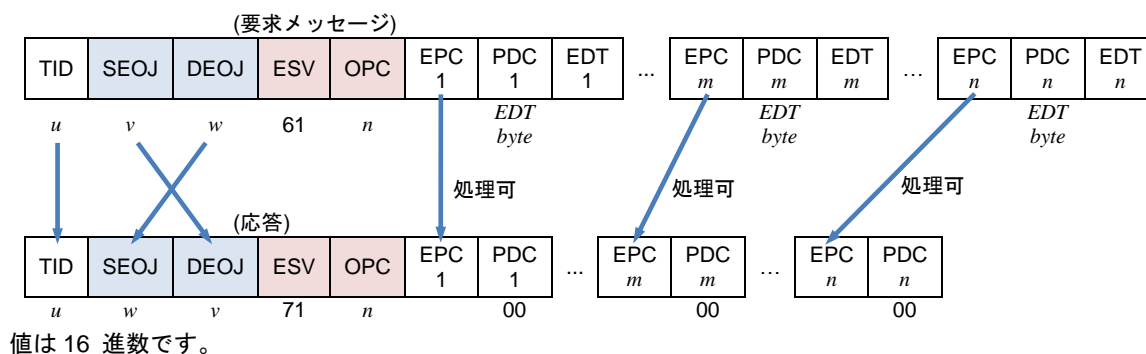


図 2.1.2-2: SetC (ESV 0x61) への応答

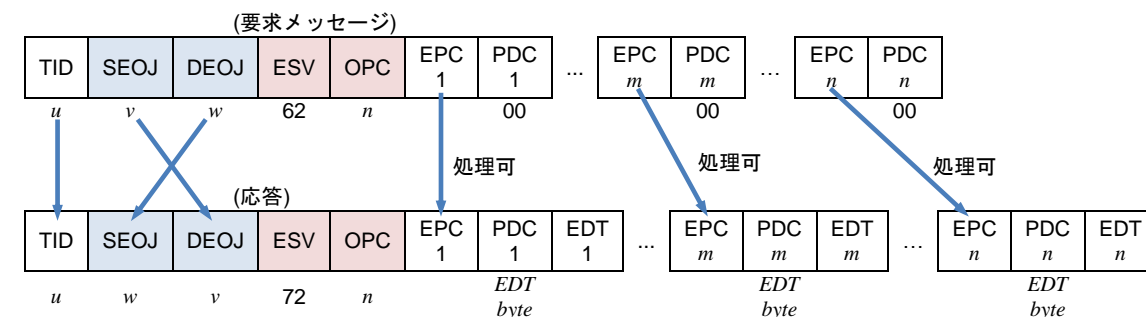
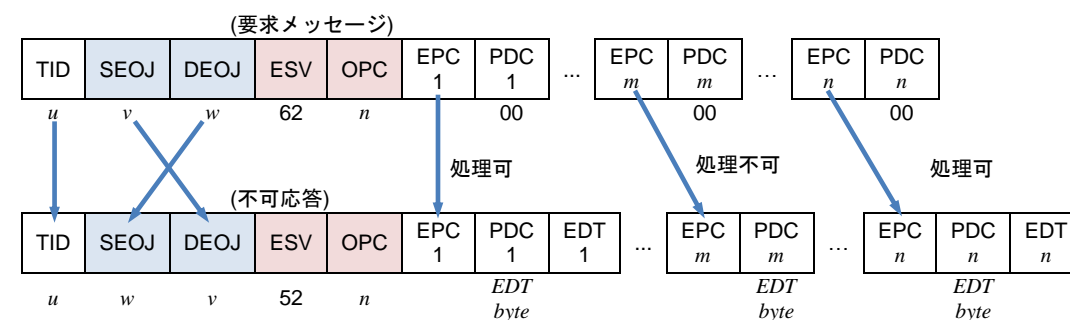


図 2.1.2-3: Get (ESV 0x62) への応答



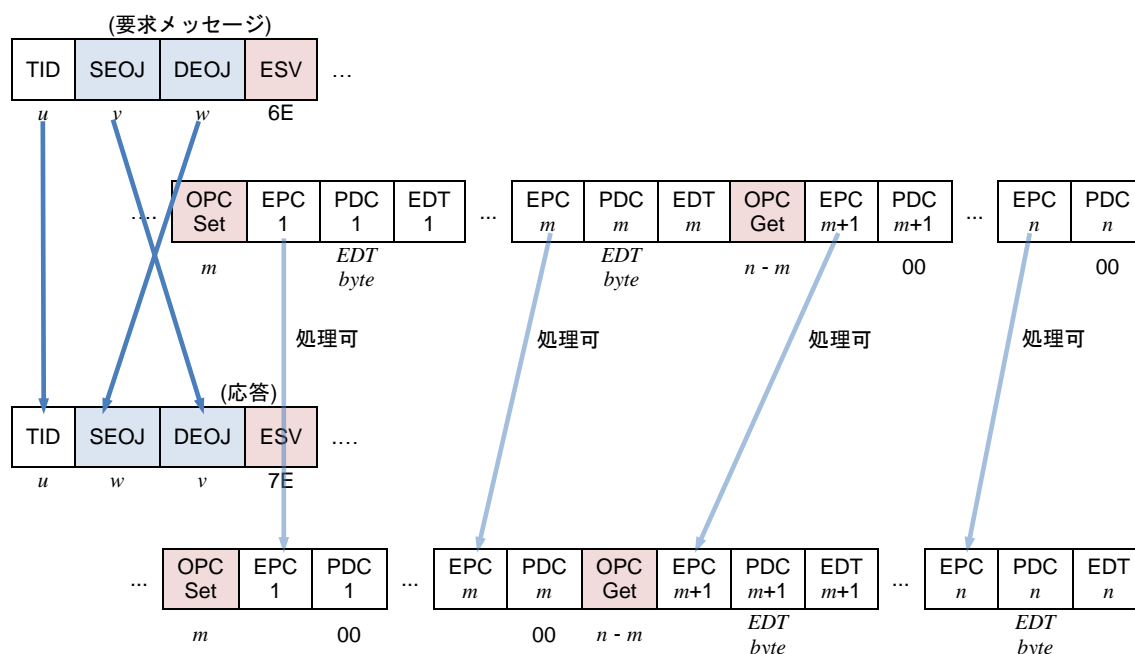
## スマートメーターと HEMS

### 第 6 回 ECHONET Lite (2)



値は 16 進数です。

図 2.1.2-4: Get (ESV 0x62) への不可応答 (一致しない EPC がある場合)



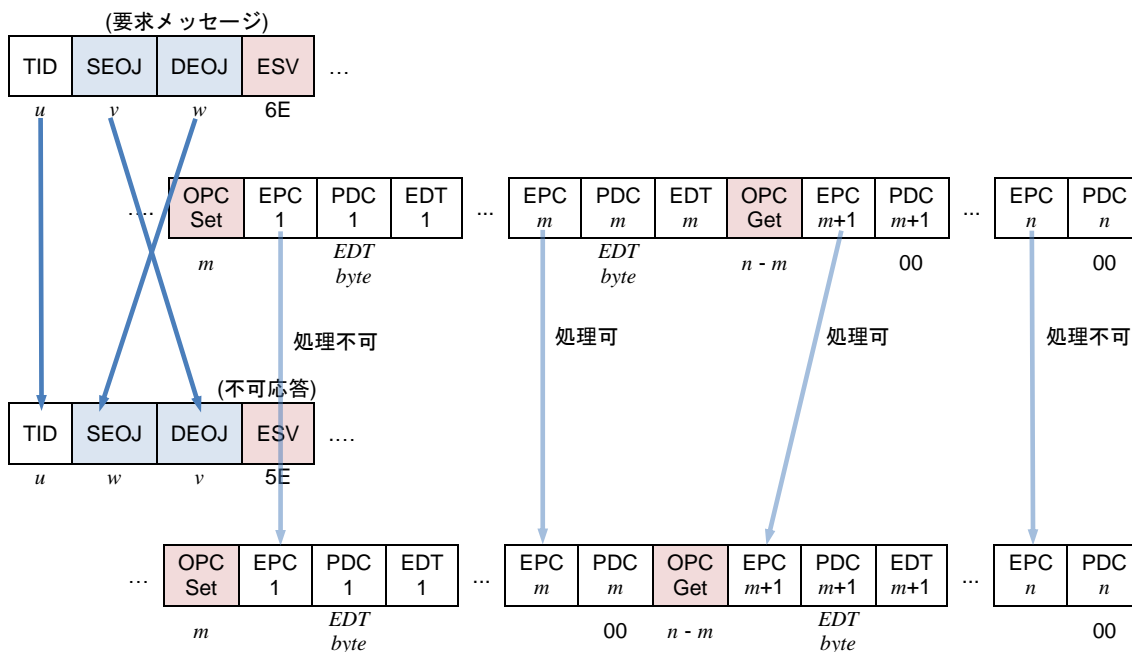
値は 16 進数です。

図 2.1.2-5: SetGet (ESV 0x6E) への応答



## スマートメーターと HEMS

### 第 6 回 ECHONET Lite (2)



値は 16 進数です。

図 2.1.2-6: SetGet (ESV 0x6E) への不可応答 (一致しない EPC がある場合)

### 2.1.3 通知 INF (自発的) のシーケンス

プロパティ値の「通知」(INF) ESV 0x73 には、プロパティ値の「通知要求」(INF\_REQ) に対する応答としての INF (後述) と自発的な INF があります。

自発的な INF は、他から要求はないけれど、自分のプロパティ値を誰かに知らせたいときに使います。

ユニキャスト、ブロードキャスト、両方可能です。

シーケンスを図 2.1.3-1 に示します。

また、INF のフレームは、図 2.1.3-2 のようになります。(EHD 1 と EHD 2 は省略です)

自発的 INF では、DEOJ にノードプロファイルオブジェクトが指定されるのが一般的です。

このフレームは、一見、SetI や SetC のフレームと同じ形をしています。EPC は、自ノード EPC になります。一方、SetI や SetC では、EPC は送信先の EPC を指しています。

このように、ESV によって、EPC の意味が変わってきます。(同じことが要求と応答の間でも起こります)

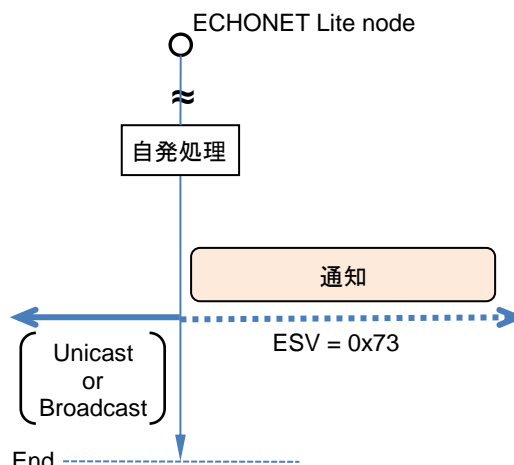
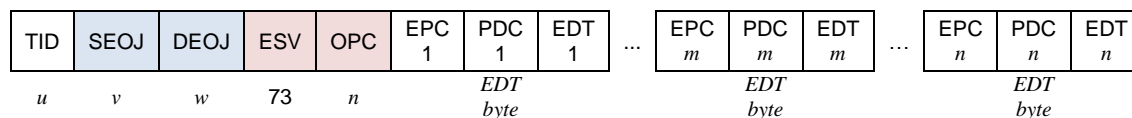


図 2.1.3-1: 通知のシーケンス

## スマートメーターと HEMS

## 第 6 回 ECHONET Lite (2)



値は 16 進数です。

図 2.1.3-2: 通知のフレーム

### 2.1.4 通知要求 INF\_REQ のシーケンス

プロパティ値の「通知要求」(INF\_REQ) ESV 0x63 は、送信先にプロパティ値を通知させるときに使用します。通知の範囲は、ブロードキャストです。

INF\_REQ 自身は、ユニキャストとブロードキャストが可能です。

シーケンスを図 2.1.4-1 に示します。

SetI や SetC、または Get、SetGet と同じように、送信先は、DEOJ フィールドに指定された DEOJ コードが、自ノードに搭載されている ECHONET オブジェクトの EOJ コードと一致しない場合は、メッセージを破棄します。

また、EPC フィールドに指定された EPC コードが、自ノードに搭載されている ECHONET オブジェクトの EPC コードと一致しない場合や、EPC の数が多くて要求を処理できない場合は、要求を受理せず、「不可応答」(INF\_SNA) ESV 0x53 メッセージを返します。応答の範囲はユニキャストです。(どの ESV の不可応答も、不可応答はすべてユニキャストのみです)

INF\_REQ への応答は INF になります。

INF\_REQ のフレームと INF のフレームを、図 2.1.4-2 に示します。(EHD 1 と EHD 2 は省略しています)

INF\_REQ のフレームは、Get のフレームと同じ形です。EDT を付けずに、要求する EPC を指定します。

応答の INF も、Get\_Res とほぼ同じです。ただし、応答の範囲はブロードキャストです。

<不可応答について>

不可応答は Get\_SNA と同じです。

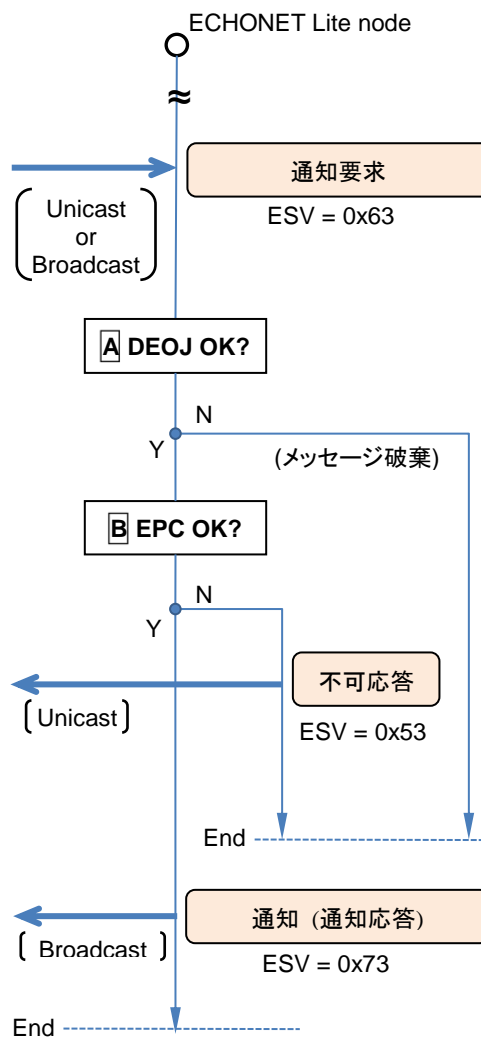


図 2.1.4-1: 通知要求のシーケンス

### 2.1.5 通知 (応答要) INFC のシーケンス

プロパティ値の「通知 (応答要)」(INFC) ESV 0x74 は、他から要求はないけれど、自分のプロパティ値を誰かに知らせつつ、受け取り確認もほしいときに使います。

シーケンスを図 2.1.5-1 に示します。

## スマートメーターと HEMS

### 第 6 回 ECHONET Lite (2)

送信先は、DEOJ フィールドに指定された DEOJ コードが、自ノードに搭載されている ECHONET オブジェクトの EOJ コードと一致しない場合は、メッセージを破棄します。

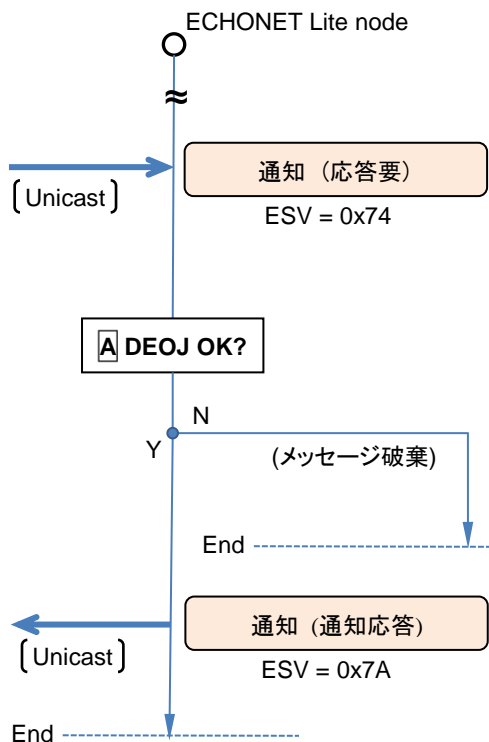


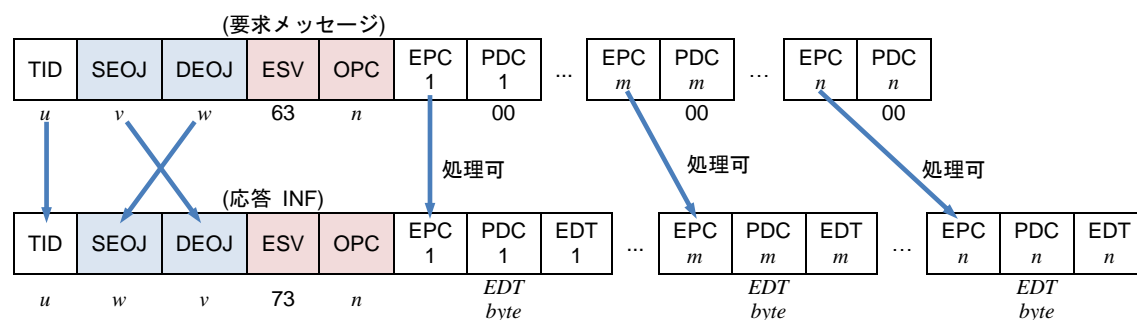
図 2.1.5-1: 通知 (応答要) のシーケンス

INFC への応答 (INFC\_Res) は、図 2.1.5-2 のようにします。応答の範囲はユニキャストです。

- 受信した EPC を列挙します。これは、要求メッセージの EPC と一致するはずですが。
- 各 EPC には、EDT を付けません。PDC には 0 バイトを入れます。
- OPC (EPC の数) には、受信した EPC の数を入れます。これは、要求メッセージの OPC と一致するはずですが。
- TID は、要求メッセージと同じ値を入れます。
- SEOJ と DEOJ が要求メッセージと逆になります。

<不可応答について>

INFC には、不可応答はありません。



値は 16 進数です。

図 2.1.4-2: INF\_REQ (ESV 0x63) への応答 (INF)

# スマートメーターと HEMS

## 第 6 回 ECHONET Lite (2)

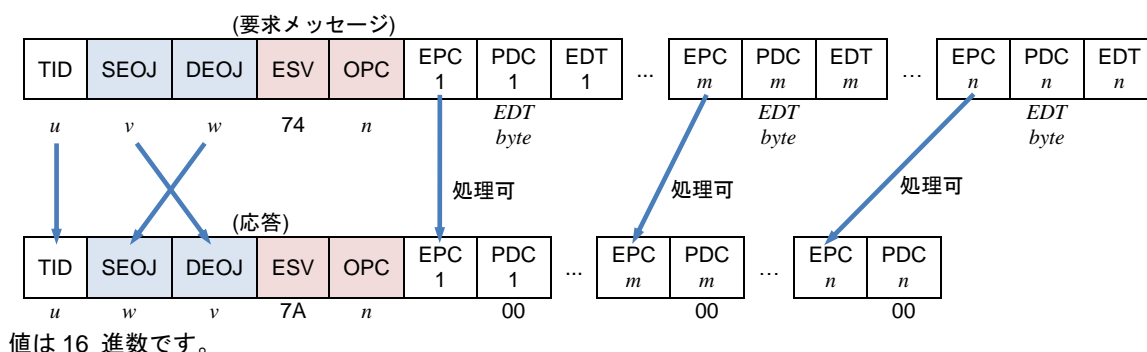


図 2.1.5-2: INFC (ESV 0x74) への応答

### 2.2 ノードがスタートする時の 基本シーケンス

機器の内部設定と、下位通信層によるネットワークに参加する過程が終わると、ECHONET Lite の出番になります。

このとき、まず、自分のことを他の ECHONET Lite 機器に知らせるために、ECHONET Lite ノードは、「インスタンスリスト通知」をブロードキャストで送信します。(文献 [1] 第 2 部 4.3.1)

シーケンスを図 2.2-1 に示します。

「インスタンスリスト通知」は、ノードプロファイルクラスに規定されているプロパティの一つです。自分 (ノード) にぶら下がっているインスタンスの数と、その EOJ コードを通知します。

文献 [1] から、「インスタンスリスト通知」のプロパティのパラメータを、表 2.2-1 に示します。

表 2.2-1: 「インスタンスリスト通知」のプロパティ

クラスグループ	プロファイルクラスグループ 0x0E
クラス	ノードプロファイルクラス 0xF0
EPC コード	0xD5
値域	<ul style="list-style-type: none"> <li>• 1 byte 目: インスタンスの総数</li> <li>• 2 byte 目以降: EOJ コード (3 byte) を列挙 (最大 253 byte 目まで)</li> </ul>
データの型	unsigned char
サイズ	253 byte (MAX.)
アクセスルール	Anno
搭載の必要性	必須
状態時アナウンス	要

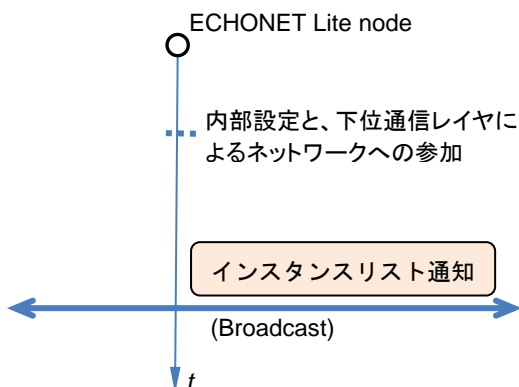


図 2.2-1: スタートする時のシーケンス

たとえば家庭用エアコンでは、この「インスタンスリスト通知」の ECHONET Lite フレームは図 2.2-1 のようになります。

## スマートメーターと HEMS

## 第 6 回 ECHONET Lite (2)

EHD 1	EHD 2	TID	SEOJ	DEOJ	ESV	OPC	EPC	PDC	EDT
10	81	xx xx	0E F0 01	0E F0 01	73	01	D5	04	01 01 30 01

値は 16 進数です。

図 2.2-1: 家庭用エアコンの「インスタンスリスト通知」

TID には、いくつが入っても構いません。また、今の場合、TID はあまり重要ではありません。TID の主な役割は、要求メッセージとその応答メッセージを関連付けることですが、このメッセージには、不可応答を含めて、応答が必要ないからです。

SEOJ は、このエアコンを一般ノードだと仮定すると、インスタンスコードが 0x01 なので、0x0E F0 01 になります。もし送信専用ノードなら、0x0E F0 02 です。ただし、送信専用ノードというのは特殊です<sup>[3]</sup>。

DEOJ は、一般ノード (0x0E F0 01) です。(送信専用ノードは、受信ができません)

ESV には、通知 (INF) の 0x73 を使います。

通知 (INF) の処理対象カウンタは、OPC です。

OPC には、このフレームで扱う プロパティは一つなので、1 を入れます。

「インスタンスリスト通知」は、他のプロパティと一緒にせず、OPC を 1 にして、単独で送信するのが望ましい、とされています。一度に一つのプ

ロパティしか処理できない ECHONET Lite ノードもあるからです。

EPC には、「インスタンスリスト通知」の EPC、0xD5 を設定します。

PDC には、EDT のバイト数を入れます。

EDT は 0x01 01 30 01 になります。

EDT の後の 3 バイトは、家庭用エアコンの EOJ コード 0x01 30 01 です。最初の 1 バイト、0x01 は、このノードのインスタンスの数で、エアコンが一つということです。

### 3 おわりに

今回は、主に、ECHONET Lite の基本シーケンスを紹介させて頂きました。ご質問などがありましたら、お問い合わせください。

約一年間、不定期に続けて参りましたが、今回で最終回となります。

最後まで読んで頂き、誠に有難うございました。

付表 1: 要求用 ESV コード

ESV	内容	記号
0x60	書き込み要求 (応答不要)	SetI
0x61	書き込み要求 (応答要)	SetC
0x62	読み出し要求	Get
0x63	通知要求	INF_REQ
0x64 ~ 0x6D	for future reserved	
0x6E	書き込み・読み出し要求	SetGet
0x6F	for future reserved	

## スマートメーターと HEMS

## 第 6 回 ECHONET Lite (2)

付表 2: 応答・通知用 ESV コード

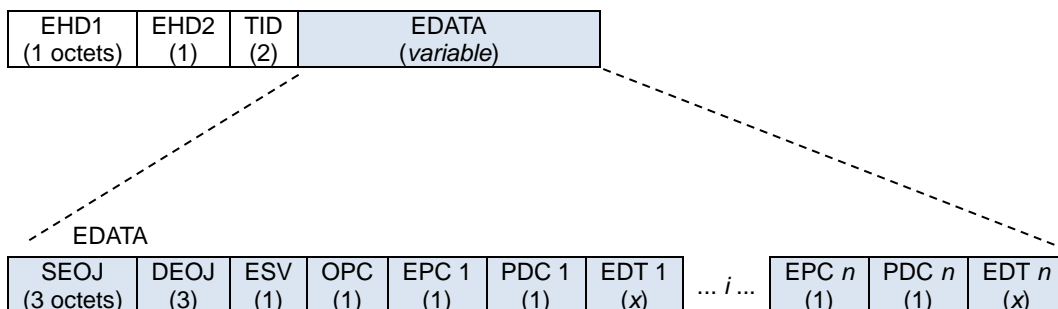
ESV	内容	記号
0x71	書き込み応答	Set_Res
0x72	読み出し応答	Get_Res
0x73	通知	INF
0x74	通知 (応答要)	INFC
0x75 ~ 0x79	for future reserved	
0x7A	通知応答	INFC_Res
0x7B ~ 0x7D	for future reserved	
0x7E	書き込み・読み出し応答	SetGet_Res
0x7F	for future reserved	

付表 3: 不可応答用 ESV コード

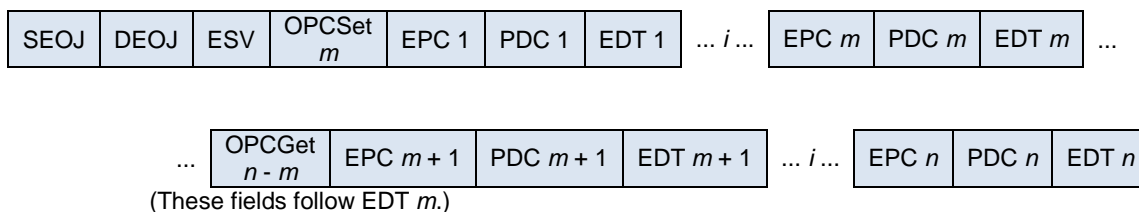
ESV	内容	記号
0x50	書き込み不可応答	SetI_SNA
0x51	書き込み不可応答	SetC_SNA
0x52	読み出し不可応答	Get_SNA
0x53	通知 不可応答	INF_SNA
0x54 ~ 0x5D	for future reserved	
0x5E	書き込み・読み出し 不可 応答	SetGet_SNA
0x5F	for future reserved	

## スマートメーターと HEMS

## 第 6 回 ECHONET Lite (2)



付図 1: フレーム フォーマット



付図 2: ESV = SetGet の EDATA

## 注

[\*1]: 日本の家電メーカーや通信会社、電力会社などが加入している団体です。人と環境に調和した豊かな社会を実現するために、1997 年に設立されました。2014 年 4 月に、法人格をもたない任意団体から一般社団法人になりました。ホームページは以下です。

<http://www.echonet.gr.jp/index.htm>

[\*2]: これを、サービスと呼んでいます。

[\*3]: 省電力に徹し、周囲に煩わされるのを嫌うノードです。

## 参考文献

[1] エコーネットコンソーシアム:「ECHONET Lite 規格書 Version 1.11」(2014 年 7 月 9 日)  
現在 最新バージョンは 1.11 です。以下のホームページから閲覧できます。

<http://www.echonet.gr.jp/spec/index.htm>

[2] エコーネットコンソーシアム:「Appendix ECHONET 機器オブジェクト詳細規定 Release F」  
(2014 年 9 月 4 日)  
ECHONET Lite 規格書の Appendix で、現在 最新バージョンは Release F です。規格書と共に、以下のホームページから閲覧できます。  
文献 [1] と同じホームページから閲覧できます。