

## ECHONET Lite 規格書のオープンデータベースシステム

原 笑† 松原 侃樹† 藤田 裕之‡ 一色 正男† 杉村 博†

神奈川工科大学 創造工学部 ホームエレクトロニクス開発学科†

神奈川工科大学 スマートハウス研究センター‡

## 1. はじめに

3.11 以降の節電意識の高まりによりスマートハウスが普及しつつある。スマートハウスとは一般的には家庭内のエネルギー消費を管理する HEMS (Home Energy Management System) が備えられている住宅のことを指しており、政府は 2030 年までに全ての住まいに HEMS を設置することを目指している<sup>[1]</sup>。経済産業省より HEMS 標準プロトコルとしてエコーネットコンソーシアムにて策定された ECHONET Lite<sup>[2]</sup>が推奨されている<sup>[3]</sup>。ECHONET Lite を利用することで機器の動作の制御や動作の状態を取得等が可能である。以上のことから ECHONET Lite 対応機器の開発、普及が期待されている。

ECHONET Lite はオープンな規格であり、誰もがダウンロードできる。ECHONET Lite プロトコルを理解するためにはまず ECHONET Lite 規格書と Appendix オブジェクト詳細規定である。しかし、PDF ファイルでの配布であるため開発者は規格書の膨大なページ数の中から自ら目を通して必要な情報を探し出すことが求められており、非効率的であると考えられる。

そこで、オブジェクト詳細規定をプログラムで検索可能になれば ECHONET Lite 対応機器の開発、製作の効率化を図ることができるのではないかと考えた。通常はデータをプログラムで検索可能にするためにはリレーショナルデータベースを利用する。しかし、ECHONET 機器オブジェクト詳細規定は 1 年に 2 度改定されており、ECHONET Lite 規格も数年に 1 度改定されているためデータベースのスキーマを定義することは難しい。

## 2. 関連技術

関連技術として神奈川工科大学スマートハウスセンターの藤田氏が開発した ELViewer<sup>[4]</sup>が GitHub で公開されている。ELViewer (ELDeviceObjectsViewer) は「ECHONET 機器オブジェクト詳細規定」をデータ化した JSON file を読み込み表示する iOS (iPhone/iPad) 用のソフトウェアである。ELViewer には Super Class を除く Release G の機器オブジェクト、Node Profile (Super Class + 機器オブジェクト)、Super Class の JSON file が含まれている。また、その JSON file も GitHub に ECHONET-APPENDIX<sup>[5]</sup>として合わせて公開されている。

本ソフトウェアはシンプルな GUI で直感的に操作、情報の取得ができる。しかし iOS (iPhone/iPad) 用のソフトウェアなので Android やパソコンからの閲覧は難しく、また XCode でビルドしてインストールする必要があるため macOS が搭載されているパソコンが必要である。

## 3. ECHONET 機器オブジェクト詳細規定を表示するオープンなシステム

このシステムを利用するユーザは ECHONET Lite 対応機器の開発者を想定しているため、システムは以下の 2 つの要求仕様を満たす必要がある。

- (1) 誰もが手軽に利用できるオープンなシステム
- (2) 規格データの変更、追加等の対応

要求(1)では、開発者各々が普段利用している端末を問うことなく容易に閲覧できるようにするため Web システムとして構築する。また、ブラウザにアクセスするだけで利用できるように、特別なソフトウェアのインストールは不要とする。

要求(2)では、ECHONET 機器オブジェクト詳細規定に誤植や変更、追加情報等あった際はユーザが修正、補足文等編集できるような wiki のように複数人で編集できるシステムにする。

## 4. 実装

Web サーバには Node.js を利用して構築した。規格書のデータベースとしてスキーマが柔軟な XML データベースが考えられるが、本システムでは JavaScript のサブセットで親和性が高く、スキーマが柔軟な JSON を用いる。JSON データベースは XML データベースと比べ簡潔に構造化されたデータを記述することが可能なので記述が容易で人間が理解しやすいデータフォーマットであると言える。JSON データベースを利用することで冗長な XML データベースと比べて通信時のデータ量を削減が期待できることから JSON データベースを利用する。

サーバ側ファイルとクライアント側の Web ブラウザ用のファイルを作成する。ファイルの役割とつながりの概要を Fig. 1 に示す。ユーザは Web ブラウザでサーバにアクセスしサーバは index.html を返す。index.html は内部で index.js と index.css が動作し、index.js はサーバから ECHONET Lite 機器オブジェクト詳細規定の JSON ファイルである deviceObject\_G.json を取得する。deviceObject\_G.json は index.js で解釈され最終的に index.html としてユーザに提供される。

Open Database System of ECHONET Lite Specification

†Emi Hara, †Naoki Matsubara, ‡Hiroyuki Fujita, †Masao Isshiki and †Hiroshi Sugimura

†Kanagawa Institute of Technology

‡Smart House Research Center Kanagawa Institute of Technology

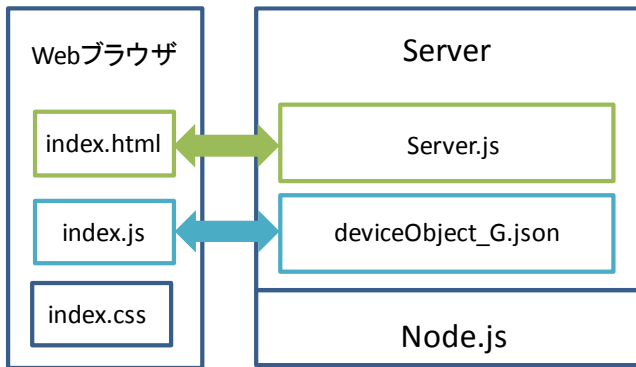


Fig. 1 ファイルの役割

実装したシステムのフローを Fig.2 に示す. Web ブラウザはサーバから HTML ファイルを取得しユーザは HTML ファイル上のリストや検索フォームから適当な機器を選択することで index.js は server.js に対応したデータを JSON ファイルから読み出して index.js に返して HTML として表示させる.

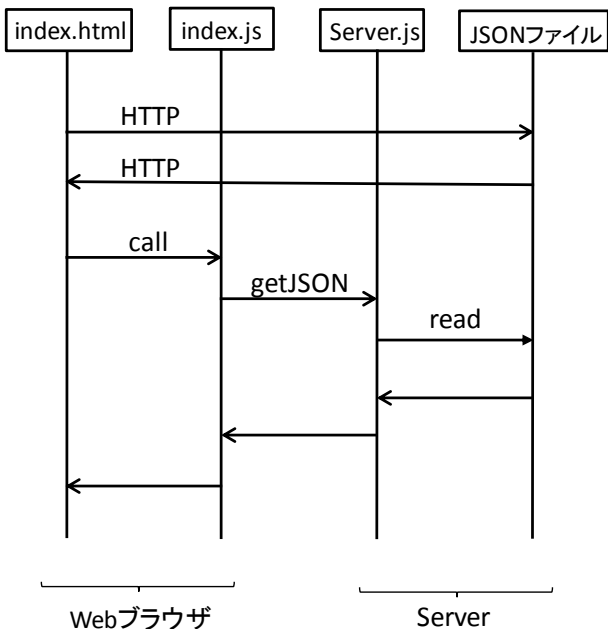


Fig. 2 実装したシステムのフロー

実際に作成したシステムの表示画面を Fig.3 に示す. 表示画面はHTMLを用いて作成した. 情報家電をグループごとにまとめ TOP ページにドロップダウンメニューで表示させ探しやすいものにする. さらに, 検索フォームで情報家電名, EPC (ECHONET プロパティ) 等のキーワードで検索可能にする. 同ページ内に該当クラス規定を表示させる. さらに, 注釈, 補足説明があるプロパティは表の下部に表示させる. 加えてプロパティ名称にリンクをつけ該当する補足文へページ内ジャンプ可能にする. 規定の内容が多い機器はページが長くなりユーザビリティが悪くなることが考えられる. そのため, ある程度ページをスクロールさせたらページのトップに戻るリンクを表示させるようにし利便性を図る.

**ECHONET Lite 規格**

キーワードを入力

---

機器オブジェクト    センサ    空調    住宅設備    調理家電    健康・管理・操作・AV

**換気扇クラス規定**

クラスグループコード:0x01 クラスコード:0x03 インスタンスコード:0x01~0x7F(0x00:全インスタンス指定コード)

プロパティ名称	EPC	プロパティ内容 値域(10進表示)	データ型	データサイズ	単位	アクセス モード	必須 項目	状態事 象	ア ナ ラ イ ズ	備 考
動作状態	0x80	ON/OFFの状態を示す ON=0x01 OFF=0x01	unsigned char	1Byte	-	Set/Get	○	○		
換気自動設定	0x8F	AUTO/非AUTO AUTO=0x41 非AUTO=0x42	unsigned char	1Byte	-	Set/Get				
換気風量設定	0xA0	換気風量レベルおよび換気風量自動状態を設定. 換気風量レベルを3段階で設定. 換気風量自動状態=0x41 換気風量レベル=0x01~0x38	unsigned char	1Byte	-	Set/Get				

(1) 動作状態(機器オブジェクトスーパークラスのプロパティを継承) 空調換気扇の運転/停止を示す. 運転/停止にそれぞれ, 0x30/0x31のプロパティ値が対応するものとする.

(2) 換気自動設定 換気自動運転のAUTO/非AUTOを設定する. AUTO=0x41, 非AUTO=0x42

(3) 換気風量設定 換気風量レベルおよび換気風量自動状態の設定を示す. 換気風量自動状態のプロパティの値は, 0x41とする. 風量レベルを3段階で設定. 0x01~0x38のプロパティ値を取るものとする. 各風量レベルの具体的な値は, 規定しないが, 0x01を風量最少, 0x38を風量最大とする.

Fig. 3 システムの表示画面

### 5. 今後の課題

規格書のデータを複数人で書き換えることで, データの信頼性の問題やデータファイルが破損する可能性が考えられる. 誰もが簡単に修正, 変更などの編集を行えるとしてもスパムや望まれない編集機能の悪用は別である. それらの防止策として最低限のアクセス管理機構の設置を検討したい. 利用者ごとの ID 発行を行い, ID を保持するものだけがデータの追加, 編集を行えるようにすることでデータの安全性の確保を図る.

### 6. おわりに

本システムはECHONET Lite 対応機器の開発, 製作の効率化を図る一助として開発した. 誰もが手軽に利用できるオープンなシステムのために node.js と JSON データベースを利用して web システムとして構築した. Web に公開し誰もが利用できる状態である. 規格データの変更, 追加等の対応させるために編集機能を加えた. 規格書の内容を充実させるために多くの開発者に使っていただきたい.

### 参考文献

- [1] 内閣官房 国家戦略室, グリーン政策大綱(2012)  
<http://www.cas.go.jp/jp/seisaku/npu/policy09/pdf/20121127/shiryoo4-1.pdf>
- [2] エコーネットコンソーシアム  
<http://www.echonet.gr.jp/>
- [3] JSCA 国際標準化 WG スマートハウス標準化検討会とりまとめの公表  
<http://www.meti.go.jp/press/2011/02/20120224007/20120224-0-07.html>
- [4] GitHub - KAIT-HEMS/ELViewer: Viewer for EL-JSON data  
<https://github.com/KAIT-HEMS/ELViewer>
- [5] GitHub - KAIT-HEMS/ECHONET-APPENDIX: ECHONET 機器オブジェクト詳細規定の JSON データ  
<https://github.com/KAIT-HEMS/ECHONET-APPENDIX>