

[研究紹介] HEMS サービス開発の要となる ECHONET Lite の 実行環境や国際競争力を考慮した SDK の開発

藤田裕之¹・関家一雄¹・笹川雄司¹・杉村博²・一色正男^{1,2}

1. 工学教育研究推進機構 スマートハウス研究センター
2. 創造工学部 ホームエレクトロニクス開発学科

Development of ECHONET Lite SDKs Considering Multiple Platforms and International Competition

Hiroyuki FUJITA, Kazuo SEKIYA, Yuji SASAGAWA, Hiroshi SUGIMURA, Masao ISSHIKI

Abstract

HEMS Certification Support Center has been doing activities to promote ECHONET Lite, communication protocol that is a key component to build Home Energy Management System (HEMS). We introduce some Software Development Kits (SDKs) we have developed in this report. In order to build HEMS, various devices and controllers need to implement ECHONET Lite protocol, and run-time environment of software varies between embed system, PC and smartphone in terms of hardware specifications and type of Operating System (OS). So we developed several Software Development Kit (SDK) to support various platforms. On the other side, there are several communication protocols for Home Automation and HEMS in the world, such as Superior Energy Performance (SEP), KNX, HomeKit and NEST. Considering a case that we need to collaborate with other protocol, we developed a SDK that can control ECHONET Lite devices utilizing HomeKit framework.

Keywords: HEMS, ECHONET Lite, communication protocol, SDK

1. はじめに

HEMS 認証支援センターではマルチベンダー機器を使用した Home Energy Management System (HEMS) を構築する上で要となる機器間通信規格の ECHONET Lite¹⁾を普及・促進するための活動を行っている。活動の一つとして ECHONET Lite 用 SDK を開発し HEMS 認証支援センターのホームページで公開している。この研究紹介ではこれまで開発した SDK と現在開発中の SDK について説明する。なお、ECHONET Lite とは IP/UDP 通信を前提とした OSI 参照層の第 5 層以上で定義された通信プロトコルである。

ECHONET Lite 用 SDK の種類としては ECHONET Lite 通信のミドルウェア、ECHONET Lite コマンドを送受信するツール、ECHONET Lite 機器エミュレーターがある。

HEMS を構築するにはさまざまな機器やコントローラ

が ECHONET Lite に対応することが必要である。ソフトウェアの実行環境としては組み込み機器や PC、スマートフォンのようにハードウェアリソース、仕様、OS 等が異なるさまざまなプラットフォームが考えられる。そこで HEMS 認証支援センターではさまざまなプラットフォームやプログラミング言語に対応した複数の SDK を開発した。

また世界的には、Home Automation や HEMS を想定した機器間通信規格として SEP²⁾、KNX³⁾、HomeKit⁴⁾、NEST⁵⁾などの規格が存在する。これらの規格と ECHONET Lite 機器の協調が必要なシステム構成の場合も考えられる。そこで Apple 社の HomeKit framework を利用して ECHONET Lite 機器を制御できる SDK を開発した。

2. SDK の概要

HEMS は家電機器や住宅設備機器の「デバイス」とデバイスを制御する「コントローラ」で構成される。デバイスやコントローラのソフトウェア開発を効率的にするためには ECHONET Lite の Middleware、ECHONET Lite のコマンドを送受信する Tool、デバイスの Emulator などの SDK が必要とされる。HEMS 認証支援センターでは Table 1 に示す SDK を開発し、公開している。

Table 1 SDK の概要

	SDK Name	Category	Platform (OS)
1	KAIT EL	Middleware	Linux
2	OpenECHO for Processing	Middleware	Windows, Mac OS, Linux
3	SSNG	Tool	Windows
4	SSNG for iOS	Tool	iOS
5	KAIT-4S -EZ-	Middleware	Android, iOS
6	KAIT-4S -Canvas-	Programing Tool	Windows, Mac OS, Linux
7	KAIT-4S -HA-	Middleware	iOS
8	Light Emulator-1	Emulator	Windows, Mac OS, Linux

Middleware を利用することで、以下の機能を容易に実装することができる。

- IP/UDP レイヤーのデータの送受信
- ECHONET Lite プロトコルパケット生成
- 受信データが ECHONET Lite 規格に準拠しているかの確認
- 受信データを ECHONET Lite 規格の各要素に分解
- 機器オブジェクトに応じた送信データの作成

ソフトウェアの実行環境 (CPU、メモリ容量、OS) はターゲットに応じてさまざまであるため、Middleware もそれぞれに対応したものが必要とされる。例えば、小規模なデバイスの場合 CPU 能力やメモリ容量は制限があり開発言語は C が用いられることが多い。一方コントローラ側は CPU 能力やメモリ容量の制限が緩やかであるが Android や iOS などの実行環境に合わせる必要がある。

デバイスの Emulator は、コントローラやサービスを開発する際に実機の代わりとして利用するものである。実機を用意できない場合や実機がサポートしていない機能の動作を確認する場合に役立つ。

Tool としては任意の ECHONET Lite コマンドを作成して UDP で送信できるツールを用意した。これとオープンソースで入手できる Wireshark (IP Packet の解析ツール) を組み合わせて利用することで動作確認や問題の切り分けが容易になる。

KAIT4S シリーズの SDK はスマートフォンや Web Browser を利用してコントローラやサービスを容易に開発できる SDK である。

3. SDK の詳細

3.1 KAIT-EL

KAIT-EL はハードウェアとして INTEL x86 系 CPU、OS として Linux という構成のプラットフォームをターゲットとした C 言語用の ECHONET Lite Middleware である。Fig. 1 に示すように、ECHONET Lite 通信処理部、機器オブジェクト、プロファイルオブジェクトで構成される。

組み込み機器の ECHONET Lite 実装に適している。重点 8 機器を含め多数の機器オブジェクトに対応し、コントローラ側・デバイス側の両方に対応する。



Fig. 1 KAIT-EL ソフトウェア構成

3.2 OpenECHO for Processing

OpenECHO⁶⁾ は Java で実装した ECHONET Lite Middleware である。OpenECHO for Processing は Fig. 2 に示すように統合開発環境の Processing 上で OpenECHO を利用できるようにしたものである。

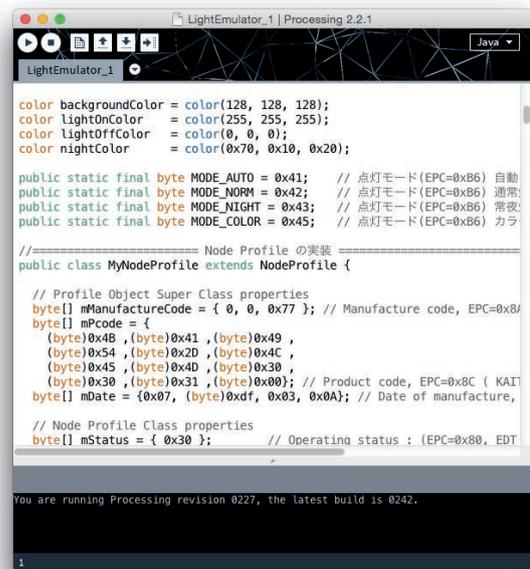


Fig. 2 OpenECHO for Processing の開発画面

OpenECHO for Processing の開発環境としては Windows/Mac/Linux がサポートされている。開発したソフトウェアは Processing 上で実行することができるだけでなく、Java の実行コードをエクスポートすることで Java Runtime Engine (JRE) がインストールされた Windows/Mac/Linux/Android で実行することができる。

OpenECHO はソニーコンピューターサイエンス研究所のオープンソースプロジェクトとして開発され Git Hub からソースコードが公開されている⁷⁾。Java を利用するためハードウェアリソースとして比較的リッチな環境が必要となるため、コントローラやエミュレーターの開発やラズベリーパイを使ったデバイスのプロトタイプングに向いている。後述する Light Emulator-1 は OpenECHO for Processing を使って開発した。

3.3 SSNG

SSNG は任意の ECHONET Lite Packet を送信できるツールである。受信したデータを表示する機能もある。ECHONET Lite デバイスの開発中に機能の動作確認をする場合や、ECHONET Lite 製品を使用したシステムを構築する場合の問題解析などに便利なツールである。

SSNG は LabVIEW を使って開発されており、Windows 用 LabVIEW の実行環境上で動作する。

ECHONET Lite Packet の TID、DEOJ、ESV、OPC、EPC、EDT などの各要素の値をプルダウンメニューで選択するか数値を直接入力することで任意の ECHONET Lite Packet を生成することができる (Fig.3 参照)。ECHONET Lite 規格では OPC にプロパティ数を設定することで複数プロパティへの問い合わせ (ESV : GET) または設定 (ESV : SET) を記述できる。SSNG では OPC = 4 までの記述ができるように 4 組の EPC と EDT の設定ができる。受信データはログデータとして表示される。

また、ECHONET Lite 機器の探索機能がある。これを使うとローカルネットワーク内の ECHONET Lite 機器を探索し、探索した機器の IP address を接続先プルダウンメニューに表示して選択することができる。

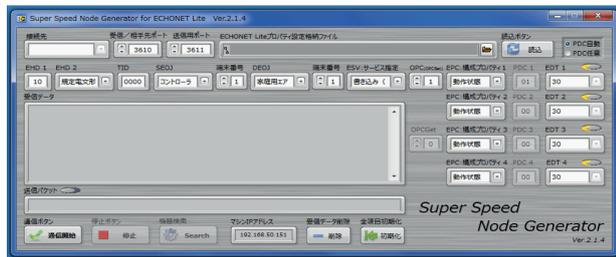


Fig. 3 SSNG の GUI 画面

3.4 SSNG for iOS

ECHONET Lite 機器の動作確認を手軽に行えるように、SSNG と同等の機能を iPhone 上で動作するツール SSNG for iOS を現在開発中である。以下の 3 点を考慮して仕様を策定した。

1. iPhone の画面サイズを考慮した GUI
表示領域が狭いため、設定できる項目は本質的なものに限定する。パラメータはデータをタイプ入力するのではなくピッカービューで選択する。
2. ECHONET Lite 規格書を見なくても使える
パラメータを選択すると、それに対応した簡単な説明を表示する。IP アドレスにはメーカーコードとメーカー名を表示する。
3. インテリジェンスの実装
IP Address を選択するとそれに対応した DEOJ の値を表示する。DEOJ を選択するとそれに対応した EPC の値を表示する。EPC を選択するとそれに対応した EDT の値を表示する。

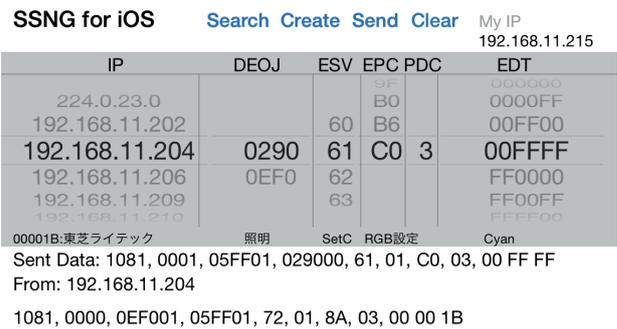


Fig. 4 SSNG for iOS の GUI 画面

3.5 KAIT-4S -EZ-

KAIT-4S -EZ-は Android スマートフォンまたは iOS デバイス (iPhone/iPad) 上で動作するコントローラアプリケーションを開発するための SDK である。Android スマートフォンや iOS デバイスで標準的に利用できるデータベースの SQLite を利用して ECHONET Lite 機器のプロパティをデータベースに定義した。アプリケーションが SQL 文で DB への read や write を行うと SDK は ECHONET Lite プロトコルで Get や Set の命令を実行する。これにより ECHONET Lite 規格の詳細な通信規約に精通していなくても、SQL のスキルがあれば ECHONET Lite 機器制御のアプリケーションプログラムを作成できる (Fig.5 参照)。また Android と iOS という異なるプラットフォームに対しても同一のモデルで記述できるので移植性が高いソフトウェアを開発できる。

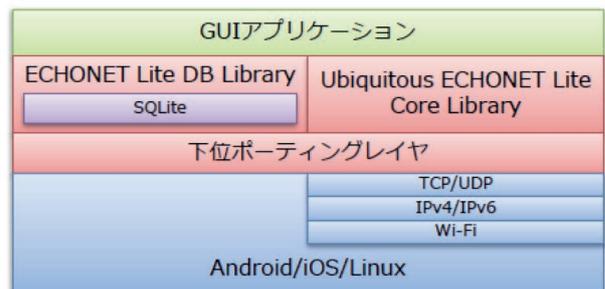


Fig. 5 KAIT-4S -EZ- の構成図

3.6 KAIT-4S -Canvas-

KAIT-4S -Canvas-は ECHONET Lite プロトコルの知識がなくても機器制御プログラムが容易に作成できる Visual Programming ツールである。ECHONET Lite 通信を使用した機器操作のトライアルや、機器連携アプリケーションのプロトタイピングに向いている。

SDK を Server にインストールし Web Browser でアクセスして利用する。条件分岐や繰り返しなどのプログラミングの骨格となるブロックに機器制御や機器状態のブロックをはめ込んでプログラムを作成する (Fig.6 参照)。



Fig. 6 KAIT-4S -Canvas- を利用したプログラム作成

プログラムを実行すると Browser は web API で Android Tablet 上の Kadecot というアプリと通信し、最終的には Kadecot が ECHONET Lite 機器と通信を行う (Fig.7 参照)。

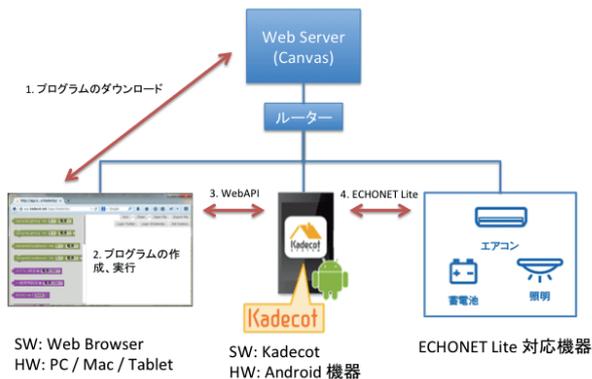


Fig. 7 システム構成図

3.7 KAIT-4S -HA-

iOS デバイスから ECHONET Lite 機器を制御するには、iOS デバイスが UDP 通信で ECHONET Lite プロトコルを送受信するソフトウェアを開発すればよい。

一方 iOS8 で導入された HomeKit framework は HEMS や Home Automation のソフトウェア開発を容易にするものである。この framework で直接制御できる機器は HomeKit プロトコルを実装したものに限られるが、それ以外の機器を制御するためにブリッジデバイスの定義がある。そこで HomeKit - ECHONET ブリッジを開発しこれを利用する HomeKit framework 用 SDK を開発した (Fig. 8 参照)。

HomeKit - ECHONET ブリッジは HomeKit プロトコルと ECHONET Lite プロトコルの変換を行う。iOS デバイ

スとの通信は Apple 社が規定した MFi⁷⁾プロトコルで行う。本ブリッジの開発は MFi 製品開発の経験があるユカイ工学株式会社に業務を委託した。

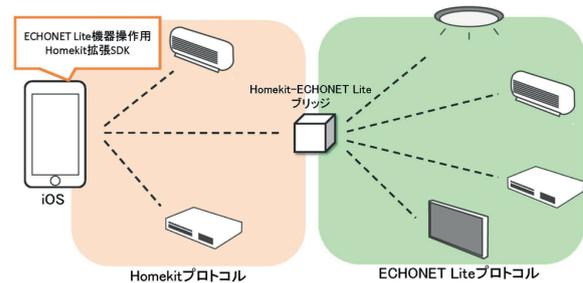


Fig. 8 KAIT-4S -HA- システム構成図

3.8 Light Emulator-1

Light Emulator-1 は OpenECHO for Processing で開発した一般照明エミュレーターである。点灯モード設定: EPC=0xB6、照度: EPC=0xB0、カラーモード時 RGB 設定: EPC=0xC0 が実装してある。ECHONET Lite 通信で照明の ON/OFF だけでなく照度レベルや色を変更できる。

エミュレーターの GUI には、照明の状態 (ON/OFF、照度レベル、色) を示す円形のビューや ON/OFF、点灯モード設定の切り替え、照度レベル設定、カラーモード時 RGB 設定などのコントロールがある。

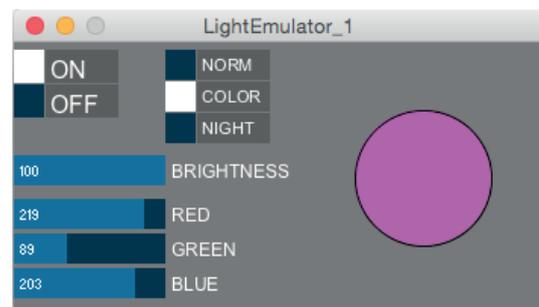


Fig. 9 Light Emulator-1 の GUI 画面

4. 終わりに

これまで開発した SDK により、ECHONET Lite の普及促進に貢献できた。今後は ECHONET Lite を利用したスマートハウス実現に向けて Internet of Things (IOT) やクラウドサービスを想定した SDK を開発する予定である。

参考文献

- [1] ECHONET Lite 規格書: <http://echonet.jp/spec/>
- [2] SEP: <http://www.energy.gov/eere/amo/superior-energy-performance>
- [3] KNX: <http://www.knx.org/knx-en/index.php>
- [4] Apple HomeKit: <https://developer.apple.com/HomeKit/>
- [5] Google NEST: <https://nest.com/works-with-nest/>
- [6] OpenECHO: <https://github.com/SonyCSL/OpenECHO>
- [7] MFi: <https://developer.apple.com/jp/programs/mfi/>