

# PUCC プロトコルを用いたスマートデバイスからの ECHONET Lite 準拠家電制御システムの開発

斉藤匠平<sup>†1</sup> 石川憲洋<sup>†2</sup> 槌屋洋亮<sup>†2</sup>

家庭における消費電力の削減や、家電機器の利便性を向上させるための仕組みとして、ホームネットワークが注目を集めている。ホームネットワークを構築するための規格には ECHONET や DLNA などがあるが、これらの規格を利用して構築されたホームネットワークは各規格に閉じたネットワークであり、互いに互換性がなく、ホームネットワーク外部からのホームネットワーク上の機器の制御や、ホームネットワーク上の機器からのクラウドサーバへのデータ送信などできない。

一方、PUCC 規格は、HGW を介して様々な規格で構築されたホームネットワークを、外部から統一的に制御及び管理出来る規格である。そこで本研究では、現在、ホームエネルギー管理システム (HEMS) などの分野で注目を集めている ECHONET Lite プロトコルを対象に、PUCC 規格を実装したスマートデバイスから、HGW を介して、ホームネットワーク上の ECHONET Lite 準拠家電を制御するシステムを開発し、その有効性を検証した。

## 1. はじめに

近年、情報家電の普及などに伴い、ホームネットワークが注目を集めている。ホームネットワークとは、主に家庭で用いられる様々な機器を接続したネットワークで、複数の機器を連携して動作させることで、エネルギー利用の効率化や、機器の利便性向上を実現することを狙いとしている。

現在、ホームネットワークを構築するための複数の規格が機器の分野ごとに制定されており、代表的なものに DLNA/IEEE1394, UPnP, ECHONET Lite などがある。中でも、近年エネルギー消費の削減が喫緊の課題となっていることから、消費電力の大きい白物家電などを対象とした ECHONET Lite が注目を集めている。

しかし、ECHONET Lite 規格を用いて構築出来るホームネットワークは ECHONET Lite 規格に閉じたネットワークであり、ECHONET Lite 規格だけではネットワーク外部からの ECHONET Lite 準拠機器の制御、機器の消費した電力情報のクラウドサーバへの送信などできないという課題がある。

また、ホームネットワークを構築するための様々な規格同士に互換性がなく、異種ホームネットワーク上の機器同士を連携させることが難しいという問題がある。

そこで本研究では、既存のネットワーク上にオーバーレイネットワークを形成し、異なるネットワークを介して機器間の接続を実現する PUCC プロトコルを用いて、スマートデバイス (タブレット端末など) から、ホームゲートウェイ (HGW) を介して、ホームネットワーク上にある ECHONET Lite 準拠の家電機器を制御するシステムを開発

した。

本論文では、まず背景として ECHONET Lite と PUCC の概要について述べる。次に、今回開発した家電制御システムのシステム構成について述べた後、開発環境及びソフトウェア開発について述べる。次に開発したシステムの動作について述べる。最後に、本開発の結果を踏まえ、まとめと今後の課題について述べる。

## 2. 背景

### 2.1 ECHONET Lite の概要

ホームネットワークには種類やメーカーが異なる様々な機器が接続される事が想定されるが、種類やメーカーが異なれば、機器が保持している情報のフォーマットや制御方法も異なるため、そのままでは機器同士を相互接続することが難しい。そのためホームネットワークを構築する際には、機器間の通信プロトコルなどを定めた共通の規格を用いる必要がある。

ECHONET (Energy Conservation and HOMecare NETwork) 規格はホームネットワーク規格の一つであり、エアコンや照明などの白物家電、電気料金モニタや各種センサなどの設備機器を対象としている。

ECHONET Lite は、ECHONET 規格の冗長な部分を改善し、プロトコルスタックを軽量化した規格である。以下の図 1 に ECHONET Lite 機器の通信レイヤ構成を示す。

<sup>†1</sup> 駒澤大学グローバル・メディア研究科

<sup>†2</sup> 駒澤大学グローバル・メディア・スタディーズ学部

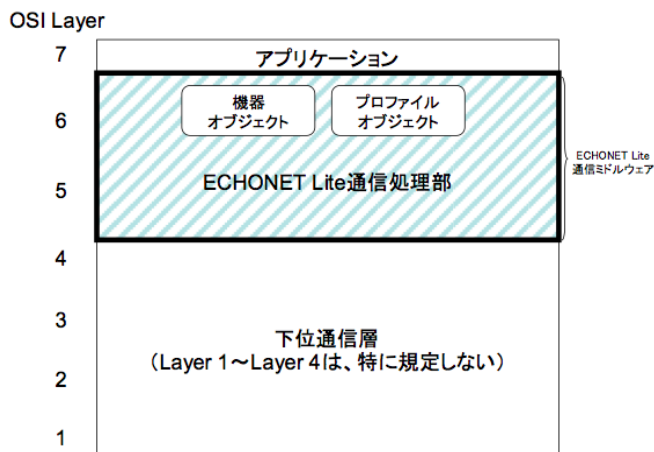


図1 ECHONET Lite 機器の通信レイヤ構成[1]

ECHONET Lite 機器の通信レイヤ構成は、アプリケーションソフトウェア、通信ミドルウェア、下位通信層の3層に分けられる。このうちECHONET Lite 規格が規定するのは、ECHONET Lite 通信ミドルウェア部分の仕様である。

ECHONET Lite 規格では、機器の持つ情報や機能を、オブジェクト指向の考えによりモデル化し、ECHONET オブジェクトとして規定することにより、機器へのアクセスインターフェイスを統一している。

ECHONET オブジェクトには、機器の種別毎に規定される機器オブジェクトと、機器に固有でないプロフィール情報についてのオブジェクトであるプロフィールオブジェクトがある。

機器オブジェクトの例として、例えばエアコンにはエアコン機器オブジェクトが規定されており、エアコン機器オブジェクトは動作状態、運転モード、室内温度などのプロパティ値を持っている。メーカーや機器種別が異なっても、同じ操作でこれらのプロパティを参照・設定することにより、機器からの情報取得、機器の制御などを実現できる。

ECHONET Lite 通信ミドルウェア同士はECHONET Lite フレームと呼ばれる電文をやり取りすることで、ECHONET オブジェクトを取得・操作し、機器情報の取得や機器の操作を行う。図2にECHONET Lite フレームの電文構成を示す。

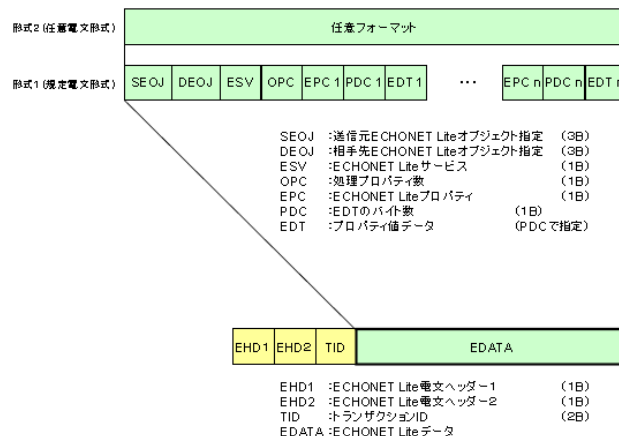


図2 ECHONET Lite フレームの電文構成[2]

ECHONET Lite フレームは相手先のECHONET Lite オブジェクトや要求の種類(状態取得や書き込みなど)についての情報を持つ。

ECHONET Lite フレームがECHONET Lite 通信ミドルウェアによって処理され、ECHONET オブジェクトへの書き込みが行われると、その値がECHONET Lite 通信ミドルウェアからアプリケーションソフトウェアに渡され、処理される事で、機器が制御される。

## 2.2 PUCCの概要

### 2.2.1 PUCCのプロトコルスタック

先述したように、ホームネットワーク規格は複数存在し、互いに互換性がなく、異なるネットワーク上の機器へのアクセスが難しいなどの課題が存在している。また、これらの規格によって構築されたホームネットワーク上の機器を、外部からのアクセスすることが困難であるなどの課題も存在している。

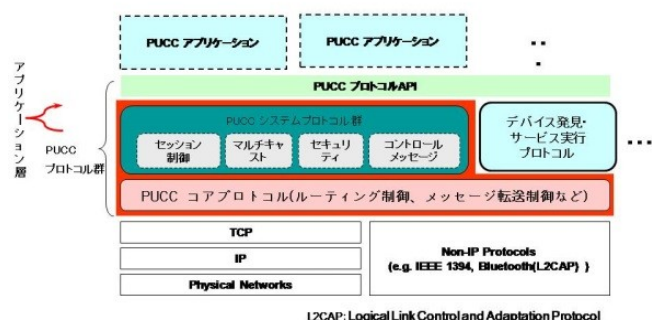
これらの課題を解決するために、P2P/オーバーレイネットワーク技術とデバイスを制御するためのデバイス・メタデータ技術を利用したホームネットワーク制御技術がPUCC(P2P Universal Computing Consortium)によって開発され、規格化されている。PUCCの最新規格は、2012年3月に制定されたPUCC仕様リリース3である(図3)。

- (1) PUCC Architecture Ver. 2
- (2) PUCC Basic Protocol Ver. 3
- (3) PUCC Basic Protocol - Light Profile Ver. 2
- (4) PUCC Device Discovery Service Invocation Protocol Ver. 3
- (5) PUCC Printing Protocol Ver. 3
- (6) PUCC Imaging Protocol Ver. 2
- (7) PUCC Streaming Protocol Ver. 3
- (8) PUCC Device and Service Metadata Template Ver. 3
- (9) PUCC Metadata Specification –
  - Part1: IEEE 1394 Devices Ver. 1
  - Part2: UPnP Devices Ver. 2
  - Part3: ECHONET Devices Ver. 1
  - Part4: Other Devices Ver. 1
  - Part5: Camera Device Ver. 1
  - Part6: RFID Reader Device Ver. 1
  - Part7: Felica Reader Device Ver. 1
  - Part8: Barcode Reader Device Ver. 1
  - Part9: Location Information Ver. 1
  - Part10: IEEE 11073 Device Ver. 1
  - Part11: OSGI-based Device Ver. 1

### PUCC仕様リリース3

図 3 PUCC 仕様一覧

PUCC では、既存の通信規格の上位プロトコルとして、アプリケーション層にオーバーレイプロトコルを定義している。このプロトコルを PUCC プロトコルと呼ぶ。PUCC プロトコルのプロトコルスタックを図 4 に示す。



### PUCCプロトコルスタック

図 4 PUCC プロトコルスタック

PUCC プロトコルは、TCP/IP だけでなく、Bluetooth (L2CAP)、IEEE1394、OBEX/IrDA などの非 IP プロトコルも含む既存の通信規格の上位プロトコルとして、アプリケーション層で定義されたプロトコルあり、異種ネットワークを介してオーバーレイネットワークを構築し、機器間のシームレスな相互接続を実現している。

PUCC プロトコルはアプリケーションに依存しない汎用的なミドルウェアとして開発可能であり、PUCC プロトコル API を用いることで PUCC プロトコル群を用いた PUCC アプリケーションを開発することができる。

### 2.2.2 PUCC デバイス発見・サービス実行プロトコル

PUCC プロトコルでは様々な機器を統一的に制御、管理することを可能とするために、ネットワーク上のデバイスを発見して、機器制御などのサービスを実行するデバイス発見・サービス実行プロトコルを定義している。

PUCC 規格では、機器の名称、種類、属性、機器が提供するサービスなどを記述するデバイス・メタデータを定義している。図 3 に示すように、ECHONET、UPnP/DLNA などの既存ホームネットワーク規格と互換性のあるデバイス・メタデータを定義している。これらのメタデータと PUCC プロトコルは汎用的な設計となっており、プロトコル変換と PUCC デバイス・メタデータへの変換を行うことにより、既存ホームネットワーク規格を用いて構築されたホームネットワーク上の機器を発見し、その機器を制御、管理することを可能としている。

デバイス発見・サービス実行プロトコルのメッセージシーケンス例を図 5 に示す

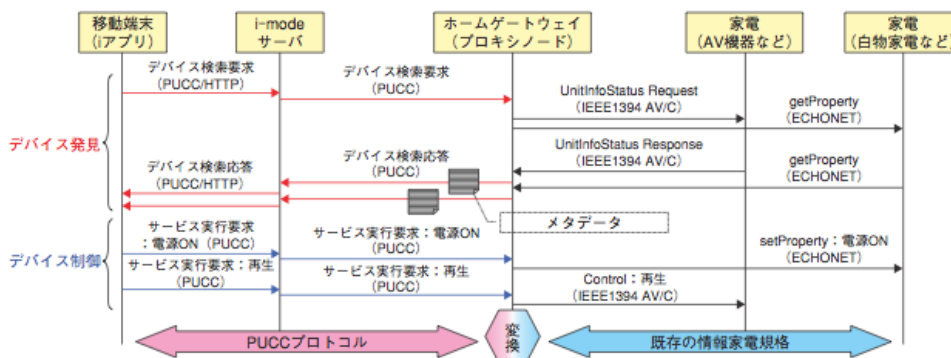


図 5 デバイス発見・サービス実行プロトコルのメッセージシーケンス例[3]

図 5 に示すように、PUCC 規格を実装した携帯電話等から既存の規格を用いて構築されたホームネットワーク上の機器を制御、管理する際には、HGW にて、PUCC プロトコルから既存通信プロトコルへの変換を行う。加えて、既存の規格に準拠した機器に対応する PUCC デバイス・メタデータを HGW 上で生成し、これらのメタデータを利用して、HGW 上で機器を制御するメッセージに変換することにより、既存の規格を利用して構築されたホームネットワーク上の機器とその機器が提供する機能を動的に発見して制御を行う。

### 2.2.3 PUCC デバイス・メタデータ

PUCC 規格では、デバイス・メタデータを XML を用いて記述している。PUCC で規定しているデバイス・メタデータの概要を図 6 に示す。

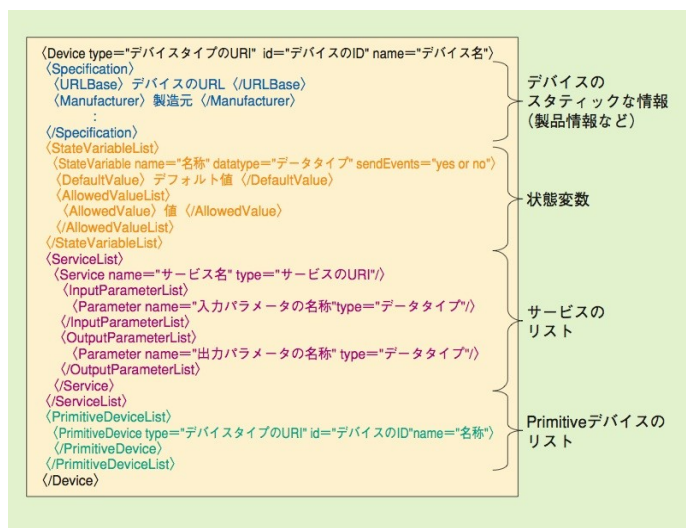


図 6 PUCC デバイス・メタデータ[4]

図 6 に示すように、PUCC では、デバイスのメタデータとして、デバイスの名称、種類、製品番号などのスタティック情報、デバイスが提供するサービス、デバイスの状態変数（例えば、センサデバイスが取得したセンサデータなど）などを規定している。加えて、PUCC デバイス・メタデータは、デバイスのメタデータを Primitive デバイスのリストとして記述することが可能であり、この機能を利用して、複数のデバイス機能（例えば、DVD プレイヤーと HDD レコーダなど）を持つデバイスを、複数の Primitive デバイスから構成される複合デバイスとして記述することを可能としている。

## 3. システム開発

### 3.1 システム構成

今回開発したシステムの構成を図 7 に示す。

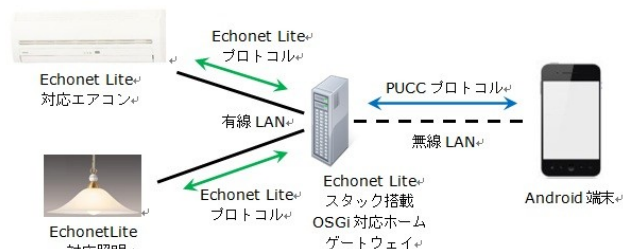


図 7 システム構成

本システムは、ECHONET Lite 準拠のエアコンと照明を、OSGi 対応 HGW を介して、Android 端末（スマートフォンまたはタブレット）から制御するシステムである。

エアコンおよび照明と Android 端末の間には、予め ECHONET Lite を実装した ECHONET Lite バンドルを搭載した OSGi 対応 HGW を設置している。Android 端末は無線 LAN で HGW と接続し、PUCC プロトコルを用いて通信する。エアコンおよび照明は、有線 LAN で HGW と接続し、ECHONET Lite プロトコルを用いて通信する。

HGW 上で PUCC プロトコルと ECHONET Lite プロトコルを変換することにより、Android 端末からの ECHONET Lite 準拠家電の制御を実現する。

### 3.2 開発環境

#### 3.2.1 システム開発環境

##### (1) エアコン・照明

今回開発したシステムで用いるのは、市販されている ECHONET Lite 準拠のエアコン及び照明である。

##### (2) HGW

今回開発したシステムでは OSGi 準拠の HGW を用い、HGW 上には OSGi 準拠 ECHONET Lite バンドルが搭載されている。

HGW に関してはソフトウェア開発環境(SDK)が提供されており、SDK を利用して OSGi バンドルの開発が可能である。

##### (3) スマートデバイス

本システムで用いるスマートデバイスは、Android OS のバージョン 4.1 以上のタブレット端末及びスマートフォンである。

#### 3.2.2 ソフトウェア開発環境

ソフトウェアの開発には PUCC ミドルウェアプラットフォーム

ホームを使用した。図 8 に PUCC ミドルウェアプラットフォームの構成要素を示す。

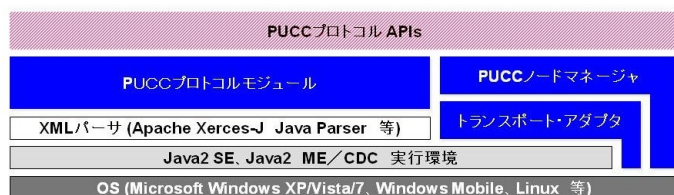


図 8 PUCC ミドルウェアプラットフォームの構成要素

PUCC ミドルウェアプラットフォームは Java 環境 (Java2 SE/ME) で実装されており、Windows や Linux など様々な OS 上で動作する。

PUCC ミドルウェアプラットフォームは PUCC プロトコル API を提供し、PUCC プロトコルを用いるアプリケーションの開発を可能とする。

PUCC ミドルウェアプラットフォームの主な構成要素は、PUCC プロトコルモジュール、PUCC ノードマネージャ、トランスポートアダプタである。

PUCC プロトコルモジュールは PUCC コアプロトコル、PUCC システムプロトコル、PUCC アプリケーションプロトコルを含む PUCC プロトコル群を実装している。PUCC ノードマネージャは PUCC ノードの管理を行い、PUCC トランスポートアダプタは PUCC プロトコルの下位層トランスポートプロトコルへのマッピングを行う。

### 3.3 ソフトウェア開発

#### 3.3.1 HGW

今回のシステムを開発するにあたって、PUCC アプリケーションを HGW 上で用いるために、PUCC ミドルウェアプラットフォームを OSGi 対応 HGW 上で動作する OSGi バンドルの形に変換した。

また HGW 上で動作する、HGW、エアコン、照明の PUCC ノードに対応する PUCC アプリケーションを開発した。

#### 3.3.2 Android 端末

PUCC アプリケーションを Android 端末で用いるために、PUCC ミドルウェアプラットフォームを Android 端末上に移植した。

また Android 端末上で動作し、PUCC ノードとしてユーザーインターフェースの提供などを行う PUCC アプリケーションを開発した。

図 9 に HGW 及び Android 端末のソフトウェア構成を示す。

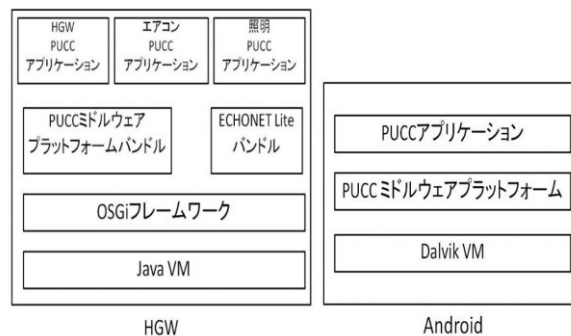


図 9 HGW と Android 端末のソフトウェア構成

### 3.4 システムの動作

#### 3.4.1 起動時の動作

HGW が起動すると、PUCC のミドルウェアプラットフォームバンドルの PUCC ノードマネージャが起動する。次に、PUCC ノードマネージャが HGW の仮想 PUCC ノードアプリケーションを起動する。

エアコン及び照明は、起動して HGW に接続されると、ECHONET Lite 規格で構築されたネットワーク上に自機器の情報を一斉同報送信する。HGW 上ではエアコンあるいは照明の ECHONET Lite 準拠家電が発見された時点で、PUCC ノードマネージャがそれぞれに対応する PUCC ノードアプリケーションを起動する。

Android 端末の PUCC アプリケーションが起動した際は、PUCC プロトコルを用いて HGW 上の仮想 PUCC ノードアプリケーションを探索し、発見に成功すれば PUCC セッションを張る。

#### 3.4.2 機器の探索・制御時の動作

本システムでは Android 端末を利用して、PUCC デバイス発見・サービス実行プロトコルを実行することにより、ECHONET Lite 準拠エアコンと照明を探索し、制御する。

図 10 はデバイスを発見し PUCC メタデータを取得するまでの流れを表したシーケンス図である。

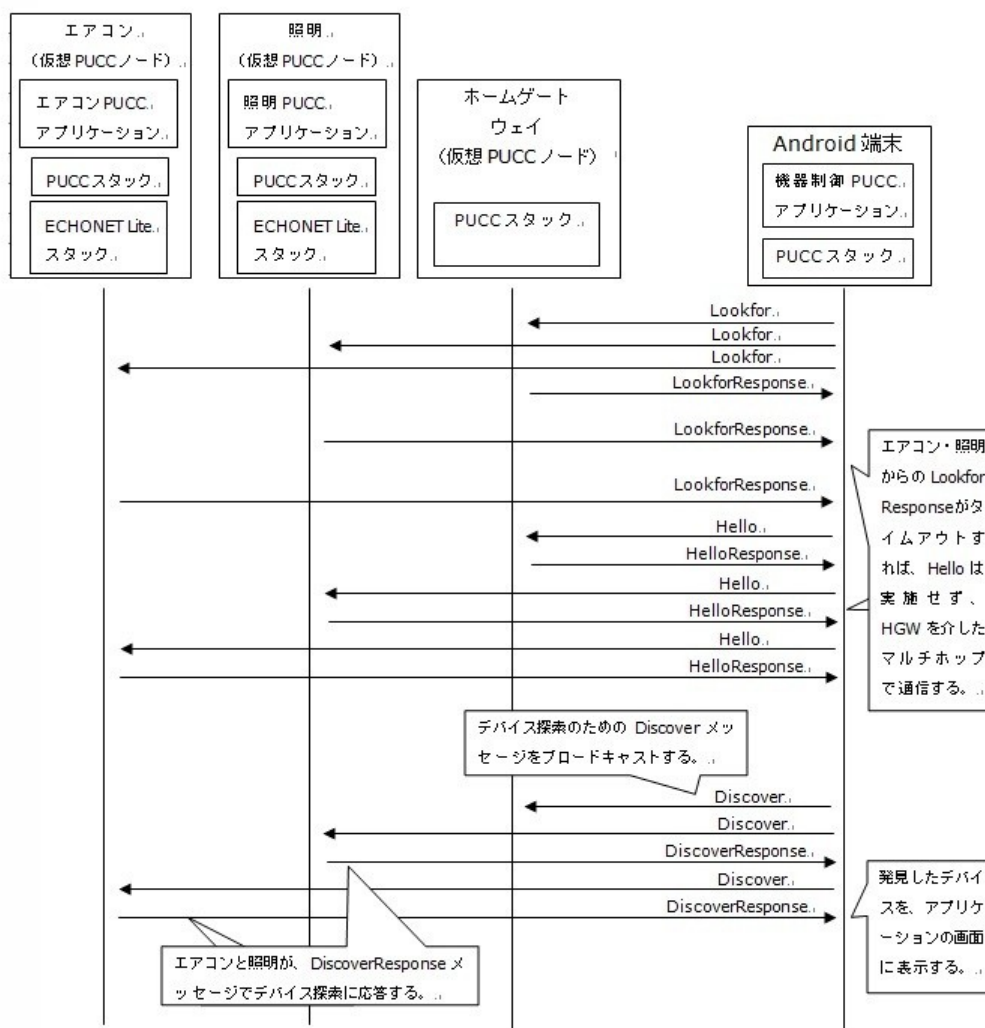


図 10 PUCC ネットワークの構築及びデバイス探索のシーケンス

PUCC プロトコルを用いた機器の探索・制御の流れは次のようなものである。

①Android 端末の PUCC ノードは、起動時に Lookfor メソッドで PUCC ノードを探索する。

②Android 端末の PUCC ノードは、Lookfor メソッドで発見した PUCC ノード (HGW, エアコン, 照明の最大 3 つ) に対し、Hello メソッドで PUCC セッションを構築する。

③Android 端末の PUCC ノードは、Discover メソッドでエアコン及び照明の PUCC ノードが持つ、PUCC デバイス・メタデータを取得することで、機器の URI, 名称, 利用できるサービスの情報などを取得する。

図 11 は、本システムで用いた照明の PUCC デバイス・メタデータである。

続いて、発見したデバイスのメタデータを利用して、デバイス制御を行う。

図 12 はその流れのエアコンの例を表したシーケンス図

である。

PUCC デバイス・メタデータを用いたデバイス制御は次のように行う。

④Android 端末の PUCC ノードは、発見したデバイス (エアコン, 照明) に対し、Invoke メソッドでデバイスの状態取得, 及びデバイス制御を実施する。

具体的にはエアコン及び照明の PUCC ノードに対して、取得した PUCC デバイス・メタデータから、利用するサービスの URI とそのサービスで利用するパラメータを選んで送信する。

⑤エアコン及び照明の PUCC ノードアプリケーションがサービス利用のための Invoke メッセージを受け取ると、PUCC デバイス・メタデータを ECHONET オブジェクトに変換する。

⑥変換された ECHONET オブジェクトを用いて、HGW の ECHONET Lite バンドルが相手先の機器にプロパティ値の書き込み (SetC) 要求を送り、機器を制御する。

```
<Device type="http://www.pucc.jp/2006/04/Device/Light" id="Light:1" name="居間の照明">
  <Specification>
    <Manufacturer>東芝家電製造株式会社</Manufacturer>
    <ManufacturerURL>http://www3.toshiba.co.jp/feminity/</ManufacturerURL>
    <ModelDescription>照明</ModelDescription>
    <ModelName>東芝Feminity</ModelName>
    <ModelNumber>TOSHIBA LIGHT 00z000Z</ModelNumber>
    <ModelURL>http://www3.toshiba.co.jp/feminity/feminity_eng/service/service_06.html</ModelURL>
    <SerialNumber>1234567890</SerialNumber>
    <IconList>
    </IconList>
  </Specification>
  <StateVariableList>
    <StateVariable name="OperationStatus" datatype="string" sendEvents="yes">
      <AllowedValueList>
        <AllowedValue>ON</AllowedValue>
        <AllowedValue>OFF</AllowedValue>
      </AllowedValueList>
    </StateVariable>
  </StateVariableList>
  <ServiceList>
    <Service type="http://www.pucc.jp/2006/04/Device/Light/Service/GetOperationStatus" name="GetOperationStatus">
      <InputParameterList>
      </InputParameterList>
      <OutputParameterList>
        <Parameter name="operationStatus" datatype="string" relatedStateVariable="OperationStatus" />
      </OutputParameterList>
    </Service>

    <Service type="http://www.pucc.jp/2006/04/Device/Light/Service/SetOperationStatus" name="SetOperationStatus">
      <InputParameterList>
        <Parameter name="operationStatus" datatype="string" relatedStateVariable="OperationStatus" />
      </InputParameterList>
      <OutputParameterList>
      </OutputParameterList>
    </Service>
  </ServiceList>
</Device>
```

図 11 照明の PUC デバイス・メタデータ

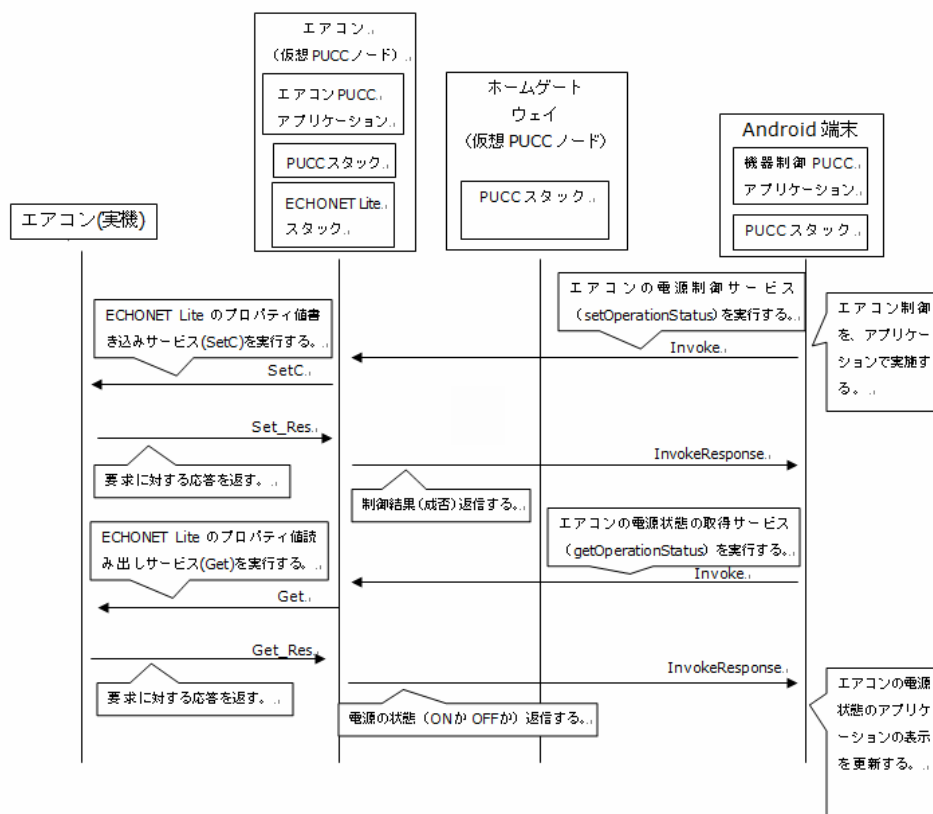


図 12 デバイス制御 (例: エアコンの電源操作) のシーケンス

図 13 は、Android 上で動作する PUCC アプリケーションの、実際の画面である。

アプリケーションを起動し、デバイス探索を行うと、図 13 (a) の画面 1 のように操作できる機器が画面に表示される。

その状態からエアコンを選択すると、図 13 (b) の画面 2 のような画面が表示され、機器の機能を選び制御する事ができる。



(a)画面1

(b)画面2

図 13 Android アプリケーションの画面

#### 4. まとめ

本論文では、ECHONET Lite プロトコルと PUCC プロトコルを変換する機能を実装した OSGi 準拠の HGW を用いて、スマートデバイスから ECHONET Lite 準拠家電を制御するシステムの開発について述べた。

本システムの開発の結果、実際にスマートフォンやタブレット端末を用いて、エアコンや照明を制御する事が可能である事が確認出来た。

本研究以外にもホームネットワークの外部からホームネットワーク上の機器を制御する研究は存在する[5]が、既存の研究はクラウドサーバなどを活用する大規模なシステムを構築するものであり、本研究のようにユーザーが操作する端末からホームネットワークに接続し、直接機器を制御する、ホームネットワークシステムについての研究は他に存在しない。

本研究はスマートデバイス、HGW、家電機器による、低コストなスマートデバイスからの家電制御システムの実現可能性を示した。

今後は本研究の成果を踏まえた上で、ホームエネルギー制御システム (HEMS) の開発を行う。

特に消費電力量に応じた機器の制御が重要であると考えられる。しかし消費電力に関する ECHONET オブジェクトは存在するが、対応する PUCC メタデータが存在しないので、消費電力量 PUCC メタデータの作成と実装を進めていく予定である。

**謝辞** 本研究は、株式会社日立製作所にご協力を頂き実現しました。ここに厚く御礼を申し上げます。

#### 参考文献

- [1] ECHONET CONSORTIUM: ECHONET Lite 規格書, Ver1.11(日本語版),第1部 ECHONET Lite の概要 (2001).
- [2] ECHONET CONSORTIUM: ECHONET Lite 規格書, Ver1.11(日本語版),第2部 ECHONET Lite 通信ミドルウェア仕様 (2001).
- [3] 角野宏光, 内田良隆, 小佐野智之, 石川憲洋: 移動端末からの情報家電・プリンタ制御に関する PUCC 実証実験, NTT docomo テクニカルジャーナル, vol14, no4, pp. 13-17 (2007).
- [4] 石川憲洋, 角野宏光, 加藤剛志, 内田良隆: 生活ケータイの新たな利用シーンを創出する移動端末からのホームネットワーク制御技術, NTT docomo テクニカルジャーナル, vol16, no1, pp. 26-35 (2008).
- [5] 増尾剛, 中村二郎, 松岡茂登, 長谷川剛, 村田正幸, 松田和浩: リアルタイム Web 技術による HEMS サービスクラウド化の検討, 電子情報通信学会技術研究報告. NS, ネットワークシステム 112(350), pp. 1-6 (2012).