

## 柔軟な設定変更が可能なレガシー家電機器用スマート化システムの設計

上野 大輝<sup>†1</sup> 岡田 真実<sup>†2</sup> 鈴木 秀和<sup>†2</sup><sup>†1</sup> 名城大学工学部 <sup>†2</sup> 名城大学大学院理工学研究科

## 1 はじめに

近年, ECHONET Lite などの通信規格を利用して機器を制御できる情報家電が普及しつつある. しかし, 現在の家庭には通信機能を持たない家電機器 (レガシー家電機器) も多数存在する. 汎用赤外線リモコンやマイコンに ECHONET Lite を搭載することにより, レガシー家電機器を ECHONET Lite で制御する仕組みを実現した研究 [1] があるが, 1 台のマイコンで 1 台の特定のレガシー家電機器しか制御できない.

そこで本稿では, 1 台のマイコンで複数のレガシー家電機器を制御することが可能で, かつ Web ブラウザから柔軟にマイコンの設定を変更可能なスマート化システムの設計について述べる.

## 2 設計システム

## 2.1 概要

本システムでは Raspberry Pi を介して, タブレットなどの操作端末からレガシー家電機器を ECHONET Lite 機器として操作することを実現する. 図 1 に本システムの構成を示す. 本システムでは, Raspberry Pi に赤外線送信機能と ECHONET Lite 通信機能を実装し, 家電が保持する情報や操作可能な機能をモデル化した ECHONET Lite オブジェクトを Raspberry Pi 内に組み込む. これにより, タブレットなどの操作端末からレガシー家電機器を ECHONET Lite 通信で制御することが可能となる. また, Raspberry Pi に複数の ECHONET Lite 機器オブジェクトを生成することにより, 複数のレガシー家電機器をスマート化させることができる.

さらに, HTTPS 通信を用いて ECHONET Lite オブジェクトの種類や数を変更することにより, 操作可能なレガシー家電機器を動的に変更できる.

## 2.2 レガシー家電機器の ECHONET Lite 対応化

Raspberry Pi に ECHONET Lite オブジェクトを生成するために, ECHONET Lite のオープンソース実装である OpenECHO [2] を Raspberry Pi に組み込む.

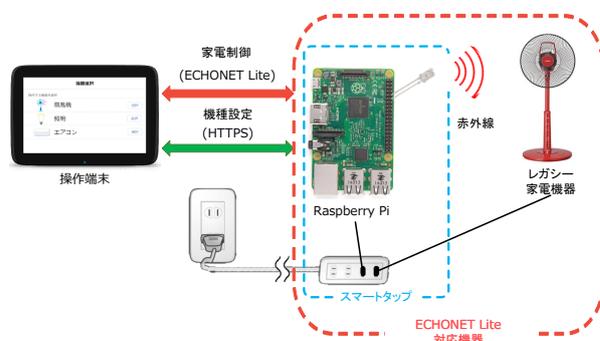


図 1 システムの構成図

OpenECHO の機器クラスを参照し, ECHONET Lite の制御信号に従って赤外線信号を送信する機能を持った機器のオブジェクトを作成する. このオブジェクトを呼び出すことにより, ECHONET Lite 通信を行うことが可能となる.

Raspberry Pi が起動する際に設定ファイルが読み込まれ, 登録されているレガシー家電機器に対応した ECHONET Lite オブジェクトが自動的に生成される. 設定ファイルは次節の操作により作成される.

## 2.3 Web ブラウザを利用した ECHONET Lite 機器の設定変更

制御したいレガシー家電機器の ECHONET Lite オブジェクトを生成するために, ユーザは Web ブラウザを利用してレガシー家電機器の情報や赤外線信号を登録を行う. レガシー家電機器の登録には, 機器名, 機器の種類, メーカー名, 等を入力または選択し, 赤外線信号の登録では Raspberry Pi に向けてレガシー家電機器のリモコン操作を行う. 具体的には赤外線送受信モジュールの受信部分にリモコンの送信部を向けて, 登録したい機能のボタンを押すことにより, 当該操作に関わる赤外線信号を登録できる. 赤外線の登録が完了すると機器の情報が設定ファイルに保存される.

## 3 実装

## 3.1 実装

機器の制御を行うための赤外線送受信プログラムと機器登録や機器確認を行うことができる Web システムを実装した. 赤外線受信プログラムは赤外線送受信モジュールが受信した赤外線の信号データをテキストファ

イルに保存するプログラムである。赤外線送信プログラムは信号データが保存されているテキストファイルを読み込み、赤外線送受信モジュールから赤外線を送信するプログラムである。

Web ブラウザで Raspberry Pi にアクセスすると機器登録や登録済み機器の確認といった項目が表示され、項目を選択すると各処理フローに進む。機器登録は 2.3 節で説明した手順で行う。登録済み機器の確認は機器の情報が保存されている設定ファイルからデータを取得し、登録されている機器の機器名を一覧で表示する。

### 3.2 動作確認

実装したシステムの動作確認を行った。今回の動作確認では照明、扇風機の 2 種類のレガシー家電機器を使用し、コントローラには ECHONET Lite 機器を検知し制御可能なアプリである iHAC アプリケーション [3] を使用した。

Web ブラウザから Raspberry Pi にアクセスし、登録画面からレガシー家電機器の情報および赤外線信号の登録を行う。登録する赤外線信号として、照明及び扇風機の電源 ON/OFF を登録する。その後、iHAC アプリケーションを起動し、照明および扇風機の電源を ON にする制御を行った。動作確認の結果、iHAC アプリケーション上にレガシー家電機器が認識され、電源 ON を選択すると実際にレガシー家電機器の電源が ON になり、1 台のマイコンでレガシーな照明と扇風機を正しく制御できることを確認した。

## 4 評価

### 4.1 測定方法

本システムにおける処理が ECHONET Lite で規定されているタイムアウト時間内に完了するかを確認するために、RTT と処理時間を評価する。図 2 に測定環境を示す。コントローラとして ECHONET Lite の電文を送受信できるプログラムを作成した。市販の ECHONET Lite 機器は TOSHIBA の LEDD-LT1 (照明) を使用し、PC, Raspberry Pi, LEDD-LT1 を同一ネットワークに接続し、コントローラから対象機器に制御メッセージを送り、実際に機器が動作して応答が返ってくるまでの時間を計測する。試行回数は 10 回で、計測時間から通信遅延時間を除くことにより ECHONET Lite に関わる処理時間を算出した。なお、比較のために市販の ECHONET Lite 照明を制御するのに要した時間も合わせて計測した

### 4.2 結果

表 1 に計測結果の平均値を示す。制御メッセージを送信してから応答を受信するまでの時間は、市販の ECHONET Lite 家電機器の場合が 102.61[ms]、提案システムが 203.52[ms] だった。これらの計測時間か

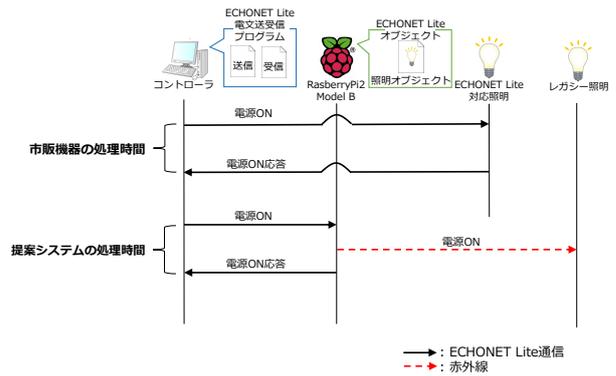


図 2 測定環境

表 1 RTT 測定

	市販機器	提案システム
処理時間 [ms]	192.61	203.52
RTT[ms]	0.11	0.21

ら RTT を除いた 102.50[ms] および 203.31[ms] が市販の ECHONET Lite 家電機器、提案システムにおける Raspberry Pi で動作した処理時間と見積もることができる。各機器における処理時間を比較すると、提案システムは市販の機器の約 2 倍かかることがわかる。これはレガシー家電機器を操作するための赤外線信号送信処理を実施している影響だと考えられる。

しかし、ECHONET Lite では機器が要求を受信してから応答を返信するまでのタイムアウト時間を 5000[ms] と規定されている [4]。本システムの処理時間はタイムアウト時間のわずか 4.07% に過ぎないため、提案システムの処理時間は実用上問題ないといえる。

## 5 まとめ

本稿では、1 つのマイコンで複数のレガシー家電機器を制御することが可能で、かつ Web ブラウザから柔軟にマイコンの設定を変更可能なスマート化システムを設計し、動作検証により実用上問題ない時間で処理できることがわかった。今後はより多くの機器を制御できるようにシステムを拡張する。

### 参考文献

- [1] 武内. 他 : CDS, 情報処理学会研究報告, Vol.2015-CDS-13, No.2, pp.1-4, 2015.
- [2] GitHub-SonyCSL/OpenECHO : <https://github.com/SonyCSL/OpenECHO>
- [3] 梅山. 他 : 情報処理学会論文誌 (CDS), Vol.6, No.1, pp.84-93, 2016.
- [4] 第 2 部 ECHONETLite 通信ミドルウェア仕様 : ECHONET Consortium, 2015.