

# 組込みコンポーネントシステムに基づくビジュアルプログラミング 対応した ECHONET Lite フレームワーク

齊 峰<sup>†</sup> 大山 博司<sup>‡</sup> 長島 宏明<sup>\*</sup> 安積 卓也<sup>†</sup>

埼玉大学<sup>†</sup> オークマ<sup>‡</sup> コアーズ<sup>\*</sup>

## 1. はじめに

スマートホームは快適で便利な生活環境を提供できる。近年、IoT デバイスの増加に伴い、スマートホーム間の接続はより複雑になってきている。デバイスソフトウェアはデバイスの状態を制御しながら、システムとの通信を処理する必要がある。スマートホームには多くの機能があるため、既存のデバイスソフトウェアのメンテナンス性が低いという問題がある。さらに、ユーザにとっては、従来のアプリケーションと比べて、スマートホームを個人の環境やライフスタイルに合わせてカスタマイズできるツールが必要になる。

本研究では、組込みコンポーネントシステムに基づくビジュアルプログラミング対応したスマートホームフレームワークを提案する。組込みコンポーネントシステムはデバイスソフトウェアの通信部と制御部を分離し、デバイスをコンポーネント化する。ビジュアルプログラミング環境をユーザインタフェースとして、スマートホームを制御する。スマートホーム仕様書で記述されるデバイス情報によって、コンポーネント記述言語、ソフトウェアドライバとユーザインタフェースを自動生成するコードジェネレータを開発することで、前述の問題を解決する。

## 2. システムモデル

本章ではシステムモデル図 1 の構成要素について説明する。ターゲットボード上でシステムが実行され、TECS によってデバイスの制御を定義し、ECHONET Lite ミドルウェアから通信処理を行う。ユーザは Blockly を介してコマンドを送信する。

### 2.1. TECS

TOPPERS Embedded Component System (TECS) は組込みシステムを開発するために設計されたコンポーネントシステムである。組込みソフト

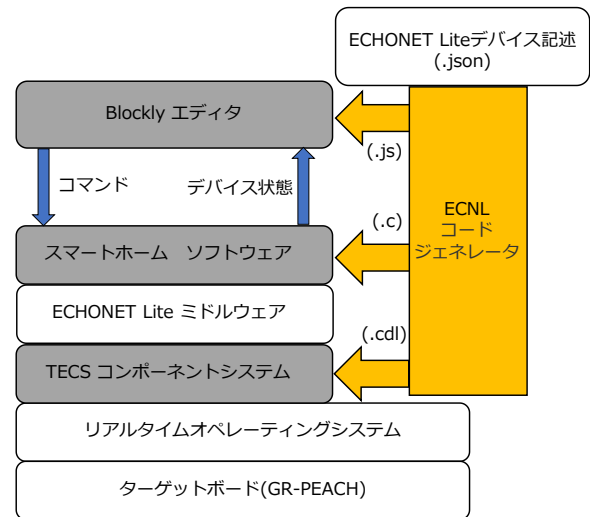


図 1 : システムモデル。

ウェアをコンポーネントに分割できるようにすることでソフトウェアの再利用性を向上させる[1]。TECS はコンポーネントを静的に展開および結合し、コンポーネントを結合しても大きなオーバーヘッドは発生せず、メモリ要件が削減される[2]。TECS ジェネレータによりカーネルオブジェクトの静的インタフェースコードを自動的に生成することもできる。したがって、開発者がスマートホームシステム用のアプリケーションを作成するのに効果的な方法である。

### 2.2. ECHONET Lite

ECHONET Lite は、スマートホーム向けに設計されたプロトコルであり、ISO 規格、IEC 規格及び HEMS (Home Energy Management System) の標準プロトコルとしても認証されている。通信仕様や各機器の制御コマンドを共通化することで、スマートホームシステム構築を実現する[3]。ネットワークに接続される機器の種類が増え、ホームネットワークによるサービスも多様化している。

### 2.3. Blockly

ブロックベースのプログラミングは、初心者向けのプログラミング方法であり、ユーザはパズルを通じて一連の論理コマンドを完了するこ

ECHONET Lite Framework based on Embedded Component Systems with Visual Programming

<sup>†</sup> Feng Qi, Takuya Azumi, Saitama University

<sup>‡</sup> Hiroshi Oyama, OKUMA Corporation

<sup>\*</sup> Hiroaki Nagashima, Cores Co., Ltd

とができる。Blockly はブロックベースのビジュアルプログラミングツールであり、ビジュアルブロックを使用して、変数、論理式、ループなどの要素を表す。テキストベースの言語でコードを生成するためのブロックエディタのユーザインタフェースとフレームワークを提供する。

### 3. 提案フレームワーク

ECNL コードジェネレータは JSON で記述した ECHONET 機器オブジェクト詳細規定を抽出し、デバイスに関する必要な情報をコンポーネント記述言語、ソフトウェアドライバとユーザインタフェースコードとして生成する。本章では生成する部分とユースケースについて説明する。

#### 3.1. ECNL コードジェネレータ

コンポーネント記述言語：デバイスのプロパティをコンポーネントのインタフェースとして定義し、コンポーネントの規定とコンポーネント間の接続も定義する。機器をコンポーネント化し、コンポーネント記述言語は TECS ジェネレータでテンプレートファイルを生成できる。開発者は様々な機器の仕様に従って、プロパティを制御する方法をテンプレートファイルに記述する。

ソフトウェアドライバ：デバイスドライバはコールバック関数として ECHONET Lite ミドルウェアによって呼び出される。この関数は ECHONET Lite 仕様で定義された番号に基づいてコンポーネントのインタフェースを呼び出す。そのため、ECHONET Lite 通信と機器の制御する機能が分離できる。

ユーザインタフェースコード：ユーザインタフェースコードは Blockly のビジュアルブロックを定義するために使用される。JSON 情報を抽出することによって生成された Javascript スクリプトはビジュアルブロックの形状、デバイスの名前、プロパティおよび制御オプションを定義する。ブロックをリストの形でブラウザで表示。

Blockly はブロックを作成するための Blockly Developer Tools[4]を提供するが、自動生成されたインタフェースコードは手動の作成プロセスを節約し、開発の効率を向上させる。

#### 3.2. 提案フレームワークを用いた実装例

提案フレームワークではビジュアルプログラミングの環境で組込みコンポーネントシステムを実行したデバイスの制御を実現できる。

図2上の図は ECHONET Lite 仕様に基づく自動生成したブロックで組み合わせて作ったサンプルプログラムである。サンプルプログラムでは、

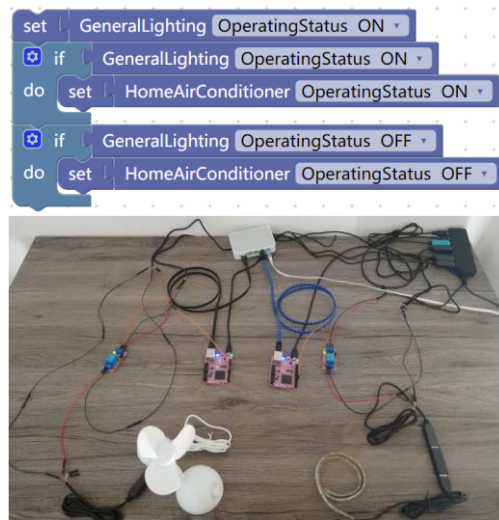


図2：提案フレームワークを用いた実装例。

スマートホームシステムで複数のデバイスの管理とデバイス間の相互制御を行う。下の図は提案フレームワークを用いた実装例である。組込みコンポーネントシステムは GR-PEACH と呼ばれるマイコンボードで実行され、GPIO はリレーを接続することでライトと扇風機を制御する。マイコンボードと Blockly クライアント間の通信はルータとの接続を介して実現され、サンプルプログラムによりライトと扇風機の制御が実現できる。

### 4. まとめ

本研究は TECS で組込みコンポーネントシステムを構築し、スマートホームソフトウェアの通信部とデバイス制御部を分離した。デバイスをコンポーネント化することにより、ソフトウェアの再利用性を向上させる。また、本研究で開発した ECNL コードジェネレータは ECHONET 機器規定の情報を抽出し、コンポーネント記述言語、ソフトウェアドライバとユーザインタフェースコードを自動で生成することにより、開発の効率を向上させた。

### 参考文献

- [1] Azumi, T., Yamamoto, M., Kominami, Y., Takagi, N., Oyama, H. and Takada, H.: A New Specification of Software Components for Embedded Systems, in Proc. of ISORC, IEEE, pp. 46–50 (2007).
- [2] Morisaki, S., Shirata, S., Oyama, H. and Azumi, T.: Unit Testing Framework for Embedded Component Systems, in Proc. of EUC, IEEE, pp. 41–48 (2020).
- [3] Jiang, J., Oyama, H., Nagashima, H. and Azumi, T.: Case Study: ECHONET Lite Applications based on Embedded Component Systems, in Proc. of APRIS, pp. 25–32 (2020).
- [4] Bak, N., Chang, B. and Choi, K.: Smart Block: A visual programming environment for SmartThings, in Proc. of COMPSAC, IEEE, pp. 32–37 (2018).