

DIY ユーザのためのスマートハウス構築支援システム

理学専攻 情報科学コース 矢田 和沙 (指導教員: 椎尾 一郎)

1 はじめに

日々の生活を快適・便利にするためのコンピュータ、センサー、ネットワークを導入した住宅、スマートハウスが注目を集めている。スマートハウスを実現するために、通信規格、OS、フレームワークなどが多数提案・開発・販売されている。Microsoft、Apple、GoogleなどのOS開発会社もスマートハウスに対応したOS拡張機能を開発している。その1つに、Apple社が提唱しているHomeKit¹がある。HomeKitはスマートハウスをApple社のOSであるiOSから操作するためのフレームワークであり、これに対応し動作する製品が多く販売されている。一方で、3Dプリンタや開発が容易な組み込みコンピュータなどの普及により、DIYにより電子工作を行いスマートハウスを実現しようとする人たちも現れている。しかしながら、組み込みコンピュータ上のプログラムをHomeKitから利用する過程は複雑で困難である。

本研究では、最新のOSライブラリやネットワーク技術に疎い電子工作者を対象とし、彼らが自作したデバイスとHomeKitの接続設定を支援するシステムを提案・実装した。

2 関連研究

スマートハウス実現のために、通信プロトコル、OS、フレームワークなどが多数提案実装され、販売されている。例えばスマートハウスの家電を操作する通信プロトコルとして、日本からはEchonet lite²、北米からSEP2.0³、欧州からはKNX⁴が国際規格として提案されている。また、機器の制御を行うアプリケーションを支援するためにデバイスの抽象化が必要であるとして、HomeOS [?]の提案もされている。ただしこれらのプロトコルやOSを利用してDIY機器を利用するためには、プロトコルやアプリケーションの実装が必要である。一方で既存のOSに、スマートハウス向けの機能を拡張し、スマートハウス機器を音声操作できる製品が提案されている。Google Home⁵は音声アシスタントアプリケーションを搭載した小型デバイスであり、対応するスマートハウス機器を操作できる。また、HomeKitはスマートハウス機器をコントロールするためのiOSのフレームワークであり、音声アシスタントSiriにより機器の音声コントロール機能を提供する。

HomeKitなどから操作するデバイスを開発するために必要な情報は、一般には公開されていない。しかし有志の手でデバイスに対応させるツールが開発され、オープンソースとして配布されている。例えば、Homebridge⁶はサーバー機能を提供するNode.jsライブラリでHomeKitにアクセスするための機能を兼ね備えており、様々なOSの上で動作させることがで

きる。操作したいデバイスに合わせたHomebridgeのプラグインを随時インストールし設定を行うことで、iOSからHomebridgeを通してDIYで作成した機器が使用可能になる。しかし、Homebridgeを導入し、適切なプラグインを選択し、インストール・設定する作業は最新のサーバーサイドプログラミングやネットワーク技術に疎い電子工作者にとって難易度が高い。これを容易にする目的で、Homebridge設定をGUIで行うHomebridge-server⁷が開発されているが、プラグインは既存のものをインストールするため、Siriによる音声対話操作に必要なサービス指定を柔軟に行うことができない。

3 HomeKit対応機器DIY支援ツール

本システムは、Linux環境を構築できる組み込みコンピュータであるRaspberry Pi⁸を使ってスマートハウスのための電子工作とプログラミングを行う初級・中級の人々(以下、DIYユーザ)を対象とする。上述の通り、HomeKit/Homebridgeに対応した機器を自作するための、Linux環境で動作するオープンソースツールが開発・公開されているものの、最新のサーバやネットワーク技術に疎いDIYユーザにとってはこれらの導入は難易度が高い。そこで本研究では、Siri音声対話制御が可能なHomeKit対応デバイスを、DIYユーザが容易に作成するための包括的なツールの作成を目指した。これらのツール群を以下で説明する。

3.1 環境構築の自動化ツール

Homebridgeが動作する環境を構築するには、Raspberry Pi上のLinux環境に、Node.jsや他のパッケージをインストールし、Homebridgeを動かすための様々な設定を行う必要がある。またそのインストールや設定にはシェルコマンドを使用する必要があり、一部の初心者には困難である。そこで、Homebridgeの環境構築を行うための手順をシェルスクリプトにまとめ、コマンドを1つ実行するだけでRaspberry Pi上に環境を構築できるようにした。

3.2 マルチプラグイン

スマートハウスで使用する様々なデバイス(以下、アクセサリ)に対応するHomebridgeプラグイン(以下、プラグイン)が、多くの開発者により開発され公開されている。DIYユーザは、この中から自分の開発したアクセサリに対応するプラグインを、選択しインストールを行い、さらに設定ファイルに各プラグインの指定した設定を書き込むことで、HomeKit対応アクセサリを作ることができる(図1(a))。しかし、開発したアクセサリに最適なプラグインを、公開されている多数の中から選び出すことは時間がかかり、専用プラグインを自作することはさらに困難である。

HomeKitでは、スマートハウスが提供するサービスに対してそれぞれアクセサリが割り当てられており、それぞれのサービスには複数の属性(Characteristics)が用意されている。例えば、Switchのサービスには

¹<http://www.apple.com/jp/ios/home/>

²<https://echonet.jp/>

³<http://www.zigbee.org/>

⁴<https://www.knx.org/>

⁵<https://madeby.google.com/home/>

⁶<https://github.com/nfarina/homebridge>

⁷<https://www.npmjs.com/package/homebridge-server>

⁸<https://www.raspberrypi.org>

「ON」という属性があり、スマートフォンアプリの操作や Siri の音声対話でこの属性に記載された動作を実行する。また Siri の音声対話では、それぞれのサービスの1つの属性に複数の音声コマンドが割り当てられている。

Homebridge のプラグイン開発者は、各自が開発したデバイスに特化したプラグインを公開する傾向にあり、DIY ユーザが開発した多様なデバイスに柔軟に対応できるプラグインが存在していなかった。そこで本研究では、Homebridge が提供している全サービス进行操作できる汎用性の高いプラグイン（以下、マルチプラグイン）を開発した。DIY ユーザは設定ファイル中で、アクセサリが提供するサービスを選択し、そのサービスの各属性に対応する実行コマンドを指定する。HomeKit を Siri から利用する場合には、1つの属性に複数の音声コマンドが対応するので、属性の指定によりアクセサリを制御する音声コマンドを指定することになる。これにより様々な種類の DIY アクセサリに対して、操作するプログラムを容易に割り当てることが可能である。

3.3 環境設定 Web ページ

Homebridge は多数のサービスを提供している。その中から、DIY ユーザが作成したアクセサリに適したサービスと、操作に対応する属性を選ぶのは難易度が高い。サービスや属性の詳細を知るためには、Homebridge のソースコードを読み解く必要があった。これを容易にする目的で、Raspberry Pi のローカルネットワークで動作する Web ページ上でマルチプラグインの設定を行えるようにした(図 1(b))。

設定ページでは、Homebridge に認識させる各アクセサリのサービスをセレクトボックスから選択する。するとそのサービスが対応している操作属性が自動的に表示される。次に DIY ユーザは、あらかじめ作成したプログラムやデバイスを動作させるコマンドを操作属性のフォームに入力する。このことでユーザは様々なサービスから自分が操作したい属性に対応するプログラムやコマンドを設定することができる。これによりソースコードを読み解くことなく、開発したアクセサリを Homebridge へ繋ぐことが可能になる。

ここで表示する属性やサービスは、Homebridge のソースコードより読み込み表示させている。しかし、対象の属性にどの Siri の音声コマンドが対応しているかに関しては、ソースコードにも書かれておらず、プラグインの属性の記載から判断しなければならない。そこで Homebridge のソースコードからサービスと属性を読み出し、有志の人々で編集ができる Wiki を作成した(図 2)。Wiki が充実すれば、サービスと属性に対する音声コマンド情報が整備されていくであろう。

4 ワークショップ

本システムの有用性を確認するためにワークショップを開催し、想定するユーザに利用してもらった。ワークショップでは、OS をインストールした Raspberry Pi を用意し、参加者にはワークショップのための手順書⁹に従い、Homebridge の環境構築と LED を点灯させる設定を行ってもらった。ワークショップに参加者は情報科学を専攻する学生 4 人である。いずれもコンピュー

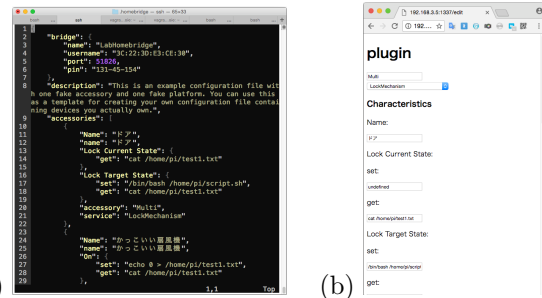


図 1: (a) テキストエディタでの設定ファイルの編集 (b) GUI でのプラグイン設定画面

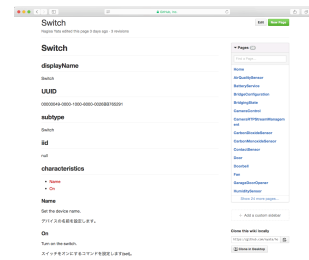


図 2: Wiki

タプログラミングの経験があるものの、HomeKit や Homebridge を扱った経験はない。その後、有用性を確認するためにアンケートを実施した。

Homebridge の設定にはコマンドコンソールを使用し、スクリプトを実行し環境構築を行った。アンケートによると、今まで経験した環境構築に比べて容易であったとの回答を得た。

次に、Raspberry Pi にあらかじめ配線した LED 回路と、これを点灯させるコマンドラインプログラムを参加者に提供し動作確認をしてもらった。参加者は、本システムのプラグイン設定ページへアクセスしこのプログラムを Homebridge のサービスとして設定した。この結果、参加者 4 名のうち最初に作業を実施した 1 名は HomeKit から LED を点灯するよう設定できなかった。この原因は作業手順書に不備があったためであり、手順書を修正した結果、残りの 3 名は設定することができ、アンケートでも非常に容易であったとの回答を得た。

5 まとめ

DIY デバイスによりスマートハウスを構築しようとする DIY ユーザのために、自作デバイスと HomeKit の接続設定を支援するシステムを提案し実装した。このために、汎用性の高い Homebridge マルチプラグインを開発し、これを GUI ベースで設定する Web ページを提供した。またワークショップを企画し、本システムの有用性を確認した。

現在の手法では、シェルコマンドで実行し、値を取得・操作できるアクセサリしか動作させることができない。今後シェルコマンド以外に、HTTP リクエストによる操作へ拡張することにより、幅広いアクセサリに対しても本手法を容易に適用できると考えている。

⁹<http://qiita.com/nyata/items/4c9fa9b9dbd6a88924b1>