

## クラウドを活用した家電リモコンサービスの検討

李 俊昊<sup>†</sup> 鷹取 敏志<sup>†</sup> 佐伯 幸郎<sup>†</sup> まつ本真佑<sup>†</sup> 中村 匡秀<sup>†</sup>

<sup>†</sup> 神戸大学 〒657-8501 神戸市灘区六甲台町 1-1

E-mail: †{junho,takatori}@ws.cs.kobe-u.ac.jp, ††sachio@carp.kobe-u.ac.jp,

†††{shinsuke,masa-n}@cs.kobe-u.ac.jp

あらまし 学習リモコンシステムは宅内の様々な家電機器のリモコンを統合できる便利なツールであり、最新のホームネットワークシステムでも利用されている。しかしながら、機器の信号は各家庭の住人自らが登録・管理しなければならず、機器や操作数が増えた場合に住人の負担が大きくなる。そこで本論文では、クラウドを活用した家電リモコンサービス (Remocon as a Service, RaaS) を提案する。RaaS のキーアイデアは、機器の信号をクラウド上で管理して複数のユーザで共有し、様々なアプリケーションからサービスとして利用することである。クラウドを活用することで、RaaS は従来の課題であった信号登録の負担や、制御機能と操作インターフェースとの密結合の課題を解決できる。本稿では、赤外線通信を対象を絞り、RaaS のアーキテクチャ、主要な DB の設計、操作インターフェースの考察を行う。また、Raspberry Pi と LIRC を用いた赤外線信号モジュールの試作を行い、RaaS の実現可能性を確認する。  
キーワード 学習リモコンシステム、クラウドコンピューティング、Remocon as a Service、RaspberryPi、LIRC

## Developing Cloud Service for Controlling Home Appliances

Junho LEE<sup>†</sup>, Satoshi TAKATORI<sup>†</sup>, Sachio SAIKI<sup>†</sup>, Shinsuke MATSUMOTO<sup>†</sup>, and Masahide NAKAMURA<sup>†</sup>

<sup>†</sup> Kobe University Rokko-dai-cho 1-1, Nada-ku, Kobe, Hyogo, 657-8501 Japan

E-mail: †{junho,takatori}@ws.cs.kobe-u.ac.jp, ††sachio@carp.kobe-u.ac.jp,

†††{shinsuke,masa-n}@cs.kobe-u.ac.jp

**Abstract** The programmable remote control system is a convenient tool that can consolidate multiple controllers of various house-hold appliances. It is also adopted in the emerging home network systems. However, the control signals of appliances are registered and maintained within individual home, which imposes a heavy burden on a home user. To cope with the problem, this paper presents *Remocon as a Service (RaaS)*, an appliance remote controller service using the cloud computing. The key idea is to manage and share all the control signals within the cloud, and provide the signals as service for various users and applications. By using the cloud, RaaS overcomes the problem of the signal registration and management. It also decouples the feature of remote control from user interface, which achieves a variety of remote control applications. In this paper, we focus on the conventional infrared remote control (IrDA), and discuss the architecture, primary databases, and user interface of RaaS. We also implement a prototype of a signal transmission module of RaaS using Raspberry Pi and LIRC, in order to see practical feasibility.

**Key words** Learning Remote Control, Cloud Computing, Remocon as a Service, RaspberryPi, LIRC

### 1. はじめに

現在、一般消費者の家庭には、TV やエアコン、DVD プレーヤ等の様々な家電・情報機器が導入され、住人の便利な暮らしを支えている。一方で、こうした家電機器の増加によって、住人は多種多様なリモコンを管理・操作しなければならず、混乱を招く一因となっている。典型的な問題として、あまり利用し

ないリモコンの場所がわからなくなる、頻繁に使う機能を沢山のボタンの中から探し出すのが大変といった問題がある。

こうした問題を解決するための 1 つの手段として、学習リモコン [1] が存在する。様々なメーカーのリモコン信号を登録・学習させることで、ユーザは一つのリモコンで複数の機器を操作できるようになる。一般的な学習リモコンは機器制御用の赤外線信号を記録・発信できる独立したデバイスであるが、近年 PC が

ら制御できるもの [2] やネットワークに接続可能なもの [3] [4] [5] も登場しており、ホームネットワークシステム (HNS) やスマートホームにおける機器制御にも利用されてきている。

その一方で、従来の学習リモコンには次のような課題がある。まず、利用するためにはユーザは自分で信号をリモコンシステムに登録し、動作確認を行う必要がある。機器や操作の数の増加に伴い、この登録作業は非常に煩雑になる。また、ユーザインターフェースがリモコンシステムのものに固定されてしまい、自分が望むリモコンを選択できない不便さが挙げられる。さらに、同じ型の機器が複数の家庭にある場合、同じ信号をそれぞれの家庭で分散して登録・管理するため非効率である。

そこで本研究では、クラウドを活用した家電リモコンサービス (Remocon as a Service, RaaS) を提案する。RaaS のキーアイデアは、あらゆる機器の制御信号をクラウド上で管理して複数のユーザで共有し、様々なアプリケーションからサービスとして利用することである。各家庭には信号を送信するための最小限の機能を持った軽量のネットワークデバイス (信号送信モジュール) だけが設置され、信号の登録やリモコン操作はすべてクラウド上のサービスから実行される。

はじめに家電機器メーカーは、RaaS の DB に自社が販売する全ての家電機器の信号を登録する。次にユーザは、クラウドに登録されている機器の中から、自分が購入・設置したものを選択し、自分用のリモコンを構成する。ユーザがリモコンを操作すると、RaaS は適切な信号を DB から検索してネットワーク経由で信号送信モジュールに指示する。最後に信号送信モジュールは、RaaS に指示された信号を送出し機器制御が行われる。

提案する RaaS では、正確な機器の信号が機器メーカーによって登録されるため、ユーザは信号登録の手間から開放される。また、リモコン操作や信号の利用がクラウドサービス (API) として利用できるため、個々のユーザのニーズに合致したりリモコン・アプリケーションが構築可能となる。さらに、機器の信号の登録・管理がクラウド上で一元管理されるため、同じ機器を持つユーザ間で信号が共有され効率的である。また、最新機器への迅速な対応も容易である。

本論文では、現在のリモコンの主流である赤外線通信を対象を絞り、RaaS の主要な DB の設計を行った。具体的には、機器製品ごとに固有の制御信号を管理する信号 DB と、家ごとに異なる家電機器の構成を管理する構成 DB の設計を行った。また、RaspberryPi [6] と LIRC [7] を用いた信号送信モジュールを試作し、RaaS の実現可能性を確認した。

## 2. 準備

### 2.1 学習リモコンシステム

学習リモコンシステムは、家電機器のリモコン信号を内部の記憶領域に登録 (学習) させることができるシステムである。ユーザは学習リモコンの各ボタンに、自分の好きな機器の信号を登録できる。ユーザが学習リモコンのボタンを押すことで、登録された信号が送信され、機器を操作できる。家電機器毎に存在する多種多様なリモコンの信号を 1 台の学習リモコンに集約できるため、ユーザは煩雑なりモコン管理から開放される。

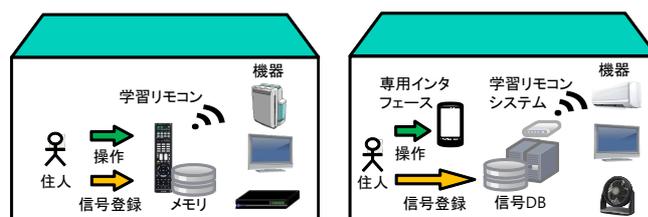


図 1 学習リモコンシステムによる機器制御

図 1 に学習リモコンシステムによる機器制御の概略を示す。左の図は一般的な赤外線通信 (IrDA) 用の学習リモコンを表している。図中の信号登録は次のように行われる。ユーザである住人はまず学習リモコンを「学習モード」に設定し信号を割り当てたいボタンを選択する。次にユーザは、家電リモコンを学習リモコンにつき合わせ、家電リモコンの登録したい信号のボタンを押し、学習リモコンに信号を送信する。学習リモコンは、送られた信号を内蔵メモリに保存し、ボタンに割り当てる。最後にユーザは学習リモコンを「操作モード」に切り替えてボタンを押し、信号登録が正常に行われたかをチェックする。

図 1 右は、家庭内のネットワークに接続可能なより最近の学習リモコンシステムを表している。機器の信号はサーバ機能を持つ情報機器で管理され、住人はスマートフォンやタブレット、音声等の先進的なインターフェースで操作できるようになっている。しかしながら、機器の信号を各家庭で個別に登録・管理するという点では、左図の学習リモコンと大きな差はない。

学習リモコンは多数のリモコンを 1 つに集約できるメリットはあるものの、信号登録の作業が煩雑というデメリットがある。図 1 に示すように、操作したい家電機器は家庭ごとに異なるため、学習リモコンへの登録作業は住人自身で (あるいは専門業者に依頼して) 行う必要がある。そのため、機器や操作の数が増えるに従い、住人の負担が大きくなる。また、信号の登録ミスや学習リモコン故障による信号消失のリスクも大きい。代表的な機器の信号をプリセットしたりリモコンも存在するが、随時登場する最新機器の信号には対応することができない。

### 2.2 クラウドコンピューティング

クラウドコンピューティングとは、サーバやソフトウェア、データ等の様々な計算資源がネットワークで管理され、ユーザは必要に応じてそれらをサービスとして利用する計算パラダイムである。情報処理に必要な計算資源はユーザの手元にはなく、複数ユーザでネットワーク越しに共用するクラウドサービスとして提供される。そのため、ユーザの手元にはサービスにアクセスするための軽量の端末 (シン・クライアント) があればよい。一般的なクラウドサービスでは、API が Web サービスで公開されており、Web の標準技術を用いて様々なプログラムからサービスを利用出来る。

本研究では、クラウドコンピューティングが持つ様々な特徴のうち、特に次のものに注目している。

オフローディング: ユーザの手元の煩雑な処理をクラウドに任せるといった性質。

プログラマビリティ: 人手を介さずプログラムから API で直

接サービスを利用できる性質。

マルチテナンシ：1つのサービスや計算資源を複数人で共有できる性質。

### 2.3 本研究で解決する課題

本研究は、クラウドを活用して、従来の学習リモコンシステムにおける以下の3つの課題を解決することを目的とする。

#### 課題 P1: 住人による信号登録作業の問題

2.1で述べたとおり、宅内で使用する機器の数や機能の数が増えると、学習リモコンシステムへの信号登録の作業が負担になる。また、信号の登録ミスや、リモコンの故障による信号消失のリスクがある。提案手法ではこうした住人の手間やリスクを削減し、より簡単で便利なりモコンシステムを実現したい。

#### 課題 P2: 機能とインターフェースの密結合の問題

従来の学習リモコンシステムでは、システムの機能とユーザーインターフェースが密結合しており、ユーザはシステムが提供する専用リモコンで操作する必要があった。このことは必ずしも全てのユーザにとって使いやすいシステムにはならない。信号を学習・送信するというシステムの機能をユーザーインターフェースから分離し、住人一人ひとりに個人適応したユーザーインターフェースを実現したい。また、人手を介さずに、様々なアプリケーションやプログラムから直接機器を制御するホームネットワークシステム [8] にも利用したい。

#### 課題 P3: 機器信号の管理効率の問題

図1に示すとおり、従来のシステムでは各家庭で住人が信号を管理する。したがって、同じ機種の機器が複数の家庭にある場合でも、同じ信号をそれぞれの家で重複して登録・管理する必要があり、非効率である。また、新たな家電機器が開発された場合にも、信号の登録は購入した住人によって手動で行われるため、全家庭のリモコンシステムに新たな信号が行きわたるまでには時間がかかる。また信号の変更が必要な場合にも、各家庭での更新が手間である。

## 3. Remocon as a Service

### 3.1 キーアイデア

前節の課題 P1, P2, P3 を解決するために、本研究ではクラウドコンピューティングを活用した新しいリモコンサービス Remocon as a Service (RaaS) を提案する。RaaS のキーアイデアは、2.2 で述べた3つの特徴を生かして、リモコンサービスの課題を次のように解決することである。

- 各家庭で住人が管理していた煩雑なりモコン信号管理を RaaS へオフローディングすることで、課題 P1 を解決する。
- RaaS のサービス API を利用して、様々なリモコンのユーザーインターフェースをプログラム可能にすることで、課題 P2 を解決する。
- 全ての機器の信号を RaaS 内でマルチテナンシで管理し、複数人で共有することで課題 P3 を解決する。

### 3.2 アーキテクチャ

図2に提案する RaaS のアーキテクチャを示す。クラウド上には、全ての機器の信号を管理する信号 DB と、各家庭の家電機器の構成情報を管理する構成 DB が置かれる。これらのデー

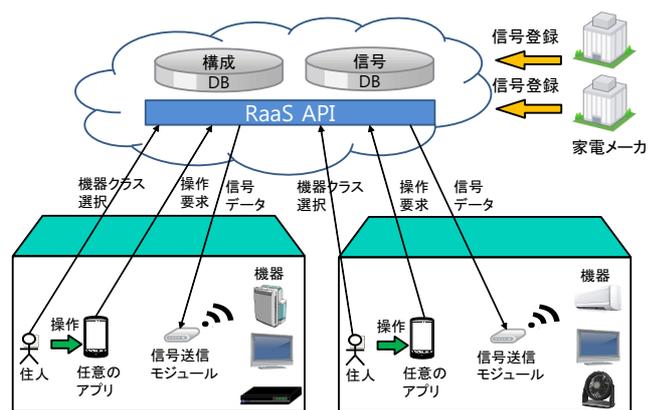


図2 Remocon as a Service のアーキテクチャ

タへのアクセスはサービス API (RaaS API) を介して行われる。各家庭には、宅内の機器に信号を送信するための最小限のネットワークデバイス (信号送信モジュール) が置かれ、RaaS と接続される。

提案アーキテクチャでは、各家電メーカーは自社の全ての製品の機器信号を信号 DB に登録する。家電機器の信号は機種ごとに共通であるから、メーカーが正確な信号を一度登録しておくことで、同じ機種の機器を所有する複数のユーザで共有することができる。したがって RaaS では、住人自らが信号登録および動作確認を行う必要がなくなる。

次にユーザである住人は、RaaS に登録されている機種 (機器クラス) から、自分の宅内に設置されている家電機器の機種を選択する。こうして得られた宅内機器と機種のひも付けは、RaaS 内の構成 DB に保存され、宅内の機器それぞれの信号を信号 DB から検索できるようになる。

住人が操作するインターフェースは、RaaS API を利用した任意のアプリとして実現される。住人は自分が使いやすいリモコンアプリをダウンロード、または、自分で作成し、利用する。住人がインターフェースを操作すると、RaaS API に機器操作要求が発行され、構成 DB から機種を検索し、信号 DB から対応する機種の信号が検索される。最後に信号データが信号送信モジュールに送られ、機器が制御される。

本研究では、機器の制御信号を従来の赤外線通信 (IrDA) に対象を限定し、信号送信モジュール、RaaS の2種類の DB、および、ユーザーインターフェースについて更なる考察を行う。

### 3.3 信号送信モジュール

提案手法における信号送信モジュールは、宅内機器へ信号を送信するための最小限の機能を持ったネットワークデバイスである。このデバイスは、クラウドから送信すべき信号データを受け取ると、その信号をそのまま信号を宅内の物理空間へ伝播する。赤外線通信の場合には、クラウドから信号のビットストリームが送られてきて、それを赤外線 LED から照射するという実装になる。各信号送信モジュールには、ネットワークアドレス (例えば IP アドレス) が付与されているものとする。クラウドからは、このアドレスを指定して目的の家庭のモジュールへ信号を送信する。



図 3 家電リモコンサービスのデータベース設計

### 3.4 RaaS データベース

RaaS を実現する主要なデータベースとして、ここでは信号 DB と構成 DB を説明する。図 3 に文献 [9] の表記法に従った ER 図を示す。四角はデータエンティティ(テーブル)を表しており、横に主キー(下線)、属性から構成されるデータ項目(スキーマ)で定義されている。エンティティの下にはインスタンスを併記している。エンティティ間の関連として ( + ... ) は参照関係を表している。

信号 DB は、機器の操作一つひとつに割り当てられた信号を記録する。通常、家電機器の場合は同一の機種であれば、共通の信号で操作可能である。例えば、ある機種のテレビ (VIERA-TH58PZ) の電源 (POWER) に割り当てられた赤外線信号は、0x00045c という値であるといった具合である。このテレビには他にもチャンネルや音量、ボリュームといった様々な信号が存在する。したがって、図 3 上に示すとおり、信号 DB は機器の機種(機器クラス)と操作名を複合キーとして持つスキーマで定義される。通常、機器の各操作の信号はその機器のメーカーによって決定される。したがって、信号 DB への信号の登録は、全て家電機器メーカーによって行われるものとする。

一方、宅内がどのような機種の家電機器で構成されているかは、家庭によって異なる。したがって、各家庭のそれぞれの機器がどの機種であるかは、ユーザ自身がひも付ける必要がある。この情報は、構成 DB によって管理され、RaaS のユーザである住人によって保守される。図 3 下に示すとおり、構成 DB はユーザとそれぞれの機器(のインスタンス)を複合キーとして持ち、その機器の機種(機器クラス)を属性として持つ。また、その機器がどの信号送信モジュールから操作されるかを表すモジュールアドレスも属性として持っている。例えば、ユーザ Junho の TV は VIERA-TH58PZ で、アドレス 133.xxx.yyy.zzz の信号送信モジュールから制御されることを表している。

いま、Junho の TV の電源を操作する信号を取得することを考える。まず構成 DB からその TV の機種が VIERA-TH58PZ であることがわかる。次に、信号 DB から VIERA-TH58PZ の POWER を探し、0x00045c であることがわかる。すなわち、2 つの DB から、Junho の TV の電源を操作するには、0x00045c の信号を、アドレス 133.xxx.yyy.zzz の信号送信モジュールに送れば良いことがわかる。

その他にも、家電機器の型番やメーカーを管理する機器クラス DB や、登録ユーザの情報を管理するユーザ DB 等が考えられるが、これらはリモコンサービスに特化したものではないため、本稿では割愛する。ホームネットワークやスマートシティ



図 4 操作アプリケーションのイメージ図：(左)機種選択画面、(右)操作画面

等の構成情報を利用することも考えられる。

### 3.5 ユーザインターフェース

ユーザが RaaS を利用するためのインターフェースは、RaaS API を利用したアプリケーションである。RaaS API は、クラウド上の Web サービスとして公開されるため、ブラウザを用いた Web アプリケーションや内部から Web サービスを呼び出すネイティブアプリケーションとして実装できる。スマートフォンやタブレットから操作できるようにすれば、リモコンとしての携帯性が向上する。

図 4 にスマートフォン上に実装されたりモコンアプリのイメージ図を示す。左の図は、ユーザが RaaS にアクセスして、機器の機種を選択する画面である。まず画面下部に機器の名称と信号送信モジュールを入力して、「追加」を押すと RaaS の構成 DB に登録されて、画面上部に現れる。TV, Light, 扇風機など必要な機器を選択する。

次に、プルダウンから登録した機器の機種を、プルダウンから選択する。ユーザの登録作業はこれだけである。従来のように、各ボタンの信号を手動で学習リモコンに登録する必要はない。

図 4 右は、機器を操作する画面である。この画面では、TV のリモコンを表している。ボタンを操作すると、アプリケーションは RaaS に対応する機器操作を要求する。RaaS は、要求された機器操作の信号を検索し、信号送信モジュールに送信を指示する。信号送信モジュールは信号を送信し、機器が制御される。

こうしたリモコンアプリは、家電機器メーカーやサードパーティ開発者によって開発され、住人は自分の好みのものを選んで利用すると仮定している。また、RaaS API は要求された機器操作の信号を検索・送信する汎用的なサービスであるから、その用途はリモコンアプリのみに閉じるものではない。複数の機器を同時に制御したり、様々な情報やコンテキストと機器制御とを連動するようなより高度なアプリケーションも実現可能

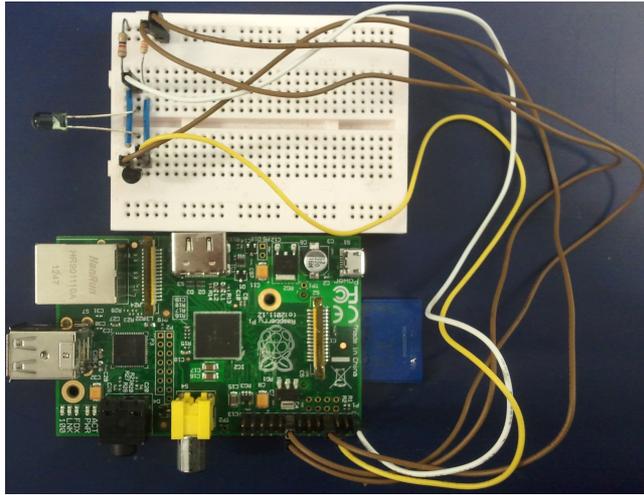


図 5 試作した信号送信モジュール

である。

#### 4. 信号送信モジュールの試作

提案手法の実現可能性を確認するため、本節では赤外線信号を対象とした信号送信モジュールの試作を行う。

##### 4.1 ハードウェアの選定

RaaSにおける信号送信モジュールは、ネットワークに接続可能な軽量デバイスであることから、その基盤ハードウェアとして Raspberry Pi [6] を選択した。Raspberry Pi は Linux ベースの安価なシングルボードコンピュータであり、GPIO (汎用入出力) ポートを介して様々な電子回路を接続することができる。

試作機では、まず赤外線を送信するための回路 (赤外線送信回路) を作成し、Raspberry Pi の GPIO に接続する。次に、Linux の赤外線信号アプリケーションである LIRC [7] を Raspberry Pi にインストールして、赤外線の送信を行う。

図 5 に試作した赤外線送信モジュールの概観を示す。図の下側にあるのが Raspberry Pi であり、上側にある赤外線送信回路と GPIO を介してケーブルで接続されている。

##### 4.2 赤外線送信回路の構成

赤外線信号の送信は、赤外線 LED (IRED) を信号のパターンにあわせて点滅することで行われる。図 6 に赤外線送信回路の回路図を示す。左は一般的な LED を駆動する回路であり、右は試作した IRED の駆動回路である。両者の違いは、IRED は LED に比べより低電圧で駆動できるという点である。低電圧で駆動するには大きな抵抗を使用すればよいが、IRED の出力が弱くなる。一方で、抵抗を小さくすれば出力は強くなるが、IRED と Raspberry Pi に負荷がかかるため、適当な抵抗値を設定する必要がある。

試作機で使用した IRED の規格は、定格 1.35V、最大 1.6V、ピーク周波数 940nm、画角  $\pm 10$  度である。一方、Raspberry Pi の通常出力は 3.3V で最大 5V まで使用可能である。

通常の家電機器用リモコンで利用される回路では、赤外線信号の到達距離をある程度 (5m 程度) 確保するために、数十から数百 mA の電流を流している。したがって、図 6 の左のよう

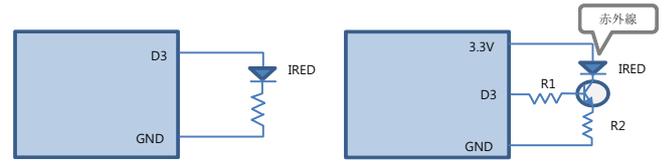


図 6 信号送信モジュール LED 部の回路図

に、トランジスタを利用して電流を増幅制御する。Raspberry Pi の GPIO からの低電流でトランジスタの on/off を制御し、実際 IRED の電流供給は比較的大きな電流を使用することができる。試作機では、R1 は 330  $\Omega$ 、R2 は 22  $\Omega$  の抵抗を使用した。

試作機をテストしたところ、研究室のあらゆる場所から方向に関係なく信号が届くことを確認した。5V の電圧は少々高かったため、Raspberry Pi のデフォルトである 3.3V を使用した。試作機では一時的に大きな電流が流れるが、赤外線通信は非常に短い間の駆動のため、電力消費はそれほど大きくない。

##### 4.3 LIRC を利用した赤外線送信

LIRC (Linux Infrared remote control) は、Linux ベースのコンピュータシステムを使用して、赤外線信号を送受信するためのソフトウェアである。本試作機では、LIRC を利用して Raspberry Pi の GPIO に赤外線信号のパターンを出力する。

LIRC は、図 7 に示すような信号定義ファイルを参照して、赤外線信号を管理する。操作 (コード) が要求されると、LIRC はこのファイルを参照し、対応する赤外線信号を指定されたポートに出力する。今回の試作機では、Panasonic 製の TV リモコンの信号定義ファイルを利用して、プラズマテレビ (VIERA TH58-PZ) に信号送信を行い、動作を確認した。

提案する RaaS では、この LIRC の信号定義ファイルに相当するものを、各家電機器メーカーが準備して、RaaS API を通じて信号 DB に登録することを想定している。信号送信モジュール自身はこうした信号定義を持たずに、クラウドから指示された信号を単に送信するだけの軽量な端末 (シン・ターミナル) として機能すべきである。今後の研究で完成させていきたい。

## 5. 考察

### 5.1 特長と限界

提案する RaaS の特長は、従来の学習リモコンシステムの 3 つの課題 (P1 ~ P3、2.3 参照) をクラウドを活用することで解決できることである。機器制御の信号はメーカーが登録したものをクラウドで一括管理し、複数のユーザで共用することで、住人は煩雑なりモコン信号登録の作業から開放される。また、RaaS API によってリモコンの機能とインターフェースが分離され、住人が利用しやすいリモコンアプリを選択して利用することができる。また、新たな機器が登場した際にも、機器メーカーは RaaS に信号を一度登録するだけで、全ての家庭で即座に新たな信号を利用できるようになる。そのため、新しい機器の迅速な流通を後押しすることができる。

一方で RaaS では、信号データや信号検索の処理プログラムが全てクラウド上にあるため、ネットワークが切断されると

```

name TV
bits 24
flags SPACE_ENC
eps 30
aeps 100

header 3437 1641
one 477 1197
zero 477 345
ptrail 475
pre_data_bits 24
pre_data 0x40040D
gap 72100
toggle_bit_mask 0x0

begin codes
POWER 0x00EBCB1
DRIVE_SELECT 0x003439
TV 0x00000D
1 0x000805
2 0x008885
3 0x004845
4 0x00C8C5
5 0x002825
6 0x00A8A5
7 0x006865
8 0x00E8E5
9 0x001815
DELETE 0x00222F
0 0x009895
INPUT_SELECT 0x00A07
CH+ 0x002C21
CH- 0x00ACA1

end codes
end remote

```

図 7 信号登録ファイルの例 (LIRC 形式)

サービスが利用できなくなるという問題がある。これは RaaS に限らず全てのクラウドサービスにも言えることである。ネットワークが切断された際にも、基本的な機器制御が出来るように、各家電機器は手で操作できるインターフェースを機器本体に備えるべきである。

## 5.2 関連研究

ネットワークに接続可能な赤外線リモコンとして、iRemocon [4] が有名である。その役割は我々の赤外線送信モジュールに類似しているが、制御信号を各 iRemocon の本体で登録・管理する点において異なっており、従来の学習リモコンシステムに分類される。

我々の研究グループでは、赤外線リモコンを Web サービスでラップしたホームネットワークシステム (HNS) を開発している [8]。しかしながら、赤外線リモコンの部分は従来の学習リモコンを利用しており、各家庭で信号を制御する方法を採ってきた。今後、提案する RaaS を適用していきたい。

また我々は最近の研究 [10] において、従来は宅内にあった HNS のホームサーバをクラウド上で管理するクラウド型 HNS の提案を行っている。クラウド型 HNS では、機器制御に限らず様々な住宅データ (ログ、構成情報) を、特定の用途や目的に依らない HNS の標準的なサービスとして、サービス事業者が提供する新たなアーキテクチャである。本論文で提案する RaaS は現状赤外線に限定した実装になっているが、今後クラウド型 HNS の枠組みに組み入れ、赤外線以外の通信方式にも柔軟に対応できるよう拡張していく予定である。

## 6. おわりに

本研究では、従来の学習リモコンシステムの課題を解決するために、クラウドを活用した家電機器リモコンサービス (Remocon as a Service, RaaS) の提案を行った。クラウドを活用

することで、住人は煩雑な信号登録から開放され、様々なユーザインターフェースを持つリモコンアプリを利用でき、メーカーが提供する正確な信号を複数のユーザで効率的に共有することが可能になる。また、赤外線信号を対象を限定し、RaaS の主要なデータベースの設計、ユーザインターフェースの考察、信号送信モジュールの試作を行った。その結果、RaaS の実現可能性を確認できた。今後の課題としては、赤外線信号以外への RaaS の拡張、RaaS API の設計、クラウド型 HNS との統合、ユーザビリティの向上等が挙げられる。

謝辞 この研究の一部は、科学技術研究費 (基盤研究 C 24500079, 基盤研究 C12877795, 基盤研究 B 23300009)、および、積水ハウスの研究助成を受けて行われている。

## 文 献

- [1] Wikipedia, “学習リモコン”. <http://ja.wikipedia.org/wiki/%E5%AD%A6%E7%BF%92%E3%83%AA%E3%83%A2%E3%82%B3%E3%83%B3>.
- [2] Buffalo, Inc., “パソコン用学習リモコン pc-op-rs1”. <http://buffalo.jp/products/catalog/item/p/pc-op-rs1/>.
- [3] パナソニック, “ライフィニティ・ワイヤレスネットリモコンシステム”. <http://www2.panasonic.biz/es/densetsu/lifinity/images/system.pdf>.
- [4] GLAMO Inc., “iRemocon - smartphone home electronics controller”. <http://i-remocon.com/>.
- [5] I. Universal Remote Control, “Whole house control: Simply brilliant”. <http://www.universalremote.com/>.
- [6] The Raspberry Pi Foundation, “RaspberryPi - an arm gnu/linux box for \$25. take a byte!”. <http://www.raspberrypi.org>.
- [7] C. Bartelmus, “LIRC: Linux infrared remote control”. <http://www.lirc.org/>.
- [8] M. Nakamura, A. Tanaka, H. Igaki, H. Tamada, and K. ichiMatsumoto, “Constructing home network systems and integrated services using legacy home appliances and web services,” International Journal of Web Services Research, vol.5, no.1, pp.82–98, Jan. 2008.
- [9] 渡辺幸三, 販売管理システムで学ぶモデリング講座, 翔泳社, May 2008.
- [10] 鷹取敏志, まつ本真佑, 佐伯幸郎, 中村匡秀, “マルチベンダサービスを実現するクラウド型ホームネットワークシステムの提案,” 電子情報通信学会技術報告, 第 113 巻, pp.53–58, Nov. 2013.