

Web ブラウザ上での容易な機器制御を可能にするエンドユーザ向けフレームワーク

春原 雅志[†] 岩崎 陽平[‡] 河口 信夫[‡]

[†]名古屋大学大学院情報科学研究科 [‡]名古屋大学大学院工学研究科

Simple Device Control Framework for End user on Web browser

Masashi Sunohara[†] Yohei Iwasaki[‡] Nobuo Kawaguchi[‡]

[†]Graduate School of Information Science, Nagoya University

[‡]Graduate School of Engineering, Nagoya University

1. はじめに

近年、ユビキタス環境を構築するための研究が進められる中で、身の回りの空間に存在する機器をネットワークに接続し、利用する場面が多くなりつつある。これまで我々が機器を利用する際には、機器を制御するための専用リモコンや機器本体のスイッチを直接操作する必要があった。これらの機器をネットワークに接続することにより、複数の機器の制御情報を集約することや、遠隔地などの離れた場所からの機器利用が可能になる。例えば、我々の研究室で提案している cogma[1]をベースに作られた WebCodget システム[2]では、様々な機器の制御を Web ブラウザ上で行うことを可能にしている。このシステムでは、予め機器を制御するための情報や Web ブラウザ上に表示するための HTML 情報を Codget と呼ばれる移動ソフトウェア内に記述し、ネットワークに接続する機器に組み込んでおく。機器がネットワークに接続されると、Codget はネットワーク上の Web サーバを cogma のサービス発見機構を用いて自動的に発見し、自身のコピーを Web サーバ上に登録する。これにより、ネットワーク上の機器の情報が Web サーバ上に集約される。機器を利用したいユーザは Web ブラウザを用いて Web サーバにアクセスすることにより複数の機器の利用が可能になる。最近では Web ブラウザはパソコンだけでなく、携帯端末上でも利用可能なため、Web ブラウザ上での機器利用を可能にすることで様々な端末からの利用が可能になる。しかし、これらのシステムを実現するためには、プログラミングの知識を持っている開発者が機器ごとにソフトウェアの開発を行い、機器に組み込んでおかなければならない。エンドユーザが、新しく機器をシステムに追加したいと思ってもそれを簡単に実現することは困難である。できるだけ簡単な手順で機器をネットワーク対応にし、Web ブラウザからの制御が実現できると便利である。

本稿では、ユビキタス環境やスマートルームに存在する様々な機器を、容易にネットワークコントロール可能にするためのフレームワーク構築を目的とする。ユーザは Web ブラウザを用いて、簡単な操作で機器の制御情報を Web サーバに登録する。制御情報とは、例えばシリアル通信を用いて制御する際の制御コマンドなどである。Web サーバに登録した制御情報は、Web サーバにアクセス可能な他のユーザと共有することが可能になる。これにより、高度なプログラミングの知識を持っていないユーザでも、身近な機器を手軽に Web ブラウザ上で制御可能になる。

本論文の構成は以下の通りである。まず第 2 節で我々の身の回りにある機器とその制御方法について述べ、第 3 節で本フレームワークの手法について述べる。第 4 節で本論文についてまとめ、今後の課題を述べる。

2. 機器の制御

2.1. ユビキタス環境における様々な機器

日常生活において、我々はオフィスや研究室などの様々な場所で多くの機器を利用している。ユビキタス環境ではこれらの機器をネットワークに接続し、機器の設置場所に依存せず利用したり、機器同士を連携させて利用したりといった場面が見られる。図 1 に示すスマートルームと呼ばれる空間には、環境情報を取得する各種センサや、情報を表示するためのプロジェクタ、プラズマディスプレイなど多くの機器が設置されている。スマートルームではこれらの機器をネットワークを介して利用可能であるが、このような環境を実現するためには、予め機器をネットワークに対応させるためのプログラムを作成し、事前に様々な設定を行っておく必要がある。これらの手順は新しい機器をスマートルーム内に導入する度に必要となり、プログラムを作成する知識を持たないエンドユーザにとっては容易なことではない。ユーザの負担をできるだけ軽くし、手軽に機器の追加が可能であることが求められる。



図 1 スマートルーム内の様々な機器

2.2. ネットワークを用いた機器制御

ネットワークを介して利用する機器には、相互に機能を提供しあうために UPnP[3]や DLNA[4]、Bonjour[5]といった技術を利用しているものが見られる。これらの技術を用いることにより、複雑な機器設定をしなくても簡単にネットワークに接続して利用することが可能となる。また、機器の中にはパソコンなどからコマンドを送り制御するためにシリアルポートを備えている機器がある。そのような機器はシリアルポートに直接コマンドを送ることにより機器を制御することが可能である。

2.3. シリアル通信を用いた機器制御

機器の多くは、機器を操作するための専用のリモコンや、機器自体に備え付けられているボタンを用いて操作する。既存の機器をスマートルーム内のネットワークに接続して利用するためには、機器に対して信号を送り制御する手段が必要となる。例えば、シリアル通信により制御可能な機器は機器に対してコマンドを送信することにより制御が可能となる。例えば、プロジェクタやマトリクススイッチャ等の機器にはシリアル通信コネクタを装備しているものが見られる。

シリアル通信を用いて機器を制御するには一般的に機器ごとに専用のクライアントソフトなどを作成する必要がある。シリアル通信を行う際には、機器ごとに定められた通信速度などのパラメータを設定して通信する必要がある。

Java 言語にはシリアルポートを制御するための API (Java Communication API) が用意されており、この API にパラメータを設定することによりシリアル通信を用いたソフトウェアを作成することが可能である。しかし、例えば第 1 節で述べた WebCodget システム [2] に機器を対応させる場合でも、Java のプログラミングの知識と WebCodget システムの仕組みを理解しなければならず、エンドユーザが作成することは容易ではない。

2.4. シリアル-Ether 変換機を用いた機器制御

通常、シリアル通信を行う際には、制御するパソコンに直接機器を接続する必要があるが、シリアルケーブルの長さや、シリアルポートのポート数の制約を受ける事になる。特に機器の数が増加した際にはその影響を大きく受けることになる。シリアル-Ether 変換機は、シリアル通信で用いられる信号を TCP/IP のパケットに変換する製品である。これを用いることにより、Ethernet 経由でのシリアル機器の制御が可能になる。シリアル-Ether 変換機を利用する際には制御を行うパソコンに仮想シリアルポートを作成し、この仮想ポートに対して信号を送ることにより Ethernet ワーク上に接続されたシリアル機器を制御することが可能になる。シリアル機器は Ethernet 上に設置可能であるので、機器の設置場所に柔軟に対応できる。

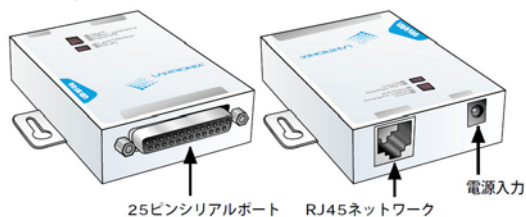


図 2 シリアル-Ether 変換機 (UDS-100¹)

2.5. Web サーバを用いた機器制御

シリアル通信の利用により、機器に特定のコマンドを送ることで制御可能である。また、シリアル-Ether 変換機を用いることにより、シリアルケーブルの長さにとらわれず、柔軟な設置が可能になる。しかし、シリアル-Ether 変換機を用いて、機器を制御するための専用のクライアントソフトを作成した場合、そのソフトを実行しているパソコンからしか機器を制御することができない。これは、複数のユーザで機器を共有したいような場合には不便である。ユビキタス環境にお

いては、普段利用しているパソコンや携帯端末から、ネットワーク上の機器を自由に利用することが求められる。

そこで、ここでは Web サーバを用いた機器制御について説明する。Web サーバを用いた機器利用は、WebCodget でも用いられている。まず、機器の情報を Web サーバ上に集約する。そしてユーザはパソコンや携帯端末の Web ブラウザから Web サーバにアクセスし、機器の情報を取得したり、GET や POST といったメソッドを Web サーバに制御情報を送ることにより、機器の制御を行う。ユーザは専用のクライアントソフト等を利用しなくてもブラウザ上での機器操作が可能になる。また、Web サーバにアクセスできるユーザであれば、部屋の中に設置した機器を複数のユーザが利用することが可能になる。さらに、機器の情報が Web サーバ上に集約されているので、複数の機器の統合的な管理や機器同士の連携を実現することができる [2]。

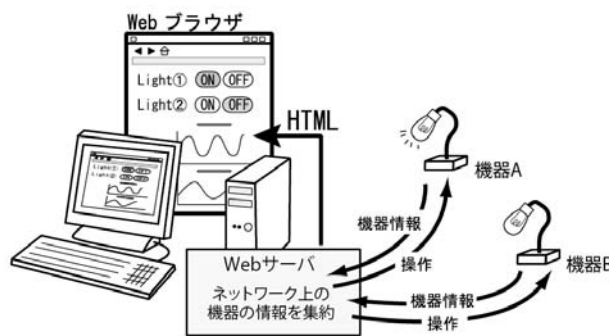


図 3 Web サーバを用いた機器制御

3. エンドユーザ向けフレームワーク

3.1. フレームワークの目標

第 2 節ではユビキタス環境にある機器と、その制御手法について述べた。2.4 節で述べたような Web ブラウザ上の機器制御を実現するためには、予め機器を制御するためのプログラムの作成や、ブラウザに表示するための HTML を記述するといった準備が必要となる。しかし、これらの準備は専門的な知識を持たないエンドユーザにとっては容易なことではない。そこで、本稿ではそのようなエンドユーザでも Web ブラウザ上からの簡単な操作で、機器の情報を Web サーバに登録し、ネットワークを介して操作可能な状態にすることを目指す。

3.2. エンドユーザによる機器操作の実現

本フレームワークでは、機器を利用するエンドユーザ自身が簡単な操作で機器の操作情報を Web サーバに登録し、ネットワークを介して機器の利用を行う。シリアル通信で制御する機器は、シリアルポートに対して機器ごとに定義されたコマンドを送信する必要があるが、その役割を Web サーバが行う。これを実現するために、Web サーバ上に機器を制御する情報を容易に登録するためのページを生成するためのスクリプトを用意する。エンドユーザは普段利用している Web ブラウザ上で、フレームワーク上で用意されているボタン等とコマンドを組み合わせて、機器の情報を登録する。そして Web ブラウザから Web サーバに対して機器操作のクエリを送ることにより、Web サーバを介して機器の利用が可能となる。

¹ (株) 日新システムズ <http://www.co-nss.co.jp/>

3.3. シリアル機器の通信方式

本稿ではシリアル通信を用いて制御可能な機器に焦点を当て、容易にネットワークに対応させるフレームワークを提案する。

シリアル通信を用いて機器を制御する際には機器との間で制御コマンドやデータの送受信が行われる。送受信の方法には以下に挙げるようなパターンが考えられる。

(1) 機器に対してコマンドの送信を行うとそれに対するレスポンスが発生するパターン

このパターンは機器を操作するためのコマンドを送ると、機器側でのコマンドの受信と、それに対する動作が正常に終了したかどうかをレスポンスとして返すものである。機器によっては複数のコマンドを送る際に、レスポンスの状態を毎回確認し、正常であれば次のコマンドを送信する必要がある。

(2) 機器に対するコマンドを送信してもレスポンスが発生しないパターン

このパターンは機器に対してコマンドを送信しても機器側からはデータが送信されないものである。機器に対してはコマンドを送信するだけになる。指定した動作が正常に行われたかどうかについては保証されない。

(3) 機器に対してコマンドの送信を行うことにより機器側からデータが送信されるパターン

このパターンは、機器に対してデータを取得するためのコマンドを送ると、機器が保持しているデータを送信するパターンである。例えばセンサ機器等にコマンドを送り、センサ情報を取得する場合が考えられる。

(4) コマンドの送信を行わなくても機器側からデータを送信するパターン

このパターンは、機器側から断続的または連続的にデータを送信するものである。例えば、センサで取得したデータをシリアルポートに対してデータを出力し続ける機器などが考えられる。

3.4. フレームワークの構成

本フレームワークの構成を図 4 に示す。

—Web ブラウザ

本フレームワークを利用するエンドユーザはパソコンや携帯端末などの Web ブラウザを通じて操作を行う。Web サーバへの機器の情報の登録も Web ブラウザ上から行う。

—Web サーバ

Web サーバ上には Web ブラウザから登録された機器情報が集約されている。Web ブラウザからアクセスがあると利用可能な機器の一覧の HTML データを返す。また、Web ブラウザから送られてきた機器を制御するための GET や POST メソッドからスクリプトを呼び出す。

—PHP/Java スクリプト

Web サーバ上で実行されるスクリプトで、Web サーバ上に登録されている機器情報を元に、動的な HTML を生成する。ユーザが Web ブラウザ上での機器操作を直感的に行うことを補助するために、Ajax や Java スクリプトなどを

用いて動的なページを生成する。また、GET や POST メソッドを用いて送られてきた制御情報をシリアル制御プログラムへ渡す役割をする。

—機器情報

Web ブラウザから Web サーバに登録された機器の情報を XML ファイルの形式でサーバ上に保持している。詳細は 3.3 節にて述べる。

—シリアル制御プログラム

シリアル制御プログラムは Web サーバ上で動作しているスクリプトとシリアルポートの間に位置し、シリアルコマンドの送受信機能の役割を果たす。シリアル制御プログラムでは、3.3 節で述べたようなパターンに対してスクリプトから渡された制御情報を元に、実際にシリアルポートに対して信号を送り、機器の制御を行う。また、機器側の状態などの情報を受け取り、スクリプトに渡す役割をする。シリアルポートは、Web サーバ上から直接利用できるポートだけでなく、シリアル—Ether 変換機を利用した仮想シリアルポートも同様に利用可能である。

—シリアル—Ether 変換機

シリアル—Ether 変換機は 2.4 節で説明したように、仮想シリアルポートから Ethernet 上に送られた TCP/IP パケットを受け取り、機器を制御するコマンドをシリアル機器に送る役割をする。また、シリアル機器から出力されたデータを TCP/IP パケットに変換する。

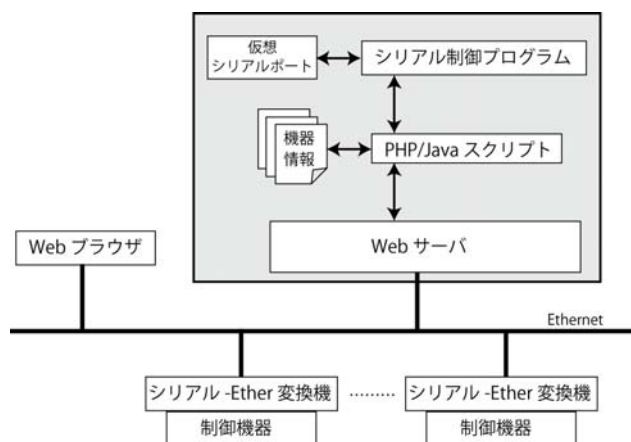


図 4 フレームワークの構成

3.5. Web サーバへの機器情報の登録

シリアル通信で制御を行う機器の場合、2.3 節で述べたように、機器固有のパラメータや、機器を操作するためのコマンドを登録する必要がある。本フレームワークでは、これらの情報を図 5 に示すような Web ブラウザ上の入力フォームを利用することにより Web サーバに登録を行う。機器を Web ブラウザ上から操作する際に利用するボタンなどはフレームワーク内に予め用意されているものを利用する。ユーザによって入力された機器情報は、Web サーバ上では図 6 に示すような XML ファイルによって管理を行う。図 6 の XML ファイルは 3.4 節で述べたシリアル通信のパターンのうち、(1) と (2) のパターンに対応するものである。XML ファイルには

シリアルポート番号や通信速度といった機器を制御するために必要なパラメータと、機器を操作する際に送信するコマンド、機器側から送られてくるレスポンスなどが保存されている。PHP のスクリプトは XML ファイルを参照し、そこに記述されている内容に応じてユーザが機器を操作するための Web ページを生成する。

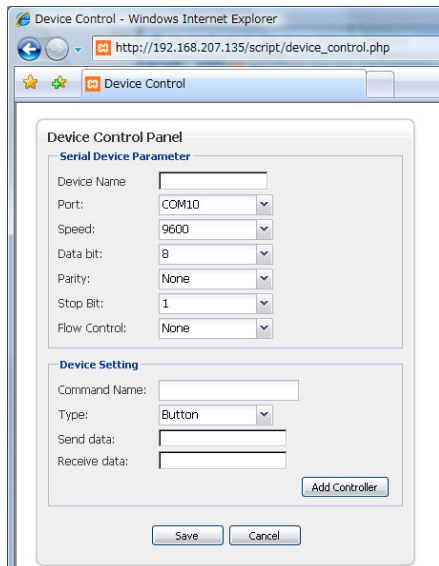


図 5 Web ブラウザ上での機器登録

```
<?xml version="1.0" encoding="UTF-8"?>
<devices>
  <device>
    <description>Projector</description>
    <port>COM10</port>
    <speed>9600</speed>
    <data_bit>8</data_bit>
    <parity>none</parity>
    <stop_bit>1</stop_bit>
    <flow>none</flow>
    <command>
      <name>Switch ON</name>
      <type>button</type>
      <data>#0C24</data>
      <return>OK</return>
    </command>
    <command>
      .....
    </command>
  </device>
  <device>
    .....
  </device>
</devices>
```

図 6 機器情報 XML の例

3.6. 本フレームワークを用いた機器制御例

本フレームワークを用いることにより、プログラミングを行う知識を持っていないエンドユーザでも、Web ブラウザ上から機器の追加を行い、ネットワークを介して機器が利用可能な環境を手軽に構築することができる。

具体的な例として、シリアルコネクタを搭載したプロジェクトタを本フレームワークを用いて Web サーバに登録するまでの手順を説明する。

1) プロジェクトタのシリアルポートとシリアル-Ether 変換機を接続し、Ethernet へ接続する。シリアル-Ether 変換機はネットワーク上の DHCP サーバより IP アドレスを取得し、TCP/IP による通信が可能な状態になる。

2) ユーザは Web ブラウザを用いて本フレームワークの Web サーバにアクセスする。

3) ユーザは機器の情報を Web サーバ上に登録する作業を行う。機器固有のパラメータや機器を制御するためのコマンドは登録する機器のマニュアルを参照し、それらの情報を Web ブラウザ上から設定する。ユーザによって入力された機器制御情報は Web サーバに送られ、XML ファイルとしてサーバ上に記録される。

1) ~ 3) までの手順を行うことにより、新しく追加する機器の情報が Web サーバに登録され、複数のユーザが Web ブラウザ上から機器を利用することができるようになる。このように、機器の登録から実際の利用までの手順を全て Web ブラウザ上で行うことで、エンドユーザでも容易に様々な機器をネットワーク制御可能にすることが可能になる。

4. まとめ

本稿では、ユビキタス環境やスマートルームに存在する様々な機器を、容易にネットワークコントロール可能にするためのフレームワークを提案した。本フレームワークを用いることにより、高度なプログラミング知識を持たないエンドユーザでも、手軽に機器の情報を Web サーバ上に登録することができる。一度登録した機器の情報は複数のユーザで共有することが可能になり、Web ブラウザ上から機器の制御や状態を取得することができる。

今後の課題として、3.3 節であげた(3)や(4)の通信パターンに対応することがあげられる。センサ等から取得したデータはシリアル通信を通じて連続的に送信される場合がある。そのようなデータを利用するためには区切り文字や改行コードでデータの切り出しを行い、エンドユーザにとって扱いやすい形にする必要がある。これを実現するために、予めいくつかの切り出しパターンを用意しておく方法や、より多くのデータに対応できるように正規表現や簡単なスクリプトを Web ブラウザ上から入力できるようにしておく方法などが考えられる。

参考文献

[1] 河口信夫, 稲垣康善: cogma:動的ネットワーク環境における組み込み機器間の連携用ミドルウェア, 情報処理学会コンピュータシステム・シンポジウム, pp.1-8 (2001).
 [2] 河口信夫, 春原雅志: WebCodget : Web サーバに移動して動作する組込み機器向け移動ソフトウェア, 情報処理学会ユビキタスコンピューティング研究会, 2005-UBI-009 pp.41-44 (2005).
 [3] UPnP: <http://www.upnp.org/>
 [4] DLNA: <http://www.dlna.org/>
 [5] Bonjour: <http://www.apple.com/macosx/features/bonjour/>