

通信路を問わない汎用通信フォーマットの提案と何でもリモコンの作成

Enhancing the functional independence of Universal Communication Format
and Implement to Universal Remote Controller

平中幸雄† 渡部良平† 渡邊高成† 武田利浩†
Yukio Hiranaka Ryohei Watanabe Takashige Watanabe Toshihiro Taketa

1. はじめに

インターネットの普及やハードウェアの進化によって、家庭内でネットワークを構築することが珍しいものではなくなった。この流れを受けて、ネットワークと連動する家電が現れてきている。

ところが、そういった機器と PC の間で相互通信を図ろうとすると、その機器に対応したプログラムを用意しなくてはならない。これは標準的なフォーマットが定まっていなかったため生じるものである。

本研究の目的は汎用的な通信フォーマットを創ることである。また、そのフォーマットを用いたリモコンプログラムを試作・検証し実用性について考察する。

2. 汎用通信フォーマット UCF
(Universal Communication Format)

汎用通信フォーマット(以下 UCF)は機器に依存したデータ表現を排除し、あらゆる通信において汎用的に使用する目的で開発しているフォーマットである[1]。

現在開発されている UCF は次の仕様に従う。

- I. データは ASCII 文字で表現する
- II. データの表現は<tag>data.....</tag>という形式にする。
- III. タグ名はデータを受け取るオブジェクト名(デバイスやプログラムを示す)を指す。
- IV. オブジェクトは階層構造を持ち、内部のオブジェクトにデータを転送できる。
- V. <s>タグで送信元のオブジェクト名を表す。
- VI. <def>で新たなオブジェクトを生成できる。

UCF は通信を意識したフォーマットであり、III. で定義したタグ名を、II. で定義したデータの宛先として扱う。言い換えればタグ名をアドレスとして解釈する。こうして汎用的なテキスト表現のみで通信を行える。

しかし、これまでの UCF はアプリケーション層における利用を意識していた。ゆえに、エラーの検出を行う機能や、相手の通信バッファに対して大きいデータを送るときに、データを分割する機能は含まれておらず、TCP/IP を利用することで問題を回避していた。

ところが、あらゆるオブジェクトに対するデータ通信を想定すると、いつも TCP/IP のような信頼性が高い通信を行えるとは限らない。例えば、シリアルケーブルを用いたデータ通信などが挙げられる。

そこで、我々は UCF に次の機能の実装を提案する。

- (A) エラー検出機能
- (B) データ分割機能
- (C) (B)に関連したフロー制御を行う機能

3. 汎用通信フォーマットの機能拡張
及びリモートコントローラの製作・検証

3.1. エラー検出の為の仕様

データや通信環境によってはエラーチェックが必要にならない事から、チェックを行う箇所を指定できるような方法が良いと考え、次に示す手順を採用した。

図 1 の例のような UCF データ表現によるエラー検出を考える。図中の“ A ”はオブジェクト名を示す文字列、“ CRCCode ”は CRC 値を文字列表したもので、“ data ”は“ A ”宛てのデータである。

```
<A>
  <crc>begin</crc>
  ...data...
  <crc>end,CRCCode</crc>
</A>
```

図 1 エラー検出に対応した UCF データ表現例

<crc>begin</crc>と<crc>end, CRCCode </crc>の間の部分データがエラー検出の対象である。

エラー検出機能が有効になっている UCF オブジェクトが、<crc>begin</crc>というデータを受け取った時は、それ以降の文字列をエラー検出対象と判断し、受信した文字列をメモリに記憶していく。

その後、<crc>end,CRCCode</crc>という文字列を受信したらメモリ中にある文字列を基に、CRC 値を生成し、CRCCode の値と比較する。

生成した CRC 値が CRCCode と一致した場合は受信データを正しいデータとして処理する。

もし、一致しない場合は破損データとして扱い、必要に応じて破棄や再送要求を行う。

3.2. データ分割の為の仕様

UCF データを取り扱うオブジェクトに MTU 値を持たせる。MTU 値とは 1 度に送れるデータの最大サイズである。送信しようとするタグ名を含めた UCF データのサイズが MTU 値よりも大きくなる場合は、複数の UCF データに分割して送信する。

以下にその例を示す。図 2 が分割対象となる巨大なデータ、図 3 はデータサイズが MTU 値より小さいデータとなっている。図中の“ A ”はオブジェクト名を示す。

```
<A>...too...long...data.....</A>
```

図 2 長い UCF データの例

```
<A>...too...long</A>
<A>...data.....</A>
```

図 3 分割した UCF データの例

† 山形大学, Yamagata University

3.3. データ分割に関するフロー制御

受け取ったデータを単純に結合すると順序が入れ替わって不正なデータになるおそれがある．そこでオブジェクトは1個の UCF データを受け取る度に、送信側に ACK(応答)を返すように提案した．このやり方ならば、受け取ったデータを着た順に結合することで元のデータに復元することができる．ただしこの手法は遅延が大きいため、改良の余地がある．

4. 何でもリモコンの製作と製作した UCF の検証

提案した UCF を利用する「何でもリモコン」というリモコンプログラムを製作する．またリモコンを動作させて、機能を追加した UCF の検証を行う．

このリモコンのコンセプトは、「あらゆるデバイスを操作できるリモコン」である．UCF を用いることでメーカーに依存したデータ形式や、機器制御の手順を排除している．その結果、メーカーが異なる機器間の通信が容易になったり、複数の制御プログラムを導入したりする手間を省けるといった利点がある．

プログラムの具体的な動作はデバイス側を制御する PC とデバイス間の通信は、機器依存の形式をとり、各 PC 間の通信は UCF を用いて行う．以上の形で前述した汎用性を実現する．

このプログラムは Java で作成したために、JavaVM を導入していれば、どの OS でも利用することが出来る．そのためリモート側で足りないモジュールがあるときは、デバイス側にあるクラスファイルを取得し、リモート側に組み込む事で機能の拡張が行える．以上のような方法で、「何でもリモコン」でデバイス側 PC に接続されている全ての機器をコントロールすることを可能にしている．図 4 にこのプログラムの概要を示した．

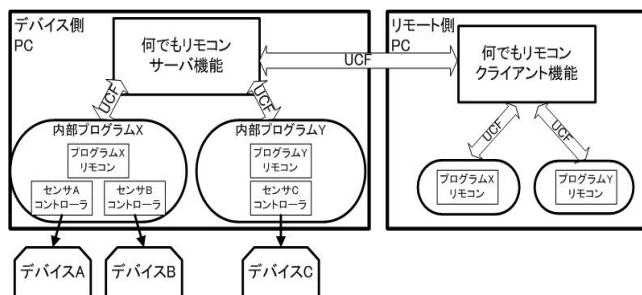


図 4 何でもリモコンの概要

5. 何でもリモコンの動作結果

リモコンプログラムを利用して実際にオシロスコープと USB カメラの制御を行った．エラー検出と分割を有効にした場合でも、オシロスコープの波形レンジの操作や取得、カメラ画像の取得が行えた(図 5)．また、エラーデータを流した際にはそれを検出できた．

以上の結果より、提案したエラー検出機能と、データ分割機能がきちんと動作しているといえる．

右上には UCF 形式で現された画像データを複合して出来た画像がある．右下には UCF 形式の波形データをグラフ化したものが表示されている．左上と、左下にはオシロスコープの制御を行うためのものである．「波形取得」といったボタンが並んでおり、オシロスコープのメーカーやバージョンの違いを意識せずに指示を出せる．

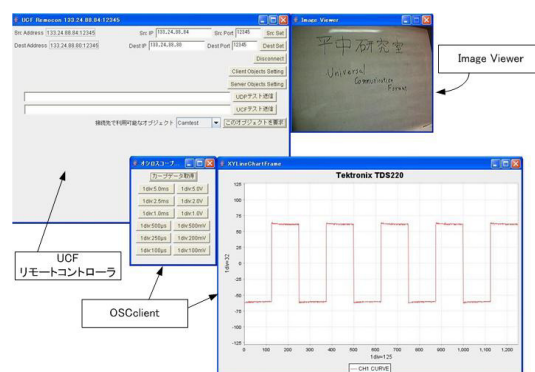


図 5 何でもリモコンの実行結果

6. まとめと課題

今まで提案された UCF に新たな機能を追加することで、TCP/IP プロトコルに対する依存度を小さく出来た．

今回追加したエラー検出機能はエラー検出の対象箇所を指定できる．今までは速度を重視するなら UDP を、信頼性を重視するなら TCP を利用するといった、用途に応じた伝送路の選択を行っていたが、UCF を用いれば、1本の伝送路でいずれの要求も満たすことができる．図 6 に具体例を示した．この図は UCF データによる動画ストリーミングの制御と配信を表している．視聴者による再生要求や停止要求は正確さを重視するためエラー検出を行う．それに対し動画サーバから配信される動画データは速度を重視し、エラー検出を行わない．



図 6 制御データと実データを同じ伝送路上で扱う例

このようにして TCP/IP を用いない通信方法を取るデバイスが UCF を利用する時でも、通信の信頼性を得ることが可能になった．しかもデータ送信側にエラー制御の有無を選択できるという利点があった．全く新しい通信形式が現れた時に、この UCF の柔軟性の高さが役に立つと考えられる．

今回の研究に残された課題としては

- ・ 実験では予め設定した MTU を基にデータ分割を行ったが、動的に最適 MTU を取得すべきである
- ・ 効率性を上げるためにフロー制御の手順の改善が必要である

といった点があげられる．今後はこれらの点をふまえた上で、より効率的な手法を提案する必要がある．

参考文献

- [1] 平中幸雄 菅井栄治 渡部修平 武田利浩
汎用通信フォーマットによる移動型オブジェクトの実現, M-038, 第 4 回情報科学技術フォーラム, 2005 .