



# 家 in ZAURUS の開発

---

2003年8月30日

応募者: 上原 昭宏

応募番号: 1060

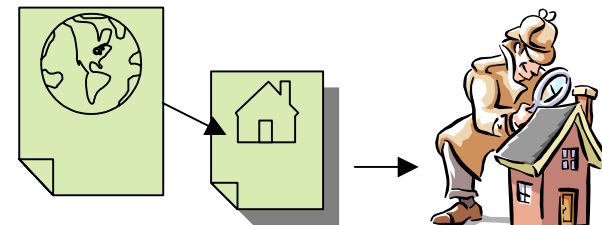
## 目次

1. “家 in ZAURUS” の開発背景
2. “家 in ZAURUS” とは?
3. “家 in ZAURUS”稼動例

# はじめに-不便だぞ!!家電コンセント-

- Ipv4家電コンセントを導入した 携帯で家電がOn/Offできる!!便利だぞ!
- はじめは 携帯からコンセントをOn/Offできて**便利**
- そのうち **不満爆発**. 家のコンセントをOn/Offするのに, いったいいくつのリンクをたどればいいのか? お風呂をONしたら, エアコンを10分後に“自動で”ONして欲しいのに...自分でONしないとだめなの? たくさんコンセントがあるけど, Iアドレスが違うから, いちいち1つずつアクセスしないとだめなんて不便
- 解決策 お家にサーバーを置いて, サーバで一括制御
- サーバーにアクセスするだけで家の家電が制御できるようにした
- これでも不満が出る 離れたところにある家とオフィス(SOHO)を管理するのに, 場所ごとにいちいちサーバー置かないとダメなんて!! 一括管理できないの?

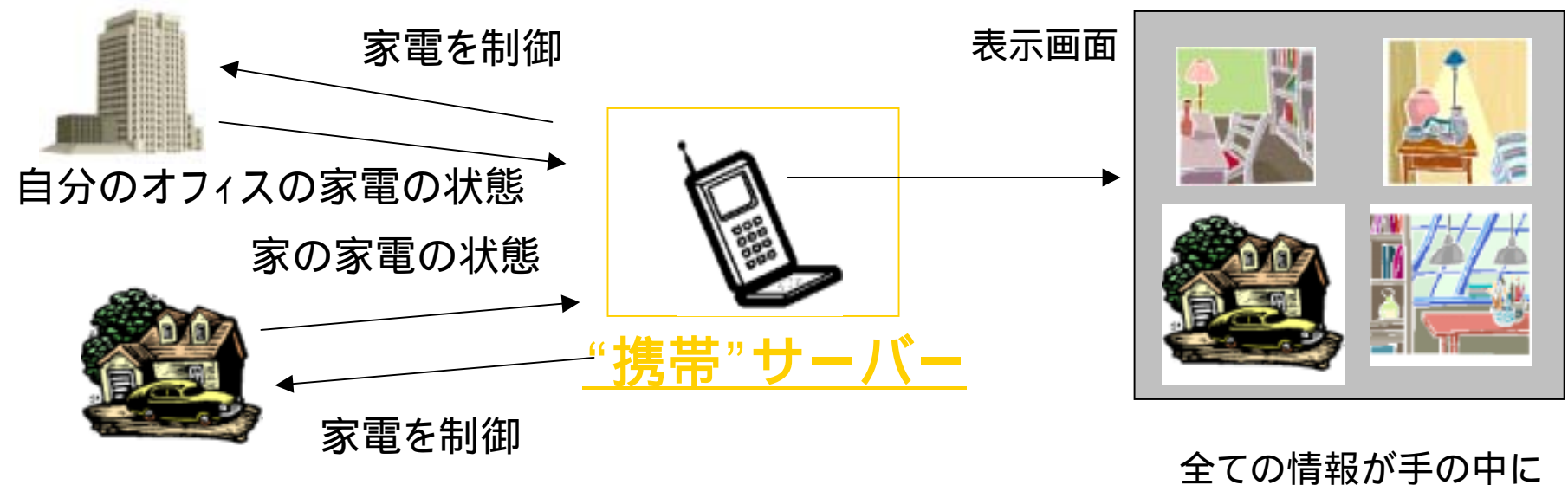
そんなこといわれたって...どうすりゃいいの...  
サーバが“場所”に固定されてるのが悪いんだ!



“うち”までウェブをたどるの面倒

# 提案内容-俺はこれが欲しいんだ-

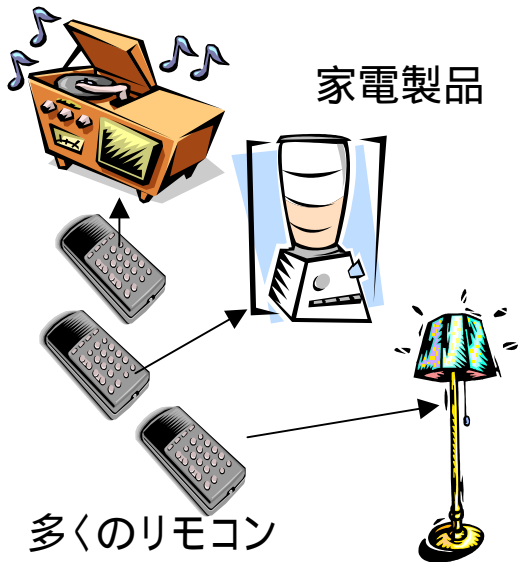
- 制御する“場所“にサーバを置かなくてもいいじゃないか
- “携帯”サーバーで行こう
- サーバーを持ち歩くと,こんなに便利
  - リアルタイム:画面を見れば,即自に家電の状態が把握OK
  - 即時性:いちいちネットにアクセスしなくても即時制御OK
  - 統一性:オフィスだろうが家だろうが場所に寄らず即制御OK



# “携帯サーバ”の実装検討(1): 要求は何か

- ユーザーの立場から、情報家電をどう使いたいかを分析
  - ネットワーク上の“自分の所有している”機器を操作したい
  - (要求)自分の所有している情報家電を操作する“万能リモコン”

## ネットワーク化前



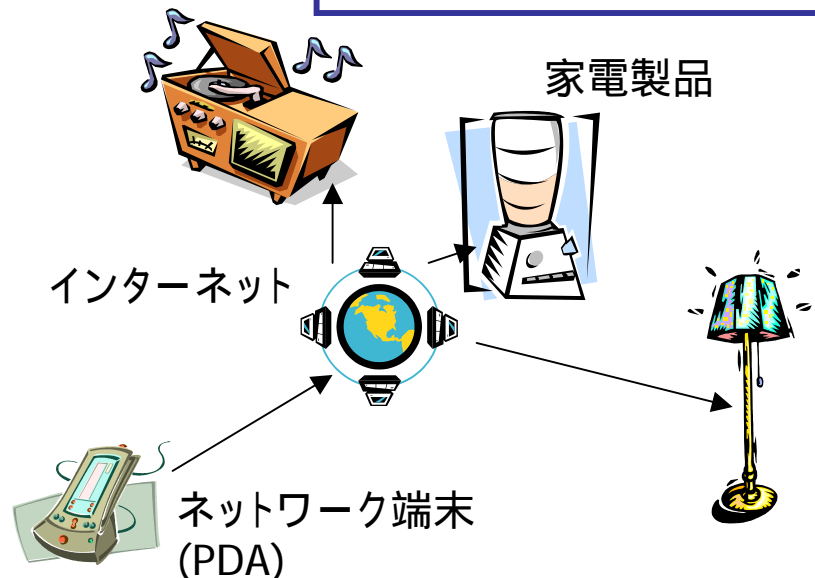
どこからでも  
いつでも  
操作したい



ネット化



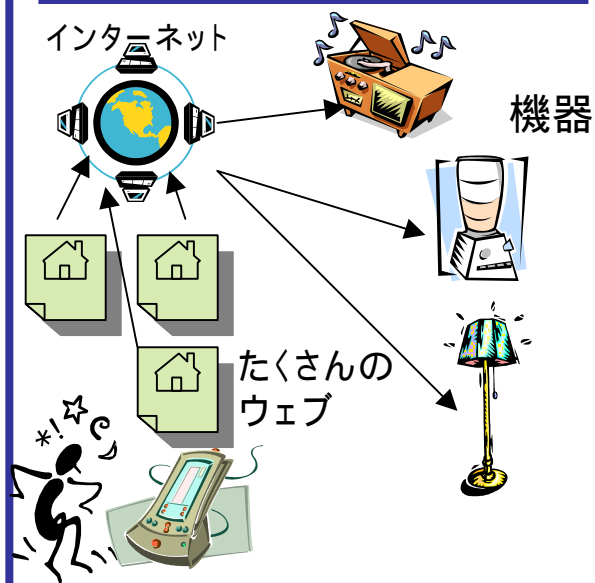
## ネットワーク化後



# “携帯サーバ”の実装検討(2): 現状はどうか

- ウェブ・ブラウザを通して操作するものが多い
  - 最初に述べたように, 操作性に不満がでる
  - (機器構成の分析) ウェブ経由で操作するため, “多数のリモコンを所有”している状態になる  
ネット化前と操作の不便さはかわらない

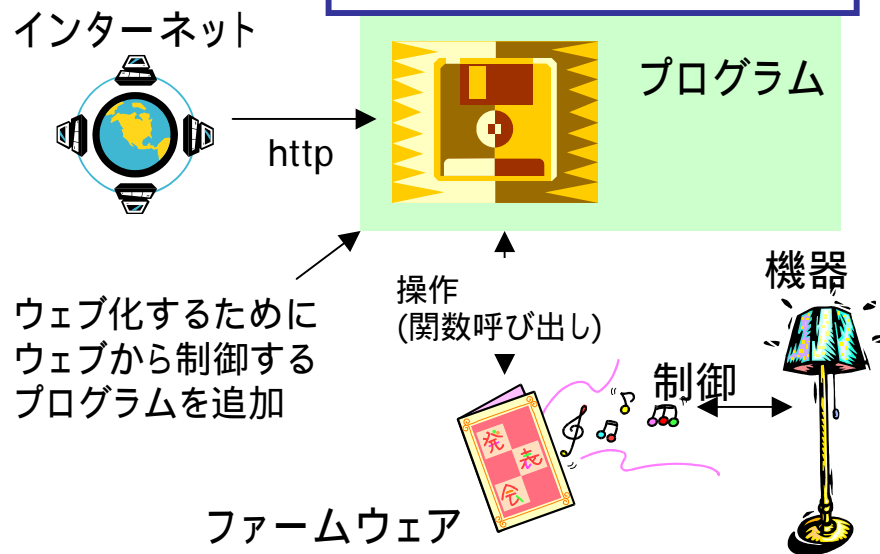
## ウェブ経由で制御



## 機器の内部構成をみてみよう

リモコンをウェブ向けに作り変え

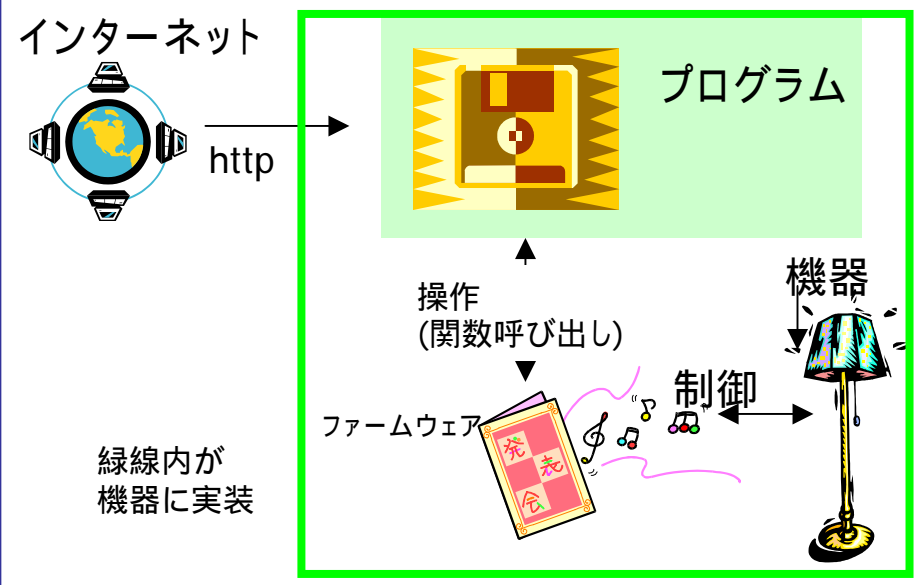
## 機器の内部構成



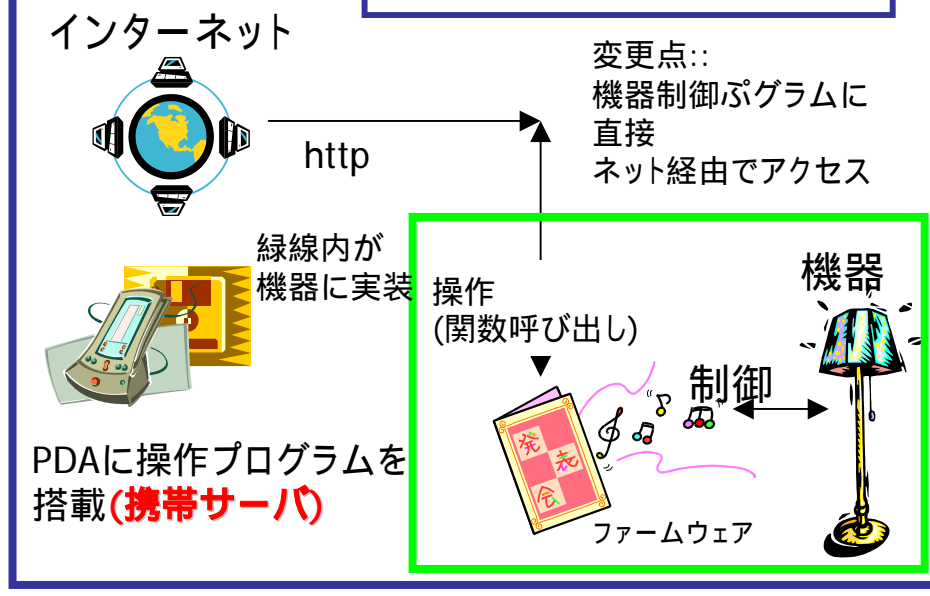
# “携帯サーバ”の実装検討(3): 実装内容

- 情報家電の制御プログラムに, 直接ネット経由でアクセス
- “携帯サーバ”から, 機器を直接操作する
  - 機器操作プログラム(従来の家電のファームウェア)に, ネットワーク経由で, 直接アクセスできるように, プログラムを作る

従来の情報家電内部構成

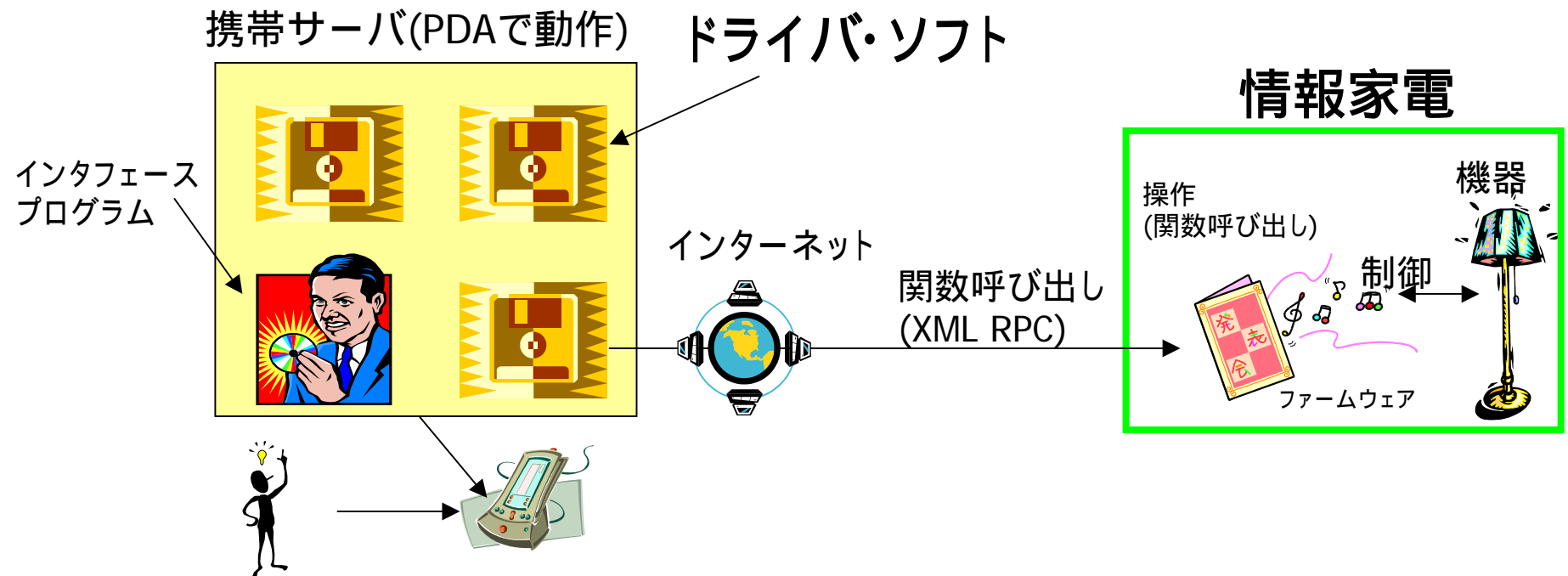


今回の内部構成



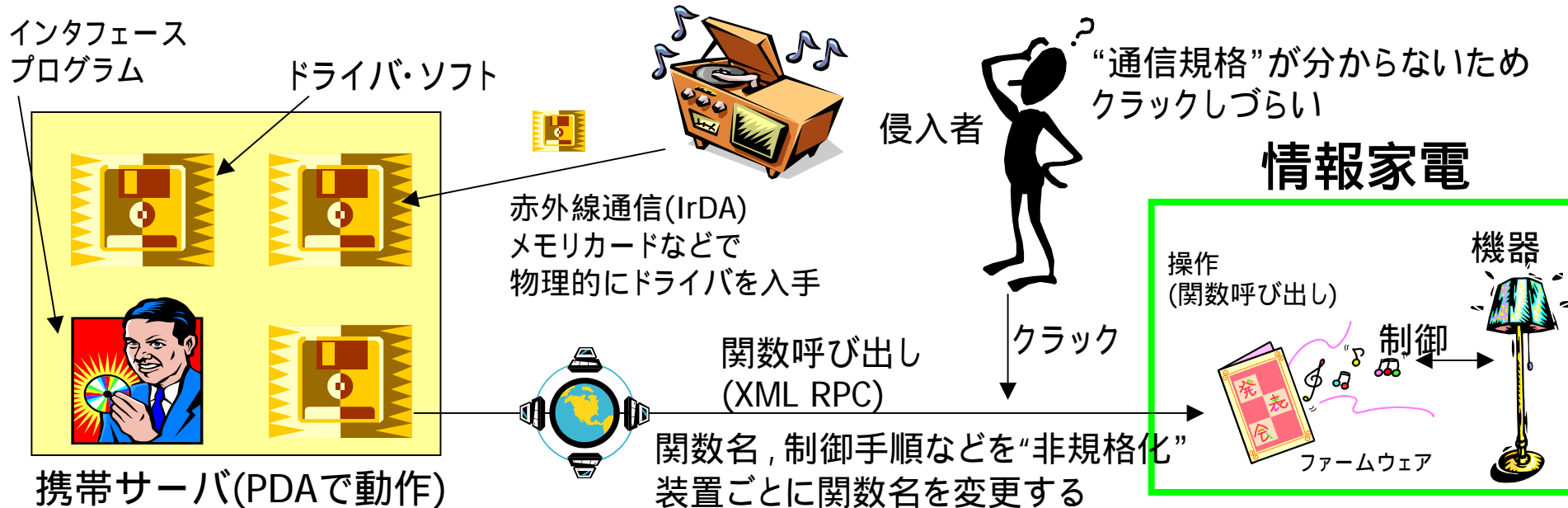
# “携帯サーバ”の実装(1): プロトコルと言語

- ネット経由で操作関数にアクセスするには(API呼び出しの実装)
  - プロトコルにXML RPC( <http://www.xmlrpc.com/spec> )を採用
- 簡単に改造でき, 自分好みにできる“携帯サーバ”を作るには
  - スクリプト言語で作るのがよい      オブジェクト指向言語Rubyを採用




## “携帯サーバ”の実装(2): P&Pとセキュリティ

- 簡単に情報家電が使えるようにするには
  - Plug & Play が不可欠      ドライバソフトウェアをインストールする方式
- セキュリティ: 赤の他人が勝手に情報家電を使えないようにする
  - 工夫1: ドライバソフトウェアは, “家電内部に保存”      物理的に取得
  - 工夫2: ドライバソフトウェアと情報家電間の通信は, “ブラックボックス化”







# “携帯サーバ”の実装(3): なぜ今回の実装形式を選んだのか

## ■ Plug & Play

- C++言語を使用した場合      ドライバのダイナミックローディングなど, 結構ややこしい.
- スクリプト言語で構成したので, 簡単にPlug & Play 化ができました.

## ■ コスト

- 情報家電に対応するためには      家電の製造コストを考えたい.
- XML RPCならば, 最小の製造コストだろうと考察した.

## ■ セキュリティ

- (ユーザの声) 実際に情報家電に触れない人が, 装置を操作できるなんて, おかしいでしょ?      装置に触れる人だけが, ネット経由でも, 操作できるようにしようよ.
- 今回の実装は, 機器呼び出し制御をおこなうプログラムを, PDAで動作している携帯サーバ内の“ドライバソフト”にした.
- ドライバソフトのAPIは, 規格に準拠すればよい.
- 装置の制御手順は, “独自規格”でOK. ドライバソフトが“独自規格”をサポート.

## ■ その他

- 分散オブジェクト技術, XMLのデータ処理技術, IPv6, RPCなどのネットワーク技術, セッション管理など, いろいろな技術を使っています.



# 実装した範囲

---

## ■ 携帯サーバー

- GUIインタフェースにウェブを使用：ウェブのセッション管理を実装
- ドライバソフトのPlug & Play 機能：Rubyのソースコードに直接手書きして実現
- ドライバソフト：XML RPCのクライアント・ソフトとして実装

## ■ 情報家電

- 仮想的な情報家電を，サンプルプログラムとして提供した
- XML RPCのサーバとして動作

## ■ 今後の実装

- 情報家電コンセント(RS6, IPv6普及・高度化推進協議会 様より提供)にXML RPCサーバを搭載し，RS6を制御可能とする
- RS6向けのXML RPCサーバは，フリーソフトとして提供されている．
- あとは，“自分向け”プログラムを書くだけです

# アプリケーション構築例

