

応募番号 : 2058



## IPv6情報家電プロトコルスタックの開発 (CyberLink for IPv6)

2003年6月22日

今野 賢 (Satoshi Konno)

[skonno@cybergarage.org](mailto:skonno@cybergarage.org)

<http://www.cybergarage.org>

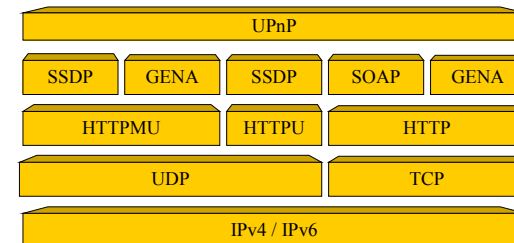
## ■ 情報家電普及に必須なZeroconf(Zero Configuration Networking)

|   | 背景/課題                                | 状況/問題点                       |
|---|--------------------------------------|------------------------------|
| 1 | 情報家電は年々高機能/ネットワーク化                   | セットアップは機器別/複雑化(Zeroconfではない) |
| 2 | IPv6情報家電の利点                          | アドレスの自動割り当て/豊富なアドレス          |
| 3 | 情報家電のZeroconfには、サービス検索/制御レベルまでの対応が必須 | Rendezvous、UPnP™が存在          |
| 4 | Rendezvous?                          | サービス検索後の標準化が未着手              |
| 5 | UPnP™?                               | 仕様がオープンでデバイス標準化/対応機器多数       |
| 6 | IPv6+UPnP™の組み合わせが情報家電Zeroconfに最適     | UPnP™の実装は複雑?                 |

## ■ IPv6+UPnP™の実装は複雑?

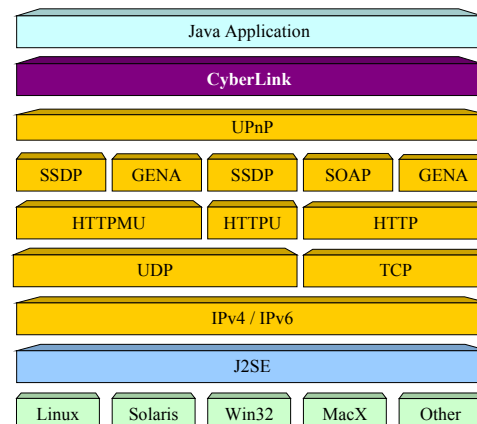
IPv6+UPnP™ 情報家電アプリケーション構築には、IPv6の基礎知識およびUPnP™ のGENA, SSDP, SOAP, HTTPU, HTTPなど多種多様なプロトコルの理解および実装が不可欠

|   | Zeroconfレベル | 実現方法/プロトコル           |
|---|-------------|----------------------|
| 1 | アドレス割り当て    | IPv6, AutoIP(IPv4)   |
| 2 | デバイス検索      | HTTP,HTTPU,SSDP,GENA |
| 3 | サービス確認      | HTTP,SOAP            |
| 4 | サービス/イベント制御 | HTTP,SOAP,GENA       |



## ■ 特徴

- ①世界で唯一オープンソースなUPnP™ Java実装
- ②UPnP™ V1.0 IPv6拡張 (Annex A)への対応
- ③ホストインターフェイスのIPv4/IPv6自動認識
- ④UPnP™の多彩なプロトコル/イベントを自動管理  
→ ユーザーレベルでのUPnP™管理は一切不要
- ⑤サービス/イベントリスナー形式によるユーザー拡張
- ⑥BSDライセンスでの配布



## ■ 全てのJavaアプリケーションをIPv6+UPnP™対応に！

既存JavaアプリケーションをIPv6+UPnP™デバイスとして動作させるには、CyberLinkでデバイス定義/サービスリスナーを設定するだけの手軽さ！

- ①デバイス定義(Description)XMLファイルの作成
- ②サービスリスナーの設定

The diagram shows two code snippets. The first is an XML file for device description, and the second is a Java class implementing the UPnPDevice interface. A yellow arrow points from the first list item to the XML snippet, and another yellow arrow points from the second list item to the Java snippet.

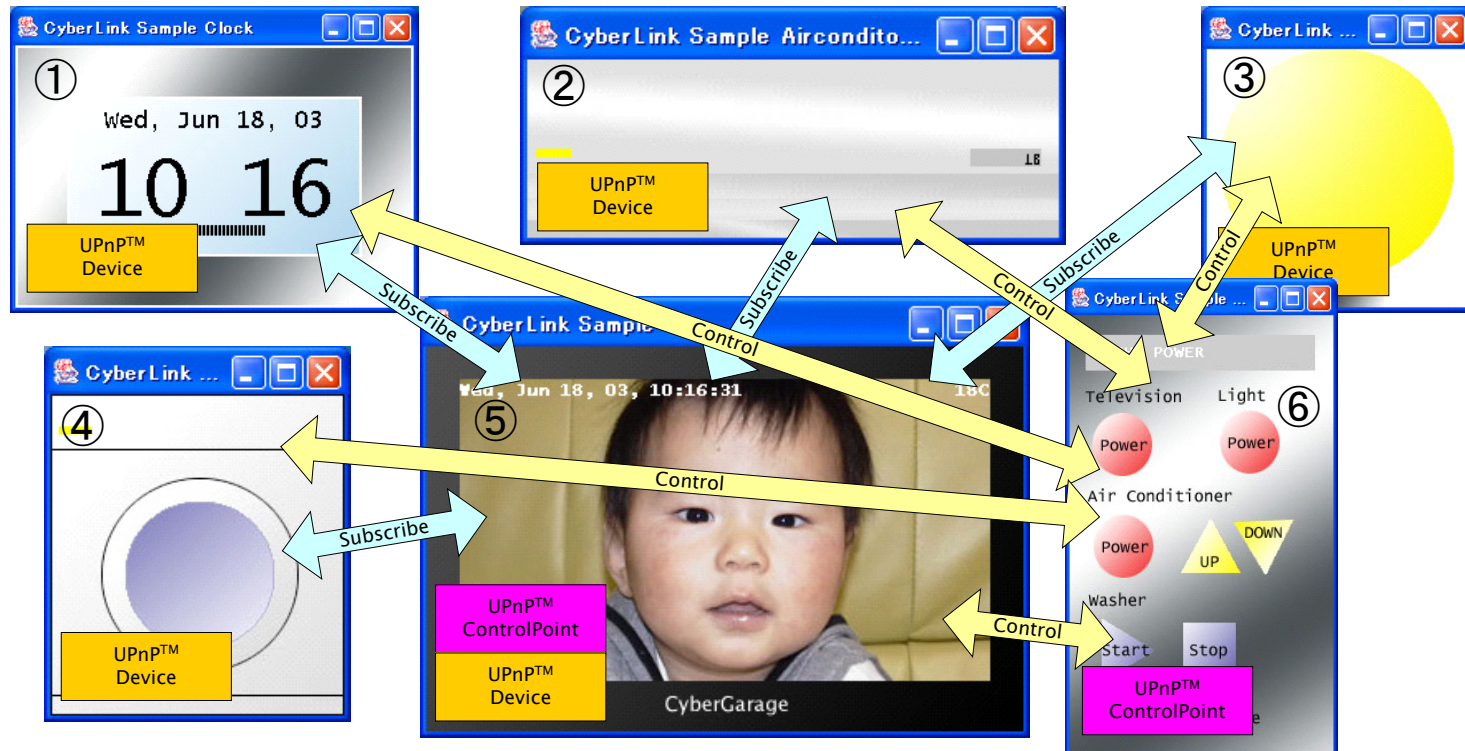
```
<?xml version="1.0" ?>
<root xmlns="urn:schemas-upnp-org:device-1-0">
  <device>
    <serviceList>
      <service>
        <SCPD>
        </SCPD>
      </service>
    </serviceList>
    <presentations>
    </presentations>
    <iconList>
    </iconList>
    <url>
    </url>
  </device>
</root>
```

```
public class UPnPDevice extends Device implements ActionListener, QueryListener {
    public UPnPDevice() {
        super("xxx:description.xml");
        Action action = getAction("xxxx");
        action.setActionListener(this);
        Service service = getService("xxxx");
        service.setQueryListener(this);
    }
    public boolean actionControlReceived(Action action) {
        ArgumentList argList = action.getArgumentList();
        String actionName = action.getName();
        if (actionName.equals("xxxx") == true) {
            return true;
        }
    }
    public String queryControlReceived(Service service, String varName) {
        if (varName.equals("xxx") == true) {
            return "xxxx";
        }
        return "";
    }
}
```

## ■ 次期バージョン計画

- ①コンパクト版(ミニマムXMLパーサー開発) – J2ME(携帯)レベルを念頭に
- ②C++版(開発中) – 対応プラットフォームの拡充

## ■ 情報家電アプリケーション構築サンプル



|   | デバイス | 提供サービス            | 制御サービス    | 提供イベント    | 購読イベント    |
|---|------|-------------------|-----------|-----------|-----------|
| ① | 時計   | 時刻取得              | -         | 時刻        | -         |
| ② | エアコン | 電源ON/OFF, 温度設定    | -         | 電源/温度状態   | -         |
| ③ | 電球   | 電源ON/OFF          | -         | 点灯状態      | -         |
| ④ | 洗濯機  | 電源ON/OFF, 洗濯開始/停止 | -         | 電源/洗濯進行状態 | -         |
| ⑤ | テレビ  | 電源ON/OFF          | -         | 電源状態      | ①～④提供イベント |
| ⑥ | リモコン | -                 | ①～⑤提供サービス | -         | -         |