

論文

無線データ通信組み込み応用装置の開発

佐藤正利* 森 久直* 高野哲寿* 坂巻佳壽美* 宮田勝雄*

Development of an embedded wireless-data-communication system

Masatoshi SATO, Hisanao MORI, Norihisa TAKANO, Kazumi SAKAMAKI and Katsuo MIYATA

Abstract A technology which includes the functioning of a wireless LAN and Bluetooth in a Linux board was established. Since it can be utilized for inclusion in the existing product, as well, the inclusive technology of the wireless-data-communications obtained in this research and development will also be effective from the viewpoint that it will give added value to products.

In order to utilize the functioning of a wireless-data-communications (wireless LAN and Bluetooth), an inclusion Linux board was adopted. A wireless-data-communications function is incorporated on an open source Linux, and simultaneously with a power supply injection, it becomes possible to start a kernel starting, automatic login, and application program. It enables interface with loading devices to give flexibility to programming which operates the GPIO port on a Linux board. It was applicable as a form of distant place monitor loading technology for use with simple types of camera equipment.

Keywords Wireless-data-communications, Wireless LAN, Bluetooth, Linux board, Camera equipment

1. はじめに

中小企業において、IT化の一端である無線データ通信を利用した組み込み製品への期待は大きい。その要求に応えるため、本研究開発において無線データ通信を組み込む場合に必要となる開発環境の構築と、無線データ通信の特性を考慮した、最適な組み込み装置を開発することを目的とした。無線データ通信には、無線LAN、Bluetooth、UWB、ZigBeeなど、様々な規格がある。現在普及している通信規格は、無線LANとBluetoothであり、この2つを組み込んだ応用装置を開発した。同時に、開発環境の構築も行った。

ここで開発した無線データ通信ユニットは、小型基板の中に必要な機能が全て組み込まれている。そのまま既存の機器へ組み込むことや、単体機能で動作する製品を設計することに適している。

また本研究開発では、オープンソースであるLinuxを活用した。Linuxの活用により、安価で信頼性のある無線データ通信組み込み装置を開発でき、広い応用分野に適用できる。各種設定や制御コマンドを追加するという最小限度の開発工数で、無線データ通信の製品を開発することが可能である。

2. 開発環境の構築とドライバの開発

2.1 無線LAN

無線データ通信機能を組み込むためには、クロス開発環境が必要となる。本開発研究では、フリーのソフトウェアを組み合わせることで、カスタマイズが行いやすく、操作性の優れた開発環境の構築を行った。PC上でターゲットとするCPUを搭載した組み込み基板に適した開発を行うことができ、アプリケーションソフトウェアの開発も行いやすく、また多くのCPUに対応することが可能である。

開発言語として、gccをベースにしたC言語を活用したことで、アプリケーションソフトウェアの開発が容易となる。また、UNIXエミュレータ上で、プログラム開発とLinuxカーネルの再構築が、簡易にできることに重点をおいた。

小型Linux組み込みボード上で動作するカーネルやアプリケーションソフトウェアを開発するための機能、Linuxボード上でのプログラムの動作を確認する機能、開発用PCとLinuxボードとのデータ転送を可能にする機能の3点を搭載したクロス開発環境である。実際に開発を行っている状況写真を図1に、クロス開発環境の動作画面を図2に示す。

*情報システム技術グループ



図1 開発中の状況

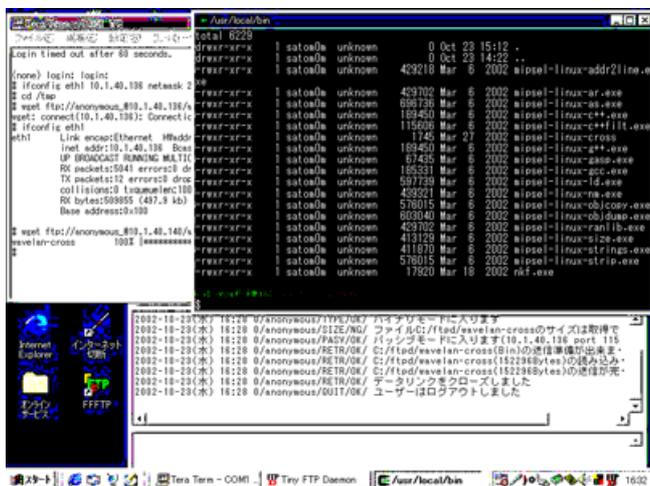


図2 無線LANのクロス開発環境

無線データ通信の組み込み開発に使用したLinuxボードの仕様を表1に示す。

表1 小型Linuxボード仕様

| | |
|-----------|---|
| CPU | VR4181 66MHz |
| SDRAM | 16MB |
| Flash ROM | 16MB |
| シリアル | RS232C * 1ch |
| LAN | 10BASE-T 1ch |
| 拡張 BUS | VR4181 系擬似 ISA 信号 (64pin コネクタ * 2 主として外部入出力に使用) |
| 拡張カード I/F | Compact Flash |
| 消費電力 | 0.55W(最大 1.5W) |
| OS | Linux (kernel2.4) |

カーネルの再構築やアプリケーションプログラムの開発は、主として UNIX エミュレータ上で行い、データ転送には、FTP サーバとクライアント機能を使用した。また、Linux ボード上での動作確認は、telnet クライアント機能のコンソール上で行うようにした。

開発手順のブロック図フローを図3に示す。

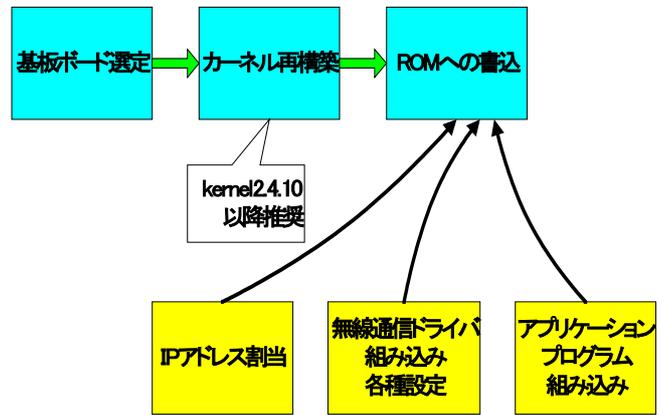


図3 無線LANの開発手順

小型LinuxボードのROM上には、すでに初期設定されたカーネルが搭載されている。このカーネルでは、無線LANが使用できないため、カーネルを再構築し、無線LANが使用できるようにしなければならない。

ここで使用した無線LANは、市販の無線LAN PCカードであり、/drivers/pcmcia/ディレクトリの pccard.h 内にその記述を追加した。クロス開発環境上でコンパイル、make して作成された vmlinux.srec をボード上のフラッシュROMへ書き込む。無線LANカードをLinuxボードのコンパクトフラッシュ端子へ挿し込み電源を入れると、eth1として認識する。

ifconfig コマンドで IP アドレスを割り当てることにより通信が可能になる。

OSがLinuxであるため、オープンソースとして公開されている wireless tools をクロスコンパイルして使用することもできる。wireless tools をボード上に転送後、ifconfigによりIPアドレス、ネットマスクを設定する。

```
# ifconfig eth1 192.168.1.2 netmask 255.255.255.0
iwconfigにより eth1 のモードをインフラストラクチャモードにする。
```

```
# iwconfig eth1 mode "infrastructure"
# iwconfig eth1
# ifconfig eth1 up
```

この操作により、無線LANアクセスポイントとの通信が可能になり、イントラネット上からアクセスできるようになった。

2.2 Bluetooth

Bluetoothの開発には、プロトコルスタックの入手が必要になる。今回の開発では、開発キット(UBQ-Blue)添付のプロトコルスタックを使用した。APIリファレンスの中からライブラリとして使用できる関数群を利用し、上位のアプリケーションプログラムを作成する。作成したプログラムソースとライブラリを一緒にコンパイルすることで、Bluetoothのアプリケーションを完成させる。

アプリケーションプログラムを作成し、ライブラリとリンクしてコンパイルしたバイナリコードを開発基板上にダウンロードし、搭載マイコン上で実行することで、Bluetooth組み込みの開発を行うことができる。

Bluetoothの開発環境構築は、次の手順により行った。

① Bluetoothモジュールの選定

Bluetoothモジュールは、太陽誘電社製EYSF2SAXX(UARTインターフェース)を選定。

② HCI transportの選定

HCI Transport LayerとしてHCI UART(RS232C)を使用した。

③ CPUの選択とOSの選定

CPUは開発キットに搭載されているSH-2とし、OSは無線LAN開発環境と同じLinuxに組み込むことにした。

④ ユーザーカスタマイズ

Windows搭載PCにGNUProを導入し、SH-2用gccを使用する環境を導入した。また、作成するプログラムのリモートデバッグ用としてVisualMonitorを利用した。

評価ボードへのプロトコルスタックのインストールは、GNUPro環境より以下の手順で行った。

評価ボード用ライブラリの「ubqsh2kit-1.16.tar.gz」と「ubqsh2kit-reali-1.16.tar.gz」を作業フォルダ(/work)にコピーして展開後、GNUProを起動し、作業フォルダに移動し以下のコマンドを実行する。

```
# tar xvzf ubqsh2kit-1.16.tar.gz
```

```
# tar xvzf ubqsh2kit-reali-1.16.tar.gz
```

作業フォルダの下に「ubq-blue」フォルダが作成される。次に作業フォルダに「reali/lib」というフォルダ(/work/reali/lib)を作成し、Visual Monitorの「Sh2ul.a」とREALiの「REALiSH2.a」を「reali/lib」フォルダにコピーする。

次に「uartio.c」ファイルを使用する太陽誘電モジュール用にカスタマイズする。

⑤ HCIレイヤとOSALレイヤを作成

USBまたは他のいずれかのHCIトランスポート経由でプロトコルスタックを実行するために、HCIをポーティングする。また、別のOSでのプロト

コルスタック実行を可能にするために、OSALをポーティングする。

⑥ 各レイヤのバッファサイズ等のパラメータ設定を行う。

⑦ 開発キットで使用するアプリケーションを作成する。ここで作成するアプリケーションは、C言語によって作成したものであり汎用性がある。

以上の手順により、Bluetooth開発に必要な環境を構築した(図4)。

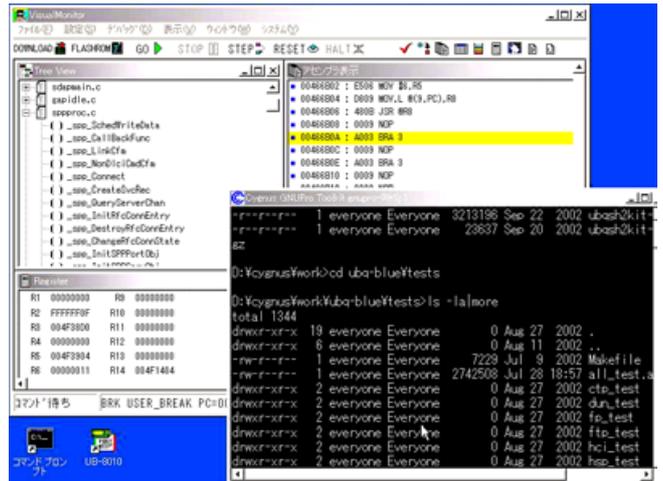


図4 Bluetoothのクロス開発環境

本開発は、SH-2をターゲットに開発したが、他のCPUをターゲットにした場合を考慮して、ターゲットの非依存部と依存部を切り分けて開発を行った。開発キットでは、依存部と非依存部に関する情報がオープンになっていたため、依存部は開発キットのコマンドをそのまま使用し、非依存部に関してはCPUターゲットに適した形式で、C言語で開発した。なお、本開発において対応したスタックレイヤを表2に、プロファイルを表3に示す。

表2 プロトコルスタックレイヤ対応表

| レイヤ名 | 概要説明 |
|--------|--|
| HCI | スタックとBluetoothモジュール間の標準的な通信プロトコルを提供する。 |
| L2CAP | ACLタイプのデータの処理、特に長メッセージのセグメント化を行う。 |
| SDP | アクセス可能なデバイスを検索、デバイスのサービス取得、サービス属性にアクセスする。 |
| RFCOMM | 仮想RS232Cリンク機能を実行する。多くのプロファイルが、データ送受信の手段としてRFCOMMを使用する。 |

表3 プロファイル対応表

| プロファイル名 | 概要説明 |
|---------|--|
| GAP | Bluetooth デバイス間の接続を管理する。GAP は、アプリケーション・プログラムがスタックを構築し、遠隔デバイスへの接続を管理する。 |
| SDAP | 遠隔 Bluetooth デバイスにサポートされているサービスを問い合わせるのに用いられる。SDAP は、アプリケーション・プログラムが遠隔デバイスで使用可能なサービスを検索するのに用いる。 |
| SPP | Bluetooth インターフェース越しのエミュレートされた仮想シリアルポートを提供する。SPP は、アプリケーション・プログラムが標準シリアルポートに類似した方法でデータ送受信を行うのに用いる。 |

3. 無線データ通信ユニットの開発

ここでは試作開発として、無線 LAN, Bluetooth 機能を搭載した簡易カメラ装置を開発し、製品への応用が可能であることを確認した。

組み込みボードは、Linux を搭載した L-Card+ を使用し、最終的に LinuxOS を活用した無線 LAN や Bluetooth 組み込みカメラ装置を開発することができた。開発工程を以下に示す。

① 無線データ通信モジュール（ドライバ）の組み込み
カーネル構築時にモジュールとして組み込む方法が Linux では一般的だが、カーネル起動後 insmod 関数で組み込む方法をとった。アプリケーション的な使い方ができ、デバッグも行いやすい。

② 搭載カメラドライバの開発

カメラ出力は YUV 信号で出ている。その信号を RGB 変換し、JPEG 画像として取り出せる環境を作り上げる。

YUV 信号から RGB 信号への変換式を以下に示す。

$$R = (U - 128) + Y$$

$$G = 0.98Y - 0.53(U - 128) - 0.19(V - 128)$$

$$B = (V - 128) + Y$$

③ CPU の GPIO 端子を制御するドライバの作成

VR4181(CPU)には、16 個の GPIO ポートがある。この GPIO ポートを制御するためのドライバを作成した。様々な入出力に対応した割り込み機能が必要なため、Linux で用意されている /dev/mem を利用した I/O 操作では対応できない。そこで、起動時にシステムに組み入れることができ、I/O デバイスの操作をファイルの操作と同じ形式で扱え、Linux コマンドから直接動作させることが可

能なドライバを作成した。

コンパイルして出来上がった GPIO オブジェクトファイルを組み込みボード上に転送し、デバイスファイルを作成する。GPIO デバイスファイルを操作することで、GPIO に接続された I/O デバイスの制御ができた。

④ CGI により動作可能な機能の搭載

Linux ボード上で http デーモンを動かし、カメラ制御用 CGI プログラムを実行した。これにより、PC 上のブラウザを通して画像の確認ができるようになった。CGI プログラムは、Perl ではなく C 言語にて開発を行った。

以上の開発工程を経て製作した簡易カメラ装置を、図5に示す。



図5 簡易カメラ装置

4. まとめ

本研究開発において、以下の成果が得られた。

- ① 無線 LAN および Bluetooth の機能を活用するために、組み込み Linux ボードを採用し、Linux 上に無線通信機能を組み込み、カーネルの立ち上がりと共に自動的にプログラムを起動することが可能になった。LinuxOS の使用により、既存の Linux 資源を有効に利用することができ、開発期間の短縮が図れた。
- ② Linux ボード上の CPU が持っている GPIO ポートを自在に操作するプログラミングにより、多くのインターフェースへの対応ができることを実証した。
- ③ 簡易カメラ装置を開発したことで、データ転送やインターフェースの開発が確認でき、無線データ通信技術を組み込んだ遠隔制御・遠方監視に有効に対応できる技術を確立した。

参考文献

- 1) UBQ-Blue アプリケーションノート・ポーティングガイド, キヤノン・アイテック株式会社(2002).
- 2) エンジニアリング Linux 応用技法, CQ 出版社(2002).