

PHS通信モジュールW-SIMを利用したLinuxベース携帯電話の開発

第4回

PHS 携帯電話ミドルウェアやデバイス・ドライバの開発

河合孝時
福島孝浩

これまで3回にわたり、Sandgate W-SIM Phoneを利用したアプリケーション・ソフトウェア開発について説明してきた。第1回は開発環境を、第2回はアプリケーション・フレームワークを、第3回は各種ミドルウェアの利用方法を中心に解説した。今回は、デバイス・ドライバの追加方法やそれを利用するミドルウェアの開発、そしてミドルウェアを利用するアプリケーション・ソフトウェアの開発方法について説明する。(筆者)

1 開発内容

今回は、文字入力アプリケーション・ソフトウェアの作成を例に解説を進めていきます。現在、ほぼすべての携帯電話端末にはFEP(Front End Processor)が搭載されており、日本語文字入力がサポートされています。通常、その文字入力は、携帯電話端末上のボタンを押すことによって行われます。

しかし今回は、ホスト・パソコンから入力した文字列をそのままSandgate W-SIM Phone上に表示する、というちょっと変わった機能を実現してみます。今回想定するホスト・パソコンとSandgate W-SIM Phoneの構成図を図1に示します。

最近の携帯電話にはBluetoothが搭載されるようになり、Bluetooth対応のキーボードを使用することで、同じような機能を実現できます。Sandgate W-SIM PhoneもBluetoothを搭載しています(ソフトウェアは未サポート)。今回はより簡単に実現するために、Bluetoothではなく開発キット付属のUSBケーブルを利用して文字データのシリアル通信を行います。このために、USB Serial Gadget^{注1}の機能を利用します。

注1: USB Gadgetとは、USBケーブルを介してホスト・パソコンとデータ通信を行うためのもので、<http://www.linux-usb.org/gadget/>で各種APIフレームワークが定義されている。今回は、その中のシリアル・インターフェースを使用する。

ここでは、まずホスト・パソコンのキーボードからの入力データを、ターミナル・ソフトウェア minicom を使用してUSBデバイスに送り込みます。そして、Sandgate W-SIM Phone側からそれらのデータを読み込みます。今回作成するSandgate W-SIM Phone上のソフトウェア構成を図2に示します。

USB Serial Gadget ドライバは、ホスト・パソコンからの入力を受信します。ミドルウェアは、Linuxのデバイス・ファイルを介して、USB Serial Gadget ドライバからデータを吸い上げ、Sandgate W-SIM Phone アプリケーション・フレームワークを利用し、該当アプリケーション・ソフトウェアに通知します。通知を受けたアプリケーション・ソフトウェアは、受信したデータを文字列として画面上に表示します。

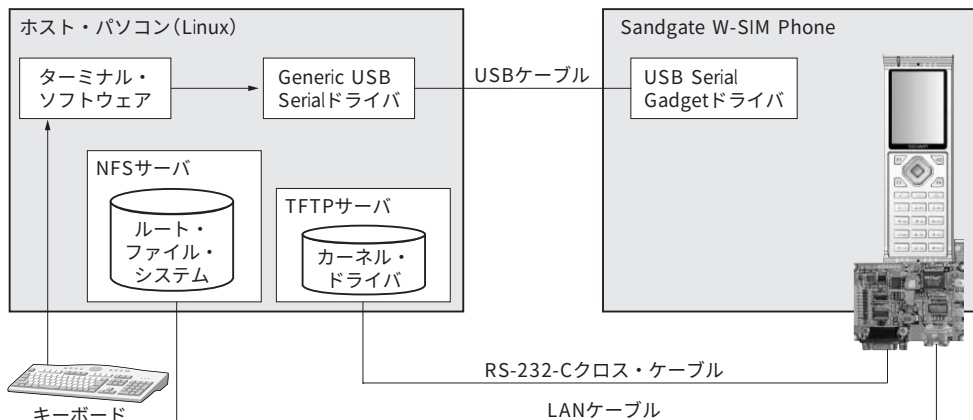
ホスト・パソコンにもデバイス・ドライバを追加する必要がありますので、その手順については後ほど説明します。本機能の実現を例にとって、Sandgate W-SIM Phoneのデバイス・ドライバやミドルウェア、アプリケーション・ソフトウェアの追加について順に説明していきます。

2 NFS 環境の準備

連載第1回(2006年12月号, pp.121-128)でクロス開発環境の構築方法について説明しました。ここでもう一度、NFS環境によるシステムの起動方法について簡単に説明します。ホス

図1
ホスト・パソコンと Sandgate W-SIM Phone の関係

Sandgate W-SIM Phone とホスト・パソコンは、シリアル・クロス・ケーブルとLANケーブルで接続され、NFS環境(デバッグ環境)によって動作する。今回はUSBケーブルを介してシリアル通信を行う





ト・パソコンについては、ここでは、Fedora Core4 のフル・インストール環境を前提にしています。

● クロス開発環境を構築

まず、クロス開発環境を /opt にインストールします。

```
# mkdir -p /opt
# cd /opt
# tar -jxvf [CDROM]/bin/
tools-SGWP-V1.0L10.tar.bz2
```

インストールが成功すると、/opt に xscale-V1.0L10 というディレクトリ名でホスト・パソコンと Sandgate W-SIM Phone 本体のクロス開発環境が展開されます。

● ヘッダ・ファイルとライブラリをインストール

ヘッダ・ファイルとライブラリのインストールを行います。

```
# cd /opt/xscale-V1.0L10
/i686-pc-linux-gnu/
arm-xscale-linux-gnueabi/
# tar -jxvf [CDROM]/bin/sysroot-SGWP-
V1.0L10.
tar.bz2
```

ヘッダ・ファイルとライブラリの格納ディレクトリは、以下のように、コンパイル時のパスに設定する必要があります。

```
[ヘッダ・ファイル]
/opt/xscale-V1.0L10/i686-pc-linux-gnu/arm-
xscale-linux-gnueabi/sys-root/usr/wtk/
include
[ライブラリ]
/opt/xscale-V1.0L10/i686-pc-linux-gnu/
arm-xscale-linux-gnueabi/sys-root/usr
/wtk/lib
```

● ホスト・パソコンの環境を設定

クロス・コンパイラを PATH に設定し、PKG_CONFIG_PATH を設定します。

```
# export PATH=/opt/xscale-V1.0L10/i686-pc-
linux-gnu/bin:$PATH
# export PKG_CONFIG_PATH=/opt/xscale-V1.0L10/
i686-pc-linux-gnu/arm-xscale-linux-gnueabi/
sys-root/usr/lib/pkgconfig
```

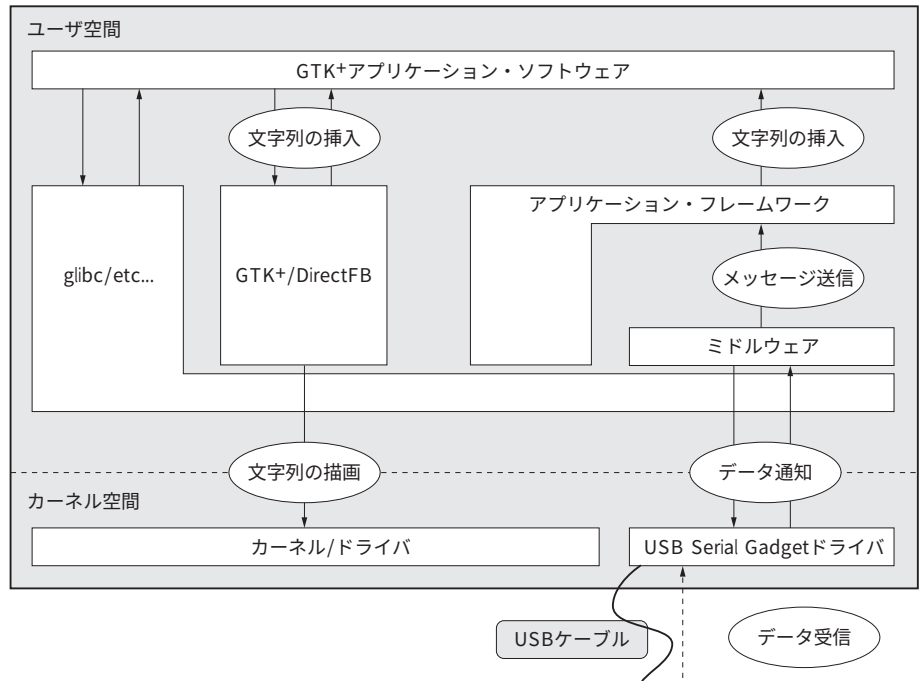


図2 Sandgate W-SIM Phone のソフトウェア構成

USB Serial Gadget ドライバは、ホスト・パソコンからのデータ受信を行う。ミドルウェアはデバイス・ドライバへのデータ受信を監視し、アプリケーション・フレームワークを利用して各アプリケーション・ソフトウェアに対して通知を行う。通知を受けたアプリケーション・ソフトウェアは、受信したデータ(文字列)を画面に描画する

● ルート・ファイル・システムとカーネルをインストール

ルート・ファイル・システムを /work にインストールします。

```
# mkdir /work
# cd /work
# tar -jxvf [CDROM]/bin/rootfs-SGWP-V1.0L10.
tar.bz2
```

インストールが成功すると、/work に rootfs-SGWP-V1.0L10 というディレクトリ名でルート・ファイル・システムが展開されます。

カーネルを tftp の公開ディレクトリである /tftpbboot に展開します。

```
# cp -p [CDROM]/bin/kernel/sandgatewp-
V1.0L10.zImage /tftpbboot
```

● 実機を起動

RedBoot(ブート・ローダ)でカーネルをメモリにロードし、ルート・ファイル・システムのマウント情報を指定してカーネルを起動します。

```
RedBoot> load -r -v -m tftp -b 0xA1000000
-h 192.168.0.1 sandgatewp-V1.0L10.zImage
RedBoot> exec -c "console=ttyS0,115200"
```