
インターフェイスの街角 (56) — GPS 携帯電話の活用

増井俊之

au (KDDI) は、2002 年 4 月から 144Kbps の高速通信をサポートする「CDMA2000 1x サービス」を開始しました。これにともない、このサービスに対応した「第 3 世代携帯電話」(写真 1)の人气が高まっています。

au の第 3 世代携帯電話では、高速通信だけでなく、以下のようなサービスも利用できます。

- WAP 2.0
- 高精度の位置情報サービス「eznavigation」
- 動画配信サービス「ezmovie」
- Java サービス「ezplus」

WAP 2.0 への対応で、HDML (Handheld Devices Markup Language) で記述したページだけでなく、i モード用に Compact HTML で記述されたページ(i モード対応のいわゆる「勝手サイト」)の閲覧も可能になりました¹。NTT ドコモの i モード携帯電話や FOMA でも、Java や動画配信などのサービスがおこなわれています。しかし、正確な位置情報サービスが使えるのは、現在のところは au の携帯電話だけです。

今回は、GPS 携帯電話を用いた位置情報取得の方法について解説します。

au の位置情報サービス

PHS や携帯電話では、端末と基地局との距離や方向などの情報をもとに、端末のおおよその位置を計算する位置情報サービスが何年も前からおこなわれていました。しかし、この方法では端末のおおまかな位置しか分からず、精度さが求められる用途には不向きでした。

¹ 公式サイトは閲覧できません。

写真 1 GPS 機能付き第 3 世代携帯電話 (A3014S)



一方、GPS はカーナビなどでひろく使われていますが、携帯端末で使おうとすると、通常は衛星の捕捉に何分もかかってしまいます。さらに、屋内では GPS 信号が使えないため、実用性という点ではいまひとつでした。

au の eznavigation サービスでは、gpsOne という技術が使われています。これは、GPS と基地局の両方の情報を用いて上記の問題を解決しようというものです。これによって、正確な位置情報を高速に取得できるようになりました。

GPS による携帯情報サービスの歴史

GPS 衛星信号のみを利用して端末の位置を計算しようとする、使用可能な衛星の捕捉から始めなければなりません。携帯端末単体で衛星の捕捉や信号の解析、位置の計算をおこなうと、通信や計算にかなりの時間がかかってしまいます。この問題を解決するために、SnapTrack という技術が開発されました。この技術では、端末は衛星から取得した信号のスナップショット・データをサーバーに送

ります。そして、サーバー側でデータを解析して結果を端末に返すため、端末の位置情報が高速に取得できます。

この技術を開発した SnapTrack²は、1999 年ごろから NTT ドコモや NEC などの企業と提携を進めました。これにともない、位置情報を用いた携帯端末向けの各種のサービスが増えてきました。たとえば、NTT ドコモは 2000 年 1 月に SnapTrack 技術を使った「Naviewn」という携帯端末を発売し、「どこ Navi」というサービスを開始しました³。一方、ドコモのライバルである au では、通信方式として CDMA を利用しています。この技術の“元締め”である Qualcomm は、基地局および GPS の情報を併用する gpsOne という位置取得技術を AT&T 傘下の Lucent と共同で開発していました。その Qualcomm が 2000 年 1 月に SnapTrack を買収し、両者の技術を融合した新しい gpsOne 技術の開発を始めました。

この買収の結果、ドコモは au(当時は DDI/IDO)と関係の深い Qualcomm からライセンスを受けなければならぬことになってしまいました。そのためかどうかは分かりませんが、Naviewn は当時としては画期的なシステムだったにもかかわらず、販売にそれほど熱意は感じられませんでした(私の知るかぎり、当時はドコモショップでも実物を見ることはできませんでした)けっきょく、商品としてはほとんど知られることなく消えてしまいました。

ドコモは、携帯電話での位置情報利用そのものにも消極的になったようで、FOMA や 504 シリーズにも GPS や位置情報機能は付いていません。しかし、au の携帯電話の位置情報機能が好評を博したためか、将来的には GPS 機能付きの携帯電話を発売すると発表しています。

位置情報サービスの現状

au 携帯電話の eznavigation メニューには、レストラン、コミュニケーション、交通、天気、旅行といったカテゴリ別のサービスが数多く登録されており、GPS を利用したものもいろいろと提供されています。

これらのサービスは、下記のように大別できます。

- 現在位置をもとにした検索
- 現在位置を他人に通知

2 <http://www.snaptrack.com/>

3 <http://www.nttdocomo.co.jp/new/contents/99/whatnew0107b.html>

- 場所に応じて反応が変わるもの
- 行動の記録
- 位置を利用したゲーム

以下、それぞれの内容を簡単に紹介します。

現在位置をもとにした検索

Web 上には地図や時刻表、天気予報など、位置情報を活かせるサービスがたくさんあります。たとえば、自分の現在位置を検索キーとして与えれば、いっそう便利に使えます。

この種のサービスでもっとも需要が高いのは、周囲の地図を表示するものでしょう。近隣の駅の時刻表や道路の渋滞情報、地域ごとの天気予報なども重宝しそうです。変わったところでは、ゴルフ場でのピンまでの距離と方向を計算したり、気象情報を提供するものもあるようです⁴。

現在位置を他人に通知

自分の現在位置を他人に知らせるためのサービスもあります。

KDDI の「Team Factory」⁵は、位置情報をグループで共有できるグループウェアです。これを利用すれば、メンバーが近くにいるかどうか簡単に分かります。

セコムは、急病などになったときに自分の現在位置を通知する「ココセコム」⁶というサービスをおこなっています。緊急時に自分の現在位置を知らせる「ヘルプネット」⁷や、トラックなどの運行状況をモニターする「NIPS ナビ」⁸なども、位置情報の通知にもとづくサービスです。

場所に応じて反応が変わるもの

ユーザーの移動状況や現在位置によって挙動が変わるサービスもいくつか提供されています。

たとえば、「さちぐまラッキーナビ」⁹は、現在地の“風水”に応じて適当な画像を配信するサービスです。このほかに、現在位置に応じてメッセージが変化する「GPS 対応シーマン」などもあります。

4 http://www.doall.co.kr/jp/Product_gen_PDA.htm

5 <http://www.kddi.com/release/2002/0603/index.html>

6 http://www.855756.com/info/m_ez_top.html

7 <http://www.au.kddi.com/mobile/helpnet/>

8 <http://www.nihon-densan.co.jp/navi01.html>

9 http://keitai.channel.or.jp/contents/sachiguma_lucky_ez.html

行動の記録

GPS 携帯電話は、自分の行動を記録する場合も便利です。

訪れた場所と関連づけて日記が書ける、「Lycos あしあと日記」¹⁰ というサービスが提供されています。また、写真などと位置情報を関連づけて保存しておけば、記録としての有用性も高まるでしょう。

位置を利用したゲーム

宝探し、RPG、お絵描き、鬼ごっこ、マインスイーパー、フォックス・ハンティングなど、位置情報を利用したゲームもいろいろと提供されています。

考えられる位置サービス

位置情報を利用したサービスやゲームは、まだまだ大きな可能性をもっていると思われます。簡単に実現できそうなものをいくつか考えてみました。

位置に応じたコミュニケーション

とくに相手特定せずに、近くにいる人とコミュニケーションをとりたいという状況も考えられます。たとえば、高速道路でひどい渋滞に巻き込まれたときなど、周囲のドライバーと世間話ができれば情報収集やイライラの解消に役立つかもしれません。秋葉原の交差点の位置を ID とする掲示板を作り、PC のオークションをすることもできるでしょう。

特定の場所で特定の目的をもつ不特定多数の人と通信する場合、位置情報を利用する方式は便利だと思います。

ポインティング・デバイスとして使う

自分が動き回る「人間マウス」となって計算機を操作すれば、普通のアプリケーションや古いゲームも、位置情報を活用するゲームに生まれ変わります。たとえば通常のデスクトップ環境に人間マウスを導入した場合、和歌山でボタンを押すとメニューが表示され、大阪まで行ってボタンを離すとアプリケーションが起動されたりします。マウスを使って対戦するゲームに適用すれば、運動不足の解消になるかもしれません。

プログラミング

自分の現在位置や移動方向をプログラミング要素にすれば、移動によってプログラムを作成することも可能です。

¹⁰ <http://ashiato.lycos.co.jp/nikki/index.asp>

「秋葉原に行ったらメールが届く」というように、場所そのものを情報として利用してもいいでしょう。あるいは、山手線の各駅をアイコンと考えて移動し、ビジュアル・プログラミングをおこなうこともできるかもしれません。

CGI を用いた位置情報の取得

au の GPS 携帯電話では、CGI または Java プログラムで位置情報を取得することができます。事実、さきほど紹介したサービスのほとんどは、端末の位置情報を CGI で取得する仕組みになっているようです。

位置情報の取得方法としては、GPS を使わずに基地局の情報だけを利用する「簡易位置情報」によるものと、GPS と基地局の両方の情報を利用する gpsOne 機能によるものが用意されています。前者はカラーの cdmaOne 全機種で使えますが、後者をサポートしているのは GPS 対応の携帯電話(A3000/A5000 シリーズ)だけです。

CGI による簡易位置情報の取得

簡易位置情報の取得方法に関しては、KDDI から技術情報が公開されています¹¹。この機能を利用するには、CGI の URL を以下の書式で指定します。

```
device:location?url=CGIのURL
```

端末がこの URL にアクセスしようとする、位置情報を送出してもよいかを確認する画面が端末に表示されます。ユーザーが許可した場合、端末側では上記の URL に位置情報を引数として追加し、下記の形式の URL を呼び出します(誌面の都合上、⇒ で折り返しています。以下同様)

```
http://CGIのURL?datum=AAA&unit=BBB&⇒  
lat=XXX&lon=YYY
```

datum は測地系¹²を、lat と lon は緯度/経度を示しています。

測地系には、明治時代に測量法施行令第 2 条で定められた日本測地系と GPS 測量で用いられる世界測地系の 2 種

¹¹ <http://www.au.kddi.com/ezfactory/tec/spec/eznavi.html>

¹² 測量において、地球上の位置を緯度/経度で表すための基準となる座標系および地球を表す楕円体の形状を総称して測地系(測地基準系)といわれます。

図 1 簡易位置情報取得のための HTML の例

```
<a href="device:location?url=loc1.cgi">
  簡易位置情報取得</a>
```

図 2 簡易位置情報取得のための CGI (loc1.cgi)

```
#!/usr/bin/env perl

use CGI;

$q = new CGI;
$datum = $q->param('datum');
$unit = $q->param('unit');
$lat = $q->param('lat');
$lon = $q->param('lon');

print <<EOF;
Content-type: text/html

<html>
<head><title>loc</title></head>
<body>
$datum <br>
$unit <br>
$lat <br>
$lon
</body>
</html>
EOF
```

類があります。しかし、測地の基準となる回転楕円体の中心が異なるうえに、明治時代の測量技術の限界や地殻変動の影響などもあり、両者の違いはかなり大きくなってきています(世界測地系で示された緯度/経度を日本測地系の地図で使用すると、実際よりも 450m ほど北西にずれてしまいます) そのため、現在、世界測地系への移行が進められています¹³。

現行の地図サービスは、ほとんどが日本測地系にもとづいています。ところが、簡易位置情報取得 CGI では世界測地系にもとづいて緯度/経度を得るため、地図サービスなどで利用する場合は補正が必要になります¹⁴。

図 1~2 に、簡易位置情報取得機能を利用するための HTML と CGI の記述例を示します。これを使えば、次のような位置情報が取得できます。

```
tokyo
dms
35.40.37.13
139.45.40.23
```

13 <http://www.gsi.go.jp/LAW/G2000/g2000.htm>

14 2 つの測地系間の変換方法については、<http://vldb.gsi.go.jp/sokuchi/tky2jgd/>などを参照してください。

さきほど述べたように、簡易位置情報では GPS 信号は使わず、基地局の情報のみを用いているので、測地系を補正しても端末の位置との誤差が数百メートルになってしまうこともあります。

gpsOne による位置情報の取得

GPS 対応の携帯電話で gpsOne による位置情報機能を利用するには、以下のような形式で CGI を呼び出します (ver=1 の場合は世界測地系が、ver=0 の場合は日本測地系が使われます)。

```
<a href="device:gpsone?url=loc2.cgi&ver=0&datum=0&unit=0&acry=0&number=0">gpsOne位置情報取得</a>
```

このように指定して CGI を呼び出すことにより、GPS 信号の受信が可能な場所であれば、数メートル程度の誤差で現在位置を取得できます。gpsOne による位置情報取得は、簡易位置情報の取得に比較して時間がかかりますが、それでもたいていは数秒程度で正確な位置が得られます。

前述の位置ベースのサービスのほとんどは、gpsOne の機能を使って位置情報を得ているようです。

Java を用いた位置情報の取得

au は、2001 年から「ezplus」¹⁵ という Java 対応のサービスを提供しています。これは、携帯機器向けの Java の標準仕様 MIDP (Mobile Information Device Profile)¹⁶に、パイプレータやサウンドなど、携帯電話特有の「KDDI-P ライブラリ」を追加したものです。

ezplus では、2001 年に簡易位置情報取得などに対応した Phase 1 のサービスが開始されました。第 3 世代携帯電話ではこれが Phase 2 に進化し、gpsOne による位置情報の取得をはじめ、HTTP による通信、メール、携帯電話の内部データ参照などの機能が使えるようになっていきます。

Java の標準開発環境 (J2SE : Java 2 Standard Edition)¹⁷と MIDP 用開発環境 (Java 2 Micro Edition,

15 <http://www.au.kddi.com/ezweb/ezplus/>
<http://www.au.kddi.com/ezfactory/tec/spec/ezplus.html>

16 <http://java.sun.com/products/midp/>

17 <http://java.sun.com/j2se/>

携帯電話と Java2

Java2 プラットホームは、サーバー環境用の J2EE (Java2 Enterprise Edition)、デスクトップ環境用の J2SE (Java2 Standard Edition)、組み込み機器用の J2ME (Java2 Micro Edition) の 3 種類の “Edition” に大別されます。

J2ME では、ネットワーク機器用の基本ライブラリ仕様である CDC (Connected Device Configuration) と、携帯機器用基本ライブラリ仕様である CLDC (Connected Limited Device Configuration) の 2 つの “Configuration” が定義されています。また、各端末の細かい仕様を定義するため、CLDC ではさらに端末の種類ごとに異なる “Profile” を使うようになっており、携帯電話では MIDP (Mobile Information Device Profile) が使われています。

それぞれの詳細は、以下の URL を参照してください。

- CLDC : <http://java.sun.com/products/cldc/>
- MIDP : <http://java.sun.com/products/midp/>

Wireless Toolkit¹⁸は Sun から、KDDI-P のライブラリやドキュメント、エミュレータなどは KDDI から提供されています¹⁹。このように、品質の高い開発環境がすべて無償で提供されており、プログラムの公開やダウンロードも自由におこなえるようになっています。

これらの開発ツールを使えば、gpsOne を用いた高精度な位置情報を利用する第 3 世代携帯電話用の Java アプリケーションを誰でも容易に開発できます。

携帯 Java のプログラム開発の手順

ezplus アプリケーションの開発から公開までは、以下の手順でおこないます。

1. MIDP、KDDI-P ライブラリを使って Java プログラムをコンパイルする。
2. 事前検証ツール preverify を使ってクラスファイルの形式を変換する。

18 Sun が無償で公開している開発環境で、CLDC/MIDP の開発ツール、サンプル・アプリケーション、エミュレータなどが含まれています。現在、Windows、Solaris、Linux 版があります(執筆時の最新版は、英語版が 1.0.4、日本語版が 1.0.3 です)

<http://java.sun.com/products/j2mewtoolkit/>

19 この開発環境はゼンテックが開発したもので、ezplus エミュレータと KJX 作成ツール、サンプル・アプリケーション、ビルドのための GUI などが含まれています。

<http://www.zentek.com/>

<http://www.au.kddi.com/ezfactory/tool/index.html>

<http://www.au.kddi.com/ezfactory/tec/spec/>

[ezplus_kit.html](http://www.au.kddi.com/ezfactory/tec/spec/ezplus_kit.html)

一般の Java プログラムでは、実行時にクラスの検証をおこないます。しかし、資源の乏しい携帯端末では困難なので、クラスファイルを検証しやすい形式に変換してから端末にダウンロードします。preverify は、このようなクラスファイルの変換をおこなうツールです²⁰。

3. クラスファイルとリソースをまとめて jar ファイルを作成する。
4. jar ファイルを KJX 形式(ezplus のダウンロード形式)に変換する。
5. KJX 形式のファイルに CRC チェックサムを付加する。
6. Web 上にダウンロード用 CGI と KJX ファイルを配置する。

1~3 は MIDP のプログラム開発に共通で、Sun の提供する開発環境だけで実行できます。

4 の実行には KDDI の提供する KJX 作成ツールが必要です。ezplus エミュレータを使うことにより、作成された KJX ファイルを Windows 上で実行することができます。

作成されたプログラムを Web 上で公開するには、専用ツールで KJX ファイルに CRC チェックサムを付加し、ダウンロード用の HDML と CGI を使う必要があります。ダウンロード CGI は KDDI から提供されています。

gpsOne 情報取得プログラム

末尾のリスト 1 に、gpsOne 情報を読み出す ezplus プログラムの例を示します。位置情報を取得するための API は au の資料で公開されています。GpsLocation.getLocation() により端末は gpsOne による位置情報取得を開始し、getLat() や getLon() などのメソッドで結果を取得します。

KDDI の提供する GUI 環境による開発

KDDI の提供する ezplus 開発キットに付属する GUI ベースの KJX 作成ツール “CmdTool” を利用すれば、上記の手順の 1~4 を実行することができます。

以下のようにして CmdTool を起動すると図 3 のウィンドウが現れるので、この上で開発を進めていきます。

```
# java -jar c:/ezplusTools/Tools/CmdTool/⇒  
CmdTool.jar
```

20 http://java.sun.com/products/cldc/docs/CLDC1.0_ja_docs.zip

図 3 CmdTool (1)

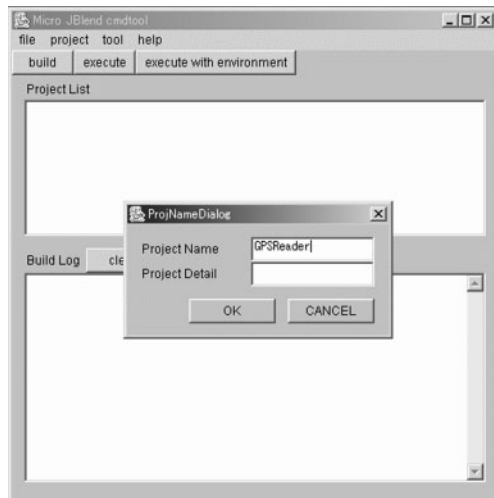


図 4 CmdTool (2)

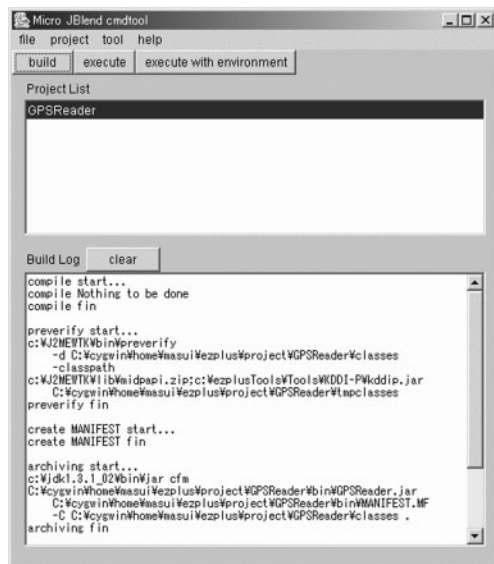
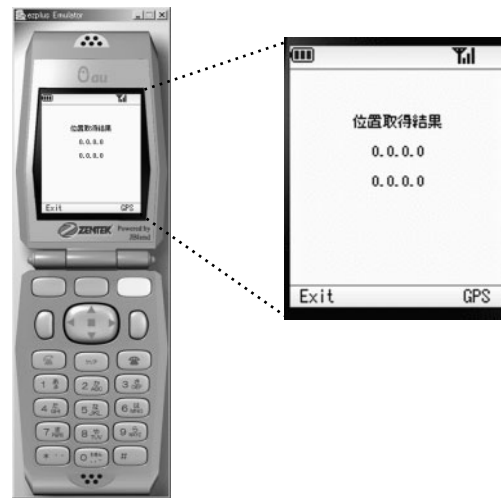


図 4 は、GPSReader.java をコンパイルして KJX ファイルを生成しているところです。この結果、GPS-Reader.kjx が作成され、エミュレータ上で実行できるようになります(図 5)。エミュレータ上での実行なので、この図では緯度と経度がゼロになってしまっていますが、実際に携帯電話で実行すると正しい値が表示されます。

Makefile を用いた開発

CmdTool を使って開発する場合には、ソースなどを特定のディレクトリに置かなければならないといった制約が

図 5 エミュレータ上での実行結果の表示



あってちょっと不便です。しかし、GUI の CmdTool を使わなくても、テキストベースのツールをいくつか用意することにより、Makefile を用いてすべての開発プロセスを制御することができます。

リスト 2 は、Cygwin 環境で前述のツールを使い、ez-plus プログラムの開発のすべての工程を指示する Makefile です。

javac や preverify などは、Sun が提供するコマンドをそのまま利用します。jar ファイルの作成に必要な Manifest ファイルは、makemanifest プログラム(リスト 3)で作ります。この Manifest ファイルを、KJX ファイルの作成に必要な jad ファイル²¹に変換するには manifest2jad プログラム(リスト 4)を使います。

Java プログラムの公開とダウンロード

完成した KJX ファイルを携帯電話からダウンロードできるようにするには、以下の処理が必要です。

1. ダウンロードのための特殊な CGI を用意する。
2. KJX ファイルに CRC チェックサムを付加する。
3. ダウンロードのための HDML ファイルを用意する。

1 については、KDDI が提供しているもの²²をそのまま使えばいいのですが、2~3 の作業はユーザーがおこな

²¹ jar ファイルのサイズやファイル名を指定するためのファイルです。

²² http://www.au.kddi.com/ezfactory/tec/dl/cgi/download_1.html

表 1 ezplus や MIDP のプログラミングの参考になる Web ページ

ezplus 公式ページ	http://www.au.kddi.com/ezweb/ezplus/
sinsen.org(服部隆志氏)	http://www.sinsen.org/midp/
	http://www.sinsen.org/midp/ezplus/
@IT の記事(服部隆志氏)	http://www.atmarkit.co.jp/fmobile/kaisetsu/ezplus/ezplus.html
ezplus ことはじめ(綿 100 号氏)	http://homepage1.nifty.com/cotton100/ezplus.htm
ezplus アプリ開発(さかきけい氏)	http://member.nifty.ne.jp/violet/mkezplus.html
MIDP の API の解説(さかきけい氏)	http://godwood.allnet.ne.jp/violet/midpapi/
Micro Java Network(MIDP 情報)	http://www.microjava.com/
J2ME CLDC/KVM Palm チュートリアル (Palm 用の CLDC を使ったプログラム開発)	http://fujisawa.org/palm/tutorial.html

う必要があります。

リスト 2 の Makefile では、addcrc というプログラム (リスト 5) で CRC を KJX ファイルに付加し、downloadhtml というプログラム(リスト 6) でダウンロードのための HDML ファイルを作成しています。

おわりに

今回は、GPS 携帯電話を使った位置情報の活用について書こうと思ったのですが、位置情報取得のプログラムを書き始めた段階で時間切れになってしまいました。GPS 情報を活用するプログラムについては、機会をあらためて紹介したいと思います。Java アプリケーションの開発に使ったプログラムとサンプル・プログラムは、私の Web ページで公開しています²³。

²³ <http://www.csl.sony.co.jp/person/masui/ezplus/ezplustools.tar.gz>

MIDP 開発ツールは Windows と Linux、Solaris で使えますが、KJX 作成ツールは Windows 版しかないのので、現時点では ezplus プログラムは Windows 上でしか開発できません。しかし、Cygwin 環境や Emacs を使えば、UNIX とほぼ同じ感覚で開発作業をすることができます。ezplus や MIDP のプログラミングに関しては、表 1 のページが参考になります。

今回説明したように、ezplus の開発環境はかなり整備されており、位置情報の取得や HTTP 通信など、数多くの API が用意されています。今後、Java を用いた便利なアプリケーションが続々と開発されることを期待しています。

(ますい・としゆき ソニー CSL)

リスト 1 gpsOne を用いた位置情報取得プログラム (GPSReader.java)

```
import javax.microedition.midlet.*;
import javax.microedition.lcdui.*;

import com.kddi.system.*;

public class GPSReader extends MIDlet implements CommandListener
{
    GPSCanvas gpsCanvas;

    Command exitCommand, gpsCommand;
    Display display;

    public GPSReader()
    {
        display = Display.getDisplay(this);
        exitCommand = new Command("Exit",Command.EXIT,1);
        gpsCommand = new Command("GPS",Command.SCREEN,1);
    }
}
```



```

        gpsCanvas = new GPSCanvas();
        gpsCanvas.setCommandListener(this);
        gpsCanvas.addCommand(exitCommand);
        gpsCanvas.addCommand(gpsCommand);
        display.setCurrent(gpsCanvas);
    }

    public void startApp(){ }
    public void pauseApp(){ }
    public void destroyApp(boolean unconditional){ }

    public void commandAction(Command command,Displayable screen)
    {
        if(command==exitCommand){
            destroyApp(false);
            notifyDestroyed();
        }
        if(command==gpsCommand){
            gpsCanvas.showLocation();
        }
    }

    public class GPSCanvas extends Canvas
    {
        Location location;
        String latitude = "", longitude = "";
        Font font;
        int width, height;

        public GPSCanvas()
        {
            font = Font.getFont(Font.FACE_MONOSPACE,Font.STYLE_PLAIN,Font.SIZE_SMALL);
            width = getWidth();
            height = getHeight();
        }

        public void showLocation()
        {
            location = GpsLocation.getLocation();
            latitude = location.getLat();
            longitude = location.getLon();
            repaint();
        }

        public void paint(Graphics g) {
            g.setColor(255,255,200);
            g.fillRect(0,0,width,height);
            g.setColor(0,0,0);
            g.setFont(font);
            g.drawString("位置情報取得結果",width/2,30,Graphics.HCENTER | Graphics.VCENTER );
            g.drawString(latitude,width/2,50,Graphics.HCENTER | Graphics.VCENTER );
            g.drawString(longitude,width/2,70,Graphics.HCENTER | Graphics.VCENTER );
        }
    }
}

```

リスト 2 KJX を作成するための Makefile

```
#
# アプリケーション名/ベンダー名
#
TARGET = GPSReader
VENDOR = Masui

#
# 標準Java開発環境 (from Sun)
# http://java.sun.com/j2se/
#
JDK2 = c:/jdk1.3.1_02
JAVAC = $(JDK2)/bin/javac
JAVA = $(JDK2)/bin/java
JAR = $(JDK2)/bin/jar

#
# Java Micro Edition Wireless Toolkit (from Sun)
# http://java.sun.com/products/j2mewtoolkit/
#
J2MEWTK = c:/J2mewtk # 英語版の場合は c:/WTK104 に変更
PREVERIFY = $(J2MEWTK)/bin/preverify
MIDPLIB = $(J2MEWTK)/lib/midpapi.zip

#
# ezplus用開発環境 (from KDDI)
# http://www.au.kddi.com/ezfactory/tool/
#
EZPLUSTOOLS = c:/ezplusTools/Tools
KJXARCHIVER = $(EZPLUSTOOLS)/CmdTool/KJXArchiver.jar
KDDILIB = $(EZPLUSTOOLS)/KDDI-P/kddip.jar

EZCLASSPATH = '$(MIDPLIB);$(KDDILIB)'

#
# 補助ツール (by masui)
# http://www.csl.sony.co.jp/person/masui/ezplus/
#
EZTOOLDIR = /usr/local/bin
MAKEMANIFEST = $(EZTOOLDIR)/makemanifest
MANIFEST2JAD = $(EZTOOLDIR)/manifest2jad
ADDCRC = $(EZTOOLDIR)/addcrc
DOWNLOADHML = $(EZTOOLDIR)/downloadhml

all: bin classes tmpclasses
    $(JAVAC) -encoding SJIS -g:none \
        -d tmpclasses -bootclasspath $(EZCLASSPATH) \
        -classpath tmpclasses src/$(TARGET).java
    $(PREVERIFY) -d classes -classpath $(EZCLASSPATH) tmpclasses
    $(MAKEMANIFEST) $(TARGET) $(VENDOR) > bin/MANIFEST.MF
    $(JAR) cfm bin/$(TARGET).jar bin/MANIFEST.MF -C classes .
    $(JAR) ufm bin/$(TARGET).jar bin/MANIFEST.MF -C res .
    cd bin; $(MANIFEST2JAD) $(TARGET); cd ..
    $(JAVA) -jar $(KJXARCHIVER) -c bin/$(TARGET).jad \
        bin/$(TARGET).jar bin/$(TARGET).kjsx
```

```

# ここまでがKJX作成ツールと同じ処理
mv bin/$(TARGET).kjax bin/$(TARGET)_nocrc.kjax
$(ADDCRC) bin/$(TARGET)_nocrc.kjax bin/$(TARGET).kjax
cd bin; $(DOWNLOADHDM) $(TARGET) > download.html ; cd ..

bin classes tmpclasses:
    -mkdir bin classes tmpclasses
clean:
    -\rm -r -f bin classes tmpclasses *~ src/*~

cmdtool:
    $(JAVA) -jar $(EZPLUSTOOLS)/CmdTool/CmdTool.jar
emulator:
    cd $(EZPLUSTOOLS)/Emulator; $(JAVA) -jar jbmip.jar

#####

WEBDEST = www.csl.sony.co.jp:/user/masui/masui/ez
web:
    scp bin/$(TARGET).kjax bin/download.html $(WEBDEST)

TOOLDEST = www.csl.sony.co.jp:/user/masui/masui/ezplus
dist: clean
    cd ..; tar cvf - GPSReader makemanifest manifest2jad addcrc downloadhtml \
        | gzip > /tmp/ezplustools.tar.gz
    scp /tmp/ezplustools.tar.gz $(TOOLDEST)

```

リスト 3 makemanifest

```

#!/usr/bin/env perl

$target = shift;
$vendor = shift;
exit if !$target || $vendor;

print <<EOF;
MIDlet-Name: $target
MIDlet-Vendor: $vendor
MIDlet-Version: 1.0
MicroEdition-Configuration: CLDC-1.0
MicroEdition-Profile: MIDP-1.0
MIDlet-1: $target,,$target
EOF

```

リスト 4 manifest2jad

```

#!/usr/bin/env perl

$target = shift;

exit unless $target;

$jarfile = $target . '.jar';
$jadfile = $target . '.jad';
$manifestfile = 'MANIFEST.MF';

```

```

$jarsize = (stat($jarfile))[7];

open(out,"> $jadfile");
print out "MIDlet-Jar-Size: $jarsize\n";
print out "MIDlet-Jar-URL: $jarfile\n";
open(in,$manifestfile);
while(<in>){
    print out $_;
}
close(in);
close(out);

```

リスト 5 KJX ファイルに CRC を付加するプログラム (addcrc)

```

#!/usr/bin/env perl

$oldfile = shift;
$newfile = shift;
exit if !$oldfile || !$newfile;

@crc_tbl = (
    0x0000,0x1021,0x2042,0x3063,0x4084,0x50A5,0x60C6,0x70E7,
    0x8108,0x9129,0xA14A,0xB16B,0xC18C,0xD1AD,0xE1CE,0xF1EF,
    0x1231,0x0210,0x3273,0x2252,0x52B5,0x4294,0x72F7,0x62D6,
    0x9339,0x8318,0xB37B,0xA35A,0xD3BD,0xC39C,0xF3FF,0xE3DE,
    0x2462,0x3443,0x0420,0x1401,0x64E6,0x74C7,0x44A4,0x5485,
    0xA56A,0xB54B,0x8528,0x9509,0xE5EE,0xF5CF,0xC5AC,0xD58D,
    0x3653,0x2672,0x1611,0x0630,0x76D7,0x66F6,0x5695,0x46B4,
    0xB75B,0xA77A,0x9719,0x8738,0xF7DF,0xE7FE,0xD79D,0xC7BC,
    0x48C4,0x58E5,0x6886,0x78A7,0x0840,0x1861,0x2802,0x3823,
    0xC9CC,0xD9ED,0xE98E,0xF9AF,0x8948,0x9969,0xA90A,0xB92B,
    0x5AF5,0x4AD4,0x7AB7,0x6A96,0x1A71,0x0A50,0x3A33,0x2A12,
    0xDBFD,0xCBDC,0xFBBF,0xEB9E,0x9B79,0x8B58,0xBB3B,0xAB1A,
    0x6CA6,0x7C87,0x4CE4,0x5CC5,0x2C22,0x3C03,0x0C60,0x1C41,
    0xEDAE,0xFD8F,0xCDEC,0xDDCD,0xAD2A,0xBD0B,0x8D68,0x9D49,
    0x7E97,0x6EB6,0x5ED5,0x4EF4,0x3E13,0x2E32,0x1E51,0x0E70,
    0xFF9F,0xEFBE,0xDFDD,0xCFFC,0xBF1B,0xAF3A,0x9F59,0x8F78,
    0x9188,0x81A9,0xB1CA,0xA1EB,0xD10C,0xC12D,0xF14E,0xE16F,
    0x1080,0x00A1,0x30C2,0x20E3,0x5004,0x4025,0x7046,0x6067,
    0x83B9,0x9398,0xA3FB,0xB3DA,0xC33D,0xD31C,0xE37F,0xF35E,
    0x02B1,0x1290,0x22F3,0x32D2,0x4235,0x5214,0x6277,0x7256,
    0xB5EA,0xA5CB,0x95A8,0x8589,0xF56E,0xE54F,0xD52C,0xC50D,
    0x34E2,0x24C3,0x14A0,0x0481,0x7466,0x6447,0x5424,0x4405,
    0xA7DB,0xB7FA,0x8799,0x97B8,0xE75F,0xF77E,0xC71D,0xD73C,
    0x26D3,0x36F2,0x0691,0x16B0,0x6657,0x7676,0x4615,0x5634,
    0xD94C,0xC96D,0xF90E,0xE92F,0x99C8,0x89E9,0xB98A,0xA9AB,
    0x5844,0x4865,0x7806,0x6827,0x18C0,0x08E1,0x3882,0x28A3,
    0xCB7D,0xDB5C,0xEB3F,0xFB1E,0x8BF9,0x9BD8,0xABBB,0xBB9A,
    0x4A75,0x5A54,0x6A37,0x7A16,0x0AF1,0x1AD0,0x2AB3,0x3A92,
    0xFD2E,0xED0F,0xDD6C,0xCD4D,0xBDAA,0xAD8B,0x9DE8,0x8DC9,
    0x7C26,0x6C07,0x5C64,0x4C45,0x3CA2,0x2C83,0x1CE0,0x0CC1,
    0xEF1F,0xFF3E,0xCF5D,0xDF7C,0xAF9B,0xBFBA,0x8FD9,0x9FF8,
    0x6E17,0x7E36,0x4E55,0x5E74,0x2E93,0x3EB2,0x0ED1,0x1EF0
);

```

```

$crc = 0xFFFF;

open(out,"> $newfile");
open(in,$oldfile);
while(($c = getc(in)) ne ''){
    print out $c;
    $crc = $crc_tbl[($crc>>8 ^ ord($c)) & 0xff] ^ (($crc<<8) & 0xffff);
#    print ord($c,"\n");
}
close(in);

#print out ($crc >> 8) & 0xff;
#print out "\n";
#print out $crc & 0xff;
#print out "\n";

print out pack('C',($crc >> 8) & 0xff);
print out pack('C',$crc & 0xff);

close(out);

```

リスト 6 ダウンロードのための HDML ファイルを作成するプログラム (downloadhtml)

```

#!/usr/bin/env perl

exit if $#ARGV < 0;

print <<EOF;
<html version="3.0" markable="true">
<choice key="url" title="ezplus">
<action type="accept" task="gosub" dest="device:data/dnld?url=\$url" label="OK">
<br>
<center>ezplus download
EOF

foreach $target (@ARGV){
    $kjsxfile = "$target.kjx";
    $kjsxsize = (stat($kjsxfile))[7];
    print <<EOF;
<ce value="download.cgi&name=$target.kjx&size=$kjsxsize&disposition=devkdjx&title=$target">
    [$target]
EOF
}

print <<EOF;
</choice>
</html>
EOF

```