

# ネット経由でモーション変更可能な二足歩行ロボットの開発



## 第1回 SPC-101Cの概要とコマンド・サーボ

以前、本誌(2005年5月号)で解説したスピーシーズの二足歩行ロボットが、ネットワークに対応し、低価格になって再登場した。新機種のロボットSPC-101CはNetBSDベースのOSを搭載し、マイコンを内蔵したコマンド・サーボを搭載するなど、従来の機種と比較して高機能なのが特徴だ。本連載では、SPC-101Cを題材に、ネットワークに対応したロボットのアプリケーション開発について解説する。今回は、SPC-101Cの概要と、本ロボットの技術的な鍵であるコマンド・サーボについて説明する。

(編集部)

### 1 ネットワーク制御が可能なロボット SPC-101C



SPC-101Cは、2006年4月にスピーシーズが発表した二足歩行の人型エンターテインメント・ロボットです(写真1)。汎用性の高いUNIX系のNetBSDをベースとしたOSを内蔵しています。無線LANを通じてネットワークに常時接続することで、サーバやパソコンから自在に操作可能な「インターネット・ロボット」です。

2007年の秋からは、カラー・カメラを内蔵し、録画・再生機能や画像認識、音声認識によるボイス・コマンドによる操作も行えるようになりました。

#### 目的はロボット・ソフト開発を容易にすること

スピーシーズは、ロボットのソフトウェア開発を目指す人が便利のように、いろいろなサンプル・ソフトウェアを提供しています。本ロボット用に開発した制御用プロセッサ・ボード「RPU-50(写真2)」のOSにはNetBSDを搭載しており、NetBSDのさまざまな機能がパソコン上のNetBSDと全く同じように利用できます。そして、サンプル・

アプリケーションを数種類提供し、開発開始の際のハードルを引き下げ、ロボットの高度なプログラム開発を容易に行えるようにしました。

これだけそろっていれば、「ロボットで何が可能なのか」をイメージしやすく、ロボットのソフトウェア開発にかなり容易に取り組めるのではないのでしょうか。

今回発売したカメラ搭載インターネット・ロボットは、ソフトウェア開発のプロフェッショナルはもちろん、日曜プログラマにもぜひ試してもらいたいロボット・ソフトウェアの開発環境です。

#### 既存の人型ロボットとの違い

本ロボットの一番の特徴は、UNIXベースのOSを搭載していることです。これにより、ネットワーク機能を標準で利用できます。ネットワークを通してインターネット上のサーバやLAN内のパソコンと通信できるということは、ロボットの機能や性能が事実上無限に広がるということを意味します。

筆者らは、ロボット内のプロセッサだけですべての処理を実行しなければならないとは考えていません。ロボットに搭載しているプロセッサでは、サーボ動作のためのタイ



写真1 SPC-101Cの外観



写真2 制御用プロセッサ・ボードRPU-50

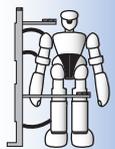


表1 SPC-101C ネットワークの接続形態

socket	SOAP
制御指向	サービス指向
ロボットがサーバ	インターネット上のサーバに接続可能(ロボットはクライアント)
利点: ● 細やかな制御が可能 ● 処理が軽いので、応答も早い ● 階層化されており、必要な部分を抜き取り可能(現在開発中) 欠点: ● パケットを編集しなければならない ● 一部、ライブラリ化予定	利点: ● HTTP でラッピングされた RPC ● Web ブラウジングを行える環境があれば、インターネット上のサーバに接続可能 ● WSDL によりインタフェースが定義されるため、開発、拡張、変更が容易 欠点: ● socket と比べるとオーバーヘッド大

ミング制御が最も重要です。また、サイズや価格、消費電力などの制約により、パソコンと比較すると大きく性能が劣ります。そのため、高性能な外部のコンピュータでより高度な処理を行い、その結果を(ネットワーク上の出力端末としての)ロボットに転送して動作させるという考え方に基づいて開発しています。

本ロボットをネットワークに接続するに当たり、UNIX ソケットと XML ベースの SOAP による接続が利用可能です(表1)。これら標準的な技術の採用により、プログラミングの敷居を引き下げています。

本ロボットでは頭部に CMOS カラー・カメラを搭載していますが、この画像をパソコン側で受け取り、画像処理してその結果を本ロボットに出力するという構成です。

また、本ロボットは組み立てキットではなく完成品です。ソフトウェア開発者からすると、購入してすぐに動かしたい場合が多く、キットの組み立て作業は煩わしいのではないのでしょうか。本ロボットは、入手後、一部の設定を行えばすぐに動きます。

### ロボット用インターフェース OpenRoads

既存のロボット(SPC-001 ~ 003)では搭載している制御用プロセッサ・ボード(RPU-100)が十分な処理能力を持っていたため、ロボット内で動作させることを主眼とした開発キットを提供してきました(詳細は本誌2005年5月号, pp.138-145を参照)。RPU-100ではNetBSD/evbppcのクロス開発となるのですが、以下の点から、開発環境の構築に苦労した方が多かったようです。

- クロス開発環境を特に用意せず、NetBSDのソースから、  
<http://www.netbsd.org/docs/guide/en/chap-build.html>  
を参考にユーザがビルドを行っていた
- 機械工学を専門とするユーザが多く、クロス開発その

ものに慣れていない場合が多かった

これらを改善するため、OpenRoads(Open Robot Operation and Design Specification)としてネットワーク・パケットの仕様を公開し、クロス開発なしで周辺アプリケーションを作成できるようにしました。

OpenRoadsは、一部SPC-101シリーズのみ有効な部分がありますが、SPC-001 ~ 003でも大部分は同じです(SPC-001 ~ 003で使用できないのは、サーボ・ステータス読み込み要求のみ)。

### RPU-50上で動作するアプリの作成も可能

本ロボットの制御用プロセッサ・ボードのOSはNetBSDであり、もちろん本ボード上で動作するアプリケーション

#### コラム1 クロス開発への対応について

クロス開発アプリケーションのフラッシュROMへの書き込みや自動起動については、現在ユーザに開放していません。現時点では、手動で起動する必要があります。ゆくゆくは自動起動に対応しようと考えていますが、対応時期は未定です。

最終的にはクロス開発アプリケーション作成を目的としていても、最初の段階ではパソコン上のNetBSDを使用することも可能です。この方法は、デバッグ作業が非常に楽であることが大きな利点です。dddなどの普段使用しているグラフィカル・デバッガを使用して、機能レベルのデバッグを完了させてから、最終目的であるターゲット環境への移行を行います。

詳細は次回以降に解説する予定ですが、環境変数を2種類設定してmakeコマンドを変更すれば、それでRPU-50アプリケーションになってしまいます。後は、タイミングやRPU-50上のデバイスなどの動作確認を行えば、ターゲット・アプリケーションが完成します。