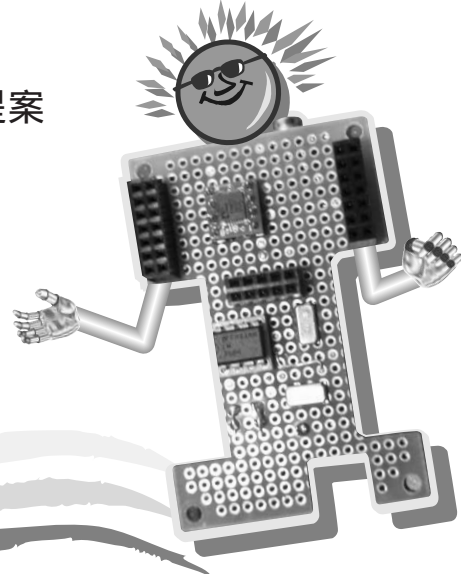


オープン・ソースの ロボット用プラットフォーム RoboShellシステム

大橋 修



ロボット開発がブームだが...

近年、日本をはじめアジアでも、企業や研究団体、アマチュアを問わずロボット開発がブームです。とくに2足歩行ロボットは、RoboOneなどの競技会やキットの販売を通して、下は小学生から上はプロフェッショナルまで、非常に幅広く開発されています。ロボットのキットも、安いものから企業の研究プラットフォームになりえるものまで、いろいろなものが販売されています。また、制御用のコントローラやサーボ・モータも単体で発売されています。

しかしロボットは、それぞれがさまざまな技術の集合体です。ちょっとまともなロボットを作ろうとすると、すぐに障壁が現れます。競技用の2足歩行ロボットではサーボ・モータが多用されますが、それらは非常に高価です。競技ロボット界では10,000円を「1サーボ」と呼ぶくらいですが、競技に使う高性能なサーボ・モータは1個10,000円では購入できません。当然、これらを制御するサーボ・モータ・コントローラも必要になります。かなり高価(40,000円程度)なうえに、コピーされることを恐れてか、開発元からの情報開示が進んでいません。開発環境も専用のものが多く、同じ系列のボードを使わない限り、習得した技術を次へ生かすことが難しくなります。

もちろん、これらのボードには開発費がかかっています。たいした数が出るわけではなく、量産効果による低価格化も期待できないため、このような価格帯になることは理解できます。しかし、価格が手ごろでないとボードは売れません。腕に覚えのあるユーザは、自分自身で各社から発売されている評価基板やキットなどのマイコン・ボードを購入して、サーボ・モータを制御できるボードを開発することになります。

こうして自作されたこれらのサーボ・モータ・コントローラも、競技会への参加を前提にしているユーザが開発していることが多いため、情報開示はあまり期待できません。また、同じマイコン・ボードを使っても構成は少しずつ違うので、まったく同一のソフトウェアを動作させることは難しくなります。さらに、あまりにも専用化された開発ツールも気になります。

そうなると技術の蓄積が進まず、昨今の開発に重要な概念である標準化もできなくなり、いつまでたっても「車輪の再開発」の状態から抜け出すことができません。

RoboShellシステムを開発

このような状況を打開するため、筆者はハードウェアなどの回路図も含めたオープン・ソースのロボット用プラットフォームとして、「RoboShellシステム」を開発しました。

RoboShellシステムとは、ロボット・コントローラ用サーボ・モータEZ-SERVO(オプティマイズ製)と、EZ-SERVO上で動作するロボット制御用ソフトウェアのフレームワークRoboShellから構成されます。RoboShellは筆者の所属するTeamKNOxが開発・配布を行っています。

RoboShellシステムの目的は、業界標準の技術を用いてロボット制御に利用できる低価格でオープンなサーボ・モータ・コントロールを構築することにあります。さらにその製作過程を通して、ロボット制御に必要な知識(制御理論や電子回路、ユーザ・インターフェースなど)を学びます。また、得られた技術をコミュニティで共有して技術の向上に役立てることにあります。

RoboShellシステムの基本構成

図1にRoboShellシステム全体の基本構成を示します。RoboShellシステムといった場合は、ハードウェア(現在の参照実装はEZ-SERVO)とソフトウェア(RoboShell)で構成されます。miniEZ-USBを介してパソコンとUSBで接続し、プログラムの開発とデバッグに利用します。フロントエンドでは、開発環境にEclipseを使います。シリアルも装備されていて、このシリアルはモーション作成ソフトウェアであるRoboShell Composerに利用されます。また、ほかのホスト用のマイコンを用いてコマンドを投入することも可能です。

RoboShellシステムの特徴

ほかのサーボ・モータ・コントローラと比べると、RoboShellシステムには次のような特徴があります。

(1) 高機能

- 32ビットRISC CPU搭載
- 最大32チャンネルのPWM方式のサーボ・モータ・コントローラ搭載
- 512Kバイトと大容量のオンチップ・フラッシュ・メモリ搭載
- 60MHzの内部ブロックによる高速動作が可能

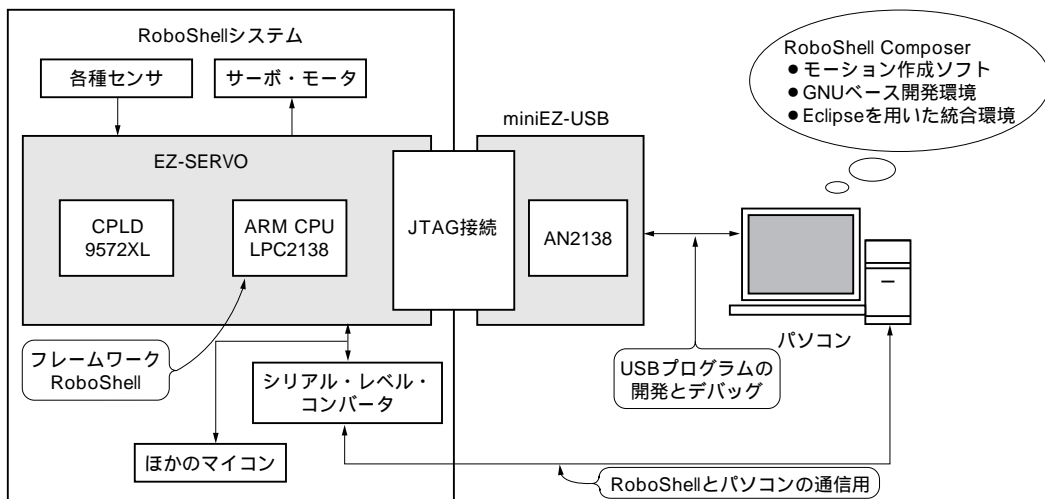


図1 RoboShell システム基本構成

(2) 業界標準技術の導入

ハードウェアとしては、次のデバイスやインターフェースを採用しています。

- ARM7TDMI コア CPU LPC2138(NXP Semiconductors 社)
 - CPLD XC9572XL(Xilinx 社)
 - USB
 - JTAG
 - PWM 方式のサーボ・モータ・コントローラ
- 〔宮田耕二氏によって発表された Little Burning Core(以下 LBC)をベースにして、近藤科学製のレッド・バージョンのサーボ・モータに一部対応した拡張版〕
- 開発環境には次のものを採用しています
- GCC や GDB などの GNU ツール・チェーン
 - Eclipse + CDT
 - CPLD Writer
 - Flash Programming ツール

以上の組み合わせで、USB を介してパソコン(PC)と接続し、JTAG によるプログラムの書き込みとデバッグが可能です。写真1にCPU + CPLD 基板であるEZ-SERVOと、JTAG デバッグを実現するUSB 基板 miniEZ-USB(オプティマイズ製)を接続した状態を示します。これらは広く使われている技術で構築されており、習得した技術がほかの用途にも使えるなど、将来の資産になります。

(3) 柔軟性(CPU + CPLD)

RoboShell システムでは、サーボ・モータを最大 32 個接続できます。このようにサーボ・モータの数が増えれば、CPU の I/O ポートの数もそれなりに必要になります。しかし、そうするとポートの多いCPUが必要になってきます。ポートの多いCPUは種類が限られてくるので、選択の幅が狭くなります。

RoboShell ではCPLD を利用することにより、この問題に対応しています。CPLD はユーザ側でプログラミングが可能なので、現在の制御方法からシリアル通信方式などに変更すること

もできます。

(4) 小型

サイズは72mm x 47mm と小型です。これは電子工作を行うユーザにはおなじみの、秋月電子通商のC基板と同サイズになります。穴位置が合わせてあるため、拡張も容易です。

(5) 低価格

EZ-SERVO のキットは5,000 円、完成品は8,000 円と低価格です。ロボットに使えるサーボ・モータ・コントローラを入手するには、市販品を購入するか、各種マイコンの評価用ボードを独自に拡張して利用するのは先に述べたとおりです。市販品の価格に対してとやかく言うつもりはありませんが、ユーザから見ると高価な気がします。評価用ボードはそれなりに安価ですが、サーボ・モータを動かすところまで拡張するにはそれなりに手間とお金がかかります。

(6) ネイティブ対応

逆運動理論を実装するためにはネイティブ環境が必須です。

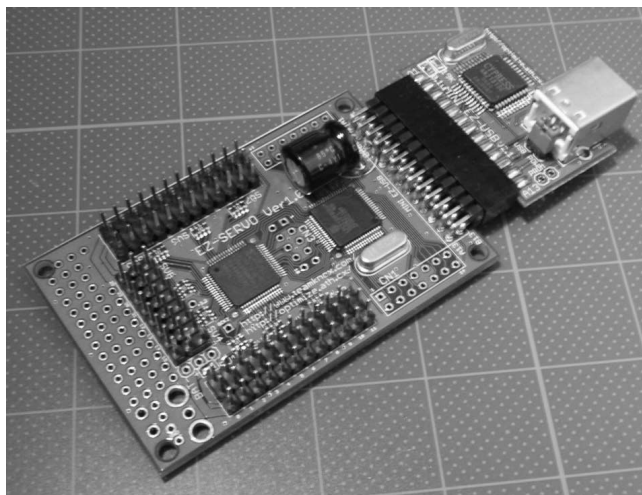


写真1 EZ-SERVO + mini EZ-USB の外観