

自立ロボットシステムのための USB 接続による 小型モータコントローラの開発

光永 法明, 細田 耕, 浅田 稔
大阪大学大学院 工学研究科 知能機能創成工学専攻

Small USB motor controller for self contained robot systems

Noriaki Mitsunaga, Koh Hosoda and Minoru Asada
Dept. of Adaptive Machine Systems, Graduate School of Engineering Osaka Univ.

Abstract— This paper presents a motor controller connected via USB (Universal Serial Bus). The small size of USB motor controller and a note-book PC enable the small robot systems. This paper briefly describes implementation of the controller.

Key Words: USB Interface, Motor Controller, PID Control

1. はじめに

ロボットシステムのためのメインコントローラとして、ノート PC を利用することにより、小型軽量の自立ロボットシステムが実現できると考えられる。ところで、最近ではほとんどの PC に USB (Universal Serial Bus) が標準装備されるようになっており、ノート PC も例外ではない。USB はシリアルバスで、細い配線で機器を接続する事ができ、ISA バス以上の転送速度 (USB 1.1 では 12Mbps) を持つ。また比較的デバイスの設計製作が容易である、他の市販の機器と混在して使用する事が可能、最大デバイス数は 127 台、活線挿抜が可能といった特徴がある。ただし USB ホスト (PC) なしの USB デバイス間の通信は出来ず、ホストは 1 台のみであり、完全分散処理型のシステムの構築には向かない、内部バスとは異なり転送レートや転送間隔に関して、任意にはならないといった問題がある。しかしながら、USB デバイス側で低レベルの制御を行うことで、十分な制御性能を持つロボットシステムが実現できると考えられる。

そこで USB を通して指令値を受け取りモータを PID 制御するコントローラの開発を行ったので報告する。Fig.1 にノート PC と本コントローラにより構成したロボットシステム例を示す。開発したコントローラは 4 つのモータを制御周期 1[ms] で制御する事が可能である。モータドライバは内蔵せず、必要とする容量に応じたドライバあるいは、FET による H ブリッジを用意するものとした。

2. 概要

内蔵周辺回路が豊富であり、乗除算命令を持つことから、日立の 16 ビットマイコン H8/3067F を選択した。開発には H8/3067F (20MHz) とナショナルセミコンダクタの USB コントローラ USBN-9604、シリアル EEPROM の載った AW 電子の UB-H8-256A-04 (基板サイズ 75mm × 75mm × 15mm) を用いた。UB-H8-256A-04 から不要な部分を削除した基板を設計することで、さらに小型化も可能である。

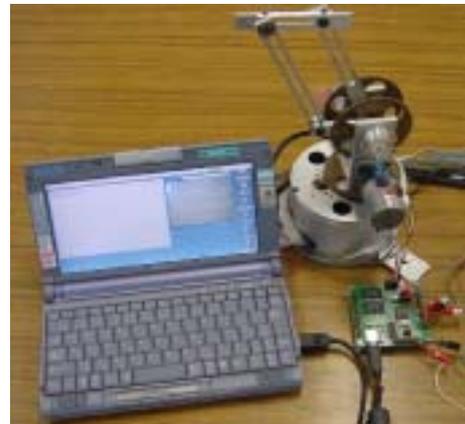


Fig.1 ノート PC と本コントローラにより構成したロボットシステム例 (右下がモータコントローラ)

H8/3067F は、3 チャンネルの 16bit timer / counter (内 1 チャンネルはロータリエンコーダの 2 相パルスをカウント可能)、4 チャンネルの 8bit timer / counter (2 チャンネルを 1 組として 16bit で利用可能)、8 チャンネルの A/D converter (内 2 チャンネルは D/A コンバータとピンが兼用)、2 チャンネルの D/A converter を内蔵している。16bit timer はシステムクロックを基準とすることができるのに対し、8bit timer は、その 1/8 が最大クロックとなっている。そこで、最大 4 チャンネルのモータを制御することとし、16bit timer を、2 相カウンタと、2 チャンネルの PWM 出力として利用、8bit timer をタイマ割り込みに利用し、4 チャンネルの A/D と 2 チャンネルの D/A を利用することにした。

PID 制御可能なチャンネルは 4 つあり、各チャンネルの入出力を Table1 に示す。用途に応じて A/D コンバータ、ロータリエンコーダによる位置入力、D/A コンバータによるモータドライバ用出力、H ブリッジ用 PWM 出力が選択可能である。PWM による D/A

チャンネル	入力	出力
0	A/D(10bit) またはハードウェアカウンタ (16bit)	D/A(8bit)
1	A/D(10bit) またはソフトウェアカウンタ (16bit)	D/A(8bit)
2	A/D(10bit), ソフトウェアカウンタ (16bit) またはハードウェアカウンタ (16bit, チャンネル0 と共用)	PWM による D/A(12bit) または Hブリッジ用 PWM と回転方向出力 (11bit)
3	A/D(10bit) またはソフトウェアカウンタ (16bit)	PWM による D/A(12bit) または Hブリッジ用 PWM と回転方向出力 (11bit)

Table 1 制御チャンネルと入出力のデバイス

制御出力時の PWM の周期は $4.8[\text{kHz}] (20[\text{MHz}]/2^{12})$ である。適当なローパスフィルタを通すことで、D/A 出力となる。Hブリッジ用制御出力の PWM の周期は $19.5[\text{kHz}] (20[\text{MHz}]/2^{10})$ である。ロータリエンコーダの入力の内3つはソフトウェアカウンタによるものである。ソフトウェアカウンタは4通倍のカウントを行っており、パルス読み込み間隔は平均 $0.019[\text{ms}]$ 、最大 $0.074[\text{ms}]$ となった。 $0.074[\text{ms}]$ 間隔 ($13600[\text{パルス/s}]$) のパルスは、一回転 500 カウントのエンコーダを使用する場合には、 $6.8[\text{回転/s}](400[\text{rpm}])$ に相当する。

USB から、目標位置(入力値)、目標速度、PID ゲインなどの設定が可能である。また現在の各チャンネルの入出力値を読むことができる。出力オフセット、D/A(PWM)/Hブリッジの使用(チャンネル2,3)はEEPROMに記憶することが可能である。制御出力 u は次式から $1[\text{ms}]$ 毎に計算、出力される。

$$u = A - \frac{K_P}{K_{P_x}}(x_d - x) - \frac{K_D}{K_{D_x}}(\dot{x}_d - \dot{x}) - \frac{K_I}{K_{I_x}} \sum_0^t (x_d - x) \quad (1)$$

整数演算のみとするため、ゲインを分母、分子に分けた。ゲインの分子 K_P, K_D, K_I は 16bit 整数 (-32768 から 32767)、分母 $K_{P_x}, K_{D_x}, K_{I_x}$ は 16bit の正整数 (0 から 65535)、オフセット A と目標値 x_d, \dot{x}_d は 16bit 整数とした。各項でまず乗算を行い結果を 32bit で保持し 32bit で除算、32bit の和(差)の計算を行い、最後に 16bit に丸めている。従って通常のゲイン指定では桁溢れは起きない。制御出力 u は 16bit の正の整数であるが、D/A, PWM の精度に応じて上位ビットが使われる。ソフトウェアカウンタの測定能力は下がるが、さらに制御周期を短くすることも可能である。また基本は PID 制御による位置コントローラであるが、 K_D を 0 とし、目標速度、オフセットを利用して、速度制御、トルク制御も可能である。

これら以外に、デジタル入出力を用意した。デジタル出力により、モータドライバへのフリー指令、ブレーキ指令などが可能である。またデジタル入力により、カメラや他のコントローラと同期することも可能である。

USB デバイスは Plug & Play により自動的にデバイスが認識されるが、同じデバイスを複数利用する場合には、認識される順序が変わる場合などがある。そこで EEPROM ヘデバイスを識別するためのマジックナンバを記録することにより、複数の本コントローラを識別可能とした。

3. エンドポイント

エンドポイント(USBにおける仮想通信路の端点)は USBN-9604 では 6 つ利用可能であるが、そのうち 4 つを用いた。1 つは制御のための目標値、ゲインなどを設定するためのものである。1 つはオフセットの値やカウンタのリセット、マジックナンバの EEPROM への記憶などを指示するためのものである。

現在の A/D やカウンタの値、制御出力などを知るためのエンドポイントは 2 つ用意しており、その値を読んだ時点での内部時計の値も返す。内部時計の値は $1[\text{ms}]$ 毎にカウントアップしていく。エンドポイントの 1 つからは、いつでも値を読むことができるが、その値は前回読み出された直後のものである。もう 1 つは別のエンドポイントから、データを送るよう PC から指示し、H8 が要求を受けたときの値を読むことができる。このような実装としているのは USB では通信要求が来てから、最新のデータを用意することは困難であること、連続して値が読みたい場合もあれば、USB のバルク転送を利用しているため、負荷の重いときなどは必ずしも一定間隔で読むことはできず、場合によっては古い値を読むことになる場合があることが理由である。

4. まとめ

USB を通して指令値を受け取りモータを PID 制御するコントローラの開発を行い、その概略を述べた。このコントローラに FreeBSD / Linux などより指令値を与え、モータを制御することが容易に実現可能となった。既に USB 接続のモータコントローラは報告されている¹⁾が、本機ではより周辺機器の豊富な H8 を採用したことで一つのコントローラで複数のモータを制御することが可能となった。

現在小型のカメラヘッドを開発中であり、本コントローラを用いて FreeBSD/Linux より制御する予定である。

参考文献

- 1) 徳山, 精廬, 橋本. USB インターフェース搭載の小型モータコントローラの開発. 第 18 回日本ロボット学会学術講演会講演予稿集, pp. 119-120. 日本ロボット学会, 2000.