解説

周辺機器と接続のための通信

—あえてシリアルポートをつかう—

Communication Lines for Peripheral Devices

—Legacy Serial Port for Simple and Robust System Development—

熊谷 正 朗* *東北学院大学工学部

Masaaki Kumagai* *Tohoku Gakuin University

1. はじめに

なぜあえて、レガシーな（旧来の）シリアル接続，すなわち，PC のシリアルポート（RS-232）[†]をいまさら話題にするかといえば，研究室での試作としてのロボット開発において，「使える」という観点では最適な手段であると筆者自身が確信しているからである。

想定しているロボットシステムは

- 中央に主たる制御を司る PC がある。
- 周囲に局所的な制御用やセンサ用のマイコンがある。
- PC 直結の市販装置（レーザレンジファインダ等）も，のようなものである。つまり，ある程度コンピュータが分散されたロボットシステムである。

これまで，PC のみ（MS-DOS，Linux）による制御，オンボード PC の採用，小規模の組込マイコンのみで制御，FPGA 上に専用の入出力処理回路を構築など，20 以上のシステムを開発したなかで，意外にも開発が楽なのが上記手法（もしくはマイコンが主でパソコンは調整用端末）であった。具体的には，ロボットに低レベル（モータの回転制御やセンサの一次情報処理）の処理を行う小型マイコンボードを搭載し，その上位に PC（Windows 等）を接続して制御している。リアルタイムな部分はマイコンが責任を持ち，扱いが容易で資源が潤沢な PC 側では，時間に比較的余裕のある主制御プログラムを実行／停止しつつ開発する。

このようなシステムでは PC とマイコンの間の接続手段が必要であるが，そこに古くから使われてきて“レガシーな”インターフェイス（IF）といわれるシリアルポートを使用する。PC 側にはまだ多少残っているほか，USB-シリアル変換ケーブルもある。90 年代から一般に普及し始めた汎用の組込マイコン（現：ルネサス社 H8 シリーズなど）のほとんども同様の通信機能を内蔵している。対して，現在の PC 周辺の接続手段として，より高速高機能な USB，IEEE1394，ネットワーク等があるが，マイコンでは処理

図 1 シリアルポートの典型的な通信波形。信号レベル変化の最短間隔が 1 [bit] 当たりの時間=ビットレートの逆数なので，26 [us/bit]≒38,400 [bit/s] が確認できる。通信に不具合が生じたときも波形を見れば，最低限，通信の有無と実際の速度が確認できる。その気になれば，通信内容も読める。多くの場合は，‘0’（で電圧は高い）のスタートビット，下位から順に 8 [bit] のデータ，‘1’のストップビット（1 [bit] 以上）で構成されている。‘0x52 0x53 0x4A’=?

の負担が重く，実装するには相応の知識が要求される。まして，PC 側で自前のドライバが必要となることもある。一方，マイコン側の通信手段として I²C バスや CAN バスなどもあるが，内蔵するマイコンは一部にとどまり，逆に PC 側では専用の IF の追加を要する。これらハード的な観点で，シリアルポートはいまだ有用である（そもそもシリアル接続を要求する市販装置も多い）。また，今時の通信方式は，一人の優秀な技術屋が実装をこなせたとしてもブラックボックスにせざるをえず，当人が去ったあとに何らかの改修を行おうとしても不可能に近い。原始的なシリアル通信であれば，ソフトを読めさえすれば，対応できる可能性は高く，信号も解析しやすい（図 1）。

そこであえて，シリアル通信のようなレガシーでシンプルな接続を採用することを本稿では推し，実践的な活用事例をソフトおよびハードの面から紹介する。

2. PC-マイコン通信の実際

本章では，筆者の元で全ロボットシステムに採用しているシリアルポート通信方式“SerialLoop”（SL）を紹介する。

原稿受付 2007 年 12 月 3 日

キーワード：Serial Port, USB, Legacy Device, Embedded Controller

*〒 985-8537 多賀城市中央 1-13-1

*Tagajo-shi, Miyagi

[†]パラレルに対するシリアルという点では USB も IEEE1394 も PCI Express もすべてシリアルであるが，以降，本稿でシリアルという場合は狭義のいわゆるシリアルポート，RS-232（422）規格による調歩同期式通信のことを指すものとする。

図2 SerialLoopの各ノードの構成。ノード上のレジスタ群はそのノード上のプログラムと、その他のノードから任意にアクセスできる（TxD：送信端子，RxD：受信端子）

マイコンとPCでデータを送受する場合、可読なテキストでの通信と、値をバイト列で直接送るバイナリデータでの通信がある。可読なものは汎用の端末ソフトで対応できるが、バイナリのほうが転送レートは高い[†]。

ロボット制御用であるため、効率を重視してバイナリを採用したいが、なるべく汎用性も確保しておきたいと考えて、以下のような仕様でSLを開発した（図2）。

- マイコンやPCを通信のノードとし、各ノードには16[bit]（C言語のshort型）と32[bit]（同long型）の整数変数（レジスタ）を各16個ずつ持たせる。
- 各レジスタはノードのメモリ上に置かれ、ノード上の制御プログラムなどは変数として直接読み書きできる。
- SLの通信経路の明示的な要求によって、各ノードのレジスタを他のノードが自由に読み書きできる。
- SLの通信は割り込み関数で実装し、主ルーチンからは意識することなく、いつの間にかレジスタ値が変化して送信されている。

例えば、ロボットのモータコントローラを作る場合、マイコンのメインループでは、制御に必要な各種ゲインや指令値を常にレジスタから参照する。逆に、現在の回転角度や速度をレジスタに常に書き込む。これに対して、PC側からはシリアル回線とSLを通して、ゲインや指令値を書き込み、現在値を取得する。つまり、マイコンごとにも共有メモリを用意し、マイコンのメインループとPC側（通信側）から読み書きできるようにしたものである。

本手法は文字通り、シリアル通信でループを作り、多数のマイコンを少ない回線でつなぐことを目的に設計したものである。図3に示すようにノードを環状に配置し、二つ目の送信を二つ目の受信に、二つ目の送信を三つ目の受信に、と接続し、1周させる。一般には、セットで使用する送受端子を異なる接続に分けて使用することが特徴である。

図3 ノードによるループ。すべてのノードを環状に直列接続し、要求に関与しないノードはただ次に転送する

各ノードにはアドレスを振り、データもパケット化して送信先（+返信）アドレスを付ける。受信したパケットが自分宛でなければ、そのまま次に転送する。自分宛は処理し、必要なら応答を次に送信する。これにより、ノード対ノードで通信が成立する。市販のLSIなどでも同発想のものがあるが、本手法はあくまでマイコン標準装備の範囲で実装することで、すぐに使えることを重視した。なお、簡単なチェックサム程度しか保護はなく、1ノードが停止しただけでも全体が停止するという点では脆弱である。

本ライブラリは、PCとマイコンの両方で制御系を構築する際のデータの授受に使用しているほか、マイコンのみで実装できる小型のロボット（学生のコンテスト参加ロボットなど）のデバッグなどにも活用している^{††}。また、無線装置を使うことで、複数ロボットの協調動作にも使用した。

3. USB-シリアル変換器の試作

本章では、ロボット用に試作したUSB-シリアル変換器を紹介する。

USB-シリアル変換には1,000円以下の安価な変換ケーブルも市販されている。しかし、基本的に1回線（ch）のみで、また信号もRS-232に限られている。そこで、FTDI社[1]の変換LSIのFT2232を使用して変換回路（図4）を試作した。本LSIを採用した理由は、ほとんど追加部品を必要とせずに1チップで2[ch]の通信が可能で、詳細なデータが公開されていること、また通信速度が3[Mbps]までかなり自由に設定できること、そして入手性の良さである。

回路の小型化を図って専用の基板を製作した。FT2232周辺はメーカー推奨回路の通りとし、部品実装の有無により

- RS-232×2ch（TxD，RxDのみ）
- RS-232×2ch（TxD，RxD，RTS，CTS）
- RS-422×1ch（TxD，RxD）

+ RS-232×1ch（TxD，RxD，RTS，CTS）

の3通りに使える設計とした（部品代は2,000円程度）。

基板は送受インジケータのLED^{†††}や両面テープで固定することを想定して1面のみの実装である。また、USBのコネクタも削減し、ケーブルを直接半田付けする[‡]。

[†]例えば、十進数なら、1/2[byte]の数値を送るのに3/5文字[byte]が必要で、改行などを入れるとさらに増える。

^{††}設計時には想定外の使い方である「要求されていないのにレジスタ値を定期的に返答する」という手でテレメトリにもなる。

^{†††}市販の変換ケーブルで付いているものは見かけないが、少なくとも送受されていることが分かり、動作確認には便利である。

[‡]コネクタの実装面積を削減する意味もあるが、コネクタの価格とケーブルの価格を見比べると、ケーブルを切ったほうが安い。

図4 FT2232によるUSB-シリアル変換基板。左の基板は半実装済み。中央がFT2232, 左はRS-232ドライバ, 右はRS-232, RS-422ドライバの追加実装用パターン

以下、本回路の使い道を紹介する。

3.1 H8/3052 マイコン用 Dual-link 変換器

論文等で記述を見かけることは少ないが、ロボット自作派の間でしばしば使用される組込マイコンにルネサス社製H8/3052マイコンがある。組込マイコンとしては比較的豊富な入出力機能（カウンタ5[ch], AD 10[bit] 8[ch], DA 8[bit] 2[ch] など）があり、プログラムは内蔵フラッシュメモリにシリアルポート経由で書き込むことができ、簡単なロボットの制御には手頃である。なにより、秋月電子通商社[2]でそのまま使えるマイコンボードの形に実装して販売しており、入手性もよい（通販で翌日に入手できることが多い）。筆者も頻繁に使用しており、前述のSL通信ライブラリも、このH8用に書いている。また、学生らのロボット製作の基礎ライブラリとして、ステッピングモータ制御（独立2輪駆動ロボット用、デッドレコニング付き）、ラジコンサーボの指令パルスジェネレータ（DMAを使い、ソフト的負荷ゼロで16[ch]）なども開発してきた。

このマイコンボードには通信（SCI, RS-232 2[ch]）および書込制御線、リセットをまとめた端子が出ている。ロボットの開発においては、マイコンは往々にして手元から距離のあるところに組み込まれているため、手で書込制御とリセットが可能な延長ユニットを製作して使用している。ユニットは図5に示すように、マイコン側とPC側の2個からなり、その間はRJ45端子のついたネットワークケーブルで接続する。ネットワークケーブルを採用した理由は、様々な長さ、太さのものを入手しやすく、状況に応じて適切なものを選択できる点にある[†]。PC側には書き込み、リセットの各スイッチと、シリアルポートに接続する2個のDsub（9[pin]）のコネクタがついている。

実際に使用してみると、ノートパソコンにUSB-シリアル変換ケーブルをつなぎ、それをPC側のユニットに接続するという使い方が非常に多く、煩わしかった。また、H8/3052にプログラムを書き込む際にはSCIの一方を使用するが、

[†]CAT5eのネットケーブルは、高速信号を流すことにも耐えうるツイストペアの汎用ケーブルとして案外使い道がある。

図5 マイコン用信号線延長ユニット。右のユニットの線をマイコンに接続し、左のユニットの両脇のDsubコネクタをPCに接続する（この写真ではELECOM社製USB-シリアル変換ケーブルを接続）。両者はRJ45端子付きケーブル（100BASE-TX用）で接続する

図6 USB接続版マイコン用信号線延長ユニット。PC側は図4の基板を内蔵する。従来品（図5）と比較し、Dsubコネクタが消え、送受信インジケータのLEDが追加された

デバッグなどに共用していた場合は通信ソフトをいったん切断する必要がある。2本のSCIの1本を書き込み用、1本を通信用にすれば切断の必要はないが、変換ケーブルを2本つなぐことになる。そこで、ワンチップで1本のUSBを2本のシリアルに変換できるFT2232に着目し、前述の開発に至った。変換回路を組み込むことで、図6に示すようにすっきりとまとまった。

ただし、書き込みに使用してみたところ、極端に速度が低下した。結論をいえば、FT2232は「受信バッファにある程度（62[byte]）たまる」か「指定時間の経過」によって、シリアル側から受信したデータをPC側に送信するため、デフォルト（指定時間＝16[ms]）では少量データによるハンドシェイクのレスポンスが悪化するようであった。これをドライバの設定で最短の1[ms]に変更したところ、遜色なく動作するようになった。

3.2 SICK LMS200 を USB 経由で接続

ロボットの環境センシングに使われるレーザレンジファインダの一つとして、SICK社のLMS200がある（図7）。本来の性能として、毎秒75回（横分解能をあげると37.5回/18.7回）のスキャン計測が可能であるが、そのために

図7 開発中のトレーラロボットの牽引部。前面に搭載されているのがレーザレンジファインダ。その直下にロボットの制御マイコンとモータドライバがあり、マイコンで電流および速度フィードバック、ステアリング指令を、PCからシリアル経由（SL）で受信した数値を元に行う。同時に、走行情報、連結部の角度を定期的にPCに送信する。PC側はマイコンからの通信を制御周期として使用する。本ロボットでは開発した変換器を2個使用する

はRS-422で500[kbps]の通信が可能なIFを必要とする。SICK社で対応品を用意しているほか、インタフェース社[3]のRS-422/485通信PCIボード（PCI-466102）、ノート用CardBusカード（CSI-466402）なども使用できた[†]。

しかし、近年のPCの小型化傾向によりPCIバスが削減される、そもそもボードの抜き差しが面倒、などの制約があり、最近では拡張バスより潤沢なUSBで接続できれば便利である、と想着て、試してみた。LMS200で使われているRS-422とRS-232の違いは電気的な特性の違い（差動か否か、電圧レベルなど）のみで、ビット列としての通信様式（調歩同期式）は同等である。そのため、FT2232による変換回路で、送受信ドライバのみをRS-422用に変えて接続してみた（ここまで想定して回路を設計していた）。

その結果、まったく問題なく動作した。Windowsでは、SICK社で公開されているサンプルアプリケーションが9,600, 19,200, 38,400[bps]で正常に動作し、一部アプリケーションでは500[kbps]での動作も確認できた（ソフトの仕様のためFT2232ドライバのレジストリの書き換えが必要）。Linux 2.6.18（CentOS 5）では、通常のシリアルポートで通信するように作っていたプログラム（もとは38400[bps]まで）において“/dev/ttyUSB1”をopen()し、500[kbps]に設定^{††}したのみで通信できた。Linuxでは、すでに500[kbps]や1[Mbps]などの設定が用意されており、また、それら速度をサポートしているFTDI製変換チップ用のドライバも今のLinux 2.6に（FTDI社によれば2.6.9

より）標準で含まれており、まさに「なにもすることなく」使えてしまった。

移動ロボットに搭載するPCは必然的に小型になり、拡張バスの制約も多く、USB接続できるようになったことには意義があると思われる。今回は自作の回路という再現性の悪い手段によったが、秋月電子社で販売されている1[ch]変換用のFT232Rを搭載した変換モジュール基板にRS-422ドライバをつなぐことでも同じ結果が得られると期待される。

なお、これらの変換回路はUSBハブ経由の接続も可能である。Windowsにおいて、バスパワー型の小型ハブに上記2種類の変換器を接続し、LMS200で500[kbps]の連続通信をさせつつ、マイコンへの115.2[kbps]にてプログラム書き込みを行ってみたところでは、両者とも速度低下は見られなかった。

4. おわりに

本稿では、古くから使われるシリアル通信の今の活用事例を紹介した。通信速度は遅く効率も悪いが、なによりシンプルであり、通信手段としては万人に分かりやすく、それゆえ研究開発・デバッグ・継承という観点からは“ロバスト”であると言える。その上で、使えるツールとするための仕様を検討し、実装して、実際に多用していたのが前半の事例である。現在のところ、筆者の元で大小10台以上のロボットが同じ方式を使っており、有効性は確認できている。後半では、USBという今時の接続を活用することでシリアル接続をより便利にした事例を紹介した。手間やコストを考えると自作は割に合わないが、その自由度の高さから、以前は面倒だったことが互換性を保ちつつ低コストで簡単？に実現できてしまう可能性もあることを示した。

本稿は「こういうこともできます」という紹介にとどまっているが、実際のソースや設計データなどの技術資料は、当方のWEBサイト[4]で公開する。それらも含めて、何らかの参考となれば幸いである。

参考文献

- [1] FTDI Chip Home Page: <http://www.ftdichip.com/>
- [2] (株)秋月電子通商: <http://www.akizukidenshi.com/>
- [3] (株)インタフェース: <http://www.interface.co.jp/>
- [4] 熊谷正朗: ロボット開発工学 熊谷研, <http://www.mech.tohoku-gakuin.ac.jp/rde/>
※各URLを直接指定するより名称を検索したほうが早い。

熊谷正朗 (Masaaki Kumagai)

2000年東北大学大学院修了、博士（工学）。同年東北大学助手。2003年東北学院大学講師。2006年同助教授（現：准教授）。専門はメカトロニクス全般（技術開発系）。現在は主に、学生が希望したロボットを共同開発することで、多種多様なロボットの効率的な実装手法を検討し、要素技術の開発を行っている。（日本ロボット学会正会員）

[†]ただし、その当時対応ドライバがなかったLinux 2.6を使う必要があったため、ハードのIOを直接操作した。少なくとも、ハードとして対応していることは確かだが、現在提供されている通信ドライバについては未評価。

^{††}LMS200に速度設定コマンドを送信、およびtcsetattr()にてポートに通信速度パラメータB500000を指定した。