

玄箱とシリアル通信を利用したデータ収集システムの構築を予定

岸 哲夫

ここでは、誌面の都合により今回掲載できなかったシステム構築の事例、および記事執筆時点に間に合わなかったシステム構築の事例について紹介します。これらの話題は、次号以降で取り上げていく予定です。

● シリアル・ポート経由でセンサ情報を取得

玄箱にシリアル・ポートを取り付ける改造を行い、センサなどのデータを Linux 上で読み出す実験を行います。RS-232-C インターフェース経由でセンサからパソコンにデータを取り込む感覚で、玄箱にデータを取り込んでいきます(図1)。

シリアル・ポートは1本の信号線で1ビットずつデータを伝送するインターフェースです。多くの場合、「シリアル」と言うと RS-232-C インターフェースを指しますが、IrDA や USB、IEEE 1394 などシリアル伝送の一種です。

シリアル・ポートは、基本的にデータの受け渡しに使用します。例えば、人間が常駐できない場所の温度や湿度、音、光、X線などを測定し、コンピュータにデータを受け渡します。また、その結果を機械に戻して制御する場合があります。筆者が構築を経験したシステムは、溶鉱炉の各個所に温度センサを取り付け、温度を計測するものや、電気炉の温度を管理するものなどです。これは、センサで取り出した温度によって熱源を制御するシステムでした。

また、単純な RS-232-C 機器の中には FAX モデムとして機能するものもあります。FAX 信号を受信すると FAX ソフトウェア

が起動し、受信を始めます。ひと昔前の BBS(パソコン通信サービス)では、モデム信号を受信するとソフトウェアが起動し、BBS が使えるようになるというものもありました。

● 専用のデバイス・ドライバの導入が必要に

例えばモデムの場合、“at”と伝送すると“ok”という応答が返ってきます。ok を受信できれば、データを読み出せたことになります。ソフトウェアのプログラミングを行う上では、IRQ (Interrupt Request Number) や I/O アドレスを意識する必要はないと思います。

複数のシリアル・ポートが存在する場合、コンピュータはそれぞれのシリアル・ポートと通信する必要があります。OS は、「各シリアル・ポートが存在する」とこと、「それらが存在する場所(I/O アドレス)」を知っていなければなりません。また、シリアル・ポートが CPU に何かを要求するときに、どの線(IRQ 番号)を使わなければならないかについて、OS は知っている必要があります。シリアル・ポートは、この線に割り込み信号を送ることによって何かを要求します。そして、シリアル・ポートにつながる各デバイスは、I/O アドレスと IRQ の両方の情報を不揮発性メモリなどに保存していなければなりません。

ここで言う I/O アドレスは、メモリ・アドレスとは別物です。I/O アドレスがコンピュータのアドレス・バスに設定されると、別の配線に信号が流れます。この信号が流れているとメイン・メモリはアドレスを無視しますが、固有の I/O アドレスを持っているすべてのデバイス(シリアル・ポート)は設定されたアドレスに注目し、それがデバイスのアドレスにマッチするかどうかをチェックします。もしデバイスのアドレスに一致していれば、その I/O デバイスはデータ・バス上のデータを読み込みます。

各シリアル・ポートには、ttyS0、ttyS1 などの名前が付けられます。これらは通常、DOS や Windows における COM1、COM2 などに対応します。/dev ディレクトリには、それぞれのポートに関する特殊なファイルが存在します。ここでは「ls /dev/ttyS*」を実行して、これらのファイルを確認してみましょう。ただし、例えば ttyS3 ファイルがあるからといって、必ずしもそこに物理的なシリアル・ポート(以下、物理ポート)が存在するというわけではありません。

以下に示すのは、改造していない玄箱の /dev/ttyS* です(この場合、実際には物理ポートは存在しない)。

```
ls /dev/ttyS*
/dev/ttyS0 /dev/ttyS1 /dev/ttyS2 /dev/ttyS3
```

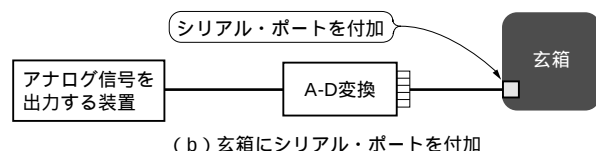
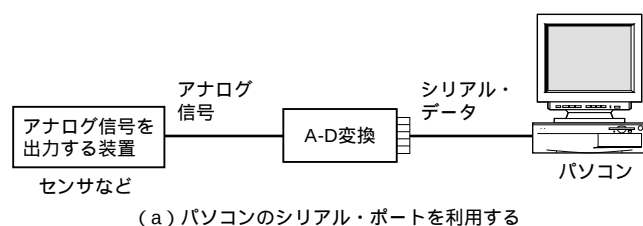


図1 シリアル・ポートの利用

測定システムでは、(a)のように、センサなどから出力されるアナログ信号を、A-D変換ボードを使ってシリアル・データに変換し、パソコンに取り込むことが多い。ここでは、(b)のように、パソコンの代わりに玄箱にデータを取り込めるようにする計画。

リスト1 setserial コマンドの実行結果

```
setserial
bash: setserial: command not found
```

(a) 玄箱の場合

ttyS0 や ttyS1 といった名前のどれがどの物理ポートを指すのかは、以下のようにして決まります。シリアル・ドライバ(ソフトウェア)は、どの I/O アドレスがどの ttyS に対応するかを示す表を管理しています。名前(例えば ttyS1)から I/O アドレス(と IRQ)への対応の設定および表示は、setserial コマンドによって行えます。

このプログラムは、ハードウェアそのものの I/O アドレスや IRQ の設定を行いません(ハードウェアの設定は、ジャンパやプラグ&プレイの設定プログラムによって行う)。このように、どの物理ポートが ttyS1 などに対応するかは、setserial によるシリアル・ドライバの設定とハードウェアの設定の両方に依存します。設定が間違っていると、物理ポートを(ttyS2 などの)名前に対応させることができなくなり、結果として使えなくなります。

玄箱上で setserial コマンドを実行した結果がリスト1(a)です。このように、玄箱には setserial なるコマンドは入っていません。一方、リスト1(b)は、Fedora Core 4(Linux ディストリビューション)をインストールしたサーバ(パソコン)に setserial コマンドを入力した結果です。玄箱でもこうした仕組みを使えるようにすることが、今回の実験の目標です。

● インターネット・サーバを構築

インターネットの普及に伴い、メール・サーバや Web サーバの需要が急増しています。一般に、こうしたサーバを構築する際には、専用のハードウェア(パソコンなど)を用意することが多いと思います。ここでは玄箱に BIND や Postfix, Apache, PHP, PostgreSQL といったフリー・ソフトウェアを導入し、固定 IP(Internet Protocol)または DDNS(Dynamic DNS)のサーバを構築します(図2)。このようなサーバを実現する場合、玄箱に Debian Linux を導入し、apt-get コマンドを使ってソフトウェアをインストールするのが早道です。

なお、玄箱付属の CD-ROM からインストールしなくても、Web ブラウザで IP を指定すれば Samba(ファイル/プリント・サーバ)の設定などは行えます。/www フォルダにそのための html ファイルが入っています。つまり、/etc/tthttpd.conf の設定を変えれば、簡単に Web サーバを構築できることにな

```
setserial
setserial version 2.17, 27-Jan-2000

usage:  setserial serial-device -abqvVWz [cmd1 [arg]] ...
        setserial -g [-abGv] device1 ...

Available commands: (* = Takes an argument)
                (^ = can be preceded by a '^' to turn off the option)
* port          set the I/O port
* irq           set the interrupt
* uart          set UART type (none, 8250, 16450, 16550, 16550A,
                16650, 16650V2, 16750, 16850, 16950, 16954)
* baud_base     set base baud rate (CLOCK_FREQ / 16)
* divisor       set the custom divisor (see spd_custom)
* close_delay   set the amount of time (in 1/100 of a
                second) that DTR should be kept low
                while being closed
* closing_wait  set the amount of time (in 1/100 of a
                second) that the serial port should wait for
                data to be drained while being closed.
^ fourport      configure the port as an AST Fourport
autoconfig      automatically configure the serial port
^ auto_irq      try to determine irq during autoconfiguration
^ skip_test     skip UART test during autoconfiguration

^ sak          set the break key as the Secure Attention Key
^ session_lockout Lock out callout port across different sessions
^ pgrp_lockout  Lock out callout port across different process groups
^ callout_nohup Don't hangup the tty when carrier detect drops
                on the callout device
^ split_termios Use separate termios for callout and dailin lines
^ hup_notify    Notify a process blocked on opening a dial in line
                when a process has finished using a callout
                line by returning EAGAIN to the open.
^ low_latency   Minimize receive latency at the cost of greater
                CPU utilization.
get_multiport   Display the multiport configuration
set_multiport   Set the multiport configuration

* rx_trigger    Set RX trigger level (ESP-only)
* tx_trigger    Set TX trigger level (ESP-only)
* flow_off      Set hardware flow off level (ESP-only)
* flow_on       Set hardware flow on level (ESP-only)
* rx_timeout    Set receive timeout (ESP-only)
* dma_channel   Set DMA channel (ESP-only)

spd_hi          use 56kb instead of 38.4kb
spd_vhi         use 115kb instead of 38.4kb
spd_shi         use 230kb instead of 38.4kb
spd_warp        use 460kb instead of 38.4kb
spd_cust        use the custom divisor to set the speed at 38.4kb
                (baud rate = baud_base / custom_divisor)
spd_normal     use 38.4kb when a baud rate of 38.4kb is selected

Use a leading '0x' for hex numbers.
CAUTION: Using an invalid port can lock up your machine!
```

(b) Fedora Core 4 が稼働するサーバの場合

- DNSサーバ・ソフトウェア(BIND)
- 電子メール・サーバ・ソフトウェア(Postfix)
- Webサーバ・ソフトウェア(Apache)
- Web構築用スクリプト言語(PHP)
- データベース管理システム(PostgreSQL)

インストール

玄箱

図2 玄箱をインターネット・サーバに

玄箱に BIND や Postfix, Apache, PHP, PostgreSQL といったフリー・ソフトウェアを導入して、インターネット・サーバを構築する。

ります。ただし、標準で入っている tthttpd という Web サーバ・ソフトウェアは、必ずしも機能が充実しているとは言えません。

きし・てつお