

# ワンボードマイコン及び FreeBSD マシンを使用した 遠隔電圧測定システム

高エネルギー加速器研究機構 小菅 隆、三菱電機システムサービス 久積啓一

## 1.はじめに

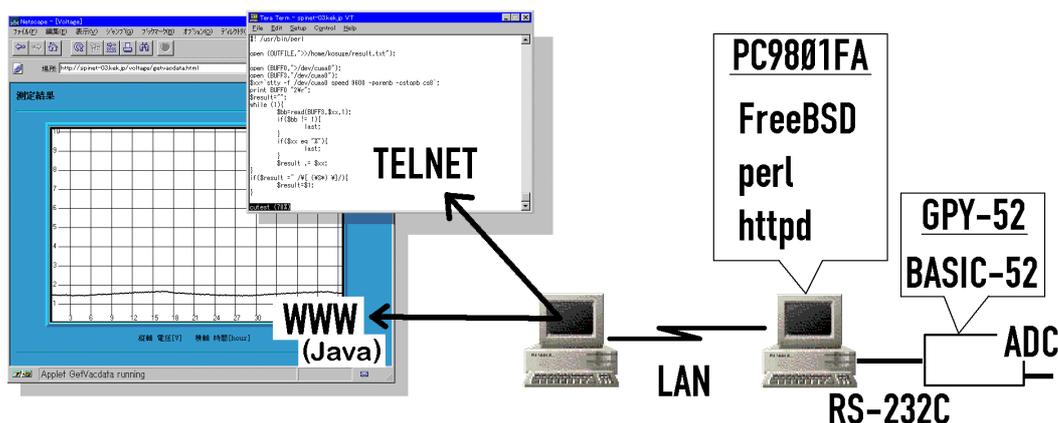
頻繁に立ち入る事が出来ない場所に設置された測定システムなどは、簡単なソフトウェア修正でも大きな労力が必要となる。また、遠隔地で監視を行うようなシステムも同様で、ソフトウェア修正が必要な場合、現場との往復はかなりの頻度となる。

今回、この問題を克服すべく BASIC インタープリタを内蔵した機器組み込み用のコンパクトなマイクロコンピュータと、FreeBSD をインストールしたパーソナルコンピュータ(以下 PC)を併用し、安価な遠隔測定システムを構築した。このシステムを利用すると、現場との往復が減り、効率のよいソフトウェアの修正が可能となる。更に、PC 上で httpd(WWW サーバプログラム)を動作させ、Java アプレットを用いて、他のコンピュータの WWW ブラウザ上で、この組み込み用マイクロコンピュータが測定したデータのグラフ表示が行えるようにした。

## 2.システムの概要

本システムは、RS-232C、パラレル I/O ポートを備え、インタープリタを内蔵したワンボードマイコン GPY-52(日本コムネット株式会社)と FreeBSD をインストールした PC9801FA(NEC)で構成されている。これら、GPY-52 と PC9801FA は RS-232C を介して接続される。また、PC9801FA は Ethernet Card を使用することで、所内のネットワークに接続される。ソフトウェアの開発や修正は現場の 9801FA 本体で行

図 1.システムの概要



う事も可能であるが、TELNET<sup>1</sup>等でネットワークを介して行う事も出来る。

このシステムを、今回は放射光研究施設の実験ホールに置かれた真空チェンバーの真空計に接続し、居室から真空度を確認する試みを行った。

<sup>1</sup> TCP/IP 上で利用される仮想端末用のプロトコル

### 3.GPY-52

GPY-52 は Basic インタープリターを内蔵した機器組み込み用のマイクロコンピュータで、仕様は表1の通りである。また、GPY-52 をケースに入れた様子を写真 1 に示す。

表1.GPY-52 の仕様

販売元	日本コムネット株式会社(開発製造イビデン株式会社)
CPU	80C52-BASIC
クロック	11.0592MHz
S-RAM	32k バイト
EEP-ROM	32k バイト
RS-232C 入出力	通信用コントロール機能
RS-232C 出力	シリアルプリンタ機能 TTL ・ 2 出力
アナログ入力	4 チャンネル 12 ビット 10 $\mu$ sec 標準 0~5V ・ 4 入力
デジタル入出力	8 ビット ・ 3ポート
電源	DC5V $\pm$ 10%
消費電流	35mA(標準)
外形寸法	100mm $\times$ 70mm $\times$ 19mm(Height)

ソフトウェアの入力は GPY-52 の RS-232C ポートを通して行う。実際には PC とパソコン通信用のソフトウェア等を利用して RS-232C 経由で GPY-52 と通信するとよい。

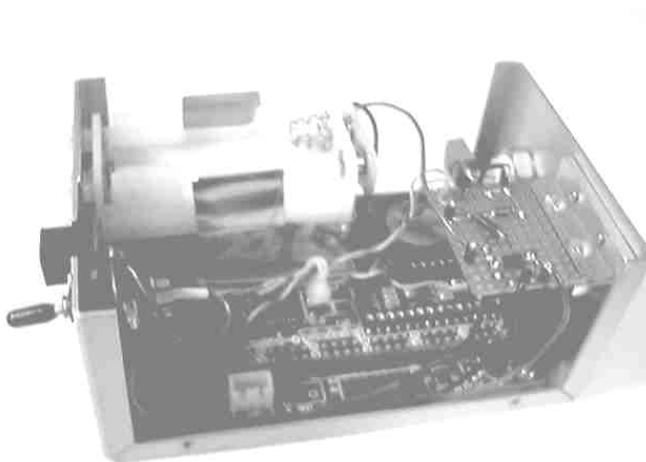
初期設定で GPY-52 は通信速度など自動検出するようになっているが、ユーザが任意に通信速度等を設定することも出来る。GPY-52 と接続がうまくいって通信が始まると、パソコン通信ソフトの画面に昔懐かしい“READY”という表示が現れる。

あとは行番号を入力してコマンドを記述してゆけば GPY-52 にプログラムをインプットすることができる。ここで“LIST”と入力すると、プログラムの内容が表示され、また、“RUN”と入力するとプログラムが実行される。インタープリタなので、容易に「トライアンドエラー」を繰り返しながらプログラムを構築する事が可能である。

プログラムは通信ソフトから決められたコマンドを入力する事で GPY-52 の EEP-ROM 上に保存されるが、通信ソフト側で通信内容のログをとるなどして手元の PC に保存しても良い。必要に応じて「アップロード」を行えば、再び GPY-52 にプログラムを入力できる。

また、GPY-52 は実行速度を速くする必要がある場合、機械語を使用した関数を作成することも出来る。しかし、ADC の読み込みなどの関数は予め用意されているので、そちらを利用すれば良い。

写真 1.GPY-52 をケースに組み込んだ様子



今回は、FreeBSD をインストールした PC-9801FA に GPY-52 を接続した為、FreeBSD マシン上の“cu”や“kermit”などのソフトウェアでプログラムを入力した。大部分の作業はネットワークを介して居室で(お茶を飲みながら)余裕をもって行った。(実際にはハードウェアの設定を行った後、PC9801FA の CRT とキーボードは外した状態で設置した。)なお、作成したプログラムの内容を図 2 に示す。

#### 4.PC-UNIX

Windows95 を動作させるには非力なマシン<sup>2</sup>でも、FreeBSD や Linux といったオペレーティングシステムは十分動作する。また、ネットワーク関連の機能が充実している事は大きな魅力で、メンテナンスやプログラム作成等もネットワークを介して行う事が出来る。

今回は、たまたま利用できる PC が PC9801FA であったため、FreeBSD を使用することにした。

#### データの読み込み

GPY-52 からデータを取り込む部分は“perl”(プログラミング言語)を利用して RS-232C から入力されたデータをファイルに書き込むようにした。

また、“cron”というソフトウェアを使用して、このプログラムを 5 分ごとに呼び出すように設定、自動的にデータを追加記録するようにした。実際のプログラムを図 3 に示す。

#### Samba の利用

作成されたファイルは必要に応じて利用すれば良いが、ここで“Samba”というソフトウェアを利用して、Windows 上から簡単にこのファイルにア

図 2.GPA-52 のプログラム

```
spinet-03: {24} cu -l /dev/cuaa0 -s 9600
Connected.

>/list
10 XBY(0FF27H)=02H
20 INPUT "%",C
25 IF C=2 THEN C=1 ELSE C=0
26 VT=0
27 FOR LP=1 TO 25
30 XBY(0FF24H)=C
40 LI=XBY(0FF24H)
50 HI=XBY(0FF25H)
90 VI=(HI*256+LI)
95 VT=VT+VI
100 NEXT LP
110 VO=VT/20480*1.02479
116 PRINT "[",VO,"]"
120 GOTO 20

READY
>run

%2
[ 1.2309489 E-2 ]

%
```

図 3.データを読み込む為のプログラム

```
#!/usr/bin/perl
open (OUTFILE,">>/home/kosuge/result.txt");
open (BUFF0,">/dev/cuaa0");
open (BUFF3,"/dev/cuaa0");
$xx=
`stty -f /dev/cuaa0 speed 9600 -parenb -cstopb cs8`;
print BUFF0 "2\r";
$result="";
while (1){
    $bb=read(BUFF3,$xx,1);
    if($bb != 1){
        last;
    }
    if($xx eq "%"){
        last;
    }
    $result .= $xx;
}
if($result =~ /\[ (\S*) \]/){
    $result=$1;
}
$dde=`date`;
chop($dde);
print OUTFILE "$dde\t$result\n";
close(BUFF3);
close(BUFF0);
close(OUTFILE);
```

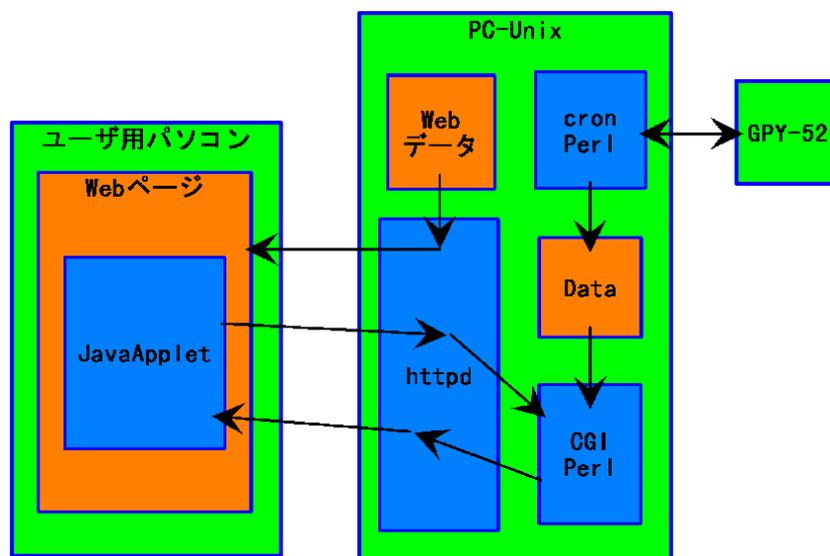
<sup>2</sup>FreeBSD の場合は文献によると CPU:386SX 以上、メモリ 5MB 以上、ディスク 60MB 以上とのこと。また、Linux については CPU:386SX、メモリ 4MB、ディスク 230MB(内約 130MB は使用していない)のマシンで動作を確認した。但し、メモリーが少ない場合についてのドキュメントを参照して、多少特殊な方法でインストールを行った。

クセスが出来るようにした。その結果手元のパソコンで、表計算ソフト等を利用し、簡単にデータの整理が行えるようになった。

### httpd 及び Java の利用

本システムの PC-9801 上で httpd を実行することで、他のユーザに計測データを公開することが出来る(図4参照)。更に Java アプレットを作成すれば、Web ブラウザ上でグラフ表示を行う事が可能である。

図 4.計測データの公開



また、この時データの監視を希望するユーザには Java アプレットが置かれた URL を知らせるだけで良い。なお、Java アプレットのメンテナンスは必要に応じて居室のパソコンで行う。

## 5.実際に使用してみて

まず、GPY-52 についてであるが、組み込み用のマイクロコンピュータのソフトウェアがここまで簡単に作成できるとは思ってもいなかった。実際に書いたソースコードはまさに図2に示すものだけである。

また、ネットワークを介してソフトウェアに関する殆どの作業が可能なのは、予想以上に効力を発揮する。今回も何度かソフトウェアの手直しを行ったが、現場に直接出向く必要も無く簡単に手直しが行えた。修正後はまさに「放っておけば勝手にデータをとってくれる」という状態で、機能を追加する場合は気の向いた時に居室の端末から作業を行えば良かった。

今後の展開としては Java アプレットを使った便利なツールの作成や、無線 LAN を利用してシステムを容易に移動出来るようにする事等を考えている。