報　文

組込みＯＳを用いた機器制御システムの構築と検証

秋元英二*

Development of machine control system that uses Embedded Technology

AKIMOTO Eiji

組込み技術で利用可能な OS の Linux、μITRON 及び Windows Embedded に着目して、その組込みシステムの構築をするとともに、機器制御を目的とし通信ポートから外部機器を動かすための実験装置とそれを動かすプログラムを作成し評価を行った。この結果、各組込み OS の特性が評価されるとともに、その上で動くプログラムと実験装置がうまく連動して動作することが検証できた。

キーワード：組込み、Linux、μITRON、Embedded、制御

はじめに

産業機器や各種家電製品に内蔵されるコンピュータシステムにおいては、機器を動作させるハードとそれを制御するソフトが１つの組込みシステムとして成り立っている。この組込みシステムの制御に用いられる OS は「組込み OS」と云われ、Linux、μITRON、Windows Embedded などがある。組込みの市場性は、表１、表２に示されるように近年急速に拡大している。

表１　組込み Linux における市場

分　野	2006 年	2009 年予測	伸率
業務/産業用	20 億円	28 億円	140%
民生用	12 億円	17 億円	136%
計	32 億円	45 億円	138%

表２　エンデベデッド市場

分　野	2006 年	2009 年予測	伸率
OS 市場	89 億円	108 億円	121%
SI 市場※１	7,200 億円	9,400 億円	130%
ボード市場	243 億円	281 億円	115%

※１　SI：組込みシステムインテグレート

表１、表２(株)富士経済 HP より

このような状況にもかかわらず、国内で組込み技術の開発者不足が顕著であり９万９千人が不足するとされている。1) 県内の機械製造メーカを中心とした聞き取り調査でも「組込み技術」に対する関心は高いものの、開発の効率化や資産の継承性の面から有利な組込み OS を取り入れた開発をしているのは、ごくわずかである。

そこで本研究では、今後、県内の機械製造業に必要とされる組込み技術について、機械制御を有効に行う上で重要となる組込み OS を用いたシステムの構築と外部機器制御のモデルを製作し、ソフトウェアとハードウェアの両面からその有用性について検証したので報告する。

実験方法

１．Linux OS の shrink

組込み OS は、図１のイメージ図にあるように従来、ハードウェアに依存して、ソフトウェアを構築していたためにハードウェアへの依存性が高くなり開発効率が悪くなるという問題があった。そこで、ソフトウェアをハードに近い部分を担当するもの(組込み OS)と、実際に作業をさせる部分のソフト(応用ソフトウェア)にわけることで、ハードウェアの変更にともなうシステム改変のボリュームを小さくし、結果的にトータルコストの削減やソフトの継承が可能である。

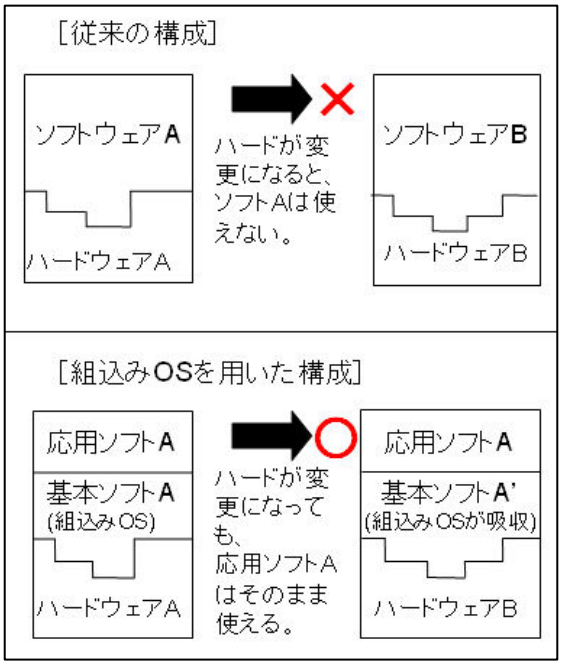

図１　組込み OS のイメージ図

*1　現土木部道路都市局建築住宅課営繕室

この研究は、「組込み技術を用いた機器制御システムの研究開発」の予算で実施した。

また、組込みシステムの分野では、ここ数年の間に応用ソフトのプログラム量が激増してきており(表３)今後さらに増大するものと思われる。

表３　組込みソフトウェアのステップ(行)数の増加

分　野	2001 年	2008 年	伸率
携帯市場	100 万行	500 万行	5 倍以上
自動車市場	100 万行	500 万～1000 万行	5 倍～10 倍
DVD 市場	20 万行	100 万行	5 倍以上
PC 市場	1500 万業	5000 万行	3 倍以上

※H20 年 経済産業省資料より

組込みシステムでは、プログラムの格納量には制限があるため、増え続ける応用ソフトを収納するには、OS のコンパクト化が考えられる。このことについて、Linux OS をモデルにして組込み OS の構成ファイルにおけるサイズの縮小(shrink)について実験を行った。

実験方法としては、図２にあるように Windows 上で任意の仮想マシンを構築できる VMware の中に Linux(Fedoar)を投入し、その上で組込み用 OS を作り込み、出来上がった各プログラムに対して strip を実行した。strip は、コマンドやライブラリに含まれるシンボル情報を管理するシンボル・テーブルを取り除くことができる。組込みの場合、このシンボル情報を取り除いても特に問題となるものではない。また、比較のためにベースとして Fedora5 と fedora8 の両者でも実験した。

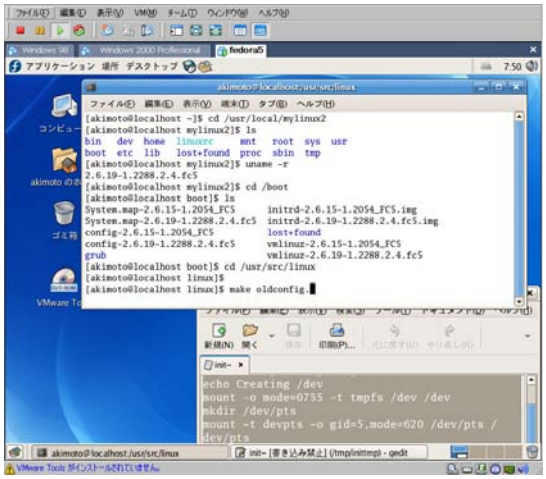

図２　VMware で組込み用の Linux を構築中

２．μITRON OS でのモータ制御

μITRON は、Linux や Windows に比べて起動時間が短いのが特徴である。このことは、電動車などのスタートキーを入れるとすぐに動作して欲しいものでは重要な要件である。図３にあるμTRON ボードを用いてモータを駆動するプログラムを作り、電源投入後に即応することを確認する。そのための開発環境として当初 Windows マシン上で Cygwin を用いて開発を行っていたが、開発の効率化のために eclipse を導入した。

また、プログラムの転送には、ターミナルエミュレータの TeraTerm を用いて実施した。

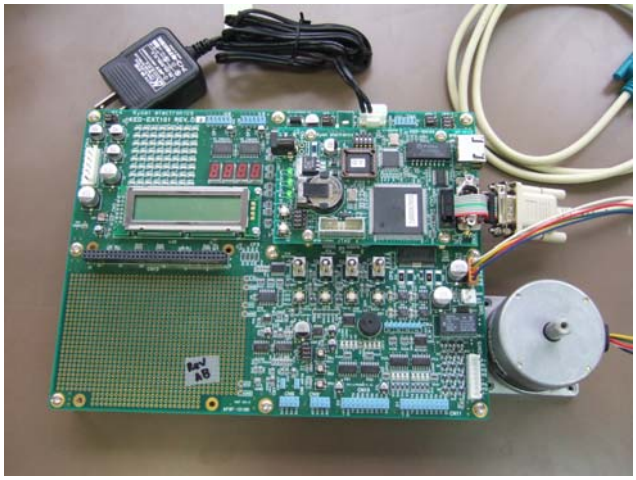

図３　Tron ボード (図中右 ステッピングモータ)

３．Windows Enbedded でのプログラム動作

組込み OS の中で、Windows(XP Embedded あるいは CE)を選択する場面も多い。経済産業省の 2007 年版組込みソフトウェア産業実態調査報告書によれば、組込み OS のシェアは、TRON(27.1%)、Windows(22.9%)、Linux(14.7%)、自社独自(9.0%)となっている。

さらに同報告では、使用言語として、C(56.4%)、C++(36.6%)、Java(0.9%)、その他[C#や Visual Basic など](3.6%)となっている。

上記言語が全て使え、最もとりつきやすいのが Windows である。そのため、開発者にとっては同じ系統の Windows Enbedded には違和感が少ない。

Windows 上で開発したプログラムが Enbedded でも利用できる点は魅力である。しかし、この OS の構築には高度なノウハウが必要となる。

ここでは C#でプログラムを書き、そのプログラムを Windows Enbedded に移植しその動作について、検討した。

そのモデルとして、図４の装置を作った。これは、４個の圧力センサを組み合わせて、かかる加重の量を情報として取り込むものである。データは、そのままでは取り込めないので、途中にレベル変換をするモジュールを挿入して、TTL レベルから RS232C レベルにしている。この実験では、移植されたプログラムの動作検証とリアルタイムな変化をうまくキャッチできるかをみている。

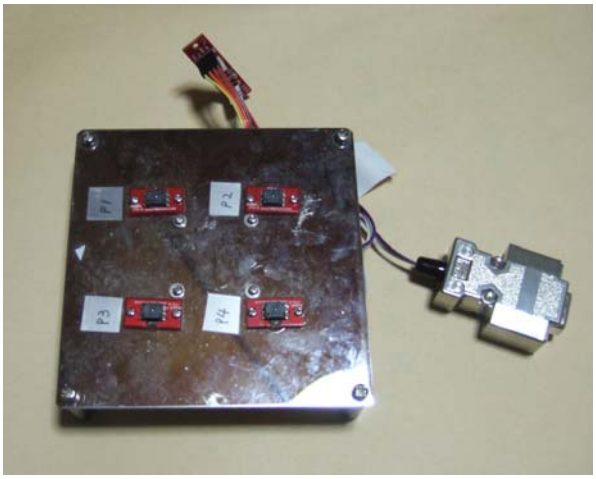

図４　圧力センサを配置した実験装置

４．端末の拡張

本研究の組込みの応用先として、電動福祉車両あるいは農耕車などの車両用センシングシステムが考えられる。組込みシステムは接続する端末により、いかようにも用途が拡大する。その１つの事例として、今回、「耳たぶセンサ」を試作した。（図５）

耳たぶに、この装置の赤外線センサを挟み込むと脈拍に応じて血流が変化するのでそれを赤外線の変化として取り込むものである。健康チェックの他には、心拍状況をうまく解析することで居眠り防止装置としても利用ができる。

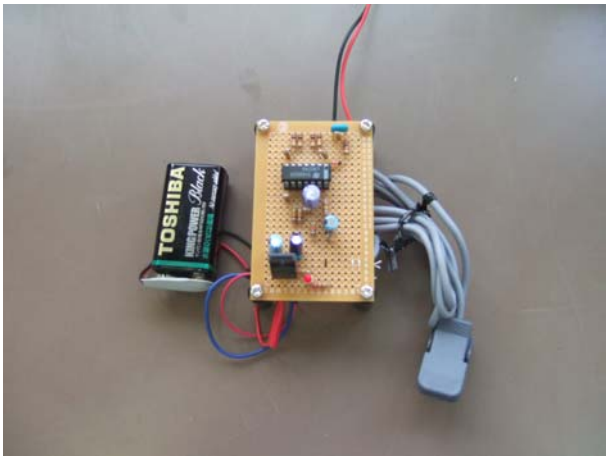

図５　耳たぶセンサ(試作)

５．開発環境

今回、実験に用いた装置及びソフトを以下に示しておく。

(1)Linux 関係
- VMWare 5.5.1
- Fedoar 2.6.20-1.2320.fc5
- Fedoar 2.6.24.3-12.fc8
- 組込み装置 PC-CUBE（株）FASE

(2) μITRON 関係
- 組込みボード KED+SH101（株)協栄エレクトロニクス
- eclipse 3.2.0:M20060921-0945
- Tera Term Professional 4.53
- μITRON 4.0

(3) Windows Embedded 関係
- Windows XP Embedded
- 組込み装置 PC-CUBE（株）FASE
- Visual C#2005 8.0.50727.762

結 果 と 考 察

１．Linux OS の shrink

実験方法で述べた手順により、VMware 上に Fedoar5 と Fedora8 をインストールした。この時、ターゲットとなる CF の容量が 1GB であるため、組込み用に構築する作業エリアのパーティションサイズも同容量の 1GB に割り当てた。（図６）

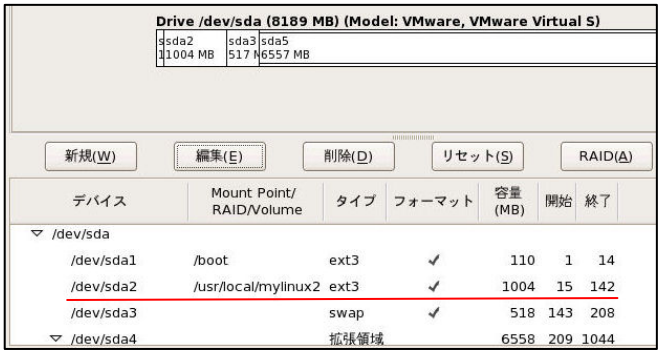

図６　ターゲット容量の設定(/dev/sda2)

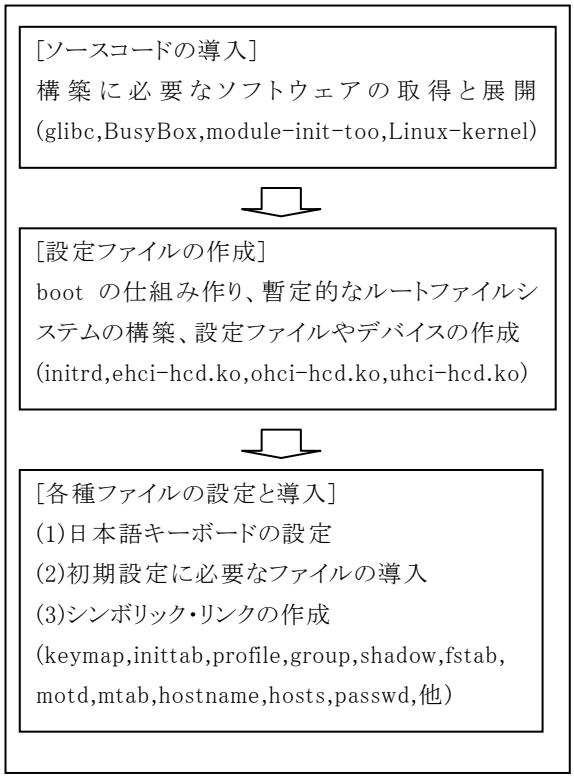

図７　組込み用 Linux の構築フロー

この作業後、図７にあるフローにそって組込み用 OS を構築していく。これを Fedora5 と fedora8 の両方に実施した。

shrink するのは、出来上がった OS のファイルシステム群を CF に書き込む前、つまり tar.gz 形式で圧縮送信する前に strip コマンドを使って実施した。
ただし、ターゲットには、数多くのコマンドやライブラリなどがあるので自動で shrink できるスクリプトを作成して行った。

図８が、Fedora5 における組込み OS の処理前で図９が処理後である。

```
[root@localhost mylinux2]# ls
bin   dev  home  linuxrc     mnt   root  sys  usr
boot  etc  lib   lost+found  proc  sbin  tmp
[root@localhost mylinux2]# du -s
409724  .
```

図８　組込み OS（Fedora5Base)における strip 前

```
[root@localhost ~]# cd /usr/local/mylinux2
[root@localhost mylinux2]# du -s
88608   .
```

図９　組込み OS（Fedora5Base)における strip 後

続いて図１０、図１１にあるように、バージョンを変えて Fedora8 で実施した。これらを表４にまとめた。

```
[root@localhost local]# cd mylinux2
[root@localhost mylinux2]# ls
bin   dev  home  linuxrc     mnt   root  sys  usr
boot  etc  lib   lost+found  proc  sbin  tmp
[root@localhost mylinux2]# du -s
380632  .
```

図１０　組込み OS(Fedora8Base)における Strip 前

```
[root@localhost local]# cd mylinux2
[root@localhost mylinux2]# du -s
64388   .
```

図１１　組込み OS(Fedora8Base)における Strip 後

表４　ファイルサイズの状況

Base	strip 前	strip 後	縮小率
Fedora5	409724byte	88608byte	0.22
Fedora8	380632byte	64388byte	0.17

表４から分かるように、約８０％ものサイズ縮小ができた。この効果は大きく、増大する応用ソフトの収容領域確保に貢献するとともに、CF の容量自体を小さいものにすることも可能である。つまり、装置の価格を低くすることができる。

また、fedora のバージョンの違いによる縮小率の違いは、バージョン７からシステム設定の変更が行われており、それにともない各ファイルシステムのサイズや付随してシンボル情報の有無の違いが出てきたものと思われる。

２．μITRON OS でのモータ制御

モータのドライバーコントロールには、PWM 制御の DC モータやパルス入力のステッピングモータがあるが、今回は OA 機器に多く用いられているステッピングモータをμITRON 上で用いることにした。この制御には、パルス数や周期の設定を変えることにより回転角度や回転速度を可変とすることができる。

図１２が、μITRON 上で動作するモータのプログラムフローチャートである。

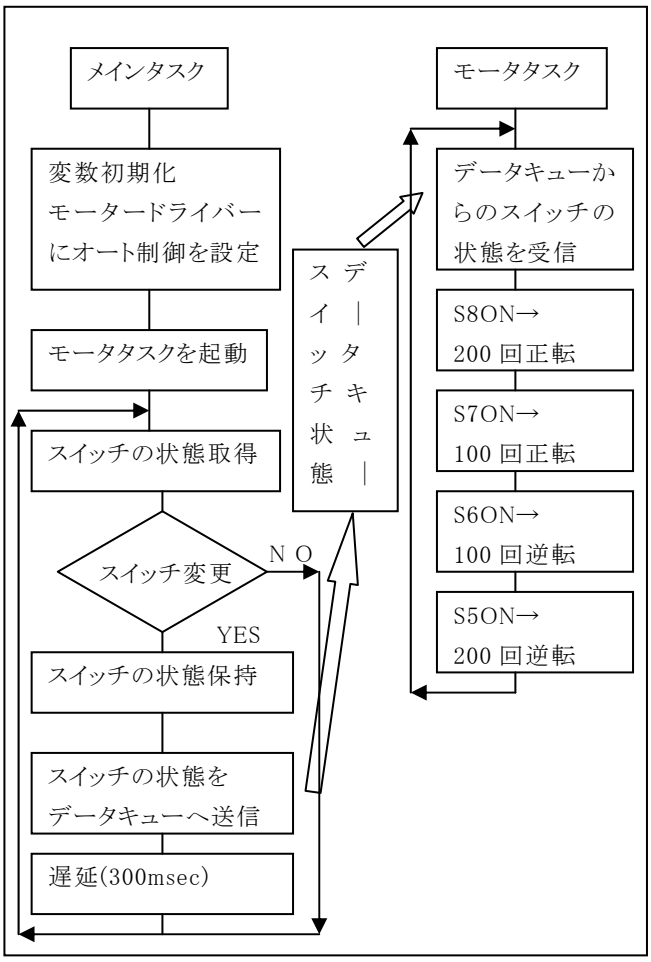

図１２　フローチャート

これをもとにプログラムを作成し、eclipse 上で depend build と make build を実施することでモトローラ形式のバイナリファイルが生成される。
これを、ターミナルエミュレータの Tera Term とクロスケーブルを用いてボードに転送する。図１３がその転送中の様子であり、図１４は転送が成功し完了したところである。

図１３　モトローラプログラムの転送

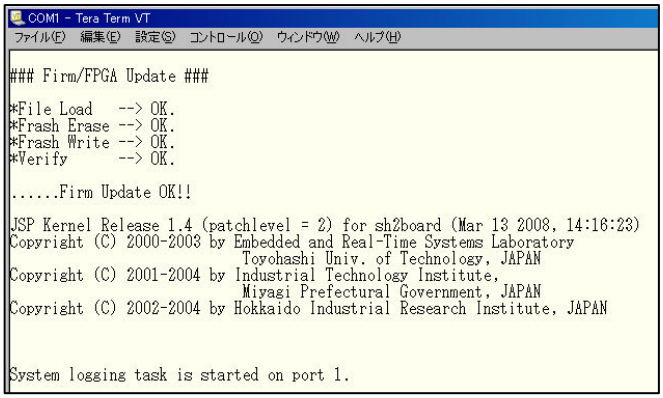

図１４　転送が成功し完了

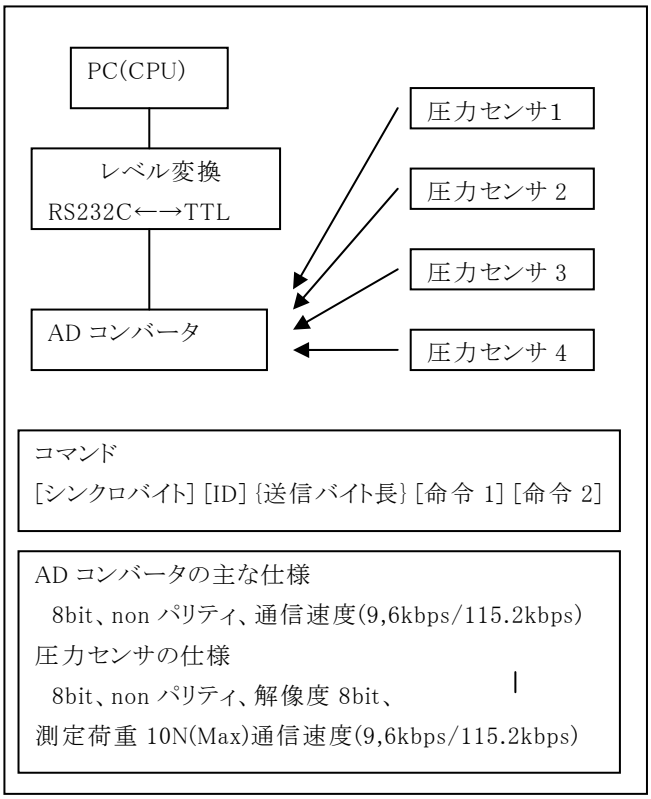

図１５　圧力センサの概要

転送完了後、ボードを一度リセットしトグルスイッチの ON/OFF により、正常にモータが回転することを確認できた。

また、電源投入後のモータの立ち上がり時間をみたところ１秒足らずで起動した。Linux や Windows ではできない仕業であり電動車などの時間待ちができない機器への適用には有効であることが分かった。ただし、プログラムの作り込みにおいてはアドレスの取り扱いに注意が必要である。

３．Windows Enbedded でのプログラム動作

図４にある４個の圧力センサの用途としては、車いすや電動の福祉車両、あるいは乗用タイプの農耕車などにおいて、搭乗者の安全や事故防止に役立つ。つまり、居眠りや急病などにより姿勢が偏った場合には、圧力に偏った分布が現れるので、そこから異常を感知することが出来る。

図１５には、システムの接続フローとコマンド、使用した AD コンバータと圧力センサの仕様を示している。これらをもとに、C#でプログラムを作成したところ圧力が数値で表示された状態が、図１６に示されている。

実験の結果、圧力に応じてリアルタイムに数値が変化して現れた。実用化するには、人の体重に耐えうるセンサに取り替えることで対応可能である。今後は、４つのセンサからの情報について、閾値を含めてどのように処理するか、あるいは制御機器との連携が課題となる。

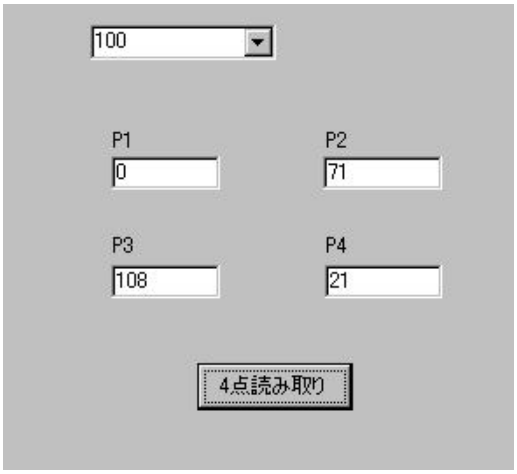

図１６　圧力センサからのデータ画面

４．端末の拡張

今回試作した耳たぶセンサの回路図が図１７である。センサからの情報をオペアンプで増幅している。実際にオシロで取り出した情報が、図１８に示しているが、脈拍に応じた波形が電圧に変換されて出力されていることが分かる。

しかしながら、今回の試作では、次の問題点があった。

・波形の頂上部がカットされることがある。
・波形が途切れることがある。
・波形自体が乱高下する。

これらの原因は、センサ自体の特性として非常に低周波のドリフトが乗ることによる影響であると考えられる。

この対策としては、ハイパスフィルタの挿入かあるい

はソフトウェアでデジタルフィルタを構築することで、このドリフトをキャンセルすることである。また、信号が切れているのは、実際には途切れていないのでソフトウェアで補完してやることで解決できる。

今回、反射型センサを用いたが、透過型センサのほうが安定しているというデータがあるのでセンサの変更により改善の見込みがあると思われる。

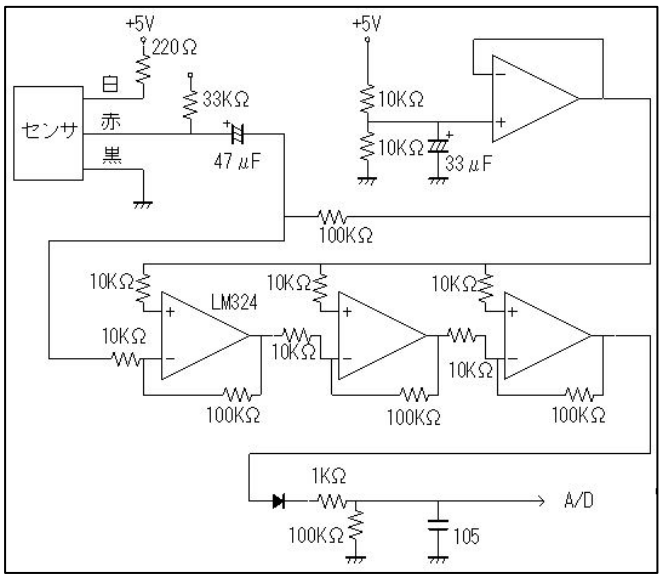

図１７　試作した耳たぶセンサの回路図

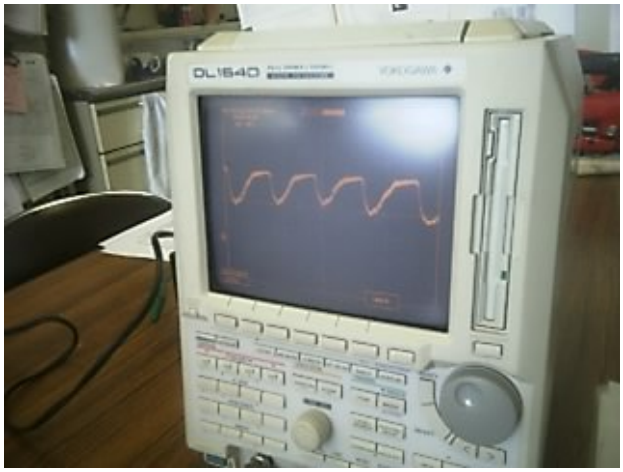

図１８　耳たぶセンサの波形

ま と め

組込み技術を用いた機器制御システムの開発として、今後さらに普及してくるであろう組み込み OS の代表的な３者である Linux、μITRON 及び Windows Embedded について、実験モデルの試作とともに検証してきた。
そのまとめとして､各種の組込み OS について表５に示す。

各 OS とも一長一短であるが、その特徴を端的に述べると、Linux はオープンソースであるためにカーネルの構築などは、参考情報が多く容易にカスタマイズができる。

μITRON は、システムの起動が素早くでき電動車などのセンシング装置の制御にむいている。Windows Embedded は､移植性に優れており Windows プログラムの流用がそのままできる。

表５　組込み OS の比較

	Linux	μITRON	WinED
OS 構築	○	△	△
OS 立ち上がり	△	○	△
普及率(国内)	△	○	△
開発・移植性	○	△	○
ドライバ	○	△	○

※WinED=Windows Embedded

また、各 OS 用のプログラムを作成し機器を動作させたところ、いずれもうまく動作し外部機器との接続は可能であることを確認した。

どの OS を用いるかは、ユーザの利用環境によるところが大きいが、資産の継承性やリアルタイム性・マルチタスク性あるいは開発システムのボリュームなどをトータル的に考慮して最適な組み込み OS を用いることが肝要である。

文　　献

1) 経済産業省:2007 年版組込みソフトウェア産業実態調査報告書.
2) 株式会社協栄エレクトロニクス:技術解説書(2007).
3) 重松保弘:UNIX プログラミング入門講座(アスキー出版局)(1994).
4) 日経 BP 社:日経 Linux6(2006).