平成14年度東北地区情報技術教育研究会 発表資料

KARACRINによるオートメーションサーバの構築

メイ	ンメニュー					SYS	TEM/ENV ?
				888. See			
	and the second				áihu a Irría		
- 1963 2000							pri la se
		Service States			\leq	Kanagawaok	1Nan ie/Hokusai
Ē	監視/手動操作	トレンドク	"ラフ 制御プ	ログラム こ	スケジュール起動	画像記録	
		A			\bigcap		
				Ĩ∠∕C	ιζ.J		KODDCDIU
							KHRHUKIK



期 日:平成14年6月20~21日 場 所:田沢湖高原温泉郷「山麓荘」

岩手県立千厩高等学校 産業技術科

梅村 吉明

はじめに

インターネットの爆発的な普及により、今では誰でも Web で情報の発信や検索、 メールでの情報のやり取りが出来るようになりました。言うまでもなく、インター ネットを利用することにより、一人の人間が得ることができる情報量が数年前と較 べて飛躍的に増加しました。

一方、パソコンのOSといえばMicroSoftのWindowsが圧倒的な シェアを誇っていますが、近年、注目されているのがPC-UNIXのLinux です。LinuxはフリーソフトのOSでありながら、UNIXと同様に高度なネッ トワーク機能を持っています。簡単に各種のサーバ設定ができ、運用時の安定性には 定評があります。

今回の研究ではOSにLinuxを用いて、その上にkaracrixという計 測制御の機能を提供するソフトをインストールします。Linuxのネットワーク 機能がそのまま利用できますので、いままで前述のような使い方しかしていないブ ラウザやメーラーで、インターネットあるいはイントラネットの形でサーバパソコ ンに接続されている機器を制御することが可能になります。

事例を示しながら、インターネット時代の計測・制御はどういうものかをみなさん と一緒に考察したいと思います。

2. Linuxと各種サーバ設定

L i n u x はフィンランドのヘルシンキ大学の学生だった Linus Torvalds 氏を中心 とするメンバーで、UNIX のソースコードを1行も使用しないでゼロから書き上げた OS です。それまで高価なコンピュータでしか動かなかったUNIXが安価なパソコン でも利用できるようになりました。

L i n u x はUNIXと同様にネットワーク機能がすぐれているので、コンピュータ ネットワークシステムのサーバとして広く用いられています。

今回の研究では、Linuxマシン1台とWINDOWSマシン1台の2台で構成され る小規模なネットワークで実験しました。なお、Linuxのデストリビューション にはいくつかの種類がありますが、今回用いたのはTurbo Linux4.2です。

A.システム構成

サーバ 使用パソコン 組立てDOS/V

CPU Intel Celeron700MHz ビデオチップ S3 ViRGE/DX ハードデスク 5.0GByte (windows98) 3.0GByte (TurboLinux4.2) *8MByteのHDDをパーテンションコマンダーで分割した。 ネットワークカード Laneed LD-PCITL

ドメイン名	ту.со.јр
サーバ名	ns.my.co.jp
w e b サーバ名	www.my.co.jp*
メールサーバ名	mail.my.co.jp*
f t p サーバ名	ftp.my.co.jp*
	*ns.my.co.jp と同義である。
TCP/IP 設定	192.168.0.1
ネットマスク	255.255.255.0
ゲートウェイ	192.168.0.254

クライアント 使用パソコン SONY VAIO PCG-FX55/BP

ノートパソコン Windows me

CPU Intel Celeron 7 0 0 MH z

モデム、ネットワークカード内臓

TCP/IP 設定 192.168.0.3

- B. 各種サーバ設定
 - ア.DNS サーバ

今回のネットワークは小規模ですのでホスト名と IP アドレスの対応は/etc/hosts ファイルで管理すれば十分ですが、勉強をかねて BIND を用いました。 BIND を使うには次の3つのファイルを設定する必要があります。

/etc/named.conf

/ v a r / n a m e d / n a m e d . h o s t (正引きデータベース) / v a r / n a m e d / n a m e d . r e v (逆引きデータベース) さらに、クライアントのWINDOWSではネットワークコンピュータのプロ パティでDNS設定を行わなければなりません。

/etc/named.confの設定

// generated by named-bootconf.pl options { directory "/var/named"; * If there is a firewall between you and nameservers you want * to talk to, you might need to uncomment the query-source * directive below. Previous versions of BIND always asked * questions using port 53, but BIND 8.1 uses an unprivileged * port by default. // query-source address * port 53; }; 11 // a caching only nameserver config 11 zone "." IN { type hint; file "named.ca"; }: zone "0.0.127.in-addr.arpa" IN { type master; file "named.local"; } zone "my.co.jp" { type master; file "named.host"; }; zone "0.168.192.in-addr.arpa" { type master; file "named.rev"; };

;/var/na @ IN S	amed/named.host GOA ns.my.co.jp. root.ns.my.co.jp. (1999081501;Serial 3600;Refresh 900;Retry 3600000;Expire 3600):Minimum
;	IN NS ns.my.co.jp. IN MX 10 mail
; ns mail www ftp	IN A 192.168.0.1 IN CNAME ns IN CNAME ns IN CNAME ns

/var/named/named.revの設定

;/var/na	amed/named.rev
; @	IN SOA ns.my.co.jp. root.ns.my.co.jp. (1999081501;Serial 3600;Refresh 900;Retry 3600000;Expire 3600);Minimum
;	IN NS ns.my.co.jp.
i	IN PTR ns.my.co.jp.

イ. FTPサーバ

パソコンとサーバの間である程度大きなファイルをやり取りする場合はFTPを 用います。アクセス制御を行わなければ、基本的にはLinuxをインストールし た状態で使用できます。

ウ.webサーバ

webサーバを構築するには現在もっとも広く利用されている Apache を使用します。Apache は httpd デーモンと設定ファイル httpd.conf、srm.conf、access.conf で構成されます。「webサーバのホスト名」と「webサーバの管理者のメールアドレス」の2点を設定し、httpdをスタートさせることにより利用できます。

一太郎でアニメーション GIF 入りホームページを作成。index.htmlと保存。

A	(本入力)	0000	494	(200		* ×	* •	1 07 67 6		
423	-	AS REPORTED	17.0	. 8	AB	1 .		EES		
1 7	- <u>à</u>	7	7	7				7	B.	199 E
1	24	にちわ								
	8									
		-	-							
		00								
	2	4 24 2	6							
		191								
F 0 Sheet I	/New 1		EK6-1	18 17	9 1E 4	- 100				
FWA	MS HINLE	+079870+						1-+	21410	
-40" 79.291	WW 1-8	E SARAN	-	71.6-8	-	IME Ph	ド:読み方の	和10,42(文字)	入力的起来。	PUTCE
1.9-1 5	10 2 4	*	S - Dindex.itd	1					3100	130

MSDOS プロンプトより、FTPでサーバにデレクトリ public_html を作成。そこに index.html と GIF ファイルを転

送する。

C¥WINDOWS>ftp ftp.my.co.jp Connected to ns.my.co.jp. 220 sima.sis. FTP server (Version wu-2.4.2-VR16(1) Tue Aug 24 01:16: 9) ready. User (ns.my.co.jp:(none)): umemura 331 Password required for umemura. Password: 230 User umemura logged in. ftp> cd public_html 250 CWD command successful. ftp> pwd 257 "/home/umemura/public_html" is current directory. ftp> put

ブラウザのアドレスに注目 (http://www.my.co.jp/~umemura/)

1000000000000000000000000000000000000	enter - Mercoult a	omet Explorer 9 ツール(D へルプロ	9				
44 . 49 . 112 . 41	2 J	4 2	89023.0	3	前 载	adh	
PIFUA® Thttp://www	ny.co.jp/"unemute/					· C- 1540	100
こんにちわ							

TAT							
999							
A AMPENDALS				indexe as admini		1.4.25 421	FER:
29-1 2 10 11	M * mto//w	www.my.ca.p/				3/	55 a

エ・メールサーバ

電子メールの配送プログラム(MTA)として sendmail を使いました。

windows マシンとメールのやり取りをするには、さらにpopサーバも構築しな ければなりません。残念ながら TurboLinux4.2 では、インストール時にpopサー バを構築する imap パッケージが入りません。TurboLinux Japan のサイトから imap -4.6-3.i386.rpm をダウンロードし、rpmコマンドでインストールする必要があり ます。

その後、/etc/inetd.confファイルのpop-3に関するコメントを外してinetdを起動します。

windowsマシンではメーラーoutlookにサーバ名とログイン名と パスワードを設定する必要があります。 umemura@my.co.jpの自分自身のアドレスにメールを出す。

	aronott Guribok			-02
271/2 MER	· · · · · · · · · · · · · · · · · · ·	7-AD 799200 AN70		
Outlook Sta-h			C Save bank as	
-	2##5-11 × = ∰ Outlook Today - (11)		18 1	学信日時 01710/20(1)
	「京下書き 「日前時:派かアイテム 日本	兼出人: une 件名: hent	電気: umemura®my.co.jp GC:	
1	の 逆信をい の 逆信第30アイアム の 予定表	メール受信のテストです。		
	の加速			
Ø				
Ø				
- (**)				
50億のショートカ. 仲をの561件旅	* *			*あ般 8 章 玉 8 二
A29-1 3	📴 🧊 🤐 * 🔯 🕿 📖 😂	of - Microsof		32 4 Q 139

オ.ファイルサーバ

Samba を使用することにより, windows マシンからファイルサーバとして利用 できます。windows マシンと Linux マシンのワークグループ名を一致させ、/etc /smb.conf の暗号パスワードを有効にします。Samba 用のパスワードを設定して smb サービスを起動することによりファイルサーバとして働きます。

ネットワークコンピュータns(Linuxマシン)にホームデルかり umemura と各ユーザ共有の public が 表示される。



カ.データベースサーバ

Linuxd ではデータベースとして PostgreSQL が利用できます。Linux のインスト ール時には入っていませんので、後からインストールする必要があります。

TELNET - mailiny.co.ip				
Database = mydb	1KU ~16769			
Owner	1	Relation	Туре	
umemura	mytbl		table	
mydb=> ¥d mytbl				
Table = mytbl				
Fie	Id	İ	Туре	Length
id name score		int4 char() int4		4 20 4
novdb=> select * fro idiname 1 fukuzawa yukich 2 梁 式部 3]夏目 漱石 (3 rows) mydb=>	om motbl: score 80 70 90			

キ. IPマスカレード

IPマスカレードはグローバルアドレスとプライベートアドレスの変換を行い ます。IPアドレスの有効利用に役立つほか、ローカルネットワークのファイヤ・ ウォールとしても機能します。ipchains をインストールしてネットワークインター フェイスやルーティンググの設定を行います。

サーバを含む 3 台のパソコンで 28.8kbps のモデム 1 台を使って、プロバイダに接続し、3 つの異なるホーム ページを見ることができる。



3. karacrixCONT

karacrix(エスアイ創房製)はLinux上で動作する計測制御を目的とした ソフトウェアです。これまでのようにパソコン単体でも計測制御ができますが、Linux の提供するネットワーク機能を使って、他のパソコンから遠隔操作で計測制御が可能になり ます。



次のような機能を持っています。

ア.監視パネル

センサ,アクチュエータなどのオブジェクトをグラフィカルユーザインターフェ イス(GUI)からリモートで監視、操作することができます。

イ.監視パネルCAD

監視パネルをユーザが自分で作成・編集することができます。

ウ.短期トレンドグラフ

オブジェクトのデータを計測しながら同時にグラフ表示できます。データのサン プリング周期は1~3600秒で自由に設定できます。

エ.制御プログラム

各種計測制御アプリケーションをユーザレベルで作成するための開発環境が標準 で搭載されています。プログラムからコントロールできるものは,センサ、アクチュ エータ及び計測制御プログラム自身もコントロールできます。プログラムはC言語 で作りますが、ユーザアプリケーションインターフェイスとして、各種センサやア クチュエータをオブジェクトとして扱う制御ライブラリーが用意されています。

オ.スケジュール機能

各種機器のスケジュール運転を用途に応じて設定することができます。

力.画像記録

ビデオキャプチャカード経由でカメラから画像を取り込めます。JPEG圧縮し て保存、蓄積することができます。画像データのサンプリング周期は1~3600秒の 間で自由に設定できます。エンドレス画像記録とワンウェイ画像記録が選択できま す。 キ.web 監視制御

Webブラウザからアクセスすることによりセンサ、アクチュエータ、カメラ画像や、トレンドグラフの状態を取得できます。また、各種機器のON,OFFやパラメータの設定をブラウザから行うことができます。

ク.Eメール監視制御

Eメールに取得したいオブジェクトの情報を記述して送信することによりセンサ、 アクチュエータの状態を返信メールとして取得できます。監視パネル画像やトレン ドグラフは添付ファイルとして取得できます。また、操作したいオブジェクトを指 定したメールを送信することで、各種機器をEメールで遠隔操作できます。

A.インストールについて

インストールはそれほど難しくありませんが、Linuxの操作にある程度慣れている 必要があります。画像を扱う場合は、Bt848系のチップを使ったビデオキャプチャボ ードが必要です。今回使用したビデオキャプチャボードはI・OデータのGV-VCP/ PCIです。さらにTurboLinux4.2ではシステム内にデバイスファイルがありません ので、これをエスアイ創房からいただき、設定する必要があります。(当方でも提供可能)

B.オブジェクトについて

karacrixでは監視、計測制御の対象を総称してオブジェクトと呼んでいます。 実際に監視、計測制御の対象になるものには、様々なセンサやアクチュエータが接続されることになります。

このオブジェクトには次のような種類があります。

ア. BI, BOオブジェクト

BIとは(Binary Input)の略で2値入力という意味です。BIオブジェクトは 2つの状態を持つセンサなどに使用するオブジェクトです。

- 同様に BO は(Binary Output)の略で2値出力という意味です。BOオブジェクトはリレーなどのON/OFFに使用します。
- イ.AI.i、AI.fオブジェクト

A I とは(Analog Input)の略で各種アナログセンサのA / D変換した値を 入力するときに使用します。AI.i(整数値)とAI.f(実数値)の2タイプを使 用できます。

ウ.IMG オブジェクト

IMG オブジェクトとは動画イメージを扱うオブジェクトです。CCD カメラ などからのビデオ信号をキャプチャボードで取り込み、デジタルデータ化した ものを対象とします

- エ.擬似オブジェクト 通常オブジェクトにはセンサやアクチュエータなどの具体的な機器が関連づけ られていますが、擬似オブジェクトはそれがありません。プログラムで処理され た結果などを使いたい時に結果をオブジェクトとして使用可能です。
- オ.プログラムオブジェクト

karacrixでは、制御プログラムのような実行可能なプログラムもオブ ジェクトとして扱えるように拡張されています。オブジェクトとして制御の対象 になりますが、直接外部機器と対応づけられていません。



オブジェクトの概念

C.アナログデジタルIOボードについて

karacrixをインストールしたサーバとRS232Cを通してセンサのデータを 送ったり、アクチュエータに信号を出力するインターフェイスボードとして、アナログデ ジタルI/OボードKCXH-IOB10A(エスアイ創房製)を使います。このボードに は温度センサ(LM35DZ)、湿度センサ、光りセンサ(Cds)が載っていてすぐに使えます。 このボードの入出力は次のようになっています。

ア.接点入力6ビット

- イ.フォトMOS出力3ビット
- ウ.アナログ入力 4 c h。 8 c h まで使用可。

12 ビットA/Dコンバータで入力レンジは0~4095mV 1 ビットの分解能が1 mV

出力が3ビットと少ないのが残念です。もっと多くの出力が必要な時はKCXH-I OB20Aを購入するか、自作するしかありません。

このボードにはPICが使われていて、RS232Cを通してサーバと間で各オブ ジェクトと対応しているセンサなどのデータのやり取りをしています。実際に制御を行 う場合にはサーバ側で標準で登録されている「KCXH-IOB10A」の計測制御用通 信プログラムを起動します。



4. karacrixによるオートメーションサーバの構築事例

オートメーションサーバの事例として制御対象物をパンの生地に直接交流100Vをかけ て焼くパン焼き器としました。

この題材はよく理科関係の本(物理がおもしろい ガリレオ工房著など)で見うけられ、 すでに7年ほど前に、当時福岡工業高校に勤務していた桑畑義行先生(現釜石工業高校)に よりポケコンで制御されています。

I/Oボードの少ない出力ビットで制御ができて、電流センサ、温度センサなどを用いれ ば、その出力変化をグラフ化でき、カメラで監視した場合、パンの生地に短時間で変化が表 れるといったkaracrixの多くの機能を使って大変興味のある題材だと思いました。







A.ハードウェア

電流センサによりパンの生地を流れる交流電流を、白金測温抵抗体でパン内部の温度を 測定するものとします。また、サーバからの指令により電源がリレーでON/OFFする ものとします。

したがって、電流センサ回路、白金測温抵抗体回路、リレー回路の3つの回路を自作す ることとしました。

ア.電流センサ回路

オペアンプを使った絶対値回路で電流センサからの出力を全波整流した後に、増幅してコンデンサで平滑しています。





電流センサ CTL-6-S32-8F-CL (ユー・アール・ディー製)

イ.白金測温抵抗体回路

1段目のオペアンプで白金測温抵抗体の温度による抵抗変化を電圧に変換した後で 10mV/になるように2段目で増幅度を調整します。





E-52-P6F(オムロン製)

ウ.リレー回路



B.ソフトウェア

webブラウザや電子メールで制御して、各センサからの出力を観測できますが、専用の制御プログラムを作り、できるだけ自動的に動くようにしました。

karacrixにはkcxライブラリといってC言語から外部装置と各種データの入出 力を行うための関数が用意されていますので、これを利用しました。

このパン焼き制御プログラム(オブジェクト名ctl02)は電子メールにより起動される と、まずアナログデジタルI/OボードKCXH-IOB10Aの制御通信プログラム(ct 110)を起動させます。そして、パン焼き器の電源(Bo01)をONにします。電流センサの 電流値(aif02)が0.1[A]より下がった時、あるいは電源ONから15分が経過したときに 自動的に電源をOFFにします。それから、パンが焼きあがったことを電子メールで送信します。 メールには焼き始め時刻と焼きあがり時刻と所要時間が書かれています。また、このメールに は焼き上がりまでのパン内部の温度と流れた電流がトレンドグラフで添付されます。

焼きあがりを知らせ、トレンドグラフを添付するメールはkcxライブラリの kcxsnd_email_texts_append()関数を使います。

```
0002: /* 千厩東のパン屋さん制御プログラム
                                                */
                                                */
0003: /*
                     2001.11.7
0004: /*
                             by umemura
                                               */
0006:
0007: #include <karacrix.h>
0008:
0009: main( argc, argv )
0010: int
            argc;
           *argv[];
0011: char
0012: {
0013:
       struct tm start time.end time:
0014:
       int obj_bo01,obj_aif02,obj_ctl10,i,minutes,seconds;
0015:
        float
             fdata:
0016:
        /* KRACRIX ライブラリの初期化(先頭に必須) */
0017:
0018:
       kcxinit( argc, argv );
0019:
       obj_bo01 = kcxobj_open("bo01");
0020:
       obj_aif02 = kcxobj_open("aif02");
0021:
       obj_ctl10 = kcxobj_open("ctl10");
0022:
        kcxobj_stat_iwt(obj_ctl10,1); /* 通信制御プログラム起動 */
sleep(1); /* 時間稼ぎ */
0023 \cdot
0024:
        kcxobj_sndistat_tokcx(obj_bo01,1); /* 電源スイッチon */
0025:
                                      /* 開始時刻の取得 */
0026:
        kcxtim_whattime(&start_time);
0027:
0028:
       for(i=0; i < 90; i++){
                kcxobj_stat_frd(obj_aif02, &fdata); /* 電流値取得 */
0029:
0030:
                if(i > 5 \&\& fdata < 0.1) break;
                sleep(10);
0031:
0032:
        }
0033:
0034:
        kcxtim_whattime(&end_time);
                                  /* 終了時刻の取得 */
        kcxobj_sndistat_tokcx(obj_bo01,0); /* 電源スイッチoff */
sleep(1); /* 時間稼ぎ */
0035:
0036:
                                 /* 通信制御プログラム停止 */
0037:
        kcxobj_stat_iwt(obj_ctl10,0);
0038:
0039:
       jikan(start_time, end_time, &minutes, &seconds); /* 焼き上がり所要時間の計算 */
0040:
       report_mail(start_time, end_time, minutes, seconds);
0041:
0042: }
0044: /* 焼き上がりを所要時間を計算する関数
                                                    */
0046: jikan(stime,e_time,min,sec)
0047: struct tm s_time,e_time;
0048: int
         *min,*sec;
0049: {
0050:
       if (e time.tm min < s time.tm min ) *min = 60 - s time.tm min + e time.tm min;
0051:
          else *min = e_time.tm_min - s_time.tm_min;
0052:
       if (e_time.tm_sec < s_time.tm_sec ) { *sec = 60 - s_time.tm_sec + e_time.tm_sec;
0053:
                                         *\min = *\min - 1; \}
0054:
          else *min = e_time.tm_sec - s_time.tm_sec;
0055: }
```

0056: /***	******					
0057./*	焼き上がったことを知らせるメールを送信する関数 */					
0058 /**						
0059: report_mail(s_time, e_time, min, sec)						
0060: stru	act tm s time, e time;					
0061: int	min.sec;					
0062: {						
0063:`	char toaddr[64];					
0064:	char ccaddr[64];					
0065:	char bccaddr[64]:					
0066:	char subject[64]:					
0067:	char *mtext[10]:					
0068:	int i.mtexts:					
0069:	char appendtext[64]:					
0070:						
0071:	strcpy(toaddr, "umemura@my.co.jp");					
0072:	strcpy(ccaddr, "");					
0073:	strepy(becaddr, "");					
0074:	strcpy(subject, "パンが焼き上がりました。"):					
0075:						
0076:	mtexts=6:					
0077:	for(i=0:i < mtexts:i++)					
0078:	mtext[i] = (char *)malloc(64+1):					
0079:						
0080:	strcpv(mtext[0], "* * * パンが焼き上がったことを報告します。"):					
0081:	sprintf(mtext[1], "開始時間 %d 時 %d 分 %d 秒",s time.tm hour, s time.tm min, s time.tm sec);					
0082:	sprintf(mtext[2], "終了時間 %d 時 %d 分 %d 秒",e time.tm hour, e time.tm min, e time.tm sec);					
0083:	sprintf(mtext[3], "所要時間 %d分 %d秒",min.sec):					
0084:	strcpv(mtext[4], "このメールには温度、電流のグラフが添付されています。"):					
0085:	strcpv(mtext[5], "そちらもご覧ください。"):					
0086:						
0087:	strcpy(appendtext, "tre 1");					
0088:	kcxsnd email texts append(toaddr,ccaddr,bccaddr,subject,mtext,mtexts,appendtext):					
0089: }						
0090:						

C.運 用

ア.PPPサーバの構築(Linux側の設定)

ここでは、できるだけ遠隔操作の雰囲気をだすためLinuxマシンにPPPサーバ を構築してモデムを使い、windowsマシンからダイヤルアップでサーバと交信でき るようにします。



よく本に書かれているLinuxでのPPP設定はLinuxマシンからのダイヤルアッ プでプロバイダに接続してホームページを見るというクライアントの設定です。(ほとんどの解説書 がそうである。)

しかし、次のように設定するとLinuxではモデムをとおしてクライアントに対して各種サーバの機能を提供できます。

そのためには、Linuxマシンでmgettyというソフトを使います。モデム自身着信機能を 持っていますが、この場合mgettyの判断で着信させるため、モデム自身が自動着信しないよう にATコマンドで設定します。

A T S 0 = 0ただし、通常は購入時には自動着信しない設定になっています。 mgettyの設定 /etc/mgetty+sendfax/mgetty.configのファイルに次の3行を追加します。 data-only У ttyS1 port init-chat "" ATS0=000E1L0M0 OK mgettyがブート時に起動し、mgettyがプログラム動作を終了したら、 再起動するように設定。 /etc/inittab ファイルに mg:2345:respawn:/sbin/mgetty ttyS1 を追加。ttyS1 はシリアルポート2 を表します。(シリアルポート1は karacrix の I/O ボードが使用しているので、シリアルポート2にモデムを接続した。) ログイン時の認証に PAP 認証を設定。 /etc/mgetty+sendfax/login.config ファイルで /AutoPPP/ - a_ppp /usr/sbin/pppd auth login -chap +pap kdebug7 debug と設定。 /etc/ppp/pap-secrets ファイルに接続されるユーザ名、パスワード、接続許可する IP アドレスを記述。特に指定しない時は次のように記述する。 * /etc/ppp/options ファイルに次の7行を記述。192.168.0.1:192.168.0.3 はローカルIPア ドレス (サーバの I P アドレス) とリモート I P アドレス (接続してきたクライアン トに割り当てられる)です。 lock - detach asyncmap 0 modem crtscts 192.168.0.1:192.168.0.3 proxvarp イ.オブジェクトの登録(karacrix側の設定) 今回使用するオブジェクトを登録します。 オブジェクト名 電源スイッチ b o 0 1 aif01 温度センサ 下限值 0 上限値150 aif02 下限值0[A] 上限值 6[A] 電流センサ ウ.監視パネルCADによる監視パネルの作成(karacrix側の設定) ユーザが画面を見ながら制御が容易に行えるように監視パネルを専用のCADで作成 します。電源スイッチ、温度値、電流値、カメラ画像は対応するオブジェクトを貼り付け

ます。

監視パネル CAD でパン焼き器の GUI を作成します。



監視パネルで見ると、CADで貼り付けられたオブジェ クトには画像や数値が表示されます。



エ.パン焼き制御プログラムの起動(windowsマシンの動作)

karacrixのメールコマンドにより、パン焼き制御プログラムを起動するメー ルを作成しサーバに送信します。メールの内容はわずか2行からなります。 @kcxusr umemura rupin3はkaracrixのユーザ認証で、あらかじめ登録して あるユーザ名とパスワードを書きます。

@kcxset obj ctl02 on はパン焼き制御プログラム(ctl02)を起動するためのものです。

○ パンを焼し ファイル(E)	いてください。- メッセージ(テキスト形式)
0 20xyt	ージはまだ送信されていません。オプション
宛先。	karacrix
a start of the second start of	
件名(J)	ノベンを焼いてください。
件名(小 @kcxusr un @kcxset ob	パンを焼いてください。 nemura rupin3 j ct102 on

オ.webブラウザで監視(windowsマシンの動作)

ブラウザで監視パネルやトレンドグラフをサーバから取得して、パン焼き器のカメラ 画像や電流、温度の変化などを見ることができます。



カ.焼きあがり通知メールの受信(windowsマシンの動作)

パンが焼きあがると制御プログラムからトレンドグラフが添付されたメールが届きま す。

焼きあがりを知らせるメールを受信。



メールに添付されてきたトレンドグラフを開く。



5.おわりに

Linux上にkaracrixでオートメーションサーバを構築することにより、ネット ワーク機能を利用した、まさにインターネット時代にふさわしい制御を生徒に体験させること ができます。そして、今後はこのような形態の制御技術が重要視されて行くと思われます。

また、県内の学校においても教育情報ネットワークが整備されるなどネットワーク機能が身 近で利用できる状況となってきました。単に情報のやり取りだけに使用するのではなく工業高 校(学科)的視点から学校内・学校間の利用方法を考えてみたいと思います。

(参考文献)

1.TurboLinux でつくるインターネットサーバ	工学社
2.TurboLinux でインターネットサーバを作る本	日刊工業新聞社
3.センサ応用回路の設計・製作	CQ 出版
4 . k a r a c r i x ユーザーズマニュアル	エスアイ創房

* この資料にはkaracrixのマニュアル等から一部、引用、転載されています。 快く御許可下さいました、株式会社エスアイ創房様に感謝いたします。 なお、株式会社エスアイ創房のホームページアドレスは下記のとおりです。

http://www.karacrix.jp

