小型ネットワーク・アダプタで
何でもネットワーク化計画！

シリアル-イーサネット変換器 XPortの試用リポート

〈後編〉ウェブ・サーバ機能と汎用I/Oの使い方

小野 泰正
Yasumasa Ono

■ はじめに

XPortは，前回紹介したシリアル・イーサネット変換器として使えるほか，小型のウェブ・サーバとしても使用できます．今回は，ウェブ・サーバ機能の使い方と，XPortがもつ汎用I/Oとウェブ・サーバ機能を組み合わせた事例を主に紹介します．

ウェブ・サーバ機能の利用

ウェブ・サーバ機能を単独で使用する用途としては，XPortを載せる装置の情報をXPortそのものに登録してしまうことが考えられます．例えば数年後に装置を使うとき，紙の説明書を紛失してしまっている可能性があるでしょう．そこで保険という意味で，XPortに説明書やメンテナンス情報を入れておくということです．ほかの機能を使わずとも，この目的だけでXPortを使うことも考えられます．

■ ウェブ・ページのデータを作る

● 元データの作成

簡単なウェブ・ページを制作してみましょう．制作したウェブ・ページのHTMLファイルを**リスト1**に示します．HTMLファイルは，テキスト・エディタなどを使ってテキスト・ファイルとして作成した後，拡張子を**.html**に変えると簡単に作成できます．

リスト1のHTMLファイルでは，JPEG形式の画像ファイルを一つ使っていますが，このファイルの大きさはHTMLファイルとの合計が64Kバイト以下になるよう調整してください．なお，それぞれのファイル名は以下のようにしました．

〈リスト1〉簡単なウェブ・ページのHTMLファイル

```
1 <TITLE>Test Home Page</TITLE>
2
3 <H2>○×研究室直販</H2>
4 <HR>
5 <H3>7/1 XPortを入手。応用製品にかかります！</H3>
6 <IMG SRC="XPort.jpg">
```

- HTMLファイル：**Test.html**
- JPEGファイル：**XPort.jpg**

● ファイル形式の変換

XPortにウェブ・ページのデータを登録するには，**.cob**というファイル形式に変換する必要があります．まず作業用のフォルダ(ここでは**C:¥xport**)を作成して，そこに**C:¥Program Files¥Lantronix¥XPort Installer3.2**フォルダ内にある**web2cob.exe**と**mimetype.ini**という二つのファイルをコピーします．これらは，ファイル形式を変換するためのプログラムと設定データです．

次に作業用フォルダの下に，変換したいファイルを格納するフォルダ(ここでは**C:¥xport¥web**)を作り，先ほど作成したウェブ・ページのファイルをコピーします．

変換プログラムはコマンド・プロンプトから実行します．先ほどプログラムを保存したフォルダに移動して，以下のようにプログラムを起動します．

```
web2cob  /o Test.cob  /d web
```

/oオプションの後にあるのは出力ファイル名，**/d**オプションの後にあるのは変換対象となるフォルダ名で

〈図19〉web2cobを実行したようす

■ プログラムの入手方法　この記事の関連プログラムは@niftyのFDEVICEのデータ・ライブラリ#7「CQ出版トランジスタ技術提供データ」と小社ホーム・ページに登録します．(編集部) http://www.cqpub.co.jp/

す．プログラムを実行すると，図19のように対象フォルダ内のファイルをまとめたTest.cobというファイルを生成したという表示が出て，プロンプトに戻ります．

■ ウェブ・ページのデータを登録する

● ツールを使ってウェブ・ファイルを登録する

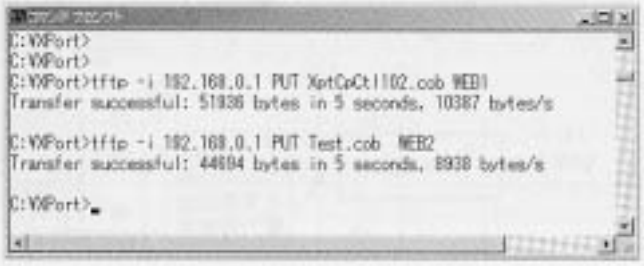
XPortインストーラを起動し，ツール・バーの[Search]ボタンを押して動作中のXPortを検索します．検索で見つかったXPortを選択すると，ウィンドウが図20のようになります．ここで[Upgrade]ボタンを押すと図21のダイアログ・ボックスが出て，ウェブ・ファイルを登録できるようになります．

図21をみるとわかるように，ウェブ・ページの登録場所は6区画あります．出荷時状態では，WebPage6にシリアル-イーサネット変換関連の設定ページが登録されています．なお，残念ながらXPortインストーラ3.2では，現在何が登録されているかを知る手段が用意されていません．

ここでは，先ほど作成したウェブ・ページの.cobファイルをWebPage1に登録してみます．WebPage1の欄をクリックすると，登録するファイルを指定できます．図21のようにファイル名をC:¥xport¥Test.cobとしたあと[OK]ボタンを押すとファイルが登録され，XPortが再起動します．

〈図20〉検索で見つかったXPortを選択したようす

〈図22〉tftpクライアント機能を使ってファイルを登録する



● ツールを使わないでウェブ・ファイルを登録する

tftpクライアント機能を備えるマシンからであれば，XPortインストーラを使わなくてもウェブ・ファイルを登録できます．Windowsでは，Windows 2000，XP，NT4.0にtftpクライアント機能が装備されています．

tftpクライアント機能は，コマンド・プロンプトから以下のように起動します

```
tftp -i 192.168.0.1 PUT XptCpl102
.cob WEB1
```

ただし，-i：バイナリ・モード指定，192.168.0.1：転送先XPortのIPアドレス(任意)，PUT：書き込みモード，XptCpl102.cob：登録するファイル名(任意)，WEB1：登録先(任意)

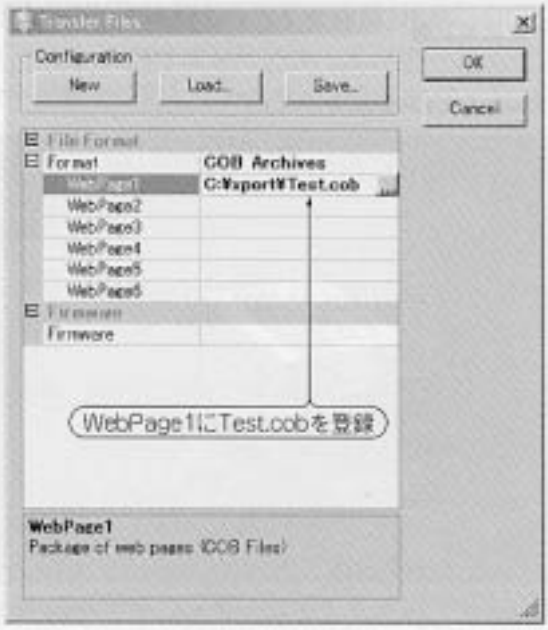
登録先は，ウェブ・ページを登録するならWEB1～WEB6を，ファームウェアを登録するならX1を指定します．実行例を図22に示します．

この登録方法とweb2cob.exeを使えば，クライアント・マシン上のプログラムからXPort内のウェブ・ファイルを自動的に更新することも可能です．

■ ウェブ・ページを表示する

XPortに登録したウェブ・ページをブラウザから呼び出してみます．XPortのIPアドレスが192.168.0.1なら，ブラウザのアドレス入力欄にhttp://192.168.0.1/Test.htmlと入力してページを開きます．なお，ブラウザから呼び出す対象は.cobに変換する前のファイル名です．表示例を図23に示します．

〈図21〉ファイル登録を行うダイアログ・ボックス



■ XPortに登録できる最大データ量は384Kバイト

一つの区画は64Kバイト固定なので，6区画全部を使用した場合64Kバイト×6＝384Kバイトのデータを登録できます．ただし，64Kバイトより大きなサイズの.cobファイルは登録できないので，その場合は分割して.cobファイルを作成してください．区画が分かれていても，実際には全区画を一つのフォルダとして扱うので，別区画に登録したファイルも直接呼び出せます．なお，同じ名前のファイルを別区画に登録した場合は，区画番号が若いほうのファイルだけが呼び出し可能です．

汎用I/O機能の利用

XPortには，フロー制御端子としても使用可能な3本の汎用I/O端子が用意されています．この汎用I/Oは，図2(c)や図2(d)の用途で使用できます．ここでは汎用I/Oの使用例として，ネットワーク接続型赤外線センサを作ってみます．

■ ネットワーク接続型赤外線センサの製作

● 製作したボード

回路図を図24に，外観を写真6に示します．前編で製作したシリアル-イーサネット変換ボードの電源部を作り直し，ここにリレーと赤外線センサを取りつけました．なお図24では，EIA-232ドライバ/レシーバ部を省略しています．

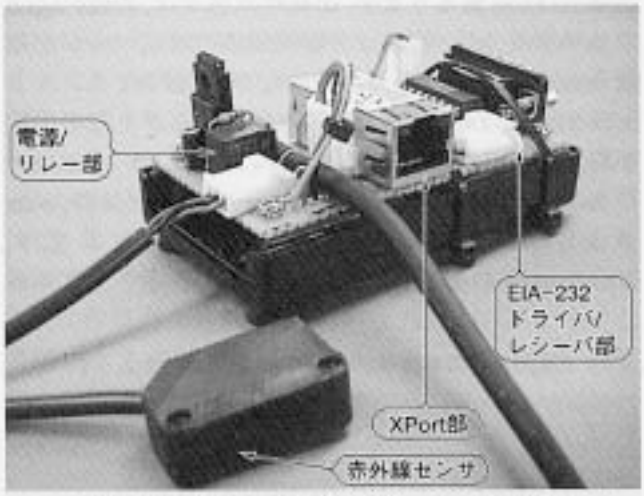
〈図23〉作成したウェブ・ページの表示例

センサの前を人が通るとリレーがONになり，CP2の入力がLレベルになります．

● 使用した部品

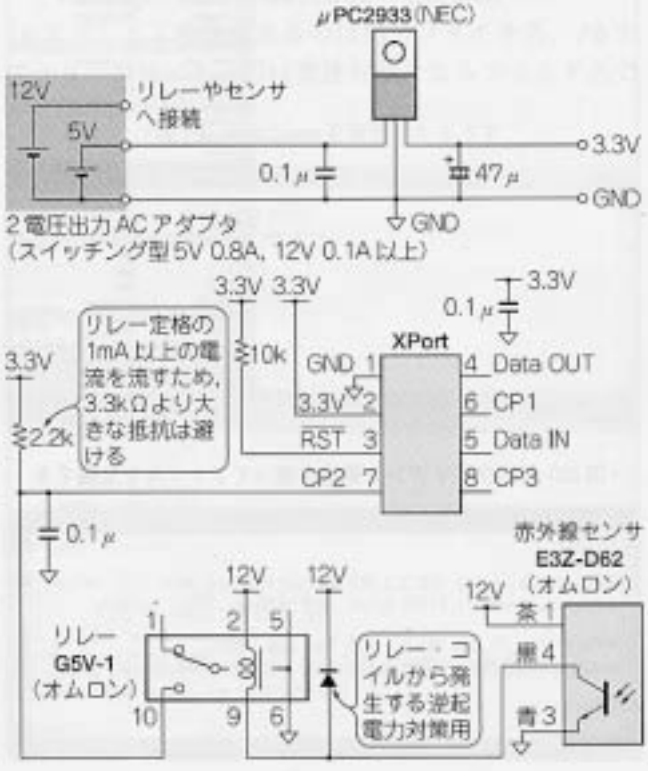
赤外線センサは，検出距離1mの拡散反射型赤外線センサE3Z-D62(オムロン)を使用しました．この方式のセンサには，検出距離が最大6mもあって自動車の通過も検出可能なQMT42VN6DX(Banner社)など，さまざまな製品があります．用途に合わせて選んでください．

また，入手しやすさから機械式のリレー(G5V-1，オムロン)を使用しましたが，動作頻度が高い場合はフォト・カプラを使うなど，ほかの選択肢もあります．

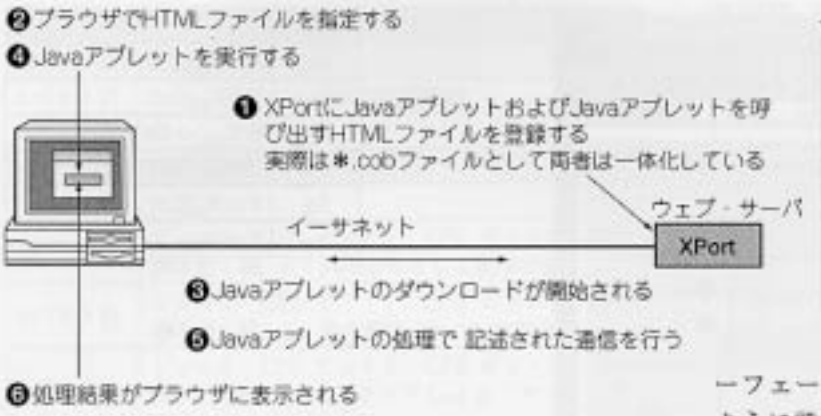


〈写真6〉製作した赤外線センサ・ボードの外観

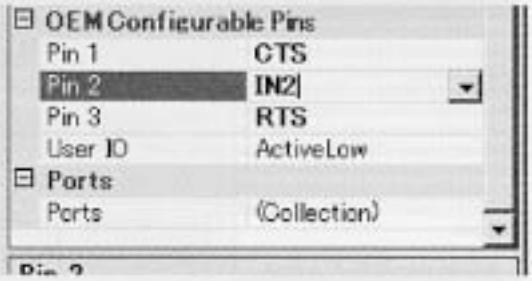
〈図24〉製作した赤外線センサ・ボードの回路図



〈図25〉Javaアプレット実行の流れ



〈図26〉赤外線センサ・ボードを接続するためCP2を入力に設定する



■ 汎用I/Oの状態をみるソフトウェア

XPortの添付CD-ROMには，汎用I/Oを制御するJavaアプレットのサンプルが収録されています．このJavaアプレットは，図25のように実行されます．これを使って，赤外線センサの状態をモニタしてみます．

● Javaアプレットの登録

汎用I/O制御のJavaアプレットのサンプルは，Javaアプレットと呼び出し用のHTMLファイルが一体になった.cobファイル形式になっています．まずはこのファイルをXPortに登録します．

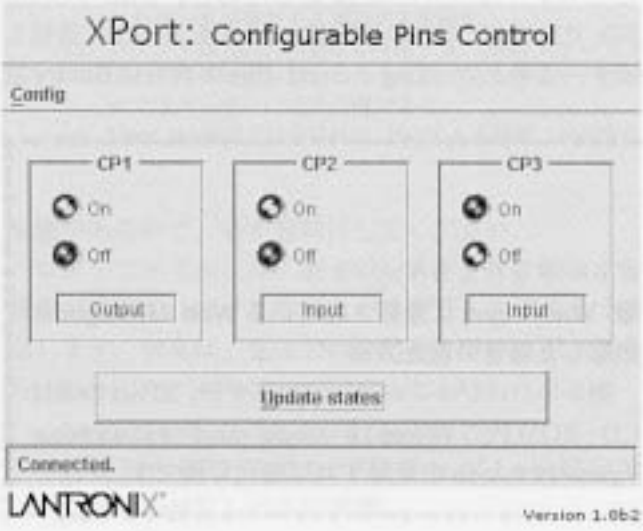
先ほど説明したウェブ・ファイルの登録方法にしたがって，添付CD-ROMの¥Sample Code and Solutionsフォルダに収録されているXptCpl102.cobを，任意のWebPageに登録します．

● XPortの設定

汎用I/O(CP1～CP3)の入力/出力は，図20の右下あるOEM Configurable Pinsの部分で，各ピンごとにINまたはOUTに設定します．ツール・バーの[Update]ボタンを押すと，設定が更新されます．

図24の回路では，CP1とCP3はシリアル・インターフェースのフロー制御に使っているので，図26のように設定します．

〈図27〉Javaアプレットのサンプルを実行したようす(CP1を汎用I/O入力，CP2～3を汎用I/O出力に設定した場合)



● Javaアプレットの呼び出し

XPortのIPアドレスが192.168.0.1なら，ブラウザのアドレス入力欄にhttp://192.168.0.1/cp_ctl.htmlと入力して，XPortに登録したJavaアプレットを呼び出します．このJavaアプレットを実行するには，ブラウザにJavaプラグイン1.4.xが登録されている必要があります．Javaプラグインが登録されていない場合，ブラウザがサン・マイクロシステムズ社のウェブ・サイトに自動的にアクセスして，Javaプラグインのダウンロードを要求してきます．

XPortに登録したJavaアプレットが実行されると，図27の汎用I/Oの入力状態確認と出力コントロールの画面が表示されます．

[Update states]ボタンを押すと，汎用I/OのON/OFF状態が更新されます．またConfig→Set Configを選択すると，画面の自動更新間隔を100ms単位で設定できます．

● 汎用I/O機能はシリアル-イーサネット変換機能と同時に使用できる

汎用I/O機能とシリアル-イーサネット変換機能を同時に動かしているようすを図28に示します．この画面ではCP2を汎用I/Oとして使用(状態はON)しており，CP1とCP3はシリアル・インターフェースのフロー制御で使用しています．

■ 汎用I/Oの電気的特性と初期状態

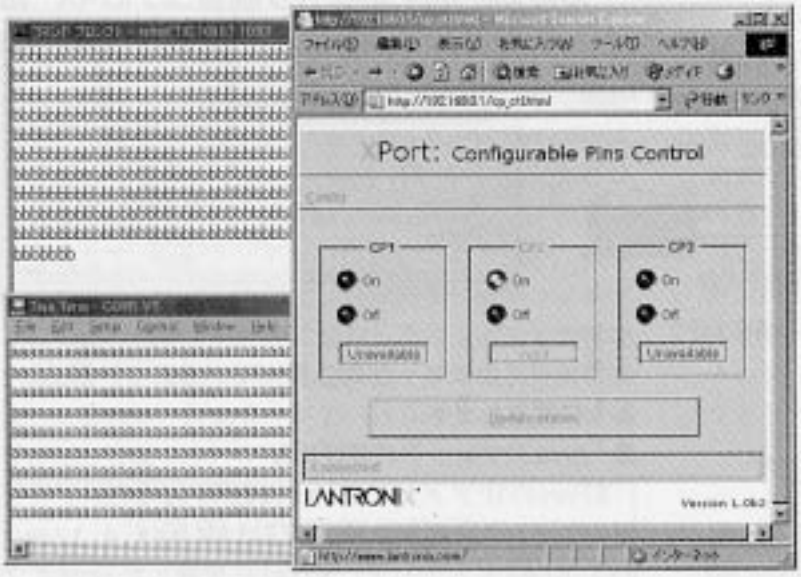
汎用I/Oの電気的特性を表3に，XPort起動時の初期状態を表4に示します．入力に設定した場合は内部プルアップが有効になり，端子解放時は常にHレベルになります．

■ ユーザ・プログラムから汎用I/Oを利用する方法

● 汎用I/Oを制御するためのメッセージ

XPortの汎用I/Oは，ほかのマシンからTCP接続してメッセージを送受信することで制御できます．TCP接続で汎用I/Oを制御するためのメッセージ内

〈図28〉汎用I/O機能とシリアル－イーサネット変換機能は同時に使用できる



〈表3〉汎用I/Oの電気的特性

項目	値
V_{OL}	0.4 $V_{max.}$
V_{OH}	2.4 $V_{min.}$
V_{IL}	0 $V_{min.}$～0.8 $V_{max.}$
V_{IH}	2 $V_{min.}$～3.3 $V_{max.}$
I_{OL}	8 $mA_{max.}$
I_{OH}	8 $mA_{max.}$

〈表4〉XPort起動時の汎用I/Oの初期状態

設定	ピン電圧	状態
アクティブLow入力	3.3 V付近	非アクティブ
アクティブHigh入力	3.3 V付近	アクティブ
アクティブLow出力	3.3 V付近	非アクティブ
アクティブHigh出力	0 V付近	非アクティブ

容を表5に示します．

● クライアント側プログラムの記述言語

XPortへ接続するクライアント側の通信プログラムは，どの記述言語を使ってもかまいません．しかし，XPort内蔵のウェブ・サーバに登録してダウンロード，実行，表示するには，Javaで記述するのが良いでしょう．前述のサンプル・プログラムでも，ブラウザで実行するJavaアプレット形式をとっています．

● 簡単な動作確認ならターミナル・ソフトウェアも使える

クライアント側のプログラムを作らなくても，汎用I/Oの簡単な動作確認ならターミナル・ソフトでも可能です．ここではフリー・ソフトウェアTera Term Pro Ver.2.3を使いました．

〈図29〉Tera Termから汎用I/Oを操作するときの接続設定

まず適当なバイナリ・エディタを使って，内容を0x01とした1バイトのコマンド・ファイルを作ります．次にTera Term Proを起動して図29のようにXPortに接続します．メニューからFile→Logを選択してログ・ファイルを設定したあと，File→Send Fileで先ほどのコマンド・ファイルをXPortに送信します．このとき，LogとSend Fileの両方にBinary設

■ XPortへのファイル登録に関する補足

● XPortのファームウェアは更新できる

図20の右上から2行目に，今使っているファームウェアのバージョン表示があります．図20では1.20と表示されており，バージョンが1.20であることがわかります．今後新しいファームウェアがリリースされた場合は，図21のFirmware欄でアップグレードを行えます．

● WebPage6に登録されているWeb Managerを削除した場合の復元方法

細かいバージョンの違いが出ますが，XPortの添付CD-ROM内の¥Sample Code and Solutions¥genw325.cobを登録すれば復元可能です．

〈表5〉TCP接続で汎用I/Oを制御するためのメッセージ

ポート番号　30704(0x77f0)[1]	
XPortへの設定状態確認コマンド	
1バイト目	0x01：汎用I/Oの設定状態確認
2バイト目	なし
XPortからの設定状態確認レスポンス	
1バイト目	0x01
2バイト目	ビット2：CP3 ビット1：CP2 ビット0：CP1 読み出し値　1：汎用I/Oとして使用可能
3バイト目	ビット2：CP3 ビット1：CP2 ビット0：CP1 読み出し値　1：出力設定　0：入力設定
4バイト目	ビット2：CP3 ビット1：CP2 ビット0：CP1 読み出し値　1：アクティブLow　0：アクティブHigh
5バイト目	ビット2：CP3 ビット1：CP2 ビット0：CP1 読み出し値　1：アクティブ　0：非アクティブ ビット7：Javaアプレットでセーブされたデータの有無[2]
6バイト目	なし
XPortへの出力設定コマンド	
1バイト目	0x02：出力設定
2バイト目	ビット2：CP3 ビット1：CP2 ビット0：CP1 書き込み値　1：アクティブ　0：非アクティブ
3バイト目	なし
XPortからの出力設定レスポンス	
1バイト目	0x02
2バイト目	ビット2：CP3 ビット1：CP2 ビット0：CP1 読み出し値　1：アクティブ　0：非アクティブ
3バイト目	なし
XPortへの状態確認コマンド	
1バイト目	0x03：入力状態の確認
2バイト目	なし
XPortからの状態確認レスポンス	
1バイト目	0x03
2バイト目	ビット2：CP3 ビット1：CP2 ビット0：CP1 読み出し値　1：アクティブ　0：非アクティブ
3バイト目	なし

注▶ (1) ポート番号30704でソケット接続時，XPortは0xFF 0xFB 0x01 0xFF 0xFB 0x03 0xA5の7バイトのデータを無条件に出力する．これはファームウェア1.20の場合であり，メーカによると今後のファームウェアではこの動作をどうするかは未定である．
(2) このバイトのビット7は，Javaアプレットでセーブされたデータが存在する場合に1を示す．
(3) コマンドは0x04，0x05も存在する．Javaアプレットでのデータのセーブ，ロードに関するものだが，現時点ではユーザーには非公開である．
(4) XPortは1接続だけ受け付け，後からくる接続は拒絶する．

定義があるので，必ず有効にしてください．

ログ・ファイルには，表5のレスポンスに該当する情報が保存されます．内容はバイナリ・エディタで確認します．例えば，先ほどの赤外線センサ・ボードなら，0x01コマンドに対するレスポンスは，

```
0x01 0x02 0x00 0x02 0x05
```

ただし，CP2は入力/アクティブLow設定で，CP2端子がHレベルの状態

〈リスト3〉汎用I/Oの状態表示を行うJavaアプレットを呼び出すHTMLファイル

```
<HTML>
<BODY>
<CENTER>
<APPLET code="SendO1App.class" width="300" height="150">
</APPLET>
</CENTER>
</BODY>
</HTML>
```

〈図30〉PATHの設定からコンパイルまでを実行したようす

となります．

JavaアプレットでXPortと通信する

■ 汎用I/Oを制御する

先ほど紹介した赤外線センサの例のように，XPortのウェブ・サーバにJavaアプレットを置けば，汎用I/Oとブラウザ操作の連携が可能です．ここでは表5で示したコマンドを発行する簡単なJavaアプレットを作成します．

● Javaコンパイラの入手と設定

今回はサン・マイクロシステムズのウェブ・サイトから入手しました．http://java.sun.com/j2se/1.4.1/download.htmlにアクセスして，J2SE v1.4.1_03 SDK(Windows用)をダウンロードします．

SDKをインストールしたら，SDKをインストールしたフォルダ下の¥binフォルダに対してパスを設定してください．例えばSDKをC:¥j2sdk1.4.1にインストールしたなら，以下のように設定します．

〈リスト2〉汎用I/Oの状態表示を行うJavaアプレット

```
//
//  ファイル名  Send01App.java  － コマンド0x01 発行  アプレット －
//
import java.net.*;          // ネットワークのパッケージの全てのクラスをロード
import java.io.*;           // 入出力のパッケージの全てのクラスをロード
import java.applet.Applet;  // アプレットクラスをロード
import java.awt.Graphics;   // java.awtパッケージのGraphicsクラスをロード

public class Send01App extends Applet {                    // File名と一致させる
   public void paint(Graphics g) {                           // paintメソッド
      String str =null;   int res=0, i=0, flag=0;
      g.drawRect( 0,0,getSize().width-1,getSize().height-1); // 箱の枠線を書く

      // エラー発生時にcatch以下を実行する前提で処理を開始
      try{
         Socket theSock=new Socket("192.168.0.1", 10704); // XPortへ接続
         g.drawString("ソケット 接続完了",50,20);

         // 出力ストリーム と 入力ストリーム を設定
         DataOutputStream sockout=new DataOutputStream(theSock.getOutputStream());
         DataInputStream  sockin =new DataInputStream(theSock.getInputStream());
         g.drawString( "入出力ストリーム 設定完了",50,40);

         sockout.writeByte(0x01); sockout.flush();          // コマンド0x01を送信
         g.drawString( "コマンド0x01 発行完了",50,60);

         do{
            res=sockin.readByte();                          // 最初のバイトを受信
            if( (res&0xff)==0xff ){                         // 最初のバイトを確認し、
               for(i=1;i<7;++i)                             // 0xffならば、7バイト
                  res = sockin.readByte();                  // 読み出すのみ
               g.drawString( "非レスポンスの読み出し完了",50,80);
            }
            if( (res&0xff)==0x01 ){                         // コマンド0x01
               for(i=1;i<5;++i){                            // のレスポンスの
                  res = sockin.readByte();                  // 読み出しを行い、
                  str = "0"+(byte)(res & 0x07);             //
                  g.drawString( str,50+i*25,100);           // 表示する
               }
               flag = 1;
            }
            if( (res&0xff)==0x02 || (res&0xff)==0x03 ){     // コマンド0x02,03
               res = sockin.readByte();                     // 読み出しを行い、
               str = "0"+(byte)(res & 0x07);                //
               g.drawString( str,75,100);                   // 表示する
               flag = 1;
            }
         } while(flag==0);
         g.drawString( "レスポンス 受信完了",50,120);
         sockin.close();  sockout.close();  theSock.close(); // ソケット他をClose
      }
      catch(Exception e){                                   // エラー発生時の処理
         g.drawString( "エラー発生！",200,30);
      }
   }
}
```

```
path C:¥j2sdk1.4.1¥bin
```

● 汎用I/Oの状態表示を行うプログラムの制作

リスト2は，XPortに対してコマンド0x01を発行してレスポンスを表示する，ソケット接続のクライアント・プログラム(Javaアプレット)です．表5と対応させながら読み進めると，内容が理解できると思います．またリスト3は，リスト2のJavaアプレットの呼び出すためのHTMLファイルです．

Javaアプレットのソース・リストを作成したら，そのファイルをコンパイルします．コンパイルは，以下のコマンドで行います．

```
javac Send01App.java
```

エラーがないなら，Send01App.classというファイルが生成され，何も表示せずプロンプトに戻ります．

ここまでの操作を図30に示します．なお，生成したJavaアプレットの実行には，前述のJavaプラグイン1.4.xが必要です．

Javaアプレットは本来init，start…destroyの各メソッドから構成されますが，リスト2はpaintという描画のためのメソッド内に処理をまとめ，エラー処理も簡略化しています．本格的にJavaアプレットを作る場合は，Javaの解説書などを参考にしてください．

● XPortにファイルを登録して呼び出す

先ほど説明したウェブ・ファイルの登録方法にしたがって.cobファイルを生成し，任意のWebPageに登録します．そして，Javaアプレット呼び出し用のHTMLファイルをブラウザで表示します．

XPortのIPアドレスが192.168.0.1なら，ブラウザのアドレス入力欄にhttp://192.168.0.1/Send01App.htmlと入力します．すると図31の画面が現れ，先ほど説明したコマンド0x01に対するレスポンスの2バイト目以降(0x02 0x00 0x02 0x05)が表示されます．

なお，リスト2のコマンド送信部を変更すれば，汎用I/Oの出力の状態変更なども行えます．

■ シリアル-イーサネット変換をJava経由で使う

リスト2のJavaアプレットを改造すれば，図32のようにシリアル・ポートのデータをブラウザに表示させることも可能です．製作した赤外線センサ・ボードのシリアル・ポートをパソコンのCOM1に接続して，シリアル・ポートとブラウザ間で通信する例を紹介します．

〈図31〉制作した汎用I/Oの状態表示用Javaアプレットを実行したようす

リスト2の変更部分をリスト4に示します．先ほどの汎用I/Oの場合と同じくコンパイルしたあと，.cobファイルを生成してXPortに登録します．

ターミナル・ソフトウェアからCOM1を開いて，ブラウザと通信しているようすを図33に示します．ターミナル・ソフトウェアに表示されている1は，Javaアプレット側，つまりパソコンからイーサネットを通してシリアル・ポート側に送信したデータです．またターミナル・ソフトウェア側から0，1，2，3，4とキー入力すると，入力順に文字コードの下位3ビットがブラウザ側に表示されていきます．

JavaはC++よりも少ない行数でプログラムを作成でき，コンパイラも無償なのでハードルは低いと思います．この機会にJavaを試してみてはいかがでしょうか．

XPortのその他の機能

これまで説明した数々の機能は，XPortがサーバになって，他の機器から接続されるものでした．しかし用途によっては，サーバにネットワーク接続を行って，XPortをクライアントとして動作させたい場合もあるでしょう．

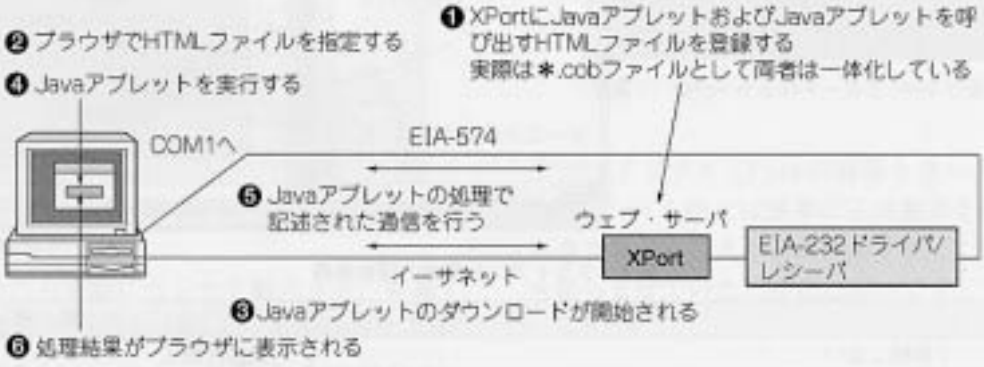
ここでは，XPortがクライアントになる機能を紹介します．

■ XPortがクライアントとなる設定

図20右下にあるportsをクリックするとダイアログ・ボックスが現れ，右側のスクロール・バーを上に移動させると図34のような画面になります．ここで，クライアント接続をするための設定を行います．

Active Connectionは，接続開始条件の設定です．

〈図32〉シリアル-イーサネット変換のJavaアプレット実行の流れ



〈リスト4〉シリアル-イーサネット変換を行うための変更部分(左の行番号はリスト2と対応している)

```
16 | Socket theSock=new Socket("192.168.0.1", 10001); // XPortへ接続 ―― ポート番号を10001に変更
24 | sockout.writeByte('1'); sockout.flush();          // '1' 送信   ―― 送信文字を'1'に変更
25 | g.drawString( "データ 1 送信完了",50,60);
34 |    if( (res&0xff)=='0'){                          // '0' 照合   ―― レスポンスの照合を'0'に変更
```

接続開始条件は表6のとおりです．またRemote HostにはサーバのIPアドレス，Remote Portにはポート番号を設定します．

設定が終わったら［OK］ボタンを押してXPortインストーラに戻り，ツール・バーの［Update］ボタンを押して設定を更新します．これでXPortが再起動し，クライアント接続が可能になります．

■ メール送信機能

この機能は図2(d)で紹介したもので，セキュリティ用途や，装置の稼動開始/停止/故障などの状態を管理者に通知したり，UPSの稼動開始通知など，さまざまな用途に使用可能です．

汎用I/Oの状態変化や，シリアル・ポートからユーザ指定のデータが入ると，XPortがメール・サーバ(SMTPサーバ)にメールを送信します．

● メール・サーバ情報の設定

図20の右側にある，Email Notificationの部分で設定します．メール・サーバの設定は以下の三つです．

Domain name：メール・サーバのドメイン名
Mail Server：メール・サーバのIPアドレス
Unit name：メール・サーバのアカウント

送信するメールの`From:`ヘッダは`Unit name@ Domain name`となります．

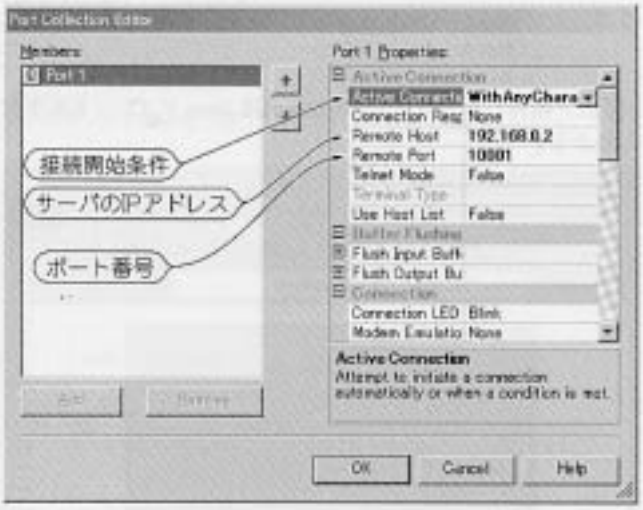
● メール送信先とトリガの設定

図20右側のEmail Notificationの中にあるRecipientsをクリックすると，図35の画面が表示されます．ここでメールの送信先を設定します．右側のEmail Address欄に，送信先のメール・アドレスを入力してください．アドレスは2件設定できます．

同じくEmail Notificationの中にあるTriggersをクリックすると，図36の画面が表示されます．ここでメール送信のトリガ条件を設定します．先ほど製作した赤外線センサ・ボードを使用することを想定して，`Test Sending`というタイトルのメールを，CP2がアクティブに変化した場合に送信するよう設定します．トリガ条件の設定は2番目と3番目もありますが，ここでは設定していません．設定が終わったら［OK］ボタンを押します．

ここまでの設定が終わったら，ツール・バーの［Update］ボタンを押して，XPortの設定を更新します．設定が終わった状態を図37に示します．なお，送信されるメールの本文はなにもありません．

〈図34〉クライアント機能の設定



〈図33〉ターミナル・ソフトウェアとブラウザ間で通信しているようす

〈表6〉クライアント接続の開始条件

設定	動作	備考
None	接続しない	出荷時設定
WithAnyCharacter	シリアル側から何かデータが送られると接続する	
WithCarriageReturn	シリアル側からキャリッジ・リターンが送られると接続する	
ManualConnection	シリアル側からのコマンドで接続する	C192.168.0.2/1001ならIPアドレス192.168.0.2，ポート10001に接続する
AutoStart	XPortの起動時に自動的に接続する	
WithActiveDTR	CP3をDTR(入力)に設定し，入力がアクティブになると接続する	

■ メール送信機能に関する注意点

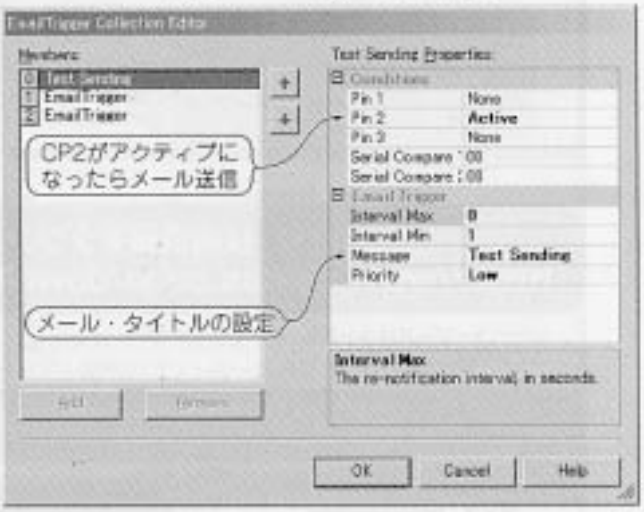
● POP before SMTPには対応していない

XPortファームウェア1.20では，POP before SMTPに対応していません．POP before SMTPとはメール・サーバの不正使用を防ぐ機能です．メール受信のためのPOPサーバでアカウントとパスワードを確認したあと，プロバイダが定めた一定時間内だけSMTPサーバでのメール送信が行えます．

プロバイダのSMTPサーバがPOP before SMTP機能を使っている場合，XPortが送信するメールを受け付けない可能性があります．回避手段としては，ルータやパソコンから一定時間ごとにメール・チェックを行うようにしておくことが考えられます．

〈図35〉メール送信先の設定

〈図36〉メール送信のトリガ設定



〈図37〉メール送信の設定が完了した状態

● メール送信機能のテスト

設定が終わったら実際にセンサを動作させ，**写真7**のように携帯電話やPHSなどにメールが到着するか確認します．`Subject:`ヘッダは`Notification:`(通知)という言葉から始まり，続いて先ほど設定した`Test Sending`というタイトルが付いています．

〈写真7〉XPortからのメールをPHSで受信したようす

■ おわりに

本稿では前編，後編の2回にわたり，小型ネットワーク・アダプタXPortの概要と使い方を解説しました．XPortを使えば簡単に応用機器を作れるので，本記事を参考にさまざまな機能を使った製品のアイデアを膨らませていただけると幸いです．

◆参考文献◆

(6) AB-186アンプ内蔵型光電スイッチ(小型)E3Z説明書，オムロン㈱．

(7) G5V-1マイクロリレー説明書，オムロン㈱．

(8) Java2の日本語解説，サン・マイクロシステムズ㈱．
http://java.sun.com/j2se/1.4/ja/docs/ja/api/index.html

■ 本記事は2003年8月時点のXPortを元にしています．記事中の仕様は変更される場合がありますので，ご使用の際は最新のデータシートを参照してください．