

# 「PHS経由でネットに接続できる Ethernetアダプタのファームウェアをハック」

後編

連載の締めくくりとして、Ethernetアダプタ「OSX-1」を利用した遠隔監視システムの製作事例を紹介する。画像メールを自動返信する仕組みを用意して、遠隔監視を実現した。  
(編集部)

本稿の前編(本誌2007年8月号, pp.128-135)では、ウィルコム(PHS)のPHSモジュール「W-SIM」を利用してインターネットに接続できるEthernetアダプタ「OSX-1(愛称: つないでイーサ)」(写真1)をハックし、開発環境を導入して簡単なプログラムを動作させてみました。中編(本誌2007年9月号, pp.130-137)では、導入した開発環境を利用してSMTP(Simple Mail Transfer Protocol)サーバやPOP(Post Office Protocol)サーバにアクセスし、メールを自動返信するシステムを構築しました。このようにメールを利用すれば、ファイア・ウォールの内側にある機器を携帯電話などで容易に制御できます。

前編と中編を通じて、本機が単なるW-SIMを利用したダイヤルアップ・サーバではなく、ネットワークを使うちょっとしたシステムの構築に利用できる機器であることを理解していただけたかと思います。本機に組み込まれているインタープリタ言語のSilentCサイレントシーを利用すれば、セルフ開発環境でソケットを利用したネットワーク・プログラムを容易に記述できます。

カメラ画像を添付して返信するシステムを構築する自分のペットは元気になっているだろうか? 現在の部屋の温度はどうなっているのだろうか? 積雪の状態を監視して融雪システムを制御できないだろうか? など、携帯電話を利用して画像を確認したいという需要は多く、各社からさまざまなシステムが提供されています。しかし、一般家庭に設置できる価格帯の製品はまだ少ないようです。

その理由としては、中編で解説したように企業内や一般家庭内のインターネット環境において外部からファイア・ウォールを超えて内部にアクセスするのは難しいことと、カメラ画像を扱うためにはかなり本格的なシステムを利用しなければ実現できないことが挙げられます。

そこで後編では、中編で説明したメール自動返信システムに小規模なマイコンでも画像を扱えるJPEGカメラ・モジュールを接続して、携帯電話で気軽に画像を確認できるシステムを構築します。

実はカメラ・モジュールを利用するのは難しい

カメラ・モジュールを利用して画像を扱うシステムを構築する際に問題となってくるのが、どうやってカメラ画像をデジタル化して取り込むかという点です。現在市販されているカメラ・モジュールのほとんどが、NTSC(National Television Standards Committee)ビデオ信号を出力するものばかりです。NTSCはアナログ信号なので、そのままモニタに接続するには手軽で良いのですが、カメラ画像をデジタル・データとして扱う際には、ビデオ・キャプチャ回路が必要になります。

また、一部のカメラ・モジュールの中には、「デジタル出力」と銘打っているものもあります。しかし、実際には27MHzクロックに同期してYUVデータが洪水のよう



(a) W-SIM



(b) OSX-1

写真1 W-SIMとOSX-1の写真

ウィルコム(PHS)の無線モジュール「W-SIM」を、ウルトラエックスのEthernetアダプタ「OSX-1」に挿入して使用する。OSX-1は、W-SIMのジャケッ( W-SIMを挿入できる機器)の一つである。

にどんどん送られてくるので、かなり高い性能を持ったシステムで真剣に画像を受け取らなければ処理しきれません。高速なCPUを使ってもソフトウェアだけで処理するのはかなり難しく、いったん画像データをメモリに格納するハードウェアと組み合わせる必要があります。

組み込み機器で普通に利用されている安価なマイコンを単体で使用し、カメラ画像を扱うのは容易ではありません。扱う画像データの容量と転送速度を下げることに由り、この問題を解決することにします。

### JPEG カメラは組み込み機器にうってつけ

今回使用する COMedia 社のカメラ・モジュール「C328-7640」は、OmniVision Technologies 社の画像処理専用 LSI「OV528」を利用した、カメラ画像を扱えるモジュールです(写真2)。

モジュールの内部構造は次の通りです。画像メモリと JPEG 圧縮プロセッサを内蔵した OV528 チップに 30 万画素の出力センサが直結されており、カメラ・センサから出力されたデータをいったん OV528 内の画像メモリに格納します。格納したデータはそのままシリアル・ポート経由で取り出すこともできますが、VGA サイズ(640ピクセル×480ピクセル)の画像だと16ビット・カラーでもデータ・サイズが600Kバイト以上になってしまいます。

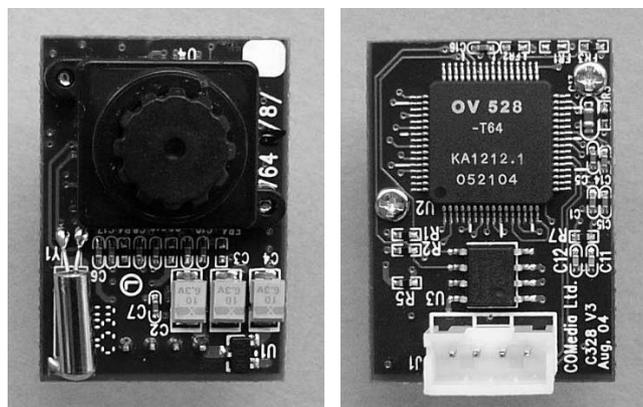
生データのままではまだ大きすぎるので、内蔵の JPEG 圧縮プロセッサを利用して30Kバイト程度にデータを圧縮します。この程度のサイズになれば、メールに添付したり

ネットワーク経由で送ったりすることが容易になります。

接続コネクタも極めてシンプルで、3.3V 電源とシリアル入出力の4本だけです(写真2(b))。カメラを制御するためにはシリアルでコマンドを送出します。表1に示すコマンドを用いて、以下のように対話的にカメラを制御します。

- SYNC コマンドでカメラを起動してボーレートを確定させる
  - Initial コマンドで画面の解像度や色数を設定する
  - Snapshot コマンドで写真を撮る
  - Get Picture コマンドで圧縮された JPEG 画像データの送信開始を指示する
  - Data コマンドでカメラから得られる画像データのサイズを把握する
  - 設定したサイズ(デフォルトは512バイト)のパケットが送られてくるので、受け取ったら ACK コマンドを返す
- 今回の実験に使用するハードウェアについて

前編と中編では OSX-1 のファームウェアを書き換えることで、さまざまな事例を解説してきました。ファームウェアをいくら書き換えてもリカバリ処理を行えば完全に元に戻るの、思い切ったハックが行えました。一方今回は、JPEG カメラをハードウェア的に接続する必要があります。もちろん OSX-1 をケースから取り出して内部基板に4本の信号線を接続すれば OSX-1 でもカメラ画像を扱えるようになります。ただし、これだと当然メーカーの保証適用外です。そこで今回は、OSX-1 の内部に採用されている



(a) 表

(b) 裏

### 写真2 JPEG カメラ・モジュールの外観

(a)の表側に、30万画素のイメージ・センサが実装されている。リングを調節すると、 $\pm 5$ cmの範囲でピントが合う。大きさは横20mm、縦28mm。(b)に示す裏側には、OmniVision Technologies 社製の画像処理専用 LSI「OV528」が実装されている。接続コネクタの信号は、左から  $V_{cc}$ (3.3V)、Tx、Rx、GND。

表1 C328の制御コマンド表

コマンド	説明
Initial	カメラの解像度や色数などを設定する
Get Picture	画像の転送を開始する
Snapshot	現在の画像をバッファに保存する
Set Packet Size	カメラから転送される画像データのパケット・サイズを指定する
Set Baudrate	シリアルのボーレートを設定する
Reset	C328 をリセットしてステートをアイドルにする
Power Off	C328 をスリープ・モードにする。 SYNC コマンドで復帰する
Data	カメラから送られる画像データの種類とサイズを取得する
SYNC	ボーレートを自動検出して設定する。 スリープから復帰させる
ACK	肯定的なレスポンス。データ転送時の同期に用いる
NAK	エラー・レスポンス。 各種のエラー・コードが返される
Light Frequency	照明の周波数を設定する (50Hzまたは60Hz、フリッカ防止用)