

簡単なネットワーク対応アプリケーションの作り方

中本 伸一

付属ColdFireマイコン基板を利用して実際にネットワーク通信を行ってみる。最も単純な通信プロトコルであるUDPからスタートして、最も広く利用されているTCP/IPを利用してメールサーバに接続し、メールの送受信を実際に体験してみる。締めくくりとして、指定した時間にメールを送信するパーソナル・リマインダ・システムを作成する。(筆者)

1. UDP を使ってみよう

● 0～9999までの数字を指定されたIPアドレスに連続的に送信する

付属 ColdFire マイコン基板を使って、実際にネットワーク通信を体験してみましょう。

ネットワークでデータを送受信するためには何通りかのプロトコルがありますが、まず一番初めにチャレンジするのはUDP通信です。UDP通信のメリットは、高速でいつでも好きなときにデータを送信できることですが、相手がデータを確実に受け取ったかどうかを確認できないというデメリットもあります。しかし、データを何度も送信したり確認のパケットを受信すれば、十分実用的なデータ交換ができます。付属 ColdFire マイコン基板を使用し、UDP通信でデータをパソコンに向けて送信してみましょう。

ダウンロード・サイト (<http://www.cqpub.co.jp/interface/>) からプログラム・ソースをダウンロードして解凍したら、2-1 というフォルダ内の Main を tftp で転送してください。既に基板には Main が存在しているため、転送の前に Main を削除する必要があります。

あるいは telnet で基板に接続してリスト1のプログラムをそのまま入力してください。その後、save Main で入力

したプログラムを Main という名前で保存します。どちらの方法で Main を作成しても結構ですが、最終的に list コマンドでリスト1が表示される状態にしてください。

このプログラムは0～9999までの数字を指定されたIPアドレスへ連続的に送信します。次に、送信されたUDPデータをパソコン側で受け取り、表示するソフトウェアを用意しましょう。マイクロソフト社の Web サイトから無料でダウンロードできる Visual C# 2005 Express Edition をパソコンにインストールし、リスト2のプログラムを入力してコンパイルします。こうした作業が面倒であれば、2-1 フォルダ内にある UDP_Dispatch.exe を利用してください。実行の際は.NET Framework 2.0が必要なので、あらかじめパソコンにインストールしておいてください。

UDP通信の実験の前に、パソコンのIPアドレスを調べておきます。コマンド・プロンプトから ipconfig を実行すると、次のような画面が表示され、パソコン側のIPアドレスを調べることができます。

```
C:\>ipconfig
Windows IP Configuration
Ethernet adapter □-カル・エリア接続:
Connection-specific DNS Suffix . :
IP Address . . . . . : 192.168.1.2
Subnet Mask . . . . . : 255.255.255.0
Default Gateway . . . . . : 192.168.1.1
```

リスト1

0～9999までの数字を指定されたIPに連続的に送信するプログラム

```
10 main(char *s){char *d,soc=CreateSocket(0);long ip=GetIP(s);int n; //引き数からIPアドレスを得る
20 d=MemoryAlloc(6);for(n=0;n<10000;n++){GetDigit(n,d); //数字バッファを確保して10000までのループ開始
30 if(SendTo(soc,ip,30049,d,StrLen(d))<=0)break;} //数字をUDPで送信する
40 CloseSocket(soc);MemoryFree(d);} //ソケットと数字バッファを開放
```

リスト2 UDP 通信プログラム

```
using System;
using System.Collections.Generic;
using System.Text;
namespace ConsoleApplication1
{
    class Program
    {
        static void Main(string[] args)
        {
            System.Text.Encoding enc = System.Text.Encoding.UTF8; //UTF8を指定する
            System.Net.Sockets.UdpClient udp = //ローカル・ポート番号をバインドする
                new System.Net.Sockets.UdpClient(30049);
            for(;;) //データを受信するループ
            {
                System.Net.IPEndPoint EndPoint = null; // UDP エンドポイントを用意する
                byte[] rcvBytes = udp.Receive(ref EndPoint); // UDP データを受信する
                string rcvMsg = enc.GetString(rcvBytes); // 受信したデータから文字列に変換
                Console.WriteLine("送信元アドレス:{0}/ポート番号:{1}/受信したデータ:{2} \n",
                    EndPoint.Address, EndPoint.Port, rcvMsg); // 到着したデータの表示
                if(Console.KeyAvailable) // キーをセンス
                    break; // ループを抜ける
            }
            udp.Close(); //UDP 接続を終了
        }
    }
}
```

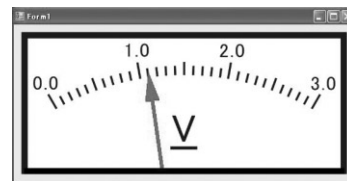


図1 UDP 電圧計の画面

れば、いつでもデータを送信できます。しかし相手が受け取ったかどうかは全く分かりません。もし確実に相手がデータを受け取ったかどうかをチェックしたいなら、相手からもデータを受け取った確認をUDPでデータ送信することになります。

この操作で、パソコンのIPアドレスは192.168.1.2であることが判明しました。これで準備完了です。パソコン側でUDP_Dispatch.exeを実行します。初回の起動時にはWindowsのファイアウォールからの警告ダイアログが出るので「ブロックを解除する」をクリックしてください。次に、telnetで付属ColdFireマイコン基板に接続してOKプロンプトを確認し、以下のように入力します。

```
main("192.168.1.2")
OK
```

mainに与えるIPアドレスは、先ほど調べたパソコンのIPアドレスに合わせて変更してください。パソコンのコマンド・プロンプト画面上に、0～9999までの数字が表示されます。10,000個のパケットを送信する時間を測定すれば、1個当たりの送信時間を算出できます。筆者の環境では27秒ほどかかったので、1パケット当たり2.7msという計算になります。

Main(リスト1)の動作としては、まず与えられた引き数からIPアドレスを得ます。次に数字を格納するためのバッファを6バイト確保し、for文で0～9999までループします。ループの中ではループ変数を数字文字列に変換し、SendToを用いて対象のIPアドレスの30049番ポートへ対して数字を送信します。ループが終了したらソケットを開放し、数字を格納するバッファを開放して終了しています。

UDP通信は相手のIPアドレスとポート番号さえ指定す

2. UDP を利用した電圧計を作成

● 加速度センサから出力される電圧を A-D 変換し、そのデータを UDP で送信する

MCF52233にはA-D変換器が内蔵されています。UDPを利用して、このA-D変換器で得られた値をパソコンの画面上でアナログ・メータのように表示させてみましょう。UDPのデータ転送そのものは前の節とあまり変わりません。この節では、付属ColdFireマイコン基板に実装されている加速度センサから出力される電圧をA-D変換し、そのデータをUDPで送信してみましょう。基板を傾けるとそれに応じてパソコンのメータの針が動くのは、なかなか楽しいものです(図1)。

パソコン側のソフトウェアはWindowsアプリケーションです。皆さんの環境でコンパイルするには、リソースを含むプロジェクト・ファイルのすべてが必要になります。そのため、2-2フォルダ内にVisual C# 2005 Express Edition向けのプロジェクト・ファイルが入っています。実行部分のソース・ファイルをリスト3に示します。

プログラムの動作は、基板からのUDPデータを受け取ってudpvalという変数に格納するというスレッドを起動しておきましょう。タイマ割り込みを利用して200ms間隔でudpvalの値に従ってアナログ・メータ風の画面をウィンドウに描画します。

このアナログ・メータをそのまま利用するなら、2-2