



# ColdFireマイコン基板搭載 CインタプリタSilentC活用テクニック

第5回 SilentC用プログラミング・テクニック(その1)

中本 伸一

今回からSilentCを活用するためのさまざまなプログラミング・テクニックについて解説する。今回はパラメータを保存したり、保存したパラメータを読み出したりするファイル・アクセスの方法と、ソケット・プログラミングによるTCP/IP通信プログラム、そしてHTTPプロトコルを処理するサーバ・プログラミングについて解説する。(編集部)

## 1. ColdFire マイコン基板側の準備

今月から2008年9月号付属ColdFireマイコン基板(以降マイコン基板)で、実用的なアプリケーションを構築するためのサンプル・プログラムを解説していきます。このサンプルの理解がマイコン基板を活用する際の参考になれば幸いです。併せてSilentCのちょっとしたTipsも解説します。

今回は、「設定メニューと設定の保存」と「簡単加速度サーバ」、「簡単HTTPサーバ」の三つをサンプルとして紹介します。そしてこれらを、SilentCを利用して同種の実用的なプログラムを作成する際のテンプレートとして活用することを目標にします。

### ● SilentC を最新版にアップデート

マイコン基板には、ファームウェアを最新版にアップデートする機能が内蔵されています。これを利用して、SilentCを最新版にアップデートしましょう。まずは筆者の会社のWebサイト(<http://www.silentsystem.jp>)から、最新版のファームウェアを入手します。ダウンロードしたファイルはLZH形式で圧縮されているので解凍します。

次に、マイコン基板をネットワークに接続して電源を供給します。コマンド・プロンプトを起動し、pingコマンドでマイコン基板のIPアドレスにアクセスできるかどうかを確認します。

ダウンロードしたアーカイブ内には、Update.batというバッチ・ファイルが含まれているのでこれを起動します。マイコン基板のIPアドレスを入力してEnterを押すと、新ファームウェアをマイコン基板へ転送します。実際にはSILENTC.BINというファイルをtftpのバイナリ・モード

で転送しています。

転送が終わるとチェックサムが画面に表示されるのでメモしておきましょう。Enterを押すとtelnetが起動しておなじみのOKプロンプトが表示されるので、`util:::update`と入力して先ほどメモしたチェックサム値を入力します。Ready?という問いにyで答えるとファームウェアの書き換えを開始します。しばらくしてマイコン基板上のLED2が点滅すれば完了です。書き換えの最中には、電源を切ったりしないよう十分に注意してください。

### ● サンプル・プログラムを転送

まず本誌Webページ(<http://www.cqpub.co.jp/interface/>)のダウンロード・コーナーあるいは筆者の会社のWebサイトから、今月号のアーカイブをダウンロードします。解凍したものにはSend.batというバッチ・ファイルが含まれているので、これを起動します。ColdFireマイコン基板のIPアドレスを入力するとmenu、server、websの三つのプログラムと、http、head、tailの三つのデータ・ファイルが転送されます。同名のファイルがすでに存在している場合には自動的に削除されるため、消されてまずい場合はあらかじめ別名にリネームしておいてください。またこのバッチ・ファイルは、自動的にdefragコマンドでファイルの空き領域を最適化します。バッチ・ファイルの実行が終了した後、telnetでログインし、dirコマンドで上記の六つのファイルが正しく転送されていることを確認してください。

## 2. 「設定メニューと設定の保存」のためのサンプル・プログラム

### ● パラメータの保存と読み出し方法

SilentCを利用して設定メニューで各種の値を設定し、

## リスト1 「設定メニューと設定の保存」のための サンプル・プログラ ム (menu)

```

getn(char *p){int v;for(;;){if(*p<'0' || *p>'9')break; // Get decimal number
v=v*10+*p++-'0';}return v;} // multiply each digit
long getip(char *p){long v;v=getn(p)<<24;while(*p&&*p!='.')p++; // Get IP adrs from string
if(*p=='.')p++;v=getn(p)<<16;while(*p&&*p!='.')p++; // Get 2nd digit
if(*p=='.')p++;v=getn(p)<<8;while(*p&&*p!='.')p++; // Get 3rd digit
if(*p=='.')p++;v=getn(p);return v;} // Get 4th digit
p(char a){unsigned char b;b=a;PrNum(b);PrChar('.');} // Put number + .
q(char a){unsigned char b;b=a;PrNum(b);} // Put number only
main(char a){char *p,*b,*f,*fp,c,*k;int *s;long *ip,*n; // main() start,declare vars
p=MemoryAlloc(8);ip=p;s=p+4;b=p+6;f=p+7; // Alloc user setting buffers
fp=OpenFile("UserSetting"); // Try to read setting file
if(fp){ReadFile(fp,p,8);CloseFile(fp);} // Exists then read setting
else{*ip=0xc0a8010a;*s=9999;*b=1;*f=0;} // Else set default value
k=MemoryAlloc(80);n=MemoryAlloc(28); // Alloc input and baud rate list buffer
n[0]="115200";n[1]="57600";n[2]="38400";n[3]="28800";n[4]="19200"; // Set baud list
n[5]="14400";n[6]="9600"; // Set baud list
for(;;){PrStr("¥033[2J¥033[1;1H1:IP = "); // Menu loop begins here
p(*ip>>24);p(*ip>>16);p(*ip>>8);q(*ip); // Show IP adrs xxx.xxx.xxx.xxx format
PrStr("¥033[2;1H2:Baud = ");PrStr(n[*b]); // Set baud rate number
PrStr("¥033[3;1H3:Mode = ");PrStr(*f?1:"Master":"Slave"); // Show mode
PrStr("¥033[4;1H4:Number = ");PrNum(*s); // Show number
PrStr("¥033[6;1HCommand(1-4,w,q)="); // Command menu
#stop 0;for(;;){c=Getc(0);if(c)break;}#stop 1;switch(c){ // Get command and switch
case '1':PrStr("¥033[1;1H¥033[2K1:IP = "); // Command 1 (change IP adrs)
Gets(k,80,1);*ip=getip(k);break; // Get IP adrs and set var
case '2':*b+=1;if(*b>6)*b=0;break; // Command 2 rotate baud rate list
case '3':*f^=1;break; // Command 3 flip flag
case '4':PrStr("¥033[4;1H¥033[2K4:Number = "); // Command 4 (change integer)
Gets(k,80,1);*s=getn(k);break; // Get number and replace value
case 'w':DeleteFile("UserSetting");CreateFile("UserSetting");// Command w then create file
WriteFile(p,8);CloseFile(0);break;}if(c=='q')break;} // Write to setting file
MemoryFree(n);MemoryFree(k);MemoryFree(p);} // Clean up and quit

```

それをファイルに保存するサンプルを説明します。特に通信先のIPアドレスなどは、メニューで簡単に変更できるようにしておけば大変便利です。IPアドレスやボーレートなどのリスト選択、ON/OFFを反転するフラグ、16ビットの数値の値をメニューで設定して、ファイルに保存してみましょう。

プログラム・ファイルのmenu(リスト1)を見て下さい。各行にコメントを付加しているので、内容は理解しやすいと思います。まず設定を保存してあるファイルを読み込んで、過去の設定値を変数にセットします。もしファイルがなければ、デフォルト値を初期値として変数に設定します。次に設定メニューを表示して、現在の設定値をユーザに表示してコマンド待ちになります。

### ● ブレーク・チェックの制御

ちょっとしたテクニックですが、SilentCではCTRL+Cを常に監視してプログラムを中断できるようにブレーク・チェックが行われています。ブレーク・チェックを行うと、入力した文字が消えてしまいます。そのためGetc(0)でキーが押されたかどうかチェックしたい場合、#stopディレクティブを利用して、ブレーク・チェックを無効してからGetc(0)を呼び出します。文字が入力された後に#stop 1でブレーク・チェックを有効に戻します。この1文字入力は便利なので、ぜひこの方法を利用してください。

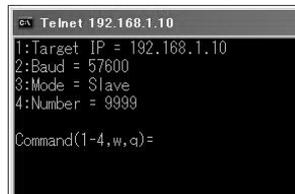


図1  
「設定メニューと設定の保存」のためのサンプル・プログラム動作時のtelnet動作画面

### ● 動作確認

それではサンプルを実行してみましょう。telnetでSilentCにログインし、menu::mainと入力してEnterを押すとmenuプログラムを実行できます。すると図1のような設定メニューが表示されます。1から4のコマンドを入力すると、それぞれの数字によって設定値を書き換えます。IPやNumberの場合にはキーボードから有効な数値を入力し、Enterを押せば設定が変更できます。またBaudやModeの設定は、数字を押すたびに設定値が直ちに変更されます。

設定を変更してもそれを保存しなければ、変更は無効になります。wを入力するとUserSettingというファイルに設定値を書き出します。qで設定メニューを抜けてプログラムを終了します。ユーザの設定が正しく保存されているかどうかを確認するため、再度menu::mainを実行して設定メニューを起動し、先ほど設定したデータが反映さ