

Cインタプリタ SilentCの使い方

中本 伸一

● mainプログラムの実行/修正

付属 ColdFire マイコン基板上(以降、本基板)には、出荷時から SilentC が管理するファイル領域があります。これに書き込まれている Main ファイルには、簡単なサンプルが含まれています。“OK”というプロンプトが出ているときに run と入力すると、画面におなじみの“Hello World!!”が表示されます。

```
run
Hello World!!
OK
```

この Main ファイルの中身をのぞいてみましょう。ファイル内容の表示には type コマンドを使います。SilentC で利用できるコマンドに関しては第1章の表3(p.59)を参照してください。

```
type Main
main(){PrStr("Hello World!!");}
OK
```

表示されたのは最もシンプルな C 言語のプログラムです。main() 関数の中で呼ばれている PrStr という関数は、文字列を表示するためのライブラリ関数です。SilentC に用意されているライブラリ関数に関しては表 A を参照してください。

それではこの Main を編集してみましょう。まず、load コマンドで編集したいファイルを編集モードにします。

```
load Main
Main loaded
OK
```

list コマンドは、編集モードにしたファイルの内容を行番号を付加して表示します。

```
list
10 main(){PrStr("Hello World!!");}
OK
```

先頭の 10 は BASIC の行番号と同じ働きをします。先頭に数字を付けてテキストを入力すると、そのテキストは先頭の番号順に編集バッファ内に蓄積されます。もし同じ番号を入力した場合にはその行を入れ替えます。また番号のみを入力するとその行は削除されます。

では表示される文字列を“Hello Ethernet!!”に変更してみましょう。edit 10 と入力すると 10 番の行が編集できる状態になるので、カーソルと Delete キーを利用して World を Ethernet に変更してください(telnet のターミナル・ソフトウェアとしてハイパーターミナルを使っている場合、Delete キーを入力したときはシフト + Backspace キーを入力)。変更したら、list

コマンドで入力したテキストを表示してください。

```
list
10 main(){PrStr("Hello Ethernet!!");}
OK
```

ために run コマンドを入力します。しかし実行結果はまだ“Hello World!!”のままです。これはまだ編集した結果を書き戻していないのが原因です。プログラムに変更を加えたら save コマンドで書き込まなければ変更は反映されません。

それではファイルに書き出します。save コマンドを実行してください。Main に書き出してよいかどうかの確認には“y”と答えます。このコマンドで変更したテキストが Main に保存されます。このとき、先ほどテキストの先頭に付加されていた行番号は取り除かれ、普通のテキスト・ファイルとして書き出されます。type Main で変更した内容を確認できます。

```
save
save Main?(Y or N)y
OK
```

run コマンドを実行すると次のように表示されます。ちなみに run コマンドは Main ファイルの中の main() という関数を実行します。もし何かエラーが出た場合は edit 10 というコマンドを入力し、カーソルを利用してテキストを修正した後に save コマンドで編集結果を保存してください。

```
run
Hello Ethernet!!
OK
```

このように SilentC には簡単な編集機能が内蔵されています。けれども今後解説するサンプル・プログラムなどは SilentC の編集機能で編集するよりも、パソコン上で編集してから tftp で送り込んだ方が使いやすいと思います。

例えば Main を送り込むのであればコマンド・プロンプトで、

```
C:\>tftp 192.168.1.10 get DeleteFileMain
C:\>tftp 192.168.1.10 put Main
```

の2行をバッチ・ファイルなどにしておけば、パソコンでファイルを修正後に簡単に基板に転送できます。

● ファイル内の関数を呼び出す

SilentC はコンパイラではなくインタプリタなので、入力したテキストをコンパイルせずにそのまま実行します。perl などのスクリプト言語に共通する手軽さがあります。すべてのプログラムはファイル上に存在しています。そこで、よく使うプログラム(ライブラリ)はファイルにまとめて入れておけばいつ

も呼び出せます。

少し実験してみましょう。先ほど入力したプログラムがまだ編集バッファに残っているはずですが、念のため list コマンドで確認してください。もし消えてしまっていたら、load Main コマンドで編集バッファに読み込みます。

次に、このプログラムを save Hello というコマンドで Hello というファイルに書き込みます。この時点では、先ほど作成した Main と今書き出した Hello という二つのファイルには、全く同じ内容が格納されているはずですが。

さて、今作成した Hello ファイルを早速実行してみましょう。次のようにコマンドを打ち込んでください。ファイル名::関数名() で指定したファイルの指定した関数を呼び出せます。またこの main 関数には引き数がないので () を省略できます。

```
save Hello
OK
Hello::main
Hello Ethernet!!
OK
```

edit 10 と入力すると、現在の編集バッファにあるプログラムを修正します。カーソル・キーの左右でカーソルを移動させ、次のようにプログラムを修正してください。for を利用して何度か関数を呼び出すサンプルを作成します。修正が終わったら save Main コマンドで編集結果を Main へ保存しておきます。

```
10 main(){int i;for(i=0;i<5;i++)Hello::main;}
save Main
save Main?(Y or N)y
OK
```

run で実行させると、期待通りに動作するはずですが、ちなみに run コマンドは、内部的には Main::main を実行しています。

● ダイレクト実行はデバッグ時に便利

SilentC はファイル内の関数を実行しますが、昔の BASIC と同じようにコマンド・ラインからのダイレクト実行も可能です。ためしに次のコマンドを入力してください。? コマンドは、式の値を 10 進数で表示します(?? なら 16 進数で表示する)。

```
?2+2/2
3
OK
```

C 言語の仕様では、{ } で囲まれたブロックは一つの実行文として扱われます。次の例は、{ } で囲まれたブロックの内部でローカル変数を宣言し、for ループの変数として使用しています。

```
{int i;for(i=0;i<40;i++)PrNum(i);}
0123456789101112131415161718192021222324
252627282930313233343536373839
OK
```

プログラムが無制限ループに入っても、CTRL + C キーの入力

で中断できます。CTRL + C で中断するとブレイク・モードに入ります。ブレイク・モードでは編集コマンドが使用できません。使えるコマンドは abort, cont, type, ? コマンドのみです。abort でプログラムの実行を完全に中断して編集モードに戻ります。cont は実行を再開します。[CTRL] + [C] で中断したときに上矢印を入力すると abort が自動的に入力され、そのまま Enter を押せば実行を中断できます。ほとんどのケースで CTRL + C と abort はセットで入力することになるので、覚えておいてください。

SilentC では、Main ファイルの先頭で宣言している変数がグローバル変数として扱われます。また配列はポインタ変数に添え字を付けてアクセスするポインタ型配列のみ利用可能です。配列を利用するには、まず必要な型のポインタ変数を宣言して MemoryAlloc() を利用して必要な領域を確保し、ポインタ変数にセットしてから利用します。構造体はポインタ参照 (-> 演算子) のみ利用可能ですが、かなり使い方が難しいので今は使わないでプログラムすることをお勧めします。

● 大容量ファイル領域が使える OS-2

本基板に搭載されている SilentC は、ユーザが開発したプログラムを MCF52233 内蔵フラッシュ ROM に書き込みます。しかし、残念ながら容量が最大 48K バイトしかありません。実際には十分な容量ですが、実用的で見栄えの良いプログラムを作成しようとするとうまく領域が足りなくなってしまう。

また SilentC は、一度作成したプログラムをいつでも「ファイル名::関数名」で呼び出せます。これを利用すると、自分の使いやすい関数をファイルとして保存しておくだけでユーザ・ライブラリとしていつでも利用できます。けれども本基板は容量制限があり、実際にはライブラリ・ファイルを保存しておく領域がすぐにいっぱいになってしまいます。

「SilentC を使って思いっきりプログラムを作りたい!」という場合は、SilentC フルスペック・バージョンを搭載している OS-2 (写真 A) を利用してはいかがでしょうか?

OS-2 の大きな特徴は、プログラムの格納用に外付け 1M バイトのフラッシュ ROM を実装している点です。これだけの容量があればほとんどのアプリケーションに対応できます。また MicroSD カードを挿入できるコネクタも実装しています。本体

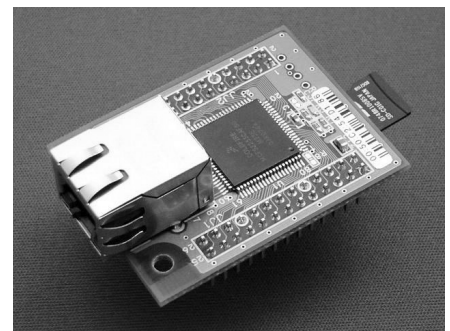


写真 A
大容量ファイル領域 & SilentC フルスペック・バージョン搭載 OS-2