



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10回 (最終回) ネイティブ・プログラムの作成

中本 伸一

付属ColdFire基板を活用する最後として、プロトコル・スタックであるSilentMoonを呼び出すネイティブ・コードを作成し、作成したコードをColdFire基板に転送して実行する方法を解説する。SilentMoonを利用するとマルチスレッド関係のライブラリが利用できるようになり、SilentCでは不可能だった本格的な開発が可能になる。

(編集部)

今回が付属 ColdFire マイコン基板 (以下、ColdFire 基板) を活用する最後のテーマです。SilentC を使わずに ColdFire 実行用のネイティブ・コードを作成し、ColdFire 基板上に転送して実行する方法を解説します。ネイティブ・コードを利用すれば、SilentC では不可能だった大規模プログラムやマルチスレッド機能も利用できます。

1. 本格的な開発をするには

● SilentC は機能制限がきつい

ColdFire 基板に標準で実装されている、インタプリタ型の C 言語である SilentC は、セルフ環境でプログラミングが可能で、telnet で接続するだけで手軽にネットワーク・プログラムを作成できます。リアルタイム性をあまり必要としないプログラムであれば、SilentC で十分に記述できました。

しかし、SilentC は会話型で手軽に開発できるという利点はあるものの、次のような欠点もあります。

- 容量の制限があり、大きなプログラムが作成できない
- インタプリタなので、実行速度が遅い
- シンタックス・チェックが甘く、エラーの発見が難しい
- C 言語ライクであるが、厳密には C 言語をそのまま移植できない

これはすべてサイズが小さいインタプリタ型の言語ゆえともいえるでしょう。インタプリタはちょっとした実験には便利ですが、本格的な応用にはかなり制限があるのも事実

実です。

● gdbstub は上級者向けで難しい!

また ColdFire 基板には、SilentC 以外の開発を行うために gdbstub も搭載されています。gdbstub を利用すれば、gcc で作成したネイティブ・コードを ColdFire 基板上に転送して実行できます。

しかし TCP/IP スタックを利用するには、TOPPERS などの OS のソースを準備して、環境を詳細に設定しビルドした上、無償の TCP/IP スタックを組み込み、最後に自分のプログラムをリンクするという面倒な作業が必要でした。本誌上では簡単に解説していますが、このような作業はかなりの上級者向けであり、豊富なノウハウと高いスキルが要求される開発スタイルです。

● プロトコル・スタック SilentMoon を活用する

そこで今回は、SilentC のプロトコル・スタックである SilentMoon を活用します。

SilentMoon は約 36K バイトのコンパクトなモニタ・プログラムです。マルチスレッド機能とメモリ管理機能をベースに、TCP/IP プロトコル・スタックが標準実装されています。ネイティブ・プログラムから SilentMoon の内部ライブラリを利用すると、SilentC ではサポートされていなかったマルチスレッド関係のライブラリが使用できます。実は、SilentC はこの SilentMoon 上の一つのアプリケーションとして動作しています。

SilentMoon の内部ライブラリへは、ベクタ参照でアクセスします。MCF52233 のフラッシュ ROM の先頭部分は割込みベクタ領域ですが、このベクタ領域の大部分は未使用です。その未使用領域を利用して、SilentMoon の内部ライブラリへのベクタが格納されています。

ユーザはベクタで示される番地を、単に参照コールするだけで内部ライブラリを利用できます。今回はこの機能を

◆マイコン基板付き号&バックナンバーの問い合わせ先◆

販売部 Tel : (03) 5395-2141 / FAX : (03) 5395-2106
e-mail : sales@cqpub.co.jp
詳細は <http://shop.cqpub.co.jp/> をご覧ください