

## 第5章

GNUクロス開発環境と  
C言語標準関数ライブラリで

# オリジナル・モニタ・ プログラムを作ろう

山際 伸一

ここでは、オリジナルのモニタ・プログラムを作成する方法について解説する。 (編集部)

本章で使用しているプログラムは  
[http://www.cqpub.co.jp/  
interface/download/](http://www.cqpub.co.jp/interface/download/)から  
ダウンロードできる。

### 1. モニタは“のぞき”行為?!

組み込みプログラムの作成をする場合、いざ動作確認という段階になって実際に動作させてみたら、何かの弾みにプログラムが停止してしまった。そういう、俗に言う「わけのわからない」バグにはまる場面に遭遇することがよくあります。

このようなとき、組み込み機器の多くはOSが動作していないため、どのような状況で固まっているのかを判断できません。その原因は、マイコンがとんでもないアドレスを読んでいて、その値の変化待ちだったり、周辺機器のバグで、変化するべき値が変化しなかったり、本質的にアルゴリズムがまちがっていたりするものです。

このような判断の難しい状況に陥った際に、その時点でどのようなことが起こっているかを外からのぞき見できれば、多くの問題を解決することができます。このような「のぞき」を行うソフトウェアのことをモニタと呼びます。

一般的にICE(In Circuit Emulator)ではこのモニタ機能が内蔵されています。つまり、任意の個所で処理をストップし、マイコン周りで発生している事象を調べることができるようになっています。

本章は、このような機能を実現するモニタを自分で作ってしまおう、というのがねらいです。モニタの基本的な動作のしくみを把握すると、いろいろな機能を付けたくってきます。オリジナルのモニタを作り込んでみるのも、おもしろいと思います。それでは、まず、仕様を考えてみましょう。

### 2. どのようなモニタを作るか

仕様を考える

ここに示す機能には賛否両論あるかと思いますが、最低限「使える」機能を実装しようと思います。

プログラムのダウンロード機能

プログラムをダウンロードして実行するまでの一連の処理を行うようにします。認識する実行形式としてはルネサステクノロジーのFDTでも認識可能なモトローラSフォーマットにします。

メモリ読み出し機能

任意のメモリ空間を読み出して表示させる機能を付けます。

メモリ書き込み機能

バイト単位で書き込み可能な機能を付けます。

レジスタ・ダンプ

レジスタを一覧表示させる機能を付けます。意味不明な状態で停止してしまったときに、レジスタ一覧を見ることができると、ソース・ファイルと対比してまちがいを修正できることもあります。この機能は、そのような用途をねらっています。また、マイコンの状態(SH-2ではSR)を見て、マイコンの実行状態や、割り込みの状況などの判断をユーザに促すことができます。

モニタからリターンする機能

モニタから抜け出す機能も付けようと思います。たとえば、起動直後の状況をモニタで確認してから、アプリケーションの中核に突入するような場合、この機能が必要になります。

ユーザからの入出力はシリアル・ポート

付録基板をターゲットにするので、シリアル・ポートを使います。今回は、前章で説明したnewlibの入出力関数を利用します。

ユーザの割り込みで起動できる機能

ダウンロードしたプログラムの実行中にシリアル・ポートから入力があった場合に、モニタが起動するようにします。この機能がこれから作るモニタの中核となります。この機能により、電源を切ることなく、動作の状況把握が可能になります。

簡単なシェル機能

コマンドを入力したり、アドレスを入力するときにまちがえてしまうことがあるでしょう。そのような場合に対処できる、単純な補正機能を付けようと思います。また、コマンドのヒス

トリ機能を付けて、以前に発行したコマンドを簡単に再呼び出しできるようにします。

\* \* \* \*

以上の機能をどのように実現するか、もう少し詳しく考えてみましょう。

### コマンドを考える

ここで考えているモニタは、コマンド・ベースで動作するようにします。そこで、以下のようなコマンドを定義します。

#### メモリ読み出しコマンド系列

この系列には、rb, rw, rl コマンドを含みます。これらのコマンドには、読み出しアドレスとなる16進数の引き数を一つ要求します。

rb コマンドは、8ビット単位で引き数に与えられたアドレスを読み、データを表示します。rw, rl コマンドに関しては、それぞれ、16ビット、32ビット単位での読み出しです。

#### メモリ書き込みコマンド系列

この系列には、wb, ww, wl コマンドを含みます。これらのコマンドには、書き込み先アドレスとなる16進数と、書き込み対象のデータを引き数として要求します。

それぞれのコマンドは、与えられた引き数を元に、8ビット、16ビット、32ビット単位で引き数のアドレスに書き込みます。

#### メモリ・ダンプ・コマンド系列

このコマンド系列には、db, dw, dl コマンドが含まれます。このコマンドでは、16進数の引き数が与えられたときには、それを開始アドレスとして、256バイト分、整形してダンプします。引き数が与えられない場合は、前回ダンプした開始アドレスから再度ダンプします。

#### ダウンロード&ラン・コマンド系列

このコマンド系列には、dr コマンドのみが含まれます。dr コマンドは、引き数なしで実行され、ダウンロードを促すプロンプトを出し、ダウンロードの完了後、対象の開始アドレスに分岐するようにします。

### ヘルプ・コマンド系列

ヘルプはhelp コマンドだけを含むようにします。help コマンドに引き数を与えて、コマンド系列ごとに用意するなど、拡張も考えられますが、ここではhelp コマンドですべての項目を表示するようにします。

### レジスタ・ダンプ・コマンド系列

この系列には、sr コマンドだけを含みます。sr コマンドはモニタが起動する直前のレジスタ状態を表示するようにします。

### 終了コマンド系列

em コマンドでモニタから抜けられるようにします。

### 詳細な設計

#### ▶ プログラム全体の流れ

前述の全体仕様から、図1のような動作フローを実現するプログラムを作ればよいことになります。

まず、モニタはmain関数で起動します。モニタはプロンプトを出力して、コマンドを待ちます。プログラムのdrコマンドが入力されると、モトローラSフォーマットを認識しながら、プログラムをダウンロードします。ここで、もうひとつふうします。ダウンロードされるプログラムはあくまでも関数とします。つまり、スタートアップ・ルーチンなしの単純な引き数と返り値なしの関数とします。こうすることで、モニタに制御を戻すことが可能になり、複数のプログラムを何度もダウンロードと実行を行うことが可能になります。

ダウンロードしたプログラムの実行中にモニタを呼び出すには、SCIの受信例外をダウンロードした関数を呼び出す直前に有効にしておきます。そうすれば、その関数を実行中にシリアル・ポートに入力があると例外が発生し、モニタを呼び出すことが可能になります。

レジスタ・ダンプ機能については、レジスタのスナップ・ショットを取るため、C言語では記述できません。この部分については、アセンブリ言語で記述します。また、この機能の利用方法は、ダウンロードされた関数のあるタイミングでのレジスタの状態を見るわけなので、シリアル・ポートからの入力で

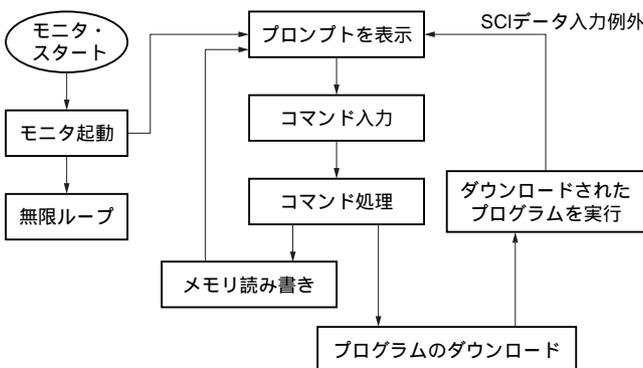


図1 モニタの処理フロー

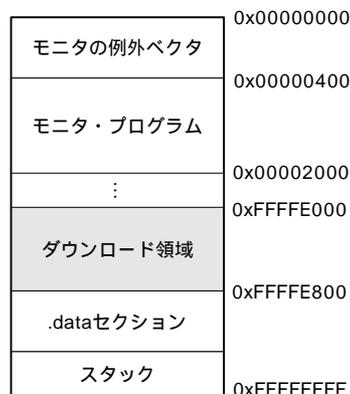


図2 モニタを付録基板に実装した場合のメモリ配置