

マイコンを正しく操縦するための作法

基礎から学ぶC言語講座

岡田 好一

Yoshikazu Okada

第11回 割り込みプログラムの使い方



割り込みと言えば、入出力と通信です。今回はシリアル・インターフェースと、R8C/Tinyの特徴の一つであるデータ用フラッシュ・メモリを使ってみます。

応用例として、少し風変わりな規則の迷路を作ってみます。

フラッシュ・メモリの特性

▶電源が切れてもデータが消えないデータ・フラッシュ領域

本稿で使っているR8C/1Bにはデータ用として、アドレス2400hから連続した2組の1Kバイトのデータ・フラッシュ(メモリ)が内蔵されています(図11-1)。RAMは電源が切れれば内容が失われてしまうので、データを続けて使うのならフラッシュ・メモリを利用することになります。

フラッシュ・メモリの初期値はFFh(=255)で、1バイトずつ書き込むことができ、消去は1ブロック(1Kバイト)ずつです。

▶二つの書き換えモードEW1とEW0の違い

単純な書き込みと消去は簡単ですが、わざわざ割り込みの話を持っていたのは書き換えの特性にあります。R8C/1Bのハードウェア・マニュアルによると、

ROM上のプログラムから書き換え可能なEW1モードでは、書き込み時と消去時にはCPUがホールド状態になってしまいます。

RAM上に書き換えプログラムを展開するEW0モードではCPUは動作を続けます。ただし、クロック周波数を5MHz以下にする必要があります。また、使用できない命令(UND, INTO, BRK)があります。

時間が厳しい应用ならEW0モードを選択しないといけませんが、プログラミングはEW1のほうが楽ですし、貴重なRAMを消費しません。

ですから、できるならばEW1モードを使いたいのです。

1バイトの書き込み時間は標準値50μs、最大値400μsで、割り込みの最大待機時間になります。この程度だと、我慢できる应用ソフトウェアも多いでしょう。

図11-1 R8C/1B(ROM16Kバイト)のデータ・フラッシュ・メモリのブロック
1Kバイトのデータ・フラッシュ・メモリがアドレス2400hから連続して2組内蔵されている

2400h	ブロックA
27FFh	
2800h	ブロックB
2BFFh	

Keyword 1

フラッシュ・メモリ

フラッシュ・メモリ(flash memory)は、一定の大きさのブロック単位で消去するEEPROM(Electronically Erasable and Programmable Read Only Memory)の一種です。

1バイトずつ書き込みできるNOR型は数千~数百Kバイトの容量のものが多く、R8C/1BのROMも16Kバイト(プログラム)+2Kバイト(データ)の構成です。

デジタル・カメラのデータ格納用などで使われるNAND型のフラッシュ・メモリはずっと大容量ですが、ブロック書き込みしかできないそうです。

フラッシュの語源は写真機のフラッシュだそうです。実

際、かつてよく使われた紫外線消去型のEPROMの内容は写真機のストロボ光で消去できます。

フラッシュ・メモリのおかげで、ROMの扱いが非常に楽になり、アマチュアでも手軽にR8C/Tinyなどのマイコンが使用できるようになりました。

▶ ブロック消去で引き起こされる問題とイレーズ・サスペンド機能

問題はブロック消去で、標準値0.2秒、最大値は9秒にもなります。さすがに9秒のCPU停止はきつし、さらに深刻なのはインターバル・タイマ(例題プログラムでは10ms)が空回りすることです。

つまり、同一の割り込みは単純に待機させられるだけで、記憶されません。CPUが停止中にインターバル・タイマから何回割り込み依頼が入っても、再開時には1回の割り込み処理が起こるだけです。これでは計時ができません。

そこで、イレーズ・サスペンド機能が用意されています。これを用いると、割り込みには100 μ s程度で応答します。ただし、消去は途中で止まってしまうので、再開の操作が必要です。再開はループを作るだけなので簡単です。

● 消去可能回数

フラッシュ・メモリには寿命があります。マニュアルではデータ・フラッシュには**最低限10000回の消去が保証**されています。それでも、マイコンのプログラムでうかつに消去を繰り返すと、あっという間に寿命が来てしまいそうです。

制限があるのは消去回数であって、書き込み回数ではありません。フラッシュ・メモリのブロックは1Kバイトあるわけですから、数バイト書くごとにブロックを消去するとむだが生じます。同じアドレスの2度書きは禁止ですが、空き領域に次々にデータを書いていけば、満杯になるまで消去は必要ありません。

そのため、ルネサス テクノロジからフラッシュ・メモリを長期に使用するための例題プログラムとユーティリティが用意されています。しかし、ここでは工夫が可能なことを述べるにとどめておきます。例題では、毎回ブロックを消去してからデータを書き込む素朴な使い方にします。

● エミュレータのファームウェアの移動

今まで、本連載ではE8エミュレータのファームウェアはデータ・フラッシュ上に確保していました。今回はデータ・フラッシュを使うのですから、プログラムと同じユーザ・フラッシュ領域にファームウェアを転送することにします。

ファームウェアを書き込む際に、関係のないフラッシュ・メモリまでが初期化されることに注意が必要です。

● フラッシュ・メモリ書き込みルーチン

(`fm_write()`, `fm_erase()`)

基本部分は、1バイト書き込みと、ブロック消去の2サブルーチンがあればよく、メーカ推奨の手順が参考プログラムに書かれています。

今回例示したプログラムでも、マニュアルそのままの手順を採用しています。最適化を阻止する`asm()`関数の配置以外は、特に注意する点はありません。消去ルーチンはイレーズ・サスペンド機能を使っています。

プログラムが複雑になるので、例題プログラムではエラー処理は行っていません。通常は必要ですから、書き込みの後にマニュアル通りのエラー処理を行います。

R8Cのシリアル通信モジュール UART0とUART1

● R8C/1BのUART0とUART1

スタータキットにはRS-232Cへのレベル変換ICが付属していて、市販のコネクタをはんだ付けし、回路図でR62とR63と記入されたジャンパを接続するとパソコンなどと通信が可能になります(図11-2)。

面白いことに、通常使用されるUART0ではなく**UART1**に接続されています。UART1は店頭で手に入りやすいR8C/15では使用できません。UART1が採用された理由は、開発で使われるブート・モードで

Keyword 2

インターバル・タイマ

インターバル・タイマ(interval timer)は、一定時間ごとにフラグを立てる、または割り込みするタイマです。R8C/1BではタイマXのタイマ・モードなどが使われます。

本稿の例題プログラムでは、10msごとにタイマXが割り込みを要求し、その処理ルーチンではスイッチなどの状態を確認しています。

では、10msごとに必ず処理ルーチンが起動されるかというとそうとも限りません。割り込みできない状態が長時間続いた後に受け付け状態に戻っても、処理ルーチンは1回起動するだけです。

割り込み要求は残っていますが、要求の回数を数える機能はないからです。

インターバル・タイマを利用して計時が可能なのは、保留の無いことが明らかなきのみです。

本当の経過時間が知りたいのなら、時計機能の内蔵されたマイコンを採用すべきです。