

RXマイコン用統合開発環境 HEW 評価版を使い切る!



関連データ

村藤 真幸

マイコンの開発環境の評価版には何らかの制限があることが多い。ここでは本誌5月号付属RX62Nマイコン基板を
活用するため、統合開発環境HEWの評価版の制限の範囲内で、工夫して使いこなす方法について解説する。(編集部)

ルネサス エレクトロニクスから提供されているRXマイコンの結合開発環境HEW (High-performance Embedded Workshop) 評価版は、初回ビルドから60日を経過するとリンクできるロード・モジュール(実行可能な機械語ファイル)のサイズが128Kバイトに制限されます。128Kバイトに制限されると、例えばグラフィックス表示をしたい場合、QVGAサイズでも153.6Kバイトのデータ・サイズとなるため、画像データだけで制限を超えてしまいます。

本誌2011年5月号の付属RX62Nマイコン基板で評価を始めた場合は、既にこの制限を受けている読者も多いでしょう。しかし評価を継続したいという要望もあると思われるので、ここでは制限を受けた後でも128Kバイトを超えるプログラム・コードやデータを扱う方法を紹介しします。

なおこの制限を回避する方法としてGNUコンパイラを使うこともできます。インドKPIT社から提供されるRX用gccコンパイラについては、近号で紹介する予定です。

1. LCD表示用の2次元配列が制限を超えている

▶対処法：配列をポインタで扱う

RX62Nマイコンの応用事例として、高機能タイマと外部バスDMAコントローラを活用したLCDによるグラフィックス表示があります。このとき、リスト1(a)のようにLCD画面表示用のフレーム・バッファを2次元配列で宣言することで、X座標とY座標をそのまま配列の要素としてアクセスできるようになります。この例の場合、フレーム・バッファ用のLCDセクションが実体としてマッピングされるので、128Kバイトの制限を超えます。

そこで、フレーム・バッファを2次元配列として確保せずにリスト1(b)のように宣言します。実際にフレーム・バッファにアクセスする場合は、リスト1(c)のようにextern指定子を付けて2次元配列として定義すれば、アクセスできます。

図1にプロジェクトarray_pointerのマッピング状況を示します。このように配列をポインタで扱えるというのは、C言語の自由度の高さを示す典型的な例ですが、ポインタは配列の範囲を越えても操作できてしまうので、アクセス範囲に十分注意してプログラムする必要があります。

リスト1 フレーム・バッファの定義

```
#pragma section LCD
volatile unsigned short global_lcd_framebuffer[240][320];
#pragma section
```

(a) 2次元配列をまるごとリンクする

この宣言で153.6Kバイトが実体としてリンク対象になる。セクション名は指定したLCDにデフォルトのBxxxxx_2が連結されてBLCD_2となる

```
#pragma section LCD
volatile unsigned short global_lcd_framebuffer;
#pragma section
```

(b) アドレス情報のみリンクする

この宣言では2バイトの大きさのglobal_lcd_framebufferが実体としてリンク対象になる。global_lcd_framebufferは変数名でありアドレス情報がある

```
extern unsigned short global_lcd_framebuffer[240][320];
/* 点描画 */
void dot(short x, short y, unsigned short color){
    global_lcd_framebuffer[y][x] = color;
}
```

(c) フレーム・バッファへのアクセス例

global_lcd_framebufferは2次元配列で宣言されたとして扱える