

第4章 ルネサス製コンパイラとGCCで効き目を確認してみた

# ちょこっと実験! 「サイズ」の最適化

中村 憲一

PRresetPRG	FFFC0000	FFFC0034
D_BMP_DEFAULT	FFFC0100	FFFC3135
D_BMP_HIT	FFFC3138	FFFC526D
D_BMP_MISS	FFFC5270	FFFC73A5
D_BMP_STATE0	FFFC73A8	FFFC94DD
D_BMP_STATE1	FFFC94E0	FFFCB615
D_BMP_STATE2	FFFCB618	FFFCD74D
D_BMP_STATE3	FFCD750	FFFCF885
D_BMP_FONT	FFFCF888	FFFD2BED
C_2	FFFD2BEE	FFFE4B87
C	FFFE4B88	FFFE4CCF
C\$DSEC	FFFE4CD0	FFFE4CF3
C\$BSEC	FFFE4CF4	FFFE4D0B
C\$VECT	FFFE4D0C	FFFE510B
D	FFFE510C	FFFE5207
D_2	FFFE5208	FFFE5249
D_1	FFFE524A	FFFE529A
P	FFFE529B	FFFE529A
PIntPRG	FFFE529B	FFFE529A
W_1	FFFE529B	FFFE529A
W_2	FFFE529B	FFFE529A

ここから  
ここまで

図1 before…スピード優先かつ最適化レベルmaxでビルドしたときのメモリ割り当て(セクション)情報  
 エンド・アドレスとスタート・アドレスの差0xFFFD30D-0xFFFC0000+0x1=0x2D30E (185102バイト)がプログラムのサイズ

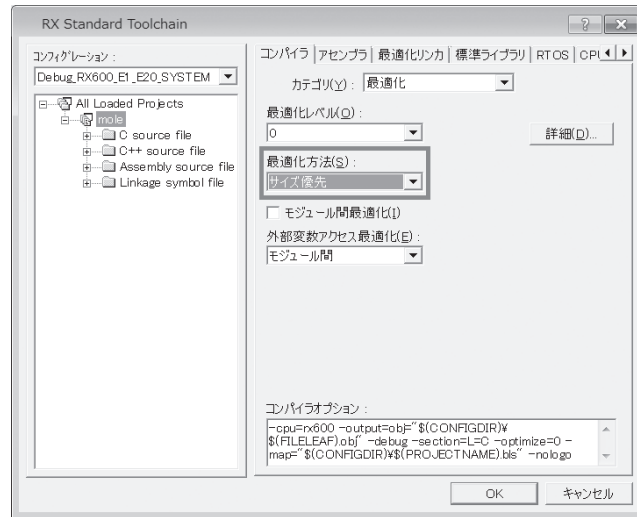


図2 コンパイラの最適化の設定画面で、サイズ優先と設定する

スピード優先かつ最適化レベルmaxでビルドしたときのプログラム本体のサイズは、

0xFFFD30D-0xFFFC0000+0x1=0x2D30E (185102バイト) となっています。

最適化オプションを使用して小さくしてみましょう。HEWの場合は、「ビルド」メニューからRX Standard Toolchainを選択し、RX標準ツールチェーンの設定画面を表示します。「コンパイラ」タブのカテゴリから最適化を選択し、図2に示すように「スピード優先」から「サイズ優先」に変更します。最適化レベルは0を選択します。

F7キーまたはビルド・メニューからビルドします。ルネサス純正デバッグE1が接続されている場合は、新しくビルドしたバイナリをダウンロードするかを尋ねてきますのでダウンロードすると、セクション情報も更新されます。

サイズ優先かつ最適化レベル0でビルドしたときのプログラム本体のサイズは、図3に示すように、

0xFFFE8C5F-0xFFFC0000+0x1=0x28C60 (167008バイト)

本稿では、CPU メーカー純正コンパイラ HEW (ルネサスエレクトロニクス) と定番オープンソース・コンパイラ GCC を例に、プログラム・サイズの最適化オプションの効き目を簡単に調べてみます。(編集部)

## CPU メーカー純正コンパイラで試す

● RXマイコンの例  
 HEWでビルドしたプログラムの大きさを見るためには、表示メニューからマップを選択し、マップ種別の選択で「Map Section Information」を選択します。すると、図1のようにセクション情報が表示されます。