

第6章 限られたハードウェアをフル活用！ できるだけ多くの処理を高速に！

純正ツールHEWでコード・サイズと処理時間を最適化

藤澤 幸穂 Yukiho Fujisawa

安価なマイコンをできるだけ高速に動かして、できるだけ多くの仕事をさせるのが、エンジニアの腕の見せどころです。ルネサスの純正開発環境HEWは、チェックを入れるだけで、コード・サイズを小さくしたり、処理速度を上げたりできる「最適化機能」を備えます。その効果と使い方を説明します。

HEWがもつ「最適化」機能とは

● 意味のない処理を自動的に省く

最適化(optimize)とは、意味のない処理をやめて、より実行時間を短くしたり、コード・サイズを小さくしたり、小さなROMに効率良く収まるようにする処理です。

例えば、プログラム上は記述があっても、意味がなければ機械語を作成せずに済ませると実行時間は短くなり、コード・サイズも小さくなります。

例えば次のソースを見てください。

```
void main(void)
{
    int i,j;
    for(i=0,j=0;i<100;i++)j+=i;
}
```

この場合 i と j はともに main 関数だけで使うロー

カル変数です。main 関数が終了したら消滅して構いません。返却値として利用されることもありません。果たしてこのプログラムの演算は必要でしょうか？

基本的にはそのような演算は不要です(作者の意図と合っているかは分からないが)。ならばmain関数は何もしないで「呼び出し元に戻る」だけでよいはず。このように不要と思われる処理を徹底的に削るのが最適化です。

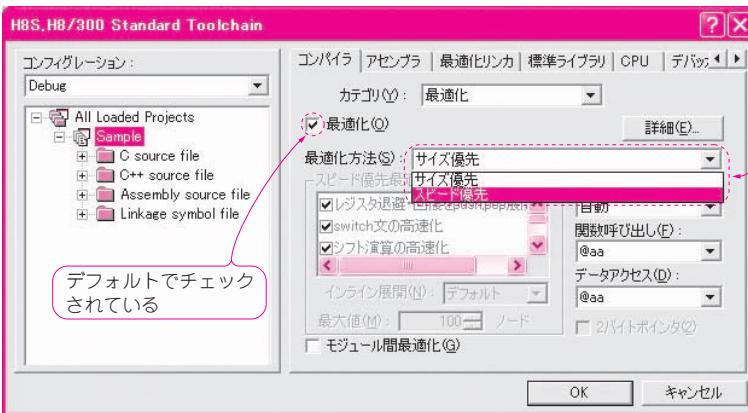
● 処理時間とコード・サイズはトレードオフ

ただし実行速度とサイズの改善は相反する場合があります。

次のソースは100回の繰り返しです。

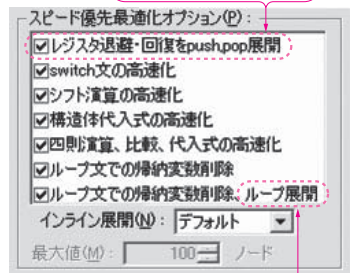
```
for(i=0;i<100;i++){
    処理;
}
```

繰り返しには条件判断と分岐処理が必要で、その分、実行時間がかかります。これを、



サイズ優先がスピード優先かを選ぶ

H8S, H8SXでは無効



ループ展開の最適化はシステムの性能を大きく左右する

デフォルトでチェックされている

図1 HEWのビルド・オプション次第で性能に違いが出る