

M16C/64A グループ

R01AN0044JJ0100

Rev.1.00

2010.07.30

High-performance Embedded Workshop スタートアッププログラムの説明

1. 要約

この資料は、High-performance Embedded Workshop(以下、HEW)で新規プロジェクトワークスペースを作成した際に、HEWが生成するソースファイルについて説明します。

2. はじめに

この資料で説明する応用例は、次のマイコン、条件での利用に適用されます。

- マイコン : M16C/64A グループ
- HEW のバージョン : Version 4.07
- M16C Series, R8C Family C Compiler[M3T-NC30WA] : V.5.45 Release 01

本アプリケーションノートは、上記グループと同様のSFR(周辺機能制御レジスタ)を持つM16Cファミリマイコンでも使用できます。ただし、一部の機能を変更している場合がありますのでユーザーズマニュアルで確認してください。また、本アプリケーションノートで説明しているプログラムを使用される場合は十分な評価を行ってください。

3. 概要

本アプリケーションノートは、HEWで「C source startup Application」を選択して、新規プロジェクトワークスペースを作成したときに生成されるソースファイルについて説明します。

3.1 HEWが生成するソースファイル

表 3.1にHEWが生成するCソースファイルを、表 3.2にHEWが生成するヘッダファイルを示します。

表 3.1 HEWが生成するCソースファイル

ファイル名	概要	内容
プロジェクト名.c	メインファイル	メイン関数を記載するソースファイルです。
firm.c	ファームウェア用の定義ファイル	OnChipDedebugger選択時のFoUSB/E8のfirmが使用するプログラム領域及びワークスペース領域をダミーとして確保します。
firm_ram.c	ファームウェア用のRAM定義ファイル	
fvector.c	固定ベクタテーブルの定義ファイル	固定ベクタテーブルが定義されています。 OFS1番地および、IDコードを設定しています。 ノンマスカブル割り込み処理を使用する場合は、固定ベクタテーブルを変更してください。
heap.c	ヒープ領域定義ファイル	使用するヒープ領域を定義しています。ヒープサイズ(<code>__HEAPSIZE__</code>)はプロジェクト作成時、設定した値が <code>cstartdef.h</code> に定義されています。ヒープサイズを変更する場合は、 <code>__HEAPSIZE__</code> の値を変更してください。
initsct.c	RAM初期化ファイル	RAMを初期化する処理が記載されています。ユーザが新たに追加した領域を初期化したい場合、初期化する領域を追加してください。
intprg.c	可変ベクタテーブルの定義ファイル	可変ベクタテーブルおよび、割り込み関数(空関数)が定義されています。 マスカブル割り込みを使用する場合は、割り込み関数に処理を追加してください。
resetprg.c	C言語用スタートアッププログラムファイル	main関数を実行するまでの初期設定を行っています。 ユーザブート機能を使用する場合はこのファイルでユーザブートコード領域を設定してください。

表 3.2 HEWが生成するヘッダファイル

ヘッダファイル名	概要	使用内容
cstartdef.h	スタックサイズ定義ヘッダ	スタックサイズやヒープサイズの定義を行っています。 リセット直後のウォッチドッグタイマの自動起動の使用/不使用を選択します。
initsct.h	セクション初期化用マクロの定義ヘッダ	initsct.cで使用するマクロの定義を行っています。
resetprg.h	各種初期設定ヘッダ	スタックサイズの設定、CPUレジスタの設定、ヒープ領域の初期化を行っています。
sfrXX.h (注1)	SFRレジスタ定義ヘッダ	使用する製品のSFRレジスタの定義を行っています。
typedefine.h	データ型定義ヘッダ	データ型の定義を行っています。

注1. XXにはマイコンのグループ名が入ります。

4. HEWが生成するソースファイルの設定

4.1 固定ベクタテーブルの設定

固定ベクタテーブルは `fvector.c` に定義されています。リセットスタート後、固定ベクタテーブルのリセットベクタに設定されている番地からプログラムを実行します。生成されたファイルでは、リセットベクタに `start` 関数の番地が設定されています。他のノンマスカル割り込みの設定も同ファイル内で設定します。

図 4.1 に固定ベクタテーブルの設定を示します。

```
#pragma sectaddress fvector,ROMDATA 0xffffdc
////////////////////////////////////
#pragma interrupt/v _dummy_int //udi
#pragma interrupt/v _dummy_int //over_flow
#pragma interrupt/v _dummy_int //brki
#pragma interrupt/v _dummy_int //address_match
#pragma interrupt/v _dummy_int //single_step
#pragma interrupt/v _dummy_int //wdt
#pragma interrupt/v _dummy_int //dbc
#pragma interrupt/v _dummy_int //nmi
#pragma interrupt/v start

#if __WATCH_DOG__ != 0
_asm(" .ofsreg 0FEH");
#else
_asm(" .ofsreg 0FFH");
#endif

_asm(" .id ""#FFFFFFFFFFFFFFF");

#pragma interrupt _dummy_int()
void _dummy_int(void);
void _dummy_int(void){}
```

固定ベクタテーブル
各割り込み要因については、ユーザーズマニュアルハードウェア編を参照してください。

ウォッチドッグタイマ割り込みを使用する場合の変更例
extern void wdt_int_func(void);
#pragma interrupt/v wdt_int_func //wdt

リセットベクタ

オプション機能選択レジスタ (OFS1) の設定
OFS1の設定に関しては「4.9 オプション機能選択レジスタ1番地(OFS1)の設定」を参照してください。

IDコードの設定
IDコードの設定に関しては「4.10 IDコード書き込み」を参照してください。

ダミー関数の定義

図 4.1 固定ベクタテーブルの設定

4.2 cstartdef.h

プロジェクト作成ウィザードでの設定値を変更する場合は、cstartdef.h のファイルで変更してください。
図 4.2 に cstartdef.h を示します。

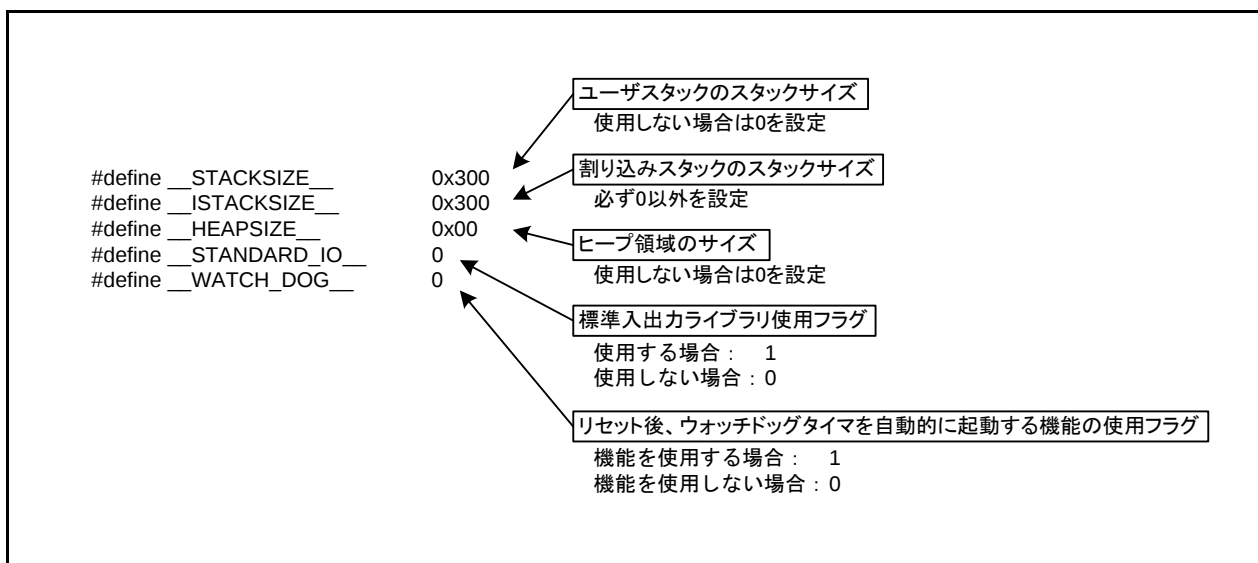


図 4.2 cstartdef.h

4.3 start関数の設定

resetprg.cにstart関数が作成されます。

図 4.3にstart関数の設定を示します。

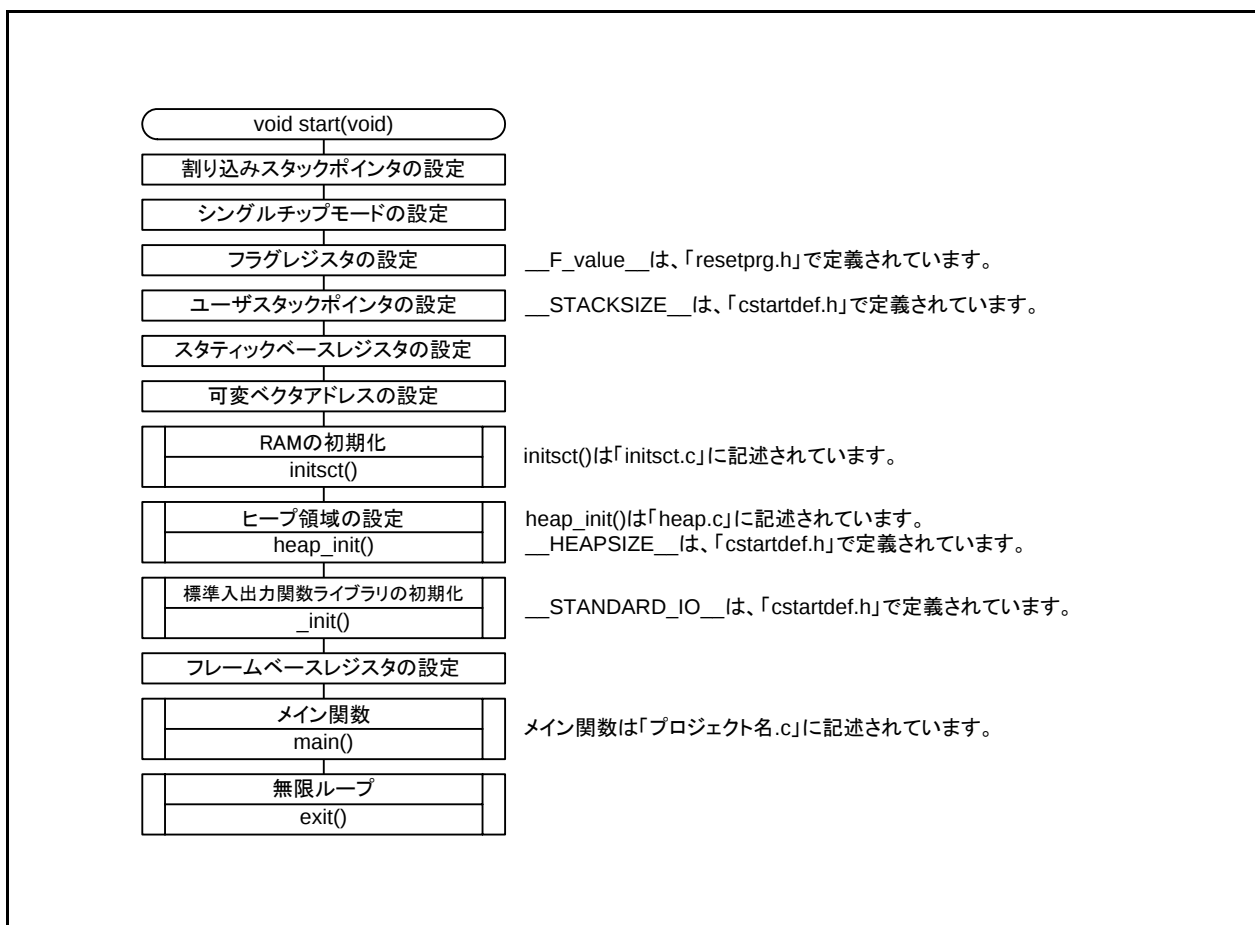


図 4.3 start関数の設定

4.4 RAMの初期化

initsct.cにRAMの初期化処理が作成されます。initsct.cではRAM領域をゼロクリアします。また、ROM領域の初期値をRAM領域に転送します。

図 4.4に新たに追加した領域をinitsct.cを使用して初期化する例を示します。

```
sclear("bss_SE","data","align");
sclear("bss_SO","data","noalign");
sclear("bss_NE","data","align");
sclear("bss_NO","data","noalign");
sclear_f("bss_FE","data","align");
sclear_f("bss_FO","data","noalign");
// add new sections
// bss_clear("new section name");
sclear("test_bss_NE","data","align");
sclear("test_bss_NO","data","noalign");
scopy("data_SE","data","align");
scopy("data_SO","data","noalign");
scopy("data_NE","data","align");
scopy("data_NO","data","noalign");
scopy_f("data_FE","data","align");
scopy_f("data_FO","data","noalign");
```

新たに追加した領域をinitsct.cを使用して初期化する場合
test_bss_NEとtest_bss_NOというセクションを追加し、
initsct.cで初期化したい場合は、左の2行を追加してください。

図 4.4 新たに追加した領域をinitsct.cを使用して初期化する例

4.5 ヒープ領域の設定

heap.cにヒープ領域を確保する処理が作成されます。

図 4.5にヒープ領域の設定を示します。

```
#pragma SECTION bss heap
_UBYTE heap_area[_HEAPSIZE_];
```

これ以降に記述される初期値のないデータをheapセクションに配置します。

ヒープ領域の確保

図 4.5 ヒープ領域の設定

4.6 メイン関数の設定

プロジェクト名.cにメイン関数が作成されます。このファイルにプログラムを追加してください。

4.7 割り込み関数の設定

intprg.cに可変ベクタテーブルが作成されます。必要に応じて、各割り込み処理はintprg.cに追加してください。

4.8 ユーザブート機能の設定

resetprg.c にユーザブート機能の設定が作成されます。13FF0h~13FFFh 番地はユーザブートコード領域です。13FF0h~13FF7h 番地に ASCII コードで “UserBoot” と設定されると、ユーザブート機能が有効になります。ユーザブート機能を使用する場合は、ユーザーズマニュアル ハードウェア編を参照してください。

図 4.6 にユーザブート機能の設定を示します。

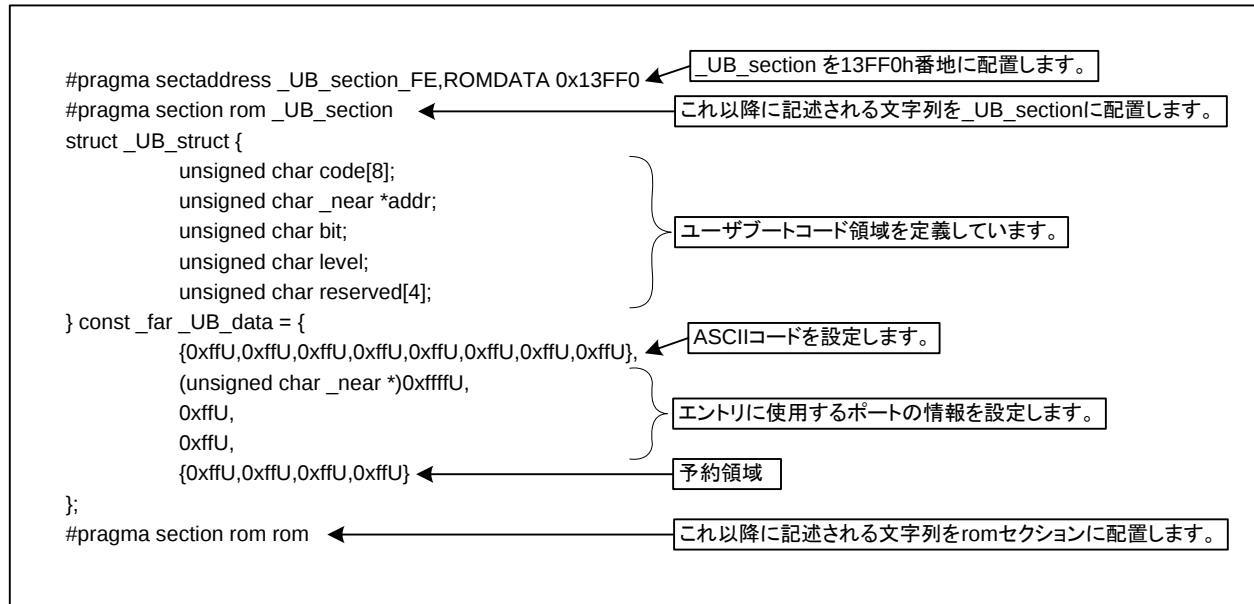


図 4.6 ユーザブート機能の設定

4.8.1 ユーザブート機能を使用しない場合

ユーザブート機能を使用しない場合は、図 4.6 で示す部分を削除または、コメントアウトしてください。また、13FF0h~13FFFh 番地はユーザブート領域であるため、プログラムを配置しないでください。

4.9 オプション機能選択1番地(OFS1)の設定

fvector.cにオプション機能選択1番地(OFS1)の設定が作成されます。

図 4.7に、オプション機能選択1番地の設定を示します。

デフォルトでFFhを設定しています。OFS1番地の設定値を変更したい場合は下記の部分にある値を変更してください。



```
_asm(\".ofsreg 0FFH\");
```

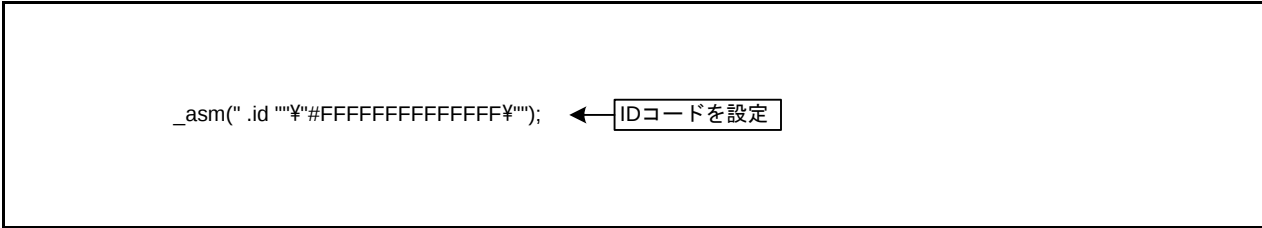
図 4.7 オプション機能選択1番地(OFS1)の設定

4.10 IDコード書き込み

fvector.cにIDコードの書き込み処理が作成されます。

図 4.8に、IDコード書き込み処理を示します。

デフォルトでFFFFFFFFFFFFFFFhを設定しています。IDコードの設定を変更したい場合は、下記の設定値を変更してください。



```
_asm(\".id \"¥\"#FFFFFFFFFFFFFF¥\");
```

図 4.8 IDコード書き込み処理

5. その他の設定

5.1 ウォッチドッグタイマの設定

cstartdef.h にウォッチドッグタイマの設定が作成されます。__WATCH_DOG__ を設定することでリセット解除後のウォッチドッグタイマの起動状態を選択できます。

なお、HEW が生成するソースファイルでウォッチドッグタイマを使用する場合、以下の制限があります。

- CPU クロックは、125kHz オンチップオシレータの 8 分周以上で使用してください。
- ウォッチドッグタイマのカウントソース保護モードは使用しないでください。

上記の制限が問題になる場合、initsct.h 内にあるウォッチドッグタイマのリフレッシュ処理が間に合うようにプログラムを変更してください。

図 5.1 にリセット後、ウォッチドッグタイマを自動的に起動する機能の使用フラグを示します。

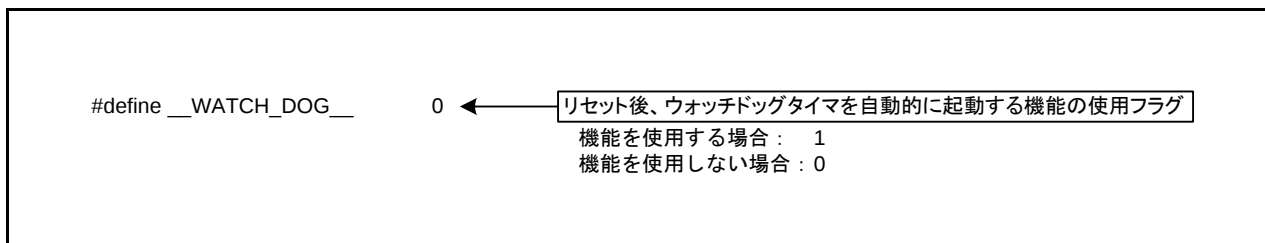


図 5.1 リセット後、ウォッチドッグタイマを自動的に起動する機能の使用フラグ

5.2 セクションの設定方法

HEW でのセクションの設定方法を示します。通常、HEW 上の [ビルド] → [Renesas M16C Standard Toolchain...] → [リンカ] → [リンカオプション] よりセクションの配置設定を行います。本アプリケーションノートでは、マップウィンドウを用いた設定例を示します。

マップウィンドウを用いた方法の場合、ワークスペースを作成する際に、プロジェクトタイプ「C source startup Application」を選択する必要があります。

ここでは、test_bss_NOセクションと test_bss_NEセクションを追加し、変数 reset_times を配置するまでを説明します。図 5.2~図 5.12 にセクションの追加を示します。

- (1) マップウィンドウを表示させます。メニューの「表示」から「マップ」を選択してください。

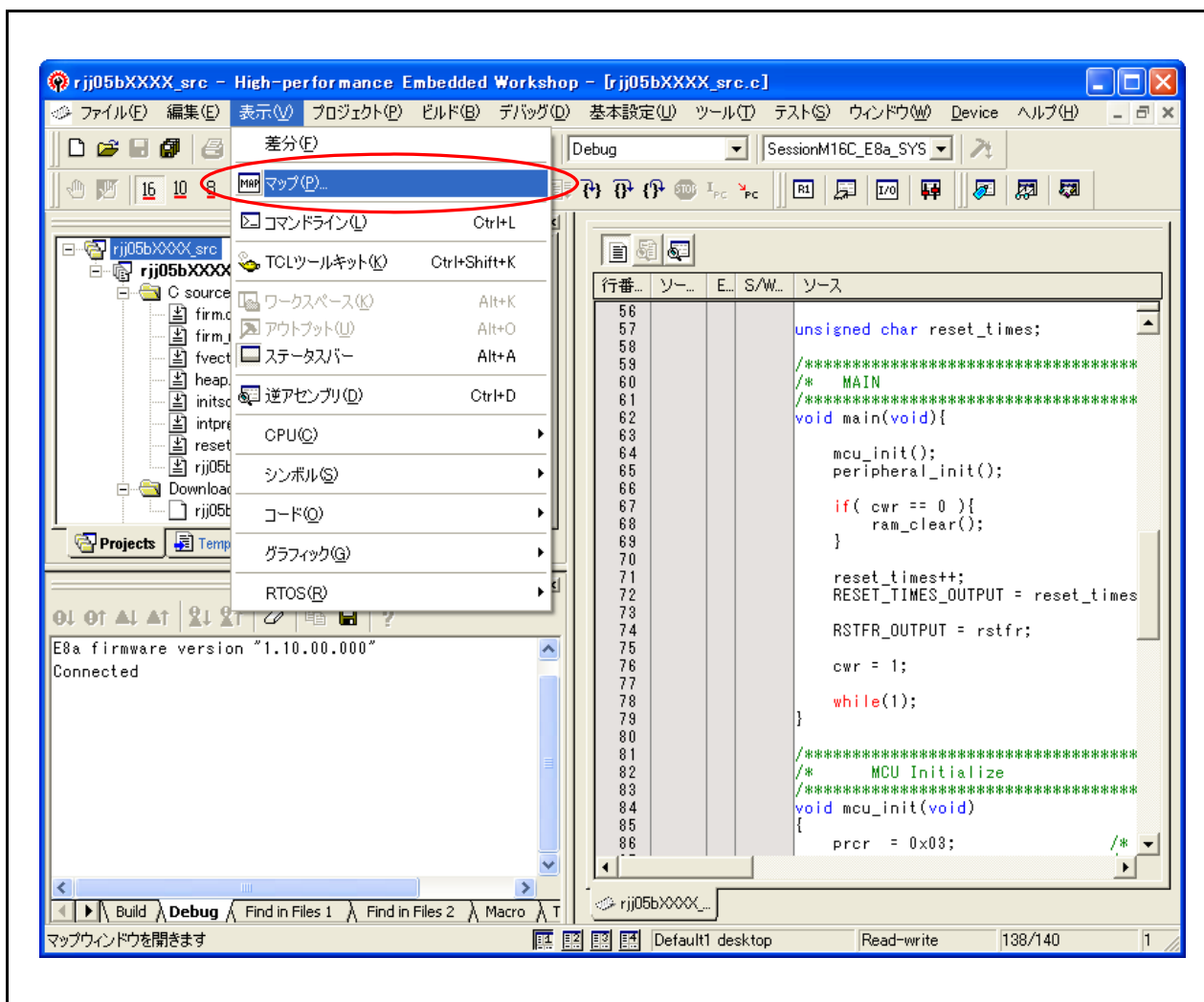


図 5.2 セクションの追加(1)

- (2) マップ種別の選択で「Map Section Information」を選択してください。
同様の手順で「Map Symbol Information」も選択してください。

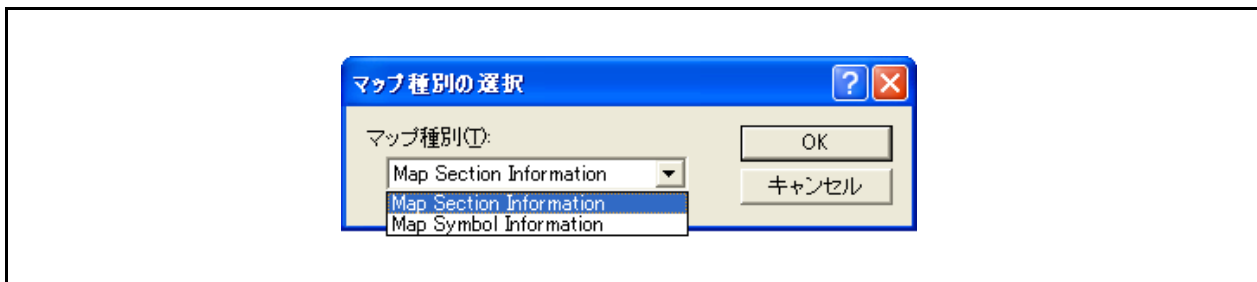


図 5.3 セクションの追加(2)

- (3) 「編集モードボタン」を押して、セクション名を編集します。

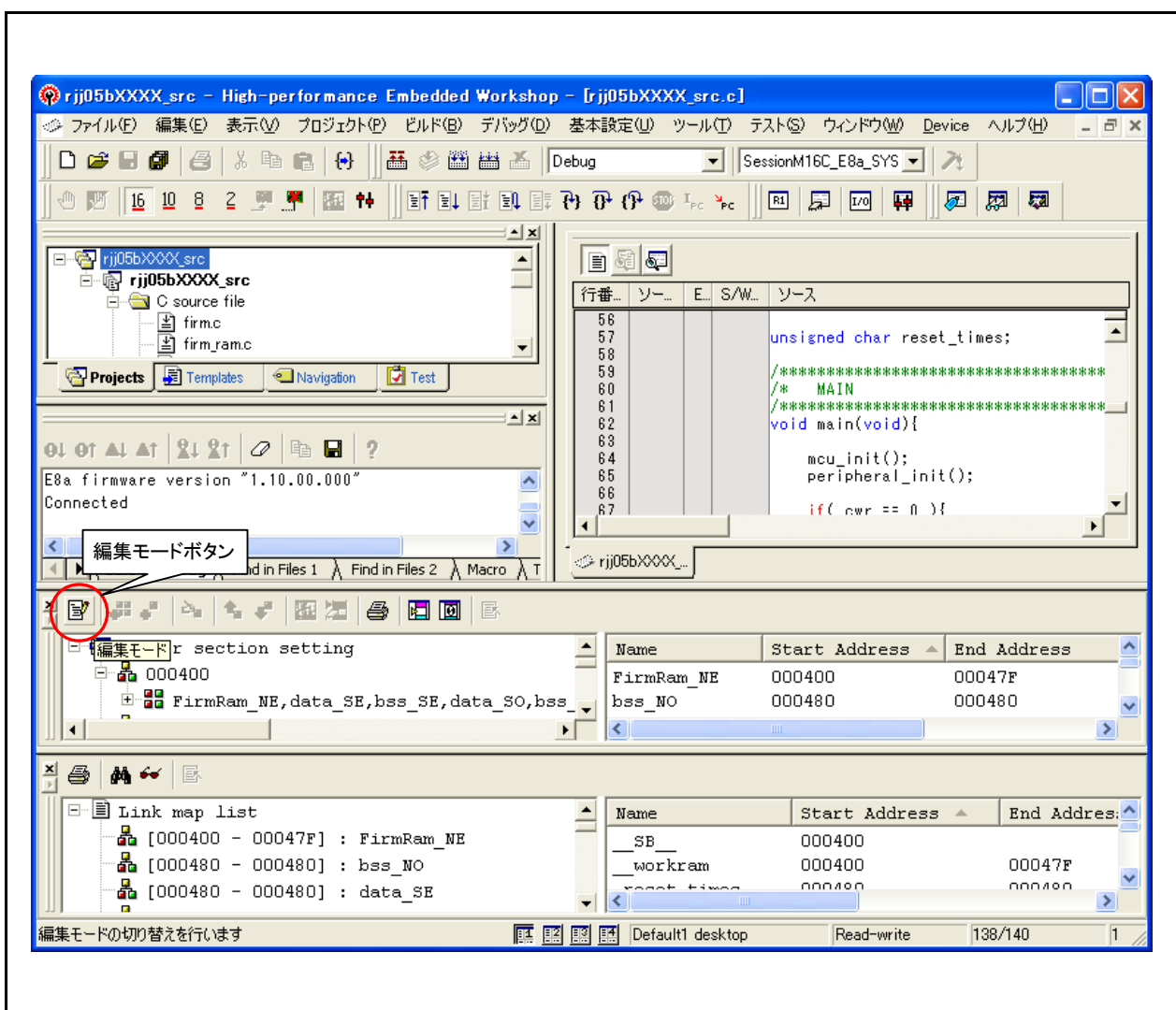


図 5.4 セクションの追加(3)

- (4) 「セクション追加ボタン」を押して、「test_bss_NO」を追加してください。

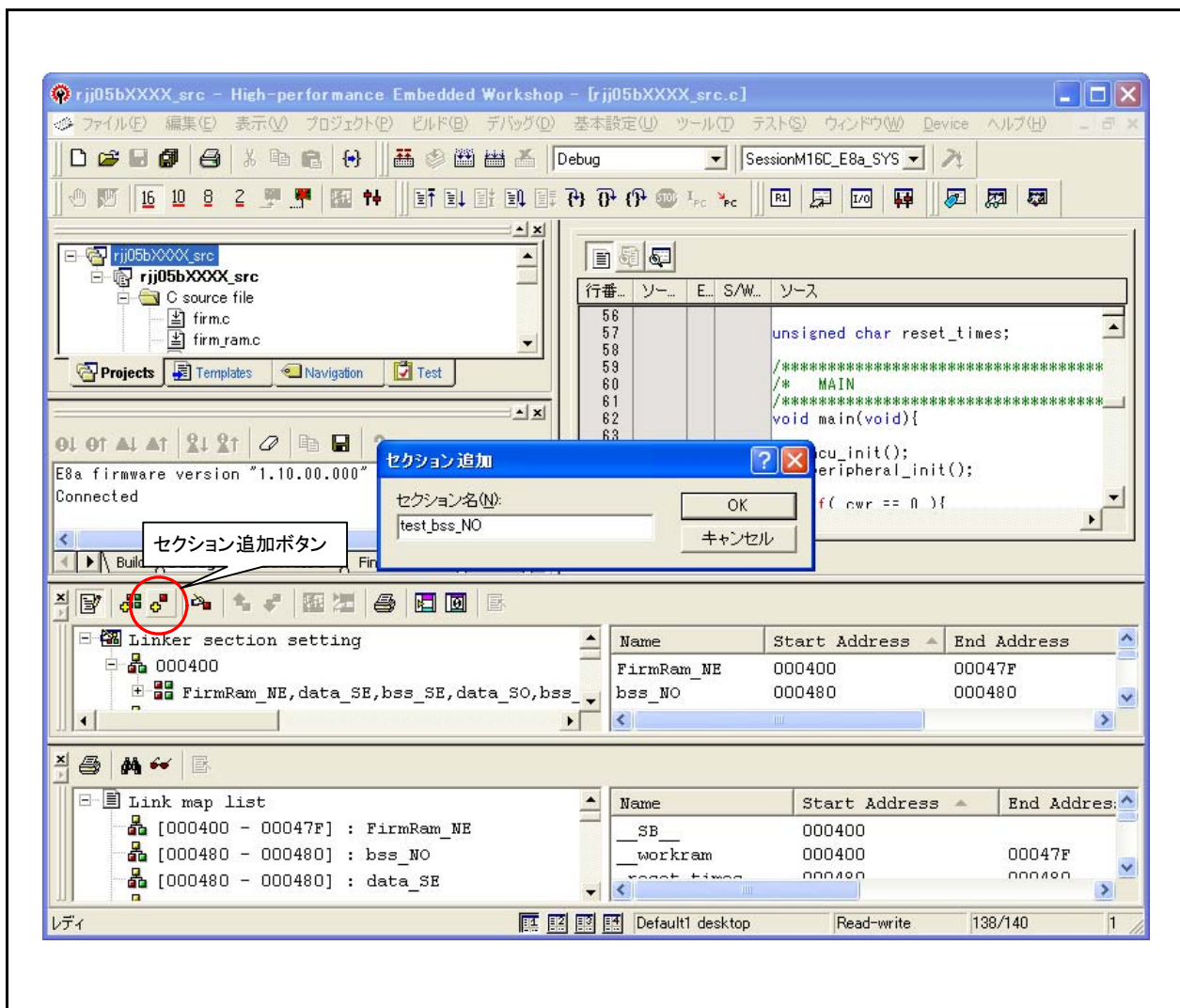


図 5.5 セクションの追加(4)

- (5) 同様の手順で、「test_bss_NE」を追加してください。

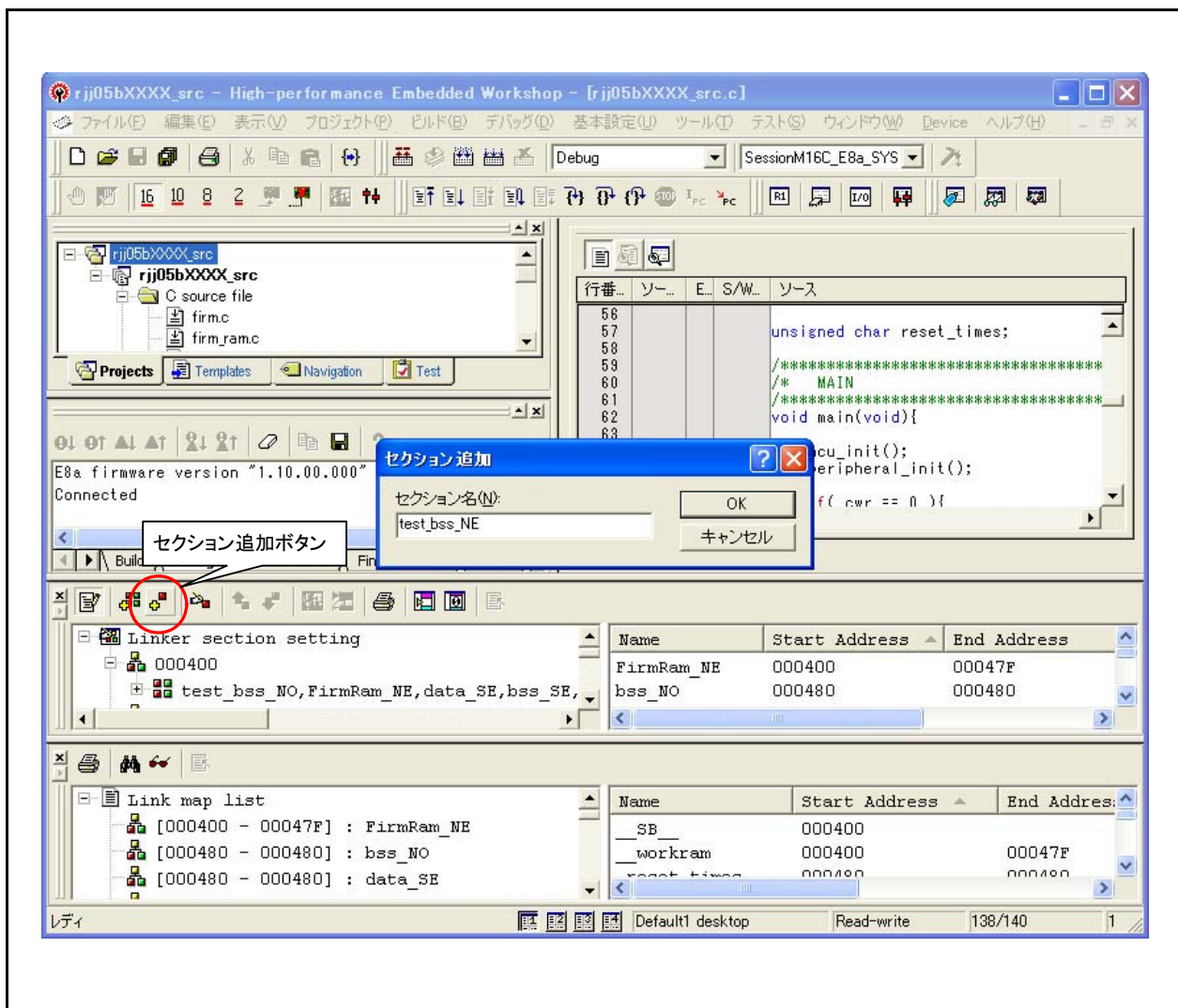


図 5.6 セクションの追加(5)

- (6) 「編集モードボタン」を押して、変更したリンクセクション情報を確定してください。

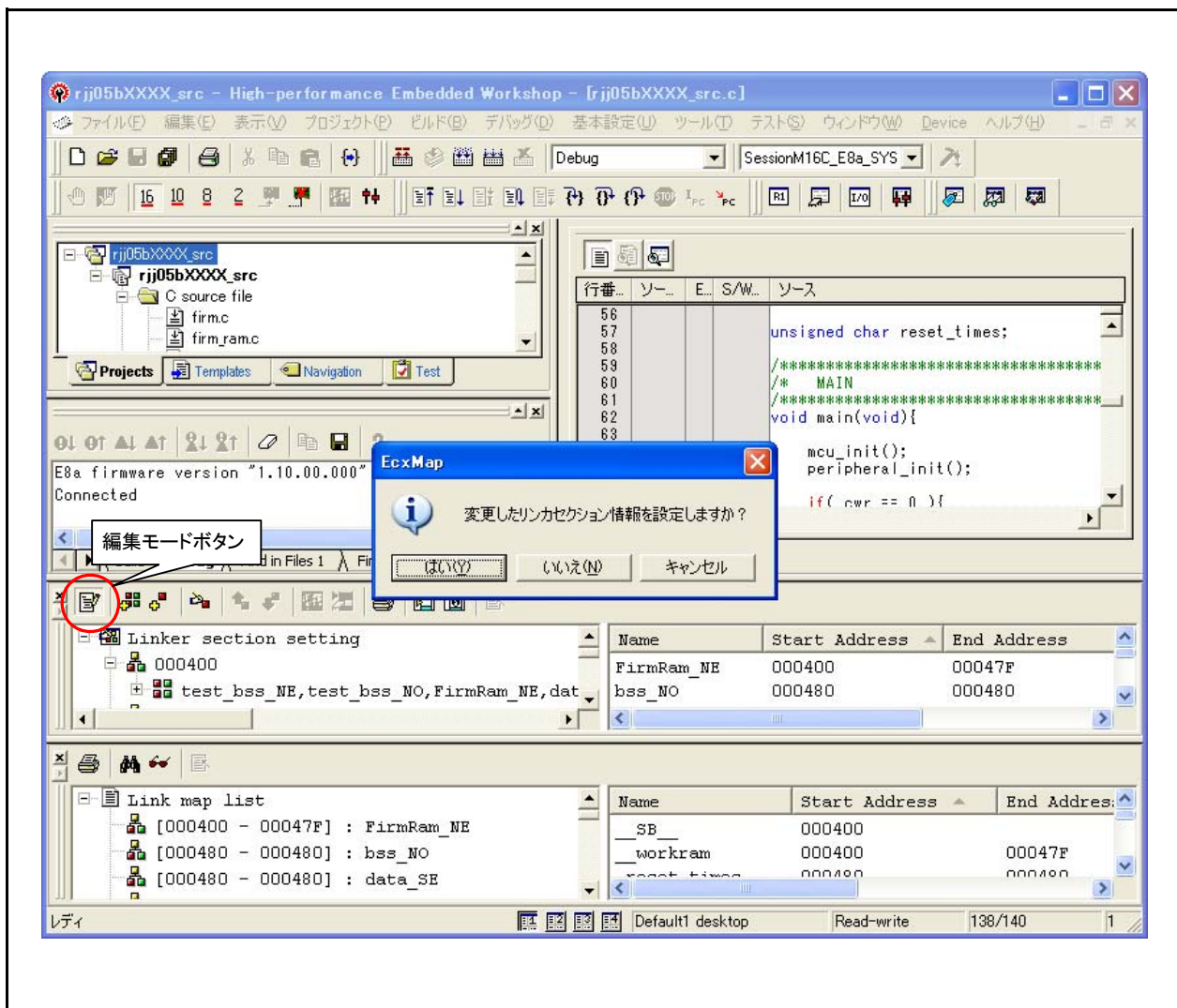


図 5.7 セクションの追加(6)

- (7) test_bss_NE セクションと test_bss_NO セクションが配置されます。

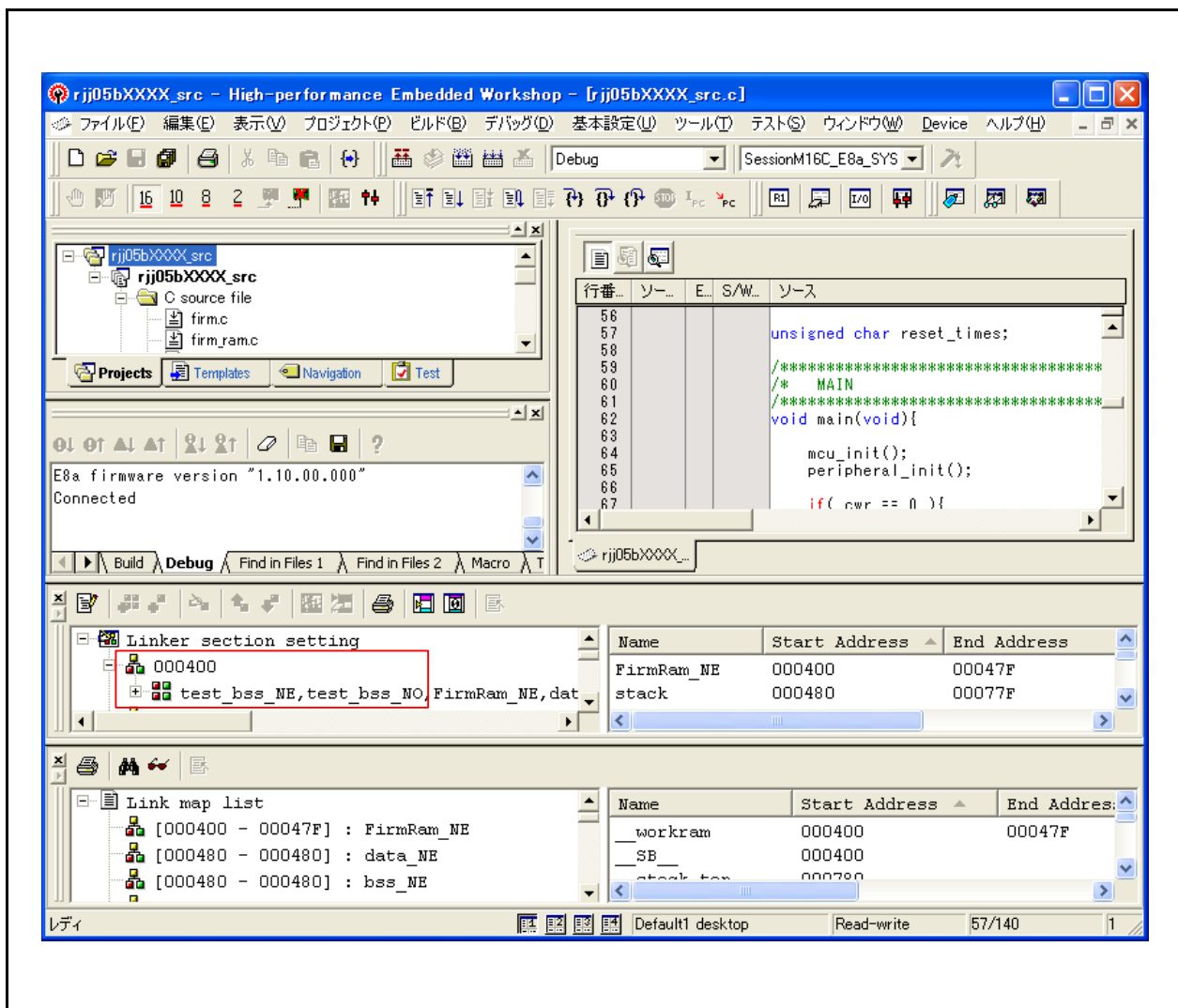


図 5.8 セクションの追加(7)

- (8) プログラムでtest_bssセクションに変数reset_timesを配置します。

```
#pragma SECTION bss test_bss
unsigned char reset_times;
```

図 5.9 変数をtest_bssセクションに配置する方法

#pragma SECTION 宣言以降に記述された初期値のない変数は、test_bssセクションに配置されます。

int i;	←	bssセクションに配置されます
#pragma SECTION bss test_bss	←	これ以降に記述される初期値のない変数(bssセクションに配置される変数)をtest_bssセクションに配置します
int j;	←	test_bssセクションに配置されます
int k = 10;	←	初期値のある変数はdataセクションに配置されます

図 5.10 変数の宣言場所によるセクション配置例

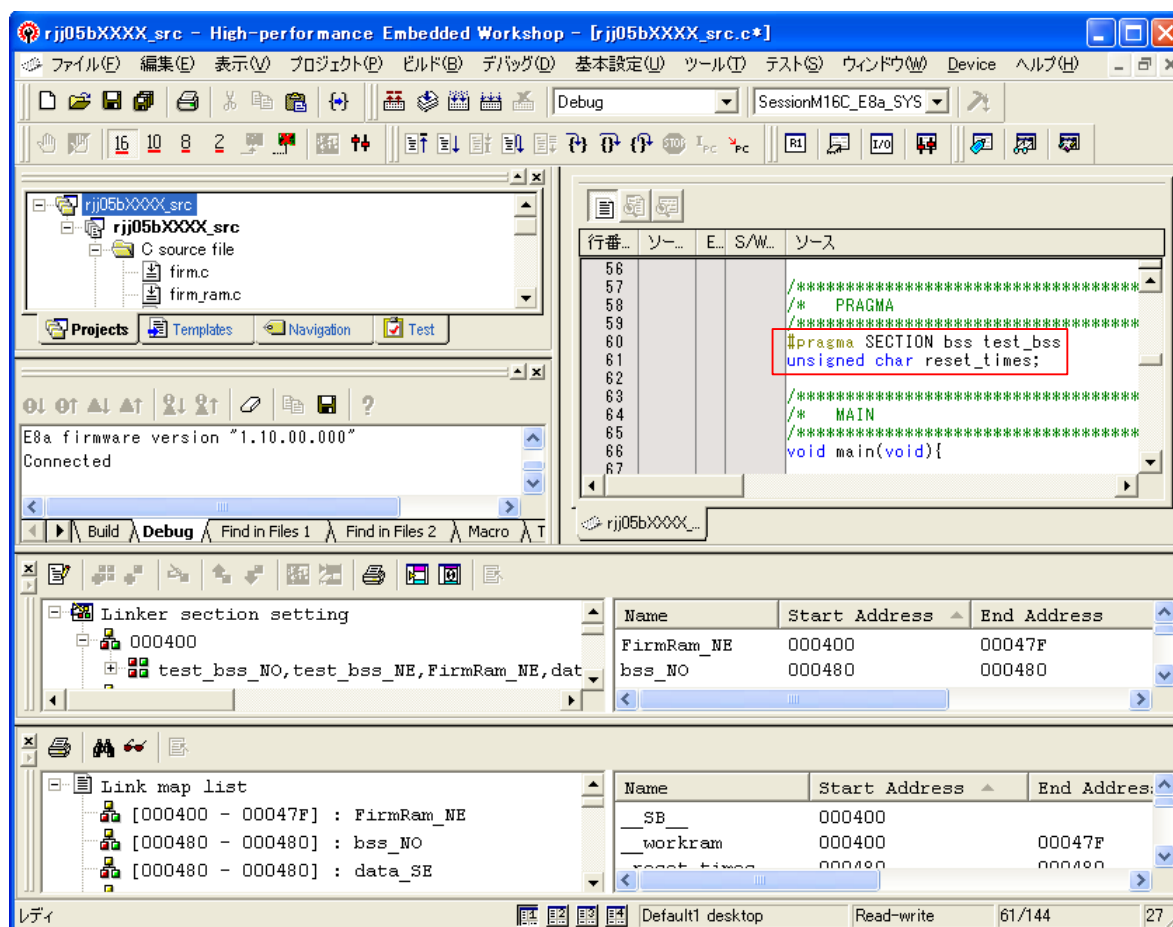


図 5.11 セクションの追加(8)

- (9) ビルドすると、test_bss_NO セクションに変数 reset_times が配置されます。

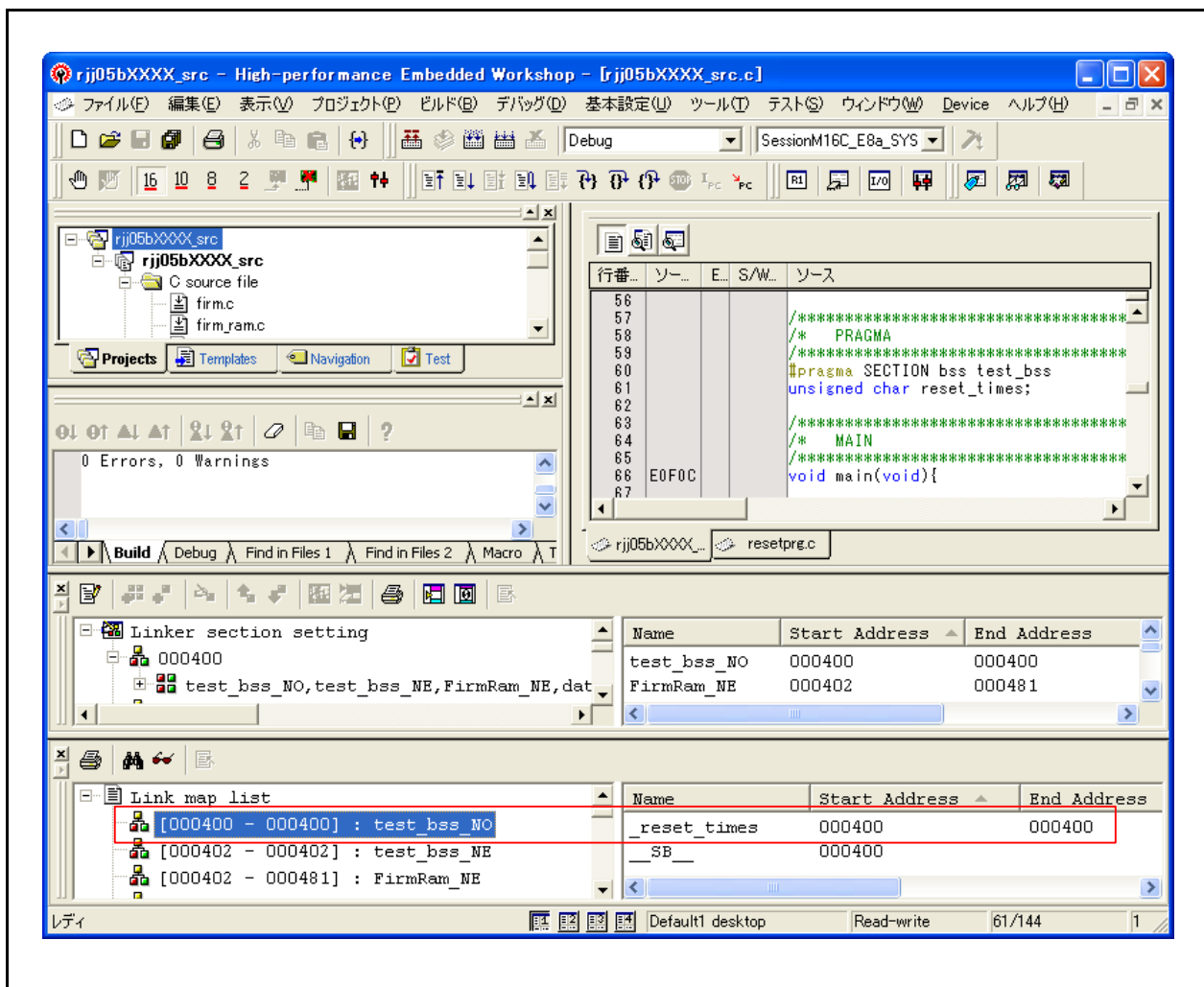


図 5.12 セクションの追加(9)

5.3 標準ライブラリの標準入出力関数の使用について

M16C シリーズで使用する標準ライブラリの標準入出力関数(注1)には、次の制限があります。

- nc30lib.lib の標準入出力関数では、UART1 を使用し、U1TB レジスタのアドレスを 3AAh としています。U1TB レジスタのアドレスが 25Ah のマイコン (M16C/64,65 等) には適していません。
- 標準入力関数 (scanf 等) 向けの低水準関数には受信バッファレジスタをリードするサイズに誤りがあります。

上記の制限を解除するために、本アプリケーションノートに付属している `device.c` ファイルを、ユーザプログラムと一緒にコンパイルしてリンクしてください。(注2)

注1. 標準入出力関数は、次の標準入力関数および標準出力関数を示します。

標準入力関数 : `fgetc`, `getc`, `getchar`, `fgets`, `gets`, `fread`, `scanf`, および `fscanf`

標準出力関数 : `fputc`, `putc`, `putchar`, `fputs`, `puts`, `fwrite`, `printf`, `fprintf`, `vfprintf`, および `vprintf`

注2. High-performance Embedded Workshop を使用している場合は、`device.c` ファイルをプロジェクトに登録してください。既に `device.c` ファイルが存在する場合は差し換えてください。

6. 参考ドキュメント

テクニカルニュース/テクニカルアップデート

(最新の情報をルネサス エレクトロニクスホームページから入手してください。)

C コンパイラマニュアル

M16C シリーズ, R8C ファミリー用 C コンパイラパッケージ V.5.45

C コンパイラユーザーズマニュアル Rev.2.00

(最新版をルネサス エレクトロニクスホームページから入手してください。)

ホームページとサポート窓口

ルネサス テクノロジホームページ

<http://japan.renesas.com/>

お問合せ先

<http://japan.renesas.com/inquiry>

csc@renesas.com

改訂記録	M16C/64A グループ High-performance Embedded Workshop スタートアッププログラムの説明
------	---

Rev.	発行日	改訂内容	
		ページ	ポイント
1.00	2010.07.30	－	初版発行

すべての商標および登録商標は、それぞれの所有者に帰属します。

製品ご使用上の注意事項

ここでは、マイコン製品全体に適用する「使用上の注意事項」について説明します。個別の使用上の注意事項については、本文を参照してください。なお、本マニュアルの本文と異なる記載がある場合は、本文の記載が優先するものとします。

1. 未使用端子の処理

【注意】未使用端子は、本文の「未使用端子の処理」に従って処理してください。

CMOS 製品の入力端子のインピーダンスは、一般に、ハイインピーダンスとなっています。未使用端子を開放状態で動作させると、誘導現象により、LSI 周辺のノイズが印加され、LSI 内部で貫通電流が流れたり、入力信号と認識されて誤動作を起こす恐れがあります。未使用端子は、本文「未使用端子の処理」で説明する指示に従い処理してください。

2. 電源投入時の処置

【注意】電源投入時は、製品の状態は不定です。

電源投入時には、LSI の内部回路の状態は不確定であり、レジスタの設定や各端子の状態は不定です。

外部リセット端子でリセットする製品の場合、電源投入からリセットが有効になるまでの期間、端子の状態は保証できません。

同様に、内蔵パワーオンリセット機能を使用してリセットする製品の場合、電源投入からリセットのかかる一定電圧に達するまでの期間、端子の状態は保証できません。

3. リザーブアドレス（予約領域）のアクセス禁止

【注意】リザーブアドレス（予約領域）のアクセスを禁止します。

アドレス領域には、将来の機能拡張用に割り付けられているリザーブアドレス（予約領域）があります。これらのアドレスをアクセスしたときの動作については、保証できませんので、アクセスしないようにしてください。

4. クロックについて

【注意】リセット時は、クロックが安定した後、リセットを解除してください。

プログラム実行中のクロック切り替え時は、切り替え先クロックが安定した後に切り替えてください。

リセット時、外部発振子（または外部発振回路）を用いたクロックで動作を開始するシステムでは、クロックが十分安定した後、リセットを解除してください。また、プログラムの途中で外部発振子（または外部発振回路）を用いたクロックに切り替える場合は、切り替え先のクロックが十分安定してから切り替えてください。

5. 製品間の相違について

【注意】型名の異なる製品に変更する場合は、製品型名ごとにシステム評価試験を実施してください。

同じグループのマイコンでも型名が違うと、内部 ROM、レイアウトパターンの相違などにより、電氣的特性の範囲で、特性値、動作マージン、ノイズ耐量、ノイズ輻射量などが異なる場合があります。型名が違う製品に変更する場合は、個々の製品ごとにシステム評価試験を実施してください。

ご注意書き

1. 本資料に記載されている内容は本資料発行時点のものであり、予告なく変更することがあります。当社製品のご購入およびご使用にあたりましては、事前に当社営業窓口で最新の情報をご確認いただきますとともに、当社ホームページなどを通じて公開される情報に常にご注意ください。
2. 本資料に記載された当社製品および技術情報の使用に関連し発生した第三者の特許権、著作権その他の知的財産権の侵害等に関し、当社は、一切その責任を負いません。当社は、本資料に基づき当社または第三者の特許権、著作権その他の知的財産権を何ら許諾するものではありません。
3. 当社製品を改造、改変、複製等しないでください。
4. 本資料に記載された回路、ソフトウェアおよびこれらに関連する情報は、半導体製品の動作例、応用例を説明するものです。お客様の機器の設計において、回路、ソフトウェアおよびこれらに関連する情報を使用する場合には、お客様の責任において行ってください。これらの使用に起因しお客様または第三者に生じた損害に関し、当社は、一切その責任を負いません。
5. 輸出に際しては、「外国為替及び外国貿易法」その他輸出関連法令を遵守し、かかる法令の定めるところにより必要な手続を行ってください。本資料に記載されている当社製品および技術を大量破壊兵器の開発等の目的、軍事利用の目的その他軍事用途の目的で使用しないでください。また、当社製品および技術を国内外の法令および規則により製造・使用・販売を禁止されている機器に使用することができません。
6. 本資料に記載されている情報は、正確を期すため慎重に作成したのですが、誤りがないことを保証するものではありません。万一、本資料に記載されている情報の誤りに起因する損害がお客様に生じた場合においても、当社は、一切その責任を負いません。
7. 当社は、当社製品の品質水準を「標準水準」、「高品質水準」および「特定水準」に分類しております。また、各品質水準は、以下に示す用途に製品が使われることを意図しておりますので、当社製品の品質水準をご確認ください。お客様は、当社の文書による事前の承諾を得ることなく、「特定水準」に分類された用途に当社製品を使用することができません。また、お客様は、当社の文書による事前の承諾を得ることなく、意図されていない用途に当社製品を使用することができません。当社の文書による事前の承諾を得ることなく、「特定水準」に分類された用途または意図されていない用途に当社製品を使用したことによりお客様または第三者に生じた損害等に関し、当社は、一切その責任を負いません。なお、当社製品のデータ・シート、データ・ブック等の資料で特に品質水準の表示がない場合は、標準水準製品であることを表します。
標準水準： コンピュータ、OA機器、通信機器、計測機器、AV機器、家電、工作機械、パーソナル機器、産業用ロボット
高品質水準： 輸送機器（自動車、電車、船舶等）、交通用信号機器、防災・防犯装置、各種安全装置、生命維持を目的として設計されていない医療機器（厚生労働省定義の管理医療機器に相当）
特定水準： 航空機器、航空宇宙機器、海中継機器、原子力制御システム、生命維持のための医療機器（生命維持装置、人体に埋め込み使用するもの、治療行為（患部切り出し等）を行うもの、その他直接人命に影響を与えるもの）（厚生労働省定義の高度管理医療機器に相当）またはシステム等
8. 本資料に記載された当社製品のご使用につき、特に、最大定格、動作電源電圧範囲、放熱特性、実装条件その他諸条件につきましては、当社保証範囲内でご使用ください。当社保証範囲を超えて当社製品をご使用された場合の故障および事故につきましては、当社は、一切その責任を負いません。
9. 当社は、当社製品の品質および信頼性の向上に努めておりますが、半導体製品はある確率で故障が発生したり、使用条件によっては誤動作したりする場合があります。また、当社製品は耐放射線設計については行っておりません。当社製品の故障または誤動作が生じた場合も、人身事故、火災事故、社会的損害などを生じさせないようお客様の責任において冗長設計、延焼対策設計、誤動作防止設計等の安全設計およびエージング処理等、機器またはシステムとしての出荷保証をお願いいたします。特に、マイコンソフトウェアは、単独での検証は困難なため、お客様が製造された最終の機器・システムとしての安全検証をお願いいたします。
10. 当社製品の環境適合性等、詳細につきましては製品個別に必ず当社営業窓口までお問合せください。ご使用に際しては、特定の物質の含有・使用を規制するRoHS指令等、適用される環境関連法令を十分調査のうえ、かかる法令に適合するようご使用ください。お客様がかかる法令を遵守しないことにより生じた損害に関し、当社は、一切その責任を負いません。
11. 本資料の全部または一部を当社の文書による事前の承諾を得ることなく転載または複製することを固くお断りいたします。
12. 本資料に関する詳細についてのお問い合わせその他お気付きの点等がございましたら当社営業窓口までご照会ください。

注1. 本資料において使用されている「当社」とは、ルネサス エレクトロニクス株式会社およびルネサス エレクトロニクス株式会社がその総株主の議決権の過半数を直接または間接に保有する会社をいいます。

注2. 本資料において使用されている「当社製品」とは、注1において定義された当社の開発、製造製品をいいます。



ルネサス エレクトロニクス株式会社

■営業お問合せ窓口

<http://www.renesas.com>

※営業お問合せ窓口の住所・電話番号は変更することがあります。最新情報につきましては、弊社ホームページをご覧ください。

ルネサス エレクトロニクス販売株式会社 〒100-0004 千代田区大手町2-6-2（日本ビル）

(03)5201-5307

■技術的なお問合せおよび資料のご請求は下記へどうぞ。
総合お問合せ窓口：<http://japan.renesas.com/inquiry>