

お客様各位

---

## カタログ等資料中の旧社名の扱いについて

---

2010年4月1日を以ってNECエレクトロニクス株式会社及び株式会社ルネサステクノロジが合併し、両社の全ての事業が当社に承継されております。従いまして、本資料中には旧社名での表記が残っておりますが、当社の資料として有効ですので、ご理解の程宜しくお願ひ申し上げます。

ルネサスエレクトロニクス ホームページ (<http://www.renesas.com>)

2010年4月1日

ルネサスエレクトロニクス株式会社

【発行】ルネサスエレクトロニクス株式会社 (<http://www.renesas.com>)

【問い合わせ先】 <http://japan.renesas.com/inquiry>

## 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、誤りが無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、海底中継機器、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と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。

# M3T-ICC740 V.1.00A

ユーザーズマニュアル

740 ファミリ用クロスツールキット<<IAR Systems 社製ツール版>>

- Microsoft、MS-DOS、Windows および Windows NT は、米国 Microsoft Corporation の米国およびその他の国における登録商標です。
- HP-UX は、米国 Hewlett-Packard Company のオペレーティングシステムの名称です。
- Sun、Java およびすべての Java 関連の商標およびロゴは、米国およびその他の国における米国 Sun Microsystems, Inc. の商標または登録商標です。
- UNIX は、X/Open Company Limited が独占的にライセンスしている米国ならびに他の国における登録商標です。
- Linux は、Linus Torvalds 氏の米国およびその他の国における登録商標あるいは商標です。
- Turbolinux の名称およびロゴは、Turbolinux, Inc. の登録商標です。
- IBM および AT は、米国 International Business Machines Corporation の登録商標です。
- HP 9000 は、米国 Hewlett-Packard Company の商品名称です。
- SPARC および SPARCstation は、米国 SPARC International, Inc. の登録商標です。
- Intel, Pentium は、米国 Intel Corporation の登録商標です。
- Adobe および Acrobat は、Adobe Systems Incorporated (アドビシステムズ社) の登録商標です。
- Netscape および Netscape Navigator は、米国およびその他の諸国の Netscape Communications Corporation 社の登録商標です。
- その他すべてのブランド名および製品名は個々の所有者の登録商標もしくは商標です。

#### 安全設計に関するお願い

- 弊社は品質、信頼性の向上に努めておりますが、半導体製品は故障が発生したり、誤動作する場合があります。弊社の半導体製品の故障又は誤動作によって結果として、人身事故火災事故、社会的損害などを生じさせないような安全性を考慮した冗長設計、延焼対策設計、誤動作防止設計などの安全設計に十分ご注意ください。

#### 本資料ご利用に際しての留意事項

- 本資料は、お客様が用途に応じた適切なルネサス テクノロジー製品をご購入いただくための参考資料であり、本資料中に記載の技術情報について株式会社ルネサス テクノロジーおよび株式会社ルネサス ソリューションズが所有する知的財産権その他の権利の実施、使用を許諾するものではありません。
- 本資料に記載の製品データ、図、表、プログラム、アルゴリズムその他応用回路例の使用に起因する損害、第三者所有の権利に対する侵害に関し、株式会社ルネサス テクノロジーおよび株式会社ルネサス ソリューションズは責任を負いません。
- 本資料に記載の製品データ、図、表、プログラム、アルゴリズムその他全ての情報は本資料発行時点のものであり、株式会社ルネサス テクノロジーおよび株式会社ルネサス ソリューションズは、予告なしに、本資料に記載した製品又は仕様を変更することがあります。ルネサス テクノロジー半導体製品のご購入に当たりましては、事前に株式会社ルネサス テクノロジー、株式会社ルネサス ソリューションズ、株式会社ルネサス販売又は特約店へ最新の情報をご確認頂きますとともに、ルネサス テクノロジーホームページ (<http://www.renesas.com>) などを通じて公開される情報に常にご注意ください。
- 本資料に記載した情報は、正確を期すため、慎重に制作したものです。万一本資料の記述誤りに起因する損害がお客様に生じた場合には、株式会社ルネサス テクノロジーおよび株式会社ルネサス ソリューションズはその責任を負いません。
- 本資料に記載の製品データ、図、表に示す技術的な内容、プログラム及びアルゴリズムを流用する場合は、技術内容、プログラム、アルゴリズム単位で評価するだけでなく、システム全体で十分に評価し、お客様の責任において適用可否を判断してください。株式会社ルネサス テクノロジーおよび株式会社ルネサス ソリューションズは、適用可否に対する責任を負いません。
- 本資料に記載された製品は、人命にかかわるような状況の下で使用される機器あるいはシステムに用いられることを目的として設計、製造されたものではありません。本資料に記載の製品を運輸、移動体用、医療用、航空宇宙用、原子力制御用、海中継用機器あるいはシステムなど、特殊用途へのご利用をご検討の際には、株式会社ルネサス テクノロジー、株式会社ルネサス ソリューションズ、株式会社ルネサス販売又は特約店へご照会ください。
- 本資料の転載、複製については、文書による株式会社ルネサス テクノロジーおよび株式会社ルネサス ソリューションズの事前の承諾が必要です。
- 本資料に関し詳細についてのお問い合わせ、その他お気付きの点がございましたら株式会社ルネサス テクノロジー、株式会社ルネサス ソリューションズ、株式会社ルネサス販売又は特約店までご照会ください。

#### 製品内容及び本書についてのお問い合わせ先

インストーラが生成する以下のテキストファイルに必要事項を記入の上、ツール技術サポート窓口 [support\\_tool@renesas.com](mailto:support_tool@renesas.com) まで送信ください。

※SUPPORT※製品名※SUPPORT.TXT

株式会社ルネサス ソリューションズ マイコンツール部	
ツール技術サポート窓口	<a href="mailto:support_tool@renesas.com">support_tool@renesas.com</a>
ユーザ登録窓口	<a href="mailto:regist_tool@renesas.com">regist_tool@renesas.com</a>
ホームページ	<a href="http://www.renesas.com/jp/tools">http://www.renesas.com/jp/tools</a>

# 目次

はじめに.....	7
<b>1. 概要 .....</b>	<b>8</b>
<b>2. インストール.....</b>	<b>10</b>
<b>3. TM の起動と終了.....</b>	<b>12</b>
3.1. TM の起動.....	12
3.2. エディタの登録.....	12
3.3. TM の終了.....	13
<b>4. クイックツアー.....</b>	<b>14</b>
4.1. 新規のプロジェクトを作成する.....	16
4.2. 新規プロジェクトの補完をする.....	17
4.3. ファイルを作成する.....	18
4.4. ファイルを登録する.....	18
4.5. プロジェクトを保存する.....	19
4.6. ビルドする.....	20
4.7. デバッガを起動する.....	21
4.8. デバッグする.....	23
4.8.1. プログラムをロードする.....	24
4.8.2. main 関数まで実行する.....	24
4.8.3. 割り込み発生の確認を行う.....	24
<b>5. プロジェクトの新規作成.....</b>	<b>25</b>
5.1. 新規作成.....	26
5.2. プロジェクトの補完.....	27
<b>6. プロジェクトの編集.....</b>	<b>29</b>
6.1. CSTARTUP.S31 の編集.....	29
6.1.1. スタックページの変更.....	29
6.1.2. 割り込みベクトル領域の変更.....	29
6.2. LNK740.XCL ファイルの編集.....	30
6.2.1. スタック領域の変更.....	30
6.2.2. ゼロページ先頭アドレスの変更.....	31
6.2.3. N ページ終了アドレスの変更.....	31

6.2.4.	ROM 領域アドレスの変更.....	32
6.2.5.	割り込みベクトル領域の変更.....	32
6.2.6.	ライブラリの変更.....	33
6.3.	ICC740 オプションの変更.....	34
6.3.1.	プロセッサ・グループの変更.....	35
6.3.2.	メモリモデルの変更.....	36
6.3.3.	最適化レベルの変更.....	36
6.4.	A740 オプションの変更.....	37
6.4.1.	プロセッサ・グループの変更.....	37
6.4.2.	メモリモデルの変更.....	38
6.5.	XLINK オプションの変更.....	39
<b>7.</b>	<b>プロジェクトの開発.....</b>	<b>40</b>
7.1.	ソースファイルの作成および登録.....	40
7.2.	LNK740.XCL ファイルの変更.....	41
<b>8.</b>	<b>プロジェクトのビルド.....</b>	<b>42</b>
8.1.	C コンパイラ ICC740 およびアセンブラ A740 のエラー.....	42
8.2.	リンカ XLINK のエラー.....	43
8.3.	ソースファイル単位のオプションの設定.....	45
<b>9.</b>	<b>プロジェクトのデバッグ.....</b>	<b>46</b>
<b>10.</b>	<b>ヘキサファイルの作成.....</b>	<b>48</b>

## 目次

図 1 M3T-ICC740 のインストール手順	11
図 2 プロジェクトバー	12
図 3 プロジェクトバーをデスクトップの上端に貼り付けた状態	12
図 4 [TOOLS INFORMATION]ダイアログ	12
図 5 プロジェクトバーの右クリックメニュー	13
図 6 プロジェクトバーの[NEW PROJECT]ボタン	16
図 7 プロジェクトエディタ	16
<b>図 8 新規プロジェクトの補完</b>	17
<b>図 9 ソースファイルの登録</b>	19
図 10 プロジェクトエディタの[保存]ボタン	19
図 11 プロジェクトバーの[部分ビルド]、[ビルド]、[リビルド]ボタン	20
<b>図 12 ビルダー</b>	20
図 13 プロジェクトバーの[デバッガを起動する]ボタン	21
図 14 [TOOLS INFORMATION]ダイアログの[DEBUG TOOL]タブ	21
図 15 デバッガの登録	22
図 16 M3T-PD38SIM	23
<b>図 17 M3T-PD38SIM の[INIT]ダイアログ</b>	23
図 18 プロジェクトバー	26
図 19 プロジェクトエディタ	26
図 20 LNK740.XCL ファイルの登録	28
<b>図 21 [オプションブラウザ]の起動</b>	34
図 22 [オプションブラウザ]ダイアログ	34
図 23 ICC740 の[オプション]ダイアログ	35
図 24 ターゲットカテゴリと[選択オプション]ダイアログ	35
図 25 [選択オプション]ダイアログの最適化レベル変更	36
図 26 A740 の[オプション]ダイアログ	37
図 27 プロジェクトバーの[エディタを開く]ボタン	40
<b>図 28 ソースファイルの登録</b>	40
図 29 プロジェクトバーのビルド関連ボタン	42
図 30 [オプションブラウザ]ダイアログでのソースファイル単位オプション設定方法	45
図 31 プロジェクトバーの[デバッガの起動]ボタン	46
図 32 [TOOL INFORMATION]ダイアログの[DEBUG TOOL]タブ	46
図 33 [プロジェクトプロパティ]ダイアログ	48

図 34 [オプションブラウザ]ダイアログの XLINK(LMC)登録.....	48
--	----

## 表目次

表 1 新規作成ウィザードの実行ステップ .....	16
表 2 プロセッサ・グループ、メモリモデル、スタック領域.....	25
表 3 変更箇所.....	25
表 4 新規作成ウィザードの実行ステップ .....	27
表 5 ワーキングディレクトリの新規作成ファイル.....	27
表 6 CSTARTUP.S31 の編集項目 .....	29
表 7 LNK74.XCL の編集項目 .....	30
表 8 プロセッサグループ、メモリモデルとライブラリ .....	33
表 9 ICC740 のオプション .....	34
表 10 A740 のデフォルトオプション .....	37
表 11 XLINK のデフォルトのオプション (コマンドライン) .....	39
表 12 XLINK(LMC)のデフォルトのオプション (コマンドライン) .....	49



## はじめに

このユーザズマニュアルをお読みになる前に、製品に添付されているリリースノートを必ずお読みください。製品構成、製品の扱い、注意事項などの重要な内容が記述されています。

### 対象読者

このユーザズマニュアルでは、次の方を対象としています。

- ・ C 言語による組み込み用プログラム開発およびデバッグ経験のある方
- ・ ルネサス統合化開発環境を初めてご使用になる方

### 参考マニュアル

ルネサス統合化開発環境(TM)に関する詳しい用語説明、機能説明、操作については、次のマニュアルを参照してください。このマニュアルはオンラインでも参照することができます。

- ・ ルネサス開発サポートツール用統合化開発環境  
TM V.3.20A ユーザズマニュアル (TM V.3.20A に付属)

IAR Systems 社製 C コンパイラ ICC740 (以下 ICC740 と略します) に関する詳しい用語説明、機能説明、操作については、ICC740 に付属の PDF マニュアルを参照してください。

## 1. 概要

M3T-ICC740 は、ルネサス 8 ビット 740 ファミリ用のクロスツールキットです。製品パッケージには IAR Systems 社製 C コンパイラ ICC740 (以下 ICC740 と略します) と統合化開発環境 (TM) が含まれており、TM を用いた C 言語およびアセンブリ言語によるプログラム開発を支援します。

M3T-ICC740 には、次のソフトウェアが含まれています。

- ICC740

IAR Systems 社製の 740 ファミリ用 C コンパイラです。

ICC740 の使用権許諾は IAR Systems 社の使用権許諾契約に従います。使用権許諾契約書はインストール時に表示されます。

本製品はコマンドライン入力方式となっており、IAR Systems 社製組み込みワークベンチ (Embedded Workbench) は含まれていません。

- TM

コンパイラ / アセンブラ / デバッガ / エディタなどのツール群を共通グラフィカルユーザーインターフェース (GUI) に統合して、ソフトウェアの開発効率を改善するためのツールです。

- TW74

ICC740 を TM で使用できるようにするためのデータ設定を行うツールです。インストーラ形式で設定を行い、この処理で完了します。

本ツールが実行されていないと、TM 上で ICC740 のプロジェクト開発ができません。

また、以下のソフトウェアを同梱しています。

- ・ M3T-PD38SIM (体験版)

740 ファミリ用シミュレータデバッガの体験版を同梱しています。

- ・ SC74

ルネサス製 740 ファミリアセンブラ M3T-SRA74 用に作成したソースファイルを ICC740 に付属のアセンブラ A740 のソースファイル形式に変換するソースファイルコンバータです。技術サポート対象外ソフトウェアです。SC74 の取り扱いは付属のファイル license.sj を参照してください。

本マニュアルでは、各ツールのインストールと、TM 上での ICC740 プロジェクトの開発方法について説明します。

## 2. インストール

M3T-ICC740 では、以下のツールをインストールします。

- ・ ICC740
- ・ TM
- ・ TW74

インストール手順を示します (図 1)。

- ・ ICC740

icc740 フォルダ内の autorun.exe を実行してください。表示されるダイアログの「View the Japanese “Read me first”」にインストール方法と製品の登録情報が紹介されていますので参考にしてください。

インストーラーの指示に従ってインストールしてください。ICC740 用のライセンス ID が必要となります。

QuickStart キーでの使用期間は 1 ヶ月です。インストール後 1 ヶ月以内に「永久キー」を IAR Systems 社から入手してください。「永久キー」の入手方法および設定方法は「View the Japanese “Read me first”」を参照してください。

- ・ TM

TM は V3.20 以上を使用します。

V3.20 以上がインストールされていない場合はインストールしてください。

インストーラーの指示に従ってインストールしてください。

- ・ TW74

TW74 は ICC740 および TM をインストールした後にインストールしてください。

インストーラーの指示に従ってインストールしてください。

本ツールで M3T-ICC740 のライセンス認証を行います。

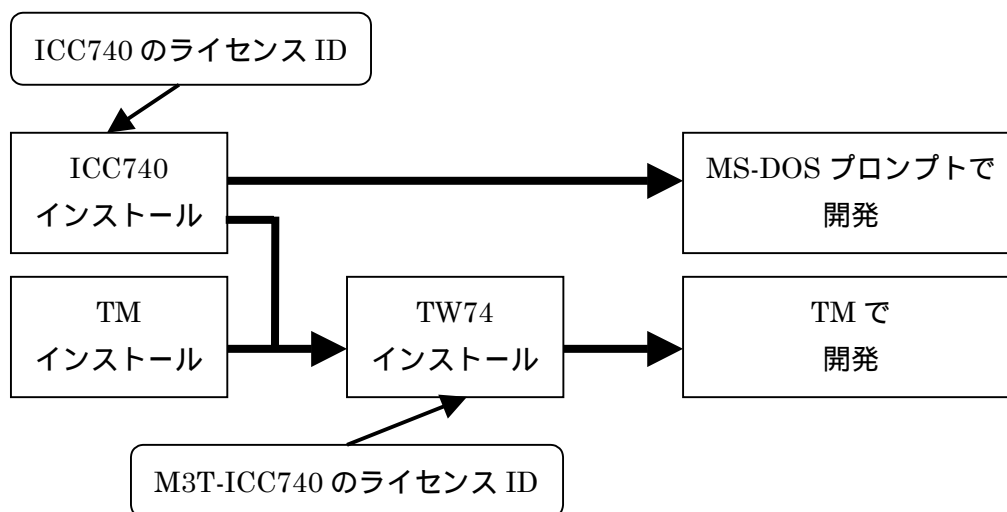


図 1 M3T-ICC740 のインストール手順

同梱ソフトウェアのインストール方法を示します。

- ・ M3T-PD38SIM (体験版)

本書のクイックツアーを実行される場合は、インストールしてください。

インストーラーの指示に従ってインストールしてください。インストール後 4 ヶ月使用できます。

- ・ SC74

SC74 にはインストーラ はありません。そのまま使用できます。

### 3. TMの起動と終了

#### 3.1. TMの起動

正常にインストールすると、スタートメニューから

[プログラム] [RENESAS-TOOLS] [TM V.3.xx]

の中から TM を起動できるようになります。起動すると、プロジェクトバー（図 2）が現れます。



図 2 プロジェクトバー

初期起動時、プロジェクトバーは「フローティング」な状態です。このウィンドウをつかみ画面の上に持っていけば、デスクトップの上端に貼り付けることができます（図 3）。



図 3 プロジェクトバーをデスクトップの上端に貼り付けた状態

#### 3.2. エディタの登録

TM では、特にエディタを内部に持っていません。お好みの市販エディタを自由に登録できます。エディタの登録は[ツールの登録]ボタンをクリックしてください。[Tools Information]ダイアログ（図 4）が表示されますので、[EDIT TOOL]タブを選択してお好みのエディタを登録してください。

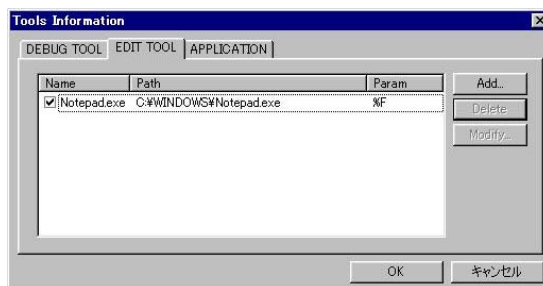


図 4 [Tools Information]ダイアログ

### 3.3. TMの終了

TMの終了は、プロジェクトバーの[閉じる]ボタン（「フローティング」な状態時）または右クリックで表示されるメニュー（図5）の[終了]項目の選択で行います。

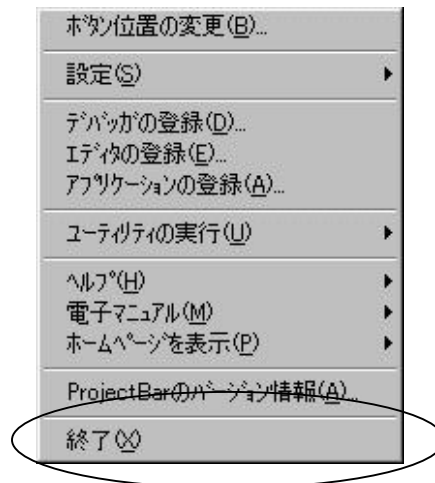


図5 プロジェクトバーの右クリックメニュー

## 4. クイックツアー

クイックツアーにより、TM を実際に使ってみましょう。

TM では、プロジェクト名とワーキングディレクトリを指定してプロジェクトを開発します。クイックツアーでは以下の設定としますが、必要に応じて変更してください。

ワーキングディレクトリ：c:\work\proj\_1

プロジェクト名：proj\_1

ソースファイル：tutor3.c

TM ではプロジェクト毎にディレクトリを自動作成しません。ワーキングディレクトリにプロジェクトを作成します。ワーキングディレクトリはユーザが必要に応じて作成してください。

ソースファイル tutor3.c は、IAR Systems 社製のサンプルプログラムを M3T-PD38SIM 用に変更したものです。このプログラムの処理およびソースは以下のとおりです。

### < 処理 >

プログラムは main 関数の while 文の中で無限ループとなります。

変数 my\_char は関数により 'a' ~ 'z' の値をランダムに持ちます。

変数 my\_char が 'i' になると、関数により brk 割り込みが発生します。

関数 brk\_interrupt は brk 割り込みにより実行される関数で変数 my\_char を '.' に設定します。この関数の "interrupt [0x00]" 記述は 3803 グループの brk 割り込みベクトルに関数 brk\_interrupt のアドレスを設定します。

その後、main 関数の無限ループの処理が繰り返されます。



## ソースプログラム tutor3.c

```
/*-----
 * File: tutor3.c
 *
 * Purpose: Handling real time interrupts
 *
 * Usage: ICC -r -L -q tutor3.c
 *        XLINK -r -f <link file> tutor3
 *
 * Description: Using BRK vector to print a character
 *
 * Copyright 1997 IAR Systems
 *
 * $Id: tutor3.c 1.3 1998/01/15 09:24:55Z Laban Exp $
 *-----*/

#pragma language=extended /* enable use of extended keywords */

#include <stdlib.h>
#include <stdio.h>
#include "intr740.h" /* include intrinsics */

/*****
 * Variables
 *****/

char my_char = '!';
int call_count = 0;

/*****
 * Start of code
 *****/

void interrupt [0x00] brk_interrupt(void)
{
    // putchar('I');
    my_char='!';
}

void execute_brk(void)
{
    break_instruction(); /* Use intrinsic function */
}

void do_foreground_process(void)
{
    call_count++;
    // putchar(my_char);
    my_char = rand() % 26 + 'a';
}

void main(void)
{
    while (1)
    {
        do_foreground_process();
        if (my_char=='i') execute_brk();
    }
}
```

#### 4.1. 新規のプロジェクトを作成する

プロジェクトバーの[New Project]ボタンをクリックします ( 図 6 )。

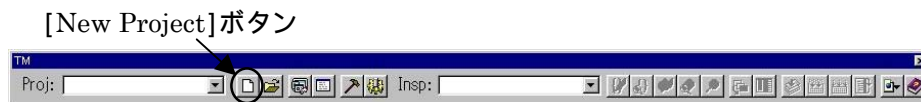


図 6 プロジェクトバーの[New Project]ボタン

プロジェクトエディタの新規作成ウィザードが起動されますので、表 1の選択をしてください。

表 1 新規作成ウィザードの実行ステップ

ステップ	選択
ステップ 1	ターゲットチップは[740 Family(ICC740)]を選択してください。
ステップ 2	C 言語プロジェクトを選択します。
ステップ-コンパイラ	コンパイラパッケージは[TW74]を選択します。この選択により、コンパイル時にはICC740 が使用されます。 スタートアッププログラムは[標準のスタートアッププログラムを使用する]選択します。
ステップ-完了	[完了]ボタンをクリックしてください。

プロジェクトエディタ ( 図 7 ) に新規プロジェクトが表示されます。

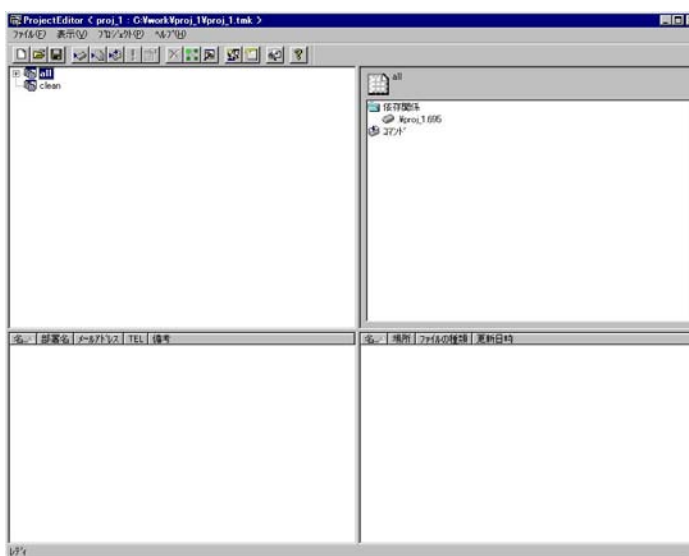


図 7 プロジェクトエディタ

## 4.2. 新規プロジェクトの補完をする

TM で ICC740 のプロジェクトを開発する場合、新規プロジェクトの補完を行う必要があります。

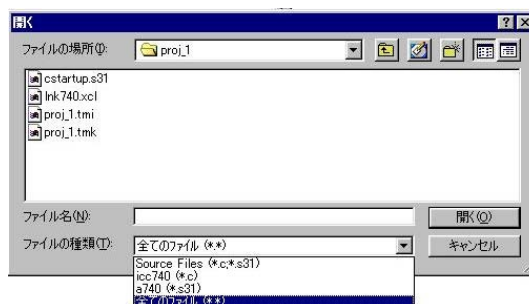
以下の手順で補完を行ってください。

1. プロジェクトエディタのプロジェクトツリーで all の「+」をクリックして、`.$proj_1.695` を表示します。
2. all に続く `.$proj_1.695` の「+」をクリックして、`.$proj_1.cmd` を表示します。
3. `.$proj_1.cmd` をクリックして、[ファイルの追加]ボタンをクリックします(図 8 (a))。
4. [開く]ダイアログがオープンしますので、[ファイルの種類]で[全てのファイル]を選択して、ワーキングディレクトリの `lnk740.xcl` ファイルを追加してください(図 8 (b))。
5. `lnk740.xcl` ファイルを追加すると図 8 (c)の画面となり、`lnk740.xcl` がプロジェクトに登録されます。

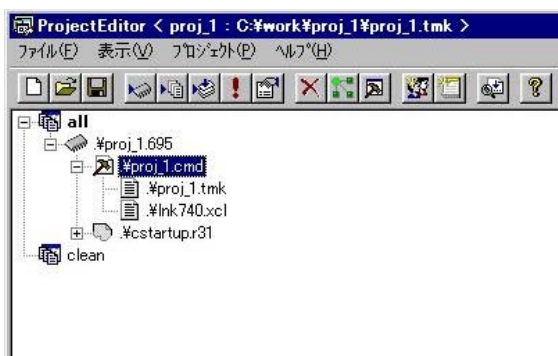
[ファイルの追加]ボタン



(a) 登録前



(b) [開く] ダイアログ



(c) 登録後

図 8 新規プロジェクトの補完




#### 4.3. ファイルを作成する

ソースプログラムをエディタで作成します。なお、tutor3.c は SmpTw74 フォルダにありますので、ワーキングディレクトリにコピーしてください。SmpTw74 フォルダは TW74 をインストールしたフォルダ（通常は¥MTOOL）に作成されます。

#### 4.4. ファイルを登録する

作成したソースファイルはプロジェクトに登録する必要があります。

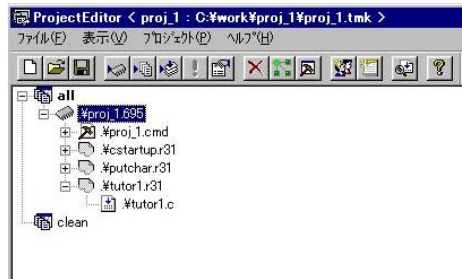
ワーキングディレクトリに置いただけでは登録されませんので、以下の手順で登録してください。

1. プロジェクトエディタで、¥proj\_1.695 ファイルをクリックして、[ファイルの追加] ボタンをクリックします（ 9 (a)）。
2. [開く]ダイアログがオープンしますので、[ファイルの種類]で[Source Files]を選択して、ソースファイルを追加してください。
3. ¥proj\_1.695 の下に、拡張子が.r31 となったソースファイルが登録されます（ 9 (b)）。
4. 拡張子が.r31 となったソースファイルの「+」をクリックすると、ソースファイルが表示されます（ 9 (c)）。



(a) ソースファイル追加前

(b) ソースファイル追加後



(c) ソースファイルの表示

図 9 ソースファイルの登録

#### 4.5. プロジェクトを保存する

作成したプロジェクトは保存する必要があります。プロジェクトエディタの[保存]ボタンをクリックしてください(図 10)。

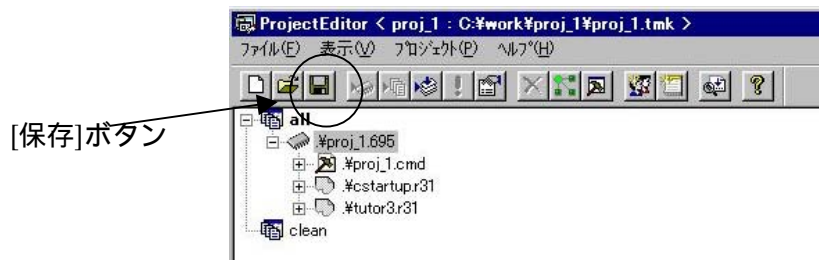


図 10 プロジェクトエディタの[保存]ボタン

#### 4.6. ビルドする

ビルドは、プロジェクトバーの[ビルド]、[リビルド]、または[部分ビルド]ボタンにより行います。ここでは、[リビルド]ボタンをクリックしてください(図 11)。

ビルダーを起動して、リビルド処理を行います(図 12)。

エラーが発生せずに完了した場合は、ワーキングディレクトリに *proj\_1.695* ファイルを生成します。

[部分ビルド]、[ビルド]、[リビルド]ボタン



図 11 プロジェクトバーの[部分ビルド]、[ビルド]、[リビルド]ボタン

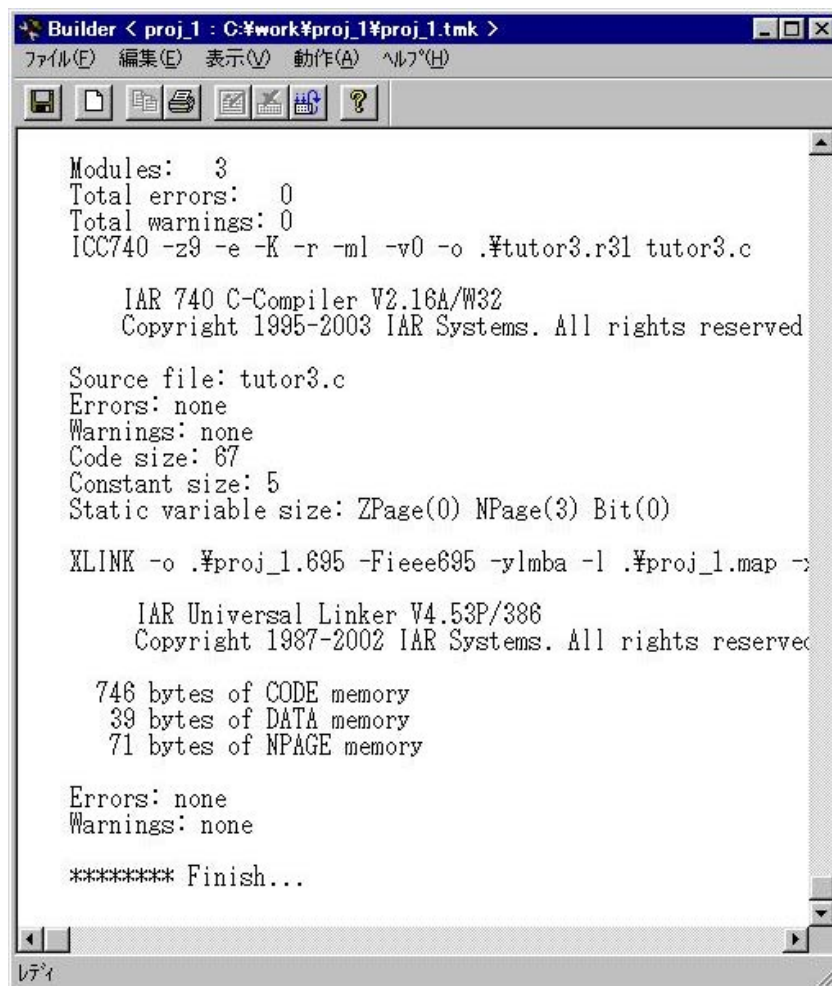


図 12 ビルダー

#### 4.7. デバッガを起動する

ビルドが完了したプロジェクトでは、ワーキングディレクトリにアブソリュートファイル *proj\_1.695* が生成されます。

この、*proj\_1.695* は M3T-PD38 または M3T-PD38SIM でデバッグすることができます。プロジェクトバーの[デバッガを起動する]ボタンをクリックしてください(図 13)。

[デバッガを起動する]ボタン



図 13 プロジェクトバーの[デバッガを起動する]ボタン

初期状態では、デバッガが登録されていないので、[Tools Information]ダイアログが表示されます(図 14)。

M3T-PD38SIM または M3T-PD38 が、TW74 インストール前にインストールされている場合は、各々表示されていますので、使用するデバッガのチェックボックスにチェックを入れてください。

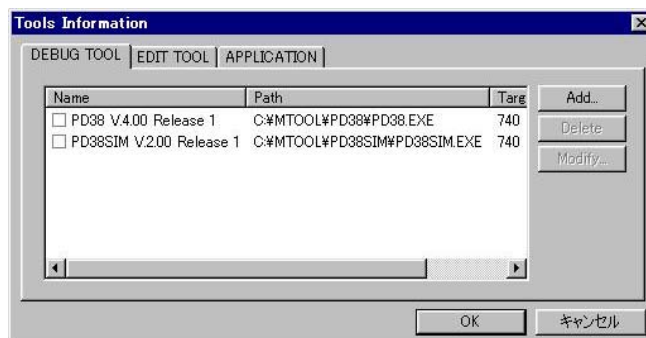


図 14 [Tools Information]ダイアログの[DEBUG TOOL]タブ

M3T-PD38SIM および M3T-PD38 が表示されない場合は、[Add]ボタンによりデバッガを登録してください。登録後、使用するデバッガのチェックボックスにチェックを入れてください(図 15)。

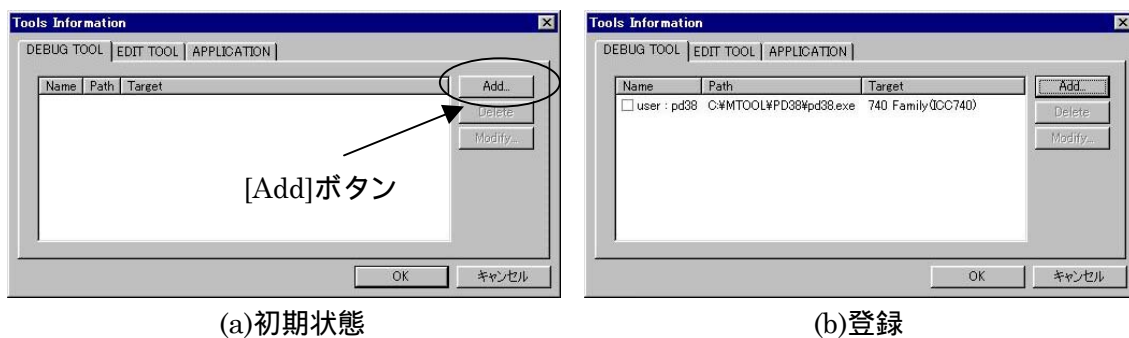


図 15 デバッガの登録



#### 4.8. デバッグする

M3T-PD38SIM ( 図 16 ) でデバッグを行います。

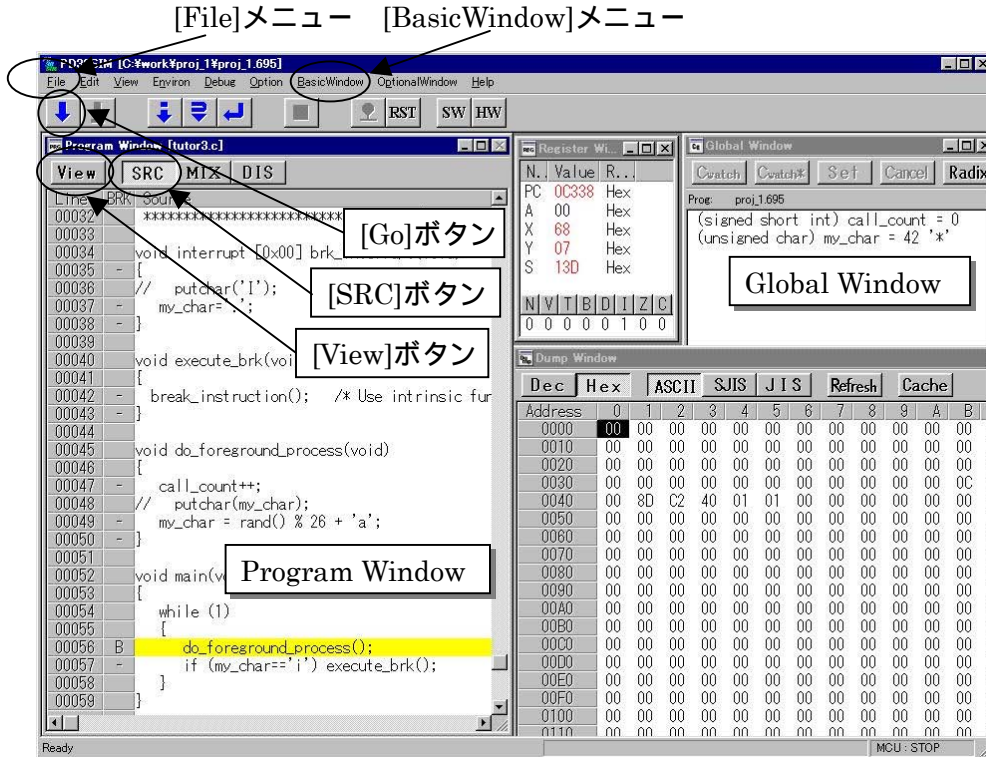


図 16 M3T-PD38SIM

M3T-PD38SIM を初めて起動した場合は、ターゲット MCU に対応する SFR ファイルを設定する必要があります。[Init]ダイアログ ( 図 17 ) の一番上の[Refer]ボタンにより設定してください。M3T-PD38SIM をインストールしたフォルダの M38000.sfr ファイルを選択してください。

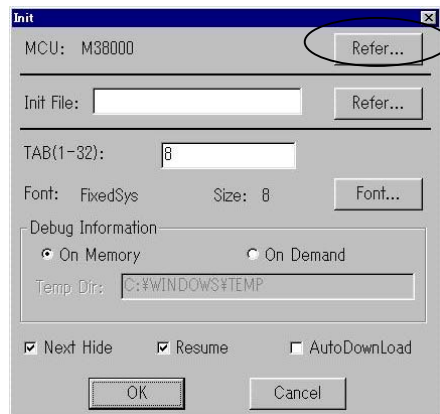


図 17 M3T-PD38SIM の[Init]ダイアログ

#### 4.8.1. プログラムをロードする

プログラムのロードは[File]メニュー [Download] [Load module]で行います。  
[download]ダイアログが表示されますので、*proj\_1.695* ファイルを選択します。  
プログラムがロードされ、プログラムカウンタの位置のプログラムが表示されます。  
プログラムカウンタの位置は黄色のラインで表示されます。

#### 4.8.2. main 関数まで実行する

まず、main 関数まで実行します。

main 関数を表示するには、[View]ボタン [Source]ボタン *tutor3.c* main [OK]ボタン、[SRC]ボタンの選択により行います。

Program Window の BRK 欄には、ブレークポイントを設定できる位置に'-'が表示されます。Line 欄の 00056 行の'-'をダブルクリックしてください。'-'が'B'に変わります。これはブレークポイントが設定されたことを示します。

この状態で[Go]ボタンをクリックしてください。この位置までプログラムを実行し、00056 行は黄色のラインに変わります。

#### 4.8.3. 割り込み発生の確認を行う

割り込み発生の確認を行います。

*brk\_interrupt* 関数は、どの関数からも呼び出されません。BRK 命令を実行することで呼び出されます。

BRK 命令の実行は *execute\_brk* 関数内の *break\_instruction* 関数が行います。この *break\_instruction* 関数は BRK 命令に置き換えられるインライン関数です。

*execute\_brk* 関数は変数 *my\_char* が'i'に設定されると呼び出されます。

変数 *my\_char* の値をウォッチするには、Global Window で行います。[BasicWindow]メニュー [C Watch Window] [Global Window]を選択してください。

Line 欄の 00037 行にブレークポイントを設定して、[Go]ボタンを何度かクリックしてください。

変数 *my\_char* が'i'に設定された時に、00037 行が黄色のラインとなっていることで割り込み発生の確認を行います。

M3T-PD38SIM の[File]メニュー [Exit]で M3T-PD38SIM を終了します。

## 5. プロジェクトの新規作成

ICC740 では、プロセッサ・グループ、メモリモデル、およびスタック領域を設定して、プログラムの開発を行う必要があります。M3T-ICC740 は、表 2 の設定で新規プロジェクトを作成しています。

なお、本マニュアルではプロジェクト名を *proj\_1* としています。

**表 2 プロセッサ・グループ、メモリモデル、スタック領域**

選択項目	選択肢	M3T-ICC740 の新規時設定
プロセッサ ・グループ	<ul style="list-style-type: none"> <li>・ MUL/DIV を持つマイコン ( 拡張メモリアクセスは持たない)</li> <li>・ MUL/DIV を持たないマイコン</li> <li>・ 拡張メモリアクセスを持つマイコン ( MUL/DIV を持つ)</li> </ul>	<ul style="list-style-type: none"> <li>・ MUL/DIV を持つマイコン ( 拡張メモリアクセスは持たない)</li> </ul>
メモリモデル	<ul style="list-style-type: none"> <li>・ Tiny モデル</li> <li>・ Large モデル</li> </ul>	<ul style="list-style-type: none"> <li>・ Large モデル</li> </ul>
スタック領域	<ul style="list-style-type: none"> <li>・ ゼロページ ( 000H ~ 0FFH )</li> <li>・ 1 ページ ( 100H ~ 1FFH )</li> </ul>	<ul style="list-style-type: none"> <li>・ 1 ページ ( 100H ~ 1FFH )</li> </ul>

M3T-ICC740 の新規時設定と選択肢が異なる場合、表 3 に示したファイル、オプション等を変更する必要があります。

**表 3 変更箇所**

選択項目	変更箇所 ( 章番号 )
プロセッサ・グループ	ライブラリ ( 6.2.6 ) ICC740 のオプション ( 6.3.1 ) A740 のオプション ( 6.4.1 )
メモリモデル	ライブラリ ( 6.2.6 ) ICC740 のオプション ( 6.3.2 ) A740 のオプション ( 6.4.2 )
スタック領域	cstartup.s31 ファイル ( 6.1.1 ) lnk740.xcl ファイル ( 6.2.1 )

cstartup.s31 および lnk740.xcl ファイルは、M38037M8 ( 3803 グループ、RAM 1KByte, ROM 32KByte ) 用に編集されていますので、ターゲットマイコンに合わせて、編集してください。なお、ワーキングディレクトリに生成される \*.tmk, \*.tmi, \*.cmd ファイルは TM が管理するファイルですので、書き換えをしないでください。

## 5.1. 新規作成

プロジェクトの新規作成は TM を起動することから始まります。

スタートメニューから TM を起動してください。

TM を起動すると図 18のプロジェクトバーが現れます。



図 18 プロジェクトバー

TM ではプロジェクト毎にプロジェクト用ディレクトリを自動作成しません。ワーキングディレクトリにプロジェクトを作成します。ワーキングディレクトリはユーザが必要に応じて作成してください。

プロジェクトの新規作成には次の 3 つの方法があります。

1. プロジェクトバーの[New Project]ボタン
2. プロジェクトエディタ (図 19) の[新規プロジェクト]ボタン
3. プロジェクトエディタの[ファイル]メニューの[プロジェクト新規作成]項目

プロジェクトエディタはプロジェクトバーの[Project Editor を開く]ボタン (図 18) で起動します。何れの場合も、プロジェクトエディタの新規作成ウィザードが起動されますので、ウィザードの実行ステップに従って作成してください (表 4参照)。

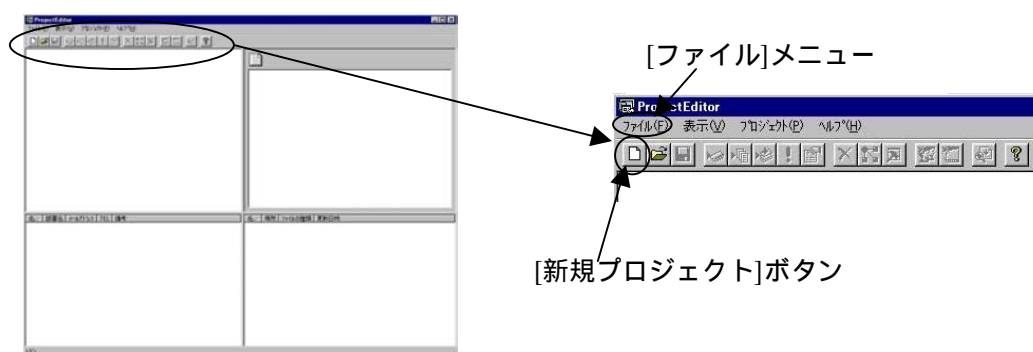


図 19 プロジェクトエディタ

表 4 新規作成ウィザードの実行ステップ

ステップ	選択
ステップ 1	ターゲットチップは[740 Family(ICC740)]を選択してください。M3T-SRA74 がインストールされている場合、ターゲットチップに [740 Family] も表示されますが、 [740 Family(ICC740)]を選択してください。
ステップ 2	C 言語プロジェクトを選択します。
ステップ-コンパイラ	コンパイラパッケージは[TW74]を選択します。この選択により、コンパイル時には ICC740 が使用されます。スタートアッププログラムは[標準のスタートアッププログラムを使用する]を選択します。
ステップ-完了	[完了]ボタンをクリックしてください。

この時点で、ワーキングディレクトリには、表 5のファイルが生成されます。

表 5 ワーキングディレクトリの新規作成ファイル

<i>proj_1.tmk</i>	プロジェクトファイル (メイクファイル形式)
<i>proj_1.tmi</i>	プロジェクトファイル (開発メンバー、ドキュメント情報)
<i>startup.s31</i>	スタートアッププログラム
<i>lnk740.xcl</i>	XLINK 用リンク・コマンド・ファイル

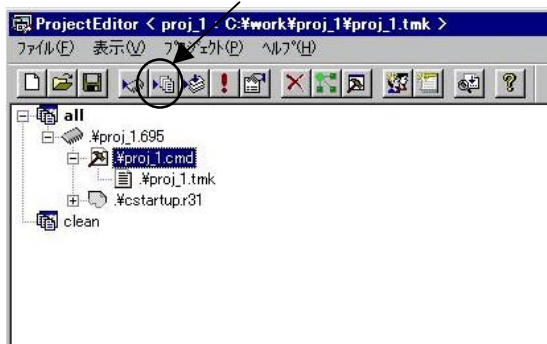
## 5.2. プロジェクトの補完

ICC740 のプロジェクトでは、新規プロジェクトを補完する必要があります。

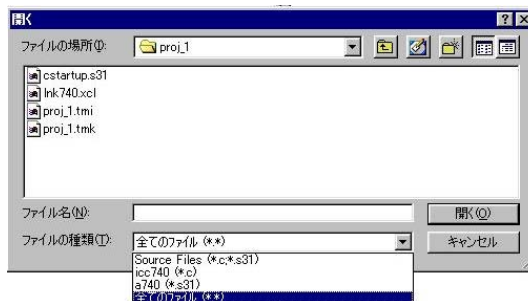
新規プロジェクト作成時にワーキングディレクトリにコピーされる *lnk740.xcl* はリンク XLINK のコマンドファイルです。この中にはセグメントの配置方法等が記述されています。このファイルはプロジェクトには登録されていません。以下の手順で登録してください。

1. プロジェクトエディタのプロジェクトツリーで *all* の「+」をクリックして、*proj\_1.695* を表示します。
2. *all* に続く *proj\_1.695* の「+」をクリックして、*proj\_1.cmd* を表示します。
3. *proj\_1.cmd* をクリックして、[ファイルの追加]ボタンをクリックします (図 20 (a))。
4. [開く]ダイアログがオープンしますので、[ファイルの種類]で[全てのファイル]を選択して、ワーキングディレクトリの *lnk740.xcl* ファイルを追加してください(図 20 (b))。
5. *lnk740.xcl* ファイルを追加すると図 20 (c)の画面となり、*lnk740.xcl* がプロジェクトに登録されます。

[ファイルの追加]ボタン



(a) 登録前



(b) [開く]ダイアログ



(c) 登録後

図 20 lnk740.xcl ファイルの登録

.¥proj\_1.cmd ファイルは、プロジェクトの新規作成時にはありません。ビルドまたはリビルド時に作成されます。TM が管理するファイルですので、書き換えをしないでください。

## 6. プロジェクトの編集

新規プロジェクト作成後、必要に応じて cstartup.s31、lnk740.xcl の編集およびオプションの変更をする必要があります。

### 6.1. cstartup.s31 の編集

cstartup.s31 では、表 6の項目を必要に応じて編集します。

表 6 cstartup.s31 の編集項目

編集項目	要因	該当行
スタックページ	スタック領域	137 行
割り込みベクトル領域	ターゲットマイコン	375 行

#### 6.1.1. スタックページの変更

137 行で、スタックページの設定を行っています。ここでは 3803 グループの CPU モードレジスタに対してスタックページを 1 ページに設定しています。

```
LDM #0CH, 3BH ; set stack page : 3803 Group
```

ターゲットマイコンに合わせてスタックページを設定してください。

スタックページをゼロページに設定した場合は、lnk740.xcl ファイルの CSTACK セグメントの設定を変更する必要があります。

#### 6.1.2. 割り込みベクトル領域の変更

375 行で、割り込みベクトル領域のサイズ設定を行っています。ここでは 3803 グループの割り込みベクトル領域のサイズを設定しています。

```
BLKB OFFFEH - 0FFDCH - 2 ; 3803 Group
```

この BLKB で設定する領域ではリセットベクトル分を引いてください(上記の-2 が該当します)。

なお、割り込みベクトル領域の先頭アドレスの設定は lnk740.xcl ファイルで行います。

## 6.2. lnk740.xcl ファイルの編集

ターゲットマイコン、メモリモデル等の設定に応じて、lnk740.xcl ファイルの編集が必要となります。

表 7 Lnk74.xcl の編集項目

編集項目	要因	lnk740.xcl の該当行
スタック領域	メモリモデル、ターゲットマイコン	60
ゼロページ先頭アドレス	ターゲットマイコン	38
N ページ終了アドレス	ターゲットマイコン	65
ROM 領域アドレス	ターゲットマイコン	77
割り込みベクトル	ターゲットマイコン	91
ライブラリ	プロセッサ・グループ	110

### 6.2.1. スタック領域の変更

740 ファミリでは、スタック領域はゼロページ(00h~FFh)または1ページ(100h~1FFh)を選択して使用します。

ICC740 では、このスタック領域を CSTACK セグメントとして使用します。

M3T-ICC740 の標準では、1ページの100h~13Fhまでを使用する設定となっています。

```
-Z(NAPGE)CSTACK+40=100
```

変更例を示します。

例1) スタックの使用領域を1ページの120h~14Fhまでを使用する場合は、以下のように変更してください。

```
-Z(NAPGE)CSTACK+30=120
```

例2) スタック領域をゼロページに変更する場合は、以下のように変更してください。

```
-Z(ZPAGE)CSTACK+40
```

ゼロページには、SFR 領域およびゼロページに配置しなくてはならないセグメントがあります。上記の記述は、54 行目に記述している INT\_EXPR\_STACK セグメント以降の40hバイトを使用する設定となります。



なお、スタック領域をゼロページに変更する場合は、cstartup.s31 ファイルの変更も必要となります。

### 6.2.2. ゼロページ先頭アドレスの変更

ICC740 では、RAM 領域をゼロページと N ページ (100h 以降) に分けて設定します。ゼロページの設定には”-Z(ZPAGE)”を指定します。

```
-Z(ZPAGE)ZPAGE,C_ARGZ,Z_UDATA,Z_IDATA=41-FF
```

上記の設定は、SFR 領域 (0h ~ 40h) 後の 41h 番地から FFh 番地までの RAM 空間に、ZPAGE、C\_ARGZ、Z\_UDATA、および Z\_IDATA セグメントを配置する設定です。これらのセグメントは ICC740 が使用するセグメントですので、削除しないでください。変更例を示します。

例) 7220 シリーズ等 SFR 領域が 0h 番地から設定されていないマイコンでは以下のように変更してください。

```
-Z(ZPAGE)ZPAGE,C_ARGZ,Z_UDATA,Z_IDATA=0-BF
```

7200 シリーズでは、SFR 領域が C0h 番地から開始されますので、BFh 番地までとします。

### 6.2.3. N ページ終了アドレスの変更

N ページの設定には”-Z(NPAGE)”を指定します。

N ページの終了アドレスは、ターゲットマイコンの RAM の終了アドレスを設定してください。

```
-Z(NPAGE)NPAGE,C_ARGN,N_UDATA,N_IDATA,ECSTR=100-43F
```

上記の設定は、100h 番地から 43Fh 番地までの RAM 空間に、NPAGE、C\_ARGN、N\_UDATA、N\_IDATA、および ECSTR セグメントを配置する設定です。これらのセグメントも ICC740 が使用するセグメントですので、削除しないでください。1 ページに CSTACK セグメントを配置している場合は、NPAGE セグメントは CSTACK セグメント後のアドレス (標準の設定では 140h 番地) から配置されます。終了アドレスはターゲットマイコンの RAM 領域の最終アドレスを設定してください。こ

の設定により、RAM 領域のオーバーフローが検査できます。

38C2 グループ等、SFR 領域が N ページにも存在するマイコンでは、その領域を外してください。

#### 6.2.4. ROM 領域アドレスの変更

ターゲットマイコンに合わせて ROM 領域の設定を行ってください。一般プログラム領域、スペシャルページの設定を行います。

一般プログラム領域の設定を行ってください。

```
-Z(CODE)RCODE,Z_CDATA,N_CDATA,C_ICALL,C_RECFN,CSTR,  
CCSTR,CODE,CONST=C080-FE9F
```

上記の設定は、ROM 領域の先頭アドレス C080h からスペシャルページ前 FE9Fh までの ROM 空間に、ROM 領域用セグメントを配置する設定です。

なお、予約 ROM 領域を持つマイコンでは、先頭アドレスをその領域以降にしてください。終了アドレスはスペシャルページの手前まで、または、割り込みベクトル領域の手前までを指定してください。

スペシャルページの終了アドレスは割り込みベクトル領域の手前までを指定してください。C\_FNT セグメントがスペシャルページ領域を示します。

この設定により、割り込みベクトル領域へのオーバーラップが検査できます。

```
-Z(CODE)C_FNT=FF00-FFDB
```

#### 6.2.5. 割り込みベクトル領域の変更

割り込みベクトル領域の変更を行います。

ICC740 では INTVEC セグメントが割り込みベクトル領域を示します。

```
-Z(CODE)INTVEC=FFDC-FFFF
```

上記は、3803 グループの割り込みベクトル領域の設定です。ターゲットマイコンに合わせて設定してください。

## 6.2.6. ライブラリの変更

プロセッサグループ、メモリモデルに対応したライブラリを表 8から選択してください。

**表 8 プロセッサグループ、メモリモデルとライブラリ**

	Tiny モデル	Large モデル
MUL/DIV を持つマイコン	cl7400t.r31	cl7400l.r31
MUL/DIV を持たないマイコン	cl7401t.r31	cl7401l.r31
拡張メモリアクセスを持つマイコン	cl7402t.r31	cl7402l.r31

ライブラリ指定箇所では、“-C”の記述があります。この“-C”は削除しないでください。

### 6.3. ICC740 オプションの変更

ICC740 のデフォルトのオプション設定は表 9のとおりです。

表 9 ICC740 のオプション

オプション	カテゴリ	説明
-z9	コード生成	コードサイズ優先の最適化を指定します。
-e	コード生成	拡張仕様(zpage, npage 等)を有効にします。
-K	コード生成	C++コメント(“//”)を有効にします。
-r	デバッグ	デバッグ情報を出力します。
-ml	ターゲット	メモリモデルを Large モデルにします。
-v0	ターゲット	MUL/DIV を持つ MCU を選択します。
-o	コマンドライン	オブジェクトファイル名を設定します。

ICC740 のオプションの変更は、プロジェクトエディタの[オプションブラウザ]により行います。[オプションブラウザ]の起動は、[プロジェクト]メニューの[オプションブラウザ]項目または[オプションブラウザ]ボタンにより行います(図 21)。[オプションブラウザ]を選択すると図 22の[オプションブラウザ]ダイアログが起動されます。



図 21 [オプションブラウザ]の起動

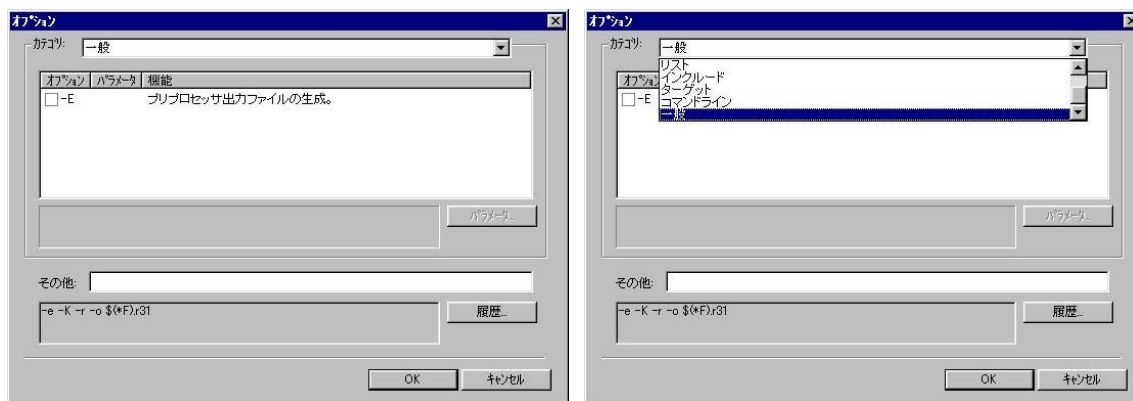


図 22 [オプションブラウザ]ダイアログ

オプションマクロの CFLAGS をクリックして、[編集]ボタンをクリックしてください。ICC740 用の[オプション]ダイアログがオープンします (図 23 (a))。

各オプションはカテゴリを選択してから、設定を行います (図 23 (b))。

なお、オプションの詳細につきましては ICC740 付属のマニュアルを参照してください。



(a) オープン時

(b) カテゴリ選択

図 23 ICC740 の[オプション]ダイアログ

### 6.3.1. プロセッサ・グループの変更

カテゴリから[ターゲット]を選択してください (図 24)。表示されるダイアログのリスト項目の[-v0, -v1, -v2]をクリックして[選択]ボタンをクリックします。[選択オプション]ダイアログが表示されますので、プロセッサグループを選択します。



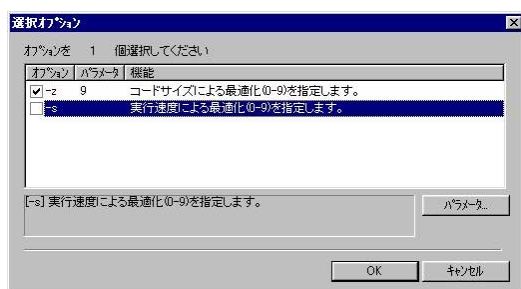
図 24 ターゲットカテゴリと[選択オプション]ダイアログ

### 6.3.2. メモリモデルの変更

カテゴリから[ターゲット]を選択してください。表示されるダイアログのリストの[-ml, -mt] をクリックして[選択]ボタンをクリックします。[選択オプション]ダイアログから、該当するメモリモデルを選択します。

### 6.3.3. 最適化レベルの変更

最適化レベルを変更する場合は、カテゴリから[コード生成]を選択してください。表示されるリストの[-z, -s] をクリックして[選択]ボタンをクリックします。[選択オプション]ダイアログ (図 25 (a)) から、[-z]または[-s]を選択します。最適化レベルを設定する必要がありますので、さらに[パラメータ]ボタンをクリックしてください。[パラメータ指定]ダイアログ (図 25 (b)) で 0~9 までの値を選択して[OK]ボタンをクリックしてください。



(a) [選択オプション]ダイアログ



(b) [パラメータ指定]ダイアログ

図 25 [選択オプション]ダイアログの最適化レベル変更

## 6.4. A740 オプションの変更

A740 のデフォルトのオプション設定は表 10のとおりです。

表 10 A740 のデフォルトオプション

オプション	カテゴリ	説明
-uN	コード生成	16 ビットアドレッシングを設定します。
-v0	ターゲット	MUL/DIV を持つ MCU を選択します。
-o	コマンドライン	オブジェクトファイル名を設定します。

A740 のオプションの変更は ICC740 オプションの変更と同じく [オプションブラウザ] ダイアログで行います。

オプションマクロの AFLAGS をクリックして、[編集] ボタンをクリックしてください。A740 用の [オプション] ダイアログがオープンします (図 26)。

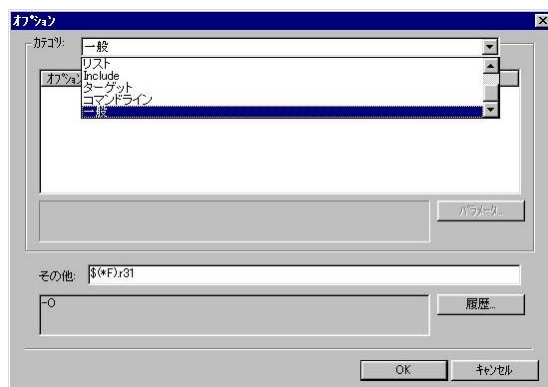


図 26 A740 の [オプション] ダイアログ

### 6.4.1. プロセッサ・グループの変更

カテゴリから [ターゲット] を選択してください。表示されるダイアログのリスト項目の [-v0, -v1, -v2] をクリックして [選択] ボタンをクリックします。[選択オプション] ダイアログが表示されますので、プロセッサグループを選択します。

#### 6.4.2. メモリモデルの変更

カテゴリから[コード生成]を選択してください。参照するシンボルが全てゼロページに配置されている場合は、表示されるダイアログのリスト項目の[-uN]オプションのチェックを外してください。[-uN]オプションは外さなくても、プログラムの動作には影響を与えません。しかし、外すことにより、ROM サイズおよび処理速度の改善が行われる場合があります。

なお、参照するシンボルがゼロページ以外にある場合に[-uN]オプションを外した場合、XLINK でエラーが発生することがあります。この場合の対処方法は8.2章を参照してください。



## 6.5. XLINK オプションの変更

XLINK のオプションは lnk740.xcl ファイルおよび TM 上のオプションで設定されます。

TM 上のデフォルトのオプション設定は表 11のとおりです。

**表 11 XLINK のデフォルトのオプション (コマンドライン)**

オプション	カテゴリ	説明
-o	出力ファイル	オブジェクトファイル名を設定します。
-Fieee695	出力ファイル	フォーマットを指定します。M3T-PD38/M3T-PD38SIM でデバッグできるように IEEE-695 形式を示す ieee695 を指定します。
-ylmba	出力ファイル	IEEE-695 の付加情報です。M3T-PD38/ M3T-PD38SIM 用には lmba を指定します。
-l	リスト制御	マップファイルを出力します。
-x	リスト制御	マップファイルに相互参照リストを出力します。
-f	Include	コマンドファイルを指定します。 TM で自動的に設定されます。

XLINK のオプションの変更は、ICC740 のオプションの設定と同じく[オプションブラウザ]ダイアログで行います。

オプションマクロの LFLAGS をクリックして、[編集]ボタンをクリックしてください。  
XLINK 用のオプション選択ダイアログがオープンします。

XLINK では、プロジェクトの編集に関するオプションはありません。

## 7. プロジェクトの開発

### 7.1. ソースファイルの作成および登録

C ソースファイル、アセンブリ言語ソースファイルはお好みのエディタで作成してください。お好みのエディタがプロジェクトバーに登録されている場合は、[エディタを開く]ボタンで起動できます(図 27)。

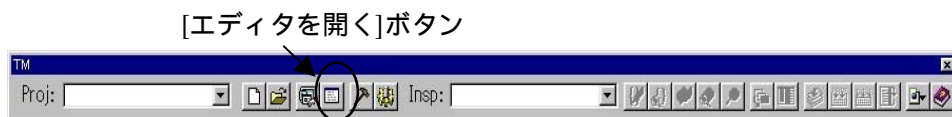


図 27 プロジェクトバーの[エディタを開く]ボタン

作成したソースファイルはプロジェクトに登録する必要があります。ワーキングディレクトリに置いただけでは登録されません。以下の手順で登録してください。

1. プロジェクトエディタで、`¥proj_1.695` ファイルをクリックして、[ファイルの追加]ボタンをクリックします(図 28(a))。
2. [開く]ダイアログがオープンしますので、[ファイルの種類]で[Source Files]を選択して、ソースファイルを追加してください。
3. `¥proj_1.695` の下に、拡張子が `.r31` となったソースファイルが登録されます(図 28(b))。
4. 拡張子が `.r31` となったソースファイルの「+」をクリックすると、ソースファイルが表示されます。

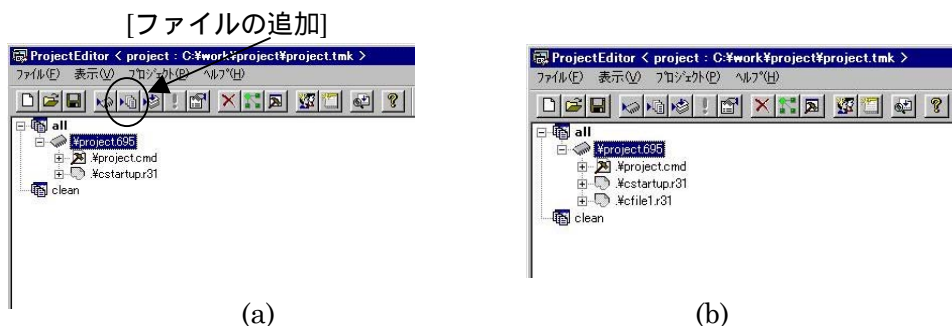


図 28 ソースファイルの登録

## 7.2. lnk740.xcl ファイルの変更

アセンブラソースファイル等で新しいセグメントを作成している場合は、そのセグメントの配置を lnk740.xcl ファイルに設定する必要があります。

以下に例を示します。

```
< sample.s31 >
...
RSEG RAM_DATA
BLKB 10
...
RSEG ROM_DATA
BYTE 'Please enter your name'
...
```

```
< lnk740.xcl >
...
-Z(NPAGE)NPAGE,C_ARGN,N_UDATA,N_IDATA,ECSTR=100-43F

-Z(NPAGE) RAM_DATA=100-43F
...
-Z(CODE)RCODE,Z_CDATA,N_CDATA,C_ICALL,C_RECFN,CSTR,CCSTR,CODE,
CONST=C080-FEFF

-Z(CODE) ROM_DATA= C080-FEFF
...
```

上記では、RAM\_DATA セグメントが、ECSTR セグメントの後に、また、ROM\_DATA セグメントが CONST セグメントの後に配置されます。

## 8. プロジェクトのビルド

ソースファイルの作成および登録が完了しましたら、ビルドによりアブソリュートモジュールを作成します。ビルドはプロジェクトバーより行います。[ビルド]、[リビルド]、または[部分ビルド]ボタンをクリックしてください(図 29)。



図 29 プロジェクトバーのビルド関連ボタン



ビルド関連ボタンをクリックすると、ビルダーが起動します。

ビルダーでは、クリックされたボタンに従い、プロジェクトのビルドを行い、結果を表示します。

エラーが発生した場合は、メッセージに従ってプロジェクトの修正をしてください。

### 8.1. C コンパイラ ICC740 およびアセンブラ A740 のエラー

C コンパイラ ICC740 およびアセンブラ A740 の実行時にエラーが発生した場合は、該当ファイルを修正してください。

ビルダーではエラー発生時に 、ウォーニング発生時に  のマークを付けて表示します。TM に登録されているエディタが、行番号を指定してオープンできる機能を持っている場合、マークの付いている行をダブルクリックすることで、エラー箇所を表示します。これにより、エラー発生箇所の特定が行いやすくなります。

## 8.2. リンカ XLINK のエラー

リンカ XLINK の実行時にエラーが発生した場合は、メッセージに従ってプロジェクトの修正をしてください。

ここでは、よく発生するエラーについて対処方法を説明します。

- Error[e16]

セグメントにデータが入りきらない場合、XLINK が以下のエラーを出力します。

```
Error[e16]: Segment セグメント is too long for segment definition.
```

この場合は、セグメントのサイズの変更またはセグメントの移動を行ってください。  
なお、以下のセグメントはゼロページから移動できません。

```
ZPAGE, Z_UDATA, Z_IDATA, C_ARGZ, EXPR_STACK, INT_EXPR_STACK
```

Z\_UDATA, Z\_IDATA, C\_ARGZ セグメントは変数が配置されているセグメントです。  
変数に npage を付加することにより、該当変数を N\_UDATA, N\_IDATA, C\_ARGN セグメントに移動することが可能です。但し、参照する側の外部宣言およびプロトタイプ宣言にも npage を忘れないようにしてください。

- Error[e18]

N ページに配置したデータを、ゼロページアドレッシングモードでアクセスした場合、以下のエラーが発生します。

```
Error[e18]: Range error in ( モジュール ), segment セグメント at address アドレス. Value 値, in tag タグ, is out of bounds (0x0-0xff)
```

この場合は N ページのデータをゼロページに配置するか、または参照方法を変更してください。

ゼロページに配置：

C ソースでは、定義時に zpage を付加してください。

アセンブラソースでは、該当シンボルがゼロページに配置されるようにしてください。

N ページに配置した場合の参照方法：

C ソースでは、外部宣言時に npage を付加してください。関数引数に npage

を指定している場合は、プロトタイプ宣言にも `npage` が必要です。

```
extern npage int n_il;  
void func( char a, npage int n_b);
```

アセンブラソースでは、参照時に”np:”を付加してください。

```
lda np:n_data
```

- Error[e27]

同名のシンボル（含 C 言語の変数および関数）が、複数のファイルで定義されている場合、以下のエラーが発生します。

```
Error[e27]: Entry "シンボル" in module モジュール 1 ( ファイル 1 ) redefined in  
module モジュール 2 ( ファイル 2 )
```

この場合は、該当ファイルを調べて、シンボル名が重複しないように変更してください。

- Error[e46]

未定義シンボルがある場合、以下のエラーが発生します。

```
Error[e46]: Undefined external "シンボル" referred in モジュール ( ファイル )
```

この場合は、シンボル名の確認を行ってください。

### 8.3. ソースファイル単位のオプションの設定

C ソースファイルおよびアセンブラソースファイルでは、ファイル単位でオプションを設定する場合があります。

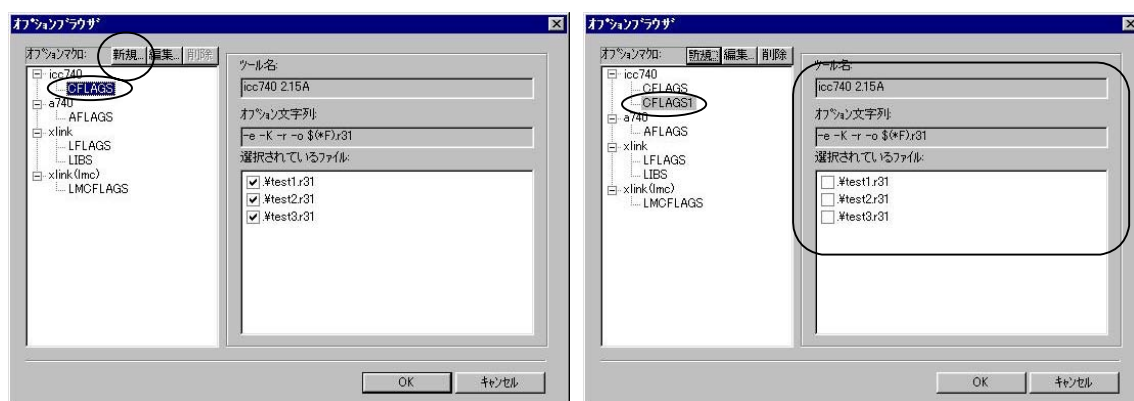
ファイル単位のオプションマクロを追加作成します。

作成方法は、C ソースファイルおよびアセンブラソースファイルともに同じですので、C ソースファイルを例に説明します。

[オプションブラウザ]ダイアログで[icc740]または[CFLAGS]をクリックします(図 30 (a))。

[新規]ボタンがアクティブになりますので、このボタンをクリックしてください。

[オプション]ダイアログがオープンされます。[オプション]ダイアログの各オプションの値は[CFLAGS]と同じになっていますので必要に応じて変更を行い、[OK]ボタンをクリックすると[CFLAGS1]オプションマクロが作成されます。これを繰り返すことにより、オプションマクロが増えていきます(図 30 (b))。



(a) オプションマクロ設定前

(b) オプションマクロ追加後

図 30 [オプションブラウザ]ダイアログでのソースファイル単位オプション設定方法

新規作成したオプションマクロは、ファイルに割り当てられていません。オプションマクロをクリックして、[選択されているファイル]項目で、ファイルをクリックすることにより、選択したオプションマクロが有効になります。

## 9. プロジェクトのデバッグ

ビルドが完了したプロジェクトでは、ワーキングディレクトリにアブソリュートファイル *proj\_1.695* が生成されます。

この、*proj\_1.695* は M3T-PD38 または M3T-PD38SIM でデバッグすることができます。TM では、プロジェクトバーの[デバッガの起動]ボタンにより、デバッグを起動することができます ( 図 31 )。

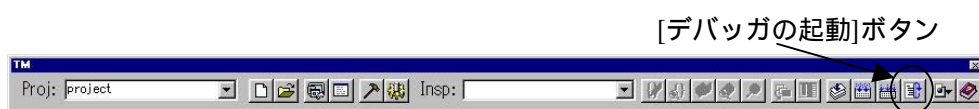
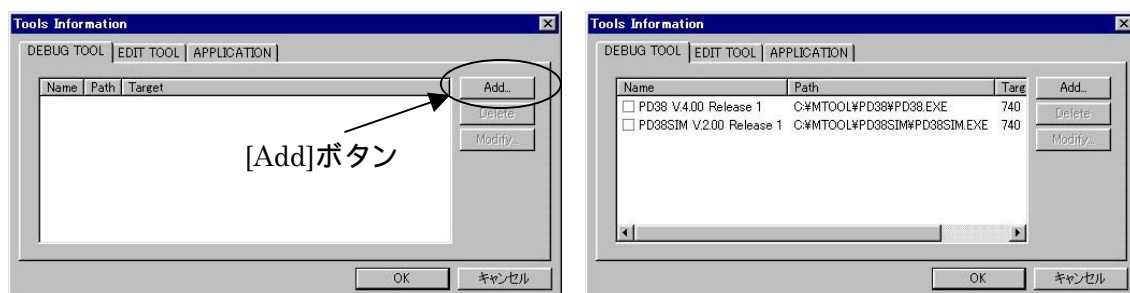


図 31 プロジェクトバーの[デバッガの起動]ボタン

起動されるデバッガは、TM に登録したデバッガです。

デバッガが登録されていない場合は、[Tool Information]ダイアログがオープンしますので、[DEBUG TOOL]タブで登録してください ( 図 32 )。



(a) リストアップ無し

(b) リストアップ有り

図 32 [Tool Information]ダイアログの[DEBUG TOOL]タブ

また、TM では TW74 インストール前に M3T-PD38 または M3T-PD38SIM がインストールされている場合、これらを自動的にリストアップします ( 図 32 (b) )。

TW74 インストール後に M3T-PD38 または M3T-PD38SIM をインストールした場合は、リストアップされません。TW74 を再度インストールすることで、自動リストアップが行われます。

リストアップされたデバッガを登録するには、チェックボックスのチェックを行ってください。

その他のデバッグツールは自動的に表示されませんので、[Add]ボタンによりデバッグツールを登録してください。



M3T-PD38 または M3T-PD38SIM は TW74 インストールの後にインストールすると、他のデバッグツールの登録と同じように登録する必要があります。この場合、他のデバッグツールと同じように Name 欄に”user”が付きます。違いは[デバッガの起動]ボタンでデバッグツールを起動した場合、プロジェクトの.695 ファイルが自動的にロードされるか、されないか、ということです。

Name 欄に”user”が付いた場合は自動的にロードされませんので、M3T-PD38 または M3T-PD38SIM の[File]メニューによりプロジェクトの.695 ファイルをロードする必要があります。

[デバッガの起動]ボタンのクリックにより M3T-PD38 または M3T-PD38SIM の自動ダウンロードを行う場合は、TW74 を再度インストールしてください。

## 10. ヘキサファイルの作成

Hex ファイルを作成する場合は、ロードモジュールコンバータとして xlink(lmc)を TM に登録してください。

まず、プロジェクトエディタの[プロジェクト]メニューの[情報]項目を選択してください。[プロジェクトプロパティ]ダイアログ (図 33) がオープンされますので[ツール]タブを選択してください。



図 33 [プロジェクトプロパティ]ダイアログ

次にパッケージ情報の[xlink(lmc)]項目にチェックを入れて[OK]ボタンをクリックしてください。

[オプションブラウザ]ダイアログに xlink(lmc)が登録され (図 34) xlink(lmc)が xlink の後に実行されるようになります。



図 34 [オプションブラウザ]ダイアログの xlink(lmc)登録

なお、xlink(lmc)の TM 上のオプション設定は表 12のとおりです。

表 12 xlink(lmc)のデフォルトのオプション(コマンドライン)

オプション	カテゴリ	説明
-o	出力ファイル	オブジェクトファイル名を設定します。
-Fintel-standard	出力ファイル	フォーマットを指定します。インテル Hex ファイルを出力するため intel-standard を指定します。
-Y0	出力ファイル	intel-standard の付加情報です。マスクファイル変換ユーティリティ MM 用に 0 を指定します。
-f	Include	コマンドファイルを指定します。 TM で自動的に設定されます。

ビルドを実行すると IEEE695 ファイルと Hex ファイルを作成するようになります。

xlink(lmc)は、xlink の生成ファイルを IEEE695 から Hex に変更したものです。xlink と同じ lnk740.xcl ファイルを使用します。

# MEMO

# M3T-ICC740 V.1.00A ユーザーズマニュアル

---

Rev. 1.00  
03.07.01  
RJJ10J0179-0100Z

COPYRIGHT ©2003 RENESAS TECHNOLOGY CORPORATION  
AND RENESAS SOLUTIONS CORPORATION ALL RIGHTS RESERVED

M3T-ICC740 V.1.00A  
ユーザーズマニュアル



ルネサスエレクトロニクス株式会社  
神奈川県川崎市中原区下沼部1753 〒211-8668

RJJ10J0179-0100Z