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い 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

- 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ものですが、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、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 て、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。



740ファミリ用 Cコンパイラパッケージ V.1.01 ^{ユーザーズマニュアル}



Rev.1.01 2007.06

	→→→→→→ 本資料ご利用に際しての留意事項 →→→→→→→→→→→→
1.	本資料は、お客様に用途に応じた適切な弊社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弊社または第三者の知的財産権その他の権利の実施、使用を許諾また
	は保証するものではありません。
2.	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など全ての情報
	の使用に起因する損害、第三者の知的財産権その他の権利に対する侵害に関し、弊社は責任を負い
	ません。
3.	本資料に記載の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、あるいはその他
	軍事用途の目的で使用しないでください。また、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、それらの定めるところにより必要な手続を行ってください。
4	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などの全ての
	情報は本資料発行時点のものであり。 整社は本資料に記載した製品または仕様等を予告なしに変更
	することがあります。整社の半導体製品のご購入およびご使用に当たりましては、事前に整社営業
	ッ の こ こ が の う の う 。 テ にの 「 寺 戸 祝 品 の こ 点 う の ひ ひ こ に 「 か に う に う の ひ こ に 、
	通じて公開される情報に堂にご注音下さい
5	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が 万一本資料の記述の誤りに
5.	本資料に記載した情報は、正確で到すため展望に助けしたものですが、方 本資料の記述の戻りに お田する場実がお安様に生じた提合においてま 酸社けその書任を負いません
6	たる料に記載の制品データ 図 素たどに示す技術的な内容 プログラム アルゴルズムその他
0.	本具料に記載の表面が、ク、、、、、、、、、、、、、、、、、、、、、、、、、、、、、、、、、、、
	主体で「力に計画し、の各体の負任にのいて適用う日を判断して下てい。并性は、適用う日に対する害任け合いません。
7	カラる貝はほういるとい。 太容料に記載された制具は、冬種完全装置や運輸・充通田、医病田、燃焼制御田、航空宇宙田
/ .	平貝科に記載C10に表面は、百怪女主衣直に運動。又遮而、広原而、燃焼耐岬而、加工于田市、 百乙市、海底市線田の燃業。シフテムなど、その故院や記動作が古法本会た務がしたるいけ本体に
	尿丁刀、 (海底中総用の)(成品・クス) ムなと、 ての) 取得 (2 武動) にか 直接入 叩 を 育か しのるい は 八 仲 に 合実 た 乃 ぼ す お そ わ の あ る ト こ か 機 哭 ・ シ フ テ し や 特 に 喜 度 か 只 感 ・ 信 語 州 が 亜 立 さ わ る 機 哭 ・
	心苦を及ばすのとれののなるうな機能、ノステムで行に同反な加負、 に根にが安水される機能、
	システムしの使用を息凶して設計、表担されたものではのりません(笄柱か日動車用と相圧する制度を自動車に使用する埋合を除きます)。これにの用途に利用されることをご検討の際には
	表面を日勤半に使用する场口を际さよす)。これらの用途に利用されるここをしていの际には、 必ず車前に戦社営業家口を「昭今てさい」なな、トロ田冷に使用されたことにより発生した提案等
	必り争前に笄社呂未芯山へこ思云下さい。なの、上部用返に使用されたことにより光主した損苦寺 について敵社はその妻にた色いかわますのでごて丞頤います
o	について笄社はての員任を見いかねよりのてこ」承願いより。 第7項にかかわたず、太洛料に記載された制中は、て記の田途には使用したいでてさい、これたの
ο.	第7頃にかかりうり、半貝科に記載された装加は、下記の用座には使用しないて下さい。これらの 田冷に使用されたニレにトリ惑生」た提定笑につきましてけ、敵社は一切の害にた合いません
	□ 小 ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○
	2) 人体に押め込み使田さろもの
	2) 八体になり込み反用する 500。 2) 治病行为(串郭切り出し、 液剤処片等) た行からまの
	3)石原口病(芯叩切り山し、采用112月年)を11なりもの。 4)その他 百座人会に影響を与えるもの
٥	*)での他、直接八中に影響でうれるもの。 本容料に記載された制品のご体田につき、性に易士定枚、動作電循電圧範囲、故熱性性、実装冬性
9.	本員科に記載された表面のと使用につき、特に取べたれ、動作電脈電圧範囲、 放然特に、 夫衣示作 お上がその他送冬姓につきましてけ 一酸社保証範囲内でご使用ください 一酸社保証値を拭えて制品
	のよいての他間示けにつきよりては、井牡床亜単四内でと使用くたとい。井牡床亜値を越んて表面 たご信用された提合の故障お上が重故につきましてけ、敵社けその害任た色いません。
1 0	をこぼ用されに物口の収焊のよび事以にうさよりては、井社はての員任を良いよされ。 敵社け制只の只質おとび信頼州の向上に努めておりますが、特に半道休制只けある確率でお陪が
10	発生したり
	元王したり、使用赤叶によりては映動中したりする場合があります。并任表面の政障なたは映動下が生じた提合主人自主物の必須支払う社会的指定などた生じさせないとう。お友達の書任において
	の主いた場合も八才争取、八次争取、社会的損害などを主じてきないより、の各様の負任にのいて 同目記針 孤焼対笑記針 誤動作防止認計などの空会認計(会れルードウェアなとび)フトウェ
	ルで設計、延焼対象設計、鉄動作的工設計などの女主設計(含むハートウエアのよびファトウエ マンキト7パエージング加理等 機器またけシフテムとしての出荷保証をお願いいたします。特に
	アノのよびエーシノソ処理寺、機能よにはシステムとしての山何休証をの願いいたします。村に
	マイコノシノドウエアは、半強しの快証は困難なため、の合体が装迫された取除の機品・システム
1 1	こしてい父主!!?!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
1 1	・ 半貝科に記載の表面は、 これで活動した表面から剥かれた场合、 初先か日に入れて決断する寺の 車坊の各陸州がおります。 お安塔の制中への実装後に家目に太制中が割がわてて レがかキャン
	⇒RUURRはルのリより。の合体U表面へU天衣役に谷勿に平炭面が割かれることがなさよう、 な安塔の書任において十公な空会認計たお願いします。な安塔の制口なら割がわた担合の声かに
	の合体の貝讧にのいて「カな女土衣司での願いしより。の各体の发血から刻かれに场音の事故に つきましてけ 敵社けるの害任た色いません
1 0	ノこよしては、汗化はてい貝にで良いよとい。 太姿料の今朝またけ二朝た敵社の立まにトス車前の予護かしに転載またけ海制オスニレを中ノ
ΙZ	・ 半貝科の主命よには一部で汗性の入音による事前の単語なしに転戦まには後表することを回く や いかします
	ことで、「ないない」では、「ないない」では、「ないない」では、「ないない」では、「ないない」では、「ないない」では、「ないない」では、「ないない」では、「ないない」では、「ないない」では、「ないない」では、

13.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弊社営業 窓口までご照会下さい。

製品の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、コンタクトセンタ csc@renesas.comまで送信ください。 ¥SUPPORT¥製品名¥SUPPORT.TXT

株式会社ルネサス	テクノロジ
コンタクトセンタ	csc@renesas.com
ユーザ登録窓口	regist_tool@renesas.com
ホームページ	http://japan.renesas.com/tools

- 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商標または登録商 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。

目次

は	じめに	<u>-</u>	8
1.	概要	ह	9
2.	クイ	イックツアー1	10
-	2.1.	新規のプロジェクトを作成する1	0
4	2.2.	ソースファイルを作成し、登録する1	3
-	2.3.	ビルドする1	6
6	2.4.	デバッガを起動する1	6
	2.4.	.1. プログラムをロードする	17
	2.4.	.2. main関数まで実行する	18
	2.4.	.3. 割り込み発生の確認を行う	19
3.	プロ	コジェクトの新規作成2	!1
	3.1.	新規作成2	21
4.	プロ	コジェクトの編集2	27
4	4.1.	オプションの編集	27
	4.1.	.1. プロジェクトオプションの設定	28
4	4.2.	ICC740 オプションの設定	29
	4.2.	.1. ヘッダファイルの登録	30
	4.2.	.2. リストファイルの作成	31
4	4.3.	A740 オプションの設定8	31
4	4.4.	XLINKオプションの設定	33
5.	プロ	コジェクトの開発	35
ļ	5.1.	ソースファイルの作成および登録	35
ł	5.2.	セグメントの追加	35
ł	5.3.	セグメントの変更	36
6.	プロ	コジェクトのビルド 8	37
(6.1.	CコンパイラICC740 およびアセンブラA740 のエラー	37
(6.2.	リンカXLINKのエラー	37
(6.3.	リンカXLINKの注意事項	39
7.	プロ	コジェクトのデバッグ4	1

8.	ヘキサ	ファイルの作成4	2
9.	リビジョ	ョンアップ時の注意4	3
9	.1. V.1	1.01 RELEASE 01 からのリビジョンアップ4	3
9	.2. V.1	1.01 RELEASE 01 で作成したプロジェクトのコンバート	3
10.	CSTA	ARTUP.S31 とLNK740.XCLファイルの編集 4	:6
1	0.1.	CSTARTUP.S31 の編集	6
	10.1.1.	スタックページの変更	16
	10.1.2.	割り込みベクトル領域の変更	16
1	0.2. 1	LNK740.XCLファイルの編集4	7
	10.2.1.	スタック領域の変更4	!8
	10.2.2.	ゼロページ先頭アドレスの変更	!8
	10.2.3.	Nページ終了アドレスの変更	19
	10.2.4.	ROM領域アドレスの変更	19
	10.2.5.	割り込みベクトル領域の変更	50
	10.2.6.	ライブラリの削除	50
	10.2.7.	セグメントの追加	51

図目次

図 1: [NEW PROJECT WORKSPACE]ダイアログボックス	10
図 2: [NEW PROJECT]ウィザード	
図 3: コンフィグレーションとセッションの表示	12
図 4: ワークスペースウィンドウのプロジェクトタブ	12
図 5:ファイルを登録したプロジェクトタブ	13
図 6: [ビルド]、[すべてをビルド]、[ファイルのコンパイル]ボタン	16
図 7:セッションの表示	16
図 8:740 シミュレータのINITダイアログ	16
図 9: [ダウンロード]が追加されたプロジェクトタブ	17
図 10:ダウンロード後のプロジェクトタブ	17
図 11: リセット後のCSTARTUP.S31	
図 12: ブレークポイントの設定	
図 13: DEBUG RUN TOOLBAR	
図 14:ブレークポイントでのプログラム停止	19
図 15:C WATCH WINDOW	19
図 16: [NEW PROJECT WORKSPACE]ダイアログボックス	21
図 17: [NEW PROJECT]ウィザード:ステップ1	22
図 18: [NEW PROJECT]ウィザード:ステップ 2	22
図 19: [NEW PROJECT]ウィザード:ステップ 3	23
図 20: [NEW PROJECT]ウィザード:ステップ4	23
図 21: [NEW PROJECT]ウィザード:ステップ 5	24
図 22: [Toolchain]ダイアログボックス	27
図 23: [TOOLCHAIN]ダイアログボックス: CタブのCATEGORY:SOURCE	
図 24: CタブのCATEGORY: LIST	
図 25: LINKタブ	
図 26: CATEGORY:MEMORYの設定	
図 27: [MODIFY SEGMENT]ダイアログボックス	
図 28: [ビルド]、[すべてをビルド]、[ファイルのコンパイル]ボタン	
図 29:セッション	
図 30:740 シミュレータのINITダイアログ	41
図 31: [TOOLCHAIN]ダイアログボックスのLINKタブのCATEGORY:OUTPUT	
図 32:ツールチェインの変更メッセージ	43
図 33:ツールチェイン変更後のLNK740.XCLファイルの設定箇所	

表目次

表	1:コンフィグレーションとセッション	12
表	2:コンフィグレーションとセッション	26
表	3: CPUタブ	.28
表	4:LARGEモデルとTINYモデル	.28
表	5 : Cタブ	29
表	6:ICC740 のデフォルトオプション	.29
表	7: Assemblyタブ	31
表	8:A740 のデフォルトオプション	32
表	9:LINKタブ	.33
表	10:XLINKのデフォルトオプション	.33
表	11: TYPE OF OUTPUT FILEの内容	42
表	12: CSTARTUP.S31 の編集項目	46
表	13:LNK74.XCLの編集項目	.47

はじめに

このユーザー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

このユーザーズマニュアルでは、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

・High-performance Embedded Workshop を初めてご使用になる方

■参考マニュアル

High-performance Embedded Workshop に関する詳しい用語説明、機能説明、操作については、High-performance Embedded Workshop のユーザーズマニュアルを参照してください。

IAR Systems 社製 C コンパイラ ICC740 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

740 ファミリ用 C コンパイラパッケージは IAR Systems 社製 C コンパイラ ICC740 (以下 ICC740 と略します)と High-performance Embedded Workshop を組み合わせた開発環境 システムです。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。

以下のソフトウェアを同梱しています。

 \cdot SC74

ルネサス製 740 ファミリ用アセンブラパッケージで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、High-performance Embedded Workshop での ICC740 プロジェクトの 開発方法について説明します。

2. クイックツアー

本章ではクイックツアーにより ICC740 を使用した一般的な開発手順を説明します。 各手順の詳細は3章以降を参照してください。

2.1. 新規のプロジェクトを作成する

新規ワークスペースを作成します。

740 ファミリ用C コンパイラパッケージの[新規プロジェクトワークスペース]ダイアログ ボックス (図 1) で、以下の選択を行います。

CPU family :	740 Family
Tool chain :	IAR ICC740 (740 Family)

新規プロジェクトワークスペース			? 🔀
新規プロジェクトワークスペース プロジェクトタイプ ⑦ Application (Enhance Version) ⑦ Empty Application ● Library	ワークスペース名(W): フロジェクト名(P): ディレクトリ(D): C:¥WorkSpace CPU種別(Q): 740 Family ツールチェイン(T): IAR ICC740 (740 Family)		? 区
ブロパティ			
		OK	

図 1: [New Project Workspace]ダイアログボックス

プロジェクトタイプで[Application (Enhance Version)]を選択後、ワークスペース名、プロ ジェクト名を設定して、OK ボタンをクリックします。 [New Project]ウィザード(図 2)によりプロジェクトを作成します。 [New Project]ウィザードの(3)では、[740 Simulator]にチェックマークを指定します。



(5) ステップ 5

(6) Summary

図 2: [New Project]ウィザード

740 ファミリ用 C コンパイラパッケージでは、以下のコンフィグレーションとセッション を作成します。

表 1: コンフィグレーションとセッション

Configuration

8	
Debug	デバッグ用コンフィグレーション
Release	リリース用コンフィグレーション
Session	
DefaultSession	ターゲットを選択していないセッション
Session740_Simulator	シミュレータ用セッション

新規ワークスペース作成時はコンフィグレーションとしてDebug、セッションとして DefaultSessionが選択されています(図 3)。

🛛 🧇 🛗 🚟 👗 Debug 💽 DefaultSession 📼	I
------------------------------------	---

図 3: コンフィグレーションとセッションの表示

また、ワークスペースウィンドウのプロジェクトタブは以下の表示となります(図 4)。



図 4: ワークスペースウィンドウのプロジェクトタブ

2.2. ソースファイルを作成し、登録する

ソースプログラムを作成します。今回は、SmpTw74フォルダにある tutor3.c を使用します。 tutor3.c をプロジェクトフォルダにコピーしてください。

SmpTw74 フォルダは ICC740 をインストールしたフォルダ (通常は¥Program Files¥IAR Systems¥ew23¥740) に作成されています。

ファイルをプロジェクトフォルダに配置しただけでは、登録されませんので、[プロジェクト]メニューの[ファイルの追加...]により登録してください(図 5)。



図 5:ファイルを登録したプロジェクトタブ

また、proj_1.c と intf.c は以下のように修正してください。エディタはファイルをダブルク リックすることで起動できます。

proj_1c の修正			
#include <intr740.h></intr740.h>			
void tutor3(void);			
void main(void)			
tutor30;			





ソースファイル tutor3.c について:

ソースファイル tutor3.c は、IAR Systems 社作成のサンプルプログラムを 740 シミュレー タ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 tutor30 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ï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、tutor30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 <li
                       -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
                                *
 yoid interrupt [0x00] brk_interrupt(void)
//
    putchar('I');
   my_char='.';
yoid execute_brk(void)
   break_instruction(); /* Use intrinsic function */
void do_foreground_process(void)
    call_count++;
    putchar(my_char);
\parallel
    my_char = rand() \% 26 + 'a';
yoid tutor3(void)
    while (1)
       do_foreground_process();
if (my_char=='i') execute_brk();
   }
```

2.3. ビルドする

ビルドは、[ビルド]、[すべてをビルド]、または[ファイルのコンパイル]ボタンにより行います。ここでは、[すべてをビルド]ボタンをクリックしてください(図 6)。

🛛 🕸 🛗 📥 👗 🛛 Debug	DefaultSession	•
-------------------	----------------	---

M	۶.	「ドルド]	[オベアたビルド]	[ファイルのコンパイル]ボタン
凶	0:		$[9 \land (\mathcal{E} \vdash \mathcal{V} \vdash)]$	[ノアイルのユンハイル] 小グン

2.4. デバッガを起動する

ビルドによるエラーの発生が無くなれば、デバッグを行います。

デバッガを起動するためセッションを選択します(図7)。

今回は、740シミュレータを使用しますので、Session740_Simulatorを選択します。

🗇 🛗 🛗 👗 Debug	✓ DefaultSession ✓	XQ
	DefaultSession Session740_Simulator	1

図 7:セッションの表示

Session740_Simulatorを選択すると以下のダイアログが起動されます(図 8)。

Init (740 Simulator)	
MCU デバッグ情報 起動スクリプト	
MCU:	步 照
時間計測リソース 動作周波数: <u>3</u> MHz / 2	
OK キャンセル 道用(A) 「今後、このダイアロ	

図 8:740 シミュレータの Init ダイアログ

このダイアログでは MCU を選択する必要があります。[Refer..]ボタンをクリックして M38000.mcu ファイルを選択してください。

[OK]ボタンをクリックすると、図 9の表示になります。



図 9: [ダウンロード]が追加されたプロジェクトタブ

2.4.1. プログラムをロードする

プログラムのダウンロードは、[デバッグ]メニューの[ダウンロード]で行います。 ダウンロードが完了するとプロジェクトタブ上の[ダウンロード]のアイコンが変わります(図 10)。



図 10:ダウンロード後のプロジェクトタブ

プログラムを実行するには、まず CPU のリセットを行ってください。

[デバッグ]メニューの[CPU のリセット]を選択します。

cstartup.s31 ファイルが図 11のように表示されます。

6			; RLODE - WN	ere the execution actua	Ily Degi	ns
5			; PCC PCO	NE • POOT		
í			init C	DE.NOOT		
1	01080	⇔	CLD		; se	t default mode
2	01081		CLT			
3	01082		LDM	#CPUM_INIT, 3BH	; se	t stack page
1	01085		LDX	<pre>#LOW (SFE(CSTACK)-1)</pre>	; se	t up stack pointer
-	01097		PVT			

図 11: リセット後の cstartup.s31

2.4.2. main関数まで実行する

先ず、main0関数まで実行します。 main0関数にブレークポイントを設定します。 proj_1.c をダブルクリックしてファイルをオープンしてください。 そして、行番号 7 の表示場所にマウスを移動してダブルクリックください。 図 12のようにブレークポイントが設定されます。



図 12: ブレークポイントの設定

この状態で[実行]ボタンをクリックしてください(図 13)。



🗵 13 : Debug Run toolbar

この位置までのプログラムを実行し、行番号7には黄色の矢印が付加されます(図 14)。



図 14: ブレークポイントでのプログラム停止

次に[ステップイン]ボタンをクリックしてください。tutor3.c が表示されます。

2.4.3. 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_brk0関数内の break_instruction0関数が行います。この break_instruction0関数は BRK 命令に置き換えられるインライン関数です。

execute_brk0関数は変数 my_char がï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、[C ウォッチ]ウィンドウで行います。[表示]メニュ ーから[シンボル]を選択し、[C ウォッチ]を選んでください。

[C ウォッチ]ウィンドウがオープンします (図 15)。

×	🗁 🗙 🖉 16 2 Per 🔯 👼
	Watch Local File Local Global
	Name Value
	(signed short) call_count 0
	(unsigned char) my_char 42 '*'

🗵 15 : C Watch Window

この中から[Global]タブを選択してください。

tutor3.c の行番号 37 と 57 にブレークポイントを設定して、[実行]ボタンを何度かクリック してください。

変数 my_char がïiに設定された時に、次の[実行]ボタンのクリックで行番号 37 黄色の矢印 となっていることで割り込み発生の確認を行います。

これにてクイックツアーを終了します。

3. プロジェクトの新規作成

740 ファミリ C コンパイラパッケージの[新規プロジェクトワークスペース]ダイアログボ ックス (図 16) で、以下の選択を行います。

> CPU family : 740 Family Tool chain : IAR ICC740 (740 Family)

新規プロジェクトワークスペース			? 🛛
新規プロジェクトワークスペース フロジェクト フロジェクトタイプ Pplication (Enhance Version) で Empty Application しibrary	ワークスペース名(W): プロジェクト名(P): プロジェクト名(P): 「 ディレクトリ(D): [C*WorkSpace] CPU種別(Q): [740 Family ツールチェイン(T): [IAR ICC740 (740 Family)		? 🗙 参照(<u>D</u>)
]	ОК	

図 16: [New Project Workspace]ダイアログボックス

3.1. 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。740 ファミリ C コンパイラパッケージ V.1.01 Release 02 では、[New Project]ウィザードにより新規プロジェクトを作成します。

New Project-1/4-Select Target CPU.To	polchain version	? 🔀
	Toolchain version :	
	1.02.00	-
	Which CPU do you want to use for this project?	
	CPU Series:	_
124	38000	
B		
	CPU Group:	_
	3803H 38031	^
	3804	
2000	3804H 3804L	~
Manager and P.	If there is no CPU group to be selected, select the "CPU Group" that a similar to hardware specification or select "Other"	
< Back	Next > Finish Ca	ncel

図 17: [New Project]ウィザード: ステップ1

CPU のシリーズとグループを選択します。

New Project-2/4-Setting of Gener	ration file and Options	? 🛛
and the second s	Memory Model:	
	RAM End Address:	0083F
	ROM Top Address:	01080
	-Stack Size :	
	CSTACK:	0x00040
	EXPR_STACK:	0×00010
al al	INT_EXPR_STACK:	0×00010
	Generate main0 Funct	ion
PP PP PP	C source file	•
< Back	Next >	Finish Cancel

図 18 : [New Project] ウィザード : ステップ 2

メモリモデル、RAM と ROM のアドレス、およびスタックサイズを設定します。 これらの値は各 CPU グループのメモリ容量が最小の場合の値を表示していめす。 ご使用の製品に合わせて変更してください。

New Project-3/4-Setting the Tar	get System for Debugging 🛛 🛛 💽
	Targets : 740 Compact Emulator 740 Simulator 740 Simulator External Debugger :
<pre> Back</pre>	Target type : 38000

図 19 : [New Project]ウィザード : ステップ 3

デバッグターゲットを選択します。図では既にコンパクトエミュレータがインストールされている場合となります。

デバッグターゲットを選択すると以下のダイアログボックスを表示します。

New Project-4/5-Setting the Debugg	er Options
	Target name : 740 Simulator Configuration name : D to 740 Simulator
	Detail options :
	Item Setting
S Back	Next > Finish Cancel

図 20: [New Project] ウィザード: ステップ 4

コンフィグレーション名を設定します。

File Na Ext Description cstartup s31 cstartup file intf c intf file mcu_def inc mcu define str_3803h h sfr file proj_1 c Main Program		The followin generated:	ng sourc	e files will be	
Marian and M		File Na cstartup intf mcu_def sfr_3803h proj_1	Ext s31 c inc h c	Description cstartup file intf file mcu define sfr file Main Program	
	PRONS	<			>

図 21 : [New Project] ウィザード : ステップ 5

プロジェクトに登録されるファイルを表示します。

表 2: 生成ファイル

ファイル	内容
cstartup.s31	スタートアップファイル。
	[New Project]ウィザードのステップ1の内容が反映される
intf.c	割り込み関数のテンプレート
mcu_def.inc	MCU 情報ファイル。
	[New Project]ウィザードのステップ1の内容が反映される
sfr_3803h.h	MCU に対応した SFR ヘッダファイル。
	[New Project]ウィザードのステップ1の内容が反映される
proj_1.c	関数 main()を含んだ C ソースファイル

ご使用のマイコン、システムに応じて変更してください。ご使用のマイコンについては、 マイコンデータシートを参照してください。 ・mcu_def.inc の説明

インクルードファイル (mcu_def.inc) は、cstartup.s31 でインクルードしています。 下記内容をご使用のマイコンにあわせて設定してください。

	CPU -	モードレジスタ (003B₁₆番地) の設定	値	
#define CDUM INIT ACH デ体田のマイコンの設定				
#define QZ 1			07	
#define IDCODE 0		マスク BOM 版	0	0
		OzBOM版	1	0
		フラッシュ版またけ	0	1
		ID コードが必要たもの	Ŭ	
		 ・・・・・・・・・・・・・・・・・・・・・・・・・・・・・	って思わる	は合があり
		本 設定値は、イーン 印催によ	ノンカルー	ドファイル
RSEG RESERVE1		(meu define) のフメント	な参照して	ください
BLKB 01H				
BLKB 01H		<qzrom 版のマクロ=""></qzrom>		
BLKB 01H		機能設定 ROM 領域及び ROM	コードプロ	テクトの
BLKB 01H	K-T	設定。基本的に、BLKB を記載	している行	の変更は
BLKB 01H		必要ありません。		
BLKB 01H		※この設定を含むオブジェクトは	t、そのまま	マスク
RSEG FUNCTION SET ROM	1	ROM 版に使用できます。		
RSEG RESERVEZ		機能設定 ROM データ	の設定	
			ı	
IDCODEMACRO MACRO		フロクラム上では影	えをしません)
RSEG ID CODE				
BYTE OFFH		<フラッシュ版または ID コードか	ぶ必要なもの)のマクロ>
BYTE 0FFH		ID コード及び ROM コードプロ	ュテクトの副	子定
BYTE 0FFH		※この設定を含むオブジェクトに	ト そのます	マスク
BYTE 0FFH	\sim	BOM版に使用できます	~~ C~~ ~ ~ ~	
BYTE 0FFH				
BYTE OFFH		ID コードの設定		
BOMOD:				
		ROM コードプロテク	トの設定	

<注意事項>

- "CPUM_INIT"はリセットスタート時に CPU モードレジスタへ設定する値です。
 CPU モードレジスタには1度しか書き込みできないビットを含むマイコンがあります。対象マイコンのデータシートでご確認ください。
- #define 行には、コメントを書かないでください。

・intf.c の説明

割り込み関数ファイル(intf.c)は、各マイコンで持っている割り込み関数を記述しています。 ご使用になる割り込み関数のプログラムをこのファイル上に記述して使用できます。必要 に応じてプロジェクトファイルに追加してください。

void interrupt[0] BRK(void){		
}		
void interrupt[2] AD_SIO3T(void	d){	
}	割り込み関数名	
void interrupt[4] Int4_CNTR2(void){		
} : 割り込みべ :	クタの先頭アドレスからのオフセット値	

表 3のコンフィグレーションとセッションを作成します。

表 3: コンフィグレーションとセッション

Configuration

Debug	デバッグ用コンフィグレーション
Release	リリース用コンフィグレーション
Debug_740_Simulator	シミュレータ用コンフィグレーション
	[New Project]ウィザードのステップ1

Session

DefaultSession	ターゲットを選択していないセッション
Session740_Simulator	シミュレータ用セッション
	デバッグターゲットに"740 Simulator"を選択した
	場合のセッション。
Session740_Compact_Emulator	コンパクトエミュレータ用セッション。
	デバッグターゲットに"740 Compact Emulator"を
	選択した場合のセッション。

4. プロジェクトの編集

新規プロジェクト作成後、必要に応じてコンパイルオプション等を変更する必要があります。

4.1. オプションの編集

オプション編集は、[ビルド]メニューの[IAR ICC740 Toolchain...]により行います。この 選択により 図 22の[Toolchain]ダイアログボックスが起動されます。

IAR ICC740 (740 Family) Standard To Configuration : Debug All Loaded Projects Converse file	olchain C Assembly Link Librarian CPU Toolchain Option Category: Source Show entries for: Include file directories
Assembly source file	Add Insert Remove Move up
	Move down Options C : -v0 -ml "-0\$(CONFIGDIR)¥" -r -z9 -e -K -h OK

図 22: [Toolchain]ダイアログボックス

4.1.1. プロジェクトオプションの設定

[New Project]ウィザードにより、CPUタブに表 4の内容を設定します。

表 4: CPU タブ

Item	内容	該当 option
Processor	・740 with MUL/DIV 一般的なマイコン	-v0
configuration	• 740 without MUL/DIV	-v1
	MUL/DIV 命令が無いマイコン	
	•740 with MUL/DIV and extended data memory access	-v2
	拡張メモリアクセスを持つマイコン	
Memory model	• Large	-ml
	• Tiny	-mt
	• 0 page only	-mt

・Large モデルと Tiny モデルについて

Large モデルと Tiny モデルでは、変数の配置場所が異なります。

Large モデルでは 0x100 番地以降に配置され、Tiny モデルでは 0x00 番地から 0xFF 番 地に配置されます。

Large モデルで変数を 0x00 番地から 0xFF 番地に配置するには拡張キーワード zpage を 使用します。アセンブルソースプログラムではオペランド zp:を用いてアクセスすること でコードサイズを削減可能です。

Tiny モデルで変数を 0x100 番地以降に配置する場合は、拡張キーワード npage を使用し ます。アセンブルソースプログラムではオペランド np:を用いてアクセスします。 オペランド np:を使用しない場合、リンク時にエラーが発生します。

表 5: Large モデルと Tiny モデル

項目	Large モデル	Tiny モデル
変数の配置場所	0x100 番地以降	0xFF 番地以下
C言語での0x100番地以		npage を使用して定義。
降への配置		npage int v1;
		extern npage int v2;
C 言語での 0xFF 番地へ	zpage を使用して定義。	
の配置	zpage int v3;	
	extern zpage int v4;	
アセンブルソースプロ		オペランド np:を使用
グラムでの 0x100 番地		lda np∶v1
以降のアクセス方法		
アセンブルソースプロ	オペランド zp:を使用するこ	
グラムでの 0x100 番地	とでコードサイズ削減	
以下のアクセス方法	lda zp:v3	

拡張キーワード zpage, npage は外部変数、自動変数、および関数引数に指定できます。

4.2. ICC740 オプションの設定

オプションの設定はCタブで行います。Cタブの内容は表 6の通りです。

Category	Show entries for	Item	該当 option
Source	Include file	登録 box	·I
	directories		
	Defines	登録 box	-D
	Undefines	チェック box	-U
Object	Output	Global strict type check	-g
		Output directory	-0
	Debug	Generate debug information	-r
List		Generate list file	-L, -q, -i, -T, -t
Optimize		Optimization	-z, -s
Other		Miscellaneous options	-e, -c, -K , -C, -w -h
		User defined options	その他

表 6:Cタブ

オプションの詳細は、ICC コンパイラ・プログラミング・ガイド (icc740_jp.pdf)の 27~ 52ページを参照してください。ICC コンパイラ・プログラミング・ガイドの GUI 画面は IAR Systems 社の Embedded Workbench のものです。

ICC740のデフォルトのオプション設定は表 7のとおりです。

表 7: ICC740 のデフォルトオプション

オプション	説明
-z9	コードサイズ優先の最適化で最高レベルを指定します。
-е	拡張仕様(zpage, npage 等)を有効にします。
-K	"//"コメントを有効にします。
-r	デバッグ情報を出力します。
-ml	メモリモデルを Large モデルに設定します。
-mt	メモリモデルを Tiny モデルまたは 0 page only モデルに設定します。
-v0	MUL/DIV を持つ MCU を選択します。(一般的な MCU です。)
	オブジェクトファイルフォルダを設定します。
-0	\$(CONFIGDIR)はDebugフォルダまたはReleaseフォルダを指定しま
	す。
-h	多重割り込みをサポートします。

4.2.1. ヘッダファイルの登録

740 ファミリ用 C コンパイラパッケージ V.1.01 Release 02 では、ソースファイルと同じ フォルダにあるヘッダファイルは自動的に登録されます。

他のフォルダにある場合は"Include file directories"を使用して、該当するフォルダを登録してください。

IAR ICC740 (740 Family) Standard To	olchain 🔹 🛛 🛛
Configuration : Debug All Loaded Projects C source file Assembly source file	C Assembly Link Librarian CPU Toolchain Option Category: Source Show entries for: Include file directories Add Insert Remove Move up Move gown Options C : -v0 -ml "-O\$(CONFIGDIR)¥" -r -z9 -e -K -h
	() チャンセル

図 23 : [Toolchain]ダイアログボックス : C タブの Category:Source

4.2.2. リストファイルの作成

デフォルトではリストファイルを作成しませんので、必要な場合は図 24の設定を行ってく ださい。

IAR ICC740 (740 Family) Standard To	oolchain 🔹 💽 🔀
Configuration : Debug Configuration : All Loaded Projects Configuration	C Assembly Link Librarian CPU Toolchain Option Category: List Category: List Cate
	Options C : -v0 -ml ~-O\$(CONFIGDIR)¥″ -r ~-L\$(CONFIGDIR)¥″ -q -z9 -e -K -h OK キャンセル

図 24:Cタブの Category:List

4.3. A740 オプションの設定

オプションの設定はAssemblyタブで行います。Assemblyタブの内容は表 8の通りです。

表	8	:	Assembly	タ	ブ
---	---	---	----------	---	---

Category	Show entries for	Item	該当 option
Source	Include file	登録 box	-I
	directories		
	Defines	登録 box	-D
	Undefines		-U
	Output	Output directory	-0
	Debug	Generate debug information	-r
List		Generate list file	-L, -i, -t
Other		Miscellaneous options	-s, -w
		User defined options	

オプションの詳細は、IMA アセンブラ・プログラミング・ガイド (A740_jp.pdf) の 21~ 34 ページを参照してください。740 Assembler, Linker, and Librarian Programming Guide の GUI 画面は IAR Systems 社の Embedded Workbench のものです。 A740のデフォルトのオプション設定は表 9のとおりです。

オプション	説明		
N	16 ビットアドレッシングを設定します。		
-un	Large モデル選択時に登録します		
-v0	MUL/DIV を持つ MCU を選択します。(一般的な MCU です。)		
-r	デバッグ情報を出力します。		
-O\$(CONFIGDIR)¥	オブジェクトファイル名を設定します。		

表 9: A740 のデフォルトオプション

4.3.1. リストファイルの作成

デフォルトではリストファイルを作成しませんので、必要な場合は図 25の設定を行ってく ださい。

AR ICC740 (740 Family) Standard T	oolchain ? 👔
Configuration : Debug All Loaded Projects C source file Assembly source file	C Assembly Link Librarian CPU Toolchain Option Category : List Category
	Options Assembly : -v0 -uN -r "-O\$(CONFIGDIR)¥" "-L\$(CONFIGDIR)¥"

図 25 : Assembly タブの Category : List

4.4. XLINKオプションの設定

オプションの設定はlinkタブで行います。linkタブの内容は表 10の通りです。

Category	Show entries for	Items	該当 option
Memory			-Z
		Use segment definition	-f
		subcommand file	
Input	Library files	登録 box	無し
	Object files	登録 box	無し
	Defines	登録 box	-D
Output		Type for output file	-F
		Fill unused code memory	-H
		Output file path	-0
List		Generate list file	-L, -x
Other		Miscellaneous options	-z, -w
		User defined options	その他
Subcommand		Use external subcommand file	-f
file			

表 10 : link タブ

オプションの詳細は、IMA アセンブラ・プログラミング・ガイド(A740_jp.pdf)の 141 ~169 ページを参照してください。IMA アセンブラ・プログラミング・ガイドの GUI 画面 は IAR Systems 社の Embedded Workbench のものです。

XLINKのデフォルトのオプション設定は表 11のとおりです。

オプション	説明
-C cl7400l.r31	ライブラリを指定します。
-c740	740 ファミリを指定します。
-Fmotolora	モトローラ形式ファイルを指定します。
	通常は IEEE695 形式ファイルも作成します。
-o\$(CONFIGDIR)¥\$(P	オブジェクトファイル名を設定します。
ROJECTNAME).mot	
-1\$(CONFIGDIR)¥\$(PR	マップファイルを出力します。
OJECTNAME).map	
-xmnos	マップファイルに相互参照リストを出力します。
-Z	セグメントの配置を指定します

表 11: XLINK のデフォルトオプション

V.1.01 Release 02 では、XLINKのオプションのGUI化を行い、lnk740.xclを使用しない形 式にしました(図 26)。

AR ICC740 (740 Family) Standard To Configuration :	olchain C A	sembly Link Librar	rian CPU To	olchain Option
	Ca	tegory: Memory		
	Type	Address	Segment 🗡	Add
E⊸wigv proj_ ⊕ ∽ige C source file ⊕ ∽ige Assembly source file	BIT ZPAGE ZPAGE	00000040 00000041-000000FF	BITVARS ZPAGE,C EXPB_ST/	Insert
	ZPAGE NPAGE NPAGE	00000100 00000100-0000083F	INT_EXPR CSTACK+ NPAGE,C	Remove
	NPAGE CODE	00001080-0000FEFF	RF_STACI RCODE,Z_	Move up
	< C			Move <u>d</u> own
		egment definition subco	ommand file :	<u>M</u> odify
	Options -C "cl74 ¥\$(PRO ¥\$(PRO	Link : IOOtr31″ -c740 -Fmotor JECTNAME).mot″ -l ″\$(JECTNAME).map″ -xmn	′ola −o ″\$(CONF CONFIGDIR) os −Z(BIT)BITV/	IGDIR)
l			ОК	キャンセル

図 26 : Link タブ

lnk740.xcl を使用する場合は"Use segment definition subcommand file:"にチェックを入 れてください。

5. プロジェクトの開発

5.1. ソースファイルの作成および登録

ソースファイルは[ファイル]メニューの[新規作成]によりエディタが起動されます。 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。プロジェクトディレ クトリに置いただけでは登録されません。[プロジェクト]メニューの[ファイルの追加...]で 登録してください。

5.2. セグメントの追加

アセンブリ言語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 Link タブの Category:Memory に設定する必要があります。 以下に例を示します(図 27)。

< complexed >	IAR ICC740 (740 Family) Standard Toolchain
< sample.so1 >	Configuration : C Assembly Link Librarian CPU Toolchain Option
 RSEG RAM_DATA BLKB 10 	Debue Category: Memory Image: Contract of the second of the s
	OK

アセンブリ言語ソースファイル

Insert segment		2	IAR ICC740 (740 Family) Standard T	oolchain
Type: Start_address: I⊽ End address: I⊽ Stack_ IT Address: Sige: Segment:	NPAGE Image: Constraint of the second of the s		Configuration : Debue	C Assembly Link Librarian OPU Toolchain Option Category: Memory Type Address Bervnent BIT 00000040 BITVARS ZPAGE 000000041-000000FF ZPAGEO. ZPAGE 00000100-000008F RAMROT NPAGE 00000100-000008F RAMROT NOVE down
		OK Cancel		Options Link :

[Insert Segment]ダイアログボックス Category:Memory<After>

Category:Memory<Before>

図 27: Category:Memory の設定

5.3. セグメントの変更

セグメントのアドレスまたはサイズを変更する場合は、Link タブの Category:Memory で 該当セグメントをダブルクリックして変更してください。

[Modify Segment]ダイアログボックス(図 28)にて該当項目を変更してください。

Modify segment	28
<u>T</u> ype :	CODE
Start <u>a</u> ddress :	0×00000080
₩ End address :	0x0000FEFF
∏ Stac <u>k</u>	
F Address :	0×0000000
Size :	0×0000000
<u>S</u> egment :	RCODEZ_CDATAN_CDATAC_ICALLC_RECFNCSTRCCSTRCODEC
	OK Cancel

図 28 : [Modify Segment]ダイアログボックス

6. 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]、[すべてをビルド]、または[ファイルのコンパイル]ボタンを クリックしてください。

Debug	DefaultSession	•
-------	----------------	---

図 29: [ビルド]、[すべてをビルド]、[ファイルのコンパイル]ボタン

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。

6.1. 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。

行番号を表示しているエラーでは、その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。

6.2. 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 id, 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_i1;

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 ($フ_{P}$ イル) この場合は、シンボル名の確認を行ってください。

6.3. リンカXLINKの注意事項

● 割り込み処理に関する注意事項

割り込み処理実行中に呼び出す関数が、割り込み処理関数以外の関数からも呼び出さ れる場合、リンク時に出力されるべき以下のウォーニングが出力されません。

Warning[w16]: Function "name" in module (file) is called from two function trees (with roots name1 and name2)

ICC740 では、関数のローカル変数(関数引数および auto 変数)を静的に配置しています。 そのため、ローカル変数を持つ関数を割り込み処理関数と、割り込み処理以外の関数 の両方で使用するとローカル変数を破壊することがあります。 割り込み処理関数と割り込み処理以外の関数で同じ関数を使用しないでください。

```
発生例:
```

```
_____
void func2( int );
interrupt[2] void intr_1(void)
{
   func2(2);
             /* func10で、func20を実行中に割り込みが発生 */
              /* すると、割り込みから復帰した func2()の
                                                */
              /* ローカル変数が破壊される
                                                 */
}
void func1( void )
{
   func2(1);
}
void main( void )
{
   func1();
}
```

● 未定義関数に関する注意事項

未定義関数を呼び出した場合、リンク時に出力されるべき以下のエラーが出力されま せん。

Error[e46:] Undefined external "external" referred in module(file) 未定義関数を呼び出していないかどうか確認してください。 呼び出している場合はその関数を定義してください。

発生例: void func3(int); void main(void) { func30; /* 未定義関数 */ }

7. 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、Debug フォルダにアブソリュートファイルが生成されます。

デバッグを行う場合、セッションから該当するデバッガを選択してください(図 30)。

🗇 🎬 🛗 👗 Debug 💽	DefaultSession 🗾	2. 🖗
	Session740_Simulator	·
	Session740_E8_SYSTEM	
	DefaultSession	
	Session740_PC4701_Emulator	

図 30:セッション

該当するデバッガがインストールされている場合、デバッガの初期化ダイアログがオープ ンしますので、各種値を設定して[OK]ボタンをクリックしてください。 740 シミュレータの場合は、図 31のダイアログとなります。

Init (740 Simulator)	
MCU デバッグ情報 起動スクリプト	
MCU:	参照
- 時間計測リソース 動作周波数: 3 MHz / 2	
	 用(<u>A</u>) ヘルプ Dダイアログボックスを表示しない

図 31:740 シミュレータの Init ダイアログ

MUC ファイルは[参照]ボタンにより選択してください。ターゲットマイコンに該当する MCU ファイルが見つからない場合は、「740 シミュレータデバッガ V.1.2 ユーザズマニュ アル」の「4.3 MCU ファイルの作成」を参照してください。

セッションの変更が完了しましたら、プログラムのダウンロードを行います。 [デバッグ]メニューの[ダウンロード]により行ってください。

8. ヘキサファイルの作成

V.1.01 Release 02 では、通常、アブソリュートロードモジュールとして IEEE695 形式フ ァイルとモトローラ形式の 2 つのファイルを作成します。

アブソリュートロードモジュールの変更は[Toolchain]ダイアログボックスのLinkタブの Outputカテゴリ(図 32)で選択できます。

AR ICC740 (740 Family) Standard To Configuration :	C Assembly Link Librarian CPU Toolchain Option
Debug Image: All Loaded Projects Image: Comparing the source file Image: Comparing the source file Image: Comparing the source file	Category : Output Type of output file : Stype and IEEE695 Fill unused code memory : IEEE695 Output file path : Stype and IEEE695 \$(CONFIGDIR)*\$(PROJECTNAME).mot Modify Options Link : Options Link :
	-C "cl7400lr31" -c740 -Fmotorola -o "\$(CONFIGDIR) ¥\$(PROJECTNAME).mot" -I "\$(CONFIGDIR) ¥\$(PROJECTNAME).map" -xmnos -Z(BIT)BITVARS=00000200 ☑
	OK キャンセル

図 32 : [Toolchain]ダイアログボックスの Link タブの Category:Output

表	12	:	Type	of	output	file	の内容
---	----	---	------	----	--------	------	-----

Type of output file	ファイル
IEEE695	IEEE695 形式ファイルを作成
Hex and IEEE695	IEEE695 形式ファイルとインテル Hex 形式ファイルを作成
Stype and IEEE695	IEEE695 形式ファイルとモトローラ形式ファイルを作成

9. リビジョンアップ時の注意

9.1. V.1.01 Release 01 からのリビジョンアップ

リビジョンアップアップ時は、V.1.01 Release 01 をアンインストールしてください。 V.1.01 Release 01 のプロジェクトタイプは使用できません。

Application
Application(Tiny)
Application for E8
Application for E8 (Tiny)
V.1.01 Release 01 のプロジェクトタイプ

9.2. V.1.01 Release 01 で作成したプロジェクトのコンバート

V.1.01 Release 01 のプロジェクトをV.1.01 Release 02 で使用する場合は以下のダイアログ ボックスを表示します(図 33)。メッセージに従って変更してください。

Change Toolchain Version Summary	
<u>S</u> ummary :	
Project name : old_proj IAR ICC740 Toolchain (740 Family) was upgraded 1.01.01 -> 1.02.00. The options in the xcl file is not converted. Therefore, the options might conflict. In this case, the options error occurs if the Build is invoked. Please delete the options in the xcl file of the cause of the error. For example: -c740 -C cl74001r31 -C cl7400tr31	
Generate Upgrade.txt as a summary file in the project directory	<u>×</u>

図 33:ツールチェインの変更メッセージ

V.1.01 Release 02 では、マイコン指定(-c740)、ライブラリ指定(-C cl7400l.r31 等)はデフォルトで設定しますので、lnk740.cxl ファイルから削除してください。

lnk740.xclの指定はLinkタブのOtherカテゴリに設定してあります(図 34)。

Configuration :	C Assembly Link Librarian CPU Toolchain Option
Debug All Loaded Projects C source file	Category: Other Miscellaneous options : Segment overlap warnings Disable warnings User defined options : -f "\$(PRO JDIR)¥Ink740.xcl"
	Options Link : -C "cl7400lr31" -c740 -Fieee695 -yImba -o "\$(CONFIGDIR) ¥\$(PROJECTNAME).695" -I "\$(CONFIGDIR) ¥\$(PROJECTNAME).map" -x -f "\$(PROJDIR)¥Ink740.xcl"

図 34: ツールチェイン変更後の lnk740.xcl ファイルの設定箇所

なお、ツールチェイン変更後のLinkタブのMemoryカテゴリは以下の通りとなります(図 35)。



図 35:ツールチェイン変更後のメモリカテゴリ

セグメント配置は lnk740.xcl ファイルに設定があるため、この状態で使用してください。 Memoryカテゴリでセグメント配置を行う場合は 図 35のlnk740.xclファイルの設定を削除 してください。

10. cstartup.s31とlnk740.xclファイルの編集

新規プロジェクト作成で[Empty Application]を選択した場合等、ICC740 に付属の cstartup.s31 と lnk740.xcl 使用する場合は、これらを編集する必要があります。

10.1. cstartup.s31 の編集

cstartup.s31では、表 13の項目を必要に応じて編集します。

表 13: 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

10.1.1. 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

H, 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。

なお、CPU モードレジスタの他のビットはターゲットマイコンに合わせて設定してください。

10.1.2. 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 0FFFEH - 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 1nk740. xcl ファイルで行います。

割り込みベクトル領域に直接、各割り込みのベクトルを記述する方法もあります。

この場合は全てのベクトルを登録してください。使用しない割り込みはリセットベクトル と同じアドレスにする等の設定を行い、不定アドレスへジャンプしないよう対策を行って ください。

?CSTARTUP INTVEC:				
WORD init C	;	+0x00	:	BRK
WORD init C	;	+0x02	:	AD SIO3T
WORD init C	;	+0x04	:	INT4 CNTR2
WORD init C	;	+0x06	:	INT3
WORD Int2	;	+0x08	:	INT2
WORD init C	;	+0x0a	:	SIO2 TimerZ
WORD init C	;	+0x0c	:	CNTRI SIO3R
WORD init C	;	+0x0e	:	CNTR0
WORD init C	;	+0x10	:	Timer2
WORD Timer1	;	+0x12	:	Timer1
WORD init C	;	+0x14	:	TimerY
WORD init C	;	+0x16	:	TimerX
WORD init C	;	+0x18	:	SIO1T
WORD init C	;	+0x1a	:	SIO1R
WORD init C	;	+0x1c	:	INT1
WORD Int0	;	+0x1e	:	INTO TimerZ
?CSTARTUP RESETVEC:				—
WORD init C	;	+0x20	:	reset
ENDMOD init C	-			

10.2. 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	14	:	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

10.2.1. 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。

740 ファミリ C コンパイラパッケージの標準では、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。

10.2.2. 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(Oh~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 グループ等 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 グループでは、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。

10.2.3. 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。

10.2.4. 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-FEFF

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 FEFFh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

10.2.5. 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-FFFD

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。

10.2.6. ライブラリの削除

V.1.01 Release 02 では、CPUの選択に応じてライブラリを指定します。 lnk740.xclのライブラリ指定箇所は削除してください。

-! -C cl7400l -!

10.2.7. セグメントの追加

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。

MEMO

740ファミリ用 Cコンパイラパッケージ V.1.01 ユーザーズマニュアル

発行年月日 2007年6月16日 Rev.1.01

発行 株式会社 ルネサス テクノロジ 営業企画統括部 〒100-0004 東京都千代田区大手町2-6-2

編集 株式会社 ルネサス ソリューションズ ツール開発部

© 2007. Renesas Technology Corp. and Renesas Solutions Corp., All rights reserved. Printed in Japan.

740 ファミリ用 C コンパイラパッケージ V.1.01 ユーザーズマニュアル

