

マイクロコントローラ技術情報

技術通知 78K0 アセンブラ・パッケージ RA78K0 使用制限事項の件		発行番号	ZBG-CD-07-0064号	1 / 2
		発行日	2007年 9月18日	
		発行部門	NEC エレクトロニクス株式会社 マイクロコンピュータ事業本部 汎用マイコンシステム事業部 開発ツールソリューショングループ	
文書分類	○ 使用制限事項	バージョン・アップ	ドキュメント誤記訂正 (正誤表)	その他
関連資料	RA78K0 アセンブラ・パッケージ Ver.3.80 構造化アセンブリ言語編		資料番号: U17197JJ1V0 (第1版)	
	RA78K0 アセンブラ・パッケージ Ver.3.80 言語編		資料番号: U17198JJ1V0 (第1版)	
	RA78K0 アセンブラ・パッケージ Ver.3.80 操作編		資料番号: U17199JJ1V0 (第1版)	
	PM+ Ver.6.30		資料番号: U18416JJ1V0 (第1版)	
	78K0アセンブラ・パッケージ RA78K0 V4.00 使用上の留意点		資料番号: ZUD-CD-07-0102	

1. 対象製品

RA78K0 V4.00

2. 新たな制限事項

今回新たに制限事項 No.43 を追加しました。詳細は、別紙1を参照してください。

・No.43 ファイル名のプライマリ・ネームが9文字以上で E3410 エラーとなる制限

3. 回避策

今回追加した制限事項の回避策です。詳細は、別紙1を参照してください。

・No.43 コンパイラでアセンブラ・ソース・モジュール・ファイルの出力を無効にしてください。

4. 改善計画

No.43 は、下記日程で改善することを予定しています。

RA78K0 V4.01 2007年10月リリース予定

※ 本日程については予告なく変更する場合がございますので、改善品のリリース日程については、別途、弊社営業までお問い合わせください。

5. 制限事項一覧

制限事項の履歴とその詳細情報を使用制限事項一覧として別紙1に記載します。

6. 発行文書履歴

78K0 アセンブラ・パッケージ RA78K0 使用制限事項一覧 発行文書履歴

文書番号	発行日	記事
SBG-TT-0004	2001.10.12	統合版初版
SBG-DT-03-0308	2003.12.17	新規制限事項追加(No.28 ~ No.34) 修正バージョン記述間違いの修正(No.8)
SBG-DT-04-0111	2004.3.12	新規制限事項追加(No.35 ~ No.37)
ZBG-CD-07-0039	2007.6.28	新規制限事項追加(No.38 ~ No.42)
ZBG-CD-07-0064	2007.9.18	新規制限事項追加(No.43)

以上

RA78K0 の使用制限事項一覧

1. 製品履歴

No.	仕様変更・追加／不具合事項	バージョン					
		V3.50	V3.51	V3.60	V3.70	V3.80	V4.00
1	構造化アセンブリ言語記述で範囲が交差する制御文がエラーとなる制限	×	×	×	×	×	×
2	構造化アセンブラで SFR のリード/ライトのチェックを行わない制限	○	○	○	○	○	○
3	構造化アセンブリ言語でのレジスタの絶対名称の正しい記述がエラーになる制限	○	○	○	○	○	○
4	構造化アセンブリ言語でのレジスタの絶対名称の不正な記述がエラーにならない制限	○	○	○	○	○	○
5	saddr.bit の値を持つビット・シンボルを EQU 定義する際、saddr の部分に最適化の影響を受けるレーベルが記述されるとエラー等が不正になる制限	×	×	×	×	×	×
6	アセンブラ・ソースの 1 行に記述可能な文字数は 256 文字以内である制限	○	○	○	○	○	○
7	漢字のインクルード・ファイル名が認識されない制限	○	○	○	○	○	○
8	インクルード・ファイルの最後に“^Z”が無い場合、正常動作しない制限(Windows 版のみ)	○	○	○	○	○	○
9	マクロ疑似命令 IRP をネストさせるとコンカティネート(&)が結合されない。	×	×	×	×	×	×
10	ライブラリアンでコンパイラ付属のライブラリ・ファイルを操作できない制限	○	○	○	○	○	○
11	プロジェクト・マネージャから構造化アセンブラを起動する場合のインクルード・ファイルの依存関係の制限	×	×	×	×	×	×
12	プロジェクト・マネージャから構造化アセンブラを起動するとアセンブラも必ず起動する制限	○	○	○	○	○	○
13	プロジェクト・マネージャの[オプション]メニューの「ディバグ」がチェックされていても構造化アセンブラにディバグ情報出力オプションが設定されない制限	○	○	○	○	○	○
14	プロジェクト・マネージャから構造化アセンブラのインクルード・パス指定オプション-I を指定しても有効にならない制限	○	○	○	○	○	○
15	プロジェクト・マネージャからアセンブラを起動する場合のインクルード・ファイルの依存関係の制限	×	×	×	×	×	×
16	プロジェクト・マネージャの[オプション]メニューの「ディバグ」がチェックされていてもアセンブラにディバグ情報出力オプションが設定されない制限	○	○	○	○	○	○
17	プロジェクト・マネージャからリンカのパラメータ・ファイル指定オプション(-F)が指定できない制限	○	○	○	○	○	○
18	プロジェクト・マネージャからアセンブラ等のヘルプ指定オプション(-?)が指定できない制限	○	○	○	○	○	○
19	コマンドラインが 127 文字を超えた分は無視される制限	○	○	○	○	○	○

×:該当する

○:該当しない

-:対象外

No.	仕様変更・追加／不具合事項	バージョン					
		V3.50	V3.51	V3.60	V3.70	V3.80	V4.00
20	プロジェクト・マネージャを使用する場合ソース・ファイルはプロジェクト・ディレクトリにあるもののみ指定可能である制限	○	○	○	○	○	○
21	dosinst.bat でインストールした場合 uPD7800xx 以外のデバイスを品種指定できない不具合	—	—	—	—	—	—
22	setup31.exe でアセンブラ・パッケージのみインストールした場合、アセンブラ等のコマンドが起動できない不具合	—	—	—	—	—	—
23	構造化アセンブリ言語ソースから作成されたオブジェクト中のファイル名が不正となる不具合	○	○	○	○	○	○
24	アセンブラにオブジェクト・モジュール・ファイル非出力オプション-NO を指定すると Windows の不正処理エラーが出力される制限	○	○	○	○	○	○
25	プロジェクト・マネージャから構造化アセンブラ、アセンブラ等へ出力ファイル名指定オプション-O を指定するとエラーになる制限	○	○	○	○	○	○
26	プロジェクト・マネージャから ID78K0-NS(V2.xx)、ID78K0(V2.xx)、SM78K0(V2.xx)を単体で起動して作成したプロジェクトファイルを開くとエラーになる制限	×	×	×	×	×	×
27	C ソースとアセンブラ・ソースが混在している場合アセンブラ記述の EXTRN が最初に出現すると、リンカ、オブジェクト・コンバータでエラーとなる不具合	○	○	○	○	○	○
28	構造化アセンブラをコマンド・プロンプトから起動した時、デバイス・ファイルが無いとコマンド・プロンプトがエラーを出力する制限	×	○	○	○	○	○
29	アセンブラ・ソースのマクロ定義内の全角文字のコメントが、アセンブル・リスト・ファイル中のマクロ展開部分で表示されない制限	×	○	○	○	○	○
30	アセンブラの-D オプションでシンボルに数値を指定しても、マクロは真偽値としてしか利用できない制限	×	○	○	○	○	○
31	アセンブラ・ソース中の最終セグメントにある BR 疑似命令を最適化した後にコード数の計算を誤り、出力されるコードが不正になる場合がある制限	×	○	○	○	○	○
32	ディレクティブで”MERGE AT”指定したセグメントの配置優先順位が最優先されない制限	×	○	○	○	○	○
33	PUBLIC/EXTRN シンボルに、コンパイラから出力される補助シンボルテーブルが存在するとディバッガ、シミュレータでエラーになる制限	×	○	○	○	○	○
34	セグメント名が 8 文字の時、リンカのエラーメッセージ表示が不正になる場合がある制限	×	○	○	○	○	○
35	サイズが 0 のセグメントの配置が不正になる制限	×	×	×	○	○	○
36	アセンブラのネーム定義がオブジェクト・コンバータで A106 エラーになる制限	×	×	×	○	○	○
37	ソース・ファイル名が 14 文字より長い場合にリンカで A402 エラーになる制限	×	×	×	×	×	○
38	RA78K0 で -Dsymbol オプションを指定すると F4106 エラーとなる制限	×	×	×	×	×	○

×:該当する
○:該当しない
—:対象外

No.	仕様変更・追加／不具合事項	バージョン					
		V3.50	V3.51	V3.60	V3.70	V3.80	V4.00
39	Cソースから外部参照した関数名の定義側が 構造化アセンブラ・ソース、あるいはアセンブラ・ソースの PUBLIC シンボルである場合に、リスト・コンバータで F6105 エラーになる制限	×	×	×	×	×	○
40	ORG 疑似命令のセグメント名を省略した場合に、リスト・コンバータで F6106 エラーになる制限	×	×	×	×	×	○
41	複数の Cソースを登録しアセンブラ・ソース出力オプションを指定した場合に、アブソリュート・アセンブル・リストが最初の 1 つ分しか出力しない制限	×	×	×	×	×	○
42	PM+のオブジェクト・コンバータ・オプションの設定の充てんのスタート・アドレスおよびサイズ指定にて、最終バンク領域のアドレスを指定できない制限	○	○	○	○	×	○
43	ファイル名のプライマリ・ネームが 9 文字以上で E3410 エラーとなる制限	—	—	—	—	—	×

×:該当する

○:該当しない

—:対象外

2. 使用制限事項の詳細

No. 1 構造化アセンブリ言語記述で範囲が交差する制御文がエラーとなる制限

【内 容】制御文を`#ifdef~#endif`で分割または交差するように囲んだ場合、`#ifdef`が真の時に制御文がエラーになります。

(例)

```
switch(mode)
#ifdef stsw
case 1:
break
#endif
default:
break
ends
```

← `#ifdef~#else/#endif` の範囲
← `case~次の case/default/ends` までの範囲

【回避策】ネスト(入れ子)の場合には、正しく処理されます。制御文の範囲を交差させないようにソースを書き直してください。

(例)

```
#ifdef stsw
switch(mode)
case 1:
break
default:
break
ends
#else
switch(mode)
default:
break
ends
#endif
```

← `#ifdef~#else/#endif` の範囲
← `case~次の case/default/ends` までの範囲

【改善策】制限事項とします。

No. 2 構造化アセンブラで SFR のリード/ライトのチェックを行わない制限

【内 容】レジスタ指定に SFR を記述した場合に、SFR のリード/ライトのチェックを行っていないため、アセンブル時にエラーとなることがあります。

【回避策】SFR のリード/ライトを確認のうえ、ご使用ください。

【改善策】V3.30 で修正済みです。

No. 3 構造化アセンブリ言語でのレジスタの絶対名称の正しい記述がエラーになる制限

【内 容】レジスタの絶対名称を記述した場合、正常に処理すべき記述がエラーとなります。

```
AX = PR1    PR1=AX    AX<->PR1
AX = PR2    PR2=AX    AX<->PR2
AX = PR3    PR3=AX    AX<->PR3
```

【回避策】機能名称で記述してください。

```
AX = BC    BC=AX    AX<->BC
AX = DE    DE=AX    AX<->DE
AX = HL    HL=AX    AX<->HL
```

【改善策】V3.30 で修正済みです。

No. 4 構造化アセンブリ言語でのレジスタの絶対名称の不正な記述がエラーにならない制限

【内 容】レジスタの絶対名称を記述した場合、エラーとなるべき記述が正常終了してしまいます。

A=R1

R1=A

A<->R1

【回避策】ありません。

【改善策】V3.30で修正済みです。

No. 5 saddr.bit の値を持つビット・シンボルを EQU 定義する際、saddr の部分に最適化の影響を受けるレーベルが記述されるとエラー等が不正になる制限

【内 容】saddr.bit の値を持つビット・シンボルを EQU 定義する際に、saddr の部分に最適化の影響を受けるレーベルが記述されると、不正な処理を行います。

【現 象】以下のケースで、不正な処理をします。

- saddr.bit が 0FD20H の時、レーベルがパス 1 で領域外、パス 2 で領域内となる場合、EQU 定義行に対し、パス 1 ではエラーを出力しますが、パス 2 では出力されず、不正なオブジェクトを生成します。
- saddr が 0FF1FH の時、レーベルがパス 1 で領域内、パス 2 で領域外となる場合、EQU 定義行に対し、パス 1 ではエラーになりませんが、パス 2 ではエラーを出力します。この EQU シンボルを参照した後に定義するレーベルに対し、以下のアセンブラエラーが出力されます。

「F410 Phase error」

また、このレーベルを参照するとオブジェクトが不正になります。

【回避策】ありません。

【改善策】制限事項とします。

No. 6 アセンブラ・ソースの 1 行に記述可能な文字数は 256 文字以内である制限

【内 容】アセンブラ・ソースの記述で 1 行が 256 文字を超えた場合、アセンブラが以下のワーニングを出力し、256 文字を超えた分を次の行としてアセンブルしてしまいます。尚、2 バイト文字は 2 文字分としてカウントされます。

「 W701 Too long source line 」

【回避策】1 行を 256 文字以内(復帰・改行文字を含む)で記述してください。

【改善策】V3.50で修正済みです。

No. 7 漢字のインクルード・ファイル名が認識されない制限

【内 容】インクルード・ファイル名が漢字で記述された場合、正しく認識されない場合があります。

【回避策】インクルード・ファイル名に漢字を使用しないでください。

【改善策】V3.50で修正済みです。

No. 8 インクルード・ファイルの最後に“^Z”が無い場合、正常動作しない制限(Windows 版のみ)

【内 容】インクルード・ファイルの最後に“^Z”が無い場合、エラーあるいは不正なデバッグ情報を出力することがあります。

【回避策】インクルード・ファイルの最後に“^Z”のみの行を入れてください。

【改善策】V3.50で修正済みです。

No. 9 マクロ疑似命令 IRP をネストさせるとコンカティネート(&)が結合されない。

【内 容】文字列の結合記号“&”を含む IRP をネストさせると、パラメータが置換されず、マクロの展開結果が不正になります。

(例)

```
IRP ZZZ,<1,2,3>
```

```
IRP XXX,<4,5,6>
```

```
LABEL&ZZZ&XXX: ; 展開結果が不正になります
```

```
ENDM
```

```
ENDM
```

【回避策】IRP をネストさせた場合は、“&”を使用しないでください。

【改善策】制限事項とします。

No. 10 ライブラリアンでコンパイラ付属のライブラリ・ファイルを操作できない制限

【内 容】ライブラリアンでコンパイラ付属のライブラリ・ファイルを操作すると、以下のエラーが出力され、操作できません。

```
「F104 Illegal library file – different target chip」
```

【回避策】ありません。uPD78014 用のデバイス・ファイルがあれば操作可能になります。

【改善策】V3.50 で修正済みです。

No. 11 プロジェクト・マネージャから構造化アセンブラを起動する場合のインクルード・ファイルの依存関係の制限

【内 容】プロジェクト・マネージャでのメイク・ファイル作成時において、構造化アセンブラ言語で記述されたインクルード・ファイルの依存関係のチェックは、コメントと文字列削除処理のみ行い、#ifdef などの条件は無視します。

【現 象】下記の例において、#ifdef、#else、#endif 行は無視されます。そのため#include に指定されたファイル名は参照の有無にかかわらず実際にファイルが存在するかのチェックが行われます。#include に指定されたファイルが存在しないとビルド時にエラーになります。

(例)

```
#ifdef SYM
```

```
#include “func1.inc”
```

```
#else
```

```
#include “func2.inc”
```

```
#endif
```

【回避策】ありません。

【改善策】制限事項とします。

No. 12 プロジェクト・マネージャから構造化アセンブラを起動すると、アセンブラも必ず起動する制限

【内 容】プロジェクト・マネージャからは、構造化アセンブラだけを起動することができません。アセンブラも必ず起動されます。

【回避策】ありません。

【改善策】V3.50 で修正済みです。

- No. 13 プロジェクト・マネージャの[オプション]メニューの「ディバグ」がチェックされていても構造化アセンブラにディバグ情報出力オプションが設定されない制限
- 【内 容】プロジェクト・マネージャの[オプション]メニューの「ディバグ」のチェックが、構造化アセンブラのディバグ情報出力オプション指定の有効/無効と連動していません。
- 【回避策】ディバグ情報オプションを有効にする場合は、プロジェクト・マネージャの[構造化アセンブラオプションの設定]メニューで-GS オプションを指定してください。
- 【改善策】V3.30 で修正済みです。
- No. 14 プロジェクト・マネージャから構造化アセンブラのインクルード・パス指定オプション-Iを指定しても有効にならない制限
- 【内 容】プロジェクト・マネージャから構造化アセンブラのインクルード・パス指定オプション-I を指定すると、アセンブラのオプションと解釈してしまう場合があります。
- 【回避策】構造化アセンブリ言語で記述されたインクルード・ファイルは、カレント・ディレクトリに置か、あるいはソース中でインクルードする際にディレクトリ名付きで記述するようにしてください。
- 【改善策】V3.30 で修正済みです。
- No. 15 プロジェクト・マネージャからアセンブラを起動する場合のインクルード・ファイルの依存関係の制限
- 【内 容】プロジェクト・マネージャでのメイク・ファイル作成時において、アセンブラ言語で記述されたインクルード・ファイルの依存関係のチェックは、コメントと文字列削除処理のみ行い、\$if、\$ifなどの条件は無視します。
- 【現 象】下記の例において、\$ifdef、\$else、\$endif 行は無視されます。そのため\$include に指定されたファイル名は参照の有無にかかわらず実際にファイルが存在するかのチェックが行われます。\$include に指定されたファイルが存在しないとビルド時にエラーになります。
- (例)
- ```
$if (SYM)
#include(func1.h)
$else
#include(func2.h)
$endif
```
- 【回避策】ありません。
- 【改善策】制限事項とします。
- No. 16 プロジェクト・マネージャの[オプション]メニューの「ディバグ」がチェックされていてもアセンブラにディバグ情報出力オプションが設定されない制限
- 【内 容】プロジェクト・マネージャの[オプション]メニューの「ディバグ」のチェックがアセンブラのディバグ情報出力オプション指定の有効/無効と連動していません。
- 【回避策】ディバグ情報オプションを有効にする場合は、プロジェクト・マネージャの[アセンブラオプションの設定]メニューで-G、-GA オプションを指定してください。
- 【改善策】V3.30 で修正済みです。
- No. 17 プロジェクト・マネージャからリンカのパラメータ・ファイル指定オプション(-F)が指定できない制限
- 【内 容】プロジェクト・マネージャがメイク・ファイルを自動生成する際に-Fオプションを自動生成してしまうため、-F オプションを指定するとリンカが以下のエラーメッセージを出力します。
- ```
「A107 Too many parameters '-fxxxx.xxx」
```
- 【回避策】ありません。-F オプションを指定しないでください。
- 【改善策】V3.50 で修正済みです。

No. 18 プロジェクト・マネージャからアセンブラ等のヘルプ指定オプション(--o)が指定できない制限

【内 容】 --オプションを指定すると、“Press RETURN to continue ...”というメッセージが出力されますが、PRJTMMAKE ウィンドウではキーを受け付けられないため、ヘルプ表示から戻らなくなります。

【回避策】 --オプションを指定しないでください。ヘルプを参照する場合にはオプションの設定ダイアログの“ヘルプ”ボタンをご使用ください。

【改善策】 V3.30 で修正済みです。

No. 19 コマンドラインが 127 文字を超えた分は無視される制限

【内 容】 アセンブラ・コマンド等のプロジェクト・マネージャから起動されるコマンドは DOS で起動されるコマンドです。よってプロジェクト・マネージャから起動する場合にも、コマンドラインでの起動同様 127 文字までとなり、それ以降の記述は無視されます。

【回避策】 自動的に出力されるコマンド、オプションを考慮の上、オプションを 127 文字以内に設定してください。

【改善策】 V3.30 で修正済みです。

No. 20 プロジェクト・マネージャを使用する場合ソース・ファイルはプロジェクト・ディレクトリにあるもののみ指定可能である制限

【内 容】 ソース・ファイルはプロジェクト・ディレクトリにあるものしか登録できません。

【回避策】 ソース・ファイルはプロジェクト・ディレクトリに置いてください。

【改善策】 V3.30 で修正済みです。

No. 21 dosinst.bat でインストールした場合 uPD7800xx 以外のデバイスを品種指定できない不具合

【内 容】 dosinst.bat でインストールした場合、ra78k0.is2 がインストールされないため uPD7800xx 以外のデバイスを品種指定するとエラーになります。

【回避策】 以下のいずれかの方法で回避してください。

- Windows のインストーラでインストールする方法

Windows のインストーラ setup31.exe または setup95.exe を起動してインストールしてください。

- DOS 上で EXPAND コマンドを起動する方法

dosinst.bat でインストールを行った後に、次のように実行してください。

a:¥をフロッピー・ディスク・ドライブ、c:¥をインストール先ドライブとします。

```
a:¥expand ra78k0.i2_ c:¥necools¥bin¥ra78k0.is2 > nul
```

【改善策】 V3.30 で修正済みです。

No. 22 setup31.exe でアセンブラ・パッケージのみインストールした場合、アセンブラ等のコマンドが起動できない不具合

【内 容】 setup31.exe でアセンブラ・パッケージのみインストールした場合、windpmi.386 がインストールされないため、Windows3.1 上でアセンブラ、リンカ、オブジェクト・コンバータ等のコマンドが起動できません。

【回避策】 プロジェクト・マネージャもインストールしてください。

【改善策】 V3.30 で修正済みです。

- No. 23 構造化アセンブリ言語ソースから作成されたオブジェクト中のファイル名が不正となる不具合
- 【内 容】 構造化アセンブラが生成しアセンブラ・ソース中に出力するデバッグ情報用のファイル名は、ディレクトリを含んでいるために、ディレクトリ名付きを想定していないアセンブラは、不正なファイル名を生成しオブジェクトに出力してしまいます。
- 【回避策】 ありません。
- 【改善策】 V3.30 で修正済みです。
- No. 24 アセンブラにオブジェクト・モジュール・ファイル非出力オプション-NO を指定すると Windows の不正処理エラーが出力される制限
- 【内 容】 デバッグ情報を生成する処理において-NO オプションを指定すると、Windows が以下のメッセージを出力し、異常終了します。
- 「このプログラムは不正な処理を行ったので強制終了されます…」
- 【回避策】 ありません。-NO オプションを指定しないでください。
- 【改善策】 V3.50 で修正済みです。
- No. 25 プロジェクト・マネージャから構造化アセンブラ、アセンブラ等へ出力ファイル名指定オプション-O を指定するとエラーになる制限
- 【内 容】 出力ファイル名指定オプション-O でファイル名を変更した場合、変更されたファイル名を入力ファイルとして起動するコマンドに、ファイル名の変更が伝わらないためエラーとなります。
- 【回避策】 ありません。-O オプションを使用しないでください。
- 【改善策】 V3.50 で修正済みです。
- No. 26 プロジェクト・マネージャから ID78K0-NS(V2.xx)、ID78K0(V2.xx)、SM78K0(V2.xx)を単体で起動して作成したプロジェクト・ファイルを開くとエラーになる制限
- 【内 容】 ID78K0-NS(V2.xx)、ID78K0(V2.xx)、SM78K0(V2.xx)を単体で起動して作成したプロジェクト・ファイル(プロジェクト・マネージャ以外のツールが作成したプロジェクト・ファイル)を、プロジェクト・マネージャで開こうとすると、以下のエラーになります。
- 「F150:PRJ ファイルの読み込みに失敗しました。」
- 【回避策】 ID78K0-NS(V2.xx)、ID78K0(V2.xx)、SM78K0(V2.xx)を単体で起動して作成したプロジェクト・ファイルと同名のファイルを、プロジェクト・マネージャで新規プロジェクト・ファイルとして作成してください。新規作成後、プロジェクト・マネージャの[オプション]メニューから[デバッグの選択]を選択し、使用するデバッグ(ID または SM)を指定してください。
- 【改善策】 制限事項とします。
- No. 27 C ソースとアセンブラ・ソースが混在している場合アセンブラ記述の EXTRN が最初に出現すると、リンカ、オブジェクト・コンバータでエラーとなる不具合
- 【内 容】 C ソースとアセンブラ・ソースが混在し、シンボル定義とそれを参照しているモジュールが複数ある時、アセンブラ記述の EXTRN が最初に出現すると、リンカで A402 エラー、またはオブジェクト・コンバータで A106 エラーになる場合があります。
- 【回避策】 C ソースから生成されたモジュールをアセンブラ・ソースから生成されたモジュールよりも先にリンカに指定してリンクしてください。また、プロジェクト・マネージャ上では、C ソースを優先してソース登録してください。ただし、スタート・アップ・ルーチンを使用している場合は、スタート・アップ・ルーチンを最初にリンクしてください。
- 【改善策】 V3.50 で修正済みです。

No. 28 構造化アセンブラをコマンド・プロンプトから起動した時、デバイス・ファイルが無いとコマンド・プロンプトがエラーを出力する制限

【内 容】 構造化アセンブラをコマンド・プロンプトから起動した時、デバイス・ファイルが無いと、コマンド・プロンプトがエラーを出力するか、または反応がなくなってしまいます。

【回避策】 デバイス・ファイルがインストールされているか確認してください。また、構造化アセンブラまたはデバイス・ファイルのインストール・フォルダがデフォルトでない場合、-Y オプションでデバイス・ファイルのパスを指定してください。

【改善策】 V3.51 で修正済みです。

No. 29 アセンブラ・ソースのマクロ定義内の全角文字のコメントが、アSEMBル・リスト・ファイル中のマクロ展開部分で表示されない制限

【内 容】 マクロの定義部分のコメントに全角文字を記述すると、アSEMBル後に作成されるアSEMBル・リスト・ファイル中の、マクロ展開された部分のコメントが、全角文字だけ表示できません。

(例)

sample.asm

NAME sample

MAC1 MACRO

 NOP ; comment1 全角コメント ハンカ
 ENDM

CSEG

SAMPLE1:

 MAC1

 END

sample.prn

Assemble list

ALNO	STNO	ADRS	OBJECT	M I	SOURCE STATEMENT
1	1				NAME SAMPLE
2	2			M	MAC1 MACRO
3	3			M	NOP ; comment1 全角コメント ハンカ
4	4			M	ENDM
5	5				
6	6	----			CSEG
7	7	0000			SAMPLE1:
8	8				MAC1
9	9	0000	00	#1	NOP ; comment1 ハンカ
10	10				
10	11				END

【回避策】 ありません。

【改善策】 V3.51 で修正済みです。

No. 30 アセンブラの-D オプションでシンボルに数値を指定しても、マクロは真偽値としてしか利用できない制限

【内 容】アセンブラ・ソース中で\$set/\$reset/EQU を記述する代わりに、-D オプションでシンボル値を指定しても、-D で指定したシンボル値が有効になりません。

(例1)

「-Dsyms=0 以外」でも(2)が有効になってしまうことがあります。

また、-D 指定時に\$set(syms)/\$reset(syms)の記述があった場合 A109 エラーにならないかもしれませんが、\$set(syms)/\$reset(syms)が有効となり-D オプションは無視されます。

```
$if (syms)
```

```
    ; (1) 「-Dsyms」または「-Dsyms=0 以外」が指定された時
```

```
$else
```

```
    ; (2) 「-Dsyms=0」が指定された時
```

```
$endif
```

(例2)

-D オプションが無視され、F303、F407 エラーになってしまいます。

また、-D 指定時に EQU の記述があった場合、A109 エラーにならないかもしれませんが、-D が無効のため、EQU が有効になっています。

```
$_if ( symb = OFFH )
```

```
    ; (3) 「-Dsymb=OFFH」が指定された時
```

```
$_elseif ( symb = 01H )
```

```
    ; (4) 「-Dsymb」または「-Dsymb=1」が指定された時
```

```
$else
```

```
    ; (5) symb に(3),(4)以外の値が指定された時
```

```
$endif
```

【回避策】-D オプションを使用せず、\$set(syms)/\$reset(syms)/EQU を記述してください。

【改善策】V3.51 で修正済みです。

No. 31 アセンブラ・ソース中の最終セグメントにある BR 疑似命令を最適化した後にコード数の計算を誤り、出力されるコードが不正になる場合がある制限

【内 容】アセンブラ・ソース中の最終セグメントにある BR 疑似命令を最適化した後にコード数の計算を誤り、出力されるコードが不正になる場合があります。

アセンブル・リスト・ファイル(*.prn)、リンク・リスト・ファイル(*.map)の情報は正しいですが、リロケータブル・オブジェクト・ファイル(*.rel)の情報が不正になり、結果、ロード・モジュール・ファイル(*.lmf)、HEX 形式オブジェクト・モジュール・ファイル(*.hex)の情報も不正になります。

【回避策】以下のいずれかの方法で回避してください。

- BR 疑似命令ではなく、命令長の明確な BR 命令を記述してください。
- コードが欠落しているセグメントの最後に、欠落しているサイズ分の NOP を挿入してください。

【改善策】V3.51 で修正済みです。

No. 32 ディレクティブで”MERGE AT”指定したセグメントの配置優先順位が最優先されない制限

【内 容】ディレクティブで”MERGE AT”指定したセグメントは、最優先で、AT で指定されたアドレスへの配置が行われなければなりません、最優先されないために、AT 指定のないセグメントが先に配置されてしまい、F206 または F304 エラーになることがあります。

【回避策】”MERGE AT”指定されたセグメントを配置する予定の領域に、既に配置されてしまったセグメントの配置場所を移動してください。

【改善策】V3.51 で修正済みです。

No. 33 PUBLIC/EXTRN シンボルに、コンパイラから出力される補助シンボルテーブルが存在するとディバグ、シミュレータでエラーになる制限

【内 容】リンカの制限により、PUBLIC/EXTRN シンボルに、コンパイラから出力される補助シンボルテーブルが存在すると、ディバグ、シミュレータが以下のエラーメッセージを出力します。
「b019(A):ファイルのシークに失敗しました。」

【回避策】ありません。

【改善策】V3.51 で修正済みです。

No. 34 セグメント名が 8 文字の時、リンカのエラーメッセージ表示が不正になる場合がある制限

【内 容】セグメント名が 8 文字の時、リンカのエラーメッセージ中のセグメント名表示が不正になる場合があります。

【回避策】ありません。

【改善策】V3.51 で修正済みです。

No. 35 サイズが 0 のセグメントの配置が不正になる制限

【内 容】次の条件をすべて満たす場合に、配置が不正になる場合があります。

(1) サイズが 0 のセグメントが存在する

(2) (1)のセグメントがアドレス指定されていない

(3) (1)以外のセグメントにアドレス指定されたセグメントが存在する

リンク・リストのセグメントのアドレスが、アドレス昇順に出力されていないセグメントが不正に配置されたセグメントになります。

【回避策】不正に配置されているセグメントをディレクティブ・ファイルでアドレス指定してください。

【改善策】V3.70 で修正済みです。

No. 36 アセンブラのネーム定義がオブジェクト・コンバータで A106 エラーになる制限

【内 容】次の条件をすべて満たす場合に、シンボル情報が不正になる場合があり、オブジェクト・コンバータで、A106 エラーになります。また、ディバグでダウンロードすると「Ab019:ファイルの読み込みに失敗しました」とエラーになります。

ただし、不正コードにはなりません。

(1) 構造化アセンブラ・ソース、または、アセンブラ・ソースでネームを定義している

(2) (1)のネームを C ソースから関数名として外部参照をしている

(3) リnkの順番が(2)の C ソースの後に(1)のファイルである

【回避策】Cソースを、構造化アセンブラ・ソースやアセンブラ・ソースより後ろにリンクするようにしてください。

【改善策】V3.70 で修正済みです。

No. 37 ソース・ファイル名が 14 文字より長い場合にリンカで A402 エラーになる制限

【内 容】 次の条件をすべて満たす場合に、リンカにて A402 エラーになります。

- (1) ソース・ファイル名が 14 文字より長い
- (2) アセンブラでデバッグ情報を出力していない
デバッグ情報を出力しない指定は、次のいずれかの場合です。
 - PM または PM plus にて「Release Build」を選択する
 - アセンブラにて“-nga -ng”オプションを指定する

【回避策】 以下のいずれかの方法で回避してください。

- (1) ソース・ファイル名を 14 文字までに変更する。
- (2) PM または PM plus にて「Debug Build」を選択する、またはアセンブラにて“-ga -g”オプションを指定(デフォルトで指定)して、使用する。
アセンブラのデバッグ情報は、生成される HEX ファイルには影響を与えません。

【改善策】 V4.00 で修正済みです。

No. 38 RA78K0 で -Dsymbol オプションを指定すると F4106 エラーとなる制限

【内 容】 CC78K0 からデバッグ情報を付加したアセンブラ・ソース・ファイルに対し、RA78K0 で -Dsymbol オプションの symbol を指定すると、不正なシンボル情報(symbol 名が参照できない)を出力するため、OC78K0 でシンボルテーブルに対する F4106 エラーになります。

【回避策】 CC78K0 で「アセンブラ・ソース・モジュールの出力」オプションを指定する場合には、アセンブラ・オプションの”シンボル定義”オプション指定をしないでください。
アセンブラ・ソース・ファイルに対して、個別アセンブラ・オプションの設定から”シンボル定義”オプション指定をしてください。。

【改善策】 V4.00 で修正済みです。

No. 39 C ソースから外部参照した関数名の定義側が 構造化アセンブラ・ソース、あるいはアセンブラ・ソースの PUBLIC シンボルである場合に、リスト・コンバータにて F6105 エラーになる制限

【内 容】 C ソースから外部参照した関数名の定義側が 構造化アセンブラ・ソース、あるいはアセンブラ・ソースの PUBLIC シンボルである場合に、リスト・コンバータにて F6105 エラーになります。

【回避策】 リスト・コンバータを起動しないでください。

【改善策】 V4.00 で修正済みです。

No. 40 ORG 疑似命令のセグメント名を省略した場合に、リスト・コンバータで F6106 エラーになる制限

【内 容】 ORG 疑似命令のセグメント名を省略した場合に、リスト・コンバータで F6106 エラーになります。

【例】

```
-----アセンブラ・ソース-----  
      ORG 1000H  
      MOV  A, SADR2  
      MOVW AX, SADR3
```

⋮

【回避策】 ORG 疑似命令のセグメント名を記述してください。

【例】

```
-----アセンブラ・ソース-----  
MAIN1 ORG 1000H  
      MOV  A, SADR2  
      MOVW AX, SADR3
```

⋮

【改善策】 V4.00 で修正済みです

No. 41 複数の C ソースを登録しアセンブラ・ソース出力オプションを指定した場合に、アブソリュート・アセンブル・リストが最初の 1 つ分しか出力しない制限

【内 容】 複数の C ソースを登録しアセンブラ・ソース出力オプションを指定した場合に、アブソリュート・アセンブル・リストが最初の 1 つ分しか出力しません。

【回避策】 コマンド・プロンプトから LCNV78K0 を用いて必要なアブソリュート・アセンブル・リストを生成してください

【改善策】 V4.00 で修正済みです。

No. 42 PM+のオブジェクト・コンバータ・オプションの設定の充てんのスタート・アドレス領域およびサイズ指定にて、最終バンク領域のアドレスを指定できない制限

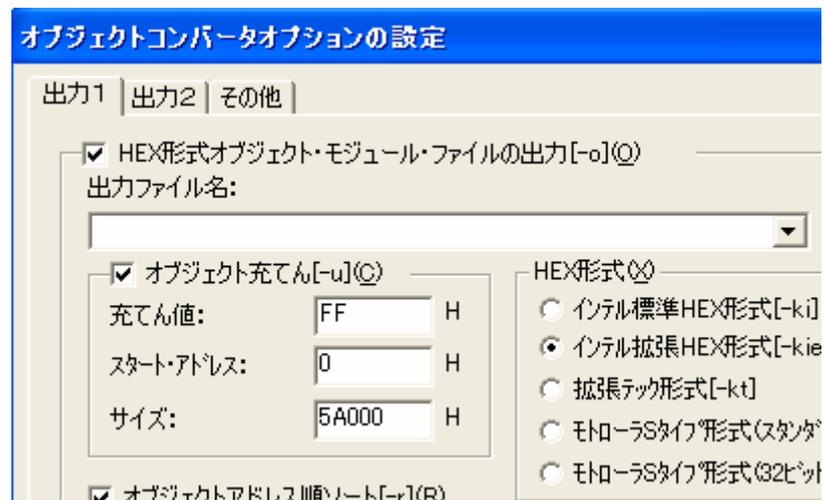
【内 容】 PM +のオブジェクト・コンバータ・オプションの設定で、充てんのアドレス領域のサイズの指定時にサイズ指定が不正になります。

以下の条件の場合です。

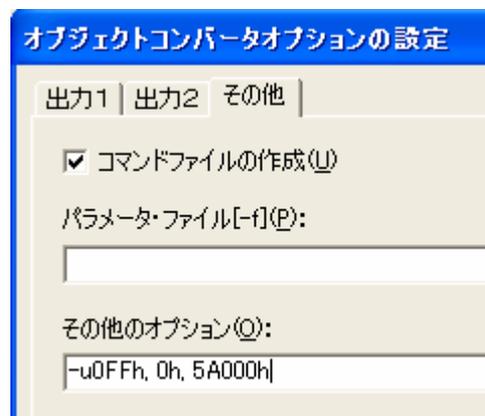
(1)ターゲットがメモリ・バンクのあるデバイス

(2)スタート・アドレス、あるいはアドレス領域のサイズのいずれかが 58000H 以上である

【例】



【回避策】 PM +のオブジェクト・コンバータ・オプションの設定の”その他”タブのその他のオプションのテキストボックスに”-u0FFh, 0h, 5A000h”と設定してください。



【改善策】 V4.00 で修正済みです。

No. 43 ファイル名のプライマリ・ネームが9文字以上でE3410エラーとなる制限

【内 容】 ファイル名のプライマリ・ネームが9文字以上で、先頭から8文字までが同名の場合にE3410エラーになります。

【回避策】 Cソース・ファイルの場合は、コンパイラ CC78K0 のオプションでアセンブラ・ソース・モジュール・ファイルの出力を無効にしてください。

【改善策】 V4.01 で修正する予定です。

3. その他注意事項

特にありません。

以上