

お客様各位

## 資料中の「ラピスセミコンダクタ」等名称の ラピステクノロジー株式会社への変更

2020年10月1日をもって、ラピスセミコンダクタ株式会社のLSI事業部門は、ラピステクノロジー株式会社に分割承継されました。従いまして、本資料中にあります「ラピスセミコンダクタ株式会社」、「ラピスセミ」、「ラピス」といった表記に関しましては、全て「ラピステクノロジー株式会社」に読み替えて適用するものとさせていただきます。なお、会社名、会社商標、ロゴ等以外の製品に関する内容については、変更はありません。以上、ご理解の程よろしくお願いいたします。

2020年10月1日  
ラピステクノロジー株式会社

Dear customer

LAPIS Semiconductor Co., Ltd. ("LAPIS Semiconductor"), on the 1<sup>st</sup> day of October, 2020, implemented the incorporation-type company split (shinsetsu-bunkatsu) in which LAPIS established a new company, LAPIS Technology Co., Ltd. ("LAPIS Technology") and LAPIS Technology succeeded LAPIS Semiconductor's LSI business.

Therefore, all references to "LAPIS Semiconductor Co., Ltd.", "LAPIS Semiconductor" and/or "LAPIS" in this document shall be replaced with "LAPIS Technology Co., Ltd."

Furthermore, there are no changes to the documents relating to our products other than the company name, the company trademark, logo, etc.

Thank you for your understanding.

LAPIS Technology Co., Ltd.  
October 1, 2020

# マップファイルの使い方

---

初版 発行日 2020年2月28日

## ご注意

- 1) 本資料の記載内容は改良などのため予告なく変更することがあります。
- 2) ラピスセミコンダクタは常に品質・信頼性の向上に取り組んでおりますが、半導体製品は種々の要因で故障・誤作動する可能性があります。  
万が一、本製品が故障・誤作動した場合であっても、その影響により人身事故、火災損害等が起こらないようご使用機器でのディレーティング、冗長設計、延焼防止、バックアップ、フェイルセーフ等の安全確保をお願いします。定格を超えたご使用や使用上の注意書が守られていない場合、いかなる責任もラピスセミコンダクタは負うものではありません。
- 3) 本資料に記載されております応用回路例やその定数などの情報につきましては、本製品の標準的な動作や使い方を説明するものです。したがって、量産設計をされる場合には、外部諸条件を考慮していただきますようお願いいたします。
- 4) 本資料に記載されております技術情報は、本製品の代表的動作および応用回路例などを示したものであり、それをもって、当該技術情報に関するラピスセミコンダクタまたは第三者の知的財産権その他の権利を許諾するものではありません。したがって、上記技術情報の使用に起因して第三者の権利にかかわる紛争が発生した場合、ラピスセミコンダクタはその責任を負うものではありません。
- 5) 本製品は、一般的な電子機器(AV機器、OA機器、通信機器、家電製品、アミューズメント機器など)および本資料に明示した用途への使用を意図しています。
- 6) 本資料に掲載されております製品は、耐放射線設計はなされていません。
- 7) 本製品を下記のような特に高い信頼性が要求される機器等に使用される際には、ラピスセミコンダクタへ必ずご連絡の上、承諾を得てください。
  - ・ 輸送機器(車載、船舶、鉄道など)、幹線用通信機器、交通信号機器、防災・防犯装置、安全確保のための装置、医療機器、サーバー、太陽電池、送電システム
- 8) 本製品を極めて高い信頼性を要求される下記のような機器等には、使用しないでください。
  - ・ 航空宇宙機器、原子力制御機器、海底中継機器
- 9) 本資料の記載に従わないために生じたいかなる事故、損害もラピスセミコンダクタはその責任を負うものではありません。
- 10) 本資料に記載されております情報は、正確を期すため慎重に作成したのですが、万が一、当該情報の誤り・誤植に起因する損害がお客様に生じた場合においても、ラピスセミコンダクタはその責任を負うものではありません。
- 11) 本製品のご使用に際しては、RoHS 指令など適用される環境関連法令を遵守の上ご使用ください。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、ラピスセミコンダクタは一切の責任を負いません。本製品の RoHS 適合性などの詳細につきましては、セールス・オフィスまでお問合せください。
- 12) 本製品および本資料に記載の技術を輸出又は国外へ提供する際には、「外国為替及び外国貿易法」、「米国輸出管理規則」など適用される輸出関連法令を遵守し、それらの定めにしたがって必要な手続を行ってください。
- 13) 本資料の一部または全部をラピスセミコンダクタの許可なく、転載・複写することを堅くお断りします。

Copyright 2020 LAPIS Semiconductor Co., Ltd.

## ラピスセミコンダクタ株式会社

〒222-8575 神奈川県横浜市港北区新横浜 2-4-8

<http://www.lapis-semi.com>

## 目次

1. はじめに.....	1
1.1. 関連するドキュメント .....	1
2. 確認方法.....	2
2.1. ROM・RAM 使用量の確認方法 .....	2
2.2. 関数・変数のアドレスの確認方法.....	3
2.3. 関数のサイズの確認方法 .....	4
2.3.1. 通常の場合 (SEGCODE/SEGINTR プラグマを使わない場合) .....	4
2.3.2. SEGCODE/SEGINTR プラグマを使った場合 .....	5
2.4. スタック領域の確認方法 .....	7

## 1. はじめに

本書は、ビルド後の結果を確認するために使用するマップファイルについて、以下の項目に対する対処方法について説明しています。

No.	項目	本ドキュメントの参照箇所
1	ROM・RAM 使用量を確認したい	3.1. ROM・RAM 使用量の確認方法
2	関数・変数のアドレスを確認したい	3.2. 関数・変数のアドレスの確認方法
3	関数のサイズを確認したい	3.3. 関数のサイズの確認方法
4	スタック領域のアドレスやサイズを確認したい	3.4. スタック領域の確認方法

マップファイルの各項目の詳細については、『MACU8 アセンブラパッケージ ユーザーズマニュアル』の「7.7 マップファイル」を参照してください。

### 1.1. 関連するドキュメント

本書に関連するドキュメント類を以下に示します。併せて参照してください。

マニュアル名称	説明	備考
関数・変数等の特定領域への配置方法	関数・変数等を特定領域に配置する方法について記載したドキュメントです。 プラグマを使って関数や変数等の配置を制御する場合に参照ください。	ラピスセミコンダクタのサポートサイトよりダウンロードいただけます。
MACU8 アセンブラパッケージ ユーザーズマニュアル	アセンブラ、リンカ、オブジェクトコンバータ、ライブラリアンの使用方法を記載したマニュアルです。 アセンブリファイルの疑似命令やマップファイルの見方などを確認したい場合に参照ください。	Windows の スタートメニュー [U8 Tools] > [V2_xx_xx] > [ドキュメント] から参照いただけます。 *1
CCU8 ユーザーズマニュアル	CCU8 C コンパイラの使用方法を記載したマニュアルです。 プラグマや組み込み関数の記述の仕方などを確認したい場合に参照ください。	
CCU8 プログラミングガイド	CCU8 C コンパイラ向けの C プログラムの記述の仕方や注意点などを記載したドキュメントです。 割込み関数やアセンブリ言語と組み合わせる場合の記述の仕方などを確認したい場合に参照ください。	
LEXIDE-U16 ユーザーズマニュアル	統合開発環境 LEXIDE-U16 の使用方法を記載したマニュアルです。 LEXIDE-U16 のオプションとコンパイラなど各ツールのオプションの対応を確認したい場合に参照ください。	Windows の スタートメニュー [U8 Tools] > [nX-U8 ドキュメント] から参照いただけます。

\*1: U8/U16 Development Tools Release 2.0.0 以降を対象としています。

## 2. 確認方法

### 2.1. ROM・RAM 使用量の確認方法

ROM・RAM の全体の使用量は、マップファイルに出力される情報で確認できます。  
下記のようにマップファイルの最後に出力されます。

```
Total ROM usage = 00C40    (3136)
Total RAM usage  = 00401    (1025)

End of mapfile.
```

ROM の合計使用量は、関数やコードオプションに該当する CODE タイプの ROM コードと、const 変数や変数の初期化データに該当する TABLE タイプの ROM コードの合計サイズとなります。

RAM の合計使用量は、const なしの変数やスタック領域(いずれも DATA タイプ)の合計サイズとなります。

上記の ROM・RAM 使用量は、『CCU8 プログラミングガイド』の「3.3 ROM サイズの算出方法」および「3.4 RAM サイズの算出方法」に基づいて算出されています。





## 2.3. 関数のサイズの確認方法

関数のサイズの確認は、通常の場合とそうでない場合で確認方法が異なります。  
通常の場合は、SEGCODE/SEGINTR プラグマを使わない場合が該当します。  
そうでない場合は、SEGCODE/SEGINTR プラグマを使った場合が該当します。

### 2.3.1. 通常の場合 (SEGCODE/SEGINTR プラグマを使わない場合)

C 言語で記述した関数は、コンパイラにより関数単位のセグメント\*1に分けられます。  
そして、C 言語で記述した関数に該当するセグメント名は、以下のように \$\$<関数名>\$<ソースファイル名> という名前で出力されます。  
関数単位でセグメントが分けられるので、関数に該当するセグメントを参照すれば、関数のサイズを確認できます。

マップファイル上には、以下のように出力されます。

Type	Start	Stop	Size	Name
...				
S CODE	00:0AD4	00:0AE9	0016 (22)	\$\$s_wait\$lcgu_main_0001
S CODE	00:0AEA	00:0AED	0004 (4)	\$\$lcg_initPeri\$lcg_initperi
S CODE	00:0AEE	00:0B63	0076 (118)	\$\$main\$lcgu_main_0001
...				
	↑		↑	↑    ↑
	関数のアドレス		関数のサイズ	関数名    ソースファイル名

\*1:セグメントとは、アドレスが連続したひとまとまりの領域です。



それぞれの関数のサイズは、上記のシンボル情報とセグメントのサイズから算出してください。  
算出の際は、アドレス昇順で並べ替えてから算出してください。

シンボル名	アドレス	算出方法	サイズ(10進)
CODE_MAIN	00:03E6	—	136 …(4)
_s_initUser	00:03E6 …(1)	アドレスの差分より算出 算出式:(2) – (1)	4 …(5)
_s_wait	00:03EA …(2)	アドレスの差分より算出 算出式:(3) – (2)	14 …(6)
_main	00:03F8 …(3)	最後の関数のサイズは、セグメントのサイズから、最後以外の関数のサイズの合計を差し引いて算出 算出式:(4) – ((5) + (6))	118

## 2.4. スタック領域の確認方法

スタック領域はデータメモリのリンクマップから確認できます。  
 スタック領域は、\$STACK という名前のセグメントとして、以下のようにマップファイルに出力されます。

Type	Start	Stop	Size	Name
Q ROMWIN	00:0000	00:DFFF	E000 (57344)	(ROMWIN)
...				
S DATA	00:EBFC	00:EFFB	0400 (1024)	\$STACK ←
...				

スタック領域に該当するセグメント (スタックセグメント)  
 ↑ スタック領域のアドレス  
 ↑ スタック領域のサイズ

DTU8 デバッガのブレイク条件の設定にて、スタックポインタオーバーフローブレイクのオーバーフロー/アンダーフローのアドレス設定には、\$STACK を使うことができます。

### 【スタックオーバーフロー選択時の RAM アドレス設定】



### 【スタックアンダーフロー選択時の RAM アドレス設定】



改版履歴

ドキュメント No.	発行日	ページ		変更内容
		改版前	改版後	
FJXT_MCU_HOWTO_USE_MAPFIE -01	2020.2.28	—	—	初版発行