

ラピステクノロジー8/16ビットマイコン データ・フラッシュ領域の初期値設定

発行日 2023年1月13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) 本製品をご使用の際は、最新の製品情報をご確認の上、絶対最大定格、動作条件その他の指定条件の範囲内でお使いください。指定条件の範囲を超えて使用された場合や、使用上の注意を守ることなく使用された場合、その後に発生した故障、誤動作等の不具合、事故、損害等については、ラピステクノロジー株式会社(以下、「当社」といいます)はいかなる責任も負いません。また、指定条件の範囲内のご使用であっても、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。万が一が本製品が故障・誤作動した場合でも、その影響により人身事故、火災損害等が起らないよう、お客様の責任において、デレーティング、冗長設計、延焼防止、バックアップ、フェイルセーフ等お客様の機器・システムとしての安全確保を行ってください。
- 3) 本資料に記載されております応用回路例やその定数、ソフトウェア等の情報は、半導体製品の標準的な動作例や応用例を説明するものです。お客様の機器やシステムの設計においてこれらの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。また、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。これらのご使用に起因して生じた損害等に関し、当社は一切その責任を負いません。
- 4) 本資料に記載された製品データ、図、表、プログラム、アルゴリズム、応用回路例等の技術情報は、それをもって当該技術情報に関する当社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、当該技術情報を使用されたことによる第三者の知的財産権に対する侵害またはこれらに関する紛争について、当社は何ら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ご使用を意図しています。
本製品を、特に高い信頼性が要求される機器(車載・船舶・鉄道等の輸送機器、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、送電システム等)に使用される際は、必ず当社へご連絡の上、書面にて承諾を得てください。
当社の意図していない用途に製品を使用したことにより損害が生じても、当社は一切その責任を負いません。
また、本製品は直接生命・身体に危害を及ぼす可能性のある機器・システム、極めて高い信頼性を要求される機器(航空宇宙機器、原子力制御機器、海底中継機器等)には、使用できません。
- 6) 本資料に掲載されております製品は、耐放射線設計がなされておられません。
- 7) 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、当社はその責任を負うものではありません。
- 8) 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、当社は一切の責任を負いません。
- 9) 本製品および本資料に記載の技術を輸出また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。
- 10) 本資料に記載されている内容または本製品についてご不明な点がございましたらセールスオフィスまでお問い合わせください。
- 11) 本資料の一部または全部を当社の許可なく、転載・複製することを堅くお断りします。

Copyright 2018-2023 LAPIS Technology Co., Ltd.

ラピステクノロジー株式会社

〒222-8575 神奈川県横浜市港北区新横浜 2-4-8

<https://www.lapis-tech.com/>

目次

1. はじめに.....	1
2. 対象商品	1
3. データ・フラッシュ領域の初期値設定.....	2
3.1 Cソースファイル, およびスタートアップファイル*1での設定方法.....	2
3.2 スタートアップファイルのみでの設定方法	3
4. データ・フラッシュ領域を含む ROM コードデータの出力方法.....	4
4.1 LEXIDE-U16 で設定する場合	4
4.2 HTU8 で設定する場合	4
5. 改版履歴	5

1. はじめに

本ドキュメントはデータ・フラッシュ領域に初期値を設定する方法について記載しています。

2. 対象商品

ML62Q1000 シリーズ

ML62Q2000 シリーズ

ML610Q100

ML610Q300

ML620Q100

ML620Q500

3. データ・フラッシュ領域の初期値設定

データ・フラッシュ領域のアドレスはマイクロコントローラによって異なります。データ・フラッシュ領域のアドレスについては、お使いのマイクロコントローラのマニュアルを参照ください。

本書では、ML62Q1000 シリーズで、アドレス 0x1f:0000～0x1f:0007 の連続した 8 バイト、およびアドレス 0x1f:0400～0x1f:0407 の連続した 8 バイトに初期値を設定する例を示します。

2 種類の方法について以下に記載します。

3.1 C ソースファイル、およびスタートアップファイル*1 での設定方法

C ソースファイルに、const 修飾した変数を `__far` で指定し、その変数に対して `segconst` プラグマにより配置アドレスを指定してください。

<C ソースファイル>

```
#pragma segconst __far 0x1f:0x0000
const char __far data_flash_init1[8] = {
    0x00, 0x10, 0x20, 0x30, 0x40, 0x50, 0x60, 0x70};
#pragma segconst

#pragma segconst __far 0x1f:0x0400
const char __far data_flash_init2[8] = {
    0x80, 0x90, 0xa0, 0xb0, 0xc0, 0xd0, 0xe0, 0xf0};
#pragma segconst
```

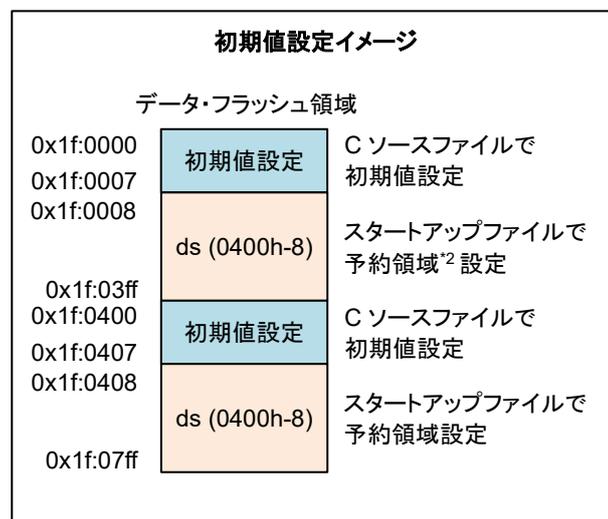
なお、この状態でそのままビルドすると、スタートアップファイルに設定してあるデータフラッシュ領域と上記で設定した初期値のデータがオーバーラップしている旨のワーニングが表示されてしまいます。オーバーラップのワーニングが表示されないようにするために、スタートアップファイルも修正してください。

以下に例を示します。

<スタートアップファイル>

```
-----
; Keeping the Self-Programming area (for ML621000)
;-----
; 修正前:
; tseg #1fh at 00000h
; ds 0800h
; 修正後:
; tseg #1fh at 00008h
; ds (0400h-8)

; tseg #1fh at 00408h
; ds (0400h-8)
```



1: スタートアップファイルは RAM および C 変数を初期化するアセンブリファイル(.ASM)です。

*2: 予約領域はデータ・フラッシュ領域にプログラムコードが割り付けられないようにする領域です。

3.2 スタートアップファイルのみでの設定方法

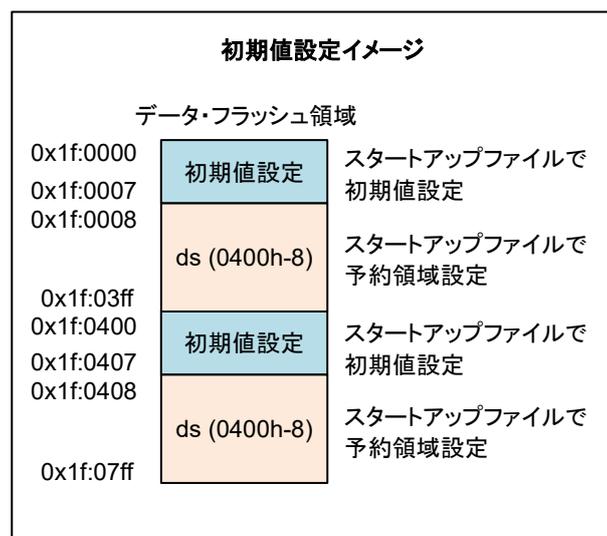
スタートアップファイルに直接初期値を記入してください。

```

<スタートアップファイル>
;-----
;      Keeping the Self-Programming area (for ML621000)
;-----
; 修正前:
;      tseg #1fh at 00000h
;      ds      0800h
; 修正後:
      tseg #1fh at 00000h          ; address
      db      000h                ; 00000h
      db      010h                ; 00001h
      db      020h                ; 00002h
      db      030h                ; 00003h
      db      040h                ; 00004h
      db      050h                ; 00005h
      db      060h                ; 00006h
      db      070h                ; 00007h
      ds      (0400h-8)

      tseg #1fh at 00400h          ; address
      db      080h                ; 00400h
      db      090h                ; 00401h
      db      0a0h                ; 00402h
      db      0b0h                ; 00403h
      db      0c0h                ; 00404h
      db      0d0h                ; 00405h
      db      0e0h                ; 00406h
      db      0f0h                ; 00407h
      ds      (0400h-8)

```



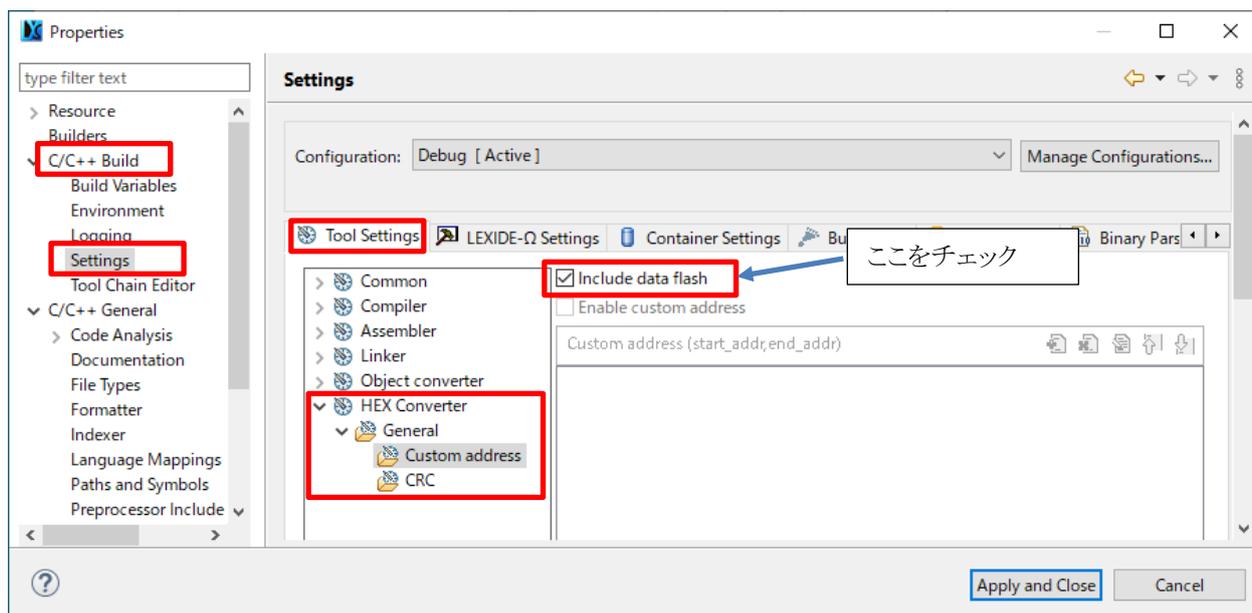
4. データ・フラッシュ領域を含む ROM コードデータの出力方法

データ・フラッシュ領域を含めてマイクロコントローラのフラッシュ・メモリに書き込む場合などは、プログラム領域とデータ・フラッシュ領域をあわせて ROM コードデータに出力する必要があります。

データ・フラッシュ領域を含めて ROM コードデータに出力するには、データ・フラッシュ領域のデータを ROM コードデータに出力することを明示的に設定してください。

4.1 LEXIDE-U16 で設定する場合

LEXIDE-U16 では[Properties]ダイアログの[C/C++ Build]>[Settings]>[Tool Settings]タブ
>[HEX Converter]>[General]>[Custom address]で
[Include data flash]をチェックします。



上記を設定してビルドすることにより、データフラッシュ領域を含む ROM コードデータを作成できます。
LEXIDE-U16 のビルドオプションおよびビルド方法については、『LEXIDE-U16 ユーザーズマニュアル』の「4.3 ビルドオプション設定」および「4.7 ビルド」を参照してください。

4.2 HTU8 で設定する場合

HTU8 では、[ターゲットチップ]の下の[データフラッシュを含む]をチェックします。



上記を設定して[実行]ボタンをクリックするとデータフラッシュ領域を含む ROM コードデータを作成できます。
HTU8 の設定および操作方法については、『HTU8 ユーザーズマニュアル』の「4. 機能 (GUI 編)」を参照してください。

5. 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_DFLASHINIT-01	2018.6.28	—	—	初版発行
FJXT_MCU_DFLASHINIT-02	2023.1.13	—	4	「4.データ・フラッシュを含む ROM コードデータの出力方法」を追加