

RL78ファミリ用Cコンパイラ CA78K0R スタートアップルーチンの変更について

RL78/G10向けのコードサイズの削減について

株式会社ルネサス ソリューションズ
ツールビジネス本部 ツール技術部

2014/6/20 Rev. 1.00

R20UT3046JJ0100

はじめに

- **本資料は、RL78/G10を使用する際に、スタートアップルーチンのコードサイズを削減する方法について説明します。**
- **CA78K0Rのスタートアップルーチンは、RL78で共通の仕様です。**
- **RL78/G10が、次の仕様であることを考慮した変更方法の説明です。そのため、他のマイコンを使用した際には正常に動作しない可能性がありますので、ご注意ください。**
 - **RL78-S1コアである**
 - **汎用レジスタバンクを搭載していない**
 - **外部メモリアクセス機能を搭載していない**
 - **MAAが0固定**
 - **RAMサイズが小さい(512バイト)**

- CA78K0Rのスタートアップルーチン内の処理概要
- RL78/G10向けのスタートアップルーチンのカスタマイズ概要
- CubeSuite+でのスタートアップルーチンの登録
- スタートアップルーチンの編集

CA78K0Rのスタートアップルーチン内の処理概要

- cstartn.asm の処理の概要は次のようになっています。

setting the register bank RB

setting the mirror area

setting the stack pointer

call hardware initialization

clear stack area

clear saddr area

copy external variables having initial value (near)

copy external variables having initial value (far)

copy external variables which doesn't have initial value (near)

copy external variables which doesn't have initial value (far)

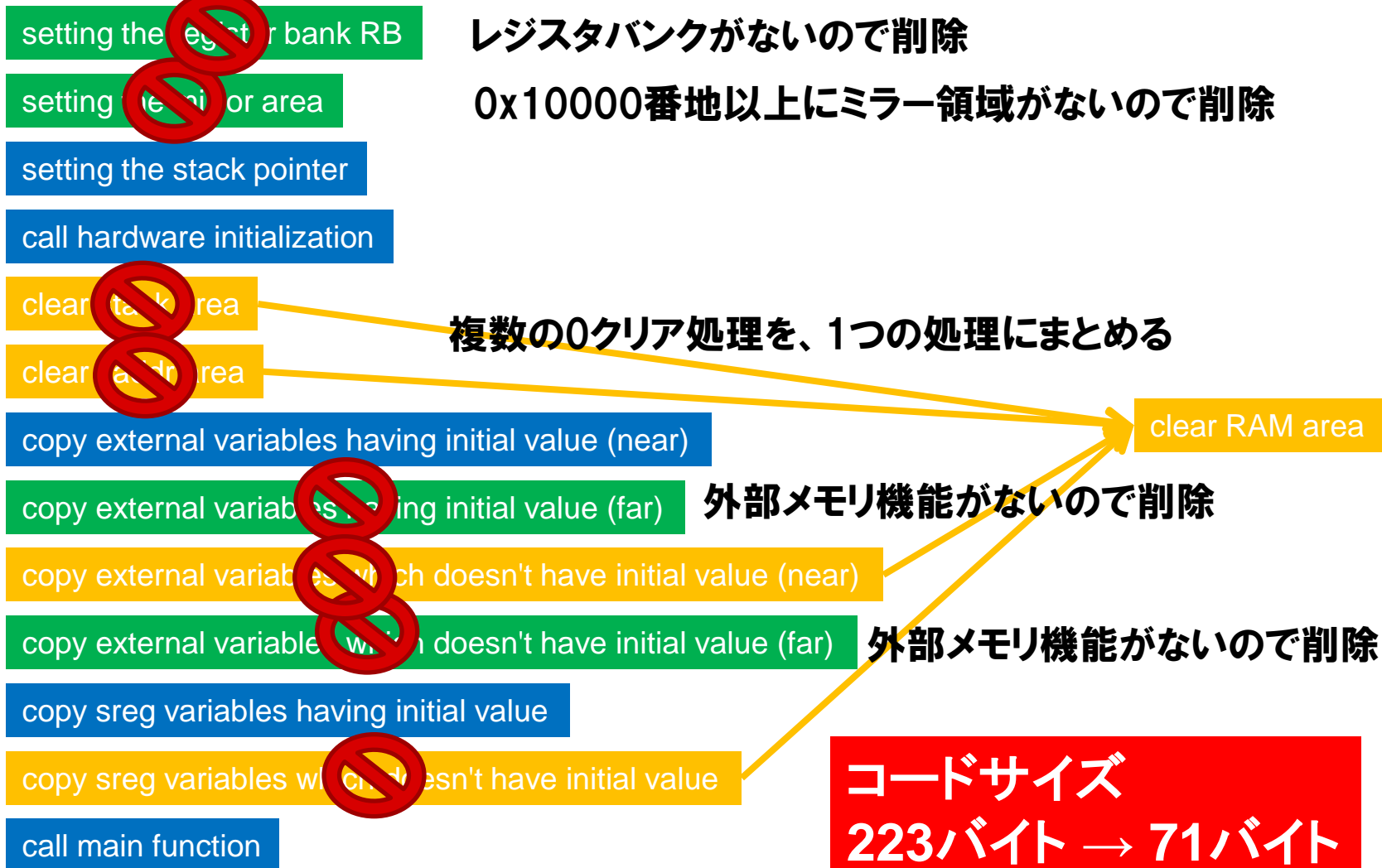
copy sreg variables having initial value

copy sreg variables which doesn't have initial value

call main function

RL78/G10向けのスタートアップルーチンのカスタマイズ概要

- cstartn.asmに対して、次の変更を行います。

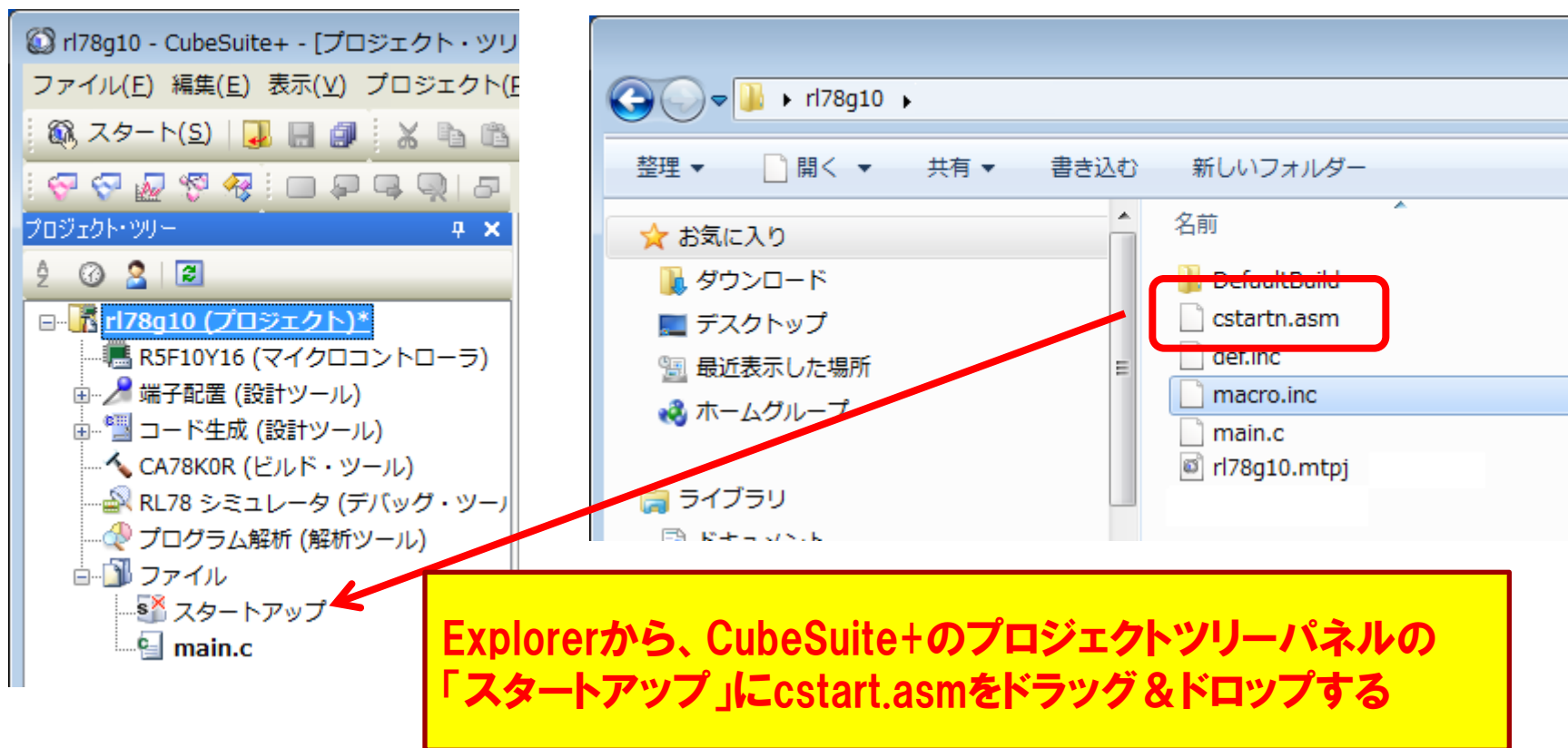


CubeSuite+ でのスタートアップルーチンの登録(1/3)

- インストールフォルダからスタートアップルーチンのファイルを任意のフォルダコピーする
 - コピー対象ファイル
 - cstartn.asm
 - def.inc
 - macro.inc
 - デフォルトのインストールフォルダ(スタートアップ関連ファイル)
 - C:¥Program Files¥Renesas Electronics¥CubeSuite+¥CA78K0R¥バージョン¥Src¥cc78k0r¥src
(バージョン:CA78K0Rのバージョンです)

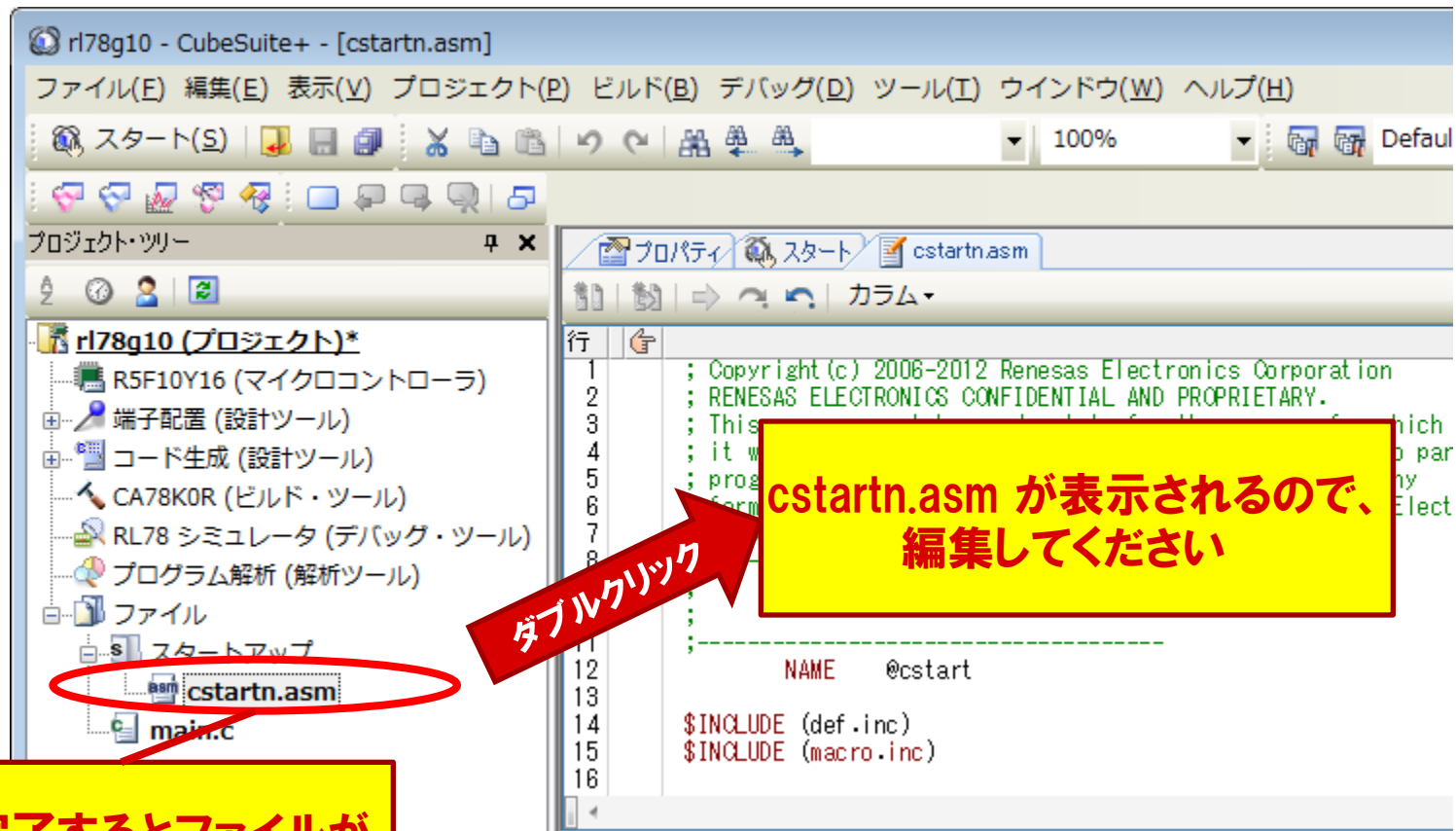
CubeSuite+ でのスタートアップルーチンの登録(2/3)

■ コピーしたスタートアップルーチンをCubeSuite+に登録する



CubeSuite+ でのスタートアップルーチンの登録(3/3)

- プロジェクトツリーに登録されたら、編集してください。



登録完了するとファイルが表示される

スタートアップルーチンの編集(1/13)

変更なし

```
; Copyright(c) 2006-2012 Renesas Electronics Corporation  
; RENESAS ELECTRONICS CONFIDENTIAL AND PROPRIETARY.  
; This program must be used solely for the purpose for which  
; it was furnished by Renesas Electronics Corporation. No part of this  
; program may be reproduced or disclosed to others, in any  
; form, without the prior written permission of Renesas Electronics  
; Corporation.
```

```
-----  
;           W-1      cstart (no init)  
;           ;  
;           ;
```

```
-----  
;           NAME      @cstart
```

```
$INCLUDE (def.inc)  
$INCLUDE (macro.inc)
```

スタートアップルーチンの編集(2/13)

変更なし

```
-----  
; declaration of symbol  
;  
; attention):          change to EQU value 1 -> 0 if necessary  
-----  
;  
FLOATSW EQU 1          ;floating point variables use  
  
        PUBLIC  _@cstart,_@cend  
  
$_IF(FLOATSW)  
        PUBLIC  _errno  
$ENDIF
```

スタートアップルーチンの編集(3/13)

```
-----  
; external declaration of symbol for stack area  
;  
; _@STBEG has value of the end address +1 of compiler's stack area.  
; _@STEND has value of the start address of compiler's stack area.  
; _@STBEG and _@STEND are created by linker with -S option.  
; Accordingly, specify the -S option when linking.  
;  
; _@MAA has value of the mirror area.  
; _@SADSIZ has value of the size of the saddr area.  
; _@SADBEG has value of the start address of the saddr area.  
-----  
EXTRN    _main,_@STBEG,_@STEND,_stkinit,_hdwinit,_@SADSIZ,_@SADBEG  
$_IF( MIRROR)  
EXTRN    _@MAA  
$ENDIF  
EXTRN    _?R_INIT,_?RLINIT,_?R_INIS,_?DATA,_?DATAL,_?DATS
```



スタートアップルーチンの編集(4/13)

変更なし

```
-----  
; allocation area which library uses  
; _errno   errno number code  
-----  
@@DATA   DSEG      BASEP      ;near  
_@DATA:  
$_IF(FLOATSW)  
_errno:   DS        2  
$ENDIF  
  
@@VECT00CSEG   AT        0  
                DW      _@cstart  
  
@@LCODE CSEG      BASE  
  
_@cstart:
```

スタートアップルーチンの編集(5/13)

```
-----  
; setting the register bank RBO as work register set  
SEL      RBO ← 削除  
-----  
; setting the mirror area  
-----  
$_IF(_MIRROR)  
  MOV     A,#_@MAA  
  MOV1    CY,A.0  
  MOV1    MAA,CY  
$ENDIF ← 削除  
-----  
; setting the stack pointer  
; _@STBEG is created by linker with -S option.  
-----  
      MOVW   SP,#LOWW _@STBEG ;SP <- stack begin address  
-----  
; hardware initialization  
-----  
      CALL   !!_hdwinit
```

スタートアップルーチンの編集(6/13)

```
-----  
; clear stack area  
;-----  
; _@STEND is created by linker with -S option.  
-----  
MOVW    HL,#LOWW _@STEND    ; HI <- stack end address  
CALL    !!_stkinit ← 削除  
-----  
; clear saddr  
-----  
MOV     B,#_@SADSIZ        ; 0FEDFH - _@SADBEG + 1  
CLRW   AX  
LSADR1:  
DEC    B  
DEC    B  
MOVW  _@SADBEG[B],AX  
BNZ   $LSADR1 ← 削除
```

スタートアップルーチンの編集(7/13)

追加

```
-----  
; clear Internal RAM ; add for RL78/G10  
-----
```

```
MOVW    HL,#LOWW OFFDE0H ; Start address of Internal RAM  
MOVW    AX,#LOWW OFFEE0H ; End address of Internal RAM + 1  
BR      $LDATA2
```

LDATA1:

```
MOV     [HL+0],#0  
INCW   HL
```

LDATA2:

```
CMPW   AX,HL  
BNZ    $LDATA1
```

スタートアップルーチンの編集(8/13)

変更なし

```
-----  
; ROM data copy  
-----  
  
; copy external variables having initial value  
$_IF(_ESCOPY)  
    MOV     ES,#HIGHW _@R_INIT  
$ENDIF  
  
    MOVW   HL,#LOWW _@R_INIT  
    MOVW   DE,#LOWW _@INIT  
    BR     $LINIT2  
  
LINIT1:  
$_IF(_ESCOPY)  
    MOV     A,ES:[HL]  
$ELSE  
    MOV     A,[HL]  
$ENDIF  
  
    MOV     [DE],A  
    INCW   HL  
    INCW   DE  
  
LINIT2:  
    MOVW   AX,HL  
    CMPW   AX,#LOWW _?R_INIT  
    BNZ    $LINIT1
```


スタートアップルーチンの編集(9/13)

```
$_IF(_DATAFAR)  
; copy external variables having initial value  
    MOVW    HL,#LOWW_@RLINIT  
    MOVW    DE,#LOWW_@INITL  
    BR      $LINIT4  
  
LINIT3:  
    MOV     ES,#HIGHW_@RLINIT  
    MOV     A,ES:[HL]  
    MOV     ES,#HIGHW_@INITL  
    MOV     ES:[DE],A  
    INCW    HL  
    INCW    DE  
  
LINIT4:  
    MOVW    AX,HL  
    CMPW    AX,#LOWW_?RLINIT  
    BNZ     $LINIT3  
  
$ENDIF
```



スタートアップルーチンの編集(10/13)

```

; copy external variables which doesn't have initial value
      MOVW    HL,#LOWW_@DATA
      MOVW    AX,#LOWW_?DATA
      BR      $LDATA2
LDATA1:
      MOV     [HL+0],#0
      INCW   HL
LDATA2:
      CMPW   AX,HL
      BNZ   $LDATA1
$_IF(_DATAFAR)
; copy external variables which doesn't have initial value
      MOV     ES,#HIGHW_@DATAL
      MOVW   HL,#LOWW_@DATAL
      MOVW   AX,#LOWW_?DATAL
      BR     $LDATA4
LDATA3:
      MOV     ES:[HL+0],#0
      INCW   HL
LDATA4:
      CMPW   AX,HL
      BNZ   $LDATA3
$ENDIF

```



スタートアップルーチンの編集(11/13)

変更なし

```
; copy sreg variables having initial value
$_IF(_ESCOPY)
    MOV        ES,#HIGHW_@R_INIS
$ENDIF
    MOVW       HL,#LOWW_@R_INIS
    MOVW       DE,#LOWW_@INIS
    BR         $LINIS2
LINIS1:
$_IF(_ESCOPY)
    MOV        A,ES:[HL]
$ELSE
    MOV        A,[HL]
$ENDIF
    MOV        [DE],A
    INCW       HL
    INCW       DE
LINIS2:
    MOVW       AX,HL
    CMPW       AX,#LOWW_?R_INIS
    BNZ        $LINIS1
```

スタートアップルーチンの編集(12/13)

```
; copy sreg variables which doesn't have initial value
```

```
MOVW HL,#LOWW_@DATS
```

```
MOVW AX,#LOWW_?DATS
```

```
BR $LDATS2
```

```
LDATS1:
```

```
MOV [HL+0],#0
```

```
INCW HL
```

```
LDATS2:
```

```
CMPW AX,HL
```

```
BNZ $LDATS1
```



削除

```
-----
```

```
; call main function
```

```
-----
```

```
CALL !!_main ;main();
```

```
BR $$
```

```
;
```

```
__@cend:
```

スタートアップルーチンの編集(13/13)

変更なし

```
-----  
; define segment and label used by ROMable processing  
-----
```

```
@@R_INIT CSEG      UNIT64KP  
_R_INIT:  
$_IF(_DATAFAR)  
@@RLINIT CSEG      UNIT64KP  
_RLINIT:  
$ENDIF  
@@R_INIS CSEG      UNIT64KP  
_R_INIS:  
@@INIT DSEG        BASEP  
_INIT:  
$_IF(_DATAFAR)  
@@INITL DSEG       UNIT64KP  
_INITL:  
$ENDIF  
$_IF(_DATAFAR)  
@@DATAL DSEG       UNIT64KP  
_DATAL:  
$ENDIF  
@@INIS DSEG        SADDRP  
_INIS:  
@@DATS DSEG        SADDRP  
_DATS:  
@@CALT CSEG        CALLT0  
@@CNST CSEG        MIRRORP  
@@CNSTL CSEG       PAGE64KP  
@@BITS BSEG  
;  
END
```

RENEASAS

ルネサス ソリューションズ株式会社

© 2014 Renesas Solutions Corp.