
从 R8C 转至 RL78 的迁移指南 (CcnvNC30)

R01AN3508CC0100
Rev.1.00
2017.03.31

要点

本篇应用说明介绍了将程序从 R8C 转至 RL78 的方法。

对象 MCU

RL78、R8C 族

将本篇应用说明应用到其他 MCU 时，请根据 MCU 的规格进行详细的评价。

目录

1.	将程序从 R8C 转至 RL78 的方法.....	3
2.	通过 CcnvNC30 进行的程序转换.....	3
2.1	CcnvNC30.....	3
2.2	CcnvNC30 的用法.....	4
2.3	不使用 CcnvNC30 时.....	6
3.	外围功能的程序转换.....	7
3.1	程序的自动生成.....	7
3.2	程序的添加.....	9
3.3	不使用代码生成工具时.....	9
4.	置换实例.....	10
4.1	R8C 参考例程 (使用 RTC 时钟运行模式).....	10
4.1.1	使用 CcnvNC30 为 CC-RL 导入源文件.....	10
4.1.2	程序的自动生成.....	12
4.1.3	程序的添加.....	14
4.1.4	其他变更事项.....	18
4.1.5	置换后的参考例程.....	19
4.2	R8C 的参考例程 (DTC 运行).....	20
4.2.1	使用 CcnvNC30 为 CC-RL 导入源文件.....	20
4.2.2	程序的自动生成.....	21
4.2.3	程序的添加.....	24
4.2.4	其他变更事项.....	25
4.2.5	置换后的参考例程.....	26
4.3	参考例程的动作确认条件.....	27
5.	参考例程.....	27
6.	参考文献.....	27
	公司主页和咨询窗口.....	27

1. 将程序从 R8C 转至 RL78 的方法

本节介绍将程序从 R8C 转至 RL78 的方法。

首先通过 C 源代码转换器 CcnvNC30 将 C 编译器 NC30 的扩展功能转换为 C 编译器 CC-RL 的扩展功能。

然后通过集成开发环境 CS+ 或者 e2studio 创建工程。由于 R8C 与 RL78 的外围功能不同，所以不使用 R8C 的外围功能程序而选择 RL78 的代码生成工具来生成 RL78 的外围功能程序。

将通过 Ccnv NC30 转换的程序与上述外围功能程序组合后替换程序。

2. 通过 CcnvNC30 进行的程序转换

2.1 CcnvNC30

CcnvNC30 利用 NC30 的 C 源程序，将扩展语言说明（如宏名称、保留字、#pragma 指令、扩展功能）转换为 CC-RL 的扩展语言说明。

CcnvNC30 是一种将用于 NC30 的程序转为 CC-RL 上可用的程序的软件。但因无法保证转换后的程序一定可以运行，因此务必要在系统上对转换后的程序进行动作确认。

另外，配置地址、SFR 的访问以及汇编语言编码等依存于设备的编码是无法转换的，因此需要根据具体情况手动修改以适用于 RL78。

详情请参照 CcnvNC30 C Source Converter User's Manual (R20UT3685E)。

2.2 CcnvNC30 的用法

通过 CcnvNC30 转换程序的方法如下所示。

- (1) 将 CcnvNC30 (CcnvNC30.exe) 与用于 NC30 的程序置于同一个任意的文件夹中。
- (2) 在 Windows 中启动命令提示符 (Command Prompt)。
- (3) 将当前目录变更为存有 CcnvNC30 的文件夹。



图 2.1 命令提示符窗口

- (4) 执行前在 -o 选项中指定输出文件名。执行后即输出用于 CC-RL 的程序。另外，如要将信息输出至指定文件中，则使用 -r 选项。

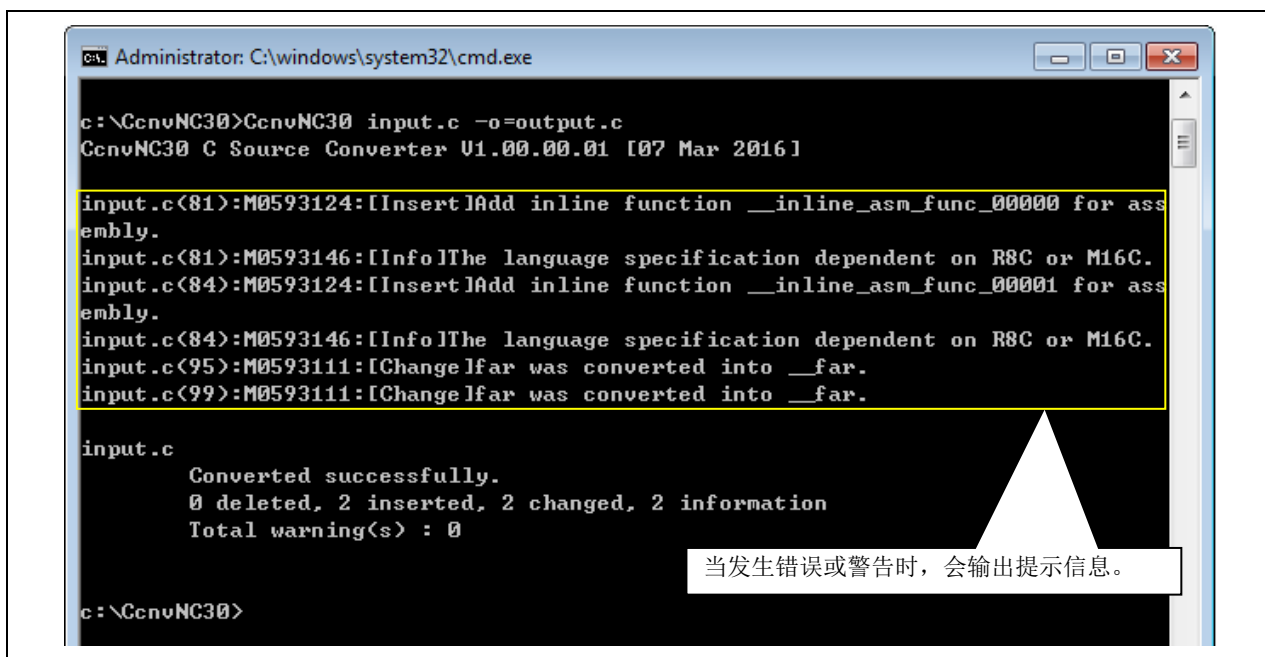


图 2.2 CcnvNC30 执行界面

从 R8C 转至 RL78 的迁移指南 (CcnvNC30)

- (5) 若要同时转换多个文件，需创建编目文件后执行-l 选项。执行后，CC-RL 程序会被输出至指定的文件夹中。

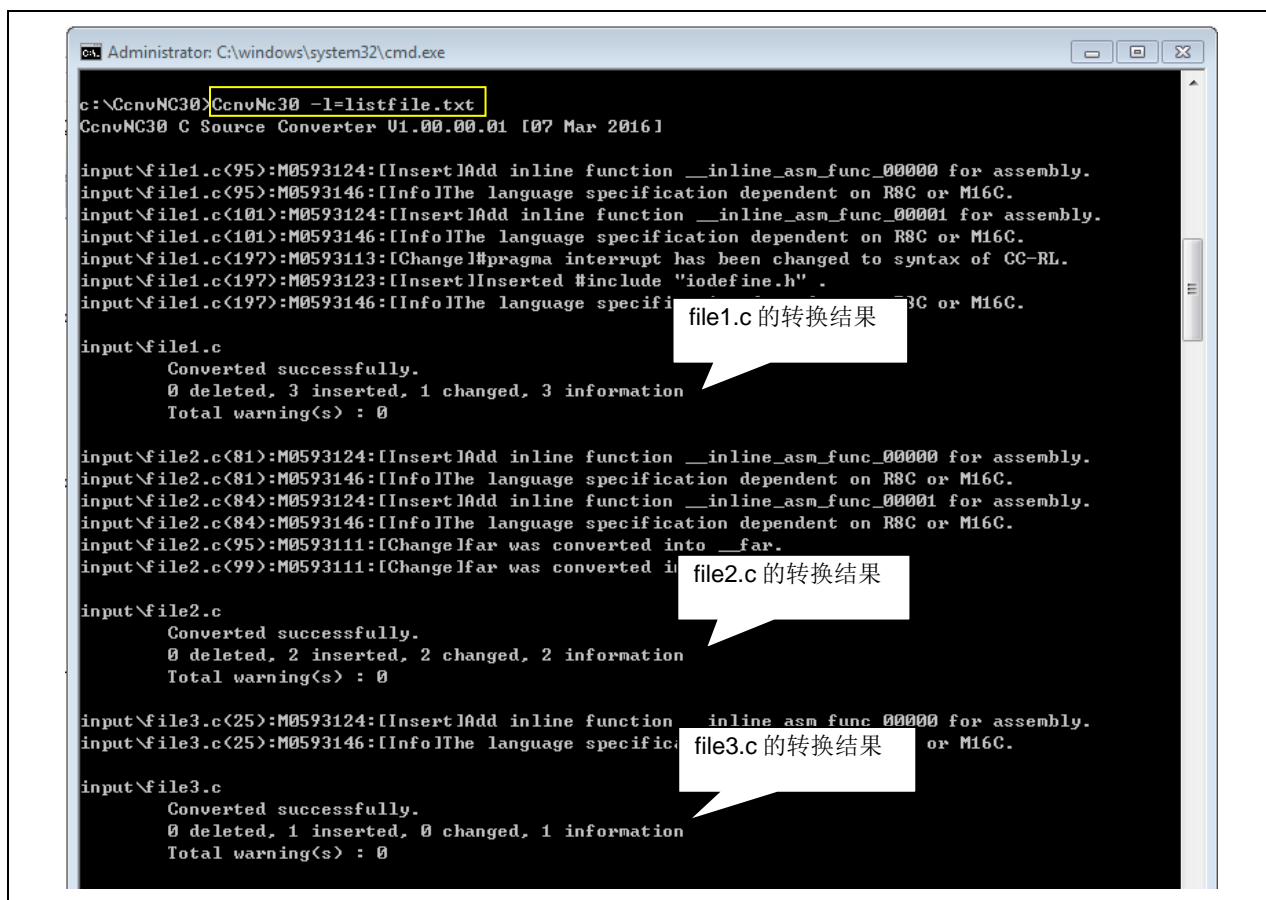


图 2.3 CcnvNC30 执行界面 (多个文件)

以下为编目文件记述例。

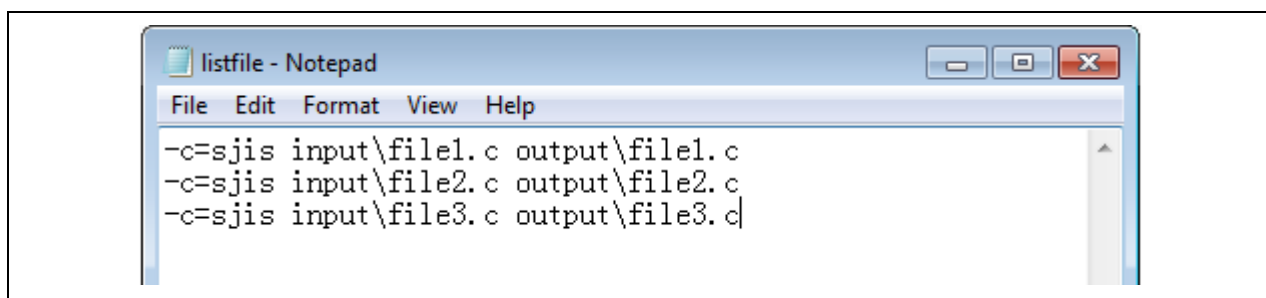


图 2.4 编目文件记述例

- (6) 对 CcnvNC30 未转换的地方进行修改。具体修改处请参照 “CcnvNC30 C Source Converter User's Manual (R20UT3685E)” 中的 “CONVERSION SPECIFICATIONS”。

2.3 不使用 CcnvNC30 时

不使用 CcnvNC30 时，必须手动将 NC30 的扩展功能修改为 CC-RL 的扩展功能。关于 CC-RL 的扩展语言说明，请参照 “CC-RL Compiler User's Manual (R20UT3123E)”。

3. 外围功能的程序转换

3.1 程序的自动生成

通过集成开发环境 CS+或 e2studio 中用于 RL78 的代码生成器，为 RL78 的外围功能自动生成对应于 R8C 族所用外围功能的程序。关于代码生成工具的操作方法，请参照“CS+ Code Generator Integrated Development Environment User's Manual: Peripheral Function Operation (R20UT3104E)”。

- (1) 在“Project Tree”中点击“Code Generator (Design Tool)”下的“Clock Generator”，并进行“Pin assignment”的设置。

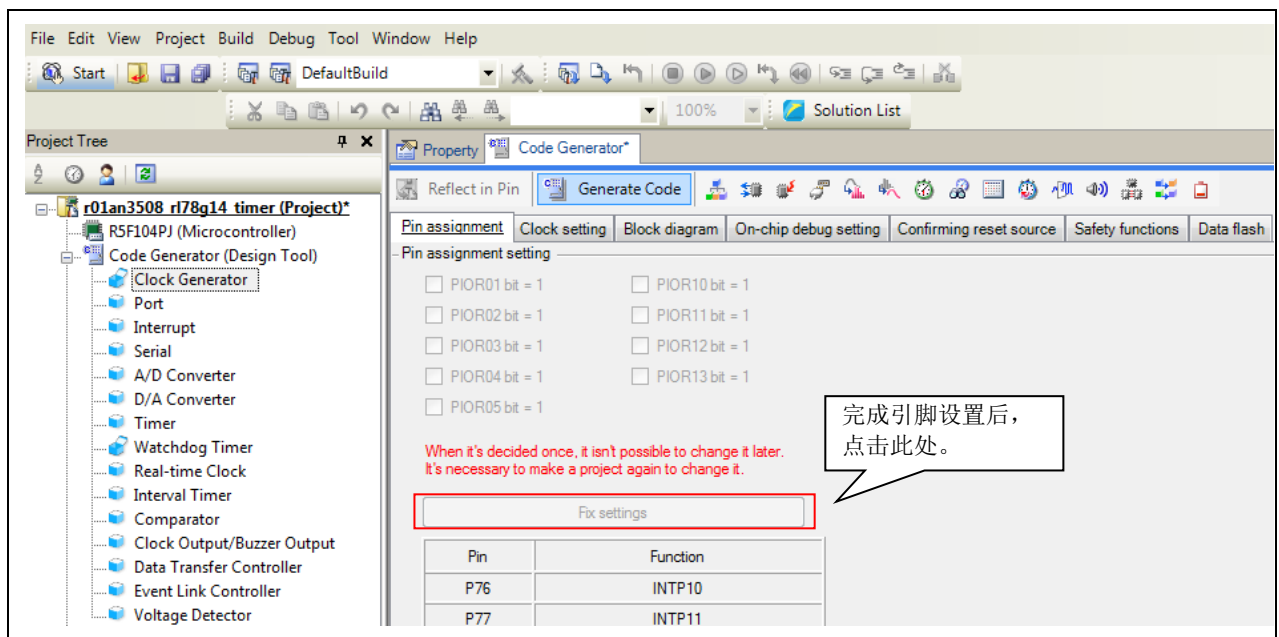


图 3.1 代码生成器窗口 (1)

- (2) 参照 R8C 族的程序设置每个功能。

从 R8C 转至 RL78 的迁移指南 (CcnvNC30)

- (3) 所有外围功能设置完成后，请点击位于界面上方的“Generate Code”生成代码（即程序的自动生成）。然后用自动生成的外围功能函数置换程序。

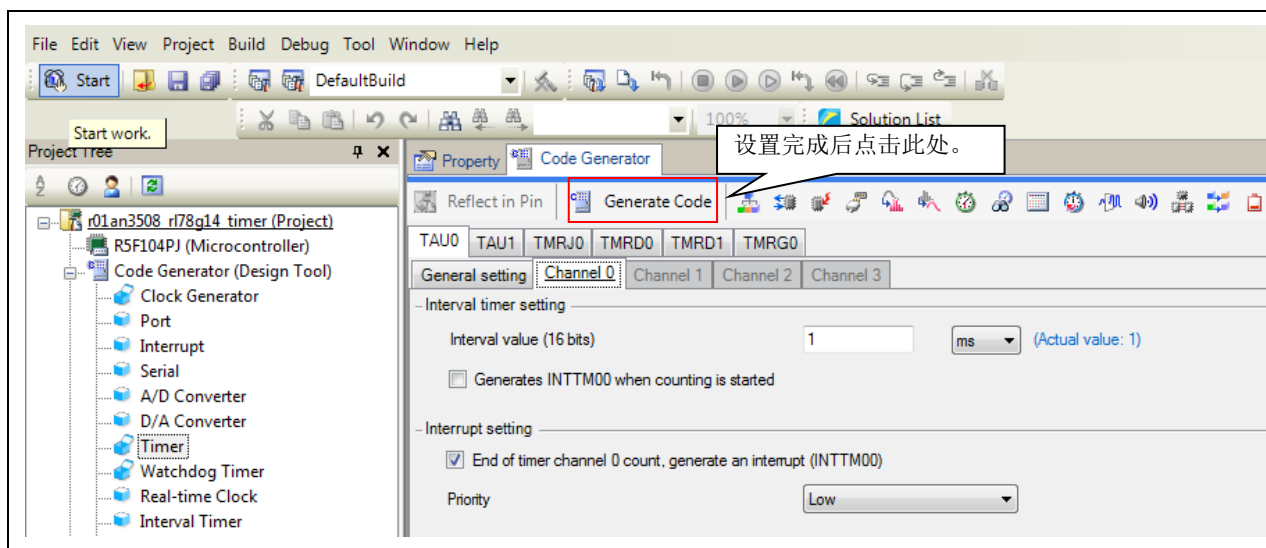


图 3.2 代码生成器窗口 (2)

3.2 程序的添加

无法通过代码生成工具自动生成的程序（如 main 函数、中断函数处理以及变量等），需手动添加。

在自动生成的各文件中的 “/* Start user code for adding. Do not edit comment generated here */” 和 “/* End user code. Do not edit comment generated here */” 之间添加程序。程序的添加需要手动进行。另外请注意，如在上述范围之外添加程序，在自动生成程序时，会被自动删除。

请务必使用添加好的程序对系统进行动作确认。

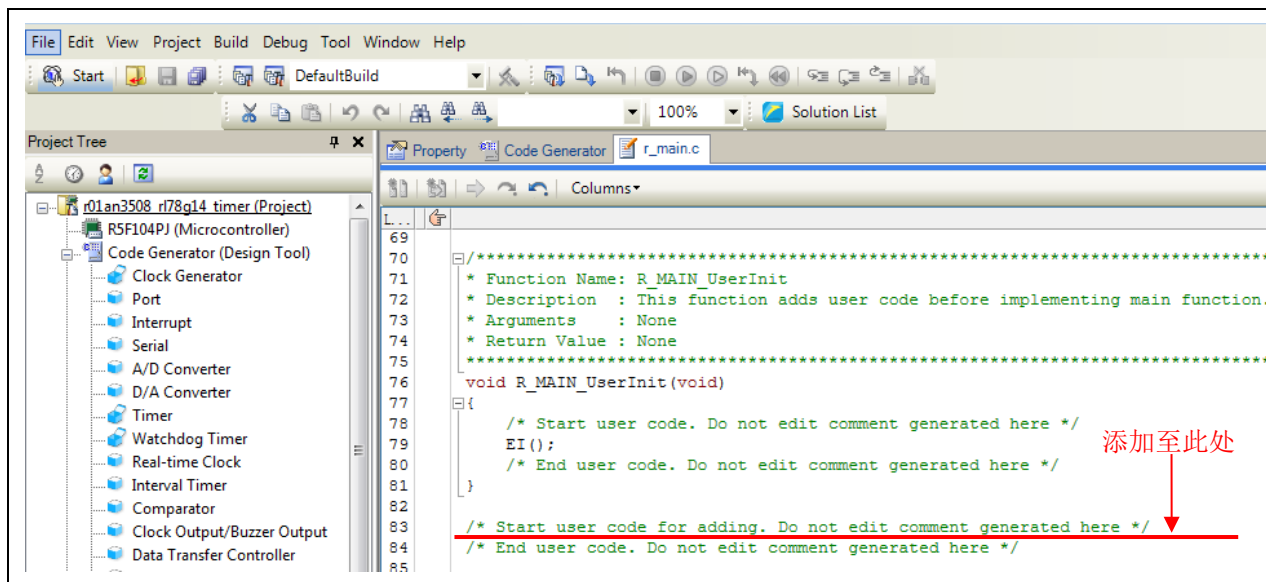


图 3.3 添加现有程序

3.3 不使用代码生成工具时

不使用代码生成工具时，通过集成开发环境 CS+或 e2studio 创建了一个新程序后，需手动创建外围功能程序。关于外围功能的详细情况，请参照 RL78 用户手册。

4. 置换实例

4.1 R8C 参考例程 (将 RTC 用作时钟的运行)

“R8C/35A Group Clock Operation Using RTC (R01AN0079E)” 的程序被替换为 RL78/G14 的程序。替换后的程序文件为 “r01an3508_rl78g14_rtc”。

该程序使用定时器 RE (实时时钟模式) 运行时钟。以 20MHz 的 XIN 时钟作为 CPU 时钟, 以 fC4 (XCIN 时钟 (32.768kHz) 的 4 分频) 作为定时器 RE 计数源。

4.1.1 使用 CcnvNC30 为 CC-RL 导入源文件

- (1) 创建编目文件以指定一个要转换的 C 源文件。

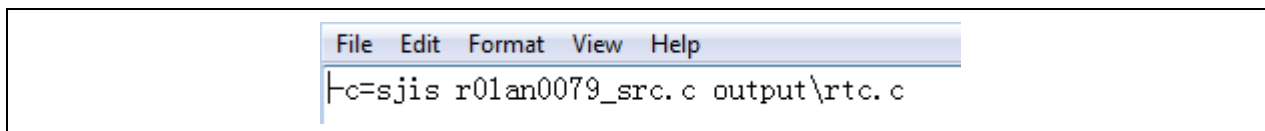


图 4.1 编目文件记述例 (将 RTC 用作时钟的运行)

- (2) 启动命令提示符 (Command Prompt), 对编目文件指定的 C 源文件进行转换。输出的转换结果文件中标有变更处。

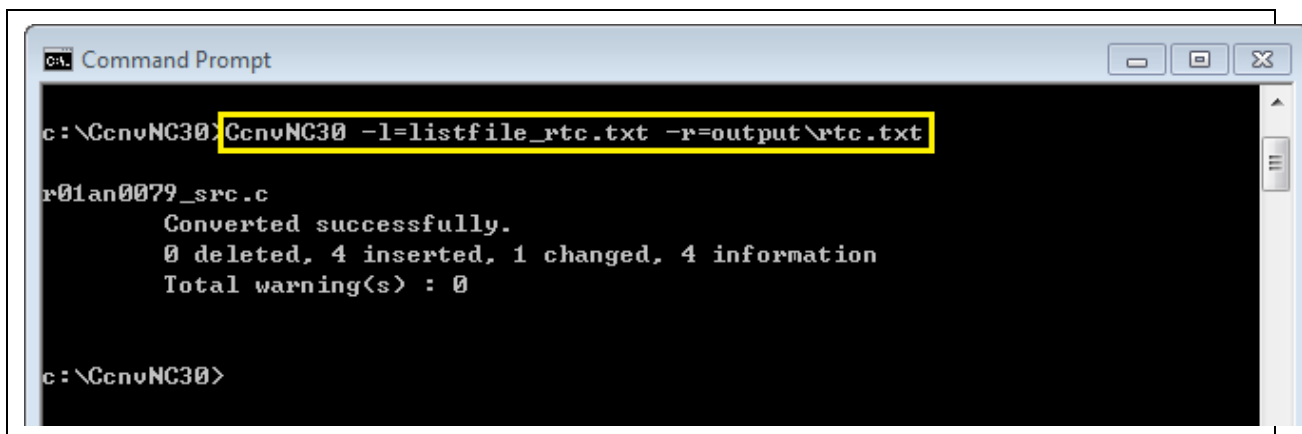


图 4.2 CcnvNC30 执行界面 (将 RTC 用作时钟的运行)

转换结果文件中记载着转换结果 (如下图所示)。关于具体的转换结果, 请参照 “CcnvNC30 C Source Converter User's Manual (R20UT3685EJ)”。

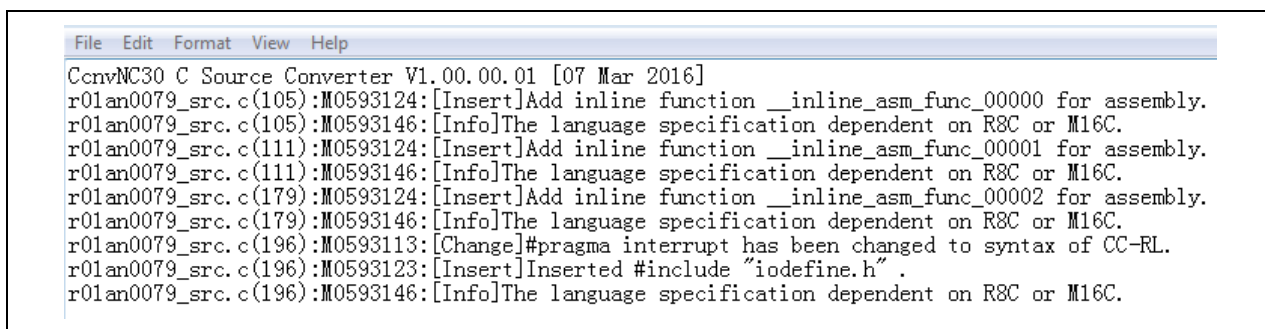


图 4.3 具体的转换结果 (将 RTC 用作时钟的运行)

从 R8C 转至 RL78 的迁移指南 (CcnvNC30)

(3) 修改转换后的 C 源文件。

有时会出现中断函数声明重复的情况。这种情况下 CC-RL 会发生错误，因此要删掉转换来的 #pragma 指令。

```
240 // [CcnvNC30] #pragma interrupt/B _timer_re(vect=10)
241 #pragma interrupt _timer_re(vect=10, bank=RB1) ← 删除 #pragma 指令。
242 void _timer_re(void)
243 {
244     sec = trespsec & 0x7f; /* Set second to RAM */
245     min = trespmin & 0x7f; /* Set minute to RAM */
246     hr = trespshr & 0x3f; /* Set hour to RAM */
247     wk_old = wk; /* Set last-time value */
248     wk = trespwk & 0x07; /* Set day to RAM */
249     up_flg = UPDATE; /* Set update flag */
250 }
```

图 4.4 更改中断函数的声明

4.1.2 程序的自动生成

- (1) 通过集成开发环境 CS+或者 e2studio 创建一个新的工程。
- (2) 利用代码生成工具设置各个功能。

CPU 时钟设为 16MHz 的高速内部振荡器。

RTC 及间隔定时器运行时钟设为 32.768kHz。

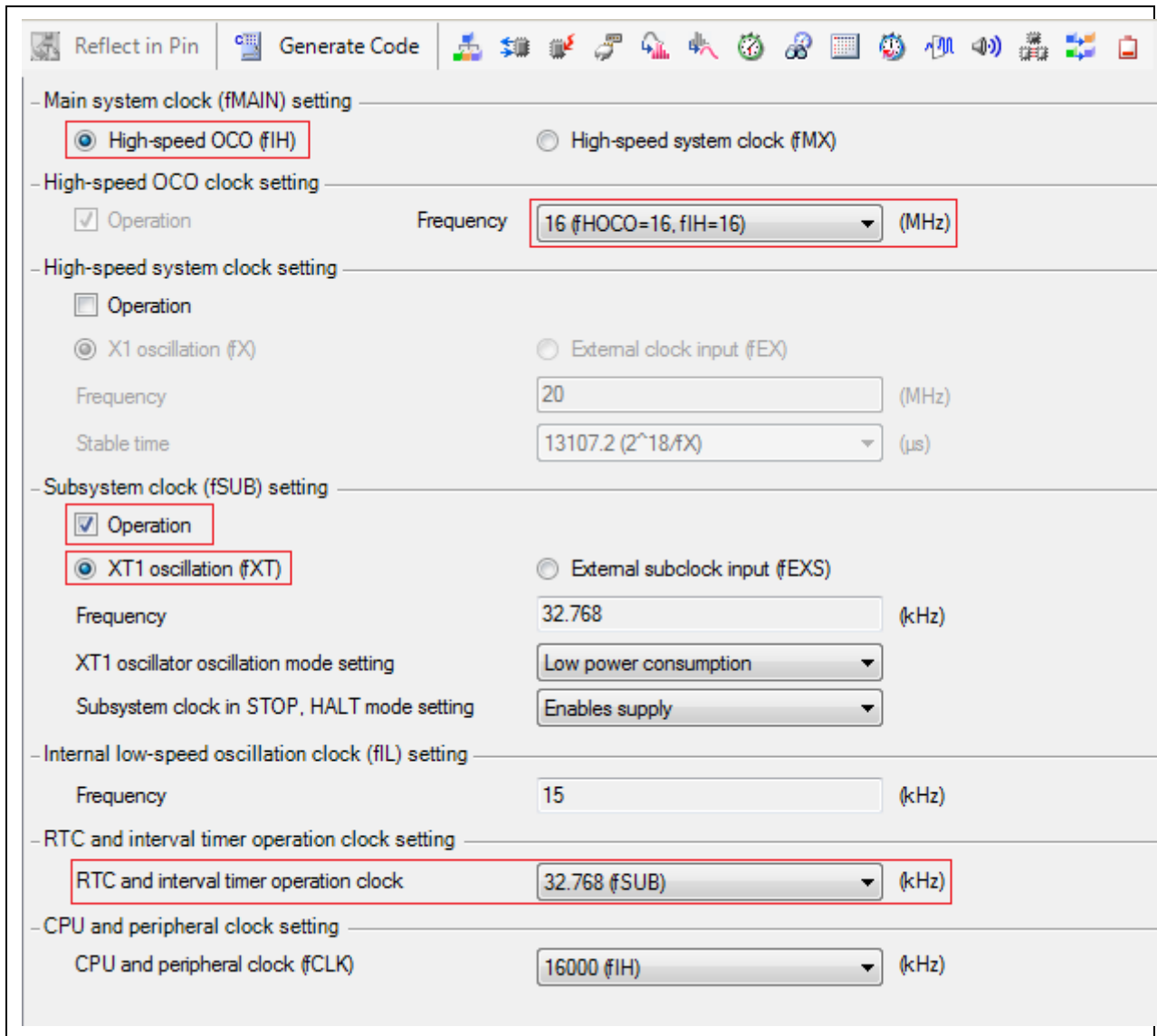


图 4.5 代码生成工具的设置界面（时钟）

从 R8C 转至 RL78 的迁移指南 (CcnvNC30)

对等同于 R8C 族的定时器 RE 的实时时钟功能进行设置。

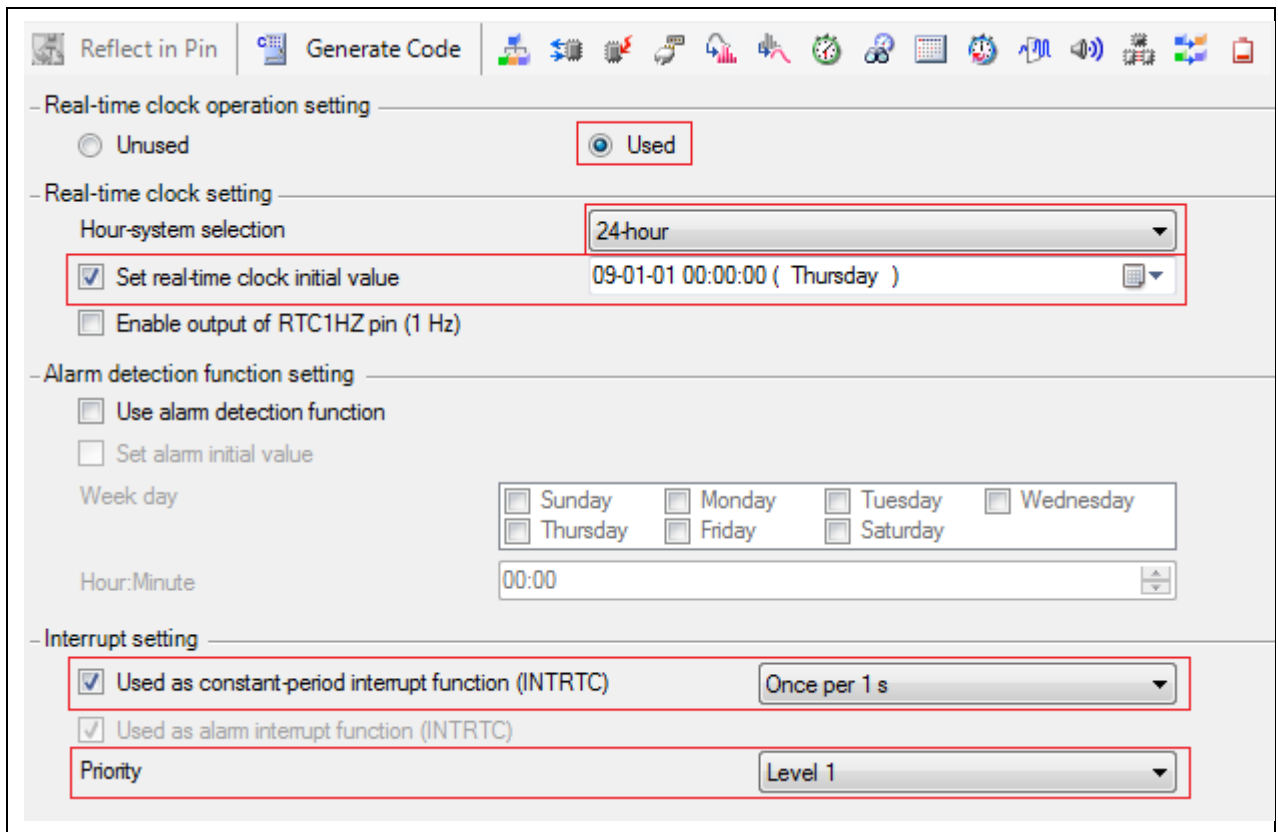


图 4.6 代码生成工具的设置界面（实时时钟）

- (3) 设置“Port Functions”、“Watchdog Timer”以及“Voltage Detector”。
- (4) 点击“Generate Code”以生成文件。

4.1.3 程序的添加

本节介绍在代码生成的程序上添加符号定义和主函数处理的内容。关于其他程序（如时钟设置、RTC 功能设置等），请使用由代码生成的程序。

- 符号定义
在 r_cg_rtc.h 上添加符号定义。

```
R8C 的程序
94  /******
95  /* DEFINE
96  /******
97  #define NO_UPDATE      0          /* No update */
98  #define UPDATE         1          /* Update */
99  #define COMM           0          /* Common year */
100 #define LEAP           1          /* Leap year */
101 #define DEC            0x12       /* December */
102 #define WEEK           0x04       /* A day of the week(thursday) */

RL78/G14 的 r_cg_rtc.h 文件
146  /* Start user code for function. Do not edit comment generated here */
147  #define NO_UPDATE      0          /* No update */
148  #define UPDATE         1          /* Update */
149  #define COMM           0          /* Common year */
150  #define LEAP           1          /* Leap year */
151  #define DEC            0x12       /* December */
```

图 4.7 符号定义的置换

从 R8C 转至 RL78 的迁移指南 (CcnvNC30)

- 主函数

使用了 RL78/G14 的代码生成工具后，在执行 main 函数之前会先执行 R_Systeminit 函数。通过 R_Systeminit 函数对时钟和 RTC 进行初始设置。因此只需对红框内的处理进行程序的手动添加。通过 R_RTC_Start 函数启动 RTC 的运行。

使用 R8C/35A 时，用户可将定时器 RE 用作实时时钟。使用 RL78/G14 时，用户可将 RTC 用作实时时钟。在“R8C/35A Group Clock Operation Using RTC (R01AN0079E)”的主函数中，定时器 RE 的初始化函数和开始指令被执行。在“r01an3508_rl78g14_rtc”的主函数中，RTC 的初始化函数（RTC 每秒产生一个中断）和开始函数被执行。

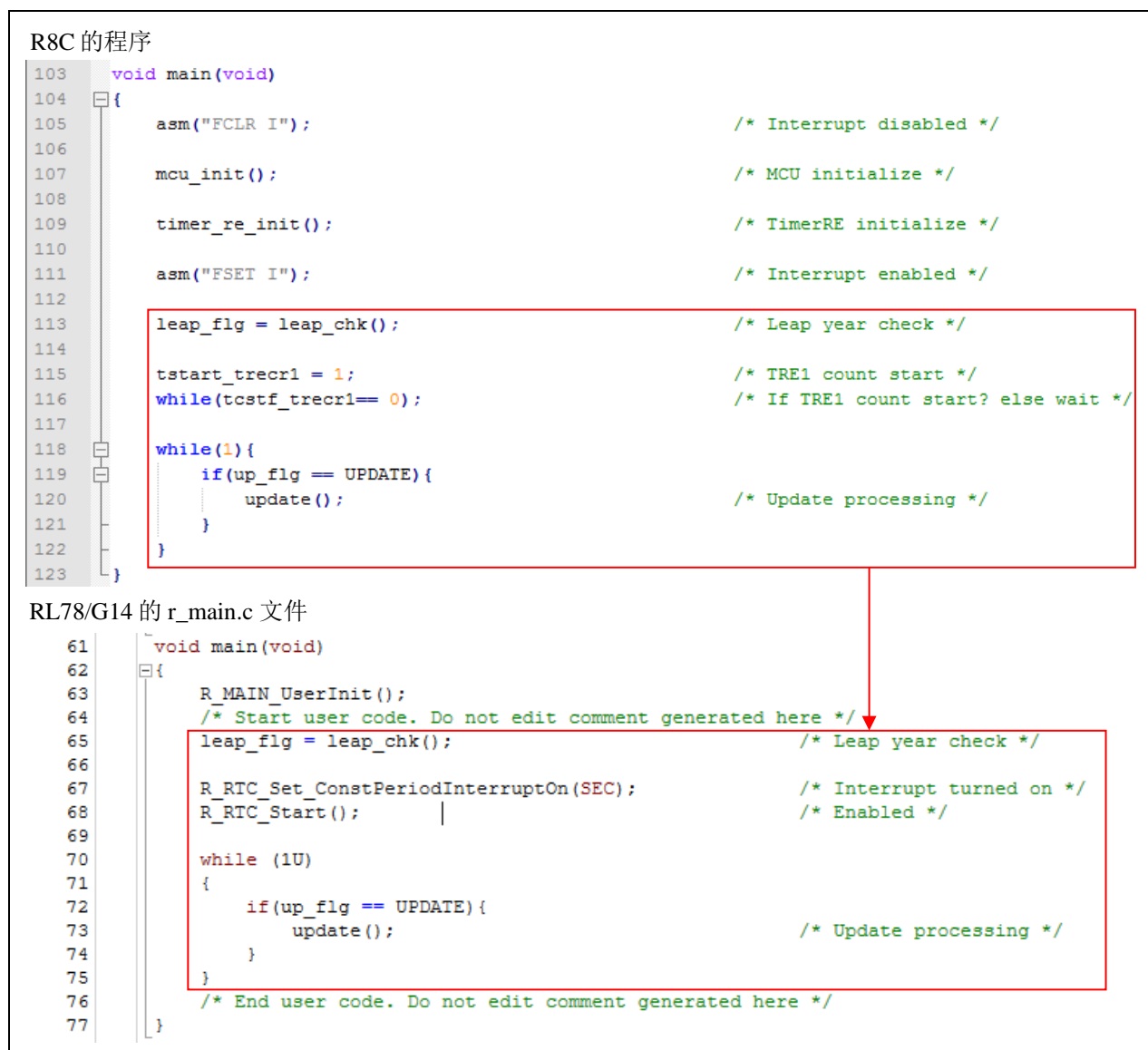


图 4.8 主函数的置换

从 R8C 转至 RL78 的迁移指南 (CcnvNC30)

另外，“main”函数中用到的“update”和“leap_chk”函数需添加到 r_main.c 文件中。在这两个函数中用到的变量也需要添加到 r_main.c 文件中。

R8C 的变量

```
63  /*.....*/
64  /* RAM */
65  /*.....*/
66  unsigned short year = 0x2009; /* Year */
67  unsigned char month = 0x01; /* Month */
68  unsigned char day = 0x01; /* Day */
69  unsigned char wk = WEEK; /* Sun,Mon,Tue,Wed,Thu,Fri,Sat */
70  unsigned char hr = 0x00; /* Hour */
71  unsigned char min = 0x00; /* Minute */
72  unsigned char sec = 0x00; /* Second */
73
74  /* Work area */
75  unsigned char wk_old = 0x00; /* Last-time value of 'wk' */
76
77  /* Flags */
78  unsigned char up_flg = NO_UPDATE; /* Update flag */
79  unsigned char leap_flg = COMM; /* Leap flag */
80
81  /*.....*/
82  /* ROM */
83  /*.....*/
84  unsigned char const MONTH_DAYS[13] = { /* This table contains the number of days in different months */
85     0x00,0x31,0x28,0x31,0x30,0x31,0x30,0x31,0x31,0x30,0x31,0x30,0x31
86  };
87
88  unsigned char const MONTH_DAYS_L[13] = { /* This table contains the number of days in different months (leap year) */
89     0x00,0x31,0x29,0x31,0x30,0x31,0x30,0x31,0x31,0x30,0x31,0x30,0x31
90  };
```

RL78/G14 的 r_main.c 文件

```
46  /*.....*/
47  Global variables and functions
48  /*.....*/
49  /* Start user code for global. Do not edit comment generated here */
50  unsigned short year = 0x2009; /* Year */
51  unsigned char month = 0x01; /* Month */
52  unsigned char day = 0x01; /* Day */
53  unsigned char wk = 0x04; /* Sun,Mon,Tue,Wk,Thu,Fri,Sat */
54  unsigned char hr = 0x00; /* Hour */
55  unsigned char min = 0x00; /* Minute */
56  unsigned char sec = 0x00; /* Second */
57
58  unsigned char wk_old = 0x00; /* Last-time value of 'wk' */
59
60  unsigned char up_flg = NO_UPDATE; /* Update flag */
61  unsigned char leap_flg = COMM; /* Leap flag */
62
63  unsigned char const MONTH_DAYS[13] = { /* This table contains the number of days in different months */
64     0x00,0x31,0x28,0x31,0x30,0x31,0x30,0x31,0x31,0x30,0x31,0x30,0x31
65  };
66
67  unsigned char const MONTH_DAYS_L[13] = { /* This table contains the number of days in different months (leap year) */
68     0x00,0x31,0x29,0x31,0x30,0x31,0x30,0x31,0x31,0x30,0x31,0x30,0x31
69  };
70
```

图 4.9 添加变量

R8C 的程序

```

207  /*"FUNC COMMENT"*****
208  * Outline           : Update processing
209  * Declaration       : void update(void)
210  * Description       : Update check
211  * Argument          : none
212  * Return Value      : none
213  /*"FUNC COMMENT END"*****/
214  void update(void)
215  {
216      unsigned char  num = 0;
217
218      up_flg = NO_UPDATE;           /* Update flag clear */
219
220      if(wk != wk_old){           /* Change day ? */
258  }
259
260  /*"FUNC COMMENT"*****
261  * Outline           : Leap year check processing
262  * Declaration       : unsigned char leap_chk(void)
263  * Description       : Leap year check
264  * Argument          : none
265  * Return Value      : none
266  /*"FUNC COMMENT END"*****/
267  unsigned char leap_chk(void)
268  {
269      unsigned char  chk = 0;           /* Result of check */
270      unsigned short dec = 0;
271      unsigned short work = 0;
272
273      work = year;                 /* Copy year */
274
275      dec = (((work & 0xF000) >> 12) * 1000); /* Conversion to decimal */
276      dec += (((work & 0x0F00) >> 8) * 100);
277      dec += (((work & 0x00F0) >> 4) * 10);
278      dec += (((work & 0x000F) >> 0) * 1);
279
280      chk = COMM;                 /* Set Common year */
281
282      if((dec % 4) == 0){
288          return(chk);
289  }

```

RL78/G14 的 r_main.c 文件

```

113  /* Start user code for adding. Do not edit comment generated here */
114
115  * Function Name: update
116  * Description  : Update check
117  * Arguments   : None
118  * Return Value: None
119  *****
120  void update(void)
121  #[ ... ]
165
166  * Function Name: leap_chk
167  * Description  : Leap year check
168  * Arguments   : None
169  * Return Value: None
170  *****
171  unsigned char leap_chk(void)
172  #[ ... ]
173
195  /* End user code. Do not edit comment generated here */

```

图 4.10 添加函数

4.1.4 其他变更事项

如果通过代码生成器设置了 RTC，则自动生成中断处理。需将用户程序添加至中断程序中。

R8C 的程序

```

189  /*"FUNC COMMENT"*****
190  * Outline           : Timer RE interrupt processing
191  * Declaration      : void_timer_re(void)
192  * Description      : Timer RE interrupt
193  * Argument         : none
194  * Return Value     : none
195  *"FUNC COMMENT END"*****/
196  #pragma interrupt/B _timer_re(vect=10)
197  void_timer_re(void)
198  {
199      sec = trespsec & 0x7f;           /* Set second to RAM */
200      min = tremin & 0x7f;           /* Set minute to RAM */
201      hr = trehr & 0x3f;             /* Set hour to RAM */
202      wk_old = wk;                   /* Set last-time value */
203      wk = trewk & 0x07;             /* Set day to RAM */
204      up_flg = UPDATE;               /* Set update flag */
205  }
                
```

RL78/G14 的 r_cg_rtc_user.c 文件

```

62  /*"*****
63  * Function Name: r_rtc_interrupt
64  * Description  : This function is INTRTC interrupt service routine.
65  * Arguments   : None
66  * Return Value: None
67  *"*****/
68  static void __near r_rtc_interrupt(void)
69  {
70      if (1U == RIFG)
71      {
72          RTCC1 &= (uint8_t)~_08_RTC_INTC_GENERATE_FLAG; /* clear RIFG */
73          r_rtc_callback_constperiod();
74      }
75  }
76
77  /*"*****
78  * Function Name: r_rtc_callback_constperiod
79  * Description   : This function is real-time clock constant-period interrupt service handler.
80  * Arguments    : None
81  * Return Value  : None
82  *"*****/
83  static void r_rtc_callback_constperiod(void)
84  {
85      /* Start user code. Do not edit comment generated here */
86      sec = SEC & 0x7f;           /* Set second to RAM */
87      min = MIN & 0x7f;           /* Set minute to RAM */
88      hr = HOUR & 0x3f;          /* Set hour to RAM */
89      wk_old = wk;                /* Set last-time value */
90      wk = WEEK & 0x07;          /* Set day to RAM */
91      up_flg = UPDATE;           /* Set update flag */
92      /* End user code. Do not edit comment generated here */
93  }
                
```

图 4.11 添加中断程序

另外，这些变量需要在 r_cg_rtc_user.c 文件中添加外部声明。

```

48  /* Start user code for global. Do not edit comment generated here */
49  extern unsigned short year;           /* Year */
50  extern unsigned char month;          /* Month */
51  extern unsigned char day;            /* Day */
52  extern unsigned char wk;             /* Sun, Mon, Tue, Wed, Thu, Fri, Sat */
53  extern unsigned char hr;             /* Hour */
54  extern unsigned char min;            /* Minute */
55  extern unsigned char sec;            /* Second */
56
57  extern unsigned char wk_old;         /* Last-time value of 'wk' */
58
59  extern unsigned char up_flg;
60  /* End user code. Do not edit comment generated here */
                
```

图 4.12 添加外部声明

4.1.5 置换后的参考例程

请从瑞萨电子官网下载“an-r01an3508cc0100-rl78-migrate.zip”参考例程。

“workspace”文件夹中的“rl78g14_migrate_rtc”为置换“R8C/35A Group Clock Operation Using RTC (R01AN0079E)”的参考例程。

4.2 R8C 的参考例程 (DTC 运行)

“R8C/35C Group DTC Operation in Chain Transfers (R01AN0372E)” 中包含的程序被替换为 RL78/G14 的程序。替换后的程序文件为 “r01an3508_rl78g14_dtc”。

该程序使用 DTC (链传送)。

4.2.1 使用 CcnvNC30 为 CC-RL 导入源文件

- (1) 创建编目文件以指定一个要转换的 C 源文件。

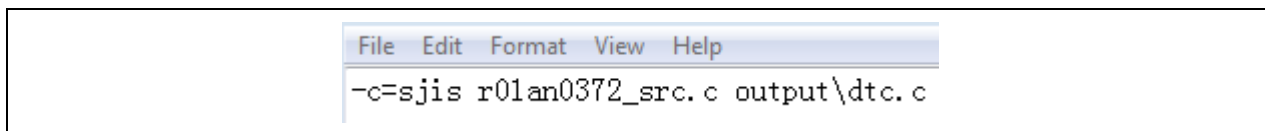


图 4.13 编目文件记述例 (链传送中的 DTC 运行)

- (2) 启动命令提示符 (Command Prompt)，对编目文件指定的 C 源文件进行转换。输出的转换结果文件中标有变更处。

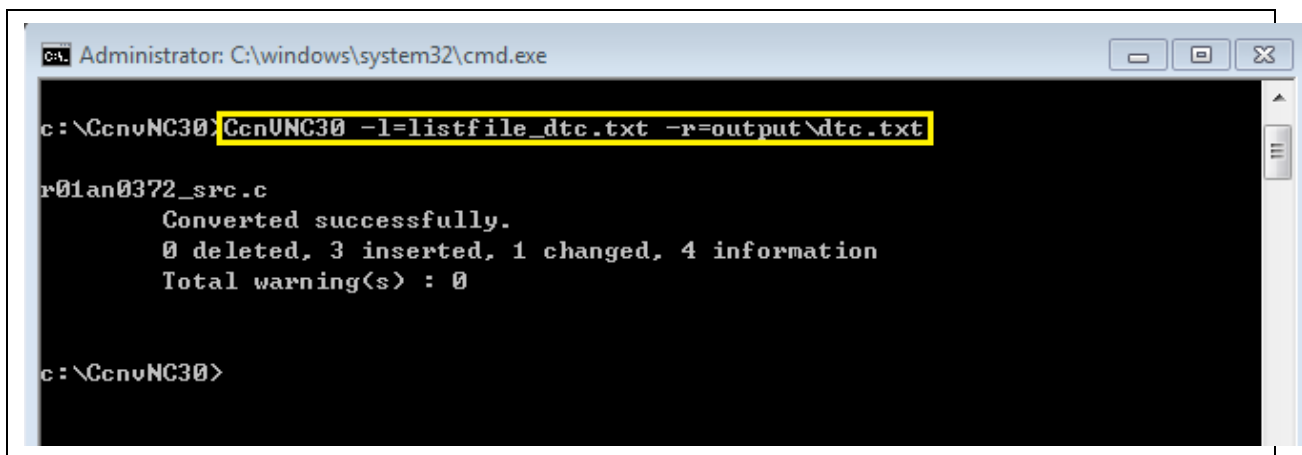


图 4.14 CcnvNC30 执行界面 (链传送中的 DTC 运行)

转换结果文件中记载着转换结果 (如下图所示)。关于具体的转换结果, 请参照 “CcnvNC30 C Source Converter User's Manual (R20UT3685E)”。

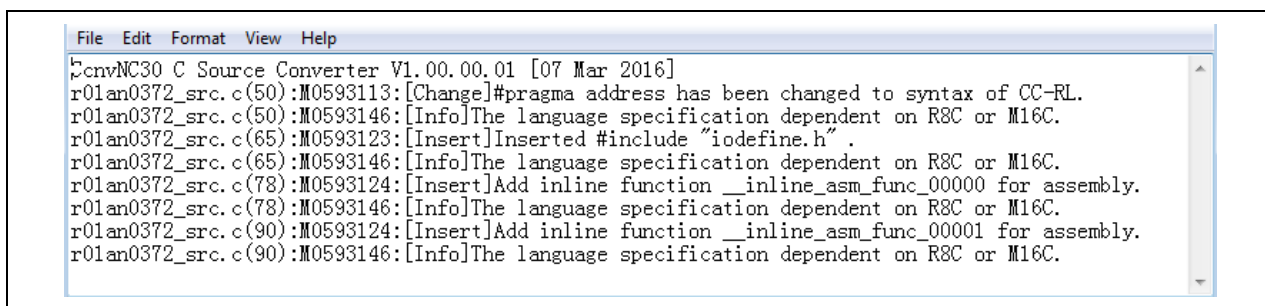


图 4.15 具体的转换结果 (链传送中的 DTC 运行)

从 R8C 转至 RL78 的迁移指南 (CcnvNC30)

(3) 修改转换后的 C 源文件。

有时会出现中断函数声明重复的情况。这种情况下 CC-RL 会发生错误，因此要删掉转换来的 #pragma 指令。

```
86 // [CcnvNC30] #pragma ADDRESS ad_value 00600H /* Destination address of A/D conversion data */
87 #pragma address ad_value=0x00600 ← 删除 #pragma 指令。
```

图 4.16 更改中断函数的声明

4.2.2 程序的自动生成

- (1) 通过集成开发环境 CS+ 或者 e2studio 创建一个新的工程。
- (2) 利用代码生成工具设置各个功能。
CPU 时钟设为 16MHz 的高速内部振荡器时钟。

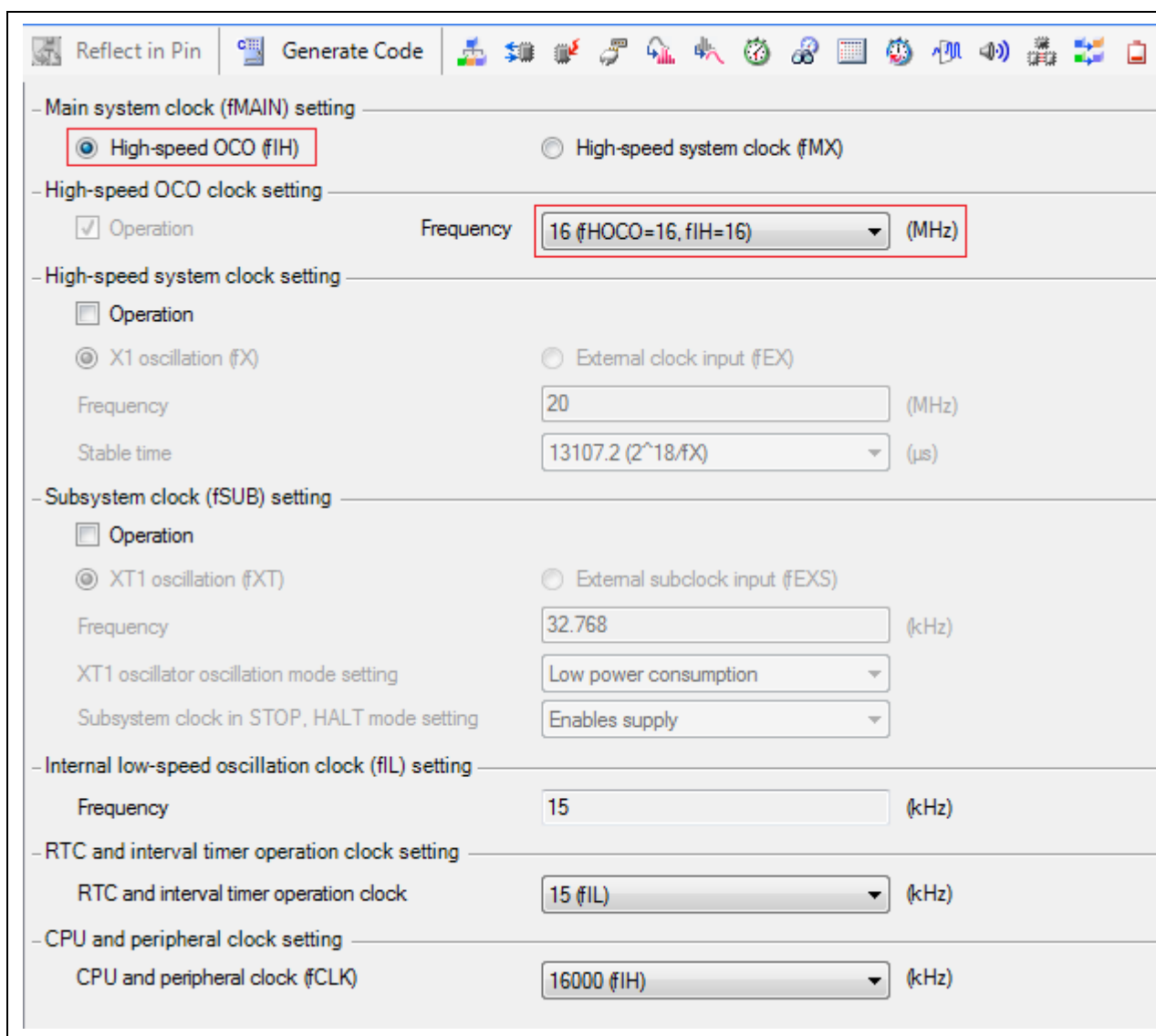


图 4.17 代码生成工具的设置界面 (时钟)

从 R8C 转至 RL78 的迁移指南 (CcnvNC30)

对等同于 R8C 族 ADC 的 A/D 转换器进行设置。
设置 A/D 转换器：软件触发模式（扫描模式、单次转换模式）。

Reflect in Pin | Generate Code

- A/D convertor operation setting
 Unused Used

- Comparator operation setting
 Stop Operation

- Resolution setting
 10 bits 8 bits

- VREF(+) setting
 VDD AVREFP Internal reference voltage

- VREF(-) setting
 VSS AVREFM

- Trigger mode setting
 Software trigger mode
 Hardware trigger no wait mode
 Hardware trigger wait mode
INTTM01

- Operation mode setting
 Continuous select mode Continuous scan mode
 One-shot select mode One-shot scan mode
ANI0 - ANI14 analog input selection ANI0 - ANI3
ANI16 - ANI20 analog input selection
 ANI16 ANI17 ANI18 ANI19
 ANI20
A/D channel selection ANI0 - ANI3

- Conversion time setting
Conversion time mode Normal 1
Conversion time 30.4 (608/fCLK) (μs)

- Conversion result upper/lower bound value setting
 Generates an interrupt request (INTAD) when $ADLL \leq ADCRH \leq ADUL$
 Generates an interrupt request (INTAD) when $ADUL < ADCRH$ or $ADLL > ADCRH$
Upper bound (ADUL) value 255
Lower bound (ADLL) value 0

- Interrupt setting
 Use A/D interrupt (INTAD)
Priority High

图 4.18 代码生成工具的设置界面 (ADC)

从 R8C 转至 RL78 的迁移指南 (CcnvNC30)

对等同于 R8C 族 DTC 的数据传送控制器 (DTC) 进行设置。

“R8C/35C Group DTC Operation in Chain Transfers (R01AN0372E)” 的参考例程中，通过 DTC 链传送，存在 A/D 寄存器 (AD0 ~ AD3) 中的 A/D 转换值被传送至内部 RAM 中。然而 RL78/G14 只有一个 A/D 转换值寄存器，因此使用 DTC (使用通常模式，不使用链传送)。在扫描模式下，对输入到 ANI0 ~ ANI3 引脚的模拟输入电压运行 A/D 转换，并使用 DTC 传送将 A/D 转换值存到 RAM 中。连续对每个引脚进行 A/D 转换。每对一个引脚进行完 A/D 转换，转换值就被存到 10 位 A/D 转换值寄存器 (ADCR) 中，并激活 DTC，A/D 转换值从 ADCR 传送至 RAM 中。当所有引脚都完成了 A/D 转换和 DTC 传送，会产生一个 A/D 转换结束中断请求。

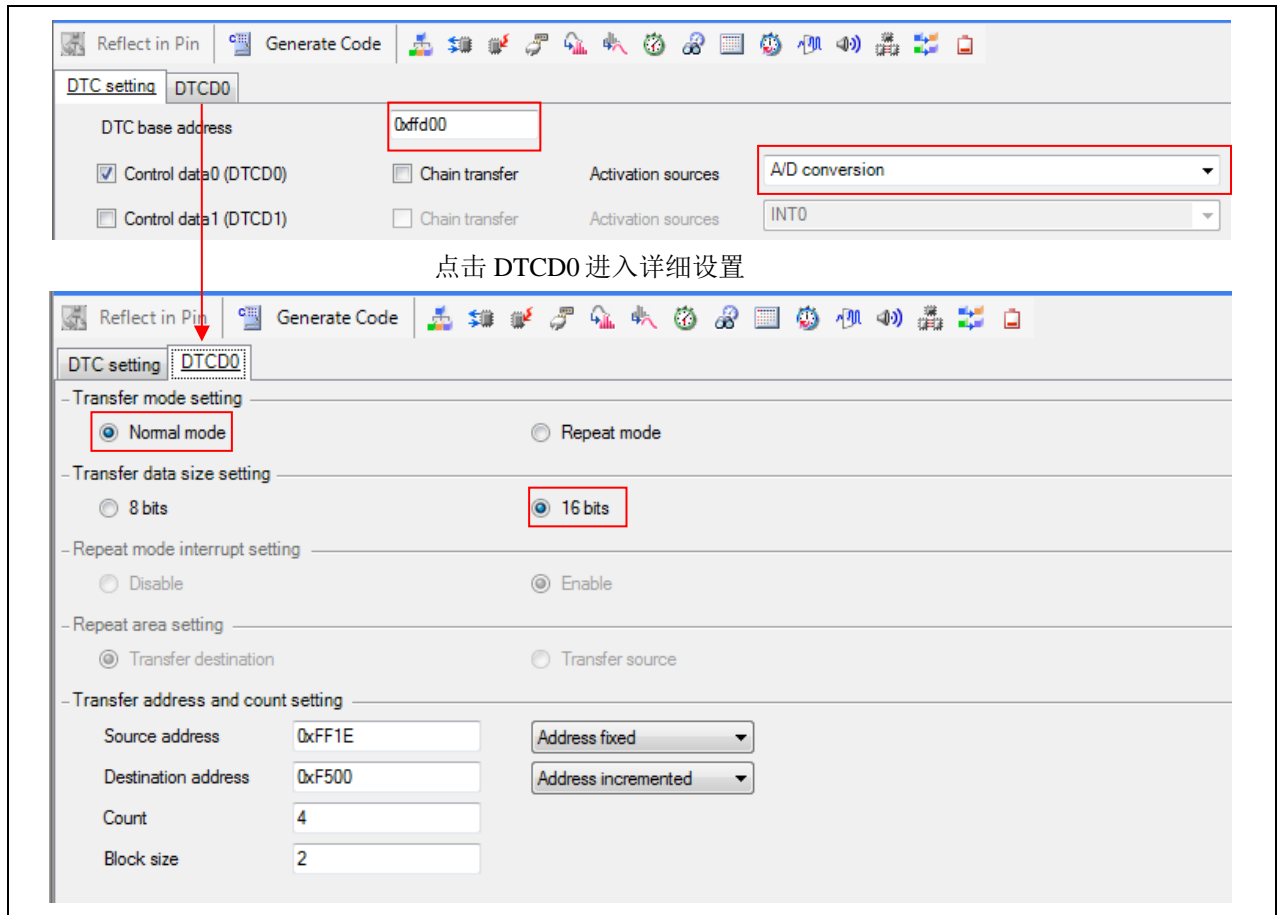


图 4.19 代码生成工具的设置界面 (DTC)

- (3) “Port Functions”、“Watchdog Timer”以及“Voltage Detector”。
- (4) 点击“Generate Code”以生成文件。

4.2.3 程序的添加

本节介绍在代码生成的程序上添加符号定义、主函数和中断函数处理的内容。关于其他程序（如时钟设置、ADC 功能设置以及 DTC 功能设置等），请使用由代码生成的程序。

- 主函数

使用了 RL78/G14 的代码生成工具后，在执行 main 函数之前会先执行 R_Systeminit 函数。通过 R_Systeminit 函数对时钟、ADC 和 DTC 进行初始设置。因此只需对红框内的处理进行程序的手动添加。通过 R_DTCD0_Start 函数启动 DTC 的运行，通过 R_ADC_Start 函数启动 ADC 的运行。

```
R8C 的程序
75 void main(void)
76 {
77     /* ==== Interrupt disabled ==== */
78     asm("FCLR I");
79
80     /* ==== Set High-speed on-chip oscillator clock to System clock ==== */
81     mcu_init();
82
83     /* ==== A/D converter initialize ==== */
84     ad_converter_enable();
85
86     /* ==== DTC initialize ==== */
87     dtc_enable();
88
89     /* ==== Interrupt enabled ==== */
90     asm("FSET I");
91
92     dtcen16 = 1; /* Activation of A/D interruption enable */
93
94     adst = 1; /* A/D conversion starts */
95
96     /* ==== Main loop ==== */
97     while(1){
98     }
99 }
```

```
RL78/G14 的 r_main.c 文件
59 void main(void)
60 {
61     R_MAIN_UserInit();
62     /* Start user code. Do not edit comment generated here */
63     R_DTCD0_Start(); /* DTC enabled */
64     R_ADC_Set_OperationOn();
65     R_ADC_Start(); /* A/D converter enabled */
66
67     while (1U)
68     {
69         NOP();
70     }
71     /* End user code. Do not edit comment generated here */
72 }
```

图 4.20 主函数的置换

如果“while (1U)”指令中没有命令，即便产生了 DTC 传送请求，DTC 传送也会处于等待状态。因此需在“while (1U)”中添加“NOP()”命令。

4.2.4 其他变更事项

如果通过代码生成器设置了 ADC，则自动生成中断处理。需将用户程序添加至中断程序中。

R8C 的变量

```
49  /* **** Global Variables **** */
50  #pragma ADDRESS ad_value 00600H /* Destination address of A/D conversion data */
51  unsigned short ad_value[4]; /* A/D data from AN8 to AN11 addressed from */
52  /* 0x0600 to 0x0607 */
53
54  unsigned short an8_value; /* AN8 value */
55  unsigned short an9_value; /* AN9 value */
56  unsigned short an10_value; /* AN10 value */
57  unsigned short an11_value; /* AN11 value */
```

RL78/G14 的 r_cg_adc_user.c 文件

```
37  #pragma address ad_value=0xFF500
38  unsigned short ad_value[4]; /* A/D data from ANI0 to ANI3 addressed from */
39  /* 0xFF500 to 0xFF507 */
40
41  unsigned short an0_value; /* ANI0 value */
42  unsigned short an1_value; /* ANI1 value */
43  unsigned short an2_value; /* ANI2 value */
44  unsigned short an3_value; /* ANI3 value */
```

图 4.21 为中断程序添加变量

```

R8C 的程序
301 void ad_converter_int(void)
302 {
303     /* ==== Setting A/D conversion data ==== */
304     an8_value = (ad_value[0]&0x03FF); /* Setting AN8 value */
305     an9_value = (ad_value[1]&0x03FF); /* Setting AN9 value */
306     an10_value = (ad_value[2]&0x03FF); /* Setting AN10 value */
307     an11_value = (ad_value[3]&0x03FF); /* Setting AN11 value */
308
309     /* ==== A/D conversion starts ==== */
310     dtcct0 = 1; /* One-time is set to transfer */
311     dtcct1 = 1; /* One-time is set to transfer */
312     dtcen16 = 1; /* Activation of A/D interruption enable */
313     adst = 1; /* A/D conversion starts */
314 }

RL78/G14 的 r_cg_adc_user.c 文件
66 static void __near r_adc_interrupt(void)
67 {
68     /* Start user code. Do not edit comment generated here */
69     an0_value = (ad_value[0U] & 0xFFCOU) >> 6U;
70     an1_value = (ad_value[1U] & 0xFFCOU) >> 6U;
71     an2_value = (ad_value[2U] & 0xFFCOU) >> 6U;
72     an3_value = (ad_value[3U] & 0xFFCOU) >> 6U;
73     /* ==== A/D conversion starts ==== */
74     R_DTC_Create();
75     R_DTCDO_Start(); /* DTC enabled */
76     R_ADC_Start(); /* A/D converter enabled */
77     /* End user code. Do not edit comment generated here */
78 }
    
```

图 4.22 添加中断程序

4.2.5 置换后的参考例程

请从瑞萨电子官网下载“an-r01an3508cc0100-rl78-migrate.zip”参考例程。

“workspace”文件夹中的“rl78g14_migrate_dtc”为置换“R8C/35C Group DTC Operation in Chain Transfers (R01AN0372E)”的参考例程。

4.3 参考例程的动作确认条件

置换后的参考例程是在下面的条件下进行动作确认的。

表 4.1 动作确认条件

项目	内容
所用微控制器	RL78/G14 (R5F104PJ)
集成开发环境 (CS+)	CS+ for CC V4.01.00 (瑞萨电子开发)
C 编译器 (CS+)	CC-RL V1.03.00 (瑞萨电子开发)
集成开发环境 (e ² studio)	e ² studio V5.2.0.020 (瑞萨电子开发)
C 编译器 (e ² studio)	CC-RL V1.03.00 (瑞萨电子开发)
所用电路板	瑞萨自制

5. 参考例程

参考例程请从瑞萨电子网页上取得。

6. 参考文献

用户手册

RL78 family User's Manual: Software (R01US0015E)
RL78 CC-RL Compiler User's Manual (R20UT3123E)
CS+ Code Generator Integrated Development Environment User's Manual: Peripheral Function Operation (R20UT3104E)
CcnvNC30 C Source Converter User's Manual (R20UT3685E)

应用说明

R8C/35A Group Clock Operation Using RTC (R01AN0079E)
R8C/35C Group DTC Operation in Chain Transfers (R01AN0372E)

(最新版本请从瑞萨电子网页上取得)

公司主页和咨询窗口

瑞萨电子主页

- <http://www.renesas.com/zh-cn/>

咨询

- <https://www.renesas.com/zh-cn/support/contact.html>

修订记录

Rev.	发行日	修订内容	
		页	要点
1.00	2017.03	—	初版发行

所有商标及注册商标均归其各自拥有者所有。

产品使用时的注意事项

本文对适用于单片机所有产品的“使用时的注意事项”进行说明。有关个别的使用时的注意事项请参照正文。此外，如果在记载上有与本手册的正文有差异之处，请以正文为准。

1. 未使用的引脚的处理

【注意】将未使用的引脚按照正文的“未使用引脚的处理”进行处理。

CMOS产品的输入引脚的阻抗一般为高阻抗。如果在开路的状态下运行未使用的引脚，由于感应现象，外加LSI周围的噪声，在LSI内部产生穿透电流，有可能被误认为是输入信号而引起误动作。未使用的引脚，请按照正文的“未使用引脚的处理”中的指示进行处理。

2. 通电时的处理

【注意】通电时产品处于不定状态。

通电时，LSI内部电路处于不确定状态，寄存器的设定和各引脚的状态不定。通过外部复位引脚对产品进行复位时，从通电到复位有效之前的期间，不能保证引脚的状态。

同样，使用内部上电复位功能对产品进行复位时，从通电到达到复位产生的一定电压的期间，不能保证引脚的状态。

3. 禁止存取保留地址（保留区）

【注意】禁止存取保留地址（保留区）

在地址区域中，有被分配将来用作功能扩展的保留地址（保留区）。因为无法保证存取这些地址时的运行，所以不能对保留地址（保留区）进行存取。

4. 关于时钟

【注意】复位时，请在时钟稳定后解除复位。

在程序运行中切换时钟时，请在要切换成的时钟稳定之后进行。复位时，在通过使用外部振荡器（或者外部振荡电路）的时钟开始运行的系统中，必须在时钟充分稳定后解除复位。另外，在程序运行中，切换成使用外部振荡器（或者外部振荡电路）的时钟时，在要切换成的时钟充分稳定后再进行切换。

5. 关于产品间的差异

【注意】在变更不同型号的产品时，请对每一个产品型号进行系统评价测试。

即使是同一个群的单片机，如果产品型号不同，由于内部ROM、版本模式等不同，在电特性范围内有时特性值、动作容限、噪声耐量、噪声辐射量等不同。因此，在变更不认同型号的产品时，请对每一个型号的产品进行系统评价测试。

Notice

1. Descriptions of circuits, software and other related information in this document are provided only to illustrate the operation of semiconductor products and application examples. You are fully responsible for the incorporation of these circuits, software, and information in the design of your equipment. Renesas Electronics assumes no responsibility for any losses incurred by you or third parties arising from the use of these circuits, software, or information.
2. Renesas Electronics has used reasonable care in preparing the information included in this document, but Renesas Electronics does not warrant that such information is error free. Renesas Electronics assumes no liability whatsoever for any damages incurred by you resulting from errors in or omissions from the information included herein.
3. Renesas Electronics does not assume any liability for infringement of patents, copyrights, or other intellectual property rights of third parties by or arising from the use of Renesas Electronics products or technical information described in this document. No license, express, implied or otherwise, is granted hereby under any patents, copyrights or other intellectual property rights of Renesas Electronics or others.
4. You should not alter, modify, copy, or otherwise misappropriate any Renesas Electronics product, whether in whole or in part. Renesas Electronics assumes no responsibility for any losses incurred by you or third parties arising from such alteration, modification, copy or otherwise misappropriation of Renesas Electronics product.
5. Renesas Electronics products are classified according to the following two quality grades: "Standard" and "High Quality". The recommended applications for each Renesas Electronics product depends on the product's quality grade, as indicated below.
"Standard": Computers; office equipment; communications equipment; test and measurement equipment; audio and visual equipment; home electronic appliances; machine tools; personal electronic equipment; and industrial robots etc.
"High Quality": Transportation equipment (automobiles, trains, ships, etc.); traffic control systems; anti-disaster systems; anti-crime systems; and safety equipment etc.
Renesas Electronics products are neither intended nor authorized for use in products or systems that may pose a direct threat to human life or bodily injury (artificial life support devices or systems, surgical implantations etc.), or may cause serious property damages (nuclear reactor control systems, military equipment etc.). You must check the quality grade of each Renesas Electronics product before using it in a particular application. You may not use any Renesas Electronics product for any application for which it is not intended. Renesas Electronics shall not be in any way liable for any damages or losses incurred by you or third parties arising from the use of any Renesas Electronics product for which the product is not intended by Renesas Electronics.
6. You should use the Renesas Electronics products described in this document within the range specified by Renesas Electronics, especially with respect to the maximum rating, operating supply voltage range, movement power voltage range, heat radiation characteristics, installation and other product characteristics. Renesas Electronics shall have no liability for malfunctions or damages arising out of the use of Renesas Electronics products beyond such specified ranges.
7. Although Renesas Electronics endeavors to improve the quality and reliability of its products, semiconductor products have specific characteristics such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Further, Renesas Electronics products are not subject to radiation resistance design. Please be sure to implement safety measures to guard them against the possibility of physical injury, and injury or damage caused by fire in the event of the failure of a Renesas Electronics product, such as safety design for hardware and software including but not limited to redundancy, fire control and malfunction prevention, appropriate treatment for aging degradation or any other appropriate measures. Because the evaluation of microcomputer software alone is very difficult, please evaluate the safety of the final products or systems manufactured by you.
8. Please contact a Renesas Electronics sales office for details as to environmental matters such as the environmental compatibility of each Renesas Electronics product. Please use Renesas Electronics products in compliance with all applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive. Renesas Electronics assumes no liability for damages or losses occurring as a result of your noncompliance with applicable laws and regulations.
9. Renesas Electronics products and technology may not be used for or incorporated into any products or systems whose manufacture, use, or sale is prohibited under any applicable domestic or foreign laws or regulations. You should not use Renesas Electronics products or technology described in this document for any purpose relating to military applications or use by the military, including but not limited to the development of weapons of mass destruction. When exporting the Renesas Electronics products or technology described in this document, you should comply with the applicable export control laws and regulations and follow the procedures required by such laws and regulations.
10. It is the responsibility of the buyer or distributor of Renesas Electronics products, who distributes, disposes of, or otherwise places the product with a third party, to notify such third party in advance of the contents and conditions set forth in this document, Renesas Electronics assumes no responsibility for any losses incurred by you or third parties as a result of unauthorized use of Renesas Electronics products.
11. This document may not be reproduced or duplicated in any form, in whole or in part, without prior written consent of Renesas Electronics.
12. Please contact a Renesas Electronics sales office if you have any questions regarding the information contained in this document or Renesas Electronics products, or if you have any other inquiries.
(Note 1) "Renesas Electronics" as used in this document means Renesas Electronics Corporation and also includes its majority-owned subsidiaries.
(Note 2) "Renesas Electronics product(s)" means any product developed or manufactured by or for Renesas Electronics.

以下"注意事项"为从英语原稿翻译的中文译文, 仅作参考译文, 英文版的"Notice"具有正式效力。

注意事项

1. 本文件中记载的关于电路、软件和其他相关信息仅用于说明半导体产品的操作和应用实例。用户如在设备设计中应用本文件中的电路、软件和相关信息, 请自行负责。对于用户或第三方因使用上述电路、软件或信息而遭受的任何损失, 瑞萨电子不承担任何责任。
2. 在准备本文件所记载的信息的过程中, 瑞萨电子已尽量做到合理注意, 但是, 瑞萨电子并不保证这些信息都是准确无误的。用户因本文件中记载的信息的错误或遗漏而遭受的任何损失, 瑞萨电子不承担任何责任。
3. 对于因使用本文件中的瑞萨电子产品或技术信息而造成的侵权行为或因此而侵犯第三方的专利、版权或其他知识产权的行为, 瑞萨电子不承担任何责任。本文件所记载的内容不应视为对瑞萨电子或其他人所有的专利、版权或其他知识产权作出任何明示、默示或其它方式的许可及授权。
4. 用户不得更改、修改、复制或其他方式部分或全部地非法使用瑞萨电子的任何产品。对于用户或第三方因上述更改、修改、复制或以其他方式非法使用瑞萨电子产品的行为而遭受的任何损失, 瑞萨电子不承担任何责任。
5. 瑞萨电子产品根据其质量等级分为两个等级: "标准等级"和"高质量等级"。每种瑞萨电子产品的推荐用途均取决于产品的质量等级, 如下所示:
标准等级: 计算机、办公设备、通讯设备、测试和测量设备、视听设备、家用电器、机械工具、个人电子设备以及工业机器人等。
高质量等级: 运输设备(汽车、火车、轮船等)、交通控制系统、防灾系统、预防犯罪系统以及安全设备等。
瑞萨电子产品无意用于且未被授权用于可能对人类生命造成直接威胁的产品或系统以及可能造成人身伤害的产品或系统(人工生命维持装置或系统、植入体内的装置等)中, 或者可能造成重大财产损失的产品或系统(核反应堆控制系统、军用设备等)中。在将每种瑞萨电子产品用于某种特定应用之前, 用户应先确认其质量等级。不得将瑞萨电子产品用于超出其设计用途之外的任何应用。对于用户或第三方因将瑞萨电子产品用于其设计用途之外而遭受的任何损害或损失, 瑞萨电子不承担任何责任。
6. 使用本文件中记载的瑞萨电子产品时, 应在瑞萨电子指定的范围内, 特别是在最大额定值、电源工作电压范围、移动电源电压范围、热辐射特性、安装条件以及其他产品特性的范围内使用。对于在上述指定范围之外使用瑞萨电子产品而产生的故障或损失, 瑞萨电子不承担任何责任。
7. 虽然瑞萨电子一直致力于提高瑞萨电子产品的质量和可靠性, 但是, 半导体产品有其自身的具体特性, 如一定的故障发生率以及在某些使用条件下会发生故障等。此外, 瑞萨电子产品均未进行防辐射设计, 所以请采取安全保护措施, 以避免当瑞萨电子产品在发生故障而造成火灾时导致人身事故、伤害或损害的事故。例如进行软硬件安全设计(包括但不限于冗余设计、防火控制以及故障预防等)、适当的老化处理或其他适当的措施等。由于难于对微机电系统单独进行评估, 所以请用户自行对最终产品或系统进行安全评估。
8. 关于环境保护方面的详细内容, 例如每种瑞萨电子产品的环境兼容性等, 请与瑞萨电子的营业部门联系。使用瑞萨电子产品时, 请遵守对管制物质的使用或含量进行管理的所有相关法律法规(包括但不限于《欧盟RoHS指令》)。对于因用户未遵守相关法律法规而导致的损害或损失, 瑞萨电子不承担任何责任。
9. 不可将瑞萨电子产品和技术用于或者嵌入日本国内或海外相应的法律法规所禁止生产、使用及销售的任何产品或系统中。也不可对本文件中记载的瑞萨电子产品或技术用于与军事应用或者军事用途有关的目的(如大规模杀伤性武器的开发等)。在本文件中记载的瑞萨电子产品或技术进行出口时, 应当遵守相应的出口管制法律法规, 并按照上述法律法规所规定的程序进行。
10. 向第三方分销或处分产品或者以其他方式将产品置于第三方控制之下的瑞萨电子产品买方或分销商, 有责任事先向上述第三方通知本文件规定的内容和条件; 对于用户或第三方因非法使用瑞萨电子产品而遭受的任何损失, 瑞萨电子不承担任何责任。
11. 在事先未得到瑞萨电子书面认可的情况下, 不得以任何形式部分或全部转载或复制本文件。
12. 如果对本文件所记载的信息或瑞萨电子产品有任何疑问, 或者用户有任何其他疑问, 请向瑞萨电子的营业部门咨询。
(注1) 瑞萨电子: 在本文件中指瑞萨电子株式会社及其控股子公司。
(注2) 瑞萨电子产品: 指瑞萨电子开发或生产的任何产品。



SALES OFFICES

Renesas Electronics Corporation

<http://www.renesas.com>

Refer to "<http://www.renesas.com/>" for the latest and detailed information.

Renesas Electronics America Inc.
2801 Scott Boulevard Santa Clara, CA 95050-2549, U.S.A.
Tel: +1-408-588-6000, Fax: +1-408-588-6130

Renesas Electronics Canada Limited
3251 Yonge Street, Suite 8309 Richmond Hill, Ontario Canada L4C 9T3
Tel: +1-905-237-2004

Renesas Electronics Europe Limited
Dukes Meadow, Millboard Road, Bourne End, Buckinghamshire, SL8 5FH, UK
Tel: +44-1628-585-100, Fax: +44-1628-585-900

Renesas Electronics Europe GmbH
Arcadiastrasse 10, 40472 Düsseldorf, Germany
Tel: +49-211-6503-0, Fax: +49-211-6503-1327

Renesas Electronics (China) Co., Ltd.
Room 1709, Quantum Plaza, No.27 ZhiChunLu Haidian District, Beijing 100191, P.R.China
Tel: +86-10-8235-1155, Fax: +86-10-8235-7679

Renesas Electronics (Shanghai) Co., Ltd.
Unit 301, Tower A, Central Towers, 555 Langao Road, Putuo District, Shanghai, P. R. China 200333
Tel: +86-21-2226-0888, Fax: +86-21-2226-0899

Renesas Electronics Hong Kong Limited
Unit 1601-1611, 16/F., Tower 2, Grand Century Place, 193 Prince Edward Road West, Mongkok, Kowloon, Hong Kong
Tel: +852-2265-6988, Fax: +852-2886-9022

Renesas Electronics Taiwan Co., Ltd.
13F, No. 363, Fu Shing North Road, Taipei 10543, Taiwan
Tel: +886-2-8175-9600, Fax: +886-2-8175-9670

Renesas Electronics Singapore Pte. Ltd.
80 Bendemeer Road, Unit #06-02 Hyflux Innovation Centre, Singapore 339949
Tel: +65-6215-0200, Fax: +65-6215-0300

Renesas Electronics Malaysia Sdn.Bhd.
Unit 1207, Block B, Menara Amcorp, Amcorp Trade Centre, No. 18, Jin Persiaran Barat, 46050 Petaling Jaya, Selangor Darul Ehsan, Malaysia
Tel: +60-3-7855-9390, Fax: +60-3-7855-9510

Renesas Electronics India Pvt. Ltd.
No. 777C, 100 Feet Road, HAL, Stage, Indiranagar, Bangalore, India
Tel: +91-80-67208700, Fax: +91-80-67208777

Renesas Electronics Korea Co., Ltd.
12F., 234 Teheran-ro, Gangnam-Gu, Seoul, 135-080, Korea
Tel: +82-2-558-3737, Fax: +82-2-558-5141