

# M16C/26

## C Compiler Startup Files

---

### 1.0 Abstract

The following article describes the startup files for the NC30 and KNC30 C compilers. A set of customized startup files is given for the M30262 version of the M16C/26 microcontroller.

### 2.0 Introduction

The Renesas M30262 is a 16-bit MCU based on the M16C/60 series CPU core. The MCU features include up to 64KB of Flash ROM, 2KB of RAM, and 4KB of virtual EEPROM. The peripheral set includes 10-bit A/D, UARTs, Timers, DMA, and GPIO. The M16C series is ideally suited for programming using the C language.

C compilers for microcontrollers typically require some sort of assembler ‘startup’ file to set processor modes, initialize variables, and so forth. For the NC30 compiler, the startup file also includes section information so the linker knows where, in physical memory, to put variables, constants, code, and so on. The default files included with the NC30 are [ncrt0.a30](#), which is the startup file, and [sect30.inc](#), which contains the section information.

### 3.0 NCRT0.A30

The NCRT0.A30 file is a generic startup, which was written for most of the M16C/60 series microcontrollers. After reset, program execution begins with the code in this startup file. The stack pointer is set to point to a free area in RAM, and the processor mode is set. C requires that all (global) un-initialized variables be set to zero and initialize variables are copied from ROM into RAM.

A customized startup file for the M16C/26 starter kits is described in section 7.1 and referred to as [ncrt0\\_26.a30](#).

### 4.0 SECT30.INC

The sect30.inc file is a generic section file for the M16C series microcontrollers. The purpose of the section definition file is to set the location of C language sections in the microcontroller’s physical memory map. This file contains information that the linker will use to determine where aligned variables (integers), non-aligned variables (characters), code (in ROM), interrupt vectors, and so forth, are to be placed.

A customized section definition file for the M16C/26 starter kits is described in section 7.2 and referred to as [sect30\\_26.inc](#).

At the beginning of the section file are several settings that allow users to customize memory allocations for HEAPSIZE and STACKSIZE. The HEAP memory is used by memory allocation functions. If memory allocation functions are not needed in your application, HEAPSIZE should be set to “00h”. Figure 1 shows an example of the memory map when HEAPSIZE is set to “0100h”. Figure 2 shows an example of the memory map when HEAPSIZE is set to “00h”.

Address(size)	Section	Label:
000000(000400)		
000400(000010)	[D] data_NE	[G] 000400: __pool [G] 000408: __ment
000410(000012)	[D] bss_NE	[G] 000410: _i [G] 000412: _j [G] 000414: _k [G] 000416: __mbase [G] 00041a: __mnext [G] 00041e: __msize
000422(000300)	[D] stack	
000722(000100)	[D] heap	
000822(00efde)		
00f800(0e0800)		
0f0000(000010)	[R] data_NE1	
0f0010(0000cb)	[C] interrupt	[G] 0f0010: start [G] 0f00d8: \$exit [G] 0f00d8: _exit
0f00db(000760)	[C] program	[G] 0f00dc: main [G] 0f013e: \$calloc [G] 0f017e: free [G] 0f01c2: \$malloc [G] 0f021c: \$realloc [G] 0f02a6: \$getmem [G] 0f0464: \$r1smem [G] 0f073e: memcpy [G] 0f0784: \$memset [G] 0f07b8: __i4div
0f083b(00efc5)		
0ff800(0000c0)	[C] vector	
0ff8c0(00071c)		
0ffffdc(000024)	[C] fvector	

Figure 1 Memory Map with HEAP allocated

Address(size)	Section	Label:
000000(000400)		
000400(000006)	[D] bss_NE	[G] 000400: _i [G] 000402: _j [G] 000404: _k
000406(000300)	[D] stack	
000706(00f0fa)		
00f800(0e0800)		
0f0000(0000a9)	[C] interrupt	[G] 0f0000: start [G] 0f00a6: \$exit [G] 0f00a6: _exit
0f00a9(000063)	[C] program	[G] 0f00aa: _main
0f010c(00f6f4)		
0ff800(0000c0)	[C] vector	
0ff8c0(00071c)		
0ffffdc(000024)	[C] fvector	

Figure 2 Memory Map with no HEAP allocated

## 5.0 Default or Custom Startup in Tool Manager

When starting a new project using Tool Manager (or TM) IDE, the project wizard will ask if a customized or a default startup file will be used. If a customized startup file will be used, you can specify the file that you intend to use. If the default startup file is used, the startup files installed under the **C:\MTOOL\SRC30\STARTUP** folder are copied to the new project's working directory. You can replace the startup files that were installed during compiler installation with customized ones as long as you keep the filenames. It is strongly recommended that you back up the default files first before replacing it.

---

**Note:** This assumes that the compiler was installed under the default directory, **C:\MTOOL**. If you have an M16C/26 SKP, the customized startup files for M16C/26 can be found under **C:\MTOOL\MSV30262-SKP\Sample\_Code\Startup\_files** folder.

---

## 6.0 Reference

### Renesas Technology Corporation Semiconductor Home Page

<http://www.renesas.com>

### E-mail Support

[support\\_apl@renesas.com](mailto:support_apl@renesas.com)

### Data Sheets

- M16C/26 datasheet, M30262eds.pdf

### User's Manual

- C Language Programming Manual, 6020EC.PDF

## 7.0 Software Code

The following is a set of customized startup files for the M30262 MCU that is included with the M16C/26 SKP. Except for adding entries into the interrupt vector tables, these files should suffice as-is for most applications. If using different versions of the M16C/26, the ROM starting address will need to be modified.

### 7.1 Customized NCRT0 ([ncrt0\\_26.a30](#))

```

;***** ;
;      C COMPILER for M16C/60, M16C/20
;      Copyright 1995-2003 Renesas Technology America, Inc.
;      All Rights Reserved.
;
;      ncrt0_26.a30 : NC30 startup program
;
;      Content:   Startup assembly file for M16C/26.
;                This program is applicable when using the basic I/O library.
;
;=====
;      $Log:$
;      $Id: ncrt0.a30,v 1.12 2000/05/18 06:44:37 simomura Exp $
;                ncrt0_26.a30, v 1.1 2003/05/15 11:36      R. Seville
;=====
;*****

;-----
; Section allocation and definitions
;-----

        .list OFF
        .include sect30_26.inc
        .list ON

;=====
; Interrupt section start
;-----
;      .insf  start,S,0      ; for stkviewer (see TM and NC30 manuals)
;      .glb   start
;      .section      interrupt

start:
;-----
; After reset, program execution starts here.
; Upon reset, the processor clock (BLCK) defaults to divided by 8 (f/8).
;-----
        ldc    #istack_top, isp      ; set istack pointer

```

```

;-----
; two higher bits (b7 & b6) must be set to be able to use INT4 & INT5
;-----
        mov.b   #0c0h,035Fh           ; set b7 & b6 if application will use INT4 & INT5
        ldc     #data_SE_top, sb      ; set sb register
        ldintb #VECTOR_ADR

;=====
; NEAR area initialize.
;-----
; bss zero clear
;-----
        N_BZERO bss_SE_top,bss_SE
        N_BZERO bss_SO_top,bss_SO
        N_BZERO bss_NE_top,bss_NE
        N_BZERO bss_NO_top,bss_NO

;-----
; initialize data section
;-----
        N_BCOPY data_SEI_top,data_SE_top,data_SE
        N_BCOPY data_SOI_top,data_SO_top,data_SO
        N_BCOPY data_NEI_top,data_NE_top,data_NE
        N_BCOPY data_NOI_top,data_NO_top,data_NO

;=====
; heap area initialization - can be remove if not using memory allocate
; functions
;-----
        .IF     HEAPSIZE!=00h
                HEAPINIT
        .ENDIF

;=====
; Initialize standard I/O
;-----
; SKP sample codes/programs does NOT use this routine but instead call
; an mcu_init() routine from main().
;         .glb   _init
;         .call  _init,G
;         jsr.a  _init

;=====
; Call main() function
;-----
        .glb   _main
        jsr.a  _main

;=====
; exit() function
;-----
        .glb   _exit
        .glb   $exit
_exit:                                     ; End program

```

```

$exit:
    jmp        _exit
    .einsf

;=====
; dummy interrupt function
;-----
dummy_int:
    reit
    .end

;*****
; C COMPILER for M16C/60, M16C/20
; Copyright 1995-2003 Renesas Technology America, Inc.
; All Rights Reserved.
;*****

```

## 7.2 Customized SECT30 (sect30\_26.inc)

```

;*****
; C Compiler for M16C/60, M16C/20
; Copyright 1995-2003 Renesas Technology America, Inc,
; All Rights Reserved.
;
; Written by T.Aoyama
;
; sect30_26.inc : section definition
; This program is applicable when using the basic I/O library
;
;=====
; $Log:$
; $Id: sect30.inc,v 1.9 2000/06/20 09:07:11 simomura Exp $
;          sect30_26, v 1.1 2003/05/15 11:36 R. Seville
;=====
;*****

;-----
; HEAP SIZE definition. Only used for memory allocate functions
; (malloc, realloc, etc). If not, set to 0h.
;-----
HEAPSIZE      .equ    0h          ; Heap memory not needed.

;-----
; STACK SIZE definition. Unless the system is running an RTOS, both
; interrupts and function calls should use the istack only (default startup
; configuration). If not, set to 0h.
;-----
STACKSIZE     .equ    0h          ;

;-----
; INTERRUPT STACK SIZE definition
;-----
ISTACKSIZE    .equ    100h       ;

```

```

;-----
; INTERRUPT VECTOR ADDRESS.
; Do NOT set within a flash memory block used by the ROM Monitor which
; starts at 0FF900h. Size of vector table is 256 bytes.
;-----
VECTOR_ADR      .equ      0ff800h

;=====
; Initialize Macro declarations. These macro's are used in the startup
; file (ncrto_26.a30) for initializing C variables. Clears global
; variables, sets intialized variables, etc.
;-----
N_BZERO .macro TOP_ ,SECT_
    mov.b    #00H, R0L
    mov.w    #(TOP_ & 0FFFFH), A1
    mov.w    #sizeof SECT_ , R3
    sstr.b
    .endm

N_BCOPY .macro FROM_,TO_,SECT_
    mov.w    #(FROM_ & 0FFFFH),A0
    mov.b    #(FROM_ >>16),R1H
    mov.w    #TO_ ,A1
    mov.w    #sizeof SECT_ , R3
    smovf.b
    .endm

HEAPINIT .macro
    .glb    __mbase
    .glb    __mnext
    .glb    __msize
    mov.w    #(heap_top&0FFFFH), __mbase
    mov.w    #(heap_top>>16), __mbase+2
    mov.w    #(heap_top&0FFFFH), __mnext
    mov.w    #(heap_top>>16), __mnext+2
    mov.w    #(HEAPSIZE&0FFFFH), __msize
    mov.w    #(HEAPSIZE>>16), __msize+2
    .endm

;-----
; Special page vectors. This macro puts the jump address of
; functions defined as special page into the special page vector table.
; See example calls below. See the M16C Software Manual and the NC30
; manual for more information on special page vectors.
;-----
;      macro define for special page
;
;Format:
;      SPECIAL number
;

```

```

SPECIAL .macro NUM
        .org    0FFFFEH-(NUM*2)
        .glb    __SPECIAL_@NUM
        .word   __SPECIAL_@NUM & 0FFFFH
.endm

;-----
; Section allocation. The following declarations sets the location of the
; sections in the physical memory map. DO not change these settings
; without referring to the NC30 manual on startup files.
;-----
; Near RAM data area
;-----
; SBDATA area
        .section    data_SE,DATA
        .org    400H

data_SE_top:

        .section    bss_SE,DATA,ALIGN
bss_SE_top:

        .section    data_SO,DATA
data_SO_top:

        .section    bss_SO,DATA
bss_SO_top:

; near RAM area
        .section    data_NE,DATA,ALIGN
data_NE_top:

        .section    bss_NE,DATA,ALIGN
bss_NE_top:

        .section    data_NO,DATA
data_NO_top:

        .section    bss_NO,DATA
bss_NO_top:

;-----
; Stack area. If the USP is not required, and the RAM
; allocated to the USP is needed, do not modify the declarations
; below, simply set the USTACKSIZE (above) to zero.
;-----
        .section    stack,DATA
        .blkb    STACKSIZE
stack_top:

        .blkb    ISTACKSIZE
istack_top:

```



```

;-----
; Heap section. If the heap is not required, and the RAM
; allocated to the heap is needed, do not modify the declarations
; below, Simply set the HEAPSIZ (above) to zero.
;-----
        .section          heap,DATA
heap_top:
        .blkb    HEAPSIZ

;-----
; Near ROM data area. For "near const".
; By definition, Near ROM is all ROM below address 10000h
; Virtual EEPROM can be found in 0F000h to 0FFFFh. However,
; the user program must enable access (set pm10 to 1) to it
; before using it.
;-----
        .section          rom_NE,ROMDATA
        .org              0F000H          ; Virtual EEPROM - 2KB x 2
rom_NE_top:

        .section          rom_NO,ROMDATA
rom_NO_top:

;-----
; Far ROM data area
;-----
        .section          rom_FE,ROMDATA
        .org              0F0000H
rom_FE_top:

        .section          rom_FO,ROMDATA
rom_FO_top:

;-----
; Initial data of 'data' section
;-----
        .section          data_SEI,ROMDATA
data_SEI_top:

        .section          data_SOI,ROMDATA
data_SOI_top:

        .section          data_NEI,ROMDATA
data_NEI_top:

        .section          data_NOI,ROMDATA
data_NOI_top:

        .section          data_FEI,ROMDATA
data_FEI_top:

        .section          data_FOI,ROMDATA
data_FOI_top:

```

```

;-----
; Switch Table Section
;-----
        .section      switch_table,ROMDATA
switch_table_top:

;-----
; code area
;-----
        .section      program

        .section      interrupt

        .section      program_S      ; special page code must be in
                                       ; address range from F0000H to FFFDCh
                                       ; Start of M30262F8 ROM address will
                                       ; vary depending on M30262 used.

;-----
; variable vector section
; For proper interrupt operation, replace "dummy_int" with the assembler
; label or absolute address of the interrupt service routine
;-----
        .section      vector          ; variable vector table
        .org      VECTOR_ADR

        .lword      dummy_int          ; vector 0 (BRK)
        .org      (VECTOR_ADR +16)
        .lword      dummy_int          ; INT3
        .lword      dummy_int          ; Reserved
        .lword      dummy_int          ; Reserved
        .lword      dummy_int          ; Reserved
        .lword      dummy_int          ; INT5 - bits b7 & b6 of the IFSR (035Fh) must be set
when using INT4 &
        .lword      dummy_int          ; INT4 - INT5 (bits are set in ncrct0_26.a30 before
main() is called).
        .lword      dummy_int          ; UART2 Bus collision detection iic
        .lword      dummy_int          ; DMA0 (for user)
        .lword      dummy_int          ; DMA1 (for user)
        .lword      dummy_int          ; Key-on wakeup (for user)
        .lword      dummy_int          ; AD Converter (for user)
        .lword      dummy_int          ; UART2 transmit/NACK, with iic mode NACK is selected.
        .lword      dummy_int          ; UART2 receive/ACK, with iic mode ACK is selected.
        .lword      dummy_int          ; UART0 transmit (for user)
        .lword      dummy_int          ; UART0 receive: use for UART0 receive interrupt.
        .lword      dummy_int          ; UART1 transmit (for user)
        .lword      dummy_int          ; UART1 receive: Not needed for M16C/26, USB-ICD uses
DBC irq.
        .lword      dummy_int          ; TIMER A0 (for user)
        .lword      dummy_int          ; TIMER A1 (for user)
        .lword      dummy_int          ; TIMER A2 (for user)
        .lword      dummy_int          ; TIMER A3 (for user)
        .lword      dummy_int          ; TIMER A4 (for user) (vector 25)
        .lword      dummy_int          ; TIMER B0 (for user) (vector 26)
        .lword      dummy_int          ; TIMER B1 (for user) (vector 27)

```

```

.lword dummy_int          ; TIMER B2 (for user) (vector 28)
.lword dummy_int          ; INT0 (for user) (vector 29)
.lword dummy_int          ; INT1 (for user) (vector 30)
.lword dummy_int          ; Reserved
.lword dummy_int          ; vector 32 (for user or MR30)
.lword dummy_int          ; vector 33 (for user or MR30)
.lword dummy_int          ; vector 34 (for user or MR30)
.lword dummy_int          ; vector 35 (for user or MR30)
.lword dummy_int          ; vector 36 (for user or MR30)
.lword dummy_int          ; vector 37 (for user or MR30)
.lword dummy_int          ; vector 38 (for user or MR30)
.lword dummy_int          ; vector 39 (for user or MR30)
.lword dummy_int          ; vector 40 (for user or MR30)
.lword dummy_int          ; vector 41 (for user or MR30)
.lword dummy_int          ; vector 42 (for user or MR30)
.lword dummy_int          ; vector 43 (for user or MR30)
.lword dummy_int          ; vector 44 (for user or MR30)
.lword dummy_int          ; vector 45 (for user or MR30)
.lword dummy_int          ; vector 46 (for user or MR30)
.lword dummy_int          ; vector 47 (for user or MR30)
; to vector 63 from vector 32 is used by MR30

```

```

;=====
; fixed vector section
;-----
        .section      fvector          ; fixed vector table
;=====
; special page definition
;-----
; Special page functions can be specified using
; "#pragma SPECIAL" directive and the macro defined above.
; Uncomment the proper line below to call the macro.
; See NC30 manual for more information.
;-----
;     SPECIAL 255
;     SPECIAL 254
;     SPECIAL 253
;     :
;     :
;     (omitted)
;     :
;     :
;     SPECIAL 24
;     SPECIAL 23
;     SPECIAL 22
;     SPECIAL 21
;     SPECIAL 20
;     SPECIAL 19
;     SPECIAL 18
;
;=====
; fixed vector section.
;-----
        .org      0fffdch

```

```
UDI:      .lword  dummy_int
OVER_FLOW:      .lword  dummy_int
BRKI:      .lword  dummy_int
ADDRESS_MATCH:      .lword  dummy_int
SINGLE_STEP:      .lword  dummy_int
WDT:      .lword  dummy_int
DBC:      .lword  dummy_int
NMI:      .lword  dummy_int
RESET:      .lword  start

;
;*****
;      C Compiler for M16C/60, M16C/20
;      Copyright 1995-2003 Renesas Technology America, Inc.
;      All Rights Reserved.
;*****
```

## Keep safety first in your circuit designs!

- Renesas Technology Corporation puts the maximum effort into making semiconductor products better and more reliable, but there is always the possibility that trouble may occur with them. Trouble with semiconductors may lead to personal injury, fire or property damage. Remember to give due consideration to safety when making your circuit designs, with appropriate measures such as (i) placement of substitutive, auxiliary circuits, (ii) use of nonflammable material or (iii) prevention against any malfunction or mishap.

## Notes regarding these materials

- These materials are intended as a reference to assist our customers in the selection of the Renesas Technology Corporation product best suited to the customer's application; they do not convey any license under any intellectual property rights, or any other rights, belonging to Renesas Technology Corporation or a third party.
- Renesas Technology Corporation assumes no responsibility for any damage, or infringement of any third-party's rights, originating in the use of any product data, diagrams, charts, programs, algorithms, or circuit application examples contained in these materials.
- All information contained in these materials, including product data, diagrams, charts, programs and algorithms represents information on products at the time of publication of these materials, and are subject to change by Renesas Technology Corporation without notice due to product improvements or other reasons. It is therefore recommended that customers contact Renesas Technology Corporation or an authorized Renesas Technology Corporation product distributor for the latest product information before purchasing a product listed herein.  
The information described here may contain technical inaccuracies or typographical errors.  
Renesas Technology Corporation assumes no responsibility for any damage, liability, or other loss arising from these inaccuracies or errors.  
Please also pay attention to information published by Renesas Technology Corporation by various means, including the Renesas Technology Corporation Semiconductor home page (<http://www.renesas.com>).
- When using any or all of the information contained in these materials, including product data, diagrams, charts, programs, and algorithms, please be sure to evaluate all information as a total system before making a final decision on the applicability of the information and products. Renesas Technology Corporation assumes no responsibility for any damage, liability or other loss resulting from the information contained herein.
- Renesas Technology Corporation semiconductors are not designed or manufactured for use in a device or system that is used under circumstances in which human life is potentially at stake. Please contact Renesas Technology Corporation or an authorized Renesas Technology Corporation product distributor when considering the use of a product contained herein for any specific purposes, such as apparatus or systems for transportation, vehicular, medical, aerospace, nuclear, or undersea repeater use.
- The prior written approval of Renesas Technology Corporation is necessary to reprint or reproduce in whole or in part these materials.
- If these products or technologies are subject to the Japanese export control restrictions, they must be exported under a license from the Japanese government and cannot be imported into a country other than the approved destination.  
Any diversion or reexport contrary to the export control laws and regulations of Japan and/or the country of destination is prohibited.
- Please contact Renesas Technology Corporation for further details on these materials or the products contained therein.