



GENTOS

Application Note #011

(AN011-V1.3)

AN

[GENTOS] Guideline for Assembly Programming

V1.3

December 2007

- ◆ CORERIVER Semiconductor reserves the right to make corrections, modifications, enhancements, improvements, and other changes to its products and services at any time.
- ◆ CORERIVER shall give customers at least a three month advance notice of intended discontinuation of a product or a service through its homepage.
- ◆ Customers should obtain the latest relevant information before placing orders and should verify that such information is current and complete.
- ◆ The CORERIVER products listed in this document are intended for usage in general electronics applications. These CORERIVER products are neither intended nor warranted for usage in equipment that requires extraordinarily high quality and/or reliability or a malfunction or failure of which may cause loss of human life or bodily injury..

Contents

1. Flow Chart : Assembly Programming
2. Assembly Programming
 - 1) The Inclusion of Header File
 - 2) The Declaration of Variable
 - 3) The Declaration of Interrupt Vector
 - 4) The Initialization of Program
 - 5) Main Function
 - 6) Sub Function
 - 7) Interrupt Function

1. Flow Chart : Assembly Programming

Initialization

◆ The Inclusion of Header File.

◆ The Declaration of Variables.
◆ The Declaration of MACRO.
◆ The Declaration of Interrupt Vector

◆ The Declaration of Interrupt Vectors.

Program

◆ SFR Setting
◆ The Initialization of Variables

◆ Main Program

Functions

◆ Sub Functions

◆ Interrupt Functions

[Header Files]

◆ MiDAS1.0 Family

- ✓ 44-PLCC / LQFP, 40-PDIP : GC80C520_ASM_PL44I.H
- ✓ 28-SPDIP / SOIC : GC80C520_ASM_SO28I.H

◆ MiDAS1.1 Family

- ✓ 20/16/14/8-SPDIP / SOIC : GC80C510_ASM_SO20I.H

◆ MiDAS2.0 Family

- ✓ 44-PLCC/TQFP (A Version) : GC80C590A0_ASM_TQ44C.H
- ✓ 44-PLCC/TQFP (G Version), and 64/80/100-TQFP : GC80C590A0_ASM_TQ100C.H

◆ MiDAS2.1 Family

- ✓ 32/28/LQFP/SPDIP/SOIC/MLF : GC89C520_ASM_TQ32I.H

◆ MiDAS3.0 Family

- ✓ 44/32/MQFP/LQFP/POFP /PLCC/SOIC/MLF : GC89L591A0_MQ44_ASM.H

◆ ATOM1.0 Family

- ✓ 24/20/8/SOIC/DIP : GC49C501G0_ASM.H

[Notes]

◆ Comment Symbol

- ✓ Semicolon ";"

◆ Except of comment and label, All directives and instructions need one or more space in the front of code line.

◆ This AN011 deals with MiDAS1.1 Family.

◆ The Inclusion of Header File

```
.include "GC80C510_ASM_SO20I.H"
```

◆ The Declaration of Variable

✓ Global Variable

```
.globl time0 ← Variable : time0  
time0:  
.ds 1 ← Data Size : 1 byte
```

✓ Byte Type Assignment at Internal RAM

```
; Classic(GenASM), Keil Style(A51)  
loop_cnt EQU 0x70 ← Internal RAM  
add_value EQU 0x21 Address
```

```
; New Style(ASX8051)  
loop_cnt = 0x70  
add_value = 0x21
```

```
; 4-bit(ASAC51)  
loop_cnt = 0x70  
.equ add_value, 0x21  
sub_value .equ 0x21
```

- ※ Classic Style is Old style that used GenASM assembler.
- ※ Keil Style is style that used A51 assembler.
- ※ New Style is style that used ASX8051 assembler.
- ※ 4-bit is style that used ASAC51 assembler for ATOM.

✓ Bit Type Assignment at Internal RAM

```
; Classic(GenASM), Keil Style(A51)  
test_bit0 EQU 0x00 ← Internal RAM  
test_bit1 EQU 0x01 Bit Address
```

```
; New Style(ASX8051)  
test_bit0 = 0x00  
test_bit1 = 0x01
```

```
; 4-bit(ASAC51)  
test_bit0 = 0x00  
.equ test_bit1, 0x01  
test_bit2 .equ 0x02
```

◆ The Declaration of Interrupt Vector

```
-----  
; Interrupt Vector (Refer to Brief Manual)  
  
; Program Start  
.org 0x0000  
ljmp reset ;Program (Init.)  
  
; INT0B Interrupt  
.org 0x0003  
ljmp INT0B_int  
  
; Timer0 Interrupt  
.org 0x000B  
ljmp timer0_int ;  
  
; ADC Interrupt  
.org 0x003B  
ljmp ADC_int  
  
; INT2 Interrupt  
.org 0x0043  
ljmp INT2_int  
  
; Watchdog Timer Interrupt  
.org 0x0063  
ljmp WDT_int  
  
.org 0x0100 ; Program Start
```

◆ The Initialization Program

✓ SFR Setting & The Initialization of Variables

```
-----  
; PROGRAM START  
reset:  
  
-----  
; System Clock Setting (MiDAS1.1)  
; - In order to use External Crystal OSC.  
crystal_osc_amp_ready:  
mov a, STATUS ;STATUS.4(XTUP)  
jnb acc4, crystal_osc_amp_ready  
  
orl EXIF, #XTRG_ ;EXIF.3(XTRG)  
  
-----  
; Initialization  
;1. for Ports  
mov P0, #0000_0000b  
mov P2, #0000_0000b  
  
;2. for Port Input/Output Control  
mov P0DIR, #0000_0101B  
mov P2DIR, #1111_1100B  
  
;3. for Port Pull-up R Setting  
mov P0SEL, #0000_0101B  
mov P2SEL, #0000_0100B  
  
.  
.  
.  
  
clr test_bit0 ;Clear bit  
setb test_bit1 ;Set bit
```

◆ Main Program

- ✓ User can use "sjmp Main" when "Main" is not long.

```
-----  
; Main Function  
Main:  
  
; Call Sub Function  
lcall test_comm_start  
  
; Call Macro Function  
mov r0, #0x40 ;INT0B Enable  
mov r1, #0xF0 ;INT2 Enable  
  
ljmp Main
```

◆ Sub Function

```
-----  
; Sub Function  
test_comm_start:  
jnb test1_sck, setup0  
setb test2_sck  
  
setup0:  
jnb test_bit0, setup1  
setb test_bit1  
  
                  •  
                  •  
                  •  
  
ret ;Function Return
```

◆ Interrupt Function

```
-----  
; External INT0B Interrupt  
INT0B_int:  
; Insert code for executed user code  
;  
clr IE0 ;INT0 Flag Clear  
reti ;Interrupt Return  
  
-----  
; Timer0 Interrupt Function  
timer0_int:  
; Insert code for executed user code  
;  
clr TF0 ;Timer 0 Flag Clear  
reti ;Interrupt Return
```