TI DSP, MCU 및 Xilinx Zynq FPGA 프로그래밍 전문가 과정

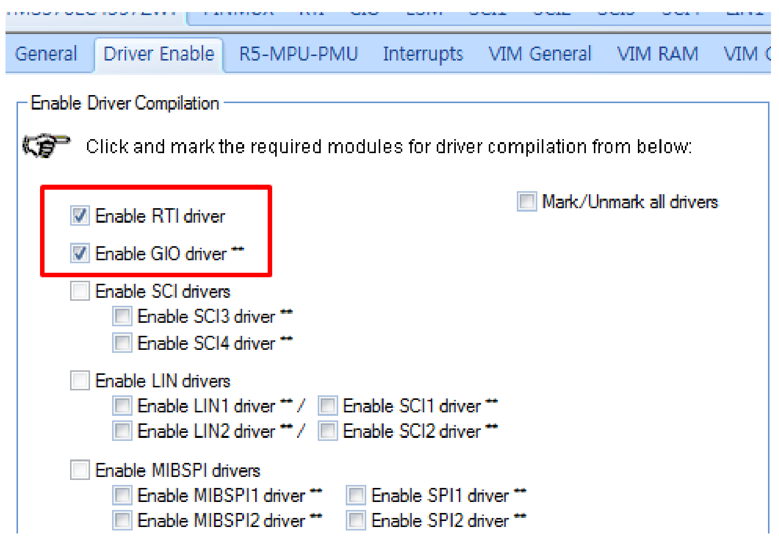
강사 – Innova Lee(이상훈)
gcccompil3r@gmail.com

학생 – GJ (박현우)
uc820@naver.com

목차

4. Cortex-R5F Hercules Safety MCU – (OC_CIRCUIT 설정)

<1>



General | Driver Enable | R5-MPU-PMU | Interrupts | VIM General | VIM RAM | VIM

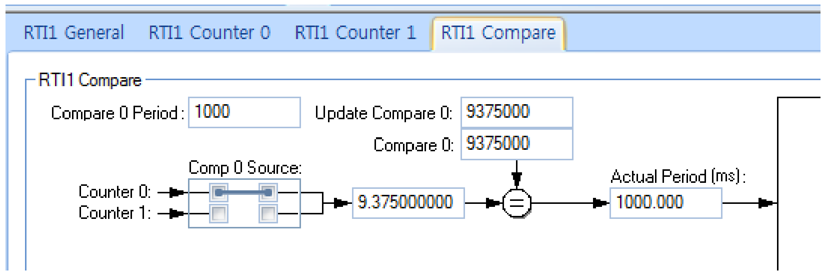
Enable Driver Compilation

Click and mark the required modules for driver compilation from below:

- [✓] Enable RTI driver
- [✓] Enable GIO driver **
- [ ] Enable SCI drivers
  - [ ] Enable SCI3 driver **
  - [ ] Enable SCI4 driver **
- [ ] Enable LIN drivers
  - [ ] Enable LIN1 driver ** / [ ] Enable SCI1 driver **
  - [ ] Enable LIN2 driver ** / [ ] Enable SCI2 driver **
- [ ] Enable MIBSPI drivers
  - [ ] Enable MIBSPI1 driver ** [ ] Enable SPI1 driver **
  - [ ] Enable MIBSPI2 driver ** [ ] Enable SPI2 driver **

[ ] Mark/Unmark all drivers

<2>



RTI1 General | RTI1 Counter 0 | RTI1 Counter 1 | RTI1 Compare

RTI1 Compare

Compare 0 Period : 1000

Update Compare 0: 9375000

Compare 0: 9375000

Comp 0 Source:

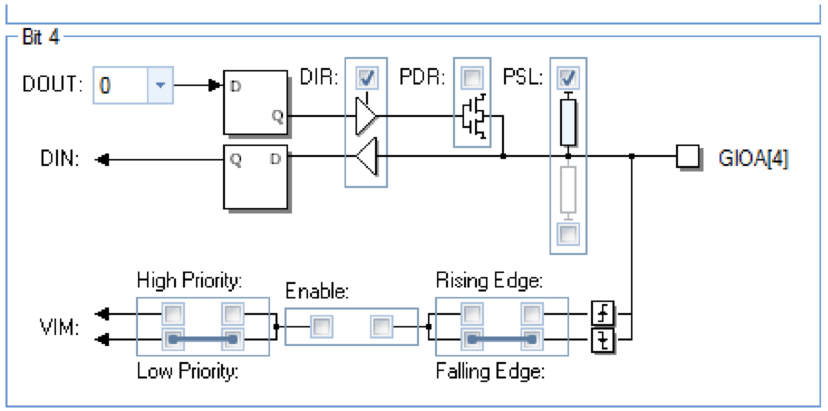
Counter 0:

Counter 1:

9.375000000

Actual Period (ms) : 1000.000

<3>



Bit 4

DOUT: 0

DIR: [✓]

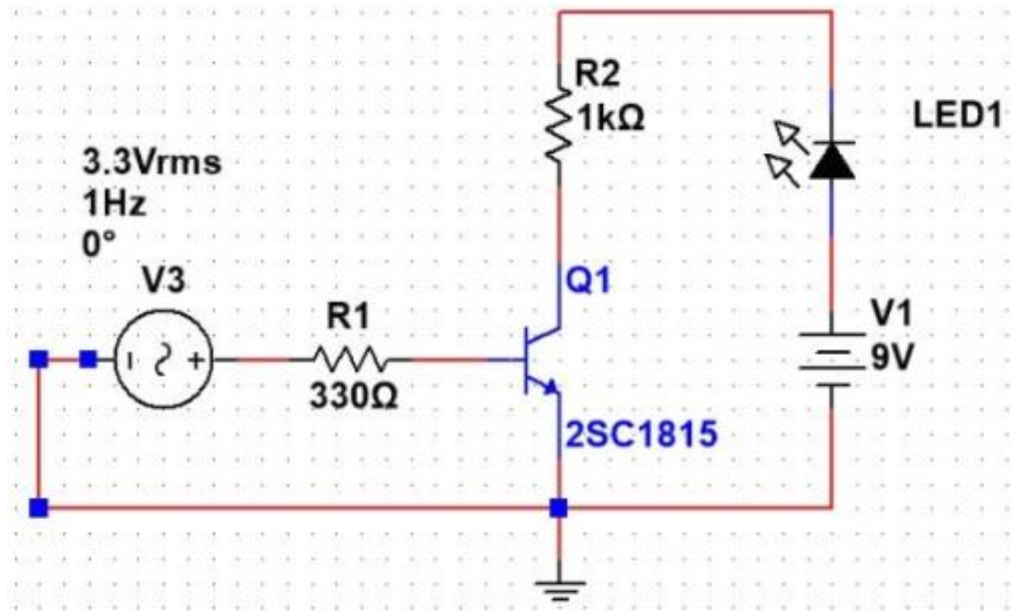
PDR: [ ]

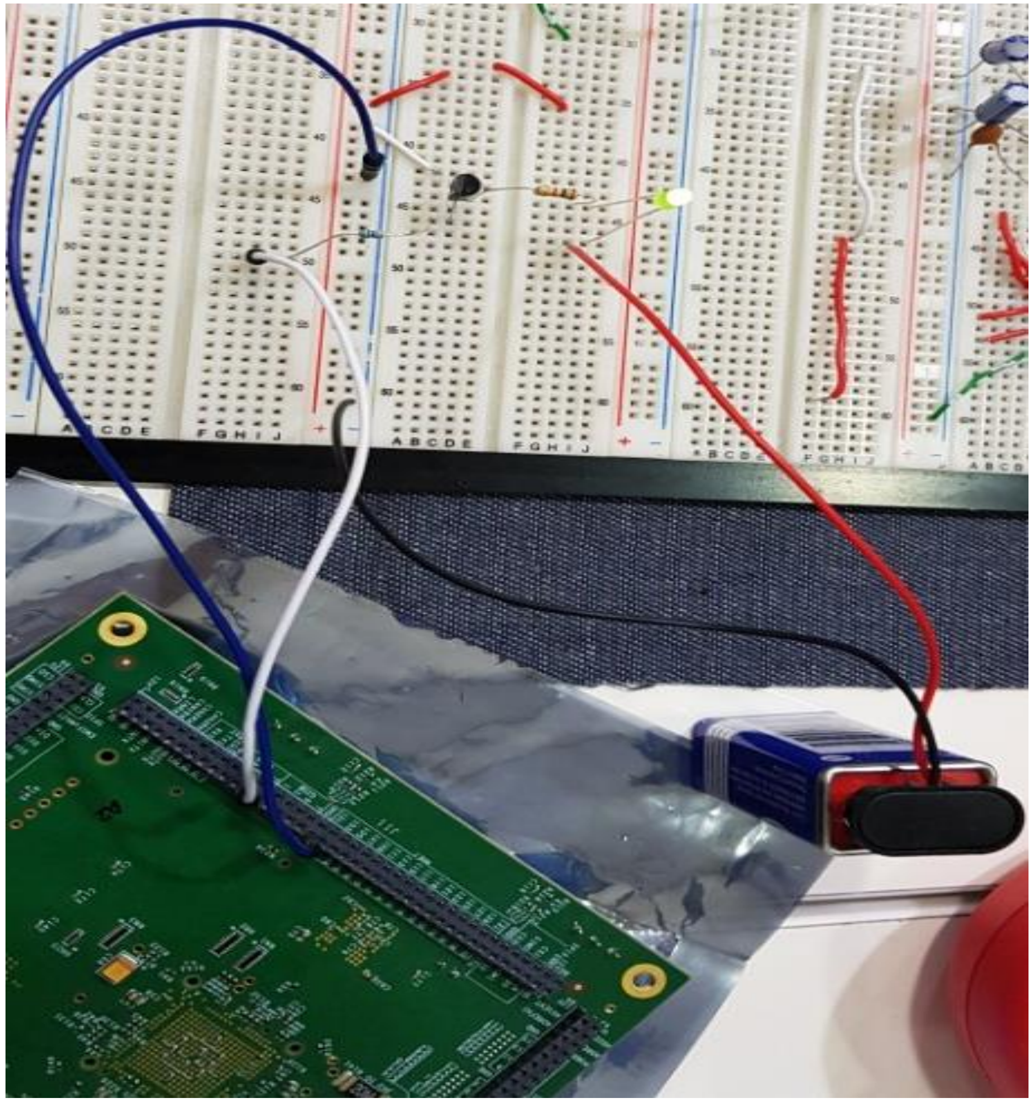
PSL: [✓]

DIN:

GIOA[4]

4. Cortex-R5F Hercules Safety MCU – (OC_CIRCUIT 회로 구성)



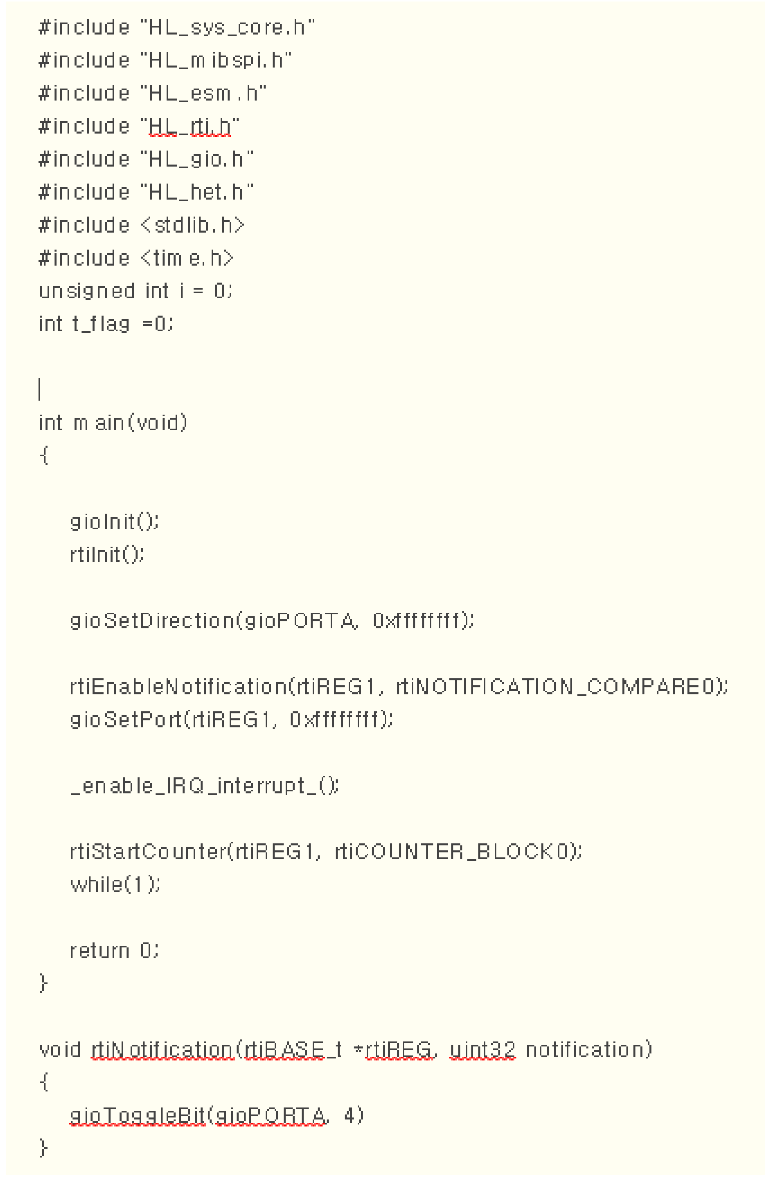


오픈 콜렉터는 왜 사용하나?

1. 구성된 회로에서의 과전압으로 MCU에 고장을 방지

2. 특정 센서에서 들어오는 신호로 구성한 회로에 원할 때만 동작을 유도

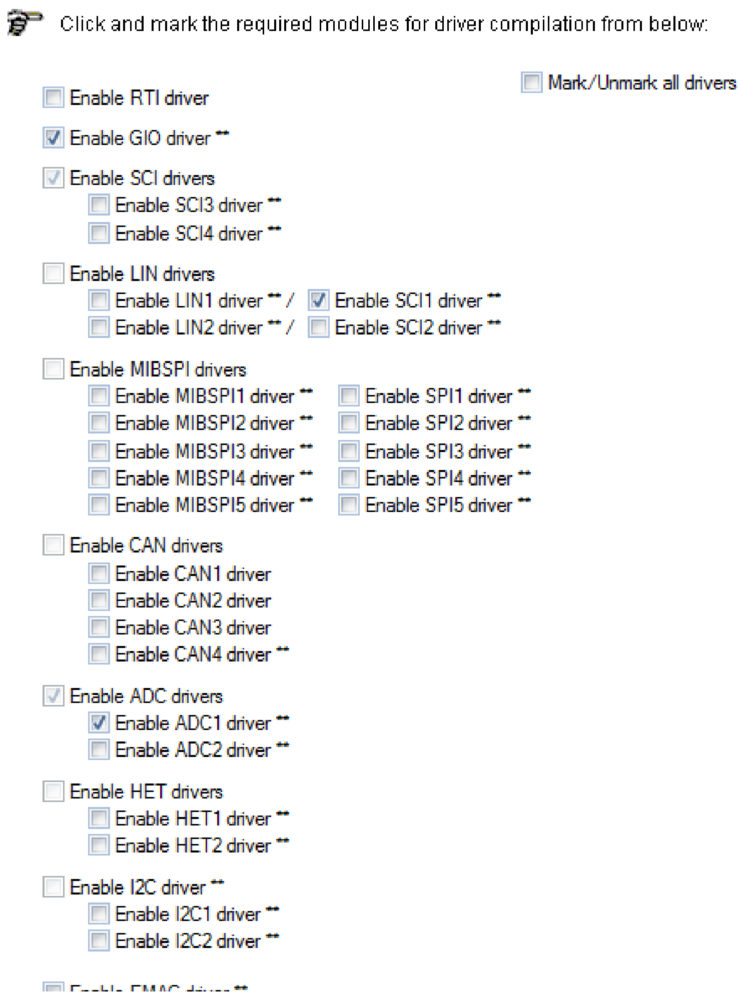
4. Cortex-R5F Hercules Safety MCU - (OC_CIRCUIT 예제 코드)



```
#include "HL_sys_core.h"
#include "HL_mibspi.h"
#include "HL_esm.h"
#include "HL_rti.h"
#include "HL_gio.h"
#include "HL_het.h"
#include <stdlib.h>
#include <time.h>
unsigned int i = 0;
int t_flag =0;


int main(void)
{

    gioInit();
    rtiInit();

    gioSetDirection(gioPORTA, 0xffffffff);

    rtiEnableNotification(rtiREG1, rtiNOTIFICATION_COMPARE0);
    gioSetPort(rtiREG1, 0xffffffff);

    _enable_IRQ_interrupt_();

    rtiStartCounter(rtiREG1, rtiCOUNTER_BLOCK0);
    while(1);

    return 0;
}

void rtiNotification(rtiBASE_t *rtiREG, uint32 notification)
{
    gioToggleBit(gioPORTA, 4)
}
```

4. Cortex-R5F Hercules Safety MCU – (ADC 설정 & 회로 구성)

<1>

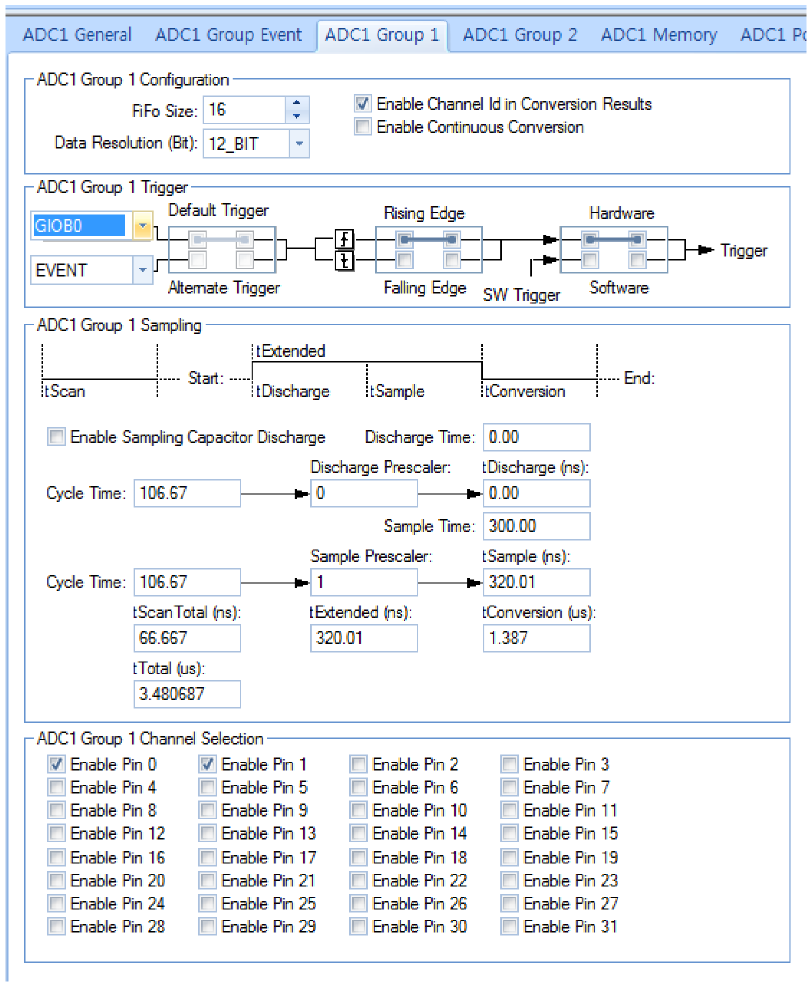


Click and mark the required modules for driver compilation from below:

Mark/Unmark all drivers

- Enable RTI driver
- [✓ Enable GIO driver **]
- [✓ Enable SCI drivers]
  - Enable SCI3 driver **
  - Enable SCI4 driver **
- Enable LIN drivers
  - Enable LIN1 driver ** / [✓ Enable SCI1 driver **]
  - Enable LIN2 driver ** / Enable SCI2 driver **
- Enable MIBSPI drivers
  - Enable MIBSPI1 driver ** / Enable SPI1 driver **
  - Enable MIBSPI2 driver ** / Enable SPI2 driver **
  - Enable MIBSPI3 driver ** / Enable SPI3 driver **
  - Enable MIBSPI4 driver ** / Enable SPI4 driver **
  - Enable MIBSPI5 driver ** / Enable SPI5 driver **
- Enable CAN drivers
  - Enable CAN1 driver
  - Enable CAN2 driver
  - Enable CAN3 driver
  - Enable CAN4 driver **
- [✓ Enable ADC drivers]
  - [✓ Enable ADC1 driver **]
  - Enable ADC2 driver **
- Enable HET drivers
  - Enable HET1 driver **
  - Enable HET2 driver **
- Enable I2C driver **
  - Enable I2C1 driver **
  - Enable I2C2 driver **

<2>



ADC1 General | ADC1 Group Event | ADC1 Group 1 | ADC1 Group 2 | ADC1 Memory | ADC1 P

ADC1 Group 1 Configuration

FiFo Size: 16
Data Resolution (Bit): 12_BIT
[✓ Enable Channel Id in Conversion Results]
Enable Continuous Conversion

ADC1 Group 1 Trigger

GIOB0
EVENT

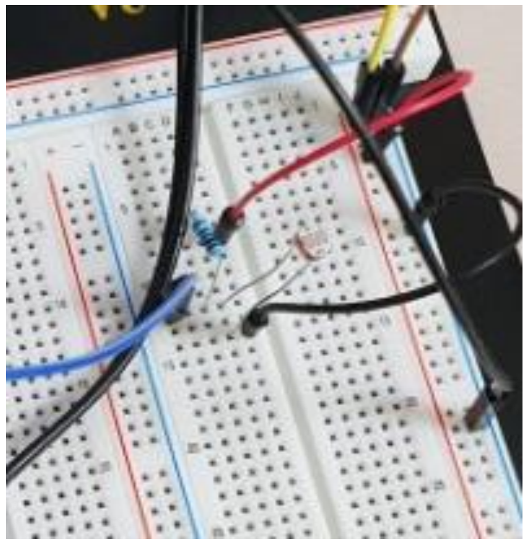
ADC1 Group 1 Sampling

Enable Sampling Capacitor Discharge
Discharge Time: 0.00
Cycle Time: 106.67 → Discharge Prescaler: 0 → tDischarge (ns): 0.00
Sample Time: 300.00
Cycle Time: 106.67 → Sample Prescaler: 1 → tSample (ns): 320.01
tScanTotal (ns): 66.667
tExtended (ns): 320.01
tConversion (us): 1.387
tTotal (us): 3.480687

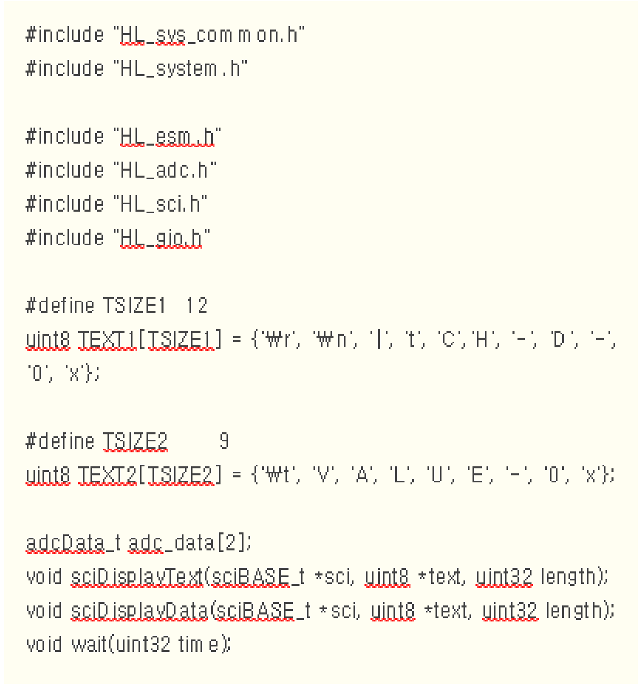
ADC1 Group 1 Channel Selection

[✓ Enable Pin 0]	[✓ Enable Pin 1]	Enable Pin 2	Enable Pin 3
Enable Pin 4	Enable Pin 5	Enable Pin 6	Enable Pin 7
Enable Pin 8	Enable Pin 9	Enable Pin 10	Enable Pin 11
Enable Pin 12	Enable Pin 13	Enable Pin 14	Enable Pin 15
Enable Pin 16	Enable Pin 17	Enable Pin 18	Enable Pin 19
Enable Pin 20	Enable Pin 21	Enable Pin 22	Enable Pin 23
Enable Pin 24	Enable Pin 25	Enable Pin 26	Enable Pin 27
Enable Pin 28	Enable Pin 29	Enable Pin 30	Enable Pin 31

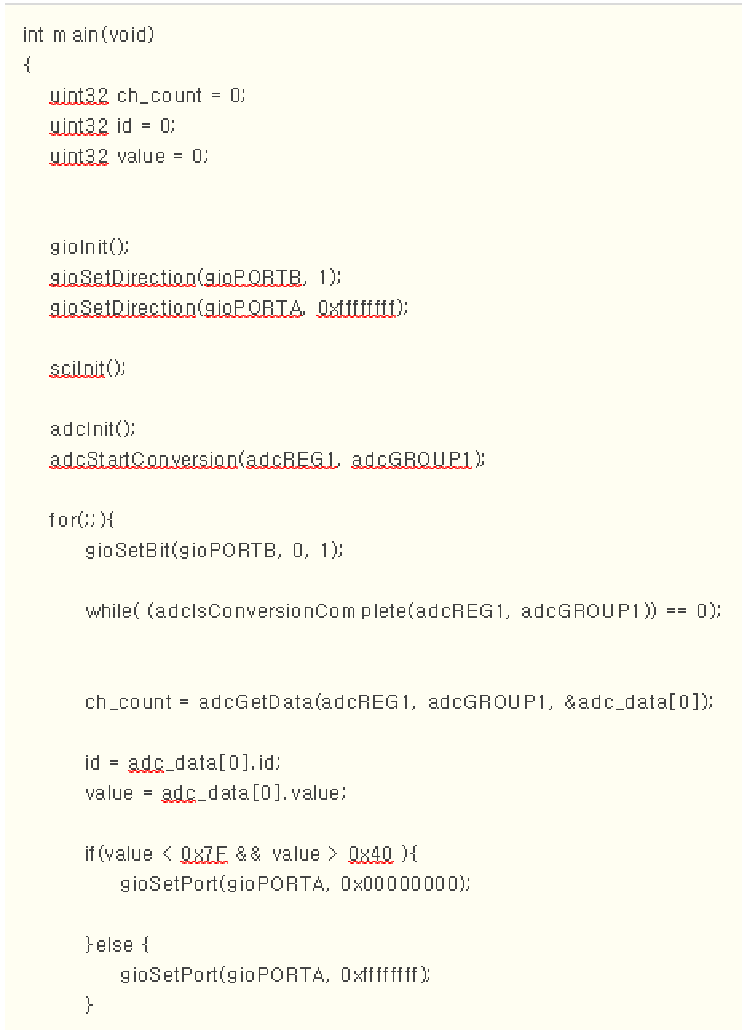
<3>



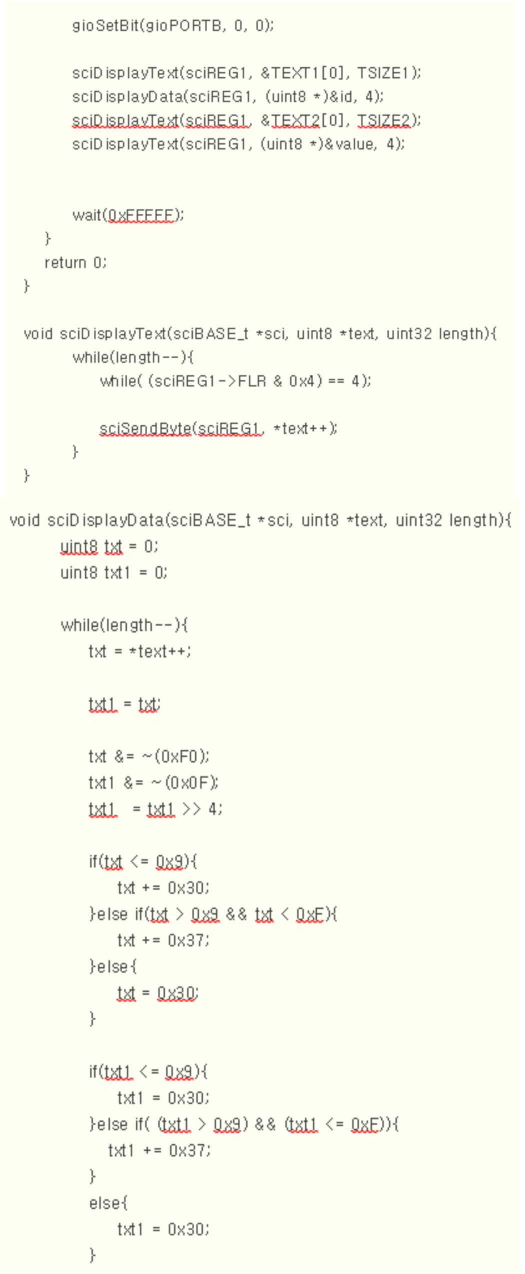
4. Cortex-R5F Hercules Safety MCU – (ADC 예제 코드)



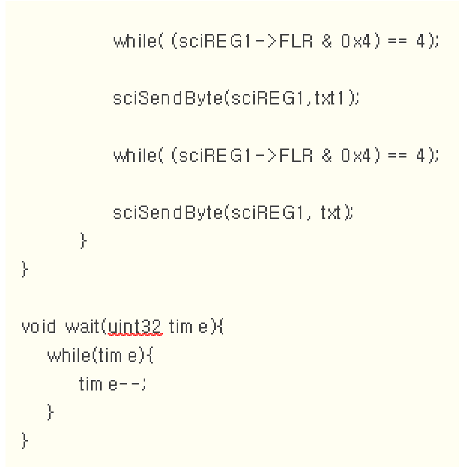
```c
#include "HL_sys_common.h"
#include "HL_system.h"

#include "HL_esm.h"
#include "HL_adc.h"
#include "HL_sci.h"
#include "HL_gio.h"

#define TSIZE1  12
uint8 TEXT1[TSIZE1] = {'₩r', '₩n', '|', 't', 'C','H', '-', 'D', '-',
'0', 'x'};

#define TSIZE2      9
uint8 TEXT2[TSIZE2] = {'₩t', 'V', 'A', 'L', 'U', 'E', '-', '0', 'x'};

adcData_t adc_data[2];
void sciDisplayText(sciBASE_t *sci, uint8 *text, uint32 length);
void sciDisplayData(sciBASE_t *sci, uint8 *text, uint32 length);
void wait(uint32 time);
```



```c
int main(void)
{
   uint32 ch_count = 0;
   uint32 id = 0;
   uint32 value = 0;


   gioInit();
   gioSetDirection(gioPORTB, 1);
   gioSetDirection(gioPORTA, 0xffffffff);

   sciInit();

   adcInit();
   adcStartConversion(adcREG1, adcGROUP1);

   for(;;){
       gioSetBit(gioPORTB, 0, 1);

       while( (adcIsConversionComplete(adcREG1, adcGROUP1)) == 0);


       ch_count = adcGetData(adcREG1, adcGROUP1, &adc_data[0]);

       id = adc_data[0].id;
       value = adc_data[0].value;

       if(value < 0x7F && value > 0x40 ){
           gioSetPort(gioPORTA, 0x00000000);

       }else {
           gioSetPort(gioPORTA, 0xffffffff);
       }
```



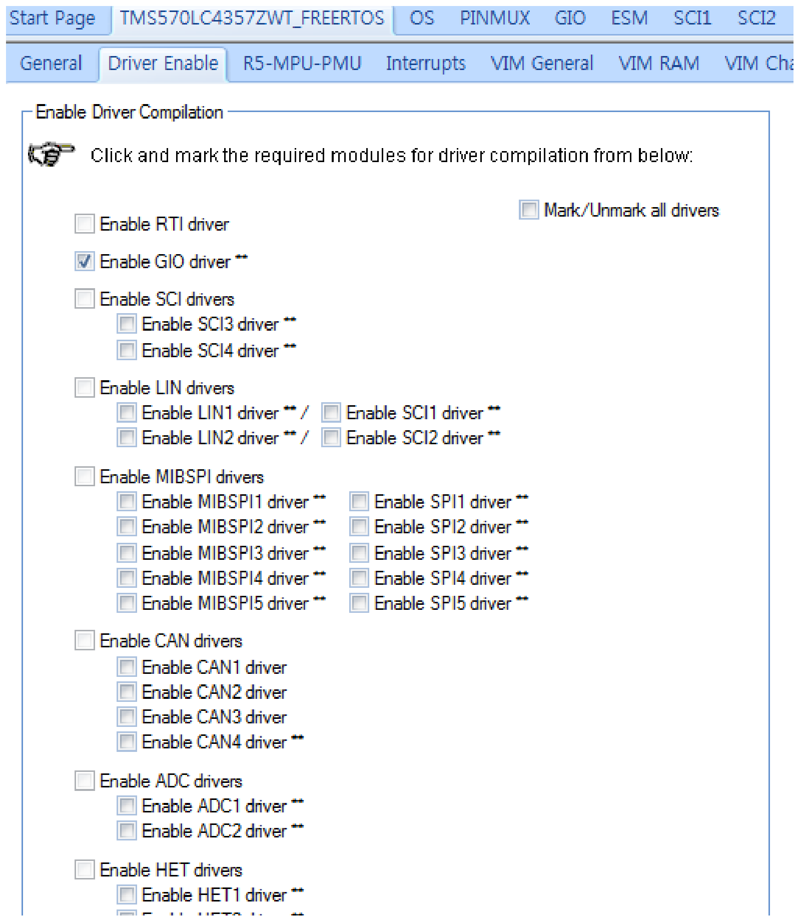
```c
       gioSetBit(gioPORTB, 0, 0);

       sciDisplayText(sciREG1, &TEXT1[0], TSIZE1);
       sciDisplayData(sciREG1, (uint8 *)&id, 4);
       sciDisplayText(sciREG1, &TEXT2[0], TSIZE2);
       sciDisplayText(sciREG1, (uint8 *)&value, 4);


       wait(0xFFFFF);
    }
    return 0;
  }

  void sciDisplayText(sciBASE_t *sci, uint8 *text, uint32 length){
       while(length--){
           while( (sciREG1->FLR & 0x4) == 4);

           sciSendByte(sciREG1, *text++);
       }
  }
void sciDisplayData(sciBASE_t *sci, uint8 *text, uint32 length){
      uint8 txt = 0;
      uint8 txt1 = 0;

      while(length--){
         txt = *text++;

         txt1 = txt;

         txt &= ~(0xF0);
         txt1 &= ~(0x0F);
         txt1  = txt1 >> 4;

         if(txt <= 0x9){
            txt += 0x30;
         }else if(txt > 0x9 && txt < 0xF){
            txt += 0x37;
         }else{
            txt = 0x30;
         }

         if(txt1 <= 0x9){
            txt1 = 0x30;
         }else if( (txt1 > 0x9) && (txt1 <= 0xF)){
           txt1 += 0x37;
         }
         else{
            txt1 = 0x30;
         }
```



```c
            while( (sciREG1->FLR & 0x4) == 4);

            sciSendByte(sciREG1,txt1);

            while( (sciREG1->FLR & 0x4) == 4);

            sciSendByte(sciREG1, txt);
        }
}

void wait(uint32 time){
    while(time){
        time--;
    }
}
```

4. Cortex-R5F Hercules Safety MCU – (freeRTOS 설정)

<1>



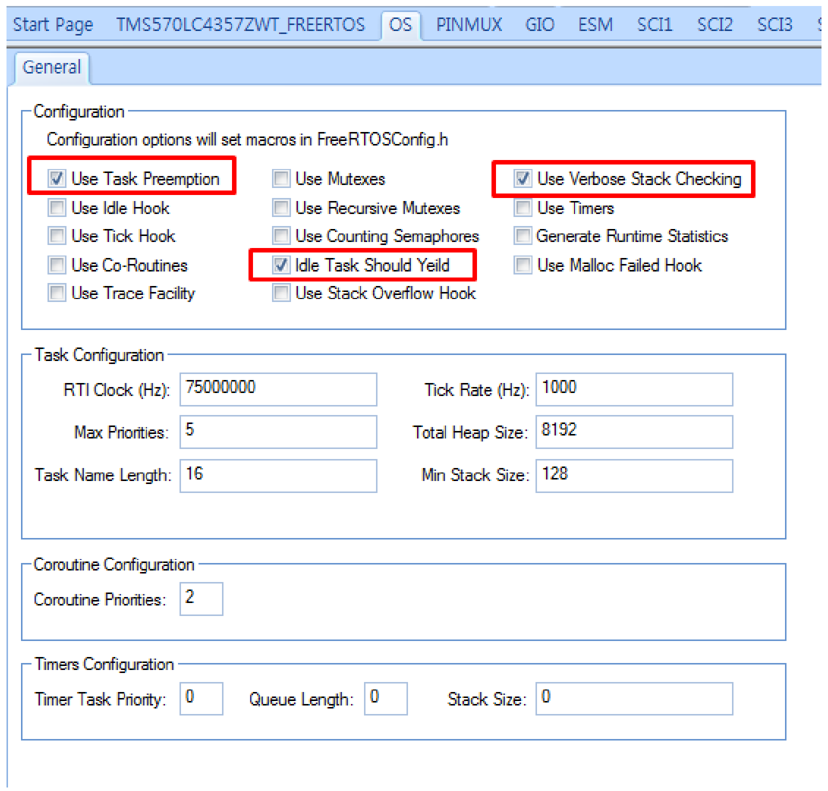
Enable Driver Compilation

Click and mark the required modules for driver compilation from below:

☐ Mark/Unmark all drivers

- ☐ Enable RTI driver
- ☑ Enable GIO driver **
- ☐ Enable SCI drivers
  - ☐ Enable SCI3 driver **
  - ☐ Enable SCI4 driver **
- ☐ Enable LIN drivers
  - ☐ Enable LIN1 driver ** / ☐ Enable SCI1 driver **
  - ☐ Enable LIN2 driver ** / ☐ Enable SCI2 driver **
- ☐ Enable MIBSPI drivers
  - ☐ Enable MIBSPI1 driver ** ☐ Enable SPI1 driver **
  - ☐ Enable MIBSPI2 driver ** ☐ Enable SPI2 driver **
  - ☐ Enable MIBSPI3 driver ** ☐ Enable SPI3 driver **
  - ☐ Enable MIBSPI4 driver ** ☐ Enable SPI4 driver **
  - ☐ Enable MIBSPI5 driver ** ☐ Enable SPI5 driver **
- ☐ Enable CAN drivers
  - ☐ Enable CAN1 driver
  - ☐ Enable CAN2 driver
  - ☐ Enable CAN3 driver
  - ☐ Enable CAN4 driver **
- ☐ Enable ADC drivers
  - ☐ Enable ADC1 driver **
  - ☐ Enable ADC2 driver **
- ☐ Enable HET drivers
  - ☐ Enable HET1 driver **

<2>



Configuration

Configuration options will set macros in FreeRTOSConfig.h

☑ Use Task Preemption	☐ Use Mutexes	☑ Use Verbose Stack Checking
☐ Use Idle Hook	☐ Use Recursive Mutexes	☐ Use Timers
☐ Use Tick Hook	☐ Use Counting Semaphores	☐ Generate Runtime Statistics
☐ Use Co-Routines	☑ Idle Task Should Yeild	☐ Use Malloc Failed Hook
☐ Use Trace Facility	☐ Use Stack Overflow Hook	

Task Configuration

RTI Clock (Hz):	75000000	Tick Rate (Hz):	1000
Max Priorities:	5	Total Heap Size:	8192
Task Name Length:	16	Min Stack Size:	128

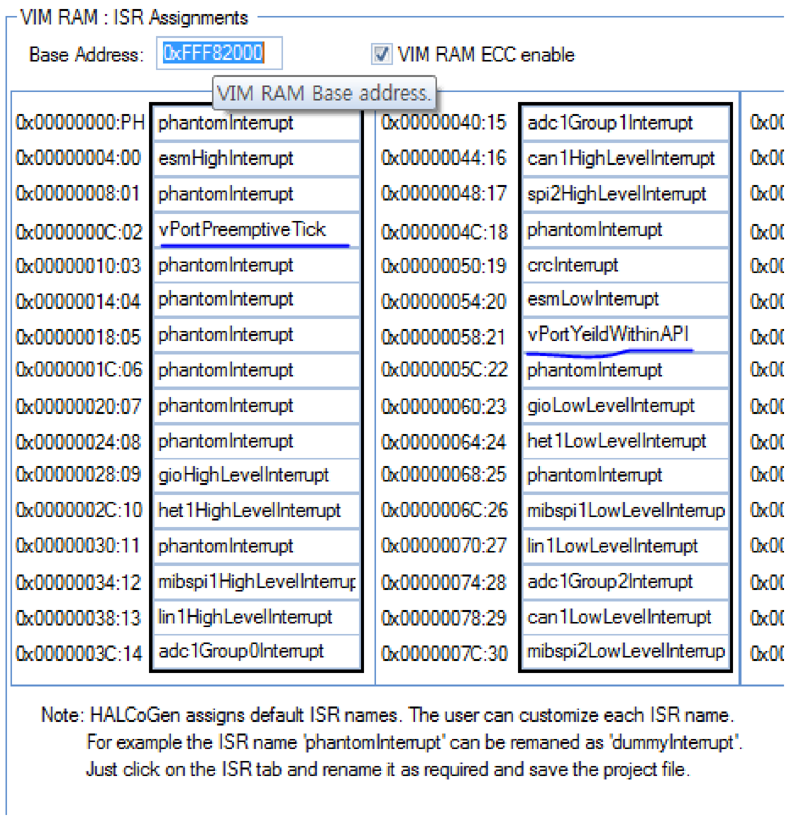
Coroutine Configuration

Coroutine Priorities: 2

Timers Configuration

Timer Task Priority: 0 Queue Length: 0 Stack Size: 0

<3>



VIM RAM : ISR Assignments

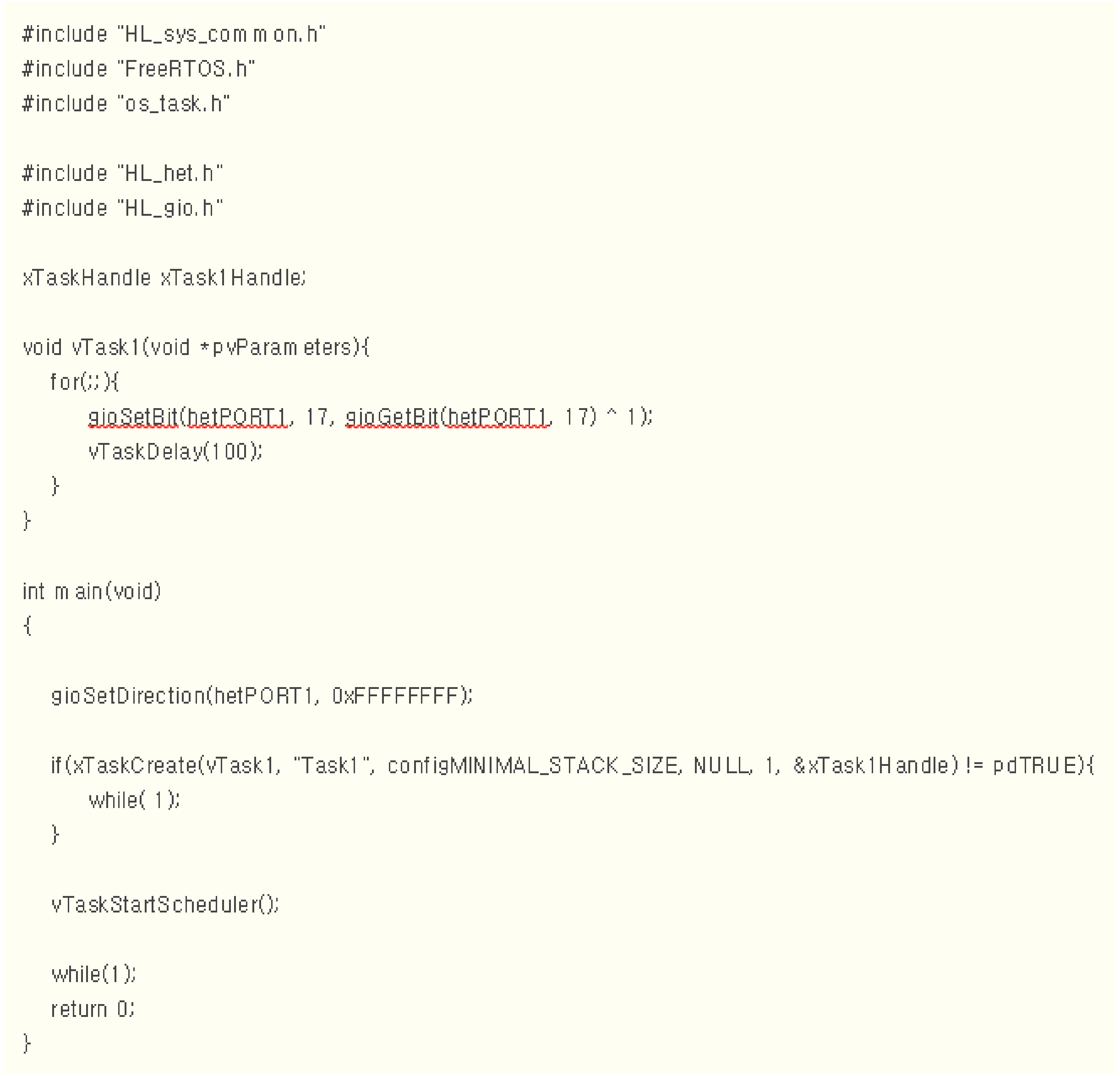
Base Address: 0xFFF82000 ☑ VIM RAM ECC enable

VIM RAM Base address.

0x00000000:PH	phantomInterrupt	0x00000040:15	adc1Group1Interrupt
0x00000004:00	esmHighInterrupt	0x00000044:16	can1HighLevelInterrupt
0x00000008:01	phantomInterrupt	0x00000048:17	spi2HighLevelInterrupt
0x0000000C:02	vPortPreemptiveTick	0x0000004C:18	phantomInterrupt
0x00000010:03	phantomInterrupt	0x00000050:19	crcInterrupt
0x00000014:04	phantomInterrupt	0x00000054:20	esmLowInterrupt
0x00000018:05	phantomInterrupt	0x00000058:21	vPortYeildWithinAPI
0x0000001C:06	phantomInterrupt	0x0000005C:22	phantomInterrupt
0x00000020:07	phantomInterrupt	0x00000060:23	gioLowLevelInterrupt
0x00000024:08	phantomInterrupt	0x00000064:24	het1LowLevelInterrupt
0x00000028:09	gioHighLevelInterrupt	0x00000068:25	phantomInterrupt
0x0000002C:10	het1HighLevelInterrupt	0x0000006C:26	mibspi1LowLevelInterrup
0x00000030:11	phantomInterrupt	0x00000070:27	lin1LowLevelInterrupt
0x00000034:12	mibspi1HighLevelInterrup	0x00000074:28	adc1Group2Interrupt
0x00000038:13	lin1HighLevelInterrupt	0x00000078:29	can1LowLevelInterrupt
0x0000003C:14	adc1Group0Interrupt	0x0000007C:30	mibspi2LowLevelInterrup

Note: HALCoGen assigns default ISR names. The user can customize each ISR name.
For example the ISR name 'phantomInterrupt' can be remaned as 'dummyInterrupt'.
Just click on the ISR tab and rename it as required and save the project file.

4. Cortex-R5F Hercules Safety MCU – (freeRTOS 예제 코드)



```c
#include "HL_sys_common.h"
#include "FreeRTOS.h"
#include "os_task.h"

#include "HL_het.h"
#include "HL_gio.h"

xTaskHandle xTask1Handle;

void vTask1(void *pvParameters){
   for(;;){
       gioSetBit(hetPORT1, 17, gioGetBit(hetPORT1, 17) ^ 1);
       vTaskDelay(100);
   }
}

int main(void)
{

   gioSetDirection(hetPORT1, 0xFFFFFFFF);

   if(xTaskCreate(vTask1, "Task1", configMINIMAL_STACK_SIZE, NULL, 1, &xTask1Handle) != pdTRUE){
       while( 1);
   }

   vTaskStartScheduler();

   while(1);
   return 0;
}
```