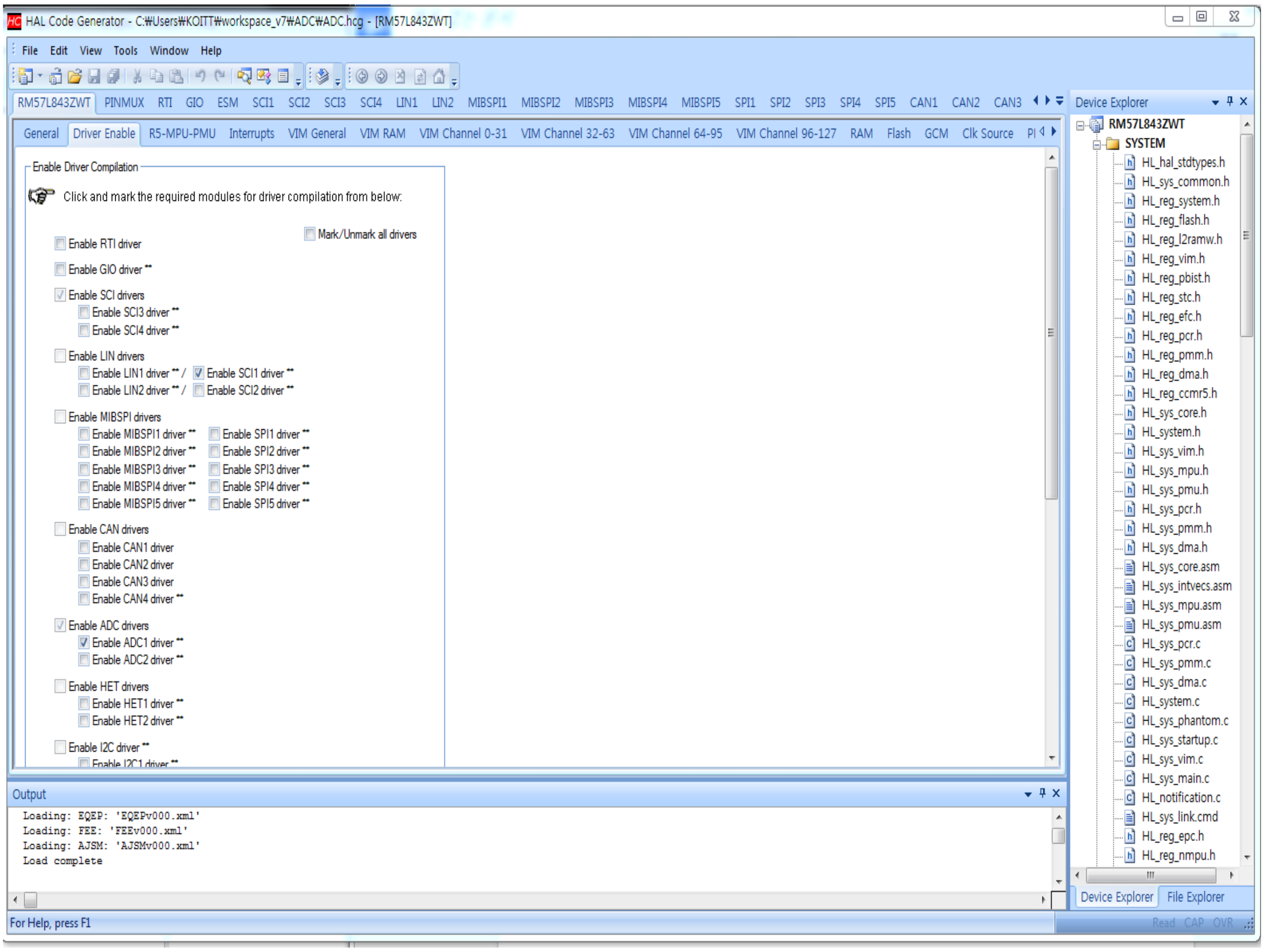


HAL Code Generator - C:₩Users₩KOITT₩workspace_v7₩ADC₩ADC.hcg - [RM57L843ZWT]
File Edit View Tools Window Help
RM57L843ZWT PINMUX RTI GIO ESM SCI1 SCI2 SCI3 SCI4 LIN1 LIN2 MIBSPI1 MIBSPI2 MIBSPI3 MIBSPI4 MIBSPI5 SPI1 SPI2 SPI3 SPI4 SPI5 CAN1 CAN2 CAN3
General Driver Enable R5-MPU-PMU Interrupts VIM General VIM RAM VIM Channel 0-31 VIM Channel 32-63 VIM Channel 64-95 VIM Channel 96-127 RAM Flash GCM Clk Source Pl
Device Explorer
Output
For Help, press F1
Device Explorer File Explorer

Enable Driver Compilation

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

□ Mark/Unmark all drivers

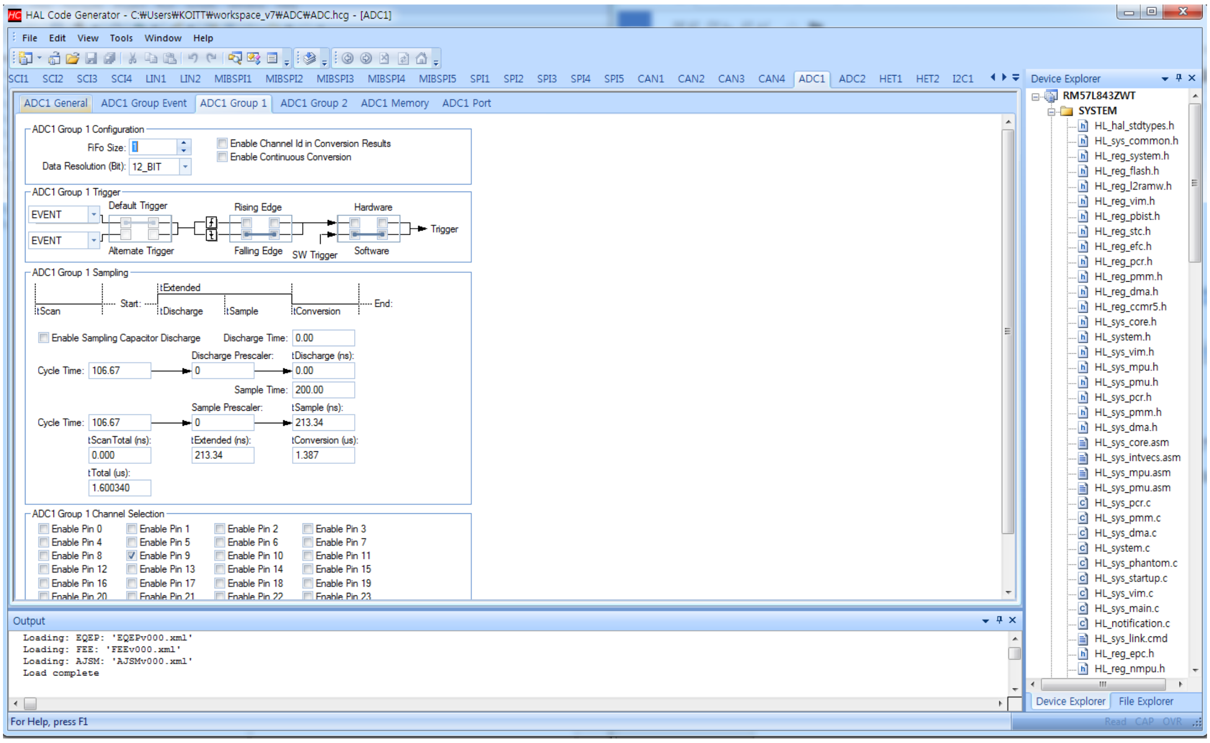
- □ Enable RTI driver
- □ Enable GIO driver **
- ☑ Enable SCI drivers
  - □ Enable SCI3 driver **
  - □ Enable SCI4 driver **
- □ Enable LIN drivers
  - □ Enable LIN1 driver ** / ☑ Enable SCI1 driver **
  - □ Enable LIN2 driver ** / □ Enable SCI2 driver **
- □ Enable MIBSPI drivers
  - □ Enable MIBSPI1 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 **
  - □ Enable HET2 driver **
- □ Enable I2C driver **
  - □ Enable I2C1 driver **

```
Loading: EQEP: 'EQEPv000.xml'
Loading: FEE: 'FEEv000.xml'
Loading: AJSM: 'AJSMv000.xml'
Load complete
```

- RM57L843ZWT
  - SYSTEM
    - HL_hal_stdtypes.h
    - HL_sys_common.h
    - HL_reg_system.h
    - HL_reg_flash.h
    - HL_reg_l2ramw.h
    - HL_reg_vim.h
    - HL_reg_pbist.h
    - HL_reg_stc.h
    - HL_reg_efc.h
    - HL_reg_pcr.h
    - HL_reg_pmm.h
    - HL_reg_dma.h
    - HL_reg_ccmr5.h
    - HL_sys_core.h
    - HL_system.h
    - HL_sys_vim.h
    - HL_sys_mpu.h
    - HL_sys_pmu.h
    - HL_sys_pcr.h
    - HL_sys_pmm.h
    - HL_sys_dma.h
    - HL_sys_core.asm
    - HL_sys_intvecs.asm
    - HL_sys_mpu.asm
    - HL_sys_pmu.asm
    - HL_sys_pcr.c
    - HL_sys_pmm.c
    - HL_sys_dma.c
    - HL_system.c
    - HL_sys_phantom.c
    - HL_sys_startup.c
    - HL_sys_vim.c
    - HL_sys_main.c
    - HL_notification.c
    - HL_sys_link.cmd
    - HL_reg_epc.h
    - HL_reg_nmpu.h

halcogen 설정

Driver Enable 에서 Enable SCI1 driver 선택 , Enable ADC1 driver 선택



ADC1 Group 1 Configuration

FiFo Size: 1

Data Resolution (Bit): 12_BIT

□ Enable Channel Id in Conversion Results

□ Enable Continuous Conversion

ADC1 Group 1 Trigger

EVENT

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: 200.00

Cycle Time: 106.67 → Sample Prescaler: 0 → tSample (ns): 213.34

tScanTotal (ns): 0.000

tExtended (ns): 213.34

tConversion (us): 1.387

tTotal (us): 1.600340

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

Output

```
Loading: EQEP: 'EQEPv000.xml'
Loading: FEE: 'FEEv000.xml'
Loading: AJSM: 'AJSMv000.xml'
Load complete
```

RM57L843ZWT
SYSTEM
HL_hal_stdtypes.h
HL_sys_common.h
HL_reg_system.h
HL_reg_flash.h
HL_reg_l2ramw.h
HL_reg_vim.h
HL_reg_pbist.h
HL_reg_stc.h
HL_reg_efc.h
HL_reg_pcr.h
HL_reg_pmm.h
HL_reg_dma.h
HL_reg_ccmr5.h
HL_sys_core.h
HL_system.h
HL_sys_vim.h
HL_sys_mpu.h
HL_sys_pmu.h
HL_sys_pcr.h
HL_sys_pmm.h
HL_sys_dma.h
HL_sys_core.asm
HL_sys_intvecs.asm
HL_sys_mpu.asm
HL_sys_pmu.asm
HL_sys_pcr.c
HL_sys_pmm.c
HL_sys_dma.c
HL_system.c
HL_sys_phantom.c
HL_sys_startup.c
HL_sys_vim.c
HL_sys_main.c
HL_notification.c
HL_sys_link.cmd
HL_reg_epc.h
HL_reg_nmpu.h

ADC1 에서 ADC1 Group1에서
FiFo size 1로변경 , 아랫쪽에 ADC Group1 Channel Selection 에서 Enable Pin9 선택

generate code 누름.

CCS로 돌아가서 메인 코드 작성.

```
/** @file HL_sys_main.c
*   @brief Application main file
*   @date 08-Feb-2017
*   @version 04.06.01
*
*   This file contains an empty main function,
*   which can be used for the application.
*/

/*
* Copyright (C) 2009-2016 Texas Instruments Incorporated - www.ti.com
*
*
*  Redistribution and use in source and binary forms, with or without
*  modification, are permitted provided that the following conditions
*  are met:
*
*    Redistributions of source code must retain the above copyright
*    notice, this list of conditions and the following disclaimer.
```

```
/* USER CODE BEGIN (0) */
#include "HL_sci.h"
#include "HL_adc.h"
#include "stdlib.h"
/* USER CODE END */

/* Include Files */

#include "HL_sys_common.h"
#include "HL_system.h"
/* USER CODE BEGIN (1) */
unsigned char command[51];
/* USER CODE END */
void sciDisplayText(sciBASE_t *sci, uint8 *command, uint32 length);
void wait(uint32 time);

#define TSIZE2  2
```

```
#define UART sciREG1
/** @fn void main(void)
*   @brief Application main function
*   @note This function is empty by default.
*
*   This function is called after startup.
*   The user can use this function to implement the application.
*/

/* USER CODE BEGIN (2) */
/* USER CODE END */

int main(void)
{
/* USER CODE BEGIN (3) */
     adcData_t adc_data;
     adcData_t *adc_data_ptr = &adc_data;
     unsigned int NumberOfChars, value, temp;



         sciInit();
         adcInit();

         while(1)
         {
             adcStartConversion(adcREG1,adcGROUP1);
             while(!adcIsConversionComplete(adcREG1,adcGROUP1));
             adcGetData(adcREG1,1U,adc_data_ptr);
             value = (unsigned int)adc_data_ptr->value;
             value &= 0xfff;
             NumberOfChars = ltoa(value,(char *)command);
 //            sciSend(sciREG1, 2, (unsigned char *)"0x");
 //            sciSend(sciREG1,NumberOfChars,command);
 //           sciSend(sciREG1, 2,(unsigned char *)"\r\n");
             sciDisplayText(UART,&command[0],TSIZE2);
             //wait(10000);
             /*
             temp=command[0] >>4;
             command[0] -= temp;
             temp= command[0];
             temp = temp << 4;
             temp = temp | TSIZE2 >> 4;
             command[0] += temp;
            */
```

```
        }
}
        void sciDisplayText(sciBASE_t *sci, uint8 *command, uint32 length)
        {
           while(length--)
           {
              while((UART->FLR & 0X4) == 4)
                 ;   /* wait until busy */
              sciSendByte(UART,*command++);  /* send out text */
           }
        }

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

/* USER CODE END */



/* USER CODE BEGIN (4) */
void adcNotification(adcBASE_t *adc,unsigned group)
{
   return ;
}
void sciNotification(sciBASE_t *sci, unsigned flags)
{
   return ;
}
void esmGROUP1Notification(int bit)
{
   return ;
}
void esmGroupt2Notification(int bit)
{
   return ;
}
/* USER CODE END */
```