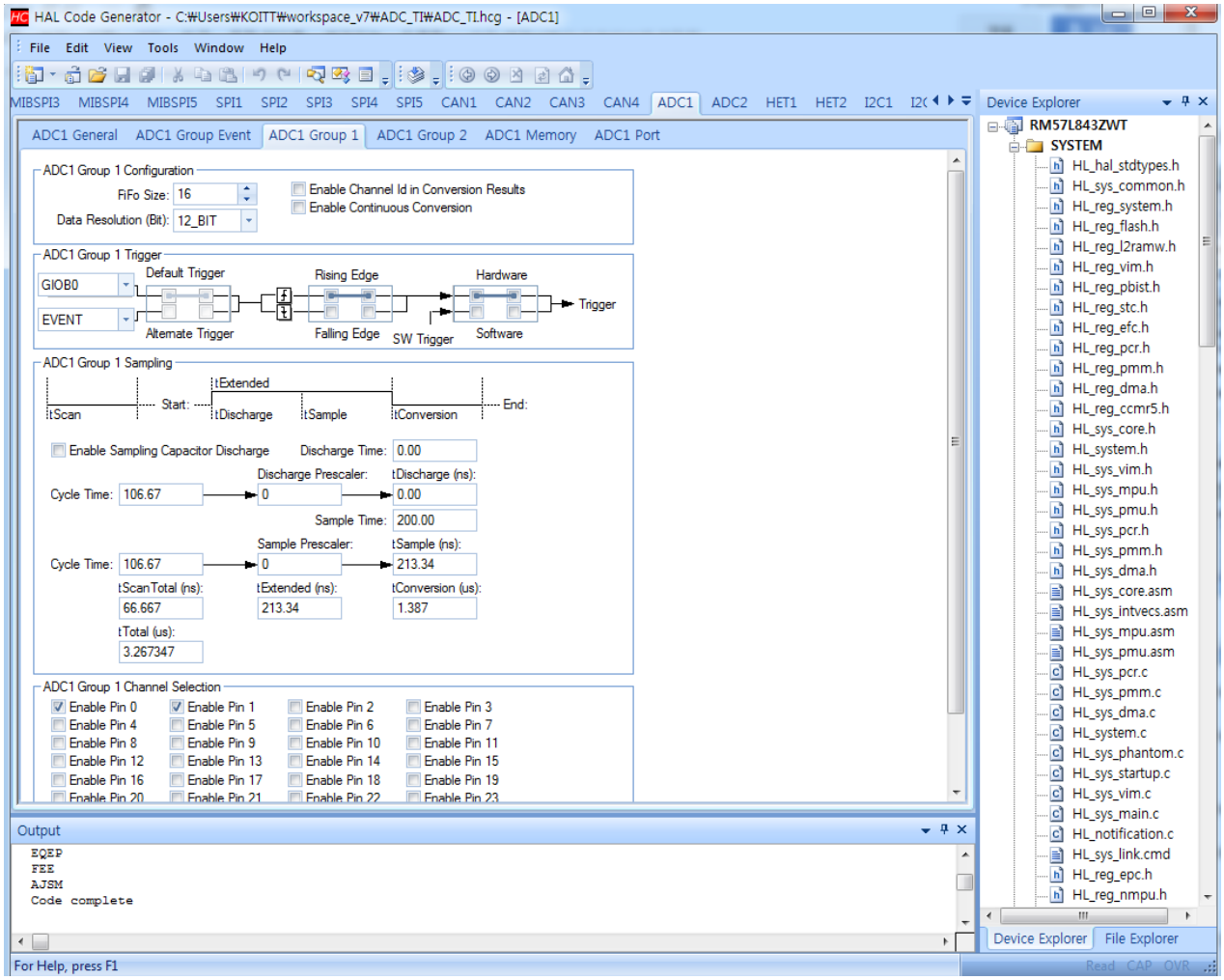
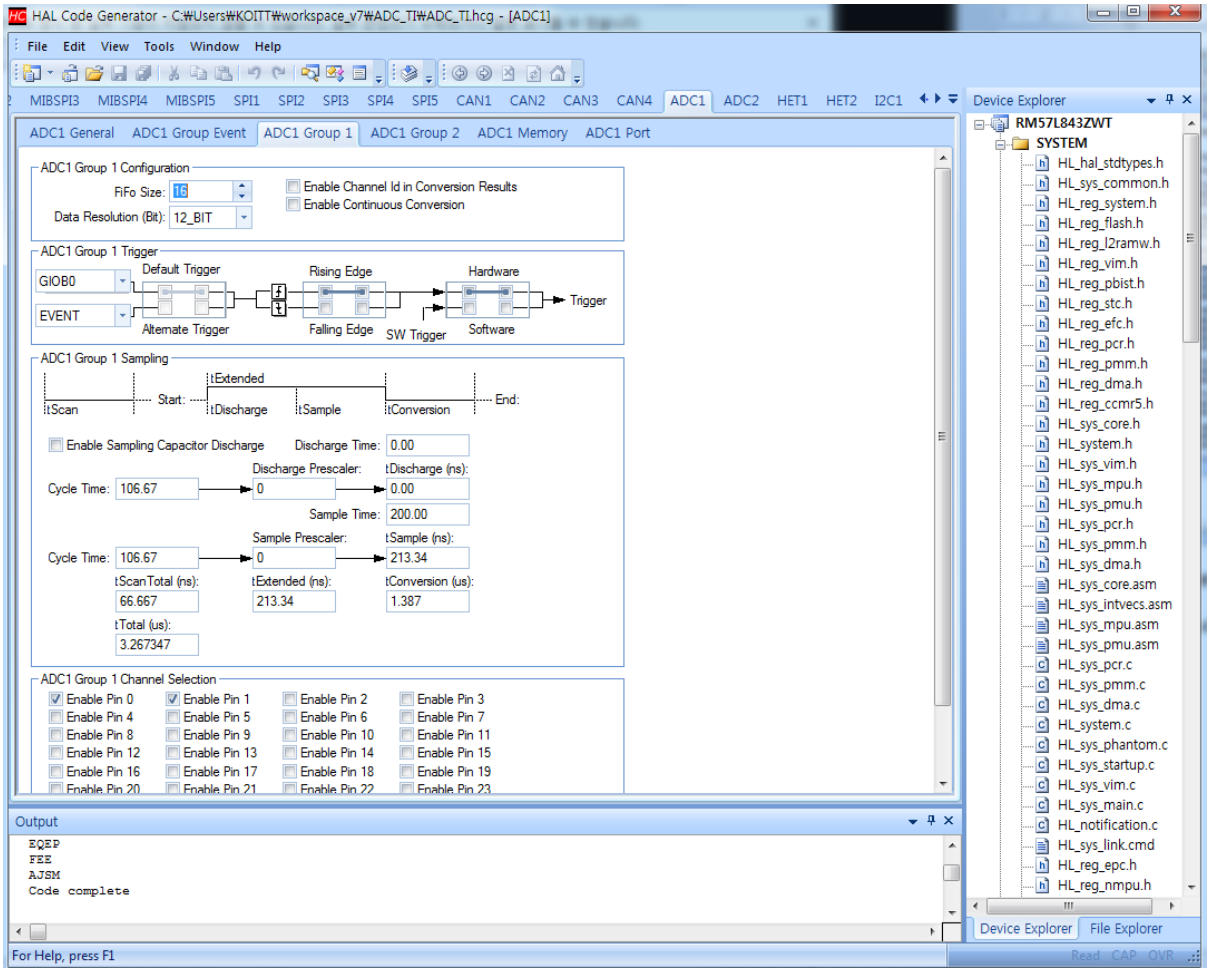


Halcogen 설정하기

1. Driver Enable에서 Enable GIO driver,Enable SCI driver,Enable ADC1 driver 를 체크함.

2. ADC1 으로 이동을 하자.



3. ADC1 Group1으로 이동한다.

4. ADC1 Group 1 Trigger에서 GIOB0을 선택하고, Rising Edge와 Hardware의 trigger edge를 선택한다.

5. ADC1 Group 1 Channel Selection 에서 Enable Pin 0 과 Enable Pin 1을 선택한다.

6.generate code를 한다.

File - new- CCS Project - ADC Project 새롭게 만들어 놓은 CCS 로 이동하자.

1. source - HL_sys_main.c 로 이동하여 아래와 같은 코드를 입력한다.

```
/* Include Files */

#include "HL_sys_common.h"
#include "HL_system.h"


/* USER CODE BEGIN (1) */
#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','.','I','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);
/* USER CODE END */

/** @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 */
void main(void)
{/* USER CODE BEGIN (3) */
    uint32 ch_count=0;
    uint32 id    =0;
    uint32 value =0;

    /* initialize gio     */
    gioInit();
    gioSetDirection(gioPORTB, 1);
```

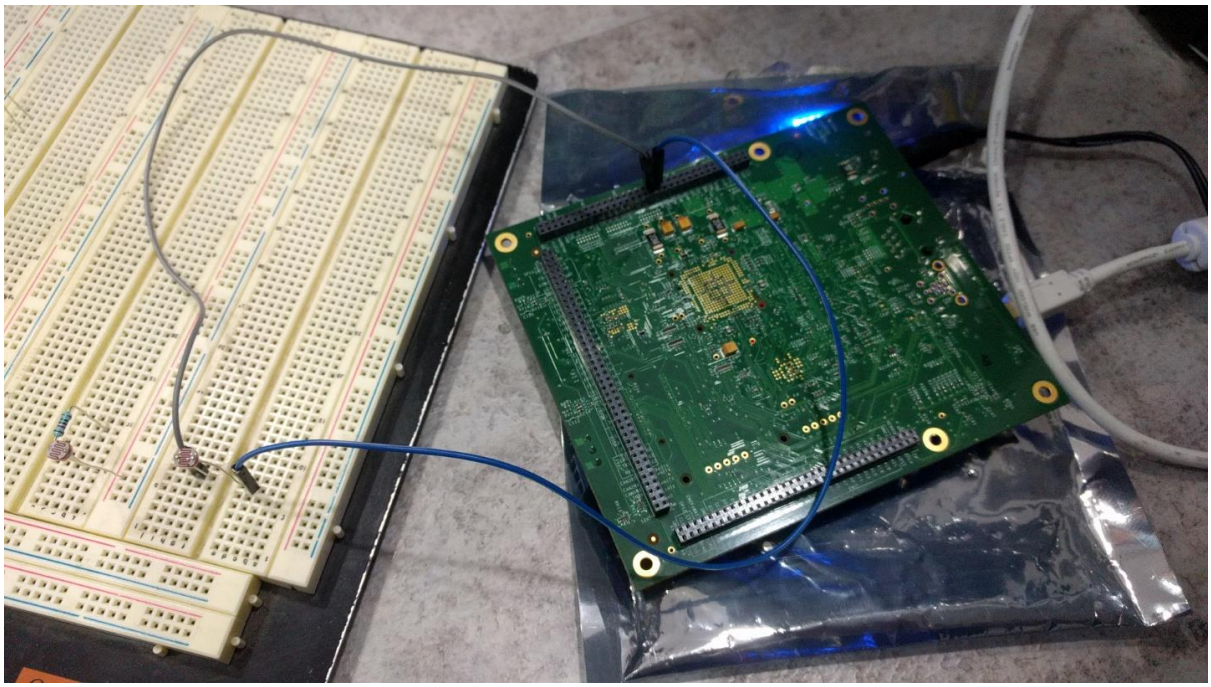
```
    /* initialize sci/sci-lin  : even parity , 2 stop bits */
    sciInit();
    /* initialize ADC  */
    /* Group1 -> Channel 0 and 1                        */
    /* HW trigger trigger source as GIOB  Pin 0         */
    adcInit();

    /* start adc conversion */
    adcStartConversion(adcREG1,adcGROUP1);

    while(1) /* ... continue forever */
    {
        /* trigger using gio port b, pin 0  */
        gioSetBit(gioPORTB, 0, 1);

        /* ... wait and read the conversion count */
        while((adcIsConversionComplete(adcREG1,adcGROUP1))==0);
        ch_count = adcGetData(adcREG1, adcGROUP1,&adc_data[0]);
        ch_count = ch_count;
        /* conversion results :                             */
        /* adc_data[0] -> should have conversions for Group1 channel1 */
        /* adc_data[1] -> should have conversions for Group1 channel2 */

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

        gioSetBit(gioPORTB, 0, 0);
        sciDisplayText(sciREG1,&TEXT1[0],TSIZE1);   /* send text 1 */
        sciDisplayData(sciREG1,(uint8*)&id,4);    /* send data 1 */
        sciDisplayText(sciREG1,&TEXT2[0],TSIZE2);   /* send text 2 */
        sciDisplayData(sciREG1,(uint8*)&value,4); /* send data 2 */

        id    = adc_data[1].id;
        value = adc_data[1].value;

        sciDisplayText(sciREG1,&TEXT1[0],TSIZE1);   /* send text 1 */
        sciDisplayData(sciREG1,(uint8*)&id,4);    /* send data 1 */
        sciDisplayText(sciREG1,&TEXT2[0],TSIZE2);   /* send text 2 */
        sciDisplayData(sciREG1,(uint8*)&value,4); /* send data 2 */

        wait(0xFFFFF);
    };
/* USER CODE END */
}
/* USER CODE BEGIN (4) */
void sciDisplayData(sciBASE_t *sci, uint8 *text,uint32 length)
{
```

```
    uint8 txt = 0;
    uint8 txt1 = 0;
    #if ((__little_endian__ == 1) || (__LITTLE_ENDIAN__ == 1))
    text = text + (length -1);
    #endif

    while(length--)
    {
#if ((__little_endian__ == 1) || (__LITTLE_ENDIAN__ == 1))
        txt = *text--;
#else
        txt = *text++;
#endif
        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;
        }
        while ((sciREG1->FLR & 0x4) == 4); /* wait until busy */
        sciSendByte(sciREG1,txt1);      /* send out text   */
        while ((sciREG1->FLR & 0x4) == 4); /* wait until busy */
        sciSendByte(sciREG1,txt);      /* send out text   */
    };
```

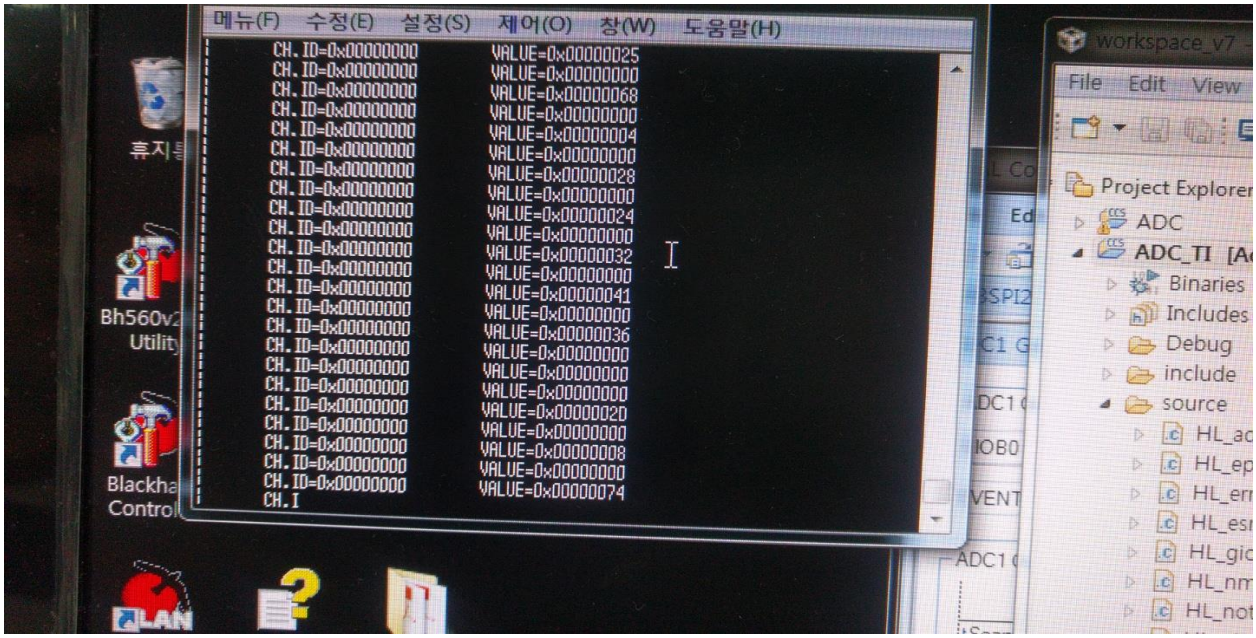
```
}

void sciDisplayText(sciBASE_t *sci, uint8 *text,uint32 length)
{
   while(length--)
   {
      while ((sciREG1->FLR & 0x4) == 4); /* wait until busy */
      sciSendByte(sciREG1,*text++);      /* send out text   */
   };
}

void wait(uint32 time)
{
   while(time){time--;};
}/* USER CODE END */
```

2. 망치를 때리고 확인해보기.

3. 조도센서 연결하고 MCU ADC Pin0 에 선 연결, GND에 선연결을 한다.



4. 보드 연결후, Tera Term을 켜서 확인한다. 자동으로 되기때문에 바로 Value 값을 눈으로 확인할수있다.


메뉴(F) 수정(E) 설정(S) 제어(O) 창(W) 도움말(H)
CH.ID=0x00000000 VALUE=0x00000025
CH.ID=0x00000000 VALUE=0x00000000
CH.ID=0x00000000 VALUE=0x00000068
CH.ID=0x00000000 VALUE=0x00000000
CH.ID=0x00000000 VALUE=0x00000004
CH.ID=0x00000000 VALUE=0x00000000
CH.ID=0x00000000 VALUE=0x00000028
CH.ID=0x00000000 VALUE=0x00000000
CH.ID=0x00000000 VALUE=0x00000024
CH.ID=0x00000000 VALUE=0x00000000
CH.ID=0x00000000 VALUE=0x00000032
CH.ID=0x00000000 VALUE=0x00000000
CH.ID=0x00000000 VALUE=0x00000041
CH.ID=0x00000000 VALUE=0x00000000
CH.ID=0x00000000 VALUE=0x00000036
CH.ID=0x00000000 VALUE=0x00000000
CH.ID=0x00000000 VALUE=0x00000000
CH.ID=0x00000000 VALUE=0x00000000
CH.ID=0x00000000 VALUE=0x0000002D
CH.ID=0x00000000 VALUE=0x00000000
CH.ID=0x00000000 VALUE=0x00000008
CH.ID=0x00000000 VALUE=0x00000000
CH.ID=0x00000000 VALUE=0x00000074
CH.I
workspace_v7
File Edit View
Project Explorer
ADC
ADC_TI
Binaries
Includes
Debug
include
source
LAN