Xilinx Zynq FPGA,TI DSP, MCU 기반의 프로그래밍 전문가 과정

날 짜 : 2018 . 5. 14

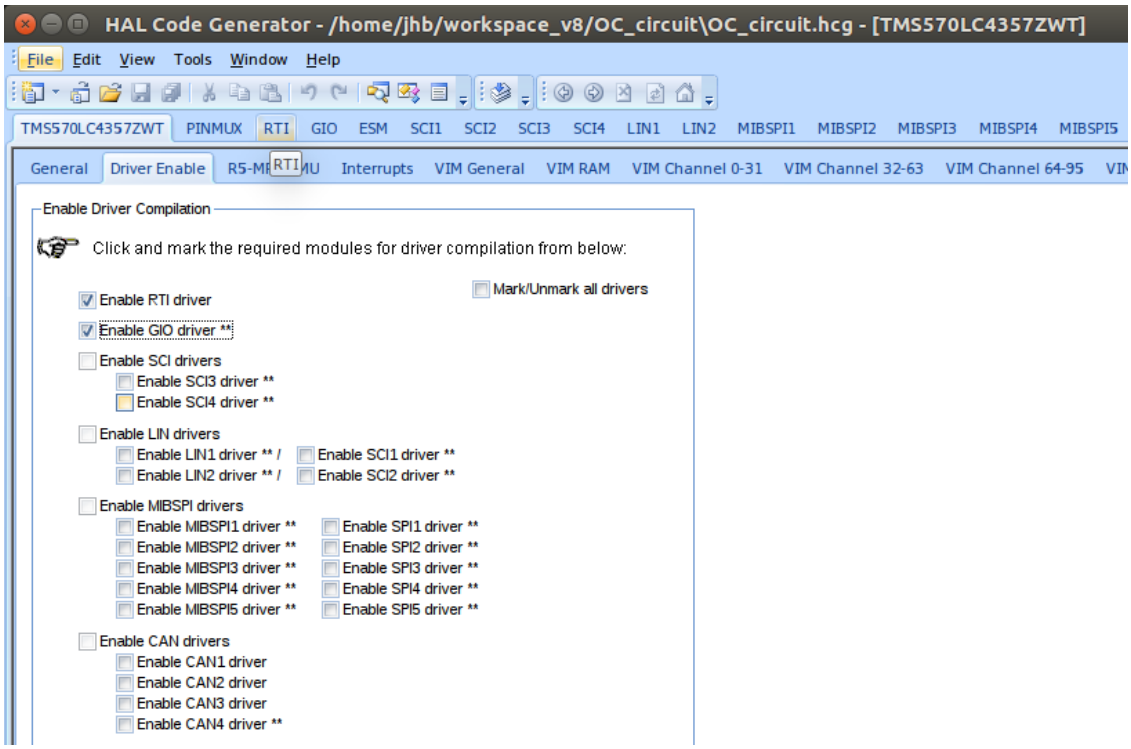
강사 – Innova Lee(이상훈)
gcccompil3r@gmail.com
학생 – 정한별
hanbulkr@gmail.com

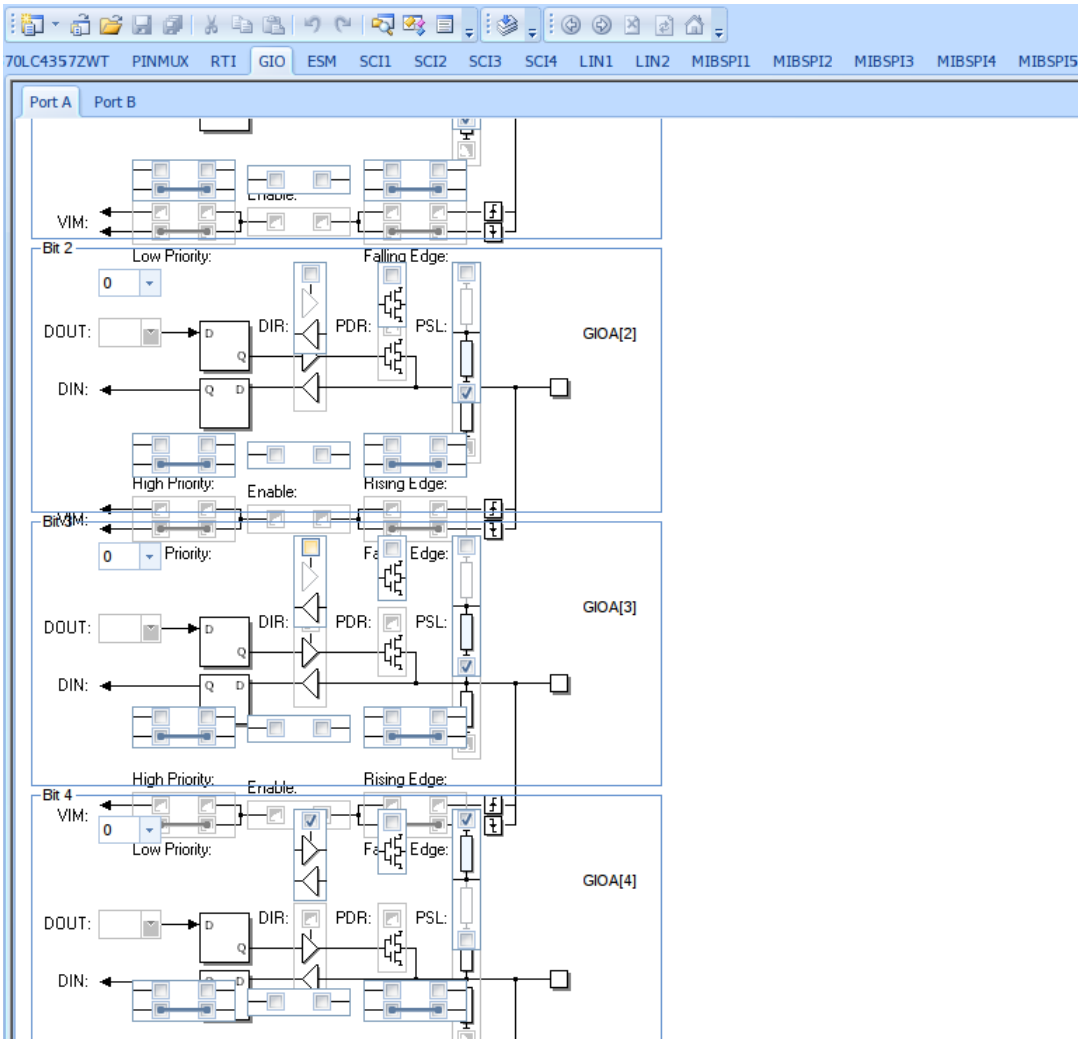
OC_circuit

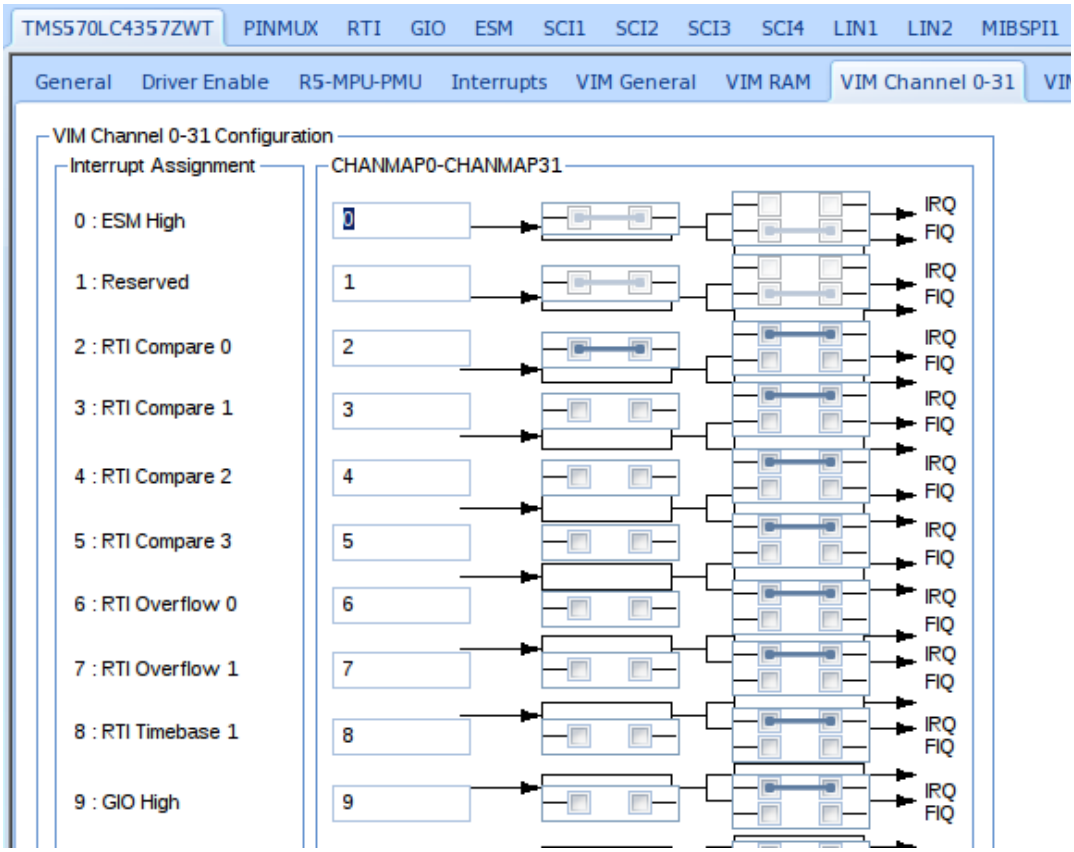
<CODE>

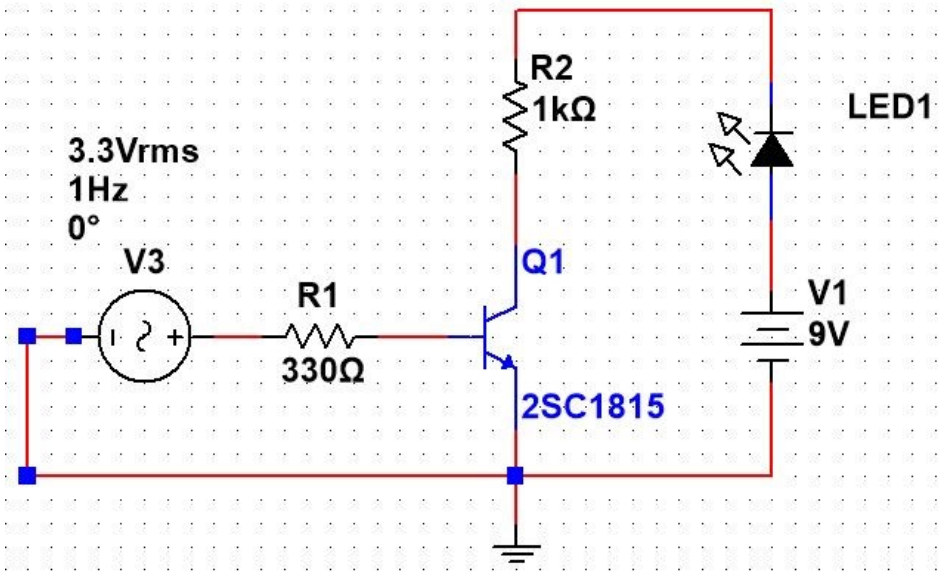
```
#include "HL_sys_common.h"
#include "HL_system.h"

#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"



int main(void)
{
/* USER CODE BEGIN (3) */

    rtiInit();

    gioSetDirection(gioPORTA, 0xFFFFFFFF);

    rtiEnableNotification(rtiREG1, rtiNOTIFICATION_COMPARE0);

    _enable_IRQ_interrupt_();


    rtiStartCounter(rtiREG1, rtiCOUNTER_BLOCK0);


    while(1)

    return 0;
}

void rtiNotification(rtiBASE_t *rtiREG, uint32 notification)
{
    int i=0;
    uint32 x[9]={0x00020000, 0x80000000,0x00000001,
                 0x00000020,0x02000000,0x00040000,0x20000000, 0x08000000,0xAA060021};

    gioSetPort(gioPORTA, gioGetPort(gioPORTA) ^ x[i]);
    i++;
    if(i==8) i=0;
}
```

OC_circuit (HALCOGEN SETTING)









- Open Collector 회로 만들기. 베이스에서 조금의 신호를 흘려주면 스위치가 열려서 9v 가 그라운드와 연결되어 전류가 흐르게 된다.

ADC_UART

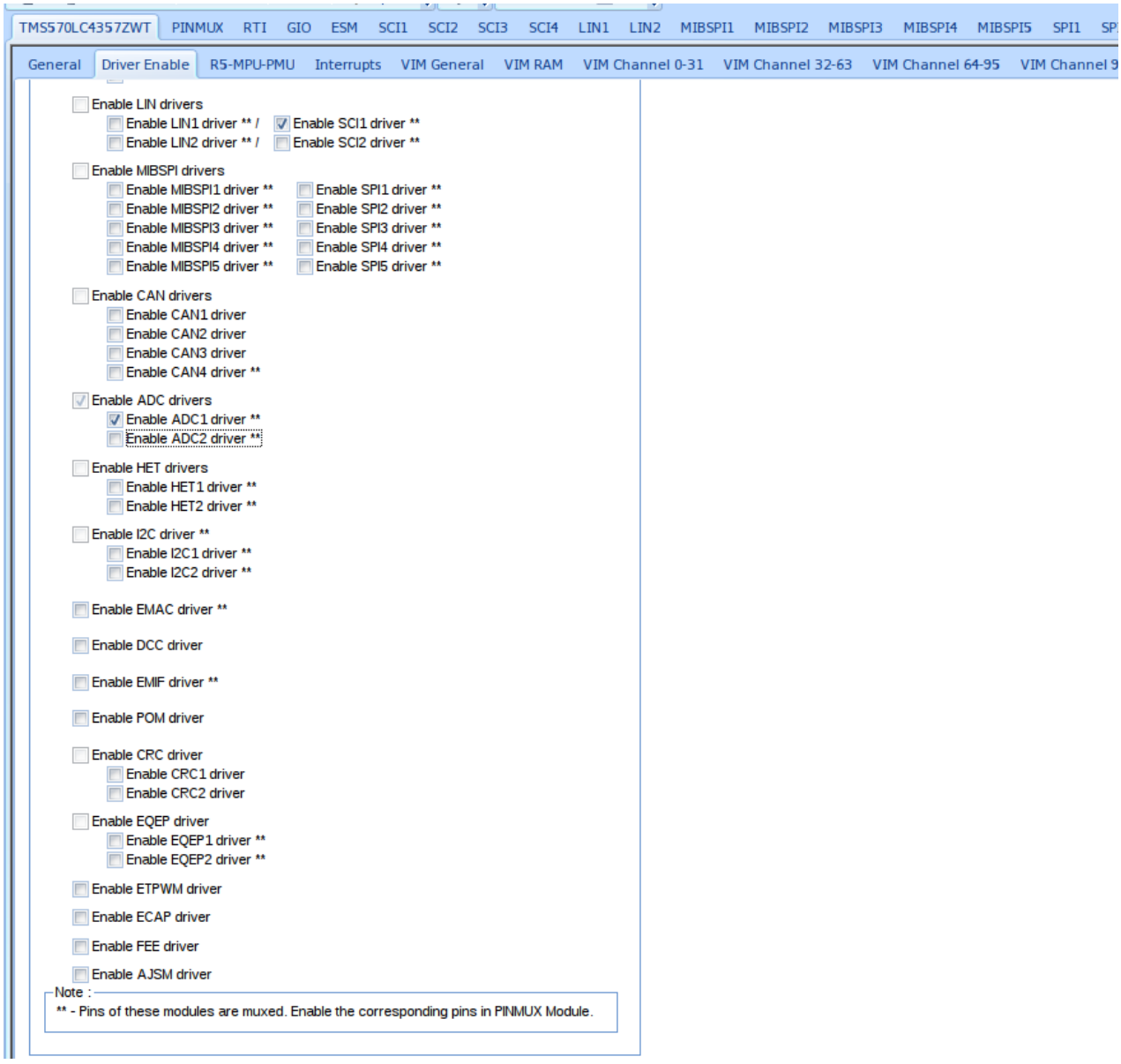
<CODE>

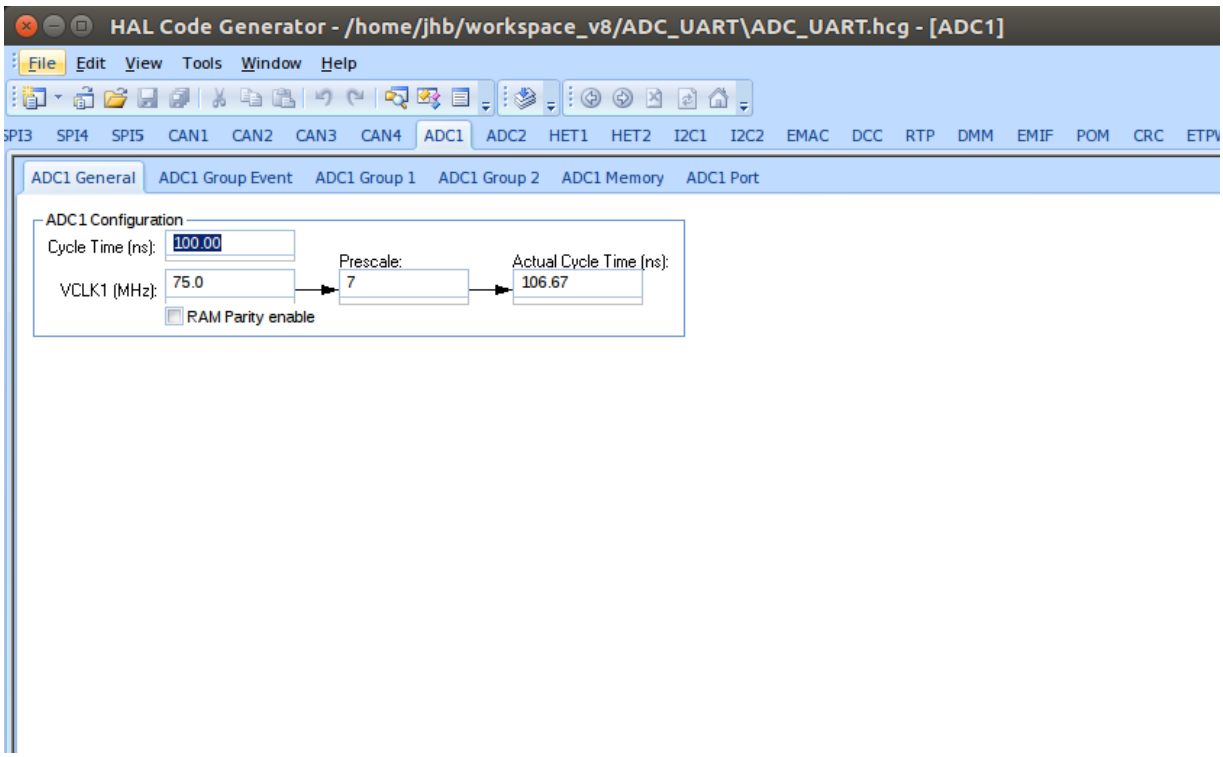
```
#include "HL_sys_common.h"
#include "HL_system.h"

#include "HL_sci.h"
#include "HL_esm.h"
#include "HL_adc.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);

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

    gioInit();
    gioSetDirection(gioPORTB, 0xFF);

    sciInit();

    adcInit();
    adcStartConversion(adcREG1, adcGROUP1);

    while (1)
    {
        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;

        gioSetBit(gioPORTB, 0, 0);

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

        if (value > 0xE00)
        {
            gioSetBit(gioPORTB, 4, 1);
        }
        else
        {
            gioSetBit(gioPORTB, 4, 0);
        }
        /*
```

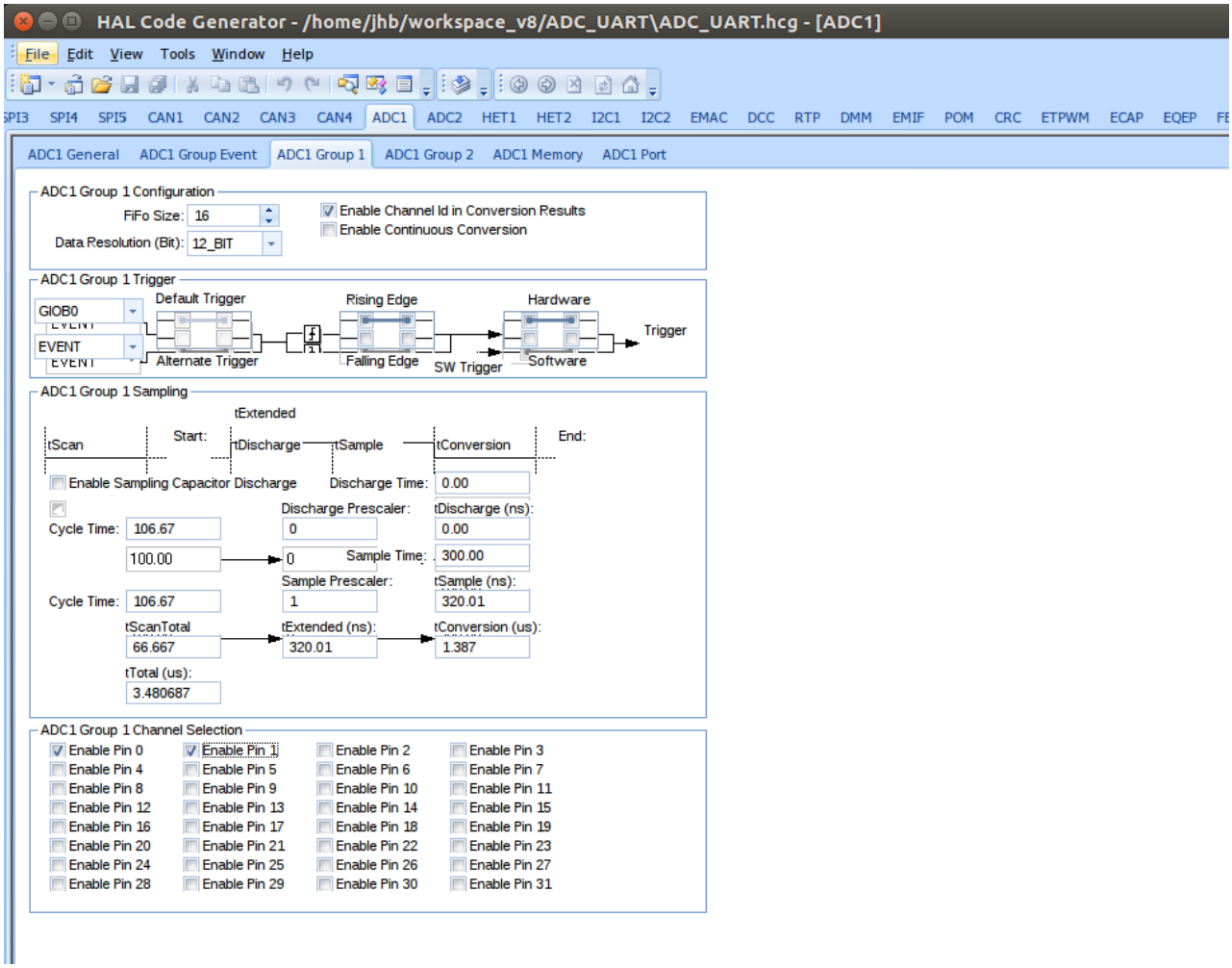
```
         id = adc_data[1].id;
         value = adc_data[1].value;
         59

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

    }

}

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;

#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)
            ;
        sciSendByte(sciREG1, txt1);

        while ((sciREG1->FLR & 0x4) == 4)
            ;
        sciSendByte(sciREG1, txt);
    }
}

void wait(uint32 time)
{
    int i;
    for (i = 0; i < time; i++)
        ;
}
```

ADC_UART (HALCOGEN SETTING)




HAL Code Generator - /home/jhb/workspace_v8/ADC_UART\ADC_UART.hcg - [ADC1]
File Edit View Tools Window Help
SPI3 SPI4 SPI5 CAN1 CAN2 CAN3 CAN4 ADC1 ADC2 HET1 HET2 I2C1 I2C2 EMAC DCC RTP DMM EMIF POM CRC ETPW
ADC1 General ADC1 Group Event ADC1 Group 1 ADC1 Group 2 ADC1 Memory ADC1 Port
ADC1 Configuration
Cycle Time (ns): 100.00
VCLK1 (MHz): 75.0
Prescale: 7
Actual Cycle Time (ns): 106.67
RAM Parity enable


HAL Code Generator - /home/jhb/workspace_v8/ADC_UART\ADC_UART.hcg - [ADC1]
File Edit View Tools Window Help
SPI3 SPI4 SPI5 CAN1 CAN2 CAN3 CAN4 ADC1 ADC2 HET1 HET2 I2C1 I2C2 EMAC DCC RTP DMM EMIF POM CRC ETPWM ECAP EQEP
ADC1 General ADC1 Group Event ADC1 Group 1 ADC1 Group 2 ADC1 Memory ADC1 Port
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
Default Trigger
Alternate Trigger
Rising Edge
Falling Edge
Hardware
SW Trigger
Software
Trigger
ADC1 Group 1 Sampling
tExtended
tScan
Start:
tDischarge
tSample
tConversion
End:
Enable Sampling Capacitor Discharge
Discharge Time: 0.00
Discharge Prescaler: 0
tDischarge (ns): 0.00
Cycle Time: 106.67
100.00
Sample Time: 300.00
Sample Prescaler: 1
tSample (ns): 320.01
Cycle Time: 106.67
tScanTotal 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