TI DSP, MCU 및 Xilinx Zynq FPGA 프로그래밍 전문가 과정

MCU Peripheral

강사 : Innova Lee(이상훈)
gcccompil3r@gmail.com
학생 : 황수정
sue100012@naver.com

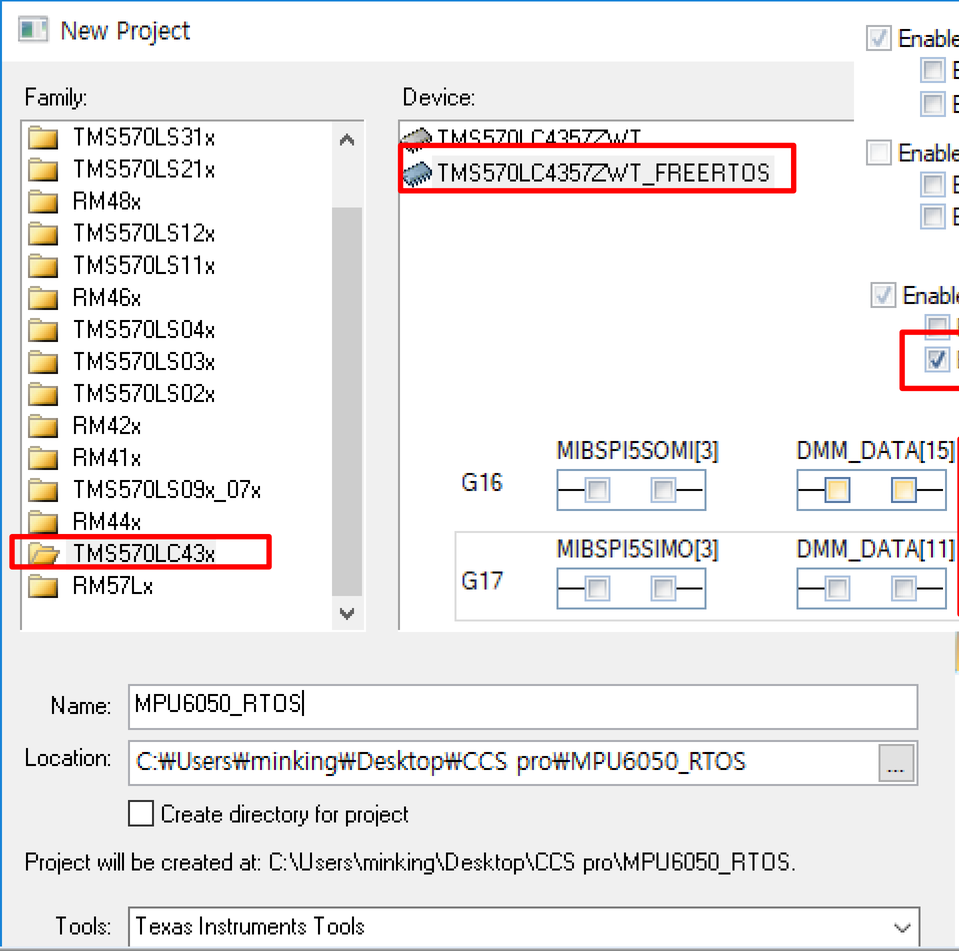
2018. 08.15

RTOS 개념 및 예제

RTOS_MPU6050

MCU Peripheral
- 전조등
- 방향지시등, 후미등

RTOS_MPU6050 Setting



New Project

Family:

- TMS570LS31x
- TMS570LS21x
- RM48x
- TMS570LS12x
- TMS570LS11x
- RM46x
- TMS570LS04x
- TMS570LS03x
- TMS570LS02x
- RM42x
- RM41x
- TMS570LS09x_07x
- RM44x
- TMS570LC43x
- RM57Lx

Device:

- TMS570LC4357ZWT
- TMS570LC4357ZWT_FREERTOS

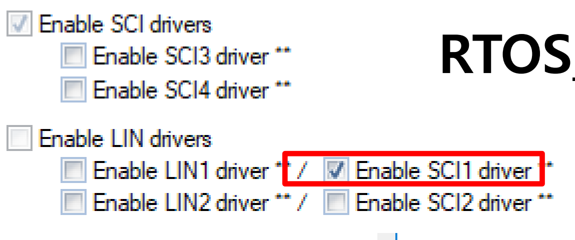
Name: MPU6050_RTOS

Location: C:₩Users₩minking₩Desktop₩CCS pro₩MPU6050_RTOS

☐ Create directory for project

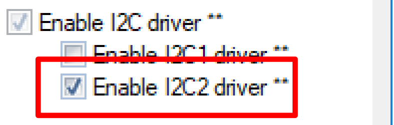
Project will be created at: C:\Users\minking\Desktop\CCS pro\MPU6050_RTOS.

Tools: Texas Instruments Tools

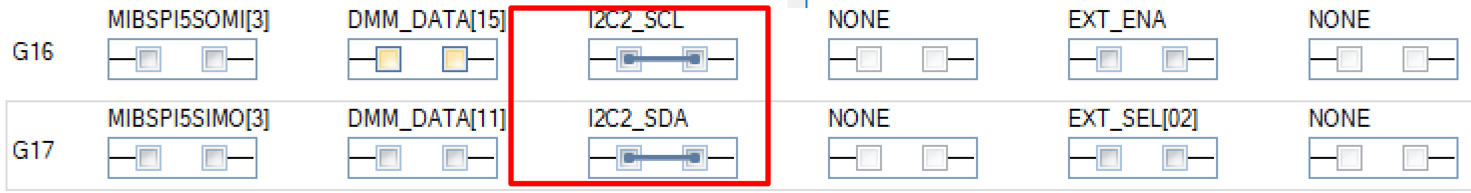


☑ Enable SCI drivers
- ☐ Enable SCI3 driver **
- ☐ Enable SCI4 driver **

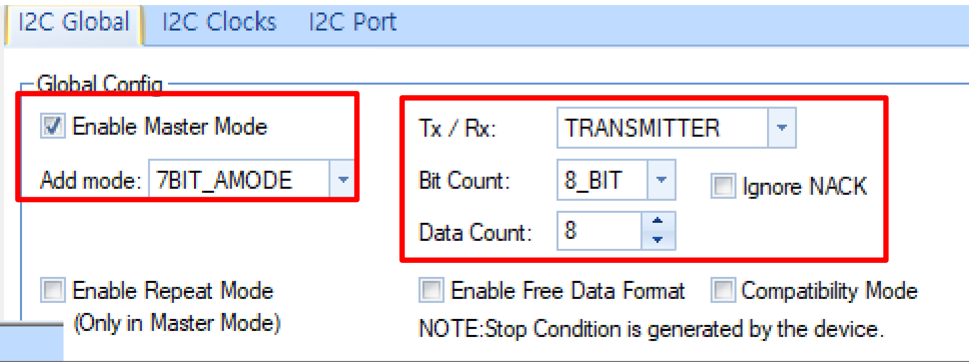
☐ Enable LIN drivers
- ☐ Enable LIN1 driver ** / ☑ Enable SCI1 driver **
- ☐ Enable LIN2 driver ** / ☐ Enable SCI2 driver **



☑ Enable I2C driver **
- ☐ Enable I2C1 driver **
- ☑ Enable I2C2 driver **



G16	MIBSPI5SOMI[3]	DMM_DATA[15]	I2C2_SCL	NONE	EXT_ENA	NONE
G17	MIBSPI5SIMO[3]	DMM_DATA[11]	I2C2_SDA	NONE	EXT_SEL[02]	NONE



I2C Global | I2C Clocks | I2C Port

Global Config

☑ Enable Master Mode

Add mode: 7BIT_AMODE

Tx / Rx: TRANSMITTER

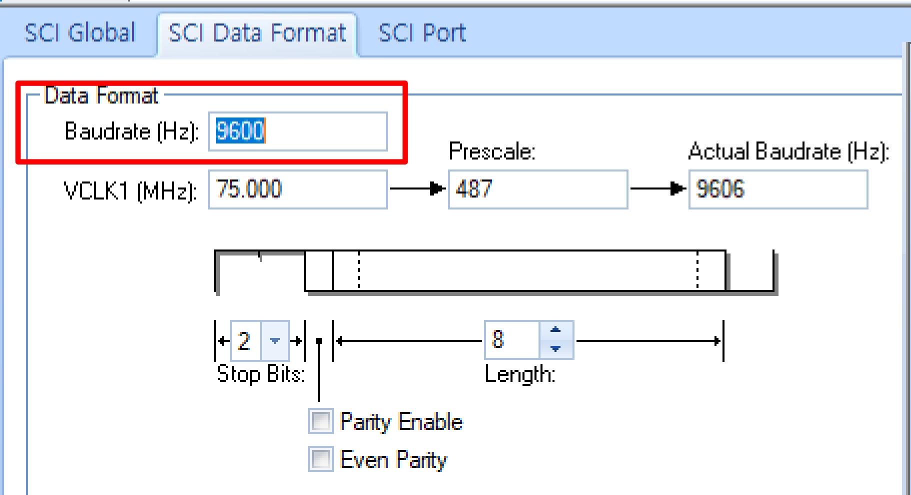
Bit Count: 8_BIT ☐ Ignore NACK

Data Count: 8

☐ Enable Repeat Mode (Only in Master Mode)

☐ Enable Free Data Format ☐ Compatibility Mode

NOTE:Stop Condition is generated by the device.



SCI Global | SCI Data Format | SCI Port

Data Format

Baudrate (Hz): 9600

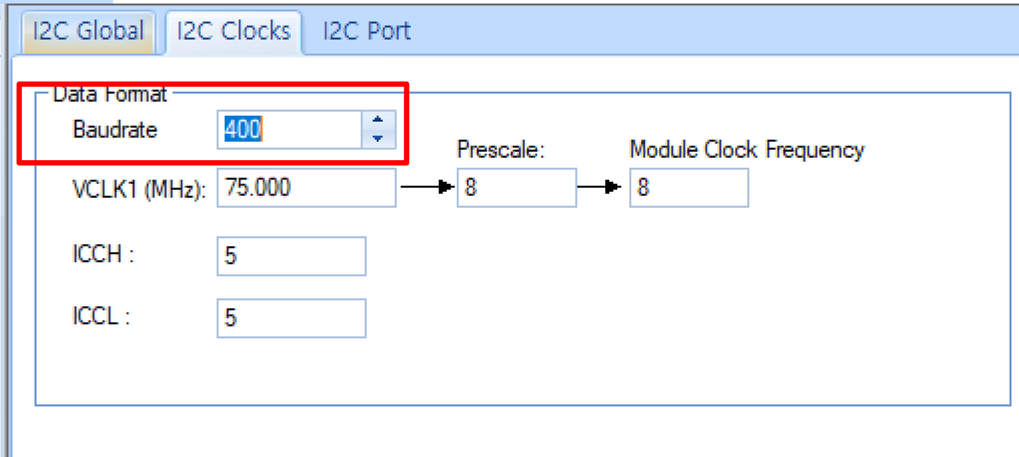
VCLK1 (MHz): 75.000 → Prescale: 487 → Actual Baudrate (Hz): 9606

Stop Bits: 2

Length: 8

☐ Parity Enable

☐ Even Parity



I2C Global | I2C Clocks | I2C Port

Data Format

Baudrate: 400

VCLK1 (MHz): 75.000 → Prescale: 8 → Module Clock Frequency: 8

ICCH : 5

ICCL : 5

```
#include <FreeRTOS.h>
#include <FreeRTOSConfig.h>
#include "HL_sys_common.h"
#include "HL_system.h"
#include "HL_sci.h"
#include "HL_i2c.h"
#include <string.h>
#include <stdio.h>
#include "HL_sys_common.h"
#include "FreeRTOS.h"
#include "os_task.h“

#define UART            sciREG1
#define MPU6050_ADDR    0x68

char txt_buf[256] = { 0 };
unsigned int buf_len;
volatile int i;
signed short acc_x, acc_y, acc_z;
double real_acc_x, real_acc_y, real_acc_z;
void sciDisplayText(sciBASE_t *sci, uint8 *text, uint32 len);
void wait(uint32 delay);
void get_data();
void MPU6050_enable(void);
void MPU6050_acc_config(void);
void disp_set(char *);
uint32 rx_data = 0;
uint32 tmp = 0;
uint32 value = 0;
volatile char g_acc_xyz[6];
volatile int g_acc_flag;
#define IDX     2
uint32 duty_arr[IDX] = { 1000, 2000 };
xTaskHandle xTask1Handle;
```

```
void vTask1(void *pbParameters)
{
    for (;;)
    {
        wait(10000000);
        get_data();
        {
            acc_x = acc_y = acc_z = 0;
            real_acc_x = real_acc_y = real_acc_z = 0.0;
            acc_x = g_acc_xyz[0];
            acc_x = acc_x << 8;
            acc_x |= g_acc_xyz[1];
            real_acc_x = ((double) acc_x) / 2048.0;
            acc_y = g_acc_xyz[2];
            acc_y = acc_y << 8;
            acc_y |= g_acc_xyz[3];
            real_acc_y = ((double) acc_y) / 2048.0;
            acc_z = g_acc_xyz[4];
            acc_z = acc_z << 8;
            acc_z |= g_acc_xyz[5];
            real_acc_z = ((double) acc_z) / 2048.0;
sprintf(txt_buf,"acc_x  %2.5lf\tacc_y  %2.5lf\tacc_z= %2.5lf\n\r\0",real_acc_x,real_acc_y,real_acc_z);
            buf_len = strlen(txt_buf);
            wait(1000000);
            sciDisplayText(sciREG1, (uint8 *) txt_buf, buf_len);
         }
    }
}
```

RTOS_MPU6050 Code

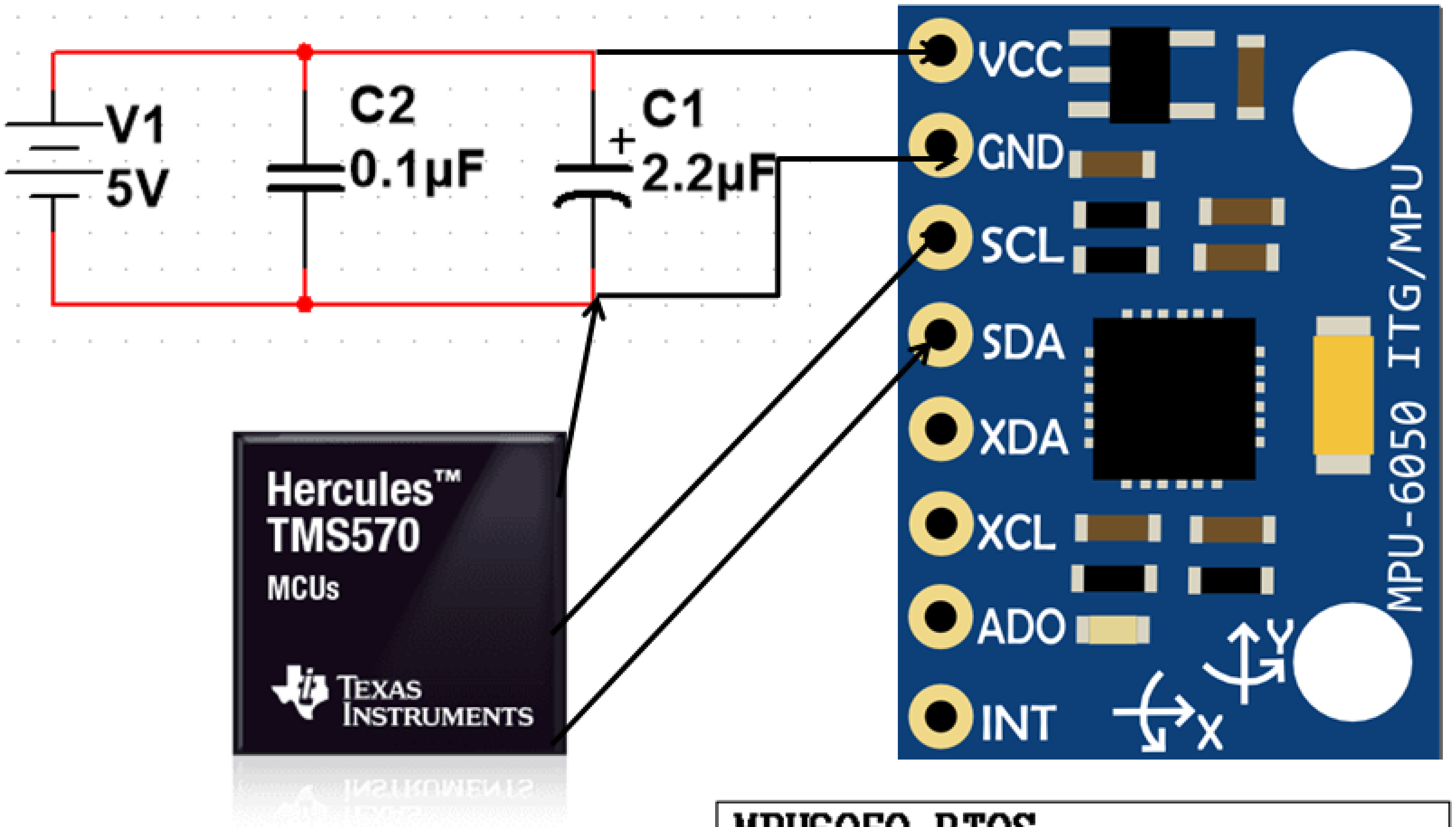
```
int main(void)
{
    sciInit();
    disp_set("SCI Configuration Success!!\n\r\0");
    i2cInit();
    wait(10000000);
    disp_set("I2C Init Success!!\n\r\0");
    MPU6050_enable();
    disp_set("MPU6050 Enable Success!!\n\r\0");
    MPU6050_acc_config();
    disp_set("MPU6050 Accelerometer Configure Success!!\n\r\0");
    if (xTaskCreate(vTask1, "task1", configMINIMAL_STACK_SIZE * 8, NULL,
1,&xTask1Handle) != pdTRUE)

    {
        while (1)
            ;
    }

    vTaskStartScheduler();

    while (1)
        ;
    return 0;
}
void wait(uint32 delay)
{
    int i;
    for (i = 0; i < delay; i++)
        ;
}
void sciDisplayText(sciBASE_t *sci, uint8 *text, uint32 len)
{
    while (len--)
    {
        while ((UART->FLR & 0x4) == 4)
            ;
        sciSendByte(UART, *text++);
    }

}
```

```
void MPU6050_enable(void)
{
    volatile unsigned int cnt = 2;
    unsigned char data[2] = { 0x00U, 0x00U };
    unsigned char slave_word_address = 0x6bU;
    i2cSetSlaveAdd(i2cREG2, MPU6050_ADDR);
    i2cSetDirection(i2cREG2, I2C_TRANSMITTER);
    i2cSetCount(i2cREG2, cnt + 1);
    i2cSetMode(i2cREG2, I2C_MASTER);
    i2cSetStop(i2cREG2);
    i2cSetStart(i2cREG2);
    i2cSendByte(i2cREG2, slave_word_address);
    disp_set("MPU6050 tmp 1 Enable Success!!\n\r\0");
    i2cSend(i2cREG2, cnt, data);
    disp_set("MPU6050 tmp 2Enable Success!!\n\r\0");
    while (i2cIsBusBusy(i2cREG2) == true)
        ;
    while (i2cIsStopDetected(i2cREG2) == 0)
        ;
    i2cClearSCD(i2cREG2);
    wait(100000);
}

void MPU6050_acc_config(void)
{
    volatile unsigned int cnt = 1;
    unsigned char data[1] = { 0x18U };
    unsigned char slave_word_address = 0x1cU;
    i2cSetSlaveAdd(i2cREG2, MPU6050_ADDR);
    i2cSetDirection(i2cREG2, I2C_TRANSMITTER);
    i2cSetCount(i2cREG2, cnt + 1);
    i2cSetMode(i2cREG2, I2C_MASTER);
    i2cSetStop(i2cREG2);
    i2cSetStart(i2cREG2);
    i2cSendByte(i2cREG2, slave_word_address);
    i2cSend(i2cREG2, cnt, data);
    while (i2cIsBusBusy(i2cREG2) == true)
        ;
    while (i2cIsStopDetected(i2cREG2) == 0)
        ;
    i2cClearSCD(i2cREG2);
    wait(1000000);
}
```

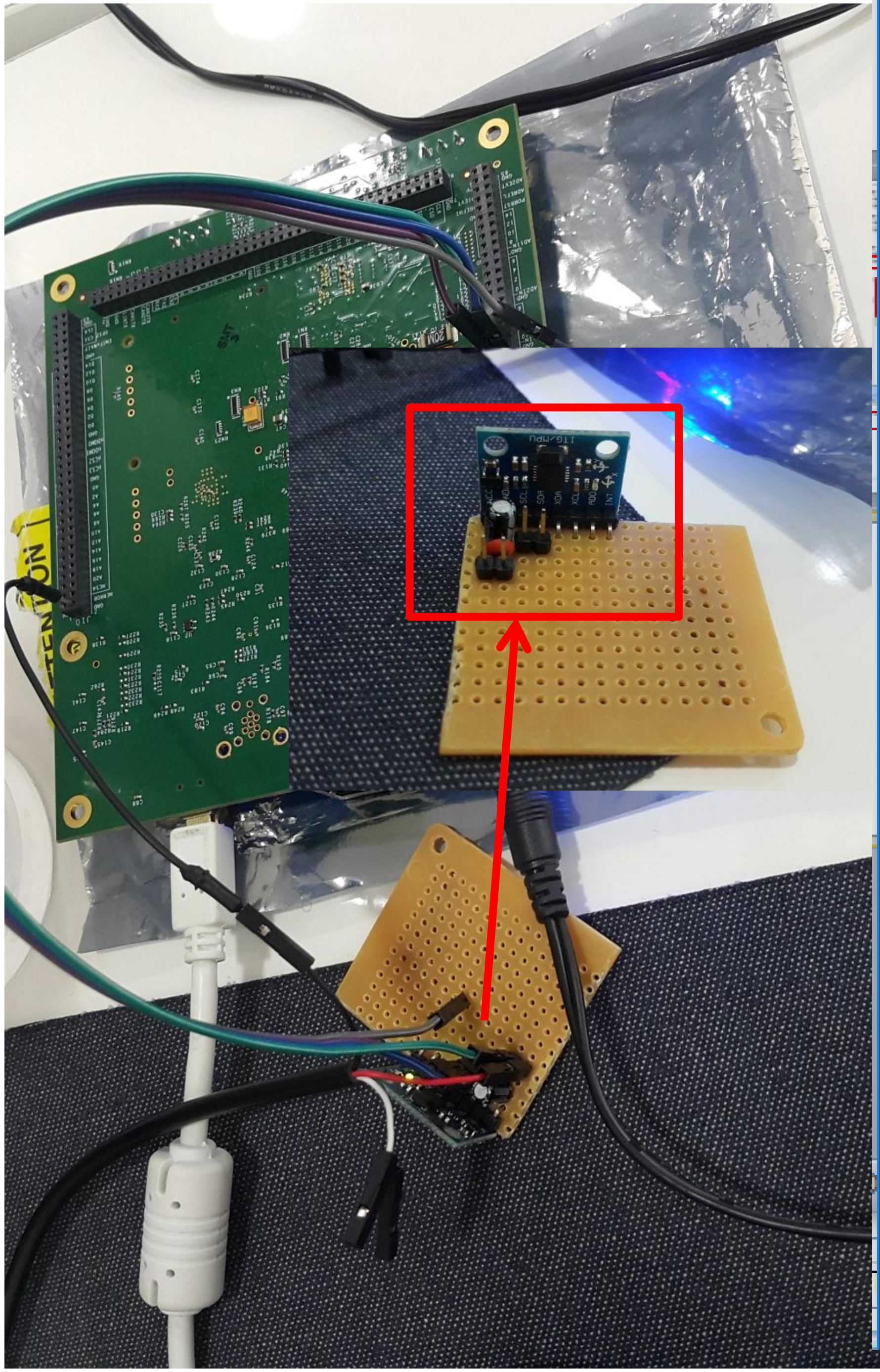
```
void get_data()

{

    unsigned char slave_word_address = 0x3B;
    wait(1000000);
    i2cSetSlaveAdd(i2cREG2, MPU6050_ADDR);
    i2cSetDirection(i2cREG2, I2C_TRANSMITTER);
    i2cSetCount(i2cREG2, 1);
    i2cSetMode(i2cREG2, I2C_MASTER);
    i2cSetStop(i2cREG2);
    i2cSetStart(i2cREG2);
    i2cSendByte(i2cREG2, slave_word_address);
    while (i2cIsBusBusy(i2cREG2) == true)
        ;
    while (i2cIsStopDetected(i2cREG2) == 0)
        ;
    i2cClearSCD(i2cREG2);
    i2cSetDirection(i2cREG2, I2C_RECEIVER);
    i2cSetCount(i2cREG2, 6);
    i2cSetMode(i2cREG2, I2C_MASTER);
    i2cSetStart(i2cREG2);
    i2cReceive(i2cREG2, 6, (unsigned char *) g_acc_xyz);
    i2cSetStop(i2cREG2);
    while (i2cIsBusBusy(i2cREG2) == true)
        ;
    while (i2cIsStopDetected(i2cREG2) == 0)
        ;
    i2cClearSCD(i2cREG2);
    g_acc_flag = 1;
}
```

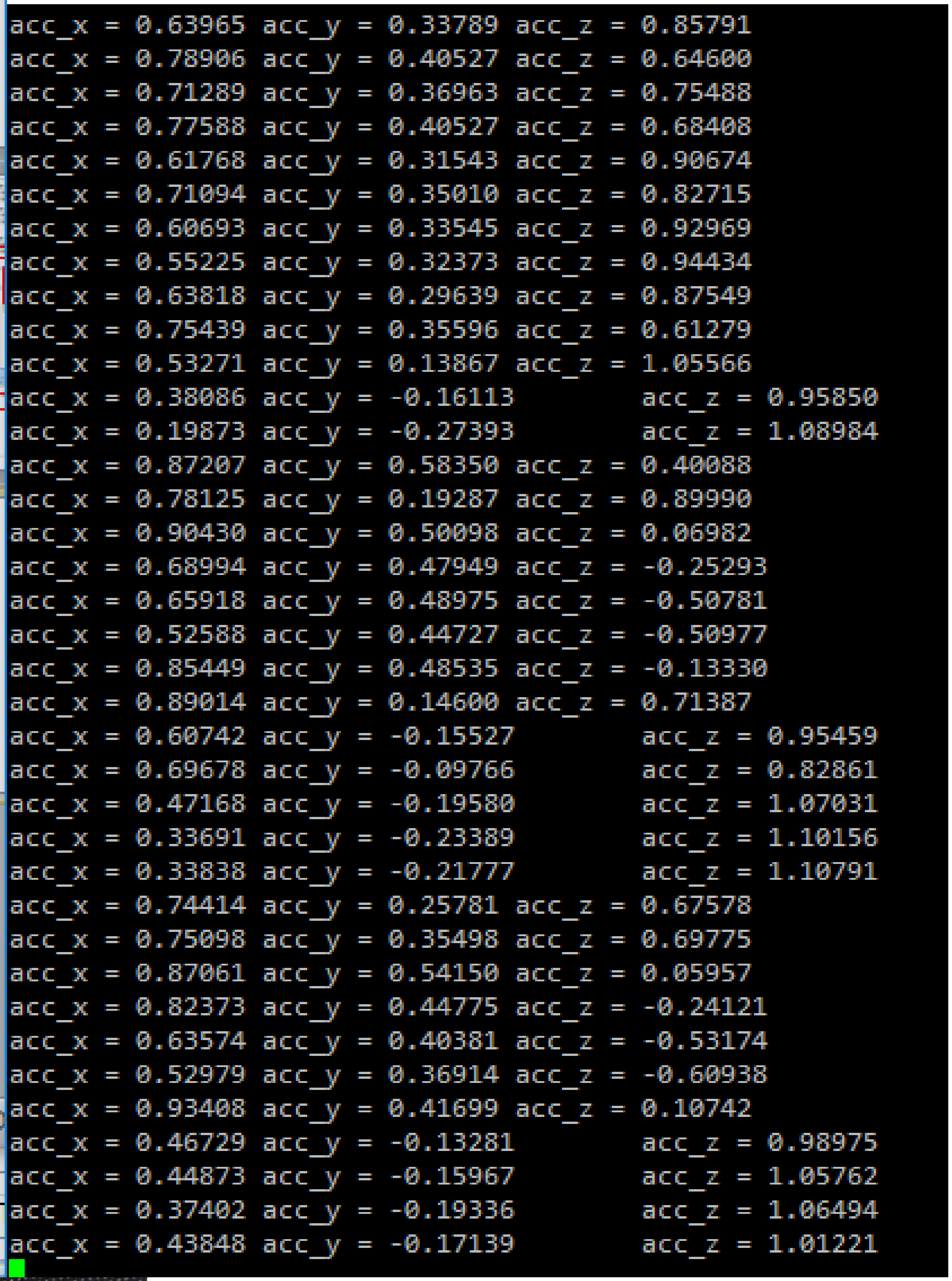
```
void disp_set(char *str)

{
    char txt_buf[256] = { 0 };
    unsigned int buf_len;
    sprintf(txt_buf, str);
    buf_len = strlen(txt_buf);
    sciDisplayText(sciREG1, (uint8 *) txt_buf, buf_len);
    wait(100000);
}
```

RTOS_MPU6050 Code

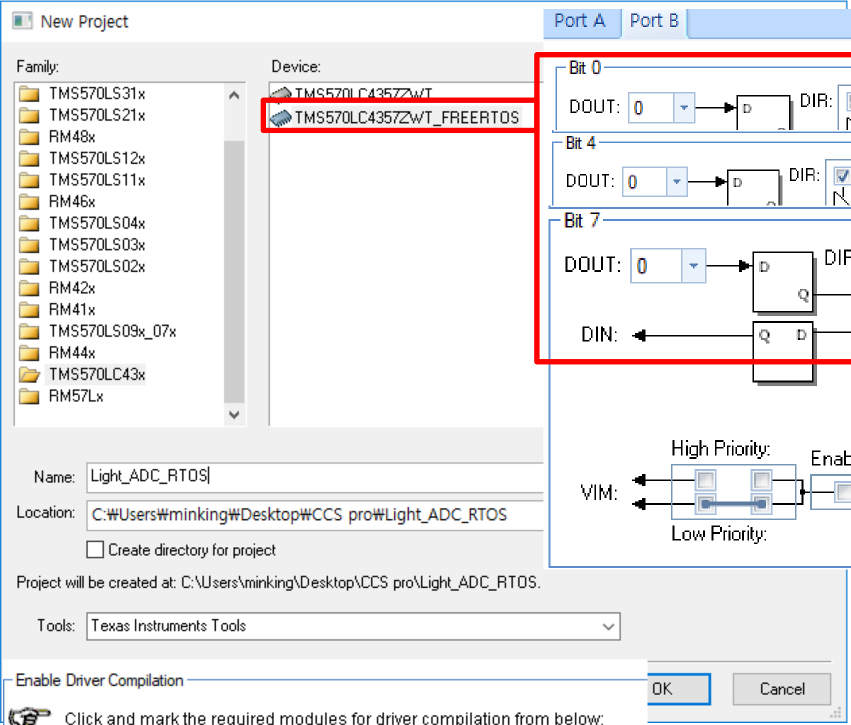


MPU6050_RTOS
Designer : 황수정 , 김민호
2018.08.14
Project AI CAR





```
acc_x = 0.63965 acc_y = 0.33789 acc_z = 0.85791
acc_x = 0.78906 acc_y = 0.40527 acc_z = 0.64600
acc_x = 0.71289 acc_y = 0.36963 acc_z = 0.75488
acc_x = 0.77588 acc_y = 0.40527 acc_z = 0.68408
acc_x = 0.61768 acc_y = 0.31543 acc_z = 0.90674
acc_x = 0.71094 acc_y = 0.35010 acc_z = 0.82715
acc_x = 0.60693 acc_y = 0.33545 acc_z = 0.92969
acc_x = 0.55225 acc_y = 0.32373 acc_z = 0.94434
acc_x = 0.63818 acc_y = 0.29639 acc_z = 0.87549
acc_x = 0.75439 acc_y = 0.35596 acc_z = 0.61279
acc_x = 0.53271 acc_y = 0.13867 acc_z = 1.05566
acc_x = 0.38086 acc_y = -0.16113         acc_z = 0.95850
acc_x = 0.19873 acc_y = -0.27393         acc_z = 1.08984
acc_x = 0.87207 acc_y = 0.58350 acc_z = 0.40088
acc_x = 0.78125 acc_y = 0.19287 acc_z = 0.89990
acc_x = 0.90430 acc_y = 0.50098 acc_z = 0.06982
acc_x = 0.68994 acc_y = 0.47949 acc_z = -0.25293
acc_x = 0.65918 acc_y = 0.48975 acc_z = -0.50781
acc_x = 0.52588 acc_y = 0.44727 acc_z = -0.50977
acc_x = 0.85449 acc_y = 0.48535 acc_z = -0.13330
acc_x = 0.89014 acc_y = 0.14600 acc_z = 0.71387
acc_x = 0.60742 acc_y = -0.15527         acc_z = 0.95459
acc_x = 0.69678 acc_y = -0.09766         acc_z = 0.82861
acc_x = 0.47168 acc_y = -0.19580         acc_z = 1.07031
acc_x = 0.33691 acc_y = -0.23389         acc_z = 1.10156
acc_x = 0.33838 acc_y = -0.21777         acc_z = 1.10791
acc_x = 0.74414 acc_y = 0.25781 acc_z = 0.67578
acc_x = 0.75098 acc_y = 0.35498 acc_z = 0.69775
acc_x = 0.87061 acc_y = 0.54150 acc_z = 0.05957
acc_x = 0.82373 acc_y = 0.44775 acc_z = -0.24121
acc_x = 0.63574 acc_y = 0.40381 acc_z = -0.53174
acc_x = 0.52979 acc_y = 0.36914 acc_z = -0.60938
acc_x = 0.93408 acc_y = 0.41699 acc_z = 0.10742
acc_x = 0.46729 acc_y = -0.13281         acc_z = 0.98975
acc_x = 0.44873 acc_y = -0.15967         acc_z = 1.05762
acc_x = 0.37402 acc_y = -0.19336         acc_z = 1.06494
acc_x = 0.43848 acc_y = -0.17139         acc_z = 1.01221
```


New Project

Family:
- TMS570LS31x
- TMS570LS21x
- RM48x
- TMS570LS12x
- TMS570LS11x
- RM46x
- TMS570LS04x
- TMS570LS03x
- TMS570LS02x
- RM42x
- RM41x
- TMS570LS09x_07x
- RM44x
- TMS570LC43x
- RM57Lx

Device:
- TMS570LC4357ZWT
- TMS570LC4357ZWT_FREERTOS

Name: Light_ADC_RTOS

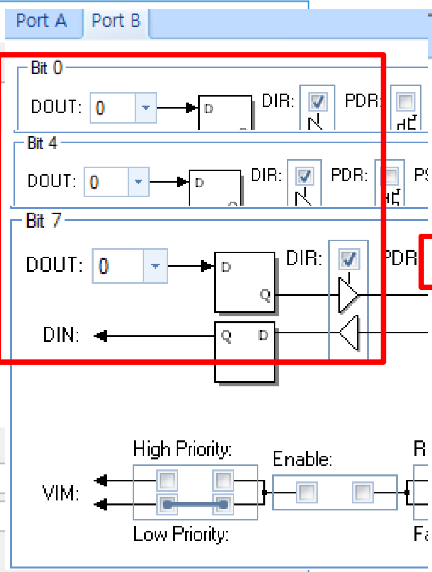
Location: C:₩Users₩minking₩Desktop₩CCS pro₩Light_ADC_RTOS

Create directory for project

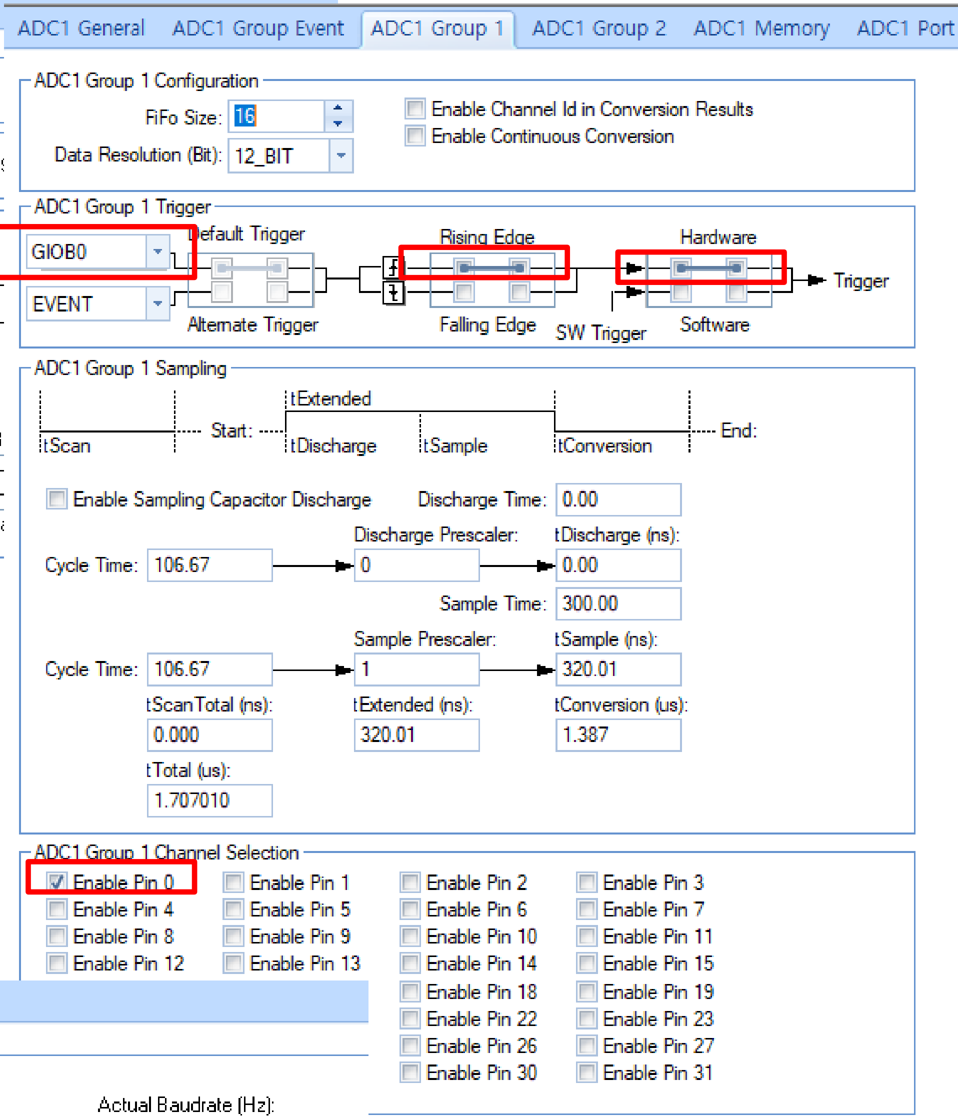
Project will be created at: C:\Users\minking\Desktop\CCS pro\Light_ADC_RTOS.

Tools: Texas Instruments Tools

OK Cancel


Port A | Port B

Bit	DOUT	DIR	PDR
Bit 0	0	✓	
Bit 4	0	✓	
Bit 7	0	✓	


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

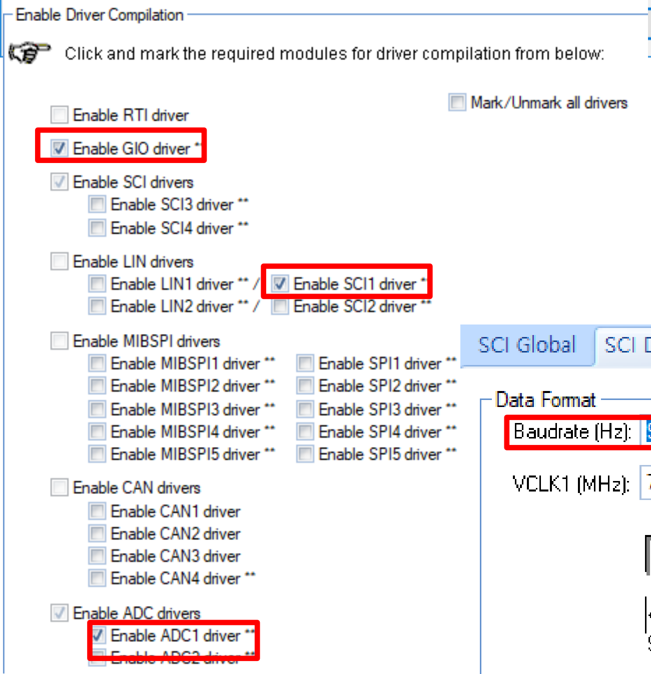
ADC1 Group 1 Trigger
- Default Trigger: GIOB0
- Alternate Trigger: EVENT
- Rising Edge
- Falling Edge
- Hardware
- Software
- SW Trigger
- Trigger

ADC1 Group 1 Sampling

- Enable Sampling Capacitor Discharge
- Discharge Time: 0.00
- Discharge Prescaler: 0
- tDischarge (ns): 0.00
- Cycle Time: 106.67
- Sample Time: 300.00
- Sample Prescaler: 1
- tSample (ns): 320.01
- Cycle Time: 106.67
- tScanTotal (ns): 0.000
- tExtended (ns): 320.01
- tConversion (us): 1.387
- tTotal (us): 1.707010

ADC1 Group 1 Channel Selection

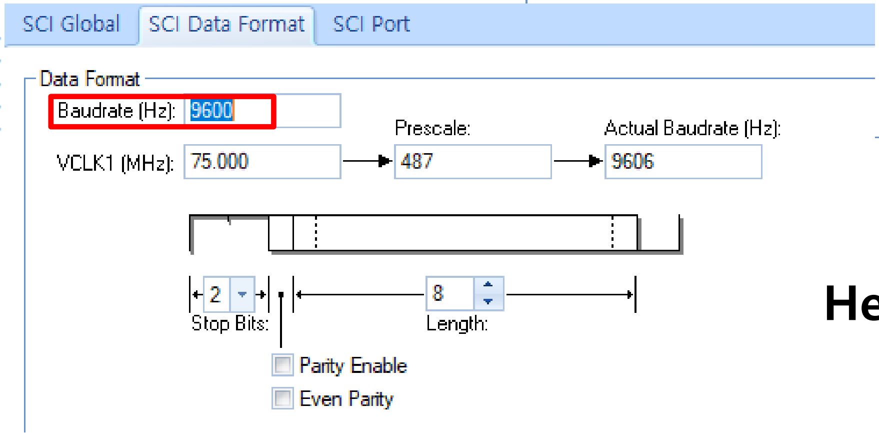
✓ Enable Pin 0	Enable Pin 1	Enable Pin 2	Enable Pin 3
Enable Pin 4	Enable Pin 5	Enable Pin 6	Enable Pin 7
Enable Pin 8	Enable Pin 9	Enable Pin 10	Enable Pin 11
Enable Pin 12	Enable Pin 13	Enable Pin 14	Enable Pin 15
		Enable Pin 18	Enable Pin 19
		Enable Pin 22	Enable Pin 23
		Enable Pin 26	Enable Pin 27
		Enable Pin 30	Enable Pin 31


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 **


SCI Global | SCI Data Format | SCI Port

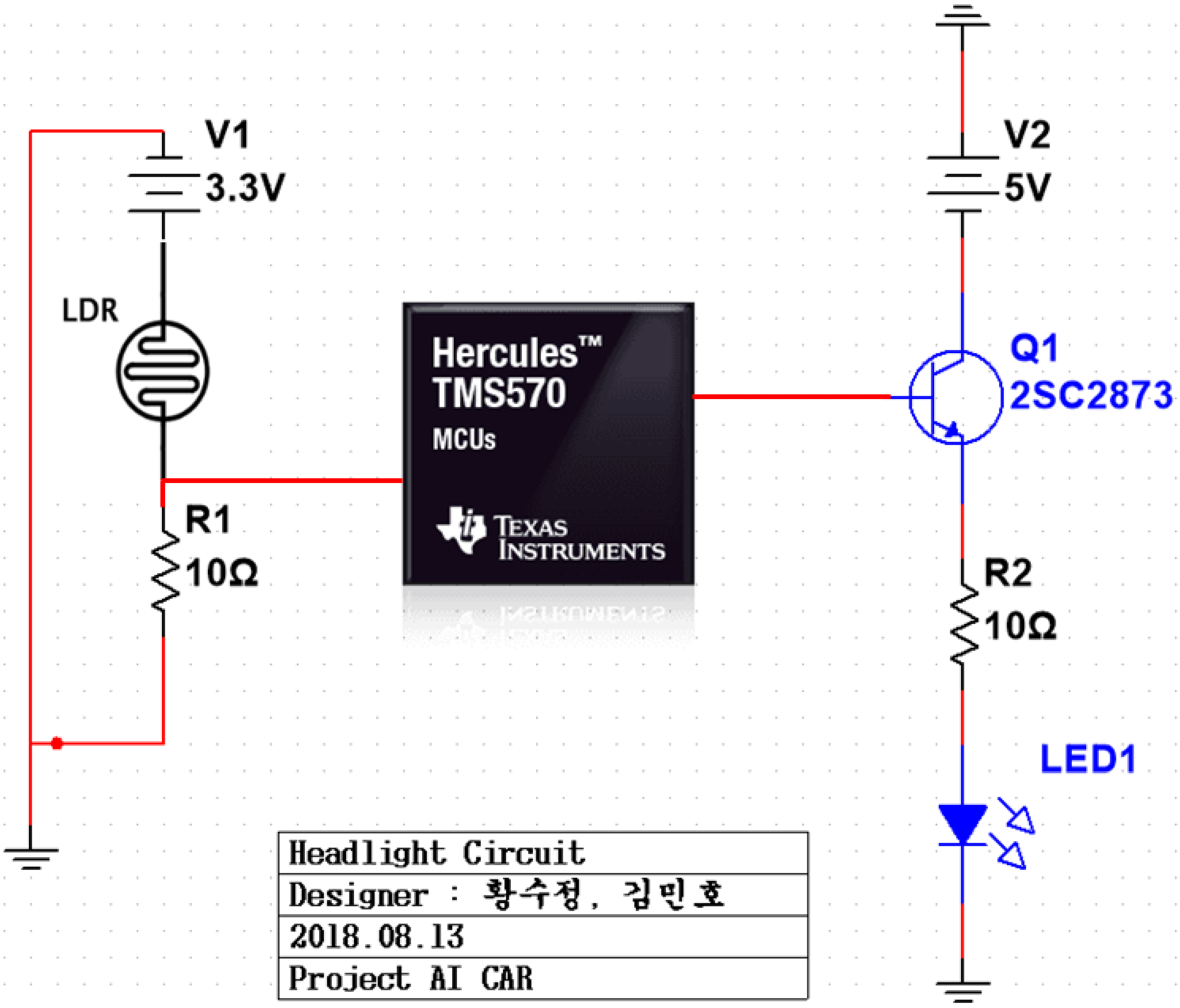
Data Format
- Baudrate (Hz): 9600
- VCLK1 (MHz): 75.000
- Prescale: 487
- Actual Baudrate (Hz): 9606
- Stop Bits: 2
- Length: 8
- Parity Enable
- Even Parity

HeadLight RTOS_Setting

HeadLight RTOS_CODE

```
#include <HL_gio.h>
#include <HL_reg_gio.h>
#include <stdio.h>
#include <FreeRTOS.h>
#include <FreeRTOSConfig.h>
#include <HL_hal_stdtypes.h>
#include <os_mpu_wrappers.h>
#include <os_projdefs.h>
#include <os_semphr.h>
#include <os_task.h>
#include <string.h>
#include <HL_reg_sci.h>
#include <HL_sci.h>
#include <HL_adc.h>
#include <HL_reg_adc.h>
adcData_t counter;
uint8 msg[32] = { 0, };
uint32 value;
xTaskHandle xTask1Handle;
QueueHandle_t mutex;
void vTask1(void* pvParameters);
void send_data(sciBASE_t *sci, uint8 *msg, uint32 length)
{
    int i;
    for (i = 0; i < length; i++)
    {
        sciSendByte(sciREG1, msg[i]);
    }
}
void led(int bri)
{
    if (bri <= 5)
    {
        gioSetBit(gioPORTB, 7, 1);
    }
    else
    {
        gioSetBit(gioPORTB, 7, 0);
    }
}
```

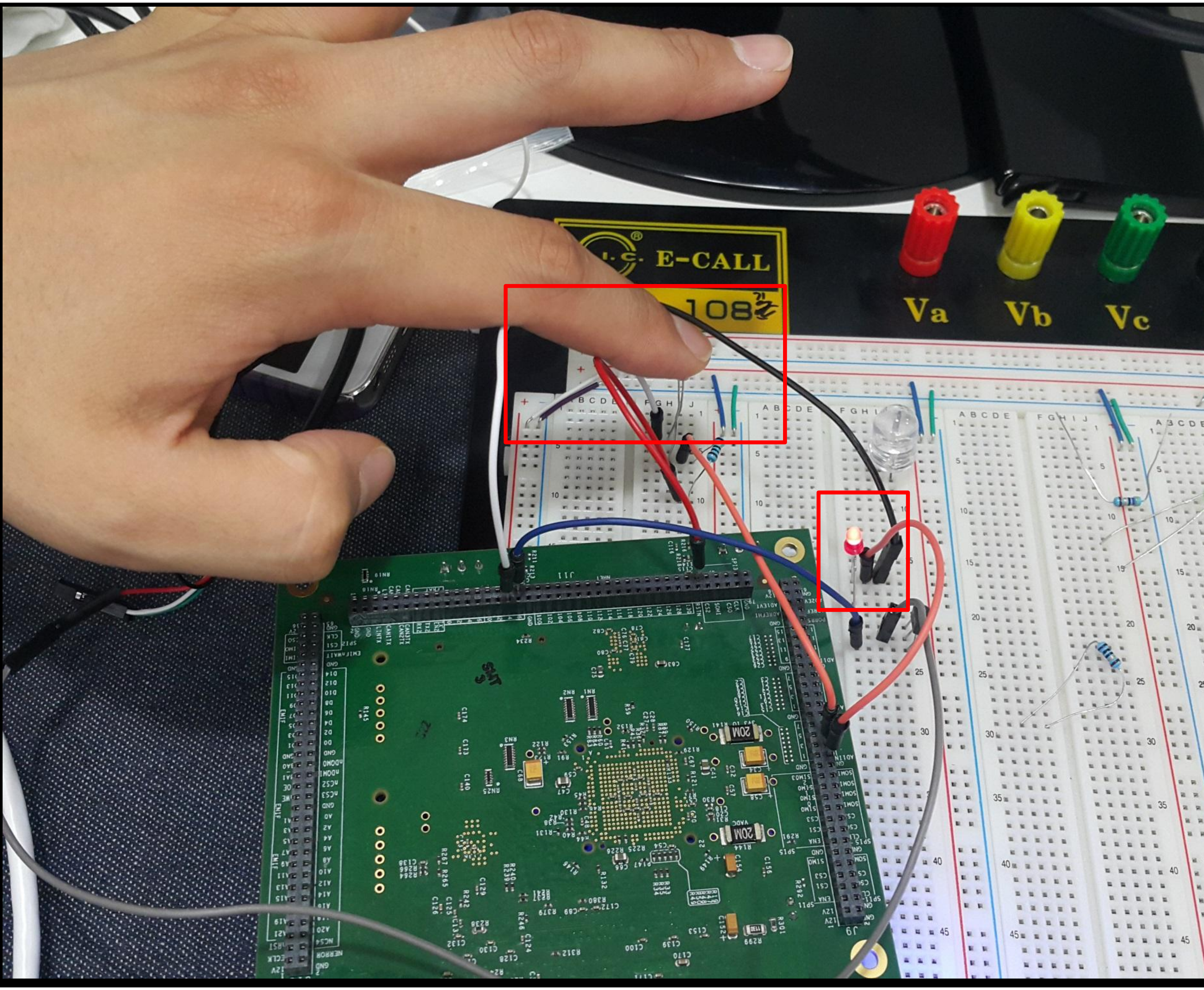
```
int main()
{

    sciInit();
    gioInit();
    adcInit();
    adcStartConversion(adcREG1, adcGROUP1);
    gioSetBit(gioPORTB, 0, 0);

    if (xTaskCreate(vTask1, "Task1", configMINIMAL_STACK_SIZE * 8, NULL, 1,
&xTask1Handle) != pdTRUE)
    {
        while (1)
            ;
    }

    vTaskStartScheduler();
    while (1)
        ;
    return 0;

}

void vTask1(void *pbParameters)
{

    while (1)
    {
        gioSetBit(gioPORTB, 0, 1);
        gioSetBit(gioPORTB, 4, 1);
        while (adcIsConversionComplete(adcREG1, adcGROUP1) == 0)
            ;
        adcGetData(adcREG1, adcGROUP1, &counter);
        sprintf(msg, "value = %d₩r₩n", counter.value);
        send_data(sciREG1, msg, strlen(msg));
        led(counter.value);
        vTaskDelay(80);
        gioSetBit(gioPORTB, 0, 0);
    }

}
```

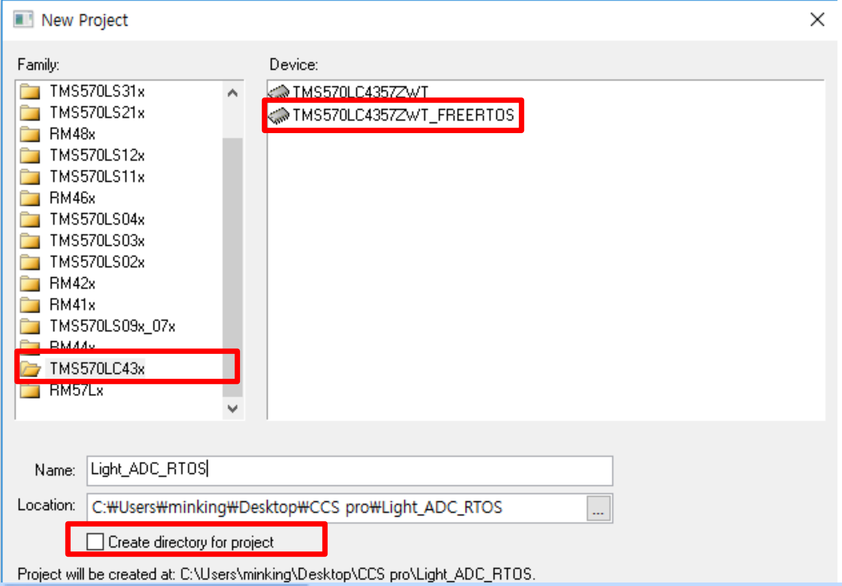


Headlight Circuit
Designer : 황수정, 김민호
2018.08.13
Project AI CAR

COM4 - PuTTY

```
value = 2
value = 1
value = 2
value = 2
value = 3
value = 2
value = 2
value = 2
value = 1
value = 2
value = 3
value = 2
value = 2
value = 2
value = 2
value = 2
value = 2
value = 3
value = 6
value = 10
value = 11
value = 12
value = 11
value = 12
value = 11
value = 11
value = 10
value = 11
value = 11
value = 12
value = 12
value = 11
value = 11
value = 11
value = 12
value = 12
value = 10
value = 11
value = 11
value = 12
value = 11
value = 11
value = 11
value = 11
value = 10
value = 11
value = 12
value = 11
value = 11
value = 11
value = 12
value = 10
value = 11
value = 11
value = 11
value = 11
value = 11
value = 11
value = 12
value = 11
value = 11
value = 11
value = 11
value = 10
value = 10
value = 10
```

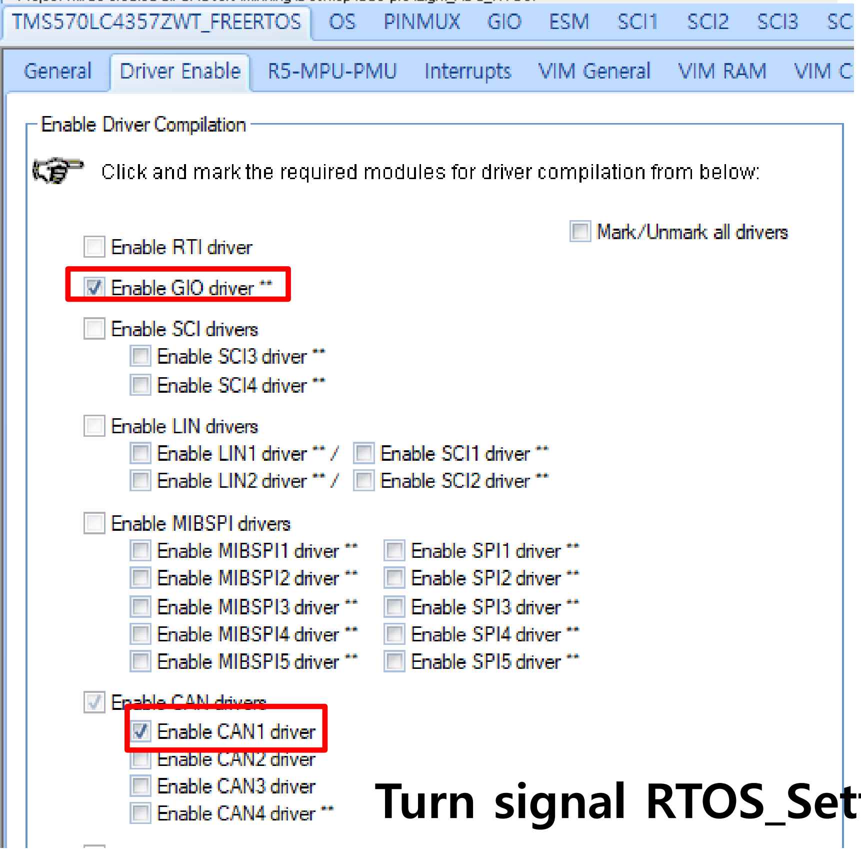



New Project

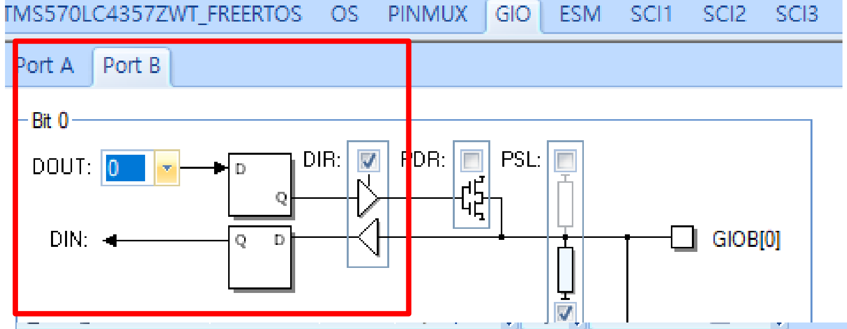
Family:
TMS570LS31x
TMS570LS21x
RM48x
TMS570LS12x
TMS570LS11x
RM46x
TMS570LS04x
TMS570LS03x
TMS570LS02x
RM42x
RM41x
TMS570LS09x_07x
RM44x
TMS570LC43x
RM57Lx

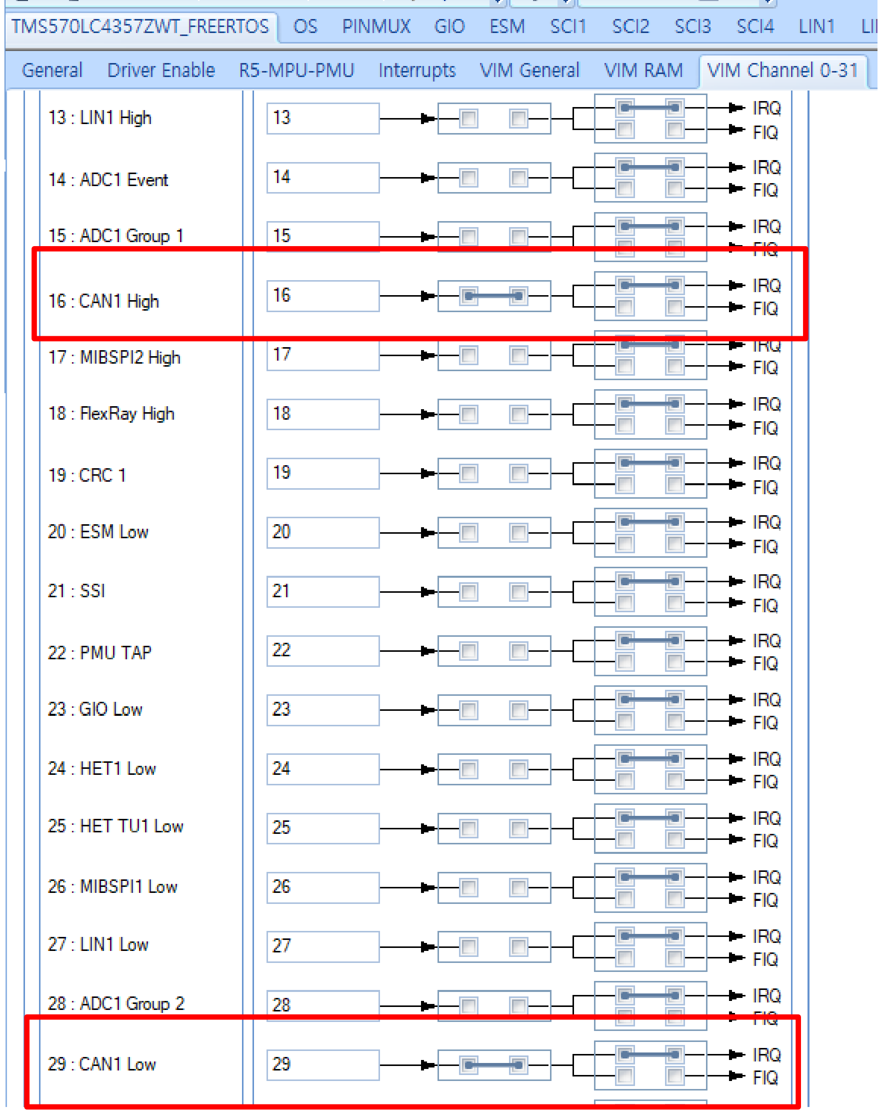
Device:
TMS570LC4357ZWT
TMS570LC4357ZWT_FREERTOS

Name: Light_ADC_RTOS
Location: C:₩Users₩minking₩Desktop₩CCS pro₩Light_ADC_RTOS
☐ Create directory for project
Project will be created at: C:\Users\minking\Desktop\CCS pro\Light_ADC_RTOS.


TMS570LC4357ZWT_FREERTOS | OS | PINMUX | GIO | ESM | SCI1 | SCI2 | SCI3 | SC
General | Driver Enable | R5-MPU-PMU | Interrupts | VIM General | VIM RAM | VIM C

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 **

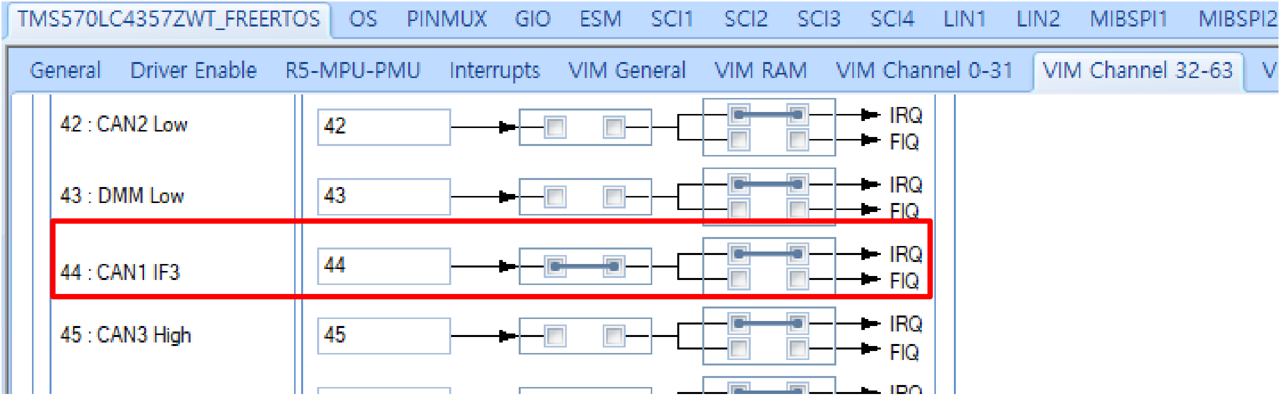

TMS570LC4357ZWT_FREERTOS | OS | PINMUX | GIO | ESM | SCI1 | SCI2 | SCI3
Port A | Port B
Bit 0
DOUT: 0
DIN:
DIR: ☑
PDR: ☐
PSL: ☐
GIOB[0]


TMS570LC4357ZWT_FREERTOS | OS | PINMUX | GIO | ESM | SCI1 | SCI2 | SCI3 | SCI4 | LIN1 | LI
General | Driver Enable | R5-MPU-PMU | Interrupts | VIM General | VIM RAM | VIM Channel 0-31

Channel	Number	Enabled
13 : LIN1 High	13	☐
14 : ADC1 Event	14	☐
15 : ADC1 Group 1	15	☐
16 : CAN1 High	16	☑
17 : MIBSPI2 High	17	☐
18 : FlexRay High	18	☐
19 : CRC 1	19	☐
20 : ESM Low	20	☐
21 : SSI	21	☐
22 : PMU TAP	22	☐
23 : GIO Low	23	☐
24 : HET1 Low	24	☐
25 : HET TU1 Low	25	☐
26 : MIBSPI1 Low	26	☐
27 : LIN1 Low	27	☐
28 : ADC1 Group 2	28	☐
29 : CAN1 Low	29	☑

Turn signal RTOS_Setting

Turn signal RTOS_Setting



TMS570LC4357ZWT_FREERTOS | OS | PINMUX | GIO | ESM | SCI1 | SCI2 | SCI3 | SCI4 | LIN1 | LIN2 | MIBSPI1 | MIBSPI2

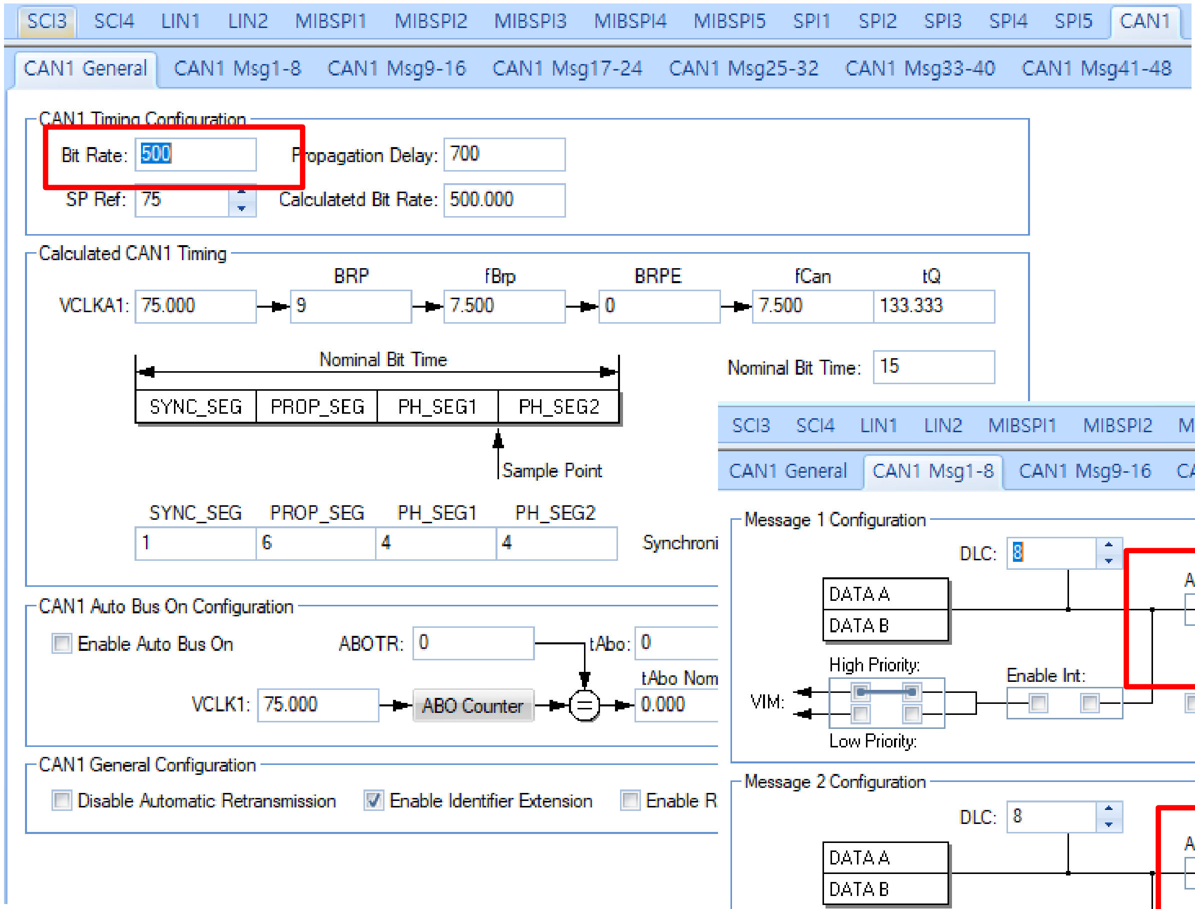
General | Driver Enable | R5-MPU-PMU | Interrupts | VIM General | VIM RAM | VIM Channel 0-31 | VIM Channel 32-63 | V

42 : CAN2 Low — 42 — IRQ / FIQ

43 : DMM Low — 43 — IRQ / FIQ

44 : CAN1 IF3 — 44 — IRQ / FIQ

45 : CAN3 High — 45 — IRQ / FIQ



SCI3 | SCI4 | LIN1 | LIN2 | MIBSPI1 | MIBSPI2 | MIBSPI3 | MIBSPI4 | MIBSPI5 | SPI1 | SPI2 | SPI3 | SPI4 | SPI5 | CAN1

CAN1 General | CAN1 Msg1-8 | CAN1 Msg9-16 | CAN1 Msg17-24 | CAN1 Msg25-32 | CAN1 Msg33-40 | CAN1 Msg41-48

CAN1 Timing Configuration

Bit Rate: 500 Propagation Delay: 700

SP Ref: 75 Calculatetd Bit Rate: 500.000

Calculated CAN1 Timing

	BRP	fBrp	BRPE	fCan	tQ
VCLKA1: 75.000	9	7.500	0	7.500	133.333

Nominal Bit Time

SYNC_SEG	PROP_SEG	PH_SEG1	PH_SEG2

Sample Point

Nominal Bit Time: 15

SYNC_SEG	PROP_SEG	PH_SEG1	PH_SEG2
1	6	4	4

Synchroni

CAN1 Auto Bus On Configuration

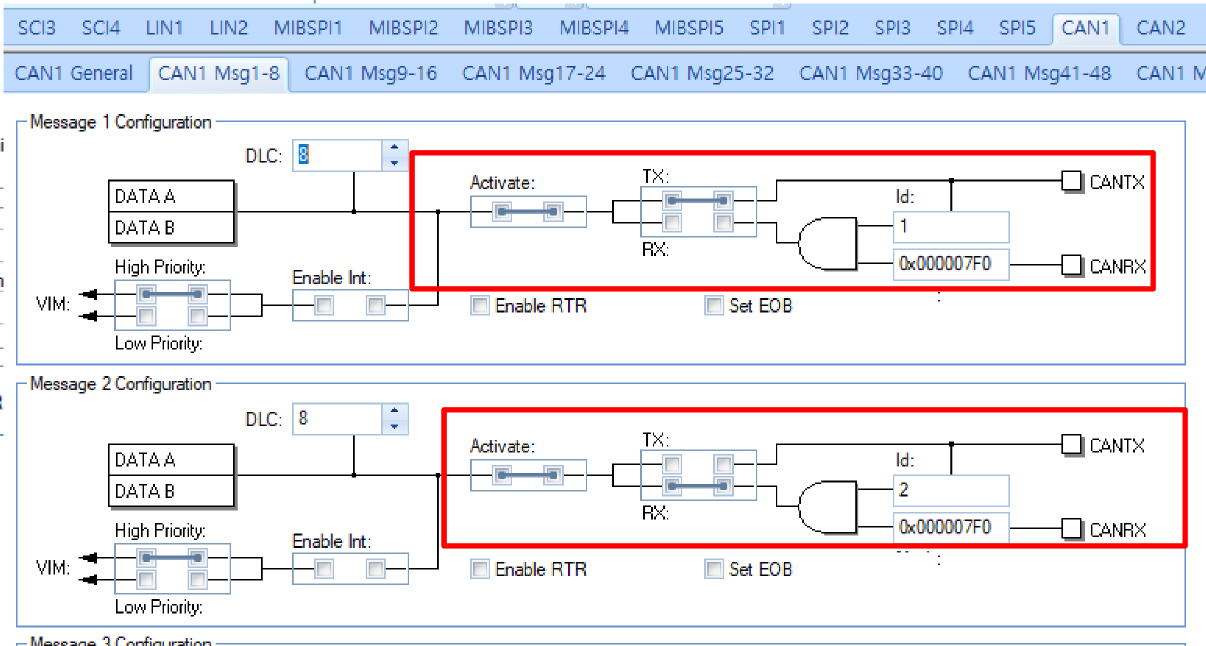
☐ Enable Auto Bus On ABOTR: 0 tAbo: 0

tAbo Nom

VCLK1: 75.000 ABO Counter 0.000

CAN1 General Configuration

☐ Disable Automatic Retransmission ☑ Enable Identifier Extension ☐ Enable R



SCI3 | SCI4 | LIN1 | LIN2 | MIBSPI1 | MIBSPI2 | MIBSPI3 | MIBSPI4 | MIBSPI5 | SPI1 | SPI2 | SPI3 | SPI4 | SPI5 | CAN1 | CAN2

CAN1 General | CAN1 Msg1-8 | CAN1 Msg9-16 | CAN1 Msg17-24 | CAN1 Msg25-32 | CAN1 Msg33-40 | CAN1 Msg41-48 | CAN1 M

Message 1 Configuration

DLC: 8

DATA A / DATA B

Activate: TX: / RX:

Id: 1

0x000007F0

CANTX / CANRX

High Priority: / Low Priority: VIM:

Enable Int:

☐ Enable RTR ☐ Set EOB

Message 2 Configuration

DLC: 8

DATA A / DATA B

Activate: TX: / RX:

Id: 2

0x000007F0

CANTX / CANRX

High Priority: / Low Priority: VIM:

Enable Int:

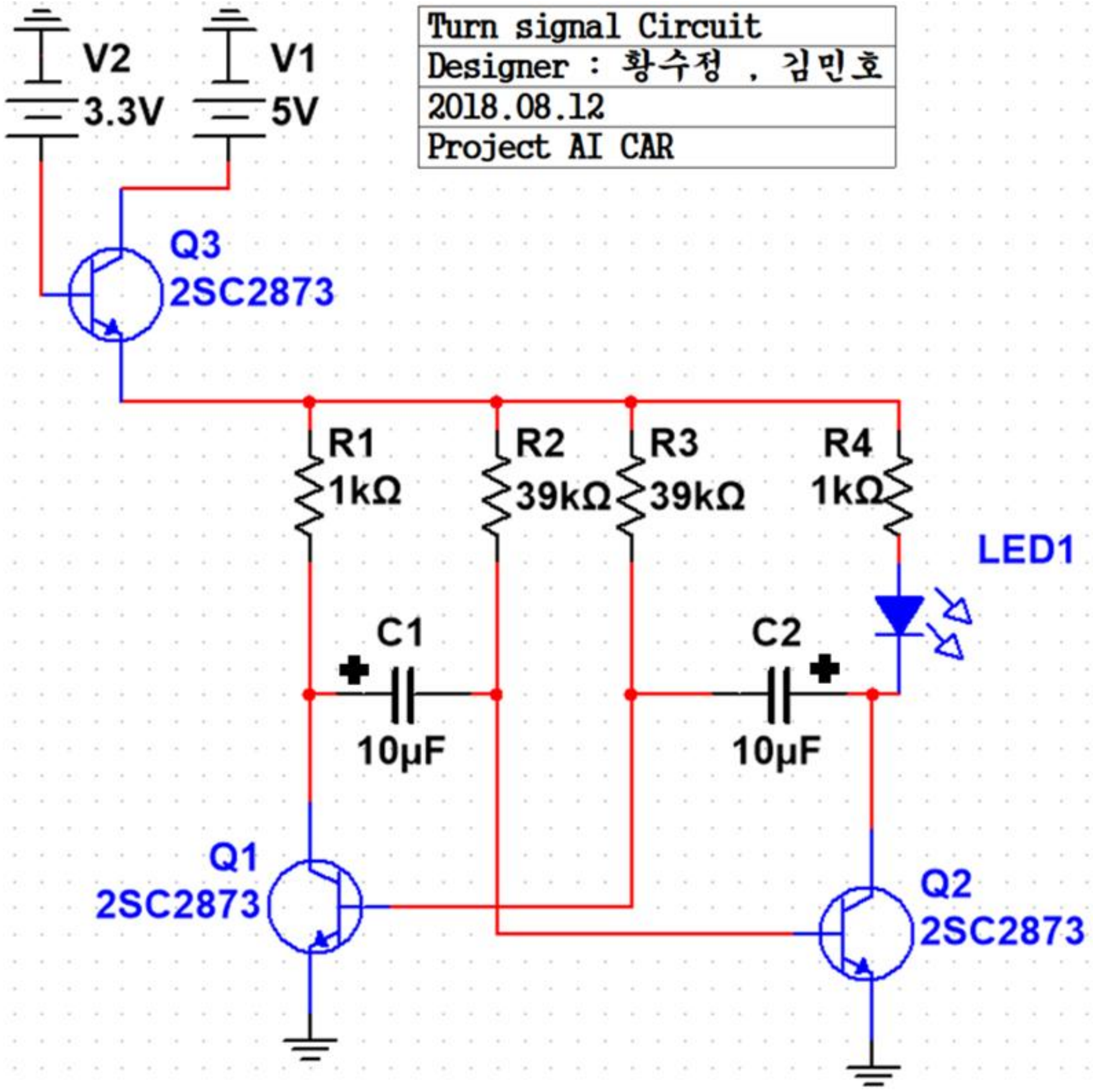
☐ Enable RTR ☐ Set EOB

Message 3 Configuration

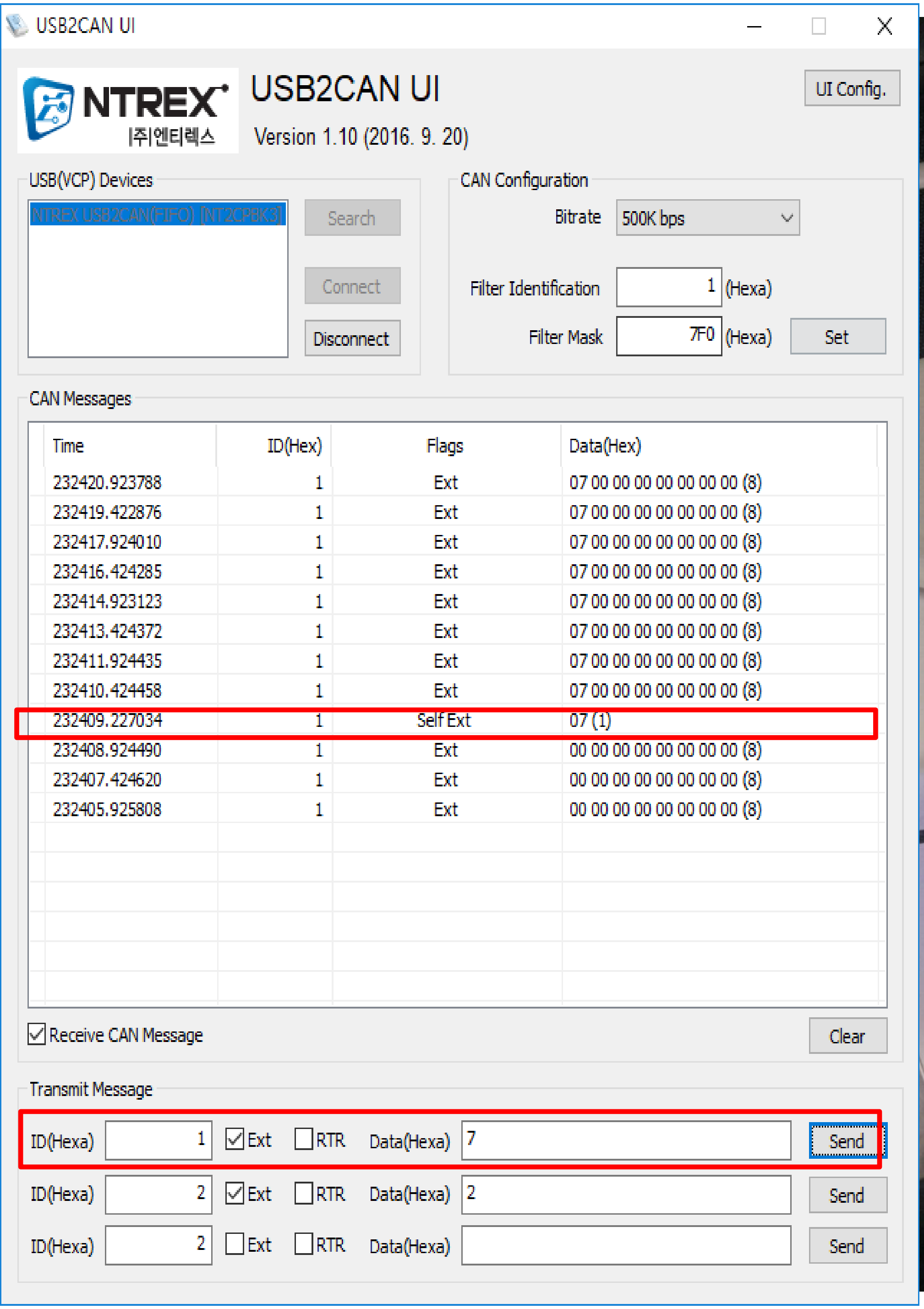
```c
#include <HL_can.h>
#include <HL_gio.h>
#include <HL_reg_can.h>
#include <HL_reg_gio.h>
#include <stdio.h>
#include <FreeRTOS.h>
#include <FreeRTOSConfig.h>
#include <HL_hal_stdtypes.h>
#include <os_mpu_wrappers.h>
#include <os_projdefs.h>
#include <os_semphr.h>
#include <os_task.h>
#include <string.h>

char num;
xTaskHandle xTask1Handle;
QueueHandle_t mutex;
void vTask1(void* pvParameters);

void delay(int num)
{
    int a;
    for (a = 0; a < num; a++)
        ;
}
int main()
{

    gioInit();
    canInit();
    delay(10000);
    char num = 0;
    //vSemaphoreCreateBinary(mutex)
    if (xTaskCreate(vTask1, "Task1", configMINIMAL_STACK_SIZE*8, NULL,
1,&xTask1Handle) != pdTRUE)
    {
        while (1)
            ;
    }

    vTaskStartScheduler();
    while (1)
        ;
    return 0;

}
```

```c
void vTask1(void *pbParameters)
{
        while (1)
        {

            canTransmit(canREG1, canMESSAGE_BOX1, &num);
            vTaskDelay(500);

            canIsRxMessageArrived(canREG1, canMESSAGE_BOX2);
            vTaskDelay(500);
            canGetData(canREG1, canMESSAGE_BOX2, &num);

            switch(num)
            {
                case 7:
                gioSetBit(gioPORTB, 0, 1);
                vTaskDelay(500);
                break;


                default:
                gioSetBit(gioPORTB, 0, 0);
                vTaskDelay(500);
                break;
            }
        }
}
```

Turn signal RTOS_CODE



Turn signal Circuit
Designer : 황수정 , 김민호
2018.08.12
Project AI CAR



USB2CAN UI

NTREX (주)엔티렉스

USB2CAN UI

Version 1.10 (2016. 9. 20)

UI Config.

USB(VCP) Devices

NTREX USB2CAN(FIFO) [NT2CP8K3]

Search

Connect

Disconnect

CAN Configuration

Bitrate: 500K bps

Filter Identification: 1 (Hexa)

Filter Mask: 7F0 (Hexa)

Set

CAN Messages

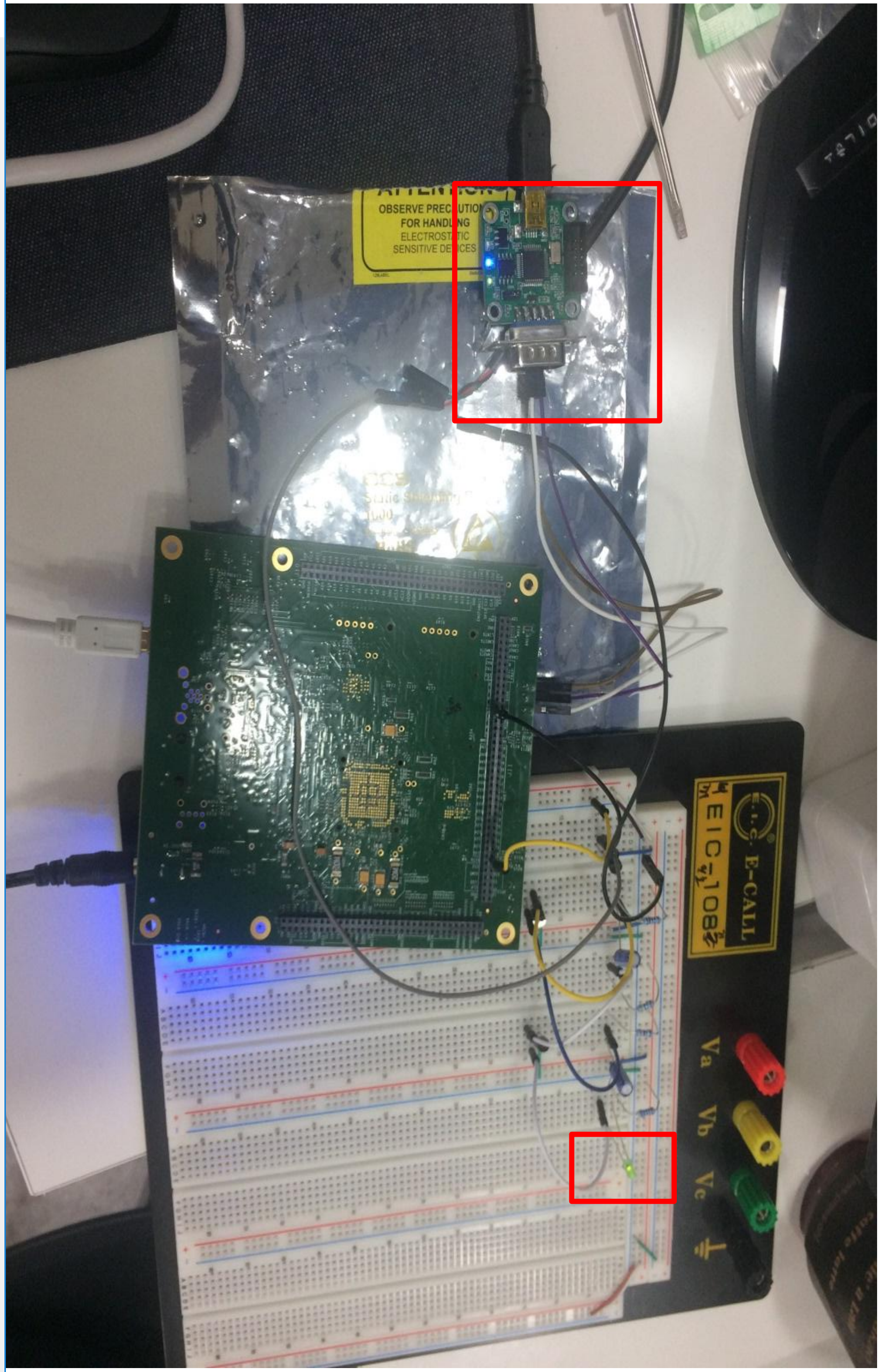
Time	ID(Hex)	Flags	Data(Hex)
232420.923788	1	Ext	07 00 00 00 00 00 00 00 (8)
232419.422876	1	Ext	07 00 00 00 00 00 00 00 (8)
232417.924010	1	Ext	07 00 00 00 00 00 00 00 (8)
232416.424285	1	Ext	07 00 00 00 00 00 00 00 (8)
232414.923123	1	Ext	07 00 00 00 00 00 00 00 (8)
232413.424372	1	Ext	07 00 00 00 00 00 00 00 (8)
232411.924435	1	Ext	07 00 00 00 00 00 00 00 (8)
232410.424458	1	Ext	07 00 00 00 00 00 00 00 (8)
232409.227034	1	Self Ext	07 (1)
232408.924490	1	Ext	00 00 00 00 00 00 00 00 (8)
232407.424620	1	Ext	00 00 00 00 00 00 00 00 (8)
232405.925808	1	Ext	00 00 00 00 00 00 00 00 (8)

☑ Receive CAN Message

Clear

Transmit Message

ID(Hexa)	Ext	RTR	Data(Hexa)	
1	☑	☐	7	Send
2	☑	☐	2	Send
2	☐	☐		Send



감사합니다