

TI DSP, MCU 및 Xilinx Zynq FPGA 프로그래밍 전문가 과정

강사 : Innova Lee(이상훈)

gcccompil3r@gmail.com

학생 : 황수정

sue100012@naver.com

54일차 (2018. 05. 15)

목차

- Cortex-R5F
 - > OC Circuit
 - > ADC_UART
- FreeRTOS_Blinky

OC Circuit

```
#include "HL_sys_common.h"
#include "HL_gio.h"
#include "HL_rti.h"

int main(void)
{
    gioInit();
    rtiInit();
    gioSetDirection(gioPORTA, 0xffffffff);
    rtiEnableNotification(rtiREG1, rtiNOTIFICATION_COMPARE0);
    gioSetPort(gioPORTA, 0xffffffff);
    _enable_IRQ_interrupt_();
    rtiStartCounter(rtiREG1, rtiCOUNTER_BLOCK0);
    for(;;)
        ;
    return 0;
}

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

ADC_UART

```
#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] = {'W', 'r', ' ', 't', ' ', 'C', 'H', ' ', '-', 'D', ' ', '-', '0', 'x'};
#define TSIZE2 9
uint8 TEXT2[TSIZE2] = {'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;
    gpiolInit();
    gpioSetDirection(gpioPORTB, 1);

    scilInit();

    adclInit();
    adcStartConversion(adcREG1, adcGROUP1);

    for(;;)
    {
        gpioSetBit(gpioPORTB, 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;

        gpioSetBit(gpioPORTB, 0, 0);

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

        while( (sciREG1->FLR & 0x4) == 4);
        sciSendByte(sciREG1,txt1);
        while( (sciREG1->FLR & 0x4) == 4);
    }
}

```

```

        sciSendByte(sciREG1, txt);
    }
}

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

```

FreeRTOS_Blinky

```

#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]={'W','r','\n','|','W','t','C','H',' ',' ','I','D',' ','-',' ','0','x'};
#define TSIZE2 9
uint8 TEXT2[TSIZE2]={'W','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();

```

```

adclnit();
adcStartConversion(adcREG1, adcGROUP1);

while(1){
    gpioSetBit(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;

    gpioSetBit(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 > 0xEA0){
        gpioSetBit(gioPORTB, 4, 1);
    }
    else{
        gpioSetBit(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++);
}

```

