

# Xilinx Zynq FPGA, TI DSP, MCU 기반의 프로그래밍 및 회로 설계 전문가 과정

## MCU I2C-LCD Control

강사 : Innova Lee(이 상훈)

학생 : 김 시윤

## MCU Source code result

아두이노 함수들을 참고하여 아두이노 함수를 MCU 로 그대로 가져왔다.

custom function

**void** lcd\_init(**void**) = LCD 초기 설정 (Config)

**void** lcd\_Backlight() = Backlight 를 켜주는 함수.

**void** lcd\_noBacklight() = Backlight 를 꺼주는 함수.

**void** lcd\_set\_cursor(**int** row, **char** col) = 문자를 출력할 행과 열을 설정하는 함수.

-16 X 2 LCD 기준.

-Row = 0(첫번째 줄) or 1(두번째 줄)

-col = 0~15 → 16글자. 0 = 1번째 부터 display 15 = 16번째부터 display

**void** lcd\_send\_cmd(**char** cmd) = LCD 에 보낼 설정값을 전송해 주는 함수.

**void** lcd\_send\_data(**char** data) = 출력할 데이터를 보내주는 함수.

**void** lcd\_sned\_string(**char** \*str) 출력 할 문자열을 보내주는 함수.

**void** lcd\_clear() = lcd 를 clear 해주는 함수

```
#include <string.h>
#include <stdio.h>

#include "HL_sys_common.h"
#include "HL_sys_core.h"
#include "HL_i2c.h"

#define LCD_ADDRESS 0x3F

void lcd_Backlight();
void lcd_noBacklight();
void lcd_sned_string(char *str);
void lcd_init(void);
void lcd_send_cmd(char cmd);
void lcd_send_data(char data);
void lcd_set_cursor(int row, char col);
void lcd_clear();

unsigned char LCD_BACKLIGHT =0X0;

void lcd_send_string (char *str)
{
    while (*str) lcd_send_data (*str++);
}

int main(void)
```

```

{
    volatile int i;

    for(i = 0; i< 10000000; i++);

    i2cInit();

    for(i = 0; i< 10000000; i++);
        lcd_init();

        while(1)
        {

            lcd_set_cursor(0, 3);
            lcd_send_string("Handsome");
            lcd_set_cursor(1, 3);
            lcd_send_string("SiYunKim");
            for(i=0; i<80000000; i++);
            lcd_clear();

        }
}

void lcd_send_cmd(char cmd)
{
    volatile unsigned int cnt = 4;
    unsigned char data_u, data_l;
    uint8_t data_t[4];
    data_u = (cmd&0xf0);
    data_l = ((cmd<<4)&0xf0);
    data_t[0] = data_u|(0x04+LCD_BACKLIGHT); //en=1, rs=0
    data_t[1] = data_u|(0x00+LCD_BACKLIGHT); //en=0, rs=0
    data_t[2] = data_l|(0x04+LCD_BACKLIGHT); //en=1, rs=0
    data_t[3] = data_l|(0x00+LCD_BACKLIGHT); //en=0, rs=0

    i2cSetSlaveAdd(i2cREG2, LCD_ADDRESS);
    i2cSetDirection(i2cREG2, I2C_TRANSMITTER);
    i2cSetCount(i2cREG2, cnt+1);
    i2cSetMode(i2cREG2, I2C_MASTER);
    i2cSetStop(i2cREG2);
    i2cSetStart(i2cREG2);
    i2cSendByte(i2cREG2, LCD_ADDRESS);
    i2cSend(i2cREG2, cnt, data_t);

    while(i2cIsBusBusy(i2cREG2)==true);
    while(i2cIsStopDetected(i2cREG2)==0);
    i2cClearSCD(i2cREG2);

    for(cnt = 0; cnt < 1000000; cnt++);
}

void lcd_send_data(char data)
{
    volatile unsigned int cnt = 4;
    char data_u, data_l;
    uint8_t data_t[4];

    data_u = (data&0xf0);
    data_l = ((data <<4)&0xf0);
    data_t[0] = data_u|(0x05+LCD_BACKLIGHT); //en=1, rs=0

```

```

data_t[1] = data_u|(0x01+LCD_BACKLIGHT); //en=0, rs=0
data_t[2] = data_l|(0x05+LCD_BACKLIGHT); //en=1, rs=0
data_t[3] = data_l|(0x01+LCD_BACKLIGHT); //en=0, rs=0

i2cSetSlaveAdd(i2cREG2, LCD_ADDRESS);
i2cSetDirection(i2cREG2, I2C_TRANSMITTER);
i2cSetCount(i2cREG2, cnt+1);
i2cSetMode(i2cREG2, I2C_MASTER);
i2cSetStop(i2cREG2);
i2cSetStart(i2cREG2);
i2cSendByte(i2cREG2, LCD_ADDRESS);
i2cSend(i2cREG2, cnt, data_t);

while(i2cIsBusBusy(i2cREG2)==true);
while(i2cIsStopDetected(i2cREG2)==0);
i2cClearSCD(i2cREG2);

for(cnt = 0; cnt < 1000000; cnt++);
}

void lcd_init(void)
{
    lcd_send_cmd(0x02);
    lcd_send_cmd(0x28);
    lcd_send_cmd(0x0c);
    lcd_send_cmd(0x80);
}

void lcd_set_cursor(int row, char col)
{
    if( row == 0)
        lcd_send_cmd(0x80+col);
    else if(row == 1)
        lcd_send_cmd(0xc0+col);
}

void lcd_Backlight()
{
    LCD_BACKLIGHT = 0x08;
}

void lcd_noBacklight()
{
    LCD_BACKLIGHT = 0x00;
}

void lcd_clear()
{
    lcd_send_cmd(0x01);
}

```