

TI DSP, MCU 및 Xilinx Zynq FPGA 프로그래밍 전문가 과정

2018-08-16 (프로젝트 6회차)

강사 - Innova Lee(이상훈)

gcccompil3r@gmail.com

학생 - 정유경

ucong@naver.com

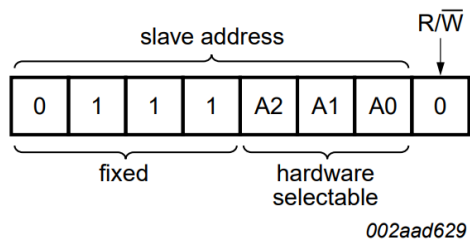
I2C device(PCF8574A)를 이용해서 LCD인터페이스

1. Slave Address
2. 명령어 입력방법
3. 소스코드

1. Slave Address

How do we modify the address of PCF8574A?

Slave Address를 Datasheet에서 확인해보면 0x3F임을 알 수 있다.



b. PCF8574A

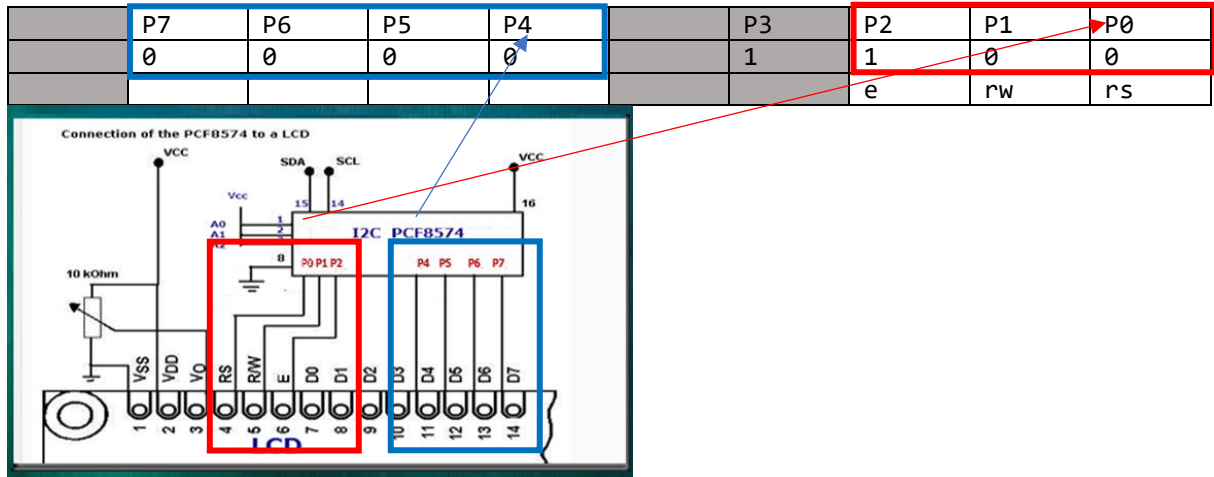
Why we modify the address?

Using A0, A1, A2 pins, we can connect up to 8 **PCF8574A** to the same line

2. 명령어 입력방법

PCF8574A 출력과 명령어입력

0x0C 0000 P[7:4] 1100 P[3:0]



11.9 Instruction Table

Instruction	Instruction code										Description	Execution time (fosc= 270 KHZ)
	RS	R/W	DB ⁷	DB ⁶	DB ⁵	DB ⁴	DB ³	DB ²	DB ¹	DB ⁰		
Clear Display	0	0	0	0	0	0	0	0	0	1	Write "20H" to DDRA and set DDRAM address to "00H" from AC.	1.53ms
Return Home	0	0	0	0	0	0	0	0	1	-	Set DDRAM address to "00H" From AC and return cursor to its original position if shifted. The contents of DDRAM are not changed.	1.53ms
Entry mode Set	0	0	0	0	0	0	0	1	I/D	SH	Assign cursor moving direction And blinking of entire display	39us
Display ON/OFF control	0	0	0	0	0	0	1	D	C	B	Set display (D), cursor (C), and Blinking of cursor (B) on/off Control bit.	
Cursor or Display shift	0	0	0	0	0	1	S/C	R/L	-	-	Set cursor moving and display Shift control bit, and the Direction, without changing of DDRAM data.	39us
Function set	0	0	0	0	1	DL	N	F	-	-	Set interface data length (DL: 8-Bit/4-bit), numbers of display Line (N: =2-line/1-line) and, Display font type (F: 5x11/5x8).	39us
Set CGRAM Address	0	0	0	1	AC ⁵	AC ⁴	AC ³	AC ²	AC ¹	AC ⁰	Set CGRAM address in address Counter.	39us
Set DDRAM Address	0	0	1	AC ⁶	AC ⁵	AC ⁴	AC ³	AC ²	AC ¹	AC ⁰	Set DDRAM address in address Counter.	39us
Read busy Flag and Address	0	1	BF	AC ⁶	AC ⁵	AC ⁴	AC ³	AC ²	AC ¹	AC ⁰	Whether during internal Operation or not can be known By reading BF. The contents of Address counter can also be read.	0us
Write data to Address	1	0	D ⁷	D ⁶	D ⁵	D ⁴	D ³	D ²	D ¹	D ⁰	Write data into internal RAM (DDRAM/CGRAM).	43us
Read data From RAM	1	1	D ⁷	D ⁶	D ⁵	D ⁴	D ³	D ²	D ¹	D ⁰	Read data from internal RAM (DDRAM/CGRAM).	43us

3. 소스코드

HalCoGen 설정

- Enable I2C2 Driver
- PINMUX: I2C2 체크

I2C Global I2C Clocks I2C Port

Global Config

☒ Enable Master Mode Tx / Rx: TRANSMITTER

Add mode: 7BIT_AMODE 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.

I2C Global I2C Clocks I2C Port

Data Format

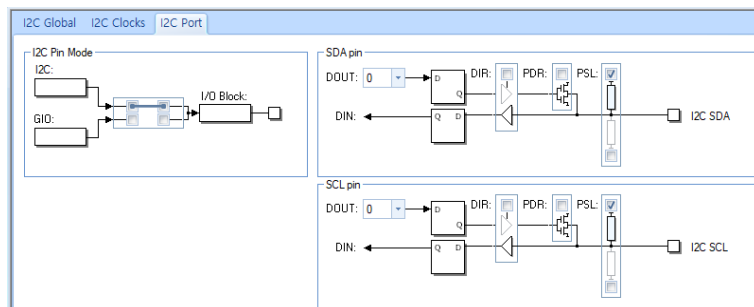
Baudrate: 400

Prescale: 8 Module Clock Frequency: 8

VCLK1 (MHz): 75.000

ICCH: 5

ICCL: 5



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	

CCS 작성

```
#include <string.h>
#include <stdio.h>
#include "HL_sys_common.h"
#include "HL_sys_core.h"
#include "HL_gio.h"
#include "HL_i2c.h"

#define LCD_ADDRESS 0x3F

void lcd_send_string(char *str);
void lcd_init(void);
void lcd_send_cmd(char cmd);
void lcd_send_data(char data);
//void lcd_shift_string(char *str);
void lcd_set_cursor(char row, char column);
//void lcd_Backlight_on_off(char control);

int main(void)
{
    volatile int i,j;
    for (i = 0; i < 10000000; i++)
        ;
    i2cInit();
    for (i = 0; i < 10000000; i++)
        ;

    lcd_init();
    lcd_send_cmd(0x80); // Set DataRAM Address
    lcd_send_string("Let's Start");

    lcd_send_cmd(0xc0);
    lcd_send_string("HELLO WORLD!!");
    lcd_send_cmd(0x01); // clear display
    lcd_send_cmd(0x80);

    while(1)
    {
        for(j = 16; j >=0 ; j--)
        {
            lcd_set_cursor(0, j);
            lcd_send_string("^-^");
            lcd_send_cmd(0xc0); // 2nd line
            lcd_set_cursor(1, j);
            lcd_send_string("Goodbye!");
            for(i=0; i<8000000; i++);
            lcd_send_cmd(0x01);
        }
        j=0;
    }
}

void lcd_send_string(char *str)
{
    while (*str) // 아스키 NULL은 정수로 0
        lcd_send_data(*str++);
}
```

```

    lcd_send_cmd(0x14);
}

void lcd_send_cmd(char cmd) // 0x80
{
    volatile unsigned int cnt = 4;
    unsigned char data_u, data_l;
    uint8_t data_t[4];
    data_u = (cmd & 0xf0); //1000 0000
    data_l = ((cmd << 4) & 0xf0); // 0000 0000

    /* 명령어:0 */
    data_t[0] = data_u | 0x0C; //en=1, rs=0
    // [2]번 비트: en [1]번비트: rs, // backlight P[3]!!!
    data_t[1] = data_u | 0x08; //en=0, rs=0 // 1000 1000:
    data_t[2] = data_l | 0x0C; //en=1, rs=0 // 0000 1100:
    data_t[3] = data_l | 0x08; //en=0, rs=0 // 0000 1000:

    i2cSetSlaveAdd(i2cREG2, LCD_ADDRESS); // LCD_ADDRESS = 0x3F
    i2cSetDirection(i2cREG2, I2C_TRANSMITTER); // I2C_Transmitter가 512
    i2cSetCount(i2cREG2, cnt + 1); // cnt가 4니까 SetCount를 5로 한다.
    // stop condition이 생성되기 위한 카운트 조건
    i2cSetMode(i2cREG2, I2C_MASTER); // Master는 1024
    i2cSetStop(i2cREG2);
    i2cSetStart(i2cREG2);

    i2cSendByte(i2cREG2, LCD_ADDRESS);
    i2cSend(i2cREG2, cnt, data_t);

    while (i2cIsBusBusy(i2cREG2) == true)
        ; // i2c 버스가 전송중이지 않으면 넘어간다.
    while (i2cIsStopDetected(i2cREG2) == 0)
        ; // stop bit가 인지되면 넘어간다.
    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);

    /* 데이터:1 */
    data_t[0] = data_u | 0x0D; //en=1, rs=1
    data_t[1] = data_u | 0x09; //en=0, rs=1 // enable 토글(1->0)시켜서 데이터 전송.
    data_t[2] = data_l | 0x0D; //en=1, rs=1

```

```

data_t[3] = data_1 | 0x09; //en=0, rs=1
i2cSetSlaveAdd(i2cREG2, LCD_ADDRESS);
i2cSetDirection(i2cREG2, I2C_TRANSMITTER);
i2cSetCount(i2cREG2, cnt + 1); // LCD_ADDRESS + data_t[0~3] 이므로 총 4개
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);
}

void lcd_init(void)
{
    lcd_send_cmd(0x02); // Return Home
    lcd_send_cmd(0x28); //4bit mode/ 2라인모드/ 5X8 Format
    lcd_send_cmd(0x0C); // 디스플레이 On /커서 off/커서위치의 문자를 깜빡이지 않는다
    lcd_send_cmd(0x80); // DDRAM의 주소를 설정한다.
    lcd_send_cmd(0x00);
}

void lcd_set_cursor(char row, char column)
{
    if (row == 0)
        lcd_send_cmd(0x80 + column);
    else if (row == 1)
        lcd_send_cmd(0xc0 + column);
    else
        printf("Row number is 0 or 1\n");
}

```