

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

New CCS Project

CCS Project

Create a new CCS Project.



Target:


<select or type filter text>

TMS570LC43xx

Connection:

Texas Instruments XDS100v2 USB Debug Probe

Verify...

Cortex R [ARM]

Project name:

LCD_MODULE

☒ Use default location

Location:

/home/siyun/workspace_v8/LCD_MODULE

Browse...

Compiler version:

Ti v18.1.1.LTS

More...

Tool-chain

Project templates and examples

type filter text

Empty Projects

Empty Project

Empty Project (with main.c)

Empty Assembly-only Project


Empty RTSC Project

Basic Examples

Hello World

Creates an empty project initialized for the selected device.

Open **Resource Explorer** to browse a wide selection of example projects...



< Back

Next >

Cancel

Finish

New Project

Family:

- TMS570LS04x
- TMS570LS03x
- TMS570LS02x
- RM42x
- RM41x
- TMS570LS09x_07x
- RM44x
- TMS570LC43x
- RM57Lx

Device:

- TMS570LC4357ZWT
- TMS570LC4357ZWT_FREE...

Name: LCD_MODULE

Location: /home/siyun/workspace_v8/LCD_MODULE

☐ Create directory for project

Project will be created at: /home/siyun/workspace_v8/LCD_MODULE.

Tools: Texas Instruments Tools

OK Cancel



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 **
 - ☐ Enable HET drivers
 - ☐ Enable HET1 driver **
 - ☐ Enable HET2 driver **
 - ☒ Enable I2C driver **
 - ☐ Enable I2C1 driver **
 - ☒ Enable I2C2 driver **

TMS570LC4357ZWT PINMUX RTI GIO ESM SCI1 SCI2 SCI3 SCI4 LIN1

Pin Muxing Input Pin Muxing Special Pin Muxing

Enable / Disable Peripherals

<input type="checkbox"/> HET1	<input type="checkbox"/> GIOA	<input type="checkbox"/> MIBSPI2	<input type="checkbox"/> MIBSPI1	<input type="checkbox"/> SCI3	<input type="checkbox"/> RMII
<input type="checkbox"/> HET2	<input type="checkbox"/> GIOB	<input type="checkbox"/> MIBSPI4	<input type="checkbox"/> MIBSPI3	<input type="checkbox"/> SCI4	<input type="checkbox"/> MII
<input type="checkbox"/> EMIF	<input type="checkbox"/> EQEP	<input type="checkbox"/> AD1EVT	<input type="checkbox"/> MIBSPI5	<input type="checkbox"/> LIN2/SCI2	<input type="checkbox"/> CAN4
<input type="checkbox"/> ETPWM	<input type="checkbox"/> ECAP	<input type="checkbox"/> AD2EVT	<input type="checkbox"/> I2C1	<input checked="" type="checkbox"/> I2C2	

TMS570LC4357ZWT PINMUX RTI GIO ESM SCI1 SCI2 SCI3 SCI4 LIN1 LIN2

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.

Interrupts

AL INT: ☐ ☐

I2C interrupt

TMS570LC4357ZWT PINMUX RTI GIO ESM SCI1 SCI2 SCI3 SCI4 LIN1 LIN2

I2C Global I2C Clocks I2C Port

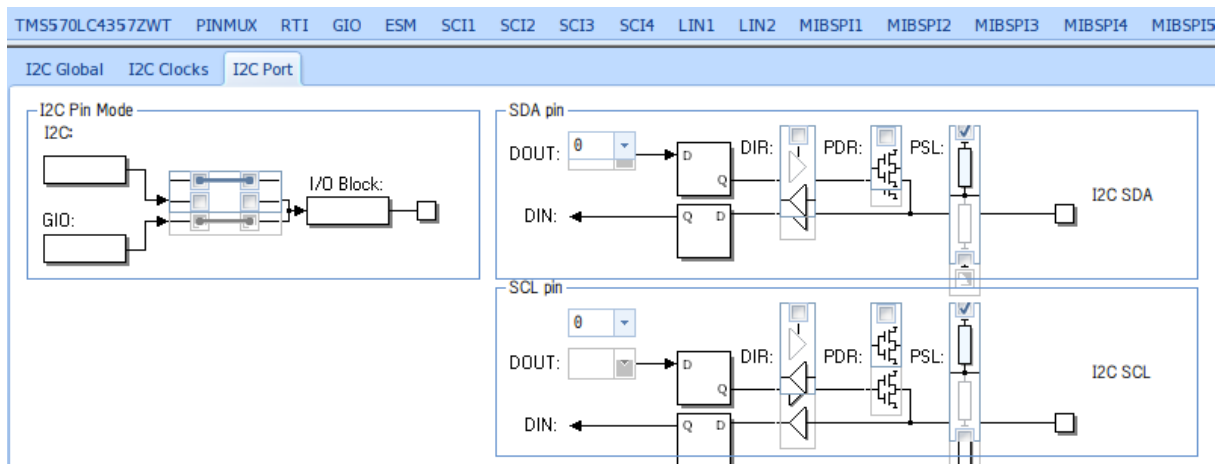
Data Format

Baudrate (KHz): 400 Prescale: 8 Module Clock Frequency 8

VCLK1 (MHz): 75.0

ICCH: 5

ICCL: 5



include option

Add directory path

Directory:

Workspace... Variables... Browse...

Cancel OK

```

#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_sned_string(char *str);
void lcd_init(void);
void lcd_send_cmd(char cmd);
void lcd_send_data(char data);

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_send_cmd(0x80);
        lcd_send_string("    Handsome    ");
        lcd_send_cmd(0xc0);
        lcd_send_string("    SiYunKim    ");
        for(i=0; i<80000000; i++);
        lcd_send_cmd(0x01);
    }
}

/* USER CODE BEGIN (4) */
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|0x0C; //en=1, rs=0
    data_t[1] = data_u|0x08; //en=0, rs=0
    data_t[2] = data_l|0x0C; //en=1, rs=0
    data_t[3] = data_l|0x08; //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|0x0D; //en=1, rs=0
    data_t[1] = data_u|0x09; //en=0, rs=0
    data_t[2] = data_l|0x0D; //en=1, rs=0
    data_t[3] = data_l|0x09; //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);
}

/* USER CODE END */

```

