**Міністерство Освіти І НАУКИ України**

**Національний університет "Львівська політехніка"**

Інститут **КНІТ**

Кафедра **ПЗ**

**ЗВІТ**

До лабораторної роботи № 5

**На тему:** *“* *Вивчення протоколу SPI на прикладі LCD-дисплею”*

**З дисципліни:** *“Основи програмування вбудованих систем”*

**Лектор:**

викладач каф. ПЗ

Марусенкова Т. А.

**Виконав:**

ст. гр. ПЗ-31

Далявський В. С.

**Прийняв:**

асистент каф. ПЗ

Ладанівський Б. Т.

«\_\_\_\_\_» \_\_\_\_\_\_\_\_ 2019 р.

∑= \_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_

Львів – 2019

**Тема роботи:** Вивчення протоколу SPI на прикладі LCD-дисплею.

**Мета роботи**: Засвоїти принципи роботи інтерфейсу SPI та здобути практичні навички організації взаємодії мікроконтролера з периферійними пристроями через SPI за допомогою бібліотек CMSIS, SPL та HAL, дисплею моделі ET-NOKIA LCD 5110.

**Індивідуальне завдання:**

Написати функцію ініціалізації GPIO та SPI для обміну даними з HAL.

Написати функцію для відображення на дисплеї парних рядків зображення.

**Теоретичні відомості**

6. Які є режими роботи SPI?

Є 4 режими роботи SPI. Вони визначаються комбінацією значень CPOL та CPHA.

**CPOL** визначає стан лінії SCL в час, коли дані не передаються.

Якщо **CPOL**=0, то на SCL низький логічний рівень, переднім фронтом вважатиметься перехід 0 -> 1. Якщо **CPOL**=1, то на SCL високий логічний рівень, переднім фронтом вважатиметься перехід 1 -> 0.

**CPHA** визначає, за яким фронтом слід здійснювати вибірку. Якщо **CPHA** = 0, то по першому фронту, якщо **CPHA** = 1 – то по другому.

**Код програми**

**LCD.h**

#ifndef LCD\_H

#define LCD\_H

#include "main.h"

#define LCD\_CS\_ON() HAL\_GPIO\_WritePin(GPIOC,GPIO\_PIN\_9,GPIO\_PIN\_SET)

#define LCD\_CS\_OFF() HAL\_GPIO\_WritePin(GPIOC,GPIO\_PIN\_9,GPIO\_PIN\_RESET)

#define LCD\_RST\_ON() HAL\_GPIO\_WritePin(GPIOC,GPIO\_PIN\_8,GPIO\_PIN\_SET)

#define LCD\_RST\_OFF() HAL\_GPIO\_WritePin(GPIOC,GPIO\_PIN\_8,GPIO\_PIN\_RESET)

#define LCD\_DC\_ON() HAL\_GPIO\_WritePin(GPIOC,GPIO\_PIN\_7,GPIO\_PIN\_SET)

#define LCD\_DC\_OFF() HAL\_GPIO\_WritePin(GPIOC,GPIO\_PIN\_7,GPIO\_PIN\_RESET)

#define LCD\_SCK\_ON() HAL\_GPIO\_WritePin(GPIOA,GPIO\_PIN\_8,GPIO\_PIN\_SET)

#define LCD\_SCK\_OFF()HAL\_GPIO\_WritePin(GPIOA,GPIO\_PIN\_8,GPIO\_PIN\_RESET)

void SysTick\_Handler(void);

void delay\_ms(uint16\_t delay\_temp);

void LCD\_MO(unsigned char OnOff);

void LCD\_GPIO\_Config(void);

void LCD\_write\_byte(unsigned char dat, unsigned char mode);

void LCD\_clear(void);

void LCD\_init(void);

void LCD\_write\_string(char \*s);

void LCD\_set\_XY(unsigned char X,unsigned char Y);

void LCD\_write\_char(unsigned char c);

void LCD\_write\_char\_reg(unsigned char c);

void LCD\_Write\_Dec(unsigned int b);

void LCD\_write\_bitmap(const uint8\_t\* pic);

void LCD\_bitmap\_lines(const uint8\_t\* pic);

#endif

**LCD.c**

#include "LCD.h"

#include "Font.h"

uint16\_t delay\_count=0;

void SysTick\_Handler(void)//1ms

{

if (delay\_count > 0){delay\_count--;}

}

void delay\_ms(uint16\_t delay\_temp)

{

delay\_count = delay\_temp;

while(delay\_count){}

}

void LCD\_MO(unsigned char OnOff)

{

if (OnOff) HAL\_GPIO\_WritePin(GPIOC,GPIO\_PIN\_6,GPIO\_PIN\_SET);

else HAL\_GPIO\_WritePin(GPIOC,GPIO\_PIN\_6,GPIO\_PIN\_RESET);

}

void LCD\_write\_string(char \*s)

{

unsigned char ch;

while(\*s!='\0')

{

ch = \*s;

LCD\_write\_char(ch);

s++;

}

}

void LCD\_GPIO\_Config(void)

{

\_\_GPIOA\_CLK\_ENABLE();

\_\_GPIOC\_CLK\_ENABLE();

GPIO\_InitTypeDef GPIOA\_Init;

GPIO\_InitTypeDef GPIOC\_Init;

GPIOA\_Init.Mode = GPIO\_MODE\_OUTPUT\_PP;

GPIOA\_Init.Pull = GPIO\_NOPULL;

GPIOA\_Init.Pin = GPIO\_PIN\_8;

GPIOA\_Init.Speed = GPIO\_SPEED\_LOW;

HAL\_GPIO\_Init(GPIOA, &GPIOA\_Init);

GPIOC\_Init.Pin = GPIO\_PIN\_6|GPIO\_PIN\_7|GPIO\_PIN\_8|GPIO\_PIN\_9;

GPIOC\_Init.Mode = GPIO\_MODE\_OUTPUT\_PP;

GPIOC\_Init.Pull = GPIO\_NOPULL;

GPIOC\_Init.Speed = GPIO\_SPEED\_LOW;

HAL\_GPIO\_Init(GPIOC, &GPIOC\_Init);

}

void LCD\_write\_byte(unsigned char dat, unsigned char mode)

{

unsigned char i;

LCD\_CS\_OFF();

if (!mode) LCD\_DC\_OFF(); else LCD\_DC\_ON();

for(i = 0; i < 8; i++)

{

LCD\_MO(dat & 0x80);

dat = dat<<1;

LCD\_SCK\_OFF();

LCD\_SCK\_ON();

}

LCD\_CS\_ON();

}

void LCD\_clear()

{

unsigned char i,j;

for(i=0;i<6;i++)

for(j=0;j<84;j++)

LCD\_write\_byte(0,1);

}

void LCD\_write\_bitmap(const uint8\_t\* pic){

unsigned char i,j;

for(i=0;i<6;i++)

for(j=0;j<84;j++)

LCD\_write\_byte(pic[i\*84+j],1);

}

void LCD\_bitmap\_lines(const uint8\_t\* pic){

unsigned char i,j;

for(i=0;i<6;i++)

for(j=0;j<84;j++)

LCD\_write\_byte(pic[i\*84+j]&0xAA,1);

}

void LCD\_init()

{

LCD\_GPIO\_Config();

LCD\_DC\_ON();

LCD\_MO(1);

LCD\_SCK\_ON();

LCD\_CS\_ON();

LCD\_RST\_OFF() ;

delay\_ms(40);

LCD\_RST\_ON();

LCD\_write\_byte(0x21,0);

LCD\_write\_byte(0xc6,0);

LCD\_write\_byte(0x06,0);

LCD\_write\_byte(0x13,0);

LCD\_write\_byte(0x20,0);

LCD\_clear();

LCD\_write\_byte(0x0c,0);

}

void LCD\_set\_XY(unsigned char X,unsigned char Y)

{

unsigned char x;

x = 6\*X;

LCD\_write\_byte(0x40|Y,0);

LCD\_write\_byte(0x80|x,0);

}

void LCD\_write\_char(unsigned char c) // write char

{

unsigned char line;

unsigned char ch = 0;

c = c - 32;

for(line=0;line<6;line++)

{

ch = font6\_8[c][line];

LCD\_write\_byte(ch,1);

}

}

void LCD\_write\_char\_reg(unsigned char c) // write invert char

{

unsigned char line;

unsigned char ch = 0;

c = c - 32;

for(line=0;line<6;line++)

{

ch = ~font6\_8[c][line];

LCD\_write\_byte(ch,1);

}

}

void LCD\_Write\_Dec(unsigned int b)

{

unsigned char datas[4];

datas[0] = b/1000; b = b - datas[0]\*1000;

datas[1] = b/100; b = b - datas[1]\*100;

datas[2] = b/10; b = b - datas[2]\*10;

datas[3] = b;

datas[0]+=48; datas[1]+=48; datas[2]+=48; datas[3]+=48;

LCD\_write\_char(datas[0]); LCD\_write\_char(datas[1]); LCD\_write\_char(datas[2]); LCD\_write\_char(datas[3]);

}

**Pictures.h**

#ifndef PICTURES\_H

#define PICTURES\_H

#include "main.h"

const uint8\_t pic1[] = {

0x00, 0x00, 0x01, 0x07, 0x07, 0x29, 0xd6, 0x01, 0xbc, 0x60, 0x60, 0x60, 0x49, 0xdf, 0xff, 0xd7, 0x07, 0xff, 0xff, 0x6f, 0xbf, 0xbf, 0x6f, 0xd7, 0xbb, 0x6d, 0xdb, 0x35, 0xae, 0x53, 0x6d, 0x6d, 0x9d, 0x6b, 0xb5, 0xef, 0x36, 0xcd, 0xfb, 0x96, 0xed, 0xe6, 0xce, 0xd4, 0xce, 0xcc, 0xc8, 0xdc, 0xb0, 0xd8, 0xb0, 0xb0, 0xb0, 0x60, 0x60, 0xa0, 0x40, 0x20, 0x00, 0x08, 0x40, 0x00, 0x48, 0x80, 0xd0, 0xc0, 0xe4, 0x78, 0x70, 0xd4, 0xf8, 0xce, 0x1c, 0x1c, 0x06, 0x04, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,

0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x03, 0x05, 0x0c, 0x0c, 0xf1, 0xc8, 0xc0, 0xd1, 0x06, 0xe4, 0xfb, 0x1f, 0xf5, 0xce, 0x39, 0xd7, 0xaa, 0x55, 0xba, 0x45, 0xdb, 0x2c, 0xd1, 0x6b, 0x6b, 0x0c, 0x51, 0x8b, 0x22, 0x15, 0x43, 0x26, 0x1d, 0x87, 0x16, 0x2d, 0x0b, 0x4b, 0x2a, 0x99, 0x28, 0x18, 0x2d, 0x4b, 0x45, 0x45, 0xa2, 0x53, 0x3e, 0xd5, 0xb2, 0x40, 0x00, 0x20, 0x6d, 0x9e, 0x05, 0x37, 0x2e, 0x59, 0x2d, 0x5c, 0x26, 0x82, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,

0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0xdf, 0xff, 0x2c, 0x51, 0xef, 0xde, 0x11, 0x66, 0xbd, 0xcb, 0xb4, 0x5b, 0xae, 0xd1, 0xad, 0x36, 0xc9, 0xb6, 0x55, 0x55, 0x89, 0x66, 0x98, 0x21, 0x52, 0x84, 0x10, 0x40, 0x00, 0x00, 0x00, 0x02, 0x00, 0x00, 0x04, 0x01, 0x00, 0x01, 0x02, 0xe4, 0xe4, 0x51, 0x2a, 0xc4, 0x29, 0x4a, 0x04, 0x00, 0x49, 0x24, 0x01, 0x09, 0x92, 0x41, 0x08, 0x93, 0x20, 0x0a, 0x04, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,

0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0xfb, 0xff, 0x46, 0x98, 0x77, 0xed, 0x96, 0x78, 0xe7, 0x99, 0x77, 0xaa, 0x55, 0xe5, 0x1b, 0xc4, 0x9b, 0x51, 0x26, 0x26, 0x99, 0x26, 0x41, 0xac, 0x12, 0x81, 0x24, 0x12, 0x40, 0x08, 0x00, 0x40, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x06, 0x0b, 0x0b, 0x44, 0x12, 0x09, 0x12, 0x34, 0x02, 0x40, 0x22, 0x84, 0x50, 0x00, 0x89, 0x24, 0x80, 0x81, 0x80, 0x40, 0xc0, 0x40, 0x40, 0x40, 0xc0, 0xa0, 0x60, 0xa0, 0xd0, 0x50, 0xa0, 0xd8, 0x60, 0xa8,

0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0xc0, 0xf0, 0xf0, 0xff, 0xff, 0x24, 0xdb, 0xf7, 0x2c, 0xdb, 0xed, 0x36, 0xdb, 0xad, 0xb7, 0x4e, 0xbb, 0x3c, 0xeb, 0x74, 0xca, 0xb4, 0x49, 0x49, 0x92, 0x64, 0x89, 0x10, 0x62, 0x84, 0x49, 0x20, 0x92, 0x09, 0x60, 0x8a, 0x09, 0x58, 0x1c, 0x6a, 0xfc, 0xdd, 0xd6, 0xbc, 0xbc, 0x59, 0x9e, 0xfd, 0xd9, 0xbd, 0xdb, 0x7f, 0x8a, 0xe6, 0xf4, 0xf8, 0x5f, 0xb7, 0x6d, 0xdd, 0xaa, 0x77, 0xfa, 0xaf, 0x55, 0x55, 0xfe, 0x35, 0xcb, 0x3e, 0x75, 0xcb, 0x3e, 0x4d, 0x9b, 0x35,

0x00, 0x00, 0x80, 0xc0, 0x40, 0xb0, 0xd8, 0x6c, 0xb7, 0xdd, 0xdd, 0xff, 0xff, 0xfa, 0x6d, 0xd7, 0xba, 0x6d, 0xd2, 0xaf, 0xd5, 0xba, 0x4d, 0xb6, 0x69, 0x46, 0x94, 0x29, 0x52, 0x8d, 0x23, 0x23, 0x93, 0x8e, 0x24, 0x9b, 0x1e, 0x30, 0x2c, 0x73, 0xc8, 0xd2, 0xa4, 0x88, 0x11, 0x22, 0x40, 0x02, 0x40, 0x00, 0x84, 0x01, 0x01, 0x91, 0x24, 0x89, 0xe4, 0x72, 0xbc, 0xcf, 0x77, 0xbb, 0xe7, 0x5f, 0xaf, 0xdf, 0xbe, 0x7d, 0xfb, 0xfa, 0xf5, 0xe2, 0xf1, 0xf1, 0xc4, 0xa9, 0x54, 0xa1, 0x52, 0xc8, 0x21, 0xca, 0x50, 0x09

};

**#endif**

**main.h**

#ifndef MAIN\_H

#define MAIN\_H

#include "stm32f4xx\_hal.h"

#include "LCD.h"

#endif

**main.c**

#include "main.h"

#include "pictures.h"

int main ()

{

HAL\_Init();

LCD\_init();

LCD\_set\_XY(0, 0);

LCD\_write\_bitmap(pic0);

while(1){}

}

**Висновки**

На даній лабораторній роботі, я навчився організовувати взаємодію мікроконтролера з периферійними пристроями через інтерфейс SPI. Написав програму, яка надсилає команди та дані для контроллера PCD8544, який керує LCD-дисплеєм.