



南方科技大学  
SOUTHERN UNIVERSITY OF SCIENCE AND TECHNOLOGY

# Embedded System and Microcomputer Principle

---

LAB10 Film Transistor-Liquid Crystal Display (TFTLCD)

---

2022 Fall  
wangq9@mail.sustech.edu.cn



# CONTENTS

- 1 TFTLCD Principle Description
- 2 TFTLCD Function Description
- 3 How to Program
- 4 Practice



01

# TFTLCD Principle Description



# 1. TFTLCD Principle Description

## -- What is TFTLCD

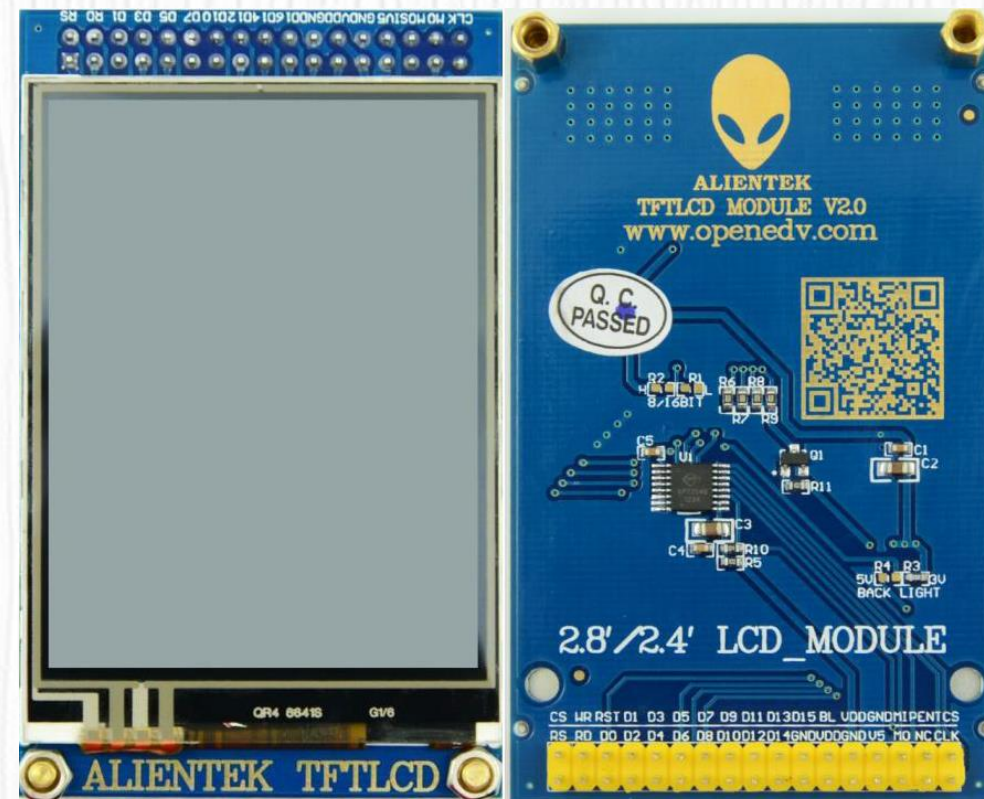
- Film Transistor-Liquid Crystal Display, 缩写TFTLCD, 即薄膜晶体管液晶显示器
- TFT液晶为每个像素都设有一个半导体开关, 每个像素都可以通过点脉冲直接控制, 因而每个节点都相对独立, 并可以连续控制, 不仅提高了显示屏的反应速度, 同时可以精确控制显示色阶, 所以TFT液晶的色彩更真
- 特点: 亮度好、对比度高、层次感强、颜色鲜艳等特点
- 目前最主流的LCD显示器
- 广泛应用于电视、手机、电脑、平板等各种电子产品



# 1. TFTLCD Principle Description

## -- ALINETEK TFTLCD

- ALINETEK 2.8寸 TFTLCD模块
- 分辨率：240\*320
- 驱动IC：ILI9341
- 16位并口驱动
- 16位真彩显示（65536色）
- 自带电阻触摸屏
- 自带背光电路
- 模块是3.3V供电的，不支持5V电压的MCU，如果是5V MCU，必须在信号线串接120R电阻使用





# 1. TFTLCD Principle Description

## -- ATK-2.8-inch TFTLCD interface

- LCD\_CS: LCD片选信号
- LCD\_WR: LCD写信号
- LCD\_RD: LCD读信号
- DB[17:1]: 16位双向数据线
- LCD\_RST: 硬复位LCD信号
- LCD\_RS: 命令/数据标志  
( 0: 命令, 1: 数据)
- BL\_CTR: 背光控制信号
- T\_MISO, T\_MOSI, T\_PEN, T\_CS, T\_CLK: 触摸屏接口信号

### 显示部分

LCD1			
LCD CS	1	LCD_CS	RS
LCD WR	3	WR/CLK	RD
LCD_RST	5	RST	DB1
DB2	7	DB2	DB3
DB4	9	DB4	DB5
DB6	11	DB6	DB7
DB8	13	DB8	DB10
DB11	15	DB11	DB12
DB13	17	DB13	DB14
DB15	19	DB15	DB16
DB17	21	DB17	GND
BL_CTR3		BL	VDD3.3
VCC3.3		VDD3.3	GND
GND	27	GND	BL_VDD
T_MISO	29	MISO	MOSI
T_PEN	31	T_PEN	MO
T_CS	33	T_CS	CLK

TFT\_LCD

### 触屏控制部分

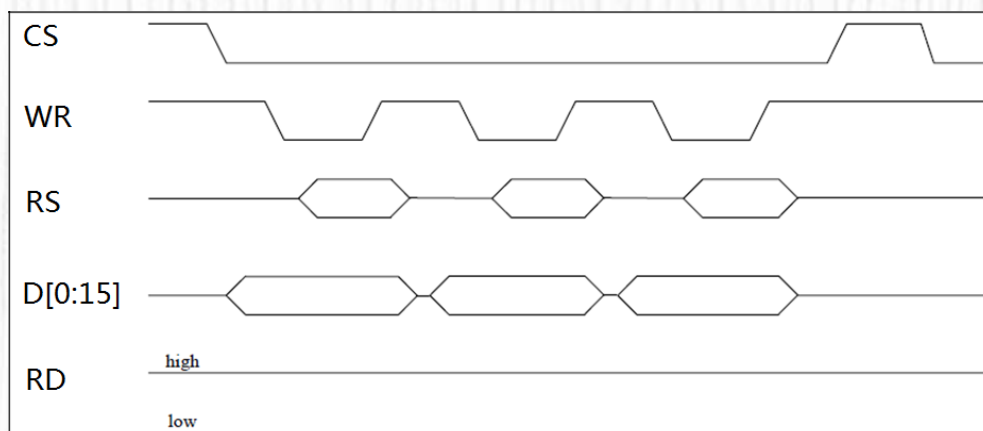




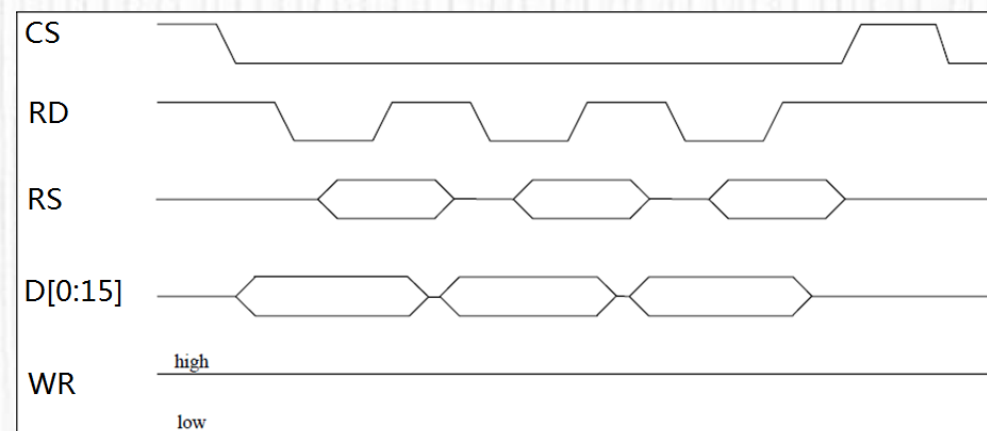
# 1. TFTLCD Principle Description

## -- 8080 Parallel port principle

- 8080并口读/写的过程为：
- 先根据要数据类型，设置RS为高（数据）/低（命令），然后拉低片选，选中ILI9341，接着根据是读数据，还是要写数据置RD/WR为低，然后：
  - 1.读数据：在RD的上升沿，读取数据线上的数据（D[15:0]）；
  - 2.写数据：在WR的上升沿，使数据写入到ILI9341里面



Writing timing diagram

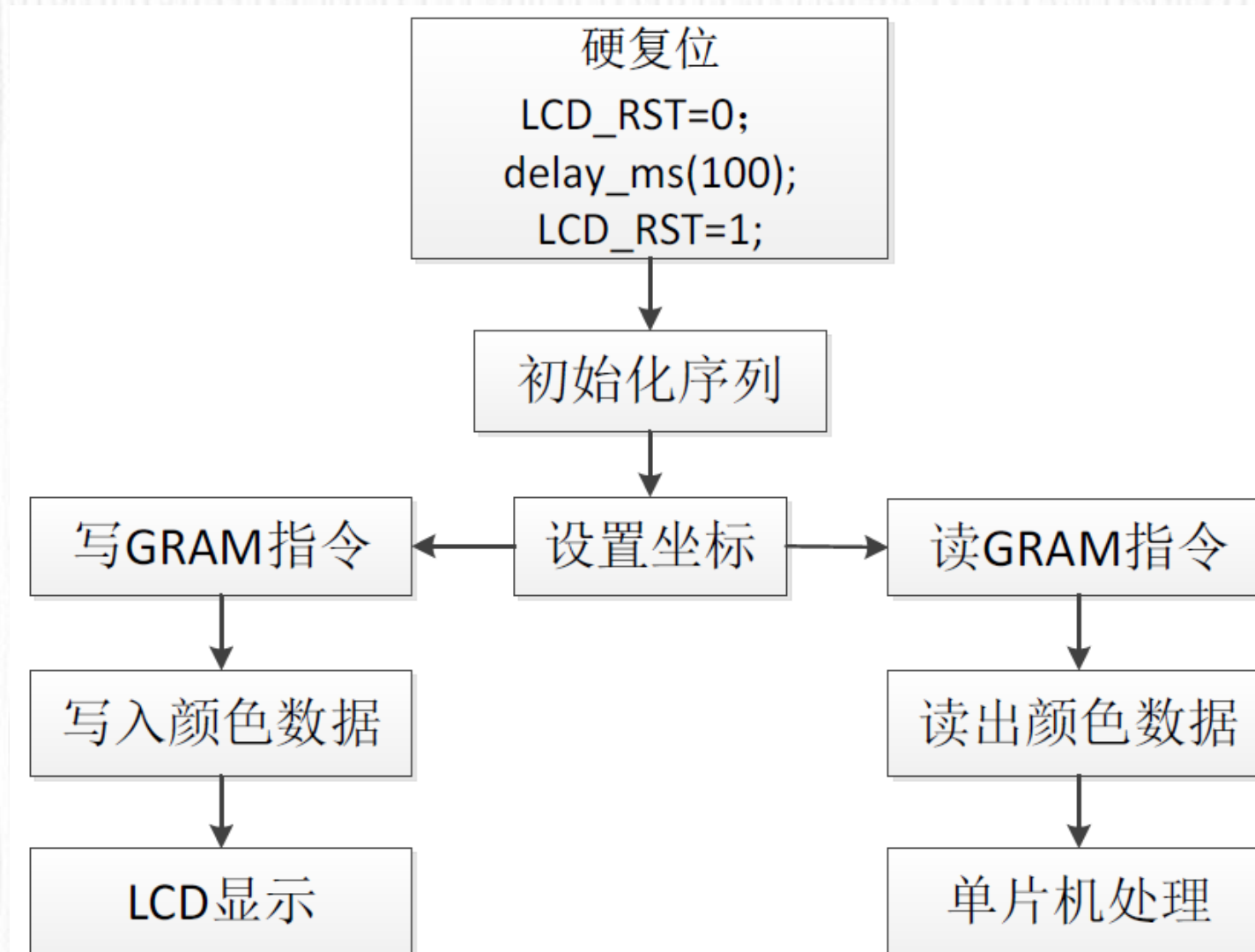


Reading timing diagram



# 1. TFTLCD Principle Description

## -- TFTLCD driving process







# 1. TFTLCD Principle Description

## -- RGB565 format description

- 模块对外接口采用16位并口
- 颜色深度为16位
- 格式为RGB565

数据线	D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
LCD GRAM	R[4]	R[3]	R[2]	R[1]	R[0]	G[5]	G[4]	G[3]	G[2]	G[1]	G[0]	B[4]	B[3]	B[2]	B[1]	B[0]



# 1. TFTLCD Principle Description

## -- ILI9341 instruction format description

- ILI9341所有的指令都是8位的（高8位无效），且参数除了读写GRAM（graphics RAM，图像寄存器）的时候是16位，其他操作参数，都是8位的。

### Regulative command set

Command Function	D/CX	RDX	WRX	D17-8	D7	D6	D5	D4	D3	D2	D1	D0	Hex
No Operation	0	1	↑	XX	0	0	0	0	0	0	0	0	00h
Software Reset	0	1	↑	XX	0	0	0	0	0	0	0	1	01h
Read Display Identification Information	0	1	↑	XX	0	0	0	0	0	1	0	0	04h
	1	↑	1	XX	X	X	X	X	X	X	X	X	XX
	1	↑	1	XX	ID1 [7:0]								XX
	1	↑	1	XX	ID2 [7:0]								XX
	1	↑	1	XX	ID3 [7:0]								XX
	1	↑	1	XX									
Read Display Status	0	1	↑	XX	0	0	0	0	1	0	0	1	09h
	1	↑	1	XX	X	X	X	X	X	X	X	X	XX
	1	↑	1	XX	D [31:25]								X
	1	↑	1	XX	X	D [22:20]			D [19:16]				61
	1	↑	1	XX	X	X	X	X	X	D [10:8]			00
	1	↑	1	XX	D [7:5]			X	X	X	X	X	00
Read Display Power Mode	0	1	↑	XX	0	0	0	0	1	0	1	0	0Ah
	1	↑	1	XX	X	X	X	X	X	X	X	X	XX
	1	↑	1	XX	D [7:2]								0
Read Display MADCTL	0	1	↑	XX	0	0	0	0	1	0	1	1	0Bh
	1	↑	1	XX	X	X	X	X	X	X	X	X	XX
	1	↑	1	XX	D [7:2]								0
Read Display Pixel Format	0	1	↑	XX	0	0	0	0	1	1	0	0	0Ch
	1	↑	1	XX	X	X	X	X	X	X	X	X	XX
	1	↑	1	XX	RIM	DPI [2:0]			X	DBI [2:0]			06
Read Display Image Format	0	1	↑	XX	0	0	0	0	1	1	0	1	0Dh
	1	↑	1	XX	X	X	X	X	X	X	X	X	XX
	1	↑	1	XX	X	X	X	X	X	D [2:0]			00
Read Display Signal Mode	0	1	↑	XX	0	0	0	0	1	1	1	0	0Eh
	1	↑	1	XX	X	X	X	X	X	X	X	X	XX
	1	↑	1	XX	D [7:2]								0
Read Display Self-Diagnostic Result	0	1	↑	XX	0	0	0	0	1	1	1	1	0Fh
	1	↑	1	XX	X	X	X	X	X	X	X	X	XX
	1	↑	1	XX	D [7:6]			X	X	X	X	X	00

[illegible]





Read Display Brightness	0	1	↑	XX	0	1	0	1	0	0	1	0	52
	1	↑	1	XX	X	X	X	X	X	X	X	X	XX
	1	↑	1	XX	DBV [7:0]								00
Write CTRL Display	0	1	↑	XX	0	1	0	1	0	0	1	1	53
	1	1	↑	XX	X	X	BCTRL	X	DD	BL	X	X	00
Read CTRL Display	0	1	↑	XX	0	1	0	1	0	1	0	0	54
	1	↑	1	XX	X	X	X	X	X	X	X	X	XX
	1	↑	1	XX	X	X	BCTRL	X	DD	BL	X	X	00
Write Content Adaptive Brightness Control	0	1	↑	XX	0	1	0	1	0	1	0	1	55
	1	1	↑	XX	X	X	X	X	X	X	C [1:0]		00
Read Content Adaptive Brightness Control	0	1	↑	XX	0	1	0	1	0	1	1	0	56
	1	↑	1	XX	X	X	X	X	X	X	X	X	XX
	1	↑	1	XX	X	X	X	X	X	X	C [1:0]		00
Write CABC Minimum Brightness	0	1	↑	XX	0	1	0	1	1	1	1	0	5E
	1	1	↑	XX	CMB [7:0]								00
Read CABC Minimum Brightness	0	1	↑	XX	0	1	0	1	0	1	1	1	5F
	1	↑	1	XX	X	X	X	X	X	X	X	X	XX
	1	↑	1	XX	CMB [7:0]								00
Read ID1	0	1	↑	XX	1	1	0	1	1	0	1	0	DA
	1	↑	1	XX	X	X	X	X	X	X	X	X	XX
	1	↑	1	XX	Module's Manufacture [7:0]								XX
Read ID2	0	1	↑	XX	1	1	0	1	1	0	1	1	DB
	1	↑	1	XX	X	X	X	X	X	X	X	X	XX
	1	↑	1	XX	LCD Module / Driver Version [7:0]								XX
Read ID3	0	1	↑	XX	1	1	0	1	1	1	0	0	DC
	1	↑	1	XX	X	X	X	X	X	X	X	X	XX
	1	↑	1	XX	LCD Module / Driver ID [7:0]								XX

# 1. TFTLCD Principle Description

## -- ILI9341 instruction set(continued)

Extended command set (continued)

Command Function	D/CX	RDX	WRX	D17-8	D7	D6	D5	D4	D3	D2	D1	D0	Hex	
RGB Interface	0	1	↑	XX	1	0	1	1	0	0	0	0	B0h	
Signal Control	1	1	↑	XX	ByPass_MODE	RCM [1:0]		X	VSPL	HSPL	DPL	EPL	40	
Frame Control (In Normal Mode)	0	1	↑	XX	1	0	1	1	0	0	0	1	B1h	
	1	1	↑	XX	X	X	X	X	X	X	DIVA [1:0]		00	
	1	1	↑	XX	X	X	X	RTNA [4:0]					1B	
Frame Control (In Idle Mode)	0	1	↑	XX	1	0	1	1	0	0	1	0	B2h	
	1	1	↑	XX	X	X	X	X	X	X	DIVB [1:0]		00	
	1	1	↑	XX	X	X	X	RTNB [4:0]					1B	
Frame Control (In Partial Mode)	0	1	↑	XX	1	0	1	1	0	0	1	1	B3h	
	1	1	↑	XX	X	X	X	X	X	X	DIVC [1:0]		00	
	1	1	↑	XX	X	X	X	RTNC [4:0]					1B	
Display Inversion Control	0	1	↑	XX	1	0	1	1	0	1	0	0	B4h	
	1	1	↑	XX	X	X	X	X	X	NLA	NLB	NLC	02	
Blanking Porch Control	0	1	↑	XX	1	0	1	1	0	1	0	1	B5h	
	1	1	↑	XX	0	VFP [6:0]							02	
	1	1	↑	XX	0	VBP [6:0]							02	
	1	1	↑	XX	0	0	0	HFP [4:0]					0A	
	1	1	↑	XX	0	0	0	HBP [4:0]					14	
Display Function Control	0	1	↑	XX	1	0	1	1	0	1	1	0	B6h	
	1	1	↑	XX	X	X	X	X	PTG [1:0]		PT [1:0]		0A	
	1	1	↑	XX	REV	GS	SS	SM	ISC [3:0]					82
	1	1	↑	XX	X	X	NL [5:0]						27	
	1	1	↑	XX	X	X	PCDIV [5:0]						XX	
Entry Mode Set	0	1	↑	XX	1	0	1	1	0	1	1	1	B7h	
	1	1	↑	XX	X	X	X	X	0	GON	DTE	GAS	07	

Extended command set (continued)

Backlight Control 1	0	1	↑	XX	1	0	1	1	1	0	0	0	B8h
	1	1	↑	XX	X	X	X	X	X	X	X	X	XX
	1	1	↑	XX	X	X	X	X	TH_UI [3:0]				04
Backlight Control 2	0	1	↑	XX	1	0	1	1	1	0	0	1	B9h
	1	1	↑	XX	X	X	X	X	X	X	X	X	XX
	1	1	↑	XX	TH_MV [3:0]				TH_ST [3:0]				B8
Backlight Control 3	0	1	↑	XX	1	0	1	1	1	0	1	0	BAh
	1	1	↑	XX	X	X	X	X	X	X	X	X	XX
	1	1	↑	XX	X	X	X	X	DTH_UI [3:0]				04
Backlight Control 4	0	1	↑	XX	1	0	1	1	1	0	1	1	BBh
	1	1	↑	XX	X	X	X	X	X	X	X	X	XX
	1	1	↑	XX	DTH_MV [3:0]				DTH_ST [3:0]				C9
Backlight Control 5	0	1	↑	XX	1	0	1	1	1	1	0	0	BCh
	1	1	↑	XX	X	X	X	X	X	X	X	X	XX
	1	1	↑	XX	DIM2 [3:0]				X	DIM1 [2:0]			44
Backlight Control 7	0	1	↑	XX	1	0	1	1	1	1	1	0	BEh
	1	1	↑	XX	PWM_DIV [7:0]								0F
Backlight Control 8	0	1	↑	XX	1	0	1	1	1	1	1	1	BFh
	1	1	↑	XX	X	X	X	X	X	LEDONR	LEDONPOL	LEDPWMOPL	00
Power Control 1	0	1	↑	XX	1	1	0	0	0	0	0	0	C0h
	1	1	↑	XX	X	X	VRH [5:0]						26
Power Control 2	0	1	↑	XX	1	1	0	0	0	0	0	1	C1h
	1	1	↑	XX	X	X	X	X	X	BT [2:0]			00
VCOM Control 1	0	1	↑	XX	1	1	0	0	0	1	0	1	C5h
	1	1	↑	XX	X	VMH [6:0]							31
VCOM Control 2	1	1	↑	XX	X	VML [6:0]							3C
	0	1	↑	XX	1	1	0	0	0	1	1	1	C7h
	1	1	↑	XX	nVM	VMF [6:0]							C0

# 1. TFTLCD Principle Description

## -- ILI9341 instruction set(continued)

### Extended command set (continued)

NV Memory Write	0	1	↑	XX	1	1	0	1	0	0	0	0	D0h
	1	1	↑	XX	X	X	X	X	X	PGM_ADR [2:0]			00
	1	1	↑	XX	PGM_DATA [7:0]								XX
NV Memory Protection Key	0	1	↑	XX	1	1	0	1	0	0	0	1	D1h
	1	1	↑	XX	KEY [23:16]								55
	1	1	↑	XX	KEY [15:8]								AA
	1	1	↑	XX	KEY [7:0]								66
NV Memory Status Read	0	1	↑	XX	1	1	0	1	0	0	1	0	D2h
	1	↑	1	XX	X	X	X	X	X	X	X	X	XX
	1	↑	1	XX	X	ID2_CNT [2:0]			X	ID1_CNT [2:0]			XX
	1	↑	1	XX	BUSY	VMF_CNT [2:0]			X	ID3_CNT [2:0]			XX
Read ID4	0	↑	1	XX	1	1	0	1	0	0	1	1	D3h
	1	↑	1	XX	X	X	X	X	X	X	X	X	XX
	1	↑	1	XX	0	0	0	0	0	0	0	0	00
	1	↑	1	XX	1	0	0	1	0	0	1	1	93
	1	↑	1	XX	0	1	0	0	0	0	0	1	41
Positive Gamma Correction	0	1	↑	XX	1	1	1	0	0	0	0	0	E0h
	1	1	↑	XX	X	X	X	X	VP0 [3:0]				08
	1	1	↑	XX	X	X	VP1 [5:0]						0E
	1	1	↑	XX	X	X	VP2 [5:0]						12
	1	1	↑	XX	X	X	X	X	VP4 [3:0]				05
	1	1	↑	XX	X	X	X	VP6 [4:0]					03
	1	1	↑	XX	X	X	X	X	VP13 [3:0]				09
	1	1	↑	XX	X	VP20 [6:0]						47	
	1	1	↑	XX	VP36 [3:0]				VP27 [3:0]				86
	1	1	↑	XX	X	VP43 [6:0]						2B	
	1	1	↑	XX	X	X	X	X	VP50 [3:0]				0B
	1	1	↑	XX	X	X	X	VP57 [4:0]					04
	1	1	↑	XX	X	X	X	X	VP59 [3:0]				00
	1	1	↑	XX	X	X	VP61 [5:0]						00
	1	1	↑	XX	X	X	VP62 [5:0]						00
	1	1	↑	XX	X	X	X	X	VP63 [3:0]				00



### Extended command set (continued)

Negative Gamma Correction	0	1	↑	XX	1	1	1	0	0	0	0	1	E1h	
	1	1	↑	XX	X	X	X	X	VN0 [3:0]				08	
	1	1	↑	XX	X	X	VN1 [5:0]						1A	
	1	1	↑	XX	X	X	VN2 [5:0]						20	
	1	1	↑	XX	X	X	X	X	VN4 [3:0]				07	
	1	1	↑	XX	X	X	X	VN6 [4:0]					0E	
	1	1	↑	XX	X	X	X	X	VN13 [3:0]				05	
	1	1	↑	XX	X	VN20 [6:0]							3A	
	1	1	↑	XX	VN36 [3:0]					VN27 [3:0]				8A
	1	1	↑	XX	X	VN43 [6:0]							40	
	1	1	↑	XX	X	X	X	X	VN50 [3:0]				04	
	1	1	↑	XX	X	X	X	VN57 [4:0]					18	
	1	1	↑	XX	X	X	X	X	VN59 [3:0]				0F	
	1	1	↑	XX	X	X	VN61 [5:0]						3F	
	1	1	↑	XX	X	X	VN62 [5:0]						3F	
	1	1	↑	XX	X	X	X	X	VN63 [3:0]				0F	
Digital Gamma Control 1	0	1	↑	XX	1	1	1	0	0	0	1	0	E2h	
1 <sup>st</sup> Parameter	1	1	↑	XX	RCA0 [3:0]					BCA0 [3:0]				XX
:	1	1	↑	XX	RCAx [3:0]					BCAx [3:0]				XX
16 <sup>th</sup> Parameter	1	1	↑	XX	RCA15 [3:0]					BCA15 [3:0]				XX
Digital Gamma Control 2	0	1	↑	XX	1	1	1	0	0	0	1	1	E3h	
1 <sup>st</sup> Parameter	1	1	↑	XX	RFA0 [3:0]					BFA0 [3:0]				XX
:	1	1	↑	XX	RFAx [3:0]					BFAx [3:0]				XX
64 <sup>th</sup> Parameter	1	1	↑	XX	RFA63 [3:0]					BFA63 [3:0]				XX
Interface Control	0	1	↑	XX	1	1	1	1	0	1	1	0	F6h	
	1	1	↑	XX	MY_EOR	MX_EOR	MV_EOR	X	BGR_EOR	X	X	WEMODE	01	
	1	1	↑	XX	X	X	EPF [1:0]		X	X	MDT [1:0]		00	
	1	1	↑	XX	X	X	ENDIAN	X	DM [1:0]		RM	RIM	00	





# 1. TFTLCD Principle Description

## -- 0xD3 instruction

- 该指令为读ID4指令，用于读取LCD控制器的ID。因此，同一个代码，可以根据ID的不同，执行不同的LCD驱动初始化，以兼容不同的LCD屏幕。

顺序	控制			各位描述									HEX
	RS	RD	WR	D15~D8	D7	D6	D5	D4	D3	D2	D1	D0	
指令	0	1	↑	XX	1	1	0	1	0	0	1	1	D3H
参数 1	1	↑	1	XX	X	X	X	X	X	X	X	X	X
参数 2	1	↑	1	XX	0	0	0	0	0	0	0	0	00H
参数 3	1	↑	1	XX	1	0	0	1	0	0	1	1	93H
参数 4	1	↑	1	XX	0	1	0	0	0	0	0	1	41H

# 1. TFTLCD Principle Description

## -- 0x36 instruction

- 该指令为**存储访问控制指令**，可以控制ILI9341存储器的读写方向，简单的说，就是在连续写GRAM的时候，**可以控制GRAM指针的增长方向**，从而控制显示方式（读GRAM也是一样）。

顺序	控制			各位描述									HEX
	RS	RD	WR	D15~D8	D7	D6	D5	D4	D3	D2	D1	D0	
指令	0	1	↑	XX	0	0	1	1	0	1	1	0	36H
参数	1	1	↑	XX	MY	MX	MV	ML	BGR	MH	0	0	0

Bit	Name	Description
MY	Row Address Order	These 3 bits control MCU to memory write/read direction.
MX	Column Address Order	
MV	Row / Column Exchange	
ML	Vertical Refresh Order	LCD vertical refresh direction control.
BGR	RGB-BGR Order	Color selector switch control (0=RGB color filter panel, 1=BGR color filter panel)
MH	Horizontal Refresh ORDER	LCD horizontal refreshing direction control.

控制位			效果
MY	MX	MV	LCD 扫描方向（GRAM 自增方式）
0	0	0	从左到右, 从上到下
1	0	0	从左到右, 从下到上
0	1	0	从右到左, 从上到下
1	1	0	从右到左, 从下到上
0	0	1	从上到下, 从左到右
0	1	1	从上到下, 从右到左
1	0	1	从下到上, 从左到右
1	1	1	从下到上, 从右到左

# 1. TFTLCD Principle Description

## -- 0x2A instruction

- 该指令是列地址设置指令，在从左到右，从上到下的扫描方式（默认）下面，该指令用于设置横坐标（x坐标）。
- 在默认扫描方式时，该指令用于设置x坐标，该指令带有4个参数，实际上是2个坐标值：SC和EC，即列地址的起始值和结束值，SC必须小于等于EC，且 $0 \leq SC/EC \leq 239$ （与分辨率相关）。一般在设置x坐标的时候，我们只需要带2个参数即可，也就是设置SC即可，因为如果EC没有变化，我们只需要设置一次即可（在初始化ILI9341的时候设置），从而提高速度。

顺序	控制			各位描述									HEX
	RS	RD	WR	D15~D8	D7	D6	D5	D4	D3	D2	D1	D0	
指令	0	1	↑	XX	0	0	1	0	1	0	1	0	2AH
参数 1	1	1	↑	XX	SC15	SC14	SC13	SC12	SC11	SC10	SC9	SC8	SC
参数 2	1	1	↑	XX	SC7	SC6	SC5	SC4	SC3	SC2	SC1	SC0	
参数 3	1	1	↑	XX	EC15	EC14	EC13	EC12	EC11	EC10	EC9	EC8	EC
参数 4	1	1	↑	XX	EC7	EC6	EC5	EC4	EC3	EC2	EC1	EC0	



# 1. TFTLCD Principle Description

## -- 0x2B instruction

- 该指令是页地址设置指令，在从左到右，从上到下的扫描方式（默认）下面，该指令用于设置纵坐标（y坐标）。
- 在默认扫描方式时，该指令用于设置y坐标，该指令带有4个参数，实际上是2个坐标值：SP和EP，即页地址的起始值和结束值，SP必须小于等于EP，且 $0 \leq SP/EP \leq 319$ （与分辨率相关）。一般在设置y坐标的时候，我们只需要带2个参数即可，也就是设置SP即可，因为如果EP没有变化，我们只需要设置一次即可（在初始化ILI9341的时候设置），从而提高速度。

顺序	控制			各位描述									HEX
	RS	RD	WR	D15~D8	D7	D6	D5	D4	D3	D2	D1	D0	
指令	0	1	↑	XX	0	0	1	0	1	0	1	0	2BH
参数 1	1	1	↑	XX	SP15	SP14	SP13	SP12	SP11	SP10	SP9	SP8	SP
参数 2	1	1	↑	XX	SP7	SP6	SP5	SP4	SP3	SP2	SP1	SP0	
参数 3	1	1	↑	XX	EP15	EP14	EP13	EP12	EP11	EP10	EP9	EP8	EP
参数 4	1	1	↑	XX	EP7	EP6	EP5	EP4	EP3	EP2	EP1	EP0	

## -- 0x2C instruction

- | 顺序    | 控制 |    |    | 各位描述      |    |    |    |    |    |    |    |    | HEX |
|-------|----|----|----|-----------|----|----|----|----|----|----|----|----|-----|
|       | RS | RD | WR | D15~D8    | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |     |
| 指令    | 0  | 1  | ↑  | XX        | 0  | 0  | 1  | 0  | 1  | 1  | 0  | 0  | 2CH |
| 参数 1  | 1  | 1  | ↑  | D1[15: 0] |    |    |    |    |    |    |    |    | XX  |
| ..... | 1  | 1  | ↑  | D2[15: 0] |    |    |    |    |    |    |    |    | XX  |
| 参数 n  | 1  | 1  | ↑  | Dn[15: 0] |    |    |    |    |    |    |    |    | XX  |

顺序	控制			各位描述												HEX
	RS	RD	WR	D15~D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0	
指令	0	1	↑	XX				0	0	1	0	1	1	1	0	2EH
参数 1	1	↑	1	XX												dummy
参数 2	1	↑	1	R1[4:0]	XX			G1[5:0]						XX	R1G1	
参数 3	1	↑	1	B1[4:0]	XX			R2[4:0]					XX		B1R2	
参数 4	1	↑	1	G2[5:0]			XX	B2[4:0]					XX		G2B2	
参数 5	1	↑	1	R3[4:0]	XX			G3[5:0]						XX	R3G3	
参数 N	1	↑	1	按以上规律输出												





02

## TFTLCD Function Description



## 2. TFTLCD Function Description

-- \_lcd\_dev structure

```
typedef struct {  
    u16 width;           //LCD 宽度  
    u16 height;          //LCD 高度  
    u16 id;              //LCD ID  
    u8  dir;             //横屏还是竖屏控制： 0， 竖屏； 1， 横屏。  
    u16 wramcmd;         //开始写GRAM指令  
    u16 setxcmd;          //设置x坐标指令  
    u16 setycmd;          //设置y坐标指令  
}_lcd_dev;  
//LCD参数 in lcd.h  
extern _lcd_dev lcddev;  //管理LCD重要参数
```



# 1. TFTLCD Principle Description

## -- Some functions

- LCD初始化函数: `void LCD_Init(void)`
- 写寄存器值函数(通过 8080 并口向 LCD 模块写入寄存器命令):  
`void LCD_WR_REG(u16 regval)`
- 写数据函数(写入16位的数据): `void LCD_WR_DATA(u16 data)`
- 读数据函数: `u16 LCD_RD_DATA(void)`
- 写寄存器内容函数: `void LCD_WriteReg(u16 LCD_Reg, u16 LCD_RegValue)`
- 读寄存器内容函数: `u16 LCD_ReadReg(u16 LCD_Reg)`
- 开始写GRAM函数: `void LCD_WriteRAM_Prepare(void)`
- 写GRAM函数: `void LCD_WriteRAM(u16 RGB_Code)`
- 坐标设置函数: `void LCD_SetCursor(u16 Xpos, u16 Ypos)`
- 画点函数: `void LCD_DrawPoint(u16 x, u16 y)`
- LCD读点函数: `u16 LCD_ReadPoint(u16 x, u16 y)`





# 1. TFTLCD Principle Description

## -- Some functions(continued)

- LCD字符显示函数: void LCD\_ShowChar(u16 x, u16 y, u8 num, u8 size, u8 mode)
- void LCD\_Clear(uint16\_t Color); // Clear the screen with specific color
- void LCD\_DrawLine(uint16\_t x1, uint16\_t y1, uint16\_t x2, uint16\_t y2);  
// Draw a line
- void LCD\_DrawRectangle(uint16\_t x1, uint16\_t y1, uint16\_t x2, uint16\_t y2); // Draw a rectangle
- void LCD\_Fill(uint16\_t sx, uint16\_t sy, uint16\_t ex, uint16\_t ey, uint16\_t color); // Fill the area with color
- void LCD\_ShowNum(uint16\_t x, uint16\_t y, uint32\_t num, uint8\_t len, uint8\_t size); // Display number without the leading zeros
- void LCD\_ShowString(uint16\_t x, uint16\_t y, uint16\_t width, uint16\_t height, uint8\_t size, uint8\_t \*p); // Display a string



# 1. TFTLCD Principle Description

## -- Character code table

- **//PC2LCD2002取模方式设置：阴码+逐列式+顺向+C51格式**
- //总共：3个字符集（12\*12、16\*16和24\*24） in font.h
- //每个字符所占用的字节数为： $(size/8 + ((size\%8)?1:0)) * (size/2)$ , 其中size是字库生成时的点阵大小(12/16/24...)

### //12\*12 ASCII字符集点阵

```
const unsigned char asc2_1206[95][12]={
{0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00},/*" ",0*/
{0x00,0x00,0x00,0x00,0x3F,0x40,0x00,0x00,0x00,0x00,0x00,0x00},/*"! ",1*/
.....
{0x40,0x00,0x80,0x00,0x40,0x00,0x20,0x00,0x20,0x00,0x40,0x00},/*"~",94*/};
```

### //16\*16 ASCII字符集点阵

```
const unsigned char asc2_1608[95][16]={
{0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00},/*" ",0*/
.....
```

```
{0x00,0x00,0x60,0x00,0x80,0x00,0x80,0x00,0x40,0x00,0x40,0x00,0x20,0x00,0x20,0x00},/*"~",94*/};
```

### //24\*24 ASCII字符集点阵 .....



# 1. TFTLCD Principle Description

## -- PCtoLCD2002

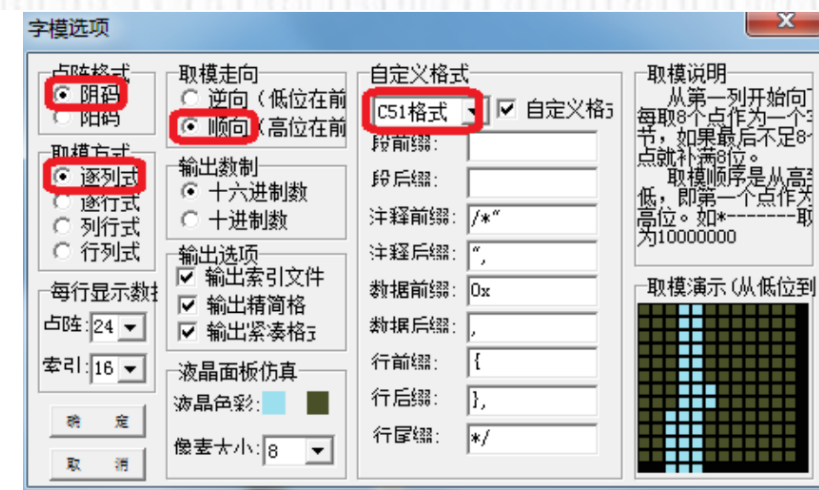
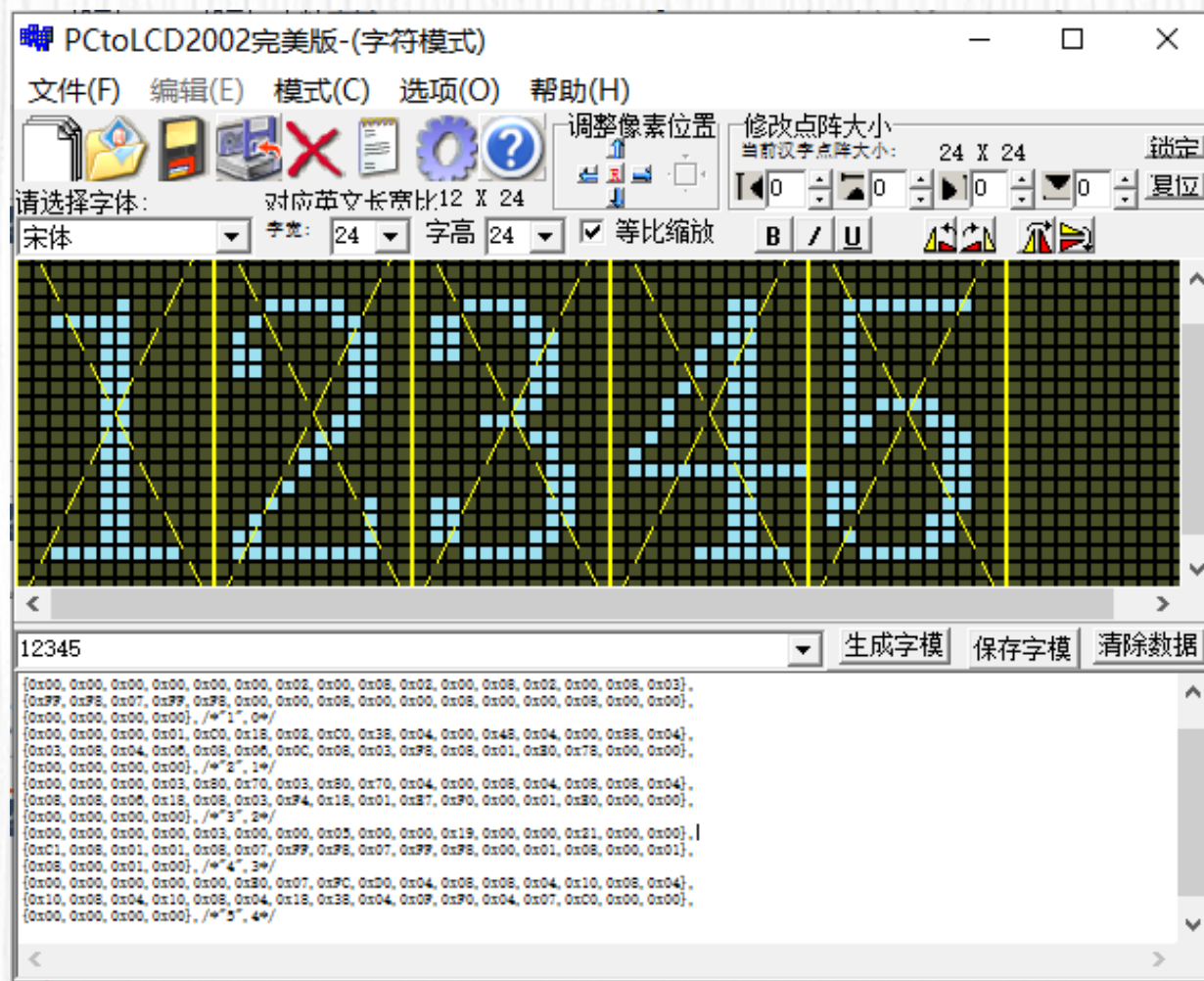
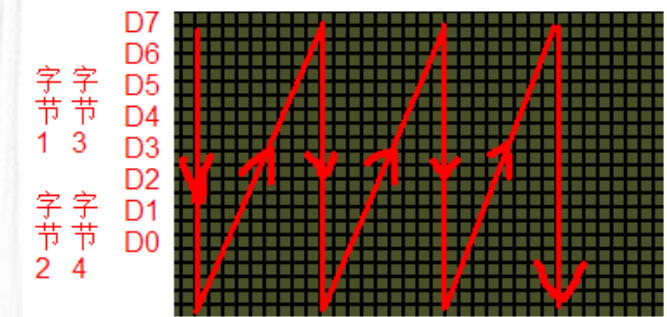


图 17.3.2 设置取模方式

上图设置的取模方式, 在右上角的取模说明里面有, 即: 从第一列开始向下每取 8 个点作为一个字节, 如果最后不足 8 个点就补满 8 位。取模顺序是从高到低, 即第一个点作为最高位。如\*-----取为 10000000。其实就是按如图 17.3.3 所示的这种方式:







03

## How to Program

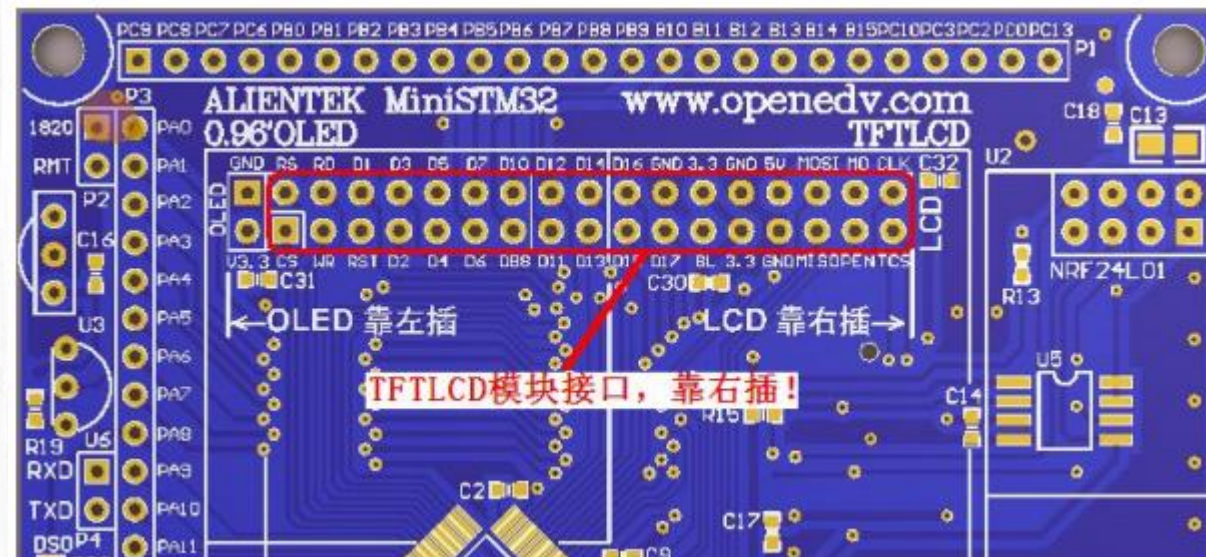
### 3. How to Program

- Our Goal
  - Show charactes on TFTLCD



### 3. How to Program

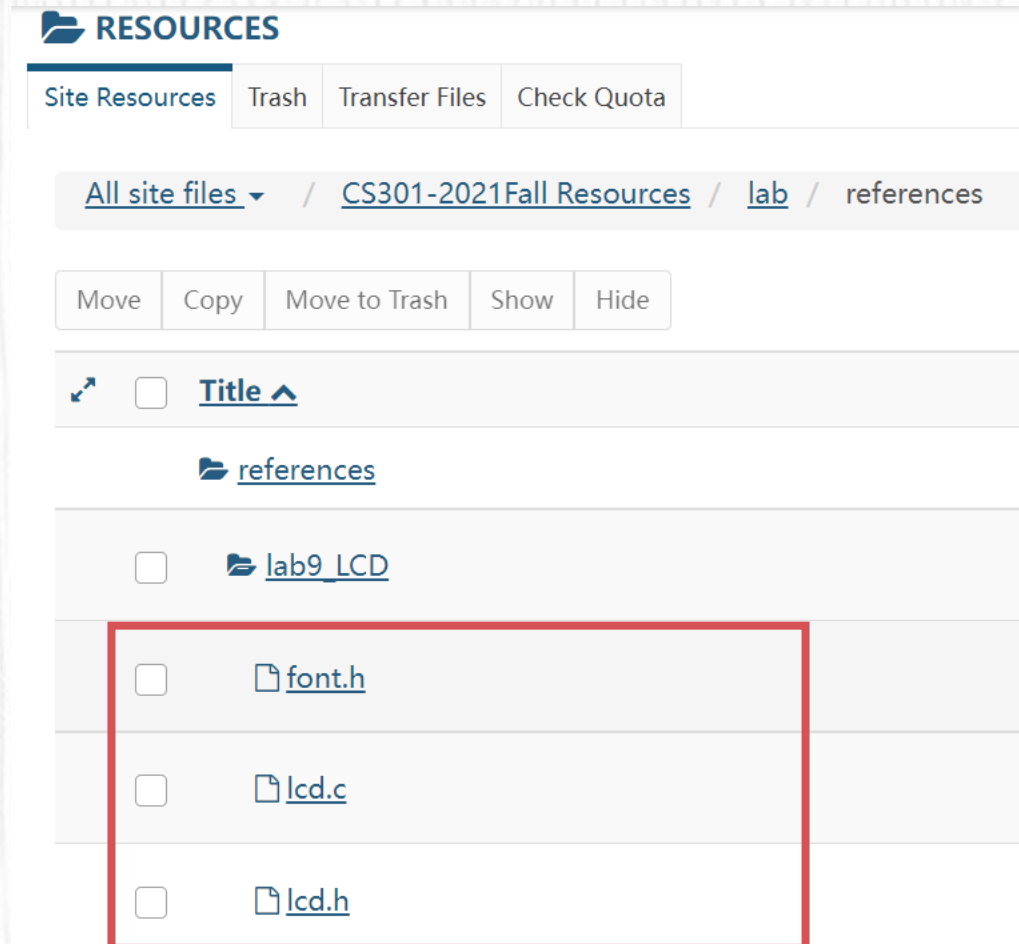
- Hardware connection
  - MiniSTM32 开发板底板的 LCD 接口和 ALIENTEK TFTLCD 模块直接可以对插（靠右插！）
  - 多出的 2 个口是给 OLED 用的
  - TFTLCD 模块与 MiniSTM32 开发板的 IO 口对应关系如下：
    - LCD\_LED 对应 PC10
    - LCD\_CS 对应 PC9
    - LCD\_RS 对应 PC8
    - LCD\_WR 对应 PC7
    - LCD\_RD 对应 PC6
    - LCD\_D[17:1] 对应 PB[15:0]





### 3. How to Program

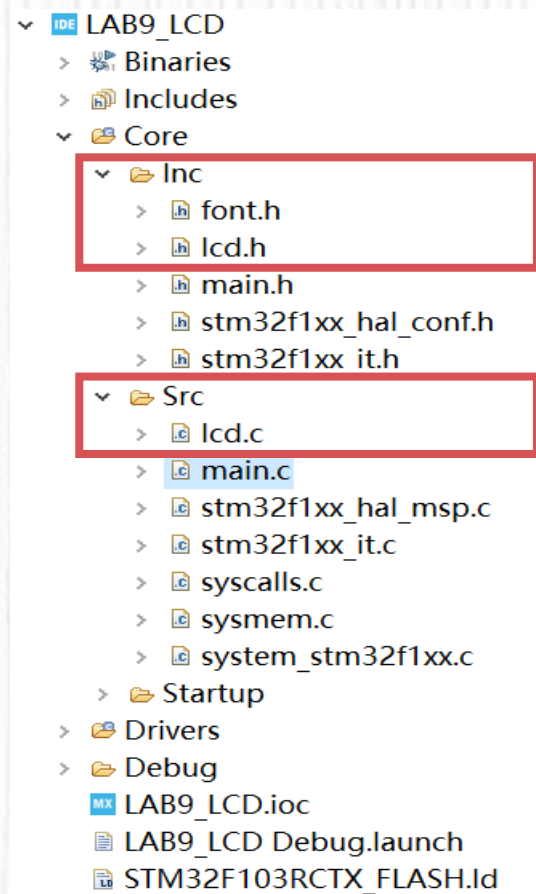
- Download lcd.c, lcd.h and font.h from Sakai site





### 3. How to Program

- Add the files in your STM32Cube project. Source files should be in Src folder, while header files should be in Inc folder.





## 3. How to Program

- Add the following codes in main.c

```
// .....
/* USER CODE BEGIN Includes */
#include "lcd.h"
/* USER CODE END Includes */
int main(void)
{
    // ... ..
    /* USER CODE BEGIN SysInit */
    LCD_Init();
    /* USER CODE END SysInit */
    // .....
    /* Infinite loop */
    /* USER CODE BEGIN WHILE */
    uint8_t x = 0;
    while (1) {
        /* USER CODE END WHILE */
```

```
/* USER CODE BEGIN 3 */

switch (x) {
case 0: LCD_Clear(WHITE); BACK_COLOR = WHITE; break;
case 1: LCD_Clear(BLACK); BACK_COLOR = BLACK; break;
case 2: LCD_Clear(BLUE); BACK_COLOR = BLUE; break;
case 3: LCD_Clear(RED); BACK_COLOR = RED; break;
case 4: LCD_Clear(MAGENTA); BACK_COLOR = MAGENTA; break;
case 5: LCD_Clear(GREEN); BACK_COLOR = GREEN; break;
case 6: LCD_Clear(CYAN); BACK_COLOR = CYAN; break;
case 7: LCD_Clear(YELLOW); BACK_COLOR = YELLOW; break;
case 8: LCD_Clear(BRRED); BACK_COLOR = BRRED; break;
case 9: LCD_Clear(GRAY); BACK_COLOR = GRAY; break;
case 10: LCD_Clear(LGRAY); BACK_COLOR = LGRAY; break;
case 11: LCD_Clear(BROWN); BACK_COLOR = BROWN; break;
} //end of switch
```





### 3. How to Program

- Add the following codes in main.c

```
POINT_COLOR = RED;  
LCD_ShowString(30, 40, 200, 24, 24, (uint8_t*) "Mini STM32 ^_^");  
LCD_ShowString(30, 70, 200, 16, 16, (uint8_t*) "TFTLCD TEST");
```

```
POINT_COLOR = BLACK;  
LCD_DrawRectangle(30, 150, 210, 190);  
LCD_Fill(31, 151, 209, 189, YELLOW);
```

```
x++;
```

```
if (x == 12)  
    x = 0;
```

```
HAL_Delay(2000);  
} //end of while(1)
```

```
} //end of main
```



04

Practice

## 4. Practice

- Run the demo on MiniSTM32 board

