Embedded System and Microcomputer Principle

LAB3 Film Transistor-Liquid Crystal Display (TFTLCD)

2024 Fall

wangq9@mail.sustech.edu.cn

CONTENTS

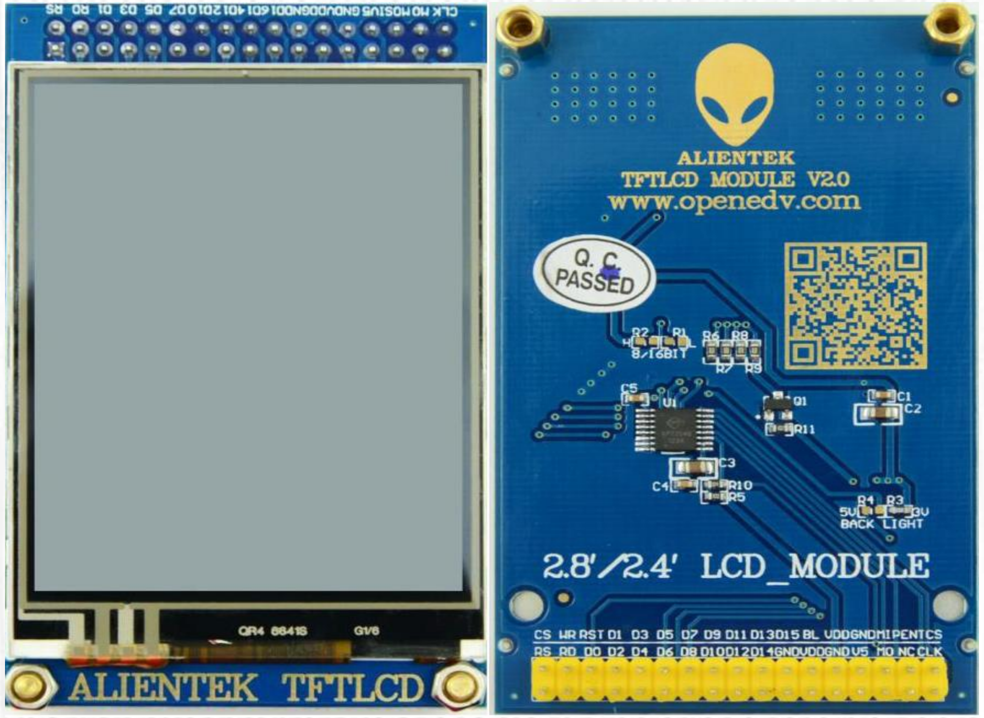
01

TFTLCD Principle Description

1. TFTLCD Principle Description
-- What is TFTLCD

- Film transistor liquid crystal display (TFTLCD)
- TFT liquid crystal is equipped with a semiconductor switch for each pixel, and each pixel can be directly controlled by point pulse, so each node is relatively independent and can be continuously controlled
- Features: good brightness, high contrast, strong sense of hierarchy, bright color and so on
- At present, the most mainstream LCD display
- It is widely used in TV, mobile phone, computer, tablet and other electronic products

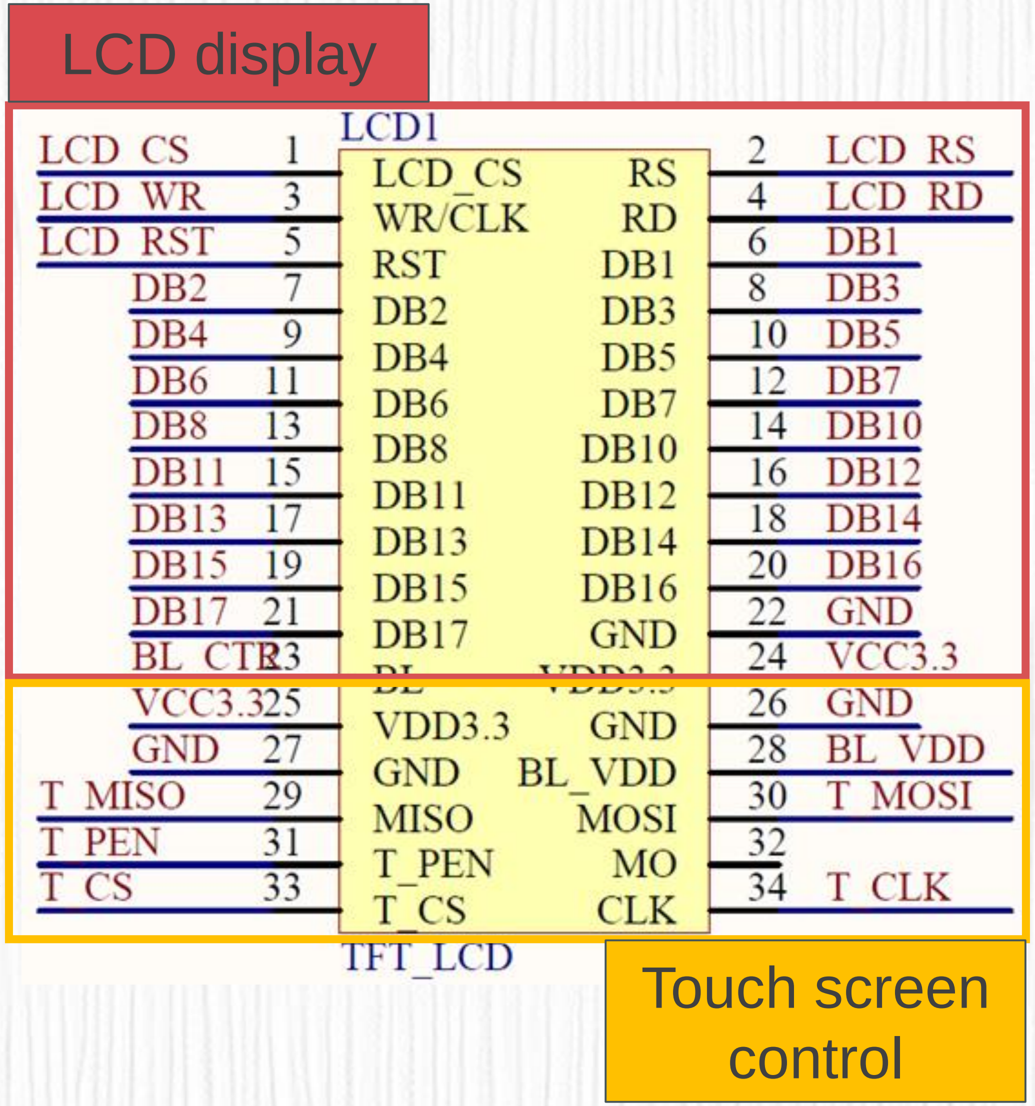
1. TFTLCD Principle Description
-- ALINETEK TFTLCD



- ALINETEK 2.8 inch TFTLCD
- 240RGBx320 Resolution
- 262K color
- Driving IC： V3: ILI9341
V4: ST7789
- Built in resistive touch screen
- Built in backlight circuit
- Supports parallel 16-bit data bus interface
- 3.3V powered MCU that does not support 5V voltage. In case of 5V MCU, 120R resistor must be connected

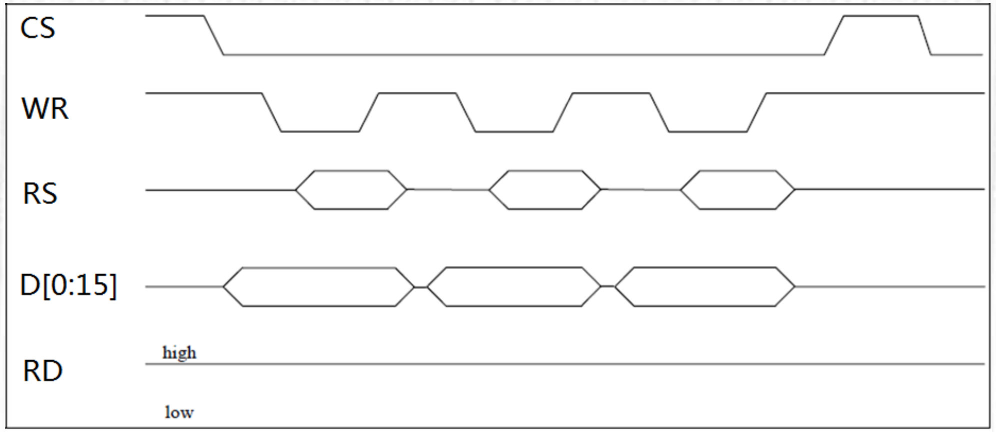
1. TFTLCD Principle Description
-- ATK-2.8-inch TFTLCD interface

- LCD_CS： LCD chip selection
- LCD_WR： LCD write signal
- LCD_RD： LCD read signal
- DB[17:1]： 16-bit dual data bus
- LCD_RST： LCD reset
- LCD_RS： command/data flag (0:command,1:data)
- BL_CTR： backlight control
- T_MISO/T_MOSI/T_PEN/T_CS/T_CLK， touch screen control

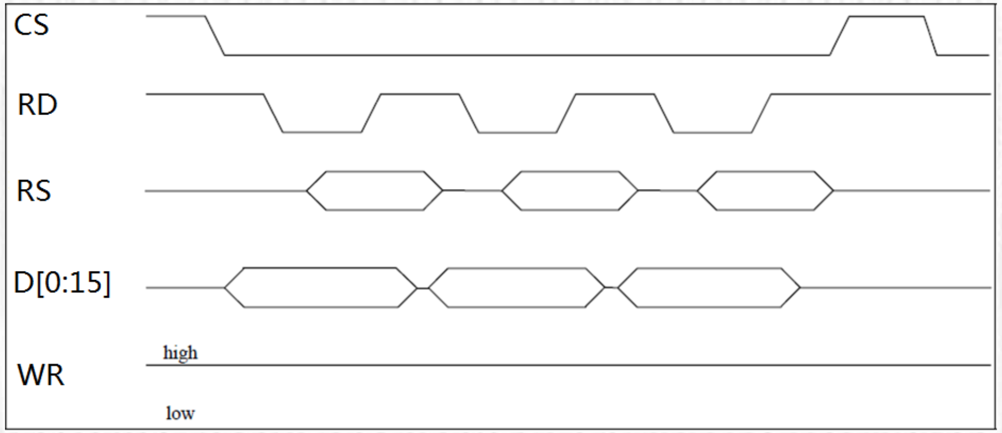


1. TFTLCD Principle Description
-- 8080 Parallel port principle

- The process of 8080 parallel port reading / writing is as follows:
- First, set RS to high (data) / low (command) according to the data type, then pull down the chip selection to select ILI9341, and then set RD/WR to low according to whether to read data or write data, and then
- 1. Read data: on the rising edge of RD signal, read the data on the data bus (D[15:0]);
- 2. Write data: write data into ILI9341 at the rising edge of WR

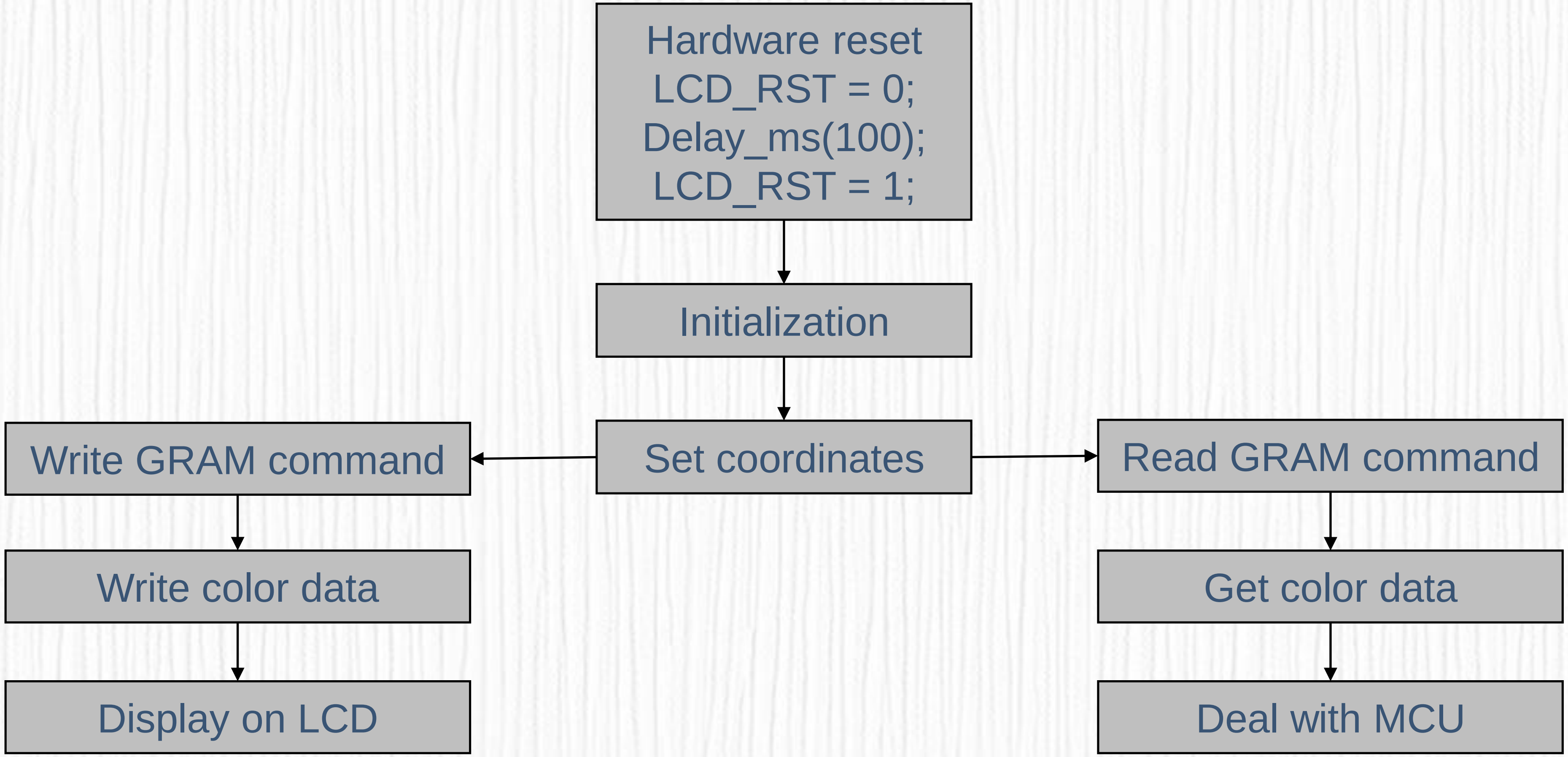


Writing timing diagram



Reading timing diagram

1. TFTLCD Principle Description
-- TFTLCD driving process



1. TFTLCD Principle Description

-- RGB565 format description

- The external interface adopts 16-bit parallel port
- The color depth is 16 bits
- The format is RGB565

Data line	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

- All instructions of ILI9341 are 8 bits (the upper 8 bits are invalid)
- The parameters are 16 bits when reading and writing GRAM (graphics RAM), and other operation parameters are 8 bits.

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
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	00
	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	0	08
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	0	00
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	0	00
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	X	00

1. TFTLCD Principle Description
-- ILI9341 instruction set(continued)

Regulative command set (continued)

Enter Sleep Mode	0	1	↑	XX	0	0	0	1	0	0	0	0	10h
Sleep OUT	0	1	↑	XX	0	0	0	1	0	0	0	1	11h
Partial Mode ON	0	1	↑	XX	0	0	0	1	0	0	1	0	12h
Normal Display Mode ON	0	1	↑	XX	0	0	0	1	0	0	1	1	13h
Display Inversion OFF	0	1	↑	XX	0	0	1	0	0	0	0	0	20h
Display Inversion ON	0	1	↑	XX	0	0	1	0	0	0	0	1	21h
Gamma Set	0	1	↑	XX	0	0	1	0	0	1	1	0	26h
	1	1	↑	XX	GC [7:0]								01
Display OFF	0	1	↑	XX	0	0	1	0	1	0	0	0	28h
Display ON	0	1	↑	XX	0	0	1	0	1	0	0	1	29h
Column Address Set	0	1	↑	XX	0	0	1	0	1	0	1	0	2Ah
	1	1	↑	XX	SC [15:8]								XX
	1	1	↑	XX	SC [7:0]								XX
	1	1	↑	XX	EC [15:8]								XX
	1	1	↑	XX	EC [7:0]								XX
Page Address Set	0	1	↑	XX	0	0	1	0	1	0	1	1	2Bh
	1	1	↑	XX	SP [15:8]								XX
	1	1	↑	XX	SP [7:0]								XX
	1	1	↑	XX	EP [15:8]								XX
	1	1	↑	XX	EP [7:0]								XX

Regulative command set (continued)

Memory Write	0	1	↑	XX	0	0	1	0	1	1	0	0	2Ch
	1	1	↑	D [17:0]									XX
Color SET	0	1	↑	XX	0	0	1	0	1	1	0	1	2Dh
	1	↑	1	XX			R00 [5:0]						XX
	1	↑	1	XX			Rnn [5:0]						XX
	1	↑	1	XX			R31 [5:0]						XX
	1	↑	1	XX			G00 [5:0]						XX
	1	↑	1	XX			Gnn [5:0]						XX
	1	↑	1	XX			G64 [5:0]						XX
	1	↑	1	XX			B00 [5:0]						XX
	1	↑	1	XX			Bnn [5:0]						XX
	1	↑	1	XX			B31 [5:0]						XX
Memory Read	0	1	↑	XX	0	0	1	0	1	1	1	0	2Eh
	1	↑	1	XX	X	X	X	X	X	X	X	X	XX
	1	↑	1	D [17:0]									XX
Partial Area	0	1	↑	XX	0	0	1	1	0	0	0	0	30h
	1	1	↑	XX	SR [15:8]								00
	1	1	↑	XX	SR [7:0]								00
	1	1	↑	XX	ER [15:8]								01
	1	1	↑	XX	ER [7:0]								3F
Vertical Scrolling Definition	0	1	↑	XX	0	0	1	1	0	0	1	1	33h
	1	1	↑	XX	TFA [15:8]								00
	1	1	↑	XX	TFA [7:0]								00
	1	1	↑	XX	VSA [15:8]								01
	1	1	↑	XX	VSA [7:0]								40
	1	1	↑	XX	BFA [15:8]								00
	1	1	↑	XX	BFA [7:0]								00
Tearing Effect Line OFF	0	1	↑	XX	0	0	1	1	0	1	0	0	34h
Tearing Effect Line ON	0	1	↑	XX	0	0	1	1	0	1	0	1	35h
	1	1	↑	XX	X	X	X	X	X	X	X	M	00

1. TFTLCD Principle Description

-- ILI9341 instruction set(continued)

Regulative command set (continued)

Memory Access Control	0	1	↑	XX	0	0	1	1	0	1	1	0	36h
	1	1	↑	XX	MY	MX	MV	ML	BGR	MH	X	X	00
Vertical Scrolling Start Address	0	1	↑	XX	0	0	1	1	0	1	1	1	37h
	1	1	↑	XX	VSP [15:8]								00
	1	1	↑	XX	VSP [7:0]								00
Idle Mode OFF	0	1	↑	XX	0	0	1	1	1	0	0	0	38h
Idle Mode ON	0	1	↑	XX	0	0	1	1	1	0	0	1	39h
Pixel Format Set	0	1	↑	XX	0	0	1	1	1	0	1	0	3Ah
	1	1	↑	XX	X	DPI [2:0]			X	DBI [2:0]			66
Write Memory Continue	0	1	↑	XX	0	0	1	1	1	1	0	0	3Ch
	1	1	↑	D [17:0]									XX
Read Memory Continue	0	1	↑	XX	0	0	1	1	1	1	1	0	3Eh
	1	↑	1	XX	X	X	X	X	X	X	X	X	XX
	1	↑	1	D [17:0]									XX
Set Tear Scanline	0	1	↑	XX	0	1	0	0	0	1	0	0	44h
	1	1	↑	XX	X	X	X	X	X	X	X	STS [8]	00
	1	1	↑	XX	STS [7:0]								00
Get Scanline	0	1	↑	XX	0	1	0	0	0	1	0	1	45h
	1	↑	1	XX	X	X	X	X	X	X	X	X	XX
	1	↑	1	XX	X	X	X	X	X	X	GTS [9:8]		00
	1	↑	1	XX	GTS [7:0]								00
Write Display Brightness	0	1	↑	XX	0	1	0	1	0	0	0	1	51h
	1	1	↑	XX	DBV [7:0]								00

Regulative command set (continued)

Read Display Brightness	0	1	↑	XX	0	1	0	1	0	0	1	0	52h
	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	53h
	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	54h
	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	55h
	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	56h
	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	5Eh
	1	1	↑	XX	CMB [7:0]								00
Read CABC Minimum Brightness	0	1	↑	XX	0	1	0	1	0	1	1	1	5Fh
	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	DAh
	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	DBh
	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	DCh
	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 Signal Control	0	1	↑	XX	1	0	1	1	0	0	0	0	B0h
	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
	1	1	↑	XX	X	VML [6:0]							3C
VCOM Control 2	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
1st Parameter	1	1	↑	XX	RCA0 [3:0]				BCA0 [3:0]				XX
:	1	1	↑	XX	RCAx [3:0]				BCAx [3:0]				XX
16th 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
1st Parameter	1	1	↑	XX	RFA0 [3:0]				BFA0 [3:0]				XX
:	1	1	↑	XX	RFAx [3:0]				BFAx [3:0]				XX
64th 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

- This instruction is to read ID4, which is used to read the ID of the LCD controller. Therefore, the same code can perform different LCD driver initialization according to different IDs to be compatible with different LCD screens.

D3h	RDID4 (Read ID4)											
	D/CX	RDX	WRX	D17-8	D7	D6	D5	D4	D3	D2	D1	D0	HEX
Command	0	1	↑	XX	1	1	0	1	0	0	1	1	D3h
1st Parameter	1	↑	1	XX	X	X	X	X	X	X	X	X	X
2nd Parameter	1	↑	1	XX	0	0	0	0	0	0	0	0	00h
3rd Parameter	1	↑	1	XX	1	0	0	1	0	0	1	1	93h
4th Parameter	1	↑	1	XX	0	1	0	0	0	0	0	1	41h

1. TFTLCD Principle Description

-- 0x36 instruction

- This instruction is a storage access control instruction, which can control the reading and writing direction of ILI9341 memory. In short, when writing GRAM continuously, it can control the growth direction of GRAM pointer, so as to control the display mode (the same is true for reading gram).

36h	MADCTL (Memory Access Control)												
	D/CX	RDX	WRX	D17-8	D7	D6	D5	D4	D3	D2	D1	D0	HEX
Command	0	1	↑	XX	0	0	1	1	0	1	1	0	36h
Parameter	1	1	↑	XX	MY	MX	MV	ML	BGR	MH	0	0	00

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.

1. TFTLCD Principle Description
-- 0x2A instruction

- This instruction is a column address setting instruction. Under the scanning mode (default) from left to right and from top to bottom, this instruction is used to set the abscissa (x coordinate).
- The instruction has four parameters, actually two coordinate values: SC and EC, that is, the start value and end value of the column address. SC must be less than or equal to EC, and 0 ≤ SC / EC ≤ 239. Generally, when setting the X coordinate, we only need to take two parameters, that is, set SC, because if EC does not change, we only need to set it once (set when initializing ILI9341).

2Ah	CASET (Column Address Set)											
	D/CX	RDX	WRX	D17-8	D7	D6	D5	D4	D3	D2	D1	D0	HEX
Command	0	1	↑	XX	0	0	1	0	1	0	1	0	2Ah
1st Parameter	1	1	↑	XX	SC15	SC14	SC13	SC12	SC11	SC10	SC9	SC8	Note1
2nd Parameter	1	1	↑	XX	SC7	SC6	SC5	SC4	SC3	SC2	SC1	SC0	
3rd Parameter	1	1	↑	XX	EC15	EC14	EC13	EC12	EC11	EC10	EC9	EC8	Note1
4th Parameter	1	1	↑	XX	EC7	EC6	EC5	EC4	EC3	EC2	EC1	EC0	

1. TFTLCD Principle Description
-- 0x2B instruction

- This instruction is a page address setting instruction. Under the scanning mode (default) from left to right and from top to bottom, this instruction is used to set the ordinate (Y coordinate).
- The instruction has four parameters, actually two coordinate values: SP and EP, that is, the start value and end value of the page address. SP must be less than or equal to EP, and 0 ≤ SP / EP ≤ 319. Generally, when setting the Y coordinate, we only need to take two parameters, that is, set SP, because if EP does not change, we only need to set it once (when initializing ILI9341).

2Bh	PASET (Page Address Set)											
	D/CX	RDX	WRX	D17-8	D7	D6	D5	D4	D3	D2	D1	D0	HEX
Command	0	1	↑	XX	0	0	1	0	1	0	1	1	2Bh
1st Parameter	1	1	↑	XX	SP15	SP14	SP13	SP12	SP11	SP10	SP9	SP8	Note1
2nd Parameter	1	1	↑	XX	SP7	SP6	SP5	SP4	SP3	SP2	SP1	SP0	
3rd Parameter	1	1	↑	XX	EP15	EP14	EP13	EP12	EP11	EP10	EP9	EP8	Note1
4th Parameter	1	1	↑	XX	EP7	EP6	EP5	EP4	EP3	EP2	EP1	EP0	

1. TFTLCD Principle Description
-- 0x2C instruction

- This instruction is to write GRAM. After sending this instruction, we can write color data into the LCD GRAM. This instruction supports continuous writing (address is automatically incremented).
- After receiving the instruction 0x2C, the data effective bit width becomes 16 bits, we can continuously write the LCD GRAM value, and the GRAM address will increase automatically according to the scanning direction set by MY / MX / MV. For example, assuming that the scanning mode is set from left to right and from top to bottom, after setting the starting coordinates (SC and SP), the GRAM address will automatically increase by 1 (SC++) for each color value written. If it encounters EC, it will return to SC, and SP++, until EC and EP, there is no need to set the coordinates again.

2Ch	RAMWR (Memory Write)												
	D/CX	RDX	WRX	D17-8	D7	D6	D5	D4	D3	D2	D1	D0	HEX
Command	0	1	↑	XX	0	0	1	0	1	1	0	0	2Ch
1st Parameter	1	1	↑	D1 [17:0]									XX
:	1	1	↑	Dx [17:0]									XX
Nth Parameter	1	1	↑	Dn [17:0]									XX

1. TFTLCD Principle Description
-- 0x2E instruction

- This instruction is to read GRAM, which is used to read the video memory (GRAM) of ILI9341. This instruction also supports continuous reading (automatic address increment).
- After ILI9341 receives the instruction, the first time it outputs dummy data (invalid), and reads valid gram data from second time (starting from coordinates: SC, SP). The output law is: each color component accounts for 8bit, and 2 color components are output at a time. For example, the first output is R1G1, and the subsequent rule is: B1R2→G2B2→R3G3→B3R4→G4B4→R5G5... and so on.

2Eh	RAMRD (Memory Read)											
	D/CX	RDX	WRX	D17-8	D7	D6	D5	D4	D3	D2	D1	D0	HEX
Command	0	1	↑	XX	0	0	1	0	1	1	1	0	2Eh
1st Parameter	1	1	↑	XX	X	X	X	X	X	X	X	X	X
2nd Parameter	1	1	↑	D1 [17:0]									XX
:	1	1	↑	Dx [17:0]									XX
(N+1)th Parameter	1	1	↑	Dn [17:0]									XX

南方科技大学
SOUTHERN UNIVERSITY OF SCIENCE AND TECHNOLOGY

02

TFTLCD Function Description

2. TFTLCD Function Description

-- _lcd_dev structure

```
typedef struct {
        u16 width;              //LCD width
        u16 height;             //LCD height
        u16 id;                 //LCD ID
        u8  dir;                //Horizontal screen or vertical screen control:
                                   0, vertical screen; 1, horizontal screen
        u16    wramcmd;  // Start writing GRAM instruction
        u16  setxcmd;              // Set X coordinate command
        u16  setycmd;              // Set XY coordinate command
}_lcd_dev;
//LCD variable in lcd.h
extern _lcd_dev lcddev;        // important parameters to manage LCD
```

2. TFTLCD Function Description
-- Some functions(v3)

- void LCD_Init(void); // LCD initialization function
- void LCD_WR_REG(u16 regval); // Write register value function (write register command to LCD module through 8080 port)
- void LCD_WR_DATA(u16 data); // Write 16bits data function
- u16 LCD_RD_DATA(void); //read data function
- void LCD_WriteReg(u16 LCD_Reg, u16 LCD_RegValue); //Write register contents function
- u16 LCD_ReadReg(u16 LCD_Reg); // Read register contents
- void LCD_WriteRAM_Prepare(void); // Start writing GRAM
- void LCD_WriteRAM(u16 RGB_Code); //Write GRAM
- void LCD_SetCursor(u16 Xpos, u16 Ypos); // Coordinate setting
- void LCD_DrawPoint(u16 x,u16 y); //Draw point function
- u16 LCD_ReadPoint(u16 x, u16 y) ; //Read point function

2. TFTLCD Function Description
-- Some functions(v3)(continued)

- void LCD_ShowChar(u16 x, u16 y, u8 num, u8 size, u8 mode); // LCD character display function
- 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

2. TFTLCD Function Description
-- Character code table

- //PCtoLCD2002 mode setting: negative code + column by column + forward + C51 format
- //Total: 3 character sets (12*12, 16*16 and 24*24) in font. H
- //The number of bytes occupied by each character is: (size / 8 + ((size% 8)? 1:0)) * (size / 2), where size is the lattice size when the font is generated (12 / 16 / 24...)

//12*12 ASCII Character set lattice

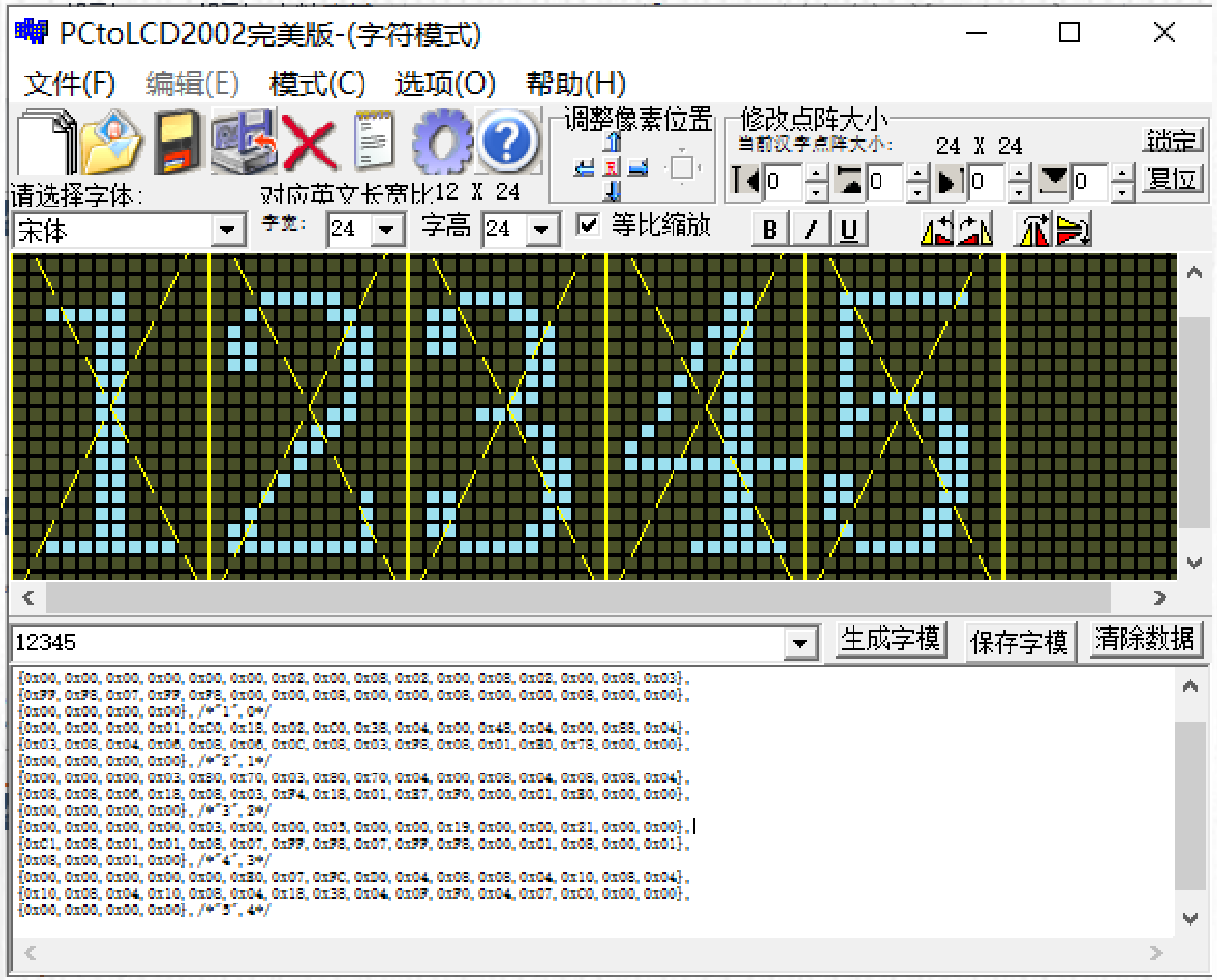
```
const unsigned char asc2_1206[95][12]={
{0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00},/*" ",0*/
{0x00,0x00,0x00,0x00,0x3F,0x40,0x00,0x00,0x00,0x00,0x00,0x00},/* "!" ,1*/     ……
{0x40,0x00,0x80,0x00,0x40,0x00,0x20,0x00,0x20,0x00,0x40,0x00},/*"~",94*/};
```

//16*16 ASCII Character set lattice

```
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 ASICII Character set lattice ......

2. TFTLCD Function Description
-- PCtoLCD2002



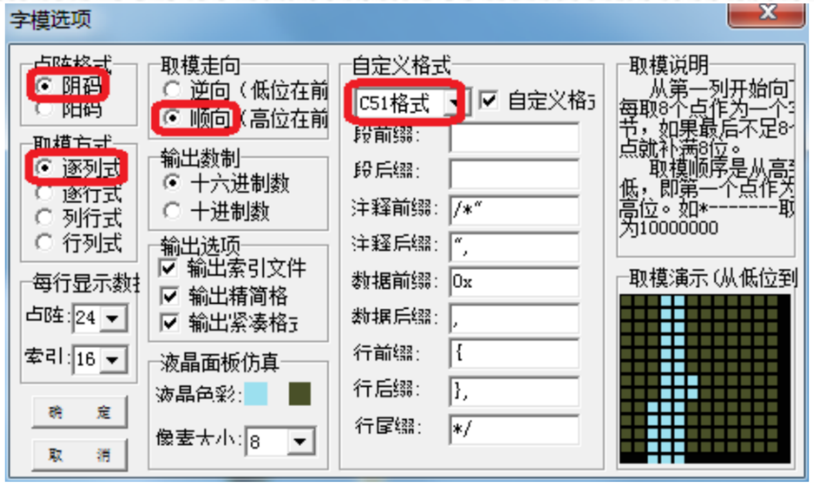
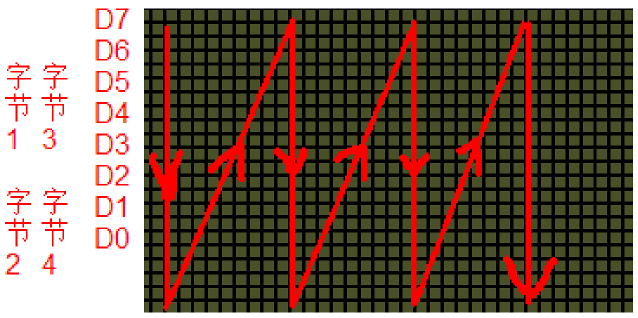


图 17.3.2 设置取模方式

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：



南方科技大学
SOUTHERN UNIVERSITY OF SCIENCE AND TECHNOLOGY

03

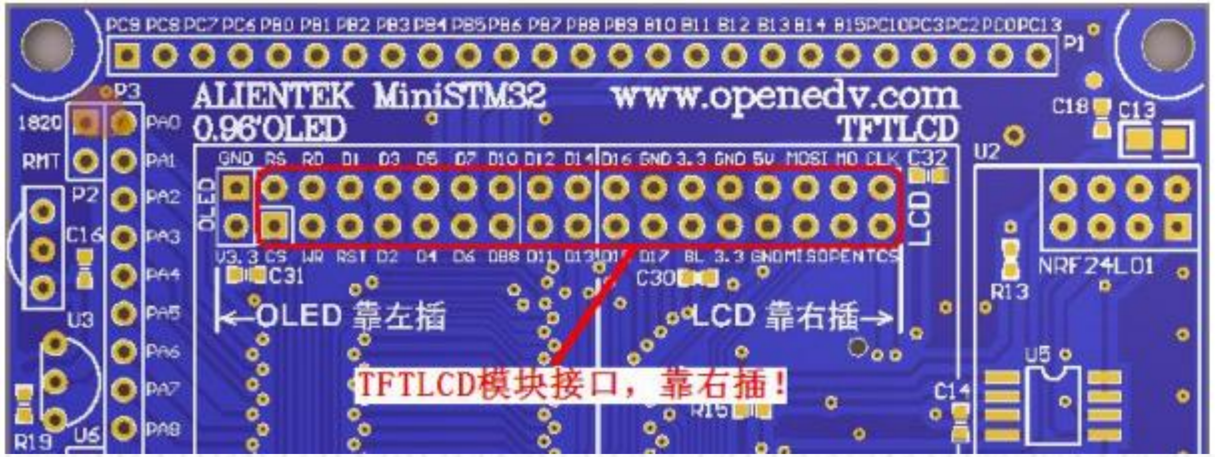
How to Program

3. How to Program

- Our Goal
  - Show charactes on TFTLCD

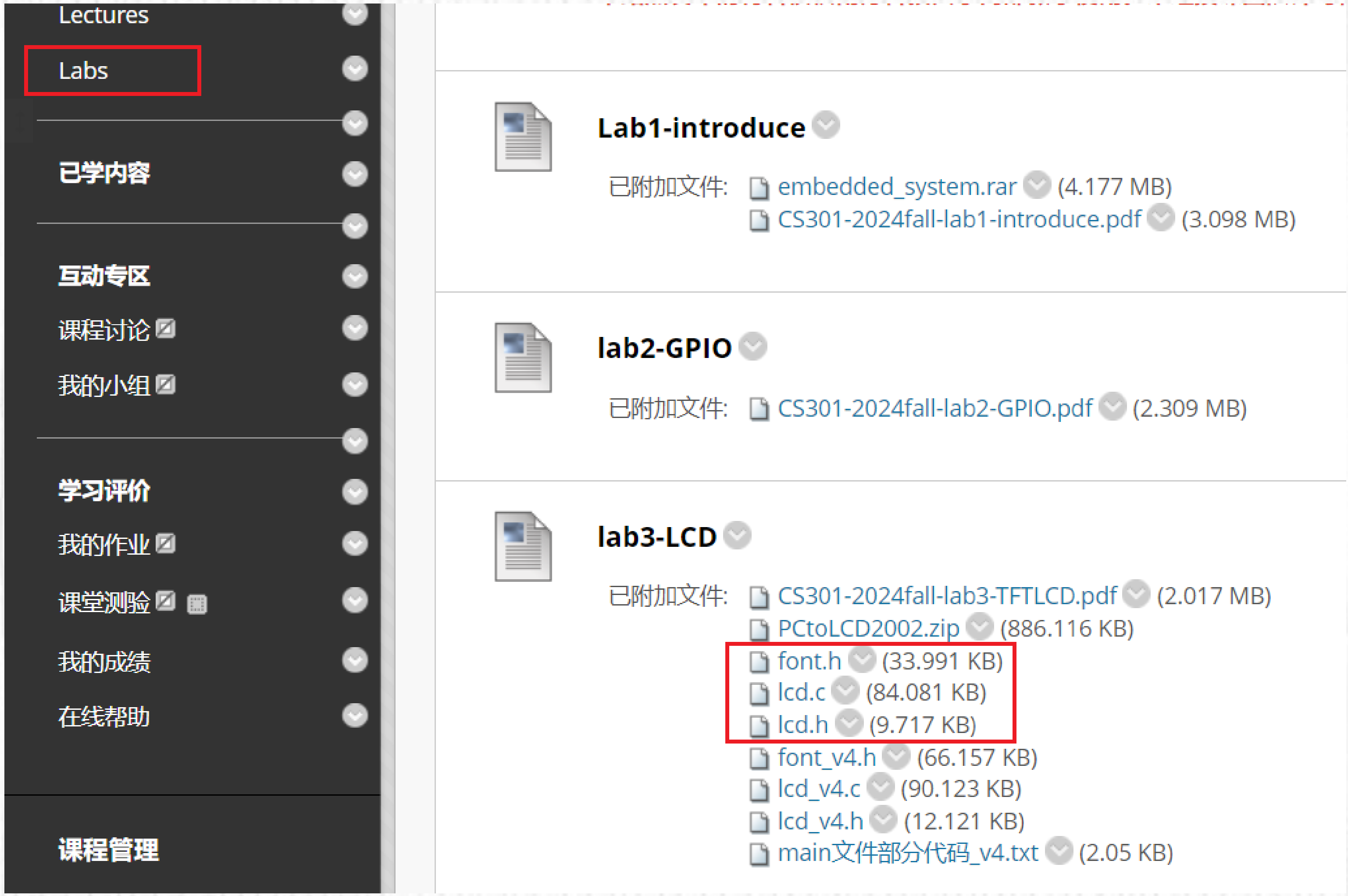
3. How to Program

- Hardware connection
  - The LCD interface of the MiniSTM32 development board backplane and the ALIENTEK TFTLCD module can be directly plugged into each other (insert on right!)
  - The two extra ports are for OLEDs
  - The corresponding relationship between TFTLCD module and GPIO port of MiniSTM32 development board is as follows:
    - LCD_LED <-> PC10
    - LCD_CS <-> PC9
    - LCD _RS <-> PC8
    - LCD _WR <-> PC7
    - LCD _RD <-> PC6
    - LCD _D[17:1] <-> PB[15:0]



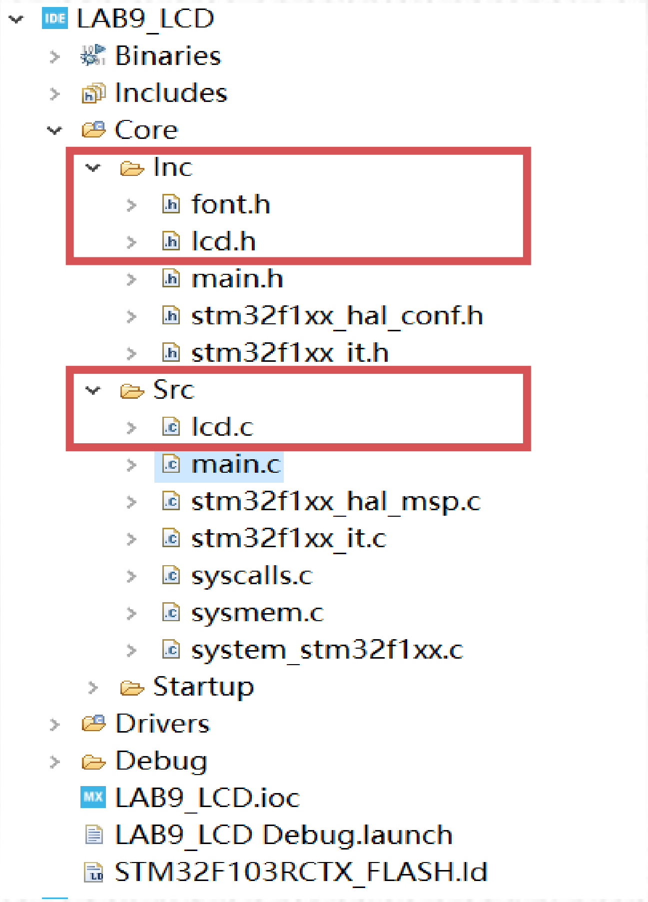
3. How to Program (v3)

- Download lcd.c, lcd.h and font.h from Blackboard



3. How to Program (v3)

- 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 (v3)

- 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
```

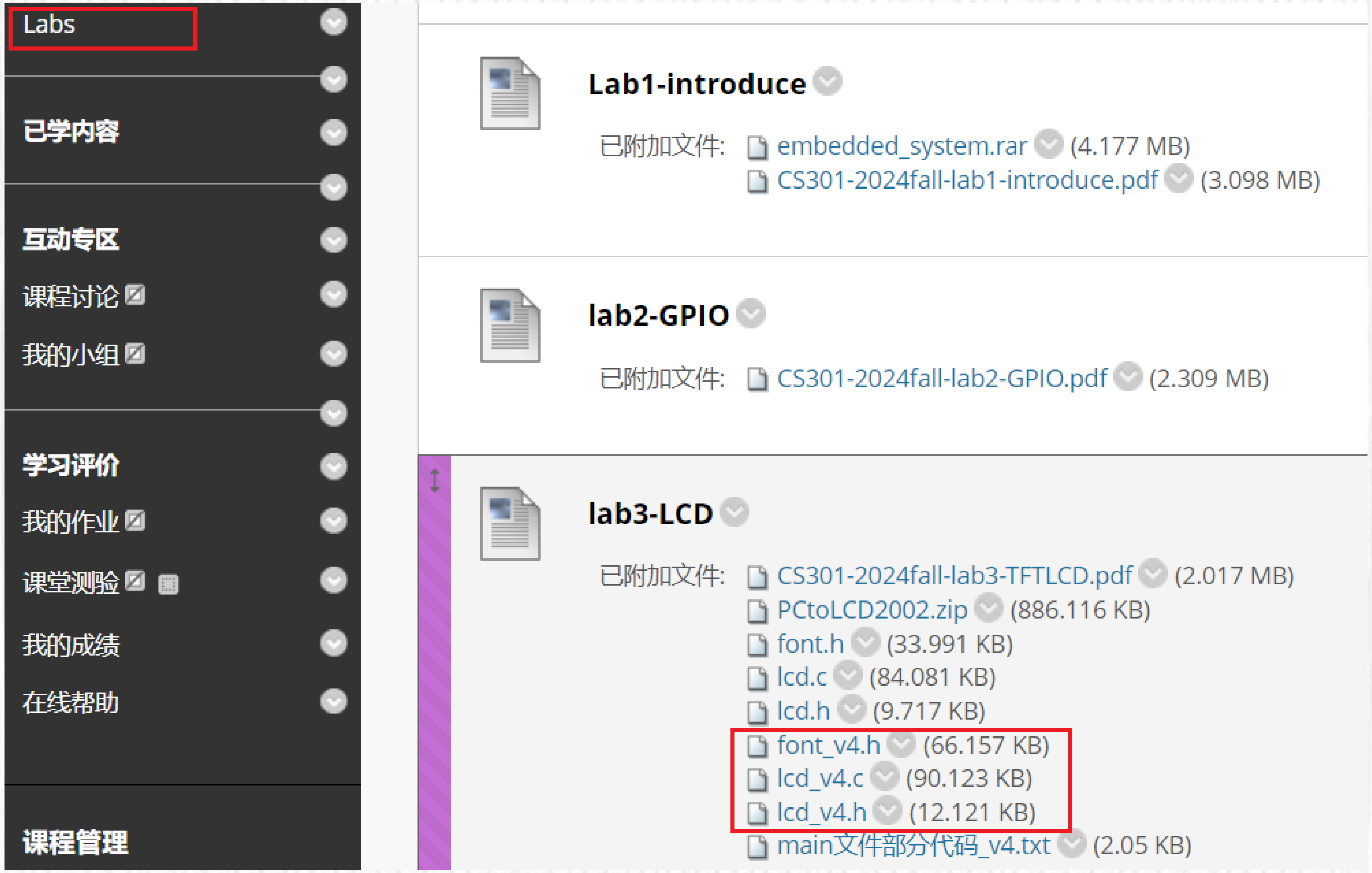
3. How to Program (v3)

- 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");
		/* Code of showing address of GPIOA->CRL (represented in hexadecimal) BEGIN */
		/* Code of showing address of GPIOA->CRL (represented in hexadecimal) END */

		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
```

3. How to Program (v4)

- Download lcd_v4.c, lcd_v4.h and font_v4.h from Blackboard



3. How to Program (v4)

- Add the following codes in main.c

```
// ......
/* USER CODE BEGIN Includes */
#include "lcd_v4.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); g_back_color = WHITE; break;
    case 1: lcd_clear(BLACK); g_back_color = BLACK; break;
    case 2: lcd_clear(BLUE); g_back_color = BLUE; break;
    case 3: lcd_clear(RED); g_back_color = RED; break;
    case 4: lcd_clear(MAGENTA); g_back_color = MAGENTA; break;
    case 5: lcd_clear(GREEN); g_back_color = GREEN; break;
    case 6: lcd_clear(CYAN); g_back_color = CYAN; break;
    case 7: lcd_clear(YELLOW); g_back_color = YELLOW; break;
    case 8: lcd_clear(BRRED); g_back_color = BRRED; break;
    case 9: lcd_clear(GRAY); g_back_color = GRAY; break;
    case 10: lcd_clear(LGRAY); g_back_color = LGRAY; break;
    case 11: lcd_clear(BROWN); g_back_color = BROWN; break;
}//end of switch
```

3. How to Program (v4)

- Add the following codes in main.c

```
        lcd_show_string(30, 40, 200, 24, 24, "Mini STM32 ^_^", RED);
        lcd_show_string(30, 70, 200, 16, 16, "TFTLCD TEST", RED);
        /* Code of showing address of GPIOA->CRL (represented in hexadecimal) BEGIN */
        /* Code of showing address of GPIOA->CRL (represented in hexadecimal) END */

        lcd_draw_rectangle(30, 150, 210, 190, BLACK);
        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

- Complete codes of the demo, show the address of GPIOA -> CRL on LCD screen.