SWM19SCBT7-50

TFT-LCD 驱动演示板
应用和注意事项
(MPU-LCM I8080 16bit 接口)


synwit
华芯微特
可靠的MCU伙伴
Design for Reliability
华芯微特MCU产品手册

目录

一、SWM19x 资源状况

synwit 华芯微特 | 可靠的MCU伙伴 Design for Reliability

SWM190系列

应用框图 Block Diagram

产品特点 Product Features

内核
- 32位ARM® Cortex™-M0 内核

时钟源
- 20MHz、48MHz 精度可达 1%的片内时钟源
- 32KHz片内时钟源片外2-32MHz 片外晶振
- 通过片上PLL，最高支持 60MHz 时钟

内置存储器
- FLASH 120KB、20KB SRAM

内置LDO
- 供电电压范围2.3V-3.6V

串行接口
- 4*UART模块
- 2*SPI模块
- 2*I²C模块

PWM控制模块
- 独立8通道 28 位 PWM 产生器，互补模式下可扩展为 16 通道

定时器模块
- 24位系统定时器
- 4 路 32 位加强定时器
- 1 路支持 HALL 接口
- 4 路 24 位基础定时器
- 32 位看门狗定时器
- 内置低功耗定时器模块

DMA模块
- 共计 4 通道，支持 UART/SPI/ADC 模块及存储模块之间数据交互

GPIO
- 最多可达51个GPIO
- 非模拟复用IO支持5V输入

模拟外设
- 2 路 12 位 8 通道，采样率高达 1MSPS高精度 SAR ADC
- 3 路模拟比较器
- 4 路运算放大器

欠压检测
- 支持欠压中断和复位选择

低功耗
- 正常模式：19mA@48MHz
- 浅睡眠：90uA

ISP/IAP
- 可自定义BOOT程序

工作温度
- -40℃-105℃

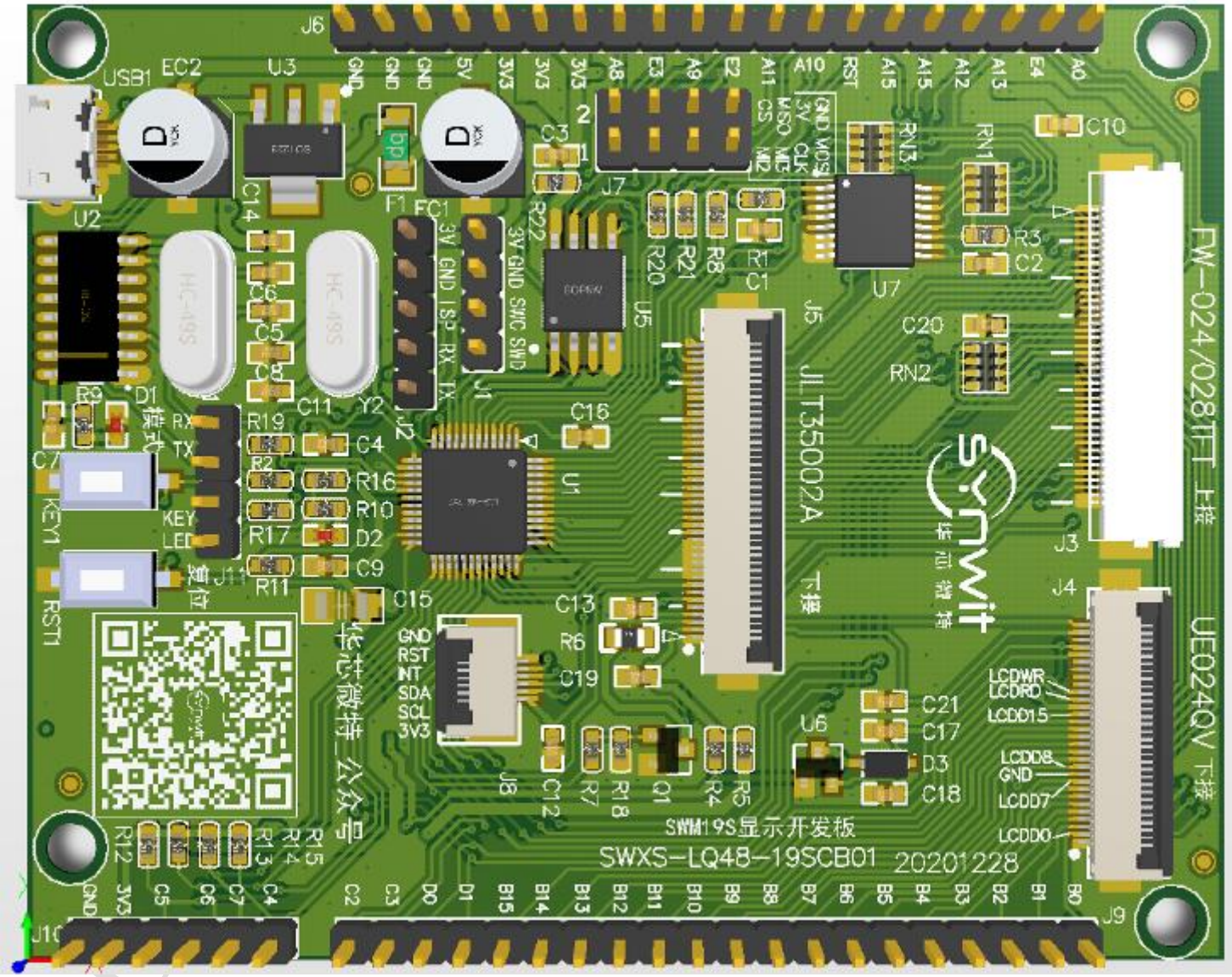
封装
- LQFP64 LQFP48 LQFP32

推荐应用
- 仪器仪表、工业控制、电机驱动、白色家电、可穿戴设备

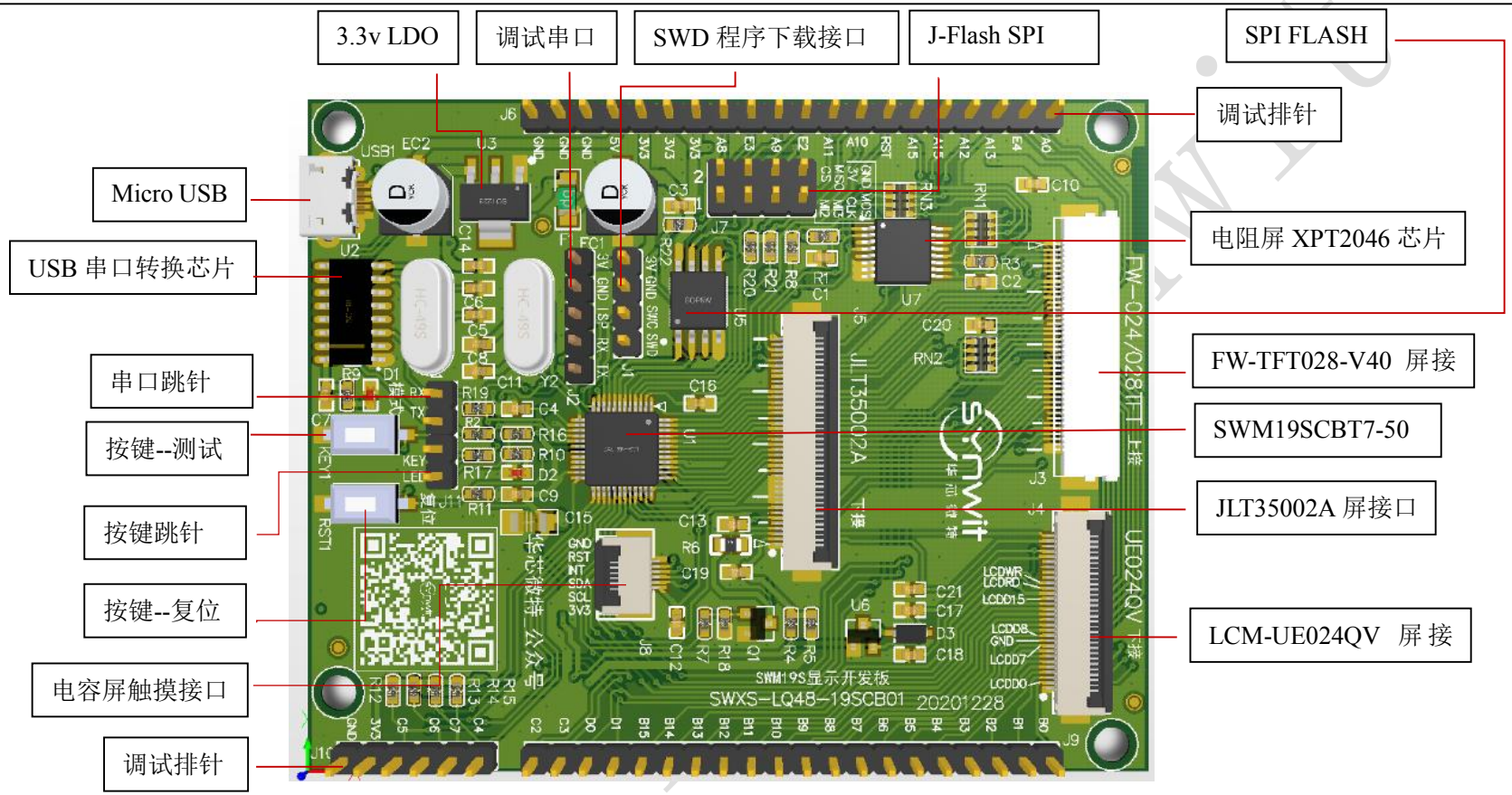
7

二、最小系统板介绍

2.1、SWM19SCBT7-50 v3.0 最小系统板板框预览图



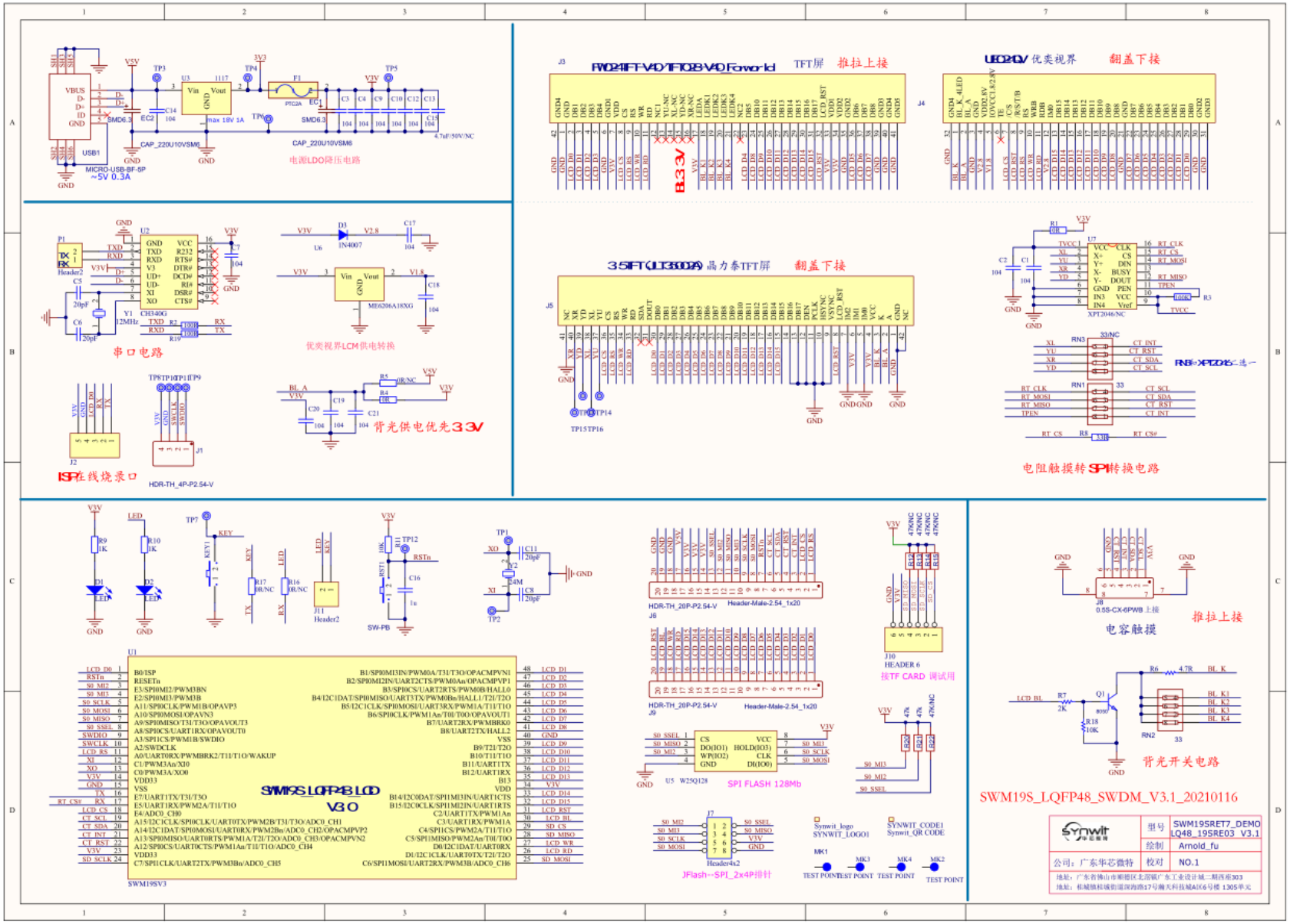
2.2、SWM19SCBT7-50 v3.0 最小系统板板接口描述和配置



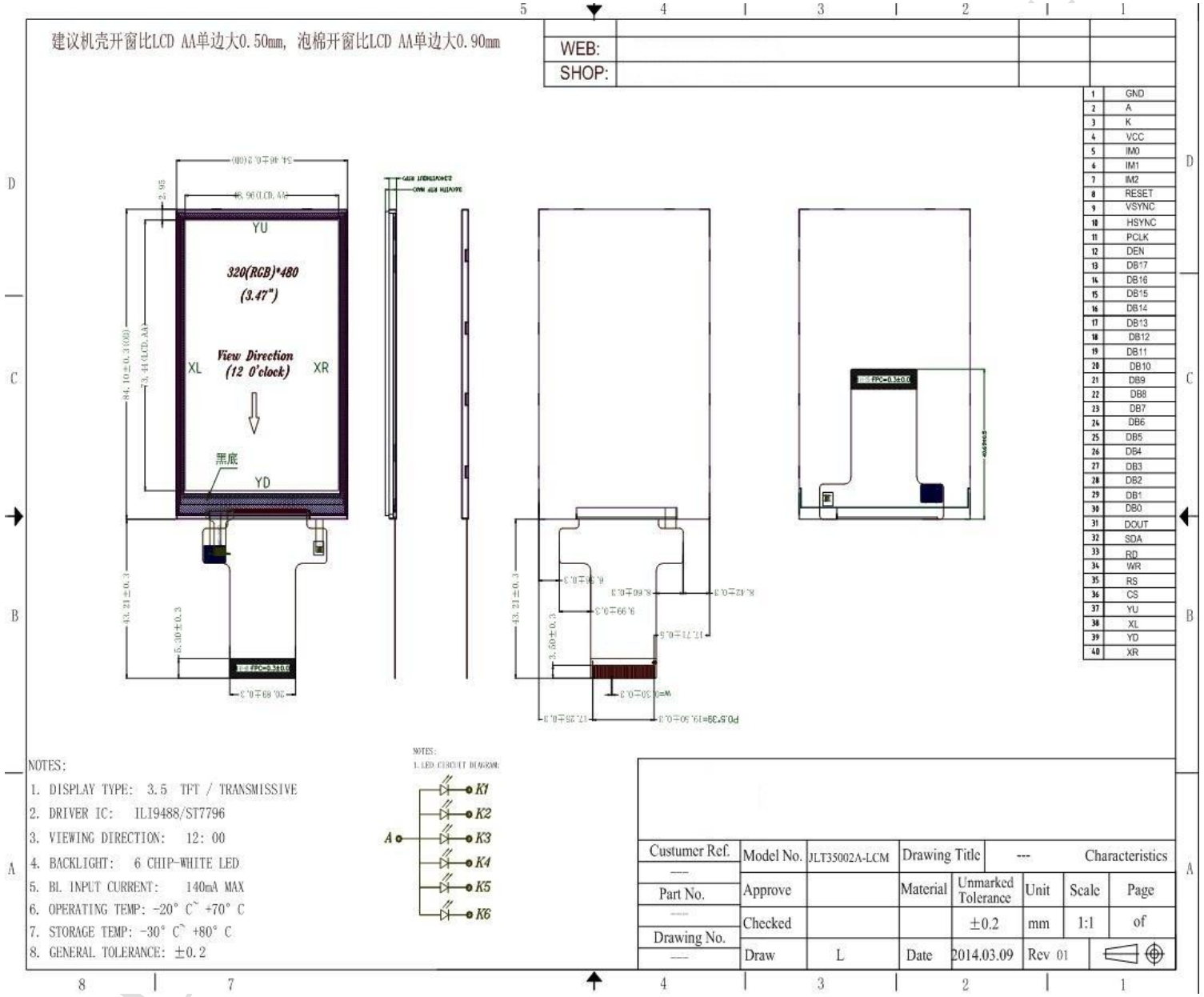
演示板硬件配置：
MCU：SWM19SCBT7-50
TFT-LCD 模组：3.5’ – JLT35002A，COG IC -- ILI9488；
2.8’ – FW-TFT028-V40，COG IC – ST7789V；
2.4’ – LCM-UE024QV-RB30-L022B，COG IC – GC9307
电容触摸驱动：NONE
SPI FLASH：W25Q32 或 W25Q128
例程对应：3.5’ ---- SWM19S_ILI9488_16bit_202101111127.rar
2.8’ ---- SWM19S_7789V_16bit_202101111127.rar
2.4’ ---- SWM19S_GC9307_16bit_202101111127.rar

接口相关说明
1、“调试排针”接口，MCU 调试使用排针，测试备用；
2、SWM19SCBT7-50 兼容了三种规格 MPU I8080 的显示 LCM，例程需要匹配好才能正常显示；
3、“J-FLASH SPI”接口，应用 SEGGER 软件组中的 J-LINK SPI 工具软件进行 SPI-FLASH 的烧写。具体描述请查看 3.6 节中的描述。
4、通过“串口跳针”将 MCU 串口功能端口与 USB 串口进行连接。
5、通过“按键跳针”将 MCU 按键功能端口与按键进行连接。

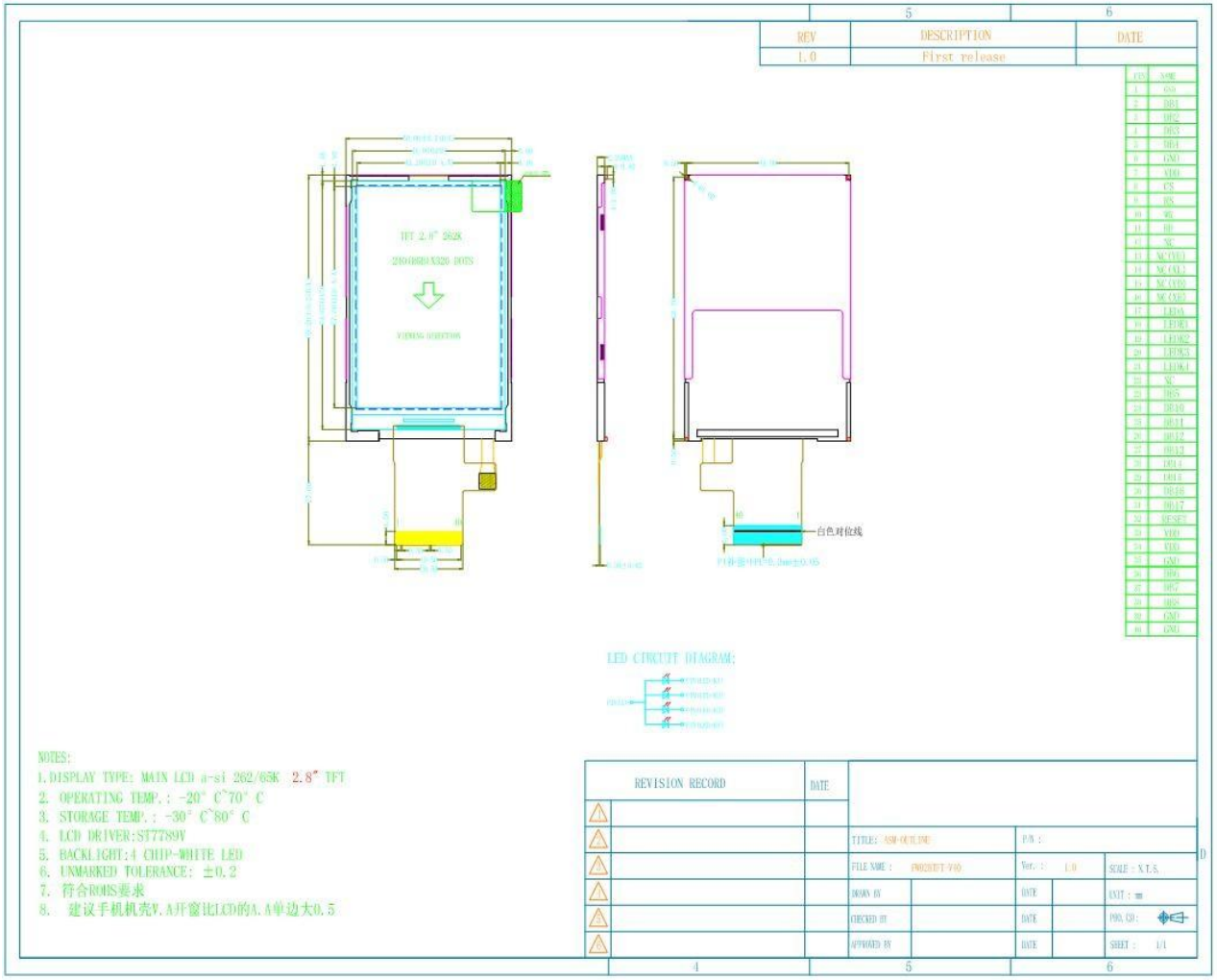
2.3、SWM19SCBT7-50 v3.0 最小系统板原理图



2.4、3.5’ – JLT35002A **模组规格**



2.5、2.8’ – FW-TFT028-V40 **模组规格**



3. Interface signals

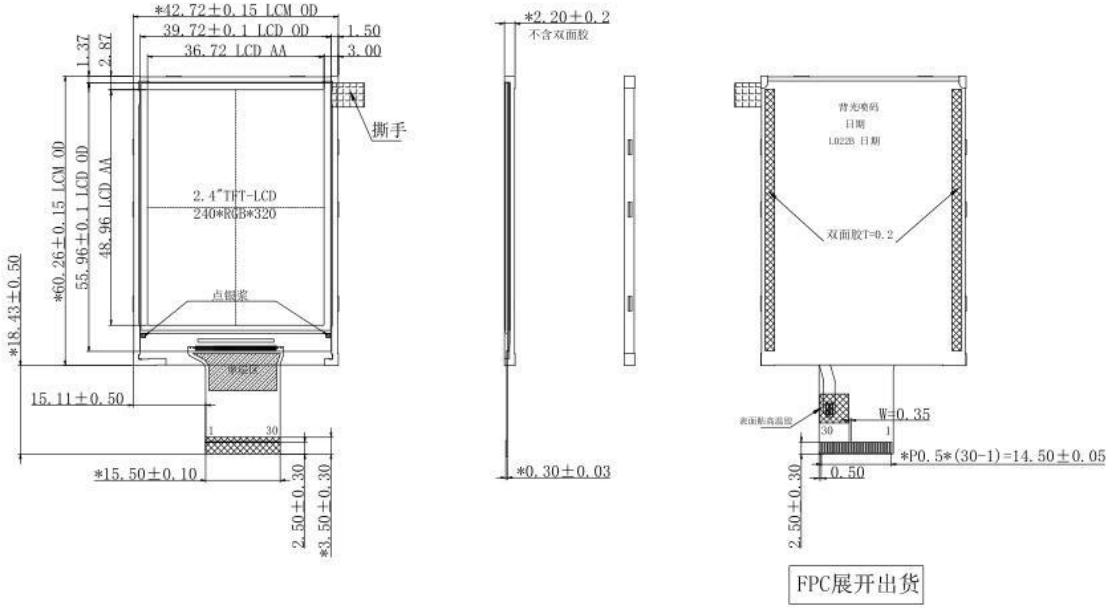
Table 2 Pin assignment

Pin No.	Symbol	Description
1	GND	GROUND
2	DB1	
3	DB2	
4	DB3	
5	DB4	
6	GND	GROUND
7	VDD	Power spply
8	CS	Chip select input pin("Low"enable).
9	RS	This pin is used to select "Data orCommand" in the parallel
10	WR	
11	RD	
12	NC	NC
13	NC	NC(YU)
14	NC	NC(XL)
15	NC	NC(YD)
16	NC	NC(XR)
17	LEDA	
18-21	LEDKI-LEDK4	NC
22	NC	
23	DB5	
24-31	DB10-DB17	
32	RESET	RESET
33	VDD	
34	VDD	
35	GND	
36	DB6	
37	DB7	
38	DB8	
39	GND	
40	GND	

2.6、2.4’ – LCM-UE024QV-RB30-L022B **模组规格**

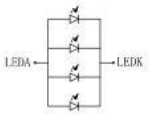
REV.	CONTENT	DESIGN	DATE
A	FIRST REALEASE		2020.08.01
B	改IC		2020.10.24

正视图 侧视图 背视图



PIN脚	名称
1	BL_K
2	BL_A
3	GND
4	VDD2.8V
5	IOVCC1.8/2.8V
6	TE
7	/CS
8	/RSTB
9	RS
10	WRB
11	RDB
12	IM0
13	DB15
14	DB14
15	DB13
16	DB12
17	DB11
18	DB10
19	DB9
20	DB8
21	GND
22	DB7
23	DB6
24	DB5
25	DB4
26	DB3
27	DB2
28	DB1
29	DB0
30	GND

NOTES:
1. DISPLAY TYPE:2.4″ 240*320 TFT LCD
   TRANSMISSVE NORMALLY BLACK
2. VIEWING DIRECTION:ALL FREE
3. Driver IC:GC9307
4. OPERATING TEMPERATURE:-20° C TO 70° C;
   STORAGE TEMPERATURE:-30° C TO 80° C,
5. BL LED: 4 CHIP-WHITE LEDS, If=80mA Vf=3.2V(TYP)
   LCM Luminance: 250cd/m² MIN,300cd/m² TYP;
   White Chromaticity:X:0.27±0.04,Y:0.31±0.04;
6. Unspecified TOLERANCE:±0.2mm;
7. RoHS2.0 Compliant;
8. With ‘*’ is the focus of inspection dimension



CUSTOMER APPROVE					
	MODUEL NO.	LCM-UE024QV-RB30-L022B	DRAWN	刘春发	
	TITLE	LCM	CHECKED	肖喜亮	UNIT MM
	SCALE 1:1	第三视角	APPROVE	钟祥林	PAGE 1/1

2.7、表 1.1、LCDC 端口分布 MPU I8080-16bit

序号	名称	SWM19SCRET7-50 LQFP48	备注
1	**LCD_D0(DB0)**	Pin1- B0	1、“B0~B15”用于 I8080-16bit 的数据端口，可以提高数据通讯的效率不建议调整。 2、“LCD_CS”\“LCD_RS”\“LCD_WR”\“LCD_RD”\“LCD_BL”\“LCD_RST” 等端口，用户根据实际产品进行调整。 建议不做调整。 3、Pin1(B0/ISP)也是 ISP 触发引脚。此引脚当在芯片上电后检测到持续 1ms 以上的高电平，将会进入 ISP 模式，此时可以通过 E5(RX)，E7(TX)进行串口方式通讯，做在线编程。
2	**LCD_D1(DB1)**	Pin48- B1	
3	**LCD_D2(DB2)**	Pin47- B2	
4	**LCD_D3(DB3)**	Pin46- B3	
5	**LCD_D4(DB4)**	Pin45- B4	
6	**LCD_D5(DB5)**	Pin44- B5	
7	**LCD_D6(DB6)**	Pin43- B6	
8	**LCD_D7(DB7)**	Pin42- B7	
9	**LCD_D8(DB8)**	Pin41- B8	
10	**LCD_D9(DB9)**	Pin39- B9	
11	**LCD_D10(DB10)**	Pin38- B10	
12	**LCD_D11(DB11)**	Pin37- B11	
13	**LCD_D12(DB12)**	Pin36- B12	
14	**LCD_D13(DB13)**	Pin35- B13	
15	**LCD_D14(DB14)**	Pin33- B14	
16	**LCD_D15(DB15)**	Pin32- B15	
17	**LCD_CS**	Pin18-E4	
18	**LCD_RS**	Pin11- A0	
19	**LCD_WR**	Pin27-D0	
20	**LCD_RD**	Pin26- D1	
21	**LCD_BL**	Pin30-C3	
22	**LCD_RST**	Pin31-C2	
	Total	22	

2.8、表 1.2、SpiFlash 端口分布

序号	名称	SWM19SCRET7-50 LQFP48	备注
1	**S0_SSEL**	Pin8-A8 /SPI0CS	S0_MI2 / S0_MI3 为 SpiFlash 四线 QSPI 方式的另外一对接口，如产品对 SpiFlash 的速度没有要求，也可以普通模式读取。此对端口可做其对应的其它功能使用。
2	**S0_SCLK**	Pin5-A11 / SPI0CLK	
3	**S0_MOSI**	Pin6-A10 / SPI0MOSI	
4	**S0_MISO**	Pin7-P9 /SPI0MISO	
5	**S0_MI2**	Pin3-E3/SPI0MI2	
6	**S0_MI3**	Pin4-E2/SPI0MI3	
	Total	6	

2.9、表 1.3、SDIO 端口分布

序号	名称	SWM19SCRET6-50 LQFP48	备注
1	**SD_CS**	Pin29- C4/SPI1CS	SD 卡的 Spi 通讯接口，此处只做硬件参考。软件例程中此功能，由于资源和速度原因，未做应用。 如无 SDIO 应用，此 4 个 IO 端口可做其对应的其它功能使用。
2	**SD_CLK**	Pin24- C7/SPI1CLK	
3	**SD_MOSI**	Pin25- C6/SPI1MOSI	
4	**SD_MISO**	Pin28- C5/SPI1MISO	
	Total	4	

2.10、表 1.4、CTP 触摸 端口分布

序号	名称	SWM19SCRET7-50 LQFP48	备注
1	**CT_SCL**	Pin19- A15 / I2C1CLK	如无触摸应用，此 4 个 IO 端口可做其对应的其它功能使用。
2	**TP_SDA**	Pin20- A14 / I2CDAT	
3	**TP_INT**	Pin21- A13/	
4	**TP_RST**	Pin22- A12 /	
	Total	4	

2.11、表 1.5、其它端口分布

序号	名称	SWM19SCRET7-50 LQFP48	备注
1	**SWDIO**	Pin9-A8/ SWDIO	可按实际项目需求进行设置。演示板作为 SWDIO 应用。 如需要设置为普通 IO 应用，设计中要考虑如何再激活为 SWD 应用作为程序的调试或下载。
2	**SWCLK**	Pin10-A3/ SWDCLK	可按实际项目需求进行设置。演示板作为 SWDCLK 应用。 如需要设置为普通 IO 应用，设计中要考虑如何再激活为 SWD 应用作为程序的调试或下载。
3	**XI**	Pin12-C1 / XI0	可按实际项目需求进行设置。演示板无应用
4	**XO**	Pin13-C0/ XO0	可按实际项目需求进行设置。演示板无应用
5	**M_KEY / TX**	Pin16-E7/ UART1TX	可按实际项目需求进行设置。演示板作为按键或串口 TX 应用
6	**RT_CS# /RX**	Pin17-E5/ UART1RX	可按实际项目需求进行设置。演示板作为按键或串口 RX 应用
	Total	6	

三、硬件设计及调试

3.1、TFT-LCD 接口要求

接口要求为 I8080-16 bit。

建议在原理图设计中，每个数据线的连接预留串接电阻 33 欧姆，作为抗干扰作用。

3.2、Demo 板上 SpiFlash 可以放置多少张 320x240 的整图

MPU I8080-16bit LCM 以 320x240 分辨率居多，以 320x240 为例。

Demo 板上的 SpiFlash 是 W25Q128，是 128M bit，即 16M Byte，16x1024 BYTE；一张 320x240 的图片 是 480x272x2=153600 Byte。 (16 x 1024 x 1024) / 153600 = 109 张整图。

3.3、ISP 烧写方式的应用

B0 端口是 ISP 方式的触发条件，在正常工作状态应该接下拉电阻，有效过滤外界的干扰导致系统上电后进入 ISP 模式。当需要 ISP 烧写是，通过外围接入高电平，系统上电过程检测 B0 端口位高电平进入 ISP 状态，配合 E5 / E7 端口 和 PC 上位机软件“SYNWIT_ISP_V3.1.2.exe”，可以进行程序的下载烧写。

3.4、IO 端口数量

如果产品只需要用到显示，根据最小系统板 IO 端口表格的描述，最多可以有 14 个 IO 端口提行其对应功能进行应用。

3.5、TFT-LCD 调试中常见问题

1)、显示白屏

调试过程中出现白屏现象，通常为 LCM COG 芯片（如 ILI9488）初始化不成功。按照例程 或 LCM 供应商提供的平初始化程序，确认复位情况、分辨率等参数符合，重点可以查看硬件上两两引脚上是否存在短路、短路的现象。

3.6、SpiFlash 中 UI 图片数据如何存储

1)、通过通用烧写器方式，进行 UI 图片数据的存储编程写入。

2)、通过 JLINK 方式。 通过 SEGGER 集成软件中“J-LINK SPI”工具软件进行存储编程写入。

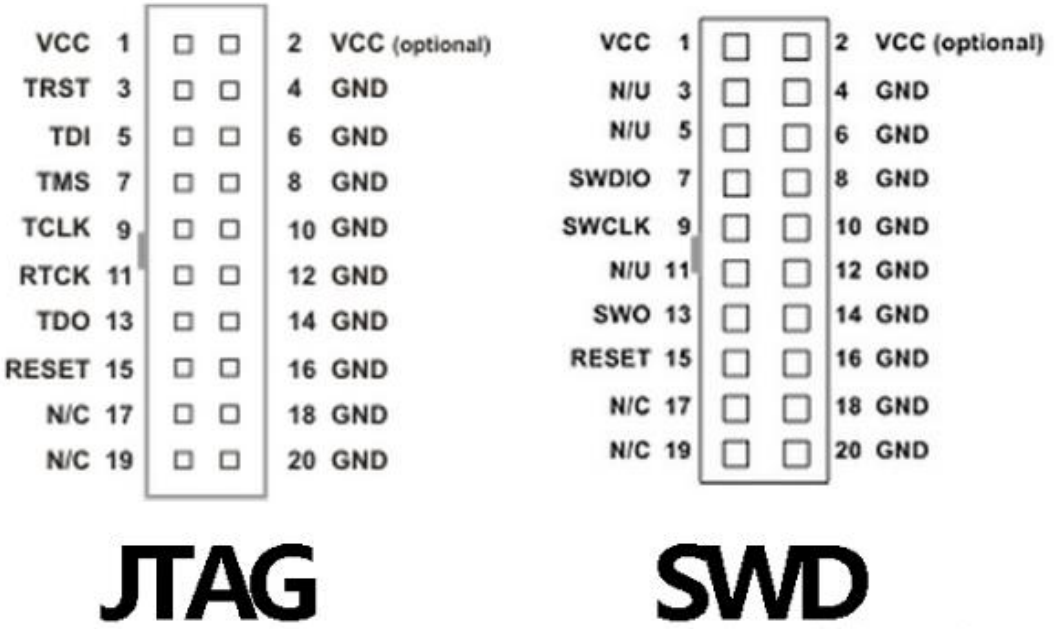
JLINK V9 或 DAP-JLINK PRO 支持 SPI 协议，可以进行 SpiFlash 的编程烧写。

硬件连接方式如下：

JLINK 的 20PIN 管脚定义如下：

Jlink接口的Jtag和SWD接口定义

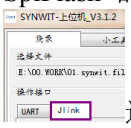
下面是标准的接口排列：



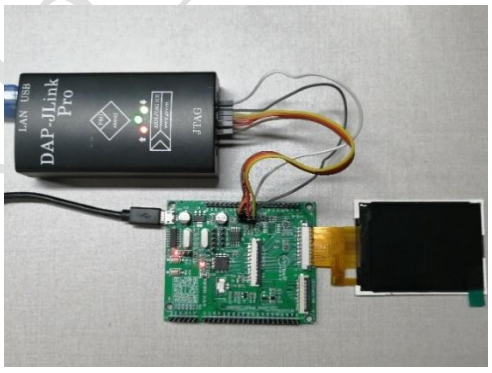
JTAG/JLINK 管脚序号和定义	SpiFlash 管脚序号和定义
5---TDI	Pin5---MOSI
7---TMS	Pin1---NSS/CS
9---TCK	Pin6---SCK
13---TDO	Pin2---MISO

除了 SPI 接口外，还要连接 JLINK 和目标板的 VCC 和 GND，否则也识别不到芯片。硬件连接完成后，就可以打开 J-Flash SPI.exe 测试烧录。

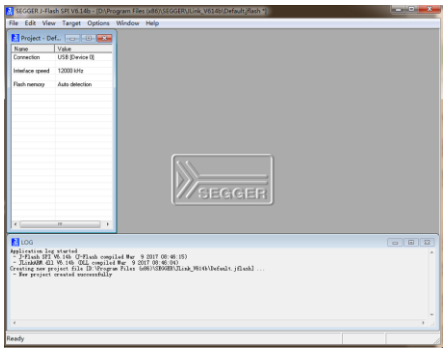
需要注意：在进行 J-Flash SPI 烧写过程中，需要停止 SWM19SCBT7-50 与 SpiFlash 的通讯，否则 J-Flash SPI 与 SpiFlash 的通讯不成功。可以先擦除整片 SWM19SCBT7-50 的程序，采用“SYNWIT-上位机_v3.1.2”软件中“Jlink”



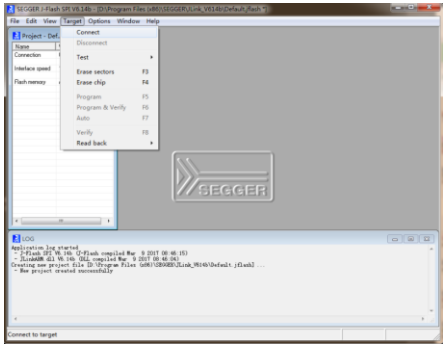
进行擦除。



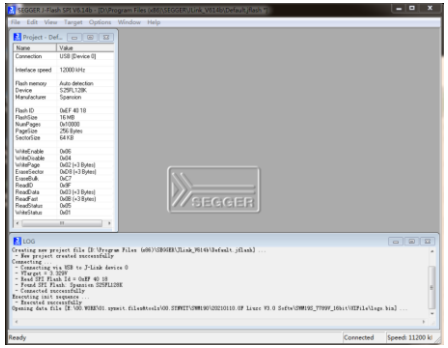
连接示意图



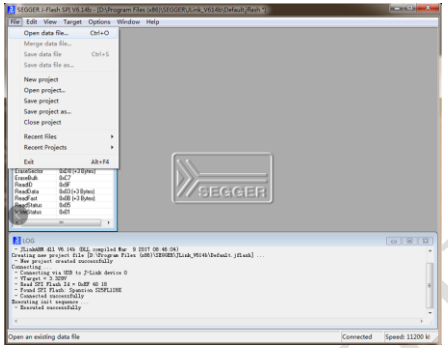
J-Flash SPI 图标



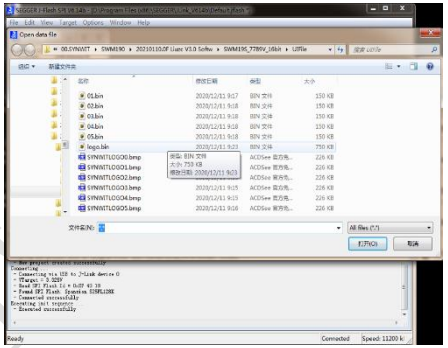
1)点击连接



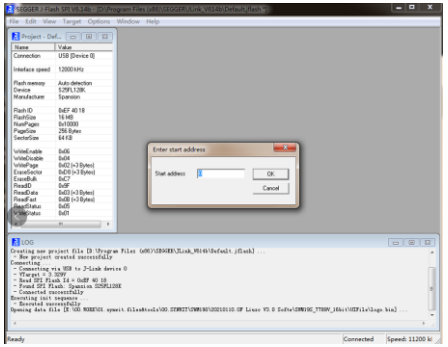
2)成功连接后显示状态



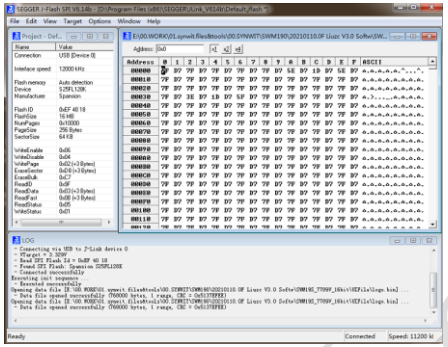
3)打开要扫写的 bin 文件



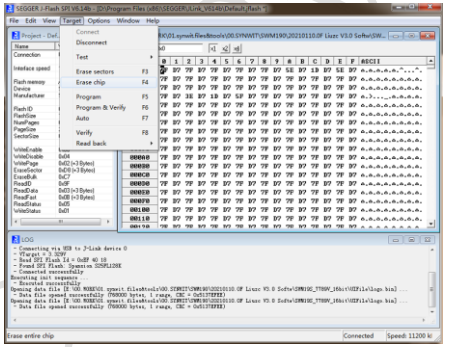
4)选择烧写 bin 文件



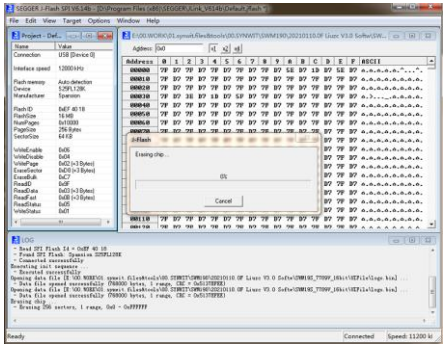
5)选择起始地址为 0 开始



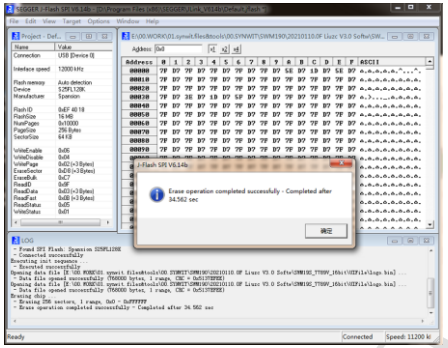
6)打开 bin 文件后的状态



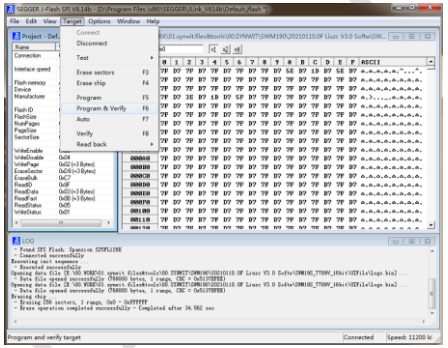
7)选择擦除整片 Flash



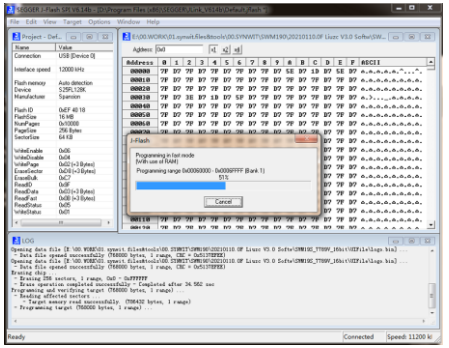
8)擦除等待过程



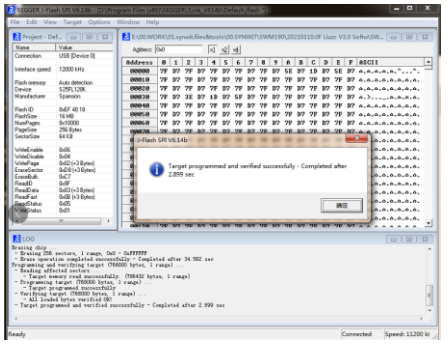
9)成功擦除后的显示



10)选择编程和校验



11)编程和校验的过程



12)数据成功烧写后的显示

四、软件设计

4.1、TFT-LCD 的初始化

所有 8080 接口的 LCD 的应用需要先通过 8080 并口写入初始化指令的写入 8080 并口通讯的每个参数需要了解，TFT-LCD 模组厂或驱动 IC 的 Application Note 提供初始化参数。

4.2、TFT-LCD 驱动的设置

8080 驱动时序可以通过 IO 口模拟，需要初始化对应的 IO 口。而且由于是 GPIO 模拟，需要提高 IO 速度和效率，所以模拟时序时用到的 IO 操作，大多数通过位带和宏定义预编译的方式，提高 IO 的速度，从而减少刷图需要的时间。

```
void TFTInit(void)
{
  uint32_t i,PINx;

//  SYS->CLKEN0 |= (0x01 << SYS_CLKEN0_GPIOB_Pos);
//  GPIOB->DIR = 0xFFFFF; //GPION.0¡«7 <=> LCD_DATA0¡«7,  GPION.16~19 <=> LCD_RD¡¢LCD_CS¡¢LCD_RS¡¢LCD_WR
//  GPIOB->ODR = 0xFFFFF;
  for(PINx=PIN0;PINx<=PIN15;PINx++)
  GPIO_Init(GPIOB,PINx,1,0,0,0);
  GPIOB->ODR = 0xFFFF;
  //GPIOB->ODR = 0xAAAA;
  PORT->PORTB_INEN = 0xFFFF;
  GPIO_Init(GPIOC,PIN3,1,0,0,0);//LCD_BL
  GPIO_SetBit(GPIOC,PIN3);
  GPIO_Init(PORT_RS,PIN_RS,1,0,0,0);
  GPIO_Init(PORT_WR,PIN_WR,1,0,0,0);
  GPIO_Init(PORT_CS,PIN_CS,1,0,0,0);
  GPIO_Init(PORT_RD,PIN_RD,1,0,0,0);
  GPIO_Init(PORT_RST,PIN_RST,1,0,0,0);
```

```
#define PORT_WR  GPIOD
#define PORT_RD  GPIOD
#define PORT_CS  GPIOE
#define PORT_RS  GPIOA
#define PORT_RST GPIOC

#define PIN_WR  PIN0
#define PIN_RD  PIN1
#define PIN_CS  PIN4
#define PIN_RS  PIN0
#define PIN_RST PIN2

#define CS_CLR  *((volatile uint32_t *) (uint32_t)&PORT_CS->DATAPIN0 + PIN_CS)=0
#define RS_CLR  *((volatile uint32_t *) (uint32_t)&PORT_RS->DATAPIN0 + PIN_RS)=0
#define RD_CLR  *((volatile uint32_t *) (uint32_t)&PORT_RD->DATAPIN0 + PIN_RD)=0
#define WR_CLR  *((volatile uint32_t *) (uint32_t)&PORT_WR->DATAPIN0 + PIN_WR)=0
#define RST_CLR *((volatile uint32_t *) (uint32_t)&PORT_RST->DATAPIN0 + PIN_RST)=0

#define CS_SET  *((volatile uint32_t *) (uint32_t)&PORT_CS->DATAPIN0 + PIN_CS)=1
#define RS_SET  *((volatile uint32_t *) (uint32_t)&PORT_RS->DATAPIN0 + PIN_RS)=1
#define RD_SET  *((volatile uint32_t *) (uint32_t)&PORT_RD->DATAPIN0 + PIN_RD)=1
#define WR_SET  *((volatile uint32_t *) (uint32_t)&PORT_WR->DATAPIN0 + PIN_WR)=1
#define RST_SET *((volatile uint32_t *) (uint32_t)&PORT_RST->DATAPIN0 + PIN_RST)=1

#define LCD_I80_WriteCmd(data_L) {CS_CLR;RS_CLR;RD_SET;WR_SET;PINs_out(GPIOB)=data_L;WR_CLR;WR_SET;RS_SET;CS_SET;}
#define LCD_I80_WriteData(data_L) {CS_CLR;RS_SET;RD_SET;WR_SET;PINs_out(GPIOB)=data_L;WR_CLR;WR_SET;RS_SET;CS_SET;}
#define LCD_I80_WriteData_16bit(data_L) {CS_CLR;RS_SET;RD_SET;WR_SET;PINs_out(GPIOB)=data_L;WR_CLR;WR_SET;RS_SET;CS_SET;}
```

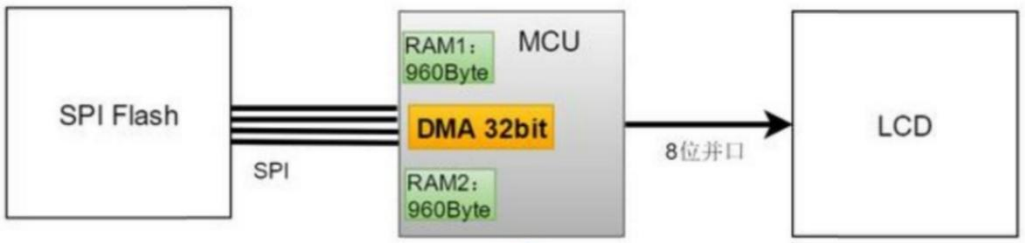
4.3、SPI_Flash 设置

需要使能 DMA 和二分频模式，加快 SPI_Flash 的数据搬运，并且 DMA 每次搬运 LCD_HOR 数量的数据，LCD_HOR 即 LCD 一行的数据量 320*2=640Byte。并且 SWM19S 不挂载文件系统 FatFS，图片文件均以二进制的 bin 文件格式线性存储于 FLASH，并根据文件大小和偏移地址来读写 SPI_Flash。

```
void SPI_Flash_Init(void)
{
  SPI_InitStructure SPI_initStruct;
  PORT_Init(PORTE, PIN2, PORTE_PIN3_SPI0_DAT2, 1); //IO2
  PORT_Init(PORTE, PIN3, PORTE_PIN2_SPI0_DAT3, 1); //IO3

  GPIO_Init(GPIOA, PIN8,  1, 0, 0,0);       // 输出，接CS

  PORT_Init(PORTA, PIN11, PORTA_PIN11_SPI0_SCLK, 0);
  PORT_Init(PORTA, PIN10, PORTA_PIN10_SPI0_MOSI, 1);
  PORT_Init(PORTA, PIN9,  PORTA_PIN9_SPI0_MISO, 1);

  SPI_initStruct.clkDiv = SPI_CLKDIV_2;
  SPI_initStruct.FrameFormat = SPI_FORMAT_SPI;
  SPI_initStruct.SampleEdge = SPI_FIRST_EDGE;
  SPI_initStruct.IdleLevel = SPI_LOW_LEVEL;
  SPI_initStruct.WordSize = 8;
  SPI_initStruct.Master = 1;
  SPI_initStruct.RXThresholdIEn = 0;
  SPI_initStruct.TXThresholdIEn = 0;
  SPI_initStruct.TXCompleteIEn  = 0;
  SPI_Init(SPI_PORT_W25X, &SPI_initStruct);
  SPI_Open(SPI_PORT_W25X);
}
void SPI_DMA_init(void)
{
    DMA_InitStructure SPI_RX_DMA_initStruct;
    uint32_t txdata = 0xFF;
  //SPI0->CTRL |= (1 << SPI_CTRL_DMATXEN_Pos);
  DMA->EN = 1;       //每个通道都有自己独立的开关控制，所以总开关可以是一直开启的

  DMA_CH_Close(DMA_CH0);     //关闭后配置

  DMA->CH[DMA_CH0].CR = (DMA_MODE_SINGLE << DMA_CR_AUTORE_Pos) |
                        (((LCD_HOR * 2)- 1) << DMA_CR_LEN_Pos);

  DMA->CH[DMA_CH0].SRC = (uint32_t)&txdata;
  DMA->CH[DMA_CH0].DST = (uint32_t)&SPI0->DATA;

  DMA->CH[DMA_CH0].AM = (0 << DMA_AM_SRCAM_Pos)   |
                        (0 << DMA_AM_DSTAM_Pos)   |
                        (DMA_UNIT_BYTE << DMA_AM_SRCBIT_Pos) |
                        (DMA_UNIT_BYTE << DMA_AM_DSTBIT_Pos);


 while(1==1)
 {

 SPI_DMA_PICREAD(Base_Addr,0,0,320,240);
 for(uint32_t i=0;i<5000000;i++);

 SPI_DMA_PICREAD(Base_Addr+IMG_SIZE,0,0,320,240);
 for(uint32_t i=0;i<5000000;i++);

 SPI_DMA_PICREAD(Base_Addr+IMG_SIZE*2,0,0,320,240);
 for(uint32_t i=0;i<5000000;i++);

 SPI_DMA_PICREAD(Base_Addr+IMG_SIZE*3,0,0,320,240);
 for(uint32_t i=0;i<5000000;i++);

 SPI_DMA_PICREAD(Base_Addr+IMG_SIZE*4,0,0,320,240);
 for(uint32_t i=0;i<5000000;i++);

 }
```

4.4、SPI_DMA 读取 Flash

调用 DMA 实现双 Buff 乒乓缓存。读取 LCD_Buf1 的时候,LCD_Buf2 用于刷图。读取 LCD_Buf2 的时候,LCD_Buf1 用于刷图。如此往复，直到读取完一张图的数据。



1. 初始化DMA控制器,数据宽度为32位，使能传输完成中断，配置DMA从Flash读第1&2行像素（960字节）保存到缓存RAM1中
2. 等传输完成后，配置DMA读下两行到RAM2，通过并口写RAM1到LCD
3. 等传输完成后，配置DMA读下两行到RAM1，通过并口写RAM2到LCD
4. 重复2和3，直到所有数据写到LCD。

```
void SPI_DMA_PICREAD(uint32_t ReadAddr,uint16_t StartX,uint16_t StartY,uint16_t Display_HOR,uint16_t Display_VER)
{
  uint16_t line = 0;
  uint16_t c=0x00;

  SPI_FLASH_LCDFastRead(ReadAddr + (Display_HOR *line*2)-1,(uint16_t*)LCD_buf_1);
  adderset(0,Display_HOR-1,0,Display_VER-1);

  while(line < LCD_VER)
  {
      switch(status_lcd)
      {
          case status_lcd1:
          {
              CS_CLR;
              RS_SET;
              RD_SET;

              for(uint16_t i=0;i<Display_HOR;i++)
              {
                PINs_out(GPIOB) = (uint16_t)LCD_buf_1[i];
                WR_CLR;
                WR_SET;
              }
              CS_SET;
              RS_SET;
       while(spi_dma_finish == 0)
           __NOP();
       spi_dma_finish = 0;

       line++;
       if(line < LCD_VER /*&& status_lcd!=status_lcdstop*/)
       {
           switch(status_lcd)
           {
               case status_lcd1:
               {
                   status_lcd = status_lcd2;
                   SPI_FLASH_LCDFastRead(ReadAddr+ (Display_HOR * line*2)-1,(uint16_t*)LCD_buf_2);
                   break;
               }
               case status_lcd2:
               {
                   status_lcd = status_lcd1;
                   SPI_FLASH_LCDFastRead(ReadAddr+ (Display_HOR * line*2)-1,(uint16_t*)LCD_buf_1);
                   break;
               }
               default:
                   break;
```

五、辅助工具的应用

5.1、转换工具为 Img2Lcd.exe

转换工具为 Img2Lcd.exe 软件，可以将 BMP 文件转换成 bin 格式。

参考上图设置❶❷❸❹❺选项，再点击“保存”，储存为转换工具默认的“*.ebm”文件格式后，可以直接更改文件后缀名为“*.bin”

5.2、UI 素材的 bin 文件的合并。

采用“SYNWIT_ISP_V3.1.2.exe”工具软件

图 2.1 启动界面

图 2.2 bin 文件合并工具

如图 2.2 所示，将需要合并 UI 素材 bin 文件打开导入“Step1-请选择文件 1”、“Step2-请选择文件 2”，同时选择合并输出的文件名称“Step3-请选择输出路径”，然后点击“合并”，生成“文件 1”和“文件 2”的“文件 3”。以此类推，将“文件 3”和“文件 4”合并生产“文件 5”。 。。。。。。

将 UI 素材文件合并成一个文件后，通过“3.6”中描述，将 Bin 文件编程存储到 SpiFlash 中。

六、如何更改其它分辨率 TFT-LCD 模组的应用

根据 LCM 供应商提供的规格书，查阅确认 COG 芯片的对应的水平和垂直显示点数的参数设置。

编写记录

日期	版本	描述	备注
2021-01-18	20210125	整理 SWM19SCBT7-50	Liuzc yinfc rk
2021-01-29	20210129	整理 J-Flash SPI 烧写的内容。更新最小系统板原理图和板框描述	rk