

# ATK-4.3'TFTLCD 用户手册

## 4.3 寸 TFTLCD 电容触摸屏模

ALIENTEK

广州市星翼电子科技有限公司

### 修订历史

版本	日期	原因
V1.0	2014/6/8	第一次发布
V1.1	2014/10/29	新增 GT9147 触摸屏支持

## 目 录

1. 特性参数.....	1
2. 使用说明.....	2
2.1 模块简介.....	2
2.2 模块引脚说明.....	3
2.3 LCD 接口时序.....	4
2.4 LCD 驱动说明.....	5
2.5 电容触摸屏接口说明.....	9
2.5.1 电容式触摸屏简介.....	9
2.5.2 OTT2001A 简介.....	10
2.5.3 GT9147 简介.....	12
3. 结构尺寸.....	13
4. 其他.....	14

# 1. 特性参数

ATK-4.3' TFTLCD-V1.2 (V1.2 是版本号, 下面均以 ATK-4.3' TFTLCD 表示该产品) 是 ALIENTEK 推出的一款高性能 4.3 寸电容触摸屏模块。该模块屏幕分辨率为 800\*480, 16 位真彩显示, 采用 NT35510 驱动, 该芯片直接自带 GRAM, 无需外加驱动器, 因而任何单片机, 都可以轻易驱动。模块采用电容触摸屏, 支持 5 点同时触摸, 具有非常好的操控效果。

ATK-4.3' TFTLCD 模块各项参数如表 1.1 和表 1.2 所示。

项目	说明
接口类型	LCD: Intel8080-16 位并口 触摸屏: IIC
颜色格式	RGB565
颜色深度	16 位
LCD 驱动器芯片	NT35510
电容触摸芯片	OTT2001A
LCD 分辨率	800*480
触摸屏类型	电容触摸
触摸点数	最多 5 点同时触摸
坐标输出速率 <sup>1</sup>	75Hz
屏幕尺寸	4.3'
工作温度	-20℃~70℃
存储温度	-30℃~80℃
外形尺寸	62.25mm*117.5mm

表 1.1 ATK-4.3' TFTLCD 基本特性

项目	说明
电源电压	背光: 5V 其他: 3.3V
IO 口电平 <sup>2</sup>	3.3V
Voh	2.64V (Min)
Vol	0.66V (Max)
Vih	2.31V (Min)
Vil	0.99V (Max)
功耗 <sup>3</sup>	30~180mA

表 1.2 ATK-4.3' TFTLCD 电气特性

注 1: 75Hz 是指单点触摸的时候, 坐标输出的速度, 多点触摸的时候, 速率会相应降低 (点数越多, 速率越慢)。

注 2: 3.3V 系统, 可以直接接本模块 (供电必须 5V&3.3V 双供电), 如果是 5V 系统, 建议串接 1K 左右电阻, 做限流处理。

注 3: 30mA 对应背光关闭时的功耗, 180mA 对应背光最亮时的功耗, 此数据是模块的总功耗 (包括 3.3V 和 5V 电源部分), 实际应用中功耗会由于电源电压的波动而略微变化。

## 2. 使用说明

### 2.1 模块简介

ATK-4.3' TFTLCD 模块是 ALIENTEK 推出的一款高性能 TFTLCD 显示模块,外观漂亮、性能优异、结构紧凑。模块通过 1 个 2\*17P 的 2.54mm 间距排针与外部连接,模块外观如图 2.1.1 所示:



图 2.1.1-1 ATK-4.3' TFTLCD 电容触摸屏模块正面图



图 2.1.1-2 ATK-4.3' TFTLCD 电容触摸屏模块背面图

从图 2.1.1 可以看出，模块丝印标注非常详细，并带有安装孔位，利于安装，可方便应用于各种产品设计。

ATK-4.3' TFTLCD 模块具有如下特点：

- 高分辨率：800\*480，显示更清晰细腻；
- 自带驱动，无需外加驱动器，单片机直接使用；
- 速度超快，理论上最高刷屏速度可达 78.9 帧/秒；
- 采用电容触摸屏，最大支持 5 点同时触摸，操控效果佳；
- 板载背光电路，只需要 3.3V&5V 供电即可，无需外加高压；
- 接口简单（LCD 采用 16 位 8080 并口，触摸屏采用 IIC 接口），使用方便；
- 采用全新元器件加工，纯铜镀金排针，坚固耐用；
- 采用国际 A 级 PCB 料，沉金工艺加工，稳定可靠；
- 人性化设计，各接口都有丝印标注，使用起来一目了然；
- PCB 尺寸为 62.25mm\*117.5mm，并带有安装孔位，安装方便；

## 2.2 模块引脚说明

ATK-4.3' TFTLCD 电容触摸屏模块通过 2\*17 的排针（2.54mm 间距）同外部连接，模块可以与 ALIENTEK 的 STM32 开发板直接对接，我们提供相应的例程，用户可以在 ALIENTEK STM32 开发板上直接测试。模块通过 34（2\*17）个引脚同外部连接，对外接口原理图如图 2.2.1 所示：

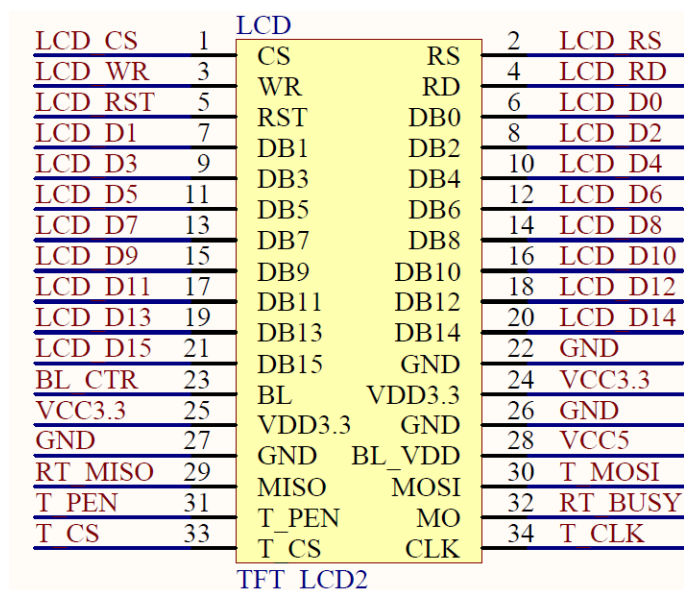


图 2.2.1 模块对外接口原理图

对应引脚功能详细描述如表 2.2.1 所示：

序号	名称	说明
1	CS	LCD 片选信号（低电平有效）
2	RS	命令/数据控制信号（0，命令；1，数据；）
3	WR	写使能信号（低电平有效）
4	RD	读使能信号（低电平有效）
5	RST	复位信号（低电平有效）
6~21	D0~D15	双向数据总线

22,26,27	GND	地线
23	BL_CTRL	背光控制引脚（高电平点亮背光，低电平关闭）
24,25	VCC3.3	主电源供电引脚（3.3V）
28	VCC5	背光供电引脚（5V）
29	MISO	NC，电容触摸屏未用到
30	MOSI	电容触摸屏 IIC_SDA 信号(CT_SDA)
31	PEN	电容触摸屏中断信号(CT_INT)
32	BUSY	NC，电容触摸屏未用到
33	CS	电容触摸屏复位信号(CT_RST)
34	CLK	电容触摸屏 IIC_SCL 信号(CT_SCL)

表 2.2.1 ATK-4.3' TFTLCD 模块引脚说明

从上表可以看出，LCD 控制器总共需要 21 个 IO 口，背光控制需要 1 个 IO 口，电容触摸屏需要 4 个 IO 口，这样整个模块需要 26 个 IO 口驱动。

## 2.3 LCD 接口时序

ATK-4.3' TFTLCD 模块采用 16 位 8080 总线接口，总线读写时序如图 2.3.1 所示：

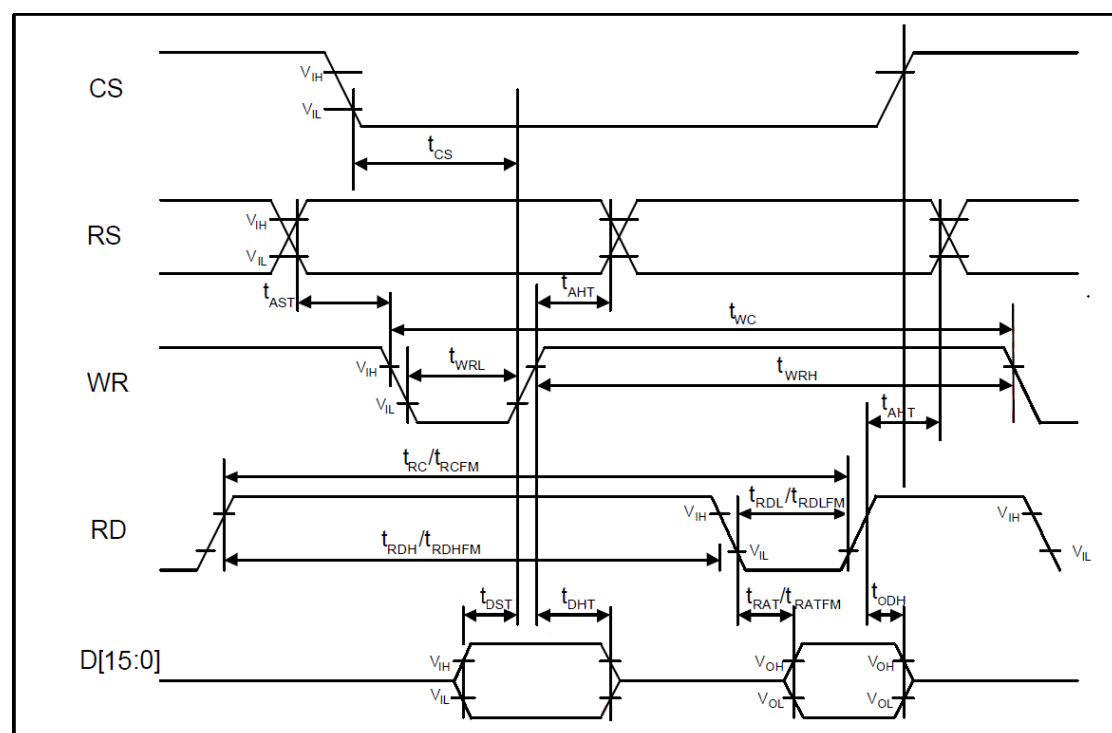


图 2.3.1 总线读写时序

图中各时间参数见表 2.3.1 所示：

Signal	Symbol	Parameter	MIN	MAX	Unit	Description
WR	t <sub>WC</sub>	Write cycle	33	-	ns	
	t <sub>WRH</sub>	Control pulse "H" duration	15	-	ns	
	t <sub>WRL</sub>	Control pulse "L" duration	15	-	ns	
RD(ID)	t <sub>RC</sub>	Read cycle (ID)	160	-	ns	When read ID data
	t <sub>RDH</sub>	Control pulse "H" duration (ID)	90	-	ns	
	t <sub>RDL</sub>	Control pulse "L" duration (ID)	45	-	ns	
RD(FM)	t <sub>RCFM</sub>	Read cycle (FM)	400	-	ns	When read from frame memory
	t <sub>RDHFM</sub>	Control pulse "H" duration (FM)	250	-	ns	
	t <sub>RDLFM</sub>	Control pulse "L" duration (FM)	150	-	ns	
RS	t <sub>AST</sub>	Address setup time (Write)	0	-	ns	
		Address setup time (Read)	10	-	ns	
	t <sub>AHT</sub>	Address hole time	2	-	ns	
D[15:0]	t <sub>DST</sub>	Data setup time	15	-	ns	
	t <sub>DHT</sub>	Data hold time	10	-	ns	
	t <sub>RAT</sub>	Read access time (ID)	-	40	ns	
	t <sub>RATFM</sub>	Read access time (FM)	-	150	ns	
	t <sub>ODH</sub>	Output disable time	5	-	ns	

表 2.3.1 16 位 8080 并口读写时间参数

从上表可以看出，模块的写周期是非常快的，只需要 33ns 即可，理论上最大速度可以达到：3030W 像素每秒，即刷屏速度可以达到每秒钟 78.9 帧。模块的读取速度相对较慢：读 ID (RD(ID)) 周期是 160ns，读显存周期是 400ns (RD(FM))。

LCD 详细的读写时序，请看 NT35510 数据手册第 28 页。

## 2.4 LCD 驱动说明

ATK-4.3' TFTLCD 模块采用 NT35510 作为 LCD 驱动器，该驱动器自带 LCD GRAM，无需外加独立驱动器，并且，在指令上，基本兼容 ILI9341，使用非常方便。模块采用 16 位 8080 并口与外部连接（不支持其他接口方式，仅支持 16 位 8080 并口），在 8080 并口模式下，LCD 驱动需要用到的信号线如下：

CS：LCD 片选信号。

WR：向 LCD 写入数据。

RD：从 LCD 读取数据。

D[15: 0]：16 位双向数据线。

RST：硬复位 LCD。

RS：命令/数据标志（0，读写命令；1，读写数据）。

除了以上信号，我们一般还需要用到这 2 个信号：RST 和 BL\_CTRL，其中 RST 是液晶的硬复位脚，低电平有效，用于复位 NT35510 芯片，实现液晶复位，在每次初始化之前，我们强烈建议大家先执行硬复位，再做初始化。BL\_CTRL 则是背光控制引脚，高电平有效，BL\_CTRL 自带了 100K 下拉电阻，所以如果这个引脚悬空，背光是不会亮的。必须接高电平，背光才会亮，另外可以用 PWM 控制 BL\_CTRL 脚，从而控制背光的亮度。

NT35510 自带 LCD GRAM (480\*864\*3 字节)，并且最高支持 24 位颜色深度（1600 万色），不过，我们一般使用 16 位颜色深度（65K 色），RGB565 格式，这样，在 16 位模式下，可以达到最快的速度。

在 16 位模式下，NT35510 采用 RGB565 格式存储颜色数据，此时 NT35510 的低 16 位数据总线（高 8 位没有用到）与 MCU 的 16 位数据线以及 24 位 LCD GRAM 的对应关系如表 2.4.1 所示：



35510总线 (16位)	D15	D14	D13	D12	D11				D10	D9	D8	D7	D6	D5			D4	D3	D2	D1	D0				
MCU数据 (16位)	D15	D14	D13	D12	D11				D10	D9	D8	D7	D6	D5			D4	D3	D2	D1	D0				
LCD GRAM (24位)	R[4]	R[3]	R[2]	R[1]	R[0]	R[4]	R[3]	R[2]	G[5]	G[4]	G[3]	G[2]	G[1]	G[0]	G[5]	G[4]	B[4]	B[3]	B[2]	B[1]	B[0]	B[4]	B[3]	B[2]	

表 2.4.1 16 位总线与 24 位 GRAM 对应关系

从上表可以看出，NT35510 的 24 位 GRAM 与 16 位 RGB565 的对应关系，其实就是分别将高位的 R、G、B 数据，搬运到低位做填充，“凑成”24 位，再显示。

MCU 的 16 位数据中，最低 5 位代表蓝色，中间 6 位为绿色，最高 5 位为红色。数值越大，表示该颜色越深。另外，特别注意 NT35510 的指令是 16 位宽，数据除了 GRAM 读写的时候是 16 位宽，其他都是 8 位宽的（高 8 位无效），这个和 ILI9320 等驱动器不一样，必须加以注意。

接下来，我们介绍一下 NT35510 的几个重要命令，因为 NT35510 的命令很多，我们这里就不全部介绍了，有兴趣的大家可以找到 NT35510 的 datasheet 看看。里面对这些命令有详细的介绍。我们将介绍：0XDA00，0XDB00，0XDC00，0X3600，0X2A00~0X2A03，0X2B00~0X2B03，0X2C00，0X2E00 等 14 条指令。

首先来看指令：0XDA00，0XDB00，0XDC00，这三条指令是读 ID1，ID2，ID3 指令，也就是用于读取 LCD 控制器的 ID，该指令如表 2.4.2 所示：

顺序	控制			各位描述									HEX
	RS	RD	WR	D15	D14	D13	D12	D11	D10	D9	D8	D7~D0	
指令 1	0	1	↑	1	1	0	1	1	0	1	0	00H	DA00H
参数 1	1	↑	1	0	0	0	0	0	0	0	0	00H	00H
指令 2	0	1	↑	1	1	0	1	1	0	1	1	00H	DB00H
参数 2	1	↑	1	0	0	0	0	0	0	0	0	80H	80H
指令 3	0	1	↑	1	1	0	1	1	1	0	0	00H	DC00H
参数 3	1	↑	1	0	0	0	0	0	0	0	0	00H	00H

表 2.4.2 读 ID 指令描述

从上表可以看出，LCD 读 ID，总共由 3 个指令（0XDA00、0XDB00 和 0XDC00）构成，每个指令输出一个参数，每个 ID 以 8 位数据（即指令后的参数）的形式输出（高 8 位固定为 0），不过这里输出的 ID，并不包含 5510 这样的字样，仅有指令 0XDB00 会输出 ID:0X80，其他两个指令读到的 ID 都是 0。将 3 个指令的输出，组合在一起，可以得到 NT35510 的 ID 为：0X8000。

通过这个 ID，即可判别所用的 LCD 驱动器是什么型号，这样，我们的代码，就可以根据控制器的型号去执行对应驱动 IC 的初始化代码，从而兼容不同驱动 IC 的屏，使得一个代码支持多款 LCD。

接下来看指令：0X3600，这是存储访问控制指令，可以控制 NT35510 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）。该指令如表 2.4.3 所示：

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	↑	36H	0	0	0	0	0	0	0	0	3600H
参数	1	1	↑	00H	MY	MX	MV	ML	BGR	MH	RSMX	RSMY	00XXH

表 2.4.3 0X3600 指令描述

从上表可以看出，0X3600 指令后面，紧跟一个参数，这里我们主要关注：MY、MX、MV 这三个位，通过这三个位的设置，我们可以控制整个 NT35510 的全部扫描方向，如表



2.4.4 所示:

控制位			效果 LCD 扫描方向 (GRAM 自增方式)
MY	MX	MV	
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

表 2.4.4 MY、MX、MV 设置与 LCD 扫描方向关系表

这样, 我们在利用 NT35510 显示内容的时候, 就有很大的灵活性了, 比如显示 BMP 图片, BMP 解码数据, 就是从图片的左下角开始, 慢慢显示到右上角, 如果设置 LCD 扫描方向为从左到右, 从下到上, 那么我们只需要设置一次坐标, 然后就不停的往 LCD 填充颜色数据即可, 这样可以大大提高显示速度。

接下来看指令: 0X2A00~0X2A03, 这几个是列地址设置指令, 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), 该指令如表 2.4.5 所示:

顺序	控制			各位描述									HEX
	RS	RD	WR	D15~D8	D7	D6	D5	D4	D3	D2	D1	D0	
指令 1	0	1	↑	2AH	0	0	0	0	0	0	0	0	2A00H
参数 1	1	1	↑	00H	SC15	SC14	SC13	SC12	SC11	SC10	SC9	SC8	SC[15:8]
指令 2	0	1	↑	2AH	0	0	0	0	0	0	0	1	2A01H
参数 2	1	1	↑	00H	SC7	SC6	SC5	SC4	SC3	SC2	SC1	SC0	SC[7:0]
指令 3	0	1	↑	2AH	0	0	0	0	0	0	1	0	2A02H
参数 3	1	1	↑	00H	EC15	EC14	EC13	EC12	EC11	EC10	EC9	EC8	EC[15:8]
指令 4	0	1	↑	2AH	0	0	0	0	0	0	1	1	2A03H
参数 4	1	1	↑	00H	EC7	EC6	EC5	EC4	EC3	EC2	EC1	EC0	EC[7:0]

表 2.4.5 0X2A00~0X2A03 指令描述

在默认扫描方式时, 这 4 个指令用于设置 x 坐标, 每条指令带有 1 个参数, 实际上总共就是 2 个坐标值: SC 和 EC (SC 和 EC 都是 16 位的, 由 2 个 8 位组成), 即列地址的起始值和结束值, SC 必须小于等于 EC, 且  $0 \leq SC/EC \leq 479$ 。一般在设置 x 坐标的时候, 我们只需要 0X2A00 和 0X2A01 两条指令即可, 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 NT35510 的时候设置), 从而提高速度。

与 0X2A00~0X2A03 指令类似, 指令: 0X2B00~0X2B03, 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)。该指令如表 2.4.6 所示:

顺序	控制			各位描述									HEX
	RS	RD	WR	D15~D8	D7	D6	D5	D4	D3	D2	D1	D0	
指令 1	0	1	↑	2BH	0	0	0	0	0	0	0	0	2B00H
参数 1	1	1	↑	00H	SP15	SP14	SP13	SP12	SP11	SP10	SP9	SP8	SP[15:8]
指令 2	0	1	↑	2BH	0	0	0	0	0	0	0	1	2B01H
参数 2	1	1	↑	00H	SP7	SP6	SP5	SP4	SP3	SP2	SP1	SP0	SP[7:0]

指令 3	0	1	↑	2BH	0	0	0	0	0	0	1	0	2B02H
参数 3	1	1	↑	00H	EP15	EP14	EP13	EP12	EP11	EP10	EP9	EP8	EP[15:8]
指令 4	0	1	↑	2BH	0	0	0	0	0	0	1	1	2B03H
参数 4	1	1	↑	00H	EP7	EP6	EP5	EP4	EP3	EP2	EP1	EP0	EP[7:0]

表 2.4.6 0X2B00~0X2B03 指令描述

在默认扫描方式时，这 4 个指令用于设置 y 坐标，每条指令带有 1 个参数，实际上总共就是 2 个坐标值：SP 和 EP（SP 和 EP 都是 16 位的，由 2 个 8 位组成），即页地址的起始值和结束值，SP 必须小于等于 EP，且  $0 \leq \text{SP/EP} \leq 799$ 。一般在设置 y 坐标的时候，我们只需要带 0X2B00 和 0X2B01 两条指令即可，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 NT35510 的时候设置），从而提高速度。

接下来看指令：0X2C00，该指令是写 GRAM 指令，在发送该指令之后，我们便可以往 LCD 的 GRAM 里面写入颜色数据了，该指令支持连续写，指令描述如表 2.4.7 所示：

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	↑	2CH	0	0	0	0	0	0	0	0	2C00H
参数 1	1	1	↑	D1[15: 0]									XX
.....	1	1	↑	D2[15: 0]									XX
参数 n	1	1	↑	Dn[15: 0]									XX

表 2.4.7 0X2C 指令描述

从上表可知，在收到指令 0X2C00 之后，数据有效位宽变为 16 位，我们可以连续写入 LCD GRAM 值，而 GRAM 的地址将根据 MY/MX/MV 设置的扫描方向进行自增。例如：假设设置的是从左到右，从上到下的扫描方式，那么设置好起始坐标（通过 SC，SP 设置）后，每写入一个颜色值，GRAM 地址将会自动自增 1（SC++），如果碰到 EC，则回到 SC，同时 SP++，一直到坐标：EC，EP 结束，其间无需再次设置的坐标，从而大大提高写入速度。

最后，来看看指令：0X2E00，该指令是读 GRAM 指令，用于读取 NT35510 的显存（GRAM），该指令在 NT35510 的数据手册上面的描述是有误的，真实的输出情况如表 2.4.8 所示：

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	↑	2EH				0	0	0	0	0	0	0	0	2E00H		
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														

表 2.4.8 0X2E00 指令描述

该指令用于读取 GRAM，如表 2.4.8 所示，NT35510 在收到该指令后，第一次输出的是 dummy 数据，也就是无效的数据，第二次开始，读取到的才是有效的 GRAM 数据（从坐标：SC，SP 开始），输出规律为：每个颜色分量占 8 个位，一次输出 2 个颜色分量。比如：第一次输出是 R1G1，随后的规律为：B1R2→G2B2→R3G3→B3R4→G4B4→R5G5... 以此类推。如果我们只需要读取一个点的颜色值，那么只需要接收到参数 3 即可，如果要连续读取（利用 GRAM 地址自增，方法同上），那么就按照上述规律去接收颜色数据。

以上，就是操作 NT35510 常用的几个指令，通过这几个指令，我们便可以很好的控制

NT35510 显示我们所要显示的内容了。

一般 TFTLCD 模块的使用流程如图 2.4.1:

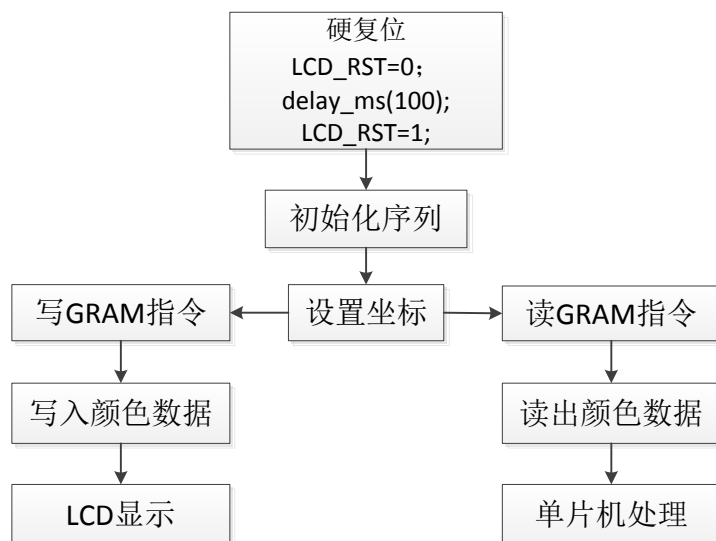


图 2.4.1 TFTLCD 使用流程

任何 LCD，使用流程都可以简单的用以上流程图表示。其中硬复位和初始化序列，只需要执行一次即可。而画点流程就是：设置坐标→写 GRAM 指令→写入颜色数据，然后在 LCD 上面，我们就可以看到对应的点显示我们写入的颜色了。读点流程为：设置坐标→读 GRAM 指令→读取颜色数据，这样就可以获取到对应点的颜色数据了。

## 2.5 电容触摸屏接口说明

### 2.5.1 电容式触摸屏简介

现在几乎所有智能手机，包括平板电脑都是采用电容屏作为触摸屏，电容屏是利用人体感应进行触点检测控制，不需要直接接触或只需要轻微接触，通过检测感应电流来定位触摸坐标。

ALIENTEK 4.3/7 寸 TFTLCD 模块自带的触摸屏采用的是电容式触摸屏，下面简单介绍下电容式触摸屏的原理。

电容式触摸屏主要分为两种：

#### 1、表面电容式电容触摸屏。

表面电容式触摸屏技术是利用 ITO(铟锡氧化物，是一种透明的导电材料)导电膜，通过电场感应方式感测屏幕表面的触摸行为进行。但是表面电容式触摸屏有一些局限性，它只能识别一个手指或者一次触摸。

#### 2、投射式电容触摸屏。

投射电容式触摸屏是传感器利用触摸屏电极发射出静电场线。一般用于投射电容传感技术的电容类型有两种：自我电容和交互电容。

自我电容又称绝对电容，是最广为采用的一种方法，自我电容通常是指扫描电极与地构成的电容。在玻璃表面有用 ITO 制成的横向与纵向的扫描电极，这些电极和地之间就构成一个电容的两极。当用手或触摸笔触摸的时候就会并联一个电容到电路中去，从而使在该条扫描线上的总体的电容量有所改变。在扫描的时候，控制 IC 依次扫描纵向和横向电极，并根据扫描前后的电容变化来确定触摸点坐标位置。笔记本电脑触摸输入板就是采用的这种方式，笔记本电脑的输入板采用 X\*Y 的传感电极阵列形成一个传感格子，当手指靠近触摸输

入板时，在手指和传感电极之间产生一个小量电荷。采用特定的运算法则处理来自行、列传感器的信号来确定手指的位置。

交互电容又叫做跨越电容，它是在玻璃表面的横向和纵向的 ITO 电极的交叉处形成电容。交互电容的扫描方式就是扫描每个交叉处的电容变化，来判定触摸点的位置。当触摸的时候就会影响到相邻电极的耦合，从而改变交叉处的电容量，交互电容的扫描方法可以侦测到每个交叉点的电容值和触摸后电容变化，因而它需要的扫描时间与自我电容的扫描方式相比要长一些，需要扫描检测  $X*Y$  根电极。目前智能手机/平板电脑等的触摸屏，都是采用交互电容技术。

ALIENTEK 所选择的电容触摸屏，也是采用的是投射式电容屏（交互电容类型），所以后面仅以投射式电容屏作为介绍。

投射式电容触摸屏采用纵横两列电极组成感应矩阵，来感应触摸。以两个交叉的电极矩阵，即：X 轴电极和 Y 轴电极，来检测每一格感应单元的电容变化，如图 2.5.1.1 所示：

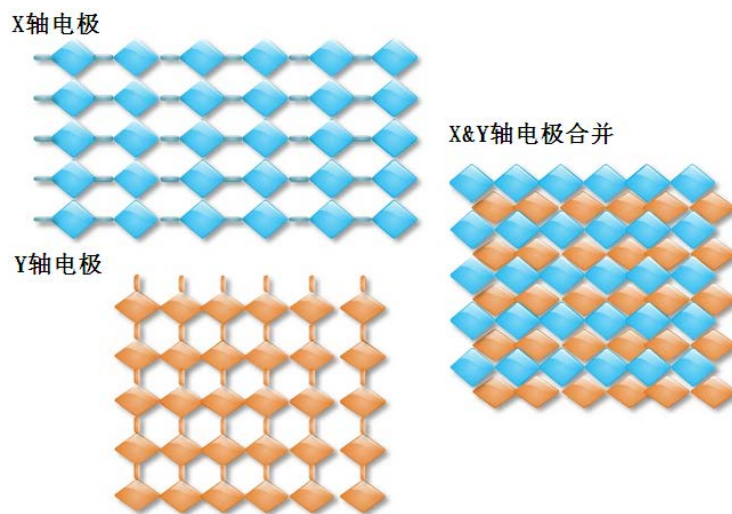


图 2.5.1.1 投射式电容屏电极矩阵示意图

示意图中的电极，实际是透明的，这里是为了方便大家理解。图中，X、Y 轴的透明电极电容屏的精度、分辨率与 X、Y 轴的通道数有关，通道数越多，精度越高。以上就是电容触摸屏的基本原理，接下来看看电容触摸屏的优缺点：

电容触摸屏的优点：手感好、无需校准、支持多点触摸、透光性好。

电容触摸屏的缺点：成本高、精度不高、抗干扰能力差。

这里提醒大家电容触摸屏对工作环境的要求是比较高的，在潮湿、多尘、高低温环境下面，都是不适合使用电容屏的。

电容触摸屏一般都需要一个驱动 IC 来检测电容触摸，且一般是通过 IIC 接口输出触摸数据的。ALIENTEK 7' TFTLCD 模块的电容触摸屏，采用的是 15\*10 的驱动结构（10 个感应通道，15 个驱动通道），采用的是 GT811 做为驱动 IC。ALIENTEK 4.3' TFTLCD 模块有两种成触摸屏：1，使用 OTT2001A 作为驱动 IC，采用 13\*8 的驱动结构（8 个感应通道，13 个驱动通道）；2，使用 GT9147 作为驱动 IC，采用 17\*10 的驱动结构（10 个感应通道，17 个驱动通道）。

这两个模块都只支持最多 5 点触摸，在本例程，仅支持 ALIENTEK 4.3 寸 TFTLCD 电容触摸屏模块，所以这里介绍仅 OTT2001A 和 GT9147，GT811 的驱动方法同这两款 IC 是类似的，大家可以参考着学习即可。

## 2.5.2 OTT2001A 简介

OTT2001A 是台湾旭曜科技生产的一颗电容触摸屏驱动 IC，最多支持 208 个通道。支

持 SPI/IIC 接口，在 ALIENTEK 4.3' TFTLCD 电容触摸屏上，OTT2001A 只用了 104 个通道，采用 IIC 接口。IIC 接口模式下，该驱动 IC 与 STM32F4 的连接仅需要 4 根线：SDA、SCL、RST 和 INT，SDA 和 SCL 是 IIC 通信用的，RST 是复位脚（低电平有效），INT 是中断输出信号，关于 IIC 我们就不详细介绍了，请参考开发板 IIC 实验。

OTT2001A 的器件地址为 0X59(不含最低位,换算成读写命令则是读:0XB3,写:0XB2)，接下来，介绍一下 OTT2001A 的几个重要的寄存器。

#### 1, 手势 ID 寄存器

手势 ID 寄存器 (00H) 用于告诉 MCU，哪些点有效，哪些点无效，从而读取对应的数据，该寄存器各位描述如表 2.5.2.1 所示：

手势 ID 寄存器 (00H)				
位	BIT8	BIT6	BIT5	BIT4
说明	保留	保留	保留	0, (X1, Y1) 无效 1, (X1, Y1) 有效
位	BIT3	BIT2	BIT1	BIT0
说明	0, (X4, Y4) 无效 1, (X4, Y4) 有效	0, (X3, Y3) 无效 1, (X3, Y3) 有效	0, (X2, Y2) 无效 1, (X2, Y2) 有效	0, (X1, Y1) 无效 1, (X1, Y1) 有效

表 2.5.2.1 手势 ID 寄存器

OTT2001A 支持最多 5 点触摸，所以表中只有 5 个位用来表示对应点坐标是否有效，其余位为保留位（读为 0），通过读取该寄存器，我们可以知道哪些点有数据，哪些点无数据，如果读到的全是 0，则说明没有任何触摸。

#### 2, 传感器控制寄存器 (ODH)

传感器控制寄存器 (ODH)，该寄存器也是 8 位，仅最高位有效，其他位都是保留，当最高位为 1 的时候，打开传感器（开始检测），当最高位设置为 0 的时候，关闭传感器（停止检测）。

#### 3, 坐标数据寄存器 (共 20 个)

坐标数据寄存器总共有 20 个，每个坐标占用 4 个寄存器，坐标寄存器与坐标的对应关系如表 2.5.2.2 所示：

寄存器编号	01H	02H	03H	04H
坐标 1	X1[15:8]	X1[7:0]	Y1[15:8]	Y1[7:0]
寄存器编号	05H	06H	07H	08H
坐标 2	X2[15:8]	X2[7:0]	Y2[15:8]	Y2[7:0]
寄存器编号	10H	11H	12H	13H
坐标 3	X3[15:8]	X3[7:0]	Y3[15:8]	Y3[7:0]
寄存器编号	14H	15H	16H	17H
坐标 4	X4[15:8]	X4[7:0]	Y4[15:8]	Y4[7:0]
寄存器编号	18H	19H	1AH	1BH
坐标 5	X5[15:8]	X5[7:0]	Y5[15:8]	Y5[7:0]

表 2.5.2.2 坐标寄存器与坐标对应表

从表中可以看出，每个坐标的值，可以通过 4 个寄存器读出，比如读取坐标 1 (X1, Y1)，我们则可以读取 01H~04H，就可以知道当前坐标 1 的具体数值了，这里我们也可以只发送寄存器 01，然后连续读取 4 个字节，也可以正常读取坐标 1，寄存器地址会自动增加，从而提高读取速度。

OTT2001A 相关寄存器的介绍就介绍到这里，更详细的资料，请参考：OTT2001A IIC 协议指导.pdf 这个文档。OTT2001A 只需要经过简单的初始化就可以正常使用了，初始化流



程:复位→延时 100ms→释放复位→设置传感器控制寄存器的最高位位 1,开启传感器检查。就可以正常使用了。

另外, OTT2001A 有两个地方需要特别注意一下:

- 1, OTT2001A 的寄存器是 8 位的, 但是发送的时候要发送 16 位 (高 8 位有效), 才可以正常使用。
- 2, OTT2001A 的输出坐标, 默认是以: X 坐标最大值是 2700, Y 坐标最大值是 1500 的分辨率输出的, 也就是输出范围为: X: 0~2700, Y: 0~1500; MCU 在读取到坐标后, 必须根据 LCD 分辨率做一个换算, 才能得到真实的 LCD 坐标。

### 2.5.3 GT9147 简介

下面我们简单介绍下 GT9147, 该芯片是深圳汇顶科技研发的一颗电容触摸屏驱动 IC, 支持 100Hz 触点扫描频率, 支持 5 点触摸, 支持 18\*10 个检测通道, 适合小于 4.5 寸的电容触摸屏使用。

和 OTT2001A 一样, GT9147 与 MCU 连接也是通过 4 根线: SDA、SCL、RST 和 INT。不过, GT9147 的 IIC 地址, 可以是 0X14 或者 0X5D, 当复位结束后的 5ms 内, 如果 INT 是高电平, 则使用 0X14 作为地址, 否则使用 0X5D 作为地址, 具体的设置过程, 请看: GT9147 数据手册.pdf 这个文档。本章我们使用 0X14 作为器件地址 (不含最低位, 换算成读写命令则是读: 0X29, 写: 0X28), 接下来, 介绍一下 GT9147 的几个重要的寄存器。

#### 1, 控制命令寄存器 (0X8040)

该寄存器可以写入不同值, 实现不同的控制, 我们一般使用 0 和 2 这两个值, 写入 2, 即可软复位 GT9147, 在硬复位之后, 一般要往该寄存器写 2, 实行软复位。然后, 写入 0, 即可正常读取坐标数据 (并且会结束软复位)。

#### 2, 配置寄存器组 (0X8047~0X8100)

这里共 186 个寄存器, 用于配置 GT9147 的各个参数, 这些配置一般由厂家提供给我们 (一个数组), 所以我们只需要将厂家给我们的配置, 写入到这些寄存器里面, 即可完成 GT9147 的配置。由于 GT9147 可以保存配置信息 (可写入内部 FLASH, 从而不需要每次上电都更新配置), 我们有点注意的地方提醒大家: 1, 0X8047 寄存器用于指示配置文件版本号, 程序写入的版本号, 必须大于等于 GT9147 本地保存的版本号, 才可以更新配置。2, 0X80FF 寄存器用于存储校验和, 使得 0X8047~0X80FF 之间所有数据之和为 0。3, 0X8100 用于控制是否将配置保存在本地, 写 0, 则不保存配置, 写 1 则保存配置。

#### 3, 产品 ID 寄存器 (0X8140~0X8143)

这里总共由 4 个寄存器组成, 用于保存产品 ID, 对于 GT9147, 这 4 个寄存器读出来就是: 9, 1, 4, 7 四个字符 (ASCII 码格式)。因此, 我们可以通过这 4 个寄存器的值, 来判断驱动 IC 的型号, 从而判断是 OTT2001A 还是 GT9147, 以便执行不同的初始化。

#### 4, 状态寄存器 (0X814E)

该寄存器各位描述如表 2.5.3.1 所示:

寄存器	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
0X814E	buffer 状态	大点	接近 有效	按键	有效触点个数			

表 2.5.3.1 状态寄存器各位描述

这里, 我们仅关心最高位和最低 4 位, 最高位用于表示 buffer 状态, 如果有数据 (坐标/按键), buffer 就会是 1, 最低 4 位用于表示有效触点的个数, 范围是: 0~5, 0, 表示没有触摸, 5 表示有 5 点触摸。这和前面 OTT2001A 的表示方法稍微有点区别, OTT2001A 是每个位表示一个触点, 这里是有效触点值就是多少。最后, 该寄存器在每次读取后, 如

果 bit7 有效，则必须写 0，清除这个位，否则不会输出下一次数据!! 这个要特别注意!!!

5，坐标数据寄存器（共 30 个）

这里共分成 5 组（5 个点），每组 6 个寄存器存储数据，以触点 1 的坐标数据寄存器组为例，如表 2.5.3.2 所示：

寄存器	bit7~0	寄存器	bit7~0
0X8150	触点 1 x 坐标低 8 位	0X8151	触点 1 x 坐标低高位
0X8152	触点 1 y 坐标低 8 位	0X8153	触点 1 y 坐标低高位
0X8154	触点 1 触摸尺寸低 8 位	0X8155	触点 1 触摸尺寸高 8 位

表 2.5.3.2 触点 1 坐标寄存器组描述

我们一般只用到触点的 x, y 坐标，所以只需要读取 0X8150~0X8153 的数据，组合即可得到触点坐标。其他 4 组分别是：0X8158、0X8160、0X8168 和 0X8170 等开头的 16 个寄存器组成，分别针对触点 2~4 的坐标。同样 GT9147 也支持寄存器地址自增，我们只需要发送寄存器组的首地址，然后连续读取即可，GT9147 会自动地址自增，从而提高读取速度。

GT9147 相关寄存器的介绍就介绍到这里，更详细的资料，请参考：GT9147 编程指南.pdf 这个文档。

GT9147 只需要经过简单的初始化就可以正常使用了，初始化流程：硬复位→延时 10ms→结束硬复位→设置 IIC 地址→延时 100ms→软复位→更新配置（需要时）→结束软复位。此时 GT9147 即可正常使用了。

然后，我们不停的查询 0X814E 寄存器，判断是否有有效触点，如果有，则读取坐标数据寄存器，得到触点坐标，特别注意，如果 0X814E 读到的值最高位为 1，就必须对该位写 0，否则无法读到下一次坐标数据。

### 3. 结构尺寸

ATK-4.3' TFTLCD 电容触摸屏模块的尺寸结构如图 3.1 所示：

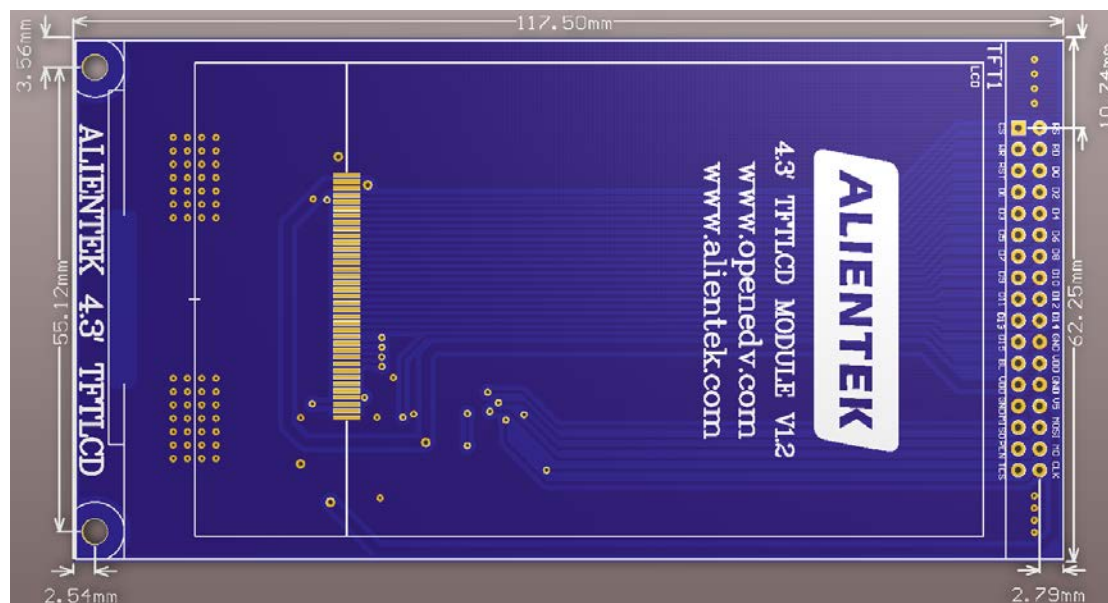


图 3.1 ATK-4.3' TFTLCD 模块尺寸图



## 4. 其他

### 1、购买地址:

官方店铺 1: <http://shop62103354.taobao.com>

官方店铺 2: <http://shop62057469.taobao.com>

### 2、资料下载

ATK-4.3' TFTLCD 模块资料下载地址: <http://www.openedv.com/posts/list/34751.htm>

### 3、技术支持

公司网址: [www.alientek.com](http://www.alientek.com)

技术论坛: [www.openedv.com](http://www.openedv.com)

传真: 020-36773971

电话: 020-38271790

