零死角玩转STM32



电容触摸屏—触摸画板

淘宝: firestm32.taobao.com

论坛: www.firebbs.cn



扫描进入淘宝店铺

主讲内容



01

触摸屏简介

02

电容触摸屏控制芯片

03

电容触摸屏—触摸画板实验

参考资料:《零死角玩转STM32》

"电容触摸屏—触摸画板"章节



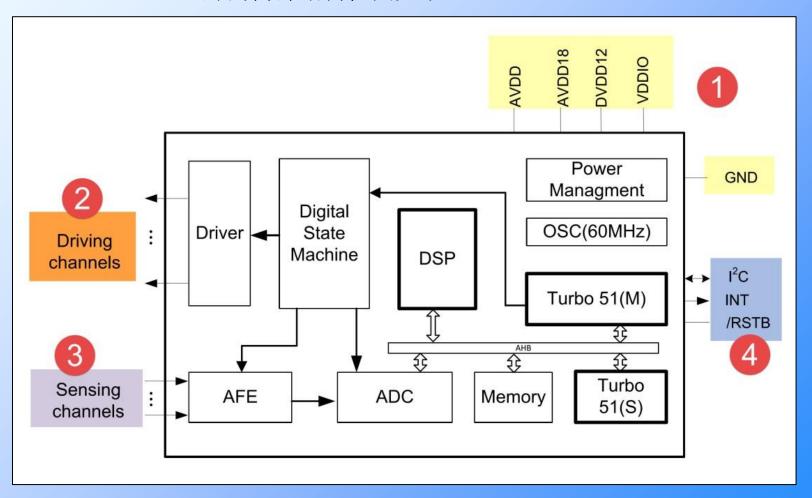
电容触摸屏控制芯片

相对来说,电容屏的坐标检测比电阻屏的要复杂,因而它也有专用芯片用于检测过程,下面我们以本章重点讲述的电容屏使用的触控芯片GT5688为例进行讲解,关于它的详细说明可从《gt91x编程指南》和《电容触控芯片GT5688》文档了解。



GT5688芯片的引脚

GT5688芯片的内部结构框图如下:





GT5688芯片的引脚

该芯片对外引出的信号线介绍如下:

信号线	说明
AVDD、AVDD18、DVDD12、VDDDIO、GND	电源和地
Driving channels	激励信号输出的引脚,一共有0-25个引脚,它连接到电容屏ITO层引出的各个激励信号轴
Sensing channels	信号检测引脚,一共有0-13个引脚,它连接到电容屏ITO层引出的各个电容量检测信号轴
I2C	I2C通信信号线,包含SCL与SDA,外部控制器通过它与GT5688芯片通讯,配置GT5688的工作方式或获取坐标信号
INT	中断信号,GT5688芯片通过它告诉外部控制器有新的触摸事件
/RSTB	复位引脚,用于复位GT5688芯片,在上电时还与INT引脚配合设置IIC通讯的设备地址



GT5688芯片的引脚

若把电容触摸屏与液晶面板分离开来,在触摸面板的背面,可看到它的边框有一些电路走线,它们就是触摸屏ITO层引出的XY轴信号线,这些信号线分别引出到GT5688芯片的Driving channels及Sensing channels引脚中。也正是因为触摸屏有这些信号线的存在,所以手机厂商追求的屏幕无边框是比较难做到的。

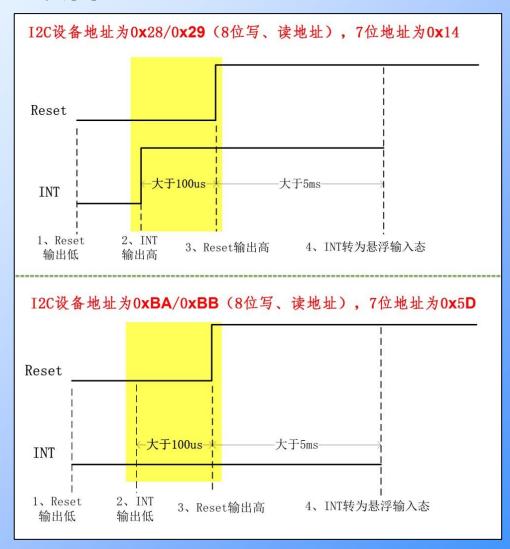


上电时序与I2C设备地址

GT5688触控芯片有两个备选的I2C通讯地址,这是由芯片的上电时序设定的,上电时序有Reset引脚和INT引脚生成,若Reset引脚从低电电平转变到高电平期间,INT引脚为高电平的时候,触控芯片使用的I2C设备地址为0x28/0x29(8位写、读地址),7位地址为0x14;若Reset引脚从低电电平转变到高电平期间,INT引脚一直为低电平,则触控芯片使用的I2C设备地址为0xBA/0xBB(8位写、读地址),7位地址为0x5D。



上电时序与I2C设备地址





寄存器配置

上电复位后,GT5688芯片需要通过外部主控芯片加载寄存器配置,设定它的工作模式,这些配置通过I2C信号线传输到GT5688,它的配置寄存器地址都由两个字节来表示,这些寄存器的地址从0x8047-0x8100,一般来说,我们实际配置的时候会按照GT5688生产厂商给的默认配置来控制芯片,仅修改部分关键寄存器。



寄存器配置

寄存器	Config Data	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	
0x8047	Config_ Version	配置文件的版本号(新下发的配置版本号大于原版本,或等于原版本号但配置内容有变化时保存,版本号版本正常范围: 'A'~'Z',发送 0x00 则将版本号初始化为'A')								
0x8048	X Output Max (Low Byte) X Output Max	x 坐标输出最大值								
0x8049	(High Byte)									
0x804A	Y Output Max (Low Byte)	Y 坐标输出最大值								
0x804B	Y Output Max (High Byte)									
0x804C	Touch Number	Reserved				输出触点个数上限: 1~10				
0x804D	Module_ Switch1	Stylus_pi (预知	riority E义)	Stret	ch_rank	X2Y (X,Y 坐标交 换)	Sito (软件 降噪)	INT 触 00: 上升 01: 下降 02: 低年 03: 高年	十沿触发 锋沿触发 电平查询	

0x804E-0x80FE寄存器省略

0x80FF	Config_Chksum	配置信息校验(0x8047 到 0x80FE 之字节和的补码)
0x8100	Config_Fresh	配置已更新标记(由主控写入标记)



寄存器配置

• 配置版本寄存器

Ox8047配置版本寄存器,它包含有配置文件的版本号,若新写入的版本号比原版本大,或者版本号相等,但配置不一样时,才会更新配置文件到寄存器中。其中配置文件是指记录了寄存器Ox8048-0x80FE控制参数的一系列数据。

为了保证每次都更新配置,我们一般把配置版本寄存器设置为 "0x00",这样版本号会默认初始化为'A',这样每次我们修改其它寄存器 配置的时候,都会写入到GT5688中。



寄存器配置

· X、Y分辨率

0x8048-0x804B寄存器用于配置触控芯片输出的XY坐标的最大值, 为了方便使用,我们把它配置得跟液晶面板的分辨率一致,这样就能使触控芯 片输出的坐标一一对应到液晶面板的每一个像素点了。 触点个数

0x804C触点个数寄存器用于配置它最多可输出多少个同时按下的触点坐标,这个极限值跟触摸屏面板有关,如我们本章实验使用的触摸面板最多支持5点触控。



寄存器配置

• 模式切换

0x804D模式切换寄存器中的X2Y位可以用于交换XY坐标轴;而INT 触发方式位可以配置不同的触发方式,当有触摸信号时,INT引脚会根据这里的配置给出触发信号。

配置校验

0x80FF配置校验寄存器用于写入前面0x8047-0x80FE寄存器控制参数字节之和的补码,GT9157收到前面的寄存器配置时,会利用这个数据进行校验,若不匹配,就不会更新寄存器配置。

• 配置更新

0x8100配置更新寄存器用于控制GT9157进行更新,传输了前面的寄存器配置并校验通过后,对这个寄存器写1,GT9157会更新配置。

Addr	Access	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
0x8140	R	Product ID (Lowest Byte,ASCII 码)							
0x8141	R		Product ID (Third Byte,ASCII 码)						
0x8142	R	1							
0x8143	R		Produ	ct ID (Highest By	te, ASCII	马)			
0x8144	R		Firmwa	re version (16进	制数 LowBy	yte)			
0x8145	R		Firmwa	re version (16进	制数 HighB	yte)			
0x8146	R		х со	ordinate resolution	n (low byte)			
0x8147	R	2	x cod	ordinate resolution	(high byte)			
0x8148	R	6	у со	ordinate resolution	n (low byte)			
0x8149	R		у сос	ordinate resolution	n (high byte)			
0x814A	R		Ver	ndor_id(当前模组	且选项信息)			
0x814B	R			Reserved					
0x814C	R	3	Reserved						
0x814D	R	0	Reserved						
0x814E	R/W	buffer status	large detect	Proximity Valid	HaveKey	numb	er of t	ouch p	oints
0x814F	R		track id						
0x8150	R		рс	int 1 x coordinate	(low byte)				
0x8151	R		point 1 x coordinate (high byte)						
0x8152	R	4	point 1 y coordinate (low byte)						
0x8153	R	point 1 y coordinate (high byte)							
0x8154	R	Point 1 size (low byte)							
0x8155	R	point 1 size (high byte)							
0x8156	R	Reserved							
0x8157	R	track id							
0x8158	R	point 2 x coordinate (low byte)							
0x8159	R	point 2 x coordinate (high byte)							
0x815A	R	point 2 y coordinate (low byte)							
0x815B	R	point 2 y coordinate (high byte)							
0x815C	R	point 2 size (low byte)							
0x815D	R	point 2 size (high byte)							
0x815E	R	Reserved							







读取坐标信息

上述寄存器主要是由外部主控芯片给**GT9157**写入配置的,而它使用 下面的寄存器向主控器反馈触摸信息。

· 产品ID及版本

0x8140-0x8143 寄存器存储的是产品ID,上电后我们可以利用I2C读取这些寄存器的值来判断I2C是否正常通讯,这些寄存器中包含有"9157"字样;而0x8144-0x8145则保存有固件版本号,不同版本可能不同。

• X/Y分辨率

0x8146-0x8149寄存器存储了控制触摸屏的分辨率,它们的值与我们前面在配置寄存器写入的XY控制参数一致。所以我们可以通过读取这两个寄存器的值来确认配置参数是否正确写入。



读取坐标信息

• 状态寄存器

0x814E地址的是状态寄存器,它的Buffer status位存储了坐标状态,当它为1时,表示新的坐标数据已准备好,可以读取,0表示未就绪,数据无效,外部控制器读取完坐标后,须对这个寄存器位写0。number of touch points位表示当前有多少个触点。其余数据位我们不关心。

• 坐标数据

从地址0x814F-0x8156的是触摸点1的坐标数据,从0x8157-0x815E的是触摸点2的坐标数据,依次还有存储3-10触摸点坐标数据的寄存器。读取这些坐标信息时,我们通过它们的track id来区分笔迹,多次读取坐标数据时,同一个track id号里的数据属于同一个连续的笔划轨迹。



读坐标流程

上电、配置完寄存器后, GT5688就会开监测触摸屏, 若我们前面的配置使INT采用中断上升沿报告触摸信号的方式,整个读取坐标信息的过程如下:

- 待机时INT引脚输出低电平;
- 有坐标更新时, INT引脚输出上升沿;
- INT输出上升沿后,INT 脚会保持高直到下一个周期(该周期可由配置 Refresh_Rate 决定)。外部主控器在检测到INT的信号后,先读取状态 寄存器(0x814E)中的number of touch points位获当前有多少个触摸点,然后读取各个点的坐标数据,读取完后将 buffer status位写为 0。外部 主控器的这些读取过程要在一周期内完成,该周期由0x8056地址的 Refresh_Rate寄存器配置;



读坐标流程

- 上一步骤中INT输出上升沿后,若主控未在一个周期内读走坐标,下次 GT5688 即使检测到坐标更新会再输出一个 INT 脉冲但不更新坐标;
- 若外部主控一直未读走坐标,则 GT5688 会一直输出 INT 脉冲。

零死角玩转STM32





论坛: www.firebbs.cn

淘宝: firestm32.taobao.com



扫描进入淘宝店铺