

# GT5668 编程指南文件

## 目 录

1. 接口说明.....	2
2. I <sup>2</sup> C 通信时序.....	3
2.1 主机对 GT5668 进行写操作时序.....	3
2.2 主机对 GT5668 进行读操作时序.....	3
3. 寄存器列表.....	4
3.1 实时命令.....	4
3.2 配置信息 (R/W) .....	5
3.3 坐标信息.....	19
3.4 手势信息.....	22
3.5 GT5668 的命令状态寄存器.....	23
3.6 HotKnot 的状态寄存器.....	23
3.7 HotKnot 的发送缓冲区.....	24
3.8 HotKnot 的接收缓冲区.....	25
4. 上电初始化与寄存器动态修改.....	26
4.1 GT5668 上电时序.....	26
4.2 上电或复位 I <sup>2</sup> C 地址选择.....	27
4.3 上电发送配置信息.....	28
4.4 主控响应 “INT Request”.....	28
4.5 寄存器动态修改.....	28
5. 坐标读取.....	29
6. 工作模式切换.....	30
6.1 Normal Mode.....	30
6.2 Green Mode.....	30
6.3 Gesture Mode.....	31
6.4 Sleep Mode.....	31
6.5 Approach Mode.....	32
6.6 Receive Mode.....	32
6.7 Send Mode.....	33
7. Gesture 模式驱动修改.....	34
7.1 灭屏后进入 Gesture 模式.....	34
7.2 灭屏后进入 Sleep 模式.....	34
7.3 按电源键（或 home 键）开屏.....	34
8. Gesture 模式坐标读取.....	35
9. 版本修订记录.....	36

## 1. 接口说明

GT5668 与主机接口共有 6 PIN，分别为：VDD、GND、SCL、SDA、INT、RESET。

主控的 INT 口线需具有上升沿或下降沿中断触发功能，并且当其在输入态时，主控端必需设为悬浮态，取消内部上、下拉功能；主机通过输出高、低来控制 GT5668 的 RESET 口为高或低。为保证可靠复位，建议 RESET 脚输出低 100us 以上。

GT5668 与主机通信采用标准 I2C 通信，最高速率可以支持至 400K bps。当主机采用 200K 以上的通信速率时，需要特别注意 I2C 口的外部上拉电阻阻值，以保证 SCL、SDA 边沿足够陡峭。GT5668 在通信中始终作为从设备，其 I2C 设备地址由 7 位设备地址加 1 位读写控制位组成，高 7 位为地址，bit 0 为读写控制位。GT5668 有两个从设备地址可供选择，如下表：

7 位地址	8 位写地址	8 位读地址
0x5D	0xBA	0xBB
0x14	0x28	0x29

每次上电或复位时需要使用 INT 脚进行 I2C 地址设置，方法请参考“上电或复位 I2C 地址选择”一章。

## 2. I<sup>2</sup>C 通信时序

### 2.1 主机对 GT5668 进行写操作时序



S: 起始信号。

Address\_W: 带写控制位的从设备地址。

ACK: 应答信号。

Register\_H、Register\_L: 待写入的 16 位寄存器首地址。

Data\_1 至 Data\_n: 数据字节 1—n。

E: 停止信号。

设定了写操作寄存器首地址后，可以只写 1 字节数据，也可以一次性写入多个字节数据，GT5668 自动将其往高地址顺序存储。

### 2.2 主机对 GT5668 进行读操作时序

先通过前述写操作时序设定需要读取的寄存器首地址，重新发送起始信号进行读寻址，读取寄存器数据。



Address\_R: 带读控制位的从设备地址。

NACK: 最后 1 字节读完主控回 NACK。

设定了读操作寄存器地址后，主控可以一次读取 1 字节，也可以一次性读取多个字节数据，GT5668 自动递增寄存器地址，将后续数据顺序发送。

设定完读操作寄存器地址后的停止信号（上图中的第一个 E 信号）可发可不发，但是重新开始 I<sup>2</sup>C 通信的起始信号必须再次发送。

### 3. 寄存器列表

#### 3.1 实时命令

Addr	Name	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
0x8040	Command	0x00: 读坐标状态 0x03: 基准更新（内部测试） 0x05: 关屏 0x07: 退出充电模式 0x0b: 手模式(不支持弱信号) 0x12: 开启 Proximity 功能 0x20: 进入从机接近检测模式 0x22: 进入数据传输模式 0x28: 退出从机接近检测模式 0x31: 保存自定义手势模版 0x37: 删除某个手势模版 0xAA: ESD 保护机制使用，由驱动定时写 AA 并定时读取检查 0x01, 0x02: 差值原始值 0x04: 基准校准（内部测试） 0x06: 进入充电模式 0x08: 进入手势唤醒模式 0x0c: 自动模式（自动切换手和手套） 0x13: 关闭 Proximity 功能 0x21: 进入主机接近检测模式 0x29: 退出主机接近检测模式 0x35: 清空触控 IC 中保存的手势模版信息 0x39: 查询手势模版信息							
0x8041	Command_Data	与命令对应的数据（不需要下发数据的命令数据区下发 0）							
0x8042	Command_Checksum	命令与数据的累加和校验（ $\text{sum}(0x8040 \sim 0x8042) == 0$ ）							
0x8043	ESD_Check	ESD 保护机制使用，在初始化时清零，之后由驱动进行读写操作							
0x8044	Request	FW 主动发送给主控的请求							
0x8045	FW_Status_L	程序状态字							
0x8046	FW_Status_H								

## 3.2 配置信息（R/W）

Addr	Config Data	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
0x8050	Config_Version	bit7 为是否固化标记（0：普通，1:固化），bit0~bit6 为对应的版本号							
0x8051	X Output Max (Low Byte)	X 坐标输出最大值							
0x8052	X Output Max (High Byte)								
0x8053	Y Output Max (Low Byte)	Y 坐标输出最大值							
0x8054	Y Output Max (High Byte)								
0x8055	Touch Number	Reserved				输出触点个数上限：1~10			
0x8056	Module_Switch1	Water_SpeedLimit_En	Water_LargeRestraining_En	Stretch_Rank 00: 0.4P 01: 0.4P 10: 0.4P 11: 自定义系数		X2Y	Sito	INT 触发方式	
0x8057	Module_Switch2	STP_SE: 不同图案异常特殊处理		FirstFilter_Dis: 首次去抖加大使能	Water_SITO	Water_Proof_Disable	SelfCap_ResistCommon 自容共模使能开关	SelfCap_WaterProof 自容防水使能开关	Touch_Key 0:无按键 1:有按键
0x8058	Module_Switch3	Key_Restrain_Dis: 按键与屏体抑制使能开关	Force_DownEdgeRelease 0:兼容原来 1:开启下降沿更新	INT_Wakeup 唤醒电平 0:高电平唤醒 1:低电平唤醒	Check_Screen_Neg 0: 关全屏负值更新; 1: 开全屏负值更新;	Water_Single_Disks 0:单指进防水功能 1:单指关闭进防水功能	Water_Shape_En: 水状态下形变处理开关	Y_Invert: H 形黄光出线方式 0: 下出线; 1: 上出线;	Shape_En 形变处理开关 0: Dis 1: En
0x8059	Module_Switch4	Monitor_En: 临时配置是否进行基准监控	Drop_Water_En:	Driver_Resersal: 驱动逆序	Sensor_Resersal: 感应逆序	Force_Self_Rect_Filter:	LowPower_Dis:关闭进入低功耗功能	LargeCoo_r_En:大面积上报坐	Expand_En: 跟随性使能开关

0x805A	Module_Switch5	Reserved			RC_TYPE: 00:RC 默认参数 01:RC 配置参数 10:RC 扫描参数		Edge_Res_AllScreen_En: n:边缘全屏抑制开关	Edge_MultiRes_En: 边缘抑制多点使能开关	Reserved
0x805B	Noise_Reduction	H 形黄光图案削底系数 N 一般情况只有 H 型才能配				噪声消除值（系数为 1，0~15 有效）			
0x805C	Screen_Touch_Level	屏上触摸点从无到有的阈值							
0x805D	Screen_Leave_Level	屏上触摸点从有到无的阈值							
0x805E	Sys_Control	Refresh_Rate 坐标上报率（周期为 5+N ms）				Low_Power_Control 进低功耗时间（0~15s）			
0x805F	Shake_Count	手指松开去抖次数（不能超过按下去抖次数）				手指按下去抖次数			
0x8060	X_Threshold	X 坐标输出门限：0~255（以 1 个最终坐标点为单位，配置为 0 则一直输出坐标）							
0x8061	Y_Threshold	Y 坐标输出门限：0~255（以 1 个最终坐标点为单位，配置为 0 则一直输出坐标）							
0x8062	Space	上边框的空白区（以 32 个原始坐标点为系数）				下边框的空白区（以 32 个原始坐标点为系数）			
0x8063		左边框的空白区（以 32 个原始坐标点为系数）				右边框的空白区（以个原始坐标点 32 为系数）			
0x8064	Stretch_RM	各区间基数							
0x8065	Stretch_R0	区间 1 系数							
0x8066	Stretch_R1	区间 2 系数							
0x8067	Stretch_R2	区间 3 系数							
0x8068	Filter	First_Filter: 手指按下首次移动时使用此滤波参数		Normal_Filter（原始坐标窗口滤波值，系数为 4）					
0x8069	Mini_Filter	Reserved	Reserved	去抖值为 First_Filter	FirstFilter_Middle:非边缘（最外一个通道）的 FirstFilter	X 和 Y 方向划线过程中的小 Filter 设置，当一个方向坐标变化较大，另一个方向坐标变化较小，配置此值可防止变化较小的方向的坐标抖动			
0x806A	Combine_Dis	充电器模式下，大面积梯度阈值上限的增加值为该值*5 建议配置 7				合框距离，0~15 可配，合点距离为配置值的 2 倍平方跟 pitch。为了兼容老配置，配 0 默认与之前处理一样，合点距离为 2 pitch。			



0x806B	Split_Set	大面积框拆点距离设置, 0~15 可配, 拆点距离为配置值的 2 倍平方根 pitch。为了兼容老配置, 配 0 默认与之前处理一样, 大面积框拆点距离为 12 平方根 pitch。		正常触摸拆点距离设置, 0~15 可配, 拆点距离为配置值的 2 倍平方根 pitch。为了兼容老配置, 配 0 默认与之前处理一样, 正常触摸拆点距离为 7 平方根 pitch。	
0x806C	WaterFrameTime	水状态下, 更新备份数据帧的时间, 以 1 个主循环的周期为单位计时, 配置小于 8 默认为 32。为了应对慢滑时差分值不够, 导致慢滑无输出, 建议该值不要小于 8			
0x806D	WaterUpdateTime	水状态下, 没有手指时进行快速更新的时间, 以 1 个主循环的周期为单位计时, 配置小于 10 默认为 100。防止更新太快, 手指缓慢靠近时无输出, 建议该值不要小于 10			
0x806E	S_FeedBack	S 型改善负反馈量 (1/256 为单位)			
0x806F	Screen_Neg_Thres	全屏负值检测统计阈值 0~15: 配置值为 N*5; 当差值小于 (-N*5) 时, 统计负值个数 +1; 其中当配置 0 时为-Leavelevel/2。		全屏负值检测更新阈值 0~15: 配置值为 N*10; 若全屏小于阈值 (高四位) 的负差值个数+N*10 ≥ g_CellTotal/2 , 则更新基准	
0x8070	Shape_Control_Val	差值大于此配置值时进入新形变重按和超重按处理阈值 进入重按处理阈值 = 配置值 *Touchlevel 进入超重按处理阈值 = (配置值 +1)*Touchlevel 按压差值大于此值时进入新形变处理, 配置小于 3 时关闭		Touch 判量的倍数, 配置越大, 消减越多, 建议配置 1~4 配置 0 则进入重按时不处理, 进入超重按时默认为 2 倍	
0x8071	ExpandFactor	ExpandFactor 拉伸系数 0~15: 配置值为 N; 值越小则拉伸越多, 跟随性越好; 值越大则拉伸少, 滞后大; 该配置会引起画线超前的现象		Smooth_Weight 平滑权重 0~15: 配置值为 N; 值越大则越平滑, 但滞后越多; 值越小则越不平滑, 但滞后少;	
0x8072	Charging_Level_Factor	发送充电器命令后, 屏体阈值放大系数 (原始阈值乘以该值, 然后除以 256, 再加上原始阈值就是最终阈值)			
0x8073	UNEVEN_DENSITY	疏密不均系数			
0x8074	ObjAvg	校准后全屏平均值			
0x8075	ObjMinW	ObjMin		ObjW:	
0x8076	Back_Shape_Restrain	水状态时, 背面按压形变使能开关	背面按压形变抑制处理: 0-127 可配; 配 0 关闭处理, 1-127 为形变抑制一致性阈值		
0x8077	GreenMode_Control	GreenMode 周期 (ms 为单位)			
0x8078	EDGE_COMPLEM_THRES	边缘补值阈值 (0~255, 配值越大补值条件越严格)			
0x8079	EDGE_COMP	X 方向补值系数 (0~255, 配 0 不补值, 配的越大补值越多)			

	LEMENT_X					
0x807A	EDGE_COMP LEMENT_Y	Y 方向补值系数（0~255，配 0 不补值，配的越大补值越多）				
0x807B	Large_Top_Li mit	判断大面积的高阈值				
0x807C	Large_Low_Li mit	判断大面积的低阈值				
0x807D	Large_Touch	大面积触摸点个数				
0x807E	Drv_GroupA_ Num	AllDrivi ng 使 能开关	Reserved		Driver_Group_A_Number	
0x807F	Drv_GroupB_ Num	Reserved		Dual_Fr eq 使能 开关	Driver_Group_B_Number	
0x8080	Sensor_Num	Sensor_Number				
0x8081	FreqA_Factor	驱动组 A 的驱动频率倍频系数    GroupA_Frequency = 倍频系数 * 基频				
0x8082	FreqB_Factor	驱动组 B 的驱动频率倍频系数    GroupB_Frequency = 倍频系数 * 基频				
0x8083	Pannel_BitFre qL	驱动组 A、B 的基频（61.0352*1~61.0352*256*5 Hz）				
0x8084	Pannel_BitFre qH					
0x8085	Self_Tx_Ctrl	Self_R x_PG A_AA F_Cor ner	RG_R XRINZ	Self_TxRef_TRIM （4 档可调）	Self_Tx Ref_SE L	Self_DAC_Gain（8 档可调） 0: Gain 最大 7: Gain 最小
0x8086	Self_Rx_Ctrl	Self_P GA_C	Self_PGA_R（4 档 可调）	Self_Rx_Vcml（4 档 可调）		Self_PGA_GainC（8 档可调）
0x8087	Pannel_Tx_Ctr l	Pannel _Rx_P GA_A AF_Co rner	RG_R XRINZ	Pannel_TxRef_TRI M （4 档可调）	Pannel_ TxRef_S EL	Pannel_DAC_Gain（8 档可调） 0: Gain 最大 7: Gain 最小
0x8088	Pannel_Rx_Ct rl	Pannel _PGA _C	Pannel_PGA_R（4 档可调）	Pannel_Rx_Vcml（4 档可调）		Pannel_PGA_GainC（8 档可调）
0x8089	Pannel_Dump _Shift	Reserved			屏原始值放大系数（2 的 N 次方）	
0x808A	Drv_Frame_C ontrol	Temp Config 临时配	Repeat_Num: 采 样重复次数		SubFrame_DrvNum: 采样驱动分组数 可配置为 3、4、5、6、7； 不要配置为除上述外的其它组数，否则可能出错	



		置使能				
0x808B	ADC_TAB_Period	Reserved				
0x808C	PGA_GAINF	Reserved				
0x808D	Freq_Hopping_Start	跳频范围的起点频率（以 BitFreq 为单位）				
0x808E	Freq_Hopping_End	跳频范围的终点频率（以 BitFreq 为单位）				
0x808F	Hopping_Flag	Hopping_En: 跳频使能开关	Delay_Hopping: 有手指不跳频开关	Dis_Force_Ref: 跳频更新基准开关	Seamless_Hopping_En: 无缝跳频开关	Reserved
0x8090	Noise_Detect_Times	Detect_Stay_Times 单次噪声检测中每个频率点检测次数，建议小屏 2 大屏 1		Detect_Confirm_Times（多次噪声检测后确定噪声量，建议 20~30）		
0x8091	Hopping_Threshold	Fast_Hopping_Limit 当前频率的干扰值大于 Fast_Hopping_Limit*4 的时候才会启动快速跳频判断. 0~5: 配置值为 5; 6~15: 配置值为 N;			Hopping_Hit_Threshold （最优频率选定条件，当前工作频率干扰量一最小干扰量>设定值 x4，则选定最优频率和跳频）	
0x8092	Noise_Min_Threshold	当 ESD 导致最小干扰点大于此阈值时，进行快速消减处理。0 为禁止此功能，设很大的值（255）也相当于禁止此功能。需要此功能时，建议的设置值是在正常干扰的最低频点（取 LCD 和共模干扰的大者）基础上加上 5~20。				
0x8093	Noise_PGA_GAIN	Reserved			Noise_PGA_GAINF	
0x8094	Noise_Dump_Shift	Reserved	Noise_Dump_RightShift		Noise_Dump_Shift	
0x8095	Hopping_seg1_Normalize	Seg1 Normalize 系数（Rawdata = 采样数据*设置值/128）				
0x8096	Hopping_seg1_Factor	跳频检测区间频段 1 中心点倍频系数（适用于驱动 A，驱动 B 在此基础上换算出来）				
0x8097	Hopping_seg2_Normalize	Seg2 Normalize 系数（Rawdata = 采样数据*设置值/128）				
0x8098	Hopping_seg2_Factor	跳频检测区间频段 2 中心点倍频系数（适用于驱动 A，驱动 B 在此基础上换算出来）				
0x8099	Hopping_seg3_Normalize	Seg3 Normalize 系数（Rawdata = 采样数据*设置值/128）				
0x809A	Hopping_seg3	跳频检测区间频段 3 中心点倍频系数（适用于驱动 A，驱动 B 在此基础上换算出来）				

	_Factor		
0x809B	Hopping_seg4_Normalize	Seg4 Normalize 系数 (Rawdata = 采样数据*设置值/128)	
0x809C	Hopping_seg4_Factor	跳频检测区间频段 4 中心点倍频系数 (适用于驱动 A, 驱动 B 在此基础上换算出来)	
0x809D	Hopping_seg5_Normalize	Seg5 Normalize 系数 (Rawdata = 采样数据*设置值/128)	
0x809E	Hopping_seg5_Factor	跳频检测区间频段 5 中心点倍频系数 (适用于驱动 A, 驱动 B 在此基础上换算出来)	
0x809F	Hopping_seg6_Normalize	Seg6 Normalize 系数 (Rawdata = 采样数据*设置值/128)	
0x80A0	Jitter_Threshold	无缝跳频备份原始值抖动阈值, 低灵敏度建议 15, 高灵敏度可以放宽到 30	
0x80A1	Avg_Thre	无缝跳频平均值阈值, 为该值*2	
0x80A2	Max_Thre	无缝跳频最大值阈值, 为该值*4	
0x80A3	Key 1	Key 1 位置: 0-255 有效 (其中 0 表示无按键 4 个键位置均为 8 的倍数时表示为独立按键, 需清 0 其它部位 rawdata)	
0x80A4	Key 2	Key 2 位置: 0-255 有效 (其中 0 表示无按键 4 个键位置均为 8 的倍数时表示为独立按键, 需清 0 其它部位 rawdata)	
0x80A5	Key 3	Key 3 位置: 0-255 有效 (其中 0 表示无按键 4 个键位置均为 8 的倍数时表示为独立按键, 需清 0 其它部位 rawdata)	
0x80A6	Key 4	Key 4 位置: 0-255 有效 (其中 0 表示无按键 4 个键位置均为 8 的倍数时表示为独立按键, 需清 0 其它部位 rawdata)	
0x80A7	Key_Area	长按更新时间 (1~16 s) 0 表示抑制 3 秒	按键有效区间设置 (单侧): 0~15 有效
0x80A8	Key_Touch_Level	触摸按键按键阈值	
0x80A9	Key_Leave_Level	触摸按键松键阈值	
0x80AA	Key_Sens	KeySens_1 (按键 1 灵敏度系数)	KeySens_2 (按键 2 灵敏度系数)
0x80AB	Key_Sens	KeySens_3 (按键 3 灵敏度系数)	KeySens_4 (按键 4 灵敏度系数)
0x80AC	Key_Restrain	手指从屏上离开后抑制按键的时间 (以 100ms 为单位)。	独立按键临键抑制参数 (当次大值超过最大值的 Key_Restrain/16 时则不输出按键), 推荐设置 7±2。
0x80AD	Key_DownEdge_Filter	Reserved	手指滑动并从屏体最下端离开后的按键抑制时间 (以 100ms 为单位) 0~15: 配置值为 (N*100)ms;
0x80AE		Reserved	
0x80AF		Reserved	

0x80B0	LINK_SWITCH	Period_Switch_Dis(检测周期切换禁止开关)	LinkLarge_En (大面积辅助接近检测开关)	Reserved	Move_Window_Dis(移动窗格功能禁止开关)	Hotknot cycle: (active 模式多少个周期插入一次 hotknot 检测) 0~3: 配置值为 N+1;	LINK_PXY_EN (接近使能)	LINK_FUNCTION_EN (Link 功能使能)
0x80B1	LINK_THRESHOLD	Data_NoiseThreshold 数据传输的阈值						
0x80B2	PXY_THRESHOLD	Pxy_NoiseThreshold 接近检测的阈值						
0x80B3	Link_DUMP_SHIFT	Period_Default (默认检测周期)	Reserved	RG_RX_RINZ	Rx_Self :自容接收使能	Link_Dump_shift		
0x80B4	Link_Rx_Ctrl	Link_PGA_C	Link_PGA_R (4 档可调)		Link_Rx_Vcml (4 档可调)		Link_PGA_GainC (8 档可调)	
0x80B5	Freq_Gain0	400K 增益调准,调整量为 N/8,N 等于 0 时无效				450K 增益调准,调整量为 N/8,N 等于 0 时无效		
0x80B6	Freq_Gain1	300K 增益调准,调整量为 N/8,N 等于 0 时无效				350K 增益调准,调整量为 N/8,N 等于 0 时无效		
0x80B7	Freq_Gain2	200 增益调准,调整量为 N/8,N 等于 0 时无效				250K 增益调准,调整量为 N/8,N 等于 0 时无效		
0x80B8	Freq_Gain3	Reserved				150K 增益调准,调整量为 N/8,N 等于 0 时无效		
0x80B9	SELF_LARGE_THRESHOLD	自容大面积阈值 (辅助接近检测)						
0x80BA	SELF_LARGE_CONSISTENCY	自容大面积一致性阈值 (设定值 × 50 = 阈值,辅助接近检测)						
0x80BB	SELF_LARGE_TIME	自容大面积持续时间 (辅助接近检测)						
0x80BC	PXY_THRESHOLD_HIGH	没有大面积时抬高的接近检测阈值 (默认为 PXY_THRESHOLD × 3 / 2)						
0x80BD	Edge_Res_Area	左右边缘坐标抑制范围, 1/64 pitch 为单位						
0x80BE	Edge_Res_Bottom	下边缘坐标抑制范围, 1/64 pitch 为单位						
0x80BF	HighSens_Delay	高灵敏度状态持续时间, 为 0 表示手触摸前不退出高灵敏度状态						
0x80C0	HighSens_AddTime	滑动进高灵敏度检测时间 (以 1 个主循环的周期计时)						
0x80C1	HighSens_Dis	滑动进高灵敏度检测距离 (距离为绝对距离平方, 1/4 通道为单位)						

0x80C2	HighSens_Click_Time1	点击进高灵敏度最短时间（以 1 个主循环的周期计时）	
0x80C3	HighSens_Click_Time2	点击进高灵敏度最长时间（以 1 个主循环的周期计时）	
0x80C4	Level_Shift	H_Combine_factor:高灵敏度状态下拆合点放大系数，在高灵敏度状态下，拆合点距离为正常状态下的 $1/2 * \sqrt{4 + H\_Combine\_factor}$ 倍。	手指触摸阈值、灵敏度检测阈值放大系数
0x80C5	Key_Finger_Leave_Level	按键手指触摸松键阈值	
0x80C6	Key_Finger_Touch_Level	按键手指触摸按键阈值	
0x80C7	Key_HighSens_Low_Level	按键高灵敏度检测下阈	
0x80C8	Key_HighSens_High_Level	按键高灵敏度检测上阈	
0x80C9	Finger_Leave_Level	屏体手指触摸松键阈值	
0x80CA	Finger_Touch_Level	屏体手指触摸按键阈值	
0x80CB	HighSens_Low_Level	屏体高灵敏度检测下阈	
0x80CC	HighSens_High_Level	屏体高灵敏度检测上阈	
0x80CD	SelfCancel_R0	SelfCap RC 参数，驱动第一组 R(0-255)	
0x80CE	SelfCancel_R1	SelfCap RC 参数，驱动第二组 R(0-255)	
0x80CF	SelfCancel_R2	SelfCap RC 参数，感应 R(0-255)	
0x80D0	Self_BitFreqL	主系统自容的基频（61.0352*1~61.0352*256*5 Hz）	
0x80D1	Self_BitFreqH		
0x80D2	Self_Factor	SelfCap 的驱动频率倍频系数 $Frequency = \text{倍频系数} * \text{基频}$	
0x80D3	Selfcap_Frame_Num	SelfCap_Dump_Shift	Reserved
0x80D4	Self_Drv_TouchLevel	自容驱动线有 Touch 阈值	
0x80D5	Self_Sen_TouchLevel	自容感应线有 Touch 阈值	
0x80D6	Self_LeaveLev	自容 Leave 阈值，主要用于自容基准更新	

	el		
0x80D7	HighSens_Self_Drv_TouchLevel	自容驱动线有 Touch 阈值（高灵敏度）。	
0x80D8	HighSens_Self_Sen_TouchLevel	自容感应线有 Touch 阈值（高灵敏度）	
0x80D9	SelfCancel_C0	SelfCap RC 参数，驱动第一组 C(0-127)	
0x80DA	SelfCancel_C1	SelfCap RC 参数，驱动第二组 C(0-127)	
0x80DB	SelfCancel_C2	SelfCap RC 参数，感应 C(0-127)	
0x80DC~ 0x80E9	Sensor_CH0~ Sensor_CH13	ITO Sensor0 对应的芯片通道号	
0x80EA~ 0x8103	Driver_CH0~ Driver_CH25	ITO Driver0 对应的芯片通道号	
0x8104	Driver_Gain0~1	通道 1 调整系数 N，调整量为 (N)/16,N 等于 0 时无效	通道 0 调整系数 N，调整量为 (N)/16,N 等于 0 时无效
0x8105	Driver_Gain2~3	通道 3 调整系数 N，调整量为 (N)/16,N 等于 0 时无效	通道 2 调整系数 N，调整量为 (N)/16,N 等于 0 时无效
0x8106	Driver_Gain4~5	通道 5 调整系数 N，调整量为 (N)/16,N 等于 0 时无效	通道 4 调整系数 N，调整量为 (N)/16,N 等于 0 时无效
0x8107	Driver_Gain6~7	通道 7 调整系数 N，调整量为 (N)/16,N 等于 0 时无效	通道 6 调整系数 N，调整量为 (N)/16,N 等于 0 时无效
0x8108	Driver_Gain8~9	通道 9 调整系数 N，调整量为 (N)/16,N 等于 0 时无效	通道 8 调整系数 N，调整量为 (N)/16,N 等于 0 时无效
0x8109	Driver_Gain10~11	通道 11 调整系数 N，调整量为 (N)/16,N 等于 0 时无效	通道 10 调整系数 N，调整量为 (N)/16,N 等于 0 时无效
0x810A	Driver_Gain12~13	通道 13 调整系数 N，调整量为 (N)/16,N 等于 0 时无效	通道 12 调整系数 N，调整量为 (N)/16,N 等于 0 时无效
0x810B	Driver_Gain14~15	通道 15 调整系数 N，调整量为 (N)/16,N 等于 0 时无效	通道 14 调整系数 N，调整量为 (N)/16,N 等于 0 时无效
0x810C	Driver_Gain16~17	通道 17 调整系数 N，调整量为 (N)/16,N 等于 0 时无效	通道 16 调整系数 N，调整量为 (N)/16,N 等于 0 时无效
0x810D	Driver_Gain18~19	通道 19 调整系数 N，调整量为 (N)/16,N 等于 0 时无效	通道 18 调整系数 N，调整量为 (N)/16,N 等于 0 时无效
0x810E	Driver_Gain20~21	通道 21 调整系数 N，调整量为 (N)/16,N 等于 0 时无效	通道 20 调整系数 N，调整量为 (N)/16,N 等于 0 时无效
0x810F	Driver_Gain22~23	通道 23 调整系数 N，调整量为 (N)/16,N 等于 0 时无效	通道 22 调整系数 N，调整量为 (N)/16,N 等于 0 时无效
0x8110	Driver_Gain24~25	通道 25 调整系数 N，调整量为 (N)/16,N 等于 0 时无效	通道 24 调整系数 N，调整量为 (N)/16,N 等于 0 时无效
0x8111	Edge_Res_Area	退出坐标边沿抑制去抖次数 0: 配置值为 50; 1~15: 配置值为 N*5; 以一个主循环周期为单位;	Reserved



0x8112	Edge_Diff_Threshold	判断 1X3、2X4 的框为手掌边缘触摸阈值，框左右差值大于阈值判定为手掌触摸	
0x8113	Proximity_Area_Limit	接近感应限制阈值	
0x8114	Proximity_Press_Time1	手指按下最短时间（以 1 个主循环的周期计时）	
0x8115	Proximity_Press_Time2	手指按下最长时间（以 1 个主循环的周期计时）	
0x8116	Proximity_Large_Touch	接近感应模式下的大面积（设置为 0 时与正常屏体共用一个大面积参数）	
0x8117	Proximity_Drv_Select	Drv_Start_Ch(驱动方向起始通道)	Drv_End_Ch（结束通道，为起始通道加此值）
0x8118	Proximity_Sense_Select	Sens_Start_Ch(感应方向起始通道)	Sens_End_Ch（结束通道，为起始通道加此值）
0x8119	Proximity_Touch_Level	设定值×10=接近感应生效阈值	
0x811A	Proximity_Leave_Level	设定值×10=接近感应无效阈值	
0x811B	Proximity_Sample_Add_Times	采样值累加次数	
0x811C	Proximity_Shake_Count	退出接近感应去抖次数	
0x811D	Proximity_Function_Switch	Reserved	
		MainCh Sel: 0: 驱动 1: 感应	
0x811E	Proximity_Line_Dist_Limit	接近感应划线判定距离阈值，固件中将此值*32	
0x811F	Proximity_DownUpdate	Reserved	下降沿更新,设置值*Proximity_Leave_Level/4
0x8120	Proximity_Stable_Time	接近感应常态基准更新时间（以 1 个主循环的周期计时）	
0x8121		Reserved	
0x8122		Reserved	
0x8123	Centroid_Ctl	质心算法在坐标计算中的比重:边缘区比重 bit0-3:中心区比重 bit4-7	
0x8124	DS_UNSP_Level	Reserved	
0x8125	Gesture_Edge_Res	手势左右边缘坐标抑制范围，1/64 pitch 为单位	
0x8126	Gesture_Time_Set	Reserved	dbClickLimit_X 手势双击 X 方向限制 (远近端限制一样)
0x8127	DbClickAreaLimit	手势双击 Y 方向近端限制	手势双击 Y 方向远端限制



	t									
0x8128	Gesture_Switch4	Reserved							Gesture_Hop_Dists: 手势唤醒关跳频开关	Mul_stroke_En: 多笔自定义手势使能开关
0x8129	Gesture_Switch1	左滑	上滑	右滑	w	o	m	e	c	
0x812A	Gesture_Switch2	下滑	z	s	^	>	v	Reserved	Reserved	
0x812B	Gesture_Switch3	Custom_En: 自定义手势识别开关	协议 2 上报开关	Reserved	Reserved	滑动最后驱动	屏双击	键双击	键单击	
0x812C	Gesture_DumpShift	GestureMutualDumpShift（手势系统互容原始值放大系数（2 的 N 次方））				GestureSelfDumpShift（手势系统自容原始值放大系数（2 的 N 次方））				
0x812D	Pannel_PGA_GainC	Reserved						Pannel_Mutual_PGA_GainC（8 档可调）		
0x812E	Gesture_BitFreqL	手势系统驱动组 A、B 的基频（61.0352*1~61.0352*256*5 Hz）								
0x812F	Gesture_BitFreqH									
0x8130	Gesttrue_Self_Rx_Ctrl	Gestur e_Self_PGA_C	Gesture_Self_PG A_R（4 档可调）	Gesture_Self_Rx_Vc mi（4 档可调			Gesture_Self_PGA_GainC（8 档可调）			
0x8131	Gesture_Ctrl_1	Gesture_INT_Time: 手势系统上报脉冲宽度设置 0~15: 配置值为（N+1）*250us； 配置为 15 时，唤醒成功后持续输出高直到主控读完唤醒信息				Gesture_Combine_Dis: 手势唤醒合框距离 0~15 可配； 合点距离为配置值的 2 倍开根号 pitch。 配 0 默认合点距离为 2 pitch				
0x8132	Gesture_Refresh_Rate	手势唤醒坐标上报率(周期为 5+ms)								
0x8133	Gesture_Large_Touch	手势唤醒大面积触摸点个数								
0x8134	Gesture_Dis	Gesture_Width 手势唤醒左右滑动有效宽度，单位为屏宽度的 1/16。 配 0 默认为 5。				Gesture_Height 手势唤醒上下滑动有效高度，单位为屏高度的 1/16。 配 0 默认为 8。				
0x8135	Gesture_TimeOut	DoubleClick_TimeOut 双击唤醒非法时间（单位为 100ms）					Gesture_Self_Si	Gesture_Self_Sample 0x10:采感应		

			to	0x11:采驱动 其它：驱动感应全采
0x8136	Gesture_Touch_Level	手势唤醒触摸阈值。		
0x8137	NewGreenWakeUpLevel	手势唤醒 NewGreen 唤醒阈值		
0x8138	GESTURE_CTRL_2	单指多笔自定义每笔间隔时间(单位 100ms, 最大限制为 1S)	Reserved	
0x8139	GESTURE_MIN_RESTRAIN	DoubleClick_Dist:双击唤醒两点间距离（以 0.5pitch 为单位）	手势唤醒最小字符限制，字符宽度/高度小于 3+（Gesture_Min_Restrain/2）通道时认为字符非法	
0x813A	HV_Factor	X 坐标方向滑动唤醒斜率配置 0~15 可配置；1（40 度）<= k <= 15（13 度））配置值越大，滑动要求与 X 坐标方向夹角越小	Y 坐标方向滑动唤醒斜率配置 0~15 可配置；（1（51 度）<= k <= 15（78 度））配置值越大，滑动要求与 Y 坐标方向夹角越小	
0x813B	Gesture_Limit_Timer	单次手势唤醒最短时间限制（以 100ms 为单位）。0~15：配置值为 N*100ms；当单次滑手势的时间小于配置时间时，不识别该次唤醒操作；配置为 0 时，该限制无效；	单次手势唤醒最长时间限制（以 1S 为单位）0~15：配置值为 N 秒；当单次滑手势的时间大于配置时间时，不识别该次唤醒操作；配置为 0 时，该限制无效	
0x813C	Config_Chksum_H	配置信息 16 位累加和校验（大端模式：高位存入在低地址）		
0x813D	Config_Chksum_L			
0x813E	Config_Fresh	配置已更新标记（主控在此写入 1）		

部分寄存器增补说明如下:

#### [0x8056] Module\_Switch1

Bit7: Water\_SpeedLimit\_En: 水状态下距离限制开关

Bit6: Water\_LargeRestrain\_En: 水状态下大面积抑制开关

Bit5-bit4: Stretch\_rank, 拉伸方式

00,01,02: 弱拉伸 0.4P

03: 自定义拉伸

Bit3: X2Y : X 与 Y 坐标调换

Bit2:Sito: SITO 使能开关

Bit1-Bit0: INT 触发方式 00: 上升沿触发 01: 下降沿触发 02: 低电平查询 03: 高电平查询

#### [0x8057] Module\_Switch2

Bit7-bit6: STP\_SE: 不同图案特殊处理

00: 无特殊处理;

01: 丝印 H 型图案;

11: 黄光 H 型图案;

Bit5: FirstFilter\_Dis: 首次去抖加大使能

Bit4: Water\_SITO: 水状态下 SITO 处理开关

0: Dis

1: En

Bit3: Water\_Proof\_Disable: 防水功能开关

0: 开启防水

1: 关闭防水

Bit2: SelfCap\_ResistCommon: 自容共模使能开关

Bit1: SelfCap\_WaterProof: 自容防水使能开关

Bit0: Touch\_Key: 按键使能开关

#### [0x8059] Module\_Switch4

Bit7: Monitor\_En: 临时配置是否进行基准监控

0: 临时配置关闭基准监控;

1: 临时配置使能基准监控

Bit6: Drop\_Water\_En: 解决滴水冒点使能开关

Bit5: Driver\_Resersal: 驱动逆序

0: 不逆序;

1: 逆序;

Bit4: Sensor\_Resersal: 感应逆序

0: 不逆序;

1: 逆序;

Bit3: Force\_Self\_RectFilter\_En: 强制使能自容滤框功能开关

Bit2: LowPower\_Dis: 关闭进入低功耗功能

0: 正常进入低功耗;

1: 不进入低功耗;

Bit1: LargeCoor\_En: 大面积上报坐使能开关

Bit0: Expand\_En: 跟随性使能开关

#### [0x8068] Filter

Bit7-bit6: First\_Filter: 手指按下首次移动时使用此滤波参数

0x00: 去抖 126 个坐标点;

0x01: 去抖 90 个坐标点;

0x10: 去抖 54 个坐标点;

0x11: 去抖 Normal\_Filter/2

Bit5-bit0: Normal\_Filter: 原始坐标窗口滤波值, 系数为 4

#### [0x8069] Mini\_Filter

Bit4: FirstFilter\_Middle: 非边缘 (最外一个通道) 的 FirstFilter

0: 去抖值为 First\_Filter;

1: 去抖值为 54;

Bit3-bit0: Mini\_Filter: X 和 Y 方向划线过程中的小 Filter 设置, 当一个方向坐标变化较大, 另一个方向坐标变化较小, 配置此值可防止变化较小的方向的坐标抖动: 原始坐标窗口滤波值, 系数为 4

0: 配置值为 4;

1~15: 配置值为 N;

#### [0x80B3] Link\_DUMP\_SHIFT

Bit7: Period\_Default: 默认检测周期

0: 默认使用 80us 检测周期;

1: 默认使用 160us 检测周期;

Bit5: RG\_RXRINZ

Bit4: Rx\_Self: 自容接收使能

0: 普通模式接收;

1: 自容模式接收;

Bit3-Bit0: Link\_Dump\_shift

配置的具体作用及调试方法详见 GT5668 配置说明。

### 3.3 坐标信息

Addr	Access	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
0x8140	R	Product ID（First Byte, ASCII 码, 如 5）							
0x8141	R	Product ID（Second Byte, ASCII 码, 如 6）							
0x8142	R	Product ID（Third Byte, ASCII 码, 如 6）							
0x8143	R	Product ID（Forth Byte, ASCII 码, 如 8）							
0x8144	R	CID							
0x8145	R	Patch 主版本号							
0x8146	R	Patch 副版本号							
0x8147	R	MASK 主版本号							
0x8148	R	MASK 副版本号							
0x8149	R	MASK 内部版本号							
0x814A	R	BondingOption				Vendor_ID			
0x814B	R	checksum							
0x814C	R	Reserved							
0x814D	R	Reserved							
0x814E	R/W	Buffer Status	Large Detect	Proximity Valid	HaveKey	Number of Touch Points			
0x814F	R	Touch Sta	Hover Sta	HotKnot	Rsvd	track_id			
0x8150	R	Hotknot=0: point 1 x coordinate (low byte) Hotknot=1: Bit7,Pxy_Status; Bit6,Approch_Valid							
0x8151	R	Hotknot=0: point 1 x coordinate (high byte) Hotknot=1: Bit7,Pxy_Status; Bit6,Approch_Valid							
0x8152	R	point 1 y coordinate (low byte)							
0x8153	R	point 1 y coordinate (high byte)							
0x8154	R	Point 1 size (W)							
0x8155	R	point 1 size (H)							
0x8156	R	Reserved							
0x8157	R	Touch Sta	Hover Sta	Reserved		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 (W)							
0x815D	R	point 2 size (H)							
0x815E	R	Reserved							
0x815F	R	Touch Sta	Hover Sta	Reserved		track_id			
0x8160	R	point 3 x coordinate (low byte)							
0x8161	R	point 3 x coordinate (high byte)							
0x8162	R	point 3 y coordinate (low byte)							

0x8163	R	point 3 y coordinate (high byte)			
0x8164	R	point 3 size (W)			
0x8165	R	point 3 size (high byte)			
0x8166	R	Reserved			
0x8167	R	Touch Sta	Hover Sta	Reserved	track_id
0x8168	R	point 4 x coordinate (low byte)			
0x8169	R	point 4 x coordinate (high byte)			
0x816A	R	point 4 y coordinate (low byte)			
0x816B	R	point 4 y coordinate (high byte)			
0x816C	R	point 4 size (W)			
0x816D	R	point 4 size (H)			
0x816E	R	Reserved			
0x816F	R	Touch Sta	Hover Sta	Reserved	track_id
0x8170	R	point 5 x coordinate (low byte)			
0x8171	R	point 5 x coordinate (high byte)			
0x8172	R	point 5 y coordinate (low byte)			
0x8173	R	point 5 y coordinate (high byte)			
0x8174	R	point 5 size (W)			
0x8175	R	point 5 size (H)			
0x8176	R	Reserved			
0x8177	R	Touch Sta	Hover Sta	Reserved	track_id
0x8178	R	point 6 x coordinate (low byte)			
0x8179	R	point 6 x coordinate (high byte)			
0x817A	R	point 6 y coordinate (low byte)			
0x817B	R	point 6 y coordinate (high byte)			
0x817C	R	point 6 size (W)			
0x817D	R	point 6 size (H)			
0x817E	R	Reserved			
0x817F	R	Touch Sta	Hover Sta	Reserved	track_id
0x8180	R	point 7 x coordinate (low byte)			
0x8181	R	point 7 x coordinate (high byte)			
0x8182	R	point 7 y coordinate (low byte)			
0x8183	R	point 7 y coordinate (high byte)			
0x8184	R	point 7 size (W)			
0x8185	R	point 7 size (H)			
0x8186	R	Reserved			
0x8187	R	Touch Sta	Hover Sta	Reserved	track_id
0x8188	R	point 8 x coordinate (low byte)			
0x8189	R	point 8 x coordinate (high byte)			
0x818A	R	point 8 y coordinate (low byte)			
0x818B	R	point 8 y coordinate (high byte)			
0x818C	R	point 8 size (W)			



0x818D	R	point 8 size (H)			
0x818E	R	Reserved			
0x818F	R	Touch Sta	Hover Sta	Reserved	track_id
0x8190	R	point 9 x coordinate (low byte)			
0x8191	R	point 9 x coordinate (high byte)			
0x8192	R	point 9 y coordinate (low byte)			
0x8193	R	point 9 y coordinate (high byte)			
0x8194	R	point 9 size (W)			
0x8195	R	point 9 size (H)			
0x8196	R	Reserved			
0x8197	R	Touch Sta	Hover Sta	Reserved	track_id
0x8198	R	point 10 x coordinate (low byte)			
0x8199	R	point 10 x coordinate (high byte)			
0x819A	R	point10 y coordinate (low byte)			
0x819B	R	point 10 y coordinate (high byte)			
0x819C	R	point 10 size (W)			
0x819D	R	point 10 size (H)			
0x819E	R	Reserved			
0x819F	R	KeyValue			
0x81A0	R	Checksum( sum(0x814E:cur,len)==0 ), 长度 len="Touch Points"*8+3			

部分寄存器增补说明如下：

#### [0x814A] Bit3~Bit0:Vendor\_ID

当前模组选项信息，由电路上的 sensor\_opt1 和 sensor\_opt2 引脚来共同决定标识，当两个选项脚外部连接状态不同时，分别表示 6 种不同的 sensor，如下表所示：

sensor_opt1	sensor_opt2	Vendor_id
GND	GND	0
VDDIO	GND	1
NC	GND	2
GND	300K	3
VDDIO	300K	4
NC	300K	5

#### [0x814E]

Bit7: Buffer status, 1 表示坐标（或按键）已经准备好，主控可以读取；0 表示未就绪，数据无效。

当主控读取完坐标后，必须通过 I<sup>2</sup>C 将此标志（或整个字节）写为 0。

Bit4: HaveKey, 1 表示有按键, 0 表示无按键 (已经松键)。

Bit3~0: Number of touch points, 屏上的坐标点个数。

#### [0x814F]

Bit7: touch\_sta, 1 表示是高灵敏度触摸坐标; 0 表示正常灵敏度触摸坐标。

Bit6:Hover\_sta, 1 表示是 Hover 坐标; 0 表示非 Hover 坐标。

Bit5:Hotknot, 1 表示是 Hotknot 接近信息; 0 表示非 hotknot 接近信息。

Bit3~0: track id, 触摸点 ID 号。

#### [0x8177] KeyValue

按键值, KeyValue 的位置并不固定, 而是跟在有效坐标的后面。例如 0x8177 是屏上有 5 个坐标时的按键位置, 而有 4 个坐标时按键位置则在 0x816F。

### 3.4 手势信息

(手势特征信息: 复用坐标信息地址)

Addr	Access	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
0x8140	R	Product ID（First Byte，ASCII 码，G）							
0x8141	R	Product ID（Second Byte，ASCII 码，E）							
0x8142	R	Product ID（Third Byte，ASCII 码，S）							
0x8143	R	Product ID（Forth Byte，ASCII 码，T）							
0x8144	R	手势主版本号							
0x8145	R	手势副版本号							
0x8146	R	手势内部版本号							
0x8147	R	MASK 主版本号							
0x8148	R	MASK 副版本号							
0x8149	R	MASK 内部版本号							
0x814A	R	BondingOption				Vendor_ID			
0x814B	R	checksum							
0x814C	R/W	手势类型（字符 ASCII 码表示 0x21-0x7E），右滑（0xAA），左滑（0xBB），下滑（0xAB），上滑（0xBA），双击（0xCC），按键单击（0xC1、0xC2、0xC4、0xC8，低四位改为按键键值），自定义（0x01~0x0A）							
0x814D	R	手势触摸点个数（坐标存放位置 0x8A40）							
0x814E	R	0x02：全部点协议 0x03：关键点协议 0x10：单笔手势 0x20：多笔手势。							
0x814F	R	缓冲区长度							
0x8150~0x816F	R	缓冲区内容							
0x8170	R	校验和（从起始地址 0x814C 开始算校验和）							

(手势坐标信息)

Addr	Access	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
0x8A40	R	Gesture point 1 x coordinate (low byte)							
0x 8A41	R	Gesture point 1 x coordinate (high byte)							
0x 8A42	R	Gesture point 1 y coordinate (low byte)							
0x 8A43	R	Gesture point 1 y coordinate (high byte)							
0x 8A44~ 0x 8B3F	R	Gesture point 2~64 coordinate (坐标个数为 0x814D 的值)							

### 3.5 GT5668 的命令状态寄存器

Addr	Access	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
0x81A8	R	GT5668_Status: 0x00: 纯触控检测状态; 0x88: 从接近检测状态; 0x99: 主接近检测状态; 0xAA: 数据接收状态; 0xBB: 数据发送状态, 表明发送缓冲区被正确刷新。							
0x81A9	R	GT5668_Status_Bak: GT5668_Status 的备份							

该区域的目的是可以通过查询 GT5668 的状态来判断是否命令发送成功, 该区域的数据任意时刻都有效。

### 3.6 HotKnot 的状态寄存器

Addr	Access	Items	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
0x8800	R	HotKnotSendStatus	HotKnot 发送状态寄存器							
0x8801	R	HotKnotRevStatus	HotKnot 接收状态寄存器							
0x8802	R	HotKnotSendStatusBak	HotKnot 发送状态寄存器的备份, 与 HotKnotSendStatus 内容是一样的							
0x8803	R	HotKnotRevStatusBak	HotKnot 接收状态寄存器的备份, 与 HotKnotRevStatus 内容是一样的							
0x8804~ 0x880E		NC	Reserved							
0x880F	R/W	HotKnotNotifyStatus	当有事物需要 Host 来处理时, HotKnotNotifyStatus 写入 0xAA, 以 INT 的方式通知 Host, Host 处理完事物, 写入一个非 0xAA 的数, IC 再往下执行, 否则 IC 会最多死等 252 个 Tick (10ms) 的时间							

该区域的数据只有在 Receive mode 或 Send mode 模式下, 才有效。

### 3.7 HotKnot 的发送缓冲区

Addr	Access	Items	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
0x8980	W	DataLength	有效数据的长度							
0x8981	W	Data0	第 1 个字节数据							
0x8982	W	Data1	第 2 个字节数据							
.....	W	.....	.....							
0x89FF	R/W	Data127	第 127 个字节数据							
0x8A00		DataChkSum	包括长度数据在内和值的补码，注意跟随在数据在后，并不固定在此位置							
0x8A7F~ 0x8A80		NC	Reserved							
0x8A81	W	DataFresh	固定位置，数据缓冲区已更新标记(由主控写入标记 0xAA)							

- 1) 该区域的数据只有在 **Receive mode** 模式下，才可以往该区域写入数据才，否则会产生不可预期的结果。
- 2) 通过 I2C 发送完待发送数据后，必须再往 **0x8A81** 位置写入 0xAA，GT5668 的 HotKnot 功能才会启动发送。否则仍处于接收数据状态。编程时，必须是先发完待传送数据后，再往 **0x8A81** 处写 0xAA。
- 3) 在往 **0x8980** 中写入数据时，必须保证此时 IC 运行的 Hotknot\_FW，可以从其版本号区别。
- 4) 支持的数据包长度为：2、4、8、16、32、64、128 （单位为 Byte）。

### 3.8 HotKnot 的接收缓冲区

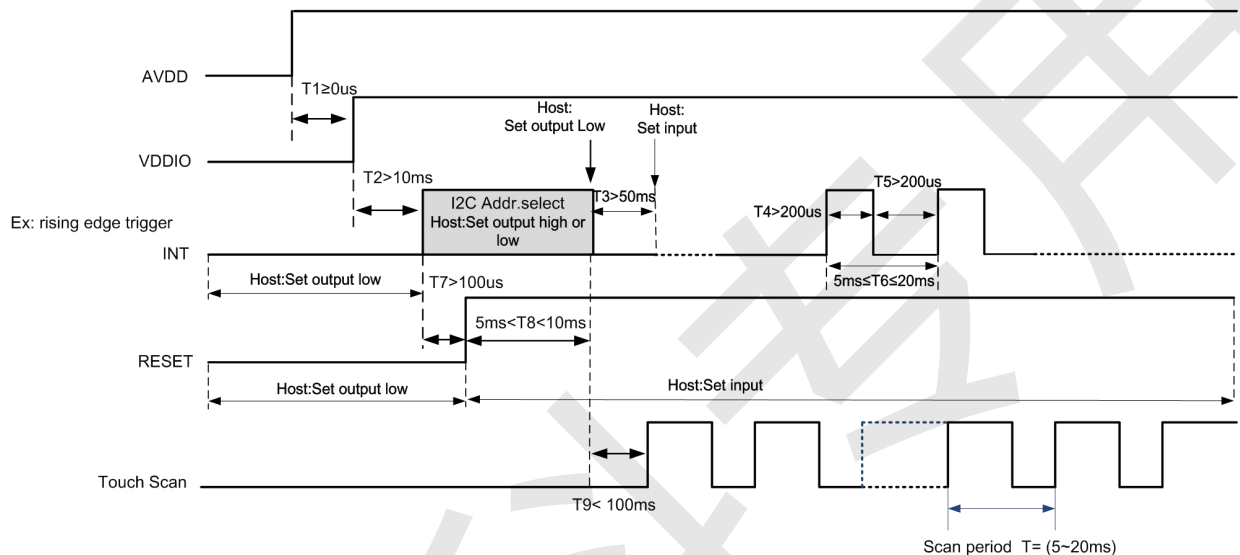
Addr	Access	Items	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
0x8B00	R/W	HotKnotRevStatus	buffer status							
0x8B01	R/W	DataLength	有效数据的长度							
0x8B02	R	Data0	第 1 个字节数据							
0x8B03	R	Data1	第 2 个字节数据							
.....	R	.....	.....							
0x8B81	R	Data127	第 128 个字节数据							
0x8B82 -0x8B83	R	DataChkSum	数据 CRC16 校验，注意仅随在数据在后，并不固定在此位置							

- 1) **0x8B00.bit7**: Buffer status 为 1 时，接收数据缓冲区数据已准备好，可读取。同时对应的 HotKnotRevStatus 状态为 0x03。
- 2) 该区域的数据只有在 Receive mode 模式下才有效。buffer status 为 1 时，表明接收数据缓冲区数据已准备好，可读取。
- 3) DataLength 必须小于等于 128。
- 4) 支持的数据包长度为：2、4、8、16、32、64、128（单位为 Byte）。

## 4. 上电初始化与寄存器动态修改

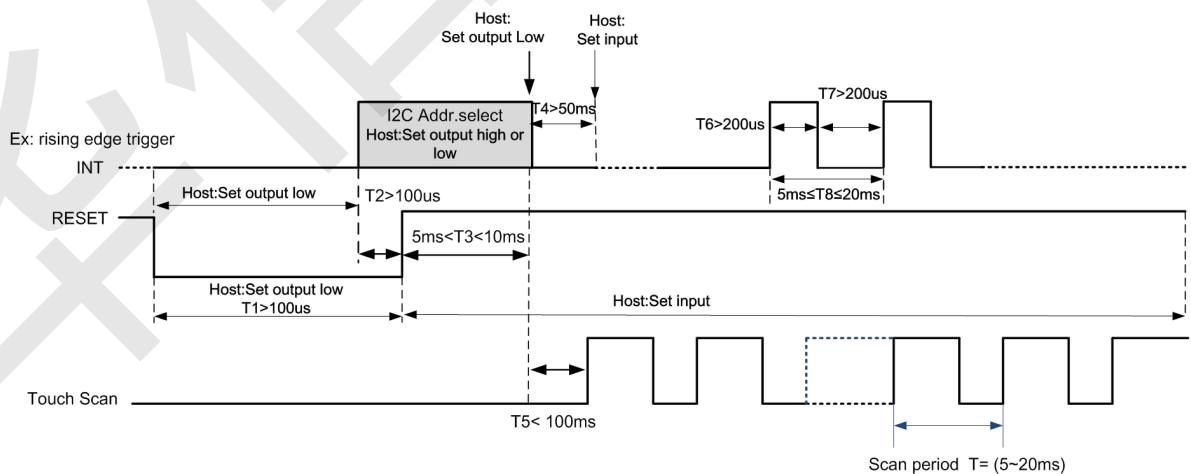
### 4.1 GT5668 上电时序

主机上电后，需要控制 GT5668 的 AVDD、VDDIO、INT、Reset 等脚位，控制时序请遵从如下时序图：



INT T2 时间后，主控是要输出高，还是低，取决于主机要用何 I<sup>2</sup>C 从设备地址与 GT5668 芯片通信，若用地址 0x28/0x29，则输出高；若用地址 0xBA/0xBB，则输出低。

主机复位 GT5668 的控制时序如下：

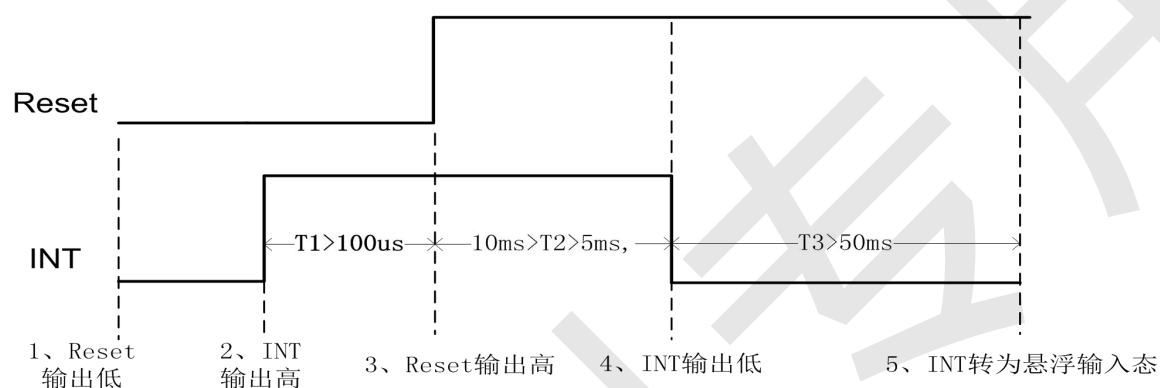




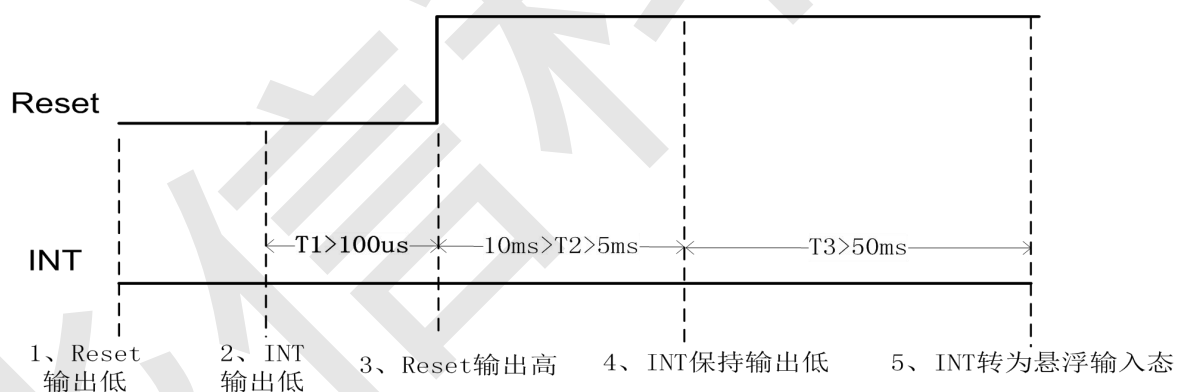
## 4.2 上电或复位 I2C 地址选择

GT5668 的 I2C 从设备地址有两组，分别为 0xBA/0xBB 和 0x28/0x29。主控在上电初始化时或通过 Reset 脚复位（唤醒）时，均需要设定 I2C 设备地址。控制 Reset 和 INT 口时序可以进行地址设定，设定方法及时序图如下：

设定地址为 0x28/0x29 的时序：



设定地址为 0xBA/0xBB 的时序：



### 4.3 上电发送配置信息

主机控制 GT5668 上电过程中，当主控将自身 INT 转化为悬浮输入态后，需要延时 50ms 再判断是否需要发送配置信息。如果收到了“0x01”请求，就需要发送配置信息，否则不需要。

### 4.4 主控响应 “INT Request”

#### 1) Request 数据说明

数据	说明
0x00	主控发给触控 IC 的 ACK
0x01	请求主控下发配置信息
0x03	请求主控复位 GT5668
0xFF	IDLE，无需处理
Others	保留，无需处理

#### 2) 如何响应“0x01”请求

- 主控发生 INT 中断且读取到 0x814E=0 时，请读取 0x8044 “Request”寄存器，如果 0x8044=0x01，则表明是“0x01”请求。
- 将所有配置信息通过 I2C 写入到配置信息区中（参考 3.2 配置信息一节）。
- 通过 I2C 将 0x8044 写 0，完成“0x01”请求的响应。

#### 3) 如何响应“0x03”请求

- 主控发生 INT 中断且读取到 0x814E=0 时，请读取 0x8044 “Request”寄存器，如果 0x8044=0x03，则表明是“0x03”请求。
- 按照复位时序对 GT5668 进行复位，完成“0x03”请求。

### 4.5 寄存器动态修改

GT5668 支持寄存器动态修改，当按照第 2 节时序对配置区内（0x8050—0x813B）任何寄存器修改时，需要更新 Config\_Chksum（0x813C/0x813D），并在最后将 Config\_Fresh（0x813E）写为 1，否则不生效；对配置区外的寄存器改写则无需更改 Config\_Chksum 和 Config\_Fresh。

## 5. 坐标读取

主控可以采取轮询或 INT 中断触发方式来读取坐标，采用轮询方式时可采取如下步骤读取：

- 1) 按第二节时序，先读取寄存器 0x814E，若当前 buffer（buffer status 为 1）数据准备好，则依据手指个数读、按键状态取相应个数的坐标、按键信息。
- 2) 若在 1 中发现 buffer 数据（buffer status 为 0）未准备好，则等待 1ms 再进行读取。

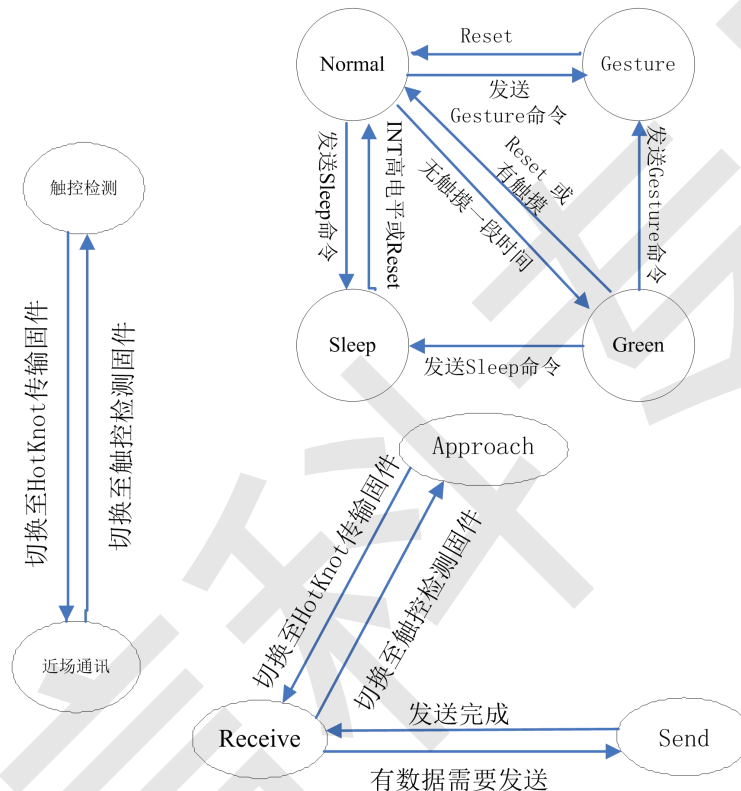
采用中断读取方式，触发中断后按上述轮询过程读取坐标。

GT5668 中断信号输出时序为（以输出上升沿为例，下降沿与此时序类同）：

- 1) 待机时 INT 脚输出低。
- 2) 有坐标更新时，输出上升沿。
- 3) 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）。请在一个周期内将坐标读走并将 Buffer status(0x814E)写为 0。
  - a) 若主控未在一个周期内读走坐标，下次 GT5668 即使检测到坐标更新会再输出一个 INT 脉冲但不更新坐标。
  - b) 若主控一直未读走坐标，则 GT5668 会一直打 INT 脉冲。

## 6. 工作模式切换

GT5668 工作模式分为 Normal、Low Power(Green)、Sleep、Gesture 四种，各种工作状态间相互转换关系如下图所示：



### 6.1 Normal Mode

GT5668 在 Normal mode 时，最快的坐标刷新周期为 5ms-20ms 间（依赖于配置信息的设定，配置信息可控周期步进长度为 1ms）。

### 6.2 Green Mode

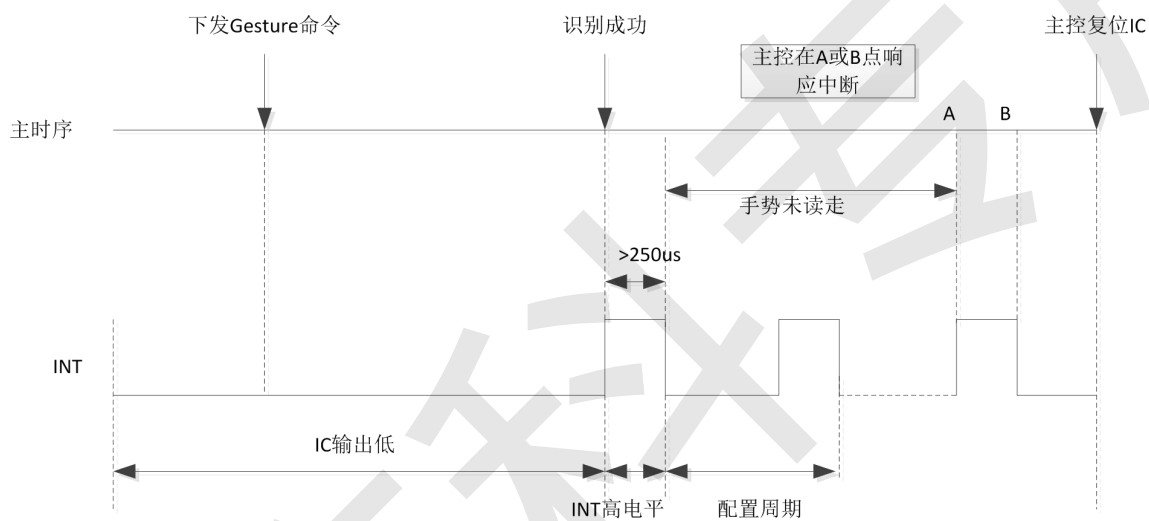
Normal mode 状态下，一段时间无触摸事件发生，GT5668 将自动转入 Green mode，以降低功耗。GT5668 无触摸自动进入 Green mode 的时间可通过配置信息设置，范围为 0~15s，步进为 1s。在 Green mode 下，GT5668 扫描周期约为 40ms，若检测到有触摸动作发生，自动进入 Normal mode。

### 6.3 Gesture Mode

若主 CPU 通过 I2C 命令，GT5668 进入 Gesture 模式后，可通过滑动屏体、双击或在屏体书写特定小写字母实现唤醒。

在 Gesture 模式下，GT5668 检测到手指在屏体上滑动足够的长度、双击动作、书写特定字符、书写自定义字符，INT 就会输出一个 250us 以上的脉冲或一直维持高/低(配置可配)，主控检测到脉冲/电平状态后醒来亮屏。

以 INT 高唤醒主控为例（低唤醒则 INT 波形相反）：

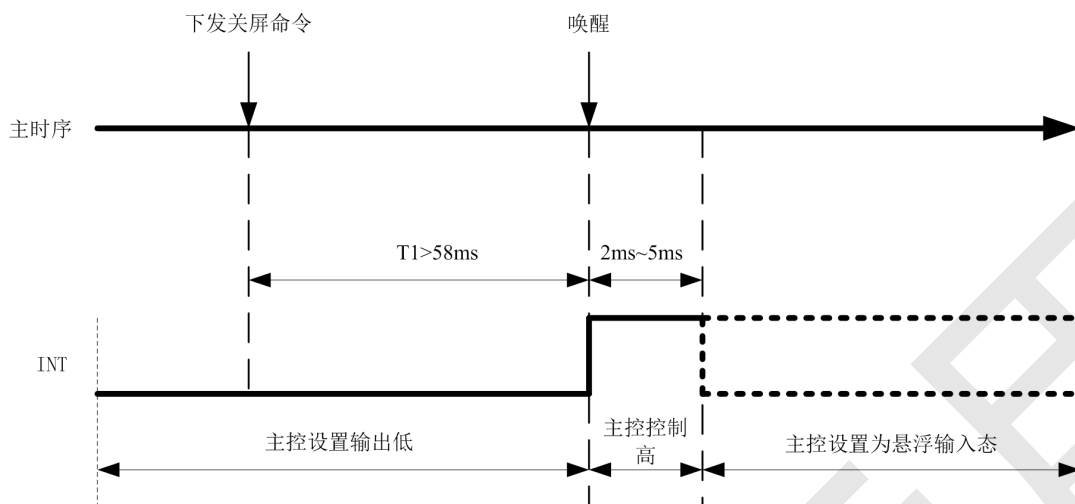


### 6.4 Sleep Mode

主 CPU 通过 I2C 命令，使 GT5668 进入 Sleep mode（需要根据配置先将 INT 脚输出低电平、高电平或 Floating）。主控 INT 的配合由 INT 脚的外部状态及配置决定，原则是要省电以及保证 GT5668 睡眠时的 INT 输入+上（下）拉能够得到对应的高（低）电平。

当需要 GT5668 退出 Sleep mode 时，主机根据配置输出一个 2~5ms 的高（低）电平到 INT 脚，唤醒后 GT5668 将进入 Normal mode。下发 I2C 关屏命令与唤醒之间的时间间隔要求大于 58ms。

以 INT 高唤醒触控 IC 为例（低唤醒则 INT 波形相反）：



## 6.5 Approach Mode

在 GT5668 运行在 Normal mode 或 Green mode 时,主 CPU 通过下发 0x20 或 0x21 命令,使 GT5668 进入 Approach mode。该模式下,触控检测和近场的接近检测相间进行。Approach mode 在发送端与接收端模式存在区别:在发送端是会通过驱动感应通道发送约定规律约定频率的信标,发送完再检测是否收到接收端返回的约定规律约定频率的信标,以此判定有无接收端存在。在接收端,Approach mode 一直检测是否收到发送端发来的约定规律约定频率的信标,若检测到,返回约定规律约定频率的信标通知发送端。在 Approach mode 下,当发现近场范围存在可通讯终端,会以 INT 的方式通知主 CPU 来获取状态。为了保证收发双方可靠的检测到对方,当获取到接近状态后,须继续保持至少 150ms 检测,主 CPU 再下发 HotKnot 传输固件进入 Receive mode。

## 6.6 Receive Mode

在 GT5668 运行在 Approach mode 时,主 CPU 获取到 GT5668 检测到可通讯终端,主 CPU 再下发 HotKnot 传输固件使 GT5668 进入 Receive mode。在该模式下,不断地检测有无起始帧信号,检测到后,开始检测数据,接收完成后,进行校验,若校验失败,重新开始接收;若接收成功,则以 INT 方式通知主 CPU 来接收缓冲区读取数据。



## 6.7 Send Mode

在 GT5668 运行在 Receive mode 时，主 CPU 将待发数据发送至发送缓冲区，GT5668 检测到发送缓冲区被刷新且有数据需要发送时，自动从 Receive mode 切换到 Send mode。在该模式下，先发送起始帧信号，并检测到接收端有返回 ACK，紧接着发送数据信号，发送完一个数据序列，开始检测 ACK；若 ACK 没有或不对，重发刚发过的字节，重发若超过五次都失败，会将本帧数据重新开始发送，直到主 CPU 超时使其退出。数据成功发送完成后，待主 CPU 处理完或超时后，自动切换到 Receive mode。

## 7. Gesture 模式驱动修改

### 7.1 灭屏后进入 Gesture 模式

- 按电源键（或其他按键）关屏时，往 0x8040~0x8042 下发命令 0x08, 0x00, 0xF8;
- 手机自动灭屏时的修改与按电源键（或其他按键）关屏时的修改一致;
- 在灭屏的过程中，滑动、双击屏体或书写特定字符 INT 会输出一个 250us 以上（可配置）的脉冲，主控收到脉冲后读取 0x814C 的值，如满足唤醒条件则醒来亮屏，否则清零 0x814C 等待下一次脉冲。

### 7.2 灭屏后进入 Sleep 模式

- 按电源键（或其他按键）关屏时，往 0x8040~0x8042 下发命令 0x05, 0x00, 0xFB;
- 手机自动灭屏时的修改与按电源键（或其他按键）关屏时的修改一致;
- 此模式下只能通过电源键（或 home 键）唤醒。

### 7.3 按电源键（或 home 键）开屏

任何模式下按开屏键（或 Home 键）开屏，直接按照复位时序复位 IC，执行复位流程。

## 8. Gesture 模式坐标读取

在 Gesture 模式下，主控读取到 0x814C 非 0 时，可以读取手势特征信息或者手势坐标信息来描绘用户的唤醒轨迹。

手势协议类型：主控读取 0x814E 寄存器，获取手势协议类型，当前支持协议类型如下：

Bit5~bit4（辅助信息）：

0x01：单笔手势；

0x02：多笔手势。

其它：NULL；

Bit1~bit0（手势坐标信息）：

0x02：全部点协议

0x03：关键点协议

其它：NULL

Bit7：1，手势校验

手势辅助信息：主控读取 0x814F 寄存器，获取坐标信息长度，然后以这个长度读取 0x8150~0x8170 坐标信息

多笔手势断点位置：多笔手势每笔画结束位置，笔画序号从 0 开始。

手势坐标信息：主控读取 0x814D 寄存器，获取到手势轨迹点数，按照每 4 个寄存器对应一个触摸点数，然后读取 0xA2A0~0xA39F 寄存器，通过这些信息可以描绘出用户真实触摸轨迹，手势轨迹十六位信息校验和放在 0x8171~0x8172。

## 9. 版本修订记录

文件版本	修订日期	修订
Rev.00	2015-05-15	预发布版
Rev.01	2015-10-13	发布版
Rev.01a	2015-12-25	修订关于以下寄存器的描述：0x8058, 0x8069, 0x806F, 0x80A0, 0x80A1, 0x80A2, 0x8057.