

GT917S 编程指南

V0.1

2019-08-26

===== 免责声明 =====

本出版物中所述的器件应用信息及其他类似内容仅为您提供便利，它们可能由更新之信息所替代。确保应用符合技术规范，是您自身应负的责任。深圳市汇顶科技股份有限公司（以下简称“GOODIX”）对这些信息不作任何明示或暗示、书面或口头、法定或其他形式的声明或担保，包括但不限于针对其使用情况、质量、性能、适销性或特定用途的适用性的声明或担保。GOODIX 对因这些信息及使用这些信息而引起的后果不承担任何责任。未经 GOODIX 书面批准，不得将 GOODIX 的产品用作生命维持系统中的关键组件。在 GOODIX 知识产权保护下，不得暗中以其他方式转让任何许可证。

目 录

1. 接口说明.....	3
2. I ² C 通信时序	4
2.1 主机对 GT917S 进行写操作时序.....	4
2.2 主机对 GT917S 进行读操作时序.....	4
3. 寄存器列表.....	5
3.1 实时命令	5
3.2 配置信息 (RW)	6
3.3 坐标信息	18
3.4 手势信息	22
3.5 GT917S 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 GT917S 上电时序	26
4.2 上电或复位 I ² 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
7.4 建议可与 IR 配合	34
8. Gesture 模式坐标读取及校验	35
8.1 坐标读取	35
8.2 坐标及轨迹信息校验	35
9. 版本修订记录.....	37

1. 接口说明

GT917S 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 GT917S 的 RESET 口为高或低。为保证可靠复位，建议 RESET 脚输出低 100us 以上。

GT917S 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。GT917S 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。GT917S 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. I²C 通信时序

2.1 主机对 GT917S 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。

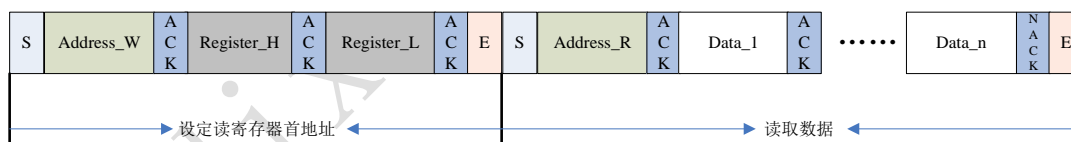
Data_1 至 Data_n: 数据字节 1-n。

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，GT917S 自动将其往高地址顺序存储。

2.2 主机对 GT917S 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，GT917S 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 I²C 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 0x01、0x02: 差值原始值 0x03: 基准更新（内部测试） 0x04: 基准校准（内部测试） 0x05: 关屏 0x06: 进入充电模式 0x07: 退出充电模式 0x08: 进入手势唤醒模式 0x0b: 手模式(不支持弱信号) 0x0c: 自动模式（自动切换手和手套） 0x20: 进入从机接近检测模式 0x21: 进入主机接近检测模式 0x22: 进入数据传输模式 0x28: 退出从机接近检测模式 0x29: 退出主机接近检测模式 0x2A: 退出数据传输模式（Reserved, 实际为主控直接复位 IC） 0x31: 保存自定义手势模版 0x35: 清空触控 IC 中保存的手势模版信息 0x36: 删除某个手势模版 0x37: 查询手势模版信息。 0xAA: ESD 保护机制使用, 由驱动定时写 AA 并定时读取检查							
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（sum(0x8040~0x8042)==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	Temp Config	First Filter New	Stretch_Rank 00: 0.4P 01: 0.4P 10: Reserved 11: 自定义系数		X2Y	Sito	INT 触发方式 00: 上升沿触发 01: 下降沿触发 10: 低电平查询 11: 高电平查询		
0x8057	Module_Switch2	Rsvd		Water_Disable_Pen	Water_SITO 水状态下 SITO 处理开关 0: Dis 1: En	Water_Proof_Disable 防水功能开关 0: 开启防水 1: 关闭防水	LargeTouch_Res_Points 0: 只抑制大面积 touch 的框 1: 有大面积 touch 则在全屏范围内抑制坐标输出	Expand Dist_Dis 坐标拉伸处理开关 0: En 1: Dis	Touch_Key 0: 无按键 1: 有按键	
0x8058	Module_Switch3	Monitor_Relate_Back_References 0: 备份基准没学好才在 Monitor	SelfCapEdge_Res_Disable 开启自容滤框后, 边缘 2 个通道内的框是	INT_Wakeup 唤醒电平 0: 高电平唤醒 1: 低电平唤醒	Rsvd	Water_Single_Dis 单指不进防水, 置 1 则使能此处理。	Rsvd	Inner_Couple_Dec 内部串扰消减处理开关 0: Dis 1: En	Shape_En 形变处理开关 0: Dis 1: En	

		r更新时做一致性判断 1: 无论是否学好在Monitor更新时都做一致性判断	否滤除 0:默认滤除 1:不滤除							
0x8059	Module_Switch4	SelfCap_GreenMode: Green Mode 用自容采样 0: Dis 1: En	Custimize 0: 关闭自定义手势。 1: 开启自定义手势	Gesture_Protocol_2 0:手势原始坐标上报采用默认协议。 1:手势原始坐标上报采用协议2。	FallingEdgeRefUp 按键下降沿更新 0: Dis 1:En	FirstFilter_Dis 首次按下去抖加大功能 0: 去抖加大 1: 正常去抖	Force_Self_RectFilter 0:Dis 1:En	Obj_Normalize_Sen 以感应为单位的全屏 Rawdata 归一化处理开关 0: Dis 1: EN	Obj_Normalize_Drv_Dis 以驱动为单位的全屏 Rawdata 归一化处理开关 0: En 1: Dis	
0x805A	Module_Switch5	Water_Large_En 水状态下大面积抑制开关 0: dis不抑制 1: En抑制	Rsvd	KEY_Form_En 1T3R 独立按键使能开关 0: 不使能 1: 使能	KEY_Sample 1T3R 独立按键采样方式 0: 互容 Single Driving 1: 自容	SelfCap_Resist Common 0: Dis 1: En	SelfCap_Water Proof 0: Dis 1: En	Speed_Stretch_En 边缘速度拉伸开关 0: Dis 1: En	sensor_key 0: driver 1: sensor	
0x805B	Noise_Reduction	高阻抗拖尾降噪平均值统计上限 (0为关闭此处理,非0为小于最大差值的N/32则认为是噪声,参与平均值统计)				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，0~14s 可配置，配置 15 则一直不进入低功耗
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/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）		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	Edge_Diff_Threshold	判断 1*3、1*4 的框为手掌边缘触摸阈值，框左右差值有一个大于阈值时判定为手掌触摸配置越小越容易抑制，（没有默认值）配置为 0 则只要有 1*3、1*4 的框就进入抑制。		
0x8062	Space	上边框的空白区（以 32 为系数）		下边框的空白区（以 32 为系数）
0x8063		左边框的空白区（以 32 为系数）		右边框的空白区（以 32 为系数）
0x8064	Large_Touch	大面积触摸点个数		
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	Normal_Filter（原始坐标窗口滤波值，系数为 4）	
0x8069	Mini_Filter	Restraining_Time 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），手指滑动并从屏体最下端离开后开始计时，若在该时间段内 touch 按键，则按键会一直被抑制，直到松开按键再次按下（配 0 则不进行该处理）。		划线过程中的小 Filter 设置，最小为 4。
0x806A	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。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。		合框距离，0~15 可配，合点距离为配置值的 2 倍平方 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	高四位为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。		低四位为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	WaterFrame Time	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。		
0x806D	WaterUpdate Time	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。		
0x806E	S_FeedBack	S 型改善负反馈量（1/256 为单位）		
0x806F	Gesture_SpaceD	双击唤醒驱动线起点裁边，以 0.5pitch 为单位，配 0 则不裁剪		双击唤醒驱动线末端裁边，以 0.5pitch 为单位，配 0 则不裁剪
0x8070	Shape_Control_Val	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		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

		按压差值大于此值时进入新形变处理， 配置小于 3 时关闭		
0x8071	Selfcap_ Frame_Num	SelfCap_Dump_Shift		SelfCap_Frame_Num
0x8072	Consistency RefUp_Level	弱信号模式下一致性变好更新阈值。 当前 RawData 比 RefData 一致性相比变好 N*5 个点则触发一致性更新。 (此配置开出来可以配更大的目的是为了避开屏一致性不好时弱信号操作会误触发一致性更新而消点。配置 0 为(600>>diff_shift))		
0x8073	Consistency Match_Level	弱信号模式下一致性最小值匹配更新阈值。 当前 RawData 比历史最小一致性相比变好 N*5 个点，且多指 touch 或有大面积 touch 则触发一致性更新。 (此配置开出来可以配更大的目的是为了避开屏一致性不好时弱信号操作会误触发一致性更新而消点。配置 0 为(ConsistencyRefUp_Level>>1))		
0x8074	ObjAvg	校准后全屏平均值		
0x8075	ObjMinW	ObjMin		ObjW:
0x8076	HV_Gradient	X 轴滑动唤醒斜率配置，0~15 可配置 (0 (45 度) <= k <= 15 (80 度))		Y 轴滑动唤醒斜率配置，0~15 可配置 (0 (40 度) <= k <= 15 (10 度))
0x8077	Gesture_Limit	DbClk_Dis: 双击唤醒两点间距离，以 0.5pitch 为单位，配置为 0 默认是 1.5pitch		Gesture_Min_Time: 手势唤醒滑动最短时间 限制，以 100ms 为单位，配置为 0 则没有 最短时间限制
0x8078	NC	Reserved		
0x8079	NC	Reserved		
0x807A	NC	Reserved		
0x807B	NC	Reserved		
0x807C	Speed_Stretch_Ctrl	Speed_Stretch_Area 边沿速度拉伸有效区域。有效区域为(N*0.5) pitch。 没有默认值，开启了速度拉伸功能后请将此 值配上，例如 3		Speed_Stretch_Limit 边沿速度拉伸速度阈值。从外往里划若在边 沿速度拉伸范围内速度大于 N/16 pitch 每 帧，则直接拉伸到边。 没有默认值，开启了速度拉伸功能后请将此 值配上，例如 8
0x807D	Slope_Weight	优化线性度的三角算法斜率 k 值		
0x807E	Drv_GroupA_ Num	All Driving	Reserved	Driver_Group_A_Number
0x807F	Drv_GroupB_ Num	Reserved	Dual_ Freq	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_ BitFreqL	驱动组 A、B 的基频 (57.2205*1~57.2205*256*5 Hz)		
0x8084	Pannel_ BitFreqH			

0x8085	Self_Tx_Ctrl	Self_Rx_PGA_AAF_Corner	Self_ADC_8M_EN	Self_TxRef_TRIM (4 档可调)	Self_TxRef_SEL	Self_DAC_Gain (8 档可调) 0: Gain 最大 7: Gain 最小
0x8086	Self_Rx_Ctrl	Self_PGA_C	Self_PGA_R (4 档可调)	Self_Rx_VCMRZ	Self_Rx_Vcmi (4 档可调)	Self_PGA_GainC (4 档可调)
0x8087	Pannel_Tx_Ctrl	Pannel_Rx_PGA_AAF_Corner	Pannel_ADC_8M_EN	Pannel_TxRef_TRIM (4 档可调)	Pannel_TxRef_SEL	Pannel_DAC_Gain (8 档可调) 0: Gain 最大 7: Gain 最小
0x8088	Pannel_Rx_Ctrl	Pannel_PGA_C	Pannel_PGA_R (4 档可调)	Pannel_RxVCMRZ	Pannel_Rx_Vcmi (4 档可调)	Pannel_PGA_GainC (4 档可调)
0x8089	Pannel_Dump_Shift	手势唤醒采样原始值放大系数 (2 的 N 次方)			屏原始值放大系数 (2 的 N 次方)	
0x808A	Drv_Frame_Control	Gesture_Mutual_Split 0: Dis 1: En	Repeat_Num	SubFrame_DrvNum		
0x808B	Gesture_SpaceS	双击唤醒感应线起点裁边, 以 0.5pitch 为单位, 配 0 则不裁剪			双击唤醒感应线末端裁边, 以 0.5pitch 为单位, 配 0 则不裁剪	
0x808C	PGA_GAINF	Reserved				PGA_GAINF (8 档可调)
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	Delay_Hopping	Dis_Force_Ref	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, 该设置最小为 5		Hopping_Hit_Threshold (最优频率选定条件, 当前工作频率干扰量-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 (如 200 或更大) 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_GAINC	Noise_PGA_GAINF
0x8094	Noise_Dump_Shift	SNR_Improving_Shift (only for SDC) 对 no-touch 区域差值进行右移处理, 仅在查看差值数据时进行		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_Factor	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)		
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	Special_Monitor_Sen_1	需要特别监控的感应线。若此条感应上 1/3 的节点(Touch 节点不算)都冒起 2*MonitorLevel 的差值且它的一致性变化不大, 则认为是异常状态, 更新基准。2 至感应总数-1 有效。配置为 0 或 1 则为不设置。		
0x80A1	Special_Monitor_Sen_2	需要特别监控的感应线。若此条感应上 1/3 的节点(Touch 节点不算)都冒起 2*MonitorLevel 的差值且它的一致性变化不大, 则认为是异常状态, 更新基准。2 至感应总数-1 有效。配置为 0 或 1 则为不设置。		
0x80A2	Special_Monitor_Level	特别监控阈值。配置为 0 则关闭此监控机制, 配置为非 0 则默认监控上下左右四根通道。任意一根通道 1/3 的节点 (Touch 节点不算) 都冒起 2*MonitorLevel 的差值且它的一致性		

		变化不大，则认为是异常状态，更新基准。				
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），0 表示抑制 600ms。		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	Self_Drv_TouchLevel	自容驱动线有 Touch 阈值，实际使用值为“配置值*2”				
0x80AE	Self_Sen_TouchLevel	自容感应线有 Touch 阈值，实际使用值为“配置值*2”				
0x80AF	NC	Reserved				
0x80B0	LINK_SWITCH	Reserved			LINK_PXY_EN (接近使能)	LINK_FUN_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	Reserved			Link_Dump_shift	
0x80B4	Link_Rx_Ctrl	Link_PGA_C	Link_PGA_R (4 档可调)	Link_Rx_VCMRZ	Link_Rx_Vcmi (4 档可调)	Link_PGA_GainC (4 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			350K 增益调准,调整量为 N/8,N 等于 0 时无效	

		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	HotKnotNoise Map	Reserved	200K	250K	300K	350K	400K	450K	
0x80BA	ExpandFactor	SmoothWeight			坐标平滑拉伸系数				
0x80BB	NC	Reserved							
0x80BC	SelfCancel_ Drv_R	SelfCap RC 参数, 驱动 R							
0x80BD	SelfCancel_ Drv_C	SelfCap RC 参数, 驱动 C							
0x80BE	SelfCancel_ Sen_R	SelfCap RC 参数, 感应 R							
0x80BF	SelfCancel_ Sen_C	SelfCap RC 参数, 感应 C							
0x80C0	Gesture_ Switch1	左滑	上滑	右滑	w	o	m	e	c
0x80C1	Gesture_ Switch2	下滑	z	s	^	>	v	双下滑	双上滑
0x80C2	Gesture_ Switch3	Reserved	Gestur e_Self _Wate r_En 手势系 统自容 防水开 关 0: Dis 1:En	Gestur e_ _Water_ En 手势系 统防水 处理开 关 0: Dis 1:En	Gestur e_ _Edge_ Restrain 手势系 统边沿 抑制开 关 0: Dis 1:En	<,本开关 和 c 互斥	屏双击	键双击	键单击
0x80C3	Pen_Delay	笔状态持续时间, 为 0 表示手触摸前不退出笔状态							
0x80C4	Pen_AddTime	进入笔状态检测周期							
0x80C5	Pen_Dist	进入笔状态检测距离, 在 Pen_AddTime 周期内达到该距离认为是有效笔触摸 (如果配成 0x00 则关闭笔)							
0x80C6	Pen_Click_ Time1	笔点击最短时间 (单位: 1/刷新率)							
0x80C7	Pen_Click_ Time2	笔点击最长时间 (单位: 1/刷新率)							
0x80C8	Finger_Leave_ Level	手指触摸松键阈值, 高于此值 2 倍认为是手指有效触摸							
0x80C9	Finger_Touch_ Level	手指触摸阈值, 高于此值 2 倍个数为 1 就认为是手指有效触摸							
0x80CA	Pen_Touch_	被动笔触摸上限阈值, 低于此值 2 倍认为是被动笔触摸							

	High_Level						
0x80CB	Low_Signal_Control	Low_Signal_Point_Max_Num 弱信号输出点数限制 (1~5 有效, 配无效值默认为 1)	Low_Signal_Ident 弱信号是否需要识别合法后才能输出	Glove_En 0: 不支持手套 1: 支持手套	Pen_En 0: 不支持被动笔 1: 支持被动笔	Low_Signal_Priority 0: 手指优先于弱信号 1: 弱信号优先于手指	Multi_Finger_Dis_Low_Signal 1: 多手指不支持弱信号
0x80CC	High_Sens_Switch	Hold_High_Sense 配 1: 关屏再开屏保持高灵敏度模式	Pen_Long_Time_Restrain 长按抑制时间 (1~6s) 配 0 表示不抑制 配大于 6 的数, 程序中会强制改配为 6	Key_High_Sens_Shift			
0x80CD	Key_Finger_Touch_Level_Base	高灵敏度下按键的 Touch level 基值 $Key_Finger_Touch_Level = Key_Finger_Touch_Level_Base * (2^{Key_High_Sens_Shift})$					
0x80CE	Key_Finger_Leave_Level_Base	高灵敏度下按键的 Leave level 基值 $Key_Finger_Leave_Level = Key_Finger_Leave_Level_Base * (2^{Key_High_Sens_Shift})$					
0x80CF	Key_High_Level_Base	高灵敏度下的弱信号上限阈值 $Key_High_Level = Key_High_Level_Base * (2^{Key_High_Sens_Shift})$ 配置了高灵敏度并且有按键时, 请务必把此值配上 (非 0), 否则 Key_High_Level 将会是随机数					
0x80D0	STRETCH_Extend1	Stretch_R0_down		Stretch_R1_down			
0x80D1	STRETCH_Extend2	Stretch_R2_down		Stretch_R0_left			
0x80D2	STRETCH_Extend3	Stretch_R1_left		Stretch_R2_left			
0x80D3	STRETCH_Extend4	Stretch_R0_right		Stretch_R1_right			
0x80D4	STRETCH_Extend5	Stretch_R2_right		Extend_RM 四边单独拉伸系数基数 配置不为 0 且为自定义拉伸时, 则进行四边各自单独拉伸			
0x80D5	Diff_Ratio_Stretch_Min	差值映射拉伸起点差值(配置为 0 时, 关闭该算法), 当 $LeaveLevel \leq 极大值 \leq LeaveLevel + Min$ 输出边界坐标点。					

	(only for SDC)					
0x80D6	Diff_Ratio_Stretch_Val (only for SDC)	<p>差值映射拉伸差值范围。</p> <p>对于 $LeaveLevel+Min \leq \text{极大值} \leq LeaveLevel+Min+Val$ 且三段拉伸已拉到极限的点，将根据 $(\text{极大值}-LeaveLevel-Min)/Val$ 的结果在三段拉伸的盲区中再一次拉伸，以优化到边效果。</p>				
0x80D7	Special_Monitor_Const	<p>特殊监控感应线的一致性变化值上限。配置为 0 则关闭此监控机制。满足 $Special_Monitor_Level$ 且一致性变化大于 $Special_Monitor_Const*100/3$ 且小于 $Special_Monitor_Const*100$，则认为是盖铁板后拿起的异常冒点状态，需要强制更新基准。</p>				
0x80D8	Gesture_BitFreqL	<p>手势系统驱动组 A、B 的基频 ($57.2205*1 \sim 57.2205*256*5$ Hz)</p>				
0x80D9	Gesture_BitFreqH					
0x80DA	Gesture_TouchLevel	<p>手势唤醒 Touch 阈值</p>				
0x80DB	Gesture_Self_Level	<p>手势唤醒自容唤醒阈值</p>				
0x80DC~ 0x80FB	Sensor_CH0~ Sensor_CH31	<p>ITO Sensor 对应的芯片通道号</p>				
0x80FC~ 0x8111	Driver_CH0~ Driver_CH21	<p>ITO Driver 对应的芯片通道号</p>				
0x8112	Gesture_W_Limit	<p>起点终点高度限制，配置为 0 为整体高度 1/2，配置为其它值则为整体高度 N/16</p>		<p>上凸点高度限制，配置为 0 为整体高度 1/2，配置为其它值则为整体高度 N/16</p>		
0x8113	GestureKey_GAIN_Shift	Reserved	GestureKey_PGA_GainC (4 档可调)	GestureKey_Dump_Shift (8 档可调)		
0x8114	ObjAvg_Sen	<p>校准后全屏平均值 (以感应为单位)</p>				
0x8115	ObjMinW_Sen	ObjMin_Sen			ObjW_Sen	
0x8116	Area_Slide	<p>Slide_Channel 滑动唤醒有效通道起始位置，进行滑动唤醒操作时，仅支持滑动该通道以下的区域 (以通道为单位，0~25 有效)。</p>			<p>Slide_Height 滑动唤醒限制高度，从 Slide_Channel 开始，向下多大范围内操作有效，配置为 0 则无限制，配置为非 0，则限制范围为 $0.5+0.5* \text{配置值}$ (即 1~4)。</p>	
0x8117		<p>Customize_Limit_Time 自定义手势超时时间 (单位为 s，配置为 0 则和固定手势长按超时时间相同)。</p>		<p>Doze_Auto_Wake_Time 手势多指长按进 Doze 后，从 Doze 模式自动醒来跑互容时间 ($N*7*48ms$)，配置为 0 时，默认为 5S</p>		
0x8118	OneRectLarge Count	<p>一个框内实际触摸点超出该阈值，判断为大面积</p>				
0x8119	WholeLarge Count	<p>全屏实际触摸点数超过该阈值，判断为大面积</p>				
0x811A	MutualKEY_Rx_Ctrl	Mutual KEY_	MutualKEY_PGA_R (4 档可调)	Mutual KEY_	MutualKEY_Rx_Vcmi (4 档可调)	MutualKEY_PGA_GainC (4 档可调)

		PGA_C		RxVCM RZ	
0x811B	MutualKEY_ GAIN_Shift	Reserved	MutualKEY_Dump_Shift		MutualKEY_PGA_GAINF (8档可调)
0x811C	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	
0x811D	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	
0x811E	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	
0x811F	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	
0x8120	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	
0x8121	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	
0x8122	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	
0x8123	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	
0x8124	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	
0x8125	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	
0x8126	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	
0x8127	Inner_Couple_ Noise_Limit	IC 内部串扰值上限。当监测出来的串扰值大于此值则将其卡在此值。实际使用大小为 N*20。			
0x8128	NC	Reserved			
0x8129	Edge_Res_ Rect	Edge_Res_Rect_Width1 边沿抑制坐标抑制区域外需要抑制的框大小, 1*(N+5)。		Edge_Res_Rect_Width2 边沿抑制坐标抑制区域外需要抑制的框大小, 2*(N+4)。	
0x812A	Edge_Res_ Shake	Reserved		Edge_Res_Shake 边缘抑制框的宽度配置, 以一个通道为单位, 建议配置 2~3。	
0x812B	Edge_Res_ Area	Edge_Res_Area_1 Y 轴末端(2.5P 角落处)边缘坐标抑制范围单独配置, 以 0.25P 为单位, 建议配置比低 4 位多 0.5P, 配置为 0 时默认使用低 4 位配置的抑制范围。		Edge_Res_Area_2 边缘坐标抑制范围, 0.25 pitch 为单位配置为 0 则关闭边缘抑制功能。	
0x812C	Charging_ Level_Factor	发送充电器命令后, 屏体阈值放大系数 (新阈值=原阈值*(1+设置值)/256)			
0x812D	Ghot_Level_	HotKnot 接近检测到一次后, 屏体阈值放大系数 (新阈值=原阈值*(1+设置值)/256)			

Factor						
0x812E	Gesture_Ctrl (互容)	Reserved	PGA_GainC	Dump_Shift		
0x812F	Gesture_FreqA_Factor	手势系统自容 Doze 的驱动频率倍频系数 $GroupA_Frequency = \text{倍频系数} * \text{基频}$				
0x8130	Gestrue_Rx_Ctrl (自容)	Gestur e_PG A_C	Gesture_PGA_R (4 档可调)	Gesture _RxVC MRZ	Gesture_Rx_Vcmi (4 档可调)	Gesture_PGA_GainC (4 档可调)
0x8131	Gesture_Ctrl_1	Pulse_Time 手势唤醒脉冲宽度: 250us 为单位, 配置为 0~14 时, 手势唤醒脉冲宽度为: (配置值+1)*250us, 配置为 15 时, 唤醒成功后一直维持高(或低, 由 Int_Wakeup 决定)直到主控清 0 寄存器 0x814C。			Gesture_Self_Frame_Num	
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, 以 100ms 为单位, 配置为 0 时默认为 15*100ms			Press_TimeOut 手势模式长按超时时间, 以 1s 为单位, 配置为 0 时默认为 2s, 配置大于 10 时为 10s	
0x8136	Gesture_Ctrl_2	Multi_Strokes_Interval 自定义手势多笔间隔时间 (100ms 为单位)。			Gesture_Min_Restrain 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。	
0x8137	Finger_Mode_Shift	O_Limit: 手势唤醒 O 的开口限制, 开口必须小于字符高或宽的 O_Limit/16			手模式下差值右移缩小系数	
0x8138	Self_Green_Level	自容 Green Mode 唤醒阈值				
0x8139	Self_BitFreqL	主系统自容的基频 (57.2205*1~57.2205*256*5 Hz)				
0x813A	Self_BitFreqH					
0x813B	Self_Factor	SelfCap 的驱动频率倍频系数 $Frequency = \text{倍频系数} * \text{基频}$				
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)				

配置的具体作用及调试方法详见 GT917S 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 9)								
0x8141	R	Product ID (Second Byte, ASCII 码, 如 2)								
0x8142	R	Product ID (Third Byte, ASCII 码, 如 8)								
0x8143	R	Product ID (Forth Byte, ASCII 码, 如 6)								
0x8144	R	Patch 版本号之 CID。CID 为软件型号标记, 为 2 位十进制数 (软件存储时请用 BCD 码表示), 从 00 开始, 最大为 26。"0"表示公版, 不需要显示出来; 1~26 为定制版, 转换为字母 A-Z 方式显示								
0x8145	R	Patch 版本号之主版本号(2 位压缩 BCD 码)								
0x8146	R	Patch 版本号之次版本号(2 位压缩 BCD 码)								
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	Rsvd	Have Key	Number of Touch Points				
0x814F	R	Touch Sta	Rsvd	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	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	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 (H)	
0x8166	R	Reserved	
0x8167	R	Touch 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	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	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	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	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	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	Reserved	track_id
0x8198	R	point 10 x coordinate (low byte)		
0x8199	R	point 10 x coordinate (high byte)		
0x819A	R	point 10 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 I²C 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。

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	Gesture ID (First Byte, ASCII 码, G)								
0x8141	R	Gesture ID (Second Byte, ASCII 码, E)								
0x8142	R	Gesture ID (Third Byte, ASCII 码, S)								
0x8143	R	Gesture 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 0xBDA8)								
0x814E	R	缓冲区 1 (起始位置 0x8150) 协议类型 (0x01: 单笔手势, 0x02: 多笔手势)				缓冲区 2 (起始位置 0xBDA8) 协议类型 (0x02: 上报所有点, 0x03: 上报特征点)				
0x814F	R	缓冲区 1 数据个数								
0x8150~0x819E	R	缓冲区 1 内容								
0x8151	R	Check Sum (地址位置根据缓冲区长度变化) 校验起始地址: 0x814C 校验长度 (u8): Gesture Data length (缓冲区长度) + 4								
0x819F	R	手势轨迹坐标 16 位 Check Sum (low byte) (地址固定) 校验起始地址: 0xBDA8 校验长度 (u16): Gesture Points Count * 2								
0x81A0	R	手势轨迹坐标 16 位 Check Sum (high byte) (地址固定)								

(手势坐标信息)

Addr	Access	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
0xBDA8	R	Gesture point 1 x coordinate (low byte)							
0xBDA9	R	Gesture point 1 x coordinate (high byte)							
0xBDAA	R	Gesture point 1 y coordinate (low byte)							
0xBDAB	R	Gesture point 1 y coordinate (high byte)							
0xBDAC~0xBEA7	R	Gesture point 2~64 coordinate (坐标个数为 0x814D 的值)							

3.5 GT917S 的命令状态寄存器

Addr	Access	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
0x81A8	R	GT917S_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	GT917S_Status_Bak: GT917S_Status 的备份							

该区域的目的是可以通过查询 GT917S 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 128 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) 只有在 **Receive mode** 模式下，才可以往该区域写入数据，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，GT917S 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 运行在 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	HotKnotRev Status	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，注意仅随在数据之后，并不固定在此位置							

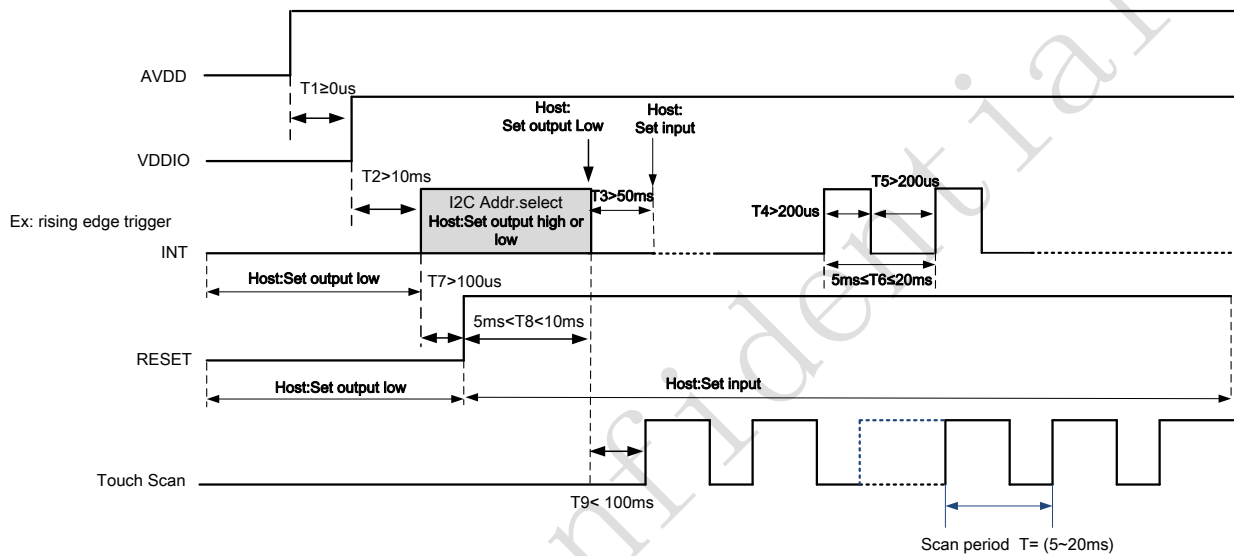
- 1) **0x8B00.bit7:** Buffer status 为 1 时，接收数据缓冲区数据已准备好，可读取。同时对应的 HotKnotRevStatus 状态为 0x03。
- 2) 该区域的数据只有在 Receive mode 模式下才有效。buffer status 为 1 时，表明接收数据缓冲区数据已准备好，可读取。
- 3) DataLength 必须小于等于 128。
- 4) 支持的数据包长度为：2、4、8、16、32、64、128（单位为 Byte）。

4. 上电初始化与寄存器动态修改

4.1 GT917S 上电时序

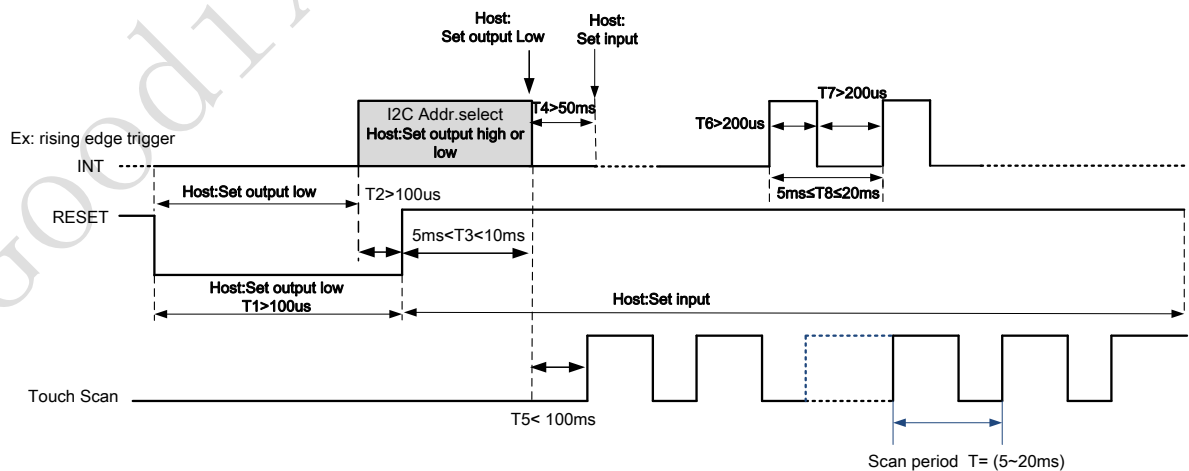
主机上电后，需要控制 GT917S 的 AVDD、VDDIO、INT、Reset 等脚位，控制时序请遵从如下时序

图：



INT T2 时间后，主控是要输出高，还是低，取决于主机要用何 I²C 从设备地址与 GT917S 芯片通信，若用地址 0x28/0x29，则输出高；若用地址 0xBA/0xBB，则输出低。

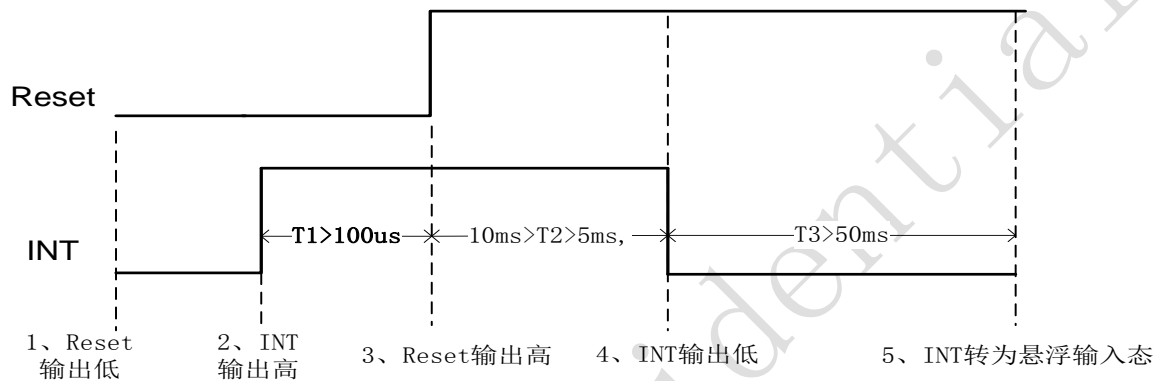
主机复位 GT917S 的控制时序如下：



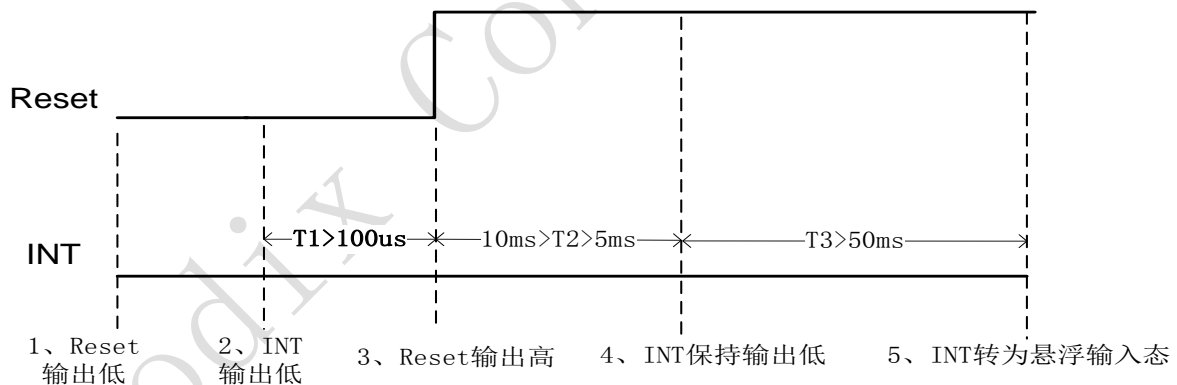
4.2 上电或复位 I2C 地址选择

GT917S 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 GT917S 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 GT917S
0xFF	DLE，无需处理
Others	保留，无需处理

2) 如何响应“0x01”请求

- a) 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。
- b) 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）。
- c) 通过 I2C 将 0x8044 写 0，完成“0x01”请求的响应。

3) 如何响应“0x03”请求

- a) 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。
- b) 按照复位时序对 GT917S 进行复位，完成“0x03”请求。

4.5 寄存器动态修改

GT917S 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。

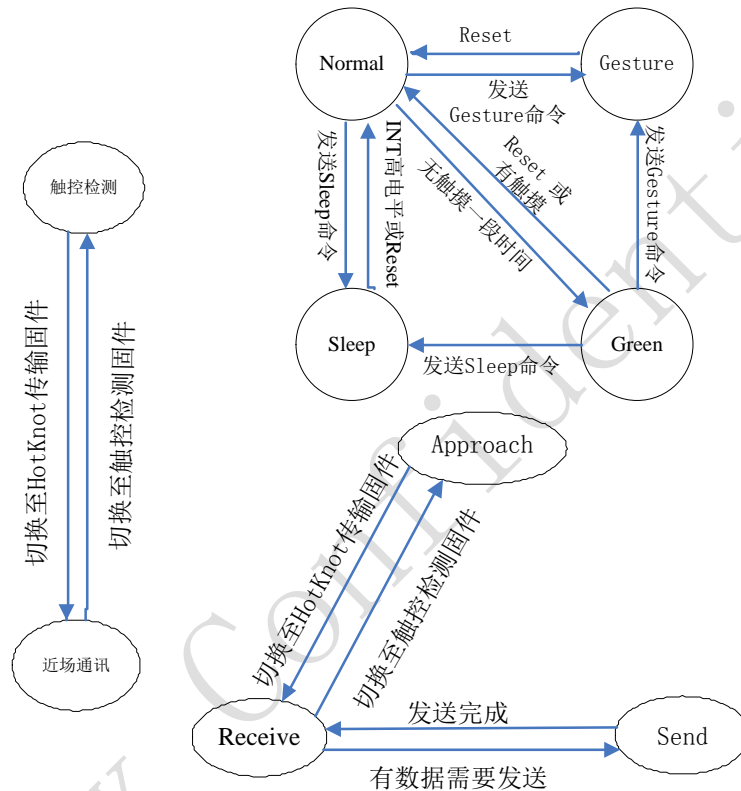
采用中断读取方式，触发中断后按上述轮询过程读取坐标。

GT917S 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 GT917S 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 GT917S 会一直打 INT 脉冲。

6. 工作模式切换

GT917S 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

GT917S 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

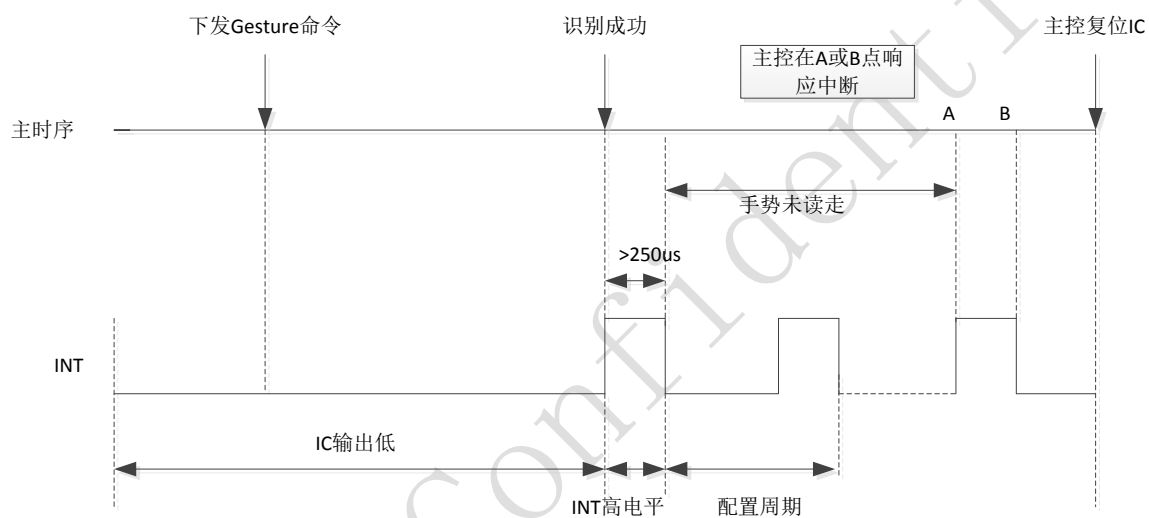
Normal mode 状态下，一段时间无触摸事件发生，GT917S 将自动转入 Green mode，以降低功耗。GT917S 无触摸自动进入 Green mode 的时间可通过配置信息设置，范围为 0~14s，步进为 1s，Green Mode 可通过配置关闭。在 Green mode 下，GT917S 扫描周期约为 40ms，若检测到有触摸动作发生，自动进入 Normal mode。

6.3 Gesture Mode

若主 CPU 通过 I2C 命令，GT917S 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，GT917S 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）：

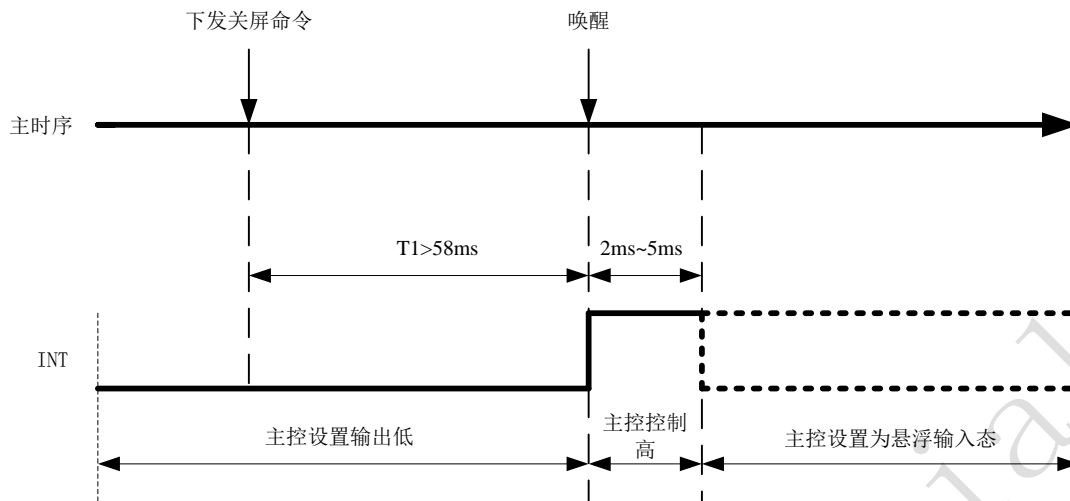


6.4 Sleep Mode

主 CPU 通过 I2C 命令，使 GT917S 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 GT917S 睡眠时的 INT 输入+上（下）拉能够得到对应的高（低）电平。

当需要 GT917S 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 GT917S 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 GT917S 运行在 Normal mode 或 Green mode 时,主 CPU 通过下发 0x20 或 0x21 命令,使 GT917S 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 GT917S 运行在 Approach mode 时,主 CPU 获取到 GT917S 检测到可通讯终端,主 CPU 再下发 HotKnot 传输固件使 GT917S 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 GT917S 运行在 Receive mode 时，主 CPU 将待发数据发送至发送缓冲区，GT917S 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。

Goodix Confidential

7. Gesture 模式驱动修改

7.1 灭屏后进入 Gesture 模式

- a) 按电源键（或其他按键）关屏时，往 0x8040~0x8042 下发命令 0x08, 0x00, 0xF8;
- b) 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;
- c) 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

- a) 按电源键（或其他按键）关屏时，往 0x8040~0x8042 下发命令 0x05, 0x00, 0xFB;
- b) 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;
- c) 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。

7.4 建议可与 IR 配合

如果可以用 IR 来配合，灭屏时当 IR 检测到有物体遮挡，可进入原 Sleep 模式，使耗电更少；检测无遮挡则进入手势唤醒模式，进入不同模式的方法同上所述（需复位再下发命令）。

8. Gesture 模式坐标读取及校验

8.1 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：

bit7~bit4（辅助信息）：

0x00: NULL;

0x01: Reserved;

0x02: 多笔手势断点位置。

bit3~bit0（手势坐标信息）：

0x00: NULL;

0x01: Reserved;

0x02: 手势触摸轨迹点（每隔相同距离获取的手指触摸位置）；

0x03: 手势触摸特征轨迹点（根据手势类型抽取出的触摸特征轨迹点，例如：W 的特征轨迹点为起点、三个转折点、终点）。

手势辅助信息：主控读取 0x814F 寄存器，获取辅助信息长度，然后以这个长度读取 0x8150~0x81A0 寄存器，可以获取到手势的一些辅助信息。

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 0xBDA8 ~ 0xBEA7 寄存器，通过这些信息可以描绘出用户真实触摸轨迹。

8.2 坐标及轨迹信息校验

为了加强系统的可靠性，与主控交互的数据均需采用校验的方式，如果是主控读取，需采用多次读取的方式保证可靠性。

校验方式分两种，一种是对手势极值点坐标上报时的校验，一种是对手势完整轨迹坐标点上报时的校

验。当主控收到中断需要来读取手势信息时，先判断 0x814E 最高位是否为 1，为 1 则表示需要校验功能，按下面的两种方式进行校验；否则不需要校验功能，直接按照 8.1 来读取坐标。

a) 手势极值点坐标上报校验（采用 8 位的校验方式）

从 0x814C 地址开始，连续读取（缓冲区长度+5）个 u8 的数据，相加为 0 表示校验通过，否则校验不通过，需重新读取一遍；若仍错，丢弃本次数据，同时清除 0x814C（FW 收到此标记清除，回到重新识别触摸的状态）。

b) 手势轨迹上报校验（采用 16 位校验方式）

从 0xBDA8 地址开始，连续读取（手势触摸点个数*2）个 u16 的数据，再读起始地址 0x819F 的一个十六位校验和数据，相加为 0 表示校验通过，否则校验不通过，需重新读取一遍；若仍错，丢弃本次数据，同时清除 0x814C（FW 收到此标记清除，回到重新识别触摸的状态）。

注意：0x819F 是校验和的低位，0x81A0 是校验和高位，0x814E 最高位置 1，表示有校验功能。

Goodix Confidential

9. 版本修订记录

文件版本	修订日期	修订
V0.1	2019-08-26	预发布版

Goodix Confidential