

GT9147 编程指南文件

(适用于 1030 以上版本软件)

目 录

一、接口说明	2
二、通信时序	2
2.1 主机对 GT9147 进行写操作时序	2
2.2 主机对 GT9147 进行读操作时序	3
三、寄存器列表	3
3.1 实时命令 (Write only)	3
3.2 配置信息 (R/W)	4
3.3 坐标信息	13
3.4 GT9147 的命令状态寄存器	17
3.5 Hotknot 的状态寄存器	18
3.6 Hotknot 的发送缓冲区	19
3.7 Hotknot 的接收缓冲区	20
3.8 Hotknot 注意事项	22
四、上电初始化与寄存器动态修改	22
4.1 GT9147 上电时序	22
4.2 上电或复位 I2C 地址选择	22
4.3 上电发送配置信息	23
4.4 寄存器动态修改	23
五、坐标读取	23
六、工作模式切换	24
七、版本修订记录	26

一、接口说明

GT9147 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，主控端必须设为悬浮态，取消内部上下拉功能；主机通过输出高、低来控制 GT9147 的 RESET 口为高或低。为保证可靠复位，建议 RESET 脚输出低 100 μ s 以上。

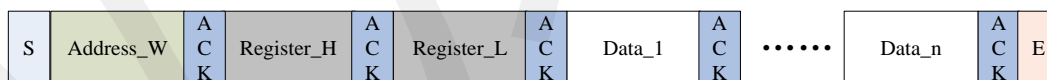
GT9147 与主机通信采用标准 I²C 通信，最高速率可以支持至 400Kbps。当主机采用 200 Kbps 以上的通信速率时，需要特别注意 I²C 的外部上拉电阻阻值，以保证 SCL、SDA 边沿足够陡峭。GT9147 在通信中始终作为从设备，其 I²C 设备地址由 7 位设备地址加 1 位读写控制位组成，高 7 位为地址，bit 0 为读写控制位。GT9147 有两个从设备地址可供选择，如下表：

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“上电初始化与 I2C 地址选择”一章。

二、通信时序

2.1 主机对 GT9147 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。

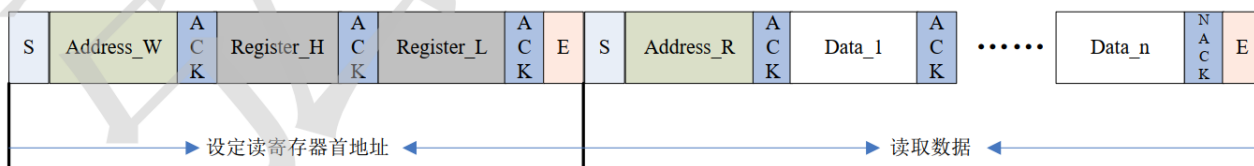
Data_1 至 Data_n：数据字节 1—n。

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，GT9147 自动将其往高地址顺序存储。

2.2 主机对 GT9147 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，GT9147 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。

三、寄存器列表

3.1 实时命令（Write only）

Addr	Name	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
0x8040	Command	0: 读坐标状态 1、2: 差值原始值 3: 基准更新（内部测试） 4: 基准校准（内部测试） 5: 关屏 6: 进入充电模式 7: 退出充电模式 8: 切换手势唤醒固件 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 0x23:进入主机传输模式 0x28: 退出从机检测模式 0x29: 退出主机接近检测模式 0x2A: 退出数据传输模式 0xAA:ESD 保护机制使用，由驱动定时写入 0xAA 并定时读取检查 其余值无效							
0x8041	ESD_Check	ESD 保护机制使用，在初始化时清零，之后由驱动写入 0xAA 并定时读取检查							
0x8042	Proximity_En	接近感应开关							
0x8043	NC	Reserved							
0x8044	NC	Reserved							
0x8045	NC								
0x8046	Command_Check	下发 0x20~0x2F 间的命令，需将命令先写此处，再写 0x8040，命令才会生效							

部分寄存器补充说明如下:

[0x8040] Command

HotKnot 的命令处在区间 0x20~0x2F,命令执行存在优先级关系。具体优先级关系如下:从接近检测的优先级最低,主接近检测的优先级高。当在低优先级状态下运行时,收到高优先级的命令,高优先级命令立即执行,低优先级暂停运行,低优先级等待高优先级退出后再次运行。例如当前处于只进行触控检测的状态,收到 0x21 命令,执行 0x21 任务,收到 0x20 后,仍然执行 0x21 任务,收到 0x29 后,才执行 0x20 命令;再收到 0x28 命令,则只做触控检测。

[0x8046] Command_Check

对于范围在 0x20~0x2F 间的 HotKnot 命令,为了增强抗 ESD 能力,需要先将命令写往 0x8046 处,再将命令写往 0x8040 处,命令才能生效。非此范围的命令,只需写向 0x8040 就会生效。

3.2 配置信息 (R/W)

寄存器	Config Data	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
0x8047	Config_Version	配置文件的版本号							
0x8048	X Output Max (Low Byte)	X 坐标输出最大值							
0x8049	X Output Max (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：1~5			
0x804D	Module_Switch1	Water_SpeedLimit_En	Water_LargeRestrain_En	Stretch_rank		X2Y (X,Y 坐标交换)	Sito (软件降噪)	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	
0x804E	Module_Switch2	STP_SE 不同图案特殊处理		First_Filter_Dis	Water_SITO	Water_Proof_Disable	Reserved	Key_Noise	Touch_Key
0x804F	Shake_Count	手指松开去抖次数				手指按下去抖次数			
0x8050	Filter	First_Filter		Normal_Filter(原始坐标窗口滤波值,系数为 4)					
0x8051	Large_Touch	大面积触摸点个数							
0x8052	Noise_	H 形黄光图案削底系数 N				噪声消除值（系数为 1,0-15 有效）			

	Reduction	(0 关闭, 系数为 1/(N+5))							
0x8053	Screen_Touch_Level	屏上触摸点从无到有的阈值							
0x8054	Screen_Leave_Level	屏上触摸点从有到无的阈值							
0x8055	Low_Power_Control	Reserved				进低功耗时间(0~15s)			
0x8056	Refresh_Rate	Reserved				坐标上报率(周期为 5+N ms)			
0x8057	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)							
0x8058	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)							
0x8059	Gesture_Switch1	左滑	上滑	右滑	W	O	M	E	C
0x805A	Gesture_Switch2	滑动最后驱动	Z	S	^	>	V	双击	下滑
0x805B	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)			
0x805C		左边框的空白区 (以 32 为系数)				右边框的空白区 (以 32 为系数)			
0x805D	Mini_Filter	Reserved				划线过程中的小 filter 设置, 配 0 默认为 4			
0x805E	Stretch_R0	拉伸区间 1 系数							
0x805F	Stretch_R1	拉伸区间 2 系数							
0x8060	Stretch_R2	拉伸区间 3 系数							
0x8061	Stretch_RM	各拉伸区间基数							
0x8062	Drv_GroupA_Num	All_Driving	Reserved		Driver_Group_A_number				
0x8063	Drv_GroupB_Num	Reserved		Dual_Freq	Driver_Group_B_number				
0x8064	Sensor_Num	Sensor_Group_B_Number				Sensor_Group_A_Number			
0x8065	FreqA_factor	驱动组 A 的驱动频率倍频系数 GroupA_Frequency = 倍频系数 * 基频							
0x8066	FreqB_factor	驱动组 B 的驱动频率倍频系数 GroupB_Frequency = 倍频系数 * 基频							
0x8067	Pannel_BitFreqL	驱动组 A、B 的基频(1526HZ<基频<14600Hz)							
0x8068	Pannel_BitFreqH								
0x8069	MODULE_SWITCH4			Driver_resersal	sensor_resersal			Approch_En	HotKnot_En
0x806A	Gesture_Long_Press_Time	手势长按进入 New_Green 时间							
0x806B	Pannel_Tx_Gain	Reserved			Pannel_Drv_output_R 4 档可调		Pannel_DAC_Gain 0: Gain 最大 7: Gain 最小		

0x806C	Pannel_Rx_Gain	Pannel_PGA_C	Pannel_PGA_R	Pannel_Rx_Vcml(4 档可调)	Pannel_PGA_Gain(8 档可调)				
0x806D	Pannel_Dump_Shift	手势唤醒原始值放大系数（2 的 N 次方）			屏原始值放大系数（2 的 N 次方）				
0x806E	Drv_Frame_Control	Reserved	SubFrame_DrvNum				Repeat_Num(采样累加次数)		
0x806F	S_FeedBack	S 型改善负反馈量(1/256 为单位)							
0x8070	Module_Switch3	Key_Restrain_Dis: 按键与屏体抑制失能开关, 默认为 0: 与之前版本兼容, 1: 按键与屏体不做抑制	Gesture_Hop_Dis: 手势唤醒关跳频省电	INT_Wakeup: 唤醒电平 默认为 0: 高电平唤醒, 1: 低电平唤醒	Check_Screen_Neg: 统计全屏负值更新, 默认为 0 关闭, 对于单层 H 型建议打开	Water_Single_Dis: 单指不进防水, 默认为 0, 单指拖尾严重的 TP 置 1	Water_Shape_En: 水状态下是否开启形变处理, 1: 开启, 0: 关闭	Y_Invert H 形黄光出线方式 0: 下出线 1: 上出线	Shape_En 形变处理
0x8071	Screen_Neg_Thres	全屏负值检测统计阈值, 单位为 5, 为 0 时默认为 leavelevel				全屏负值检测统计数量（0-15, 配 0 默认为 5） 若全屏小于（-阈值/2）的负差值个数+此值*2 ≥ g_CellTotal/2，则更新基准			
0x8072	shape_control	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 时默认关闭。				Touch 判量的倍数, 配置越大, 消减越多, 建议配置 1~4。配 0 则进入重按时不处理，进入超重按时默认为 2 倍。			
0x8073	EDGE_COMPLEMENT_THRES	边缘补值阈值（0~255, 配置越大补值条件越严格）							
0x8074	EDGE_COMPLEMENT_X	X 方向补值系数（0~255, 配 0 不补值, 配置越大补值越多）							
0x8075	EDGE_COMPLEMENT_Y	Y 方向补值系数（0~255, 配 0 不补值, 配置越大补值越多）							
0x8076	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。							
0x8077	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 10。							
0x8078	Charging_Level_Up	主控下发充电器命令后，IC 进入充电器状态，把 Touch_Level 和 Leave_Level 提高 充电模式下使用的 level 值=原配置的 Level 值+配置值 配置 0 时，不提高阈值，原配置的 Level 值							

0x8079	Gesture_Control	双击唤醒非法时间（100ms 为单位，配置为 0 则为 1.5s）				GestureDrv_PGA_Gain(8 档可调)			
0x807A	Freq_Hopping_Start	跳频范围的起点频率(以 BitFreq 为单位)							
0x807B	Freq_Hopping_End	跳频范围的终点频率(以 BitFreq 为单位)							
0x807C	Noise_Detect_Times	Detect_Stay_Times (一次噪声检测中每个频率点上检测次数,建议 2)		Detect_Confirm_Times (多次噪声检测后确定噪声量,1-63 有效，建议 20)					
0x807D	Hopping_Flag	Hopping_En	Range_Ext	Dis_Force_Ref	Delay_Hopping	Reserved			
0x807E	Hoppging_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)			
0x807F	HotKnotNoiseMap	Reserved	Reserved	200K	250K	300K	350K	400K	450K
0x8080	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							
0x8081	NC	Reserved							
0x8082	Hopping_Sensor_Group	跳频 Noise 侦测分段数（建议分 4 段）							
0x8083	Hopping_seg1_Normalize	Seg1 Normalize 系数（采样值*N/128，得到 Rawdata）							
0x8084	Hopping_seg1_Factor	Seg1 中心点 Factor							
0x8085	Main_Clock_Adjust	微调主频配置，范围-7~+8							
0x8086	Hopping_seg2_Normalize	Seg2 Normalize 系数（采样值*N/128，得到 Rawdata）							
0x8087	Hopping_seg2_Factor	Seg2 中心点 Factor							
0x8088	NC	Reserved							

0x8089	Hopping_seg3_Normalize	Seg3 Normalize 系数（采样值*N/128，得到 Rawdata）	
0x808A	Hopping_seg3_Factor	Seg3 中心点 Factor	
0x808B	NC	Reserved	
0x808C	Hopping_seg4_Normalize	Seg4 Normalize 系数（采样值*N/128，得到 Rawdata）	
0x808D	Hopping_seg4_Factor	Seg4 中心点 Factor	
0x808E	NC	Reserved	
0x808F	Hopping_seg5_Normalize	Seg5 Normalize 系数（采样值*N/128，得到 Rawdata）	
0x8090	Hopping_seg5_Factor	Seg5 中心点 Factor	
0x8091	NC	Reserved	
0x8092	Hopping_seg6_Normalize	Seg6 Normalize 系数（采样值*N/128，得到 Rawdata）	
0x8093	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)	
0x8094	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)	
0x8095	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)	
0x8096	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)	
0x8097	Key_Area	长按更新时间(1~15s) ; 长按更新时间为 1~15s，配置为 0 时取消 长按更新	按键有效区间设置(单侧):0-15 有效
0x8098	Key_Touch_Level	触摸按键按键阈值	
0x8099	Key_Leave_Level	触摸按键松键阈值	
0x809A	Key_Sens	KeySens_1(按键 1 灵敏度系数)	KeySens_2 (按键 2 灵敏度系数)
0x809B	Key_Sens	KeySens_3(按键 3 灵敏度系数)	KeySens_4 (按键 4 灵敏度系数)
0x809C	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 600ms 抑制	独立按键邻键抑制参数(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
0x809D	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）
0x809E	Proximity_Valid_Time	接近感应生效时间（以 1 个主循环的周期计时）	

0x809F	Proximity_Press_Time1	手指按下最短时间（以 1 个主循环的周期计时）			
0x80A0	Proximity_Press_Time2	手指按下最长时间（以 1 个主循环的周期计时）			
0x80A1	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）			
0x80A2	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）
0x80A3	Proximity_Sens_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）
0x80A4	Proximity_Touch_Level	设定值×10=接近感生效应阈值			
0x80A5	Proximity_Leave_Level	设定值×10=接近感应无效阈值			
0x80A6	Proximity_Sample_Add_Times	采样值累加次数			
0x80A7	Proximity_Sample_Dec_ValL	采样值减此值（16 位）后再累加，低字节			
0x80A8	Proximity_Sample_Dec_ValH	采样值减此值（16 位）后再累加，高字节			
0x80A9	Proximity_Leave_Shake_Count	退出接近感应去抖次数			
0x80AA	Data_Threshold	Data_NoiseThreshold 数据传输的阈值			
0x80AB	Pxy_Threshold	Pxy_NoiseThreshold 接近检测的阈值			
0x80AC	Dump_Shift	Reserved		Rx_Self	原始值放大系数（2 的 N 次方）
0x80AD	Rx_Gain	Pannel_PGA_C	Pannel_PGA_R	Reserved	Pannel_PGA_Gain(8 档可调)
0x80AE	FREQ_GAIN0	400K 增益调准,调整量为 N/8			450K 增益调准,调整量为 N/8
0x80AF	FREQ_GAIN1	300K 增益调准,调整量为 N/8			350K 增益调准,调整量为 N/8
0x80B0	FREQ_GAIN2	200K 增益调准,调整量为 N/8			250K 增益调准,调整量为 N/8
0x80B1	FREQ_GAIN3	Reserved			150K 增益调准,调整量为 N/8
0x80B2	Gesture_Refresh_Rate	手势唤醒坐标上报率(周期为 5+ms)			
0x80B3	Combine_Dis	手势唤醒合框距离			合框距离，0~15 可配。合点距离为配置值的 2 倍开平方 pitch。为兼容老配置，配 0 默认与之前的处理一样，合点距离为 2 pitch
0x80B4	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 倍开平方 pitch。为兼容老配置，配 0 默认与之前处理一样，拆点距离为 12 的平方根 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 倍开平方 pitch。为兼容老配置，配 0 默认与之前处理一样，正常触摸拆点距离为 7 的平方根 pitch
0x80B5	Gesture_Touch_L	手势唤醒触摸阈值			

	level		
0x80B6	NewGreenWake UpLevel	手势唤醒 NewGreen 唤醒阈值	
0x80B7 ~ 0x80C4	Sensor_CH0 ~ Sensor_CH13	ITO Sensor 对应的芯片通道号	
0x80C5~ 0x80D4	NC	Reserved	
0x80D5~ 0x80EE	Driver_CH0~ Driver_CH25	ITO Driver 对应的芯片通道号	
0x80EF~ 0x80F1	NC	Reserved	
0x80F2	Driver_Gain 0~1	通道 1 调整系数 N, 调整量为 $(N+1)/16$ N 等于 15 时无效	通道 0 调整系数 N, 调整量为 $(N+1)/16$ N 等于 15 时无效
0x80F3	Driver_Gain 2~3	通道 3 调整系数 N, 调整量为 $(N+1)/16$, N 等于 15 时无效	通道 2 调整系数 N, 调整量为 $(N+1)/16$, N 等于 15 时无效
0x80F4	Driver_Gain 4~5	通道 5 调整系数 N, 调整量为 $(N+1)/16$, N 等于 15 时无效	通道 4 调整系数 N, 调整量为 $(N+1)/16$, N 等于 15 时无效
0x80F5	Driver_Gain 6~7	通道 7 调整系数 N, 调整量为 $(N+1)/16$, N 等于 15 时无效	通道 6 调整系数 N, 调整量为 $(N+1)/16$, N 等于 15 时无效
0x80F6	Driver_Gain 8~9	通道 9 调整系数 N, 调整量为 $(N+1)/16$, N 等于 15 时无效	通道 8 调整系数 N, 调整量为 $(N+1)/16$, N 等于 15 时无效
0x80F7	Driver_Gain 10~11	通道 11 调整系数 N, 调整量为 $(N+1)/16$, N 等于 15 时无效	通道 10 调整系数 N, 调整量为 $(N+1)/16$, N 等于 15 时无效
0x80F8	Driver_Gain 12~13	通道 13 调整系数 N, 调整量为 $(N+1)/16$, N 等于 15 时无效	通道 12 调整系数 N, 调整量为 $(N+1)/16$, N 等于 15 时无效
0x80F9	Driver_Gain 14~15	通道 15 调整系数 N, 调整量为 $(N+1)/16$, N 等于 15 时无效	通道 14 调整系数 N, 调整量为 $(N+1)/16$, N 等于 15 时无效
0x80FA	Driver_Gain 16~17	通道 17 调整系数 N, 调整量为 $(N+1)/16$, N 等于 15 时无效	通道 16 调整系数 N, 调整量为 $(N+1)/16$, N 等于 15 时无效
0x80FB	Driver_Gain 18~19	通道 19 调整系数 N, 调整量为 $(N+1)/16$, N 等于 15 时无效	通道 18 调整系数 N, 调整量为 $(N+1)/16$, N 等于 15 时无效
0x80FC	Driver_Gain 20~21	通道 21 调整系数 N, 调整量为 $(N+1)/16$, N 等于 15 时无效	通道 20 调整系数 N, 调整量为 $(N+1)/16$, N 等于 15 时无效
0x80FD	Driver_Gain 22~23	通道 23 调整系数 N, 调整量为 $(N+1)/16$, N 等于 15 时无效	通道 22 调整系数 N, 调整量为 $(N+1)/16$, N 等于 15 时无效
0x80FE	Driver_Gain 24~25	通道 25 调整系数 N, 调整量为 $(N+1)/16$, N 等于 15 时无效	通道 24 调整系数 N, 调整量为 $(N+1)/16$, N 等于 15 时无效
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 0x01)	

部分寄存器补充说明如下:

[0x804D] Module_Switch1

Bit5-Bit4: Stretch_rank, 拉伸方式

00,01,02: 弱拉伸 0.4P

03: 支持自定义拉伸系数, 必须配置 Stretch_R0、Stretch_R1、Stretch_R2、Stretch_RM

[0x804E] Module_Switch2

Bit7-Bit6: STP_SE, 不同图案特殊处理

00: 不使能

01,10: Reserved

11: 黄光 H 型图案

Bit4: Water_SITO: 水状态判断是否需要先 SITO(0,不要; 1,需要)。

Bit3: Water_Proof_Disable: 置 1 时关掉防水处理, 清 0 时开启防水处理。

Bit2-Bit1: 保留。

Bit0: Touch_key: 触摸按键, 置 1 表示有按键, 清 0 无按键。

[0x805B-0x805C] Space

屏的 4 个边缘的空白区配置, 用于在 ITO 超出实际可视区时对边缘进行裁剪。可设范围 0~15 (表示裁剪 $N \times 32$ 个原始坐标点)。其中 0 表示无裁剪, 最大裁剪范围为 $15 \times 32 = 480$ 个原始坐标点 (一个 Pitch 有 512 个原始坐标点, 若裁剪需要超过一个 Pitch, 直接在配置中先减少一个 Pitch 即可)。

[0x8070] Module_Switch3

Bit7: Key_Restrain_Dis, 按键与屏体抑制失能开关

Bit6: Gesture_Hop_Dis, 手势唤醒关跳频省电

Bit5: INT_Wakeup: 唤醒电平, 默认为 0: 高电平唤醒, 1: 低电平唤醒

Bit4: Check_Screen_Neg: 统计全屏负值更新, 默认为 0 关闭, 对于单层 H 型建议打开

Bit3: Water_Single_Dis, 单指不进防水, 默认为 0, 单指拖尾严重的 TP 置 1。

Bit2: Water_Shape_En: 水状态下是否开启形变处理, 置 1 开启, 置 0 关闭。

Bit1: 保留。

Bit0: Shape_En: 形变处理, 置 1 开启, 清 0 关闭。

[0x807C] Noise_Detect_Times

Bit7~6: Detect_Stay_Times, 一次噪声检测中每个频率点上检测次数, 通常设置为 2。

Bit5~0: Detect_Confirm_Times, 多次噪声检测后确定噪声量,通常设置为 15~20。

[0x807D] Hopping_Flag

Bit7: Hopping_En,跳频使能位, 1 使能, 0 禁止。

Bit6: Range_Ext,跳频范围扩展标志, 固定为 1。

Bit5: Dis_Force_Ref,清 0 表示跳频后强制更新基准, 置 1 表示跳频后不强制更新基准。

Bit4: Delay_Hopping, 置 1 表示有手指按压不跳频, 清 0 表示有手指按压跳频。

Bit3~0: Detect_Time_Out, 噪声检测超时时间, 以秒为单位。

[0x807E] Hopping_Threshold

Bit3~0: 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。

[0x809A-0x809B] Key_Sens

4 个独立按键的灵敏度系数配置, 可以设置为 0~15 共 16 级, 越大则灵敏度越高。仅对独立按键有效, 主要为了避免独立按键在设计时节点电容较容易产生偏差而导致按键灵敏度不一样的问题。

[0x809C] Key_Restrain

Bit3~0: 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

[0x8069] Hotknot_Switch

Bit1: HotKnot 接近检测功能模块使能位。1 使能, 0 禁止。只有在使能的情况下, 发送 0x20、0x21 等命令才有效, 才能进行接近检测。

Bit0: HotKnot 数据传输功能模块使能位。目前该位无需配置。

[0x80AA] Data_Threshold

HotKnot 是使用频率来代表数据, 两个 HotKnot 终端通过收发约定频率的信号实现数据的通讯。

Data_Threshold 用来表示有无信号的一个阈值。一般关闭 LCD, 无信号的情况下, HotKnot 接收到白噪声的大小在 5 以内。推荐设置应该比白噪声水平大 5, 但最小应该在 10 以上。

[0x80AB] Pxy_Threshold

HotKnot 接近检测的功能是在 LCD 未关闭时启用的, 采用差分的过滤干扰的方法。该阈值为差分阈值。噪声起伏大时, 差分阈值相应设大。可根据需两个 HotKnot 需贴合面积的大小, 距离的远近适当调准。建议该值设置在 15 以上, 推荐值 20。

[0x80AC] Dump_Shift

该 Dump_Shift 适用于 HotKnot 的原始值放大，一般配置在 2~4 之间。

[0x80AD Rx_Gain

该 Rx_Gain 适用于 HotKnot 的接收硬件设置，机理同 TP 的 Rx_Gain 设置。

[0x80AE-0x80B1] Freq_Gain0~3

HotKnot 使用频率的信号软增益系数。HotKnot 使用的 7 个频率是 150KHz~450KHz，50KHz 的步进值。

根据各个频点实际采样 rawdata 调整软增益，使各个频点数据一致性较好，缩小不同频率间信号的差异。

3.3 坐标信息

Addr	Access	bit7		bit6	bit5	bit4	bit3	bit2	bit1	bit0
0x8140	R	Product ID (first Byte, ASCII 码)								
0x8141	R	Product ID (second Byte, ASCII 码)								
0x8142	R	Product ID (third Byte, ASCII 码)								
0x8143	R	Product ID (forth Byte, ASCII 码)								
0x8144	R	Firmware version (HEX.low byte)								
0x8145	R	Firmware version (HEX.high byte)								
0x8146	R	x coordinate resolution (low byte)								
0x8147	R	x coordinate resolution (high byte)								
0x8148	R	y coordinate resolution (low byte)								
0x8149	R	y coordinate resolution (high byte)								
0x814A	R	Vendor_id (当前模组选项信息)								
0x814B	R	Reserved								
0x814C	R	Reserved								
0x814D	R	Reserved								
0x814E	R/W	buffer status	large detect	Proximity Valid	HaveKey	number of touch points				
0x814F	R	track id 为 32，表明为接近检测的信号								
0x8150	R	PxyOk	Reserved							
0x8151	R	PxyOk	Reserved							
0x8152	R	Reserved								
0x8153	R	Reserved								
0x8154	R	Reserved								
0x8155	R	Reserved								
0x8156	R	Reserved								
0x8157	R	track id								
0x8158	R	point 1 x coordinate (low byte)								
0x8159	R	point 1 x coordinate (high byte)								
0x815A	R	point 1 y coordinate (low byte)								

0x815B	R	point 1 y coordinate (high byte)
0x815C	R	point 1 size (low byte)
0x815D	R	point 1 size (high byte)
0x815E	R	Reserved
0x815F	R	track id
0x8160	R	point 2 x coordinate (low byte)
0x8161	R	point 2 x coordinate (high byte)
0x8162	R	point 2 y coordinate (low byte)
0x8163	R	point 2 y coordinate (high byte)
0x8164	R	point 2 size (low byte)
0x8165	R	point 2 size (high byte)
0x8166	R	Reserved
0x8167	R	track id
0x8168	R	point 3 x coordinate (low byte)
0x8169	R	point 3 x coordinate (high byte)
0x816A	R	point 3 y coordinate (low byte)
0x816B	R	point 3 y coordinate (high byte)
0x816C	R	point 3 size (low byte)
0x816D	R	point 3 size (high byte)
0x816E	R	Reserved
0x816F	R	track id
0x8170	R	point 4 x coordinate (low byte)
0x8171	R	point 4 x coordinate (high byte)
0x8172	R	point 4 y coordinate (low byte)
0x8173	R	point 4 y coordinate (high byte)
0x8174	R	point 4 size (low byte)
0x8175	R	point 4 size (high byte)
0x8176	R	Reserved
0x8177	R	track id
0x8178	R	point 5 x coordinate (low byte)
0x8179	R	point 5 x coordinate (high byte)
0x817A	R	point 5 y coordinate (low byte)
0x817B	R	point 5 y coordinate (high byte)
0x817C	R	point 5 size (low byte)
0x817D	R	point 5 size (high byte)
0x817E	R	Reserved
0x817F	R	KeyValue

部分寄存器增补说明如下:

[0x814A] 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。

Bit5: Proximity Valid, 接近感应有效置 1。

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]

HotKnot 接近检测功能运行后，当检测到另一个具有 HotKnot 的终端靠近时，会以坐标的形式上报检测结果，因此 Number of touch points 会加 1。其 track id 固定为 32，并且 PxyOk 置 1。注意，位置是固定的，处于第一个坐标的位置。

[0x8177] KeyValue

按键值，KeyValue 的位置并不固定，而是跟在有效坐标的后面。例如无接近检测结果的情况下，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。

（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	Gesture Firmware version (HEX.low byte)							
0x8145	R	Gesture Firmware version (HEX.high byte)							
0x8146	R	x coordinate resolution (low byte)							
0x8147	R	x coordinate resolution (high byte)							
0x8148	R	y coordinate resolution (low byte)							
0x8149	R	y coordinate resolution (high byte)							
0x814A	R	Reserved							

0x814B	R/W	手势类型（字符 ASCII 码表示 0x21-0x7F），右滑（0xAA），左滑（0xBB），下滑（0xAB），上滑（0xBA），双击（0xCC）
0x814C	R	手势触摸点个数（坐标存放位置 0x9420）
0x814D	R	Gesture start point x coordinate（low byte）
0x814E	R	Gesture start point x coordinate（high byte）
0x814F	R	Gesture start point y coordinate（low byte）
0x8150	R	Gesture start point y coordinate（high byte）
0x8151	R	Gesture end point x coordinate（low byte）
0x8152	R	Gesture end point x coordinate（high byte）
0x8153	R	Gesture end point y coordinate（low byte）
0x8154	R	Gesture end point y coordinate（high byte）
0x8155	R	Gesture Width（low byte）
0x8156	R	Gesture Width（high byte）
0x8157	R	Gesture Height（low byte）
0x8158	R	Gesture Height（high byte）
0x8159	R	Gesture Mid X coor(low byte)
0x815A	R	Gesture Mid X coor(high byte)
0x815B	R	Gesture Mid Y coor(low byte)
0x815C	R	Gesture Mid Y coor(high byte)
0x815D	R	Gesture P1 X coor(low byte)
0x815E	R	Gesture P1 X coor(high byte)
0x815F	R	Gesture P1 Y coor(low byte)
0x8160	R	Gesture P1 Y coor(high byte)
0x8161	R	Gesture P2 X coor(low byte)
0x8162	R	Gesture P2 X coor(high byte)
0x8163	R	Gesture P2 Y coor(low byte)
0x8164	R	Gesture P2 Y coor(high byte)
0x8165	R	Gesture P3 X coor(low byte)
0x8166	R	Gesture P3 X coor(high byte)
0x8167	R	Gesture P3 Y coor(low byte)
0x8168	R	Gesture P3 Y coor(high byte)
0x8169	R	Gesture P4 X coor(low byte)
0x816A	R	Gesture P4 X coor(high byte)
0x816B	R	Gesture P4 Y coor(low byte)
0x816C	R	Gesture P4 Y coor(high byte)

（手势坐标信息）

Addr	Access	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
0x9420	R	Gesture point 1 x coordinate（low byte）							
0x9421	R	Gesture point 1 x coordinate（high byte）							
0x9422	R	Gesture point 1 y coordinate（low byte）							

0x9423	R	Gesture point 1 y coordinate (high byte)
0x9424~ 0x951F	R	Gesture point 2~64 coordinate (坐标个数为 0x814C 的值)

3.4 GT9147 的命令状态寄存器

Addr	Access	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
0x81A8	R	GT9147_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	GT9147_Status_Bak: GT9147_Status 的备份							

[0x81A8] GT9147_Status

0x00: 表示 GT9147 当前只进行触控检测, 不进行 HotKnot 的任何相关操作。

0x88: 在 HotKnot 接近检测功能模块使能情况下, 主控下发 0x20 命令, GT9147 进入 HotKnot 从机接近检测模式。该模式下, HotKnot 的接近检测与触控检测是相间进行, 当成功检测到主机靠近时, 会以坐标 (track id 为 32) 的形式上报接近检测结果。在此状态时, 主控下发 0x28 命令可退出此状态。

0x99: 在 HotKnot 接近检测功能模块使能情况下, 主控下发 0x21 命令, GT9147 进入 HotKnot 主机接近检测模式, HotKnot 的主机接近检测与触控检测是相间进行, 当成功检测到从机靠近时, 会以坐标 (track id 为 32) 的形式上报接近检测的结果。在此状态时, 主控下发 0x29 命令可让其退出此状态。

0xAA: 当成功检测到通讯对方的存在时, 主控下发 HotKnot 传输固件并运行, GT9147 进入 HotKnot 数据传输模式, 默认为接收数据检测状态 Receive mode, 该模式下 GT9147 不再执行触控检测相关的操作, 一直检测是否有数据从发送端发送过来, 当成功接收到一帧数据, GT9147 会以 INT 的形式通知主控处理。

0xBB: 当成功检测到通讯对方的存在时, 主控下发 HotKnot 传输固件并运行, GT9147 进入 HotKnot 数据传输模式, 默认为接收数据检测状态 Receive mode, 在该模式, 当发送缓冲区被正确刷新, 立即切换为发送数据模式 Send mode, 当将缓冲区的数据成功发出去后, GT9147 会以 INT 的形式通知主控处理, 主控处理完后, 模式自动切换为 Receive mode 并进行离开检测, 直至发送缓冲区再次被正确刷新。

当执行 HotKnot 相关操作时, 主控可以通过查询 GT9147_Status, 可判断之前下发的命令是否成功, 以决定是否重新发送。或者依据当前状态, 决定下发何种 HotKnot 命令。

[0x81A9] GT9147_Status_Bak

GT9147_Status 的备份，建议主控同时读取 GT9147_Status 和 GT9147_Status_Bak，当二者相同时，才认为状态有效，以降低 I2C 总线遭受干扰带来数据出错。

3.5 Hotknot 的状态寄存器

Addr	Access	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
0xAB10	R	SendStatus:发送状态寄存器							
0xAB11	R	RevStatus:接收状态寄存器							
0xAB12	R	SendStatusBak:发送状态寄存器备份							
0xAB13	R	RevStatusBak:接收状态寄存器备份							
...	R	NC 预留 11 个字节							
0xAB1F	R/W	当有事物需要 Host 来处理时,GT9147 会在此处写入 0xAA,以 INT 的方式通知主控,主控处理完事物, 写入一个非 0xAA 的数, GT9147 再往下执行,否则 GT9147 等待 2.5S 时间							

部分寄存器增补说明如下：

该区域必须是 GT9147 处于 Receive Mode 或 Send Mode 模式下，即 GT9147_Status 为 0xAA 或 0xBB 的情况下，读出来的数据才有效。

[0xAB10] SendStatus

该寄存器表示 Send Mode 情况下，发送的具体情况。

0x01:表明 GT9147 处于空闲模式下，当发完一帧数据，且无数据需要发送，GT9147 会立即自动切换到 Receive Mode 并进行离开检测，主控在此状态下发将待发送数据发送至 HotKnot 发送缓冲区。

0x02:正在发送数据状态，此时主控不能再修改发送缓冲区的数据。

0x03:发送缓冲区的数据成功发送完，GT9147 会以 INT 的形式通知主控处理，主控读取到此状态后，先往 0xAB1F 写入一个非 0xAA 的数，GT9147 会自动切换到空闲模式，进入 Receive Mode 并进行离开检测。

0x04:主控下发到发送缓冲区的数据校验通不过，即不正确，或者长度不对，GT9147 会以 INT 形式通知主控处理，主控读取到此状态后，须先往 0xAB1F 写入一个非 AA 的数，然后重新发送刚才下发的数据。

0x05:GT9147 发送完一帧数据，但发送失败，不会以 INT 的形式通知主控，GT9147 会自动重新发送本帧数据。

注意，GT9147 是没有发送失败的概念，当发送不成功，会自动重复发送，直至发送成功，因此发送失败需由主控设置超时或检测到离开状态等方法来实现。

0x07:GT9147 检测到接收方离开。主控获取到此状态后，可退出数据模式。当成功发送完一帧数据，

GT9147 会开启离开检测。GT9147 通过发送扫频序列是否收到回复的方式来判断通讯对方是否存在，1S 持续未检测到回复，认为对方已经离开。

[0xAB11] RevStatus

该寄存器表示 Receive Mode 情况下，接收数据的具体情况。

0x01:表明 GT9147 当前处于空闲模式下，检测是否有数据从发送端发送过来，但还尚未检测到有效信号。

0x02:表明 GT9147 已检测到起始信号，正在接收数据中。

0x03:表明 GT9147 成功接收到一帧数据，并已更新到接收缓冲区，GT9147 会以 INT 的形式通知主控处理，主控读取完接收缓冲区的数据后，须往 0xAB1F 写入一个非 AA 的数。

0x04:表明 GT9147 刚接收到一帧数据 CRC16 校验通不过，不会以 INT 的形式通知主控，无需主控处理，GT9147 会自动重新开始检测起始信号。

注意，GT9147 是没有接收失败的概念，当接收 CRC 校验失败或收到过长的空信号，会自动重新检测起始信号，直至接收成功，因此接收失败需由主控设置超时来实现。

0x07:GT9147 检测到发送方离开。主控获取到此状态后，可以退出数据模式。当成功接收完一帧数据，GT9147 会开启离开检测。GT9147 通过否收到发送方发来的扫频序列的方式来判断发送方是否存在，1S 持续未检测到信号，认为对方已经离开。

[0xAB12] SendStatusBak

SendStatus 的备份区，当 GT9147 以 INT 形式通知主控前，SendStatus 会将值赋给 SendStatusBak，主控在读取到 SendStatus 与 SendStatusBak 相同的情况下，SendStatus 的状态才是有意义的。若不相同，可延时 2ms 再读取。目的是为了增强抗 ESD 的能力。

[0xAB13] RevStatusBak

RevStatus 的备份区，当 GT9147 以 INT 形式通知主控前，RevStatus 会将值赋给 RevStatusBak，主控在读取到 SendStatus 与 SendStatusBak 相同的情况下，SendStatus 的状态才是有意义的。若不相同，可延时 2ms 再读取。

3.6 HotKnot 的发送缓冲区

Addr	Access	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
0xAC90	W	DataLength:有效数据的长度，需小于 129							
0x AC91	W	Data0							
0xAC92	W	Data1							

...	W	...
0xAD10	W	Data127
0xAD11	W	Checksum 和的补码校验
...	NC	Reserved
0xAD91	W	Data_Fresh 数据已更新标记(由主控写入标记 0xAA)

部分寄存器增补说明如下：

该区域必须是 GT9147 处于 Receive Mode 模式下，即 GT9147_Status 为 0xAA 的情况下，才能执行写入操作。否则会带来不可预测的结果。

[0xAC90] DataLength

HotKnot 支持一帧最长的数据为 128 Byte，DataLength 必须小于或等于 128，且必须为偶数长度。

[0xAD11] CheckSum

Checksum 的位置不固定，跟随在有效的数据后面，校验是从 0xAC90 开始计算的。例如 2 个数据时，其位置在 0xAC93。值为和的补码。

[0xAD91] Data_Fresh

主 CPU 在写入时，先写非 0xAD91 处数据，写好数据后，再往 0xAD91 写入 0xAA，即置上了刷新发送缓冲区的标志，GT9147 查询到此标志后，会先检查缓冲区内数据是否校验正确，长度是否符合规范，若正确，会立即启动发送，切换到 Send Mode，若不正确，会以 INT 的形式通知主控处理。

3.7 Hotknot 的接收缓冲区

Addr	Access	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
0xAE10	R/W	buffer status							
0xAE11	R/W	DataLength 有效数据的长度，小于 129							
0xAE12	R	Data0 第 1 个字节数据							
0xAE13	R	Data1 第 2 个字节数据							
...	R	...							
0xAE91	R	Data127 第 128 个字节数据							
0xAE92~ 0xAE93	R	Crc16Check 数据 CRC16 校验，注意跟随在数据在后，并不固定在此位置，大端模式							

部分寄存器增补说明如下：

该区域必须是 GT9147 处于 Receive Mode 模式下，且 GT9147_Status 为 0xAA 的情况下，RevStatus 的状态为 0x03，该区域的数据才有效。

[0xAE10]buffer status

bit7:buffer status,为 1 时，表明接收数据缓冲区数据已准备好，可读取。

[0x AE11]DataLength

有效数据的长度，该值不会大于 128。

[0xAE92~0xAE93] Crc16Check

数据的 CRC-CICTT 校验，大端存储模式。

校验机制说明：

对于长度为 n 的数据帧，CRC 校验结果是：n 个数据+长度的校验。例如，长度为 112，主控需要从 0xAE12 位置读取(112 byte 数据+2byte CRC16 校验)共 114 个字节。主控计算出“112 个数据+长度”的 CRC 校验，与位置在(0xAE12+112)处的 CRC 进行比较，若一致，校验通过，若不一致，校验不通过。注意，计算 CRC，长度是最后计算的，不是在最前面。

Crc16 计算方法参考代码,注意此处为大端模式：

```
#define FREQ_CRC_SEED  0x1021

//计算出 SrcData 中 length 个数据的 CRC16 的值
unsigned short  Crc16(unsigned char *SrcData,unsigned char length)
{
    unsigned short  crc=0xFFFF;
    unsigned char   i,j;
    unsigned char   value;
    bit flag;
    bit c15;

    for (i= 0; i < length; i++)
    {
        value=SrcData[i];
        for (j= 0; j < 8; j++)
        {
            flag = (value & 0x80);
            c15 = (crc & 0x8000);
            value <<= 1;
            crc <<= 1;
            if(c15^flag)
                crc ^= FREQ_CRC_SEED;
        }
    }
    return crc;
}
```

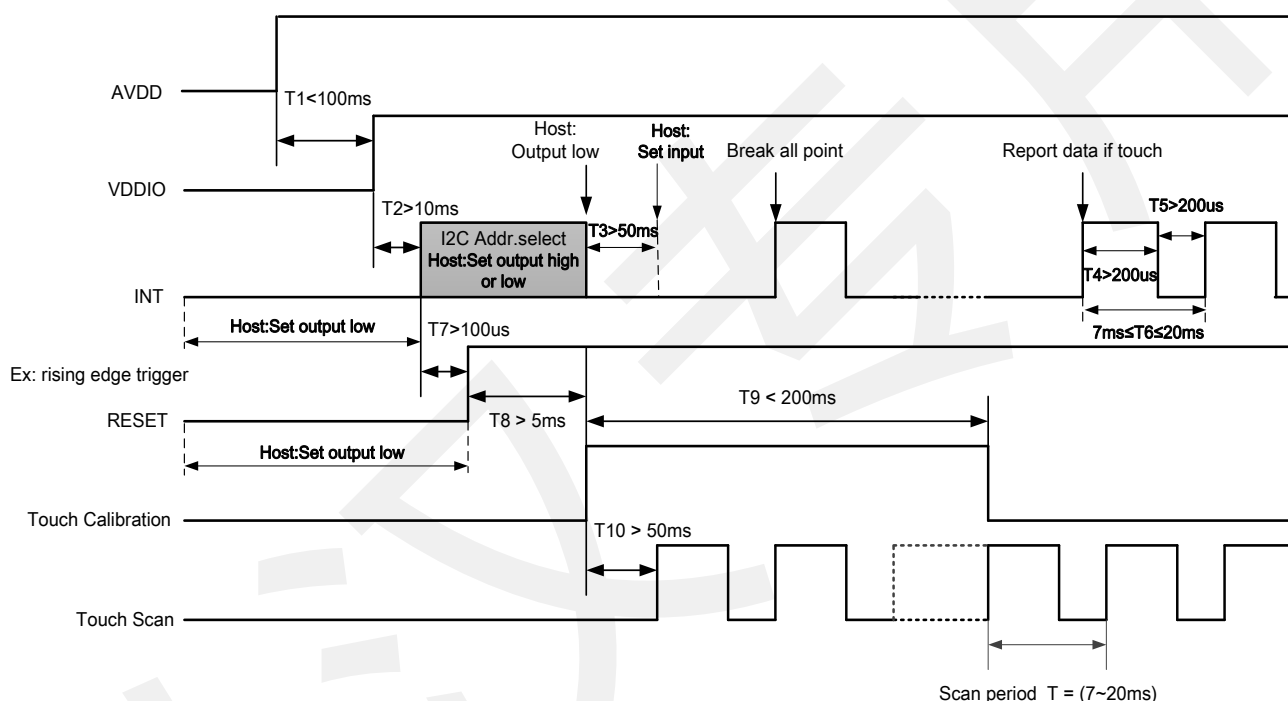
3.8 Hotknot 注意事项

Hotknot 传输系统是在接近检测成功后由驱动下发，如果发送方先于接收方 1S 完成 Hotknot 传输系统的下发，发送方会检测到接收方离开，导致 Hotknot 通信失败。因此，规定下发 Hotknot 传输系统时间控制在 800ms 以内，即 I2C 速率需要保证在 200Kbps 以上。

四、上电初始化与寄存器动态修改

4.1 GT9147 上电时序

主机上电后，需要控制 GT9147 的 AVDD、VDDIO、INT、Reset 等脚位，控制时序请遵从如下时序图：

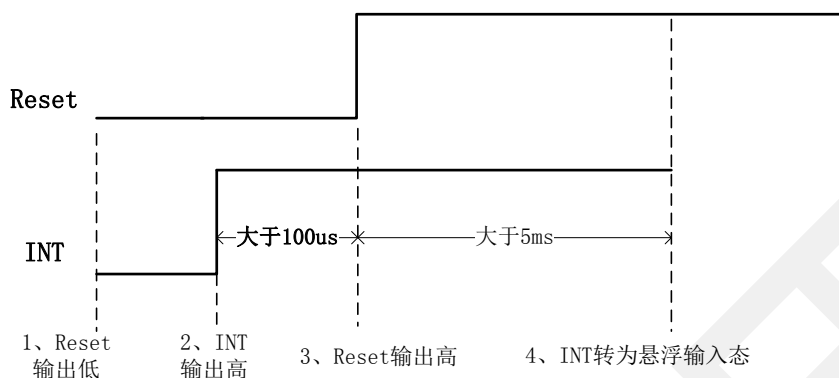


INT T2 时间后，主控是要输出高，还是低，取决于主机要用何 I2C 从设备地址与 GT9147 芯片通信，若用地址 0x28/0x29，则输出高；若用地址 0xBA/0xBB，则输出低。

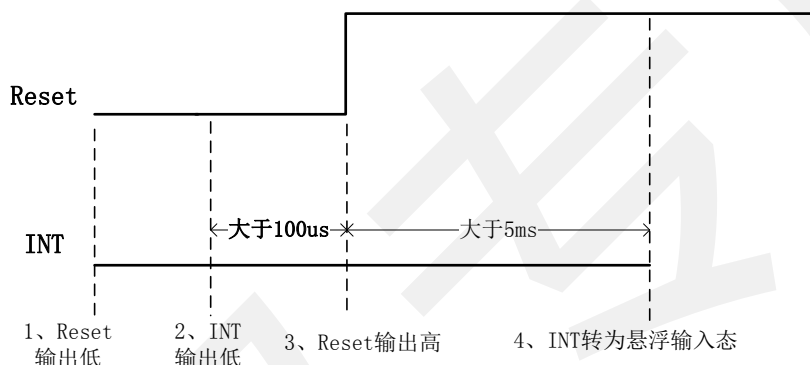
4.2 上电或复位 I2C 地址选择

GT9147 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 GT9147 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 再发送配置信息。

4.4 寄存器动态修改

GT9147 支持寄存器动态修改, 当按照第 2 节时序对配置区内 (0x8047—0x80FE) 任何寄存器修改时, 需要更新 Config_Chksum (0x80FF), 并在最后将 Config_Fresh (0x8100) 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。

五、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。

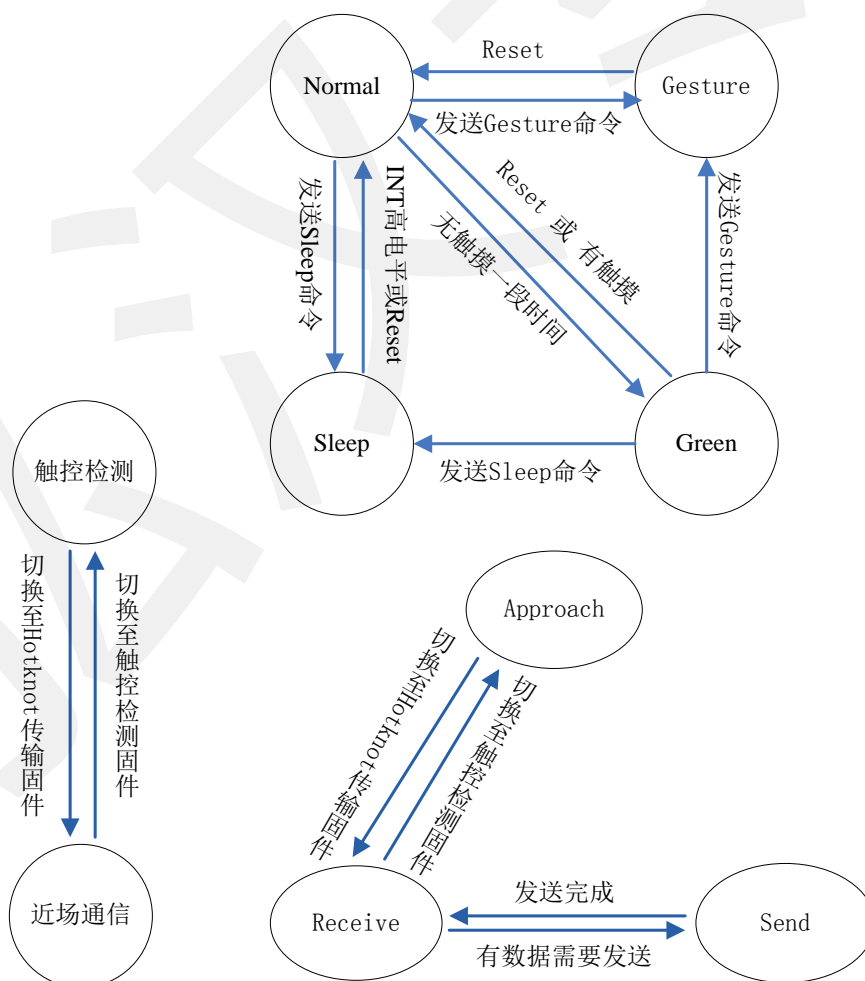
采用中断读取方式，触发中断后按上述轮询过程读取坐标。

GT9147 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、2 中输出上升沿后，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。
- 4、2 中输出上升沿后，若主控未在一个周期内读走坐标，下次 GT9147 即使检测到坐标更新会再输出一个 INT 脉冲但不更新坐标。
- 5、若主控一直未读走坐标，则 GT9147 会一直打 INT 脉冲。

六、工作模式切换

6.1 工作模式



a) Normal Mode

GT9147 在 Normal mode 时，最快的坐标刷新周期为 7ms-10ms 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）。

Normal mode 状态下，一段时间无触摸事件发生，GT9147 将自动转入 Green mode，以降低功耗。GT9147 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。

b) Green Mode

在 Green mode 下，GT9147 扫描周期约为 40ms，若检测到有触摸动作发生，自动进入 Normal mode。

c) Gesture mode

主 CPU 通过下发 I2C 命令 8 到 0x8046，再下发命令 8 到 0x8040，让 GT9147 进入 Gesture mode 后，可通过滑动屏体、双击或者在屏体上书写特定字符实现唤醒。

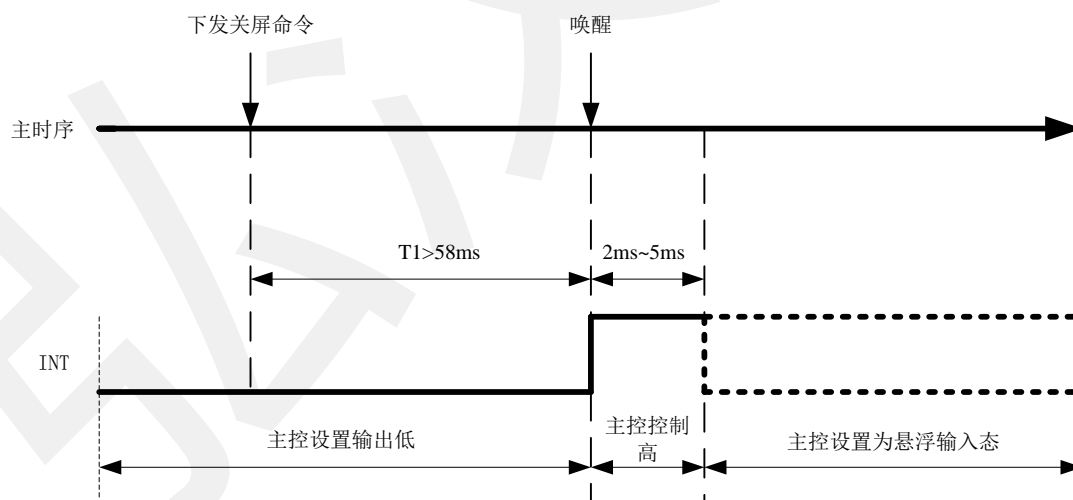
在 Gesture mode 下，GT9147 检测到手指在屏体上滑动足够的长度，INT 就会输出一个 250us 左右的脉冲，主控收到脉冲后醒来亮屏。

在 Gesture mode 下，GT9147 检测到手指在屏体上发生双击动作，INT 也会输出一个 250us 左右的脉冲，主控收到脉冲后醒来亮屏。

在 Gesture mode 下，GT9147 检测到手指在屏体上书写特定字符，INT 也会输出一个 250us 左右的脉冲，主控收到脉冲后醒来亮屏。

d) Sleep Mode

主 CPU 通过 I2C 命令，使 GT9147 进入 Sleep mode（需要先将 INT 脚输出低电平）。当需要 GT9147 退出 Sleep mode 时，主机输出一个高电平到 INT 脚（主机打高 INT 脚 2~5ms），唤醒后 GT9147 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。



e) Approach Mode

当使能 HotKnot 接近检测功能后，GT9147 默认运行在 Approach mode 下，当退出此模式后，主 CPU 可通过下发 0x20 或 0x21 命令，使 GT9147 进入 Approach mode。该模式下，触控检测和 HotKnot 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。

f) Receive Mode

在 GT9147 运行在 Approach mode 时，主 CPU 获取到 GT9147 检测到可通讯终端，主 CPU 再下发 HotKnot 传输固件使 GT9147 进入 Receive mode。在该模式下，不断地检测有无通讯信号，检测到后，开始接收数据，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。

g) Send Mode

在 GT9147 运行在 Receive mode 时，主 CPU 将待发数据发送至发送缓冲区，GT9147 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，先发送导频连接信号，并检测到接收端有返回序列，再接着发送数据序列，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。

七、版本修订记录

文件版本	日期	修订
Rev.00	2014-04-03	首次发布
Rev.01	2014-05-20	增加下发 Hotknot 传输系统时间规范说明