

GT9 非单层多点系列编程指南文件

GT913/GT915/GT927/GT928/GT9110

GT913P/GT915P /GT9110P

目 录

一、接口说明.....	2
二、通信时序.....	2
2.1 主机对 GT9 系列进行写操作时序	2
2.2 主机对 GT9 系列进行读操作时序	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.....	11
四、上电初始化与寄存器动态修改.....	14
4.1 GT9 系列上电时序.....	14
4.2 I2C 地址选择.....	15
4.3 寄存器动态修改.....	16
五、坐标读取.....	16
六、工作模式切换.....	17
七、版本修订记录.....	18

一、接口说明

GT9 非单层多点系列（以下简称 GT9 系列）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 GT9 系列的 RESET 口为高或低。为保证可靠复位，建议 RESET 脚输出低 100μs 以上。

GT9 系列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。GT9 系列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。GT9 系列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 GT9 系列进行写操作时序



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。

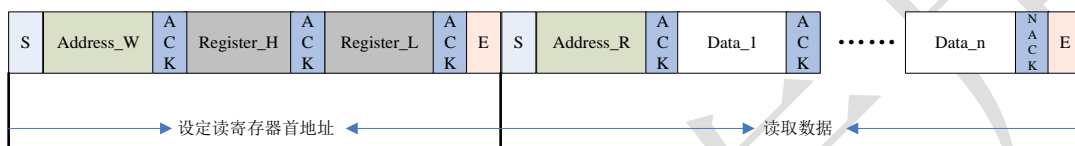
Data_1 至 Data_n: 数据字节 1—n。

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，GT9 系列自动将其往高地址顺序存储。

2.2 主机对 GT9 系列进行读操作时序

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，GT91XX 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)

0x8040	Command	0: 读坐标状态 3: 基准更新 (内部测试) 其余值无效	1: 差值原始值 4: 基准校准 (内部测试)	2: 软件复位 5: 关屏
0x8041	NC	Reserved		
0x8042	Proximity_En	接近感应开关		

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(新下发的配置版本号大于原版本,或等于原版本号但配置内容发生变化时保存,版本号版本正常范围: 'A'~'Z',发送 0x00 则将版本号初始化为'A')							
0x8048	X Output Max (Low Byte)	x 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: 1~10			
0x804D	Module_Switch1	Stylus_priority (预定义)		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	Reserved				WP_dis	Reserved	SCM_en	Touch_key
0x804F	Shake_Count	Reserved				手指按下/松开去抖次数			
0x8050	Filter	First_Filter		Normal_Filter(原始坐标窗口滤波值,系数为1)					
0x8051	Large_Touch	大面积触摸点个数							
0x8052	Noise_Reduction	Reserved				噪声消除值(系数为1,0-15 有效)			
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	Reserved							
0x8058	y_threshold								
0x8059	X_Speed_Limit	Reserved							
0x805A	Y_Speed_Limit								
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	Stretch_Rate	Reserved			弱拉伸的拉伸程度（拉伸 x/16 Pitch） （beta 版占用，发布版无效）
0x805E	Stretch_R0	区间 1 系数			
0x805F	Stretch_R1	区间 2 系数			
0x8060	Stretch_R2	区间 3 系数			
0x8061	Stretch_RM	各区间基数			
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	Pannel_Sensor_TimeL	相邻两次驱动信号输出时间间隔（以 us 为单位）,Reserved（beta 版占用，发布版无效）			
0x806A	Pannel_Sensor_TimeH				
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	Reserved			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	NC	Reserved			
0x8070	NC	Reserved			
0x8071	NC	Reserved			
0x8072	Stylus_Tx_Gain	暂未定义（当 stylus_priority=0 时无效）			
0x8073	Stylus_Rx_Gain	暂未定义（当 stylus_priority=0 时无效）			
0x8074	Stylus_Dump_Shift	笔原始值放大系数（2 的 N 次方）			

0x8075	Stylus_Driver_Touch_Level	笔触摸有效阈值（驱动方向）		
0x8076	Stylus_Sensor_Touch_Level	笔触摸有效阈值（感应方向）		
0x8077	Stylus_Control	触摸笔超时退出时间（以秒为单位）		
0x8078	Base_reduce	S 型改善量（通常为 2-4）		削底系数 N，削底量=Base+Base/2 的 N 次方（通常 2-4）
0x8079	NC	Reserved		
0x807A	Freq_Hopping_Start	跳频范围的起点频率（以 2KHz 为单位，例如 50 表示 100KHz）		
0x807B	Freq_Hopping_End	跳频范围的终点频率（以 2KHz 为单位，例如 150 表示 300KHz）		
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	Reserved	Detect_Time_Out (噪声检测超时时间，以秒为单位)
0x807E	Hoppging_Threshold	Large_Noise_Threshold (beta 版占用，发布版无效，Reserved)		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	Noise_Threshold	判别有干扰的门限（所有频率点上干扰量小于此值认为无干扰）		
0x8080	NC	Reserved		
0x8081	NC	Reserved		
0x8082	Hopping_seg1_BitFreqL	跳频检测区间频段 1 中心点基频（适用于驱动 A、B）		
0x8083	Hopping_seg1_BitFreqH			
0x8084	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）		
0x8085	Hopping_seg2_BitFreqL	跳频检测区间频段 2 中心点基频（适用于驱动 A、B）		
0x8086	Hopping_seg2_BitFreqH			
0x8087	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）		
0x8088	Hopping_seg3_	跳频检测区间频段 3 中心点基频（适用于驱动 A、B）		

	BitFreqL		
0x8089	Hopping_seg3_ BitFreqH		
0x808A	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）	
0x808B	Hopping_seg4_ BitFreqL	跳频检测区间频段 4 中心点基频（适用于驱动 A、B）	
0x808C	Hopping_seg4_ BitFreqH		
0x808D	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）	
0x808E	Hopping_seg5_ BitFreqL	跳频检测区间频段 5 中心点基频（适用于驱动 A、B）	
0x808F	Hopping_seg5_ BitFreqH		
0x8090	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）	
0x8091	NC	Reserved	
0x8092	NC	Reserved	
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 (1~16s)	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	NC	Reserved	
0x809E	NC	Reserved	
0x809F	NC	Reserved	
0x80A0	NC	Reserved	
0x80A1	NC	Reserved	

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=接近感应生效阈值	
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	Self_Cap_Tx_gain	自电容发送增益	
0x80AB	Self_Cap_Rx_gain	自电容接收增益	
0x80AC	Self_Cap_Dump_Shift	自电容原始值放大系数 (2 的 N 次方)	
0x80AD	SCap_Diff_Up_Level_Drv	自电容抑制悬浮上升阈值 (驱动方向)	
0x80AE	SCap_Merge_Touch_Level_Drv	自电容 Touch Level (驱动方向)	
0x80AF	SCap_Pulse_TimeL	自电容采样时间 (低字节)	
0x80B0	SCap_Pulse_TimeH	自电容采样时间 (高字节)	
0x80B1	SCap_Diff_Up_Level_Sen	自电容抑制悬浮上升阈值 (感应方向)	
0x80B2	SCap_Merge_Touch_Level_Sen	自电容 Touch Level (感应方向)	
0x80B3	NC	Reserved	
0x80B4	NC	Reserved	
0x80B5	NC	Reserved	
0x80B6	NC	Reserved	
0x80B7	Sensor_CH0	ITO Sensor0 对应的芯片通道号	
0x80B8	Sensor_CH1		
0x80B9	Sensor_CH2		
0x80BA	Sensor_CH3		
0x80BB	Sensor_CH4		

0x80BC	Sensor_CH5	
0x80BD	Sensor_CH6	
0x80BE	Sensor_CH7	
0x80BF	Sensor_CH8	
0x80C0	Sensor_CH9	
0x80C1	Sensor_CH10	
0x80C2	Sensor_CH11	
0x80C3	Sensor_CH12	
0x80C4	Sensor_CH13	
0x80C5	Sensor_CH14	
0x80C6	Sensor_CH15	
0x80C7	Sensor_CH16	
0x80C8	Sensor_CH17	
0x80C9	Sensor_CH18	
0x80CA	Sensor_CH19	
0x80CB	Sensor_CH20	
0x80CC	Sensor_CH21	
0x80CD	Sensor_CH22	
0x80CE	Sensor_CH23	
0x80CF	Sensor_CH24	
0x80D0	Sensor_CH25	
0x80D1	Sensor_CH26	
0x80D2	Sensor_CH27	
0x80D3	Sensor_CH28	
0x80D4	Sensor_CH29	
0x80D5	Driver_CH0	ITO Driver0 对应的芯片通道号
0x80D6	Driver_CH1	
0x80D7	Driver_CH2	
0x80D8	Driver_CH3	
0x80D9	Driver_CH4	
0x80DA	Driver_CH5	
0x80DB	Driver_CH6	
0x80DC	Driver_CH7	
0x80DD	Driver_CH8	
0x80DE	Driver_CH9	
0x80DF	Driver_CH10	
0x80E0	Driver_CH11	
0x80E1	Driver_CH12	
0x80E2	Driver_CH13	
0x80E3	Driver_CH14	
0x80E4	Driver_CH15	
0x80E5	Driver_CH16	

0x80E6	Driver_CH17	
0x80E7	Driver_CH18	
0x80E8	Driver_CH19	
0x80E9	Driver_CH20	
0x80EA	Driver_CH21	
0x80EB	Driver_CH22	
0x80EC	Driver_CH23	
0x80ED	Driver_CH24	
0x80EE	Driver_CH25	
0x80EF	Driver_CH26	
0x80F0	Driver_CH27	
0x80F1	Driver_CH28	
0x80F2	Driver_CH29	
0x80F3	Driver_CH30	
0x80F4	Driver_CH31	
0x80F5	Driver_CH32	
0x80F6	Driver_CH33	
0x80F7	Driver_CH34	
0x80F8	Driver_CH35	
0x80F9	Driver_CH36	
0x80FA	Driver_CH37	
0x80FB	Driver_CH38	
0x80FC	Driver_CH39	
0x80FD	Driver_CH40	
0x80FE	Driver_CH41	
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)

部分寄存器补充说明如下:

[0x804D] Module_Switch1

Bit7-bit6: Stylus_priority, 供拓展使用, 暂无功能。

00: 不包含笔的应用

01: 笔优先级高于屏

02: 屏优先级高于屏

03: 笔屏优先级相同

Bit5-bit4: Stretch_rank, 拉伸方式

00,01: 弱拉伸 0.4P

02: 兼容 8110 产品 42 通道特殊拉伸

03: 自定义拉伸

[0x804D] Module_Switch2

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

Bit1: SCM_en 抑制悬浮开关, 置 1 开启, 清 0 关闭。

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)。

[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)

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

[0x807E] Hopping_Threshold

Bit7~4: Large_Noise_Threshold, 工作频率择优干扰量门限, 当工作频点上干扰量大于 $\text{Large_Noise_Threshold} / 16 * \text{最大干扰值}$ 时进入频率择优处理

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 > 当前工作频率上干扰量 * $\text{Hopping_Hit_Threshold} / 16$ 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 $\text{Key_Restrain} / 16$ 时则不输出按键, 推荐设置 7 ± 2

3.3 坐标信息

Addr	Access	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
0x8140	R	Product ID (Lowest Byte, ASCII 码)							
0x8141	R	Product ID (Third Byte, ASCII 码)							
0x8142	R	Product ID (Second Byte, ASCII 码)							

0x8143	R	Product ID (Highest Byte, ASCII 码)				
0x8144	R	Firmware version (16 进制数 LowByte)				
0x8145	R	Firmware version (16 进制数 HighByte)				
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 (笔上报坐标 ID 复用此位置, 为特殊的 128)				
0x8150	R	point 1 x coordinate (low byte)				
0x8151	R	point 1 x coordinate (high byte)				
0x8152	R	point 1 y coordinate (low byte)				
0x8153	R	point 1 y coordinate (high byte)				
0x8154	R	Point 1 size (low byte)				
0x8155	R	point 1 size (high byte)				
0x8156	R	Reserved				
0x8157	R	track id				
0x8158	R	point 2 x coordinate (low byte)				
0x8159	R	point 2 x coordinate (high byte)				
0x815A	R	point 2 y coordinate (low byte)				
0x815B	R	point 2 y coordinate (high byte)				
0x815C	R	point 2 size (low byte)				
0x815D	R	point 2 size (high byte)				
0x815E	R	Reserved				
0x815F	R	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 (low byte)				
0x8165	R	point 3 size (high byte)				
0x8166	R	Reserved				
0x8167	R	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 (low byte)				

0x816D	R	point 4 size (high byte)
0x816E	R	Reserved
0x816F	R	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 (low byte)
0x8175	R	point 5 size (high byte)
0x8176	R	Reserved
0x8177	R	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 (low byte)
0x817D	R	point 6 size (high byte)
0x817E	R	Reserved
0x817F	R	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 (low byte)
0x8185	R	point 7 size (high byte)
0x8186	R	Reserved
0x8187	R	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 (low byte)
0x818D	R	point 8 size (high byte)
0x818E	R	Reserved
0x818F	R	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 (low byte)
0x8195	R	point 9 size (high byte)
0x8196	R	Reserved

0x8197	R	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 (low byte)
0x819D	R	point 10 size (high byte)
0x819E	R	Reserved
0x819F	R	KeyValue

部分寄存器增补说明如下：

[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

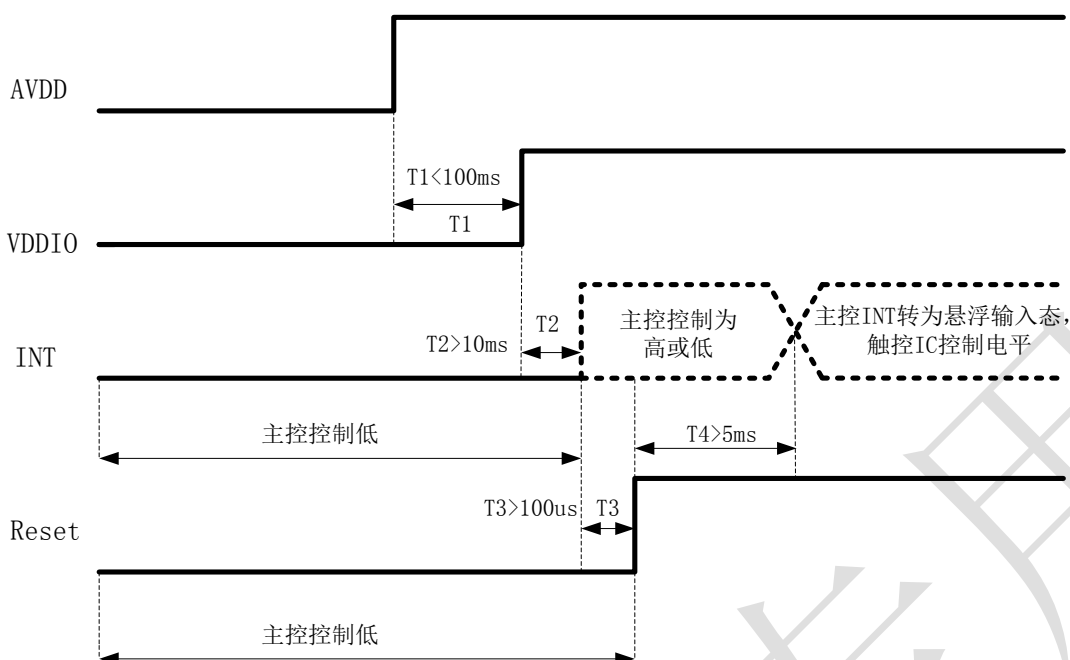
[0x819F]: KeyValue

按键值，KeyValue 的位置并不固定，而是跟在有效坐标的后面。例如 0x819F 是屏上有 10 个坐标时的按键位置，而有 9 个坐标时按键位置则在 0x8197。

四、上电初始化与寄存器动态修改

4.1 GT9 系列上电时序

主机上电后，需要控制 GT9 的 AVDD、VDDIO、INT、Reset 等脚位，控制时序请遵从如下时序图：

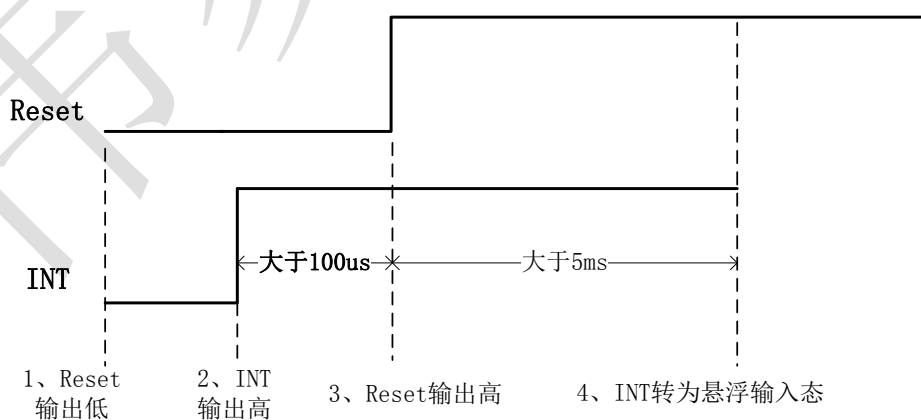


INT T2 时间后, 主控是要输出高, 还是低, 取决于主机要用何 I2C 从设备地址与 GT9 芯片通信, 若用地址 0x28/0x29, 则输出高; 若用地址 0xBA/0xBB, 则输出低。

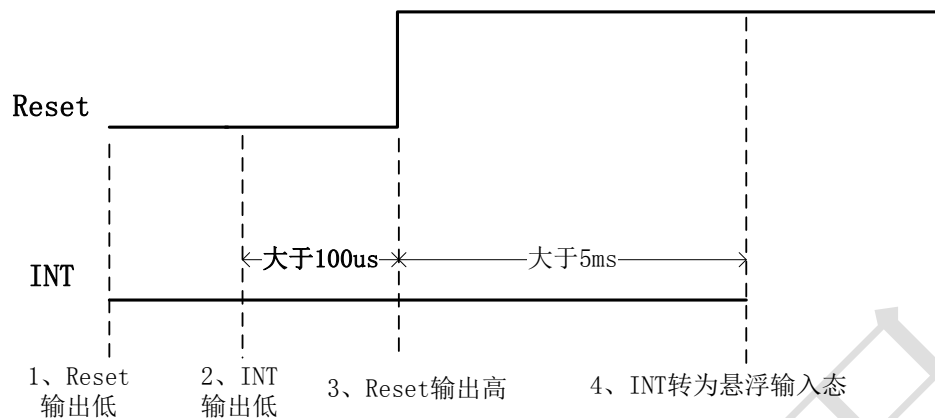
4.2 上电或复位 I2C 地址选择

GT9 系列的 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 GT9 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

GT9 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。

GT9 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

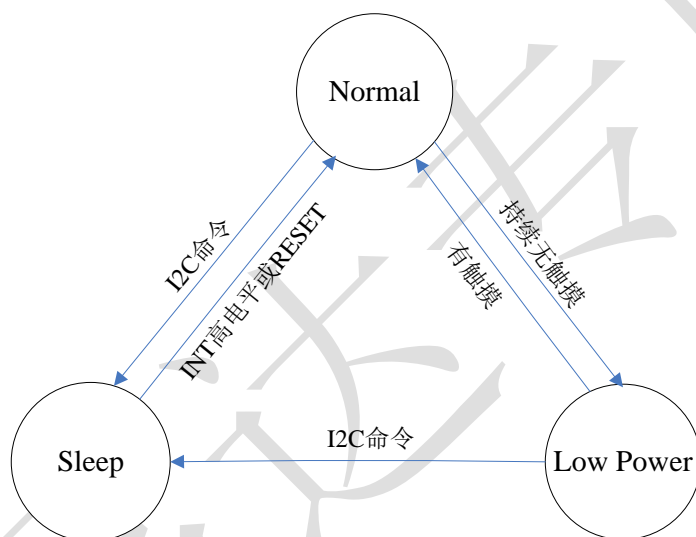
在一个周期内将坐标读走并将 `buffer status(0x814E)` 写为 0。

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 GT9 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 GT9 会一直打 INT 脉冲。

六、工作模式切换

GT9 工作模式分为 Normal、Low Power(Green)、Sleep 三种，各种工作状态间相互转换关系如下图所示：



默认情况下，GT9 工作自动切换 Normal 和 Low Power 工作模式，按键时及松键后的一段时间（这段时间由配置参数 `Low_Power_Control` 设定，0~15 秒可设）工作在 Normal mode，若该段时间后还处于无按键状态，则进入 Low Power 工作模式（低速扫描）。

Normal 模式

GT9 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）。

Normal mode 下，一段时间无触摸事件发生，GT9 将自动转入 Low Power mode，以降低功耗。GT9 无触摸自动进入 Low Power mode 的时间可通过配置信息设置，范围为 0~15s，步进为 1s。

Low Power(Green) mode

在 LowPower mode 下，GT9 扫描周期固定为 40ms，若检测到有触摸动作发生，自动进入 Normal mode。

Sleep mode 及唤醒

主 CPU 通过 I2C 命令，使 GT9 进入 Sleep mode（需要先将 INT 脚输出低电平）。当需要 GT9 退出 Sleep mode 时，主机可采用 INT 高电平唤醒或 reset 唤醒。若采用 INT 高电平唤醒，操作时序为：输出高电平到 INT 脚（主机打高 INT 脚 2~5ms，然后转悬浮输入态），唤醒后 GT9 将进入 Normal mode；当采用 reset 脚唤醒时，需要按前述上电初始化过程控制 INT 脚和 reset 脚。

七、版本修订记录

文件版本	修订
Rev1.0	首次发布
Rev1.1	2012-9-24 更新配置信息内容，删除跳频描述
Rev1.2	2012-10-8 修改部分表述不清晰的地方
Rev1.3	2012-10-23 1、增加上电初始化发送配置信息时序控制说明。 2、增加 INT 唤醒和 reset 唤醒时序说明。 3、更改工作模式切换中 sleep INT 唤醒为高电平唤醒。