

GT1151Q 编程指南

===== 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，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

GT1151Q 编程指南	1
1. 接口说明.....	3
2. I ² C 通信时序	4
2.1 主机写操作时序	4
2.2 主机读操作时序	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	6
3.4 手势信息	10
4. 相关操作时序.....	11
4.1 上电时序	11
4.2 复位时序	11
4.3 掉电时序	11
4.4 上电发送配置信息	12
4.5 主控响应 “INT Request”	12
4.6 寄存器动态修改	13
5. 坐标读取.....	14
6. 工作模式切换.....	15
6.1 Normal Mode	15
6.2 Green Mode	15
6.3 Gesture Mode.....	15
6.4 Sleep Mode.....	16
7. Gesture 模式驱动修改	17
7.1 灭屏后进入 Gesture 模式	17
7.2 灭屏后进入 Sleep 模式	17
7.3 按电源键（或 home 键）开屏	17
7.4 建议可与 IR 配合	17
8. Gesture 模式坐标读取及校验	18
8.1 坐标读取	18
8.2 坐标及轨迹信息校验	18
9. 版本修订记录.....	20

1. 接口说明

GT1151Q 与主机接口共有 6 PIN，分别为：VDD、GND、SCL、SDA、INT、RESET。

主机的 INT 口线需具有下降沿中断触发功能，且必须置为输入态，并通过主机内部设置或外部增加上拉电阻对 INT 口上拉。

主机通过输出高、低来控制 GT1151Q 的 RESET 口为高或低。为保证可靠复位，建议 RESET 脚输出低 100us 以上。

GT1151Q 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 200Kbps 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。GT1151Q 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。从设备地址如下表：

7 位地址	8 位写地址	8 位读地址
0x14	0x28	0x29

2. I²C 通信时序

2.1 主机写操作时序



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。

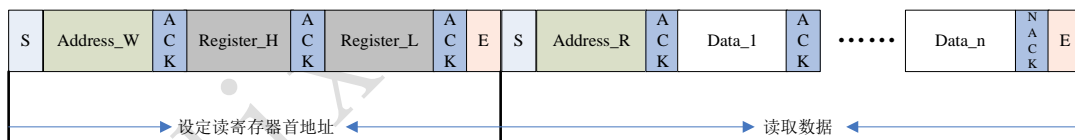
Data_1 至 Data_n: 数据字节 1-n。

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，GT1151Q 自动将其往高地址顺序存储。

2.2 主机读操作时序

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，GT1151Q 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） 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)

常规配置

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 ~ 0x813B		配置内容。							
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)							

扩展配置

Addr	Config Data	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
0xBF7B ~ 0xBFF8		配置内容。							
0xBFF9	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)							
0xBFFA	Config_Chksum_L								

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	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							

Addr	Access	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
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	Reserved			track_id			
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 (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)							

Addr	Access	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
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)							

Addr	Access	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
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_opt 1 和 sensor_opt 2 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。

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 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)								

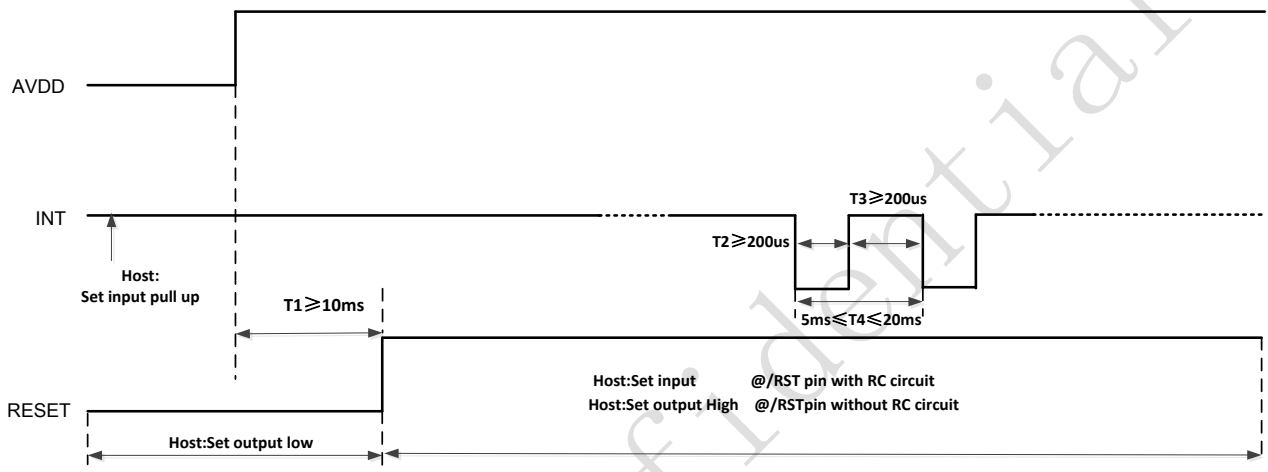
(手势坐标信息)

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)							

4. 相关操作时序

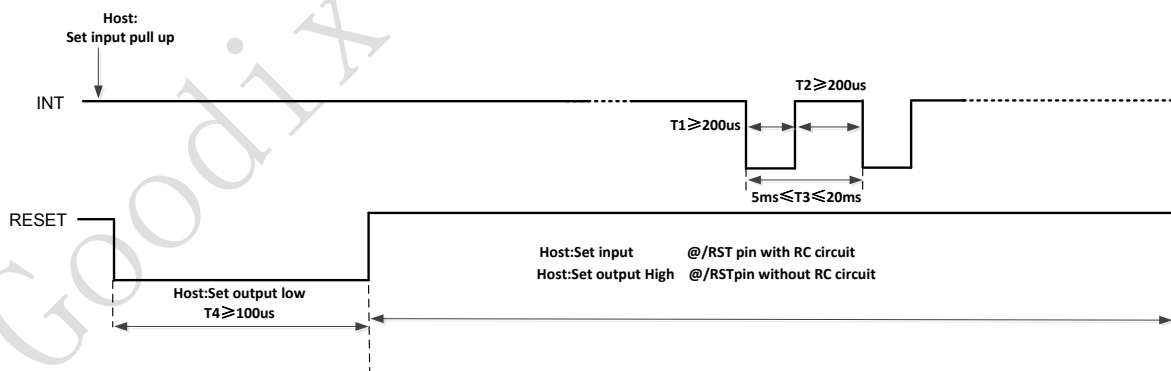
4.1 上电时序

主机上电后，需要控制 GT1151Q 的 AVDD、INT、RESET 等脚位，控制时序请遵从如下时序图：



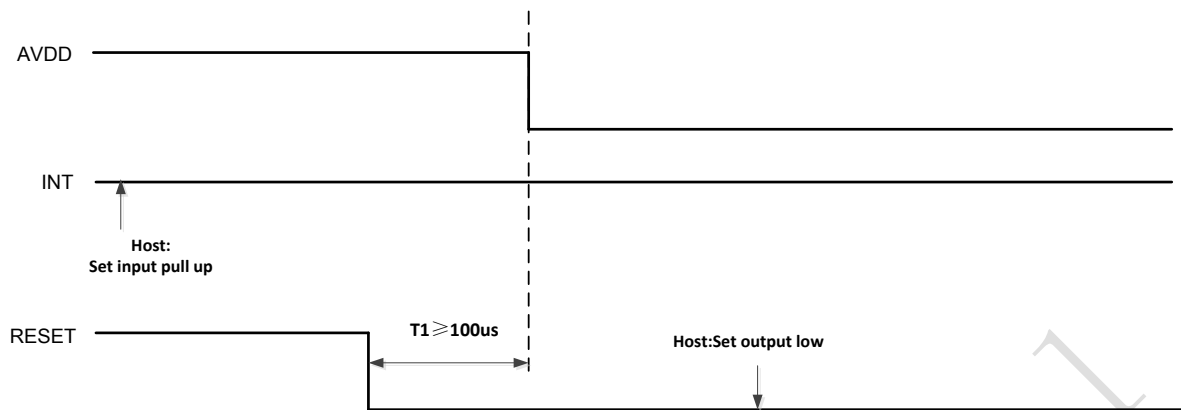
4.2 复位时序

主机复位 GT1151Q 的控制时序如下：



4.3 掉电时序

对 GT1151Q 掉电的控制时序如下：



4.4 上电发送配置信息

主机控制 GT1151Q 上电过程中，当主控拉高 RESET 后，需要延时 60ms 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。

4.5 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 GT1151Q
0xFF	DLE，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。

- b) 按照复位时序对 GT1151Q 进行复位，完成“0x03”请求。

4.6 寄存器动态修改

GT1151Q 支持寄存器动态修改，当按照第 2 节时序对配置区内（包括常规配置和扩展配置）任何寄存器修改时，需要更新对应的 Config_Chksum，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。

Goodix Confidential

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。

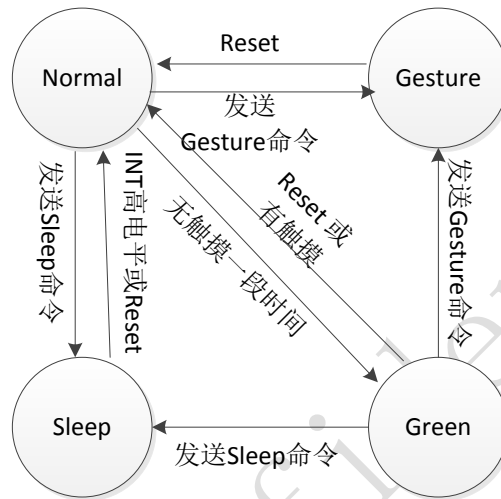
采用中断读取方式，触发中断后按上述轮询过程读取坐标。

GT1151Q 中断信号输出时序为：

- 1) 待机时 INT 脚为高。
- 2) 有坐标更新时，输出下降沿。
- 3) 输出下降沿后，INT 脚会保持低直到下一个周期（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，下一周期 GT1151Q 仍然会继续进行坐标计算，并输出下降沿，但不更新坐标信息区的内容。
 - b) 若主控一直未读走坐标，则 GT1151Q 会一直打 INT 脉冲。

6. 工作模式切换

GT1151Q 工作模式分为 Normal、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

GT1151Q 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

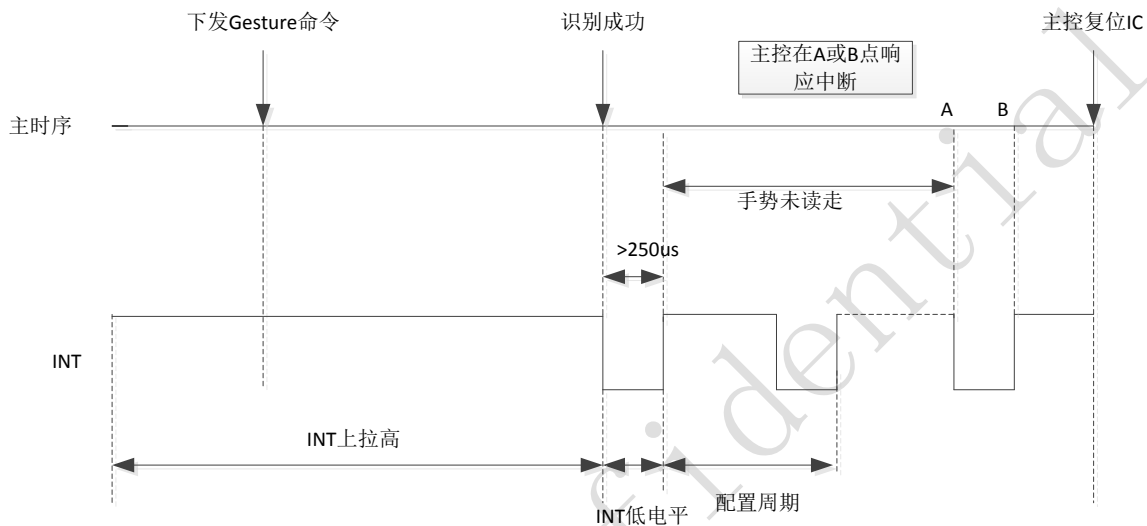
Normal mode 状态下，一段时间无触摸事件发生，GT1151Q 将自动转入 Green mode，以降低功耗。GT1151Q 无触摸自动进入 Green mode 的时间可通过配置信息设置，范围为 0~14s，步进为 1s，Green Mode 可通过配置关闭。在 Green mode 下，GT1151Q 扫描周期约为 40ms，若检测到有触摸动作发生，自动进入 Normal mode。

6.3 Gesture Mode

若主 CPU 通过 I²C 命令，GT1151Q 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，GT1151Q 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（时间长度可通过配置修改）以上的低电平脉冲，主控检测到脉冲后醒来亮屏，并复位 GT1151Q 使其退出 Gesture Mode。

唤醒时序如下：



6.4 Sleep Mode

主 CPU 通过 I²C 命令，使 GT1151Q 进入 Sleep mode。当需要 GT1151Q 退出 Sleep mode 时，主 CPU 对 GT1151Q 执行复位操作。复位后 GT1151Q 将进入 Normal mode。下发命令与复位之间的时间间隔要求大于 58ms。

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 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）个无符号 16 位的数据，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 是校验和的高位，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
Rev.01	2014-07-17	首版发布。

Goodix Confidential