GT1151Q Programming Guide

= = = = = Disclaimer = = = = =

The device application information and other similar content described in this publication are provided solely for your use and may be superseded by updated information. It is your own responsibility to ensure that the application complies with the specifications. GOODIX makes no representations or warranties, express or implied, written or oral, statutory or otherwise, including, but not limited to, representations or warranties of use, quality, performance, merchantability, or fitness for a particular purpose, with respect to this information. GOODIX assumes no responsibility for this information and the consequences of its use. GOODIX products may not be used as critical components in life support systems without the written approval of GOODIX. No license may be transferred, implicitly or otherwise, under GOODIX intellectual property protection.

Contents

1. Interface Description 3
2. I²C Communication timing 4
2.1 Host Write Operation Timing 4
2.2 Host Read Operation Timing 4
3. Register List 6
3.1 Real-time Command 6
3.2 Configuration Information (R/W) 7
3.3 Coordinate Information 7
3.4 Gesture Information 11
4. Related Operation Timing 13
4.1 Power-on Sequence 13
4.2 Reset Sequence 13
4.3 Power Down Sequence 13
4.4 Power on to Send Configuration Information 14
4.5 Master response "INT Request" 14
4.6 Register Dynamic Modification 15
5. Coordinate Reading 16
6. Operating mode switching 17
6.1 Normal Mode 17
6.2 Green Mode 17
6.3 Gesture Mode 17
6.4 Sleep Mode 18
7. Gesture Mode Driver Modification 19
7.1 Enter Gesture mode After the Screen Off 19
7.2 Enter Sleep Mode After the Screen Off 19
7.3 Press Power Button (or home button) to Open the Screen. 19
7.4 Recommended for Use with IR 19
8. Coordinate Reading and Verification in Gesture Mode 20
8.1 Coordinate Reading 20
8.2 Coordinate and Trajectory Information Verification 21
9. Version revision record 22

1. Interface Description

There are 6 PINs in the interface between GT1151Q and the host: VDD, GND, SCL, SDA, INT, and RESET.

The INT port line of the host must have the falling edge interrupt trigger function and must be set to the input state, and the pull-up resistor is added to the INT port through the internal setting of the host or externally.

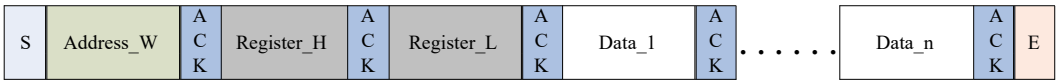
The host controls the RESET port of the GT1151Q to be high or low by outputting high or low. To ensure a reliable reset, it is recommended that the RESET pin output be more than 100us low.

The GT1151Q uses standard I2C communication to communicate with the host, and the maximum rate can be supported by up to 400Kbps. When the host uses a communication rate above 200Kbps, special attention needs to be paid to the external pull-up resistor resistance of the I2C port to ensure that the SCL and SDA edges are steep enough. The GT1151Q is always used as a slave device in communication, and its I2C device address consists of 7 bits of device address plus 1 bit of read/write control, with the high 7 bits being the address and bit 0 being the read/write control bit. The slave device addresses are listed in the following table:

7-bit address	**8-bit** write address	**8-bit** read address
0x14	0x28	0x29

2. I²C Communication timing

2.1 Host write operation timing



S: Start signal.

Address_W: Slave device address with write control bit.

ACK: Acknowledge signal.

Register_H, Register_L: the first address of the 16-bit register to be written.

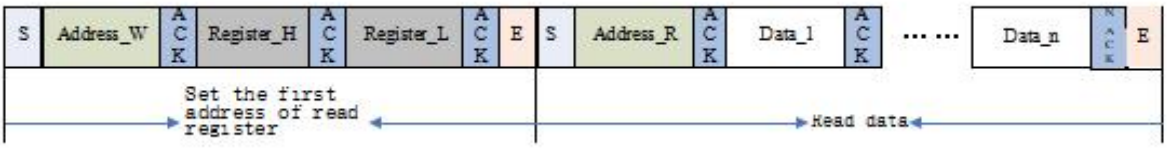
Data_1 to Data_n: Data byte 1-n.

E: Emergency stop signal.

After setting the starting address of the write operation register, it is possible to write only one byte of data or to write multiple bytes of data at once, and GT1151Q will automatically store them in the order of higher addresses.

2.2 Host Read Operation Timing

Firstly, set the first address of the register to be read by the write operation timing as mentioned above, and then resend the start signal for read addressing to read the register data.



Address_R: Slave device address with read control bits.

NACK: The master returns NACK after the last 1 byte is read.

After setting the read operation register address, the master can read 1 byte at a time, or read multiple bytes of data at once, and the GT1151Q automatically increments the register address to send the subsequent

data in order.

The stop signal (the first E signal in the above figure) after setting the read operation register address can be sent or not, but the start signal to restart I2C communication must be sent again.

3. Register List

3.1 Real-time Command

Addr	Name	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
0x8040	Command	**0x00:** Read coordinate status **0x01, 0x02:** Difference original value **0x03:** Reference update (internal test) **0x04:** Reference calibration (internal test) 0x05: Screen off **0x06:** Enter charging mode **0x07:** Exit charging mode 0x08: Enter gesture wake-up mode **0x0b:** Hand mode (weak signal not supported) 0x0c: Auto mode (automatically switch between hand and glove) **0x31:** Save custom gesture templates **0x35:** Clear the gesture template information saved in the touch IC **0x36:** delete a gesture template **0x37:** query gesture template information **0xAA:** ESD protection mechanism used by the driver to write AA at regular intervals and read checks at regular intervals							
0x8041	Command_Data	Data corresponding to the command (no need to send data down the command data area to send down 0)							
0x8042	Command_ Checksum	Cumulative sum check of command and data (sum(0x8040~0x8042)==0)							
0x8043	ESD_Check	ESD protection mechanism is used, cleared at initialization, followed by read and write operations by the driver							
0x8044	Request	FW Active request sent to the master							
0x8045	FW_Status_L	Program status word							
0x8046	FW_Status_H								

3.2 Configuration Information (R/W)

General Configuration

Addr	Config Data	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
0x8050	Config_ Version	Bit 7 is the flag for whether it has been permanently stored. (0: normal, 1:permanently stored), bit0~bit6 is the corresponding version number							
0x8051 ~ 0x813B		Configuration content.							
0x813C	Config_ Chksum_H	16-bit checksum of the configuration information using the big-endian format, with the high-order byte stored in the low-order address.							
0x813D	Config_ Chksum_L								
0x813E	Config_Fresh	Configuration updated flag (master writes 1 here)							

Extended Configuration

Addr	Config Data	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
0xBF7B ~ 0xBFF8		Configuration content.							
0xBFF9	Config_ Chksum_H	16-bit checksum of the configuration information using the big-endian format, with the high-order byte stored in the low-order address.							
0xBFFA	Config_ Chksum_L								

3.3 Coordinate Information

Addr	Access	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
0x8140	R	Product ID（First Byte, ASCII code）							
0x8141	R	Product ID (Second Byte, ASCII code)							
0x8142	R	Product ID (Third Byte, ASCII code)							
0x8143	R	Product ID (Forth Byte, ASCII code)							
0x8144	R	CID is a software model identifier in Patch version number. It is a two-digit decimal number (please use BCD code for storage) starting from 00 and the maximum value is 26. "0" represents the public version and does not need to be displayed; 1-26 are customized versions, which can be displayed in alphabetical order from A to Z..							
0x8145	R	Main version number of Patch version number (2-bit compressed BCD code)							
0x8146	R	Sub version number of Patch version number (2-bit compressed BCD code)							
0x8147	R	Main Mask version number							
0x8148	R	Sub MASK version number							
0x8149	R	Internal MASK version number							

Addr	Access	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
0x814A	R	BondingOption				Vendor_ID			
0x814B	R	CheckSum							
0x814C	R	Reserved							
0x814D	Reserved	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), length len="Touch Points "*8+3							

Some register additions are described as follows:

[0x814A] Bit3~Bit0:Vendor_ID

The current module option information is identified by the sensor_opt 1 and sensor_opt 2 pins on the circuit. When the external connection status of the two option pins is different, they indicate 6 different sensors, as shown in the following table:

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 means that the coordinate (or key) is ready to be read by the master; 0 means that it is not ready and the data is invalid. After the master has read the coordinates, this flag (or the whole byte) must be written to 0 via I2C.

Bit4: HaveKey, 1 means key is present, 0 means no key (key has been released).

Bit3~0: Number of touch points, the number of coordinate points on the screen.

[0x814F]

Bit7: touch_sta, 1 means high sensitivity touch coordinate; 0 means normal sensitivity touch coordinate.

Bit3~0: Track ID, touch point id number.

[0x8177] KeyValue

The position of KeyValue is not fixed, but follows the effective coordinate. For example, 0x8177 is the key

position when there are five coordinates on the screen, and the key position is 0x816F when there are four coordinates.

3.4 Gesture Information

(Gesture Feature Information: Multiplex Coordinate Information Address)

Addr	Access	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
0x8140	R	Product ID (First Byte, ASCII code, G)							
0x8141	R	Product ID (Second Byte, ASCII code, E)							
0x8142	R	Product ID (Third Byte, ASCII code, S)							
0x8143	R	Product ID (Forth Byte, ASCII code, T)							
0x8144	R	Gesture major version number							
0x8145	R	Gesture vice version number							
0x8146	R	Gesture internal version number							
0x8147	R	MASK major version number							
0x8148	R	MASK minor version number							
0x8149	R	MASK build number							
0x814A	R	BondingOption				Vendor_ID			
0x814B	R	CheckSum							
0x814C	R/W	Gesture type (character ASCII code for 0x21-0x7E), right slide (0xAA), left slide (0xBB), slide down (0xAB), slide up (0xBA), double click (0xCC), key click (0xC1, 0xC2, 0xC4, 0xC8, lower four bits change to key value), Custom (0x01~0x0A)							
0x814D	R	Number of gesture touch points (coordinate storage location 0xBDA8)							
0x814E	R	Buffer 1 (starting position 0x8150) Protocol type (0x01: Single-pen gesture, 0x02: Multi-pen gesture)				Buffer 2 (start location 0xBDA8) Protocol type (0x02: Report all points, 0x03: Report feature points)			
0x814F	R	Buffer 1 Number of data							
0x8150~0x819E	R	Buffer 1 content							
0x8151	R	Check Sum (Address changes according to buffer length); Checksum start address: 0x814C Checksum length (u8): Gesture Data length (buffer length) + 4							
0x819F	R	Gesture track coordinates 16-bit Check Sum (low byte) (address fixed) Checksum Start Address: 0xBDA8, Checksum length (u16): Gesture Points Count * 2							
0x81A0	R	Gesture track coordinates 16-bit Check Sum (high byte) (address fixed)							

(Gesture coordinates information)

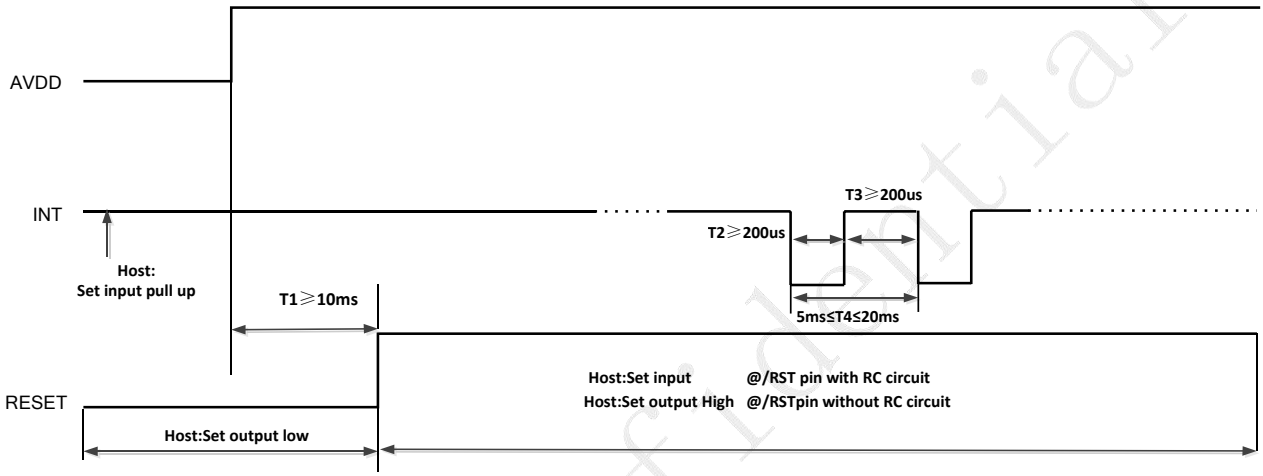
Addr	Access	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
0xBDA8	R	Gesture point 1 x coordinate (low byte)							
0xBDA9	R	Gesture point 1 x coordinate（high byte）							
0xBDAA	R	Gesture point 1 y coordinate（low byte）							

0xBDAB	R	Gesture point 1 y coordinate（high byte）
0xBDAC~ 0xBEA7	R	Gesture point 2~64 coordinate (the number of coordinates is 0x814D)

4. Related Operation Timing

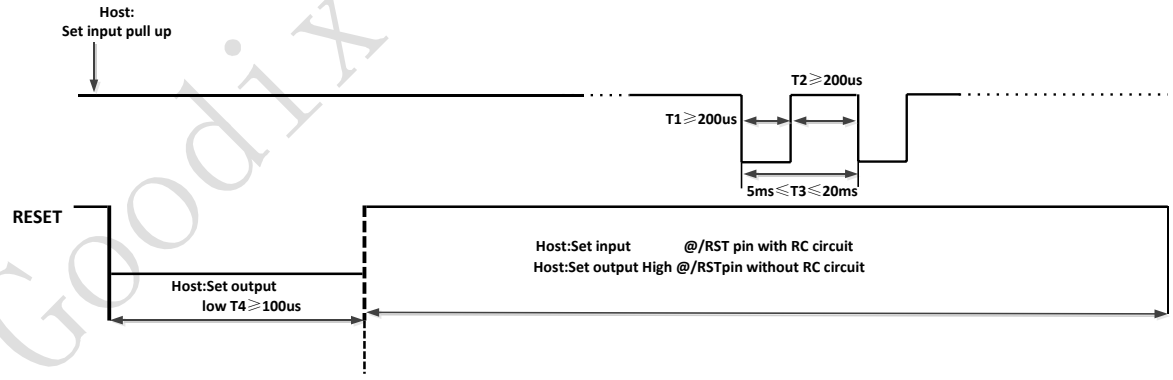
4.1 Power-on Sequence

After the main engine is powered on, it is necessary to control the pins such as AVDD, INT and RESET of GT1151Q. Please follow the following timing chart for the control sequence:



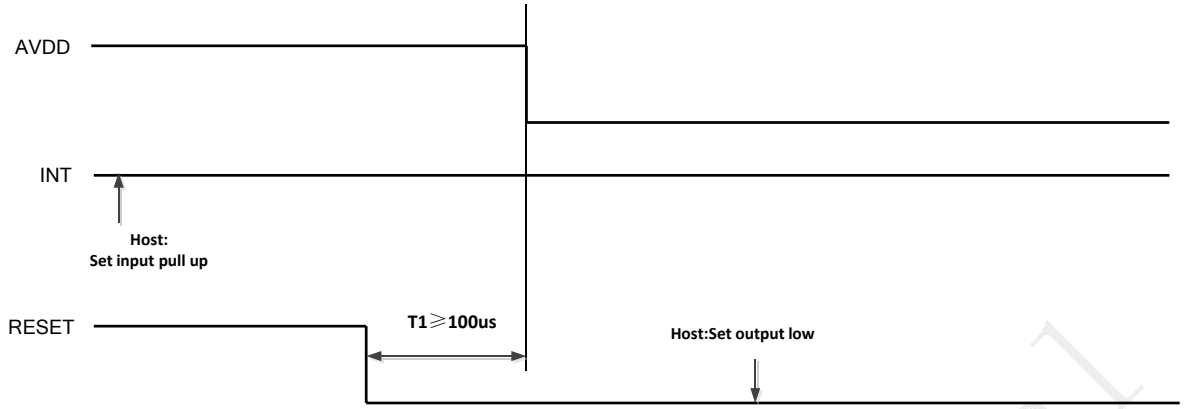
4.2 Reset Sequence

The host resets the GT1151Q with the following control timings:



4.3 Power Down Sequence

The control timing for the GT1151Q power down is as follows:



4.4 Power on to Send Configuration Information

During the host control GT1151Q power on, after the master pulls RESET high, it needs to delay 60ms to determine if it needs to send configuration information. If it receives a "0x01" request, it needs to send configuration information, otherwise, it does not need to send configuration information.

4.5 Master Response "INT Request"

1) Request Data Description:

Data	Description
0x00	ACK sent by the master to the touch IC
0x01	Request for the master to send configuration information
0x03	Request master reset GT1151Q
0xFF	DLE, no need to process
Others	Reserved, no need to process

2) How to respond to "0x01" request:

a) If the master has an INT interrupt and 0x814E=0, read the 0x8044 "Request" register, if 0x8044=0x01, it means "0x01" request.

b) Write all configuration information to the configuration information area via I2C (refer to section 3.2 Configuration Information).

c) Write 0x8044 to 0 via I2C to complete the response to the "0x01" request.

3) How to respond to "0x03" request:

a) If the master has an INT interrupt and 0x814E=0, please read 0x8044 "Request" register, if If 0x8044=0x03, it means it is "0x03" request.

b) Reset the GT1151Q according to the reset timings to complete the "0x03" request.

4.6 Register Dynamic Modification

GT1151Q supports dynamic modification of registers. When any register in the configuration area (including both regular and extended configurations) is modified according to the timing sequence in section 2, the corresponding Config_Chksum should be updated, and finally, Config_Fresh (0x813E) should be written as 1, otherwise the modification will not take effect. However, if registers outside the configuration area are rewritten, there is no need to change Config_Chksum and Config_Fresh.

5. Coordinate Reading

The master can read the coordinates by polling or by INT interrupt triggering, and the following steps can be taken when polling:

1) If the current buffer (buffer status is 1) is ready, the coordinates and key information of the corresponding number will be taken according to the number of fingers read and the key status.

2) If the buffer data (buffer status is 0) is not ready in 1, wait for 1ms to read again.

If the interrupt is triggered, the coordinates are read according to the above polling process.

The GT1151Q interrupt signal output timing is as follows

1) When standby, INT pin is high.

2) When the coordinates are updated, a falling edge is output.

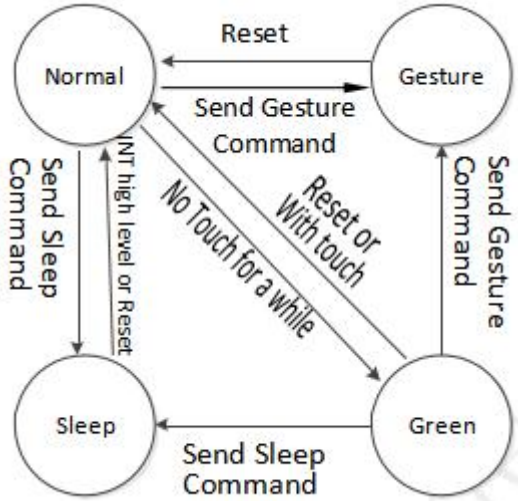
3) After the falling edge is output, the INT pin will remain low until the next cycle (the cycle can be determined by the configuration Refresh_Rate). Please read away the coordinates and write Buffer status(0x814E) to 0 within one cycle.

a) If the master does not read the coordinates within one cycle, the GT1151Q will continue to calculate the coordinates and output the falling edge in the next cycle, but will not update the contents of the coordinate information area.

b) If the master never reads the coordinates, the GT1151Q will keep hitting the INT pulse.

6. Operating mode switching

The GT1151Q working modes are divided into four types: Normal, Green, Sleep and Gesture, and the relationship between the various working states is shown in the figure below



6.1 Normal Mode

When GT1151Q is in Normal mode, the fastest coordinate refresh period is between 5 ms and 20 ms (depending on the setting of configuration information, the controllable cycle step length of configuration information is 1ms).

6.2 Green Mode

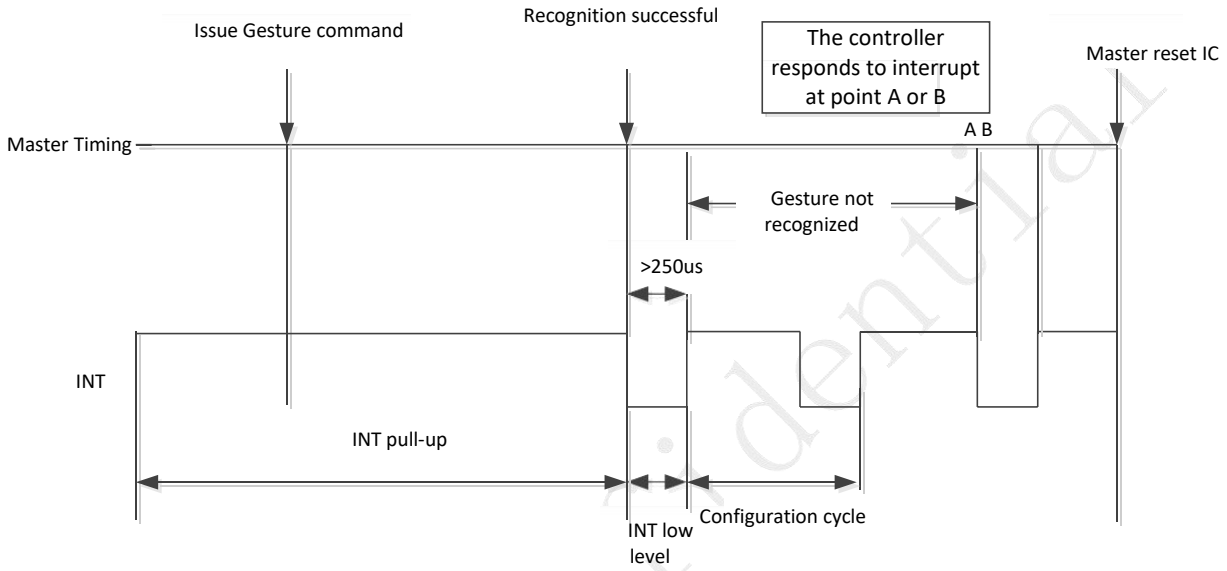
In the Normal mode state, there is no touch event for a period of time, and the GT1151Q will automatically switch to the Green mode to reduce power consumption. The time for GT1151Q to automatically enter the Green mode without touching can be set through configuration information, ranging from 0 to 0~14s, and the step is 1s, and the Green Mode can be turned off through configuration. In Green mode, the scanning period of GT1151Q is about 40ms, and if touch action is detected, it will automatically enter Normal mode.

6.3 Gesture Mode

If the main CPU passes the I2C command, GT1151Q can wake up by sliding the screen, double-clicking or writing specific lowercase letters on the screen after entering the Gesture mode.

In the Gesture mode, when GT1151Q detects that the finger slides on the screen for a sufficient length, double clicks, writes a specific character, or writes a custom character, INT will output a low-level pulse of more than 250us (the length of time can be modified by configuration). After detecting the pulse, the main controller wakes up and lights up the screen, and resets GT1151Q to exit the Gesture Mode.

The wake-up sequence is as follows:



6.4 Sleep Mode

The main CPU makes GT1151Q enter Sleep mode through I2C command. When the GT1151Q needs to exit the Sleep mode, the main CPU performs a reset operation on the GT1151Q. GT1151Q will enter Normal mode after reset. The time interval between issuing the command and resetting is required to be greater than 58 ms.

7. Gesture Mode Driver Modification

7.1 Enter Gesture mode After the Screen Off

a) Press the power button (or other buttons) to close the screen, and issue commands 0x08, 0x00 and 0xF8； to 0x8040~0x8042;

b) The modification when the mobile phone automatically turns off the screen is consistent with the modification when the screen is turned off by pressing the power button (or other buttons);

c) When the screen is turned off, sliding, double-clicking the screen or writing a specific character INT will output a pulse of more than 250us (configurable). After receiving the pulse, the master will read the value of 0x814C. If the wake-up condition is met, it will wake up and light up the screen, otherwise it will clear 0x814C and wait for the next pulse.

7.2 Enter Sleep Mode After the Screen Off

a) Press the power button (or other buttons) to close the screen, and issue commands 0x05, 0x00 and 0xFB： to 0x8040~0x8042;

b) The modification when the mobile phone automatically turns off the screen is consistent with the modification when the screen is turned off by pressing the power button (or other buttons);

c) In this mode, you can only wake up by the power key (or the home key).

7.3 Press Power Button (or home button) to Open the Screen.

Press the open button (or Home button) to open the screen in any mode, and reset the IC directly according to the reset sequence, and execute the reset process.

7.4 Recommended for Use with IR

If IR can be used to cooperate, when IR detects that there is an object blocking when the screen is turned off, it can enter the Sleep mode, which makes the power consumption less; If no occlusion is detected, enter the gesture wake-up mode, and the method of entering different modes is as above (reset is required before issuing the command).

8. Coordinate Reading and Verification in Gesture Mode

8.1 Coordinate Reading

In the Gesture mode, when the master reads 0x814C as non-zero, it can read gesture feature information or gesture coordinate information to describe the user's wake-up trajectory.

Gesture protocol type: the master reads the 0x814E register to obtain the gesture protocol type. The currently supported protocol types are as follows:

Bit7~bit4 (auxiliary information)

0x00：NULL;

0x01：Reserved;

0x02: Breakpoint position of multi-stroke gesture.

Bit3~bit0 (gesture coordinate information):

0x00：NULL;

0x01：Reserved;

0x02: gesture touch trajectory points (finger touch positions obtained at the same distance);

0x03: gesture touch feature track point (touch feature track point extracted according to the gesture type, for example: the feature track point of W is the starting point, three turning points, and the end point).

Gesture auxiliary information: the master reads the 0x814F register to obtain the length of the auxiliary information, and then reads 0x8150~0x81A0 with this length register to get some auxiliary information of the gesture.

Multi-pen gesture breakpoint location: the location where each stroke of the multi-pen gesture ends, and the stroke number starts from 0.

Gesture coordinate information: the master reads the 0x814D register to get the number of gesture track points, and then reads the 0xBDA8 ~ 0xBEA7 registers according to every 4 registers corresponding to a touch point, through which the user's real touch track can be depicted.

8.2 Coordinate and Trajectory Information Verification

In order to enhance the reliability of the system, all data that interacts with the main controller needs to be verified. If the main controller is reading the data, multiple readings should be conducted to ensure reliability.

There are two types of verification, one is to verify when the coordinates of the gesture poles are reported, and the other is to verify when the coordinates of the complete trajectory of the gesture are reported. When the master control receives an interrupt and needs to read the gesture information, it first judges whether the highest bit of 0x814E is 1, and if it is 1, it means that the verification function is needed.

Otherwise, it does not need the check function and reads the coordinates directly according to 8.1.

a) Gesture polar point coordinate reporting checksum (using 8-bit checksum)

Start from 0x814C address, read (buffer length + 5) u8 data continuously, sum up to 0 means the verification is passed, otherwise the verification is not passed, need to read again. If the verification still fails after re-reading, the data should be discarded, and the 0x814C flag should be cleared. When the FW receives this flag and clears it, the system returns to the state of re-recognizing touch.

b) Gesture track reporting verification (using 16-bit verification method)

Start from 0xBDA8 address, read (number of gesture touch points*2) 16-bit unsigned data continuously, then read a 16-bit checksum data from the starting address 0x819F, add up to 0 means the verification is passed, otherwise the verification is not passed and needs to be read again; if still wrong, discard this data and clear 0x814C (FW receives this marker to clear and return to the status of re-identifying touch state).

Note: The low byte of the checksum is 0x819F, and the high byte is 0x81A0. When the highest bit of 0x814E is set to 1, it indicates that the checksum verification function is enabled.

9. Version revision record

File Version	Revision Date	Revision
Rev.01	2014-07-17	First version released.