

FT5206 通讯协议说明

FT5206 通讯协议说明	
项目名称	
文档编号	
版本	0.2
日期	2010.12.2
所有者	王建波
保密级别	Confidential
签发	
备注	

本文档所有信息版权归普容电子有限公司所有，未获得普容电子有限公司授权，不得对本文档内容或部分内容进行修改，散布和使用。

Copyright © 2010, GCT Electronic Co., Ltd
All rights reserved

深圳宝安西乡镇黄田村光汇工业城 B 栋 2 楼
邮编：518000
电话：29966053-8004
传真：29966051
Email: wangjb@gct-sz.com
www.gct-sz.com

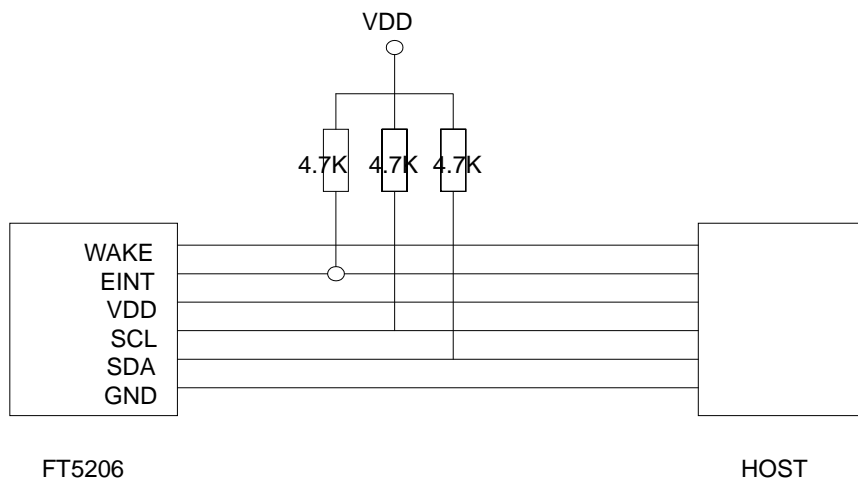
版本历史

日期	版本	变更	作者
2010.12.1	0.1	Initial draft	wangjb

FT5206 通讯协议说明

Section 1 General Discription

FT5206 采用 I2C 协议和 HOST 端进行通讯，其通讯连接结构如下：



FT5206 的电源电压范围是 2.8-3.3V。通过 6 个 pin, 连接到 HOST 端，分别是 WAKE，EINT，VDD，SCLK，SDA，GND。其中 SCLK，SDA 是 I2C 的通讯引脚；ENIT 是中断脚，以便 HOST 端正确的进行通讯；WAKE 有 2 个用途，烧录和唤醒。WAKE 脚只有处于高电平时，TP 才能正常工作，否则会一直处于烧录模式，可以通过 TP 端工具烧录程序进去。当进入休眠后，用于 TP 的唤醒。

需要注意的是：ENIT，SCLK，SDA 在使用时，HOST 需要接 4.7k 的上拉电阻。

表 1 IC 参数表

Sensor 通道	25
接口	I2C Slave 100K,200K,400K@3.3V
I2C 地址	0x70(8bit);0x38(7bit)
分辨率	最大 4096*4096(可以根据 LCD 的分辨率进行调整)
传输数据速率	10HZ-90HZ
反应时间	≅ 5ms
供电电压	2.8V-3.3V
IC 功耗	Normal Mode:4.5mA;Monitor Mode 3mA; Sleep Mode:0.1mA
运行温度	-45° to 85°
IC 封装	QFN48
IC 尺寸	5mm×5mm×0.85mm

Section2 I2C Bus Protocol

1、 I2C 总线的规范

I2C 总线的电气特性如下图

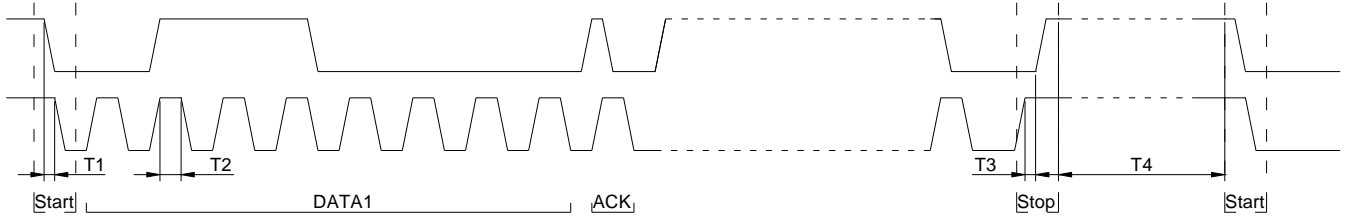


表 2 I2C Interface Timing Characteristic

Parameter	Min	Max	Unit
SCLK frequency		400	KHZ
Bus free time between a STOP and START condition (T4)	4.7		us
Hold time START condition(T1)	4		us
DATA setup time(T2)	250		us
Setup For STOP condition(T3)	4		us

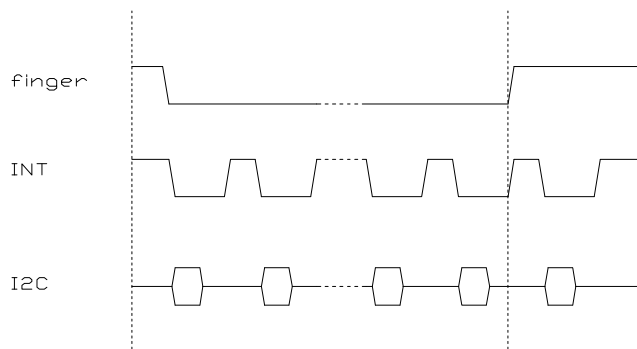
2、 INT 中断方式

FT5206 支持 2 种中断方式：trigger 和 polling。默认是 trigger。可以通过寄存器修改中断方式。

Trigger 是指当有手指按下时，EINT 引脚会不断产生下降沿脉冲，host 端在一次下降沿后触发一次中断，读取数据。

Polling 是指当有手指按下时，EINT 脚会一直保持拉低，产生一次中断触发后，HOST 端通过轮询的方式去读取数据。

当 FT5206 中断方式为 trigger 时，大致波形如下：



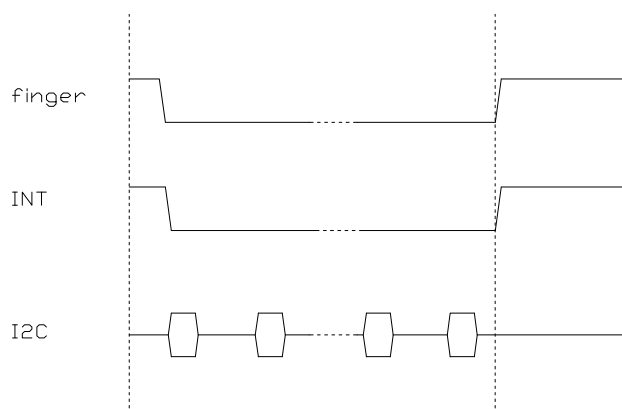
INT 的频率默认是 60HZ，可以根据需要调整。HOST 端建议触发方式调整为下降沿触发。当手

抬起时，TP 仍然会发一次中断，以便 HOST 端可以读到手抬起的信息。

注意:

- MTK 平台的某些手机 IC 不支持边沿触发，只支持电平触发。在使用 trigger 方式时，需要通过不停的翻转触发电平来实现触发。在低电平触发时，在中断函数中把触发方式改为高电平触发，而在高电平触发是，在中断函数中把触发方式改为低电平触发。
- 当 INT 来时，如果 HOST 端没有通过 I2C 正确的读取数据，在手抬起后，INT 仍然会有波形出来。原因是 FT5206 的数据也是放在一个长度为 10 的 buffer 中，如果此 buffer 中有数据，就会不断对 INT 操作，让 HOST 端进行 I2C 读写。

当 FT5206 中断方式为 polling 时，大致波形如下：



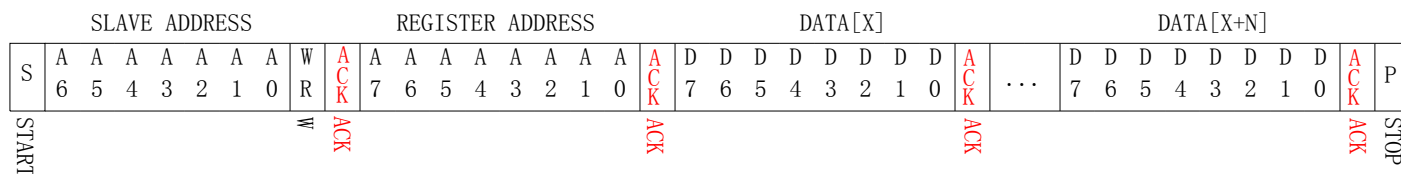
此种方式下，HOST 会隔一段时间去读取一次数据，通讯的速度完全是由 HOST 来决定。

3、 I2C 协议说明

FT5206 采用标准寄存器方式，读写方式如下。

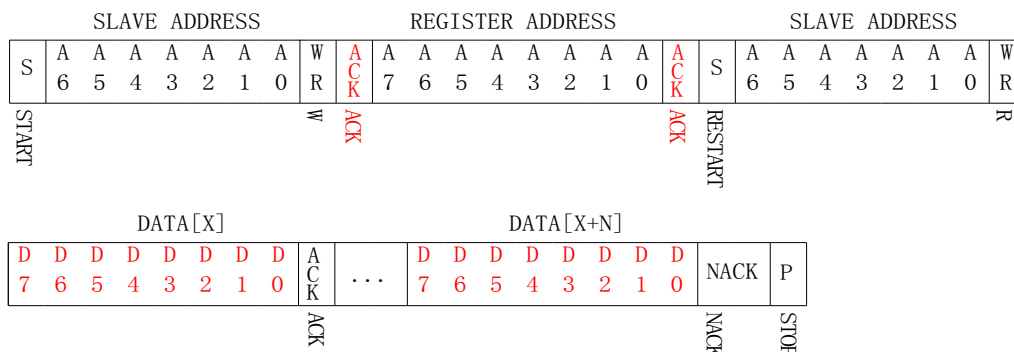
I2C write

流程: START+SLAVE ADDRESS+REGISTER ADDRESS+DATA+STOP



I2C read

流程: START+SLAVE ADDRESS+REGISTER ADDRESS+RESTART+SLAVE ADDRESS
+DATA+STOP



4、寄存器列表

表 3 IC 参数表 （提示：红色表示寄存器无法使用；黑色字体为正常使用寄存器）

NUM	Name	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	R/W
0x00	DEVIDE_MODE		Device Mode[2:0]							R
0x01	GEST_ID	Gesture ID[7:0]								R
0x02	TD_STATUS	Frame remaining				Number of touch points[3:0]				R
0x03	TOUCH1_XH	1 st Event Flag				1 st Touch X Position[11:8]				R
0x04	TOUCH1_XL	1 st Touch X Position[7:0]								R
0x05	TOUCH1_YH	1 st Touch ID[3:0]				1 st Touch Y Position[11:8]				R
0x06	TOUCH1_YL	1 st Touch Y Position[7:0]								R
0x07	TOUCH1_WEIGHT	1 st Touch Weight[7:0]								R
0x08	TOUCH1_MISC	1 st Touch Area[3:0]				1 st Touch Direction[1:0]		1 st Touch Speed[1:0]		R
0x09	TOUCH2_XH	2 nd Event Flag				2 nd Touch X Position[11:8]				R
0x0A	TOUCH2_XL	2 nd touch X Position[7:0]								R
0x0B	TOUCH2_YH	2 nd Touch ID[3:0]				2 nd Touch Y Position[11:8]				R
0x0C	TOUCH2_YL	2 nd Touch Y Position[7:0]								R
0x0D	TOUCH2_WEIGHT	2 nd Touch Weight[7:0]								R
0x0E	TOUCH2_MISC	2 nd Touch Area[3:0]				2 nd Touch Direction[1:0]		2 nd Touch Speed[1:0]		R
0x0F	TOUCH3_XH	3 rd Event Flag				3 rd Touch X Position[11:8]				R
0x10	TOUCH3_XL	3 rd Touch X Position[7:0]								R
0x11	TOUCH3_YH	3 rd Touch ID[3:0]				3 rd Touch Y Position[11:8]				R
0x12	TOUCH3_YL	3 rd Touch Y Position[7:0]								R
0x13	TOUCH3_WEIGHT	3 rd Touch Weight[7:0]								R
0x14	TOUCH3_MISC	3 rd Touch Area[3:0]				3 rd Touch Direction[1:0]		3 rd Touch Speed[1:0]		R
0x15	TOUCH4_XH	4 th Event Flag				4 th Touch X Position[11:8]				R
0x16	TOUCH4_XL	4 th Touch X Position[7:0]								R
0x17	TOUCH4_YH	4 th Touch ID[3:0]				4 th Touch Y Position[11:8]				R
0x18	TOUCH4_YL	4 th Touch Y Position[7:0]								R
0x19	TOUCH4_WEIGHT	4 th Touch Weight[7:0]								R
0x1A	TOUCH4_MISC	4 th Touch Area[3:0]				4 th Touch Direction[1:0]		4 th Touch Speed[1:0]		R
0x1B	TOUCH5_XH	5 th Event Flag				5 th Touch X Position[11:8]				R
0x1C	TOUCH5_XL	5 th Touch X Position[7:0]								R
0x1D	TOUCH5_YH	5 th Touch ID[3:0]				5 th Touch Y Position[11:8]				R
0x1E	TOUCH5_YL	5 th Touch Y Position[7:0]								R
0x1Fh	TOUCH5_WEIGHT	5 th Touch Weight[7:0]								R
0x20h	TOUCH5_MISC	5 th Touch Area[3:0]				5 th Touch Direction[1:0]		5 th Touch Speed[1:0]		R

0x21	TOUCH6_XH	6 th Event Flag		6 th Touch X Position[11:8]	R	
0x22	TOUCH6_XL	6 th Touch X Position[7:0]			R	
0x23	TOUCH6_YH	6 th Touch ID[3:0]		6 th Touch Y Position[11:8]	R	
0x24	TOUCH6_YL	6 th Touch Y Position[7:0]			R	
0x25	TOUCH6_WEIGHT	6 th Touch Weight[7:0]			R	
0x26	TOUCH6_MISC	6 th Touch Area[3:0]		6 th Touch Direction[1:0]	6 th Touch Speed[1:0]	R
0x27	TOUCH7_XH	7 th Event Flag		7 th Touch X Position[11:8]	R	
0x28	TOUCH7_XL	7 th Touch X Position[7:0]			R	
0x29	TOUCH7_YH	7 th Touch ID[3:0]		7 th Touch Y Position[11:8]	R	
0x2A	TOUCH7_YL	7 th Touch Y Position[7:0]			R	
0x2B	TOUCH7_WEIGHT	7 th Touch Weight[7:0]			R	
0x2C	TOUCH7_MISC	7 th Touch Area[3:0]		7 th Touch Direction[1:0]	7 th Touch Speed[1:0]	R
0x2D	TOUCH8_XH	8 th Event Flag		8 th Touch X Position[11:8]	R	
0x2E	TOUCH8_XL	8 th Touch X Position[7:0]			R	
0x2F	TOUCH8_YH	8 th Touch ID[3:0]		8 th Touch Y Position[11:8]	R	
0x30	TOUCH8_YL	8 th Touch Y Position[7:0]			R	
0x31	TOUCH8_WEIGHT	8 th Touch Weight[7:0]			R	
0x32	TOUCH8_MISC	8 th Touch Area[3:0]		8 th Touch Direction[1:0]	8 th Touch Speed[1:0]	R
0x33	TOUCH9_XH	9 th Event Flag		9 th Touch X Position[11:8]	R	
0x34	TOUCH9_XL	9 th Touch X Position[7:0]			R	
0x35	TOUCH9_YH	9 th Touch ID[3:0]		9 th Touch Y Position[11:8]	R	
0x36	TOUCH9_YL	9 th Touch Y Position[7:0]			R	
0x37	TOUCH9_WEIGHT	9 th Touch Weight[7:0]			R	
0x38	TOUCH9_MISC	9 th Touch Area[3:0]		9 th Touch Direction[1:0]	9 th Touch Speed[1:0]	R
0x39	TOUCH10_XH	10 th Event Flag		10 th Touch X Position[11:8]	R	
0x3A	TOUCH10_XL	10 th Touch X Position[7:0]			R	
0x3B	TOUCH10_YH	10 th Touch ID[3:0]		10 th Touch Y Position[11:8]	R	
0x3C	TOUCH10_YL	10 th Touch Y Position[7:0]			R	
0x3D	TOUCH10_WEIGH T	10 th Touch Weight[7:0]			R	
0x3E	TOUCH10_MISC	10 th Touch Area[3:0]		10 th Touch Direction[1:0]	10 th Touch Speed[1:0]	R
0x3F	Reserved					
...	...					
0x7F	Reserved					
0x80	ID_G_THGROUP	valid touching detect threshold.			R/W	
0x81	ID_G_THPEAK	valid touching peak detect threshold.			R/W	
0x82	ID_G_THCAL	the threshold when calculating the focus of touching.			R/W	
0x83	ID_G_THWATER	the threshold when there is surface water.			R/W	

0x84	ID_G_THTEMP	the threshold of temperature compensation.		R/W
0x85	ID_G_THDIFF	the threshold whether the coordinate is different from the original		R/W
0x86	ID_G_CTRL		Power control mode[1:0]	R/W
0x87	ID_G_TIME_ENTE R _MONITOR	The timer of entering monitor status		R/W
0x88	ID_G_PERIODACT IVE		Period Active[3:0]	R/W
0x89	ID_G_PERIOD MONITOR	The timer of entering idle while in monitor status		R/W
0x8A	Reserved			
...	...			
0x91	Reserved			
0x92	ID_G_L_R_OFFSE T	Maximum of the distance of X axis of the valid move up, move down gesture.		R/W
0x93	ID_G_U_D_OFFSE T	Maximum of the distance of Y axis of the valid move left, move right gesture.		R/W
0x94	ID_G_DISTANCE _LEFT_RIGHT	Minimum of the distance of X axis of the valid move left, move right gesture.		R/W
0x95	ID_G_DISTANCE _UP_DOWN	Minimum of the distance of Y axis of the valid move up, move down gesture.		R/W
0x96	ID_G_RADIAN _VALUE	Minimum of angle of Double Right/Left Rotation gesture		R/W
0x97	ID_G_ZOOM _DIS_SQR	The threshold of valid Zoom In, Zoom Out gesture		R/W
0x98	ID_G_MAX_X_HI GH	maximum resolution of X axis high byte		R
0x99	ID_G_MAX_X_LO W	maximum resolution of X axis low byte		R
0x9A	ID_G_MAX_Y_HI GH	minimum resolution of Y axis high byte		R
0x9B	ID_G_MAX_Y_LO W	minimum resolution of Y axis low byte		R
0x9C	ID_G_K_X_HIGH	the resolution coefficient of X axis high byte		R
0x9D	ID_G_K_X_LOW	the resolution coefficient of X axis low byte		R
0x9E	ID_G_K_Y_HIGH	the resolution coefficient of Y axis high byte		R
0x9F	ID_G_K_Y_LOW	the resolution coefficient of Y axis low byte		R
0xA0	ID_G_AUTO_CLB _MODE	auto calibration mode		R
0xA1	ID_G_LIB_VERSI ON_H	Firmware Library Version H byte		R
0xA2	ID_G_LIB_VERSI	Firmware Library Version L byte		R

	ON_L		
0xA3	ID_G_CHIP	Chip vendor ID	R
0xA4	ID_G_MODE	the interrupt mode to host	R
0xA5	ID_G_PMODE	Power Consume Mode	
0xA6	ID_G_FIRMID	Firmware ID	R
0xA7	ID_G_STATE	Running State	
0xA8	ID_G_FT5201ID	CTPM Vendor ID	R
0xA9	ID_G_ERR	Error Code	R
0xAA	ID_G_CLB	Configure TP module during calibration in Test Mode	R/W
0xAB	ID_G_STATIC_TH	The threshold of touching static status	R/W
0xAC	ID_G_MID_SPEED_TH	The threshold of touching normal speed status	R/W
0xAD	ID_G_HIGH_SPEED_TH	The threshold of touching high speed status	R/W

1.1.1 DEVICE_MODE

This register is the device mode register, configure it to determine the current mode of the chip.

Address	Bit Address	Register Name	Description	Default
0x00	6:4	Device Mode[2:0]	000 Normal operating Mode 001 System Information Mode (Reserved) 100 Test Mode – read raw data (Reserved)	000

1.1.2 GEST_ID

This register describes the gesture of a valid touch.

Address	Bit Address	Register Name	Description	Default
0x01	7:0	Gesture ID[7:0]	见下表	0x00

FT5206 默认只支持缩放手势，其他手势如有需要请联系相关人员，具体见下表。

表 4 手势列表

	手势名称	定义	备注
默认提供	MoveZoomIn	0x48	放大
	MoveZoomOut	0x49	缩小
默认不提供，需要请说明	MoveUp	0x10	向上拖动
	MoveDown	0x18	向下拖动
	MoveLeft	0x1c	向左拖动
	MoveRight	0x14	向右拖动
	MoveDoubleLeftRotation	0x81	向左旋转
	MoveDoubleRightRotation	0x82	向右旋转
不提供，可以协商添加	MoveDoubleClick	0x22	
	MoveSingleClick	0x20	
	MoveNone	0x00	

1.1.3 TD_STATUS

This register is the Touch Data status register.

Address	Bit Address	Register Name	Description	Default
0x02	3:0	Number of touch points[3:0]	当前触摸的手指数，最大值和 firmware 中的设定有关。	0
	7:4	Frame remaining [7:4]	Frame remaining after host's reading Range from 0 to 9	

1.1.4 TOUCHn_XH (n:1-10)

This register describes MSB of the X coordinate of the nth touch point and the corresponding event flag.

Address	Bit Address	Register Name	Description	Default
0x03 0x09 0x0F	7:6	Event Flag	00: 按下 01: 抬起 10: 保持 11: 无事件	
0x15	5:4		reserved	
0x1B	3:0	Touch X Position [11:8]	MSB of Touch X Position in pixels	0xF

1.1.5 TOUCHn_XL (n:1-10)

This register describes LSB of the X coordinate of the nth touch point.

Address	Bit Address	Register Name	Description	Default
0x04 0x0A 0x10 0x16 0x1C	7:0	Touch X Position [7:0]	LSB of the Touch X Position in pixels	0xFF

1.1.6 TOUCHn_YH (n:1-10)

This register describes MSB of the Y coordinate of the nth touch point and corresponding touch ID.

Address	Bit Address	Register Name	Description	Default
0x05 0x0B 0x11	7:4	Touch ID[3:0]	Touch ID of Touch Point	
0x17 0x1D	3:0	Touch Y Position [11:8]	MSB of Touch Y Position in pixels	0xFF

Touch ID 是按照如下的规则来定义的。

最先按下的触摸点 ID 为 0，后依次增加，最大到 4。当有触摸点抬起时，不影响其他触摸点 ID。同一个触摸点的数据并不是固定在同一寄存器中，靠后的触摸点在其他触摸点抬起后，在寄存器中放置位置也会往前。例如：按下 2 个点。第一个点 ID 是 0，第二个点 ID 是 1。第一个点数据放在 0x03-0x06 寄存器中，第二个点的数据放置在 0x09-0x0C 寄存器中。当第一个点抬起后，第二个点 ID 依然是 1，而且数据放置位置也移动到 0x03-0x06 寄存器。在第一个点位置再次按下，其 ID 值还是 0，但其数据是放置在 0x09-0x0C 寄存器中。

1.1.7 TOUCHn_YL (n:1-10)

This register describes LSB of the Y coordinate of the nth touch point.

Address	Bit Address	Register Name	Description	Default
0x06 0x0C 0x12 0x18 0x1E	7:0	Touch X Position [7:0]	LSB of The Touch Y Position in pixels	0xFF

1.1.8 TOUCHn_WEIGHT (n:1-10)

This register describes weight of the nth touch point.

Address	Bit Address	Register Name	Description	Ddfault
0x07 0x0D 0x13 0x19 0x1D	7:4	Touch Area[7:4]	The valid touching area	
	3:2	Touch Direction	0: up 1:down 2:left 3:right	
	1:0	Touch Speed	0: static 1: normal speed 2: high speed	

1.1.9 TOUCHn_MISC (n:1-10)

This register describes the miscellaneous information of the nth touch point.

Address	Bit Address	Register Name	Description	Default
0x08 0x0E 0x14 0x1A 0x1E	7:0	Touch Weight[7:0]	Valid points in X direction× Valid points in Y direction/2	

1.1.10 ID_G_THGROUP

This register describes valid touching detect threshold.

Address	Bit Address	Register Name	Description	Default
0x80	7:0	ID_G_THGROUP	触摸有效参数阈值。当触摸 diff 大于此值，才表示有效触摸。实际参数值等于寄存器*4。降低此值可以提高灵敏度。	280/4

1.1.11 ID_G_THPEAK

This register describes valid touching peak detect threshold.

Address	Bit Address	Register Name	Description	Default
0x81	7:0	ID_G_THPEAK	过滤噪音阈值。当噪音值大于此值，就会有干扰点产生。建议不要修改此值。	60

1.1.12 ID_G_THCAL

This register describes threshold when calculating the focus of touching.

Address	Bit Address	Register Name	Description	Default
0x82	7:0	ID_G_THCAL	坐标计算阈值。建议不要修改此值。	16

1.1.13 ID_G_THWATER

This register describes threshold when there is surface water.

Address	Bit Address	Register Name	Description	Default
0x83	7:0	ID_G_THWATER	防水参数。不建议改动。注意负号。	-60

1.1.14 ID_G_THTEMP

This register describes threshold of temperature compensation.

Address	Bit Address	Register Name	Description	Default
0x84	7:0	ID_G_THTEMP	温度参数。不建议改动。	400

1.1.15 ID_G_THDIFF

This register describes threshold whether the coordinate is different from the original.

Address	Bit Address	Register Name	Description	Default
0x85	7:0	ID_G_THDIFF	实际值/16。不建议改动。	160

1.1.16 ID_G_CTRL

This register describes the run mode of microcontroller controlled by host

Address	Bit Address	Register Name	Description	Default
0x86	0	ID_G_CTRL	0: not auto jump 1:auto jump	1

1.1.17 ID_G_TIMEENTERMONITOR

This register describes the time delay value when entering monitor status.

Address	Bit Address	Register Name	Description	Default
0x87	7:0	ID_G_TIME ENTERMONITOR	ACTIVE 模式切向 MONITOR 模式的延时。单位是用作延时的计数器的个数。	200

FT5206 有 3 种功耗模式，具体定义如下表。

表 5 功耗模式 PMODE

模式名称	数值	描述	实测功耗
ACTIVE	0x00	全速扫描模式，有手指按下产生触摸进入此模式，产生触摸中断/INT, 让 Host 来读取触摸数据.	4mA
MONITOR	0x01	触摸监测模式，低速扫描，不能产生触摸中断/INT. 有触摸发生后自动跳转到 ACTIVE 模式，也可以有 Host 配置进入其它功耗模式.	3mA
HIBERNATE	0x03	休眠模式. 系统主要电路停止工作，保留供电电压， 功耗在 100uA 以下. 该模式下无法进行 I2C 通信，模组需要由 WAKE 信号唤醒，唤醒后会自动进入 MONITOR 模式。	100uA

当有手指触摸时，FT5206 会自动进入 ACTIVE 模式。当手抬起后，不会立即进入 MONITOR 模式，而是有延时，延时时间就是由 TIMEENTERMONITOR 来控制。这里的延时参数并不是 s 或 ms，而是用于循环的次数。

1.1.18 ID_G_PERIODACTIVE

This register describes the period of active status.

Address	Bit Address	Register Name	Description	Default
0x88	4:0	ID_G_PERIOD ACTIVE	ACTIVE 模式的扫描周期	6
	7:4			

此寄存器影响 ACTIVE 模式下，扫描周期和报点频率。6 并不是准确表示 60HZ，而是表示在 60HZ 左右。最小值是 1，最大值是 8，，可以达到 85HZ 左右。

1.1.19 ID_G_PERIODMONITOR

This register describes period of monitor status, it should not less than 30.

Address	Bit Address	Register Name	Description	Default
0x89	4:0	ID_G_PERIOD ACTIVE		40

此寄存器影响 MONITOR 模式下的扫描周期。这里 40 和 PERIODACTIVE 不同，并不是 40HZ，而且 MONITOR 下是不报数据出来的，不存在报点频率。

1.1.20 ID_G_LEFT_RIGHT_OFFSET

This register is only used in the mode of continuous reporting gesture to host while valid gesture produced.

Address	Bit Address	Register Name	Description	Default
0x92	7:0	ID_G_LEFT_RIGHT_OFFSET	上下拖动手势参数	20

1.1.21 ID_G_UP_DOWN_OFFSET

This register is only used in the mode of continuous reporting gesture to host while valid gesture produced.

Address	Bit Address	Register Name	Description	Default
0x93	7:0	ID_G_UP_DOWN_OFFSET	左右拖动手势参数	20

1.1.22 ID_G_DISTANCE_LEFT_RIGHT

This register is only used in the mode of continuous reporting gesture to host while valid gesture produced.

Address	Bit Address	Register Name	Description	Default
0x94	7:0	ID_G_DISTANCE_LEFT_RIGHT	左右拖动手势参数	50

1.1.23 ID_G_DISTANCE_UP_DOWN

This register is only used in the mode of continuous reporting gesture to host while valid gesture produced.

Address	Bit Address	Register Name	Description	Default
0x95	7:0	ID_G_DISTANCE_UP_DOWN	上下拖动手势参数	50

1.1.24 ID_G_RADIAN_VALUE

This register is only used in the mode of continuous reporting gesture to host while valid gesture produced.

Address	Bit Address	Register Name	Description	Default
0x96	7:0	ID_G_RADIAN_VALUE	旋转手势参数	10

1.1.25 ID_G_ZOOM_DIS_SQR

This register describes minimum square of distance while zoom in or out used in both reporting mode..

Address	Bit Address	Register Name	Description	Default
0x97	7:0	ID_G_ZOOM_DIS_SQR	缩放手势参数	50

ZOOM_DIS_SQR 表示的是当 2 点间的距离改变超过此值时，就产生放大缩小手势。参数并不是实际坐标值，需要根据实际需要调整。值越大，报出的缩放手势频率越低。

1.1.26 ID_G_AUTO_CLB_MODE

This register describes auto calibration mode.

Address	Bit Address	Register Name	Description	Default
0xA0	7:0	ID_G_AUTO_CLB_MODE	0x00: enable auto calibration 0xFF: disable auto calibration	0xFF

1.1.27 ID_G_MODE

This register describes whether the host is worked in polling mode or whether it is worked in trigger mode, see details in Section 1.3 Interrupt signals from CTPM to Host.

Address	Bit Address	Register Name	Description	Default
0xA4	7:0	ID_G_MODE	中断模式寄存器：1,trigger; 0,polling。	1

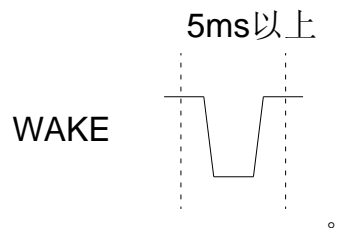
详细请参看中断说明。

1.1.28 ID_G_PMODE

This register describes the power consumption mode of the TPM when in running status.

Address	Bit Address	Register Name	Description	Default
0xA5	7:0	ID_G_PMODE	功耗模式寄存器。 0: active 1: monitor 3: hibernate(deep sleep) 详见表 7。	1

通过此寄存器可以在不同功耗模式切换。进入 hibernate 模式后，无法返回其他功耗模式，只有通过 wake 脚进行硬件唤醒，具体唤醒如下图。



而且需保证 WAKE 脚一直为高，除了唤醒的时候。

1.1.29 ID_G_FIRMWARE_ID

This register describes the firmware id of the application.

Address	Bit Address	Register Name	Description	Default
0xA6	7:0	ID_G_FIRMWARE_ID	固件 ID	5

1.1.30 ID_G_STATE

This register is used to configure the run mode of TPM.

Address	Bit Address	Register Name	Description
Op,A7h	7:0	ID_G_STATE	0: configure 1: work 3: factory 4: auto calibration

工作模式有 4 种, work, factory, clb 等。Work 模式是指正常工作模式, 输出坐标信息都在此模式下进行。Factory 模式是工厂模式, 用于对 TP 进行调试时使用。auto calibration 模式是自动校验模式, 会自动对当前 TP 进行校验, 得到合适的参数。只有 work 模式下, 客户才能正常使用 TP。

Section 3 Q&A

1. Start 后, 设备地址都没有 ACK。

答: 当前我司 TP 的地址默认都是 0x70(8bit),0x38(7bit)。请确认 host 端软件中的 I2C 设备地址是 7bit 还是 8bit。因为 linux 系统默认是 7bit, 通讯时会自动左移一位成 8bit。

2. 贵司的 I2C 是否是标准 I2C。

答: 物理层和标准 I2C 相同, 但有限制。触摸数据是在一个数据包里, 这个数据包并不是寄存器, 只能读不能写, 而且只能连续的读。

3. 26BYTES 的数据包数据是否需要全部读完

答: 数据包中的 26BYTES 数据在有些平台上测试, 发现可以不读完, 但有些平台上发现必须全部读完。所以建议把 26BYTES 的数据全部读完比较好。

4. 为什么线路连接好后, 上电后, INT 脚的电平总是低。

答: 模组的 INT 脚, 在 IC 内部及 FPC 上都是没有拉高的, 需要在 HOST 端拉高。

5. 为什么线路连接正常, 而且程序也是按照协议说明写好代码, SCL, SDA 都没有信号, 一直是低电平。

答: 和 4 中原因相同, SCL, SDA 都需要拉高。

6. 协议中 0XF9 的是寄存器地址, 还是什么

答: 数据包不是在寄存器中, 没有寄存器地址, 0XF9 仅仅是一个标识, 表示当前要读取的是数据包而不是寄存器。读取寄存器是需要另外一种协议。也可以当 0XF9 当成寄存器地址来。

7. I2C 读取数据包的流程

答: start->设备地址 0x70->0xf9(->stop)->restart->设备地址 0x71->读取 26BYTES->stop。写完 0xf9 后面的 stop 可以要也可以不需要。

8. 写完设备地址 0x70, 写完 0xf9 后, 为什么还是读不到数据

答: 请查看代码中, 写完 0xf9 后, 是否有 restart, 写设备地址 0x71。写的过程和读的过程是分开的, 需要 restart。

9. 可以读到数据, 但坐标总是不正确。

答: 坐标是 12bit 的, 需要 2 个 BYTES。例如 X1 的坐标就是 reg[5]和 reg[6]两个 BYTES 构成。需要注意有 2 个方面。A. 数据总是高位在前, 低位在后, reg[5]表示的高 8 位, reg[6]表示的低 8 位。B. 高 8 位中只有 4 位是用于坐标, 最高的 4 位是用于其它用途, 所以需要把最高 4 位滤掉。举例: x1 的坐标就是((reg[5]&0x0f)<<8)|reg[6]。Y 坐标也类似。

10. 为什么 I2C 可以读取数据, 但 X 坐标是正确的, Y 坐标都不正确。而且 CRC 校验也不正

确

答：请确认系统是否存在其它 I2C 设备，如果有请确认其 I2C 设备地址是否和 TP 相同。如果相同而且此设备没有从系统中移除，就会出现 2 个相同地址的 I2C 设备，导致 I2C 通讯过程中总线出错，因为 2 个设备都有可能相应 HOST。

11. I2C 可以通讯上而且读取正常，但出现画线的时候延时很严重，有时候手都抬起来了，手机还在画线。是否和 I2C 通讯有关

答：和系统,I2C 通讯都有些关系。存在一种配合关系。I2C 是通过中断的方式来告诉 HOST 端去读取 I2C。但中断会强制中断系统当前的工作，而执行中断。如果中断频率过高，会非常影响系统的运行效率，特别是系统性能较差的情况下。中断函数的工作内容就是去读取 I2C 数据，中断执行时间的长短跟 I2C 数据读取的时间长短决定的。MTK 系统的处理流程如下：中断来延时 1ms 读取数据，把需要的数据放置到 buffer 中。系统会隔一段时间从 buffer 中取数据。如果系统处理时间过长，下一次中断来了，更新了 buffer，系统在中断后返回继续处理上次没有处理完成的任务。等系统处理完任务，到再一次中断来更新了 buffer，系统才会从 buffer 取数据，导致丢数据的现象。而且出现 TP 的数据报过去后并不能及时的处理。所以在不能提升系统处理速度的情况下，提升 I2C 数据的读取速率非常重要。尽量要求客户把 I2C 通讯速度提高到 200K。

Section4 Sample Code

本样例程序采用的 GPIO 口模拟 I2C 来和模组通讯，硬件 I2C 的参考此代码进行移植。

```
#define SCK_TIME 2
#define START_TIME 4
#define ACK_TIME 3
#define BEFOR_RS_DELAY 5

void liC_Init(void)
{
    rPCONF=rPCONF&0xffff00; //设置为 GPIO 口
    rPCONF=rPCONF|0x000055;
    rPUPF=rPUPF&(~0x03);
    rPDATF=rPDATF|0x03;
}

void SCK_LOW(void)
{
    rPDATF= rPDATF&(~0x01); //SCK 置低
}

void SCK_HIGH(void)
{
    rPDATF= rPDATF|0x01; //SCK 置高
}

void SDA_LOW(void)
{
    rPDATF=(rPDATF&(~0x02)); // SDA 置低
    rPCONF=(rPCONF&(~0x0c))|0x04; // SDA 置为 output
}

void SDA_HIGH(void)
{
    rPDATF= rPDATF|0x02; // SDA 置高
    rPCONF= (rPCONF&(~0x0c))|0x04; // SDA 置为 output
}

int ACK_GET(void)
{
    int data;
    rPCONF=( rPCONF&(~0x0c)); //SDA 置为 input
    LCD_DelayUs(ACK_TIME);
    SCK_HIGH();
    LCD_DelayUs(1);
}
```

```
data=rPDATF&0x02;    //得到 SDA 的电平
LCD_DelayUs(1);
SCK_LOW();
return data;
}

void ACK_SET(int ack)
{
    LCD_DelayUs(SCK_TIME);
    if(ack)
    {
        rPDATF= rPDATF|0x02;    //SDA 置高
        rPCONF=(rPCONF&(~0x0c))|0x04;    //SDA 置高输出
    }
    else
    {
        rPDATF= rPDATF&(~0x02); //SDA 置低
        rPCONF=(rPCONF&(~0x0c))|0x04;    //SDA 置高输出
        LCD_DelayUs(2);
    }
    SCK_HIGH();
    LCD_DelayUs(3);    //12us
    SCK_LOW();
}

void lic_Start(void)
{
    SDA_HIGH();
    SCK_HIGH();
    LCD_DelayUs(SCK_TIME);
    SDA_LOW();
    LCD_DelayUs(START_TIME);
    SCK_LOW();
}

void lic_Stop(void)
{
    LCD_DelayUs(SCK_TIME);
    SDA_LOW();
    SCK_HIGH();
    LCD_DelayUs(SCK_TIME);
    SDA_HIGH();
}
```

```
void I2C_write_byte(U8 data)
{
    U8 mask;
    for(mask=0X80;mask!=0;mask=mask>>1)
    {
        LCD_DelayUs(SCK_TIME);
        if(data&mask)
            SDA_HIGH();
        else
            SDA_LOW();
        LCD_DelayUs(SCK_TIME - 2);
        SCK_HIGH();
        LCD_DelayUs(SCK_TIME);
        SCK_LOW();
    }
}

U8 I2C_read_byte(void)
{
    U8 mask;
    U8 data=0,val;

    rPCONF=( rPCONF&(~0x0c));
    for(mask=0X80;mask!=0;mask=mask>>1)
    {
        LCD_DelayUs(SCK_TIME*5);
        SCK_HIGH();
        val = rPDATF&0x02;
        if(val )
        {
            data = data|mask;
        }
        else
        {
            data = data&(~mask);
        }
        LCD_DelayUs(SCK_TIME*5);
        SCK_LOW();
    }
    return data;
}

//
```