Application Note for CTPM

Application Note for CTPM	
Project name	Touch panel
Document ref	[Document ref]
Version	0.8
Release date	18 Aug 2010
Owner	Xiaoxu Du
Classification	Confidential
Distribution List	[Distribution list]
Approval	

R3-B4-A, South Area, Shenzhen Hi-Tech Industrial Park,
Shenzhen, Gungdong, P.R. China
ZIP :518057
T +86 755 26588222
F +86 755 26712499
E support@focaltech-systems.com

www.focaltech-systems.com

Revision History

Date	Version	List of changes	Author + Signature
18 Jan, 2010	0.1	Initial draft	Xiaoxu Du
17 Mar,2010	0.2	Add raw data protocol	Xiaoxu Du
22 Mar,2010	0.3	Add system information protocol	Xiaoxu Du
26 Mar,2010	0.4	Add calibration related parameters	Xiaoxu Du
08 May,2010	0.5	Add information to operating mode	Xinming Wang
07 Jul, 2010	0.6	Change Protocol and add information	Yunfeng Yuan
18 Aug, 2010	0.7	Modified to release version	Xiaoxu Du
22 Dec, 2010	0.8	Modify explanation for register 0xA4	Xiaoxu Du

Table of Contents

Terminology
CTP – Capacitive touch panel
CTPM – Capacitive touch panel module

1 I²C Interface

1.1 CTPM interface to Host

Figure 1-1 shows how CTPM communicates with the Host，there are three kind of communication between CTPM and Host，we will introduce each communication in this section.

Transfer the data via I²C

Send interrupt when there is a valid touch

Host send Wakeup signal to CTPM

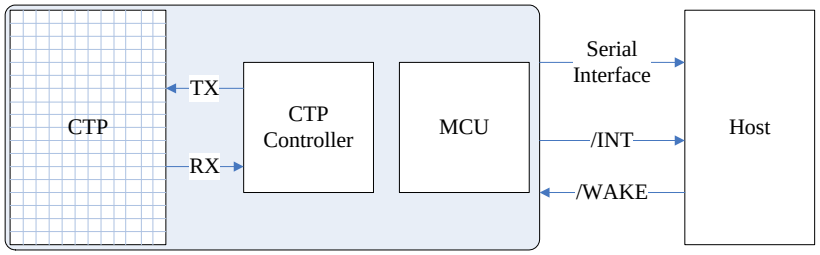


Figure 1-1 CTPM and Host connection

The Power Supply voltage of CTPM is 2.8V~3.3V, interface supply voltage is 2.8V~3.3V. There are Control Interface and Data Interface. As

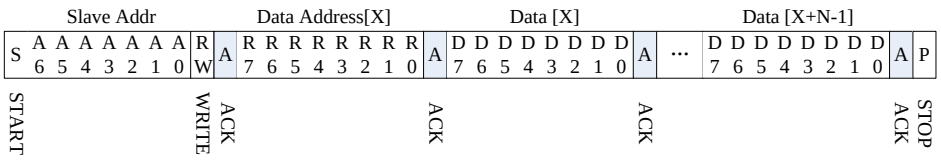
Figure 1-1 demonstrates, Serial interface is the data interface, /INT and /WAKE are the control interface. For the detail, please refer to Table 1-1.

Table 1-1 Description for TP module and Host interface

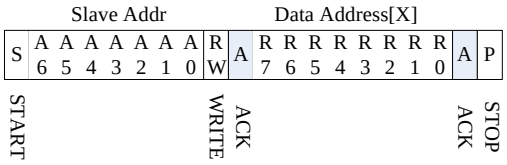
Port Name	Voltage	Polar	Description
Serial interface	2.8~3.3V		Serial interface is for data transfer between Host and CTPM. CTPM support both I2C and SPI interface
/INT	2.8~3.3V	LOW	The interrupt from the CTPM to the Host
/WAKE*	2.8~3.3V	LOW	Wakeup signal from host to the CTPM

1.2 I²C Read/Write Interface description

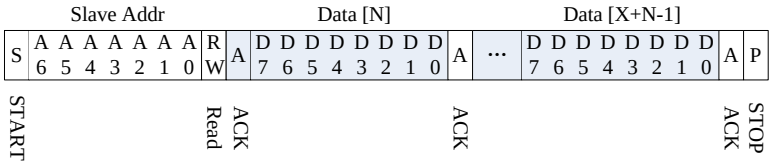
Write N bytes to I2C slave



Set Data Address



Read X bytes from I^2C Slave



1.3 Interrupt signal from CTPM to Host

As for standard CTPM, host need to use both interrupt control signal and serial data interface to get the touch data. There are two kind of method to use interrupt: interrupt trigger and interrupt query.

Here is the timing to get touch data.

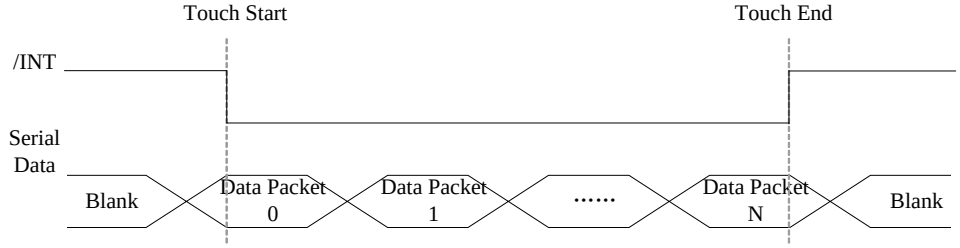


Figure 1-2 Interrupt query mode

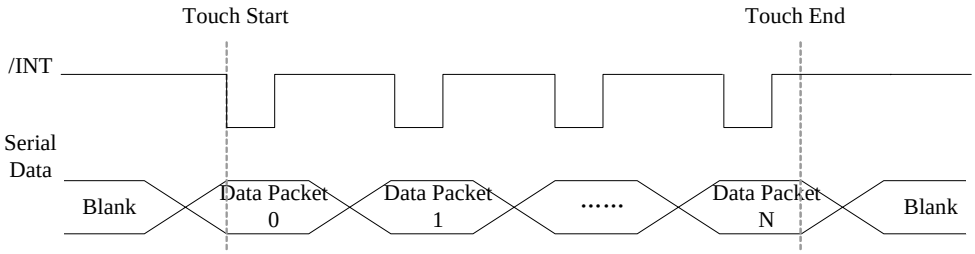


Figure 1-3 Interrupt trigger mode

Host use general I2C protocol to read the touch data or the information from CTPM . CTPM will send host a interrupt signal when there is a valid touch. Then host can use the serial data interface to get the touch data. If there is no valid touch detected, the /INT will not be pulled up, the host do not need to read the touch data.

NOTE: "valid touch" may have different definition in various systems. For example, in some systems, the valid touch is defined as there is one more valid touch point. But in some other systems, the valid touch is defined as one more valid touch with valid gestures. In usual, /INT will be pulled up when there is a valid touch point, and to be low when a touch finishes.

As for interrupt trigger mode, /INT signal will be low if there is a touch detected. But for per update of valid touch data, CTPM will produce a valid pulse for /INT signal, host can read the touch data periodically according to the frequency of this pulse. In this mode, the pulse frequency is the touch data update frequency.

.

1.4 Wakeup signal from Host to CTPM

Host can use the Wakeup Signal to wakeup the I^2C slave device.

This pin should be connected to GND when flash programming while in normal running mode it should not be connected to GND.

2 CTP Register Mapping

This chapter describes the standard FTS Capacitive Touch Panel products communication registers in address order for each device mode. The most detailed descriptions of the Standard Products communication registers are in the Register Definitions section of each chapter. The device modes are listed in the table below, along with each mode's register prefix.

Device Mode	Prefix	Val	Description
Operating	Op	000b	Read touch point and gesture
Test	Te	100b	Read raw data
System Information	Sy	001b	Read system information related Reserved

2.1 Operating Mode

In this mode the CTP is fully functional as a touch screen controller. Read and write access address is just logical address which is not enforced by hardware or firmware. Here is the operating mode register map.

Operating Mode Register Map

Address	Name	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	Host Access
Op,00h	DEVIDE_MODE		Device Mode[2:0]							RW
Op,01h	GEST_ID	Gesture ID[7:0]								R
Op,02h	TD_STATUS					Number of touch points[3:0]				R
Op,03h	TOUCH1_XH	1st Event Flag				1st Touch X Position[11:8]				R
Op,04h	TOUCH1_XL	1st Touch X Position[7:0]								R
Op,05h	TOUCH1_YH	1st Touch ID[3:0]				1st Touch Y Position[11:8]				R
Op,06h	TOUCH1_YL	1st Touch Y Position[7:0]								R
Op,07h										
Op,08h										
Op,09h	TOUCH2_XH	2nd Event				2nd Touch				R

		Flag		X Position[11:8]	
Op,0Ah	TOUCH2_XL	2nd touch X Position[7:0]			R
Op,0Bh	TOUCH2_YH	2nd Touch ID[3:0]		2nd Touch Y Position[11:8]	R
Op,0Ch	TOUCH2_YL	2nd Touch Y Position[7:0]			R
Op,0Dh					R
Op,0Eh					R
Op,0Fh	TOUCH3_XH	3rdEvent Flag		3rd Touch X Position[11:8]	R
Op,10h	TOUCH3_XL	3rd Touch X Position[7:0]			R
Op,11h	TOUCH3_YH	3rd Touch ID[3:0]		3rd Touch Y Position[11:8]	R
Op,12h	TOUCH3_YL	3rd Touch Y Position[7:0]			R
Op,13h					R
Op,14h					R
Op,15h	TOUCH4_XH	4thEvent Flag		4th Touch X Position[11:8]	R
Op,16h	TOUCH4_XL	4th Touch X Position[7:0]			R
Op,17h	TOUCH4_YH	4th Touch ID[3:0]		4th Touch Y Position[11:8]	R
Op,18h	TOUCH4_YL	4th Touch Y Position[7:0]			R
Op,19h					R
Op,1Ah					R
Op,1Bh	TOUCH5_XH	5thEvent Flag		5th Touch X Position[11:8]	R
Op,1Ch	TOUCH5_XL	5thTouch X Position[7:0]			R
Op,1Dh	TOUCH5_YH	5th Touch ID[3:0]		5th Touch Y Position[11:8]	R
Op,1Eh	TOUCH5_YL	5th Touch Y Position[7:0]			R
Op,1Fh					R
Op,20h					R
Op,21h	Reserved				
…	…				
Op,7Fh	Reserved				
Op,80h	ID_G_THGROUP	valid touching detect threshold.			R/W
Op,81h	ID_G_THPEAK	valid touching peak detect threshold.			R/W

Op,82h	ID_G_THCAL	the threshold when calculating the focus of touching.		R/W
Op,83h	ID_G_THWATER	the threshold when there is surface water.		R/W
Op,84h	ID_G_THTEMP	the threshold of temperature compensation.		R/W
Op,85h				R/W
Op,86h	ID_G_CTRL		Power control mode[1:0]	R/W
Op,87h	ID_G_TIME_ENTER _MONITOR	The timer of entering monitor status		R/W
Op,88h	ID_G_PERIODACTIVE		Period Active[3:0]	R/W
Op,89h	ID_G_PERIOD MONITOR	The timer of entering idle while in monitor status		R/W
Op,8Ah				R/W
Op,8Bh				R/W
Op,8Ch				R/W
Op,8Dh				R/W
Op,8Eh				R/W
Op,8Fh				R/W
Op,90h				R/W
Op,91h				R/W
Op,92h				R/W
Op,93h				R/W
Op,94h				R/W
Op,95h				R/W
Op,96h				R/W
Op,97h				R/W
Op,98h				R/W
Op,99h				R/W
Op,9Ah				R/W
Op,9Bh				R/W
Op,9Ch				R/W
Op,9Dh				R/W
Op,9Eh				R/W
Op,9Fh				R/W
Op,A0h	ID_G_AUTO_CLB _MODE	auto calibration mode		R/W

Op,A1h	ID_G_LIB_ VERSION_H	Firmware Library Version H byte	R
Op,A2h	ID_G_LIB _VERSION_L	Firmware Library Version L byte	R
Op,A3h	ID_G_CIPHER	Chip vendor ID	R
Op,A4h	ID_G_MODE	the interrupt status to host	R
Op,A5h	ID_G_PMODE	Power Consume Mode	
Op,A6h	ID_G_FIRMID	Firmware ID	R
Op,A7h	ID_G_STATE	Running State	
Op,A8h	ID_G_FT5201ID	CTPM Vendor ID	R
Op,A9h	ID_G_ERR	Error Code	R
Op,AAh	ID_G_CLB	Configure TP module during calibration in Test Mode	R/W
Op,ABh			R/W
Op,ACh			R/W
Op,ADh			R/W
Op,AEh	ID_G_B_AREA_TH	The threshold of big area	R/W
Op,AFh			R/W
…	…		
Op,FDh	Reserved		
Op,FEh	LOG_MSG_CNT	The log MSG count	R
Op,FFh	LOG_CUR_CHA	Current character of log message, will point to the next character when one character is read.	R

2.1.1 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
Op,00h	6:4	Device Mode [2:0]	000b Normal operating Mode 001b System Information Mode (Reserved) 100b Test Mode – read raw data (Reserved)

2.1.2 GEST_ID

This register describes the gesture of a valid touch.

Address	Bit Address	Register Name	Description
Op,01h	7:0	Gesture ID [7:0]	Gesture ID 0x10 Move UP

			0x14 Move Left 0x18 Move Down 0x1C Move Right 0x48 Zoom In 0x49 Zoom Out 0x00 No Gesture

2.1.3 TD_STATUS

This register is the Touch Data status register.

Address	Bit Address	Register Name	Description
Op,02h	3:0	Number of touch points[3:0]	How many points detected. 1-5 is valid.
	7:4		

2.1.4 TOUCHn_XH (n:1-5)

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

Address	Bit Address	Register Name	Description
Op,03h ~ Op,39h	7:6	Event Flag	00b: Put Down 01b: Put Up 10b: Contact 11b: Reserved
	5:4		Reserved
	3:0	Touch X Position [11:8]	MSB of Touch X Position in pixels

2.1.5 TOUCHn_XL (n:1-5)

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

Address	Bit Address	Register Name	Description
Op,04h ~ Op,3Ah	7:0	Touch X Position [7:0]	LSB of the Touch X Position in pixels

2.1.6 TOUCHn_YH (n:1-5)

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

Address	Bit Address	Register Name	Description
Op,05h ~ Op,3Bh	7:4	Touch ID[3:0]	Touch ID of Touch Point
	3:0	Touch X Position [11:8]	MSB of Touch Y Position in pixels

2.1.7 TOUCHn_YL (n:1-5)

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

Address	Bit Address	Register Name	Description
Op,06h ~ Op,3Ch	7:0	Touch X Position [7:0]	LSB of The Touch Y Position in pixels

2.1.8 ID_G_THGROUP

This register describes valid touching detect threshold.

Address	Bit Address	Register Name	Description
Op,80h	7:0	ID_G_THGROUP	The actual value will be 4 times of the register's value. Default:280/4

2.1.9 ID_G_THPEAK

This register describes valid touching peak detect threshold.

Address	Bit Address	Register Name	Description
Op,81h	7:0	ID_G_ THPEAK	Default:60

2.1.10 ID_G_ THCAL

This register describes threshold when calculating the focus of touching.

Address	Bit Address	Register Name	Description
Op,82h	7:0	ID_G_ THCAL	Default:16

2.1.11 ID_G_ THWATER

This register describes threshold when there is surface water.

Address	Bit Address	Register Name	Description
Op,83h	7:0	ID_G_ THWATER	Default:60

2.1.12 ID_G_ THTEMP

This register describes threshold of temperature compensation.

Address	Bit Address	Register Name	Description
Op,84h	7:0	ID_G_ THTEMP	Default:10

2.1.13 ID_G_ THDIFF

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

Address	Bit Address	Register Name	Description
Op,85h	7:0	ID_G_ THDIFF	The actual value must be 32timers of the register's value. Default :20

2.1.14 ID_G_ CTRL

This register describes the run mode of microcontroller controlled by host

Address	Bit Address	Register Name	Description
Op,86h	0	ID_G_ CTRL	0: not auto jump 1:auto jump

2.1.15 ID_G_ TIMEENTERMONITOR

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

Address	Bit Address	Register Name	Description
Op,87h	7:0	ID_G_TIME ENTERMONITOR	Default :2

2.1.16 ID_G_ PERIODACTIVE

This register describes the period of active status, it should not less than 12

Address	Bit Address	Register Name	Description
Op,88h	4:0	ID_G_ PERIOD ACTIVE	Range form 3 to 14,default 12
	7:4		

2.1.17 ID_G_ PERIODMONITOR

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

Address	Bit Address	Register Name	Description
Op,89h	7:0	ID_G_ PERIOD MONITOR	Default:40

2.1.18 ID_G_ AUTO_CLB_MODE

This register describes auto calibration mode.

Address	Bit Address	Register Name	Description
Op, A0h	7:0	ID_G_ AUTO_	8'h 00: enable auto calibration

		CLB_MODE	8'h ff: disable auto calibration

2.1.19 ID_G_ LIB_VERSION_H

This register describes library version high byte.

Address	Bit Address	Register Name	Description
Op, A1h	7:0	ID_G_LIB_VERSION_H	R: xx

2.1.20 ID_G_ LIB_VERSION_L

This register describes library version low byte.

Address	Bit Address	Register Name	Description
Op, A2h	7:0	ID_G_ LIB_VERSION_L	R: xx

2.1.21 ID_G_ CIPHER

This register describes vendor's chip id.

Address	Bit Address	Register Name	Description
OP, A3h	7:0	ID_G_ CIPHER	R: xx

2.1.22 ID_G_ MODE

This register describes the interrupt status to host.

Address	Bit Address	Register Name	Description
Op,A4h	7:0	ID_G_ MODE	0: Polling mode 1: Trigger mode

2.1.23 ID_G_ PMODE

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

Address	Bit Address	Register Name	Description
Op,A5h	7:0	ID_G_ PMODE	0: active 1: monitor 3: hibernate(deep sleep)

2.1.24 ID_G_ FIRMWARE_ID

This register describes the firmware id of the application.

Address	Bit Address	Register Name	Description
Op,A6h	7:0	ID_G_ FIRMWARE_ID	R: xx

2.1.25 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 2: calibration 3: factory 4: auto calibration

2.1.26 ID_G_ FT5201ID

This register describes vendor's chip id

Address	Bit Address	Register Name	Description
Op,A8h	7:0	ID_G_ FT5201ID	R: xx

2.1.27 ID_G_ ERR

This register describes the error code when the TPM is running.

Address	Bit Address	Register Name	Description
Op,A9h	7:0	ID_G_ ERR	ERR Code 8'h00:OK 8'h03:chip register writing inconsistent with reading 8'h05:chip start fail 8'h1A:no match among the basic input(such as TX_ORDER) while calibration

2.1.28 ID_G_ CLB

This register is used to configure the TPM when Calibration

Address	Bit Address	Register Name	Description
Op,AAh	7:0	ID_G_ CLB	Mapping the Array of G_Bank1, total length is NUM_TX+NUM_RX+1. the array address increases 1 after every write.

2.2 Test Mode

In this mode, CTP will provide some panel related information. Host can get the following information in this mode

Raw data of touch panel

Panel configure related information

Test Mode Register Map

Address	Name	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	Host Access
Te,00h	DEVIDE_MODE	Data Read Toggle	Device Mode[2:0]							RW

Te,01h	ROW_ADDR	The address of the row to be read	RW
Te,02h	START_SCAN	Start the scan command, the value stands for the scan frequency, will be set to zero when scan finishes	RW
Te,03h	ROW_NUM	Panel row number	RW
Te,04h	COL_NUM	Panel column number	RW
Te,05h	DRIVER_VOL	Driver voltage of chip	RW
Te,06h	START_RX	Setting the RX start number	RW
Te,07h	GAIN	Control the difference value for touching	RW
Te,08h	ORIGIN_XH	High byte of origin X coordinate	RW
Te,09h	ORIGIN_XL	Low byte of origin X coordinate	RW
Te,0Ah	ORIGIN_YH	High byte of origin Y coordinate	RW
Te,0Bh	ORIGIN_YL	Low byte of origin Y coordinate	RW
Te,0Ch	RES_WH	High byte of width of resolution	RW
Te,0Dh	RES_WL	Low byte of width of resolution	RW
Te,0Eh	RES_HH	High byte of height of resolution	RW
Te,0Fh	RES_HL	Low byte of height of resolution	RW
Te,10h	RAWDATA0_H	High byte of raw data 0	R
Te,11h	RAWDATA0_L	Low byte of raw data 0	R
Te,12h	RAWDATA1_H	High byte of raw data 1	R
Te,13h	RAWDATA1_L	Low byte of raw data 1	R
…	…	…	
Te,4Ah	RAWDATA29_H	High byte of raw data 29	R
Te,4Bh	RAWDATA29_L	Low byte of raw data 29	R
Te,4Ch	TH_POINT_NUM	Touch point number support	RW
Te,4Dh	Reserved		
Te,4Eh	Reserved		
Te,4Fh	Reserved		
Te,50h	TX_ORDER_0	TX Order, start from zero	RW
Te,51h	TX_ORDER_1		RW
…	…	…	RW
Te,77h	TX_ORDER_39		RW
Te,78h	ROW0_CAC	Charge Amplifier feedback Capacitance of ROW0	RW
Te,79h	ROW1_CAC	Charge Amplifier feedback Capacitance of ROW1	RW
…	…	…	
Te,9Fh	ROW39_CAC	Charge Amplifier feedback Capacitance of ROW39	RW

Te,A0h	COL0_CAC	Charge Amplifier feedback Capacitance of COL0		RW
…	…	…		
Te,BEh	COL29_CAC	Charge Amplifier feedback Capacitance of COL29		RW
Te,BFh	ROW0_1_OFFSET	Offset of ROW1	Offset of ROW0	RW
…	…	…	…	
Te,D2h	ROW38_39_OFFSET	Offset of ROW39	Offset of ROW38	RW
Te,D3h	COL0_1_OFFSET	Offset of COL1	Offset of COL0	RW
…	…	…	…	
Te,E1h	COL28_29_OFFSET	Offset of COL29	Offset of COL28	RW
…	…			
Te,FEh	LOG_MSG_CNT	The log MSG count		R
Te,FFh	LOG_CUR_CHA	Current character of log message, will point to the next character when one character is read.		R

2.2.1 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
Te,00h	7	Data Read Toggle	This bit is toggled by the Host only when a data transfer between the Host and TrueTouch device requires register based handshaking.
	6:4	Device Mode[2:0]	000b Normal operating Mode 001b System Information Mode (Reserved) 100b Test Mode – read raw data (Reserved)

2.2.2 ROW_ADDR

This register is the Touch Data status register.

Address	Bit Address	Register Name	Description
Te,01h	7:0	Row address	The address of the row to be read Please delay for more than 100us, then read the raw data

2.2.3 ROWDATAN_H

This register is the Touch Data status register.

Address	Bit Address	Register Name	Description
Te,(10+2n)h	7:0	High byte of raw data N	High byte of raw data N

			If N exceeds the column number will return 0xff

2.2.4 ROWDATAN_L

This register is the Touch Data status register.

Address	Bit Address	Register Name	Description
Te,(10+2n+1)h	7:0	Low byte of raw data N	Low byte of raw data N If N exceeds the column number will return 0xff

2.3 System information Mode

This mode provides access to all of the one-time system information. The system information is either written by the host to permanently configure the device (for example, power timers), or is written to the device at compile time for the host to read (for example, application version). To enter BIST (built in self test) mode write the BIST command required into the BIST_COMM register.

Read and write access is theoretical and is not enforce by hardware or firmware. Words have their MSB at lower address.

System Information Mode Register Map

Address	Name	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	Host Access
Sy,00h	DEVIDE_MODE	Data Read Toggle	Device Mode[2:0]							RW
Sy,01h	BIST_COMM	BIST Command[7:0]								W
Sy,02h	BIST_STAT	BIST Status[7:0]								R
Sy,03h	Unused									
Sy,04h	Unused									
Sy,05h	Unused									
Sy,06h	Unused									
Sy,07h	UID_0	Unique Silicon ID #0[7:0]								R
Sy,08h	UID_1	Unique Silicon ID #1[7:0]								R
Sy,09h	UID_2	Unique Silicon ID #2[7:0]								R
Sy,0Ah	UID_3	Unique Silicon ID #3[7:0]								R
Sy,0Bh	UID_4	Unique Silicon ID #4[7:0]								R
Sy,0Ch	UID_5	Unique Silicon ID #5[7:0]								R
Sy,0Dh	UID_6	Unique Silicon ID #6[7:0]								R
Sy,0Eh	UID_7	Unique Silicon ID #7[7:0]								R
Sy,0Fh	BL_VERH	Bootloader version[15:8]								R
Sy,10h	BL_VERL	Bootloader version[7:0]								R
Sy,11h	FTS_IC_VERH	Focal Tech IC Version[15:8]								R
Sy,12h	FTS_IC_VERL	Focal Tech IC Version[7:0]								R
Sy,13h	APP_IDH	Application ID[15:8]								R
Sy,14h	APP_IDL	Application ID[7:0]								R
Sy,15h	APP_VERH	Application Version[15:8]								R
Sy,16h	APP_VERL	Application Version[7:0]								R
Sy,17h	Unused									
Sy,18h	Unused									

Sy,19h	Unused		
Sy,1Ah	Unused		
Sy,1Bh	CID_0	Custom ID #0[0:7]	R
Sy,1Ch	CID_1	Custom ID #1[0:7]	R
Sy,1Dh	CID_2	Custom ID #2[0:7]	R
Sy,1Eh	CID_3	Custom ID #3[0:7]	R
Sy,1Fh	CID_4	Custom ID #4[0:7]	R
…	…		
Sy,FEh	LOG_MSG_CNT	The log MSG count	R
Sy,FFh	LOG_CUR_CHA	Current character of log message, will point to the next character when one character is read.	R

2.3.1 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
Sy,00h	6:4	Device Mode[2:0]	000b Normal operating Mode 001b System Information Mode (Reserved) 100b Test Mode – read raw data (Reserved)

2.3.2 BIST_COMM

This register is the BIST command register. The BIST (built in self test) function to perform is set here.

Address	Bit Address	Register Name	Description
Sy,01h	7:0	BIST Command[7:0]	BIST command to perform.

2.3.3 BIST_STAT

This register reports the status of BIST (built in self test) functions either in progress or the last function completed.

Address	Bit Address	Register Name	Description
Sy,02h	7:0	BIST Command[7:0]	Status of the last BIST function started.

2.3.4 BL_VERH

This register contains the MSB of the bootloader version specified by the application.

Address	Bit Address	Register Name	Description
Sy,0Fh	7:0	Bootloader version[15:8]	R:xx

2.3.5 BL_VERL

This register contains the LSB of the bootloader version specified by the application.

Address	Bit Address	Register Name	Description
Sy,10h	7:0	Bootloader version[7:0]	R:xx.

2.3.6 FTS_IC_VERH

This is the FTS IC version register. This register contains the MSB of the FTS IC version. The value is BCD value, for example

FT5201 – FTS_IC_VERH(0x52), FTS_IC_VERL(0x01)

FT5202 – FTS_IC_VERH(0x52), FTS_IC_VERL(0x02)

FT5206 – FTS_IC_VERH(0x52), FTS_IC_VERL(0x06)

FT5306 – FTS_IC_VERH(0x53), FTS_IC_VERL(0x06)

FT5406 – FTS_IC_VERH(0x54), FTS_IC_VERL(0x06)

Address	Bit Address	Register Name	Description
Sy,11h	7:0	Focal Tech IC version [15:8]	Focal Tech IC Version MSB

2.3.7 FTS_IC_VERL

This is the FTS IC version register. This register contains the MSB of the FTS IC version. The value is BCD value, for example

FT5201 – FTS_IC_VERH(0x52), FTS_IC_VERL(0x01)

FT5202 – FTS_IC_VERH(0x52), FTS_IC_VERL(0x02)

FT5206 – FTS_IC_VERH(0x52), FTS_IC_VERL(0x06)

FT5306 – FTS_IC_VERH(0x53), FTS_IC_VERL(0x06)

FT5406 – FTS_IC_VERH(0x54), FTS_IC_VERL(0x06)

Address	Bit Address	Register Name	Description
Sy,12h	7:0	Focal Tech IC version [7:0]	Focal Tech IC Version LSB

2.3.8 APP_IDH

This is the application ID register. This register contains the MSB of the application ID. This value is set to designate the individual project.

Address	Bit Address	Register Name	Description
Sy,13h	7:0	Application Version [15:8]	R:xx

2.3.9 APP_IDL

This is the application ID register. This register contains the MSB of the application ID. This value is set to designate the individual project.

Address	Bit Address	Register Name	Description
Sy,14h	7:0	Application Version [15:8]	R:xx

2.3.10 APP_VERH

This is the application version register. This register contains the MSB of the application version. This value should be incremented on each internal or external release of the project.

Address	Bit Address	Register Name	Description
Sy,15h	7:0	Application Version [15:8]	R:xx

2.3.11 APP_VERL

This is the application version register. This register contains the LSB of the application version. This value should be incremented on each internal or external release of the project.

Address	Bit Addr.	Reg. Name	Description
Sy,16h	7:0	Application Version [7:0]	R:xx

2.3.12 CID_n(n:0-4)

These are Custom ID registers. These regitsters contain user defined Custom ID identifiers for the FT TPM.

Address	Bit Addr.	Reg. Name	Description
Sy,1Bh~1Fh	7:0	Application Version [7:0]	R:xx

3 CTPM Application Introduction

3.1 Standard Application information of FT5X06

Figure3-1,Figure3-2,Figure3-3 demonstrate the typical FT5x06 application schematic. It consists of FT's Capacitive Touch Panel(CTP), FT5X06 chip, and some peripheral components. According to the size of CTPM, you can choose the numbers of TX and RX needed.

3.1.1 Standard application circuit of FT5206GE1

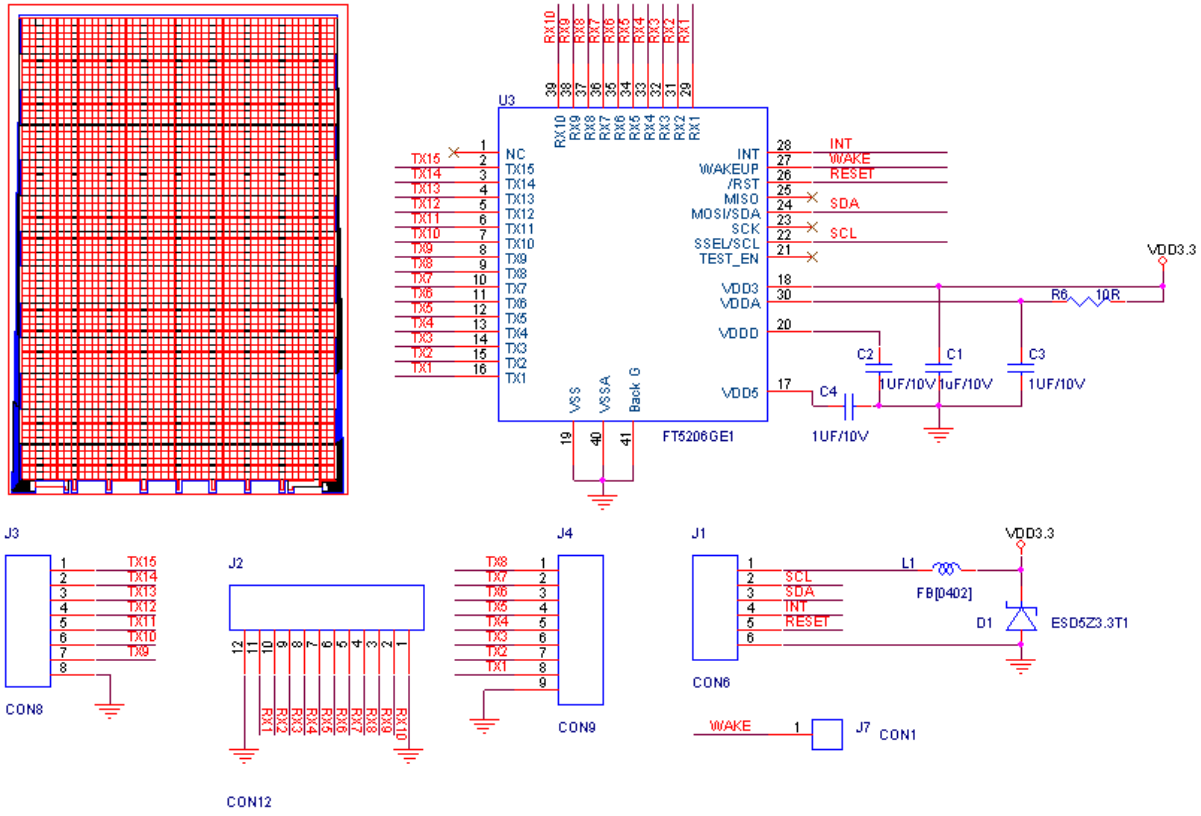


Figure 3-1 FT5206GE1 typical application schematic

3.1.2 Standard application circuit of FT5306DE4

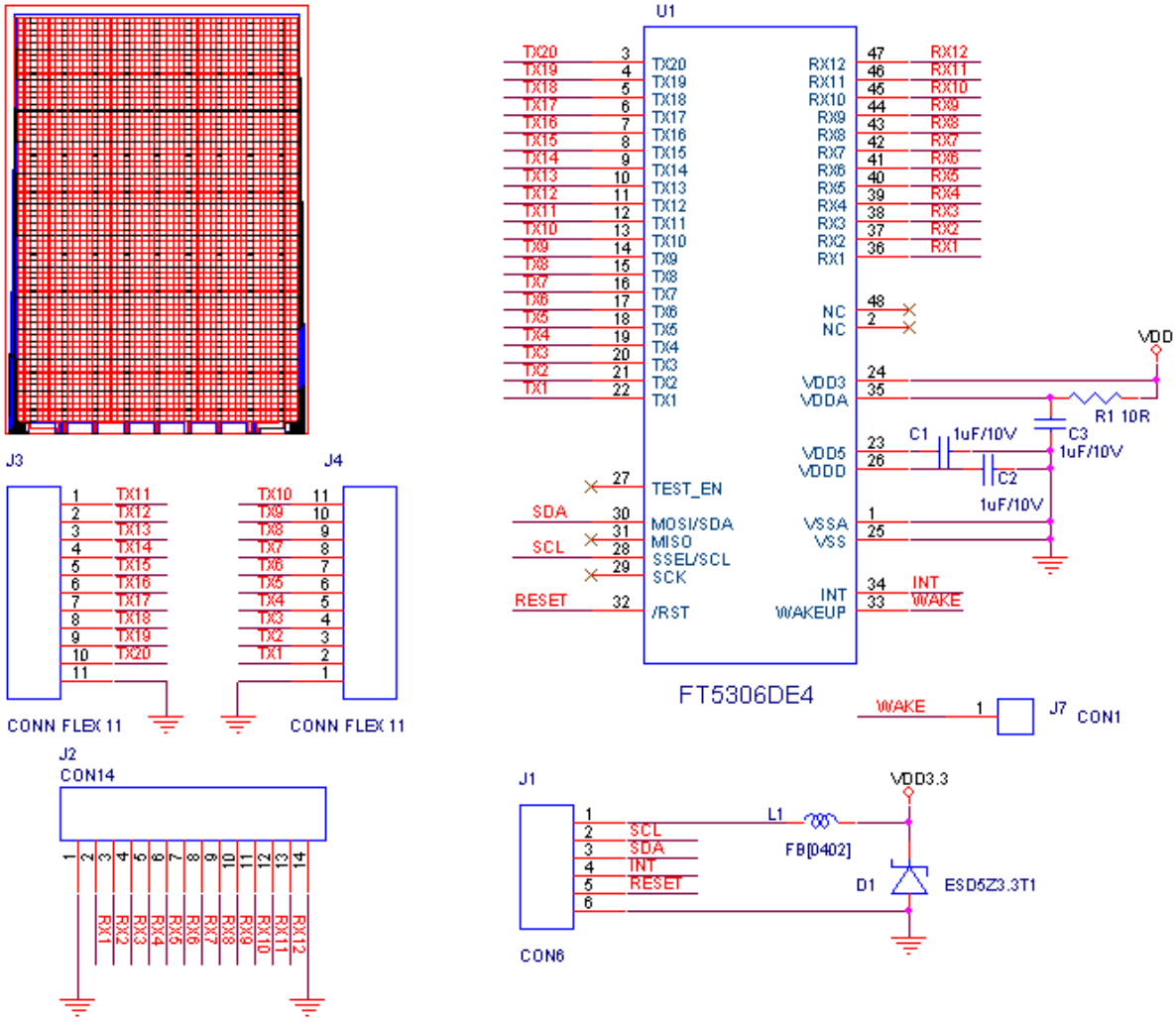


Figure 3-2 FT5306DE4 typical application schematic

3.1.3 Standard application circuit of FT5206EE8

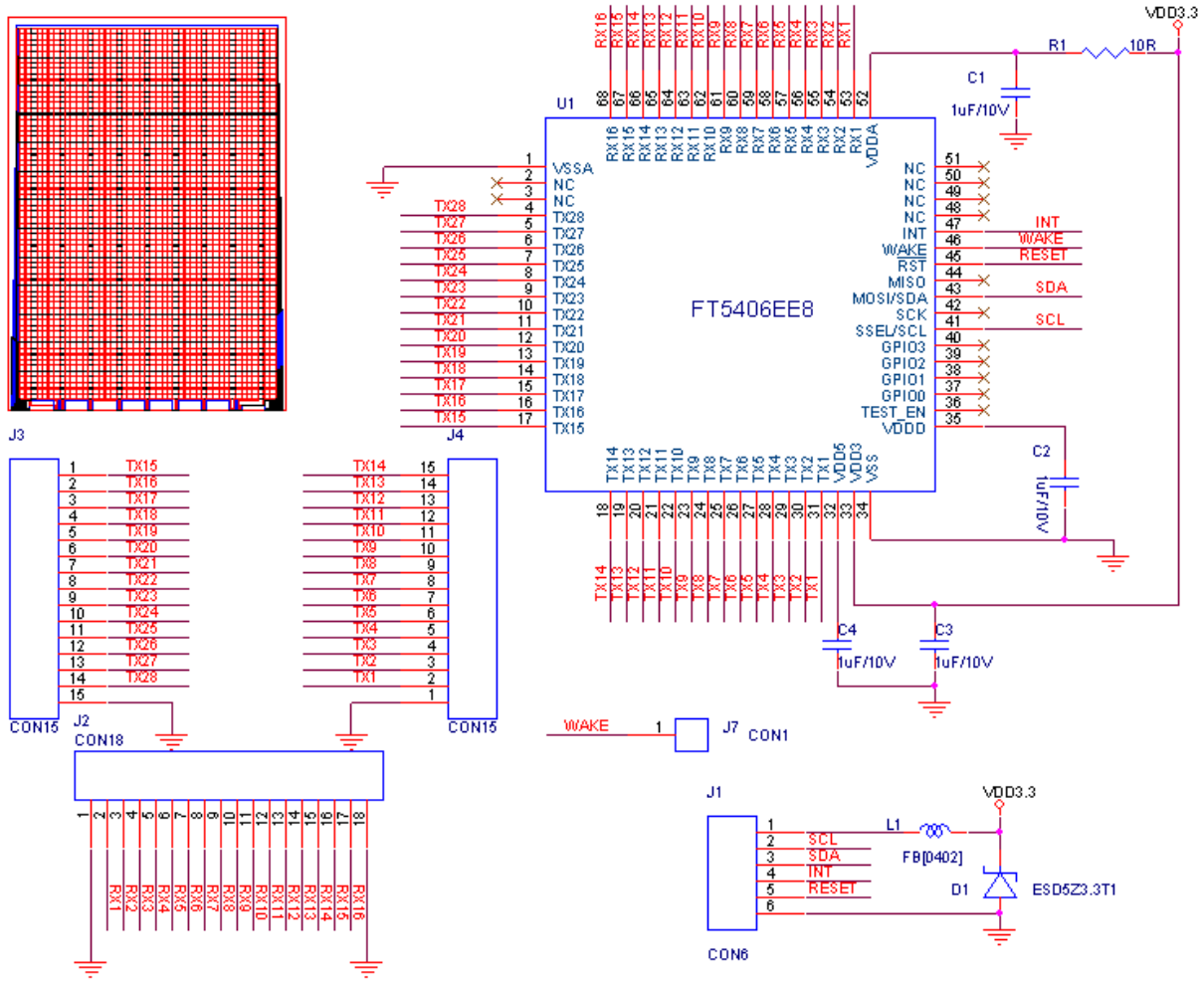


Figure 3-3 FT5406EE8 typical application schematic

4 Communication between host and CTPM

4.1 Communication Contents

The data Host received from the CTPM through serial interface are different depend on the configuration in Device Mode Register of the CTPM. Please refer to Section 2---CTP Register Mapping.

4.2 I2C Example Code

```
//////////////////////////////////////////////////////////////////
// I2C write bytes to device.
//
// Arguments: ucSlaveAdr - slave address
//                    ucSubAdr - sub address
//                    pBuf - pointer of buffer
//                    ucBufLen - length of buffer
//////////////////////////////////////////////////////////////////
void i2cBurstWriteBytes(BYTE ucSlaveAdr, BYTE ucSubAdr, BYTE *pBuf, BYTE ucBufLen)
{
    BYTE ucDummy; // loop dummy
    ucDummy = I2C_ACCESS_DUMMY_TIME;
    while(ucDummy--)
    {
        if (i2c_AccessStart(ucSlaveAdr, I2C_WRITE) == FALSE)
            continue;
        if (i2c_SendByte(ucSubAdr) == I2C_NON_ACKNOWLEDGE) // check non-acknowledge
            continue;
        while(ucBufLen--) // loop of writting data
        {
            i2c_SendByte(*pBuf); // send byte
            pBuf++; // next byte pointer
        } // while
        break;
    } // while
    i2c_Stop();
}

//////////////////////////////////////////////////////////////////
// I2C read bytes from device.
//
// Arguments: ucSlaveAdr - slave address
//                    ucSubAdr - sub address
//                    pBuf - pointer of buffer
//                    ucBufLen - length of buffer
//////////////////////////////////////////////////////////////////
void i2cBurstReadBytes(BYTE ucSlaveAdr, BYTE ucSubAdr, BYTE *pBuf, BYTE ucBufLen)
{
    BYTE ucDummy; // loop dummy

    ucDummy = I2C_ACCESS_DUMMY_TIME;
    while(ucDummy--)
```

```
    {
        if (i2c_AccessStart(ucSlaveAdr, I2C_WRITE) == FALSE)
            continue;
        if (i2c_SendByte(ucSubAdr) == I2C_NON_ACKNOWLEDGE) // check non-acknowledge
            continue;
        if (i2c_AccessStart(ucSlaveAdr, I2C_READ) == FALSE)
            continue;
        while(ucBufLen--) // loop to burst read
        {
            *pBuf = i2c_ReceiveByte(ucBufLen); // receive byte
            pBuf++; // next byte pointer
        } // while
        break;
    } // while
    i2c_Stop();
}

//////////////////////////////////////////////////////////////////
// I2C read current bytes from device.
//
// Arguments: ucSlaveAdr - slave address
//                  pBuf - pointer of buffer
//                  ucBufLen - length of buffer
//////////////////////////////////////////////////////////////////
void i2cBurstCurrentBytes(BYTE ucSlaveAdr, BYTE *pBuf, BYTE ucBufLen)
{
    BYTE ucDummy; // loop dummy

    ucDummy = I2C_ACCESS_DUMMY_TIME;
    while(ucDummy--)
    {
        if (i2c_AccessStart(ucSlaveAdr, I2C_READ) == FALSE)
            continue;
        while(ucBufLen--) // loop to burst read
        {
            *pBuf = i2c_ReceiveByte(ucBufLen); // receive byte
            pBuf++; // next byte pointer
        } // while
        break;
    } // while
    i2c_Stop();
}
```