Sitronix Touch IC

ST7123 Touch Screen Controller

Note: Sitronix Technology Corp. reserves the right to change the contents in this document without prior notice. This is not a final specification. Some parameters are subject to change.

Interface Protocol A

Version 01.11

2023/12/25

Sitronix Technology Corporation

1. CONTENT

2. REVISION HISTORY

Version	Date	Description
01.10	2023/03/06	1. First Release
01.11	2023/12/25	1. Add coordinate checksum information in Misc. Info 2. Add Reporting Differ table

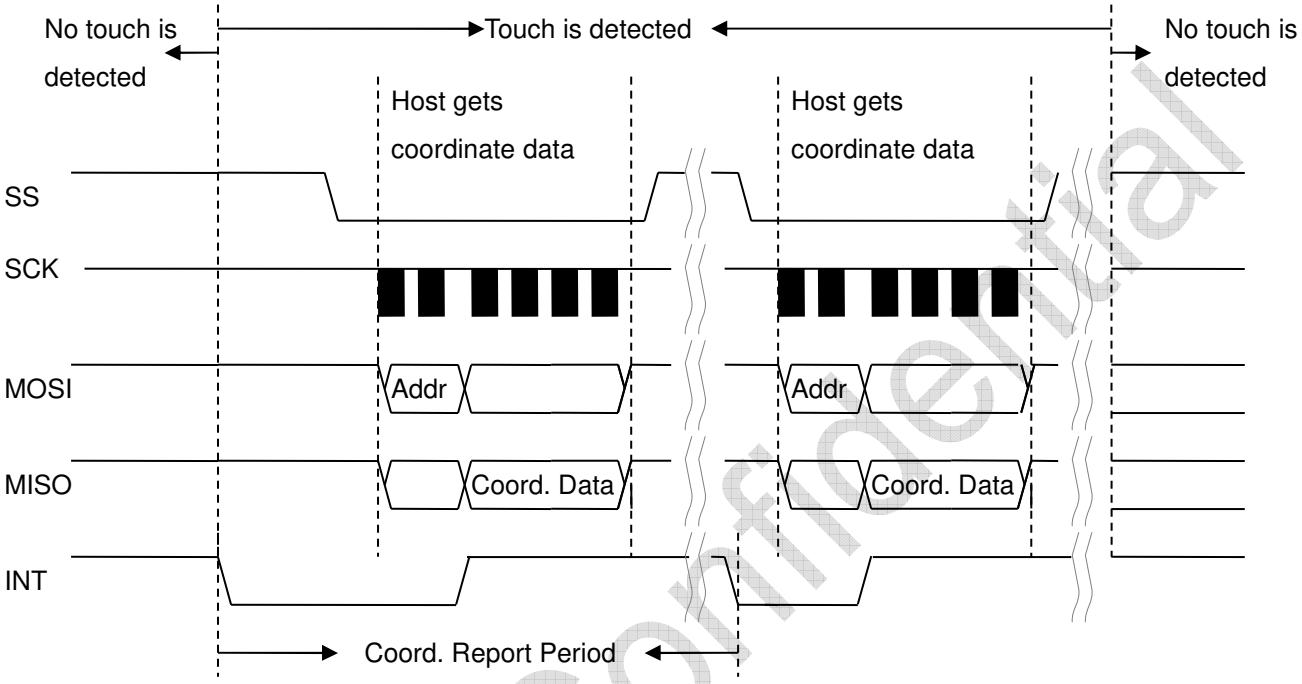
3. INTRODUCTION

Sitronix Touch IC is a low-cost, high performance single chip solution for capacitive touch screen. For application, Sitronix Touch IC supports raw data, coordinates and device control information for host side application. For communication interface, Sitronix Touch IC supports register mapped interface protocol for host device to retrieve information through Sitronix Touch IC host interfaces. Developer can get information about raw data, coordinates or device control and develop their system very easily through the register interface protocol.

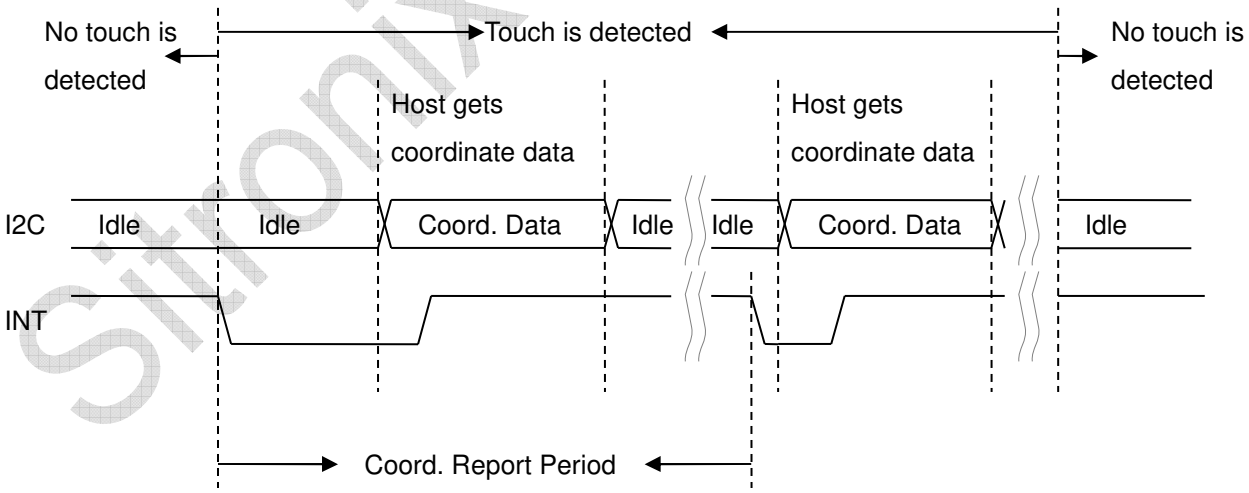
The capacitive touch sensor is covered with a plastic or glass cover lens. It provides various parameters for a wide range of capacitance on the touch sensor. The touch sensor controller converts touch sensor data into X and Y coordinates for each finger according to the motions of fingers detected by controller.

4. WAVEFORM

4.1. SPI Electrical Waveform



4.2. I2C Electrical Wave Form

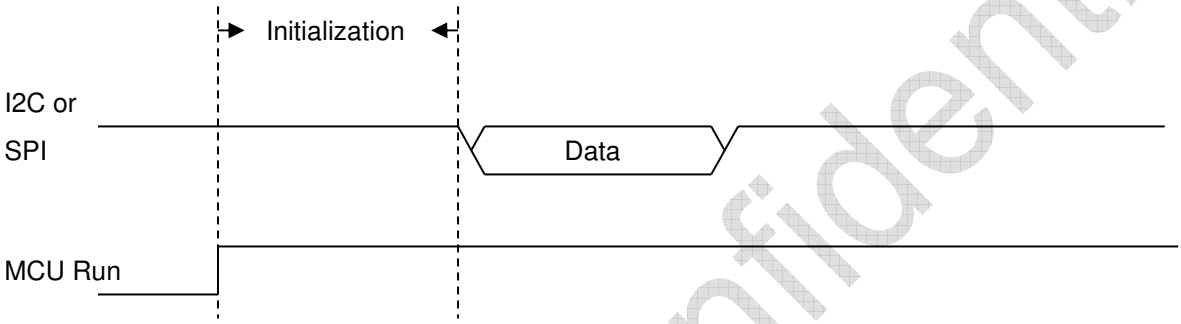


5. HOST INTERFACE PROTOCOL

5.1. Initialization

After hardware reset, touch controller needs some time for initialization. The touch controller can be accessed via I2C or SPI interface after initialization.

Touch IC	Initialization Time
ST7123	20 ms



5.2. SPI Protocol

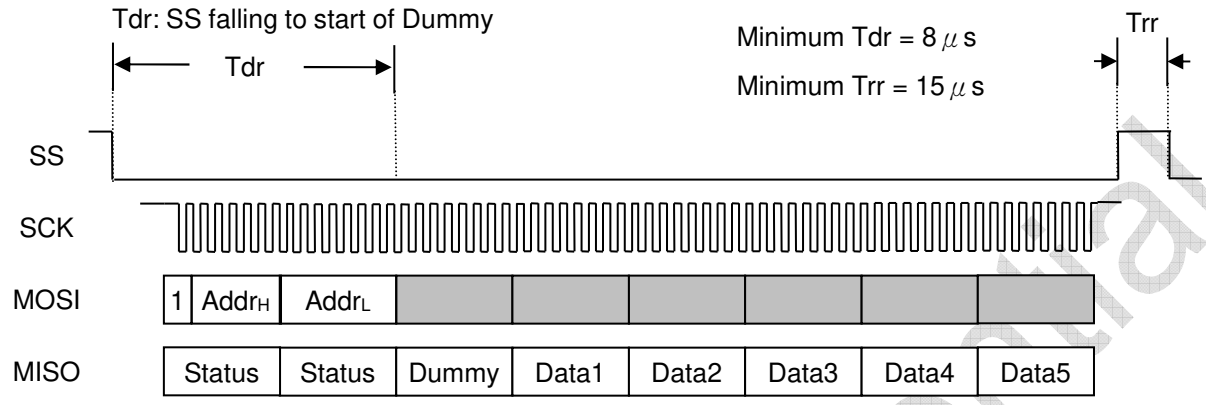
To read/write register data through SPI interface, the *Register Address* has to be transmitted on MOSI first. The *Register Address* in SPI protocol is two bytes wide, with MSB (bit 15) being '1' for SPI read transaction, and '0' for SPI write transaction.

For each read/write transaction, host can receive the *SPI Status* from device on MISO. The *SPI Status* indicates that the SPI transaction is failed or not.

SPI Status	Description
0x00	Device is normal. No error on SPI transaction.
0x80	Device is busy. SPI transaction failed.
Othes	Reserved

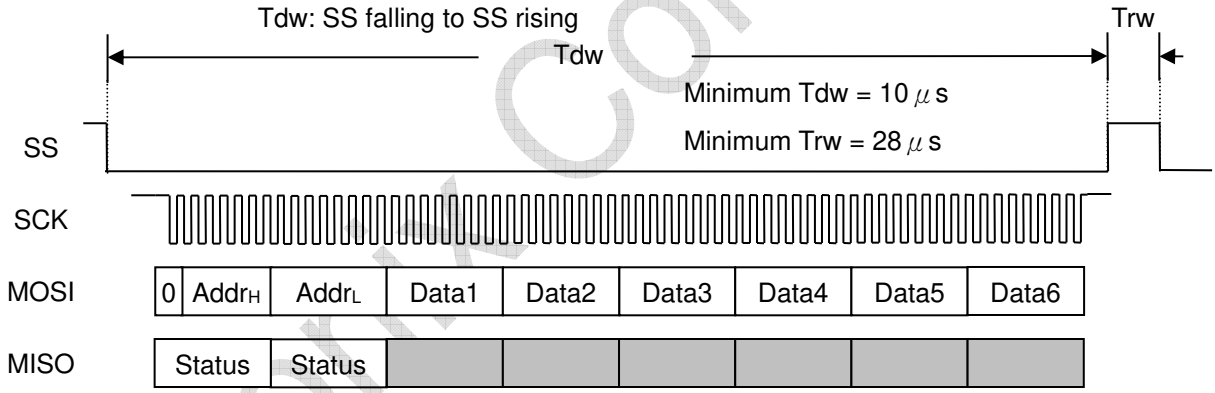
5.2.1. Register Read

The following figure presents a typical SPI read transaction. **The length of Dummy is 1 bytes.**

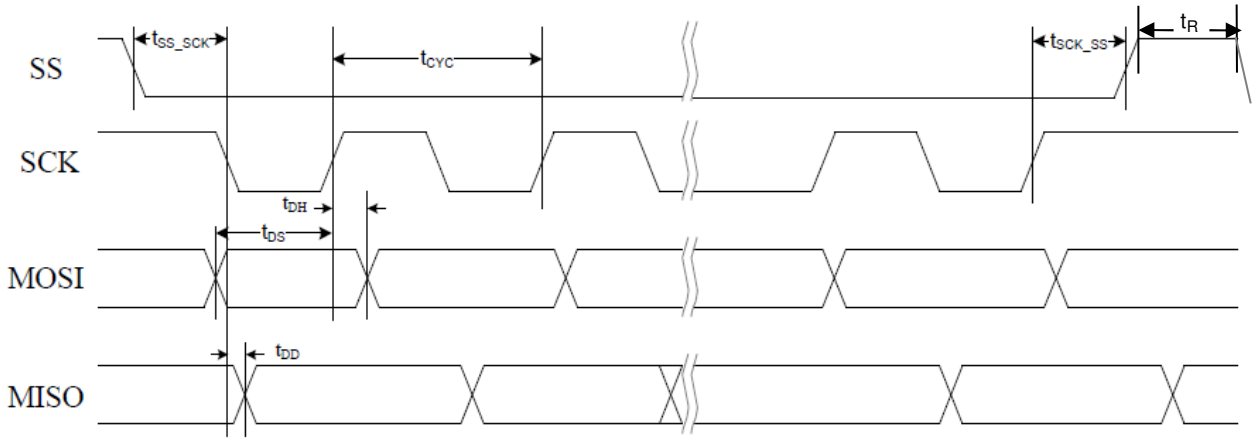


5.2.2. Register Write

The following figure presents a typical SPI write transaction. **The Max length of writing in one packet is 32 bytes.**



5.2.3. SPI Timing



Conditions: IOVDD = 1.65V~1.95V, GND = 0V, T_A = -20°C~85°C

Symbol	Parameter	Rating			Unit
		Min.	Typ.	Max.	
f_{SCK}	SCK frequency	-	-	12	Mhz
t_{CYC}	SCK cycle time	125	-	-	ns
t_{DS}	Data setup time prior SCK rising	25	-	-	ns
t_{DH}	Data hold time after SCK rising	25	-	-	ns
t_{DD}	MISO data output delay from SCK falling	-	-	50	ns
t_{SS_SCK}	SS falling to 1st SCK falling	100	-	-	ns
t_{SCK_SS}	SCK rising to SS rising	100	-	-	ns
t_R	CS recovery time	28	-	-	us

5.3. I2C Protocol

5.3.1. Default I2C Address

Default of I2C address of Sitronix Touch IC is **0x55** (7-bits address). If the I2C address conflicts with another I2C device's address on the same bus, user can change I2C address by Sitronix PC Utility.

5.3.2. Register Read

For reading register value from I2C device, host has to tell I2C device the *Start Register Address* before reading corresponding register value.

I2C Start	I2C Header (W)	Start Reg. $Addr_H$. (a)	Start Reg. $Addr_L$. (a)	I2C Stop	I2C Start	I2C Header (R)	Value of Reg(a)	Value of Reg(a+1)	…	Value of Reg(a+n)	I2C Stop

Sitronix Touch IC I2C host interface protocol supports *Repeated Register Read*. That is, once the *Start Register Address* has been set by host, consequent I2C Read(R) transactions will directly read register values starting from the *Start Register Address* without setting address first, as shown below.

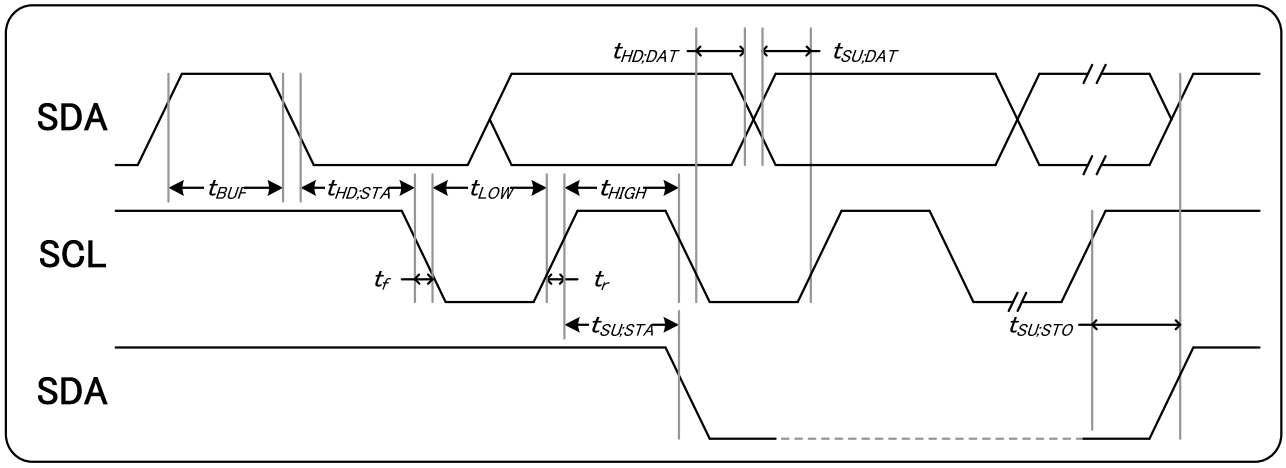
I2C Start	I2C Header (R)	Value of Reg(a)	Value of Reg(a+1)	…	Value of Reg(a+n)	I2C Stop	I2C Start	I2C Header (R)	Value of Reg(a)	Value of Reg(a+1)	…	Value of Reg(a+n)	I2C Stop

5.3.3. Register Write

For writing register to I2C device, host has to tell I2C device the Start Register Address in each I2C Register Write transaction. Register values to the I2C device will be written to the address starting from the Start Register Address described in Register Write I2C transaction as shown below.

I2C Start	I2C Header (W)	Start Reg. $Addr_H$.(a)	Start Reg. $Addr_L$.(a)	Value to Reg(a)	Value to Reg(a+1)	…	Value to Reg(a+n)	I2C Stop

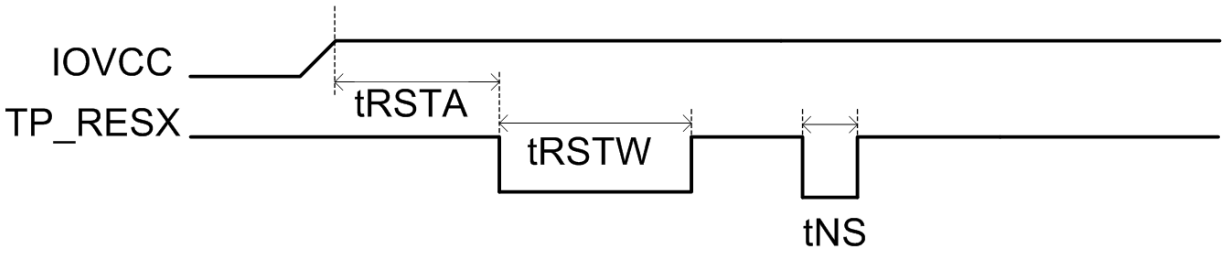
5.3.4. I2C Timing



Conditions: IOVDD = 1.65V~1.95V, GND = 0V, T_A = -20°C~85°C

Item	Signal	Symbol	Rating			Unit
			Min.	Typ.	Max.	
SCL clock frequency	SCL	fSCL			400	khz
SCL clock low period		tLOW	1250			ns
SCL clock high period		tHIGH	1250			ns
Data set-up time	SDA	tSU;DAT	100			ns
Data hold time		tHD;DAT	0			ns
Setup time for a repeated START condition	SDA	tSU;STA	600			ns
Start condition hold time		tHD;STA	600			ns
Setup time for STOP condition		tSU;STO	600			ns
Bus free time between a STOP and START		tBUF	1300			ns

5.4. Touch Reset Timing



Conditions: IOVDD = 1.65V~1.95V, GND = 0V, T_A = -20°C~85°C

Symbol	Parameter	Rating			Unit
		Min.	Typ.	Max.	
tNS	Pulse cancellation time	-	-	10	us
tRSTW	TP_RESX Width	2	-	-	ms
tRSTA	IOVCC on to TP_RESX	10	-	-	ms

6. REPORT PAGE REGISTERS

Sitronix Touch IC provides a register set for host to configure device attributes and retrieve information about fingers and raw data through device host interface. Host interface registers are listed below.

Host Interface Registers (Report Page)									
Reg. Addr.	**Name**	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
0x0000	Firmware Version	Version (RO)							
0x0001	Status	Error Code (RO)				Device Status (RO)			
0x0002	Device Control	*Reserved*		Proximity Enable (RW)	*Reserved*			Power Down (RW)	Reset (RW)
0x0003 0x0004	…	*Reserved*							
0x0005	Max X Coord(H)	*Reserved*		X Coordinate resolution high byte (RO)					
0x0006	Max X Coord(L)	X Coordinate resolution low byte (RO)							
0x0007	Max Y Coord(H)	*Reserved*		Y Coordinate resolution high byte (RO)					
0x0008	Max Y Coord(L)	Y Coordinate resolution low byte (RO)							
0x0009	Max Touches	Max Number of Touches (RO)							
0x000A	Sensing Counter (High Byte)	Sensing_Counter_H (RO)							
0x000B	Sensing Counter (Low Byte)	Sensing_Counter_L (RO)							
0x000C	Firmware Revision 3	FW_Rev_3 (RO)							
0x000D	Firmware Revision 2	FW_Rev_2 (RO)							
0x000E	Firmware Revision 1	FW_Rev_1 (RO)							
0x000F	Firmware Revision 0	FW_Rev_0 (RO)							
0x0010	Advanced Touch Info.(RO)	RstChip	Proximity Status			With Coord.	With prox. raw.	*Reserved*	*Reserved*
0x0011	…	*Reserved*							
0x0012	Gesture Info.(RO)	Gestures							
0x0013	Keys	*Reserved*		Key5	Key4	Key3	Key2	Key1	Key0

Host Interface Registers (Report Page)									
Reg. Addr.	**Name**	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
0x0014	X0 Coord. (High Byte)	Valid0	*Reserved*	X0_H (RO)					
0x0015	X0 Coord. (Low Byte)	X0_L (RO)							
0x0016	Y0 Coord. (High Byte)	*Reserved*		Y0_H (RO)					
0x0017	Y0 Coord. (Low Byte)	Y0_L (RO)							
0x0018	Touch Area 0	Touch Area 0 (RO)							
0x0019	…	Touch Intensity 0 (RO)							
0x001A	…	*Reserved*							
0x001B	X1 Coord. (High Byte)	Valid1	*Reserved*	X1_H (RO)					
0x001C	X1 Coord. (Low Byte)	X1_L (RO)							
0x001D	Y1 Coord. (High Byte)	*Reserved*		Y1_H (RO)					
0x001E	Y1 Coord. (Low Byte)	Y1_L (RO)							
0x001F	Touch Area 1	Touch Area 1 (RO)							
0x0020	…	Touch Intensity 1 (RO)							
0x0021	…	*Reserved*							
0x0022 … 0x0052	…	…							
0x0053	X9 Coord. (High Byte)	Valid9	*Reserved*	X9_H (RO)					
0x0054	X9 Coord. (Low Byte)	X9_L (RO)							
0x0055	Y9 Coord. (High Byte)	*Reserved*		Y9_H (RO)					
0x0056	Y9 Coord. (Low Byte)	Y9_L (RO)							
0x0057	Touch Area 9	Touch Area 9 (RO)							
0x0058	…	Touch Intensity 9 (RO)							
0x0059	…	*Reserved*							
0x005A	Prox.Raw Header	*Reserved*							Prox. Raw mode

Host Interface Registers (Report Page)									
Reg. Addr.	**Name**	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
0x005B … 0x00EA	Prox.Raw	Proximity rawdata							
CkAddr	checksum	Coordinate checksum							
0x00EC … 0x00EF	…	*Reserved*							
0x00F0	Misc. Info. (RO)	Smart Wake Up Flag	*Reserved*	*SupportProximity.*	*SupportCoordCheckSum*	*Reserved*	*Reserved*	*Reserved*	
0x00F1	Misc. Control	Enable Smart Wake Up (RW)	*Reserved*	*Reserved*	*Reserved*			*Reserved*	
0x0130	ExDiffEn	*Reserved*							DiffEn
0x0131	ExDiffPackNum	Report differ data package number(RO)							
0x0132	ExDiffBuf0 (RO)	Map offset diff buf0[15:8]							
0x0133		Map offset diff buf0[7:0]							

Differ data table									
		Bit7	**Bit6**	**Bit5**	**Bit4**	**Bit3**	**Bit2**	**Bit1**	**Bit0**
Offst+0x0000	IDInf0	Tx[1:0]		Rx[5:0]					
Offst+0x0001	IDInf1	ID[3:0]				Tx[5:2]			
Offst+0x0002	Node(Tx_N,Rx_M)	Differ(High byte)							
Offst+0x0003	Node(Tx_N,Rx_M)	Differ(Low byte)							
Offst+0x0004	Node(Tx_{N+1},Rx_M)	Differ(High byte)							
Offst+0x0005	Node(Tx_{N+1},Rx_M)	Differ(Low byte)							
…	…	…							
Offst+0x0060	Node(Tx_{N+5},Rx_{M+6})	Differ(High byte)							
Offst+0x0061	Node(Tx_{N+5},Rx_{M+6})	Differ(Low byte)							
Offst+0x0062	Node(Tx_{N+6},Rx_{M+6})	Differ(High byte)							
Offst+0x0063	Node(Tx_{N+6},Rx_{M+6})	Differ(Low byte)							

6.1. Firmware Version Register

Reg. Addr.	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x0000	Firmware Version	Version (RO)							

Firmware Version Register provides version information about current firmware. Host application can support version control in firmware upgrade function by reading *Firmware Version Register* and comparing with the version of new firmware binary.

6.2. Status Register

Reg. Addr.	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x0001	Status Reg.	Error Code (RO)				Device Status (RO)			

Status Register shows current status of the device to host, including *Device Status* and *Error Code*. *Init* status represents that the device is in *Init* state and not ready for host access. Host has to wait for the device to change into *Normal* state before accessing registers other than *Status Register*. If *Device Status* shows *Error*, the *Error Code* field in the *Status Register* shows the reason of error.

Device Status	
0x0	Normal
0x1	Init
0x2	Error
0x3	SmartWakeup
0x4	Idle
0x5	Power Down
0x6	Pre-SmartWakeup
0x7	Proximity
0x8	TRGT Proximity
0x9 … 0xF	*Reserved*

Error Code	
0x0	No Error
0x1	Invalid Address
0x2	Invalid Value
0x3	Invalid Platform
0x4	Dev Not Found
0x5	Stack Overflow
0x6	Invalid Firmware Parameter Table

Error Code	
0x7 … 0xF	*Reserved*

6.3. Device Control Register

Reg. Addr.	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x0002	Device Control Reg.	*Reserved*		*Proximity Enable (RW)*	*Reserved*			Power Down (RW)	Reset (RW)

Device Control Register provides device control bits for host to reset the device or power down the device.

For flash boot device, host can set Reset bit to reset the device. It will do nothing if host clears this bit.

For host boot device, Reset bit is reserved.

When host sets Power Down bit, touch sensor controller will enter power down mode. Host can clear Power Down bit to wake up the controller.

Host sets “Proximity Enable” bit to 1 to enable proximity function and clear it to disable. The proximity information is shown in “Proximity Flag” of “Advanced Touch Information” register.

Please always write 0 into reserved bits.

6.4. Sensing Counter Registers

Reg. Addr.	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x000A	Sensing Counter (High Byte)	Sensing_Counter_H (RO)							
0x000B	Sensing Counter (Low Byte)	Sensing_Counter_L (RO)							

Sensing Counter Registers provide a frame-based scan counter for host to verify current scan rate. This counter will be increased by one each time when a frame data is produced by the controller scanning system.

6.5. Firmware Revision Registers

Reg. Addr.	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x000C	Firmware Revision 3	FW_Rev_3 (RO)							
0x000D	Firmware Revision 2	FW_Rev_2 (RO)							
0x000E	Firmware Revision 1	FW_Rev_1 (RO)							
0x000F	Firmware Revision 0	FW_Rev_0 (RO)							

Firmware Revision Registers provide revision information about current firmware.

6.6. Reporting table

Reg. Addr.	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x0010	Advanced Touch Info.(RO)	RstChip	Proximity Status			With Coord.	With prox. raw.	*Reserved*	*Reserved*
0x0011	…	*Reserved*							
0x0012	Gesture Info.(RO)	Gestures							
0x0013	Keys	*Reserved*		Key5	Key4	Key3	Key2	Key1	Key0
0x0014	X0 Coord. (High Byte)	Valid0	*Reserved*	X0_H (RO)					
0x0015	X0 Coord. (Low Byte)	X0_L (RO)							
0x0016	Y0 Coord. (High Byte)	*Reserved*		Y0_H (RO)					
0x0017	Y0 Coord. (Low Byte)	Y0_L (RO)							
0x0018	Touch Area 0	Touch Area 0 (RO)							
0x0019	…	Touch Intensity 0 (RO)							
0x001A	…	*Reserved*							
0x001B … 0x0059	…	…							
0x005A	Prox.Raw Header	*Reserved*							Prox. Raw mode

Reg. Addr.	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x005B … 0x00EA	Prox.Raw	Proximity rawdata							
CkAddr	checksum	Coordinate checksum							

The touch controller will output INT pin to notify host that data in reporting table is updated.

If host read any Reg addr. from 0x0010 to Reg addr. 0x0013, and if the bit " **With Coord.**" and "**With prox. raw.**" is 0, the touch controller will clear INT pin and update next data if they're ready.

If the bit " **With Coord.**" is 1, host must read the last Reg. addr. of the last supported coordinate and the touch controller will clear INT pin. For example, if the max touch number is 5, host read Reg addr. from 0x0010 to 0x0036 or host read Reg addr. from 0x0030 to 0x0036 will notify touch controller that coordinates are read. INT pin will be cleared if the bit " **With prox. raw.**" is 0.

"**Prox.Raw**" shows the proximity rawdata of Y-direction and X-direction. The word data number of proximity rawdata of Y-direction is Y-axis channels on active area. The word data number of proximity rawdata of X-direction is X-axis channels on active area. If "**Prox. Raw mode**" is 0, "**Prox.Raw**" shows the proximity rawdata of Y-direction only. If "**Prox. Raw mode**" is 1, "**Prox.Raw**" shows the proximity rawdata of Y-direction then the rawdata of X-direction.

If the bit " **With prox. raw.**" Is 1, host must read the last Reg. addr. of "Prox. Raw" and the touch controller will be notified that prox. rawdata are read. For example of 32 RXs, host read Reg addr. from 0x005A to 0x009A or host read Reg. addr. from 0x0010 to 0x009A will both notify touch controller that prox. raw are read. INT pin will be cleared if the bit " **With Coord.**" is 0. If the bit " **With Coord.**" is 1, INT pin will be cleared after host read all coordinates and prox. rawdata.

Advanced Touch Information field provides some advanced touch information for host like the followings.

a. RstChip:

It notifies host that the touch controller should be reset.

b. Proximity Status:

If proximity status changed, the touch controller will output INT pin to notify host.

c. With Coord.: If this bit is 1, it means coordinates are updated in this INT signal. If host read the last Reg. addr of the last supported coordinate, the touch controller will clear this bit. In SWK mode, this bit is not used.

d. With prox. raw.: If this bit is 1, it means proximity rawdata are updated in this INT signal. If host read the last Reg. addr. of "Prox. Raw", the touch controller will clear this bit.

e. WithChecksum: If this bit is 1, the checksum Register is valid. Otherwise, there is no checksum of coordinates.

f. Gestures:

It defines following gestures:

Gesture Type = 0: No gesture.

Gesture Type = 1: Reserved.

Gesture Type = 2: Zoom in.

Gesture Type = 3: Zoom out.

Gesture Type = 4: Left to right slide. (→)

Gesture Type = 5: Right to left slide. (←)

Gesture Type = 6: Top to down slide. (↓)

Gesture Type = 7: Down to top slide. (↑)

Gesture Type = 8: Double taps.

Gesture Type = 9: Single tap.

Gesture Type = 0x0A: Long press.

In smart wake up mode, once the specified handwriting gesture is detected, touch controller wakes host up via “INT” pin. The identification of handwriting gesture will be put into “Gestures Register” and all handwriting coordinates during handwriting will also be output to “XY Coordinate Register”. So host may read “Gestures Register” to “XY Coordinate Register” several times to get all handwriting coordinates. Note that read from Reg addr. 0x0010 to “XY Coordinate Register” to get all handwriting coordinate is suggested so that host can monitor “RstChip register”. The bit “**With Coord.**” will not be set to 1 even the handwriting coordinates are output.

The touch controller notify host to receive each touch coordinate via “INT” pin. After host gets all handwriting coordinates, touch controller clears the value in “Gesture Register” as 0.

Host must get all handwriting coordinate until the value in “Gesture Register” is 0.

Gesture ID:

ID = 0: No any handwriting gesture is detected.

ID = 0xFF: Handwriting gesture detection is failure.

ID = 0xB0: Double taps.

ID = 0xB1: Single tap

ID = 0xB2: Long press

ID = 0xC0: Left to right slide (→).

ID = 0xC1: Right to left slide (←).

ID = 0xC2: Top to down slide (↓).

ID = 0xC3: Down to top slide (↑).

ID = 0xC4: Arrow to top (^)

ID = 0xC5: Arrow to right (>)

ID = 0xC6: Arrow to bottom (ˇ)

ID = 0xC7: Arrow to left (<)

ID = 0xC8: Two fingers top to down slide (↓ ↓)

All character identifications are defined following ASCII code.

ID = 0x63: c.

ID = 0x65: e.

ID = 0x6D: m.

ID = 0x6F: o.

ID = 0x73: s.

ID = 0x76: v.

ID = 0x77: w.

ID = 0x7A: z.

Others are reserved.

g. Key:

Key field represents which key is pressed or released. Each bit in the *Key* field represents the pressed or released state of one key. If the bit is set, it means that the corresponding key is pressed. Otherwise, the key is released.

h. *XY Coordinate Registers:*

The *XY Coordinate Registers* represents the XY coordinates for each touch point ID. Valid bit field tells that this point ID is valid and the XY information represents a real touch point on touch sensor.

Touch Area Registers provide the size of touch area.

6.7. XY Coordinate resolution, Maximum Number of Touches Register

Reg. Addr.	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x0005	Max X Coord(H)	*Reserved*		X Coordinate resolution high byte (RO)					
0x0006	Max X Coord(L)	X Coordinate resolution low byte (RO)							
0x0007	Max Y Coord(H)	*Reserved*		Y Coordinate resolution high byte (RO)					
0x0008	Max Y Coord(L)	Y Coordinate resolution low byte (RO)							
0x0009	Max Touches	Max Number of Touches (RO)							

XY Coordinate resolution Register provides the maximum value of XY coordinates. They are read-only registers.

Max Touches Register provides information about the maximum number of touches that touch controller can detect. It's a read-only register.

6.8. Miscellaneous Information Register

Reg. Addr.	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x00F0	Misc. Info. *(RO)*	Smart Wake Up Flag	*Reserved.*	*SupportProximity*	*SupportCoordCheckSum*	*Reserved.*	*Reserved.*	*Reserved.*	

Miscellaneous Information Register provides some misc. information to host.

The "Smart Wake Up" function is an option for customer. The Smart Wake Up Flag shows whether the current touch firmware support smart wake up function or not.

Smart Wake Up Flag = 0: Current touch firmware does not support smart wake up function.

Smart Wake Up Flag = 1: Current touch firmware supports smart wake up function.

SupportProximity bit shows whether the touch controller supports proximity sensing or not.

SupportCoordCheckSum bit shows whether the touch controller supports coordinate checksum or not.

6.9. Miscellaneous Control Register

Reg. Addr.	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x00F1	Misc. Control	Enable Smart Wake Up (RW)	*Reserved*	*Reserved*	*Reserved*			*Reserved*	

Miscellaneous Control Register provides miscellaneous control bits for some special functions.

Host can set/clear “Enable Smart Wake Up” bit to enable/disable “Smart Wake Up” function.

To enable smart wake up function, the “Enable Smart Wake Up” bit should be set before power down the touch controller. The touch controller will be in “Finger” mode after power down. In this mode, touch driver is still sensing the touch panel but as saving power as possible.

6.10. Reporting Differ table

Reg. Addr.	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x0130	ExDiffEn	*Reserved*							DiffEn
0x0131	ExDiffPackNum	Report differ data package number(RO)							
0x0132	ExDiffBuf0 (RO)	Map offset diff buf0[15:8]							
0x0133		Map offset diff buf0[7:0]							

Differ data table									
Reg. Addr		Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Offst+0x0000	IDInf0	Tx[1:0]		Rx[5:0]					
Offst+0x0001	IDInf1	ID[3:0]				Tx[5:2]			
Offst+0x0002	Node(Tx_N,Rx_M)	Differ(High byte)							
Offst+0x0003	Node(Tx_N,Rx_M)	Differ(Low byte)							
Offst+0x0004	Node(Tx_{N+1},Rx_M)	Differ(High byte)							
Offst+0x0005	Node(Tx_{N+1},Rx_M)	Differ(Low byte)							
…	…	…							
Offst+0x0060	Node(Tx_{N+5},Rx_{M+6})	Differ(High byte)							
Offst+0x0061	Node(Tx_{N+5},Rx_{M+6})	Differ(Low byte)							
Offst+0x0062	Node(Tx_{N+6},Rx_{M+6})	Differ(High byte)							
Offst+0x0063	Node(Tx_{N+6},Rx_{M+6})	Differ(Low byte)							
Offst+0x0064	DiffChecksum	Checksum of differ data							

The Reg. Addr of Differ data table starts from an offset defined by ExDiffBuf0.

ExDiffEn: DiffEn=1 means enable report differ data.

ExDiffPackNum: The package number of differ data. When Diffen=0, ExDiffPackNum is always 0.

ExDiffBuf0: The address offset of differ data in protocol map**.**

DiffChecksum: The checksum value calculated from IDinf0 to the last differ data.

Differ data table will be updated while XY coordinates are updated. Host must read all coordinates and all differ data table to notify touch controller to prepare next coordinates.