Sitronix

Sitronix Touch IC

Touch Screen Controller

I2C & SPI Interface

Protocol A

Version 2.7

2017/08/31

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.

Sitronix Technology Corporation

CONTENT

1. REVISION HISTORY

Version	Data	Author	Description
2.7	2017/08/31	Chun-Liang Chen	1. Remove “Continual Long Press” gesture from Advanced Touch Information Register. 2. Add “The End of Long Press” gesture into Advanced Touch Information Register.
2.6	2017/08/16	Chun-Liang Chen	1. Add “Double Taps”, “Long Press”, “Continual Long Press” and “Drag” gestures into Advanced Touch Information Register.
2.5	2017/06/05	Chun-Liang Chen	1. Add Water Flag information into Advanced Touch Information Register.
2.4	2017/01/05	Chun-Liang Chen	1. Add Deep Power Down control bit at register address 0x02. 2. Add PWM Control Registers into Host Interface Registers.
2.3	2016/06/30	Chun-Liang Chen	1. Add SPI waveform figure.
2.2	2016/05/20	Chun-Liang Chen	1. Add SPI host interface protocol section. 2. Modify "Initialization" section.
2.1	2016/02/17	Chun-Liang Chen	1. Remove "Set XY Resolution" sample code.
2.0	2015/11/12	Chun-Liang Chen	1. Add "Smart Wake Up Flag" information into "Misc. Info." register. 2. Add "Smart Wake Up Enable" control bit into "Misc. Control" register. 3. Add "Smart Wake Up ID" register.
1.9	2015/05/14	Chun-Liang Chen	1. Add single tap gesture. 2. Remove double taps gesture.
1.8	2015/04/17	Chun-Liang Chen	1. Add "Gesture Type" information at register address 0x10.
1.7	2015/01/14	Chun-Liang Chen	1. Add ST19xx part numbers into this specification. 2. Add descriptions for “Multi-Touch Disable” control bits.
1.6	2014/12/22	Chun-Liang Chen	1. Add FW initialization information into “I2C Electrical Waveform” section.
1.5	2014/12/16	Chun-Liang Chen	1. Add ST1x33i/ST1x33/ST1x24 part numbers into this specification.
1.4	2014/12/12	Chun-Liang Chen	1. Modify “XY Resolution” registers to read only registers. 2. Modify the declaration at page footer.
1.3	2013/01/17	Chun-Liang Chen	1. Add “Proximity Flag” at register address 0x10.

			2. Add “Proximity Enable” control bits at register address 0x02.
1.2	2011/07/26	Chun-Liang Chen	1. Add “Multi-Touch Disable” control bits at register address 0x02. 2. Add two error codes, “Invalid Firmware Parameter Table” and “Invalid Secondary Touch Firmware”.
1.1	2011/04/10	Chun-Liang Chen	1. Remove “Fingers” information at register address 0x10. 2. Resolution of X and Y coordinates of the touch screen are 0 ~ 2047.
1.0	2011/12/08	Chun-Liang Chen	First release.

2. 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.

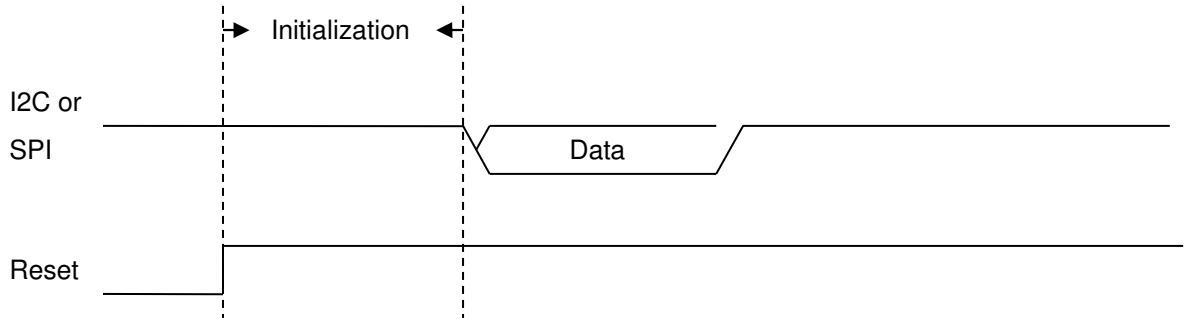
The capacitive touch sensor is covered with a plastic or glass cover lens. It provides auto-calibrate parameters for a wide range of capacitance on the touch sensor. The touch sensor controller converts touch sensor data into X and Y coordinates for each finger according to the motions of fingers detected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

3. HOST INTERFACE PROTOCOL

3.1. 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
ST1912/ST1727	65ms
ST1x32/ST1x28/ST1x30/ ST1x34/ST1x36/ST1x33i/ ST1x33/ST1x24/ST1615	50ms



3.2. I2C Host Interface Protocol

All Sitronix Touch ICs support I2C interface protocol for communication.

3.3. Default I2C Address

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

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

		(a)								

Figure 1 - Register Read Format.

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 in Figure 2.

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

Figure 2 - Repeated Register Read.

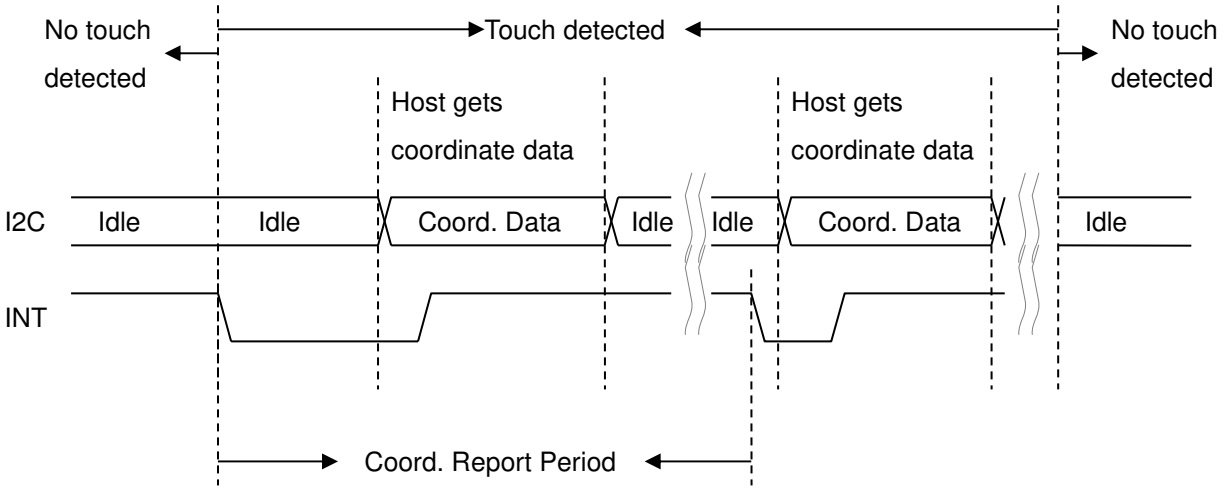
3.5. Register Write

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 in Figure 3.

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

Figure 3 - Register Write Format.

3.6. I2C Electrical Waveform



3.7. SPI Host Interface Protocol

Some Sitronix Touch ICs, like ST1727, support SPI interface protocol for communication.

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.
Others	Reserved.

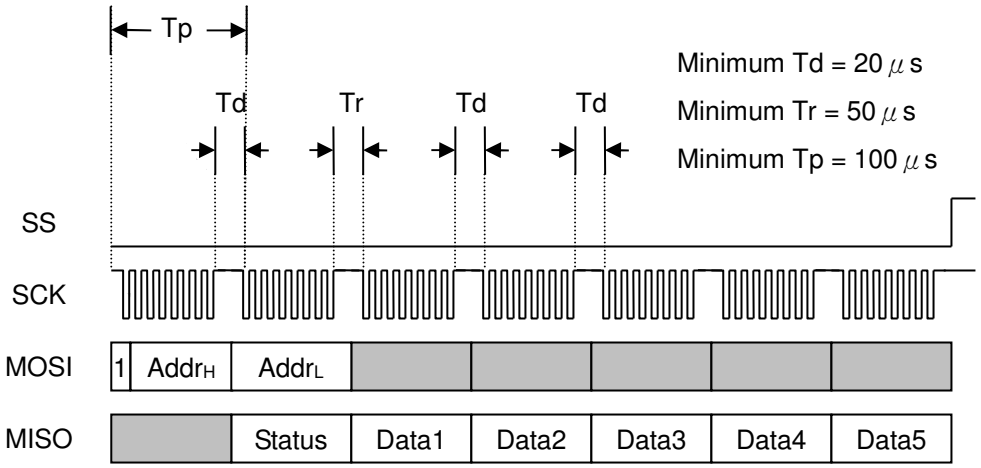
While touch controller is in Power Down mode, pulling SS pin to low triggers touch controller to wake up from Power Down mode. In this situation, touch controller needs a few time (100 μs) for hardware warm up and 1st SPI transmission data before 2nd SPI transmission starts.

The specific SPI read and write transaction are described as following.

3.8. Register Read

The following figure presents a typical SPI read transaction.

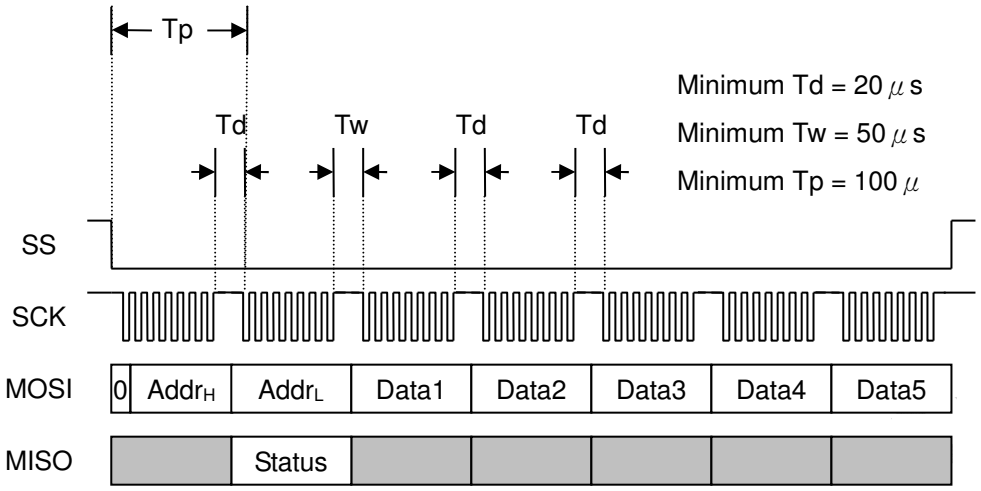
The limitation of Tp can be ignored if touch controller is not in power down mode.



3.9. Register Write

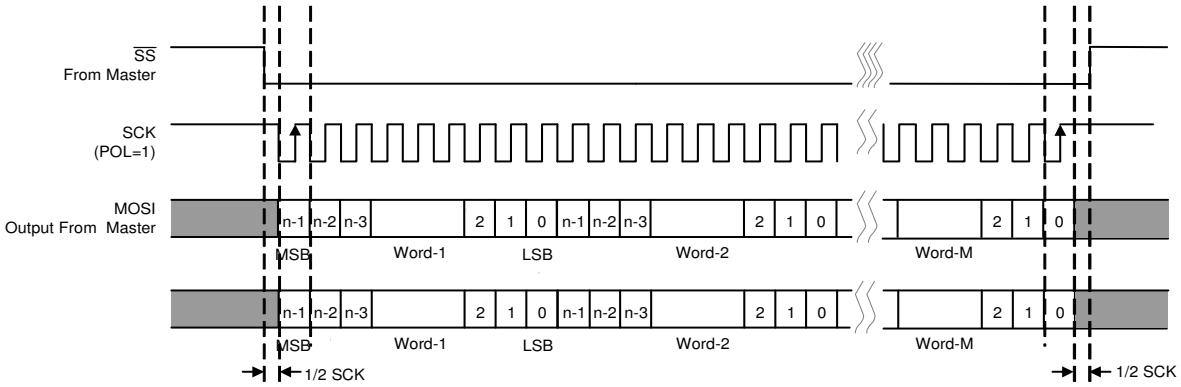
The following figure presents a typical SPI write transaction.

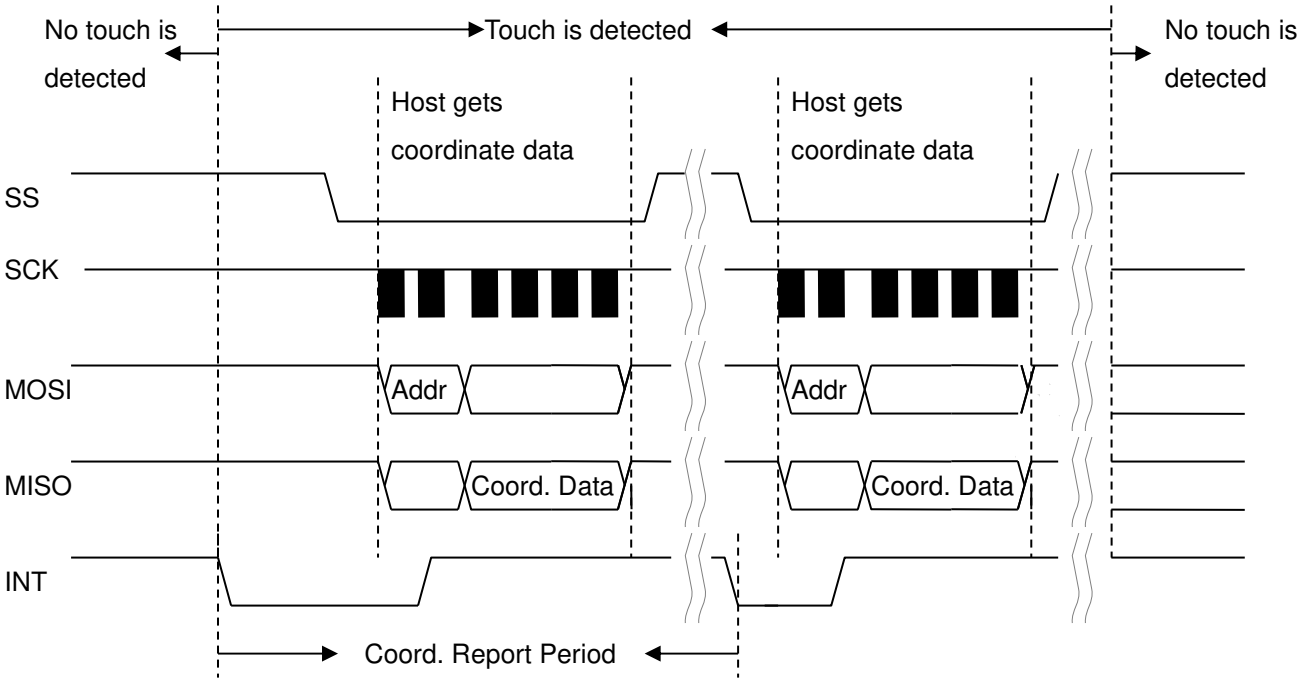
The limitation of Tp can be ignored if touch controller is not in power down mode.



3.10. SPI Electrical Waveform

Sitronix Touch ICs support SPI mode 3 (PHA = 1 & POL = 1) for communication. The SPI clock can be up to 8MHz.





4. 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**
0x00	Firmware Version	Version (RO)							
0x01	Status Reg.	Error Code (RO)				Device Status (RO)			
0x02	Device Control Reg.	*Reserved*	Multi-Touch Disable (RW)	Proximity Enable (RW)	*Reserved*	*Reserved*	Deep Power Down (RW)	Power Down (RW)	Reset (RW)
0x03	Timeout to Idle Reg.	Timeout to Idle (sec.) (RW)							
0x04	XY Resolution (High Byte)	*Reserved*	X_Res_H (RO)			*Reserved*	Y_Res_H (RO)		
0x05	X Resolution (Low Byte)	X_Res_L (RO)							
0x06	Y Resolution (Low Byte)	Y_Res_L (RO)							
0x07	Sensing Counter (High Byte)	Sensing_Counter_H (RO)							
0x08	Sensing Counter (Low Byte)	Sensing_Counter_L (RO)							
0x09 … 0x0B	…	*Reserved*							
0x0C	Firmware Revision 3	FW_Rev_3 (RO)							
0x0D	Firmware Revision 2	FW_Rev_2 (RO)							
0x0E	Firmware Revision 1	FW_Rev_1 (RO)							
0x0F	Firmware Revision 0	FW_Rev_0 (RO)							
0x10	Advanced Touch Info.	*Reserved*	Proximity Flag (RO)	Water Flag (RO)	*Reserved*	Gesture Type(RO)			
0x11	Keys Reg.	Keys (RO)							
0x12	XY0 Coord. (High Byte)	Valid 0 (RO)	X0_H (RO)			*Reserved*	Y0_H (RO)		

Host Interface Registers (Report Page)									
Reg. Addr.	**Name**	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
0x13	X0 Coord. (Low Byte)	X0_L (RO)							
0x14	Y0 Coord. (Low Byte)	Y0_L (RO)							
0x15	…	*Reserved.*							
0x16	XY1 Coord. (High Byte)	Valid 1 (RO)	X1_H (RO)			*Reserved*	Y1_H (RO)		
0x17	X1 Coord. (Low Byte)	X1_L (RO)							
0x18	Y1 Coord. (Low Byte)	Y1_L (RO)							
0x19	…	*Reserved.*							
0x1A … 0x35	…	…							
0x36	XY9 Coord. (High Byte)	Valid 9 (RO)	X9_H (RO)			*Reserved*	Y9_H (RO)		
0x37	X9 Coord. (Low Byte)	X9_L (RO)							
0x38	Y9 Coord. (Low Byte)	Y9_L (RO)							
0x39	Reserved	*Reserved.*							
0x3A … 0x3E	…	*Reserved*							
0x3F	Contact Count Max.	Max Number of Contacts Support (RO)							
0x40 … 0xCA	…	*Reserved*							
0xCB	PWM0 Duty	*Reserved*	PWM0 Duty (RW)						
0xCC	PWM1 Duty	*Reserved*	PWM1 Duty (RW)						
0xCD	PWM2 Duty	*Reserved*	PWM2 Duty (RW)						
0xCE	PWM3 Duty	*Reserved*	PWM3 Duty (RW)						
0xCF	PWM Control	PWM Trigger (RW)	PWM Clock (RW)			PWM3 Enable (RW)	PWM2 Enable (RW)	PWM1 Enable (RW)	PWM0 Enable (RW)
0xD0 … 0xEF	…	*Reserved*							
0xF0	Misc. Info.	Smart Wake Up Flag (RO)	*Reserved*						

Host Interface Registers (Report Page)									
Reg. Addr.	**Name**	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
0xF1	**Misc. Control**	**Enable Smart Wake Up (RW)**	***Reserved***						
0xF2	**Smart Wake Up ID**	**Smart Wake Up ID (RW)**							
0xF3 … 0xFE	**…**	***Reserved***							
0xFF	**Page Reg.**	**Page Number (RW)**							

Figure 4 – Host Interface Registers

4.1. Firmware Version Register

Reg. Addr.	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x00	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.

4.2. Status Register

Reg. Addr.	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x01	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* gives reason of the error.

Device Status	
0x0	Normal
0x1	Init
0x2	
0x3	Auto Tuning
0x4	
0x5	Power Down
0x6	Boot ROM
0x7	Waiting to execute Sub-AP
0x8 … 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
0x7	Invalid Secondary Touch Firmware

Error Code	
0x8 … 0xF	*Reserved*

4.3. Device Control Register

Reg. Addr.	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x02	Device Control Reg.	*Reserved*	Multi-Touch Disable (RW)	Proximity Enable (RW)	*Reserved*		Deep Power Down (RW)	Power Down (RW)	Reset (RW)

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

The “Multi-Touch Disable” control bit is only available for ST1236/ST1336/ST1530/ST1536 touch IC.

“Multi-Touch Disable” control bit is used to configure touch detector as single touch or multi-touch detector.

The default setting of this control bit is cleared to 0 and touch device can report multiple touch positions.

Set “Multi-Touch Disable” control bit to 1 makes the touch device to report only one touch position.

The “Multi-Touch Disable” control bit is useless in triangle projects.

For ST1x56/ST1x64/ST1x64A/ST1x72 series touch IC:

When host sets Power Down bit, touch sensor controller will enter power down mode. Host can pull I2C INT pin to low to wake up the controller.

For ST1x32/ST1x28/ST1x30/ST1x34/ST1x36/ST1x33i/ST1x33/ST1x24/ST1615/ST1912/ST1727/ST1727 series touch IC:

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.

The “Proximity Enable” control bit is only for some triangle projects.

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.

For ST1912/ST1727 touch IC:

Only ST1912/ST1727 touch IC supports *Deep Power Down* function. *Deep Power Down* bit provides deep power down mode to save more power consumption than *Power Down* mode. In *Power Down* mode, the power of DRAM in touch IC is still turned on. But in *Deep Power Down* mode, the power of DRAM in touch IC is turned off. All previous state information which are stored in DRAM will be missing. So, touch IC needs more initialization time for leaving *Deep Power Down* mode.

Please always write 0 into reserved bits.

4.4. Timeout to Idle Register

Reg. Addr.	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x03	Timeout to Idle Reg.	Timeout to Idle (sec.) (RW)							

Timeout to Idle Register provides timeout control to enter Idle Mode for host. The touch controller will enter Idle Mode after the number of seconds specified in Timeout to Idle Register if there is no touch detected in this period. Set this field to 0xFF will disable Idle Mode. Set this field to 0 will entering Idle Mode immediately. Idle state will be updated to Device Status field of Status Register, 0x01, after entering Idle Mode automatically. The default value of Timeout to Idle Register is set to 0x08 for 8 seconds to Idle Mode.

4.5. XY Resolution Registers

Reg. Addr.	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x04	XY Resolution (High Byte)	Reserved	X_Res_H (RO)			Reserv	Y_Res_H (RO)		
0x05	X Resolution (Low Byte)	X_Res_L (RO)							
0x06	Y Resolution (Low Byte)	Y_Res_L (RO)							

XY Resolution Registers represents resolution of X and Y coordinates of the touch screen.

4.6. Sensing Counter Registers

Reg. Addr.	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x07	Sensing Counter (High Byte)	Sensing_Counter_H (RO)							
0x08	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.

4.7. Firmware Revision Registers

Reg. Addr.	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x0C	Firmware Revision 3	FW_Rev_3 (RO)							
0x0D	Firmware Revision 2	FW_Rev_2 (RO)							
0x0E	Firmware Revision 1	FW_Rev_1 (RO)							
0x0F	Firmware Revision 0	FW_Rev_0 (RO)							

Firmware Revision Registers provide revision information about current firmware.

4.8. Advanced Touch Information Register

Reg. Addr.	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x10	Advanced Touch Info.	*Reserved*	Proximity Flag (RO)	Water Flag (RO)	*Reserved*	Gesture Type (RO)			

Advanced Touch Information field provides some advanced touch information, like proximity, for host.

Touch controller sets proximity flag to 1 to notify host that the human body is very close to proximity sensor.

The “Proximity Flag” is only shown and used for some triangle projects.

If proximity flags are changed, the touch controller will output INT pin as low to notify host.

The “Water Flag” information is only shown and used for smart watch projects. It is used to notify host that the smart watch may fall into the water. Only ST1615 touch IC supports this information.

Water Flag = 0: The smart watch doesn’t fall into the water.

Water Flag = 1: The smart watch may fall into the water. In this situation, the touch controller will output INT pin as low to notify host whether touch is detected or not.

The "Gesture Type" information is only available for ST1615 touch IC.

It defines following gestures:

Gesture Type = 0: No gesture.

Gesture Type = 1: Double taps.

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: Palm.

Gesture Type = 9: Single tap.

Gesture Type = 10: Long press.

If a finger is fixed on touch panel and continues over a specified duration, the touch controller generates a Long Press (Gesture Type = 10) gesture and outputs INT pin as Low to notify host. If the finger is still fixed on touch panel after Long Press gesture and over the specified duration again, the touch controller will generate a Long Press (Gesture Type = 10) gesture again and output INT pin as Low to notify host.

Gesture Type = 11: The end of long press.

If the finger leaves from the touch panel after Long Press gesture, the touch controller generates a The End of Long Press gesture and outputs INT pin as Low to notify host.

Gesture Type = 12: Drag.

If the finger moves on the touch panel after Long Press gesture, the touch controller generates a Drag gesture and outputs INT pin as Low to notify host.

The timing of touch controller to update latest information into “Advanced Touch Information Register”, “Keys Register” and “XY Coordinate Registers” is when host reads register data via I2C interface with specified start address 0x10 or 0x11 or 0x12.

4.9. Keys Register

Reg. Addr.	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x11	Keys	Keys (RO)							

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.

4.10. XY Coordinate Registers

Reg. Addr.		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x12	XY0 Coord. (High Byte)	Valid 0 (RO)	X0_H (RO)			*Reserved*	Y0_H (RO)		
0x13	X0 Coord. (Low Byte)	X0_L (RO)							
0x14	Y0 Coord. (Low Byte)	Y0_L (RO)							
0x15	…	*Reserved.*							

XY Coordinate Registers represent 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.

4.11. Maximum Number of Contacts Support Register

Reg. Addr.	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x3F	Contact Count Max.	Max Number of Contacts Support (RO)							

It's a read-only feature for getting the total number of contacts that the touch sensor controller supports.

4.12. PWM Control Registers

Reg. Addr.	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0xCB	PWM0 Duty	*Reserved*	PWM0 Duty (RW)						
0xCC	PWM1 Duty	*Reserved*	PWM1 Duty (RW)						
0xCD	PWM2 Duty	*Reserved*	PWM2 Duty (RW)						
0xCE	PWM3 Duty	*Reserved*	PWM3 Duty (RW)						
0xCF	PWM Control	PWM Trigger (RW)	PWM Clock (RW)			PWM3 Enable (RW)	PWM2 Enable (RW)	PWM1 Enable (RW)	PWM0 Enable (RW)

Only ST1912/ST1727 touch IC support *PWM Control Registers* field.

The *PWM Control Registers* provide some PWM configuration like clock, duty and enable/disable control.

Host can configure the PWM0 ~ PWM3 modules via *PWM Control Registers*.

PWM0 ~ PWM3 Duty registers control the high level time of each PWM duty.

0x00 = 0/64

0x01 = 1/64

0x02 = 2/64

: :

0x3F = 63/64

0x40 = 64/64

PWM0 ~ PWM3 Enable bits are used to turn on/off each PWM module.

0 = Disable PWM. (Turn off PWM)

1 = Enable PWM. (Turn on PWM)

PWM Clock bits is used to select the PWM clock source for all PWM modules.

000 = 1000 Hz

001 = 500 Hz

010 = 250 Hz

011 = 125 Hz

100 = 62.5 Hz

PWM Trigger bit is used to update new PWM configuration into all PWM module. Host sets PWM Trigger bit to notify touch IC that some PWM control registers have been modified. And touch IC updates new PWM configuration into all PWM modules when it detects that the PWM Trigger bits is set. Touch IC will clear the PWM Trigger bit automatically after it finish the PWM configuration. All PWM control registers modification, including PWM duty, PWM clock and PWM enable, will not be updated into PWM modules until host sets PWM Trigger bit.

4.13. Miscellaneous Information Register

Reg. Addr.	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0xF0	Misc. Info.	Smart Wake Up Flag (RO)	***Reserved.***						

Miscellaneous Information Register provides some misc. information to host.

The "Smart Wake Up" function is an optional for customer. The "Smart Wake Up Flag" shows that does 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.

4.14. Miscellaneous Control Register

Reg. Addr.	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0xF1	Misc. Control	Enable Smart Wake (RW)	***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 “Doze” mode after power down. In this mode, touch driver is still sensing the touch panel but as saving power as possible.

Once the specified handwriting gesture is detected, touch controller wakes host up via “INT” pin and the identification of handwriting gesture will be put into “Smart Wake Up ID” register. After host gets “Smart Wake Up ID”, host can clear the “Smart Wake Up ID” register.

Please always write 0 into reserved bits.

4.15. Smart Wake Up ID Register

Reg. Addr.	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0xF2	Smart Wake Up ID	Smart Wake Up ID (RW)							

Smart Wake Up ID Register provides various handwriting identifications of smart wake up function.

After host gets “Smart Wake Up ID” from this register, host can clear this register to zero.

Smart wake up ID:

ID = 0: No any handwriting gesture is detected.

ID = 0xFF: Handwriting gesture detection is failure.

ID = 0xB0: Left to right slide (→).

ID = 0xB4: Right to left slide (←).

ID = 0xB8: Top to down slide (↓).

ID = 0xBC: Down to top slide (↑).

ID = 0xC0: Double taps.

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.

4.16. Page Register

Reg. Addr.	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0xFF	Page Reg.	Page Number (RW)							

For ST1x56/ST1x64/ST1x64A/ST1x72 series touch IC:

The auto tune program is build-in into ST1x56/ST1x64/ST1x64A/ST1x72. *Page Register* provides changing page of Host Interface Register. Default page is Report Page.

Page Number	Description
0x00	Report Page
0x01	Auto Tune Page

For ST1x32/ST1x28/ST1x30/ST1x34/ST1x36/ST1x33i/ST1x33/ST1x24/ST1615/ST1912/ST1727/ST1727 series touch IC:

Page Register is a read only register. It can not change page by writing specified page number into this register.

5. SAMPLE CODES

5.1. Data Structures and APIs

```
typedef struct {
    u8  y_h: 3,
        reserved: 1,
        x_h: 3,
        valid: 1;
    u8  x_l;
    u8  y_l;
    u8  z;
} xyz_data_t;

typedef struct {
    u8          fingers: 4,
                reserved: 4;
    u8          keys;
    xyz_data_t      xyz_data[10];
} stx_report_data_t;

// I2C Master sends count bytes data stored in buf to I2C Slave.
// I2C package: | S | I2C Addr | W | Data (buf) | P |
extern int i2c_master_send(const char *buf, int count);

// I2C Master reads count bytes data to buf from I2C Slave.
// I2C package: | S | I2C Addr | R | Data (buf) | Nak | P |
extern int i2c_master_recv(char *buf, int count);
```

5.2. Get Version

```
static int get_fw_version(u32 *ver)
{
	u8 buf[1];
	int ret = 0;

	buf[0] = 0x0;				// Set Reg. address to 0x0 for reading FW Version.
	if (ret = i2c_master_send(buf, 1))
		goto err;
	if (ret = I2c_master_recv(buf, 1))	// Read 1 byte FW Version from Reg. 0x0 set previously.
		goto err;

	*ver = (u32) buf[0];			// Return FW Version.

	buf[0] = 0x10;				// Set Reg. address back to 0x10 for Coordinates.
	if (i2c_master_send(buf, 1))
		goto err;

err:
	return ret;
}
```

5.3. Set Power Down (PD)

```
static int set_power_down()
{
	u8 buf[2];
	int ret = 0;

	buf[0] = 0x2;				// Set Reg. address to 0x2 for Device Control Reg.
	buf[1] = 0xA;				// Keep Gesture bit and set PD bit to enter Power Down.
	if (ret = i2c_master_send(buf, 2))
		goto err;
err:
	return ret;
```

```
}
```

5.4. Read XY Coordinates

The function, get_coordinates(), reads XY Coordinate registers from I2C Slave, extracts XY information from data buffer and returns to upper layer. This function shall be called from ISR each time when host receives and INT from device.

```
static int get_coordinates(u8 *count, u32 *x0, u32 *y0, u32 *x1, u32 *y1)
{
	u8 buf[42];
	stx_report_data_t *pdata;
	int ret = 0;

	*count = 0;				// Set point detected count to 0.
	if (i2c_master_recv(buf, sizeof(buf)))	// Read Coordinates from default Reg. address 0x10.
		goto err;

	pdata = (stx_report_data_t *) buf;
	if (pdata->fingers) {
		if (pdata->xy_data[0].valid) {
			*x0 = pdata->xy_data[0].x_h << 8 | pdata->xy_data[0].x_l;
			*y0 = pdata->xy_data[0].y_h << 8 | pdata->xy_data[0].y_l;
			(*count)++;
		}
		if (pdata->xy_data[1].valid) {
			*x1 = pdata->xy_data[1].x_h << 8 | pdata->xy_data[1].x_l;
			*y1 = pdata->xy_data[1].y_h << 8 | pdata->xy_data[1].y_l;
			(*count)++;
		}
	}
err:
	return ret;
}
```