ENE TECHNOLOGY INC.

TS351x Software Application Guide

Revision 0.5
Sep. 2010

Headquarters
4F-1, No.9, Prosperity Rd.,
Science-based Industrial Park,
Hsinchu City, Taiwan, R.O.C
TEL: 886-3-6662888
FAX: 886-3-6662999
http://www.ene.com.tw

Taipei Office
4F, No.88, Bauchiau Rd.
Shindian City, Taipei,
Taiwan, R.O.C.
TEL: 886-2-89111525
FAX: 886-2-89111523

Revision

Revision	Description	Date
0.1	1st release	2010/07
0.2	1. Update the access for touch-related application data	2010/07
0.3	1. Update Example	2010/07
0.4	1. Extend FW version description 2. Fix data format description	2010/08
0.5	1. Updated and extend data format	2010/09

Content

Introduction

TS351x is ENE capacitive sensor controller for capacitive button and pad applications. TS351x supports SMBus and SPI interface to communicate with host.

Before using either interface of SMBus or SPI, hardware trap should be correctly set. The pins GPIO10/GPIO11 are used for hardware trap to configure the interface. Please be noted that the Pin NO is different by different parts.

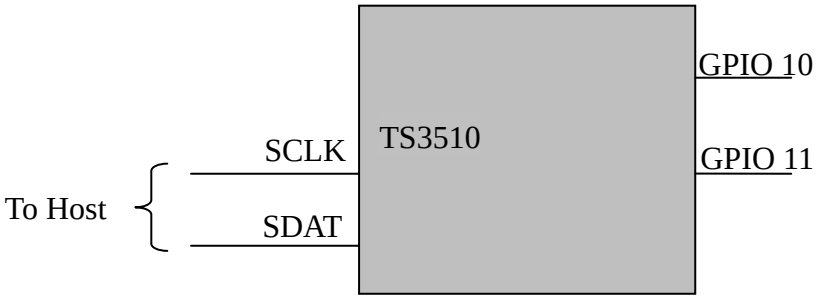
	GPIO10 Pin NO.	GPIO11 Pin NO.
TS3510	52	53
TS3511	44	45
TS3512	36	37

GPIO10 Trap Value	GPIO11 Trap Value	Interface	SMB HW Trap Address Bit
L	L	SMB	Address 0 and F/W configurable
L	H		Address 1 and F/W configurable
H	L	Reserved	
H	H	SPI	

1 TS351x SMBus interface descriptions

TS3510 acts as a SMBus slave device, and supports SMBus read and SMBus write protocols.

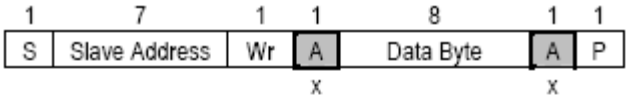
1.1 SMBus Signals Description



SCLK : SMBus clock signal
SDAT : SMBus data signal
GPIO10, GPIO11 : Hardware trap pin
Note: GPIO 10,11 should be trapped correctly to select SMBus and select slave address.

1.2 SMBus Protocols

TS3510 supports two SMBus protocols, one is write protocol and another is read protocol. General SMBus protocol is as following:



S	Start Condition
Sr	Repeated Start Condition
Rd	Read (bit value of 1)
Wr	Write (bit value of 0)
x	Shown under a field indicates that that field is required to have the value of 'x'
A	Acknowledge (this bit position may be '0' for an ACK or '1' for a NACK)
P	Stop Condition
PEC	Packet Error Code
☐	Master-to-Slave
▩	Slave-to-Master
...	Continuation of protocol

TS3510 also supports multiple SMBus slave addresses which allows multiple TS3510 devices to connect at the same bus.

Overall SMBus Slave Address should be :

1	**HW Trap**	**FW set**	**0**	**0**	**0**	**0**	**R/W**

Where the HW trap address bit value is

0 as GPIO10:GPIO11 = b'00

1 as GPIO10:GPIO11 = b'01

Where the FW set address bit value is in SMBADDR (0xF700)

In other words, the available SMBus Slave Address are as following depending R/W commands:

b' 1000 000x

b' 1010 000x

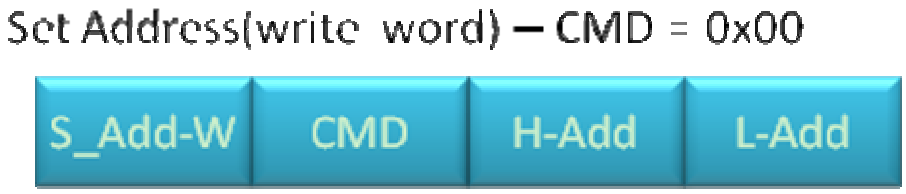
b' 1100 000x

b' 1110 000x

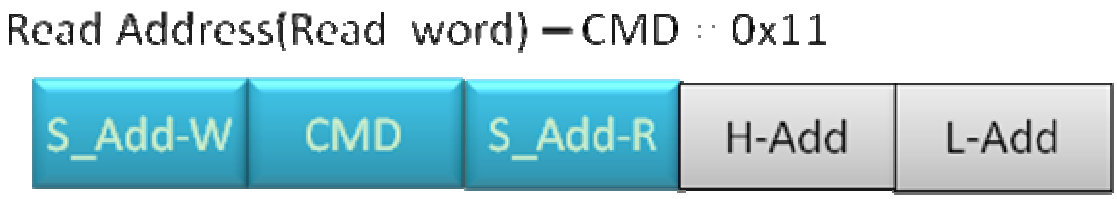
Setting the target address for Read/Write command

For R/W command to specific target registers such as Raw Count RAM for the sensing data. The target address should be set correctly by following commands:

To set address, use **Write Word** protocol with CMD=**0x00** as following illustration:

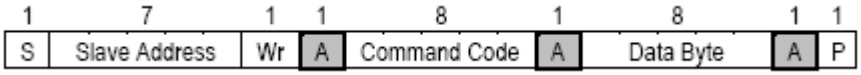


To read address, use **Read Word** protocol with CMD=**0x11** as following illustration:

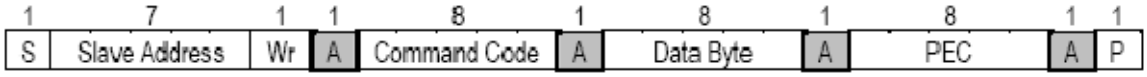


1.2.1 SMBus Write Protocol

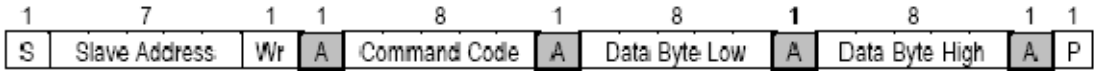
Write Byte : CMD = 0x01



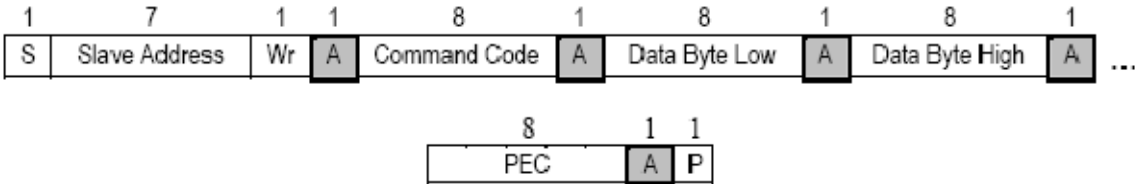
Write Byte with PEC : CMD = 0x01



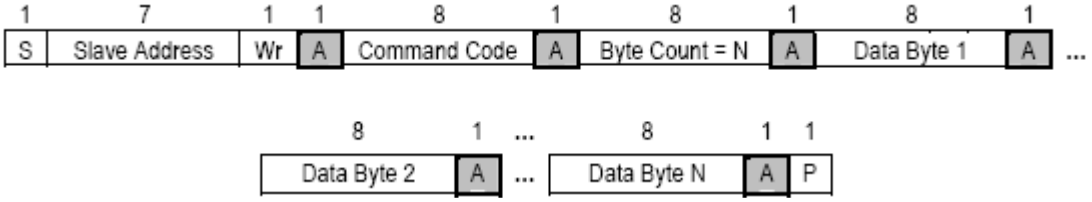
Write Word : CMD = 0x02



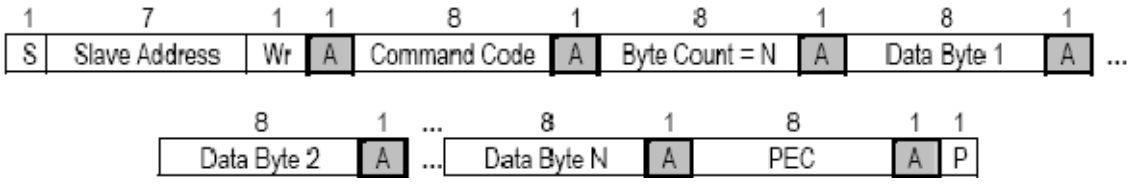
Write Word with PEC : CMD = 0x02



Write Block : CMD = 0x03

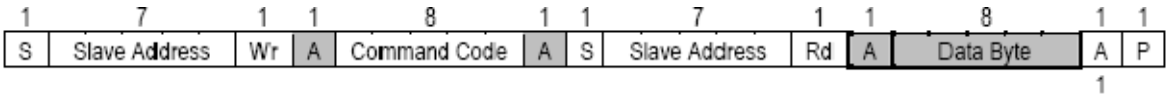


Write Block with PEC : CMD = 0x03

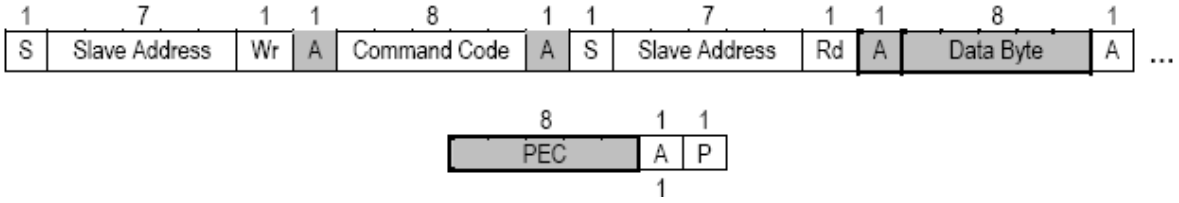


1.2.2 SMBus Read Protocol

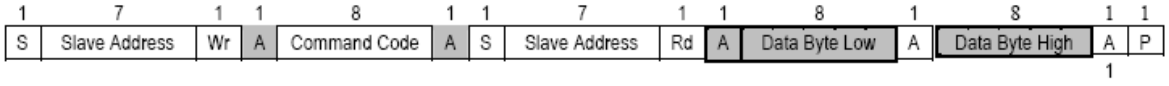
Read Byte : CMD = 0x81



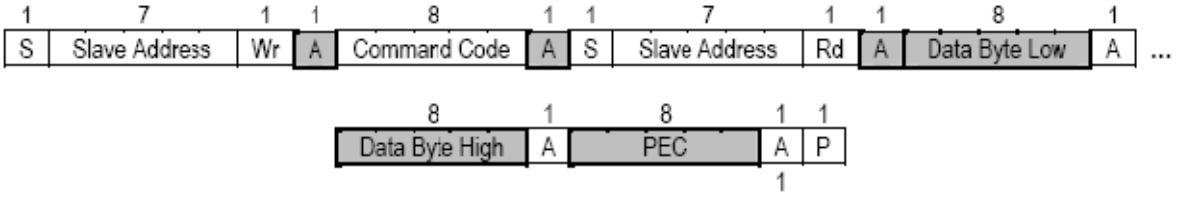
Read Byte with PEC: CMD = 0x81



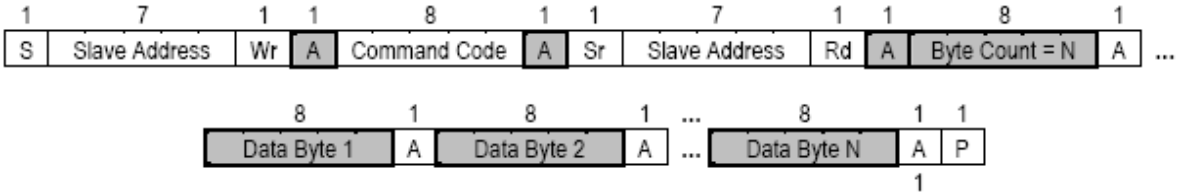
Read Word : CMD = 0x82



Read Word with PEC : CMD = 0x82



Read Block : CMD = 0x80 + Byte Count



Read Block with PEC : CMD = 0x80 + Byte Count

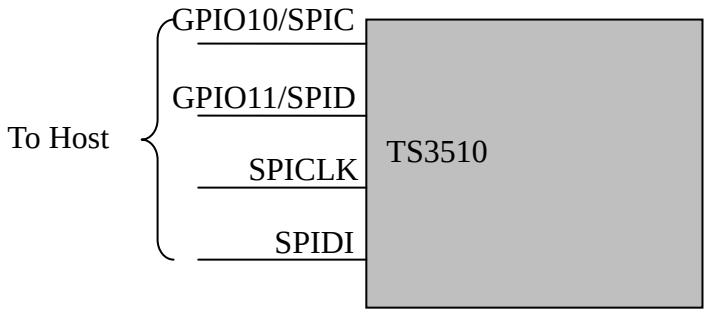
Example: To read X1, Y1 coordinate for specific touched event

```
SetAddress
{
      SMBDAT = 0xC0;       // If trap I2C Address 0xC0
      SMBDAT = 0x00;       // CMD 0x00 for set address
      SMBDAT = 0x81;       // Write chip Address, Hi Byte with 0x81
      SMBDAT = 0x08;       // Write chip Address, Low Byte with 0x08
}

GetInfo
{
      SMBDAT = 0xC0;       // If trap I2C Address 0xC0
      SMBDAT = 0x8A;       // Block read with byte count = 10
      SMBDAT = 0xC0;
      GetData // Host Read in data frame format as Section 3.2
                  // 1st , 2nd bytes are coordinate X, 3rd and 4th are coordinate Y

}
```

2 TS3510 SPI interface descriptions

TS3510 acts as a SPI slave device and supports both SPI write and read protocol.

2.1 SPI Related Signals



SPICLK : SPI clock source signal
SPICS : SPI chip select signal
SPIDI : SPI data input signal (MOSI)
SPIDO : SPI data output signal (MISO)

Note:

1. SPICSZ/GPIO10 and SPIDO/GPIO11 should be trapped both high to enable SPI interface.

2. SPI input clock is limited to the low frequency.

2.2 SPI Protocols

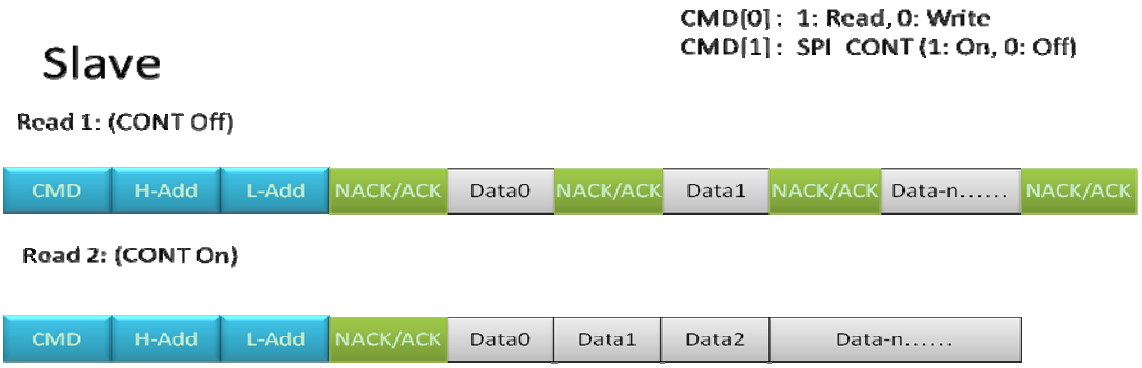
2.2.1 SPI Write Protocol

CMD: reference command table as below.

H-Add,L-Add: This 2 bytes are used to set write address.

Data: These bytes are input data from the host.

2.2.2 SPI Read Protocol



CMD: reference command table as below.

H-Add,L-Add: This 2 bytes are used to set read address.

Data: These bytes are output data to the host.

2.2.3 SPI Command Byte

SPI Cmd	description
0x00	SPI WRITE CMD
0x01	SPI READ CMD
0x02	SPI WRITE CMD, data without ack/nack
0x03	SPI READ CMD, data without ack/nack

Note: SPI mode is strongly recommend using command 0x02, 0x03.

Example: To read X1, Y1 coordinate for specific touched event

1. CMD= 0x03, H-Add,L-Add=0x8108, Read 10bytes
2. GetData // Host Read in data frame format as Section 3.2
   // 1st , 2nd bytes are coordinate X, 3rd and 4th are coordinate Y

3 Access Touch-Related Application Data

3.1 Touch-Related Application Data Address

Address	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
0x8100	Firmware Version – Customer Code							
0x8101	Firmware Version – Product Code							
0x8102	Firmware Version – Program Type				Firmware Version – Serial NO. Bit[11:8]			
0x8103	Firmware Version – Serial NO. Bit[7:0]							
0x8104	Reserved							Interrupt Mode *
0x8105	Augment Bit[15:8] (Reserved for future define)							
0x8106	Augment Bit[7:0] (Reserved for future define)							
0x8107	CMD (Reserved for future define)							
0x8108	X1 coordinate [15:8]							
0x8109	X1 coordinate [7:0]							
0x810A	Y1 coordinate [15:8]							
0x810B	Y1 coordinate [7:0]							
0x810C	X2 coordinate [15:8]							
0x810D	X2 coordinate [7:0]							
0x810E	Y2 coordinate [15:8]							
0x810F	Y2 coordinate [7:0]							
0x8110	Gesture CMD *							
0x8111	Gesture Toggle (Used for valid gesture hand-shaking)							
Address	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

* Description in later sections.

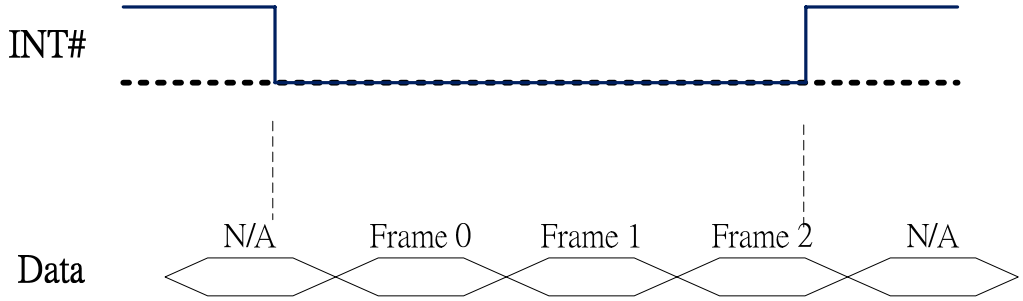
3.1.1 Interrupt Mode:

TS351x provides two different interrupt behaviors to response host control.

Mode 0 (0x8104[0]=b'0, Default Mode):

Interrupt starts at touch-down event and ends at touch-lift event.

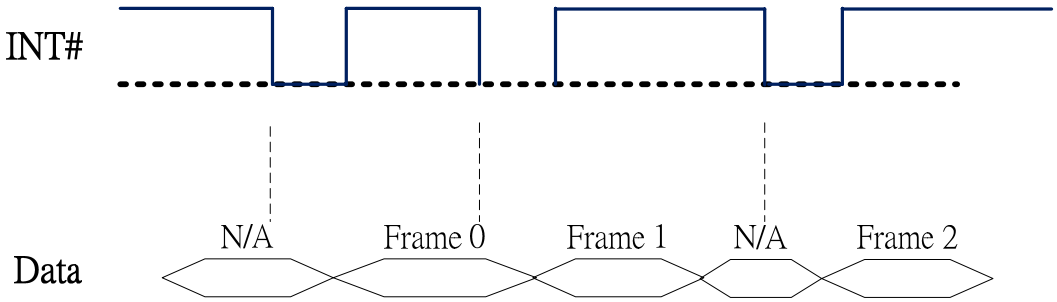
During the period, coordinate report rate is related to the rate which host issues read-coordinate command.



Mode 1(0x8104[0]=b'1):

Interrupt starts and ends whenever TS351x coordinate events.

The coordinate report rate is at TS351x coordinate update rate. (Every success touch)



3.1.2 Gesture CMD:

Gesture CMD	Description
0x00	No gesture stored
0x01	Detection of pinch (May be used for zoom in)
0x02	Detection of de-pinch (May be used for zoom out)
0x03	Detection of finger flick up
0x04	Detection of finger flick down
0x05	Detection of finger flick left
0x06	Detection of finger flick right

3.2 Touch-Related Application Data Frame Format

To response the INT# of TS351x, host can access the desired data via SMBus/SPI protocol target specific address. We need to **read 10 bytes** from 0x8108 to 0x8111.
Here, we would illustrate with SMBus protocol.

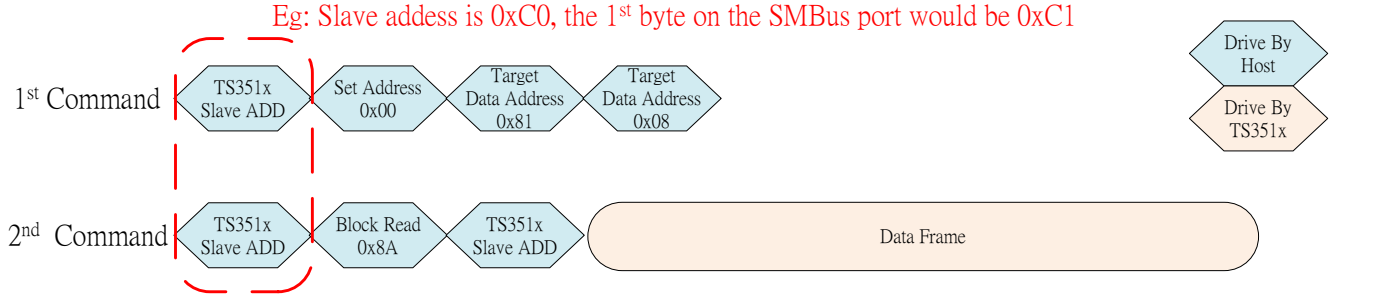
1. Issue 1st command to set address to 0x8108.

```
SetAddress
{
    SMBDAT = 0xC0;      // If trap I2C Address 0xC0
    SMBDAT = 0x00;      // CMD 0x00 for set address
    SMBDAT = 0x81;      // Write chip Address, Hi Byte with 0x81
    SMBDAT = 0x08;      // Write chip Address, Low Byte with 0x08 from X1
}
```

2. Issue 2nd command to read out the data frame.

```
GetInfo
{
    SMBDAT = 0xC0;      // If trap I2C Address 0xC0
    SMBDAT = 0x8A;      // Block read command with byte count = 10
    SMBDAT = 0xC0;
    ContinousGetData()  // Host Read in the data frame format as below
}
```

Please Note the 1st Slave ADD bit 0 should be 1 For SMBus Write Bit,
Eg: Slave addess is 0xC0, the 1st byte on the SMBus port would be 0xC1



Data Frame Structure: