ZET6221

Product Specification

[Doc.No: ZP-HW-PS-0003 Doc. Type: Product Specification. Revision Number: 1.2]

NOTICE TO PERSONS RECEIVING THIS
DRAWING AND/OR TECHNICAL INFORMATION

IDENTIFICATION

Document Title	**ZET6221 Product Specification**
Document Number	ZP-HW-PS-0003
Version	Version 1.2
Date	22-May-12

APPROVALS

Authority	Name	Date
Author	Alex	6-July-11
Author	Caspar	23-Sep-11
Author	Alex	11-Jan-12
Author	Caspar	7-Mar-12

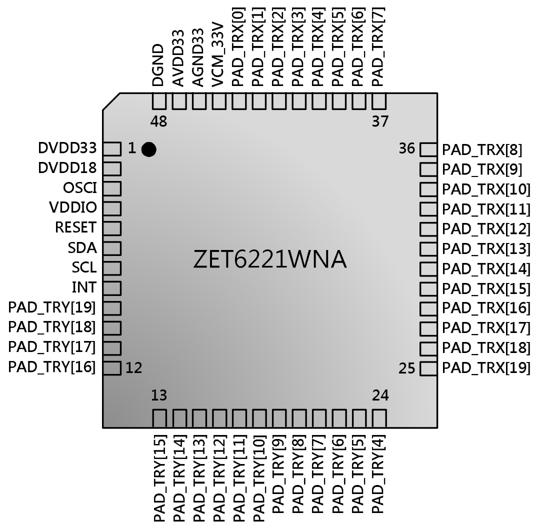
HISTORY

Version	Date	Purpose	Pages
0.1	26 –Feb-11	Initial version	All
0.2	2-Mar-11	Revise figure 3.2	All
0.3	27-Apr-11	Update I2C communication format	All
0.4	6-July-11	Update I2C communication format	All
0.5	13-Sep-11	1. Add command mode 2. Update dynamic I2C data format 3. Define power saving mode	All
0.6	14-Nov-11	Modify I2C reference code	P.25
0.7	9-Dec-11	Fixed error spec. and words	All
1.0	11-Jan-12	Add QFN48 package and application ckt.	All
1.1	7-Mar-12	Update to 5 Finger I2C Address = 0x76 and Boot Loader Flow	All
1.2	22-May-12	Modify I2C key number define Remove QFN56 and QFN28 Package	All

36-ch, touch panel controller

ZET6221 is a single chip solution for capacitive touch panel applications. You can use one ZET6221 to implement touch panel measuring up to 10.1-inch diagonally. It comprises a high performance capacitive-to-digital analog front end, x51 based MCU, and embedded flash memory. The touch solution supports I2C and GPIO interfaces for most of the devices as mobile phone, PND, digital camera, e-book, tablet, etc.

- **System operating voltage :**
  2.6V ~ 3.6V
- **System operating temperature :**
  -20 °C ~ 85 °C
- **40 touch sense channels :**
  Standard type : Up to 10.1"
- **Flexible capacitive sense method :**
  Mutual-Capacitive-Sensing
- **Minimize current consumption :**
  -- Active Operation Mode:
  Touch : 4mA (Typ.)
  Non-Touch : 1mA (Typ.)
  -- Idle Mode : 140uA (Typ.)
  -- Sleep Mode is less than 35uA (Typ.)
- **True 5 fingers Multi-Touch supported**
- **High Signal to Noise Ratio > 100**
- **Fast touch response time :**
  Up to 120Hz/frame for single touch and 80Hz/frame for multi touch.
- **Easy to program touch panel characters by Zeki App.**
- **Up to 3mm cover lens thickness.**
- **On chip auto-calibration**
- **Environment variation insensitive**
  -- Temperature Drift Compensation
  -- LCM/RFI Noise Immunity
  -- Touch Trace Variation Insensitive
  -- Charger Noise Immunity
- **Interface supported :**
  Inter-IC Interface (I2C)
- **Internal power-on reset and external reset control**
- **Customized built-in gesture function**
- **Electrostatic discharge noise :**
  Human Body Mode ≥ ±2KV
  Machine Mode ≥ ±200V
  Latch up ≥ ±200mA, +2V
- **Package features:**
  48/40 pins QFN



The ZEITEC IC is designed to detect finger touch on up to max 40 traces when used with a scanned, mutual mode. ZET6221 can be applied to almost any kind of materials, e.g. glass, plastic, and even air gap cardboard, up to thicknesses of 3 mm or more. The touch areas are defined as 2-part integrated conductive material, like copper or screened silver or carbon deposited on the rear of a covered panel. Panel shapes, sizes and placement are almost arbitrary. The sensitivity of each panel can be configured individually via simple command over the I2C port by our touch panel develop kit, Zeki. All Key parameters are stored and downloaded to main ROM as new parameter.

The ZET6221 is designed for appliances such as electronic control, control panels, personal instruments, or similar products, whose touch panels that need to be immunity to temperature, humidity, dirt accumulation, or the physical deterioration of the panel surface from abrasion, chemicals, or abuse. For this purpose, ZET6221 implemented environmental detection and self-calibration, temperature compensation and noise filtering algorithms that make the sensing functions robust. The ZET6221 features continuous self-diagnosis and reporting status, to allow its use in critical consumer appliance applications, even for application debug.

General PCB or FPC materials can be used as the circuit substrate; the overlying panel can be made of any insulation material. External circuitry consists of only a few passive components. Touch control and data transfer is via I2C port. Devices using ZEITEC advanced sensing technologies can achieve high sensitivity and high noise immunity performance.

The Touch Panel Module is a capacitive-type which customers use to conform to flat panel display. Once user's finger touches Touch Panel Module, this Touch Panel Module will response the coordinates of points that user touches and finger number.

Table of Contents

List of Figures

List of Tables

1 PRODUCT OVERVIEW

1.1 General Description

ZET6221 is a single chip solution for projected capacitive touch panel and touch pad applications. It includes all signal processing functions to provide stable sensing under a wide variety of changing conditions. Only a few external components are required in BOM.

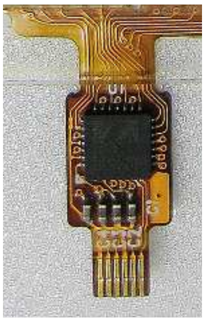


Figure 1-1 ZEITEC FPC Bom

The entire circuit can be built within a few square centimetres of single-sided FPC or PCB area (Figure 1-1).

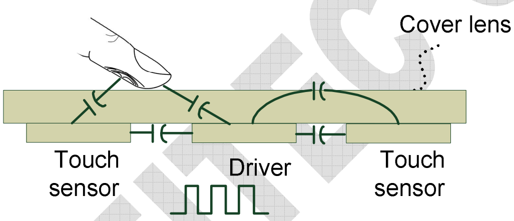


Figure 1-2 Touch Sensor

ZET6221 employs advance touch sensing, a technology that senses changes in electrical charge forced across an electrode by a pulse edge. The devices use I^2C interface to allow parameter data to be extracted and to permit individual touch panel parameter setup. The interface protocol uses simple single byte commands and responds with single byte responses in most cases. The command structure is designed to minimize the amount of data traffic while maximizing the amount of information conveyed.

In addition to normal operating and setup functions, ZET6221 can also report back actual signal strengths and error codes. Zeki software for the PC can be used to program the operation of the IC as well as read back touch sensor status and signal levels in real time.

1.2 Easy to Design-in

Zeki is designed for fine-tune firmware algorithm to fit various ITO characteristics in an easy and fast way to shorten design-in period.

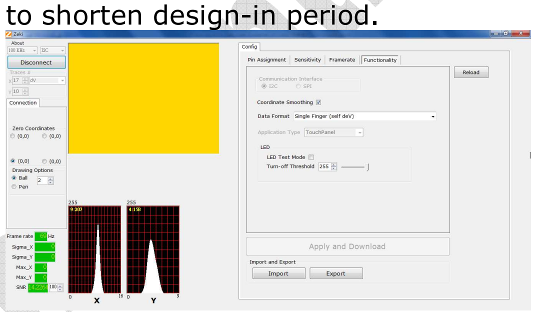


Figure 1-3 Zeki

Configuration is very easy to change for ZET6221 by using "Zeki" (Figure 1-3). Zeki provides a friendly and straight forward GUI to help users configure sensor trace numbers, defined parameters and can generate application setting code which can be reloaded to ZET6221 IC.

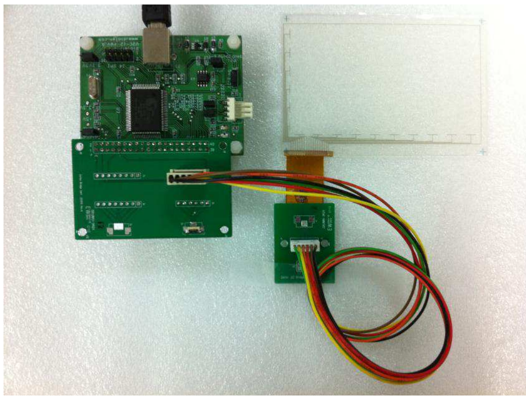


Figure 1-4 Zeki Connects to Touch Panel

Connecting Zeki hardware demo board or customer FPC with Zeki is shown as (Figure 1-4), sensor sensitivity, threshold, frame rate, touch run time status can be adjusted at any time (Figure 1-5).

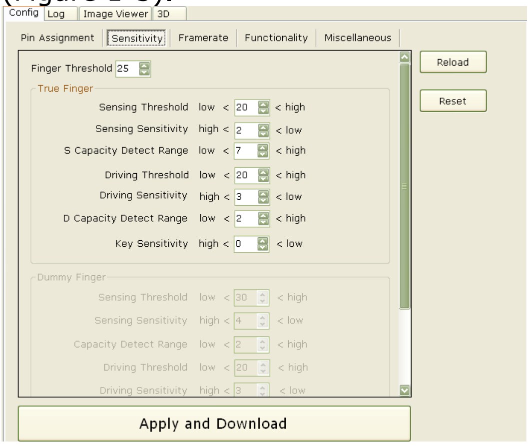


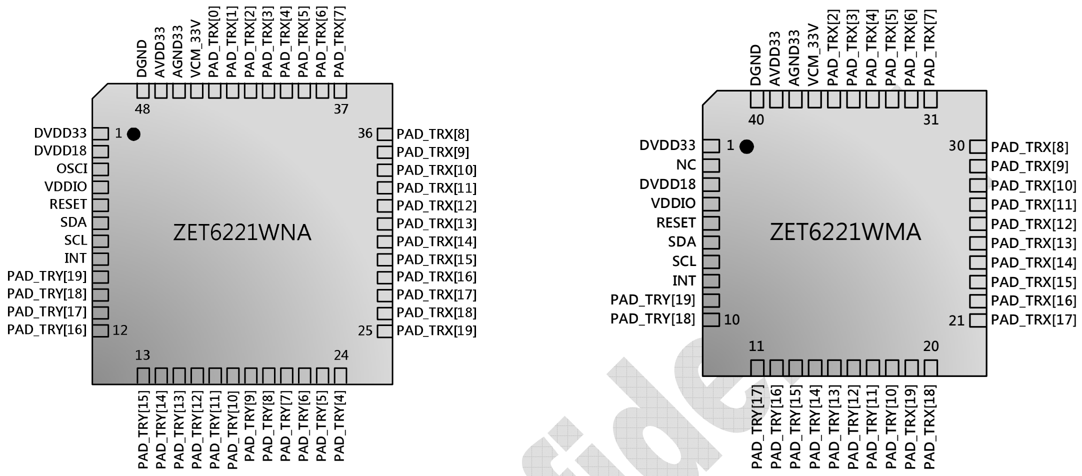
Figure 1-5 Sensitivity Settings in Zeki GUI

After completion of parameters setting such as sensitivity, framerate and other advanced options. Connecting to touch panel, Zeki can immediately show the result which is running by new settings. Customers can see what they configured immediately and to change settings back and forth until all functionalities works perfect. Finally, Zeki will save all parameters and reload to ZET6221 chip. A design of Touch product is completed, quickly and easily.

2 PRODUCT INFORMATION

2.1 Package Type

ZET6221 supports 2 types of package QFN48 and QFN40, pin assignment is shown below.



QFN48, ZET6221WNA, 6mm*6mm

QFN40, ZET6221WMA, 5mm*5mm

Figure 2-1 Pin Assignment

2.2 Pin Descriptions

Table 2-1 Pin Descriptions

QFN TYPE PIN		Symbol	Type	Description
48	40			
1	1	DVDD33	P	**Power Supply :** Chip power supply voltage source. It generally needs one or two decoupling capacitor to DGND. One is 1uF. Another is 0.01uF ~ 1uF.
3	-	OSCI	I	**OSCI :** For Zeitec Testing.
2	3	DVDD18	P	**Power Supply :** 1.8V internal regulator output power supply voltage source. It generally needs one or two decoupling capacitor to DGND. One is 1uF. Another is 0.01uF ~ 1uF.
4	4	VDDIO	P	**Power Supply :** Chip power supply voltage source. It generally needs one (1uF) decoupling capacitor to GND.
5	5	RESET	I	**External Reset :** RESET = 1, reset function is dis-activated. RESET = 0, reset function is activated. An internal 15KΩ±30% resistor is pulled high to DVDD33.
6	6	SDA	I/O	SDA : For Inter-IC interface (I2C) mode. Data input as slave device and output as master device. It need the pull high resistor to VDDIO (Internal 4.7KΩ±20% pull high resistor can be selected by Zeki).
7	7	SCL	I/O	SCL : For Inter-IC interface (I2C) mode. Clock input as slave device and output as master device. It need the pull high resistor to VDDIO (Internal 4.7KΩ±20% pull high resistor can be selected by Zeki).
8	8	INT	I/O	**Interrupt Output Single.**
9	9	PAD_TRY[19]	I/O	**Analog I/O of Capacitive Touch Sensor.**
10	10	PAD_TRY[18]	I/O	**Analog I/O of Capacitive Touch Sensor.**
11	11	PAD_TRY[17]	I/O	**Analog I/O of Capacitive Touch Sensor.**
12	12	PAD_TRY[16]	I/O	**Analog I/O of Capacitive Touch Sensor.**
13	13	PAD_TRY[15]	I/O	**Analog I/O of Capacitive Touch Sensor.**

14	14	PAD_TRY[14]	I/O	Analog I/O of Capacitive Touch Sensor.
15	15	PAD_TRY[13]	I/O	Analog I/O of Capacitive Touch Sensor.
16	16	PAD_TRY[12]	I/O	Analog I/O of Capacitive Touch Sensor.
17	17	PAD_TRY[11]	I/O	Analog I/O of Capacitive Touch Sensor.
18	18	PAD_TRY[10]	I/O	Analog I/O of Capacitive Touch Sensor.
19	-	PAD_TRY[9]	I/O	Analog I/O of Capacitive Touch Sensor.
20	-	PAD_TRY[8]	I/O	Analog I/O of Capacitive Touch Sensor.
21	-	PAD_TRY[7]	I/O	Analog I/O of Capacitive Touch Sensor.
22	-	PAD_TRY[6]	I/O	Analog I/O of Capacitive Touch Sensor.
23	-	PAD_TRY[5]	I/O	Analog I/O of Capacitive Touch Sensor.
24	-	PAD_TRY[4]	I/O	Analog I/O of Capacitive Touch Sensor.
25	19	PAD_TRX[19]	I/O	Analog I/O of Capacitive Touch Sensor.
26	20	PAD_TRX[18]	I/O	Analog I/O of Capacitive Touch Sensor.
27	21	PAD_TRX[17]	I/O	Analog I/O of Capacitive Touch Sensor.
28	22	PAD_TRX[16]	I/O	Analog I/O of Capacitive Touch Sensor.
29	23	PAD_TRX[15]	I/O	Analog I/O of Capacitive Touch Sensor.
30	24	PAD_TRX[14]	I/O	Analog I/O of Capacitive Touch Sensor.
31	25	PAD_TRX[13]	I/O	Analog I/O of Capacitive Touch Sensor.
32	26	PAD_TRX[12]	I/O	Analog I/O of Capacitive Touch Sensor.
33	27	PAD_TRX[11]	I/O	Analog I/O of Capacitive Touch Sensor.
34	28	PAD_TRX[10]	I/O	Analog I/O of Capacitive Touch Sensor.
35	29	PAD_TRX[9]	I/O	Analog I/O of Capacitive Touch Sensor.

36	**30**	**PAD_TRX[8]**	**I/O**	**Analog I/O of Capacitive Touch Sensor.**
37	**31**	**PAD_TRX[7]**	**I/O**	**Analog I/O of Capacitive Touch Sensor.**
38	**32**	**PAD_TRX[6]**	**I/O**	**Analog I/O of Capacitive Touch Sensor.**
39	**33**	**PAD_TRX[5]**	**I/O**	**Analog I/O of Capacitive Touch Sensor.**
40	**34**	**PAD_TRX[4]**	**I/O**	**Analog I/O of Capacitive Touch Sensor.**
41	**35**	**PAD_TRX[3]**	**I/O**	**Analog I/O of Capacitive Touch Sensor.**
42	**36**	**PAD_TRX[2]**	**I/O**	**Analog I/O of Capacitive Touch Sensor.**
43	**-**	**PAD_TRX[1]**	**I/O**	**Analog I/O of Capacitive Touch Sensor.**
44	**-**	**PAD_TRX[0]**	**I/O**	**Analog I/O of Capacitive Touch Sensor.**
45	**37**	**VCM_33V**	**P**	**Common Voltage Reference :** It needs a decoupling capacitor (0.1uF) to AGND33
46	**38**	**AGND33**	**P**	**Power Supply :** Analog ground voltage source.
47	**39**	**AVDD33**	**P**	**Power Supply :** Analog power supply voltage source. It generally needs two decoupling capacitor to AGND33. One is 1uF, Another is 0.01uF ~ 1uF.
48	**40**	**DGND**	**P**	**Power Supply :** Digital ground voltage source.

I : Input Only
O : Output Only
I/O : Input and Output
P : Power or Ground

2.3 Features Selection Table

The following Table shows supported functionalities for different IC model.

Table 2-2 Feature Selection Table

Flash IC	PKG.	I2C	Reset	INT	Max Sensor NO.
ZET6221WNA	QFN48	1	1	1	36 (TX:TY=20:16)
ZET6221WMA	QFN40	1	1	1	28 (TX:TY=18:10)

2.4 Hardware Architecture

2.4.1 Schematic Description

Figure 2-2 descripts the Zet6221 QFN48 IC reference circuit. It include the Power Pin (AVDD33、AGND33、DVDD33、DGND、VDDIO 、DVDD18、VCM_33V) , I2C interface and sensing line (PAD_TY[0]~PAD_TY[19], PAD_TX[0]~PAD_TX[19]). It also defines the external element which needs for implement.

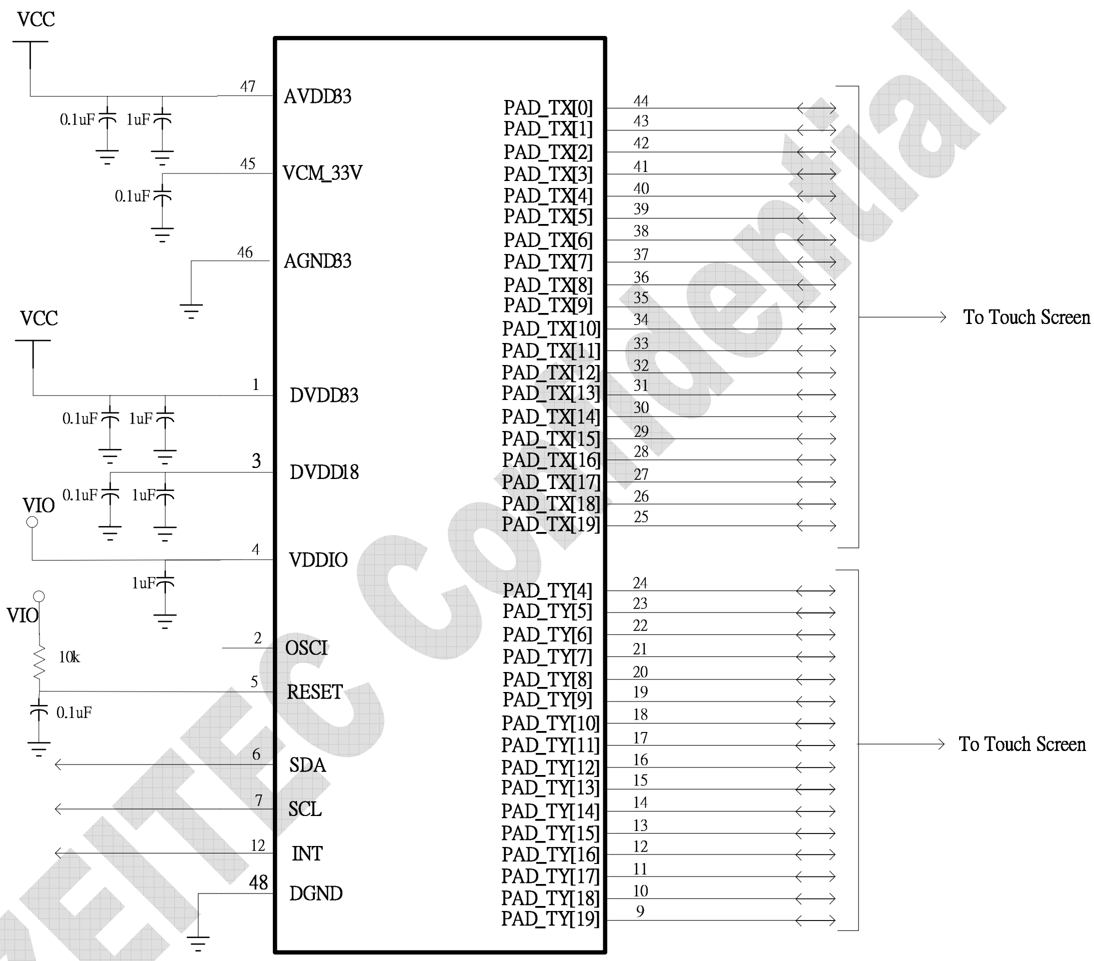


Figure 2-2 Zet6221 QFN48 Schematic

2.4.2 I/O Voltage Selection

In the Figure 2-2, the interface voltage (VIO) and operation voltage (VCC) is separated for different I/O voltage application. If the system needs the lower interface voltage than the operation voltage, it needs dual power supply from main board. One is for operation voltage and the other one is for interface voltage. Be carefully, the operation voltage must large than the interface voltage. It is showed in the Figure 2-3. The operation voltage is 3.3V and the interface voltage is 1.8V. So the interface I/O (SDA,SCL,RESET,INT) will work in 1.8V.

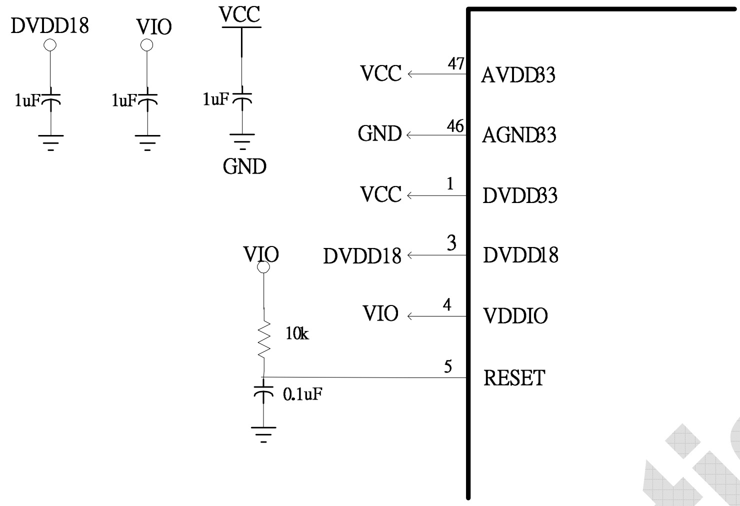


Figure 2-3 The Reference Schematic of 1.8V I/O

If the operation voltage and the interface voltage is the same, it just need only power supply. In this application, it can be shorted the interface voltage (VIO) and operation voltage (VCC). It is showed in the Figure 2-4. So the interface I/O (SDA,SCL,RESET,INT) will work in VCC.

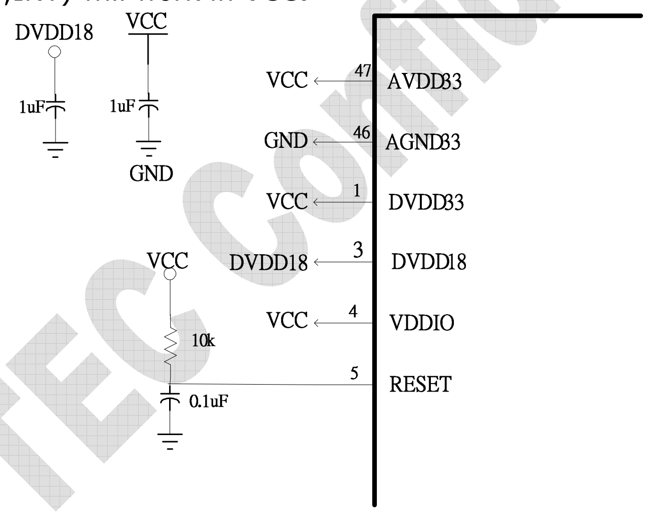


Figure 2-4 The Reference Schematic of Single Power

3 TOUCH SCREEN LAYOUT

3.1 Resolution / DPI / Trace Number

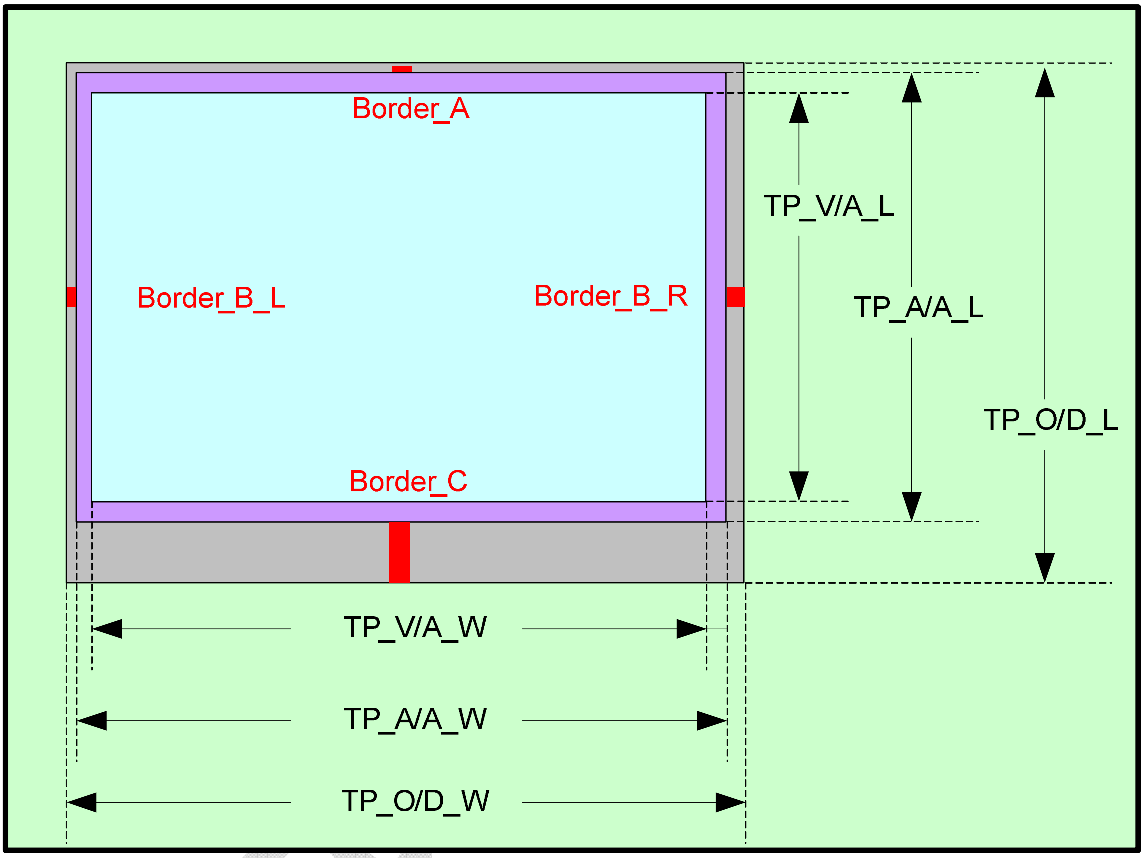


Figure 3-1 Touch Screen Outline Dimension

Table 3-1 Touch Screen Outline Define

AR	W : L
Sensor Size	S – inch
TP_O/D(Out Dimension)	W_OD : L_OD
TP_AA(Action Area)	W_AA : L_AA
TP_VA(View Area)	W_VA : L_VA
Border_A	W_Border_A
Border_B	W_Border_B
Border_C	W_Border_C
X_TraceNum	Num_X
Y_TraceNum	Num_Y
X Sensor Width	Wx
Y Sensor Width	Wy
ITO Layout	ITO Vendor Rule

3.2 ITO Sensor Cell

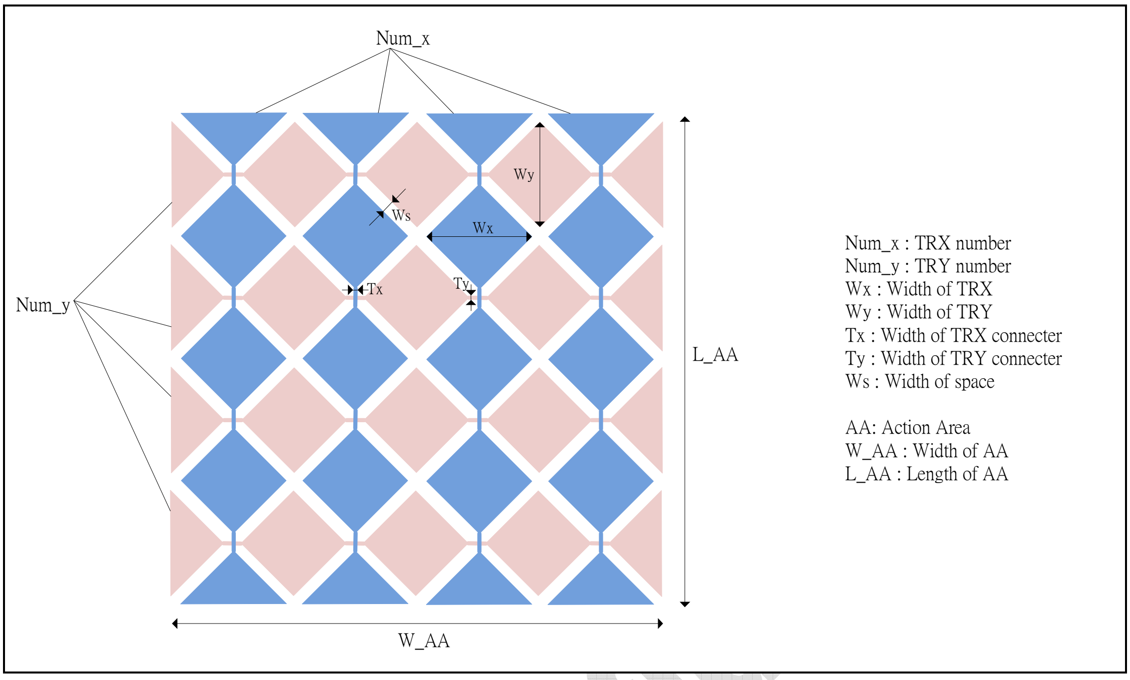


Figure 3-2 ITO Cell Paramenter Define

3.2.1 Design Rule

(a) Design Reference Data

- LCM_AA = TP_VA => W_VA : L_VA
- FPCA Location
- TP_OD => W_OD : L_OD

(b) Sensor Design Rule

- Minimum Width between TP_AA and TP_VA = 1mm
- Trace Number :
  X-trace Number = Round(W_AA / Wx)
  Y-trace Number = Round(L_AA / Wy)
- Minimum Width between TP_OD and TP_AA in Border_A
  0.7mm (for Sensor OD cutting and one shielding line)
- Minimum Width between TP_OD and TP_AA in Border_B
  0.7mm (for Sensor OD cutting and one shielding line) + Trace Pitch * RoundUp (Y_TraceNum / 2) (for trace metal line)
- Minimum Width between TP_OD and TP_AA in Border_C
  0.7mm (for Sensor OD cutting and one shielding line) + Trace Pitch * RoundUp (Y_TraceNum + X_TraceNum) (for trace metal line)+ 1.6mm (for ACF area)
- Resistance between n-Trace and n+1 Trace must be the same.

(c) DPI && Resolution

- Resolution = (number of trace – 1) * 64
- DPI = Resolution / Inch

(d) TP_A/A_W (mm) = (Num_x * Wx) + (sqrt(2) * Ws * (Num_x – 1))

(e) TP_A/A_L (mm) = (Num_y * Wy) + (sqrt(2) * Ws * (Num_y – 1))

(f) (Wx + Ws) <6mm ,

for linearity (moving speed=20mm/s, 8ϕ) ±1mm @ center area
±2mm @ edge area (6mm)

3.2.2 Design Consideration

- Trace numbers depend on panel area and dots per inch (DPI).
- The sensitivity depends on the thickness of the cover lens. When the thickness is larger, the sensitivity will be smaller.
- To avoid pulling the big metallicity, for example, the metal screw or iron plat , beside the sensor trace.
- The sensor trace of X-axis and Y-axis can be any shape but it is better to be square, hexagon or octagon, as well as try to use all sensor area as many as possible.
- The connected line of X-axis and Y-axis sensor in the border region should be isolated by shielding ground.
- The shape and size of the same axis sensor cells should be the same between different traces.
- It is better that decreasing the total resistance of the sensor trace and the inaccuracy between the two neighbouring sensor trace should be less than 10% or 500Ω.
- It is better that decreasing the total capacitance of the sensor trace and the inaccuracy between the two neighbouring sensor trace should be less than 10% or 3pF.
- The product of total resistance and the total capacitance should be less than 1e-6 (R*C<1e-6).
- The total resistance of single trace (from IC to TP) should be less than 20kΩ.
- The total capacitance of single trace (from IC to TP) should be less than 60pF.
- The isolate resistance between sensor traces should be bigger than 10MΩ.

3.3 Touch Screen Design example

3.3.1 ZEITEC Normal Mode Touch Solution

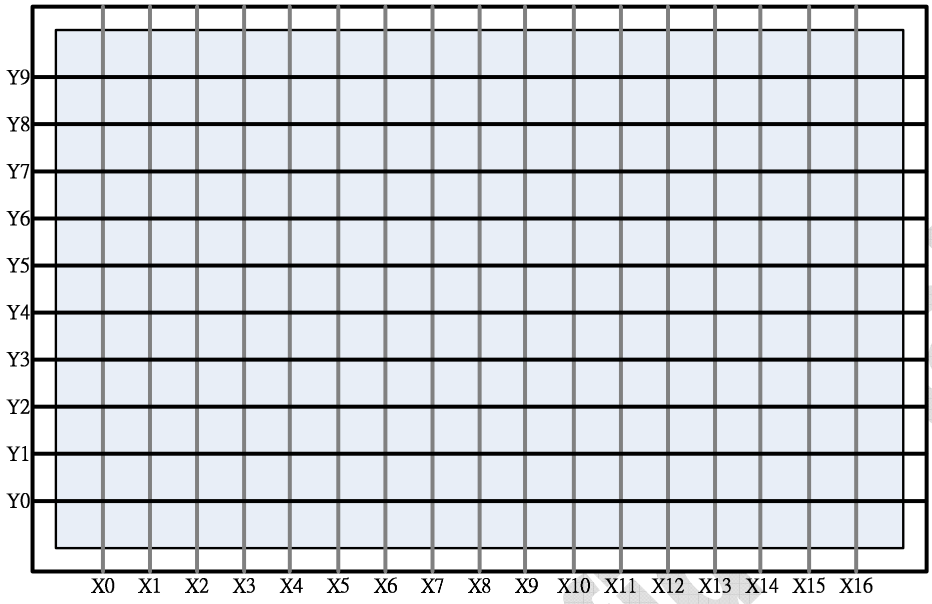


Figure 3-3 Size 17,10 Touch Screen ,Origin at (X0,Y0)

Figure 3-3 shows a touch screen, X trace number=17 and Y trace number=10. The total trace number is 27 and ZEITEC Touch solution IC package QFN40 (TX:TY=18:10) is the right choice. The connection of sensor trace and IC pin are as those in Table 3-2.And set the driving trace as Y0~Y9 and the sensing trace as X0~X16.

Table 3-2 Size 17,10 Touch Screen Design Table

Touchscreen	Zet6221 QFN40	Touchscreen	Zet6221 QFN40
Ground Ring	GND	X14	PAD_TRX[17]
X0	PAD_TRX[3]	X15	PAD_TRX[18]
X1	PAD_TRX[4]	X16	PAD_TRX[19]
X2	PAD_TRX[5]	Shielding Ground	GND
X3	PAD_TRX[6]	Y0	PAD_TRY[10]
X4	PAD_TRX[7]	Y1	PAD_TRY[11]
X5	PAD_TRX[8]	Y2	PAD_TRY[12]
X6	PAD_TRX[9]	Y3	PAD_TRY[13]
X7	PAD_TRX[10]	Y4	PAD_TRY[14]
X8	PAD_TRX[11]	Y5	PAD_TRY[15]
X9	PAD_TRX[12]	Y6	PAD_TRY[16]
X10	PAD_TRX[13]	Y7	PAD_TRY[17]
X11	PAD_TRX[14]	Y8	PAD_TRY[18]
X12	PAD_TRX[15]	Y9	PAD_TRY[19]
X13	PAD_TRX[16]	Ground Ring	GND

4 INTERFACE AND COMMUNICATION TIMING

4.1 Pin Description

Table 4-1 Pin Description

Pin	Symbol	I/O	Description
1	VCC	PWR	Power supply voltage source.
2	SDA	I/O	For Inter-IC interface (I2C) mode. Data input as slave device and output as master device.
3	SCL	I/O	For Inter-IC interface (I2C) mode. Clock input as slave device and output as master device.
4	GND	PWR	Ground voltage source.
5	Reset	I	Reset ZET6221, active low
6	/INT	O	Interrupt output signal.

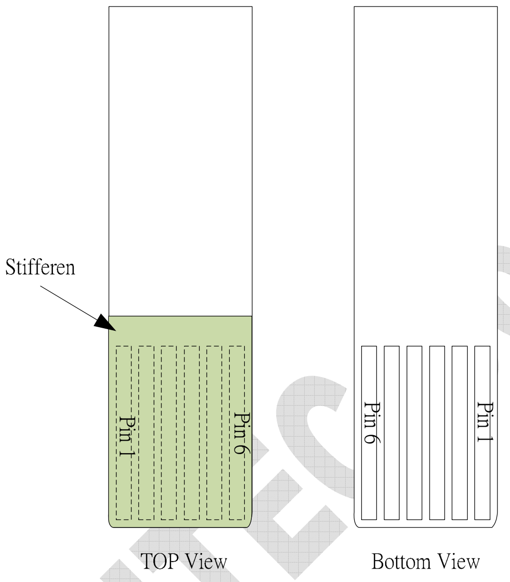


Figure 4-1 I2C Pin Define

4.2 I2C Slave Mode

For I2C slave mode selection, the SDA and SCL signal lines have to be pulled-high with R resistor (See Figure 4-2). The internal resistor of SDA and SCL is 4.7KΩ±20% and it can be enable or disable by Zeki (default enable).The TP is processing data on the I2C bus, the Touch Panel controller of TP uses the 7-bit addressing Mode and supports the bit transfer rate both of Standard-mode (up to 100 kbps) and Fast-mode (up to 400 kbps).

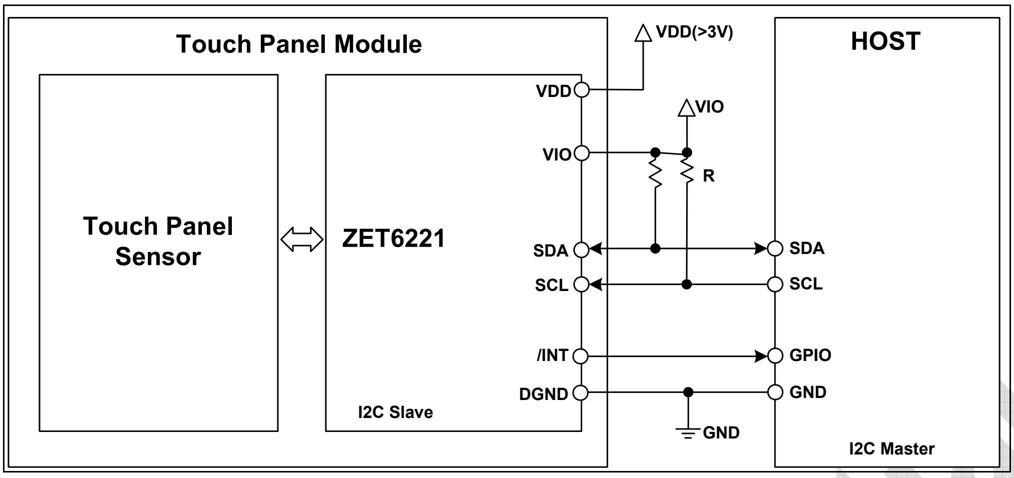


Figure 4-2 Connection Between Touch Panel and Host in I2C Slave Mode

4.3 I2C Slave Mode Communication Address

The Touch Panel controller is defined as an I2C slave device and the Host is defined as a master. The device address of the Touch Panel controller is designed as 7-bit address mode. The Touch Panel controller address is defined as 0x76 and LSB of I2C address register is Read / Write bit as shown below.

Table 4-2 I2C Address

	MSB						LSB
I2C Address	1	1	1	0	1	1	0	R/W

The first seven bits of I2C address register is device address of TP and the 8th bit means that Host wants to read data from Slave device or write data to Slave device (R(Read) = 1 and W (Write) = 0).

4.4 I2C Slave Mode Communication Dynamic Format

Device transmit X/Y coordinate to host and the packet ID is 0x3C

Protocol: When Host discovers /INT pull low, Host sends clock to receive information, the duration between byte and byte is at least 30us. The Coordinate TX Mode is Dynamic format, according to different Finger Application; the Coordinate TX Mode can support different Transmission data length.

- One Finger Application : 7 bytes
  1 Packet ID Byte, 2 Valid Bytes, 4 Finger1 Coordinate bytes
- One Finger + Key Application : 8 bytes
  1 Packet ID Byte, 2 Valid Bytes, 4 Finger1 Coordinate bytes, 1 Key Byte
- Two Fingers Application : 11 bytes
  1 Packet ID Byte, 2 Valid Bytes, 4 Finger1 Coordinate bytes, 4 Finger2

Coordinate bytes

- Two Fingers + key Application : 12 bytes
  1 Packet ID Byte, 2 Valid Bytes, 4 Finger1 Coordinate bytes, 4 Finger2 Coordinate bytes, 1 Key Byte

- Three Fingers Application : 15 bytes
  1 Packet ID Byte, 2 Valid Bytes, 4 Finger1 Coordinate bytes, 4 Finger2 Coordinate bytes, 4 Finger3 Coordinate bytes

- Three Fingers + Key Application : 16 bytes
  1 Packet ID Byte, 2 Valid Bytes, 4 Finger1 Coordinate bytes, 4 Finger2 Coordinate bytes, 4 Finger3 Coordinate bytes, 1 Key Byte

- Four Fingers Application : 19 bytes
  1 Packet ID Byte, 2 Valid Bytes, 4 Finger1 Coordinate bytes, 4 Finger2 Coordinate bytes, 4 Finger3 Coordinate bytes, 4 Finger4 Coordinate bytes

- Four Fingers + Key Application : 20 bytes
  1 Packet ID Byte, 2 Valid Bytes, 4 Finger1 Coordinate bytes, 4 Finger2 Coordinate bytes, 4 Finger3 Coordinate bytes, 4 Finger4 Coordinate bytes, 1 Key Byte

- Five Fingers Application : 23 bytes
  1 Packet ID Byte, 2 Valid Bytes, 4 Finger1 Coordinate bytes, 4 Finger2 Coordinate bytes, 4 Finger3 Coordinate bytes, 4 Finger4 Coordinate bytes, 4 Finger5 Coordinate bytes

- Five Fingers + Key Application : 24 bytes
  1 Packet ID Byte, 2 Valid Bytes, 4 Finger1 Coordinate bytes, 4 Finger2 Coordinate bytes, 4 Finger3 Coordinate bytes, 4 Finger4 Coordinate bytes, 4 Finger5 Coordinate bytes,1 Key Byte

Every Finger has four-byte information, 1^{st} byte ~ 3^{rd} byte are the X/Y coordinate, Last byte is the Z-Domain information that means the Touch Press Strength.

If Key Function Enable, the Key Byte will be placed in the last byte of Coordinate Transmission data, Max supported Key Number is 8 keys.

Figure 4-3 ~ Figure 4-9 shows the TX Data Timing of Five Fingers + Key Application.

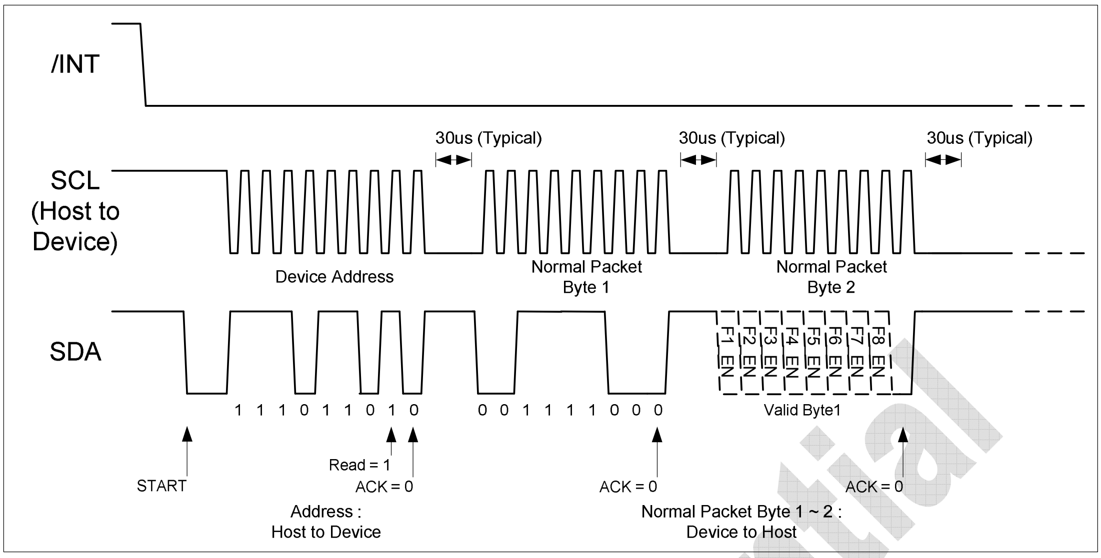


Figure 4-3 Packet – Byte 1 ~ Byte 2

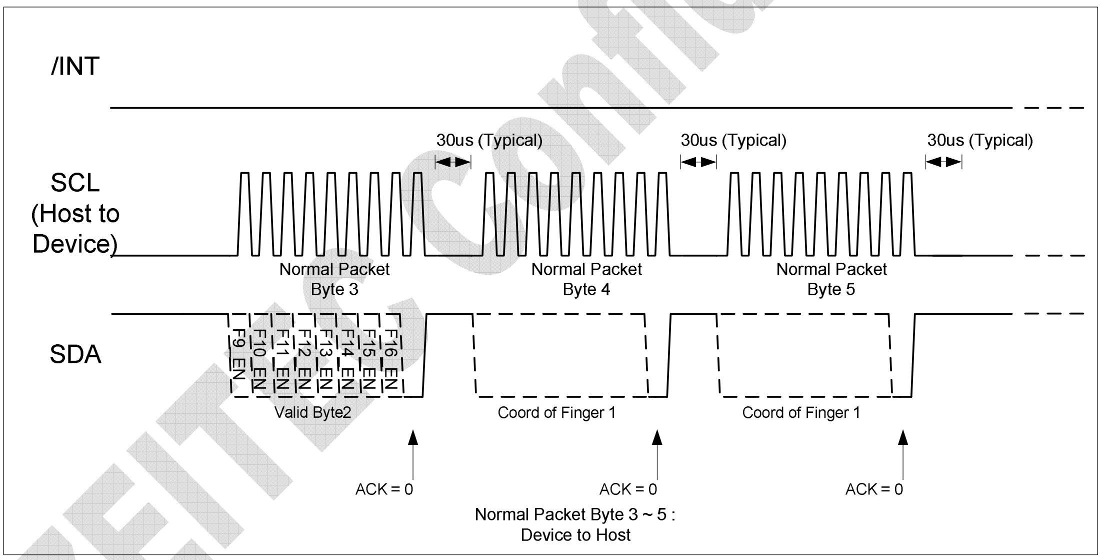


Figure 4-4 Packet – Byte 3 ~ Byte 5

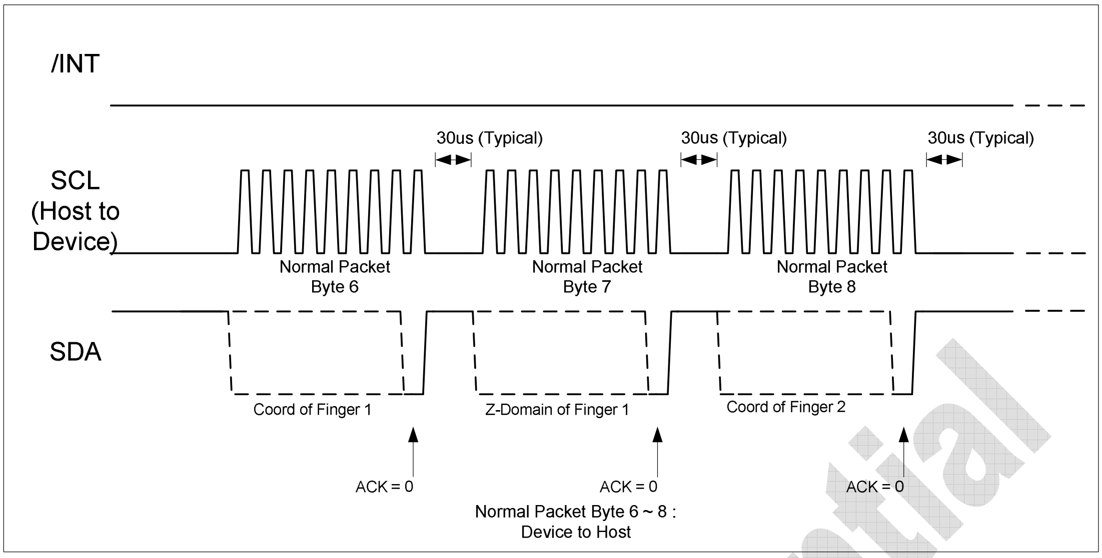


Figure 4-5 Packet – Byte 6 ~ Byte 8

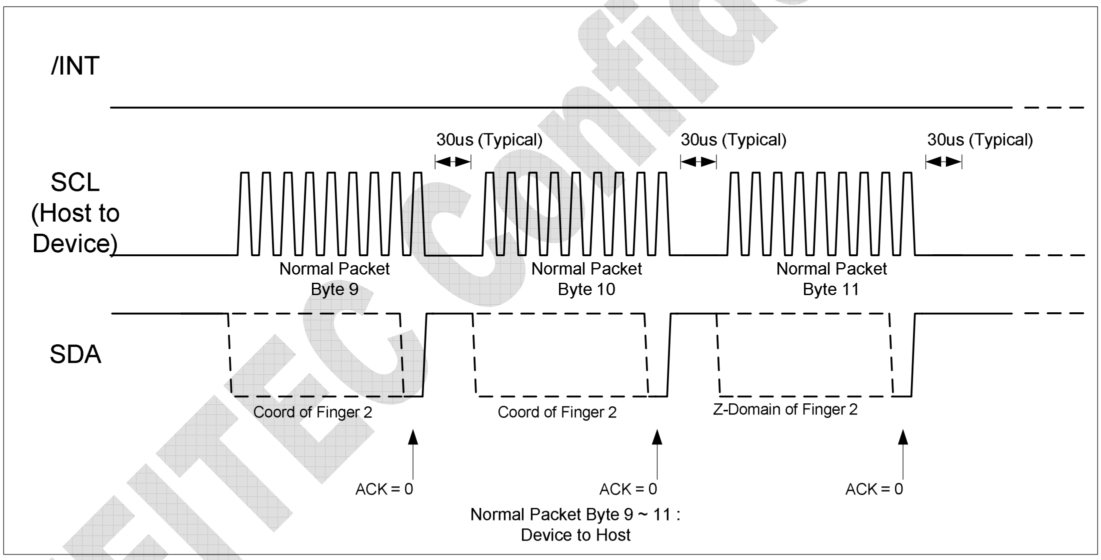


Figure 4-6 Packet – Byte 9 ~ Byte 11

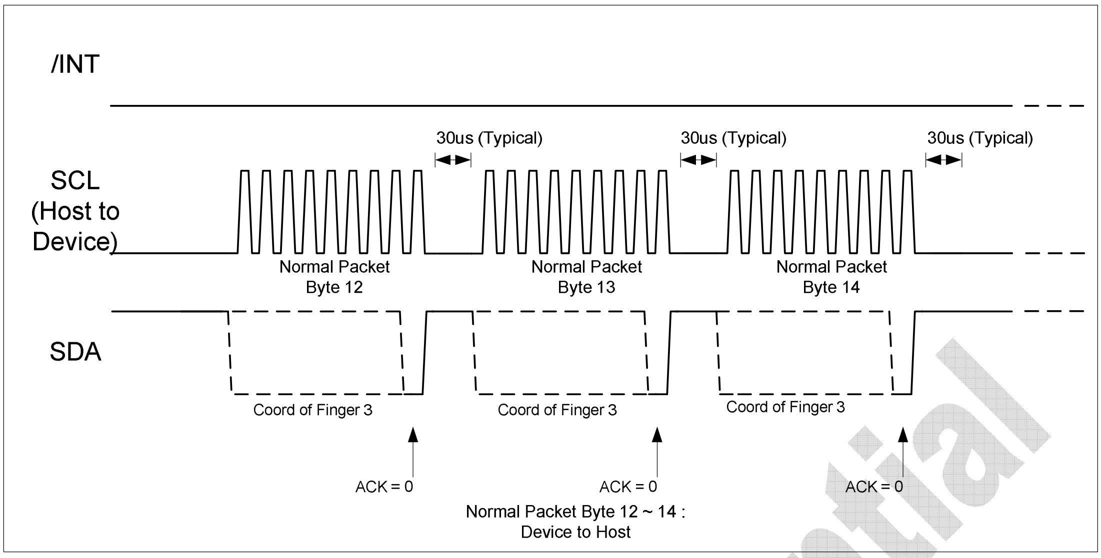


Figure 4-7 Packet – Byte 12 ~ Byte 14

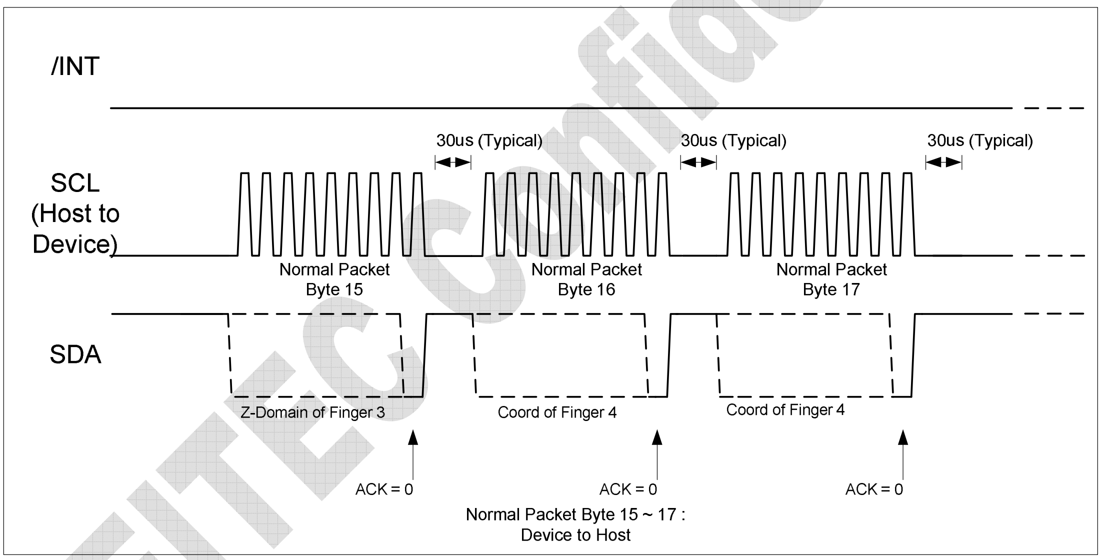


Figure 4-8 Packet – Byte 15 ~ Byte 17

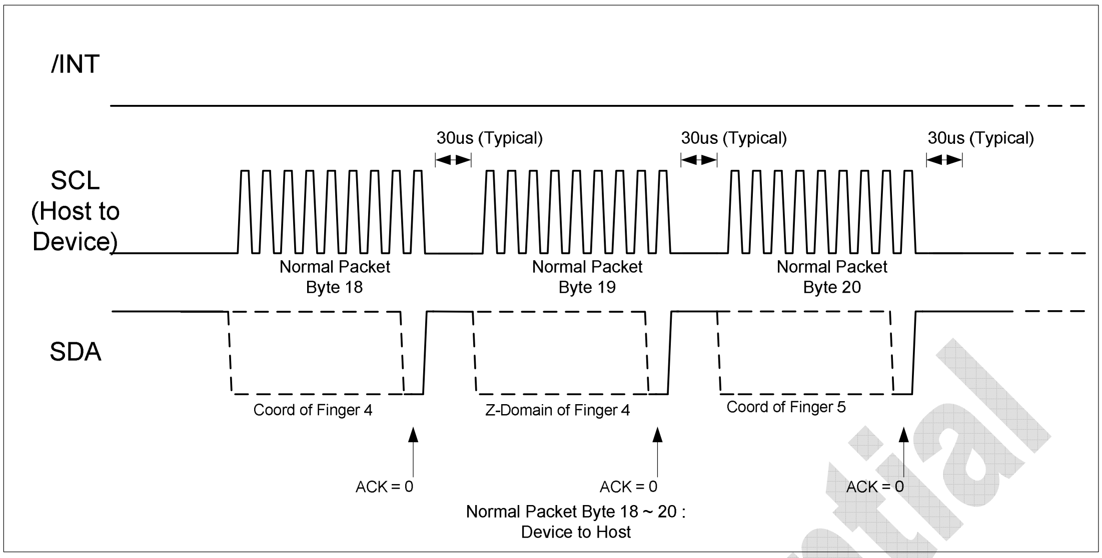


Figure 4-9 Packet – Byte 18 ~ Byte 20

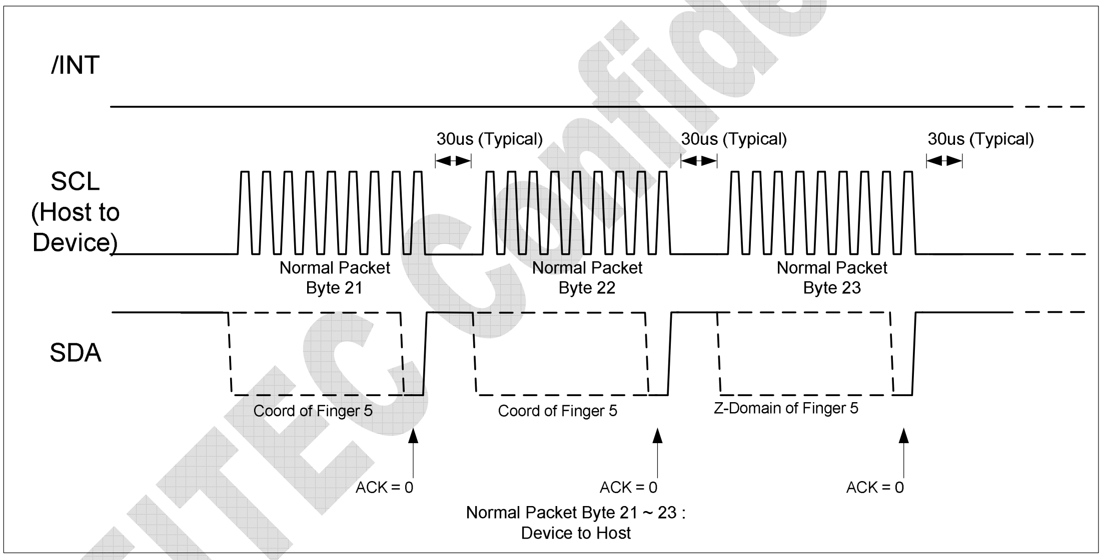


Figure 4-10 Packet – Byte 21 ~ Byte 23

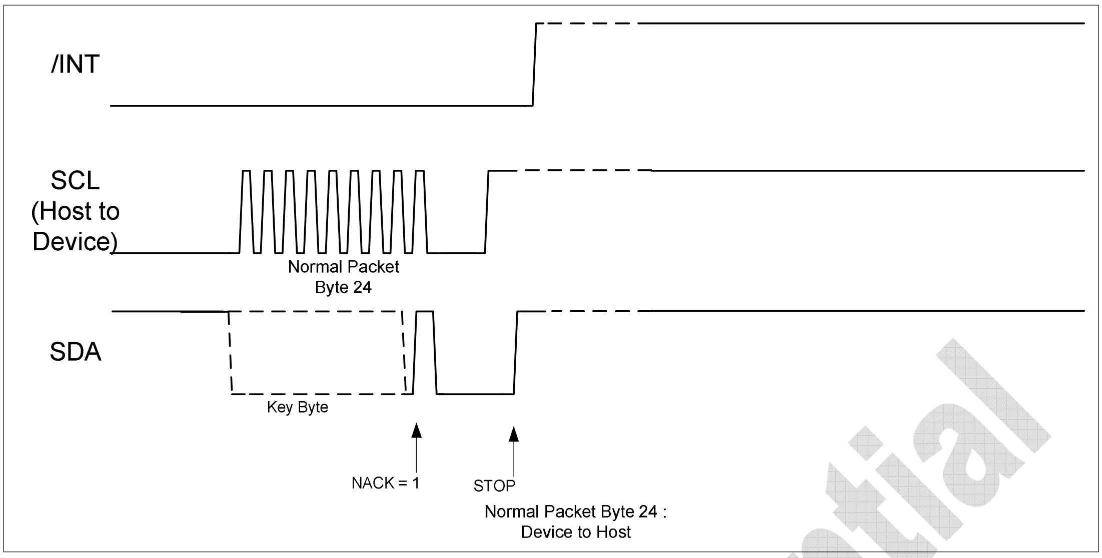


Figure 4-11 Packet – Byte 24

4.4.1 Dynamic Coordinate TX Normal Packet (Touch Panel to Host)

Table 4-3 24 Bytes Data Format (Five Fingers + Key Application)

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Byte 1	0	0	1	1	1	1	0	0
Byte 2	F1_EN	F2_EN	F3_EN	F4_EN	F5_EN	F6_EN	F7_EN	F8_EN
Byte 3	F9_EN	F10_EN	F11_EN	F12_EN	F13_EN	F14_EN	F15_EN	F16_EN
Byte 4	F1_X11	F1_X10	F1_X09	F1_X08	F1_Y11	F1_Y10	F1_Y09	F1_Y08
Byte 5	F1_X07	F1_X06	F1_X05	F1_X04	F1_X03	F1_X02	F1_X01	F1_X00
Byte 6	F1_Y07	F1_Y06	F1_Y05	F1_Y04	F1_Y03	F1_Y02	F1_Y01	F1_Y00
Byte 7	0	0	0	F1_Z04	F1_Z03	F1_Z02	F1_Z01	F1_Z00
Byte 8	F2_X11	F2_X10	F2_X09	F2_X08	F2_Y11	F2_Y10	F2_Y09	F2_Y08
Byte 9	F2_X07	F2_X06	F2_X05	F2_X04	F2_X03	F2_X02	F2_X01	F2_X00
Byte 10	F2_Y07	F2_Y06	F2_Y05	F2_Y04	F2_Y03	F2_Y02	F2_Y01	F2_Y00
Byte 11	0	0	0	F2_Z04	F2_Z03	F2_Z02	F2_Z01	F2_Z00
Byte 12	F3_X11	F3_X10	F3_X09	F3_X08	F3_Y11	F3_Y10	F3_Y09	F3_Y08
Byte 13	F3_X07	F3_X06	F3_X05	F3_X04	F3_X03	F3_X02	F3_X01	F3_X00
Byte 14	F3_Y07	F3_Y06	F3_Y05	F3_Y04	F3_Y03	F3_Y02	F3_Y01	F3_Y00
Byte 15	0	0	0	F3_Z04	F3_Z03	F3_Z02	F3_Z01	F3_Z00
Byte 16	F4_X11	F4_X10	F4_X09	F4_X08	F4_Y11	F4_Y10	F4_Y09	F4_Y08
Byte 17	F4_X07	F4_X06	F4_X05	F4_X04	F4_X03	F4_X02	F4_X01	F4_X00
Byte 18	F4_Y07	F4_Y06	F4_Y05	F4_Y04	F4_Y03	F4_Y02	F4_Y01	F4_Y00
Byte 19	0	0	0	F4_Z04	F4_Z03	F4_Z02	F4_Z01	F4_Z00
Byte 20	F5_X11	F5_X10	F5_X09	F5_X08	F5_Y11	F5_Y10	F5_Y09	F5_Y08
Byte 21	F5_X07	F5_X06	F5_X05	F5_X04	F5_X03	F5_X02	F5_X01	F5_X00
Byte 22	F5_Y07	F5_Y06	F5_Y05	F4_Y04	F5_Y03	F5_Y02	F5_Y01	F5_Y00
Byte 23	0	0	0	F5_Z04	F5_Z03	F5_Z02	F5_Z01	F5_Z00
Byte 24	K8_EN	K7_EN	K6_EN	K5_EN	K4_EN	K3_EN	K2_EN	K1_EN

Data Format of X/Y Coordinate

Normal Packet: 24 Bytes

- Packet ID: 0x3C (**Byte 1**)

- Valid Byte (**Byte2 and Byte3**)

(a) FX_EN = 1 : Coordinate of Finger X is meaningful.
(b) Total 16-Finger Valid bits can be supported in the two bytes, if the valid bit of Finger X is high, Host can catch the Coordinate of Finger X from the coordinate bytes of Finger X.

Table 4-4 Byte2 and Byte3

MSB							LSB
F1_EN	F2_EN	F3_EN	F4_EN	F5_EN	F6_EN	F7_EN	F8_EN
MSB							LSB
F9_EN	F10_EN	F11_EN	F12_EN	F13_EN	F14_EN	F15_EN	F16_EN

- X-axis Coordinate of Finger 1 (**MSB 4 bits of Byte4 and 8 bits of Byte5**):
  (a) Total 12 Bits format
  (b) X1-axis Coordinate Format

Table 4-5 MSB 4 Bits of Byte4 and 8 Bits of Byte5

MSB											LSB
F1 X11	F1 X10	F1 X09	F1 X08	F1 X07	F1 X06	F1 X05	F1 X04	F1 X03	F1 X02	F1 X01	F1 X00

- Y-axis Coordinate of Finger 1 (**LSB 4 bits of Byte4 and 8 bits of Byte6**):
  (a) Total 12 Bits format
  (b) Y1-axis Coordinate Format

Table 4-6 LSB 4 Bits of Byte4 and 8 Bits of Byte6

MSB											LSB
F1 Y11	F1 Y10	F1 Y09	F1 Y08	F1 Y07	F1 Y06	F1 Y05	F1 Y04	F1 Y03	F1 Y02	F1 Y01	F1 Y00

- X-axis Coordinate of Finger 2 (**MSB 4 bits of Byte8 and 8 bits of Byte9**):
  (a) Total 12 Bits format
  (b) X2-axis Coordinate Format

Table 4-7 MSB 4 Bits of Byte8 and 8 Bits of Byte9

MSB											LSB
F2 X11	F2 X10	F2 X09	F2 X08	F2 X07	F2 X06	F2 X05	F2 X04	F2 X03	F2 X02	F2 X01	F2 X00

- Y-axis Coordinate of Finger 2 (**LSB 4 bits of Byte8 and 8 bits of Byte10**):
  (a) Total 12 Bits format
  (b) Y2-axis Coordinate Format

Table 4-8 LSB 4 bits of Byte8 and 8 Bits of Byte10

MSB											LSB
F2 Y11	F2 Y10	F2 Y09	F2 Y08	F2 Y07	F2 Y06	F2 Y05	F2 Y04	F2 Y03	F2 Y02	F2 Y01	F2 Y00

- X-axis Coordinate of Finger 3 (**MSB 4 bits of Byte12 and 8 bits of Byte13**):
  (a) Total 12 Bits format
  (b) X3-axis Coordinate Format

Table 4-9 MSB 4 Bits of Byte12 and 8 Bits of Byte13

MSB											LSB
F3 X11	F3 X10	F3 X09	F3 X08	F3 X07	F3 X06	F3 X05	F3 X04	F3 X03	F3 X02	F3 X01	F3 X00

- Y-axis Coordinate of Finger 3 (**LSB 4 bits of Byte12 and 8 bits of Byte14**):
  (a) Total 12 Bits format
  (b) Y3-axis Coordinate Format

Table 4-10 LSB 4 Bits of Byte12 and 8 Bits of Byte14

MSB											LSB
F3 Y11	F3 Y10	F3 Y09	F3 Y08	F3 Y07	F3 Y06	F3 Y05	F3 Y04	F3 Y03	F3 Y02	F3 Y01	F3 Y00

- X-axis Coordinate of Finger 4 (**MSB 4 bits of Byte16 and 8 bits of Byte17**):
  (a) Total 12 Bits format
  (b) X4-axis Coordinate Format

Table 4-11 MSB 4 Bits of Byte16 and 8 Bits of Byte17

MSB											LSB
F4 X11	F4 X10	F4 X09	F4 X08	F4 X07	F4 X06	F4 X05	F4 X04	F4 X03	F4 X02	F4 X01	F4 X00

- Y-axis Coordinate of Finger 4 (**LSB 4 bits of Byte16 and 8 bits of Byte18**):
  (a) Total 12 Bits format
  (b) Y4-axis Coordinate Format

Table 4-12 LSB 4 Bits of Byte16 and 8 Bits of Byte18

MSB											LSB
F4 Y11	F4 Y10	F4 Y09	F4 Y08	F4 Y07	F4 Y06	F4 Y05	F4 Y04	F4 Y03	F4 Y02	F4 Y01	F4 Y00

- X-axis Coordinate of Finger 5 (**MSB 4 bits of Byte20 and 8 bits of Byte21**):
  (a) Total 12 Bits format
  (b) X5-axis Coordinate Format

Table 4-13 MSB 4 Bits of Byte20 and 8 Bits of Byte21

MSB											LSB
F5 X11	F5 X10	F5 X09	F5 X08	F5 X07	F5 X06	F5 X05	F5 X04	F5 X03	F5 X02	F5 X01	F5 X00

- Y-axis Coordinate of Finger 5 (**LSB 4 bits of Byte20 and 8 bits of Byte22**):
  (a) Total 12 Bits format
  (b) Y5-axis Coordinate Format

Table 4-14 LSB 4 Bits of Byte20 and 8 Bits of Byte22

MSB											LSB
F5 Y11	F5 Y10	F5 Y09	F5 Y08	F5 Y07	F5 Y06	F5 Y05	F5 Y04	F5 Y03	F5 Y02	F5 Y01	F5 Y00

- Key Byte (**Byte24 of TX Data**):
  (a) Total 8 Bits format
  (b) KX_EN = 1 : The Key X is active.

Table 4-15 Byte 24 of TX Data

MSB							LSB
K1_EN	K7_EN	K6_EN	K5_EN	K4_EN	K3_EN	K2_EN	K1_EN

- Z-Domain Byte (**Byte 7/11/15/19/23)**
  (a) Total 5 Bits format
  (b) There are 32-Step information to represent the Press Strength of Finger X, if touch panel with Heavy Press, the value of Z-Domain will be larger.

Table 4-16 Byte 7/11/15/19 of TX Data

MSB							LSB
0	0	0	FX_Z04	FX_Z03	FX_Z02	FX_Z01	FX_Z00

4.4.2 Reference C Program

```
BYTE InBuff [1024];

//receive coordinate report
void receiveData ( int FingerNum , int KeyNum , unsigned long x[] ,
unsigned long y[] , unsigned long z[] , u16  &fingers ,u8 &keys ) {
    int bufLength;

    if(KeyNum=0)
        bufLength  = 3+4*FingerNum;
    else
        bufLength  = 3+4*FingerNum+1;

    i2c_Read(InBuff, bufLength);

    if (InBuff[0] == 0x3c)
    {
        fingers = 0x0;
        // lnBuff[1]~InBuff[2] = finger status
        fingers = fingers | (InBuff[1] << 8) | InBuff[2];

        for(int i=0;i<FingerNum;i++)
        {
            x[i]=(BYTE)((InBuff[3+4*i])>>4)*256                    +
(BYTE)InBuff[(3+4*i)+1];
            y[i]=(BYTE)((InBuff[3+4*i])      &       0x0f)*256       +
(BYTE)InBuff[(3+4*i)+2];
            z[i]=(BYTE)((InBuff[(3+4*i)+3]) & 0x0f);
        }

        //if key enable
        if(KeyNum > 0)
            keys = InBuff[3+4*FingerNum];
    }
}
```

4.5 TP Command

Table 4-17 Supported TP Command Type

TP Command	Instruction	Byte Number	Description
Reset Mode	0xB0	1	Software Reset TP
Deep Sleep Mode	0xB1	1	Let TP enter Deep Sleep Mode
Dynamic Coordinate TX Mode	0xB2	1	After Host send this command, TP will give 17 bytes response that has Finger Application information
Idle Mode	0xB3	1	Let TP enter Idle Mode
Wake-Up Mode	0xB4	1	Wake TP Up from Deep Sleep Mode
Enable Charger Mode	0xB5	1	Let TP enter Charger Mode for Charger Noise Immunity
Disable Charger Mode	0xB6	1	Let TP disable from Charger Mode
Boot Loader Mode	0xB9	1	TP Enter Writer Mode

- Reset Mode:
  Host can send the Reset Command (0xB0) to Reset TP.

- Deep Sleep Mode :
  Host can send the Deep Sleep Command (0xB1) to reduce power consumption, when TP enter Deep Sleep Mode, the Power Consumption will be about 35uA.TP can be waked up by sending Wake-Up Command (0xB4).

- Dynamic Coordinate TX Mode :
  When TP power on, the host needs to send the Dynamic Coordinate TX Command (0xB2) to get the Finger Application information, the information shows supported finger number, resolution, data format and traces number etc . After Host sent the command, TP will provide 17 bytes Finger Application response to Host.

  After TP receives the command, TP will output the 17-Byte Finger Application information to Host device, content of the bytes are
  1. Project Code(8 Bytes) => 1st byte ~ 8th byte
  2. Resolution X(2 Bytes): Low Byte : 9th byte, High Byte : 10th byte => 9th byte ~ 10th byte
  3. Resolution Y(2 Bytes): Low Byte : 11th byte, High Byte : 12th byte => 11th byte ~ 12th byte
  4. Data Format(1 Byte): 0x40: Coordinate => 13th byte
  5. X Trace Num(1 Byte): Trace Number of X-axis => 14th byte
  6. Y Trace Num(1 Byte): Trace Number of Y-axis => 15th byte
  7. Dynamic Finger(1 Byte): LSB 7-Bit : Finger Number, MSB : Key Enable(1'b1) / Key Disable(1'b0)
  8. Reserve Byte(1Byte)

- Idle Mode :
  Host can send the Idle Command (0xB3) to reduce power consumption, when TP enter Idle Mode, the Power Consumption will be about 140uA. Host can wake TP up by pull low INT Signal or send the Wake-Up Mode command.

- Wake-Up Mode :
  The Command (0xB4) will wake TP up from Deep Sleep Mode.

- Charger Mode :

Charger Noise will affect the TP performance. The Charger Mode Command (0xB5) will let TP get Charger Noise Immunity ability.

- Discharger Mode :
  Disable the Charger Mode.

- Boot Loader Mode :
  TP will enter Writer Mode after receiving the Command(0xB9), Driver can update Flash ROM, Fig.4-10 shows the Boot Loader Flow.

4.5.1 Reference C Program

```
/*************************************************************************
    [CMD instruction] 0xB0: Reset Mode
    [function]:
               callback: Software Reset TP;
    [parameters]:
             client[in]:   struct  i2c_client  — represent  an  I2C  slave
device;

    [return]:
              1;
*************************************************************************/
u8 zet6221_ts_reset_mode(struct i2c_client *client)
{
   u8 ts_reset_cmd[1] = {0xB0};

   int ret;

   ret=zet6221_i2c_write_tsdata(client, ts_reset_cmd, 1);

   return 1;
}

/*************************************************************************
    [CMD instruction] 0xB1: Deep Sleep Mode
    [function]:
               callback: Let TP enter Deep Sleep Mode;
    [parameters]:
             client[in]:   struct  i2c_client  — represent  an  I2C  slave
device;

    [return]:
              1;
*************************************************************************/
u8 zet6221_ts_reset_mode(struct i2c_client *client)
{
   u8 ts_sleep_cmd[1] = {0xB1};

   int ret;

   ret=zet6221_i2c_write_tsdata(client, ts_sleep_cmd, 1);

   return 1;
}

/*************************************************************************
    [CMD instruction] 0xB2: Dynamic Coordinate TX Mode
```

```
    [function]:
              callback: get dynamic report information;
    [parameters]:
             client[in]:   struct  i2c_client  — represent  an  I2C  slave
device;

    [return]:
              1;
*************************************************************************/
u8 zet6221_ts_get_report_mode(struct i2c_client *client)
{
   u8 ts_report_cmd[1] = {0xB2};
   u8 ts_reset_cmd[1] = {0xB0};
   u8 ts_in_data[17] = {0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0};

   u8 pc[8];
   int ResolutionX, ResolutionY, FingerNum, KeyNum;

   int ret;

   ret=zet6221_i2c_write_tsdata(client, ts_report_cmd, 1);

   if (ret > 0)
   {
      while(1)
      {
         udelay(1);

         if (gpio_get_value(S3C64XX_GPN(9)) == 0)
         {
            ret=zet6221_i2c_read_tsdata(client, ts_in_data, 17);

            for(i=0;i<8;i++)
            {
               pc[i]=ts_in_data[i] & 0xff;
            }
            ResolutionX = ts_in_data[9] & 0xff;
            ResolutionX = (ResolutionX << 8)|(ts_in_data[8] & 0xff);
            ResolutionY = ts_in_data[11] & 0xff;
            ResolutionY = (ResolutionY << 8) | (ts_in_data[10] & 0xff);
            FingerNum = (ts_in_data[15] & 0x7f);
            KeyNum = (ts_in_data[15] & 0x80);

            if(KeyNum==0)
               bufLength  = 3+4*FingerNum;
            else
               bufLength  = 3+4*FingerNum+1;

            break;

         }
      }

   }
   return 1;
}

/*************************************************************************
```

```
    [CMD instruction] 0xB3: Idle Mode
    [function]:
               callback: Let TP enter Idle Mode;
    [parameters]:
             client[in]:   struct  i2c_client  —  represent  an  I2C  slave
device;

    [return]:
              1;
*************************************************************************/
u8 zet6221_ts_idle_mode(struct i2c_client *client)
{
   u8 ts_idle_cmd[1] = {0xB3};

   int ret;

   ret=zet6221_i2c_write_tsdata(client, ts_idle_cmd, 1);

   return 1;
}

/*************************************************************************
    [CMD instruction] 0xB4: Wake-Up Mode
    [function]:
               callback: Wake TP Up from Deep Sleep Mode;
    [parameters]:
             client[in]:   struct  i2c_client  —  represent  an  I2C  slave
device;

    [return]:
              1;
*************************************************************************/
u8 zet6221_ts_wakeup_mode(struct i2c_client *client)
{
   u8 ts_wakeup_cmd[1] = {0xB4};

   int ret;

   ret=zet6221_i2c_write_tsdata(client, ts_wakeup_cmd, 1);

   return 1;
}

/*************************************************************************
    [CMD instruction] 0xB5: Enable Charger Mode
    [function]:
               callback: Let TP enter Charger Mode for Charger Noise Immunity;
    [parameters]:
             client[in]:   struct  i2c_client  —  represent  an  I2C  slave
device;

    [return]:
              1;
*************************************************************************/
u8 zet6221_ts_charger_mode(struct i2c_client *client)
{
   u8 ts_charger_cmd[1] = {0xB5};
```

```
    int ret;

    ret=zet6221_i2c_write_tsdata(client, ts_charger_cmd, 1);

    return 1;
}

/*************************************************************************
     [CMD instruction] 0xB6: Disable Charger Mode
     [function]:
                callback: Let TP disable  from Charger Mode;
     [parameters]:
             client[in]:   struct  i2c_client — represent  an  I2C  slave
device;

     [return]:
              1;
**************************************************************************/
u8 zet6221_ts_noncharger_mode(struct i2c_client *client)
{
    u8 ts_noncharger_cmd[1] = {0xB6};

    int ret;

    ret=zet6221_i2c_write_tsdata(client, ts_noncharger_cmd, 1);

    return 1;
}

/*************************************************************************
     [CMD instruction] 0xB9: Enter Writer Mode
     [function]:
                callback: To enter writer mode to update firmware;
     [parameters]:
             client[in]:   struct  i2c_client — represent  an  I2C  slave
device;

     [return]:
              1;
**************************************************************************/
u8 zet6221_ts_writer_mode(struct i2c_client *client)
{
    u8 ts_writer_cmd[1] = {0xB9};

    int ret;

    ret=zet6221_i2c_write_tsdata(client, ts_writer_cmd, 1);

    return 1;
}
```

4.6 Boot Loader Workflow

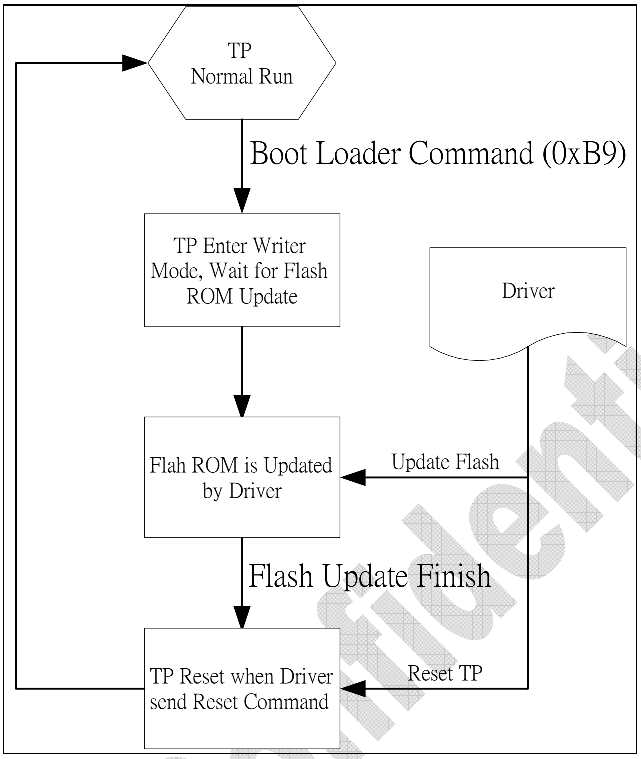


Figure 4-12 Boot Loader Flow

4.6.1 Reference C Program

```
#include "zet6221_fw.h"

static unsigned char zeitec_zet6221_page[130] __initdata;

u8 zet6221_ts_sndpwd(struct i2c_client *client)
{
    u8 ts_sndpwd_cmd[3] = {0x20,0xC5,0x9D};

    int ret;

    ret=zet6221_i2c_write_tsdata(client, ts_sndpwd_cmd, 3);

    return 1;
}

u8 zet6221_ts_sfr(struct i2c_client *client)
{
    u8 ts_cmd[1] = {0x2C};
    u8 ts_in_data[16] = {0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0};
    u8 ts_cmd17[17] = {0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0};
    int ret;
    int i;

    ret=zet6221_i2c_write_tsdata(client, ts_cmd, 1);
```

```
    ret=zet6221_i2c_read_tsdata(client, ts_in_data, 16);
    msleep(100);

    for(i=0;i<16;i++)
    {
        ts_cmd17[i+1]=ts_in_data[i];
    }

    if(ts_in_data[14]==0x3D)
    {
        ts_cmd[0]=0x2D;
        ret=zet6221_i2c_write_tsdata(client, ts_cmd, 1);
        msleep(500);

        ts_cmd17[15]=0x7D;
        ts_cmd17[0]=0x2E;
        ret=zet6221_i2c_write_tsdata(client, ts_cmd17, 17);

    }else
    {
        ts_cmd17[15]=0x3D;
        ts_cmd17[0]=0x2B;
        ret=zet6221_i2c_write_tsdata(client, ts_cmd17, 17);
    }

    return 1;
}

u8 zet6221_ts_masserase(struct i2c_client *client)
{
    u8 ts_cmd[1] = {0x24};

    int ret;

    ret=zet6221_i2c_write_tsdata(client, ts_cmd, 1);

    return 1;
}

u8 zet6221_ts_resetmcu(struct i2c_client *client)
{
    u8 ts_cmd[1] = {0x29};

    int ret;

    ret=zet6221_i2c_write_tsdata(client, ts_cmd, 1);

    return 1;
}

u8 zet6221_ts_hwcmd(struct i2c_client *client)
{
    u8 ts_cmd[1] = {0xB9};

    int ret;

    ret=zet6221_i2c_write_tsdata(client, ts_cmd, 1);
```

```
    return 1;
}

int __init zet6221_downloader( struct i2c_client *client,
unsigned short ver, unsigned char * data )
{
    int BufLen=0;
    int BufPage=0;
    int BufIndex=0;
    int ret;
    int i;

#if defined(RSTPIN_ENABLE)
    msleep(5);
    gpio_direction_output(TS_RST_GPIO, 0);
    msleep(5);
    gpio_set_value(TS_RST_GPIO,GPIO_LOW);

    msleep(200);
#else
    zet6221_ts_hwcmd(client);
    msleep(200);
#endif

    //send password
    zet6221_ts_sndpwd(client);
    msleep(200);

    //sfr
    zet6221_ts_sfr(client);
    msleep(200);

    //mass erase
    zet6221_ts_masserase(client);
    msleep(200);

    BufLen=sizeof(zeitec_zet6221_firmware)/sizeof(char);

    while(BufLen>0)
    {
        memset(zeitec_zet6221_page,0x00,130);
        if(BufLen>128)
        {
            for(i=0;i<128;i++)
            {

    zeitec_zet6221_page[i+2]=zeitec_zet6221_firmware[BufIndex];
                BufIndex+=1;
            }
            zeitec_zet6221_page[0]=0x22;
            zeitec_zet6221_page[1]=BufPage;
            BufLen-=128;
        }
        else
        {
            for(i=0;i<BufLen;i++)
            {
```

```
    zeitec_zet6221_page[i+2]=zeitec_zet6221_firmware[BufIndex];
                BufIndex+=1;
            }
            zeitec_zet6221_page[0]=0x22;
            zeitec_zet6221_page[1]=BufPage;
            BufLen=0;
        }
        ret=zet6221_i2c_write_tsdata(client,      zeitec_zet6221_page,
130);
        msleep(200);
        BufPage+=1;
    }

#if defined(RSTPIN_ENABLE)
    gpio_set_value(TS_RST_GPIO,GPIO_HIGH);
    msleep(200);
#endif

    zet6221_ts_resetmcu(client);
    msleep(100);

}
```

5 AC CHARACTERISITIC

5.1 I2C Slave Mode Communication Timing

Figure 5-1shows the timing condition of the I2C interface. The characteristics of I2C interface are given in Figure 5-1. The Touch Panel adopts a bit rate of up to 400 Kbits/sec in Fast mode. The Touch Panel is defined as I2C slave mode interface. The Host generates the clock signal through the serial clock (SCL) pin and data are transferred and received through the serial data (SDA) pin. Clock synchronization should be enabled.

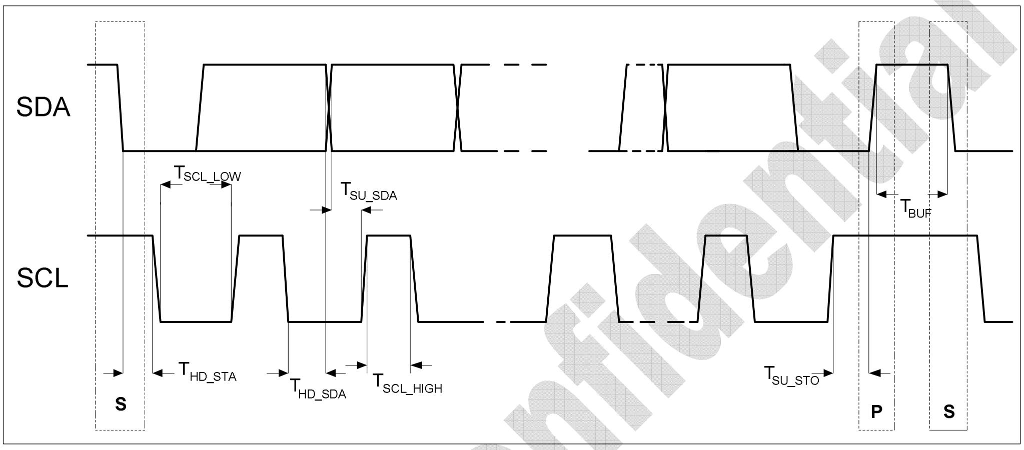


Figure 5-1 I2C Interface Timing

Table 5-1 Characteristics of the SDA and SCL Pins for I2C Interface

Symbol	Parameter	Standard Mode		Fast Mode		Unit
		Min.	Max.	Min.	Max.	
F_{SCL}	Frequency of SCL signal	0	100	0	400	kHz
T_{HD_STA}	Hold time (repeated) Start condition. After this period, the first clock pulse is generated.	4.0	–	1.2	–	µs
T_{SCL_LOW}	Low-State Period of SCL signal	5.3	–	1.5	–	µs
T_{SCL_HIGH}	High-State Period of SCL signal	2.7	–	0.8	–	µs
T_{HD_SDA}	Data Hold time	2.7	–	0.8	–	µs
T_{SU_SDA}	Data Set-Up time	2.7	–	0.8	–	us
T_{SU_STO}	Set-Up time for Stop condition	3.6	–	1.0	–	µs
T_{BUF}	Bus free time between a Stop and Start condition	30	–	30	–	µs

5.2 RC and Reset Timing

Table 5-2 IRC and Reset Timing

Symbol	Description	Min	Typ	Max	Unit
$T_{PowerOn}$	Power-on Slope Require (DVDD 0V to 3V)	5 μs	–	2.0 ms	–
T_{Init}	Power-on to Touch Available	–	–	30	ms

The input power must be a smooth and continuous ramp from 10% to 90% of 3V.

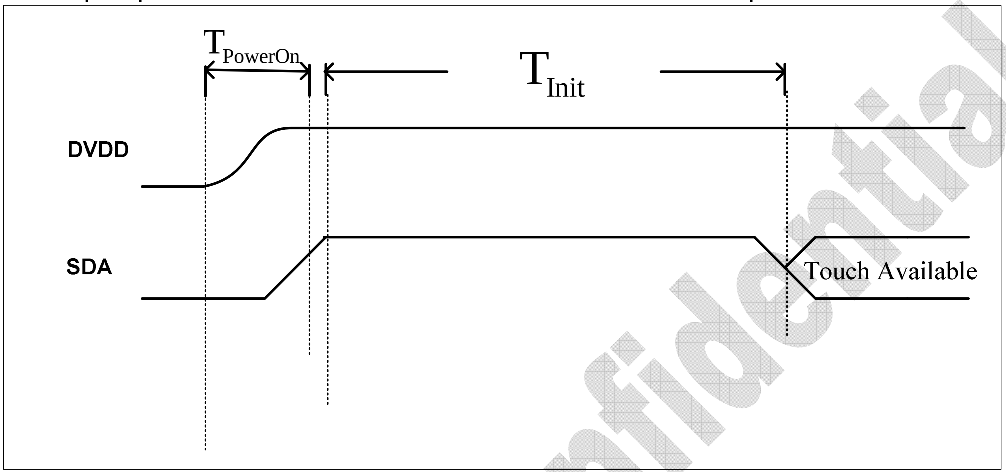


Figure 5-2 IRC Power-on Initialization Timing

6 DC CHARACTERISTICS

This section presents the DC electrical specifications of ZET6221 device. Specification are valid for -20℃ to 85℃ as specified.

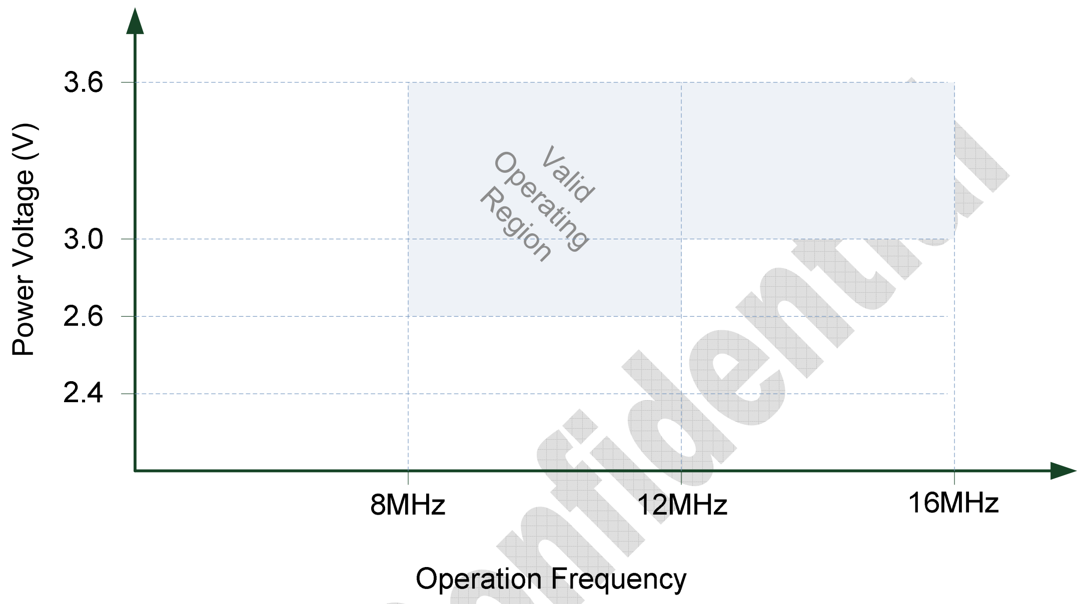


Figure 6-1 Valid Operating Region

Table 6-1 Unit of Measure

Symbol	Unit of Measure	Symbol	Unit of Measure
℃	Degree Celsius	ms	Mill second
dB	Decibels	mV	Mill volts
fF	Femto farad	nA	Nan ampere
Hz	Hertz	ns	Nanosecond
MHz	Megahertz	nV	Nan volts
μA	Microampere	Ω	Ohm
μF	Microfarad	pA	Pico ampere
μS	Microsecond	pF	Pico farad
μV	Microvolt	V	Volts
μW	Microwatts	σ	Sigma : One Standard Deviation
mA	Mill ampere	ps	Pico second

6.1 Absolute Maximum Ratings

Table 6-2 Absolute Maximum Ratings

Sym.	Parameter	Min.	Typ.	Max.	Units	Notes
T_{STG}	Storage Temperature	-55	25	+100	℃	
T_A	Ambient Temperature with Power Applied	-20	—	+85	℃	
VDD33	Supply Voltage on VDD33 Relative to VSS	-0.3	—	+3.6	V	
VDDIO	Supply Voltage on VDDIO Relative to VSS	-0.3	—	+3.6	V	
V_{IO}	DC Input Voltage	Vss-0.3	—	Vss+0.3	V	
V_{IOZ}	DC Voltage Applied to Tri-state	Vss-0.3	—	Vss+0.3	V	
I_{MIO}	Maximum Current into any Port Pin	-15	—	+30	mA	
ESD	Electrostatic Discharge Voltage	—	—	2000	V	Human Body Model ESD
LU	Latch-up Current	—	—	200	mA	

6.2 DC Electrical Characteristics

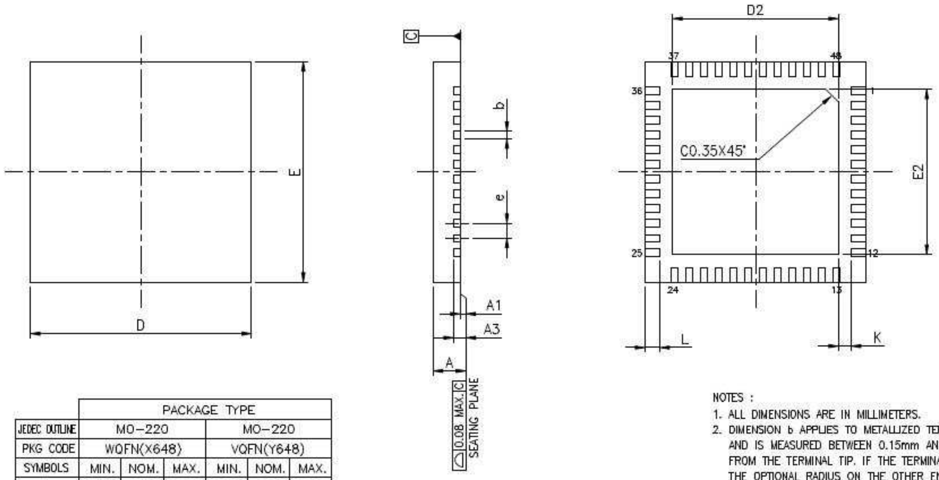
Table 6-3 DC Specifications

T_A =25℃

Sym.	Parameter	Min.	Typ.	Max.	Units	Condition
VDD33	Supply Voltage for Digital Circuit	2.6	–	3.6	V	VDD33=VDDIO
VDDIO	Supply Voltage for Digital I/O Pad	2.6	–	3.6	V	VDD33=VDDIO
VDDIO	Supply Voltage for Digital I/O Pad	1.65	–	3.6	V	VDD33>VDDIO
AVDD33	Supply Voltage for Analog Circuit	2.6	–	3.6	V	
IDD	Supply Current	1	4	8	mA	Depend on Sensor Mode and Sensor Number
V_{OH33}	High Output Voltage for Digital I/O Pad	2.8			V	VDDIO=3.3V,Iout=2~8mA
V_{OL33}	Low Output Voltage for Digital I/O Pad			0.3	V	VDDIO=3.3V,Iout=2~8mA
V_{IH33}	High Level Input Voltage for Digital I/O Pad	2.0			V	VDDIO=3.3V
V_{IL33}	Low Level Input Voltage for Digital I/O Pad			0.9	V	VDDIO=3.3V
V_{OH18}	High Output Voltage for Digital I/O Pad	1.2			V	VDDIO=1.8V,Iout=1~4mA
V_{OL18}	Low Output Voltage for Digital I/O Pad			0.3	V	VDDIO=1.8V,Iout=1~4mA
V_{IH18}	High Level Input Voltage for Digital I/O Pad	1.3			V	VDDIO=1.8V
V_{IL18}	Low Level Input Voltage for Digital I/O Pad			0.5	V	VDDIO=1.8V

7 PACKAGE INFORMATION

7.1 48Pin, QFN 6mm X 6mm



	PACKAGE TYPE					
JEDEC OUTLINE	MO-220			MO-220		
PKG CODE	WQFN(X648)			VQFN(Y648)		
SYMBOLS	MIN.	NOM.	MAX.	MIN.	NOM.	MAX.
A	0.70	0.75	0.80	0.80	0.85	0.90
A1	0.00	0.02	0.05	0.00	0.02	0.05
A3	0.20 REF.			0.20 REF.		
b	0.15	0.20	0.25	0.15	0.20	0.25
D	6.00 BSC			6.00 BSC		
E	6.00 BSC			6.00 BSC		
e	0.40 BSC			0.40 BSC		
L	0.35	0.40	0.45	0.35	0.40	0.45
K	0.20	—	—	0.20	—	—

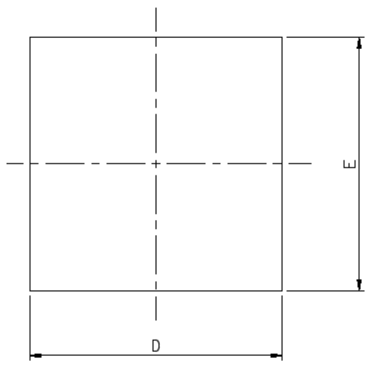
	E2			D2			LEAD FINISH		JEDEC CODE
PAD SIZE	MIN.	NOM.	MAX.	MIN.	NOM.	MAX.	Pure Tin	PPF	
185X185 MIL	4.40	4.50	4.55	4.40	4.50	4.55	V	X	W(V)JJE-1

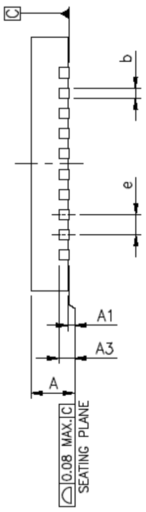
NOTES :

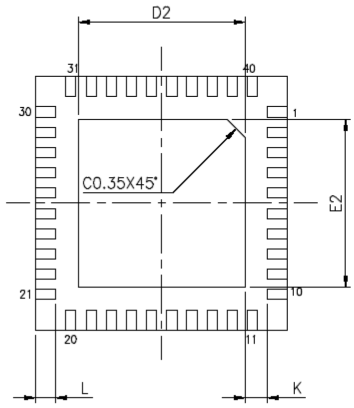
1. ALL DIMENSIONS ARE IN MILLIMETERS.
2. DIMENSION b APPLIES TO METALLIZED TERMINAL AND IS MEASURED BETWEEN 0.15mm AND 0.30mm FROM THE TERMINAL TIP. IF THE TERMINAL HAS THE OPTIONAL RADIUS ON THE OTHER END OF THE TERMINAL, THE DIMENSION b SHOULD NOT BE MEASURED IN THAT RADIUS AREA.
3. BILATERAL COPLANARITY ZONE APPLIES TO THE EXPOSED HEAT SINK SLUG AS WELL AS THE TERMINALS.

Figure 7-1 48Pin, QFN 6mm X 6mm

7.2 40Pin, QFN 5mm X 5mm







	PACKAGE TYPE					
JEDEC OUTLINE	MO-220			MO-220		
PKG CODE	WQFN(X540)			VQFN(Y540)		
SYMBOLS	MIN.	NOM.	MAX.	MIN.	NOM.	MAX.
A	0.70	0.75	0.80	0.80	0.85	0.90
A1	0.00	0.02	0.05	0.00	0.02	0.05
A3	0.20 REF.			0.20 REF.		
b	0.15	0.20	0.25	0.15	0.20	0.25
D	5.00 BSC			5.00 BSC		
E	5.00 BSC			5.00 BSC		
e	0.40 BSC			0.40 BSC		
L	0.35	0.40	0.45	0.35	0.40	0.45
K	0.20	—	—	0.20	—	—

	E2			D2			LEAD FINISH		JEDEC CODE
PAD SIZE	MIN.	NOM.	MAX.	MIN.	NOM.	MAX.	Pure Tin	PPF	
153X153 MIL	3.20	3.30	3.35	3.20	3.30	3.35	V	X	N/A

NOTES :

1. ALL DIMENSIONS ARE IN MILLIMETERS.
2. DIMENSION b APPLIES TO METALLIZED TERMINAL AND IS MEASURED BETWEEN 0.15mm AND 0.30mm FROM THE TERMINAL TIP. IF THE TERMINAL HAS THE OPTIONAL RADIUS ON THE OTHER END OF THE TERMINAL, THE DIMENSION b SHOULD NOT BE MEASURED IN THAT RADIUS AREA.
3. BILATERAL COPLANARITY ZONE APPLIES TO THE EXPOSED HEAT SINK SLUG AS WELL AS THE TERMINALS.

Figure 7-2 40Pin, QFN 5mm X 5mm

8 MARKING DEFINITION

8.1 Marking Information

Table 8-1 Ordering Information

Ordering Code	Package	Panel size (inch)	Analog Trace (I/O)
ZET6221WPA	QFN-56 (7mmX7mm, Pitch 0.4mm)	7"~10"	40
ZET6221WNA	QFN-48 (6mmX6mm, Pitch 0.4mm)	5"~8"	36
ZET6221WMA	QFN-40 (5mmX5mm, Pitch 0.4mm)	3"~7"	28
ZET6221WKA	QFN-28 (4mmX4mm, Pitch 0.45mm)	<4.3"	16

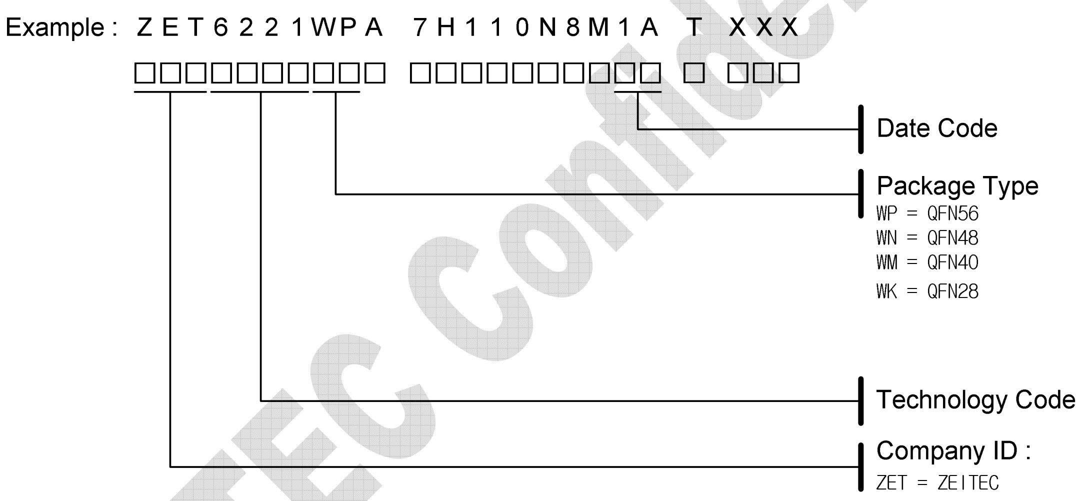


Figure 8-1 Ordering Code Definitions

Headquarters

ZEITEC Semiconductor Co., LTD.
3F., No.49, Jinshan 7th St., East Dist.,
Hsinchu City 300, Taiwan (R.O.C.)
Tel: +886 3 579 0045
Fax: +886 3 579 9960

Product Contact

Web Site
http://www.zeitecsemi.com

Sales Contact
sales@zeitecsemi.com