CIRQUE®

Pinnacle Touch Controller
Specification

Document Version 3.0

MARCH 2018
IC-DS-150408

This document provides the specification for the Cirque Pinnacle ASIC. Important features needed to READ, WRITE, and process data are included. Sample firmware is provided to show how to properly communicate with the Pinnacle ASIC using SPI or I2C protocol. This document applies to Cirque's Pinnacle 2.2 ASIC. For more information on Cirque's Pinnacle 1.4 ASIC please contact Cirque.

Document Version History

Date	Current Version	Description
APRIL 2015	2.0	Updated document to the new Cirque template.
MAY 2015	2.1	Added packaging information.
AUGUST 2015	2.1.1	Updated packaging information and added the shipping box information
MARCH 2018	3.0	Added additional electrical and interface details

Notice

This document is the sole property of Cirque Corporation. The information contained within is proprietary to Cirque Corporation. The holder of this document shall use the information only for its intended purpose, and shall protect the information in whole or in part from duplication, disclosure to any other party, or dissemination in any media without written permission from Cirque Corporation.

Note to Purchasers: All specifications subject to change without notice. Cirque shall not be liable for any damages whether non-specified, direct, indirect, special, incidental or consequential (including downtime, loss of profit or goodwill) regardless of the legal theory asserted. This document supersedes all previous versions.

Copyright, Trademark, and Patent Information

Cirque's touch controller platforms and technology solutions are protected by patents and additional U.S. and International patents pending. Contact Cirque for more information.

Table of Contents

Introduction

Pinnacle is a custom ASIC created by Cirque Corporation specifically for capacitive sensing of touchpads, touch screens, and other touch sensors. This document gives the electrical and mechanical specifications of the IC, along with the communication and register details for communicating with and using the IC in an application.

There are two types of Pinnacle ASICs (Pinnacle version 2.2 and Pinnacle AG version 1.4). While most information is the same for both versions, Pinnacle 2.2 is the latest release and this document contains information specific to the Pinnacle 2.2 ASIC. Contact Cirque if your application uses the Pinnacle AG 1.4 ASIC.

Pinnacle Overview

The Pinnacle ASIC monitors a capacitive touch sensor. When a finger (or a stylus) is placed on the surface of the sensor Pinnacle will determine the location of the object (user's finger or stylus). The position data is reported to the host as X and Y coordinates. The strength of the signal is reported as the Z coordinate.

Electrical Details

Pinnacle Pin Assignments

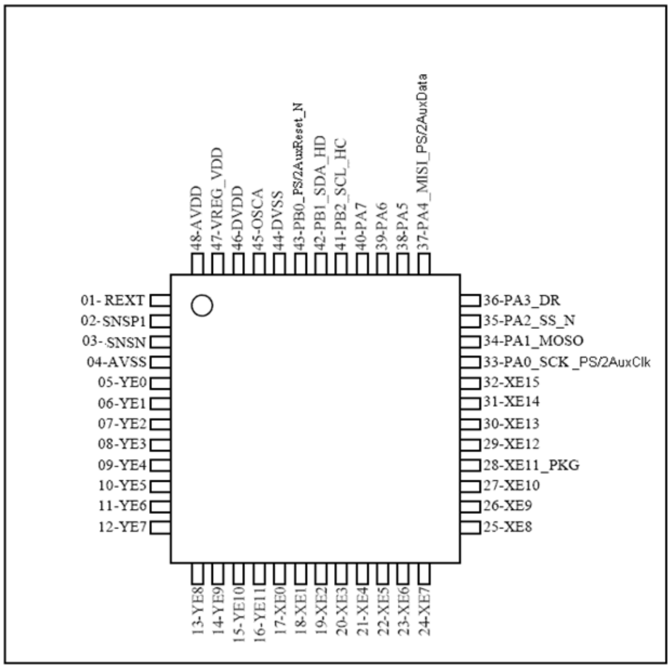


Figure 1: PINNACLE 48-PIN TQFP Package - 2.4 V -5.5 V Operating Range

Table 1. Pin Assignments for Cirque PINNACLE

Pin Number	Pin Name	Description
1	REXT	External bias resistor pin (Input). A resistor is tied between this pin and ground. It is used to adjust the sensitivity of the measurement system.
2	SNSP1	Sense Positive input pin(s) for sensor. A trace must be routed to the sensor and throughout the sensor grids. It is typically connected to AVSS through a 47 pF capacitor. **Note** *Must be shielded from noise.*
3	SNSN	Sense Negative input pin. This is an input to the analog measurement system. It is typically connected to AVSS through a 47 pF capacitor. **Note** *Must be shielded from noise.*
4	AVSS	Analog Ground
5 - 16	YE0-YE11	Y Electrode pins (outputs). Used to charge the sensor and determine finger position in the Y direction by switching patterns.
17-32	XE0-XE15	X Electrode pins (outputs). Used to charge the sensor and determine finger position in the X direction by switching patterns.
33-43	PA0~7, PB0~2	I/O pins used for Communications (PS/2, SPI, I2C), button inputs and other I/O needs.
44	DVSS	Digital Ground.
45	OSCA	Oscillator pin (input). Activates internal oscillator when connected to an external resistor to ground.
46	DVDD	Digital Power pin. Supplies digital logic, processor and memory.

Table 1. Pin Assignments for Cirque PINNACLE

Pin Number	Pin Name	Description
47	VREG_VDD	Supply voltage for 5 volt operation. Regulates the internal VDD voltage.
48	AVDD	Analog Power pin for analog measurement system and electrodes.

Sensor Configurations

Pinnacle has 28 total electrodes available for use. Within those 28 electrodes, there are four built in sensor configurations, listed below. Note that only one sense line is supported.

Possible electrode configurations:

- 6 Y electrodes by 8 X electrodes
- 6 Y electrodes by 16 X electrodes
- 12 Y electrodes by 8 X electrodes
- 12 Y electrodes by 16 X electrodes (Default)

Electrical Specifications

This section describes the Electrical specifications for the PINNACLE ASIC. The absolute maximum ratings details are provided in Table 2. These settings must be followed to ensure proper performance. IC operation cannot be guaranteed if any one of the recommended operating conditions listed below is exceeded.

The DC operating conditions are in table 3.

Table 2. PINNACLE Voltage Ranges

Symbol	Parameter	Rating	Unit
AVdd, DVdd	Power Supply	-0.3 to 4.0	V
V in / V out	Input or Output Voltage	-0.3 to Vdd+0.3	V
T stg	Storage Temperature	-55 to 150 degrees	C
R H	Relative Humidity	5 to 95%	% RH

Table 3. Pinnacle DC Operating Conditions

Symbol	Parameter	Minimum	Typical [1]	Maximum	Unit
3V Operation					
AVdd, DVdd	Power Supply	2.5	3.3	3.6	V
I active	Active Mode Current		3.2		mA
I idle	Idle Mode Current		2.0		mA
I sleep	Sleep Mode Current		40-80		µA
I shutdown[2]	Shutdown Mode Current		1	10	µA
T opr	Operating Temperature	-40		+85	C
RESET Voltage	Voltage threshold that activates RESET	2.2		2.4	V
ESD	Electro-static discharge[3]			15	kV
5V Operation					
AVdd, DVdd	Power Supply	4.5	5.0	5.5	V
I active	Active Mode Current		3.2		mA
I idle	Idle Mode Current		2.0		mA
I sleep	Sleep Mode Current		80-150		µA
I shutdown[2]	Shutdown Mode Current		40	60	µA
T opr	Operating Temperature	-40		+85	C
RESET Voltage	Voltage threshold that activates RESET	3.6		4.0	V
ESD	Module Level: lectro-static discharge[3] when properly installed in a module			15	kV

[1] Typical values are measured at a temperature of 25°C.

[2] Shutdown current requires all inputs to be at 0v or Vdd.

[3] Maximum ESD as used in reference design TCM5030.

Pinnacle Operation

Pinnacle supports three communications protocols, PS/2, I2C and SPI. Selecting the protocol is done using Hardware Configurable Options (described below). The sections below outline the electrical and timing details of each protocol.

Pinnacle has several features and options to achieve varied sensor functionality. This section describes how to establish proper communication with Pinnacle, and how to select the preferred features and options.

Using SPI and I2C the Pinnacle ASIC uses a simple Register Access Protocol (RAP) method to READ and WRITE to the Pinnacle's registers. The primary functions that aid communication with Pinnacle are described in this section.

Register Access Protocol (RAP)

Pinnacle's registers are read and written to, using a Register Access Protocol (RAP). RAP has only two functions, READ and WRITE.

Note: *See the sections on SPI or I2C for examples of how to perform the READ and WRITE operations.*

Registers are accessed by sending a byte in the format shown in Table 4. The Standard Registers have five-bit addresses that range from 0x00 to 0x1F; see Table 5 on page 9 for the register map.

Follow this process to avoid accidentally replacing existing settings in a register, when performing a WRITE sequence:

1. READ the register.
2. Modify the value using logical operators to set or clear only the intended bits.
3. WRITE the modified value to the register.

Table 4. Register Access Protocol

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
READ (0xAX)	1	0	1	Address 4	Address 3	Address 2	Address 1	Address 0
WRITE (0x8X)	1	0	0	Address 4	Address 3	Address 2	Address 1	Address 0

Standard Register Set

Pinnacle contains 32 standard registers, shown in the table below.

Table 5. Standard Register Set

Address	Pinnacle	Description		
0x00	Firmware ID	Firmware ASIC ID	R	0x07
0x01	Firmware Version	Firmware revision number.	R	0x3A
0x02	Status1	Contains status flags about the state of Pinnacle.	R/W	
0x03	SysConfig1	Contains system operation and configuration bits.	R/W	
0x04	FeedConfig1	Contains feed operation and configuration bits.	R/W	
0x05	FeedConfig2	Contains feed operation and configuration bits.	R/W	
0x06	FeedConfig3	Contains feed operation and configuration bits.	R/W	
0x07	CalConfig1	Contains calibration configuration bits.	R/W	
0x08	PS/2 Aux Control	Contains Data register for PS/2 Aux Control.	R/W	
0x09	Sample Rate	Number of samples generated per second.	R/W	
0x0A	ZIdle	Number of Z=0 packets sent when Z goes from >0 to 0.	R/W	
0x0B	Z Scaler	Contains the pen Z_On threshold.	R/W	
0x0C	Sleep Interval		R/W	
0x0D	Sleep Timer		R/W	
0x0E	Dynamic EMI Adjust Threshold	Threshold to adjust EMI settings	R/W	
0x0F	RESERVED	RESERVED	R	
0x10	RESERVED	RESERVED	R	
0x11	RESERVED	RESERVED	R	
0x12	PacketByte_0	trackpad Data	R	
0x13	PacketByte_1	trackpad Data	R	
0x14	PacketByte_2	trackpad Data	R	
0x15	PacketByte_3	trackpad Data	R	
0x16	PacketByte_4	trackpad Data	R	
0x17	PacketByte_5	trackpad Data	R	
0x18	PortA GPIO Control	Control of Port A GPIOs	R/W	
0x19	PortA GPIO Data	Data of Port A GPIOs	R/W	
0x1A	PortB GPIO Control and Data	Control and Data of PortB GPIOs	R/W	
0x1B	Extended Register Access Value	Value for extended register access.	R/W	
0x1C	Extended Register Access Address High	High byte of 16 bit extended register address	R/W	
0x1D	Extended Register Access Address Low	Low byte of 16 bit extended register address	R/W	
0x1E	Extended Register Access Control	Control of extended register access	R/W	
0x1F	Product ID	Product ID	R	

Status and Data Ready Signals

Hardware Data Ready

To interrupt or alert the host when data is ready, Pinnacle uses a Hardware Data Ready (HW_DR) signal (active HIGH) that is triggered by either the Command Complete (SW_CC) flag or the Software Data Ready (SW_DR) flag in the Status1 register address 0x02. When either software flag is set, the HW_DR (pin 36 PA2) is also asserted. The SW_DR and SW_CC flags must be manually cleared by the host. The HW_DR signal remains asserted while either SW_DR or SW_CC is asserted (See Table 6 below).

Table 6. Data Ready and Command Complete

Signal	Description
Hardware Data Ready (HW_DR)	Asserted (HIGH) with either SW_DR or SW_CC flag. Cleared when both SW_DR and SW_CC flags are cleared.
Software Data Ready (SW_DR)	Asserted with new data. Remains asserted until cleared by host.
Command Complete (SW_CC)	Asserted after calibration, POR. Remains asserted until cleared by host.

Command Complete

The Software Command Complete (SW_CC) flag (Bit [3] of Register 0x02, Status1) is asserted after successful completion of either power on reset (POR) or calibration. This flag triggers the HW_DR signal, which then alerts the Host. The host must clear the SW_CC flag, by writing 0x00 to the Status1 register (0x02), in order to clear HW_DR (see Table 7).

To clear Command Complete and Software Data Ready flags simultaneously, see the code below in *ClearFlags Example Code*. The RAP Write byte is explained in *Writing to a Register Using SPI on page 26*.

ClearFlags Example Code

```
void ClearFlags(void) {
      RAP_WriteByte(0x02,  0x00);  //  Write  0x00  to  Status1
      register  (0x02)
}
```

Software Data Ready

When a touch is detected, Pinnacle loads X and Y position data into the position registers and asserts the SW_DR flag (Bit [2] of Register 0x02, Status 1), which also triggers the HW_DR signal (see Table 7 below). While the finger/stylus is present, the position registers are updated every 10 ms and SW_DR and HW_DR are asserted. The host must clear SW_DR, by writing a value of 0x00, to the Status1 register (0x02), which will clear HW_DR and ensure no data ismissed.

Note: *SW_CC and SW_CC flags may be cleared by the same write operation.*

Table 7. Status1 - Register 0x02

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Description					Command Complete (SW_CC)	Software Data Ready (SW_DR)		
READ/WRITE					R/W	R/W		
Values					0 = Clear	0 = Clear		
Default					0	0		

Data Output

To receive position data from Pinnacle, the data feed must be enabled by asserting the Feed Enable flag (Bit [0] Register 0x04, FeedConfig1) (see Table 8). Once enabled, Pinnacle's finger tracking, sampling, and reporting begins.

Pinnacle can report position as relative data or absolute data. Relative data is identical to a mouse or pointer where each position is reported as relative to the last position. Absolute data is a grid with X positions and Y positions (Pinnacle range: X= 0 - 2047, Y= 0 - 1535, Pinnacle AG range: X= 0 - 1919, Y= 0 - 1407).

The absolute data mode allows the designer a wider range of freedom to process the data and design the preferred touch functions. In absolute mode, Y and X data can be inverted to allow different orientations of the trackpad. The data output mode (relative or absolute) is specified using Bit[1] in Register 0x04, FeedConfig1 (see Table 8). Advanced features for relative mode can be set using flags in Register 0x05, FeedConfig2 (see Table 9 on page 12).

Table 8. FeedConfig1 (Data Output Flags) - Register 0x04

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Description	Y data Invert[1]	X data Invert[1]		Y Disable[2]	X Disable[3]	Filter Disable[4]	Data mode	Feed Enable
READ/ WRITE	R/W	R/W		R/W	R/W	R/W	R/W	R/W
Values	1=Y max to 0 0=0 to Y max	1=X max to 0 0=0 to Y max		1=no Y data 0=Y data	1=no X data 0=X data	1=no filter 0=filter	1=absolute 0=relative	1=feed 0=no feed
Default	0	0		0	0	0	0	0

[1] Y and X data count invert is only available when in absolute mode.
(Pinnacle range: X= 0 - 2047, Y= 0 - 1535, Pinnacle AG range: X= 0 - 1919, Y= 0 - 1407)

[2] Disabling the Y-axis will not allow regular tracking and is not recommended for typical applications.

3. Disabling the X-axis will not allow regular tracking and is not recommended for typical applications.
4. The Filter disable bit controls whether the filtering algorithm is applied to generated data. By default the hardware filters are enabled. Cirque does not recommend disabling hardware filtering.

Table 9. FeedConfig2 (Feature flags for Relative Mode Only) - Register 0x05

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Description	Swap X & Y			GlideExtend® Disable [1]	Scroll Disable	Secondary Tap Disable [2]	All Taps Disable [3]	Intellimouse Enable [4]
READ/ WRITE	R/W			R/W	R/W	R/W	R/W	R/W
Values	1=90° rotation 0=0° rotation			1= disable 0 = enable	1= disable 0 = enable	1= disable 0 = enable	1= disable 0 = enable	1= enable 0= disable
Default	0			0	0	0	0	0

1. GlideExtend is Cirque's patented motion extender technology that allows the user to continue the drag function when an edge is reached, by lifting and repositioning the finger.

2. Secondary Taps allows a tap in the upper right corner (standard orientation) to simulate an activation of the secondary button.

3. Disabling all taps disables secondary taps, even if secondary tap is explicitly enabled.

4.Intellimouse enabled will change Pinnacle's relative data packet to four bytes rather than three. The fourth byte (PacketByte_3) will report scroll data (referred to as wheel count).

Relative Data Mode Data Packet

Pinnacle reports a change (or delta) in current position from the previous position in relative mode. The position deltas are stored in PacketByte_1 (X delta) and PacketByte_2 (Y delta). The X and Y data sign (either negative (-) or Positive (+)) indicates the direction of change and is reported in PacketByte_0.

Button Data

Relative Data Mode supports three button inputs. Button data can be generated by the actual button or by taps (if enabled) that simulate a button press. Button data is also reported in PacketByte_0. If Intellimouse is enabled (Bit[0] of Register 0x05, FeedConfig2), scroll data (wheel count) is reported in an additional byte, PacketByte_3 (See Table 10).

Table 10. Relative Position Registers Byte Format

Address	Description	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PacketByte_0	X & Y Sign, Button (or tap) data	0	0	Y sign 1=(-) 0=(+)	X sign 1=(-) 0=(+)	1	BTN Auxiliary 1=pressed 0=released	BTN Secondary (or top right corner tap) 1=pressed 0=released	BTN Primary (or tap) 1=pressed 0=released
PacketByte_1	X Delta	X7	X6	X5	X4	X3	X2	X1	X0
PacketByte_2	Y Delta	Y7	Y6	Y5	Y4	Y3	Y2	Y1	Y0
PacketByte_3	Wheel Count	W7	W6	W5	W4	W3	W2	W1	W0

Example: To go from a positive delta to a negative would look like this:

```
3
2
1
0
255(sign bit is set, making -1) 254(sign bit is set, making -2)
252(sign bit is set, making -3)
```

Absolute Data Packet

Absolute mode is entered by setting Bit[1] in FeedConfig1 (Register 0x04) (see Table 8 on page 11). The Host reads whichever registers meet the application needs. For example, if button status is not needed, the host can read the last four packet bytes.

Table 11. Absolute Position Registers Byte Format

Pinnacle Address	Pinnacle AG Address	Description	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x12	**0x10**	Button/Switch Status			SW5	SW4	SW3	SW2	SW1	SW0
0x14	**0x11**	X-Position Low Byte	X7	X6	X5	X4	X3	X2	X1	X0
0x15	**0x12**	Y-Position Low Byte	Y7	Y6	Y5	Y4	Y3	Y2	Y1	Y0
0x16	**0x13**	X & Y Position High Bits	Y11	Y10	Y9	Y8	X11	X10	X9	X8
0x17	**0x14**	Z-Level			Z5	Z4	Z3	Z2	Z1	Z0

Example Code

```
uint16_t calculateXPosition(uint8_t * registerVals) {
return registerVals[0x14] | ((registerVals[0x16] & 0x0F) << 8);
}

uint16_t calculateYPosition(uint8_t * registerVals) {
return registerVals[0x15] | ((registerVals[0x16] & 0xF0) << 4);
}
```

Power Modes

Pinnacle has four power modes - Active (touch detected), Idle (no touch), Low Power/ Sleep (lower power after ~ 5 seconds of inactivity) and Shutdown/Standby (no data reported) (See Figure 1 below).

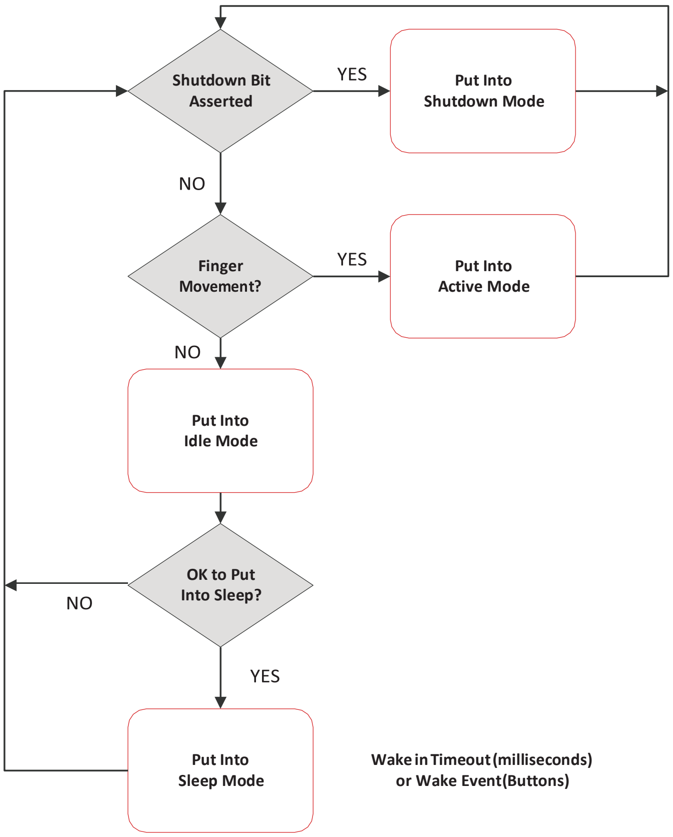


Figure 2. Pinnacle Power Mode State Machine

Active and Idle Mode

By default, Pinnacle toggles between Active and Idle mode. Pinnacle is in Active mode when a touch is detected (that is, a finger or stylus is present and is moving or tapping on the trackpad). The measurement system is active and data packets are being created and then sent to the host. Active mode begins as soon as a touch is detected. Idle mode is entered when the finger has been removed and there are no data packets to be sent. While in Idle mode, Pinnacle wakes every 10 milliseconds to check for a touch.

Low Power Sleep Mode

Enabling sleep mode will cause Pinnacle to go into a low power mode (around 50 µA) within 5 seconds of no touch detection. While in sleep mode, Pinnacle will wake within 300 ms to report any detection of a finger/stylus. To enable sleep mode, assert the Sleep Enable flag, Bit [2] of Register 0x03, SysConfig1 (see Table 12 on page 16). The bit remains asserted and Pinnacle uses low power sleep mode until the Host clears the Sleep Enable flag.

Shutdown/Standby Mode

Shutdown/Standby mode is a very low power mode and Pinnacle does not track touch in this mode. Shutdown/Standby is activated by the host when the Shutdown flag is asserted (Bit [1] Register 0x03, SysConfig1) and deactivated when the bit is cleared.

Table 12. SysConfig1 (Low Power Mode) - Register 0x03

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Description						Sleep Enable [1]	Shutdown [1]	Reset
READ/WRITE						R/W	R/W	R
Values						1=low power mode 0=normal mode	1=Shutdown 0=Active	1 = Reset
Default						0	0	0

[1] Request additional instructions from Cirque for exiting Sleep and Shutdown modes if your application requires SPI at 1 MHZ or greater.

Touch Detection (Z Information)

Z Level

The Z level is a measure of how much the capacitive field has changed. When no finger is near, the Z level will be at or near zero (0). The Z level will start increasing as a finger approaches. The Z level continues to increase as more of the surface area of the finger touches the surface of the sensor. Position data is generated for any Z level greater than zero (0). Overlay types (glass vs. plastic) on the sensor and application requirements are factors that the developer needs to consider when setting Z level thresholds for a touch and a release. With absolute data mode enabled, Z-level values are reported in Bits [0:5] of the Z-level Packet Byte (see Table 11on page 14).

Z-Idle

During Z-idle (no touch detected) and when in absolute data mode, Pinnacle will continue to send empty packets (both X and Y data set to 0x00) every 10 ms. The number of empty packets to be sent is stored in Register 0x0A, Z-idle (see Table 13 below). The default value is 0x1E (30 decimal). This value can be changed by writing to Register 0x0A. When set to zero (0), this register prevents any empty packets from being sent, and the position registers will contain the last sensed location until a new finger presence is detected.

The Z-Idle count can be a helpful design tool. For example, tap-frequency can be determined by counting the number of Z-idle packets reported between a finger lifting off and touching back down (cutting short the stream of Z-idle packets).

Table 13. Z-Idle Count - Register (0x0A)

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Description	Z-Idle Count							
READ/WRITE	R/W							
Values	0x00 through 0xFF (0 to 255 decimal)							
Default	0X1E							

Compensation

To enhance sensor performance, Pinnacle performs a calculation to compensate for the current operating environment. Pinnacle automatically executes compensation/calibration when powered on and when triggered by specific events.

Example Start-up Sequence

The following summarizes the typical sequence to use Pinnacle:

1. Power on Reset (POR); Pinnacle is in the default condition.
2. Host clears SW_CC (writes value 0x00 to Register 0x02, Status1), which clears HW_DR.
3. Host configures intended bits of registers 0x03 and 0x05.
4. Host enables the preferred output mode and enables the feed (As explained in *Touch Detection (Z Information) on page 17*.).
5. Host continuously monitors HW_DR pin for Pinnacle data ready interrupt.
   a. When received, host reads the appropriate packet data (Registers 0x10-0x17, depending on ASIC and data mode).
   b. Host clears SW_DR (writes value 0x00 to Register 0x02, Status1), which clears HW_DR.
   c. Host uses the data received.

Example Code

```
void StartupSequence(void)
{
  ClearFlags();
  RAP_WriteByte(0x03, 0x00); // Configures SysConfig1(normal
  mode, active)
  RAP_WriteByte(0x05, 0x1F); // Configures FeedConfig2(disable
  relative mode features)
  RAP_WriteByte(0x04, 0x03); // Configures FeedConfig1(absolute
  mode, enable feed)
}
```

Serial Peripheral Interface (SPI)

This section describes the basic operation, timing of SPI, as well as how it is used to communicate with Pinnacle.

SPI communication is a four-wire bus that includes a slave select line for communicating with multiple devices on the bus. Pinnacle is designed as a slave device on the SPI bus with a select line (SS), two data lines (MOSI and MISO), and a clock line (SCK), see Table 14 below.

Table 14. Pinnacle SPI Signals

Signal	Description
SPI	
SS	SPI Select (SS=0 for slave to be active and send data to master)
SCK	SPI Clock (CPOL=0=idle) (CPHA=1, latches on falling edge)
MOSI	SPI Master Out Slave In
MISO	SPI Master In Slave Out
Pinnacle	
HW_DR	Indicates Pinnacle has Data Ready (Active High) or a command is complete.
GPIO	General Purpose Input/Output
GND	Ground
VDD	Power Supply

SPI signal operation:

- A slave device will not communicate on the bus unless its slave select line (SS) is pulled low.
- Data sent by the master is placed on the Master Out, Slave In (MOSI) line.
- Data returned by Pinnacle is placed on the Master In, Slave Out (MISO) line.
- Both sets of data are latched on the falling edge of SCK.
- Data is presented Most Significant Byte first (MSB).
- Pinnacle supports data rates up to 13 MHz.

A basic example of interconnections between a single master and a single slave is to have the MOSI pins connected together and the MISO pins connected together. In this way, the data is transferred serially between master and slave (most significant bit first). The master always starts the data exchange. When the master device transmits data to a slave device on the MOSI pin, the slave sends data to the master on the MISO pin at the same time. This is full-duplex communication. All the data is synchronized using the SCK pin with a clock signal from the master device. Slave Select (SS) must be LOW in order for Pinnacle to be active and send data to the Master.

SPI Timing

Timing for the SPI data lines is shown below (see Figure 2, Figure 3 below, and Table 15 on page 21). The data exchange is started when the master pulls the Slave Select (SS) line low. Due to the full duplex nature of the SPI bus, Pinnacle always returns data at the same time that it is receiving data. The returned byte may be data from the previous command or it may be a filler byte. For both Master and Slave, data is changed on the rising edge of the clock and latched on the falling edge.

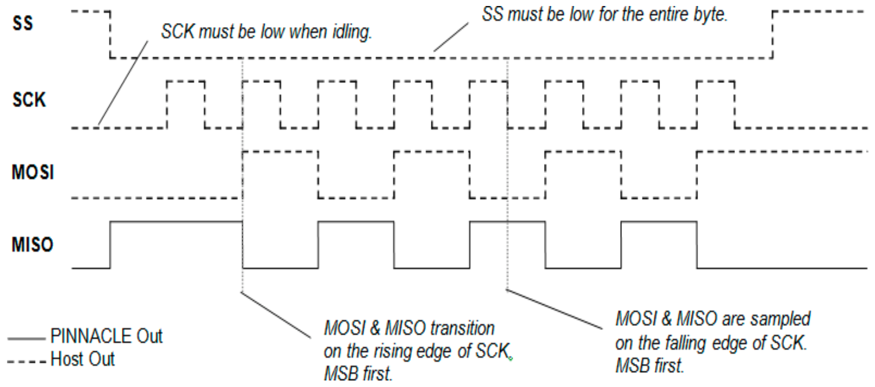

SS
SCK must be low when idling.
SS must be low for the entire byte.
SCK
MOSI
MISO
PINNACLE Out
Host Out
MOSI & MISO transition on the rising edge of SCK, MSB first.
MOSI & MISO are sampled on the falling edge of SCK. MSB first.

Figure 3. SPI Signals

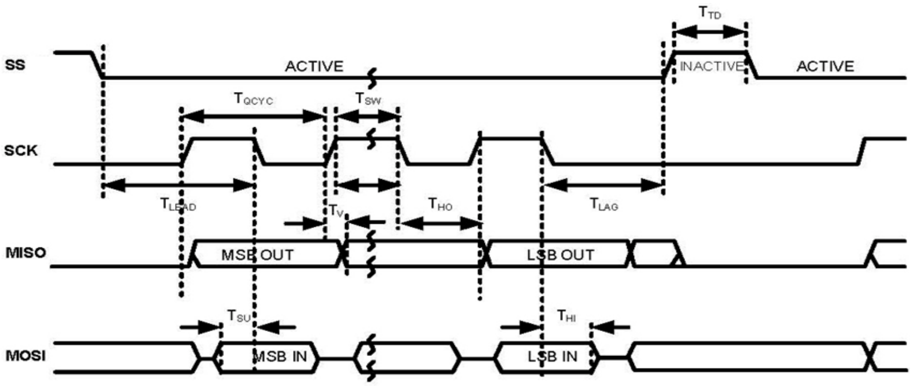


Figure 4. SPI Timing

Table 15. SPI Timing Explanation

Ref	Parameter	Symbol	Minimum	Maximum	Unit
1	Period	T_{QCYC}	76	N/A	ns
2	Clock (SCK) High or Low Time	T_{SW}	$T_{QCYC}/2$	N/A	
3	Chip Select Lag Time (last clock edge to slave select de-asserted)	T_{LAG}	30	N/A	
4	Inter-Message Transfer Delay (required by slave)	T_{TD}	50	N/A	
5	Chip Select Lead Time (slave select asserted to first SCK falling)	T_{LEAD}	T_{QCYC}	N/A	
6	Master Data Setup Time (data valid to SCK falling)	T_{SU}	$T_{QCYC}/4$	N/A	
7	Master Data Hold time (SCK falling to data invalid)	T_{HI}	$T_{QCYC}/4$	N/A	
8	Slave Data Valid Time (SCK rising to data valid)	T_V	N/A	30	
9	Slave Data Hold Time (SCK falling to data invalid)	T_{HO}	$T_{QCYC}/2$	N/A	

Register Access Using SPI

Communicating with Pinnacle requires RAP, which has only READ and WRITE commands. The byte format is repeated in Table 16.

Table 16. Register Access Protocol

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
READ (0xAX)	1	0	1	Address 4	Address 3	Address 2	Address 1	Address 0
WRITE (0x8X)	1	0	0	Address 4	Address 3	Address 2	Address 1	Address 0

Reading a Register Using SPI

SPI requires two exchanges when reading a register.

1. The host starts by sending the READ command byte. The simultaneous response byte may be data from the previous command or it may be a filler byte.

2. Three additional bytes (“filler bytes”) with a value of 0xFB, must be sent after the READ command byte to allow Pinnacle to respond. The response will be sent during the transmission of the final byte (see Table 17 below).

SPI Read Example

Read register 0x02 (Status1)

1. Use the OR operand to combine the READ command (0xAX) with the register address to be read. For this example, use register 0x02.

   `0xA0|0x02 = 0xA2`

2. Send 0xA2 followed by three-filler bytes (0xFB). The response to the third byte will be the contents of register 0x02.

Table 17. Example SPI READ Sequence

Byte	Command (MOSI)	Response (MISO)
1	0xA2	Response to a previous command or a filler byte.
2	0xFB	0xFB
3	0xFB	0xFB
4	0xFB	Content of register 0x02

SPI Read Example Code

```
void RAP_ReadByte(uint8_t address, uint8_t * data) {
    Assert_CS();
    SPI_Transfer(address | 0xA0); // Send register address OR'd with read-mask (0xA0)
    SPI_Transfer(0xFB); // Filler-byte
    SPI_Transfer(0xFB); // Filler-byte
    *data = SPI_Transfer(0xFB); // Contents are received on 3rd filler-byte
    DeAssert_CS();
}
```

Pipelining Multiple Reads

Multiple registers can be read by sending READ commands back-to-back. For greater throughput, back-to-back READ commands can overlap the last filler byte with the next READ command. Alternatively, an auto-increment command (0xFC) can be used in place of the filler byte (0xFB). Examples are provided for each method below.

Reading SPI Back-to-Back

Read register 0x02 (Status1) and 0x17 (Z Level) (see Table 18).

1. Use OR operand to combine the READ command (0xAX) with the register addresses to be read. For this example, use registers 0x02 and 0x17.

```
0xA0     | 0x02 =0xA2
0xA0     | 0x17 =0xB7
```

2. Send 0xA2 followed by three-filler bytes (0xFB). The response to the third byte will be the contents of register 0x02.

3. Send 0xB7 followed by three-filler bytes (0xFB). The response to the third byte will be the contents of register 0x17.

Table 18. Example of Back-to-Back SPI READ Commands

Byte	Command (MOSI)	Response (MISO)
1	0xA2	Response to a previous command or a filler byte.
2	0xFB	0xFB
3	0xFB	0xFB
4	0xFB	Content of register 0x02
5	0xB7	0xFB
6	0xFB	0xFB
7	0xFB	0xFB
8	0xFB	Contents of register 0x17

SPI Auto-Increment READ for Sequential Addresses Example

Read registers 0x14 through 0x16 (see Table 19).

1. Use OR operand to combine the READ command (0xAX) with the register addresses to be read.

```
0xA0  |   0x14 =0xB4
0xA0  |   0x15 =0xB5
0xA0  |   0x16 =0xB6
```

2. Send 0xB4 followed by four Auto-Increment bytes (0xFC) and end with a filler byte (0xFB) to read three sequential registers. The response to the third 0xFC will be the contents of register 0x14, the next response will be the data in register 0x15, then register 0x16 (see Table19).

Table 19. Example of SPI Auto-Incremented READ Command Sequence

Byte	Command (MOSI)	Response (MISO)
1	0xB4	Response to a previous command or a filler byte.
2	0xFC	0xFB
3	0xFC	0xFB
4	0xFC	Contents of register 0x14
5	0xFC	Contents of register 0x15
6	0xFB	Contents of register 0x16

SPI Auto-Increment READ for Sequential Addresses Example Code

```
// (note that this function can be used to read a single byte or multiple;
making the RAP_ReadByte() function redundant)
void RAP_ReadBytes(uint8_t address, uint8_t * data, uint8_t count) { uint8_t
i = 0;

  Assert_CS();
  SPI_Transfer(address | 0xA0); // Send register address OR'd with read-mask
  (0xA0) SPI_Transfer(0xFC); // Special auto-increment filler byte
  SPI_Transfer(0xFC); // Special auto-increment filler byte
  for(; i < count; i++) {
  data[i] = SPI_Transfer(0xFC); // Each transfer gets the next register's
  contents
  }
  DeAssert_CS();
  }

// RAP_ReadBytes() example to read the entire register set into an array
uint8_t registerContents[0x17];
RAP_ReadBytes(0x00,  registerContents,  0x17);  //  Reads  0x17  (quantity)
registers,  start- ing at address 0x00
```

Writing to a Register Using SPI

Writing to a register requires two SPI exchanges. The host starts by sending the WRITE command byte (0x8X) with the desired register address. The simultaneous response byte from Pinnacle may be data from the previous command or it may be a filler byte (0xFB). The host then sends the value to be written to the register in the next byte. The simultaneous response byte is 0xFB. Multiple writes are sent as consecutive single writes (with a repeating sequence of command byte/data byte).

SPI Write Example

WRITE the value 0x02 to register 0x03 (SysConfig1 Shutdown Bit) (see 20).

1. Use OR operand to combine the WRITE command (0x8X) with the register addresses to be read.

   ```
   0x80  |   0x03 =0x83
   ```

2. Send 0x83. The response will be a filler byte or the response to the previous command.

3. Send the value 0x02. The value 0x02 will then be written to register 0x03. The response will be a filler byte.

Table 20. Example of SPI WRITE Command

Byte	Command (MOSI)	Response (MISO)
1	0x83	Response to a previous command or a filler byte
2	0x02	0xFB (value 0x02 is written to register 0x03)

SPI Write Example Code

```
void RAP_WriteByte(uint8_t address, uint8_t data) {
   Assert_CS();
   SPI_Transfer(address | 0x80); // Send register address OR'd with
   write-mask (0x80) SPI_Transfer(data); // Send data to be written
   DeAssert_CS();
}
```

}

Inter Integrated Circuit (I^{2}C) Protocol

This section provides a brief overview of the I^{2}C protocol as well as how to communicate to Pinnacle using I^{2}C.

I^{2}C Operation

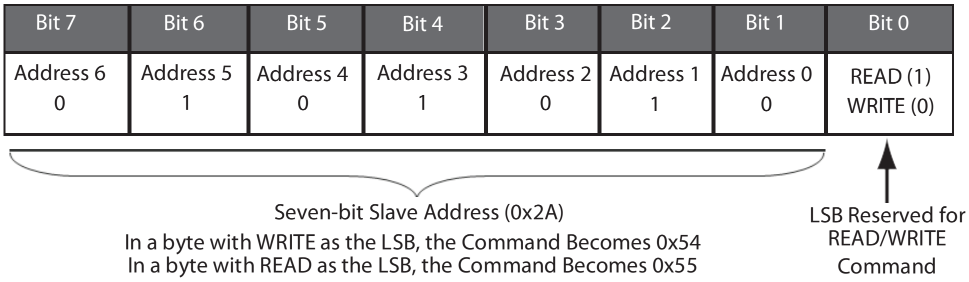
I^{2}C communication is a two-wire bus that uses module addressing for communicating with multiple devices on the bus. The I^{2}C bus contains a data signal (SDA) and a clock signal (SCL), see Table 21. The clock is provided by the master and is an input to all slave devices. The data line is both an input and an output (bidirectional).

Table 21. Pinnacle I^{2}C Signals

Signals	Description
I^{2}C	
SDA	Serial Data line for both input and output for master and slave.
SCL	I^{2}C Serial Clock Line provided by the master as an input to all slave devices.
Pinnacle	
HW_DR	Indicates Pinnacle has Data Ready to send
GPIO	General Purpose Input/Output
GND	Ground
VDD	Power supply

Pinnacle is designed as a slave device on the I^{2}C bus. As per the I^{2}C protocol, each slave is assigned a unique, seven-bit slave address. All slave devices on the bus receive all commands, but only the addressed slave device can respond to communication on the bus. The default slave address for Pinnacle is 0x2A.

I^{2}C slave addresses are only seven bits long to allow the eighth bit to signify a READ or WRITE command. In the first byte sent by the host, the slave address comes first, followed by the READ/ WRITE bit (at the lowest significant bit). Therefore, the hexadecimal value for the first byte is the slave address (0x2A) shifted up one bit (0x54), plus the READ/WRITE bit. The final value is either 0x54 for WRITE or 0x55 for READ (see Figure 4).



Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Address 6 0	Address 5 1	Address 4 0	Address 3 1	Address 2 0	Address 1 1	Address 0 0	READ (1) WRITE (0)

Seven-bit Slave Address (0x2A)
In a byte with WRITE as the LSB, the Command Becomes 0x54
In a byte with READ as the LSB, the Command Becomes 0x55

LSB Reserved for READ/WRITE Command

Figure 5. I^{2}C READ or WRITE Command with Pinnacle's Address

When the bus is idle, the SDA and SCL lines are high due to pull-up resistors on each line. The Master starts communication with a start condition, which is a high to low transition on the SDA line while the SCL line remains high. After the start signal, Data is placed on the SDA line when the SCL line is low. The receiving device latches the data when the SCL line is high. All 8-bits are transmitted and a 9th bit is designated as the acknowledge bit (ACK [0] /NACK [1]). The first byte transmitted will always be a 7-bit address of the target Slave and a READ/WRITE bit indicating the direction of data flow. Any bytes following the Slave address byte are only for the addressed Slave. When reading from the Slave, the Master must NACK the last byte to indicate to the Slave that it was the last byte to be exchanged. After at least one data byte is sent or received, the Master can cause a stop condition to free the bus. The stop condition requires SDA to transition from low to high, while SCL is high.

I^{2}C is half-duplex, and therefore responses must be read after a transmission rather than during a transfer. After a READ command, the slave hardware will ACK/NACK on the 9th bit of a byte. A NACK (9th bit asserted by slave) will denote a BUSY condition or an ERROR condition, and the slave will follow it with a status byte to indicate the error or response. A NACK from the slave requires the host to end the transfer and read a status byte carrying a response. When writing, the Slave is not required to NACK the last byte because the Slave does not know how many bytes it is to receive. A Stop Condition occurs when the SDA line rises, while the SCL line is high.

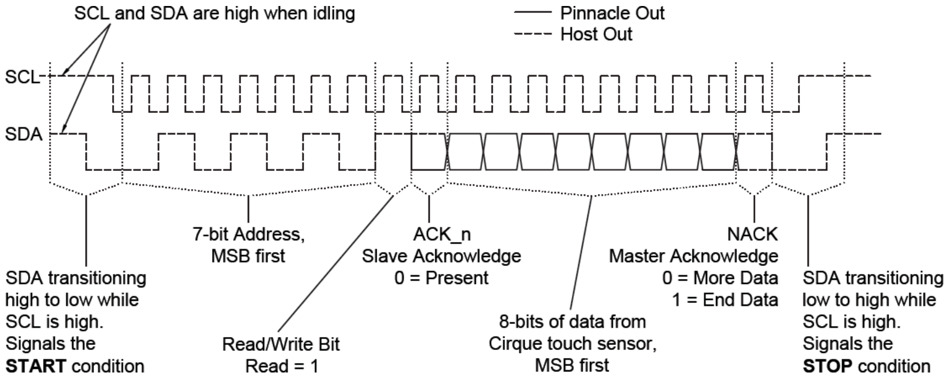

SCL and SDA are high when idling
— Pinnacle Out
---- Host Out
SCL
SDA
7-bit Address, MSB first
ACK_n Slave Acknowledge 0 = Present
NACK Master Acknowledge 0 = More Data 1 = End Data
SDA transitioning high to low while SCL is high. Signals the START condition
Read/Write Bit Read = 1
8-bits of data from Cirque touch sensor, MSB first
SDA transitioning low to high while SCL is high. Signals the STOP condition

Figure 5. I^{2}C Signals - READ

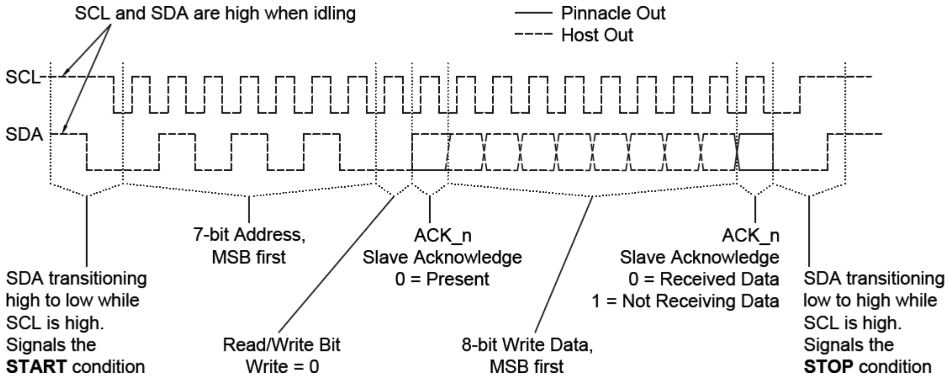

SCL and SDA are high when idling
— Pinnacle Out
---- Host Out
SCL
SDA
7-bit Address, MSB first
ACK_n Slave Acknowledge 0 = Present
ACK_n Slave Acknowledge 0 = Received Data 1 = Not Receiving Data
SDA transitioning high to low while SCL is high. Signals the START condition
Read/Write Bit Write = 0
8-bit Write Data, MSB first
SDA transitioning low to high while SCL is high. Signals the STOP condition

Figure 6. I^{2}C Signals - WRITE

Note: *Pinnacle is an I^{2}C Fast-Mode device. Timing information is available in the official I^{2}C specification from NXP, which can be found at this link: www.nxp.com/documents/user_manual/UM10204.pdf*

Register Access Protocol Using I^2C

To communicate with Pinnacle using I^2C requires RAP, which has only READ and WRITE commands with the byte format repeated in Table 22.

Table 22. Register Access Protocol

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
READ (0xAX)	1	0	1	Address 4	Address 3	Address 2	Address 1	Address 0
WRITE (0x8X)	1	0	0	Address 4	Address 3	Address 2	Address 1	Address 0

Reading a Register with I^2C

Pinnacle stores a 'Current Read Address' (CRA) for I^2C, which allows for efficient read operations. With I^2C, the Host sends an I^2C WRITE command to Pinnacle as the slave. The host must then send a RAP READ command to access Pinnacle registers. When the RAP READ command is received, Pinnacle stores the address it contains as the 'Current Read Address'. The host can then send I^2C READ commands that will start at the 'Current Read Address' and then automatically increment to the next address until a stop condition is received. When the stop condition is received, the 'Current Read Address' is reset to the address received in the original RAP READ command.

I^2C READ Example

Read the contents of register 0x12 (Button Status)

1. Send the start condition.
2. Send the I^2C WRITE command (7-bit slave address as top 7 bits and a 0 for WRITE). This means, shift the address (0x2A) one bit.

   ```
   0x2A << 1 = 0x54
   ```

3. Use the OR operand to combine the RAP READ command (0xAX) with the register address to be read. For this example, 0x12 is used.

   ```
   0xA0|0x12=0xB2
   ```

4. Send 0xB2.
5. Send a stop condition. Pinnacle sets the “Current Read Address” to 0x12 and is now setup to respond with the contents.
6. Send another start condition, followed by the I^2C READ command. (Pinnacle slave address shifted, and OR'd with a 1 to read.)

   ```
   0x2A   << 1=0x54
   0x54|0x01=0x55
   ```

7. Pinnacle will respond with the contents of register 0x12.
8. After the receiving all 8 bits, send a stop condition to end communication.

I^{2}C Multiple READ Example

Read the contents of register 0x14 through 0x16

1. Send the start condition.
2. Send the I^{2}C WRITE command. Shift address (0x2A) one bit.

   ```
   0x2A << 1 = 0x54
   ```

3. Use the OR operand to combine the RAP READ command (0xAX) with the first address to be read. For this example, 0x14 is used.

   ```
   0xA0|0x14 = 0xB4
   ```

4. Send 0xB4.
5. Send a stop condition.

   Pinnacle sets the Current Read Address (CRA) to 0x14 and is now setup to respond with the contents.

6. Send another start condition, followed by the I^{2}C READ command.

   (Pinnacle slave address shifted, and OR'd with a 1 to read.)

   ```
   0x2A << 1 = 0x54
   0x54 | 0x01 = 0x55
   ```

7. Pinnacle will respond with the contents of register 0x14 and increment CRA.
   - After the receiving all 8 bits, send another I^{2}C READ command (0x55). Pinnacle will respond with the contents of register 0x15 and increment CRA.
   - After the receiving all 8 bits, send another I^{2}C READ command (0x55). Pinnacle will respond with the contents of register 0x16 and increment CRA.
   - Send the stop condition to end communication. Pinnacle will reset the Current Read Address (CRA) to 0x14.

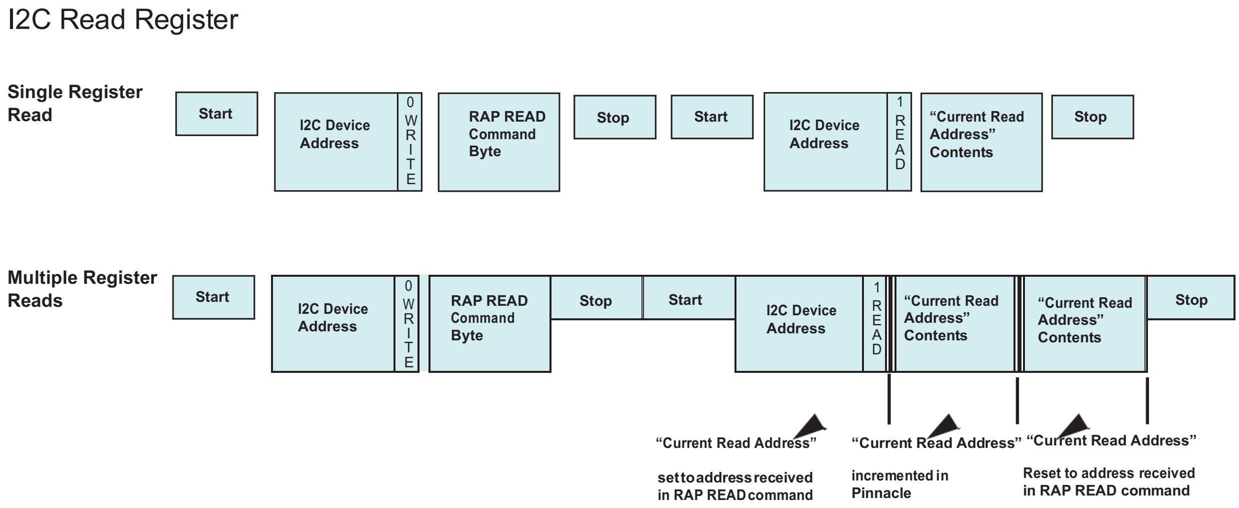


Figure 7. I^{2}C READ Register Sequence

Writing to a Register with I^2C

Writing to a register using I^2C requires sending two bytes, the WRITE Command Byte and the WRITE Value Byte. The master MUST NOT issue a stop condition between the WRITE Command Byte and the WRITE Value Byte (see Figure 8).

I^2C WRITE Example

WRITE value 0x02 to register 0x03 (SysConfig1 Shutdown Bit)

1. Send the START condition.
2. Send the I^2C WRITE command (7-bit slave address as top 7 bits and a 0 for WRITE)

   Shift address (0x2A) one bit.

   ```
   0x2A << 1 = 0x54
   ```

3. Use the OR operand to combine the RAP WRITE command (0x80) with the address to be written. For this example, 0x03 is used.

   ```
   0x80|0x03=0x83
   ```

4. Send 0x83.
5. Send the value 0x02.
6. Send the stop condition to end communication.

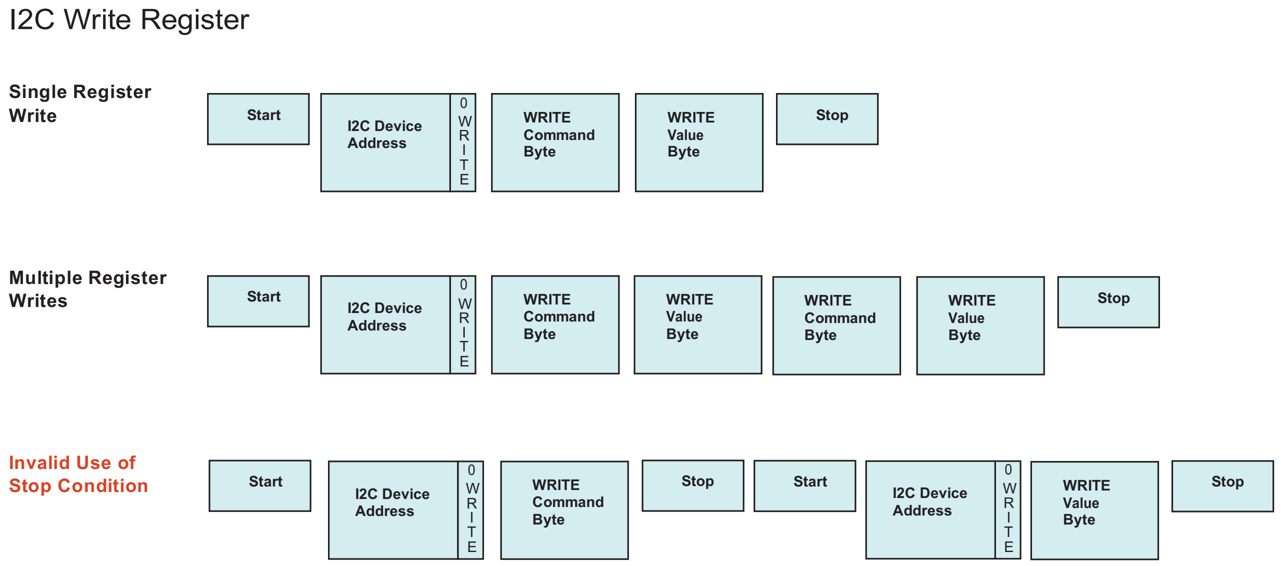


Figure 8. I^2C WRITE Register Sequence

PS/2 Protocol

Signal Definition

VDD:	Power
DATA:	Serial data line
CLK:	Clock line for data sampling
GND:	Ground

Data Stream Structure

Start bit:	1 bit (“L” level)
DATA bit:	8 bit (bit7=MSB ,bit0=LSB,positive logic)
Parity bit:	1 bit (Odd parity, positive logic) Stop bit 1 bit (“H” level)

Data Transmitting Timing

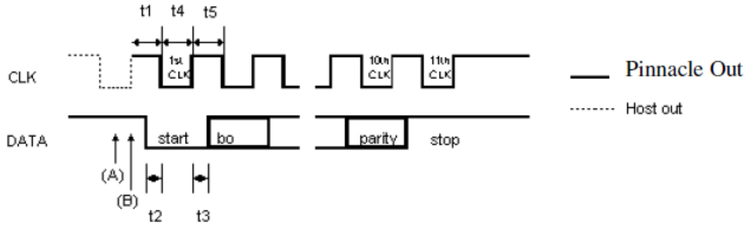


Figure 9: Data Transmitting Timing

Table 23. Data Transmitting Timing

Description	Minimum [US]	Maximum [US]
t1: Time from DATA transition “enable” to falling edge of CLK	70	200
t2: Time from DATA transition to falling edge of CLK	5	25
t3: Time from rising edge of CLK to DATA transition	5	25
t4: Time of CLK low	30	50
t5: Time of CLK High	30	50

- (A) PINNACLE checks the CLK line. When CLK line level is low, PINNACLE does not transmit.
- (B) When CLK line is high, DATA line is high and PINNACLE has output data, PINNACLE starts transmitting data to the system.

Data Receiving Timing

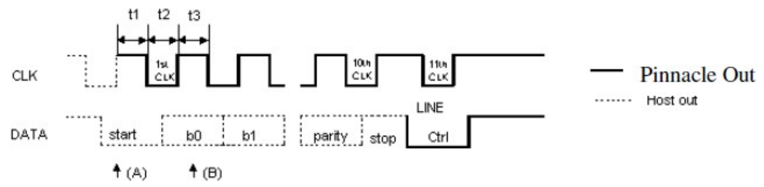


Figure 10: Data Receiving Timing

Table 24. Data Receiving Timing

Description	Minimum [US]	Maximum [US]
t1: Time from DATA receiving to falling edge of CLK	40	200
t2: Time of CLK low	30	50
t3: Time of CLK high	30	50

- (A) When the system changes the CLK line from low to high (triggered rising edge) and the DATA line is held as low level, PINNACLE starts receiving clock.
- (B) PINNACLE gets sampling data during clock high.

PS/2 Modes of Operation

Reset Mode

In this mode, a self-test is initiated during power on or by RESET command upon satisfactory completion of the diagnostics, a completion code (AAh) and an ID code (00h) are transmitted to the system. Then PINNACLE sets following defaults:

Sampling Rate:	10 ms
Scaling:	1:1
Mode:	Streaming mode
Resolution	245 x 1/2 cpi
Disable/Enable:	Disable

Stream Mode

In this mode, a data report is transmitted to the system if a switch is pressed or released, or if at least one count of movement has been detected. The maximum rate of transfer is the programmed sample rate. No transmissions occur if PINNACLE is motionless unless a switch is operated. In which case, the incremental movement report is zero.

Remote Mode

In this mode, data is transmitted only in response to the READ DATA command.

Wrap Mode

In this mode, any byte of data sent by the system, except ECh or FFh, is returned by PINNACLE.

Command Support List

Table 25. PINNACLE Command Support

PS/2 Command	Command	Mouse	Intellimouse	Absolute
Data Packet	------------	3-bytes	4-bytes	6-bytes
FF (Reset)	FF	Yes	Yes	Yes
FA (Ack)	Valid Command	Yes	Yes	Yes
FE (Resend)	FE	Yes	Yes	Yes
F6 (Set Default)	F6	Yes	Yes	Yes
F5 (Disable)	F5	Yes	Yes	Yes
F4 (Enable)	F4	Yes	Yes	Yes
F3,XX (Set Sample Rate)	F3,XX	Yes	Yes	No
F2 (Read Device ID)	F2	Yes	Yes	Yes
F0 (Set Remote Mode)	F0	Yes	Yes	No
EE (Set W rap Mode)	EE	Yes	Yes	Yes
EC (Reset W rap Mode)	EC	Yes	Yes	Yes
EB (Read Data)	EB	Yes	Yes	No
EA (Set Stream Mode)	EA	Yes	Yes	No
E9 (Status Request)	E9	Yes	Yes	Yes
E8,XX (Set Resolution)	E8,XX	Yes	Yes	No
E7 (Set Scaling 2:1)	E7	No	No	No
E6 (Reset Scaling)	E6	Yes	Yes	No
Invalid Command (1 byte)	F1,10, etc.	Yes	Yes	Yes
Invalid Command (2 byte)	F3,xx or E8,xx	Yes	Yes	Yes
Transition to IntelliMouse	F3,C8,F3,64,F3,50	Yes	No Change	No Change
Behavior when F6 is received	F6	Yes	Yes	Yes
Behavior when FF is received	FF	Yes	Yes	Yes

Finger Position Modes of Reporting

PINNACLE uses PS/2, SPI or I2C protocols for communication with the Host. Three modes are supported, but only 1 mode can be selected at a time. The modes are Mouse mode, Intellimouse mode, and Absolute mode. The data for the selected mode is stored in the RAM Register Map. The data packets are defined as follows:

- Mouse Mode
- IntelliMouse Mode
- Absolute Mode

Mouse Mode

After Power On Reset or receiving a Reset command from a system, PINNACLE is in Mouse mode. In this mode, PINNACLE emulates a standard Microsoft mouse. Taps are reported the same as the buttons.

Table 26. Mouse Packet (8 bits of X and Y counts, +127, -128)

Register	Byte Number	Bit	Description
PACKETBYTE_2	2	0-7	Y count data (bit0=LSB)
PACKETBYTE_1	1	0-7	X count data (bit0=LSB)
PACKETBYTE_0	0	7	0
		6	0
		5	Y data sign (1=negative)
		4	X data sign (1=negative)
		3	Always 1
		2	1=Middle button pressed, 0=Middle button released
		1	1=Right button (tap) pressed, 0=Right button (tap) released
		0	1=Left button (tap) pressed, 0=Left button (tap) released

IntelliMouse Mode

PINNACLE has a IntelliMouse emulation mode that is enabled by an HCO resistor. The command sequence of F3h,C8h,F3h,64h,F3h,50h must also be received from the driver. Afterwards, if PINNACLE receives an F2h command, it must reply FAh,03h.

Table 27. IntelliMouse Packet (8 bits of X and Y counts, +127, -128)

Register	Byte Number	Bit	Description
PACKETBYTE_3	3	0-7	Wheel count (bit0=LSB)
PACKETBYTE_2	2	0-7	Y count data (bit0=LSB)
PACKETBYTE_1	1	0-7	X count data (bit0=LSB)
PACKETBYTE_0	0	7	0
		6	0
		5	Y data sign (1=negative)
		4	X data sign (1=negative)

Table 27. IntelliMouse Packet (8 bits of X and Y counts, +127, -128) - (continued)

Register	Byte Number	Bit	Description
		3	Always 1
		2	1=Middle button pressed, 0=Middle button released
		1	1=Right button (tap) pressed, 0=Right button (tap) released
		0	1=Left button (tap) pressed, 0=Left button (tap) released
			1=Tap On, 0=Tap Off

Note: *byte four (4) is in binary and negative values are expressed in two's complement. Wheel count: -128 ~ +127. The internal accumulator of wheel count must not wrap around.*

Absolute Mode

Absolute mode reports X,Y positions and Z (finger signal strength) data to the system. There are 2 optional methods to enter Absolute mode:

- Method 1) Set the DATA_TYPE_REL0_ABS1 flag in the FEEDCONFIG1 Register to '1'.
- Method 2) Set the Absolute mode by hardware (HCO). The PS/2 stream is enabled bydefault.

Table 28. PS/2 Absolute Data Packet Format

	b7	b6	b5	b4	b3	b2	b1	b0
Byte0	1	0	X2	X1	X0	1	1	1
Byte1	X10	X9	X8	X7	X6	X5	X4	X3
Byte2	Y10	Y9	Y8	Y7	Y6	Y5	Y4	Y3
Byte3	1	1	Y2	Y1	Y0	SW M	SW R	SW L
Byte4	SW 6	SW 5	SW 4	P4	P3	P2	P1	P0
Byte5	PS	Touch	Z5	Z4	Z3	Z2	Z1	Z0

Table 29. SPI or I2C Absolute Data Packet Format

Register	b7	b6	b5	b4	b3	b2	b1	b0
PACKETBYTE_0	0	0	X2	X1	X0	1	1	1
PACKETBYTE_1	0	0	PS	P4	P3	P2	P1	P0
PACKETBYTE_2	X7	X6	X5	X4	X3	X2	X1	X0
PACKETBYTE_3	Y7	Y6	Y5	Y4	Y3	Y2	Y1	Y0
PACKETBYTE_4	Y11	Y10	Y9	Y8	X11	X10	X9	X8
PACKETBYTE_5	Touch	0	Z5	Z4	Z3	Z2	Z1	Z0

SW L is primary button status	0=off 1=on
SW R is secondary button status	0=off 1=on
SW M is auxiliary button status	0=off 1=on

SW 4 is 4th button status	0=off 1=on
SW 5 is 5th button status	0=off 1=on
SW 6 is 6th button status	0=off 1=on
X10-X0 are X coordinates (X10=MSB).	Range of X position: 0-2047
Y10-Y0 are Y coordinates (Y10=MSB).	Range of Y position: 0-1535
-Z5-Z0 are Z value (Z5=MSB). Z increases by two's:	0,2,...126.
-PS, Pnare palm detection data	
Touch flag	0= no_ink 1= ink

Note: *When using SPI or I2C protocols, the host can read 4-bytes, 5-bytes, or 6-bytes of data, depending on the application needs. For example, if only the X,Y,Z information is needed, then the host only needs to read Registers PACKETBYTE2~5.*

Z_Idle

Z_idle is a function in Absolute mode only. It is necessary to provide more data until PINNACLE can determine if a tap or drag is happening. The number of packets for Z_idle is controlled by a register called ZIDLE. ZIDLE begins when Z=0 and the trackpad is ready to enter Idle mode. Both touches and button presses use ZIDLE in the same manner.

If ZIDLE = 0, then no extra packets are sent.
If ZIDLE is 1~FF hex, then the number of extra packets is 1~FF accordingly. The default ZIDLE value is 1E hex (30 decimal.

Palm Detection method.

There are imbalances created by the size and shape of an object on the trackpad. This information can be used by a driver and can be combined with other touch and tracking information to avoid undesirable behaviors such as movement or tapping when the palm rests on the trackpad.

Hardware Configurable Options

Hardware Configurable Options (HCO) modify the behavior of PINNACLE. HCOs are activated by soldering 470 k resistors between adjacent electrodes on the circuit board. After Power On Reset, PINNACLE reads the HCOs and activates the chosen behaviors. The options are described in Table 30 below.

Table 30. Hardware Configurable Options

HCO	Description*	Comments				Location (48-PIN Package)
X0	ID3					pin 17 to pin 18
X1	ID4					pin 18 to pin 19
X2	ID5					pin 19 to pin 20
X3	Y axis Disable					pin 20 to pin 21
X4	X axis Disable					pin 21 to pin 22
X5	Use 6 Y electrodes					pin 22 to pin 23
X6	Absolute Mode					pin 23 to pin 24
X7	Use 8 X electrodes					pin 24 to pin 25
X8	All Taps Disable					pin 25 to pin 26
X9	Configuration	For Cirque default settings				pin 26 to pin 27
X10	Package detect	Only for 32-pin package				NA
X11	Swap XY relative					pin 28 to pin 29
X12	ID0					pin 29 to pin 30
X13	ID1					pin 30 to pin 31
X14	ID2					pin 31 to pin 32
X15						NA
Y0	X axis invert					pin 5 to pin 6
Y1	Y axis invert					pin 6 to pin 7
Y2	Right (secondary) Tap Enable	If X9 HCO is not populated				pin 7 to pin 8
Y2	Right (secondary) Tap Disable	If X9 HCO is populated				pin 7 to pin 8
Y3	Intellimouse seq. Enable	If X9 HCO is not populated				pin 8 to pin 9
Y3	Intellimouse seq. Disable	If X9 HCO is populated				pin 8 to pin 9
Y4	Button Scroll Enable					pin 9 to pin 10
Y5	180 orientation	If X9 HCO is not populated				pin 10 to pin 11
Y5	Not 180 orientation	If X9 HCO is populated				pin 10 to pin 11
Y6	Mode 0 (48-pin package)	**PS/2**	**Dual**	**I2C**	**SPI**	
		0	0	1	1	pin 11 to pin 12
Y7	Mode 1 (48 pin package)	0	1	0	1	pin 12 to pin 13
Y8	I2C Address value	0x2A	0x2B	0x2C	0x2D	
	I2C Address 0 / SPI Filler Disable	0	0	1	1	pin 13 to pin 14
Y9	I2C Address 1	0	1	0	1	pin 14 to pin 15
Y10	ID6					pin 15 to pin 16

Note: *Use X9 HCO for Cirque defaults (Right Taps, Intellimouse scroll, 180 orientation, GlideExtend always)*

Register Descriptions

Firmware ID

This register contains the Identification number of the ASIC firmware.

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
ID 7	ID 6	ID 5	ID 4	ID 3	ID 2	ID 1	ID 0

Firmware Version

This register contains the firmware revision number.

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Ver 7	Ver 6	Ver 5	Ver 4	Ver 3	Ver 2	Ver 1	Ver 0

Status 1

This register status flags about chip state.

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Reserved	Reserved	Reserved	Reserved	Command Complete	SW Data Ready	Reserved	Reserved

SysConfig 1

This register contains status flags about chip state.

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Reserved	Reserved	GPIO Control Enable	AnyMeas enable	Track disable	Sleep	Shutdown	Reset

FeedConfig 1

This register contains flags that control the data output.

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Y data Invert	X data Invert	Axis for Z Y0_X1	Y axis Disable	X axis Disable	Filter Disable	Absolute or Relative	Feed Enable

FeedConfig 2

This register contains flags for trackpad features and modes.

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Swap XY relative	Button Scroll Enable	Palm before Z Enable	GlideExtend Disable	Scroll Disable	Secondary Tap Disable	All Taps Disable	Intellimouse Seq Enable

FeedConfig 3

.

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
				Disable Noise Avoidance	Disable Palm/ NERD measurements	Disable Smoothing	DualPoint Buttons

CalConfig 1

This register contains configuration bits for the feed state. Default value is 01h.

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
RESERVED	RESERVED	RESERVED	Enable Dynamic Tap Comp	Enable Dynamic Track Error Comp	Enable Dynamic NERD Comp	Enable Dynamic Background Comp	Calibrate

Sample Rate

This register contains the number of feed samples created per second.

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
B7	B6	B5	B4	B3	B2	B1	B0

Default: 64H (100 samples/second)

The following sample rate values are supported:

Table 31. Sample Rate - Register Value

Sample Rate	Register Value
100 Sample/Second	64H
80 Sample/Second	50H
60 Sample/Second	3CH
40 Sample/Second	28H
20 Sample/Second	14H
10 Sample/Second	0Ah

Writing any other value to the Sample Rate Register will cause the sample rate to be set to the default of 64H (100 samples/second).

Z-Idle

This register contains the number of packets with x,y,z = 0 sent when there is a transition from tracking (Z > 0) to not tracking (Z = 0). Default value is 1Eh

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
B7	B6	B5	B4	B3	B2	B1	B0

Packet Byte 0

This register contains data. See desired packet format for details.

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
X7	X6	X5	X4	X3	X2	X1	X0

Packet Byte 1

This register contains data. See desired packet format for details.

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
X7	X6	X5	X4	X3	X2	X1	X0

Packet Byte 2

This register contains data. See desired packet format for details.

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
X7	X6	X5	X4	X3	X2	X1	X0

Packet Byte 3

This register contains data. See desired packet format for details.

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
X7	X6	X5	X4	X3	X2	X1	X0

Packet Byte 4

This register contains data. See desired packet format for details.

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
X7	X6	X5	X4	X3	X2	X1	X0

Packet Byte 5

This register contains data. See desired packet format for details.

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
X7	X6	X5	X4	X3	X2	X1	X0

Port A GPIO Control

.

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PA7	PA6	PA5	PA4	PA3	PA2	PA1	PA0

- Write these bits to 0 for the GPIO output Low state.
- Write these bits to 1 for the GPIO pull-up (125 k ohms) High state

Port A GPIO DATA

.

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PA7	PA6	PA5	PA4	PA3	PA2	PA1	PA0

- Read these bits to know the status of the GPIO pins.

Port B GPIO Control and Data

.

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
RESERVED	RESERVED	PB2 Control	PB1 Control	PB0 Control	PB2 Data	PB1 Data	PB0 Data

- Write the control bits: 0 for GPIO output Low state. 1 for GPIO pull-up (10 k ohms) High state.
- Read the data bits to know the status of these GPIO pins.

PS/2 Aux Port Control

.

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Aux Port Status	RESERVED	SP Coordinate Disable	GP Coordinate Disable	SP Disable	GP Disable	SP Extended Mode	Command Pass-Thru Enable

HCO ID

This register contains the HCO ID information to identify modules to the driver. B7 is set when the Configuration HCO is populated (Cirque default conditions)..

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
B 7	ID 6	ID 5	ID 4	ID 3	ID 2	ID 1	ID 0

Appendix: Extended Register Access

Using four standard RAP registers, the host can gain Extended Register Access (ERA) to Pinnacle memory. Register 0x1B is used for the value to be written or read (see Table 32). Register 0x1C is the high byte and register 0x1D is the low byte of the address to be read or written to (see Table 33 and Table 34). Register 0x1E, the ERA control register, specifies READ or WRITE and the optional Auto-Incremented READ or WRITE for sequential commands, as well as a WRITE/Verify option if both the READ and WRITE flags are set (see Table 23 on page 45). The control register value returns to 0x00 to indicate a command is complete. Use standard Register Access Protocol (RAP) to access these four registers.

It is important to note that accessing the extended registers will assert the command complete (SW_CC) flag and force the hardware data ready (HW_DR) pin high. The Pinnacle data feed should be disabled before accessing extended registers, and SW_CC should be cleared when the host is finished accessing extended registers.

Table 32. Extended Register Access Value - (RAP register 0x1B)

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Description	Value 7	Value 6	Value 5	Value 4	Value 3	Value 2	Value 1	Value 0
Read/Write	R/W							
Values	0x00 - 0xFF							
Default	0							

Table 33. Extended Register Access Address High Byte - (RAP register 0x1C)

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Description	Address15	Address14	Address13	Address12	Address11	Address10	Address9	Address8
Read/Write	R/W							
Values	0x00 - 0xFF							
Default	0							

Table 34. Extended Register Access Address Low Byte - (RAP register 0x1D)

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Description	Address 7	Address 6	Address 5	Address 4	Address 3	Address 2	Address 1	Address 0
Read/Write	R/W							
Values	0x00 - 0xFF							
Default	0							

Table 35. Extended Register Access Control - (RAP register 0x1E)

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Description					WRITE Auto-Increment	READ Auto-Increment	Write	Read
Read/Write					R/W	R/W	R/W	R/W
Values					1=enabled 0=disabled	1=enabled 0=disabled	1 = WRITE	1 = Read
Default					0	0	0	0

This register value returns to 0x00 to indicate a completed command

Asserting both Bit[1] and Bit[0] indicates a WRITE/Verify

Extended Register Access Examples

Using standard RAP to send READ and WRITE commands to Pinnacle, the following examples demonstrate the proper sequence to read and write to Pinnacle's extended registers.

Example: READ an Extended Register

1. WRITE the high byte of the 16-bit extended register address to RAP Register 0x1C (ERA High Byte).
2. WRITE the low byte of the 16-bit extended register address to RAP Register 0x1D (ERA Low Byte).
3. WRITE 0x01 (ERA READ flag) to RAP Register 0x1E (ERA Control).
4. READ the RAP Register 0x1E (ERA Control) until it contains 0x00.
5. READ the new value in RAP Register 0x1B (ERA Value).
6. WRITE 0x00 to RAP Register 0x02 (Status1) to clear Command Complete (SW_CC).

Example: READ an Extended Register with Address Increment

1. WRITE the high byte of the 16-bit extended register address to RAP Register 0x1C (ERA High Byte).
2. WRITE the low byte of the 16-bit extended register address to RAP Register 0x1D (ERA Low Byte).
3. WRITE 0x05 to RAP Register 0x1E (ERA Control) to specify auto-increment read.

   Repeat 4, 5, and 6 as needed

4. READ the RAP Register 0x1E (ERA Control) until it contains 0x00.
5. READ the new value in RAP Register 0x1B (ERA Value).
6. WRITE 0x00 to RAP Register 0x02 (Status1) to clear Command Complete (SW_CC).

   Extended Register Address is incremented. Repeat steps 4, 5, and 6 to reach the desired address.

Example: WRITE to an Extended Register

1. WRITE the value to be written in RAP Register 0x1B (ERA Value)
2. WRITE the high byte of the 16-bit extended register address to RAP Register 0x1C (ERA High Byte).
3. WRITE the low byte of the 16-bit extended register address to RAP Register 0x1D (ERA Low Byte).
4. WRITE 0x02 (ERA WRITE flag) to RAP Register 0x1E (ERA Control).
5. READ the RAP Register 0x1E (ERA Control) until it contains 0x00.
6. WRITE 0x00 to RAP Register 0x02 (Status1) to clear Command Complete (SW_CC).

Example: WRITE to an Extended Register with Address Increment

1. WRITE the value to be written in RAP Register 0x1B (ERA Value)
2. WRITE the high byte of the 16-bit extended register address to RAP Register 0x1C (ERA High Byte).
3. WRITE the low byte of the 16-bit extended register address to RAP Register 0x1D (ERA Low Byte).
4. WRITE 0x0A (ERA auto-increment WRITE) to RAP Register 0x1E (ERA Control).

   Repeat Steps 5 and 6 as needed

5. READ the RAP Register 0x1E (ERA Control) until it contains 0x00.
6. WRITE 0x00 to RAP Register 0x02 (Status1) to clear Command Complete (SW_CC).

   Extended Register Address is incremented. Repeat steps 5 and 6 to reach the desired address.

Example: WRITE to an Extended Register with Verification

1. WRITE the value to be written in RAP Register 0x1B (ERA Value)
2. WRITE the high byte of the 16-bit extended register address to RAP Register 0x1C (ERA High Byte).
3. WRITE the low byte of the 16-bit extended register address to RAP Register 0x1D (ERA Low Byte).
4. WRITE 0x03 (write and read) to RAP Register 0x1E (ERA Control).
5. READ the RAP Register 0x1E (ERA Control) until it contains 0x00.
6. READ the value in RAP Register 0x1B (ERA Value) to verify.
7. WRITE 0x00 to RAP Register 0x02 (Status1) to clear Command Complete (SW_CC).

Example Code

Example code for reading and writing extended registers is available on Cirque's online code repository located at GitHub. Follow the link below for access:

https://github.com/cirque-corp/Pinnacle_1CA027/blob/master/Circular_Trackpad/TM040040/Example_Code/SPI_Demo/TM040040_SPI.ino

Pinnacle Mechanical Specifications

Pinnacle is available in a 48-pin TQFP package. The mechanical details of the Pinnacle ASIC are shown below.

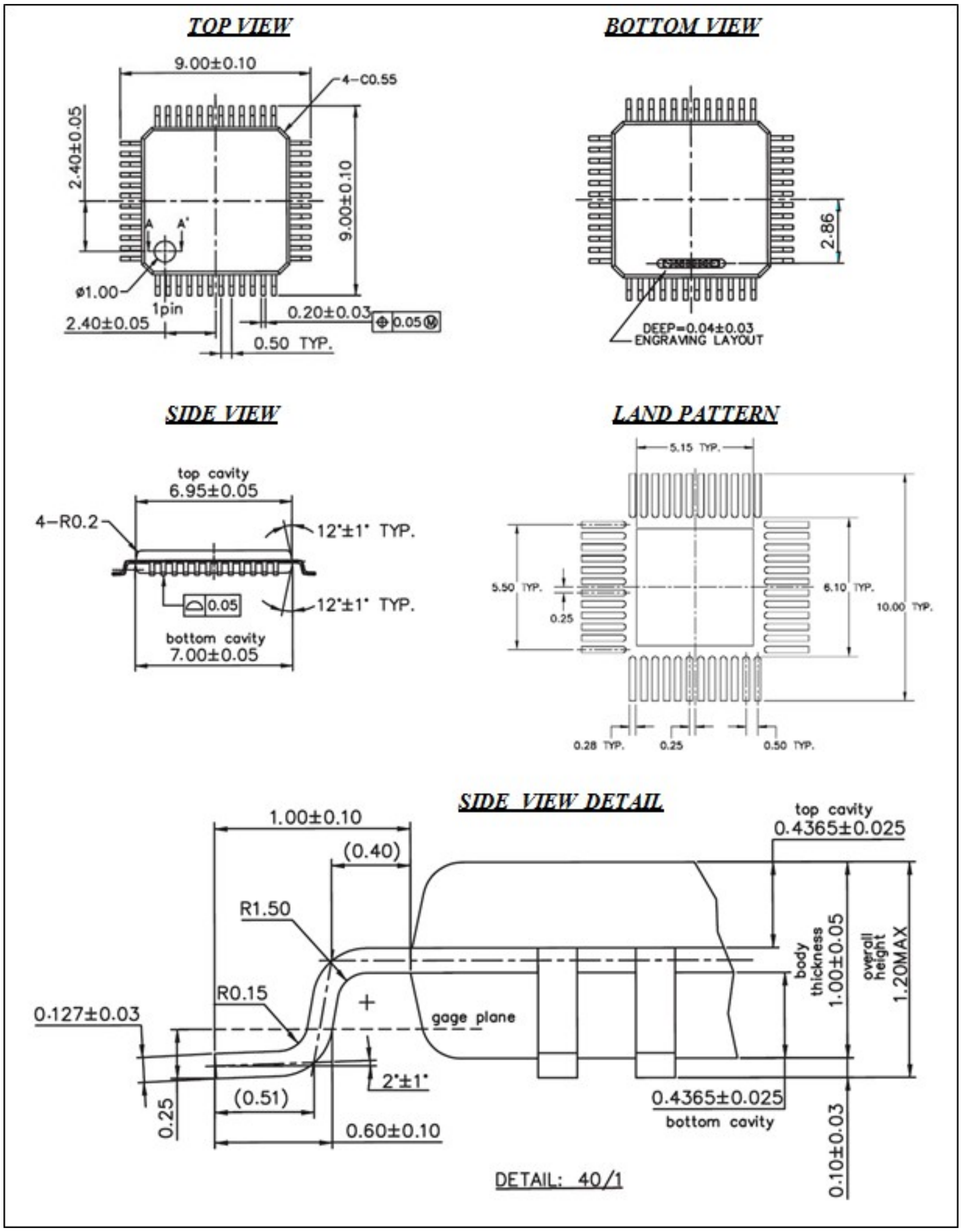
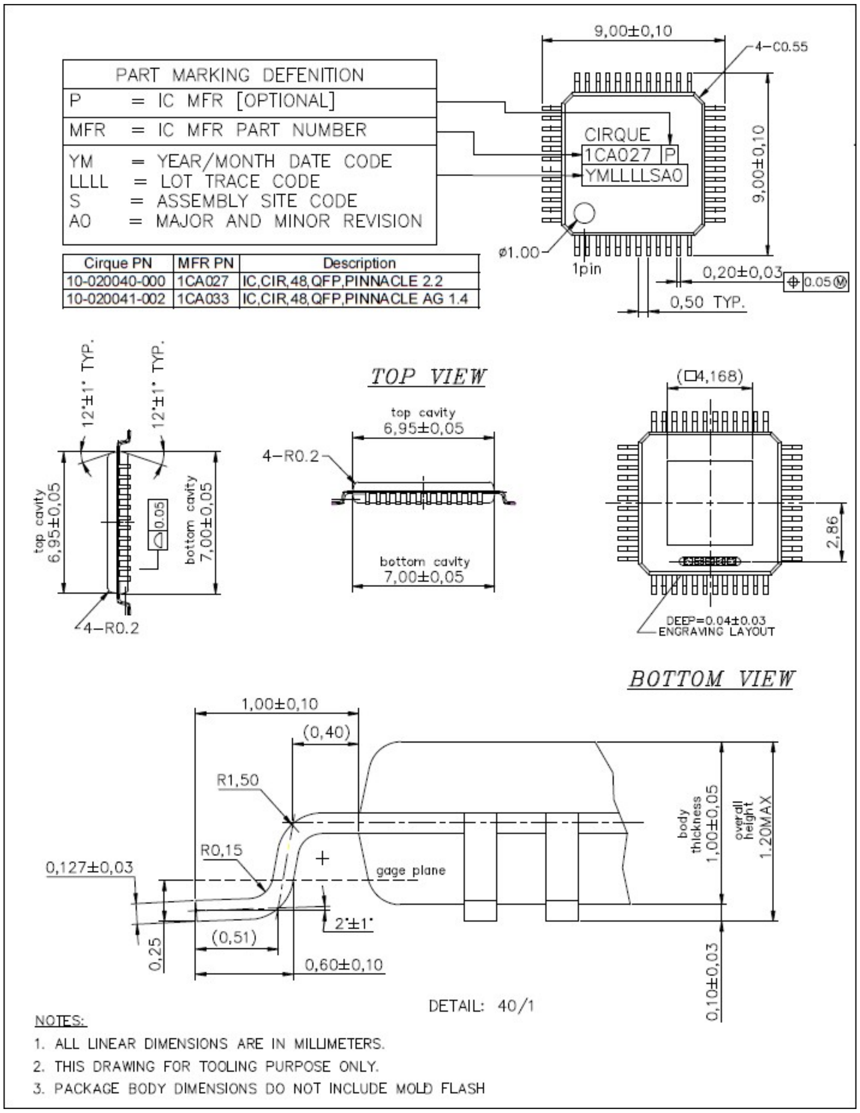


Figure 11: Cirque Pinnacle ASIC 48-Pin QFN Package Mechanical Drawing

Part Marking



PART MARKING DEFENITION		
P	=	IC MFR [OPTIONAL]
MFR	=	IC MFR PART NUMBER
YM	=	YEAR/MONTH DATE CODE
LLLL	=	LOT TRACE CODE
S	=	ASSEMBLY SITE CODE
AO	=	MAJOR AND MINOR REVISION

Cirque PN	MFR PN	Description
10-020040-000	1CA027	IC,CIR,48,QFP,PINNACLE 2.2
10-020041-002	1CA033	IC,CIR,48,QFP,PINNACLE AG 1.4

NOTES:

1. ALL LINEAR DIMENSIONS ARE IN MILLIMETERS.
2. THIS DRAWING FOR TOOLING PURPOSE ONLY.
3. PACKAGE BODY DIMENSIONS DO NOT INCLUDE MOLD FLASH

Figure 12: Cirque Pinnacle ASIC Part Markings

Reference Design

Cirque's TCM5030 is optimized as a stand-alone small footprint layout. This design follows PCB layout rules for integration of Cirque IC technology into a customer design. The sense traces (SNSP and SNSN) are ground isolated from external signals in the connection routing path (treat both signals similar to coaxial cable isolation). TCM5030 is available upon request as a reference driver PCB sample or as a reference design layout Gerber data.

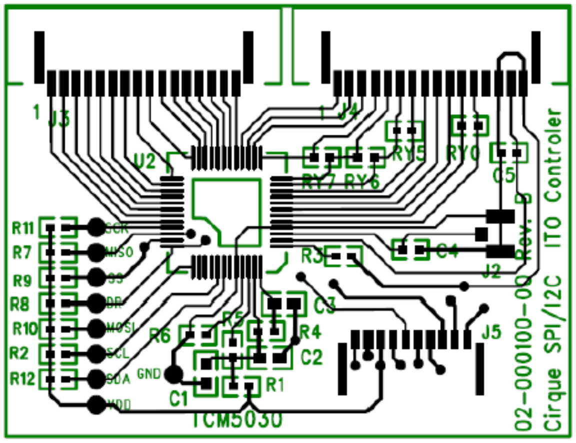


Figure 13: TCM5030 Reference Design PCB Layout Example - Layer 1

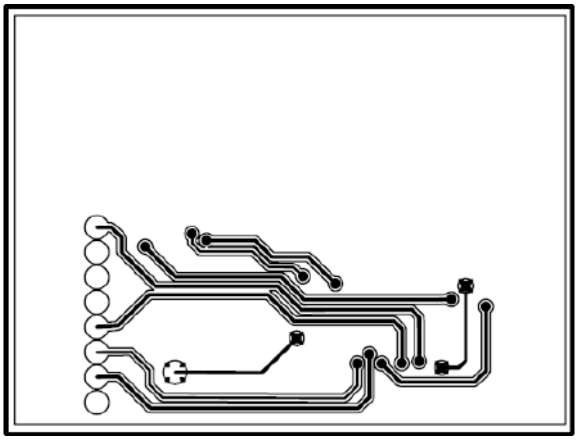


Figure 14: TCM5030 Reference Design PCB Layout Example - Layer 2

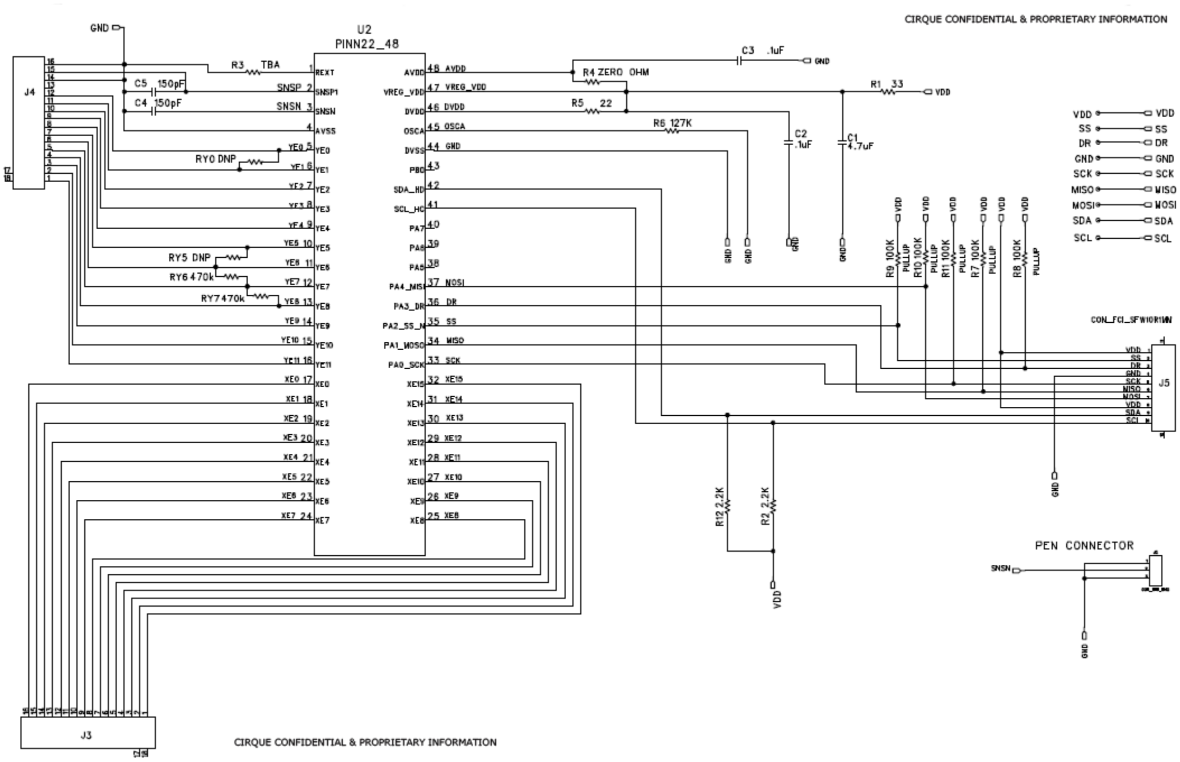


Figure 15: Reference Design (Cirque TCM5030 Schematic)

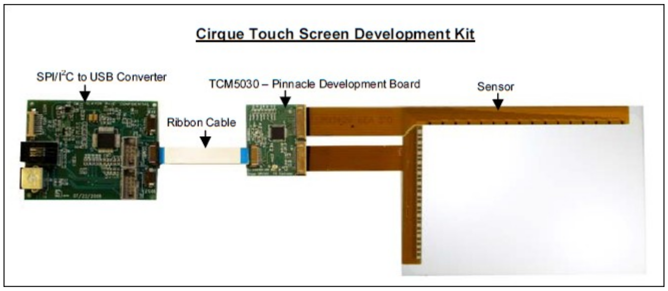

Cirque Touch Screen Development Kit
SPI/I2C to USB Converter
Ribbon Cable
TCM5030 – Pinnacle Development Board
Sensor

Figure 16: Cirque Touch Screen Development Kit

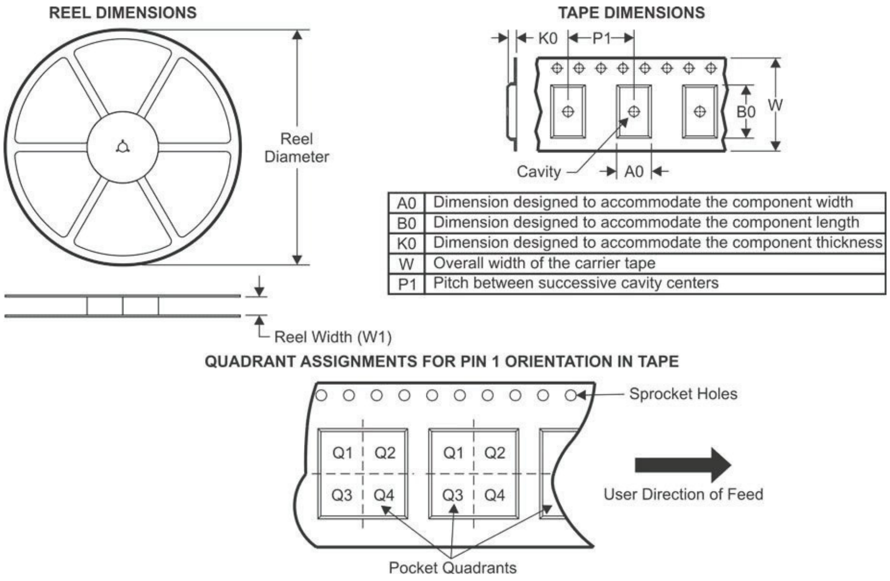
Packaging Information

The PINNACLE ASIC is shipped on tape, stored on a reel, and shipped in boxes. This section covers the information about the reels and boxes that are used when sending PINNACLE.

PINNACLE Information

This section provides the tape, reel, and box dimensions. The quantity for each reel is 1,000 pcs. The tape and reel information is in the figure below and the box information is in the figure on the next page.

Tape, Reel, and Box Dimensions



A0	Dimension designed to accommodate the component width
B0	Dimension designed to accommodate the component length
K0	Dimension designed to accommodate the component thickness
W	Overall width of the carrier tape
P1	Pitch between successive cavity centers

Figure 17: PINNACLE Tape and Reel Information

Table 36. Tape and Reel Information

Device	Package Type	Package Drawing	Pins	SPQ	Reel Diameter (mm)	Reel Width W1 (mm)	A0 (mm)	B0 (mm)	K0 (mm)	P1 (mm)	W (mm)	Pin 1 Quadrant
10-020040-xxx OR 10-020041-xxx	TQFP	PFB	48	1000	330.0	16.8	9.6	9.6	1.5	12.0	12.0	Q2

Note: *All dimensions are nominal.*

Reel Box Dimensions

The Reels are placed in a box. This section provides the box dimensions.

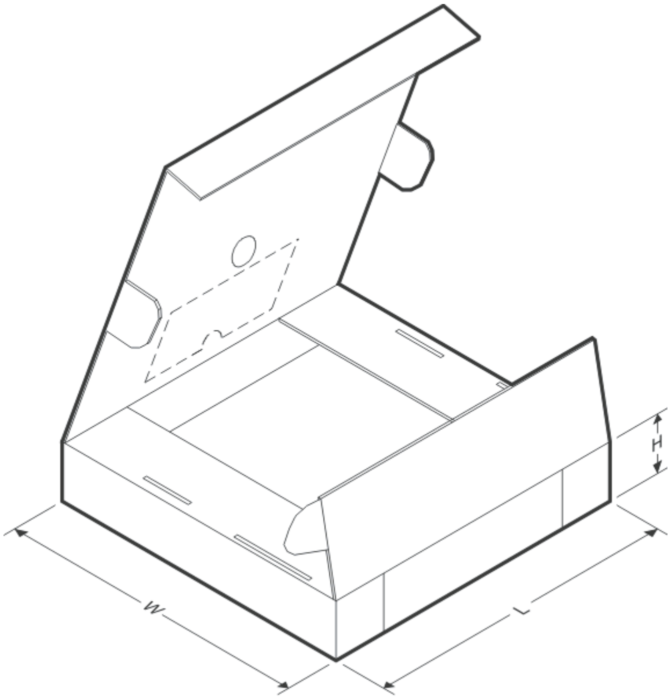


Figure 18: Box Dimensions

Table 37: Tape and Reel Information

Device	Package Type	Package Drawing	Pins	SPQ	Length (mm)	Width (mm)	Height (mm)
10-020040-xxx	TQFP	PFB	48	1000	362	362	33
10-020041-xxx	TQFP	PFB	48	1000	350	343	35

Note: *All dimensions are nominal.*

Shipping Box Information

This section provides information about the box in which the reel boxes will be sent to customers. The shipping box will be a corrugated fiberboard container. The size of the box is determined by the number of reel boxes. Filler, such as cushion, is added if space exists inside the shipping box after the reel boxes have been added.

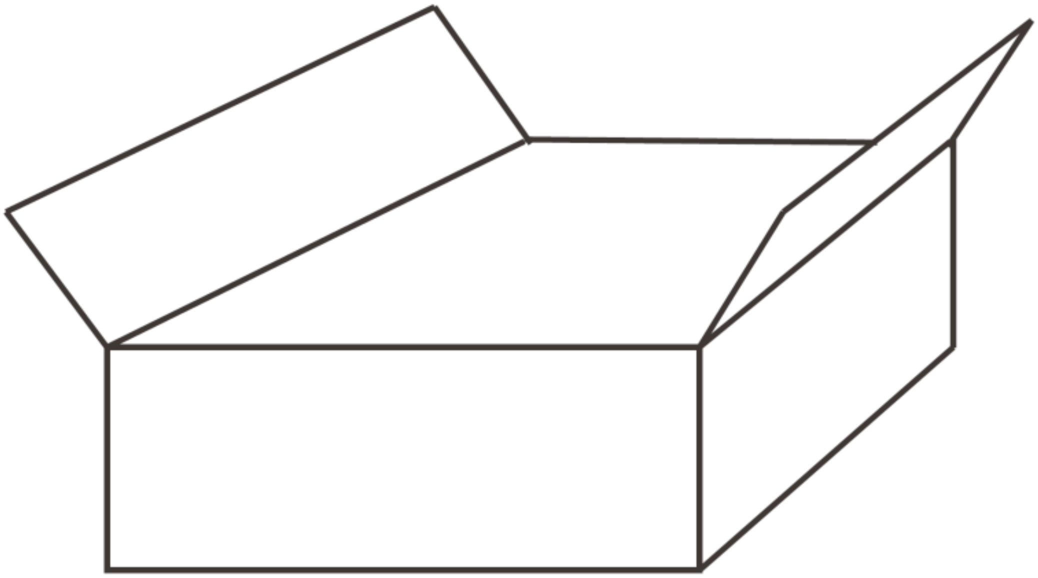


Figure 19: Example Shipping Box

Contact Information

Contact a Cirque sales representative for a complete list of Cirque's OEM products.

In United States & Canada	(800) GLIDE-75 (454-3375)
Outside US & Canada	(801) 467-1100
Fax	(801) 467-0208
Web site	http://www.cirque.com

THIS INFORMATION IS PROVIDED “AS IS.” CIRQUE SPECIFICALLY DISCLAIMS ALL WARRANTIES EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO, IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE WITH RESPECT TO THE DOCUMENTATION AND USE OF THE DOCUMENTATION IN DESIGN OF ANY PRODUCTS OR FOR ANY PARTICULAR APPLICATION.