DRAFT DOCUMENT

The feature has 3 functions and 1 events:

[0x6100]TouchPadRawXY

GetTouchPadInfo   = [0]GetTouchPadInfo()
GetRawReportState   = [1]GetRawReportState()
void   = [2]SetRawReportState(uint8_t state)

[0x6100] EVENT TouchPadRawXY

DualXYData   = [0]TwoFingersDataReport()

[0x6100] FUNCT GetTouchpadInfo()

Summary
Returns the TouchPad characteristics (size, number of dots, data ranges, etc)

Parameters
none

Returns

byte num (in payload)	description	remarks
0	X size (dots) H	max pixels reported MSB
1	X size (dots) L	max pixels reported LSB
2	Y size (dots) H	max pixels reported MSB
3	Y size (dots) L	max pixels reported LSB
4	Z data range (size)	0x00 means no range 0x0F means 16bits
5	Area data range (size)	0x0F means 16bits
6	Timestamps unit given in 0.1ms	A value of 10 means timestamps are in units of 1 mSec See **Note 1)**
7	Max number of fingers	
8	Position of the origin	0x00 = reserved 0x01 = LOWER-LEFT 0x02 = LOWER-RIGHT 0x03 = UPPER-LEFT 0x04 = UPPER-RIGHT Note: corners are defined by looking at device from above, with lower edge toward the user and upper facing the PC screen
9	RESERVED	
10	RESERVED	
11	RESERVED	
12	RESERVED	
13	DPI_H	Touchpad Resolution MSB (Same resolution in X and Y)
14	DPI_L	Touchpad Resolution LSB (Same resolution in X and Y)
15	RESERVED	

Note 1) Exception: A device that reports 8 in this field has a timestamp unit of 1ms (i.e field value equivalent to 10). There is no support for devices with 0.8 mSec timestamp unit.

Errors None

[0x6100] FUNCT GetRawReportState

Summary
Returns the TouchPad raw reporting state byte.

Parameters
none

Returns

bit 7	6	5	4	3	2	1	0
RSVD	RSVD	RSVD	raw and native	width/height 4-bit reporting	enhance	add force data on 16bits	enable raw

Byte 0: State

[0x6100] FUNCT SetRawReportState

Summary
Sets the TouchPad raw reporting state byte.

Parameters
Byte 0: state

Returns

bit 7	6	5	4	3	2	1	0
RSVD	RSVD	RSVD	eaw and native	width/height 4-bit reporting	enhance	add Z info on 16bits	enable raw

none

Reporting state details:

bit0: enable raw reporting
  0 = DISABLED (Touchpad reports on standard HID pipe. Gestures are processed in the touchpad)
  1 = ENABLED: (Touchpad reports raw data on HIDPP pipe, nothing is sent on the standard HID pipe. No gesture processing is done by the touchpad)

bit1: (add Z info on 16bits)
  0 = DISABLED (Area byte in report contains the contact area. Z is reported as 8 bit value)
  1 = ENABLED (Area byte in report contains the MSB of a 16bit Z value. No area is reported)

bit2: - Enhance
  0 = DISABLED (Normal settings for sensor)
  1 = ENABLED (Enhanced sensibility for sensor)

bit3: - Width/Height reporting (4-bit each) instead of Area

0 = DISABLED
1 = ENABLED (report width/height)

bit4: - Send RAW data as well as recognized gestures/tracking/scrolling messages. (dual mode)
0 = DISABLED
1 = ENABLED (send both, raw data on hidpp pipe and standard data on hid channel)
This mode may degrade the tracking smoothness.

Returns
 none

EVENT [0] format:
[0x6100] Event TouchPadRawTouchPoints

Summary
Sends data for two fingers, out of the N currently used. Sent by chunks of 2 fingers.

Parameters
none
Returns
DualXYData   [16 x 8bits]

byte num (in payload)	description	covers
0	Timestamp (MSB)	tags this packet (both touch points)
1	Timestamp (LSB)	
2	**2b** Contact type **6b** X (MSB)	touch point 1
3	X (LSB)	
4	**2b** Contact status **6b** Y (MSB)	
5	Y (LSB)	
6	Z/Force	
7	Area or (width + height)	
8	**4b** Finger ID **1b** proximity detection **1b** mechanical button **1b** Spurious Flag **1b** End-of-Frame (*)	
9	**2b** Contact type **6b** X (MSB)	touch point 2
10	X (LSB)	
11	**2b** Contact status **6b** Y (MSB)	
12	Y (LSB)	
13	Z/Force	
14	Area	
15	**4b** Finger ID **4b** Total number of fingers in this frame	

CONTACT TYPE:
00 = finger
01 = reserved
10 = reserved
11 = reserved

CONTACT STATUS (**2bits**)
00 = no finger (released)
01 = touch
10 = hover
11 = reserved

Note: in order to release finger the device will send an empty report (ie with all data to 0).
It is important to note that X=0, Y=0, Force=0, Area=0 are still valid values, therefore the host software (driver) should check for CONTACT_STATUS = 0x00 to assess the release of that finger.

In case more than 2 fingers are present, two or more event will be sent to complete the frame:

The **timestamp** will be the same (it represents the timestamp of the moment when the frame is captured)
The **total number of fingers** (in the frame) is repeated in all messages of the frame

The last message of the frame will be "tagged" with the **1b End-of-Frame** bit set to 1.