



TrackPoint Engineering Specification Version 4.0

Serial Supplement

B. Olyha CSS Electronic Engineering
J. Rutledge Mathematical Sciences

Originator Contacts

Bob Olyha
IBM T. J. Watson Research Center
Route 134
P.O. Box 218 M.S. 31-123
Yorktown Heights, NY 10598-0218
U.S.A.

Phone (914) 945-3210 [IBM tieline 862-3210]
FAX (914) 945-4148 [IBM tieline 862-4148]
internet: rolyha@us.ibm.com

Joe Rutledge
IBM T. J. Watson Research Center
Route 134
P.O. Box 218 M.S. 31-250
Yorktown Heights, NY 10598-0218
U.S.A.

Phone (914) 945-1175 [IBM tieline 862-1175]
FAX (914) 945-3434 [IBM tieline 862-3434]
internet: rutledg@us.ibm.com

INFORMATION IN THIS DOCUMENT IS PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT.

Information in this document may contain technical inaccuracies or typographical errors. Information may be changed or updated without notice. IBM may also make improvements and/or changes in the specifications at any time without notice. IBM assumes no liability with respect to the patents of third parties. IBM U.S. patent numbers 5,570,111; 5,579,033, and Japanese patent number 2074941 apply to TrackPoint. Additional patents are pending.

YKT3C.lwp	EC YKTB5	EC YKTB8	EC YKT3A	EC YKT3C
1 of 20	2 DEC 94	10 OCT 95	24 NOV 97	20 MAY 98

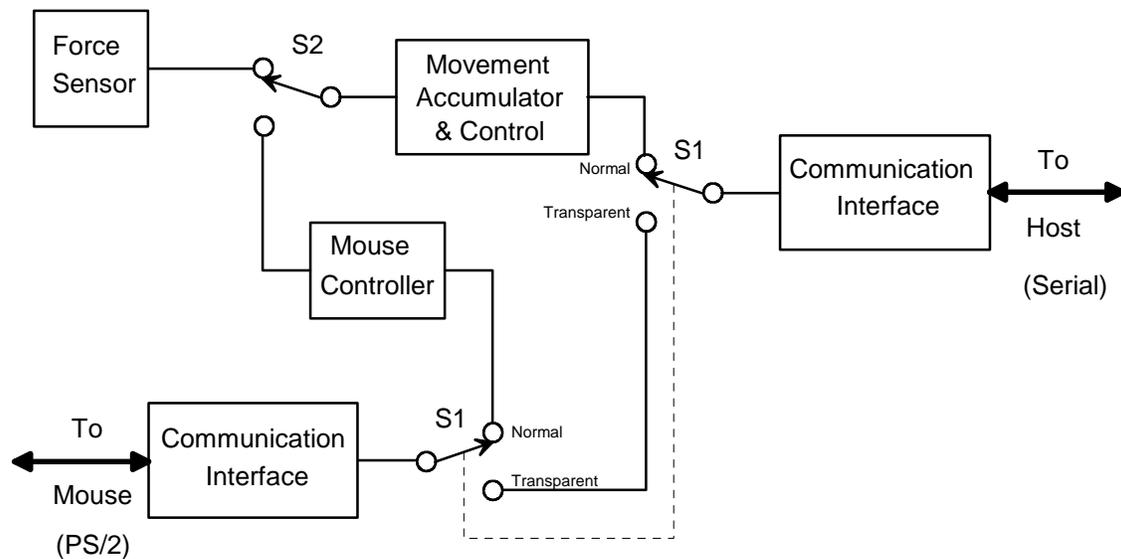
1 GENERAL DESCRIPTION

1.1 INTRODUCTION

This specification supplement defines the TrackPoint integrated pointing device, a force sensing pointing device to be used with a computer system utilizing a serial communication port. This document will specify the logical and electrical functions and interfaces that differ from the PS/2 TrackPoint implementation. The section numbers in this document should be used in conjunction with the PS/2 TrackPoint Engineering Specification Version 4.0.

1.2 DESCRIPTION

The TrackPoint has an internal processing element which interfaces with the host through a serial communication port. The logical function is completely compatible with a Microsoft compatible serial mouse, and mouse drivers and other software may be used unaltered. Specialized drivers are necessary to enable advanced TrackPoint functions, such as Press to Select. The electrical interface utilizes 0 to +5V DC signaling levels and must be translated to proper RS-232C levels (+/- 12V DC) when connected via a standard RS-232C serial port. The TrackPoint processor also serves as host to a PS/2 Mouse or compatible device (via switch S2 below), which may be used in conjunction with the TrackPoint. If the external mouse port is required for a non-compatible device, the TrackPoint processor may be commanded to disable itself and to enter transparent mode in which it relays all signals between the host and the non-compatible device (refer to schematic switch S1 below). In this mode, the TrackPoint will act as a translator between a PS/2 style interface and a RS-232 style interface.



YKT3C.1wp	EC YKTB5	EC YKTB8	EC YKT3A	EC YKT3C
2 of 20	2 DEC 94	10 OCT 95	24 NOV 97	20 MAY 98



2 FUNCTIONAL SPECIFICATION

2.1 POWER ON RESET

The TrackPoint logic shall execute a Power On Reset (POR) when power is applied to the device. Activity on the data and handshaking lines is ignored prior to the completion of the diagnostic sequence. (See RESET mode of operation.)

2.2 MODES OF OPERATION

2.2.1 RESET

Following power on (Hardware Reset) or acceptance of the RESET command (Software Reset), the TrackPoint will execute a set of diagnostics to insure maximum integrity of the device. The functions performed will be a processor test, a checksum verification of the internal program ROM, a RAM integrity test, and check that the analog input from the sensor is within limits. The buttons are not checked, but typically are tested by the keyboard controller during the keyboard POST routines. The external pointing device, if connected, will also be sent a RESET command by the processor and its response will be verified. Commands from the host shall be ignored during POST execution. Execution of the diagnostic sequence shall take a maximum of 500 msec. Note that this is in addition to the time required for power on stabilization ($VCC > 4.5V$). (This is typically under 300 msec.)

2.2.1.1 Software Reset

A software reset is generated by the acceptance of the RESET command (x"FF").

Immediately following satisfactory completion of the diagnostics, a completion code of x"AA" shall be transmitted to the host followed by an I.D. code of x"00". The TrackPoint will then be set to the following defaults: sampling rate of 100 reports per second, stream mode, disabled, resolution 02, and scaling reset. No further action will occur until a command is received from the host. Any commands sent to the TrackPoint prior to the completion of POST will be ignored.

Immediately following a failure to successfully complete the diagnostics, the TrackPoint will send an error code of x"FC" followed by an I.D. code of x"00". The source of the error can be determined by the command x"E2 25" as described on page 11. At this time, the TrackPoint will remain in its default state and await further command input from the host.

2.2.1.2 Hardware Reset

A hardware reset is generated by either the initial application of power, an assertion of the microcontroller's reset pin, or by the acceptance of the Power on Reset software command (x"E2 7F").

All internal RAM values are reset to predetermined default values, the serial UART is set to 1200 bits per second, no parity bit, and 8 data bits, and the TrackPoint will then enter Serial Mode.

YKT3C.lwp	EC YKTB5	EC YKTB8	EC YKT3A	EC YKT3C
3 of 20	2 DEC 94	10 OCT 95	24 NOV 97	20 MAY 98



2.2.1.3 Serial Mouse Reset

A Mouse reset is generated by a falling edge of the host RTS line while the DTR line is low (as seen by the TrackPoint controller's input pins). This mode is used by most standard serial port mouse drivers to determine the presence and type of pointing device connected to the host's serial port. The following sequence should be used by the host software to determine the presence of a mouse compatible device connected via a serial port. Note that "active" indicates 0V on the TrackPoint controller's input pins.

1. Set DTR = inactive, RTS = inactive
2. Wait the RTS and DTR power down time: > 100 ms
3. Set DTR = active
4. Wait the power up time : 100 ms
5. Discard all data in the serial data input buffer and make the serial port ready to receive
6. Set RTS = active
7. Wait for the "M" to come. A timeout of 200 ms is sufficient. The typical response will be within 10-20 ms.
8. Two possible cases will result:
 - a. No "M" was received (no mouse attached).
 - b. A "M" is received:

The second byte ("2"/"3") and eventually the extension recognition bytes (PnP ID), will follow within 100ms. The second byte indicates the number of physical buttons present in this TrackPoint implementation. Refer to Appendix B for further details on the PnP ID.

2.2.8 SERIAL

Serial mode is the TrackPoint's power up default state. While operating in Serial mode, the TrackPoint will transmit data packets to the host nominally at 40 hz at a baud rate of 1200 bits per second utilizing 7 data bits. The TrackPoint's internal UART is configured for 8 data bits, but the 8th data bit is forced to a "1" on transmissions, effectively producing 2 stop bits. Packet format conforms to the Microsoft+ format as described in section 2.3.1. At each transmission time, a packet is sent if there has been activity since the preceding transmission. Activity consists of either motion (a non-zero value in either Motion Accumulator integer part) or a change in button state. In Serial mode any changed button state occurring during the interval will be reported, even if the button has already returned to its previous state (*sticky button*).

The motion and button accumulators receive data from either the TrackPoint or the mouse, according to the following algorithm. At any given time either the TrackPoint or the mouse or neither is in control. If neither is in control, then the first to become active takes control. If one is in control, it remains in control until it has been inactive for at least 0.38 seconds, when it loses control. For this purpose, 'activity' includes holding a button in its 'down' state in addition to making a change in button state.

The TrackPoint will remain in Serial mode until a Set PS/2 Mode Baud Rate (x"E2 53 XX") command is received as described in section 2.4.19.2.

YKT3C.lwp	EC YKTB5	EC YKTB8	EC YKT3A	EC YKT3C
4 of 20	2 DEC 94	10 OCT 95	24 NOV 97	20 MAY 98



2.3.1 SERIAL DATA REPORT

When the TrackPoint is operating in Serial mode, a data report will be transmitted at the end of a sample interval if there is activity to report.

There is activity to report if the integer part of either the X or Y accumulator is non-zero, or if a button report state has changed since the last report. A button's report state is 'depressed' (1) if it has been depressed at any time since the last report, otherwise 'not depressed' (0).

Both motion and button state change may be reported in the same packet. However, applications and drivers differ in the order in which they used motion and button data, so if the sequence of movement and button state is important, it is necessary to report them in separate packets. This occurs in Press-to-Select, where the button action must be applied after the 'backup' movement is completed.

Negative values are expressed in two's complement where 1 output unit in the negative direction would be expressed with all bits on and the sign bit active. The count accumulator must not wrap around, if a count greater than the format allows occurs during a sample period the maximum count will be reported.

The following is the data report format. Data is transmitted serially (LSB first) in the form of seven bit bytes. X and Y data are incremental movements with Y movement considered positive to the south. (For the PS/2 data format, Y movement was considered positive to the north). The data packet format is three or four bytes long. The fourth byte is transmitted in either of the following cases

- a. The middle button is depressed
- b. A report is sent while the middle button is depressed
- c. The middle button is released

Bit 6 can be used to re-synchronize data packets, since bit 6 of byte 1 will always be a 1 and bit 6 of the remaining bytes will always be 0.

YKT3C.lwp	EC YKTB5	EC YKTB8	EC YKT3A	EC YKT3C
5 of 20	2 DEC 94	10 OCT 95	24 NOV 97	20 MAY 98



Engineering Specification

Byte 1	Bit 0	X data bit 6		
	Bit 1	X data bit 7		
	Bit 2	Y data bit 6		
	Bit 3	Y data bit 7		
	Bit 4	Right button status	1 = Depressed	
	Bit 5	Left button status	1 = Depressed	
	Bit 6	Sync Bit	Always 1	
Byte 2	Bit 0	X data bit 0		
	Bit 1	X data bit 1		
	Bit 2	X data bit 2		
	Bit 3	X data bit 3		
	Bit 4	X data bit 4		
	Bit 5	X data bit 5		
	Bit 6	Sync Bit	Always 0	
Byte 3	Bit 0	Y data bit 0		
	Bit 1	Y data bit 1		
	Bit 2	Y data bit 2		
	Bit 3	Y data bit 3		
	Bit 4	Y data bit 4		
	Bit 5	Y data bit 5		
	Bit 6	Sync Bit	Always 0	
Byte 4	Bit 0	reserved	Always 0	
	Bit 1	reserved	Always 0	
	Bit 2	reserved	Always 0	
	Bit 3	reserved	Always 0	
	Bit 4	reserved	Always 0	
	Bit 5	Middle button status	1 = Depressed	See note below
	Bit 6	Sync Bit	Always 0	

Note on middle button: This bit may be masked out for compatibility with applications which require this bit to be 0 (will also prevent 4 byte data packets). Refer to the "Read Extended Button status" command (x"E2 4B") in section 2.4.19.7 for more details.

YKT3C.lwp	EC YKT3A	EC YKT3B	EC YKT3C	EC YKT3D
6 of 20	2 DEC 94	10 OCT 95	24 NOV 97	20 MAY 98

**2.4.17 READ EXTENDED ID (PnP STRING) (x"D0")**

This command shall return the extended ID of the pointing device. The maximum length of the extended ID is 256 bytes. The pointing device will first respond with an acknowledge (x"FA"), followed by the extended ID string. If any transmission error is detected by the host, the x"D0" command should be resent, and the process restarted. If a Resend command (x"FE") is sent, only the last single byte will be resent, and the remainder of the string will be discarded. The following table describes the fields for the extended ID string. Refer to Appendix B beginning on page 17 for more details on the contents of these fields.

Field Name	Size (bytes)	YKT3C string (ASCII)	Description
Other ID	< 17	M3 980501 RSO	Additional Plug and Play ID
Begin Extended ID	1	(Start of Extended ID
PnP Revision Level	2	©\$	Plug and Play Specification Revision Level
Manufacturer ID	3	IBM	Manufacturer ID assigned by EISA
Product Number	3	378	Manufacturer assigned unique Product Number
Product Revision	1	2	Manufacturer assigned product revision number
Extend	1	\	Field present indicator
Extend	1	\	Field present indicator
Class ID	< 33	MOUSE\	Windows 95 Class Identifier
Extend	1	\	Field present indicator
Driver ID	< 41	PNP0F17	Compatible Plug and Play IDs
Extend	1	\	Field present indicator
User Name	< 41	IBM TRACKPOINT VERSION 4.0 YKT3C	Manufacturer supplied product name
Checksum	2	A5	Extended ID checksum
End Extended ID	1)	End of Extended ID

Note that the second character represents the number of physical buttons present. This may be dynamically altered by changing the *button2* TrackPoint control bit as described in section 2.4.19.7. If this character changes, the Product Revision and Checksum fields will also change.

YKT3C.lwp	EC YKTB5	EC YKTB8	EC YKT3A	EC YKT3C
7 of 20	2 DEC 94	10 OCT 95	24 NOV 97	20 MAY 98

**2.4.18 READ SECONDARY ID (x"E1")**

This command will read the secondary device ID of the pointing device (2 bytes). The least significant byte is sent first. The TrackPoint will always return x"01" for the first byte, and a second byte which denotes a specific set of functional specifications. Differing ROM versions are used to denote changes within a given functional set. Current assignments are as follows:

SECONDARY ID (HEX XXXX)	MACHINE TYPE
0101	(8xC752 PLCC implementation), Versions 8E, 98
0201	(8xC752 PDIP implementation), Version A4
0301	(8xC752 PDIP, PLCC implementations), Version AB
0401	(H8-3332 QFP implementation), Version 03
0501	(8xC749 implementation), Versions B2, B4
0601	(8xC752, 8xC749 implementations), Versions B1, B3, B5, B8 (8xC754, SSOP implementation), Version 2B
0701	(H8-3332 QFP implementation), Version ?
0801	(H8-3332 QFP implementation), Version ?
0901	(H8-3332 QFP implementation), Version ?
0A01	(8xC754 SSOP implementation), Version 35
0B01	(8xC754 SSOP implementation), Version 3A
0C01	(8xC754 SSOP implementation), Version 3C

YKT3C.lwp	EC YKTB5	EC YKTB8	EC YKT3A	EC YKT3C
8 of 20	2 DEC 94	10 OCT 95	24 NOV 97	20 MAY 98

**2.4.19 TRACKPOINT COMMAND (x"E2,...")****2.4.19.2 TrackPoint Operational Modes**

COMMAND BYTES (Hex XX)	DESCRIPTION
E2 53 XX	<p>Set PS/2 Mode Baud Rate [<i>baudrate, hibaud</i>]</p> <p>This command is used to place the TrackPoint into PS/2 mode, and also to increase the serial baud rate. The baud rate changes after the last ACK is returned. Once in PS/2 mode, data is transmitted with 8 bit data packets with odd parity, following all conventions of the normal PS/2 Auxiliary Device Protocol.</p> <p>The parameter for this command can be determined by the following formula: $256 - \frac{14745600}{384 \times Baud}$, where <i>Baud</i> is the desired baud rate in bits per second. For example a value of x"FE" (254 decimal) results in a baud rate of 19,200 bits per second.</p> <p>For proper operation of advanced TrackPoint functions, the baud rate should be set at a minimum of 9600 baud. A baud rate of 19,200 baud will improve external mouse performance and two handed mode operation, and the maximum rate of 38,400 baud will result in the lowest average power consumption.</p> <p>To return to Serial mode, use the x"E2 7F" command as described in section 2.4.19.10.</p>
E2 80 62 E2 81 62 XX	<p>Transmit Pacing delay time [<i>xmitgap</i>]</p> <p>This parameter contains the delay value for the time between the stop bit of one transmitted character and the start bit of the next transmitted character. This is sometimes referred to as transmit pacing. Each increment is 11 μs. The default value for <i>xmitgap</i> is x"09" (9 decimal) corresponding to 99 μs. This delay value is independent of baud rate. This value is not affected by a reset (x"FF") or set default (x"F6") command.</p>

YKT3C.lwp	EC YKTB5	EC YKTB8	EC YKT3A	EC YKT3C
9 of 20	2 DEC 94	10 OCT 95	24 NOV 97	20 MAY 98



2.4.19.7 Middle Button Controls

COMMAND BYTES (Hex XX)	DESCRIPTION
------------------------------	-------------

E2 47 2C 02	Toggle 2 Physical Buttons installed [<i>button2</i>]
--------------------	--

This bit is set to indicate that only 2 physical buttons exist in this hardware implementation. The default value for this bit is 0, indicating the presence of three physical buttons. The state of this bit is used in the response to the Read Extended Button Status command (x"E2 4B"), the PnP response string generated by a Serial Mouse Reset, and in the response to the Read Extended ID command (x"D0"). A reset (x"FF") or set default (x"F6") command will not affect this bit.

2.4.19.8 Calibration Parameters

COMMAND BYTES (Hex XX)	DESCRIPTION
------------------------------	-------------

E2 47 2D 20	Toggle automatic Z axis DAC step [<i>skipzstep</i>]
--------------------	---

Setting this bit will disable automatic DAC calibrations for the Z axis. The default value for this bit is 0. A reset (x"FF") or set default (x"F6") command will not affect this bit.

2.4.19.10 Debug Parameters

COMMAND BYTES (Hex XX)	DESCRIPTION
------------------------------	-------------

E2 46	Read ROM version. (READ ONLY)
--------------	-------------------------------

The TrackPoint controller will return a single byte indicating its ROM version. For YKT3C this is x"3C".

YKT3C.lwp	EC YKT3A	EC YKT3B	EC YKT3C	EC YKT3D
10 of 20	24 NOV 97	10 OCT 95	20 MAY 98	



2.4.19.12 Miscellaneous Functions

COMMAND BYTES (Hex XX)	DESCRIPTION
------------------------------	-------------

E2 52	Generate Tactile output pulse
--------------	-------------------------------

This command will cause the tactile output pin to pulse low for approximately 50 μ s.

YKT3C.lwp	EC YKT3B5	EC YKT3B8	EC YKT3A	EC YKT3C
11 of 20	2 DEC 94	10 OCT 95	24 NOV 97	20 MAY 98



3 ELECTRICAL SPECIFICATION

3.3 DATA TRANSMISSION

The TrackPoint communicates bidirectionally using the TXD and RXD signal lines, and uses the RTS and DTR lines as handshaking inputs. The TXD line is a CMOS compatible totem pole output and the RXD, RTS, and DTR lines are CMOS compatible inputs, with an internal 40Kohm pullup to +5V. During non-transmission intervals, TXD remains in the +5V (logic "1") state.

3.3.1 TXD LINE

The TrackPoint transmits data to the system on the TXD (Transmit Data) line. Prior to initiation of a transmission, the TrackPoint checks for the DTR line to be in the active state. Then after the transmit pacing delay (as described in section 2.4.19.2) the TrackPoint proceeds to transmit the start bit, data bits, parity, and stop bit. Once the transmission starts, it will run to completion unless a hardware to mouse reset is encountered.

Each transmission will consist of 9 or 11 bits transmitted serially over the TXD line. A logical "1" is transmitted as a "+" level. When operating in its power on default state, a 9 bit frame is used. After acceptance of the Set PS/2 Baud Rate command (x"E2 53 XX"), an 11 bit data frame is used. The parity bit will be "1" or "0" such that the eight data bits plus parity bit will always contain an odd number of 1's.

The TXD line is a CMOS compatible totem pole output. During non-transmission intervals, TXD remains in the +5V (logic "1") state.

1st Bit	Start Bit	Always "0"	
2nd Bit	Data Bit 0	Least Significant Bit	
3rd Bit	Data Bit 1		
4th Bit	Data Bit 2		
5th Bit	Data Bit 3		
6th Bit	Data Bit 4		
7th Bit	Data Bit 5		
8th Bit	Data Bit 6		
9th Bit	Data Bit 7	Most Significant Bit	PS/2 Mode only
10th Bit	Parity Bit	Odd Parity	PS/2 Mode only
11th Bit	Stop Bit	Always "1"	

3.3.2 RXD LINE

The TrackPoint receives data from the system on the RXD (Receive Data) line. There are no handshaking lines to control reception of bytes by the TrackPoint. The ACK protocol of the PS/2 format is used to ensure that subsequent commands are not sent before the previous command has been processed. The host software must wait for the software ACK before sending another byte to the TrackPoint.

The transmission format is identical to that which is used on the TXD line, which is specified above.

The RXD line is a CMOS compatible input, with an internal 40Kohm pullup to +5V.

YKT3C.lwp	EC YKTB5	EC YKTB8	EC YKT3A	EC YKT3C
12 of 20	2 DEC 94	10 OCT 95	24 NOV 97	20 MAY 98



3.3.3 DTR LINE

The DTR (Data Terminal Ready) input line is used by the TrackPoint to determine when the host is ready to receive a transmission on the TXD line. When the DTR line is in the active state (0V at the TrackPoint controller pin), the TrackPoint is allowed to begin transmission of data to the host. If the DTR line goes inactive during a transmission, it will cause transmission to cease after the current byte is completed.

The DTR line is a CMOS compatible input, with an internal 40Kohm pullup to +5V.

3.3.4 RTS LINE

The RTS (Request To Send) input line is used as a reset line to the TrackPoint to indicate that the host desires to reset the TrackPoint and to determine the existence and type of device connected to the host serial port. Upon the transition of the RTS line to the active state (a +5V to 0V transition at the TrackPoint controller pin), the TrackPoint will set certain defaults and output an identification string on the TXD line. This input is edge sensitive, and will only respond to a falling edge on the RTS input pin.

The RTS line is a CMOS compatible input, with an internal 40Kohm pullup to +5V.

3.3.5 TRACKPOINT DATA OUTPUT

When the TrackPoint is READY to transmit data, it must first check for DTR active. If DTR is inactive (inhibit status), data shall be continuously updated in the TrackPoint and no transmissions shall be started. If a byte is waiting in the TrackPoint's Receive Input Buffer (RIB), the TrackPoint shall input system data, and no transmissions shall be started by the TrackPoint until DTR is active and the RIB is empty.

If a transmission is required, and the DTR line is active, the TrackPoint shall proceed to output the start bit, seven or eight data bits, (optional) parity bit and stop bit.

Following a transmission, the system can inhibit the TrackPoint by deasserting the DTR line until it can service the input.

3.3.6 TRACKPOINT DATA INPUT

When the host is ready to send data to the TrackPoint, it will serially transmit a 9 or 11 bit data frame on the RXD line. Other than during a Serial Mouse Reset, the TrackPoint is always ready to receive data. There are no handshaking lines to control data from the host to the TrackPoint.

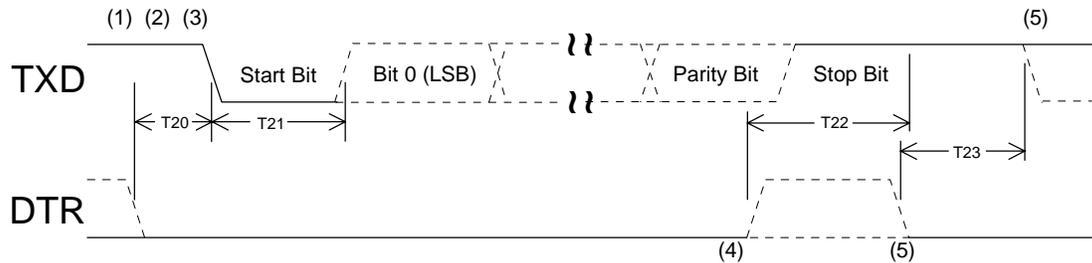
For each system command or data transmission to the TrackPoint that requires a TrackPoint response, the system must wait for the TrackPoint to respond (ACK) before sending its next output.

YKT3C.lwp	EC YKT3B5	EC YKT3B8	EC YKT3A	EC YKT3C
13 of 20	2 DEC 94	10 OCT 95	24 NOV 97	20 MAY 98

3.4 TRACKPOINT SENDING DATA TIMING

The following describes the typical sequence of events when the system is receiving data from the TrackPoint.

1. The TrackPoint checks the DTR line. If the line is inactive (high), output from the TrackPoint is not allowed.
2. The TrackPoint checks the Receive Input Buffer. If the RIB is full, the TrackPoint controller discards the byte to be transmitted, and processes the new command in the RIB.
3. The serial transmission is started.
4. The system can set the DTR line inactive (high) to inhibit the next transmission.
5. The system releases the DTR line to allow the next transmission.



Bit 7 and the parity bit are only transmitted while in PS/2 mode.

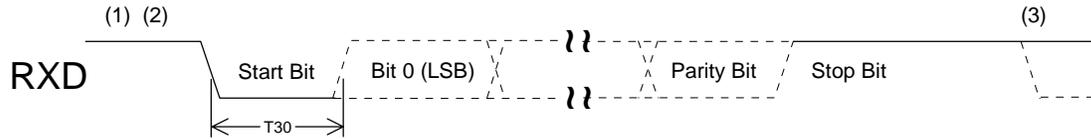
Timing Parameter	Spec Min/Max	Version 3C Min/Max, Typical
T20	Time from DTR release to falling edge of TXD	15/30μs
T21	Bit time (at 19.2Kbaud)	50/54μs
T22	DTR setup time	>15μs
T23	Transmit Pacing Delay	11-2816μs

YKT3C.lwp	EC YKT3A	EC YKT3B	EC YKT3C	EC YKT3D
14 of 20	2 DEC 94	10 OCT 95	24 NOV 97	20 MAY 98

3.5 TRACKPOINT RECEIVING DATA TIMING

The following describes the typical sequence of events when the system is sending data to the TrackPoint.

1. It is assumed that the host has previously initialized the TrackPoint with a Serial Mouse Reset.
2. The system may start the transmission at any time.
3. The system must wait for an ACK before sending another byte.



Bit 7 and the parity bit are only transmitted when in PS/2 mode.

Timing Parameter		Spec Min/Max	Version 3C Typical
T30	Bit time (at 19.2Kbaud)	50/54μs	52μs

YKT3C.lwp	EC YKT3A	EC YKT3B	EC YKT3C	EC YKT3D
15 of 20	24 NOV 97	10 OCT 95	20 MAY 98	



Appendix A - TrackPoint Address Map

Listed below are the differences from the PS/2 TrackPoint Address Map

RAM address	Bit Position	Name	Description
16		t3cnt	pseudo timer 3 counter
29	5	reg29.5	unassigned
2C		config	Configuration byte
	1	config.1	unassigned
	2	button2	2 physical buttons exist
2D		reg2d	bit fields
	5	skipzstep	skip automatic Z axis DAC step
	6	msfix	use current sampling rate for external mouse (instead of 200hz)
2E		reg2E	bit fields
	0	hibaud	baud rate changed from 1200,n,7,1
	1	saveTR3	temporary TR3 flag save
	3	tr3	pseudo timer 3 running
	4	iding	mouse PnP ID transmission in progress
62		xmitgap	serial transmit pacing delay
63			unassigned
7D		torigin	temporary origin
7E		baudrate	serial baud rate reload value

YKT3C.lwp	EC YKTB5	EC YKTB8	EC YKT3A	EC YKT3C
16 of 20	2 DEC 94	10 OCT 95	24 NOV 97	20 MAY 98

**Appendix B - PLUG AND PLAY REFERENCE**

This section details the specifications for the response to the "Read Extended ID" command (x"D0") as described in section 2.4.17 as well as the response to the Serial Mouse Reset as described in section 2.2.1.3. The maximum length of the extended ID is 256 bytes. The following table describes the fields for the extended ID string.

Field Name	Size (bytes)	Required	Description	Notes
Other ID	< 17	No	Additional Plug and Play ID	
Begin Extended ID	1	Yes	Start of Extended ID	
PnP Revision Level	2	Yes	Plug and Play Specification Revision Level	
Manufacturer ID	3	Yes	Manufacturer ID assigned by EISA	
Product Number	3	Yes	Manufacturer assigned unique Product Number	
Product Revision	1	Yes	Manufacturer assigned product revision number	
Extend	1	No	Field present indicator	3
Serial Number	8	No	Optional device serial number	
Extend	1	No	Field present indicator	3
Class ID	< 33	No	Windows 95 Class Identifier	
Extend	1	No	Field present indicator	3
Driver ID	< 41	No	Compatible Plug and Play IDs	
Extend	1	No	Field present indicator	3
User Name	< 41	No	Manufacturer supplied product name	
Checksum	2	(see note)	Extended ID checksum	4
End Extended ID	1	Yes	End of Extended ID	

Notes:

1. The entire length of the extended ID string, including all fields and delimiters, shall not exceed 256 characters. This is to minimize delays in system boot-up time.
2. Values for the extended ID fields, enclosed within the Begin PnP and EndPnP characters, must be static for a given device.
3. For all optional fields, the character string begins with the extend character "\". If an optional field is not present but subsequent fields are, that field shall be represented by a single "\" character. "\" is coded as x"5C" (or the offset ASCII value of x"3C" when operating in 6 bit mode). If the subsequent fields are not provided, then the "\" characters may be omitted.
4. The checksum is not required unless an Extend Field is present. The arithmetic checksum includes all characters in all fields from the Begin Extended ID to End Extended ID, exclusive of the checksum field.

YKT3C.lwp	EC YKTB5	EC YKTB8	EC YKT3A	EC YKT3C
17 of 20	2 DEC 94	10 OCT 95	24 NOV 97	20 MAY 98



Extended ID/PnP Field Definitions

Other ID This fields contains an optional ID. The length of this field shall not exceed 16 bytes. This field is optional and may be omitted. Use of BeginPnP or EndPnP characters (x"28",x"29",x"08",x"09") is not allowed. Use of <CR> and <LF> characters are allowed for readability.

YKT3C string (ASCII): M3 19980501 RSO
YKT3C string (Hex): 4D332031393938303530312052534F
YKT3C string (ASCII -20h): 4D330011191918101510110032332F

Begin PnP This character is used to mark the beginning of the extended ID record. The character used is the open parenthesis character, x"28" (7 bit IDs), or the offset value x"09" (6 bit IDs). For the extended ID record to be valid, there must be a matching EndPnP character. See End PnP field definition.

YKT3C string (ASCII): (
YKT3C string (Hex): 28
YKT3C string (ASCII -20h): 08

PnP Rev The two byte revision code that represents the device's compliance level with the Plug and Play ISA specification. The 12 bit level in this fields is created by extracting bits 5 to 0 of byte 1 and concatenating it with bits 5 to 0 of byte 2. These 12 bits are treated as a hexadecimal number. Dividing this number by 100 decimal yields the version number, ranging from 0.00 (x"00 00") through 1.0 (x"01 24") to 40.95 (x"3F 3F"). PnP Revision codes that contain x"29" must not be used. This will preclude certain levels. For example, level 1.05 is coded as x"01 29" and its use is not permitted.

YKT3C string (ASCII):Ⓢ
YKT3C string (Hex): 0124
YKT3C string (ASCII -20h): 0124

EISA Mfr ID Each device manufacturer must have a unique three character EISA Identifier. The device manufacturer shall report its EISA ID in this field. This field, together with the Product Number and Product Revision fields are used to generate the Product ID used by the PnP BIOS routines to identify this device.

YKT3C string (ASCII): IBM
YKT3C string (Hex): 49424D
YKT3C string (ASCII -20h): 29222D

Product Number Devices complying with this specification shall contain a unique product number in this field. The characters in this field are restricted to valid hexadecimal characters, (i.e. 0-9, and A-F). This field, together with the EISA Mfr ID and Product Revision fields are used to generate the Product ID used by the PnP BIOS routines to identify this device.

YKT3C string (ASCII): 378
YKT3C string (Hex): 333738
YKT3C string (ASCII -20h): 131718

YKT3C.lwp	EC YKTB5	EC YKTB8	EC YKT3A	EC YKT3C
18 of 20	2 DEC 94	10 OCT 95	24 NOV 97	20 MAY 98



Product Revision This field contains the revision of the product, and is restricted to a single hexadecimal character, (i.e. 0-9, and A-F). This field, together with the EISA Mfr ID and Product Number fields are used to generate the Product ID used by the PnP BIOS routines to identify this device.

YKT3C string (ASCII): 3\
YKT3C string (Hex): 335C
YKT3C string (ASCII -20h): 133C

Product Serial Number This field is optional. It contains a unique manufacturer serial number. The serial number is a 32 bit value in 8 ASCII encoded hexadecimal digits. The first character represents the most significant 4 bits.

YKT3C string (ASCII): \
YKT3C string (Hex): 5C
YKT3C string (ASCII -20h): 3C

Class Name This field is optional. The character string must not exceed 32 characters. If the device doesn't belong to a predefined class, this field must not be included in the record. The valid value for pointing devices is MOUSE. A list of valid classes can be found in the Microsoft™ Windows 95™ DDK.

YKT3C string (ASCII): MOUSE\
YKT3C string (Hex): 4D4F5553455C
YKT3C string (ASCII -20h): 2D2F3533253C

Compatible Device IDs This field is optional. The character strings must not exceed 40 characters total. The field consists of one or more strings separated by the comma character “,” (x"2C"). The Ids indicate a compatible device whose device driver will work with the current device. Each Device ID in the field has the same format as defined for EISA Mfr ID (three characters) and a Product ID (four characters). For example, a pointing device compatible with the IBM PS/2 Mouse would indicate the seven character ID of a known compatible IBM PS/2 Mouse.

YKT3C string (ASCII): PNP0F17\
YKT3C string (Hex): 504E50304631375C
YKT3C string (ASCII -20h): 302E30102611173C

User Name This field is optional. It must not exceed 40 characters. This field contains a user-readable and user-recognized product device description. It is expected that this field will be available for display by a Plug and Play User Interface. This field may not contain the EndPnP character, x"29". For readability, the User Name field may include <CR> <LF> characters.

YKT3C string (ASCII): IBM TRACKPOINT VERSION 4.0 YKT3C
YKT3C string (Hex):
49424D20545241434B504F494E542056455253494F4E20342E30
YKT3C string (ASCII -20h):
29222D00343221232B302F292E340036253233292F2E00140E10

YKT3C.lwp	EC YKTB5	EC YKTB8	EC YKT3A	EC YKT3C
19 of 20	2 DEC 94	10 OCT 95	24 NOV 97	20 MAY 98



Checksum This field is required but follows an Extend field because of its position in the record. This field is a modulo-256, 8 bit arithmetic checksum of all characters from Begin PnP field to End PnP field (inclusive), exclusive of the checksum characters themselves. The value is represented as a two character hexadecimal number. The first character is the most significant digit, and the second character is the least significant digit.

YKT3C string (ASCII): A5
YKT3C string (Hex): 4135
YKT3C string (ASCII -20h): 1415

End PnP This field is required. It is used to indicate the end of the Extended ID record. If Begin PnP is x"28" then this is "29"; if Begin PnP is x"08" then this is "09"; The End PnP character must not appear in the option information fields so that a device driver can properly detect the end of the Extended ID record.

YKT3C string (ASCII):)
YKT3C string (Hex): 29
YKT3C string (ASCII -20h): 09

These PnP specifications conform to "Plug and Play ISA Specification" Version 1.0a dated May 5, 1994 and are identical to the "Plug and Play External COM Device Specification" Version 1.00 dated February 28, 1995.

YKT3C.lwp	EC YKTB5	EC YKTB8	EC YKT3A	EC YKT3C
20 of 20	2 DEC 94	10 OCT 95	24 NOV 97	20 MAY 98