Chipsmall Limited consists of a professional team with an average of over 10 year of expertise in the distribution of electronic components. Based in Hongkong, we have already established firm and mutual-benefit business relationships with customers from,Europe,America and south Asia,supplying obsolete and hard-to-find components to meet their specific needs.

With the principle of “Quality Parts,Customers Priority,Honest Operation,and Considerate Service”,our business mainly focus on the distribution of electronic components. Line cards we deal with include Microchip,ALPS,ROHM,Xilinx,Pulse,ON,Everlight and Freescale. Main products comprise IC,Modules,Potentiometer,IC Socket,Relay,Connector.Our parts cover such applications as commercial,industrial, and automotives areas.

We are looking forward to setting up business relationship with you and hope to provide you with the best service and solution. Let us make a better world for our industry!

Contact us

Tel: +86-755-8981 8866 Fax: +86-755-8427 6832
Email & Skype: info@chipsmall.com Web: www.chipsmall.com
Address: A1208, Overseas Decoration Building, #122 Zhenhua RD., Futian, Shenzhen, China

ADNS-2700

Single Chip USB Optical Mouse Sensor

Data Sheet

Lead (Pb) Free
RoHS 6 fully compliant

Description

The ADNS-2700 is a compact, one chip USB optical mouse sensor designed for implementation of a non-mechanical tracking engine in computer mice.

It is based on optical navigation technology that measures changes in position by optically acquiring sequential surface images (frames) and mathematically determining the direction and magnitude of movement.

The sensor is in an 8-pin optical package that is designed to be used with the ADNS-5100-001 trim lens, LED clip and the HLMP-EG3E-xxxxx LED. Together, these parts provide a complete and compact mouse sensor. There are no moving parts and precision optical alignment is not required, thus facilitating high volume assembly.

The output format is USB. This device meets HID Revision 1.11 specification and is compatible with USB Revision 2.0 specification.

Frame rate is varied internally to the sensor to achieve tracking and speed performance, eliminating the need for the use of many registers.

Default resolution is specified as 1000 counts per inch, with rates of motion up to 30 inches per second.

A complete mouse can be built with the addition of a PC board, switches and Z-wheel, plastic case and cable.

NOTE: ADNS-2700 will be referred to as "sensor", ADNS-5100-001 as " trim lens" and HLMP-EG3E as "LED" hereafter.

Features

- One chip USB mouse sensor
- USB 2.0 Low Speed Compliance
- 12 bits USB motion data reporting
- Meets HID Revision 1.11
- Single 5.0 volts power supply
- High speed motion detection at 30 inches per second (ips) and acceleration up to 8 g
- Input buttons: 3 buttons
- Z-Wheel interface for vertical scroll
- Integrated oscillator
- Integrated USB D- pull-up resistor
- Product string is set to "USB Optical Mouse"
- On-chip OTP memory for device configuration flexibility without any external software driver:
  - Programmable resolution from 500 to 1250 counts per inch (cpi) with 250 cpi step.
  - Programmable sensor orientation
  - Programmable VID and PID.

Applications

- Corded optical mice
- Trackballs
- Integrated input devices

Theory of Operation

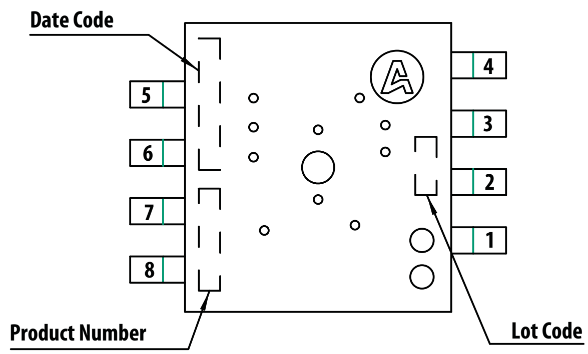
The sensor is based on Optical Navigation Technology. It contains an Image Acquisition System (IAS), a Digital Signal Processor (DSP) and USB stream output.

The IAS acquires microscopic surface images via the lens and illumination system provided by the trim lens. The clip and LED. These images are processed by the DSP to determine the direction and distance of motion. The DSP generates the Δx and Δy relative displacement values which are converted to USB motion data.

Table 1. Pin Name Description

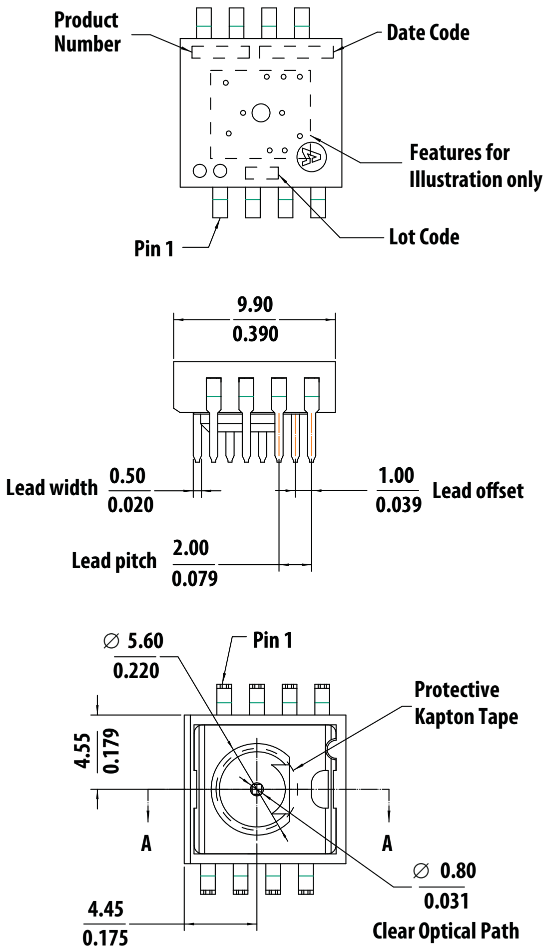
Pin Name	Input/Output	Description
XY_LED	I	XY_LED
VDDA5	–	5-Volt Power
ZB	I	Z-Wheel quadrature input
ZA	I	Z-Wheel quadrature input
SW	–	3-in-1 button pin. Do not force any voltage into this pin
GND	–	Ground
D-	I/O	USB D- line
D+	I/O	USB D+ line

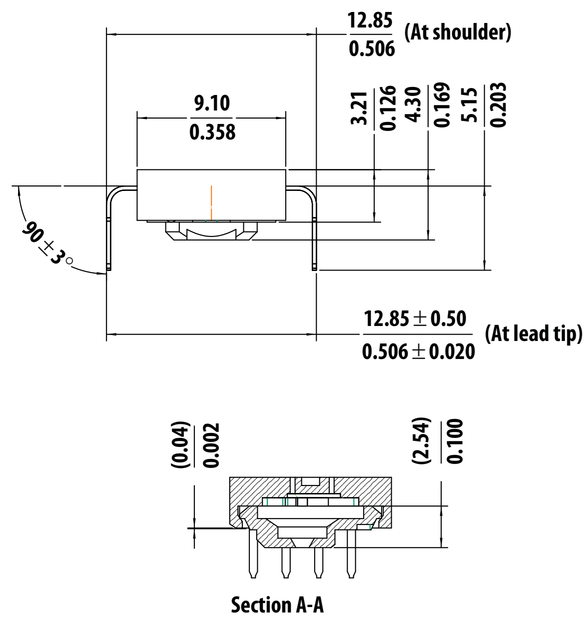
Package Pinout



Item	Marking	Remarks
Product Number	A2700	
Date Code	XYYWWZ	X = Subcon Code YYWW = Date Code Z = Sensor Die Source E = Wafer Revision
Lot Code	VVV	Numeric

Figure 1. Package outline drawing (top view)





Notes:

1. Dimensions in millimeter / inches.
2. Dimensional tolerance: ± 0.1mm.
3. Coplanarity of leads: 0.1mm.
4. Lead pitch tolerance: ± 0.15mm.
5. Non-cumulative pitch tolerance: ± 0.15mm.
6. Angular tolerance: ± 3°
7. Maximum flash: 0.2mm.
8. Brackets () indicate reference dimension.
9. Document Number: LED_DIFF_8B_PKG_002

Figure 2. Package Outline Drawing

CAUTION: It is advised that normal static precautions be taken in handling and assembling of this component to prevent damage and/or degradation which may be induced by ESD.

Overview of Optical Mouse Sensor Assembly

Avago Technologies provides an IGES file drawing describing the base plate molding features for lens and PCB alignment. The sensor is designed for mounting on a through-hole PCB. There is an aperture stop and features on the package that align to the lens. The lens provides optics for the imaging of the surface as well as the illumination of the surface at the optimum angle. Features on the lens align it to the sensor, base plate, and clip with the LED. The clip holds the LED in relation to the lens. The LED must be inserted into the clip and the LED's leads formed prior to loading on the PCB.

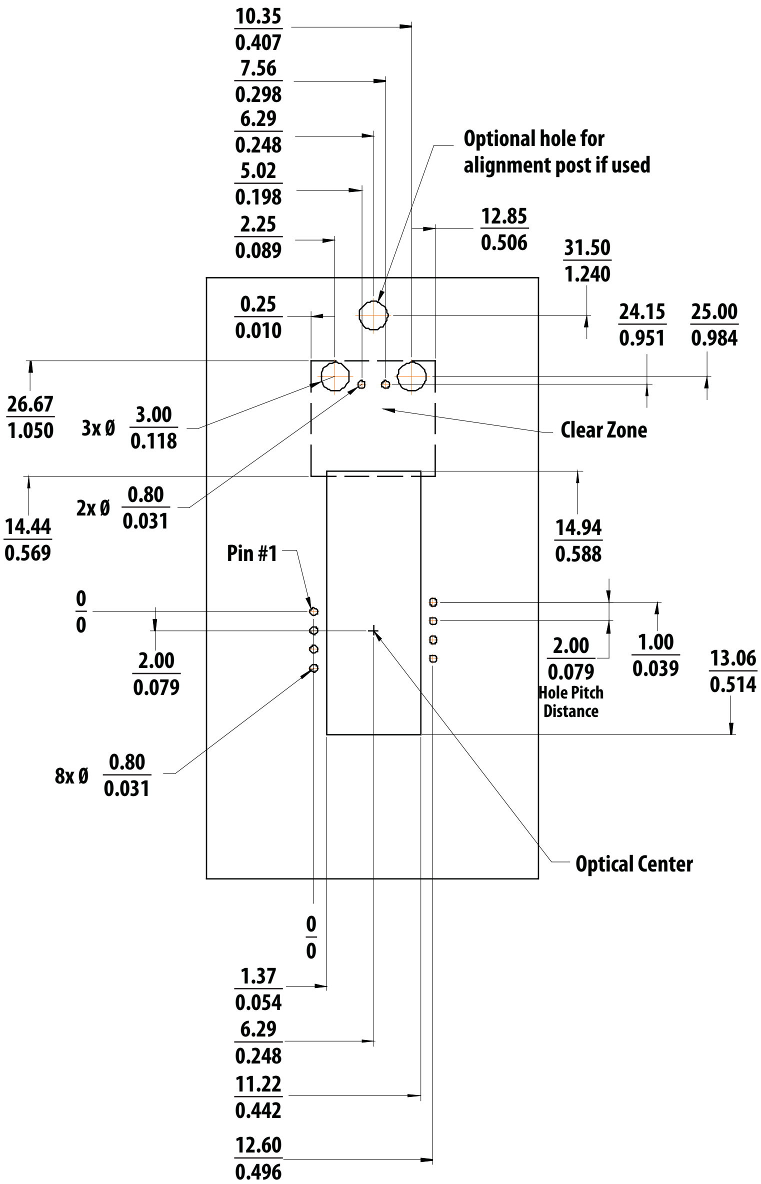
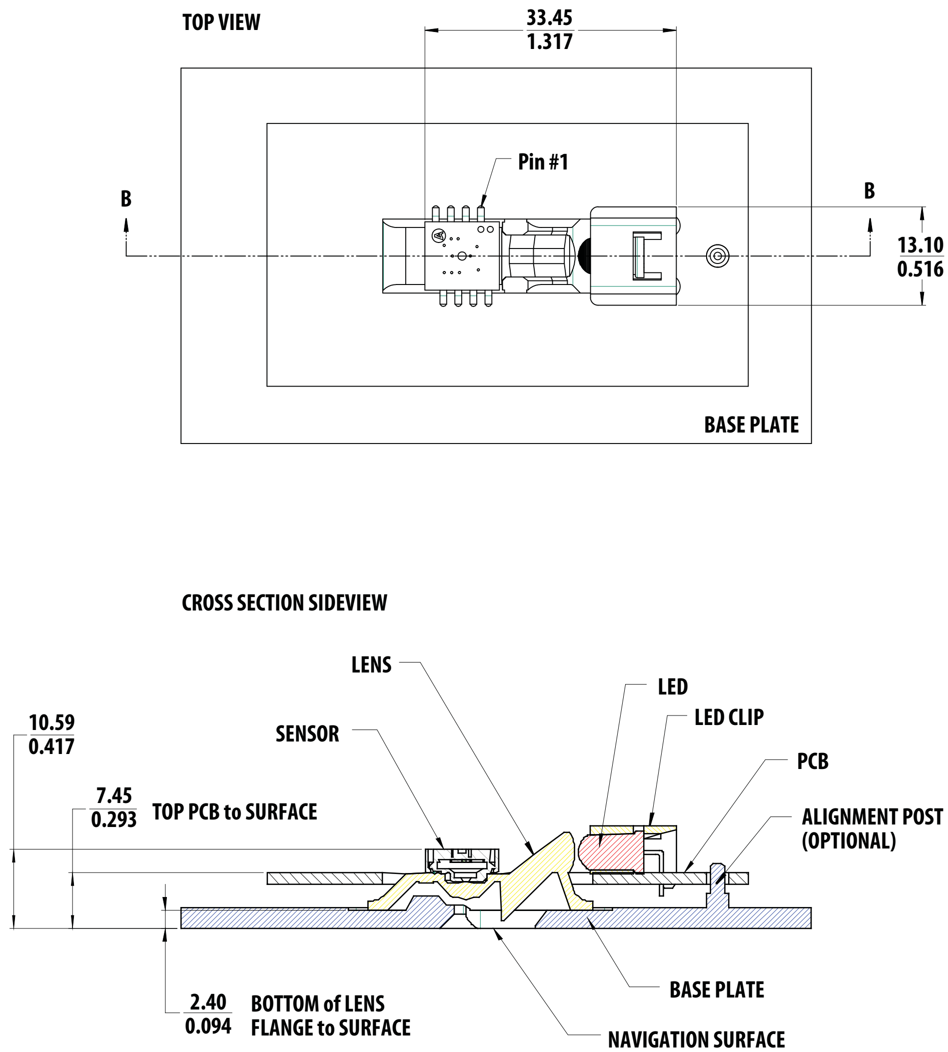


Figure 3. Recommended PCB Mechanical Cutouts and Spacing



NOTE: Dimensions in mm/Inches
Important Note: Pin 1 of sensor should be located nearest to the LED

Figure 4. 2D Assembly drawing of sensor (Top and Side View)

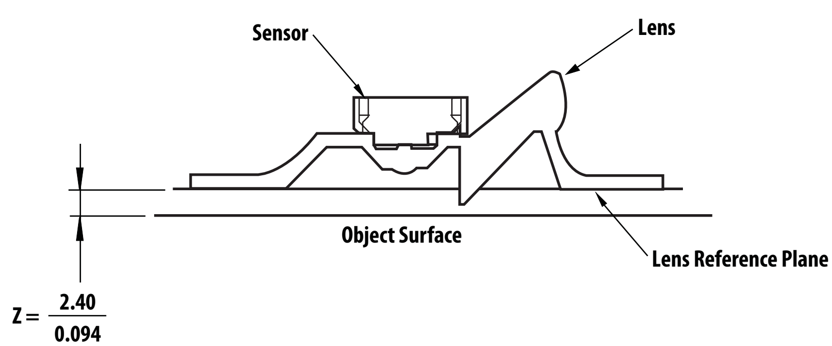
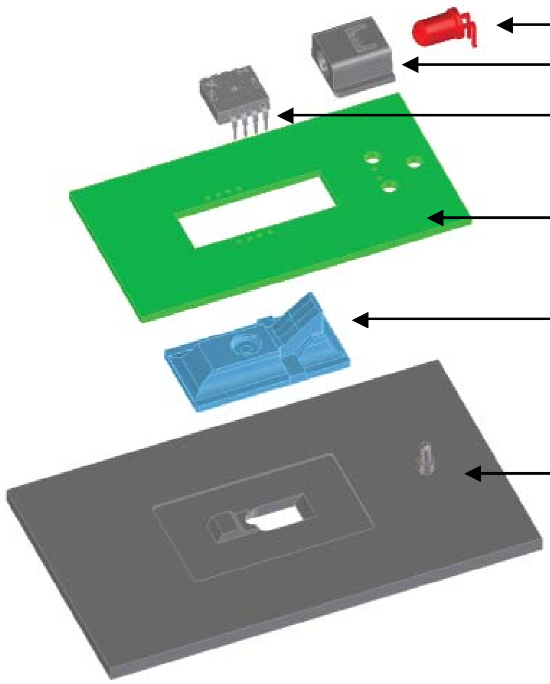


Figure 5. Distance from lens reference plane to tracking surface (Z)



IMPORTANT NOTE: P-bin LED or better is recommended.

Figure 6. Exploded View of Assembly

PCB Assembly Considerations

1. Insert the sensor and all other electrical components into PCB.
2. Insert the LED into the assembly clip and bend the leads 90 degrees.
3. Insert the LED clip assembly into PCB.
4. This sensor package is only qualified for wave-solder process.
5. Wave solder the entire assembly in a no-wash solder process utilizing solder fixture. The solder fixture is needed to protect the sensor during the solder process. It also sets the correct sensor-to-PCB distance as the lead shoulders do not normally rest on the PCB surface. The fixture should be designed to expose the sensor leads to solder while shielding the optical aperture from direct solder contact.
6. Place the lens onto the base plate.
7. Remove the protective Kapton tape from optical aperture of the sensor. Care must be taken to keep contaminants from entering the aperture. Recommend not to place the PCB facing up during the entire mouse assembly process. Recommend to hold the PCB first vertically for the Kapton removal process.
8. Insert PCB assembly over the lens onto the base plate aligning post to retain PCB assembly. The sensor aperture ring should self-align to the lens.
9. The optical position reference for the PCB is set by the base plate and lens. Note that the PCB motion due to button presses must be minimized to maintain optical alignment.
10. Install mouse top case. There MUST be a feature in the top case to press down onto the PCB assembly to ensure all components are interlocked to the correct vertical height.

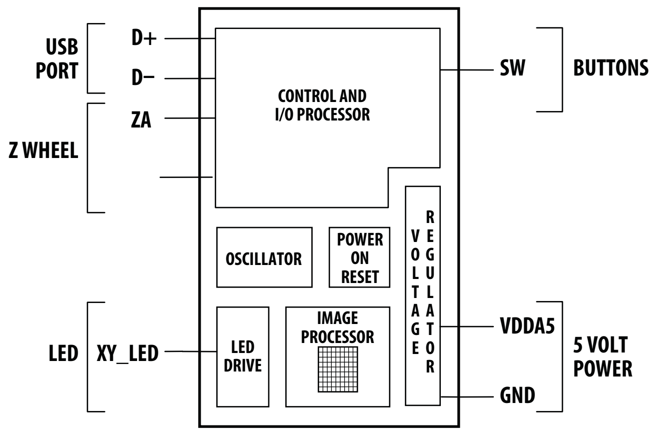


Figure 7. Block diagram of optical mouse

Design considerations for improving ESD Performance

The table below shows typical values assuming base plate construction per the Avago Technologies supplied IGES file trim lens. Stand-off of the base plate shall not be larger than 5 mm.

Typical Value	Distance (mm)
Creepage	17.9 mm
Clearance	9.2 mm

Note that the lens material is polycarbonate or polystyrene HH30, therefore, cyanoacrylate based adhesives should not be used as they will cause lens material deformation

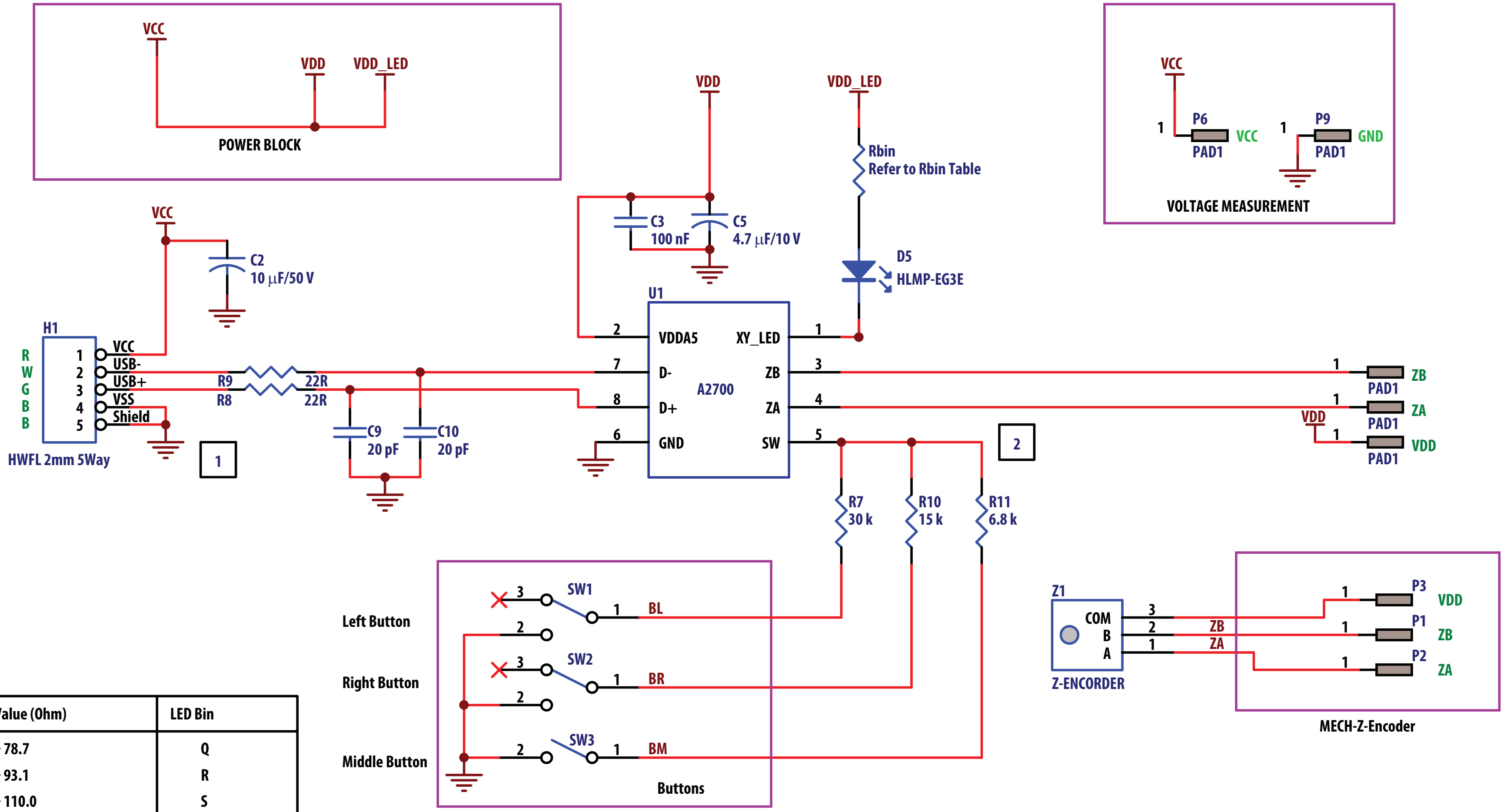


Table 1

Rbin Value (Ohm)	LED Bin
59.0 – 78.7	Q
59.0 – 93.1	R
59.0 – 110.0	S
59.0 – 143.0	T

⚠1 Use ONLY 1% tolerance resistor for buttons

1 Try to shield the HWEL with a SHIELD Plane (USB pin 5) similar size of HWFL before connect to GND. Isolate the Shield island with GND plane

2 Don't use jumper wire. Use resistor to jump instead. Keep the trace wide.
Keep (pin 5) away from noisy traces, GND plane and Power plane.
The 3 button trace for SW pin need to be same width, length.

Figure 8. Application Circuit with sensor

Regulatory Requirements

- Passes FCC B and worldwide analogous emission limits when assembled into a mouse with shielded cable and following Avago Technologies recommendations.
- Passes EN61000-4-4/IEC801-4 EFT tests when assembled into a mouse with shielded cable and following Avago Technologies recommendations.
- UL flammability level UL94 V-0.
- Provides sufficient ESD creepage/clearance distance to withstand discharge up to 8 kV when assembled into a mouse with trim lens according to usage instructions above.

Absolute Maximum Ratings

Parameter	Symbol	Minimum	Maximum	Units	Notes
Storage Temperature	T_S	-40	85	°C	
Operating Temperature	T_A	-15	55	°C	
Lead Solder Temperature			260	°C	For 10 seconds, 1.6 mm below seating plane.
Supply Voltage	V_{DDA5}	-0.5	5.5	V	
ESD			2	kV	All pins, human body model JESD22-A114
Input Voltage	V_{IN}	-0.5	V_{DDA5} +0.5	V	All I/O pins except D+, D-
		-1.0	4.6	V	D+, D-, AC waveform, see USB specification (7.1.1)
Input Short Circuit Voltage	V_{SC}	0	V_{DDA5}	V	D+, D-, see USB specification (7.1.1)

Recommended Operating Condition

Parameter	Symbol	Minimum	Typical	Maximum	Units	Notes
Operating Temperature	T_A	0		40	°C	
Power Supply Voltage	V_{DDA5}	4.25	5.0	5.25	V	For accurate navigation and proper USB operation
	V_{ddm}	4.0	5.0	5.25	V	Maintains communication to USB host and internal register contents.
Power Supply Rise Time	V_{RT}	0.1		6	ms	
Supply Noise	V_N			100	mV	Peak to peak within 0-100 MHz bandwidth
Velocity	Vel		30		ips	
Acceleration	Acc		8		g	0.5 g from Rest
Serial Port Clock Frequency	f_{SCLK}			1	MHz	50% duty cycle
Distance from Lens Reference Plane to Surface	Z	2.3	2.4	2.5	mm	See Figure 9
Light Level onto IC	IRR_{INC}	80		25000	MW/m^2	$\lambda = 639$ nm
Frame Rate			2400		fps	Internally adjusted by sensor

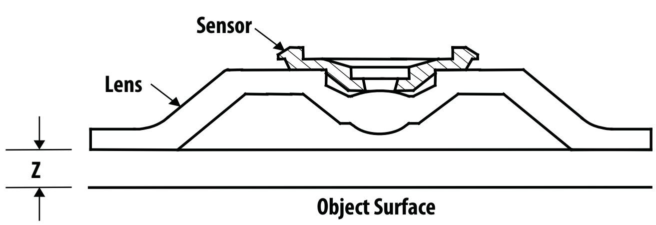


Figure 9. Distance from lens reference plane to object surface

AC Electrical Specifications

Electrical Characteristics over recommended operating conditions. Typical values at 25° C, V_{DDA5} = 5.0 V.

Parameter	Symbol	Minimum	Typical	Maximum	Units	Notes
Wakeup Delay from Rest Mode Due to Motion	T_{WUPP}		1	2	ms	
Power Up Delay	T_{PUP}			50	ms	
Debounce Delay on Button Inputs	T_{DBB}	5	9	17	ms	"Maximum" specified at 8 ms polling rate.
Scroll Wheel Sampling Period	T_{SW}	150	200	300	µs	ZA PIN
Transient Supply Current	I_{DDT}			60	mA	Max. supply current during a V_{DDA5} ramp from 0 to 5.0 V with > 500 µs rise time. Does not include charging currents for bypass capacitors.

USB Electrical Specifications

Electrical Characteristics over recommended operating conditions.

Parameter	Symbol	Minimum	Maximum	Units	Notes
Output Signal Crossover Voltage	V_{CRS}	1.5	2.0	V	C_L = 200 to 600 pF (see Figure 10)
Input Signal Crossover Voltage	V_{ICRS}	1.2	2.1	V	C_L = 200 to 600 pF (see Figure 10)
Output High	V_{OH}	2.8	3.6	V	with 15 kΩ to Ground and 7.5 kΩ to V_{BUS} on D- (see Figure 11)
Output Low	VOL	0.0	0.3	V	with 15 kΩ to Ground and 7.5 kΩ to V_{BUS} on D- (see Figure 11)
Single Ended Input	V_{SEI}		0.8	V	
Input High (Driven)	V_{IH}	2.0		V	
Input High (Floating)	V_{IHZ}	2.7	3.6	V	
Input Low	V_{IL}		0.8	V	7.5 kΩ to VDDA5
Differential Input Sensitivity	V_{DI}	0.2		V	|(D+)-(D-)| See Figure 12
Differential Input Common Mode Range	V_{CM}	0.8	2.5	V	Includes V_{DI}, See Figure 12
Single Ended Receiver Threshold	V_{SE}	0.8	2.0	V	
Transceiver Input Capacitance	C_{IN}		12	pF	D+ to V_{BUS}, D- to V_{BUS}

USB Timing Specifications

Timing Specifications over recommended operating conditions.

Parameter	Symbol	Minimum	Maximum	Units	Notes
D+/D- Transition rise time	T_{LR}	75		ns	C_L = 200 pF (10% to 90%), see Figure 10
D+/D- Transition rise time	T_{LR}		300	ns	C_L = 600 pF (10% to 90%), see Figure 10
D+/D- Transition fall time	T_{LF}	75		ns	C_L = 200 pF (90% to 10%), see Figure 10
D+/D- Transition fall time	T_{LF}		300	ns	C_L = 600 pF (90% to 10%), see Figure 10
Rise and Fall time matching	T_{LRFM}	80	125	%	T_R/T_F; C_L = 200 pF; Excluding the first transition from the Idle State
Wakeup delay from USB suspend mode due to buttons push	T_{WUPB}		17	ms	Delay from button push to USB operation Only required if remote wakeup enabled
Wakeup delay from USB suspend mode due to buttons push until accurate navigation	T_{WUPN}		50	ms	Delay from button push to navigation operation. Only required if remote wakeup enabled
USB reset time	T_{reset}	18.7		µs	
Data Rate	t_{LDRATE}	1.4775	1.5225	Mb/s	Average bit rate, 1.5 Mb/s +/- 1.5%
Receiver Jitter Tolerance	t_{DJR1}	-75	75	ns	To next transition, see Figure 13
Receiver Jitter Tolerance	t_{DJR2}	-45	45	ns	For paired transitions, see Figure 13
Differential to EOP Transition Skew	t_{LDEOP}	-40	100	ns	See Figure 14
EOP Width at Receiver	t_{LEOPR}	670		ns	Accepts EOP, see Figure 14
Source EOP Width	t_{LEOPT}	1.25	1.50	µs	
Width of SE0 interval during Differential Transition	t_{LST}		210	ns	See Figure 11.
Differential Output Jitter	t_{UDJ1}	-95	95	ns	To next transition, see Figure 15
Differential Output Jitter	t_{UDJ2}	-150	150		For paired transitions, see Figure 15

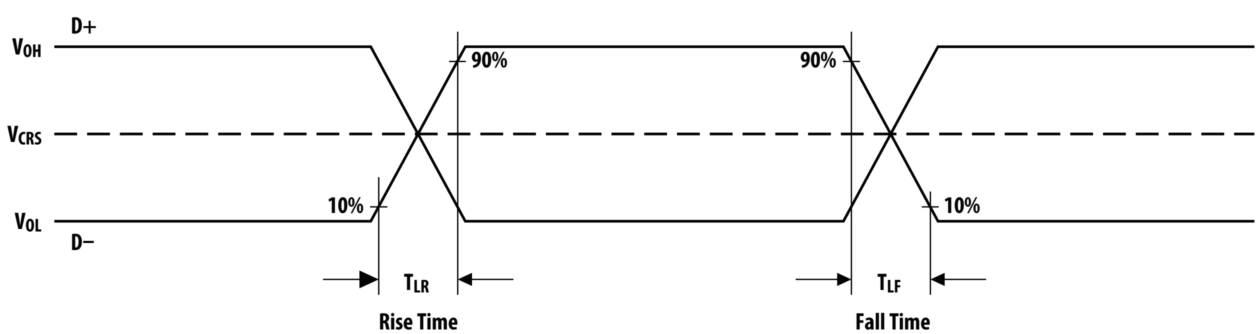


Figure 10. Data Signal Rise and Fall Times

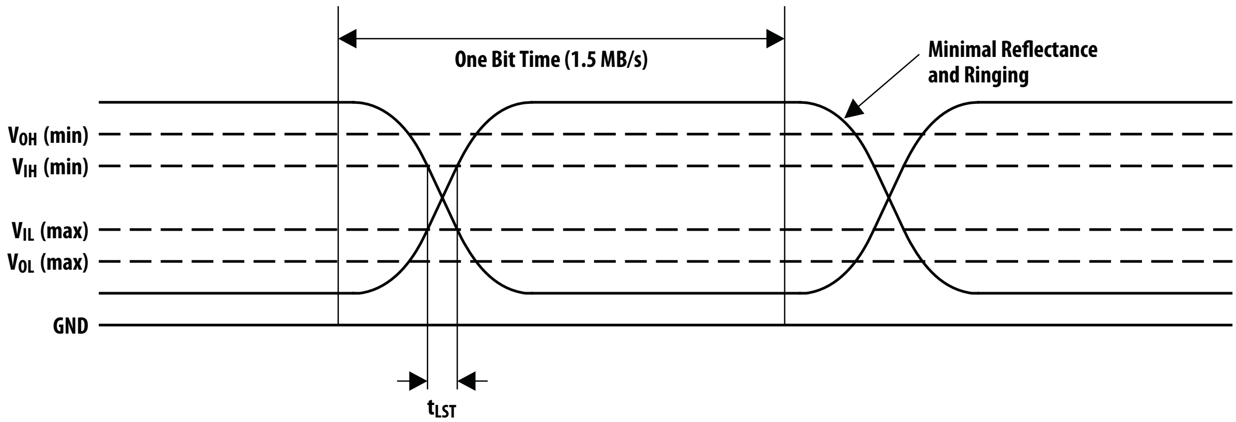


Figure 11. Data Signal Voltage Levels

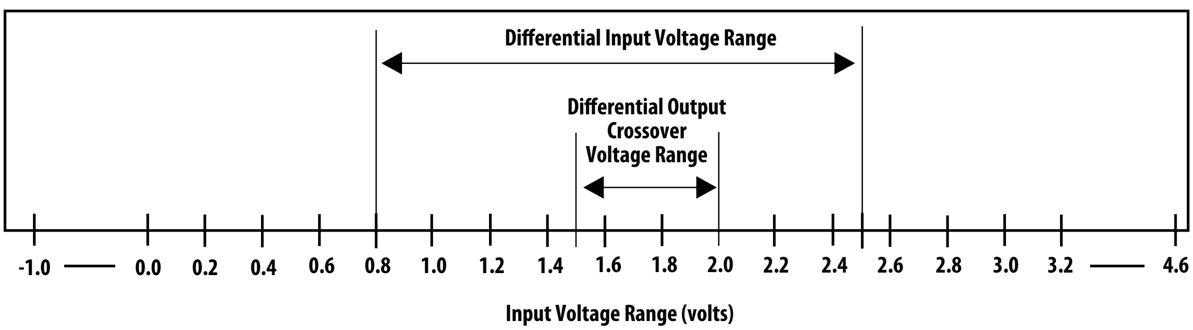


Figure 12. Differential Receiver Input Sensitivity vs. Common Mode Input Range

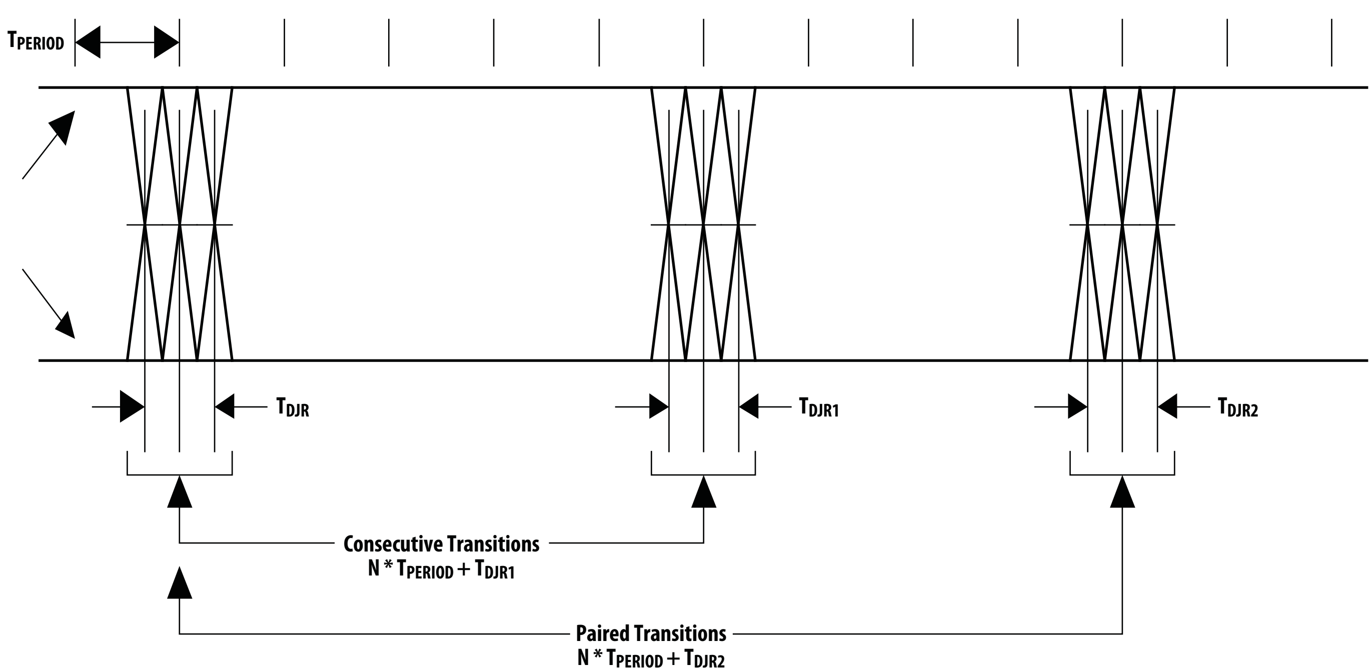


Figure 13. Receiver Jitter Tolerance

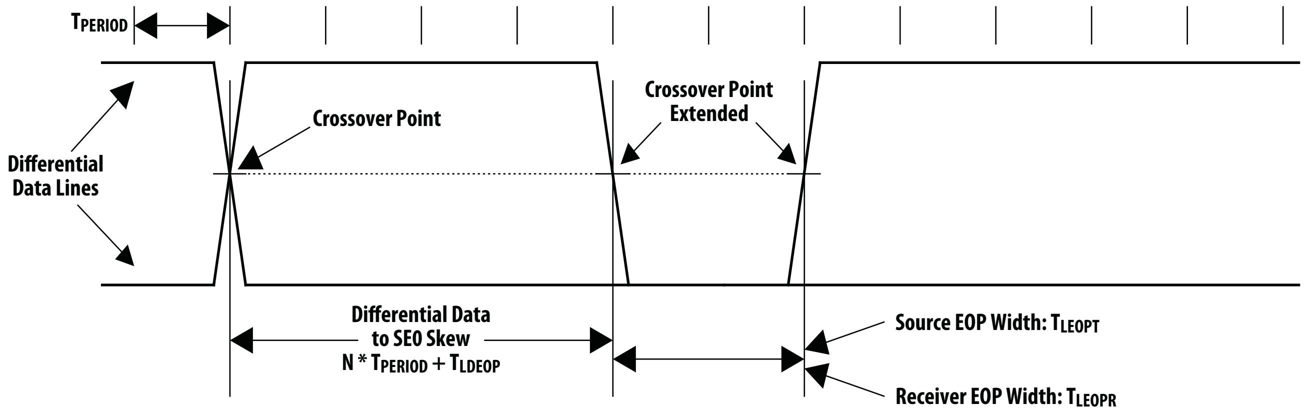


Figure 14. Differential to EOP Transition Skew and EOP Width

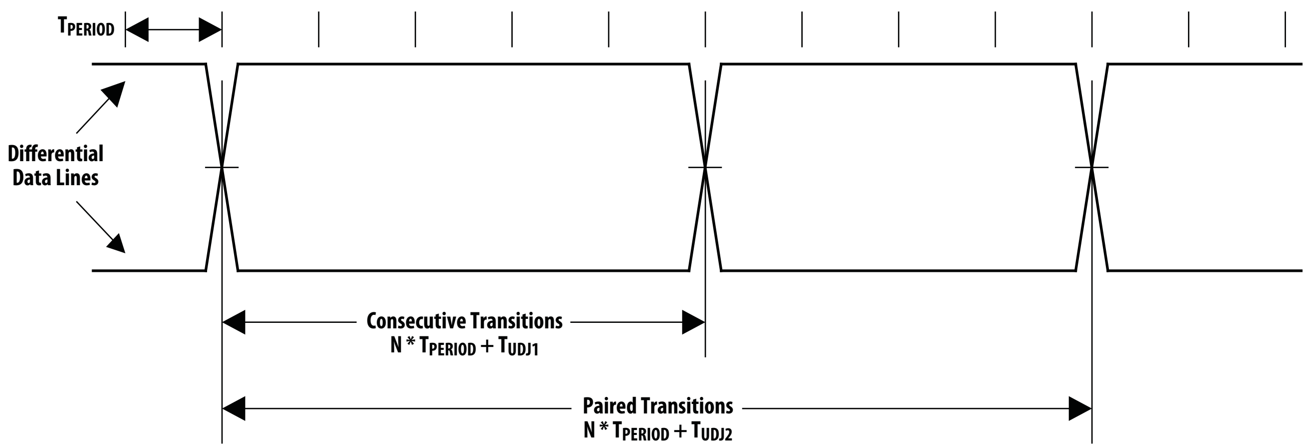


Figure 15. Differential Output Jitter

DC Electrical Specifications

Electrical Characteristics over recommended operating conditions. Typical values at 25° C, V_{DDA5} = 5.0 V.

Parameter	Symbol	Minimum	Typical	Maximum	Units	Notes
System Current, Mouse Moving	I_{DD5}		35	50	mA	Includes XY_LED current
System Current, Mouse Not Moving	I_{DD5N}		15	30	mA	Includes XY_LED current
System Current, USB Suspend Mode, Remote Wakeup Enabled	I_{DD5S}			500	µA	Includes XY_LED current and D- pullup resistor.
Supply Current (Sensor only), Mouse Moving	I_{DDS}		4.5	8	mA	No load on SW, XY-LED, ZA, ZB, D+, D-
Supply Current (Sensor only), Mouse Not Moving	I_{DDSN}		3.9	7.5	mA	No load on SW, XY-LED, ZA, ZB, D+, D-
Sensor Supply Current, USB Suspend Mode	I_{DDSS}			320	µA	No load on SW, XY-LED, ZA, ZB, D+,D-
XY_LED Current	I_{LED}			30	mA	
XY_LED Output Low Voltage	V_{OL}			1.1	V	
Input Low Voltage	V_{IL}			0.5	V	Pins: ZA, ZB V_{IL} max of 0.5 V_{DC} is at V_{DDA5} min of 4 V $_{DC}$, with a typical of 0.8 V_{DC} at V_{DDA5} of 5 V_{DC}
Input High Voltage	V_{IH}	0.6*V_{DDA5}			V	Pins: ZA, ZB

One-Time-Programmable (OTP) Memory

The on chip OTP memory allows device configuration flexibility to override the default setting of sensors without any external software driver. Once the OTP operation is enabled, all OTP registers must be programmed accordingly as the default values of un-program OTP registers are always zero when L1_USE_OTP register setting is not zero value. Tips: OTP write to the OTP register can be skipped if the setting is zero value (0x00) in order to save the OTP programming time.

OTP address space is from 0xDF to 0xE8. OTP can be programmed via USB interface using Set Vendor Test and Get Vendor Test commands.

OTP Byte Write Operation

OTP write operation flow chart is shown in Figure 16.

1. Set OTP Clock enable bit in OTP_CLOCK register, 0x42: OTP_CLOCK_EN = 1.
2. Set OTP enable bit in OTP_CONFIG register, 0x51: OTP_EN = 1.
3. Write the OTP register address byte to OTP_ADDR register, 0x52.
4. Write the OTP data byte to OTP_DATA register, 0x53.
5. Set write enable bit in OTP_CTRL register, 0x54 to enable write command to OTP: WR = 1.
6. Read the write enable bit status in OTP_CTRL register, 0x54. If WR = 1, repeat reading the bit status until it is clear.
7. Read the write status bit in OTP_CTRLSTAT register, 0x58.
   a. If WR_OK = 1, OTP write operation is completed. Repeat Step 2 for more OTP byte write operations.
   b. If WR_OK = 0, repeat Step 5.
8. If Step 6b is repeated up to 10 times, OTP write operation is failed and the chip is confirmed as defective unit.

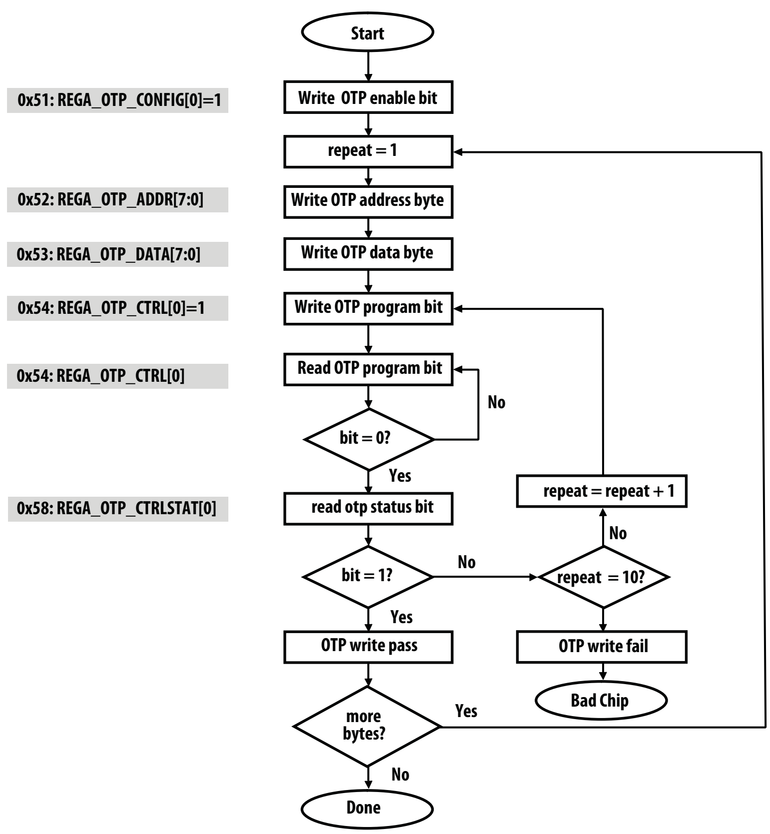


Figure 16. OTP Byte Write Flow Chart

OTP Byte Read Operation

OTP read operation flow chart is shown in Figure 17.

1. Set OTP Clock enable bit in OTP_CLOCK register, 0x42: OTP_CLOCK_EN = 1.
2. Set OTP enable bit in OTP_CONFIG register, 0x51: OTP_EN = 1.
3. Write the OTP register address byte to OTP_ADDR register, 0x52.
4. Set read enable bit in OTP_CTRL register, 0x54 to enable write command to OTP: RD = 1.
5. Read the read enable bit status in OTP_CTRL register, 0x54. If RD = 1, repeat reading the bit status until it is clear. Read the OTP data byte from OTP_DATA register, 0x53 to complete the OTP read operation.
6. Read the OTP data byte from OTP_DATA register, 0x53 to complete the OTP read operation.
7. Repeat Step 2 for more OTP read operations

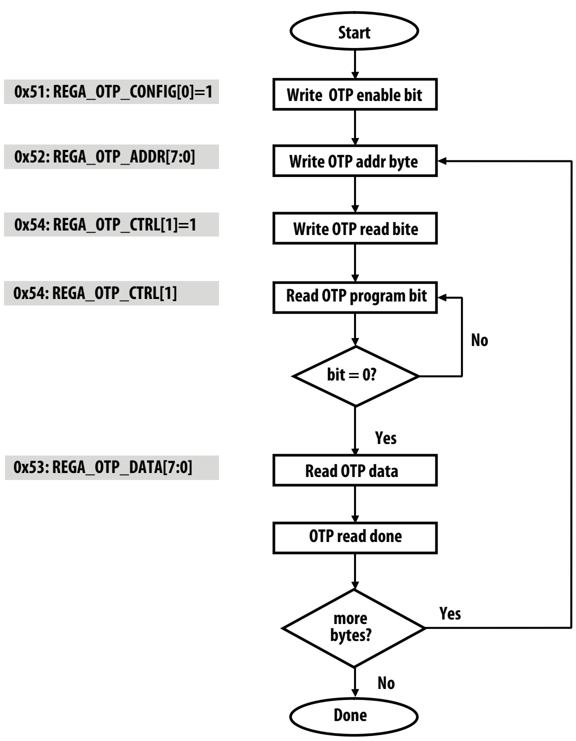


Figure 17. OTP Byte Read Flow Chart

OTP Lock Operation

OTP lock operation MUST be performed once OTP write to OTPLOCK1 register for the sensor to function. DO not reset or power up the chip right after OTP write to OTPLOCK1 register, otherwise the chip will be malfunction. The OTP lock operation flow chart is shown in Figure 18.

1. Set OTP Clock enable bit in OTP_CLOCK register, 0x42: OTP_CLOCK_EN = 1
2. After OTP write to OTPLOCK1 register, set OTP enable bit in OTP_CONFIG register, 0x51: OTP_EN = 1.
3. Set OTP lock bit in OTP_CTRL register, 0x54 to enable OTP lock command: LOCK_L1 = 1.
4. Read the OTP lock bit status in OTP_CTRL register, 0x54. If LOCK_L1 = 1, repeat reading the bit status until it is clear.
5. Read the lock status and CRC bits in OTP_CTRLSTAT register, 0x58.
   a. If both L1_LOCK_OK and L1_CRC_OK = 1, OTP lock operation is completed.
   b. If either L1_LOCK_OK or L1_CRC_OK = 0, repeat Step 2 until both bits are set.
6. If Step 4b is repeated up to 10 times, OTP lock operation is failed and the chip is confirmed as defective unit.
7. Read the CRC result stored in register 0xE9, 0xEA, 0xEB, 0xEC, if four register values not 0x00 means CRC has been generated correctly and verified as lock operation success.

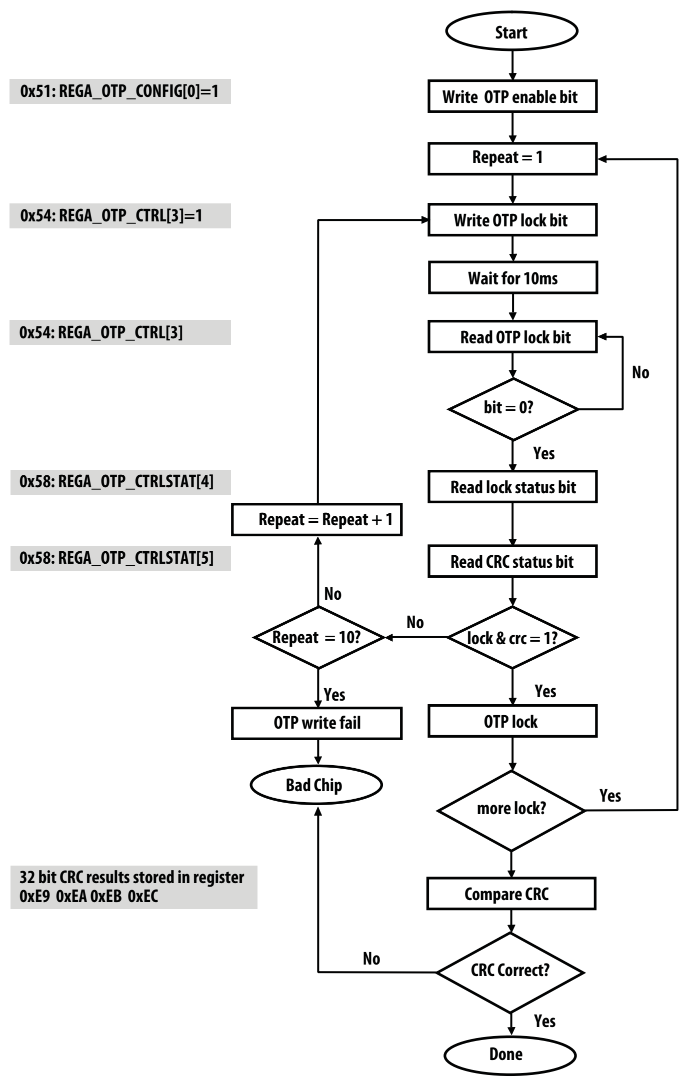


Figure 18. OTP Byte Lock Flow Chart

Buttons

The minimum time between button pressed is T_{DBB}. The button connection is described in Figure 19

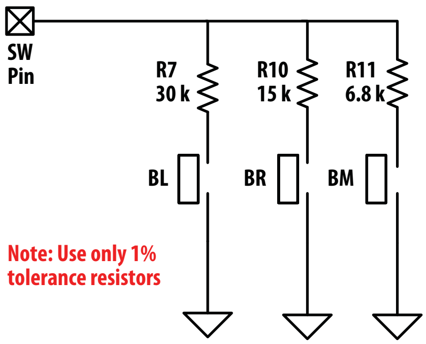


Figure 19. Button connections

Debounce Algorithm

- Button inputs B1, B2, and B3 are sampled every 6ms.
- Two consecutive low values create a button press event.
- Three consecutive high values create a button release event.

Configuration after Power up (Data Values)

Signal Function	State from Figure 9-1 of USB spec: Powered or Default Address or Configured	State from Figure 9-1 of USB spec: Suspended from Any Other State
SW	Output voltage at 1.16 V (Typ)	Output voltage at 2.7 V (Typ)
D-	USB I/O	Hi-Z Input
D+	USB I/O	Hi-Z Input
XY_LED	Always ON / Pulsing	Pulled HIGH (OFF)
ZB	Hi-Z Input	Output HIGH
ZA	Hi-Z Input	Output HIGH

Typical Performance Characteristics

Performance Characteristics over recommended operating conditions. Typical values at 25° C, V_{DD} = 5.0 V, 24 MHz

Parameter	Symbol	Minimum	Typical	Maximum	Units	Notes
Path Error (Deviation)	P_{Error}		0.5		%	Average path error as percent of total 2.5" travel on various standard surfaces

The following graphs are the typical performance of the sensor, assembled as shown in the 2D assembly drawing with trim lens, clip, and LED.

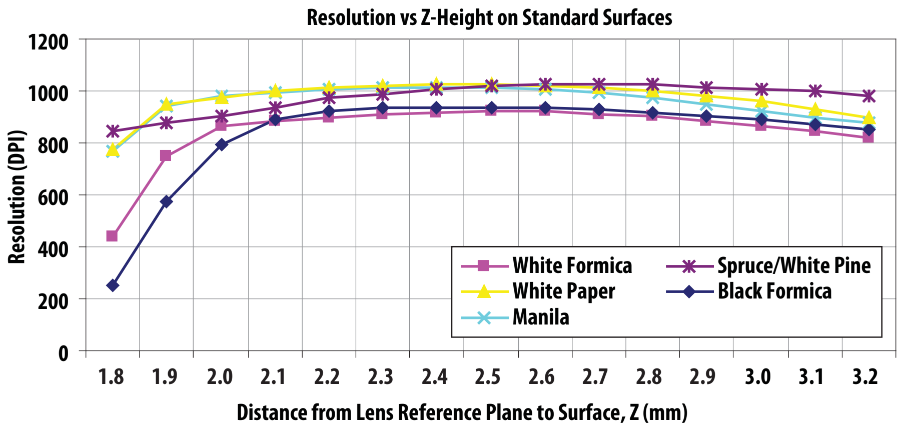


Figure 20. Typical Resolution vs. Z Height

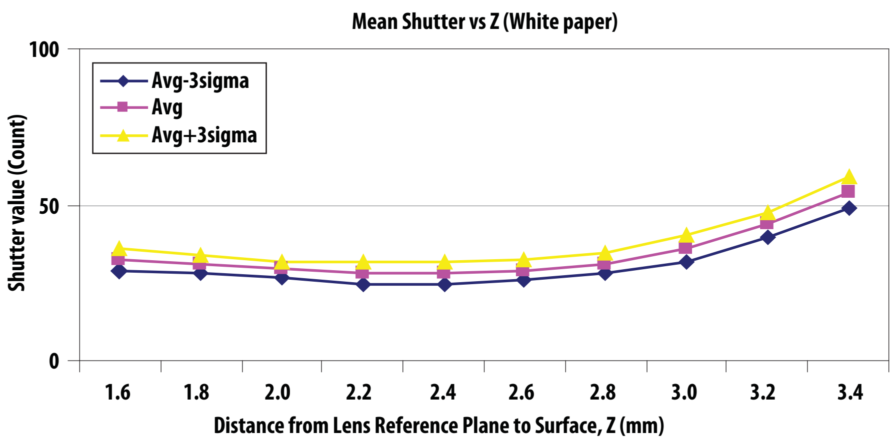


Figure 21. Mean shutter vs Z height over white paper

Notes:

1. The sensor is designed for optimal performance when used with the specified LED.
2. Z = distance from Lens Reference Plane to Surface.

USB Commands

Mnemonic	Command	Notes
USB_RESET	D+/D- low > 18.7 µs	Device Resets; Address=0
USB_SUSPEND	Idle state > 3 mS	Device enters USB low-power mode
USB_RESUME	Non-idle state	Device exits USB low-power mode
Get_Status_Device	80 00 00 00 00 00 02 00	Normally returns 00 00, Self powered 00 00, Remote wakeup 02 00
Get_Status_Interface	81 00 00 00 00 00 02 00	Normally returns 00 00
Get_Status_Endpt0	82 00 00 00 xx 00 02 00	OUT: xx=00, IN: xx=80 Normally returns 00 00
Get_Status_Endpt1	82 00 00 00 81 00 02 00	Normally returns 00 00, Halt 00 01
Get_Configuration	80 08 00 00 00 00 01 00	Return: 00=not config., 01=configured
Get_Interface	81 0A 00 00 00 00 01 00	Normally returns 00
Get_Protocol	A1 03 00 00 00 00 01 00	Normally returns 01, Boot protocol 00
Get_Desc_Device	80 06 00 01 00 00 nn 00	See USB command details
Get_Desc_Config	80 06 00 02 00 00 nn 00	See USB command details
Get_Desc_String	80 06 xx 03 00 00 nn 00	See USB command details
Get_Desc_HID	81 06 00 21 00 00 09 00	See USB command details
Get_Desc_HID_Report	81 06 00 22 00 00 nn 00	See USB command details
Get_HID_Input	A1 01 00 01 00 00 nn 00	Return depends on motion & config
Get_Idle	A1 02 00 00 00 00 01 00	Returns rate in multiples of 4 ms
Get_Vendor_Test	C0 01 00 00 xx 00 01 00	Read register xx
Set_Address	00 05 xx 00 00 00 00 00	xx = address
Set_Configuration	00 09 xx 00 00 00 00 00	Not configured: xx=00 Configured: xx=01
Set_Interface	01 0B 00 00 00 00 00 00	Only one interface supported
Set_Protocol	21 0B xx 00 00 00 00 00	Boot: xx=00, Report: xx=01
Set_Feature_Device	00 03 01 00 00 00 00 00	Enable remote wakeup
Set_Feature_Endpt0	02 03 00 00 xx 00 00 00	Halt. OUT: xx=00, IN: xx=80
Set_Feature_Endpt1	02 03 00 00 81 00 00 00	Halt
Clear_Feature_Device	00 01 01 00 00 00 00 00	Disable Remote wakeup
Clear_Feature_Endpt0	02 01 00 00 xx 00 00 00	Clear Halt; OUT: xx=00, IN: xx=80
Clear_Feature_Endpt1	02 01 00 00 81 00 00 00	Clear Halt
Set_Idle	21 0A 00 rr 00 00 00 00	rr = report rate in multiples of 4 ms
Set_Vendor_Test	40 01 00 00 xx yy 00 00	Write yy to address xx
Poll_Endpt1		Read buttons, motion, & Z-wheel

Note:
The last two bytes in a command shown as "nn 00" specify the 16-bit data size in the order of "LowByte HighByte." For example a two-byte data size would be specified as "02 00." The sensor will not provide more bytes than the number requested in the command, but it will only supply up to a maximum of 8 bytes at a time. The sensor will re-send the last packet if the transfer is not acknowledged properly.

USB COMMAND DETAILS

USB_RESET	D+/D- low for an extended period
USB Spec:	A device may reset after seeing an SE0 for more than 18.7 uS, and definitely after 10 mS.
Notes:	After power up and prior to Reset, the device will not respond to any USB commands. After the device has been given a USB Reset, the device's address will be reset to zero and the device will be in the Default state. The chip will default to Report protocol and any pending output will be flushed. All registers will be reset to a state that matches power-on-reset with the following exceptions: USB State register will be "Default" instead of "Attached".
USB_SUSPEND	Idle state for an extended period
USB Spec:	A device may suspend after seeing an idle for more than 3 mS, and definitely after 10 mS.
Notes:	The chip will take a minimum of 5 mS to start Suspend, though will definitely start after 6 mS. The chip may finish the current frame if necessary before stopping the clock. Thus, an additional frame time may be used to reach Suspend mode.
USB_RESUME	Non-idle state
USB Spec: Notes:	Remote Resume signalling from a device must be between 1 mS and 15 mS. The host is required to send Resume signaling for 20 mS plus 10 mS of resume recovery time in which it does not access any devices. This allows devices enough time to wake back up. The chip can cause a Resume if Remote Wakeup is enabled and a button has been pressed. Remote resume signalling from the chip will last 11.45 mS to 12.45 mS.
Get_Status_Device	80 00 00 00 00 00 02 00
Returns:	xx yy xx[0] = Self Powered xx[1] = Remote Wakeup xx[7:2] = 0 yy = 00 (Reserved)
Default:	Accept (undefined in USB Spec)
Addressed:	Accept
Configured:	Accept
Notes:	Use Set_Feature_Device/Clear_Feature_Device to set/clear remote wakeup.
Get_Status_Interface	81 00 00 00 00 00 02 00
Returns:	00 00
Default:	Stall (undefined in USB Spec)
Addressed:	Stall
Configured:	Accept
Notes:	Both return bytes are reserved and currently 00.

Get_Status_Endpt0	82 00 00 00 xx 00 02 00 82 00 00 00 00 00 02 00 82 00 00 00 80 00 02 00 xx = 00 = Endpt0 OUT xx = 80 = Endpt0 IN
Returns:	xx yy xx[0] = Halt xx[7:1] = 0 yy = 00 (Reserved)
Default:	Accept (undefined in USB Spec)
Addressed:	Accept
Configured:	Accept
Notes:	Use Set_Feature_Endpt0/Clear_Feature_Endpt0 to (try to) set/clear Halt bit. According to USB, “It is neither required or recommended that the Halt feature be implemented for the Default Control Pipe.” Since a new SETUP command will clear any Endpt0 halt bit, it is impossible to tell if there really is a halt bit.
Get_Status_Endpt1	82 00 00 00 81 00 02 00
Returns:	xx yy xx[0] = Halt xx[7:1] = 0 yy = 00 (Reserved)
Default:	Stall (undefined in USB Spec)
Addressed:	Stall
Configured:	Accept
Notes:	Use Set_Feature_Endpt1/Clear_Feature_Endpt1 to set/clear Halt bit.
Get_Configuration	80 08 00 00 00 00 01 00
Returns:	xx xx = config value
Default:	Accept (undefined in USB Spec) – returns 00
Addressed:	Accept – returns 00
Configured:	Accept – returns 01
Notes:	Use Set_Configuration to change.
Get_Interface	81 0A 00 00 00 00 01 00
Returns:	00
Default:	Stall (undefined in USB Spec)
Addressed:	Stall
Configured:	Accept – returns 00
Notes:	Command has no alternate interfaces, so only valid value is 00

Get_Protocol	A1 03 00 00 00 00 01 00
Returns:	xx xx = 00 = Boot protocol xx = 01 = Report protocol
Default:	Accept
Addressed:	Accept
Configured:	Accept
Notes:	Defaults to Report protocol after USB Reset. Use Set_Protocol to change.
Get_Desc_Device	80 06 00 01 00 00 nn 00 80 06 00 01 00 00 12 00
Returns:	12 01 00 02 00 00 00 08 vv vv pp pp dd dd mm PP ss 01
Example:	vv vv = vendor id pp pp = product id (vendor specified) dd dd = device id (vendor specified) (bcd rev_id byte) mm = iManufacturer PP = iProduct ss = iSerialNumber (00 – no string)
Example for Multi-button:	12 01 00 02 00 00 00 08 6D 04 pp pp 00 54 01 02 00 01 // Device Descriptor | 12 // bLength (18 decimal) | 01 // bDescriptorType | 00 // bcdUSB (Release ##.## = 02.00) | 02 | 00 // bDeviceClass | 00 // bDeviceSubClass | 00 // bDeviceProtocol | 08 // bMaxPacketSize0 | 2F // idVendor | 19 // idVendor | 16 // idProduct // based on #buttons & wheel | 09 // idProduct | 00 // bcdDevice (Dev Rel 54.00) | 54 | 00 // iManufacturer | 02 // iProduct | 00 // iSerialNumber | 01 // bNumConfigurations
Default:	Accept
Addressed:	Accept
Configured:	Accept Get_Desc_String will return "stall" if Manufacturer string is queried when iManufacturer = 0x00.

Get_Desc_Config	80 06 00 02 00 00 nn 00 80 06 00 02 00 00 22 00
Returns:	09 02 22 00 01 01 00 A0 32 09 04 00 00 01 03 01 02 00 09 21 11 01 00 01 22 rr 00 07 05 81 03 05 00 0A rr = HID Report descriptor length 47 = 12 bit motion reporting
Default:	Accept
Addressed:	Accept
Configured:	Accept
Notes:	This is the concatenation of 4 descriptors: Configuration Interface HID Endpt
Get_Desc_String	80 06 xx 03 00 00 nnl nnh xx= 00 Language String 01 Manufacturer String 02 Product String
Command Option:	xx= 00 => Language String 01 => Manufacturer String 02 => Product String Nnl nnh = varies with the string length
Returns:	ss 03 “unicode string” ss = String descriptor length
For xx = 00:	04 03 09 04 // Language ID
For xx = 01:	default: stall

Product String (xx=02)

Product String	**Returns**
USB Optical Mouse	24 03 55 00 53 00 42 00 20 00 4f 00 70 00 74 00 69 00 63 00 61 00 6c 00 20 00 4d 00 6f 00 75 00 73 00 65 00
Synopsys cmd:	No
Default:	Accept
Addressed:	Accept
Configured:	Accept

Notes:

1. A request for any other string will STALL.
2. Returned string depends on the manufacturer string section via OTP.

Get_Desc_HID	81 06 00 21 00 00 09 00
Returns:	09 21 11 01 00 01 22 rr 00 rr = HID Report descriptor length 40 = 12bit reporting
Get_Desc_HID_Report	81 06 00 22 00 00 nn 00
Returns:	This returns a report descriptor that describes how many buttons and x, y, z data.
12 bit reporting:	05 01 09 02 A1 01 09 01 A1 00 05 09 19 01 29 03 15 00 25 01 75 01 95 03 81 02 75 05 95 01 81 01 05 01 09 30 09 31 16 01 F8 26 FF 07 75 0C 95 02 81 06 09 38 15 81 25 7F 75 08 95 01 81 06 C0 C0

```
            //    HID Report
| 05 01     //    USAGE_PAGE (Generic Desktop)
| 09 02     //    USAGE (Mouse)
| A1 01     //    COLLECTION (Application)
| 09 01     //    USAGE (Pointer)
| A1 00     //    COLLECTION (Physical)
| 05 09     //    USAGE_PAGE (Button)
| 19 01     //    USAGE_MINIMUM (Button 1)
| 29 03     //    USAGE_MAXIMUM (Button #3)
| 15 00     //    LOGICAL_MINIMUM (0)
| 25 01     //    LOGICAL_MAXIMUM (1)
| 75 01     //    REPORT_SIZE (1)
| 95 03     //    REPORT_COUNT (3)
| 81 02     //    INPUT (Data,Var,Abs)
| 75 05     //    USAGE PAGE
| 95 01     //    REPORT COUNT(1)
| 81 01     //    INPUT (CNST,ARR,ABS)
| 05 01     //    USAGE PAGE (Generic Desktop)
| 09 30     //    USAGE (X)
| 09 31     //    USAGE (Y)
| 16 01 F8  //    LOGICAL MINIMUM (-127)
| 26 FF 07  //    LOGICAL MAXUMUN (128)
| 75 0C     //    REPORT_SIZE (12)
| 95 02     //    REPORT_COUNT (2)
| 81 06     //    INPUT (Data,Var,Rel)
| 09 38     //    USAGE (Zwheel)
| 15 81     //    LOGICAL MINIMUM(-127)
| 25 7F     //    LOGICAL MAXIMUM(127)
| 75 08     //    REPORT_SIZE(8)
| 95 01     //    REPORT_COUNT(1)
| 81 06     //    INPUT(Data,Var,Rel)
| C0        //    END_COLLECTION
| C0        //    END_COLLECTION
```

Default:	Accept
Addressed:	Accept
Configured:	Accept
Notes:	The length of this report is needed in the HID descriptor.

Get_HID_Input	A1 01 00 01 00 00 nn 00 nn = 06 (12 bit reporting)
Returns:	bb xx yx yy zz bb = button byte xx = X motion byte yx = XY motion byte yy = Y motion byte zz = Z motion byte
Default:	Stall
Addressed:	Stall
Configured:	Accept
Notes:	If the device is configured, it will always respond with a report for this command, even if no motion or button changes have occurred. In this case, it would report 00 for motion and simply report the current button state. If a report is pending on endpt1, the data there will be reported and the report on endpt1 cleared. The mouse will only create new button/motion packets when it is in the Configured state

USB Data Packet Format

Bit	7	6	5	4	3	2	1	0
Byte 1	0	0	0	0	0	B3(MB)	B2(RB)	B1(LB)
Byte 2	X7	X6	X5	X4	X3	X2	X1	X0
Byte 3	Y3	Y2	Y1	Y0	X11	X10	X9	X8
Byte 4	Y11	Y10	Y9	Y8	Y7	X6	X5	X4
Byte 5	Z7	Z6	Z5	Z4	Z3	Z2	Z1	Z0
Byte 6	0	0	0	0	0	0	0	0

Get_Idle	A1 02 00 00 00 00 01 00
Returns:	rr rr = rate in multiples of 4 mS
Default:	Accept
Addressed:	Accept
Configured:	Accept
Notes:	The third byte of the command is to select the Report ID. There is only one for the mouse – so, using 00 or 01 will work. See also Set_Idle.
Get_Vendor_Test	C0 01 00 00 xx 00 01 00 ii = ignore xx = address of register to read
Returns:	rr (depends on register read)
Default:	Accept
Addressed:	Accept
Configured:	Accept
Notes:	Address range (xx) is datasheet register range