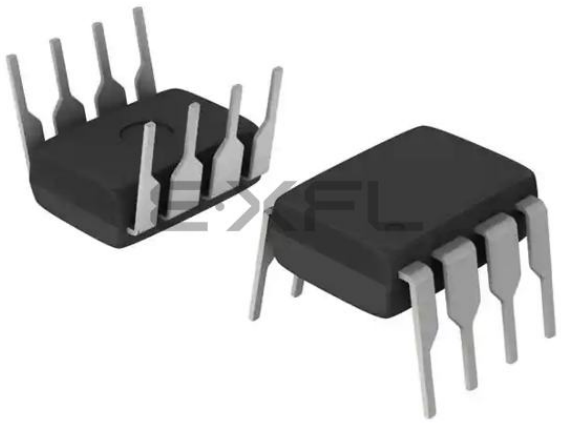


Welcome to E-XFL.COM

What is "Embedded - Microcontrollers"?

"Embedded - Microcontrollers" refer to small, integrated circuits designed to perform specific tasks within larger systems. These microcontrollers are essentially compact computers on a single chip, containing a processor core, memory, and programmable input/output peripherals. They are called "embedded" because they are embedded within electronic devices to control various functions, rather than serving as standalone computers. Microcontrollers are crucial in modern electronics, providing the intelligence and control needed for a wide range of applications.

Applications of "Embedded - Microcontrollers"

Details	
Product Status	Active
Core Processor	PIC
Core Size	8-Bit
Speed	4MHz
Connectivity	-
Peripherals	POR, WDT
Number of I/O	5
Program Memory Size	1.5KB (1K x 12)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	41 x 8
Voltage - Supply (Vcc/Vdd)	2V ~ 5.5V
Data Converters	-
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Through Hole
Package / Case	8-DIP (0.300", 7.62mm)
Supplier Device Package	8-PDIP
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic12f509-i-p

Features (continued)

- Mechanical and optical Z-Wheel interface for vertical scroll
- Tilt-Wheel function for horizontal scroll
- 12-bit Bluetooth HID motion data reporting
- Customizable SDP Service Name, Service Description, Provider Name, VID, PID, & Bluetooth Address
- 4-axis sensor rotations: 0°, 90°, 180° or 270°
- Resolution:
  - Programmable from 250-3000 counts per inch (cpi) with 250cpi incremental step
  - Up to 10 selections of On-the-Fly (OTF) resolution mode setting

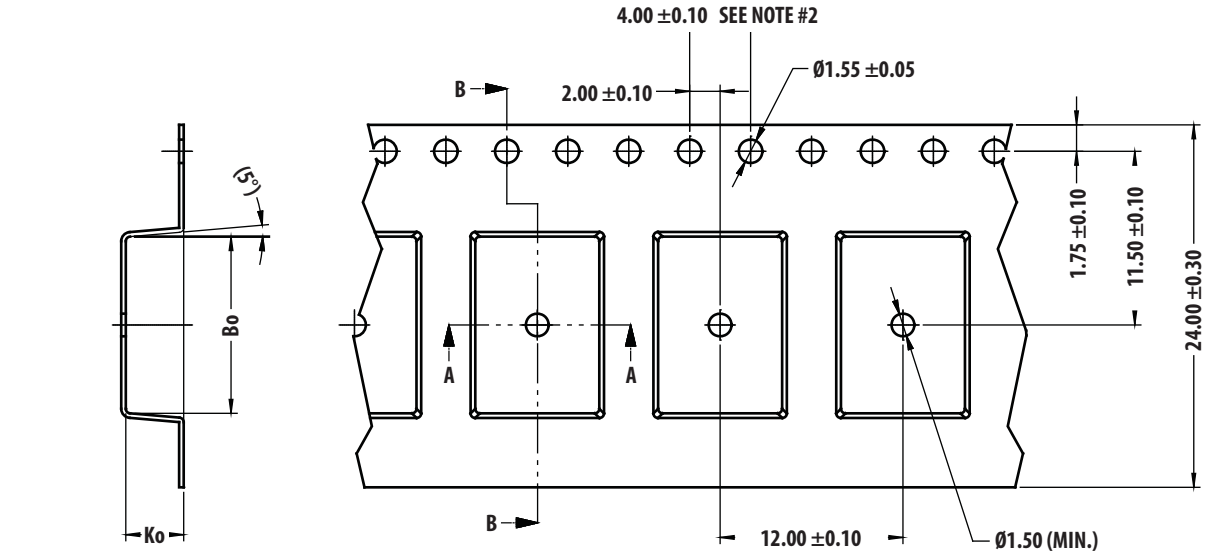
Disclaimer: All designers and manufacturers of final product with tilt wheel enabled must assure that they have all necessary intellectual property rights.

Theory of Operation

ADNS-7630 is based on LaserStream navigation technology that measures changes in position by optically acquiring sequential surface images (per frames) and mathematically determining the direction and magnitude of motion movement. It contains an Image Acquisition System (IAS), a Digital Signal Processor (DSP) and Bluetooth HID stream output. Images acquired by the IAS 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 Bluetooth HID data. The motion data and buttons input status are then transmitted in wireless mode to the Bluetooth.

Ordering Information

Part Number	Packaging Type	Minimum Order Quantity
ADNS-7630	Tube	1000 units per tube
ADNS-7630-TR	Tape and Reel	4000 units per roll



Notes:
1. Ao & Bo measured at 0.3mm above base of pocket.
2. 10 pitches cumulative tol. ±0.2mm.
3. ( ) Reference dimensions only.

Figure 1a. ADNS-7630-TR Tape and Reel Packaging Dimension

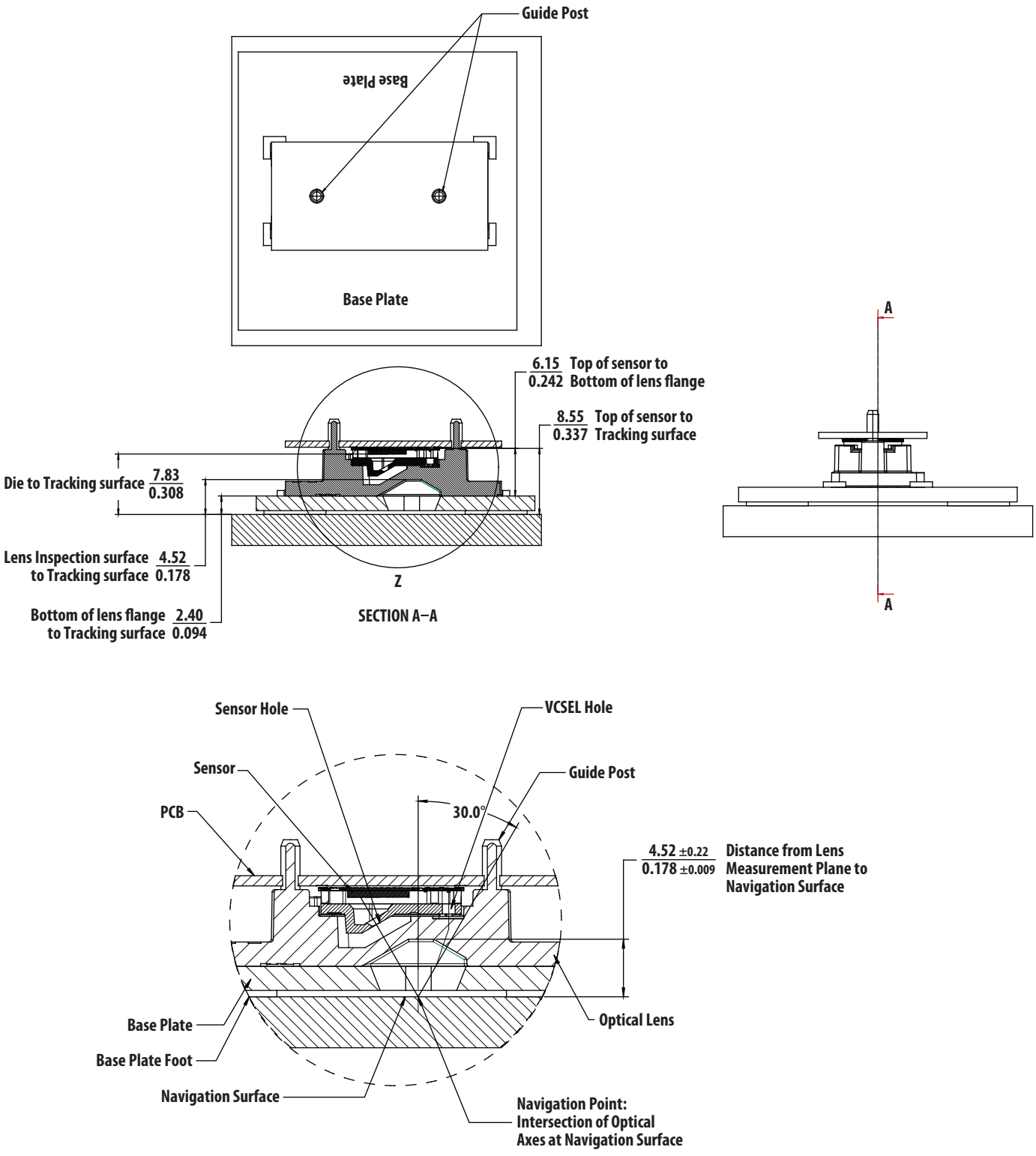


Figure 5. 2D assembly drawing of ADNS-7630 sensor coupled with ADNS-7100-001 lens, PCB & base plate

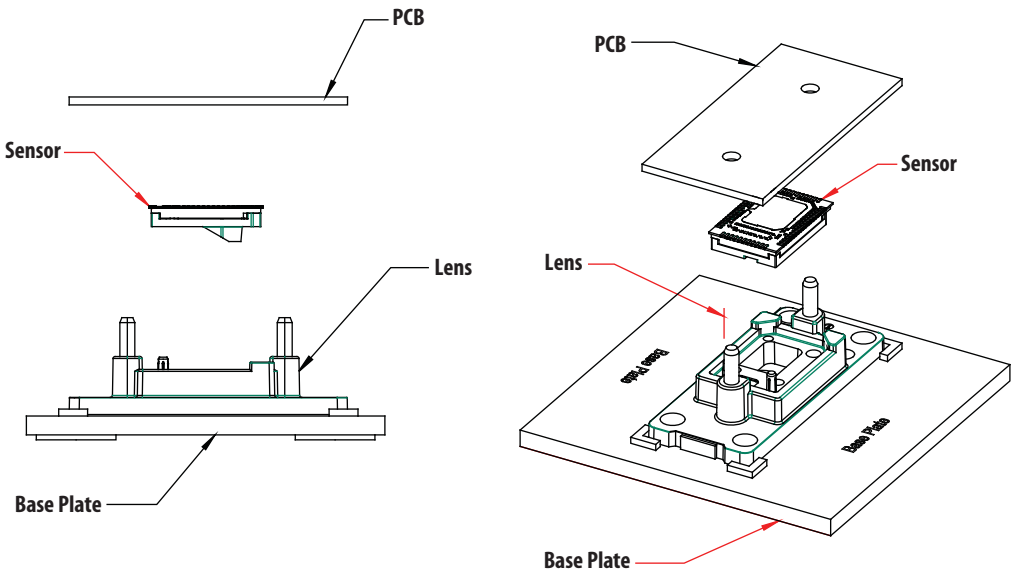


Figure 6. Exploded view drawing of ADNS-7630 sensor coupled with ADNS-7100-001 lens, PCB & base plate (front view and top side view)

As shown above, the components self align as they are mounted onto defined features on the base plate. There should be guide holes on the PCB to align the ADNS-7100-001 lens to the ADNS-7630 sensor's aperture stop. The ADNS-7630 sensor is designed for mounting on the bottom side of a PCB, looking down.

The integrated VCSEL is used for the illumination, provides a laser diode with a single longitudinal and a single transverse mode. Together with the VCSEL contained in the sensor package, the ADNS-7100-001 lens provides directed illumination and optical imaging necessary for the operation of the sensor. The lens is a precision molded optical component and should be handled with care to avoid scratching and contamination on the optical surfaces.

3D drawing files in STEP or IGES format for the sensor, lens and base plate describing the components and base plate molding features for the lens and PCB alignment is available.

Design considerations for improving ESD Performance

The table below shows typical values assuming base plate construction per the Avago Technologies supplied IGES file for ADNS-7100-001 lens. Note that the lens material is polycarbonate and therefore, cyanoacrylate based adhesives should not be used as they will cause lens material deformation.

Typical Distance	Millimeters (mm)
Creepage	11.87
Clearance	10.05

PCB Assembly Considerations and Soldering Profile

1. Prior to PCB assembly, handling precaution must be taken for ADNS-7630 sensor that is classified as MSL-3. (For more information, please refer to IPC/JEDEC J-STD-033B.1: Handling, Packing, Shipping and Use of Moisture/Reflow Sensitive Surface Mount Devices)
2. Surface-mount the sensor package and all other electrical components onto PCB.
3. Reflow the entire assembly with a no-wash solder flux process (refer to Figure 7 below).
4. Remove the protective kapton tapes from both optical apertures on the ADNS-7630 sensor by using flat-headed tweezer. Care must be taken to keep contaminants from entering the aperture. Recommend not to place the PCB facing up during the entire assembly process. Recommend to hold the PCB vertically for the kapton tapes removal process.
5. Place the PCB over the lens onto base plate. The sensor package should be self-aligned to the lens. The optical center reference for the PCB is set by base plate and lens. Note that the PCB movement due to button presses must be minimized to maintain good optical alignment.
6. Recommended: The lens can be permanently located by heat-staking or ultrasonic-staking the lens' guide posts over the PCB board.
7. Then, install the mouse top case. There MUST be feature in the top case (or other area) to press down onto the PCB assembly to ensure the sensor and lens are interlocked to correct vertical height.

Block Diagram

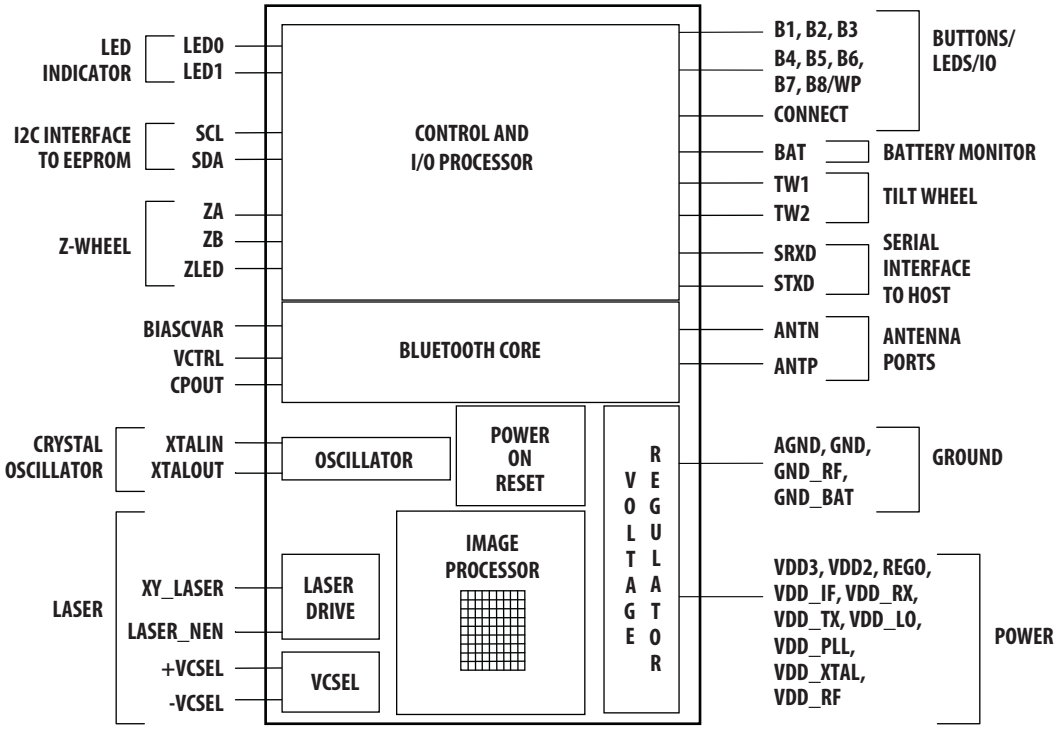


Figure 11. ADNS-7630 Block Diagram

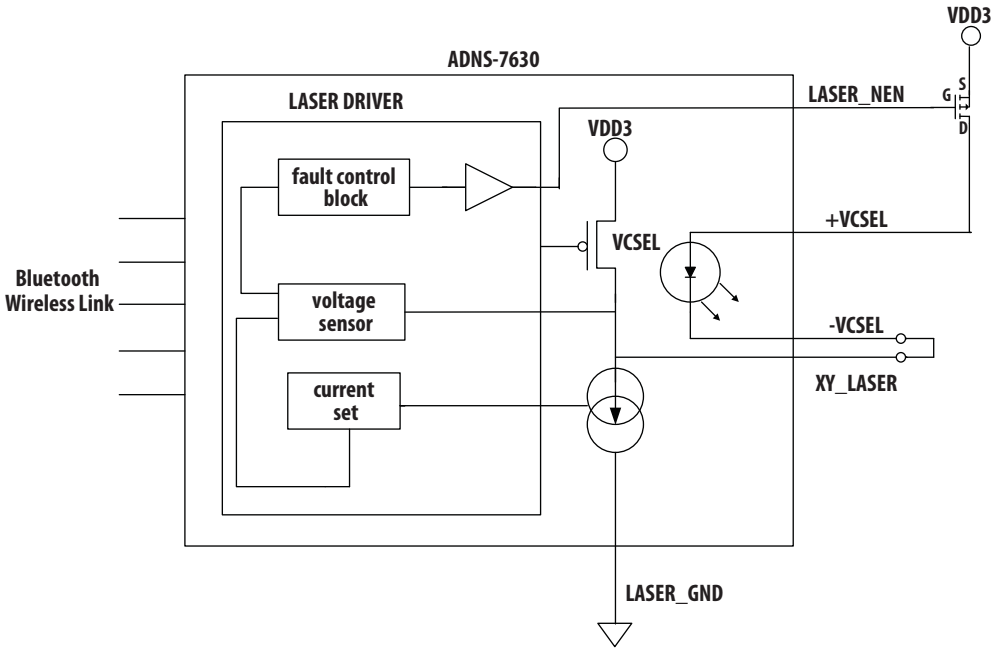


Figure 12. Single Fault Detection and Eye Safety Feature

AC Electrical Specifications

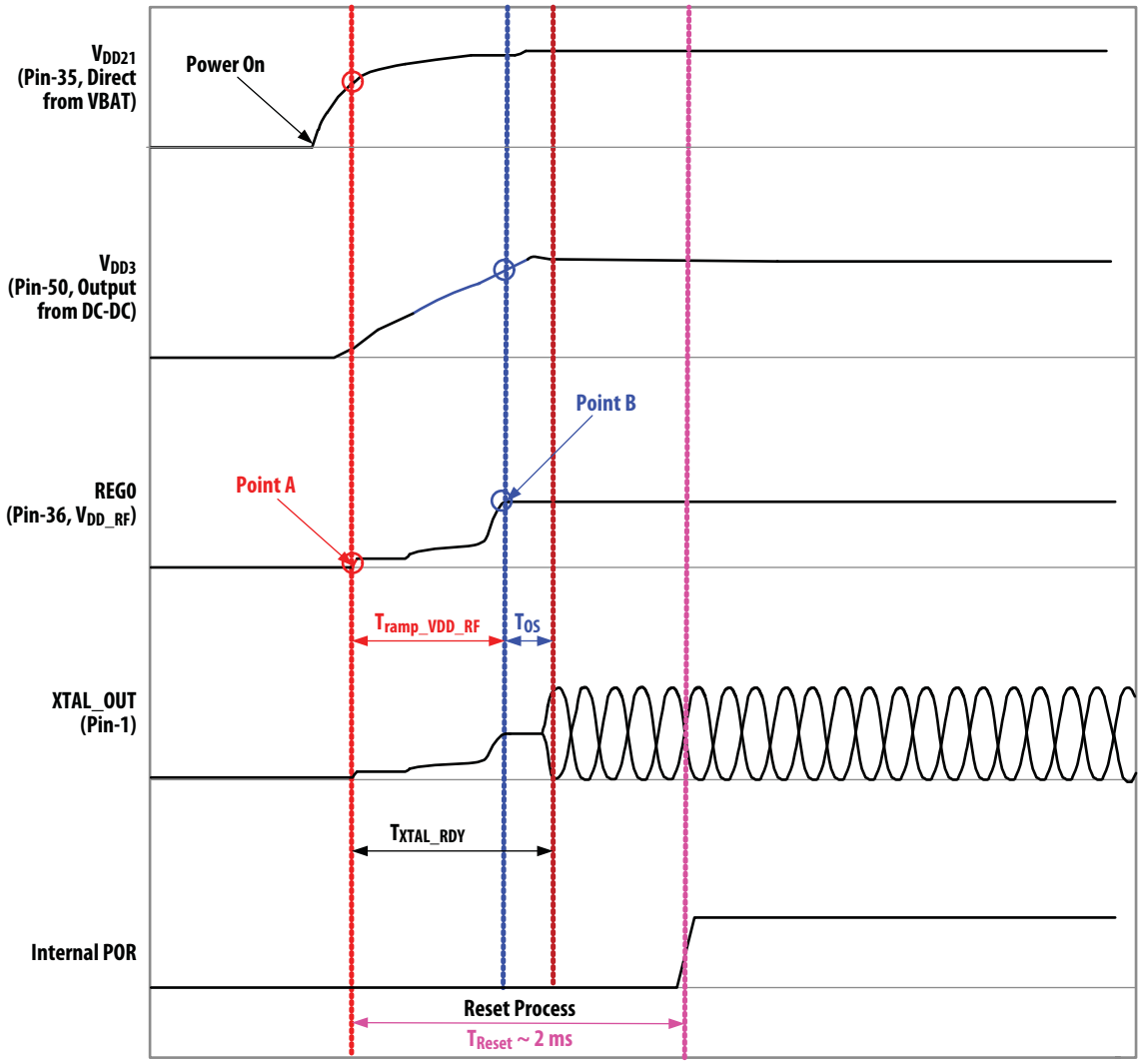
Electrical Characteristics over recommended operating conditions. Typical values at 25 °C, V_{DD21} = 2.8V, V_{DD3} = 2.8V

Parameter	Symbol	Minimum	Typical	Maximum	Units	Notes
Debounce delay on button inputs	t_{DBB}		6	7.9	ms	
Scroll wheel sampling period	t_{SW}	1.9	2.0	2.8	ms	ZA & ZB Pins.
Transient Supply Current	I_{DDT}			100	mA	V_{DD21} is tied to V_{DD3}. Max supply current during a ramp from 0 to 2.8V

DC Electrical Specifications

Electrical Characteristics over recommended operating conditions. Typical values at 25 °C, V_{DD21} = 2.8V, V_{DD3} = 2.8V

Parameter	Symbol	Minimum	Typical	Maximum	Units	Notes
Tx Current	I_{Tx}		53	57.5	mA	Transmitter and baseband are fully ON, navigation core is OFF. Buttons and I/Os are floating, LED pins pull to low
Rx Current	I_{Rx}		47	51	mA	Receiver and baseband are fully ON, navigation core is OFF. Buttons and I/Os are floating, LED pins pull to low
DM1 Tx mode Current	I_{DM1_Tx}		24.7		mA	RF sends a longest DM1 packet every 1.25ms
DM1 Rx mode Current	I_{DM1_Rx}		24.2		mA	RF receives a longest DM1 packet every 1.25ms
Sniff mode 11.25ms Current	$I_{sniff\_11.25ms}$		10	12	mA	System average current includes VCSEL current. Sniff_TimeOut = 0, Sniff_Attempt = 1
Sniff mode 67.5ms Current	$I_{sniff\_67.5ms}$		1.4	2	mA	System average current includes VCSEL current.
Sniff mode 300ms Current	I_{sniff_300ms}		0.335	0.785	mA	System average current includes VCSEL current.
Deep Sleep Current	I_{DSleep}		110	280	µA	Disconnected, wake on sensor motion. State preserved.
			80	250	µA	Disconnected, wake on button clicked. State preserved.
Input Hysteresis	V_{HYST}		285		mV	Pins: B1-B8, TW1, TW2
Button Pull-up Current	I_{PULLUP}	100	300	500	µA	Pins: B1-B8, TW1, TW2
Input Low Voltage	V_{IL}			0.2* V_{DD3}	V	Pins: B1-B8, TW1, TW2, ZA, ZB
Input High Voltage	V_{IH}	0.8* V_{DD3}			V	Pins: B1-B8, TW1, TW2, ZA, ZB
Input Leakage Current	I_{leak}		±1	±10	µA	Vin = 0.7* V_{DD3}
Output Low Voltage, LASER_NEN	V_{OL}			0.2* V_{DD3}	V	Iout= 1mA, LASER_NEN
Output High Voltage, LASER_NEN	V_{OH}	0.8* V_{DD3}			V	Iout= -0.5mA, LASER_NEN
Input Capacitance	C_{in}			10	pF	



Notes:

1. Point A = Ramp start point of REG0/V_{DD_RF} that triggers internal reset process.
2. Point B = Stable point of REG0/V_{DD_RF} that crystal will start its oscillation.
3. $T_{ramp\_VDD\_RF}$ = Ramp up time og REG0/V_{DD_RF}.
4. T_{OS} = Crystal startup time. Depends on crystal's drive level and load capacitance.
5. T_{Reset} = ADNS-7630's internal Power On Reset (POR) process duration.
6. $T_{XTAL\_RDY} < T_{Reset}$.

Figure 14. Power-Up Timing Diagram

Transmitter RF Specifications

Electrical Characteristics over recommended operating conditions based on Avago Technologies' ADNK-7633 reference design mouse. Typical values at 25 °C, V_{DD21} = 2.8V, V_{DD3} = 2.8V

Parameter	Minimum	Typical	Maximum	Units	Notes
Transmitter Section					
Spectrum frequency range	2400		2483.5	MHz	
Output power	-6	0	4	dBm	
In-Band Spurious Emission					
+/-500 kHz			-20	dBC	
Out-of-Band Spurious Emission					
30 MHz to 1 GHz		-60	-36	dBm	
1 GHz to 12.75 GHz			-30	dBm	
1.8 GHz to 1.9 GHz		-80	-47	dBm	
5.15 GHz to 5.3 GHz		-90	-47	dBm	
Lo Performance					
Lock time		130	180	µs	
Initial carrier frequency tolerance		±25	±75	kHz	
Frequency Drift					
DM1 packet		±20	±25	kHz	
DH1 packet		±20	±25	kHz	
Drift rate		10	20	kHz/50 µs	
Frequency Deviation					
Average deviation in payload (sequence used is 00001111)	140	168	175	kHz	
Maximum deviation in payload (sequence used is 10101010)	115			kHz	
Channel spacing		1		MHz	

KeyMap (KM)

The KeyMap is only supported in Bluetooth version 2.0 firmware. KM enables any available GPIO between GPIO11-GPIO15 to be assigned as keyboard shortcut key. User_Defined_Function_n_A/B/C registers (where, n=1, 2, 3, 4 or 5) allow configuration of User_Defined_Function_n_A/B/C registers (where, n=1, 2, 3, 4 or 5). Thus, the sensor can be customized to implement standard Microsoft keyboard shortcut keys or special shortcut keys used in different applications, e.g. Office, CAD, PC Games, etc.

The respective first and second byte of keyboard code A, B and C can be assigned to programmable button n (where, n=1, 2, 3, 4 or 5) in the MConfig software program. The first byte usually consists of any combinations for keys located on the either side (left or right only) of a standard keyboard as listed:

- Windows Logo Key ("**LWIN**", "**RWIN**")
- CTRL ("**LCTRL**", "**RCTRL**")
- SHIFT ("**LSHIFT**", "**RSHIFT**")
- ALT ("**LALT**", "**RALT**")

The second byte can be referred to any single keyboard key scan code available from Windows Platform Design Notes on Keyboard Scan Code Specification, which can be downloaded from:

http://www.microsoft.com/whdc/archive/scancode.mspx

Some examples of possible key combinations for programmable buttons below:

If keyboard code A of programmable button 1 is shortcut key of "**Windows Logo Key**",

Keyboard code A byte1 = "**LWin**" (or "**RWin**")
Keyboard code A byte2 = Not Support
User_Defined_Function_1_A = a1 01 08 00 03 00 00 00 00 00

If keyboard code A of programmable button 2 is shortcut key of "**Enter**",

Keyboard code A byte1 = "Not Support"
Keyboard code A byte2 = "**ENTER**"
User_Defined_Function_2_A = a1 01 00 00 28 00 00 00 00 00

If keyboard code B for programmable button 5 is shortcut key of "**Ctrl+Alt+Delete**",

Keyboard code B byte1 = "**LAlt+LCtrl**" (or "**RAlt+RCtrl**")
Keyboard code B byte2 = "**Delete**"
User_Defined_Function_5_B = a1 01 05 00 4c 00 00 00 00 00

Note: "LCtrl+RAlt" and "RCtrl+LAlt" are not supported.

EEPROM Write Protect Feature

Notice that B8/WP can either be used as a programmable button or LED indicator, or even as an I/O pin for EEPROM Write Protect function. In the event where all I/Os above are used up in a Bluetooth Mouse with tilt wheel, schematic below can be used to generate a 'pseudo I/O' for EEPROM Write Protect function. However, if all I/Os are used up in a Bluetooth Mouse without tilt wheel, there will be no EEPROM Write Protect function in the mouse. Though the possibility of EEPROM being overwritten through normal mouse operation is low, Avago Technologies highly recommends mouse makers to use either B8/WP or the "pseudo I/O" method for EEPROM Write Protect function.

Media Buttons

The Media button featuring audio control is supported in both Bluetooth version 2.0 and 2.1 firmwares. The ADNS-7630 is the first one-chip mouse sensor to support Consumer Control usages as defined in the **Consumer Page** (page 0x0C) in the *Universal Serial Bus HID Usage Tables Version 1.0 specification*. For more information, please visit http://www.usb.org/developers/hidpage/.

This feature is related to User-Defined HID Programmable Buttons listed in EEPROM registers. For example, in order to define one function of consumer page, the value should be set in the format of "a1 07 xx yy 00 00 00 00 00 00", where xx yy should be replaced by the usage ID of the target function in byte-inverted sequence, eg. "cd 00" for ID = cd and "25 02" for ID = 225. When manually setting this media button function in MConfig software program, both first and second bytes of corresponding Keyboard Code A, B or C must be set to "Not Support". The User Defined Function C for each programmable button will cease to be effective when SSP is enabled in Bluetooth-Version-2.1's firmware.

Table 15. Example of Consumer Page audio controls supported in Windows 2000.

Usage	Name	Type
0xE0	Volume*	Linear Control (LC)
0xE2	Mute*	On/Off Control (OOC)
0xE3	Bass	Linear Control (LC)
0xE4	Treble	Linear Control (LC)
0xE5	Bass Boost*	On/Off Control (OOC)
0xE7	Loudness	On/Off Control (OOC)
0xE9	Volume Increment*	Re-trigger Control (RTC)
0xEA	Volume Decrement*	Re-trigger Control (RTC)

* These controls are supported in Windows 98 (original release and Service Pack 1 release).

Note: Programmable buttons with RTC usage type controls should be assigned to single click function only. If the button is pressed continuously and not released, the event will be retriggered. Thus, there should not be any long press function assigned to these buttons. For example, if user keeps pressing the Volume Increment button, ADNS-7630 will perform the actual re-triggering of events that will lead to continuous increments of the volume until the button has been released or until the maximum volume has been reached.

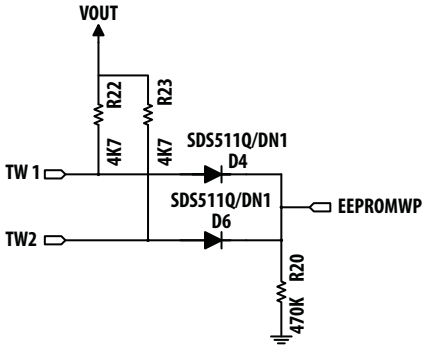


Figure 17. "Pseudo I/O" for EEPROM Write Protect Function

Registers (continued)

Domain	Register Name	Register Address Bluetooth Ver2.0	Register Address Bluetooth Ver2.1	Byte Size	Default Value Bluetooth Ver2.0	Default Value Bluetooth Ver2.1
Reconnect LED Configuration	Reconnect_LED_Enabled	0x0230		1	False	
	Reconnect_LED_PIN	0x0231		1	0	
	Reconnect_LED_GPIO_State	0x0232		1	0	
	Reconnect_LED_On_Duration	0x0233		1	3	
Battery LED Configuration	Battery_LED_Enabled	0x011a		1	True	
	Battery_LED_PIN	0x011b		1	5	
	Battery_LED_GPIO_State	0x011c		1	0	
	Battery_LED_On_Period	0x011d		1	4	
	Battery_LED_Off_Period	0x011e		1	9	
	Battery_LED_Blink_Duration	0x0238-0x0239		2	30	
	Battery_LED_Rest_Duration	0x023a-0x023b		2	0	
	Battery_LED_Active_Sniff_Mode	0x023c		1	0x06	
	Battery_LED_Total_Duration	0x023d		1	30	
	Battery_LED_Disabled _Before_Connection	0x0248		1	True	
CPI Selection Indicator	Resolution_LED_GPIO_Total	0x011f		1	0	
	Resolution_LED_GPIO_Selection1	0x0120		1	0	
	Resolution_LED_GPIO_Selection2	0x0121		1	0	
	Resolution_LED_GPIO_Selection3	0x0122		1	0	
	Resolution_LED_GPIO_Selection4	0x0123		1	0000	
	Resolution_LED_Setting1	0x0124		1	0000	
	Resolution_LED_Setting2	0x0125		1	0000	
	Resolution_LED_Setting3	0x0126		1	0000	
	Resolution_LED_Setting4	0x0127		1	0000	
	Resolution_LED_Setting5	0x0128		1	0000	
	Resolution_LED_Setting6	0x0129		1	0000	
	Resolution_LED_Setting7	0x012a		1	0000	
	Resolution_LED_Setting8	0x012b		1	0000	
	Resolution_LED_Setting9	0x012c		1	0000	
	Resolution_LED_Setting10	0x012d		1	0000	
	Resolution_LED_Duration	0x012e		1	0	
	GPIO_state_On_Resolution_LED	0x012f		1	0	
	Resolution_LED_On_Period	0x0130		1	0	
	Resolution_LED_Off_Period	0x0131		1	0	
	Resolution_LED_Blink_On_ Connection	0x024a		1	False	
Motion Configuration						
Motion Configuration	Report_Protocol	0x0132		1	Report Mode	
	Motion_Report_Size	0x0133		1	12	
	XY_Swap	0x0134		1	True	
	X_Flip	0x0135		1	False	
	Y_Flip	0x0136		1	False	
	Z_Selection	0x0137		1	Mechanical	
	Z_Configuration	0x0138		1	Z/2	
	Z_Negate	0x0139		1	False	
	X_Scale	0x013a		1	0	
	Y_Scale	0x013b		1	0	

Connection Configuration

Mouse_Power_Up_Mode

Size: 1 byte Default Value: **Discoverable mode**

USAGE: This register defines which mode the mouse will enter after power-up, if reconnection is unnecessary.
Set to "Sleep mode" to make mouse enter sleep mode;
Set to "Discoverable mode" to enter discoverable mode.

Auto_Reconnect_Enabled

Size: 1 byte Default Value: **True**

USAGE: This register defines the status of auto reconnection to host after power on

LMP_Supervision_TimeOut

Size: 2 byte Default Value: **8000**

USAGE: This register defines the LMP supervision timeout in slots of 625us each. For example, 4096 means 4096*625us = 2.56 seconds.

Page_Scan_Window

Size: 2 byte Default Value: **768**

USAGE: This register defines the page scan window in slots of 625us each. For example, 768 means 768*625us = 480ms.

Page_Scan_Interval

Size: 2 byte Default Value: **1024**

USAGE: This register defines the page scan interval in slots of 625us each. For example, 1024 means 1024*625us = 640ms.

Inquiry_Scan_Window

Size: 2 byte Default Value: **768**

USAGE: This register defines the inquiry scan window in slots of 625us each. For example, 768 means 768*625us = 480ms.

Inquiry_Scan_Interval

Size: 2 byte Default Value: **1024**

USAGE: This register defines the inquiry scan interval in slots of 625us each. For example, 1024 means 1024*625us = 640ms.

Inquiry_Scan_TimeOut

Size: 2 byte Default Value: **2250**

USAGE: This register defines the inquiry scan timeout (multiples of 80ms). For example, 750 means 750*80ms = 60 seconds.

Page_TimeOut

Size: 2 byte Default Value: **0**

USAGE: This register defines the page timeout (multiples of 80ms). For example, 30 means 30*80ms = 2.4 seconds. Set to 0 to disable page timeout.

Pairing_Mode

Size: 1 byte Default Value: **True**

USAGE: This register defines whether auto pairing or normal pairing mode is used. Data type is Boolean.
Set to "True" for auto pairing which support authentication
Set to "False" for normal paring which reject authentication

Authentication_Mode

Size: 1 byte Default Value: **False**

USAGE: This register defines whether the host or the device starts authentication. Data type is Boolean.
Set to "True" to allow mouse to initiate authentication;
Set to "False" to allow host to initiate authentication.

Connect_Button_Press_Duration

Size: 2 byte Default Value: **12**

USAGE: This register defines duration (multiples of 80ms) needed for the connect button to be held before events are generated. 10 means 10*80ms = 0.8 second.

VC_Unplug_Enable

Size: 1 byte Default Value: **True**

USAGE: This register controls whether a Virtual Cable unplug is generated on a connect button press. Data type is Boolean.
Set to "True" to enable Virtual Cable unplug when connection button is pressed;
Set to "False" to disable Virtual Cable unplug when connection button is pressed.

Low Power Configuration

Max_Sniff_Modes

Size: 1 byte Default Value: **3**

USAGE: This register defines the maximal number (less than or equal to 8) of sniff modes, for example, 3 means there are 3 sniff modes at most.

Sleep_Mode_Enabled

Size: 1 byte Default Value: **True**

USAGE: This register defines whether to enter sleep mode when last sniff mode timeout. Data type is Boolean.
Set to "True" to allow mouse enter sleep mode when last sniff mode timeout;
Set to "False" to disallow mouse enter sleep mode when last sniff mode timeout.

Wake_Up_Method

Size: 1 byte Default Value: **All event**

USAGE: This register defines through which way the mouse will be awakened.
Set to "All event" to allow a button event or motion to wake up the mouse;
Set to "Button event" to allow a button event to wake up the mouse;
Set to "Motion" to allow motion to wake up the mouse.

Sniff_Retry_Count

Size: 1 byte Default Value: **3**

USAGE: This register defines how many times of sniff will be requested by device if the master rejects the sniff request. For example, 3 means sniff will retry 3 times. Set to 0 to retry forever.

Sniff_Retry_Interval

Size: 2 byte Default Value: **1600**

USAGE: This register defines how many slots (1slot = 625us) the device should wait before resending the sniff request. For example, 1600 means 1600*625us = 1 second.

Sniff_Timeout

Size: 2 byte Default Value: **1**

USAGE: This register defines timeout (in slots) of the sniff request to master in the current state. For example, 10 means 10 transmission slots. For the HID device to work well while multiple Bluetooth devices are connected to the host, non-zeros value is recommended.

Notes: If the slave has received a packet with a matching LT_ADDR that contains ACL data (DM, DH, DV, or AUX1 packets) in the preceding Nsniff_timeout master-to-slave transmission slots, then it shall continue listening.

If the slave has transmitted a packet containing ACL data (DM, DH, DV, or AUX1 packets) in the preceding Nsniff_timeout slave-to-master transmission slots, then it shall continue listening.

If the slave has received any packet with a matching LT_ADDR in the preceding Nsniff_timeout master-to-slave transmission slots, then it may continue listening.

Sniff_Attempt

Size: 2 byte Default Value: **2**

USAGE: This register defines sniff attempt (in slots) of the sniff request to master in the current state. For example, 4 means 4 transmission slot.

Notes: If fewer than Nsniff_attempt master-to-slave transmission slots have elapsed since the sniff anchor point, then the slave shall continue listening.

Sniff_Mode_Interval0

Size: 2 byte Default Value: **18**

USAGE: This register defines sniff interval (in slot) for sniff mode 0. For example, 18 means 18*625us = 11.25ms.

Note: Only 10, 12, 14, 16, 18 and 20 are valid.

Sniff_Mode_Interval1

Size: 2 byte Default Value: **108**

USAGE: This register defines sniff interval (in slot) for sniff mode 1. For example, 108 means 108*625us = 67.5ms.

Sniff_Mode_Interval2

Size: 2 byte Default Value: **468**

USAGE: This register defines sniff interval (in slot) for sniff mode 2. For example, 480 means 480*625us = 300ms.

Sniff_Mode_Interval3

Size: 2 byte Default Value: **0**

USAGE: This register defines sniff interval (in slot) for sniff mode 3. For example, 300 means 300*625us = 187.5ms.

Sniff_Mode_Interval4

Size: 2 byte Default Value: **0**

USAGE: This register defines sniff interval (in slot) for sniff mode 4. For example, 400 means 400*625us = 250ms.

Sniff_Mode_Interval5

Size: 2 byte Default Value: **0**

USAGE: This register defines sniff interval (in slot) for sniff mode 5. For example, 500 means 500*625us = 312.5ms.

Sniff_Mode_Interval6

Size: 2 byte Default Value: **0**

USAGE: This register defines sniff interval (in slot) for sniff mode 6. For example, 600 means 600*625us = 375ms.

Sniff_Mode_Interval7

Size: 2 byte Default Value: **0**

USAGE: This register defines sniff interval (in slot) for sniff mode 7. For example, 700 means 700*625us = 437.5ms.

Sniff_Mode_Duration0

Size: 2 byte Default Value: **178**

USAGE: The register defines sniff mode duration (must be a positive multiple of corresponding sniff mode interval) for the sniff mode. For examples, 178 means 178 times of the corresponding sniff mode interval (Sniff_Mode_Interval0 in this case). Therefore the default Sniff_Mode_Duration0 means 178*11.25ms = 2002.5ms.

Sniff_Mode_Duration1

Size: 2 byte Default Value: **900**

USAGE: The register defines sniff mode duration (must be a positive multiple of corresponding sniff mode interval) for the sniff mode. For examples, 370 means 370 times of the corresponding sniff mode interval (Sniff_Mode_Interval1 in this case). Therefore the default Sniff_Mode_Duration1 means 370*67.5ms = 24.975sec.

Sniff_Mode_Duration2

Size: 2 byte Default Value: **2050**

USAGE: The register defines sniff mode duration (must be a positive multiple of corresponding sniff mode interval) for the sniff mode. For examples, 6000 means 6000 times of the corresponding sniff mode interval (Sniff_Mode_Interval2 in this case). Therefore the default Sniff_Mode_Duration2 means 6000*300ms = 1800sec (30min).

Sniff_Mode_Duration3

Size: 2 byte Default Value: **0**

USAGE: The register defines sniff mode duration (must be a positive multiple of corresponding sniff mode interval) for the sniff mode. For examples, 50 means 50 times of the corresponding sniff mode interval.

Sniff_Mode_Duration4

Size: 2 byte Default Value: **0**

USAGE: The register defines sniff mode duration (must be a positive multiple of corresponding sniff mode interval) for the sniff mode. For examples, 50 means 50 times of the corresponding sniff mode interval.

Sniff_Mode_Duration5

Size: 2 byte Default Value: **0**

USAGE: The register defines sniff mode duration (must be a positive multiple of corresponding sniff mode interval) for the sniff mode. For examples, 50 means 50 times of the corresponding sniff mode interval.

Sniff_Mode_Duration6

Size: 2 byte Default Value: **0**

USAGE: The register defines sniff mode duration (must be a positive multiple of corresponding sniff mode interval) for the sniff mode. For examples, 50 means 50 times of the corresponding sniff mode interval.

Sniff_Mode_Duration7

Size: 2 byte Default Value: **0**

USAGE: The register defines sniff mode duration (must be a positive multiple of corresponding sniff mode interval) for the sniff mode. For examples, 50 means 50 times of the corresponding sniff mode interval.

Mouse Generic Configuration

Programmable_Buttons_Total

Size: 1 byte Default Value: **0**

USAGE: This register defines the number of programmable buttons. For example, 5 means there are 5 programmable buttons.

Programmable_Buttons_Low_Power

Size: 1 byte Default Value: **1**

USAGE: This register defines the maximal number of button events to be saved during low power mode period, for example, 1 means 1 button event will be saved at most. For current IC version, this register value is fixed to "1".

Debouncing_Time

Size: 1 byte Default Value: **4**

USAGE: Define the shortest period of time (in ms) for effective button state of an operation. An integer between 1 and 30 is valid. For example, 30 means a button press/release state will be ignored if this state lasts less than 30 ms.

GPIO_Pin_Selection1, GPIO_Pin_Selection2, GPIO_Pin_Selection3, GPIO_Pin_Selection4, GPIO_Pin_Selection5

Size: 1 byte Default Value: **0**

USAGE: This register selects which pin the programmable button is connected to. An integer between 11 and 15 is valid. For example, 11 means the programmable button is connected to GPIO11.

Single_Click_Function1, Single_Click_Function2, Single_Click_Function3, Single_Click_Function4, Single_Click_Function5

Size: 1 byte Default Value: **Not Supported**

USAGE: This register defines an explicit function of each single-click function of programmable button.
Set to "Not support" to disable single click function;
Set to "Function A" to choose Function A for single click function;
Set to "Function B" to choose Function B for single click function;
Set to "Function C" to choose Function C for single click function;
Set to "Increase CPI" to choose Increase CPI for single click function;
Set to "Decrease CPI" to choose Decrease CPI for single click function;
Set to "CPI Rotation" to choose CPI Rotation for single click function.

User_Defined_Function_1_B, User_Defined_Function_2_B, User_Defined_Function_3_B, User_Defined_Function_4_B, User_Defined_Function_5_B

Size: 10 byte Default Value: **a1 00 00 00 03 00 00 00 00 00**

USAGE: Define the user-defined HID report for function B of programmable button 1 to 5. For example, in order to define one function of consumer page, the value should be set in the format of "a1 07 xx yy 00 00 00 00 00 00", where xx yy should be replaced by the usage ID of the target function in byte-inverted sequence, e.g. "cd 00" for ID = cd and "25 02" for ID = 225. When manually setting this item, keyboard code B must be set to "Not support" in both bytes.

User_Defined_Function_1_C, User_Defined_Function_2_C, User_Defined_Function_3_C, User_Defined_Function_4_C, User_Defined_Function_5_C

Size: 10 byte Default Value: **a1 00 00 00 03 00 00 00 00 00**

USAGE: Define the user-defined HID report for function C of programmable button 1 to 5. For example, in order to define one function of consumer page, the value should be set in the format of "a1 07 xx yy 00 00 00 00 00 00", where xx yy should be replaced by the usage ID of the target function in byte-inverted sequence, e.g. "cd 00" for ID = cd and "25 02" for ID = 225. When manually setting this item, keyboard code B must be set to "Not support" in both bytes. If Secure Simple Pairing is enabled, this feature will cease to be effective.

Tilt_Wheel_Enabled

Size: 1 byte Default Value: **Not Supported**

USAGE: This register enables or disables the tilt wheel function (via TW+ and TW- pins). Data type is Boolean.
Set to "Not Supported" to disable tilt wheel function;
Set to "Support TWheel Function" to activate the TW+ and TW- for tilt wheel function;
Set to "Support LED Function" to activate the TW+ and TW- as LED GPIO.

Power_On_LED_Enabled

Size: 1 byte Default Value: **False**

USAGE: This register enables or disables power-on LED indicator function. Data type is Boolean.
Set to "True" to enable power-on LED support;
Set to "False" to disable power-on LED support.

Power_On_LED_PIN

Size: 1 byte Default Value: **0**

USAGE: This register defines which GPIO pin the power-on LED is connected to. GPIO3-GPIO6 and GPIO11-GPIO15 are valid options. For example, 6 means the power-on LED is connected to GPIO6.

Power_On_LED_GPIO_State

Size: 1 byte Default Value: **0**

USAGE: This register defines the GPIO value which causes the power-on LED to turn on. The opposite value is used automatically to turn it off. Only 0 and 1 are valid. For example, 1 means GPIO value "1" causes power-on LED to turn on, and GPIO value "0" causes power-on LED to turn off.

Power_On_LED_On_Duration

Size: 1 byte Default Value: **37**

USAGE: This register defines power-on LED on period (multiples of 80ms). The range is 0 to 255. For example, 10 means 10*80ms = 0.8 second.

Discover_LED_Off_Period

Size: 1 byte Default Value: **10**

USAGE: This register defines discover LED off period (multiples of 80ms). The range is 0 to 255. For example, 10 means 10*80ms = 0.8 second.

Reconnect_LED_Enabled

Size: 1 byte Default Value: **False**

USAGE: This register enables or disables reconnect LED indicator function. Data type is Boolean.
Set to "True" to enable reconnect LED support;
Set to "False" to disable reconnect LED support.

Reconnect_LED_PIN

Size: 1 byte Default Value: **0**

USAGE: This register defines which GPIO pin the reconnect LED is connected to. GPIO3-GPIO6 and GPIO11-GPIO15 are valid options. For example, 6 means the reconnect LED is connected to GPIO6.

Reconnect_LED_GPIO_State

Size: 1 byte Default Value: **0**

USAGE: This register defines the GPIO value which causes the reconnect LED to turn on. The opposite value is used automatically to turn it off. Only 0 and 1 are valid. For example, 1 means GPIO value "1" causes reconnect LED to turn on, and GPIO value "0" causes reconnect LED to turn off.

Reconnect_LED_On_Duration

Size: 1 byte Default Value: **37**

USAGE: This register defines reconnect LED on period (multiples of 80ms). The range is 0 to 255. For example, 10 means 10*80ms = 0.8 second.

Battery_LED_Enabled

Size: 1 byte Default Value: **True**

USAGE: This register enable or disable battery LED indicator function. Data type is Boolean.
Set to "True" to enable battery LED support;
Set to "False" to disable battery LED support.

Battery_LED_PIN

Size: 1 byte Default Value: **5**

USAGE: This register defines which GPIO pin the battery LED is connected to. Only GPIO5 and GPIO6 are valid options. For example, 5 means the discover LED is connected to GPIO5.

Battery_LED_GPIO_State

Size: 1 byte Default Value: **0**

USAGE: This register defines the GPIO value which causes the battery LED to turn on. The opposite value is used automatically to turn it off. Only 0 and 1 are valid. For example, 1 means GPIO value "1" causes battery LED to turn on, and GPIO value "0" causes battery LED to turn off.

Resolution_LED_Setting1

Size: 1 byte Default Value: **0000**

USAGE: This register defines which resolution LED indicators light up when mouse is set to Resolution_Selection_1, in 4-bit big-endian binary. Set to "0000" to disable LED indication for Resolution_Selection_1. For example, "0101" means that the two GPIO pins which are asserted for LED_GPIO_Selection1 and LED_GPIO_Selection3 will output high, while the two GPIO pins which are asserted for LED_GPIO_Selection2 and LED_GPIO_Selection4 will output low.

Resolution_LED_Setting2

Size: 1 byte Default Value: **0000**

USAGE: This register defines which resolution LED indicators light up when mouse is set to Resolution_Selection_2, in 4-bit big-endian binary. Set to "0000" to disable LED indication for Resolution_Selection_2.

Resolution_LED_Setting3

Size: 1 byte Default Value: **0000**

USAGE: This register defines which resolution LED indicators light up when mouse is set to Resolution_Selection_3, in 4-bit big-endian binary. Set to "0000" to disable LED indication for Resolution_Selection_3.

Resolution_LED_Setting4

Size: 1 byte Default Value: **0000**

USAGE: This register defines which resolution LED indicators light up when mouse is set to Resolution_Selection_4, in 4-bit big-endian binary. Set to "0000" to disable LED indication for Resolution_Selection_4.

Resolution_LED_Setting5

Size: 1 byte Default Value: **0000**

USAGE: This register defines which resolution LED indicators light up when mouse is set to Resolution_Selection_5, in 4-bit big-endian binary. Set to "0000" to disable LED indication for Resolution_Selection_5.

Resolution_LED_Setting6

Size: 1 byte Default Value: **0000**

USAGE: This register defines which resolution LED indicators light up when mouse is set to Resolution_Selection_6, in 4-bit big-endian binary. Set to "0000" to disable LED indication for Resolution_Selection_6.

Resolution_LED_Setting7

Size: 1 byte Default Value: **0000**

USAGE: This register defines which resolution LED indicators light up when mouse is set to Resolution_Selection_7, in 4-bit big-endian binary. Set to "0000" to disable LED indication for Resolution_Selection_7.

Resolution_LED_Setting8

Size: 1 byte Default Value: **0000**

USAGE: This register defines which resolution LED indicators light up when mouse is set to Resolution_Selection_8, in 4-bit big-endian binary. Set to "0000" to disable LED indication for Resolution_Selection_8.

Resolution_LED_Setting9

Size: 1 byte Default Value: **0000**

USAGE: This register defines which resolution LED indicators light up when mouse is set to Resolution_Selection_9, in 4-bit big-endian binary. Set to "0000" to disable LED indication for Resolution_Selection_9.

Resolution_LED_Setting10

Size: 1 byte Default Value: **0000**

USAGE: This register defines which resolution LED indicators light up when mouse is set to Resolution_Selection_10, in 4-bit big-endian binary. Set to "0000" to disable LED indication for Resolution_Selection_10.

Resolution_LED_Duration

Size: 1 byte Default Value: **0**

USAGE: This register defines the duration in which the resolution LED indicators work, in 80ms. The range is from 0 to 255. Set to 0 to disable the LED indicator. For example, 30 means the LEDs will be on for 2.4 seconds.

GPIO_state_ON_Resolution_LED

Size: 1 byte Default Value: **0**

USAGE: This register defines the GPIO value which causes the LED to turn on. The opposite value is used automatically to turn it off. Only 0 and 1 are valid. For example, 1 means GPIO value "1" causes LED to turn on, and GPIO value "0" causes LED to turn off.

Resolution_LED_ON_Period

Size: 1 byte Default Value: **0**

USAGE: This register defines LED on period (in 80ms). The range is 0 to 255. The LED on period must not be greater than the LED duration. For example, 10 means 0.8 second.

Resolution_LED_OFF_Period

Size: 1 byte Default Value: **0**

USAGE: This register defines LED off period (in 80ms). The range is 0 to 255. The LED on period must not be greater than the LED duration. For example, 10 means 0.8 second.

Resolution_LED_Blink_On_Connection

Size: 1 byte Default Value: **False**

USAGE: This register defines whether to blink resolution LED after connection is established.

Set to "True" to allow corresponding resolution LED to blink;

Set to "False" to disallow corresponding resolution LED to blink.

Sensor Configuration

Max_Resolution

Size: 1 byte Default Value: **3000**

USAGE: This register sets the maximum sensor resolution in count per inch (cpi). For example, 3000 means the maximum sensor resolution is 3000cpi. This register value is fixed and not programmable.

Default_Resolution

Size: 1 byte Default Value: **1250**

USAGE: This register sets the default sensor resolution in count per inch (cpi). The value must be an integral multiple of 250, and be less than or equal to "Max resolution". For example, 750 means the default sensor resolution is 750cpi.

Resolution_Selection_Method

Size: 1 byte Default Value: **Not Support**

USAGE: This register defines the way to input resolution selection.

Set to "Not Support" to disable this feature;

Set to "Hotkey" to select CPI by hotkey.

Resolution_Selection_Total

Size: 1 byte Default Value: **0**

USAGE: This register sets the number of resolution setting stages to be saved, for example, 10 means 10 resolution setting stages will be saved.

Current_Resolution_Selection

Size: 1 byte Default Value: **0**

USAGE: This register sets current mouse resolution from the list of Resolution_Setting index. Set to 0 to implicitly choose the value of "Default resolution". The mouse will save the latest used resolution selection of the user, if applicable, as this index. For example, 1 means the current selection is Resolution_Setting1.

Resolution_Setting1, Resolution_Setting2, Resolution_Setting3, Resolution_Setting4, Resolution_Setting5
Resolution_Setting6, Resolution_Setting7, Resolution_Setting8, Resolution_Setting9, Resolution_Setting10

Size: 1 byte Default Value: **0**

USAGE: This register sets the resolution of each selection (from1 to 10). The value must be an integral multiple of 250, and be less than or equal to "Max resolution" in Resolution page. Set to 0 to disable this selection. For example, 750 means CPI selection 1 is "750".

SDP Programmable Features

SDP_Service_Name

Size: 64 byte Default Value: **Avago Bluetooth Mouse**

USAGE: This register defines the service name in SDP.

SDP_Service_Name_Length

Size: 1 byte Default Value: **21**

USAGE: This register defines the length of the service name in SDP.

SDP_Service_Description

Size: 16 byte Default Value: **A Mouse**

USAGE: This register defines the service description in SDP.

SDP_Service_Description_Length

Size: 1 byte Default Value: **7**

USAGE: This register defines the length of service description in SDP.

SDP_Provider_Name

Size: 16 byte Default Value: **Avago**

USAGE: This register defines the provider name in SDP.

SDP_Provider_Name_Length

Size: 1 byte Default Value: **5**

USAGE: This register defines the length of provider name in SDP.

SDP_Vendor_ID

Size: 2 byte Default Value: **0**

USAGE: This register defines the vendor ID in SDP, specific for manufacturer.

SDP_Product_ID

Size: 2 byte Default Value: **0**

USAGE: This register defines the product ID in SDP, specific for manufacturer.

SDP_Product_Version

Size: 2 byte Default Value: **0**

USAGE: This register defines the product version in SDP in hexadecimal, with higher byte representing the major version number, and lower byte representing the minor version number. For example 0x0100 means version 1.0, 0x1011 means version 16.17.