Microchip Technology - PIC12F509-I/MS Datasheet

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	Surface Mount
Package / Case	8-TSSOP, 8-MSOP (0.118", 3.00mm Width)
Supplier Device Package	8-MSOP
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic12f509-i-ms

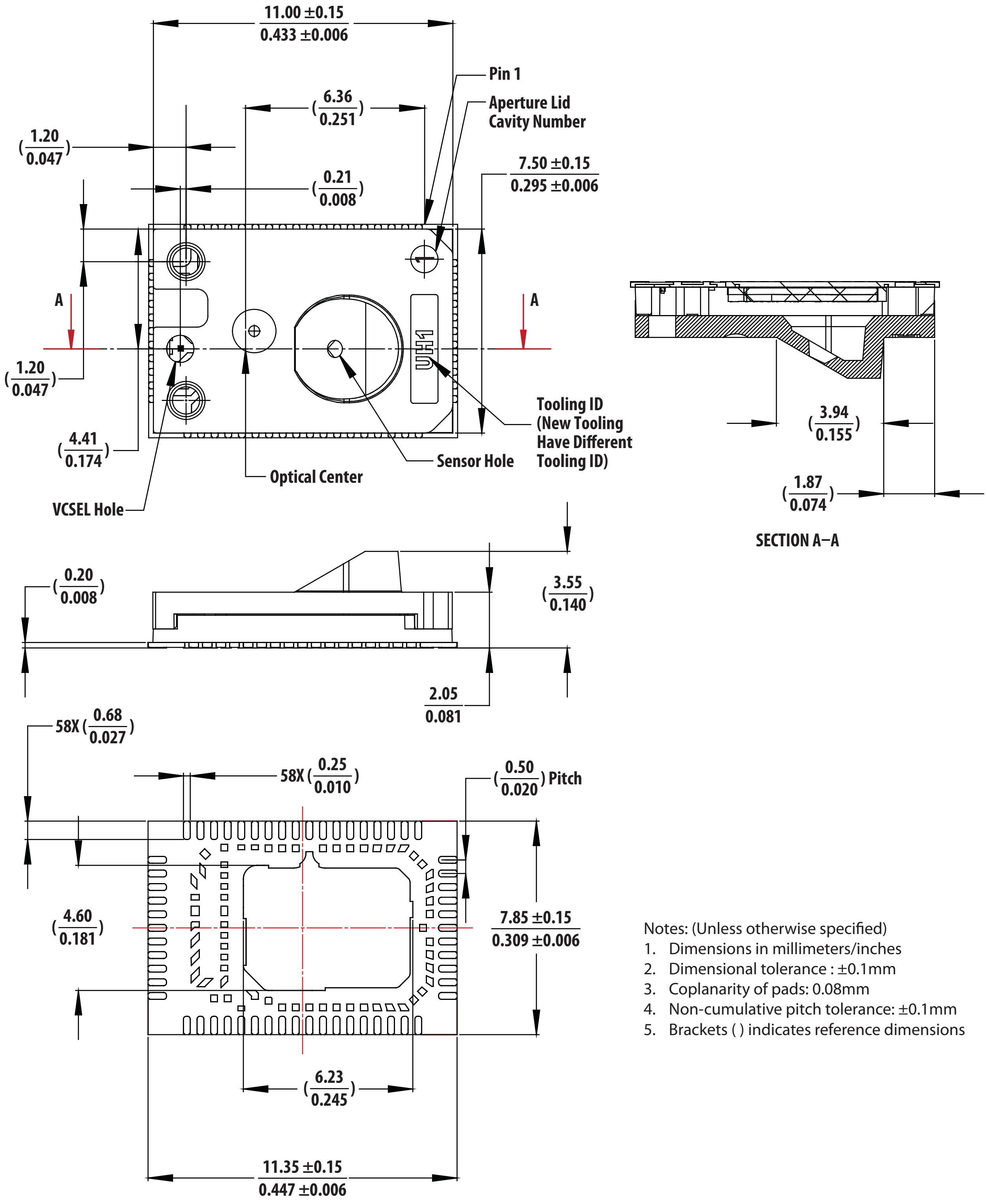
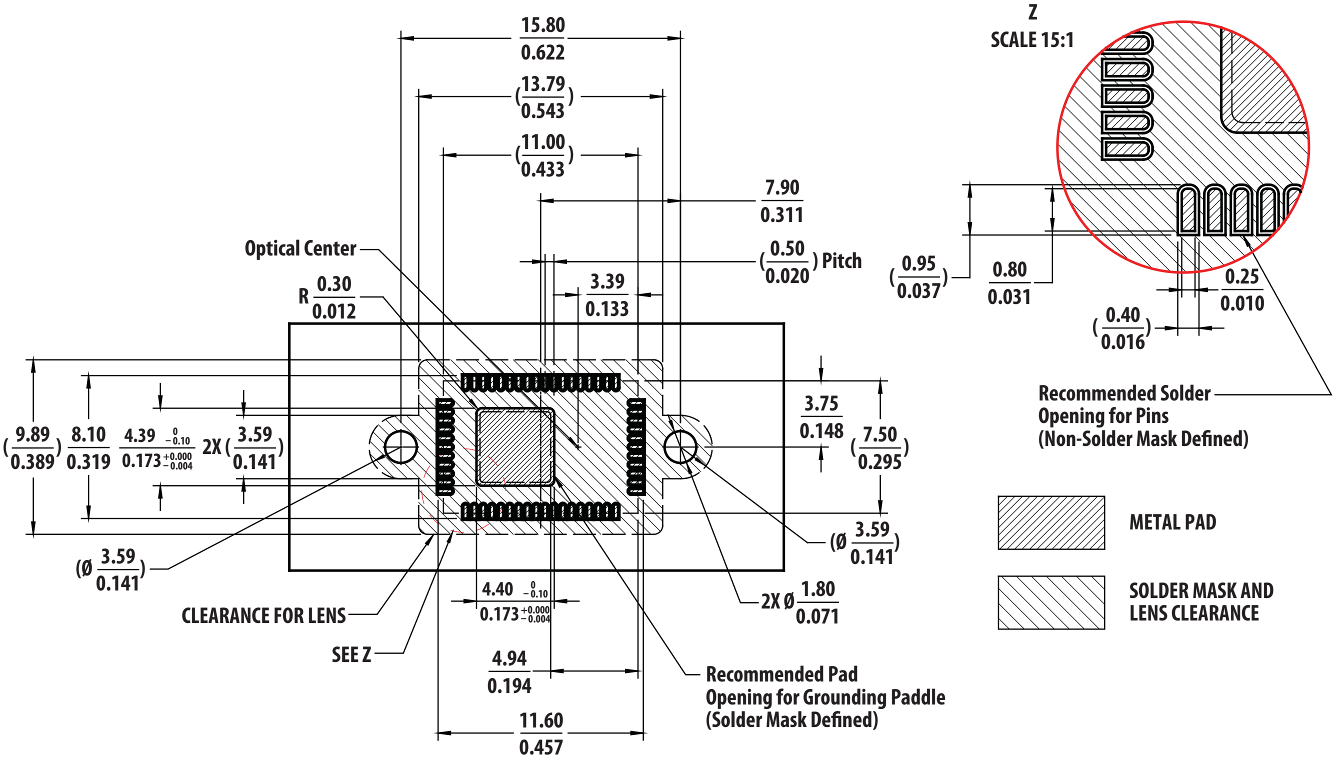


Figure 3. Package outline drawing

Notes: (Unless otherwise specified)

1. Dimensions in millimeters/inches
2. Dimensional tolerance : ±0.1mm
3. Coplanarity of pads: 0.08mm
4. Non-cumulative pitch tolerance: ±0.1mm
5. Brackets ( ) indicates reference dimensions

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



Note:

1. Dimensions in millimeters/inches

Figure 4. Recommended PCB mechanical cutouts and spacing (Top view)

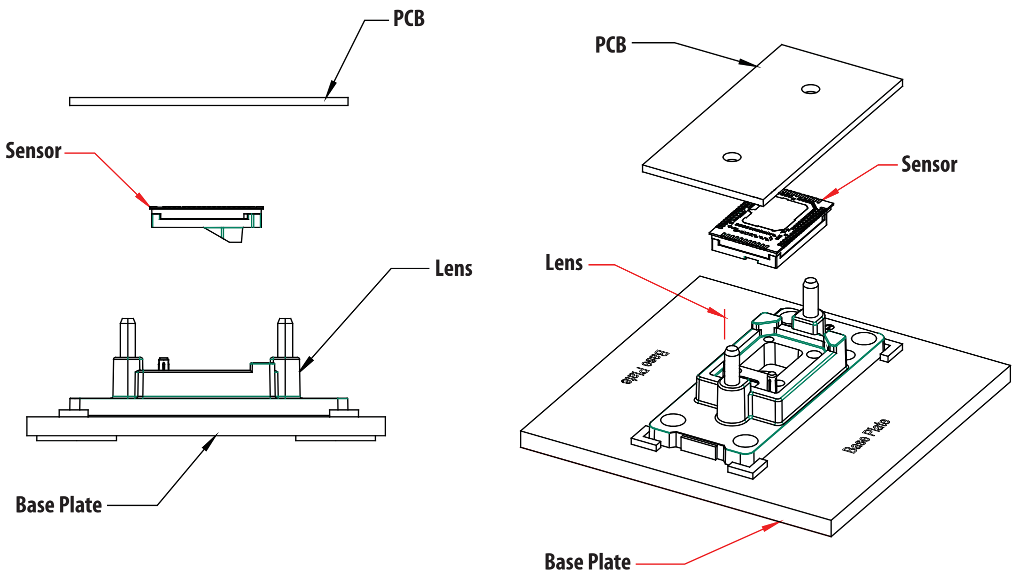


Figure 6. Exploded view drawing of ADNS-7630 sensor coupled with ADNS-7100-001 lens, PCB & base plate (front view and top side view)

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.

PCB Layout Requirements:

1. Recommended to use 4-layer PCB board, with second layer as GND plane and third layer as power plane.
2. Cut the copper beneath the antenna pattern on the GND plane, power layer and the bottom layer; no signal line is allowed beneath the antenna pattern at all of the layers. Antenna pattern is highly recommended to be located at one of the board edges, furthest away from palm coverage.
3. Keeping any metallic objects (eg. Battery terminal plates) at least 15mm away from the antenna as this is the distance of the near field for electromagnetic field.
4. Power lines should be thick and short. Big via holes are recommended whenever needed.
5. C37 and C34, C55 and C54, C57 and C56, should be placed as near as possible to pin 5, pin 4 and pin 56 respectively for effective decoupling.
6. C39 and C38, C61 and C59, C58 and C40, should be placed as near as possible to pin 57, pin 2 and pin 8 respectively for effective decoupling.
7. The ground pad beneath the centre of the ADNS-7630 QFN package should have sufficient via holes down to the same ground plane (2nd layer of the PCB). Use solder mask to prevent any unwanted short circuit. Prepare necessary area of solder pads only.
8. Components connected to CPOUT (pin 3) and VCTRL (pin 55) must as close as possible to ADNS-7630 IC. It is recommended to complete the loop within the same PCB layer.
9. Keep sufficient clearance between RF Trace class_1 (from pin ANTN to Antenna) and Ground copper (if applicable) on the top side 3 times larger than h (height of top layer to GND layer); the same requirement is needed for RF Trace class_2 (from pin ANTP to Antenna) and Ground copper (if applicable). Keep a clearance between VDD_RX (pin 5) and ANTN (pin 6) traces, as well as between ANTP (pin 7) and VDD_IF (pin 8) traces.
10. Keep ANTN and ANTP traces (from IC to antenna) parallel, short and as straight as possible without many curves. Recommended to have differential impedance between ANTN and ANTP to be 100Ω, and unbalanced trace (from C4 to ANTENNA) impedance controlled to 50Ω.
11. Keep a clearance between antenna and ground.
12. Ensure large grounding plane and more via holes at GND (pin 27, pin 32 and pin 33) down to the ground plane (2nd layer of the PCB).
13. Components connected to the pins below MUST complete the loop within the same PCB layer (no usage via holes allowed).
    a. BIASVAR (pin 54)
    b. REGO (pin 36)
    c. VDD3 (pin 31, 35, 50)
14. C17 must be as close as possible to the ADNS-7630 IC.
15. All separate AGND, GND_RF and GND paths MUST be via down to the same ground plane (2nd layer of the PCB). Ensure large grounding plane on the PCB layout for better performance on ESD and EFTB.
16. All caps MUST be as close to the power pins as possible, with the smaller capacitors nearer to the ADNS-7630 IC.
17. Frequency tolerance of crystal oscillator should follow the specification of +/- 20PPM. Recommended to use TST TZ0683B 12MHz crystal. Crystal should be placed less than 10mm (must not be more than 15mm) from ADNS-7630 XTALIN and XTALOUT pins.
18. Ceramic non-polarity caps and tantalum polarity capacitors are recommended.
19. Capacitors connected to VDD3 MUST have less than 0.2Ω ESR.
20. It is optional but highly recommended for customers to route some signals to a 2mm pin header (only to be soldered when troubleshooting is needed) on the mouse board to ease Avago's technical support in future. Refer to Design Guide – Hardware for more information.
21. Ensure that no component is placed at the lens clearance area as shown in Figure 4 so that the lens is interlocked to the PCB at the correct vertical height.
22. Add an optional π-type filter at antenna circuit to suppress 4.8G/7.2GHz harmonics.

*Disclaimer: All designers and manufacturers of this design must assure that they have all necessary intellectual property rights.

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	

Z-Wheel

ADNS-7630 can support both mechanical and optical Z-wheel design. Selection of Mechanical or Optical Z-Wheel interface can be set from EEPROM Z_Selection register (0x0137). The Z-Wheel reporting format which determines the vertical scroll resolution can be configured to Z/2 or Z/4 format when using different sensitivity optical Z-Wheel via EEPROM Z_Configuration register (0x0138). For mechanical Z-Wheel, Z/2 format is widely used as most of the commonly available mechanical Z-Wheel encoders come with low sensitivity. Optical Z-Wheel can utilize either the Z/2 or Z/4 format according to the desired sensitivity. Z_Negate (0x0139) enables correct Z-Wheel orientation in case ZA and ZB are swapped.

For mechanical Z-Wheel design, only ZA and ZB pins are connected to the physical mechanical encoder. ZLED pins should be floated (No Connect). For optical Z-Wheel design, connect all ZA, ZB and ZLED pins appropriately to the physical optical encoder system.

The direction of the Z-Wheel (positive or negative) based upon the Z-Wheel's quadrature output is shown in the state diagram below. State is shown in the form ZB ZA. Z-Wheel counts are reported only for transitions with + or - signs.

Tilt-Wheel

ADNS-7630 can support Tilt Wheel function via TW1 and TW2 pins by activating it through EEPROM register Tilt_Wheel_Enabled (0x0114). For applications without Tilt-Wheel, TW1 (GPIO3) and TW2 (GPIO4) pins can be configured as LED GPIO via the same register above.

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

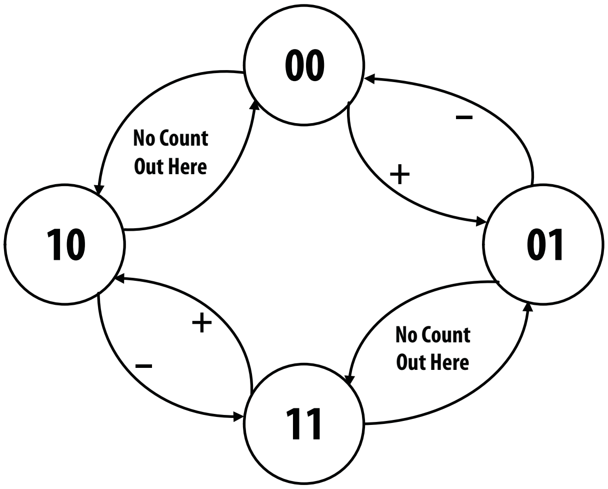


Figure 15. State Diagram for Z-Wheel

Connect Button

It is a must to have a "Connect" button in a Bluetooth mouse design to enable end users to initiate pairing/ unpairing with any Bluetooth host. Connect_Button_ Press_Duration register (0x00a4-0x00a5) allows mouse manufacturers to define duration needed for the "Connect" button to be held for a valid button pressed.

Connect "Connect" button to CONNECT pin (pin 13) for this feature.

Discover LED Indicator

It is highly recommended for mouse manufacturers to include a discover LED indicator in a Bluetooth mouse design as it enables end users to know if the mouse has entered discoverable mode successfully. See registers 0x0115-0x0119 to enable/disable discover LED support, to assign GPIO pin to be used, to define GPIO state to turn on the discover LED as well as the LED's duty cycle.

Connect the physical discover LED to LED0 (GPIO5) pin or LED1 (GPIO6) pin for this feature.

Battery LED Indicator

It is highly recommended for mouse manufacturers to include a battery LED indicator in a Bluetooth mouse design as it alerts end users when the battery power is running low, and also to remind end users to change the batteries. See registers 0x011a-0x011e, 0x0238-0x23d and 0x248 for the following configuration:

- enable/disable battery LED support
- assign GPIO pin to be used
- define GPIO state to turn on the battery LED, the LED's duty cycle, the blink/rest duration, active sniff modes and total duration
- define whether the LED is disabled before connection is established.

Connect the physical battery LED to LED0 (GPIO5) pin or LED1 (GPIO6) pin for this feature.

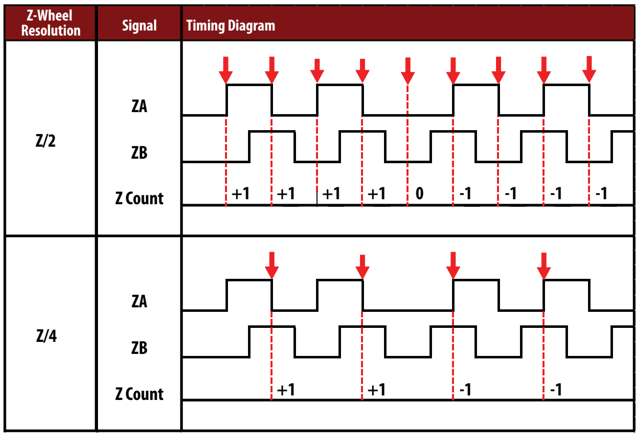


Figure 16. Timing Diagram for Z/2 and Z/4 settings

Basic Buttons & Programmable Buttons

There are a total of 3 basic buttons supported by ADNS-7630, namely B1 (left button), B2 (middle button) and B3 (right button). B4 (GPIO11) through B8/WP (GPIO15) are General Purpose Input/Output pins programmable to be buttons, LED indicator, or EEPROM write protect enabler. Access EEPROM register, Programmable_Buttons_Total (0x00d1) to define the total number of programmable buttons to be used in the mouse design. For all available programmable buttons, manufacturer can assign each button to a GPIO pin, as well as its function when the button is clicked once, double clicked or pressed for a specified duration.

Buttons, B1 through B8, TW1 and TW2 are connected to a Schmidt trigger input with 100μA current sources pulling up to +3V during run and rest modes. When used as buttons, the minimum time between button presses is T_{DBB}. T_{DBB} is programmable via the EEPROM (0x021e). The buttons are sampled every 4ms (default), typically. Five consecutive low values create a button press event. Five consecutive high values create a button release event. This is applicable to all single button click function.

ADNS-7630 also support double-click and button long-press features. The double click interval and long-press duration of each programmable button is configurable via EEPROM registers. However, button double click is functional only if SPP is disabled. Long-press duration should be programmed significantly longer than the single click duration so that end users will not be confused between single click and long press functions.

To define explicit functions single click, double click and long press functions for each programmable buttons, manufacturers can either assign the On-the-Fly (OTF) Resolution Mode or KeyMap (KM) feature through Keyboard Code A and Keyboard Code B. Refer to next section on detail description on KeyMap and On-the-Fly (OTF) Resolution Mode implementation.

On-the-Fly (OTF) Resolution Mode

The ADNS-7630 sensor is enhanced with programmable On-the-Fly (OTF) resolution mode, in which user is able to switch resolution setting anytime with OTF button single click, double click or long press. Any two available GPIOs between GPIO11-GPIO15 can be used to configure as the OTF resolution buttons. There are two types of OTF resolution mode:

a. Step by step increment or decrement using CPI+ and CPI- buttons:

   This method requires two GPIOs namely CPI+ and CPI- programmable buttons to increase or decrease the resolution setting step by step. There is a maximum of 10 resolution settings which can be enabled through EEPROM. If the current resolution setting is either in maximum or minimum level, any new button press will remain at the respective maximum or minimum level.

b. Rotational state change using CPI rotation button:

   This method requires only one GPIO to be programmed as CPI rotation button for incremental state change of resolution settings as configured in EEPROM. There is a maximum of 10 resolutions which can be enabled through EEPROM.

This OTF Resolution Mode can be enabled or disabled through EEPROM register Resolution_Selection_Method (0x0141). The OTF resolution mode types, step by step increment or decrement or rotational state change can be configured through SingleClick, LongPress or DoubleClick function in Button Configuration.

Mouse manufacturers can limit the total possible resolution settings to maximum of ten via EEPROM Resolution_ Selection _Total register (0x0142). To define all resolution settings, access registers 0x0144-0x014d. The values must be valid resolution range from 250cpi to 3000cpi.

The OTF current resolution state can be displayed with LED indication via any available GPIO between GPIO3-GPIO6 and GPIO11-GPIO15. These GPIOs can be configured to be active high output and the blinking duty cycle can also be determined via EEPROM.

Mouse manufacturers can use up to 4 GPIO to support resolution LED indicators. Refer to registers 0x011f-0x0123 for total GPIO to be used and each GPIO assignment. As there is a maximum of ten possible resolution settings, there is also a maximum of ten possible resolution LED indicator settings via registers 0x0124-0x012d. Duration for resolution LEDs to be lighted up can also be programmed via Resolution_LED_Duration (0x012e). For optimized power saving purposes, it is recommended that the LEDs are lighted up for a short moment once there is a change in the resolution setting. To define GPIO state to turn on the resolution LED indicator as well as the LED's duty cycle, access EEPROM registers 0x012f-0x0131.

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.

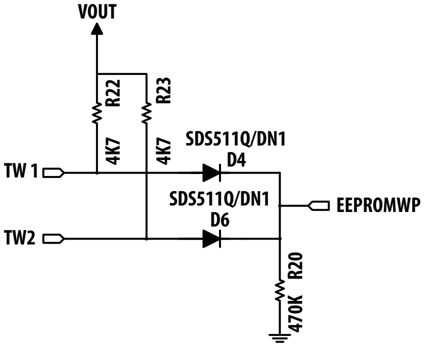


Figure 17. “Pseudo I/O” for EEPROM Write Protect Function

Typical Performance Characteristics

The following graphs are the typical performance of the ADNS-7630 sensor, assembled as shown in the 2D assembly drawing with the ADNS-7100-001 lens.

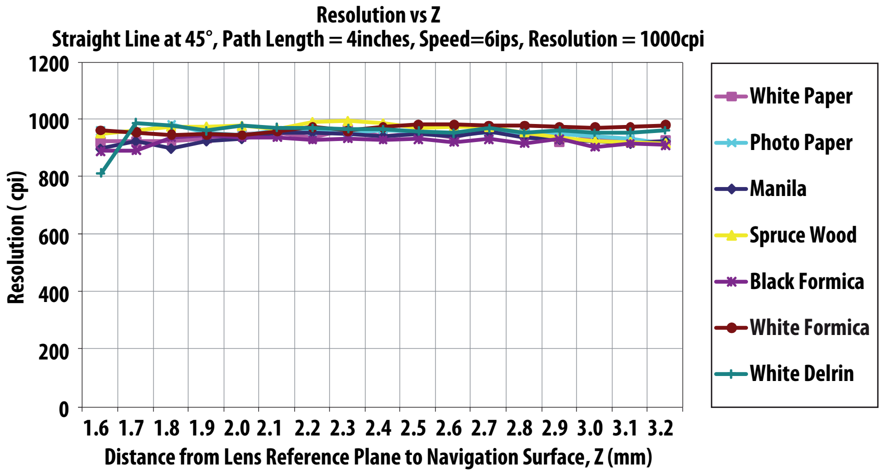


Figure 18. Mean Resolution vs. Z at 1000cpi

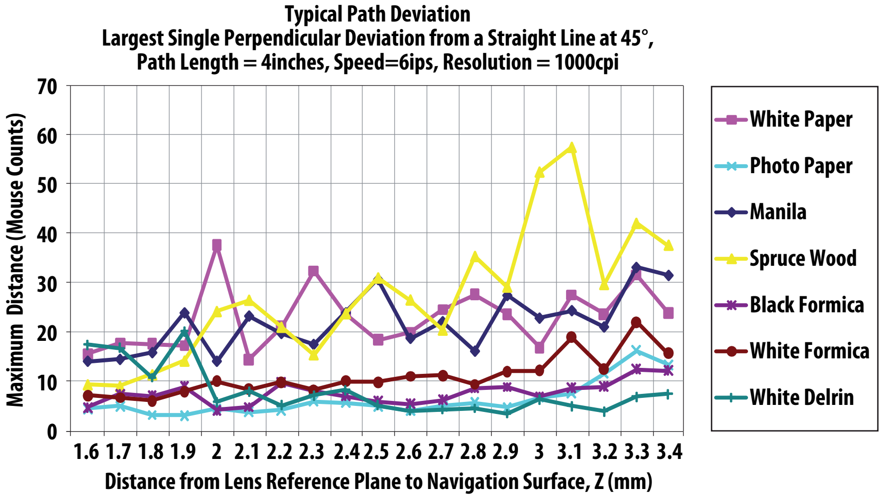


Figure 19. Average Error vs. Distance at 1000cpi (mm)

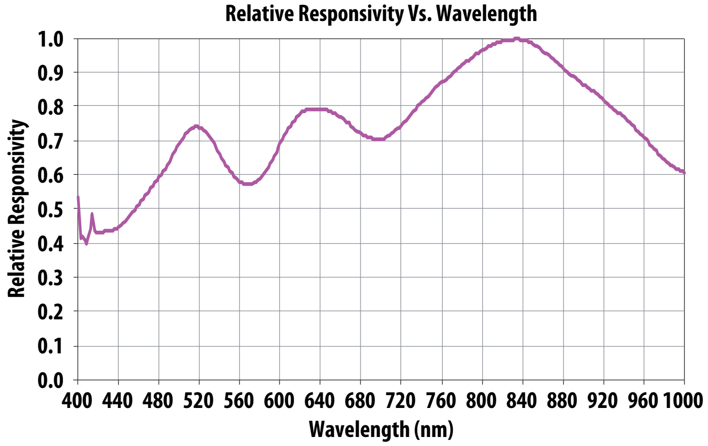


Figure 20. Wavelength Responsivity

Configuration after Power Up (Data Values)

Signal Function	Powered or Default Address or Configured	Suspended from any other states
B1	Pullup active for button use	Pullup active for button use
B2	Pullup active for button use	Pullup active for button use
B3	Pullup active for button use	Pullup active for button use
B4	Pullup active for button use	Pullup active for button use
B5	Pullup active for button use	Pullup active for button use
B6	Pullup active for button use	Pullup active for button use
B7	Pullup active for button use	Pullup active for button use
B8	Pullup active for button use	Pullup active for button use
TW1	Pullup active for button use	Pullup active for button use
TW2	Pullup active for button use	Pullup active for button use
-VCSEL	Pulsing	Pulled high (off)
ZA	Hi-Z input	Hi-Z input
ZB	Hi-Z input	Hi-Z input

Bluetooth HID Data Packet Format for 12-Bit Motion Format, 3/5 Buttons, Z-Wheel and Tilt-Wheel Mouse

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Byte 1	0	0	0	FB*	BB*	MB	RB	LB
Byte 2	X[7]	X[6]	X[5]	X[4]	X[3]	X[2]	X[1]	X[0]
Byte 3	Y[3]	Y[2]	Y[1]	Y[0]	X[11]	X[10]	X[9]	X[8]
Byte 4	Y[11]	Y[10]	Y[9]	Y[8]	Y[7]	Y[6]	Y[5]	Y[4]
Byte 5	Z[7]	Z[6]	Z[5]	Z[4]	Z[3]	Z[2]	Z[1]	Z[0]
Byte 6	TW[7]	TW[6]	TW[5]	TW[4]	TW[3]	TW[2]	TW[1]	TW[0]

* For 3 buttons mouse, FB = BB = 0.

Bluetooth HID Data Packet Format for 12-Bit Motion Format, 3/5 Buttons, Z-Wheel, Non Tilt-Wheel Mouse

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Byte 1	0	0	0	FB*	BB*	MB	RB	LB
Byte 2	X[7]	X[6]	X[5]	X[4]	X[3]	X[2]	X[1]	X[0]
Byte 3	Y[3]	Y[2]	Y[1]	Y[0]	X[11]	X[10]	X[9]	X[8]
Byte 4	Y[11]	Y[10]	Y[9]	Y[8]	Y[7]	Y[6]	Y[5]	Y[4]
Byte 5	Z[7]	Z[6]	Z[5]	Z[4]	Z[3]	Z[2]	Z[1]	Z[0]

* For 3 buttons mouse, FB = BB = 0.

Registers

The ADNS-7630 can be programmed or customized through an external EEPROM. Below is the list showing the EEPROM register mapping. Programmable configurations include basic mouse information, connection settings, mouse generic configuration, sensor configuration, and Bluetooth SDP configuration. Any changes on the register default value will effect the specifications and characteristics of ADNS-7630 as stated in this data sheet.

		Register Address			**Default Value**	
Domain	**Register Name**	**Bluetooth Ver2.0**	**Bluetooth Ver2.1**	**Byte Size**	**Bluetooth Ver2.0**	**Bluetooth Ver2.1**
Basic Information						
Firmware Version	Firmware_Version	0x0304-0x0305		2	1.0	
Bluetooth Address	Bluetooth_BDAddress	0x0008-0x000b		6	00:19:4d:11:22:33	
Pin Code	Pin_Length	0x000e		1	4	
	Pin_Code	0x000f-0x0012		4	0000	
Device	Device_Name_Length	0x0013		1	11	
	Device_Name	0x0014-0x0053		64	Avago Mouse	
Bluetooth 2.1 Function	Bluetooth_2.1_Function_Support	0x00d3		1	0x00	0x0f
	Secure Simple Pairing	Bit-0		1 bit	False	True
	Encryption Pause Resume	Bit-1		1 bit	False	True
	Extended Inquiry Response	Bit-2		1 bit	False	True
	Sniff Sub Rating	Bit-3		1 bit	False	True
Connection Configuration						
Power On Configuration	Mouse_Power_Up_Mode	0x0234		1	Discoverable Mode	
	Auto_Reconnect_Enabled	0x0249		1	True	
LMP Supervision Timeout	LMP_Supervision_TimeOut	0x0094-0x00a5		2	8000	
Inquiry and Page Parameters	Page_Scan_Window	0x0096-0x0097		2	768	
	Page_Scan_Interval	0x0098-0x0099		2	1024	
	Inquiry_Scan_Window	0x009a-0x009b		2	768	
	Inquiry_Scan_Interval	0x009c-0x009d		2	1024	
	Inquiry_Scan_TimeOut	0x009e-0x009f		2	2250	
	Page_TimeOut	0x00a0-0x00a1		2	0	
Pairing and Authentication Mode	Pairing_Mode	0x00a2		1	True	
	Authentication_Mode	0x00a3		1	False	
Connect Button Configuration	Connect_Button_Press_Duration	0x00a4-0x00a5		2	12	
	VC_Unplug_Enable	0x00a6		1	True	
Low Power Configuration	Max_Sniff_Modes	0x00a7		1	3	
	Sleep_Mode_Enabled	0x00a8		1	True	
	Wake_Up_Method	0x00a9		1	All event	
	Sniff_Retry_Count	0x00aa		1	3	
	Sniff_Retry_Interval	0x00ab-0x00ac		2	1600	
	Sniff_Timeout	0x00ad-0x00ae		2	1	
	Sniff_Attempt	0x00af-0x00b0		2	2	
	Sniff_Mode_Interval0	0x00b1-0x00b2		2	18	
	Sniff_Mode_Interval1	0x00b3-0x00b4		2	108	
	Sniff_Mode_Interval2	0x00b5-0x00b6		2	468	
	Sniff_Mode_Interval3	0x00b7-0x00b8		2	0	
	Sniff_Mode_Interval4	0x00b9-0x00ba		2	0	
	Sniff_Mode_Interval5	0x00bb-0x00bc		2	0	
	Sniff_Mode_Interval6	0x00bd-0x00be		2	0	
	Sniff_Mode_Interval7	0x00bf-0x00c0		2	0	
	Sniff_Mode_Duration0	0x00c1-0x00c2		2	178	
	Sniff_Mode_Duration1	0x00c3-0x00c4		2	900	
	Sniff_Mode_Duration2	0x00c5-0x00c6		2	2050	
	Sniff_Mode_Duration3	0x00c7-0x00c8		2	0	
	Sniff_Mode_Duration4	0x00c9-0x00ca		2	0	
	Sniff_Mode_Duration5	0x00cb-0x00cc		2	0	
	Sniff_Mode_Duration6	0x00cd-0x00ce		2	0	
	Sniff_Mode_Duration7	0x00cf-0x00d0		2	0	

Registers (continued)

		Register Address			Default Value	
Domain	**Register Name**	**Bluetooth Ver2.0**	**Bluetooth Ver2.1**	**Byte Size**	**Bluetooth Ver2.0**	**Bluetooth Ver2.1**
Mouse Generic Configuration						
Button Configuration						
Button Configuration	Programmable_Buttons_Total	0x00d1		1	0	
	Programmable_Buttons_Low_ Power	0x00d2		1	1	
	Debouncing_Time	0x021e		1	4	
Programmable Button 1	GPIO_Pin_Selection1	0x00d4		1	0	
	Single_Click_Function1	0x00d5		1	Not Support	
	Single_Click_Repeat_Delay1	0x00d6	N/A	1	0	
	Long_Press_Function1	0x00d7	0x00d6	1	Not Support	
	Long_Press_Duration1	0x00d8	0x00d7	1	25	
	Double_Click_Function1	0x00d9	N/A	1	Not Support	
	Double_Click_Interval1	0x00da	N/A	1	6	
	User_Defined_Function_1_A	0x025b-0x0264	0x0287-0x0290	10	a1 00 00 00 03 00 00 00 00 00	
	User_Defined_Function_1_B	0x0266-0x026f	0x0292-0x029b	10	a1 00 00 00 03 00 00 00 00 00	
	User_Defined_Function_1_C	0x0271-0x027a	N/A	10	a1 00 00 00 03 00 00 00 00 00	
Programmable Button 2	GPIO_Pin_Selection2	0x00db	0x00e0	1	0	
	Single_Click_Function2	0x00dc	0x00e1	1	Not Support	
	Single_Click_Repeat_Delay2	0x00dd	N/A	1	0	
	Long_Press_Function2	0x00de	0x00e2	1	Not Support	
	Long_Press_Duration2	0x00df	0x00e3	1	25	
	Double_Click_Function2	0x00e0	N/A	1	Not Support	
	Double_Click_Interval2	0x00e1	N/A	1	6	
	User_Defined_Function_2_A	0x027c-0x0285	0x029d-0x02a6	10	a1 00 00 00 03 00 00 00 00 00	
	User_Defined_Function_2_B	0x0287-0x0290	0x02a8-0x02b1	10	a1 00 00 00 03 00 00 00 00 00	
	User_Defined_Function_2_C	0x0292-0x029b	N/A	10	a1 00 00 00 03 00 00 00 00 00	
Programmable Button 3	GPIO_Pin_Selection3	0x00e2	0x00ec	1	0	
	Single_Click_Function3	0x00e3	0x00ed	1	Not Support	
	Single_Click_Repeat_Delay3	0x00e4	N/A	1	0	
	Long_Press_Function3	0x00e5	0x00ee	1	Not Support	
	Long_Press_Duration3	0x00e6	0x00ef	1	25	
	Double_Click_Function3	0x00e7	N/A	1	Not Support	
	Double_Click_Interval3	0x00e8	N/A	1	6	
	User_Defined_Function_3_A	0x029d-0x02a6	0x02b3-0x02bc	10	a1 00 00 00 03 00 00 00 00 00	
	User_Defined_Function_3_B	0x02a8-0x02b1	0x02be-0x02c7	10	a1 00 00 00 03 00 00 00 00 00	
	User_Defined_Function_3_C	0x02b3-0x02bc	N/A	10	a1 00 00 00 03 00 00 00 00 00	

Registers (continued)

		Register Address			Default Value	
Domain	**Register Name**	**Bluetooth Ver2.0**	**Bluetooth Ver2.1**	**Byte Size**	**Bluetooth Ver2.0**	**Bluetooth Ver2.1**
Battery Configuration						
Battery Configuration	Battery_Alarm_Power	0x01c0		1	2.2V	
	Battery_Alarm_Time	0x01c1		4	125	
	Battery_Sleep_Power	0x0203		1	2.0V	
Sensor Configuration						
Resolution	Max_Resolution	0x013d		1	3000	
	Default_Resolution	0x013e		1	1250	
CPI Selection	Resolution_Selection_Method	0x0141		1	Not Support	
	Resolution_Selection_Total	0x0142		1	0	
	Current_Resolution_Selection	0x0143		1	0	
	Resolution_Setting1	0x0144		1	0	
	Resolution_Setting2	0x0145		1	0	
	Resolution_Setting3	0x0146		1	0	
	Resolution_Setting4	0x0147		1	0	
	Resolution_Setting5	0x0148		1	0	
	Resolution_Setting6	0x0149		1	0	
	Resolution_Setting7	0x014a		1	0	
	Resolution_Setting8	0x014b		1	0	
	Resolution_Setting9	0x014c		1	0	
	Resolution_Setting10	0x014d		1	0	
SDP Configuration						
Vendor Information	SDP_Service_Name	0x014e-0x018d		64	Avago Bluetooth Mouse	
	SDP_Service_Name_Length	0x018e		1	21	
	SDP_Service_Description	0x018f-0x019e		16	A Mouse	
	SDP_Service_Description_Length	0x019f		1	7	
	SDP_Provider_Name	0x01a0-0x01af		16	Avago	
	SDP_Provider_Name_Length	0x01b0		1	5	
	SDP_Vendor_ID	0x01b1-0x01b2		2	abcd	
	SDP_Product_ID	0x01b3-0x01b4		2	1234	
	SDP_Product_Version	0x01b5-0x01b6		2	100	
BQB-Specific SDP Configuration	HID_Attribute_Length	0x01ff		1	0x9f	
	HID_Attribute_Offset	0x0201		1	0xeb	
EEPROM Configuration						
EEPROM Write Protection	EEPROM_WP_Flag	0x021f		1	GPIO	

Basic Information

Firmware_Version

Size: 2 byte Default Value: **1.0**

USAGE: This register contains the firmware version. Value 2 means version 2.0. This register value is fixed and not programmable.

Bluetooth_BDAddress

Size: 6 byte Default Value: **00:19:4d:11:22:33**

USAGE: This register contains the Bluetooth address in hexadecimal. The format should be 00:11:22:33:44:55, where "00:11" are NAP (Non- Significant Address Part), "22" is UAP (Upper Address Part) and "33:44:55" are LAP (Lower Address Part).

Pin_Length

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

USAGE: This register contains the fixed PIN code length. Value 4 means the length of the PIN code is 4 bits.

Pin_Code

Size: 4 byte Default Value: **0000**

USAGE: This register contains the PIN code in format of 4-byte alphanumeric string and special characters. Value 0000 means the PIN code is "0000".

Device_Name_Length

Size: 1 byte Default Value: **11**

USAGE: This register contains the length of the mouse Device Name.

Device_Name

Size: up to 64 bytes Default Value: **Avago Mouse**

USAGE: This register contains the mouse Device Name.

Bluetooth_2.1_Function_Support

Size: 1 byte Default Value: **0x00**

USAGE: This register defines which Bluetooth V2.1 feature(s) is supported.

bit 0: Secure Simple Pairing (SSP)
bit 1: Encryption Pause and Resume (EPR)
bit 2: Extended Inquiry Response (EIR)
bit 3: Sniff Sub Rating (SSR)
bit 4~7: Reserved

Secure simple pairing – Set whether to support secure simple pairing which supports "Just Works" and to enhance ease of use user experience.

Encryption pause resume – Set whether to support encryption pause and resume where better protection through encryption key refreshed during long connection period of use.

Extended inquiry response – Set whether to support extended inquiry response to enable fast discovery of device and to reduce latency.

Sniff sub-rating – Set whether to support sniff sub-rating which reduces power consumption for HID.

For example, 5 (i.e. 0x05) means SSP and EIR are supported, and the other 2 features are not supported. Set to 0 to disable support for all four Bluetooth V2.1 features.

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.

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.

Reconnect_Power_On_LED_Enabled

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

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

Reconnect_Power_On_LED_PIN

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

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

Reconnect_Power_On_LED_GPIO_State

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

USAGE: This register defines the GPIO value which causes the reconnect 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 reconnect power-on LED to turn on, and GPIO value "0" causes reconnect power-on LED to turn off.

Reconnect_Power_On_LED_On_Duration

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

USAGE: This register defines reconnect 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_Enabled

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

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

Discover_LED_PIN

Size: 1 byte Default Value: **6**

USAGE: This register defines which GPIO pin the discover LED is connected to. Only GPIO5 and GPIO6 are valid options. For example, 6 means the discover LED is connected to GPIO6.

Discover_LED_GPIO_State

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

USAGE: This register defines the GPIO value which causes the discover 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 discover LED to turn on, and GPIO value "0" causes discover LED to turn off.

Discover_LED_On_Period

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

USAGE: This register defines discover 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.

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.

BQB-Specific SDP Configuration

HID_Attribute_Length

Size: 1 byte Default Value: **0x9f**

USAGE: Define the HID descriptor list length. For example, HID descriptor list is 09 02 06 35 02 35 00 …, so the length is 7. This control byte is just for the BQB HID test.

HID_Attribute_Offset

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

USAGE: Define the offset of HID descriptor list. For example, HID descriptor list is 09 02 06 35 02 35 00 …, so the offset is length from the beginning of SDP information to 09. This control byte is just for the BQB HID test.

EEPROM Configuration

EEPROM_WP_Flag

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

USAGE: Define the way to support EEPROM write protection. If activated, the function will prevent all accidental write operations to EEPROM.

Set to "Not support" to disable EEPROM write protection;

Set to "GPIO" to support EEPROM write protection via GPIO15 (B8/WP);

Set to "TWheel" to support EEPROM write protection via T-wheel pins.

For product information and a complete list of distributors, please go to our web site: **www.avagotech.com**