ADNK-7553-SP11
USB Laser Mouse Designer's Kit

Avago TECHNOLOGIES

Design Guide

Lead (Pb) Free
RoHS 6 fully
compliant

Introduction

The Universal Serial Bus (USB) is an industry standard serial interface between a computer and peripherals such as a mouse, joystick, keyboard, UPS, etc. This design guide describes how a cost-effective USB optical mouse can be built using the Sunplus, SPCP868A USB microcontroller and the Avago Technologies ADNS-7550 optical sensor. The document starts with the basic operations of a computer mouse peripheral followed by an introduction to the Sunplus SPCP868A USB microcontroller and the Avago Technologies ADNS-7550 Optical Navigation Sensor. A schematic of the Sunplus SPCP868A USB microcontroller to the ADNS-7550 optical sensor and buttons of a standard mouse can be found in Appendix A. The software section of this application note describes the architecture of the firmware required to implement the USB mouse functions. The ADNS-7550 data sheet is available from the Avago Technologies web site at www.avagotech.com. USB documentation can be found at the USB Implementers Forum web site at www.usb.org.

The Avago Technologies ADNS-7550 laser mouse sensor, an 16-pin integrated molded lead-frame staggered dual inline package (DIP), is based on LaserStream™ Technology, which measures changes in position by optically acquiring sequential surface images (frames) and mathematically determining the direction and magnitude of movement.

The ADNS-7550 sensor along with the ADNS-6150, ADNS-6160-001 or ADNS-6170-002 lens, form a complete and compact laser mouse tracking system. There is no moving part, which means high reliability and less maintenance for the end user. In addition, precision optical alignment is not required, facilitating high volume assembly.

Features

- Complete laser mouse reference design kit
- Windows® 98SE, Windows® 2000, Windows® XP and Windows® Vista compatibility
- USB full speed data transfer rate
- User identity code to avoid conflict with other devices
- High reliability
- New LaserStream™ Technology
- Smooth surface navigation
- Enhanced SmartSpeed self-adjusting frame rate for optimum performance
- High speed motion detection up to 30 ips and 8g
- 400, 800, 1200 and 1600 cpi resolution
- No mechanical moving parts
- Laser fault detect circuitry on-chip for Eye Safety Compliance
- Transmission data rate up to 1 Mbps
- Sunplus, SPCP868A USB microcontroller
- USB interface for PC connection
- 8K Bytes of internal EPROM provided in the microcontroller to enable easy customization of firmware and lower cost
- Conforms to USB suspend mode specifications
- Internal oscillator – no clock input needed
- Minimal number of passive components.

Optical Mouse Basics

The image-based optical mouse sensor takes snap shots of the surface it is navigating on. It measures changes in position by comparing the sequential images (frames) and mathematically determines the direction and magnitude of movement. The traditional duel-channel optical encoder generates the quadrature Z-wheel movement signals. This design guide illustrates the hardware connection of a laser mouse with standard configuration; as well as the firmware management and the handling of the USB protocols. USB protocol provides a standard way of reporting mouse movement and button presses to the PC. The Windows HID driver interprets the USB data and performs the cursor movements and mouse clicks.

The functional block diagram of the reference design mouse is shown in Figure 1. The optical sensor detects the X and Y movements. A mechanical quadrature encoder provides the Z-wheel movement. Each of the button switches is pulled up normally and provides a Ground when pressed.

Introduction to Sunplus, SPCP868A

The Sunplus, SPCP868A is a general purpose OTP USB microcontroller. It has dual USB speed, namely low and full speed. It can also support PS/2 mode. The SPCP868A has a inbuilt 6Mhz oscillator and an 18/24MHz PLL. The SPCP868A has a 8-bit CPU core, capable of running at up to 12MHz clock rate. It features 8k bytes of program memory and 128 bytes of RAM.

Serial Peripheral Interface (SPI)

The Sunplus SPCP868A provides a SPI compatible interface. The SPI circuit supports byte serial transfer in either Master or Slave mode. The integrated SPI circuit allows the Sunplus SPCP868A to communicate with external SPI compatible hardware, in this case the ADNS-7550.

Hardware Implementation

The standard hardware to implement a mouse is shown in Figure 1. For X and Y movement, the optical sensor is used. The Z- wheel movement is detected by a set of optical sensors that output quadrature signals. For each button there is a switch that is pulled up internally by the built in pull up resistors.

Optical Sensor

Avago Technologies ADNS-7550 laser mouse sensor is used in this reference design as the primary navigation engine. Based on LaserStream™ Technology, the ADNS-7550 contains an Image Acquisition System (IAS), a Digital Signal Processor (DSP) and a four-wire serial port. Its high-performance architecture is capable of sensing high-speed mouse motion. The SPCP868A microcontroller periodically reads the ADNS-7550's Delta_X_L, Delta_Y_L and Delta XY_H registers to obtain any horizontal and vertical motion information happening as a result of the mouse being moved. The output of the ADNS-7550 laser mouse optical sensor is four-wire serial port.
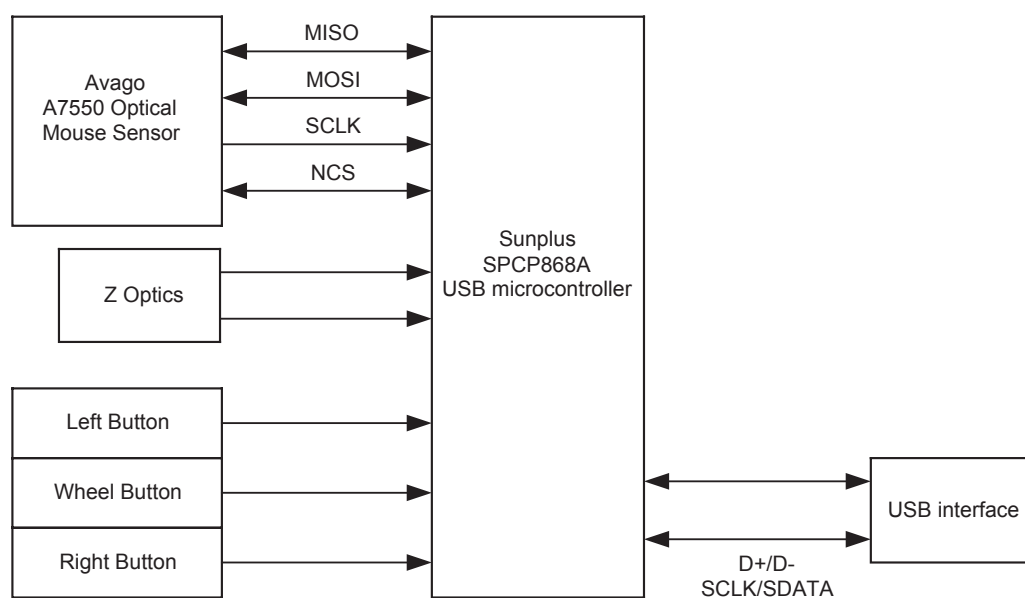

Figure 1. Sunplus SPCP868A –ADNS-7550 Optical Mouse Hardware Block Diagram

This motion information will be reported to the PC to update the position of the cursor. The advantages of using ADNS-7550 optical sensor are the best tracking accuracy, flexibility of programming the optical sensor via the SPI port, and the automatic frame rate feature. Besides, ADNS-7550 optical sensor performs excellent tracking on difficult surfaces which conventional LED based technology is unable to track such as glossy and smooth surfaces. The speed improvement is achieved by continuous data clocking to or from multiple registers.

Motion Read is activated by reading the Motion_Burst register. The ADNS-7550 will respond with the contents of the Motion, Delta_X_L, Delta_Y_L, Delta_XY_H, SQUAL, Shutter_Upper, Shutter_Lower and Maximum_Pixel registers in that order.

The ADNS-7550 sensor, along with the ADNS-6150, ADNS-6160-001 or ADNS-6170-002 lens form a complete and compact laser mouse tracking system. There are no moving parts, which mean high reliability and less maintenance for the end user. In addition, precision optical alignment is not required, facilitating high volume assembly.

Mechanical Z-Wheel

The motion of Z-wheel is detected using the traditional method by decoding the quadrature signal generated by mechanical encoder. The Z-pinwheel is connected to the Z-encoder through its shaft. The rotational movement of the shaft is decoded into on and off levels in a quadrature output pattern. Every change in the Z-encoder outputs represents a count of mouse movement. Comparing the last state of the Z-encoder to the current state derives direction information. In this reference design, only the motion at the Z-wheel is detected using this method.

Mouse Buttons

Mouse buttons are connected as standard switches. These switches are pulled up by the pull up resistors inside the microcontroller. When the user presses a button, the switch will be closed and the pin will be pulled

LOW to GND. A LOW state at the pin is interpreted as the button being pressed. A HIGH state is interpreted as the button has been released or the button is not being pressed. Normally the switches are debounced in firmware for 15-20ms. In this reference design there are three switches: left Z-wheel and right.

USB Connection

The Sunplus SPCP868A has a configuration register that switches control from the SIE to manual control on the D+ and D- pins. This allows the firmware to dynamically configure itself to operate as a USB mouse. The firmware for this reference design will automatically detect the host topology (USB). The connections for the connectors are shown in Figure 2 below.
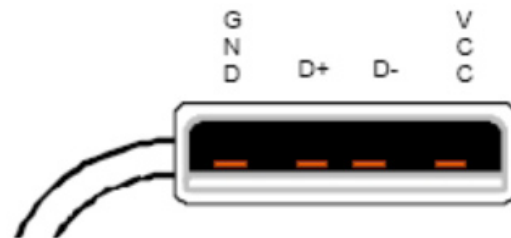
USB type-A Connector

Figure 2. USB peripheral connectors

Some details on ADNK-7553-SP11

The ADNK-7553-SP11 reference design mouse unit allows users to evaluate the performance of the Optical Tracking Engine (sensor, lens) over a USB connection. This kit also enables users to understand the recommended mechanical assembly. (See Appendix C, D, and E)

System Requirements

PCs using Windows®95, Windows®98, Windows®NT, Windows®2000, Windows®XP or Windows®Vista with standard 3-button USB mouse driver loaded.

Functionality

3-button, scroll wheel USB-mouse.

USB Operating Mode

Hot pluggable with USB port. The PC does not need to be powered off when plugging or unplugging the evaluation mouse.

To Disassemble the ADNK-7553-SP11 Unit

The ADNK-7553-SP11 comprises of the plastic mouse casing, printed circuit board (PCB), lens, buttons, and batteries (See Figure 3). Unscrewing the one screw located at the base of the unit can open the ADNK-7553-SP11 mouse unit. Lifting and pulling the PCB out of the base plate can further disassemble the mouse unit.

Caution: The lens is not permanently attached to the sensor and will drop out of the assembly.
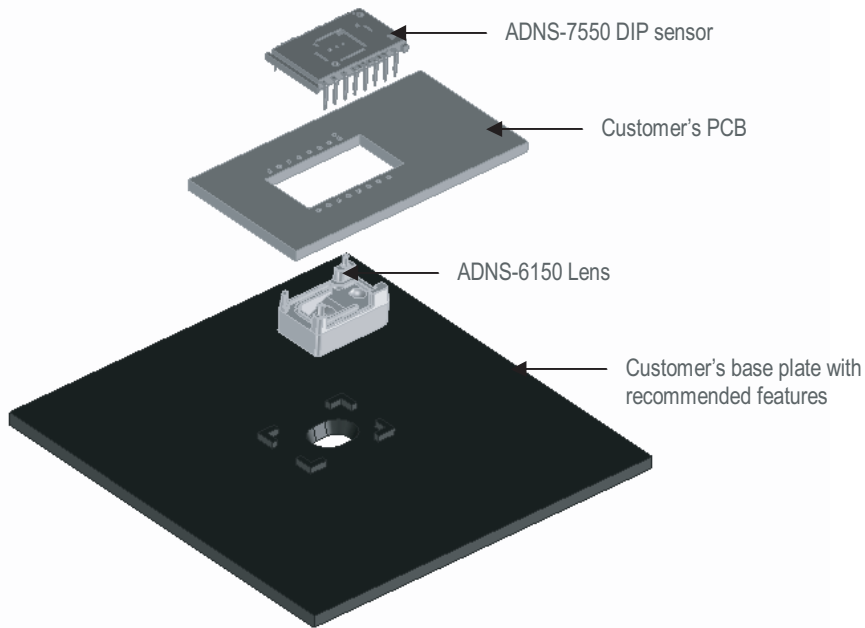
ADNS-7550 DIP sensor

Customer's PCB

ADNS-6150 Lens

Customer's base plate with recommended features

Figure 3a. Exploded view drawing of the ADNS-7550 laser mouse sensor with the ADNS-6150 lens, PCB and base plate.

ADNS-7550 DIP sensor

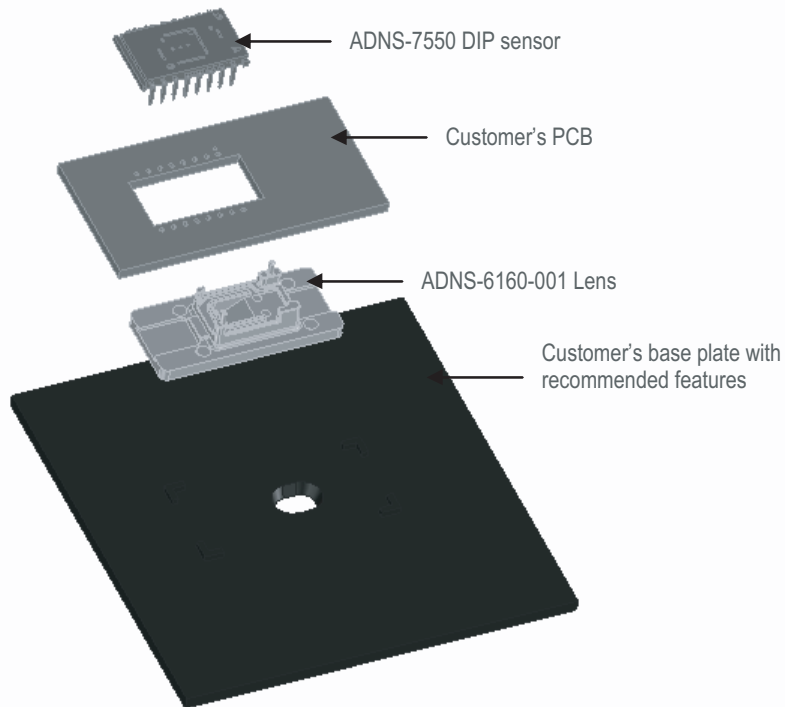Customer's PCB

ADNS-6160-001 Lens

Customer's base plate with recommended features

Figure 3b. Exploded view drawing of the ADNS-7550 laser mouse sensor with the ADNS-6160-001 lens, PCB and base plate.

4

ADNS-7550 DIP
Sensor

Customer's PCB

ADNS-6170-002
Lens

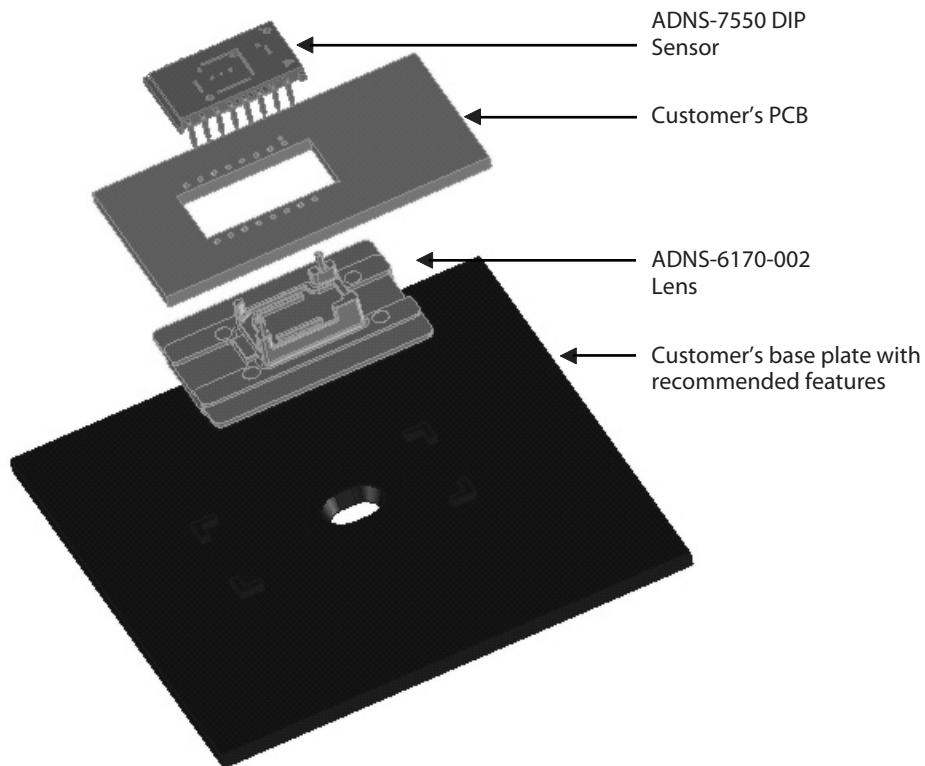Customer's base plate with
recommended features

Figure 3c. Exploded view drawing of the ADNS-7550 laser mouse sensor with the ADNS-6170-002 lens, PCB and base plate.

While reassembling the components, please make sure that the Z height (Distance from lens reference plane to surface) is valid. Refer to Figure 4.
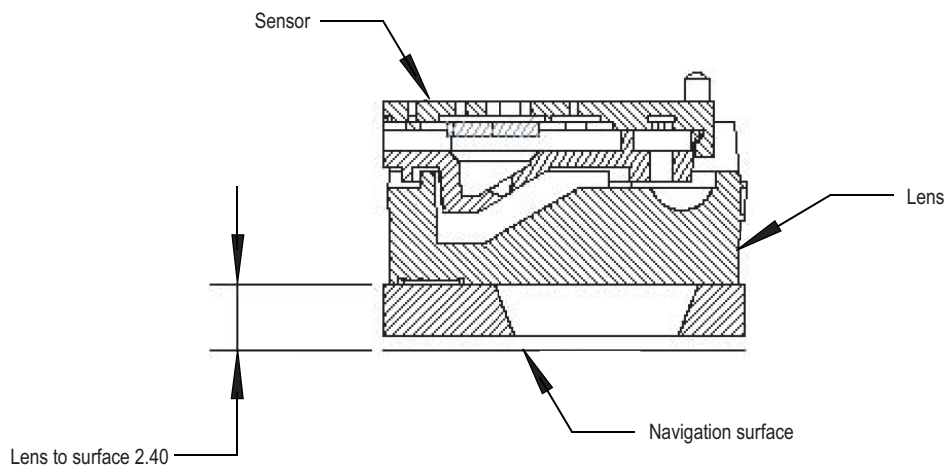

Sensor

Lens

Navigation surface

Lens to surface 2.40

Figure 4. Distance from lens reference plane to surface

Regulatory Requirements

- Passes FCC B and worldwide analogous emission limits when assembled into a mouse with shielded cable and following Avago Technologies recommendations.

- Passes IEC-1000-4-3 radiated susceptibility level 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.

- Provides sufficient ESD creepage/clearance distance to avoid discharge up to 15kV when assembled into a mouse according to usage instructions above.

- For eye safety consideration, please refer to the document, Eye Safety Calculation AN 5361 available on the Avago Technologies web site.

Eye Safety

The ADNS-7550 integrated molded lead-frame DIP sensor and the associated components are intended to comply with Class 1 Eye Safety Requirements of IEC 60825-1. Avago Technologies suggests that manufacturers perform testing to verify eye safety on each mouse. It is also recommended to review possible single fault mechanisms beyond those described below in the section "Single Fault Detection". Under normal conditions, the sensor generates the drive current for the VCSEL.

In order to stay below the Class 1 power requirements, LASER_CTRL0 (register 0x1a), LASER_CTRL1 (register 0x1f), LSRPWR_CFG0 (register 0x1c) and LSRPWR_CFG1 (register 0x1d) must be programmed to appropriate values. The ADNS-7550 integrated molded lead-frame DIP sensor which comprised of the sensor and VCSEL; is designed to maintain the output beam power within Class 1 requirements over components manufacturing tolerances and the recommended temperature range when adjusted per the procedure below and implemented as shown in the recommended application circuit of Figure A1. For more information, please refer to Eye Safety Application Note 5361.

LASER Power Adjustment Procedure

1. The ambient temperature should be 25C ± 5 °C.

2. Set V_{DD5} to its permanent value.

3. Set the Range bits (bit 7 and 6 of register 0x1a) to b'01.

4. Set the Range_C complement bits (bit 7 and 6 of register 0x1f) to b'10.

5. Enable the Calibration mode by writing to bits [3, 2, 1] of register 0x1A so the laser will be driven with 100% duty cycle.

6. Set the laser current to the minimum value by writing 0x00 to register 0x1c, and the complementary value 0xFF to register 0x1d.

7. Program registers 0x1c and 0x1d with increasing values to achieve an output power of not more than 506uW to meet class 1 Eye Safety over temperature. If this power is obtained, the calibration is complete, skip to step 12.

8. If it was not possible to achieve the power target, set the laser current to the minimum value by writing 0x00 to register 0x1c, and the complementary value 0xff to register 0x1d.

9. Set the Range and Range C bits in registers 0x1a and 0x1f, respectively, to choose to the higher laser current range.

10. Program registers 0x1c and 0x1d with increasing values to achieve an output power of not more than 506uW to meet class 1 Eye Safety over temperature.

11. Save the value of registers 0x1a, 0x1c, 0x1d, and 0x1f in non-volatile memory in the mouse. These registers must be restored to these values every time the ADNS-7550 is reset.

12. Reset the mouse, reload the register values from non-volatile memory, enable Calibration mode, and measure the laser power to verify that the calibration is correct.

Good engineering practices such as regular power meter calibration, random quality assurance retest of calibrated mice, etc. should be used to guarantee performance, reliability and safety for the product design.

LASER Output Power

The laser beam output power as measured at the navigation surface plane is specified below. The following conditions apply:

1. The system is adjusted according to the above procedure.

2. The system is operated within the recommended operating temperature range.

3. The V_{DD5} value is no greater than 300mV above its value at the time of adjustment.

4. No allowance for optical power meter accuracy is assumed.

Below is the summary of the components contained in the ADNK-7553-SP11 Designer's Kit.

Component Summary

Below is the summary of components contained in the ADNK-7553-SP11 Designer's Kit.

Sensor

The sensor technical information is contained in the ADNS-7550 Data Sheet.

Lens

The lens technical information is contained in the ADNS-6150, ADNS-6160-001 and ADNS-6170-002 Data Sheet.

LASER

The LASER technical information is contained in the Application Note 5088 Additional application notes regarding Eye Safety Requirements are also available at Avago Technologies website.

Base Plate Feature – IGES File

The IGES file on the CD-ROM provides recommended base plate molding features to ensure optical alignment. This includes PCB assembly diagrams like solder fixture in assembly and exploded view, as well as solder plate. See Appendix D for details.

Reference Design Documentation – Gerber File

The Gerber File presents detailed schematics used in ADNK-7553-SP11 in PCB layout form. See Appendix C for more details.

Overall circuit

A schematic of the overall circuit is shown in Appendix A of this document. Appendix B lists the bill of materials.

Firmware Implementation

The firmware for this reference design is written in the assembly language. The following files are required to compile the mouse firmware.

SPCP868A_A7550.asm - main mouse firmware

calibration_hid.asm – HID compliant device USB descriptor ROM tables. Load during calibration mode.

hiddesc3.asm - 3 buttons mouse mode USB descriptor ROM tables. Load during normal mouse mode.

pro_7550.asm – Routine to access ADNS-7550 sensor register.

SPCP868A.inc – The SPCP868A I/O registers definition.

ADNS7550.inc – ADNS-7550 interface constants.

delay.inc – SPCP868A delay loop subroutine.

decode_setup.inc – USB descriptor and request constants.

DET_Z.inc – SPCP868A Z-axis event handler.

DET_KEY.inc – SPCP868A button event handler.

At power up, the firmware examines the host interface and automatically determines the mouse USB host connection. Then the host firmware configures itself to operate on the detected interface.

USB Interface

All USB Human Interface Device (HID) class applications follow the same USB start-up procedure. The procedure is as follows

1. Device Plug-in

When a USB device is first connected to the bus, it is powered and running firmware, but communications on the USB remain non-functional until the host has issued a USB bus reset.

2. Bus Reset

When the host recognizes the presence of a new USB device it initiates a bus reset to that device.

3. Enumeration

The host initiates SETUP transactions that reveal general and device specific information about the mouse. When the description is received, the host assigns a new and unique USB address to the mouse. The mouse begins responding to communication with the newly assigned address, while the host continues to ask for information about the device description, configuration description and HID report description. Using the information returned from the mouse,

the host now knows the number of data endpoints supported by the mouse (3). At this point, the process of enumeration is completed.

Notes
1. idVendor should be changed to the value as supplied by the USB-IF
2. idProduct should be assigned for specific product.
3. MaxPower value should be changed as per specific circuit's current draw.

4. Post Enumeration Operation

Once communication between the host and mouse is established, the peripheral now has the task of sending and receiving data on the control and data endpoints. In this case, when the host configures endpoint 1, the mouse starts to transmit button and motion data back to the host when there is data to send. At any time the peripheral may be reset or reconfigured by the host.

USB Requests – Endpoint 0

Endpoint 0 acts as the control endpoint for the host. On power-up endpoint 0 is the default communication channel for all USB devices. The host initiates Control-Read and Control-Write (see Chapter 8 of the USB specification) to determine the device type and how to configure communications with the device. In this particular design, only Control-Read transactions are required to enumerate a mouse. For a list of valid requests see Chapter 9 of the USBG specification. In addition to the standard "Chapter 9" requests, a mouse must also support all valid HID class requests for a mouse.

USB Requests – Endpoint 1

Endpoint 1 is the data transfer communications channel for mouse button, wheel, and movement information. Requests to this endpoint are not recognized until the host configures endpoint 1. Once this endpoint is enabled, then interrupt IN requests are sent from the host to the mouse to gather mouse data. When the mouse is left idle (i.e. no movement, no new button presses, no wheel movement) the firmware will NAK requests to this endpoint. Data is only reported when there is a status change with the mouse.

Two HID report formats are used in this design. The boot protocol, as defined by the HID specification, is the default report protocol that all USB enabled systems understands. The boot protocol has a three-byte format, and so does not report wheel information. The HID report descriptor defines the report protocol format. This format is four bytes and is the same as the report format with the exception of the fourth byte, which is the wheel information. Appendix F of this document lists the USB Data Reporting Format.

USB Firmware Description

A function call map for USB operation is shown in Figure 6. The following are descriptions of the functions in SPCP868A _A7550.asm.

IO_initial – This function is use to set the Sunplus micro-controller as input or output. Port A(PA) is set as input while both Port B(PB) and Port C(PC) are set as output. This function also includes setting and enabling of pull-up resistor for left, right and middle keys.

clear_ram – This function clears the internal ram of the microcontroller.

Usb_initial – This function is used to enable the USB mode. This is done by enabling the watchdog and LVR and the low speed is selected. The USB reset event interrupt as well as the set up event interrupt are enabled.

DetectUsbReset – This routine initializes USB interface service and SPI ports. Then, the normal_mode variable is compared, if normal_mode is equal to '0' then this routine will invoke the calibration_operation routine to enter calibration loop. If normal_mode variable flag is equal to '1' then Read_LP_CFG_REG is called to load the calibrated LP_CFG0 and LP_CFG1 register value. Subsequently, the ADNS-7550 sensor is reset and AdjustLaser routine is invoked to write the LP_CFG0 and LP_CFG1 value to the sensor register. Then, SetShuttherMode routine is called to enable VCSEL in shutter mode and laser output is enabled.

Default_state – This function is used to set the default service routing.

sample_mouse – This routine returns any updates in the X, Y and Z-wheel motion information. The motion of the Z-wheel is detected using the traditional method by decoding the quadrature signal generated by the photo-transistors.. The X, Y, and Z-wheel movement is stored in the [xCount], [yCount], and [zCount] variables which will be sent to the host in the main routine.

UsbGetOpticalData - Reads the ADNS-7550 Delta_X_L, Delta_Y_L and Delta_XY_H registers for the X and Y movement.

DetectKeyLoop – This function is used to detect the left, right and middle buttons changes.

check_mouse_data –This function is used to determine whether the buttons, X, Y or Z-wheel data need to be sent to the host.

reset_adns7550 – This function is responsible for resetting the hardware. This function also sets the laser calibration values. All register settings should be set here prior to execution of the main program loop.

Report_mouse_data – This function is used to send buttons, X, Y and Z-wheel data to the computer.

USB Functions

Main_loop –This function spins in an infinite loop waiting for an event that needs servicing. sample_mouse and report_mouse_data are the functions which are called within this loop to retrieve any new motion or button information. The data received from these functions will be loaded into the endpoint 1 buffer to be sent to the host.

ep0SetupReceived – This routine is entered whenever a SETUP packet is received in on endpoint 0. It parses the packet and calls the appropriate routine to handle the packet.

ep0InReceived – This routine is entered whenever an IN packet is received on endpoint 0.

ep0OutReceived – This routine is entered whenever an OUT packet is received on endpoint 0.

setDeviceConfiguration – This routine is entered when a SET CONFIGURATION request has been received from the host.

setDeviceAddress – This routine is entered whenever a SET ADDRESS request has been received. The device address change cannot actually take place until after the status stage of this no-data control transaction, so the address is saved and a flag is set to indicate that a new address was just received. The code that handles IN transactions will recognize this and set the address properly.

setInterfaceIdle – This routine is entered whenever a SET IDLE request is received. See the HID specification for the rules on setting idle periods. This function sets the HID idle time. See the HID documentation for details on handling the idle timer.

setInterfaceProtocol – This routine is entered whenever a SET PROTOCOL request is received. This no-data control transaction enables boot or report protocol.

getInterfaceReport – This routine is entered whenever a GET REPORT request is received.

getInterfaceIdle – This routine is entered whenever a GET IDLE request is received. This function then initiates a control-read transaction that returns the idle time. See the HID class documentation for more details.

getDeviceConfiguration – This routine is entered whenever a GET CONFIGURATION Request is received. This function then starts a control read transaction that sends the configuration, interface, endpoint, and HID descriptors to the host.

requestNotSupported – Unsupported or invalid descriptor requests will cause this firmware to STALL these transactions

Manufacturer String

A request for the manufacturer string will return the following string.

"Avago Reference Design Mouse"

Product String

A request for the product string will return the following string.

"ADNS-7550 Mouse"

Configuration String

A request for the configuration string will return the following string.

"HID-Compliant Mouse"

Endpoint 1 String

A request for the endpoint string will return the following string.

"Endpoint 1 Interrupt Pipe"

Notes:
1. The Manufacturer String should be changed to the name of your company.
2. The Product String should be changed to your product's name.
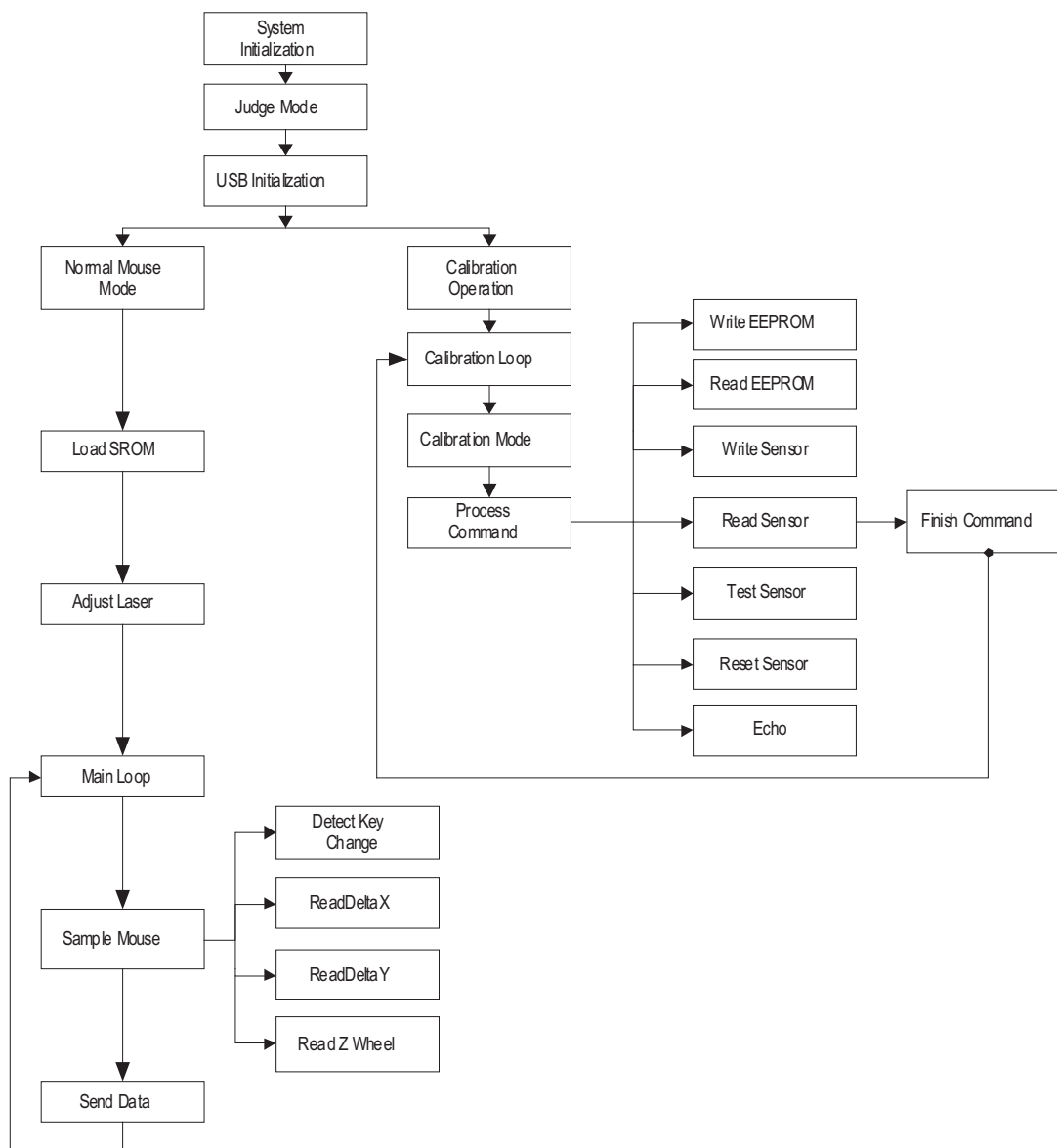
Figure 5. USB Operation Function Call Map

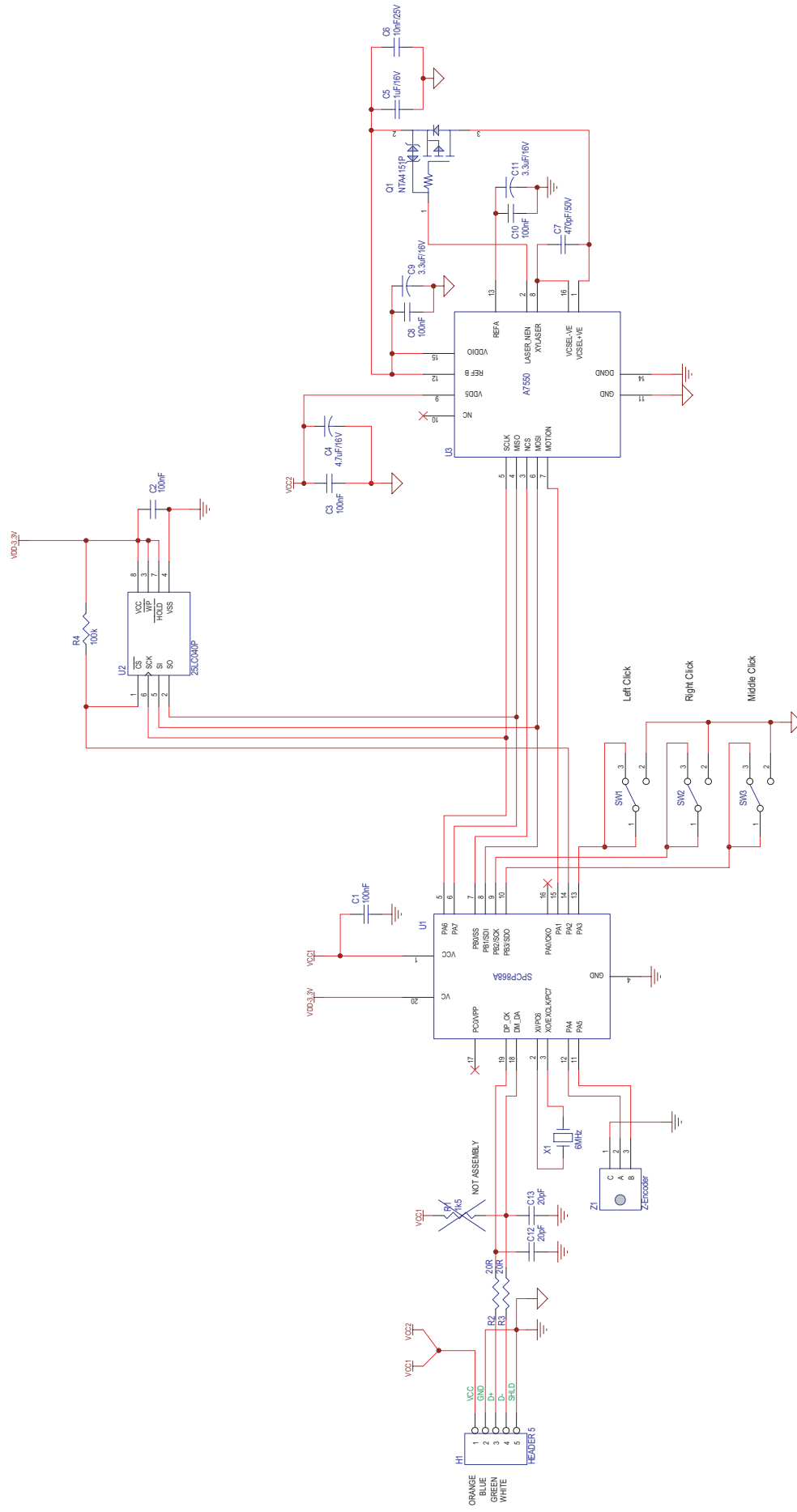# Appendix A: Schematic Design of Overall Circuit

Figure A1. Circuit diagram of SPCP868A microcontroller in the ADNK-7553-SP11 designer's kit corded laser mouse

Appendix B: Bill of Materials for ADNK-7553-SP11 Corded Laser Mouse Designer's Kit

Table B1. Bill of materials for ADNK-7553-SP11 corded laser mouse designer's kit main board

No	Description	Color / Value	Qty	Designator
01.	Resistor	20R 1%	1	R4.
02.	Resistor	100K 2%	1	R4.
03.	Capacitor Ceramic	20pF,50V	2	C12.C13
04.	Capacitor Ceramic	470pF,50V	1	C7.
05.	Capacitor Ceramic	10nF,25V	1	C6.
06.	Capacitor Ceramic	100nF,25V	5	C1.C2.C3.C8.C10.
07.	Capacitor Ceramic	1uF,16V	1	C5.
08.	Capacitor Ceramic	3.3uF,16V	2	C9.C11.
09.	Capacitor Ceramic	4.7uF,16V	1	C4.
10.	IC Mosfet	NTA4151PT1G	1	Q1.
11.	Switch - Micro Switch SPDT	3way	3	SW1.SW2.SW3.
12.	IC SMD	25LC040/SN	1	U2.
13.	IC DIP	SPCP868A	1	U1.
14.	IC Socket Double Row	20way	1	U1.
15.	Crystal	6MHz	1	X1
16.	Mouse Sensor	ADNS-7550	1	U3.
17.	Lens	ADNS-6150	1	-
18.	Z-Encoder	Black	1	Z1.
19.	Mouse Foot	-	-	-
20.	Big Casing	-	1	-
21.	Big Casing Roller	-	1	-
22.	Cable - USB Cable	-	1	-
23.	Big Casing Screw	-	1	-
24.	Big Casing White Cylinder	White		-
25.	Big Casing Mouse Button	-	-	-
26.	PCB Model : AFX A2733-M-01-B	-	1	-

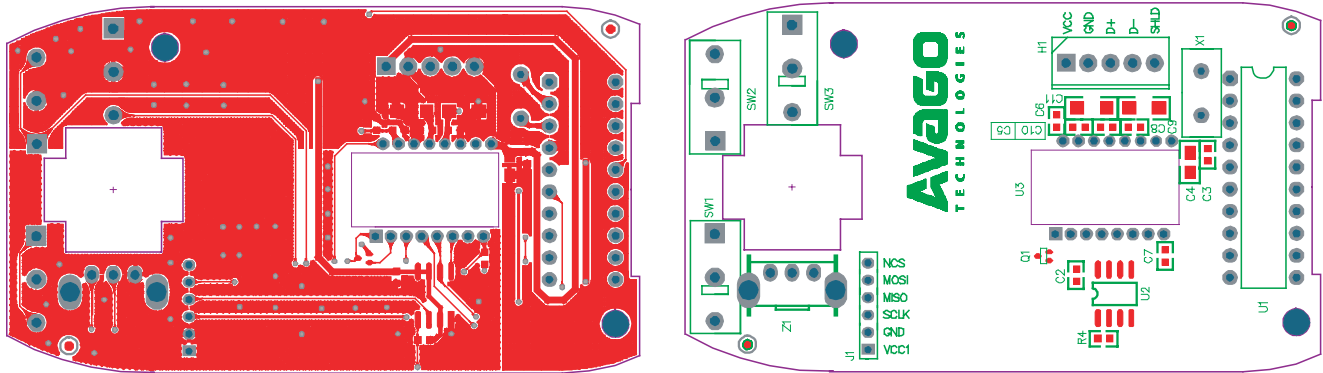## Appendix C: PCB Layout

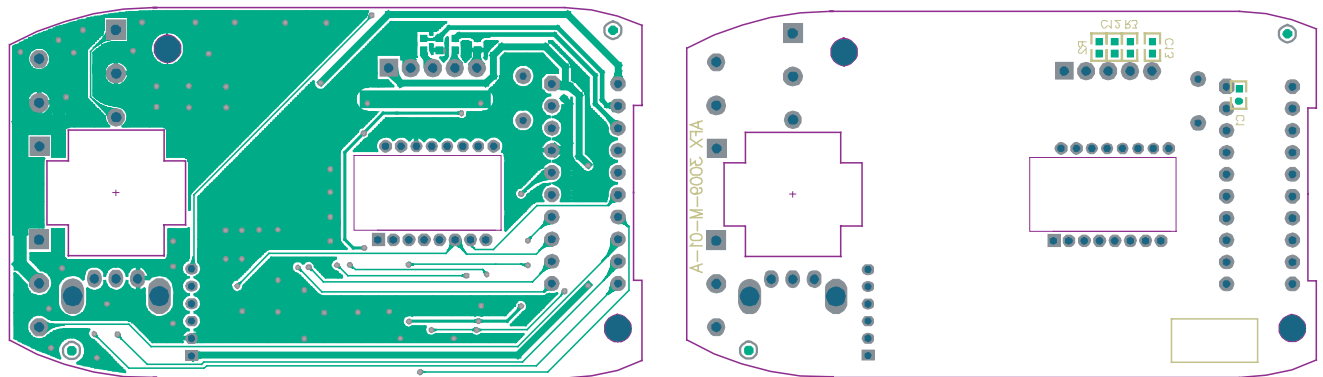**Figure C1. Top PCB layout and assembly drawing for the ADNK-7553-SP11 mouse**

Figure C1. Bottom PCB layout and assembly drawing for the ADNK-7553-SP11 mouse
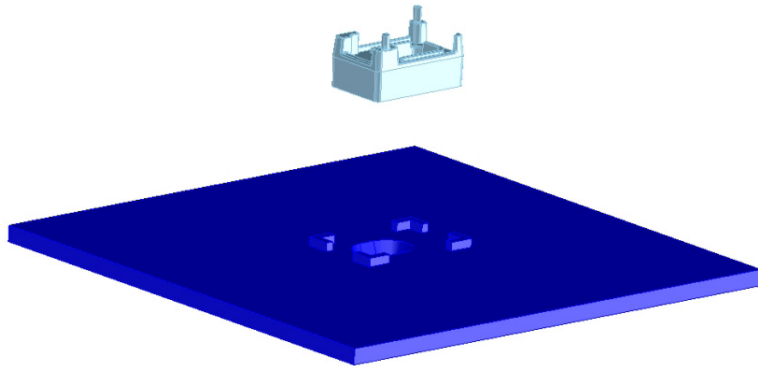
13

Appendix D: Base Plate Feature

Figure D1. Illustration of base plate mounting features for ADNS-6150 laser mouse small form factor (SFF) lens
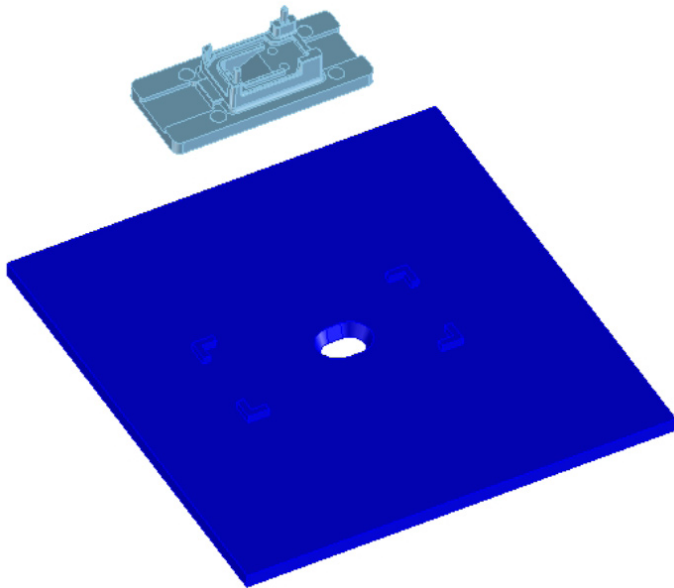

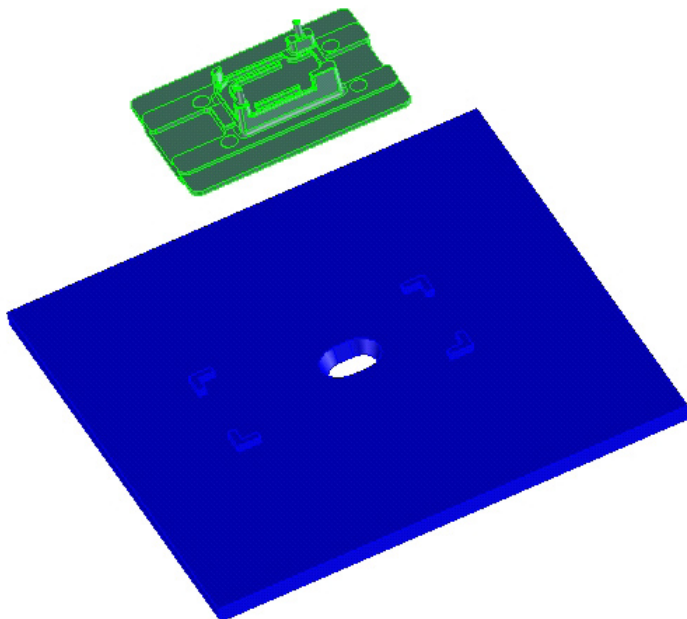**Figure D2. Illustration of base plate mounting features for ADNS-6160-001 laser mouse SFF trim lens**

Figure D3. Illustration of base plate mounting features for ADNS-6170-002 laser mouse wide trim lens

Appendix E: Sectional view of PCB assembly

Customer's PCB

Sensor

Customer Supplied
Base Plate

Lens

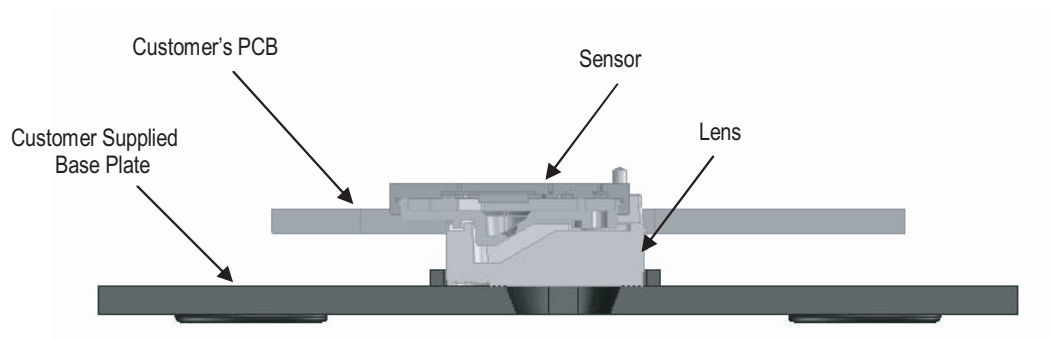**Figure E1. Sectional view of PCB assembly highlighting all optical mouse components (laser mouse sensor, ADNS-6150 lens, PCB and base plate).**
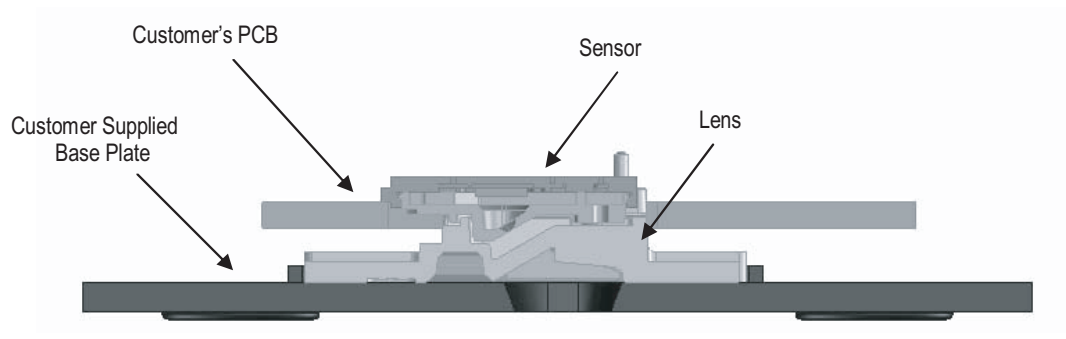
Customer's PCB

Sensor

Customer Supplied
Base Plate

Lens

Figure E2. Sectional view of PCB assembly highlighting all optical mouse components (laser mouse sensor, ADNS-6160-001 lens, PCB and base plate).
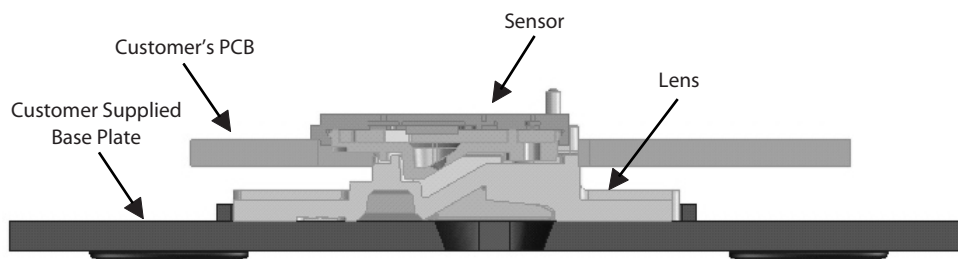
Sensor

Customer's PCB

Customer Supplied
Base Plate

Lens

Figure E3. Sectional view of PCB assembly highlighting all optical mouse components (laser mouse sensor, ADNS-6170-002 lens, PCB and base plate).

Appendix F: USB data reporting format

The USB report has two formats, depending on if boot or report protocol is enabled. The following format is the boot protocol and is understood by a USB aware BIOS.

	Bit 7							Bit 0
Byte 0	0	0	0	0	0	Middle	Right	Left
Byte 1	X	X	X	X	X	X	X	X
Byte 2	Y	Y	Y	Y	Y	Y	Y	Y

The following is the 16-bit USB report protocol format and allows the additional wheel movement information in the fourth byte. When the wheel is moved forward the fourth byte reports a 0x01, and when moved backward the fourth byte reports 0xFF. When the wheel is idle, then this byte is assigned 0x00.

	Bit 7							Bit 0
Byte 0	0	0	0	0	0	Middle	Right	Left
Byte 1	Vendor defined (X Low Byte for boot protocol, will be ignored by Windows)							
Byte 2	Vendor defined (Y Low Byte for boot protocol, will be ignored by Windows)							
Byte 3	Z	Z	Z	Z	Z	Z	Z	Z
Byte 4	X_L	X_L	X_L	X_L	X_L	X_L	X_L	X_L
Byte 5	X_H	X_H	X_H	X_H	X_H	X_H	X_H	X_H
Byte 6	Y_L	Y_L	Y_L	Y_L	Y_L	Y_L	Y_L	Y_L
Byte 7	Y_H	Y_H	Y_H	Y_H	Y_H	Y_H	Y_H	Y_H

Appendix G: Kit Components

The designer's kit contains the following components:

Part Number	Description	Name	Quantity
ADNK-7553-SP11 Mouse Set	Corded Laser Mouse	Reference Design Mouse Set	1
ADNS-7550	Laser Mouse Sensor	Sensor	5
ADNS-6150	Laser Mouse Small Form Factor (SFF) Lens Plate	Lens	5
ADNS-6160-001	Laser Mouse SFF Trim Lens Plate	Lens	5
ADNS-6170-002	Laser Mouse Wide Trim Lens Plate	Lens	5
ADNK-7553-SP11 CD-ROM	Includes Documentation and Support Files for ADNK-7553-SP11		1

Documentation
a. ADNS-7550 Data Sheet
b. ADNS-6150 Small Form Factor Lens Data Sheet
c. ADNS-6160 Trim Lens Data Sheet
d. ADNS-6170-002 Wide Trim Lens Data Sheet
e. Avago Technologies ADNS-7530 and ADNS-7550 LaserStream Sensor Mouse Eye Safety Calculation Application Note 5361
f. Sunplus IT SPCP868A Microcontroller Datasheet

Hardware Support Files
a. ADNK-7553-SP11 BOM List
b. ADNK-7553-SP11 Schematic
c. ADNK-7553-SP11 Gerber File
d. 3D Model IGES Files

Support File
a. Microcontroller Firmware – Sunplus IT SPCP868A Microcontroller

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

Avago
T E C H N O L O G I E S

Mouser Electronics

Authorized Distributor

Click to View Pricing, Inventory, Delivery & Lifecycle Information:

[Avago Technologies](#):
  [ADNK-7553-SP11](#)