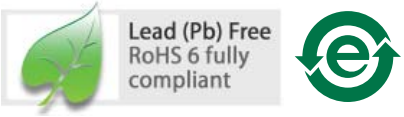
ADNK-6003-SP01
Optical Mouse Designer's Kit

AVAGO TECHNOLOGIES

Design Guide



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, SPCP825A USB microcontroller and the Avago Technologies ADNS-6000 optical sensor. The document starts with the basic operations of a computer mouse peripheral followed by an introduction to the Sunplus SPCP825A USB microcontroller and the Avago Technologies ADNS-6000 Optical Navigation Sensor. A schematic of the Sunplus SPCP825A USB microcontroller to the ADNS-6000 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 Sunplus SPCP825A data sheet is available from the Sunplus web site at www.sunplus.com . The ADNS-6000 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.

ADNS-6000 laser mouse set is the world's first laser-illuminated navigation system. With laser navigation technology, the mouse can operate on many surfaces that prove difficult for traditional LED-based optical navigation. Its high-performance architecture is capable of sensing high-speed mouse motion -- velocities up to 20 inches per second and accelerations up to 8g.

The ADNS-6000 sensor along with the ADNS-6120 lens, ADNS-6230-001 clip and ADNV-6340 laser diode form a complete and compact laser mouse tracking system. There are no moving parts, which means high reliability and less maintenance for the end user. In addition, precision optical alignment is not required, facilitating high volume assembly.

Optical Mouse Basics

The optical mouse measures changes in position by optically acquiring sequential surface images (frames), and mathematically determining the direction and magnitude of movement. The Z-wheel movement is done in the traditional method by decoding the quadrature signal generated by optical sensors. This design guide shows how to connect to and manage a standard configuration of mouse hardware, as well as handle the USB protocols. Each of these protocols provides a standard way of reporting mouse movement and button presses to the PC.

Introduction to the Sunplus, SPCP825A

The Sunplus, SPCP825A is a general purpose OTP USB microcontroller. It has dual USB speed, namely low and full speed. It can support PS/2 mode. The transceiver is fully controlled by the firmware. Moreover the USB SIE provides good flexibility for firmware to handle USB protocol. The built-in PLL allows CPU to work at 6MHz or 12MHz by using only one 6MHz crystal or resonator.

Serial Peripheral Interface (SPI)

The Sunplus SPCP825A 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 SPCP825A to communicate with external SPI compatible hardware, in this case the ADNS-6000.

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. The D - line is pulled up via a 1.3k ohm resistor connected to the VREG pin.

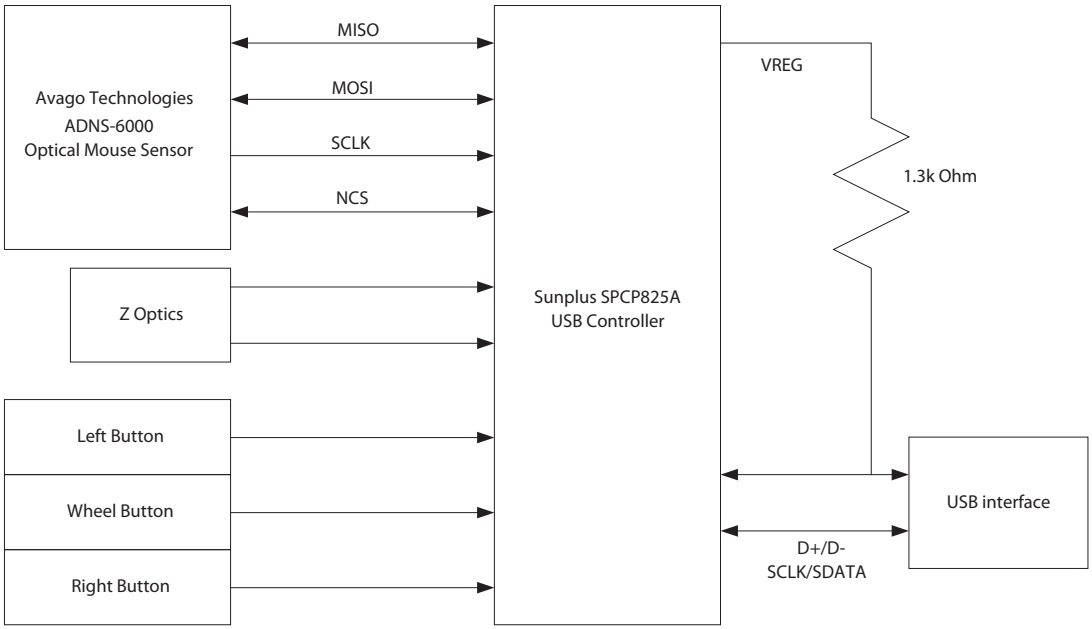


Figure 1. Sunplus SPCP825A - ADNS-6000 Optical Mouse Hardware Block Diagram

Firmware Configurable GPIO

The reference firmware is configured to use the GPIO pins as shown on the schematic in Appendix A. However, it may be more optimal to use a different I/O configuration to meet the mechanical constraints of PCB design. The reference firmware is designed to be easily configured to another set of pin connections. This is accomplished through changes in the I/O definitions at the beginning of the SPCP825A_A6000.asm listing. The following statements are the pin definitions as they exist today. The firmware will use these definitions to read and configure the GPIO pins, without any other modifications.

Communications between the Sunplus SPCP825A and the ADNS-6000 are done through the integrated SPI interface. The serial port cannot be activated while the chip is in power down mode (NPD low) or reset (RESET high). When the SPI is enabled thru PB0 (NCS), the PB2 (SCLK), PB1 (MISO) and PB3 (MOSI) GPIO pins serve special functions to enable the SPI interface to talk with external hardware. During normal operation, the Sunplus SPCP825A SPI is always configured as a Master to output the serial clock on PB7. Therefore, the USB microcontroller always initiates communication. Data sent by the ADNS-6000 optical sensor is received on the PB1 (MISO), and data is shifted out to the ADNS-6000 through the PB3 (MOSI). See the schematic in Appendix A. When writing to the ADNS-6000, the microcontroller drive both the SCLK and the MOSI lines. When reading from the ADNS-6000, the microcontroller drives both the SCLK and MOSI lines initially. After t_{SRAD} delay, the ADNS-6000 will drive

Optical Sensor

Avago Technologies ADNS-6000 optical sensor is used in this reference design as the primary navigation engine. This Optical Navigation Technology contains an Image Acquisition System, a Digital Signal Processor, a two channel quadrature output, and a four-wire serial port. The Sunplus SPCP825A periodically reads the ADNS-6000's Delta_X and Delta_Y registers to obtain any horizontal and vertical motion information happening as a result of the mouse being moved. The output of the ADNS-6000 optical sensor is 4-wire serial port.

This motion information will be reported to the PC to update the position of the cursor. The advantages of using ADNS-6000 optical sensor are the best tracking accuracy, flexibility of programming the optical sensor via the SPI port, and the automatic frame rate feature (1000fps to 6400fps). Besides, ADNS-6000 optical sensor performs excellent tracking on difficult surfaces which conventional Led based technology is unable to track such as glossy and smooth surfaces. In addition, Burst mode is another special serial port operation mode that may be used to reduce the serial transaction time for three predefined operations: motion read and SROM download and frame capture. 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-6000 will respond with the contents of the Motion, Delta_X, Delta_Y, SQUAL, Shutter_Upper, Shutter_Lower and Maximum_Pixel registers in that order. SROM download uses Burst Mode to load the Avago Technologies-supplied firmware file contents into the ADNS-6000. The firmware file is an ASCII text file with each 2-character byte on a single line. Frame Capture is a fast way to download a full array of pixel values from a single frame.

To learn more about sensor's technical information, please visit the Avago Technologies web site at http://www.avagotech.com

Mouse Optics

The motion of Z-wheel is detected using the traditional method by decoding the quadrature signal generated by optical sensors. Two phototransistors are connected in a source-follower configuration. An infrared LED shines, causing the phototransistors to turn on. In between the phototransistors and LED is a pinwheel that turns on the mouse ball rollers. The fan of this pinwheel is mechanically designed to block the infrared light such that the phototransistors are turned on and off in a quadrature output pattern. Every change in the phototransistor outputs represents a count of mouse movement. Comparing the last state of the optics to the current state derives direction information. As shown in Figure 2 below, traveling along the quadrature signal to the right produces a unique set of state transitions, and traveling to the left produces another set of unique state transitions. In this reference design, only the motion at the Z-wheel is detected using this method.

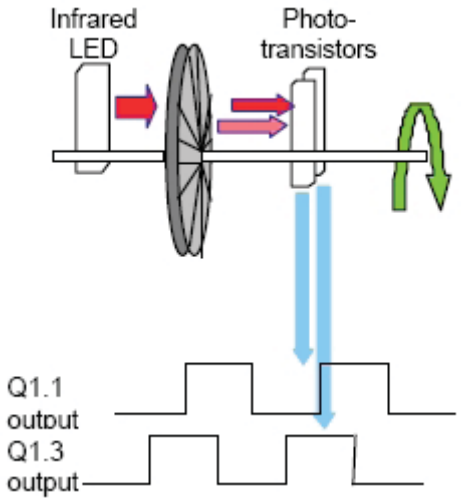


Figure 2. Optics Quadrature Signal Generation

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 SPCP825A 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 3 below.

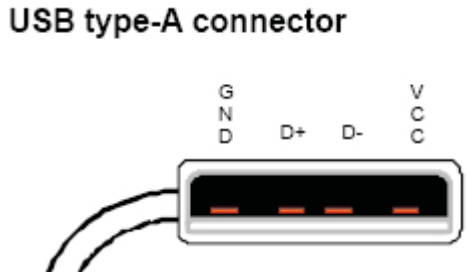


Figure 3. USB peripheral connectors

Some details on ADNK-6003-SP01

The ADNK-6003-SP01 reference design mouse unit allows users to evaluate the performance of the Optical Tracking Engine (sensor, lens, LASER assembly clip, LASER) over a USB Sunplus SPCP825A USB Controller. This kit also enables users to understand the recommended mechanical assembly. (See Appendix C and D)

System Requirements

PCs using Windows® 95/ Windows® 98/ Windows® NT/ Windows® 2000 with standard 3-button USB mouse driver loaded.

Operating (For USB 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-6003-SP01 Unit

The ADNK-6003-SP01 comprises of the plastic mouse casing, printed circuit board (PCB), lens, buttons, and USB cable. (See Figure 4.) Unscrewing the one screw located at the base of the unit can open the ADNK-6003-SP01 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.

*or ADNS-6130-001 for trim lens

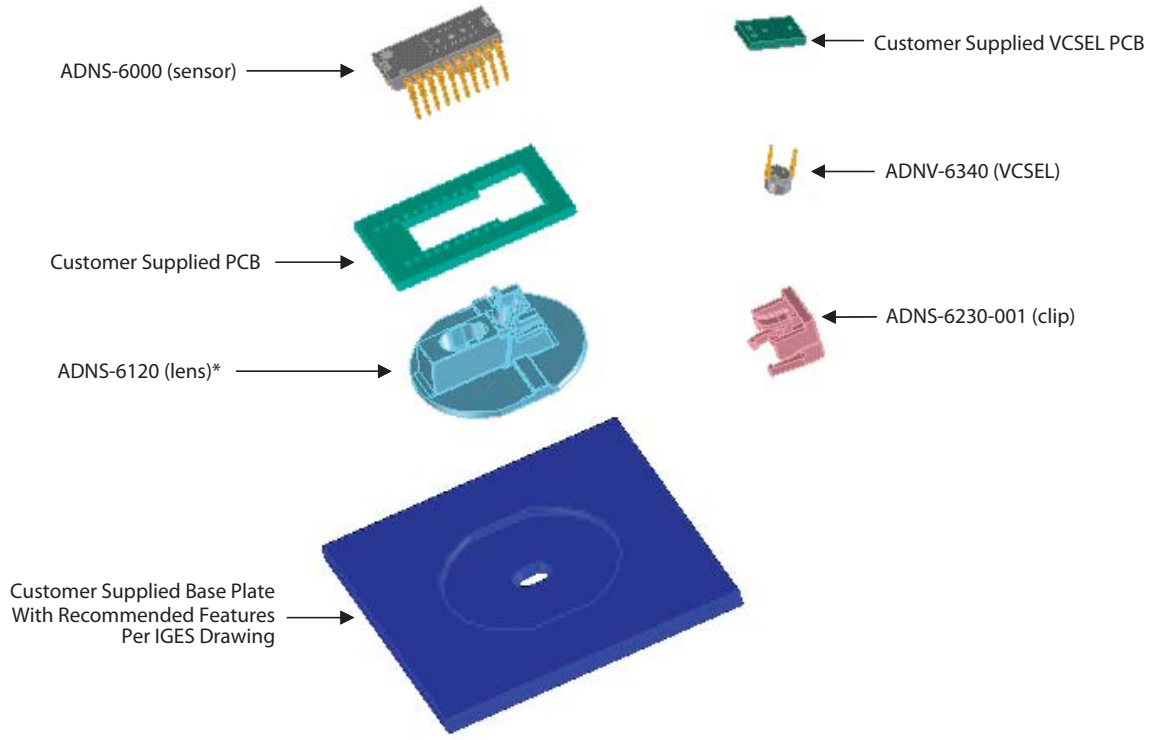


Figure 4. Exploded view drawing of optical tracking engine with ADNS-6000 Optical Mouse Sensor

While reassembling the components, please make sure that the Z height (Distance from lens reference plane to surface) is valid. Refer to Figure 5.

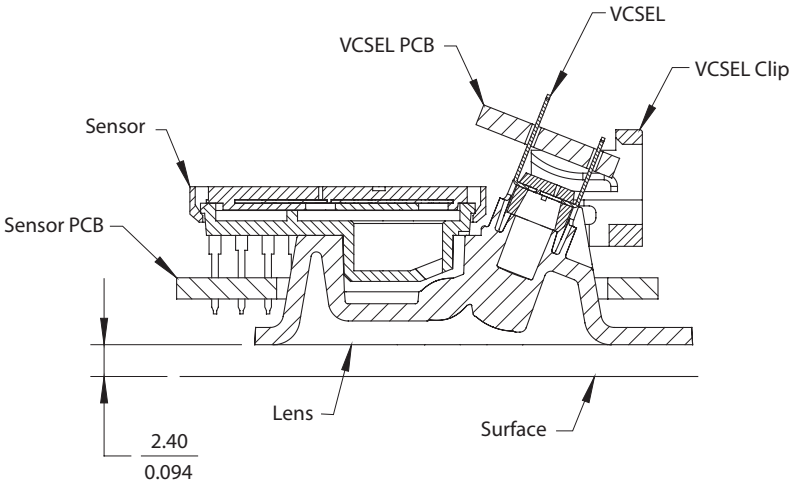


Figure 5. Distance from lens reference plane to surface

Enabling the SROM

The ADNS-6000 must operate from the externally loaded programming. This architecture enables immediate adoption of new features and improved performance algorithms. The external program is supplied by Avago Technologies as a file which may be burned into a programmable device. A microcontroller with sufficient memory may be used. On power-up and reset, the ADNS-6000 program is downloaded into volatile memory using the burst-mode procedure described in the Synchronous Serial Port section. The program size is 1986 x 8 bits.

For more information, please refer to the ADNS-6000 datasheet.

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.
- UL flammability level UL94 V-0.
- Provides sufficient ESD creepage/clearance distance to avoid discharge up to 15kV when assembled into a mouse according to usage instruction above.

Eye Safety

The ADNS-6000 and the associated components in the schematic of Figure A1 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 ADNS-6000 generates the drive current for the laser diode (ADNV-6340). In order to stay below the Class 1 power requirements, resistor Rbin must be set at least as high as the value in the bin table, based on the bin number of the laser diode and LP_CFG0 and LP_CFG1 must be programmed to appropriate values. Avago Technology recommends using the exact Rbin value specified in the bin table to ensure sufficient laser power for navigation. The system comprised of the ADNS-6000 and ADNV-6340 is designed to maintain the output beam power within Class 1 requirements over component manufacturing tolerances and the recommended temperature range when adjusted per the procedure below and when implemented as shown in the recommended application circuit of Figure A1. For more information, please refer to Eye Safety Application Note 5088.

LASER Power Adjustment Procedure

1. The ambient temperature should be 25 ° C ± 5 ° C.
2. Set VDD3 to its permanent value.
3. Ensure that the laser drive is at 100% duty cycle by setting bit 6 of register 0x0A to 0.
4. Program the LP_CFG0 and LP_CFG1 registers to achieve an output power as close to 506uW as possible without exceeding it.

Good engineering practices 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 VDD3 value is no greater than 50mV above its value at the time of adjustment.
4. No allowance for optical power meter accuracy is assumed.

Kit Component Summary

Below is the summary of the components contained in the ADNK-6003-SP01 Designer's Kit.

Sensor

The sensor technical information is contained in the ADNS-6000 Data Sheet.

Lens

The lens technical information is contained in the ADNS-6120 Data Sheet. The flange on the standard ADNS-6120 lens is for ESD protection.

LASER Assembly Clip

The information on the assembly clip is contained in the ADNS-6230-001 Data Sheet.

LASER

The LASER technical information is contained in the ADNV-6340 Data Sheet and Application Note AN-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-6003-SP01 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 Avago Technologies assembly language. The following files are required to compile the mouse.

SPCP825A _A6000.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_6000.asm – Routine to access ADNS-6000 sensor register.

spi.asm – Routine to access ADNS-6000 sensor register and EEPROM during calibration mode.

SPCP825A.inc – The SPCP825A I/O registers definition.

adns6000_srom_25.inc – ADNS-6000 SROM firmware.

ADNS6000.inc – ADNS-6000 interface constants.

delay.inc – SPCP825A delay loop subroutine.

decode_setup.inc – USB descriptor and request constants.

DET_Z.inc – SPCP825A Z-axis event handler.

DET_KEY.inc – SPCP825A button event handler.

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

The pull-up resistor on D– notifies the hub that a low speed (1.5 Mbps) device has just been connected. The host recognizes the presence of a new USB device and 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 (2). 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 SPCP825A _A6000.asm.

IO_initial – This function is use to set the Sunplus microcontroller 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-6000 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.

SPI_init – This function involves the initialization of SPI.

Read_LP_CFG_REG – This function reads the calibrated laser power configuration register from EEPROM.

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 phototransistors. The X and Y directions of the movement are obtained by calling the ReadDeltaX and ReadDeltaY routines. 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.

ReadDeltaX – Reads the ADNS-6000 Delta_X register for the X movement. Calls the ReadSPI routine to enable the SPI interface and perform reading operations through the two wire serial interface. Any new X motion information is added to the [xCount] variable.

ReadDeltaY – Reads the ADNS-6000 Delta_Y register for the Y movement. Calls the ReadSPI routine to enable the SPI interface and perform reading operations through the two wire serial interface. Any new Y motion information is added to the [yCount] variable.

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_adns6000 – This function is responsible for resetting the hardware, loading the SROM (Shadow ROM) firmware into the ADNS-6000 optical sensor. The ID from the device and program are compared. If the ID is not the same, then the program is trapped in the dead loop, i.e. the device is unusable.

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

judge_mode – This function is used to check for normal mouse or calibration mode.

detect_key_change – This function is used to detect left, right or middle button changes.

ReadMotionReg – Reads the ADNS-6000 Motion register. The data returned from this register will be used to determine if any motion has occurred or if any fault condition exists.

AdjustLaser – the AdjustLaser is used to read the value from the EEPROM and then store them into the ADNS-6000 sensor. The mouse will do calibration when the left button is pressed when the mouse is first plugged into the USB port. However if the mouse left button is not pressed, the value of calibration is already stored in the EEPROM i.e. there is no need of calibration.

Calibration operation - The calibration mode is different from the AdjustLaser as the calibration mode happens when the left button is pressed whereas the AdjustLaser is used to read the EEPROM values and put them into the ADNS-6000 sensor. This means in the Normal mouse mode, the AdjustLaser is done even in the normal mouse mode. Actually when the mouse enters the calibration mode, it cannot come back to the normal mouse mode without unplugging the USB plug.

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.

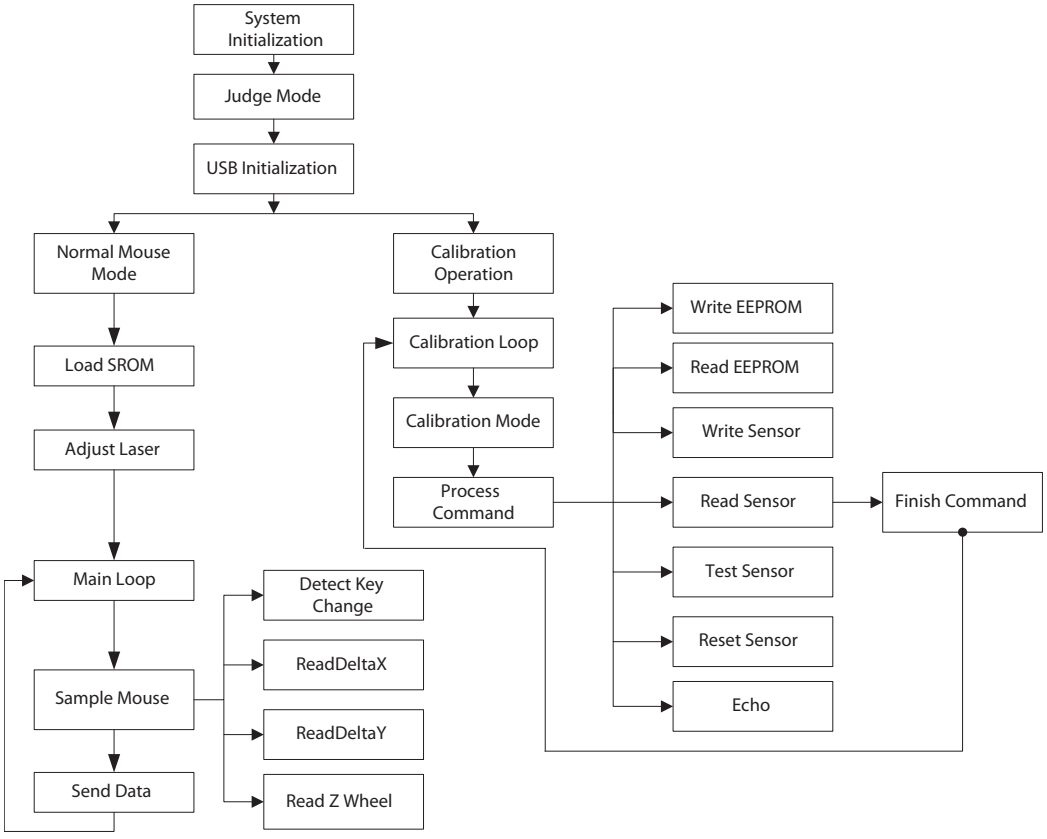


Figure 6. USB Operation Function Call Map

Manufacturer String*1

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

"Avago Technologies Reference Design Mouse"

Product String*2

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

"ADNS-6000 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.

Appendix A: Schematic Diagram of the Overall Circuit

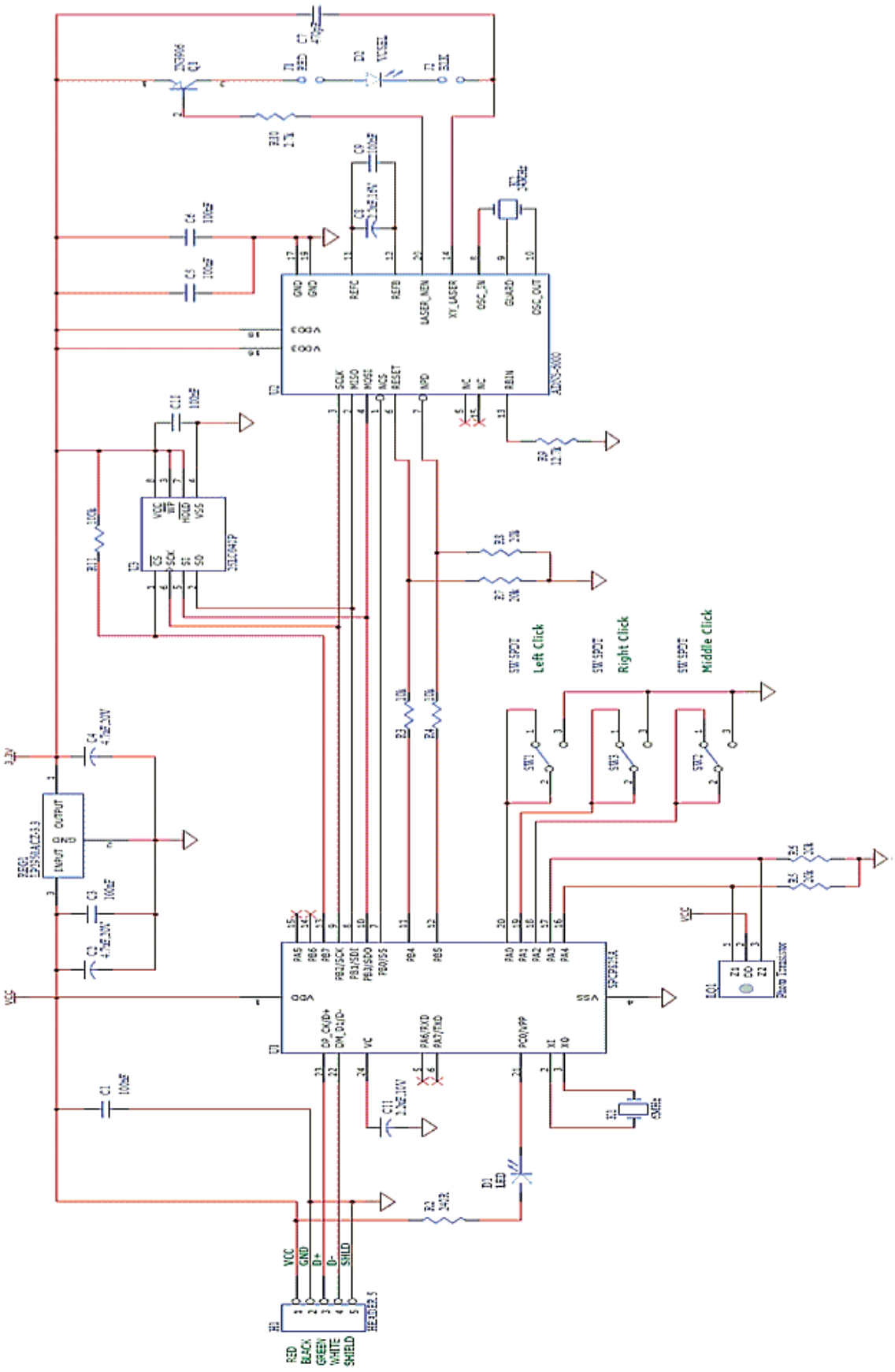


Figure A1. Circuit-level block diagram for ADNK-6003-SP01 designer's kit optical mouse using the Avago Technologies ADNS-6000 optical mouse sensor and Sunplus SPCP825A.

Appendix B: Bill of Materials for components shown on the circuit.

Comment	Footprint	Quantity
Cer. Cap 0.1uF (104)	0805	6
Cer. Cap 2.2uF, 16V	CASE A	2
Cer. Cap 4.7pF, 20V	CASE A	2
Cer. Cap 470pF	CASE A	1
Chip RES. 10K 1% 0.125W	0805	2
Chip RES. 12.7K 1% 0.125W	0805	1
Chip RES 2K7 1% 0.125W	0805	1
Chip RES 20K 1% 0.125W	0805	4
Chip RES 100K 1% 0.125W	0805	1
Chip RES 240R 1% 0.125W	0805	1
Photo Transistor	DIP	1
2N3906	TO-92	1
Crystal, 6MHz	DIP	1
Crystal, 24MHz	DIP	1
SPCP825A	DIP-24	1
ADNS-6000	DIP-20	1
25LC040/P	SO-8	1
HEADER 5	HEADLOCK5P	1
LED	DIP	1
LP2950ACZ-3.3	TO-92	1
Jumper, RED	DIP	1
Jumper, BLK	DIP	1
VCSEL	DIP	1
SW SPDT	DIP	3

Appendix C: PCB Layout

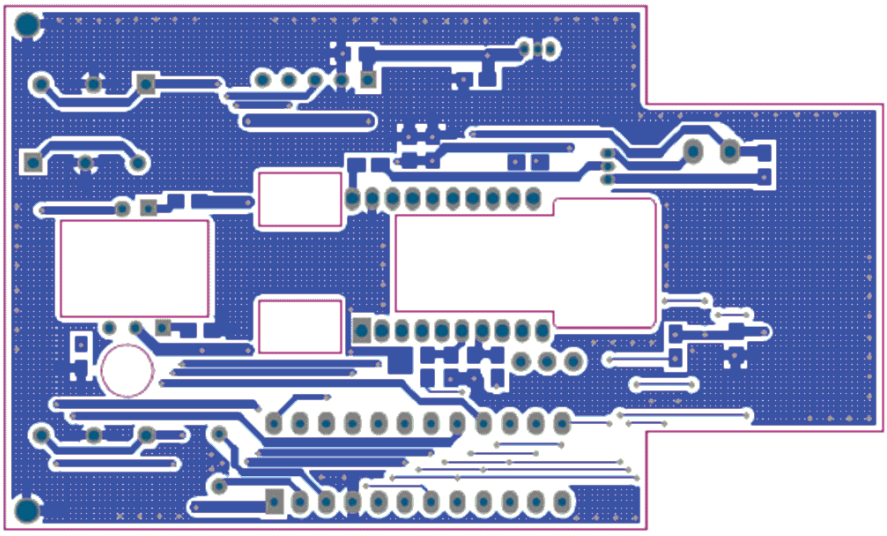


Figure C1: PCB Schematic (Bottom Layer)

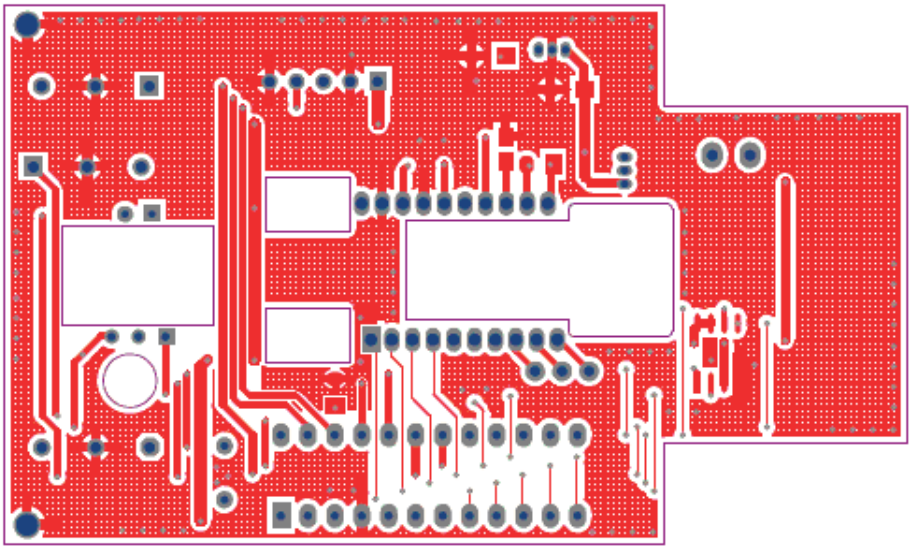


Figure C2: PCB Schematic (Top Layer)

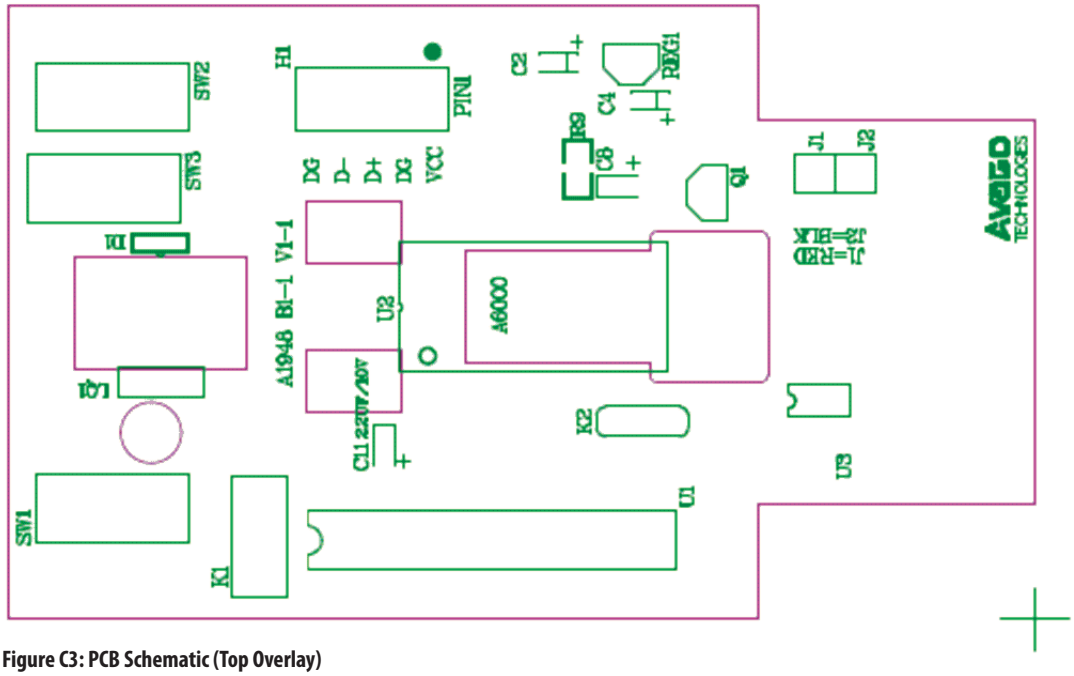


Figure C3: PCB Schematic (Top Overlay)

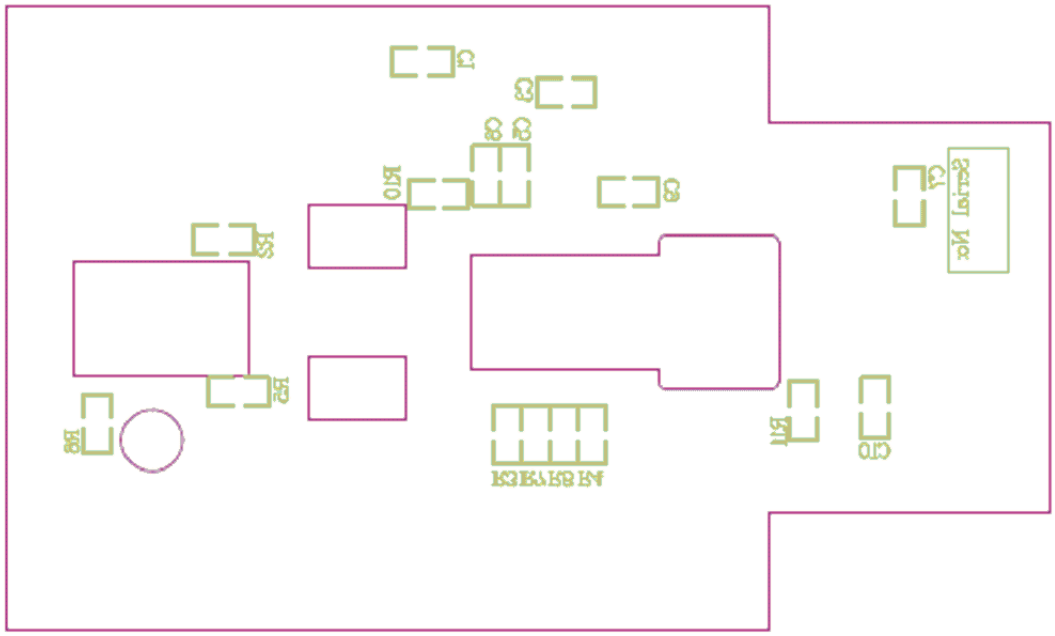


Figure C4 : PCB Schematic (Bottom Overlay)

Appendix D: Base Plate Feature

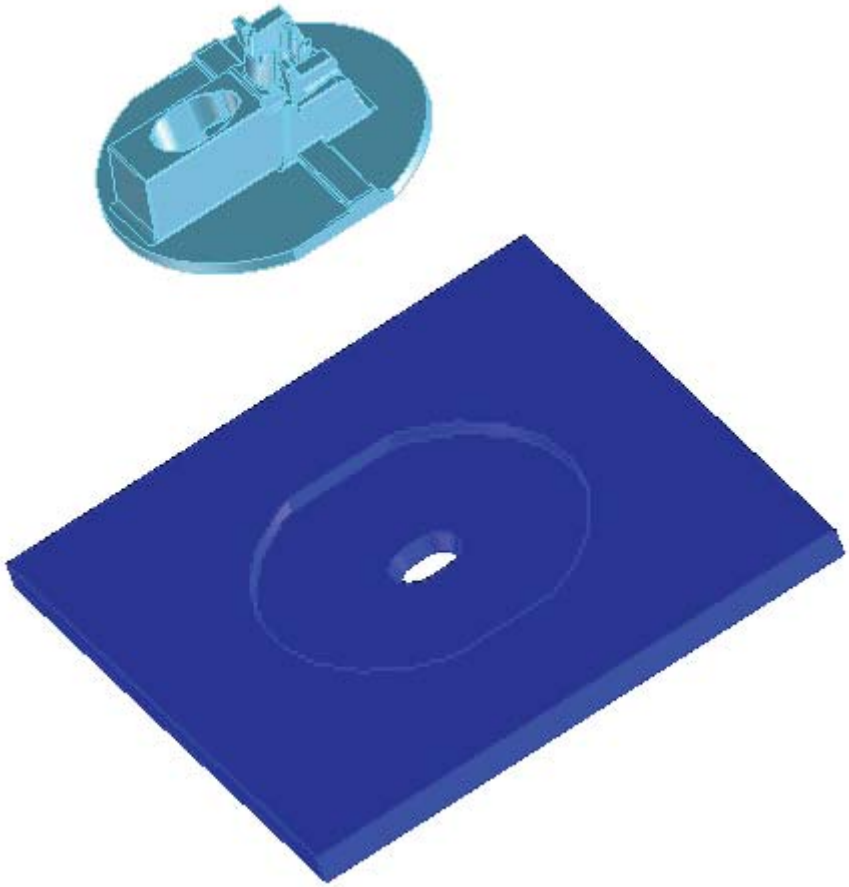


Figure D1: Overall view of base plate

Appendix E: 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 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	X	X	X	X	X	X	X	X
Byte 2	Y	Y	Y	Y	Y	Y	Y	Y
Byte 3	Z	Z	Z	Z	Z	Z	Z	Z

Appendix F: Kit Components

The designer's kit contains components as follows:

Part Number	Description	Name	Quantity
ADNS-6000	High Performance Laser Mouse Sensor	Sensor	5
ADNS-6120	Laser Mouse Round Lens Plate	Lens	5
ADNS-6130-001	Laser Mouse Trim Lens Plate	Lens	5
ADNS-6230-001	Laser Mouse VCSEL Assembly Clip	VCSEL Clip	5
ADNV-6340	Single Mode Vertical Cavity Surface Emitting LASER (VCSEL)	VCSEL	5
ADNK-6003-SP01 CD	Includes Documentation and Support Files for ADNK-6003-SP01 Documentation a. ADNS-6000 Data Sheet b. ADNS-6120 Round Lens Data Sheet c. ADNS-6130-001 Trim Lens Data Sheet d. ADNS-6230-001 VCSEL Assembly Clip Data Sheet e. ADNV-6340 VCSEL Data Sheet f. Avago Technologies ADNS-6000, ADNS-6010, ADNS-6090 and ADNS-7010 Laser Mouse Eye Safety Calculation Application Note 5088 Hardware Support Files a. ADNK-6003-SP01 BOM List b. ADNK-6003-SP01 Schematic c. 3D Model IGES FIles d. Gerber File Software Support Files a. Microcontroller Firmware		1

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

AV02-1383EN - September 9, 2008