ADNK-6710 Designer's Kit
Automated Laser Power Calibration (USB Direct)

User Guide

1. Introduction

This document explains the components of Automated Laser Power Calibration (USB Direct Version) System and demonstrates how to implement laser power calibration on Avago Technologies ADNK-6003 Designer's Kit.

2. Automated Laser Power Calibration System

The laser power calibration system consists of the following components:

a) Laser Power Calibration Application
b) Optical Power Meter & Optical Power Sensor
c) Mouse target board
d) Mechanical Jig

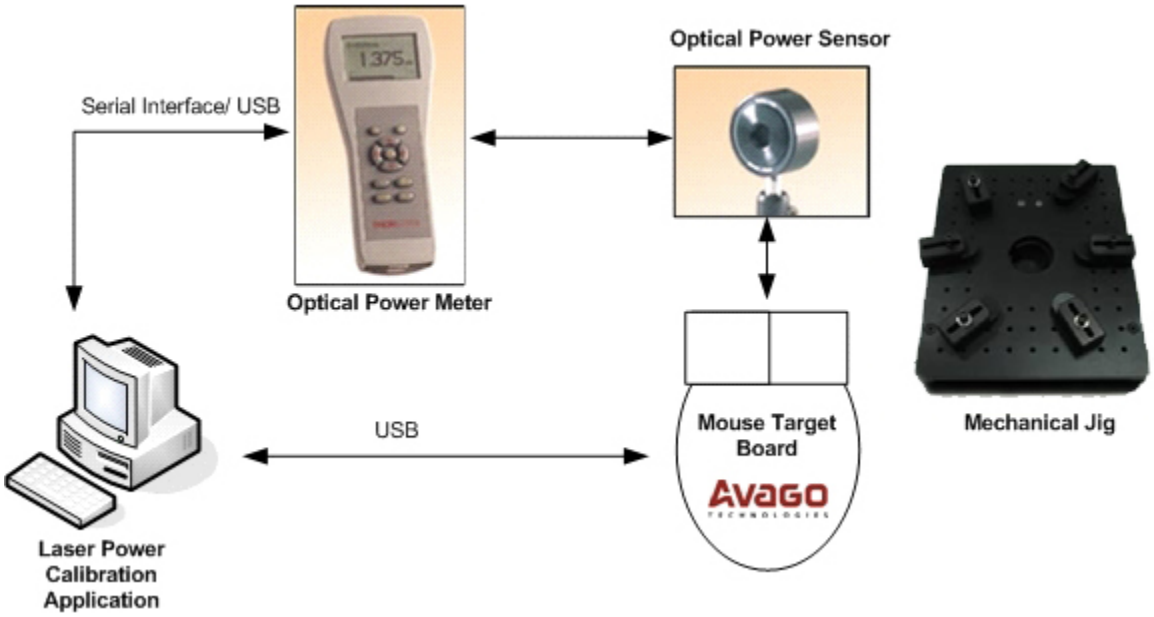


Figure 1.0 Automated Laser Power Calibration System Components

Laser Power Calibration Application

User-friendly PC application to let end-user control and perform the laser power calibration process. This application provides options for user to define the setting and pass-fail criteria for the calibration process. During the calibration process, it will query the power meter to get the meter reading and compared with the predefined setting. The laser power calibration application direct program the mouse hardware through Universal Serial Bus (USB).

Optical Power Meter & Sensor

Optical power meter to measure laser output power reading. Currently the laser output power calibration application supports RS232 interface for Thorlabs PM100 power meter and USB interface for NewPort 1930 power meter. Additional power meter can be supported by adding new driver to the laser power calibration application.

Mouse target board

Mouse hardware designed using Avago Technologies Laser Sensor technology. Designed with an optional Microchip 25LC040(4Kbit/512B) EEPROM to store calibrated laser output power value. For mouse designed without EEPROM, the laser output power calibration application can still calibrate the laser, then end-user can retrieve the calibrated value from the application and program it manually using external programmer.

Mechanical Jig

Provide mounting for optical sensor and mouse mounting stage. The mechanical jig contains both X and Y displacement lever to adjust the mouse laser output to the optical sensor input.

3. Safety & Regulations

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 instructions above.

Eye Safety

The ADNS-6000 and the associated components in the schematic of Figure 6.0 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 ADNS-6000 datasheet section.

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 Technologies 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 6.0. For more information, please refer to Eye Safety Application Note 5088.

LASER Power Adjustment Procedure

1. The ambient temperature should be 25C +/- 5C.
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.

4. PC Host Application Implementation

The Human Interface Device (HID) class specification allows designers to create USB-based devices and Windows application without the need for custom driver development. Mice is the most common example of HID device. The ALPC USB Direct version is extending this idea to add laser calibration functionality thru the USB interface.

The ALPC host application will use standard Windows's built-in HID driver to communicate with mice device in calibration mode. However, Windows XP and Windows 2000 have exclusive read/write access to HIDs that are configured as a system keyboard or mice, the "Access denied" exception will be prompted when attempting to access these HID devices.

To allow communication between host application and mice firmware, the mice firmware will have two set of USB Descriptors profile, the HID-compliant device descriptor and HID-compliant mouse descriptor. The mice device will be detected as HID-compliant device in laser calibration mode when HID-compliant device profile is use. At this moment, the mice will not have pointer navigation function but allow read/write access interface.

Device Endpoints

In USB-based systems, all data travels to or from device endpoints. The USB specification requires that all devices have a control endpoint. The host uses this endpoint to retrieve information about the device through data packets called Descriptors. Many USB devices also support additional endpoints that transfer data to and from the host. IN endpoints transfer data from the device to the host, while OUT endpoints transfer data from the host to the device. Endpoint 0 is used for input data from device while endpoint 1 is used for output data to device.

Visual Basic

This is the code that define USB endpoints (mdlUSBHid.bas) :

Public Const HidP_Input = 0

Public Const HidP_Output = 1

Laser Output Power Calibration Flow

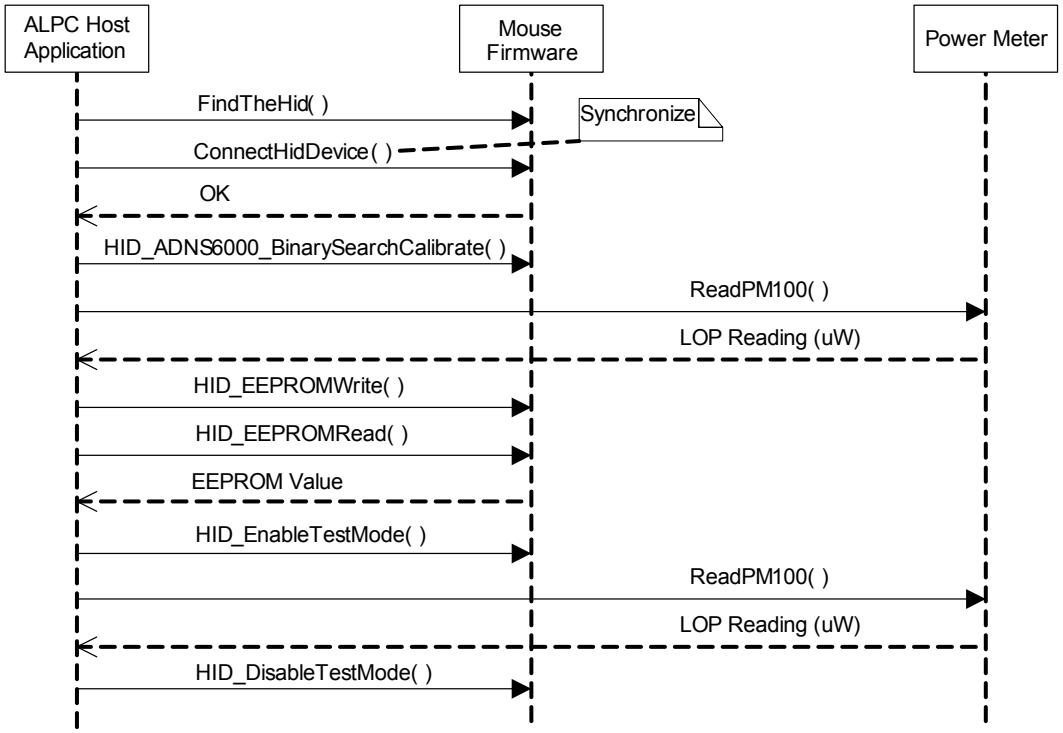


Figure 2.0 PC host application laser output power calibration flow sequence diagram

The Visual Basic mdlUSBHid (mdlHid.bas) module contains low-level APIs to access HID-compliant device.

- Public Function FindTheHid() As Boolean - Makes a series of API calls to locate the desired HID-class device. Returns True if the device is detected, False if not detected. This function compare specific Vendor ID (Vid) and Product ID (Pid) for searching the matched value in mice device.

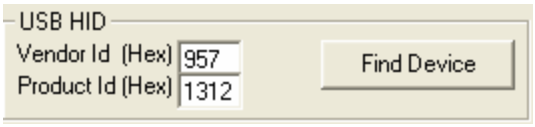
  Visual Basic

  This is the code that define the Vendor ID and Product ID:

  Dim gHidVendorID As Integer

  Dim gHidProductID As Integer

  Alternatively, the GUI contains text fields for entering the desired values.



- Public Sub ReadReport()- Return report in ReadBuffer.
- Public Sub WriteReport() – Send a report to device.
- Public Sub HidClose()-Close the open handles to the hid device.

  The mdlHidCommonAPI (mdlHidCommonAPI.bas) module contains APIs to read/write ADNS-6000 sensor register and EEPROM content.
- Public Function HID_EEPROMWrite(Address As Long, value As Byte) As Boolean – Write byte value to EEPROM.
- Public Function HID_EEPROMRead(Address As Long, ByRef DataByte As Byte) As Boolean – Read byte value from EEPROM.
- Public Function HID_RegisterWrite(Address As Long, value As Byte) As Boolean – Write byte value to sensor register.
- Public Function HID_RegisterRead(Address As Long, ByRef DataByte As Byte) As Boolean – Read byte value from sensor register.
- Public Function HID_EnableTestMode() As Boolean ¬– Enable sensor laser calibration mode.
- Public Function HID_DisableTestMode() As Boolean – Disable sensor laser calibration mode.

The command protocol between PC host application and mice firmware are defined in the next chapter.

Command Protocol

Write to CFGX register

From Host to device

[Command Code][Register Address][Register Value]

[Command Code] is a hexadecimal command code with value = 0x10.
[Register Address] register address to write to, value from 0 to FF, 256 register.
[Register Value] is the value to write to the mentioned register.

Eg. Write value of 0x41 to register address 0x07

0x100x070x41

Reply from Device to Host

0x100x010x00 if write to the register success

0x100x000x00 if write to the register fail

Read from CFGX register

From Host to device

[Command Code][Register Address][Dummy Value]

where

[Command Code] is a hexadecimal command code with value = 0x11.
[Register Address] register address to read from, value from 0 to FF, 256 register
[Dummy Value] is a dummy hex value, e.g 0x00

Eg. EEPROM Read value from register address 0x0FF

0x110xFF0x00

Reply from Device to Host

0x110x010xVV if read from the register success
where 0xVV is the value read

0x110x000xYY if read from the register fail
where 0xYY is a dummy hex value, e.g 0x00

Write to EEPROM

From Host to device

[Command Code][Register Address][Register Value]

where

[Command Code] is a hexadecimal command code with value = 0x20
[Register Address] EEPROM address to write to, value from 0 to FF, 256 bytes only
[Register Value] is the value to write to the mentioned register.

Eg. EEPROM Write value of 0x41 to EEPROM address 0x0FF

0x200xFF0x41

Reply from Device to Host

0x200x410x00 if write to the register success

0x200x000x00 if write to the register fail

Read from EEPROM

From Host to device

[Command Code][Register Address][Dummy Value]

Where

[Command Code] is a hexadecimal command code with value = 0x21
[Register Address] is the EEPROM address to read from, value from 0 to FF, 256 bytes only
[Dummy Value] is a dummy hex value, e.g 0x00

Eg. EEPROM Read value from EEPROM address 0x0FF

0x210xFF0x00

Reply from Device to Host

0x210x010xVV if read from the EEPROM success
where 0xVV is the value read

0x210x000xYY if read from the EEPROM fail
where 0xYY is a dummy hex value, e.g 0x00

Test Mouse LOP

From Host to device

[Command Code][Dummy Value][Dummy Value]

where

[Command Value] is a hexadecimal command code,
if 0x50 = Enabled test mouse LOP;
if 0x51 = Disabled test mouse LOP
and [Dummy Value] = Dummy hex value, e.g 0x00

Eg. Request Mouse to perform self LOP test

0x500x000xYY

Reply from Device to Host

0x500x010x00 if enable Test Mouse LOP success
0x500x000x00 if enable Test Mouse LOP fail

0x510x010x00 if disable Test Mouse LOP success
0x510x000x00 if disable Test Mouse LOP fail

Echo Device

From Host to device

[Command Code][Dummy Value][Dummy Value]

where

[Command Code] is a hexadecimal command code with value = 0x60
[Dummy Value] = Any dummy hex value, e.g 0x88

Eg. Request Mouse to perform echo test

0x600x880x99

Reply from Device to Host

0x600x088x99 if echo success
0x600x000x00 if echo fail

5. Laser Optical Mouse Firmware MCU Implementation

To implement USB Direct, two set of USB descriptors is required in the mouse firmware. Mouse will enter calibration mode if LEFT button is pressed and plug into specific PC USB port subsequently. In calibration mode, the mouse cursor will not move.

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

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 ANDS-6000 sensor register.

spi.asm - Routine to access ANDS-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 - ADNS6000 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.

Firmware Mode Selection Routine

At power up, the firmware examines the host interface and automatically determines if the mouse is plugged into a USB host connection. After the interface type has been determined, the host firmware configures itself to operate on the detected interface.

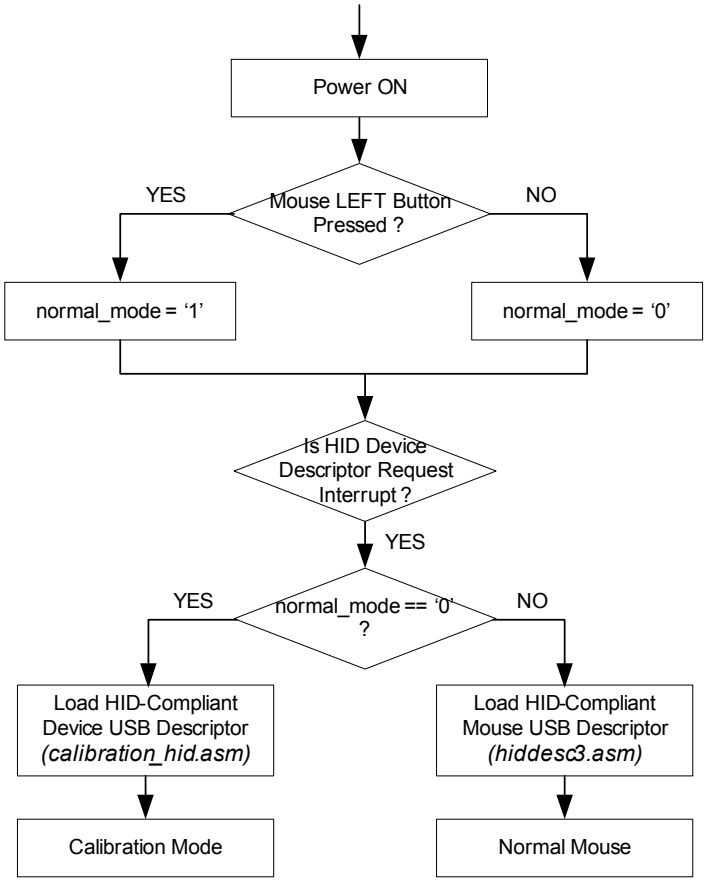


Figure 3.0 Firmware Execution Flow to Determine Mouse Mode

poweron – This routine first performs software reset function and configures input/output ports and then enter the enter USB mode.

judge_mode – This routine check LEFT button belonging port, if the port input is LOW, then set the normal_mode variable to '0'.

hard_reset – This routine resets the serial interface and the ADND-6000 internal registers by generating a pulse on the RESET pin.

Load_SROM – Called in calibration_operation after the initialization of the SPI interface. This routine is used to load the SROM (Shadow ROM) firmware into the ADNS-6000 optical sensor. It should be called after hard_reset.

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. Refer to Laser Power Calibration Routine (Level 1) for further information). 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.

getHidReportDesc – Invoke when there is a HID device descriptor request interrupt is received form host. If normal_mode variable is '1' then HID compliant mouse device's descriptor and report descriptor will be loaded, else the HID compliant device's descriptor will be loaded.

calibration_report_descriptors – Called in getHidReportDesc if normal_mode variable is '0'. This routine will load HID compliant device's report descriptor. The report descriptor defined the communication protocol and report format between the device and host.

Laser Power Calibration Routine (Level 1)

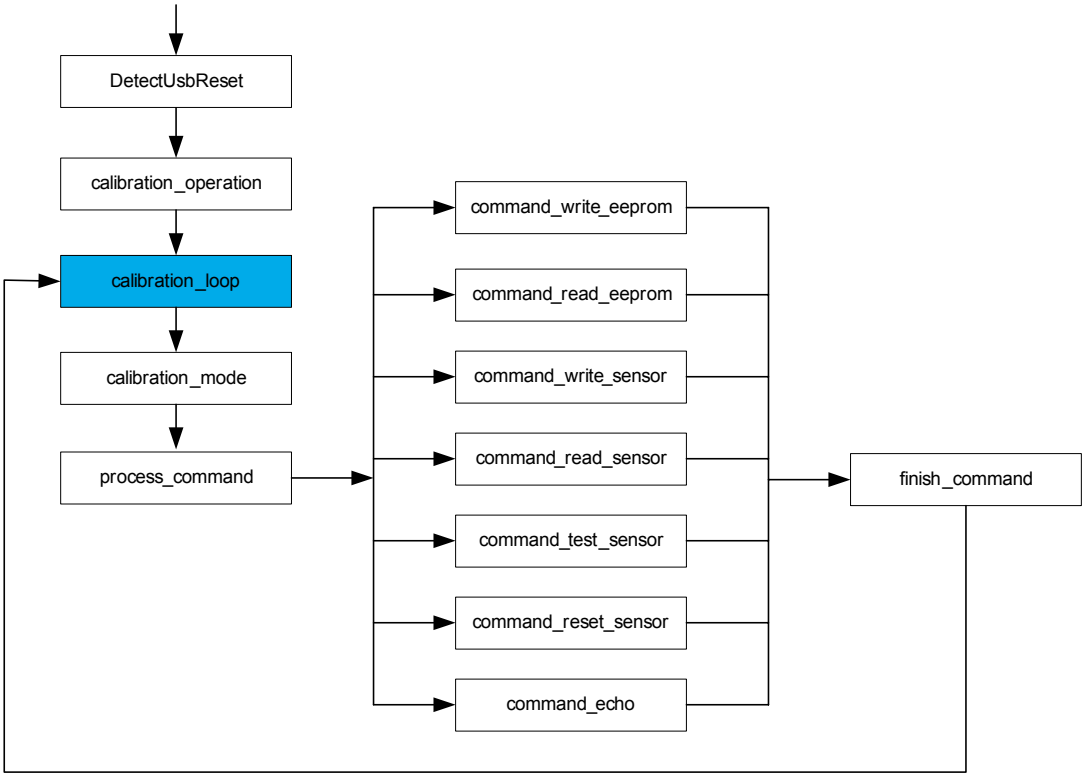


Figure 4.0 Laser Calibration Function Call Map

calibration_operation – This routine reset the sensor and call Load_SROM to load the SROM (Shadow ROM) firmware into the ADNS-6000 optical sensor.

calibration_loop – Main loop for calibration mode.

calibration_mode – This routine call process_command to service any request command receive.

process_command – This select the operation to perform based on the command received on INPUT endpoint 1.

command_write_eeprom – Called in process_command, will call Write_EEPROM routine to write value to the EEPROM for a given memory address.

command_read_eeprom – Called in process_command, will call Read_EEPROM routine to read value from specific EEPROM memory address.

command_write_sensor – Called in process_command, will call WriteSensor routine to write value to sensor register.

command_read_sensor – Called in process_command, will call ReadSensor routine to read register value from sensor.

command_test_sensor – This routine first reset the sensor and disable the laser. Then call AdjustLaser to load LP_CFG0 and LP_CFG1 value from EEPROM. Later, the sensor is set to continuous mode and laser is enabled.

command_reset_sensor – This routine reset the ADNS-6000 sensor.

command_echo – This routine echo back any data bytes upon received on INPUT endpoint 1.

finish_command – This routine format response result value and send to the USB OUTPUT endpoint 0 then return.

Laser Power Calibration Routine (Level 2)

ReadSensor – This routine read a register value from the sensor and stored the value into spi_data variable.

WriteSensor – This routine write value to the sensor for a given register address.

Read_EEPROM – This routine read a value from EEPROM memory and stored the value to spi_data variable.

Write_EEPROM – This routine write value to the EEPROM for a given memory address.

EEPROM_WREN – This routine enable the EEPROM chip-select pin.

AdjustLaser – called to calibrate the laser output power to the required 506uW. This will ensure that the laser output power meets Class 1 eye safety. Customer must ensure that the correct LP_CFG0 and LP_CFG1 register values are written into the registers for proper LASER operation.

enable_laser – Enable laser by writing value '0' to configuration II - register (0x16) force_disable field.

disable_laser – Disable laser by writing value '1' to configuration II register (0x16) force_disable field.

SetDCMode – Set the laser to continuous on mode (laser calibration mode).

Read_LP_CFG_REG – This routine reads calibrated LP_CFG0 and LP_CFG1 value from EEPROM which will be programmed to sensor in AdjustLASER routine.

EEPROM

Memory Address	Description
0x00	Stored LP_CFG0 register value
0x01	Stored LP_CFG1 register value

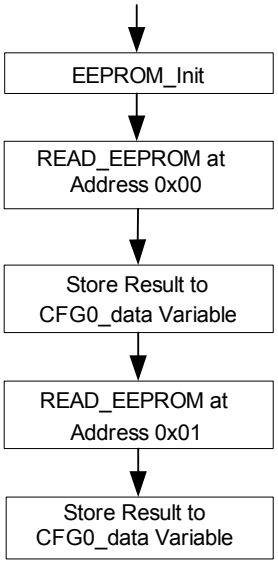


Figure 5.0 Read_LP_CFG_REG Routine Flowchart

USB Report Descriptor

Figure 6.0 Report Descriptor for HID-compliant Mouse

```
;******************************************************************************
report_des
hidReportDescTable:

        db 05h, 01h     ; usage page (generic desktop)
        db 09h, 02h     ; usage (mouse)

        db A1h, 01h     ; collection (application)
        db 05h, 09h     ; usage page (buttons)
        db 19h, 01h     ; usage minimum (1 button)
        db 29h, 03h     ; usage maximum (3 buttons)
        db 15h, 00h     ; logical minimum (0)
        db 25h, 01h     ; logical maximum (1)
        db 95h, 03h     ; report count (3 reports)
        db 75h, 01h     ; report size (1 bit each)
        db 81h, 02h     ; input (Data, Var, Abs)
        db 95h, 01h     ; report count (1 report)
        db 75h, 05h     ; report size (5 bits)
        db 81h, 03h     ; input (Cnst, Ary, Abs)
        db 05h, 01h     ; usage page (generic desktop)
        db 09h, 01h     ; usage (pointer)
        db A1h, 00h     ; collection (linked)
        db 09h, 30h     ; usage (X)
        db 09h, 31h     ; usage (Y)
        db 15h, 81h     ; logical minimum (-127)
        db 25h, 7Fh     ; logical maximum (127)
        db 75h, 08h     ; report size (8 bits each)
        db 95h, 02h     ; report count (2 reports)
        db 81h, 06h     ; input (Cnst, Var, Rel)
        db C0h          ; end collection

        db 09h, 38h     ; wheel
        db 95h, 01h     ; wheel size = 1 byte
        db 81h, 06h     ; variable data bit field with relative position
        db 09h, 3ch     ; motion wake up
        db 15h, 00h     ; 0 no movement
        db 25h, 01h     ; 1 movement
        db 75h, 01h     ; 1st bit represent movement
        db 95h, 01h     ; 1 report
        db b1h, 22h     ; variable data bit field with absolute
                        ; positioning and no preferred state
        db 95h, 07h     ; 7 reports for reversing, upper 7 bits of
                        ; byte 3
        db b1h, 01h     ; constant array bit field with absolute
                        ; positioning
        db c0h          ; end collection, end collection

hidReportDescTableEnd:
```

Figure 7.0 Report Descriptor for HID-Compliant Device

```
;*****************************************************************
report_des_0
hidReportDescTable_0:

        db      06h, A0h, FFh   ; usage page (vendor defined)
        db      09h, 01h        ; usage (vendor defined)
        db      A1h, 01h        ; collection (application)
        db      09h, 02h        ; usage (vendor defined)
        db      A1h, 00h        ; collection (linked)
        db      06h, A1h, FFh   ; usage page (vendor defined)

                                ; The input report
        db      09h, 03h        ; usage - vendor defined
        db      09h, 04h        ; usage - vendor defined
        db      15h, 80h        ; Logical Minimum (-128)
        db      25h, 7Fh        ; Logical Maximum (127)
        db      35h, 00h        ; Physical Minimum (0)
        db      45h, FFh        ; Physical Maximum (255)
        db      66h, 00h, 00h   ; Unit (None (2 bytes))
        db      75h, 08h        ; Report Size (8) (bits)
        db      95h, 03h        ; Report Count (2) (fields)
        db      81h, 02h        ; Input (Data, Variable, Absolute)

                                ; The output report
        db      09h, 05h        ; usage - vendor defined
        db      09h, 06h        ; usage - vendor defined
        db      15h, 80h        ; Logical Minimum (-128)
        db      25h, 7Fh        ; Logical Maximum (127)
        db      35h, 00h        ; Physical Minimum (0)
        db      45h, FFh        ; Physical Maximum (255)
        db      66h, 00h, 00h   ; Unit (None (2 bytes))
        db      75h, 08h        ; Report Size (8) (bits)
        db      95h, 03h        ; Report Count (2) (fields)
        db      91h, 02h        ; Output (Data, Variable, Absolute)

                                ; The Feature report
        db      85h, 01h        ; report ID
        db      09h, 05h        ; usage - vendor defined
        db      09h, 06h        ; usage - vendor defined
        db      15h, 80h        ; Logical Minimum (-128)
        db      25h, 7Fh        ; Logical Maximum (127)
        db      35h, 00h        ; Physical Minimum (0)
        db      45h, FFh        ; Physical Maximum (255)
        db      66h, 00h, 00h   ; Unit (None (2 bytes))
        db      75h, 08h        ; Report Size (8) (bits)
        db      95h, 02h        ; Report Count (2) (fields)
        db      B1h, 02h        ; Feature (Data, Variable, Absolute)
        db      C0h, C0h        ; end collection, end collection

hidReportDescTableEnd_0:
```


AV01-0999EN - January 17, 2008