ADNK-5033-FS27

USB RF Wireless Optical Mouse Designer's Kit

Design Guide

Introduction

This design guide describes how a small form power-effective optical mouse can be built using the Avago Technologies' ADNS-5030 optical navigation sensor and Freescale Semiconductor's MC68HC908QY4A and MC68HC908JB12 microcontroller. The document starts with the basic operations of a computer mouse peripheral followed by an introduction to the MC68HC908QY4A and MC68HC908JB12 microcontroller and the Avago Technologies ADNS-5030 Optical Navigation Sensor. The software section of this design guide describes the architecture of the firmware required to implement the mouse and the USB receiver functions. All schematic diagrams of the MC68HC908QY4A microcontroller to the ADNS-5030 optical sensor and buttons of a standard mouse and the USB receiver can be found in Appendix A. The ADNS-5030 data sheet is available from the Avago Technologies website at: http://www.avagotech.com. The MC68HC908QY4A and MC68HC908JB12 data sheets are available from the Freescale website at www.freescale.com/semiconductors. USB documentation can be found at the USB Implementers Forum web site at www.usb.org.

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 SPI signal generated by optical encoder. This design guide shows how to connect to and manage a standard configuration of mouse hardware, as well as handle the USB protocols. USB protocol provides a standard way of reporting mouse movement and button presses to the PC.

Mouse Optics

The motion Z-wheel is detected using the traditional method by decoding the SPI signal generated by optical encoder. 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 z-wheel. 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 represent a count of mouse movement. Comparing the last state of the optics to the current state derives direction information. As shown in figure 1, 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.

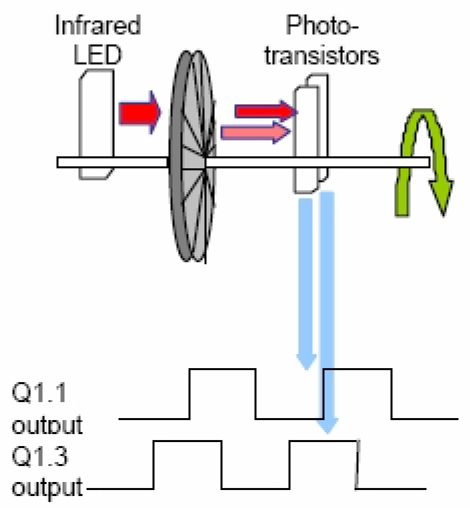


Figure 1. Optics Quadrature Signal Generation

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.

Optical Navigation Sensor

Avago Technologies ADNS-5030 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, and a four-wire serial port. The MC68HC908QY4A USB microcontroller periodically reads the ADNS-5030's Delta_X and Delta_Y registers to obtain any horizontal and vertical motion information happening as a result of the mouse being moved.

This motion information will be reported to the PC to update the position of the cursor. The advantages of using ADNS-5030 optical sensor are the efficient power management, high tracking accuracy, and flexibility of programming the optical sensor via the SPI port.

Features include:

- Precise optical navigation technology
- No mechanical moving parts
- Smooth surface navigation
- Auto frame speed
- Accurate motion up to 14ips
- 500/1000 cpi resolution selection
- High speed motion detector
- Single 3.3 volt power supply
- Power conversation mode during times of no movement
- On chip LED drive with regulated current
- On chip resonator
- Serial port registers
  - Programming
  - Data transfer

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

Freescale MC68HC908QY4A and MC68HC908JB12

The MC68HC908QY4A and MC68HC908JB12 are members of the low-cost, high-performance M68HC08 Family of 8-bit microcontroller units (MCUs). The M68HC08 Family is a Complex Instruction Set Computer (CISC) with Von Neumann architecture. All MCUs in the family use the enhanced M68HC08 central processor unit (CPU08) and are available with a veriety of modules, memory sizes and types.

HC08 Features include:

- High-performance M68HC08 CPU core
- Fully upward-compatible object code with M68HC05 Family
- In-circuit FLASH programming
- Low-voltage inhibit (LVI)
- Power-on reset
- Power saving stop and wait modes

Special Features for MC68HC908QY4A:

- 5-V and 3-V operating voltages (VDD)
- 8-MHz internal bus operation at 5 V, 4-MHz at 3V
- Trimmable internal oscillator
  - 3.2 MHz internal bus operation
  - 8-bit trim capability allows 0.4% accuracy
  - ± 25% untrimmed
- Auto wakeup from STOP capability
- 4096 bytes of –on-chip Flash memory with security feature
- 128 bytes of on-chip random access memory (RAM)

Special Features for MC68HC908JB12:

- 5-V operating voltages (VDD)
- 6-MHz internal bus frequency
- 12288 bytes of on-chip Flash memory with Security feature
- 384 bytes of on-chip random access memory (RAM)
- Universal Serial Bus specification 2.0 low-speed functions
  - 1.5Mbps data rate
  - On-chip 3.3V regulator
  - Endpoint 0 with 8-byte transmit buffer and 8-byte receive buffer
  - Endpoint 1 with 8-byte transmit buffer
  - Endpoint 2 with 8-byte transmit buffer and 8-byte receive buffer

Wireless RF Technology

A high data rate 27-MHz RF link is designed as the wireless communication media for this application. The RF frequency is determined by the crystal frequency used at the oscillator circuit stage. The transmission data rate is 4.8 kbps. See figure 2.

The high-frequency carrier signal on the transmitter side is modulated by digital encoded data from the MC68HC908QY4A using a FSK modulation scheme. The modulated RF signal is propagated through free-air space and received by an integrated chip, the xx3361, on the receiver side which includes all mixer, local oscillator, and demodulator circuits. The demodulated data output is received by the MC68HC908JB12 for decoding and processing. The data will then be converted to the USB mouse report format and sent to the host.

RF Transmitter

The RF transmitter consists of three parts:

- The crystal type oscillator
- The FSK modulation switching circuit
- The RF amplifier

The crystal oscillator works with a crystal frequency at half of the target channel frequency and the second harmonic frequency is filtered out by the RF amplifier together with a high Q-factor antenna. For example, a 13.5225-MHz crystal is used for a frequency channel at 27.045-MHz.

The FSK modulation is achieved by changing the loading capacitance at the crystal with a transistor switching circuit controlled by the encoding data generated from the MCU. The maximum data rate for a particular FSK transmission is limited by the RF bandwidth of the system and controlled by the frequency deviation which represents the logic “0” and logic “1” data. In general, the frequency deviation should be adjusted proportionally to the change on the required transmission data rate. Higher data rates require more bandwidth/frequency deviation. The frequency deviation for this 4.8-kbps data rate wireless mouse application is increased from ±2.5 kHz to ±4.5kHz which is controlled by the crystal characteristics.

The gain of the RF amplifier in the final stage should be adjusted to compensate the gain loss at the oscillator stage with extend bandwidth operation. Two stages of RF amplifier are used in this reference design to maintain the performance in communication distance.

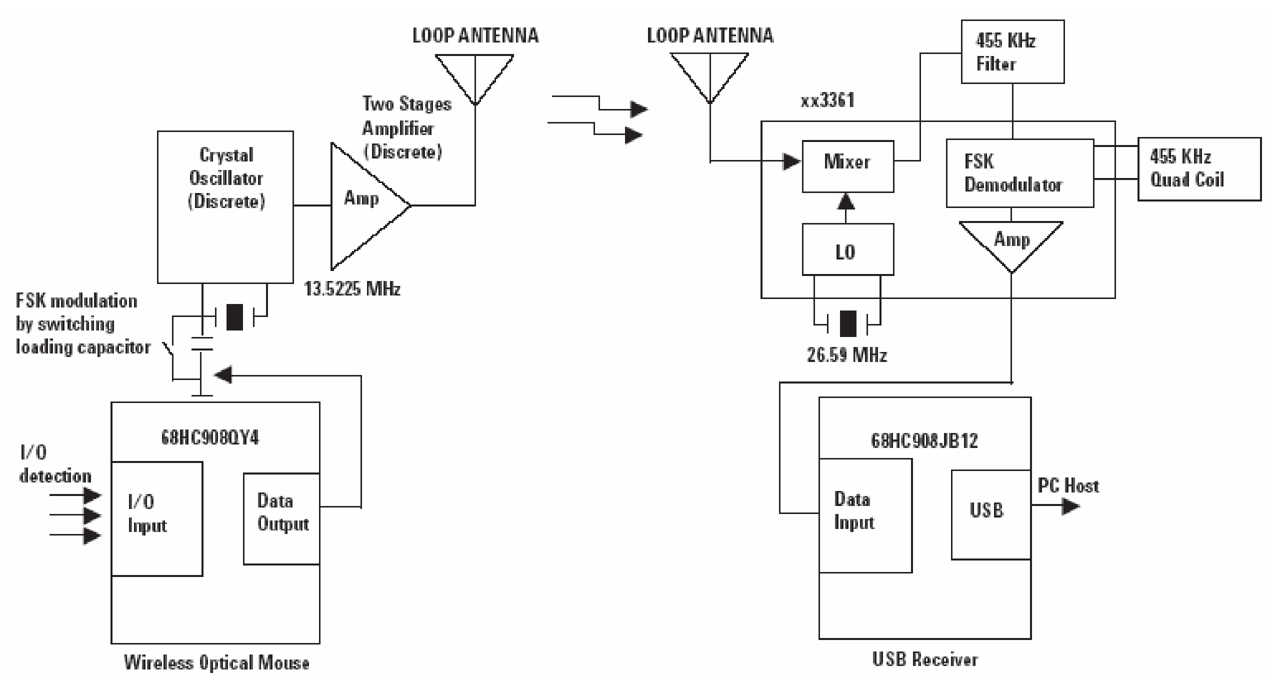


Figure 2. The Block Diagram of the 27-MHz RF Link

RF Receiver

The RF receiver is implemented by using a single-chip solution (3361 compatible part) that includes:

- A frequency downward conversion mixer
- A local oscillator circuit
- A baseband FSK quadrature demodulation unit

The RF input signal from the antenna, is frequency down converted into an IF signal at 455kHz by the mixer and oscillator circuits. The IF frequency value is equal to the RF input frequency plus or minus the LO input frequency. The higher frequency components should be filtered out by using a passive IF filter. The bandwidth should be increased from 15 kHz to 20 kHz to match the data rate change on the transmitter side.

However, the image frequency component would not be filtered out by the IF filter. This should be considered in the PCB layout in order to prevent any noise component at image frequency to be injected into the mixer input. An example of this would be a noise pattern generated from the MCU.

Hardware Implementation

The standard hardware to implement a wireless optical mouse is shown in Figure 3. For X and Y movement, the optical encoder 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 functions of the MC68HC908QY4A are to:

- Get the XY displacement from the optical sensor ADNS-5030
- Detect the Z displacement
- Check button status
- Control the RF circuitry to send out data
- Perform the overall power management

Three standard left, middle and right buttons together with one buttons together with one button for the identity device (ID) code are implemented. The ID code can be stored in the FLASH of the MC68HC908QY4A.

When the ID button in the transmitter and the one in the receiver are pressed, a random ID code is generated at the transmitter and sent to the receiver. After receiving the new ID code, the receiver stores it in the FLASH of the receiver MCU.

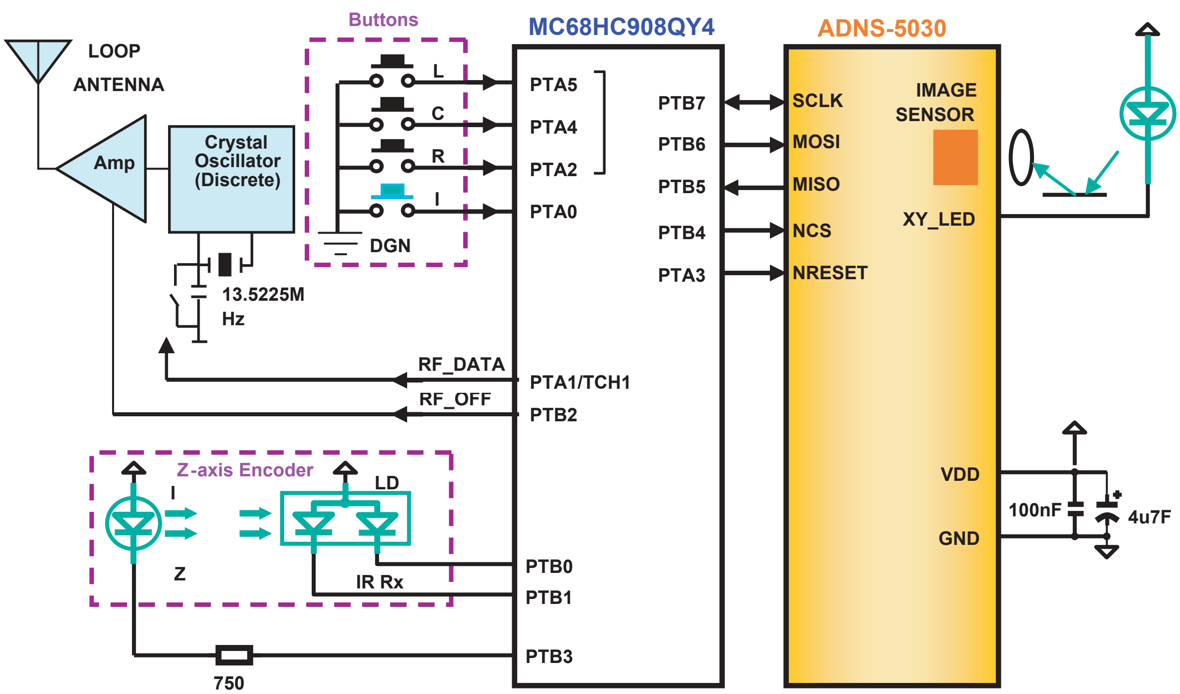


Figure 3. MC68HC908QY4A-ADNS-5030 Optical Mouse Hardware Block Diagram - Mouse

The communication between the sensor and the MC68HC908QY4A is through a serial peripheral interface (SPI) with clock input at the SCLK pin, the data interface at the MISO and MOSI pins.

The receiver consists of:

- The MC68HC908JB12
- A button for identity device (ID) code setting
- A in-circuit programming (ICP) connector
- RF front-end circuitry

The functions of the MC68HC908JB12 are to handle the USB transactions, capture and process data from RF receiver front end. The processed data is converted into USB report format and sent to the host.

The ID button is used to interrupt the MCU for a new ID code detection. The MCU will search if there is any ID code embedded in the RF data packet and store the new ID in FLASH memory. There is a transistor wakeup circuit connected at the IRQ pin and is controlled by an I/O pin. The MCU will configure the wakeup circuit before entering stop mode for power saving. The RC components connected at the transistor base input will be charged up, and eventually turn on the transistor and pull low the IRQ to wakeup the MCU.

The ICP interface is connected at port A. The ICP is used for future firmware updating through the Cyclone Programming Tools.

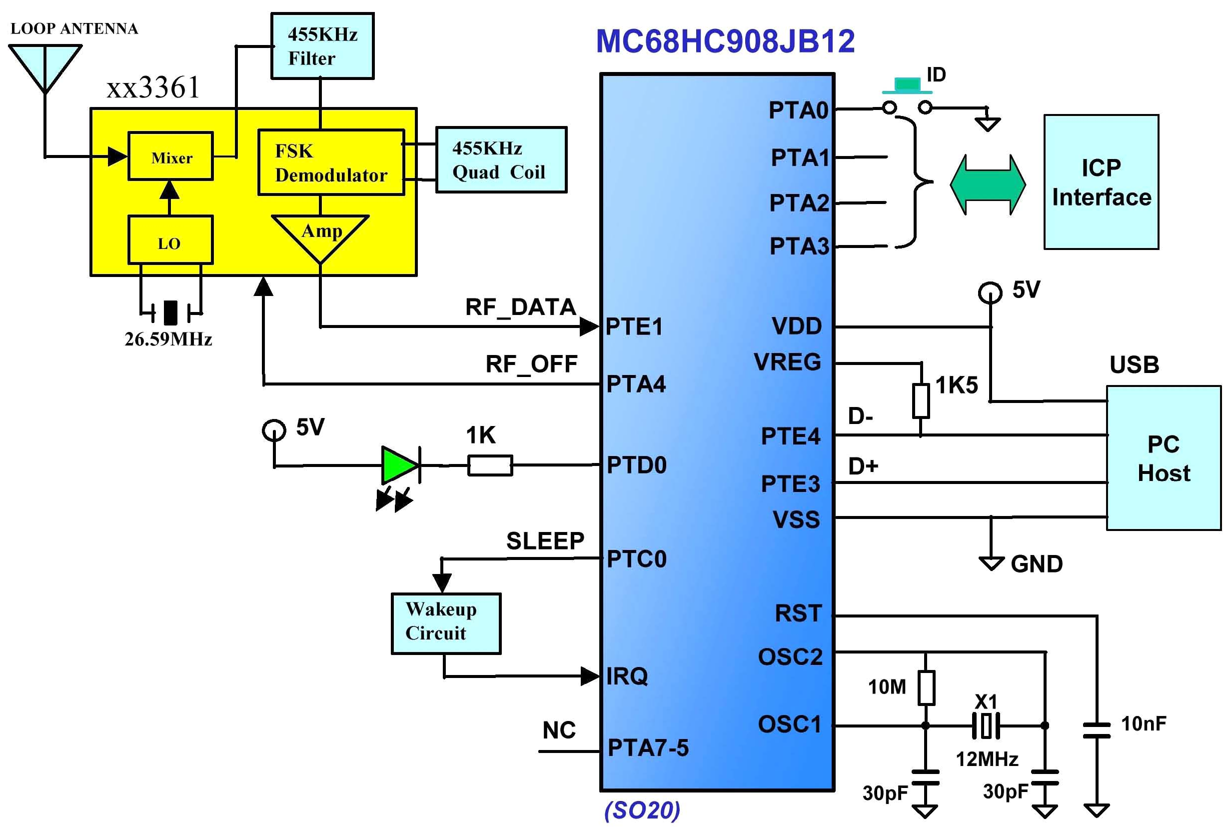


Figure 4. MC68HC908JB12-ADNS-5030 Optical Mouse Hardware Block Diagram - Receiver

PCB Layout Guidelines

Care should be taken in the PCB layout in order to avoid any noise generated from MCU coupling into the RF stage. For example:

The power supply traces used for digital and analog circuit blocks should be separated.

The location of decoupling capacitors should be as close as possible to device's supply input pins (VDD / VSS or VCC / GND).

The VDD to VSS ground loop area should be reduced to minimize the magnetic coupling effect.

The PCB trace loop formed by any input/output (I/O) signal pin should be kept to a minimum.

The RF receiver uses a loop antenna formed by using a PCB trace line.

Some details on ADNK-5033-FS27

The ADNK-5033-FS27 reference design mouse unit allows users to evaluate the performance of the Optical Tracking Engine (sensor, lens, LED assembly clip, LED) over a USB connection, using a Freescale USB Microcontroller. 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 with standard 3-button USB mouse driver loaded.

Functionality

3-buttons, scroll wheel combi-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-5033-FS27 Unit

The ADNK-5033-FS27 comprises of the plastic mouse casing, printed circuit board (PCB), lens, buttons, and USB cable (See Figure 5). Unscrewing the one screw located at the base of the unit can open the ADNK-5033-FS27 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.

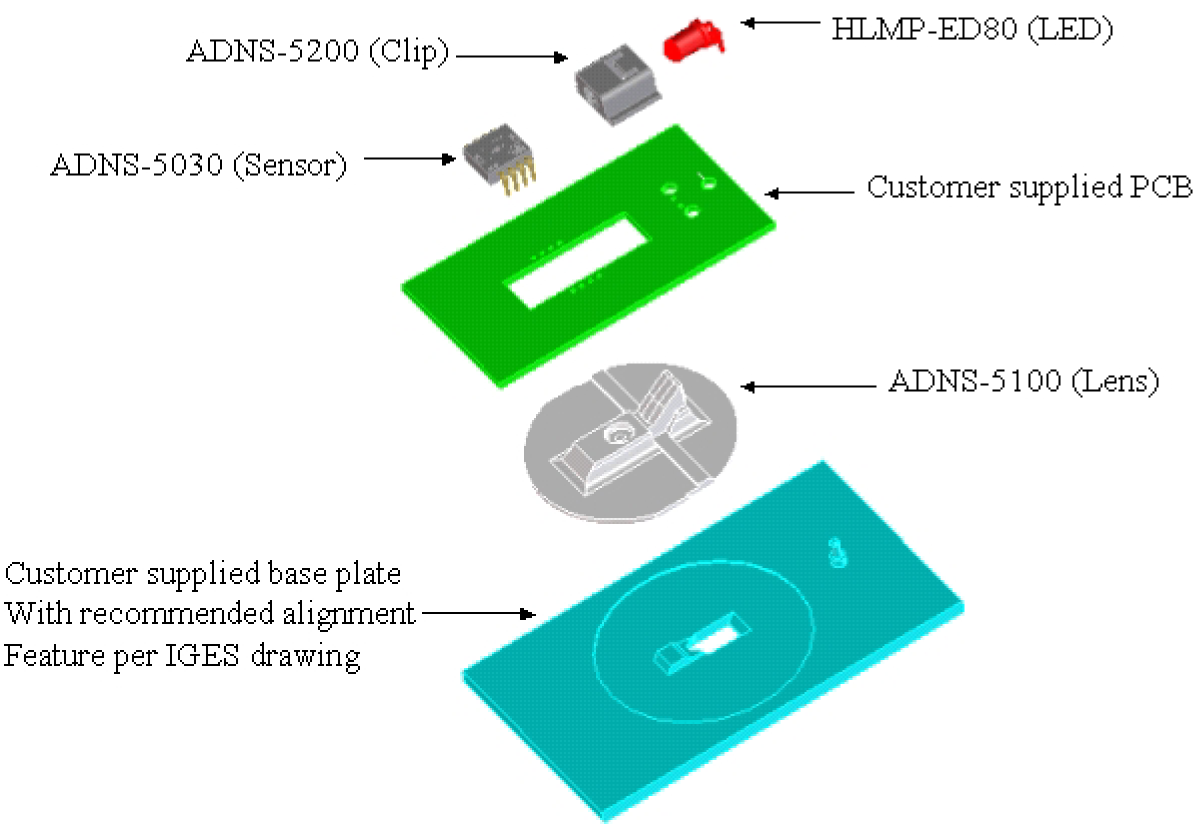


Figure 5. Exploded view drawing of optical tracking engine with ADNS-5030 optical mouse sensor.

While reassembling the components, please make sure that the Z height (Distance from lens reference plane to surface) is valid. Refer to Figure 6.

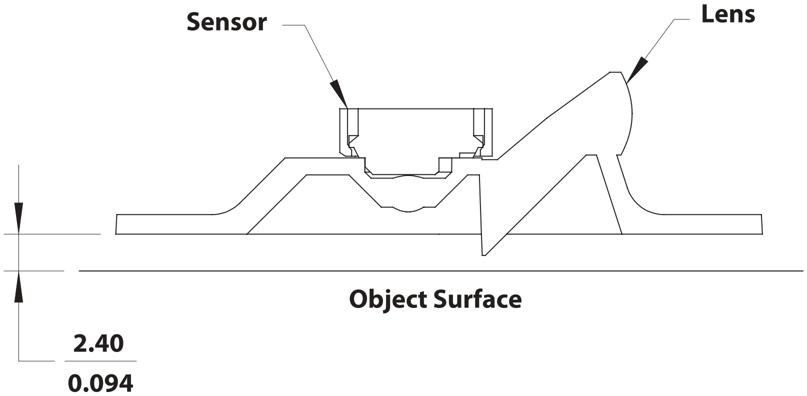


Figure 6. Distance from lens reference plane to surface.

Below is the summary of the components contained in the ADNK-5033-FS27 Designer's Kit.

Sensor

The sensor technical information is contained in the ADNS-5030 Data Sheet.

Microcontroller

Technical information on the Freescale microcontroller is contained in the MC68HC908QY4A and MC68HC908JB12 Data Sheet. The enclosed "CodeWarrior" CD-ROM contains the development tools for the MC68HC908QY4A. These tools will allow the designer to make changes and recompile the source code for MC68HC908QY4A. The assembler software "CASM08Z.EXE" for MC68HC908JB12 is available from "P&E Microcomputer Inc" (www.pemicro.com). To perform In-Circuit Emulation for easier debugging of new code development, contact Freescale to purchase the Development Kit and the Personality Board.

Programming support and programmer adaptors for the Freescale MC68HC908QY4A and MC68HC908JB12 can be found through Freescale or through most 3rd party programming companies. For further information on this product, please contact Freescale Semiconductor.

Lens

The lens technical information is contained in the ADNS-5100 Data Sheet. The flange on the standard ADNS-5100 round lens is for ESD protection.

LED Assembly Clip

The information on the assembly clip is contained in the ADNS-5200 Data Sheet.

LED

The LED technical information is contained in the HLMP-ED80-XX000 Data Sheet and Application Note AN1228. Additional application notes regarding Eye Safety Requirements are also available at Avago Technologies's 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-5033-FS27 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 MC68HC908QY4A mouse firmware is compiled under CodeWarrior environment from Freescale Semiconductor Inc. The firmware includes the following files:

QY4 _OPTICAL_MOUSE.ASM

Main program, includes power up entry point, constant and variables definition.

QY4_INT.ASM

Timer interrupt service routines RF packet transmit, timer tick generation.

SENSOR.ASM

ADNS-5030 optical sensor initialization, SPI routine for optical sensor register read/write operation.

QY4_MOTION.ASM

X-Y motion handler and button detection.

QY4_EEFLASH.ASM

Flash erase/write routines for mouse ID update.

QY4_MISC.ASM

System initialization and miscellaneous routines.

QTQ_REGISTERS.INC

QY4A register header file.

QY4_OPTICAL_MOUSE.H

System header file.

A5030_SENSOR.H

Header file for ADNS-5030 optical sensor.

Mouse Firmware Structure

The mouse firmware consists of two main parts; main routine and timer interrupt routine. A power management scheme is included to minimize the power consumption while maximize the performance.

The main program continually checks with the registers of the optical sensor to see if any XY movement happened. If any XY movement is detected, it gets the X and Y displacements from the sensor registers, puts them in the FIFO buffer and sets the corresponding flags. For every ms timer tick, it checks the Z movement and the button status. Timer interrupt is set for every 104 us which is the base time for the 4.8 kHz data rate transmission. By configuring the timer to Output Compare mode, the RF_Data output pin can be set, clear or toggled for every 104 us. The timer interrupt routine determines whether to set or clear RF_Data pin at the next interrupt time. It also, determines what the current RF_Off pin status should be.

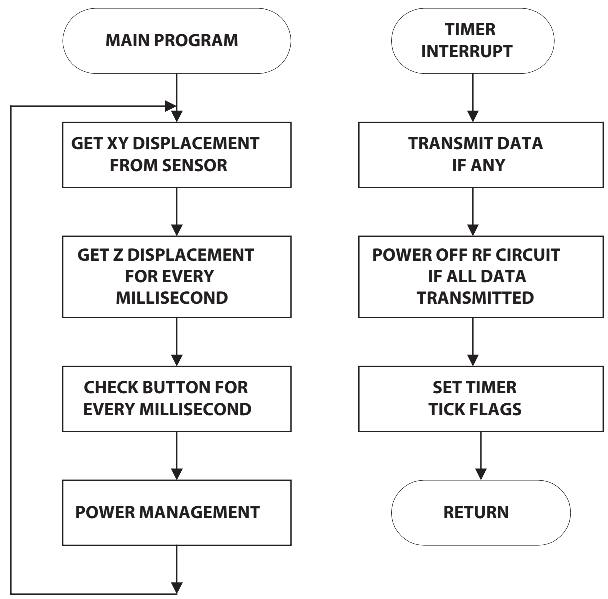


Figure 7. Mouse Firmware Structure

Power Management Scheme

Figure 8 shows the flow of the power management of the optical mouse. The ADNS-5030 has three internal power-saving modes. Each mode has a different motion detection period, affecting response time to mouse motion (Response Time). The sensor automatically changes to the appropriate mode, depending on the time since the last reported motion (Downshift Time). The parameters of each mode are shown in the chart.

The HC908QY4A MCU firmware will optimize the system power consumption according to the motion detection result of the ADNS-5030 optical sensor.

All the system components are fully operating when there is XY displacement motion. The power consumption is the maximum in this stage. The HC908QY4A MCU will continuously to check the ADNS-5030 optical sensor for any displacement motion and detect any button activities or Z-wheel motion.

If there is no motion detected from the optical sensor for 250ms, HC908QY4A MCU will suspend and get into the STOP mode. The HC908QY4A MCU will wakeup periodically approximately every 18ms. The MCU will read the optical sensor and check for any Z-wheel motion during the wakeup period.

If there is no motion detected from the sensor for another 10 seconds, the HC908QY4A MCU will then check the optical sensor and Z-wheel motion on around every 90ms. HC908QY4A will remain in STOP mode at the most of the time.

If there is no motion for further 600s, the HC908QY4A MCU will wakeup to check for the optical sensor and Z-wheel motion on about every 500ms. The system will have the least power consumption in this stage.

During any suspend stages, any XY displacement motion, Z-wheel motion or buttons action will bring the HC908QY4A back to the fully responsive stage, i.e., reading the optical sensor continuously and detecting the Z-wheel and buttons on every 1ms.

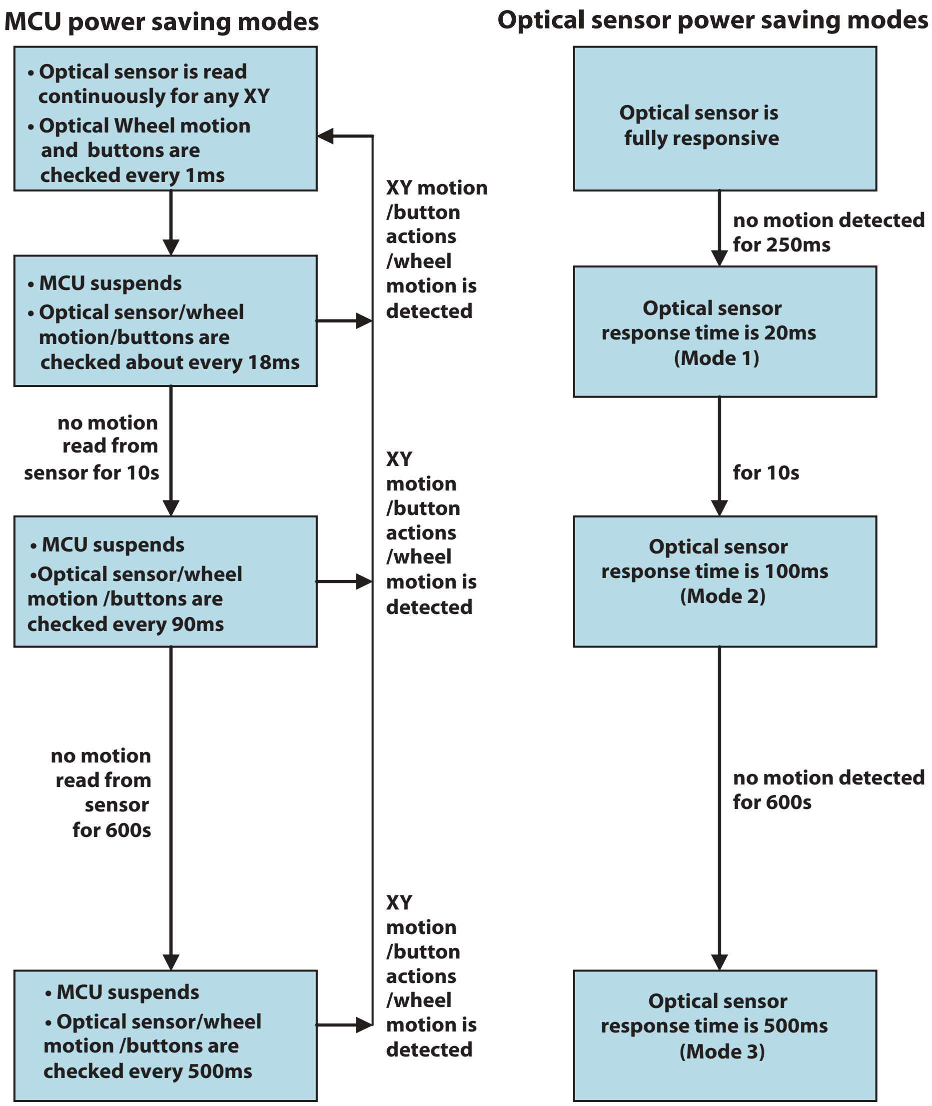


Figure 8. Power Management.

RF Packet Structure

All the mouse data is transmitted by RF front end with the FSK modulation.

There are two types of RF packet: XY movement packet or Buttons status/Z movement packet. The Button status/Z movement packet will also be used for mouse ID updating process. It will be further described in later session.

All these packets have the same structure. Each packet will consist of start field, data field and checksum field.

XY Displacement Packet:

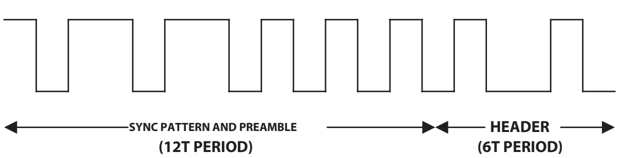
Start Field: Consist of preamble and synchronization pattern. A header pattern is also included for distinction between XY movement and button status/Z movement packet.

Data Field: There are two bytes data in Data field. The first and second bytes are 8-bit X and Y displacement respectively. They are presented in 2's complement format.

Checksum Field: The 8 bit in Checksum field contains the sum of the X, Y displacement and the stored ID of the mouse.

There is one STOP bit after each byte of Data or Checksum field. Lowest Significant Bit (LSB) will be transmitted first. The contents of these fields are encoded by using Manchester Method.

Start	X Displacement	Y Displacement	Checksum
START FIELD	DATA FIELD		CHECKSUM FIELD



1T period is equal to 104 us Timer interrupt period

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	STOP

XY displacement or Checksum byte

Button Status and Z Displacement Packet:

Start Field: Consists of preamble and synchronization pattern. The header pattern is for distinction between XY movement and button status/Z movement packet as mentioned before.

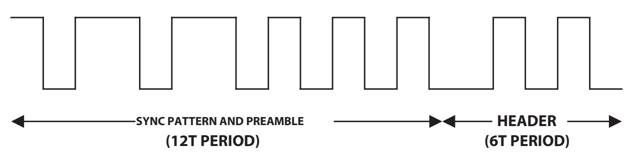
Data Field: The first byte contains the status of the four buttons. The second byte is either Z movement value or a new ID value.

In the button status byte, a "1" means the corresponding button is pressed and "0" means the button has been released. When the "ID" button status is "1", the value of the second byte will represents a new 8 bit ID to be used by the mouse and receiver pair. Otherwise, it is an normal Button status/Z displacement packet where the second byte represent the Z movement in 2's complement format.

Checksum Field: The 8 bit in Checksum Field contains the sum of the X, Y displacement and the stored ID value.

Again, there is one STOP bit after each byte of Data or Checksum field. Lowest Significant Bit (LSB) will be transmitted first. The contents of these fields are encoded by using Manchester Method.

Start	Button Status	Z or ID	Checksum
START FIELD	DATA FIELD		CHECKSUM FIELD



1T period is equal to 104 us Timer interrupt period

R	R	N-BTN	R-BTN	L-BTN	ID	R	R	STOP

Button Status Byte

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	STOP

Checksum Byte

MC68HC908JB12 Receiver Firmware

The MC68HC908JB12 receiver firmware is compiled under CASM08Z.EXE from P&E Microcomputer Inc. The firmware includes the following files:

MOTSTART.asm

Program entry point after power up or reset. Constant and Variable definition, I/O and timer initialization routines are contained in this files.

USB-MAIN.asm

USB main program. USB endpoints transmit setting, Suspend and Resume handler are included.

HID-KBD.ASM

USB Standard Device Request Handler and HID Class Device Requester handler.

KBD-MSE.H

Header file containing the USB HID device descriptors.

JB16-INT.ASM

USB Interrupt Service Routines. USB control transfer handler.

RX_T1.ASM

Timer 1 Interrupt service routine. It contains the capture routine for receiving and decoding the incoming RF mouse packet.

RX_T2.ASM

Timer 2 Interrupt service routine. Timer Capture routine can be added here for addition RF channel receiving, e.g. a keyboard channel.

JB12-EQ2.H

MC68HC908JB12 registers header file.

EE_FLASH.ASM

Flash erase/write routines to be used for update mouse ID.

KEY-USB.ASM

Key maps and routines used in keyboard reference portion.

KEYMAPTX.ASM

Routines used in Keyboard reference portion.

MACR08.ASM

MACRO library for HC08 CPU.

MC68HC908JB12 Receiver Firmware Structure

The firmware consists of three main parts:

- Main routine
- Timer interrupt routine to capture and decode mouse data
- USB interrupt routine

The USB routine includes reference codes for both universal serial bus (USB) mouse and keyboard. Thus, it can be a reference on how to implement a composite USB keyboard-mouse device using the MC68HC908JB12.

Figure 9 shows the flow of the main program for the MC68HC908JB12 receiver. The main routine continually checks to see if there is any valid mouse data in the receiver buffer queue. If there is new data received, the data will be converted to USB report format and sent to the host via USB endpoints.

Figure 10 shows the USB interrupt routines. The USB engine automatically responds to a valid

USB token with ACK, NAK, or STALL, depending on the register settings, and ignore it if it's invalid. The firmware has to set the registers for the USB engine to give correct response to the token in different stages. The USB interrupt will be executed whenever there is an EOP, resume signal from host, valid data received, or data transmitted. The USB interrupt routine also makes preparation for the next USB transaction and handles any valid command or data received.

Figure 11, Figure 12 and Figure 13 show the routines handling control transfers. Control transfers have two or three transaction stages: setup, data (optional), and status as shown below:

- Control write: SETUP, OUT, OUT, OUT…IN
- Control read: SETUP, IN, IN, IN…OUT
- No Data control: SETUP, IN

The firmware first distinguishes the kind of control transfer and then does the corresponding preparation for the next stage.

One timer interrupt routine is used to capture the RF mouse data. The firmware flow is shown in Figure 14. The timer is set to input capture mode. With interrupt happens on falling and rising edge, it can calculate the pulse width of the RF data. The routines will proceed to detect data only if the pulse width matches the data structure of the RF packet. The detected data bit will be put into a temporary buffer. The timer interrupt events will keep accruing until one complete RF packet has been received. All the received data will then be put into receiver buffer queue for the main program to process. In case when USB bus is being suspended, a received mouse packet means the USB bus and the host computer needed to wakeup. The routine will then signal the main routine to wakeup the host instead of updating the receiver buffer queue. The mechanism for the wakeup mechanism is described in more details in the "Wakeup Detection Mechanism" section.

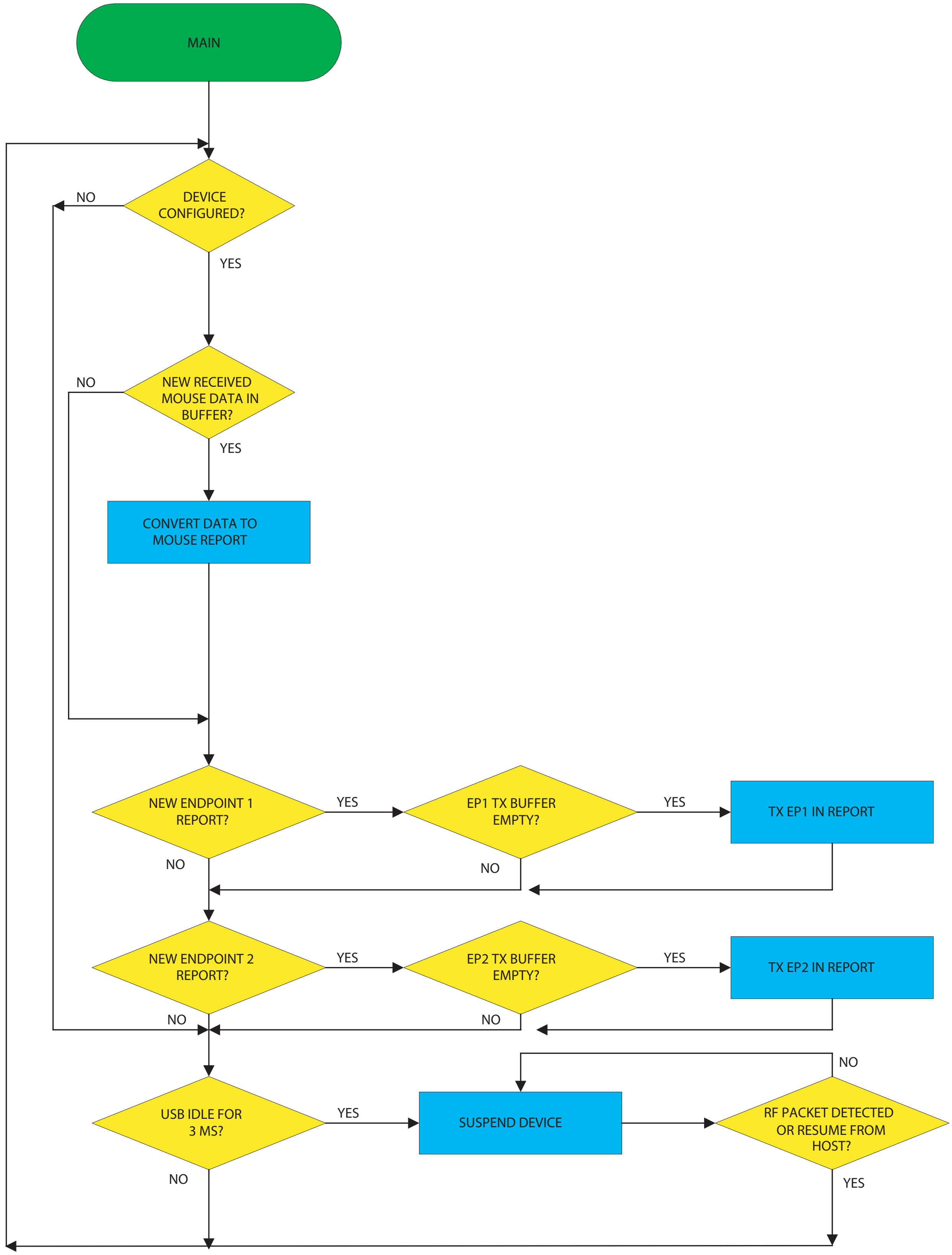


Figure 9. Firmware Main Program Flow

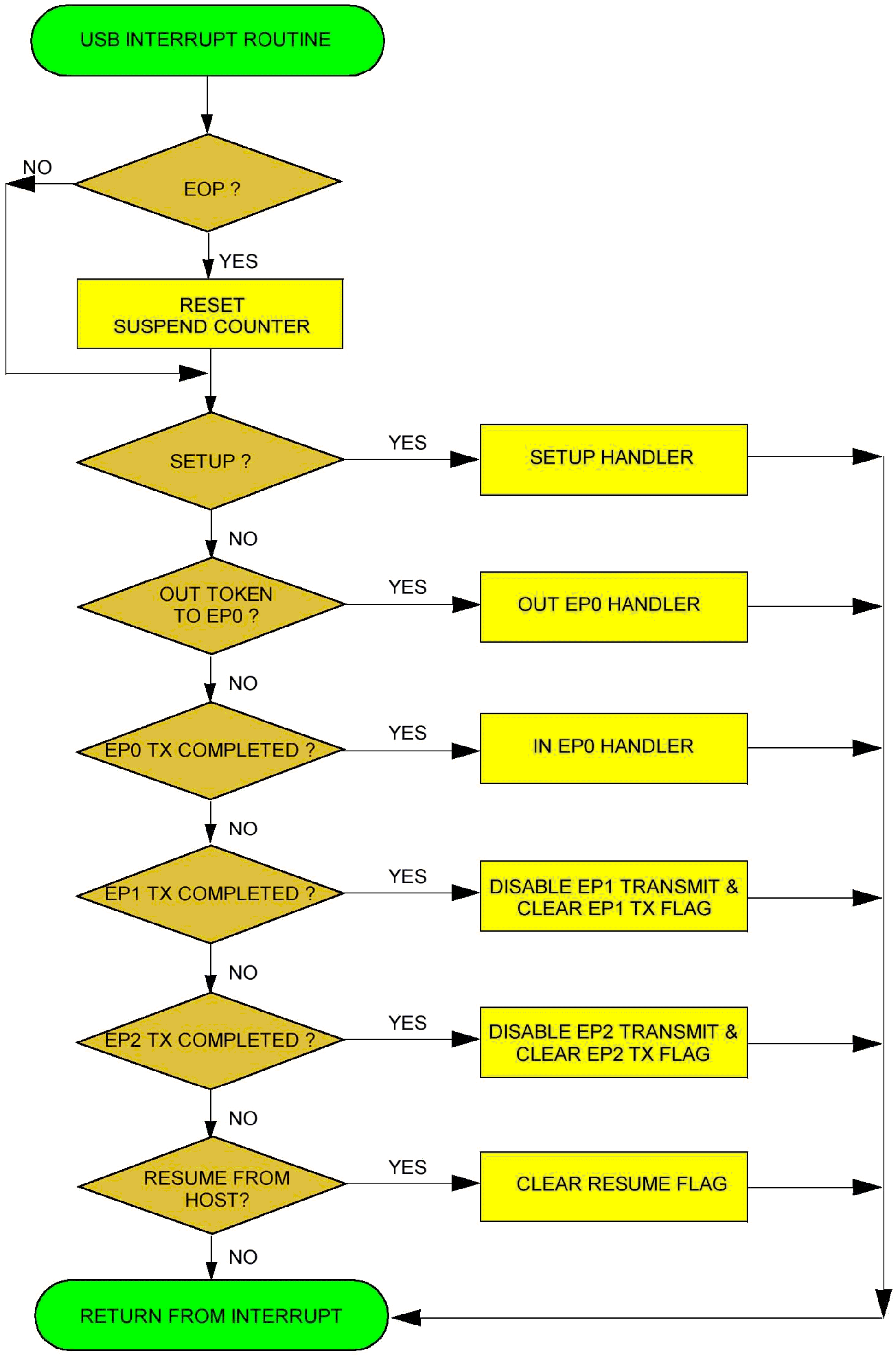


Figure 10. USB Interrupt Routine

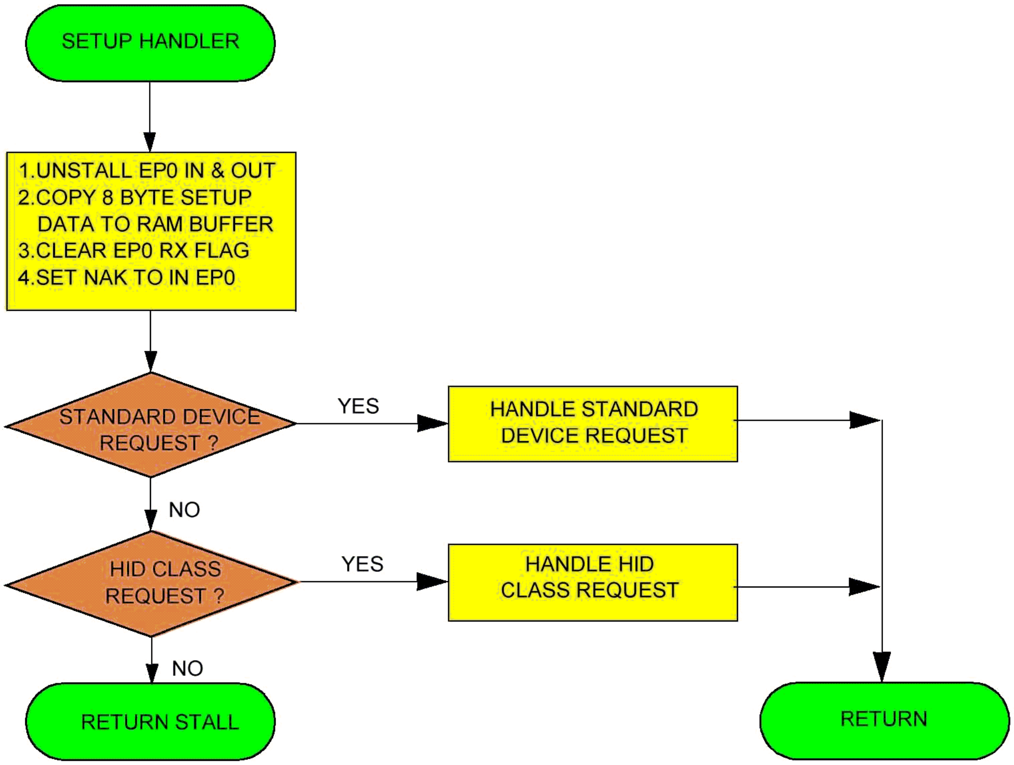


Figure 11. USB Setup Routine

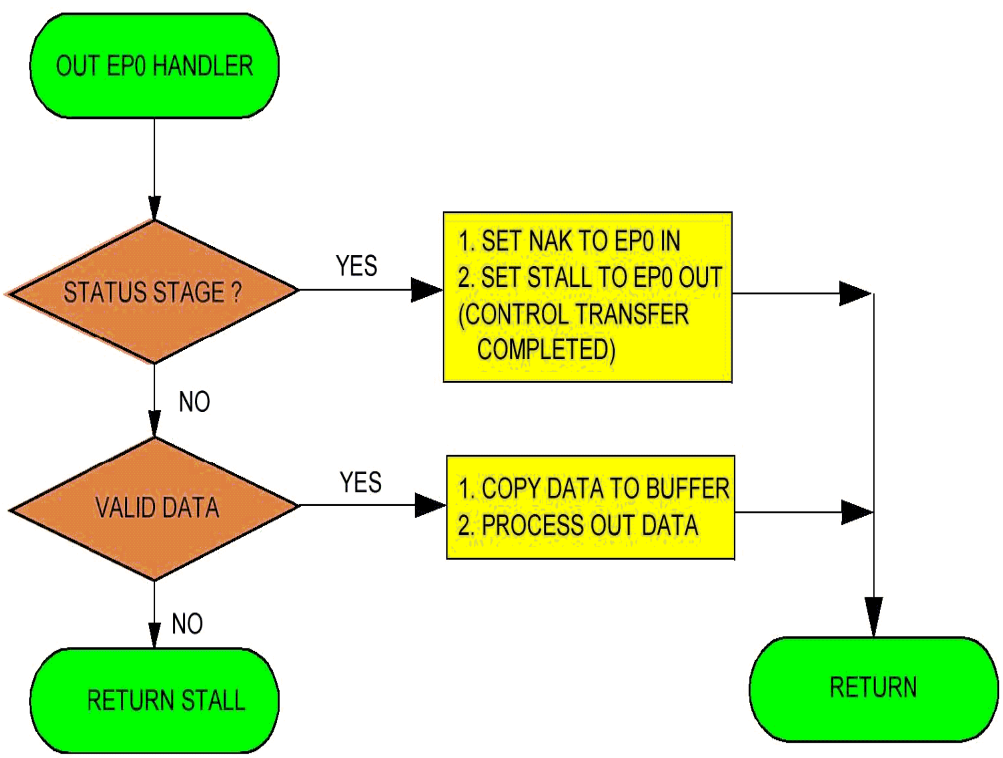


Figure 12. OUT EPO Handler

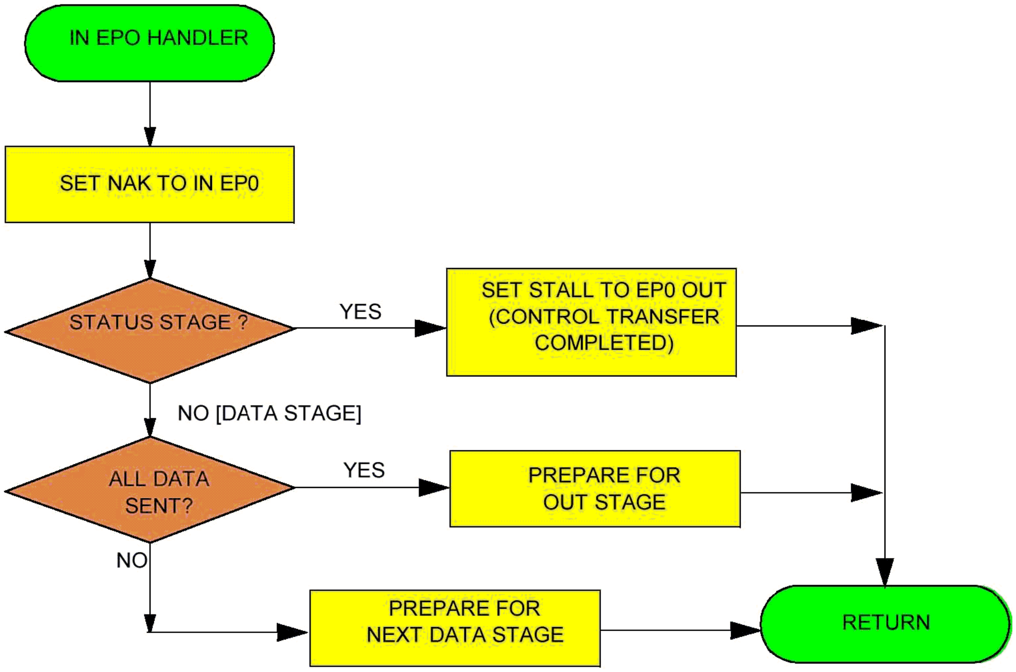


Figure 13. IN EPO Handler

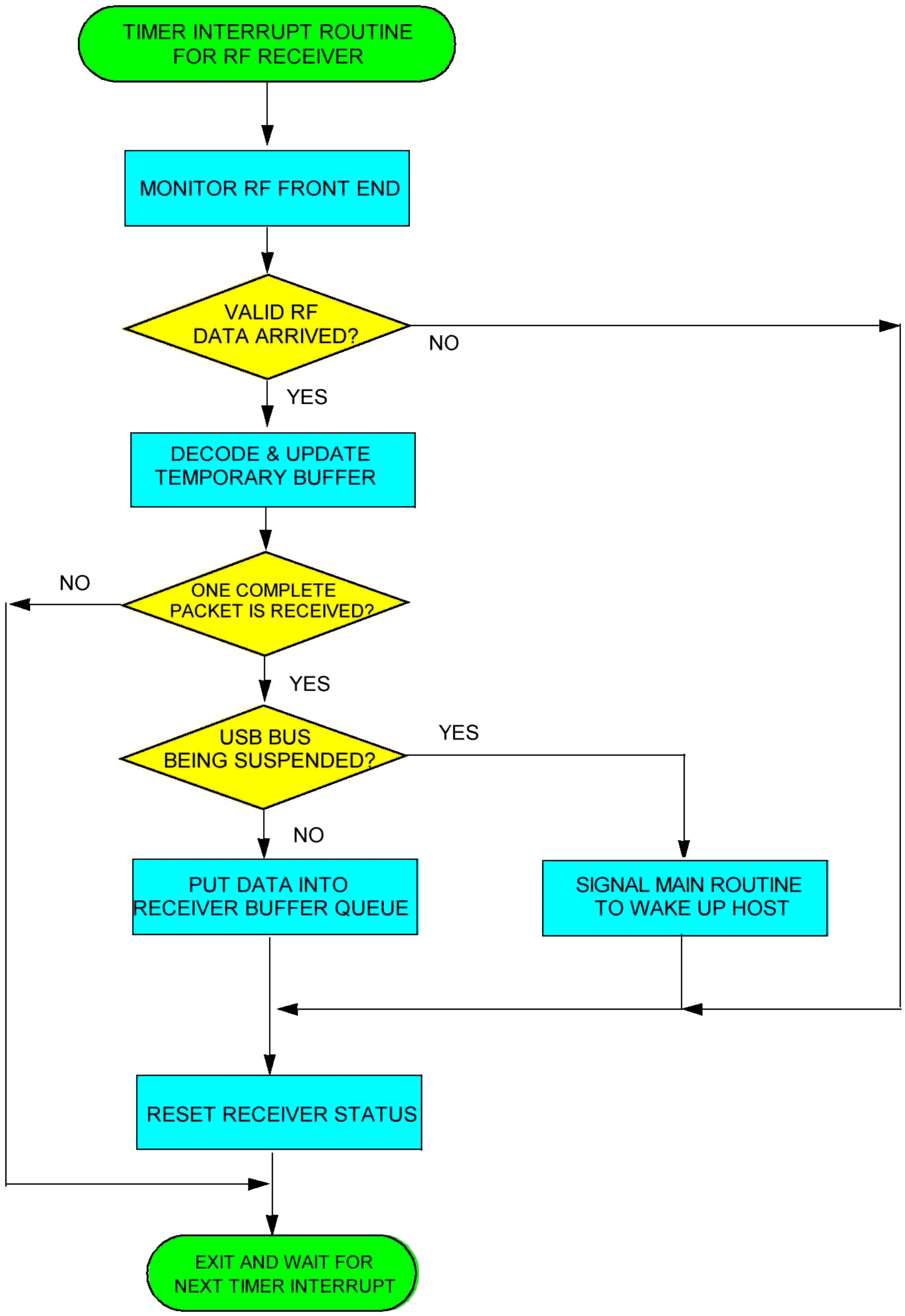


Figure 14. Timer Capture Routine for RF Receiver

USB Report Format

The USB routines in the MC68HC908JB12 act as a composite device of keyboard and mouse. It implements two HID interfaces on endpoint 1 and endpoint 2. HID interface 0 (endpoint 1) implements a standard HID keyboard with identical report and boot protocols. This interface also allows the host system to turn on and off the respective LED state indicators by 1-byte output report. HID interface 1(endpoint 2) implements multimedia, power management keys, and mouse data. Reports with ID numbers 1 and 2 in interface 1 have been reserved for keyboard power management keys and multimedia keys for the keyboard protocol. Those endpoint reports for keyboard will not actually send to the host as there will be no keyboard data received by the JB12 MCU. The mouse report uses report ID number 3 in the HID interface 1. The mouse report structure is shown in Table 1.

Remote Wakeup

The MC68HC908JB12 receiver supports remote wakeup functions that can wake up the host computer during USB suspend. During suspend, the MCU will be periodically awakened by the IRQ interrupt. The MCU then turns on the RF front end and detects whether valid mouse RF packets have arrived to wake up the host. This periodic IRQ interrupt signal is generated through the external RC charging and discharging circuit. The MCU initializes this charging and discharging cycle before it enters power saving mode.

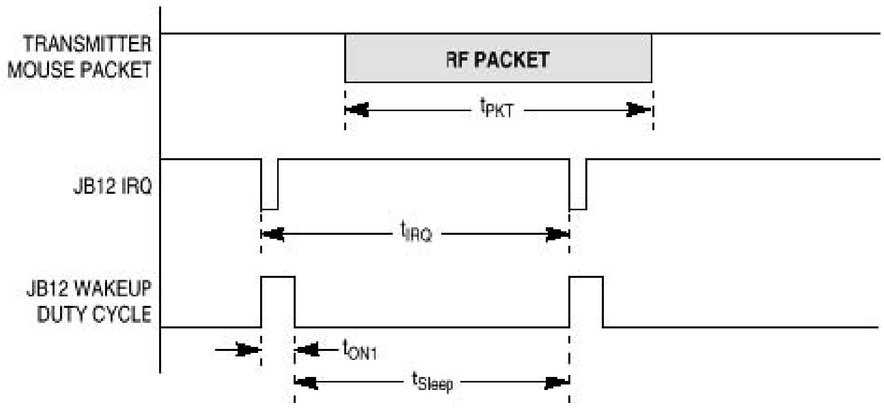
Wakeup Detection Mechanism

After suspend, the MCU will wakeup for a short period of time for each IRQ interrupt. This period is shorter than one complete RF packet. In the case that an RF packet has arrived, the MCU can only determine that a portion of the packet is being received. For this short detection period, there is the possibility that the noise hit into the RF front end would have a pattern like a packet portion. Therefore, if a packet portion is detected, the MCU will turn on for a longer duty cycle. This duty cycle is for receiving the next complete RF packet that can wake up the host.

The mechanism of the remote wakeup is illustrated in the following diagrams.

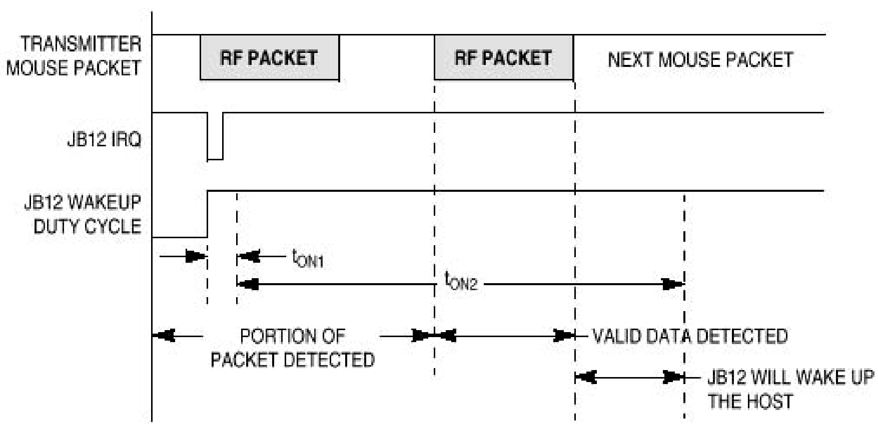
Table 1. USB Mouse Report

Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	Report ID = 3							
1						Middle button	Right button	Left button
2	X displacement							
3	Y displacement							
4	Z displacement							



t_{PKT}: Length of one complete RF packet
t_{IRQ}: Time interval of successive IRQ wakeup periods during device suspend.
t_{ON1}: Duration of MCU being turned on in each IRQ wakeup period. MCU is turned on in this period to detect a (portion of) RF packet has been received.

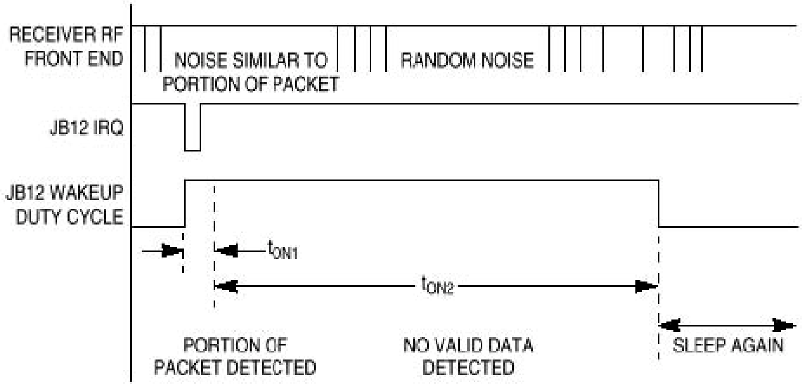
Figure 15. Timing Parameters for Packet Detection



If during a t_{ON1} period a portion of a RF packet is detected, the RF receiver will turn on (t_{ON2}) for a longer time to detect a complete RF packet.

If one complete RF packet is received, the MCU will wake up the host.

Figure 16. Detection of Valid Wakeup Packets



If during a t_{ON1} period a portion of a RF packet is detected, the RF receiver will be turned on (t_{ON2}) for a longer time to detect a complete RF packet.

If the receiver does not receive a complete RF packet, the MCU/receiver will sleep again.

If the receiver receives a complete RF packet, the MCU/receiver will wakeup the host.

Figure 17. Rejecting Packet-Like Noise to prevent false wakeup

ID Updating Process

The user must follow these steps to update the mouse ID:

1. Press and hold the “PTA0” button in the receiver
2. Press “CONNECT” button once in the mouse
3. Release the “PTA0” button in the receiver

Once these steps are complete, the new mouse ID will be updated in the receiver.

Appendix A: Schematic Design of Overall Circuit

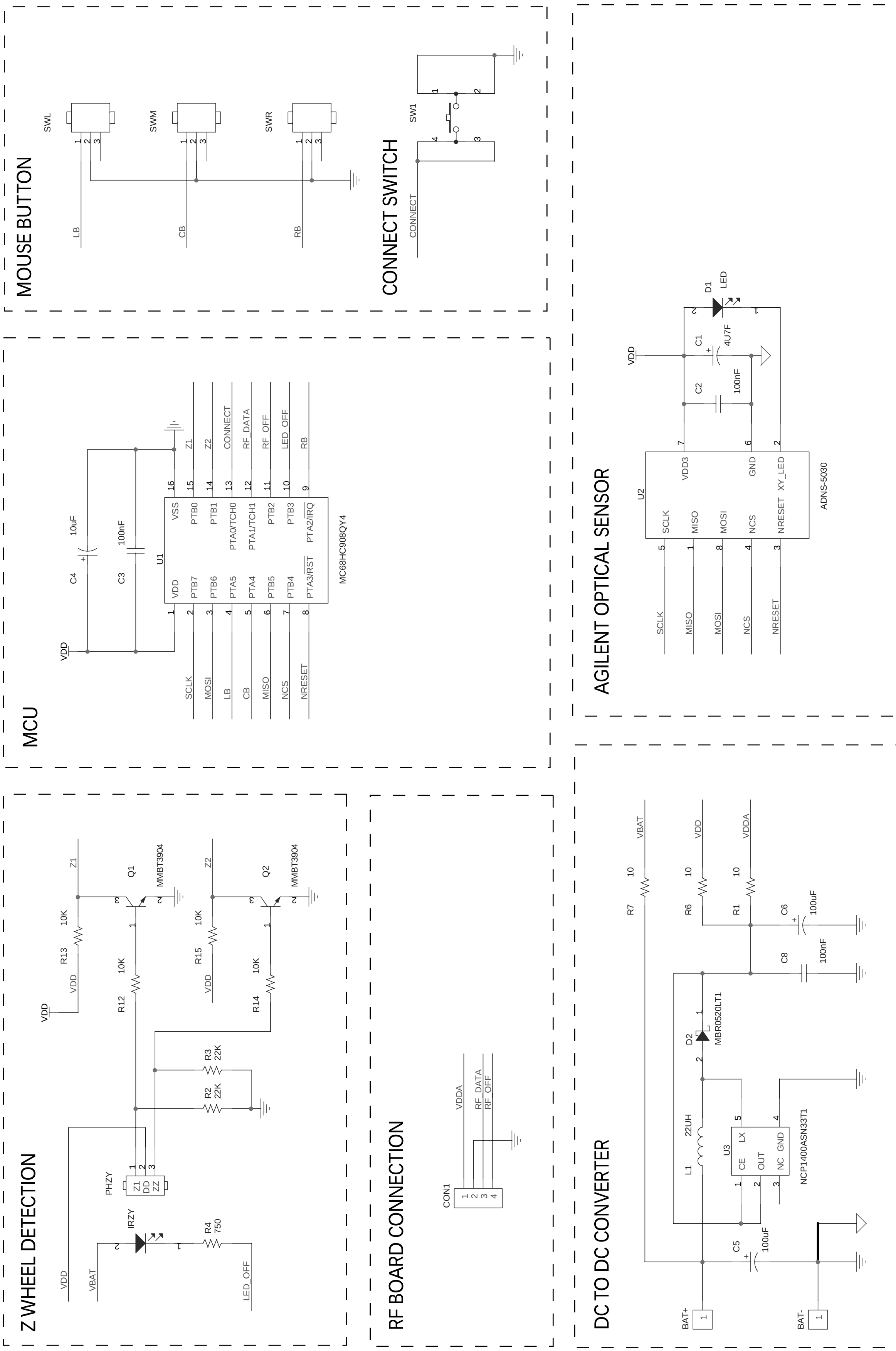


Figure A1. Schematics of the MCU and Optical Sensor PCB Board

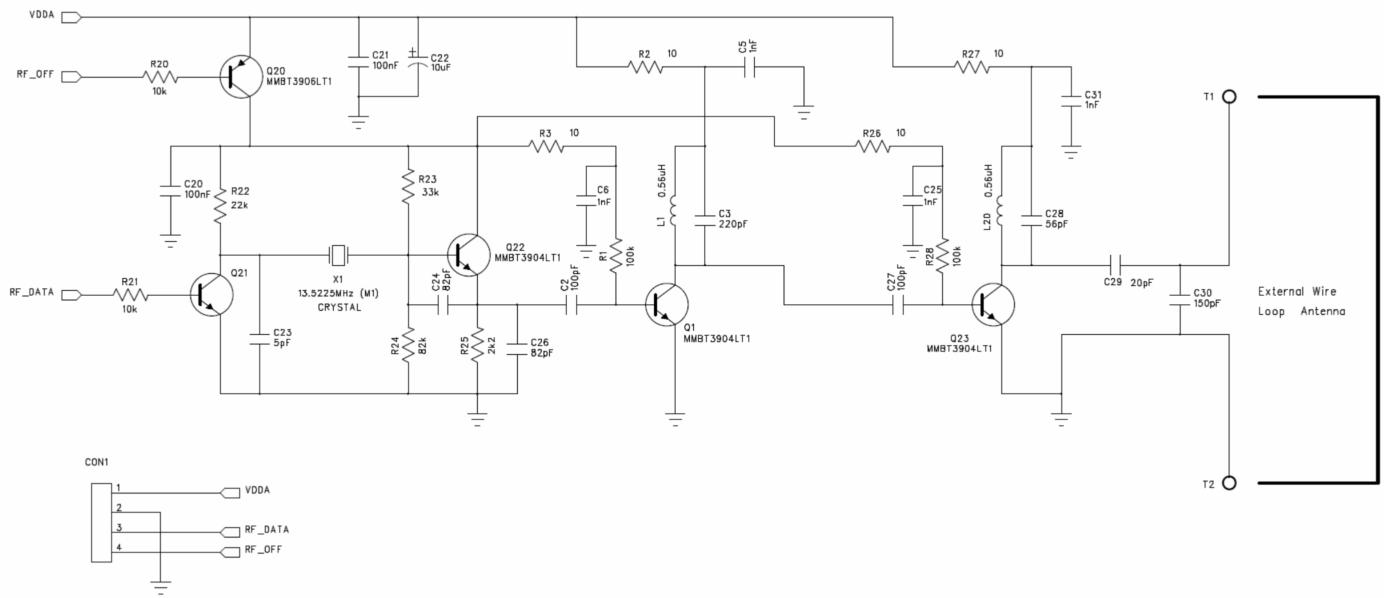


Figure A2. Schematics of RF Board

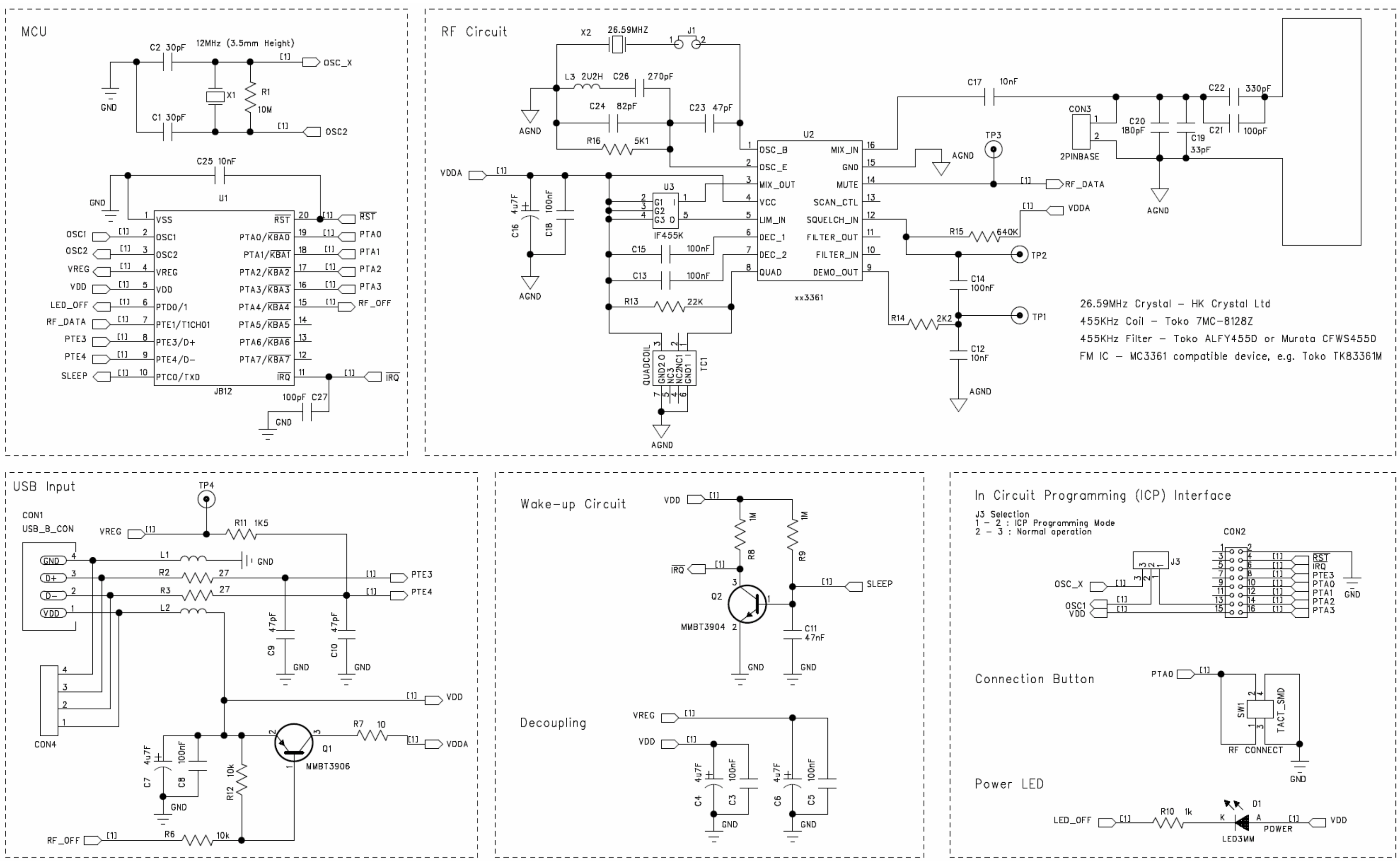


Figure A3. Schematics of USB Receiver Board

APPENDIX B : Bill Of Materials

Table B1: Mouse Main board

Part Description	Quantity	Value	Designators
SMD RES (0603)	3	10	R1 R6-7
SMD RES (0603)	4	10K	R12-15
SMD RES (0603)	2	22K	R2-3
SMD RES (0603)	1	750	R4
SMD CER CAP (0603)	3	100nF	C2-3 C8
SMD TANT CAP	1	4u7F	C1
SMD TANT CAP	1	10uF	C4
ECAP (mini size)	2	100uF	C5-6
SMD NPN TRANSISTOR	2	MMBT3904LT1	Q1-2
IC Socket	1	IC Socket DIP16	U1
SMD TACT SW	1	5mm x 10mm	SW1
DC/DC Converter (3V3)	1	NCP1400ASN33T1 (Onsemi)	U3
SMD DIODE	1	MBR0520LT1 (Onsemi)	D1
SMD INDUCTOR	1	22uH	L1
BUTTON	2	3-Pin (5mmx12mm) 3.4mm pin length	SWL1 SWR1
4-PIN BASE	1	1x4-pin (F), 2mm pitch	CON1
PLASTIC BASE	1	5mmx12mmx6mm	SWM1
BUTTON (long pin length)	1	3-Pin (5mmx12mm) 12mm pin length	SWM1
BATTERY PIN Positive	1	10mm x 12mm	BAT+
BATTERY PIN Negative	1	10mm x 12mm	BAT-
IR DIODE	1	2PIN	IRZY
IR SENSOR	1	3 PIN	PHZY
Wheel	1	20mm	For Z-Wheel
Spring	1	12mm	For Z-Wheel
Case Top Cover	1	N/A	Top
Case Bottom Base	1	N/A	Bottom
Screw	2	8mm	For Case
Screw	2	10mm	For Case
Mini Screw	2	3mm	For PCB Board
MCU	1	MC68HC908QY4A	U1
Optical Sensor Device	1	ADNS-5030	U2
Optical Sensor LED	1	HLMP-ED80	LED1
Optical Sensor Clip	1	ADNS-5200	
Optical Sensor Lens	1	ADNS-5100	N/A

Table B2: Bill Of Material for RF board

Part Description	Quantity	Value	Designators
SMD RES (0603)	4	10	R2-3 R26-27
SMD RES (0603)	2	100k	R1 R28
SMD RES (0603)	2	10k	R20-21
SMD RES (0603)	1	22k	R22
SMD RES (0603)	1	2k2	R25
SMD RES (0603)	1	33k	R23
SMD RES (0603)	1	82k	R24
SMD RES (0603)	5	Open	R4-6 R42-43
SMD CER CAP (0603)	2	100nF	C20-21
SMD CER CAP (0603)	2	100pF	C2 C27
SMD CER CAP (0603)	1	150pF	C30
SMD CER CAP (0603)	4	1nF	C5-6 C25 C31
SMD CER CAP (0603)	1	20pF	C29
SMD CER CAP (0603)	1	220pF	C3
SMD CER CAP (0603)	1	56pF	C28
SMD CER CAP (0603)	1	5pF	C23
SMD CER CAP (0603)	2	82pF	C24 C26
SMD CER CAP (0603)	1	Open	C1
SMD TANT CAP	1	10uF	C22
INDUCTOR	2	0.56uH "0.5 AXIAL"	L1 L20
SMD NPN TRANSISTOR	4	MMBT3904LT1	Q1 Q21-23
SMD PNP TRANSISTOR	1	MMBT3906LT1	Q20
Switch	1	Open	SW1
Jumper Wire	1	Jumper Wire (5mm)	X2
4-PIN HEADER	1	1x4-pin (M), 2mm pitch, 15mm length	CON1
Wire Loop Antenna	1	Wire (length = 14cm, AWG22)	T1-2
CRYSTAL (see note 1)	1	13.5225MHz (3.5mm Height)	X1

Table B3: Bill Of Materials for Receiver Board

Part Description	Quantity	Value	Designators
SMD RES (0603)	1	10	R7
SMD RES (0603)	1	10M	R1
SMD RES (0603)	2	10k	R6 R12
SMD RES (0603)	2	1M	R8-9
SMD RES (0603)	1	1k	R10
SMD RES (0603)	1	1k5	R11
SMD RES (0603)	2	27	R2-3
SMD RES (0603)	1	22k	R13
SMD RES (0603)	1	2k2	R14
SMD RES (0603)	3	open	R4-5 R17
SMD RES (0603)	1	5k1	R16
SMD RES (0603)	1	640k	R15
SMD CER CAP (0603)	7	100nF	C3 C5 C8 C13-15 C18
SMD CER CAP (0603)	2	100pF	C21 C27
SMD CER CAP (0603)	3	10nF	C12 C17 C25
SMD CER CAP (0603)	1	180pF	C20
SMD CER CAP (0603)	1	270pF	C26
SMD CER CAP (0603)	2	30pF	C1-2
SMD CER CAP (0603)	1	330pF	C22
SMD CER CAP (0603)	1	33pF	C19
SMD CER CAP (0603)	1	47nF	C11
SMD CER CAP (0603)	3	47pF	C9-10 C23
SMD CER CAP (0603)	1	82pF	C24
SMD TANT CAP	4	4u7F	C4 C6-7 C16
FERRITE INDUCTOR	2	(0.5" AXIAL Type)	L1-2
INDUCTOR	1	2u2H (0.5" AXIAL Type)	L3
SMD NPN Transistor	1	MMBT3904LT1	Q2
SMD PNP Transistor	1	MMBT3906LT1	Q1
LED	1	GREEN	D1
Crystal	1	12MHz (3.5mm Height)	X1
Crystal	1	26.59MHz (3.5mm Height)	X2
Crystal	1	open	X3
Jumper Wire	1	2.5mm	J1
Jumper Wire	1	open	J2
Jumper Wire	1	2.5mm	J3 (short pin 2-3)
USB CONNECTOR	1	USB B" Receptacle"	CON1
USB CABLE	1	USB A to B Type (Length=1.5 or 2m)	CON1
2-PIN HEADER	1	1x2 header pin (2.5mm pitch)	CON3
4-PIN HEADER	1	1x4 header pin (2.5mm pitch)	CON4
16-PIN HEADER	1	2x8 header pin (2.5mm pitch)	CON2
Test Pin	4	1x1 header pin	TP1-4
SMD TACT SW	1	5mm x 10mm	SW1
Screw	4	5mm length	H1-4
Brass Post	4	20mm height	H1-4
MCU	1	MC68HC908JB12JDW (SO20)	U1
FM IC	1	Toko TK83361M (SO16)	U2
455KHz QUAD COIL	1	Toko 7MC-8128Z	TC1
455KHz IF Filter	1	Toko ALFY455D	U3

APPENDIX C: PCB Layout

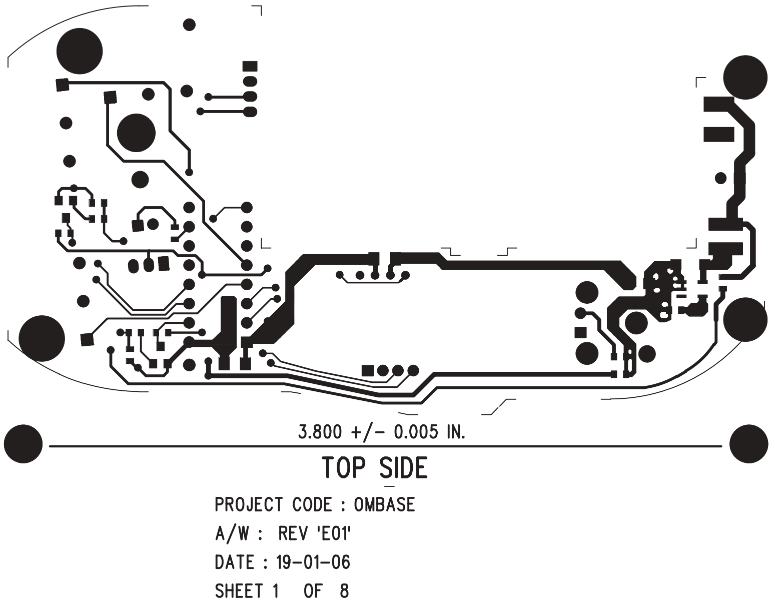


Figure C1. Top Side PCB Layout (Mouse Main Board)

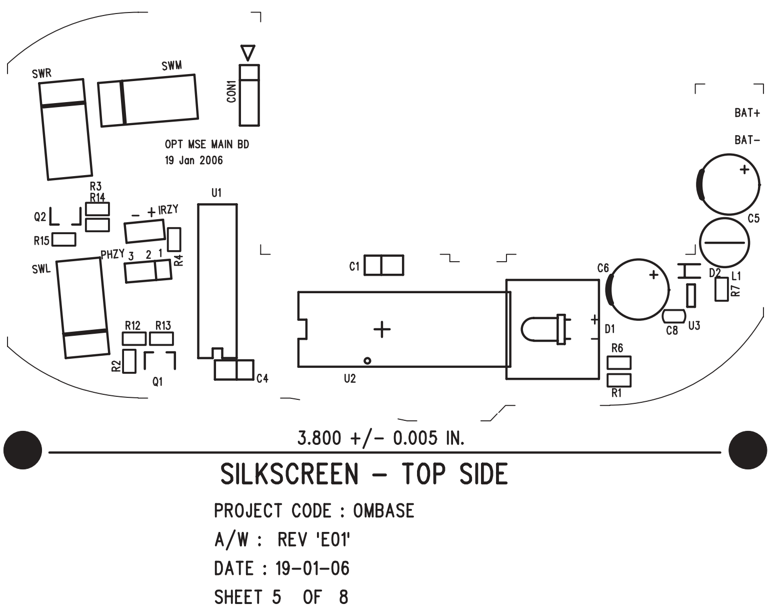


Figure C2. Top Side Components (Mouse Main Board)

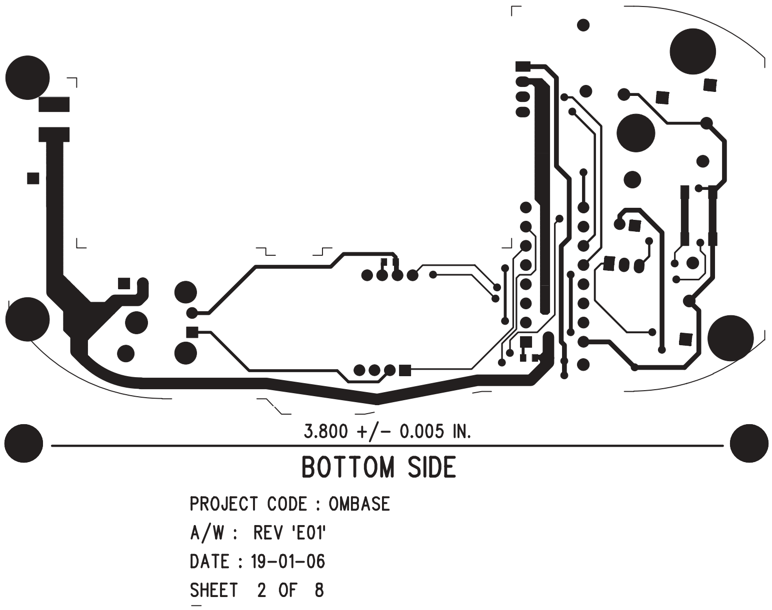


Figure C3. Bottom Side PCB Layout (Mouse Main Board)

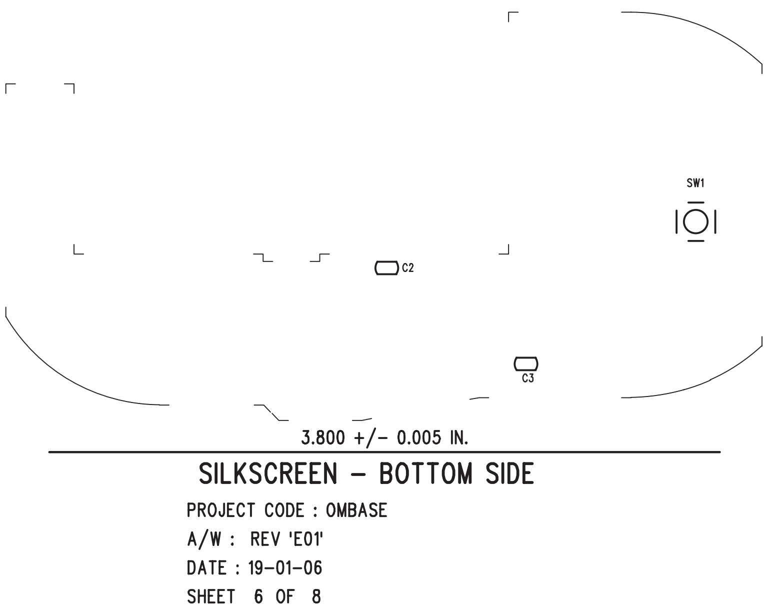


Figure C4. Bottom Side Components (Mouse Main Board)

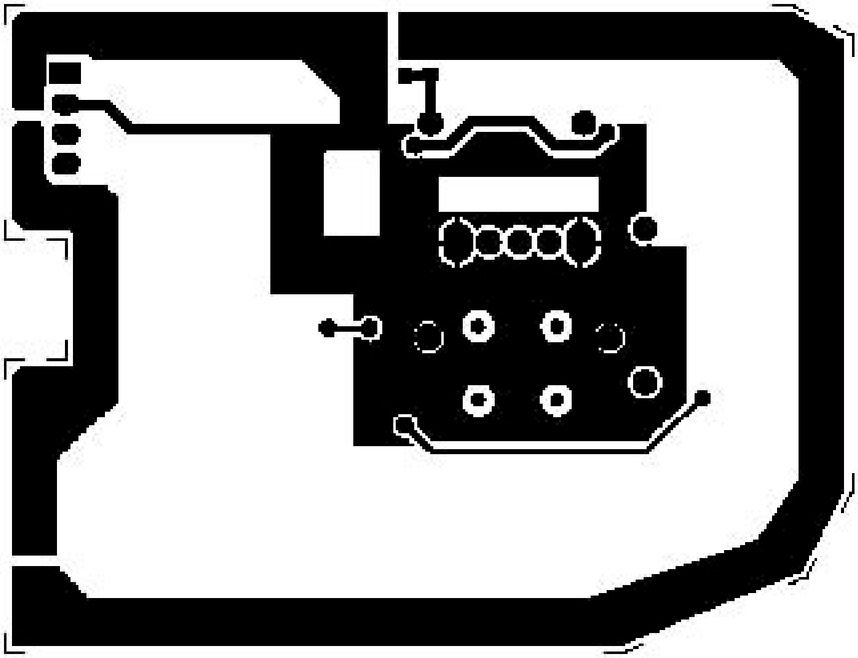


Figure C5. Optical Mouse RF Board Top Layer

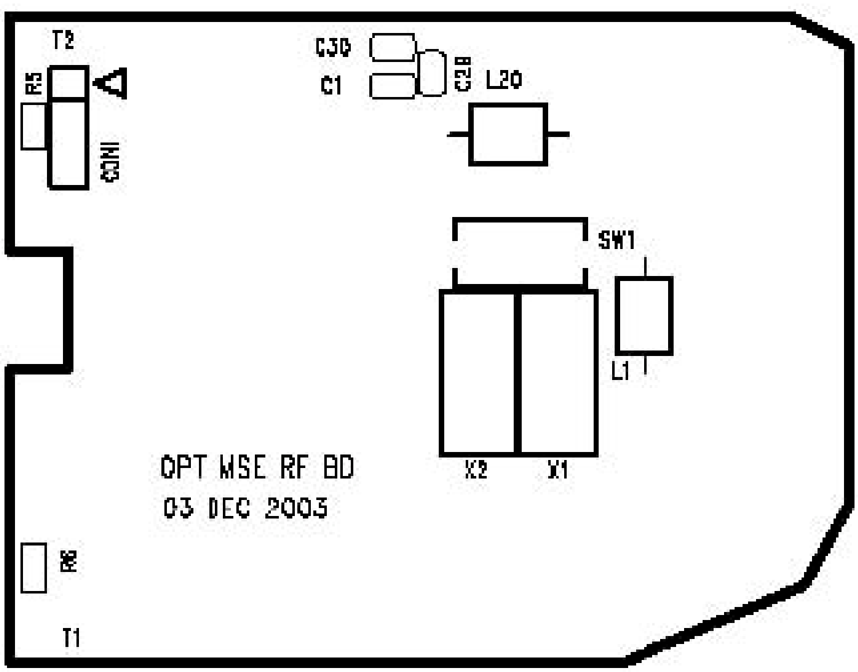


Figure C6. Optical Mouse RF Board Top Overlay

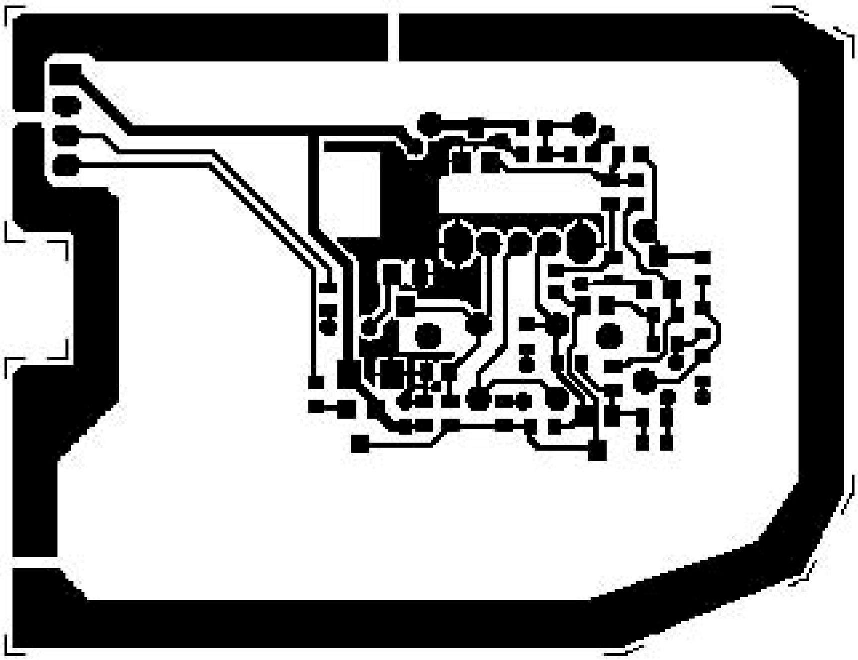


Figure C7. Optical Mouse RF Board Bottom Layer

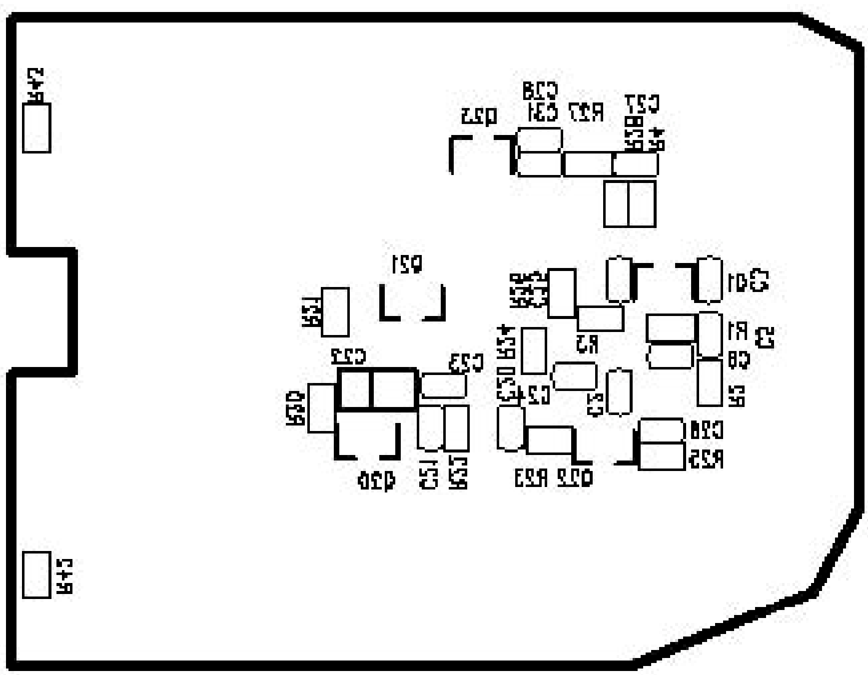


Figure C8. Optical Mouse RF Board Bottom Overlay

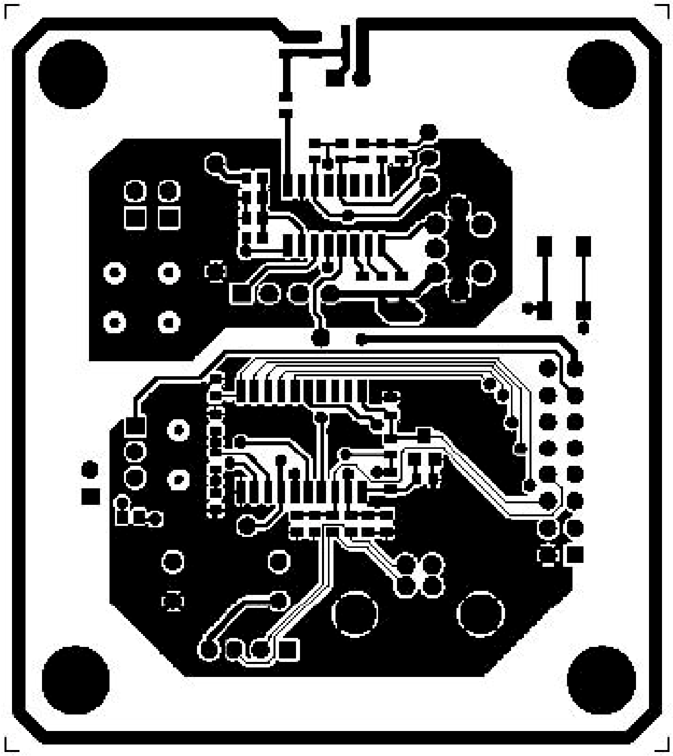


Figure C9. USB Receiver Board Top Layer

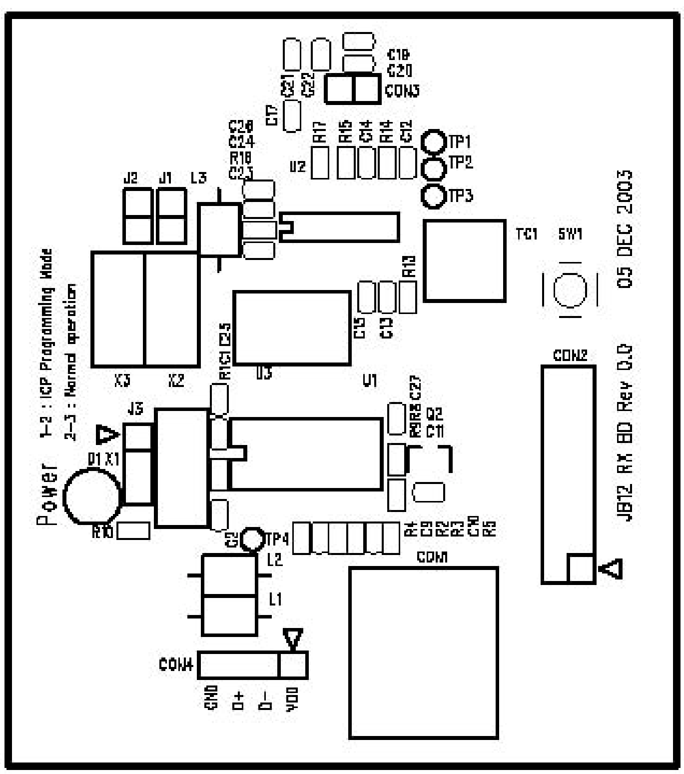


Figure C10. USB Receiver Board Top Overlay

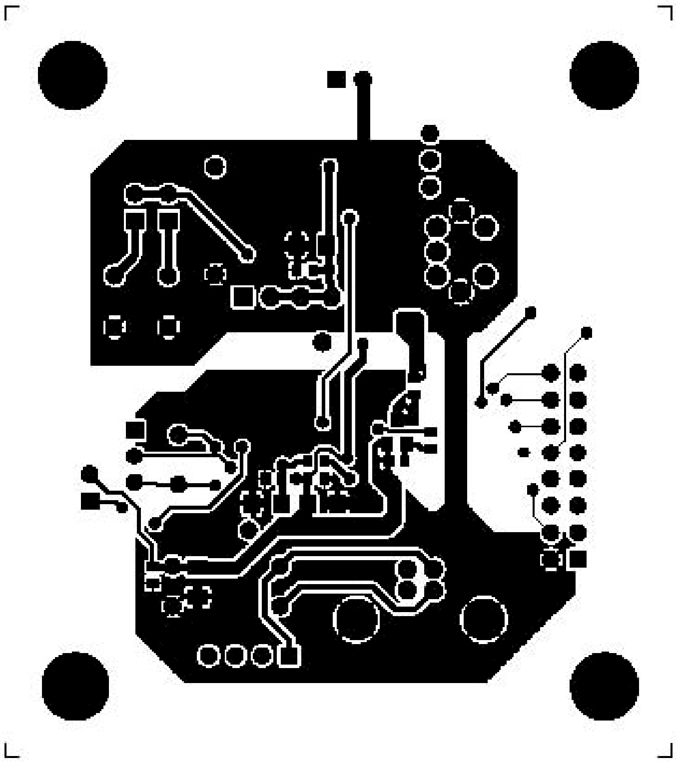


Figure C11. USB Receiver Board Bottom Layer

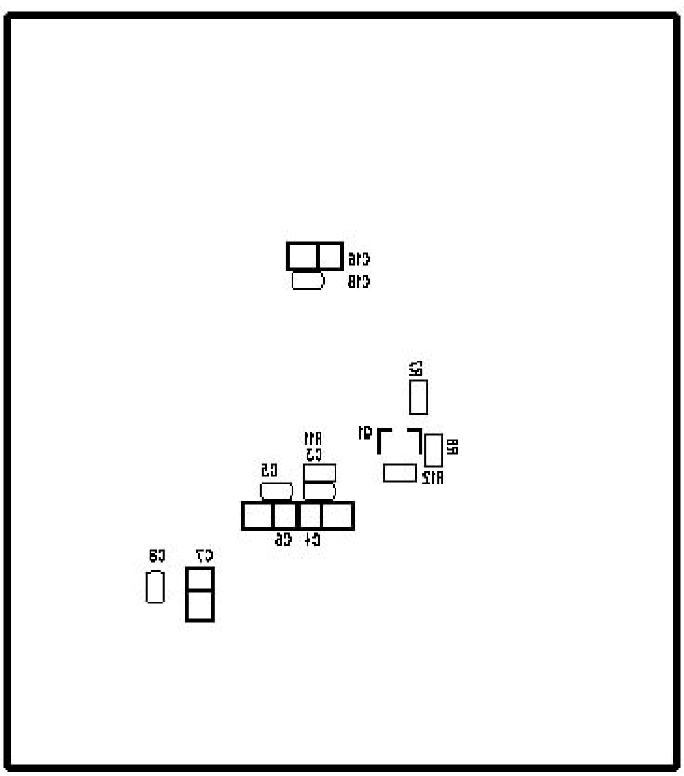


Figure C12. USB Receiver Board Bottom Overlay

Appendix H: Kit Components

The designer's kit contains components as follows:

Part Number	Description	Name	Quantity
ADNK-5033-FS27 Mouse Set	Optical Mouse Reference Design Unit includes: - RF Wireless mouse - USB Receiver Board	Reference Design Unit	1
ADNS-5030	Solid-State Optical Mouse Sensor	Sensor	5
MC68HC908JB12	Freescale USB Microcontroller	USB Microcontroller	3
MC68HC908QY4A	Freescale low-cost Microcontroller	Microcontroller	3
ADNS-5100	Round Lens Plate	Lens	5
ADNS-5100-001	Trim Lens Plate	Lens	5
ADNS-5200	LED Assembly Clip (Black)	LED Clip	5
HLMP-ED80-XX000	639 nm T 1 ¾ (5 mm) Diameter LED	LED	5
ADNK-5033-FS27 CD	Includes Documentation and Support Files for ADNK-5033-FS27 Documentation a. ADNS-5030 Data Sheet b. MC68HC908JB12 & MC68HC908QY4A Data Sheet c. ADNS-5100 Data Sheet d. ADNS-5200 Data Sheet e. HLMP-ED80-XX000 LED Data Sheet Hardware Support Files a. ADNK-5033-FS27 BOM List b. ADNK-5033-FS27 Schematic c. IGES Base Plate Feature File Software Support Files a. Microcontroller Firmware		1
CodeWarrior CD	Includes Software Development Studio for Freescale MC68HC08 Microcontroller a. CodeWarrior Debugger Software b. CodeWarrior Assembler / Compiler Software c. Code Examples d. User Manual		1

www.freescale.com/semiconductors

For product information and a complete list of distributors, please go to our web site.

For technical assistance call:

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

AV01-0003EN - July 4, 2006