MOTOROLA
intelligence everywhere™

digital dna™

USB Wireless Optical Mouse and Multimedia Keyboard Solution

Designer Reference Manual

M68HC08
Microcontrollers

DRM042/D
Rev. 0, 06/2003

MOTOROLA.COM/SEMICONDUCTORS

USB Wireless Optical Mouse and Multimedia Keyboard Solution

Designer Reference Manual — Rev 0

by: Dennis Lui
Derek Lau
Ernest Chan
Applications Engineering
Microcontroller Division
Hong Kong

Section 1. Table of Contents

Section 1. System Overview

Section 2. RF Front End

Section 3. Optical Mouse Transmitter

Section 4. Mouse Transmitter Firmware

Section 5. Multimedia Keyboard Transmitter

Section 6. Keyboard Transmitter Firmware

Section 7. USB Receiver

Section 8. Receiver Firmware

Section 9. Testing and Customization

Section 1. System Overview

1.1 Contents

1.2 Introduction

This manual describes a reference design of a 27MHz Universal Serial Bus Wireless Optical Mouse and Multimedia Keyboard solution by using the MC68HC908QY4, the MC68HC908JB8 and the MC68HC908JB16.

The whole system consists of a wireless mouse, a wireless keyboard and a USB receiver with two RF channels. All hardware schematic diagrams and firmware source codes are available as reference materials.

1.3 Features

- Two independent 27MHz RF links.
- Windows 98, Windows 2000 and Windows XP Compatible
- User Identity Code to avoid conflict with other devices
- USB 2.0 Low Speed Compliance
- 2.5 kbps transmission data rate
- 2 to 3 meter communication distance
- 3361 compatible device for RF receiver design

1.4 System Overview

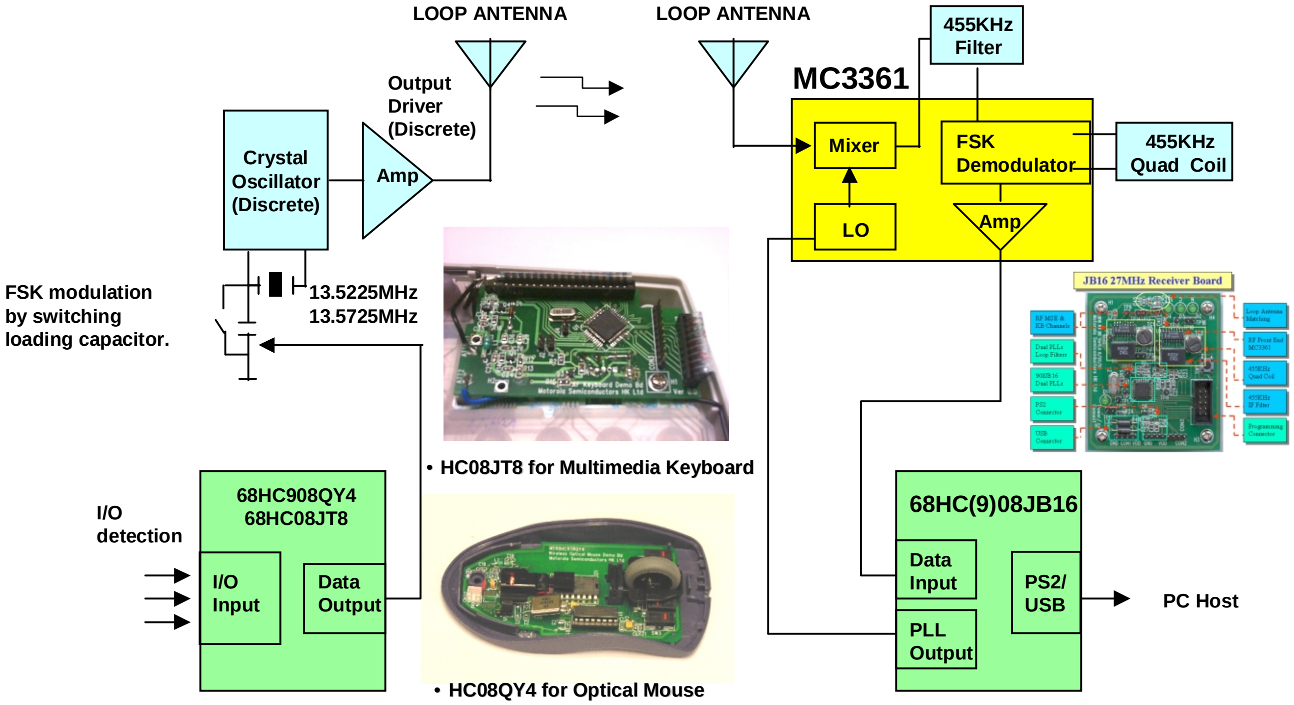


Figure 1-1. System Overview

The system consists of a wireless optical mouse using the MC68HC908QY4 (hereafter referred as QY4), a wireless multimedia keyboard using the MC68HC908JB8 (hereafter referred as JB8, production version uses the low voltage MC68HC08JT8), and the wireless receiver using the MC68HC908JB16 (hereafter referred as JB16).

The QY4 is chosen as the mouse transmitter because it includes an internal oscillator circuit and has the auto wakeup function. The JB8 is specially design for keyboard applications. The JB16 has built-in dual software configurable PLL generators with tunable range from 26MHz to 28MHz. This makes the whole system eliminate up to 5 external crystals (save one in mouse MCU, two or four crystals in the receiver RF circuitry) in compare with traditional solutions.

1.5 Transmit and Receive

In transmitter side, the data generated from key matrix in keyboard or displacement detection / button status data in mouse application is encoded with a pre-defined serial type protocol handled by firmware in MCU. The encoded data is used for FSK modulation in RF stage.

In receiver side, the captured data from RF receiver stage is decoded with corresponding packet format used for keyboard or mouse application. The final data is sent to the host through the USB interface.

Section 2. RF Front End

2.1 Contents

2.2 Introduction

Two 27MHz independent RF links are designed as the wireless communication media for this application. The circuits of the RF font end of the mouse and keyboard transmitters are the same except with different frequencies used. The channel to channel spacing is 100KHz and the transmission data rate is 2.5 Kbps.

2.3 Functional Description

For each RF channel, the high frequency carrier signal in the transmitter side is modulated by the digital encoded data from the QY4 in mouse and the JB8 in keyboard using FSK modulation scheme. The modulated RF signal is propagated through free-air space and received by an integrated chip, the 3361, in the receiver side which includes all mixer, local oscillator and demodulator circuits. The demodulated data output is received by the JB16 for decoding and processing. The data will then be converted to the USB mouse or keyboard report format and sent to the host.

2.4 RF Transmitter

The RF transmitter consists of three parts of circuits — the crystal type oscillator, the RF amplifier and the FSK modulation switching circuit.

The crystal oscillator is working 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 13.5225MHz crystal is used for frequency channel at 27.045MHz.

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 frequency deviation is around ±2.5KHz which is controlled by the crystal characteristics.

2.5 RF Receiver

The RF Receiver is implemented by using a single chip solution (3361 compatible part) which includes a frequency downward conversion mixer, a local oscillator circuit, and 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 with around 15kHz bandwidth. However, the image frequency component would not be filtered out by the IF filter, so it should be considered in PCB layout to prevent any noise component at image frequency injected into the mixer input, such as noise pattern generated from MCU.

The frequency source used for each 3361 mixer local oscillator input is generated from the on-chip PLL of the JB16. Five standard frequency channels in 27MHz band can be configured for each data link by different programming setting in PLL registers. However, the two PLLs should not be programmed with the same frequency. A difference of 50KHz or more is recommended between the two PLL outputs. (see Table 2-1)

Table 2-1. Channel Assignment

Channel	Data Link	Tx freq (MHz)	LO freq (MHz)	JB16 PLL (MHz)	Absolute offset (KHz)
1	Not use	26.995	26.54	26.54166	+1.66
2	Mouse 1	27.045	26.59	26.59171	+1.71
3	Mouse 2 (optional)	27.095	26.64	26.64179	+1.79
4	Keyboard 1	27.145	26.69	26.69189	+1.89
5	Keyboard 2 (optional)	27.195	26.74	26.74172	+1.72

2.6 PCB Layout Guide

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

- The power supply traces used for digital and analog circuit blocks should be separated.
- The location of de-coupling capacitors should be as close as possible to device's supply input pins. (V_{DD}/V_{SS} or V_{CC}/Gnd)
- The V_{DD} to V_{SS} ground loop area should be reduced to minimize the magnetic coupling effect.
- The PCB trace loop formed by any I/O signal pin should be kept minimum.
- RF receiver uses Loop antenna formed by using PCB trace line.

Section 3. Optical Mouse Transmitter

3.1 Contents

3.2 Introduction

The features of the QY4 include an internal oscillator circuit which can generate a clock of 12.8MHz with no external components needed. The auto wakeup module can generate a periodic interrupt during stop mode to wake the part up without requiring an external wake-up circuitry. Those features makes this MCU suited for Wireless Optical Mouse application. The main features of the mouse transmitter include:

- 27MHz RF Transmitter
- 2.5 kbps transmission data rate
- 800 DPI Resolution
- Smart Power Management

3.3 System Overview

The mouse transmitter consists of the QY4, the Agilent optical mouse sensor ADNS-2030 and the RF font end. Figure 3-1 shows the block diagram of the system. RF data is transmitted by means of setting and clearing the RF_Data and the RF_Off pins.

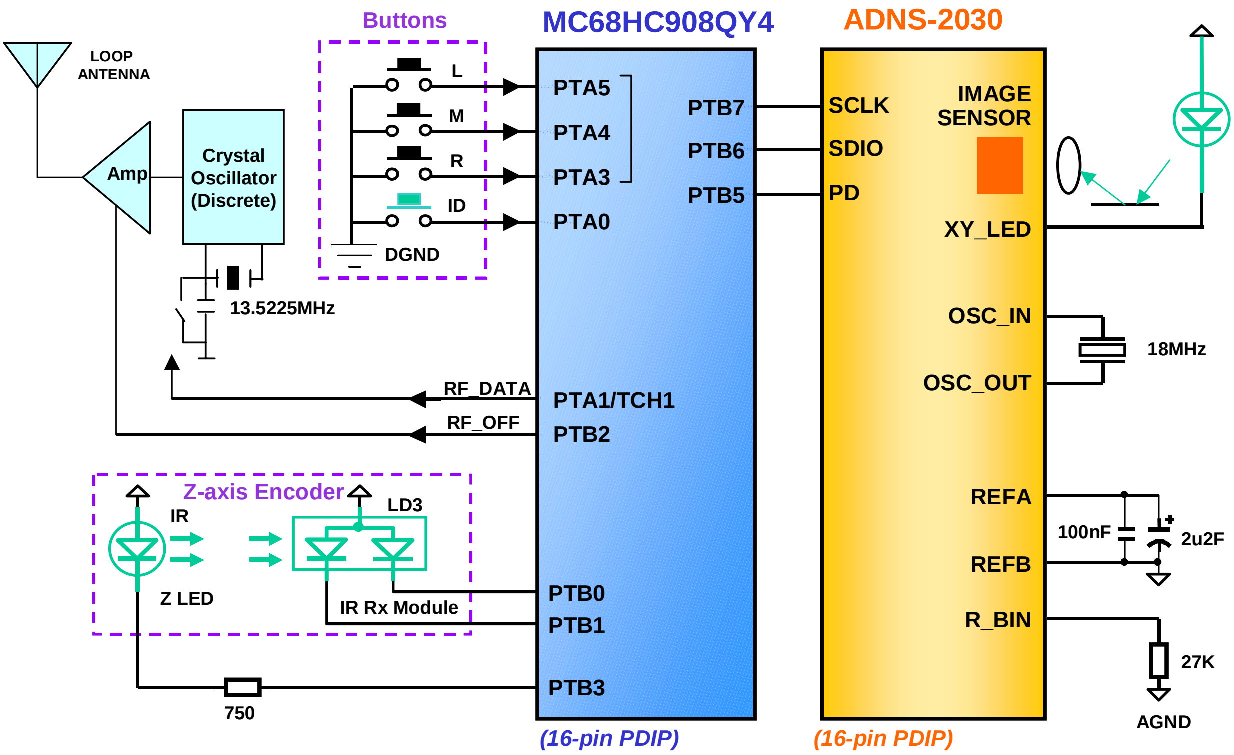


Figure 3-1. Transmitter Block Diagram

3.3.1 Microcontroller QY4

The functions of the QY4 are to get the XY displacement from the sensor, detect the Z displacement, check button status, control the RF circuitry to sending out data, and perform the overall power management.

Three standard left, middle and right buttons together with one button for Identity Device code are implemented. The ID code can either be stored in the RAM or in the FLASH of the QY4. When the ID button in the transmitter and the one in the receiver are pressed, an random ID code is generated at the transmitter and sent to the received. After receiving the new ID code, the receiver stores it in the FLASH of the receiver MCU.

3.3.2 Optical Mouse Sensor

The AN2030 is a 3V supply sensor specially designed for wireless optical mouse. The communication between the QY4 and the sensor is through Serial Peripheral Interface with clock input at the SCLK pin and bi-direction data interface at the SDIO pin. The Power Down (PD) pin is used to power down the sensor when it's not in use.

Section 4. Mouse Transmitter Firmware

4.1 Contents

4.2 Firmware Structure

The firmware structure consists of two main parts:

- Main routine
- Timer interrupt routine

Figure 4-2 shows the flow of the main program and the timer interrupt routine. It also indicates the main functions that the QY4 are to perform. The main challenge in wireless optical mouse design is the power management to minimize the power consumption and maximize the performance.

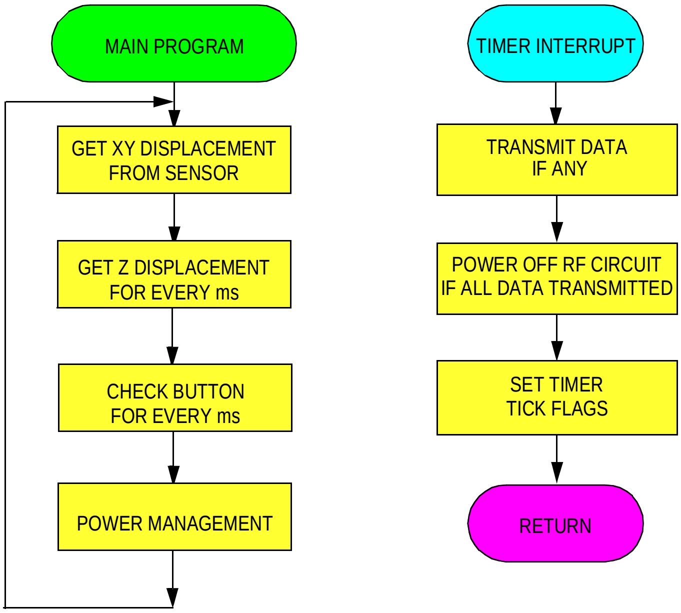

MAIN PROGRAM
GET XY DISPLACEMENT FROM SENSOR
GET Z DISPLACEMENT FOR EVERY ms
CHECK BUTTON FOR EVERY ms
POWER MANAGEMENT
TIMER INTERRUPT
TRANSMIT DATA IF ANY
POWER OFF RF CIRCUIT IF ALL DATA TRANSMITTED
SET TIMER TICK FLAGS
RETURN

Figure 4-1. Firmware Structure

The main program continually checks one of the registers of the 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 buttons status.

Timer interrupt is set for every 200µs which is the base time for the 2.5KHz data rate transmission. By configuring the timer to Output Compare mode, the RF_Data output pin can be set, clear or toggled for every 200µs. The timer interrupt routine determines whether to set or clear the RF_Data pin at the next interrupt time. It also determines what the current RF_Off pin status should be.

4.3 Power Management

Power management plays a very important role in wireless optical mouse solution.

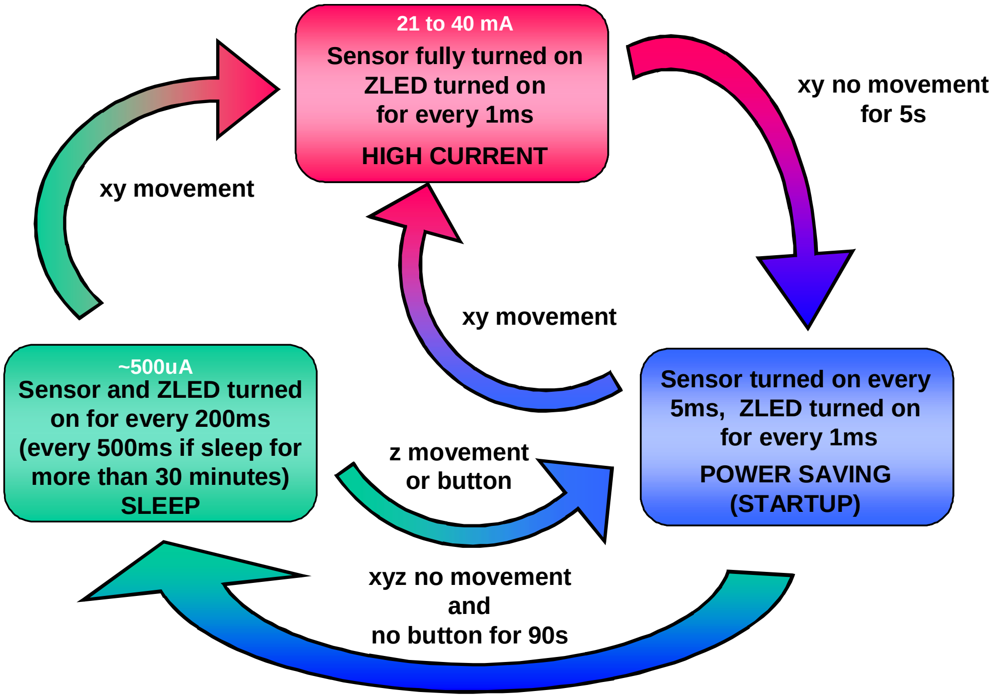


Figure 4-2. Power Management

Figure 4-2 shows the flow of the power management. There are three defined stages — Power Saving, High Current and Sleep stage.

After power up, the mouse is put in Power Saving stage. In this stage, the sensor is only turned on for every 5ms to see if any XY movement happened. The Z movement and buttons are sensed for every ms.

If there is no activities happen for 90 seconds, it enters Sleep stage. In this stage, the QY4 is put in STOP mode and will be waked up for every 200ms to monitor any activities happen. If still no activities happen for 30 minutes, the activities are monitored for every 500 ms.

Any XY movement will cause the mouse to enter High Current stage. At this stage, the sensor is powered on and XY movement is continually monitored. The Z movement and buttons activities are still monitored for every ms. If no XY movement happens for 5 seconds, it then enters Power Saving Stage.

Except for the 1ms timer tick, all of the above mentioned timings can be configured by changing their constant values.

4.4 Data Packet Format

A data packet consists of a Start field, a Data field and a Checksum field.

4.4.1 Coding Method

Except the Start field, both the Data and the Checksum fields are encoded by using the Manchester Coding method. That is, a logic '0' is represented by two equal time 'T' of a logic high of a logic low and vice verse for a logic '1'. An addition '0' is added to the end of each byte of these two field as stop bit.

The order of transmission will be from LSB to MSB, i.e. bit 0 will be transmitted first.

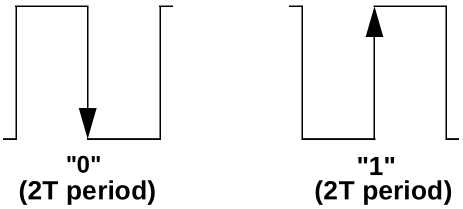


Figure 4-3. Manchester Coding

The basic time of each logic level toggle is 200µs(T). Therefore, each bit in the Data and Checksum fields will be 2T period according to the Manchester Coding.

4.4.2 Packet Types

There are two types of packets:

- X-Y Displacements packet.
- Button Status and Z Displacement packet.

As the header patterns between these two types of packets are different, the receiver will not interpret a X-Y Displacement packet as a Z and Button Status packet or vice versa.

4.4.3 Button status and Z Displacement Packet

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

Figure 4-4. Packet Format for Button & Z Displacement

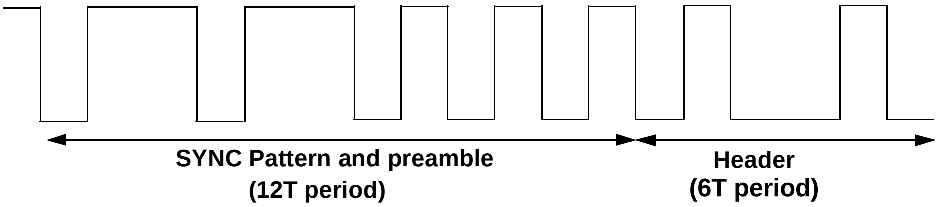


Figure 4-5. Start Field

The Start field consists of the SYNC pattern, preamble and a header as shown in Figure 4-5.

R	ID	M-BTN	R-BTN	L-BTN	R	R	R	STOP

Figure 4-6. Button Status Byte

The Button Status byte represents the status of the four buttons. It shows which buttons are pressed or released. The bit value of '1' means the button was pressed and '0' means the button was released

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

Figure 4-7. Z Displacement or ID Byte

This byte represents either the Z displacement or the new ID code. The Z displacement Byte represents the Z displacement in 2's complement if the ID bit in the Button status equals '0'. If the ID bit equals '1', it represents the new ID code.

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

Figure 4-8. Checksum Byte

The Checksum is the sum of the Button Status byte, the Z Displacement byte and the stored ID byte.

4.4.4 X-Y Displacements Packet:

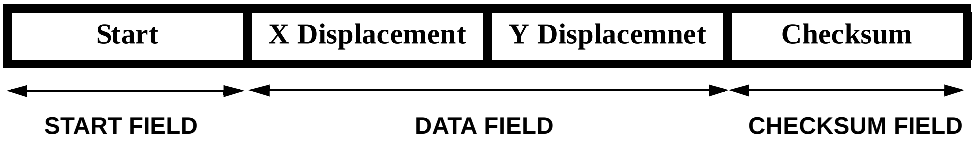


Figure 4-9. Packet Format for X-Y Displacements

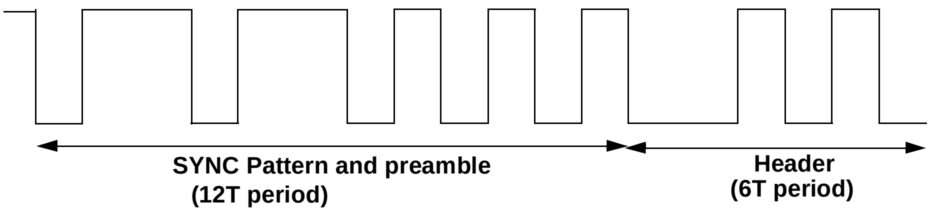


Figure 4-10. Start Field for X-Y Displacement Packet

The X and Y displacements are represented in 2's complement and the Checksum byte is the sum of the X displacement, the Y displacement and the stored ID byte.

Section 5. Multimedia Keyboard Transmitter

5.1 Contents

5.2 Introduction

The JT8 is a low voltage device specially designed for wireless keyboard transmitter applications. With CPU frequency of 2.5MHz , the minimum operating voltage is 2.0V. It contains enough I/O pins with internal pullup resistors for key matrix scanning. The main features of the keyboard include:

- 27MHz RF Transmitter
- 2.5 kbps transmission data rate
- Power management keys (power, wake and sleep) support
- Multimedia keys support

5.3 System Overview

The keyboard transmitter consists of the JB8 and the RF font end circuitry. Figure 5-1 shows the block diagram of the keyboard transmitter.

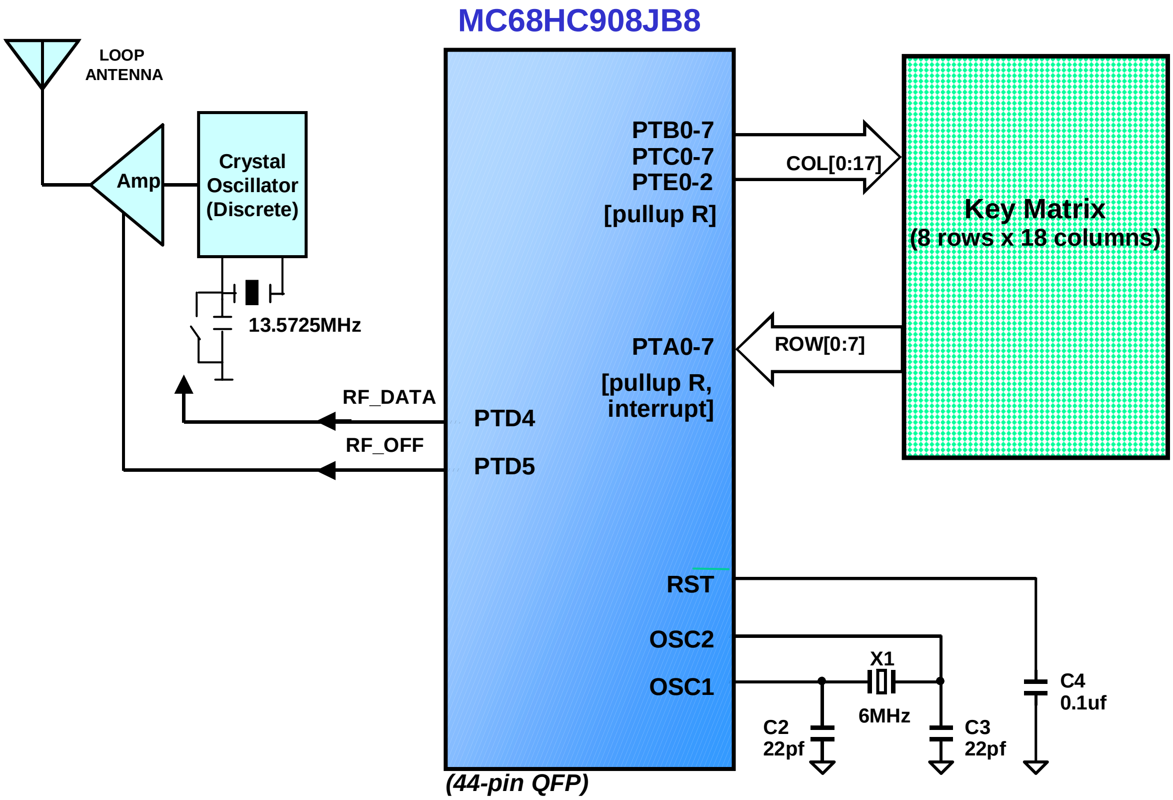


Figure 5-1. Transmitter Block Diagram

The functions of the JB8 are to scan the key matrix and determine what keys are pressed and released. The data is transmitted through RF by setting and clearing the RF_Data and the RF_Off pins.

Section 6. Keyboard Transmitter Firmware

6.1 Contents

6.2 Firmware Structure

Figure 6-1 shows the flow of the main program. The main tasks are to detect any key being pressed or been released and then send the key code through RF signal by setting or clearing the RF_Data pin and the RF_Off pin. If no key is pressed for 5 seconds, it enters power saving mode and the JB8 is put in STOP. It will only be waked up if any key is pressed.

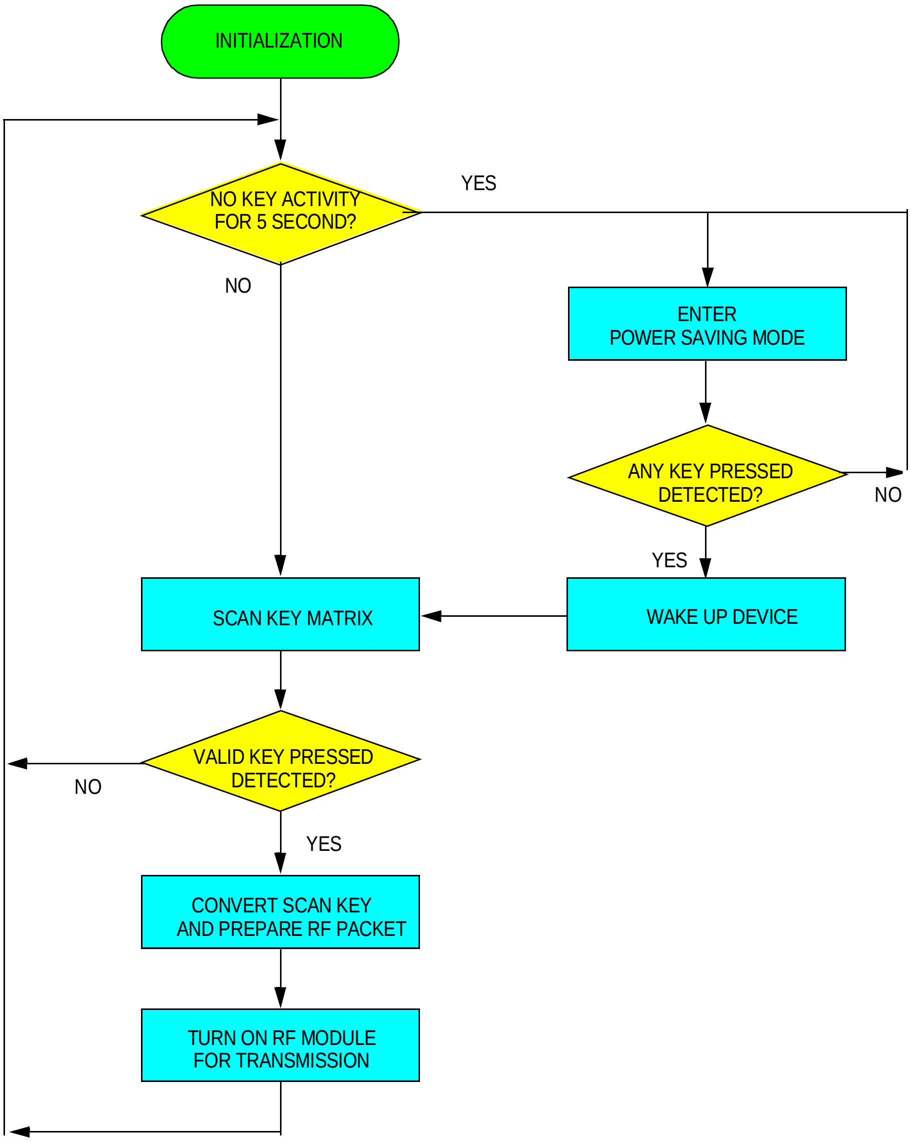

INITIALIZATION
NO KEY ACTIVITY FOR 5 SECOND?
YES
NO
ENTER POWER SAVING MODE
ANY KEY PRESSED DETECTED?
NO
YES
SCAN KEY MATRIX
WAKE UP DEVICE
VALID KEY PRESSED DETECTED?
NO
YES
CONVERT SCAN KEY AND PREPARE RF PACKET
TURN ON RF MODULE FOR TRANSMISSION

Figure 6-1. Firmware Flow

6.3 RF Packet Format

Same as the mouse packet, a keyboard data packet consists of a Start field, a Data field and a Checksum field. Both the Data and the Checksum fields are encoded by using the Manchester Coding method.

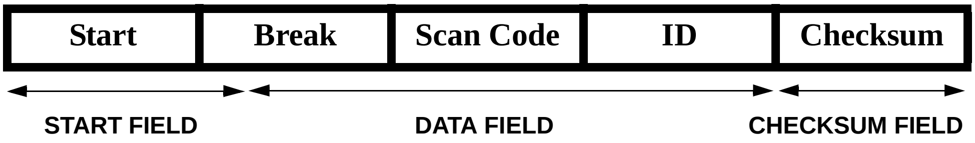


Figure 6-2. Packet Format

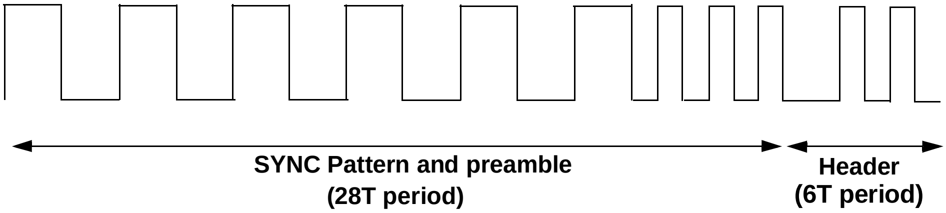


Figure 6-3. Start Field

The Start field consists of the SYNC pattern, preamble and a header as shown in Figure 6-3.

6.3.1 Normal Data Packet

The data packet consists of a Break byte, a Scan Code byte and an ID code byte.

Break	0	1	0	1	0	1	0	STOP

Figure 6-4. Break Byte

The Break byte indicates whether the packet represent a Make or a Break key. The Break bit of '1' means break key (key released) and a '0' means a make key (key pressed).

The scan code byte indicates which key is pressed or released

The ID code represents an ID for a particular Keyboard transmitter and Receiver pair.

The contents of Checksum is the sum of the Break byte, the Scan Code byte and the stored ID byte.

6.3.2 ID PACKET

Each keyboard and receiver have a default ID value of $FF after the firmware has been programmed at the first time. This ID value can be changed to avoid confliction with near-by keyboard-receiver pair using the same frequency channel.

NOTE: *The receiver actually contains two separate ID, one for the keyboard, one for the mouse.*

Dedicate buttons, on both the keyboard transmitter and the receiver, when being pressed, will initiate the ID updating process. An ID packet will be transmitted from the keyboard which contains the new ID. The receiver will recognize this packet and update the new ID value accordingly.

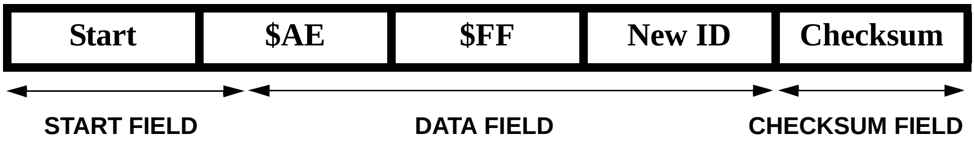


Figure 6-5. ID PACKET

The ID packet format is similar to DATA packet, except that the first byte in the DATA FIELD is always having a value of $AE. The new ID value is randomly generated which can be ranged from $00 to $FF.

6.4 ID Updating Process

To updated the keyboard ID, user have to:

1. Press the PTA0 button in the receiver.

2. Press the “CONNECT” key in the keyboard continuously. The PTD0 LED in receiver should be flashing.

3. Release the “CONNECT” key (the PTD0 LED will stop flashing).

4. The new keyboard ID will be updated in both keyboard and receiver.

Section 7. USB Receiver

7.1 Contents

7.2 Introduction

The features of the JB16 include a configurable Universal Serial Bus (USB) and PS/2 interface, which makes this MCU suited for personal computer Human Interface Devices (HID) applications, such as mice and keyboards. It has built-in configurable dual PLL generators that makes this MCU suited for Wireless HID applications. The enhanced timer functions also allows it to capture and decode data easily. The main features of the receiver include:

- 27MHz RF Transmitter
- Fully USB specification 2.0 low speed compliant
- Windows 98, 2000 and XP compatible

7.3 System Overview

The receiver consists of the JB16, LED indicators for the keyboard Num Lock, Scroll Lock and Caps Lock, a button for ID setting and the RF font end circuitry.

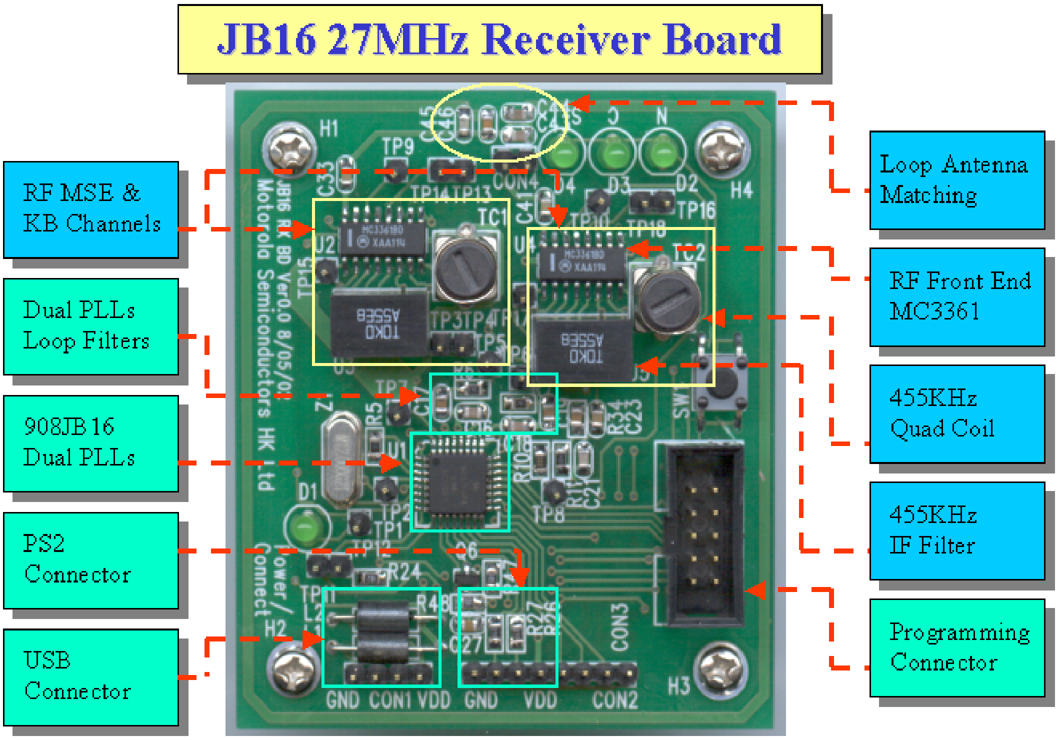


Figure 7-1. USB Receiver

The functions of the JB16 are to handle the USB transactions, capture and process data, and generate two different 27MHz range clock signals for the Demodulators use. The processed data is converted into USB report format and sent to the host.

Section 8. Receiver Firmware

8.1 Contents

8.2 Firmware Structure

The firmware consists of four main parts:

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

Figure 8-1 shows the flow of the main program and one of the timer interrupt routines. The flows of the mouse and keyboard timer interrupt routines are the same. The USB interrupt routine are not showed here. If you are interested, please refer to another reference design called **"**USB and PS2 Multimedia Keyboard Interface Reference Design**"**.

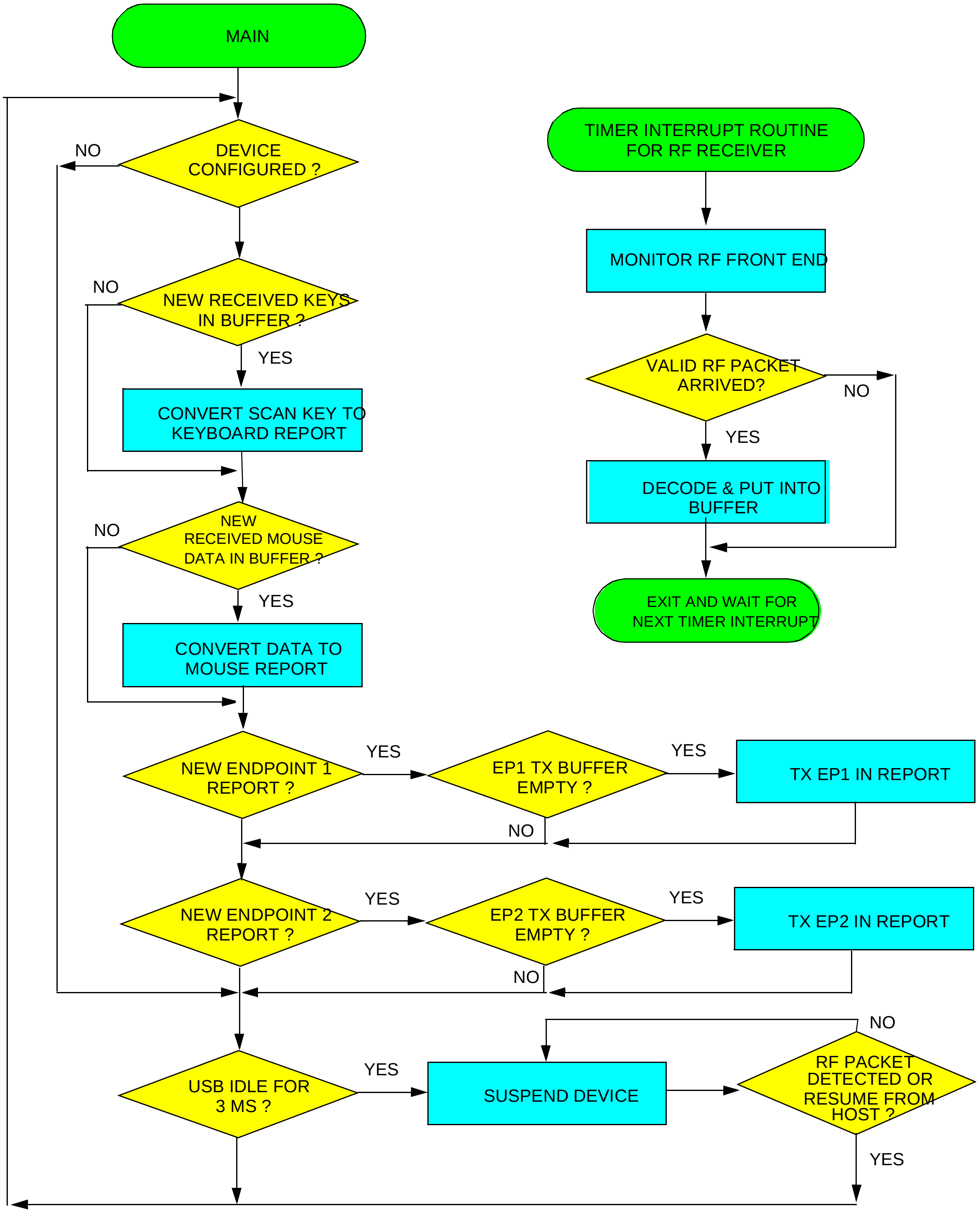


Figure 8-1. Firmware Flow

The main routine continually checks if there is any valid mouse or keyboard data. If so, the data will be converted to USB report format and sent to the host via the endpoint 1 and endpoint 2.

Two timer interrupts routines are used to capture the RF mouse and keyboard data. Timers are set to input capture mode with interrupt happens if falling and rising edge so as to calculate the pulse width of the RF data. The routines will proceed to detect data only if the pulse width match the start field requirements.

8.3 USB Report

The mouse and keyboard implements two HID interfaces on endpoint 1 and endpoint 2 in a USB composite-device fashion. HID interface 0 (endpoint 1) implements a standard HID keyboard with identical report and boot protocols. HID interface 1 (endpoint 2) implements multimedia and power management keys, and mouse data. This implementation ensures the keyboard work in BIOS setup and in DOS mode.

Interface 0 will issue 8-byte input reports that are identical to the standard keyboard boot protocol report (Table 8-1) as documented in the Device Class Definition for Human Interface Device (HID) version 1.1. This interface also allows the host system to turn on and off the respective LED state indicators, as specified by the 1-byte output report. (Table 8-2)

Table 8-1. Interface 0 Input Report

Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	**Right GUI**	**Right ALT**	**Right Shift**	**Right Control**	**Left GUI**	**Left ALT**	**Left Shift**	**Left Control**
1	Reserved							
2	Keyboard Usage ID (Key Code)							
3	Keyboard Usage ID (Key Code)							
4	Keyboard Usage ID (Key Code)							
5	Keyboard Usage ID (Key Code)							
6	Keyboard Usage ID (Key Code)							
7	Keyboard Usage ID (Key Code)							

Table 8-2. Interface 0 Output Report

Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0						Scroll Lock	Caps Lock	Num Lock

Interface 1 report contains power management key report, multimedia key input report or mouse input report, which is distinguished by a unique Report ID. The power management key uses Report ID number 1 (see Table 8-3), the multimedia key uses Report ID number 2 (see Table 8-4), while the mouse report uses Report ID number 3 (see Table 8-5).

Table 8-3. Interface 1 Power Key Input Report

Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	**Report ID = 1**							
1						Power	Wake	Sleep

Table 8-4. Interface 1 Multimedia Key Input Report

Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	Report ID = 2							
1	M7	M6	M5	M4	M3	M2	M1	M0
2	M15	M14	M13	M12	M11	M10	M9	M8
3	M23	M22	M21	M20	M19	M18	M17	M16
4	Reserved for M24 - M31							

Table 8-5. Interface 1 Mouse Input 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							

8.3.1 Input Report Example

Table 8-6 shows some input report examples. Report ID is not used in interface 0. The first byte is the modifier byte and is set on bit base. Whenever a modifier key is pressed the corresponding bit is set to one. For example, if the Left Control and the character 'A' keys are pressed, the first byte of the report equals $01, the second byte is reserved, the third byte equals $04, and the forth to the eighth bytes equal $00.

Power Management keys are reported through interface 1 with report ID 1. For example, if the Wake key is pressed, the first byte equals $01 (ID = 1), and the second byte equals $02 (corresponding bit equals 1) since Wake key is defined as the bit 2 of the second byte

Hot keys are reported through interface 1 with reported ID 2. For example, if the hot key 0 and the hot key 17 are pressed, the first byte equals $02 (ID = 2), the second byte equals $01 since hot key 0 is pressed, the third byte equals $00 since hot keys 8 to 16 are not pressed, and the forth byte equals $02 since the hot key 17 is pressed.

Table 8-6. Input Report Examples

Keys Pressed	Endpoint	In Report Data
Left Control, 'A'	1	$01,$00,$04,$00,$00,$00,$00,$00
Left Control, Right Alt, 'A', 'B'	1	$41,$00,$04,$05,$00,$00,$00,$00
Wake	2	$01,$02
Hot Key 0 & Hot Key 17	2	$02,$01,$00,$02

8.4 Remote Wakeup

The JB16 receiver supports remote wakeup function that can wake up the host computer during USB suspend.

During suspend, the MCU will be periodically waked up by the IRQ interrupt. The MCU then turns on the RF front end and detect whether valid mouse or keyboard RF packets arrived for waking up the host. This periodical IRQ interrupt signal is generated through the external RC charging and discharging circuit. The MCU initiziates this charging and discharging cycle before it enters power saving mode.

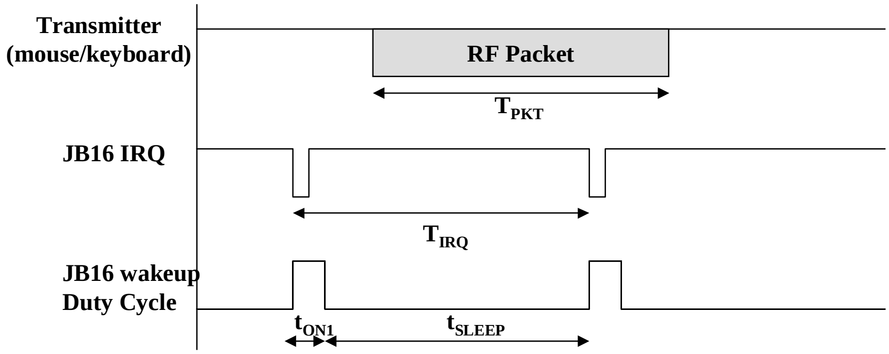
8.4.1 Wakeup Detection Mechanism

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 case a RF packet has arrived, the MCU can only determine a portion of packet is being received.

For this short detection period, it has the possibility that the noise hit into the RF front end having pattern like a portion of packet.

Therefore, if a portion of packet is detected, the MCU will turn on for a longer duty. It is for receiving next complete RF packet that can wake up the host.

The mechanism and the timing parameters for detecting wakeup packet is summarized in below diagrams.

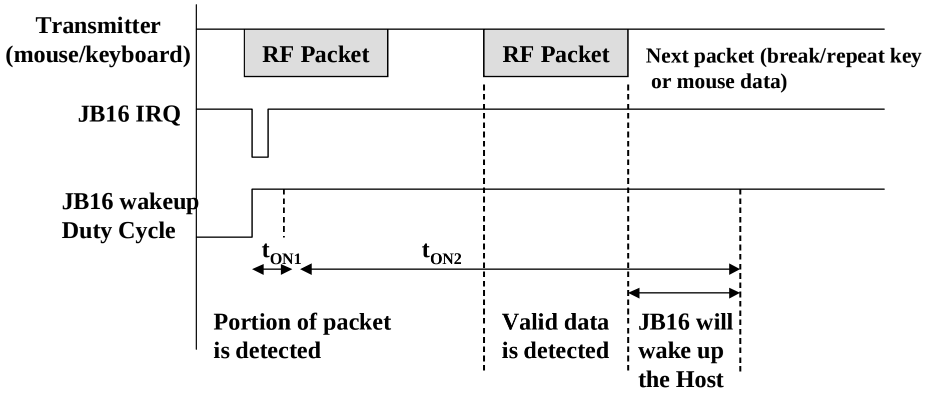


T_{PKT} : Length of one complete RF packet

T_{IRQ} : Time interval of successive IRQ wake up periods during device suspend.

t_{ON1} : Duration of MCU being turned on in each IRQ wake up period. MCU is turned on in this period to detect if a (portion of) RF packet has been received.

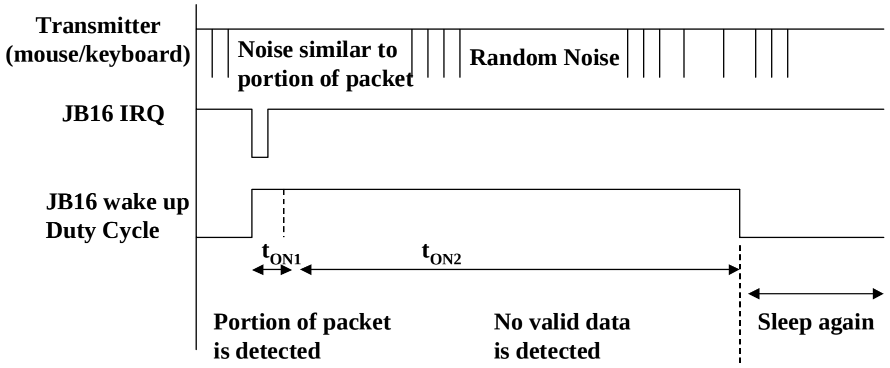
Figure 8-2. Timing Parameters for packet detection



If during a t_{ON1} period, a portion of RF packet is detected, the RF receiver will be 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 8-3. Detection for valid wakeup packets



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

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

Figure 8-4. JB16 rejecting packet like noise to prevent false wakeup

8.4.2 Power Consideration

The average power consumptionn during suspend is given by the following equation:

$$P_{average} = \frac{(P_{ON} \times t_{ON}) + [P_{Sleep} \times (T_{IRQ} - t_{ON})]}{IRQ}$$

Power consumption can be decreased by increasing $T_{IRQ.}$But if T_{IRQ} is larger than T_{PKT}, the receiver may not catch the incoming RF packet. Therefore choosing T_{IRQ} equal to T_{PKT} could be a optimizing value for T_{IRQ}.

Another way for reducing power consumption is to shorten t_{ON}. But t_{ON} cannot be too small in practice. Otherwise, the false wake-up for 2nd stage complete packet detection can occur more frequently and thus more power is actually consumed (refer to figure 8-4). This parameter can be fine tuned in the actual application system.

Section 9. Testing and Customization

9.1 Contents

9.2 Testing

The solution was tested under different Windows Operating Systems on several different PCs.

- USB compliance test using Command Verifier version 1.1
- Compatibility tests under Windows 98SE, 2000 and XP.
- Compatibility tests under AMD 750, Intel 810 and 845 chip set Desktops, and IBM Thinkpad 570, 600E, 600X and T23.

9.3 Customization

9.3.1 Hardware

9.3.1.1 Optical Mouse Transmitter

The step-up regulator LM3352 is for reference only, customers can choose any regulator they prefer. The LEDs for the sensor and Z LED can be connected to the regulator output or connected to the batteries output. The advantage of connecting to the regulator output is that the system can work in a lower voltage, but the drawback is higher current consumption. The advantage of connecting to the batteries output is the lower power consumption but the system will not work properly if battery voltage is below 2.5V.

9.3.1.2 Keyboard Transmitter

Left the unused pin open.

9.3.1.3 Receiver

If the ROM part 68HC08JB16 is used instead of the FLASH part, an external EEPROM is needed for storing the ID code.

9.3.1.4 RF Circuitry

If the PCB loop antenna is changed in term of trace length or width, the corresponding loop antenna inductance L would be different and the matching network should be adjusted to maintain the maximum signal transfer condition.

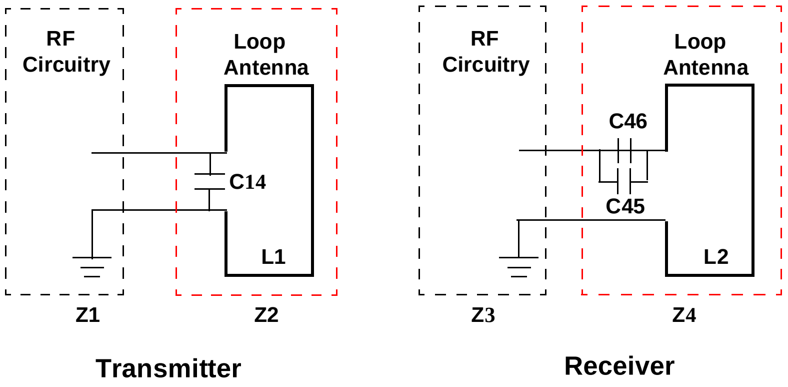


Figure 9-1. Loop Antenna Impedance

In transmitter side, the value for matching component C can be easily calculated by following equation: (f = 27MHz, L = loop antenna inductance)

$$2\pi f = \frac{1}{\sqrt{LC}}$$

Table 9-1. Tx Matching Examples

Case	L1	C = C14
Smaller Inductance	200nH	= 173pF (~180pF)
Original Reference Design	240nH	= 145pF (~150pF)
Larger Inductance	300nH	= 116pF (~120pF)

In receiver side, one of the simple method to keep the matching condition for different inductance value is to adjust the capacitance value C such that the sum of L & C impedance [XL + (- XC)] at 27MHz is equal to a pre-matching impedance value. (Z4 = 26.3j)

$$Z4 = j\omega L + \frac{1}{j\omega C} = 26.3j$$

Table 9-2. Rx Matching Examples

Case	L2	C = C45 + C46
Smaller Inductance	200nH (34j)	765.5pF (-7.7j) ~ (470 + 300) pF
Original Reference Design	236nH (40j)	430pF (-13.7j) = (330 + 100) pF
Larger Inductance	300nH (51j)	238.6pF (-24.7j) ~ (200 + 39) pF

9.3.2 Firmware

9.3.2.1 Mouse Transmitter

- Set compiler option to store the ID code into RAM or into FLASH.
- Set the timing parameters for power management.

9.3.2.2 Keyboard Transmitter

- Set compiler option to store the ID code into RAM or into FLASH.
- Modify the key matrix tables in "KEY-USB.ASM" according to customized key matrix layout.
- The CPU operating frequency is set to 3MHz. The transmitter routine has to be modified if different CPU frequency is used.

9.3.2.3 Receiver

- Change vendor ID, product ID and product revision number in the device descriptor table in "KBD-MSE.H".
- Change the report descriptor in "KBD-MSE.H" if necessary.
- The ID code is designed to use in the FLASH part only.

Section 1. Glossary

A — See “accumulators (A and B or D).”

accumulators (A and B or D) — Two 8-bit (A and B) or one 16-bit (D) general-purpose registers in the CPU. The CPU uses the accumulators to hold operands and results of arithmetic and logic operations.

acquisition mode — A mode of PLL operation with large loop bandwidth. Also see ’tracking mode’.

address bus — The set of wires that the CPU or DMA uses to read and write memory locations.

addressing mode — The way that the CPU determines the operand address for an instruction. The M68HC12 CPU has 15 addressing modes.

ALU — See “arithmetic logic unit (ALU).”

analogue-to-digital converter (ATD) — The ATD module is an 8-channel, multiplexed-input successive-approximation analog-to-digital converter.

arithmetic logic unit (ALU) — The portion of the CPU that contains the logic circuitry to perform arithmetic, logic, and manipulation operations on operands.

asynchronous — Refers to logic circuits and operations that are not synchronized by a common reference signal.

ATD — See “analogue-to-digital converter”.

B — See “accumulators (A and B or D).”

baud rate — The total number of bits transmitted per unit of time.

BCD — See “binary-coded decimal (BCD).”

binary — Relating to the base 2 number system.

binary number system — The base 2 number system, having two digits, 0 and 1. Binary arithmetic is convenient in digital circuit design because digital circuits have two permissible voltage levels, low and high. The binary digits 0 and 1 can be interpreted to correspond to the two digital voltage levels.

binary-coded decimal (BCD) — A notation that uses 4-bit binary numbers to represent the 10 decimal digits and that retains the same positional structure of a decimal number. For example,

234 (decimal) = 0010 0011 0100 (BCD)

bit — A binary digit. A bit has a value of either logic 0 or logic 1.

branch instruction — An instruction that causes the CPU to continue processing at a memory location other than the next sequential address.

break module — The break module allows software to halt program execution at a programmable point in order to enter a background routine.

breakpoint — A number written into the break address registers of the break module. When a number appears on the internal address bus that is the same as the number in the break address registers, the CPU executes the software interrupt instruction (SWI).

break interrupt — A software interrupt caused by the appearance on the internal address bus of the same value that is written in the break address registers.

bus — A set of wires that transfers logic signals.

bus clock — See "CPU clock".

byte — A set of eight bits.

CAN — See "Motorola scalable CAN."

CCR — See “condition code register.”

central processor unit (CPU) — The primary functioning unit of any computer system. The CPU controls the execution of instructions.

CGM — See “clock generator module (CGM).”

clear — To change a bit from logic 1 to logic 0; the opposite of set.

clock — A square wave signal used to synchronize events in a computer.

clock generator module (CGM) — The CGM module generates a base clock signal from which the system clocks are derived. The CGM may include a crystal oscillator circuit and/or phase-locked loop (PLL) circuit.

comparator — A device that compares the magnitude of two inputs. A digital comparator defines the equality or relative differences between two binary numbers.

computer operating properly module (COP) — A counter module that resets the MCU if allowed to overflow.

condition code register (CCR) — An 8-bit register in the CPU that contains the interrupt mask bit and five bits that indicate the results of the instruction just executed.

control bit — One bit of a register manipulated by software to control the operation of the module.

control unit — One of two major units of the CPU. The control unit contains logic functions that synchronize the machine and direct various operations. The control unit decodes instructions and generates the internal control signals that perform the requested operations. The outputs of the control unit drive the execution unit, which contains the arithmetic logic unit (ALU), CPU registers, and bus interface.

COP — See "computer operating properly module (COP)."

CPU — See “central processor unit (CPU).”

CPU12 — The CPU of the MC68HC12 Family.

CPU clock — Bus clock select bits BCSP and BCSS in the clock select register (CLKSEL) determine which clock drives SYSCLK for the main system, including the CPU and buses. When EXTALi drives the SYSCLK, the CPU or bus clock frequency (f_o) is equal to the EXTALi frequency divided by 2.

CPU cycles — A CPU cycle is one period of the internal bus clock, normally derived by dividing a crystal oscillator source by two or more so the high and low times will be equal. The length of time required to execute an instruction is measured in CPU clock cycles.

CPU registers — Memory locations that are wired directly into the CPU logic instead of being part of the addressable memory map. The CPU always has direct access to the information in these registers. The CPU registers in an M68HC12 are:

- A (8-bit accumulator)
- B (8-bit accumulator)
  - D (16-bit accumulator formed by concatenation of accumulators A and B)
- IX (16-bit index register)
- IY (16-bit index register)

- SP (16-bit stack pointer)
- PC (16-bit program counter)
- CCR (8-bit condition code register)

cycle time — The period of the operating frequency: $t_{CYC} = 1/f_{OP}$.

D — See “accumulators (A and B or D).”

decimal number system — Base 10 numbering system that uses the digits zero through nine.

duty cycle — A ratio of the amount of time the signal is on versus the time it is off. Duty cycle is usually represented by a percentage.

ECT — See “enhanced capture timer.”

EEPROM — Electrically erasable, programmable, read-only memory. A nonvolatile type of memory that can be electrically erased and reprogrammed.

EPROM — Erasable, programmable, read-only memory. A nonvolatile type of memory that can be erased by exposure to an ultraviolet light source and then reprogrammed.

enhanced capture timer (ECT) — The HC12 Enhanced Capture Timer module has the features of the HC12 Standard Timer module enhanced by additional features in order to enlarge the field of applications.

exception — An event such as an interrupt or a reset that stops the sequential execution of the instructions in the main program.

fetch — To copy data from a memory location into the accumulator.

firmware — Instructions and data programmed into nonvolatile memory.

free-running counter — A device that counts from zero to a predetermined number, then rolls over to zero and begins counting again.

full-duplex transmission — Communication on a channel in which data can be sent and received simultaneously.

hexadecimal — Base 16 numbering system that uses the digits 0 through 9 and the letters A through F.

high byte — The most significant eight bits of a word.

illegal address — An address not within the memory map

illegal opcode — A nonexistent opcode.

index registers (IX and IY) — Two 16-bit registers in the CPU. In the indexed addressing modes, the CPU uses the contents of IX or IY to determine the effective address of the operand. IX and IY can also serve as a temporary data storage locations.

input/output (I/O) — Input/output interfaces between a computer system and the external world. A CPU reads an input to sense the level of an external signal and writes to an output to change the level on an external signal.

instructions — Operations that a CPU can perform. Instructions are expressed by programmers as assembly language mnemonics. A CPU interprets an opcode and its associated operand(s) and instruction.

inter-IC bus (I^2C) — A two-wire, bidirectional serial bus that provides a simple, efficient method of data exchange between devices.

interrupt — A temporary break in the sequential execution of a program to respond to signals from peripheral devices by executing a subroutine.

interrupt request — A signal from a peripheral to the CPU intended to cause the CPU to execute a subroutine.

I/O — See “input/output (I/0).”

jitter — Short-term signal instability.

latch — A circuit that retains the voltage level (logic 1 or logic 0) written to it for as long as power is applied to the circuit.

latency — The time lag between instruction completion and data movement.

least significant bit (LSB) — The rightmost digit of a binary number.

logic 1 — A voltage level approximately equal to the input power voltage (V_{DD}).

logic 0 — A voltage level approximately equal to the ground voltage (V_{SS}).

low byte — The least significant eight bits of a word.

M68HC12 — A Motorola family of 16-bit MCUs.

mark/space — The logic 1/logic 0 convention used in formatting data in serial communication.

mask — 1. A logic circuit that forces a bit or group of bits to a desired state. 2. A photomask used in integrated circuit fabrication to transfer an image onto silicon.

MCU — Microcontroller unit. See “microcontroller.”

memory location — Each M68HC12 memory location holds one byte of data and has a unique address. To store information in a memory location, the CPU places the address of the location on the address bus, the data information on the data bus, and asserts the write signal. To read information from a memory location, the CPU places the address of the location on the address bus and asserts the read signal. In response to the read signal, the selected memory location places its data onto the data bus.

memory map — A pictorial representation of all memory locations in a computer system.

MI-Bus — See "Motorola interconnect bus".

microcontroller — Microcontroller unit (MCU). A complete computer system, including a CPU, memory, a clock oscillator, and input/output (I/O) on a single integrated circuit.

modulo counter — A counter that can be programmed to count to any number from zero to its maximum possible modulus.

most significant bit (MSB) — The leftmost digit of a binary number.

Motorola interconnect bus (MI-Bus) — The Motorola Interconnect Bus (MI Bus) is a serial communications protocol which supports distributed real-time control efficiently and with a high degree of noise immunity.

Motorola scalable CAN (msCAN) — The Motorola scalable controller area network is a serial communications protocol that efficiently supports distributed real-time control with a very high level of data integrity.

msCAN — See "Motorola scalable CAN".

MSI — See "multiple serial interface".

multiple serial interface — A module consisting of multiple independent serial I/O sub-systems, e.g. two SCI and one SPI.

multiplexer — A device that can select one of a number of inputs and pass the logic level of that input on to the output.

nibble — A set of four bits (half of a byte).

object code — The output from an assembler or compiler that is itself executable machine code, or is suitable for processing to produce executable machine code.

opcode — A binary code that instructs the CPU to perform an operation.

open-drain — An output that has no pullup transistor. An external pullup device can be connected to the power supply to provide the logic 1 output voltage.

operand — Data on which an operation is performed. Usually a statement consists of an operator and an operand. For example, the operator may be an add instruction, and the operand may be the quantity to be added.

oscillator — A circuit that produces a constant frequency square wave that is used by the computer as a timing and sequencing reference.

OTPROM — One-time programmable read-only memory. A nonvolatile type of memory that cannot be reprogrammed.

overflow — A quantity that is too large to be contained in one byte or one word.

page zero — The first 256 bytes of memory (addresses $0000–$00FF).

parity — An error-checking scheme that counts the number of logic 1s in each byte transmitted. In a system that uses odd parity, every byte is expected to have an odd number of logic 1s. In an even parity system, every byte should have an even number of logic 1s. In the transmitter, a parity generator appends an extra bit to each byte to make the number of logic 1s odd for odd parity or even for even parity. A parity checker in the receiver counts the number of logic 1s in each byte. The parity checker generates an error signal if it finds a byte with an incorrect number of logic 1s.

PC — See “program counter (PC).”

peripheral — A circuit not under direct CPU control.

phase-locked loop (PLL) — A clock generator circuit in which a voltage controlled oscillator produces an oscillation which is synchronized to a reference signal.

PLL — See "phase-locked loop (PLL)."

pointer — Pointer register. An index register is sometimes called a pointer register because its contents are used in the calculation of the address of an operand, and therefore points to the operand.

polarity — The two opposite logic levels, logic 1 and logic 0, which correspond to two different voltage levels, V_{DD} and V_{SS}.

polling — Periodically reading a status bit to monitor the condition of a peripheral device.

port — A set of wires for communicating with off-chip devices.

prescaler — A circuit that generates an output signal related to the input signal by a fractional scale factor such as 1/2, 1/8, 1/10 etc.

program — A set of computer instructions that cause a computer to perform a desired operation or operations.

program counter (PC) — A 16-bit register in the CPU. The PC register holds the address of the next instruction or operand that the CPU will use.

pull — An instruction that copies into the accumulator the contents of a stack RAM location. The stack RAM address is in the stack pointer.

pullup — A transistor in the output of a logic gate that connects the output to the logic 1 voltage of the power supply.

pulse-width — The amount of time a signal is on as opposed to being in its off state.

pulse-width modulation (PWM) — Controlled variation (modulation) of the pulse width of a signal with a constant frequency.

push — An instruction that copies the contents of the accumulator to the stack RAM. The stack RAM address is in the stack pointer.

PWM period — The time required for one complete cycle of a PWM waveform.

RAM — Random access memory. All RAM locations can be read or written by the CPU. The contents of a RAM memory location remain valid until the CPU writes a different value or until power is turned off.

RC circuit — A circuit consisting of capacitors and resistors having a defined time constant.

read — To copy the contents of a memory location to the accumulator.

register — A circuit that stores a group of bits.

reserved memory location — A memory location that is used only in special factory test modes. Writing to a reserved location has no effect. Reading a reserved location returns an unpredictable value.

reset — To force a device to a known condition.

SCI — See "serial communication interface module (SCI)."

serial — Pertaining to sequential transmission over a single line.

serial communications interface module (SCI) — A module that supports asynchronous communication.

serial peripheral interface module (SPI) — A module that supports synchronous communication.

set — To change a bit from logic 0 to logic 1; opposite of clear.

shift register — A chain of circuits that can retain the logic levels (logic 1 or logic 0) written to them and that can shift the logic levels to the right or left through adjacent circuits in the chain.

signed — A binary number notation that accommodates both positive and negative numbers. The most significant bit is used to indicate whether the number is positive or negative, normally logic 0 for positive and logic 1 for negative. The other seven bits indicate the magnitude of the number.

software — Instructions and data that control the operation of a microcontroller.

software interrupt (SWI) — An instruction that causes an interrupt and its associated vector fetch.

SPI — See "serial peripheral interface module (SPI)."

stack — A portion of RAM reserved for storage of CPU register contents and subroutine return addresses.

stack pointer (SP) — A 16-bit register in the CPU containing the address of the next available storage location on the stack.

start bit — A bit that signals the beginning of an asynchronous serial transmission.

status bit — A register bit that indicates the condition of a device.

stop bit — A bit that signals the end of an asynchronous serial transmission.

subroutine — A sequence of instructions to be used more than once in the course of a program. The last instruction in a subroutine is a return from subroutine (RTS) instruction. At each place in the main program where the subroutine instructions are needed, a jump or branch to subroutine (JSR or BSR) instruction is used to call the subroutine. The CPU leaves the flow of the main program to execute the instructions in the subroutine. When the RTS instruction is executed, the CPU returns to the main program where it left off.

synchronous — Refers to logic circuits and operations that are synchronized by a common reference signal.

timer — A module used to relate events in a system to a point in time.

toggle — To change the state of an output from a logic 0 to a logic 1 or from a logic 1 to a logic 0.

tracking mode — A mode of PLL operation with narrow loop bandwidth. Also see ‘acquisition mode.’

two’s complement — A means of performing binary subtraction using addition techniques. The most significant bit of a two’s complement number indicates the sign of the number (1 indicates negative). The two’s complement negative of a number is obtained by inverting each bit in the number and then adding 1 to the result.

unbuffered — Utilizes only one register for data; new data overwrites current data.

unimplemented memory location — A memory location that is not used. Writing to an unimplemented location has no effect. Reading an unimplemented location returns an unpredictable value.

variable — A value that changes during the course of program execution.

VCO — See "voltage-controlled oscillator."

vector — A memory location that contains the address of the beginning of a subroutine written to service an interrupt or reset.

voltage-controlled oscillator (VCO) — A circuit that produces an oscillating output signal of a frequency that is controlled by a dc voltage applied to a control input.

waveform — A graphical representation in which the amplitude of a wave is plotted against time.

wired-OR — Connection of circuit outputs so that if any output is high, the connection point is high.

word — A set of two bytes (16 bits).

write — The transfer of a byte of data from the CPU to a memory location.

HOW TO REACH US:

DRM042/D

USA/EUROPE/LOCATIONS NOT LISTED:
Motorola Literature Distribution
P.O. Box 5405
Denver, Colorado 80217
1-800-521-6274 or 480-768-2130

JAPAN:
Motorola Japan Ltd.
SPS, Technical Information Center
3-20-1, Minami-Azabu, Minato-ku
Tokyo 106-8573, Japan
81-3-3440-3569

ASIA/PACIFIC:
Motorola Semiconductors H.K. Ltd.
Silicon Harbour Centre
2 Dai King Street
Tai Po Industrial Estate
Tai Po, N.T., Hong Kong
852-26668334

HOME PAGE:
http://motorola.com/semiconductors

MOTOROLA

Information in this document is provided solely to enable system and software implementers to use Motorola products. There are no express or implied copyright licenses granted hereunder to design or fabricate any integrated circuits or integrated circuits based on the information in this document.

Motorola reserves the right to make changes without further notice to any products herein. Motorola makes no warranty, representation or guarantee regarding the suitability of its products for any particular purpose, nor does Motorola assume any liability arising out of the application or use of any product or circuit, and specifically disclaims any and all liability, including without limitation consequential or incidental damages. “Typical” parameters that may be provided in Motorola data sheets and/or specifications can and do vary in different applications and actual performance may vary over time. All operating parameters, including “Typicals”, must be validated for each customer application by customer’s technical experts. Motorola does not convey any license under its patent rights nor the rights of others. Motorola products are not designed, intended, or authorized for use as components in systems intended for surgical implant into the body, or other applications intended to support or sustain life, or for any other application in which the failure of the Motorola product could create a situation where personal injury or death may occur. Should Buyer purchase or use Motorola products for any such unintended or unauthorized application, Buyer shall indemnify and hold Motorola and its officers, employees, subsidiaries, affiliates, and distributors harmless against all claims, costs, damages, and expenses, and reasonable attorney fees arising out of, directly or indirectly, any claim of personal injury or death associated with such unintended or unauthorized use, even if such claim alleges that Motorola was negligent regarding the design or manufacture of the part.

MC68HC908QY4/JB8/JB16
Rev. 0
6/2003