



**Genesys Logic, Inc.**

# **GL310MC4D**

---

## **PS/2 4D MOUSE CONTROLLER**

**Specification Revision 1.0**

**July 16, 1998**

***Genesys Logic, Inc.***

*10F., No.11, Ln.3, Tsao Ti Wei, ShenKeng, Taipei, Taiwan, R.O.C.*

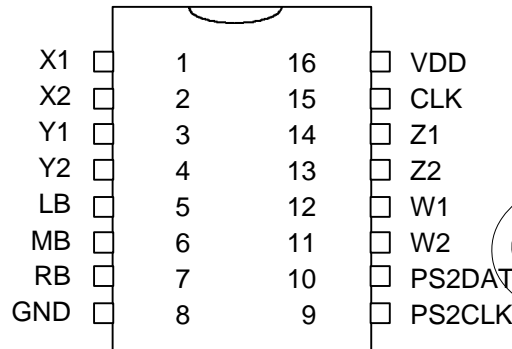
*TEL: 886-2-2664-6655 FAX:886-2-2664-5757*

*<http://www.genesyslogic.com.tw>*

## 1. Features

- Low-cost solution for PS/2 4D mouse
- Compatible with Microsoft IntelliMouse
- I/O ports
  - 8 special purpose I/O pins optimized for photo-sensor
  - 2 I/O pins with LED drive capability
- Patent pending full-range detection for photo-sensor
  - Removes the expensive process of matching LED and photo-sensor
  - Improve end-product yield rate almost to 100%
  - Huge cost saving
- Option to use wheel or button for scrolling
- Option to detect 2 or 4 pulses for one move
- R-C oscillator
  - No expensive crystal oscillator required
  - Reduce BOM
- Improved output drivers with slew-rate control to reduce EMI
- Minimal external components(2R or 3R + 2C) required for application circuit
- Tracking speed
  - Up to 100 $\mu$ s/per sampling
- 6 MHz external clock
- Internal power-on reset(POR)
- Available in cost saving 16-pin PDIP

## 2. Pin Configuration

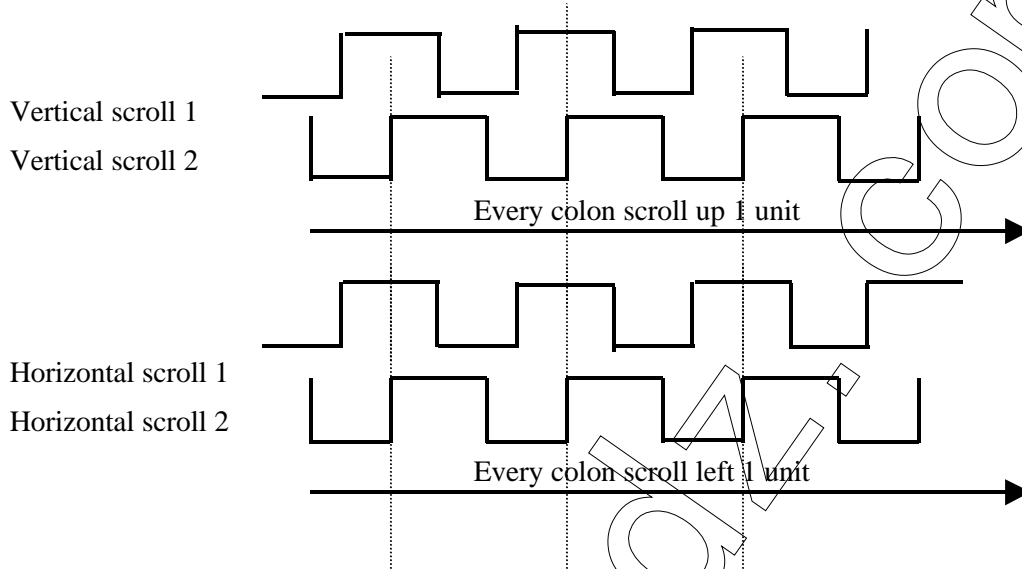


**Figure 2-1. 16-Pin DIP**

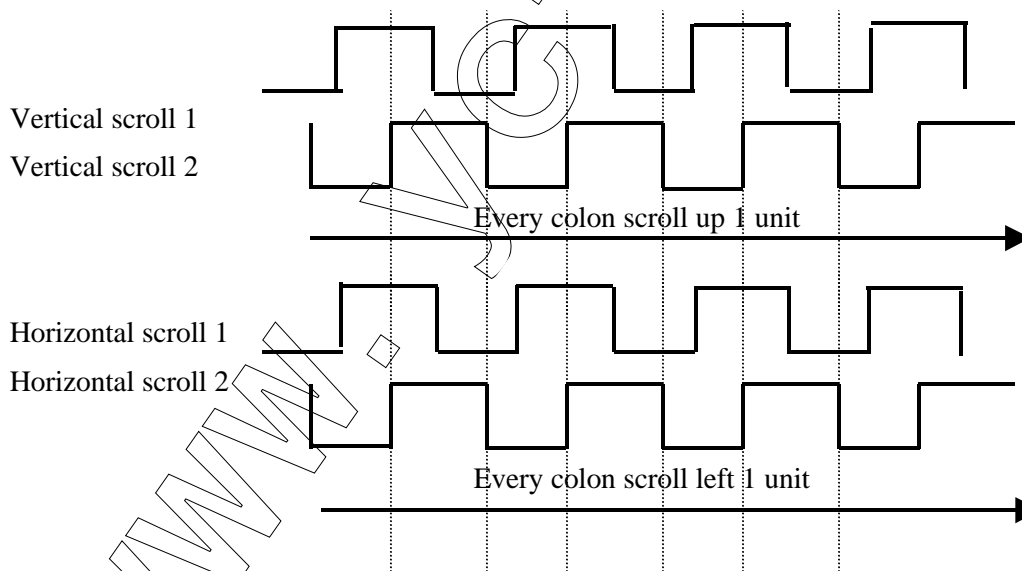
Pin	Symbol	I/O	Description
1	X1	I	Photo sensor input for X axis 1
2	X2	I	Photo sensor input for X axis 2
3	Y1	I	Photo sensor input for Y axis 1
4	Y2	I	Photo sensor input for Y axis 2
5	LB	I	Mouse left button, internal pulled high
6	MB	I	Mouse middle button, internal pulled high
7	RB	I	Mouse right button, internal pulled high
8	GND	Ground	Ground
9	PS2CLK	I/O	PS/2 mouse clock
10	PS2DAT	I/O	PS/2 mouse data
11	W2	I	Photo sensor input for vertical scroll 1 Vertical scroll up button internal pulled up
12	W1	I	Photo sensor input for vertical scroll 2 Vertical scroll down button, internal pulled up
13	Z2	I	Photo sensor input for horizontal scroll 1 horizontal scroll right button internal pulled up
14	Z1	I	Photo sensor input for horizontal scroll 2 Horizontal scroll left button internal pulled up
15	CLK	Clock	6MHz RC clock input
16	VDD	Power	Power

### 3. Scrolling control

#### 3.1 Wheel Scrolling Control (4 pulses for one move)



#### 3.2 Wheel Scrolling Control (2 pulses for one move)



### **3.3 Button scrolling control**

- Use button to control vertical/horizontal scrolling
- 320ms before 4<sup>th</sup> repeat
- 40ms after 4<sup>th</sup> repeat

### **3.4 GL310MC4D Series Selection Guide**

- GL310MC4D2P
  - PS/2 4D Mouse, wheel scrolling, 2 pulses for one move
- GL310MC4D4P
  - PS/2 4D Mouse, wheel scrolling, 4 pulses for one move
- GL310MC4D1B
  - PS/2 4D Mouse, button scrolling

## 4. PS/2 Protocol

The PS/2 protocol allows synchronous, bi-directional serial communication between the host and the device. Either side may transmit a command or data byte at any time, although only one side can transmit at one time. During initialization, the host sends command bytes to the device. Some commands are followed by argument bytes. The device acknowledges each command and argument byte with an ACK (\$FA) byte, possibly followed by one or more data bytes. If the host has enabled "Stream mode" transmission, then the device may send spontaneous data packets to the host describing finger motions and button state changes.

### 4.1 Electrical interface

The PS/2 protocol includes two signal wires as well as +5V power and ground. The signal wires, CLK and DATA, are bi-directional "open-collector" signals; they are normally held at a high (+5V) level by a 5-10K pull-up resistor on the host, but either the host or the Mouse device can pull them low at any time. When the port is idle, both signal wires are floating high. The host can inhibit the device at any time by holding the CLK wire low.

Note that neither side ever actively pulls CLK or DATA high; to output a logic 1, the wire is left undriven and allowed to float high.

The following diagram shows the interconnections between the host and the PS/2 mouse device:

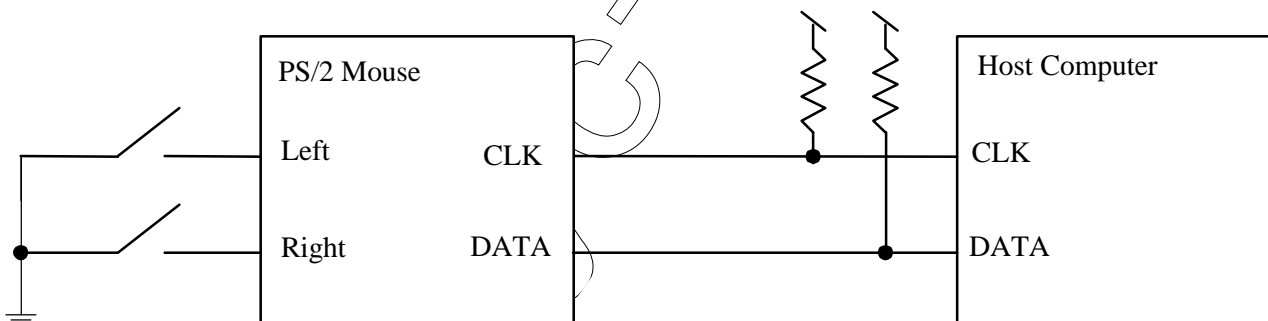


Figure 4-1. PS/2 system diagram

### 4.2 Byte transmission

Each byte transmitted between the device and the host includes a start bit ( a logic 0), eight data bits (LSB first), a parity bit (odd parity), and a stop bit (a logic 1). Odd parity means the eight data bits and the parity bit together contain an odd number of 1's . During transmission , the device pulses the CLK signal low for each of the 11bits, while either the host or the device pulls the DATA wire low to signal a logic 0 or allows DATA to float high to signal a logic 1.

Between transmissions, the bus can be in one of three states:

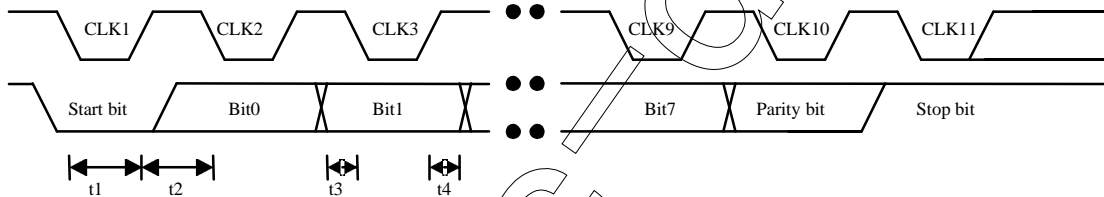
- **Idle.** If CLK and DATA are both high, there is no activity on the bus.

- **Inhibit.** If the host is holding CLK low, the device is inhibited from transmitting data. However, internal mouse processing continues to occur.
- **Request to send.** If the host is holding DATA low and CLK high, it is ready to transmit a command or argument byte to the device.

### 4.3 Output to host

The device can transmit a byte to the host whenever the bus is idle. If the bus is inhibited, the device waits for it to leave the inhibit state before transmitting. If there is a host request-to-send, the device receives and processes the host command byte rather than transmitting its own data.

If the host holds CLK low (inhibit) during a transmission, the device will recognize this within 100 $\mu$ s and abort the transmission. The device recognizes an inhibit by noting that CLK remains low during the high portion of the clock cycle. If the inhibit occurs before the rising edge of the tenth clock (the parity bit), the transmission is cancelled and will be retried as soon as the inhibit is released. (An ACK (\$FA) reply to an incoming command or argument byte is simply thrown away if cancelled, although the command being acknowledged is not cancelled.) If the inhibit begins after the tenth clock, the transmission is considered complete and the host must accept the transmitted byte. The host may hold CLK low after the transmission, effectively extending clock 11, to inhibit the device while it processes the transmission.



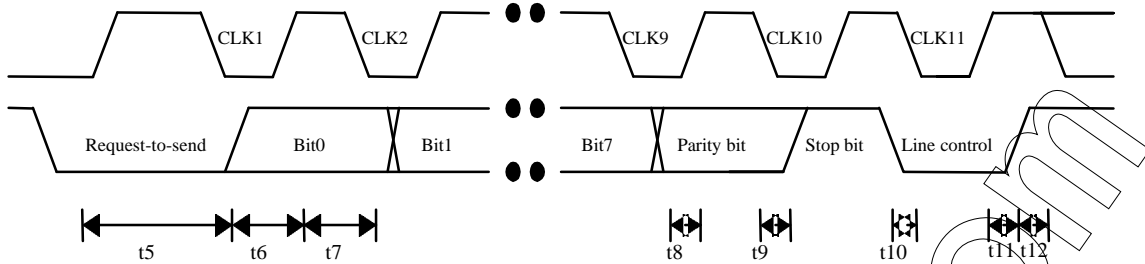
**Figure 4-2. PS/2 output waveforms**

In Figure 4-2, the CLK signal is low for 30-50 $\mu$ s ( $t_1$ ) and high for 30-50 $\mu$ s ( $t_2$ ) in each bit cell. DATA will be valid at least 5 $\mu$ s before the falling edge ( $t_3$ ) and at least 5 $\mu$ s after the rising edge ( $t_4$ ) of the clock.

#### 4.3.1 Input from host

The host signals its intent to transmit a command or argument byte by holding CLK low for at least 60 $\mu$ s, then pulling DATA low and releasing CLK, thus putting the bus into the host request-to-send state. The device checks for this state at least every 10ms ( $t_5$ ). When the device detects a request-to-send, it pulses CLK low 11 times to receive a byte. The host can abort the transmission midway through by holding CLK low for at least 100 $\mu$ s.

After the tenth clock, the device checks for a valid stop bit (DATA line high), and responds by pulling DATA low and clocking one more time (the "line control bit"). The host can then hold CLK low within 50 $\mu$ s ( $t_{12}$ ) to inhibit the device until the host is ready to receive the reply. If the device finds DATA low during the stop bit, a framing error has occurred; the device continues to clock until DATA goes high, then sends a Resend to the host as described below.



**Figure 4-3. PS/2 output waveforms**

In Figure 4-3, the CLK signal is low for 30-50 $\mu$ s ( $t_6$ ) and high for 30-50 $\mu$ s ( $t_7$ ) in each bit cell. DATA is sampled when CLK is high, and must be valid no later than 1 $\mu$ s after the rising edge of the clock ( $t_8 > -1\mu$ s.  $t_9 \geq 0\mu$ s). In the line control bit, DATA goes low at least 5 $\mu$ s before the falling edge ( $t_{10}$ ) and stays low at least 5 $\mu$ s after the rising edge ( $t_{11}$ ) of the clock.

For each command or argument byte that requires a response, the host waits for the response before sending another byte. The device responds within 20ms, unless inhibited by the host.

If the device receives an invalid byte, a byte with incorrect parity, or a framing error, it sends a Resend (\$FE) response to the host. If the next input from the host is also invalid, the device sends an Error (\$FC) response.

## 4.4 Power-on reset

At power-up, the mouse performs a self-test and calibration, then transmits the completion code \$AA and ID code \$00. If the device fails its self-test, it transmits error code \$FC and ID code \$00. This processing also occurs when a software Reset (\$FF) command is received. The host should not attempt to send commands to the device until the calibration/self-test is complete.

Power-on self-test and calibration takes 300-1000ms. Self-test and calibration following a software Reset command takes 300-500ms.

If the host wishes to use the mouse as soon as possible after power-up and is not interested in the \$AA, \$00 announcement, it can simply put the bus into the "request-to -send" state immediately on power-up. The mouse will respond by clocking in the host's first initialization command as soon as it is ready; this command will override the \$AA, \$00 announcement. If the host sends a Reset (\$FF) before the initial \$AA, \$00 announcement, then the \$AA, \$00 response to the Reset command may be delayed by the full 300-1000ms required for power-on calibration.



The reset state of the device is as follows:

- Reported sample rate is 100 samples per second.
- Reported resolution is 4 counts per mm.
- Scaling is 1:1.
- Stream mode is selected.
- Data reporting is disabled.

### 4.5 Command set

After every command, the device responds within 20ms with an ACK byte ( \$FA) and possibly one or more data bytes. For the Resend ( \$FE) command, the response sometimes does not begin with an ACK. For the Reset ( \$FF) command, the device responds with an ACK ( \$FA) within 20ms, then with the completion and ID codes ( \$AA, \$00) within 500ms.

If the device is in Stream mode, the host should disable it before sending a command that requires a response. However, if a command is sent during enable Stream mode, the device will not send any further data packets until the responding to the command is finished. If a command is sent during the transmission of a data packet or response to a previous command, the packet in progress is abandoned and the new command is processed instead.

The device maintains a count of the amount of X, Y motion since the last data packet was sent. Any command except Resend ( \$FE) clears the motion accumulators.

#### 4.5.1 Standard PS/2 commands

The commands described here are standard to all PS/2 pointing devices. As elsewhere in this document, "\$" signifies hexadecimal notation.

**\$FF Reset.** Perform a software reset as described above. Response is ACK ( \$FA), then \$AA, \$00 after a calibration delay of 300-500ms.

**\$FE Resend.** The host sends this command when it detects invalid output from the device. The device retransmits the last packet of data, for example, a three -byte motion data packet, a one-byte response to the Read Device Type ( \$F2) command, or the two-byte completion-and-ID reset response ( \$AA, \$00). The ACK ( \$FA) byte sent to acknowledge a command is not stored in any buffer or resent; however, if the last output from the device was an ACK with no additional data bytes, Resend responds with an ACK.

The device will send a Resend ( \$FE) to the host if it receives invalid input from the host.

**\$F6 Set Default.** Restore conditions to the initial power-up state. This resets the sample rate, resolution, sealing, and Stream mode as described above for power-on reset, and disables the device.

**\$F5 Disable.** Disable Stream mode reporting of motion data packets. All other device operations continue as usual.



## GL310MC PS/2 4D Mouse Controller

---

**\$F4 Enable.** Begin sends motion data packets if in Stream mode. To avoid undesirable bus contention, driver software should send the Enable as the very last command in its PS/2 initialization sequence.

Note that a PS/2 device includes two distinct state bits: the enable/disable flag controlled by commands \$F4 and \$F5, and the Stream/Remote flag controlled by commands \$EA and \$FO. The flags are independent, and both must be set properly (enabled, Stream mode) for the device to send motion packets. The intention is that disabled Stream mode means the host is not interested in motion packets, while Remote mode means the host plans to poll explicitly for motion data.

**\$F3 Set Sample Rate.** Followed by one argument byte, this command sets the PS/2 "Sample rate" parameter to the specified value in samples per second. Legal values are 10, 20, 40, 60, 80, 100, and 200 (decimal) samples per second. The mouse records this value and will respond properly to a later Status Request (\$E9) command.

**\$F2 Read Device Type.** The response is an ACK (\$FA) followed by a \$00 device ID byte.

**\$FO Set Remote Mode.** Switch to Remote mode, as distinct from the default Stream mode. In Remote mode, the device sends motion data packets only in response to a Read Data (\$EB) command.

**\$EE Set Wrap Mode.** Switch into special "echo" or "Wrap" mode. In this mode, all bytes sent to the device except Reset (\$FF) and Reset Wrap Mode (\$EC) are echoed back verbatim. No motion data packets are sent in Wrap mode.

**\$EC Reset Wrap Mode.** If the device is in Wrap mode, it returns to its previous mode of operation except that Stream mode data reporting is disabled. If the device is not in Wrap mode, this command has no effect.

**\$EB Read Data.** The device replies with an ACK (\$FA) followed by a 3-byte motion data packet as described below. This command may be used in any mode; in Remote mode, it is the only way to get a data packet. If the host is polling regularly, it is best to set the device to Remote mode. The packet is transmitted even if no motion or button events have occurred.

**\$EA Set Stream Mode.** Switch to Stream mode, the default mode of operation. In this mode, motion data packets are sent to the host whenever finger motion or button events occur and data reporting has been enabled.

**\$E9 Status Request.** Response is an ACK (\$FA), followed by a 3-byte status packet consisting of the following data:



## GL310MC PS/2 4D Mouse Controller

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Byte 1	0	Remote	Enable	Scaling	0	Left	Middle	Right
Byte 2	0	0	0	0	0	0	Resolution	
Byte 3	Sample rate							

- Remote:        1= Remote (polled) mode, 0 = Stream mode.
- Enable :       1 = Data reporting enabled, 0 =disabled. This bit only has effect in Stream mode.
- Scaling:       1 = Scaling is 2:1, 0 = scaling is 1:1.
- Left:           1=Left button is currently pressed, 0 = released.
- Middle:        1 = Middle button is currently pressed, 0 = released.
- Right:          1 = Right button is currently pressed, 0 = released.

Resolution :    The current resolution setting, from 0 to 3 as described under Set Resolution (\$E8) below.

Sample rate :    The current sample rate sample rate setting, from 10 to 200 as described under Set Sample Rate (\$F3) above.

- \$E8 Set Resolution.** Followed by one argument byte, this command sets the PS/2 " resolution" parameter. Legal argument values are \$00, \$01, \$02, and \$03, corresponding to resolutions of 1, 2, 4, and 8 counts per mm, respectively.
- \$E7 Set Scaling 2:1.** Sets the PS/2 "scaling" bit, to enable a non-linear motion gain response. The mouse records this value and will respond properly to a later Status Request (\$E9) command.
- \$E6 Set Scaling 1:1.** Clears the PS/2 "scaling" bit, as described above.
- other* If the device receives an invalid command byte, it replies with a Resend (\$FE) byte. If it immediately receives a second invalid command, it replies with an Error (\$FC) byte.

### 4.5.2 Extended PS/2 commands for Microsoft IntelliMouse

To set the mouse to the extended mode, the driver sends the following command sequence:

- Set Sample Rate 200 HZ (\$F3 \$C8)
- Set Sample Rate 100 HZ (\$F3 \$64)
- Set Sample Rate 80 HZ (\$F3 \$50)
- Set Sample Rate 60 HZ (\$F3 \$3C)

For the mouse to change modes, it must receive these commands exactly as shown with no other commands interspersed in the sequence. Once the driver sends this sequence, it must determine whether the mouse supports the extended mode, so it will know to expect the extra wheel data that is not a typical part of a PS/2-compatible mouse data packet. Once the mouse changes to the extended mode, its device ID becomes \$04.

Typically, a PS/2-compatible mouse has an ID of \$00. Immediately after sending the Set Sample Rate sequence, the driver issues a Read Device Type (\$F2) command. If the mouse returns an ID of \$04, then the driver knows a mouse device is attached and expects an extended PS/2-compatible mouse data packet. If the driver doesn't receive a device ID of \$04 in response to the Read Device Type command, it will issue a Reset (\$FF) command and start over, treating the



## GL310MC PS/2 4D Mouse Controller

device as it would a typical three-button mouse. Note that a Reset (\$FF) command will return a mouse device to its power-up default mode.

### 4.6 Data reporting

The mouse supports one format for motion data packets. The default Relative format is compatible with standard PS/2 mice.

Data packets are sent in response to Read Data (\$EB) commands. If Stream mode is selected and data reporting is enabled, data packets are also sent unsolicited whenever finger motion and/or button-state changes occur.

During transmission of a motion packet, the individual bytes of the packet will be separated by no more than 10ms (assuming the host does not inhibit the bus). While PS/2 motion packets are lacking in explicit synchronization bits, if the host sees a long delay between bytes it can assume the delay comes at a packet boundary. (Similarly, the bytes of the responses to the various PS/2 commands will be separated by no more than 10ms; the one exception is Reset (\$FF), where the \$FA and \$AA are separated by a longer calibration delay.)

#### 4.6.1 Default packet format

In the default format, each motion report consists of three bytes. The first byte encodes various status bits, and the other two bytes encode the amount of motion in X and Y.

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Byte 1	Y ovfl	X ovfl	Y sign	X sign	1	Middle	Right	Left
Byte 2					X delta			
Byte 3					Y delta			

Y ovfl: 1 = Y delta value exceeds the range -256...255, 0 = no overflow. When this bit is set, the reported Y delta will be either -256 or +255.

X ovfl: 1 = X delta value exceeds the range -256...255, 0 = no overflow. When this bit is set, the reported X delta will be either -256 or +255.

Y sign: 1 = Y delta value is negative, 0 = Y delta is zero or positive.

X sign: 1 = X delta value is negative, 0 = X delta is zero or positive.

Middle: 1 = Middle button is currently pressed, 0 = released.

Right: 1 = Right button is currently pressed, 0 = released.

Left: 1 = Left button is currently pressed, 0 = released.

X delta: This is the amount of motion that has occurred in the X (horizontal) direction since the last motion data report. This byte and the "X sign" bit of byte 1 combine to form a nine-bit signed, two's-complement integer.

Y delta: This is the amount of motion that has occurred in the Y (vertical) direction. Upward motion is positive, downward is negative.

### 4.6.2 Extended Microsoft IntelliMouse packet format

When mouse devices are in extended mode, they send their mouse data in a 4-byte data packet. The packet is the typical standard PS/2 3-byte packet plus one additional byte containing the wheel data denoted as Z in the following table.

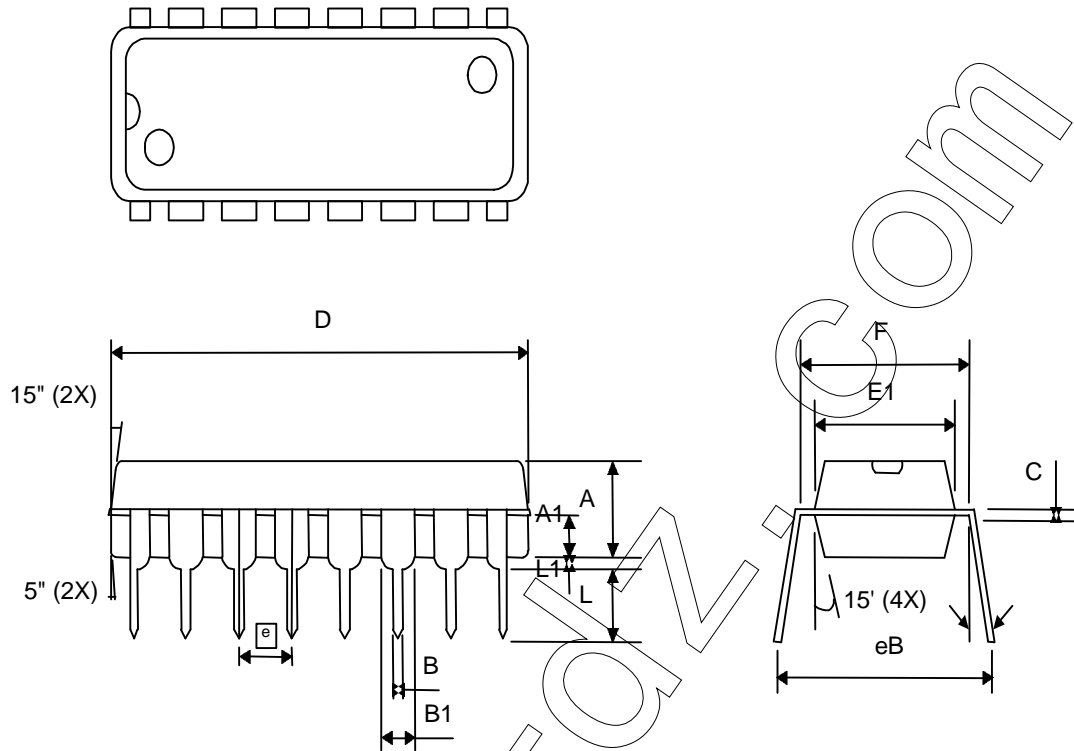
	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
<i>Byte 1</i>	0	0	Y7	X7	1	Middle	Right	Left
<i>Byte 2</i>	X delta							
<i>Byte 3</i>	Y delta							
<i>Byte 4</i>	Z delta							
<i>Byte 5</i>	W delta							

www.yc-dz.com

## 5. Electrical Characteristics

Parameter	Condition	Min.	Typ.	Max.	Unit
Operating Voltage		4.0	-	6.0	V
Operating Current	FSYS=6MHz, No Load ROSC=10K, COSC=10pF	-	1.3	-	mA
Input Low Voltage for LB, MB, RB	-	0	-	0.8	V
Input High Voltage for LB, MB, RB	-	3.0	-	5.0	V
Input Low Voltage for X1,X2,Y1,Y2,Z1,Z2,W1,W2 W1,W2	-	0	-	0.8	V
Input High Voltage for X1,X2,Y1,Y2,Z1,Z2,W1,W2 W1,W2	-	3.0	-	5.0	V
X1,X2,Y1,Y2,Z1,Z2 Sink Current	VDD=5V, VOL=0.5V	-	4	-	mA
X1,X2,Y1,Y2,Z1,Z2 Source Current	VDD=5V, VOH=4.5V	-	- 4	-	mA
W1,W2 Sink Current	VDD=5V, VOL=0.5V	-	16	-	mA
W1,W2 Source Current	VDD=5V, VOH=4.5V	-	- 16	-	mA
CLK, DATA Sink Current	VDD=5V, VOL=0.5V	-	4	-	mA
CLK, DATA Source Current	VDD=5V, VOH=4.5V	-	- 4	-	mA
Pull-High Resistance of CLK, DATA W1,W2	-	-	5	-	K $\Omega$
Pull-High Resistance of RB, MB, LB W1,W2	-	-	10	-	K $\Omega$
Pull-Low Resistance of X1,X2,Y1,Y2,Z1,Z2,W1,W2, W1,W2	-	-	8	-	K $\Omega$

## 6. Package

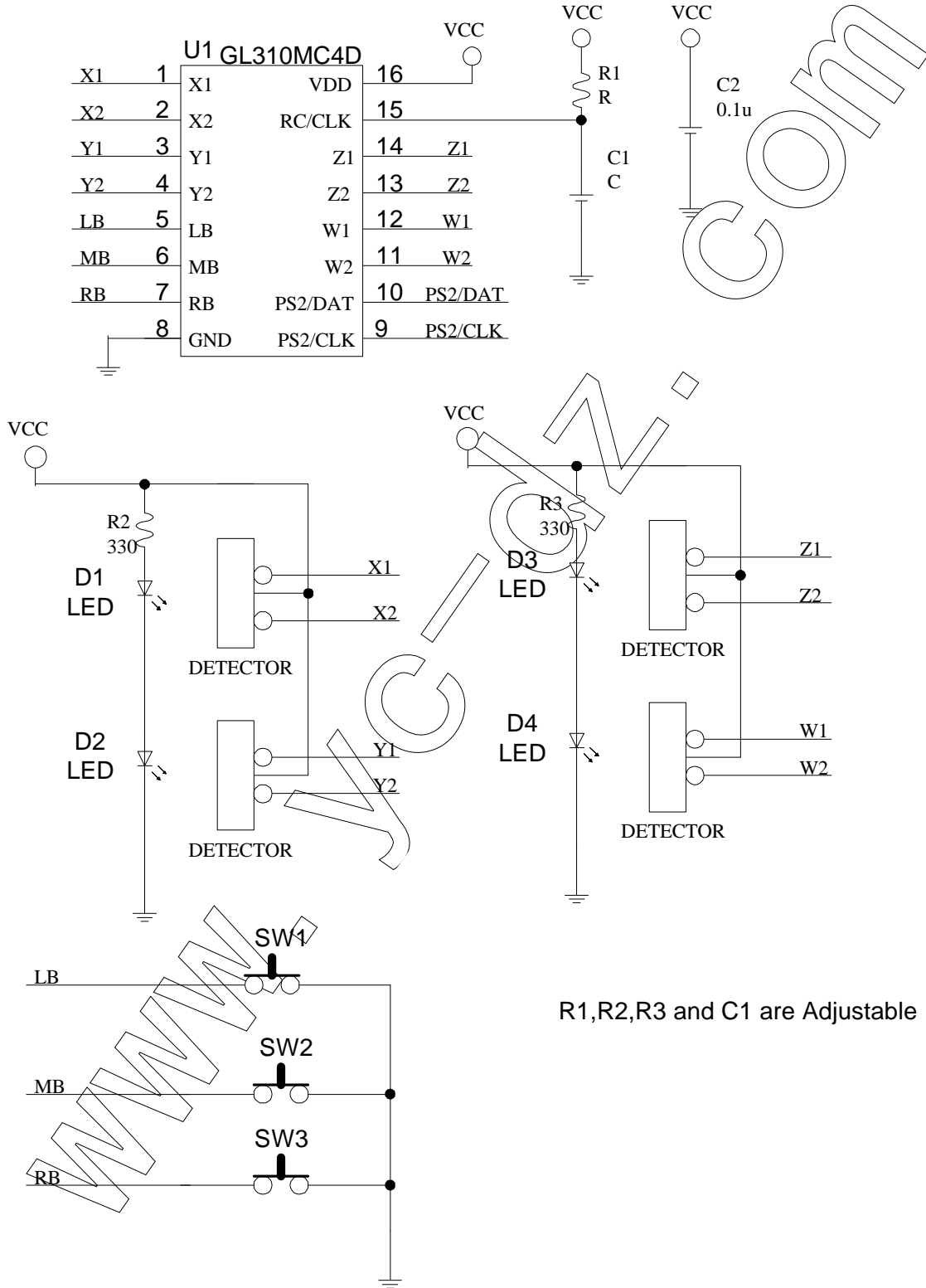


Symbol	Dimension in mils			Dimension in mils		
	Min	Nom	Max	Min	Nom	Max
A	--	130	--	--	3.302	--
A1	59	60	61	1.499	1.524	1.549
B	--	18	--	--	0.457	--
B1	--	60	--	--	1.524	--
C	--	10	--	--	0.254	--
D	740	750	760	18.796	19.050	19.304
E1	259	260	--	6.579	6.604	--
F	290	300	310	7.366	7.620	7.874
L	--	130	--	--	3.302	--
L1	--	20	--	--	0.508	--
e	--	100	--	--	2.540	--
EB	345	355	365	8.763	9.017	9.271
$\theta$	4°	5.5°	7°	4°	5.5°	7°

**Figure 6-1. Package outline dimension for P-DIP 16 pin**

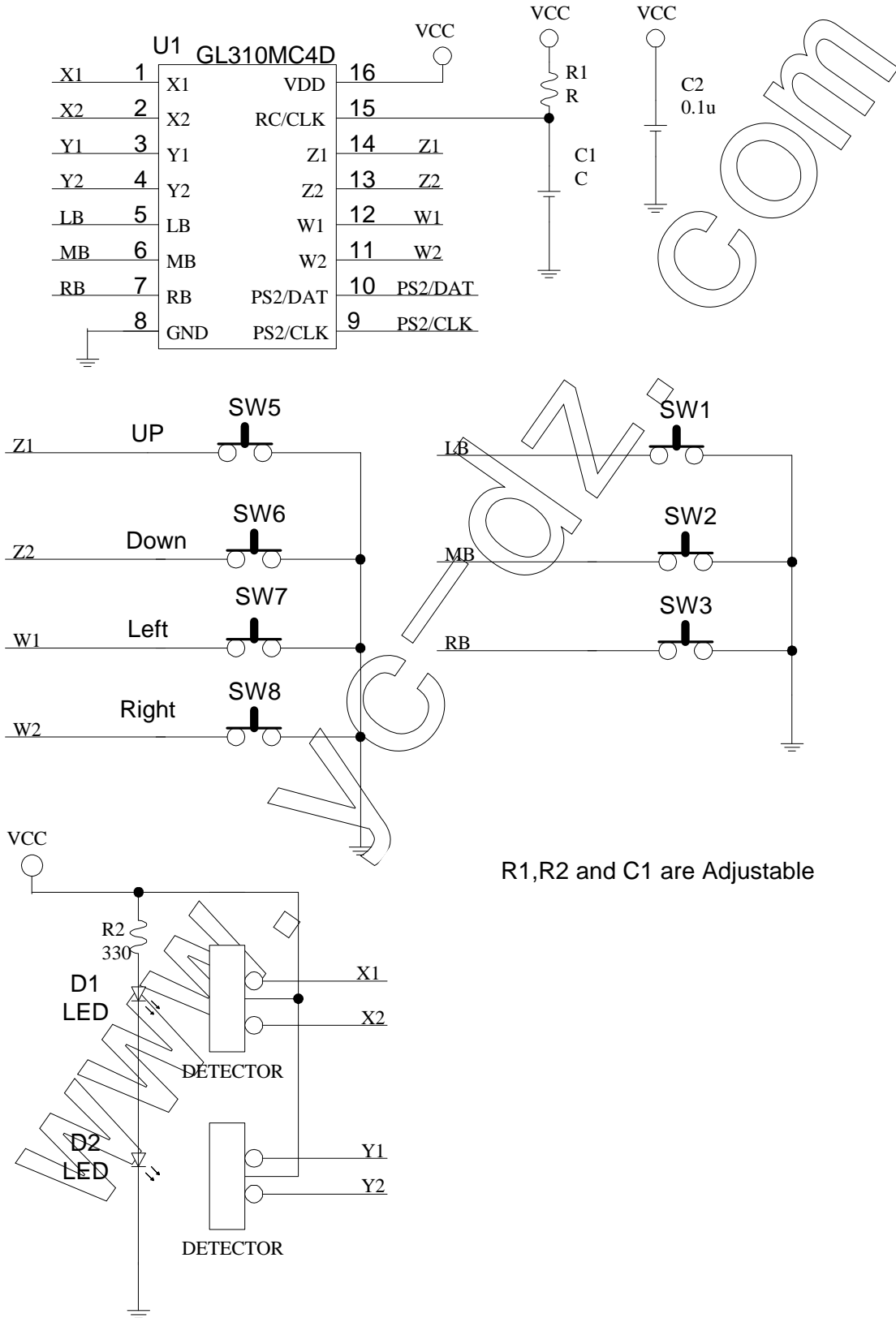
## 7. Application Circuit

### 7.1 Wheel Scrolling Mode





## 7.2 Button Scrolling Mode



## Contents

1. Features .....	1
2. Pin Configuration.....	2
3. Scrolling control .....	3
3.1 Wheel Scrolling Control (4 pulses for one move) .....	3
3.2 Wheel Scrolling Control (2 pulses for one move) .....	3
3.3 Button scrolling control.....	4
3.4 GL310MC4D Series Selection Guide .....	4
4. PS/2 Protocol .....	5
4.1 Electrical interface .....	5
4.2 Byte transmission.....	5
4.3 Output to host .....	6
4.3.1 Input from host .....	6
4.4 Power-on reset .....	7
4.5 Command set.....	8
4.5.1 Standard PS/2 commands.....	8
4.5.2 Extended PS/2 commands for Microsoft IntelliMouse .....	10
4.6 Data reporting.....	11
4.6.1 Default packet format .....	11
4.6.2 Extended Microsoft IntelliMouse packet format.....	12
5. Electrical Characteristics.....	13
6. Package .....	14
7. Application Circuit.....	15
7.1 Wheel Scrolling Mode .....	15
7.2 Button Scrolling Mode.....	16