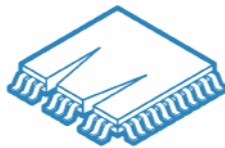
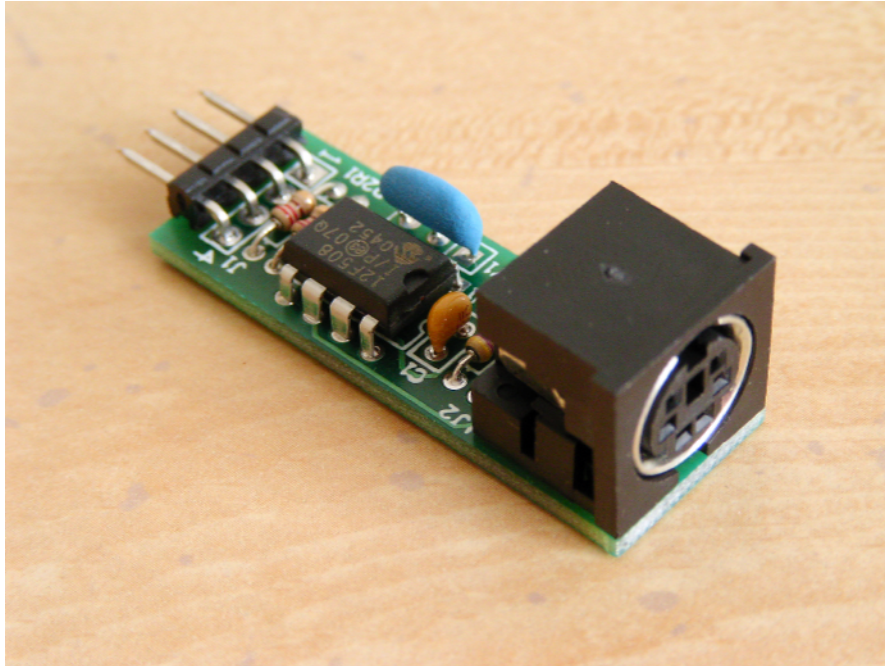


**ezMOUSE**  
**Serial Mouse Adapter**



**MULTILABS**  
LAKE FOREST, CALIFORNIA  
WWW.MULTILABS.NET  
(949) 458-7625

## FEATURES

- Easy to use 2 wire 9600 baud TTL level serial interface
- Standard PS/2 compatible receptacle
- Complete adapter, no external components needed
- Easy to use SIP package – only 1.585” by 0.551” in size
- Use with any PS/2 compatible mouse
- Simple single character command set
- No knowledge of the PS/2 protocol needed
- Responds with easy to understand data packets

## DESCRIPTION

The ezMOUSE is a simple to use adapter that allows you to use a PS/2 compatible computer mouse with microcontrollers, stamp modules, and other embedded devices. Access to the connected mouse is through a standard 2 wire 9600 baud TTL level serial interface. This allows the end-use application to talk to the mouse without any knowledge of the PS/2 protocol or command set.

The easy to use ezMOUSE commands allow the application to initialize the mouse for use, review the current state of the buttons along with movement data since the mouse was last queried, and choose any one or four resolutions to suit the application need.

The ezMOUSE comes in a simple to use SIP module that measures only 1.585 by 0.551 inches. A 4-pin header is provided to supply power to the adapter and access the serial data lines.

## SPECIFICATIONS

### Absolute Maximum Ratings

*Note: These are stress ratings only. Stresses above those listed below may cause permanent damage and/or affect device reliability. The operational ratings should be used to determine applicable ranges of operation.*

Storage Temperature	-30°C to 80°C
Operating Temperature	0°C to 70°C
Supply Voltage (Vcc)	5VDC
Voltages on Tx and Rx Pins	-0.6VDC to (Vcc +0.6VDC)

Characteristic	Symbol	Min	Typ	Max	Unit
Supply Voltage	Vcc	4.5	5.0	5.5	V
Supply Current*	Icc	-	0.5	-	mA

\*Does not include the supply current requirements of the mouse.

## OPERATION BASICS

### About Mice and Compatibility (Important!)

There are a lot of different mice and not all of them are created equal. There are 2, 3, 4, 5+ button mice, USB interface, PS/2 interface, scroll wheels, standard command set, extended command set, etc. The ezMOUSE is designed to be used with a simple, standard PS/2 compatible 2 or 3-button mouse. The ezMOUSE does not contain the extended commands that allow access to extended features, 4 or more buttons or the scroll wheel. About scroll wheels - If your mouse has a scroll wheel some manufacturers default to pushing down on the scroll wheel as a third button press. In this case, the ezMOUSE will detect it. However, some manufacturers require a special code to enable the scroll wheel as a third button. In this case the ezMOUSE will not detect it. Both cases were found during testing. The best way to find out is to try it out with your mouse.

The ezMOUSE was designed for and is guaranteed to work with a fully PS/2 compatible mouse. Multilabs has done extensive testing with various types of mice from different manufacturer's and styles to ensure compatibility. However, some mice are sold with a USB cable instead of a PS/2 cable and are instead provided with a USB to PS/2 adapter. These mice are intended to be connected to a USB port but can be connected to a PS/2 port with the provided adapter if necessary. During testing it was found that some of these mice were not fully compliant with the PS/2 standard and would not function correctly with the ezMOUSE.

To get the most out of your ezMOUSE and to ensure reliability it is recommended that only mice with a PS/2 cable and marked "PS/2 Compatible" be used. You can also check with the manufacturer of your mouse to see if it is fully compliant with PS/2 and the standard PS/2 command set.

### Connections

There are two connectors on the ezMOUSE, a 4-pin sip header and a 6-position mini-DIN receptacle. The SIP pins are marked with the numbers 1 and 4 on the unit to indicate the proper connections. The 6-position mini-DIN receptacle is for connection to a PS/2 compatible mouse. The connections to the 4-pin sip are as follows when looking at them from left to right:

Pin 1 - +5V Input supply

Pin 2 - TTL serial data from the ezMOUSE (Tx)

Pin 3 - TTL serial data to the ezMOUSE (Rx)

Pin 4 - Common ground

### Hooking-Up the ezMOUSE and Applying Power

Start by connecting your PS/2 compatible mouse to the receptacle on the ezMOUSE. Using the 4-pin SIP header, make the necessary connections between your application and the ezMOUSE. In your application you will have a line that transmits data to the ezMOUSE, connect that to pin 3 of the SIP header. The line that receives data from the ezMOUSE needs to be connected to pin 2 of the SIP header. Lastly, connect +5VDC and common power from your application to pins 1 and 4 of the SIP header respectively.

### Getting Started

Starting-up is simple, just apply power and wait for the ezMOUSE to respond. Mice manufacturers' state that typical mouse start-up times can range from a minimum of 300 milliseconds to 1 second. During this time the mouse goes through internal calibrations, initialization, and system checks. It is recommended that no commands be sent to the mouse during this time. Because of this, the ezMOUSE will not respond to any commands or pass them up to the mouse for the first second after power is applied. After that one second is past the ezMOUSE will initialize the mouse to ensure that it is ready for use. Once the initialization is complete the ezMOUSE will send out one of three responses. If the initialization was successful the ezMOUSE will respond with an acknowledge (ACK) which has a decimal value of 6. If the connected mouse responded but the data was incorrect the ezMOUSE will respond with a negative-acknowledge (NAK) which has a decimal value of 21. If this happens, check the connections and then send the initialization command again (see the Commands section). If the connected mouse does not respond at all the ezMOUSE will respond with a CANCEL which means the command was cancelled. The CANCEL response has a decimal value of 24. You will only get a CANCEL response if the ezMOUSE detects no sign of mouse activity, as if it was not even plugged in. If you get a CANCEL response the problem may be greater than a loose connection. *NOTE - If you do not monitor the transmission from the ezMOUSE at startup you should send the Initialize command after 1.5 seconds has past to ensure the connected mouse is initialized properly and is ready for use.*

### Response Data Format

When you send a command to the ezMOUSE it will always come back with a response. All commands, except for Get Data, are single byte responses. They are ACK, NAK, and CANCEL which are all explained in the Getting Started section. If you receive an ACK the command was completed successfully, a NAK indicates a possible data problem and you should repeat the command, and a CANCEL indicates that the mouse is not responding at all and needs to be checked. As stated above, the only exception is the Get Data command. If this command is successfully completed then you will receive three data bytes. These three bytes are sent back-to-back so if you only receive one byte then it is either a NAK or CANCEL response indicating a problem. The first byte contains the status flags, the second byte is the movement in the X direction and the third byte is the movement in the Y direction. Figure 1 shows the two-dimensional movement of a mouse.

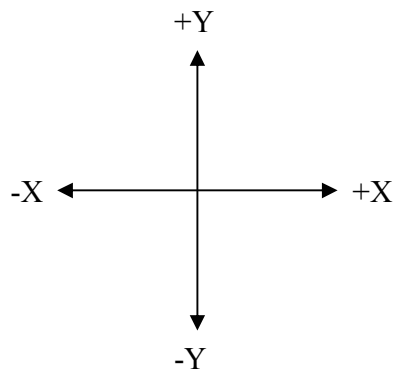


Figure 1

The following table shows the structure of the three data bytes:

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Byte 1	Y <sub>OV</sub>	X <sub>OV</sub>	Y <sub>S</sub>	X <sub>S</sub>	1	MB	RB	LB
Byte 2	X Movement							
Byte 3	Y Movement							

The first byte contains the status flags for the counters and the buttons and is defined as follows:  
Y<sub>OV</sub> – Y Overflow is set to 1 if more than 255 counts occurred since the last time the data was read. This flag is not used by all mice.

X<sub>OV</sub> – X Overflow is set to 1 if more than 255 counts occurred since the last time the data was read. This flag is not used by all mice.

Y<sub>S</sub> – Y Sign is set to 1 if movement was in the negative direction or 0 if movement was in the positive direction.

X<sub>S</sub> – X Sign is set to 1 if movement was in the negative direction or 0 if movement was in the positive direction.

1 – Bit 3 is always 1.

MB – Current status of middle button, set to 1 if the button is down or 0 if the button is up.

RB – Current status of right button, set to 1 if the button is down or 0 if the button is up.

LB – Current status of left button, set to 1 if the button is down or 0 if the button is up.

Byte 2 and byte 3 contain the X movement and Y movement data of the mouse respectively. The data contained in these bytes is reset every time the Get Data command is executed so this data represents the movement that occurred since the last Get Data command. The value in these bytes can range from 0-255, however some mice only use 8-bit signed counters instead of 9-bit so values from 0-127 are also common. If a mouse only has 8-bit counters then the overflow flags in the first byte are not used. The PS/2 protocol transmits this data from the mouse in 2's compliment form which is a common way to represent signed integers. The ezMOUSE converts this back for you so reading the data from the ezMOUSE requires no conversion; it is already in standard non-signed integer format. For X movement you simply look at the data in byte 2 and if the sign bit is 0 then add it to the mouse position in your application. If the sign bit is 1 then subtract it. Do the exact same thing for the Y position and you're done. Again, you can check the overflow flags to see if the values are greater than 255.

### Mouse Resolutions

You can program mice for four different resolutions of 8 counts, 4 counts, 2 counts, and 1 count per millimeter. The ezMOUSE gives you four commands to set these resolutions as shown in the Commands section. When the connected mouse is initialized on start-up, or when you issue the Initialization command, the connected mouse will default to 4 counts per millimeter. Depending on your application and how often you read the movement data you may need to change this default resolution. For example, if you want real fine mouse movement but don't have the ability to read the movement data often then you would want to select low resolution. On this setting the mouse will not count as much during movement by the user allowing you to read data less but still retain a fine resolution. Experimentation with your application will help you figure out which resolution is correct.

Your application can also scale the movements from the mouse if needed. For example, maybe you can only read the movement data once a second and low resolution still does not give you fine enough resolution. Your application could be setup to scale the mouse movement so, for example, you divide the mouse movement you read by 2 essentially halving the resolution. Experiment with this techniques if needed.

*NOTE: During testing it was found that optical mice are much more sensitive to movement than roller-ball style mice. When moving an optical and a roller-ball mouse the same distance the optical showed more counts. You can experiment with both types of mice as well to achieve the best results for your application.*

### Communications Protocol

The ezMOUSE uses 9600 baud TTL level asynchronous serial communications to talk with your application. This serial communications is half-duplex on the ezMOUSE. You must not transmit data from your application to the ezMOUSE while it is transmitting data to your application. Doing so will cause the ezMOUSE to misinterpret your command or miss it entirely. When receiving data from the ezMOUSE your application must wait until the entire stop bit time of the last byte that you expect to receive for the command you sent is read before transmitting another command to the ezMOUSE.

### Timing

When you send a command to the ezMOUSE that command (assuming its valid) is converted to a PS/2 command that then its communicated to the connected mouse. This communication takes time and each command varies based on how much work the connected mouse has to do. Figure 2 shows a typical timing diagram showing the command being transmitted to the ezMOUSE and the delay before the ezMOUSE sends back data. The values for  $T_{dly}$  are as follows (these are typical times and may vary slightly on different types of mice):

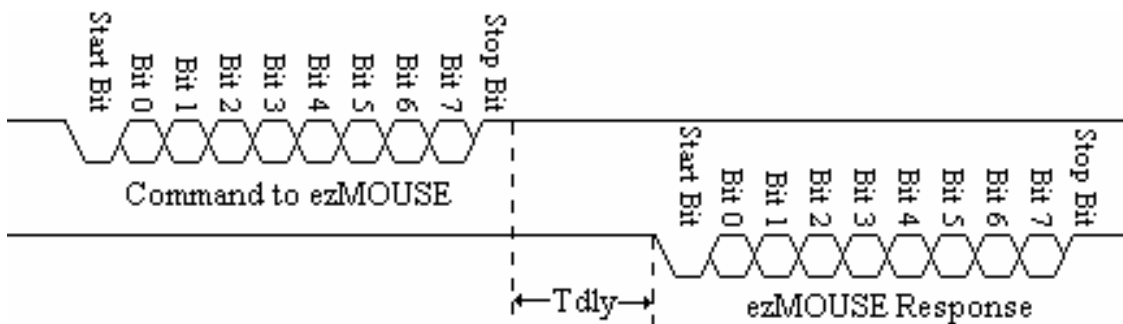


Figure 2

Values for  $T_{dly}$  -

Receive a Get Data command until the first byte of three is transmitted -  
Typically 5.5 milliseconds

Receive an Initialization command until an ACK is transmitted -  
Typically 350 milliseconds

Receive a Resolution command (all four) until an ACK is transmitted –  
Typically 4.5 milliseconds

Receive any command and receive a NAK from either the command being bad or the data from the connected mouse not being correct –  
Typically 1.5 milliseconds

Receive any command and receive a CANCEL because there is no activity from the connected mouse –  
Typically 30 milliseconds

After power on, receiving a response from the ezMOUSE after initializing the connected mouse (note: the diagram in Figure 2 does not apply to this timing since no command is sent to the ezMOUSE upon start-up) –  
Typically 1.5 seconds

## OPERATION COMMANDS

### Initialize

This command initializes the mouse for use. This command is automatically sent by the ezMOUSE when power is first applied. You may need to send it again if you receive a NAK or you hot-swap a mouse (change mice without removing power). This command must be sent in order for the ezMOUSE and connected mouse to work properly together. If it is not sent to a new mouse then operation is not guaranteed.

Syntax: code

- **Code** is the character that tells ezMOUSE that this is a initialize command. This code is a capital "I" (decimal=73).

### Get Data

This command retrieves the movement data since the last time this command was sent and also retrieves the current state of the buttons. The data is sent in three bytes (see Response Data Format section for more detail).

Syntax: code

- **Code** is the character that tells ezMOUSE that this is a get data command. This code is a capital "G" (decimal=71).

### Super High-Resolution

This command sets the mouse for a resolution of 8 counts per millimeter (see Mouse Resolutions section for more information).

Syntax: code

- **Code** is the character that tells ezMOUSE that this is a set super high-resolution command. This code is a capital "S" (decimal=83).

### High-Resolution

This command sets the mouse for a resolution of 4 counts per millimeter (see Mouse Resolutions section for more information).

Syntax: code

- **Code** is the character that tells ezMOUSE that this is a set high-resolution command. This code is a capital "H" (decimal=72).

### Medium-Resolution

This command sets the mouse for a resolution of 2 counts per millimeter (see Mouse Resolutions section for more information).

Syntax: code

- **Code** is the character that tells ezMOUSE that this is a set medium-resolution command. This code is a capital "M" (decimal=77).



**Low-Resolution**

This command sets the mouse for a resolution of 1 count per millimeter (see Mouse Resolutions section for more information).

Syntax: code

- **Code** is the character that tells ezMOUSE that this is a set low-resolution command. This code is a capital “L” (decimal=76).

**NOTE FOR ALL RESIDENTS OF THE EUROPEAN UNION**



**This symbol on the product or on its packaging indicates that this product must not be disposed of with other household waste. Instead, it is your responsibility to dispose of your waste equipment by handing it over to designated collection point for the recycling of waste electrical and electric equipment. The separate collection and recycling of your waste equipment at the time of disposal will help to conserve natural resources and ensure that it is recycled in a manner that protects human health and environment. For more information about where you can drop off your waste equipment for recycling, please contact your local city office or your local household waste disposal service.**

### **Limited Warranty**

Multilabs warrants to the original consumer purchaser of this product that, for a period of 90 days from the date of purchase, this product will be free from defects in material and workmanship and will perform in substantial conformity to the description of the product in this Owner's Manual. This warranty shall not apply to defects or errors caused by misuse or neglect. If the product is found to be defective in material or workmanship or if the product does not perform as warranted above during the warranty period, Multilabs will repair it, replace it, or refund the purchase price. The repair, replacement, or refund that is provided for above shall be the full extent of Multilabs's liability with respect to this product. For repair or replacement during the warranty period, contact Multilabs customer service by phone at (949) 458-7625 or email at [support@multilabs.net](mailto:support@multilabs.net) to receive an return authorization number and return instructions.

### **Limitations**

THE ABOVE WARRANTY IS IN LIEU OF AND MULTILABS DISCLAIMS ALL OTHER WARRANTIES, WHETHER ORAL OR WRITTEN, EXPRESS OR IMPLIED, INCLUDING ANY WARRANTY OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. ANY IMPLIED WARRANTY, INCLUDING ANY WARRANTY OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE, WHICH MAY NOT BE DISCLAIMED OR SUPPLANTED AS PROVIDED ABOVE SHALL BE LIMITED TO THE 90 DAYS OF THE EXPRESS WARRANTY ABOVE. NO OTHER REPRESENTATION OR CLAIM OF ANY NATURE BY ANY PERSON SHALL BE BINDING UPON MULTILABS OR MODIFY THE TERMS OF THE ABOVE WARRANTY AND DISCLAIMER. IN NO EVENT SHALL MULTILABS BE LIABLE FOR SPECIAL, INCIDENTAL, CONSEQUENTIAL OR OTHER DAMAGES RESULTING FROM THE POSSESSION OR USE OF THIS PRODUCT, INCLUDING WITHOUT LIMITATION DAMAGE TO PROPERTY AND, TO THE EXTENT PERMITTED BY LAW, PERSONAL INJURY, EVEN IF MULTILABS KNEW OR SHOULD HAVE KNOWN OF THE POSSIBILITY OF SUCH DAMAGES.

Some states do not allow limitations on how long an implied warranty lasts and/or the exclusion or limitation of damages, in which case the above limitations and/or exclusions may not apply to you. You may also have other legal rights, which may vary from state to state.