NAME

mouse − serial mouse interface

CONFIGURATION

Serial mice are connected to a serial RS232/V24 dialout line, see **ttyS**(4) for a description.

DESCRIPTION

Introduction

The pinout of the usual 9 pin plug as used for serial mice is:

pin	name	used for
2	RX	Data
3	TX	−12 V, Imax = 10 mA
4	DTR	+12 V, Imax = 10 mA
7	RTS	+12 V, Imax = 10 mA
5	GND	Ground

This is the specification, in fact 9 V suffices with most mice.

The mouse driver can recognize a mouse by dropping RTS to low and raising it again. About 14 ms later the mouse will send 0x4D ('M') on the data line. After a further 63 ms, a Microsoft-compatible 3-button mouse will send 0x33 ('3').

The relative mouse movement is sent as *dx* (positive means right) and *dy* (positive means down). Various mice can operate at different speeds. To select speeds, cycle through the speeds 9600, 4800, 2400, and 1200 bit/s, each time writing the two characters from the table below and waiting 0.1 seconds. The following table shows available speeds and the strings that select them:

bit/s	string
9600	*q
4800	*p
2400	*o
1200	*n

The first byte of a data packet can be used for synchronization purposes.

Microsoft protocol

The **Microsoft** protocol uses 1 start bit, 7 data bits, no parity and one stop bit at the speed of 1200 bits/sec. Data is sent to RxD in 3-byte packets. The *dx* and *dy* movements are sent as two's-complement, *lb* (*rb*) are set when the left (right) button is pressed:

byte	d6	d5	d4	d3	d2	d1	d0
1	1	lb	rb	dy7	dy6	dx7	dx6
2	0	dx5	dx4	dx3	dx2	dx1	dx0
3	0	dy5	dy4	dy3	dy2	dy1	dy0

3-button Microsoft protocol

Original Microsoft mice only have two buttons. However, there are some three button mice which also use the Microsoft protocol. Pressing or releasing the middle button is reported by sending a packet with zero movement and no buttons pressed. (Thus, unlike for the other two buttons, the status of the middle button is not reported in each packet.)

Logitech protocol

Logitech serial 3-button mice use a different extension of the Microsoft protocol: when the middle button is up, the above 3-byte packet is sent. When the middle button is down a 4-byte packet is sent, where the 4th byte has value 0x20 (or at least has the 0x20 bit set). In particular, a press of the middle button is reported as 0,0,0,0x20 when no other buttons are down.

Mousesystems protocol

The **Mousesystems** protocol uses 1 start bit, 8 data bits, no parity and two stop bits at the speed of 1200 bits/sec. Data is sent to RxD in 5-byte packets. *dx* is sent as the sum of the two two's-complement values, *dy* is send as negated sum of the two two's-complement values. *lb* (*mb*, *rb*) are cleared when the left

(middle, right) button is pressed:

byte	d7	d6	d5	d4	d3	d2	d1	d0
1	1	0	0	0	0	lb	mb	rb
2	0	dxa6	dxa5	dxa4	dxa3	dxa2	dxa1	dxa0
3	0	dya6	dya5	dya4	dya3	dya2	dya1	dya0
4	0	dxb6	dxb5	dxb4	dxb3	dxb2	dxb1	dxb0
5	0	dyb6	dyb5	dyb4	dyb3	dyb2	dyb1	dyb0

Bytes 4 and 5 describe the change that occurred since bytes 2 and 3 were transmitted.

Sun protocol

The **Sun** protocol is the 3-byte version of the above 5-byte Mousesystems protocol: the last two bytes are not sent.

MM protocol

The **MM** protocol uses 1 start bit, 8 data bits, odd parity and one stop bit at the speed of 1200 bits/sec. Data is sent to RxD in 3-byte packets. *dx* and *dy* are sent as single signed values, the sign bit indicating a negative value. *lb* (*mb*, *rb*) are set when the left (middle, right) button is pressed:

byte	d7	d6	d5	d4	d3	d2	d1	d0
1	1	0	0	dxs	dys	lb	mb	rb
2	0	dx6	dx5	dx4	dx3	dx2	dx1	dx0
3	0	dy6	dy5	dy4	dy3	dy2	dy1	dy0

FILES

/dev/mouse

A commonly used symbolic link pointing to a mouse device.

SEE ALSO

ttyS(4), **gpm**(8)

COLOPHON

This page is part of release 5.02 of the Linux *man-pages* project. A description of the project, information about reporting bugs, and the latest version of this page, can be found at https://www.kernel.org/doc/man−pages/.