NAME

rl - RealTek 8129/8139 Fast Ethernet device driver

SYNOPSIS

To compile this driver into the kernel, place the following lines in your kernel configuration file:

```
device miibus
device rl
```

Alternatively, to load the driver as a module at boot time, place the following line in loader.conf(5):

```
if_rl_load="YES"
```

DESCRIPTION

The **rl** driver provides support for PCI Ethernet adapters and embedded controllers based on the RealTek 8129 and 8139 Fast Ethernet controller chips.

The RealTek 8129/8139 series controllers use bus master DMA but do not use a descriptor-based data transfer mechanism. The receiver uses a single fixed size ring buffer from which packets must be copied into mbufs. For transmission, there are only four outbound packet address registers which require all outgoing packets to be stored as contiguous buffers. Furthermore, outbound packet buffers must be longword aligned or else transmission will fail.

The 8129 differs from the 8139 in that the 8139 has an internal PHY which is controlled through special direct access registers whereas the 8129 uses an external PHY via an MII bus. The 8139 supports both 10 and 100Mbps speeds in either full or half duplex. The 8129 can support the same speeds and modes given an appropriate PHY chip.

Note: support for the 8139C+ chip is provided by the re(4) driver.

The **rl** driver supports the following media types:

autoselect	Enable autoselection of the media type and options. This is only supported if the PHY chip attached to the RealTek controller supports NWAY autonegotiation. The user can manually override the autoselected mode by adding media options to the */etc/rc.conf* file.
10baseT/UTP	Set 10Mbps operation. The *mediaopt* option can also be used to select either *full-duplex* or *half-duplex* modes.

100baseTX	Set 100Mbps (Fast Ethernet) operation. The *mediaopt* option can also be used to select either *full-duplex* or *half-duplex* modes.

The **rl** driver supports the following media options:

full-duplex	Force full duplex operation.
half-duplex	Force half duplex operation.

Note that the 100baseTX media type is only available if supported by the adapter. For more information on configuring this device, see ifconfig(8).

HARDWARE

Adapters supported by the **rl** driver include:

- Accton "Cheetah" EN1207D (MPX 5030/5038; RealTek 8139 clone)
- Allied Telesyn AT2550
- Allied Telesyn AT2500TX
- Belkin F5D5000
- BUFFALO (Melco INC.) LPC-CB-CLX (CardBus)
- Compaq HNE-300
- CompUSA no-name 10/100 PCI Ethernet NIC
- Corega FEther CB-TXD
- Corega FEtherII CB-TXD
- D-Link DFE-520TX (rev. C1)
- D-Link DFE-528TX
- D-Link DFE-530TX+
- D-Link DFE-538TX
- D-Link DFE-690TXD
- Edimax EP-4103DL CardBus
- Encore ENL832-TX 10/100 M PCI
- Farallon NetLINE 10/100 PCI
- Genius GF100TXR
- GigaFast Ethernet EE100-AXP
- KTX-9130TX 10/100 Fast Ethernet
- LevelOne FPC-0106TX
- Longshine LCS-8038TX-R
- NDC Communications NE100TX-E
- Netronix Inc. EA-1210 NetEther 10/100
- Nortel Networks 10/100BaseTX

- OvisLink LEF-8129TX
- OvisLink LEF-8139TX
- Peppercon AG ROL-F
- Planex FNW-3603-TX
- Planex FNW-3800-TX
- SMC EZ Card 10/100 PCI 1211-TX
- SOHO (PRAGMATIC) UE-1211C

LOADER TUNABLES

dev.rl.%unit.prefer_iomap

This tunable controls which register mapping should be used on the specified device. A non-zero value enables I/O space register mapping. For controllers that have no I/O space register mapping this tunable should be set to 0 to use memory space register mapping. The default value is 1 to use I/O space register mapping.

dev.rl.%unit.twister_enable

Non-zero value enables the long cable tuning on the specified device. Disabled by default.

DIAGNOSTICS

rl%d: couldn’t map memory A fatal initialization error has occurred.

rl%d: couldn’t map interrupt A fatal initialization error has occurred.

rl%d: watchdog timeout The device has stopped responding to the network, or there is a problem with the network connection (cable).

rl%d: no memory for rx list The driver failed to allocate an mbuf for the receiver ring.

rl%d: no memory for tx list The driver failed to allocate an mbuf for the transmitter ring when allocating a pad buffer or collapsing an mbuf chain into a cluster.

rl%d: chip is in D3 power state -- setting to D0 This message applies only to adapters which support power management. Some operating systems place the controller in low power mode when shutting down, and some PCI BIOSes fail to bring the chip out of this state before configuring it. The controller loses all of its PCI configuration in the D3 state, so if the BIOS does not set it back to full power mode in time, it will not be able to configure it correctly. The driver tries to detect this condition and bring the adapter back to the D0 (full power) state, but this may not be enough to return the driver to a fully operational condition. If you see this message at boot time and the driver fails to attach the device as a network interface, you will have to perform second warm boot to have the device properly configured.

Note that this condition only occurs when warm booting from another operating system. If you power down your system prior to booting FreeBSD, the card should be configured correctly.

SEE ALSO

altq(4), arp(4), miibus(4), netintro(4), ng_ether(4), polling(4), ifconfig(8)

The RealTek 8129, 8139 and 8139C+ datasheets, https://www.realtek.com.

HISTORY

The **rl** device driver first appeared in FreeBSD 3.0.

AUTHORS

The **rl** driver was written by Bill Paul <*wpaul@ctr.columbia.edu*>.

BUGS

Since outbound packets must be longword aligned, the transmit routine has to copy an unaligned packet into an mbuf cluster buffer before transmission. The driver abuses the fact that the cluster buffer pool is allocated at system startup time in a contiguous region starting at a page boundary. Since cluster buffers are 2048 bytes, they are longword aligned by definition. The driver probably should not be depending on this characteristic.

The RealTek data sheets are of especially poor quality, and there is a lot of information missing particularly concerning the receiver operation. One particularly important fact that the data sheets fail to mention relates to the way in which the chip fills in the receive buffer. When an interrupt is posted to signal that a frame has been received, it is possible that another frame might be in the process of being copied into the receive buffer while the driver is busy handling the first one. If the driver manages to finish processing the first frame before the chip is done DMAing the rest of the next frame, the driver may attempt to process the next frame in the buffer before the chip has had a chance to finish DMAing all of it.

The driver can check for an incomplete frame by inspecting the frame length in the header preceding the actual packet data: an incomplete frame will have the magic length of 0xFFF0. When the driver encounters this value, it knows that it has finished processing all currently available packets. Neither this magic value nor its significance are documented anywhere in the RealTek data sheets.