

Guía de Usuario del Controlador Tulip

Jeff Garzik

jgarzik@mandrakesoft.com

Guía de Usuario del Controlador Tulip

por Jeff Garzik

Copyright © 2001 por Jeff Garzik

Esta documentación es software libre; puedes redistribuirla y/o modificarla bajo los términos de la GNU General Public License tal como ha sido publicada por la Free Software Foundation; por la versión 2 de la licencia, o (a tu elección) por cualquier versión posterior.

Este programa es distribuido con la esperanza de que sea útil, pero SIN NINGUNA GARANTIA; sin incluso la garantía implicada de COMERCIALIZACION o ADECUACION PARA UN PROPOSITO PARTICULAR. Para más detalles refiérase a la GNU General Public License.

Debería de haber recibido una copia de la GNU General Public License con este programa; si no es así, escriba a la Free Software Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

Para más detalles véase el archivo COPYING en la distribución fuente de Linux.

Tabla de contenidos

1. Introducción	1
2. Compatibilidad de Controlador	2
3. Parámetros específicos de la placa	3
4. Operación del Controlador	4
4.1. buffers de Anillo.....	4
4.2. Sincronización.....	4
5. Errores	5
6. Historia de Cambios del Controlador	6
6.1. Versión 0.9.14 (20 de Febrero, 2001).....	6
6.2. Versión 0.9.11 (3 de Noviembre, 2000)	6
6.3. Versión 0.9.10 (6 de Septiembre, 2000).....	6
6.4. Versión 0.9.9 (11 de Agosto, 2000)	6
6.5. Versión 0.9.8 (13 de Julio, 2000)	6
6.6. Versión 0.9.7 (17 de Junio, 2000)	7
6.7. Versión 0.9.6 (31 de Mayo, 2000).....	7
6.8. Versión 0.9.5 (31 de Mayo, 2000).....	7
6.9. Versión 0.9.4.3 (14 de Abril, 2000).....	7
6.10. Versión 0.9.4.2 (31 de Marzo, 2000).....	7
6.11. Versión 0.9.4.1 (18 de Marzo, 2000).....	8
7. Sobre la traducción.....	9

Capítulo 1. Introducción

EL controlador de la tarjeta Ethernet Tulip es mantenido por Jeff Garzik (<jgarzik@mandrakesoft.com>).

El controlador Tulip fue desarrollado por Donald Becker y cambiado por Jeff Garzik, Takashi Manabe y unos miles más.

Para 2.4.x y núcleos posteriores, el controlador Linux Tulip driver está disponible en <http://sourceforge.net/projects/tulip/>

Este controlador es para la interfaz del adaptador Ethernet Digital "Tulip" Debería de trabajar con la mayoría de las tarjetas de red/chips basados en DEC 21*4*, al igual que con los chips similares de Lite-On (PNIC) y Macronix (MXIC) y ASIX.

El autor original puede ser encontrado como becker@scyld.com, o C/O Scyld Computing Corporation, 410 Severn Ave., Suite 210, Annapolis MD 21403

Información adicional del tulip.c de Donald Becker está disponible en <http://www.scyld.com/network/tulip.html>

Capítulo 2. Compatibilidad de Controlador

Este controlador de dispositivo está diseñado para el chip de DEC "Tulip", los controladores ethernet de un sólo chip de Digital para PCI (no propietario de Intel). Los miembros soportados de la familia son 21040, 21041, 21140, 21140A, 21142, y 21143. Los chips similares de Lite-On, Macronics, ASIX, Compex y otros listados más abajo también son soportados.

Estos chips son usados en al menos 140 diseños únicos de placas PCI. El gran número de chips y diseños de placas soportadas es el motivo para el tamaño y complejidad del controlador. La mayoría del aumento de la complejidad está en la configuración de la placa y el código de selección de medio. Hay muy poco incremento en la longitud del camino crítico operacional.

Capítulo 3. Parámetros específicos de la placa

Los dispositivos del bus PCI son configurados por el sistema en tiempo de arranque, por lo tanto ningún jumper tiene que ser establecido en la placa. Preferiblemente la BIOS del sistema debería de asignar la señal PCI INTA o en otro caso una línea IRQ del sistema sin usar.

Algunas placas tienen tablas de EEPROMS con una entrada de medio por defecto. La opción por defecto de la fábrica es "autoselect". Esto debería de ser sólo sobrescrito cuando se usan conexiones transmisoras sin latido de enlace, ej. 10base2 o AUI, o (¡raramente!) para forzar full-duplex cuando es usado con compañeros de enlace viejos que no realizan autonegociación.

Capítulo 4. Operación del Controlador

4.1. buffers de Anillo

El Tulip puede usar buffers de anillo o lista de descriptores Tx y Rx. Este controlador usa anillos estáticamente asignados de descriptores Rx y Tx, establecidos en tiempo de compilación por `RX/TX_RING_SIZE`. Esta versión del controlador asigna `skbuffs` para los buffers de anillo Rx en tiempo de `open()` y pasa el campo `skb-data` al Tulip cuando recibe los buffers de datos. Cuando un frame que llega es menor que `RX_COPYBREAK` bytes de grande, un nuevo `skbuff` es asignado y el frame es copiado en el nuevo `skbuff`. Cuando el frame que llega es más grande, el `skbuff` es pasado directamente a la pila de protocolos y reemplazado por un `skbuff` nuevamente asignado.

El valor `RX_COPYBREAK` es escogido para equilibrar la memoria gastada usando un `skbuff` lleno para los frames pequeños vs. los costes de copiado para los frames grandes. Para los frames pequeños los costes de copiado son insignificantes (considerando que estamos precargando la caché con inmediatamente con información útil de la cabecera). Para frames grandes los costes de copiado no son triviales, y la copia más grande quizás borre datos útiles de la caché. Un aspecto sutil de esta elección es que el Tulip sólo recibe en buffers de palabras grandes (64 bits) alineadas, esto es, la cabecera IP con un desplazamiento de 14 no es una palabra grande alineada para un procesamiento más amplio. Los frames copiados son puestos en el nuevo `skbuff` con un desplazamiento de "+2", entonces copiando tiene el efecto beneficioso de alinear la cabecera IP y precargar la caché.

4.2. Sincronización

El controlador se ejecuta como dos flujos de control de un sólo hilo independientes. Uno es la rutina de enviar paquetes, que fuerza el uso de un hilo simple por medio de la flag `dev->tbusy`. El otro hilo es el manejador de interrupciones, que es un hilo simple por culpa del hardware y de otro software.

El hilo de enviar paquetes tiene control parcial sobre el anillo Tx y sobre la bandera `'dev->tbusy'`. Establece la bandera `tbusy` cuando está encolando un paquete Tx. Si el siguiente slot de la cola está vacío, limpia la bandera `tbusy` cuando ha acabado y en otro caso establece la bandera `'tp->tx_full'`.

El manejador de interrupciones tiene un control exclusivo sobre el anillo Tx y las estadísticas grabadas para el anillo Tx. (La interrupción Tx-done no puede ser desconectada selectivamente, por lo tanto no podemos eliminar la interrupción sobre nosotros teniendo la rutina Tx recogiendo las estadísticas Tx.) Después de recoger las estadísticas, marca la cola de entrada como vacía estableciendo la `'base'` a cero. Si la bandera `'tp->tx_full'` está establecida, entonces limpia las banderas `tx_full` y `tbusy`.

Capítulo 5. Errores

Los viejos libros de datos DEC eran ligeros en los detalles. El libro de datos 21040 pedía que CSR13, CSR14 y CSR15 deberían de ser el último registro del conjunto CSR12-15 escrito. Ummm, ¿cómo es esto posible?

El formato DEC SR0M está muy mal diseñado y no precisamente definido, llevando a la parte de selección del medio la cabecera desperdiciada posterior. Algunas placas no tienen las tablas de medios EEPROM y necesitan ser arregladas. Peor, otras placas usan el kit de diseño DEC de tablas de medios cuando no es correcto para sus placas.

No podemos usar las interrupciones MII porque no está definido el pin GPIO para atacarlas. El status del transmisor MII es preguntado usando un cronómetro del núcleo.

Capítulo 6. Historia de Cambios del Controlador

6.1. Versión 0.9.14 (20 de Febrero, 2001)

- Arreglados los problemas PNIC (Manfred Spraul)
- Añadida una nueva identificación PCI para la cometa Accton
- Soporte para tulips Davicom
- Arreglado un fallo en el chequeo de la eeprom
- Activadas las soluciones para los fallos de los primeros chipsets PCI
- IA64, soporte hppa csr0
- Soporte para los tipos de medio 5, 6
- Interpreta un bit más del tipo 3 del medio extendido 21142 SROM
- Añadido un retardo perdido en la lectura eeprom

6.2. Versión 0.9.11 (3 de Noviembre, 2000)

- Eliminados accesos extras al bus cuando se comparten interrupciones (prumpf)
- Barrier following ownership descriptor bit flip (prumpf)
- Barrera siguiendo la pertenencia del bit descriptor lanzado (prumpf)
- Fallos menores para >14 direcciones en los frames de configuración (prumpf)
- Reportado el latido del enlace al núcleo/espacio_de_usuario a través de netif_carrier_*. (kuznet)
- Mejor uso de spinlocks en set_rx_mode.
- Arreglados mensajes de error de fallo de la petición de recurso de E/S (DaveM catch)
- Fallo del asignamiento del manejador DMA.

6.3. Versión 0.9.10 (6 de Septiembre, 2000)

- Simple mitigación de interrupciones (via jamal)
- Más ids PCI

6.4. Versión 0.9.9 (11 de Agosto, 2000)

- Más ids PCI

6.5. Versión 0.9.8 (13 de Julio, 2000)

- Corregida comparación de signed/unsigned para el índice del frame dummy
- Borradas referencias fuera de fecha a struct enet_statistics

6.6. Versión 0.9.7 (17 de Junio, 2000)

- Limpieza de Cronómetros (Andrew Morton)
- Arreglada compilación en Alpha (alguien?)

6.7. Versión 0.9.6 (31 de Mayo, 2000)

- Deshecho al parche de la flag del soporte relacionado con 21143
- Añadido printk de depuración para HPPA/tabla_de_medios

6.8. Versión 0.9.5 (31 de Mayo, 2000)

- soporte HPPA (willy@puffingroup)
- bits CSR6 y limpieza de tulip.h (Chris Smith)
- Mejorados un poco los mensajes de depuración
- Añadido retardo antes de la escritura de CSR13 en t21142_start_nway
- Borrado el código sin utilizar ETHER_STATS
- Convertido 'extern inline' a 'static inline' en tulip.h (Chris Smith)
- Actualizadas las flags de soporte DS21143 en tulip_chip_info[]
- Usado spin_lock_irq, no _irqsave/restore, en tulip_start_xmit()
- Añadido bloqueo en set_rx_mode()
- Arreglada la carrera con el chip estableciendo el bit DescOwned (Hal Murray)
- Petición del 100% del espacio de recursos PIO y MMIO asignados a la tarjeta
- Borrado el mensaje de error del fallo pci_enable_device

6.9. Versión 0.9.4.3 (14 de Abril, 2000)

- Arreglado mod_timer (Hal Murray)
- Resurrección de PNIC2 (Chris Smith)

6.10. Versión 0.9.4.2 (31 de Marzo, 2000)

- Arreglado el manejo de 21041 CSR7, CSR13/14/15
- Unidas algunas ids PCI desde tulip 0.91x
- Unidas algunas banderas HAS_xxx y parámetros de banderas desde tulip 0.91x
- Arreglado de asm/io.h (remitido por varios) y limpieza
- s/HAS_NWAY143/HAS_NWAY/
- Limpieza del modo de reporte 21041
- Pequeñas limpiezas en el código

6.11. Versión 0.9.4.1 (18 de Marzo, 2000)

- Finalizada la conversión PCI DMA (davem)
- No netif_start_queue() al final de tulip_tx_timeout() (kuznet)
- Arreglado PCI DMA (kuznet)
- Limpieza del código eeprom.c
- Borrado el Xircom Tulip crudo

Capítulo 7. Sobre la traducción

Este documento es la traducción de "Tulip Driver User's Guide", documento que acompaña al código del núcleo de Linux, versión 2.4.18.

Este documento ha sido traducido por Rubén Melcón <melkon@terra.es>; y es publicado por el Proyecto Lucas (<http://lucas.hispalinux.es>)

Versión de la traducción 0.04 (Julio de 2002).

Si tienes comentarios sobre la traducción, ponte en contacto con Rubén Melcón <melkon@terra.es>