

[PATCH 475] HP300 LANCE

Source: <http://linux.derkeiler.com/Mailing-Lists/Kernel/2004-10/10221.html>

From: Geert Uytterhoeven (geert_at_linux-m68k.org)

Date: 10/31/04

Date: Sun, 31 Oct 2004 11:03:30 +0100

To: Linus Torvalds <torvalds@osdl.org>, Andrew Morton <akpm@osdl.org>, Jeff Garzik <jgarzik@pobox.com>

HP300 LANCE updates from Kars de Jong:

- Updated HP LANCE driver to use the new DIO semantics
- If only HP LANCE or MVME147 LANCE is selected, enable compile-time choice of LANCE register access. If both are defined, go through the function pointer
- Added support for CONFIG_NET_POLL_CONTROLLER
- Fixed problem with disabling board interrupts in `hplance_close()` which caused the driver to lock up

Signed-off-by: Kars de Jong <jongk@linux-m68k.org>

Signed-off-by: Geert Uytterhoeven <geert@linux-m68k.org>

--- linux-2.6.10-rc1/drivers/net/7990.c 2004-04-04 12:14:30.000000000 +0200

+++ linux-m68k-2.6.10-rc1/drivers/net/7990.c 2004-07-14 13:18:54.000000000 +0200

@@ -14,7 +14,6 @@

*/

#include <linux/crc32.h>

#include <linux/delay.h>

~#include <linux/dio.h>

#include <linux/errno.h>

#include <linux/netdevice.h>

#include <linux/etherdevice.h>

@@ -30,6 +29,7 @@

#include <linux/slab.h>

#include <linux/string.h>

#include <linux/skbuff.h>

+#include <linux/irq.h>

/* Used for the temporal inet entries and routing */

#include <linux/socket.h>

@@ -38,18 +38,58 @@

#include <asm/io.h>

#include <asm/dma.h>

#include <asm/pgtable.h>

+#ifdef CONFIG_HP300

+#include <asm/blinken.h>

+#endif

```

#include "7990.h"

#define WRITERAP(lp,x) out_be16(lp->base + LANCE_RAP, (x))
#define WRITERDP(lp,x) out_be16(lp->base + LANCE_RDP, (x))
#define READRDP(lp) in_be16(lp->base + LANCE_RDP)
+
+#if defined(CONFIG_HPLANCE) || defined(CONFIG_HPLANCE_MODULE)
+#include "hplance.h"
+
+#undef WRITERAP
+#undef WRITERDP
+#undef READRDP
+
+#if defined(CONFIG_MVME147_NET) || defined(CONFIG_MVME147_NET_MODULE)
+
+/* Lossage Factor Nine, Mr Sulu. */
-#define WRITERAP(x) (lp->writerap(lp,x))
-#define WRITERDP(x) (lp->writerdp(lp,x))
-#define READRDP() (lp->readrdp(lp))
-/* These used to be ll->rap = x, ll->rdp = x, and (ll->rdp). Sigh.
- * If you want to switch them back then
- * #define DECLARE_LL volatile struct lance_regs *ll = lp->ll
- */
-#define DECLARE_LL /* nothing to declare */
+#define WRITERAP(lp,x) (lp->writerap(lp,x))
+#define WRITERDP(lp,x) (lp->writerdp(lp,x))
+#define READRDP(lp) (lp->readrdp(lp))
+
+#else
+
+/* These inlines can be used if only CONFIG_HPLANCE is defined */
+static inline void WRITERAP(struct lance_private *lp, __u16 value)
+{
+ do {
+ out_be16(lp->base + HPLANCE_REGOFF + LANCE_RAP, value);
+ } while ((in_8(lp->base + HPLANCE_STATUS) & LE_ACK) == 0);
+}
+
+static inline void WRITERDP(struct lance_private *lp, __u16 value)
+{
+ do {
+ out_be16(lp->base + HPLANCE_REGOFF + LANCE_RDP, value);
+ } while ((in_8(lp->base + HPLANCE_STATUS) & LE_ACK) == 0);
+}
+
+static inline __u16 READRDP(struct lance_private *lp)
+{
+ __u16 value;
+ do {
+ value = in_be16(lp->base + HPLANCE_REGOFF + LANCE_RDP);

```

Linux-Kernel: [PATCH 475] HP300 LANCE

```

+ } while ((in_8(lp->base + HPLANCE_STATUS) & LE_ACK) == 0);
+ return value;
+}
+
+##endif
+##endif /* CONFIG_HPLANCE || CONFIG_HPLANCE_MODULE */

/* debugging output macros, various flavours */
/* #define TEST_HITS */
@@ -79,19 +119,18 @@
{
    volatile struct lance_init_block *aib = lp->lance_init_block;
    int leptr;
- DECLARE_LL;

    leptr = LANCE_ADDR (aib);

- WRITERAP(LE_CSR1); /* load address of init block */
- WRITERDP(leptr & 0xFFFF);
- WRITERAP(LE_CSR2);
- WRITERDP(leptr >> 16);
- WRITERAP(LE_CSR3);
- WRITERDP(lp->busmaster_regval); /* set byteswap/ALEctrl/byte ctrl */
+ WRITERAP(lp, LE_CSR1); /* load address of init block */
+ WRITERDP(lp, leptr & 0xFFFF);
+ WRITERAP(lp, LE_CSR2);
+ WRITERDP(lp, leptr >> 16);
+ WRITERAP(lp, LE_CSR3);
+ WRITERDP(lp, lp->busmaster_regval); /* set byteswap/ALEctrl/byte ctrl */

    /* Point back to csr0 */
- WRITERAP(LE_CSR0);
+ WRITERAP(lp, LE_CSR0);
}

/* #define to 0 or 1 appropriately */
@@ -192,24 +231,23 @@
static int init_restart_lance (struct lance_private *lp)
{
    int i;
- DECLARE_LL;

- WRITERAP(LE_CSR0);
- WRITERDP(LE_C0_INIT);
+ WRITERAP(lp, LE_CSR0);
+ WRITERDP(lp, LE_C0_INIT);

    /* Need a hook here for sunlance ledma stuff */

    /* Wait for the lance to complete initialization */
- for (i = 0; (i < 100) && !(READRDP() & (LE_C0_ERR | LE_C0_IDON)); i++)

```

Linux-Kernel: [PATCH 475] HP300 LANCE

```

+ for (i = 0; (i < 100) && !(READRDP(lp) & (LE_C0_ERR | LE_C0_IDON)); i++)
    barrier();
- if ((i == 100) || (READRDP() & LE_C0_ERR)) {
- printk ("LANCE unopened after %d ticks, csr0=%4.4x.\n", i, READRDP());
+ if ((i == 100) || (READRDP(lp) & LE_C0_ERR)) {
+ printk ("LANCE unopened after %d ticks, csr0=%4.4x.\n", i, READRDP(lp));
    return -1;
}

    /* Clear IDON by writing a "1", enable interrupts and start lance */
- WRITERDP(LE_C0_IDON);
- WRITERDP(LE_C0_INEA | LE_C0_STRT);
+ WRITERDP(lp, LE_C0_IDON);
+ WRITERDP(lp, LE_C0_INEA | LE_C0_STRT);

    return 0;
}
@@ -218,11 +256,10 @@
{
    struct lance_private *lp = netdev_priv(dev);
    int status;
- DECLARE_LL;

    /* Stop the lance */
- WRITERAP(LE_CSR0);
- WRITERDP(LE_C0_STOP);
+ WRITERAP(lp, LE_CSR0);
+ WRITERDP(lp, LE_C0_STOP);

    load_csrs (lp);
    lance_init_ring (dev);
@@ -245,7 +282,6 @@
#ifdef TEST_HITS
    int i;
#endif
- DECLARE_LL;

#ifdef TEST_HITS
    printk ("[");
@@ -259,8 +295,10 @@
}
    printk ("]");
#endif
-
- WRITERDP(LE_C0_RINT | LE_C0_INEA); /* ack Rx int, reenable ints */
+#ifdef CONFIG_HP300
+ blinken_leds(0x40, 0);
+#endif
+ WRITERDP(lp, LE_C0_RINT | LE_C0_INEA); /* ack Rx int, reenable ints */
    for (rd = &ib->brx_ring [lp->rx_new]; /* For each Rx ring we own... */
        !((bits = rd->rmd1_bits) & LE_R1_OWN);

```

```

        rd = &ib->brx_ring [lp->rx_new]) {
@@ -321,10 +359,12 @@
        volatile struct lance_tx_desc *td;
        int i, j;
        int status;
- DECLARE_LL;

#ifdef CONFIG_HP300
+ blinken_leds(0x80, 0);
#endif
        /* csr0 is 2f3 */
- WRITERDP(LE_C0_TINT | LE_C0_INEA);
+ WRITERDP(lp, LE_C0_TINT | LE_C0_INEA);
        /* csr0 is 73 */

        j = lp->tx_old;
@@ -349,8 +389,8 @@
                printk("%s: Carrier Lost, trying %s\n",
                        dev->name, lp->tpe?"TPE":"AUI");
                /* Stop the lance */
- WRITERAP(LE_CSR0);
- WRITERDP(LE_C0_STOP);
+ WRITERAP(lp, LE_CSR0);
+ WRITERDP(lp, LE_C0_STOP);
                lance_init_ring (dev);
                load_csrs (lp);
                init_restart_lance (lp);
@@ -366,8 +406,8 @@
                printk ("%s: Tx: ERR_BUF|ERR_UFL, restarting\n",
                        dev->name);
                /* Stop the lance */
- WRITERAP(LE_CSR0);
- WRITERDP(LE_C0_STOP);
+ WRITERAP(lp, LE_CSR0);
+ WRITERDP(lp, LE_C0_STOP);
                lance_init_ring (dev);
                load_csrs (lp);
                init_restart_lance (lp);
@@ -393,7 +433,7 @@
        j = (j + 1) & lp->tx_ring_mod_mask;
    }
    lp->tx_old = j;
- WRITERDP(LE_C0_TINT | LE_C0_INEA);
+ WRITERDP(lp, LE_C0_TINT | LE_C0_INEA);
    return 0;
}

@@ -403,26 +443,25 @@
    struct net_device *dev = (struct net_device *)dev_id;
    struct lance_private *lp = netdev_priv(dev);
    int csr0;

```

```

- DECLARE_LL;

    spin_lock (&lp->devlock);

- WRITERAP(LE_CSR0); /* LANCE Controller Status */
- csr0 = READRDP();
+ WRITERAP(lp, LE_CSR0); /* LANCE Controller Status */
+ csr0 = READRDP(lp);

    PRINT_RINGS();

    if (!(csr0 & LE_C0_INTR)) { /* Check if any interrupt has */
- spin_lock (&lp->devlock);
+ spin_unlock (&lp->devlock);
        return IRQ_NONE; /* been generated by the Lance. */
    }

    /* Acknowledge all the interrupt sources ASAP */
- WRITERDP(csr0 & ~(LE_C0_INEA|LE_C0_TDMD|LE_C0_STOP|LE_C0_STRT|LE_C0_INIT));
+ WRITERDP(lp, csr0 & ~(LE_C0_INEA|LE_C0_TDMD|LE_C0_STOP|LE_C0_STRT|LE_C0_INIT));

    if ((csr0 & LE_C0_ERR)) {
        /* Clear the error condition */
- WRITERDP(LE_C0_BABL|LE_C0_ERR|LE_C0_MISS|LE_C0_INEA);
+ WRITERDP(lp, LE_C0_BABL|LE_C0_ERR|LE_C0_MISS|LE_C0_INEA);
    }

    if (csr0 & LE_C0_RINT)
@@ -440,7 +479,7 @@
        printk("%s: Bus master arbitration failure, status %4.4x.\n",
            dev->name, csr0);
        /* Restart the chip. */
- WRITERDP(LE_C0_STRT);
+ WRITERDP(lp, LE_C0_STRT);
    }

    if (lp->tx_full && netif_queue_stopped(dev) && (TX_BUFFS_AVAIL >= 0)) {
@@ -448,8 +487,8 @@
        netif_wake_queue (dev);
    }

- WRITERAP(LE_CSR0);
- WRITERDP(LE_C0_BABL|LE_C0_CERR|LE_C0_MISS|LE_C0_MERR|LE_C0_IDON|LE_C0_INEA);
+ WRITERAP(lp, LE_CSR0);
+ WRITERDP(lp,
LE_C0_BABL|LE_C0_CERR|LE_C0_MISS|LE_C0_MERR|LE_C0_IDON|LE_C0_INEA);

    spin_unlock (&lp->devlock);
    return IRQ_HANDLED;
@@ -459,7 +498,6 @@
{

```

Linux-Kernel: [PATCH 475] HP300 LANCE

```

    struct lance_private *lp = netdev_priv(dev);
    int res;
- DECLARE_LL;

    /* Install the Interrupt handler. Or we could shunt this out to specific drivers? */
    if (request_irq(lp->irq, lance_interrupt, 0, lp->name, dev))
@@ -475,13 +513,12 @@
int lance_close (struct net_device *dev)
{
    struct lance_private *lp = netdev_priv(dev);
- DECLARE_LL;

    netif_stop_queue (dev);

    /* Stop the LANCE */
- WRITERAP(LE_CSR0);
- WRITERDP(LE_C0_STOP);
+ WRITERAP(lp, LE_CSR0);
+ WRITERDP(lp, LE_C0_STOP);

    free_irq(lp->irq, dev);

@@ -504,7 +541,6 @@
    int entry, skblen, len;
    static int outs;
    unsigned long flags;
- DECLARE_LL;

    if (!TX_BUFFS_AVAIL)
        return -1;
@@ -540,7 +576,7 @@

    outs++;
    /* Kick the lance: transmit now */
- WRITERDP(LE_C0_INEA | LE_C0_TDMD);
+ WRITERDP(lp, LE_C0_INEA | LE_C0_TDMD);
    dev->trans_start = jiffies;
    dev_kfree_skb (skb);

@@ -604,7 +640,6 @@
    struct lance_private *lp = netdev_priv(dev);
    volatile struct lance_init_block *ib = lp->init_block;
    int stopped;
- DECLARE_LL;

    stopped = netif_queue_stopped(dev);
    if (!stopped)
@@ -613,8 +648,8 @@
    while (lp->tx_old != lp->tx_new)
        schedule();

```

```

- WRITERAP(LE_CSR0);
- WRITERDP(LE_C0_STOP);
+ WRITERAP(lp, LE_CSR0);
+ WRITERDP(lp, LE_C0_STOP);
    lance_init_ring (dev);

    if (dev->flags & IFF_PROMISC) {
@@ -630,4 +665,17 @@
        netif_start_queue (dev);
    }

+ #ifdef CONFIG_NET_POLL_CONTROLLER
+ void lance_poll(struct net_device *dev)
+ {
+     struct lance_private *lp = netdev_priv(dev);
+     +
+     spin_lock (&lp->devlock);
+     WRITERAP(lp, LE_CSR0);
+     WRITERDP(lp, LE_C0_STOP);
+     spin_unlock (&lp->devlock);
+     lance_interrupt(dev->irq, dev, NULL);
+ }
+ #endif
+
+ MODULE_LICENSE("GPL");
--- linux-2.6.10-rc1/drivers/net/7990.h 2002-01-15 19:24:07.000000000 +0100
+++ linux-m68k-2.6.10-rc1/drivers/net/7990.h 2004-07-14 13:18:54.000000000 +0200
@@ -14,11 +14,8 @@
#define _7990_H

/* The lance only has two register locations. We communicate mostly via memory. */
- struct lance_regs
- {
-     unsigned short rdp; /* Register Data Port */
-     unsigned short rap; /* Register Address Port */
- };
+ #define LANCE_RDP 0 /* Register Data Port */
+ #define LANCE_RAP 2 /* Register Address Port */

/* Transmit/receive ring definitions.
 * We allow the specific drivers to override these defaults if they want to.
@@ -104,7 +101,7 @@
struct lance_private
{
    char *name;
- volatile struct lance_regs *ll;
+ unsigned long base;
    volatile struct lance_init_block *init_block; /* CPU address of RAM */
    volatile struct lance_init_block *lance_init_block; /* LANCE address of RAM */

@@ -252,5 +249,8 @@

```

Linux-Kernel: [PATCH 475] HP300 LANCE

```

extern struct net_device_stats *lance_get_stats (struct net_device *dev);
extern void lance_set_multicast (struct net_device *dev);
extern void lance_tx_timeout(struct net_device *dev);
+#ifdef CONFIG_NET_POLL_CONTROLLER
+extern void lance_poll(struct net_device *dev);
+#endif

#endif /* ndef_7990_H */
--- linux-2.6.10-rc1/drivers/net/hplance.c 2004-04-04 12:14:46.000000000 +0200
+++ linux-m68k-2.6.10-rc1/drivers/net/hplance.c 2004-07-14 13:18:56.000000000 +0200
@@ -42,7 +42,6 @@
struct hplance_private {
    struct lance_private lance;
    unsigned int scode;
- void *base;
};

/* function prototypes... This is easy because all the grot is in the
@@ -91,15 +90,17 @@
{
    int scode = dio_find(DIO_ID_LAN);

- if (!scode)
+ if (scode < 0)
        break;

        dio_config_board(scode);
        hplance_init(dev, scode);
        if (!register_netdev(dev)) {
+#ifdef MODULE
            struct hplance_private *lp = netdev_priv(dev);
            lp->next_module = root_hplance_dev;
            root_hplance_dev = lp;
+#endif
            return dev;
        }
        cleanup_card(dev);
@@ -112,11 +113,12 @@
static void __init hplance_init(struct net_device *dev, int scode)
{
    const char *name = dio_scodetname(scode);
- void *va = dio_scodetoviraddr(scode);
+ unsigned long pa = dio_scodetophysaddr(scode);
+ unsigned long va = (pa + DIO_VIRADDRBASE);
    struct hplance_private *lp;
    int i;

- printk("%s: %s; select code %d, addr", dev->name, name, scode);
+ printk(KERN_INFO "%s: %s; select code %d, addr", dev->name, name, scode);

    /* reset the board */

```

Linux-Kernel: [PATCH 475] HP300 LANCE

```

    out_8(va+DIO_IDOFF, 0xff);
@@ -123,9 +125,12 @@
    udelay(100); /* ariba! ariba! udelay! udelay! */

    /* Fill the dev fields */
- dev->base_addr = (unsigned long)va;
+ dev->base_addr = va;
    dev->open = &hplance_open;
    dev->stop = &hplance_close;
+#ifdef CONFIG_NET_POLL_CONTROLLER
+ dev->poll_controller = lance_poll;
+#endif
    dev->hard_start_xmit = &lance_start_xmit;
    dev->get_stats = &lance_get_stats;
    dev->set_multicast_list = &lance_set_multicast;
@@ -143,7 +148,7 @@

    lp = netdev_priv(dev);
    lp->lance.name = (char*)name; /* discards const, shut up gcc */
- lp->lance.ll = (struct lance_regs*)(va + HPLANCE_REGOFF);
+ lp->lance.base = va;
    lp->lance.init_block = (struct lance_init_block*)(va + HPLANCE_MEMOFF); /* CPU addr */
    lp->lance.lance_init_block = 0; /* LANCE addr of same RAM */
    lp->lance.busmaster_regval = LE_C3_BSWP; /* we're bigendian */
@@ -156,7 +161,6 @@
    lp->lance.rx_ring_mod_mask = RX_RING_MOD_MASK;
    lp->lance.tx_ring_mod_mask = TX_RING_MOD_MASK;
    lp->scode = scode;
- lp->base = va;
    printk(", irq %d\n", lp->lance.irq);
}

@@ -165,53 +169,49 @@
    */
    static void hplance_writerp(void *priv, unsigned short value)
    {
- struct hplance_private *lp = (struct hplance_private *)priv;
- struct hplance_reg *hpregs = (struct hplance_reg *)lp->base;
- do {
- lp->lance.ll->rap = value;
- } while ((hpregs->status & LE_ACK) == 0);
+ struct lance_private *lp = (struct lance_private *)priv;
+ do {
+ out_be16(lp->base + HPLANCE_REGOFF + LANCE_RAP, value);
+ } while ((in_8(lp->base + HPLANCE_STATUS) & LE_ACK) == 0);
    }

    static void hplance_writerdp(void *priv, unsigned short value)
    {
- struct hplance_private *lp = (struct hplance_private *)priv;
- struct hplance_reg *hpregs = (struct hplance_reg *)lp->base;

```

```

- do {
- lp->lance.ll->rdp = value;
- } while ((hpregs->status & LE_ACK) == 0);
+ struct lance_private *lp = (struct lance_private *)priv;
+ do {
+ out_be16(lp->base + HPLANCE_REGOFF + LANCE_RDP, value);
+ } while ((in_8(lp->base + HPLANCE_STATUS) & LE_ACK) == 0);
}

static unsigned short hplance_readrdp(void *priv)
{
- unsigned short val;
- struct hplance_private *lp = (struct hplance_private *)priv;
- struct hplance_reg *hpregs = (struct hplance_reg *)lp->base;
- do {
- val = lp->lance.ll->rdp;
- } while ((hpregs->status & LE_ACK) == 0);
- return val;
+ struct lance_private *lp = (struct lance_private *)priv;
+ __u16 value;
+ do {
+ value = in_be16(lp->base + HPLANCE_REGOFF + LANCE_RDP);
+ } while ((in_8(lp->base + HPLANCE_STATUS) & LE_ACK) == 0);
+ return value;
}

static int hplance_open(struct net_device *dev)
{
    int status;
- struct hplance_private *lp = netdev_priv(dev);
- struct hplance_reg *hpregs = (struct hplance_reg *)lp->base;
+ struct lance_private *lp = netdev_priv(dev);

    status = lance_open(dev); /* call generic lance open code */
    if (status)
        return status;
    /* enable interrupts at board level. */
- out_8(&(hpregs->status), LE_IE);
+ out_8(lp->base + HPLANCE_STATUS, LE_IE);

    return 0;
}

static int hplance_close(struct net_device *dev)
{
- struct hplance_private *lp = netdev_priv(dev);
- struct hplance_reg *hpregs = (struct hplance_reg *)lp->base;
- out_8(&(hpregs->status), 8); /* disable interrupts at boardlevel */
+ struct lance_private *lp = netdev_priv(dev);
+
+ out_8(lp->base + HPLANCE_STATUS, 0); /* disable interrupts at boardlevel */

```

Linux-Kernel: [PATCH 475] HP300 LANCE

```

    lance_close(dev);
    return 0;
}
--- linux-2.6.10-rc1/drivers/net/hplance.h 2001-10-22 01:51:53.000000000 +0200
+++ linux-m68k-2.6.10-rc1/drivers/net/hplance.h 2004-07-14 13:18:56.000000000 +0200
@@ -4,15 +4,10 @@
 */

/* Registers */
-struct hplance_reg
- {
-     u_char pad0;
-     volatile u_char id; /* DIO register: ID byte */
-     u_char pad1;
-     volatile u_char status; /* DIO register: interrupt enable */
- };
+#define HPLANCE_ID 0x01 /* DIO register: ID byte */
+#define HPLANCE_STATUS 0x03 /* DIO register: interrupt enable/status */

-/* Control and status bits for the hplance->status register */
+/* Control and status bits for the status register */
#define LE_IE 0x80 /* interrupt enable */
#define LE_IR 0x40 /* interrupt requested */
#define LE_LOCK 0x08 /* lock status register */
@@ -25,7 +20,7 @@
/* These are the offsets for the DIO regs (hplance_reg), lance_ioreg,
 * memory and NVRAM:
 */
-#define HPLANCE_IDOFF 0 /* board baseaddr, struct hplance_reg */
-#define HPLANCE_REGOFF 0x4000 /* struct lance_regs */
+#define HPLANCE_IDOFF 0 /* board baseaddr */
+#define HPLANCE_REGOFF 0x4000 /* lance registers */
#define HPLANCE_MEMOFF 0x8000 /* struct lance_init_block */
#define HPLANCE_NVRAMOFF 0xC008 /* etheraddress as one *nibble* per byte */
--- linux-2.6.10-rc1/drivers/net/mvme147.c 2004-02-18 20:36:08.000000000 +0100
+++ linux-m68k-2.6.10-rc1/drivers/net/mvme147.c 2004-07-14 13:18:56.000000000 +0200
@@ -18,7 +18,6 @@
/* Used for the temporal inet entries and routing */
#include <linux/socket.h>
#include <linux/route.h>
-#include <linux/dio.h>
#include <linux/netdevice.h>
#include <linux/etherdevice.h>
#include <linux/skbuff.h>
@@ -40,7 +39,6 @@
/* Our private data structure */
struct m147lance_private {
    struct lance_private lance;
- void *base;
    unsigned long ram;
};

```

Linux-Kernel: [PATCH 475] HP300 LANCE

@@ -51,9 +49,9 @@

*/

```
static int m147lance_open(struct net_device *dev);
static int m147lance_close(struct net_device *dev);
-static void m147lance_writerap(struct m147lance_private *lp, unsigned short value);
-static void m147lance_writerdp(struct m147lance_private *lp, unsigned short value);
-static unsigned short m147lance_readrdp(struct m147lance_private *lp);
+static void m147lance_writerap(struct lance_private *lp, unsigned short value);
+static void m147lance_writerdp(struct lance_private *lp, unsigned short value);
+static unsigned short m147lance_readrdp(struct lance_private *lp);
```

```
typedef void (*writerap_t)(void *, unsigned short);
```

```
typedef void (*writerdp_t)(void *, unsigned short);
```

@@ -122,7 +120,7 @@

}

```
lp->lance.name = (char*)name; /* discards const, shut up gcc */
- lp->lance.ll = (struct lance_regs*)(dev->base_addr);
+ lp->lance.base = dev->base_addr;
lp->lance.init_block = (struct lance_init_block*)(lp->ram); /* CPU addr */
lp->lance.lance_init_block = (struct lance_init_block*)(lp->ram); /* LANCE addr of same RAM */
lp->lance.busmaster_regval = LE_C3_BSWP; /* we're bigendian */
```

@@ -145,19 +143,19 @@

```
return dev;
```

}

```
-static void m147lance_writerap(struct m147lance_private *lp, unsigned short value)
```

```
+static void m147lance_writerap(struct lance_private *lp, unsigned short value)
```

```
{
```

```
- lp->lance.ll->rap = value;
```

```
+ out_be16(lp->base + LANCE_RAP, value);
```

```
}
```

```
-static void m147lance_writerdp(struct m147lance_private *lp, unsigned short value)
```

```
+static void m147lance_writerdp(struct lance_private *lp, unsigned short value)
```

```
{
```

```
- lp->lance.ll->rdp = value;
```

```
+ out_be16(lp->base + LANCE_RDP, value);
```

```
}
```

```
-static unsigned short m147lance_readrdp(struct m147lance_private *lp)
```

```
+static unsigned short m147lance_readrdp(struct lance_private *lp)
```

```
{
```

```
- return lp->lance.ll->rdp;
```

```
+ return in_be16(lp->base + LANCE_RDP);
```

```
}
```

```
static int m147lance_open(struct net_device *dev)
```

Gr{oetje,eeting}s,

Linux-Kernel: [PATCH 475] HP300 LANCE

Geert

--

Geert Uytterhoeven -- There's lots of Linux beyond ia32 -- geert@linux-m68k.org
In personal conversations with technical people, I call myself a hacker. But
when I'm talking to journalists I just say "programmer" or something like that.
-- Linus Torvalds

-

To unsubscribe from this list: send the line "unsubscribe linux-kernel" in
the body of a message to majordomo@vger.kernel.org
More majordomo info at <http://vger.kernel.org/majordomo-info.html>
Please read the FAQ at <http://www.tux.org/lkml/>