

DM9000 & 9000A Driver Programming Difference List

Function\Chip	DM9000A Driver V2.03	DM9000E Driver V1.28
dmfe_start_xmit()	<pre> if(db->Speed == 10) // Line#492 {if (db->tx_pkt_cnt >= 1) return 1;} else {if (db->tx_pkt_cnt >= 2) return 1;} /* packet counting */ db->tx_pkt_cnt++; db->stats.tx_packets++; db->stats.tx_bytes+=skb->len; if (db->Speed == 10) {if (db->tx_pkt_cnt >= 1) netif_stop_queue(dev);} else {if (db->tx_pkt_cnt >= 2) netif_stop_queue(dev);} /* Disable all interrupt */ iow(db, DM9KS_IMR, DM9KS_DISINTR); /* Set TX length to reg. 0xfc & 0xfd */ iow(db, DM9KS_TXPLL, (skb->len & 0xff)); iow(db, DM9KS_TXPLH, (skb->len >> 8) & 0xff); /* Move data to TX SRAM */ data_ptr = (char *)skb->data; /* ... */ #if !defined(ETRANS) // Line#536 /* Issue TX polling command */ iow(db, DM9KS_TCR, 0x1); /* Cleared after TX complete*/ #endif /* Saved the time stamp */ dev->trans_start = jiffies; db->cont_rx_pkt_cnt =0; /* ... */ </pre>	<pre> if (db->tx_pkt_cnt > 1) // Line#636 return 1; netif_stop_queue(dev); /* Disable all interrupt */ iow(db, 0xff, 0x80); /* Move data to DM9000 TX RAM */ data_ptr = (char *)skb->data; /* ... */ /* TX control: First packet immediately send, second packet queue */ if (db->tx_pkt_cnt == 0) // Line#666 { /* First Packet */ db->tx_pkt_cnt++; /* Set TX length to DM9000 */ iow(db, 0xfc, skb->len & 0xff); iow(db, 0xfd, (skb->len >> 8) & 0xff); /* Issue TX polling command */ iow(db, 0x2, 0x1); /* Cleared after TX complete */ /* saved the time stamp */ dev->trans_start = jiffies; } else { /* Second packet */ db->tx_pkt_cnt++; db->queue_pkt_len = skb->len; } /* ... */ </pre>

<pre> dmfe_init_dm9000() </pre>	<pre> iow(db, DM9KS_GPR, 1); /* Power-Down PHY */ // Line#432 udelay(500); iow(db, DM9KS_GPR, 0); /* GPR (reg_1Fh)bit GPIO0=0 pre-activate PHY */ udelay(20); /* wait 2ms for PHY power-on ready */ /* do a software reset and wait 20us */ iow(db, DM9KS_NCR, 3); udelay(20); /* wait 20us at least for software reset ok */ iow(db, DM9KS_NCR, 3); /* NCR (reg_00h) bit[0] RST=1 & Loopback=1, reset on */ udelay(20); /* wait 20us at least for software reset ok */ /* ... */ #if defined(CHECKSUM) // Line#459 iow(db, DM9KS_TCCR, 0x07); /* TX UDP/TCP/IP checksum enable */ iow(db, DM9KS_RCSR, 0x02); /*Receive checksum enable */ #endif #if defined(ETRANS) iow(db, DM9KS_ETXCSR, 0x83); #endif /* ... */ /* Activate DM9000A/DM9010 */ iow(db, DM9KS_RXCR, DM9KS_REG05 1); /* RX enable */ // Line#471 iow(db, DM9KS_IMR, DM9KS_REGFF); // Enable TX/RX interrupt mask /* ... */ </pre>	<pre> iow(db, 0x1F, 0); /* GPR (reg_1Fh)bit GPIO0=0 pre-activate PHY */ // Line#512 udelay(20); /* wait 2ms for PHY power-on ready */ /* do a software reset and wait 20us */ iow(db, DM9000_NCR, 3); udelay(20); /* wait 20us at least for software reset ok */ iow(db, 0, 3); /* NCR (reg_00h) bit[0] RST=1 & Loopback=1, reset on. Added by SPenser */ udelay(20); /* wait 20us at least for software reset ok */ /* set GPIO0=1 then GPIO0=0 to turn off and on the internal PHY */ iow(db, 0x1F, 1); /* GPR (reg_1Fh) bit[0] GPIO0=1 turn-off PHY */ iow(db, 0x1F, 0); /* GPR (reg_1Fh) bit[0] GPIO0=0 activate PHY */ udelay(4000); /* wait 4ms linking PHY (AUTO sense) if RX/TX */ /* ... */ /* Activate DM9000 */ if (dev->flags == IFF_PROMISC + IF_IFACE_T1) // Line#558 { iow(db, 0x05, db->reg5 3); printf("Into promiscuous mode\n"); } else iow(db, 0x05, db->reg5 1); /* RX enable */ iow(db, 0xff, DM9000_REGFF); /* Enable TX/RX interrupt mask */ /* ... */ </pre>
---------------------------------	--	--

<pre>dmfe_tx_done()</pre>	<pre>int nsr; // Line#594 DMFE_DEBUG(0, "dmfe_tx_done()", 0); nsr = ior(db, DM9KS_NSR); if(nsr & 0x04) db->tx_pkt_cnt--; if(nsr & 0x08) db->tx_pkt_cnt--; if (db->tx_pkt_cnt < 0) { printk("[dmfe_tx_done] tx_pkt_cnt ERROR!!\n"); db->tx_pkt_cnt =0; } if (db->Speed == 10) {if(db->tx_pkt_cnt < 1) netif_wake_queue(dev);} else {if(db->tx_pkt_cnt < 2) netif_wake_queue(dev);} return;</pre>	<pre>int tx_status = ior(db, 0x01); /* Got TX status */ // Line#746 DMFE_DEBUG(0, "dmfe_tx_done()", 0); if (tx_status & 0xc) { /* One packet sent complete */ db->tx_pkt_cnt--; dev->trans_start = 0; db->stats.tx_packets++; /* Queue packet check & send */ if (db->tx_pkt_cnt > 0) { /* Set TX length to DM9000 */ iow(db, 0xfc, db->queue_pkt_len & 0xff); iow(db, 0xfd, (db->queue_pkt_len >> 8) & 0xff); /* Issue TX polling command */ iow(db, 0x2, 0x1); /* Cleared after TX complete */ dev->trans_start = jiffies; /* saved the time stamp */ } netif_wake_queue(dev); }</pre>
---------------------------	---	--