

[PATCH 479] HP300 8250 serial for DCA and APCI ports

Source: <http://linux.derkeiler.com/Mailing-Lists/Kernel/2004-10/10225.html>

From: Geert Uytterhoeven (geert_at_linux-m68k.org)

Date: 10/31/04

Date: Sun, 31 Oct 2004 11:03:36 +0100

To: Linus Torvalds <torvalds@osdl.org>, Andrew Morton <akpm@osdl.org>, rmk+serial@arm.linux.org.uk

Created HP300 8250 driver which supports DCA and APCI ports. Support for APCI is still experimental and unfinished (no support for interrupts). Console support should be okay though (from Kars de Jong)

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/serial/8250_hp300.c 1970-01-01 01:00:00.000000000 +0100

+++ linux-m68k-2.6.10-rc1/drivers/serial/8250_hp300.c 2004-07-11 18:05:33.000000000 +0200

@@ -0,0 +1,330 @@

+/*

+ * Driver for the 98626/98644/internal serial interface on hp300/hp400

+ * (based on the National Semiconductor INS8250/NS16550AF/WD16C552 UARTs)

+ *

+ * Ported from 2.2 and modified to use the normal 8250 driver

+ * by Kars de Jong <jongk@linux-m68k.org>, May 2004.

+ */

+#include <linux/module.h>

+#include <linux/init.h>

+#include <linux/string.h>

+#include <linux/kernel.h>

+#include <linux/tty.h>

+#include <linux/serial.h>

+#include <linux/serialP.h>

+#include <linux/serial_core.h>

+#include <linux/delay.h>

+#include <linux/dio.h>

+#include <linux/console.h>

+#include <asm/io.h>

+

+#if !defined(CONFIG_HPDC) && !defined(CONFIG_HPAPCI)

+#warning CONFIG_8250 defined but neither CONFIG_HPDC nor CONFIG_HPAPCI defined, are you sure?

+#endif

+

Linux-Kernel: [PATCH 479] HP300 8250 serial for DCA and APCI ports

```
+ if (hp300_uart_scode < 0 || hp300_uart_scode > 256)
+ return 0;
+
+ scode = hp300_uart_scode;
+
+ /* Memory mapped I/O */
+ port.iotype = UPIO_MEM;
+ port.flags = UPF_SKIP_TEST | UPF_SHARE_IRQ | UPF_BOOT_AUTOCONF;
+ port.type = PORT_UNKNOWN;
+
+ /* Check for APCI console */
+ if (scode == 256)
+ {
+ #ifdef CONFIG_HPAPCI
+ printk(KERN_INFO "Serial console is HP APCI 1\n");
+
+ port.uartclk = HPAPCI_BAUD_BASE * 16;
+ port.mapbase = (FRODO_BASE + FRODO_APCI_OFFSET(1));
+ port.membase = (char *) (port.mapbase + DIO_VIRADDRBASE);
+ port.regshift = 2;
+ add_preferred_console("ttyS", port.line, "9600n8");
+ #else
+ printk(KERN_WARNING "Serial console is APCI but support is disabled (CONFIG_HPAPCI)!\n");
+ return 0;
+ #endif
+ }
+ else
+ {
+ #ifdef CONFIG_HPDCA
+ unsigned long pa = dio_scodetophysaddr(scode);
+ if (!pa) {
+ return 0;
+ }
+
+ printk(KERN_INFO "Serial console is HP DCA at select code %d\n", scode);
+
+ port.uartclk = HPDCA_BAUD_BASE * 16;
+ port.mapbase = (pa + UART_OFFSET);
+ port.membase = (char *) (port.mapbase + DIO_VIRADDRBASE);
+ port.regshift = 1;
+ port.irq = DIO_IPL(pa + DIO_VIRADDRBASE);
+
+ /* Enable board-interrupts */
+ out_8(pa + DIO_VIRADDRBASE + DCA_IC, DCA_IC_IE);
+
+ if (DIO_ID(pa + DIO_VIRADDRBASE) & 0x80) {
+ add_preferred_console("ttyS", port.line, "9600n8");
+ }
+ #else
+ printk(KERN_WARNING "Serial console is DCA but support is disabled (CONFIG_HPDCA)!\n");
+ return 0;
+ #endif
+ }
```

```

+ #endif
+ }
+
+ if (early_serial_setup(&port) < 0) {
+ printk(KERN_WARNING "hp300_setup_serial_console(): early_serial_setup() failed.\n");
+ }
+
+ return 0;
+ }
+ #endif /* CONFIG_SERIAL_8250_CONSOLE */
+
+ static int __init hp300_8250_init(void)
+ {
+ static int called = 0;
+ #ifdef CONFIG_HPDCA
+ int scode;
+ #endif
+ int line, num_ports;
+ unsigned long base;
+ struct serial_struct serial_req;
+ struct hp300_port *port;
+
+ if (called)
+ return -ENODEV;
+ called = 1;
+ num_ports = 0;
+
+ if (!MACH_IS_HP300) {
+ return -ENODEV;
+ }
+
+ #ifdef CONFIG_HPDCA
+ while (1) {
+ /* We detect boards by looking for DIO boards which match a
+ * given subset of IDs. dio_find() returns the board's scancode.
+ * The scancode to physaddr mapping is a property of the hardware,
+ * as is the scancode to IPL (interrupt priority) mapping.
+ */
+ scode = dio_find(DIO_ID_DCA0);
+ if (scode < 0)
+ scode = dio_find(DIO_ID_DCA0REM);
+ if (scode < 0)
+ scode = dio_find(DIO_ID_DCA1);
+ if (scode < 0)
+ scode = dio_find(DIO_ID_DCA1REM);
+ if (scode < 0)
+ break; /* no, none at all */
+
+ #ifdef CONFIG_SERIAL_8250_CONSOLE
+ if (hp300_uart_scode == scode) {
+ /* Already got it */

```

```

+ dio_config_board(scode);
+ continue;
+ }
+ #endif
+
+ /* Create new serial device */
+ port = kmalloc(sizeof(struct hp300_port), GFP_KERNEL);
+ if (!port)
+ return -ENOMEM;
+
+ memset(&serial_req, 0, sizeof(struct serial_struct));
+
+ base = dio_scodetophysaddr(scode);
+
+ /* If we want to tell the DIO code that this board is configured,
+ * we should do that here.
+ */
+ dio_config_board(scode);
+
+ /* Memory mapped I/O */
+ serial_req.io_type = SERIAL_IO_MEM;
+ serial_req.flags = UPF_SKIP_TEST | UPF_SHARE_IRQ | UPF_BOOT_AUTOCONF;
+ serial_req.irq = dio_scodetoipl(scode);
+ serial_req.baud_base = HPDCA_BAUD_BASE;
+ serial_req.iomap_base = (base + UART_OFFSET);
+ serial_req.iomem_base = (char*)(serial_req.iomap_base + DIO_VIRADDRBASE);
+ serial_req.iomem_reg_shift = 1;
+
+ #ifdef CONFIG_SERIAL_8250_CONSOLE
+ if (hp300_uart_scode != scode) {
+ #endif
+ /* Reset the DCA */
+ out_8(base + DIO_VIRADDRBASE + DCA_ID, 0xff);
+ udelay(100);
+ #ifdef CONFIG_SERIAL_8250_CONSOLE
+ }
+ #endif
+
+ line = register_serial(&serial_req);
+
+ if (line < 0) {
+ printk(KERN_NOTICE "8250_hp300: register_serial() DCA scode %d"
+ " irq %d failed\n", scode, serial_req.irq);
+ kfree(port);
+ continue;
+ }
+
+ /* Enable board-interrupts */
+ out_8(base + DIO_VIRADDRBASE + DCA_IC, DCA_IC_IE);
+
+ port->dio_base = base + DIO_VIRADDRBASE;

```

```

+ port->scode = scode;
+ port->line = line;
+ port->next = hp300_ports;
+ hp300_ports = port;
+
+ num_ports++;
+ }
+ #endif
+
+ #ifdef CONFIG_HPAPCI
+ if (hp300_model >= HP_400)
+ {
+ int i;
+
+ /* These models have the Frodo chip.
+ * Port 0 is reserved for the Apollo Domain keyboard.
+ * Port 1 is either the console or the DCA.
+ */
+ for (i = 1; i < 4; i++) {
+ /* Port 1 is the console on a 425e, on other machines it's mapped to
+ * DCA.
+ */
+ #ifdef CONFIG_SERIAL_8250_CONSOLE
+ if (i == 1) {
+ continue;
+ }
+ #endif
+
+ /* Create new serial device */
+ port = kmalloc(sizeof(struct hp300_port), GFP_KERNEL);
+ if (!port)
+ return -ENOMEM;
+
+ memset(&serial_req, 0, sizeof(struct serial_struct));
+
+ base = (FRODO_BASE + FRODO_APCI_OFFSET(i));
+
+ /* Memory mapped I/O */
+ serial_req.io_type = SERIAL_IO_MEM;
+ serial_req.flags = UPF_SKIP_TEST | UPF_SHARE_IRQ | UPF_BOOT_AUTOCONF;
+ /* XXX - no interrupt support yet */
+ serial_req.irq = 0;
+ serial_req.baud_base = HPAPCI_BAUD_BASE;
+ serial_req.iomap_base = base;
+ serial_req.iomem_base = (char*)(serial_req.iomap_base + DIO_VIRADDRBASE);
+ serial_req.iomem_reg_shift = 2;
+
+ line = register_serial(&serial_req);
+
+ if (line < 0) {
+ printk(KERN_NOTICE "8250_hp300: register_serial() APCI %d"

```

Linux-Kernel: [PATCH 479] HP300 8250 serial for DCA and APCI ports

```
+ " irq %d failed\n", i, serial_req.irq);
+ kfree(port);
+ continue;
+ }
+
+ port->dio_base = 0;
+ port->line = line;
+ port->next = hp300_ports;
+ hp300_ports = port;
+
+ num_ports++;
+ }
+ }
+#endif
+
+ /* Any boards found? */
+ if (!num_ports)
+ return -ENODEV;
+
+ return 0;
+}
+
+static void __exit hp300_8250_exit(void)
+{
+ struct hp300_port *port, *to_free;
+
+ for (port = hp300_ports; port; ) {
+ unregister_serial(port->line);
+
+#ifdef CONFIG_HPDCA
+ if (port->dio_base) {
+ /* Disable board-interrupts */
+ out_8(port->dio_base + DCA_IC, 0);
+
+ dio_unconfig_board(port->scode);
+ }
+#endif
+
+ to_free = port;
+ port = port->next;
+ kfree(to_free);
+ }
+
+ hp300_ports = NULL;
+}
+
+module_init(hp300_8250_init);
+module_exit(hp300_8250_exit);
+MODULE_DESCRIPTION("HP DCA/APCI serial driver");
+MODULE_AUTHOR("Kars de Jong <jongk@linux-m68k.org>");
+MODULE_LICENSE("GPL");
```

Linux-Kernel: [PATCH 479] HP300 8250 serial for DCA and APCI ports

```
--- linux-2.6.10-rc1/drivers/serial/Makefile 2004-05-11 11:09:06.000000000 +0200
+++ linux-m68k-2.6.10-rc1/drivers/serial/Makefile 2004-07-14 13:19:16.000000000 +0200
@@ -9,6 +9,7 @@
serial-8250-$(CONFIG_GSC) += 8250_gsc.o
serial-8250-$(CONFIG_PCI) += 8250_pci.o
serial-8250-$(CONFIG_PNP) += 8250_pnp.o
+serial-8250-$(CONFIG_HP300) += 8250_hp300.o

obj-$(CONFIG_SERIAL_CORE) += serial_core.o
obj-$(CONFIG_SERIAL_21285) += 21285.o
```

Gr{oetje,eeting}s,

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/
```