

Re: [patch 04/16] I/O driver for 8250-compatible UARTs

Source: <http://linux.derkeiler.com/Mailing-Lists/Kernel/2005-08/8017.html>

From: Bjorn Helgaas (bjorn.helgaas_at_hp.com)

Date: 08/31/05

To: Tom Rini <trini@kernel.crashing.org>

Date: Wed, 31 Aug 2005 13:38:52 -0600

On Monday 29 August 2005 10:09 am, Tom Rini wrote:

> *linux-2.6.13-trini/drivers/serial/kgdb_8250.c* / 594 ++++++

The existing stuff in drivers/serial is named "8250_*"; is there a reason you're using "kgdb_8250" rather than "8250_kgdb"?

> + * *serial8250_unregister_by_port* – remove a 16x50 serial port
> + * *at runtime*.
> + * *@port: A &struct uart_port that describes the port to remove*.
> + *
> + * *Remove one serial port. This may not be called from interrupt*
> + * *context. We hand the port back to the our control*.

Hand the port back to whose control?

> +*MODULE_PARM_DESC(kgdb8250, " kgdb8250=<port number>,<baud rate>\n");*

Document IRQ/MMIO/IOport stuff here too (whatever it turns out to be, see below).

It seems wrong to me to have kgdb8250 use "ttyS" names for devices. In general you don't know the ttyS name for a device until after the 8250 driver claims it. Sure, it's a convenience to use "ttyS0" instead of "io,0x3f8", but it also makes it hard to run kgdb on a PCI or MMIO UART, because you don't know what its name will be.

If you gave up the convenience and just always required an I/O port or MMIO address, you could nuke all the *old_rs_table[]* and *kgdb8250_ports[]* stuff, not to mention a bunch of config options.

> +*static int kgdb8250_local_init(void)*
> +{
> + *if (old_rs_table_copied == 0)*
> + *kgdb8250_copy_rs_table();*

This would be easier to maintain if the "if (old_rs_table_copied)" test were in the callee, not in every caller.

```
> + switch (CURRENTPORT.iotype) {
> + case UPIO_MEM:
> + if (CURRENTPORT.mapbase)
> + kgdb8250_needs_request_mem_region = 1;
> + if (CURRENTPORT.flags & UPF_IOREMAP) {
> + CURRENTPORT.membase = ioport_map(CURRENTPORT.mapbase,
> + 8 << KGDB8250_REG_SHIFT);
```

Shouldn't this be ioremap instead of ioport_map?

```
> +static int kgdb_init_io(void)
> +{
> +#ifdef CONFIG_KGDB_8250_MODULE
> + if (strlen(config)) {
> + if (kgdb8250_opt(config))
> + return -EINVAL;
> + } else {
> + printk(KERN_ERR "kgdb8250: argument error, usage: "
> + "kgdb8250=<port number>,<baud rate>");
> +#ifdef CONFIG_IA64
> + printk(",<irq>,<iomem base>");
> +#endif
```

This isn't ia64-specific.

```
> +static void __init kgdb8250_hookup_irq(void)
> +{
> +#if defined(CONFIG_SERIAL_8250) || defined (CONFIG_SERIAL_8250_MODULE)
> + /* Take the port away from the main driver. */
> + serial8250_unregister_by_port(&CURRENTPORT);
> +
> + /* Now reinit the port as the above has disabled things. */
> + kgdb8250_init();
> +#endif
> + /* We may need to call request_mem_region() first. */
> + if (kgdb8250_needs_request_mem_region)
> + request_mem_region(CURRENTPORT.mapbase,
> + 8 << KGDB8250_REG_SHIFT, "kgdb");
```

The unregister/init/request_mem_region stuff doesn't sound very much like "hookup_irq".

```
> + * Syntax for this cmdline option is "kgdb8250=ttyno,baudrate"
> + * with ",irq,iomembase" tacked on the end on IA64.
```

This syntax doesn't really make sense on ia64, because there are no fixed "ttyno/iomembase" mappings. It would be unambiguous to specify either ttyno OR iomembase, but there's no good way to use

both.

And there should be syntax to specify either MMIO or I/O port space devices. Intel ia64 boxes typically have I/O port UARTs (0x3f8, etc), and HP boxes typically have MMIO devices.

```
> +#ifdef CONFIG_IA64
> + if (*str == ',') {
> + str++;
> + KGDB8250_IRQ = simple_strtoul(str, &str, 10);
> + if (*str == ',') {
> + str++;
> + CURRENTPORT.iotype = SERIAL_IO_MEM;
> + CURRENTPORT.membase =
> + (unsigned char *)simple_strtoul(str, &str, 0);
> + }
> + }
> +#endif
```

Not ia64-specific.

```
> +config KGDB_SIMPLE_SERIAL
> + bool "Simple selection of KGDB serial port"
> + depends on KGDB_8250
> + default y
> + help
> + If you say Y here, you will only have to pick the baud rate
> + and serial port (ttyS) that you wish to use for KGDB. If you
> + say N, you will have provide the I/O port and IRQ number. Note
> + that if your serial ports are iomapped, such as on ia64, then
> + you must say Y here. If in doubt, say Y.
```

How about: "... you will have to provide the address (I/O port or MMIO address) and IRQ ..."

I don't understand the "iomapped" bit --- does that mean MMIO? And why would it make any difference whether they're in I/O port or MMIO space?

I expect that if you use "ttyS" naming to select a port, that doesn't work early in boot on ia64, because ia64 doesn't have compiled-in knowledge of where ttyS devices live. Is that related to what this is trying to say?

```
> +config KGDB_PORT
> + hex "hex I/O port address of the debug serial port"
> + depends on !KGDB_SIMPLE_SERIAL && KGDB_8250 && !IA64
> + default 3f8
> + help
> + This is the unmapped (and on platforms with 1:1 mapping
> + this is typically, but not always the same as the mapped)
> + address of the serial port. The stanards on your architecture
```

Linux-Kernel: Re: [patch 04/16] I/O driver for 8250-compatible UARTs

> + *may be found in include/asm-\$(ARCH)/serial.h.*

Not ia64-specific. The description sounds like it applies to MMIO, not to I/O port space. And s/stanards/standards/.

> + *config KGDB_IRQ*
> + *int "IRQ of the debug serial port"*
> + *depends on !KGDB_SIMPLE_SERIAL && KGDB_8250 && !IA64*
> + *default 4*
> + *help*
> + *This is the IRQ for the debug port. This must be known so that*
> + *KGDB can interrupt the running system (either for a new*
> + *connection or when in gdb and control-C is issued).*

Not ia64-specific.

-

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/>