

Re: [RFC][PATCH] SPI subsystem

Source: <http://linux.derkeiler.com/Mailing-Lists/Kernel/2005-09/8737.html>

From: David Brownell (*david-b_at_pacbell.net*)

Date: 09/30/05

Date: Thu, 29 Sep 2005 18:02:08 -0700

To: linux-kernel@vger.kernel.org, basicmark@yahoo.com

Sorry for the delay getting back to these comments; I wanted to give them proper attention, which kept not arriving.

> *Date: Sun, 18 Sep 2005 15:45:20 +0100 (BST)*

> *From: Mark Underwood <basicmark@yahoo.com>*

First comments for <linux/spi.h>:

```
> > +struct spi_device { /* this proxies the device through a master */
> > + struct device dev;
> > + struct spi_master *master;
> > + u32 max_speed_hz;
> > + u8 chip_select;
> > + u8 mode;
> > + #define SPI_CPHA 0x01 /* clock phase */
> > + #define SPI_CPOL 0x02 /* clock polarity */
> > + #define SPI_MODE_0 (0|0)
> > + #define SPI_MODE_1 (0|SPI_CPHA)
> > + #define SPI_MODE_2 (SPI_CPOL|0)
> > + #define SPI_MODE_3 (SPI_CPOL|SPI_CPHA)
>
> Would be more flexible to have this in the message or even
> the spi_transfer structure. Although I
> don't know who would need this flexibility.
```

In this case, I don't see a benefit. The chips support only one signaling method at a time. It can be changed between requests, by calling spi_setup(...), but even that will be rare. I don't think there's any point to encouraging finer grained changes.

```
> > +struct spi_master {
> > + ...
> > +};
>
> I notice that there is no bus lock. Are you expecting the adapter
> driver to handle the fact that its transfer routine could be called
> before a previous call returns?
```

Linux-Kernel: Re: [RFC][PATCH] SPI subsystem

Yes. The transfer routine is purely async, and its responsibility is to append that `spi_message` to the current queue. (Assuming the driver isn't a simple pure-PIO driver without a queue...)

That's a simple matter of a `spin_lock_irqsave/list_add_tail/unlock`.

```
> > +struct spi_transfer {  
> > + /* it's ok if tx_buf =
```