

Re: [spi-devel-general] Accelerometer, Gyros and ADC's etc within the kernel.

Re: [spi-devel-general] Accelerometer, Gyros and ADC's etc within the kernel.

Source: <http://linux.derkeiler.com/Mailing-Lists/Kernel/2008-05/msg13209.html>

- *From:* Jonathan Cameron <Jonathan.Cameron@xxxxxxxx>
 - *Date:* Thu, 29 May 2008 12:57:19 +0100
-

Ok, say the HW SPI is pushed down to some lower level driver. These things send streaming data, and sometimes signals (free-fall detect) out of band WRT the SPI comms.

Indeed. This would have to be handled using interrupts based on the generic gpio interface. (see the LIS3L02DQ driver on the SPI-devel mailing list for an example of how this might work.)

I would like to see how a higher level API could encapsulate the generalization of a class of such devices.

Also, I'm not sure how one could generalize SPI communications at a high level given its bi-directional nature.

I guess I need to see some header files showing what you are thinking about.

Sure. I'll get something put together and sent out asap. The general form of this is going to look like a combination of the hwmon interface via sysfs and something similar to the input subsystems approach of using events sent over character devices. The interesting bit is going to be working out how to provide a fairly generic interface to ring buffers, whether these reside in the kernel or on the device itself.

Keep in mind that one abstraction of these devices is they are simply streaming A2D 16 bit data values over SPI (or I2C). The size of the data set is a function of the sensor axes.

This is of course true, but there still seem to be some advantages in providing some common interfaces for these devices. Almost all digital interfaced accelerometers share a number of common elements

Re: [spi-devel-general] Accelerometer, Gyros and ADC's etc within the kernel.

Re: [spi-devel-general] Accelerometer, Gyros and ADC's etc within the kernel.

not typically found in ADC's. These include bias and gain as well as commonly found alarm signals.

Actually, on this point, it may well make sense to provide an accelerometer driver that is designed to run on top of any generic ADC driver, thus providing (based on a suitable board specific configuration) userspace access to analog accelerometers connected in this fashion. This is definitely one for some way down the line!

Also, If you are playing with accelerometer data, you likely need some real time support or at least a reliable time stamping of the data to do anything interesting.

Definitely. Obviously the accuracy of this time stamp is going to be limited by the variable nature of when an interrupt handler picks it up, but knowledge of the device specs can allow filtering of this timing data. This is certainly something we would want to consider at a later date although it may well be more sensible to leave this to userspace applications.

Another problem area is around SPI itself. There are variations of device implementations around chip select polarity, clock biasing (rising,falling, or midpoint) sampling from one SPI part to the next.

Indeed. This is well handled by the SPI subsystem. I've spent far too much time with a scope recently fiddling with these parameters for badly documented chips!

VTI SCA3000 E05 3D accelerometer
equipped with substantial ring buffer.
SPI

device. Can operate either in buffered or
direct mode. In buffered
mode, interrupts

can be used to indicate when the ring buffer
is 3/4 full triggering a
download to

a larger ring buffer in the kernel if necessary.

<http://www.vti.fi/en/products-solutions/products/accelerometers/sca3000-accelerom>

Re: [spi-devel-general] Accelerometer, Gyros and ADC's etc within the kernel.

ring buffered data can make RT applications harder...

Agreed. With this sort of device two modes would be needed (and are supported by the hardware). The first gives direct data access and the second uses the ring buffer. Which you use would obviously be dependent on the application. Some applications want the most recent data on all occasions, whilst some will be more interested in ensuring that all data is captured.

Would be nice if practical to allow the framework to include RS232 devices such as those from www.xsens.com, www.isense.com and others.

I'm not sure what you are asking for, you started off with SPI driver for interfacing a handful of accelerometer devices. Now your talking about the serial port.

The question here is concerned with standardising mainly the userspace interfaces (and hardware interfaces only when appropriate). Some of these devices (the ST accel above for example) support multiple hardware interfaces specs (I2C and SPI for that one). So in a similar way to it making sense to group all TV adapters irrespective of how they are interfaced to the computer, in order to ensure a consistent interface to user space, it makes sense to share these interfaces amongst drivers talking to these devices irrespective of what their hardware interface is.

I guess I need to see some header files to see where you are going with this idea. To be useful you'll also need to handle rotation sensors and communicate the temperature bias's up the stack too.

Definitely, although to get a reasonable number of devices integrated quickly they may not all support access to everything we would like initially.

good luck keep me posted on where your going with this, it is interesting to me.

Will do,

--

Jonathan Cameron

--

Re: [spi-devel-general] Accelerometer, Gyros and ADC's etc within the kernel.

Re: [spi-devel-general] Accelerometer, Gyros and ADC's etc within the kernel.

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