

# Serial Peripheral Interface (SPI)

SPI is the "Serial Peripheral Interface", widely used with embedded systems because it is a simple and efficient interface: basically a multiplexed shift register. Its three signal wires hold a clock (SCK, often in the range of 1-20 MHz), a "Master Out, Slave In" (MOSI) data line, and a "Master In, Slave Out" (MISO) data line. SPI is a full duplex protocol; for each bit shifted out the MOSI line (one per clock) another is shifted in on the MISO line. Those bits are assembled into words of various sizes on the way to and from system memory. An additional chipselect line is usually active-low (nCS); four signals are normally used for each peripheral, plus sometimes an interrupt.

The SPI bus facilities listed here provide a generalized interface to declare SPI busses and devices, manage them according to the standard Linux driver model, and perform input/output operations. At this time, only "master" side interfaces are supported, where Linux talks to SPI peripherals and does not implement such a peripheral itself. (Interfaces to support implementing SPI slaves would necessarily look different.)

The programming interface is structured around two kinds of driver, and two kinds of device. A "Controller Driver" abstracts the controller hardware, which may be as simple as a set of GPIO pins or as complex as a pair of FIFOs connected to dual DMA engines on the other side of the SPI shift register (maximizing throughput). Such drivers bridge between whatever bus they sit on (often the platform bus) and SPI, and expose the SPI side of their device as a `:ctype:'struct spi_master <spi_master>'`. SPI devices are children of that master, represented as a `:ctype:'struct spi_device <spi_device>'` and manufactured from `:ctype:'struct spi_board_info <spi_board_info>'` descriptors which are usually provided by board-specific initialization code. A `:ctype:'struct spi_driver <spi_driver>'` is called a "Protocol Driver", and is bound to a `spi_device` using normal driver model calls.

**System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\linux-master\Documentation\driver-api\linux-master) (Documentation) (driver-api) spi.rst, line 22);  
[backlink](#)

Unknown interpreted text role "c:type".

**System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\linux-master\Documentation\driver-api\linux-master) (Documentation) (driver-api) spi.rst, line 22);  
[backlink](#)

Unknown interpreted text role "c:type".

**System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\linux-master\Documentation\driver-api\linux-master) (Documentation) (driver-api) spi.rst, line 22);  
[backlink](#)

Unknown interpreted text role "c:type".

**System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\linux-master\Documentation\driver-api\linux-master) (Documentation) (driver-api) spi.rst, line 22);  
[backlink](#)

Unknown interpreted text role "c:type".

The I/O model is a set of queued messages. Protocol drivers submit one or more `:ctype:'struct spi_message <spi_message>'` objects, which are processed and completed asynchronously. (There are synchronous wrappers, however.) Messages are built from one or more `:ctype:'struct spi_transfer <spi_transfer>'` objects, each of which wraps a full duplex SPI transfer. A variety of protocol tweaking options are needed, because different chips adopt very different policies for how they use the bits transferred with SPI.

**System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\linux-master\Documentation\driver-api\linux-master) (Documentation) (driver-api) spi.rst, line 37);  
[backlink](#)

Unknown interpreted text role "c:type".

**System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\linux-master\Documentation\driver-api\linux-master) (Documentation) (driver-api) spi.rst, line 37);  
[backlink](#)

Unknown interpreted text role "c:type".

**System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\linux-

**master\Documentation\driver-api\ (linux-master) (Documentation) (driver-api) spi.rst, line 46)**

Unknown directive type "kernel-doc".

```
.. kernel-doc:: include/linux/spi/spi.h
   :internal:
```

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\linux-master\Documentation\driver-api\ (linux-master) (Documentation) (driver-api) spi.rst, line 49)**

Unknown directive type "kernel-doc".

```
.. kernel-doc:: drivers/spi/spi.c
   :functions: spi_register_board_info
```

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\linux-master\Documentation\driver-api\ (linux-master) (Documentation) (driver-api) spi.rst, line 52)**

Unknown directive type "kernel-doc".

```
.. kernel-doc:: drivers/spi/spi.c
   :export:
```