NAME

spi - communicate on SPI bus with slave devices

SYNOPSIS

spi [**-A**] [**-b**] [**-L**] [**-v**] [**-C** *command-bytes*] [**-c** *count*] [**-d r**|**w**|**rw**] [**-f** *device*] [**-m** *mode*] [**-s** *max-speed*]
spi [**-i**] [**-v**] [**-f** *device*]
spi [**-h**]

DESCRIPTION

The **spi** utility can be used to perform raw data transfers (read, write, or simultaneous read/write) with devices on the SPI bus, via the spigen(4) device.

Each spigen(4) device is associated with a specific "chip select" (cs) pin on the spibus, and therefore needs to be specified. If no device name is specified on the command line, **spi** assumes "spigen0.0".

For more information on the spigen device, see spigen(4).

The options are as follows:

-A Specifies ASCII mode. Both read and write data is input and output as 2-character hexadecimal values, optionally separated by white space, such as 00 01 02 etc. When combined with the **-b** flag, the data on stdin remains a sequence of ASCII hexadecimal byte values, but the output reverts to binary mode.

-b Binary (output) mode. Only has an effect when **-A** has been specified. Reverts the output back to binary (rather than ASCII), while leaving the input format as-is. Use in combination with **-A** to allow using something like "echo" to pass hexadecimal values to the SPI device, but output the received data on stdout as binary.

-C *command-bytes*
Sends one or more command bytes, skipping any bytes read-in during the transfer. The byte values should be specified as a quoted parameter, similar to the format for data on stdin for **-A**, that is, 2 character hexadecimal values, optionally separated by white space. An SPI device will typically require that a command be sent, followed by bytes of data. You can use this option to send the command without receiving any data bytes during the command sequence.

-c *count* The total number of bytes to transfer as a decimal integer. If a write or a read/write transaction is being performed, and fewer than this number of bytes are read in from stdin, the remaining bytes will be sent with a value of "0". If the length can be determined from the input file size, you can use a *count* value of "-1" to base the transfer on the input file's size.

-d r|w|rw
Transfer direction: Use **r** for read, **w** for write, and **rw** for simultaneous read and write.

-f *device* SPI device to use (default is */dev/spigen0*).

-h Print help text to stderr, explaining the command line options.

-i Displays information about the SPI device to stderr. Whenever this flag is specified, no data is read or written, and the mode and clock speed are not changed.

-L LSB bit order. The default is MSB, i.e., the highest order bit is transmitted first. Specifying **-L** caused the LSB to be transmitted and read first.

-m 0|1|2|3
SPI mode, 0 through 3. This defines the clock phase and timing with respect to reading and writing data, as per the SPI specification.

-s *speed* Specify the maximum speed, in Hz, for the SPI clock. The bus will operate at its highest available speed which does not exceed this maximum.

-v Specifies Verbose mode. Diagnostics and information are written to stderr. You can specify **-v** more than once to increase verbosity.

EXAMPLES

Here are a few examples of using the spi utility:

- Get information about the default SPI device

  spi -i

- Set the maximum clock speed to 200Khz and the mode to 3 on spigen0.1, but do not transmit nor receive any data

  spi -f spigen0.1 -s 200000 -m 3

- Send a command sequence consisting of 2 bytes, and read 2 additional bytes from the SPI device, using the current mode and speed on the default device

  spi -d r -C "00 01" -c 2

- Transmit a byte value of 5, and receive 2 bytes, displaying their values as 2-byte ASCII hexadecimal, with mode 2, and a maximum clock speed of 500khz.

  echo "05" | spi -A -d rw -m 2 -s 500000 -c 2

- Send a binary file, and output the SPI result through od(1) as hexadecimal bytes, using the current maximum clock speed and SPI mode.

  spi -d rw -c -1 <input_file.bin | od -An -t x1

- Send 2 bytes of data, receive a total of 4 bytes, and output the SPI result as binary data, piped through od(1), displaying it as two hexadecimal unsigned short integer values.

  echo "00 01" | spi -A -b -d rw -c 4 | od -t x2

- Query the manufacturer ID and size from a standard spiflash device, by sending the command byte 0x9f and displaying the 3-byte reply in ASCII hex.

  spi -f spigen0.0 -m 0 -s 1000000 -d r -c 3 -A -C 9f

SEE ALSO

spigen(4)

HISTORY

The **spi** utility appeared in FreeBSD 11.3.