

## How to use SPI driver

```
*****
1. Introduction
*****
```

SPI is a serial transfer protocol. It has four interface signals: MISO, MOSI, SCLK, and SPISS. The first two signals are data signals, SCLK is clock signal, and SPISS is chip select signal.

ADSP21535 has two SPI ports, and can generate 7 chip select signal when acting as master. Please refer to Analog Device datasheet for details.

The two SPI ports can be operated by open /dev/spi0 and/or /dev/spi1.

When SPI sends a word, it receive a word at the same time. So, the number of SPI receiving must equal to number of sending.

If SPI want to read(receive) data, it must write(send) data firstly.

```
*****
2. Act as master
*****
```

Before open driver, first make clear whether chip select signal is used or not, if used, which chip select signal is used. For example, CS 3(SPIxSEL3) is connected with SPI slave's SPISS.

- i. open /dev/spi1 (spi0 is connected with SROM)
- ii. ioctl CMD\_SPI\_OUT\_ENABLE, so MISO is in normal state.(else it is in open drain state.
- iii. 'ioctl' "CMD\_SPI\_SET\_BAUDRATE", set SCLK's baud rate. If you write a short value "0xabcd", the baud rate is SCLK/(2\*0xabcd).
- iv. ioctl' "CMD\_SPI\_SET\_CSENABLE", arg is 3.
- v. 'ioctl' "CMD\_SPI\_SET\_CSLOW", arg is 3.
- vi. 'ioctl' "CMD\_SPI\_SET\_MASTER", SPI acts as master.
- vii. 'ioctl' "CMD\_SPI\_MISO\_ENABLE", master can receive data.
- viii. 'write' a word, or data buffer to SPI driver.
- ix. 'read' a word, or data buffer from SPI driver.
- x. 'close' close SPI driver when finish operating.

```
*****
3. Act as slave
*****
```

The sequence is similar with master mode.

- i. 'open' "/dev/spi1" (spi0 is connected with SROM)
- ii. 'ioctl' "CMD\_SPI\_OUT\_ENABLE", so MISO is in normal state.(else it is in open drain state).
- vi. 'ioctl' "CMD\_SPI\_SET\_MASTER", arg 0, SPI acts as slave.
- vii. 'ioctl' "CMD\_SPI\_MISO\_ENABLE", slave can send out data.
- vii. 'write' a word, or data buffer to SPI driver.
- viii. 'read' a word, or data buffer from SPI driver.
- ix. 'close' close SPI driver when finish operating.

```
*****
```

### 3. SPI transfer format

\*\*\*\*\*  
SPI can operate in 8 bits, or 16 bits, in active high, or low,..  
User application can use ioctl to set data format before beginning  
read or write.  
Master and slave must keep in same format.

ioctl CMD\_SPI\_SET\_POLAR arg 1: active low 0: active high  
ioctl CMD\_SPI\_SET\_PHASE arg 1: toggle from beginning 0: toggle  
from middle  
ioctl CMD\_SPI\_SET\_RECVOPT arg 1: Flush old data if buffer full;  
0: Discard new data if buffer full.  
ioctl CMD\_SPI\_SET\_ORDER arg 1: LSB first 0: MSB first  
ioctl CMD\_SPI\_SET\_LENGTH16 arg 1: 16 bits 0: 8 bits.

\*\*\*\*\*  
4. Caution  
\*\*\*\*\*  
SPI will interrupt CPU when sending out a word, so we should not  
send a large block of memory each time. If dose that, CPU will  
be interrupted more than 1k per second(Interrupt times depends on  
baud rate).