**SPI Driver Basics: -**

1. **SPI Modes: -**

CPOL = 0 - > Idle state of SPI\_CLK is low

CPHA = 0 - > Data latched in falling edge

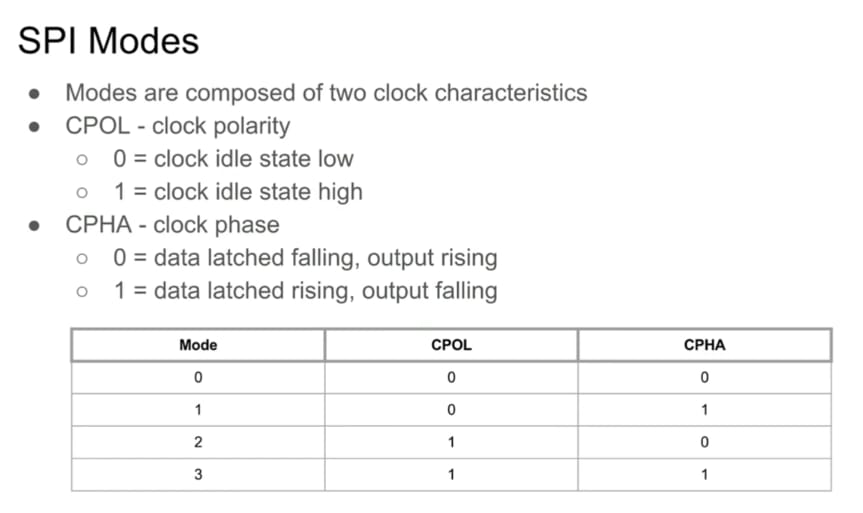


Figure - 1

1. **Types of SPI: -**

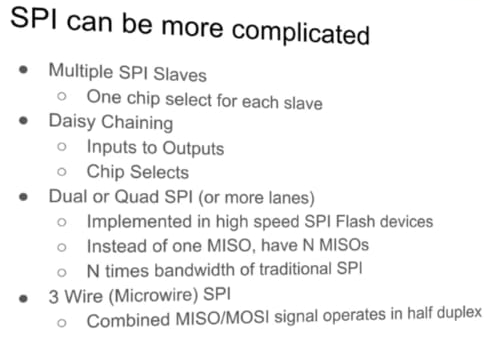


Figure – 2

1. **Linux SPI Driver: -**

There are two types of SPI driver –

1. SPI controller driver
2. SPI protocol driver

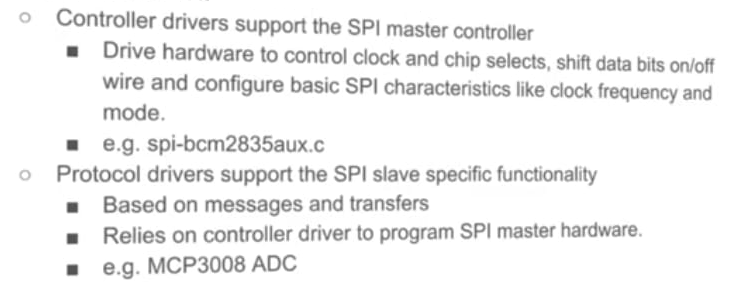


Figure – 3

1. **Linux SPI communication: -**

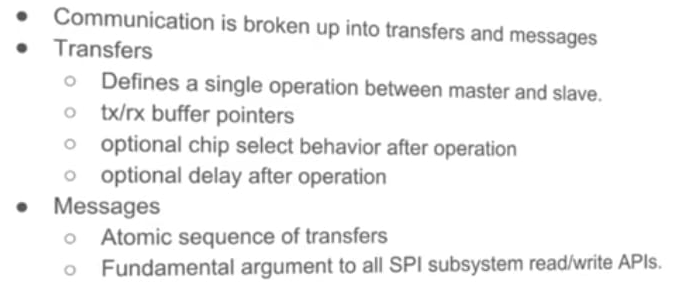


Figure – 4

struct spi\_message is basically argument to all SPI sub system read/write APIs.

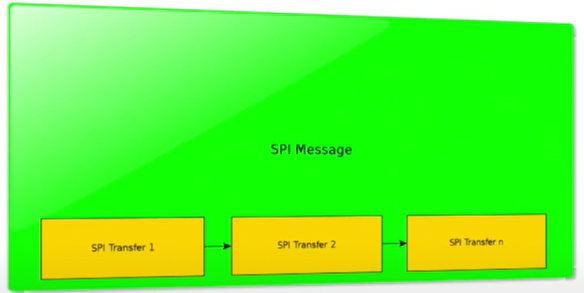


Figure – 5

1. **Adding a SPI device to a system: -**

**A picture containing text

Description automatically generated**

Figure - 6

1. **SPI device DT binding: -**

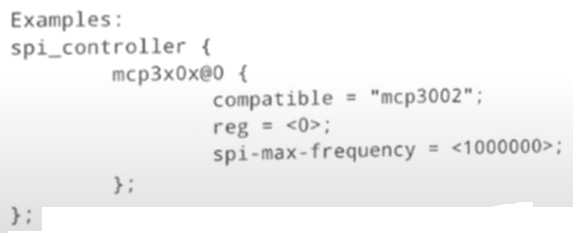
****

Figure – 7

1. **Adding SPI device using board file: -**

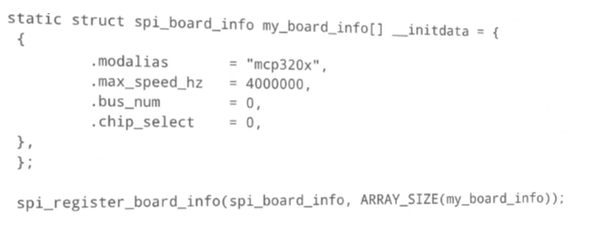
****

Figure - 8

1. **SPI protocol driver: -**

Follows standard driver model.

**Text

Description automatically generated**

Figure – 9

1. **SPI sub system kernel APIs: -**

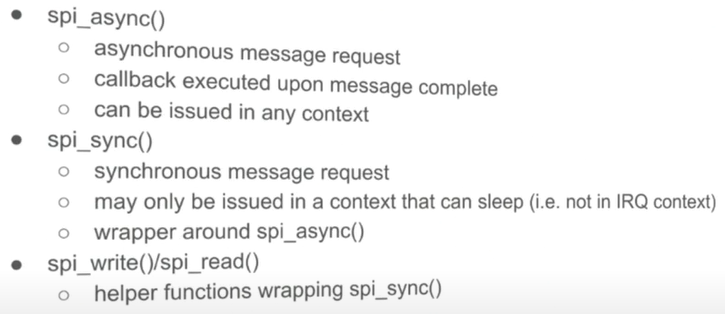
****

Figure - 10

**Graphical user interface, text, application, email

Description automatically generated**

Figure – 11

1. **SPI controller driver: -**

**Graphical user interface, text, application

Description automatically generated**

Figure – 12

transfer\_one model allows us to let the core support GPIO chip select.

1. **User space driver - spidev: -**

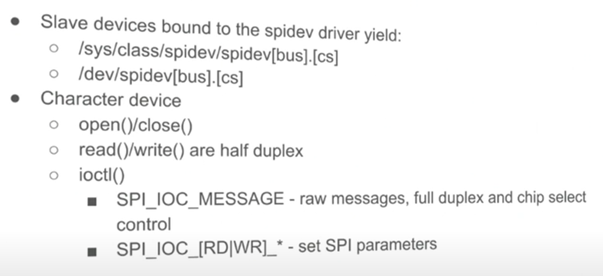
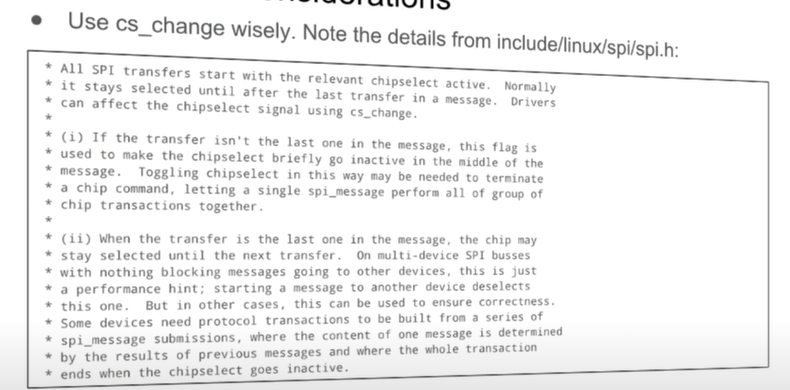


Figure – 13

SPI is inherently a full duplex bus.

1. **“spi\_transfer” cs\_change flag: -**

****

1. **Performance considerations: -**

Graphical user interface, text, application

Description automatically generated

1. **Performance tools: -**

**Graphical user interface, text

Description automatically generated**