GLOBAL EDGE
Intelligence Of Things™

SPI and V4L2

Jyothi A.

Contents

1. SPI
    - Introduction
    - Separate slave line
    - Single slave line
    - Programming in SPI
    - Advantages of SPI
    - Disadvantages of SPI

Cont...

2. V4L2

- Introduction
- Data Structures
- Input / Output
- Capturing Video

Serial Peripheral Interface (SPI)

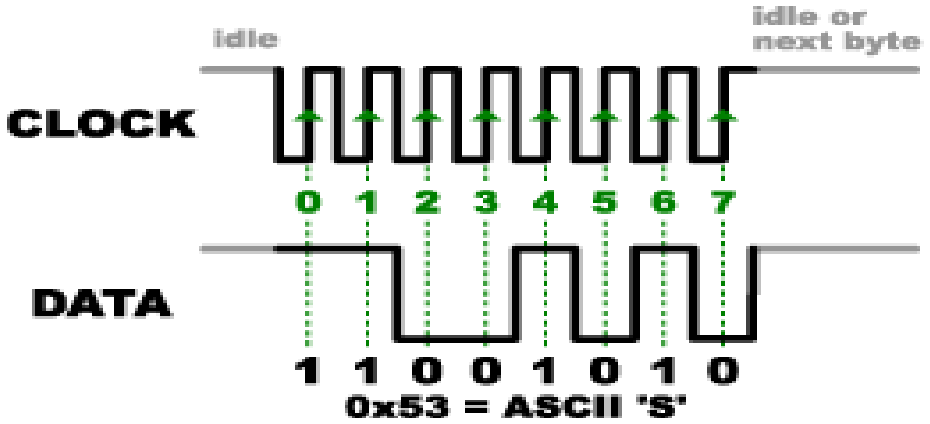
Introduction

- SPI is an interface bus commonly used to send data between microcontrollers and small peripherals such as
  - shift registers
  - sensors
  - SD cards

- SPI developed by motorola to provide full duplex synchronous serial communication.

Cont...

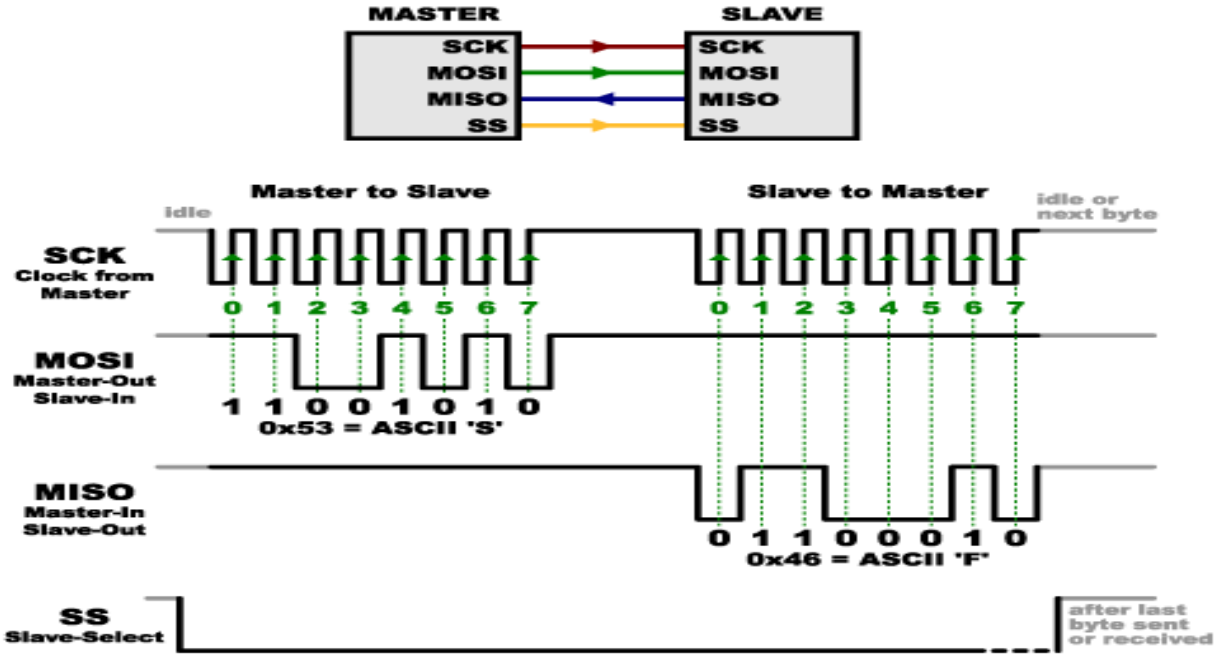
- SPI has synchronous solution to receive data.



Cont...

- SPI has four pins:
  - SCK / CLK : generates clock signal.
  - MOSI : data sent from master to slave, Master Out Slave In
  - MISO : data sent from slave to master, Master In Slave Out
  - SS : used to select the required slave

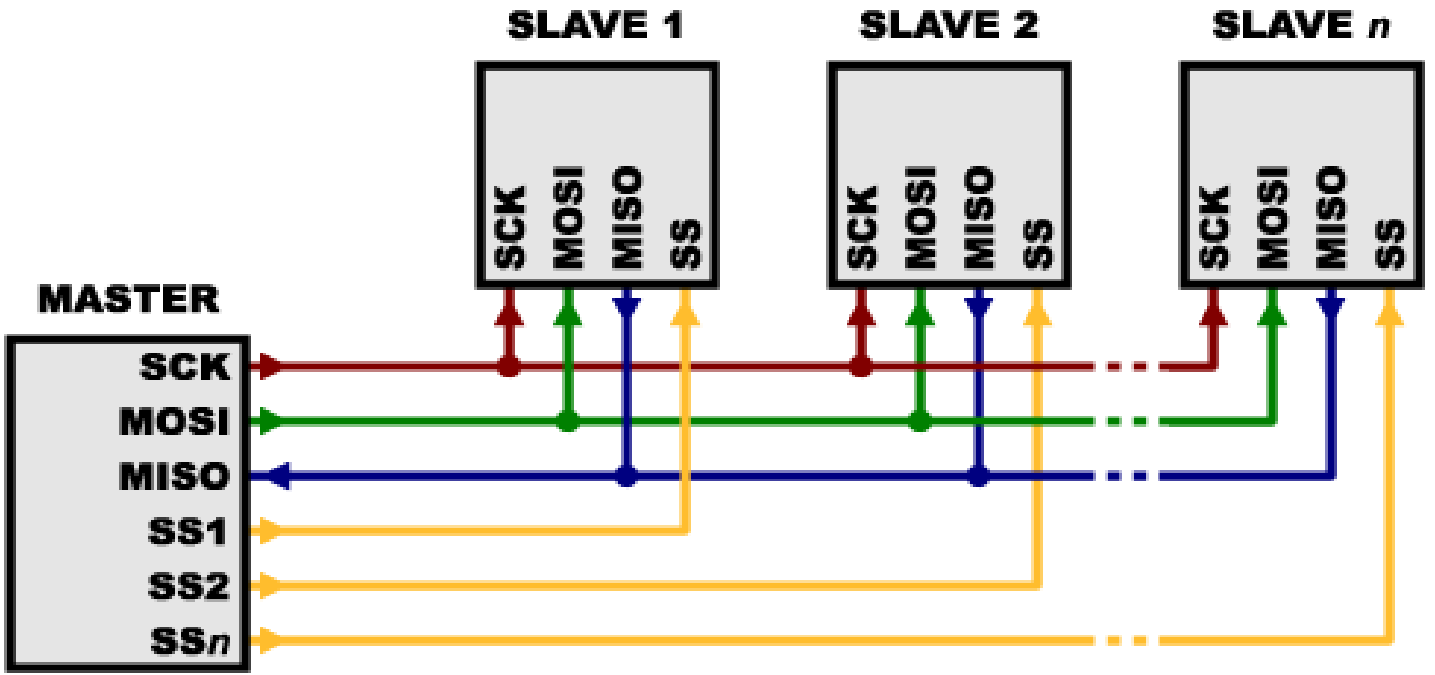
Cont...


MASTER
SCK
MOSI
MISO
SS
SLAVE
SCK
MOSI
MISO
SS
Master to Slave
Slave to Master
idle
idle or
next byte
SCK
Clock from
Master
0 1 2 3 4 5 6 7
0 1 2 3 4 5 6 7
MOSI
Master-Out
Slave-In
1 1 0 0 1 0 1 0
0x53 = ASCII 'S'
MISO
Master-In
Slave-Out
0 1 1 0 0 0 1 0
0x46 = ASCII 'F'
SS
Slave-Select
after last
byte sent
or received

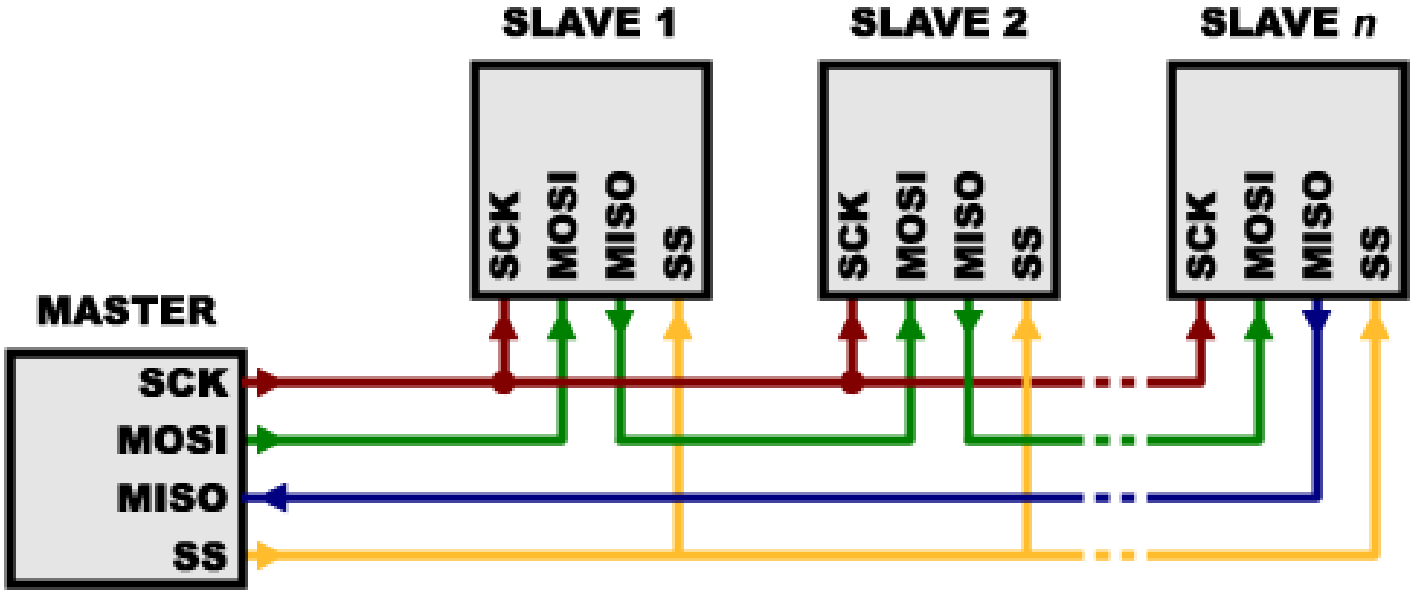
Cont...

- Multiple slaves : 2 ways
  - separate slave lines
  - single SS line (daisy chain configuration)

Separate Slave Lines



Single SS line



Cont...

- SPI transfer example
  - when master wants to initiate transfer, it must pull SS signal low for the slave it wants to communicate with
  - once SS signal is low, that slave will be listening on the bus
  - master is free to start sending data

Cont...

- There are 4 different types of SCK signal. The four modes are catagorized into two:
  - CPOL (clock polarity) : value low/high will decide the idle state of bus
  - CPHA (clock phase) : sampling of data during rising or falling edge of the clock

Cont...

SPI mode	Clock Polarity (CPOL)	Clock Phase (CPHA)	Clock Edge
0	0	0	1
1	0	1	0
2	1	0	1
3	1	1	0

Programming in SPI

- The programming interface is structured around two kinds of driver
  - Controller driver
  - Protocol driver

Cont...

```
struct spi_master{

    struct device dev;

    u16 num_chipselect;

    size_t (*)(struct spi_device *spi) max_transfer_size;

    int (*)(struct spi_device *spi,struct spi_message *mesg)
    transfer;

    void (*)(struct spi_device *spi) cleanup;

    …

    ...

};
```

Cont...

```
struct spi_device{
    struct device dev;
    struct spi_master *master;
    u8 chip_select;
    void *controller_state;
    void *controller_data;
    ...
    ...
};
```

Cont...

```
struct spi_board_info {
    const void * platform_data;
    void * controller_data;
    u16 chip_select;
    u16 mode;
    ...
    ...
};
```

Cont...

```
struct spi_driver {
    const struct spi_device_id * id_table;
    int (* probe) (struct spi_device *spi);
    int (* remove) (struct spi_device *spi);
    void (* shutdown) (struct spi_device *spi);
    struct device_driver driver;
};
```

Cont...

```
struct spi_message {
    struct spi_device * spi;
    unsigned frame_length;
    unsigned actual_length;
    int status;
    ...
    ...
};
```

Cont...

```
struct spi_transfer {
    unsigned tx_nbits:3;
    unsigned rx_nbits:3;
    u8 bits_per_word;
    u16 delay_usecs;
    u32 speed_hz;
    ...
    ...
};
```

Advantages of SPI

- It’s faster than asynchronous serial
- The receive hardware can be a simple shift register
- It supports multiple slaves

Disadvantages of SPI

- It requires more signal lines (wires) than other communications methods
- The communications must be well-defined in advance
- The master must control all communications
- It usually requires separate SS lines to each slave, which can be problematic if numerous slaves are needed

Video 4 Linux 2 (V4L2)

Introduction

- V4L is a collection of device drivers and an API for supporting real time video capture on Linux system.
- Programming a V4L2 device consists of below steps:
  - Opening the device - open()
  - Changing device properties, selecting a video and audio output, video standard, picture brightness - ioctl()
  - Negotiating a data format - ioctl()
  - Negotiating an input/output method - ioctl()
  - The actual input/output method – ioctl()
  - Closing the device - close()

Cont...

- V4L2 drivers are implemented as kernel modules.
- Driver modules plug into the “video-dev” kernel module
- Each driver has major number – 81, minor number ranges from 0 – 255
- The module options to select minor numbers are named after the device special file with a "_nr" suffix

  Ex: "video_nr" for /dev/video video capture devices

Data Structures

```
struct v4l2_capability

{

        __u32   capabilities;   /* Device capabilities */

        ...

} cap ;
```

Cont...

```
struct v4l2_format
{
        enum v4l2_buf_type type;
        union
        {
                struct v4l2_pix_format          pix;
                ...
        } fmt;
} fmt ;
```

Input/Output

- Read / Write
- Streaming
  - **Memory mapping** : Map buffers in device memory into the application’s address space
  - **User Pointer** : Buffers are allocated by the application itself. The driver must be switched into user pointer I/O mode with desired buffer type

Capturing Video

```
open_device( )

fd = open ( “/dev/video0”,   O_RDONLY ) ;


init_device ( )

ioctl ( fd ,VIDIOC_QUERYCAP, &cap ) ;

/* Select video input, video standard and format */

ioctl ( fd, VIDIOC_ENUM_FMT, &fmt ) ;


start_capturing ( )

 ioctl ( fd, VIDIOC_STREAMON, &type ) ;
```

Cont...

```
read_frame ( )

ioctl ( fd, VIDIOC_DQBUF, &buf )  ;

ioctl ( fd, VIDIOC_QBUF, &buf ) ;


process_image ( )

/* Application Dependent */
```

Cont...

```
stop_capturing ( )
 ioctl ( fd, VIDIOC_STREAMOFF, &type ) ;


uninit_device( )
free ( ) ;


close_device ( )
close ( fd ) ;
```


FEEDBACK

Large enough to Deliver, Small enough to Care

Global Village
IT SEZ
Bangalore

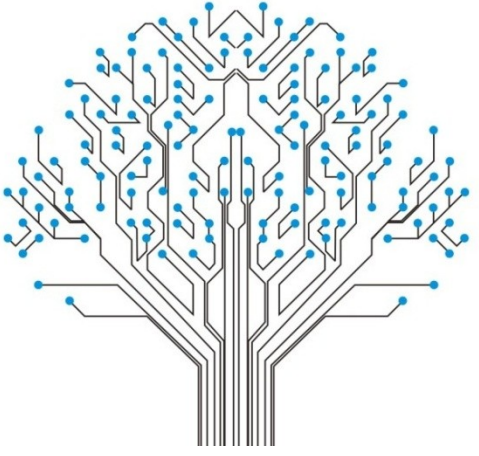
South Main Street
Milpitas
California

Raheja Mindspace
IT Park
Hyderabad

www.globaledgesoft.com

GLOBAL EDGE
Intelligence Of Things™

Thank you



Fairness

Learning

Responsibility

Innovation

Respect