

# Mastering Embedded System

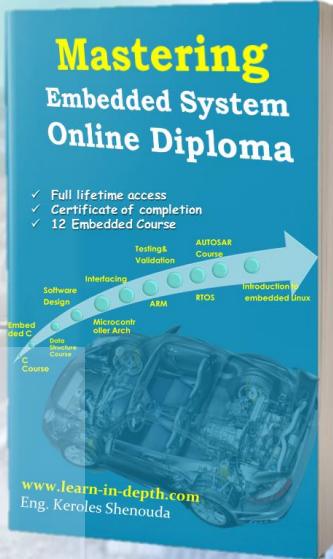
## Online Diploma

- ✓ Full lifetime access
- ✓ Access on Android mobile and PC (Windows)
- ✓ Certificate of completion
- ✓ 12 Embedded Course

### Unit 8 (MCU Interfacing) . lesson 4 SPI Protocol

- ▶ SPI (Serial Peripheral Interface) Protocol Characteristics
  - ▶ Advantages of SPI
  - ▶ Disadvantages of SPI
  - ▶ Master with Single Slave Connection
  - ▶ SPI Throughput
  - ▶ Master Slave Setup
- ▶ How SPI works ?
- ▶ SPI Bus Configuration
  - ▶ Full-Duplex Communication
  - ▶ Half-Duplex Communication
  - ▶ Simplex Communication
- ▶ Multiple slave cascaded
- ▶ Daisy Chain Configuration

- ▶ SPI With Flash Memories (custom protocol)
- ▶ SPI Bus 3-Wire
- ▶ QUAD SPI [QSPI]
- ▶ EX. QSPI connected to Spansion Flash
- ▶ SPI/I2C/UART/CAN/LIN/USB comparison
  - ▶ Speed
  - ▶ Distance
  - ▶ Properties
  - ▶ Notes



1

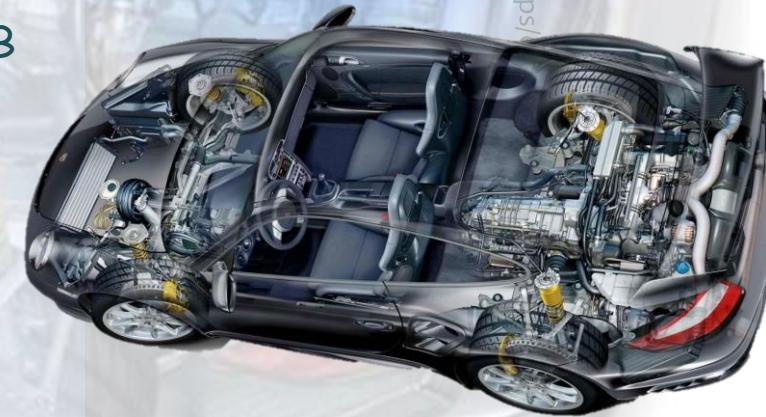


#LEARN\_IN\_DEPTH

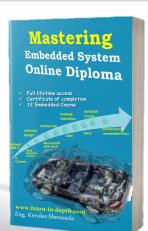
#Be\_professional\_in  
embedded\_system

eng. Keroles Shenouda

<https://www.facebook.com/groups/>



<https://www.learn-in-depth.com/>  
<https://www.facebook.com/groups/embedded.system.KS/>



2

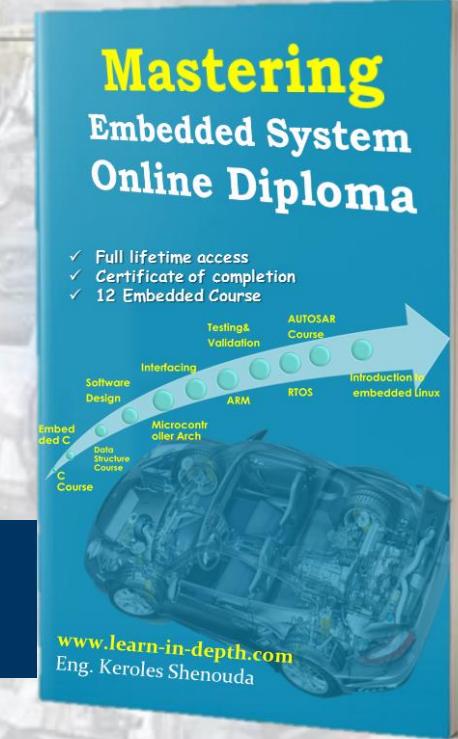
#LEARN\_IN\_DEPTH

#Be\_professional\_in\_embedded\_system

eng. Keroles Shenouda

<https://www.facebook.com/groups/embedded.system.KS/>

*it's time to wake up 😊*

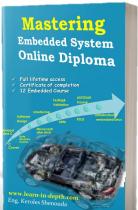


[www.learn-in-depth.com](http://www.learn-in-depth.com)  
Eng. Keroles Shenouda

**LEARN-IN-DEPTH**  
Be professional in  
embedded system



<https://www.learn-in-depth.com/>  
<https://www.facebook.com/groups/embedded.system.KS/>

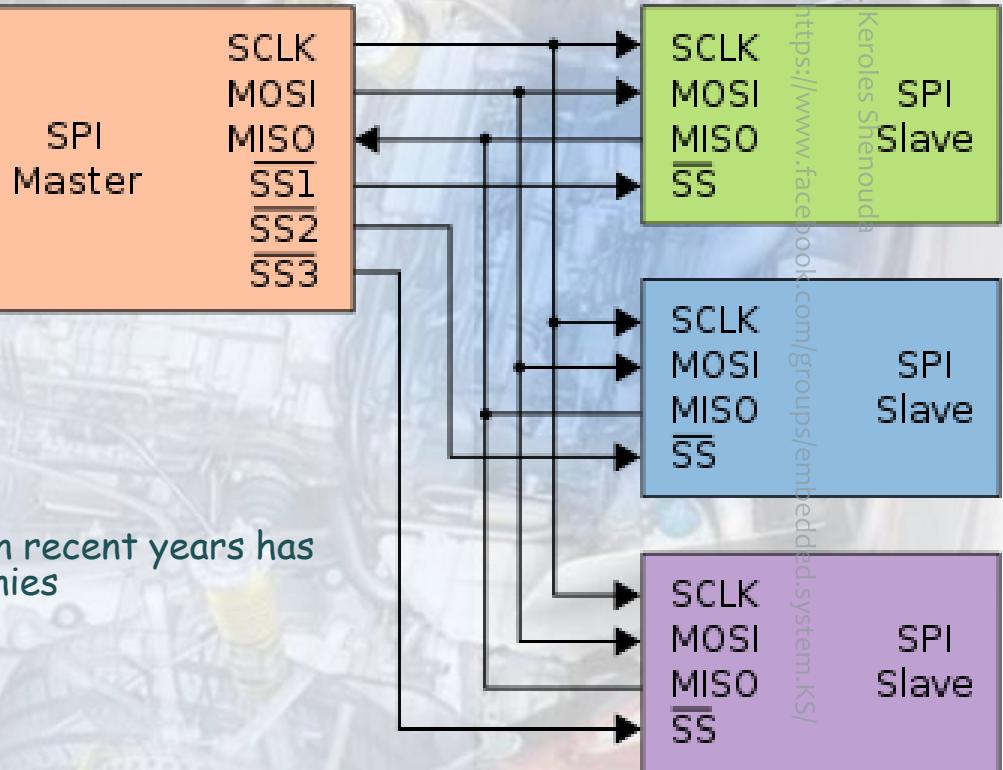


#LEARN\_IN\_DEPTH  
#Be\_professional\_in  
embedded\_system

eng. Keroles Shenouda  
<https://www.facebook.com/groups/embedded.system.KS/>

SPI Slave

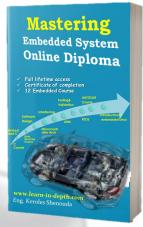
SPI Slave



# SPI (Serial Peripheral Interface)

- ▶ **synchronous** data transfer
- ▶ **Full-duplex**
- ▶ **Single Master / multi slave**
  - ▶ Serial interface
  - ▶ Multiple slave devices are supported through selection with individual slave select (SS) lines
  - ▶ The SPI bus was originally started by Motorola Corp. (now Freescale), but in recent years has become a widely used standard adapted by many semiconductor chip companies
  - ▶ It can be used to communicate with a serial peripheral device like external EEPROM or with another microcontroller with an SPI interface

<https://www.learn-in-depth.com/>  
<https://www.facebook.com/groups/embedded.system.KS/>



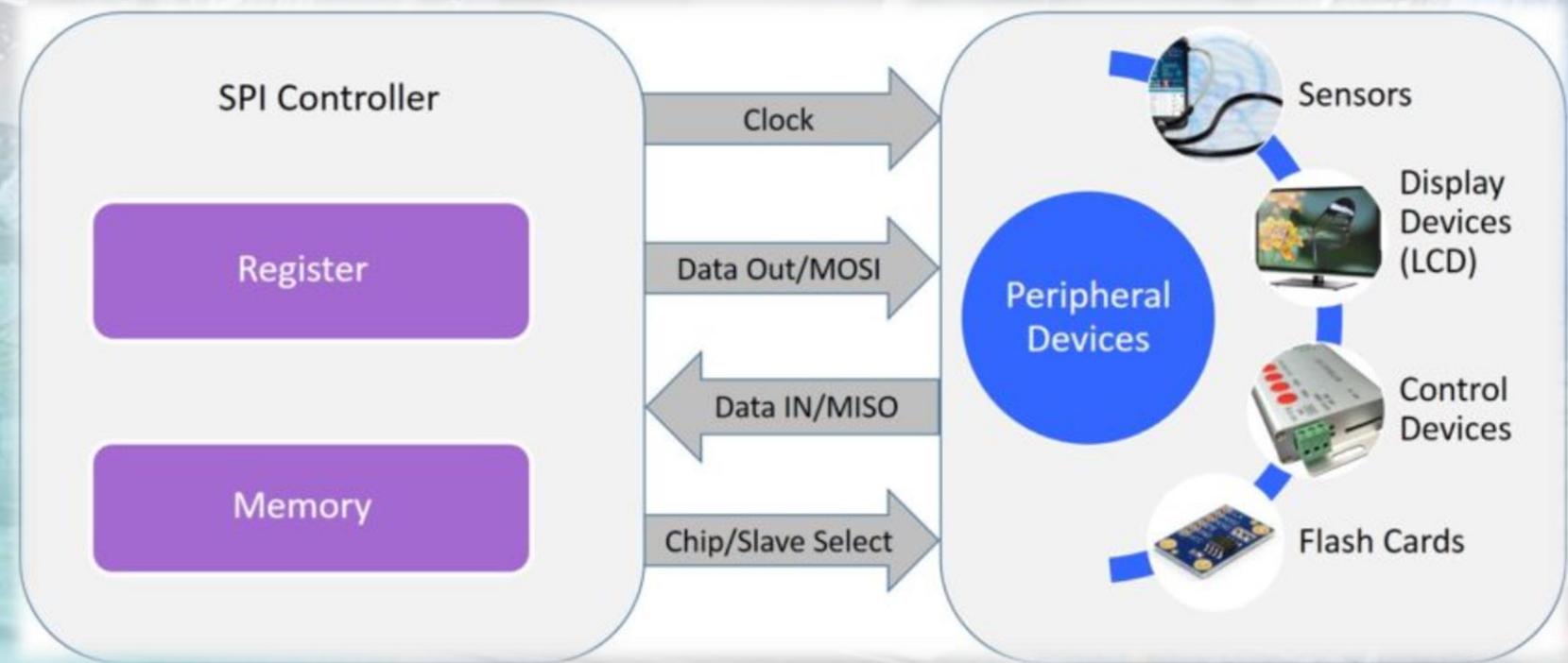
4

#LEARN\_IN\_DEPTH  
#Be\_professional\_in\_embedded\_system

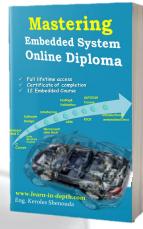
Eng. Keroles Shenouda

<https://www.facebook.com/groups/embedded.system.KS/>

# SPI (Single Master > Multiple Slaves)



<https://www.learn-in-depth.com/>  
<https://www.facebook.com/groups/embedded.system.KS/>



5

#LEARN\_IN\_DEPTH  
#Be\_professional\_in\_embedded\_system

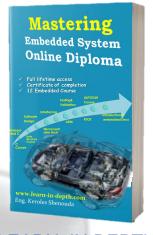
eng. Keroles Shenouda

<https://www.facebook.com/groups/embedded.system.KS/>

# Advantages of SPI

- ▶ Higher **throughput** (Faster)  
SPI: 10Mbs - 20Mbs
- ▶ Simple Receiver Hardware >>> Simple Shift Register
- ▶ Support Multiple Slaves
- ▶ Lower power Requirements than I2C.

<https://www.learn-in-depth.com/>  
<https://www.facebook.com/groups/embedded.system.KS/>



6

#LEARN\_IN\_DEPTH  
#Be\_professional\_in\_embedded\_system

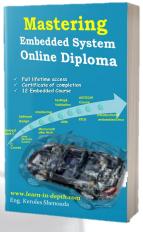
eng. Keroles Shenouda

<https://www.facebook.com/groups/embedded.system.KS/>

# Disadvantages of SPI

- ▶ Requires **more** Pins
- ▶ Master must control all communications (Slave doesn't issue the transaction).
- ▶ **Separate SS Lines.**
- ▶ No Flow Control (must know slave speed)

<https://www.learn-in-depth.com/>  
<https://www.facebook.com/groups/embedded.system.KS/>

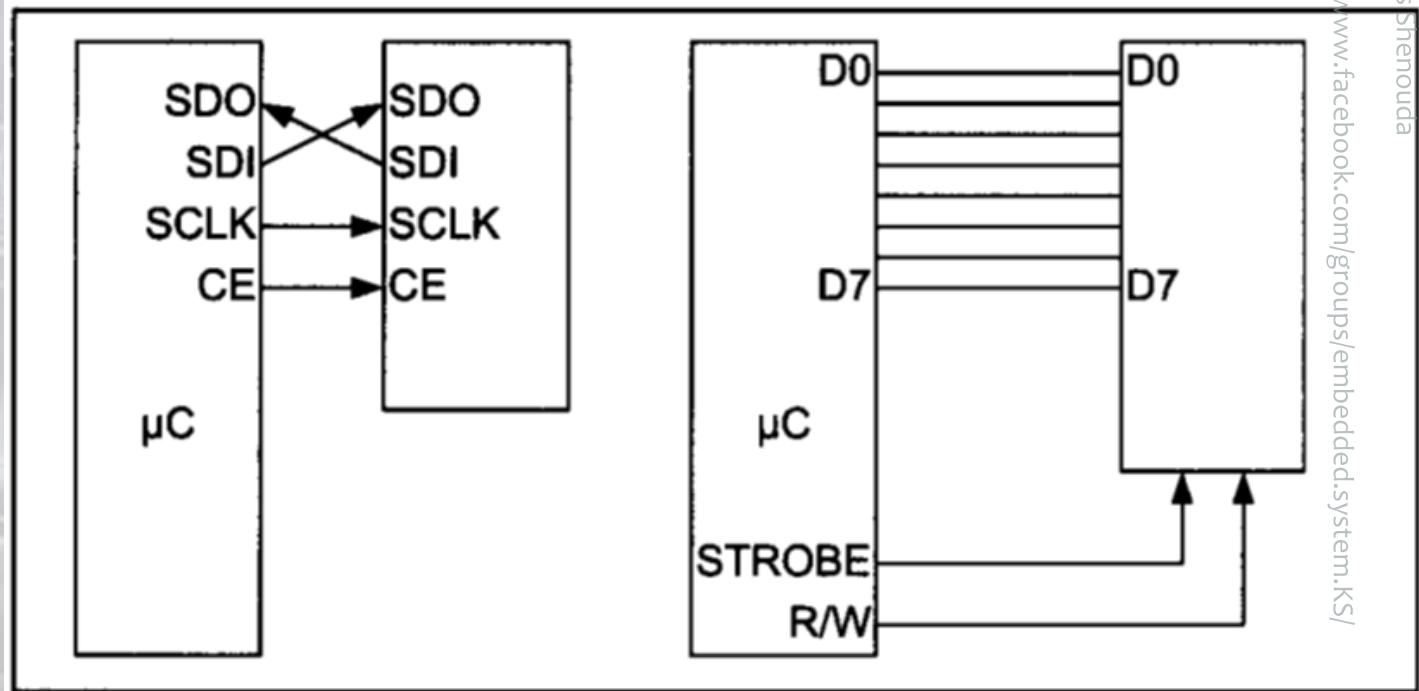


7

#LEARN\_IN\_DEPTH  
#Be\_professional\_in\_embedded\_system<https://www.facebook.com/groups/embedded.system.KS/>

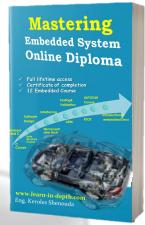
# SPI BUS Protocol

- ▶ These 4 pins, SDI, SDO, SCLK, and CE, make the SPI a 4-wire interface.
- ▶ The SDI, SDO, SCLK, and CE signals are alternatively named as MOSI, MISO, SCK, and SS.



SPI Bus vs. Traditional Parallel Bus Connection to Microcontroller

<https://www.learn-in-depth.com/>  
<https://www.facebook.com/groups/embedded.system.KS/>



8

#LEARN\_IN\_DEPTH

#Be\_professional\_in\_

eng.Keroles Shenouda

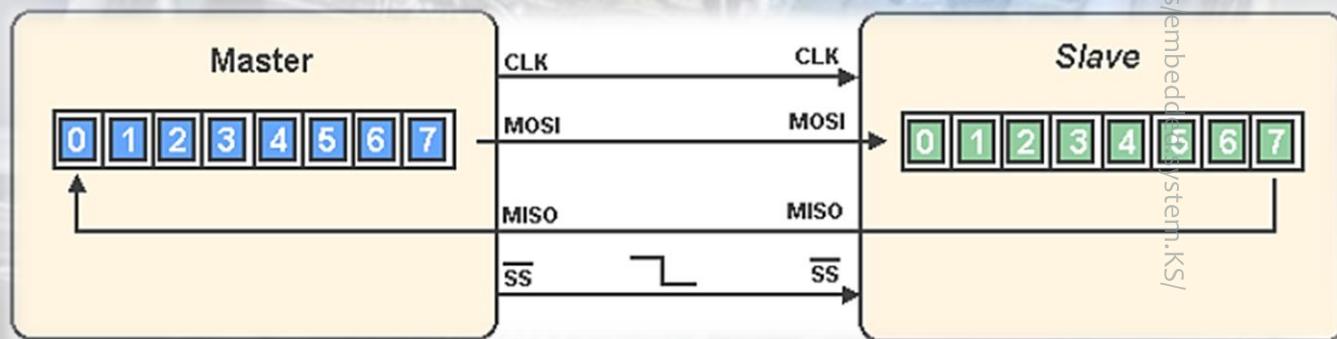
<https://www.facebook.com/groups/embedded.system.KS/>

# SPI Interface

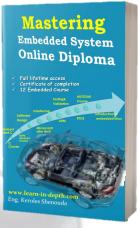
The SPI bus specifies four logic signals:

- **SCLK(SCK,CLK)** : Serial Clock (output from **master**).
- **MOSI(SIMO,SDI,DI)** : Master Output, Slave Input (output from master).
- **MISO(SIMO,SDO,DO)** : Master Input, Slave Output (output from slave).
- **SS (CS,CE,CEN)**: Slave Select (active low, output from master).

## Data Transmission



<https://www.learn-in-depth.com/>  
<https://www.facebook.com/groups/embedded.system.KS/>



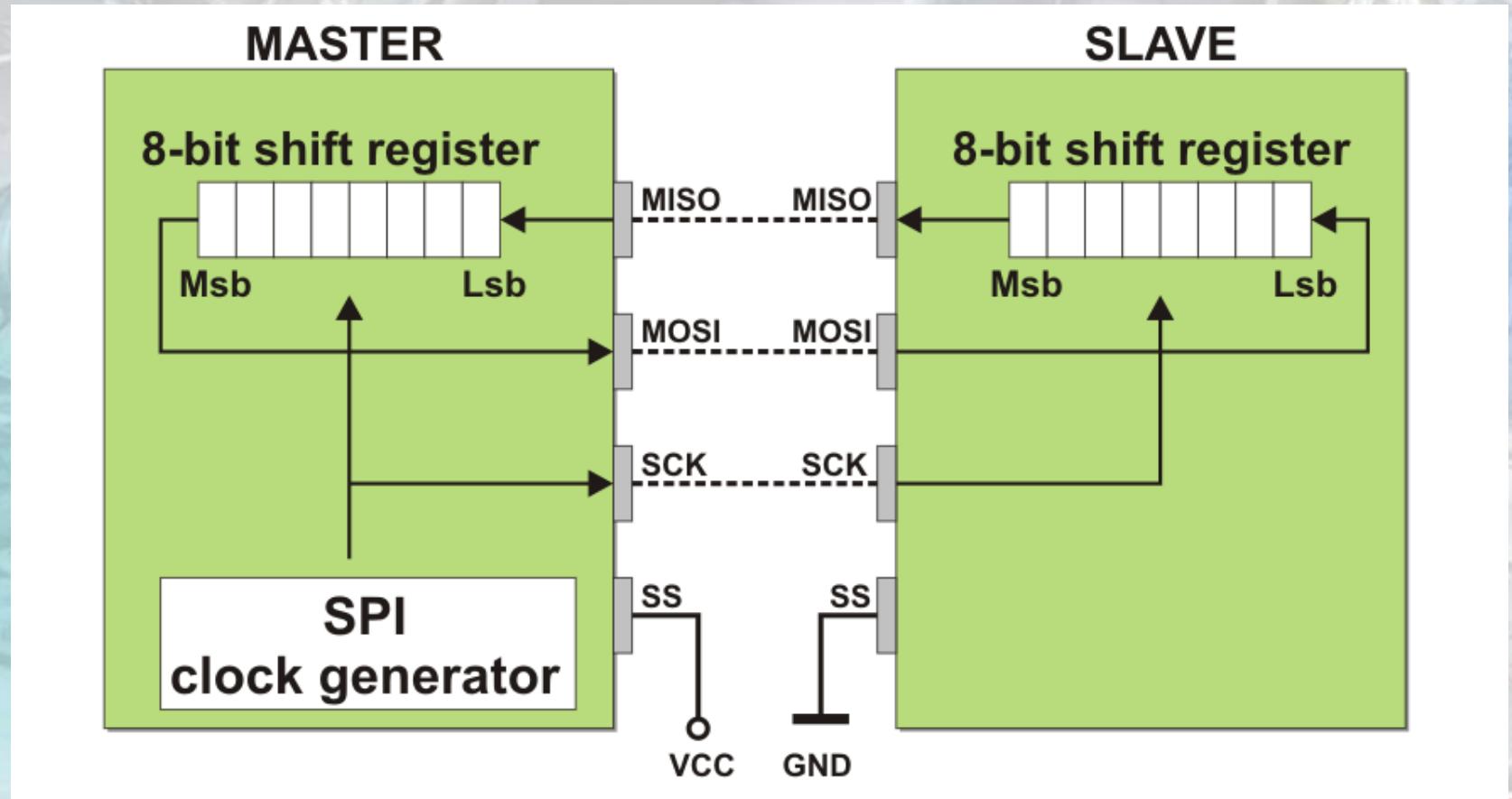
9

#LEARN\_IN\_DEPTH  
#Be\_professional\_in\_embedded\_system

eng. Keroles Shenouda

<https://www.facebook.com/groups/embedded.system.KS/>

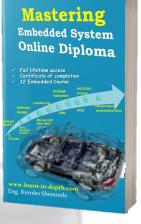
# Master with Single Slave



<https://www.learn-in-depth.com/>  
<https://www.facebook.com/groups/embedded.system.KS/>



10

#LEARN\_IN\_DEPTH  
#Be\_professional\_in\_embedded\_system

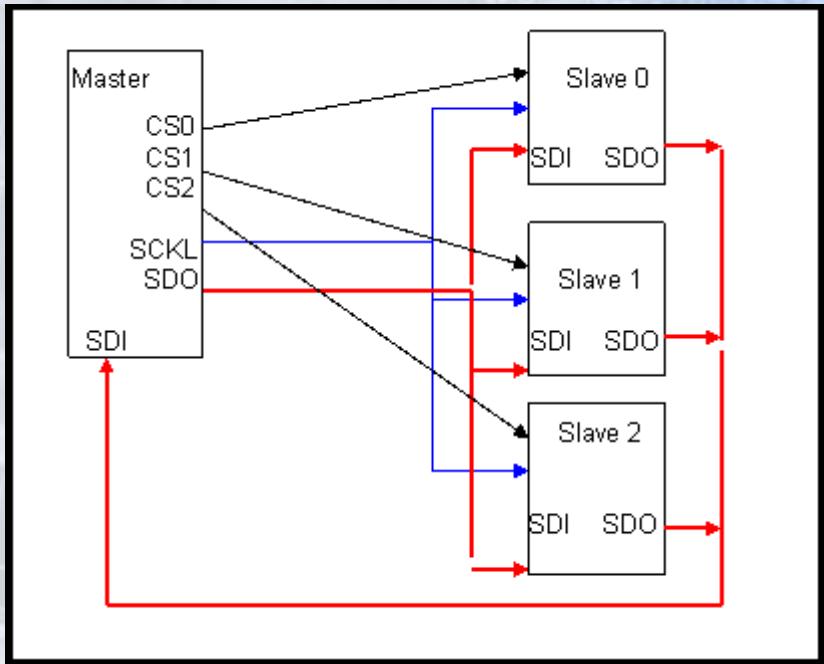
eng. Keroles Shenouda

<https://www.facebook.com/groups/embedded.system.KS/>

# Master Slave Setup

- ▶ In this setup, there are 3 slave devices. The SDO lines are tied together to the SDI line of the master.
- ▶ The master determines which chip it is talking to **by the CS lines**.
- ▶ For the slaves that are not being talked to, the data **output goes to a Hi Z state**

- ▶ Multiple Independent Slave Configuration



<https://www.learn-in-depth.com/>  
<https://www.facebook.com/groups/embedded.system.KS/>



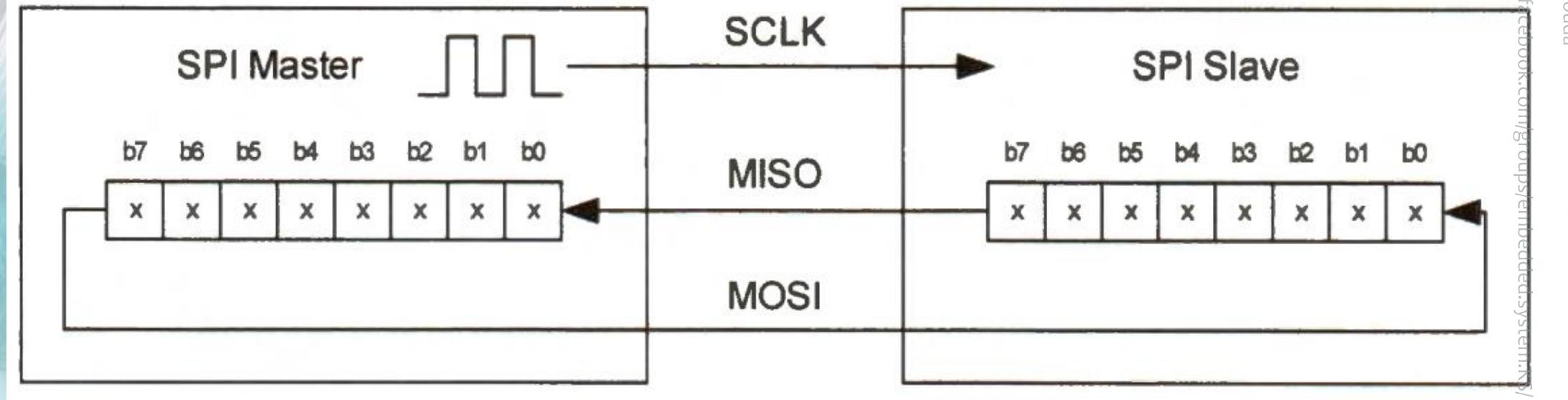
11

#LEARN\_IN\_DEPTH  
#Be\_professional\_in\_embedded\_system

eng. Keroles Shenouda

<https://www.facebook.com/groups/embedded.system.KS/>

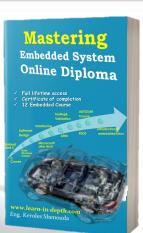
# SPI master provides clock signal (SCLK) to SPI slaves.



<https://www.learn-in-depth.com/>  
<https://www.facebook.com/groups/embedded.system.KS/>



12



#LEARN\_IN\_DEPTH

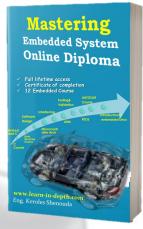
#Be\_professional\_in\_embedded\_system

eng. Keroles Shenouda

<https://www.facebook.com/groups/embedded.system.KS/>

# How SPI works ?

<https://www.learn-in-depth.com/>  
<https://www.facebook.com/groups/embedded.system.KS/>



13

#LEARN\_IN\_DEPTH  
#Be\_professional\_in\_embedded\_system

https://www.facebook.com/groups/embedded.system.KS/  
eng. Keroles Shenouda

# How SPI works ?

- ▶ SPI consists of two shift registers, one in the master and the other in the slave side.
- ▶ Also, there is a clock generator in the master side that generates the clock for the shift registers.

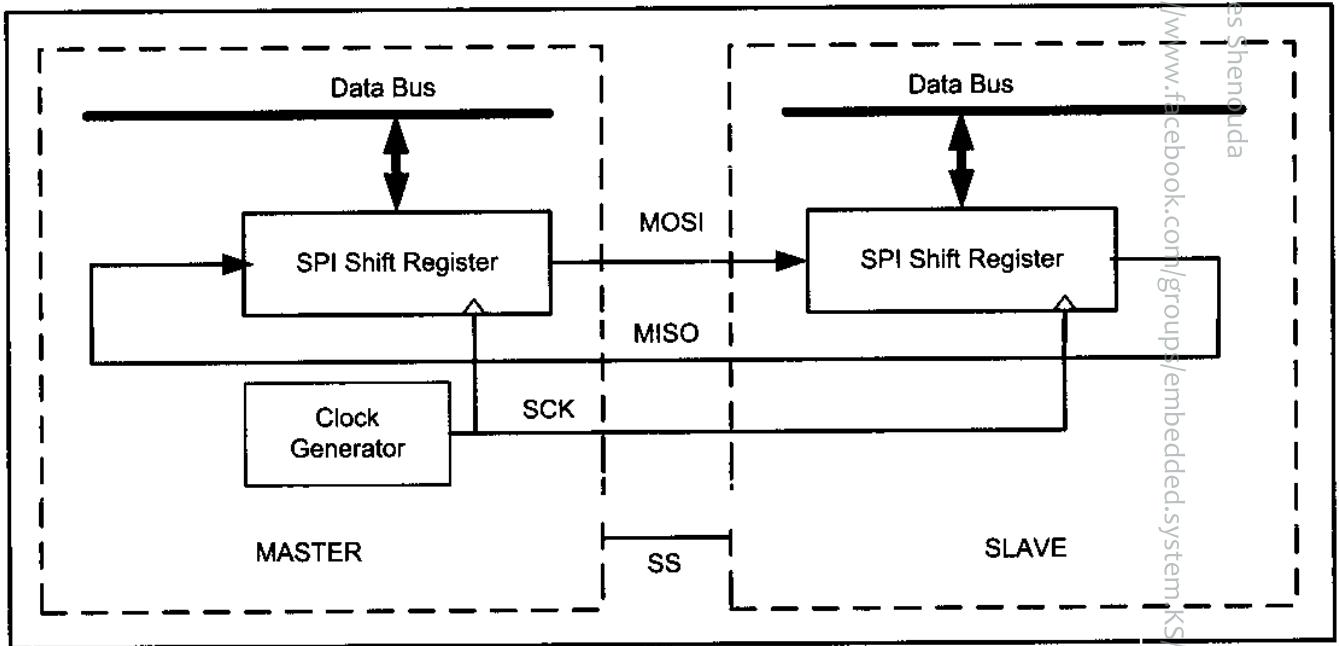
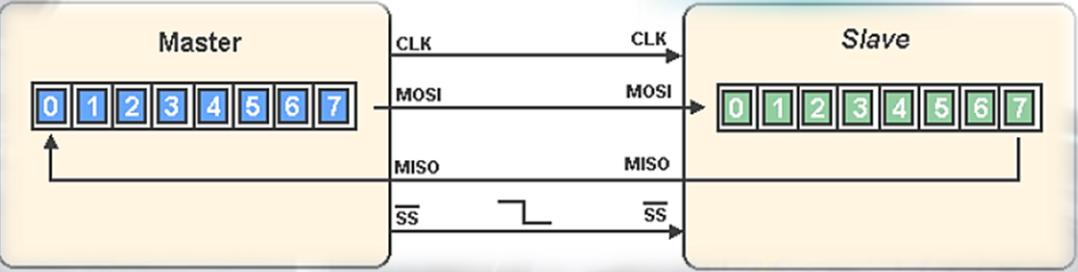


Figure: SPI Architecture

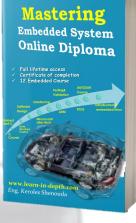
<https://www.learn-in-depth.com/>  
<https://www.facebook.com/groups/embedded.system.KS/>

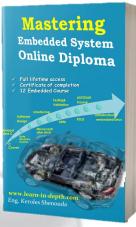
# HOW SPI WORKS?



- ✓ The serial-out pin of the master shift register is connected to the serial-in pin of the slave shift register by **MOSI (Master Out Slave In)**.
- ✓ The serial-in pin of the master shift register is connected to the serial-out pin of the slave shift register by **MISO (Master In Slave Out)**.
- ✓ The master clock generator provides clock to the shift registers in both the master and slave.
- ✓ The clock input of the shift registers can be **falling- or rising-edge triggered**.
- ✓ Shift registers are 8 bits long. So after **8 clock pulses**, the contents of the two shift registers are interchanged.
- ✓ When the master wants to **send a byte of data**, it places the byte in its **shift register** and generates **8 clock pulses**.

<https://www.learn-in-depth.com/>  
<https://www.facebook.com/groups/embedded.system.KS/>





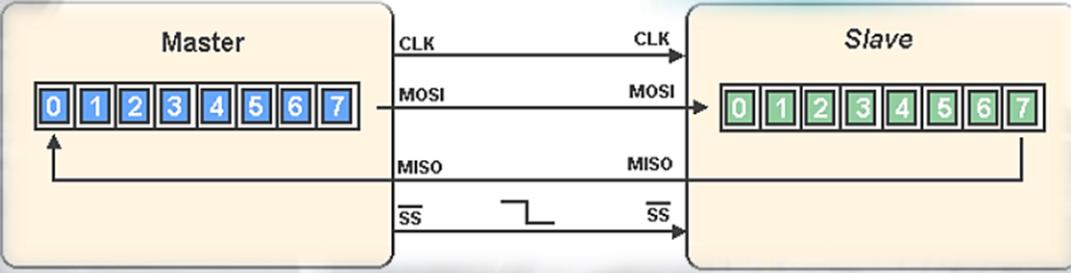
15

#LEARN\_IN\_DEPTH  
#Be\_professional\_in\_embedded\_system

eng. Keroles Shenouda

<https://www.facebook.com/groups/embedded.system.KS/>

## HOW SPI WORKS?



- ▶ After 8 clock pulses the byte is transmitted to the other shift register.
- ▶ When the master wants to receive a byte of data, the slave side should place the byte in its shift register, and after 8 clock pulses the data will be received by the master shift register.
- ▶ It must be noted that SPI is full duplex, meaning that it sends and receives data at the same time.

<https://www.learn-in-depth.com/>  
<https://www.facebook.com/groups/embedded.system.KS/>



16

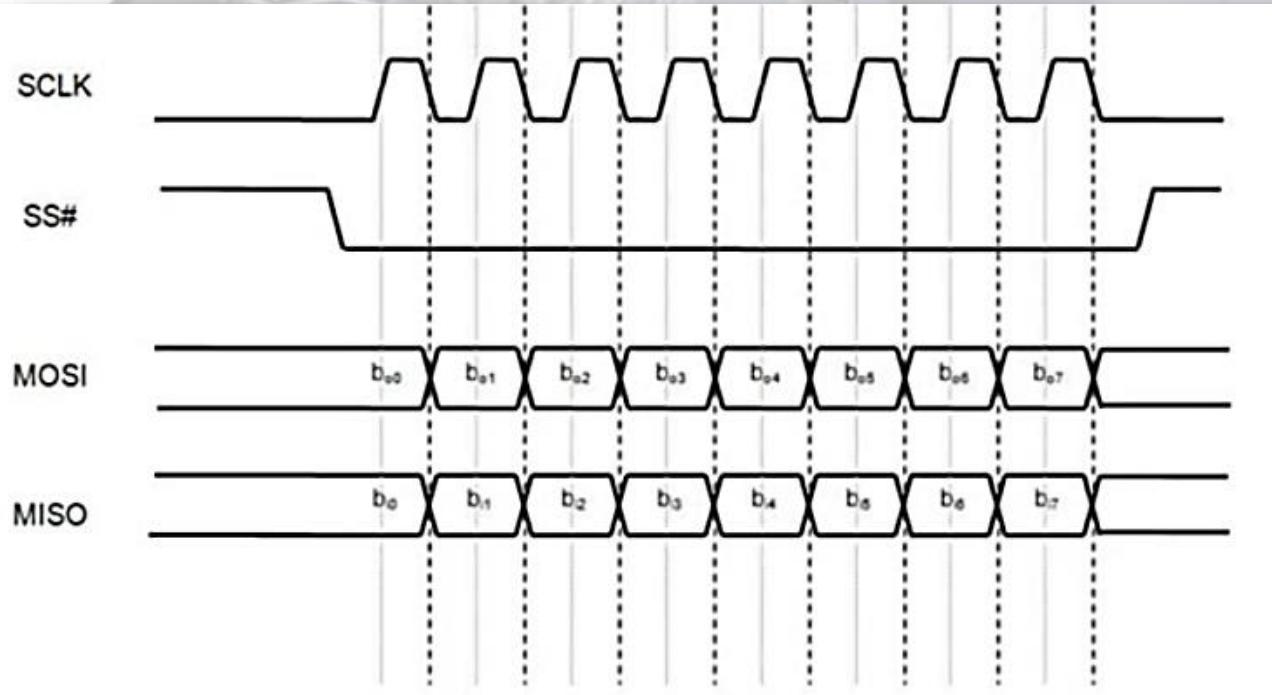
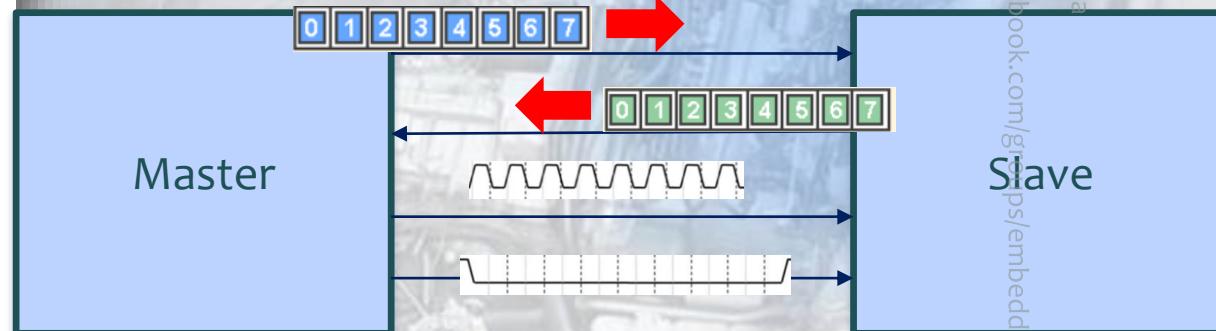


#LEARN\_IN\_DEPTH

#Be\_professional\_in  
embedded\_system<https://www.facebook.com/groups/embedded.system.KS/>

Slave

Master

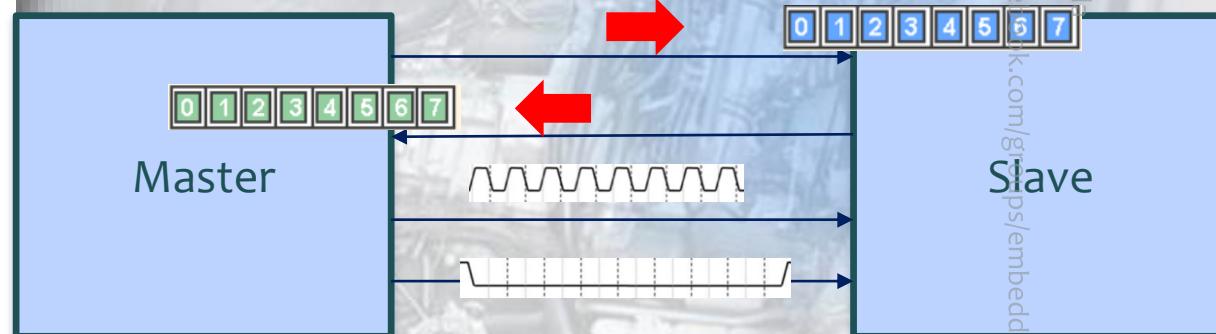
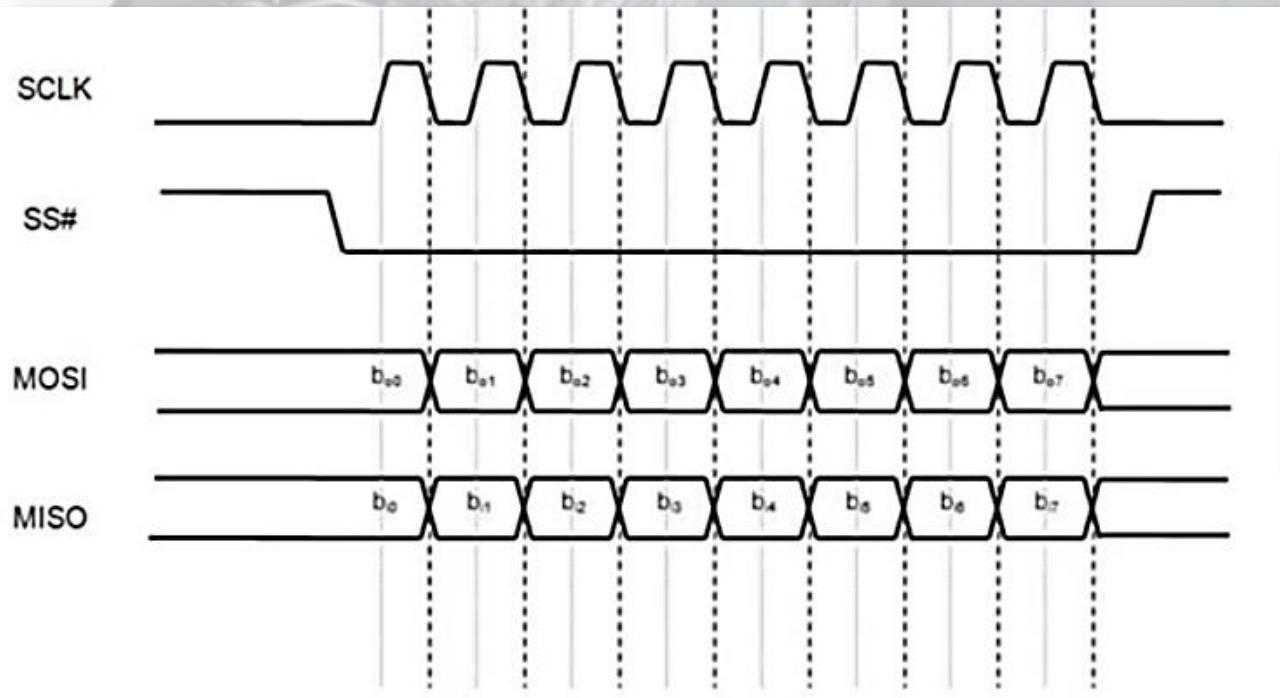


<https://www.learn-in-depth.com/>  
<https://www.facebook.com/groups/embedded.system.KS/>

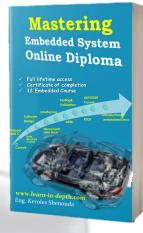


# SPI " Serial Peripheral Interface "

## How SPI works ?



<https://www.learn-in-depth.com/>  
<https://www.facebook.com/groups/embedded.system.KS/>



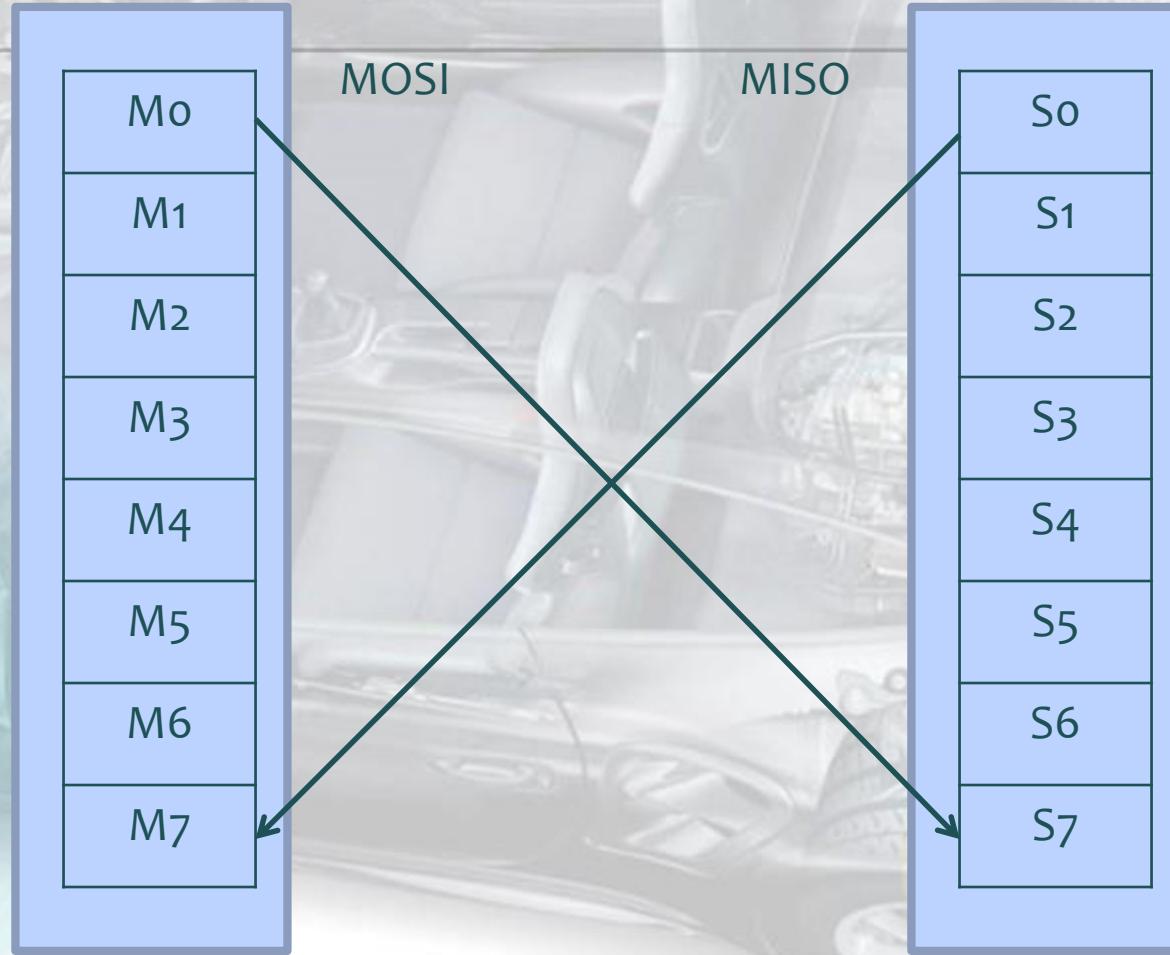
18

#LEARN\_IN\_DEPTH  
#Be\_professional\_in\_embedded\_system

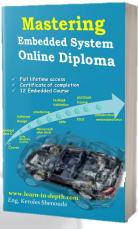
<https://www.facebook.com/groups/embedded.system.KS/>

# Data Transfer in SPI

MASTER



<https://www.learn-in-depth.com/>  
<https://www.facebook.com/groups/embedded.system.KS/>

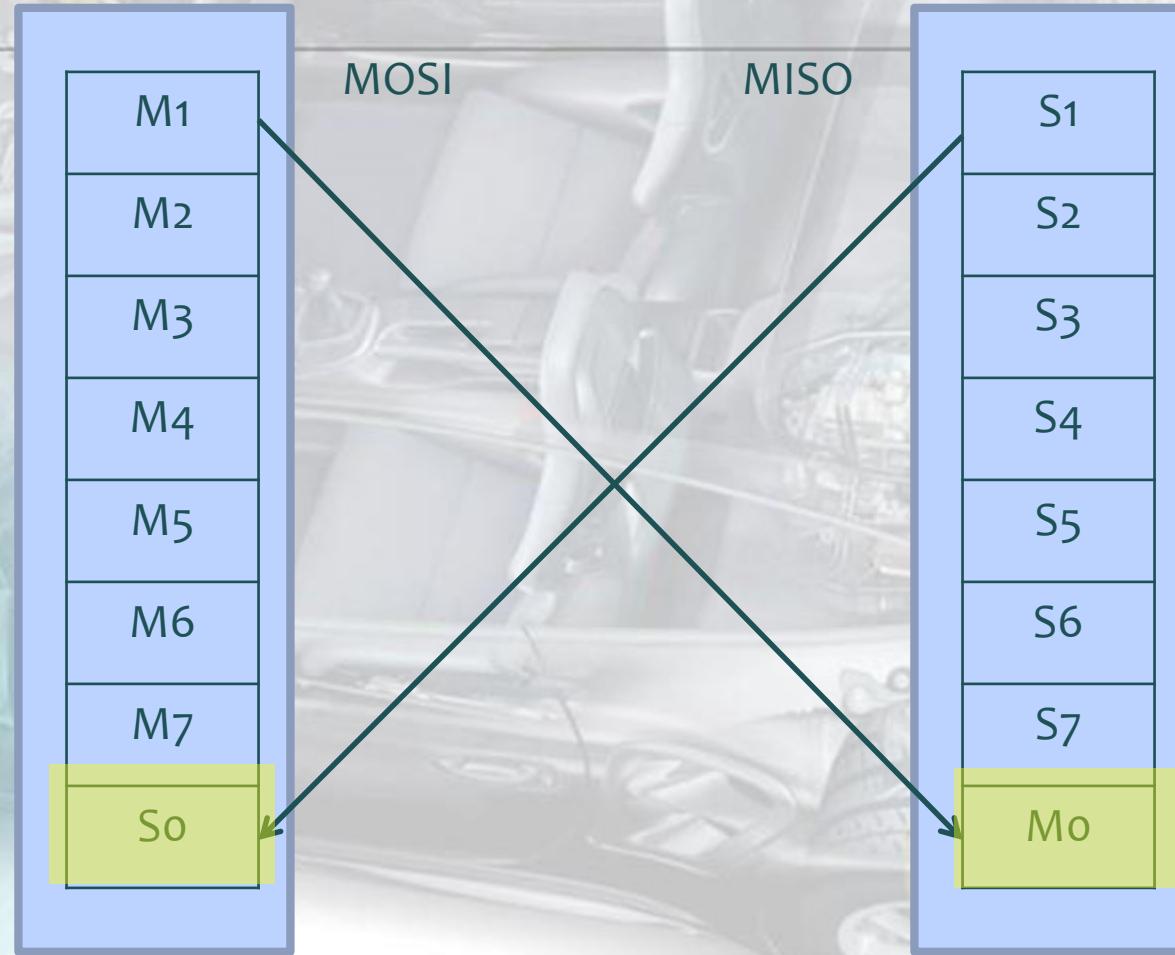


19

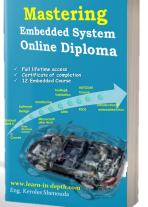
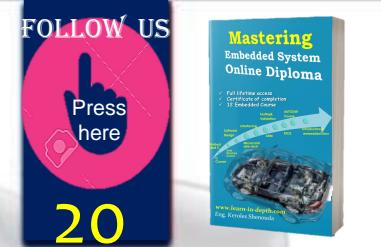
#LEARN\_IN\_DEPTH  
#Be\_professional\_in\_embedded\_system<https://www.facebook.com/groups/embedded.system.KS/>

# Data Transfer in SPI

MASTER



<https://www.learn-in-depth.com/>  
<https://www.facebook.com/groups/embedded.system.KS/>



#LEARN\_IN\_DEPTH

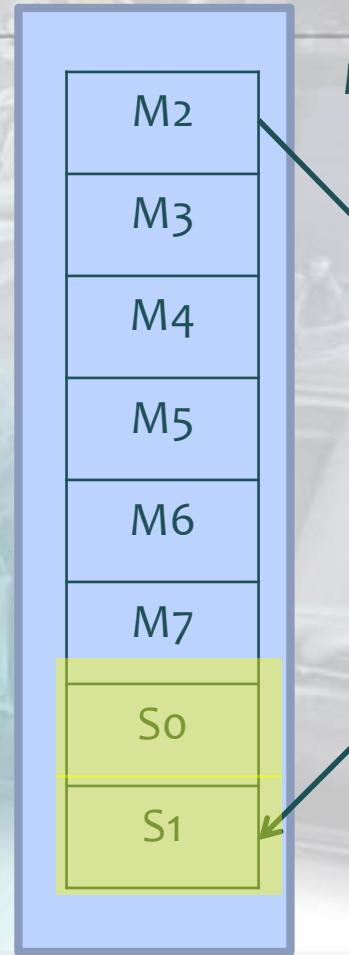
#Be\_professional\_in\_embedded\_system

eng. Keroles Shenouda

<https://www.facebook.com/groups/embedded.system.KS/>

# Data Transfer in SPI

MASTER

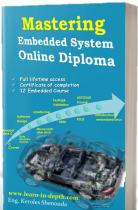


MOSI

MISO

SLAVE

<https://www.learn-in-depth.com/>  
<https://www.facebook.com/groups/embedded.system.KS/>

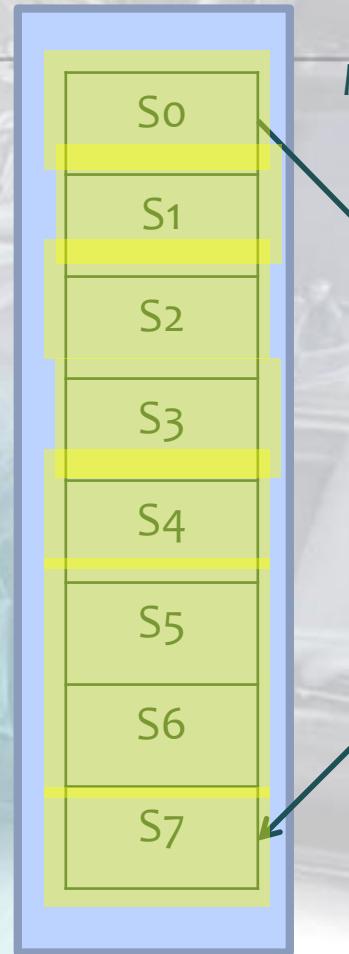


21

#LEARN\_IN\_DEPTH  
#Be\_professional\_in\_embedded\_system<https://www.facebook.com/groups/embedded.system.KS/>

# Data Transfer in SPI

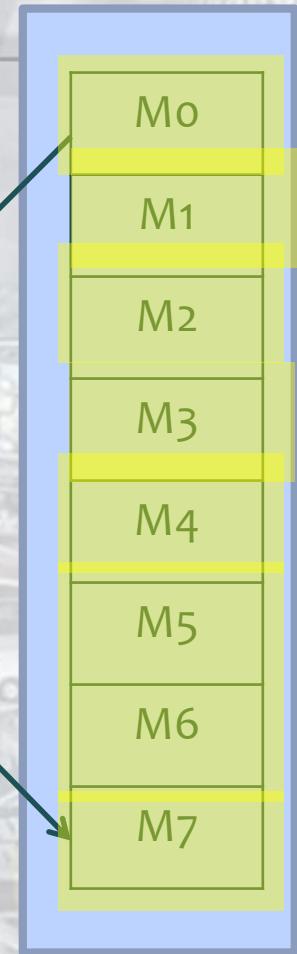
MASTER



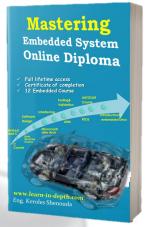
MOSI

MISO

SLAVE



<https://www.learn-in-depth.com/>  
<https://www.facebook.com/groups/embedded.system.KS/>

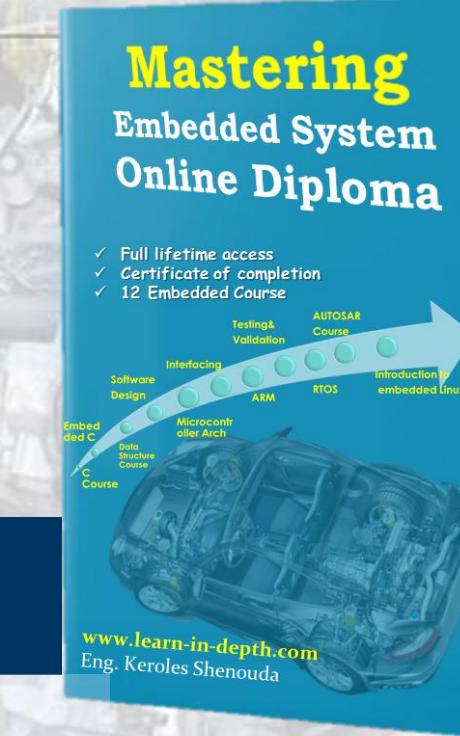


22

#LEARN\_IN\_DEPTH

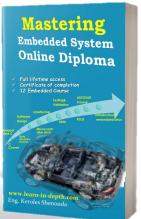
#Be\_professional\_in\_embedded\_system

eng. Keroles Shenouda

<https://www.facebook.com/groups/embedded.system.KS/>

**LEARN-IN-DEPTH**  
Be professional in  
**embedded system**

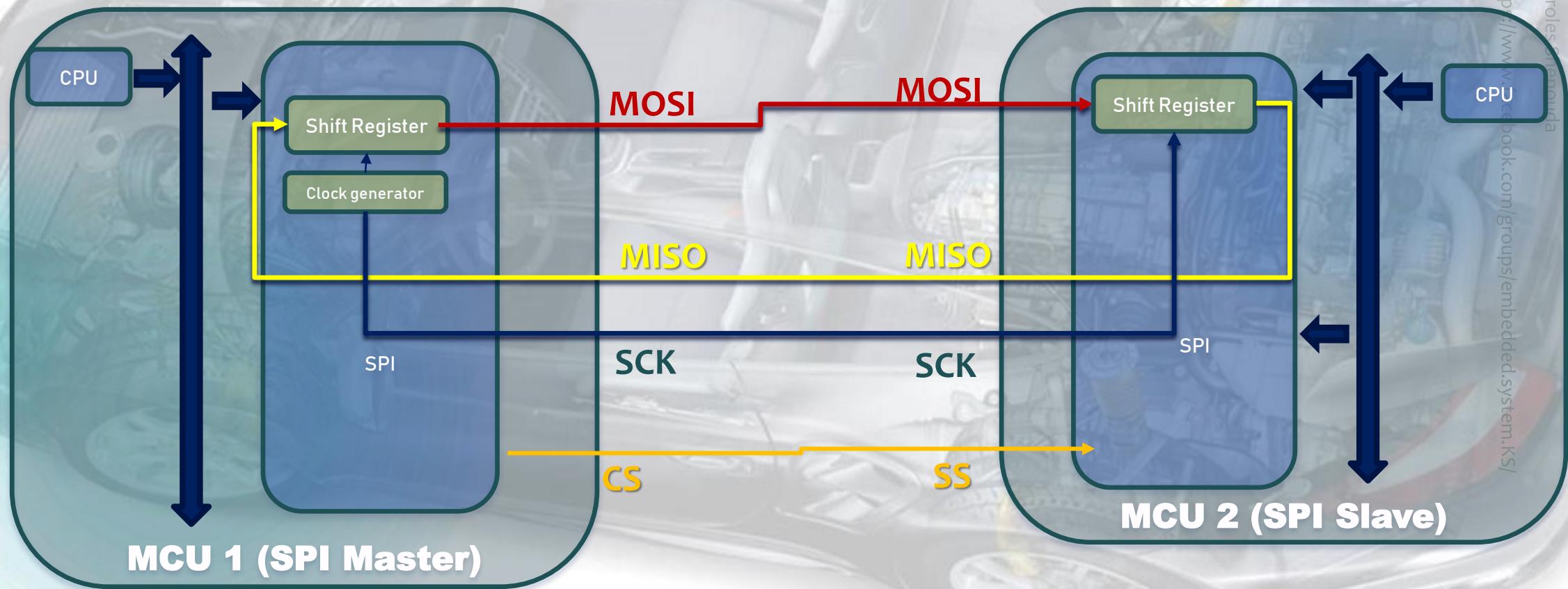
<https://www.learn-in-depth.com/>  
<https://www.facebook.com/groups/embedded.system.KS/>



23

#LEARN\_IN\_DEPTH  
#Be\_professional\_in\_embedded\_systemhttp://www.learn-in-depth.com/groups/embedded.system.KS/  
eng\_KerolesShenouda

# Full-Duplex Communication



<https://www.learn-in-depth.com/>  
<https://www.facebook.com/groups/embedded.system.KS/>



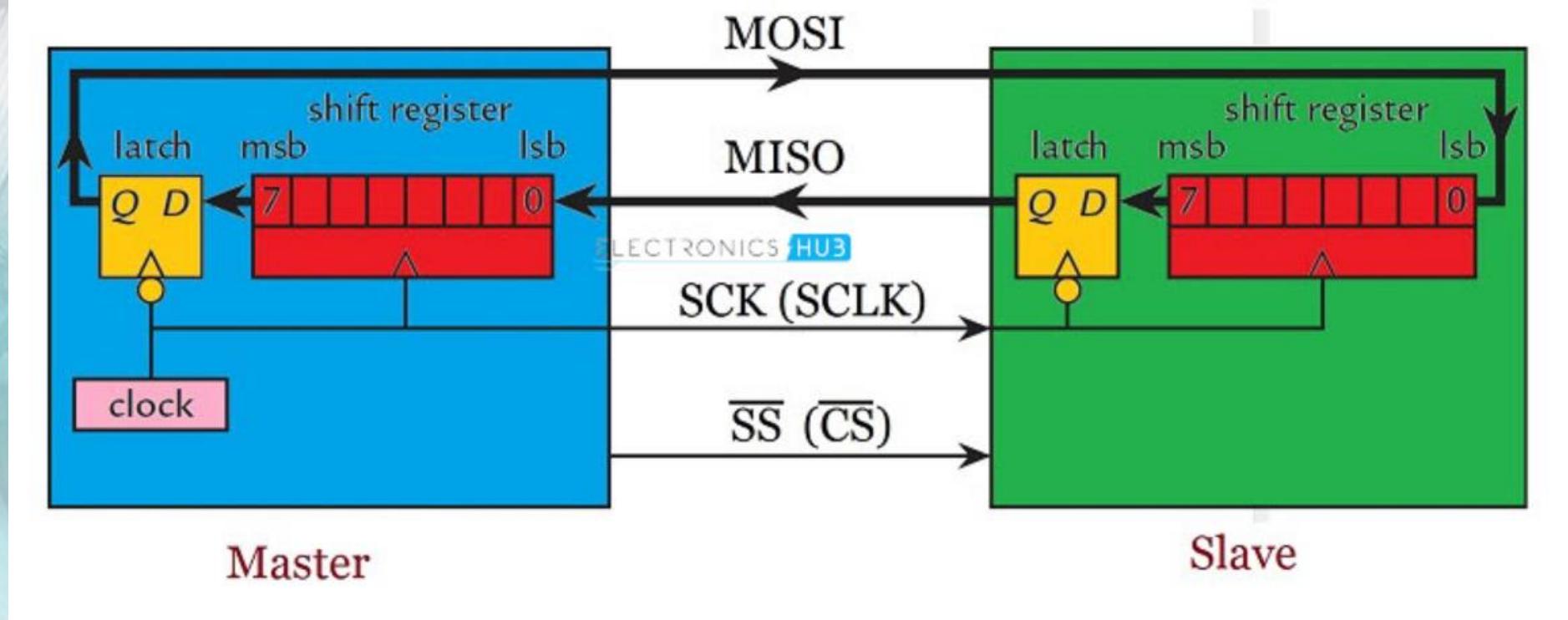
24

#LEARN\_IN\_DEPTH  
#Be\_professional\_in\_embedded\_system

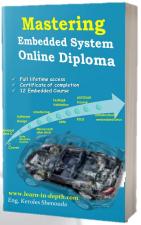
eng. Keroles Shenouda

<https://www.facebook.com/groups/embedded.system.KS/>

# Full-Duplex Communication



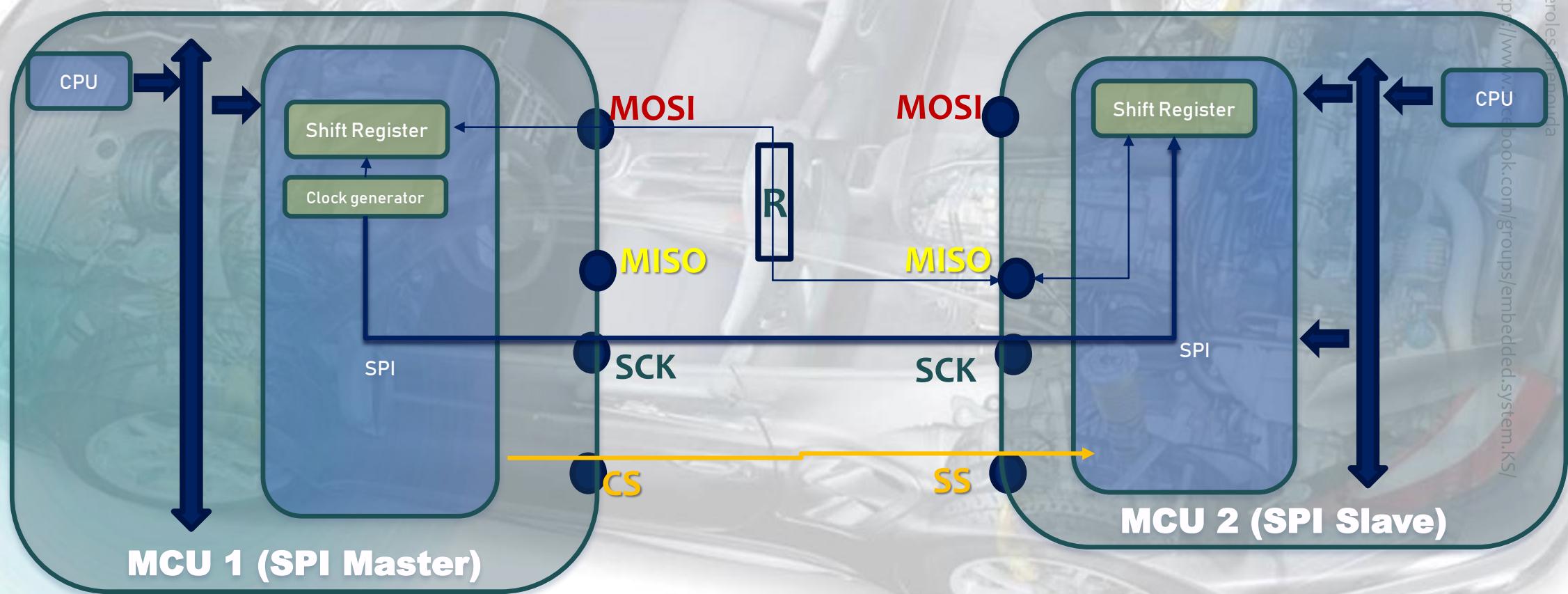
<https://www.learn-in-depth.com/>  
<https://www.facebook.com/groups/embedded.system.KS/>



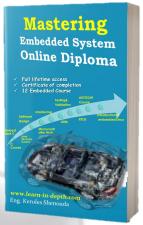
25

#LEARN\_IN\_DEPTH  
#Be\_professional\_in\_embedded\_systemhttp://www.learn-in-depth.com/groups/embedded.system.KS/  
eng\_KerolesShenouda

# Half-Duplex Communication



<https://www.learn-in-depth.com/>  
<https://www.facebook.com/groups/embedded.system.KS/>



#LEARN\_IN\_DEPTH

#Be\_professional\_in\_embedded\_system

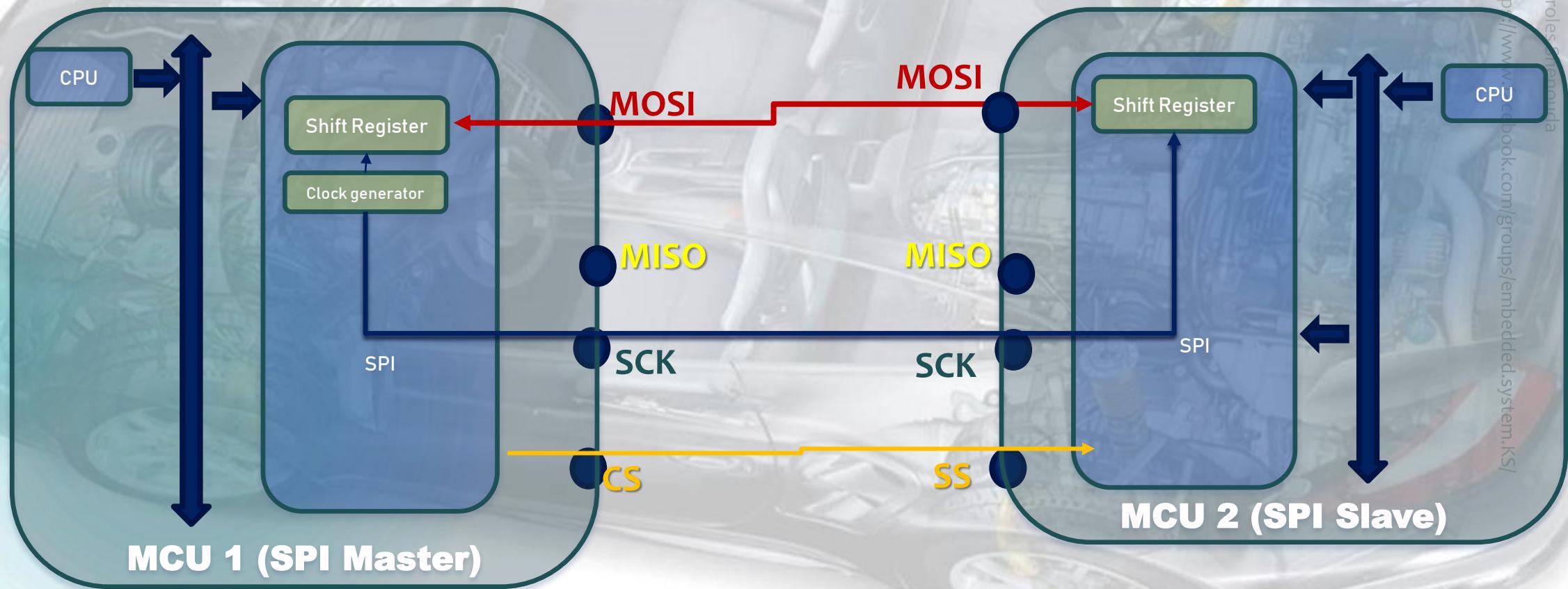
eng\_KerolesShenouda

<http://www.learn-in-depth.com/groups/embedded.system.KS/>

# Simplex Communication

## Master: Transmit only

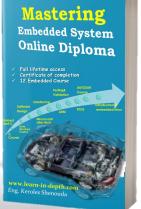
## Slave : receive only



<https://www.learn-in-depth.com/>  
<https://www.facebook.com/groups/embedded.system.KS/>



27



#LEARN\_IN\_DEPTH

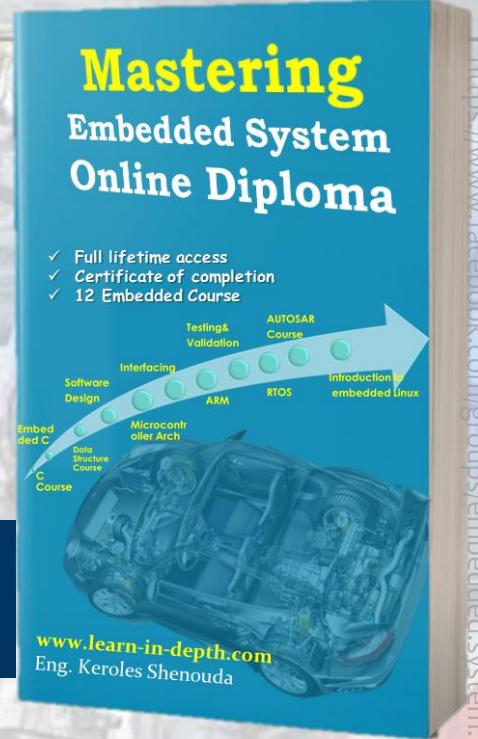
#Be\_professional\_in\_embedded\_system

eng. Keroles Shenouda

<https://www.facebook.com/groups/embedded.system.KS/>

# GOOD QUESTION

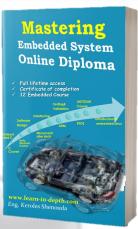
Draw this Setup  
“SPI Master Slave Setup  
with Multiple slave cascaded”



**LEARN-IN-DEPTH**  
Be professional in  
embedded system



<https://www.learn-in-depth.com/>  
<https://www.facebook.com/groups/embedded.system.KS/>



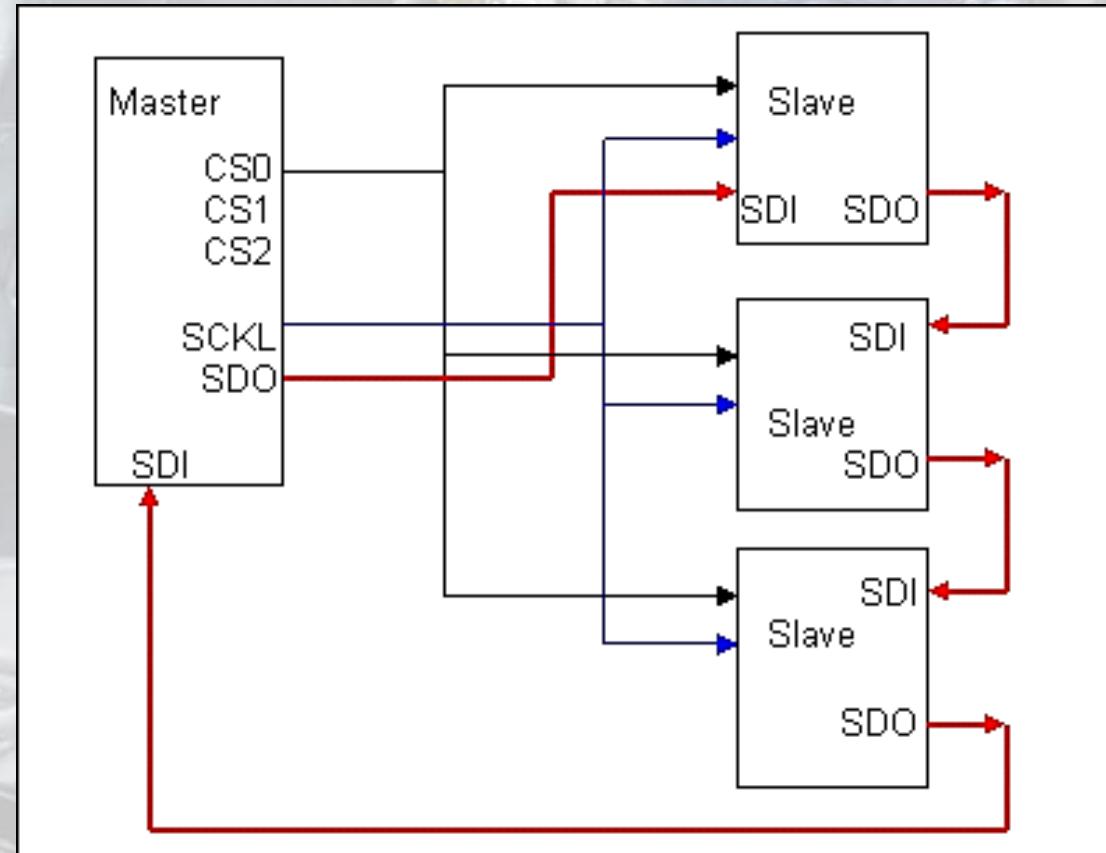
#LEARN\_IN\_DEPTH

#Be\_professional\_in\_embedded\_system

# Master Slave Setup

## Multiple slave cascaded

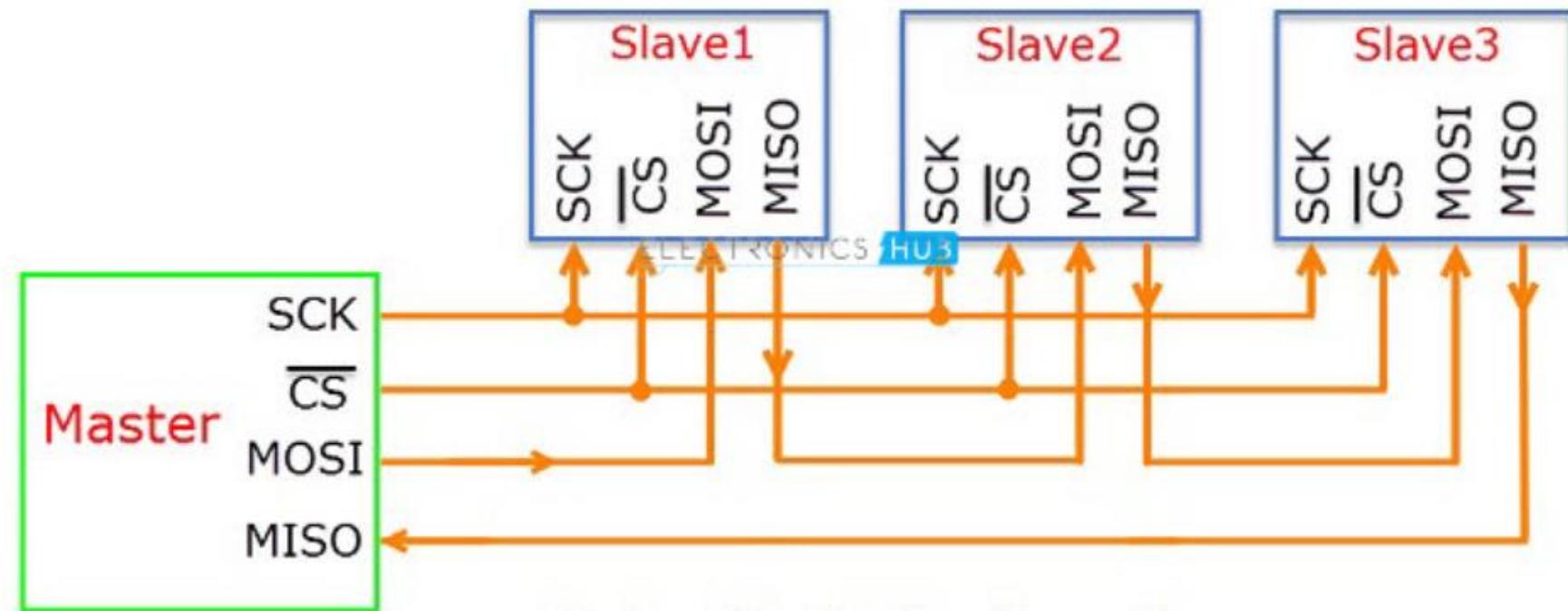
- In this example, each **slave** is **cascaded** so that the output of one slave is the input of another.
- When **cascading**, they are treated as **one slave** and connecting to **the same chip select**



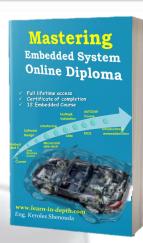
<https://www.learn-in-depth.com/>  
<https://www.facebook.com/groups/embedded.system.KS/>

28

# Daisy Chain Configuration



<https://www.learn-in-depth.com/>  
<https://www.facebook.com/groups/embedded.system.KS/>



30

#LEARN\_IN\_DEPTH

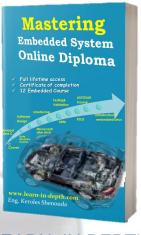
#Be\_professional\_in  
embedded\_system

eng. Keroles Shenouda

<https://www.facebook.com/groups/embedded.system.KS/>

# SPI With Flash Memories (custom protocol)

<https://www.learn-in-depth.com/>  
<https://www.facebook.com/groups/embedded.system.KS/>



31

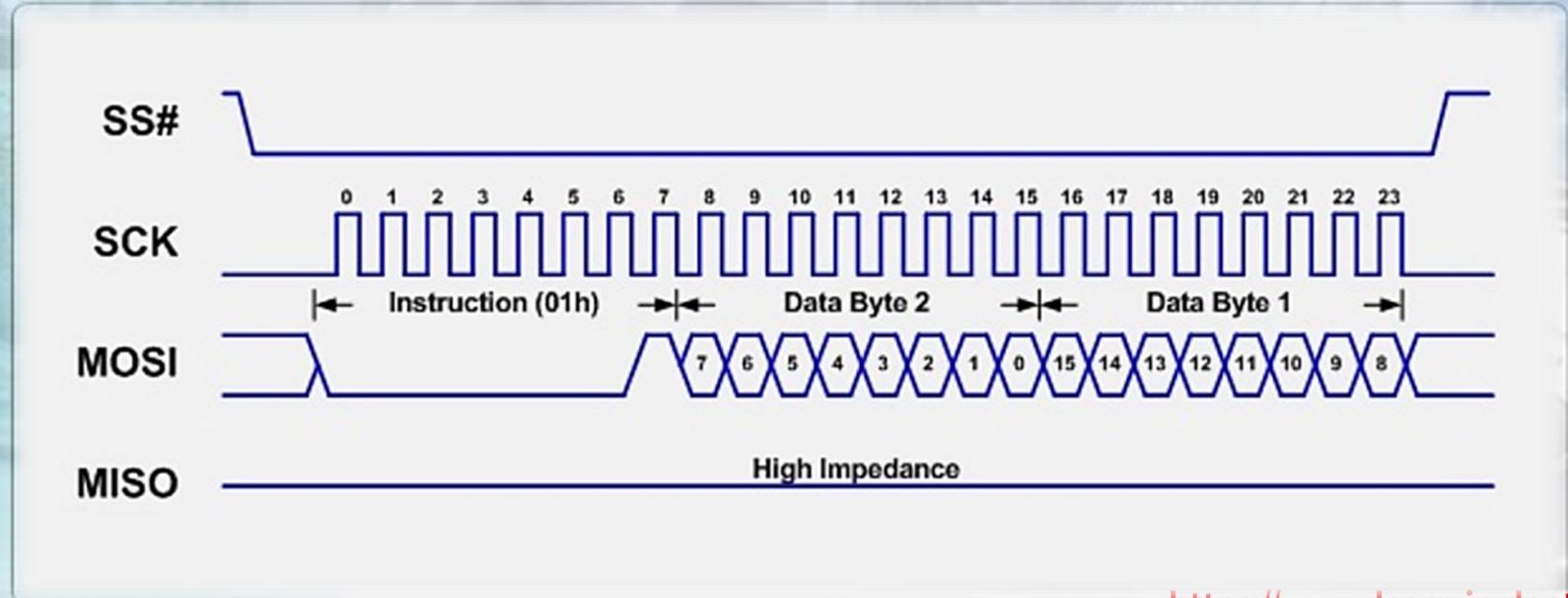
#LEARN\_IN\_DEPTH  
#Be\_professional\_in\_embedded\_system

Eng. Keroles Shenouda

<https://www.learn-in-depth.com/groups/embedded.system.KS/>

# Simple SPI Write Transaction

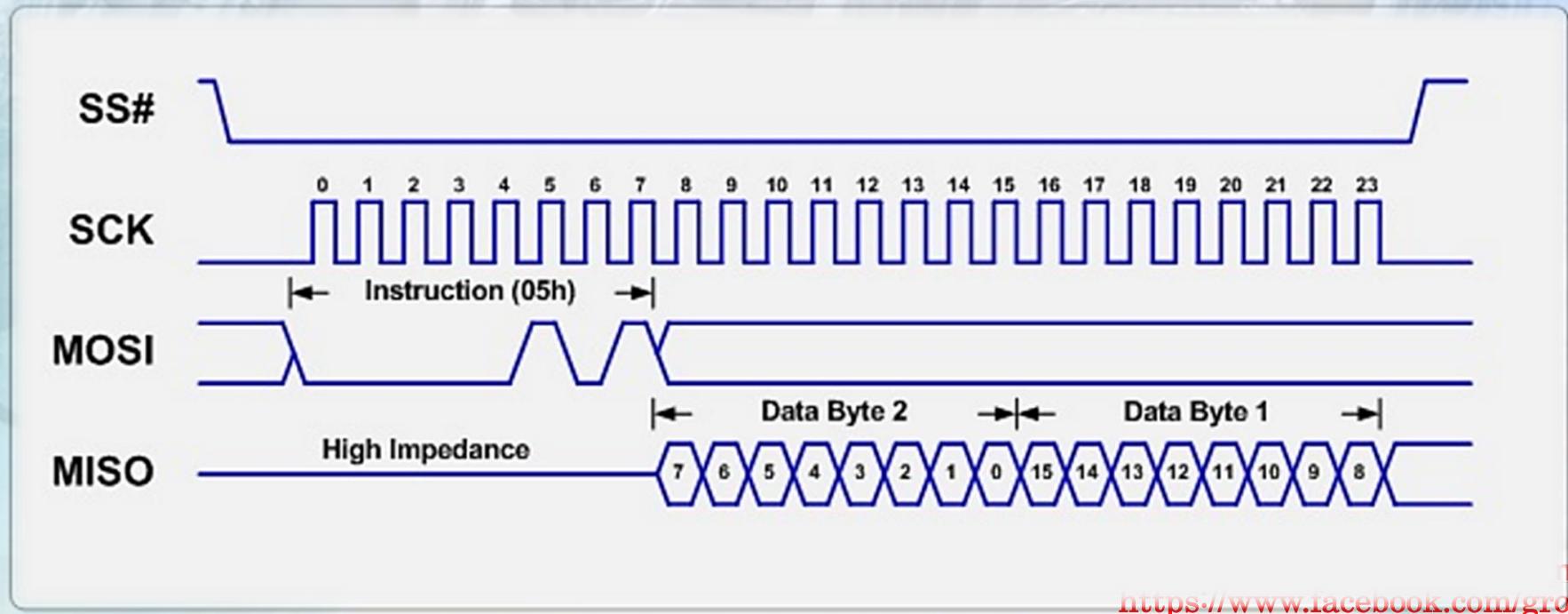
- ▶ Most **SPI flash memories** have a write **status register command** that writes one or two bytes of data.
- ▶ To write to the status register, the SPI host first enables the slave select line for the current device. The master then outputs the appropriate instruction followed by two data bytes that define the intended status register contents. Since the transaction does not need to return any data, the slave device keeps the MISO line in a high impedance state and the master masks any incoming data. Finally, slave select is de-asserted to complete the transaction.



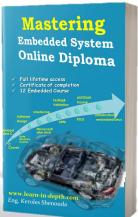
<https://www.learn-in-depth.com/>  
<https://www.facebook.com/groups/embedded.system.KS/>

# Simple SPI Read Transaction

- A status register read transaction would be similar to the write transaction, but now takes advantage of data returned from the slave. After sending the read status register instruction, the slave begins transmitting data on the MISO line at a rate of one byte per eight clock cycles. The host receives the bitstream and completes the transaction by de-asserting SS#



<https://www.facebook.com/groups/embedded.system.KS/>



33

#LEARN\_IN\_DEPTH

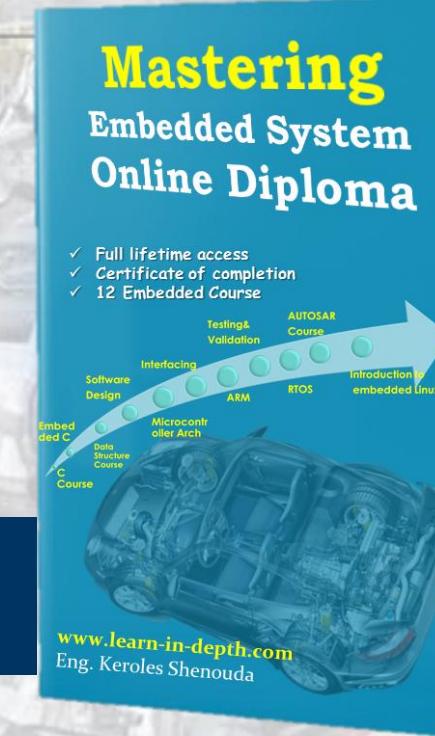
#Be\_professional\_in\_embedded\_system

eng. Keroles Shenouda

<https://www.facebook.com/groups/embedded.system.KS/>

## GOOD QUESTION

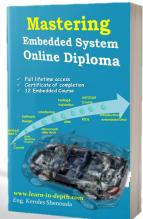
**Which type of SPI act as “half duplex” serial communication ?**



**LEARN-IN-DEPTH**  
Be professional in  
embedded system



<https://www.learn-in-depth.com/>  
<https://www.facebook.com/groups/embedded.system.KS/>



#LEARN\_IN\_DEPTH

#Be\_professional\_in  
embedded\_system

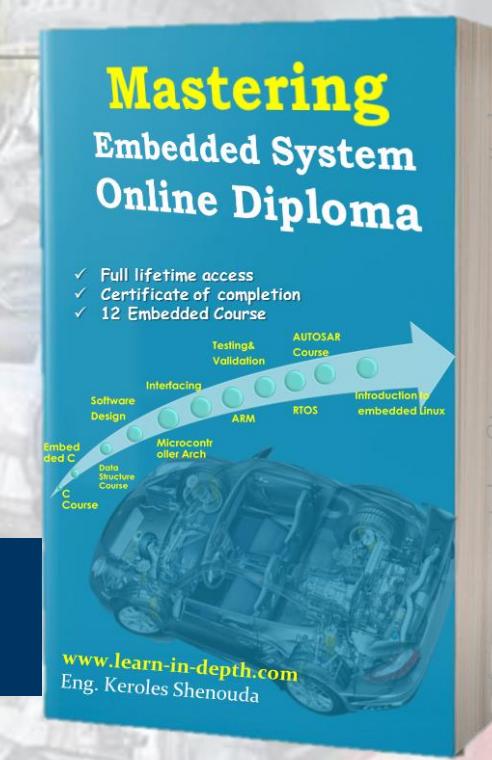
eng. Keroles Shenouda

<https://www.facebook.com/groups/embedded.system.KS/>

34



# SPI Bus 3-Wire



[www.learn-in-depth.com](http://www.learn-in-depth.com)  
Eng. Keroles Shenouda

**LEARN-IN-DEPTH**  
Be professional in  
embedded system



<https://www.learn-in-depth.com/>  
<https://www.facebook.com/groups/embedded.system.KS/>



#LEARN\_IN\_DEPTH  
#Be\_professional\_in\_embedded\_system

eng. Keroles Shenouda

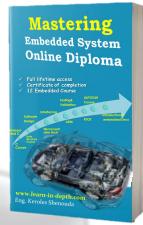
35



# SPI Bus 3-Wire and Multi-IO Configurations

- ▶ There is also a widely used standard called a **3-wire interface bus**. In a 3-wire interface bus, we have **SCLK** and **CE**, and only a single pin for data transfer.
- ▶ The SPI **4-wire bus** can become a **3-wire interface** when the **SDI** and **SDO** data pins are tied together.
- ▶ But there are some major differences between the SPI and 3-wire devices in the data transfer protocol.
- ▶ For that reason, a device must support the **3-wire protocol internally** in order to be used as a 3-wire device.
- ▶ Many devices such as the **DS1306 RTC (real-time clock)** support both SPI and 3-wire protocols.

<https://www.learn-in-depth.com/>  
<https://www.facebook.com/groups/embedded.system.KS/>



36

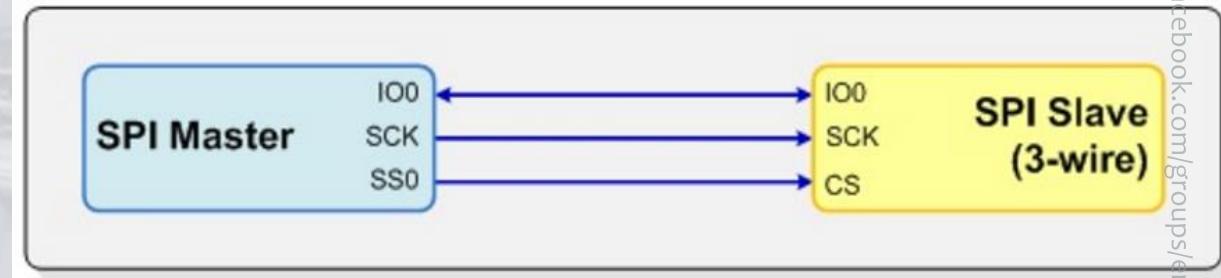
#LEARN\_IN\_DEPTH  
#Be\_professional\_in\_embedded\_system

Eng. Keroles Shenouda

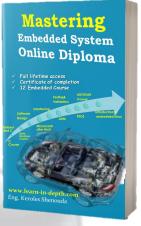
<https://www.facebook.com/groups/embedded.system.KS/>

# SPI Bus 3-Wire and Multi-IO Configurations

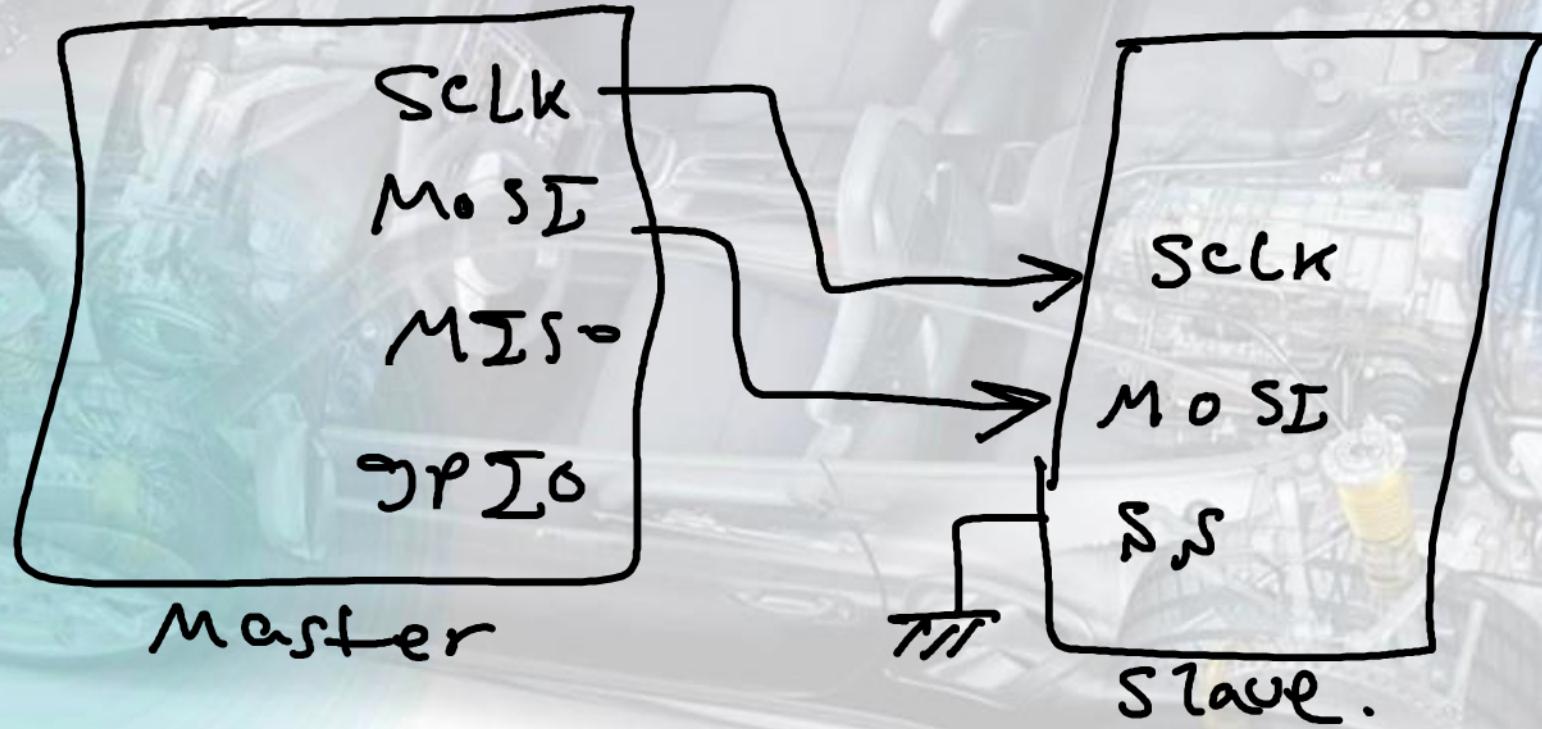
- In 3-wire mode, MOSI and MISO lines are combined to a single bidirectional data
- Transactions are half-duplex to allow for bidirectional communication. Reducing the number of data lines and operating in half-duplex mode also decreases maximum possible throughput



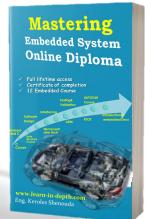
<https://www.learn-in-depth.com/>  
<https://www.facebook.com/groups/embedded.system.KS/>



# Ex if you have only one slave which is received data only :



<https://www.learn-in-depth.com/>  
<https://www.facebook.com/groups/embedded.system.KS/>



38

#LEARN\_IN\_DEPTH

#Be\_professional\_in\_embedded\_system

eng. Keroles Shenouda

<https://www.facebook.com/groups/embedded.system.KS/>

## Mastering Embedded System Online Diploma

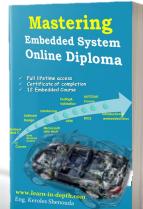
- ✓ Full lifetime access
- ✓ Certificate of completion
- ✓ 12 Embedded Course



[www.learn-in-depth.com](http://www.learn-in-depth.com)  
Eng. Keroles Shenouda

**LEARN-IN-DEPTH**  
Be professional in  
embedded system

<https://www.learn-in-depth.com/>  
<https://www.facebook.com/groups/embedded.system.KS/>



39

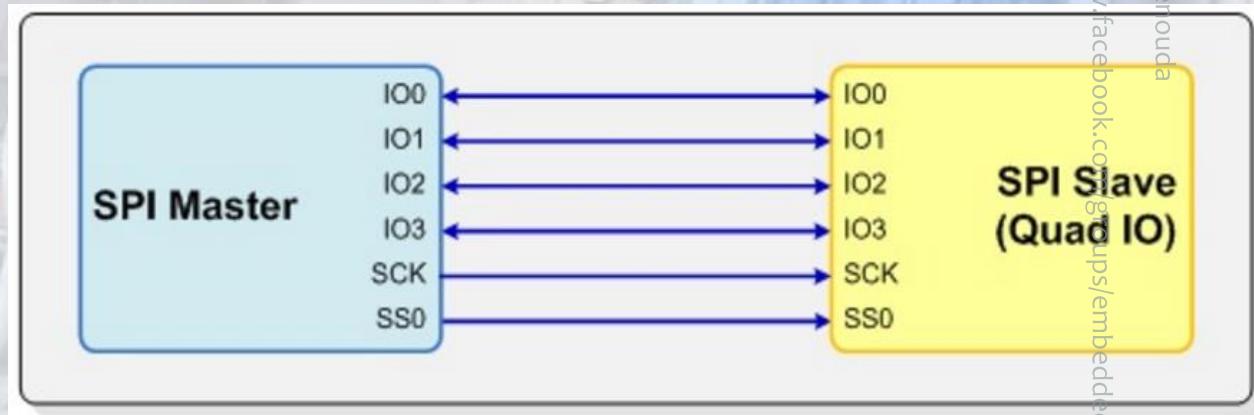
#LEARN\_IN\_DEPTH  
#Be\_professional\_in\_embedded\_system

eng. Keroles Shenouda

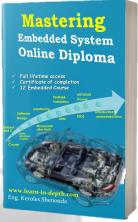
<https://www.facebook.com/groups/embedded.system.KS/>

# QUAD SPI [QSPI]

- ▶ Multi I/O variants such as dual I/O and quad I/O add additional data lines to the standard for **increased throughput**. This performance increase enables random access and direct program execution from flash memory (execute-in-place)
- ▶ A multi I/O SPI device is capable of supporting increased bandwidth or throughput from a single device.
- ▶ A **dual I/O** (two-bit data bus) interface enables transfer rates to double compared to the standard serial Flash memory devices.
- ▶ A **quad I/O** (four-bit data bus) interface improves throughput four times.



<https://www.learn-in-depth.com/>  
<https://www.facebook.com/groups/embedded.system.KS/>



40

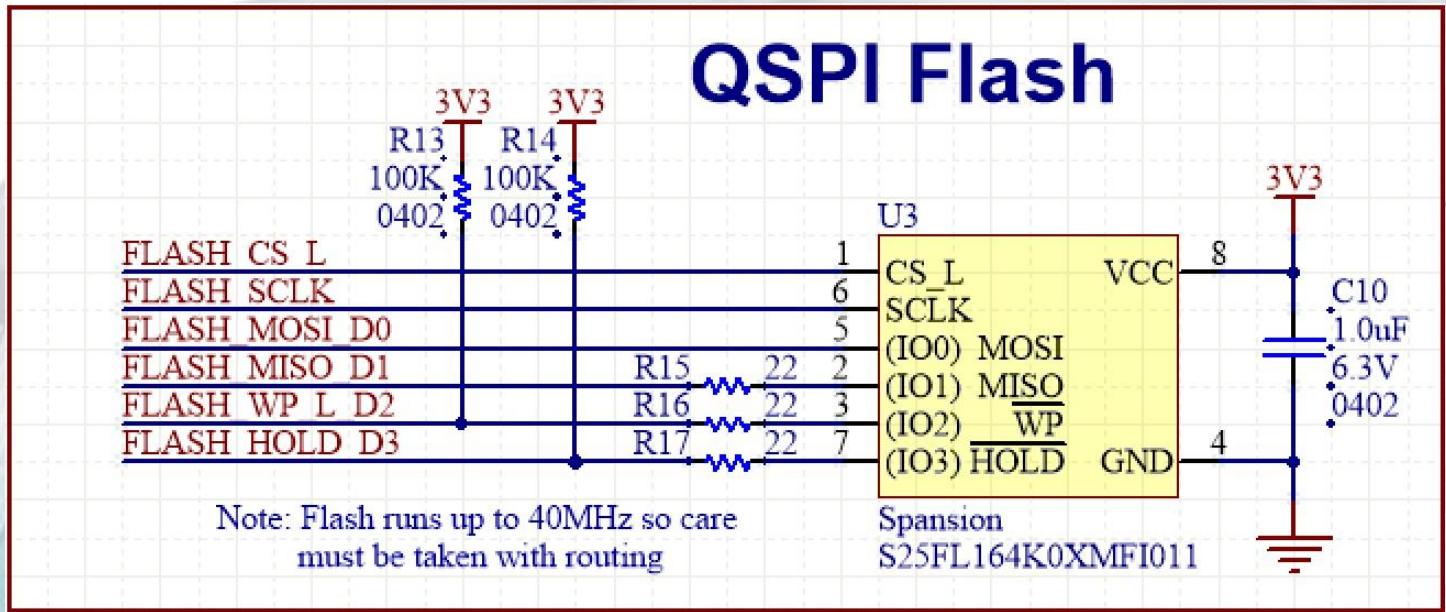
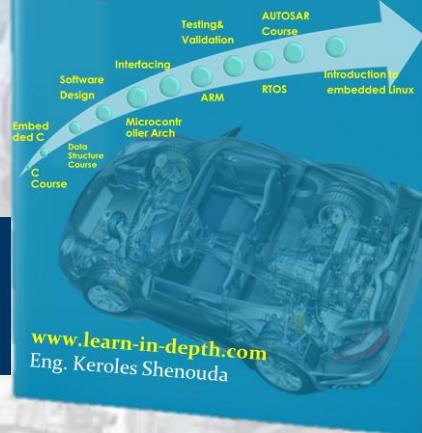
#LEARN\_IN\_DEPTH

#Be\_professional\_in  
embedded\_system

<https://www.facebook.com/groups/embedded.system.KS/>

## Mastering Embedded System Online Diploma

- ✓ Full lifetime access
- ✓ Certificate of completion
- ✓ 12 Embedded Course



Example: QSPI connected to  
Spansion Flash

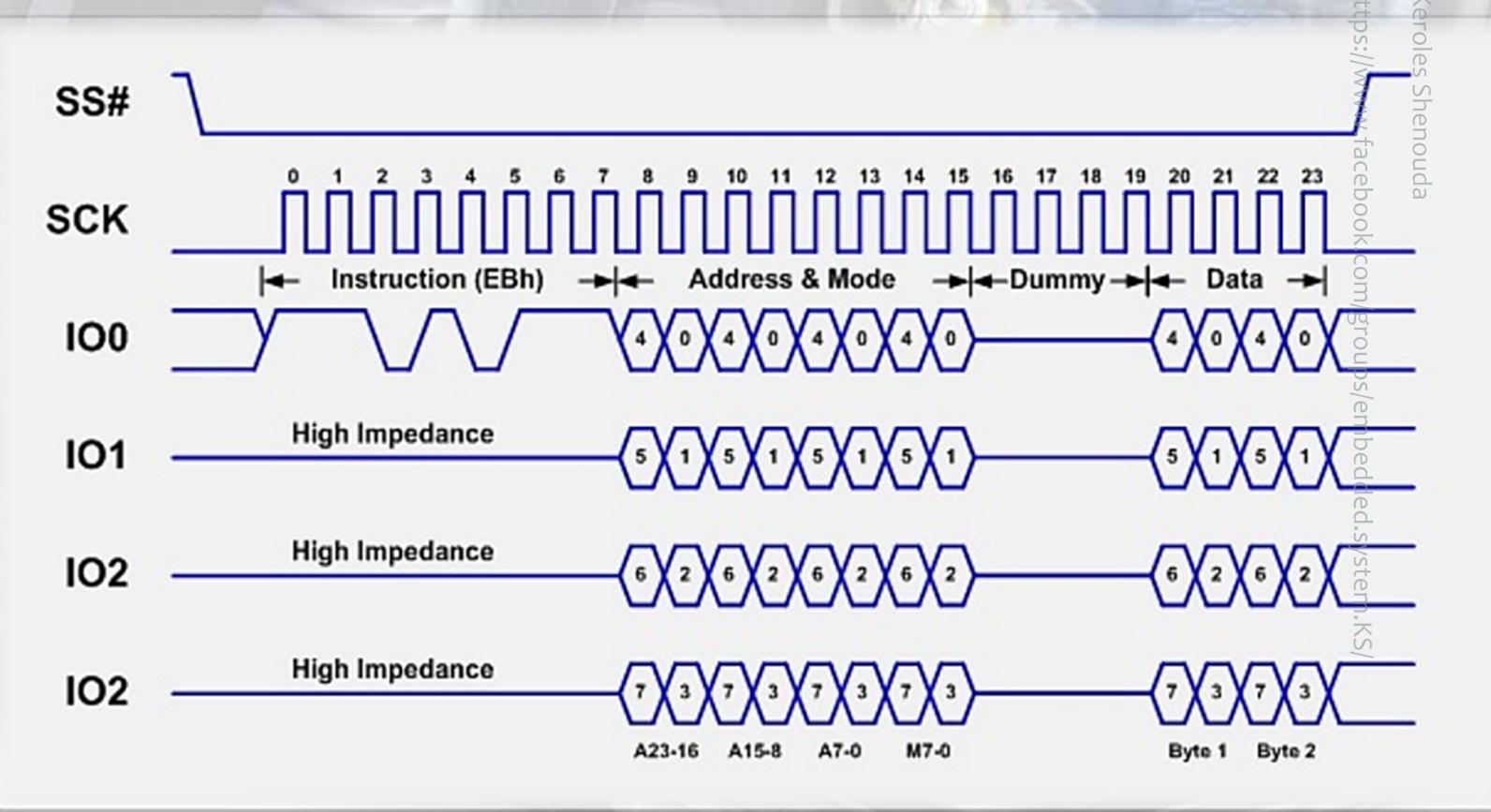
LEARN-IN-DEPTH  
Be professional in  
embedded system



<https://www.learn-in-depth.com/>  
<https://www.facebook.com/groups/embedded.system.KS/>

# Example: Quad mode fast read sequence for Spansion S25FL016K or equivalent

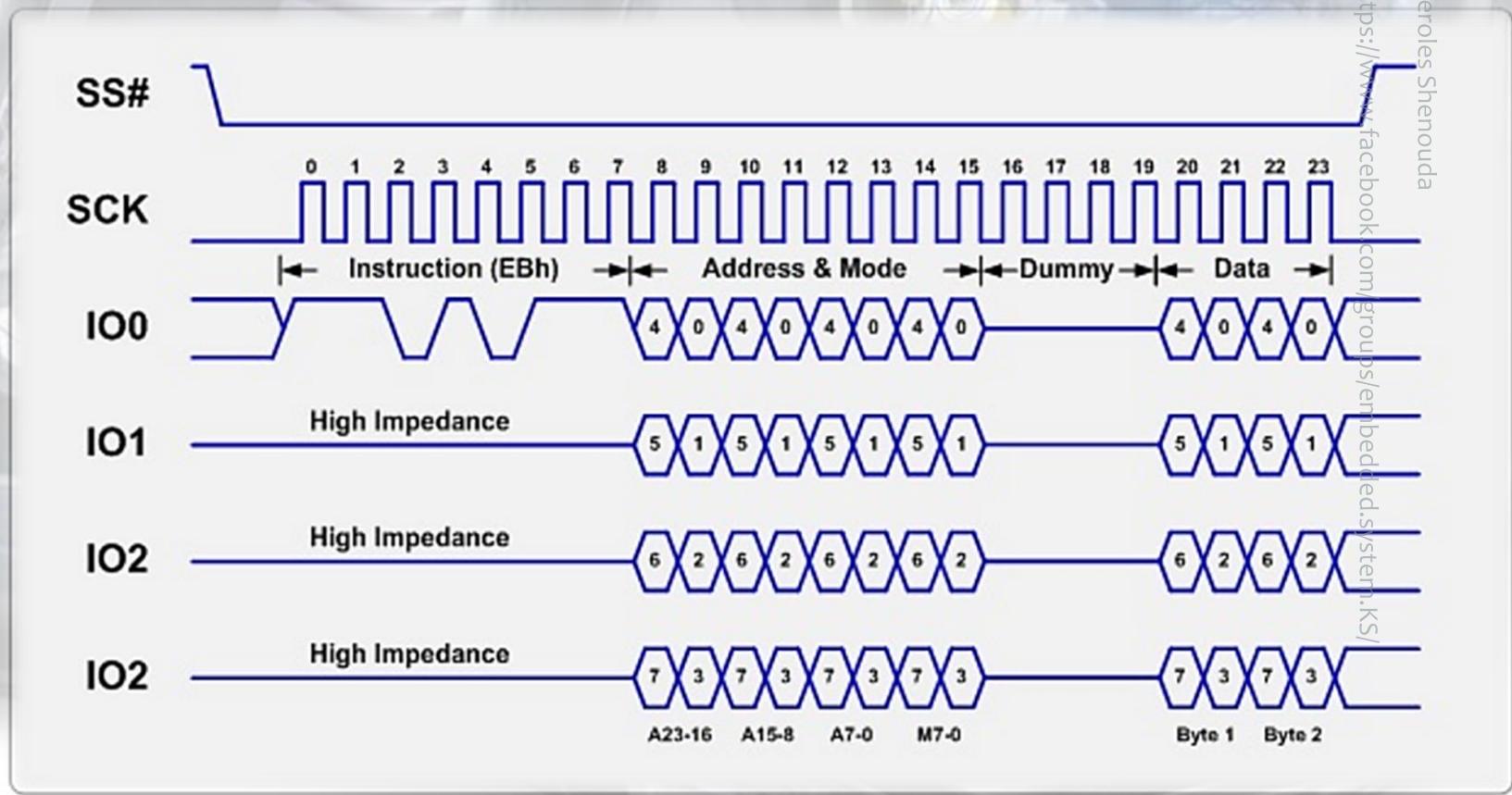
Quad IO is gaining popularity with **flash memories** for its increased performance. Instead of using a single output and single input interface, Quad IO utilizes **4 separate half-duplex data** lines **for both transmitting and receiving** data for up to four times the performance of standard 4-wire SPI.



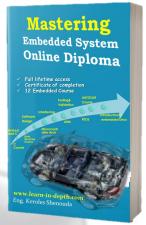
<https://www.learn-in-depth.com/>  
<https://www.facebook.com/groups/embedded.system.KS/>

# Example: Quad mode fast read sequence for Spansion S25FL016K or equivalent

This example read command for a Spansion S25FL016K serial NOR flash device. To read from the device, a fast read command (EBh) is first sent by the master on the first IO line while all others are tristated. Next, the host sends the address; since the interface now has 4 bidirectional data lines, it can utilize these to send a complete 24-bit address along with 8 mode bits in just 8 clock cycles. The address is then followed with 2 dummy bytes (4 clock cycles) to allow the device additional time to set up the initial address.



<https://www.learn-in-depth.com/>  
<https://www.facebook.com/groups/embedded.system.KS/>



43

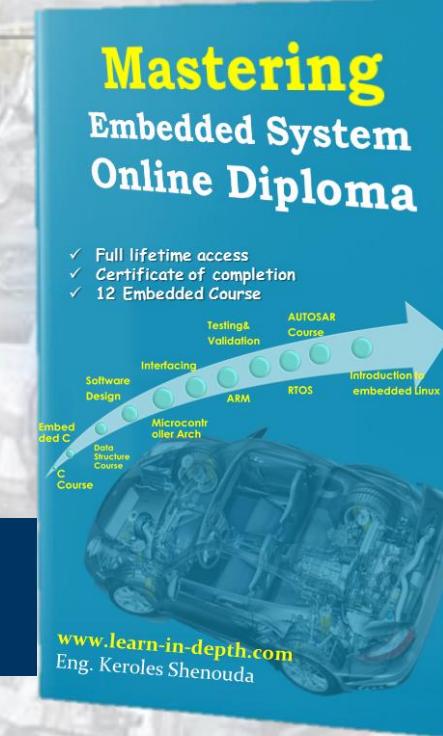
#LEARN\_IN\_DEPTH

#Be\_professional\_in  
embedded\_system

eng. Keroles Shenouda

<https://www.facebook.com/groups/embedded.system.KS/>

# Conclusion



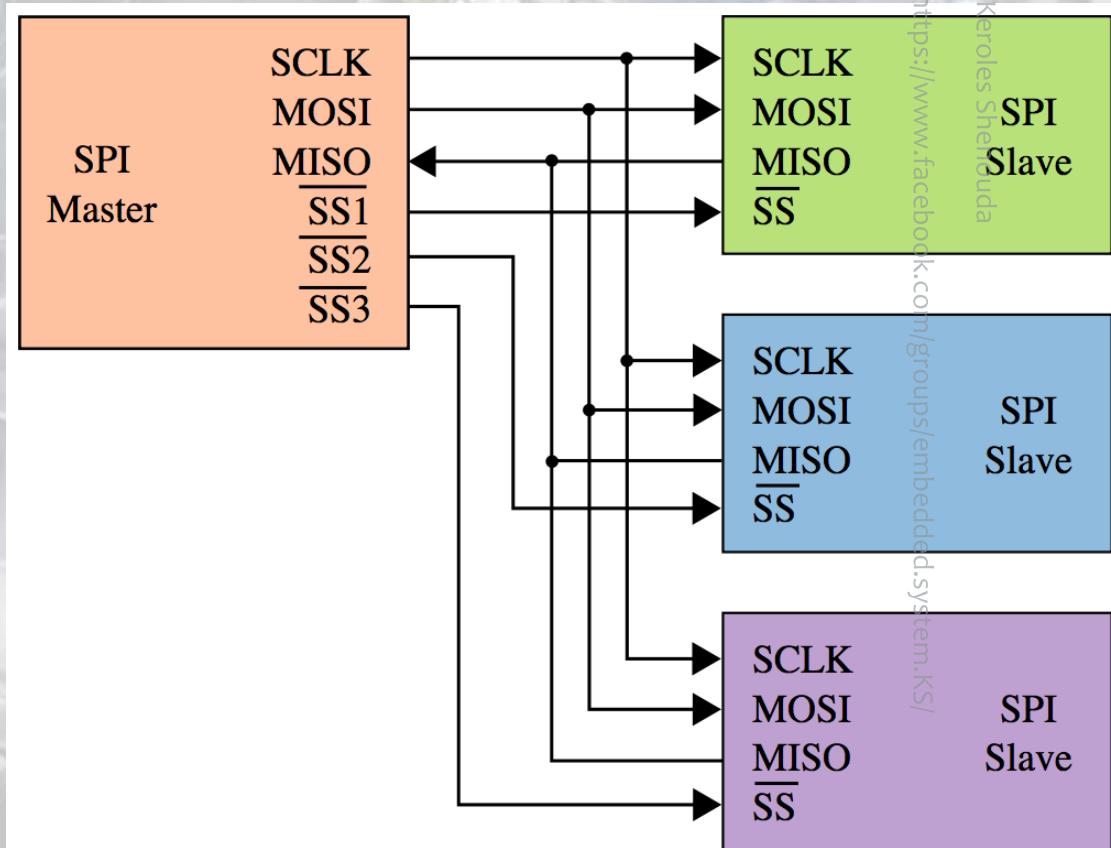
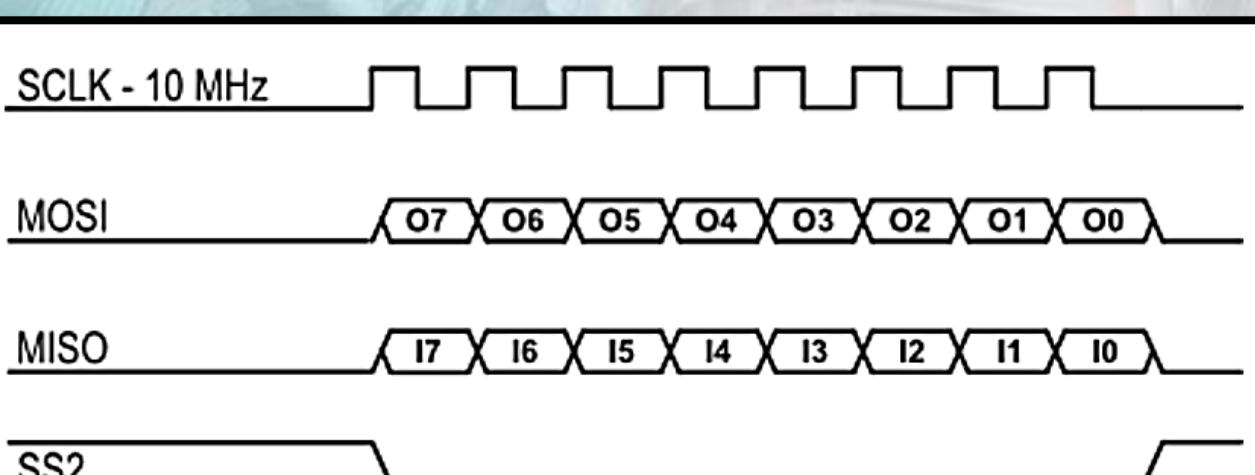
**LEARN-IN-DEPTH**  
Be professional in  
embedded system



<https://www.learn-in-depth.com/>  
<https://www.facebook.com/groups/embedded.system.KS/>

# 8-bit standard SPI access

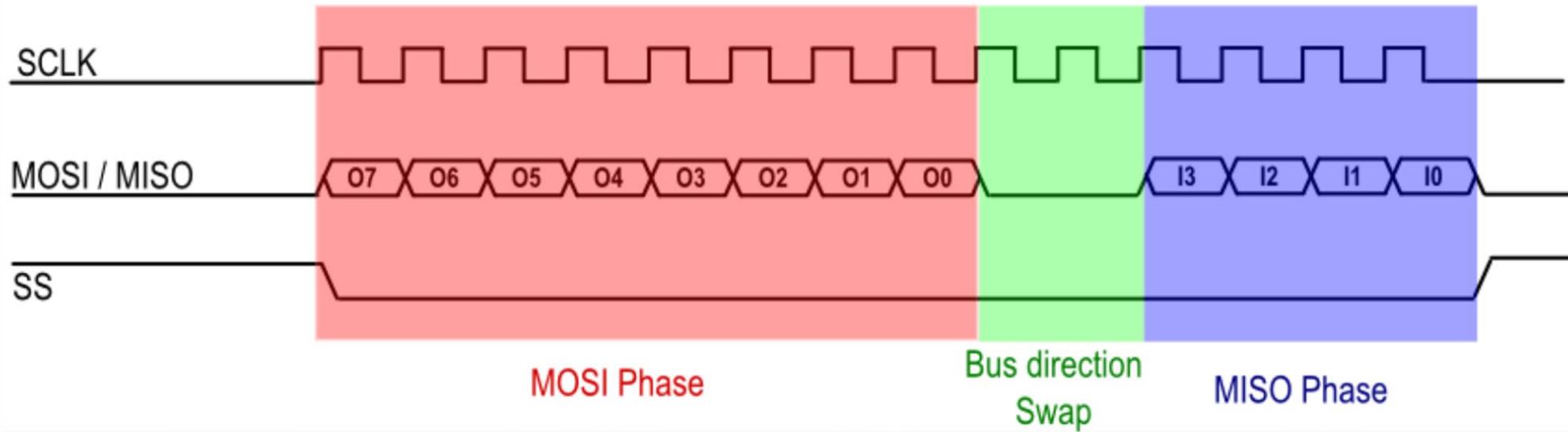
- SPI (Serial Peripheral Interface):  
**4 wires protocols** using **MOSI**, **MISO**, **SCLK** and **SS**.



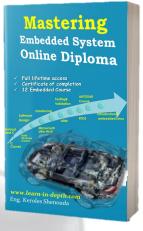
<https://www.learn-in-depth.com/>  
<https://www.facebook.com/groups/embedded.system.KS/>

# SPI-3

- variant of the SPI protocol where MOSI and MISO are merged as a single bi-directional data line, with bus turn-around phases.



<https://www.learn-in-depth.com/>  
<https://www.facebook.com/groups/embedded.system.KS/>



46

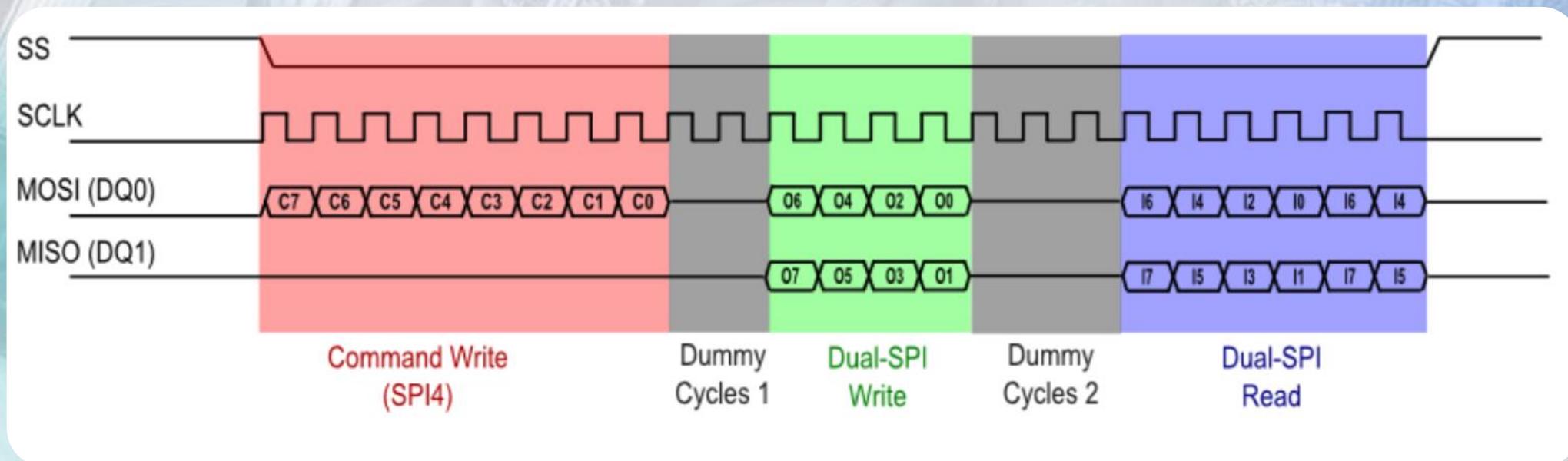
#LEARN\_IN\_DEPTH  
#Be\_professional\_in\_embedded\_system

eng. Keroles Shenouda

<https://www.facebook.com/groups/embedded.system.KS/>

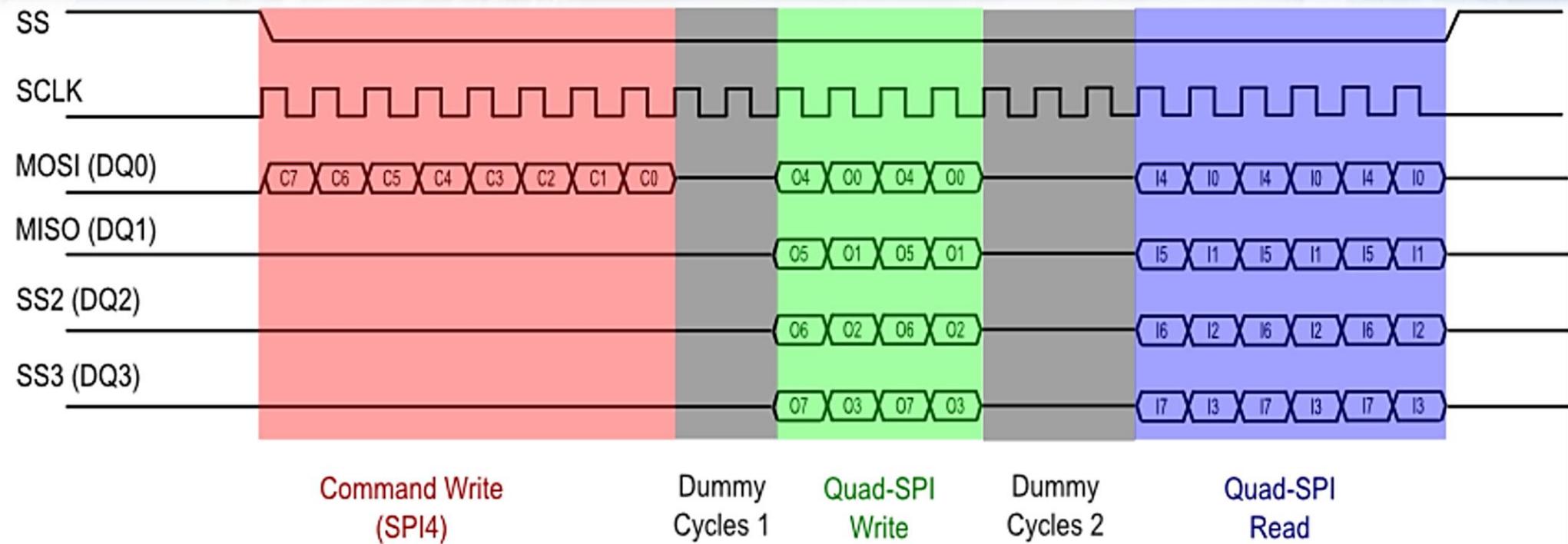
# Dual-SPI

- ▶ variant of the SPI protocol that includes modes in which 2 data lines are used for conveying data.



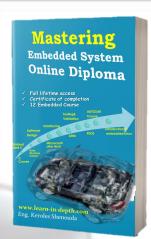
# Quad-SPI

- variant of the SPI protocol that includes modes in which 4 data lines are used for conveying data.



<https://www.learn-in-depth.com/>

<https://www.facebook.com/groups/embedded.system.KS/>



48

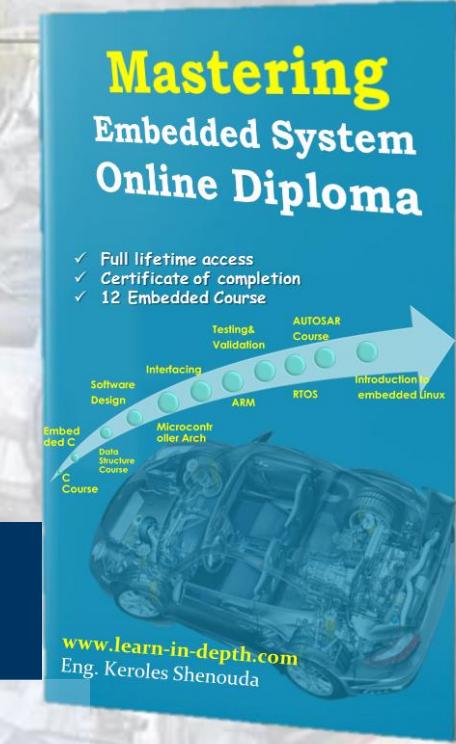
#LEARN\_IN\_DEPTH

#Be\_professional\_in\_embedded\_system

eng. Keroles Shenouda

<https://www.facebook.com/groups/embedded.system.KS/>

## SPI/I2C/UART/CAN/LIN/USB comparison



**LEARN-IN-DEPTH**  
Be professional in  
**embedded system**

<https://www.learn-in-depth.com/>  
<https://www.facebook.com/groups/embedded.system.KS/>

Protocol	Synch/ Asynch.	I/O Pins	Max Distance	Avg. Speed	Max. possible Number of devices in a Bus
UART	Asynchronous Serial Point to point	2	15m	115.2kbps	2 point to point
SPI	Synchronous	3 / 4+	0.1m	4Mbps	Virtually unlimited
I <sub>2</sub> C	Synchronous	2	0.5m	1Mbps	127
LIN	Asynchronous two wire serial communication similar to uart	2	40m	20kbps	several
CAN	Industrial differential two wire communication support more than one master	2	5km	1Mbps	several
USB	Asynchronous full duplex	2	<5m	480Mbps	127



49

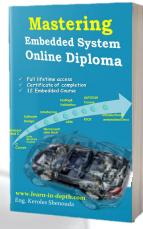
#LEARN\_IN\_DEPTH

#Be\_professional\_in  
embedded\_system

eng. Keroles Shenouda

<https://www.facebook.com/groups/embedded.system.KS/>

<https://www.learn-in-depth.com/>  
<https://www.facebook.com/groups/embedded.system.KS/>



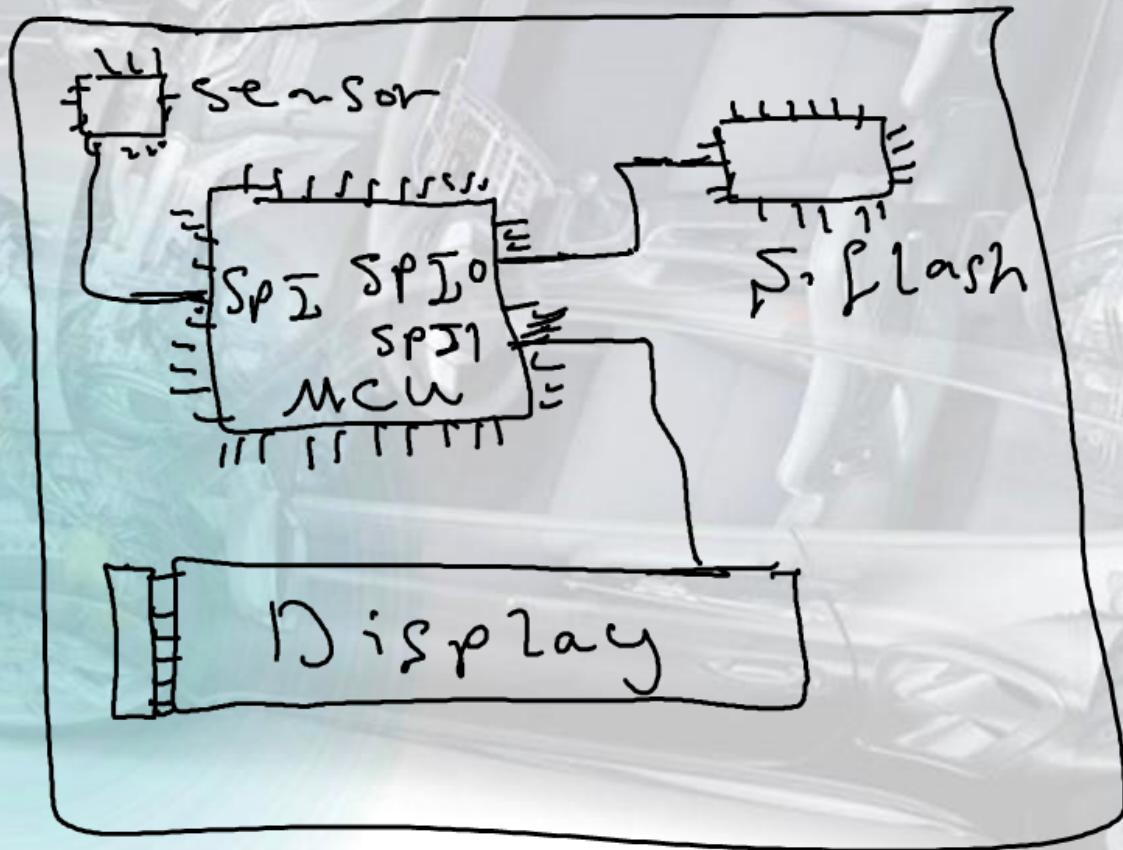
50

#LEARN\_IN\_DEPTH  
#Be\_professional\_in\_embedded\_system

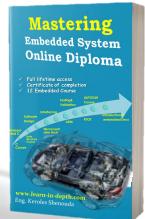
eng. Keroles Shenouda

<https://www.facebook.com/groups/embedded.system.KS/>

# Example for SPI Connection



<https://www.learn-in-depth.com/>  
<https://www.facebook.com/groups/embedded.system.KS/>



52

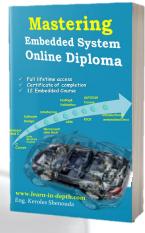
#LEARN\_IN\_DEPTH

#Be\_professional\_in\_embedded\_system

# References

- ▶ <https://docs.google.com/viewer?a=v&pid=sites&srcid=ZmtlLnVObS5teXxyaWR6dWFuLXMtd2Vic2I0ZXxneDo2ODU0NzIKM2JkOTg4MjRk>
- ▶ <http://www.avrprojects.net/index.php/avr-projects/sensors/38-humidity-and-temperature-sensor-dht11?showall=&start=1>
- ▶ <http://www.cse.wustl.edu/~lu/cse467s/slides/dsp.pdf>
- ▶ <http://www.avr-tutorials.com/>
- ▶ Microprocessor: ATmega32 (SEE3223-10)  
<http://ridzuan.fke.utm.my/microprocessor-atmega32-see3223-10>
- ▶ <http://circuitdigest.com/article/what-is-the-difference-between-microprocessor-and-microcontroller>
- ▶ [http://cs4hs.cs.pub.ro/wiki/roboticsisfun/chapter2/ch2\\_7\\_programming\\_a\\_microcontroller](http://cs4hs.cs.pub.ro/wiki/roboticsisfun/chapter2/ch2_7_programming_a_microcontroller)
- ▶ Embedded Systems with ARM Cortex-M Microcontrollers in Assembly Language and C Dr. Yifeng Zhu Third edition June 2018

<https://www.learn-in-depth.com/>  
<https://www.facebook.com/groups/embedded.system.KS/>



53

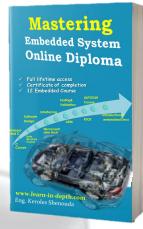
#LEARN\_IN\_DEPTH

#Be\_professional\_in\_embedded\_system

# References

- ▶ <http://techdifferences.com/difference-between-interrupt-and-polling-in-os.html>
- ▶ [http://www.bogotobogo.com/Embedded/hardware\\_interrupt\\_software\\_interrupt\\_latency\\_irq\\_vs\\_fiq.php](http://www.bogotobogo.com/Embedded/hardware_interrupt_software_interrupt_latency_irq_vs_fiq.php)
- ▶ Preventing Interrupt Overload Presented by Jiyong Park Seoul National University, Korea 2005. 2. 22. John Regehr, Usit Duogsaa, School of Computing, University.
- ▶ First Steps Embedded Systems Byte Craft Limited reference
- ▶ COMPUTER ORGANIZATION AND ARCHITECTURE DESIGNING FOR PERFORMANCE EIGHTH EDITION William Stallings
- ▶ Getting Started with the Tiva™ TM4C123G LaunchPad Workshop

<https://www.learn-in-depth.com/>  
<https://www.facebook.com/groups/embedded.system.KS/>



54

#LEARN\_IN\_DEPTH

#Be\_professional\_in\_embedded\_system

# References

- ▶ Tiva™ TM4C123GH6PM Microcontroller DATA SHEET
- ▶ Interrupts and Exceptions COMS W6998 Spring 2010
- ▶ THE AVR MICROCONTROLLER. AND EMBEDDED SYSTEMS Using Assembly and C. Muhammad Ali Mazidi.
- ▶ <http://embedded-lab.com/blog/tinkering-ti-msp430f5529/27/>

<https://www.learn-in-depth.com/>  
<https://www.facebook.com/groups/embedded.system.KS/>



55

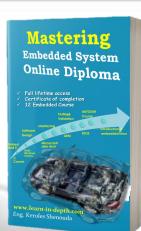
#LEARN\_IN\_DEPTH

#Be\_professional\_in\_embedded\_system

# References

- ▶ <https://docs.google.com/viewer?a=v&pid=sites&srcid=ZmtILnV0bS5teXxyaWR6dWFuLXMfd2Vic2I0ZXxneDo2ODU0Nzlkm2JkOTg4MjRk>
- ▶ <http://www.avrprojects.net/index.php/avr-projects/sensors/38-humidity-and-temperature-sensor-dht11?showall=&start=1>
- ▶ <http://www.cse.wustl.edu/~lu/cse467s/slides/dsp.pdf>
- ▶ <http://www.avr-tutorials.com/>
- ▶ Microprocessor: ATmega32 (SEE3223-10)  
<http://ridzuan.fke.utm.my/microprocessor-atmega32-see3223-10>
- ▶ <http://circuitdigest.com/article/what-is-the-difference-between-microprocessor-and-microcontroller>
- ▶ AVR Microcontroller and Embedded Systems: Using Assembly and C (Pearson Custom Electronics Technology) 1st Edition  
<https://www.amazon.com/AVR-Microcontroller-Embedded-Systems-Electronics/dp/0138003319>

<https://www.learn-in-depth.com/>  
<https://www.facebook.com/groups/embedded.system.KS/>



56

#LEARN\_IN\_DEPTH

#Be\_professional\_in  
embedded\_system

Thank You

<https://www.learn-in-depth.com/>  
<https://www.facebook.com/groups/embedded.system.KS/>