

The Serial Peripheral Interface (SPI)

Corrado Santoro

ARSLAB - Autonomous and Robotic Systems Laboratory

Dipartimento di Matematica e Informatica - Università di Catania, Italy

santoro@dmi.unict.it

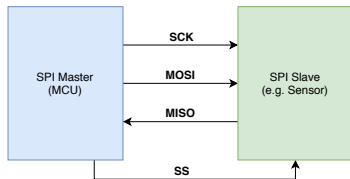


L.S.M. 1 Course

What is SPI?

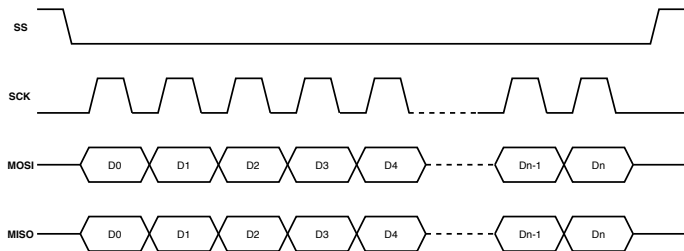
- The **SPI—Serial Peripheral Interface** is a **communication hardware** designed to interconnect *integrated circuits* to a MCU belonging to the same board.
- It has been introduced by Motorola in the '80s.
- Now, it is a standard interface supported by any MCU and many devices like sensors, memories (EEPROM, SDCards, etc.), power drivers, etc.

SPI: Philosophy and Connections



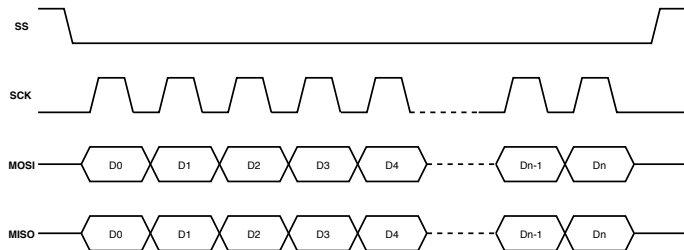
- SPI is a **Master-Slave interface**:
 - **Master**, is the “head” of the communication and is - in general - a MCU;
 - **Slave**, all the other devices which “respond” to master solicitations.
- It has (at least) **4 point-to-point unidirectional wires**:
 - **SCK**—*Master to Slave*, the serial clock, generated by the Master;
 - **MOSI**—*Master to Slave, Master-Out-Slave-In*, the serial data transmitted by the Master and sent to a Slave;
 - **MISO**—*Slave to Master, Master-In-Slave-Out*, the serial data transmitted by a Slave and sent to the Master;
 - **SS**—*Slave to Master, Slave Select*, a logic signal that is used to “wake-up” and select a slave device.

SPI: Signals and Timing



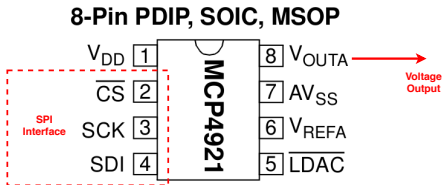
- A **SPI Transaction** is always started by the Master by setting the **SS Line** to **Logic “0”**
- The Master then starts to generating the **clock signal** in the **SCK Line**
- At each clock tick:
 - The Master puts a data-bit in the **MOSI Line**
 - The Slave puts a data-bit in the **MISO Line**
- When the transaction terminates, the Master sets the **SS Line** to **Logic “1”**

Meaning of Data Bits



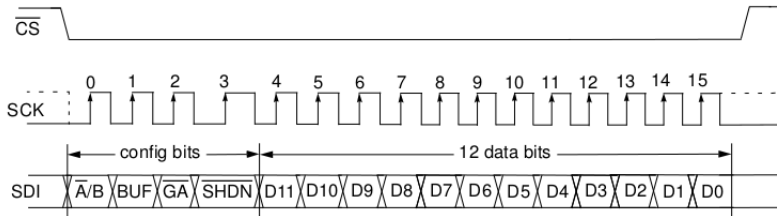
- Conceptually, communication is *packet-based*
- A **packet** is a *set of bits*, whose meaning is defined by the slave device addressed
- The number of bits of a packet may be multiple of 8, but this is not a mandatory requirement

Example: the MCP4921/MCP4922



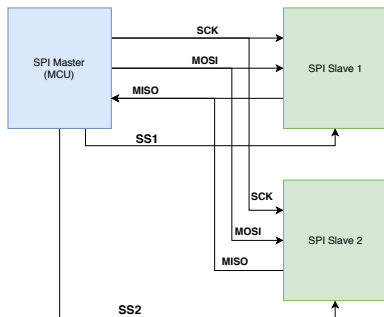
- The MCP4921/MCP4922 chip are **Digital-to-Analog Converters (DAC)**
- A **DAC** is a circuit that receives a digital data and is able to generate a **voltage signal** proportional to the received data
- The MCP4921 has only one output channel, the MCP4922 has two output channels
- The SPI interface of MCP4921 is made of the following signals:
 - **CS**, Chip Select, alternative name for **SS**
 - **SCK**, the Serial Clock (from master)
 - **SDI**, Serial Data Input, alternative name for **MOSI**

Example: the data packet of MCP4921/MCP4922



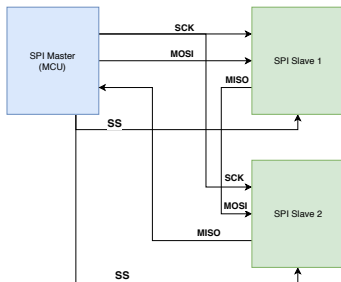
- The SPI packet, sent by the Master, is made of 16 bit:
 - The first 4 bits represent a configuration of the chip
 - $\overline{A/B}$, channel select (for the MCP4922)
 - \overline{GA} , gain (proportionality factor) control
 - \overline{SHDN} , shutdown bit (turns off the output)
 - The other 12 bits represent the DAC value
- The chip does not generate data, so it does not have a **MISO** line

Multiple Slaves: Parallel Connection



- When more than a single SPI device is present, *parallel connection* may be employed
- **SCK, MOSI and MISO** are wired together
- A **SS line per addressed device** is present
- When a specific slave needs to be addressed, the relevant **SS line** is activated

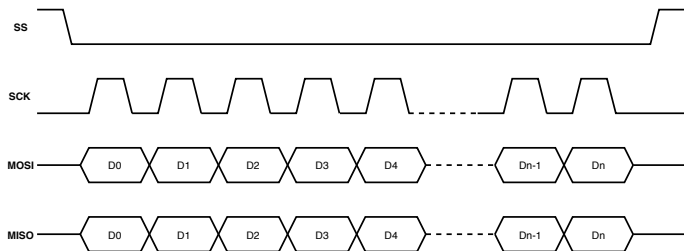
Multiple Slaves: Daisy-Chain Connection



- The *daisy-chain* connection implies a MOSI/MISO cascade connection of all SPI slave devices
- **MOSI and MISO** lines are connected in sequence
- A **single SS line for all devices** is present
- The devices must support this kind of connection
- Since all slaves are involved in each transaction, the SPI packet must contain the **address** of the device to interact with

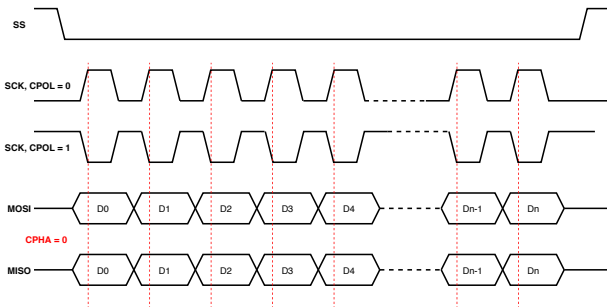
Clock Polarity and Phase

- The clock of the SPI interface can be programmed in order to set its behaviour
- There are two kind of setting:
 - **CPOL = Clock Polarity:** specifies the initial logic state of the clock
 - **CPHA = Clock Phase:** specifies the clock transition used to sample data (thus data changes in the other transition)



Clock Polarity and Phase

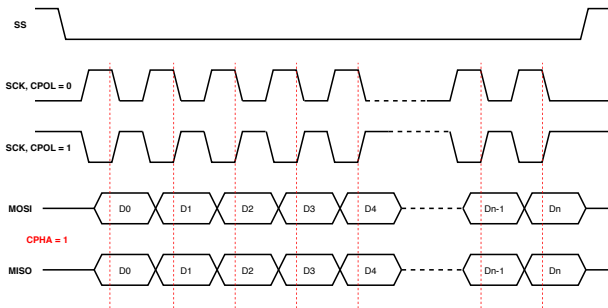
- The clock of the SPI interface can be programmed in order to set its behaviour
- There are two kind of setting:
 - **CPOL = Clock Polarity:** specifies the initial logic state of the clock
 - **CPHA = Clock Phase:** specifies the clock transition used to sample data (thus data changes in the other transition)



CPHA = 0, data is sampled in the first edge of the clock

Clock Polarity and Phase

- The clock of the SPI interface can be programmed in order to set its behaviour
- There are two kind of setting:
 - **CPOL = Clock Polarity:** specifies the initial logic state of the clock
 - **CPHA = Clock Phase:** specifies the clock transition used to sample data (thus data changes in the other transition)



CPHA = 1, data is sampled in the second edge of the clock

SPI and I²C: Comparison

- **Wiring:**

- I²C has only **two lines**
- SPI has at least **four lines**

- **Electrical Aspects:**

- I²C is a **bus** and has also a **bidirectional line**, therefore is *more critical* (from the electrical point of view) with respect to SPI
- SPI has **point-to-point unidirectional lines**

- **Speed:**

- Due to electrical characteristics, I²C is designed for a max speed of 400 *Kbps*
- SPI has **no theoretical speed limit**, even if the “rule-of-thumb” suggests 1 *Mbps*

SPI and I²C: Comparison

- **Protocol and Addressing:**

- I²C specifies a standard for device addressing and functionalities are mapped to device registers
- SPI does not define a standard data frame, each device has its own data format

- **Number of supported devices:**

- I²C is a bus and thus it has no theoretical limit on the number of devices that can be interconnected
- Limits on SPI are defined by the number of SS wires (in case of parallel connection) or latency (in case of daisy-chain connection)

The Serial Peripheral Interface (SPI)

Corrado Santoro

ARSLAB - Autonomous and Robotic Systems Laboratory

Dipartimento di Matematica e Informatica - Università di Catania, Italy

santoro@dmi.unict.it



L.S.M. 1 Course