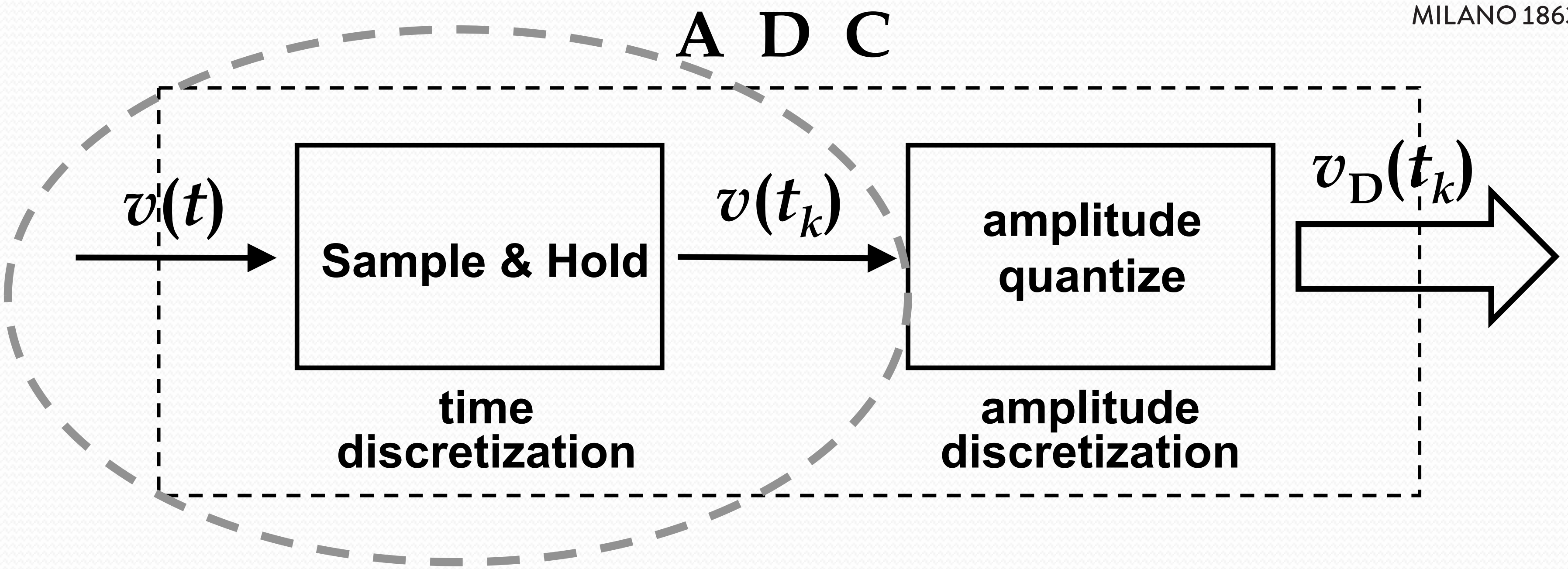
SAMPLING

DAQ BOARDS

PROTOCOLS

Ideal/Real Sampling



The sampled signal $v(kT_c)=x_c(t)$
is acquired ideally by taking the samples
in a infinitesimal lenght of time but **actually** it is needed a **finite lenght of time ($T_w \neq 0$)** to acquire from $v(t)$
the sampled signal $v(t_k)$ and to quantize

Input signal and sampling

POLITECNICO
MILANO 1863

A **voltage signal $v(t)=x(t)$** is real and conttinuos

Suppose you to work with a spectrum of signal "limited”:
transform $X(f)$ t.c. **$X(f)\equiv 0$ per $|f|>f_{max}$**

(if the signal has not bandwidth limited to f_{max},
you can use a low-pass filter)

In an **ideal sampling** the signal is multiplied by a train of **Dirac delta function $h(t)=\delta(t)$**

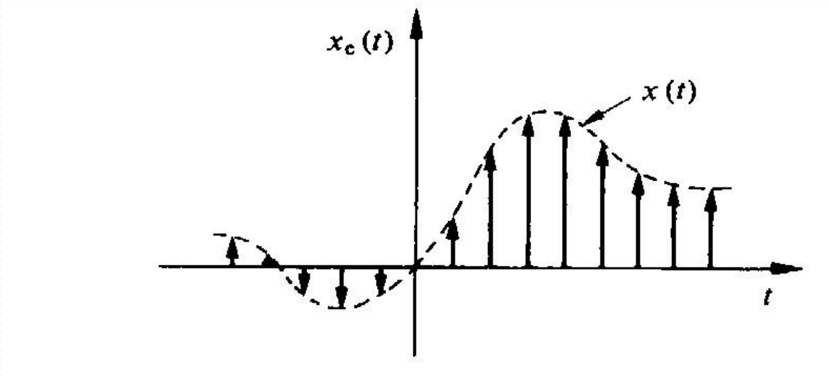
In a **real sampling** the signal is multiplied by a train of **rectangles $h(t)=\mathrm{rect}(t/T_w)$**
in which the single rectangle has finite duration T_w

Ideal sampling

T_c and f_c are period and sampling frequency

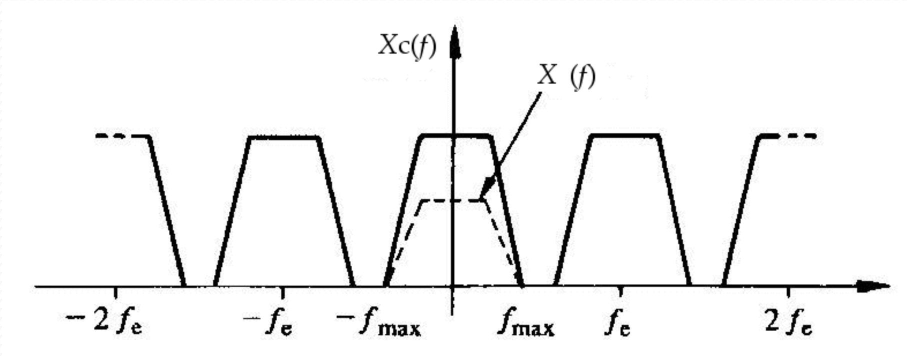
sampled signal
(time domain)

$$x_c(t) = x(t)\sum_{k=-\infty}^{+\infty}\delta(t - kT_c)$$



sampled signal
(spectral domain)

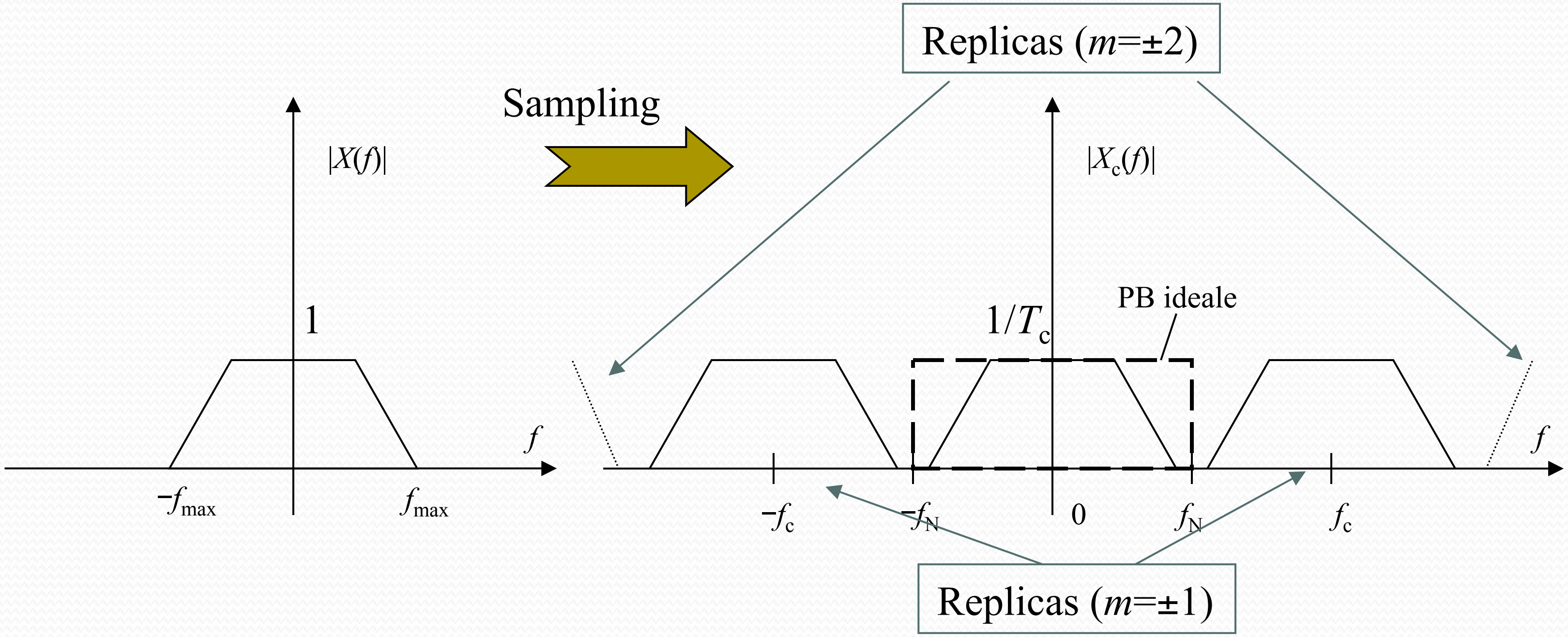
$$X_c(f) = f_c\sum_{m=-\infty}^{+\infty}X(f - mf_c)$$



The spectrum of the sampled signal is periodic and contains infinite replicas of the signal spectrum, spaced with a pitch $\boldsymbol{f_c}$

Shannon's theorem

An ideal **low-pass** filter (LP) with cut-off frequency equal to **Nyquist's frequency** $\boldsymbol{f_N = f_c / 2}$ allows to **reconstruct** the original signal, from the sampled signal, if the maximum frequency f_{max} of the input signal is such that $\boldsymbol{f_{max} < f_N}$

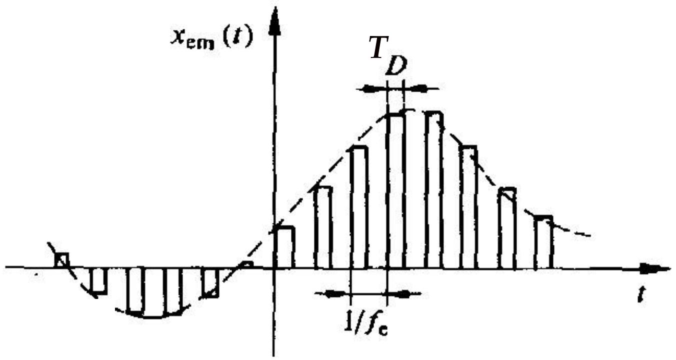


If $f_{max} > f_N$ will be ***aliasing*** on reconstructed signal.

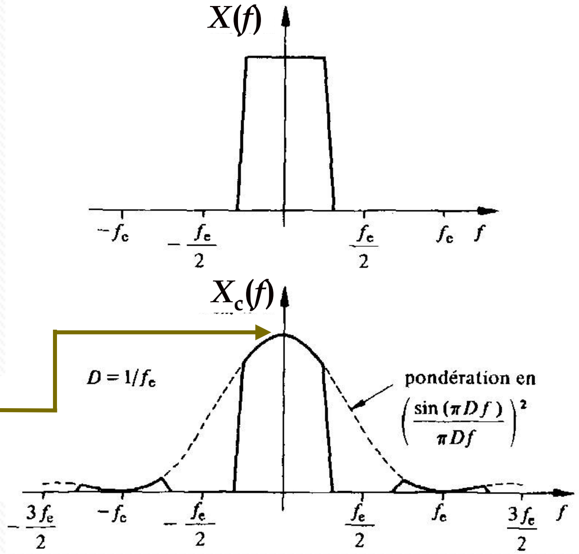
Real sampling

sampled signal
(time domain)

$$x_c(t) = x(t)\sum_{k=-\infty}^{+\infty} \mathrm{rect}\left[\left(t - kT_c\right)/T_w\right]$$



sampled signal
(spectral domain)



for fT_w<<1
the error is negligible

distortions of amplitude and phase of the reconstructed signal

$\Rightarrow T_w << 1/f_{max}$
$\Rightarrow f_{max} << 1/T_w$

$$X_c(f) = f_c T_w \frac{\sin(\pi f T_w)}{(\pi f T_w)} \sum_{m=-\infty}^{+\infty} X(f - mf_c) e^{-j\pi f T_w} \frac{\sin \pi T_c (f - mf_c)}{\pi T_c (f - mf_c)} e^{-j\pi f T_c}$$

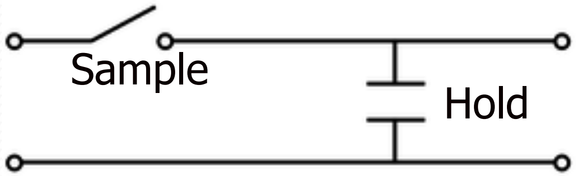
Solution to real sampling problem

POLITECNICO MILANO 1863

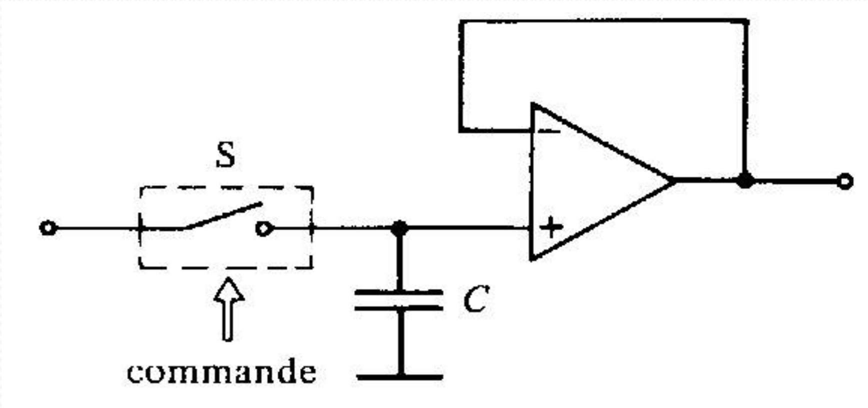
For DAQ systems with high accuracy requirements, the distortions introduced by the actual sampling should be calculated and taken into account

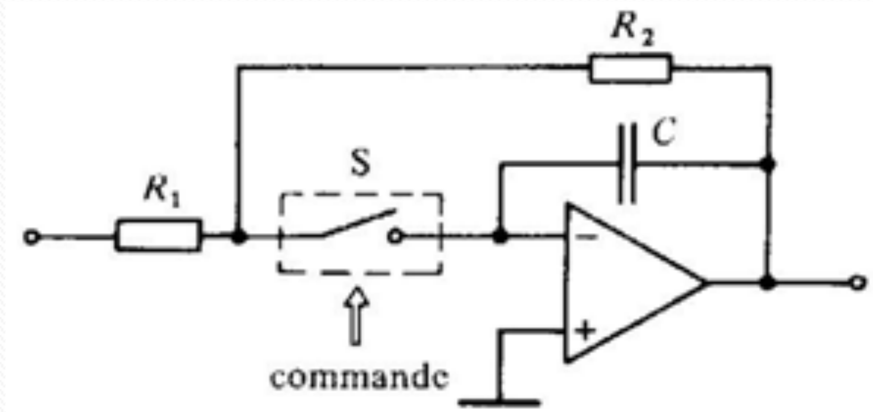
In general to reduce the effects caused by aliasing and by the finite duration of the sampling, **it is used a much higher sampling rates than the limit imposed by the Shannon theorem**
(example $f_c=10f_{c,min}=20f_{max}=20f_{s,max}$)
Of course $T_w<T_c$ and maybe $T_w<<T_c \Rightarrow T_w<<1/f_{max}$

Sample & Hold (S/H)



With the switch closed, the sampled voltage is "stored" on a capacitor (analog memory) which then keeps it when the switch is opened.





$T_w \approx \tau$ is releted with R_s $T_w \approx \tau$ is given by R_2C**=cost.**

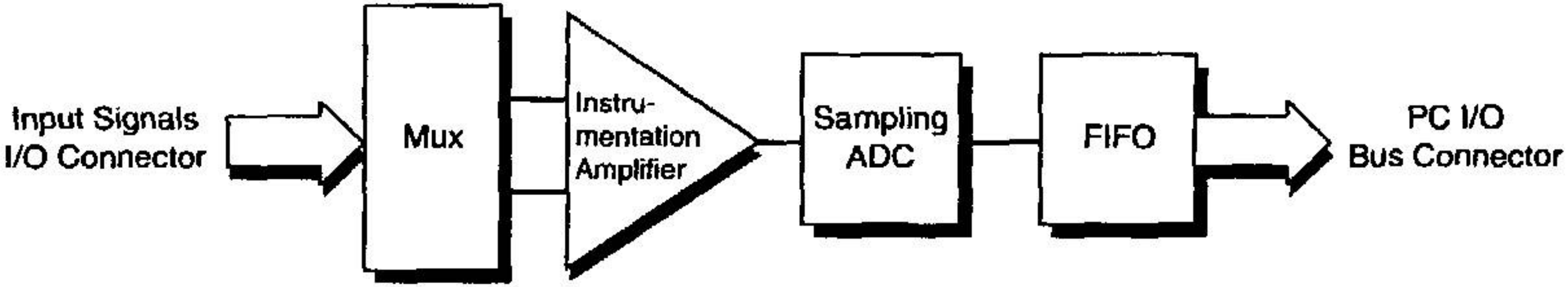
Pb. non-ideality (leakage currents)**: switch** *S* or the **capacitor** *C* **or the operational amplifier**

Data AcQuisition Board

DAQ Board

Data AcQuisition Board (DAQ)

POLITECNICO



Multiplexer – it allows you to select the different inputs available (**single-ended** or **differential**)

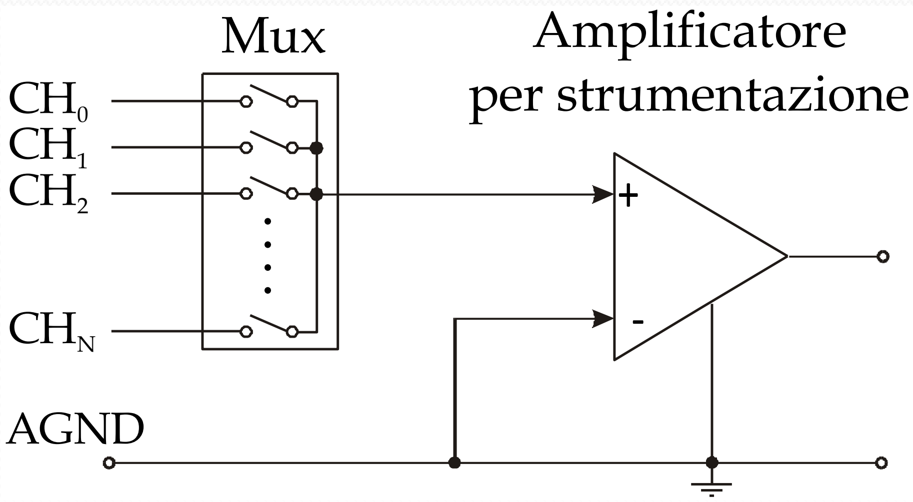
Instrumentation Amplifier (WB, variable gain) – it lets you use the full dynamic of the converter (ADC)

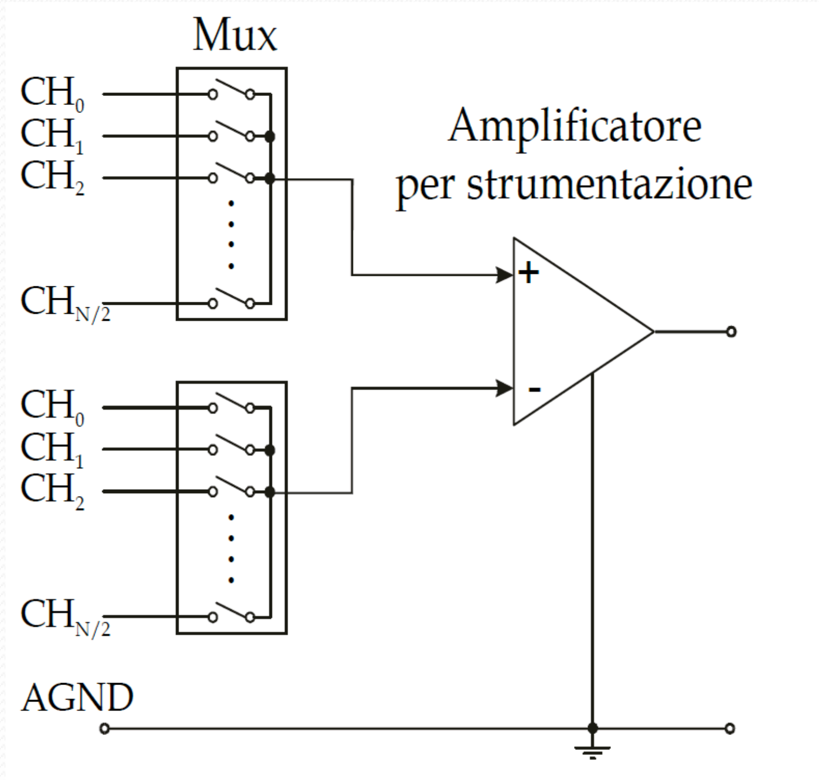
Sampling+ADC – it converts the voltage into numeric value

FIFO – it allows you to send on the PC data bus and / or directly to RAM memory (**DMA**) of PC the acquired data

The cards also have **analog outputs** (**DAC**), **I/O lines** Input/Output, and analog and digital synchronizations (**timer** e **trigger**)

Single-ended or differential inputs





If the MUX has *N* inputs (wires), the card will have **N single-ended inputs and $N/2$ differential inputs**

Typical number of **analog input channels: 8-80**

Frequency of sampling

Frequency of sampling – The maximum frequency at which the DAQ can digitize the signal. The DAQ can digitize the signal even at lower frequencies. When you make a multi-channel sampling (i.e. on multiple inputs), maximum sampling frequency on each channel is equal to:

$$f_{sample,signal} = \frac{f_{max,ADC}}{N_{signals}}$$

Typical values for the sampling frequency in the case of general purpose DAQ:

$$10\ \mathrm{kSa/s} < f_{sample} < 10\ \mathrm{MSa/s}$$

ADC dynamic

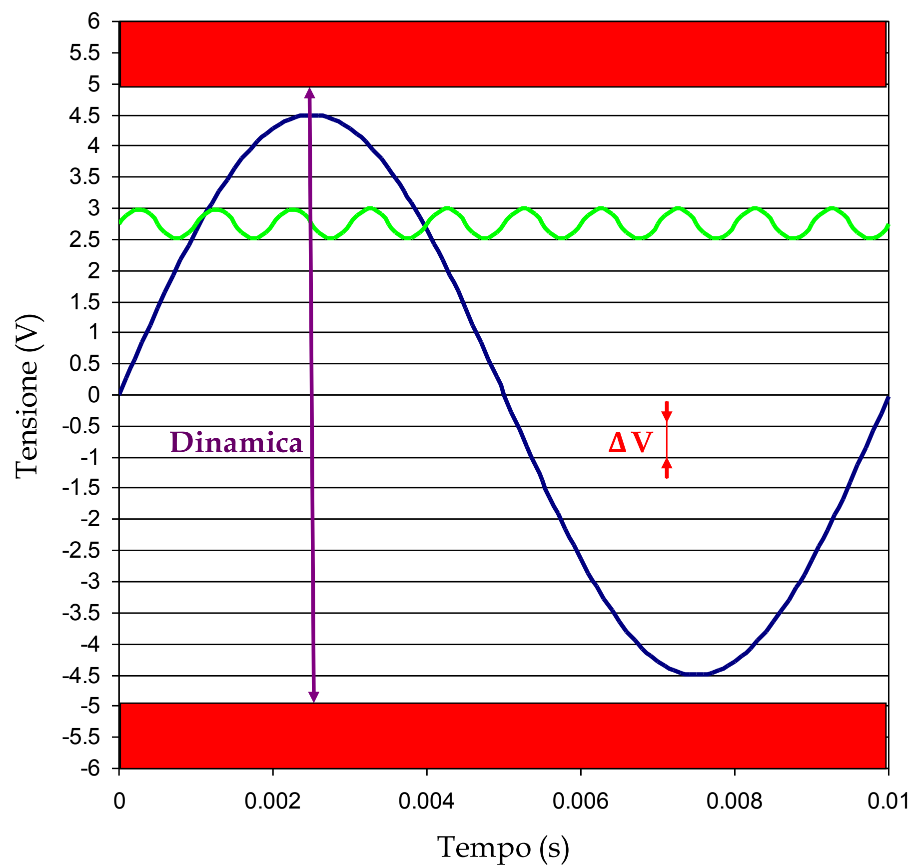
ADC dynamic: D_{ADC}
is fixed (not always fits the signal)

It maximizes the resolution amplifying the signal:

$$G = D_{ADC} / D_{signal}$$

Typical amplifier gains with **±5 V ADC dynamic:**

G=100 D_s=**±50 mV**
G=10 D_s=**±0.5 V**
G=1 D_s=**±5 V**
G=0.5 D_s=**±10 V**



$$\Delta V_{daq} = D_s/2^n = (100mV, 1V, 10V, 20V)/2^n$$

ADC resolution

POLITECNICO MILANO 1863

Resolution – The number of bits (***n***) that the ADC uses to represent **the analog input signal**. The higher the resolution, the greater the number of levels in which the dynamic (D_{ADC}) of the ADC is divided, ergo, the minimum detectable voltage (dim. res. ΔV) varies as a function of the dynamics of the input signal and the gain amplifier. The **minimum detectable voltage** for an ideal DAQ is the dimensional resolution.

$$\Delta V = \frac{D_{ADC}}{G \times 2^n}$$ **dimensional resolution**

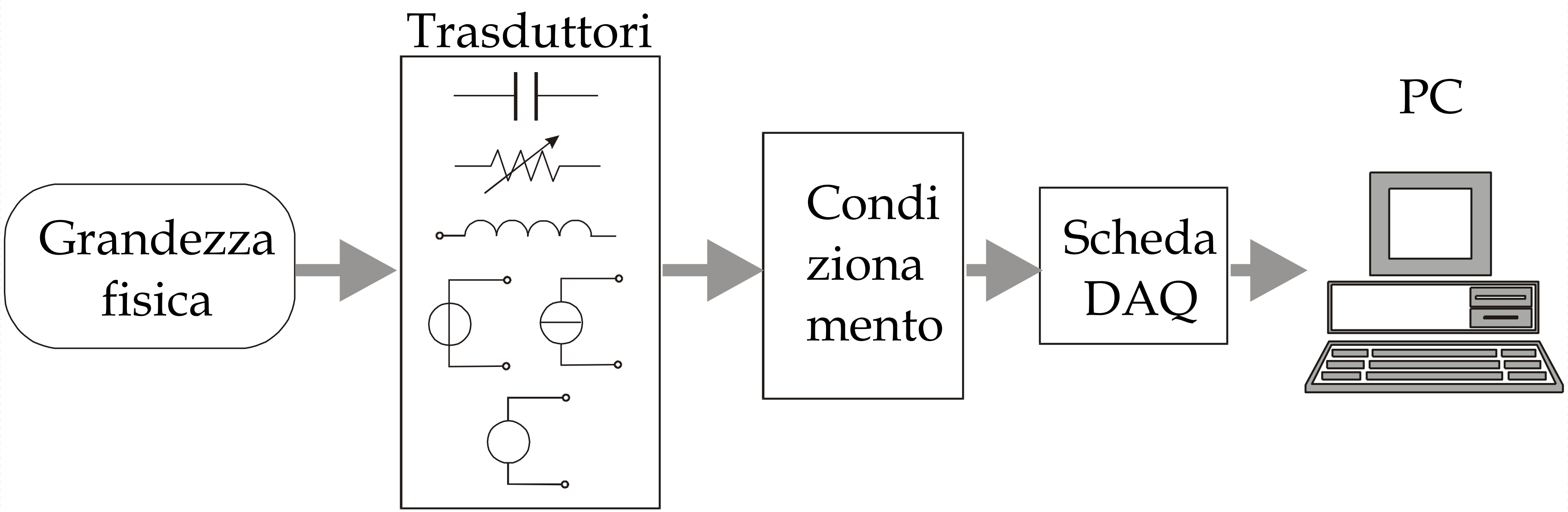
$$\delta = \frac{1}{N} = \frac{1}{2^n}$$ **adim. res.**

Typical resolution of a DAQ: **12-18 bit (δ = 2.5×10^{-4} - 4×10^{-6})**

The resolution is just one of the characteristics that describe **the accuracy of the DAQ**. Electronic noise and errors (linearity, offset, gain) need to be considered to properly describe the accuracy of the ADC.

Data acquisition from the physical world

POLITECNICO

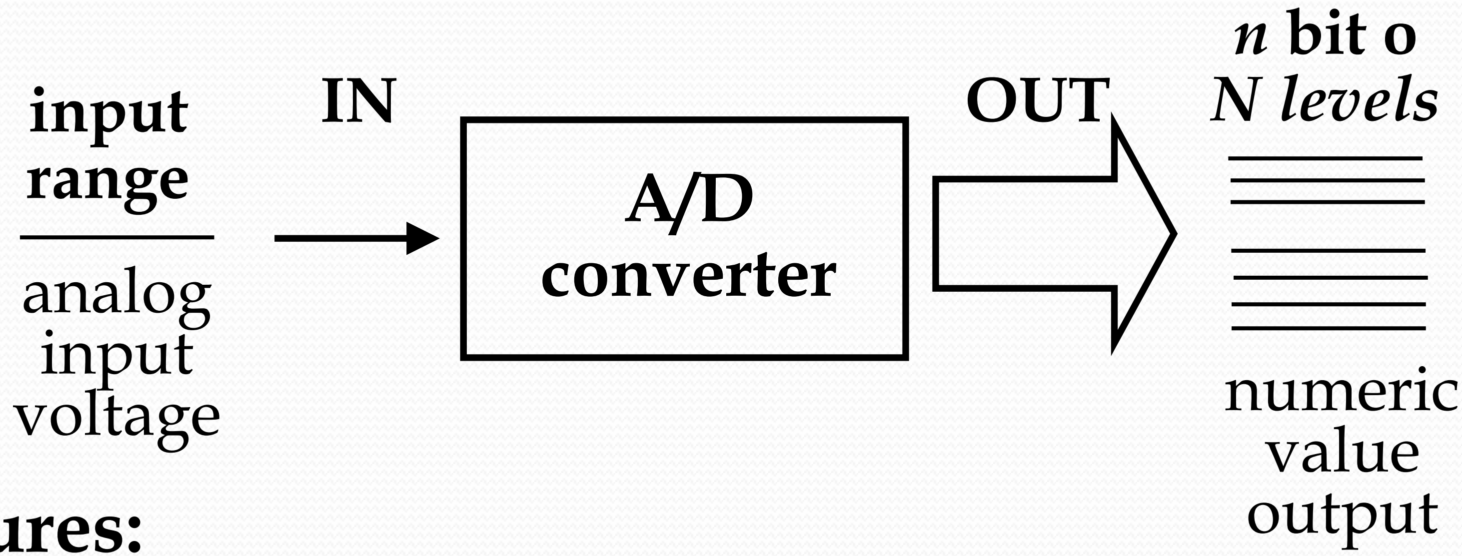


physical phenomenon → Transducer→ Conditioning→

<u>DAQ board</u> → PC+DSP (<u>elaboration and viasualization</u>)

<u>ACQUISITION</u> AND <u>DATA ANALYSIS</u>

Features of A/D converter



Features:

RANGE: ***D*** [V]

RESOLUTION: ***n*** [bit] or ***N*** [levels]

VELOCITY: f_c [Hz] o T_c [s] or also f_{sample} [Sa/s]

ELECTR. NOISE [mV_{rms}] or **BIT EQ.** n_e [bit]

EXPENSE[€] e **CONSUMPTION** [mW]

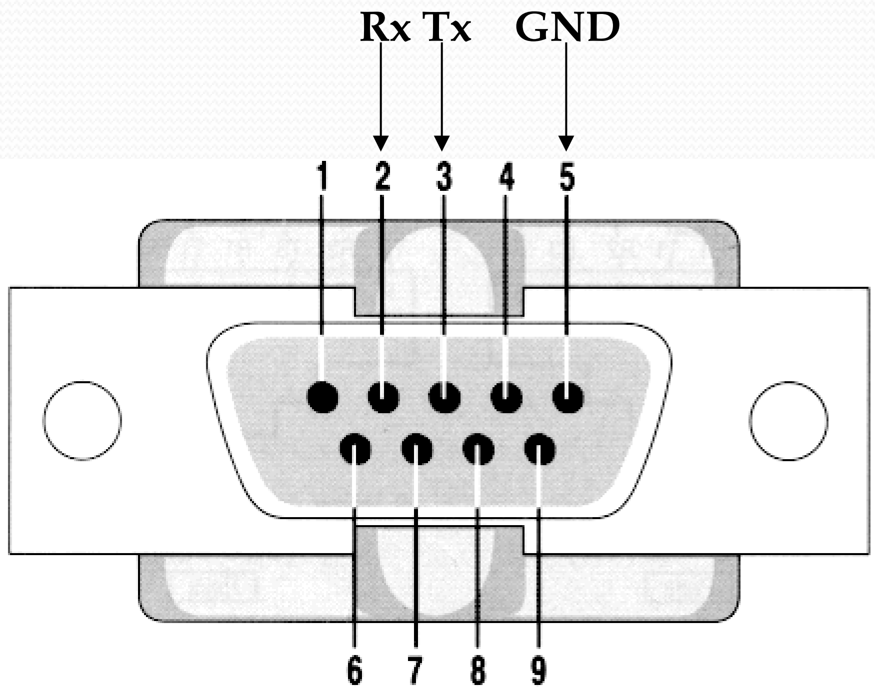
COMMUNICATION PROTOCOLS

Serial interface RS-232

Serial communication takes place through **three lines**:

(2) RX receiver;
(3) TX transmission;
(5) linea di massa **GND**;
(the RX and TX levels are then reffered to the GND).

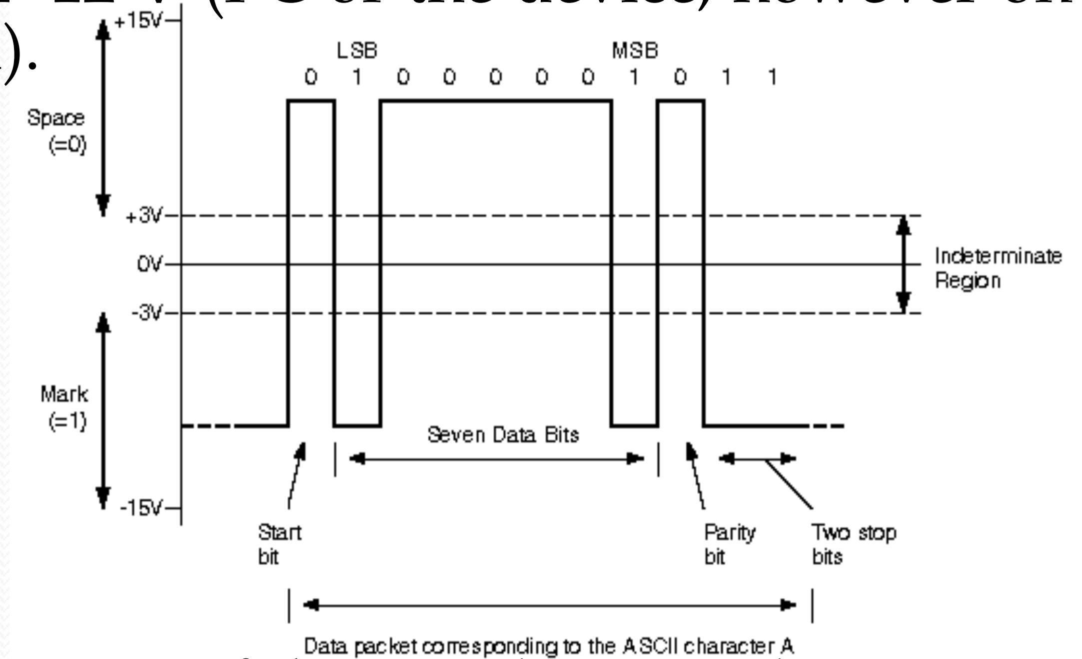
Additional lines may be available but in general are not required.



Pin	Signal	Pin	Signal
1	Data Carrier Detect	6	Data Set Ready
2	Received Data	7	Request to Send
3	Transmitted Data	8	Clear to Send
4	Data Terminal Ready	9	Ring Indicator
5	Signal Ground		

Serial interface RS-232

The corresponding voltage level to the **high** state (1) is between **+3 V and +12 V** while logic **low** (0) state is between **-3 V and -12 V** (PC or the device, however on the cable levels opposed).



Bit trasmitted

Key parameters of the **serial protocol** are:
baud rate (≈9.6 kbit/s up to 115200), **data bits (7)**, **stop bit(s)** and **parity bit (1)**. The single message is a package consisting of the data, stop and parity bits.

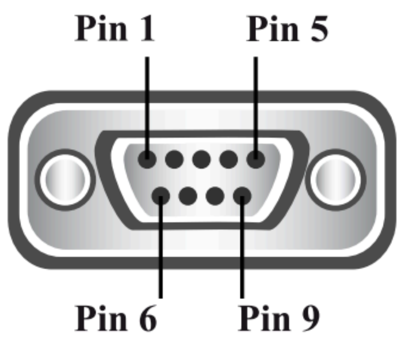
Serial interface RS-422 – RS-485

The **RS-422** interface is a protocol for **serial data** communication that involves the use of **two wires with differential line and multipoint** (balanced differential). Provides, for each pair of wires, the transmission of unidirectional and non-reversible data, of data transmission lines terminated or unterminated.

Unlike EIA RS-485, in which it differs only for the ability to be in high impedance on the line if not selected, **the EIA-422 does not allow multiple transmitters**, multiple receivers but only. **EIA RS-485, such as the RS-422 EIA can be made full-duplex using four wires** (two twisted pairs), but since the EIA-485 is a specification of the multipoint type, this is not necessary in many cases.

Pin 1	TXD-
Pin 2	TXD+
Pin 3	RTS-
Pin 4	RTS+
Pin 5	GND
Pin 6	RXD-
Pin 7	RXD+
Pin 8	CTS
Pin 9	CTS+

RS422/485 Pinout (9 Pin)

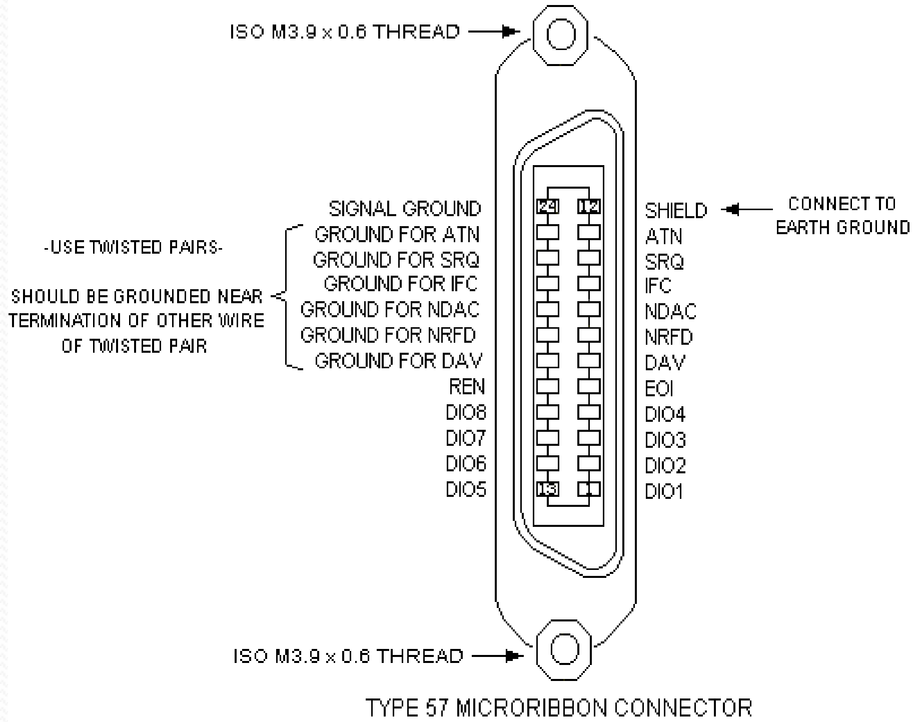


It isn't indicated neither recommended any protocol for data transmission. **EIA RS-485** allows for the configuration of local networks at low cost and multipoint communications. **It allows a very high transmission speed** (**35 Mbit/s** up to **10 m** and **100 kbit/s** to **1,200 m**).

IEEE-488 protocol (GPIB)

The essential features of the **parallel interface GPIB** (*general purpose interface bus*) are:

- **8 data lines** (DIO1-DIO8, TTL 0-5 V), **5 lines of interface management and 3 *handshake* lines**;
- the **data transfer code** is **ASCII 7 bit + 1 parity bit** (1 data = 1 byte);
- the maximum number of **connectable devices is 15** with maximum connection length of 20 m
- **maximum velocity of transmission 1 Mbyte/s** (typical ≈400 kbyte/s)
- each connected instrument has its GPIB address

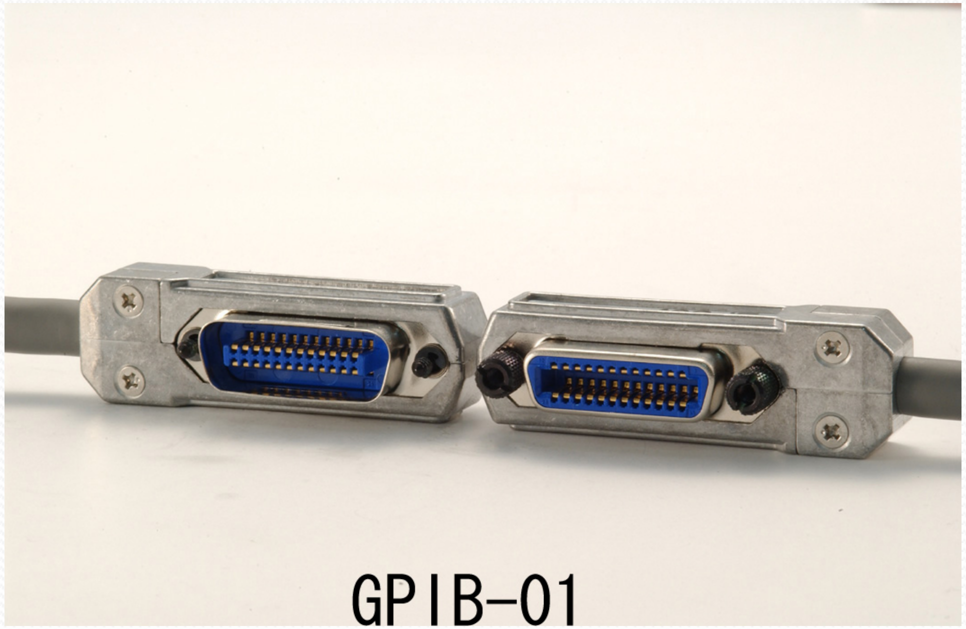


IEEE-488 protocol (GPIB)

Each instrument connected to the bus can take one of **three active roles (mode)**:

- **LISTENER** ⇒ receive data
- **TALKER** ⇒ transmit data
- **CONTROLLER** ⇒ manage the bus

A device can also assume more than one role. The minimum configuration requires a **controller** and a **talker** or a **listener**.
There is also a fourth role called **idler** in which the device is in the **standby** phase



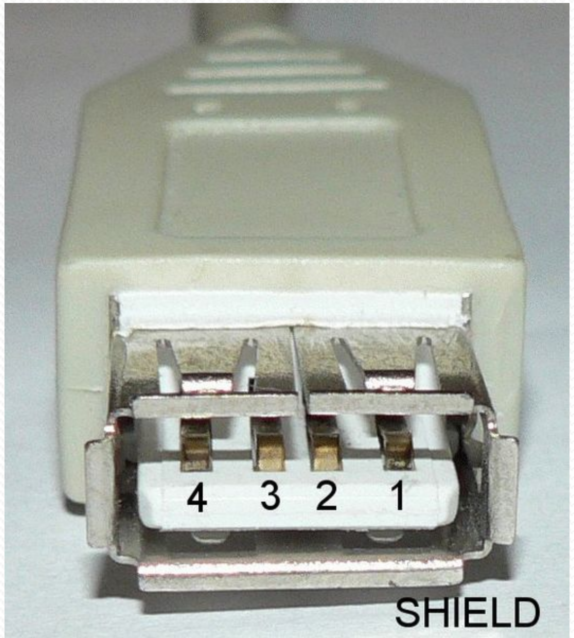
USB interface

POLITECNICO
MILANO 1863

The *Universal Serial Bus* (USB) is a **serial communication** standard that enables you to connect **multiple devices (up to 127** including the peripheral connection hub) with each other or to a computer.

1 **VBUS** ⇒ power supply (+5 V)
2 **D$^-$** ⇒ "received" data
3 **D$^+$** ⇒ "transmitted" data
4 **GND** ⇒ ground reference

The data are differential voltages with:
$\Delta V^+ \cong +3V$ e $\Delta V^- \cong -3V$ per $\Delta V=(D^+ - D^-)=2.8 \div 3.6V$



Data transfer Velocity:

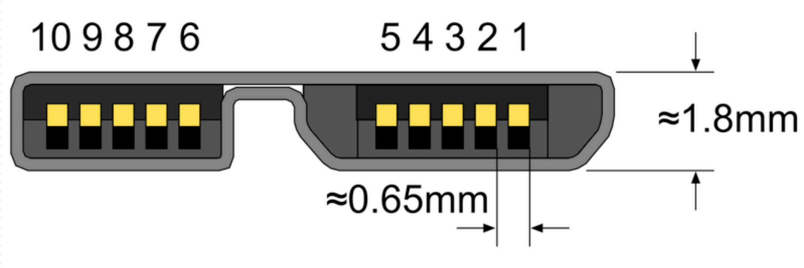
USB **1.0** ⇒ **1.5 Mbit/s**
USB 1.1 o **2.0** ***full speed*** ⇒ **12 Mbit/s**
USB **2.0** ***high speed*** ⇒ **480 Mbit/s**
USB **3.0** ***super speed*** ⇒ **4.8 Gbit/s**

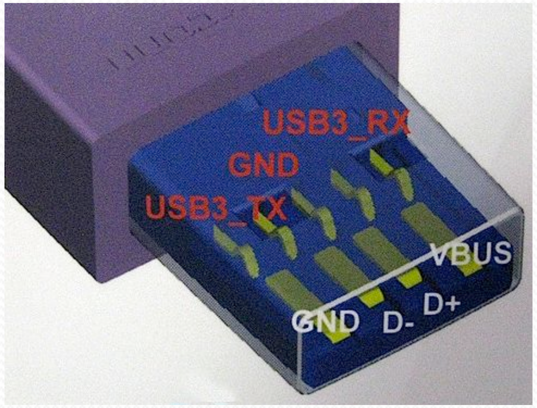
USB interface

USB protocol up to 2.0 using a ***half-duplex*** communication while 3.0 allows ***full-duplex*** transmission.

On the **power connection** (**VBUS**) a PC can deliver up to 500 mA of current at 5 V. As a result it can even power devices with a "**low power" consumption (<2.5 W**).

The USB 3.0 can deliver a current of **150 mA,** with peaks of 900 mA, which opens the door to a whole new range of USB accessories.





USB interface

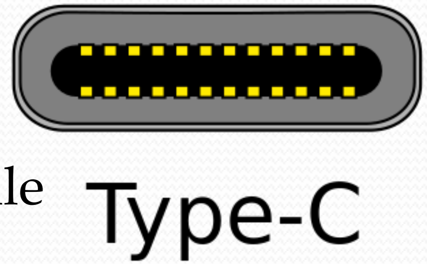
When a device or hub is connected to the tree it is assigned a logical address. After synchronized with the receiver *clock* sends a bit string indicating what kind of **data transfer** to perform:

1 **Control** ⇒ command and status operations
2 **Interrupt** ⇒ latencies guaranteed, few data transferred
3 **Bulk** ⇒ latencies not guaranteed, transfer of a large data packet
4 **Isosinchronous** ⇒ continuous data transfer (*streaming*)

A **single USB cable** can be up **to 5 m**
Connecting the hubs too, the **peripherals** can be up to **30 m by the user.**

Until today in fact if you connect multiple devices to the same USB port the data is sent to all devices and then only the person concerned picks them up, which means that everyone has to be constantly active. The new **USB 3.0**, however, is able to route traffic, so the devices are not interested can remain inactive ⇒ **energy savings**.

USB – C interface



- 24-pin double-sided connector
- non è detto che sia retrocompatibile con tutte le USB precedenti

USB evolution

Other protocols

- Firewire up to 800 Mb/s ⇒ Bus similar to GPIB
- E-sata up to 3.2 Gb/s ⇒ utilized in HD
- Thuderbolt up to 10 Gb/s ⇒ very simple protocol of communication