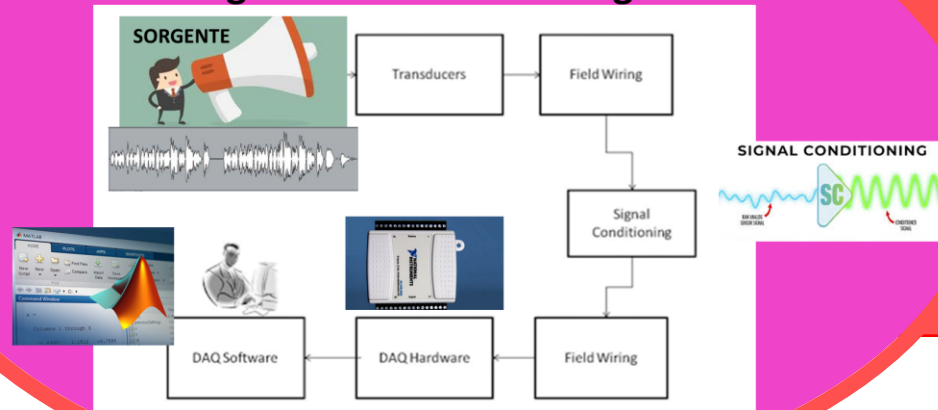


Sistemi di acquisizione dati: dalla sorgente all'analisi del segnale



LEZIONE 7:

Sistemi di acquisizione dati (DAQ)

Misure e Acquisizione di Dati Biomedici

Sarah Tonello, PhD

Dip. Ingegneria dell'Informazione

Università di Padova

Outline

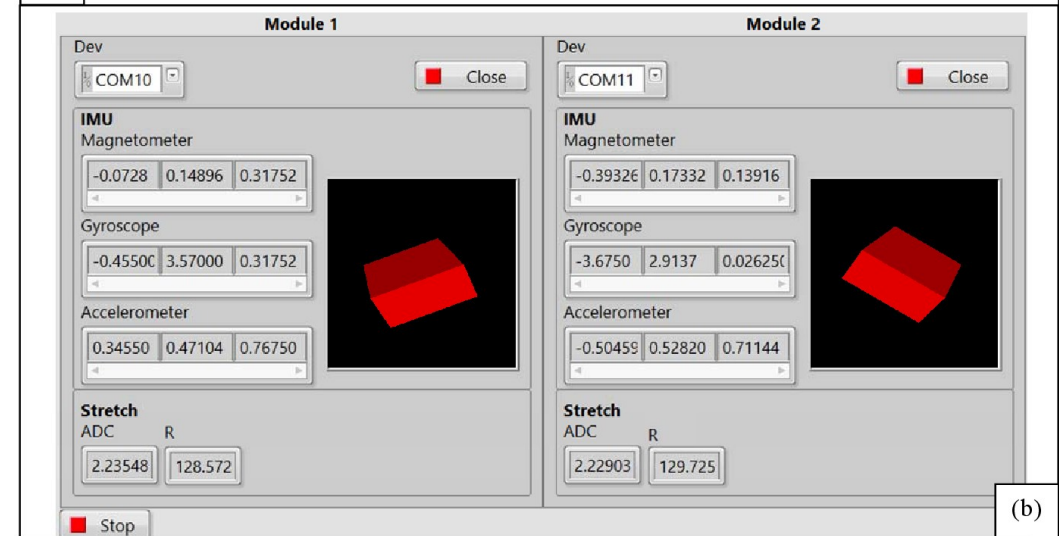
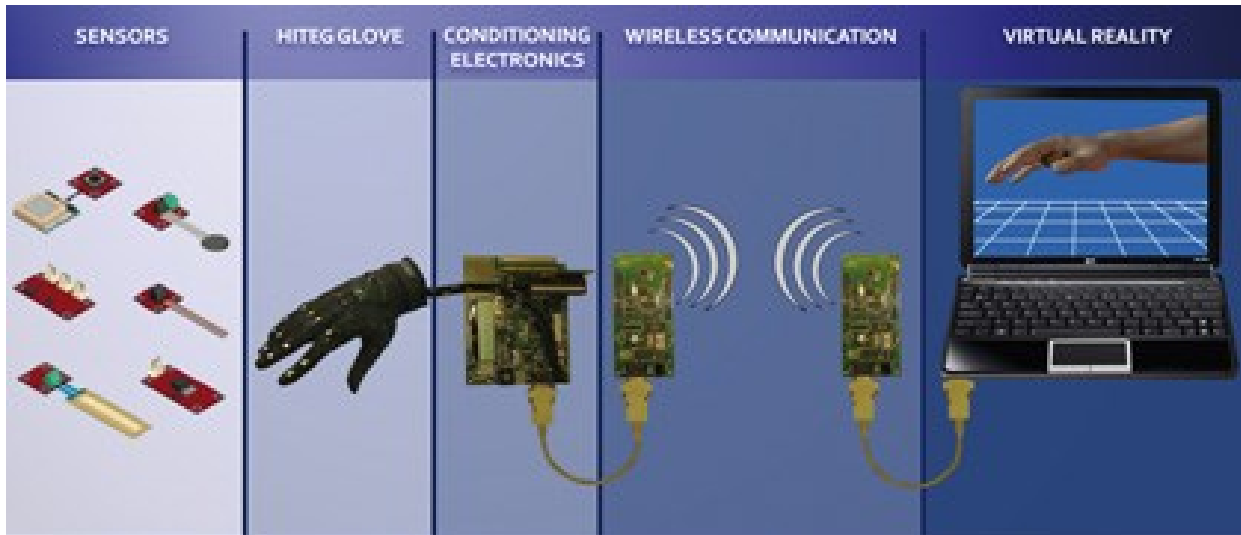
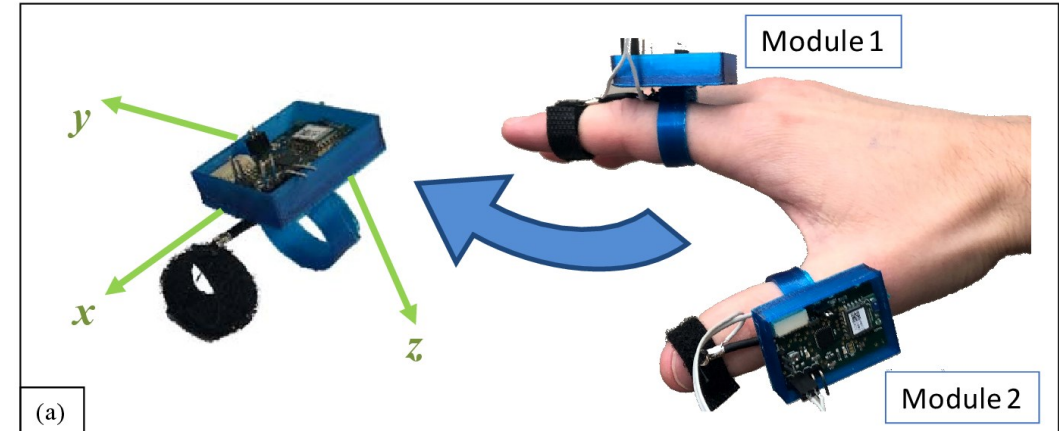
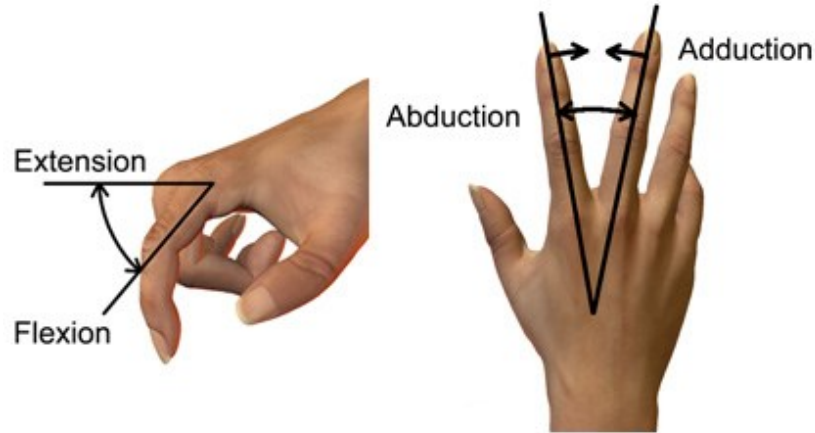
- Generalità sui sistemi DAQ
- Selettore di ingressi e PGA
- Conversione A/D
- Gestione della memoria
- Modalità di acquisizione e interfaccia utente
- Esempio di scheda di acquisizione

QUIZ



Esempio utilizzo DAQ in sistema sperimentale

APPLICAZIONE:
Tracking dei
movimenti della mano
per riabilitazione ,
gestione in remoto o
valutazione movimenti
industria 4.0



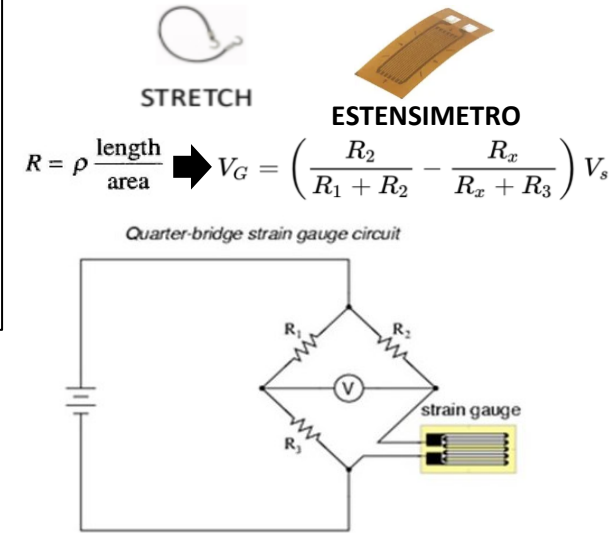
II Workshop on Metrology for Industry 4.0 and IoT (MetroInd4.0&IoT) 2019, Application of a Modular Wearable System to Track Workers' Fingers Movement in Industrial Environments, Paolo Bellitti et al.

Esempio utilizzo DAQ in sistema sperimentale

APPLICAZIONE:
Tracking dei movimenti della mano per riabilitazione o gestione in remoto

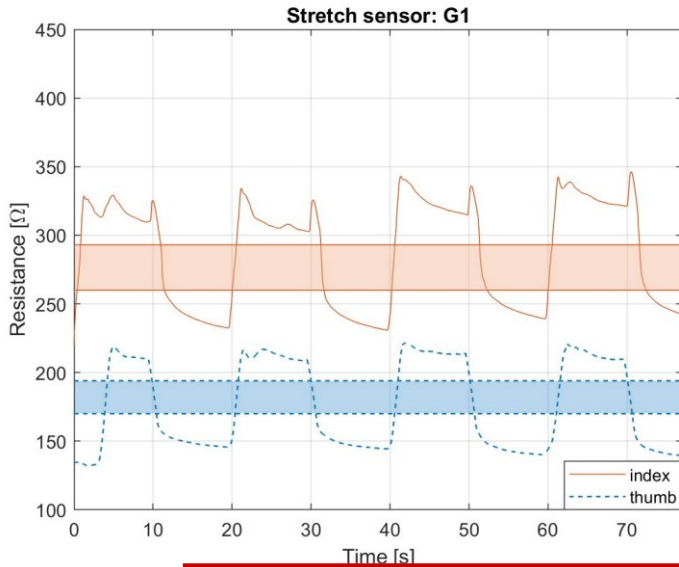
MISURE RICHIESTA:
Deformazione
Accelerazione

TRASDUTTORE:
Estensimetri (strain gauge) o stretchable sensors è uno strumento di misura utilizzato per rilevare piccole deformazioni dimensionali di un corpo sottoposto a sollecitazioni meccaniche o termiche.



SALVATAGGIO E ACQUISIZIONE DATI

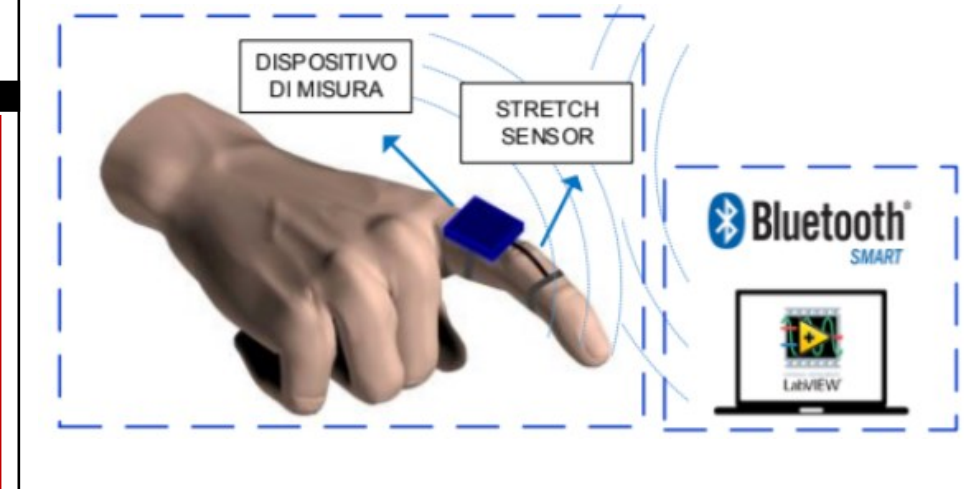
→ Segnale ottenuto



→ Interfaccia utente (GUI)

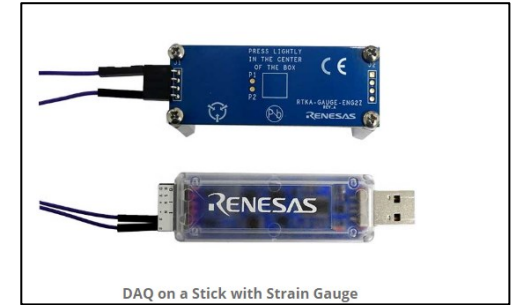


DISPOSITIVO DI MISURA



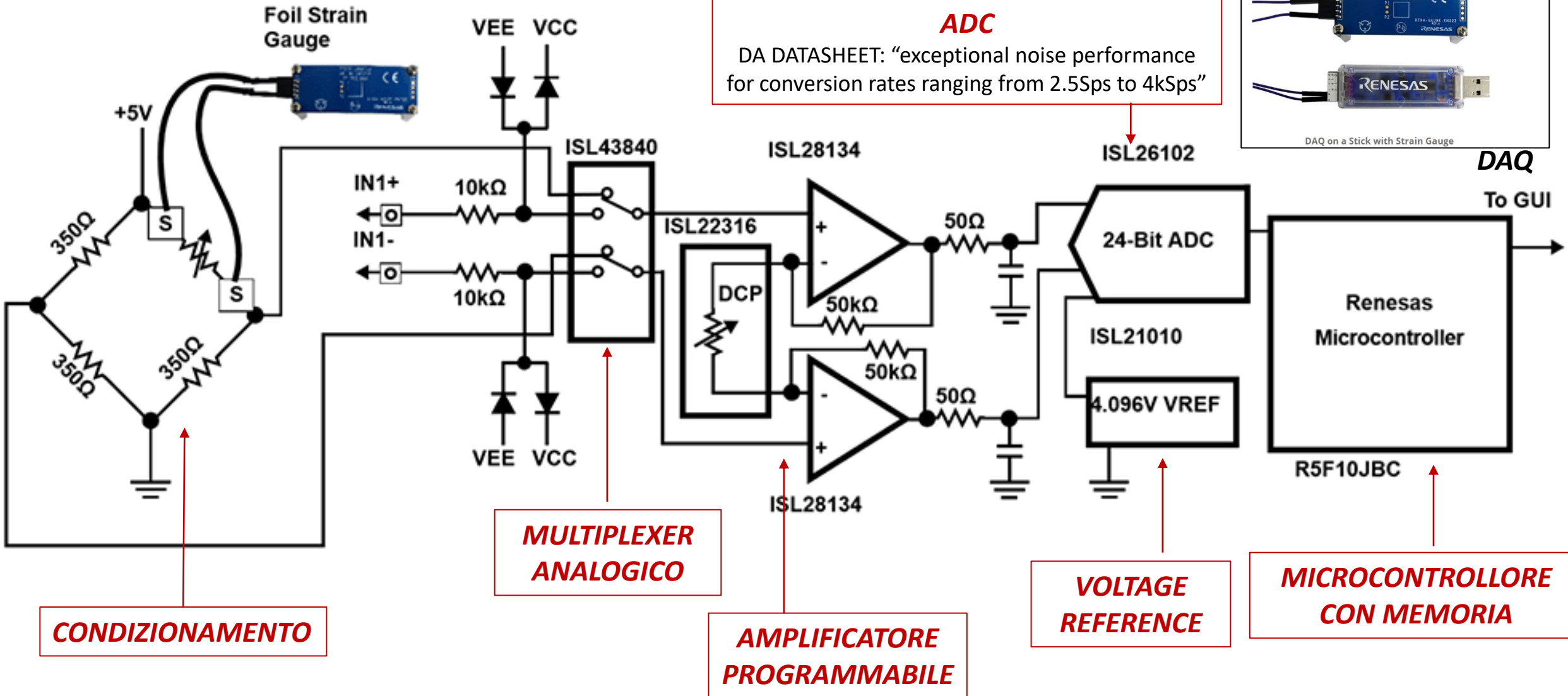
Cosa utilizzo per salvare, trasmettere i segnali acquisiti per consentirne l'elaborazione?
MODULO PROGRAMMABILE PER ACQUISIZIONE DATI

Esempio utilizzo DAQ in sistema sperimentale

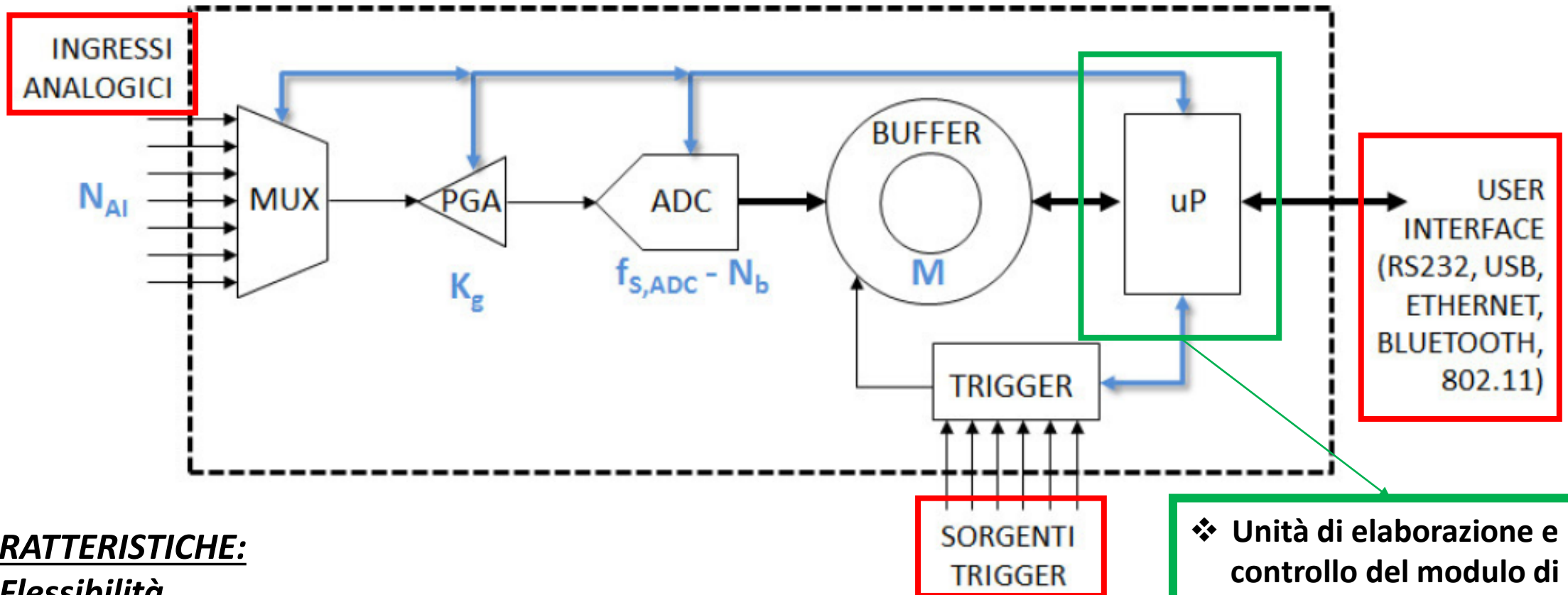


DAQ

To GUI



Struttura sistemi DAQ



CARATTERISTICHE:

- Flessibilità
- Adattabilità al variare dei requisiti richiesti
- Facilità di reimpiego
- Consentono di **ridurre tempo per progettazione e realizzazione del sistema di misura**, necessitando di una semplice configurazione dei parametri di funzionamento.

❖ **Unità di elaborazione e controllo del modulo di acquisizione dati**, solitamente microprocessore dedicato (μP) si occupa di gestire tutti i componenti con opportuni segnali di temporizzazione

Outline

- Generalità sui sistemi DAQ
- Selettore di ingressi e PGA
- Conversione A/D
- Gestione della memoria
- Modalità di acquisizione e interfaccia utente
- Esempio di scheda di acquisizione

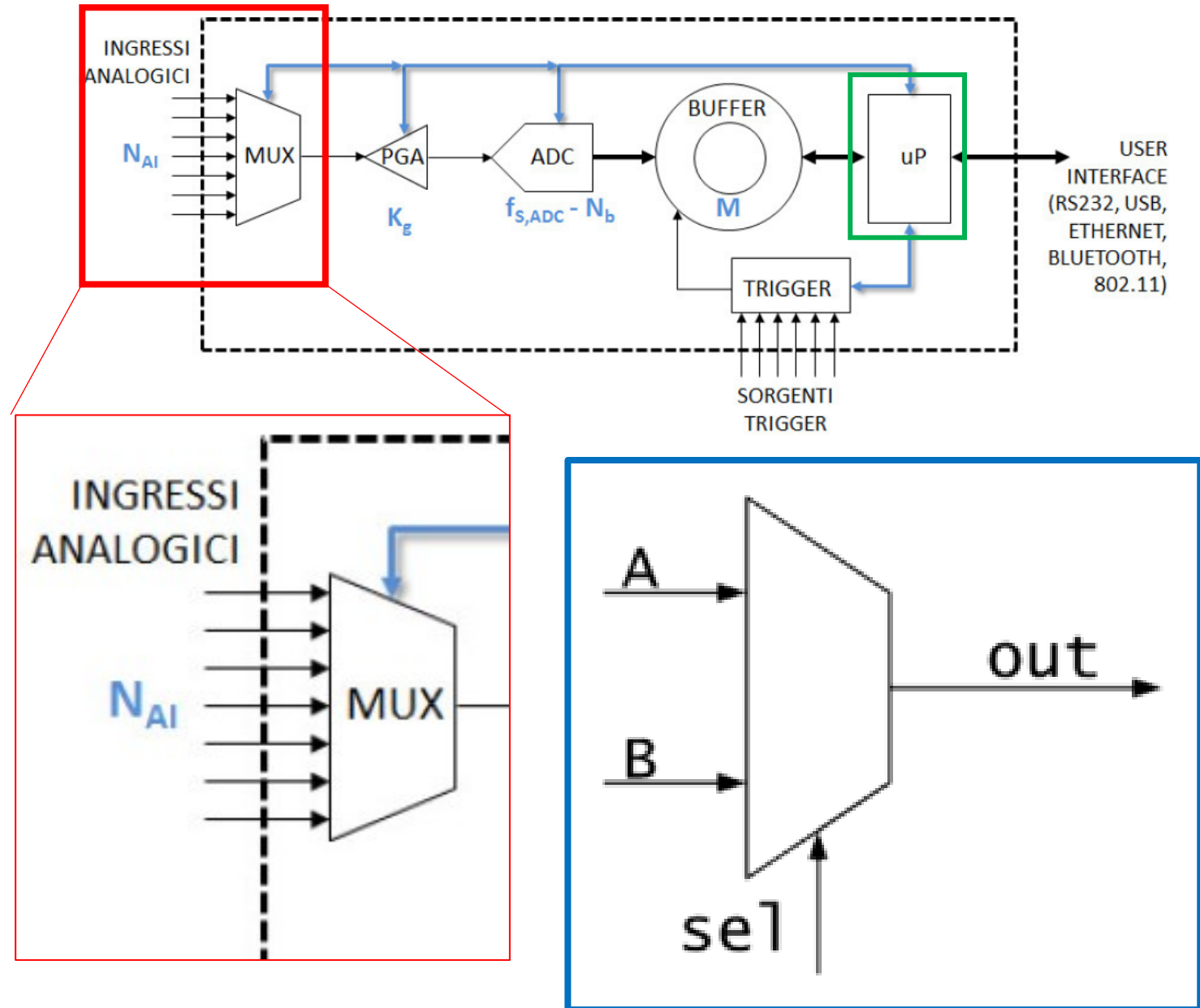
QUIZ



Struttura sistemi DAQ: ingressi e selezione

❖ N_{AI} ingressi per segnali analogici, che possono provenire dalle uscite di vari sensori.

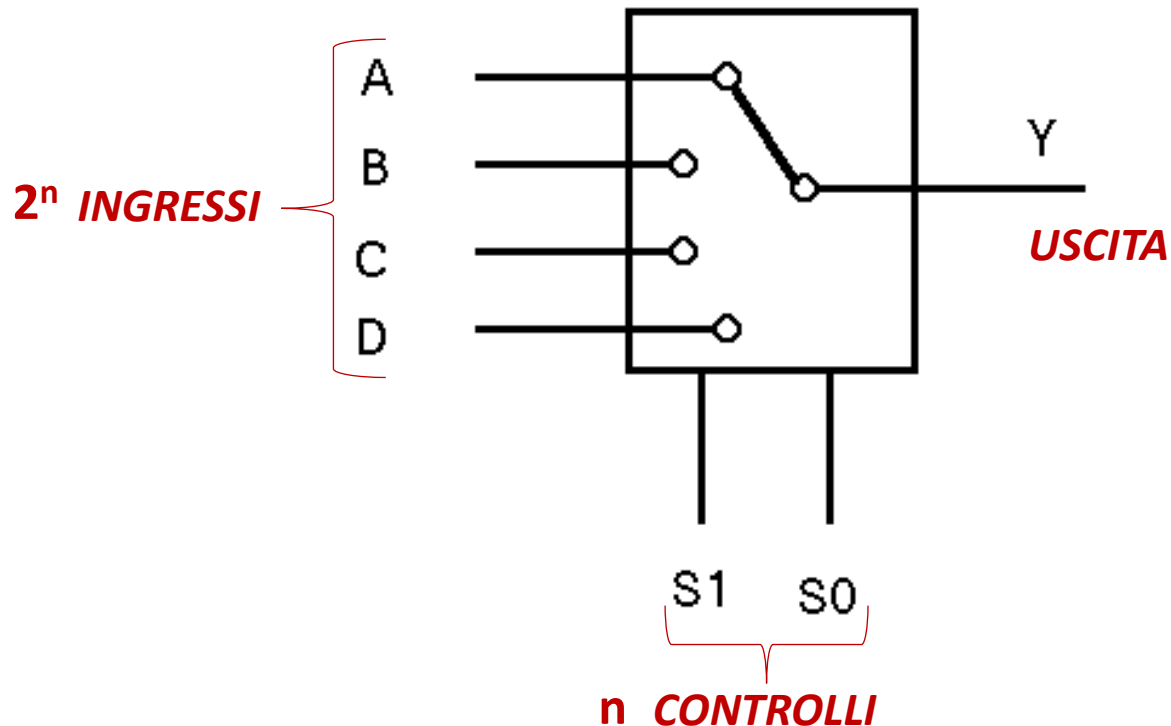
❖ **Multiplexer (MUX)**, riceve i segnali in ingresso e seleziona alla sua uscita una tra le linee di ingresso grazie a opportuni segnali di temporizzazione ricevuti dal microcontrollore



Struttura sistemi DAQ: multiplexer

→ Selettore di linee di dato in grado di selezionare **diversi segnali in ingresso** sia analogici che digitali. Una volta selezionati i segnali vengono raccolti e mandati in una **singola linea di uscita**.

→ Requisito per il segnale: no distorsione, no attenuazione



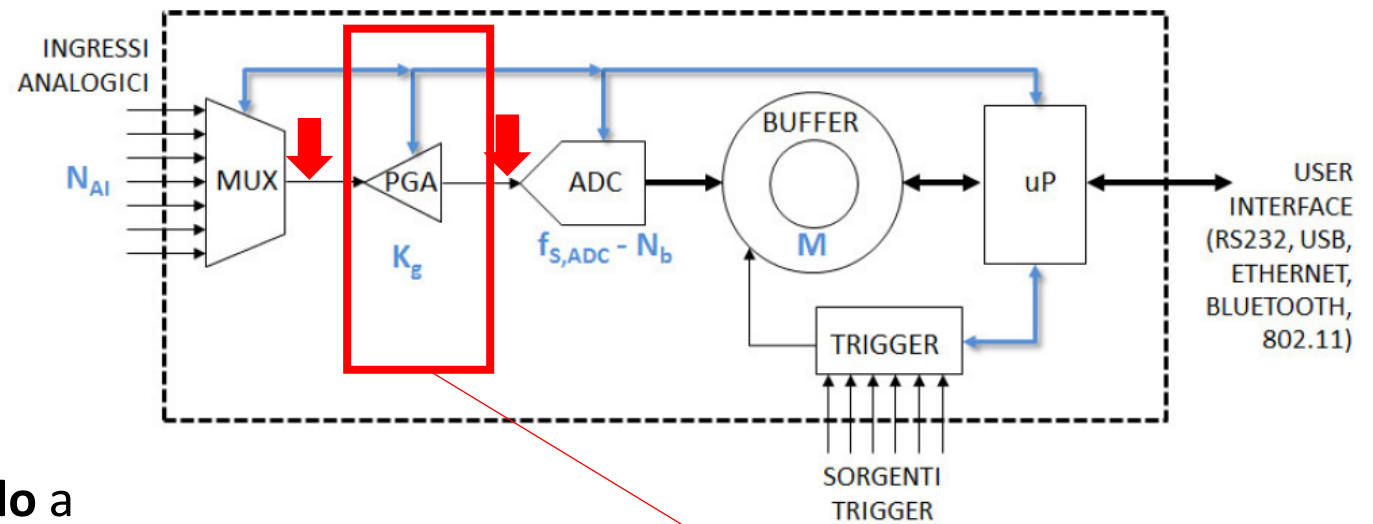
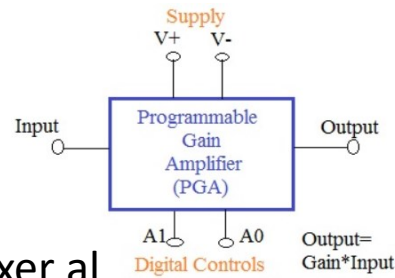
→ **Ingressi S1 e S0** (inviati dal microprocessore) gestiscono una rete combinatoria con il risultato di selezionare univocamente uno degli ingressi per ogni combinazione di S1 e S0

S1	S0	U
0	0	A
0	1	B
1	0	C
1	1	D

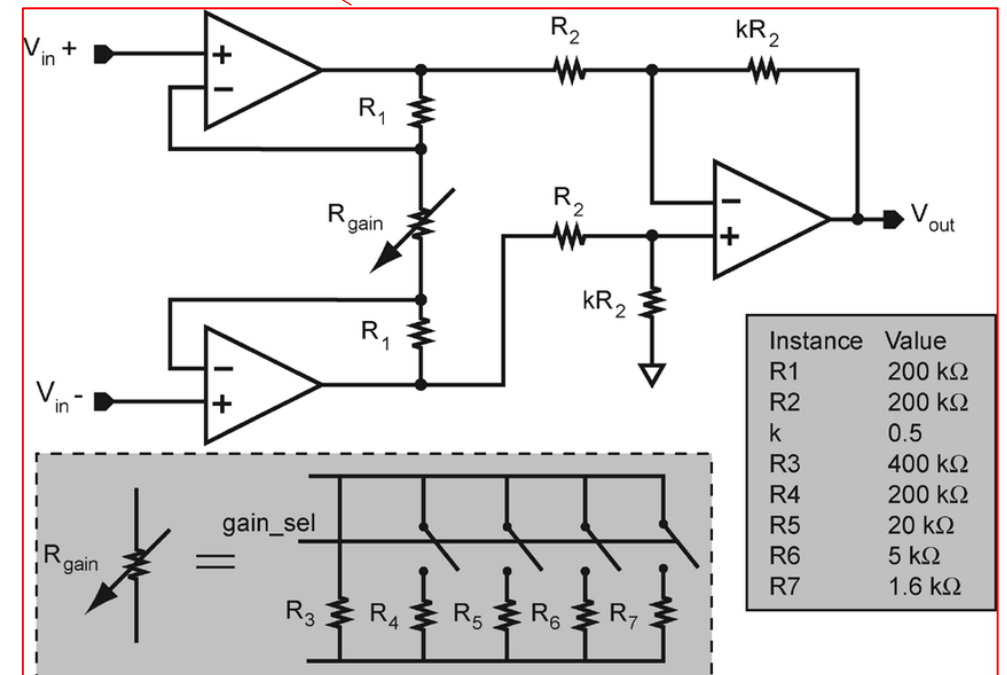
Struttura sistemi DAQ: il PGA

❖ **Amplificatore a guadagno programmabile (PGA, *programmable gain amplifier*)**

usato per adattare il segnale analogico in uscita dal multiplexer al campo di valori di ingresso dell'ADC.



- 1) GUADAGNO** modificato dall'Unità di Controllo a seconda della **linea di ingresso** selezionata tramite il multiplexer e rende possibile **l'adattamento del range di ogni linea.**
- 2) CONFIGURAZIONE** di solito **VIA SOFTWARE** trasparente all'utente:
 - noto V_{out} (ADC) generalmente utente definisce il range del segnale in input V_{in}
 - Il fattore di amplificazione del PGA viene **determinato automaticamente grazie ai comandi ricevuti dal microcontrollore**, imponendo il guadagno desiderato.



Outline

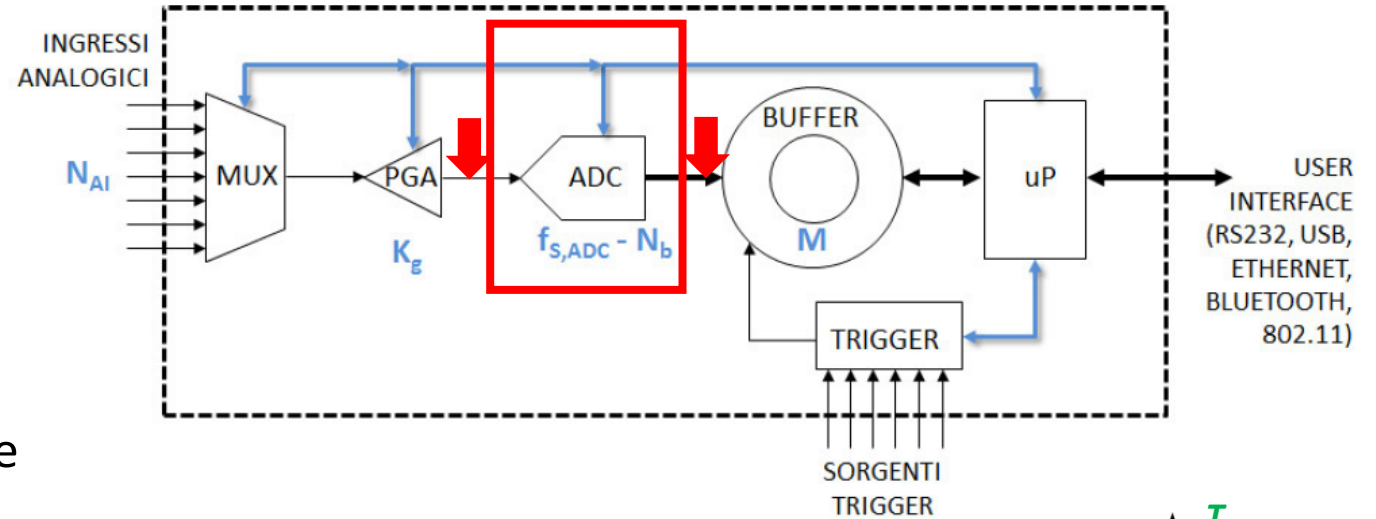
- Generalità sui sistemi DAQ
- Selettore di ingressi e PGA
- Conversione A/D
- Gestione della memoria
- Modalità di acquisizione e interfaccia utente
- Esempio di scheda di acquisizione

QUIZ

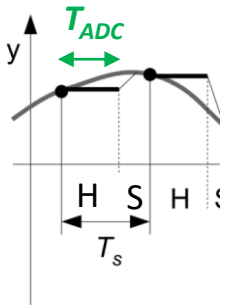


Struttura sistemi DAQ: conversione A/D

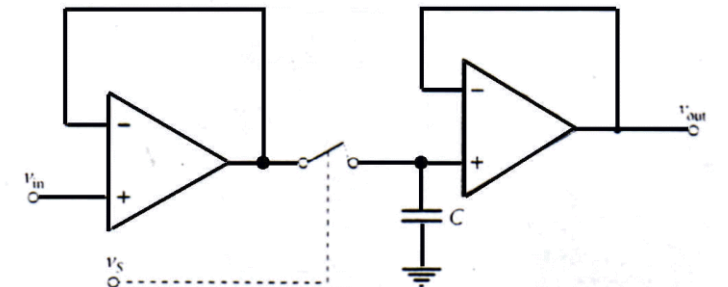
- ❖ **CONVERTITORE ANALOGICO-DIGITALE (ADC)** riceve il segnale condizionato, lo **campiona** e lo **converte** in forma numerica.
- ❖ **Generalmente per DAQ fascia medio-bassa** presente **un solo ADC** per convertire i segnali presenti sulle linee analogiche di ingresso, condiviso tra i diversi canali grazie alla scansione effettuata dal multiplexer.
- ❖ Parametri caratterizzanti:
 - **risoluzione massima** (espressa attraverso il **numero di bit b** con cui sono rappresentati i campioni in uscita)
 - **campo di ingresso** compreso tra due valori di tensione, V_{FS}^- e V_{FS}^+ , spesso simmetrici
 - **tempo di conversione T_{ADC}** .



N.B. Parametri dell'ADC non configurabili, ma **intervallo di campionamento (clock esterno)** può essere **variato**, a condizione che la sua durata non sia inferiore a T_{ADC} .

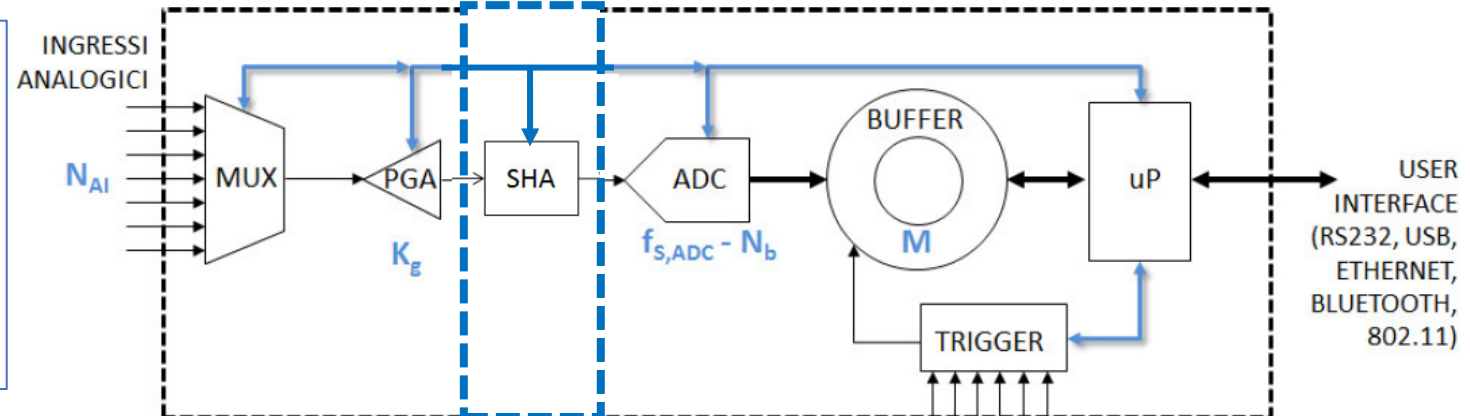


E il **SAMPLE AND HOLD AMPLIFIER?**



Struttura sistemi DAQ: campionamento e conversione A/D

- **Sample and Hold necessario** per mantenere costante il segnale durante conversione T_{ADC} (**tempo non modificabile**)
- Spesso incorporato nel blocco ADC



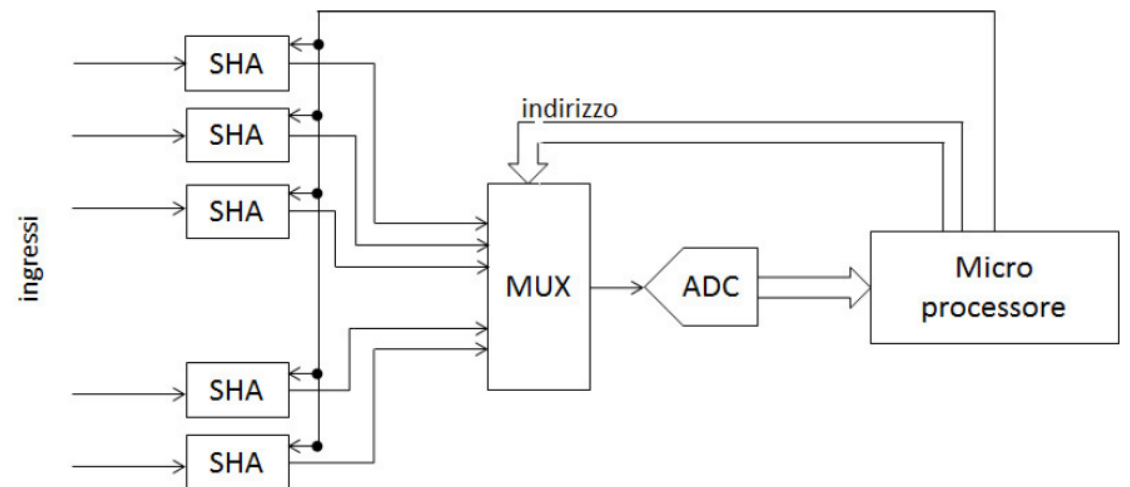
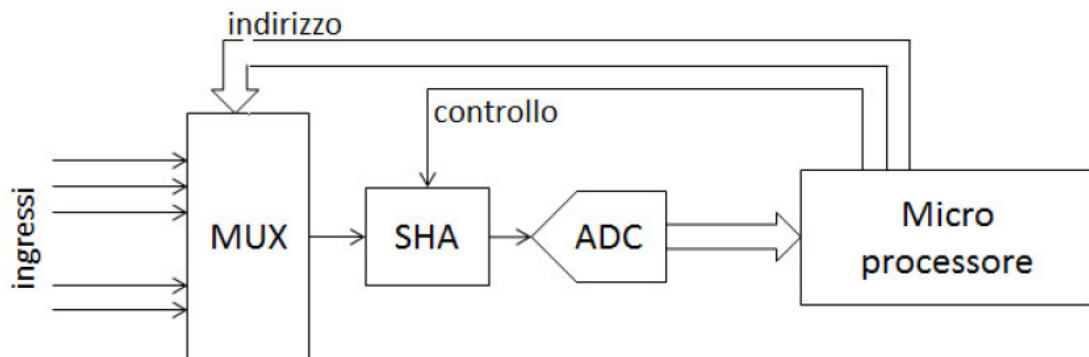
- **Attenzione: la posizione e il numero di SHA possono cambiare e questo andrà ad influenzare la gestione della conversione e di riflesso delle tempistiche di acquisizione dati**

Acquisizione dati **NON SIMULTANEA**

Acquisizione dati **SIMULTANEA**

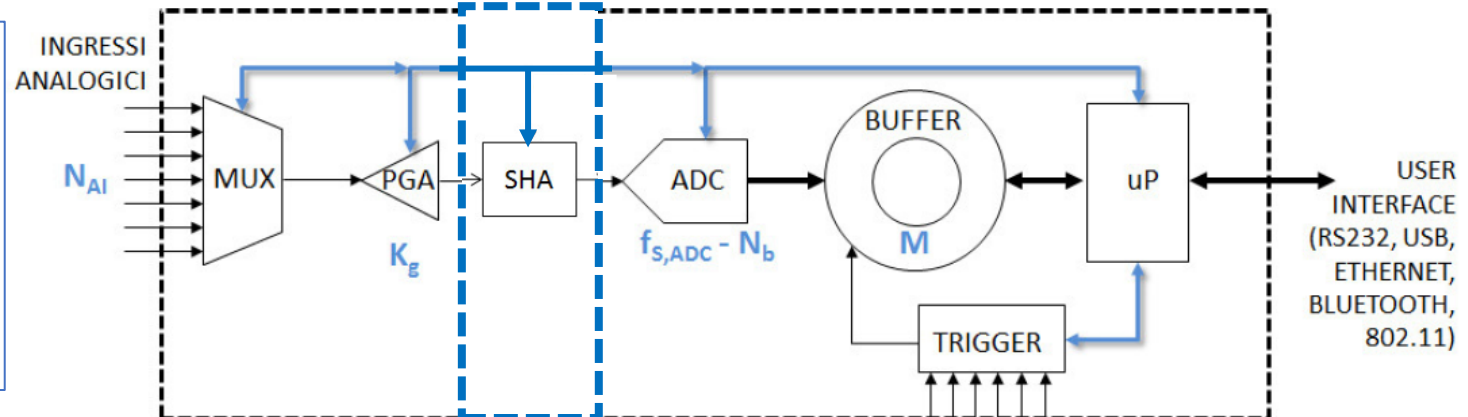
Singolo SHA → logica di accesso a divisione di tempo (TDMA, time division multiple access)

Multipli SHA → campionamento simultaneo



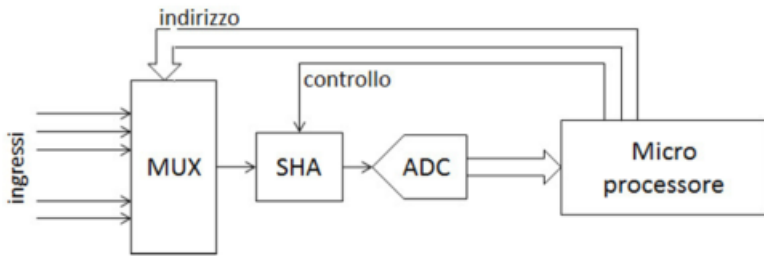
Struttura sistemi DAQ: campionamento e conversione A/D

- **Sample and Hold necessario** per mantenere costante il segnale durante conversione T_{ADC} (**tempo non modificabile**)
- Spesso incorporato nel blocco ADC



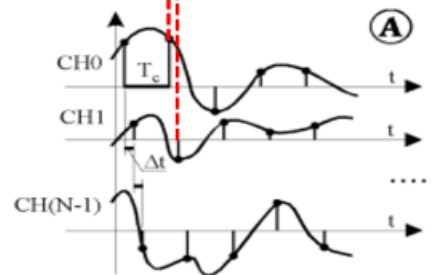
▪ **Attenzione: la posizione e il numero di SHA possono cambiare e questo andrà ad influenzare la gestione della conversione e di riflesso delle tempistiche di acquisizione dati**

Acquisizione dati **NON SIMULTANEA**



- Un solo SHA
- Stessa frequenza di campionamento f_s ($=1/T_s$)
- I campioni usati nella conversione non si riferiscono al medesimo istante temporale, bensì ad istanti di tempo successivi, distanti tra loro di quantità multiple di Δt

$$\Delta t = T_{S,ADC} = T_s / N_{AI}$$

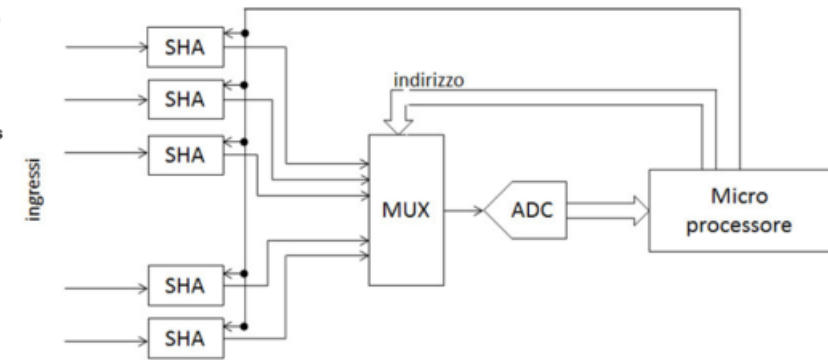


Microprocessore contiene indirizzi dei canali in ingresso attivi e gestisce la **temporizzazione dei segnali** secondo i seguenti passaggi:

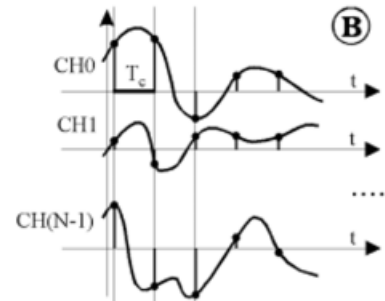
- 1) Ogni $\Delta t = T_{S,ADC} = T_s / N_{AI}$ secondi (tempo necessario per svolgere la conversione) invia un nuovo indirizzo al multiplexer per iniziare nuova conversione.
Essendo $T_s > T_{S,ADC} \Rightarrow T_s > N_{AI} \cdot T_{ADC}$
- 2) quando multiplexer si è stabilizzato, segnale di sample all'amplificatore SHA, attraverso la linea di controllo, dando inizio alla conversione.

$$f_s = 1/T_s \quad f_{MUX} = N_{AI} \cdot f_s$$

Acquisizione dati **SIMULTANEA**



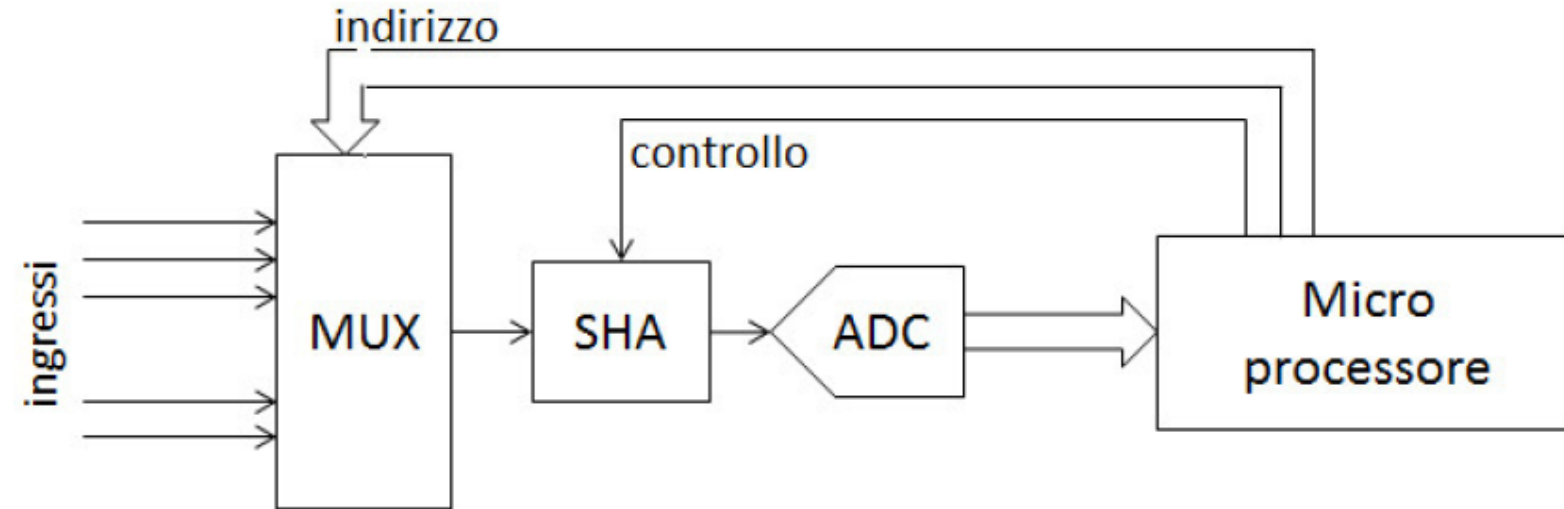
- Tanti SHA quanti canali in ingresso
- Tutti ricevono nello stesso istante segnale di controllo dal microprocessore, con la cadenza f_s



➢ Per ciascun SHA la fase di Hold tale da consentire una completa scansione di tutti gli ingressi da parte del multiplexer e quindi almeno pari a T_s . **Proporzionale al numero di ingressi N_{AI}**

➢ I campioni riferiti tutti al medesimo istante temporale.

Acquisizione dati **NON SIMULTANEA**



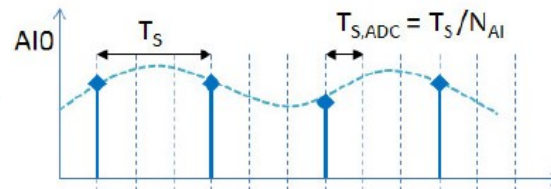
- Un solo SHA
- Stessa frequenza di campionamento $f_s (=1/T_s)$
- I campioni usati nella conversione non si riferiscono al medesimo istante temporale, bensì ad istanti di tempo successivi, distanti tra loro di quantità multiple di

$$\Delta t = T_{S,ADC} = T_s / N_{AI}$$

Microprocessore contiene indirizzi dei canali in ingresso attivi e gestisce la **temporizzazione dei segnali** secondo i seguenti passaggi:

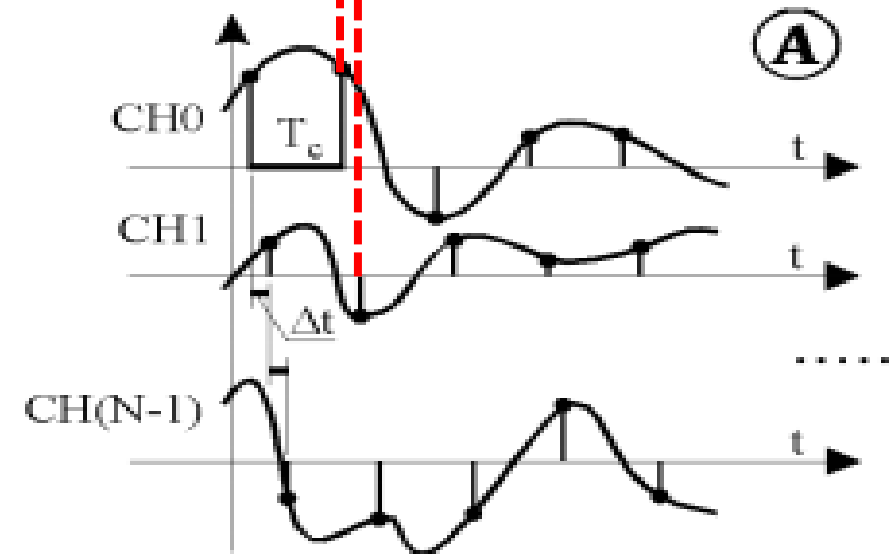
- 1) Ogni $\Delta t = T_{S,ADC} = T_s / N_{AI}$ secondi (tempo necessario per svolgere la conversione) invia un nuovo indirizzo **al multiplexer** per iniziare nuova conversione.

Essendo $T_s > T_{S,ADC} \Rightarrow T_s > N_{AI} \cdot T_{ADC}$



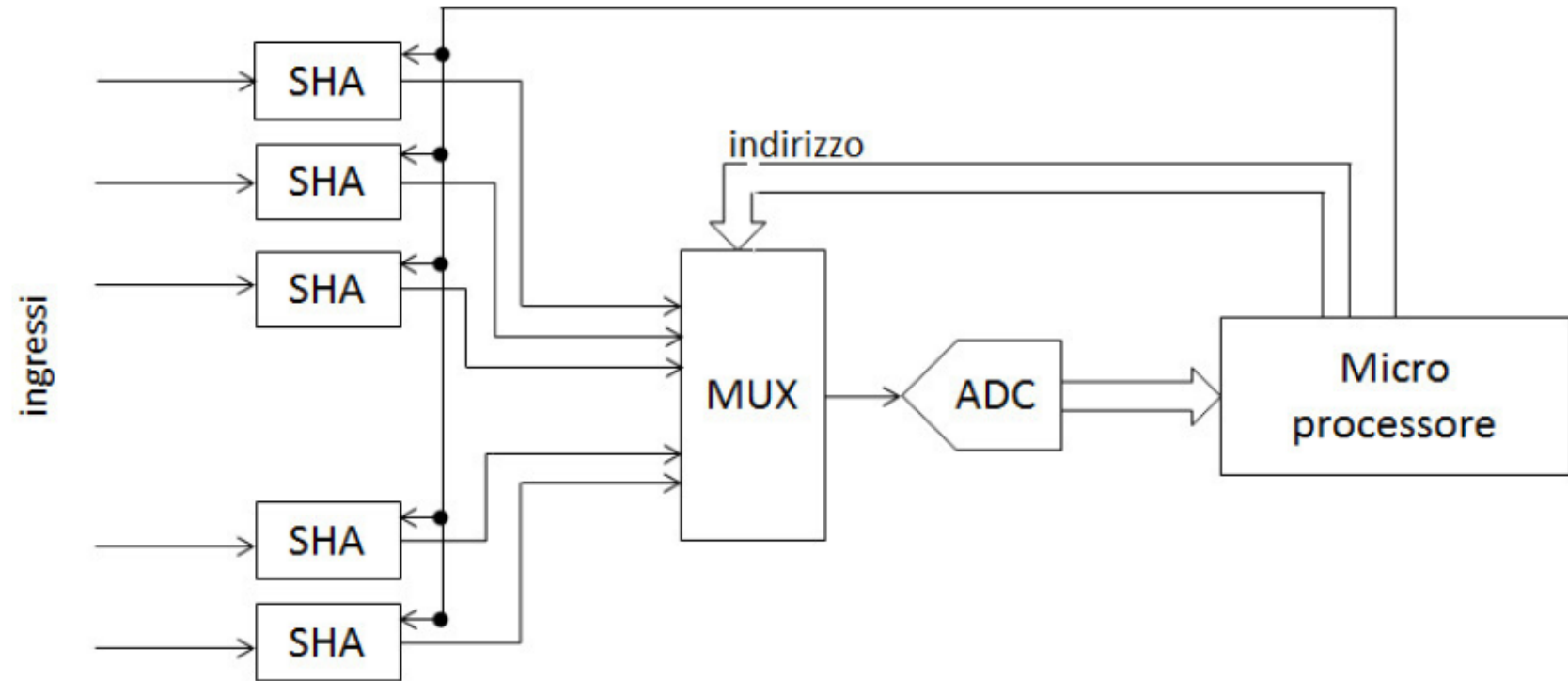
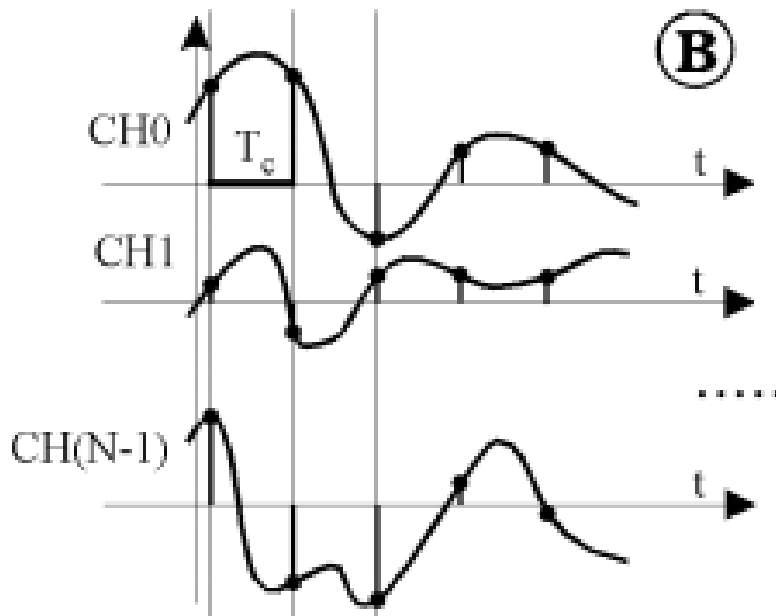
- 2) quando multiplexer si è stabilizzato, segnale di sample **all'amplificatore SHA**, attraverso la linea di controllo, dando inizio alla conversione.

$$f_s = 1/T_s \quad f_{MUX} = N_{AI} \cdot f_s$$



Acquisizione dati **SIMULTANEA**

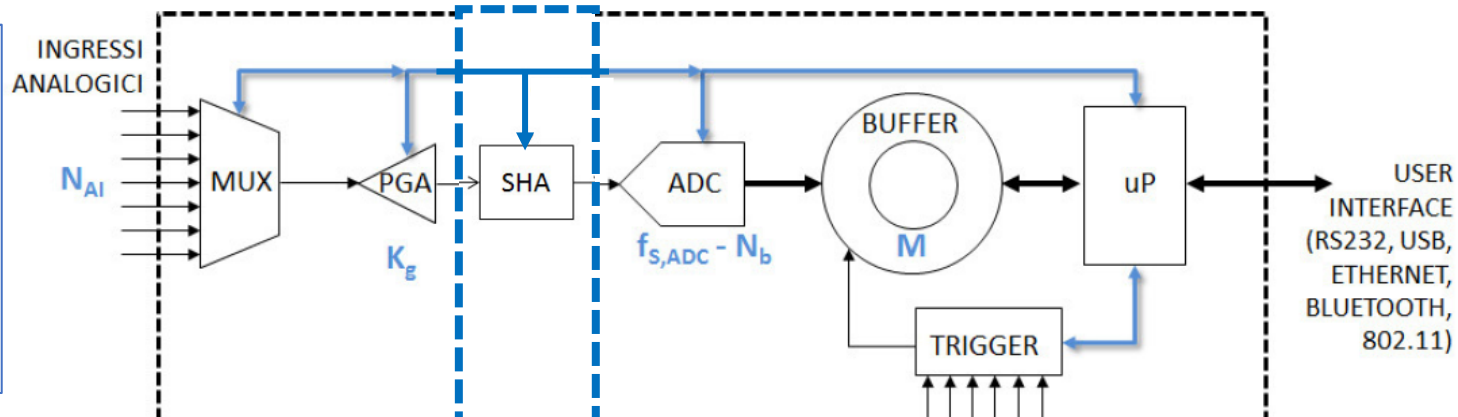
- **Tanti SHA** quanti canali in ingresso
- Tutti ricevono nello stesso istante segnale di controllo dal microprocessore, con la **cadenza f_s**



- Per ciascun SHA la **fase di Hold** tale da consentire una **completa scansione di tutti gli ingressi** da parte del multiplexer e quindi almeno pari a T_s . **Proporzionale al numero di ingressi N_{AI}**
- **I campioni riferiti tutti al medesimo istante temporale.**

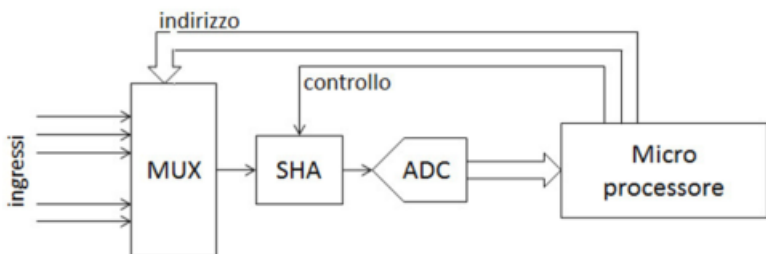
Struttura sistemi DAQ: campionamento e conversione A/D

- **Sample and Hold necessario** per mantenere costante il segnale durante conversione T_{ADC} (**tempo non modificabile**)
- Spesso incorporato nel blocco ADC



- **Attenzione: la posizione e il numero di SHA possono cambiare e questo andrà ad influenzare la gestione della conversione e di riflesso delle tempistiche di acquisizione dati**

Acquisizione dati **NON SIMULTANEA**

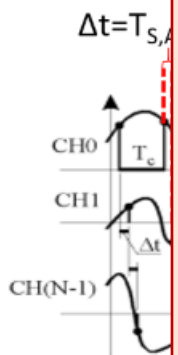


Microprocessore contiene indirizzi dei canali in ingresso attivi e gestisce la **temporizzazione dei segnali** secondo i seguenti passaggi:

- 1) Ogni $\Delta t = T_{s,ADC} = T_s / N_{AI}$ secondi (tempo necessario per svolgere la conversione) invia un nuovo indirizzo al multiplexer per iniziare nuova conversione.
Essendo $T_s > T_{s,ADC} \Rightarrow T_s > N_{AI} \cdot T_{ADC}$
- 2) quando multiplexer si è stabilizzato, segnale di sample all'amplificatore SHA, attraverso la linea di controllo, dando inizio alla conversione.

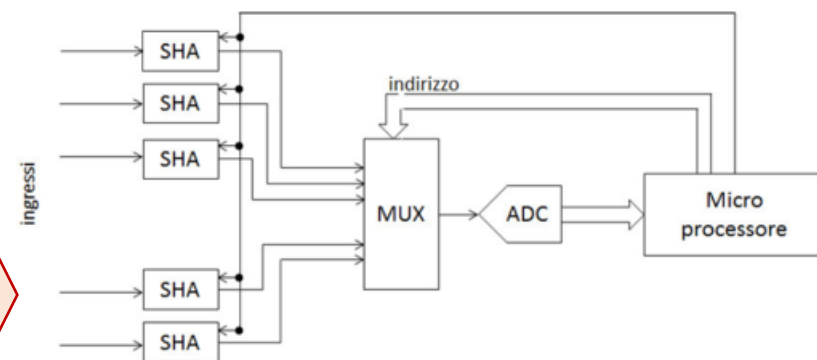
$$f_s = 1/T_s \quad f_{MUX} = N_{AI} \cdot f_s$$

- Un solo SHA
- Stessa frequenza di campionamento $f_s (=1/T_s)$
- I campioni usati riferiscono al medesimo istante temporale tra loro di quantità $\Delta t = T_{s,ADC}$



Complessità nel passare a simultaneo:
1) Aumenta numero componenti
2) Maggiore durata fase Hold

Acquisizione dati **SIMULTANEA**



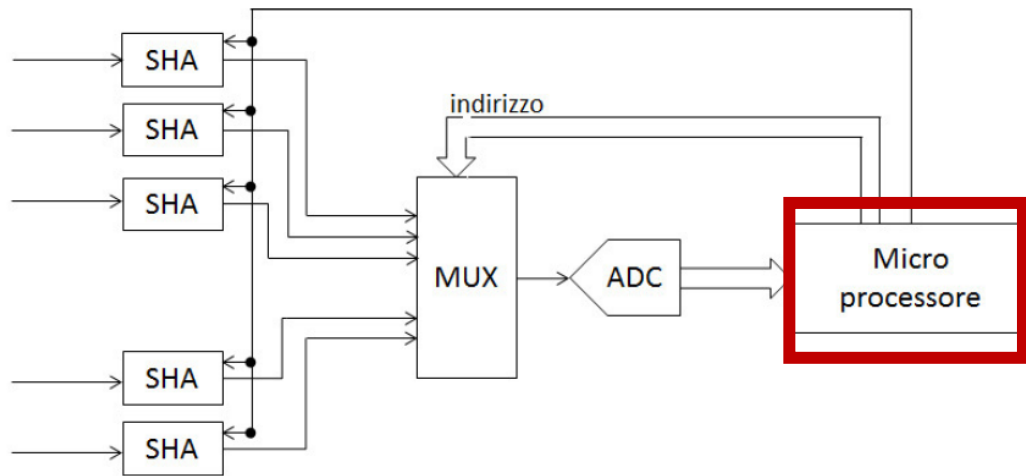
- Per ciascun SHA la **fase di Hold** tale da consentire una **completa scansione di tutti gli ingressi** da parte del multiplexer e quindi almeno pari a T_s . **Proporzionale al numero di ingressi N_{AI}**
- I campioni riferiti tutti al medesimo istante temporale.

Struttura sistemi DAQ: campionamento e conversione A/D

HARDWARE TIMING:

→ gestione fornita dal **microprocessore** interno al modulo DAQ

→ Segnali digitali inviati da un clock fisico



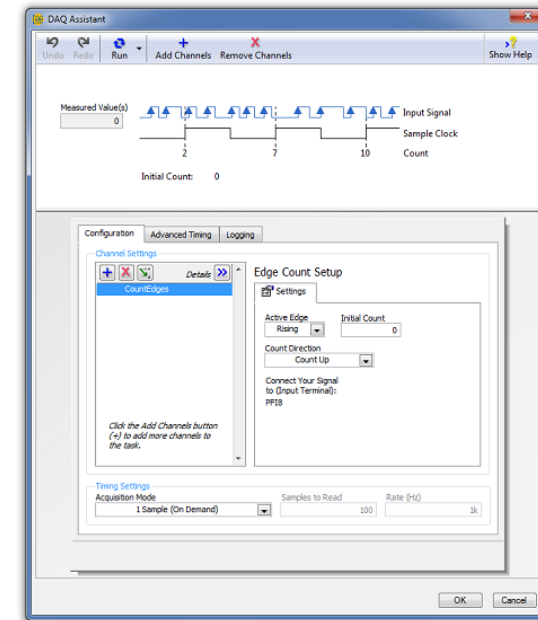
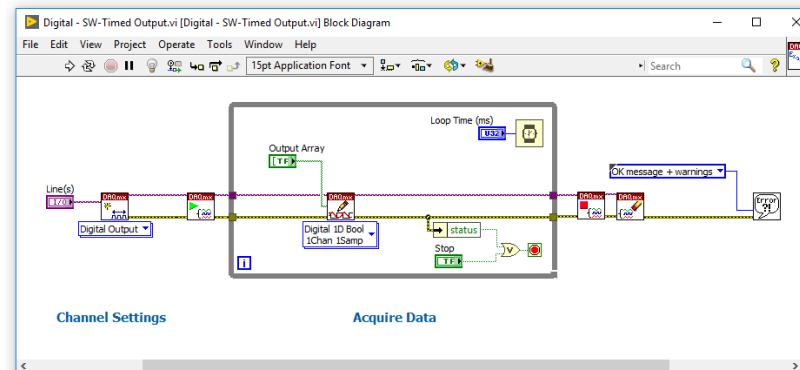
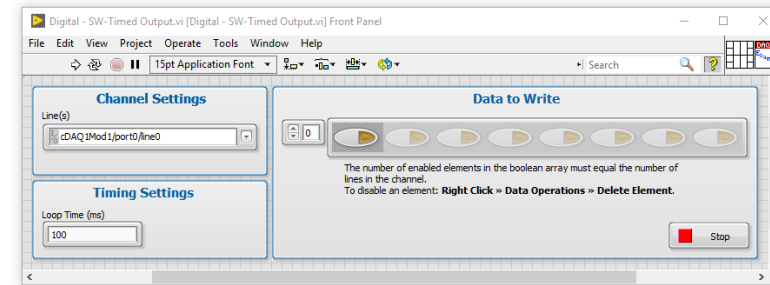
μP → $T_S > N_{AI} \cdot T_{ADC}$

μP → $f_{MUX} = N_{AI} \cdot f_S$

SOFTWARE TIMING:

→ gestione dell'acquisizione dei campioni tramite **applicazione (o utente)**.

→ comando di lettura inviato tramite software e sistema operativo del pc e ricevuto attraverso l'interfaccia di comunicazione.



SOFTWARE TIMING molto più lento a causa delle latenze dello scambio di messaggi. Da preferire **HARDWARE TIMING** soprattutto nel caso di rapide acquisizioni multicanale

Outline

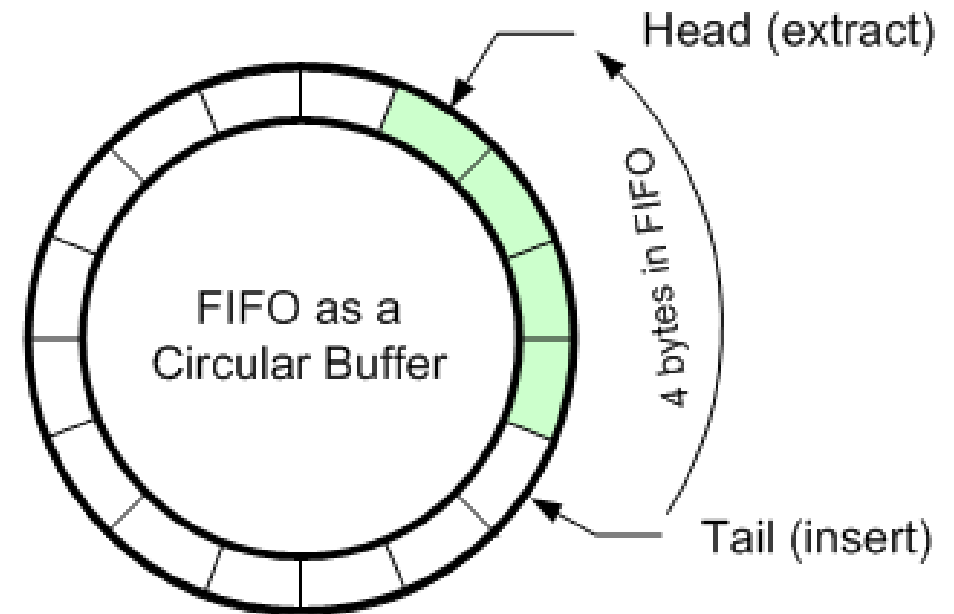
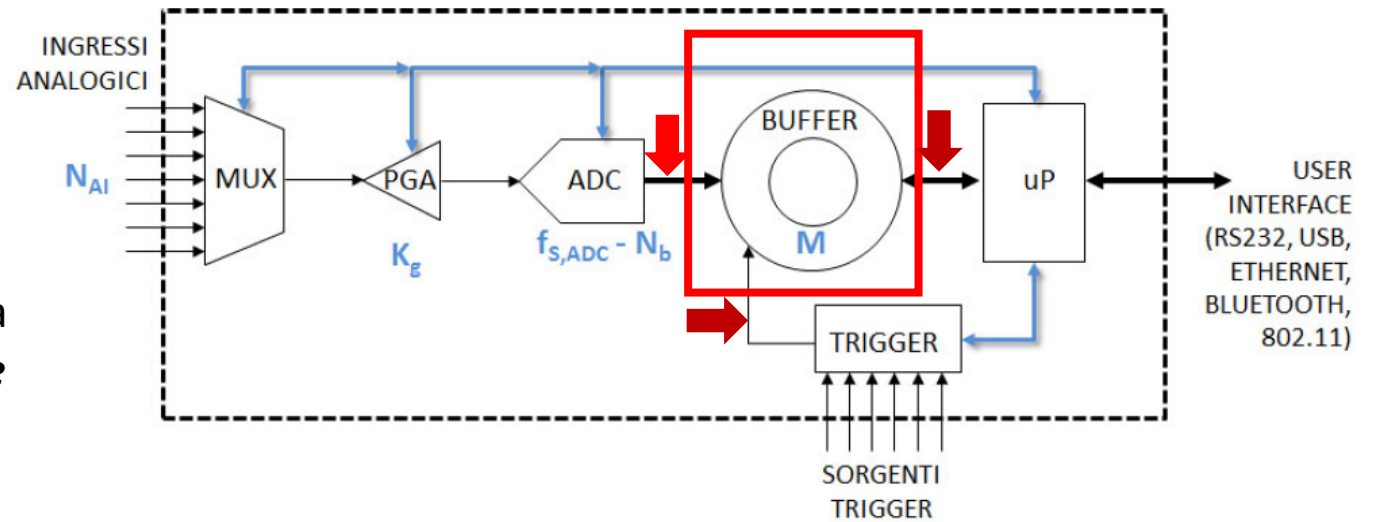
- Generalità sui sistemi DAQ
- Selettore di ingressi e PGA
- Conversione A/D
- Gestione della memoria
- Modalità di acquisizione e interfaccia utente
- Esempio di scheda di acquisizione

QUIZ



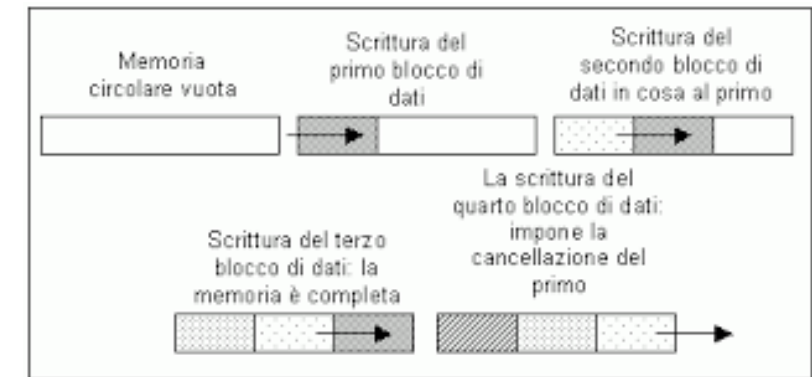
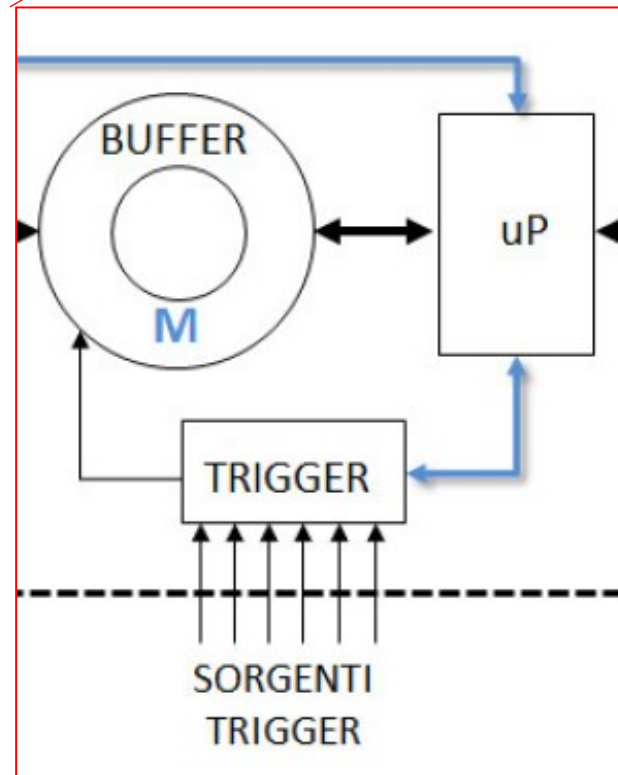
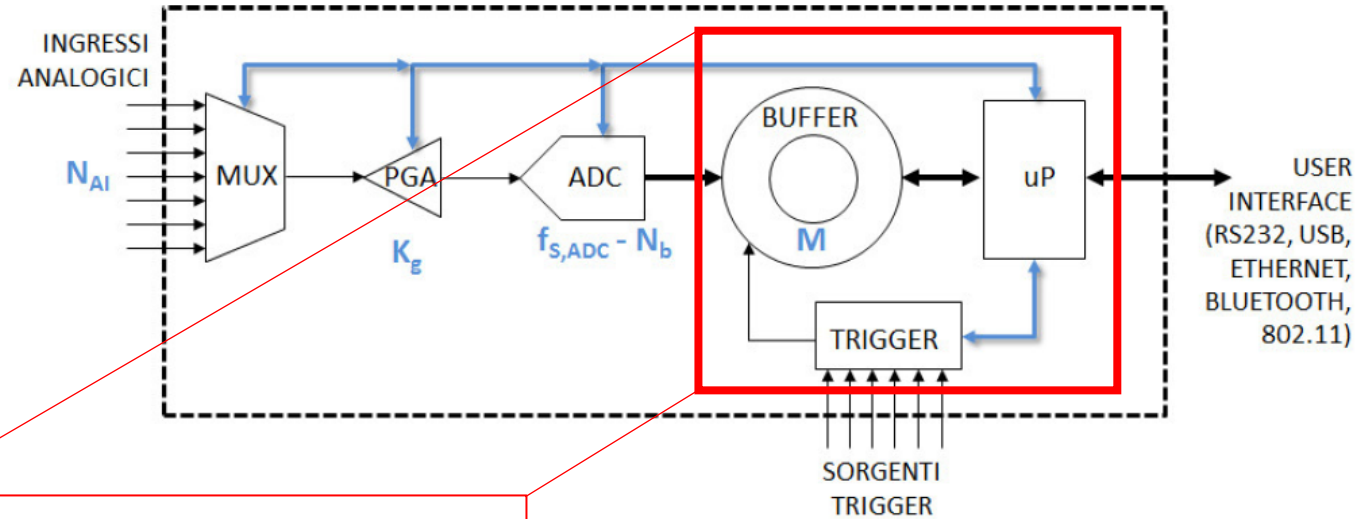
Struttura sistemi DAQ: buffer di memoria

- ❖ **Buffer, o memoria di acquisizione**, ossia un insieme di celle di memoria in grado di contenere **fino ad M campioni** del segnale per memorizzare temporaneamente le parole digitali in uscita dal convertitore.
- ❖ Può essere considerata una **memoria di transito**, detta anche intermediaria, usata per **compensare differenze di velocità nel trasferimento** o nella trasmissione di dati
- ❖ Scelta della **dimensione** fondamentale in base all'applicazione (tipicamente varia **da poche migliaia di punti** (es. schede DAQ) ad **alcune centinaia di milioni di punti** (es. Strumenti ad alta risoluzione))
- ❖ La memoria più utilizzata è quella di tipo **FIFO (First In First Out)** ed è costantemente mantenuta piena, eliminando ad ogni nuovo campionamento il dato più vecchio per far posto all'ultimo. Per questo la memoria è detta anche **buffer circolare**.



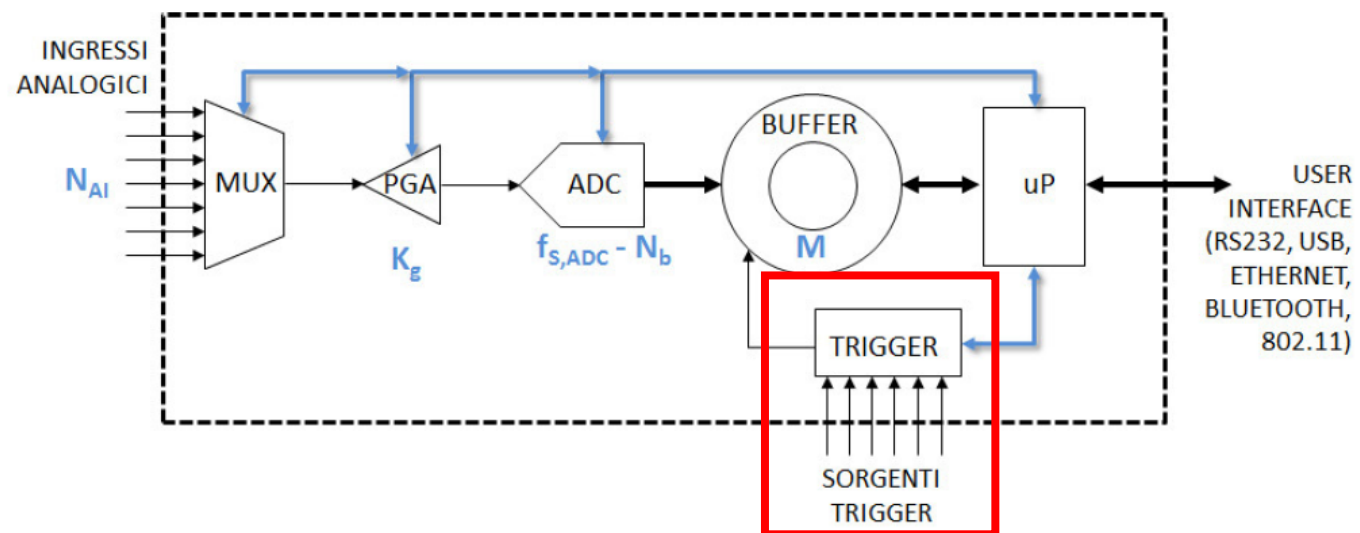
Gestione della memoria nei sistemi DAQ

- ❖ **Microprocessore (μP)** interagisce con il buffer, inviando **segnali di start** e **ricevendo dal buffer i campioni** e li elabora al fine di fornire il risultato finale della misura, che può essere:
 - 1) **trasmesso ad altri dispositivi** esterni attraverso un'interfaccia di comunicazione standard.
 - 2) **trasmesso in formato grezzo alla memoria di sistema del dispositivo che controlla** il modulo DAQ (ad esempio, un PC), dove vengono successivamente elaborati.
- ❖ **Blocco di Trigger** che riceve i **segnali di sincronizzazione esterni** e li invia al buffer in modo da:
 - 1) **sincronizzare l'inizio della memorizzazione**
 - 2) **gestire al meglio la memoria, separando tra una zona pre e post trigger**



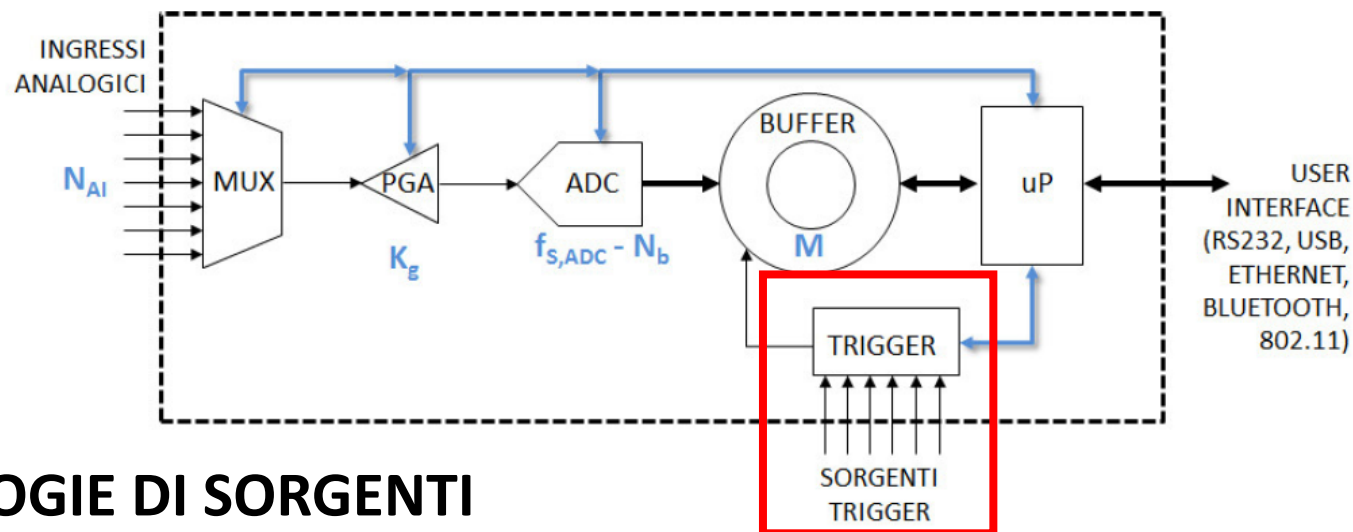
Struttura sistemi DAQ: blocco di trigger

- ❖ **Circuito elettronico che riceve in ingresso segnali di trigger e ha il compito di individuare il verificarsi della condizione di trigger necessaria per avviare l'acquisizione**



Struttura sistemi DAQ: blocco di trigger

- ❖ **Circuito elettronico che riceve in ingresso segnali di trigger e ha il compito di individuare il verificarsi della condizione di trigger necessaria per avviare l'acquisizione**

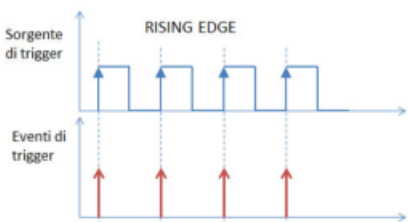


CHE TIPOLOGIE DI SORGENTI DI TRIGGER ESISTONO?

Sorgenti di trigger digitali

TRIGGER DA SEGNALE DIGITALE

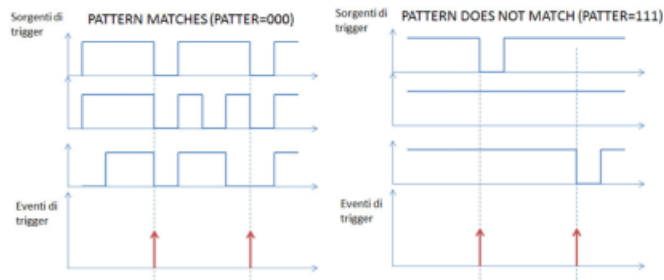
sorgente di trigger è un segnale digitale, ovvero un segnale a **due livelli logici**. Gli eventi di trigger, che danno inizio ad una nuova acquisizione, possono essere generati in corrispondenza del fronte di salita (**rising edge**) oppure di discesa (**falling edge**) del segnale di trigger.



TRIGGER DA PAROLA DIGITALE

sorgente di trigger è definita su più sorgenti di tipo digitali. L'evento di trigger corrisponde alla **rilevazione di una determinata parola binaria** (ossia, una combinazione (pattern) di bit), oppure al **cessare di questa parola**.

Nel primo caso l'evento viene denominato **pattern matches**, nel secondo caso l'evento, denominato **pattern does not match**.

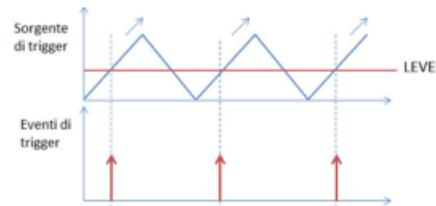


Sorgenti di trigger analogiche

1. RISING/FALLING EDGE

Il criterio più comune per la generazione degli eventi di trigger consiste nell'imporre:
 → una **condizione sul livello del segnale**
 → una **condizione sul verso della pendenza del segnale**.

Il trigger viene generato **quando il segnale passa per il livello specificato con la desiderata pendenza** (positiva o negativa).



Es. RISING EDGE

2. ENTERING/LEAVING WINDOW

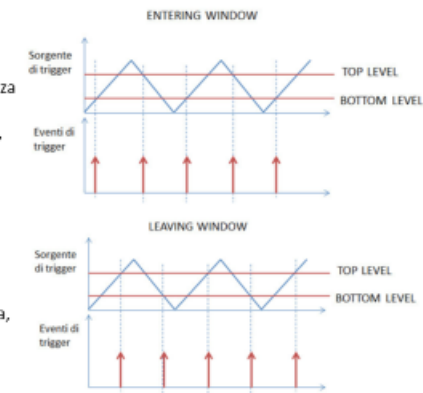
Un secondo metodo per la generazione di eventi di trigger da un segnale analogico consiste **nell'impostare gli estremi inferiore (bottom level) e superiore (top level) di una fascia di valori, o finestra**.
Si possono avere quindi due tipi di trigger:

→ ENTERING WINDOW

quando il segnale, inizialmente di ampiezza inferiore all'estremo inferiore della finestra, entra nell'intervallo di valori specificato.

→ LEAVING WINDOW

quando il segnale, partendo da una ampiezza iniziale superiore all'estremo superiore della finestra, esce dall'intervallo di valori specificati.

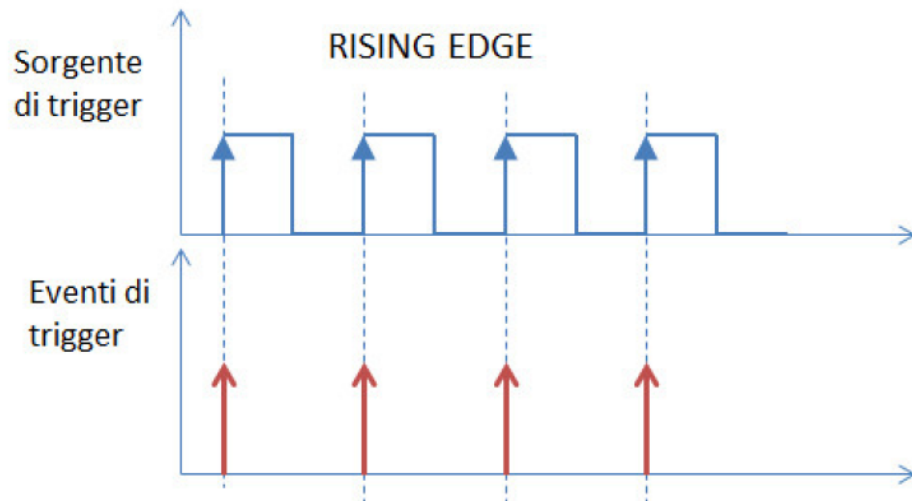


Sorgenti di trigger digitali

TRIGGER DA SEGNALE DIGITALE

sorgente di trigger è un segnale digitale, ovvero un segnale a **due livelli logici**.

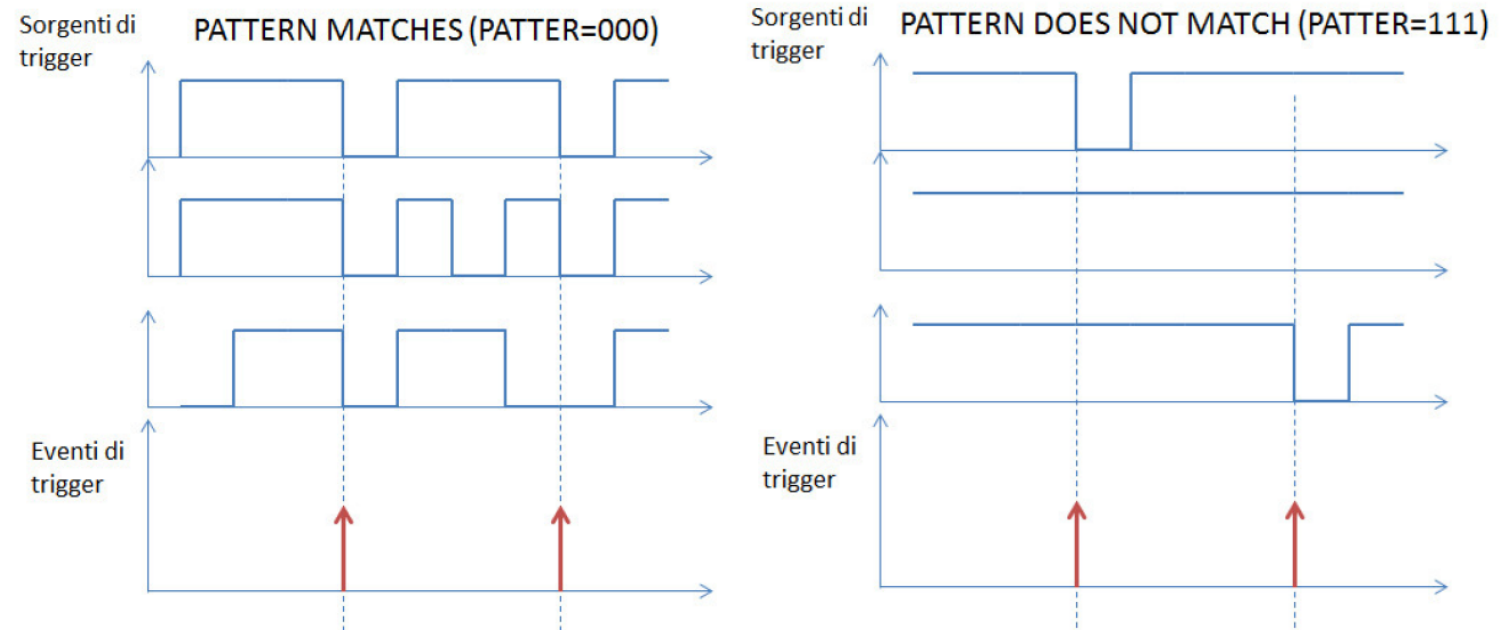
Gli eventi di trigger, che danno inizio ad una nuova acquisizione, possono essere generati in corrispondenza del fronte di salita (**rising edge**) oppure di discesa (**falling edge**) del segnale di trigger.



TRIGGER DA PAROLA DIGITALE

sorgente di trigger è definita su più sorgenti di tipo digitali. L'evento di trigger corrisponde alla **rilevazione di una determinata parola binaria** (ossia, una combinazione (pattern) di bit), oppure al **cessare di questa parola**.

Nel primo caso l'evento viene denominato **pattern matches**, nel secondo caso l'evento, denominato **pattern does not match**.



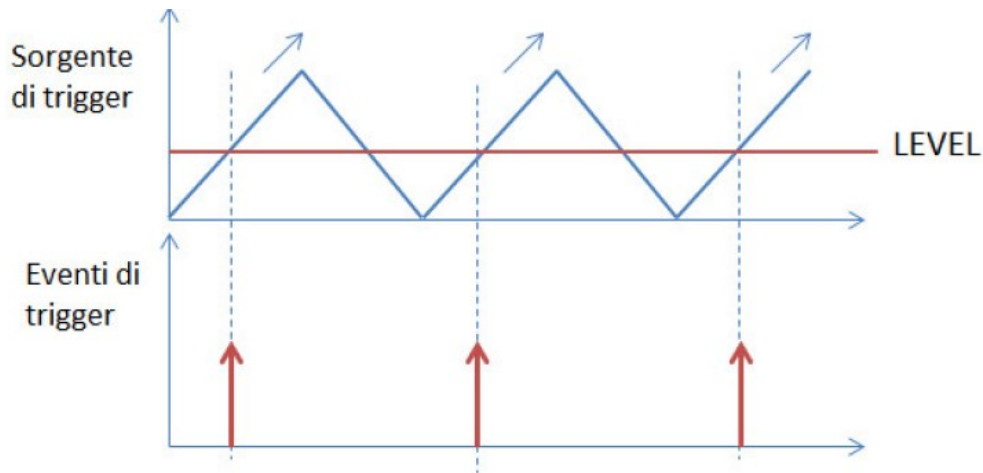
Sorgenti di trigger analogiche

1. RISING/FALLING EDGE

Il criterio più comune per la generazione degli eventi di trigger consiste nell'imporre:

- una condizione sul livello del segnale
- una condizione sul verso della pendenza del segnale.

Il trigger viene generato **quando il segnale passa per il livello specificato con la desiderata pendenza** (positiva o negativa).



Es. RISING EDGE

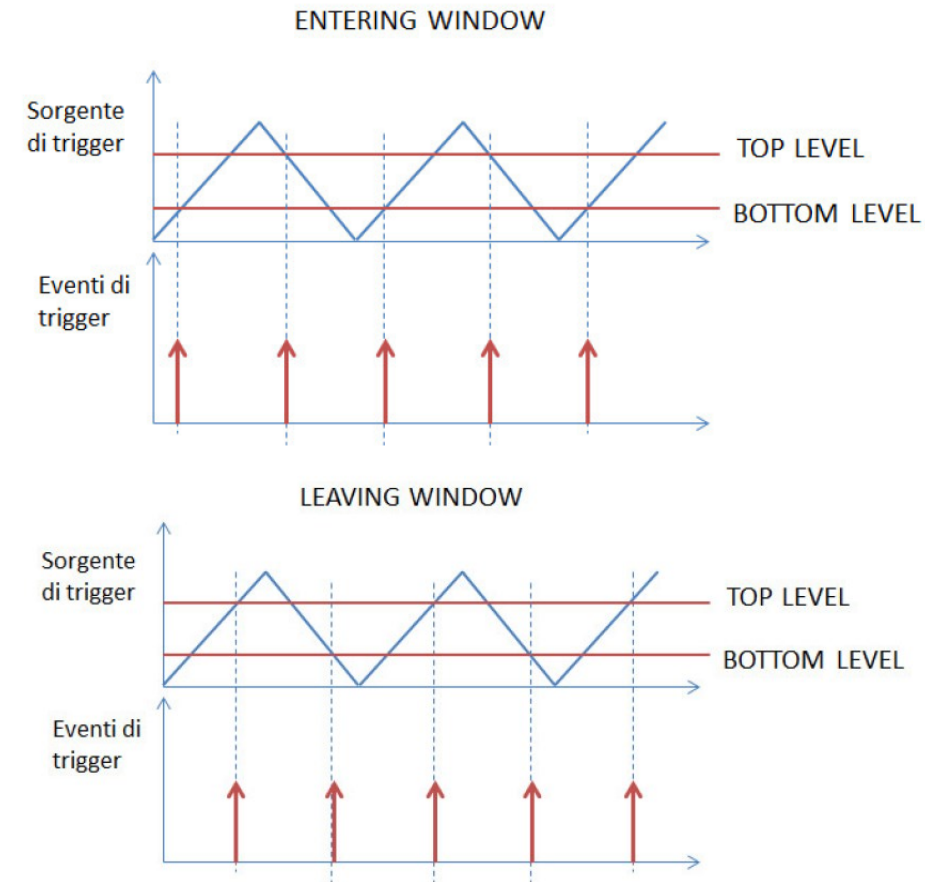
2. ENTERING/LEAVING WINDOW

Un secondo metodo per la generazione di eventi di trigger da un segnale analogico consiste nell'impostare gli **estremi inferiore (bottom level) e superiore (top level) di una fascia di valori, o finestra**.

Si possono avere quindi due tipi di trigger:

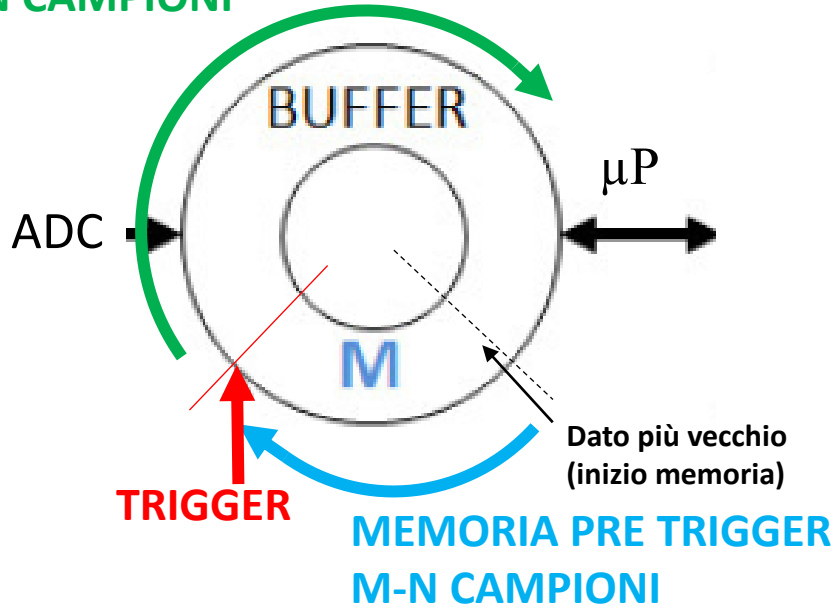
→ **ENTERING WINDOW**
quando il segnale, inizialmente di ampiezza inferiore all'estremo inferiore della finestra, entra nell'intervallo di valori specificato.

→ **LEAVING WINDOW**
quando il segnale, partendo da una ampiezza iniziale superiore all'estremo superiore della finestra, esce dall'intervallo di valori specificati.



Gestione della memoria nei sistemi DAQ

MEMORIA POST TRIGGER
N CAMPIONI



TRIGGER: come influenza la memoria?

DATI ACQUISITI PRIMA DELL'EVENTO TRIGGER

→ **MEMORIA PRE-TRIGGER:** gestita come un buffer circolare dove, una volta riempite tutte le locazioni disponibili, i campioni più vecchi vengono sovrascritti con i nuovi senza errore di overflow.

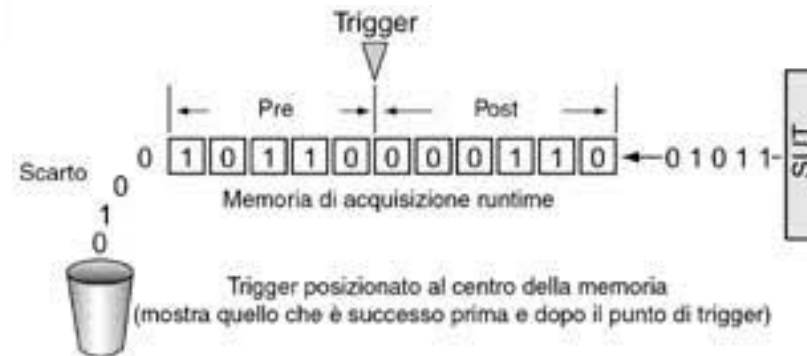
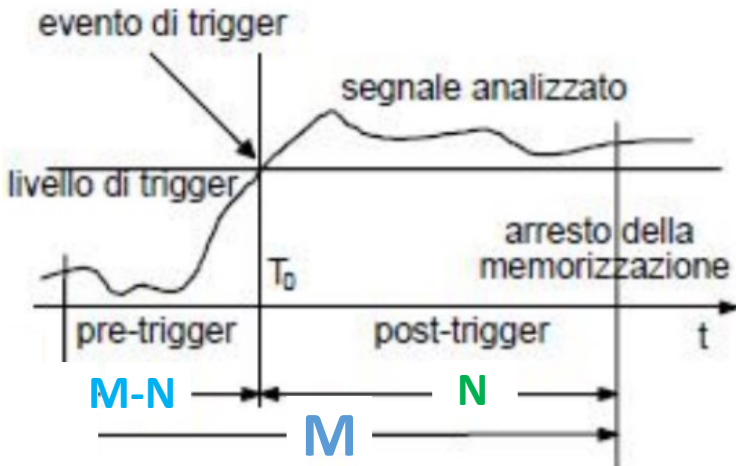
DATI ACQUISITI DOPO EVENTO TRIGGER

→ **MEMORIA POST-TRIGGER:** gestita con segnalazione di errore di overflow

TRASFERIMENTO DATI: Quando?

- quando il buffer di acquisizione è **completamente riempito**
- ad **intervalli di tempo regolari**, indipendentemente dallo stato di riempimento, valutando la tempistica con cui il buffer è letto per evitare **overflow** (*perdita di informazioni*).

L'intervallo tra due successive letture **deve essere inferiore a:**



$$T_{MAX} = \frac{M}{N_{AI} \cdot f_S}$$

Outline

- Generalità sui sistemi DAQ
- Selettore di ingressi e PGA
- Conversione A/D
- Gestione della memoria
- Modalità di acquisizione e interfaccia utente
- Esempio di scheda di acquisizione

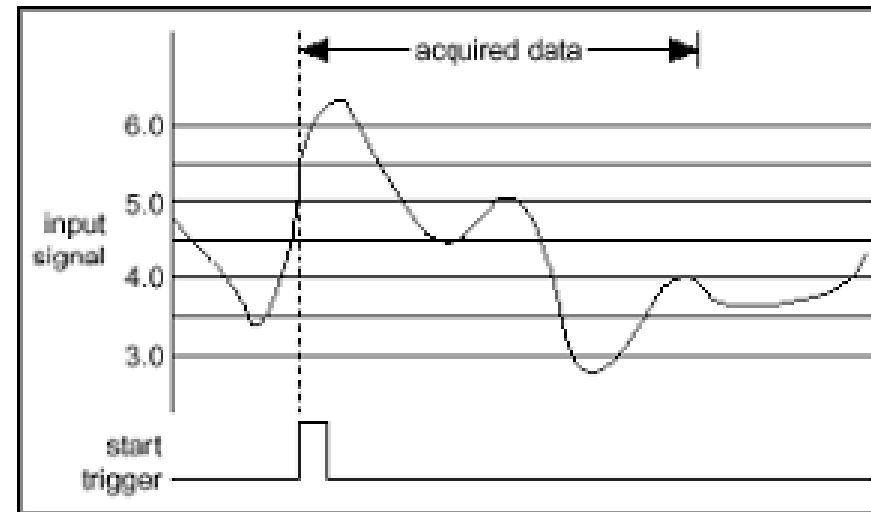
QUIZ



Modalità di acquisizione: innesco e durata

INNESCO ACQUISIZIONE

- 1) Comando **START**, inviato dal μP e ricevuto dal modulo DAQ attraverso l'interfaccia di comunicazione
- 2) Accensione del sistema (**POWER-UP**), appena il modulo viene alimentato ed inizializzato.
- 3) il manifestarsi di un **PARTICOLARE EVENTO**, denominato **trigger**. Questo viene definito ed individuato quando si verificano determinate condizioni su linee di segnale in ingresso al modulo di acquisizione, usate come **sorgenti di trigger**.



DURATA ACQUISIZIONE

ACQUISIZIONE CON DURATA PREDEFINITA (ON-DEMAND)

il processo di acquisizione termina dopo che è stato acquisito un numero prestabilito di campioni.

E' importante notare che il numero di campioni da acquisire potrebbe anche essere superiore alla dimensione **M** del buffer di acquisizione.

ACQUISIZIONE CONTINUA

il processo di acquisizione, una volta iniziato, **ha termine solo in seguito alla disattivazione del sistema**, oppure **all'invio di un comando di reset**, che può essere di tipo hardware oppure software.

Il reset software è inviato al modulo DAQ attraverso l'interfaccia di comunicazione standard.

Modalità di acquisizione: gestione della memoria

$$T_{MAX} = \frac{M}{N_{AI} \cdot f_s}$$

Esempio: *BUFFER DI MEMORIA FIFO* → **2000 CAMPIONI**
FREQUENZA DI CAMPIONAMENTO → **1000 SAMPLES/s**
FREQUENZA FONDAMENTALE DEL SEGNALE → **100 Hz**

$$T_s = 1 \text{ ms}$$

$$T_{segnale} = 10 \text{ ms}$$

ACQUISIZIONE CON DURATA PREDEFINITA (ON-DEMAND)

Durata prestabilita: $T_{lettura} = 30 \text{ ms}$



SALVATO CIO' CHE RICHIESTO

Totale campioni: $T_{lettura} * f_s = 0.03 \text{ s} * 1000 \text{ SAMPLES/s} = 30 \text{ SAMPLES}$

Durata prestabilita: $T_{lettura} = 3 \text{ s}$



SALVATI ULTIMI 2000 SAMPLES

Totale campioni: $T_{lettura} * f_s = 3 \text{ s} * 1000 \text{ SAMPLES/s} = 3000 \text{ SAMPLES}$

3 s 2000 samples

$$T_{MAX} > \frac{M}{N_{AI} \cdot f_s}$$

1000 samples/s

ACQUISIZIONE CONTINUA

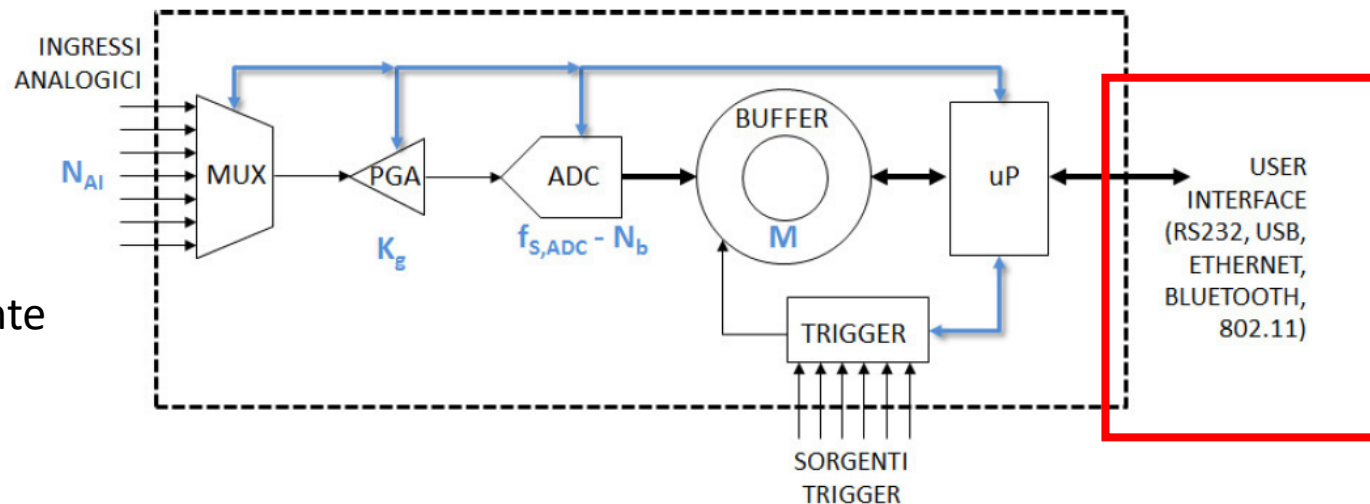
→ Fornisco solo avvio

→ Al momento in cui do lo STOP o il RESET nella memoria mi troverò gli **ultimi 2000 samples**

Struttura sistemi DAQ: interfaccia utente

❖ **Interfaccia di comunicazione utente**, attraverso cui si può:

- 1) **Programmare** la scheda di acquisizione
- 2) **Trasmettere e ricevere il risultato** della misura o i campioni grezzi acquisiti dai segnali di ingresso, durante la fase operativa.



Programming With NI MyDAQ

> Program Data Acquisition

8. Acquire Digital Data

This script example shows how to read digital data on myDAQ digital channels:

```
%Acquire Digital Data
d = daq.getDevices
%Create a session object and save it to the variable, s:
s = daq.createSession('ni')
%Add two digital input channels (lines 0 & 1)
%Note here two lines are specified as input only (read only)
addDigitalChannel(s,'myDAQ1','port0/line0','InputOnly')
addDigitalChannel(s,'myDAQ1','port0/line1','InputOnly')
%Acquire digital data:
Ddata = inputSingleScan(s)
```

The collected data shown in command window may look like this:

```
Ddata =
0      1
>>
```

line 0 data
line 1 data



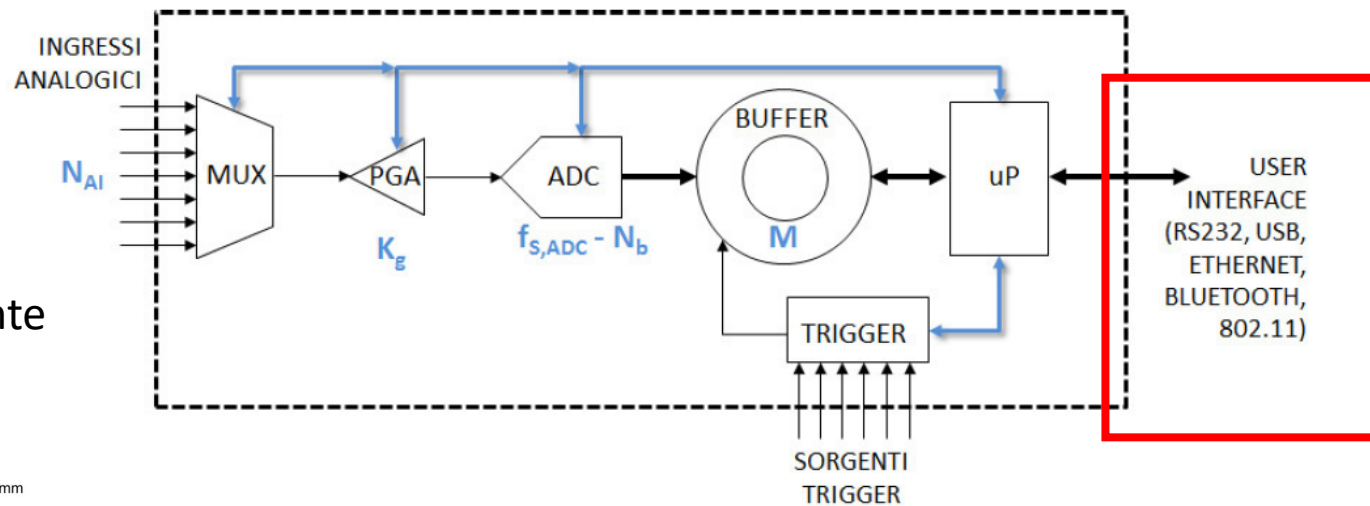
→ μ P controllato tramite vari linguaggi di programmazione (Matlab, Labview ecc)

→ Generata spesso apposita **interfaccia** in modo che utente che esegue le misure possa selezionare le scale e **adattare i parametri alla specifica applicazione e visualizzare risultati**

Struttura sistemi DAQ: interfaccia utente

❖ **Interfaccia di comunicazione utente**, attraverso cui si può:

- 1) **Programmare** la scheda di acquisizione
- 2) **Trasmettere e ricevere il risultato** della misura o i campioni grezzi acquisiti dai segnali di ingresso, durante la fase operativa.

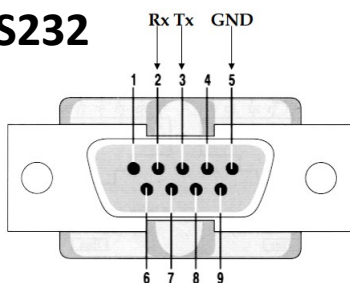


Se modulo DAQ dispositivo autonomo:

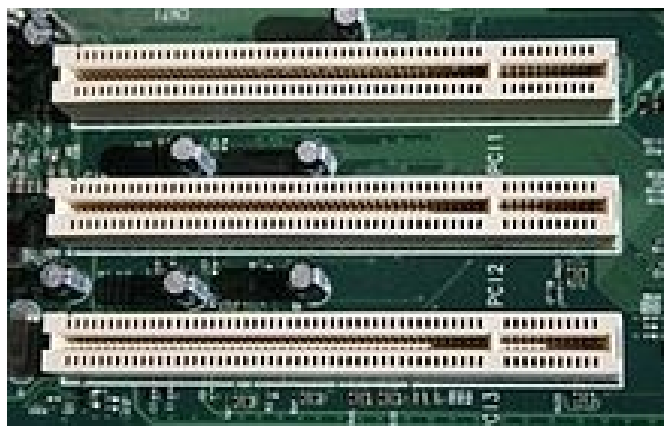
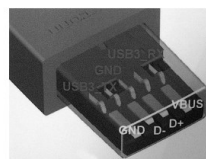
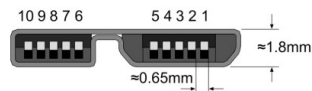
l'interfaccia, gestita anch'essa dall'unità centrale, può utilizzare un protocollo quale, ad esempio, **RS232, USB, Ethernet, Bluetooth, IEEE 802.11.**



RS232



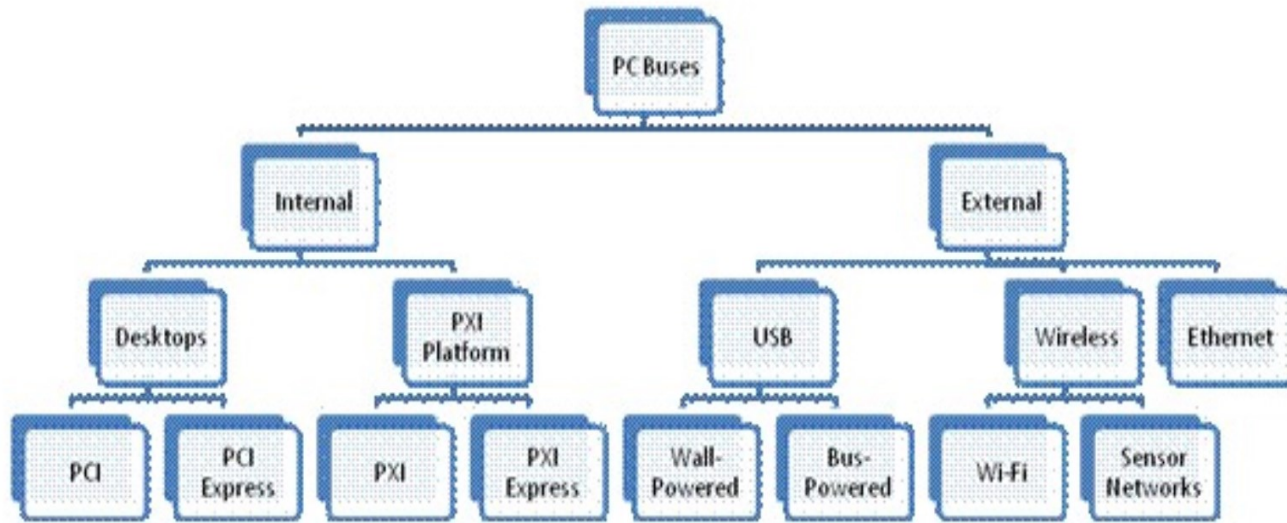
USB



Se il modulo DAQ è interno ad un PC o parte di un sistema modulare:

la connessione può essere realizzata da un **bus interno** (ad esempio, **Peripheral Component Interconnect (PCI)** o interconnessione di componente periferica)

Struttura sistemi DAQ: trasmissione dati



5 domande per la scelta del giusto bus:

1. Con che segnali avrò a che fare? Quanti dati dovrò memorizzare?
2. Ho bisogno di uscite sia input che output?
3. Devo sincronizzare molteplici device?
4. Deve essere portatile tutto il mio sistema?
5. Quanto lontane saranno le misure dal mio pc?

Bus	Waveform ¹ Streaming	Single-Point I/O	Multidevice	Portability	Distributed Measurements
PCI	132 MB/s (shared)	Best	Better	Good	Good
PCI Express	250 MB/s (per lane)	Best	Better	Good	Good
PXI	132 MB/s (shared)	Best	Best	Better	Better
PXI Express	250 MB/s (per lane)	Best	Best	Better	Better
USB	60 MB/s	Better	Good	Best	Better
Ethernet	125 MB/s (shared)	Good	Good	Best	Best
Wireless	6.75 MB/s (per 802.11g channel)	Good	Good	Best	Best

Outline

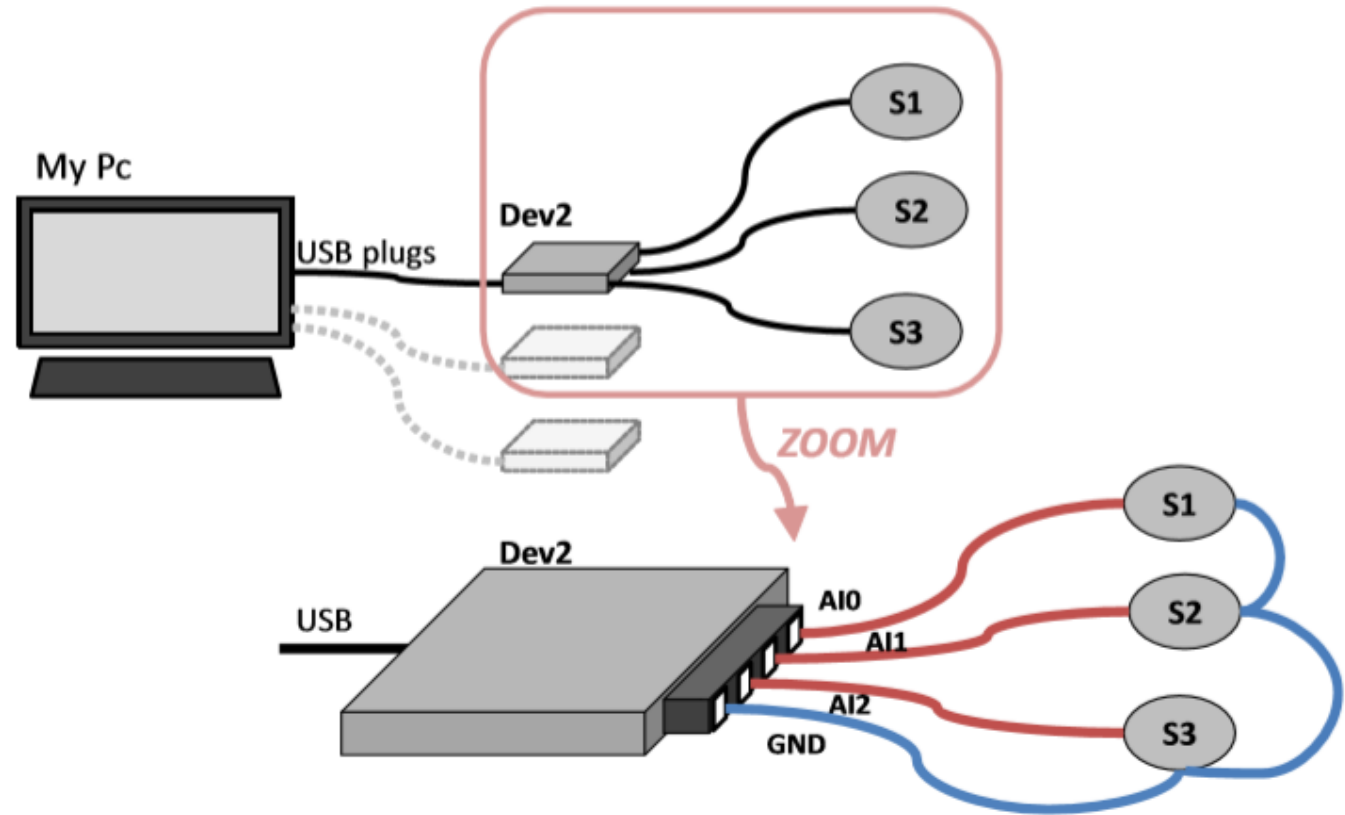
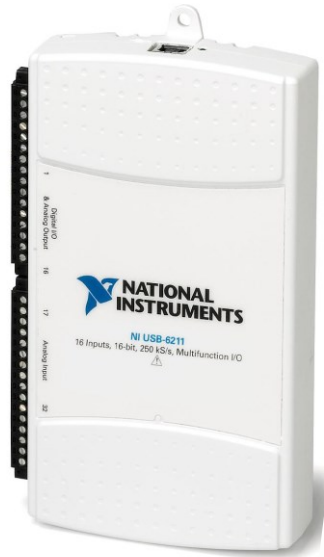
- Generalità sui sistemi DAQ
- Selettore di ingressi e PGA
- Conversione A/D
- Gestione della memoria
- Modalità di acquisizione e interfaccia utente

QUIZ



- Intro pratica lab con schede di acquisizione (24/11 e 1/12)

Esempio di schede DAQ: moduli National Instrument



Manuale National Instrument:

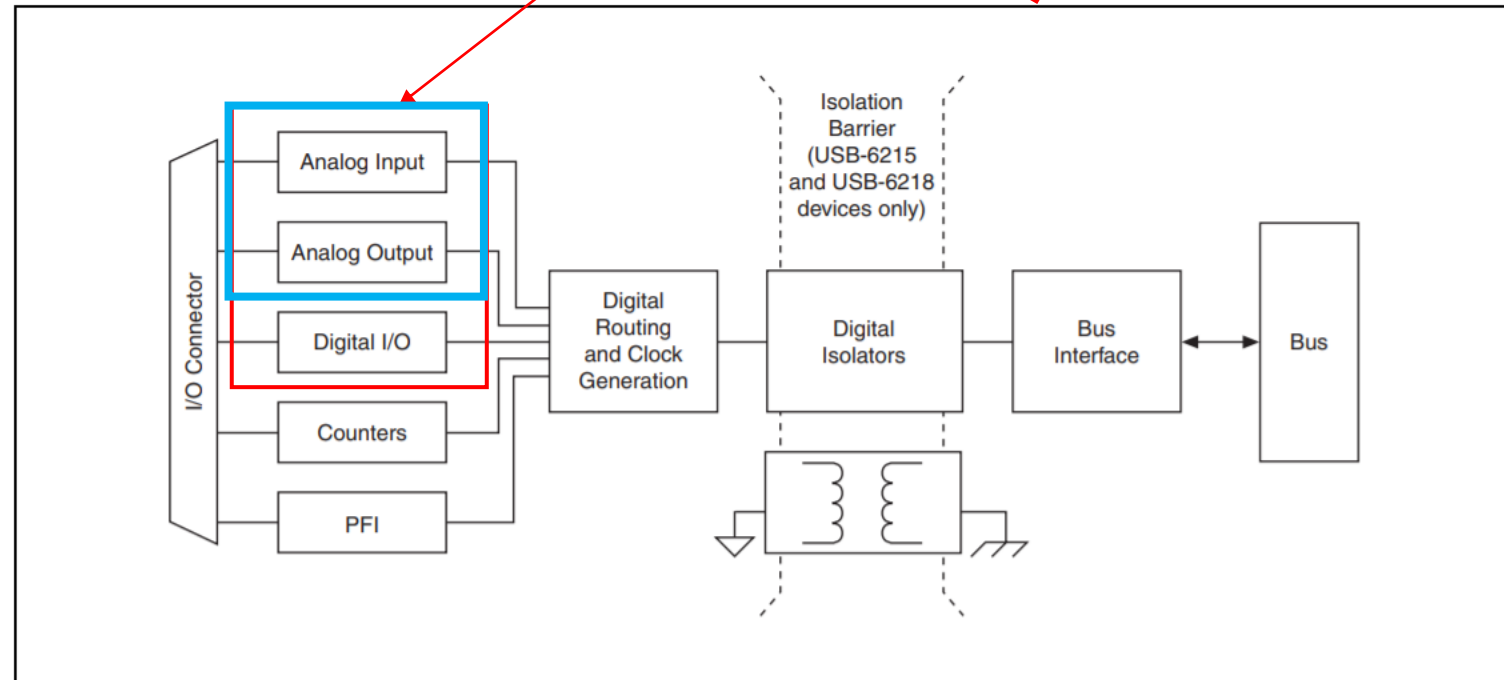
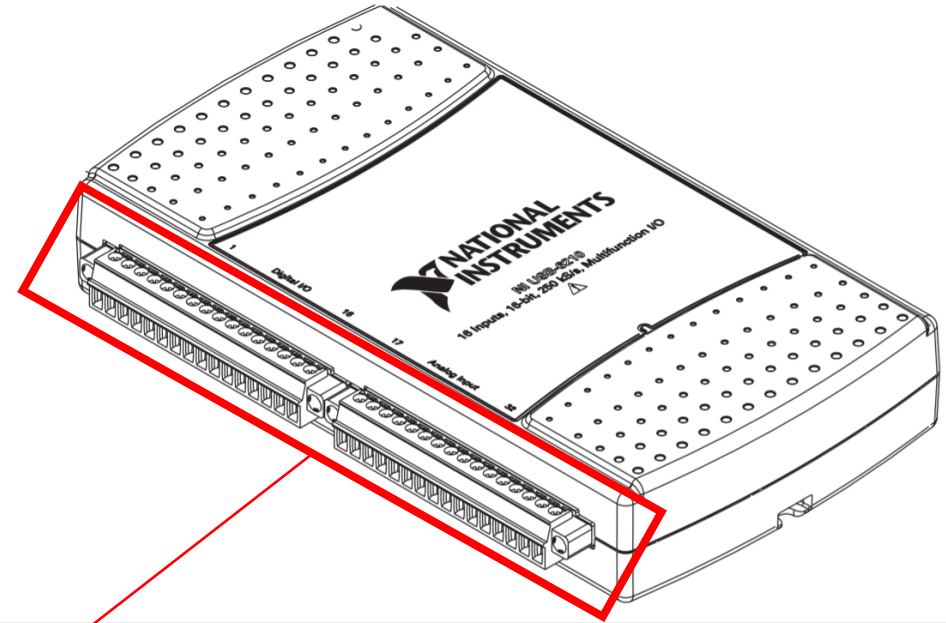
<http://www.ni.com/pdf/manuals/375195d.pdf>,
<https://www.ni.com/pdf/manuals/371931f.pdf>

Indicazioni introduttive Matlab:

<https://it.mathworks.com/help/daq/examples/getting-started-with-session-based-interface-using-ni-devices.html>

Esempio di DAQ 1: moduli NI multi ingresso

- **NI 621x devices presentano fino a:**
 - 32 canali per ingressi analogici (AI)
 - 2 canali per output analogici (AO)
 - 8 linee per input digitali (DI)
 - 8 linee per output digitali (DO)
 - 2 contatori
- Le principali funzioni di questi moduli DAQ sono **digitalizzare segnali da sensori per consentirne memorizzazione ed elaborazione**, realizzare conversion D/A per generare segnali analogici in output, misurare e controllare con segnali digitali.
- **I segnali spesso necessitano di un condizionamento** per poter essere trattati dal DAQ in un formato adeguato.



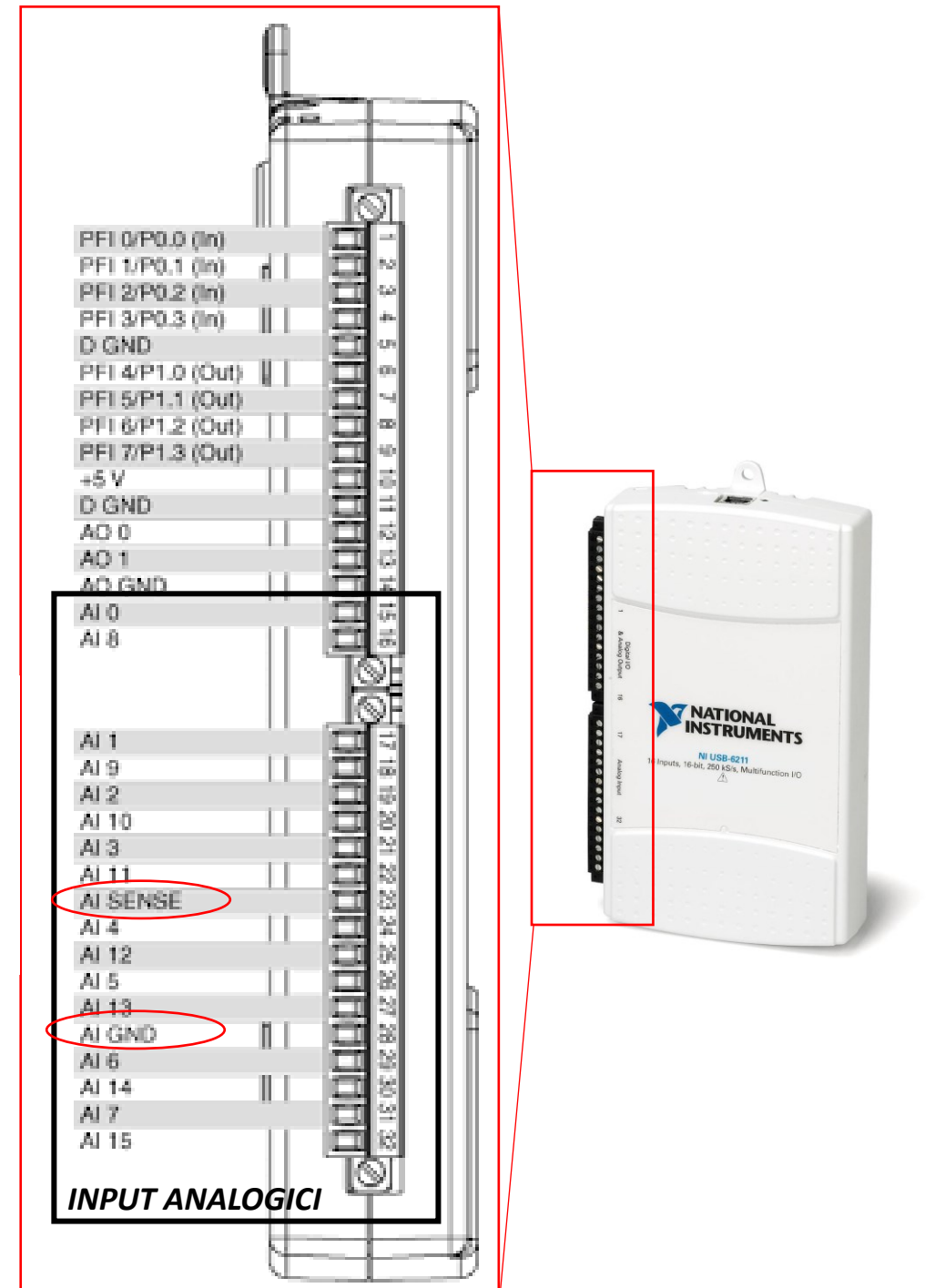
Tipologie di misure con modulo NI

La scheda permette di effettuare tre tipologie di misurazione diverse per gli ingressi analogici:

- **Differenziale**, permette di misurare la differenza di potenziale tra una coppia di morsetti.
- **Referenced Single-End Mode**, misura la differenza di potenziale tra un morsetto e quello di massa.
- **Non-Referenced Single-End Mode**, misura la differenza di potenziale tra un morsetto e l' AI SENSE input.

AI Ground-Reference Settings	Signals Routed to the Positive Input of the NI-PGIA (V_{in+})	Signals Routed to the Negative Input of the NI-PGIA (V_{in-})
RSE	AI <0..15>	AI GND
NRSE	AI <0..15>	AI SENSE
DIFF	AI <0..7>	AI <8..15>

N.B. Attenzione sempre alle misure fatte in configurazione RSE per il rischio dell'anello di terra se il segnale che state misurando è una sorgente riferita: sempre preferire NRSE o DIFF mode.



Esempio di DAQ 2: moduli NI per circuiti semplici

NI myDAQ

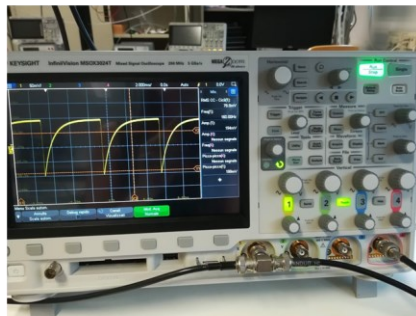
- dispositivo di **acquisizione/generazione** dati che dispone di:
- **due canali analogici differenziali** di ingresso
 - **due di uscita** (200 ks/s, 16 bit, +/- 10 Volt)
 - otto linee digitali di ingresso e uscita (compatibili TTL a 3,3 Volt)
 - un multimetro digitale (DMM) a 60 Volt per misure di tensione, corrente e resistenza.
- Alimentato tramite bus USB



Obiettivi delle prossime esercitazioni



MATLAB®
The Language of Technical Computing



Obiettivo 1: Acquisizione di segnali da generatore

LABORATORIO 24/11/2021

Obiettivo 2: Generazione di segnali

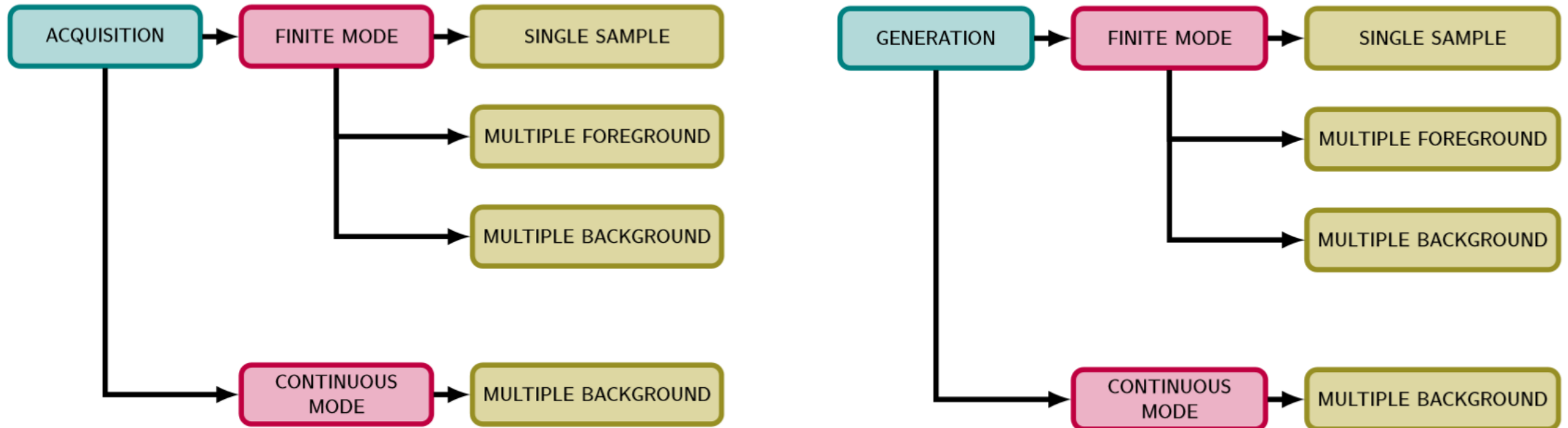
LABORATORIO 01/12/2021



Cosa si può fare con il modulo DAQ NI ?

ACQUISIRE o GENERARE SEGNALI in due modalità:

- in **foreground** (in primo piano): l'esecuzione della funzione di acquisizione sarà esclusiva, ossia il sistema sarà unicamente dedicato all'attività di acquisizione o generazione, disponibile solo in modalità finita.
- in **background** (sullo sfondo), cioè poter eseguire acquisizione mentre si può eseguire altre operazioni, ma mantenendone il controllo per potergli comunicare qualcosa o per poterlo fermare.



OBIETTIVO DELLE ESERCITAZIONI CHE FAREMO SARA' APPRENDERE L'UTILIZZO DELLA MODALITA' IN FOREGROUND. PER CHI COMUNQUE SAPESSSE GIA' UTILIZZARE QUESTA TIPOLOGIA DI SCHEDE OPPURE PER EVENTUALI NECESSITA' DI TESI VI MOSTRERO' ANCHE LA MODALITA' IN BACKGROUND.

Step essenziali per programmare il modulo DAQ

1) Come viene **ricosciuto il device da Matlab**? Quanti device sono connessi? Che caratteristiche hanno?

```
devices = daq.getDevices
```

2) **Dove salvo i dati** che acquisirò? Come interagisco con il DAQ?

```
session = daq.createSession(vendor)
```

3) Come indico a Matlab **quali canali ho collegato**? Come specifico se avrò input o output? Come specifico la tipologia di misura e di configurazione?

```
ch = s.addAnalogInputChannel(deviceName, channelID, measurementType);
```

Oppure

```
ch = s.addAnalogOutputChannel(deviceName, channelID, measurementType);
```

4) Come specifico i **parametri di acquisizione/generazione**: quanti campioni voglio acquisire, a che frequenza? Acquisizione/generazione singola o multipla?

```
s.Rate=Fs; s.NumberOfScans=Nsamples; s.DurationInSeconds=Tw;
```

5) Come **da il via all'acquisizione/generazione** dei segnali?

```
[data,time,triggerTime]=s.startForeground()
```

Oppure

```
queueOutputData(s, outputData); startForeground(s);
```

RICONOSCERE IL DEVICE

CREARE UNA SESSIONE
DI LAVORO

AGGIUNGERE I CANALI
INTERESSATI

STABILIRE I PARAMETRI DI
ACQUISIZIONE/GENERAZIONE

ACQUISIRE/GENERARE I
CAMPIONI

Take home messages

GENERALITÀ SUI SISTEMI DAQ

- Le **schede di acquisizione** sono uno strumento particolarmente utile in quanto caratterizzate da: flessibilità, adattabilità al variare dei requisiti richiesti, facilità di reimpiego, possibilità di ridurre i tempi per progettazione e realizzazione del sistema di misura.

SELETTORE DI INGRESSI E PGA

- I primi blocchi prima della conversione sono: il **multiplexer**, utilizzato per selezionare sequenzialmente i segnali in ingresso in modo da trattarli singolarmente, e il **programmable gain amplifier**, necessario per adattare il range all'ingresso dell'ADC.

CONVERSIONE A/D

- Il **blocco di conversione A/D** include **S/H** e **ADC** necessari per consentire la conversione analogico digitale. Dal punto di vista della sezione ADC due sono le configurazioni possibili: acquisizione multicanale **non simultanea** con single S/H o acquisizione **simultanea** dei canali con tanti S/H quanti segnali in ingresso. Nel secondo caso aumento complessità circuitale e tempo H.

GESTIONE DELLA MEMORIA

- La **gestione della memoria** all'interno del DAQ è resa possibile dall'interazione tra tre blocchi: **buffer di memoria** (storage temporaneo), **blocco di trigger** (segnale di sincronismo, digitale o analogico) e **microprocessore** (per consentire la temporizzazione di tutti i blocchi presenti).

MODALITÀ DI ACQUISIZIONE E INTERFACCIA UTENTE

- Le due **modalità di acquisizione** possibili sono: quella **a tempo predefinito on demand** (attenzione a compatibilità tempo osservato con memoria) e **quella continua fino a disattivazione**. L'acquisizione può essere effettuata dal comando di start, dall'accensione o a seguito di specifici segnali trigger.
- Le **modalità differenti possono essere gestite tramite delle apposite interfacce, distinte in: interfaccia utente** (per programmare e ricevere output) e **interfaccia di trasmissione** (per inviare i dati per successiva elaborazione).