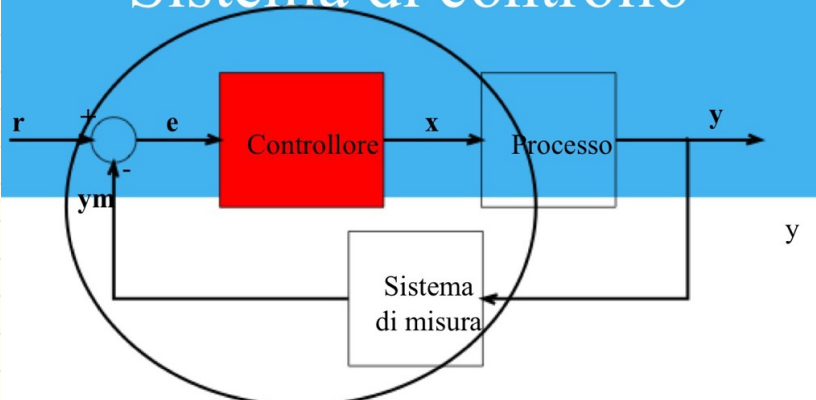


Sistema di controllo



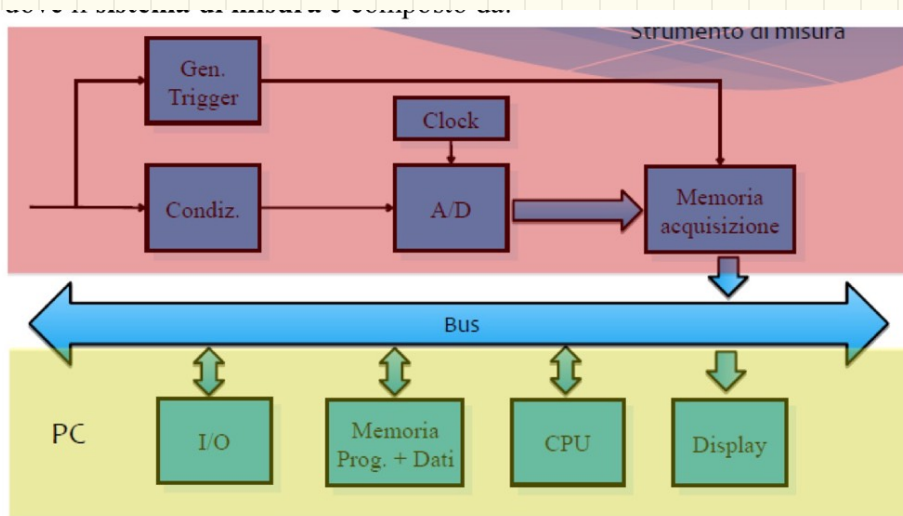
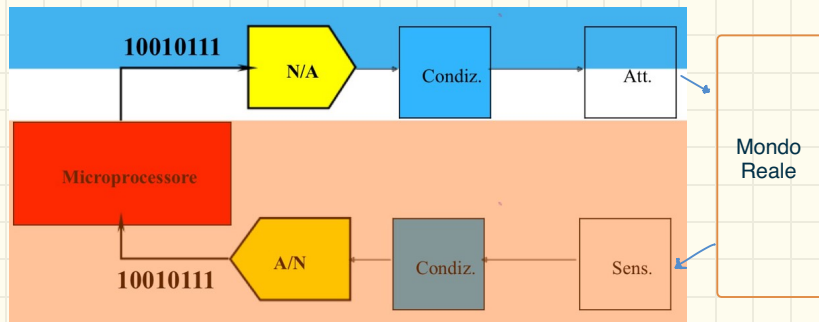
Sistemi automatici di misura

Abbiamo un vero e proprio controllo in retroazione, dove misuriamo l'uscita e tramite un controllore andiamo a pilotare il sistema.

Lo scopo della misurazione è quella di fornire una valutazione delle grandezze fisiche di interesse in maniera oggettiva (quantitativa).

Se la nostra misura è sbagliata, inevitabilmente il nostro controllo sarà sbagliato. L'accuratezza della misura incide su quella del controllo.

Oltre al controllo, lo scopo di una misurazione può anche essere quello di **monitoraggio**, senza prendere alcuna decisione. È un esempio una stazione meteorologica.



Solitamente si aggiunge un trigger che dice alla memoria di acquisire i dati che le arrivano in input. Questo trigger può essere ad esempio la tensione in ingresso che supera una certa soglia.

I dati memorizzati nella memoria vengono poi mandati su un **bus dati** che propaga i dati su altri componenti come una **cpu** che mostra poi i dati su un **display**. Possiamo anche avere un input dall'utente tramite un **I/O**.

Possiamo anche memorizzare i dati su una memoria **più grande**, sulla quale i dati possono rimanere per un lungo periodo (come un hard disk)

Cosa è un sistema automatico di misura?

Un **SAM** è un insieme di strumenti connessi in rete che possono scambiare fra loro messaggi e dati in modo coordinati per eseguire un processo di misurazione

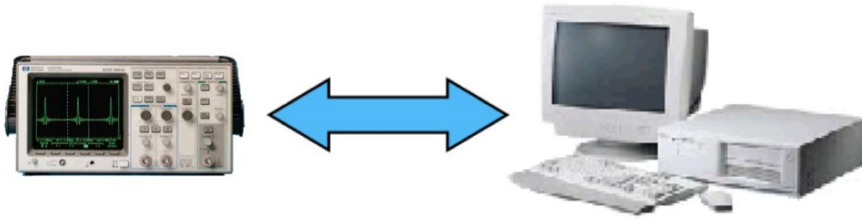
un **esempio** può essere una delle esercitazioni svolte: durante la caratterizzazione del filtro RC abbiamo usato due strumenti connessi ad un unico computer: un **generatore** comandato dal computer; possiamo infatti inviare comandi in modo da generare un input a nostro piacimento (strumento) ed un **oscilloscopio** che inviava i dati sempre al computer.

Andiamo quindi ad implementare un **nuovo strumento** utilizzando più strumenti elementari messi in comunicazione tra loro.

Problema di standardizzazione

Abbiamo un problema di standard: non possiamo usare uno standard diverso per ogni applicazione, è controproducente. Dobbiamo quindi **uniformare le modalità di connessione tra gli strumenti** usando un **sistema di interfaccia mento standard**.

IL COLLO DI BOTTIGLIA



Se volessimo avere una comunicazione in tempo reale tra lo strumento (oscilloscopio) ed un computer per l'elaborazione dei dati (appunto in tempo reale) il **problema maggiore** di velocità sarebbe dato dal **mezzo di comunicazione** e dal suo **protocollo**.

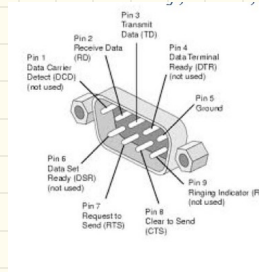
Ovviamente se lo strumento ed il PC si trovano vicini basta collegarli via cavo con un cavo Ethernet e risolveremmo il problema. Non riusciamo a risolverlo però se i due strumenti si trovano **ad elevata distanza** o dobbiamo trasmettere i dati in maniera **wireless** (come ad esempio Wi-Fi e bluetooth).

PROTOCOLLO DI COMUNICAZIONE SERIALE

RS232

- Linea Tx : Trasmissione PC → Strumento
- Linea Rx : Ricezione Strumento → PC
- Linea di MASSA GND
- Linee di HANDSHAKE

} 1 bit alla volta



Vantaggi

- questa interfaccia è presente su tutti i computers. Ad esempio la comunicazione tramite cavo USB è seriale.
- questo protocollo è **standard**, quindi possiamo

Svantaggi

- la velocità di trasmissione è limitata proprio dal mezzo trasmissivo (cavo).
- anche la distanza di comunicazione è limitata dal mezzo (15m)
- tramite una porta seriale possiamo comunicare solo con uno strumento.

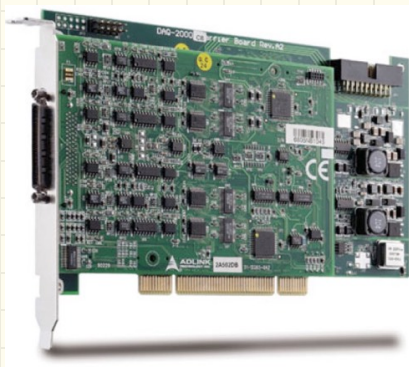
PROTOCOLLO DI COMUNICAZIONE GPIB

IEEE 488

Differenze e vantaggi rispetto all'rs232

- In questo caso abbiamo **8 linee per la trasmissione in parallelo** e possiamo quindi trasmettere **1 Byte** alla volta invece che un singolo bit.
- Inoltre ogni strumento **ha un indirizzo**, e quindi **possiamo comunicare con più di uno strumento** per una singola porta. Possiamo collegare fino a 30 strumenti.
- la velocità di trasmissione è circa 1MB/s

SCHEDE ACQUISIZIONE DATI



Queste schede vengono inserite direttamente **all'interno del computer** tramite la porta PCI express. Queste schede permettono di **trasformare il computer in un vero e proprio sistema di acquisizione dati**, e quindi risolvere il problema della trasmissione, memoria, ecc.

Queste schede vengono "controllate" tramite dei software come **LabView**: è così che è stato possibile sostenere le **esercitazioni a distanza** tramite un'interfaccia web che permetteva di pilotare ed acquisire dati dagli strumenti. LabView ci permette di **creare un'interfaccia grafica** per mostrare diverse informazioni. Possiamo anche effettuare dei calcoli in modo da avere **misure combinate** (ad esempio un ohmetro da voltmetro e amperometro).

Il bus di comunicazione in questo caso è il bus interno di computer (PCI) e quindi abbiamo un'elevatissima velocità di comunicazione. In questo caso il collo di bottiglia si sposta dal mezzo di comunicazione al sistema di acquisizione: potremmo non riuscire ad acquisire dati abbastanza velocemente da "saturare" il bus.

Abbiamo solitamente tutti i tipi di ingresso e uscita: possiamo quindi acquisire tensioni e correnti fornite da tutti i tipi di sensori. Possiamo anche **comunicare con un circuito di condizionamento** per pilotare un attuatore.

MISURE DISTRIBUITE

Si parla di misure distribuite quando andiamo ad effettuare delle misurazioni ad elevata distanza; è proprio il caso delle diverse esercitazioni a distanza che sono state effettuate durante la pandemia.

In questo caso ci serve:

- ovviamente **gli strumenti di misura** che devono però essere **controllabili a distanza** da una rete di PC.
- **computers e software**
- **reti** aventi una banda sufficientemente larga in modo da minimizzare le latenze e massimizzare la velocità di trasmissione.

Nel caso delle esercitazioni la "**parte distribuita**" era l'**interfaccia utente**: era quella la parte accessibile da diversi punti (casa degli studenti).

Potremmo anche avere l'**elaborazione distribuita**: possiamo avere diversi computers che si collegano allo strumento ed **elaborano in parallelo** i dati, in modo da avere una velocità di elaborazione nettamente maggiore.

Anche le **misure** possono essere **distribuite**: potremmo avere diversi sensori sparsi per il mondo ed un'unica unità di elaborazione.

Archivio distribuito: possiamo avere diversi sistemi di archiviazione dislocati in diverse aree geografiche.

SISTEMI STRETTAMENTE ACCOPPIATI

Abbiamo diverse unità di elaborazione che comunicano tra di loro ed elaborano indipendentemente dei dati provenienti da convertitori A/D (dedicati per ogni unità). Abbiamo però una **memoria condivisa** ed un'**unica interfaccia di I/O** con l'utente.

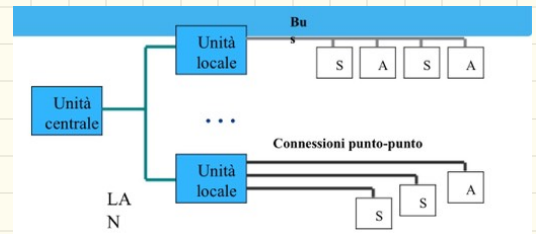
Dobbiamo **sincronizzare i clock dei processori** perché essendo la memoria condivisa, gli accessi devono essere sincronizzati in modo da non avere collisioni.

In poche parole: abbiamo diverse componenti che **condividono delle parti**, come ad esempio la memoria.

SISTEMI DEBOLMENTE ACCOPPIATI

In questi sistemi abbiamo diverse unità e strumenti **ciascuno dotato di una sua memoria e di un'unità di elaborazione dedicata**.

Possiamo quindi avere diversi algoritmi di analisi ed acquisizione. L'importante è che i dati prelevati siano consistenti tra loro.



SOFTWARE USATI PER CONTROLLARE LA STRUMENTAZIONE

STRUMENTO VIRTUALE

Parliamo di strumento virtuale perché usiamo **diversi strumenti reali** collegati tra loro in modo da **creare un nuovo strumento virtuale** che in realtà non esiste come *singola unità*. Possiamo quindi personalizzarne l'interfaccia grafica e diversi altri aspetti in modo da realizzare lo strumento che ci serve nello specifico.

Vantaggi dello strumento virtuale

- **Scalabilità**: un sistema di misura complesso può essere realizzato tramite diversi sistemi meno complessi.
- **connettività**: i dati possono essere facilmente memorizzati in un database.
- **costi più bassi**: possiamo usare un hardware a basso costo per la realizzazione dello strumento virtuale tramite il software che li "lega" creando un vero e proprio nuovo strumento.

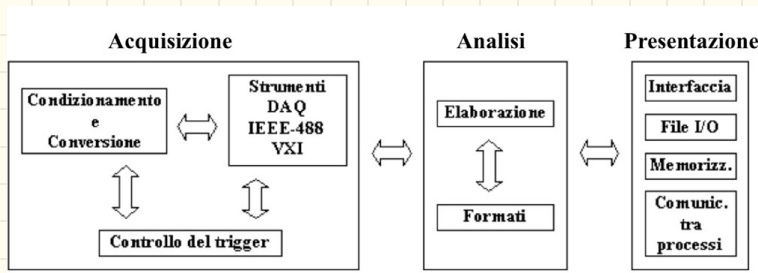
Cosa ci serve per creare uno strumento virtuale?

- Dobbiamo sicuramente avere un **hardware di misura** (strumento esterno, arduino con sensore o scheda di acquisizione dati)
- **un computer** per elaborare i dati
- **software applicativo** che ci permette di elaborare i dati e visualizzarli.

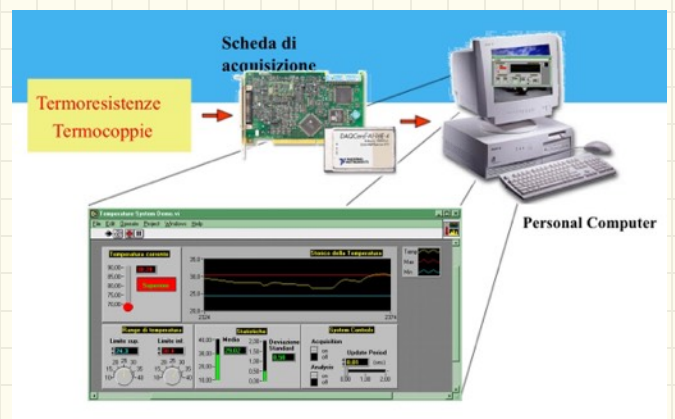
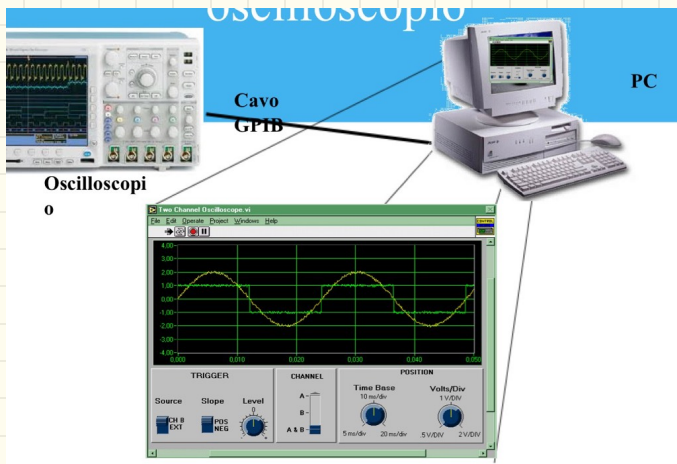
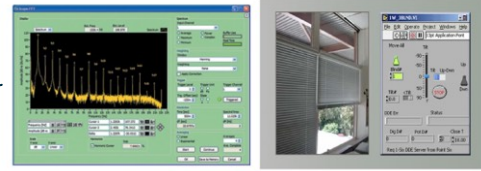
Svantaggi

Le prestazioni dello strumento virtuale sono inferiori rispetto ad uno **strumento dedicato**: **questo perché abbiamo più componente**, ed ognuna introduce una latenza ed un'incertezza. Di conseguenza perdiamo anche di accuratezza.

ARCHITETTURA DI UN VI



In questa sezione (eseguita sul pc) vengono effettuati i "calcoli" sulle misure; Ad esempio in questa sezione viene calcolata la media

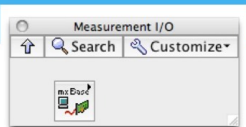


Caratteristiche

- Resolution** – Indica il massimo numero di valori di tensione *diversi* che possono essere misurati: maggiore è la risoluzione, più accurata è la rappresentazione del segnale
- Range** – Differenza fra il massimo ed il minimo valore che può essere misurato: minore è il range, più accurata è la rappresentazione del segnale
- Gain** – Amplificazione o attenuazione del segnale per rientrare nel range fissato

ACH	Input analogici (16)
DIO	I/O digitali (8)
PPI	Input programmabili per trigger e clocks esterni (10)
AISENSE	Morsetto comune per input analogico NRSE
DAC0OUT, DAC1OUT	Output analogici
EXTREF	Riferimento esterno per output analogici
SCANCLK	Clock per controllo multiplexing di AMUX64 e SCXI
EXTRSTRO BE*	Strobe esterno per controllo multiplexing di AMUX64 e SCXI
GPCTR_GAT E	Input gate per contatore general purpose
GPCTR_OUT	Output contatore general purpose
GPCTR_SOU BCF	Input contatore general purpose

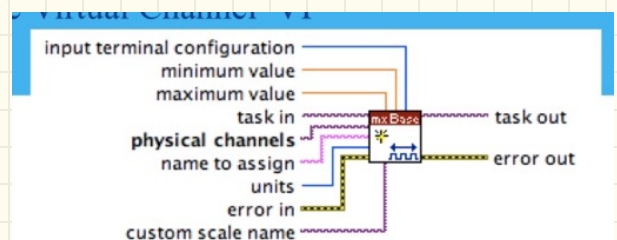
PIN Scheda Acquisizione



- Per poter gestire una scheda di acquisizione dati LabVIEW utilizza la Data Acquisition palette contenente i DAQmx VIs;

LabView è programmato in maniera **visuale**

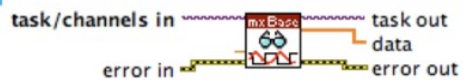
- I DAQmx VIs sono VIs divisi a seconda del tipo di task che deve eseguire la scheda di acquisizione dati.



Un **task** deve essere prima configurato e poi può essere utilizzato. Visualmente vediamo un blocco, mentre al di sotto abbiamo un codice scritto in C che viene compilato ed eseguito.

Lettura dei campioni

VI



Scrittura dei campioni

VI

