

Corso di Sistemi di Misura Distribuiti

TERZA LEZIONE:

RICHIAMI DI LABVIEW

LabView

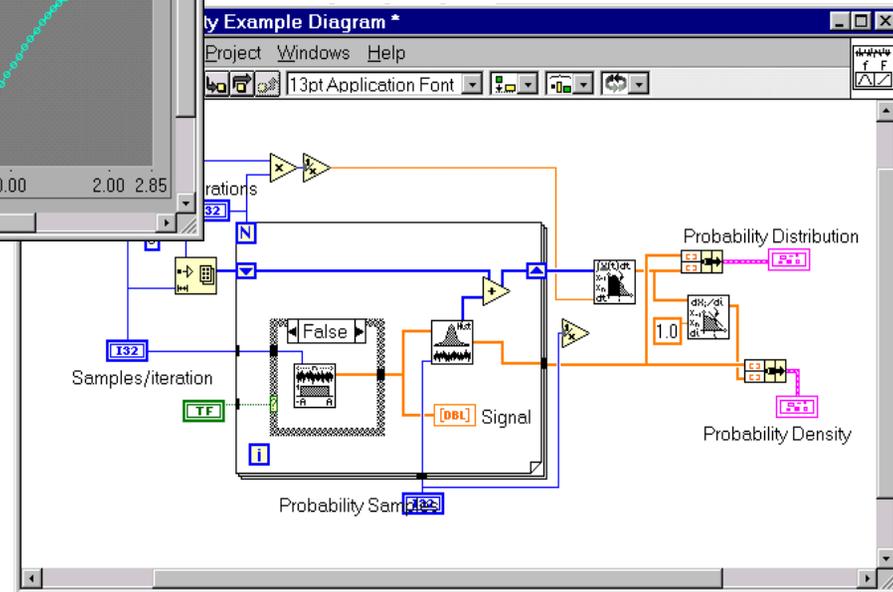
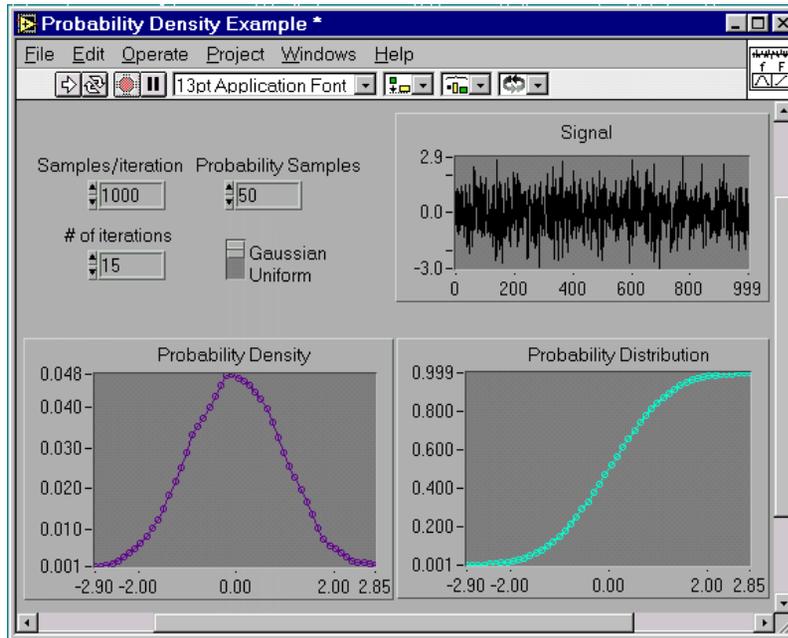
- ✓ Ambiente di sviluppo grafico:
 - Codice
 - Interfaccia utente
- ✓ Modulare
- ✓ Compilatore 32 bit
- ✓ Elevato numero di funzioni
- ✓ Librerie per la gestione della strumentazione
- ✓ Librerie per l'acquisizione dati
- ✓ Possibilità di includere *routine* in altro linguaggio

Librerie di funzioni

- Calcolo, confronto, cicli di controllo
- Operazioni su aggregati di dati (stringhe, array, ...)
- I/O su file
- Gestione strumentazione (RS232, 488, VXI, ...)
- Acquisizione dati
- Analisi (Generazione ed elaborazione segnali, Statistica, Fitting, Algebra lineare, ...)
- Comunicazione (DDE, TCP/IP, ActiveX)

Virtual Instrument (VI)

- Il Front Panel
- Il Block Diagram
- L'icona/connettore



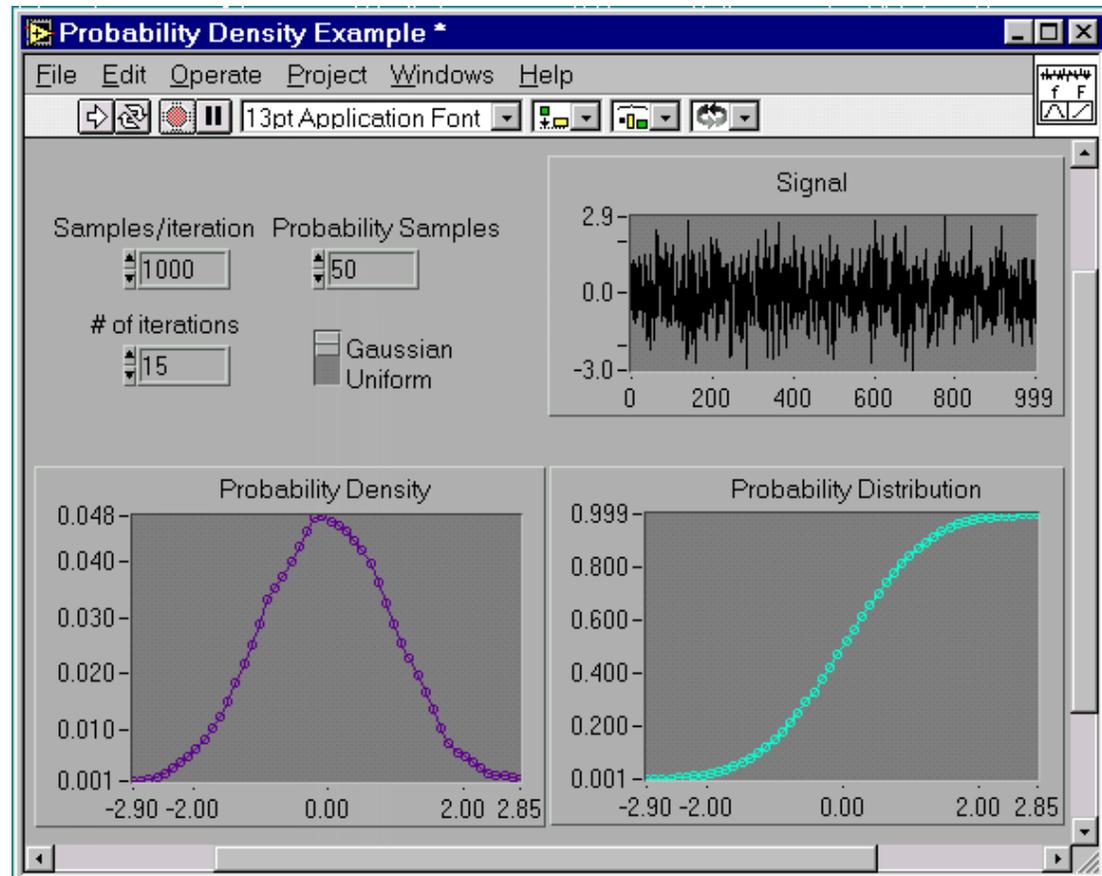
Il Front Panel

- E' l'interfaccia utente dell'applicazione

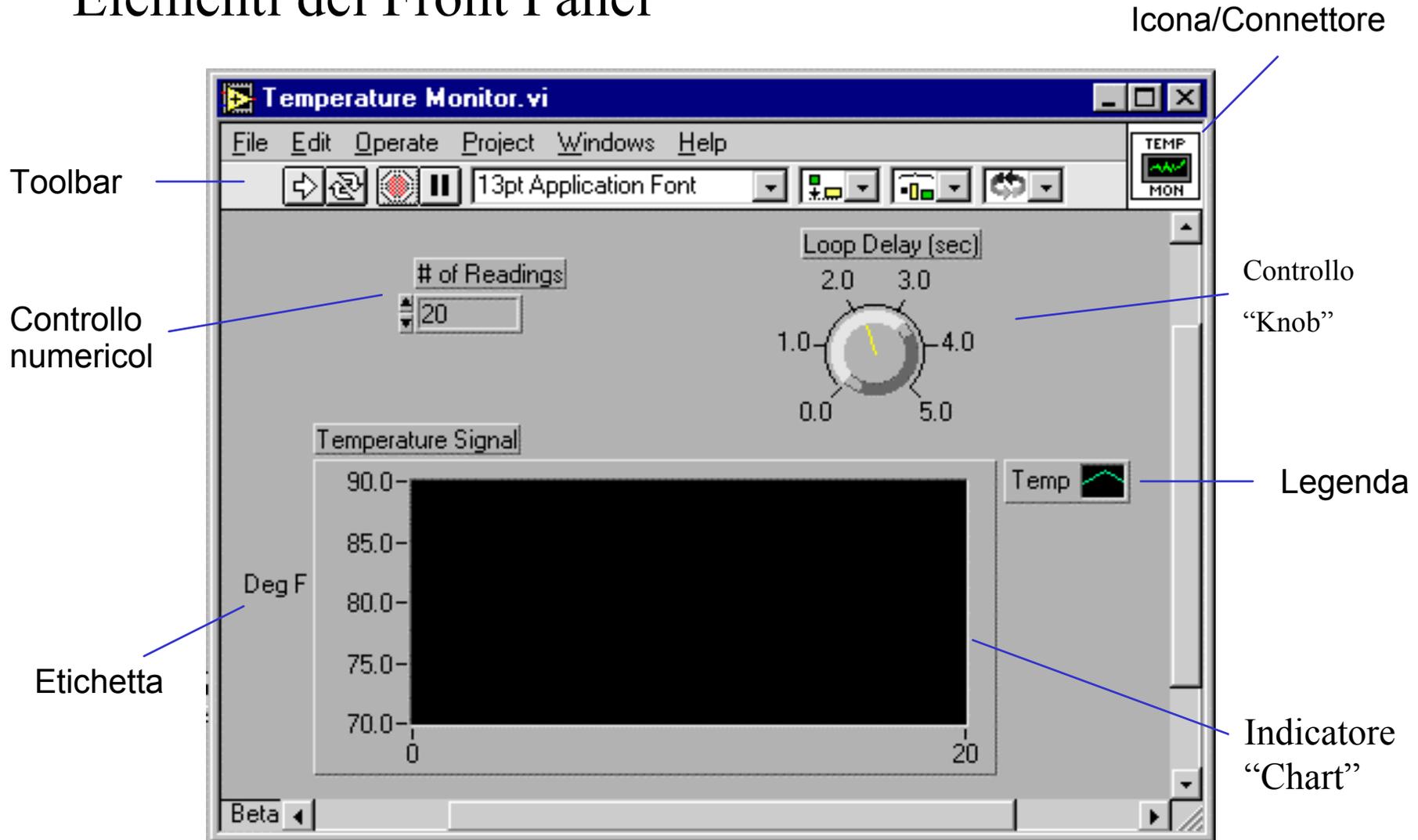
Contiene:

- **Controlli**
(input)

- **Indicatori**
(output)

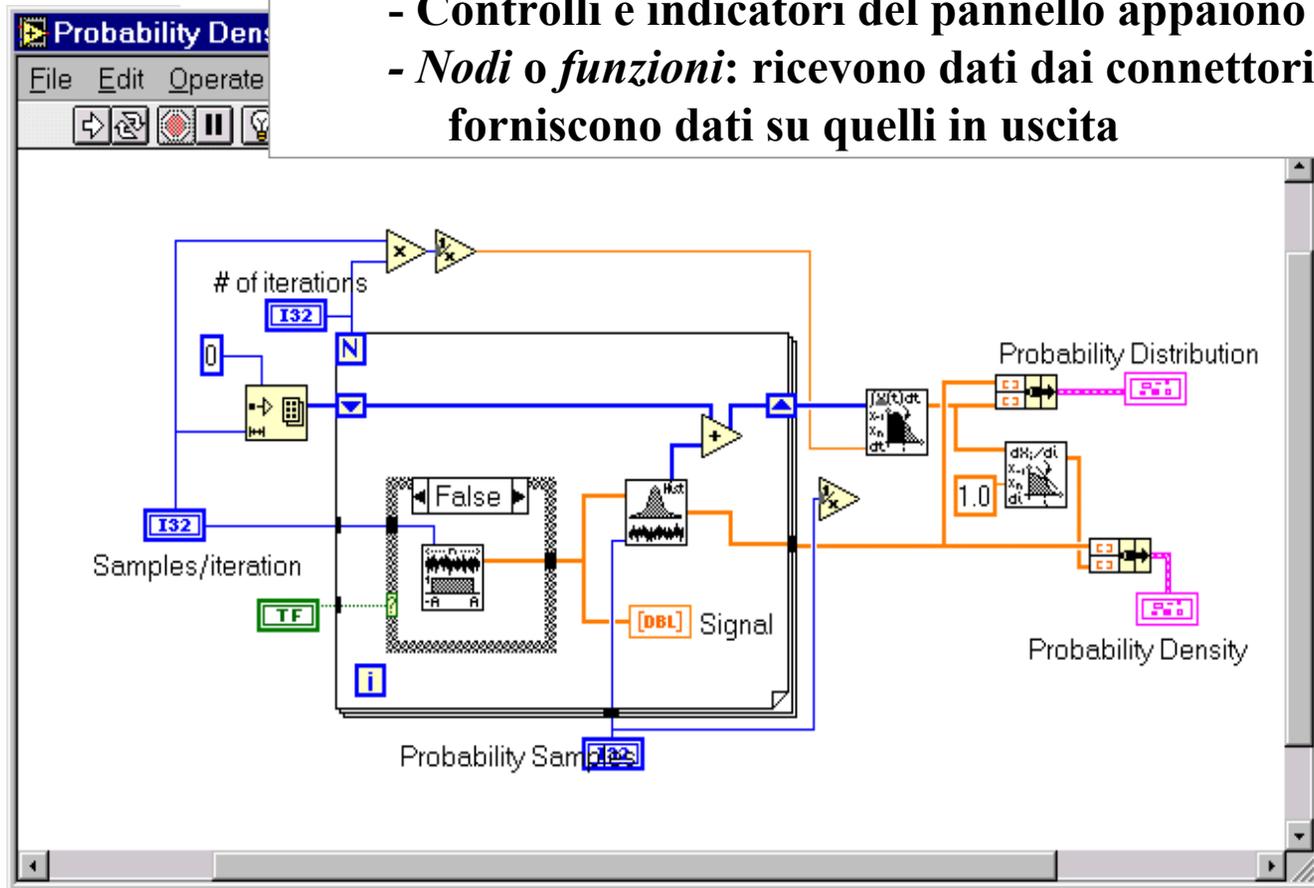


Elementi del Front Panel

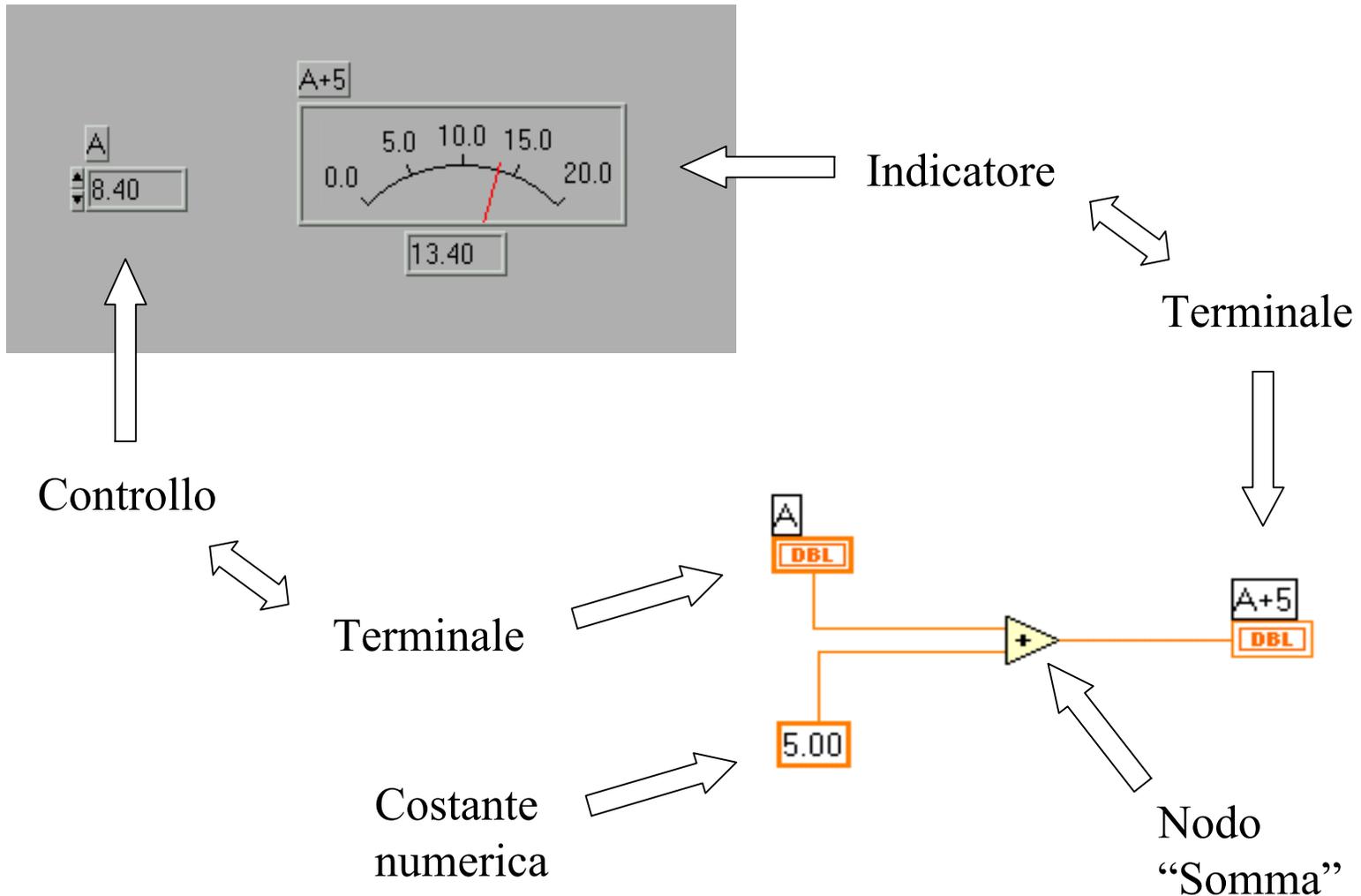


Il Block Diagram

- E' il codice sorgente dell'applicazione (in *Linguaggio G*)
- Controlli e indicatori del pannello appaiono come *Terminali*
- *Nodi o funzioni*: ricevono dati dai connettori di ingresso e forniscono dati su quelli in uscita

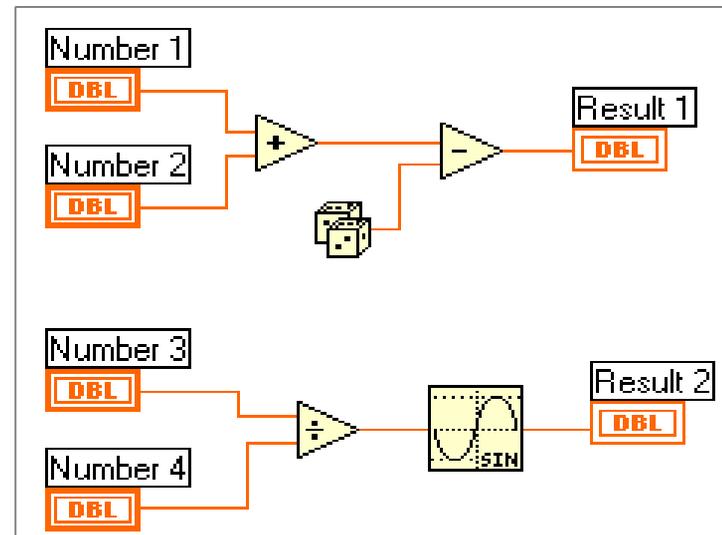
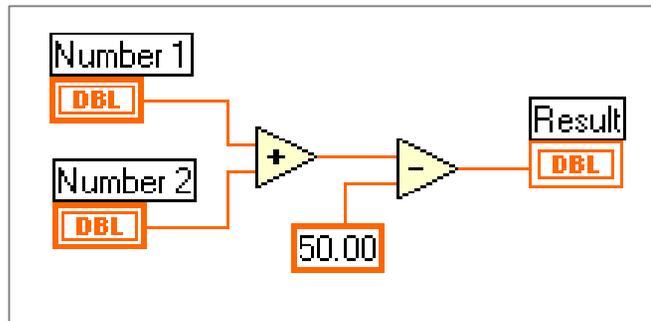


Elementi del Block Diagram



Dataflow programming

- Un nodo è eseguito quando i dati sono disponibili a **TUTTI** i suoi terminali di ingresso
- Un nodo fornisce i dati ai suoi terminali di uscita quando è terminata la sua esecuzione
- Se non esiste dipendenza tra due nodi, l'ordine con cui verranno eseguiti non è prevedibile !



Il modello di programmazione di LabView è ...

- *Data driven*

... e non è ...

- *Control driven*
- *Event driven*
- *Object oriented*

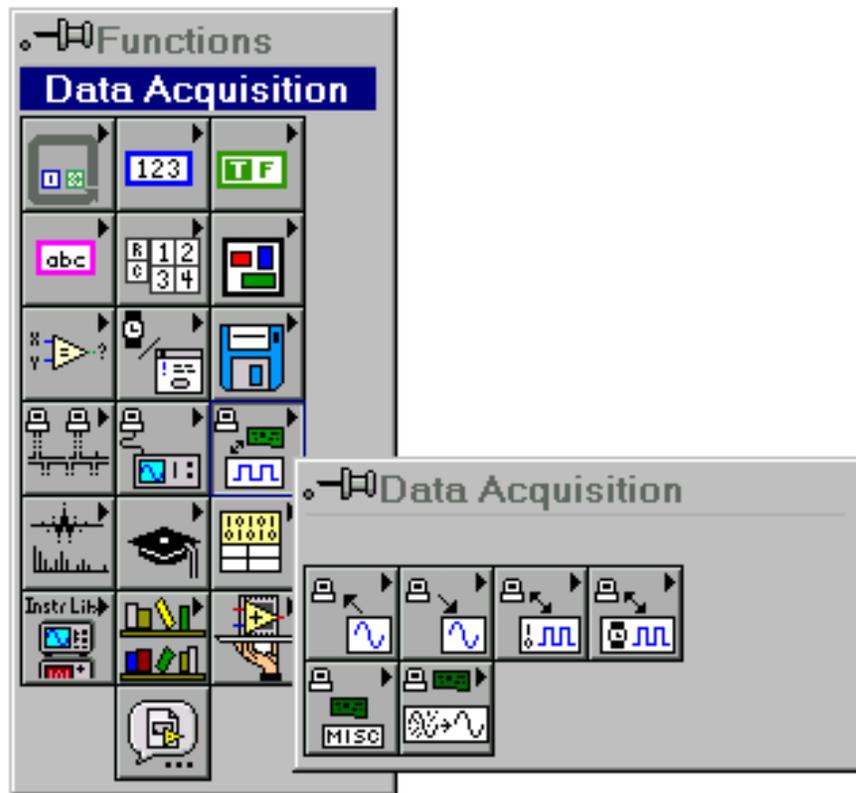
Dataflow programming

- Un filo collega un'unica sorgente di dati con uno o più nodi
- Il colore e lo spessore del filo individuano il tipo di dato

	Scalar	Array 1D	Array 2D	
Numeri				Arancione (floating point)
				Blu (interi)
Booleani				Verde
Stringhe				Viola

L'ambiente di sviluppo: Le palette

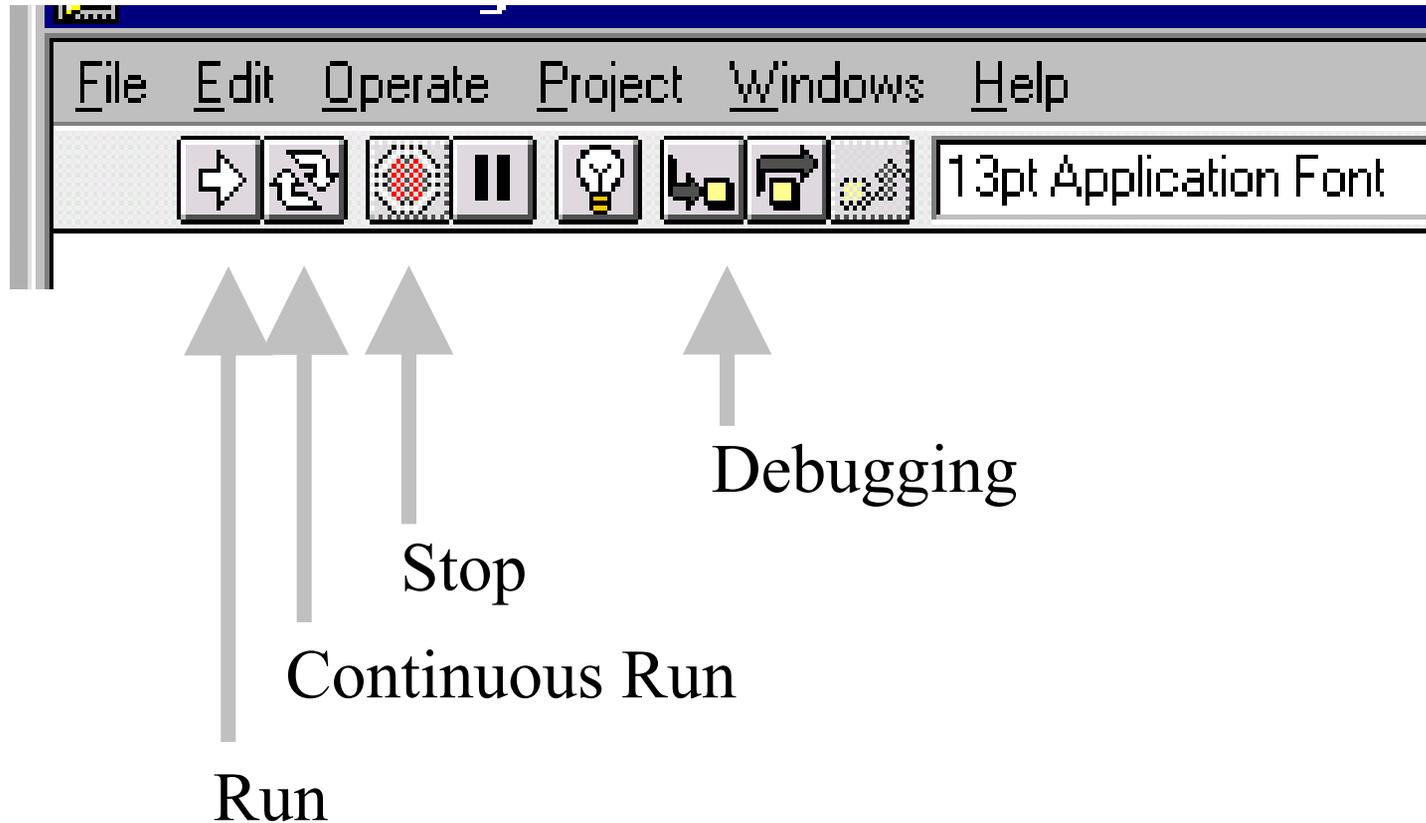
**Palette delle Funzioni
(Diagramma)**



**Palette dei Controlli
(Pannello)**



L'ambiente di sviluppo: la Toolbar



L'ambiente di sviluppo: i Tool



Operating: Inserimento dati

Positioning: Posizionamento e ridimensionamento

Labeling: Gestione etichette

Wiring: Collegamento (nel Block Diagram)

Object pop-up: Menu dell'oggetto

Scrolling: Scorrimento della finestra

Breakpoint: Inserimento stop (debug)

Probe: Visualizzazione dati nei fili (debug)

Color copy: Copia di un colore

Coloring: Per colorare un oggetto

Tastiera:

- Tab
- Spazio

Combinazioni di tasti

- <Ctrl-R>** *Esegue il VI*
- <Ctrl-E>** *Mostra pannello/diagramma*
- <Ctrl-H>** *Finestra di context help*
- <Ctrl-B>** *Cancella fili interrotti*
- <Ctrl-F>** *Contenuto della memoria*
- <Tab>** e **<Spazio>** *Tools*
- <Shift>** *Incremento/decremento veloce*

Tecniche di debugging



Click sulla freccia spezzata: Info sull'errore



Highlighting button: Esecuzione animata



Probe: Mostra il valore dei dati mentre scorrono lungo i fili



Breakpoint: ferma l'esecuzione quando i dati sono disponibili sul filo



Single Step: Esecuzione passo-passo



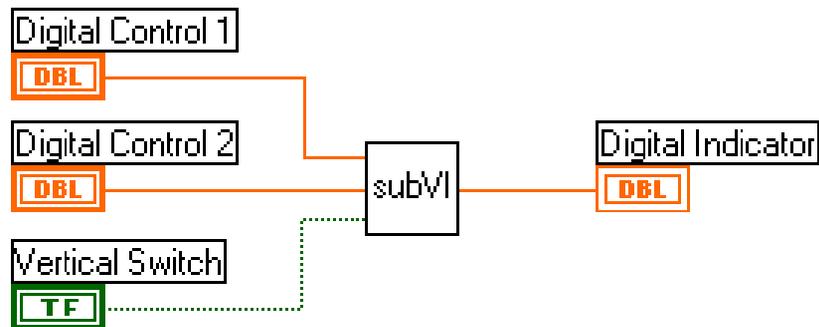
Step Over: Esecuzione passo-passo, non entra nei nodi



Step Out: Esecuzione passo-passo, esce dai nodi nei quali entrato

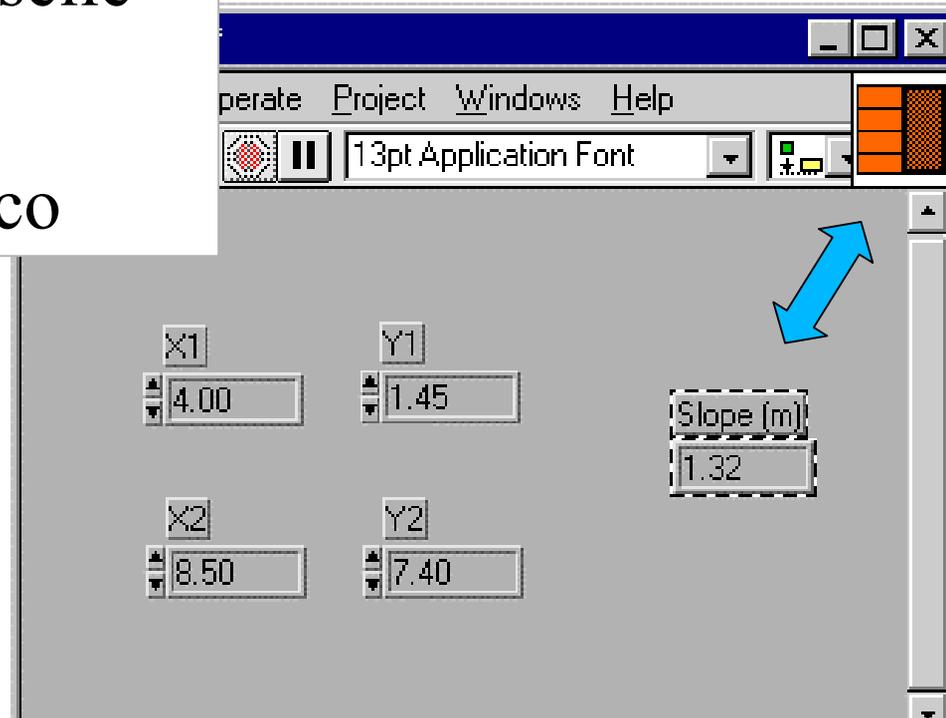
SubVI

- Un VI può essere usato nel Block Diagram di un altro VI di livello più alto
- Nel VI chiamante, un SubVI è rappresentato con la sua icona



Realizzazione di un SubVI

- Disegno dell'icona
- Associazione di controlli e indicatori alle caselle del connettore
- Salvataggio su disco



Chiamata ad un SubVI

- Functions >> Select a VI ...
- Trascinare l'icona sul diagramma chiamante

