

- Paolo Ghidetti
- paolo.ghidetti@unipv.it

LabVIEW

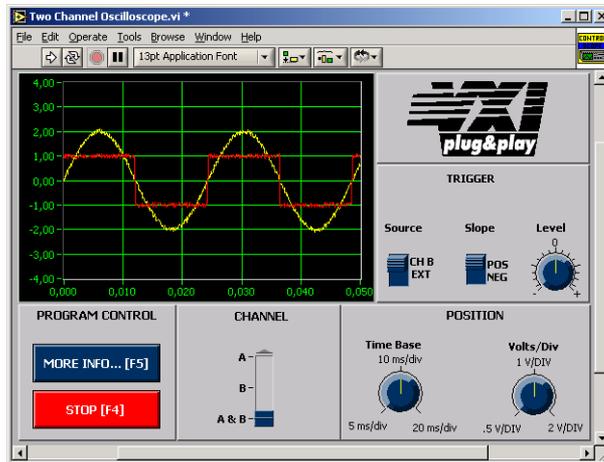
(Laboratory Virtual Instrument Engineering Workbench)

- Linguaggio di programmazione sviluppato da National Instruments
- Nato per Mac (1986), per Windows dal 1992, per Linux dal 1999
- Adatto per realizzare software di acquisizione ed elaborazione dati (affiancato da hardware della stessa National Instruments)
- Implementa anche moduli di gestione di rete, data base, DLL, Active X, XML

G – Language

è un linguaggio grafico dove i programmi sono scritti disegnando dei diagrammi a blocchi

Un programma in LabVIEW è detto: Virtual Instrument (VI)



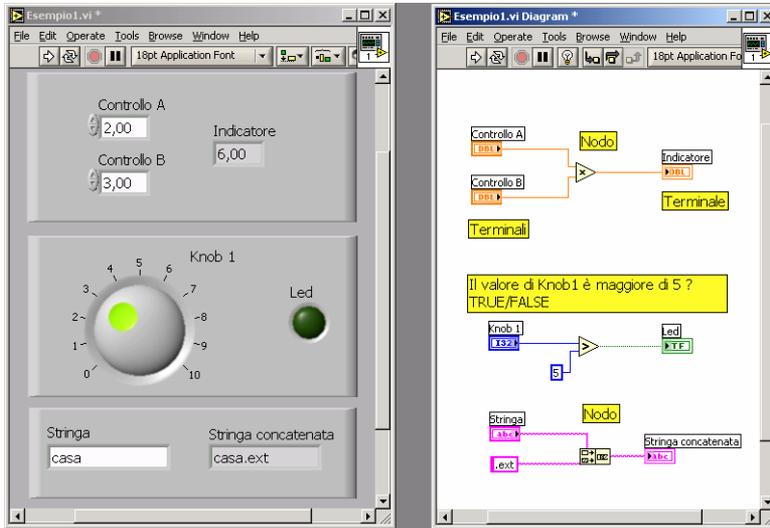
Ambiente di sviluppo

Diagram
"codice del programma"

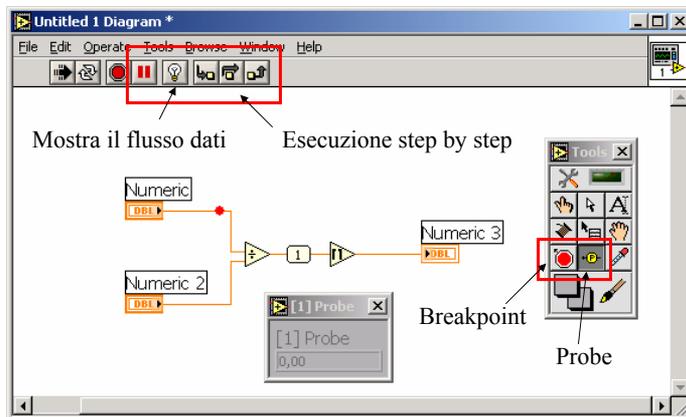
Right click per aprire i menu

Front Panel
(Interfaccia Grafica Utente)

Costruzione di un VI



Strumenti di debug



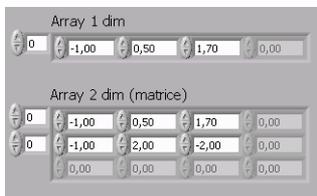
LabVIEW: alcuni tipi di dati

Control	Indicator	Data Type	Color
		Single-precision floating-point numeric	Orange
		Double-precision floating-point numeric	Orange
		Extended-precision floating-point numeric	Orange
		Complex single-precision floating-point numeric	Orange
		Complex double-precision floating-point numeric	Orange
		Complex extended-precision floating-point numeric	Orange
		Signed 8-bit integer numeric	Blue
		Signed 16-bit integer numeric	Blue
		Signed 32-bit integer numeric	Blue
		Unsigned 8-bit integer numeric	Blue
		Unsigned 16-bit integer numeric	Blue
		Unsigned 32-bit integer numeric	Blue
		Enumerated type	Blue
		Boolean	Green
		String	Pink

Altri Tipi di dati

- **Array:** insieme di dati dello stesso tipo (indicizzati)

(controls>>array & cluster>>array)



Array 1 dim


- **Cluster:** “contenitore” con all’interno dati di diverso tipo

(controls>>array & cluster>>cluster)



Cluster


LabVIEW: dataflow

- Nei linguaggi di programmazione tradizionali l'esecuzione è in modo sequenziale (top-down)
- In LabVIEW è il flusso di dati che determina l'ordine di esecuzione

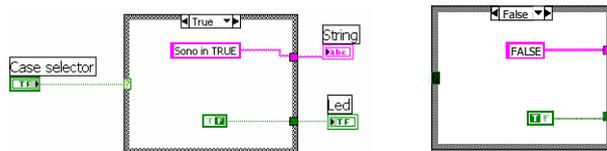


REGOLA: un nodo viene eseguito solo quando tutti i suoi ingressi sono disponibili

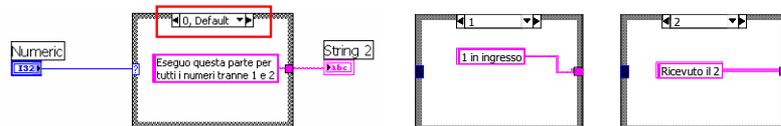
LabVIEW strutture: CASE (functions>>structures>>case structure)

Utilizzata per:

- il costrutto IF...THEN

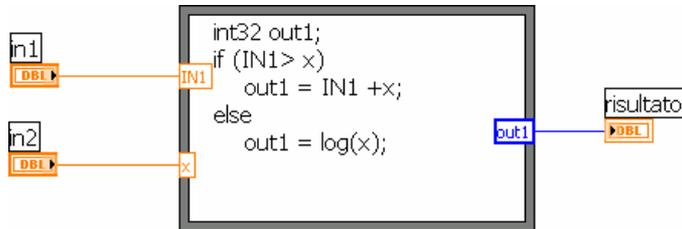


- il costrutto CASE

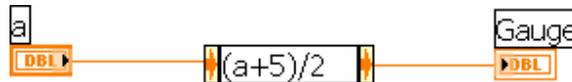


LabVIEW strutture: Formula Node (functions>>structures>>formula node)

Permette di eseguire espressioni scritte in un codice simile al C



Expression Node (functions >> numerics >> expression node)



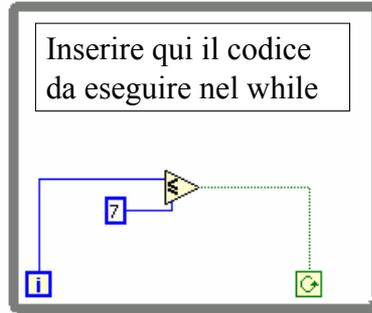
LabVIEW strutture: ciclo FOR (functions>>structures>>for loop)

- Questo esempio esegue 5 cicli
- “i” parte da zero
- “i” viene incrementato automaticamente di 1 ogni ciclo



LabVIEW strutture: ciclo WHILE (functions>>structures>>while loop)

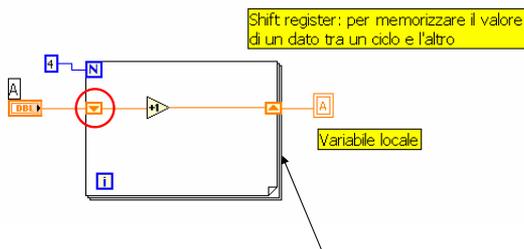
```
i= -1  
do  
    i=i+1  
    .....  
while i <= 7
```



“i” è incrementato in automatico ad ogni ciclo

Shift Register

- Si applica a cicli FOR e WHILE
- Utilizzato per trasferire valori tra un ciclo e il successivo

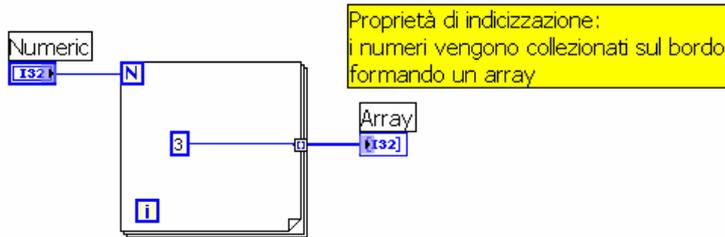


```
For i=0 to N-1 step 1  
    A=A+1  
Next i
```

Right click sul bordo per aggiungere uno shift register

Proprietà di indicizzazione dei cicli

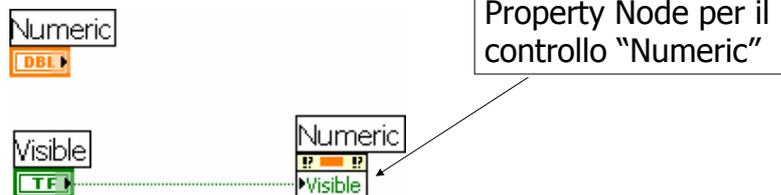
- si applica a cicli FOR e WHILE



Property Node

Right click sul terminale >> create >> property node

- Permettono di controllare le proprietà di controlli e indicatori.
- Alcune proprietà: visibile/invisibile, abilitato/disabilitato, colore, posizione,.....

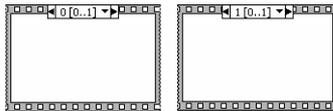


Operazioni in sequenza

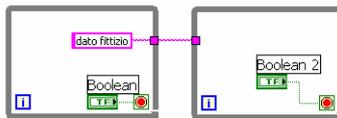
Nel caso esistano operazioni da eseguire in sequenza anche se non hanno una dipendenza diretta dei dati ?

Si usano:

- sequenze (functions>>structures>>sequence structure)



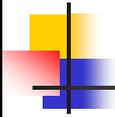
- creare una dipendenza dati "fittizia"



LabVIEW: grafici

(controls>>graph>>...)

- **Waveform Chart:** associa il nuovo dato a quelli già visualizzati (memorizza una storia di dati)
- **Waveform Graph:** riceve in ingresso un array contenente le ordinate dei punti da disegnare
- **XY Graph:** riceve in ingresso 2 array, uno per le ascisse e uno per le ordinate
- **Intensity Graph/Chart:** associano un colore al dato ricevuto



Wait functions

