

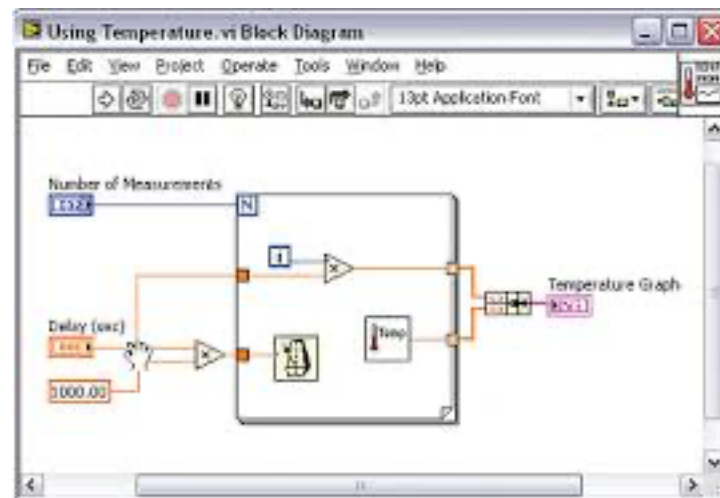
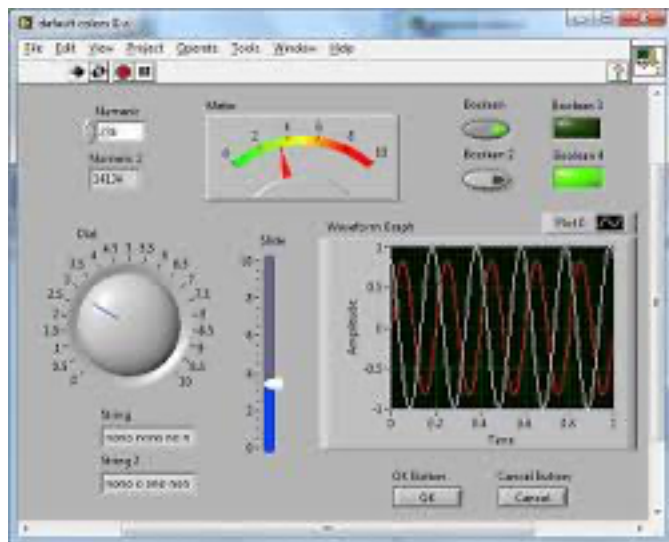
LabVIEW

National Instruments LabVIEW

- Grafikus programcsomag (ikonok és vezetékek)
- C, Pascal felépítés grafikus reprezentációja
- Mérés, vezérlés, szimuláció megvalósítása platform-alapú megközelítéssel
- Interface-k hardware-k széles skálájával
- Beépített analízis-könyvtárakkal
- .vi kiterjesztés: virtual instrumentation (VI)

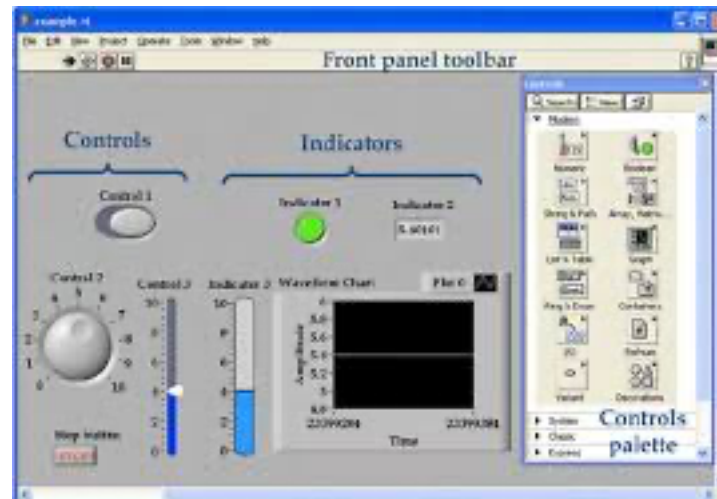
LabVIEW felépítése

| Front Panel | Block Diagram | Icon/Connector Pane |
|--|--|--|
| Felhasználói interfész | Grafikus forrása a kódnak | A VI-t lehet vele reprezentálni, illetve aVI-t (subVI) lehet használni |
| Bemenetek a vezérléshez (control), és kimenetek (indicators) | Kapcsolatokat (terminals) tartalmaz a Front Panel-től az algoritmus felé | A VI be- és kimeneteit felelteti meg |



Vezérlők és Kijelzők

| Controls | Indicators |
|--|--|
| Interaktív bemenet | Interaktív kimenet |
| Gombok, nyomógombok, tárcsázók és más bemeneti eszközök | Gráfok, LED-ek, és más kijelzők |
| A fejlesztőeszköz bemeneteit szimulálja és a block diagram-ot látja el adattal | A fejlesztőeszköz kimeneteit szimulálja és a block diagram eredményét jeleníti meg |



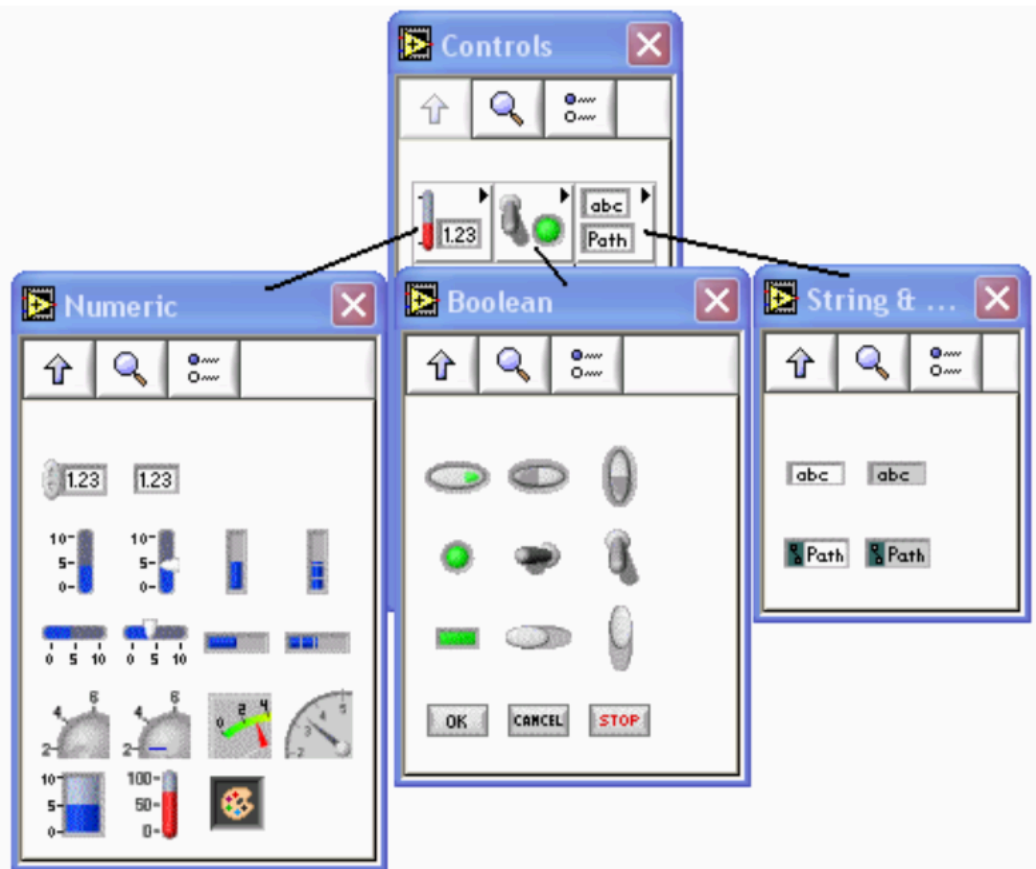
Vezérlők és Kijelzők

- Jobb egérgombra előhívható
- Window – Show Control Palette
- Minden vezérlő kijelzővé Edit-álható és vissza



4-3. ábra Controls Palette

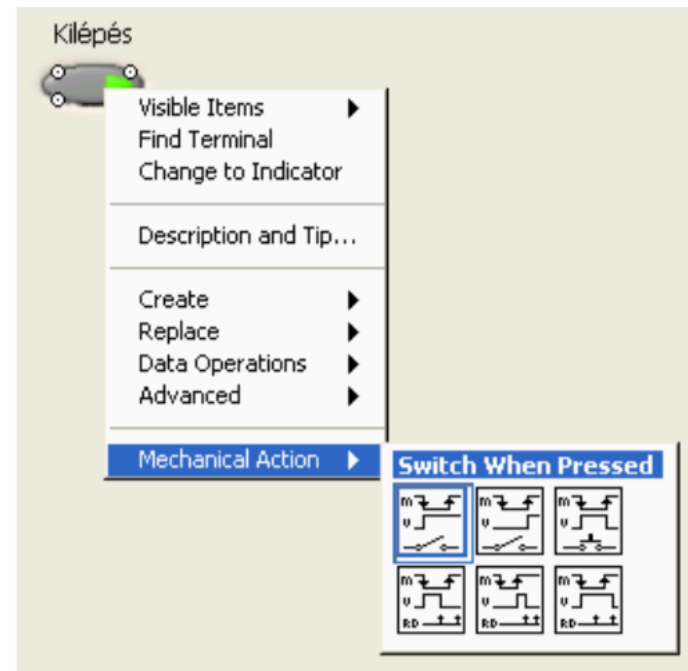
Vezérlők és Kijelzők



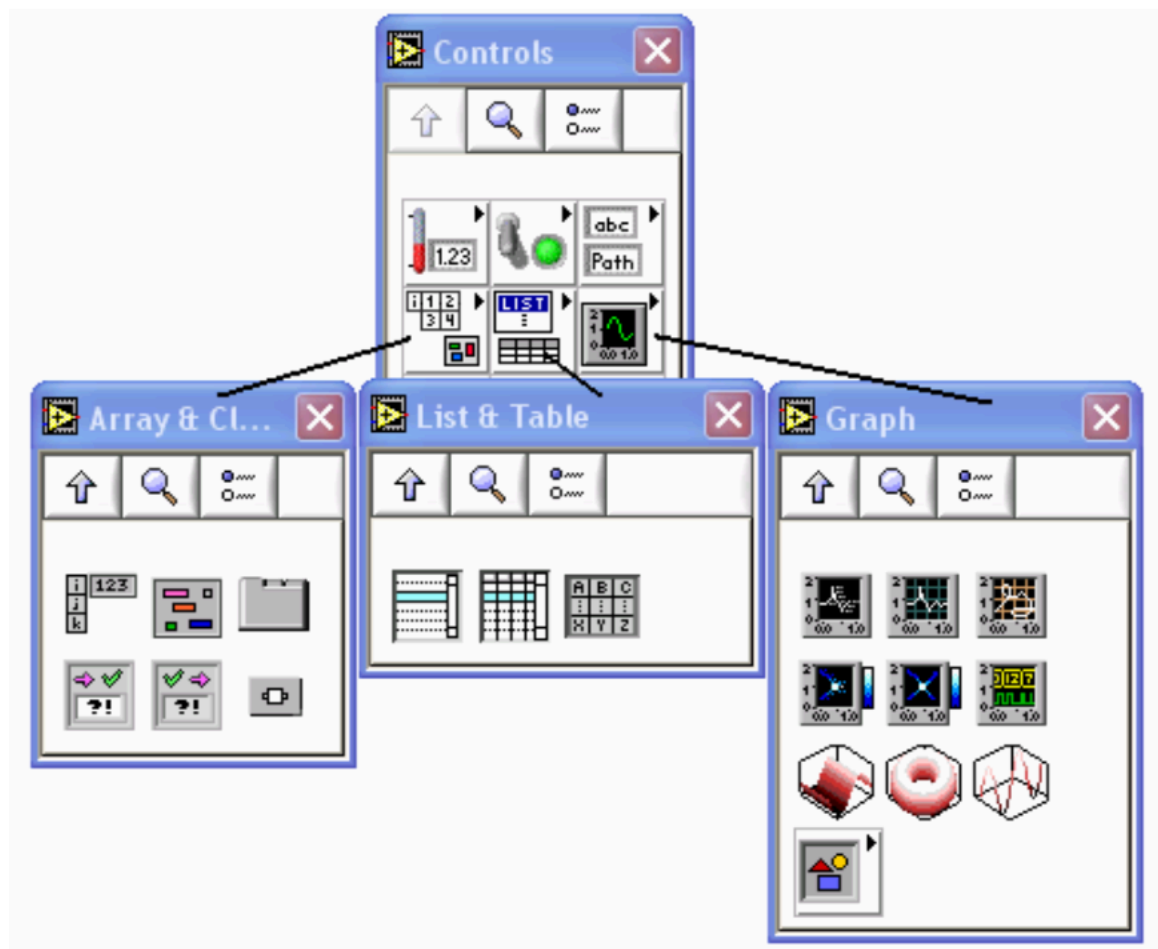
számérték megadása

kétállású kapcsolók

szövegmezők



Vezérlők és Kijelzők

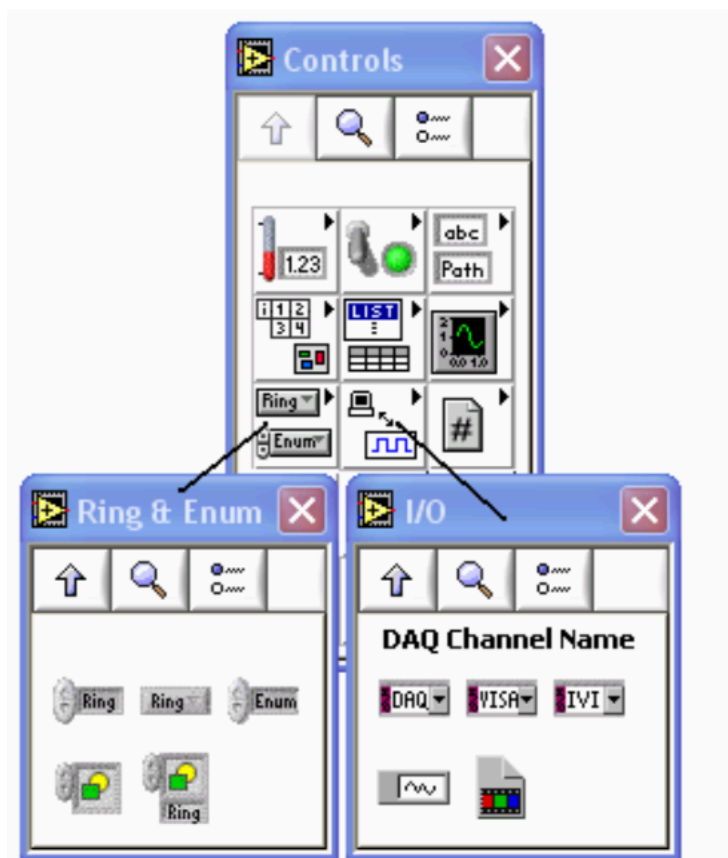


tömbök megadása

táblázatok

grafikonok (2D és 3D)

Vezérlők és Kijelzők



legördülő menükmérőcsatornák kiválasztása

Adattípusok

- String:
 - ASCII karakterek sorozata. (pl.: USER name, PWD; szöveges visszajelzés a felhasználónak)
- Bool:
 - Csak két állapotot vehet fel (igaz v. hamis, be v. ki)
- Numerikus:
 - Egész vagy valós
 - Numerikus bemenetek, numerikus indikátorok

VI Block Diagram

Block
Diagram
Toolbar

SubVI

While Loop
Structure

Numeric
Constant

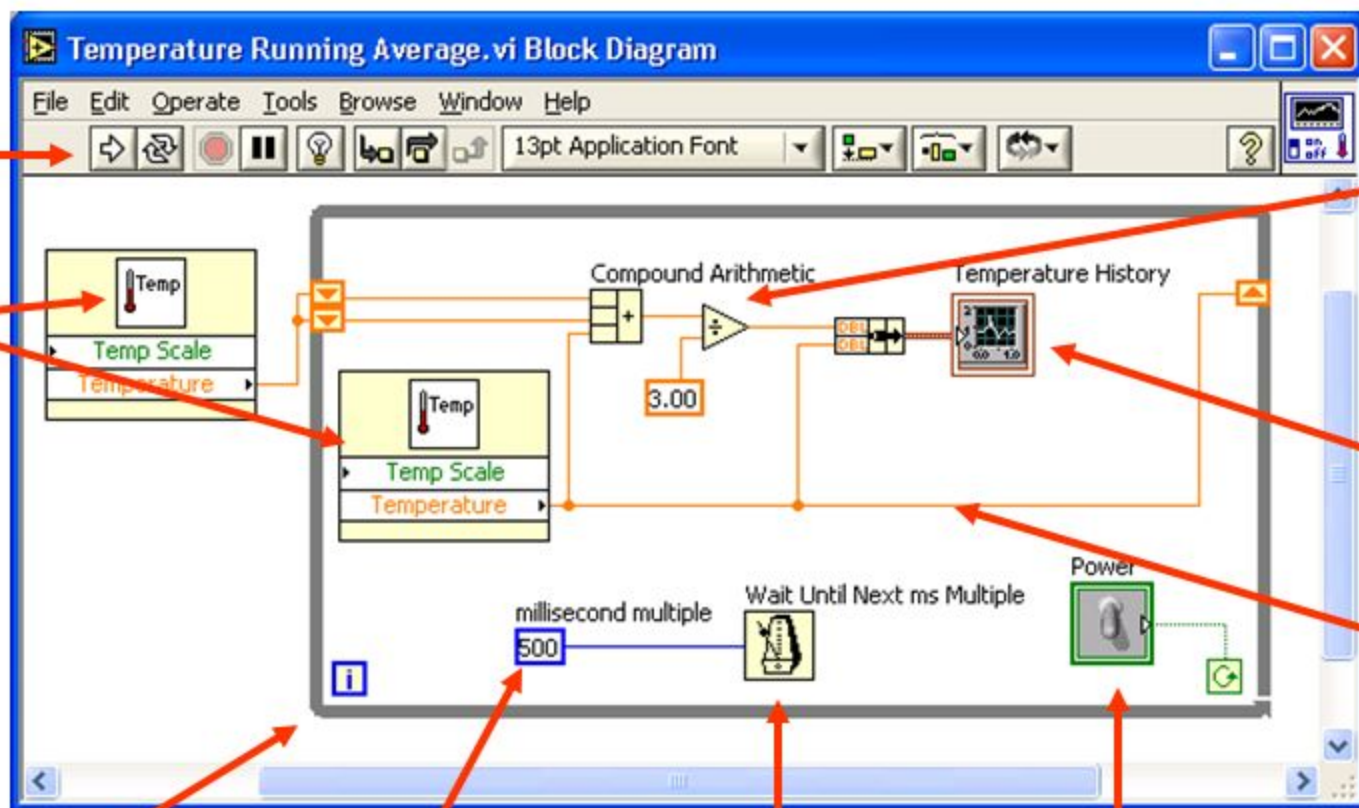
Timing
Function

Boolean Control
Terminal

Divide
Function

Graph
Terminal

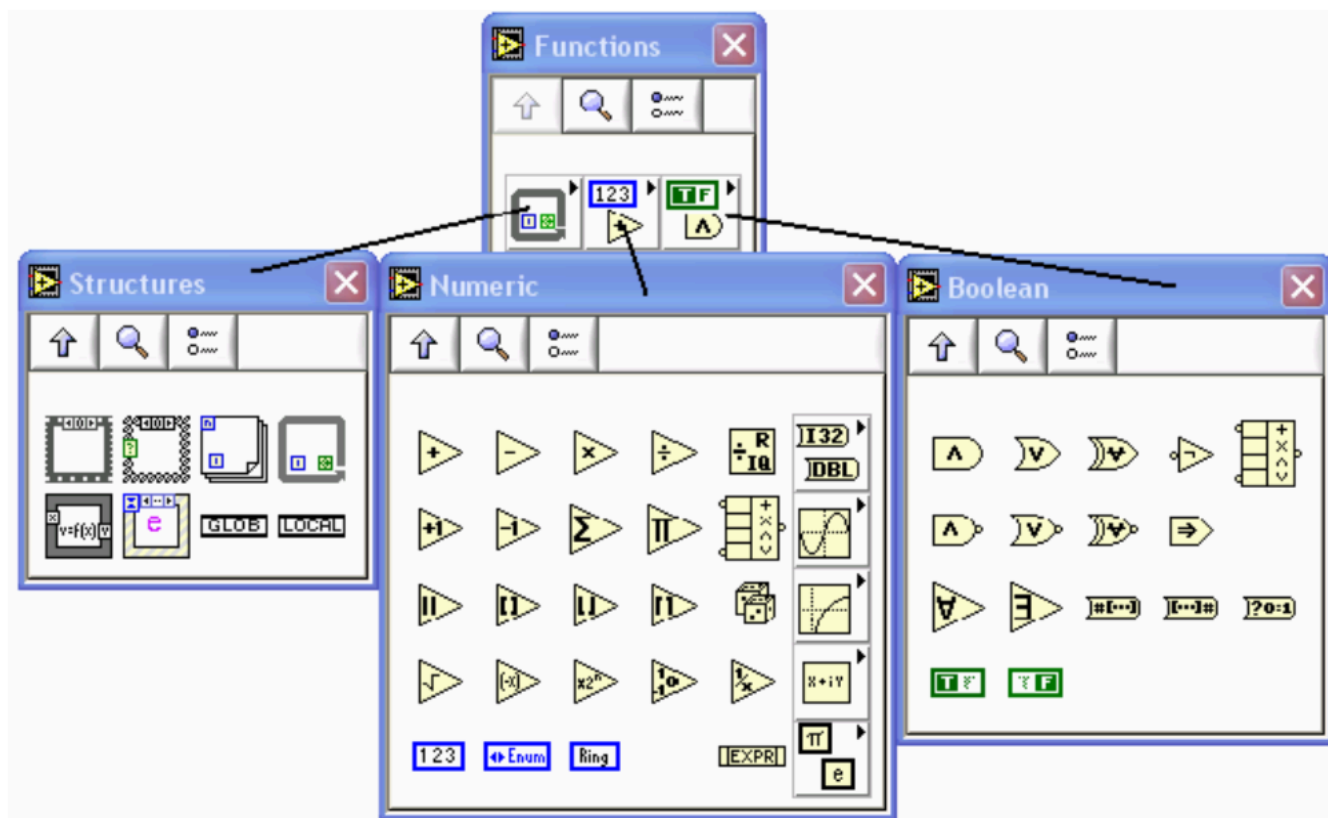
Wire
Data



Block diagram

- Terminálisok (terminals): belépő ill. kimeneti portok a Front Panel és a Block Diagram között
- Csomópontok (nodes): Block Diagramm objektumai, melyek valamilyen műveletet hajtanak végre, rendelkeznek be-/kimenetekkel. (C analógiája: függvények, állítások, operátorok, szubrutinok). Csomópontok lehetnek függvények, al-VI-k vagy struktúrák.
- Függvények (functions): LabView alapvető műveletvégző elemei. Nem rendelkeznek Front Panellel vagy Block Diagrammal. Halvány sárga háttérrel jelzik.
- al-VI (subVI): VI-n belül futó VI. Front Panellel és Block Diagrammal rendelkezik. Bármely VI lehet subVI. Ha belekattintunk, megnyílik a hozzá tartozó Front Panel és Block Diagram.
- Ikon (Icon): a VI grafikus reprezentációja.

Függvény Paletta

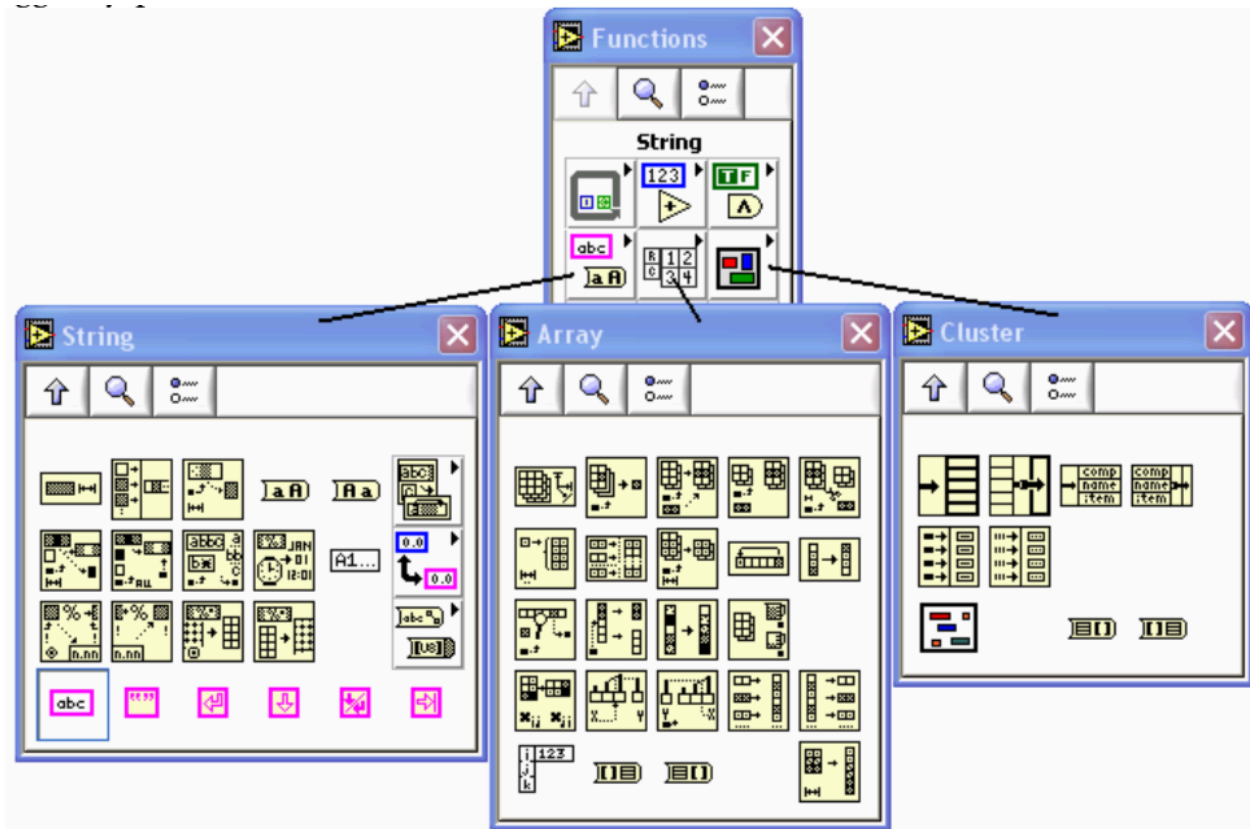


ciklusok
while, for
case, sequence
egyenletek

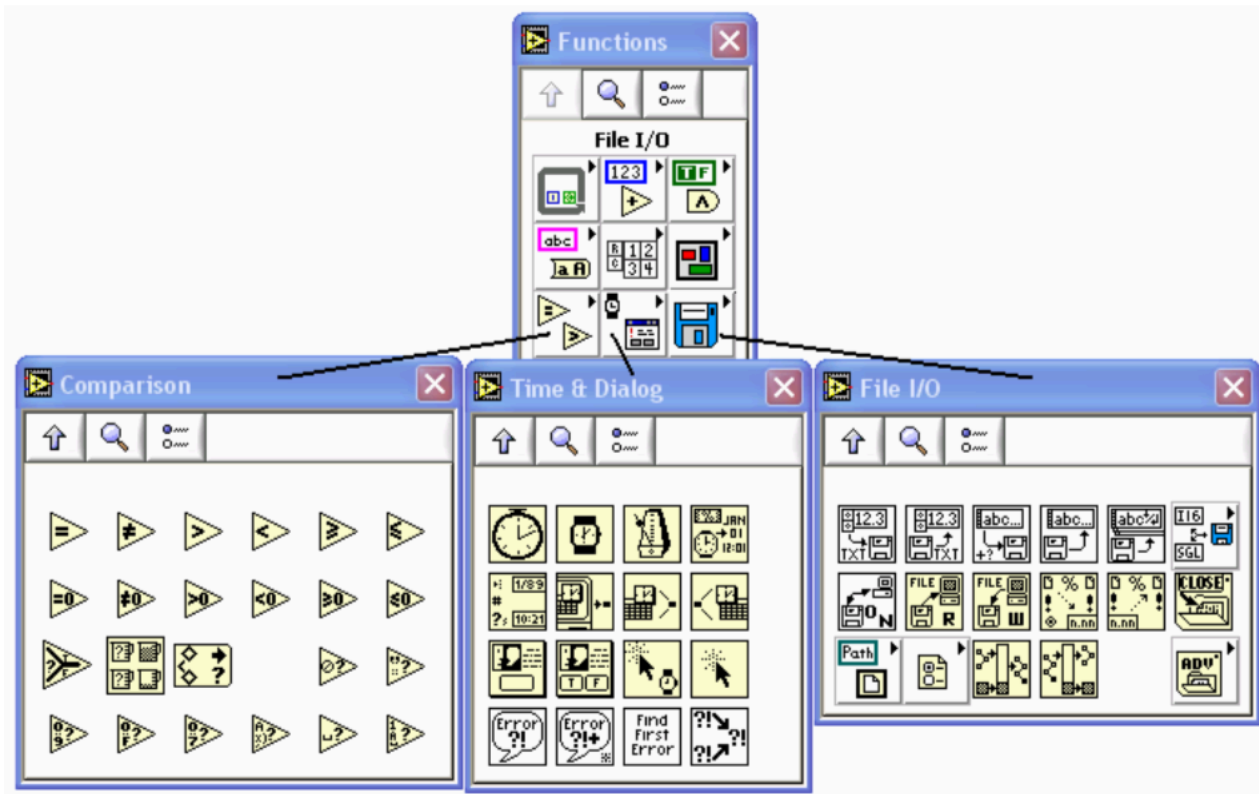
számtani alpműveletek
trigonometrikus függvények

Boolean műveletek

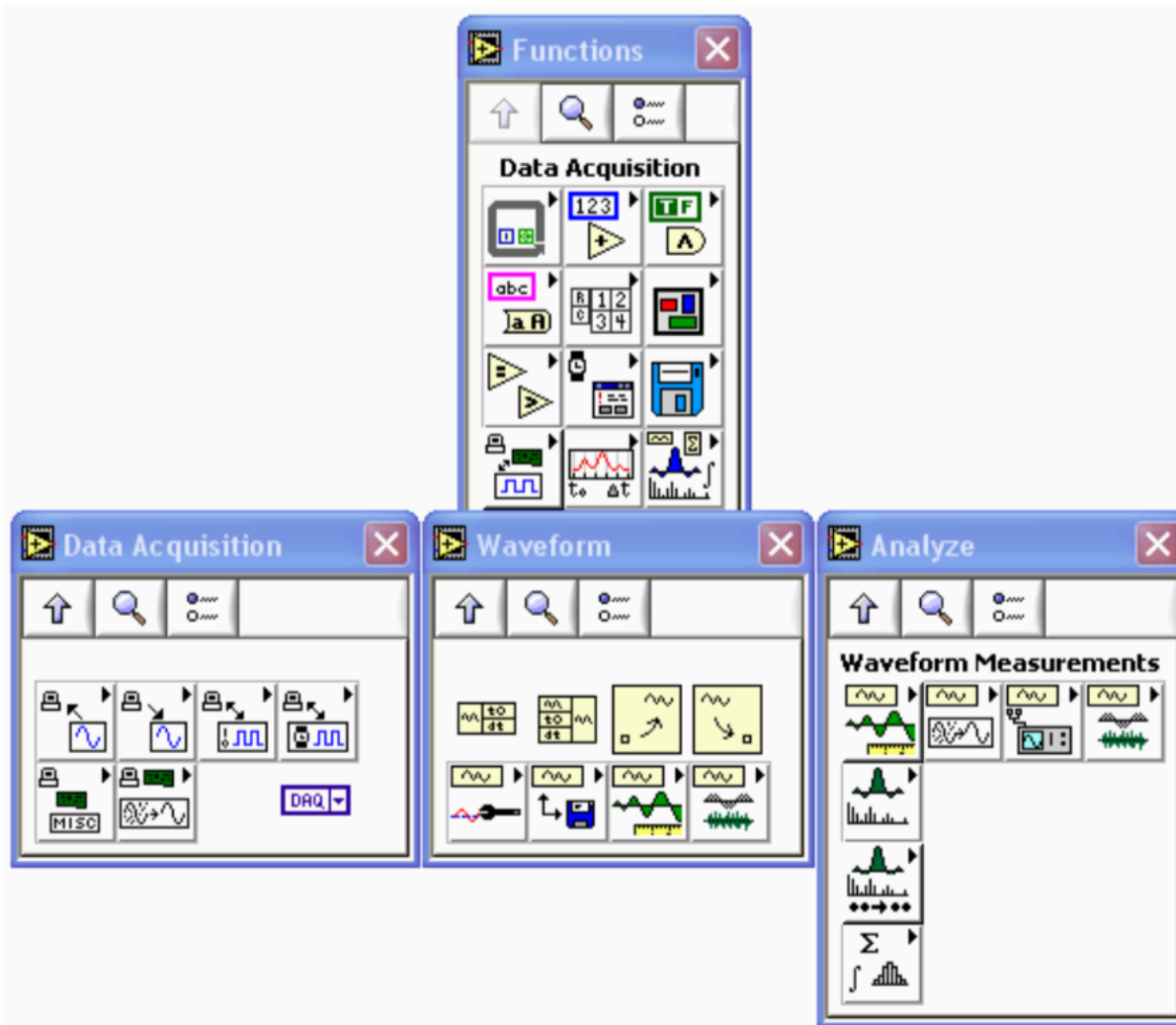
Függvény Paletta



Függvény Paletta



Függvény Paletta



Változó típusok



- integer



- double (real)



- Boolean

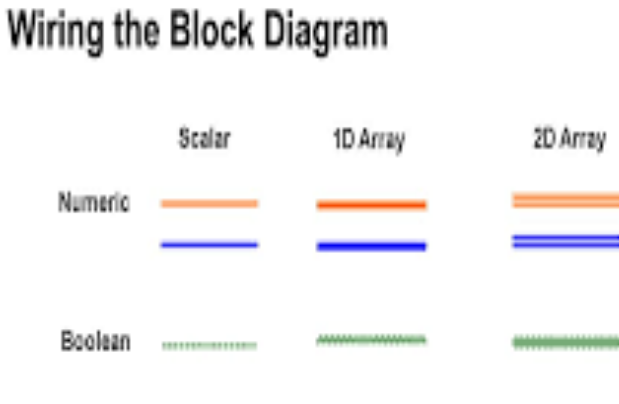


- string



- virtual channel

A leggyakoribb vezetékek típusok

| Vezeték típusok | | | | |
|-----------------|--|---------|---------|--|
| Vezeték típusa | Skalár | 1D tömb | 2D tömb | Szín |
| Numerikus |  <p>The diagram illustrates the wiring for different data types across three array configurations: Scalar, 1D Array, and 2D Array. The types shown are Numeric (orange and blue lines), Boolean (green dashed lines), String (pink wavy lines), and Dynamic (dark blue dashed lines).</p> | | | Narancssárga (lebegőpontos), Kék (integer) |
| Bool | | | | Zöld |
| Sztring | | | | Pink |

Segítség

- Környezet függő segítség (Context Help):
 - LabView objektumokról alapvető segítség (kurzorra az objektum felé állni)
 - Help -> Show Context Help, <CTRL-H>, Show Context Help gomb
- LabVIEW Help
- Példák: Help -> Find Examples

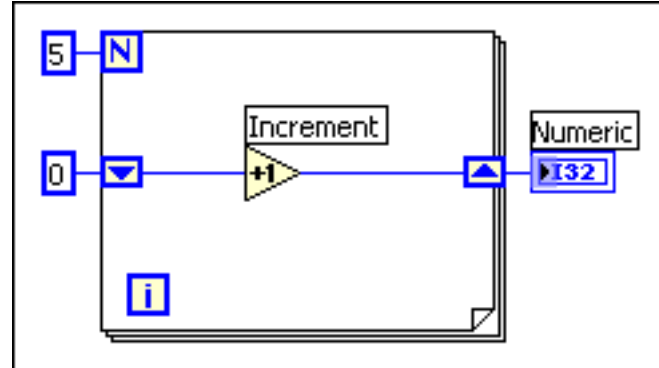
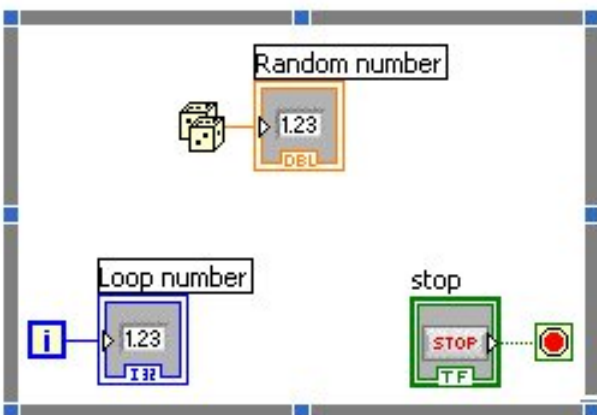
Adatfolyam programozás

- Adat-vezérelt, adat-függő: A blokkdiagramm adatáramlási sorrendben hajtódik végre. Összetettebb esetben nem határozható meg egyértelműen, hogy időben melyik az utolsó befejezett művelet.
- Egy csomópont akkor hajtódik végre, ha mindegyik bemeneti terminálisán megjelennek az adatok
- A csomópont végeredménye a kimeneti terminálison jelenik meg.

Hibakeresés

- Broken wire
- Required terminal is unwired
- Control wired to another control
- Broken subVI due to unwired required terminal
- Debugging, breakpoints

Iterációk



While Loop

Do Loop-hoz, vagy Repeat-Until-hoz hasonló

Kódszegmens ismétlése a feltétel teljesüléséig

Legalább 1x végrehajtódik

Iteration Terminal – elvégzett iterációk száma

Iteration Terminal mindig 0-val kezdődik

Conditional Terminal – az iteráció leállításának feltétele

Automatikusan az utolsó értéket csatornázza ki

For Loop

Meghatározott számú iterációt hajt végre

Count Terminal – bemenet, hányszor hajtja végre a kódrészletet

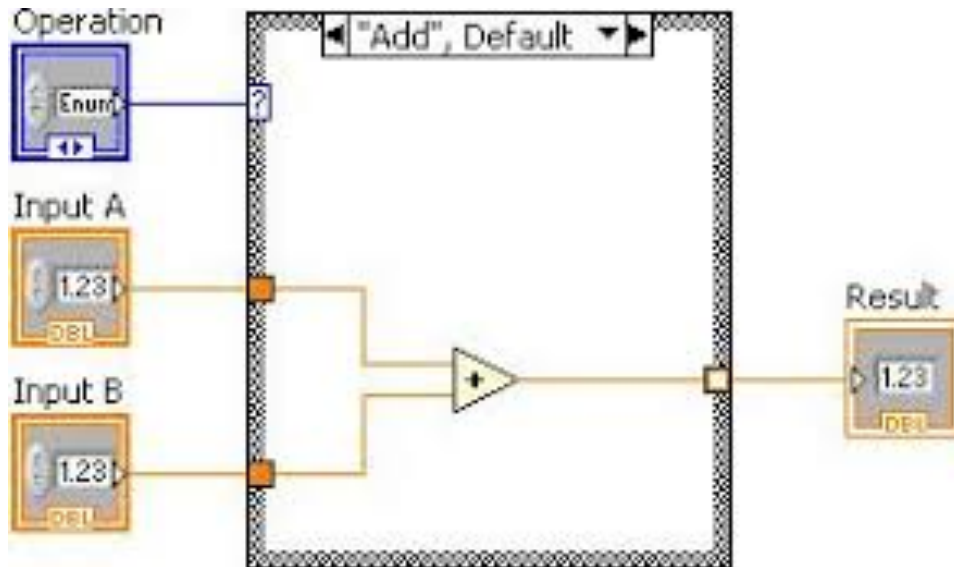
0x is végrehajtható

Iteration Terminal – kimenet, elvégzett iterációk száma

Iteration Terminal mindig 0-val kezdődik

Automatikusan adat-tömböt csatornáz ki

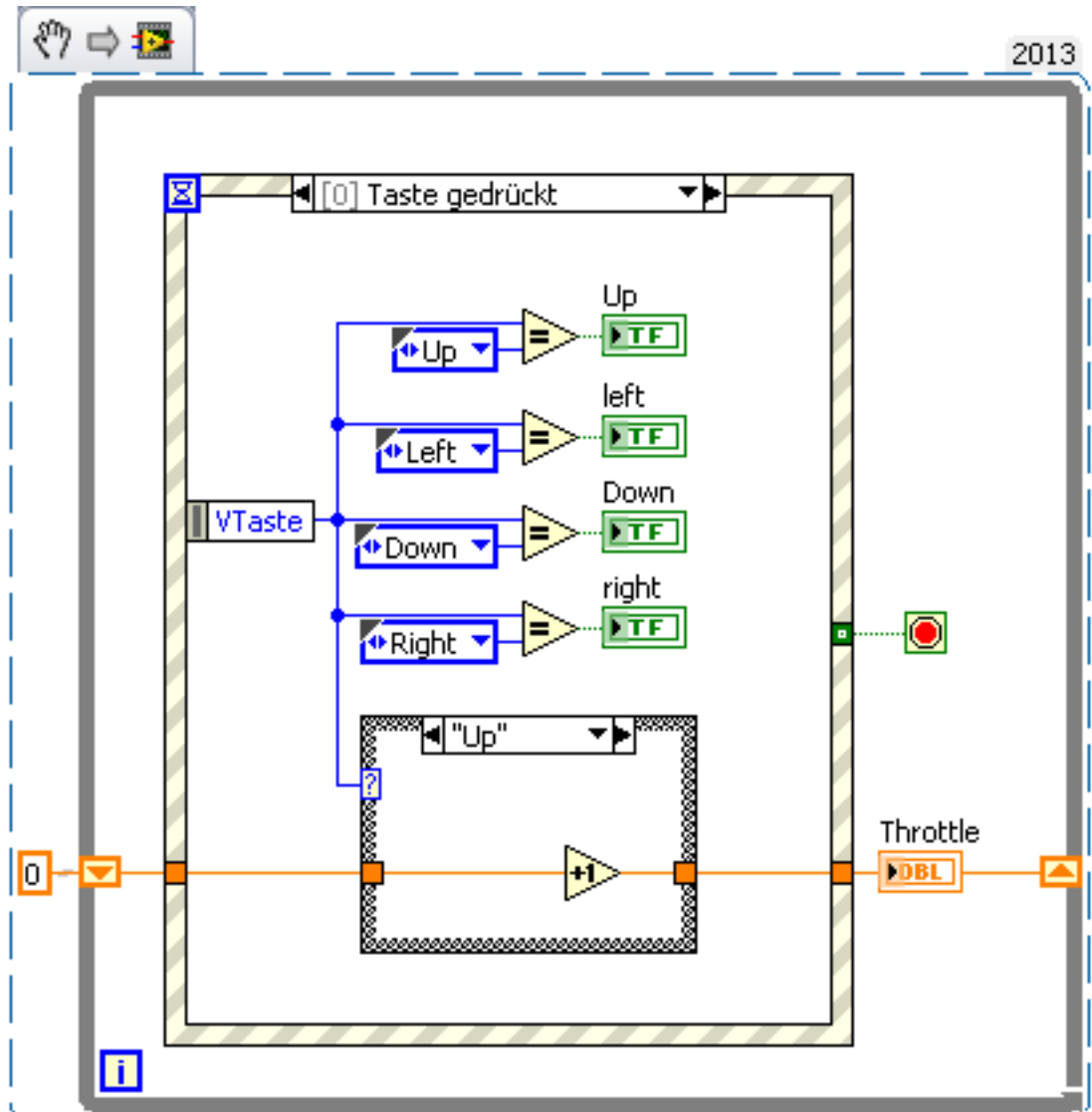
Elágazás (Case Structure)



- A bemenet függvényében hajt végre programot (sub Diagram)
- Egyszerre csak egy esetet futtat
- A Case Selector Label-lel válthatunk állapotok között
- Selector Terminal-lal kötjük be a bemenetet és választunk a végrehajtando diagrammok közül
- Selector Terminal:
 - Boolean: true, false
 - Integer: bármennyi lehet, de Default KELL! Tartományt is meg lehet adni.
 - String: Case sensitive. bármennyi lehet, de Default KELL!
 - Enum: alapértéket lehet megadni a felsorolás minden elemének.
 - Error Cluster: Error Case, No Error Case. Terminálishoz Error Cluster-t illesztve lefut a program ha van hiba, másik fut le, ha nincs hiba.

Elágazás (Event Driven Programming)

- A program egy esemény bekövetkeztére vár, mielőtt az esemény következményeként megírt program lefutna.
- Event: Asszinkron jelzés valamilyen esemény következtében.
 - Felhasználói esemény
 - Külső I/O
 - Program részei



Modularitás (subVI)

- Diszkrét blokkokra lehet osztani a programot.
- Ismétlődő blokkokat újra fel lehet használni.
- C függvényhívás.
- LabVIEW modulok a subVI-k.
- Előnye: modul változtatása könnyű.

