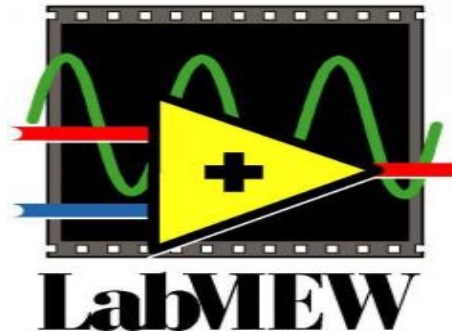


The LabVIEW graphical programming environment



Taik Salma, Guo Jian
Training Project Laboratory 2020/2021

Goals of this Laboratory

- Understand the components of a Virtual Instrument,
- Understand front panels, block diagrams, icons and connector panes,
- Use LabVIEW to create applications,
- Create and save programs in LabVIEW so you can use them as subroutines.

➔ What is LabVIEW (Laboratory Virtual Instrument Engineering Workbench)

- Graphical development platform from National Instruments
- Uses graphical diagrams rather than lines of text which makes it more intuitive for engineers (no statements or syntax rules are needed to program)
- The programs are called Virtual Instruments (VIs)
- It can be used to communicate with hardware
- Deployed for multiple purposes, such as control, data analysis, design etc.

Accessing Labview (Method 1- Virtual machine)

- 1) Login fured.cloud.bme.hu with your own **edu** id (6 number)
- 2) Click "new" and choose version "Matlab R2018a + LabVIEW 2018 (Win10) Windows 10"
- 3) Search app in your computer "Remote desktop connection", then copy and paste **computer id user name and password** from the website.
- 4) Enter the virtual machine and open Labview.

The screenshot shows the Windows Remote Desktop Connection window. A blue callout bubble points to the 'Computer' field, which contains 'vm.smallville.cloud.bme.hu:15714'. A red circle highlights the 'Host' field in the background window, which also contains 'vm.smallville.cloud.bme.hu:15714'. Red arrows point from the 'User name' field (containing 'cloud') and the 'Password' field (containing 'XXXXXX') in the background window to the 'User name' and 'Password' fields in the foreground window. The foreground window also shows a 'Show Options' button and a 'Connect' button. A 'Windows Security' dialog box is open over the foreground window, prompting for credentials.

vm.fured.cloud.bme.hu:XXXXXX

Remote Desktop Connection

Connection to

Protocol RDP

Host vm.smallville.cloud.bme.hu:15714

Host (IPv6) cloud-27973.vm.smallville.cloud.bme.hu:33

Username cloud

Password

Command

Connect (download client)

Computer: vm.smallville.cloud.bme.hu:15714

User name: cloud

Saved credentials will be used to connect to this computer. You can [edit](#) or [delete](#) these credentials.

Show Options

Windows Security

Enter your credentials

These credentials will be used to connect to vm.smallville.cloud.bme.hu.

cloud

Password

☒ Remember me

More choices

OK Cancel

Hide Options Connect Help

Accessing Labview (Method 2- Installing on your laptop)

- 1) Register an account with University email (xxx@edu.bme.hu) and Login <https://www.ni.com/hu-hu/shop/labview/labview-details.html>
- 2) Download free trial for Labview (version 2018 or above) based on the configuration of your computer
- 3) Active your software with the active code from iit department “m73x79699”

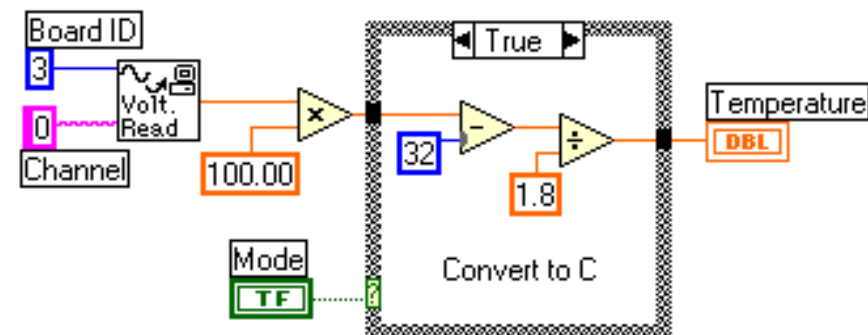
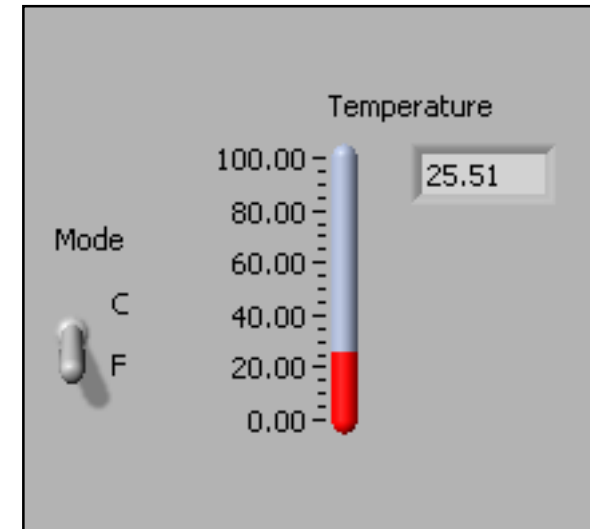
❖ Virtual Instruments (VIs)

1) Front Panel (User Interface)

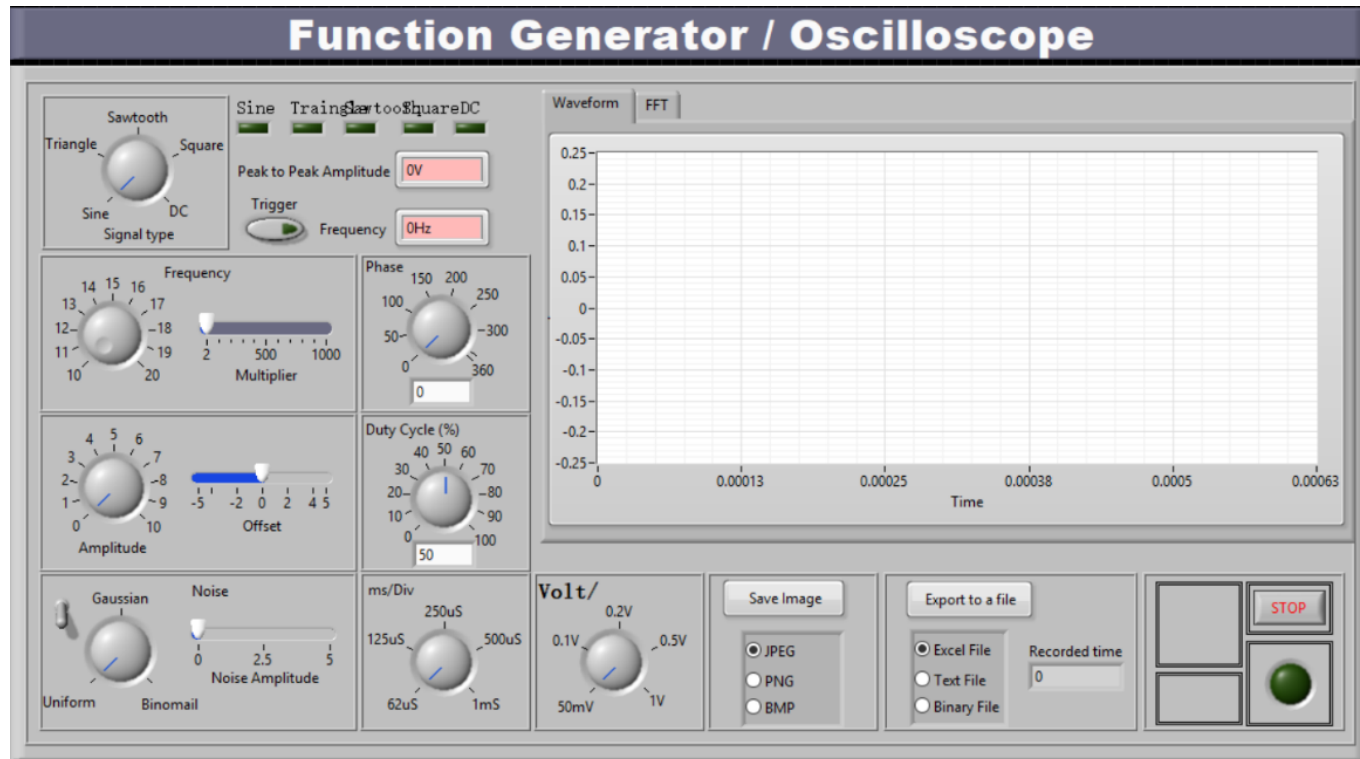
- **Controls** = Inputs
- **Indicators** = Outputs

2) Block Diagram

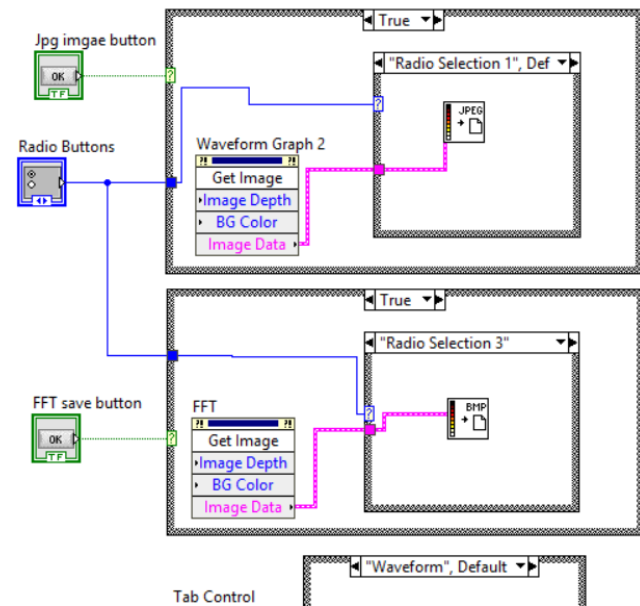
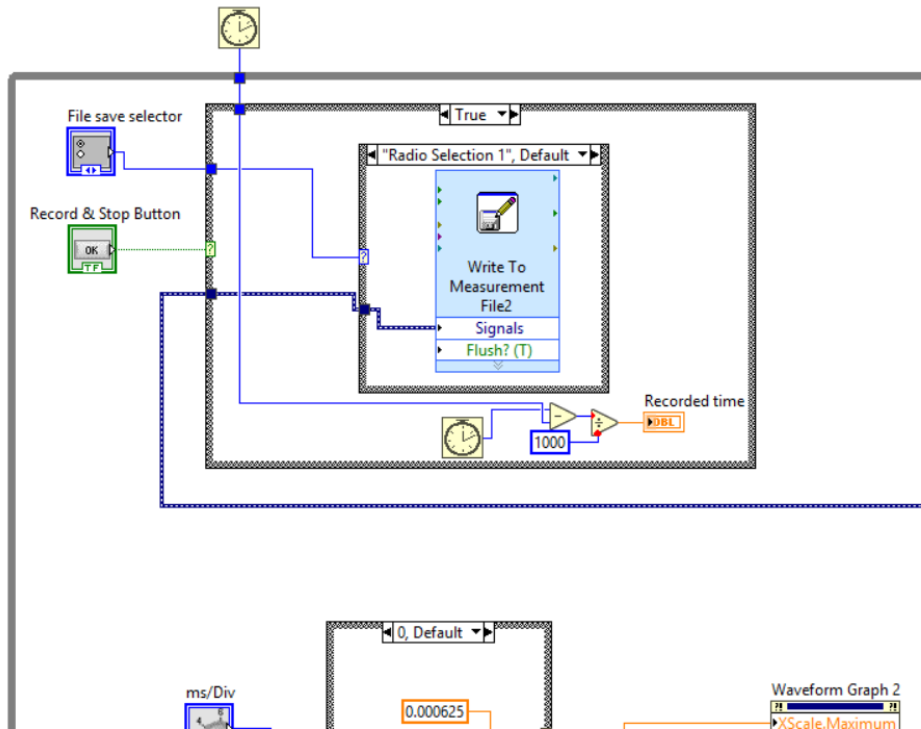
- Accompanying “program” for front panel
- Components are “wired” together
- Data travels from controls to indicators



❖ VI Front Panel

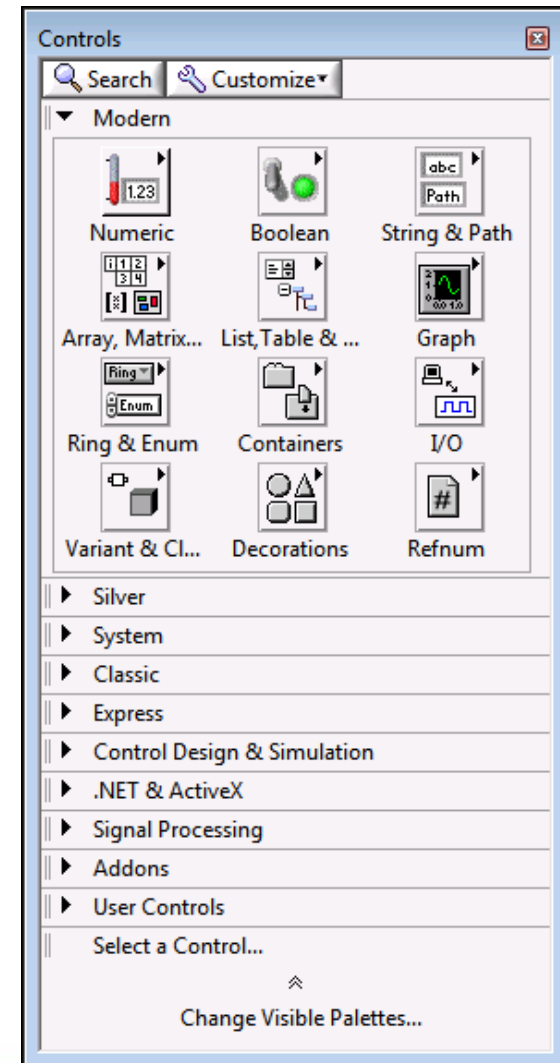


❖ VI Block Diagram



❖ Front Panel: Controls Palette

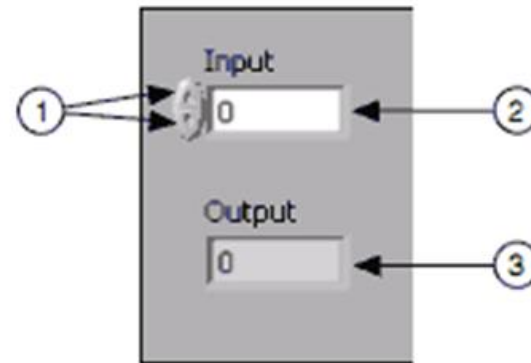
- It is available for the Front Panel only,
- Contains **controls** and **indicators** to create the front panel,
- Equivalent icons to the controls and indicators are created in the block diagram, called **Terminals**.



❖ Controls and Indicators types

➤ **Numeric**

- (1) Increment/Decrement Buttons
- (2) Numeric Control
- (3) Numeric Indicator



➤ **Boolean**

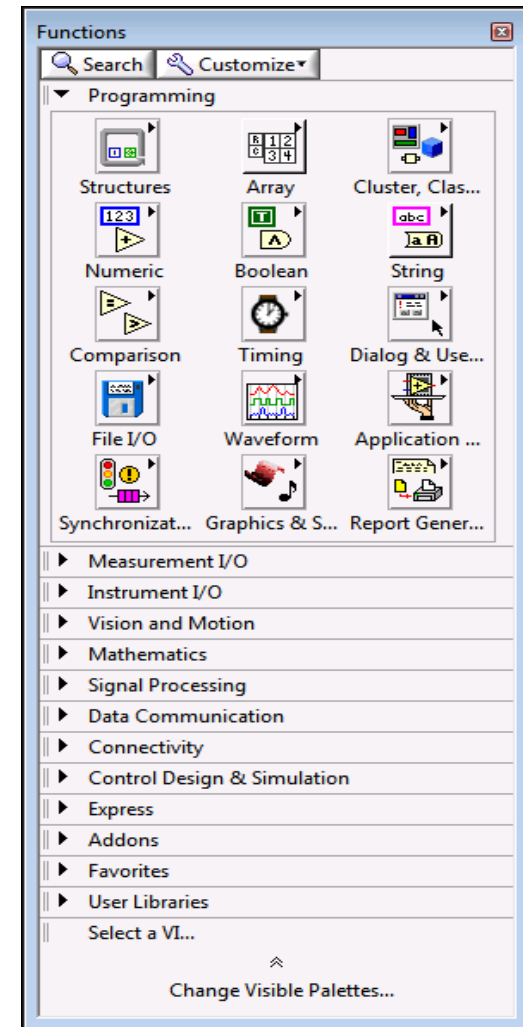


➤ **String**



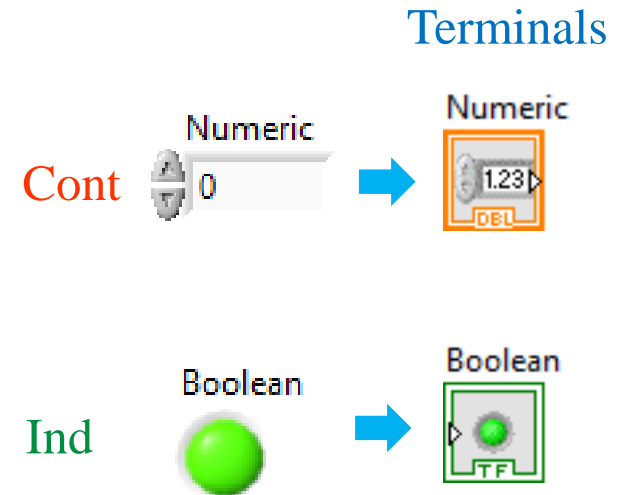
❖ Block Diagram: Functions Palette

- Contains the VIs, functions and constants you use to create the Block Diagram.



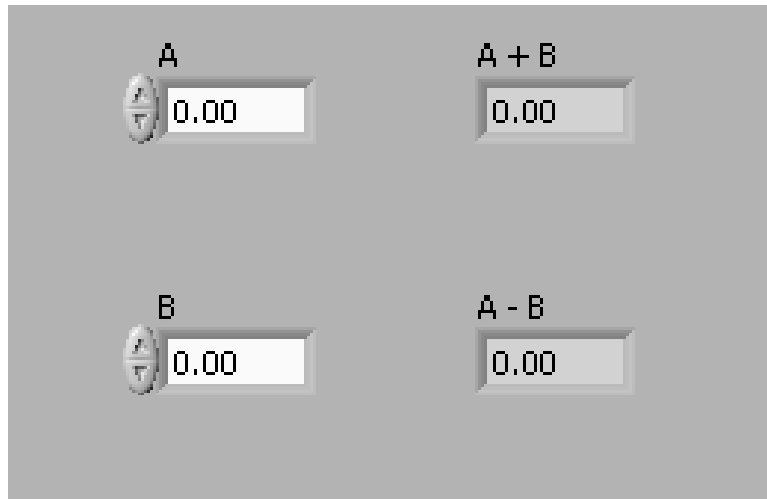
❖ Block Diagram: Terminals

- The corresponding icons for the **Controls** and **Indicators** of the Front Panel, which appear in the Block Diagram
- Created in the Block Diagram once the **Controls** and **Indicators** are created in the Front Panel
- Terminals are entry and exit ports that exchange information between the front panel and block diagram.

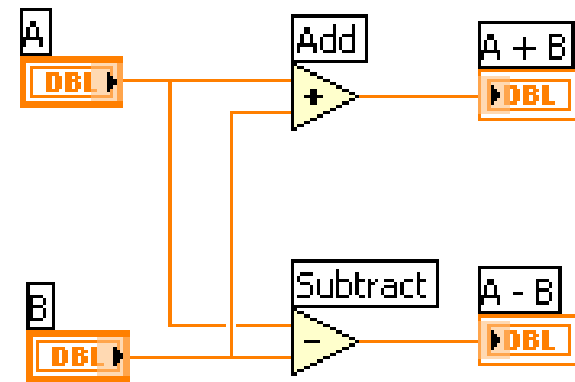


❖ Example: Creating a VI

Front Panel Window



Block Diagram Window




Control
Terminals

Indicator
Terminals

❖ Exercise 1 –Area of a Triangle

Area of a Triangle

Height (cm)

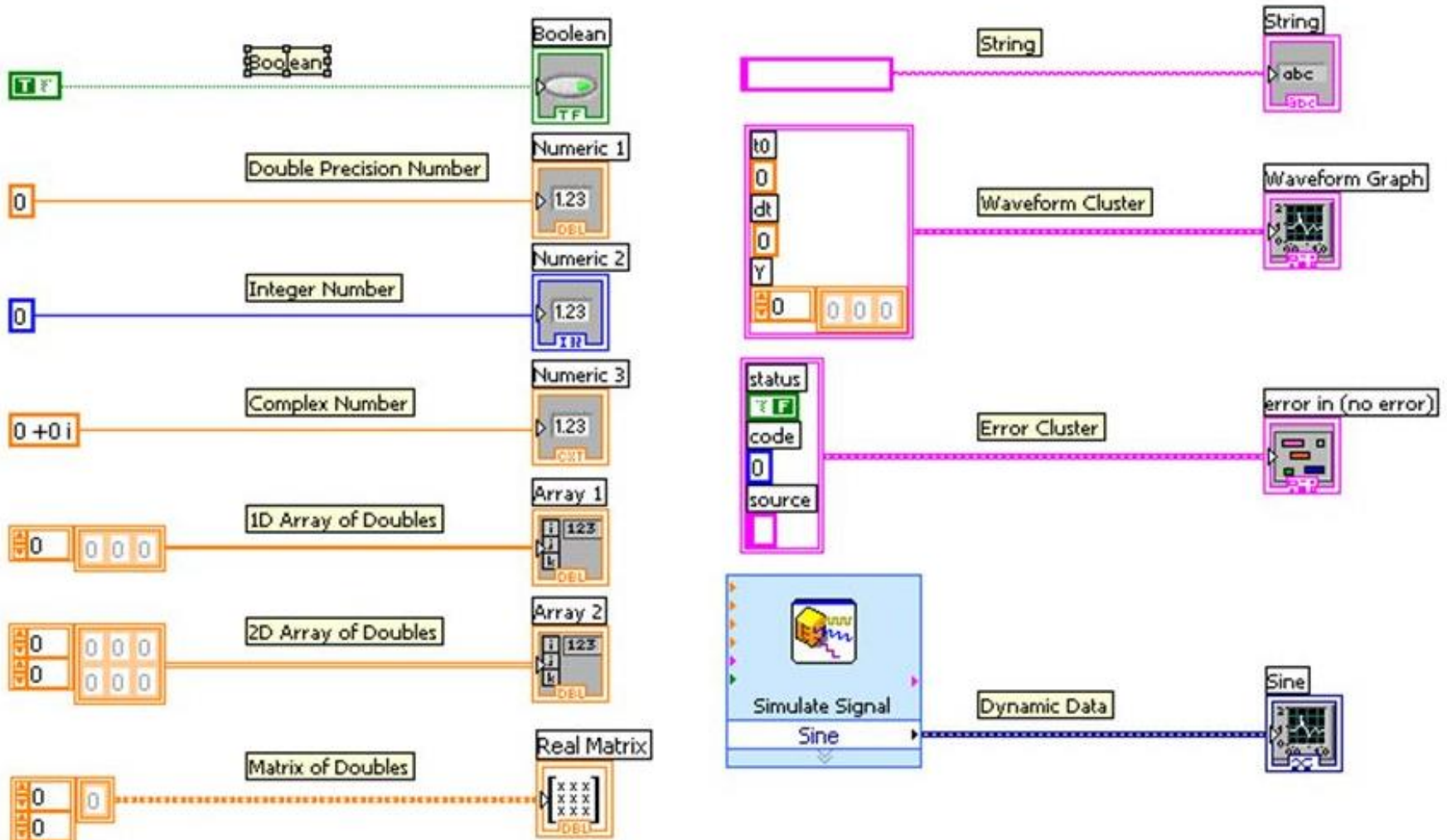


The diagram shows a triangle with a vertical dashed line from the top vertex to the base, labeled "Height" with arrows at both ends. Below the base, a horizontal dashed line with arrows at both ends is labeled "Base".

Area (cm²)

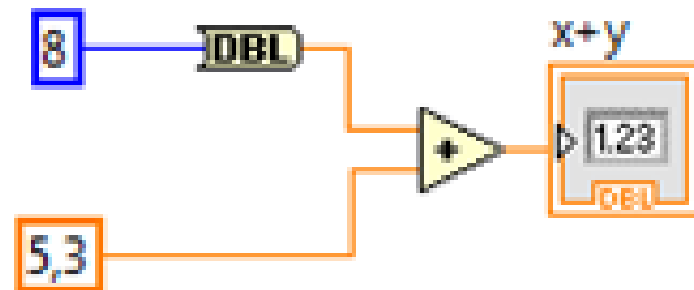
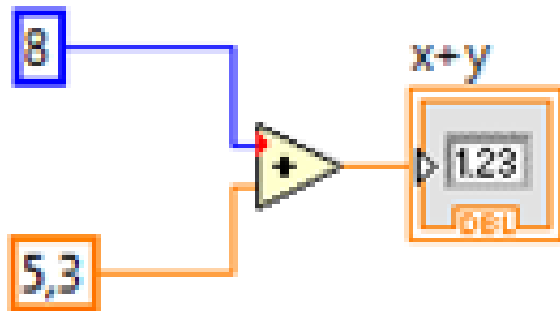
Base (cm)

❖ Major datatypes in LabVIEW: Simple datatypes



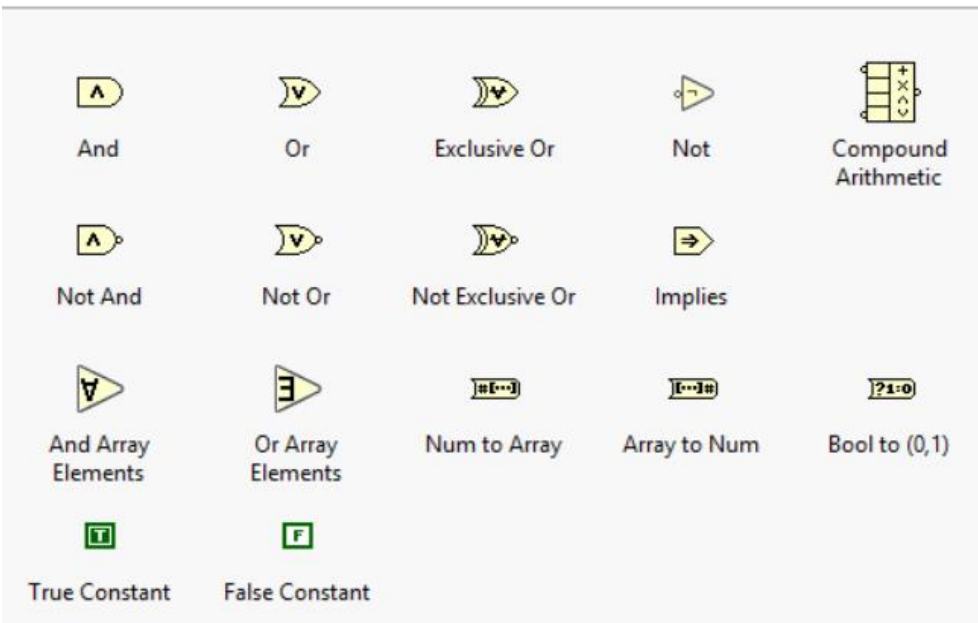
❖ Conversion of data in LabVIEW

Where this is not a misunderstanding, there are polymorphisms for each data type during operations, where manual conversions can be done manually.



❖ Datatype: Boolean

Boolean, or **boolean** logic, is a subset of algebra used for creating true/false statements. **Boolean** expressions use the operators AND, OR, XOR, and NOT to compare values and return a true or false result.



AND Gate



INPUT		OUTPUT
A	B	F
0	0	0
0	1	0
1	0	0
1	1	1

NOR Gate



INPUT		OUTPUT
A	B	F
0	0	1
0	1	0
1	0	0
1	1	0

OR Gate



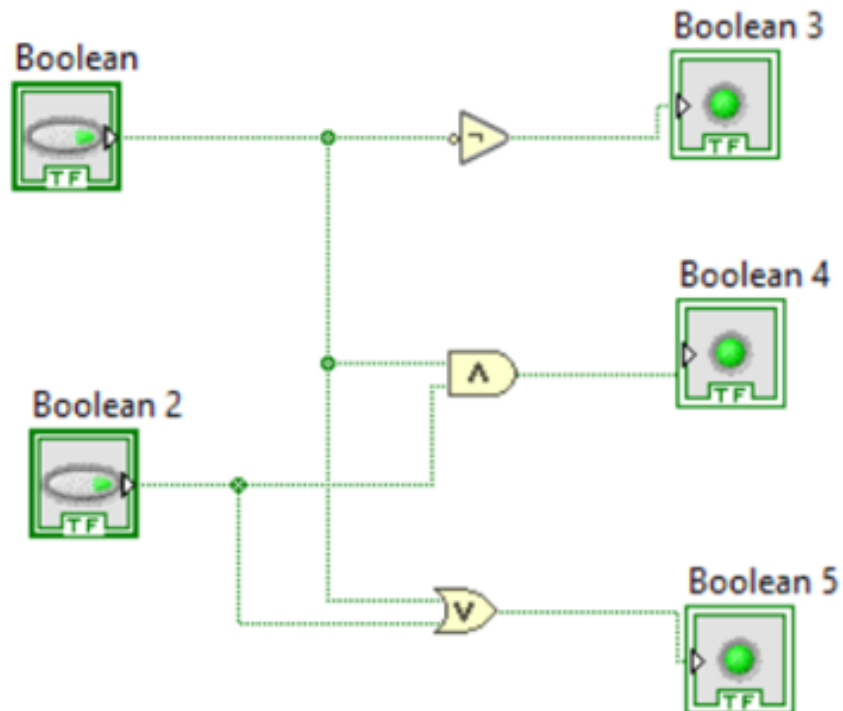
INPUT		OUTPUT
A	B	F
0	0	0
0	1	1
1	0	1
1	1	1

Exclusive OR Gate



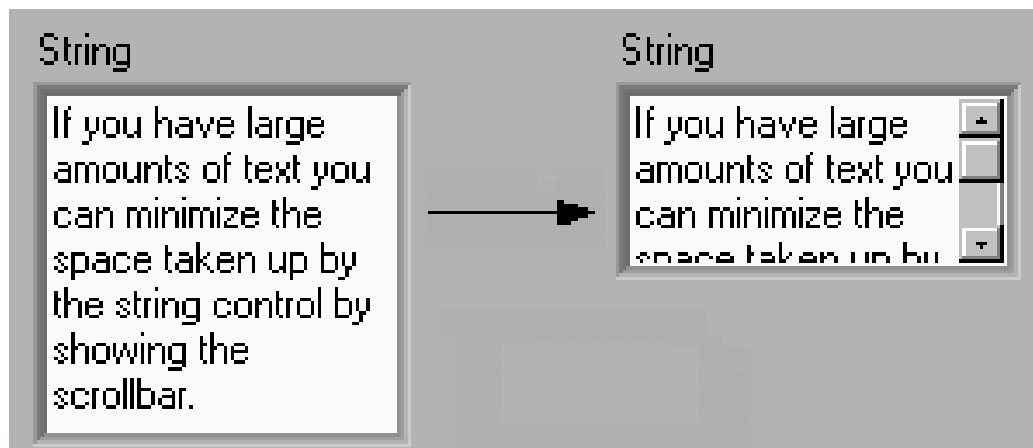
INPUT		OUTPUT
A	B	C
0	0	0
0	1	1
1	0	1
1	1	0

❖ Exercise 2 – Boolean



❖ Strings

- A string is a sequence of displayable or nondisplayable characters (ASCII)
- Many uses – displaying messages, instrument control, file I/O
- String control/indicator is in the Controls»String subpalette



❖ Strings examples

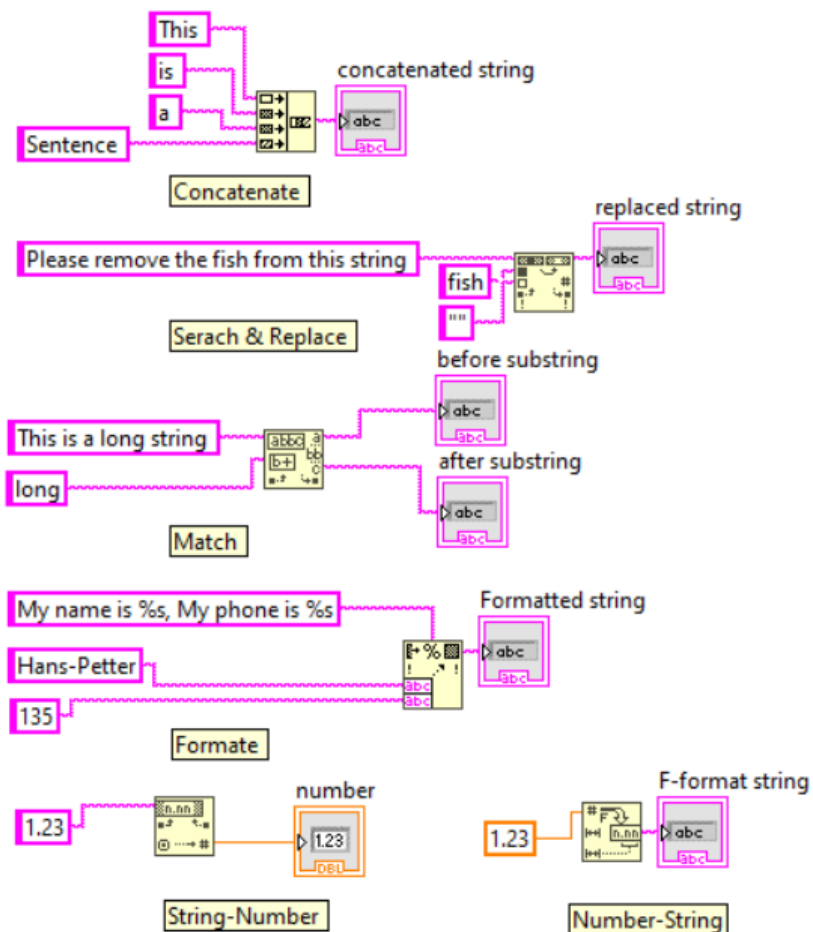
concatenated string
This is a Sentence

replaced string
Please remove the "" from this string

before substring
This is a
after substring
string

Formatted string
My name is Hans-Petter, My phone is 135

number
1.23
F-format string
1.230000

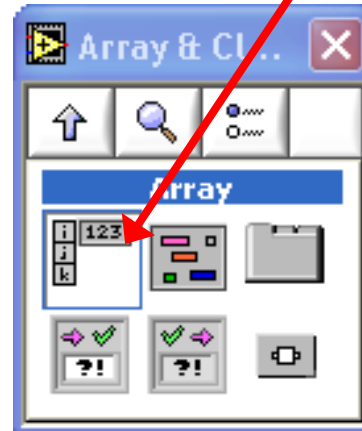
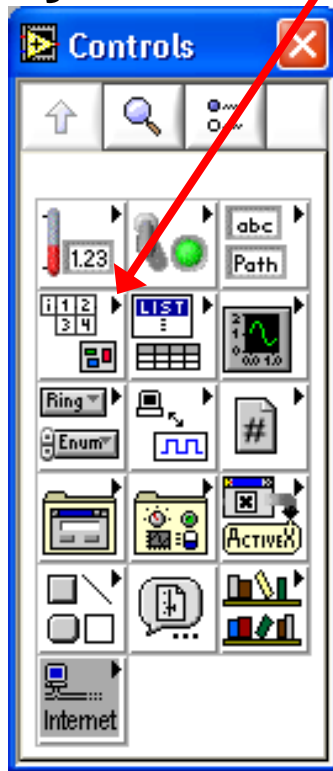


❖ Major datatypes in LabVIEW

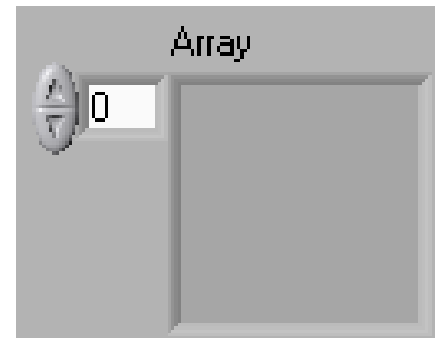
- Complex datatypes:
 - Array
 - Cluster
 - Enum

❖ Adding an Array to the Front Panel

From the **Controls** >> **Array and Cluster** subpalette, select the **Array Shell**

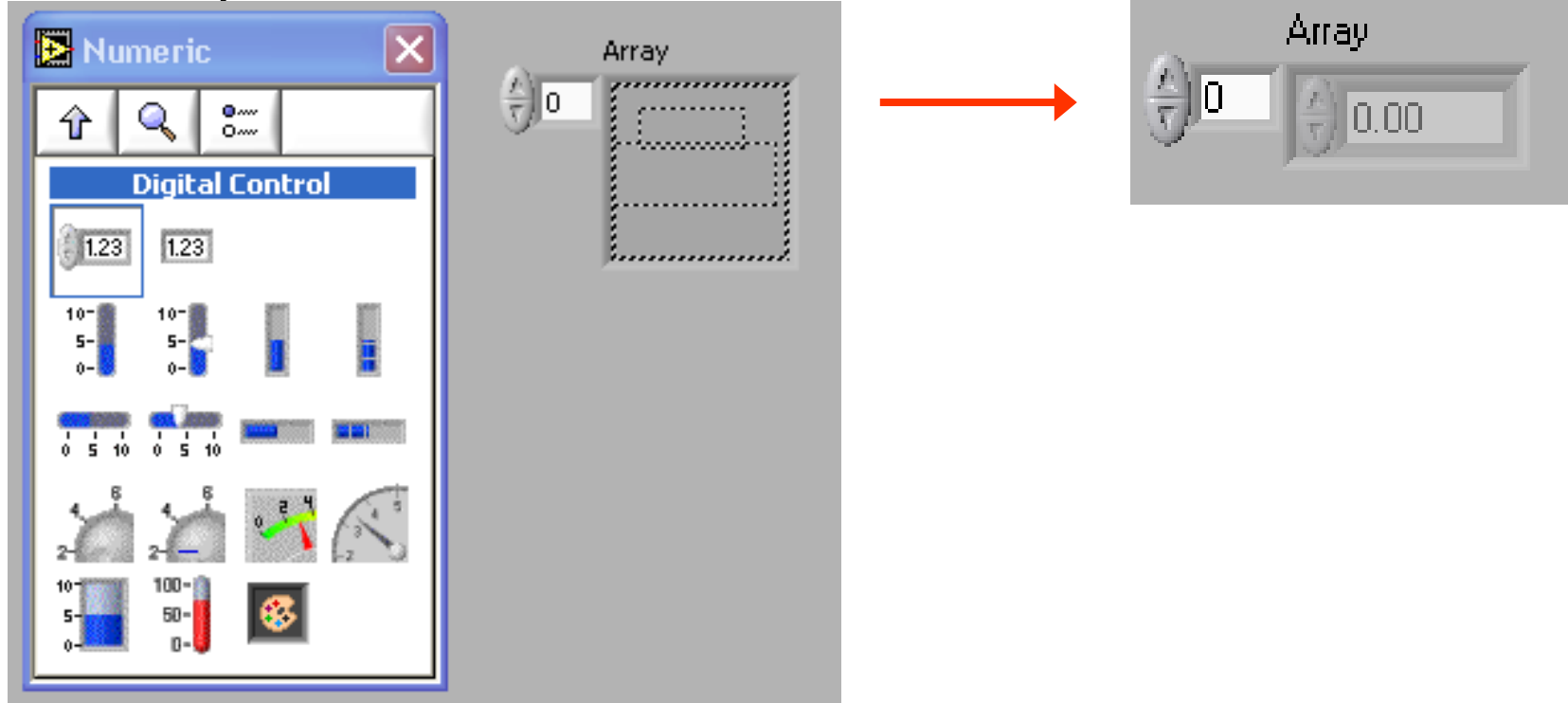


Drop it on the screen.



❖ Adding an Array (cont.)

- Place data object into shell (e.g. digital control).



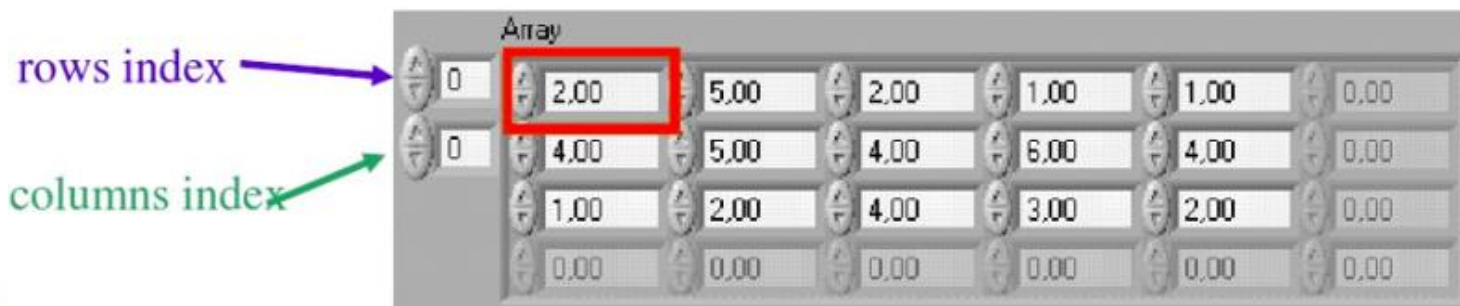
❖ 2D Arrays

The previous examples use 1D arrays. A 2D array stores elements in a grid. It requires a column index and a row index to locate an element, both of which are zero-based. It shows an eight-column by eight-row 2D array, which contains $8 \times 8 = 64$ elements.

Column Index

	0	1	2	3	4	5	6	7
0								
1								
2								
3								
4								
5								
6								
7								

Row Index



❖ Array Functions

The screenshot displays the LabVIEW Functions Programming palette with the 'Array' category selected. The palette is organized into a grid of function icons, each with a label below it. The background shows a LabVIEW block diagram with a numeric control set to 1.23, a 'number' block, and a 'DEL' block.

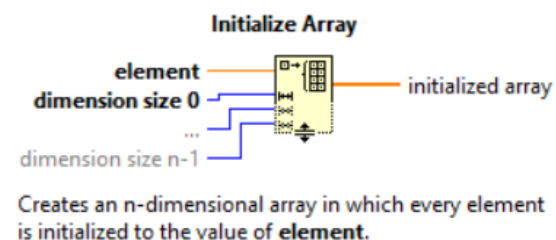
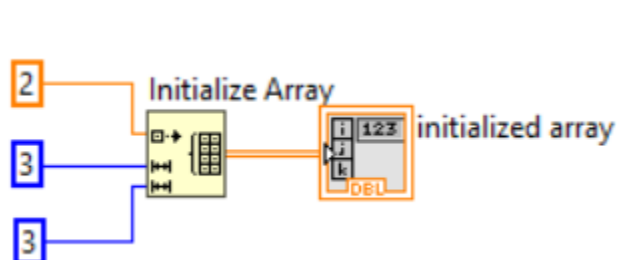
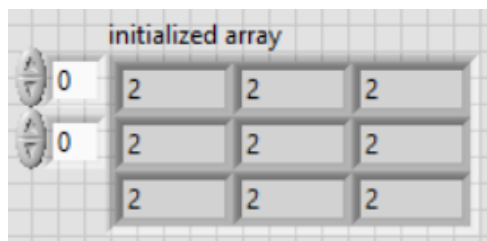
Array Functions:

- Array Size
- Index Array
- Replace Subset
- Insert Into Array
- Delete From Array
- Initialize Array
- Build Array
- Array Subset
- Max & Min
- Reshape Array
- Sort 1D Array
- Sort 2D Array.vim
- Shuffle 1D Array.vim
- Shuffle 2D Array.vim
- Reverse 1D Array
- Search 1D Array
- Split 1D Array
- Rotate 1D Array
- Increment Array Element.vim
- Decrement Array ...
- Interpolate 1D Array
- Threshold 1D Array
- Interleave 1D Arrays
- Decimate 1D Array
- Transpose 2D Array
- Array Constant
- Array To Cluster
- Cluster To Array
- Array to Matrix
- Matrix to Array
- Remove Duplicates From...
- Matrix

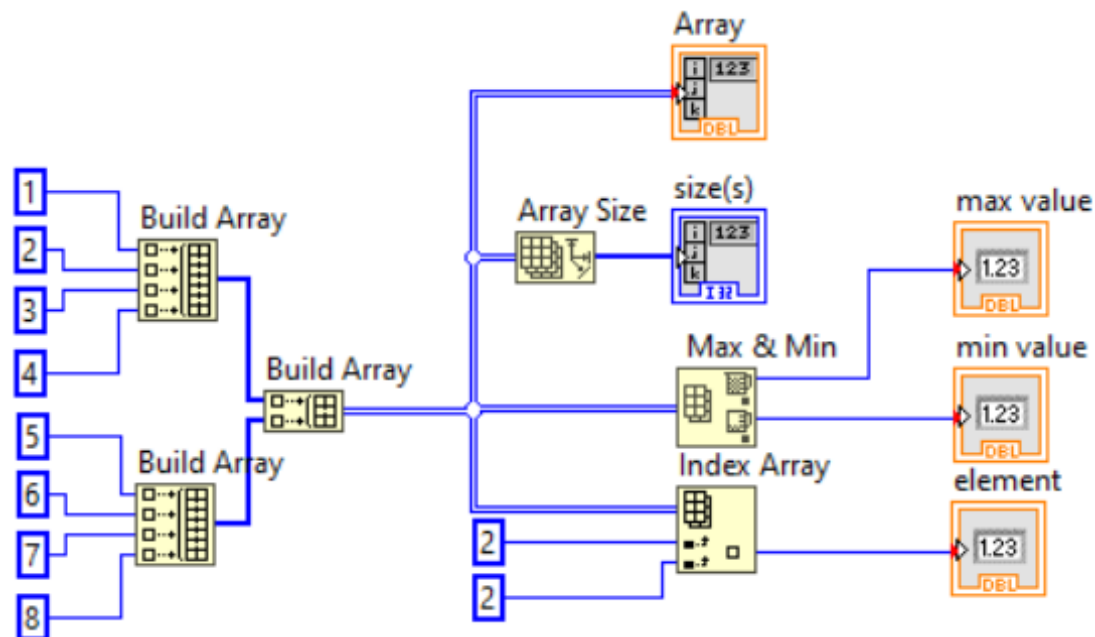
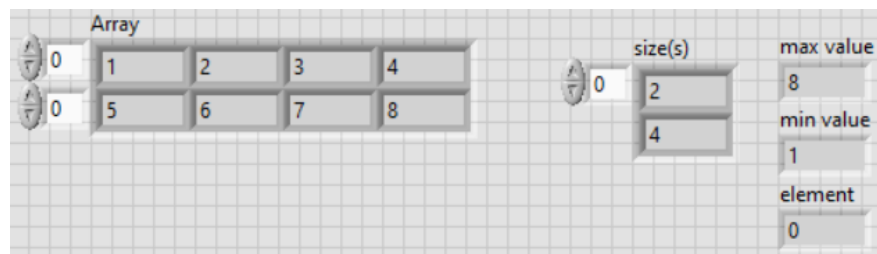
Other Categories:

- Array
- Cluster, Class, & Variant
- Boolean
- String
- Waveform
- Collection
- Timing
- Dialog & User Interface
- Graphics & Sound
- Application Control
- VI Analyzer

❖ Array Examples



[Detailed help](#)



❖ Exercise 4 –String & Array

Array		Array 2				
0	Smith, John	0	First Name	Size	Last Name	Size
	Miller, Arthur		John	4	Smith	5
			Arthur	6	Miller	6

❖ Clusters

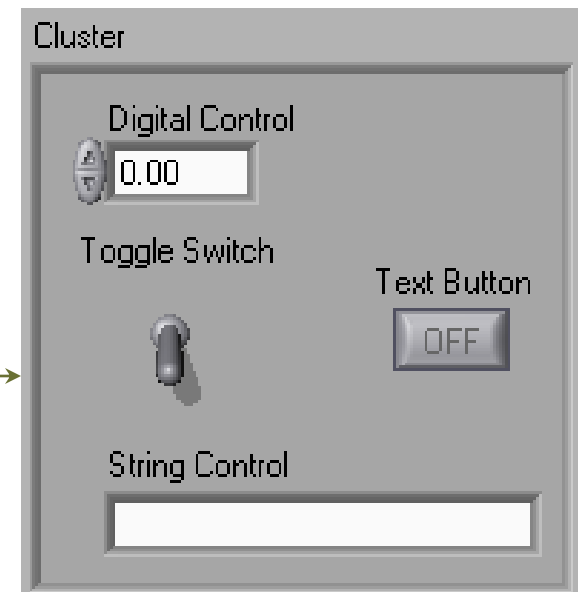
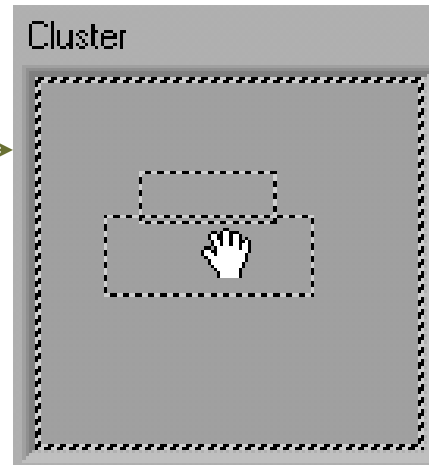
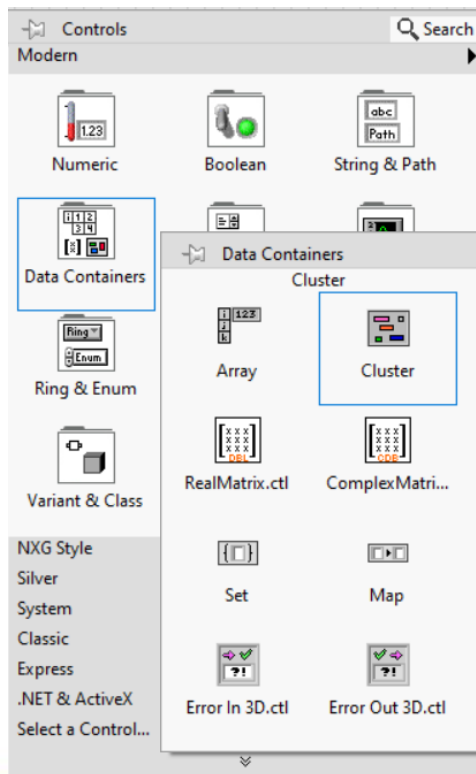
- Data structure that groups data together
- Data may be of different types
- Analogous to *struct* in C
- Elements must be either all controls or all indicators
- Thought of as wires bundled into a cable
- Like an array, a cluster is either a control or an indicator. A cluster cannot contain a mixture of controls and indicators.



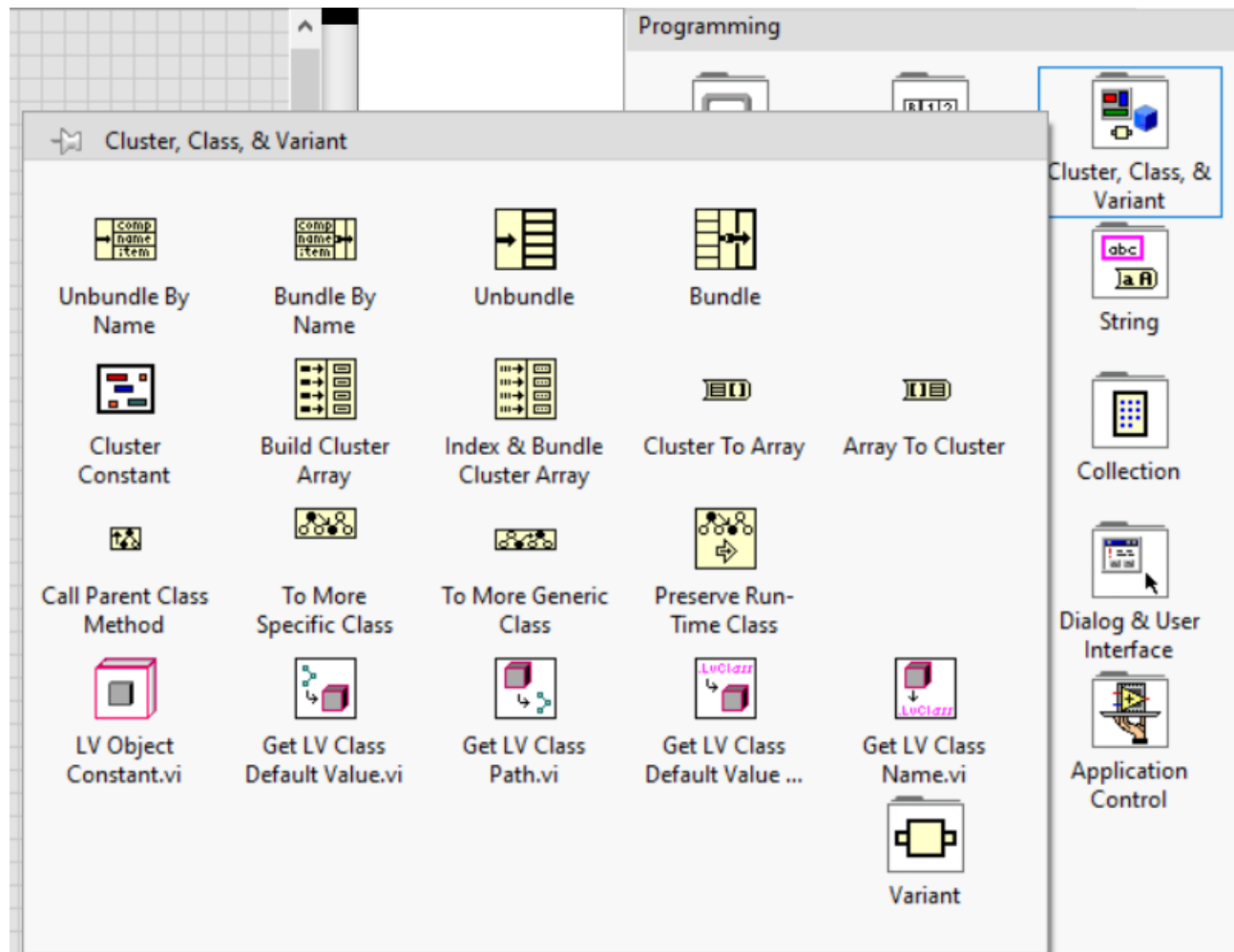
❖ Creating a Cluster

1. Select a **Cluster** shell from the **Array & Cluster** subpalette

2. Place objects inside the shell

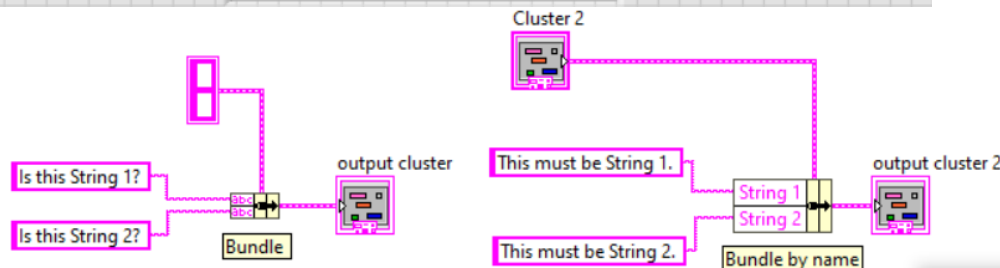
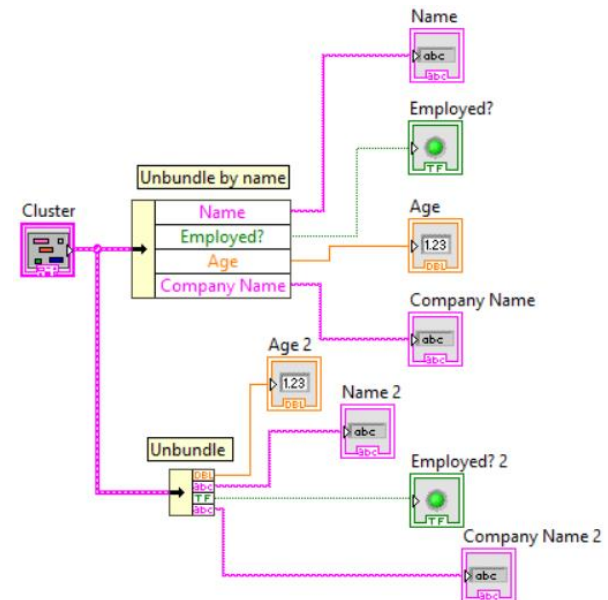
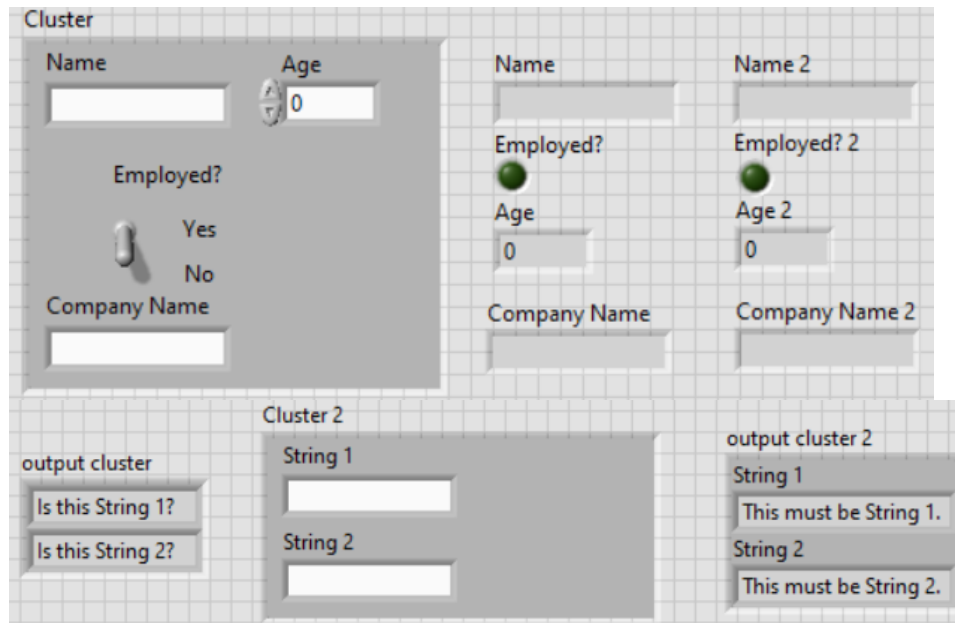


❖ Cluster Functions

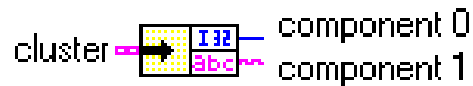


❖ Cluster Examples

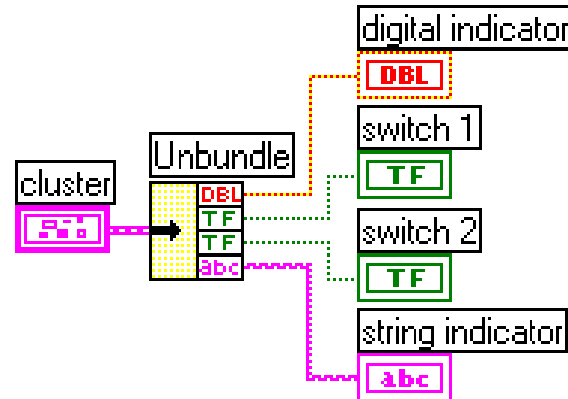
- In the **Cluster** subpalette of the **Functions** palette
- Can also be accessed by right-clicking on the cluster terminal



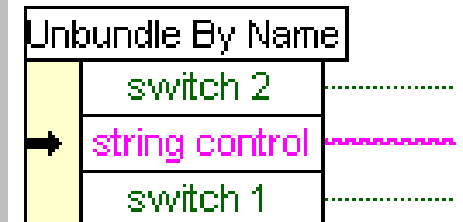
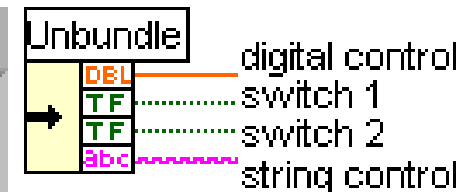
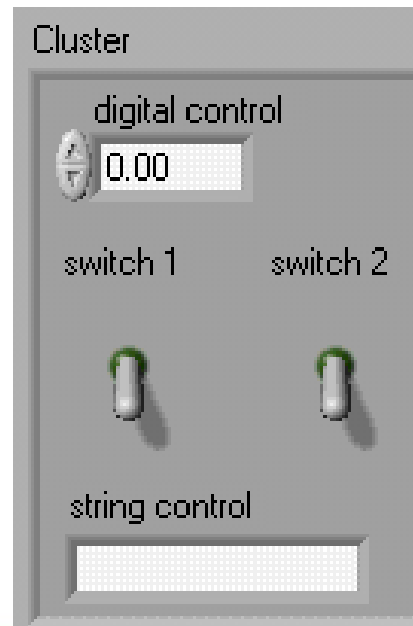
❖ Cluster Examples



Unbundle

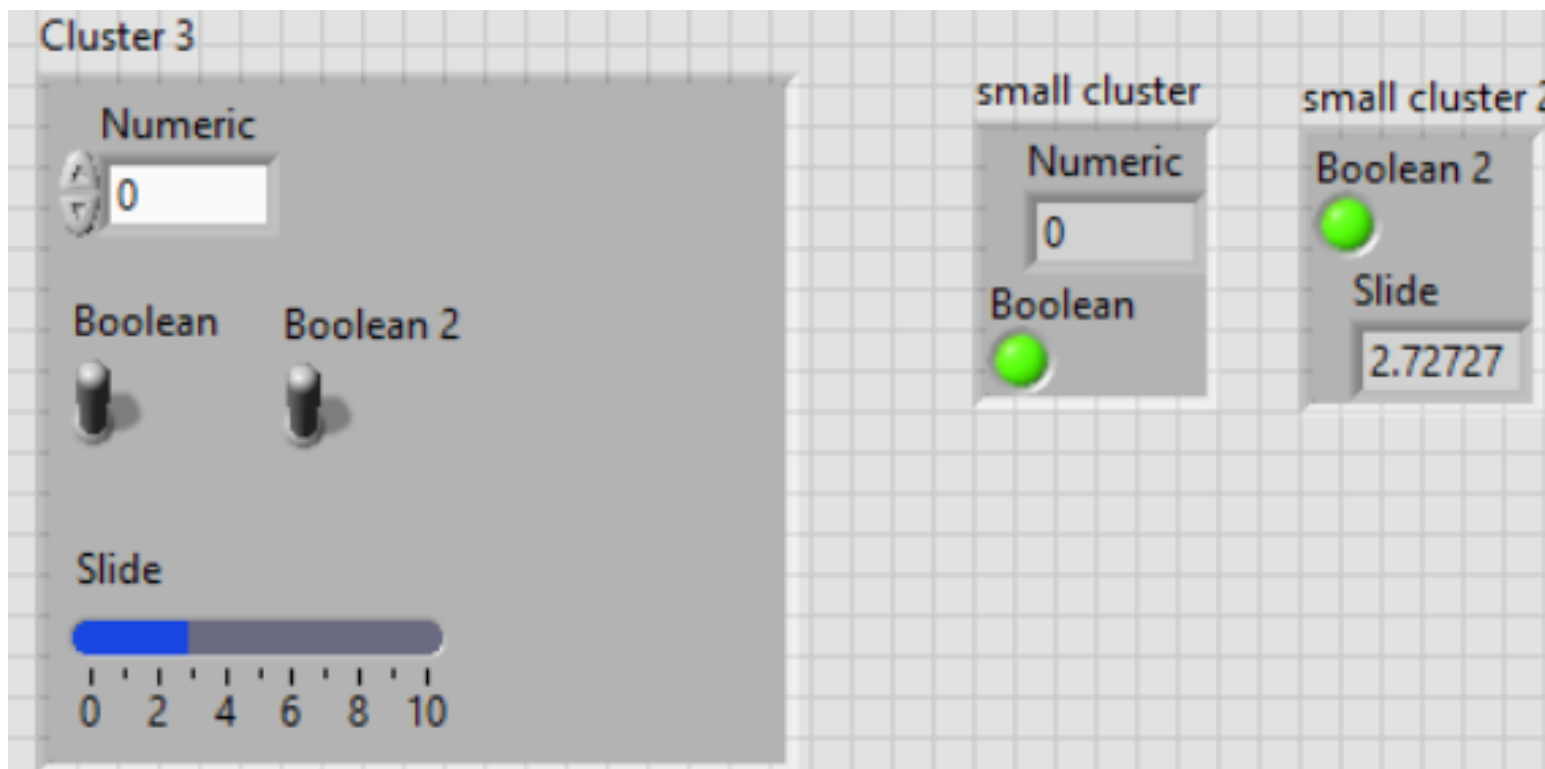


Unbundle By Name



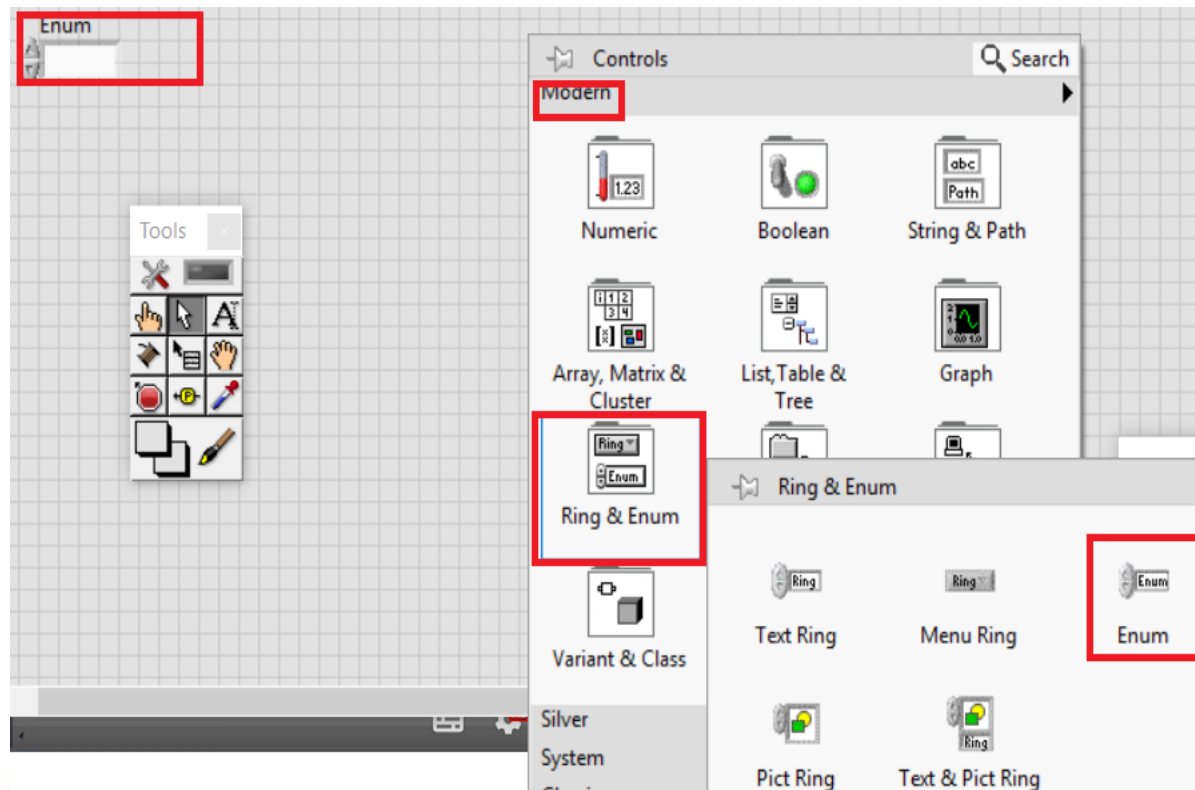
Unbundled cluster in the diagram

❖ Exercise 5 –unbundle and bundle

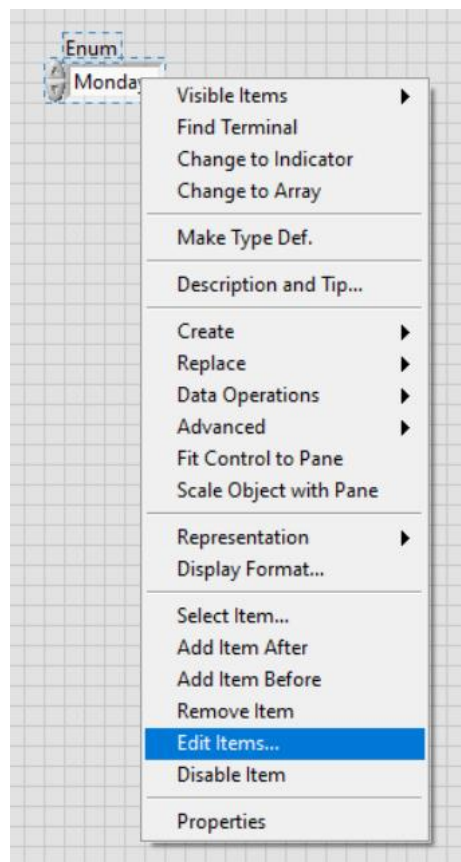


❖ Enums

An enum (enumerated control, constant, or indicator) is a combination of data types. An enum represents a pair of values, a string and a numeric, where the enum can be one of a list of values.



❖ Enum Examples



Items	Values
Monday	0
Tuesday	1
Wednesday	2
Thursday	3
Friday	4
Saturday	5
Sunday	6

