

---

---

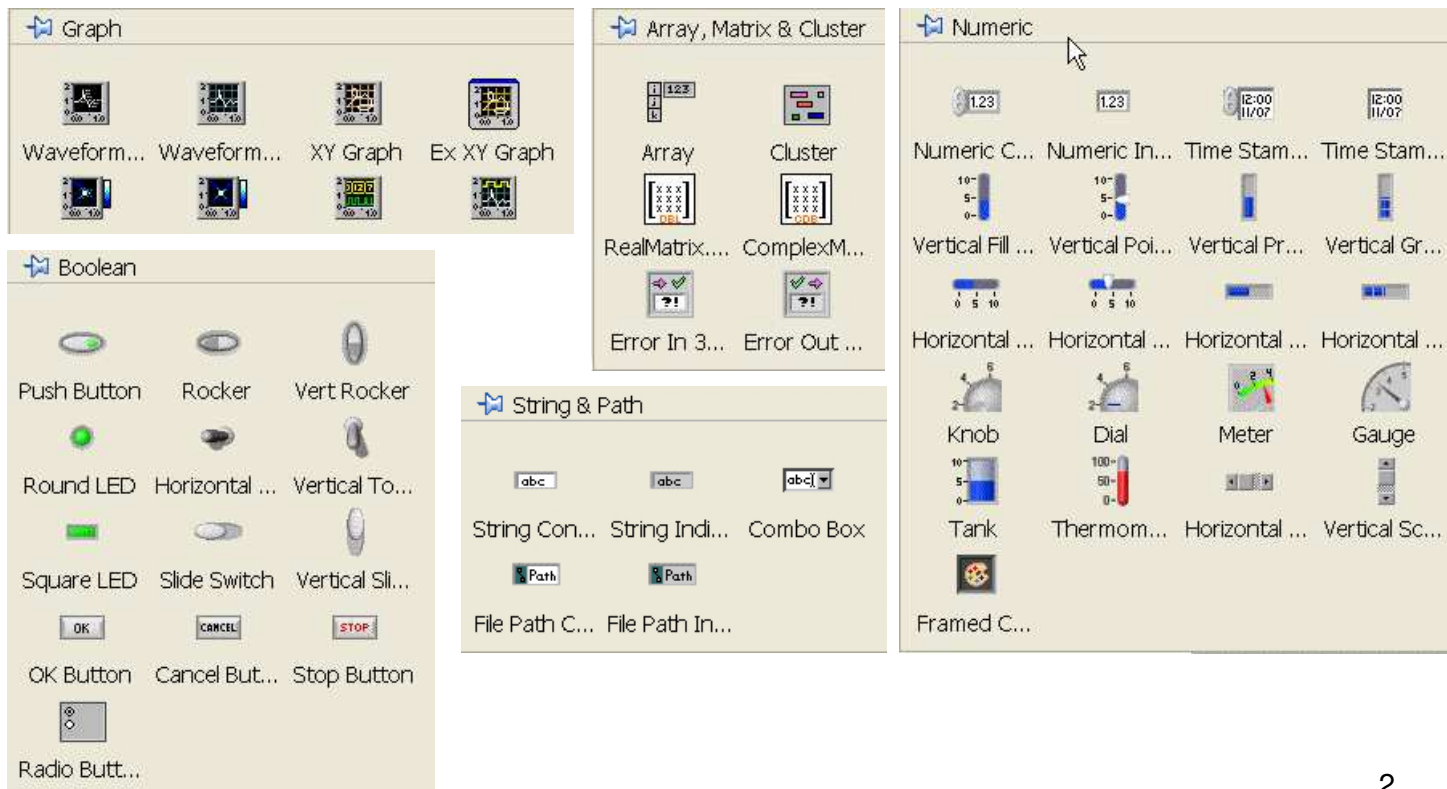
# ***Creating and saving VIs,***

***Roland Pawliczek, PhD***

## Front panel

Start create your application by organizing controls and indicators on front panel. Use **Control Palette** to do it.

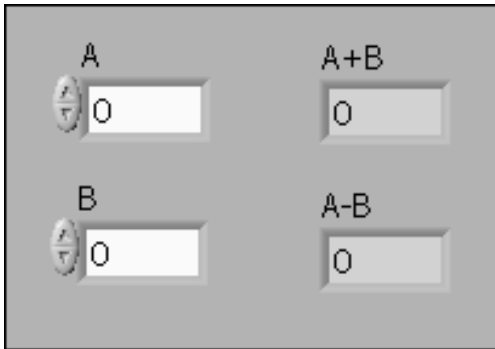
The most popular controls and indicators:



## Front panel

---

Put the control and indicator as presented below:



When you create an object on the Front Panel, a terminal will be created on the Block Diagram (see figure below). These terminals give you access to the Front Panel objects from the Block Diagram code.

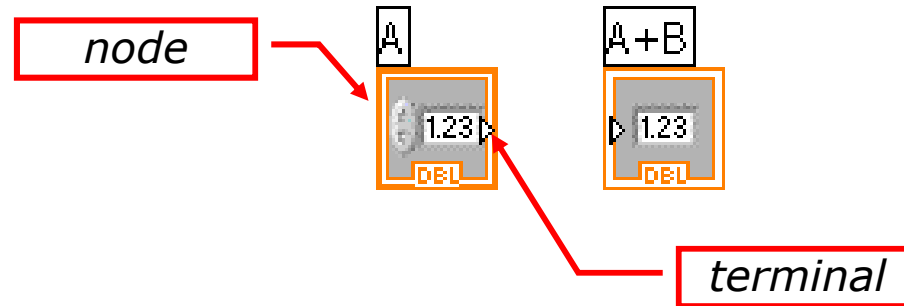
Try double-click any control on Front Panel – system automatically show that element on Block Diagram.

Use the same operation on Block Diagram to find correspond control or indicator on Front Panel.



## Block Diagram

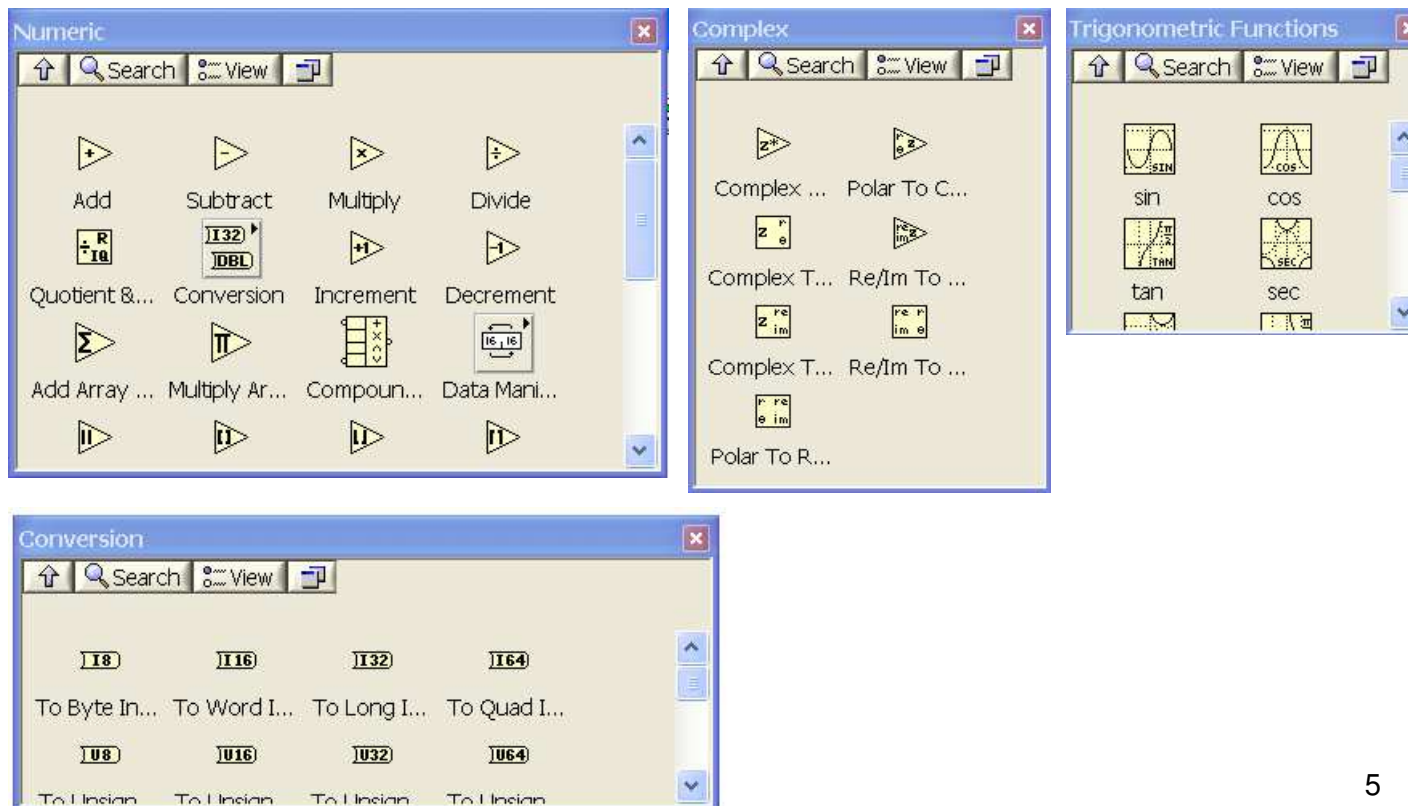
- **Controls have an arrow on the right side and have a thick border. Indicators have an arrow on the left and a thin border.**



- *Logic rules apply to wiring in LabVIEW: Each wire must have one (but only one) source (or control), and each wire may have multiple destinations (or indicators).*
- *For data operations some **functions** are necessary.*

## Block Diagram

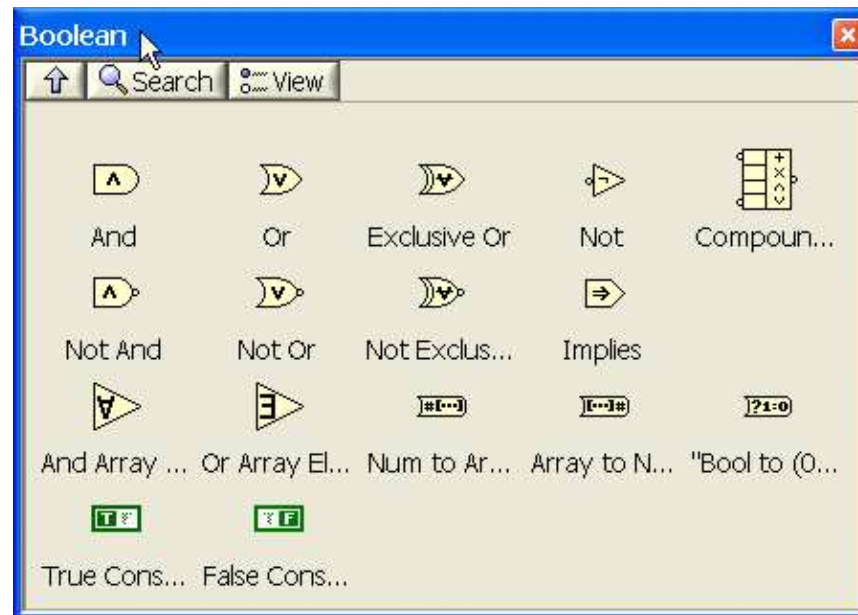
Use the Numeric and Mathematics functions to create and perform arithmetic, trigonometric, logarithmic, and complex mathematical operations on numbers and to convert numbers from one data type to another.



## Block Diagram

Use the Boolean functions to perform logical operations on single Boolean values or arrays of Boolean values, such as the following tasks:

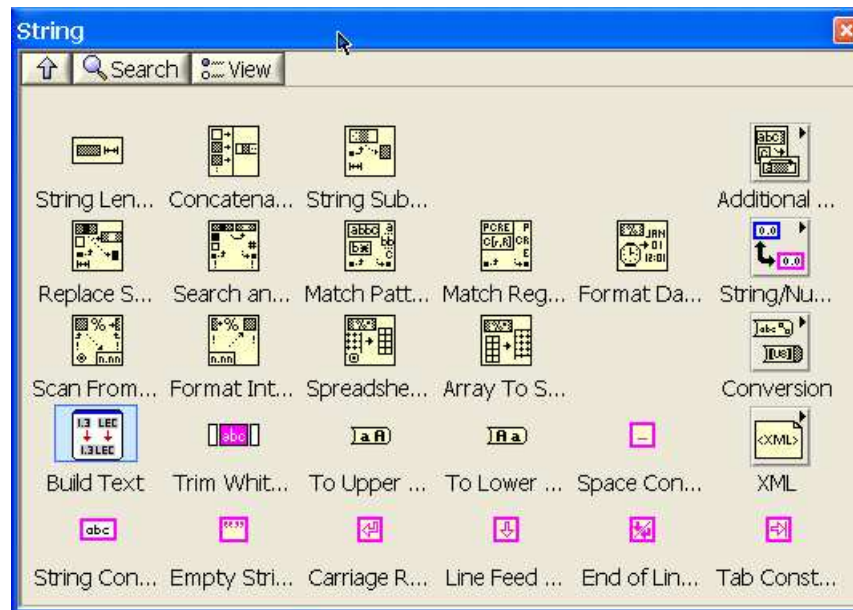
- Change a TRUE value to a FALSE value and vice versa.
- Determine which Boolean value to return if you receive two or more Boolean values.
- Convert a Boolean value to a number (either 1 or 0).
- Perform compound arithmetic on two or more Boolean values.



## Block Diagram

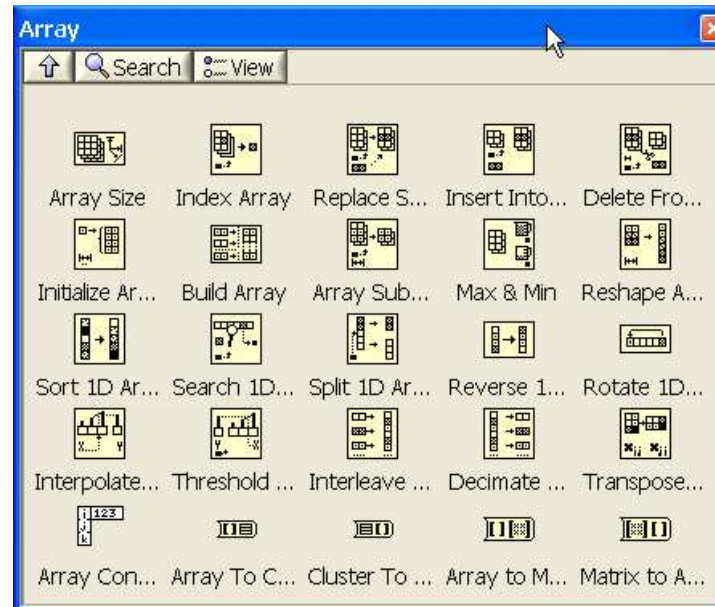
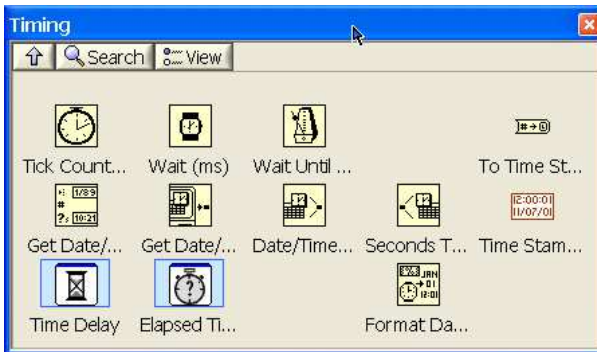
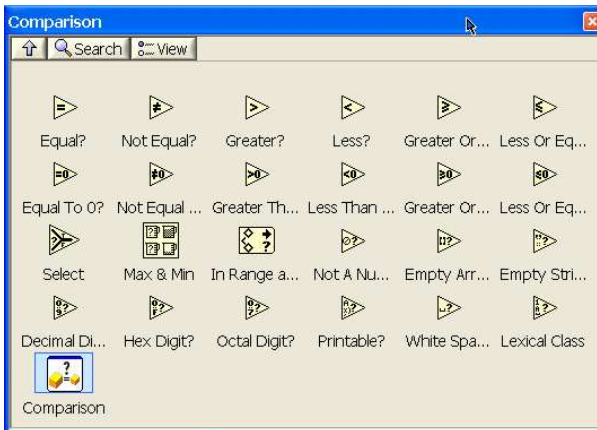
Use the String functions to perform the following tasks:

- Concatenate two or more strings.
- Extract a subset of strings from a string.
- Search for and replace characters or subsets of strings in a string.
- Convert numeric data into strings.
- Format a string for use in a word processing or spreadsheet application.



## Block Diagram

*And many, many other functions...*



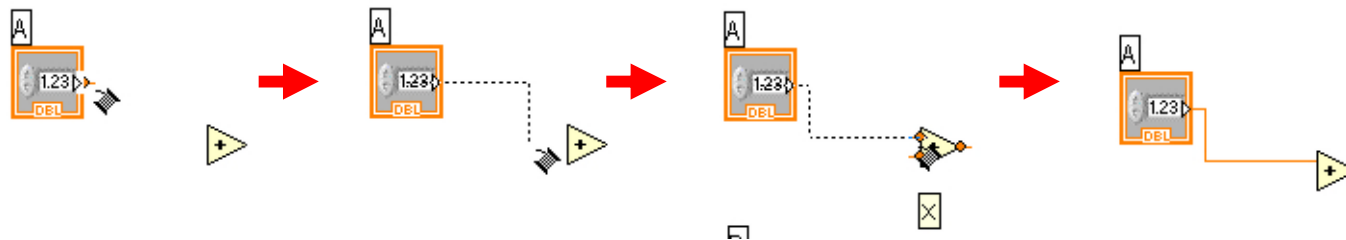


## Block Diagram

---

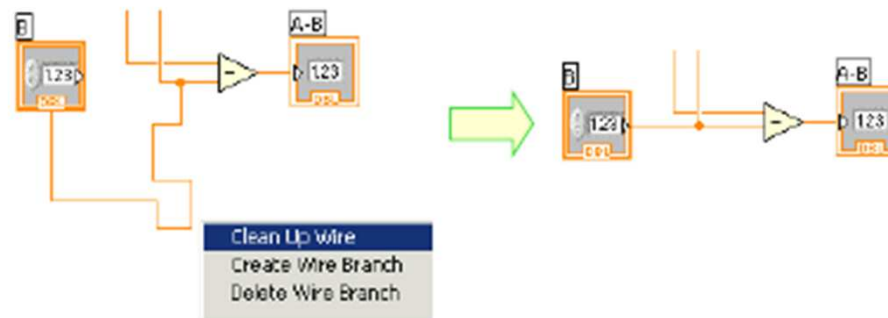


- The **wiring tool** is used to wire to the nodes of the functions. When you "aim" with the wiring tool, aim with the end of the wire hanging from the spool. This is where the wire will be placed.
- As you move the wiring tool over functions, watch for the yellow tip strip. This will tell you the name of the terminal you are wiring to.
- As you move the wiring tool over a terminal, it will flash. This will help you identify where the wire will attach.
- For more help with the terminals, right-click on the function and select **Visible Items>>Terminals**. The function's picture will be pulled back to reveal the connection terminals. Notice the colors- these match the data types used by the front panel terminals.



## Block Diagram

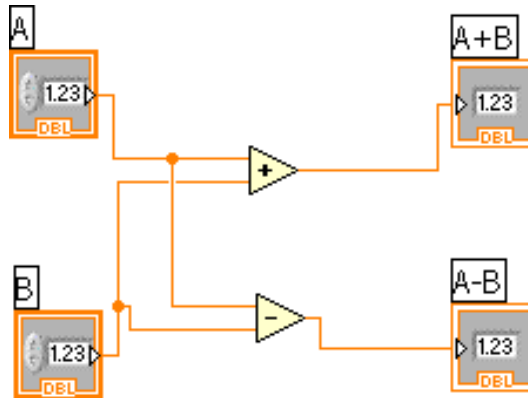
- For additional help, select *Help>>Show Context Help*, or press *CTRL+H*. This will bring up the context help window. As you move your mouse over the function, this window will show you the function, terminals, and a brief help description. Use this with the other tools to help you as you wire.
- If your wiring becomes doesn't look very good, right-click on the particular wire in question and choose **Clean Up Wire** to automatically re-route that wire.



## Block Diagram

---

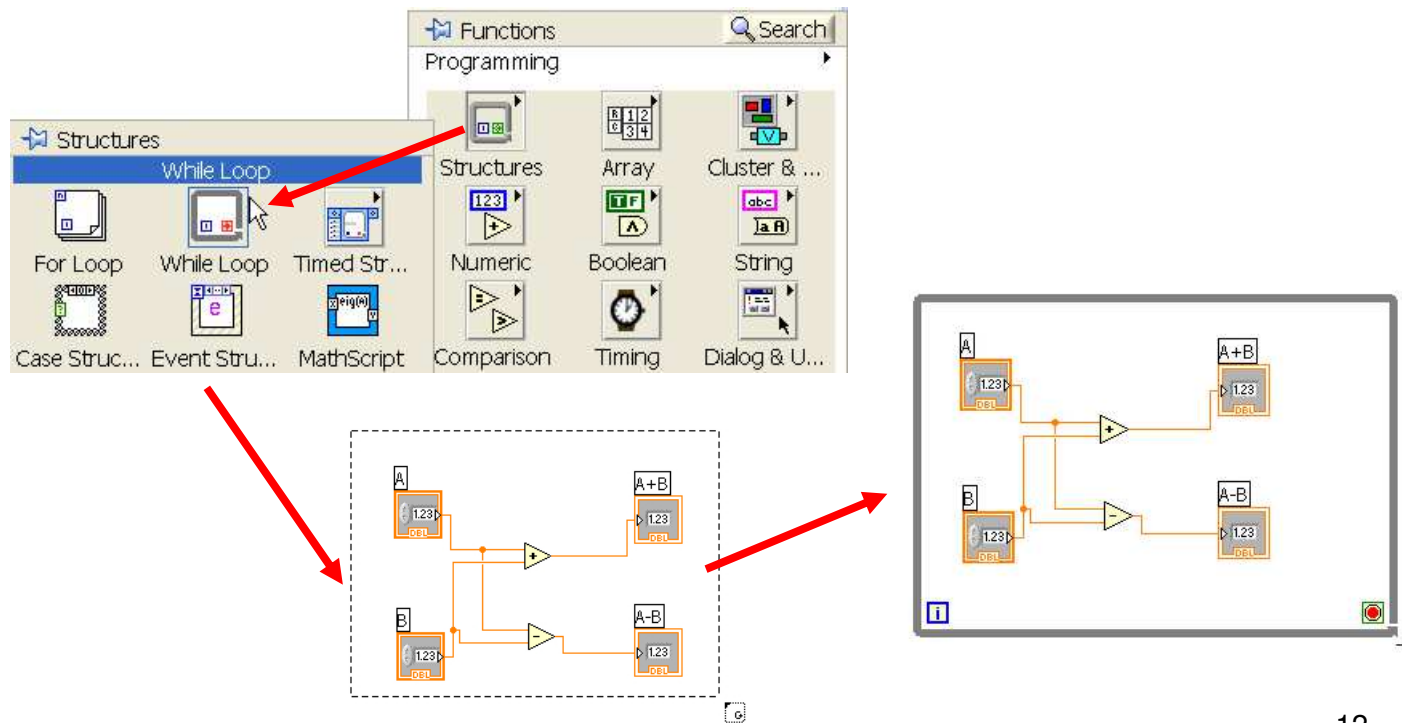
- *Complete wires to the following form:*



- *To make common point on wire just click on it and create next wire.*
- **IMPORTANT: the color of the wire provide the data type. Double-precision, floating point numbers are represented with orange terminals and the letters DBL. Boolean terminals are green with TF lettering ...**

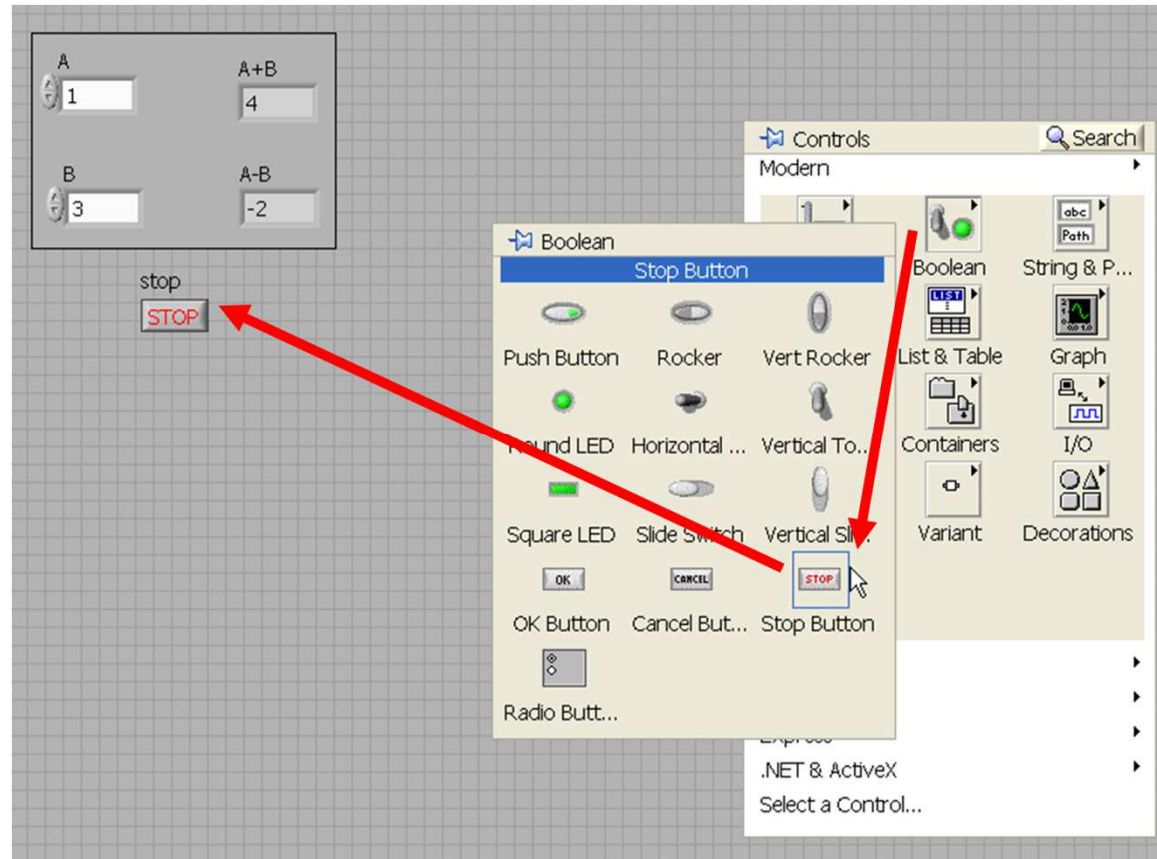
## Block Diagram

- Switch to the Front Panel, set values A and B and run VI.  
**As you can see application is executed only once!**
- To fix this problem choose **While Loop** from **Structures** functions palette and make a rectangular window around all elements on the **Block Diagram**.



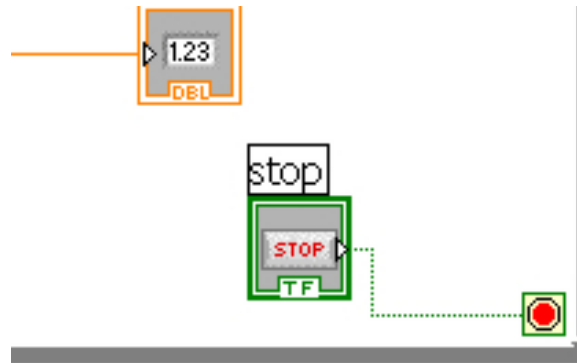
## Block Diagram

- Switch to the Front Panel and place on them **STOP Button** from **Boolean Controls Palette**.



## Block Diagram

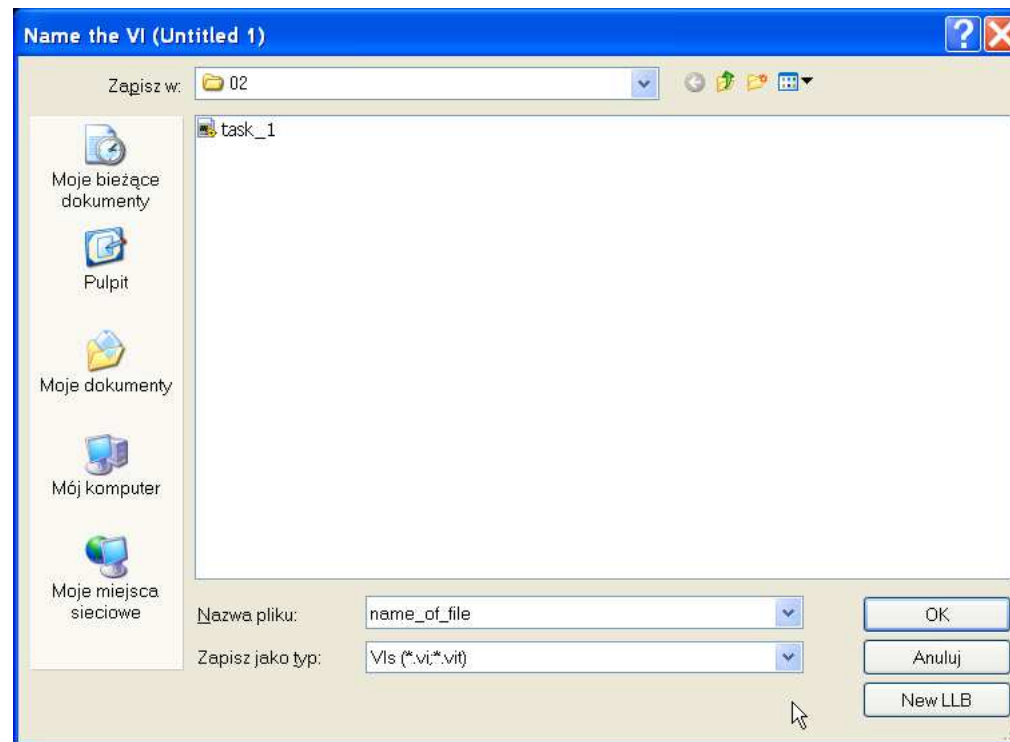
- Switch to the Block Diagram and connect STOP Button terminal as is presented below.



- Run VI once again. This time it will work continuously. To stop execution just press the STOP Button.

## Saving VI ...

*Saving works like saving in any other software. For the first time, you must specify a name of the VI and its destination in the disc hierarchy. Saving modified VI, simply changes the existing VI in a previously specified location.*

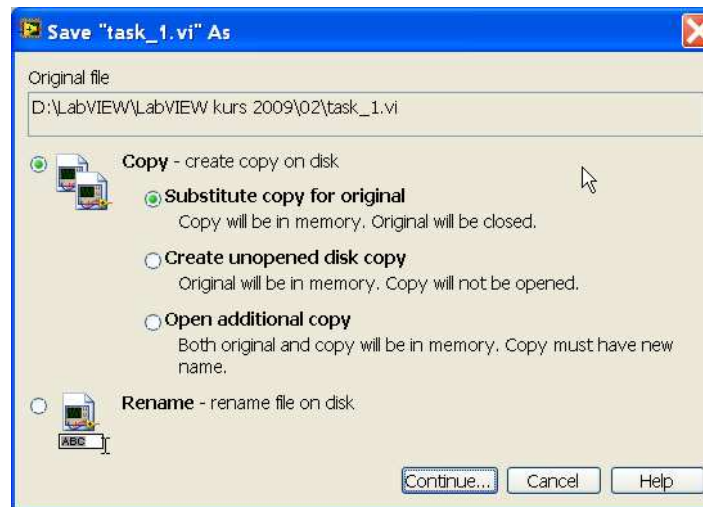


## Saving VI ...

---

**Save As...** creates a copy on disk. It has three sub-options:

- Substitute copy for original: a "traditional" Save As... operation - it renames the VI in memory and saves a copy of the VI to disk under the new name. All VIs currently in memory that call the old VI now point to the new VI.
- Create unopened disk copy: creates a copy on disk. It does not open the copy into memory.
- Open additional copy: creates a copy on disk, and then opens the copy. All VIs currently in memory that call the old VI still point to the old VI.
- Rename: renames the VI in memory and then moves (renames) the file on disk. Note that the original file is deleted.





## Saving VI ...

**Save for Previous Version...** brings up a dialog box, shown in figure below, in which you can choose to save the VI and all of its components to a previous version of LabVIEW.



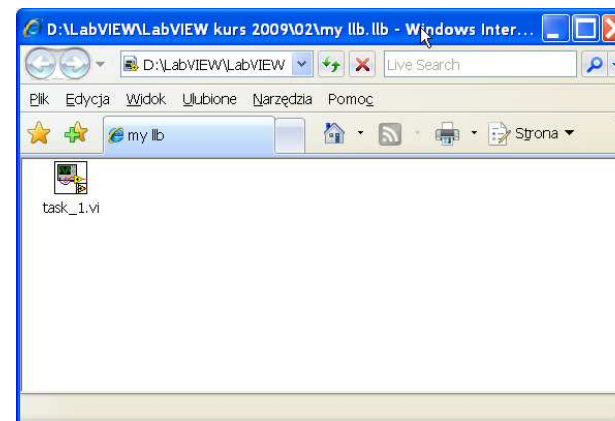
## Saving VI ...

**LLBs** are special LabVIEW files used to group several VIs together. The VIs are saved in one file with **LLB** extension, much like you would store files inside one zip file.

**LLBs** offer few advantages and many disadvantages; for example, they can contain only LabVIEW file types, not data or other files. In addition, your operating system sees **LLB** files as single files, but double clicking the file opens it in Windows Explorer or Internet Explorer, so the whole VIs are visible.

**Note:** LabVIEW now has the project library, which is a logical collection of VIs used in LabVIEW projects.

Create an LLB from the **Save** or **Save As . . .** dialog box by clicking on the **New LLB** button:

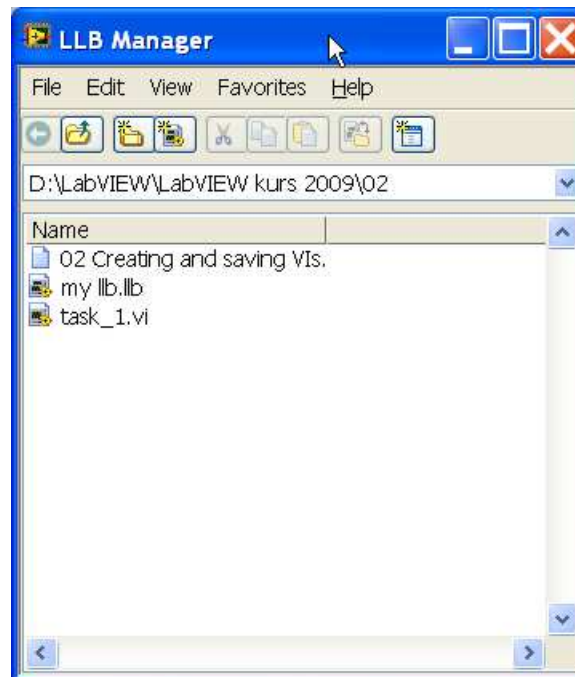


## Saving VI ...

---

### **LLB Manager**

You can use the **LLB Manager** (from the **Tools menu**) to simplify copying, renaming, and deleting files within LLBs as well as within your file system itself. You also can use this tool to create new LLBs and directories and convert LLBs to and from directories. Creating new LLBs and directories and converting LLBs to and from directories is important if you need to manage your VIs with source code control tools.

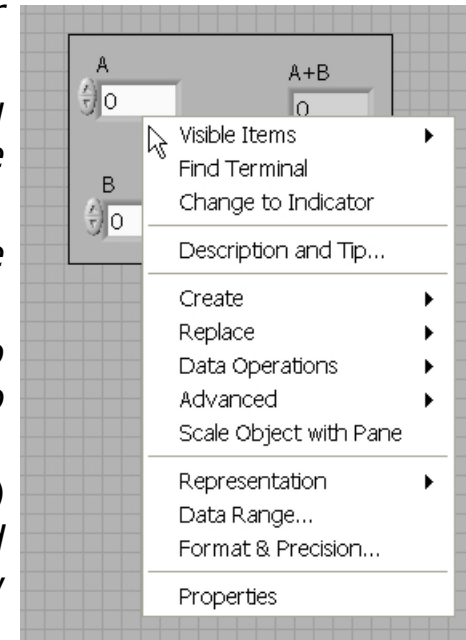


## Tricks and tips ...

---

### **For Front Panel:**

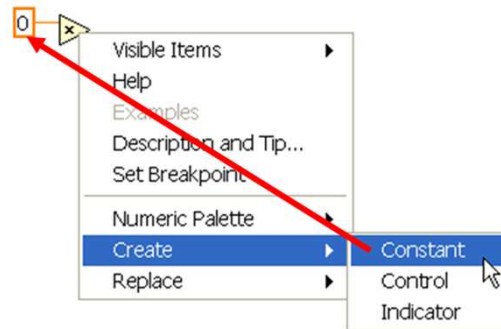
- *double-click on label of the control or indicator to edit it*
- *use **Visible Items** option in context menu (right mouse button click) to turn on/off some elements*
- *use **Replace** option in context menu to change the type of control e.g numeric → knob*
- *use **Change to ...** option in context menu to change control to indicator or indicator to control*
- *to insert more than one control (or indicator) select previous inserted one, press and hold CTRL key, use **drag-and-drop** to create new element on Front panel*
- *sometimes after saving and reopening VI you can see, that some parameters and settings of elements are changed. To omit this problem before saving use **Edit/Make Current Values Default** from main menu.*



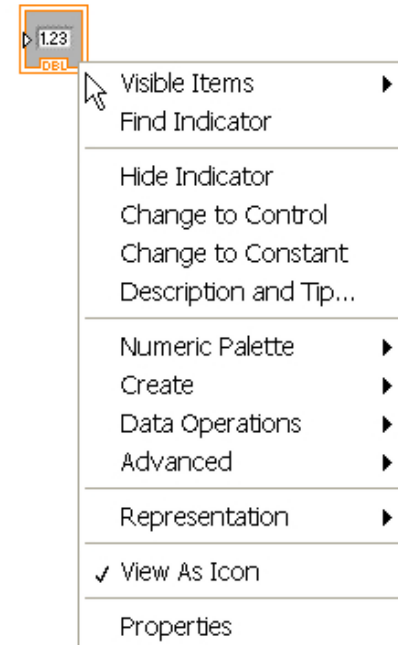
## Tricks and tips ...

### **For Block Diagram:**

- use **Change to ...** option in context menu to change control to indicator or constant, indicator to control or constant ...
- use **Create** option for automatically creation of proper element, e.g. for indicator you can create control or constant. It is possible to create elements for functions too.



- to insert more elements (e.g. add function) select that element, press and hold CTRL key, use **drag-and-drop** to multiply copy.



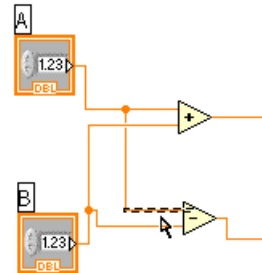
## Tricks and tips ...

---

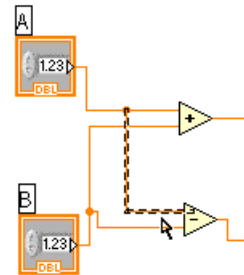
### **For Block Diagram:**

If it necessary to delete some wires:

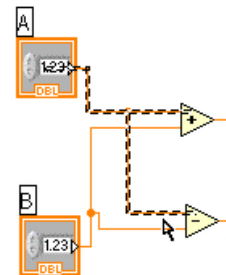
- click once on the wire,  
then DELETE key:



- click two times on the wire,  
then DELETE key:



- click tree times the wire, then DELETE key:



## Exercise ...

---

*Build temperature converter.*

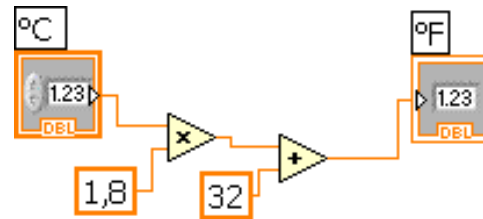
$$^{\circ}\mathbf{F} = ^{\circ}\mathbf{C} \cdot \mathbf{1,8} + \mathbf{32}$$

*Instructions: Build a VI that converts °C to °F. When run, the VI should take an input value (°C), multiply it by 1.8, add 32, and display the result (°F). The front panel should display both the input value and the result.*

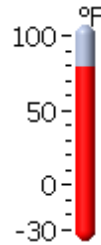
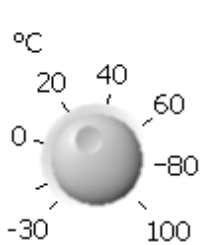
*Save the VI as Convert C to F.vi.*

## Exercise ...

Solution ...

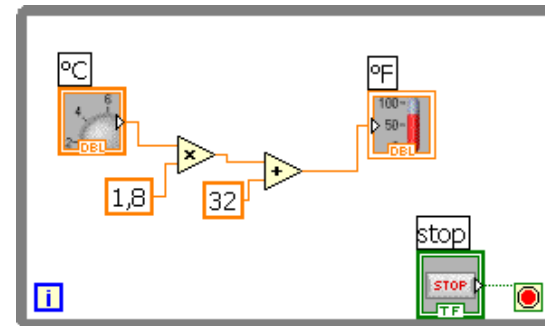


- *change the control and the indicator to the following form:*



*change the range of scale for knob and thermometer by editing first and last label on the scale*

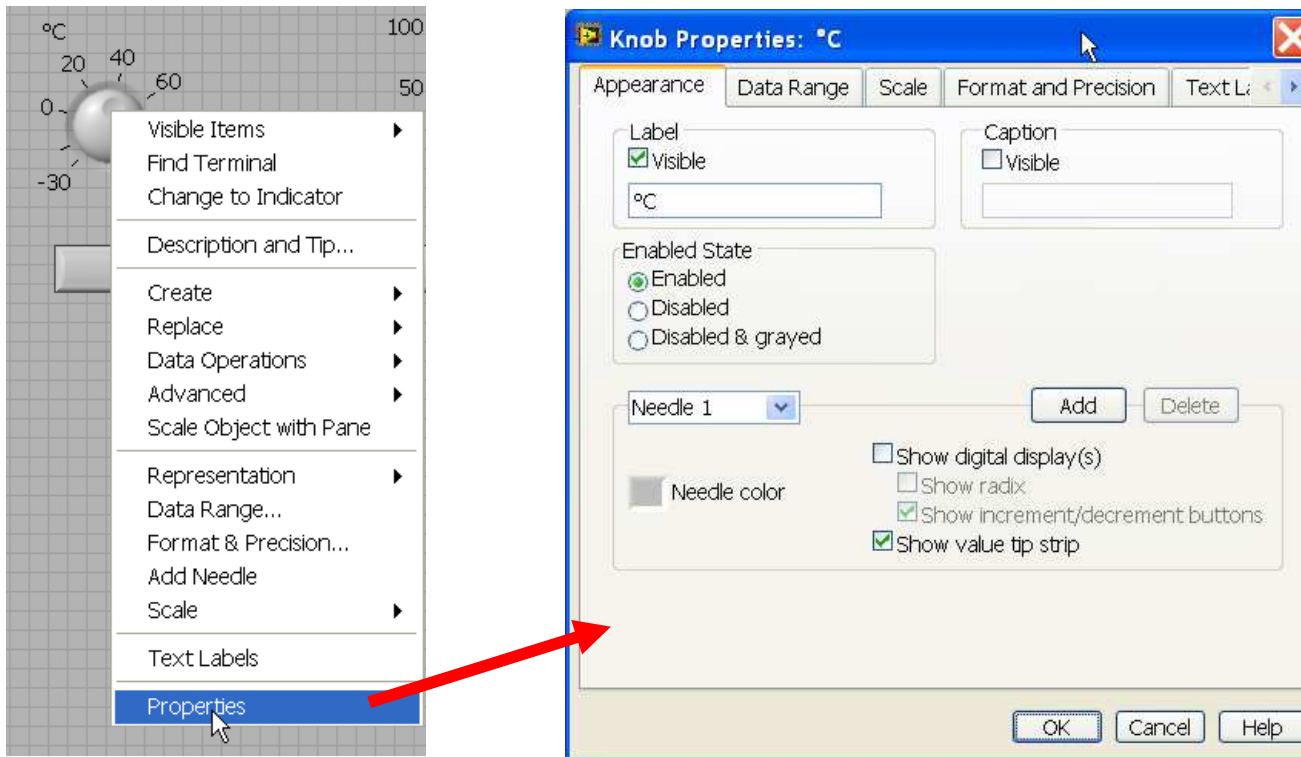
- *use **While loop** for continuous execution of the application.*





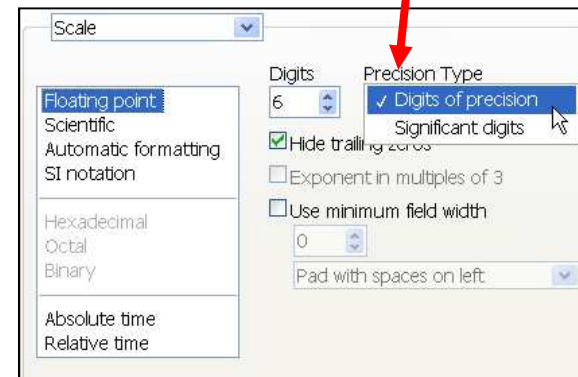
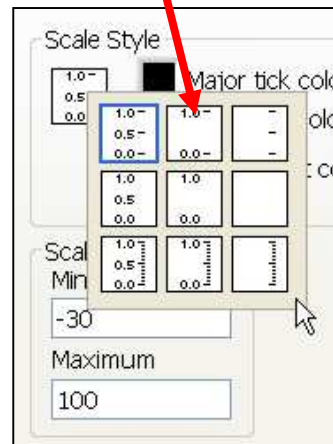
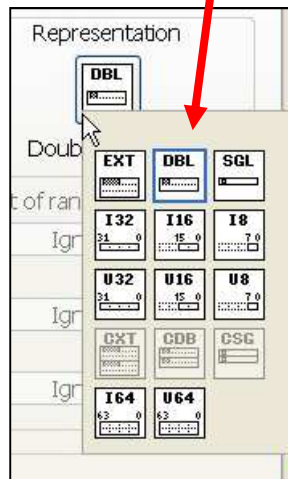
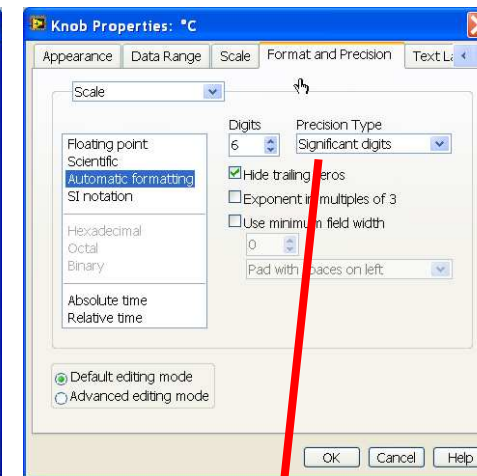
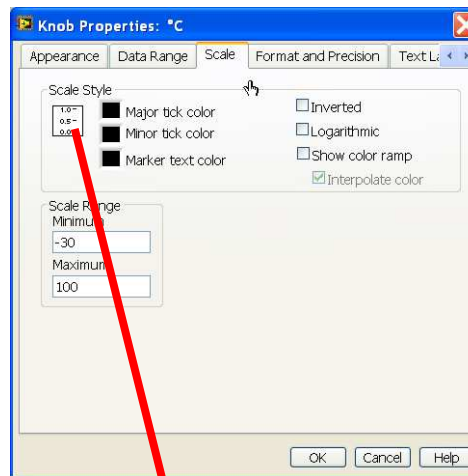
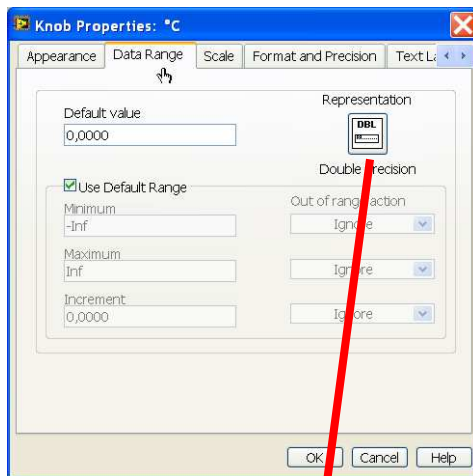
## Customizing controls and indicators

User can define a lot of parameters for each elements on Front Panel or Block Diagram. It is possible with the use of context menu and **Properties** option. It opens additional window with some numbers of tabs. Their context depend on type of selected element.



# Customizing controls and indicators

Other tabs...

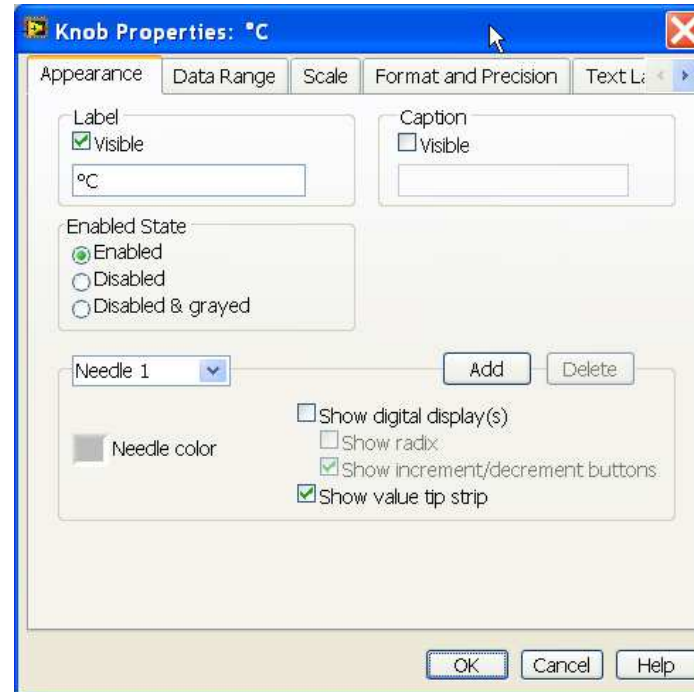
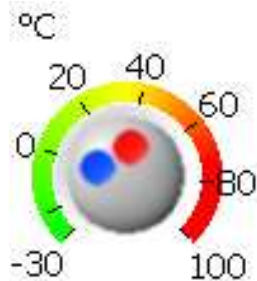


## Customizing controls and indicators

**Note, that some changes can involve an error, which not exist for previous settings.**

- Change **Needle color** then press **Add** button to create second needle. You can change their color too.
- Switch to **Scale** tab and mark **Show color ramp** checkbox.

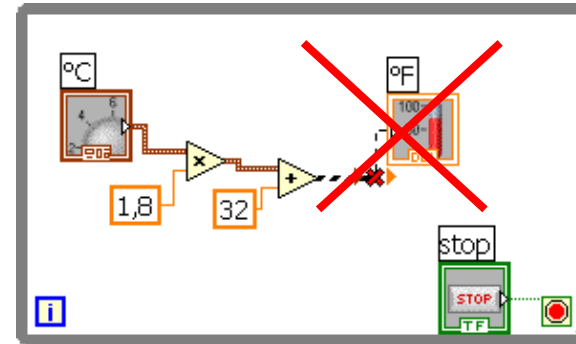
The knob should look like below.



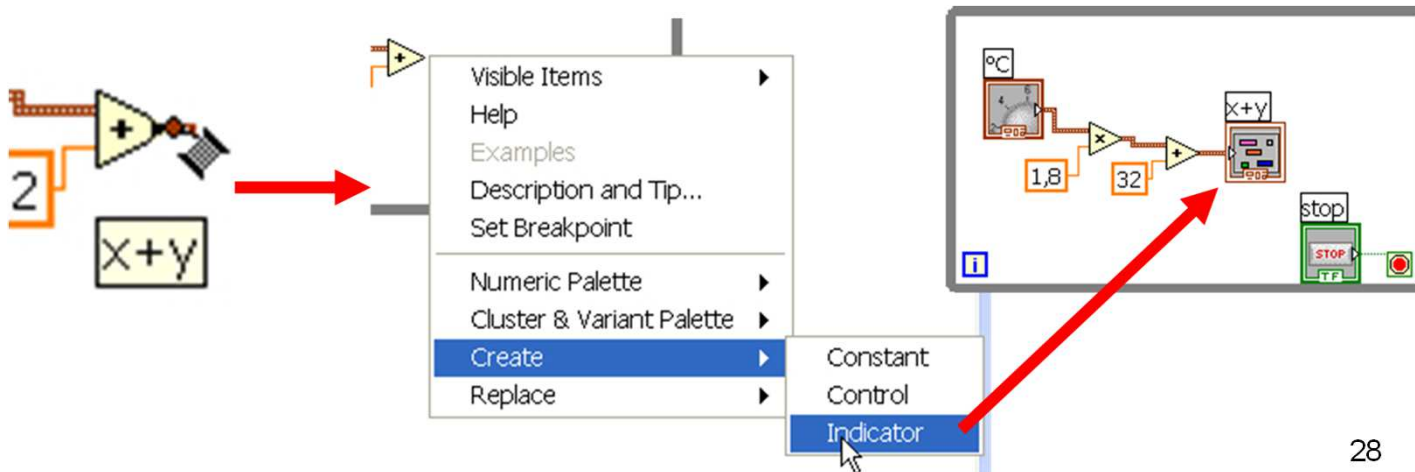
Note, that **Run arrow** on toolbar was change to **Broken arrow**.

## Customizing controls and indicators

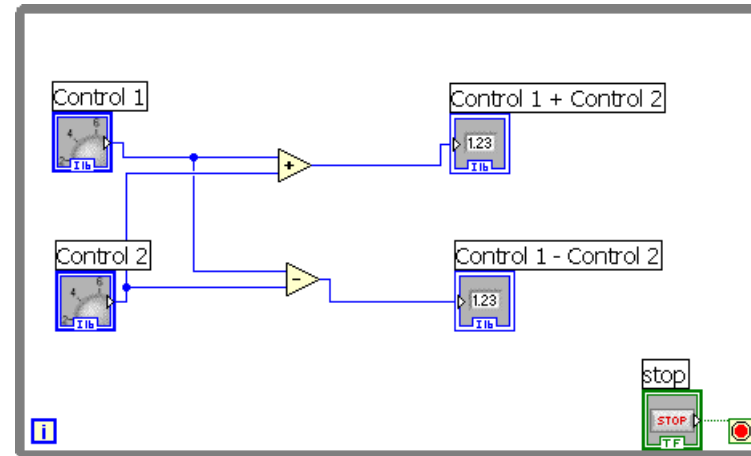
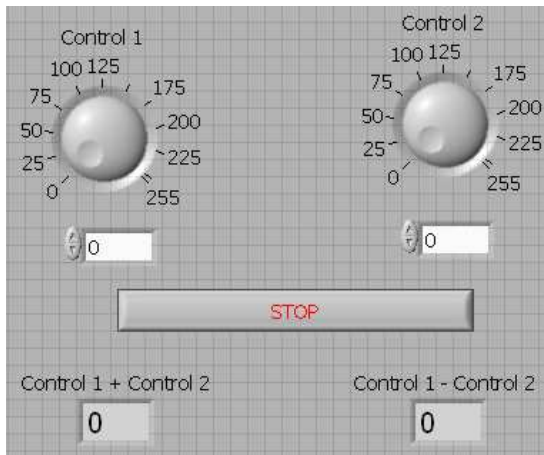
Open Block Diagram to fix error. The data type for knob control is different, and the indicator does not correspond to it. The bad wire is displayed.



Select bad wire and indicator and delete them. Place wiring tool on output terminal of the **Add function** and open Contextual menu. Select **Create** → **Indicator** option. The proper indicator will be generated automatically.



## Build application ...



- **By the way...**
- **Note!** For fast copying you can select the object and pressing Ctrl button drag&drop selected object.
- For Controls 1 and 2 as well as the sum and the difference set the representation: Word (I16).
- For Controls 1 and 2 set the range from 0 to 255. Run the application.
- Next change the difference indicator to Byte (I8) and again run the application.

**Mismatched data type can cause errors !!!**