EE-206

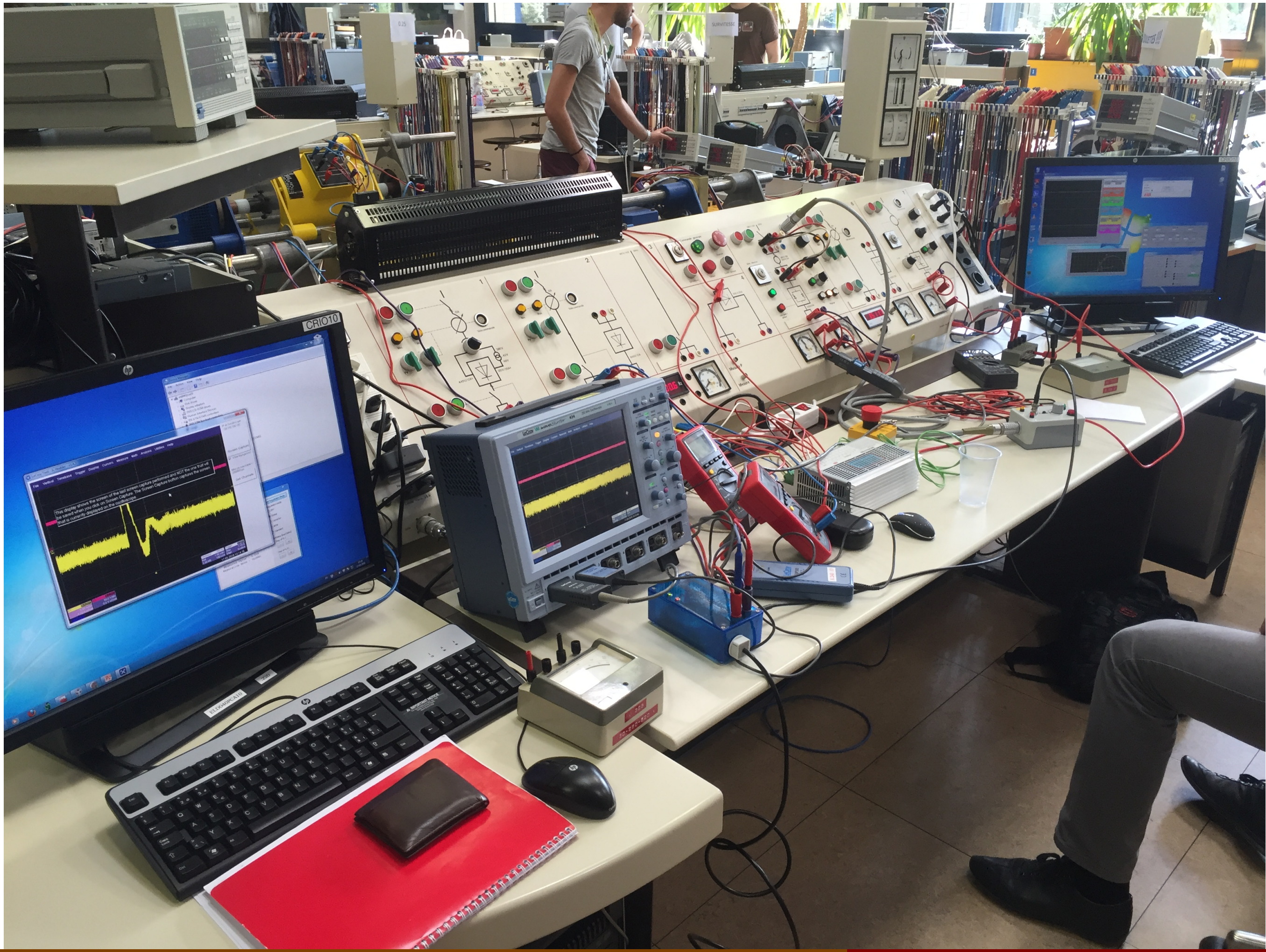
Systèmes de mesure

LabVIEW crash course

- LabVIEW: what & why?
- Getting started
- Controls vs indicators

LabVIEW crash course

- LabVIEW: what & why?



LabVIEW Trivia

- What does LabVIEW stand for?

Laboratory **V**irtual **I**nstrument **E**ngineering **W**orkbench.

- What is LabVIEW?

LabVIEW is a **graphical programming language**, typically used for data acquisition, instrument control, and industrial automation.

Introduced in 1986, it is supported by several OS (mostly Windows, Unix and Linux) and it can be installed on PCs as well as industrial controllers.

Graphical Program

MATLAB, HTML, Java are all traditional textual languages: the code consists of a sequence of operations/instructions.

LabVIEW is a graphical (G) programming language:

- no text but a sort of block diagram
- each block correspond to an operation/instrument
- each block has different options/configurations
- the connection defines the order of the different stages and the flow of the data

Advantages

LabVIEW yields also signal processing functionalities, but it is mostly used for:

1. remote instrument control
2. measurement data acquisition
3. automatic control routines

NB: a program in LabVIEW is called **V**irtual **I**nstrument (**VI**) as it behaves as an instrument with controls and outputs (either numeric results or graphs).

Examples

- Plug & play acquisition board

➔ **c**ompact **D**ata **A**c**Q**uisition platform (cDAQ)

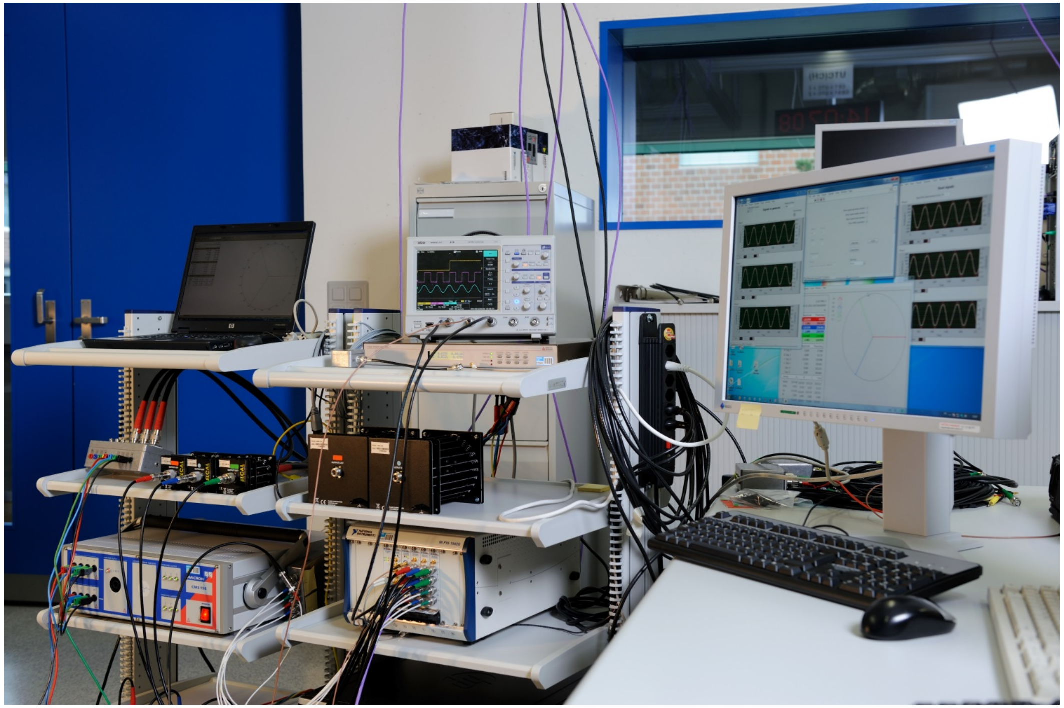
- extension of the PC
- pluggable modules
- data acquisition
- data transfer
- programmable trigger
- control outputs

Examples

- Coordination and interface between different instruments

➔ **P**CI e**X**tensions for **I**nstrumentation (**PXI**)

- waveform generator
- voltage amplifier
- current amplifier
- shunt & dividers
- GPS synchronization
- data acquisition
- graph and result logs



Examples

- Real-time stand-alone measurement unit

➔ **c**ompact **R**econfigurable **IO** modules (**cRIO**)

- real-time controller
- network interface
- pluggable modules
- FPGA board for

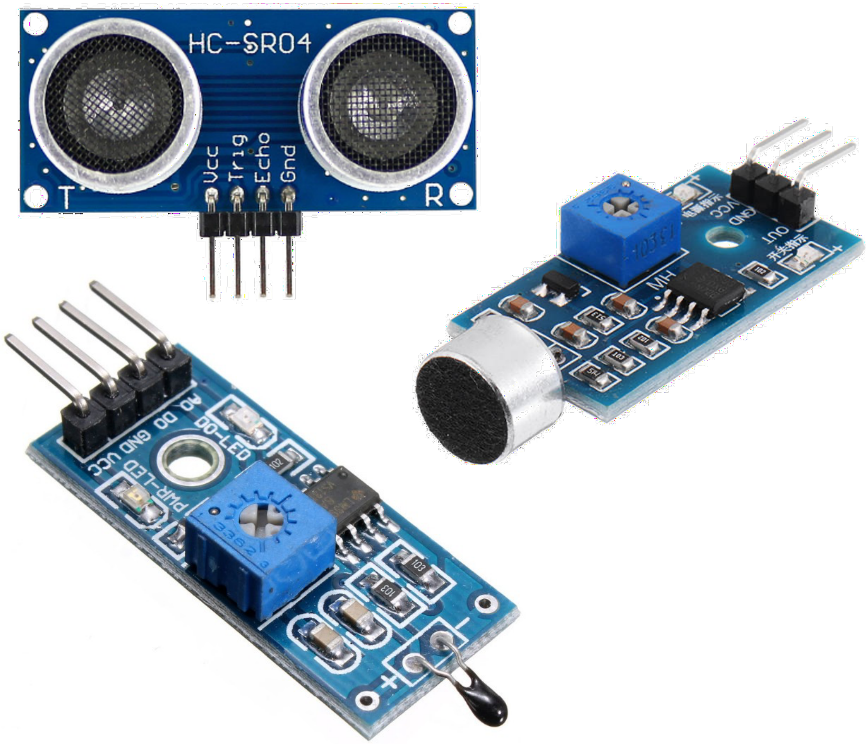
(fast, deterministic)

Examples

- Interface with third-party sensors (Arduino style)

➔ e.g. ultrasound, thermistor, microphone

- integrated libraries
- dedicated functions
- app examples
- self-test routines
- automatic scaling



Getting started

Let's launch Labview..

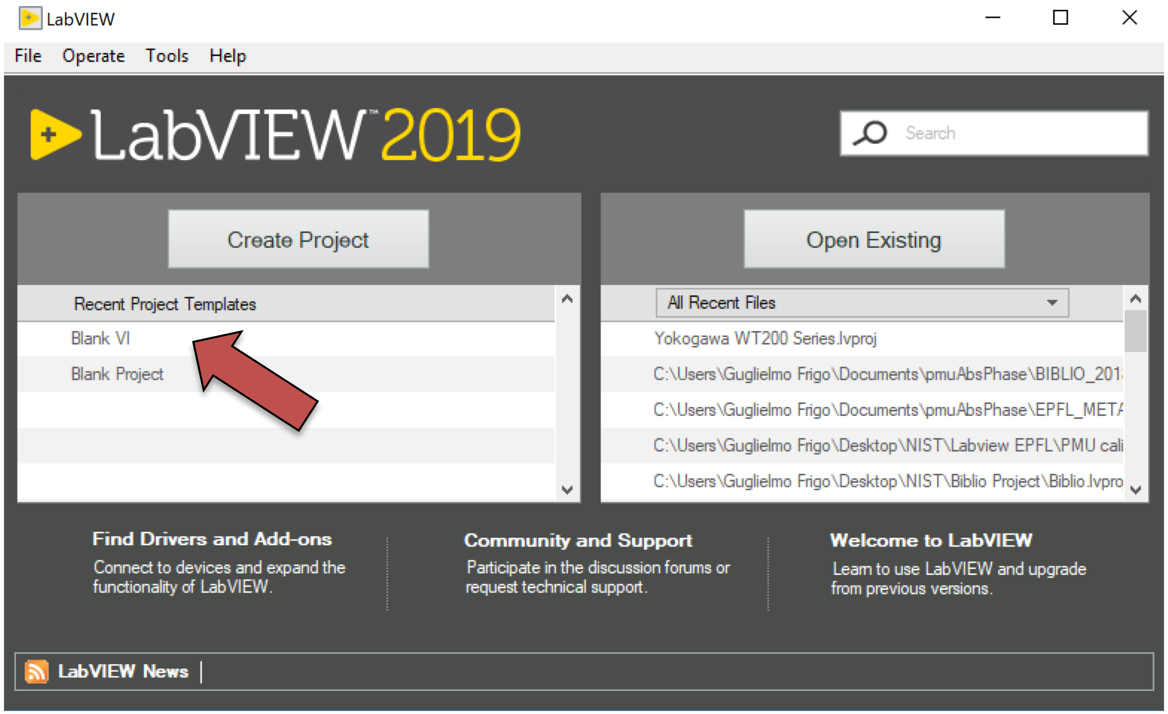
Create a file:

- VI → single file (.vi)
- Project → folder

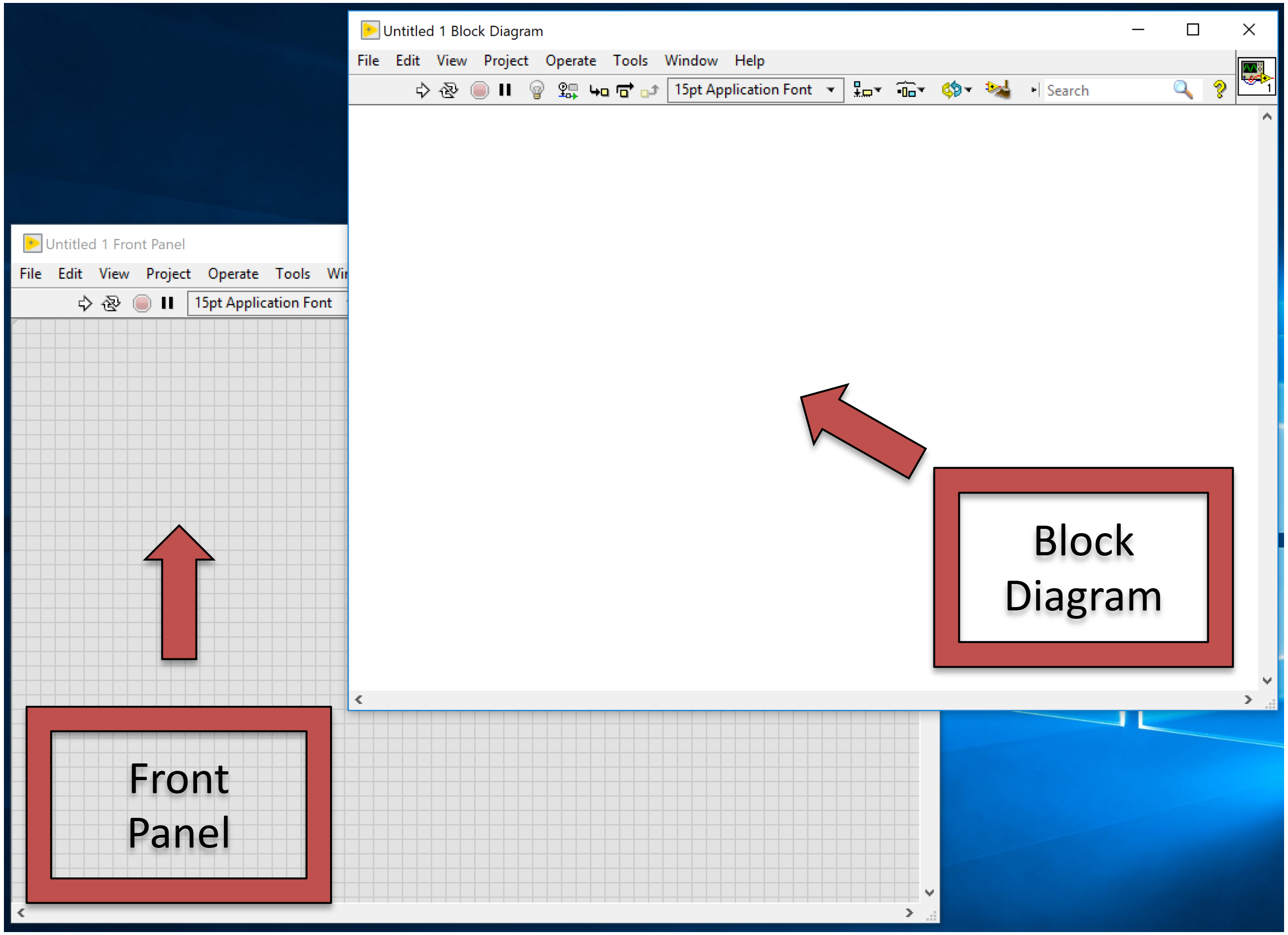
with multiple files (.lvproj)

Open a file:

- Recent files' history



Blank VI


Untitled 1 Block Diagram
File Edit View Project Operate Tools Window Help
15pt Application Font
Search
Untitled 1 Front Panel
File Edit View Project Operate Tools Wi
15pt Application Font
Block Diagram
Front Panel

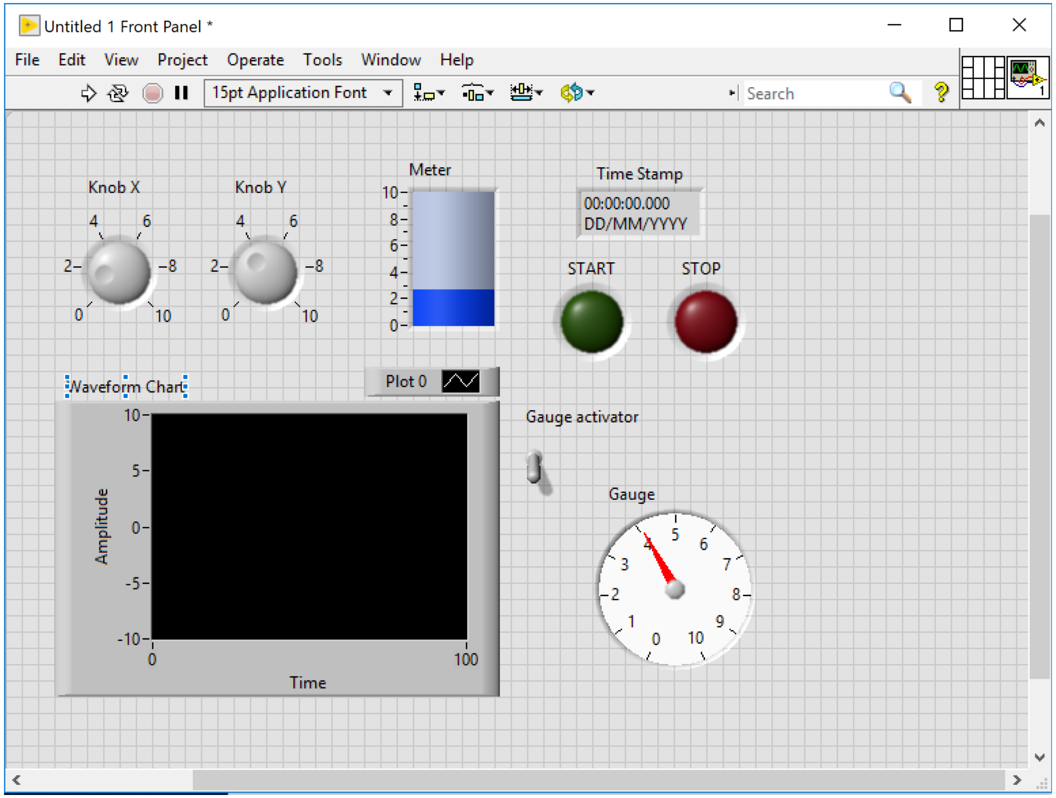
Front Panel

The Front Panel is the user interface of your VI

Once populated with:

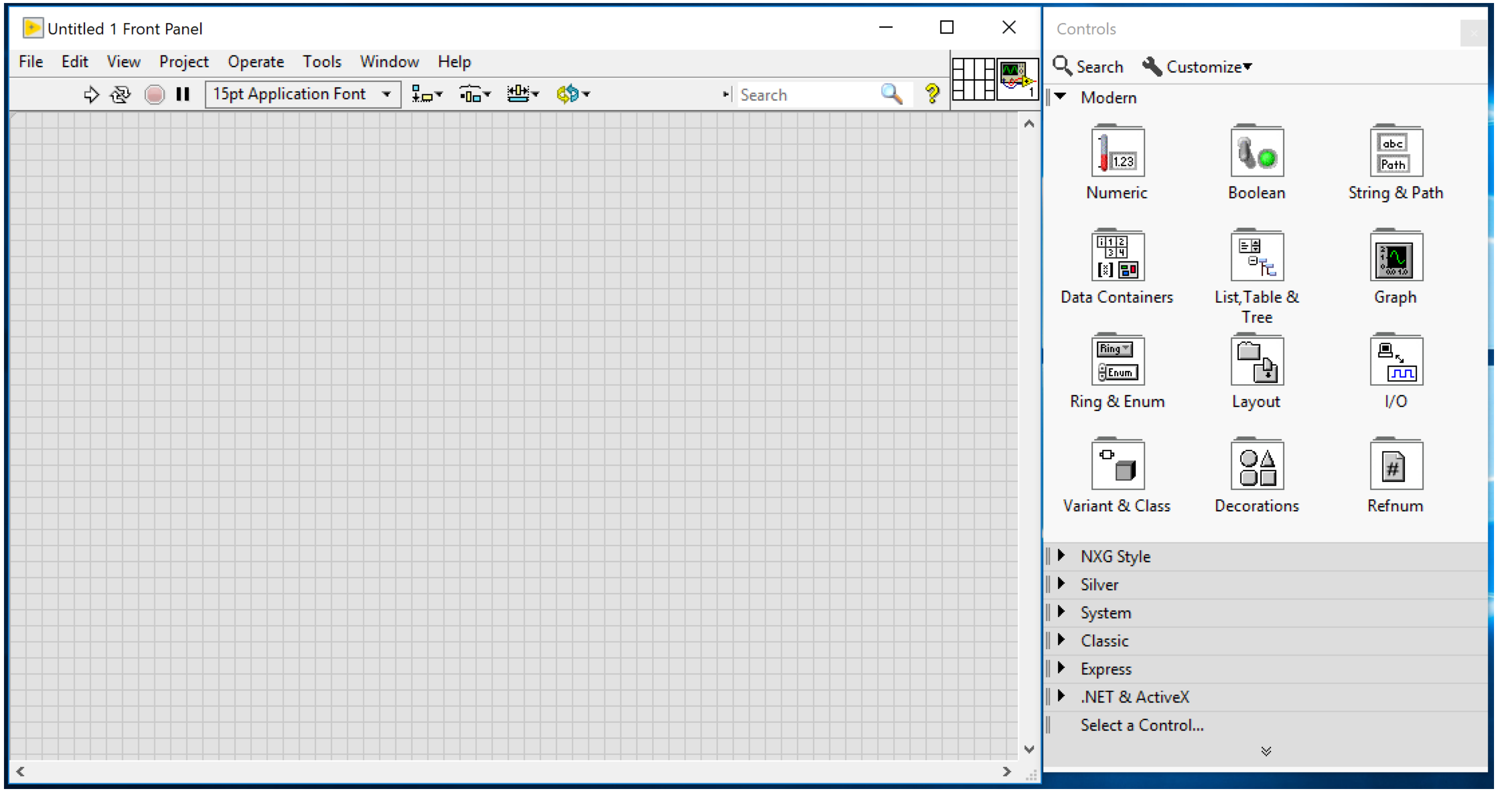
- controls
- indicators
- graphs

it will appear as the front panel of a real instrument

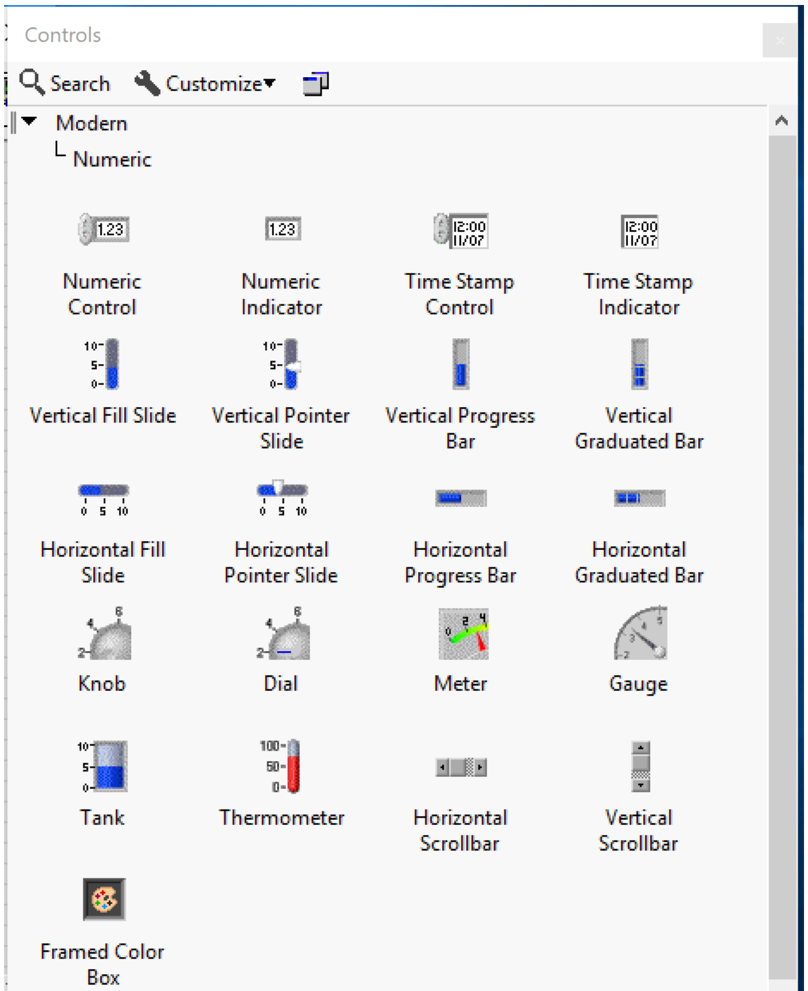


Controls Palette

In order to introduce new controls and switches we can use View → Controls Palette

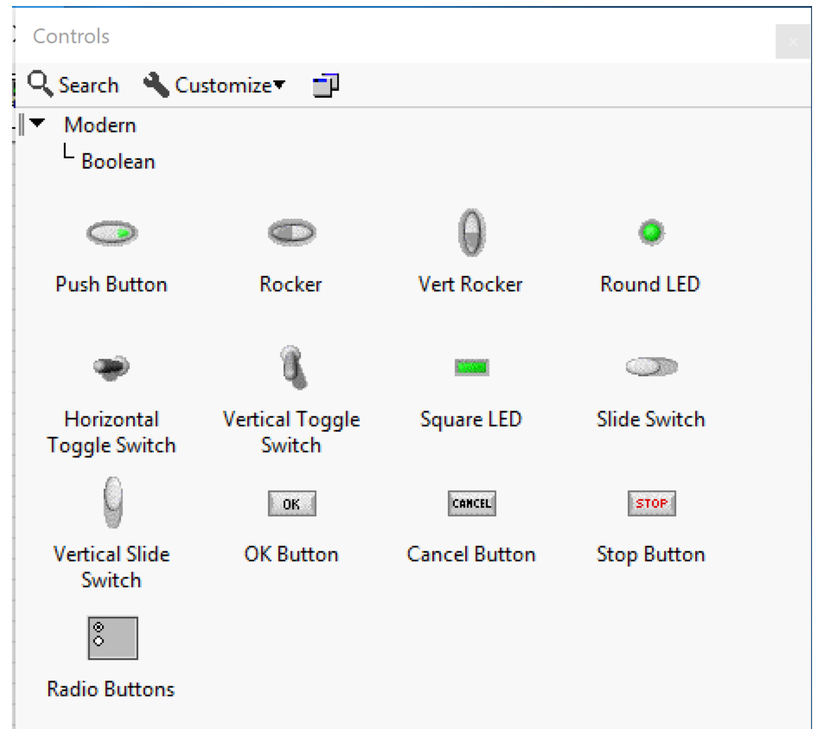


Controls types



Numeric

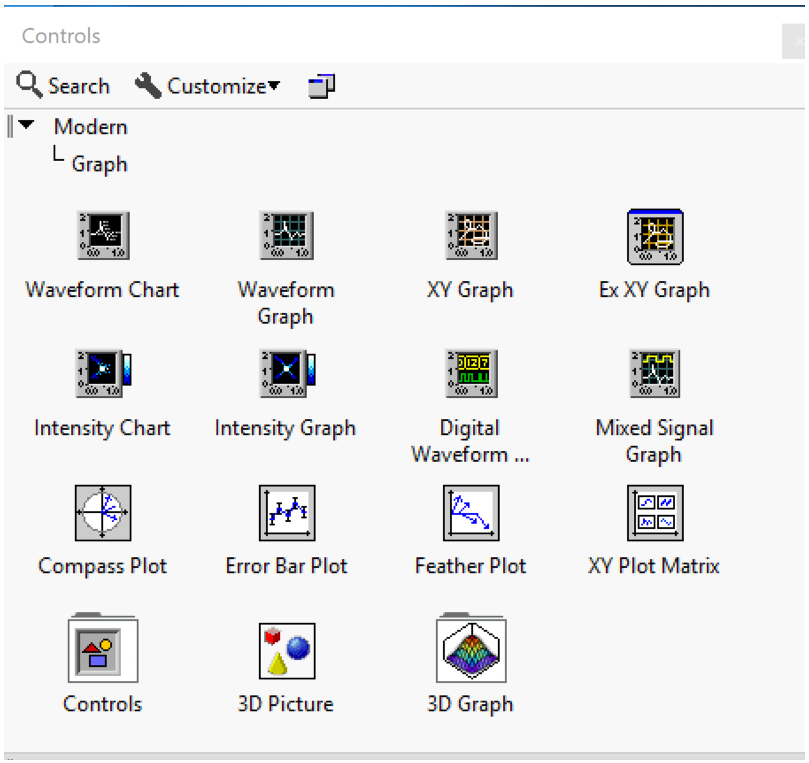
Boolean

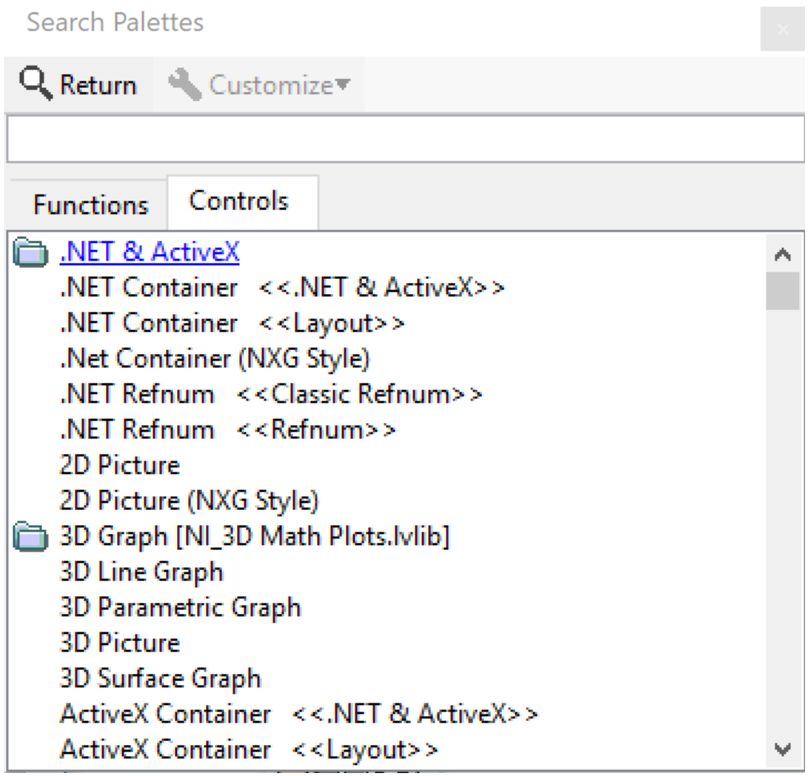


Controls types

Graph

Search into the library

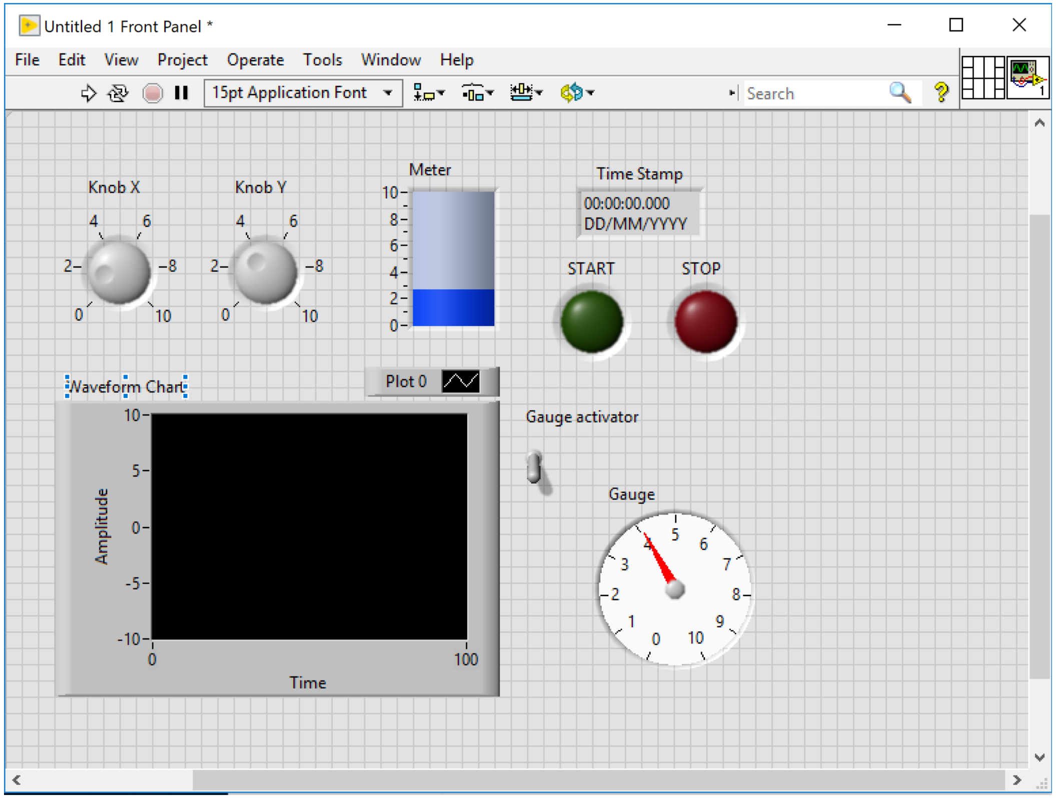




Example

Just by dragging and dropping on the front panel, we get…

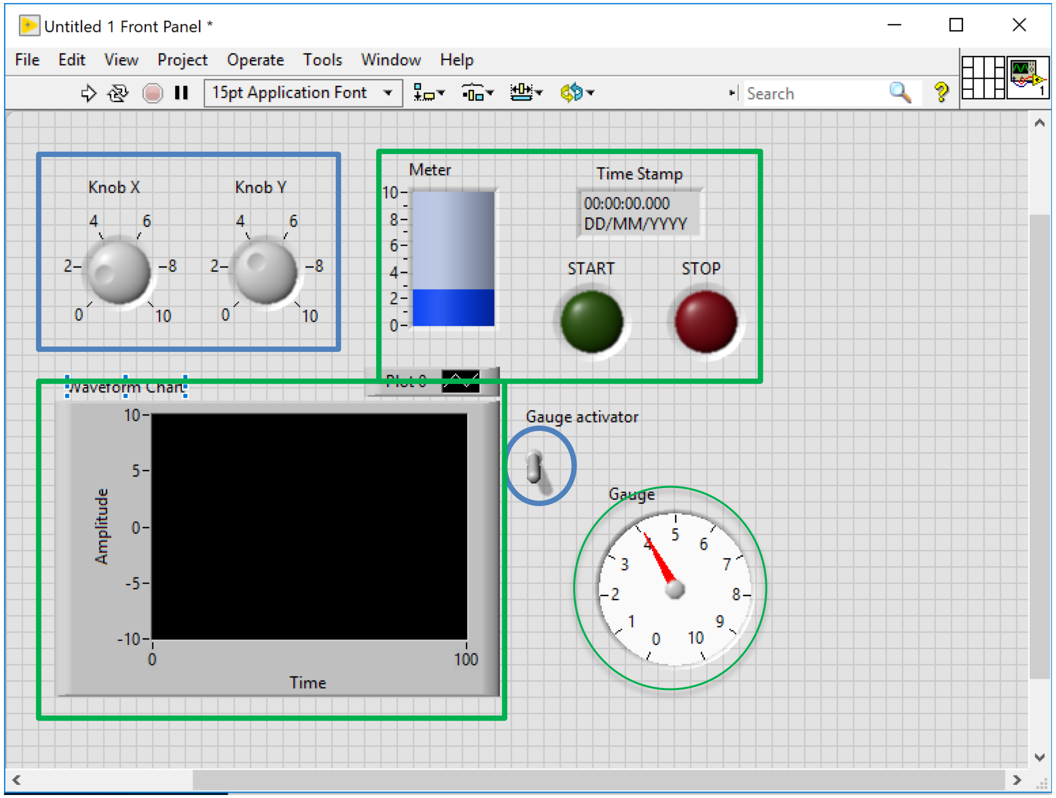
- manual controls
- digital controls
- switches/leds
- gauge/tank meter



Controls vs Indicators

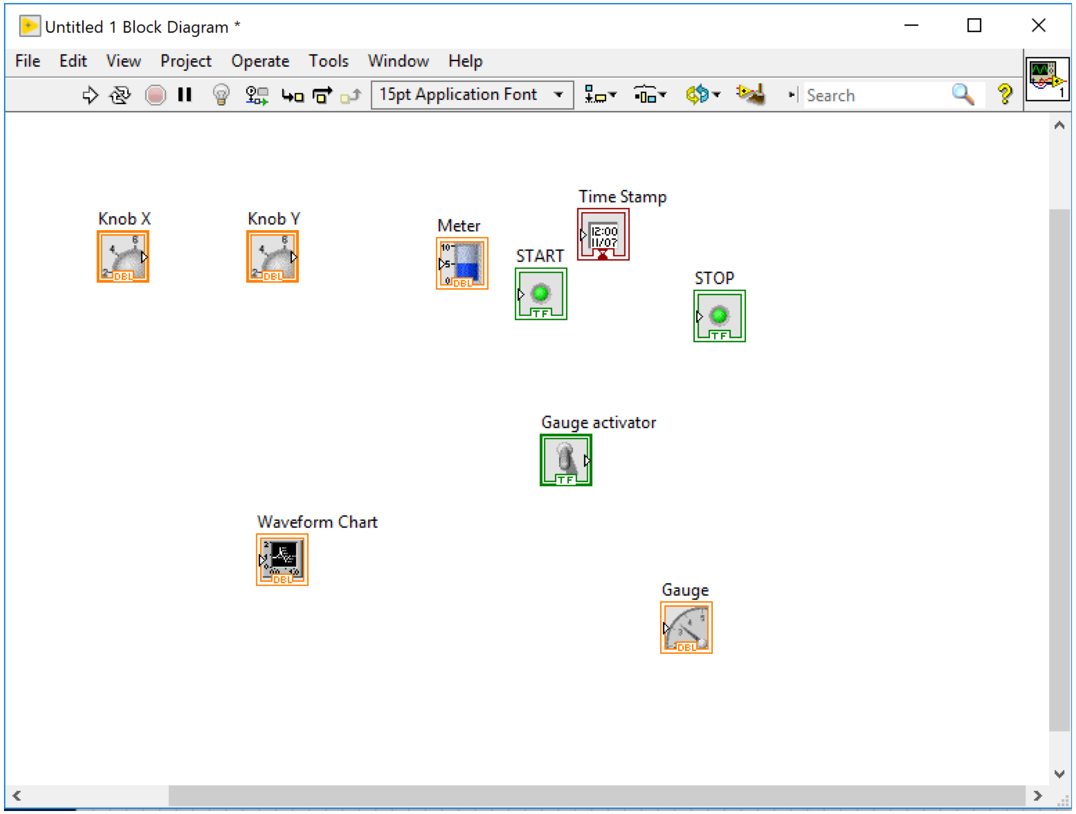
The same difference between function inputs and outputs:

- **Controls (inputs)**
  the user can set specific parameters or trigger specific operations
- **Indicators (output)**
  the user can visualize the results or the process state through numbers or graphical tools



Block Diagram

Every time you drag something on the front panel you have its counterpart on the block diagram, where you can:



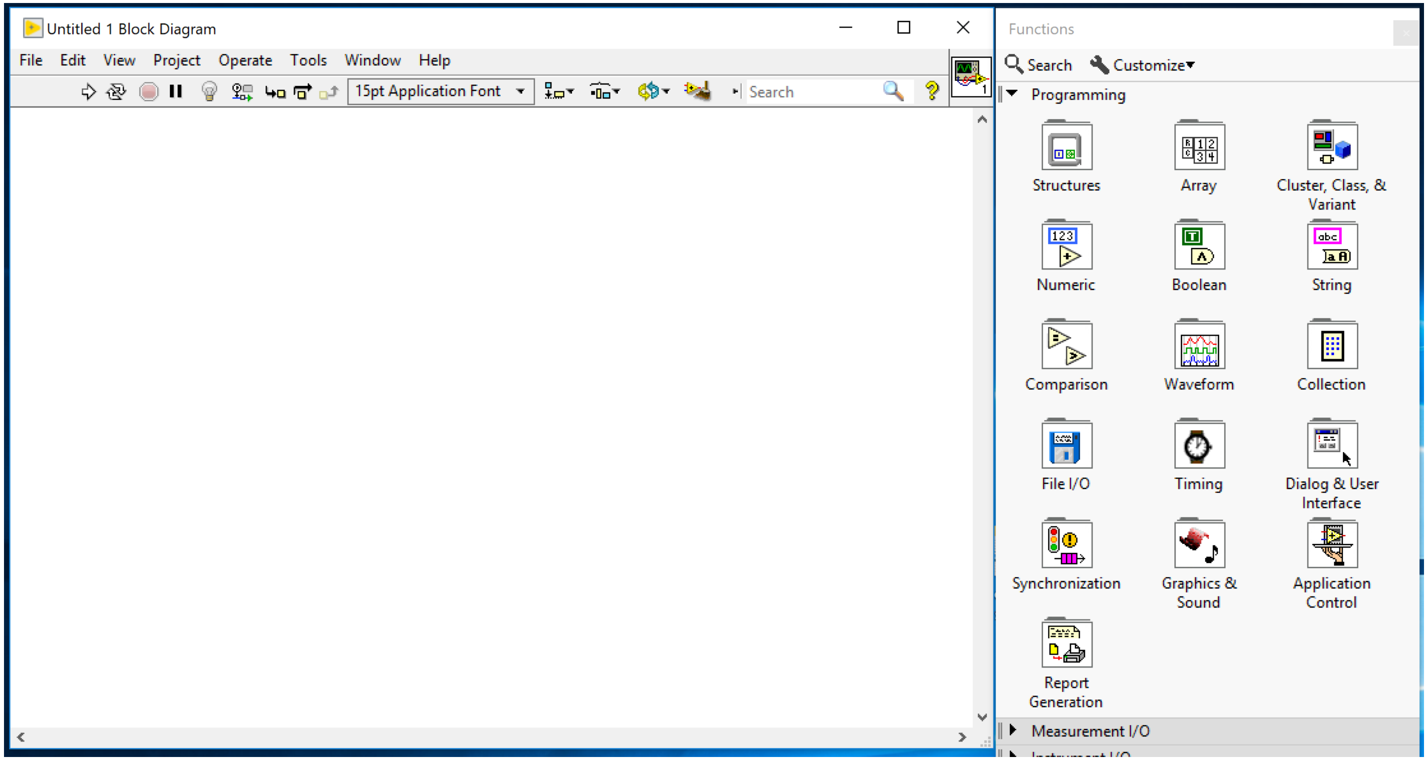
- connect them
- move them

NB#1: NO ZOOM
NB#2: if you delete here it will disappear also in the front panel!

Functions Palette

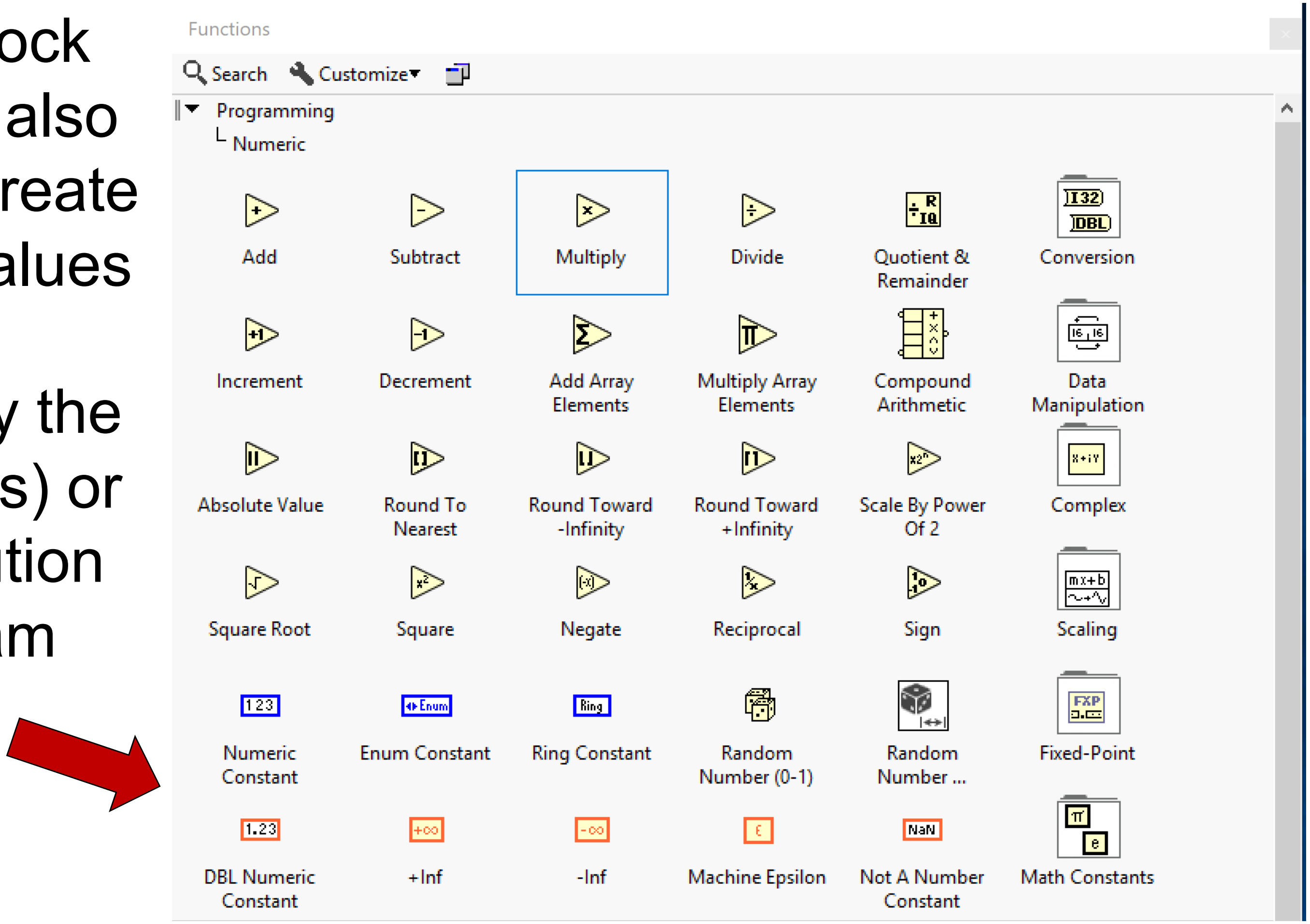
Controls and indicators can be connected in several ways

View → Functions Palette



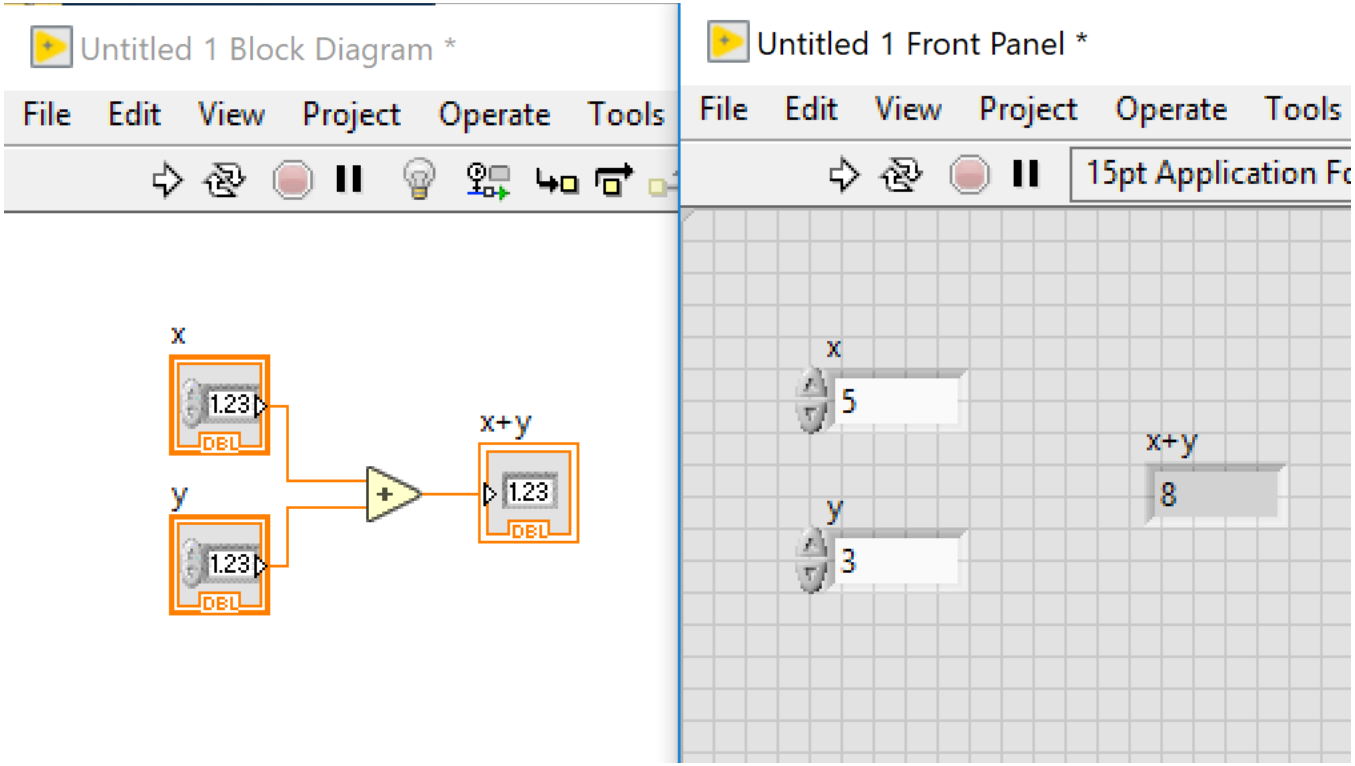
Numeric

NB: In the block diagram it is also possible to create constants: values that are not modifiable by the user (controls) or by the execution of the program (indicators).

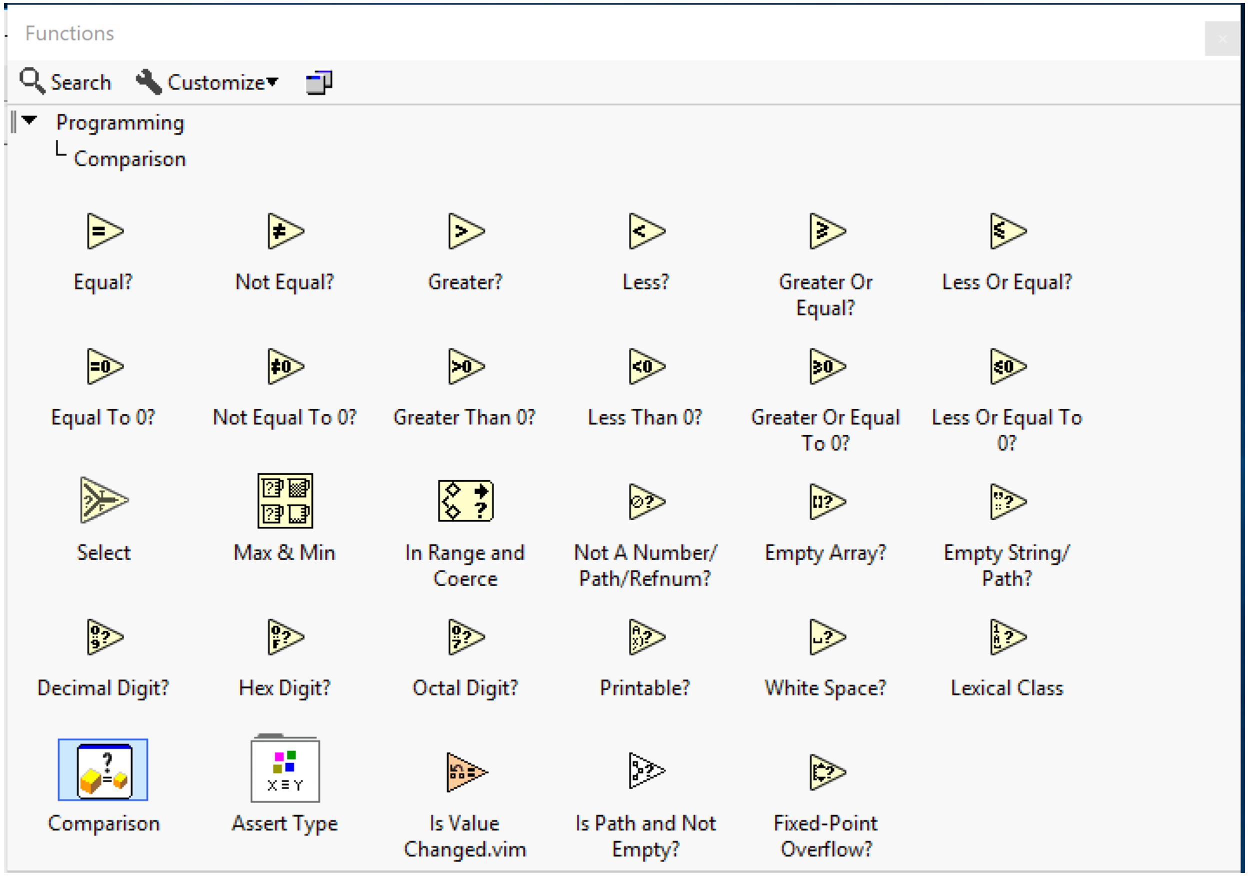


Numeric example

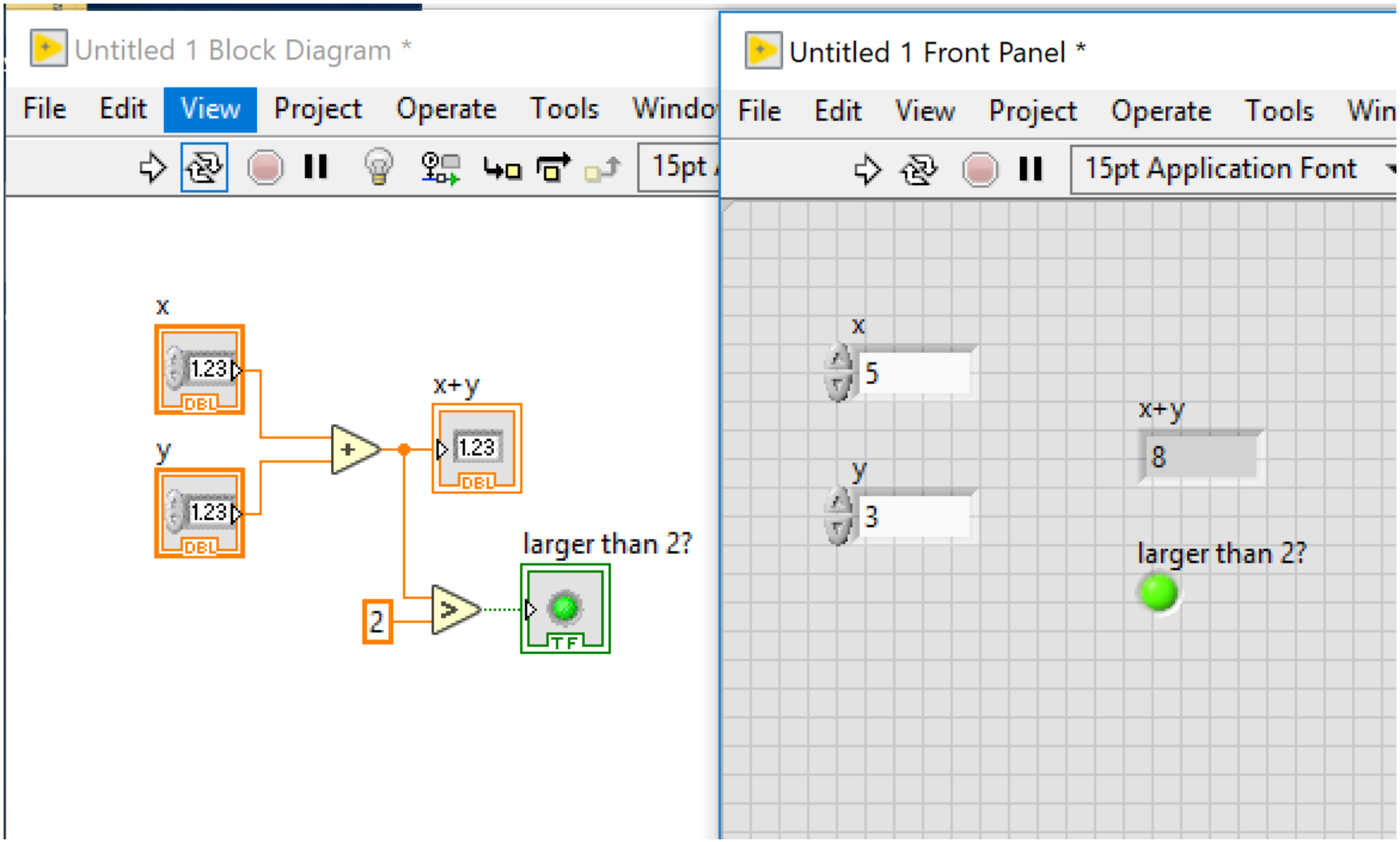
- The sum of two numbers requires two control, a function and one indicator



Comparison


Functions
Search
Customize
Programming
Comparison
Equal?
Not Equal?
Greater?
Less?
Greater Or Equal?
Less Or Equal?
Equal To 0?
Not Equal To 0?
Greater Than 0?
Less Than 0?
Greater Or Equal To 0?
Less Or Equal To 0?
Select
Max & Min
In Range and Coerce
Not A Number/ Path/Refnum?
Empty Array?
Empty String/ Path?
Decimal Digit?
Hex Digit?
Octal Digit?
Printable?
White Space?
Lexical Class
Comparison
Assert Type
Is Value Changed.vim
Is Path and Not Empty?
Fixed-Point Overflow?

Comparison example



Tools Palette

Since LabVIEW is a graphical programming language it is important to use the mouse and the cursor..

View → Tools Palette

 Automatic selection

 Operating

 Positioning

 Labeling

 Wiring

 Shortcut

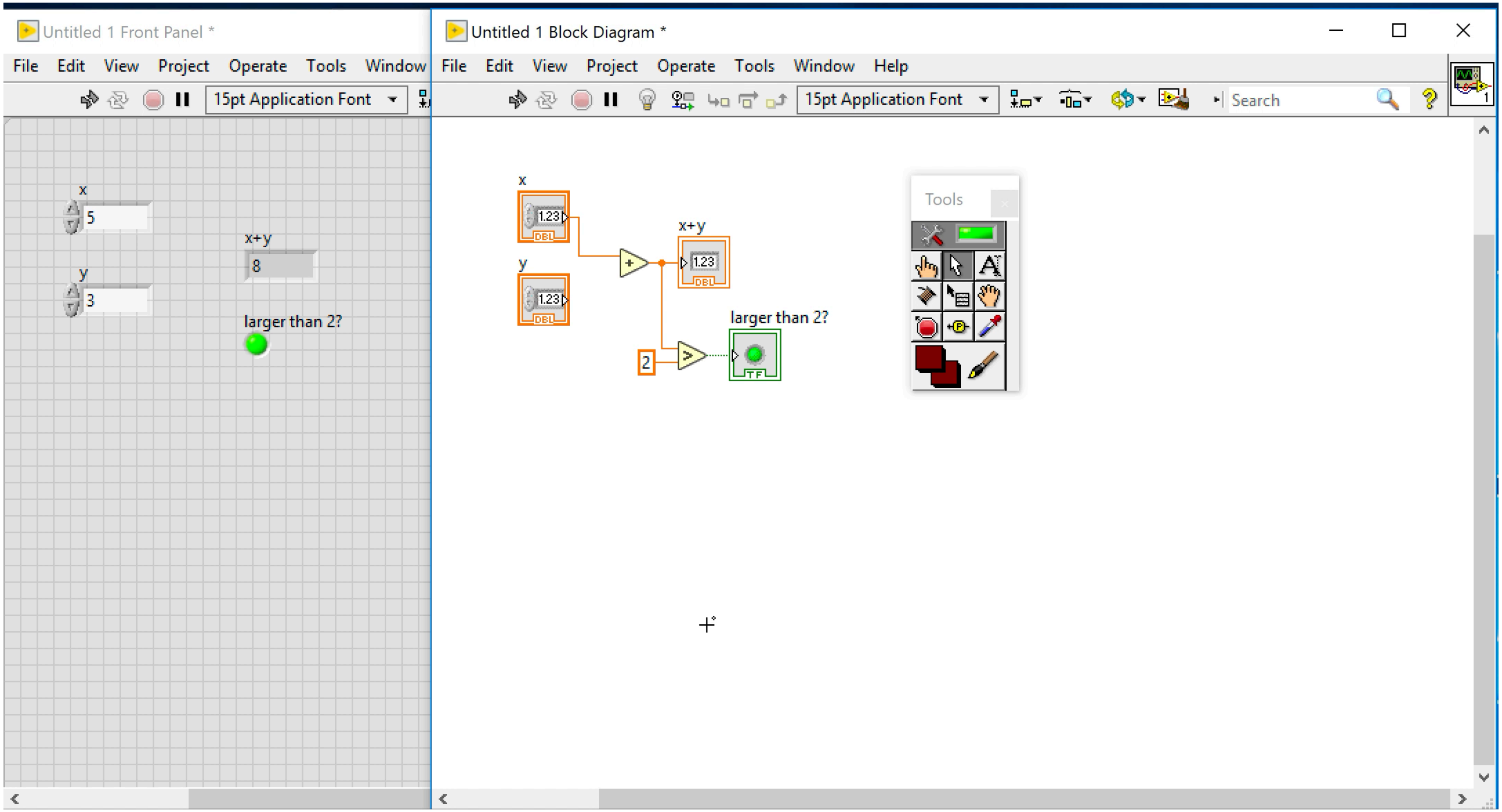
 Scrolling

 Setting a breakpoint

 Probing

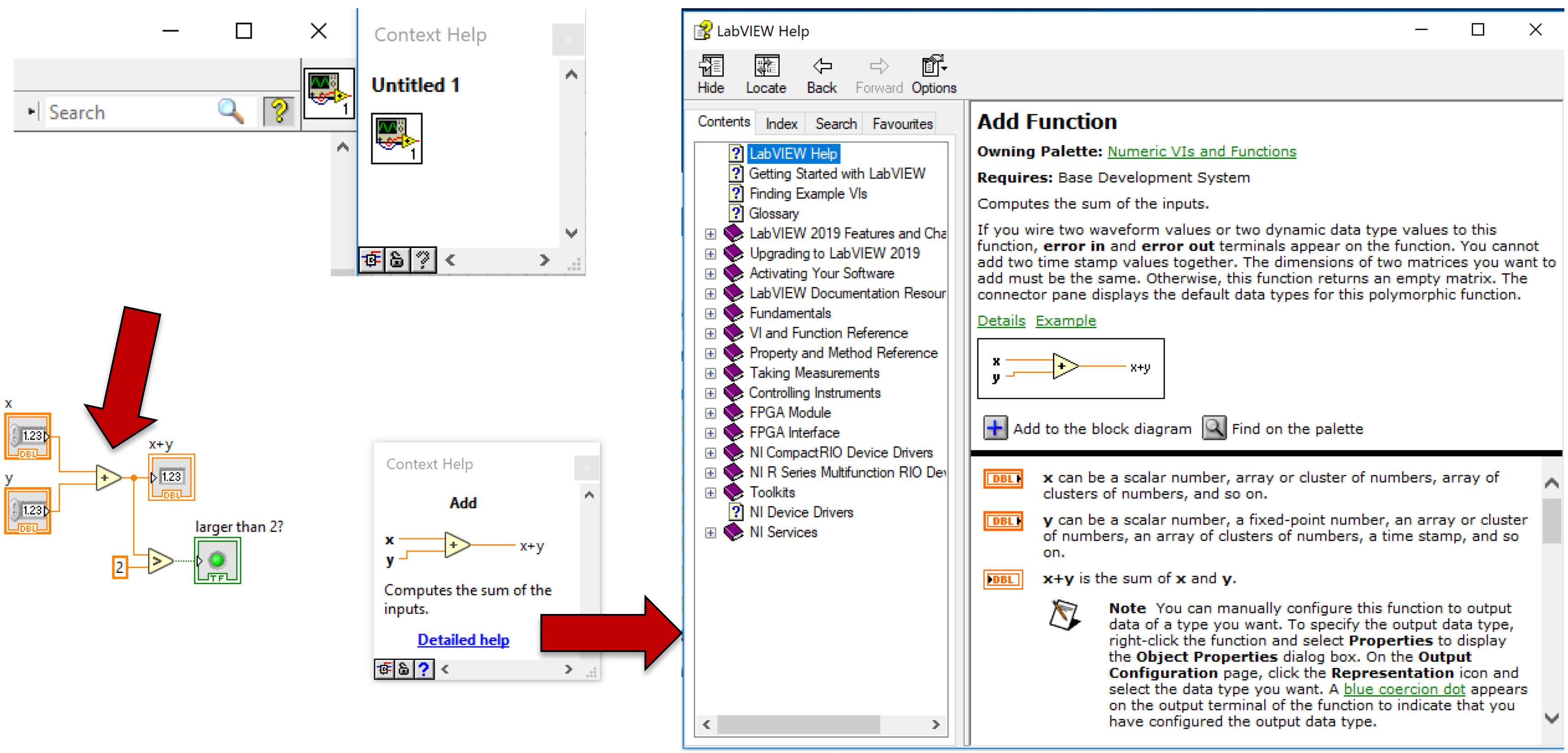
 Picking a color

Tools example



Help

If you don't know what something is, you just ask for help…



Execution modes

When the code is ready, we can run it!

 Run the code once

 Run the code iteratively

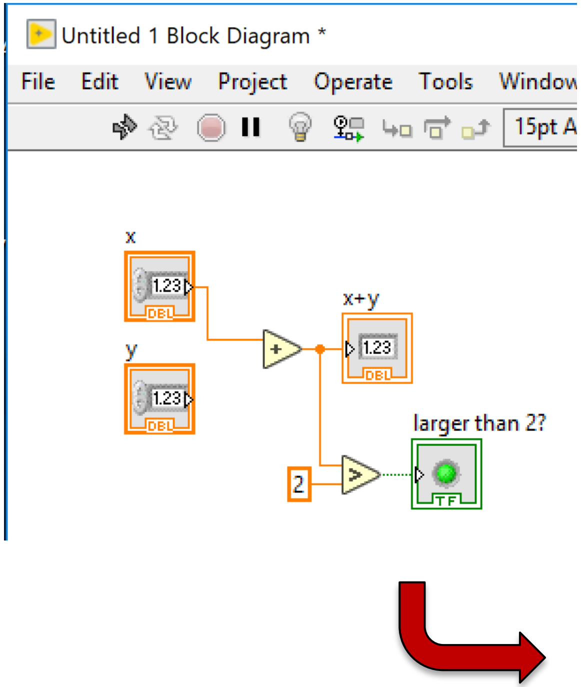
 Abort the execution

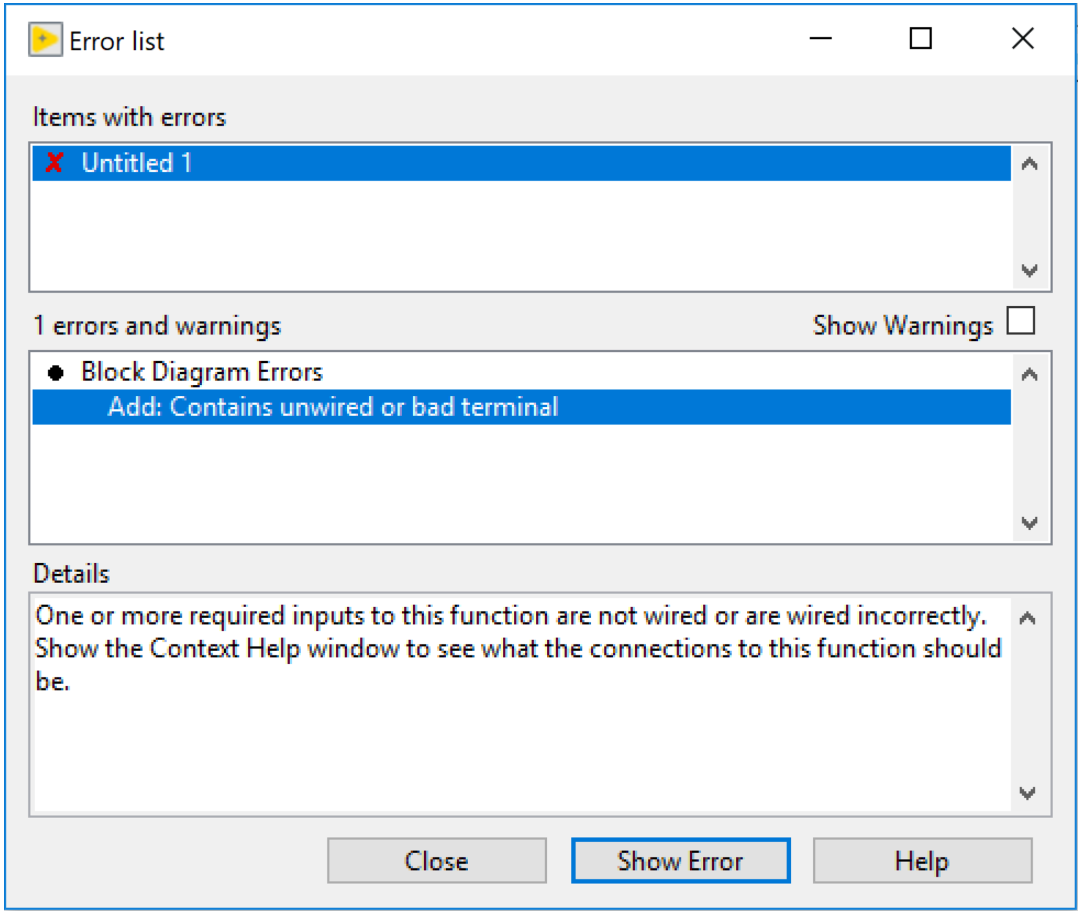
 Pause the execution

 Run in DEBUG mode

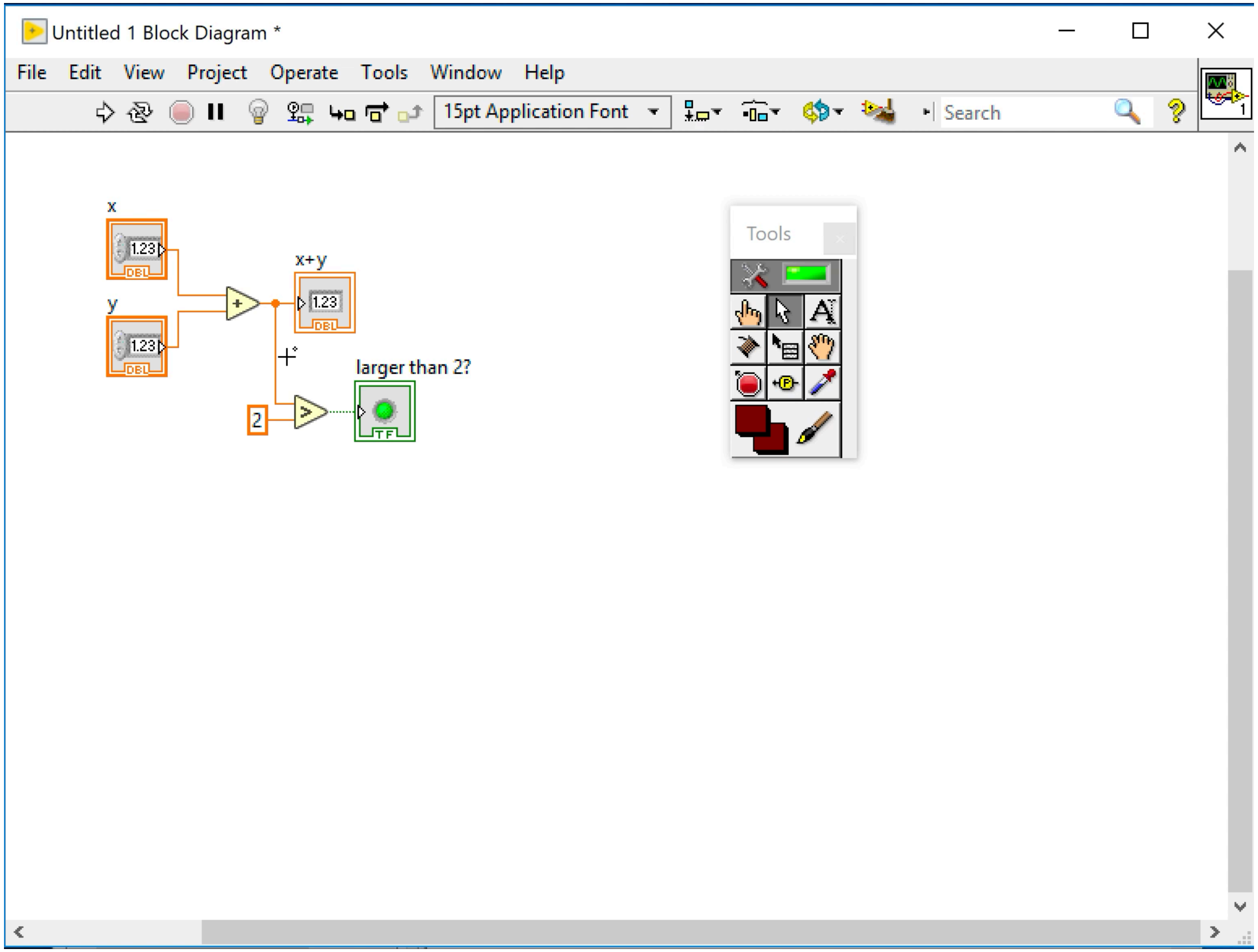
 If there is something wrong in the code, e.g. a block disconnected, the arrow is broken and the code cannot be run.

Broken arrow





Debug example



Recap: was everything clear?

- What's the difference between block diagram and front panel?

- What's the difference between a control, a constant and an indicator?

- What are the main tools in LabVIEW?

Data types & Structures

- Data Types
- Structures

Data Types

In LabVIEW we can deal with several types of data:

- String
- Numeric
- Boolean
- Enumerated
- Array
- Cluster

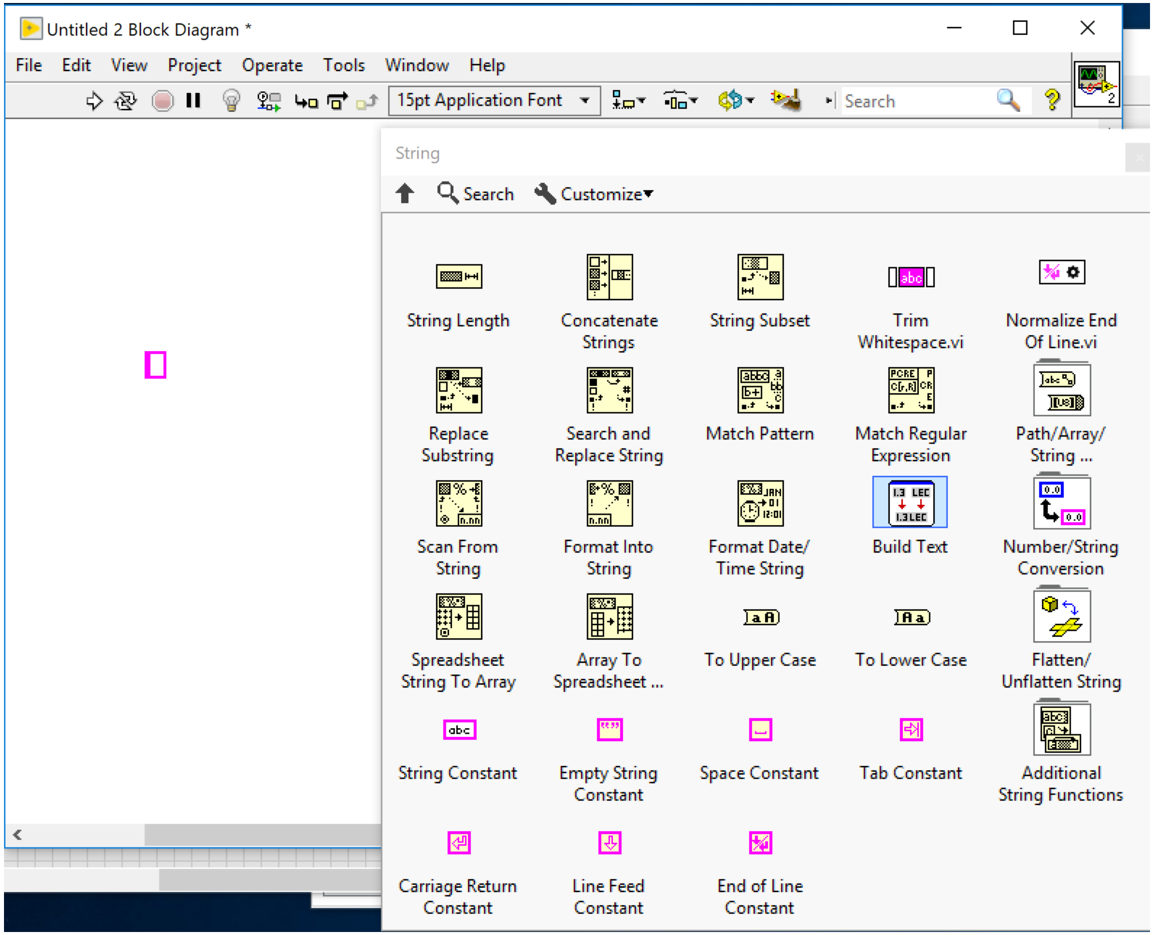
Each one is associated to a specific color in order to make it easy to distinguish them in the block diagram.

String

A string is just a sequence of ASCII characters.

In LabVIEW, it has to be delimited by a double " ".

The string colour is PINK.



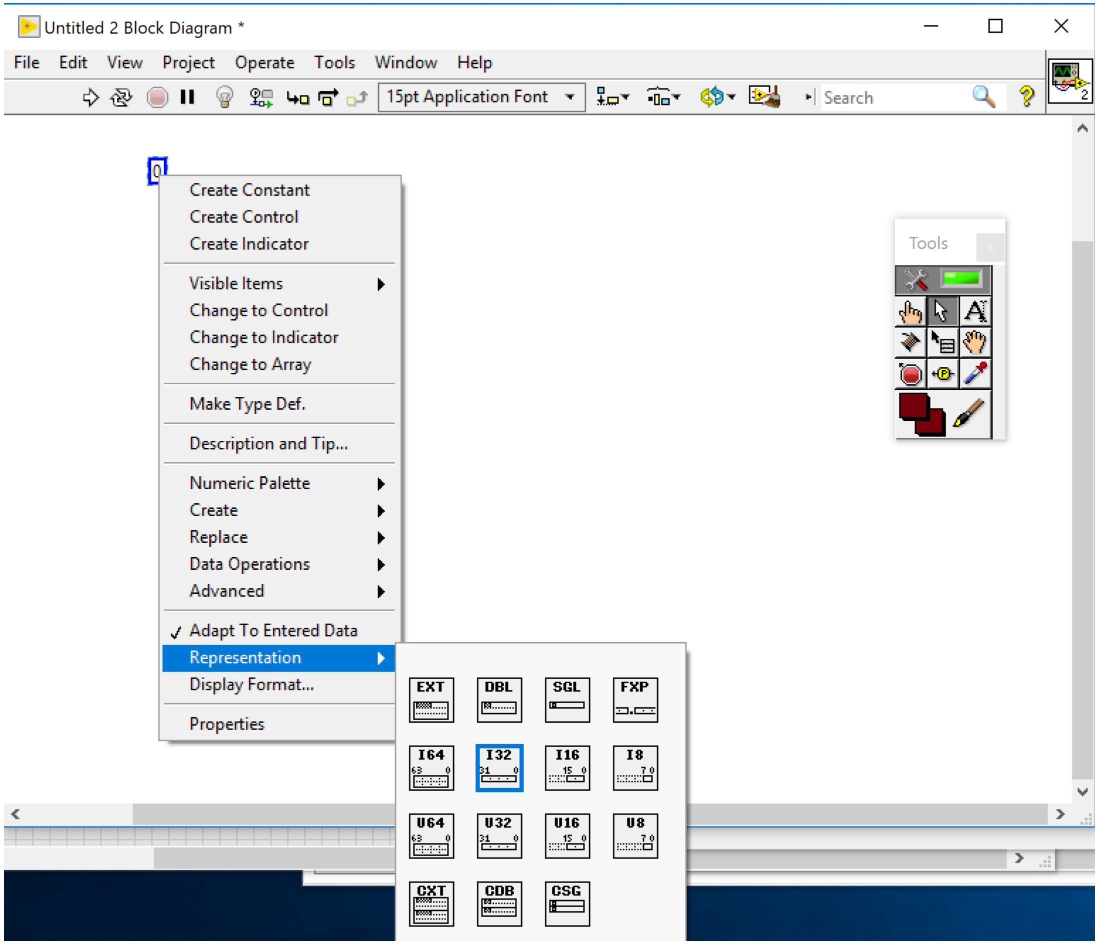
Numeric

In Labview there are several formats for numeric data:

- (un-)signed
- floating point
- fixed point
- extended

and resolutions:

- 8 bit
- 16 bit
- 32 bit

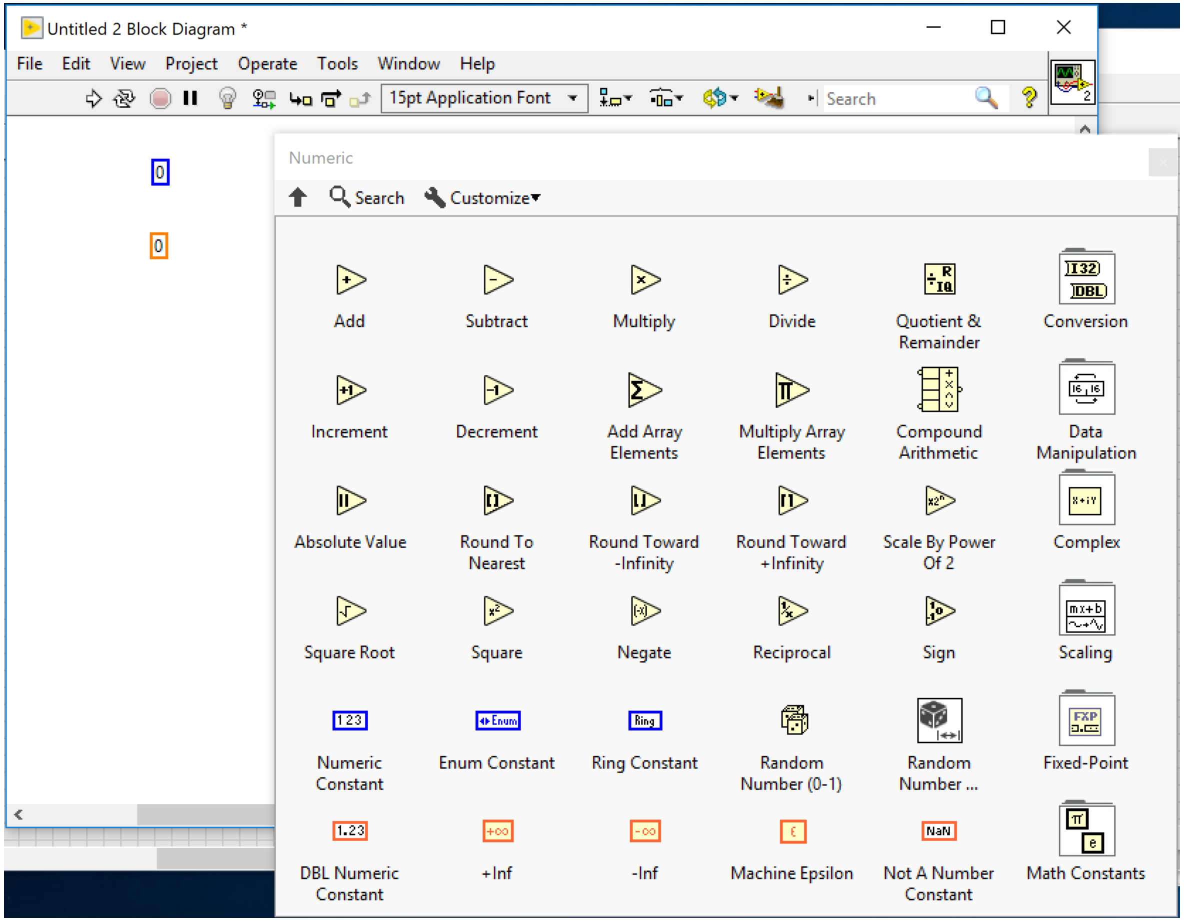


Numeric

The main numeric data types are:

integer (BLUE)

double (ORANGE)



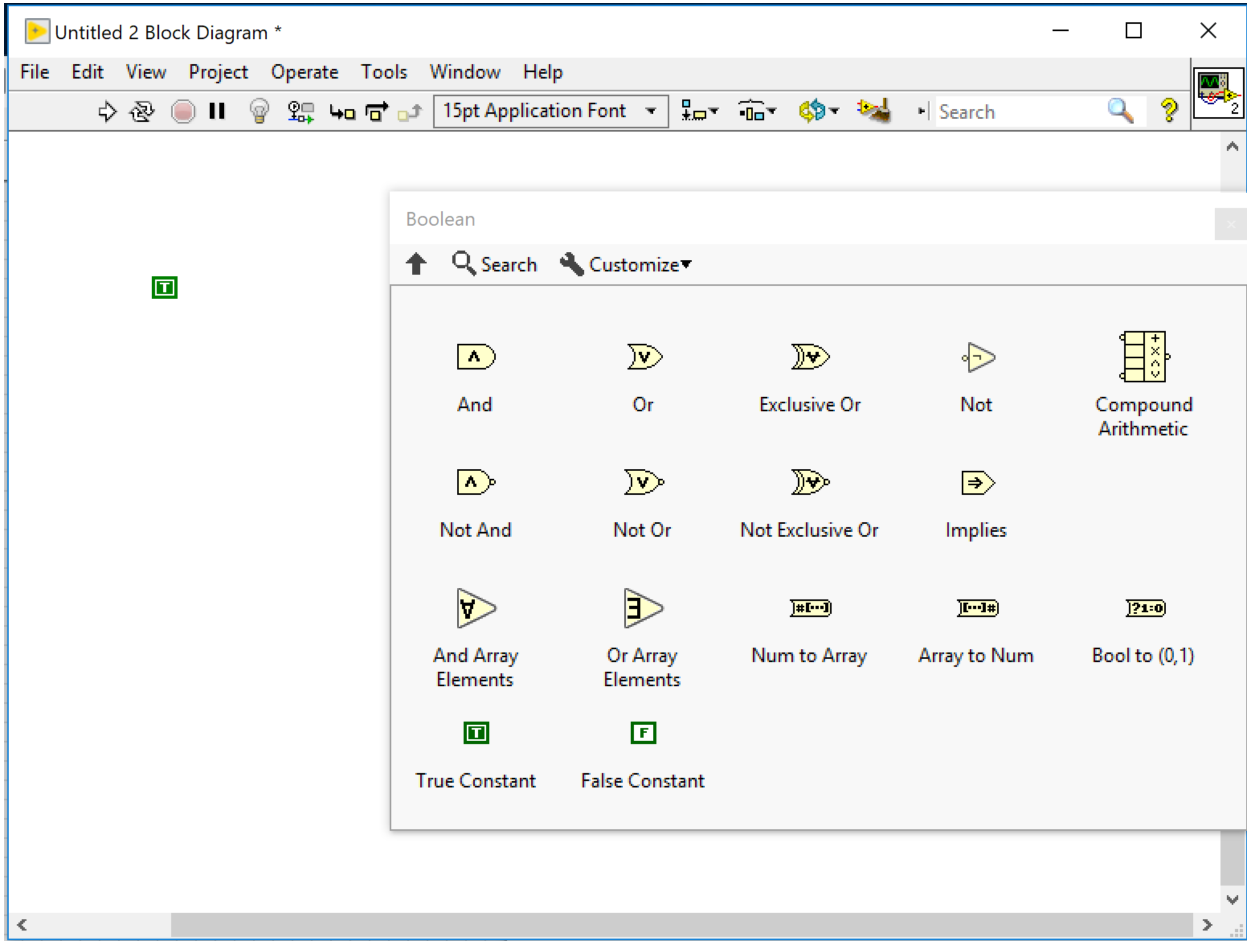
Boolean

Labview stores Boolean in 8-bit U8 constants:

0 → FALSE

1 → TRUE

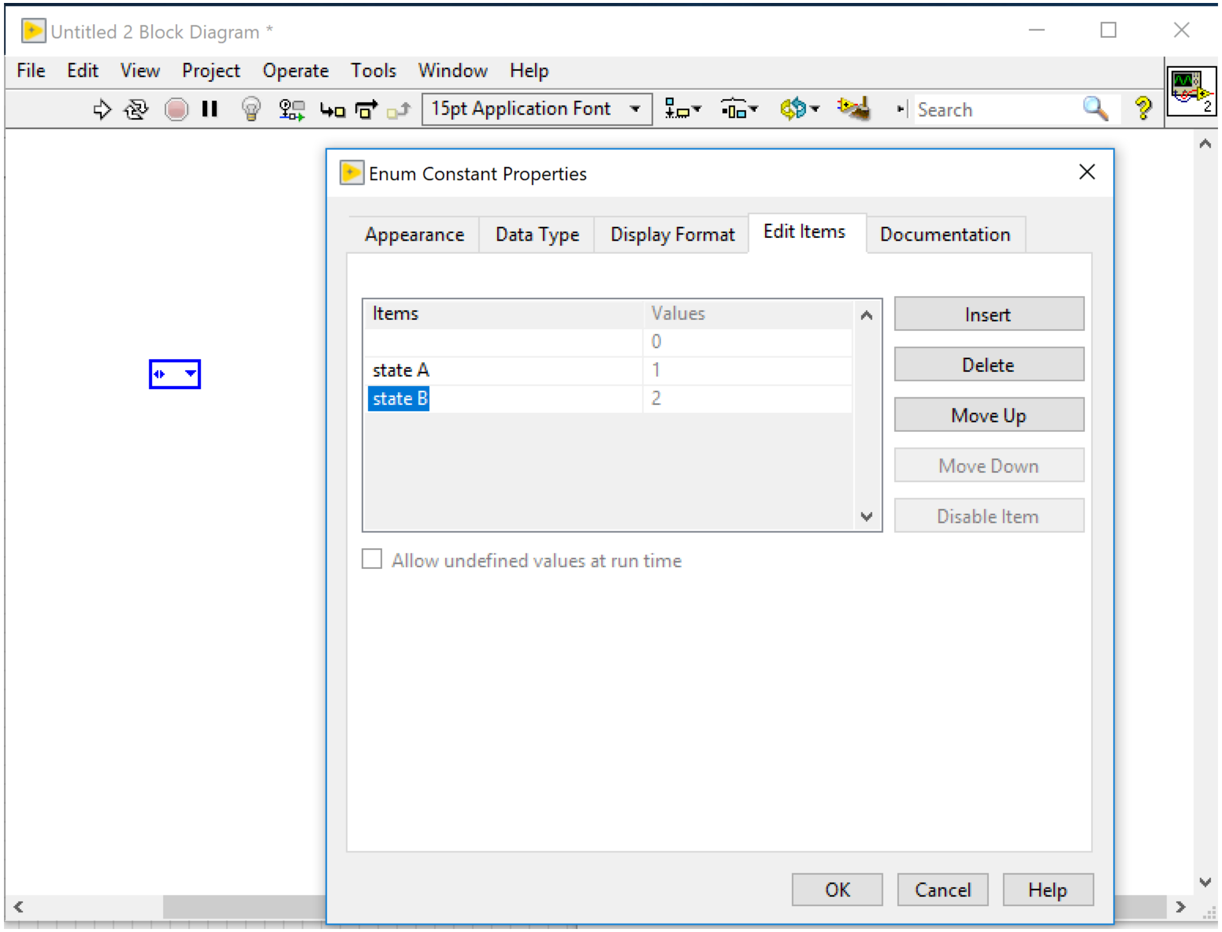
Boolean color is GREEN.



Enumerated

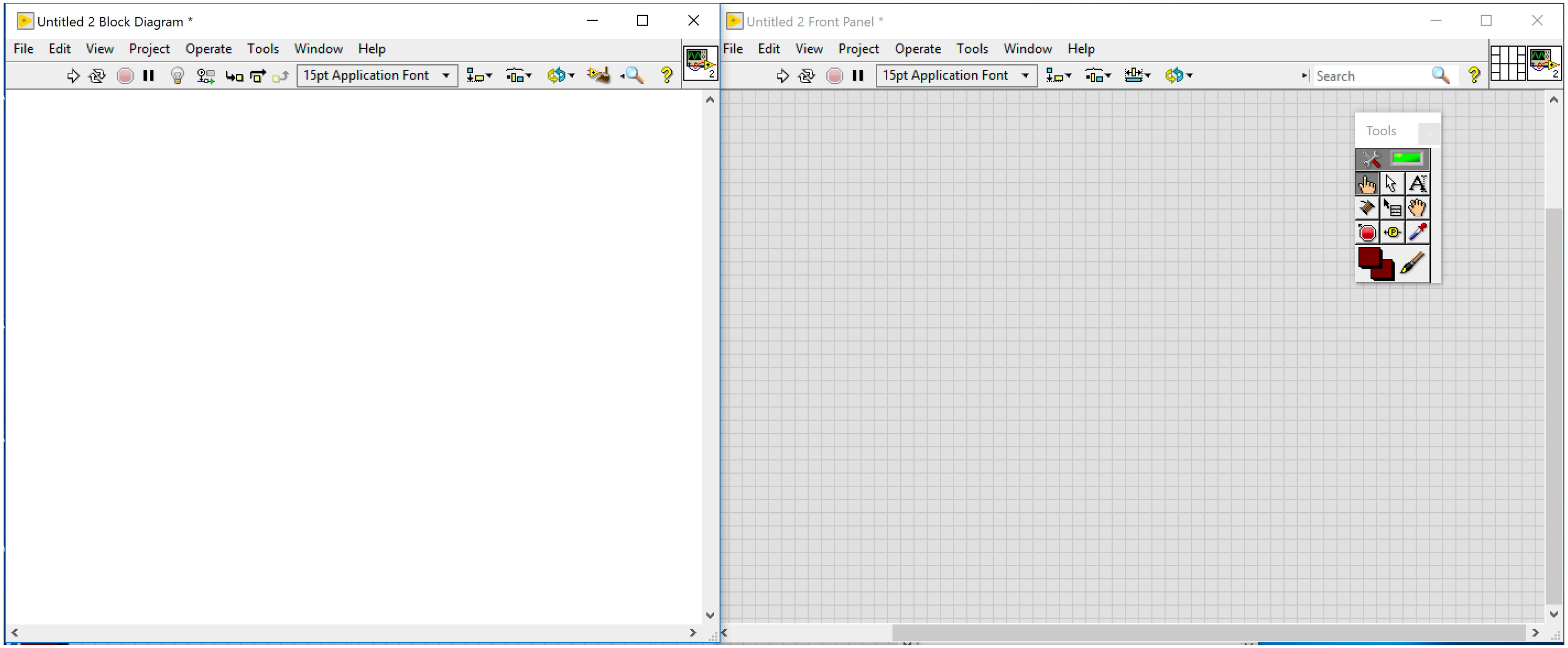
When you deal with a list of states, you can use an ENUM variable, i.e. an ordered list of strings.

Enum color is DARK BLUE (as an U8).



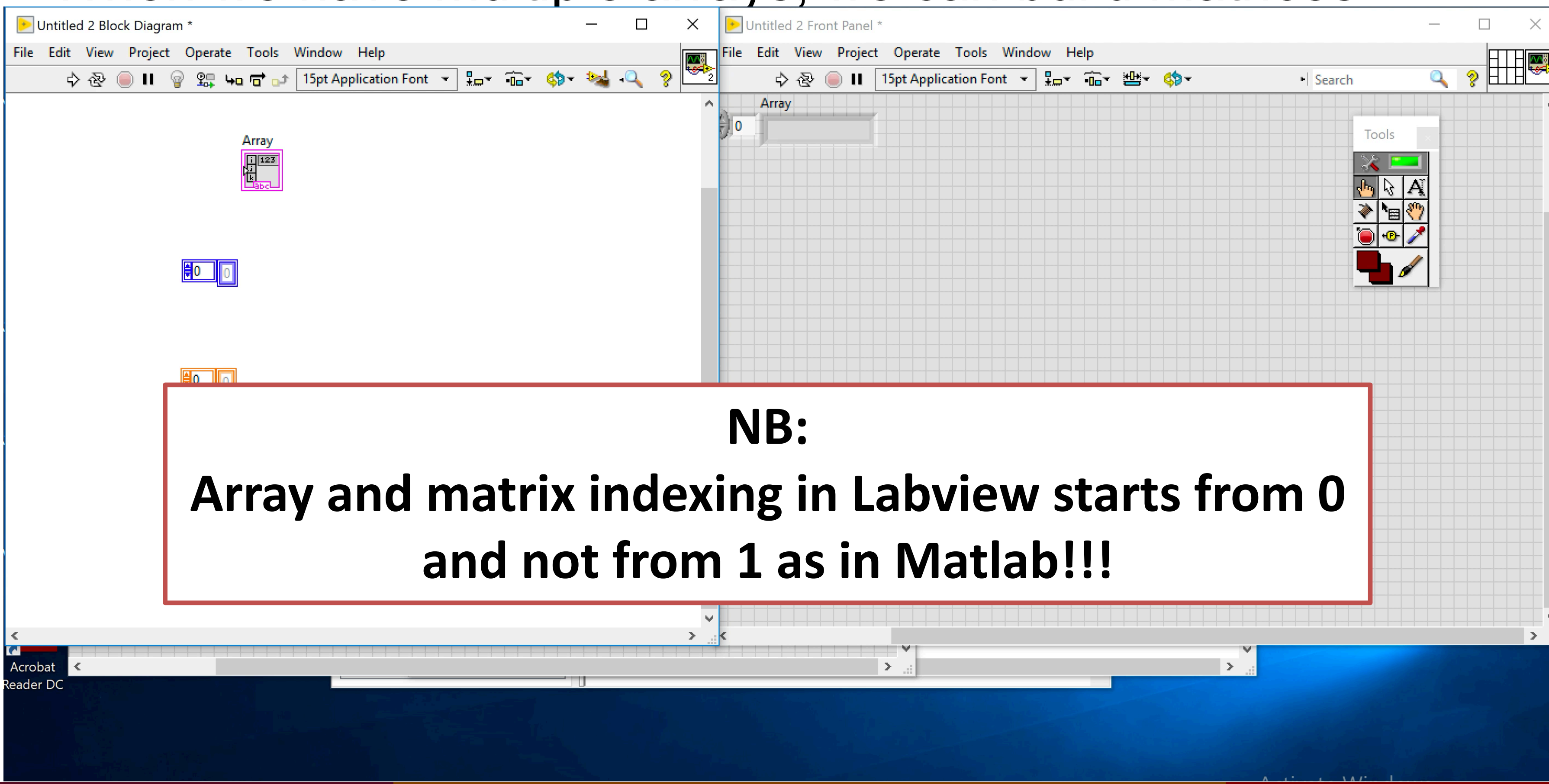
Arrays

If we need to store multiple values of the same data type:



Matrices

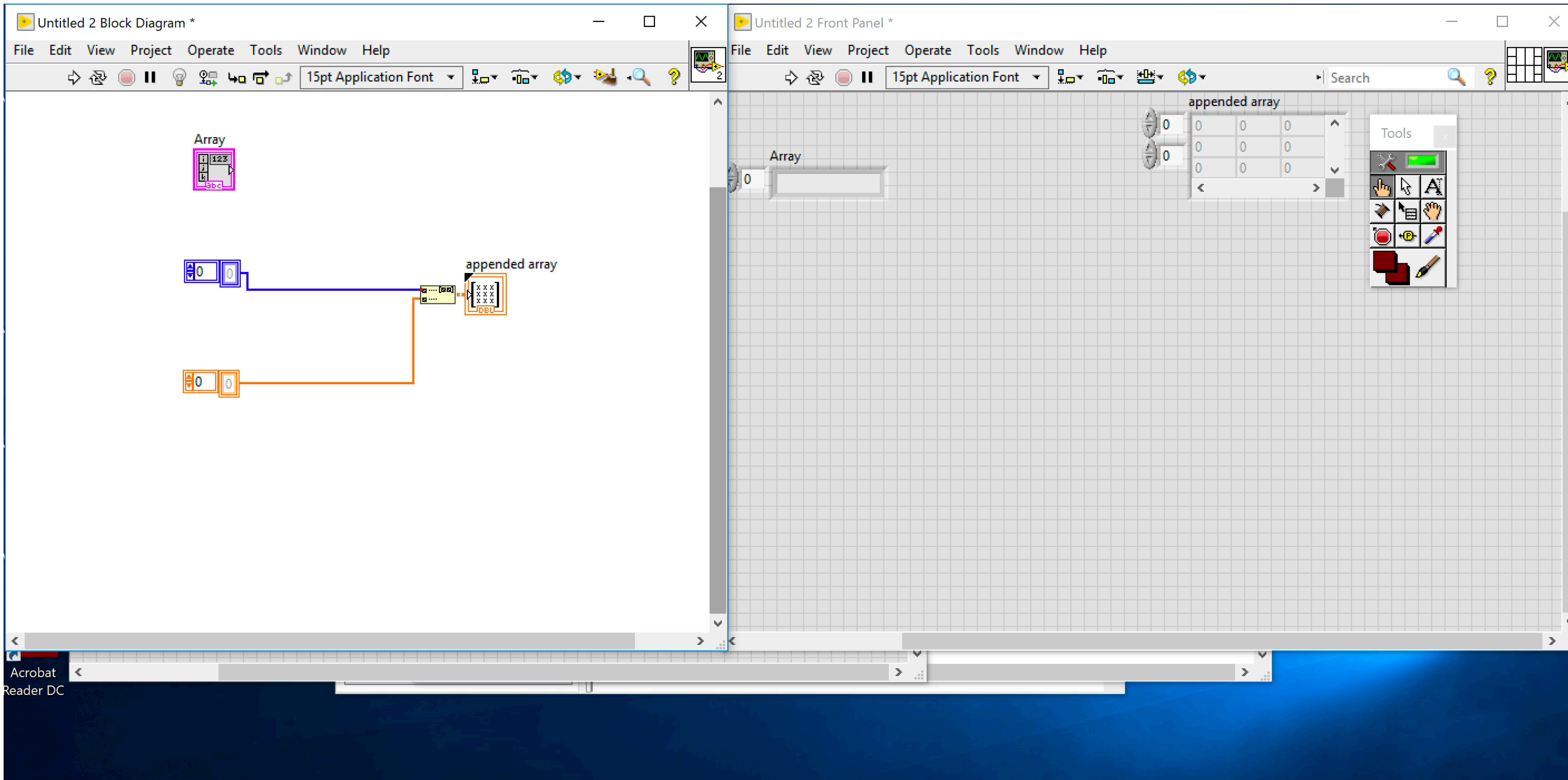
When we have multiple arrays, we can build matrices



NB:
Array and matrix indexing in Labview starts from 0 and not from 1 as in Matlab!!!

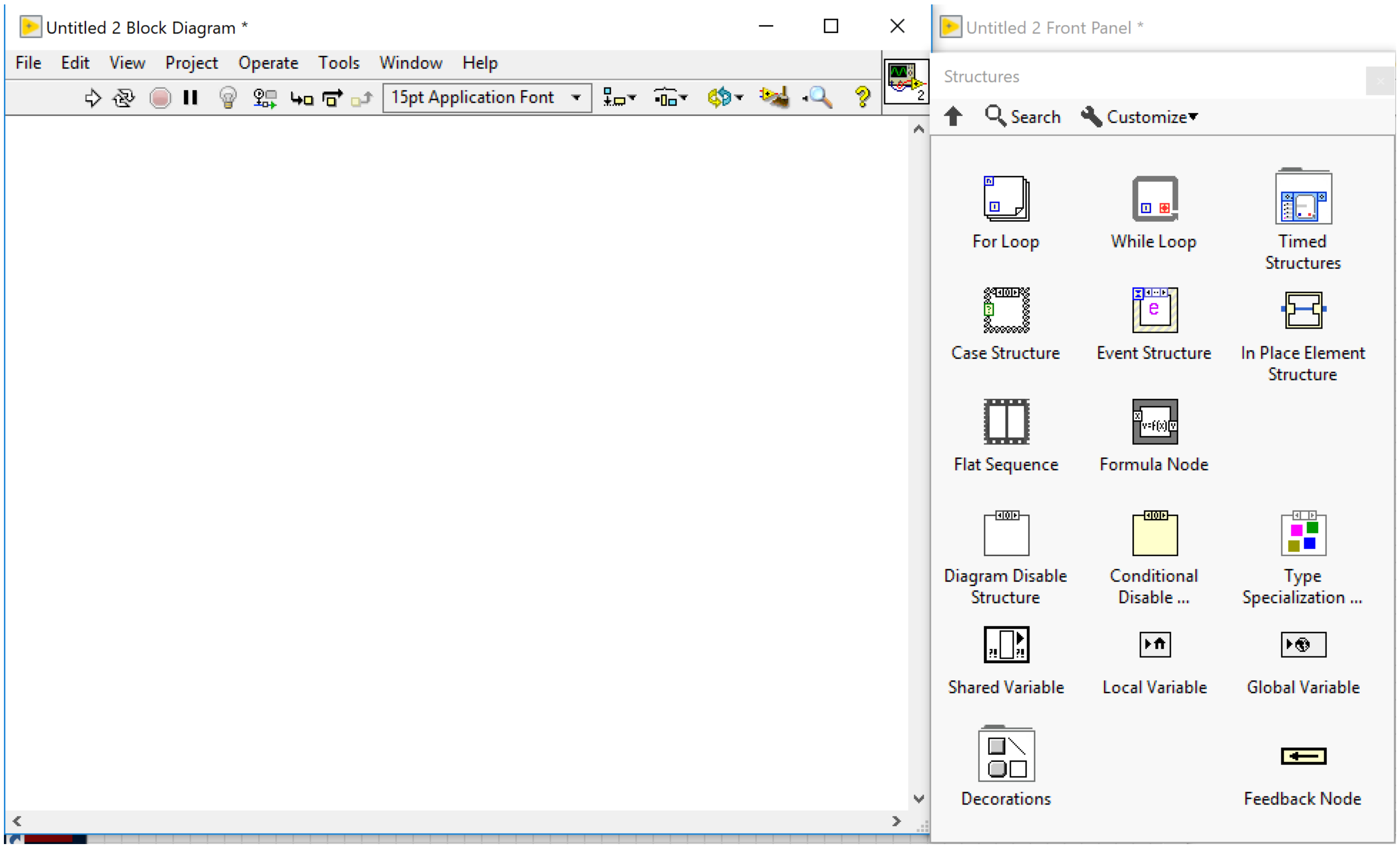
Clusters

If we need to store together different data types:



Structures

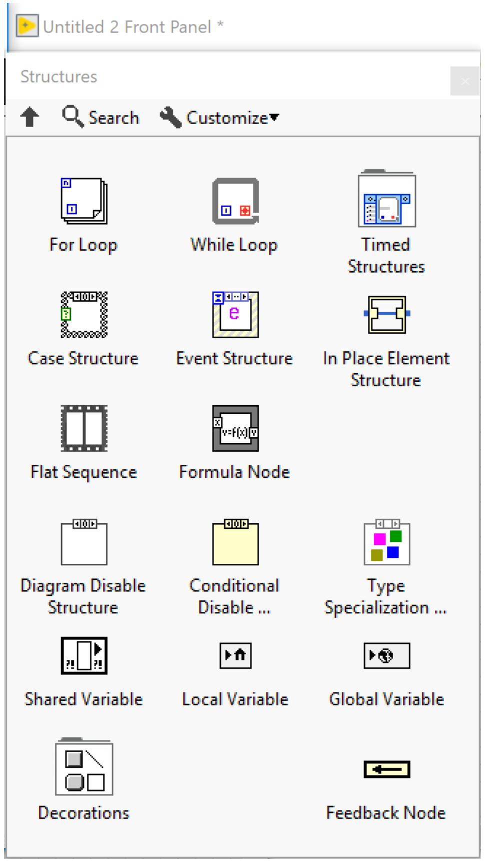
In a block diagram we can set loops and repetitions:



Structures

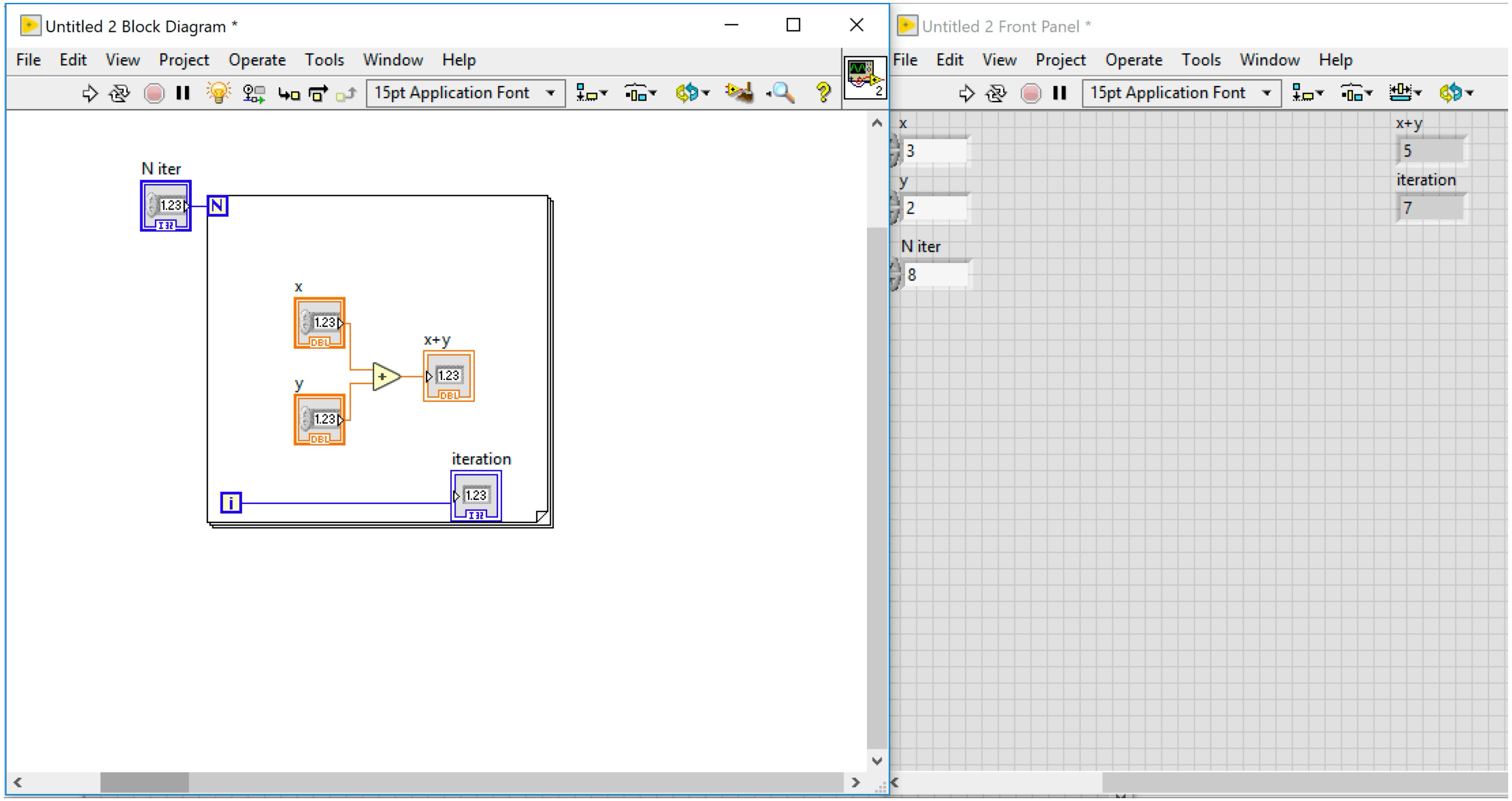
In LabVIEW the main structures that we can use are:

- For Loop
- While Loop
- Disable Structure
- Case Structure



For Loop

Whatever is within the loop is repeated iteratively N times



Tunneling options

Sometimes, we are interested in outputting some of the results computed within the loop.

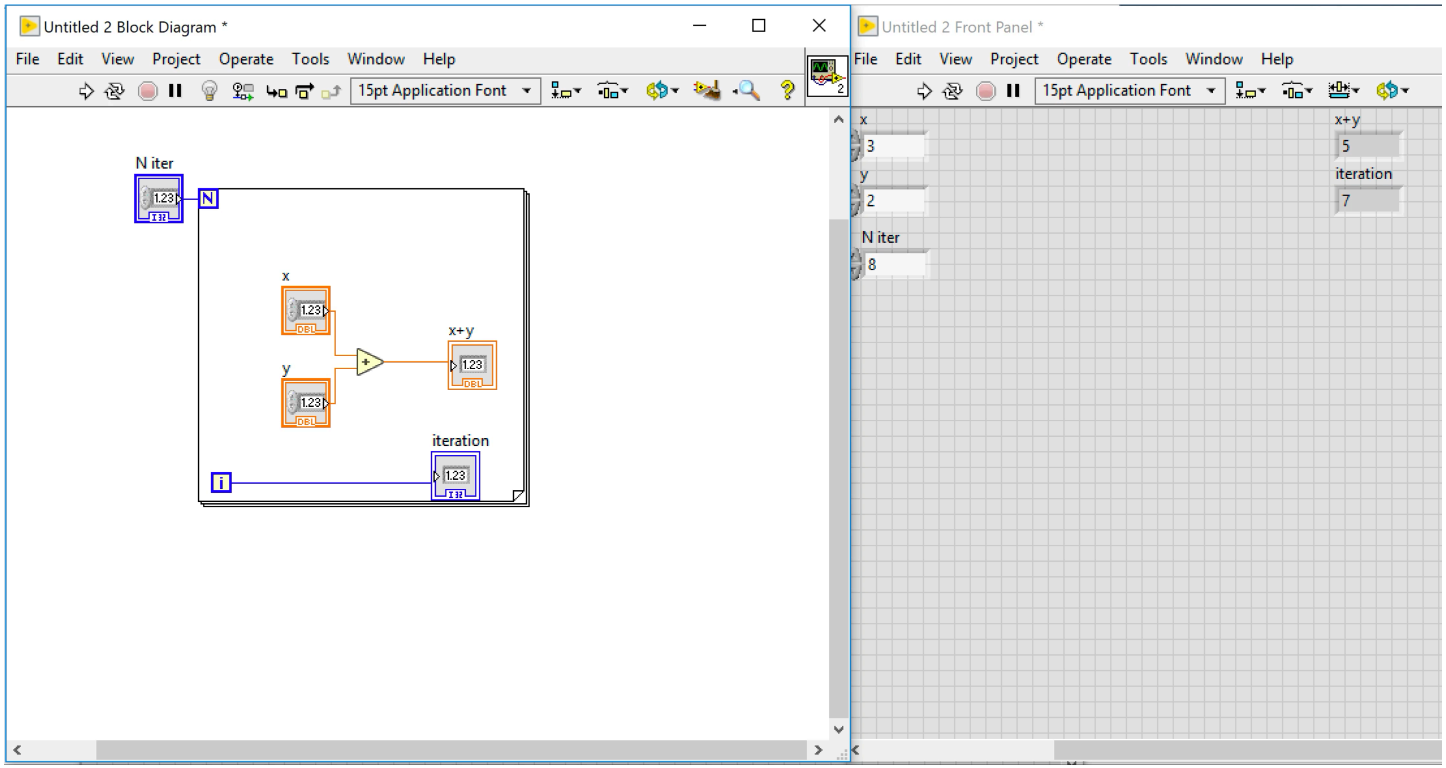
We have to create a sort of tunnel between the loop and the outer block diagram.

There are several tunneling options:

- Last value
- Indexing (concatenating for arrays)
- Conditional
- Shift register

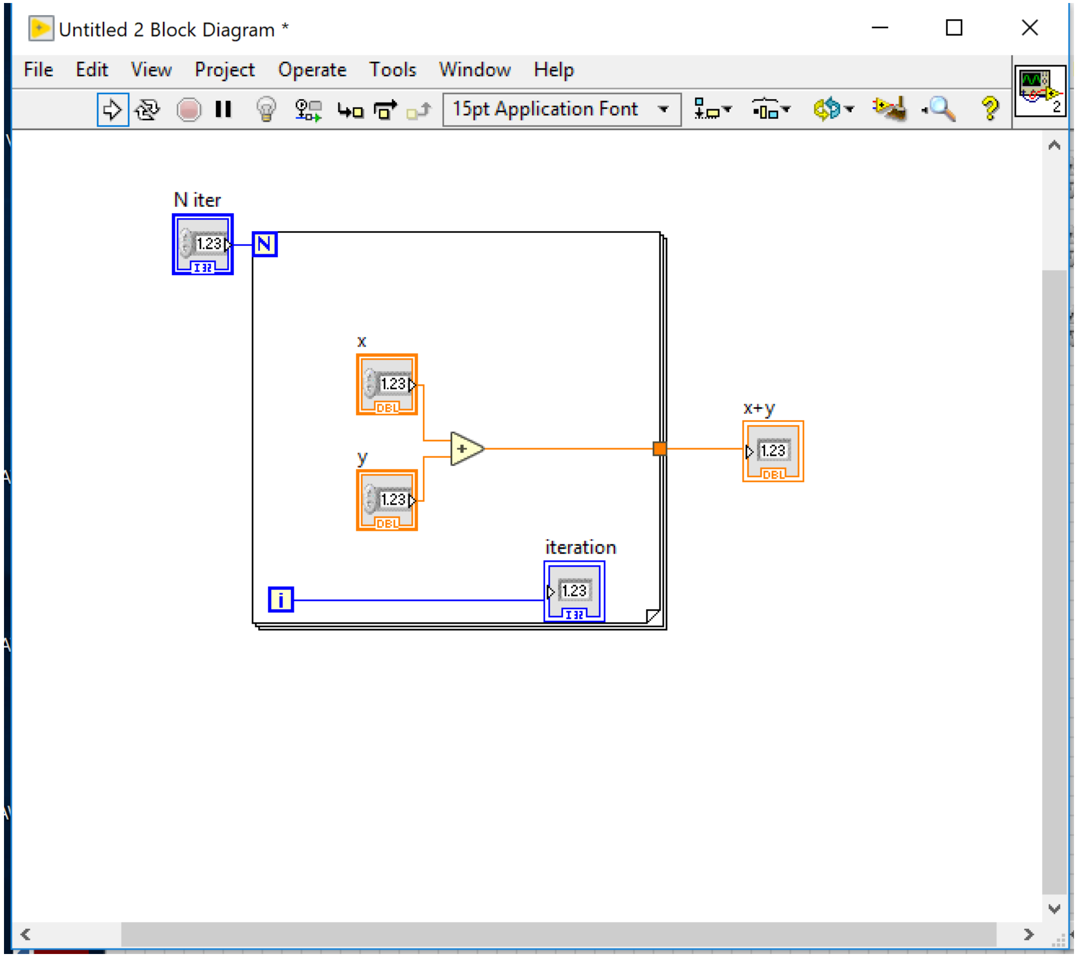
Tunneling options

Right click on the tunnel point:



Last value

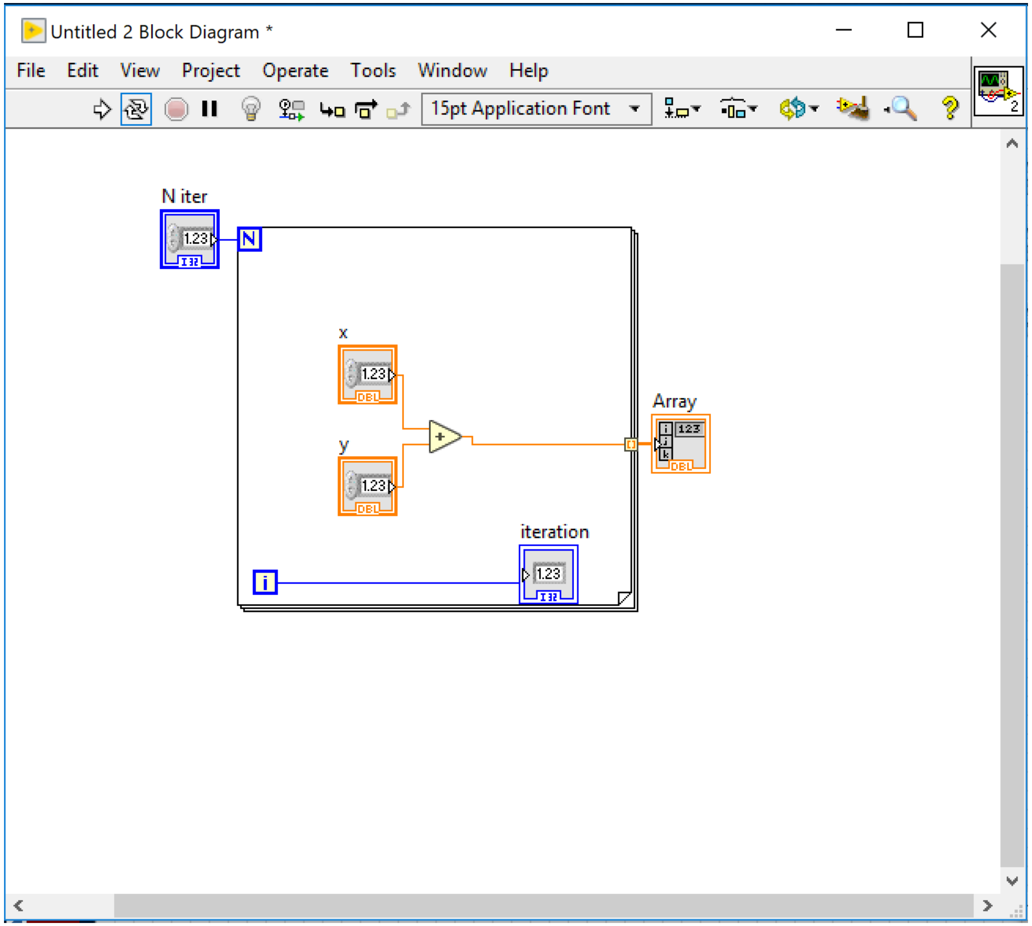
Only the value of the last iteration is saved in the output.



Indexing

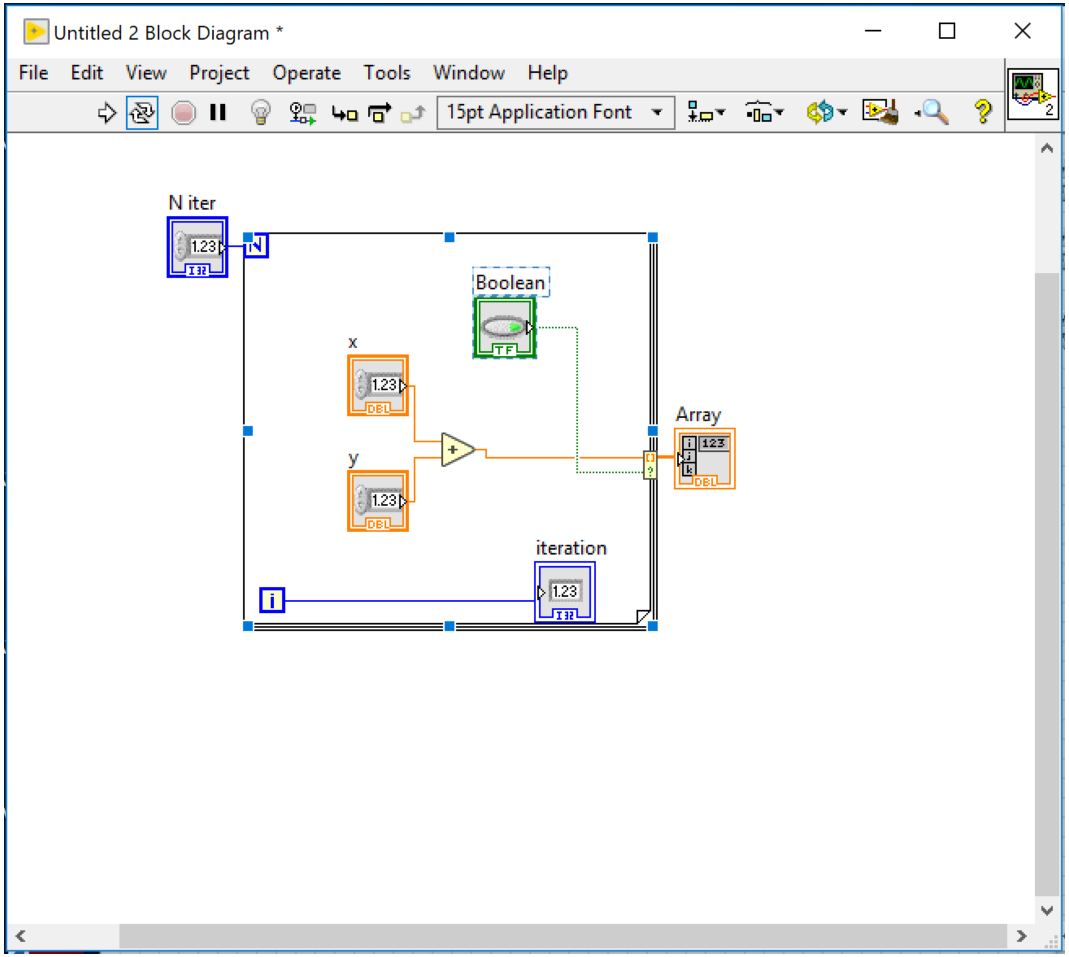
The result of each iteration is an entry of the output array.

If we are working with arrays, the option Concatenating produce a matrix.



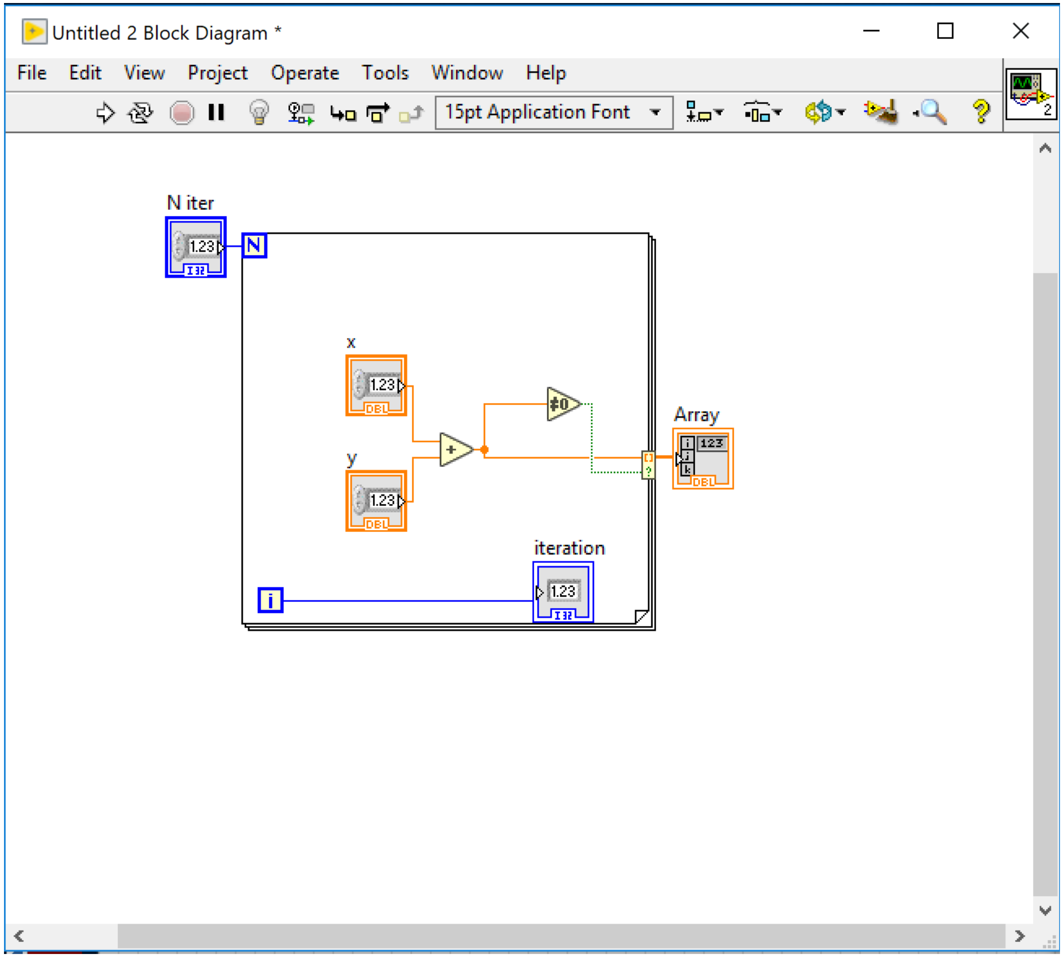
Conditional

In this case, the output is saved only if the Boolean control is set equal to TRUE.



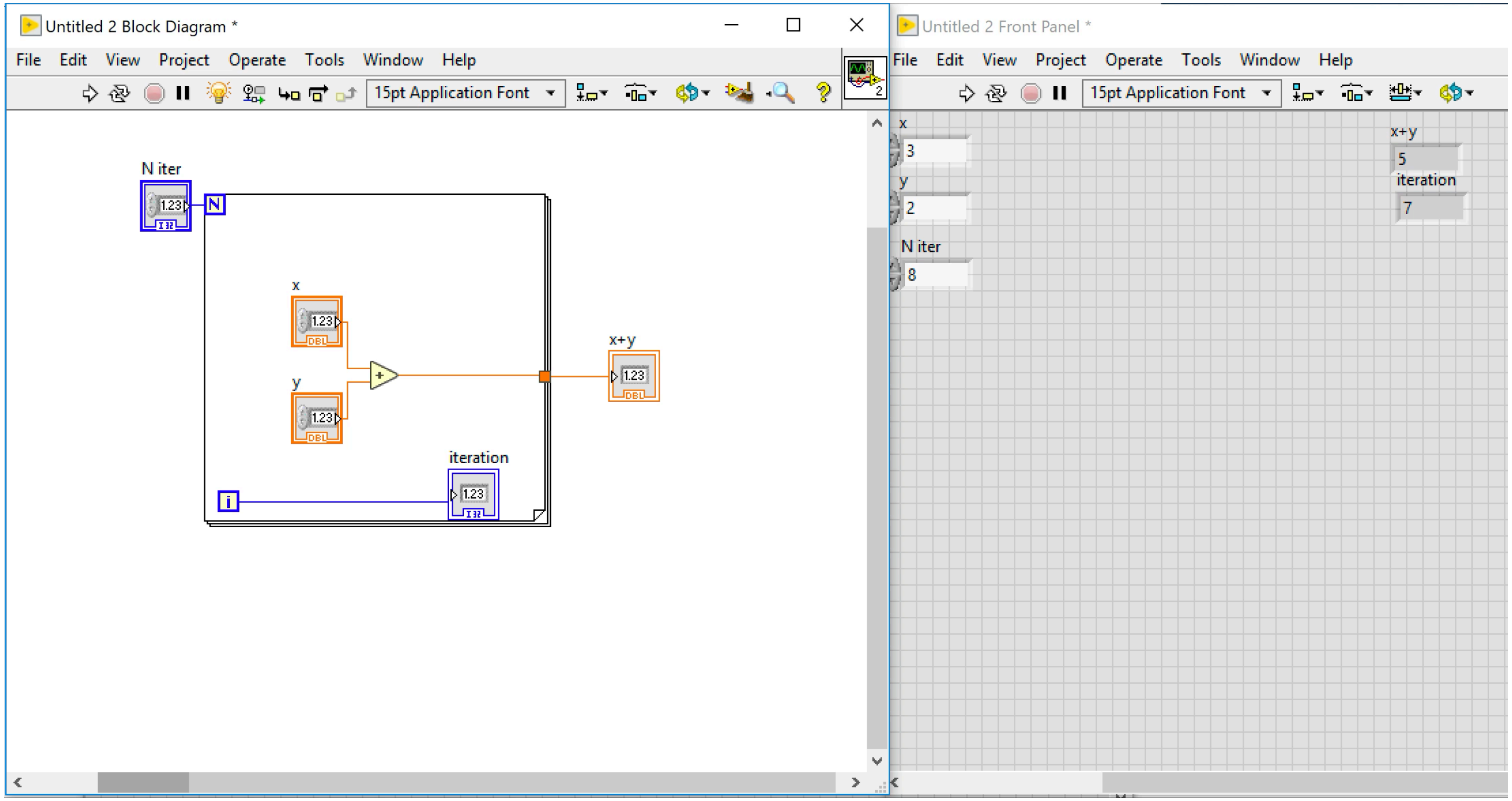
Conditional

… or a condition given by a Boolean variable.



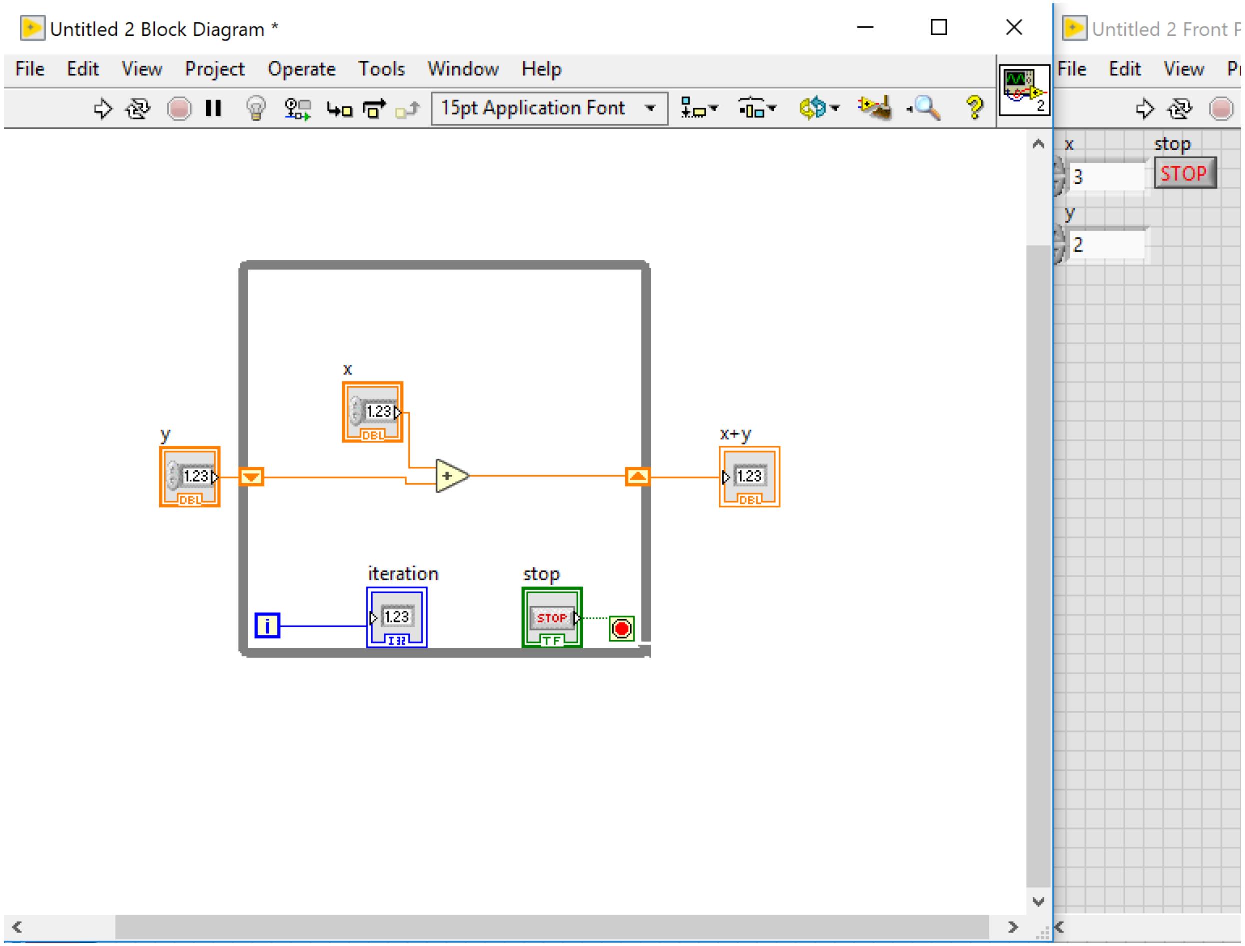
Shift register

Sometimes, we need to update the previous iteration result.



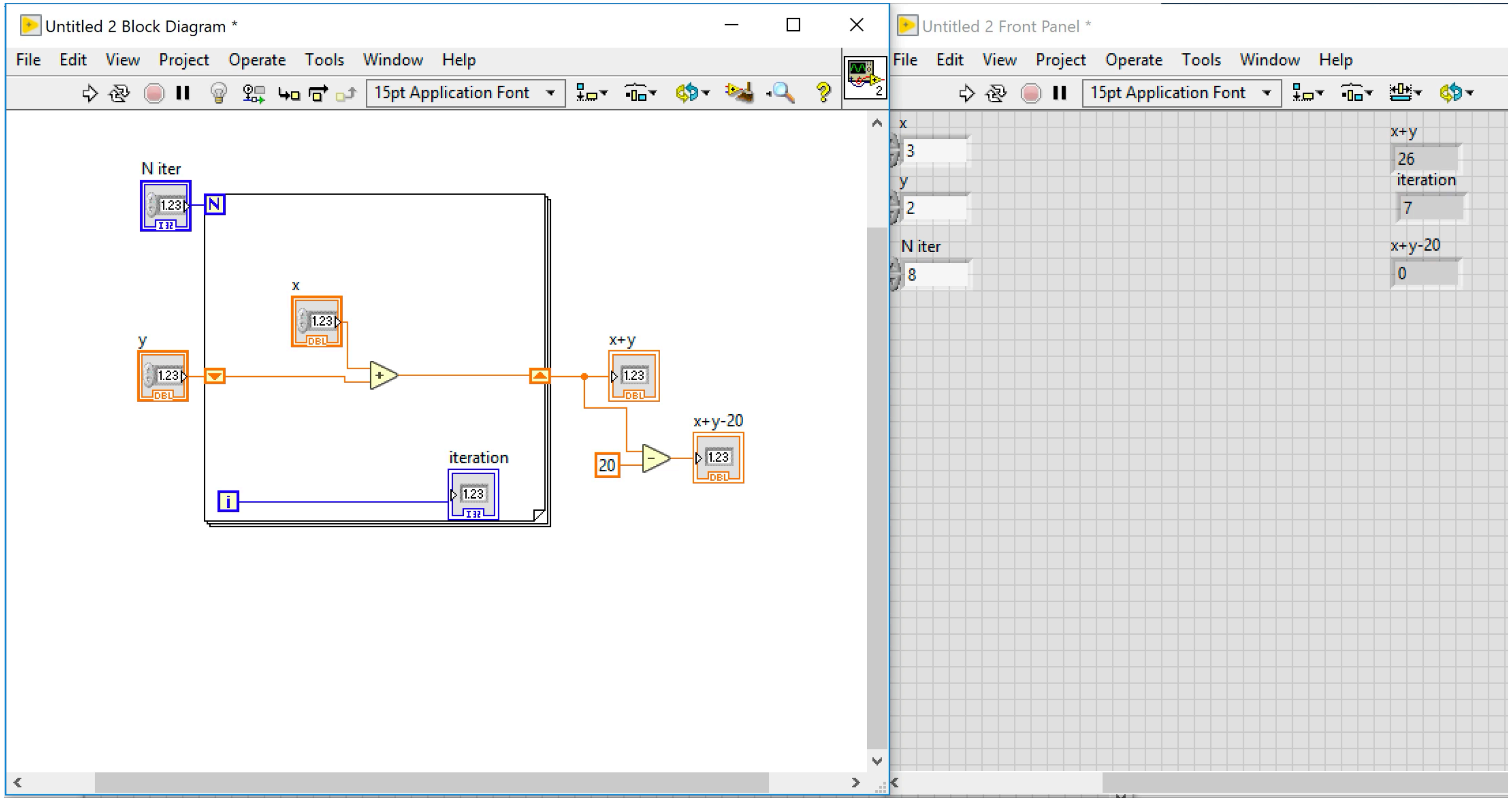
While Loop

The difference with the For Loop is just the stop criterion that is now ruled by a Boolean variable.



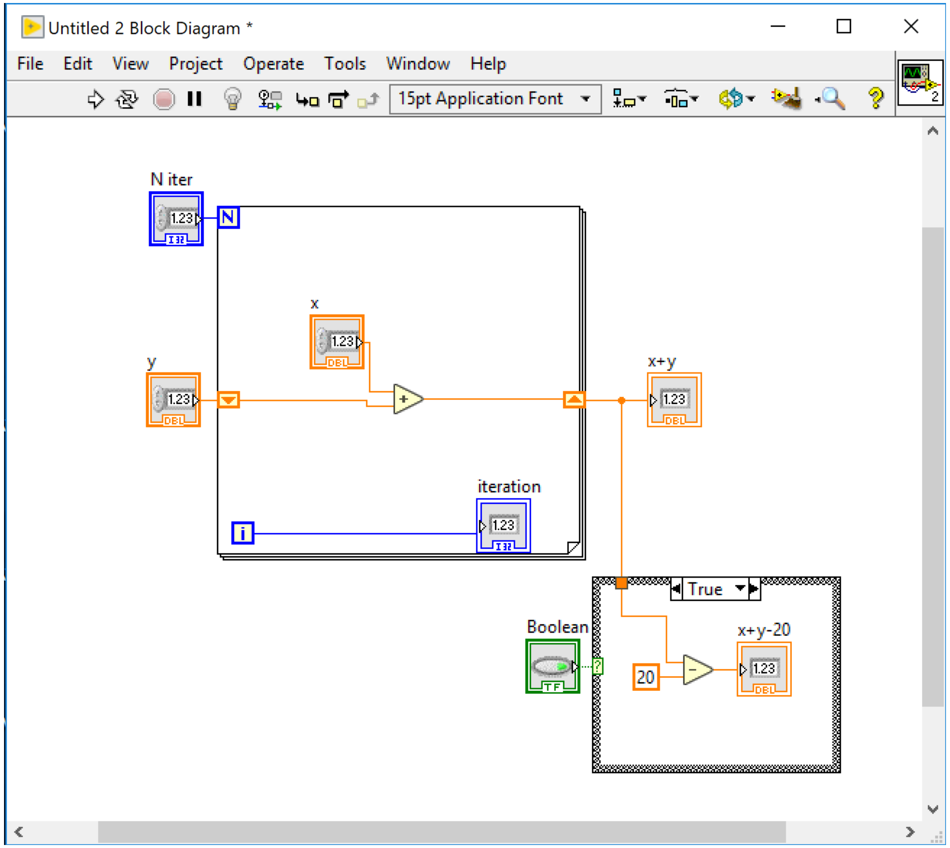
Disable structure

The Disable structure is like the comment in Matlab.



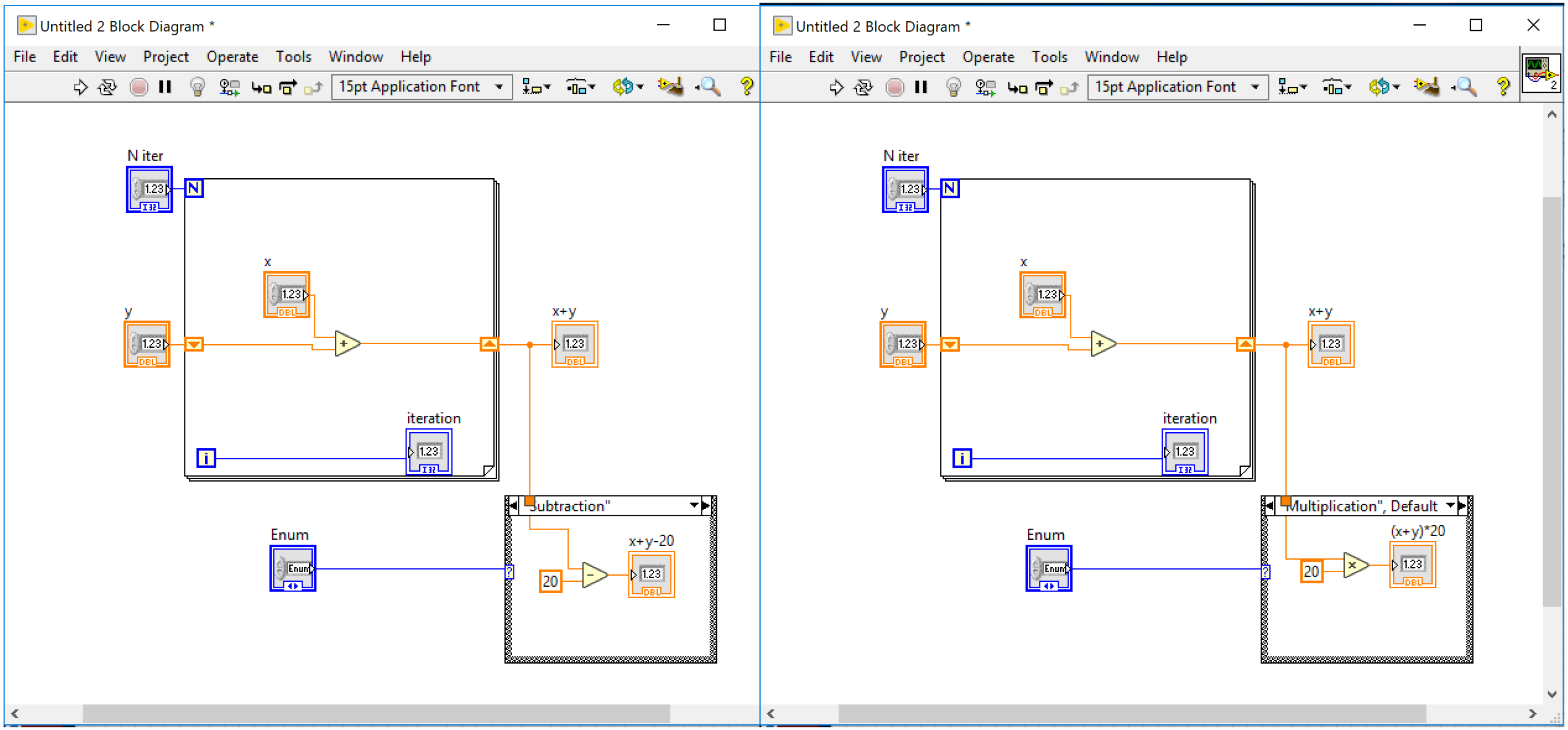
Case structure

The Case structure is like the if … else … end in Matlab.



Case structure

The Case structure is like the if … else … end in Matlab.



Recap: was everything clear?

- What are the main data types in LabVIEW?
- What are the main structures in LabVIEW?
- What are the possible tunnel options in an iterative cycle (e.g. For & While loop)?

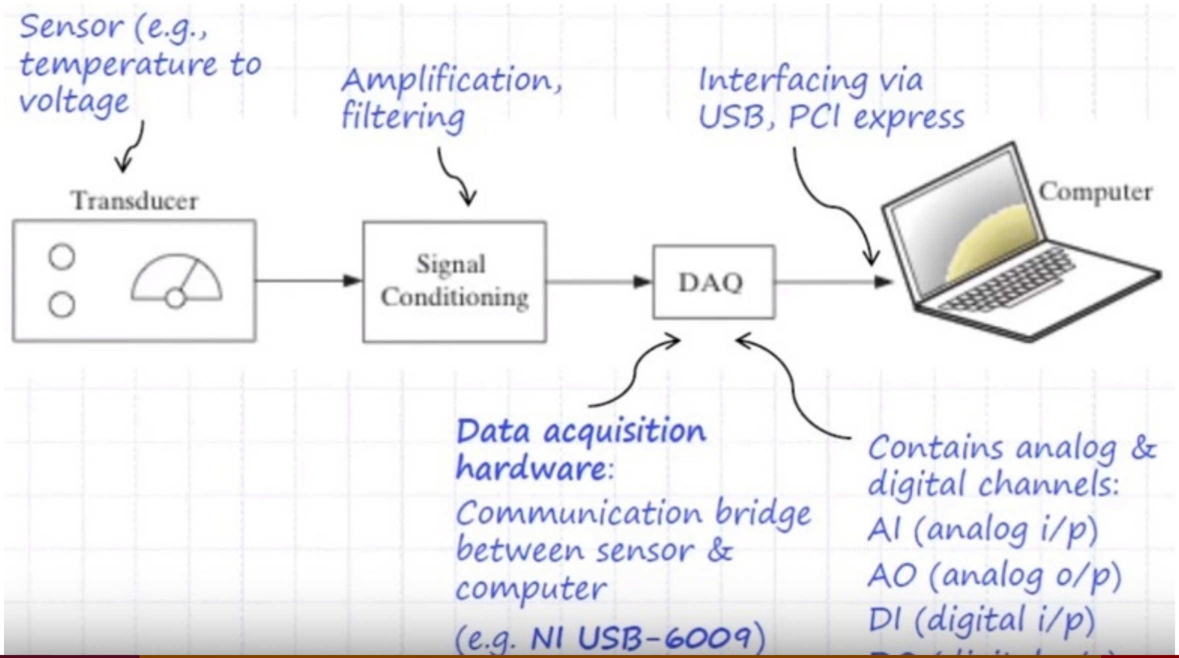
DAQmx

- Data acquisition
- DAQmx library
- Example

DAQ system

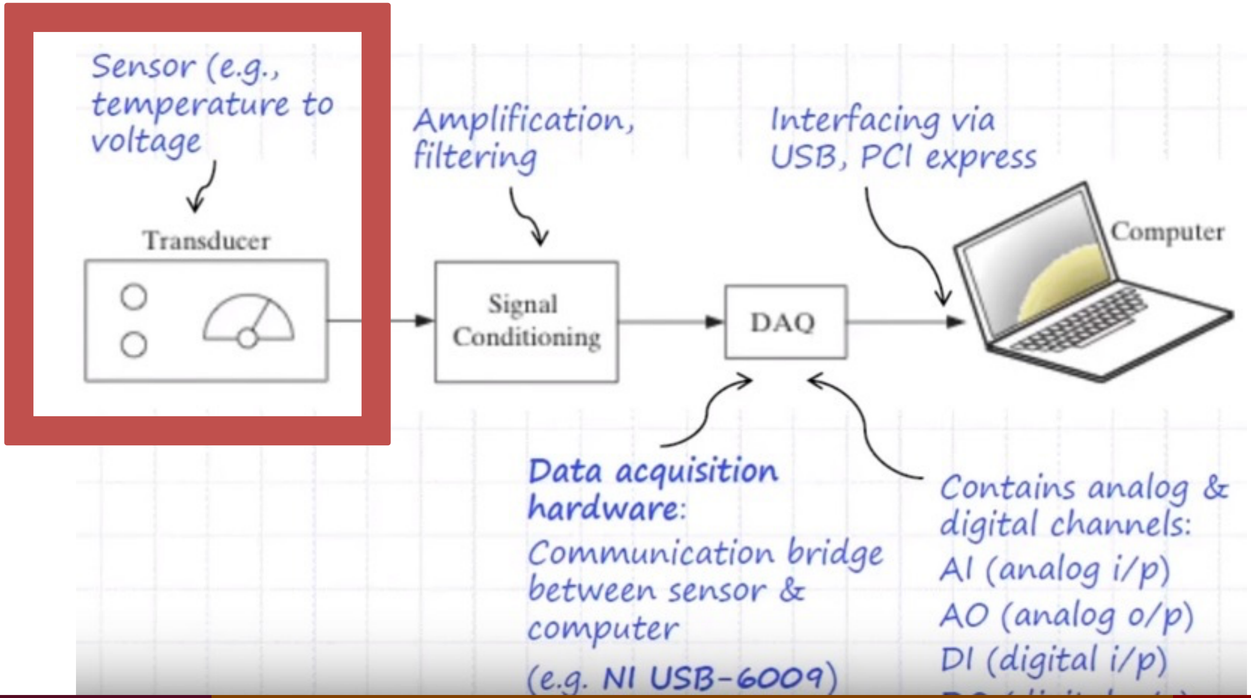
Data acquisition is a process that

1. gathers (analogue) signals from measurement sources;
2. digitizes the signals to store, analyse, and plot on PC.



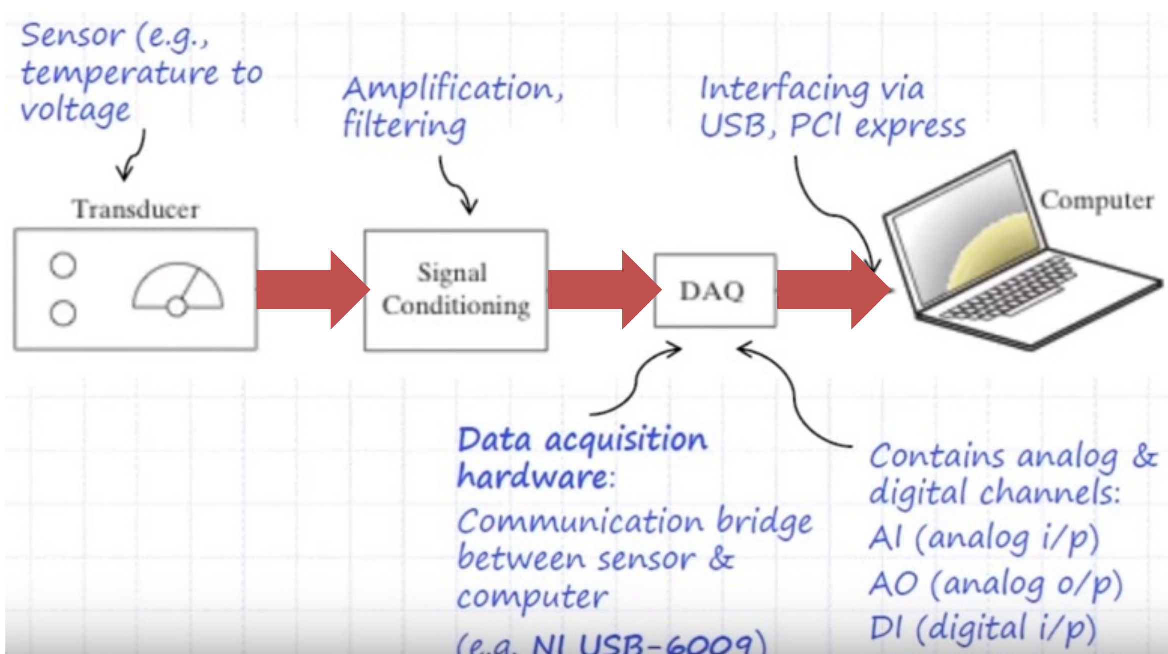
Transducer

Transducer: a device that converts a physical phenomenon into a measurable electrical signal, i.e. voltage or current.



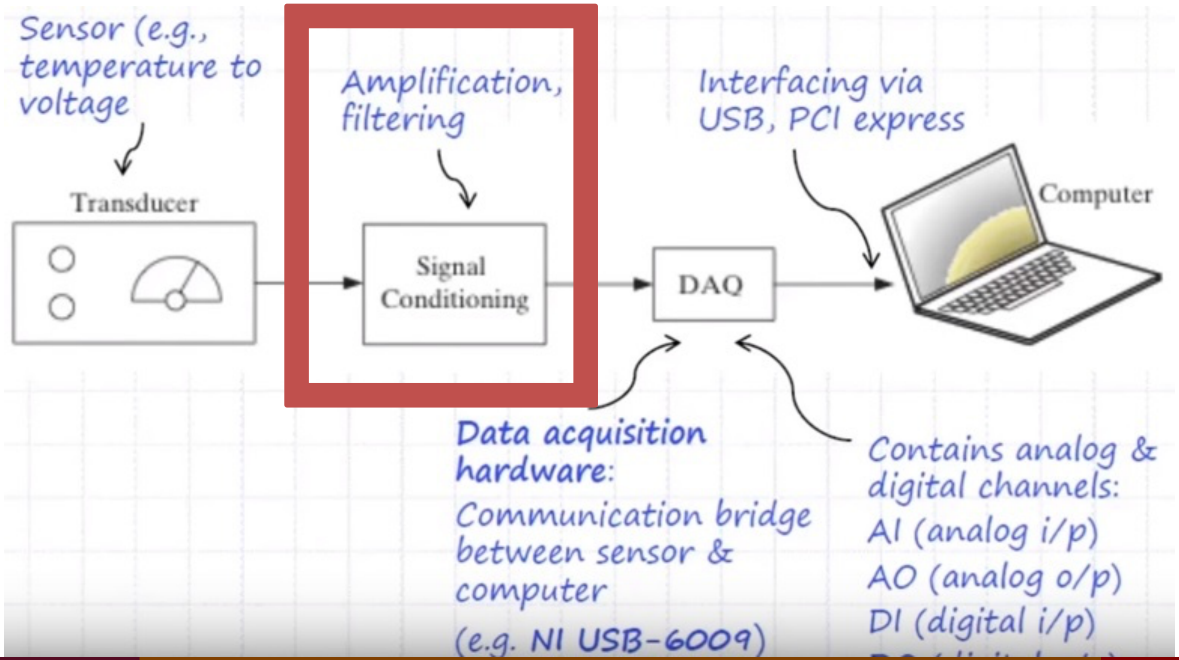
Signal

Signal: a detectable physical quantity or impulse (such as a voltage, current, or magnetic field strength) by which messages or information can be transmitted.



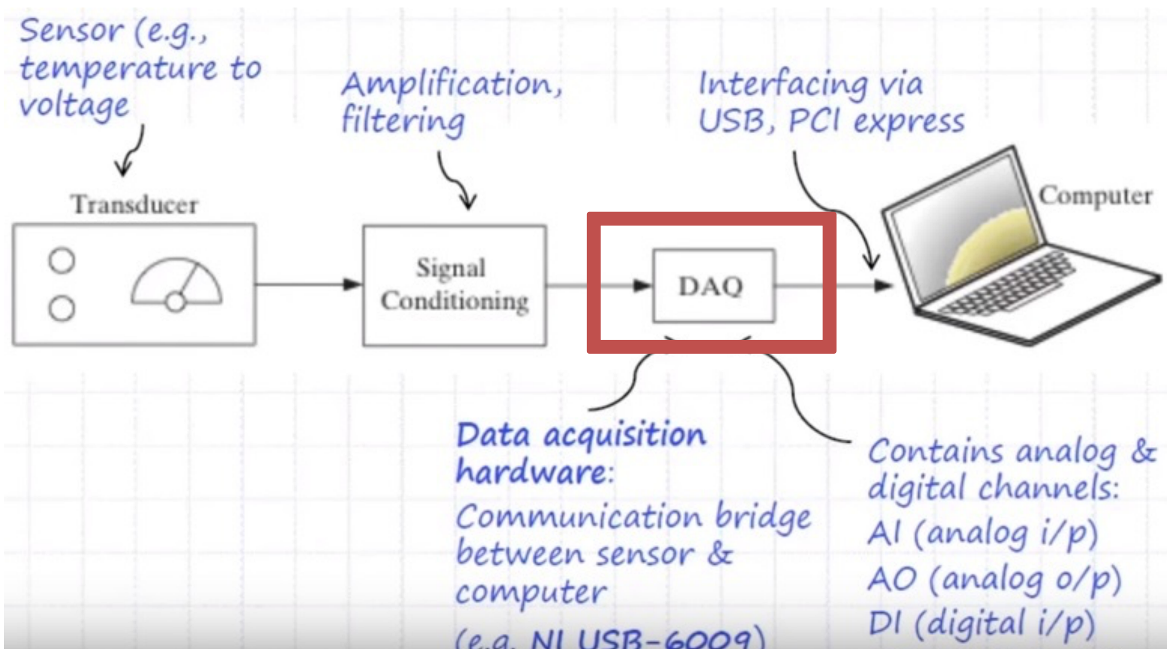
Signal conditioning

Signal conditioning: a set of operations that maximizes the accuracy of a system, allows sensors to operate properly, and guarantees safety (e.g. attenuation, isolation)



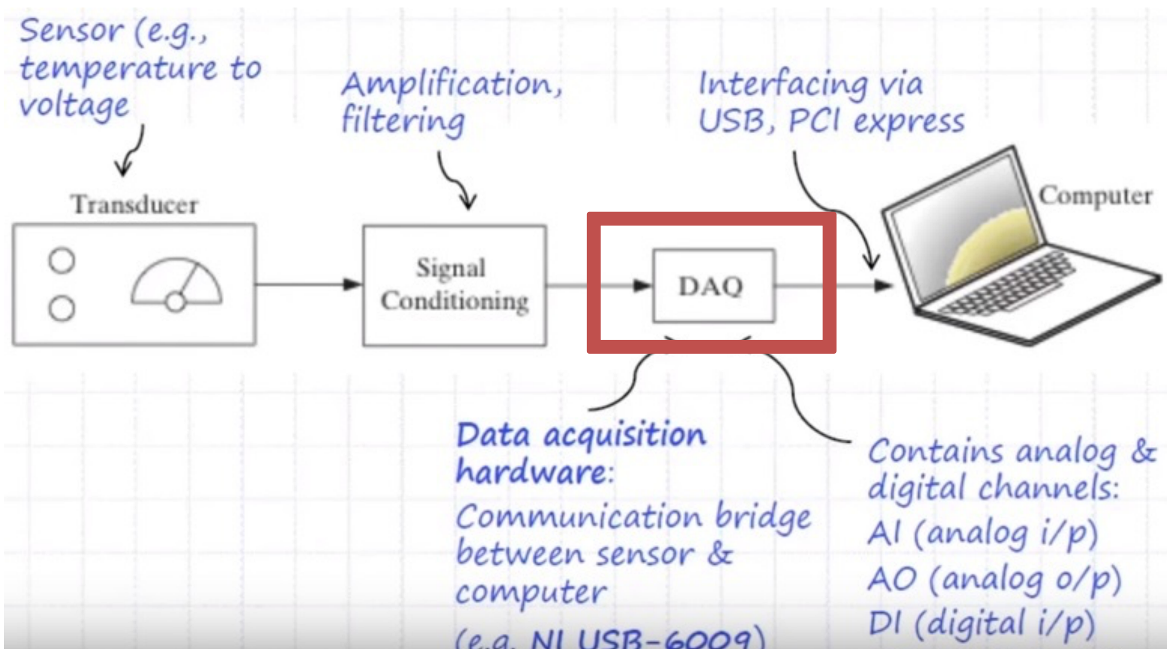
DAQ Hardware

Data acquisition hardware: a device that digitizes incoming analog signals so that the computer can interpret them.



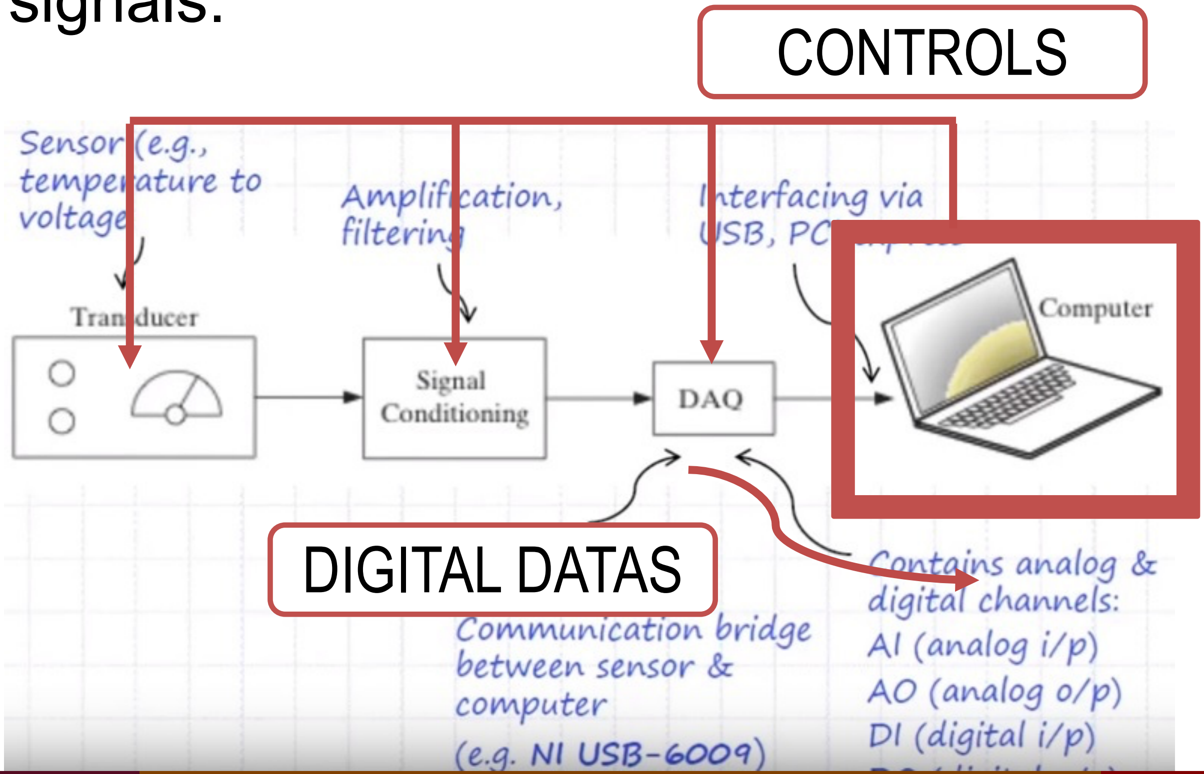
DAQ Software

Data acquisition software: transforms the PC and the data acquisition hardware into a self- tool for data acquisition, analysis, and display.

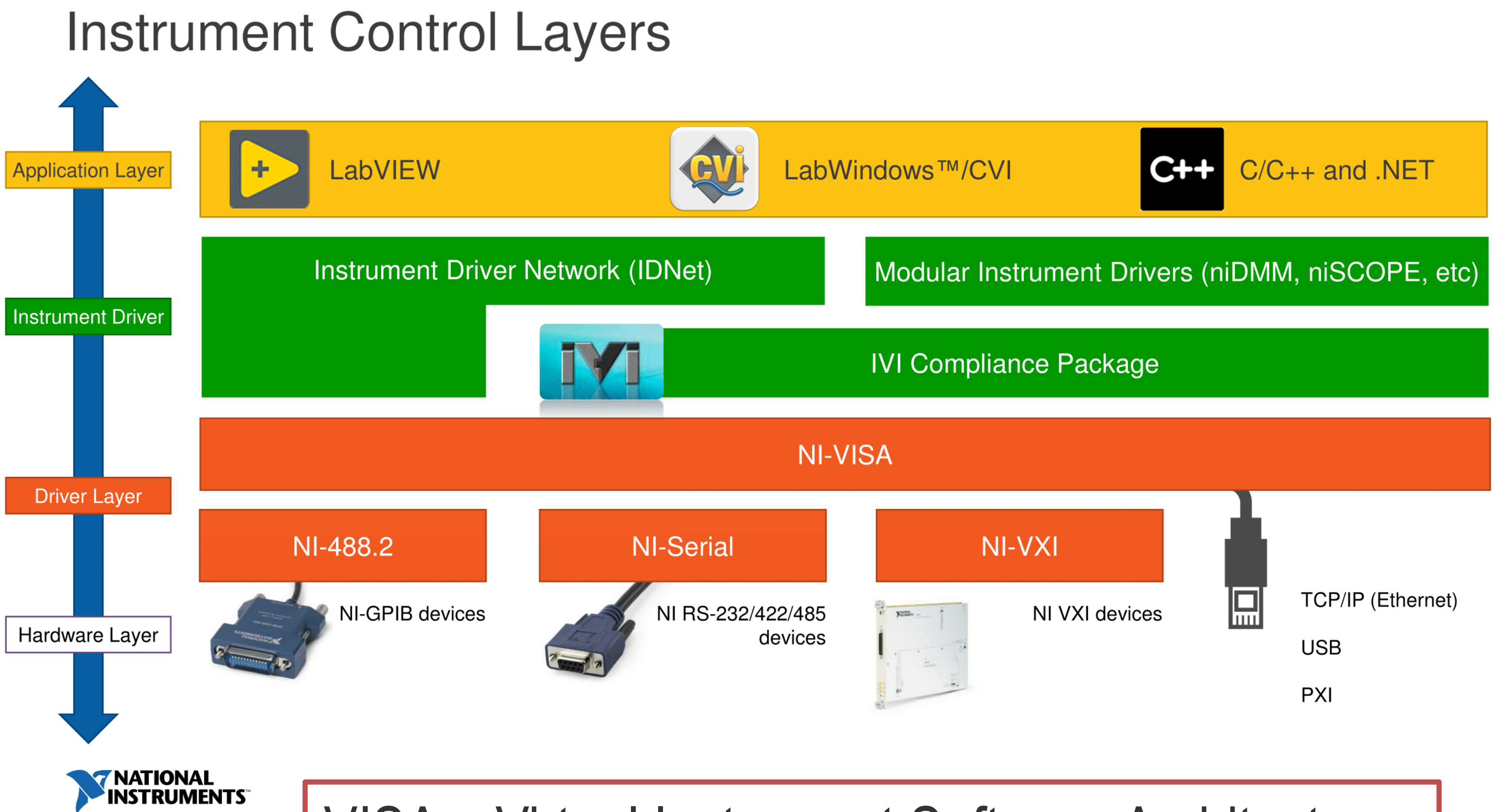


Computer Interface

Computer interface: a computer that controls and triggers the acquisition system, and stores / processes the digitally acquired signals.



Instrument control



VISA = Virtual Instrument Software Architecture

Drivers

- Instrument drivers

  a library for a specific class of instruments, typically specific for a given vendor or model

  EXAMPLE: Lecroy Oscilloscope HDO4034

- Driver Layer

  acts as an interface between the application software and the DAQ hardware, and prevents a programmer from having to do register-level programming or complicated commands.

  EXAMPLE: NI DAQmx

NI DAQmx

NI DAQmx is the driver software you use to communicate with and control your NI data acquisition (DAQ) devices.

NB: NI DAQmx is explicitly for NI hardware, other vendors instrumentation may require specific libraries.

It includes an extensive library of functions and VIs you can call from LabVIEW to program your devices, plus:

- Measurement & Automation eXplorer (MAX)
- DAQ Assistant

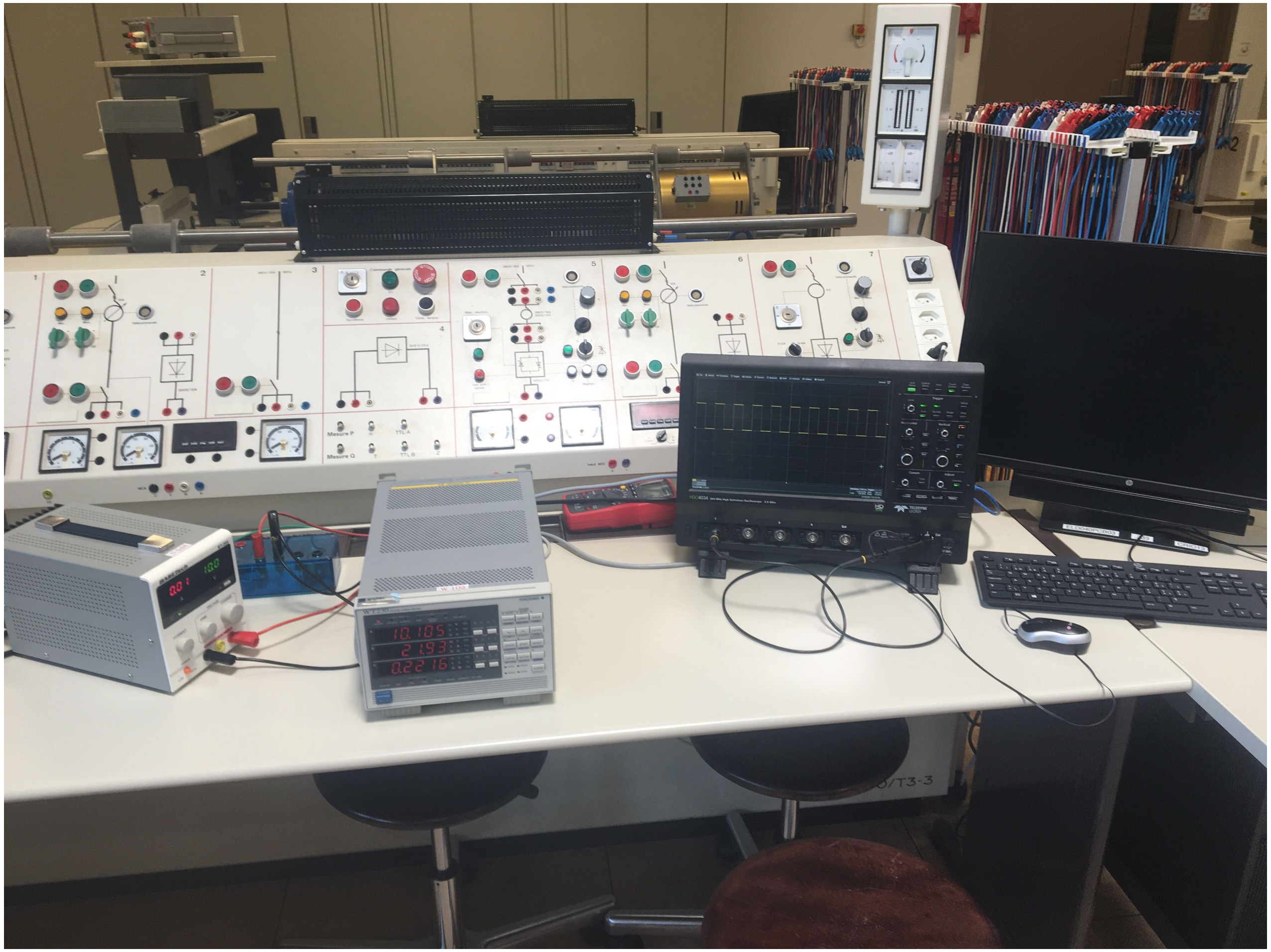
NI MAX

MAX is an application that informs other programs which devices are connected and how they are configured.

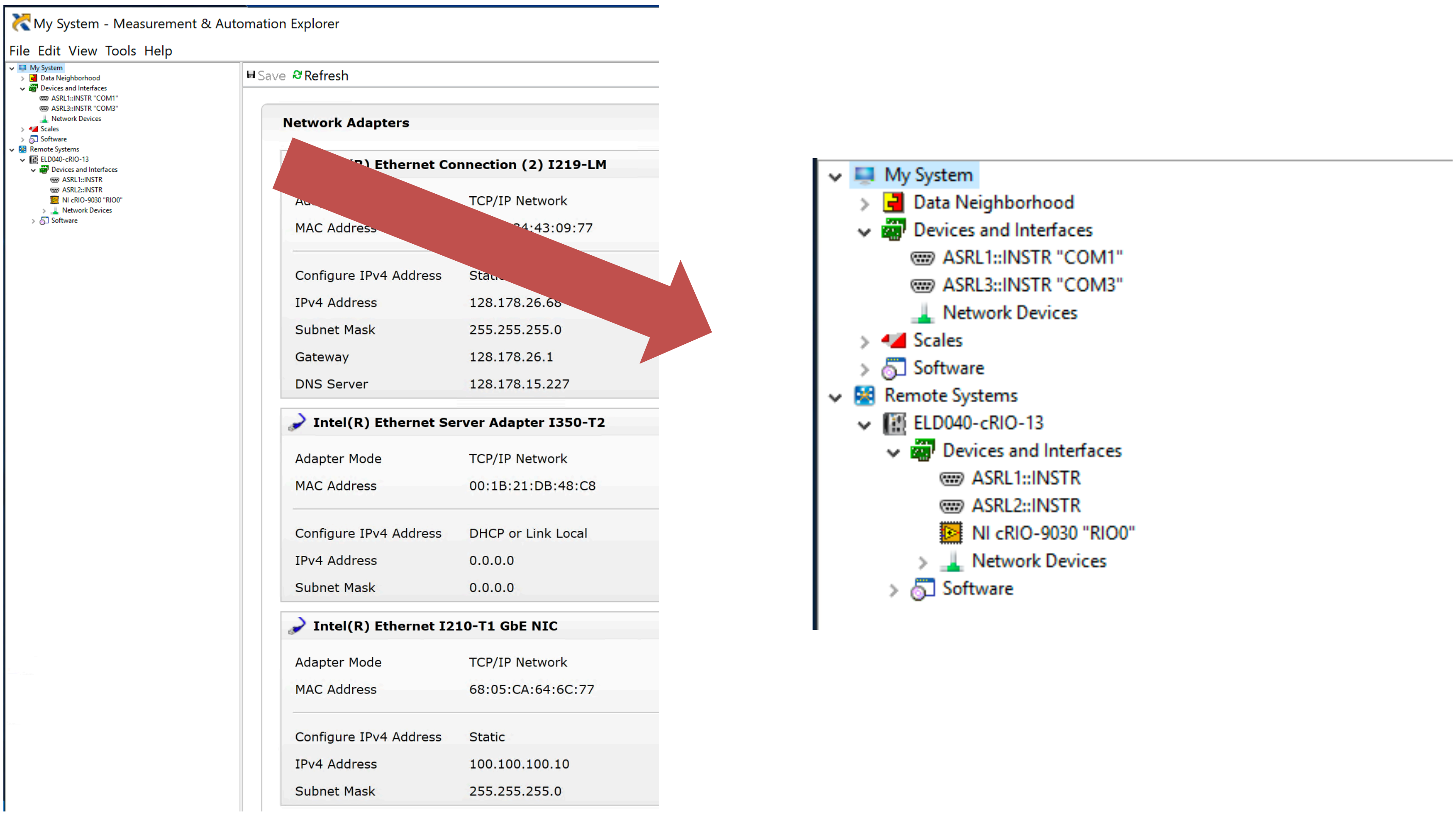
Main functionalities:

- view devices and instruments connected to your system
- configure the NI hardware and software
- create and edit channels, tasks, interfaces, scales, and virtual instruments
- execute system diagnostics
- update your National Instruments software

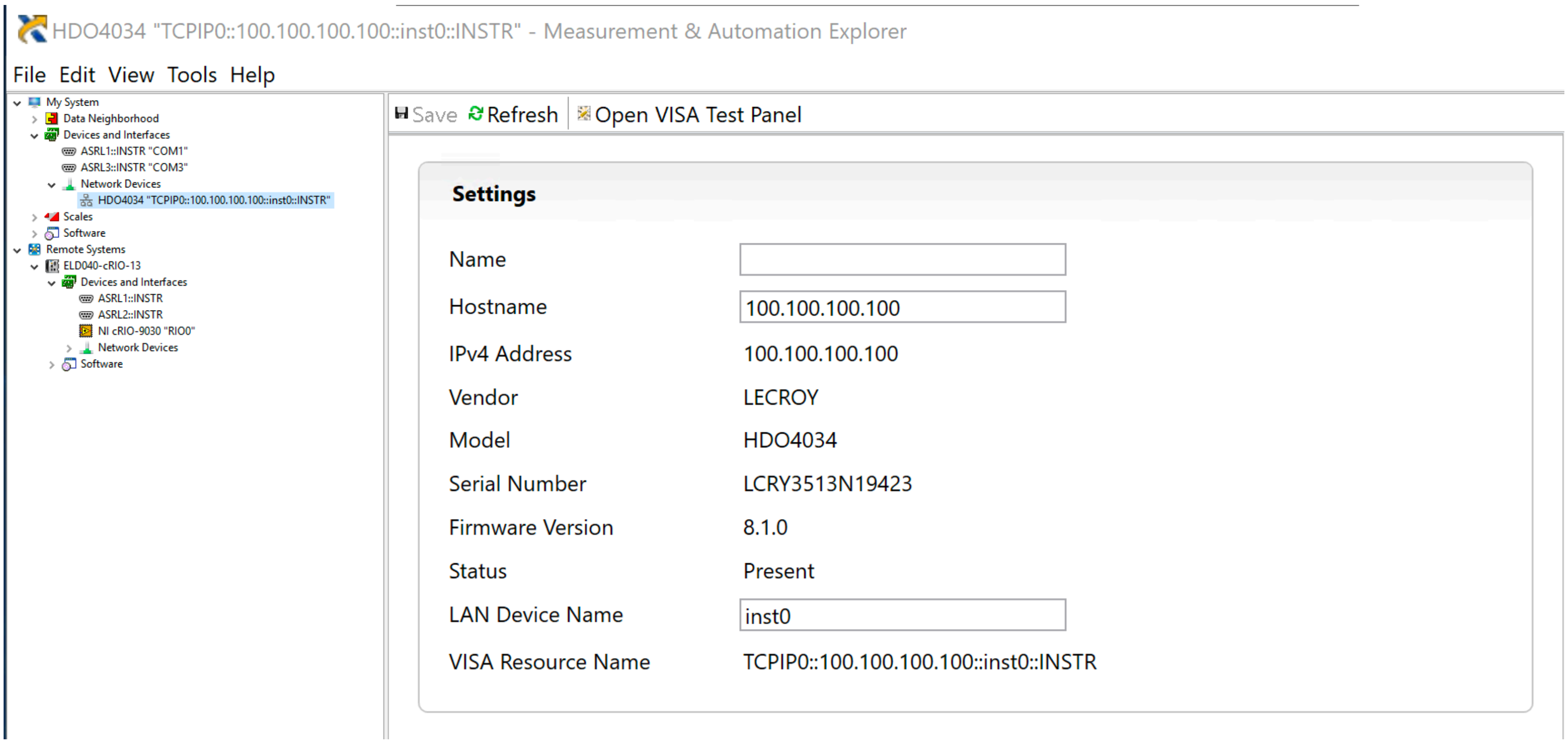
Example



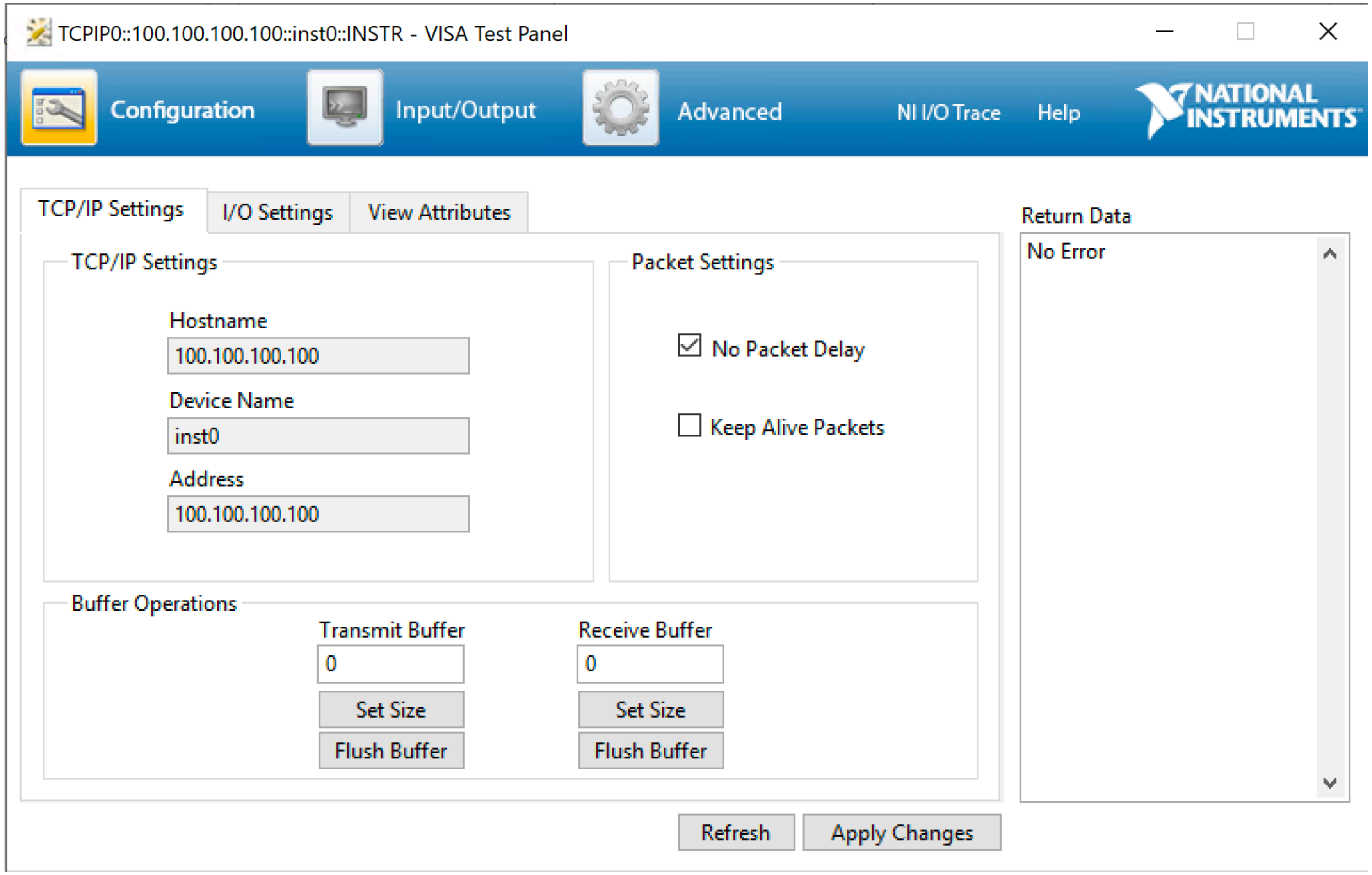
NI MAX console



Network devices



VISA test panel



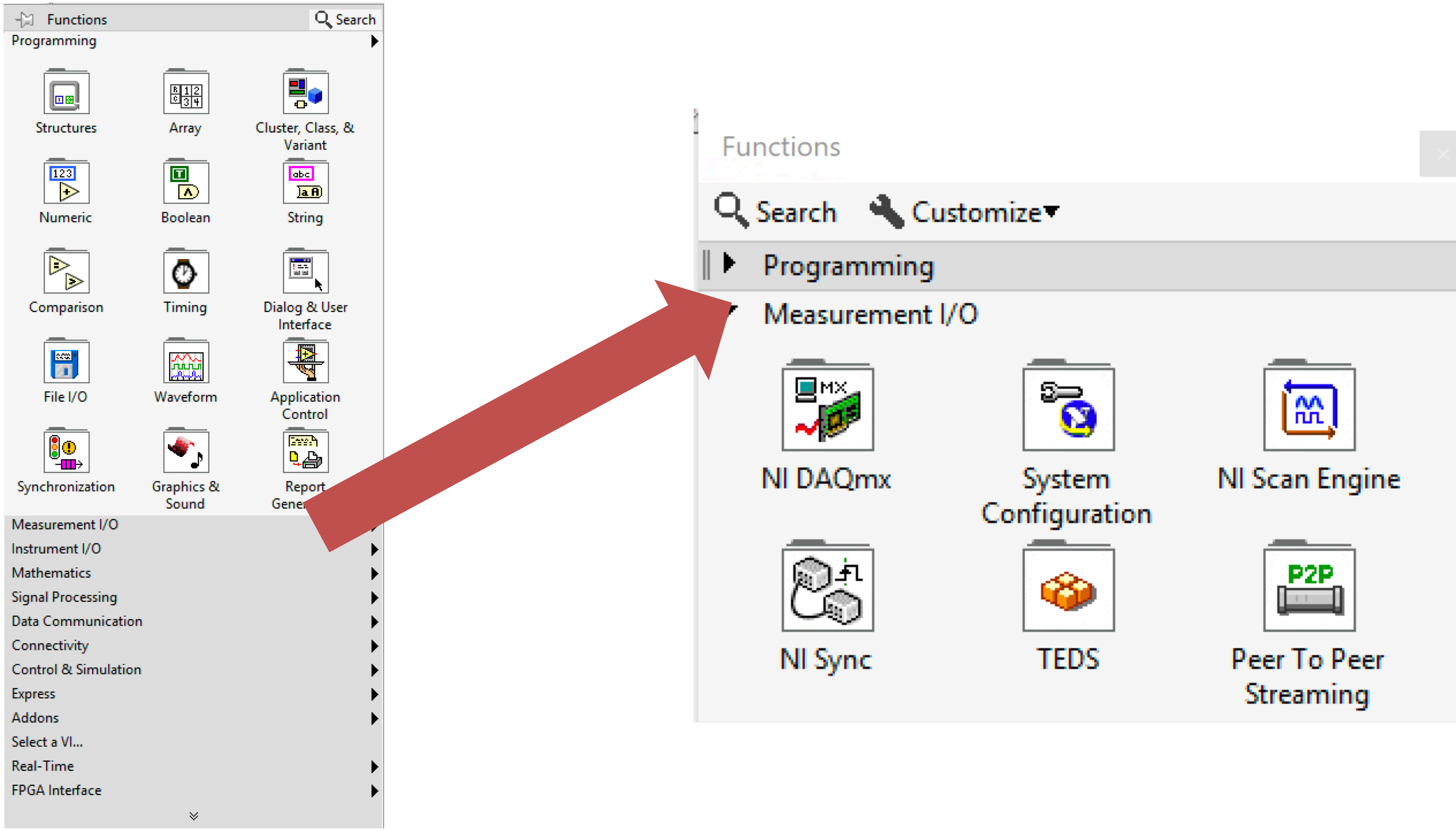
Dedicated software

LabVIEW solution

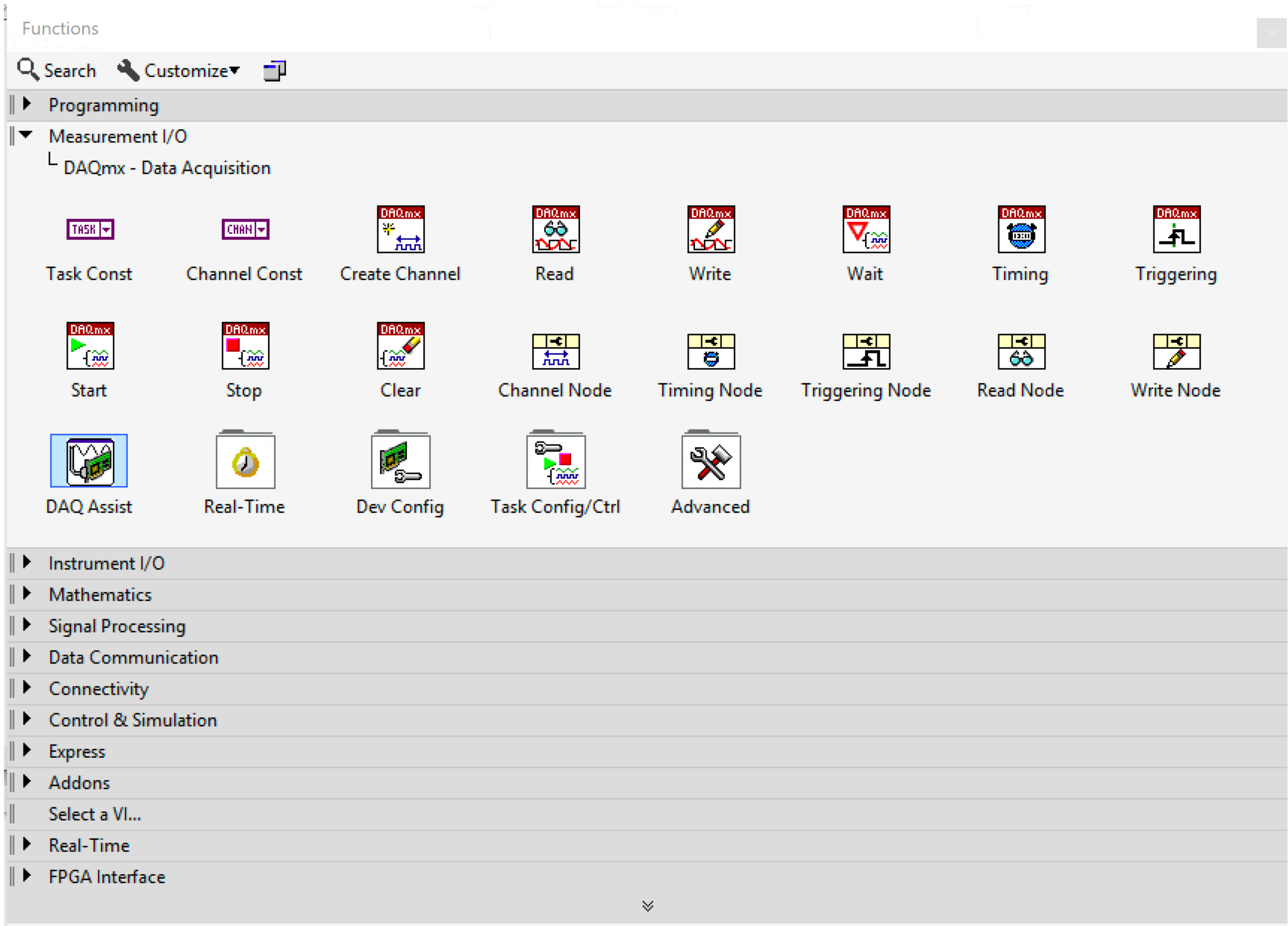
In order to control the instrument from LabVIEW, we need to check two things:

- if the instrument is supported (mostly, just NI hardware) we can use the library NI DAQmx in the palette called Measurement I/O;
- if the instrument is not supported (third part hardware) we have to install the specific library (see next lesson).

Measurement I/O

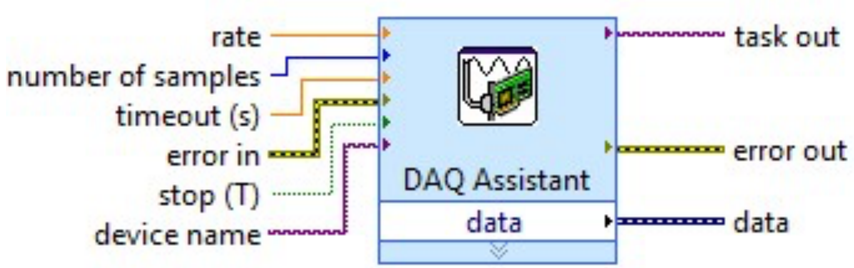


NI DAQmx library

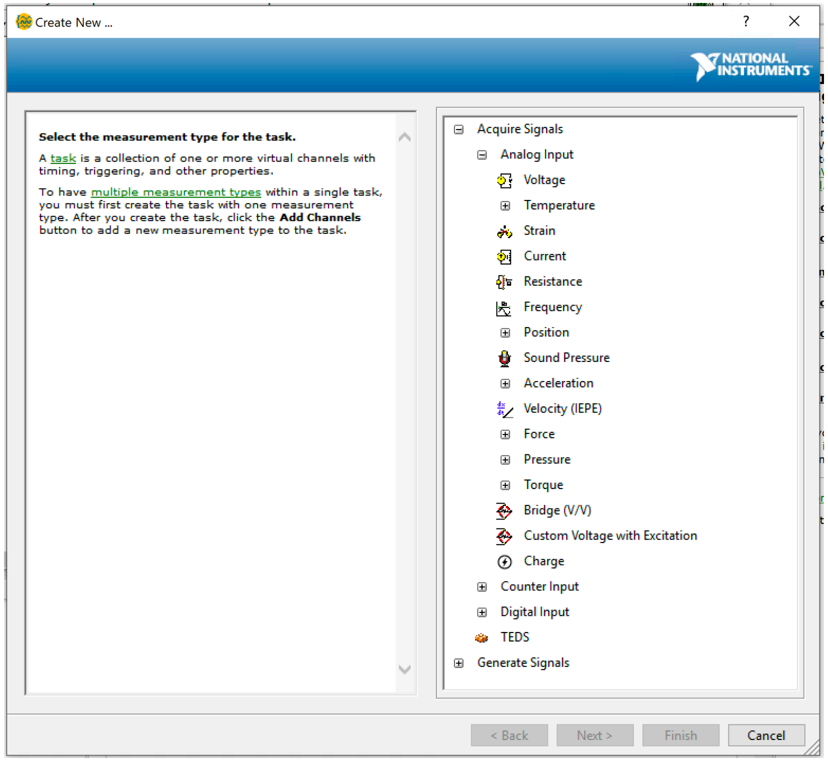


DAQ Assistant

The easiest solution is to use the DAQ assistant that can be easily customized through a multi-step procedure…

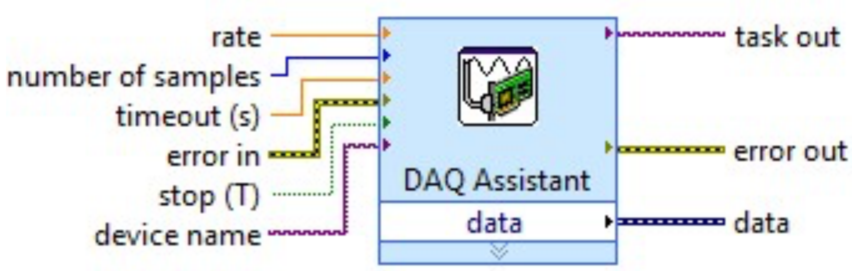


DAQ Assistant

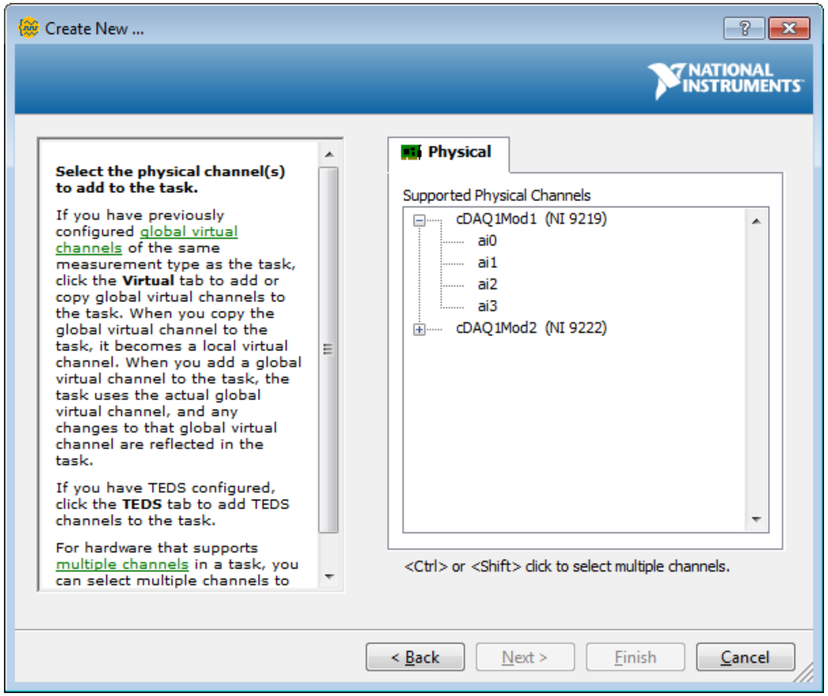


DAQ Assistant

… where Labview asks to select the instrument as well as the physical channels we want to control…

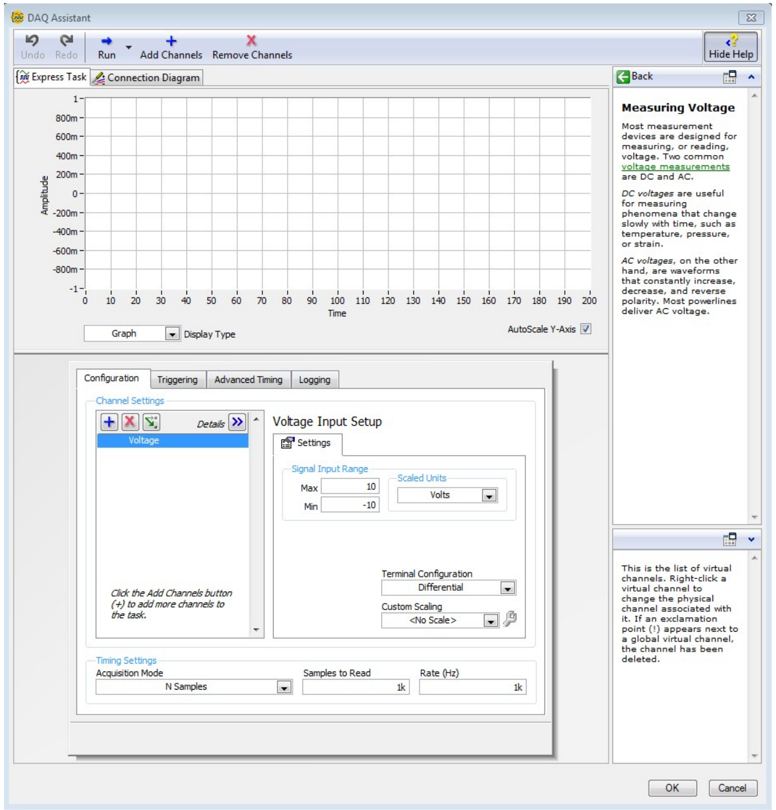


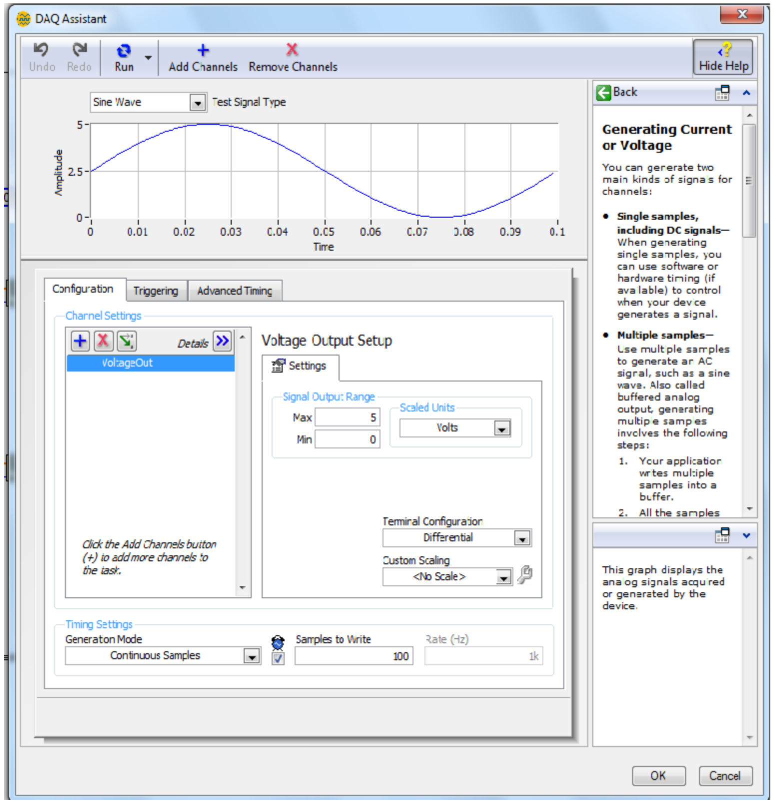
DAQ Assistant



DAQ Assistant

… and, once set up, provides an intuitive GUI:



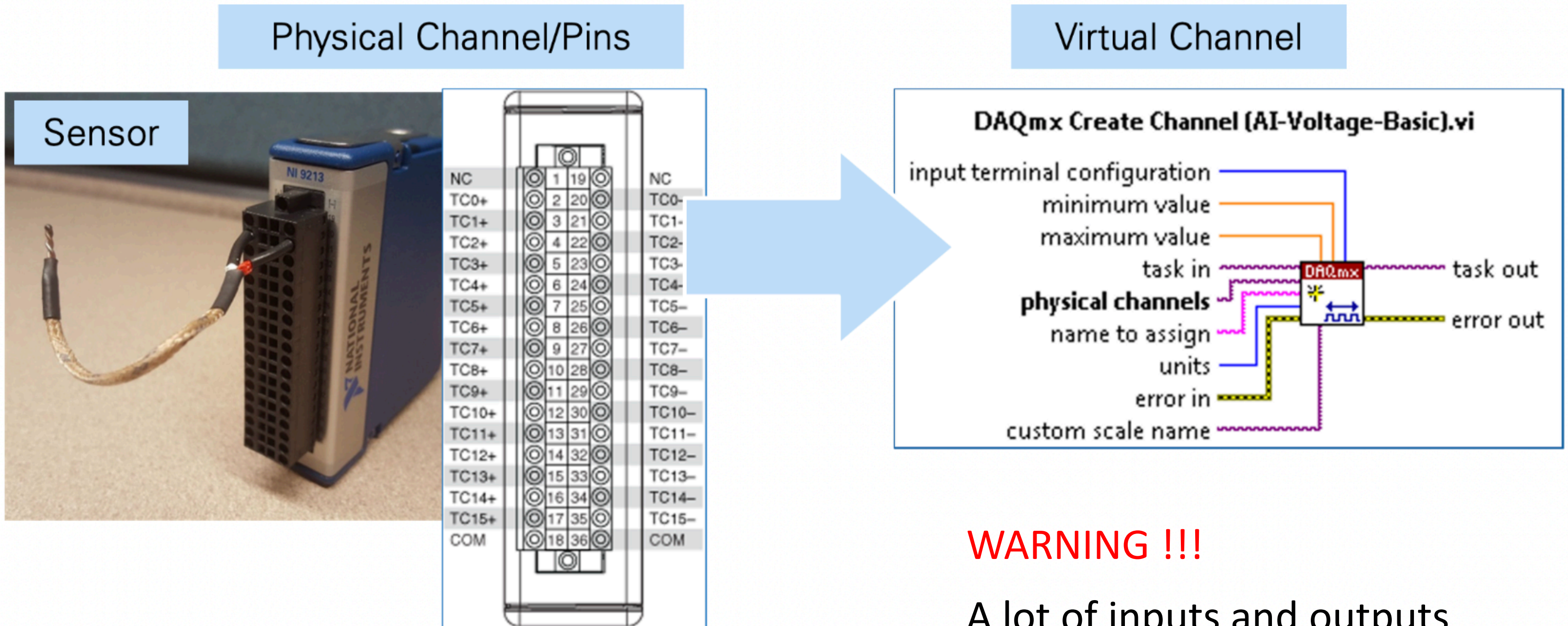


Organization of the code

But – if we don't want or can't use the DAQ assistant – we have to structure our code in a precise and repeatable way:

- we have to convert the instrument physical channels in Labview virtual channels;
- we define which task (experiment) we want to carry out;
- we define the specific parameters of our measurements (e.g. resolution, sampling rate, trigger) that are controlled by specific functions in the palette.

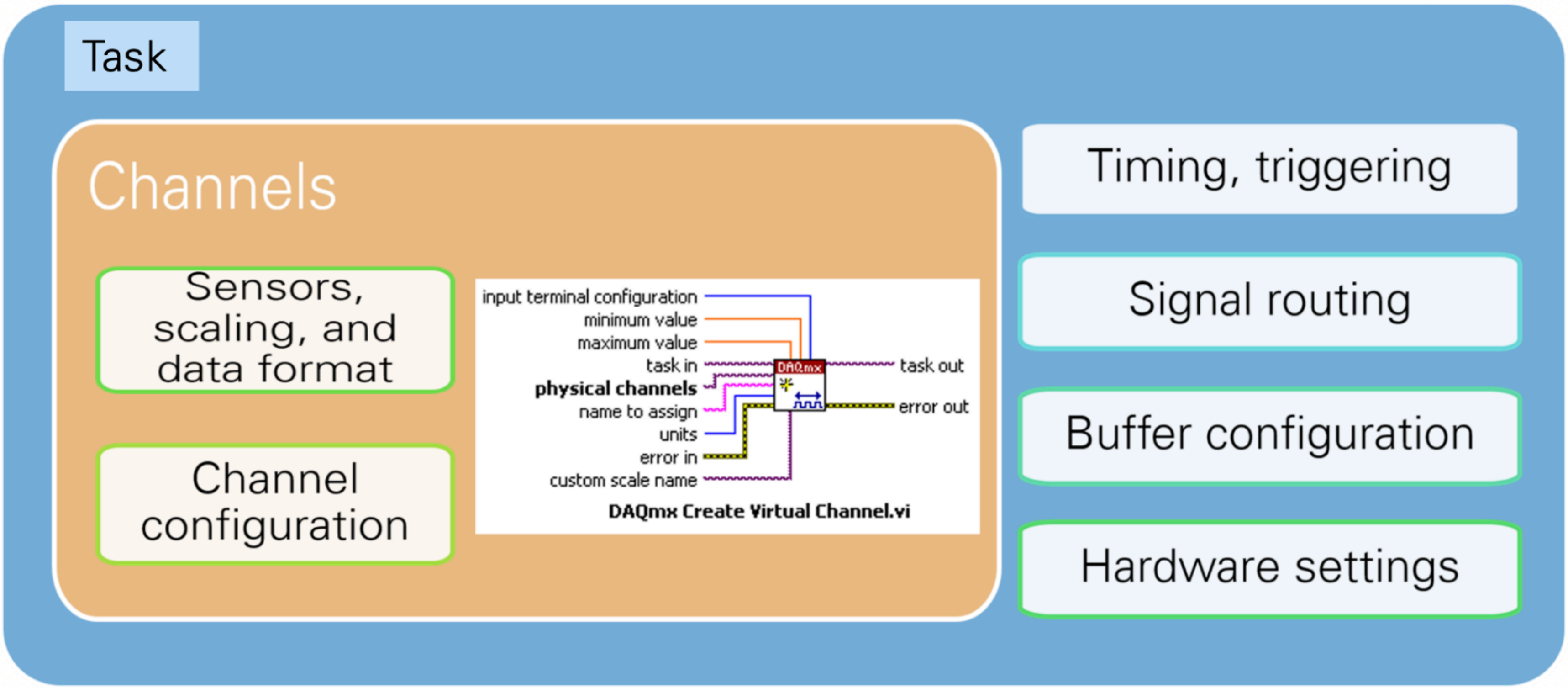
Virtual channel



WARNING !!!

A lot of inputs and outputs comparing to a simple mathematics operation

Tasks and functions



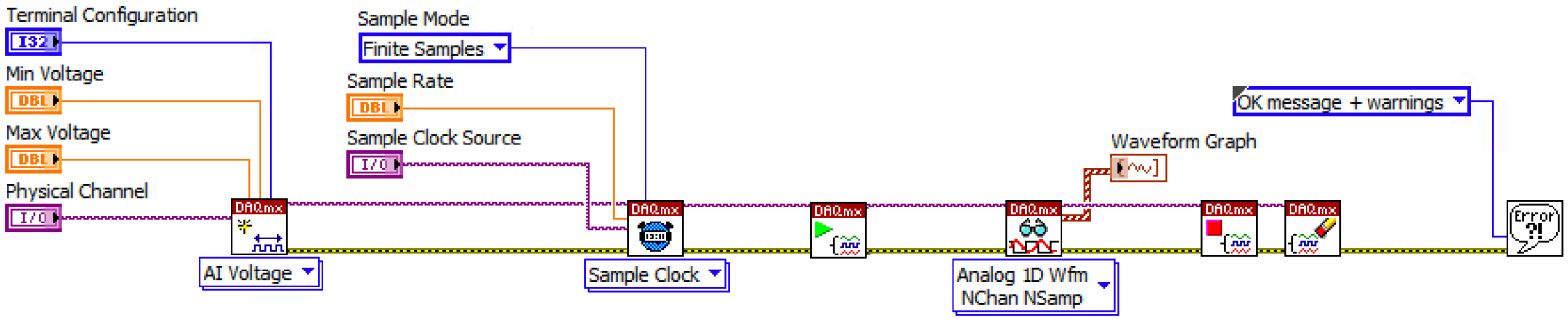
Code organization

We MUST structure the code according to this order:

1. create the task (which kind of measurement)
2. configure the task (set the instrument parameters)
3. start the task (run the experiment)
4. read/write data (acquire or generate data)

   NB: it might be repeated iteratively with a for or a while loop
5. stop the task (end the experiment)
6. clear the task (set the virtual channels free for other tasks)

Coding example



Recap: was everything clear?

- What are the main components of a DAQ system?
- What does NI DAQmx enable us to do?
- Why is NI MAX important?

Teaching lab introduction

- Digital oscilloscope
- Library installation
- Coding example

Digital oscilloscope

A digital oscilloscope is an indispensable tool to solve most measurement challenges quickly and accurately.

The full denomination is digital storage oscilloscope (DSO) as it stores and analyses the signals digitally rather than using analog techniques, with advanced functionalities:

- save point-on-wave data;
- different trigger settings;
- channel-specific scales;
- processing routines (e.g. DFT).

HDO4034a

Teledyne Lecroy HDO4034a (high definition) specs

- 4 input channels (±10 V)
- 350 MHz bandwidth
- 12-bit ADC resolution
- 10 GHz sample rate
- HD4096 12.1” monitor
- 12.5 Mpts per channel
- i3-6100 quad core 3.7 GHz
- 8 GB RAM, Windows 10

HDO4034a

It is provided with several measurement and math tools…

Measurement Tools	
Measurement Functionality	Display up to 8 measurement parameters together with statistics, including mean, minimum, maximum, standard deviation, and total number. Each occurrence of each parameter is measured and added to the statistics table. Histicons provide a fast, dynamic view of parameters and waveshape characteristics. Parameter gates define the location for measurement on the source waveform.
Measurement Parameters - Horizontal + Jitter	Delay (from trigger, 50%), Duty Cycle (50%, @level), Edges (@level), Fall Time (90-10, 20-80), Frequency (50%, @level), Period (50%, @level), Δ Period (@level), Phase (@level), Rise Time (10-90, 20-80), Skew, Time (@level), Δ Time (@level), Width+, Width-
Measurement Parameters - Vertical	Amplitude, Base, Maximum, Mean, Minimum, Peak-to-Peak, RMS, Std. Deviation, Top.
Measurement Parameters - Pulse	Area, Base, Fall Time (90-10, 80-20), Overshoot (positive, negative), Rise Time (10-90, 80-20), Top, Width+, Width-

Math Tools	
Math Functionality	Display up to 2 math functions traces (F1-F2). The easy-to-use graphical interface simplifies setup of up to two operations on each function trace, and function traces can be chained together to perform math-on-math.
Math Operators - Basic Math	Average (summed), Average (continuous), Difference (-), Envelope, Floor, Invert (negate), Product (x), Ratio (/), Reciprocal, Rescale (with units), Roof, Sum (+).
Math Operators - Filters	Enhanced resolution (to 15 bits vertical)
Math Operators - Frequency Analysis	FFT (power spectrum, magnitude), up to full record length. Select from Rectangular, VonHann, Hamming, FlatTop and Blackman Harris windows.
Math Operators - Functions	Absolute value, Derivative, Integral, Invert (negate), Reciprocal, Rescale (with units), Square, Square root, Zoom (identity).

… that makes it not only a waveform display but an actual processor for measurements.

HDO4034a

In terms of connectivity, the feasible options are:

Connectivity	
Ethernet Port	Supports 2 10/100/1000BaseT Ethernet interface (RJ45 ports)
USB Host Ports	4 side USB 3.1 Gen1 ports and 1 front USB 2.0 port support Windows compatible devices
USB Device Port	1 USBTMC port
GPIB Port (optional)	Supports IEEE - 488.2 (External)
External Monitor Port	HDMI 1.4 (Qty. 1) and DisplayPort 1.2 (Qty. 1) to support customer-supplied external monitor. Includes support for extended desktop operation with UHD 3840 x 2160 pixel resolution and split-grid capability on external monitor. Supports touch screen integration of external monitor (Note: external display can not use a Fujitsu touch-screen driver).
Remote Control	Via Windows Automation, or via LeCroy Remote Command Set

… and Labview is not contemplated, so we need to find a way to establish the connection.

Drivers library

https://teledynelecroy.com/support/softwaredownload/labview.aspx

where we need to download:

- VICP Passport plug-in
  NB: Windows only

- NI X-Stream library
  NB: NI-VISA included in your Labview version

X-Stream DSOs

Type: LabVIEW™ driver

Follow this link to National Instrument's page for LabVIEW Plug and Play drivers: NI X-Stream LabVIEW Drivers. NI has developed a "traditional" driver as well as a "project-style" driver for use in LabVIEW 8.0 and above.

Installation instructions can be found in an HTML readme file that is within the ZIP file containing the driver download. Read this first to avoid installation issues. Also contained in the readme file are instructions for finding Example programs.

Designed for:
WaveMaster 8 Zi Oscilloscopes
SDA 8 Zi Oscilloscopes
DDA 8 Zi Oscilloscopes
SDA Oscilloscopes
SDA 7 Zi Oscilloscopes
DDA Oscilloscopes
DDA 7 Zi Oscilloscopes
WaveMaster 8000(a) Oscilloscopes
WavePro 7 Zi Oscilloscopes
WavePro 7000(a) Oscilloscopes
WaveRunner 6000(a) Oscilloscopes
WaveRunner Xi/MXi(-A) Oscilloscopes
HDO6000 Oscilloscopes
HDO4000 Oscilloscopes
WaveSurfer 3000
WaveSurfer Xs/MXs(-A) Oscilloscopes
WaveSurfer 400 Oscilloscopes

Software requirements

- LabVIEW 7.0 or higher
- NI-VISA 3.0 or higher
- Latest Teledyne LeCroy VICP Passport (for VICP connections only; not required for GPIB or LXI)

Drivers installation

The Teledyne LeCroy VICP Passport is a plug-in passport for National Instruments' VISA and is needed if we wish to communicate with the DSO via TCP/IP (ethernet).

VICP Passport: download → install → DONE!

For the Labview instrument drivers, we have to be careful!

They are plug & play drivers but they have to be stored in a specific folder in order to be recognized by Labview.

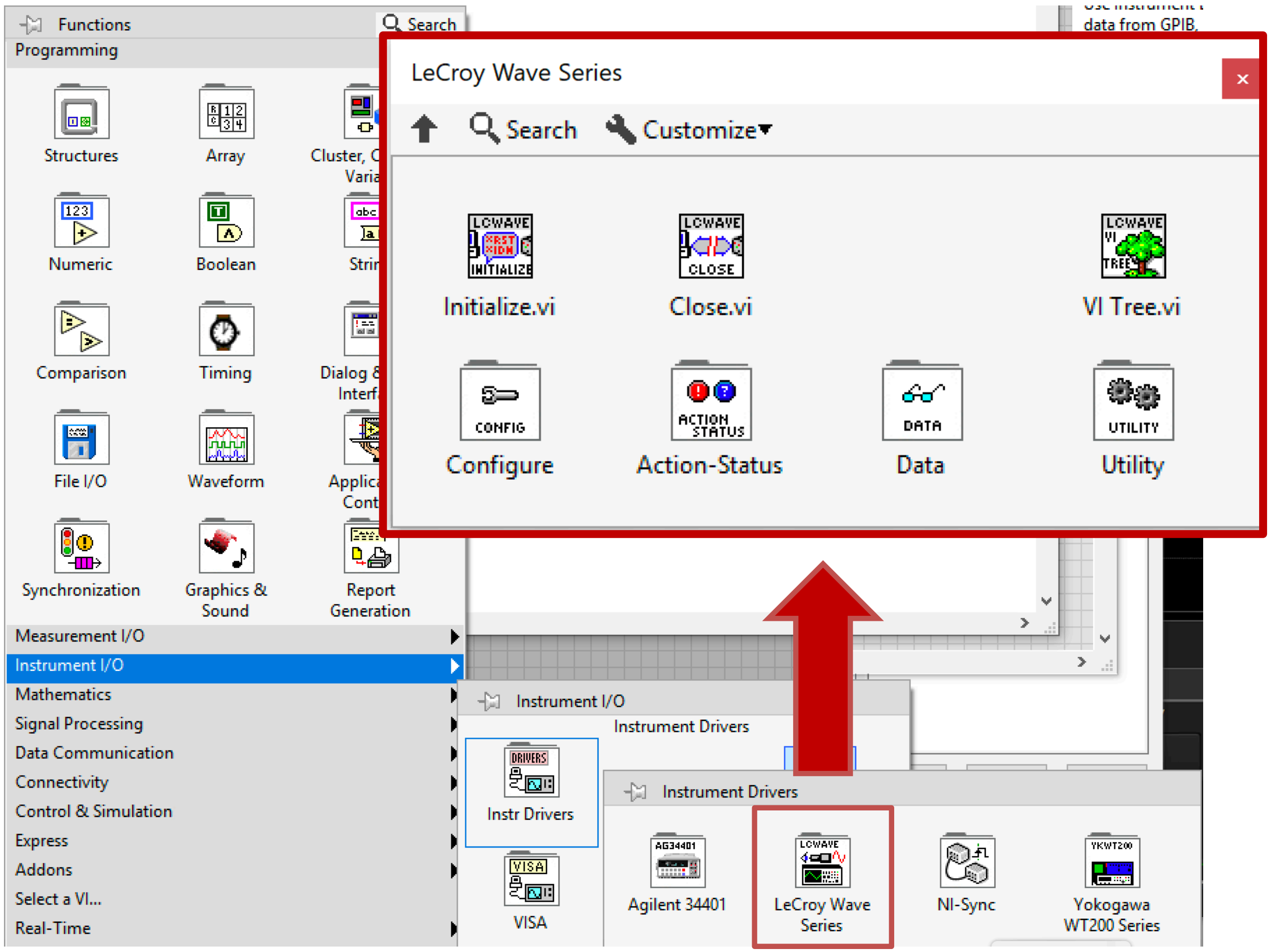
Installation procedure

This procedure is valid for most third part drivers:

1. download the driver package (compressed folder)
2. uncompress the folder (typically, a .zip archive)
3. move the folder to <LabVIEW>\instr.lib directory
4. restart Labview and open a new blank VI

NI X-Stream: download → unzip → move to instr.lib

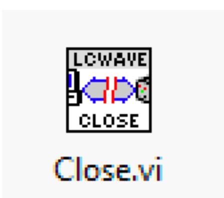
New library


Functions
Search
Programming
Structures
Array
Numeric
Boolean
Comparison
Timing
File I/O
Waveform
Synchronization
Graphics & Sound
Report Generation
Measurement I/O
Instrument I/O
Mathematics
Signal Processing
Data Communication
Connectivity
Control & Simulation
Express
Addons
Select a VI...
Real-Time
LeCroy Wave Series
Search
Customize
Initialize.vi
Close.vi
VI Tree.vi
Configure
Action-Status
Data
Utility
Instrument I/O
Instrument Drivers
Instr Drivers
VISA
Instrument Drivers
Agilent 34401
LeCroy Wave Series
NI-Sync
Yokogawa WT200 Series

Main functions

initialization, i.e. create task

read data, i.e. start task

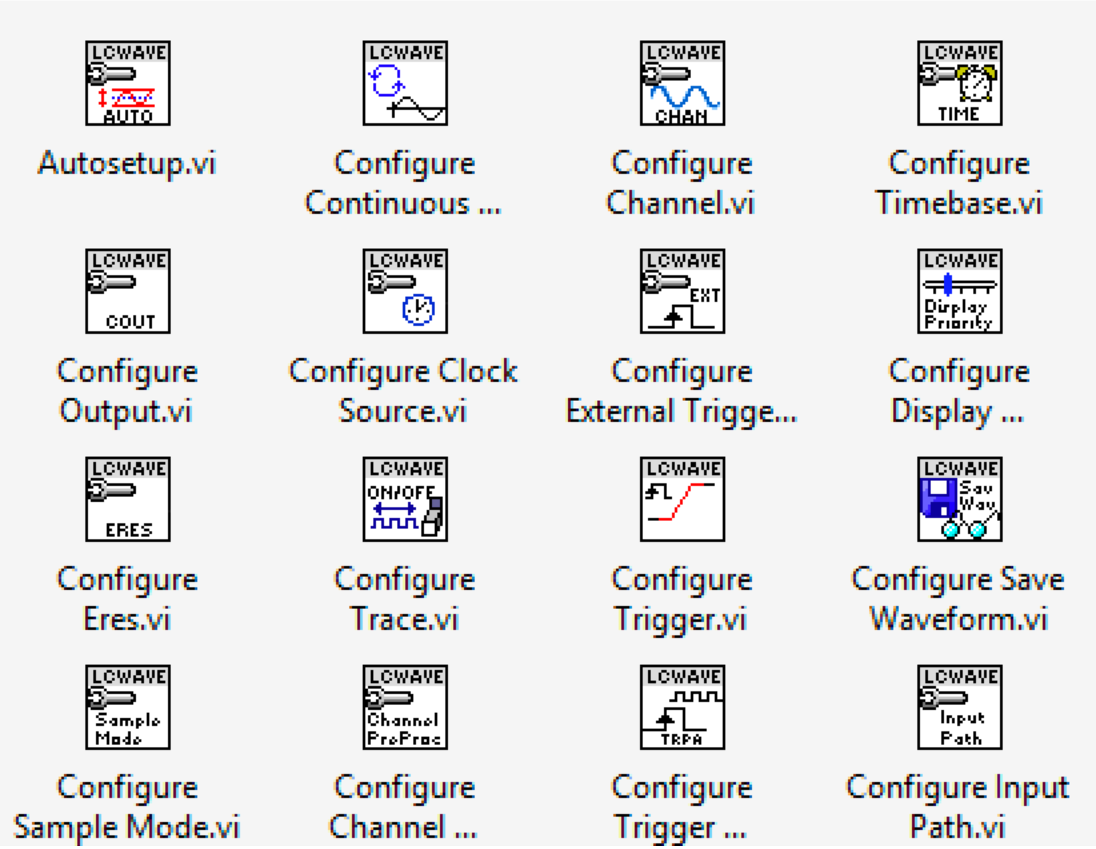


close, i.e. end task

configure, i.e. configure task

Configure menu

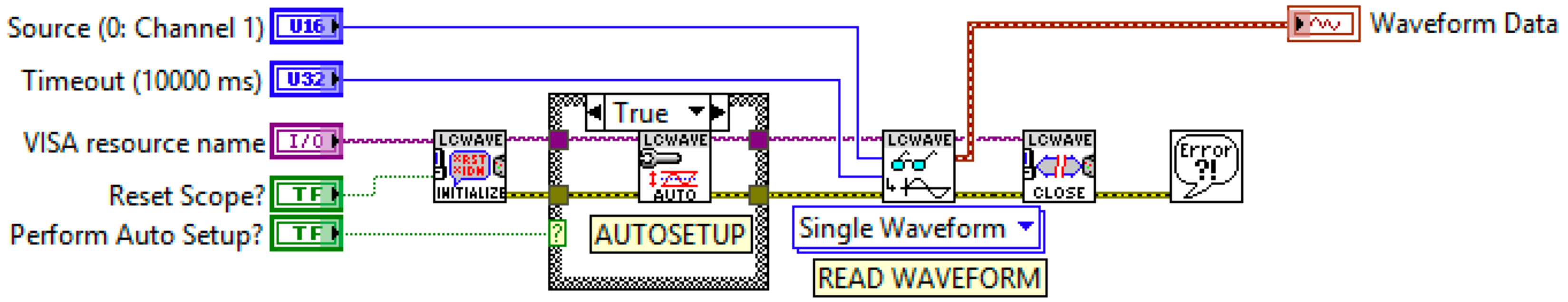
It is possible to configure almost every DSO parameter…



Coding example

Sequence of operations:

- initialization
- configuration
- data acquisition
- task closing



Recap: was everything clear?

- What is a digital oscilloscope, i.e. a DSO?
- Where has a driver library to be stored?
- What is the proper series of commands for the DSO?