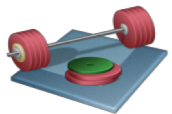
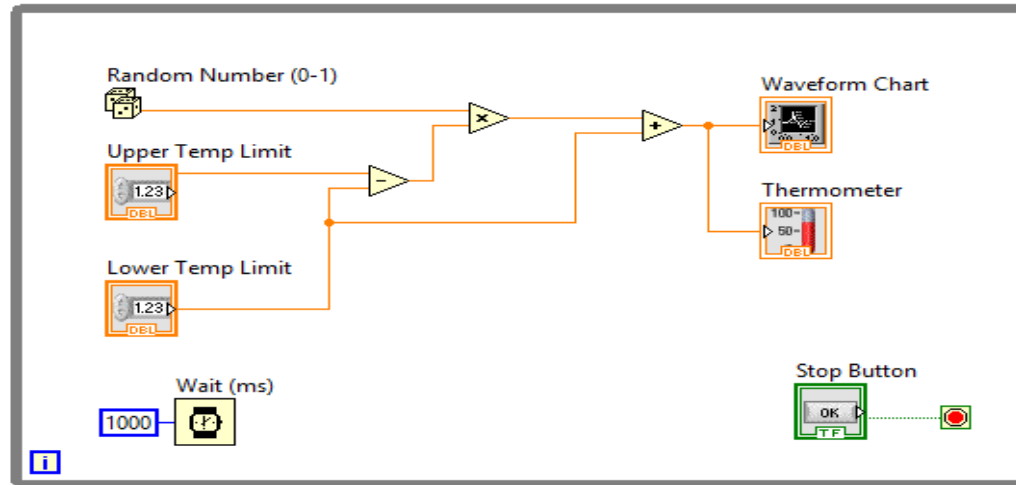




# Introduction to LabVIEW

## Basic LabVIEW Programming



# Contents

- Installation
- What is LabVIEW?
- The LabVIEW Environment
  - Front Panel and Block Diagram
  - Controls and Functions Palette
- Basic LabVIEW Programming
- Plotting
- Creating and using SubVIs
- Tips & Tricks

# LabVIEW Installation

You need the following Software

- **LabVIEW** (LabVIEW Professional Development System 32-Bit: English)
- NI-DAQmx (Hardware Driver for NI USB-6008, NI TC-01, etc.)
- **LabVIEW Control Design and Simulation Module**
- **LabVIEW MathScript RT Module**

**Note!** These packages are separate downloads!

All LabVIEW Software can be downloaded here: [www.ni.com/download](http://www.ni.com/download)

# High-Level Design Tools

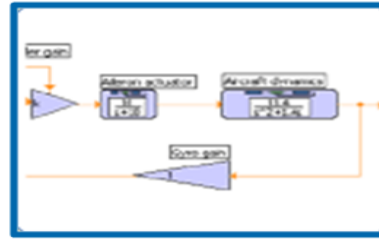
## Configuration



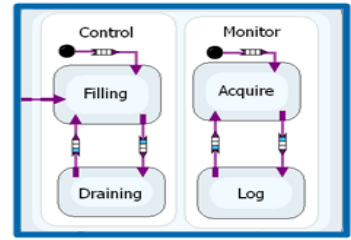
## Textual Math

```
1 c = 0.285 + 0.013i;  
2 [X Y] = meshgrid(x, y);  
3 z = X + i*Y;  
4 for k=1:30  
5     z = z.^2 + c;  
6 end
```

## Simulation



## Statechart



National Instruments is the vendor of LabVIEW

# LabVIEW

Graphical Programming

National Instruments creates both **Hardware** and **Software**

Linux®



Macintosh



Windows

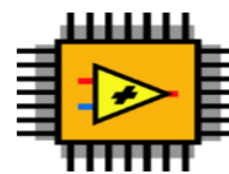


Desktop Platform

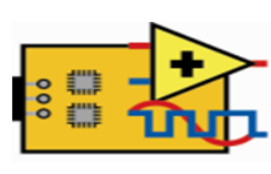
Real-Time



FPGA



MPU



Embedded Platform

# LabVIEW = Fun!

## Graphical Programming:

- Very different from traditional programming like VB, C#, Maple, MATLAB, MathScript, etc.
- It is more like a “drawing program” than a Programming Language
- This makes it easy to use for those who are not programmers (or dont like programming 😊)
- Excellent tool when using Hardware, when you need to take Measurements (DAQ), etc.
- It is fun and makes you very creative!

# LabVIEW Example

LabVIEW has the same things as other programming languages, but in a graphical way!

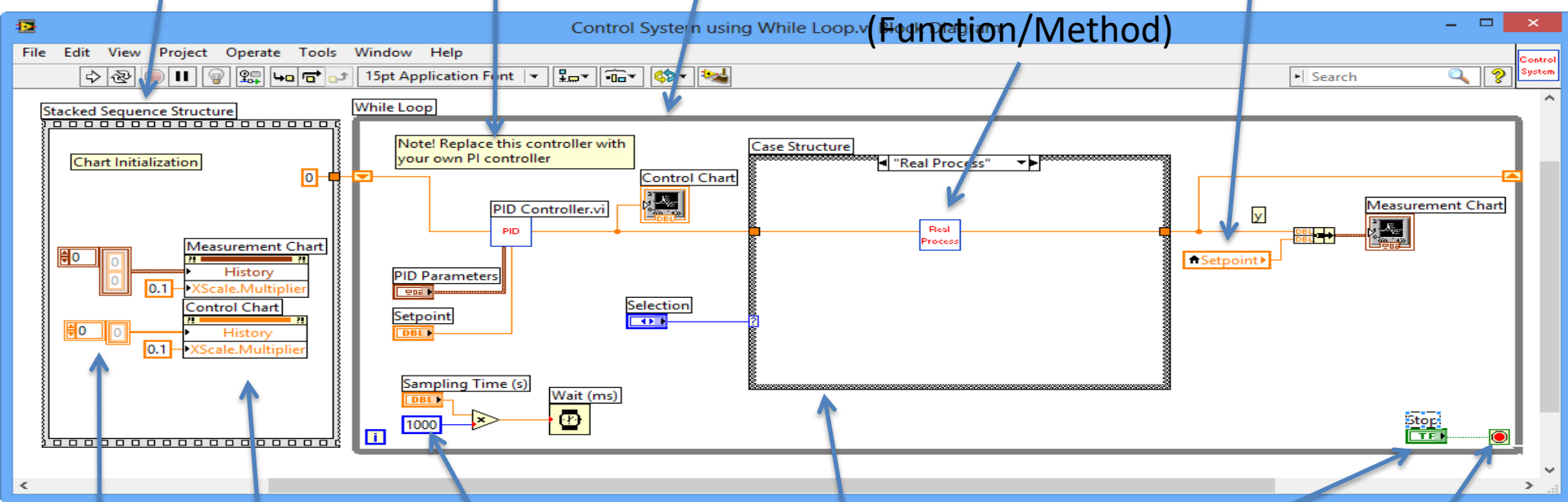
Sequence Structure

Comment

While Loop

Sub VI

Local Variable



(Function/Method)

Arrays

Property Nodes

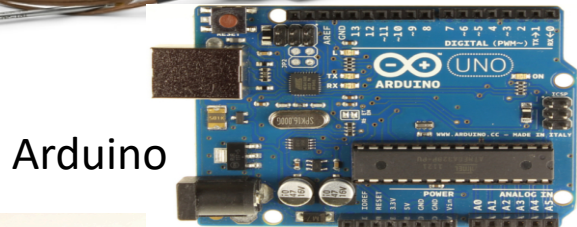
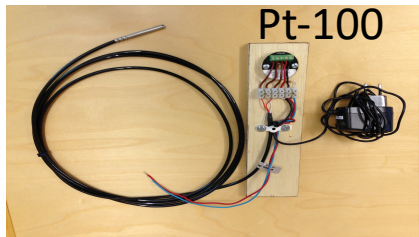
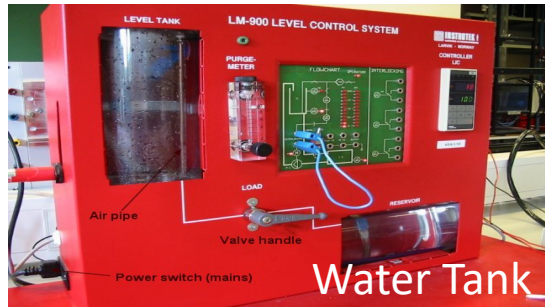
Constants

Case Structure Stop Button  
(if-else)

Condition

**Note!** To do something with an object – Right-click on it (When shall the loop end?)

# Hardware

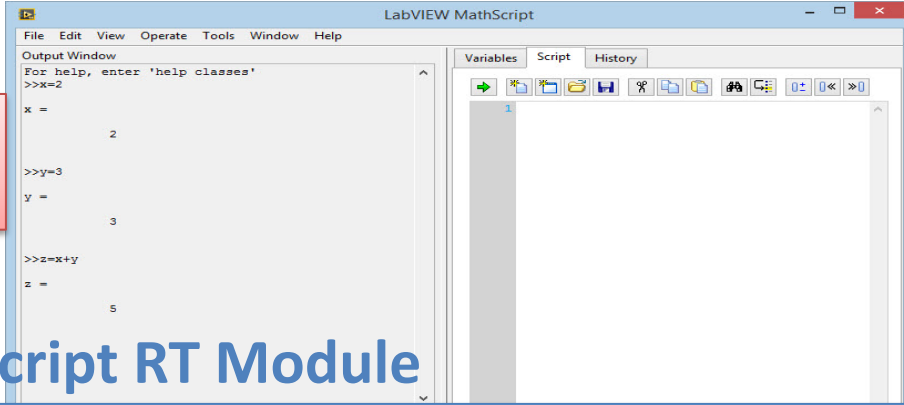
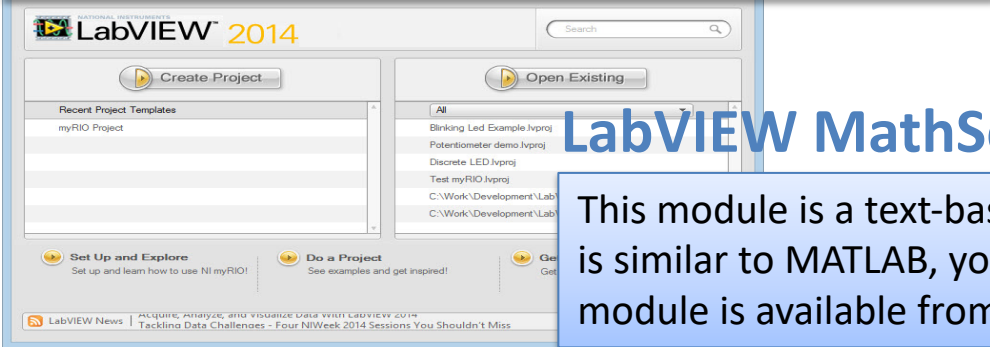


NOx Sensor



# LabVIEW

This is the core LabVIEW installation that installs the LabVIEW Programming Environment.

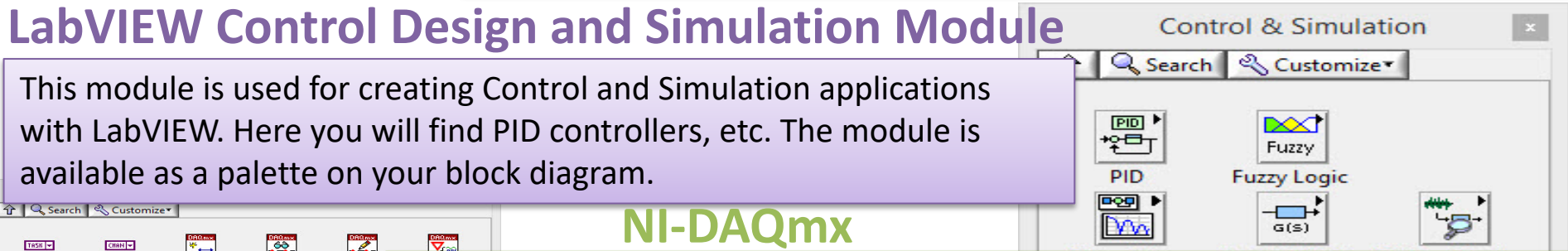


## LabVIEW MathScript RT Module

This module is a text-based tool that is very similar to MATLAB. The syntax is similar to MATLAB, you can create and run so-called m files, etc. The module is available from the Tools menu inside LabVIEW.

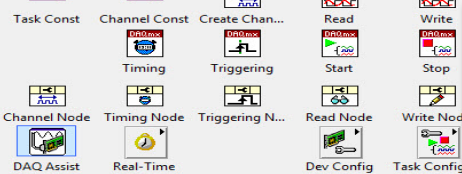
## LabVIEW Control Design and Simulation Module

This module is used for creating Control and Simulation applications with LabVIEW. Here you will find PID controllers, etc. The module is available as a palette on your block diagram.



## NI-DAQmx

DAQmx is the Hardware Driver needed in order to use hardware devices like NI USB-6008, NI TC-01, etc. inside LabVIEW. The module is available as a palette on your block diagram.







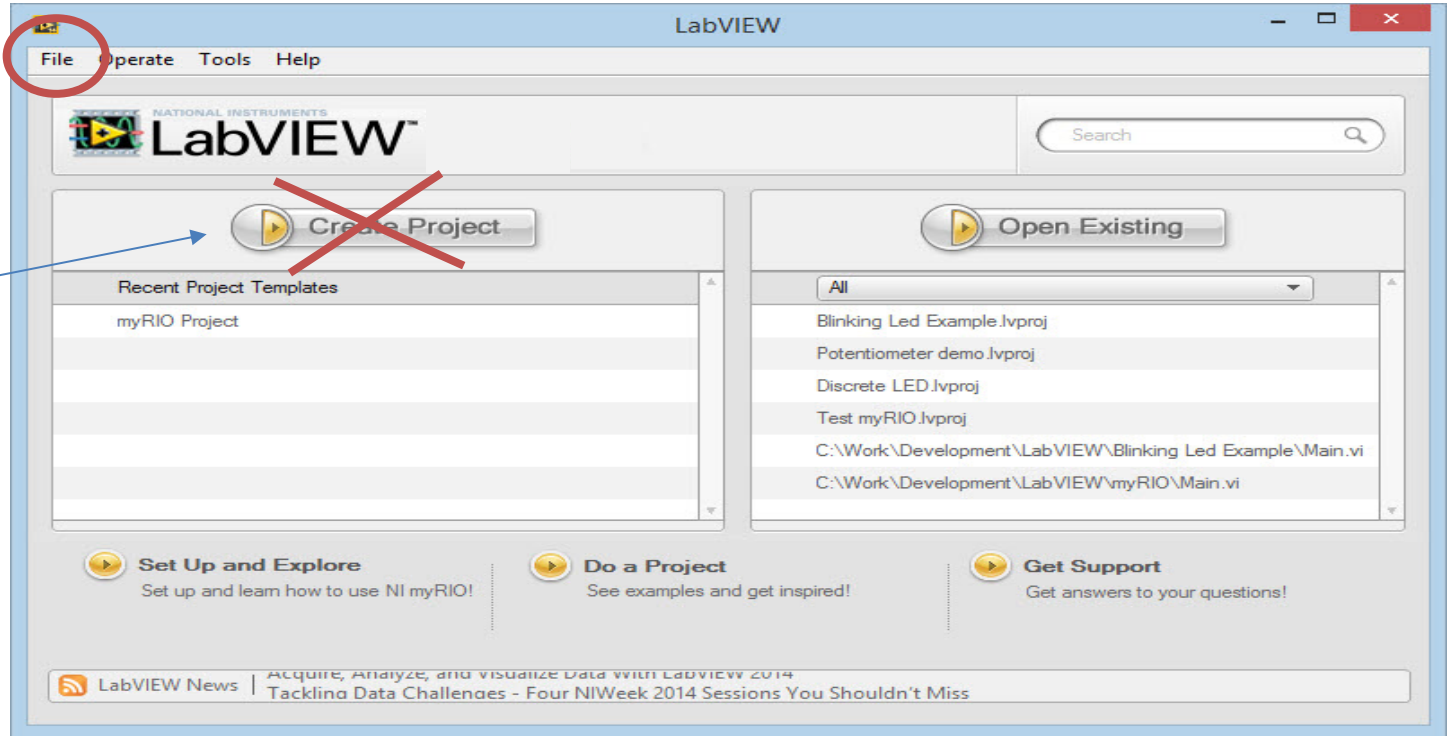
# The LabVIEW Environment (IDE)

Hans-Petter Halvorsen

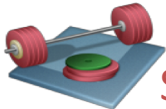
# LabVIEW Environment

This window appears when you start LabVIEW:

Select File -> New VI (Ctrl + N) in order to start using LabVIEW



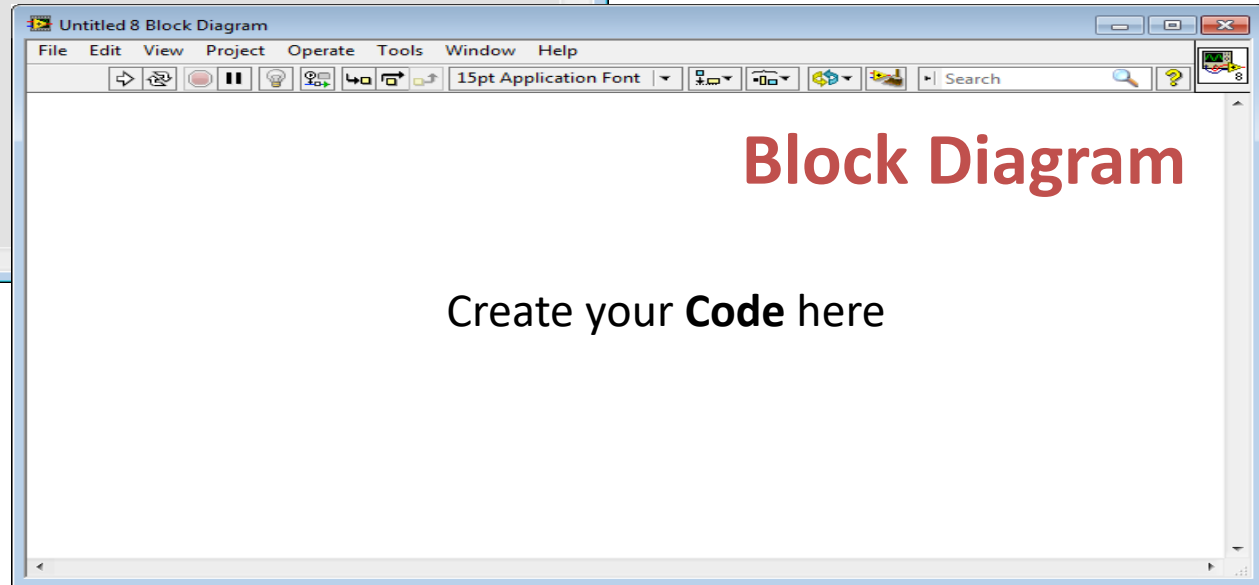
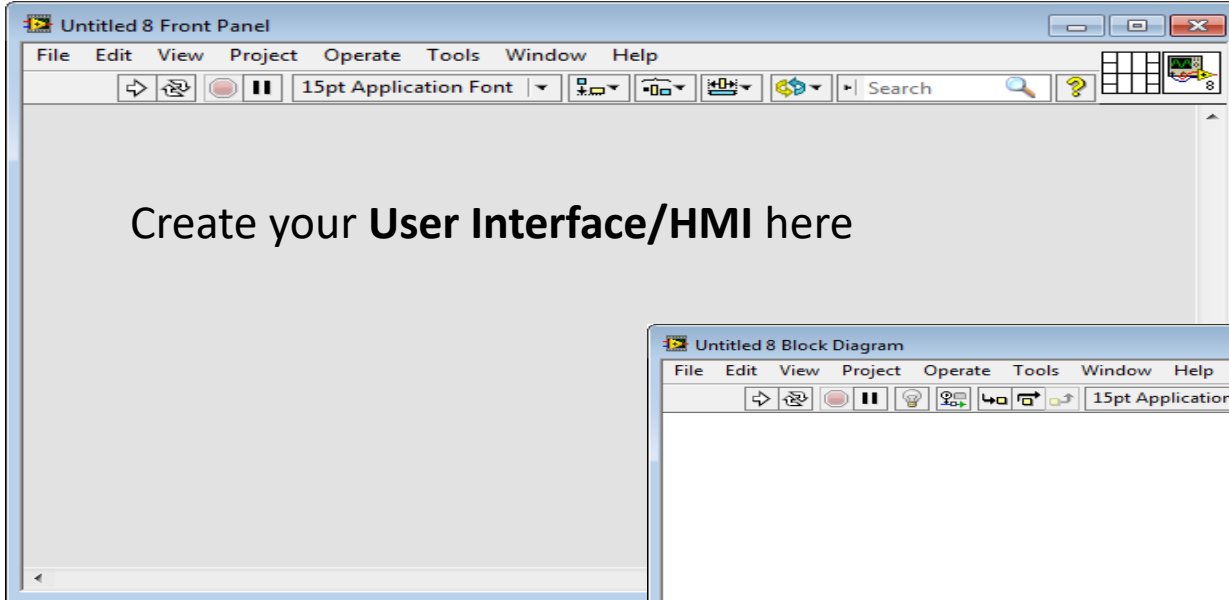
You shall **not** use this one (yet)



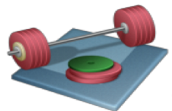
Students: Start LabVIEW in order to get started

# Front Panel LabVIEW Environment

**Note!** Both the Front Panel and the Block Diagram are stored in one single file. These files are called **VIs** (because the file extension is “.vi”). VI = Virtual Instruments



Switch between them: **Ctrl + E**

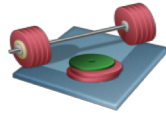
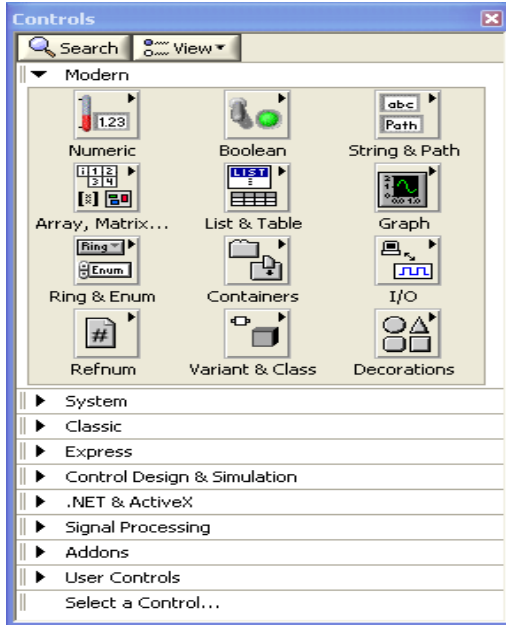
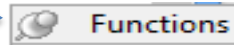
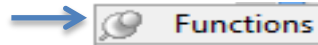


Students: Try this

# Controls and Functions Palette



You can "pin" them!

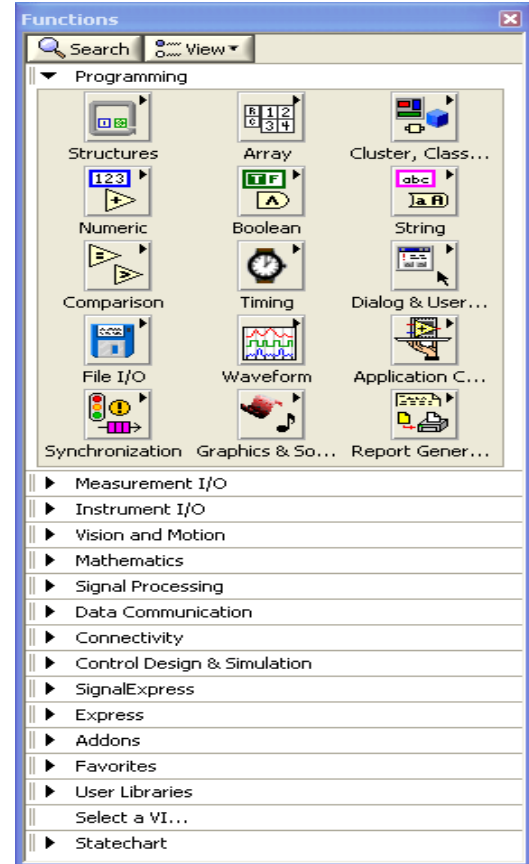


Students: Open the **Controls** and **Functions** palettes and browse the contents in the different subpalettes

Available only from the **Front Panel**

You create your User Interface with help of these Controls

Right-click on the Front Panel

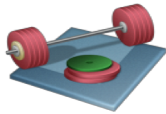
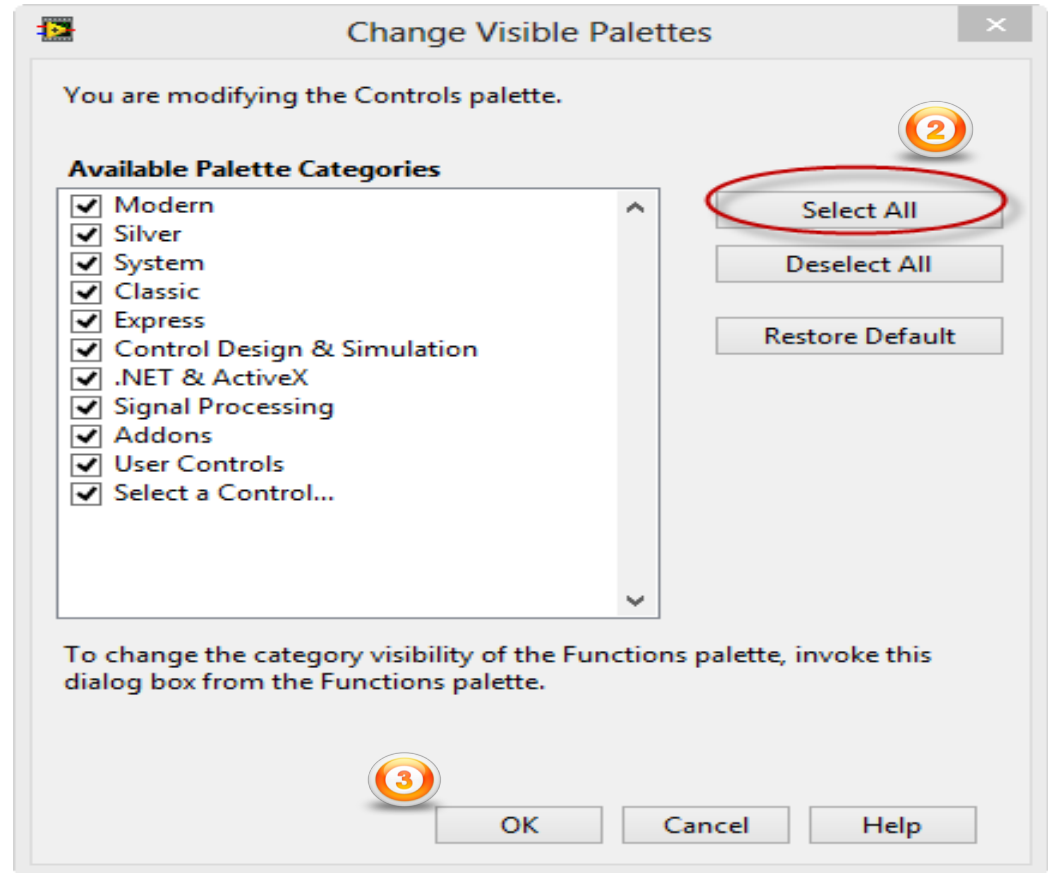
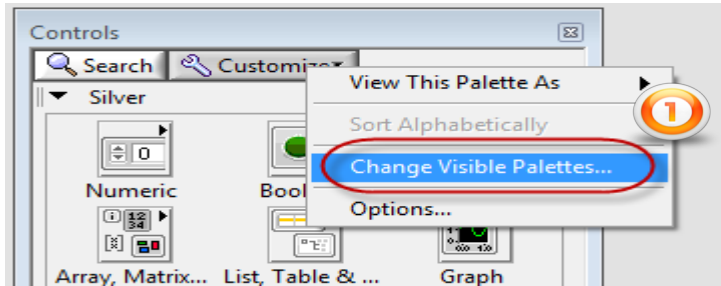


You create your Code with help of these Functions

Available only from the **Block Diagram**

Right-click on the Block Diagram

# Customizing Controls and Functions Palettes



Students: Do this for both the **Controls Palette** and the **Functions Palette**

<https://www.halvorsen.blog>



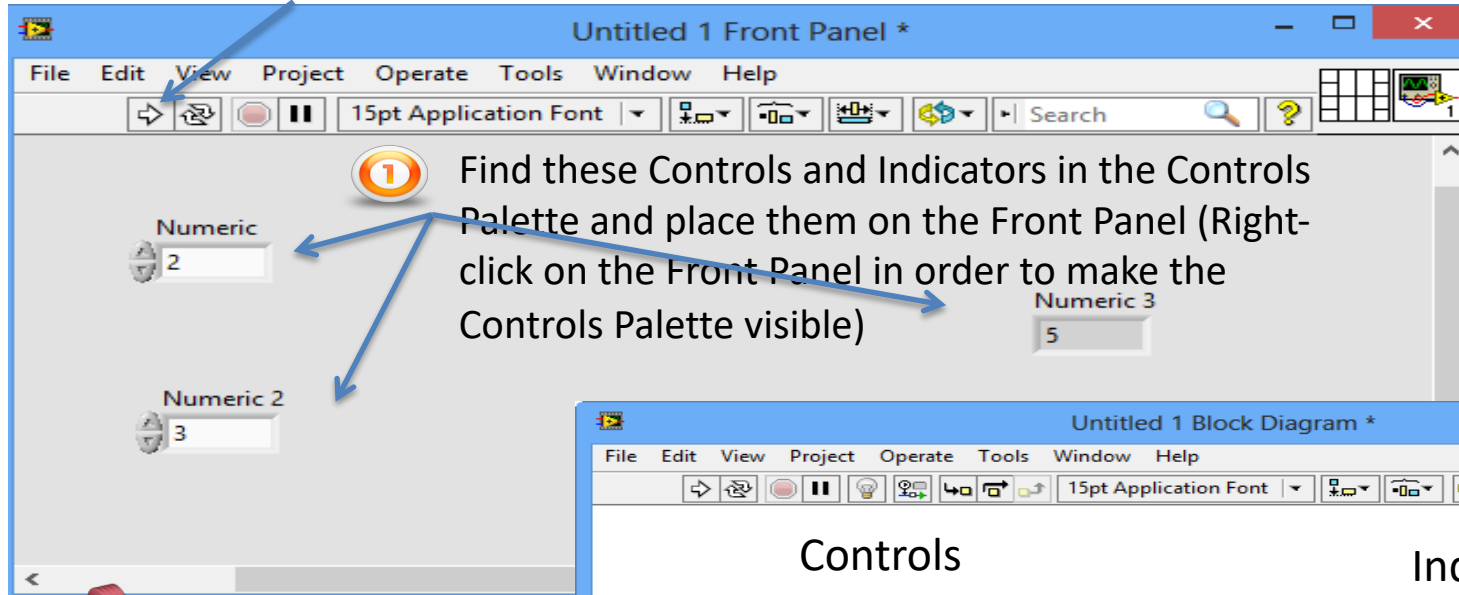
# LabVIEW Programming

Hans-Petter Halvorsen

# Simple Example

3 Run the Program

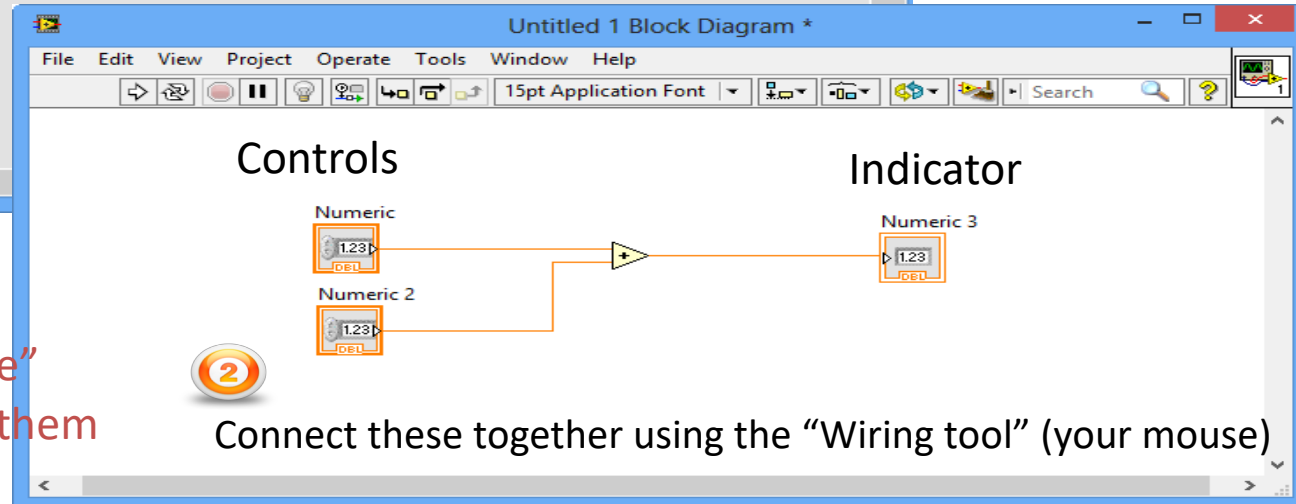
## Front Panel



Controls  
vs.  
Indicators

What is the difference?

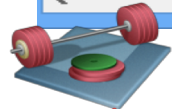
## Block Diagram

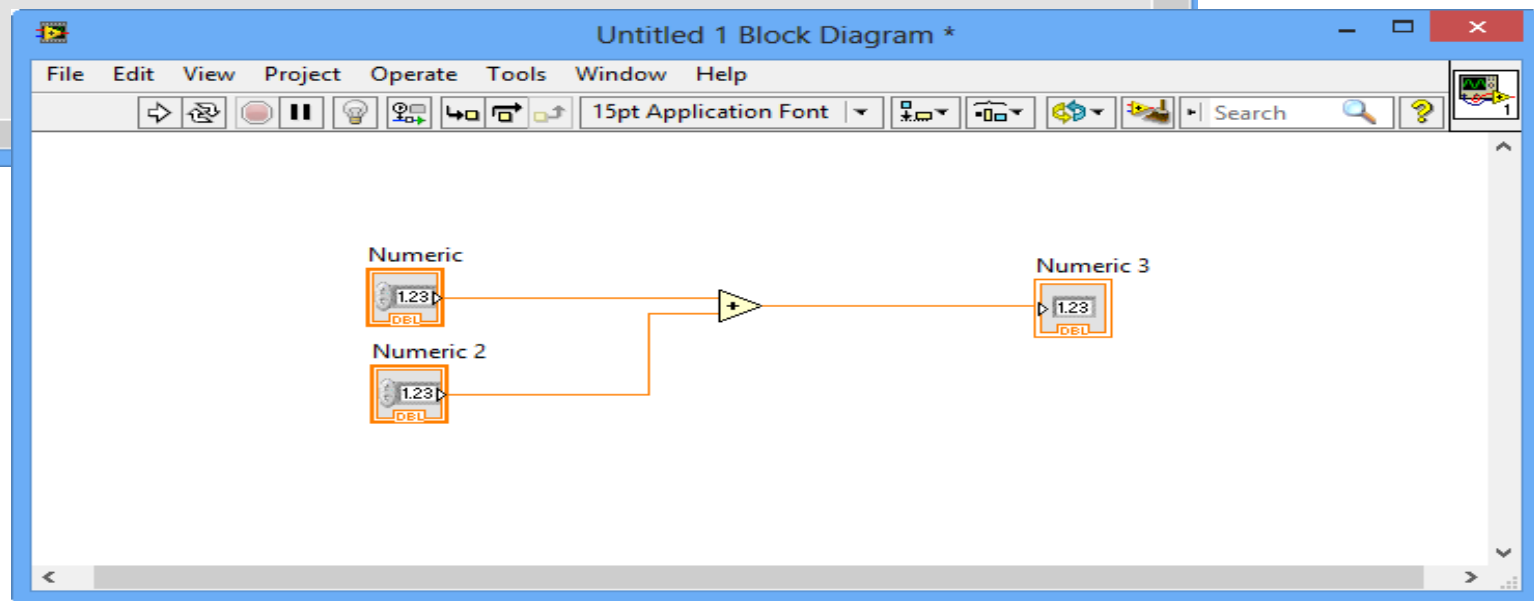
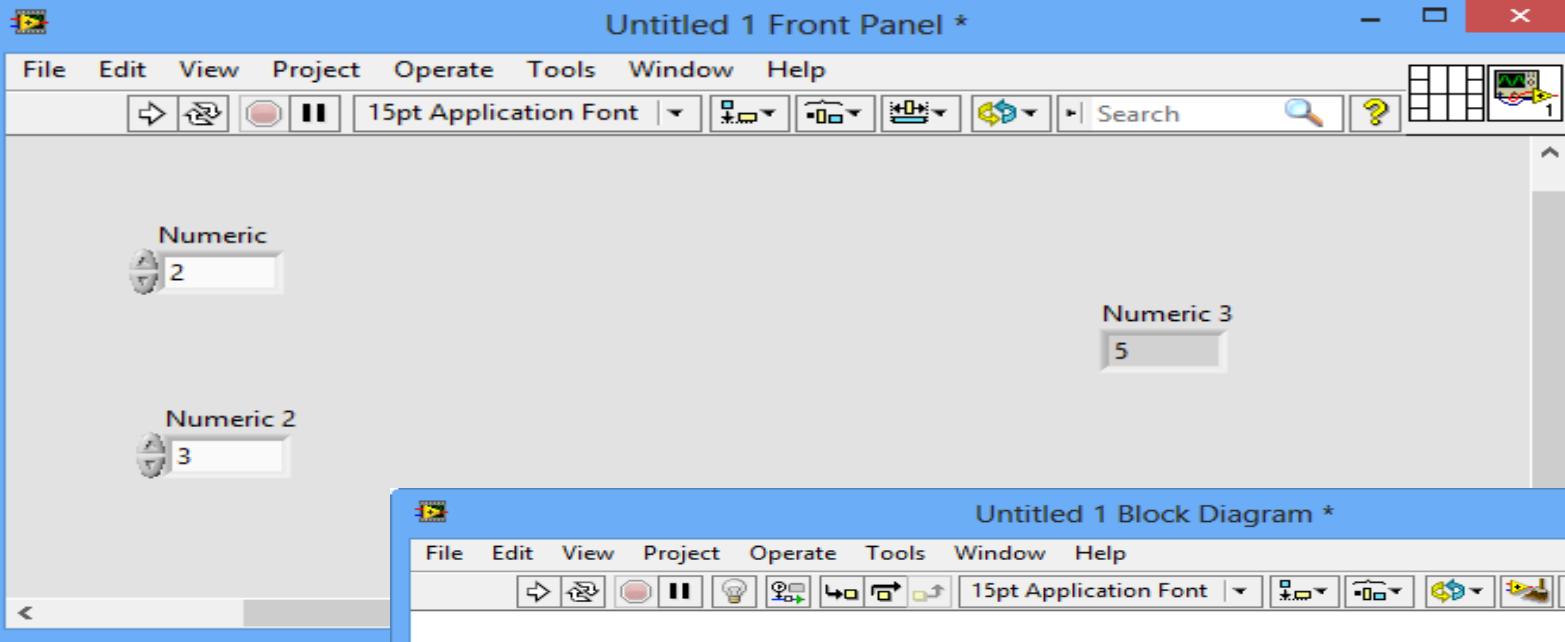


2

Connect these together using the "Wiring tool" (your mouse)

Students: Create this simple example. Try "Minus", "Divide" and "Multiply" as well. Save them as 4 different VIs

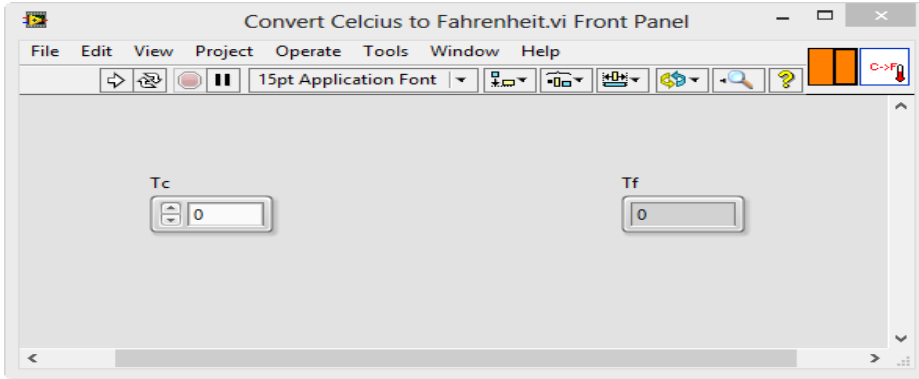






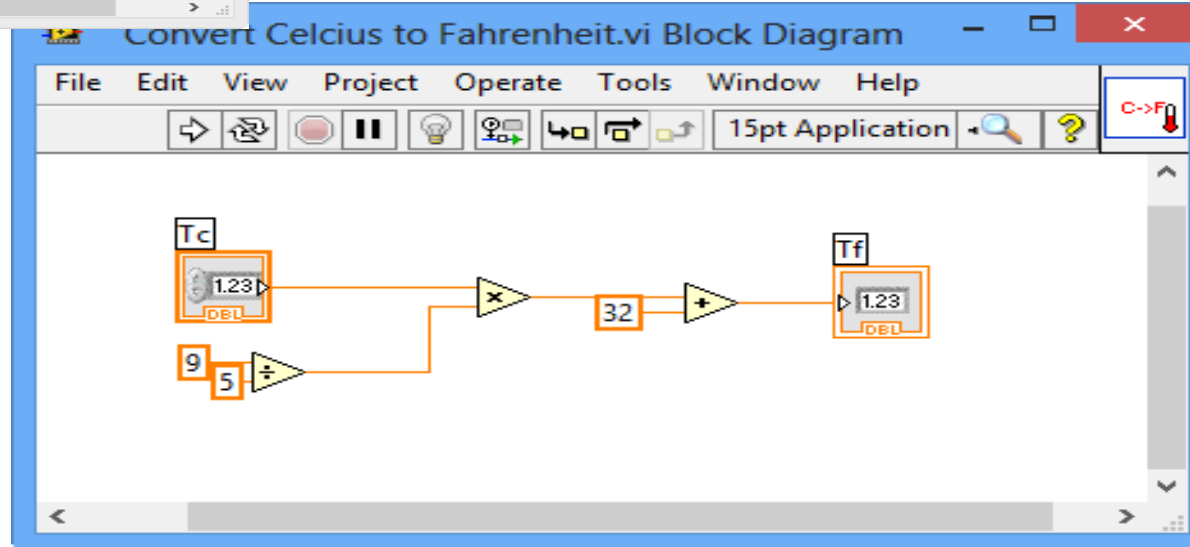
# Celcius to Fahrenheit - Example

## Front Panel

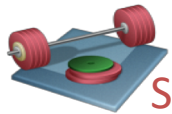


$$T_F = \frac{9}{5}T_C + 32$$

## Block Diagram



Note! The objects may look different depending on your configuration, but that doesn't matter



Students: Create the Front Panel and Block Diagram in this example.

# Convert Celcius to Fahrenheit.vi Front Panel

File Edit View Project Operate Tools Window Help

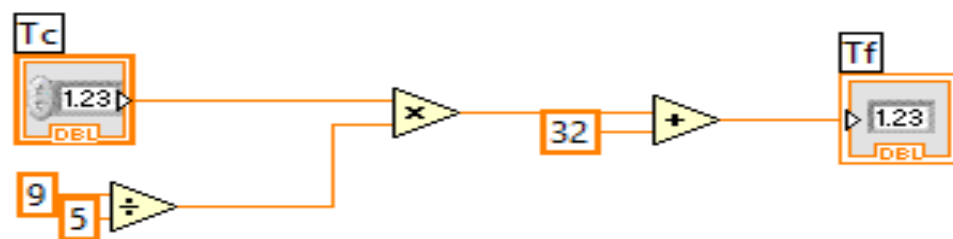


Tc

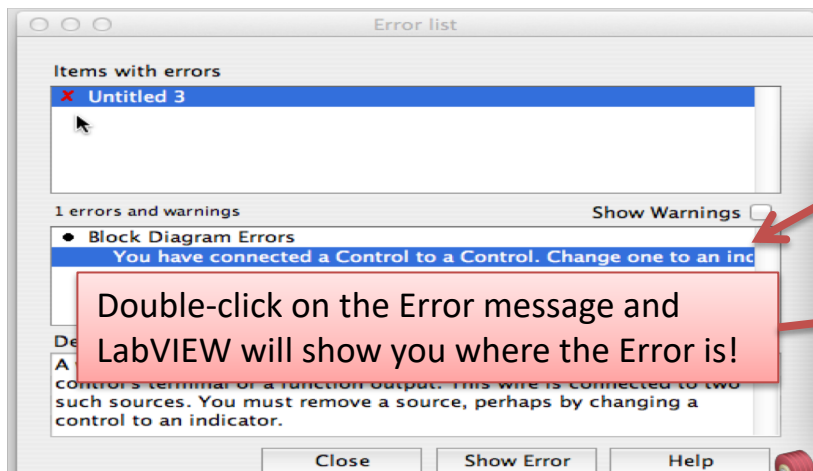
Tf

# Convert Celcius to Fahrenheit.vi Block Diagram

File Edit View Project Operate Tools Window Help



# How-To Fix Errors/Bugs



Items with errors

- Untitled 3

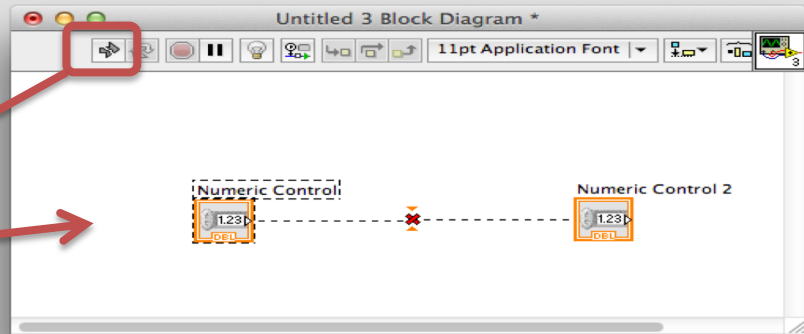
1 errors and warnings

- Block Diagram Errors

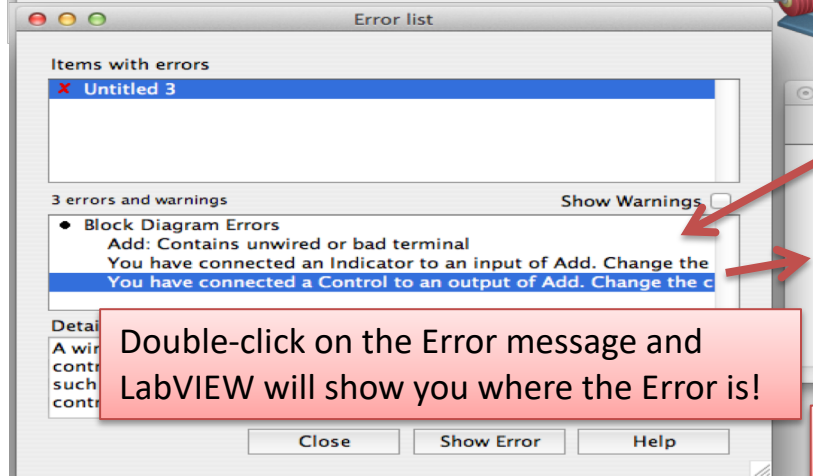
You have connected a Control to a Control. Change one to an indicator.

Close Show Error Help

Double-click on the Error message and LabVIEW will show you where the Error is!



Students: What is wrong in these 2 Examples? Try to create the same bugs as shown here.



Items with errors

- Untitled 3

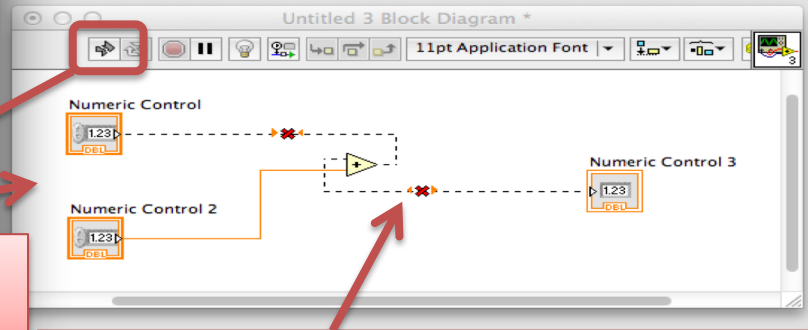
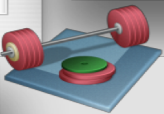
3 errors and warnings

- Block Diagram Errors

- Add: Contains unwired or bad terminal
- You have connected an Indicator to an input of Add. Change the
- You have connected a Control to an output of Add. Change the c

Close Show Error Help

Double-click on the Error message and LabVIEW will show you where the Error is!



Click **Ctrl+B** in order to remove All "broken wires"

# Data Types

Double1: 4.6

Integer1: 2

Boolean1: Off

String1: Hallo

Error In: status (On), code (12), source (Error when writing to DB)

Array - Numeric1: 3, 7, 12

Array - String1: Car, Plane, Train

Enum1: Alternative 2

Double2: 4.6

Integer2: 2

Boolean2: On

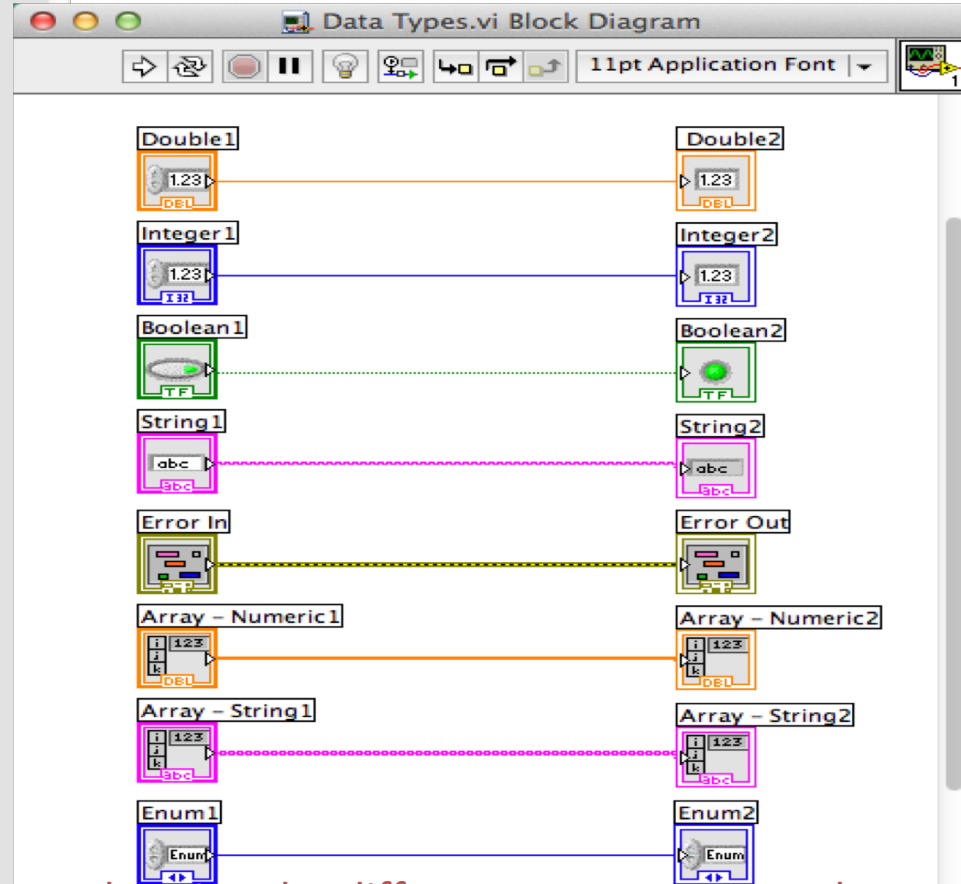
String2: Hallo

Error Out: status (On), code (12), source (Error when writing to DB)

Array - Numeric2: 3, 7, 12

Array - String2: Car, Plane, Train

Enum2: Alternative 2



Students: Create this Example using the different Data Types in LabVIEW

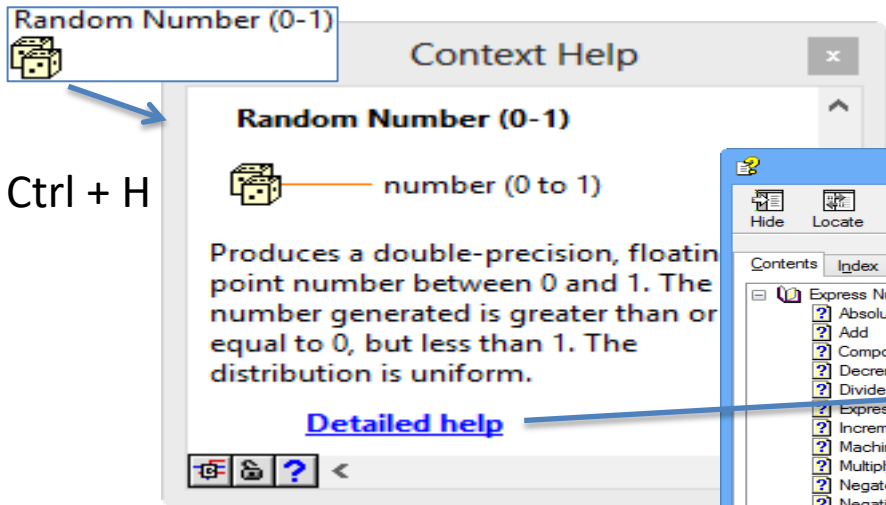
# Ctrl + H

# I need Help!!

Where can I find it?

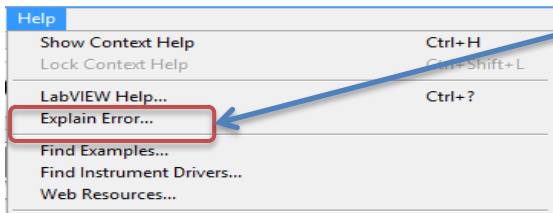
When you use **Ctrl + H**, you may click on all kind of objects (both on Front Panel and Block Diagram) on the screen to get help, e.g., how built-in functions are working, etc.

In the Help menu you will find Examples, etc.

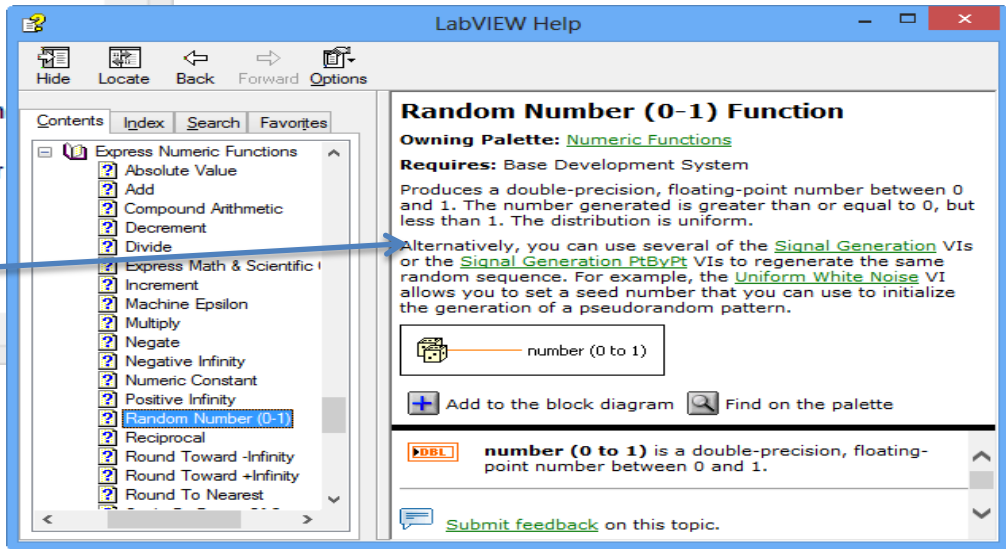


A screenshot of the LabVIEW Context Help window. The title bar reads "Context Help". The main content area shows the "Random Number (0-1)" function icon and the text: "Produces a double-precision, floating-point number between 0 and 1. The number generated is greater than or equal to 0, but less than 1. The distribution is uniform." Below the text is a "Detailed help" link. At the bottom left, there are icons for search, back, and forward. A blue arrow points from the "Random Number (0-1)" icon in the top-left corner to the function icon in the main content area. Another blue arrow points from the text "Ctrl + H" to the search icon.

Ctrl + H

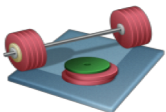


A screenshot of the LabVIEW Help menu. The menu items are: "Show Context Help" (Ctrl+H), "Lock Context Help" (Ctrl+Shift+L), "LabVIEW Help..." (Ctrl+?), "Explain Error..." (highlighted with a red box and a blue arrow pointing from the right), "Find Examples...", "Find Instrument Drivers...", and "Web Resources...".



A screenshot of the LabVIEW Help window. The title bar reads "LabVIEW Help". The main content area shows the "Random Number (0-1) Function" details, including the owning palette (Numeric Functions), requirements (Base Development System), and a description: "Produces a double-precision, floating-point number between 0 and 1. The number generated is greater than or equal to 0, but less than 1. The distribution is uniform." Below the description is a diagram of the function icon and a "number (0 to 1)" input. At the bottom, there is a "Submit feedback" link. On the left, there is a "Contents" pane with a list of functions, including "Random Number (0-1)" which is highlighted with a blue box and a blue arrow pointing from the right.

Students: Try this on different objects on your Block Diagram





# LabVIEW™ Quick Reference Guide

## Keyboard Shortcuts

File		Operate		Right-Click	
Ctrl-N	Create new VI	Ctrl-Z	Undo last action	Right-Click	Display controls/ functions palette
Ctrl-S	Save VI	Ctrl-Shift-Z	Redo last action	Shift-Right-Click	Display tools palette
Ctrl-P	Print			Ctrl-T	Tile block diagram and front panel windows
Edit		Window		Help	
Ctrl-V	Paste object	Ctrl-R	Run VI	Ctrl-H	Display context help
Ctrl-U	Clean up diagram	Ctrl-.	Abort VI		
Ctrl-Space	Activate quick drop				
Ctrl-B	Remove broken wires				
Ctrl-C	Copy an object				
Ctrl-X	Cut object				

## Editing Tools

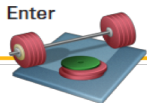
Tool	Icon	Description
Show Context Help		Display the context help window
Text Settings		Change the font setting for the VI, including size, style, and color
Align Objects		Align selected objects
Distribute Objects		Space objects evenly
Resize Objects		Resize multiple front panel objects to the same size
Reorder		Reorder the layers of the objects
Clean Up Diagram		Rearrange wires and objects on the block diagram
Enter		Appears when a new value is available to replace an old value

## Debugging Tools

Tool	Icon	Description
Run		Execute the VI
List Errors		List errors that prevent the VI from running
Run Continuously		Execute the VI continuously until abort or pause is pressed
Abort Execution		Stop VI execution immediately
Execution Highlighting		Animate data movement on the block diagram wires
Pause		Temporarily stop execution to debug a portion of the VI
Step Into		Single-step into a subVI or structure to debug it
Step Over		Execute a subVI or structure and pause at the next one
Step Out		Execute a subVI or structure and resume single-stepping

## Tools Palette

Tool	Icon	Description
Automatic Tool Selection		Automatically choose the appropriate tool
Operating Tool		Change the value of a control or select the text within a control
Positioning Tool		Position, resize, and select objects
Labeling Tool		Edit text and create free labels
Wiring Tool		Wire objects together on a block diagram
Scrolling Tool		Scroll the window without using the scroll bars
Breakpoint Tool (Used for debugging)		Set breakpoints on VIs, functions, wires, loops, sequences, and cases
Probe Tool (Used for debugging)		Create probes on wires and display intermediate values on a wire in a running VI
Get Color Tool		Copy colors for pasting with the Color Tool
Coloring Tool		Set the foreground and background colors

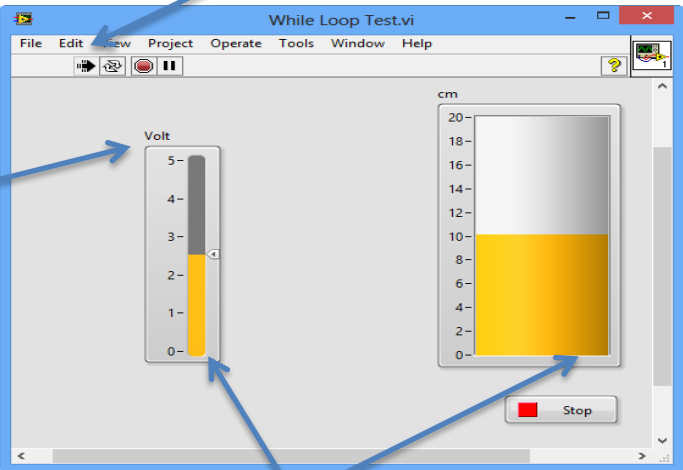


Students: Try some of these Shortcuts and Tools

# While Loop

Front Panel

4 Run the Program



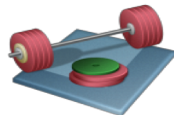
Label

1 Front Panel: Find these Controls in the Controls Palette and place them on the Front Panel with proper labels

2 Block Diagram: Find the **While Loop** in the Functions Palette and place it on the Block Diagram

Note! To do something with an object – Right-click on it!

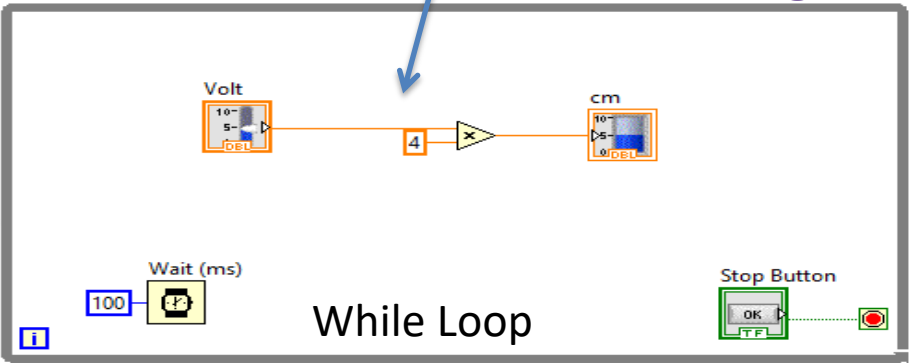
Example: A voltage signal [0-5V] from a DAQ device needs to be converted to the equivalent level values in a water tank [0-20cm]



Students: Create this Example

3 Block Diagram: Connect these together using the "Wiring tool" (your mouse)

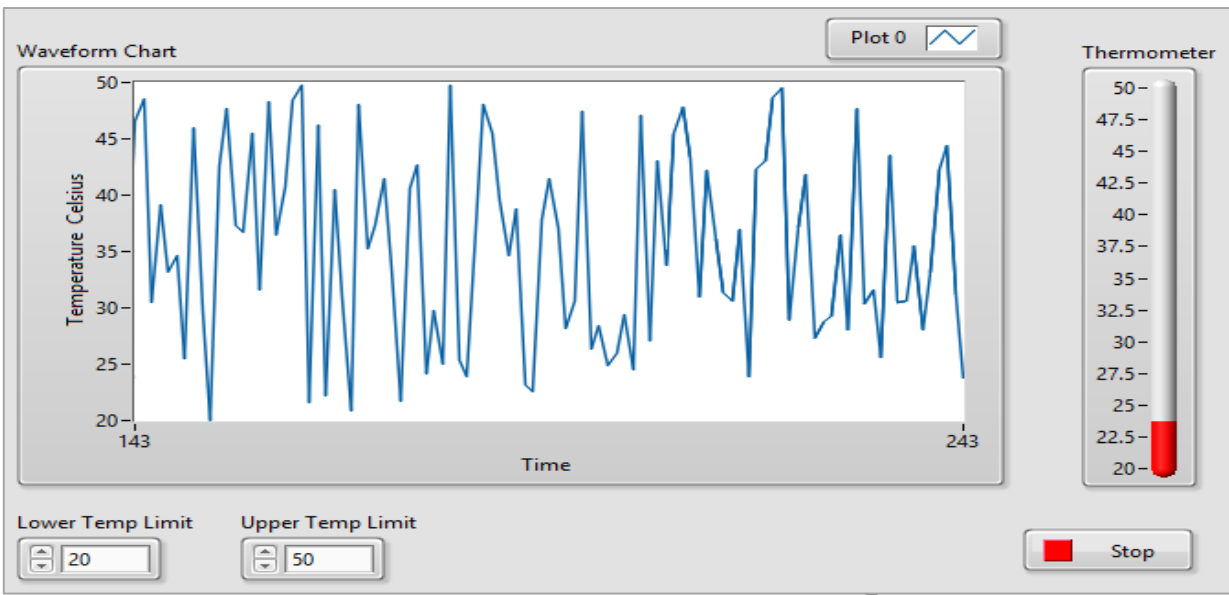
Block Diagram



While Loop

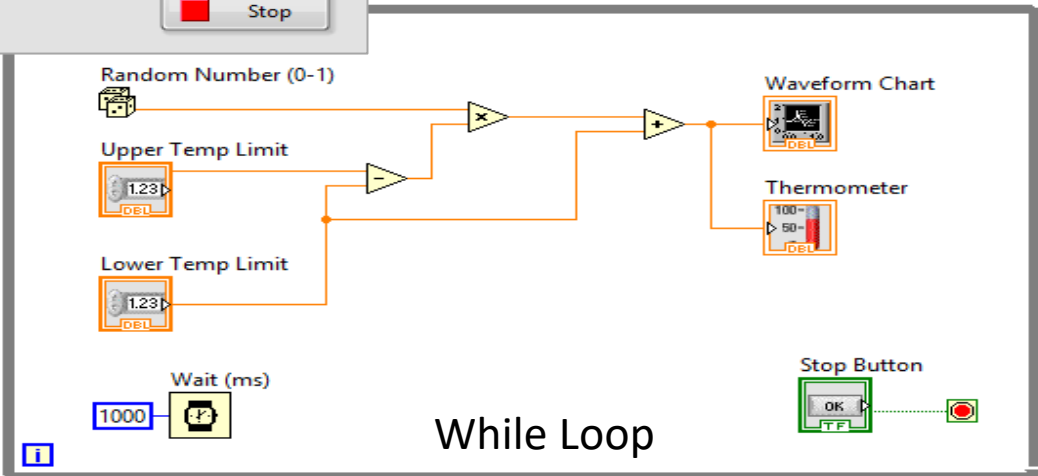
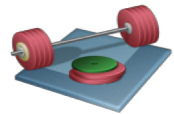
# Plotting

This example simulates the Temperature in an “Air Heater” system. The Temperature in the Air Heater should be between 20 and 50 degrees Celsius. We use the **Random Generator** in LabVIEW in this Example



Students:

- Create this Example
- Try out different options on the Chart, e.g. Autoscale, different Modes, Grid, Layout, Colors, etc. (Right-click on the Chart)

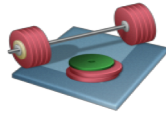


**Note!** To do something with an object – Right-click on it

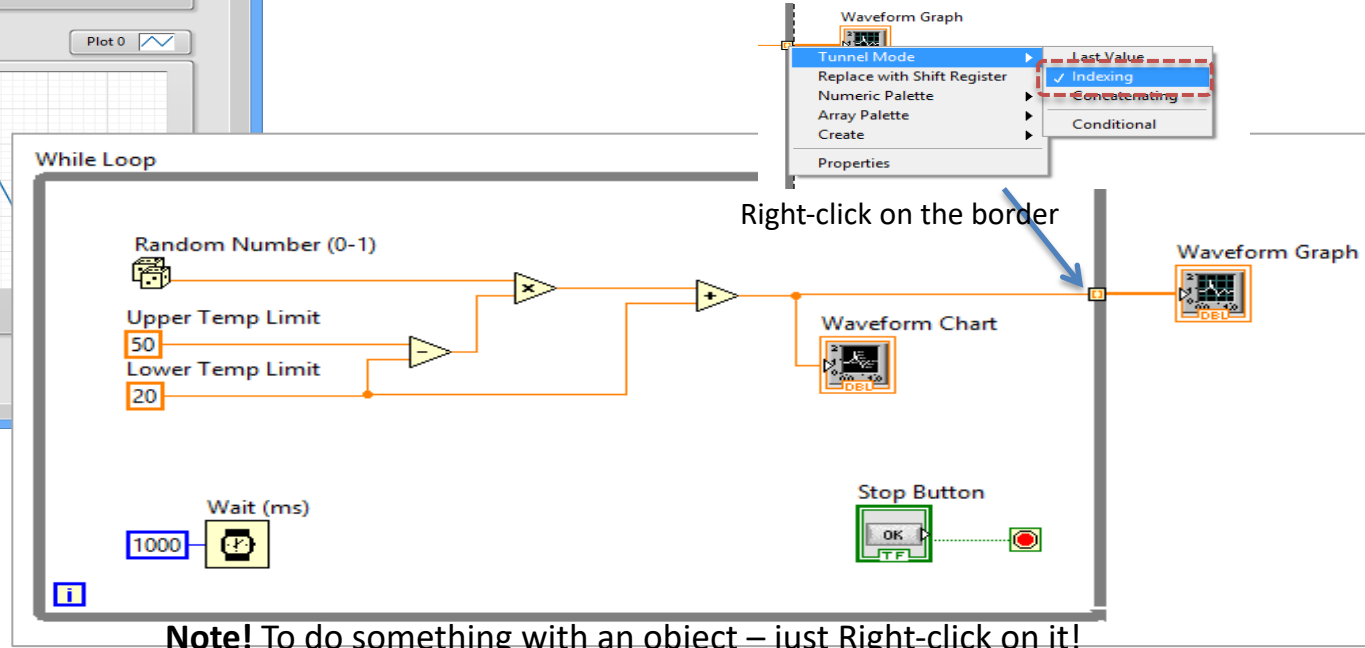
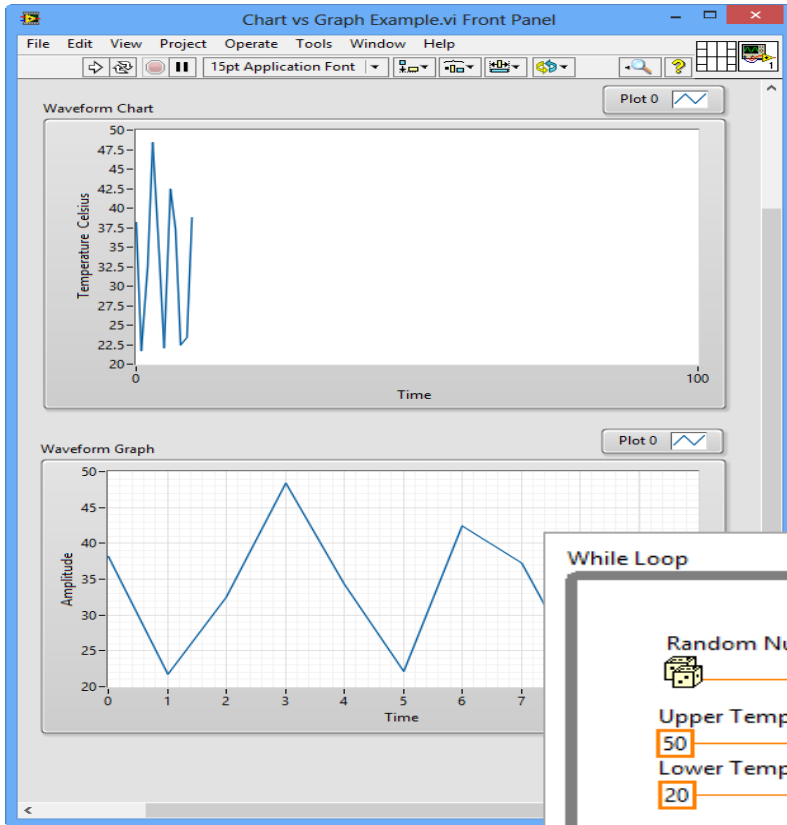


# Charts vs. Graphs

## - Example



Students: Create this VI in order to see the difference between a **Chart** and a **Graph**



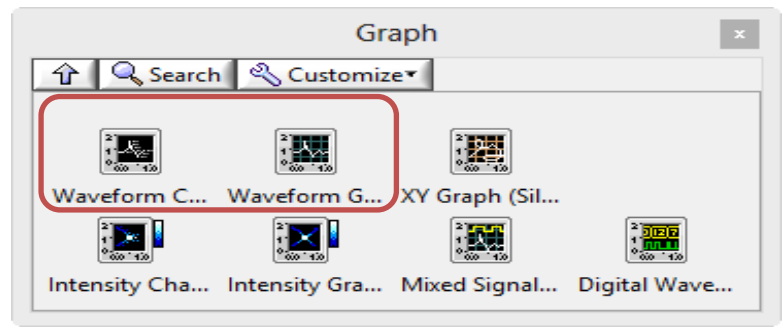
# Charts vs. Graphs

## Chart:

- Remembers history – New point added to end of the plot
- Used inside a While Loop/For Loop
- One new point is added each time

## Graph:

- You plot all the data at once – typically an array with data
- Used outside a While Loop/For Loop

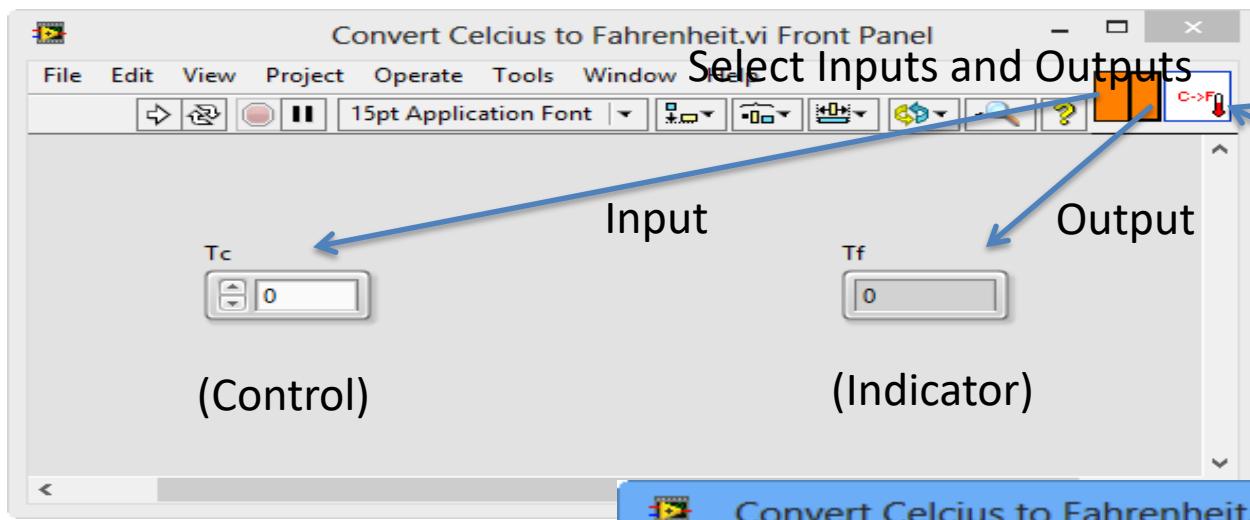


<https://www.halvorsen.blog>



# SubVIs

Hans-Petter Halvorsen

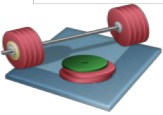


# SubVIs

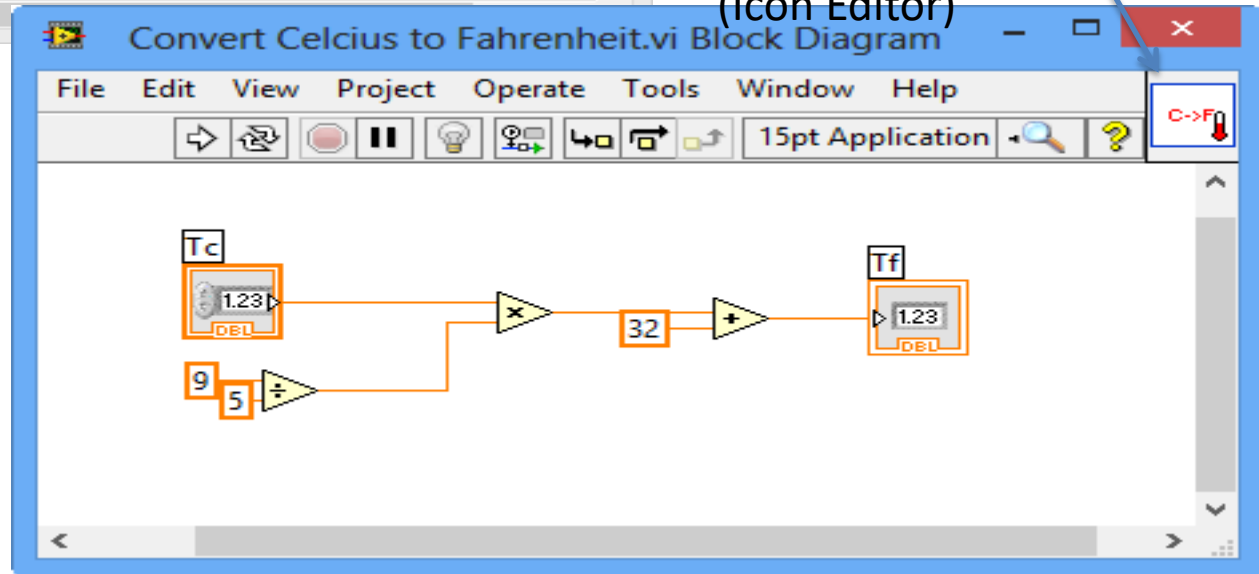
Icon that makes it easier to understand what the SubVI is doing  
 Create a nice icon as well (Icon Editor)

A SubVI is the same as a function or a method used in other languages

$$T_F = \frac{9}{5} T_C + 32$$



Students: Create this SubVI



# Icon Editor

Use the Icon Editor in order to create a descriptive icon for your SubVI.

Icon Editor (Convert Celcius to Fahrenheit.vi)

Templates Icon Text Glyphs Layers

Line 1 text   Line 1 color

Line 2 text   Line 2 color

Line 3 text   Line 3 color

Line 4 text   Line 4 color

Font

Alignment  
 10

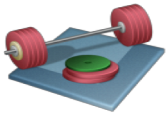
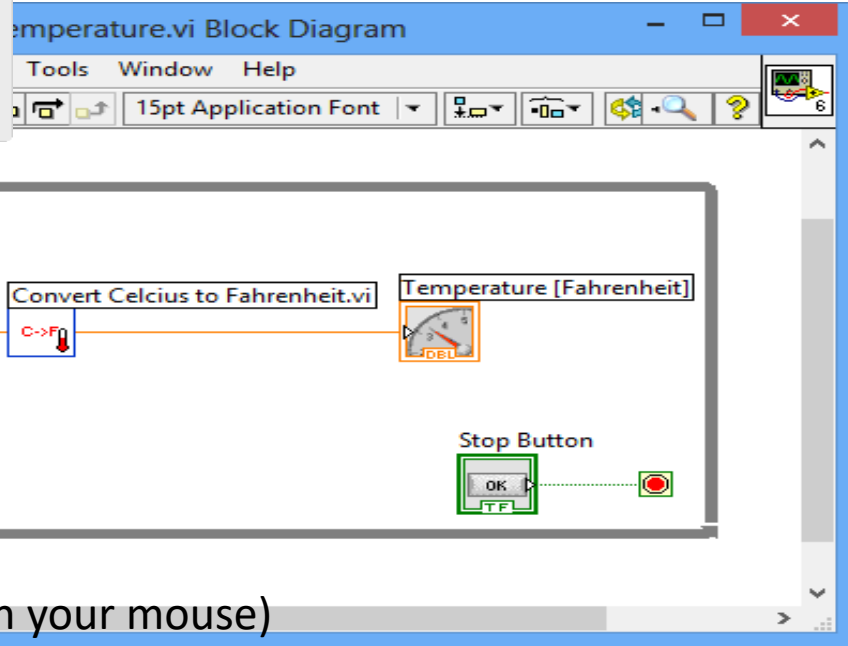
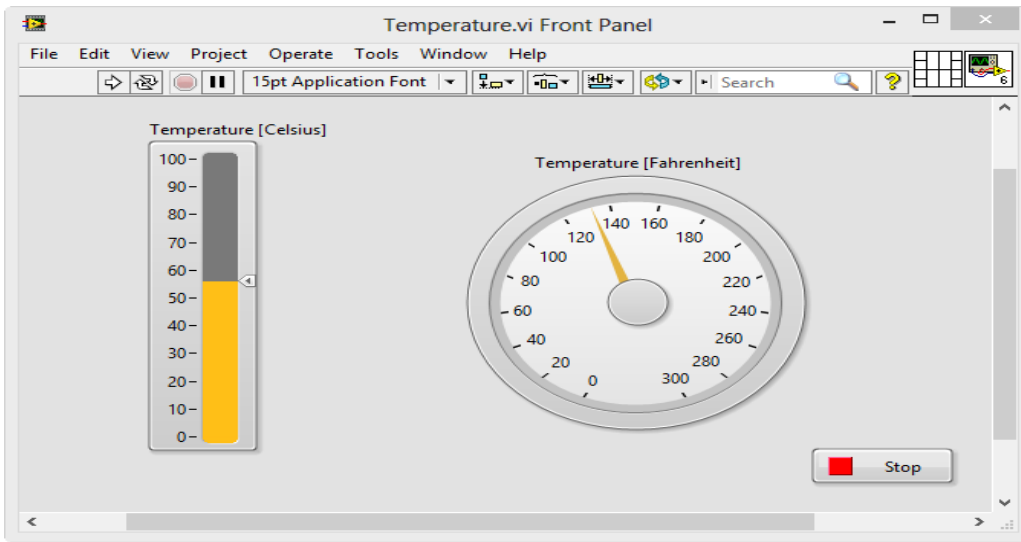
Center text vertically  
 Capitalize text

Color palette:

Color code: R: 0 X: 0  
G: 65 Y: 0  
B: 220 Z: 2

OK Cancel Help

# Using SubVIs

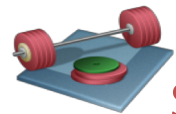
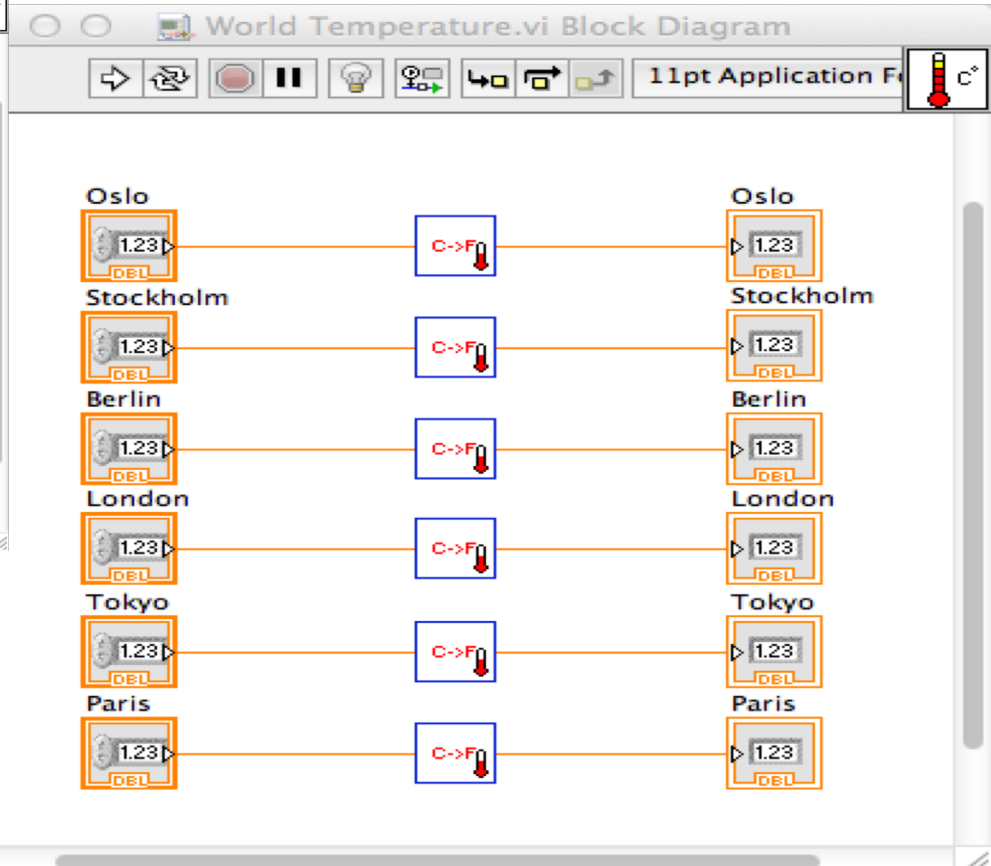
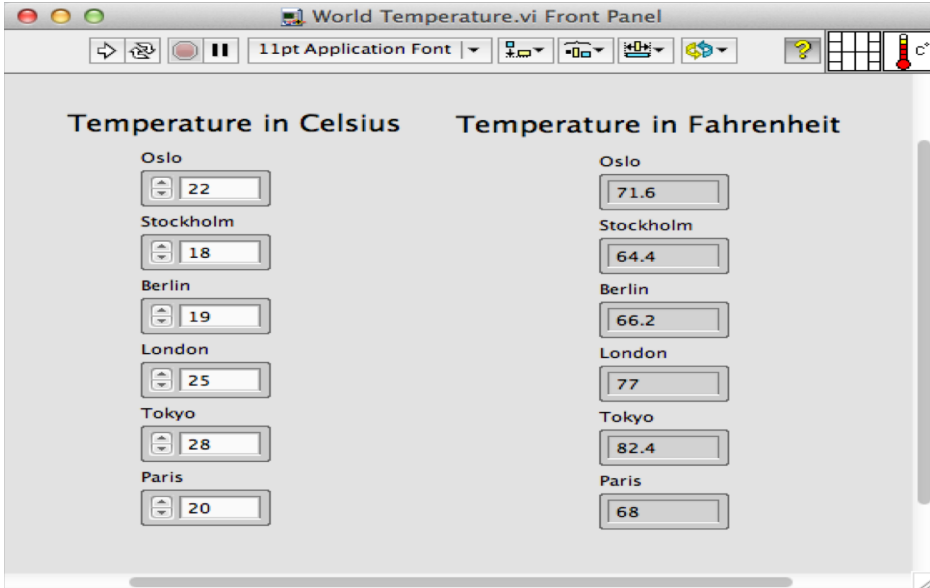


Students: Create this VI

SubVI created in previous slide  
(just drag it on the Block Diagram with your mouse)

# SubVIs

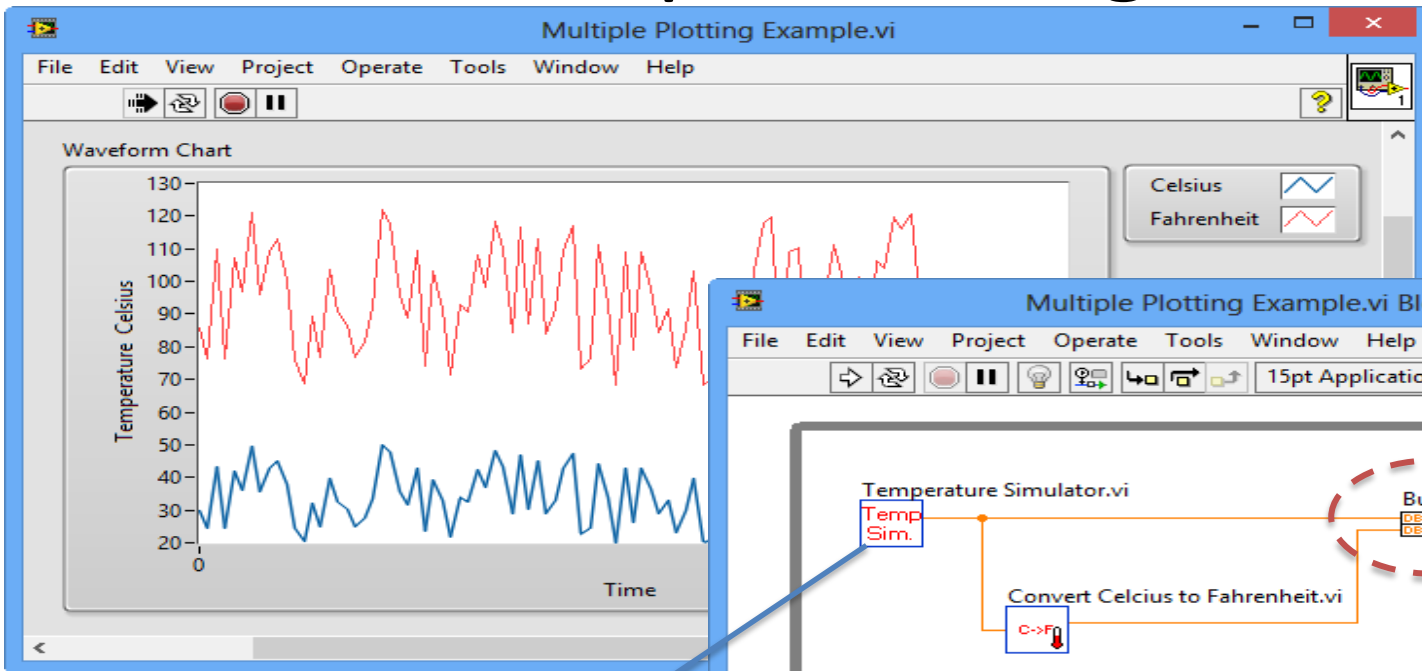
Here you see some of the advantages with using SubVIs.

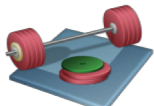


Students: Create this Example

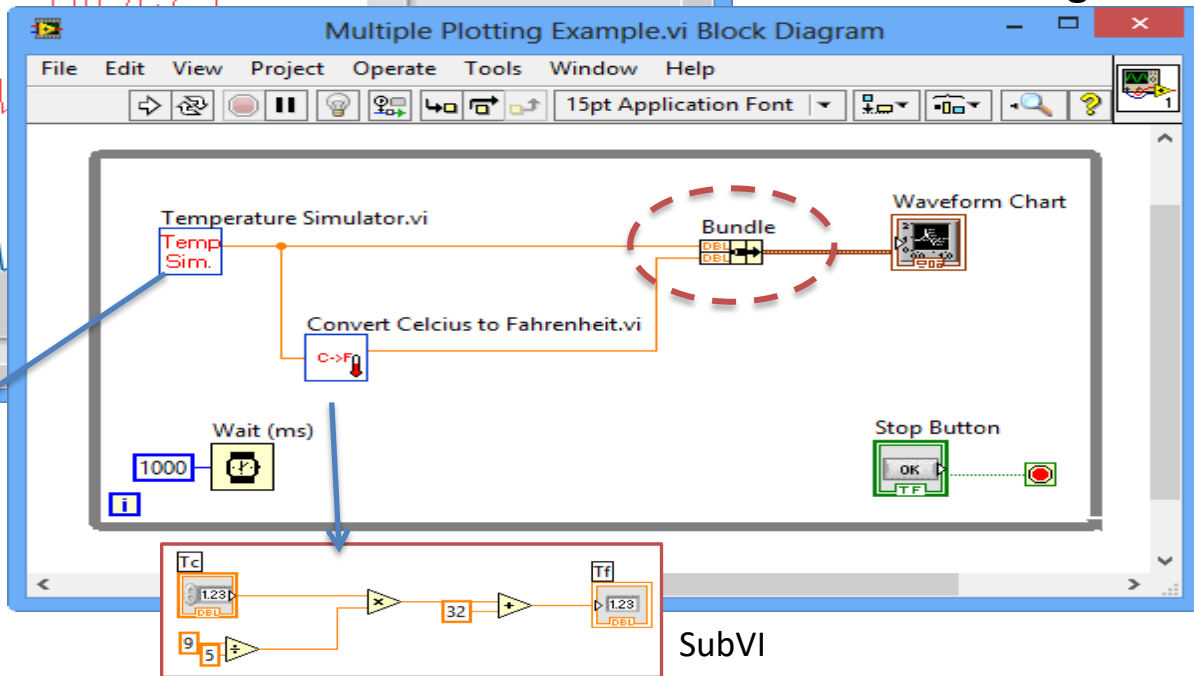
# Multiple Plotting and SubVIs

Front Panel



Students:  Create this program

Block Diagram



Do you see how simple and intuitive the code becomes when using SubVIs?

SubVI





# Writing Formulas

Hans-Petter Halvorsen

# LabVIEW Formula Node

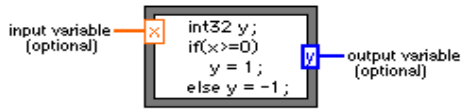
**Formula Node:** Create and use C code within LabVIEW

Example:

Very useful for mathematical expressions and simulations!

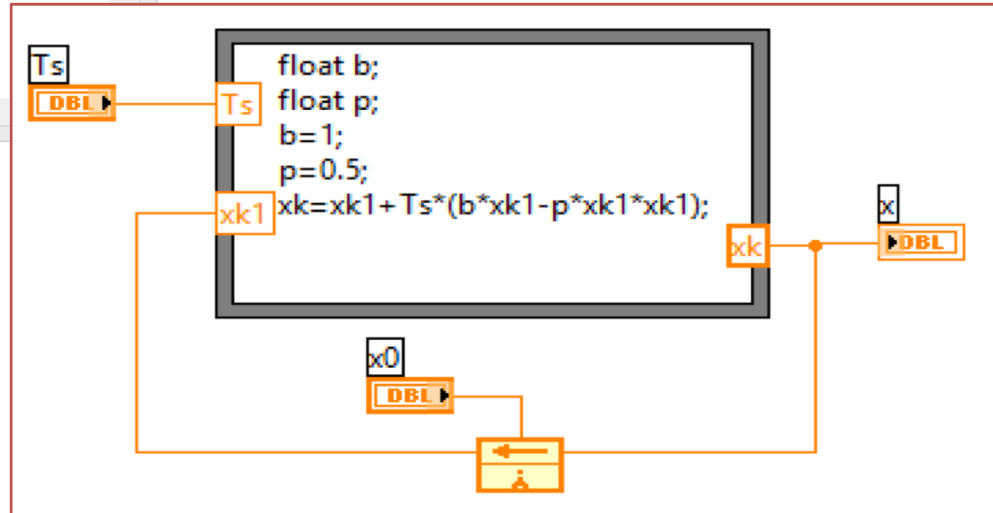
Context Help

## Formula Node



Evaluates mathematical formulas and expressions similar to C on the block diagram. The following built-in functions are allowed in formulas: abs, acos, acosh, asin, asinh, atan, atan2, atanh, ceil, cos, cosh, cot, csc, exp, expm1, floor, getexp, getman, int, intrz, ln, ln1p, log, log2, max, min, mod, pow, rand, rem, sec, sign, sin, sinc, sinh, sizeofDim, sqrt, tan, tanh. There are some differences between the parser in the Mathematics VIs and the Formula Node.

[Detailed help](#)



# LabVIEW Formula Node

Celsius to Fahrenheit:

$$T_F = \frac{9}{5}T_C + 32$$

Students: Use the LabVIEW Formula Node in order to implement this formula

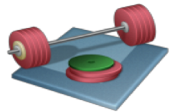
Advanced Mathematical Formula:

$$f(x) = \frac{\ln(ax^2 + bx + c) - \sin(ax^2 + bx + c)}{4\pi x^2 + \cos(x - 2)(ax^2 + bx + c)}$$

Given  $a = 1, b = 3, c = 5$

Find  $f(9)$

Try also with other values for a, b, c



Students: Use the LabVIEW Formula Node in order to implement this formula

Try also to implement this function with ordinary LabVIEW blocks. What is easiest to implement?

(The answer should be  $f(9) = 0.0044$ )

Solutions

$$f(x) = \frac{\ln(ax^2 + bx + c) - \sin(ax^2 + bx + c)}{4\pi x^2 + \cos(x - 2)(ax^2 + bx + c)}$$

Formula Node Example.vi Block Diagram

Note!

```
f = (ln(a*x*x + b*x + c) - sin(a*x*x + b*x + c)) / (4*pi*x*x + cos(x-2)*(a*x*x + b*x + c));
```

Formula Node Example.vi Front Panel

a: 1  
b: 3  
c: 5  
x: 9  
f: 0,00437378

Formula Node Example2.vi Block Diagram

Alternative Solution:

```
float g;  
g = a*x*x + b*x + c;  
f = (ln(g) - sin(g)) / (4*pi*x*x + cos(x-2)*(g));
```

Which Solution do you think is best?

# LabVIEW MathScript Node

**MathScript Node**

```
1 SumA = eye(size(A));
2 for i = 1:n
3 SumA = SumA + A^i/factorial(i);
4 end
5 Delta = SumA - expmx(A);
```

input variable (optional) **A**

input variable (optional) **n**

error in

**Delta** output variable (optional)

error out

Executes LabVIEW MathScripts and your other text-based scripts using the MathScript RT Module engine. You can use the MathScript Node to evaluate scripts that you create in the LabVIEW MathScript Window.

If a MathScript Node contains a warning glyph, LabVIEW operates with slower run-time performance for the node. You can modify your script to remove the warning glyph from the MathScript Node and improve run-time performance.

[Detailed help](#)

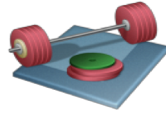
Very useful for mathematical expressions and simulations!

**MathScript Node:** You can create and use MathScript/MATLAB code within LabVIEW

# LabVIEW MathScript Node

Celcius to Fahrenheit:

$$T_F = \frac{9}{5}T_C + 32$$



Students: Use the LabVIEW MathScript Node in order to implement this formula

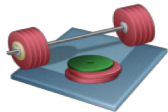
Advanced Mathematical Formula:

$$f(x) = \frac{\ln(ax^2 + bx + c) - \sin(ax^2 + bx + c)}{4\pi x^2 + \cos(x - 2)(ax^2 + bx + c)}$$

Given  $a = 1, b = 3, c = 5$

Find  $f(9)$

Try also with other values for a, b, c



Students: Use the LabVIEW MathScript Node in order to implement this formula

(The answer should be  $f(9) = 0.0044$ )

# Solutions

$$f(x) = \frac{\ln(ax^2 + bx + c) - \sin(ax^2 + bx + c)}{4\pi x^2 + \cos(x - 2)(ax^2 + bx + c)}$$

MathScript Node Example.vi Block Diagram

```
1 g = a*x^2 + b*x + c
2 f = (log(g) - sin(g)) / (4*pi*x^2 + cos(x-2)*(g))
```

Inputs: a, b, c, x (all set to 1.23)

Output: f (set to 1.23)

Formula Node Example.vi Front Panel

Inputs: a=1, b=3, c=5, x=9

Output: f = 0,00437378

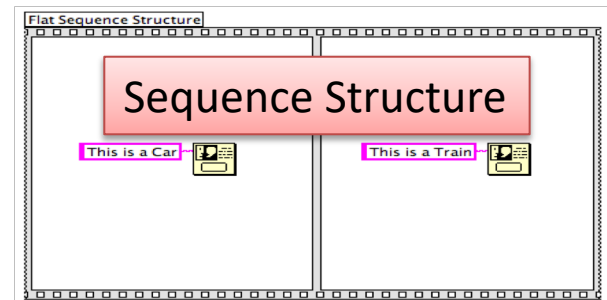
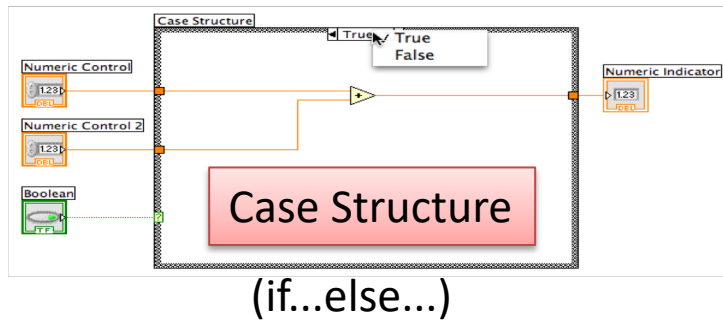
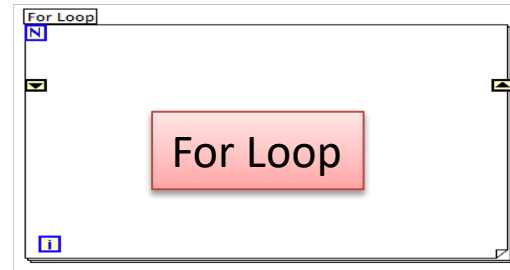
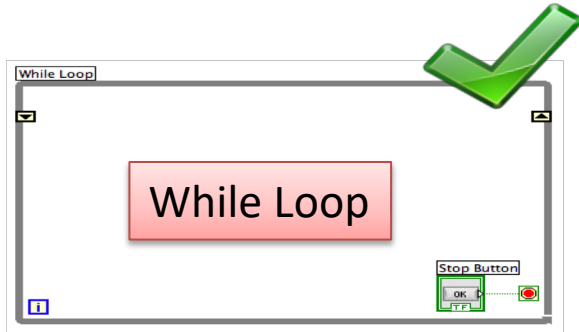


# More Loops & Structures

Hans-Petter Halvorsen



# Loops & Structures



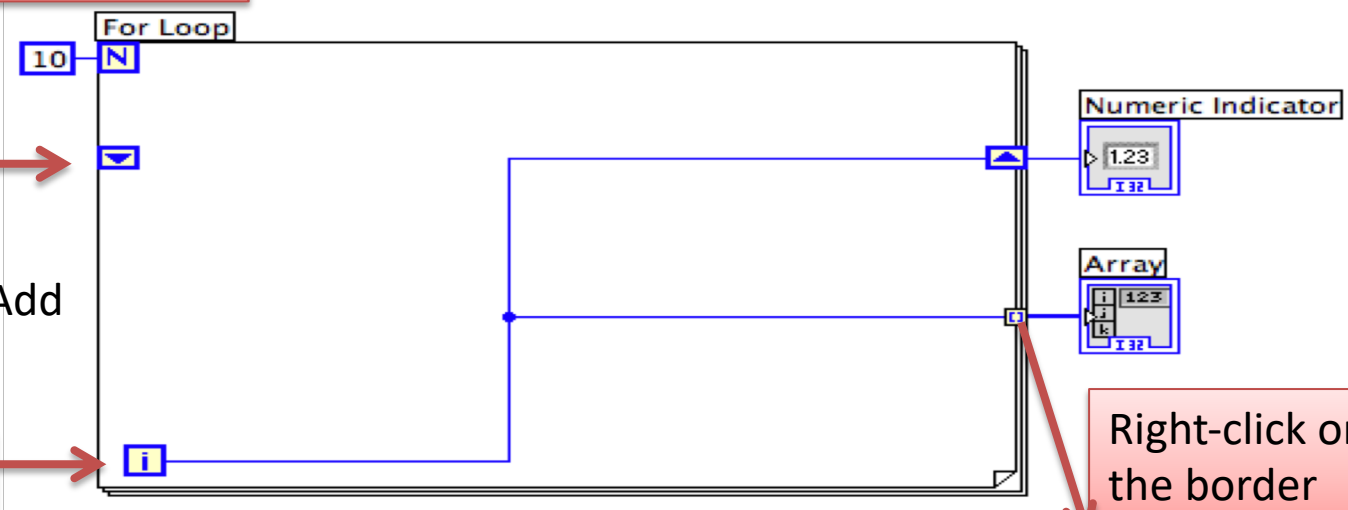
# For Loop

Specify Number of Iterations

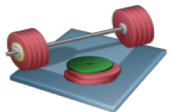
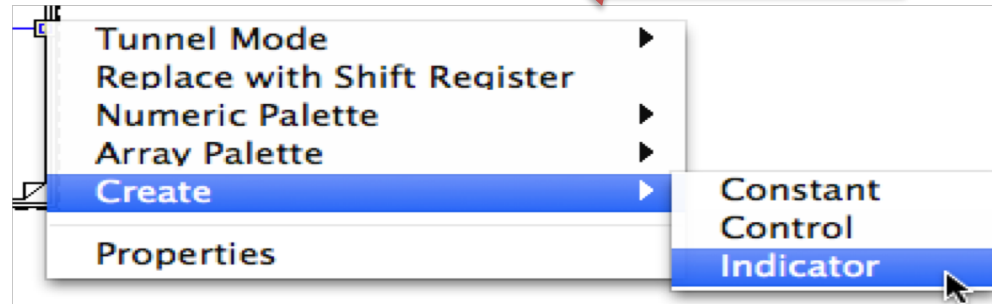
Shift Register

Right-click on the border and select "Add Shift Register"

Current Iteration

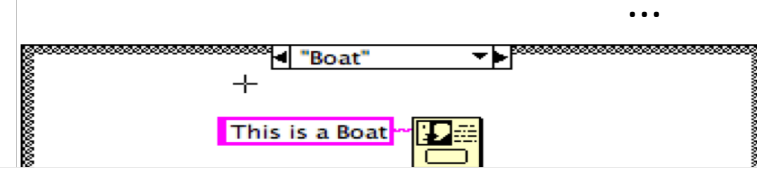
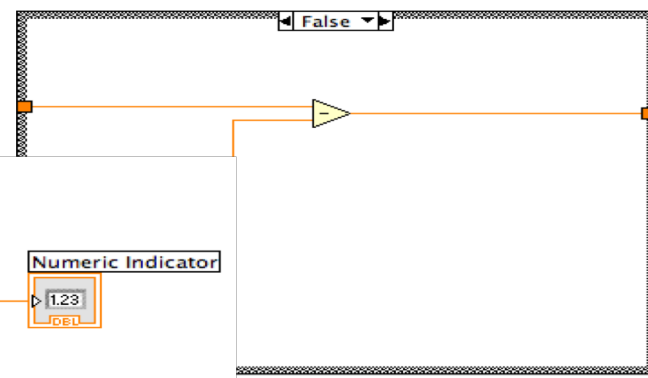
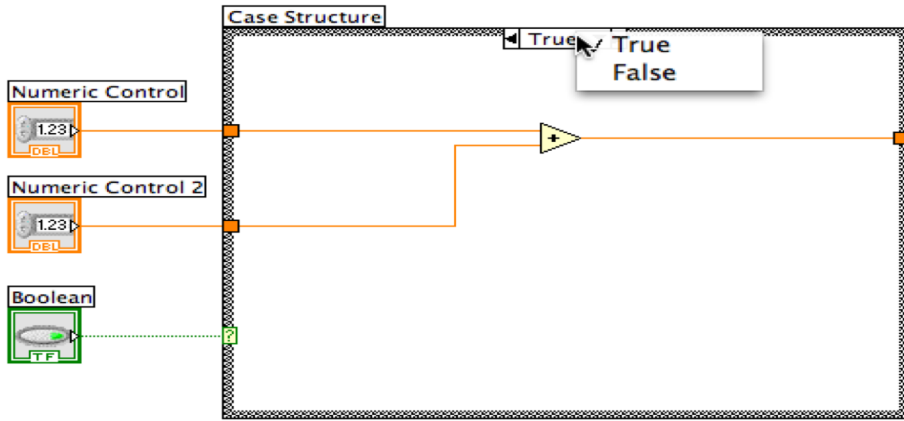


Right-click on the border



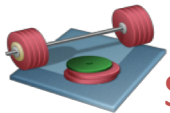
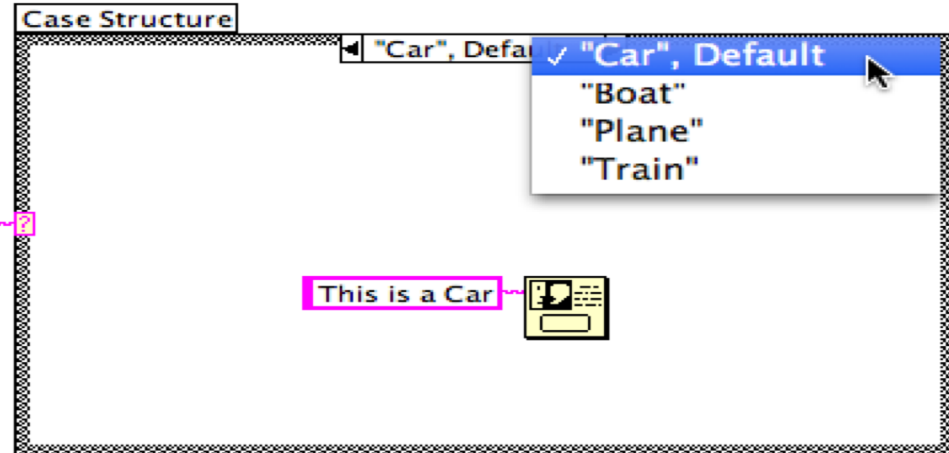
Students: Create this Example

# Case Structure



The Case structure is the same as **if... else...** used in other programming languages like C#, etc.

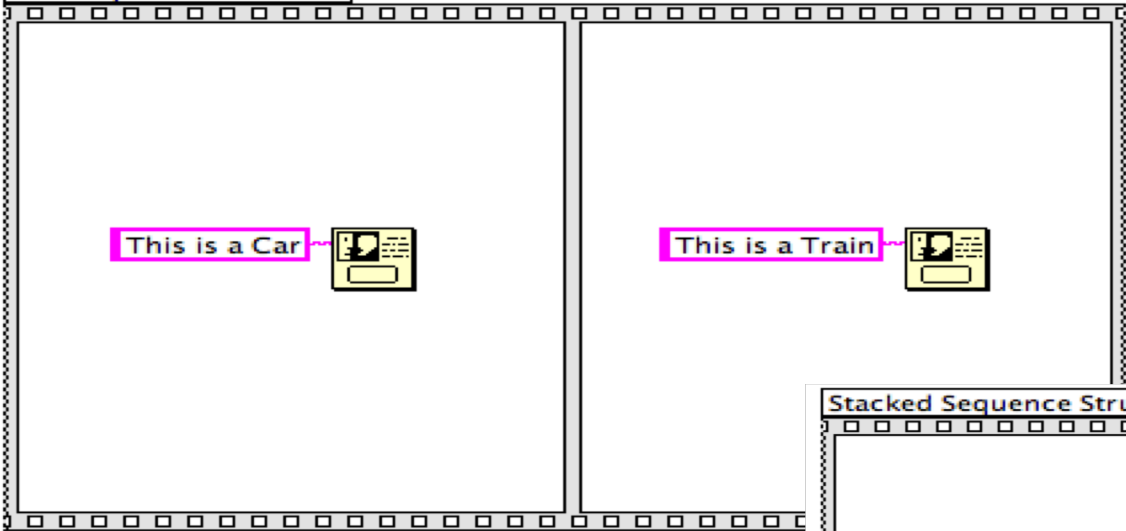
Write either "Car" or "Boat" and see what happens



Students: Create these Examples

# Sequence Structure

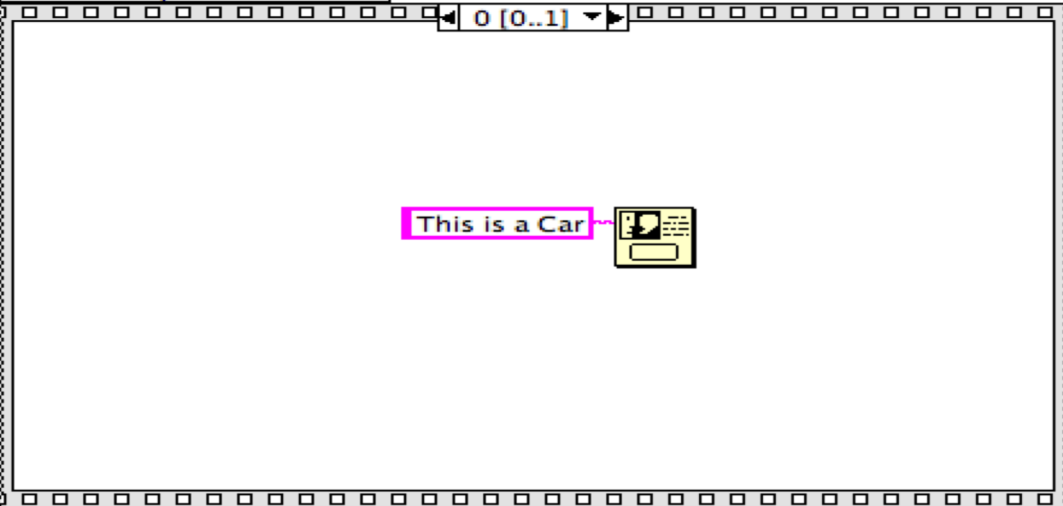
Flat Sequence Structure



Flat Sequence Structure

Stacked Sequence Structure

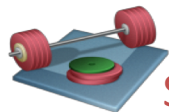
Stacked Sequence Structure



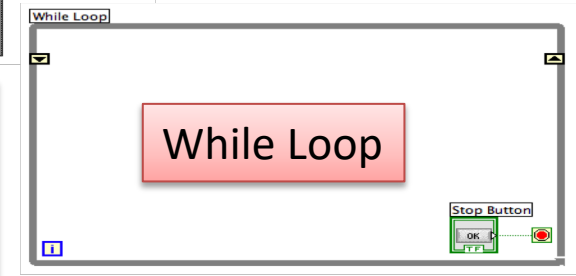
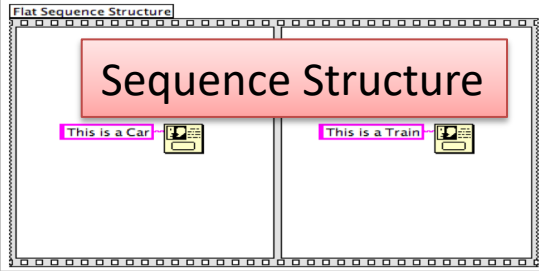
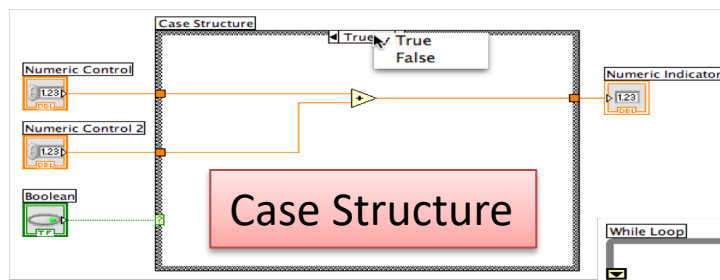
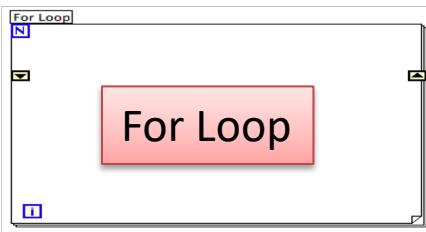
One Button Dialog



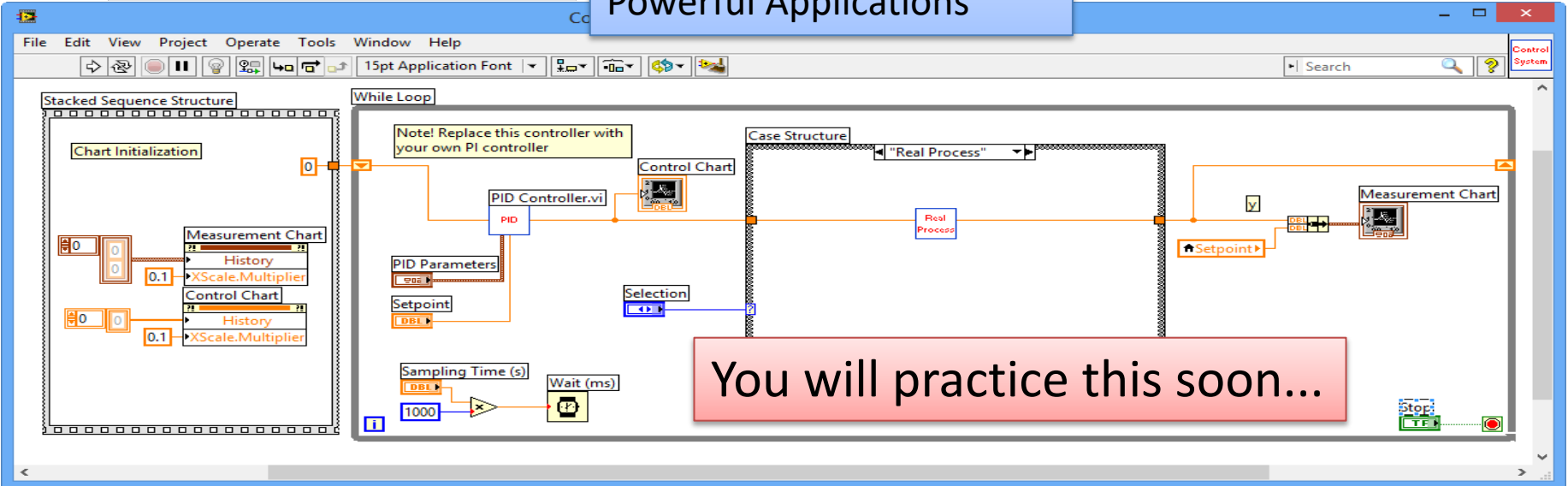
The "One Button Dialog" can be used to give a popup message to the user



Students: Create these Examples



When Combining Loops and Structures, you can create Advanced and Powerful Applications



<https://www.halvorsen.blog>

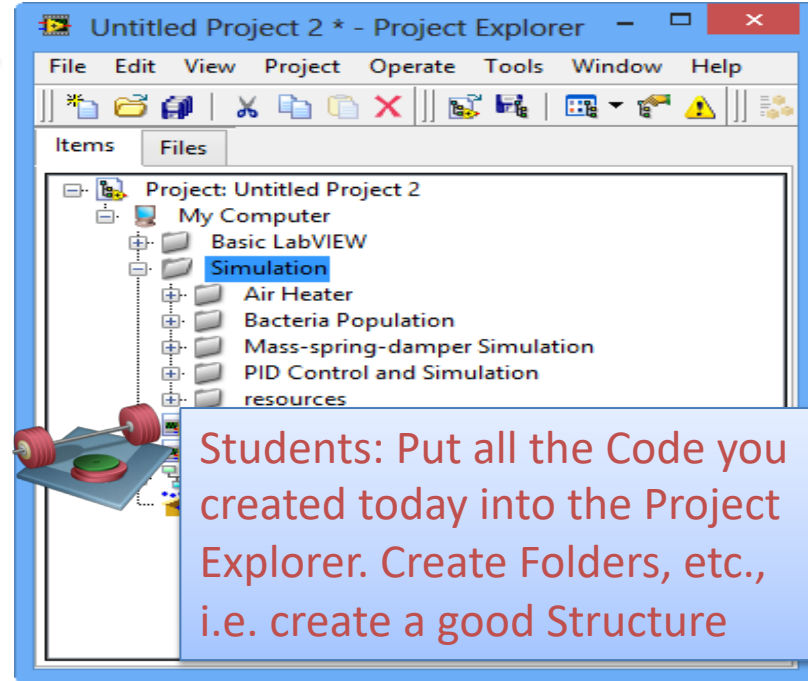
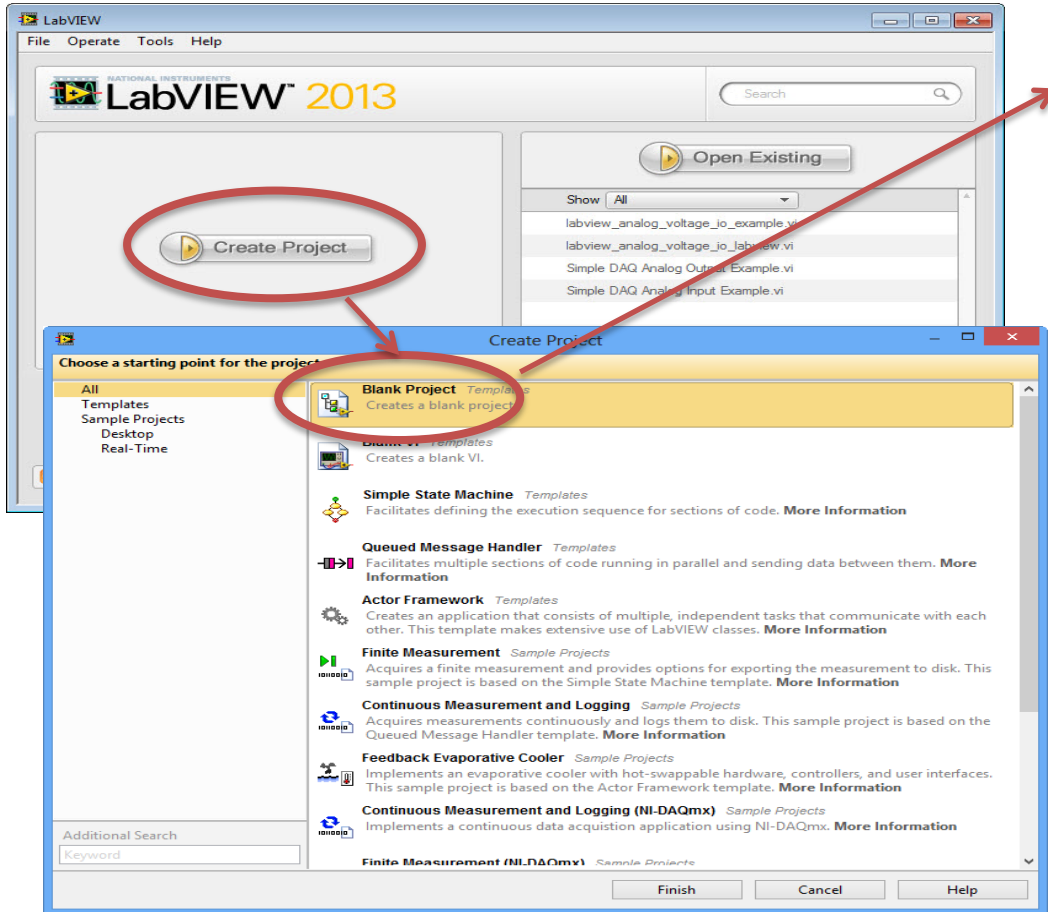


# Tips & Tricks

Hans-Petter Halvorsen

# Project Explorer

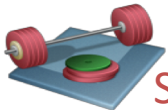
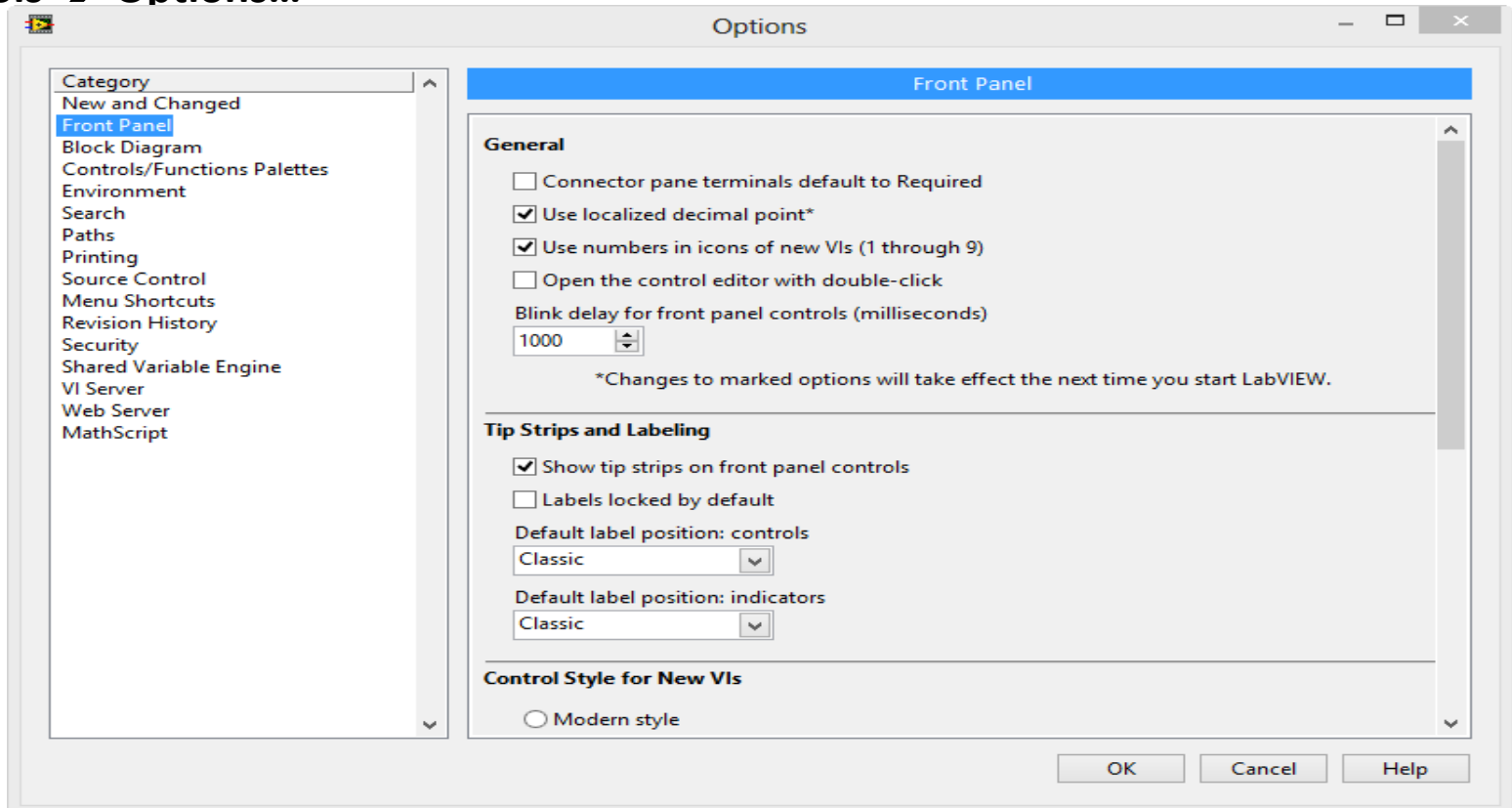
Similar to “Solutions Explorer” in Visual Studio. It Keeps all your Files for a specific project in one place.



It is recommended to use this when you create larger applications that contains lots of VIs and other files, etc.

# Customizing LabVIEW IDE

Select Tools → Options...



Students: Change the different options according to how you want to use LabVIEW

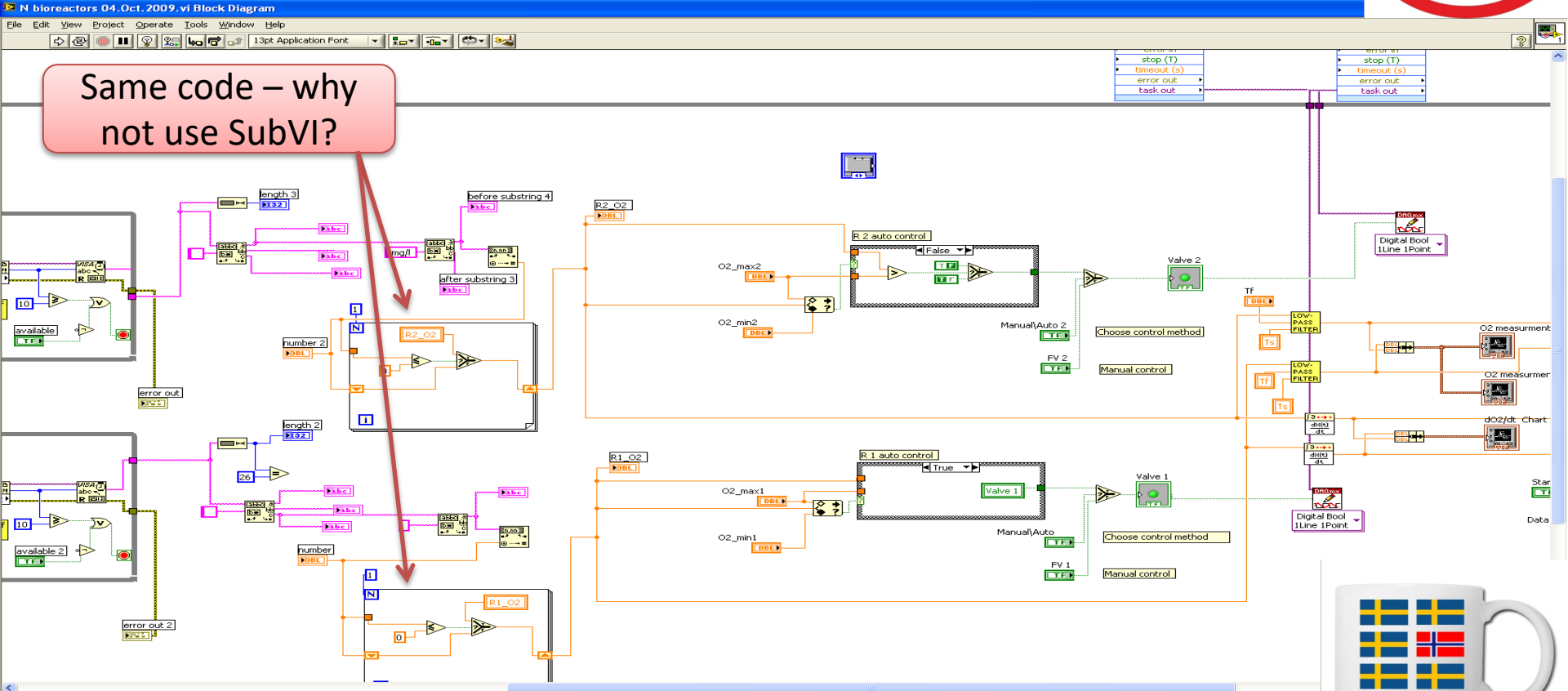




# Spaghetti Code

Since LabVIEW is a graphical programming language with lots of wires, etc., it is extremely important to have a good and clear structure in your program!

# Spaghetti Code – Bad Example 1



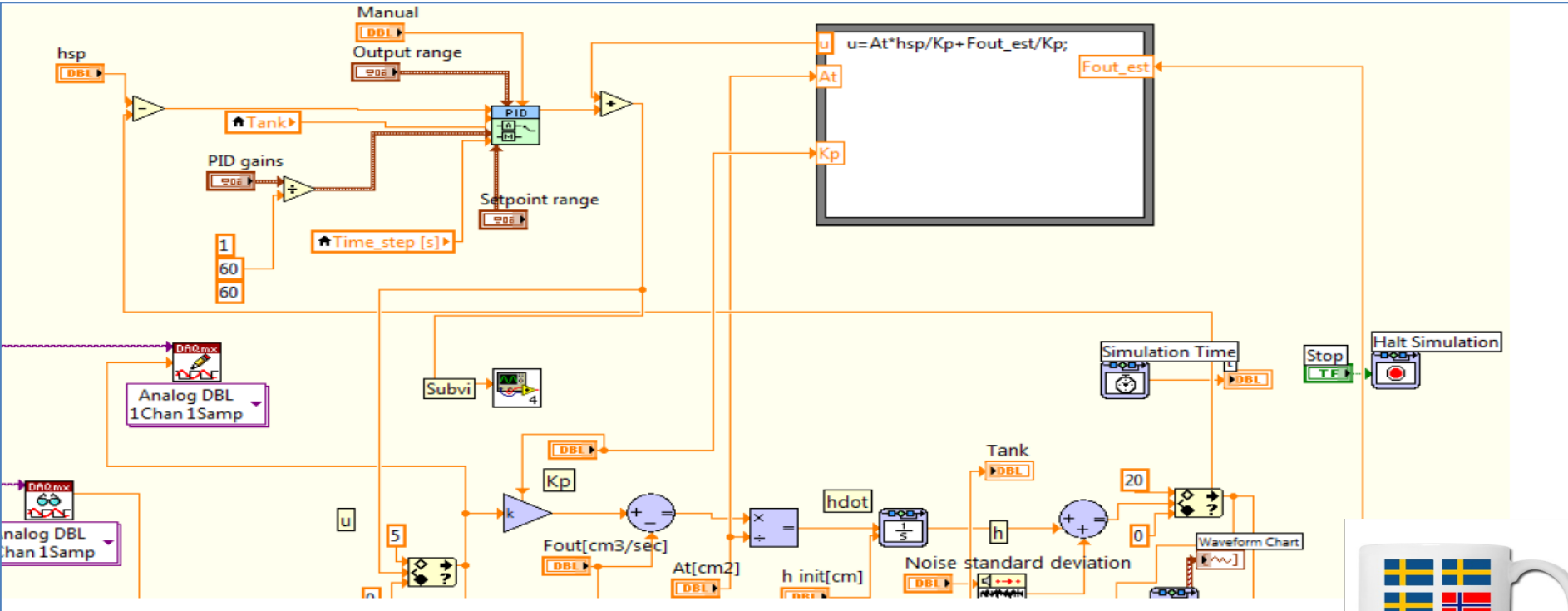
Same code – why not use SubVI?

Structure your code! Use SubVIs, Avoid Spaghetti Code, Document your Code, etc.

# Spaghetti Code – Bad Example 2

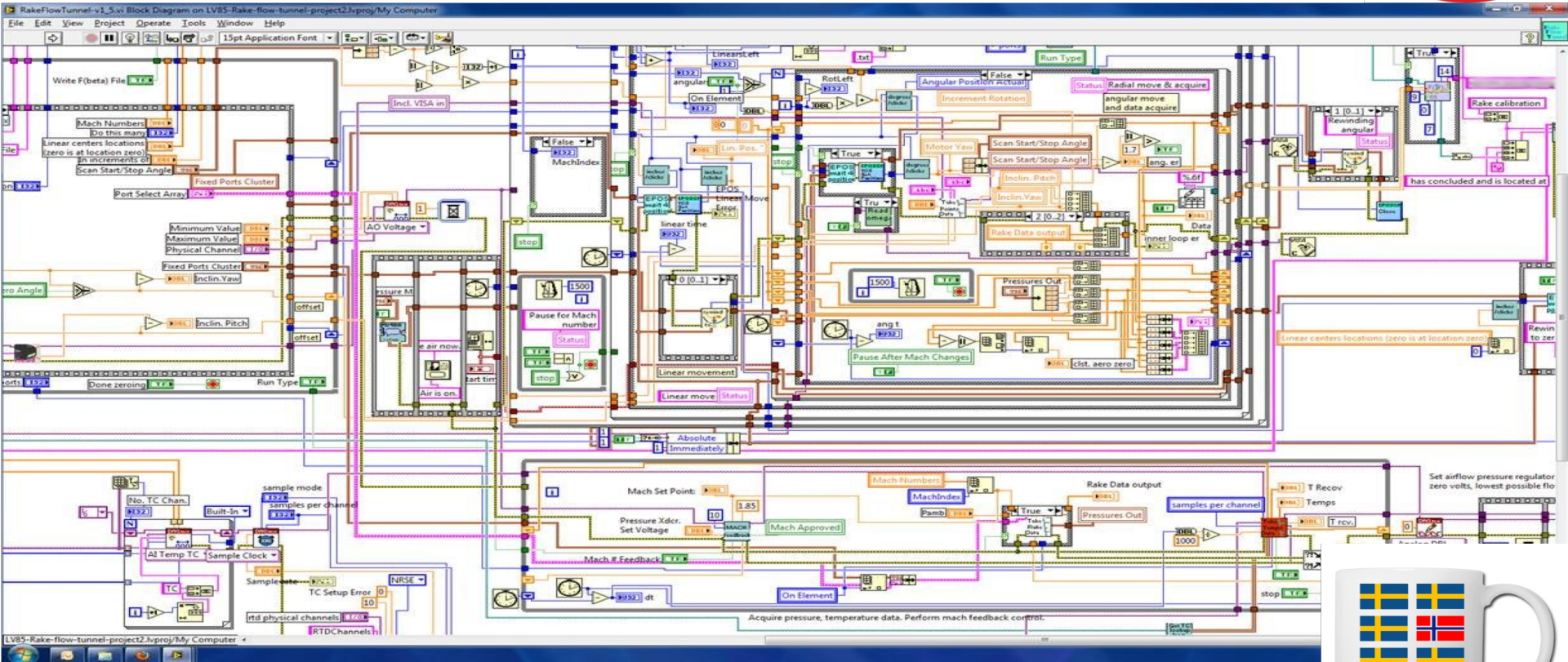
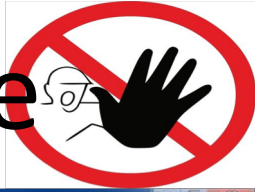


Make sure your code fits into your screen size – scrolling to see code is not good!



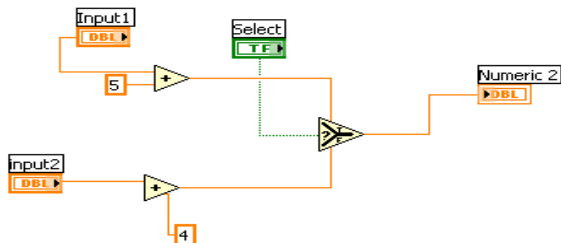
Structure your code! Use SubVIs, Avoid Spaghetti Code, Document your Code, etc.

# Spaghetti Code – Very Bad Example



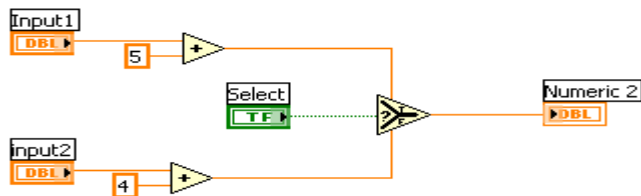
Structure your code! Use SubVIs, Avoid Spaghetti Code, Document your Code, etc.

# Bad vs. Good Code



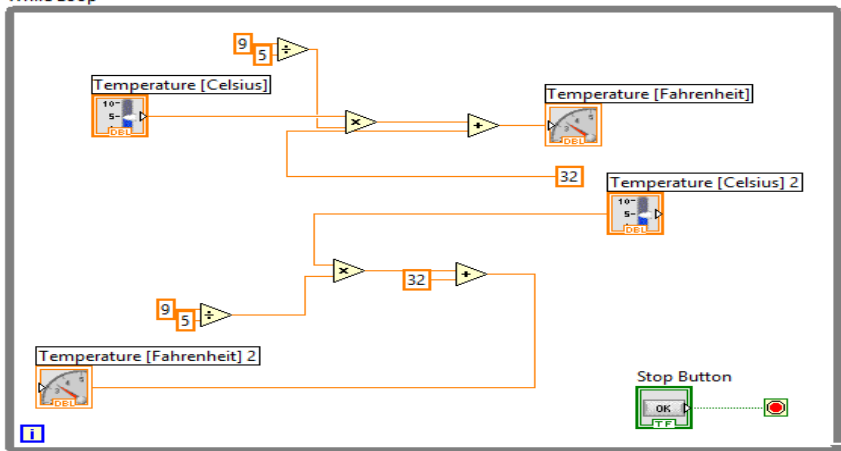
Avoid Spaghetti Code!

The Flow should go from left to right

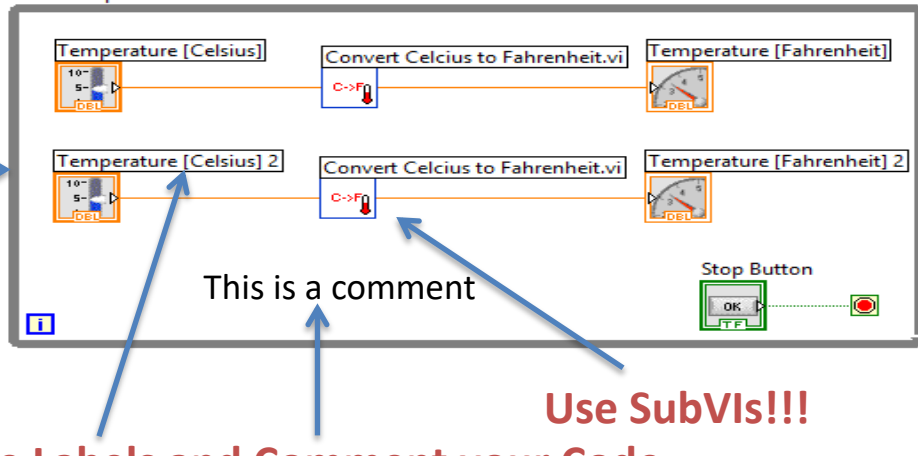


Make your code readable for others!

While Loop



While Loop



Use Labels and Comment your Code

Use SubVIs!!!



# LabVIEW™ Quick Reference Guide

## Keyboard Shortcuts

File			
Ctrl-N	Create new VI	Ctrl-Z	Undo last action
Ctrl-S	Save VI	Ctrl-Shift-Z	Redo last action
Ctrl-P	Print		
Operate			
Ctrl-V	Paste object	Ctrl-R	Run VI
Ctrl-U	Clean up diagram	Ctrl-.	Abort VI
Ctrl-Space	Activate quick drop		
Ctrl-B	Remove broken wires		
Ctrl-C	Copy an object		
Ctrl-X	Cut object		
Window			
Ctrl-E	Display block diagram/ front panel		
Right-Click			
		Right-Click	Display controls/ functions palette
Shift-Right-Click			
		Shift-Right-Click	Display tools palette
Ctrl-T			
		Ctrl-T	Tile block diagram and front panel windows
Help			
		Ctrl-H	Display context help

## Editing Tools

Tool	Icon	Description
Show Context Help		Display the context help window
Text Settings		Change the font setting for the VI, including size, style, and color
15pt Application Font		
Align Objects		Align selected objects
Distribute Objects		Space objects evenly
Resize Objects		Resize multiple front panel objects to the same size
Reorder		Reorder the layers of the objects
Clean Up Diagram		Rearrange wires and objects on the block diagram
Enter		Appears when a new value is available to replace an old value

## Debugging Tools

Tool	Icon	Description
Run		Execute the VI
List Errors		List errors that prevent the VI from running
Run Continuously		Execute the VI continuously until abort or pause is pressed
Abort Execution		Stop VI execution immediately
Execution Highlighting		Animate data movement on the block diagram wires
Pause		Temporarily stop execution to debug a portion of the VI
Step Into		Single-step into a subVI or structure to debug it
Step Over		Execute a subVI or structure and pause at the next one
Step Out		Execute a subVI or structure and resume single-stepping

## Tools Palette

Tool	Icon	Description
Automatic Tool Selection		Automatically choose the appropriate tool
Operating Tool		Change the value of a control or select the text within a control
Positioning Tool		Position, resize, and select objects
Labeling Tool		Edit text and create free labels
Wiring Tool		Wire objects together on a block diagram
Scrolling Tool		Scroll the window without using the scroll bars
Breakpoint Tool (Used for debugging)		Set breakpoints on VIs, functions, wires, loops, sequences, and cases
Probe Tool (Used for debugging)		Create probes on wires and display intermediate values on a wire in a running VI
Get Color Tool		Copy colors for pasting with the Color Tool
Coloring Tool		Set the foreground and background colors

# Short-Cuts that you must know!

Short-Cut	Description
<b>Ctrl + B</b>	Deletes all broken wires in a VI
<b>Ctrl + .</b>	Stops the Running VI
<b>Ctrl + E</b>	Toggle between the Front Panel and Block Diagram
<b>Tab</b>	Cycles through the most common Tools (Automatic Tool Selection should be disabled!)
<b>Ctrl + Mouse wheel</b>	Scrolls through subdiagrams in Case, Event or Sequence structures
<b>Ctrl + H</b>	Displays the Context Help window
<b>Ctrl + Mouse Double-click on a SubVI</b>	Opens the Block Diagram directly
<b>Ctrl + Arrows (→←↵↓)</b>	Move faster. You first have to select a SubVI, a Function, Object, etc
<b>Ctrl + W</b>	Close the SubVI
<b>Double-click on a wire</b>	Selects the hole wire

**Ctrl + drag objects with your mouse:** Copy objects (easier than Ctrl + C and Ctrl + V)



Do you need more Practice? - Select a Challenge

Hans-Petter Halvorsen





Note! This is just an example to illustrate the assignment. You can create your own User Interface with your own features. The code should include a While Loop, Case structures, SubVIs, etc. You can use the built-in Random Generator when playing Lottery.



# Create a Lottery App in LabVIEW

Below we see a sketch:

Creating this App makes you practice on basic LabVIEW Programming

Select your Numbers:

#1	#2	#3	#4	#5	#6	#7
33	4	8	9	8	11	20

+ Add

Lottery Ticket:

	#1	#2	#3	#4	#5	#6	#7
1.	33	4	8	9	8	11	20
2.							
3.							
4.							
5.							
6.							
7.							
8.							
9.							
10.							

Play Lotto

NORSK TIPPING

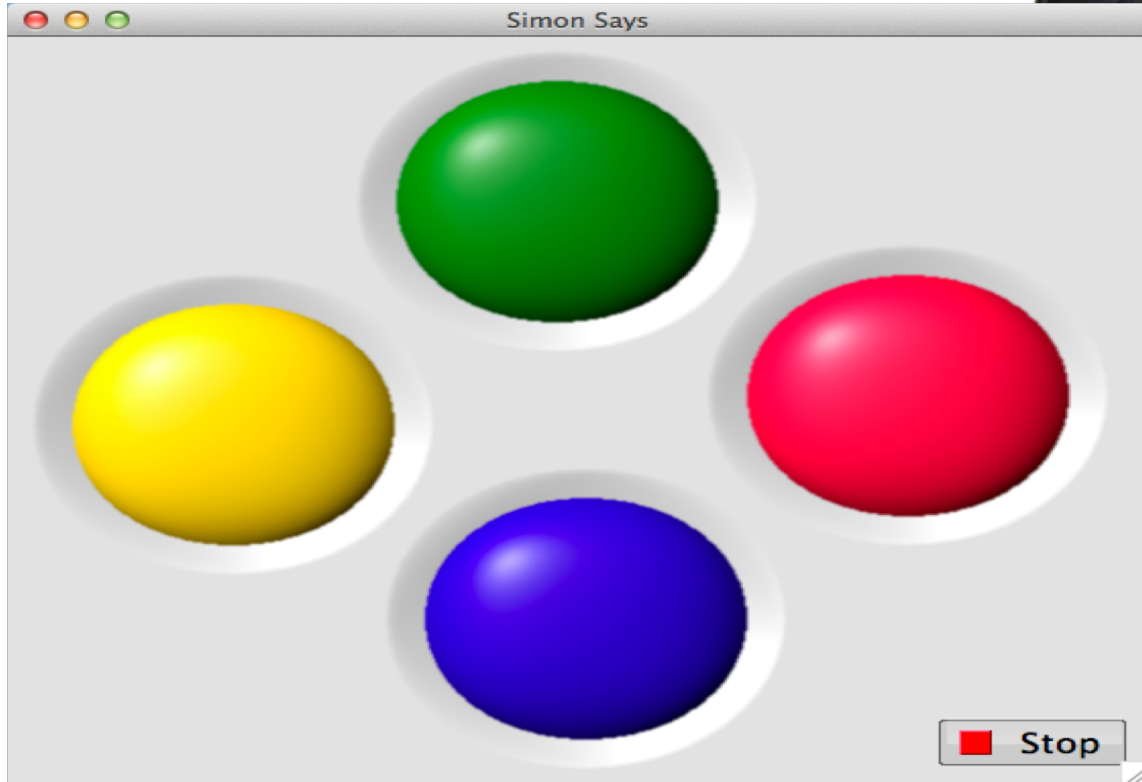
Lottery Results

#1	#2	#3	#4	#5	#6	#7
33	4	25	9	34	11	20

Your Score:

7 Correct: 0    6 Correct: 0    5 Correct: 2

# Simon Says



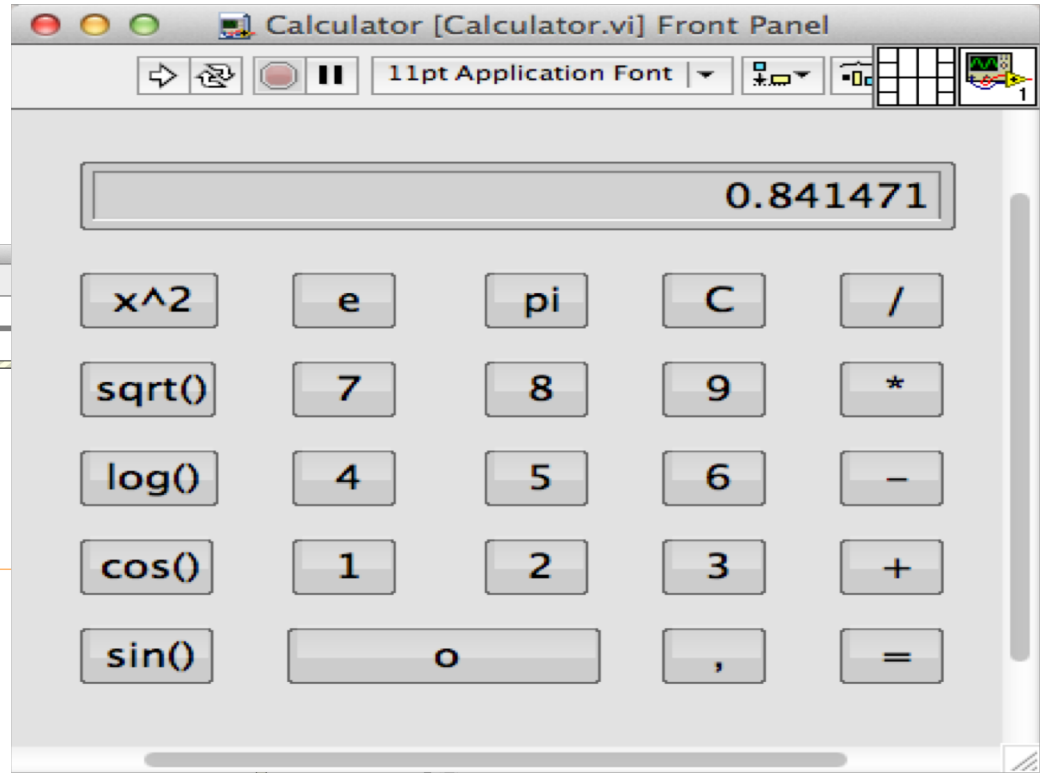
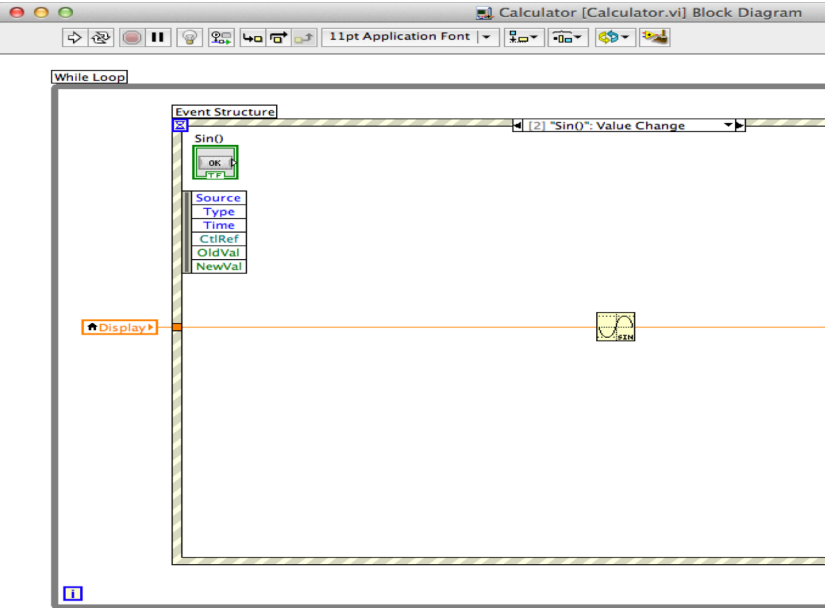
Creating this App makes you practice on basic LabVIEW Programming



# LabVIEW Calculator

3

Tip: Use an “Event Structure”  
inside a While Loop



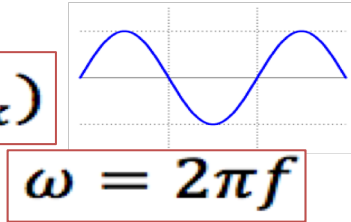


# Level Tank

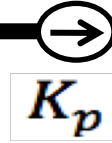
Assignment: Control the Level (close to the setpoint) in the Tank manually by adjusting the outflow of the tank with a manual Valve. Plot Level and Flow.

Flow into the Tank:

$$F_{in}(t_k) = K_p \sin(\omega t_k)$$



Pump

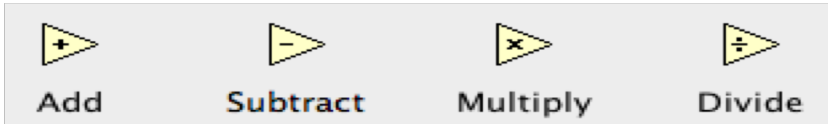


Level formula:

$$h(t_{k+1}) = h(t_k) + \frac{T_s}{A} [F_{in}(t_k) - F_{out}(t_k)]$$

Previous Level

Implement the Equations using standard LabVIEW operators like:

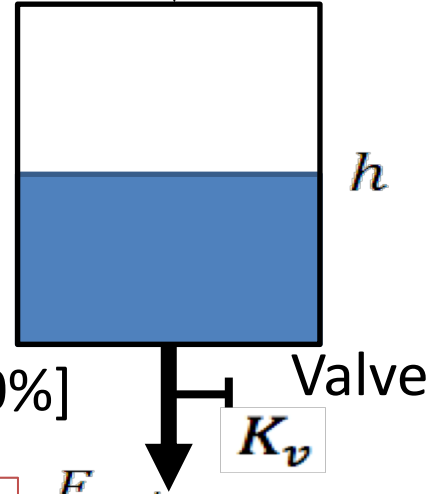


etc.

Using SubVIs are recommended!

Flow out of the Tank:

$$F_{out}(t_k) = K_v u(t_k)$$



u [0%, 100%]

Valve

# LabVIEW Implementation Example

$$h(k) = h(k-1) + (Ts/A)*(Fin - Fout)$$

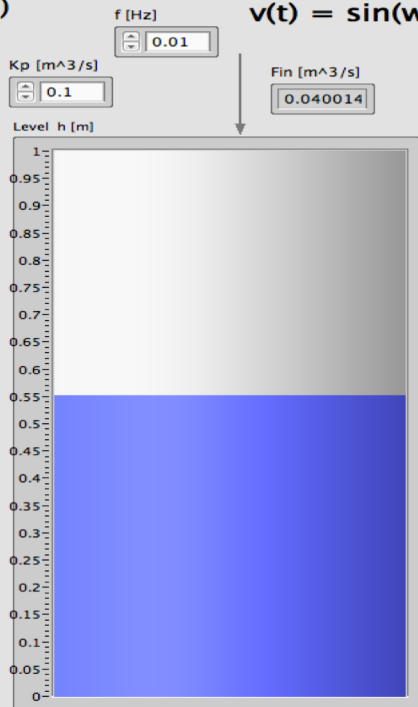
$$Fin = Kp*v(t)$$

$$v(t) = \sin(w*t)$$

$$w = 2*pi*f$$

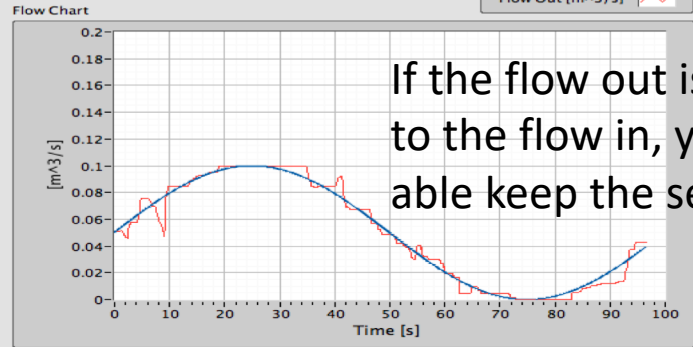
$$V = A * h$$

$$A = pi * r^2$$



Control parameters for the flow chart:

- Ts [s]: 0.1
- Flow In [m^3/s]: 0.04
- Flow Out [m^3/s]: 0.04

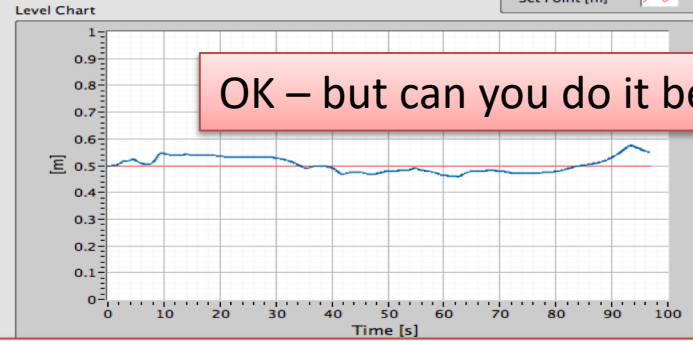


If the flow out is equal to the flow in, you are able keep the setpoint

Initial Level [m]: 0.5

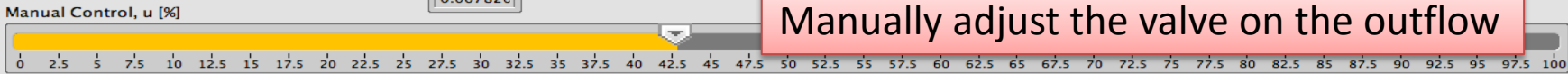
Control parameters for the level chart:

- Water Level [m]: 0.55
- Set Point [m]: 0.50



OK – but can you do it better?

Manually adjust the valve on the outflow



# Hans-Petter Halvorsen

University of South-Eastern Norway

[www.usn.no](http://www.usn.no)

E-mail: [hans.p.halvorsen@usn.no](mailto:hans.p.halvorsen@usn.no)

Web: <https://www.halvorsen.blog>

