LabVIEW Tutorial

Jessica A. Chani

jessica.chani@chalmers.se

Useful Material

NI LabVIEW webcasts:

- Gratis LabVIEW-utbildning online för studenter och lärare (90 min) (**Recommended!**)
- Introduktion till grafisk programmering med LabVIEW (35 min)

both available at: **http://sweden.ni.com/webcasts**

You may have to create an account to access to them.

Useful Material

LabVIEW Basics

- http://www.ni.com/academic/students/learn-labview/
- http://www.ni.com/getting-started/labview-basics/

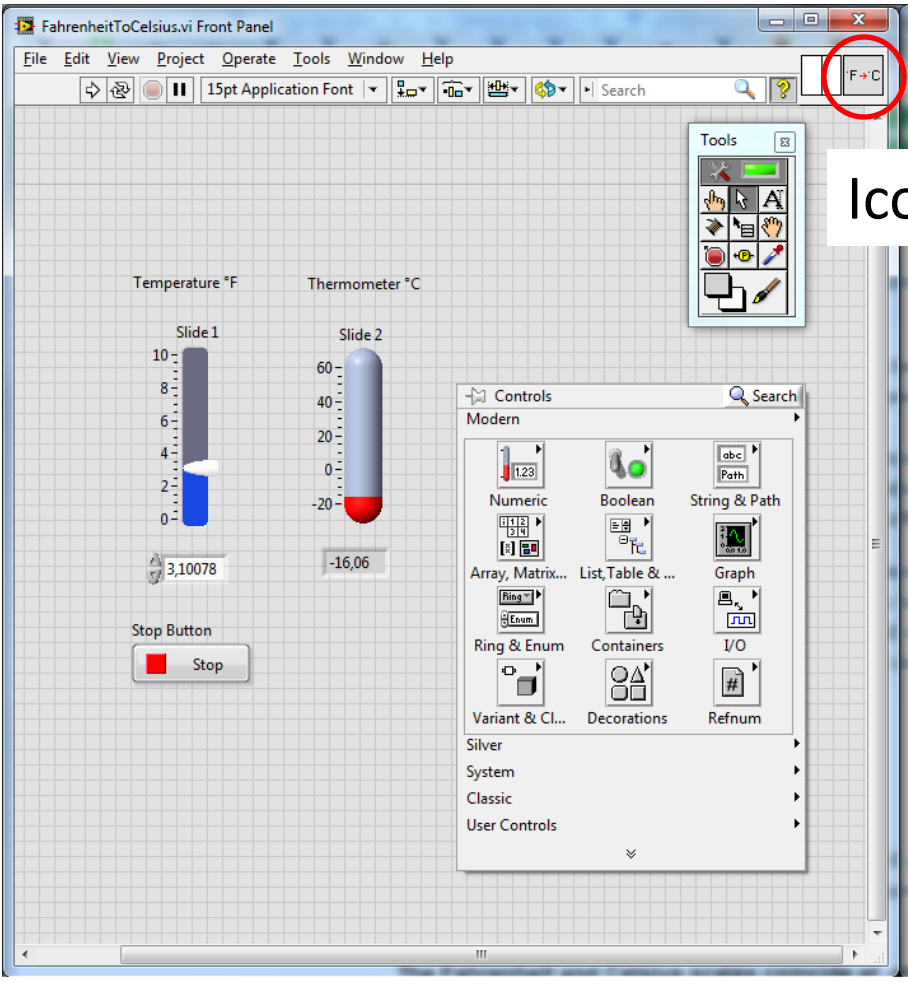
Outline

- Environment basics
- Dataflow programming basics
- Debugging tools
- Shortcuts

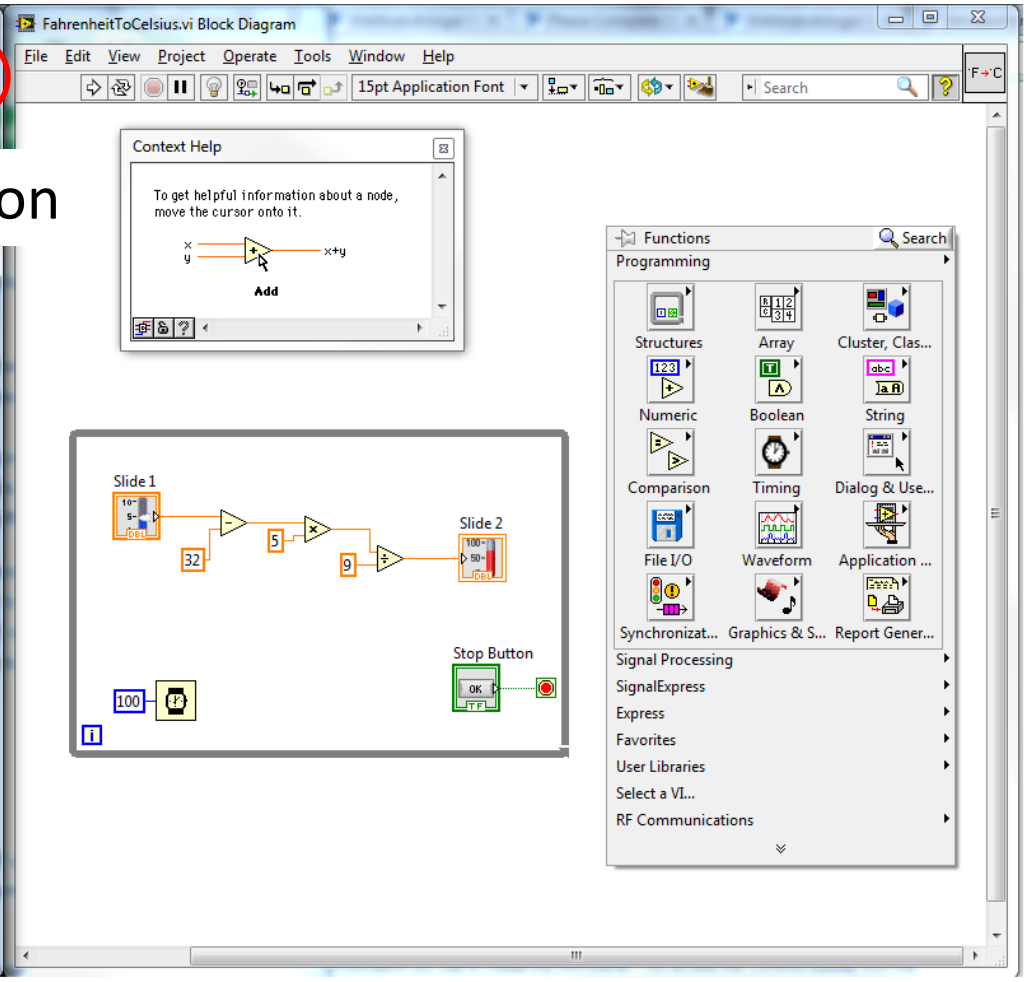
LabVIEW

- **Lab**oratory **V**irtual **I**nstrument **E**ngineering **W**orkbench
- Graphical programming environment
- Programming language is named **G**
- A program in Labview is called a **VI** (Virtual Instrument)

Virtual Instrument



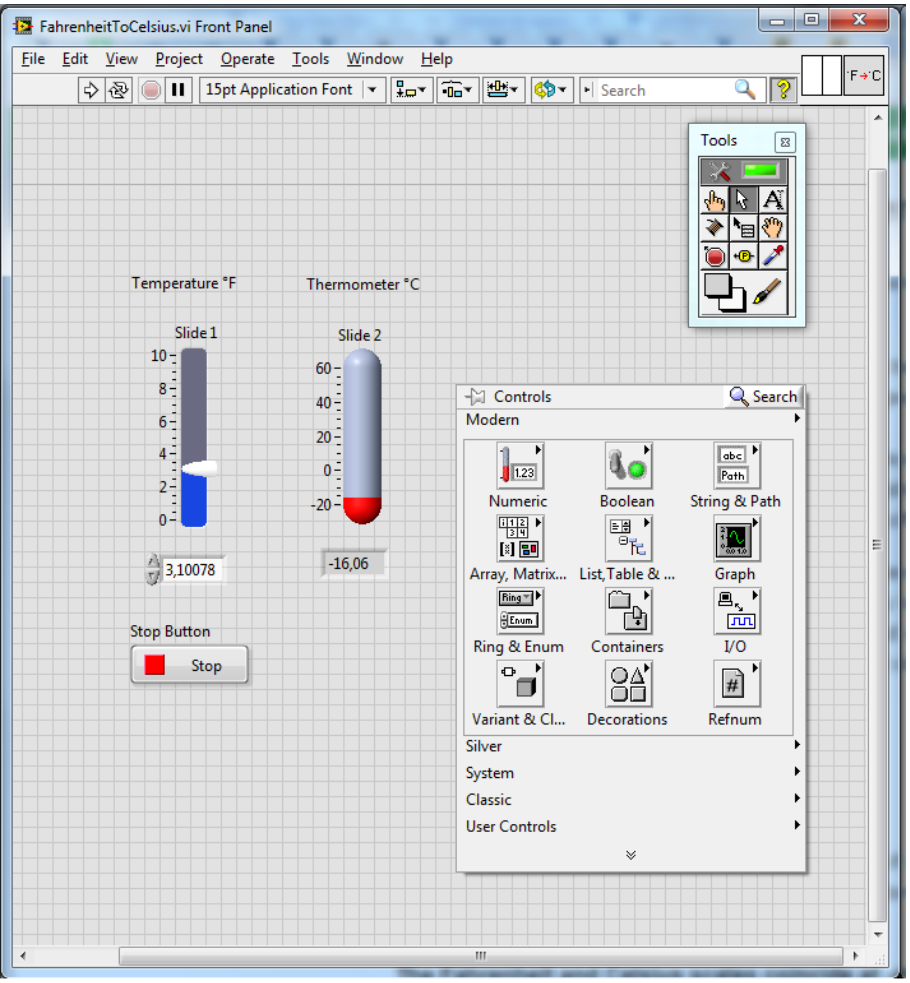
Icon



Front Panel

Block diagram

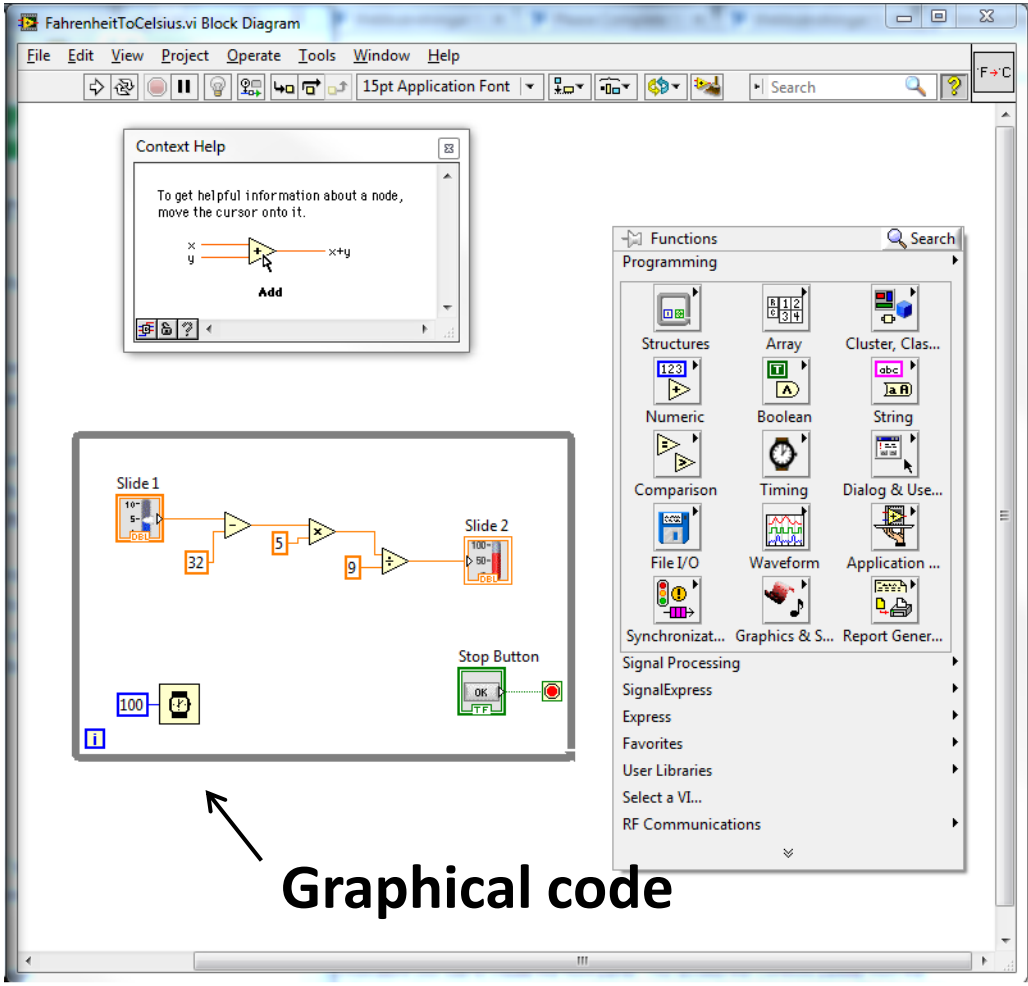
Front Panel



- User interface!
- Built using **controls** and **indicators**
- **Controls** are the **inputs** to the VI
- **Indicators** are the **outputs** of the VI

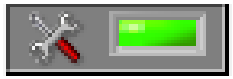
Block diagram

- Contains the **graphical code!**
- Block diagram objects include:
  - Terminals
  - Sub VIs
  - Functions
  - Constants
  - Structures
  - Wires



Tools palette

- Accessible from the **front panel** and **block diagram**
- **View>>Tools Palette**



Automatic Tool Selector

Operating Tool

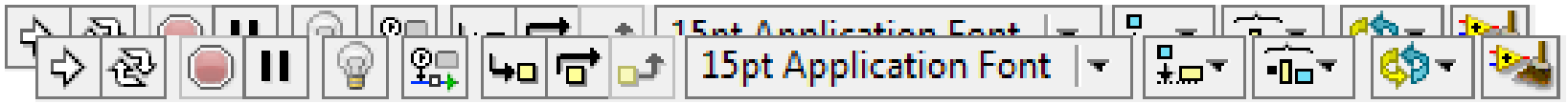
Positioning Tool

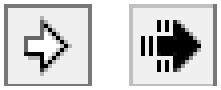
Labeling Tool

Wiring Tool

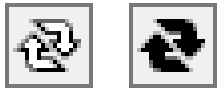
Coloring Tool

Status bar





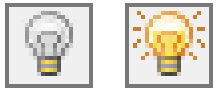
Run Button



Run Continouosly Button

Abort Execution

Pause Execution



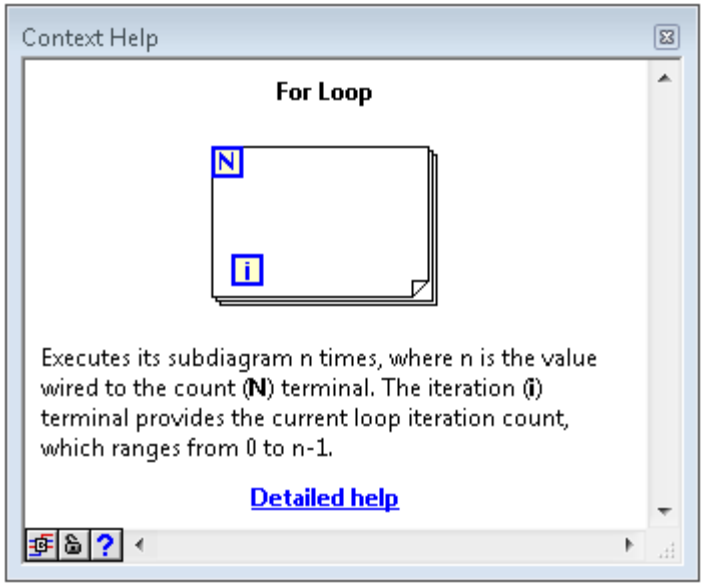
Execution Highlight Button

Step Function Buttons

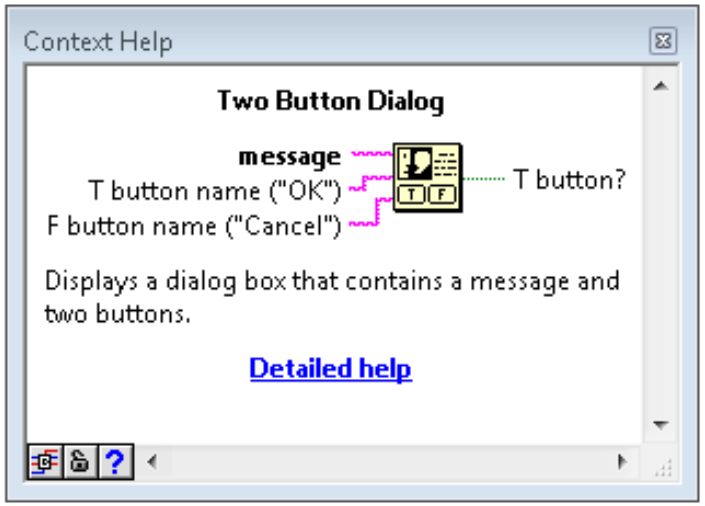
Automatic Object Aligning and Distributing

Getting Help

- **Context Help**
  - Displays **basic information** about LabVIEW objects when you **move the cursor over the object**.
  - Show/Hide **<Ctrl+H>** (**Recommended!**)



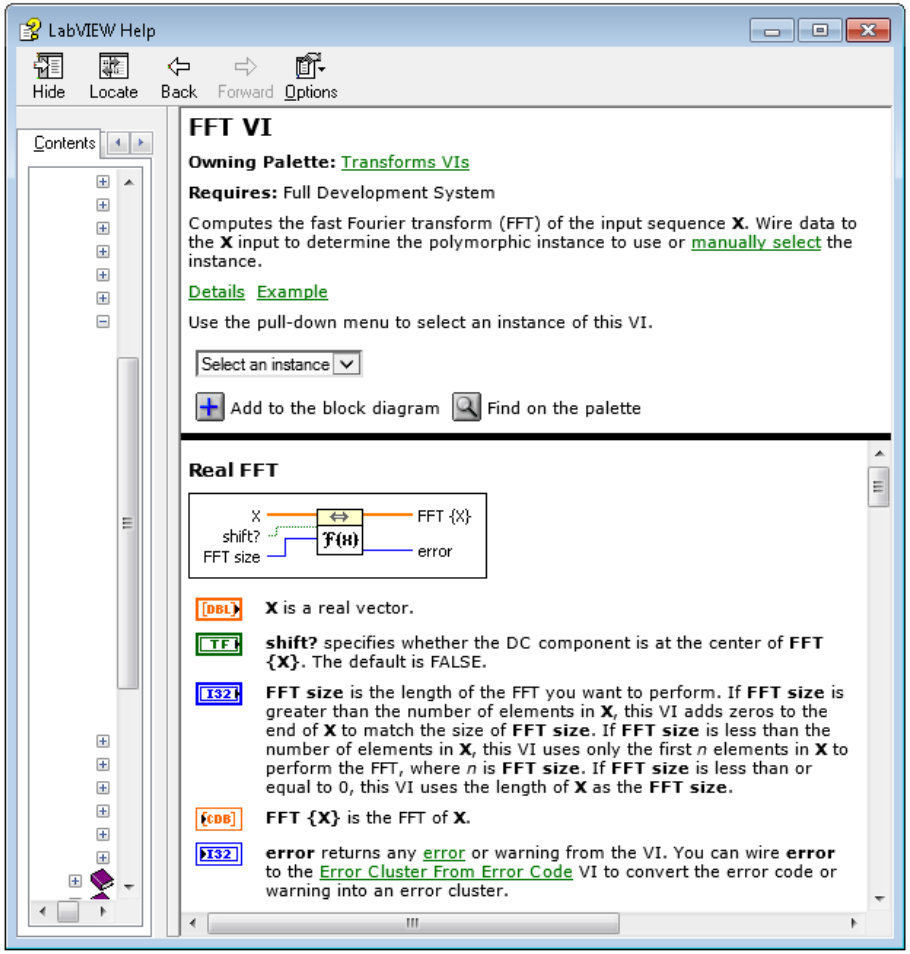
Context help of a **For Loop**



Context help of a **Two Button Dialog**

Getting Help

- **LabVIEW help**
  - **Best source** of **detailed information** about specific features and function in LabVIEW
  - Access by going to **Help>>LabVIEW Help** or by clicking the **blue Detailed Help link**

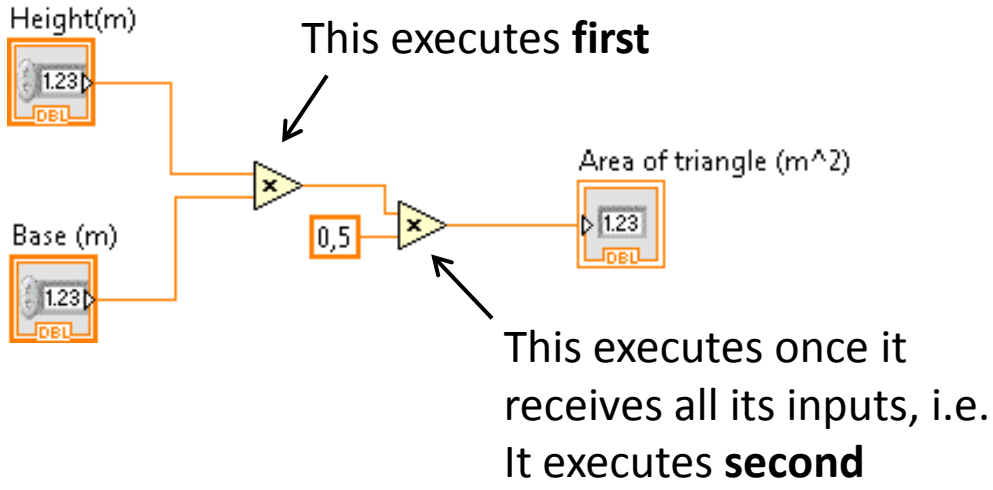


Outline

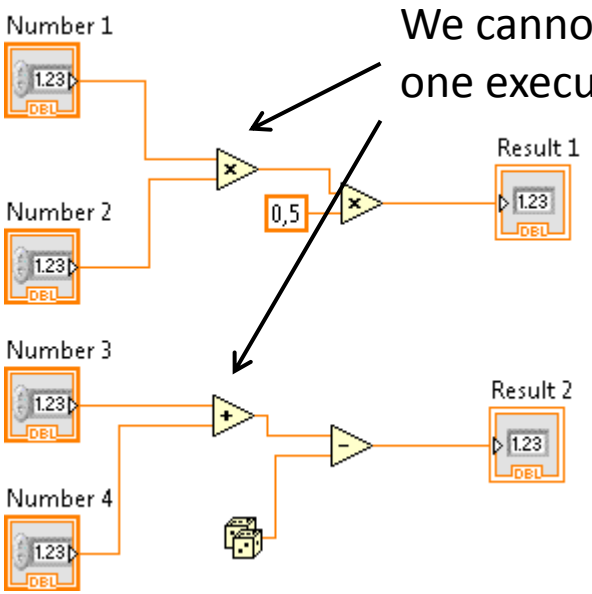
- Environment basics
- **Dataflow programming basics**
- Debugging tools
- Shortcuts

Dataflow programming basics

- A block diagram node executes when it receives all required inputs
- When a node executes, it produces output data and passes the data to the next node in the dataflow path



Dataflow programming basics



We cannot know which one executes first.

The inputs to  and  are available at the **same time**!

- If we must execute a code segment before another and no data dependency exists, we must use a **sequence structure** to **force the order of execution**. (We will discuss this in the next tutorial)

Wires

- **Transfer data** among block diagram objects through wires.
- Different **colors, styles and thickness**, depending on their **data types**

Wire Type	Scalar	1D Array	2D Array	Color
Numeric				Orange (floating-point), Blue (integer)
Boolean				Green
String				Pink

Outline

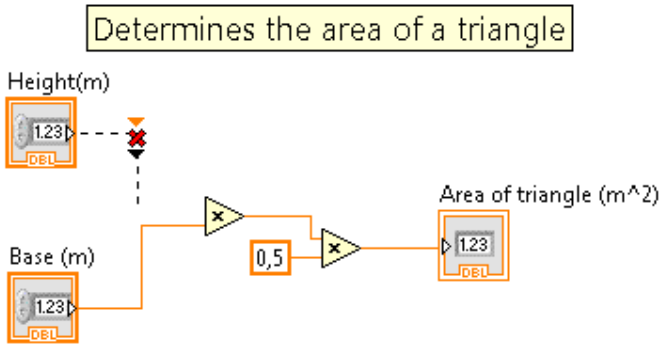
- Environment basics
- Dataflow programming basics
- **Debugging tools**
- Shortcuts

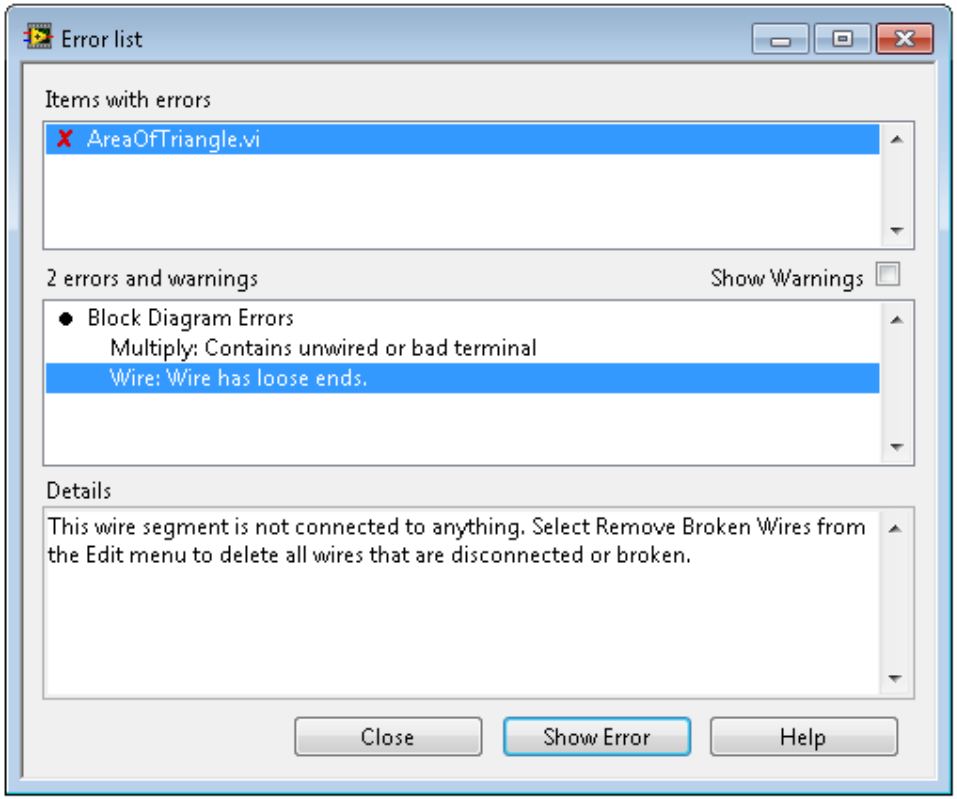
Debugging

- Two types of bugs
  - Those that prevent the program from running
  - Those that generate bad results

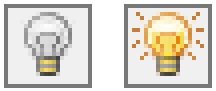
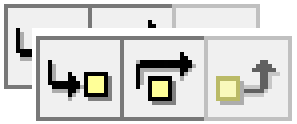
Debugging

- If LabVIEW **cannot run**, it informs you by changing the **Run arrow** to a **broken** one 
- **What to do?** Press the broken Run arrow to access the Error List Window





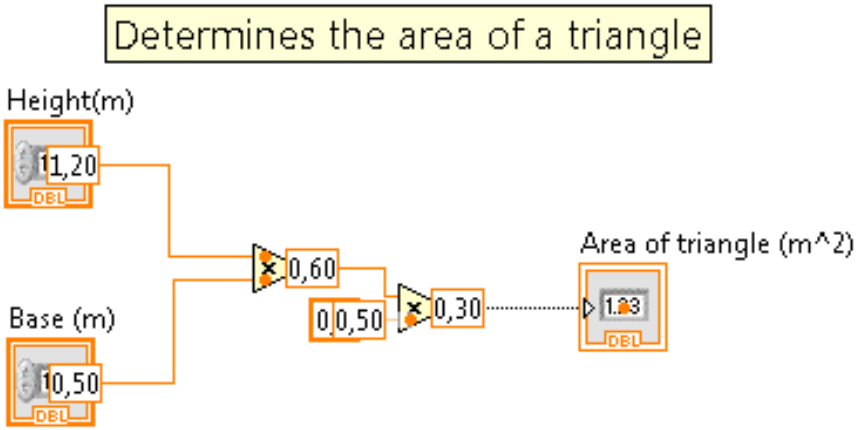
Debugging

- If your **VI runs** but produces **incorrect results**, then you have a **functional error** in your code
- **What to do?**
  - Use **Highlight Execution** 
  - Use **Step into, Step over and Step out** 
  - Use **Probe tool** 

Debugging

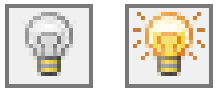
Highlight Execution

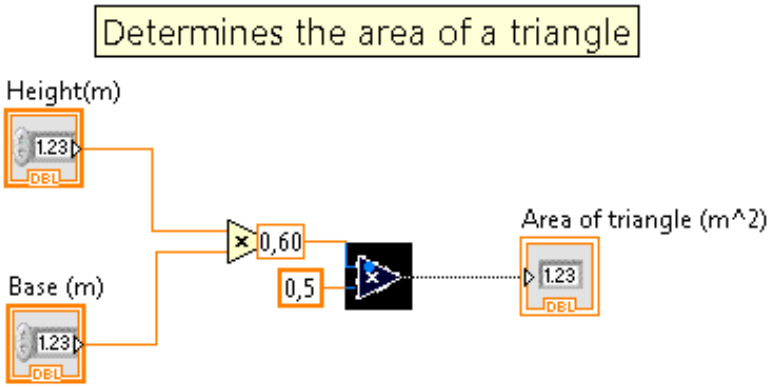
- Displays an **animation** of the block diagram when you run the VI.
- Shows the **movement of data** on the block diagram from one node to another.



Debugging

Step into, Step over and Step out

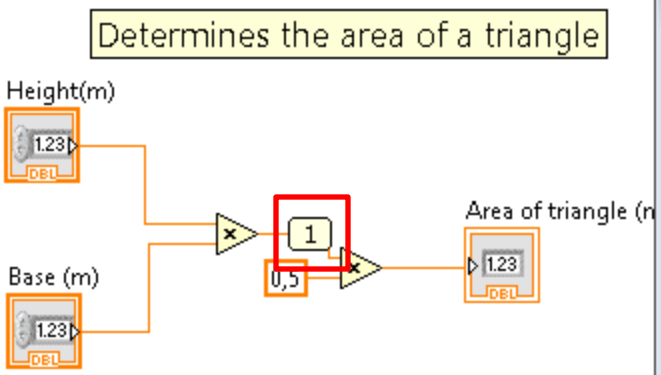
- Executes the first action and pauses.
- Use in combination with 

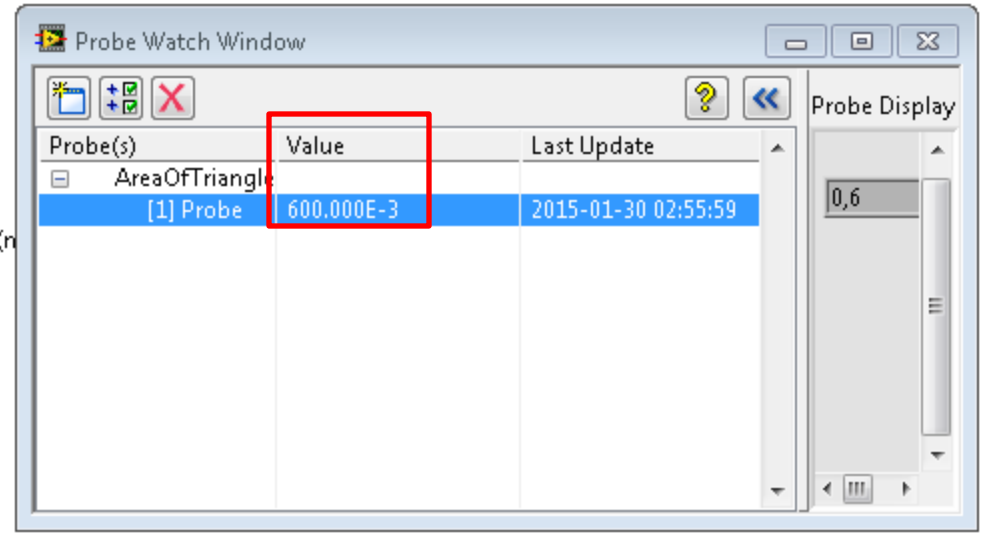


Debugging

Probe tool 

- Check **intermediate values** on a wire as a VI runs
- You can find the **probe tool** in the **Tools palette**
- Use in combination with    to **determine where the data is incorrect**





Shortcuts

Shortcut	Description
Ctrl+E	Toggle between front panel and block diagram
Ctrl+H	Display/Close **Context Help**
Ctrl+Z	Undo last action
Ctrl+R	Begins execution of VI
Ctrl+T	Tiles front panel and block diagram window
Ctrl+B	Delete broke wires