LET271: Elektriska mätsystem och mätmetoder

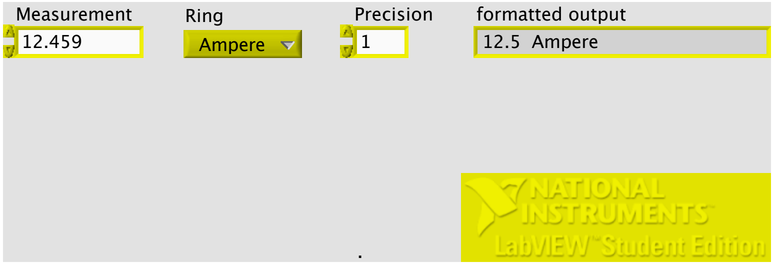
LabVIEW Exercises (Feb. 2016)

- The aim of this lab is to familiarize yourself with LabVIEW by solving some exercises.
- On the ping-pong course homepage, you find the LabVIEW User Manual and LabVIEW Getting Started by National Instruments. Consult it if you need additional information.
- I also recommend the NI LabVIEW webcasts:
  1. Gratis LabVIEW-utbildning online för studenter och lärare (90 min)
  2. Introduktion till grafisk programmering med LabVIEW (35 min)

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

Exercises

1. *Area of triangle:* Write a program computing the area of a triangle. The inputs are base and height. The output is the area. Try different controllers and indicator types.
2. *Led with boolean control:* Put three push buttons and a led on the front panel. The led should light up if both the first and the second push buttons are pressed or if the third is pressed. *Hint:* use the boolean functions *and*, *or*.
3. *Formatted measurement:* Write a program to format the result of a measurement. The program takes the measurement value as input, as well as the number of decimal units to be displayed and the unit of measure (which should be selected from a drop-down menú containing Volt, Ampere, and Ohm as options). The output is the resulting formatted measurement value followed by the unit as shown in the figure *Hint:* use *number to fractional string*, *pick line*, and a *ring control*.



Measurement: 12.459

Ring: Ampere

Precision: 1

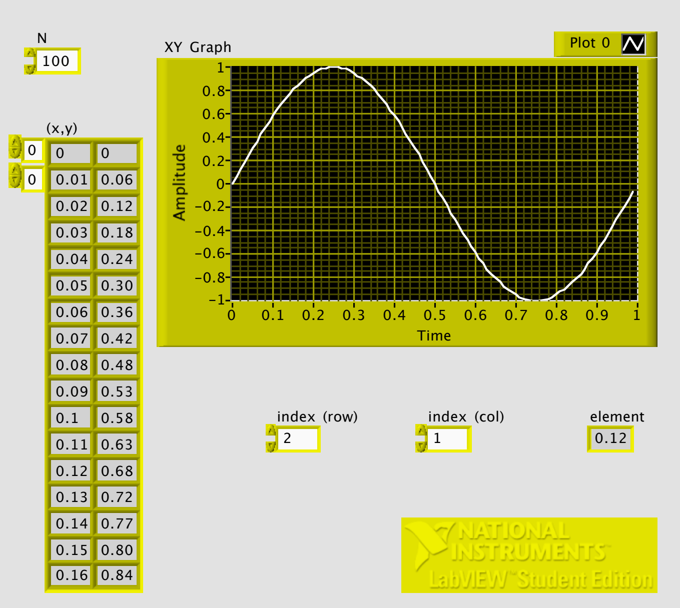
formatted output: 12.5 Ampere

Example of formatted measurement output (exercise 3)

4. *Boolean functions:* Write a program to verify the boolean functions and, *or*, *nand*, *nor*. The front panel should consist of a drop-down menú to choose the desired boolean function, two push buttons, and a led. The two push buttons should represent the input to the boolean function, and the led the output. A stop button should allow the user to stop the program. *Hint:* use a *while loop* and the *case structure*.

5. *Blinking led with timer:* Write a program to make a led blink at regular intervals. When started, the program should ask the user to select a speed: *fast*, corresponding to a state change every 0.25 seconds, and *slow*, corresponding to a state change every second. The elapsed time should be shown in the front panel. A stop button should allow the user to stop the program *Hint:* You may want to use *wait*, *two-button dialog*, *select* and a *shift register* in a *while loop*.

6. *Sinusoid:* Write a program that computes $y = \sin(2\pi x)$ for x between 0 and $N/100$, where the integer N is an input parameter, with steps of size 0.01. Display the resulting (x, y) values in a $N \times 2$ array and using an XY-Graph. The program should also take as input a row and column index of the $N \times 2$ array, and display the corresponding value. See the figure below. *Hint:* use *for loop*, *build array*, *auto-indexed tunnels*, *bundle*, and *index array*.



N: 100

(x,y)

0
0

0	0
0.01	0.06
0.02	0.12
0.03	0.18
0.04	0.24
0.05	0.30
0.06	0.36
0.07	0.42
0.08	0.48
0.09	0.53
0.1	0.58
0.11	0.63
0.12	0.68
0.13	0.72
0.14	0.77
0.15	0.80
0.16	0.84

index (row): 2

index (col): 1

element: 0.12

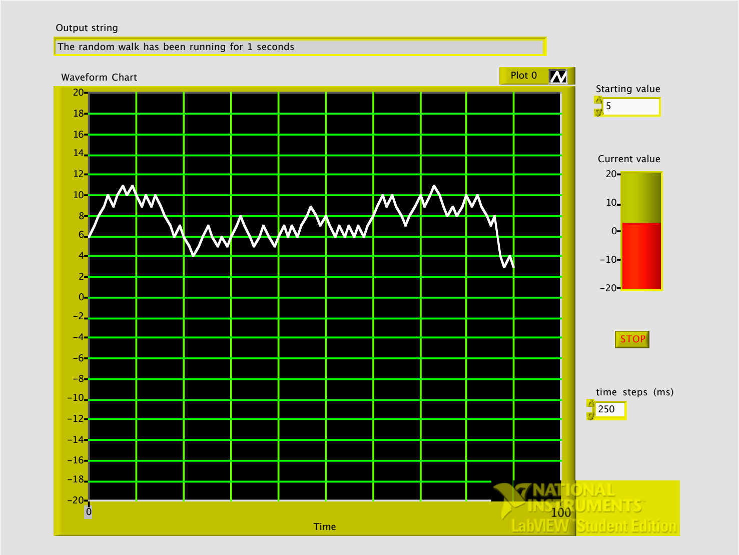
Front panel–exercise 6

7. *Multiple plots:* Write a program that computes the function $y = sin(x)e^{-x/10}$ for x between -5 and 15, with steps of size 0.1. Display the resulting function using

   - A waveform chart.
   - A waveform graph.
   - A XY graph.

   Which differences do you observe? Also save the resulting (x,y) values on a spreadsheet file. *Hint:* You may want to use *formula node*, *bundle*, *build array*, *write to spreadsheet file*.

8. *Random walk:* Write a program generating the random walk $x_k = x_{k-1} + c_{k-1}$ where the sequence $\{c_k\}$ consists of independent random variables that take value $+1$ and -1 with equal probability. The program should take as input the initial state of the random walk and the time interval (in millisecond) between each step. The output of the program consists of a *waveform chart* displaying the time behavior of the random process, a *tank* displaying the present value, and a string displaying the amount of



Output string

The random walk has been running for 1 seconds

Starting value

5

time steps (ms)

250

Front panel–exercise 8.

seconds elapsed from when the random walk was started. The front panel should approximately look like as the one in the figure

Hint: use *wait*, *number to decimal string*, *concatenate strings*. You also need a *while loop*.