Introduction to G-programming (LabVIEW programming)

(Lab. Exercise 1)

Sine wave generator

Introduction

Start LabVIEW and choose “Create Project” in the “Getting Started Window”. Double click “Blank Project”. You will now see the “Project Explorer” shown in Fig.1.

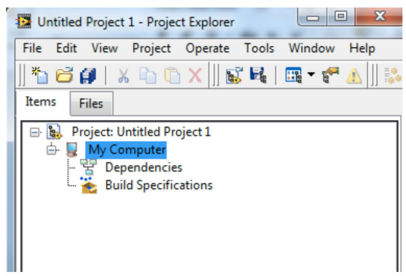


Figure. 1: The Project Explorer

Save the project as “lab_1.lvproj”.

In the “Project Explorer” choose “File/New VI”. The screen is shown in Fig. 2 with the two new windows “Front Panel” and “Block Diagram”The “Front Panel” constitutes the user interface with controls and indicators of different kinds, while the “Block Diagram” defines the program.

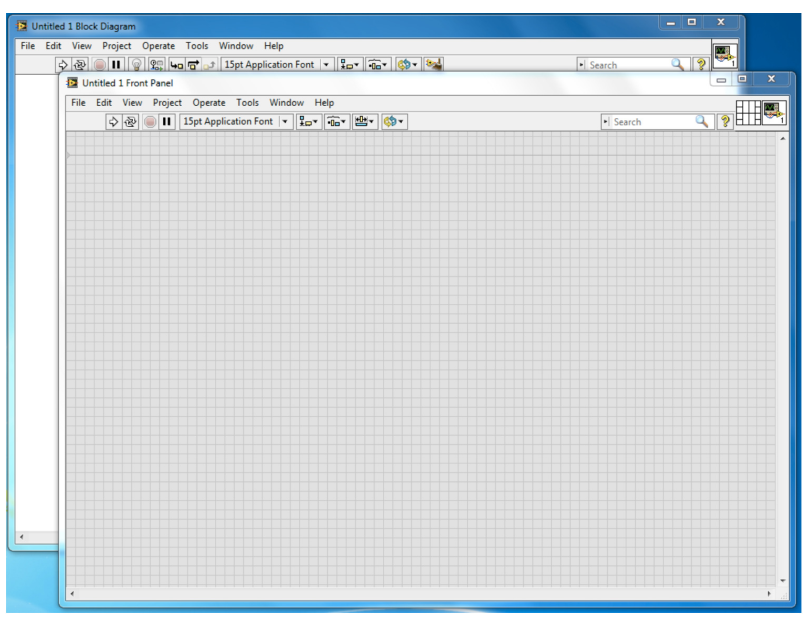


Figure. 2: Empty front panel and block diagram

Put elements in the front panel
Place controls and indicators in the front panel as shown in Fig.3. The elements that are found in libraries can be opened by right clicking the mouse when the pointer is placed above the front panel. Choose elements from the main library "Modern". "Phase", "Samples", "Amplitude", "Frequency" and "Sample Frequency" belongs to the type "Numeric Control", "Reset phase" belongs to type "Boolean Control" "XY-Display" is a "XY-graph" and "Signal out" is a "Cluster" of two "Arrays" of "Numeric Indicator". The cluster is built by first selecting a "Cluster" from "Array, Matrix, Cluster" then placing two elements of type "Array" from the same palette within the "Cluster" and finally putting a "Numeric Indicator" into each of the two arrays. Use "AutoSizing" "Size to Fit" to scale the cluster frame to the correct size. To do it, Right-click a cluster shell border and select "AutoSizing"→"Size to Fit" from the shortcut menu. If you have not yet written the names on the various controls and indicators, select "A" from the "Tools Palette" and double click in the "Name" field of each item: They will then be marked black and you can write the names as you see them in Fig. 3.

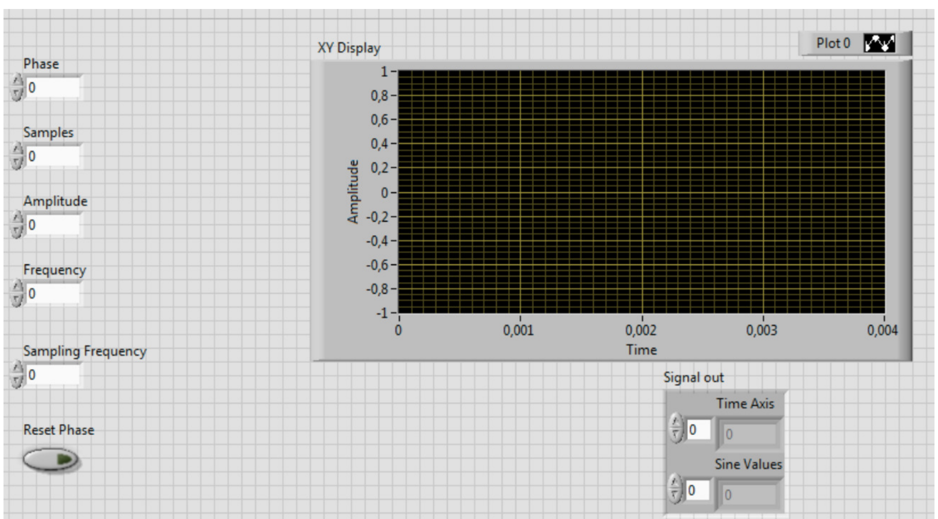


Figure. 3: Front panel of the sine generator

Formatting the elements on the front panel
All the numerical controls are initially of type float. The control named "Samples" should be changed to 32 bit integer (I-32). This is done by right clicking the icon and then selecting the submenu "Representation".

We want to maintain the type float for the controls "Frequency", "Sampling Frequency" and "Amplitude". In the controls "Frequency" and "Sampling Frequency", we will however only see the integer part, not the digits after the comma, and the "Amplitude" value should be shown with two digits after the comma (two digits of precision). All this is organized using the "Display Format" (Right click over the symbol. If it is the X axis that is to be adjusted, select "X Scale" and then "Formatting...").

In the "XY-Display", we want to see the samples presented as bars with a dot on the top and we should also see a solid curve between the samples, as shown below.

This is done by right clicking the dark area that shows the point and bar type on top of the main figure, and then choosing the sub-menus “Bar Plots” and “Point Style”.

The front panel (representing inputs and outputs of the program) is now finished and the next phase, the signal processing (program) can begin.

Construction of the block diagram (programming)

Select “show block diagrams” from “Windows” menu at the top of the front panel so that the window shown below appears.

You may want to open a “Help” window providing relevant descriptions of the various symbols (routines) used in this part of the process. The dynamic “Help” window is activated by selecting “Help”, “Show Context Help”.

In many cases more information can be found by clicking the “Detailed Help”.

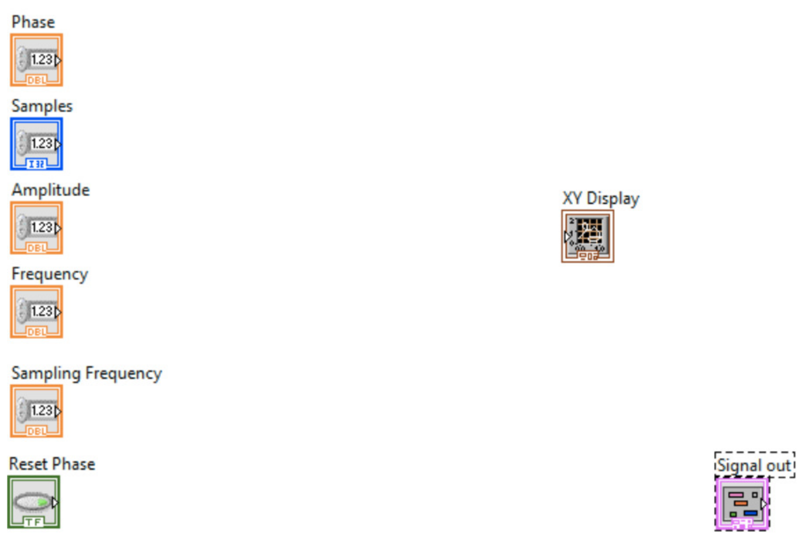


Figure. 4: Block diagram with controls and indicators from the front panel

The symbols shown in the block diagram represent the corresponding front panel input and output components, the colors indicate the types (orange = float, blue = integer, green = boolean, pink = cluster). If you double click on one of the symbols in one plane, you go to the other plane and see the icon there.

In the free space between the inputs and outputs in the block diagram, the program will be made. It will eventually appear as shown in Fig. 5.

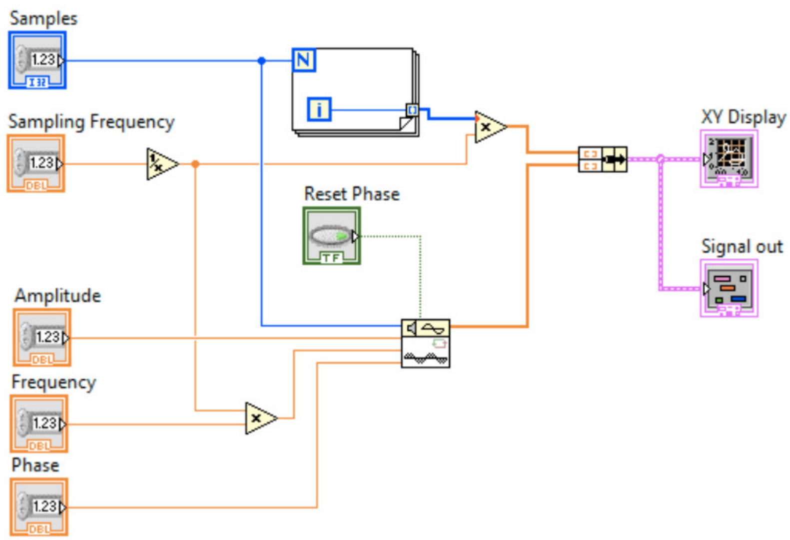


Figure. 5: The final block diagram of the sine generator

Retrieve function blocks
When you work with the block diagram you must see the menu “Tools” shown in Fig.6. Here you can choose a specific tool or let LabView guess what you need by selecting automatic.

Figure. 6: “tools”

Select the arrowhead as shown in Fig.6, then right click over the blank area in the block diagram and select “Signal Processing”→“Signal Generation”→“SineWave.vi” from the menu that appears. Clicking on this will bring it into the block diagram in the form of the symbol shown in Fig.7 a).

Figure. 7: “Sine Wave”(a), “Reciprocal” and “Multiply” (b), “For Loop” (c) and “Bundle” (d)

Use the same procedure to get the symbols “Reciprocal” and “Multiply”found in the “Programming” → “Numeric”library, “For Loopïn the “Programming” → “Structures” and “Bundleïn the “Programming” → “Cluster, Class ...”.

Defining the signal flow

Place the individual functions as shown in Fig.5. We shall now define the signal flow by making connections between the components. Click the spool icon in the left column of the “Tools” menu (or click the green field at the top to let LabView select the tool). The pointer (normally a cross) changes into a spool symbol when it is placed over a terminal point and if you then left click the mouse you can drag a wire to another terminal. The wire is terminated by another left click. Use this method to draw the wires shown in Fig. 5.

Different formats and types are indicated by colors and thicknesses of lines, for instance a thin orange line indicates a single float value, while a thick orange line indicates an array of float values.

The terminal “Signal Out” is at the moment probably a “Control”, but we will use it as an output and must therefore redefine it to an “Indicator”. This is done by right clicking the icon and select “Change to Indicator”. We have now finished the first program.

Save the program

To save the program choose “File”→ “Save as” “ov1_oscillator.vi” and store the program in an appropriate catalog.

Running the program

The program is now complete and can be run. Go to the front panel, select the finger icon in the “Tools” and select 33 samples, amplitude 1, frequency 300 Hz and sampling frequency of 8000 Hz. (phase can be 0 and the reset phase passive). The program is started by clicking the horizontal arrow on the top of the front panel (shown in Fig.8). The result (a sampled sine) can be seen in the XY graph. The X-values are so small that this axis should be ms resolution. If this is not the case, it can be changed by right-clicking over the XY-Display and then selecting “X Scale”→ “Formatting/ “Floating point” with three digit of precision.

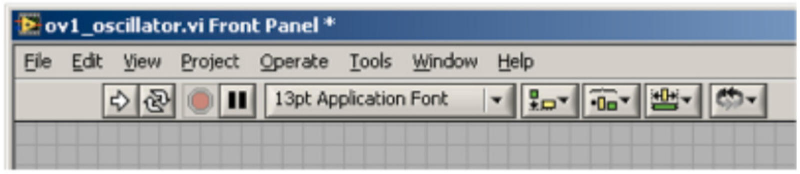


Figure. 8: The menu line at the top of the front panel

To see how the program is executed, you can light the lamp at the top of the block diagram (see Fig.9), click on the start arrow and look at the block diagram while the computations are carried out at a slow pace.

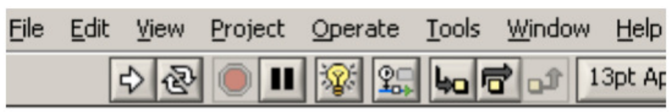


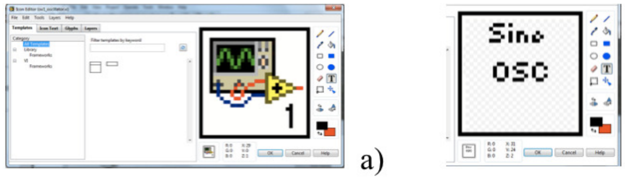
Figure. 9: The menu line at the top of the front panel, "highlight execution" is activated

Create a function block (symbol) representing the program

The program (the "VI" or "Virtual Instrument") which is now developed can be used as a function block in a program at a higher level. We must create a symbol for the "VI" Go to the front panel and right-click the block icon in the upper right, see Fig.10 and select "Edit Icon" to create a meaningful block symbol.

Figure. 10: The "Icon"for the VI (right) and its terminal map (left)

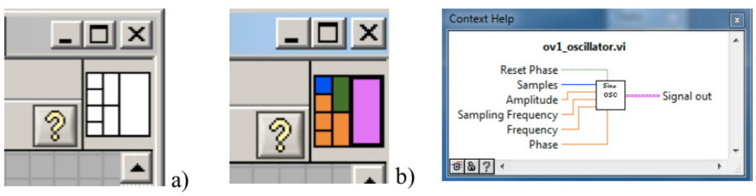
After selecting "Edit Icon" a window for editing the appearance of the block pops up, (shown in Fig.11. Remove the standard symbol by marking the area within the frame and press delete. Then choose A in the left menu field to be able to add a text (name) on the block, for example "Sine OSC" as shown in Fig.11b. Finish with OK, the symbol has now got the correct appearance, but the connections of inputs and outputs are missing yet.



a) b)

Figure. 11: Editing of the vi symbol

Right click on the terminal map in the upper right of the front panel and select "Patterns". Select the pattern shown in Fig.12a.



a) b) c)

Figure. 12: Connecting inputs and outputs to the vi symbol

Now use the spool tool to connect the front panel controls and indicators to the terminal map as follows:

Signal out: Click the large field to the right in the terminal map, then click on the "Signal Out" indicator. They are now connected.

Repeat for the other terminals as folows:
Reset phase: the upper middle
Phase: the lower middle
Samples: top left

Then in the order from left top to bottom: Amplitude, Sampling Frequency and Frequency.
The connection pattern now appears as shown in Fig.12b and in the "Context Help" it looks like in 12c. The symbol is now complete and can be stored.

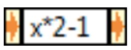
Create a program with sub-VIs
Now you shall create a new program (new VI) using the sine generator VI as function blocks.

In the "Project Explorer" select "File"→ "New VI". You will now get a blank VI.

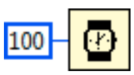
Place the block "ov1_oscillator.vi" as a block in the empty block diagram. (It is found under "Functions"→ "Select a VI"). The block "ov1_oscillator.vi" is the one you just created.

A quick way to create controls and indicators on the front panel is to point the spool at the connection pads of the blocks, then right-click and choose “Create”→“Control” or “Create”→“Indicator”.

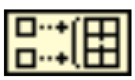
Complete the block diagram shown in Fig.13. (The XY graph must be added to the front panel as you did when you created “ov1_oscillator.vi”.). You will need a few new symbols:



This is an expression node in this case the output equals to two times of the input minus one.

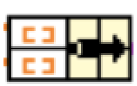


This function will make the loop wait for 100ms before the next execution.



This block builds a multidimentional array, in this case 2 cluster arrays in parallel. This will result in two curves in the XY window. To get it, select “Array” and then “Build Array”. Then extend (drag) the symbol vertically to achieve two inputs.

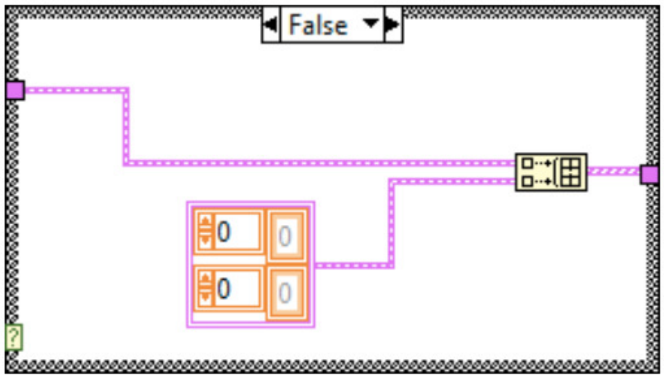
This is unbundle. It can be accessed by “Cluster, Class..” then “Unbundle”.



This is bundle. It can be accessed by “Cluster, Class..” then “Bundle”.

The frame surrounding the block diagram is a while loop controlled by the “stop” control bottom right.

The small frame within the while loop is a case structure with case “true” shown. Case “false” should look like this:



The symbol on the left hand side is a two dimentional empty array constant used as a place holder for the empty (not used) second curve on the XY graph. To make this module, go to “Cluster, Class..”→“Cluster Constant”. When the cluster constant is done, we need to put two array constant inside the cluster by selecting “Array” and then “Array Constant”. Thereafter, we should put a numeric constant into each array constant by selecting “Numeric”→“Numeric Constant” and then drag them into the array. Then change the data type to double by right click the numeric constant, then “Representation”→“DBL”.

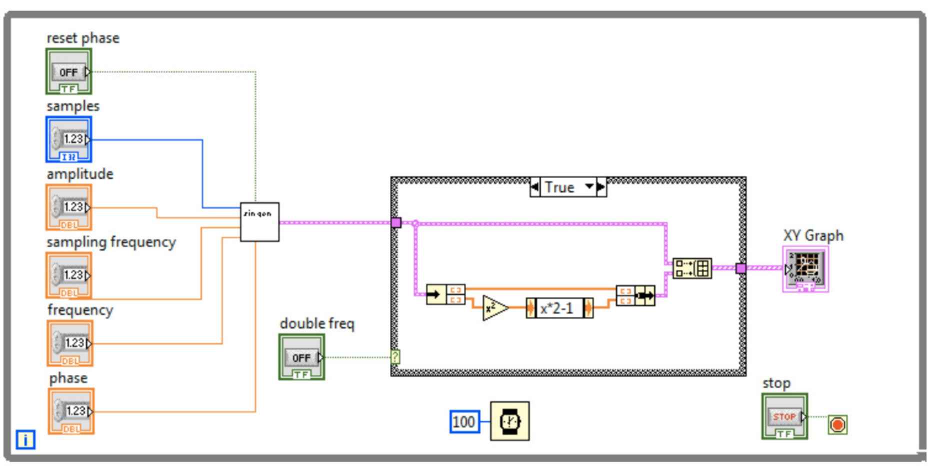


Figure. 13: Block diagram of the VI with sub-VIs

The final front panel should look as shown in Fig.14, the program is now complete and can be stored (save as “Lab_1.vi”).

When the program is started, it will run continuously until it the “Stop” button is pressed because of the while loop has the condition “Stop if True”. (This can be changed to “Continue if True” by

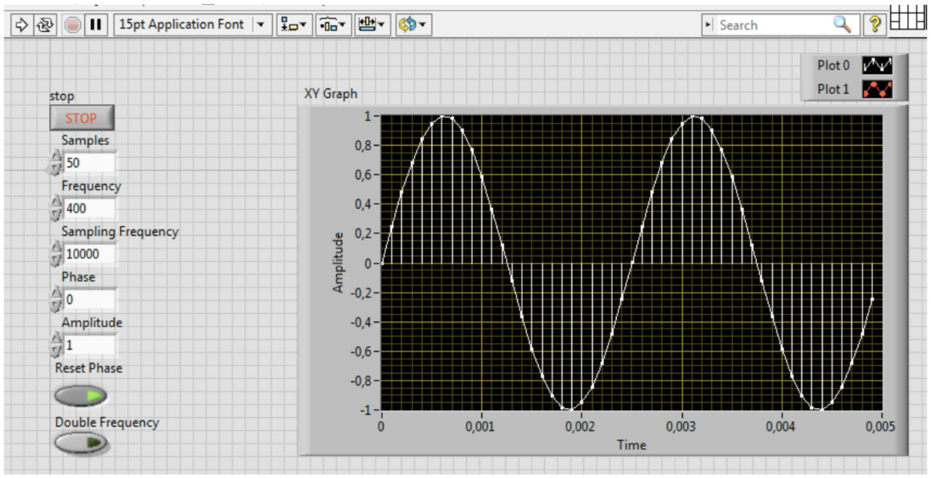


Figure. 14: Front panel of the final VI

right-clicking over the "Conditional Terminal" in the lower right corner of the while loop.)

Sampling and simple signal manipulation using the program

Sampling

Use the following input values: Samples: 50, Sampling Frequency: 10000Hz, Frequency: 400Hz, Double Frequency: off, Phase: 0, Reset Phase: on.

a) Find out what is the purpose of the "amplitude" control by changing its values.

   Set the amplitude to 1 and leave the rest unchanged.

b) What is the distance between the samples measured in time?
   (Observe and check with theory).

c) How many samples are there in one period of the sine signal?
   (Observe and check with theory).

d) Find out what is the purpose of the "Phase" control by changing its values.

   Change the frequency (the signal frequency) to 9600Hz, set "Phase" to 0 and keep all other values unchanged.

e) What is the distance between the samples measured in time now?
   (Observe and check with theory).

f) How many samples are there in one period of the sine signal?
   (Observe and check with theory).

g) What is the frequency and phase of the signal you see in the graph?
Explain the observation.

Change the frequency (signal frequency) to 10400Hz and keep all other values unchanged.

h) What is the distance between the samples measured in time now?
(Observe and check with theory).

i) How many samples are there in one period of the sine signal?
(Observe and check with theory).

j) What is the frequency and phase of the signal you see in the graph?
Explain the observation.

Use the following input values: Samples: 50, Sampling Frequency: 10000Hz,
Double Frequency: off, Reset Phase: off, Amplitude = 1, phase = 0.

k) Adjust the frequency around 400Hz and explain what you observe.

l) Set Reset Phase as on and explain its function.

Simple signal manipulation

Use the following input values: Samples: 100, Sampling Frequency: 10000Hz,
Double Frequency: on, Reset Phase: on, Amplitude = 1, Phase = 0.

a) Try different frequencies and observe how the two signals are related.

b) Show mathematically that the signal manipulation $y = 2x^2 - 1$ results in the new signal on the XY graph.

c) Is this a linear or non-linear manipulation? Justify your answer.

d) Change the amplitude to 0.5V and explain the result.