

---

# ITASDI PROJECT

## Innovative Teaching Approaches in development of Software Designed Instrumentation and its application in real-time systems

Erasmus+ KA2 2018-1-RS01-KA203-000432

---

### Course „Advanced LabVIEW Applications“

Laboratory no 6-7 - User Interface

---

**Leader Partner:** <sup>1</sup>Warsaw University of Technology

**Authors:** Dariusz Tefelski <sup>1</sup>

**Circulation:** Public

**Version:** 02

**Stage:** Final

**Date:** 10.09.2019

### Funding Disclaimer

This project has been funded with support from the European Commission. This communication reflects the views only of the authors, and the Commission cannot be held responsible for any use which may be made of the information contained therein.



# 1 Basic exercises

## 1.1 Removing the border from clusters in front panel

The cluster is a useful type of data, because it groups the different controls or indicators in one object. Sometimes it is useful to group controls/indicators in cluster but we do not want to show user the border of cluster on front panel. This exercise shows how to remove this border:

- Open the blank VI.
- Create the cluster from the classic palette with numeric control and led controls.
- Chose *View* → *Tools Palette*.
- Chose the transparent color in color picker window.
- Change the color of border of cluster to transparent.
- Create the Tab Control (*Modern* → *Containers*).
- Move the cluster to Tab Control and change the color of cluster to be non visible.

## 1.2 Custom cursors

When you run the LabVIEW application, the default cursor is used. Sometimes you need to change the cursor i.e when something is processing you can use hourglass to show that task is not completed. During this exercise you will learn how to change the cursor:

### 1. Variant 1:

- Open the blank VI.
- Create the for loop in which the sinus function will be generated.
- Add delay to the for loop i.e 50 ms.
- Display the sine function outside the for loop on Waveform Graph or XY Graph.
- Add the *Set Cursor VI* inside the for loop and right-click on *icon* input of this function. Then choose *Create* → *Control*.
- Find the icon of hourglass. Remove the control with icons.
- Add the proper value to *icon* input of *Set Cursor VI* to display hourglass when sine function is generated.
- Set the default cursor after the for loop, when it will finish.

### 2. Variant 2:

- Download or create own cursor as "cur" file. You can find many online cursor generators but remember that file extension should be **.cur**.



- Change the code from previous point to replace hourglass icon with your own cursor. Use *Create Cursor From File* function to read your cursor from *cur* file. Than add cursor ID to *Set Cursor* function.

### 1.3 Custom control

Sometimes you would like to change the appearance of a control. You can easily customize it by doing:

- Open a blank VI and create slide from classic palette.
- Right-click on a slide and choose: *Advanced* → *Customize....*
- Choose *Window* → *Show Parts Window*. Here you can find all parts of a slide. Find the slider. Right-click on slider on control and choose: *Import from File* and choose your own image of slider. You can draw slider in some program or just download from internet.
- Change the style of scale and fill color of the slider. When you finish, choose *File* → *Save as*. Now you can use your new own slider.

### 1.4 Custom Run-Time Menu

Sometimes you may want to modify a default Run-Time Menu.

- Open blank VI and choose *Edit* → *Run-Time Menu*.
- Change Default to Custom Menu.
- Add the File option by clicking on plus sign and add text *File* to Item Name.
- Add the Exit option inside the menu File. Click on plus sign and then click on a right arrow. Modify the item name and choose *Item Type* → *Application Item* → *File* → *Exit*.
- Add the Info option inside the File option. Make sure that the Item Type is *User Item*. Save the menu.
- Create the code, where you will have the event structure inside the while loop. Create the event " *Menu Selection (User)*".
- On the left side you have the element *ItemTag*, which gives you the name of selected option in menu. Create the case structure. When the user choose the *Info* option, the window with information about the application should appear.

## 1.5 Hiding the scrollbar and menu

Sometimes it is good to hidden unnecessary elements like scrollbar, menu or toolbar. Now you will do it on the base of code from 1.2:

- Right-click on scrollbar and choose: *Create* → *Property Node* → *HorizScrollbarVisibility*. Change property node to write. Right-click on input to property node and choose *Create* → *Constant*. Choose option *Off While Running*.
- Hide the vertical scrollbar in the similar way as previously.
- Add a button to switch on/off the menu. Than add property node from *Functions* → *Programming* → *Application Control* → *Property Node*. Right-click on the property node and choose: *Select class* → *VI server* → *VI*. Choose the property node *Show Menu Bar* and change it to write. Connect button to property node.
- Add button to switch on/off tool bar and create property node in similar way as previously.

## 2 Advanced exercise

### 2.1 Goal

Create an application, which will allow you to control the lights on the stage.

### 2.2 General requirements

- The application should be hierarchical and scalable. Remember to use subVIs.
- Use the state machine design pattern with enumerated type control, queue or event structure.
- Avoid using the local, global or shared variables if it is possible.
- Close all opened references and handles.
- Application shouldn't crash. Inform the user about the errors using the error cluster or a dialog box.
- Remember to prepare well documented code. Especially remember about: labels on long wires, description showing in context help, tip strips of controls and labels of constant values.



## 2.3 Specific requirements

- The user should have possibility to add the parameters of lights like: color, wait time before shining starts and duration of shining.
- The user can write the chosen configuration to file.
- The user can load configuration from file.
- The files should be saved in automatically created folder and the files should have the same name as name of configuration (from string *Configuration name*).

## 2.4 Description of steps

1. Starting the application - When the application starts, all the front panel objects must initialize to their default states:
  - Wait time = 0
  - Duration time = 0
  - Color = black
  - Configuration name = ""
  - All buttons = false
  - All leds = false

Additionally, the control panel should be deactivated (dimmed).

2. Record - After clicking the record button, the control panel should activate and the user may fill in the parameters of lights.
3. Play - After clicking the play button, the sequence is display on Output Panel and the control panel is disabled.
4. Stop - After clicking the stop button, the display is paused.
5. Save - After clicking the *Save* button, the configuration is saved to csv file.
6. Load - After clicking the *Open* button, the configuration is read from file using specific configuration name and following this the configuration is loaded into the control panel.

**Please use the front panel from public folder.**