**LabVIEW 101 – Weeks 3,4**

Before class:

1. Download updates from <https://github.com/rizett/LabVIEW-101>

Required equipment:

Computer with LV & Zoom installed

Topics:

Review exercises

Timing

More front panel controls

Saving data

More structures and loops

Property nodes

Introduction to some common errors

**1. Review exercises from last week:**

Exercise 1: Use a random number generator to display values on the front panel. Include: a numerical control device (e.g. knob) on front panel and functions on block diagram to manipulate the random number (e.g. multiply, add etc.), one chart, one other numerical display (e.g. thermometer). Save-as “labview101\_week1\_exercise1”

Exercise 2: Add an LED display to the VI above. Illuminate when random number exceeds a user-defined threshold. Save-as: “labview101\_week1\_exercise2”

Exercise 3: save the function developed in exercises 1 and 2 above as a SubVI. Wire the inputs (numerical control) and outputs (final numerical signal and T/F). Save-as: “labview101\_week1\_exercise3\_subVI”.

Exercise 4: Open a new VI. Create a while loop and add the subVI created above (labview101\_week1\_exercise3\_subVI) within the loop. Plot the number generated against computer time of day or Julian day in the subVI and use a Boolean (T/F) control to select when new data values are plotted. Remember to also include front panel controls for the input information for the subVI. Save-as: “labview101\_week1\_exercise4”.

**2. Save labview101\_week1\_exercise4 into a NEW LIBRARY (LabVIEW1010\_library)**

1. Save As > Copy > Substitute for Original > New LLB

* Open the new LLB, and note how this approach DOES NOT save the subVI to the library as well

1. Save As > Duplicate Hierarchy > Select the *LabVIEW1010\_library*

* Open the LLB, and now note that the exercise 3 sub VI is saved.

**3. Timing**

* Get computer time and functions to split time
  + **Open a new VI**
  + From the Timing Palette (Block Diagram > Timing), add a “Get Date / Time String”
    - Add 2 string indicators on the front diagram, call them “date” and “time”, and wire them to the output of the Get Date / Time String function
    - Note that the output date format can be controlled with a numeric constant/control corresponding with different data formats, and the time format can be controlled with a Boolean constant/control.
    - You can also add a “Time Stamp Constant” function to format a specific data/time
  + Now add a “Get Date / Time in Seconds” function (block diagram)
    - This reads the current computer time and returns a timestamp
    - Inserting a “Format Date / Time String” allows us to read the time as a string (add another front panel string indicator) – note that this achieves the same as the exercise above.
      * Can also be formatted using a string constant (check context help for correct date / time formatting – and pay attention to capital vs lowercase letters! E.g. %0m vs %0M, %y vs %Y)
    - Or, we can unbundle different components of the time:
      * Add a “Seconds to Date/Time” function
      * Add “Unbundle” and “Unbundle by Name” (Cluster, Class, Variant palette) and wire to the output of the “Seconds to Date/Time”
        + Note that it is possible to unbundle each component of the date / time.
        + Can use this to, for example, count the number of seconds that elapse between two actions (e.g. use convert string to number)
  + Add the Julian day subVI (provided on GitHub at /Generic VIs and subVIs/Julian Day\_subVI)
    - Note the wired output
* Other timing functions:
  + Wait 🡪 pause for a user defined number of milliseconds
  + Tick Count 🡪 returns an milliseconds value (from an arbitrary reference point); can be used for determining timing between events
  + Wait until next MS Multiple 🡪 Add to the VI above to control when to display the
  + Elapsed time 🡪 Add to the VI above to control when to perform the set of functions. Use a conditional statement

**4. More front panel controls**

Open a new VI

* + Review: Combo box (string output), Text Ring, Menu Ring, Enum (all numerical output) 🡪 have pre-defined text values, corresponding with different cases / options.
  + Path (string & path)
    - Path indicators vs path controls
      * Explore properties > browse options
  + Adjusting appearance / output of numerical indicators (number of digits, type etc.)
    - Add a random number generator linked to a front panel numerical indicator
    - Adjust the number display style and run the VI

**5. String / number / path conversions and concatenating strings**

* + String to number (vice versa)
  + String to path (vice versa)
  + Combining strings
    - Note: may need to add adjoining string elements (like “/”, “\_”, etc.) as string constants

**6. Saving**

* + See GitHub / examples and functions / output examples
  + Open /examples and functions / examples/week3\_path-save.VI
  + File I/O palette:
    - Write to Measurement file:
      * Setup options
      * Link filename / path
      * Data columns (merge signals with sig. manip.)
      * Wire-in / out options
    - Save as Ascii
    - Save as CSV

Exercise 1: Open week 1 exercise 4 (from the Library). Add a **Wait** or **Wait until next MS Multiple** function within the while loop so that the data are plotted less frequently. Use a **front panel control** to set the frequency. Save-as “labview101\_week3\_exercise1” into the LabVIEW101\_library.

Exercise 2: From labview101\_week3\_exercise1, use a combination of string and path **controls** to set the location and filename of an output data file. Use the path control to select the folder to which the data will be saved (in the following exercise), and the string control to set the filename. Append a formatted date stamp to the filename/path, so that the data file is saved as: /path/to/outputdata/filename\_YYMMDD. Add a front panel path indicator to show where the data are saved. Be selective about **where** you build the output filepath (i.e. inside vs outside of the while loop), as this will affect the output filename. Save-as “labview101\_week3\_exercise2” into the LabVIEW101\_library.

Exercise 3: Add a save to measurement file function to the “labview101\_week3\_exercise2” VI . Run the VI to **manually** select the save file name / destination. Afterwards, use the file path created in the previous exercise to automatically set the filename of the output data file. Finally, modify the VI to include three saving options: Save to measurements file, save to text file, don’t save. Use a combo box/enum and case structure for this. Add a Boolean control to toggle data saving on/off in the first 2 cases (also best to use a local variable).

Save-as “labview101\_week3\_exercise3” into the LabVIEW101\_library.

(Likely week 4):

**7. More on Structures and timing**

* Timed structures: Loop, Sequence

**8. Property nodes**

* Use to modify display properties (e.g. control plot scales, plot colours etc.)
  + E.g. Change plot appearance / colours: Front panel > Numeric > Framed Color Box

Exercise: Add a property node to control the upper and lower y-limits on the plot in your VI. Use front panel controls to set these so that the values can be adjusted while the program is running.

Save-as “labview101\_week4\_exercise1” into the LabVIEW101\_library.

**9. Some common errors**

* Loading dependencies
  + Open XYZ example
  + Note the run-time error, because of the missing subVIs
  + This error is fixed by locating the missing subVIs in the block diagram, and reloading them into the VI.
  + This is why it’s useful to use libraries!
* SubVI greyed out – means it had been modified and needs re-configuring.
  + Open LabVIEW101\_library > labview101\_week2\_exercise3\_subVI.vi
  + Add a new (empty) wire terminal to the subVI output
  + Save and close the subVI
  + Open labview101\_week2\_exercise4.vi. Note the run-time error. This is because labview101\_week2\_exercise3\_subVI was modified, and is now incompatible with the way we intend to use it in exercise4.
  + There are 2 options:
    - 1) delete the old labview101\_week2\_exercise3\_subVI and re-insert the new version from the LabVIEW101\_library
    - 2) Right click > Relink

**10. Cleaning up a diagram**

* E.g. aligning elements

Exercise: *I’ll think of some good exercises using timed events and sequences.*