**LabVIEW 101 – Weeks 1,2**

Before class:

1. Download updates from <https://github.com/rizett/LabVIEW-101>
2. Install LabVIEW and Drivers (see below)
3. Download and Install RealTerm

* <https://sourceforge.net/projects/realterm/files/latest/download>
* You only need to install the “Main Section”

1. Download and install autoIt

* <https://www.autoitscript.com/site/autoit/downloads/>
* Scroll down a bit to download the “AutoIt Full Installation”
* Choose to “Run the script” when asked for a default option for .au3 files
* You don’t need to install examples, unless you would like to

1. (Recommended) Create a desktop folder for “LabVIEW 101” (or something similar)

* For many of the exercises, we’ll probably end up buddying-up, but to start, everyone will be “assigned” a single computer

Required equipment:

Computer with LV & Zoom installed

Topics:

Review topics from last week (shaded below)

More on:

* Variables and Displays
* Structures and Loops

Exercises

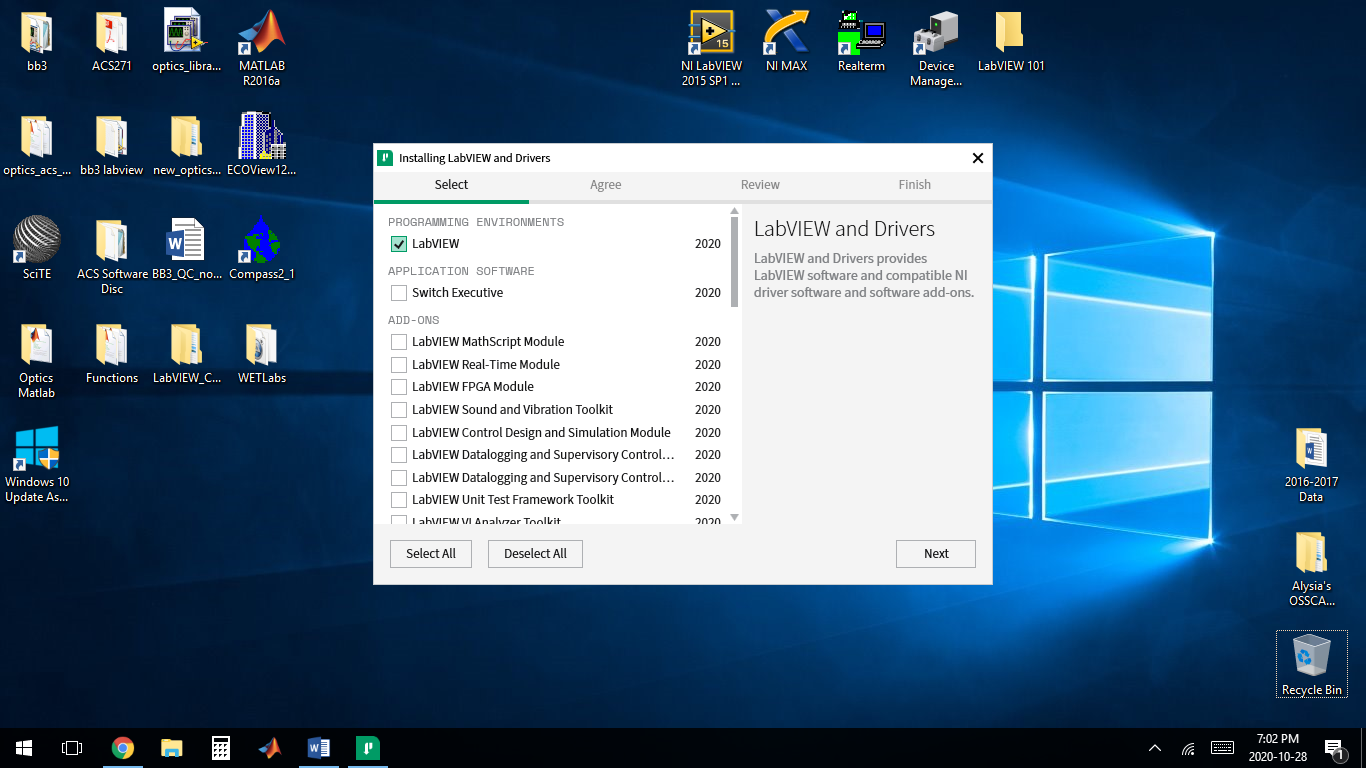
**Installing LabVIEW & Drivers**

1. Go to <https://www.ni.com/en-ca/support/downloads/software-products/download.labview.html#346254>
2. Login or create a National Instruments account.
3. Select OS, Version, Edition etc. I suggest installing the newest version, and no older than 2017. Select the 32-bit version.
4. Click “Download” to download the NI Package Manager.

Graphical user interface, application, website

Description automatically generated

1. Run the installer and accept the license agreements.
2. After the installer has initialized, select the packages to install. Select *LabVIEW*, the *Advanced Signal Processing Toolkit* and the *NI-DAQmx* and *NI­-VISA* drivers. Select any other drivers or add-ons that you desire. Click “Next”.



1. Select any additional add-ons to install. Select the *NI Certificates Installer*, *NI-DAQmx Runtime with Configuration Support*, *NI-DAQmx Support for LabVIEW 2020*, *NI-VISA Configuration Support*, *NI-VISA LabVIEW 2020 Support*, and any other desired add-ons. Click “Next”.
2. Accept the license agreements, and review the install programs / drivers. Click “Next” to begin the install, which may take a couple hours….
3. After the install is complete, launch LabVIEW (\Program Files (x86)\National Instruments\LabVIEW 20XX), and log into your NI account. Activate the UBC/EOAS license by entering the serial no. M62X63865.
4. You should be ready to go now! I recommend creating a desktop shortcut for the LabVIEW and NI Max (\Program Files (x86)\National Instruments\MAX) applications as well.

**Introduction to the virtual instrument / environment**

LabVIEW hierarchy:

Project > Library > VI > SubVI

* Project (for sharing): used to group LabVIEW files and files not specific to LabVIEW, create build specifications, and deploy or download files to targets. You must use a project to build applications and shared libraries.
* Library (for internal use): a collection of VIs, type definitions, shared variables, palette files, and other LabVIEW files.
* VI (virtual instrument): a graphical script of programs / functions / subroutines, consisting of a block diagram, front panel, connector pane.
* SubVI (a function): similar to a VI, but is called within a main VI

Examples:

Open MIMS, PIGI and Generic contents of LabVIEW-101/examples and functions (<https://github.com/rizett/LabVIEW-101/tree/main/examples%20and%20functions>) to see different examples of the components above.

* PIGI4.3\_Library\_2020.lproj = PIGI project
* PIGI4.3\_Library\_2020\_install.exe = installation package for runtime engine (no license required)
* PIGI4.3\_Library\_2020.llb = PIGI scripts library
* RUN-MIMS-LIBRARY.llb = MIMS scripts library

VI Components

Open RUN-MIMS-LIBRARY > MIMS\_scan\_2019.vi to explore the different parts of a VI:

* VI (virtual instrument) vs SubVI (~function)
* Front panel, wiring/block diagram (toggle between with Ctrl-E)
* Tools palette: how to show; what all the functions do
* Context help (Ctrl + H)
* Other view options
* Adding functions and components to diagram – drop-down menus and search function (front panel & block diagram)
  + Functions palette
* Local vs global variables (block diagram only)
* Loops and structures (block diagram only)
* Running options:
  + Run vs Run Continuously; Run errors (broken arrow)
  + Pause
  + Highlight
  + Stepping
* Stop, Booleans
* Edit options
  + Set current values as default
  + Cleanup diagram/selection
* Save options:
  + Save
  + Save As – copy or replace

**Variables and Displays (follow along on LV)**

Using a blank / new VI (File 🡪 New VI), we’ll go through the following:

* Naming components
* Adding comments; controlling text/font appearance
* Indicators vs controls & read vs write (front panel & block diagram)
* String, numbers palettes, enum (front panel & block diagram)
* True / False constants
* Numerical functions (multiply, divide etc.) and comparison (greater than; equal to) (block diagram)
* Wiring, including shortcuts (right click on wires)
* Simple graphs / plots: Chart, Graph, Express XY Graph (Build XY graph function; clear on each iteration)
  + See /examples and functions/examples/week2\_graphs-vs-charts.vi
  + Waveform graph: accepts multiple data types (array, waveform, dynamic data) and plots all received points at once; does not accept single values; unless specified, X data are indices starting at 0 and with step size of 1
  + Waveform chart: remembers and displays a certain number of points by storing them in a buffer; when the buffer is full, it begins to be over-written (i.e. waveform charts display new data, in addition to already-existing points); can accept single data points, arrays (including 2D arrays to display 2 plots)
  + XY Graph: allows user to specify X and Y data points. Can plot multiple lines/points by combining signals.
  + More information: https://knowledge.ni.com/KnowledgeArticleDetails?id=kA00Z000000P9zsSAC
  + Controlling plot appearance (plot colours, adding secondary axis, scales etc.)
  + Other numerical indicators and controls: dials, sliders, knobs
* Controlling appearances / some colours, containers, tab control etc.
* Combining / splitting signals (Express > Signal Manipulation)

**Stop button and Booleans (follow along on LV)**

In the same VI as above (variables and displays):

* Introduce stop button types, functions and latching mechanisms
* Other Booleans: controls and indicators

**Creating and adding SubVIs**

In the same VI as above (variables and displays):

* Add wiring terminals and wire output
* How to modify icon

Julian Day subVI example (see /examples and functions/Generic VIs and subVIs/):

**Structures and Loops**

In the same VI as above (variables and displays):

* While, For loops
  + Stopping While loops
* Flat sequences
* Case / conditional structures

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”.