LabVIEW Lesson Plans

These plans are more guidelines then full info. It is meant to be used as an outline by the instructor(s) who should be familiar with basics of LabVIEW, NXT in particular. While the lessons are broken into sections (about 60-90 minutes each), lessons can be stopped and started at any point.

Lesson 1A: PROGRAMMING INTRO

Think like a programmer, in steps

    -Carefully define starting configuration.

    -Tackle each action individually

    -Break each action down into simpler actions

    -Put each action in step-by-step order

    -Be **very** precise on each step

Activity: Make a list and compare to flow chart of steps to dial a phone number or bake a cake or pedal a bike.

Lesson 1B: LabVIEW INTRO

To prepare - Go to internet site: http://zone.ni.com/devzone/cda/tut/p/id/7466

1. **Do Exercises (click buttons on website, read tutorial if need more info)**:

    A.   LabVIEW environment: Launching LabVIEW module

    B.   LabVIEW environment: Front Panel module

    C.   LabVIEW environment: Block Diagram module

    D.   Passing Data…: Wires module

2. Show how to save a new project. Have group start one labeled "lvtutor"

3. Show how to save VIs, label each lvt# (number consecutively), save in project.

4. Go over numeric controls (placing and what they do):

    add, subtract, multiply, divide, increment, decrement, round up/down, random number

5. Explain how to get context help, use ctrl-H or Help

When completed, students should be able to:

1. start LabVIEW, open/save VI, open/save a project

2. navigate through front panel and block diagram

3. open and pin palettes

4. place objects into front panel and block diagram (Be familiar with basic numeric controls)

5. understand how to wire basic objects

6. Be familiar with following vocabulary:

    VI- Virtual Instrument. It is the name of the program you make.

    Front Panel- The user panel that will have all the input and output boxes.

    Controls Palette- The 'pop-up' box of items to use in the front panel

    Input- data put in to the program

    Output- data program creates and shows in designated fashion

    Block Diagram- The place where your entire program is stored.

    Functions Palette- The 'pop-up' box of items to use in the block diagram

    Numeric Control- An object used to manipulate numbers in the program (add, subtract, etc.)

LabVIEW Lesson Plans

Lesson 2: Beginning Programming

To prepare - Open project in LabVIEW, _Go to internet site: http://zone.ni.com/devzone/cda/tut/p/id/7466_

    1.   **Lecture or Watch Video then Do Exercises**:

        A.   Passing Data...: Data types...

        B.   Passing Data...: SubVI

2.   Review Constants, Controls, Indicators (make sure concept is understood well)

3.   go over the following numeric controls (what they do)

    comparison greater/less than, equal, etc. (possibly select and in range/coerce if time)

4.   go over the following boolean controls first 8 on menu

Activity: complete the following simple program

1.   make sub vi add, subtract, multiply or divide two numbers

2.   compare number from step 1) to a constant

3.   make sub vi using boolean controls, indicate whether less, equal or more than constant on front panel (use 3 leds or other indicators)

When completed, students should be able to:

1.   create a simple program using numeric and boolean controls

2.   create a front panel to display results

3.   Run a vi, use hilight debug

4.   Be familiar with the following vocabulary:

    Constants- _Value that will not change_

    Controls- _Inputs can change value either with the program or input from the user_

    Indicators- _output value, usually shown on front panel_

    boolean control- true or false value

    string control- generally holds text

    enum control-both numeric and string values

    dataflow-sequence of execution of program functions

    Debugging- Running the program to find where your errors are occurring. (Highlight Execution).

    modularity- the ability to create 'building blocks' of code (sub VIs) to use within LabVIEW. Allows the visual simplification where code is repeated.

    subVI-a section of code that can be used within another program. Indicated with its own icon.

LabVIEW Lesson Plans

Lesson 3: Intermediate Programming

To prepare - Open project in LabVIEW, _Go to internet site: http://zone.ni.com/devzone/cda/tut/p/id/7466_

    1.   **Lecture or Watch Video then Do Exercises**:

        A.   Loops: For and While loops module

        B.   Timing, Storing data: Timing, Shift registers, case structures module

        C.   Variables: Local, Global, race conditions module

Activity: complete the following simple program

1.   Use a while loop to increment up from entered variable till greater than 100.

2.   Use case structure to do boolean function to determine if number from #1 is odd or even, if odd add 1.

3.   Use for loop to divide number from #2 by 2 for 10 times, display value on front panel.

When completed, students should be able to:

1.   Use while and for loops

2.   Understand and use case structures

3.   Understand how variables are used.

4.   Be familiar with the following vocabulary:

    While loop- controls flow, repeats block of code inside loop until required condition occurs

    structure tunnels- allows data to be moved into and out of loops

    For loop- controls flow, repeats block of code inside loop for set number of times.

    iteration – quantity of repetitions of a loop

    wait function- used in loops to delay next iteration

    shift register- method used to transfer values from one iteration of a loop to the next

    case structure – a method for choosing between two sections of code depending on variable condition.

LabVIEW Lesson Plans

Lesson 4a: NXT COMMANDS

To prepare - have robot and connector cord. Open new project label it "sumo"

1.   Download NXT Tool Kit: *http://zone.ni.com/devzone/cda/tut/p/id/4435*
2.   Loading Code onto the NXT Brick
   A.   *Save file, Tools, NXT Module,*
   B.   *Locate NXT, Connect, Compile and Download*
   C.   *If firmware is not updated, Update firmware*
3.   Show use of Input (Sensors) *Touch, light, Sound, Ultra Sonic, Rotation, and NXT Buttons*
4.   Show use of Output (Motors) *Unlimited, Distance, Time, Stop, and all mentioned but synchronized*
5.   Show use of Sound (optional) *Sound File (special sound or song)-Play tone (put in the frequency)*

Activity: complete the following simple program

1.   Drive the NXT sumo bot forward, reverse and turn right/left.

2.   Use light sensor to prevent leaving the sumo field.

3.   Use ultrasonic sensor to locate opponent and trigger a 'charging' move.

LabVIEW Lesson Plans

Lesson 4b: Extras – helpful items to go over if a group finishes everything else

Compile: The act of the computer taking the user-friendly code and translating it to computer-friendly code.

Navigate: *Customize Keyboard Shortcuts (Tools, Options, Menu Shortcuts)*

Organization: *Tile Horizontally/Vertically (used for seeing the block diagram and front panel at the same time)*

Organization -*Organize Items Vertically/Horizontally (Used to organize front panel)*

Comments -*Navigating in the palette to Structures, then Decorations, will allow you to place any object that will not affect the programming, but will help other programmers or you in the* future

Errors: in/out, display error, errors in program (shows location)

Structures: *Sequence Structure,-Disable Structure, (Conditional and Standard)*

Tool Kits and how to obtain them

Toolkits are add-on command and/or component files for specific applications – such as NXT or FRC where some of the parts are known such as motors or sensors. Toolkits need to be added to the basic LabVIEW program. You can find them on the program disks or you may be able to download them for free online.