# INSTITUTE OF AERONAUTICAL ENGINEERING

(Autonomous)

Dundigal - 500 043, Hyderabad, Telangana

## COURSE CONTENT

| VIRTUAL INSTRUMENTATION LABORATORY | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| **V Semester: ECE** | | | | | | | | |
| **Course Code** | **Category** | **Hours / Week** | | | **Credits** | **Maximum Marks** | | |
| | | **L** | **T** | **P** | **C** | **CIA** | **SEE** | **Total** |
| **AECD29** | **Core** | - | - | 2 | 1 | 40 | 60 | 100 |
| **Contact Classes: Nil** | **Tutorial Classes: Nil** | **Practical Classes: 45** | | | | **Total Classes: 45** | | |
| **Prerequisite: Electronic Devices and Circuits.** | | | | | | | | |

## I. COURSE OVERVIEW:

The Laboratory Virtual Instrument Engineering Workbench (LabVIEW) is a development environment designed by National Instruments that creates graphic-based programs called virtual instruments (VIs) that simulate actual laboratory instruments. The experimental objective of this lab is to use LabVIEW to design basic operations and data acquisition using myDAQ and myRIO cards. Building these systems will demonstrate the potential for using simulated instruments in a laboratory. These programs will also obtain data from outside the computer and incorporate it into a program design.

## II. COURSE OBJECTIVES:

**The students will try to learn:**

I. The concept of virtual instrumentation used to develop basic VI programs using loops, case structures for image, signal processing and motion control applications.
II. LabVIEW tool to design basic operations and data acquisition using myDAQ and myRIO's
III. Prototype model for distribute stand-alone applications using LabVIEW.

## III.COURSE OUTCOMES:

**At the end of the course students will be able to:**

CO1 Demonstrate the LabVIEW graphical programming environment for virtual instrumentation applications
CO2 Make use of single and nested -loop design patterns for implementing iterative operations in LabVIEW.
CO3 Apply cluster and frame based techniques on data for collective and distributive data application.
CO4 Demonstrate the mathematical operations on waveforms using LabVIEW for signal processing and communication applications
CO5 Demonstrate the mathematical operations on waveforms using LabVIEW for signal processing and communication applications.
CO6 Test,control performance parameters of Electrical motors using My DAQ and My RIO.

IV. COURSE CONTENT:

# EXERCISES FOR VIRTUAL INSTRUMENTATION LABORATORY

**Note:** Students are encouraged to bring their own laptops for laboratory practice sessions.

## 1. Getting Started Exercises

### 1.0 Introduction:

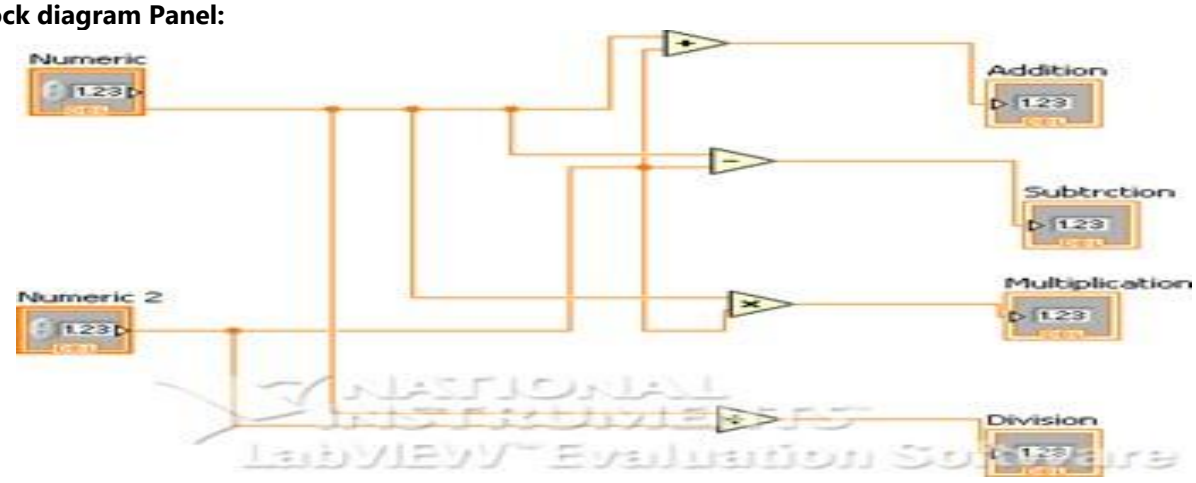
LabVIEW is a software development environment created by National Instruments. Originally it was focused on taking measurements from various lab instruments, but it's expanded drastically from its inception. Strictly speaking, LabVIEW is not a coding language, it's a development environment. The language is actually called "G", but most people refer to LabVIEW as if it's a language (i.e., most people would say "it's coded in LabVIEW").

LabVIEW is graphically-based, meaning you drag around various building blocks and connect them in a data flow architecture. It's similar to drawing a block diagram, except you're drawing your code, as opposed to text-based languages like C & VHDL where you type out in text what you want the software to do. We can implement mathematical and boolean operations graphically.

### 1.1 Arithmetic Operations

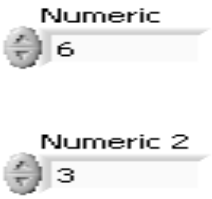
1. Start the Lab view and select the blank VI.
2. Create front and block diagram panel.
3. Numeric controls are given as inputs and numeric indicators are given as output they are selected by right clicking on the front panel.
4. Different arithmetic operators such as addition, subtraction, multiplication and division are generated in block diagram panel.
5. Using wiring operation inputs and outputs are connected to the respective operators in the block diagram panel.
6. Input values are given in the front panel and the program is executed. Hence the output is generated

**Block diagram Panel:**

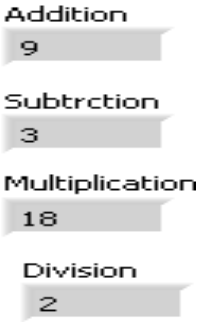


**Front Panel:**

**Input:**



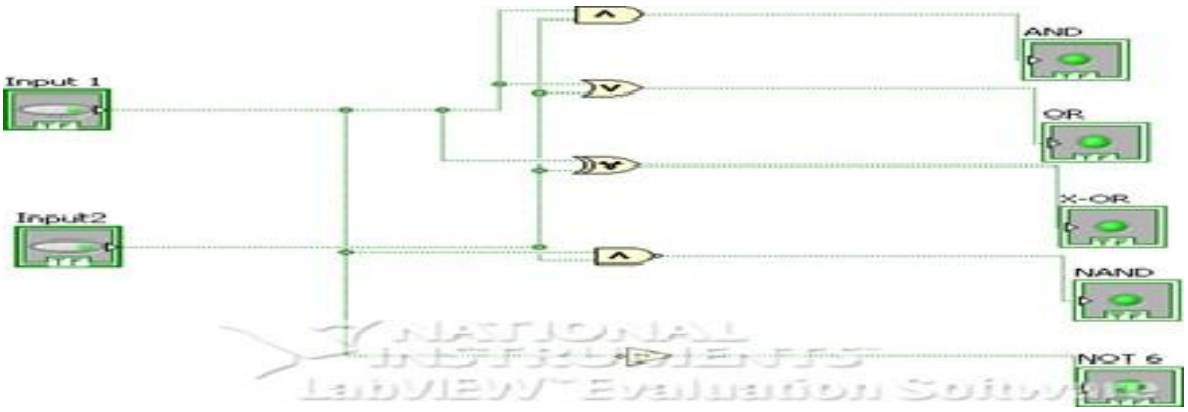

**Output:**




## 1.2 Logical Operations

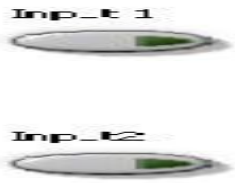
1. Start the Lab view and select the blank VI.
2. Create front and block diagram panel.
3. To perform Boolean operation push buttons are taken as inputs and round LED as output.
4. Different Boolean operations such as AND, OR, XOR, NOT, NAND are selected from the block diagram panel.
5. Boolean inputs and outputs are wired in the block diagram panel.
6. Logic values 0 & 1 are given in the front panel and the program is executed.

**Block Diagram Panel:**




**Front Panel:**

**Input:**




**Output:**

### Try:

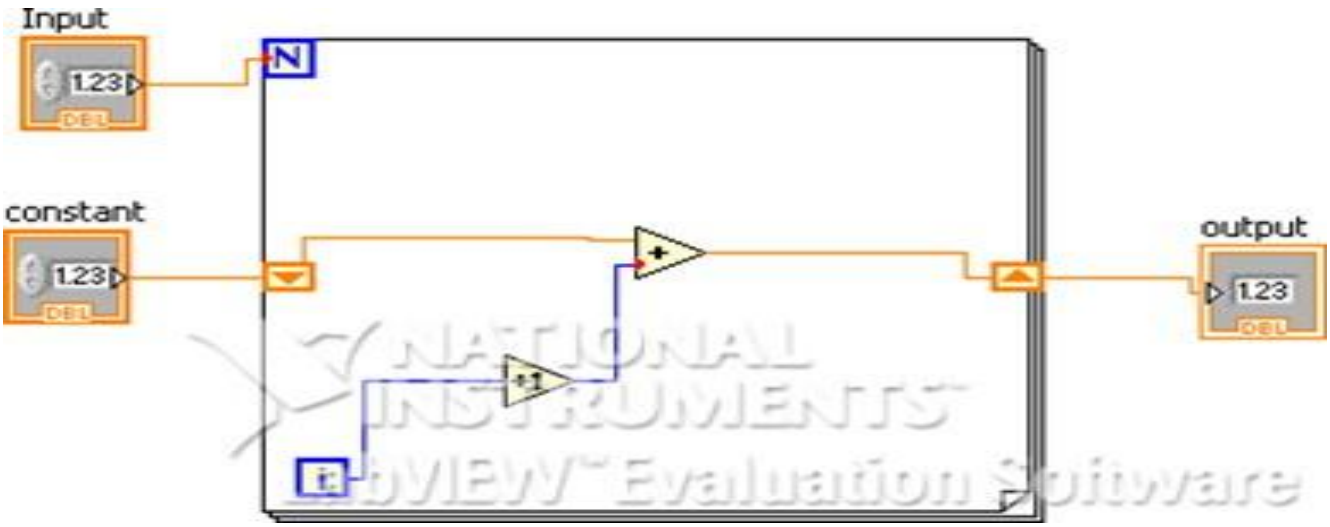
1. Build a VI (Virtual Instrumentation) to find the divide two numbers and glow an LED.
2. Build a VI to calculate the area and perimeter of the rectangle, and the area and circumference of the circle.

## 2. Exercises on sum and factorial of a given numbers using for & while loop

### 2.1 Sum of 'n' number using 'FOR' Loop

1. Create blank VI.
2. Right click on the block diagram panel, select program, go to structures and select a FOR loop.
3. Right click on the border of the FOR loop and select add shift register; borders are converted into shift register.
4. Using wiring operations required connections are given in the block diagram.
5. Inputs are given in the front panel and the program is executed.

**Block Diagram Panel:**




**Front Panel:**

**Input:**

Input
10

constant
0

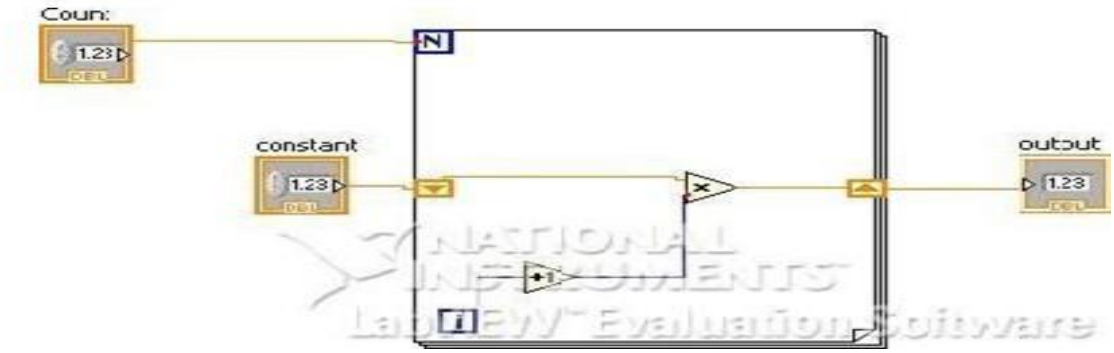
**Output:**

output
55

## 2.2 Sum of 'n' number using 'WHILE' Loop

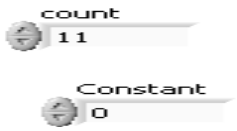
1. Create blank VI.
2. Right click on the block diagram panel, select program, go to structures and select a WHILE loop.
3. The tunnels of the border are converted to shift register.
4. Generate an adder and greater than equal to inside the WHILE loop.
5. Using wiring operation required wiring is done in the block diagram panel.
6. Input values are given in front panel and the output is generated.

**BLOCK DIAGRAM PANEL:**




**FRONT PANEL:**

**Input:**





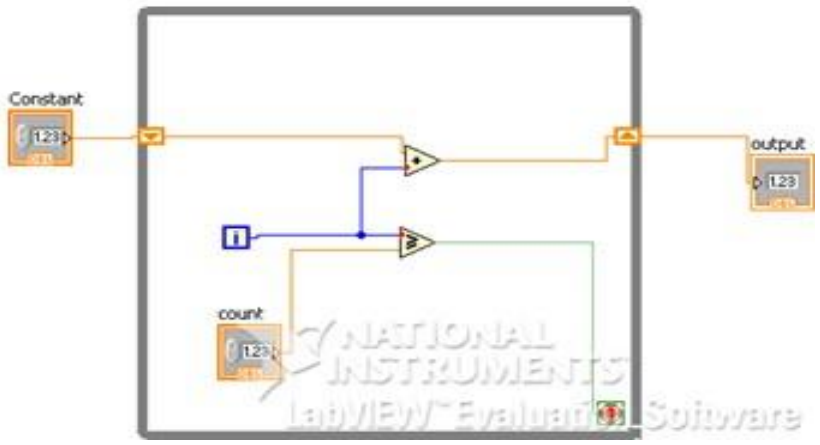


**Output:**

## 2.3 Factorial of 'n' number using 'FOR' Loop

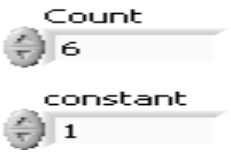
1. Create blank VI.
2. Right click on the block diagram panel , select program , go to structures and select a FOR loop.
3. Right click on the border of the FOR loop and select add shift register; borders are converted into shift register.
4. Using wiring operations required connections are given in the block diagram.
5. Inputs are given in the front panel and the program is executed.

**BLOCK DIAGRAM PANEL:**

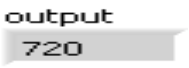



**Front Panel:**

**Input:**



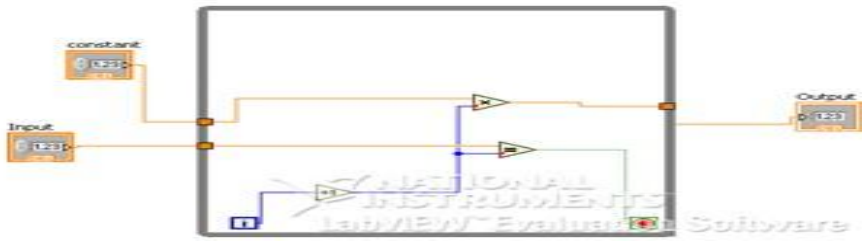

**Output:**




## 2.4 Factorial of 'n' number using 'WHILE' Loop

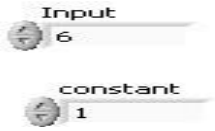
1. Create blank VI.
2. Right click on the block diagram panel, select program, go to structures and select a WHILE loop.
3. Right click on the border of the WHILE loop and select add shift register; borders are converted into shift register.
4. Using wiring operations required connections are given in the block diagram.
5. Inputs are given in the front panel and the program is executed.

**BLOCK DIAGRAM PANEL:**




**Front Panel:**

**Input:**




**Output:**

### Try: Reverse of a given string

1. Write a program in LabVIEW to generate a Fibonacci series of n numbers where n is defined by the programmer. Example for the Fibonacci series is: 1 1 2 3 5 8 13 21 34 and so on.
2. Build VI to find Sum of even numbers using FOR loop and WHILE loop.

# 3. Exercises on Modulation and Demodulation

## 3.1 Introduction

This lab introduces students to communications theory with amplitude modulation and demodulation. Students will explore the mathematical theory behind amplitude modulation and use the Analog Discovery 2 to visualize the effects of amplitude modulation in the time and frequency domains. Then, students will use LabVIEW to program an AM demodulator and use it to explore and visualize the effects of the modulation coefficient on the quality of the demodulated signal and the effects of different parameters (such as windowing and averaging) on the Fast Fourier Transform (FFT). Advanced students can challenge themselves to build a system to send data between two Analog Discovery 2s or to build an analog AM demodulator.or to coming to lab class, complete Part 0 of the Procedure.

## 3.2 Procedure

(a) Open LabVIEW development environment and select file and select a new VI or directly press 'Ctrl+N' to open a new virtual instrument. Virtual instrument window is a combination of Front panel (User Interface) and Block Diagram (functional implementation window). Now in order to arrange the windows alternatively we have to press 'Ctrl+T'.

On the front panel i.e. user interface window we have to create multiple graphs for viewing the Message, Carrier, Modulated, Demodulated and Spectrum waveforms and we have to create a carrier amplitude control in order to exhibit perfectly modulated, under modulated and over modulated patterns as well. Since the application has to run

continuously we have to create the while loop in block diagram. So in order to stop the loop we have to create the stop button from Boolean sub palette in front panel. When we right click on the front panel we will get the controls palette from which we can select the different data types.

(b) Now on the block diagram we have to create the while loop to run the application continuously until we desire to stop. we can drop a while loop from the structures sub palette by right clicking on the block diagram we will get the functions palette from which we can select the desired functionalities. We have to drop the simulate signal express palette from Express menu and the input section, In order to create the message and carrier signals. Note that when you create the message and carrier signal the samples to read must be same for both simulate signal blocks and as soon as we have dropped the simulate signal blocks we will get a pop-up window to configure the simulate signal or else you double click on the block to reconfigure.

(c) We are now mathematically constructing the AM modulated wave from the equation mentioned below. Based on which we have to drop the functionalities from the numeric sub palette , We have to drop addition, Multiplication and reciprocal functions to implement the below mentioned equation. After completion of the equation implementation we will get a modulated wave. we can create a demodulated wave by multiplying the modulated wave with the carrier signal and then passing it through the low pass filter which is tuned to message signal frequency will produce the input message signal ( Demodulated signal). Now we have to assign each and every graph to Message, Carrier, Modulated and demodulated outputs will be connected to the graphs. In the equation since we require the Cosine version of Carrier signal we have to apply the phase of 90 deg in carrier simulate signal block as mentioned below.

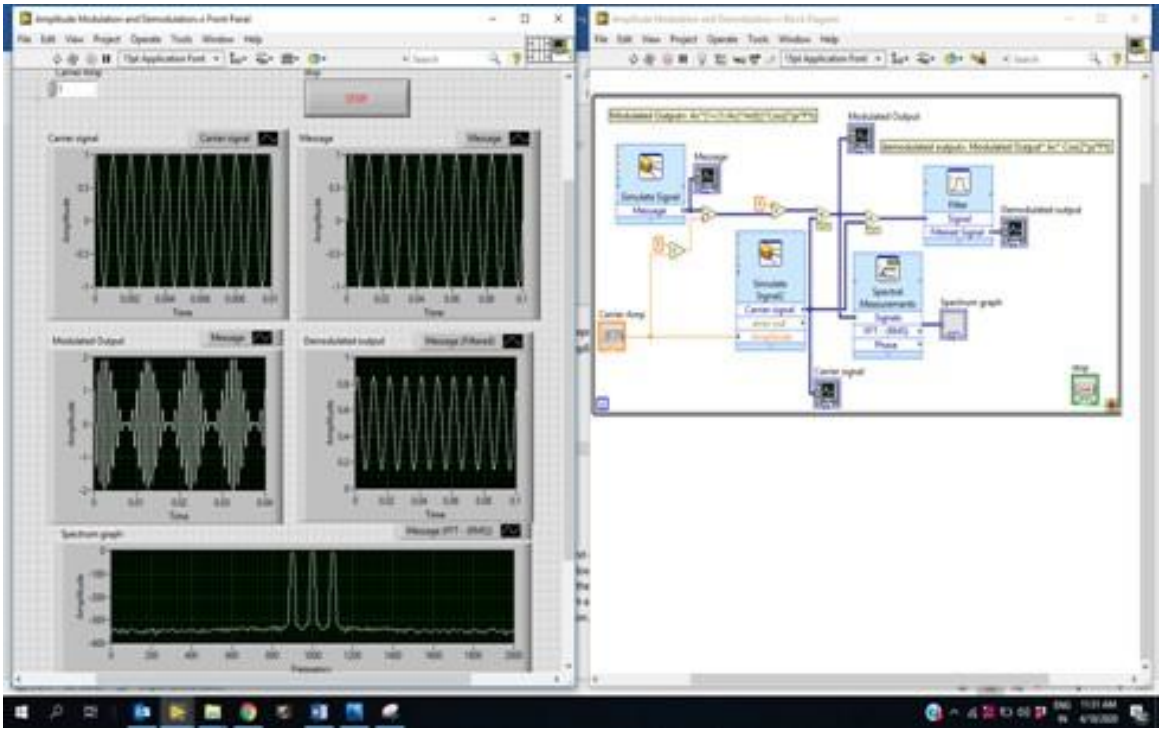
(d) Now since as mentioned in the above step in order to demodulate the signal we have to multiply the modulated wave equation with the carrier signal and then pass it through the low pass filter and the filter is available under the express sub palette and signal Analysis sub palette. Similarly inorder to view the spectrum of the signal we have to apply spectrum analysis by using FFT operation to view the spectrum of the modulated wave equation in the graph.

(e) As soon as we drop the filter and spectral measurements blocks. since these are express VI's pop window will opened as soon as we will drop these into the block diagram. We can then configure the properties of functionalities.

(f) As soon as we drop the filter and spectral measurements blocks. since these are express VI's pop window will opened as soon as we will drop these into the block diagram. We can then configure the properties of functionalities.

(g) After Finally interconnecting all the blocks and connecting the respective controls and indicators for the respective blocks as shown in the below figure the complete application will be ready to verify. We can either press "CTRL+R" to run the application or click on the run arrow on the task bar from either front panel and block diagram and verify the output in all the instants and to stop the application we can click on the stop button.

**Front Panel and Block Diagram Panel:**



## Try:

1. Build a VI to perform Double side band suppressed carrier modulation on sinusoidal signal and estimate output power.
2. Build a VI to perform Amplitude modulation and estimate output power.

# 4. Exercises on Signal Generators

## 4.1 Introduction

This lab introduces students to the fundamentals of signal generation using LabVIEW. The objective is to explore how different waveforms can be generated and visualized in the time and frequency domains. Students will use Simulate Signal Express VIs to generate standard signals such as sine, square, triangle, sawtooth, and white noise. They will also be able to control parameters such as amplitude, frequency, phase, and duty cycle through user-defined controls on the front panel. The exercise also covers real-time signal switching and monitoring through waveform graphs and spectrum analysis using FFT.

By the end of the lab, students will be capable of building a reusable virtual instrument (VI) that simulates different signal types dynamically based on user selection.

## 4.2 Procedure

(a) Open the LabVIEW development environment and select a new VI (Ctrl+N). Tile the Front Panel and Block Diagram using Ctrl+T.
On the Front Panel, create the following components:
Enum control or radio buttons to select waveform type (Sine, Square, Triangle, Sawtooth, Noise).
Numeric controls for: Amplitude,Frequency,Phase,Duty Cycle (for square wave)

A waveform graph to display the time-domain signal.
A stop button from the Boolean palette.

(b) In the Block Diagram:

Place a While Loop from Functions → Structures to ensure continuous operation.
Use the Case Structure inside the While Loop, wired to the waveform selection control.
For each case (Sine, Square, Triangle, Sawtooth, Noise):

Use the Simulate Signal Express VI (from Functions → Express → Input) to generate the corresponding signal.

Configure parameters:Waveform type,Samples per second (e.g., 1000),Number of samples (e.g., 100), Frequency, amplitude, phase

For noise, choose White Noise as the signal type.

(c) Connect user controls to each signal block:

Wire amplitude, frequency, phase inputs to the Simulate Signal configuration terminals.
For square wave, add and connect duty cycle.

(d) To enhance visualization:

Add FFT Spectral Measurement Express VI (from Express → Signal Analysis) to the output signal.
Display its output on a second graph labeled Spectrum.

(e) Finalize the VI:

Connect signal outputs to the respective waveform graph indicators on the Front Panel.
Wire the stop button to the While Loop conditional terminal.
Run the application using Ctrl+R or the Run arrow.
Front Panel and Block Diagram Panel:
Should include: waveform selector, amplitude/frequency/phase/duty controls, signal output graph, spectrum graph, and a stop button.

### Try:

1. Build a VI to perform Modify the VI to allow real-time switching between signals while the VI is running.
2. Build a VI to perform Add peak detection or zero crossing detection on the signal for real-time signal analysis.

# 5. Exercises on filters to filter unwanted frequencies

## 5.1 Introduction

This lab introduces students to digital filtering techniques used in signal processing to remove unwanted frequencies from a signal. Students will explore different types of filters including Low Pass, High Pass, Band Pass, and Band Stop filters using LabVIEW Express VIs. Using these filters, students will analyze the effects on noisy and multi-frequency signals. The lab involves the generation of composite signals using sine waves and noise, and then selectively filtering out undesired components. Real-time visualization using time-domain waveform graphs and frequency-domain FFT plots will help students understand the operation of each filter type.Before coming to the lab, students should complete Part 0 of the procedure and review the theoretical frequency response of basic filters.

## 5.2 Procedure

(a) Open LabVIEW and create a new VI (Ctrl+N). Use Ctrl+T to tile the Front Panel and Block Diagram for easy navigation.

On the Front Panel:
Add numeric controls for Amplitude, Frequency1, Frequency2, and Noise Level.
Add an Enum control or Ring control to select Filter Type (Low Pass, High Pass, Band Pass, Band Stop).
Add numeric controls for Cutoff Frequencies (F_low and F_high).

Insert waveform graphs for Original Signal, Noisy Signal, and Filtered Output.
Add a Stop Button from the Boolean palette.

(b) On the Block Diagram

Place a While Loop to allow continuous signal generation and filtering.
Use Simulate Signal Express VI to create two sinusoidal signals (e.g., 100 Hz and 500 Hz)
Use a third Simulate Signal VI to create White Noise, adjustable via the Noise Level control
Combine the sine waves and noise using Addition functions to generate the Noisy Composite Signal.
Display this signal on the Noisy Signal Graph.

(c) Drop a Filter Express VI from Functions → Express → Signal Analysis → Filter.
Wire the Noisy Composite Signal into the input of the Filter VI.
Use a Case Structure or Selector wired to the filter type control to change filter settings:

Low Pass: Set F_low as cutoff
High Pass: Set F_high as cutoff
Band Pass: Use both F_low and F_high
Band Stop: Use both F_low and F_high

Each time a filter type is selected, the filter VI must be configured accordingly (passband or stopband type, filter order, etc.).

(d) Connect the Filtered Output to its waveform graph on the front panel
Additionally, place an FFT Spectral Measurement VI on both the Noisy Signal and Filtered Output paths.
Wire these to two additional waveform graphs labeled Noisy Spectrum and Filtered Spectrum to visualize the frequency response of the filtering process.

(e) Use the Stop Button to control the While Loop exit condition.

(f) Interconnect all components and verify wiring and data types using context help (Ctrl+H). Run the VI using Ctrl+R or the Run arrow and test all filter types dynamically by switching the control.

Front Panel and Block Diagram Panel:

The front panel should contain:
Signal parameter controls (Amplitude, Freq1, Freq2, Noise)
Filter type selector
Cutoff frequency controls
Four graphs (Noisy Signal, Filtered Signal, Noisy Spectrum, Filtered Spectrum)
Stop button

### Try:

1. Build a VI to compare the output of Butterworth and Chebyshev filters on the same noisy signal.
2. Build a graphical user interface (GUI) to allow dynamic tuning of filter parameters during runtime

# 6. Exercises on Measurement and Data Acquisition using MYDAQ

## 6.1 Introduction

This lab introduces students to data acquisition principles using the NI myDAQ device. The myDAQ provides a compact and affordable way to interface real-world analog signals with the LabVIEW environment. Students will learn how to acquire analog input signals (e.g., from a function generator or sensor), visualize the signals in the time and frequency domains, and perform basic signal processing such as scaling, averaging, and filtering. Advanced learners can explore real-time monitoring, signal

logging, and dual-channel acquisition techniques.Before coming to the lab, complete Part 0 of the procedure and install the NI ELVISmx driver package.

## 6.2 Procedure

(a) Open LabVIEW and create a new Virtual Instrument (VI) using Ctrl+N. Use Ctrl+T to tile the Front Panel and Block Diagram windows.

On the Front Panel:
Create controls for Sampling Rate, Number of Samples, and Channel Selection (e.g., ai0, ai1).
Add waveform graphs labeled Acquired Signal (Time Domain) and Acquired Signal (Frequency Domain).
Add indicators for Mean, RMS, and Peak Amplitude.
Place a Stop Button from the Boolean controls palette.

(b) On the Block Diagram:
Insert a While Loop from the Structures palette to continuously read and update data until the user stops the VI.
Add the DAQ Assistant Express VI from Express → Input → DAQ Assistant.
When prompted:
Select Analog Input → Voltage
Choose channel(s) (e.g., Dev1/ai0)
Set acquisition mode to Continuous Samples
Configure Rate and Samples per Channel using Front Panel controls
Finish setup to place the DAQ Assistant block

(c) Connect the DAQ output to:
A Waveform Graph for real-time time-domain visualization.
An Express Spectral Measurement VI to perform FFT and wire its output to a second graph for frequency-domain display.

(d) For analysis:
Use Waveform Analysis VIs from Signal Processing → Waveform Measurements to compute:
Mean (DC Level)
RMS
Peak
Wire each analysis result to numeric indicators on the front panel.

(e) Add scaling or filtering options:
Insert a Filter Express VI from Signal Analysis → Filter if needed to remove noise.
Enable control over Filter Type and Cutoff Frequency from the front panel.

(f) Wire the Stop Button to the conditional terminal of the While Loop to allow manual termination of the acquisition.

(g) After interconnecting all elements and verifying signal flow, run the VI using Ctrl+R or the Run arrow on the toolbar. Adjust input frequency, amplitude, or signal type using external sources and verify the response in both time and frequency domains.

Front Panel and Block Diagram Panel:
The Front Panel should include:
Input controls: Sample Rate, Sample Size, Channel
Graphs: Time Domain Signal, FFT Signal

Numeric indicators: Mean, RMS, Peak
Stop Button
The Block Diagram should clearly show:
DAQ Assistant
Signal analysis and display logic
Optional filter or signal scaling blocks
While Loop for real-time operation

### Try:

1. Build a VI to display and compare filtered vs. unfiltered signals using a Band Pass Filter.
2. Build a VI to acquire signals from two analog channels simultaneously (ai0 and ai1) and compare their spectra.

# 7. Exercises on Generating signals using MYDAQ

## 7.1 Introduction

This lab introduces students to the signal generation capabilities of the NI myDAQ device using LabVIEW. The NI myDAQ can output analog waveforms through its AO channels, allowing students to create custom waveforms like sine, square, triangle, and arbitrary signals. This lab walks students through building a VI that allows them to generate signals of desired frequency, amplitude, and waveform type in real-time, observe the generated waveform on an oscilloscope, and optionally modulate it.Before coming to the lab, complete Part 0 of the Procedure and ensure NI ELVISmx drivers are installed and NI myDAQ is properly connected and detected in NI MAX.

## 7.2 Procedure

(a) Open the LabVIEW development environment and create a new VI using Ctrl+N. Use Ctrl+T to tile the Front Panel and Block Diagram side by side.
On the Front Panel, create:
A control for Waveform Type (Sine, Square, Triangle)
Controls for Frequency, Amplitude, Phase (Degrees), and Offset
A Boolean Stop Button to terminate the loop
Optional: A Numeric Indicator for tracking elapsed time or number of cycles

(b) On the Block Diagram:
Insert a While Loop from the Structures palette to continuously output the signal until the Stop button is pressed.
From Express → Output, place the DAQ Assistant VI into the block diagram.
Configuration Steps (upon prompt):
Select Analog Output → Voltage Output
Choose channel (e.g., Dev1/ao0)
Set generation mode to Continuous
Click Finish
(c) Inside the loop:
Add a Simulate Signal Express VI from Express → Signal Generation.
In the configuration popup:
Choose the waveform type (set default to Sine)
Set Frequency, Amplitude, and Samples per second to reasonable defaults (e.g., 1000 Hz, 1 V, 1000 S/s)

Ensure Number of Samples is consistent with DAQ configuration

To allow waveform type selection, use an Enum Control on the Front Panel and connect it to a Case Structure. In each case (Sine, Square, Triangle), configure a separate Simulate Signal VI or adjust the Signal Type input programatically.

(d) Connect the output of the Simulate Signal VI to the DAQ Assistant input to drive the AO channel.
Optional: Use a Wait (ms) block inside the loop to manage update rate (e.g., 10–50 ms)
Optional: Display waveform in a graph to verify before outputting

(e) Configure your signal dynamically:
Convert Front Panel controls (Amplitude, Frequency, Offset, Phase) to inputs to the Simulate Signal VI via the "Input/Output terminals" option in its configuration or right-click > Create > Control
Link them from the Front Panel

(f) Once the Simulate Signal and DAQ Assistant are linked and loop is complete, wire the Stop button to the Loop conditional terminal.
Optional: Include an Elapsed Time Express VI or a Shift Register to show time duration or waveform cycles

(g) Save and run the VI (Ctrl+R) or use the Run arrow. Observe output on a real oscilloscope connected to AO 0 (and ground). Vary signal parameters in real time using the Front Panel controls.
To stop the VI, press the Stop button.
Front Panel and Block Diagram Panel
Your Front Panel should include:
Controls for waveform type, amplitude, frequency, phase, offset
A waveform graph (optional)
A stop button
Your Block Diagram should include:
A While Loop enclosing the Simulate Signal VI, DAQ Assistant, and control wiring
Optional graph, wait timer, and time indicators

### Try:

1. Build a VI to generate a sweep signal (chirp) from 100 Hz to 1 kHz and observe it on an oscilloscope.
2. Design a VI to switch between waveform types (sine, square, triangle) using an enum and case structure in real-time.

# 8. Exercises on Measurement and Data Acquisition using MYRIO

## 8.1 Introduction

This lab introduces students to real-time data acquisition and signal measurement using the NI myRIO embedded hardware platform. Students will learn to acquire analog input signals (such as temperature, voltage, audio, etc.) using myRIO AI channels, visualize them in real-time using LabVIEW, and analyze them in both time and frequency domains. This helps develop core skills in interfacing with physical systems, signal conditioning, and digital acquisition.

Before coming to the lab class, ensure:

The NI myRIO Toolkit is installed

NI myRIO is connected via USB

NI MAX recognizes the device

## 8.2 Procedure

(a) Open the LabVIEW development environment and create a new VI using Ctrl+N. Press Ctrl+T to tile the Front Panel and Block Diagram side by side.
On the Front Panel, create:
A Waveform Chart to visualize the input signal in time domain
A Stop Button from the Boolean sub-palette
Optional: Frequency spectrum graph, numeric indicators for min/max/mean values

(b) On the Block Diagram:
Drop a While Loop from the Structures sub-palette to continuously read the input until stopped.
From the myRIO palette (Functions → myRIO → Devices), select and place the Analog Input Express VI.

Configuration popup:

Select myRIO-1900 (or your device)
Choose Analog Input Channel (e.g., myRIO/AI0)
Set Acquisition Mode to Continuous
Set sampling rate (e.g., 1kHz) and number of samples (e.g., 100)

(c) Inside the While Loop:
Connect the output of the Analog Input VI to the Waveform Chart to visualize the acquired signal in real-time
Add a Stop Button wired to the loop termination
Optional: Use Waveform Measurements Express VI to calculate mean, RMS, or frequency content

(d) For frequency analysis:
Drop the Spectral Measurements Express VI from Express → Signal Analysis → Spectral
Wire the output of the Analog Input VI into the Spectral block
Configure the FFT parameters: window type (e.g., Hamming), FFT length, averaging
Connect its output to a Graph on the Front Panel to visualize the spectrum

(e) To enhance usability:
Create Front Panel Controls for sample rate, averaging toggle, and FFT window type
Use Case Structures to allow real-time switching between time and frequency views

(f) After completing the wiring and configuration:
Verify connections between blocks
Ensure appropriate data types are used (e.g., waveform, dynamic data)
Save your VI

(g) Press Ctrl+R or click the Run arrow to start the application. Observe real-time data acquisition from the myRIO AI channel and corresponding waveform display. Stop the VI by pressing the Stop button. You can connect a function generator, potentiometer, or audio source to AI0 and GND to test the data acquisition setup.
Front Panel and Block Diagram Panel
Your Front Panel should include
Real-time waveform chart

Optional: spectrum graph, controls for FFT or display settings
Stop button
Your Block Diagram should include:
While Loop with Analog Input Express VI
Optional: Spectral Measurement Express VI
Indicators and controls wired appropriately

### Try:

1. Design a VI to capture audio from a microphone connected to myRIO and show its spectrogram.
2. Create a VI that calculates and displays the peak voltage and RMS of the acquired signal in real-time.

# 9. Exercises on Generating signals using MYRIO

## 9.1 Introduction

This lab exercise introduces students to the process of generating real-world analog signals using the NI myRIO. Students will design and implement virtual instruments (VIs) to generate sinusoidal, square, and modulated signals (like AM) and output them through AO channels of myRIO. These signals can be visualized on an oscilloscope or captured using an analog input (AI) channel for real-time feedback and spectrum analysis. Students also explore parameter control such as frequency, amplitude, phase, and duty cycle (for square waves).

Before the lab, ensure:
NI myRIO Toolkit is installed
myRIO is connected via USB and recognized in NI MAX
An oscilloscope or another DAQ input is available to visualize output signals

## 9.2 Procedure

(a) Open the LabVIEW development environment. Press Ctrl+N to start a new VI, and Ctrl+T to tile the Front Panel and Block Diagram.
On the Front Panel, create:
Amplitude, Frequency, and Phase controls for the generated signal
A Waveform Graph to preview the signal
A Stop button to end the application
Optionally: Signal type selector (Sine, Square, Triangle)

(b) In the Block Diagram:
Create a While Loop from the Structures sub-palette
Place the Simulate Signal Express VI (from Express → Input)
- Configure it to generate a sine wave (or allow user-selectable waveform type)
- Set samples per second (e.g., 1kS/s) and samples per loop (e.g., 100)

Wire Front Panel controls to the Simulate Signal block to allow real-time adjustment of:
- Amplitude
- Frequency
- Phase

(c) From the myRIO → Devices palette, drop the Analog Output Express VI
Configure it to use AO0 (e.g., myRIO/AO0)
Set the output mode to Continuous Output
Wire the output of the Simulate Signal to the Analog Output input
Optional:

Split signal path to feed into a Waveform Graph for preview on the front panel
Use Build Array if outputting two signals (e.g., message + carrier)

(d) For Amplitude Modulation, modify the signal path:
Drop two Simulate Signal blocks: one for message signal, one for carrier
Implement the AM modulation using the equation:
s(t)=[1+m(t)]·c(t)s(t) = [1 + m(t)] \cdot c(t)s(t)=[1+m(t)]·c(t)
Use Multiply and Add functions from the Numeric palette
Allow control over modulation depth (modulation index) via a Front Panel slider

(e) To visualize spectrum:
Drop the Spectral Measurements Express VI from Express → Signal Analysis
Wire the modulated output signal into this VI
Configure FFT properties (e.g., Hamming window, FFT length)
Display output in a Spectrum Graph
(f) Add a Stop button (from Boolean palette) wired to the While Loop condition terminal.

(g) After connecting all components and confirming functionality:
Press Ctrl+R to run the VI or click the Run arrow
View the waveform on the connected oscilloscope from AO0
Adjust frequency, amplitude, and observe the changes in real-time
Front Panel and Block Diagram Panel
Your Front Panel should include:
Amplitude, Frequency, Phase controls
Modulation index (for AM)
Signal type selector (optional)
Real-time waveform preview
Spectrum graph
Stop button
Your Block Diagram will include:
Simulate Signal VIs
Math functions (Multiply, Add)
Analog Output VI
Optional: Spectral Measurements VI
While Loop and control logic

### Try:

1. Build a VI to generate a 1 kHz square wave with variable duty cycle and output it via AO0.
2. Create a VI that toggles between sine, square, and triangle waveforms with real-time selection and signal generation.

# 10. Exercises on MYRIO Experiments

## 10.1 Introduction

This lab introduces students to embedded system applications using NI myRIO. Students will gain hands-on experience with real-time signal generation, data acquisition, and signal processing. These exercises use LabVIEW's graphical programming interface to interact with myRIO hardware, enabling students to create and test analog and digital experiments such as waveform generation, acquisition, modulation, filtering, and control systems.

By the end of this lab, students will:

Generate analog waveforms using myRIO's AO channels
Acquire analog signals using AI channels
Apply mathematical operations for modulation and filtering
Visualize signals in time and frequency domains
Control signal parameters interactively in real time

## 10.2 Procedure

(a) Getting Started:
Launch LabVIEW and open a new VI using Ctrl+N
Press Ctrl+T to tile the Front Panel and Block Diagram
Ensure that NI myRIO Toolkit is installed, and myRIO is connected and recognized in NI MAX
On the Front Panel:
Add numeric controls for Amplitude, Frequency, and Phase
Add a waveform graph to display the output
Add a Stop button from the Boolean palette

(b) Signal Generation with myRIO:
In the Block Diagram:
- Insert a While Loop from the Structures palette
- Drop a Simulate Signal Express VI (from Express → Input) and configure it for a Sine Wave
- Adjust parameters: frequency = 1 kHz, amplitude = 2V, samples = 100
- Add controls for amplitude, frequency, and phase

Drop the Analog Output Express VI from the myRIO → Devices palette
- Set it to AO0
- Wire the simulated signal to AO0

Optional: Add a Waveform Graph to view the signal in LabVIEW

(c) Signal Acquisition with myRIO:
Drop the Analog Input Express VI from myRIO → Devices
- Configure it for AI0 and set sample rate (e.g., 1kHz)

Connect AI0 to AO0 using a jumper wire on the myRIO for testing
Display the acquired waveform on a separate waveform graph

(d) Modulation (AM or DSB-SC):
Use two Simulate Signal blocks for:
- Message signal (e.g., 100 Hz sine)
- Carrier signal (e.g., 1 kHz cosine)

Construct AM signal using the equation:
$s(t)=[1+m(t)]\cdot c(t)s(t) = [1 + m(t)] \cdot c(t)s(t)=[1+m(t)]\cdot c(t)$
Use Add and Multiply blocks from the Numeric palette
Output the AM signal via AO0
Acquire it back using AI0

(e) Filtering and Demodulation:
Multiply the acquired AM signal with the carrier again
Use the Filter Express VI (Express → Signal Analysis → Filter)
Set to Low Pass, with a cutoff just above message frequency (e.g., 200 Hz)
View demodulated output in a Waveform Graph

(f) Spectrum Analysis:
Drop Spectral Measurements Express VI

Input the modulated or demodulated signal
Display frequency-domain waveform on a Spectrum Graph
Configure with proper windowing (e.g., Hamming), averaging if desired

(g) Final Integration and Testing:
Ensure that all signal paths, graphs, and controls are correctly connected
Wire the Stop button to the While Loop condition terminal
Click the Run arrow or press Ctrl+R to start the VI
Observe waveforms on the scope and in LabVIEW
Stop the VI using the Stop button
Front Panel & Block Diagram View:
Your Front Panel should include:
Amplitude, Frequency, Phase controls
Message and Carrier wave controls
Waveform Graphs for: Message, Carrier, Modulated, Demodulated, Spectrum
Stop Button
Your Block Diagram will include:
Simulate Signal VIs
Analog Output/Analog Input Express VIs
Math blocks (Add, Multiply)
Filter and Spectrum Analysis VIs
Control logic with While Loop

### Try:

1. Design a VI to generate and acquire square and triangular waveforms, adjusting duty cycle and slope in real time.
2. Implement real-time FFT on a signal captured via myRIO and display frequency peaks.

# 11. Exercises on FPGA Programming

## 11.1 Introduction

This lab introduces students to FPGA-based system design using the LabVIEW FPGA Module. Students will gain hands-on experience in developing high-speed, deterministic applications by leveraging the parallelism and timing precision of FPGAs. This experiment focuses on implementing real-time signal generation, processing, and visualization using an FPGA target like NI myRIO, NI sbRIO, or PXI FPGA cards.

By the end of this lab, students will:
Understand FPGA architecture and timing
Implement waveform generation, filtering, and real-time processing in FPGA
Compare software vs. FPGA execution in terms of speed and determinism
Deploy and test VI logic on FPGA targets

## 11.2 Procedure

(a) Open LabVIEW FPGA Project
Launch LabVIEW → Go to File → New Project
Select "NI myRIO with FPGA" or another supported FPGA target
Right-click the FPGA target in the project explorer → New → VI
This creates an FPGA-targeted VI that runs deterministically on hardware
Press Ctrl+T to tile the Front Panel and Block Diagram

On the Front Panel:
Add controls for Frequency, Amplitude, and Waveform type (Enum)
Add waveform graphs to display the input, filtered, and processed signals
Add a Boolean stop button for loop control

(b) Setup While Loop and Timing
In the Block Diagram:
- Drop a While Loop from Structures
- Add a Timed Loop (optional for precise timing)
- Drop a Loop Timer or Wait Function from FPGA Timing palette (e.g., 10 ms)

(c) Waveform Generation on FPGA
From Functions → FPGA Math & Signal Processing → Signal Generation, drop:
- Sine Wave Generator VI or implement your own using lookup tables

Use numeric operations to scale frequency and amplitude from controls
Use waveform indicator to display generated signal

(d) Real-time Signal Processing:
Perform operations on the waveform:
- Use Add, Multiply, Filter, and FFT from the FPGA Signal Processing palette
- For example: implement a moving average filter using a shift register
- Simulate modulation: Multiply signal by a high-frequency cosine signal

(e) I/O Interfacing (Optional):
Connect waveform to Analog Output (AO) pin on myRIO (e.g., AO0)
Read signal from Analog Input (AI) using FPGA I/O Nodes
Drop FPGA I/O Node, select AI0 and configure as input
Loop back AO0 to AI0 using a jumper to verify end-to-end FPGA data f

(f) Compile and Run:
After completing your design, go to Project Explorer
- Right-click the FPGA VI → Compile
- Select Cloud Compile or Local Compile Server

Once the FPGA bitfile is generated, create a Host VI
- Host VI runs on the PC or RT processor, communicates with FPGA
- Use Read/Write Control, Open FPGA VI Reference, and Close Reference

(g) Final Testing:
Run the Host VI to control and observe FPGA behavior in real-time
Monitor waveform graphs for real-time response and verify performance
Compare execution speed, response time, and CPU usage with software-only VI
Use “Ctrl+R” to run or Stop Button to halt the application

## Try:

1. Build an FPGA VI to perform real-time filtering (low-pass and moving average) on an analog input signal and visualize the result.
2. Implement Implement a VI to simulate DSB-SC modulation on FPGA, using look-up tables for carrier generation,and compute output power.

# 12. Exercises on Pulse Amplitude Modulation and Power Spectral Density calculation using USRP

## 12.1 Introduction

This lab introduces students to Pulse Amplitude Modulation (PAM) using NI USRP hardware and LabVIEW Communications System Design Suite. Students will learn to generate and transmit PAM signals through USRP, receive and demodulate them, and analyze their frequency characteristics using Power Spectral Density (PSD) estimation via FFT. This experiment bridges the gap between theory and practical SDR-based wireless communication.

Advanced students can extend the experiment to analyze PAM over noisy channels or experiment with different pulse shaping filters.

## 12.2 Procedure

(a) Setup and Launch LabVIEW

Open LabVIEW Communications System Design Suite

Select File → New VI or use Ctrl + N

Use Ctrl + T to tile Front Panel and Block Diagram

On the Front Panel:

- Add graphs for Message Signal, PAM Signal, Received Signal, and PSD Spectrum
- Add controls for PAM pulse amplitude, Symbol rate, and Carrier frequency
- Add a Stop button from the Boolean palette

(b) Generate Message Signal and PAM Waveform:

On the Block Diagram, drop a While Loop from Structures

Use Simulate Signal Express VI to create a baseband message signal

Use numeric operations (Multiply, Array Subset, Build Array, etc.) to modulate the message with pulses (implement PAM manually)

PAM equation:

$s(t) = \Sigma\, m(n) * p(t - nT)$

where m(n) is the message and p(t) is the pulse shape (rectangular/sinc)

(c) Transmit PAM Signal Using USRP:

Place a USRP Transmit VI inside the loop from NI-USRP Palette

Configure:

- Carrier Frequency (e.g., 915 MHz)
- Gain, Sample Rate, and TX Channel (e.g., TX1)

Wire the PAM waveform to the USRP TX VI

Use Build Waveform function to form a complex waveform if required by the VI

(d) Receive Signal from USRP:

Place a USRP Receive VI in the same or separate loop

Configure the RX parameters:

- Same carrier frequency, sample rate, and channel (e.g., RX1)

Output the received waveform to a graph on the Front Panel

Optional: Apply a Low-Pass Filter using the Filter Express VI to reduce noise

(e) Compute Power Spectral Density (PSD):

Use the Spectral Measurements Express VI or FFT VI to analyze the frequency domain of the received PAM signal

Connect the output to a graph titled Power Spectral Density

Configure windowing (Hamming/Hanning) and averaging as needed

(f) Final Assembly:
Connect all blocks:
Message Signal → PAM Encoder → USRP TX
USRP RX → Filter → FFT
Connect respective outputs to waveform graphs:
Message, PAM, Received Signal, and PSD
Add Wait (ms) for loop timing and Stop button condition

(g) Run and Observe:
Press "Ctrl+R" to execute or click Run
View all waveforms in real time
Modify parameters like pulse amplitude and symbol rate to observe impact on spectrum
Stop execution with the Stop button
Front Panel Overview:
Controls:
Amplitude, Symbol Rate, Carrier Frequency
Indicators:
Message Signal (Time)
PAM Signal (Time)
Received Signal (Time)
PSD (Frequency Spectrum)
Buttons:
Stop Execution

### Try:

1. Build a VI to perform 4-level PAM transmission and reception using USRP, and estimate the bit error rate. input signal.
2. Implement a square root raised cosine filter to pulse shape the PAM signal and observe its effect on the PSD.

## 13. Exercises on create a system to measure weather parameters and displays them to the user using MYDAQ, MYRIO

### 13.1 Introduction

This experiment introduces students to embedded data acquisition systems using NI myDAQ or NI myRIO. Students will build a LabVIEW-based weather monitoring system that collects environmental data such as temperature, humidity, and light intensity using suitable sensors. The system continuously monitors, displays real-time graphs, and can log data for later analysis.
This hands-on experience demonstrates the synergy of sensors, embedded systems, LabVIEW programming, and data visualization, ideal for IoT, automation, and environmental monitoring applications.

### 13.2 Procedure

(a) Hardware Setup:
Required hardware:
NI myDAQ or myRIO
Sensors:
LM35 (Temperature)
DHT11/DHT22 or SHT15 (Humidity)
LDR (Light)

Breadboard and jumper wires
Optional: BMP180 (Barometric Pressure), Rain Sensor

Wiring example:

LM35 → AI0 (Analog Input)
LDR (via voltage divider) → AI1
DHT11 (Digital I/O) → DIO0 (only with myRIO or DAQmx custom driver)

(b) Launch LabVIEW and Create New VI:

Open LabVIEW → File → New VI (Ctrl + N)
Press Ctrl + T to view Front Panel and Block Diagram side by side
On the Front Panel, add:
Numeric indicators for Temperature (°C), Humidity (%), Light Intensity (lux/level)
Graphs to plot real-time data
A stop button

(c) Read Sensor Data from myDAQ/myRIO:

On Block Diagram:
Drop a While Loop from Structures
Add DAQ Assistant or myRIO Express VI:
For myDAQ:
Select Analog Input → Voltage → Channel AI0 (LM35), AI1 (LDR)
For myRIO:
Use myRIO > I/O > Analog Input (AI0, AI1) blocks
Configure Sample Rate and Acquisition Mode (e.g., 1 sample on demand)

(d) Convert Voltage to Real Units:

Use Numeric functions to convert voltage to real-world units:
LM35: Temp (°C) = Voltage (V) × 100
LDR: Use mapping or resistance formula to estimate lux (approximate)
Humidity (DHT11): Read digital data or via UART if supported

(e) Display and Log the Data:

Connect computed values to indicators and graphs on the Front Panel
Optional:
Use Write to Measurement File Express VI to log data
Use Time Delay (e.g., 500 ms) in loop to control update rate

(f) Stop Condition and Final Run:

Use the Stop button to control the While Loop
Wire all blocks and press Ctrl + R to run the system
Verify that all sensors are reading and displaying correct values
Adjust gain/scaling factors as needed

Front Panel Overview:

Controls:
Sampling rate (optional)

Indicators:
Temperature in °C
Humidity in %
Light Level

Graphs:
Temperature, Humidity, and Light vs Time

Buttons:
Stop System

### Try:

1. Add a digital alert system: If temperature > threshold, trigger a visual/audible alert.
2. Build a weather dashboard using charts and numeric meters for better visualization.

# 14. Home Automation using MYDAQ and MYRIO

## 14.1 Introduction

This lab introduces students to the fundamentals of home automation using NI myDAQ or NI myRIO. The goal is to build a virtual instrument (VI) in LabVIEW that can monitor and control various home appliances (like lights, fans, or alarms) based on sensor inputs (such as temperature, motion, light). Students will implement automation logic and control output devices accordingly.

The lab will demonstrate how to interface digital sensors, read and analyze sensor data in real-time, and control actuators or relays based on programmed conditions, all within the LabVIEW environment.

## 14.2 Procedure

(a) Open LabVIEW and Set Up the VI:

Launch LabVIEW, and go to File → New VI or press Ctrl + N.

Press Ctrl + T to split view between Front Panel and Block Diagram.

In the Front Panel, add:

- Indicators for temperature, motion status, light level, etc.
- Boolean switches or LEDs to control/display device status (e.g., Fan ON/OFF, Light ON/OFF).
- A stop button to end the program.
- Optional: Thermometer and LED indicators for better UX.

(b) Create a While Loop for Continuous Monitoring:

On the Block Diagram, place a While Loop from the Structures palette.

Create input/output channels using:

- DAQ Assistant for myDAQ (Analog Input/Output or Digital I/O)
- myRIO Express VIs or I/O > MXP Connector options for myRIO

Typical sensor connections:

- Temperature (LM35/thermistor) → AI0
- LDR for light detection → AI1
- PIR motion sensor → DIO0
- Relay or LED for fan/light → DIO1 or DO pin

(c) Read Sensor Data:

Add DAQ Assistant or myRIO Analog Read Express VIs:

- Read temperature (AI0) → Convert voltage to temperature (e.g., Temp = V × 100 for LM35)
- Read light level (AI1) → Simple scaling or threshold
- Read motion sensor digital input → Boolean status

(d) Automation Logic:

Use case structures, comparison blocks, and Boolean logic:

- Example logic:
  - If Temp > 30°C → Turn Fan ON
  - If LDR value < threshold → Turn Light ON
  - If Motion Detected → Turn on security light/alarm

Connect the result of logic to Digital Output blocks for controlling devices.

(e) Display Values and Control Output:
Send sensor readings to appropriate indicators.
Drive Boolean indicators for device status (Light ON/OFF, Fan ON/OFF).
Optionally include timing using Wait(ms) or delay functions for realistic update cycles.

(f) Stop the Program Gracefully:
Connect the stop button from Front Panel to the condition terminal of the While Loop.
Run the VI using Ctrl + R or the Run arrow.
Monitor all indicators and verify the correct automation responses.
Front Panel Overview:
Indicators:
- Temperature (°C)
- Motion detected (Yes/No)
- Light level (lux or intensity level)

Device Status LEDs:
- Fan ON/OFF
- Light ON/OFF
- Alarm ON/OFF

Controls:
- Threshold values (Temp, Light)
- Stop Button

## Try:

1. Expand to a multi-room system with multiple sensors and control zones.
2. Use mobile-based UI or web publishing tool to remotely monitor/control devices.

## V. TEXT BOOKS:


1. Jim Kring, Jeffrey Travis, “LabVIEW for Everyone: Graphical Programming Made Easy and Fun”,
2. Prentice Hall, 3rd Edition, 2006.
3. Richard Jennings Gary W.Johnson, “Labview Graphical Programming”, McGraw-Hill Education, 4th Edition, 2011
4. Douglas Rick Bitter, Taqi Mohiuddin, Matt Nawrocki, “LabView: Advanced Programming Techniques”, CRC Press, 2nd Edition, 2006.


## VI. REFERENCE BOOKS:


1. Sanjay Gupta, “Virtual Instrumentation using LABVIEW”, McGraw-Hill Education, 2nd Edition, 2010.


## VIII. MATERIAL ONLINE:

1. Course Outline Description
2. Lab Manual