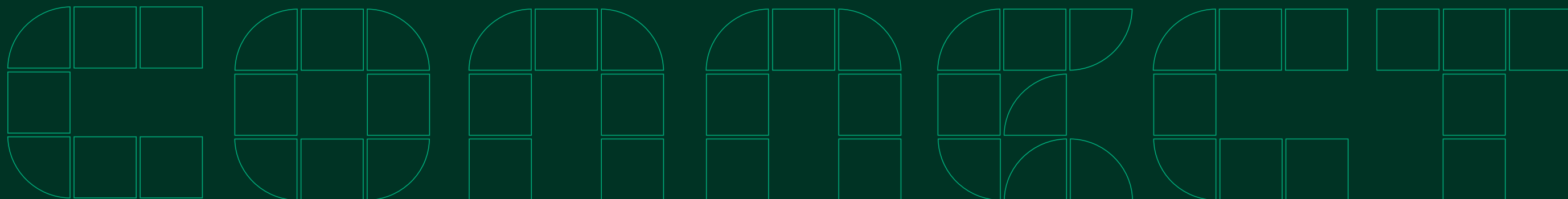


LabVIEW AI Assistant Tech Preview

Concept Exploration

LabVIEW Generative AI Early
Access Request

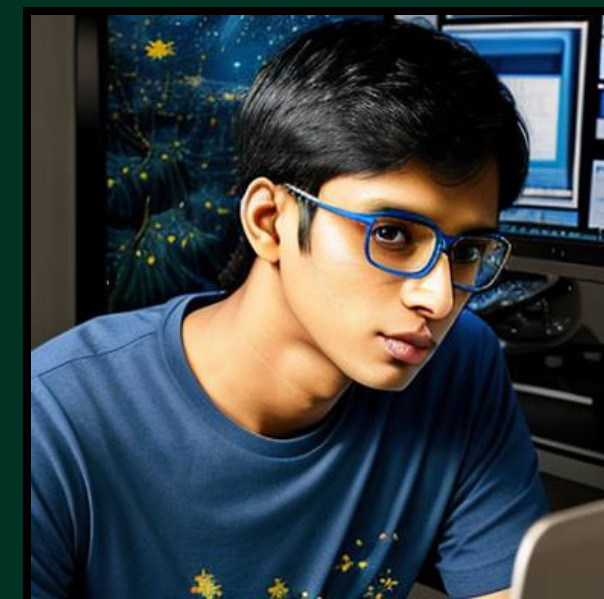




Sneak Peak: Generative AI & LabVIEW



John Bongaarts
Chief AI Technology Strategist



Asrar Rangwala
Principal Software Engineer

*Avatars crafted via prompt engineering
by text-to-image AI at dezgo.com*

AI SESSIONS

All Days

- **LabVIEW AI Assistant Demo** Expo Floor

Tuesday, May 21

- **Day 1 Keynote** 9:00 AM in the Keynote Hall (Exhibit Hall 3)
- **Sneak Peek: Generative AI and LabVIEW** 11:30 AM – 12:30 PM in Meeting Room 19A
- **Generative AI to Accelerate Test Workflows** 2:00 PM – 3:00 PM in Meeting Room 16A

You are here!

Wednesday, May 22

- **Day 2 Keynote** 9:00 AM in the Keynote Hall (Exhibit Hall 3)
- **Panel: How will AI/ML Impact the Way We Engineer & Test?** 10:15 AM – 11:15 AM in Meeting Room 3
- **AI at the Edge** 1:30 PM – 2:30 PM in Meeting Room 16A

AGENDA

- Generative AI
 - Background
 - Responsible AI Use in Test & Measurement
- LabVIEW AI Assistant Tech Preview
 - Our Goal for the Program
 - Overview of Capabilities
 - How It Works - Using Generative AI Building Blocks
- Further Concept Explorations

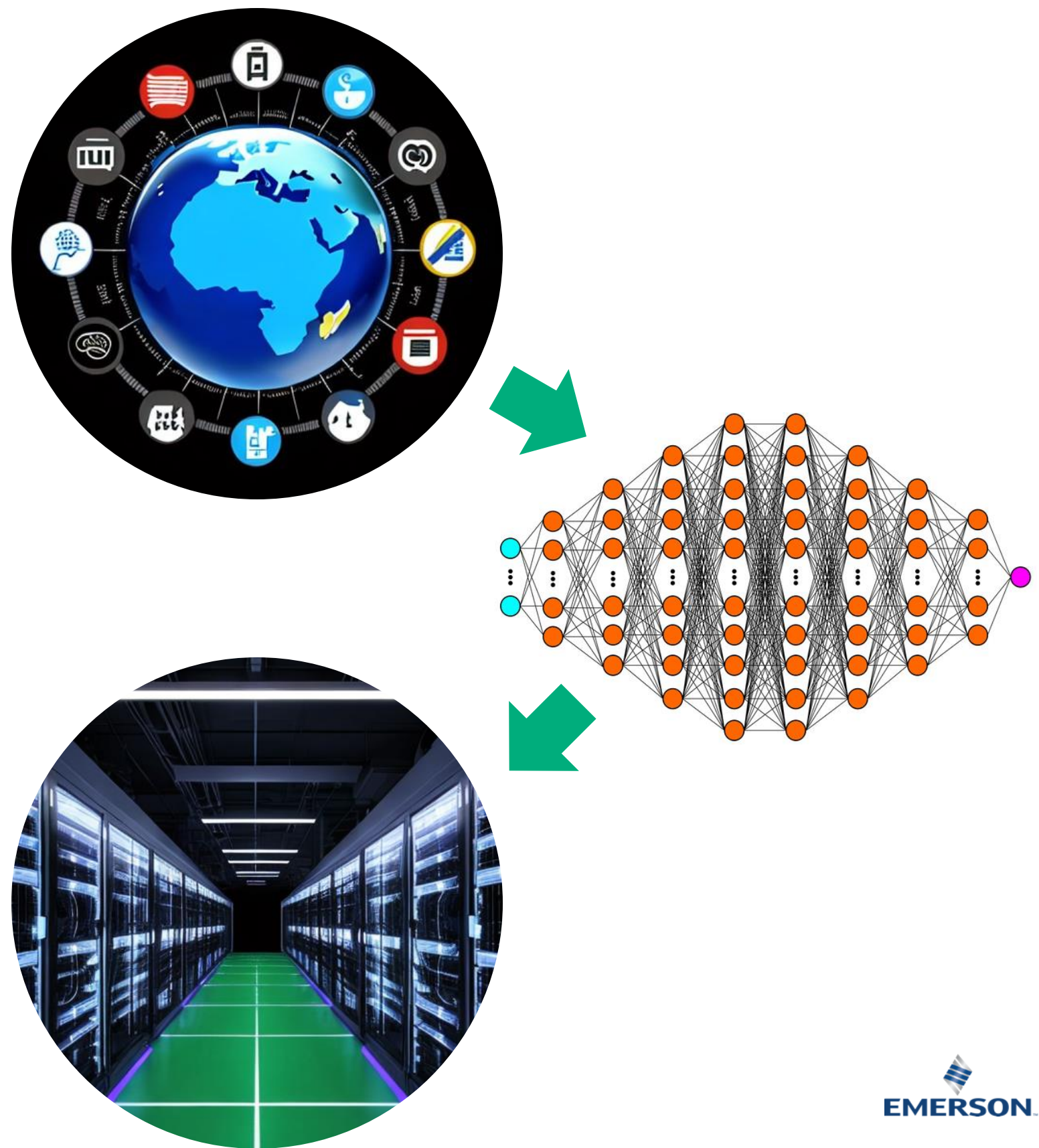
Generative AI Background

What changed?

Pre-training scale

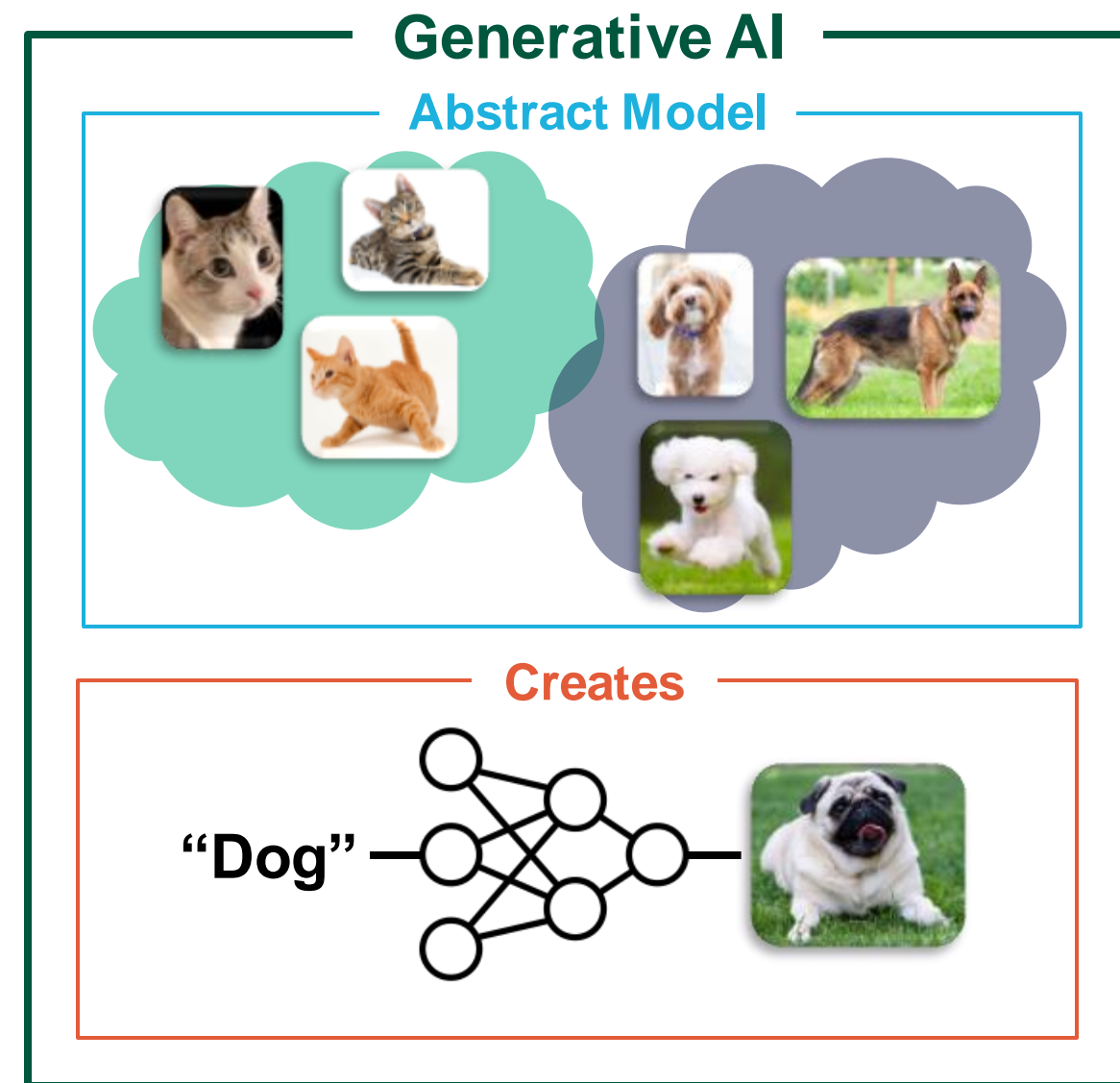
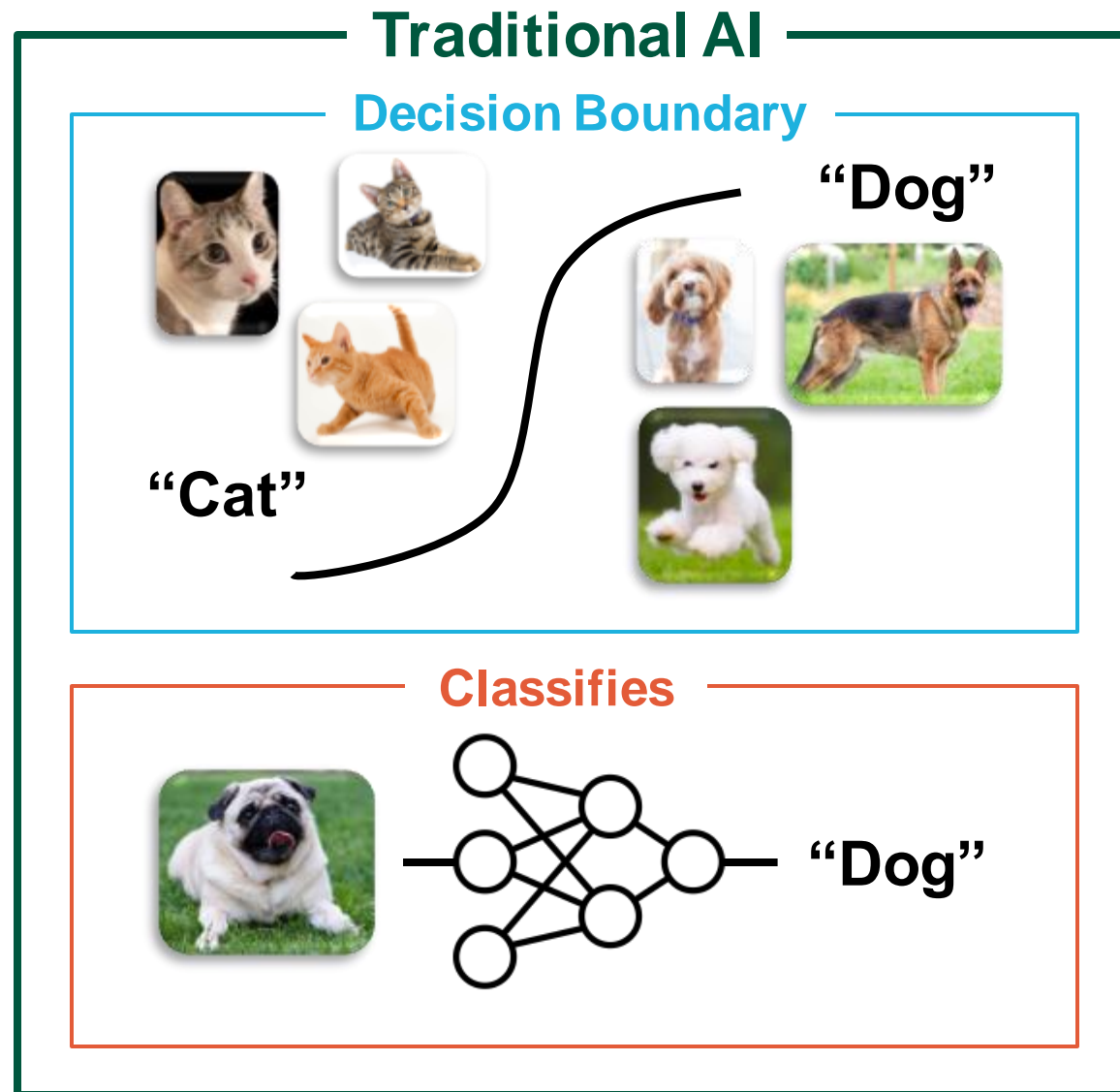
Neural network architecture

Neural network scale



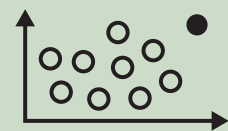
Traditional AI vs. Generative AI

“What I cannot create, I do not understand.” – Richard Feynman



Generative AI can also classify

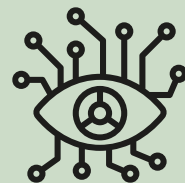
"Traditional" AI Example Use Cases



Outlier detection
(Advanced)



Escape prevention
(Equipment Health etc.)



Optical defect detection
/ ROI quality (e.g.,
welds, soldering etc.)



RMA reduction /
prediction



Parametric trend
detection



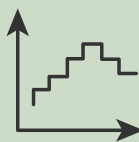
Parametric insights



Scratch detection
(wafers)



AI based RCA



Waveform anomaly
detection



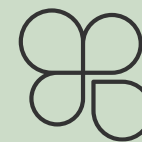
Battery analytics
(capacity, smart pairing,
etc.)



Process optimization
(e.g., Adaptive
manufacturing)



Early failure detection



Next operation reduction



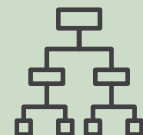
Monitoring and auto
RCA (UPH, cycle time,
yield, error code
distributions etc.)



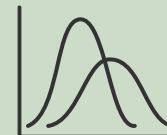
Equipment
utilization/variations



Yield trend detection



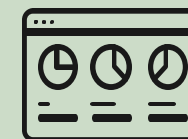
Wafer classification



Equipment
utilization/variations



Predictive / JIT
maintenance

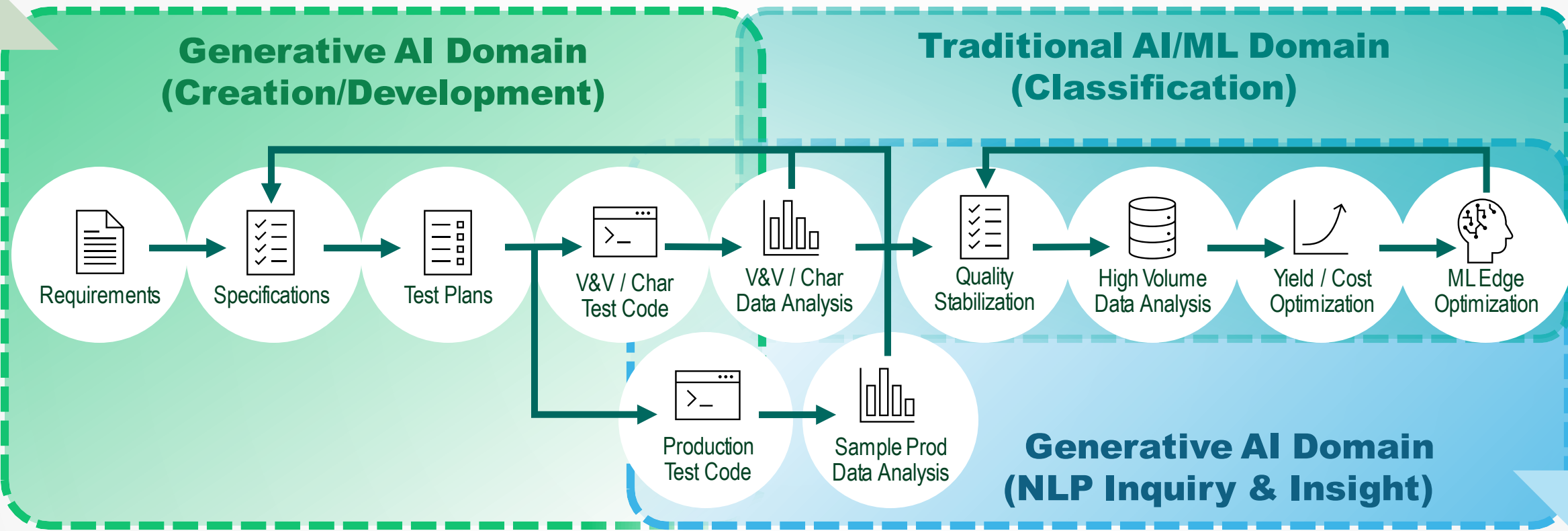


Test (program)
comparison

AI at the Edge – Tomorrow - 1:30 PM – 2:30 PM in Meeting Room 16A

Business Impact of Generative AI in Test Engineering

Test & Manufacturing Release Process



Test & Manufacturing Optimization Process

Responsible AI Practices

Managing Data Security, Privacy, Legal, and Ethics Risks with an AI Governance Strategy



Establish **Policies** aligned with industry best practices for managing the inherent risks of AI use



Develop and conduct **Responsible AI training** with employees who use AI in their work



Conduct **Risk Assessments** for AI use cases in potential products, services, offerings, and internal development and process optimizations



NI created its AI Steering committee in 2022, and since the acquisition has integrated its AI governance structure with Emerson.

The Emerson AI governance team (including NI) owns and drives development and continuous improvement of our AI policies, training, and risk assessment processes.

Data protection is fundamental to our approach

Your data is your data

We've considered this in every step of our designs

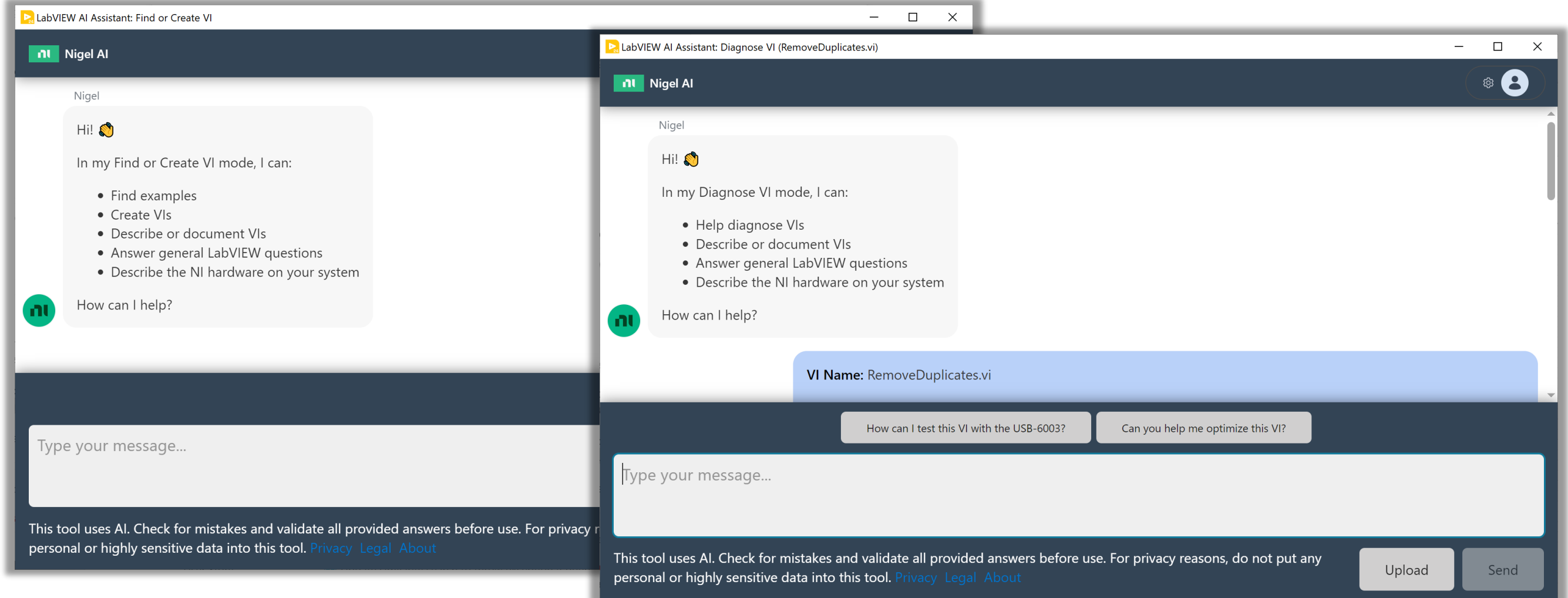
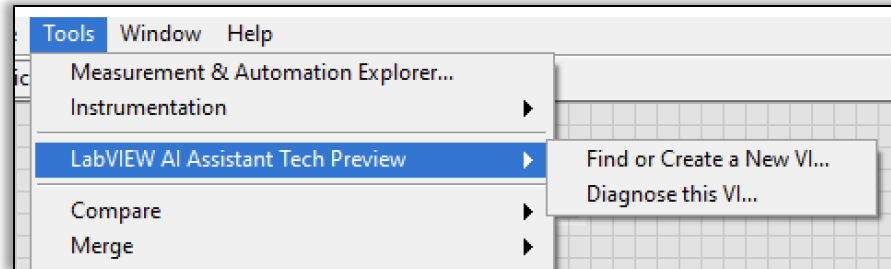
LabVIEW AI Assistant Tech Preview

LabVIEW Generative AI Early
Access Request

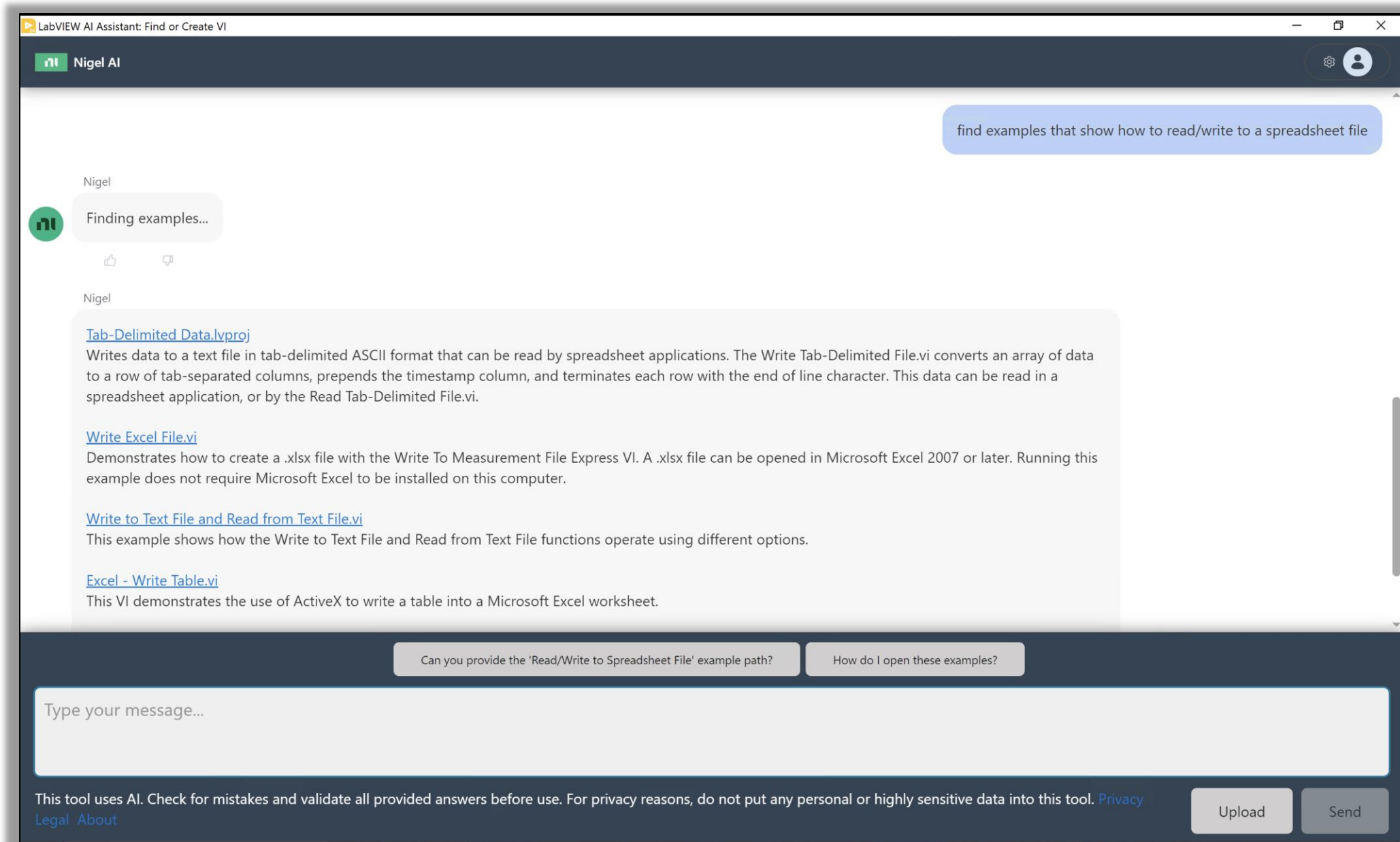


LabVIEW AI Assistant – Operates in two modes

Launch from LabVIEW



LabVIEW AI Assistant can Find Examples



LabVIEW AI Assistant can Create VIs

LabVIEW AI Assistant: Find or Create VI

Nigel AI

Write a VI that removes duplicates from an array of numbers

Nigel

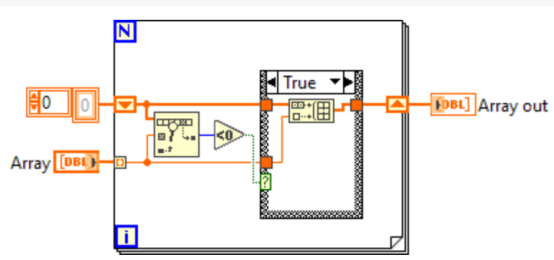
Creating VI...

15s

Nigel

VI Name: RemoveDuplicates.vi

VI Description: This VI takes an input array and processes it to remove any duplicate elements, outputting an array with unique elements only. It begins by initializing an empty array and then iterates through each element of the input array. For each element, it checks if the element already exists in the output array. If the element is not found (i.e., its index is less than 0), it is added to the output array. This process ensures that the output array contains only one instance of each unique element from the input array.



[Open the VI in LabVIEW](#)

How do I handle duplicates with non-numeric arrays?

Can you explain how the 'Search1DArray' function works?

Type your message...

This tool uses AI. Check for mistakes and validate all provided answers before use. For privacy reasons, do not put any personal or highly sensitive data into this tool. [Privacy](#) [Legal](#) [About](#)

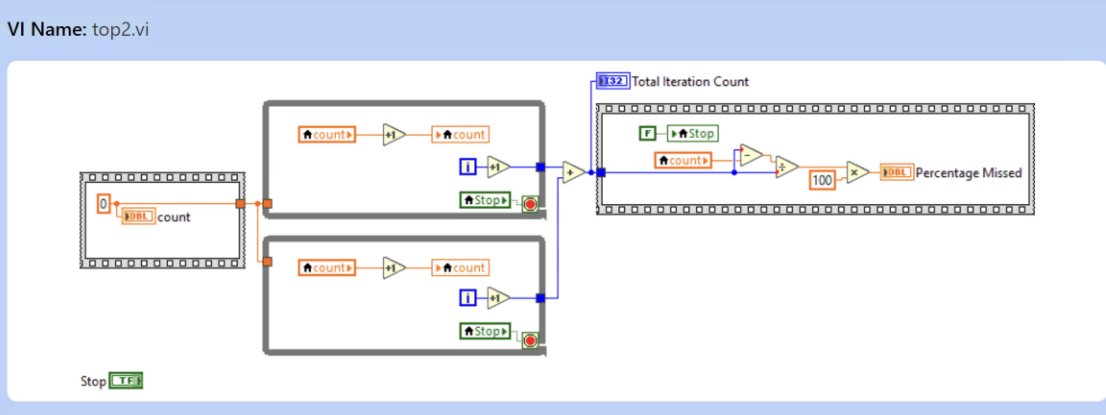
Upload Send

LabVIEW AI Assistant can Diagnose VIs

LabVIEW AI Assistant: Diagnose VI (top2.vi)

Nigel AI

VI Name: top2.vi



Stop

Nigel

I don't see any runtime errors in this VI.

my VI isn't working as expected, any ideas?

Nigel

If your LabVIEW VI isn't working as expected, let's perform a systematic review of the logical flow and individual components to troubleshoot potential issues. Here are some steps and questions to help you identify and resolve common problems:

- 1. Initial Values:**
 - Confirm that the initial value of 'Stop' (T33:false) is correctly initializing to not stop the loop.
 - Verify that the initial value of 'count' (T10:0) is set properly.
- 2. Sequence Structure:**
 - Ensure that all data operations in your sequence structure are being executed in the order that you expect.
- 3. While Loops:**
 - Check if both While Loops have appropriate terminal conditions based on 'Stop', which should be linked with a Local variable pointing at the control named "Stop".
 - Ensure that each increment within both While Loops increments correctly, without any race conditions or unintended modifications from other parts of the code.
- 4. Data Flow and Race Condition:**
 - In LabVIEW, data flows through wires from one node to another, ensuring determinism in execution order. However, when using local variables (to

LabVIEW AI Assistant can Describe VIs

The screenshot displays the LabVIEW AI Assistant interface. At the top, a dark blue header bar contains the 'ni' logo and the text 'Nigel AI'. Below this, a chat window is open, showing a conversation with 'Nigel'. The chat window has a title bar with 'ni' and 'Nigel AI', and a close button. The chat area shows a user's question: 'Can you describe the algorithm in more detail?'. Below this, a response from 'Nigel' is displayed, detailing the components of a VI named 'Triggering TTL PS and DMM.vi'. The response is structured into five numbered sections: 1. Initialization, 2. Sequence Structure, 3. Parsing Input Parameters, 4. Resource Mapping, and 5. Device Initialization. Each section contains a list of bullet points describing the specific steps and parameters involved. At the bottom of the chat window, there is a text input field with the placeholder 'Type your message...' and a 'Send' button. Below the chat window, a dark blue footer bar contains a disclaimer: 'This tool uses AI. Check for mistakes and validate all provided answers before use. For privacy reasons, do not put any personal or highly sensitive data into this tool.' followed by links for 'Privacy', 'Legal', and 'About'. There are also 'Upload' and 'Send' buttons in the footer.

Nigel AI

Can you describe the algorithm in more detail?

Nigel

Certainly! The given VI, "Triggering TTL PS and DMM.vi," describes a complex sequence of operations aimed at setting up a test that involves power supplies (PS) and digital multimeters (DMM). This is a high-level overview of the algorithm based on the provided components:

- 1. Initialization:**
 - Constants are set for various parameters like reset signals, error messages, device names, driver type etc.
 - Input parameters are taken from front panel controls such as file paths and starting error clusters.
- 2. Sequence Structure:**
 - A Sequence structure ensures operations occur in an explicit order.
 - It starts by processing any input errors followed by configuring the Measurement Graph.
- 3. Parsing Input Parameters:**
 - VIs such as "Input Parameter Parser.vi" parse the input parameters to extract information like remaining parameters, independent channels etc.
- 4. Resource Mapping:**
 - With extracted parameter details, other VIs map resources to obtain device names ("Alias"), chassis numbers etc.
- 5. Device Initialization:**
 - Devices are initialized using "Initialize Channel Switch.vi" & "niDMM Initialize.vi".
 - Conditions check if devices support specific features ("Is Atomic SMU?").

Type your message...

This tool uses AI. Check for mistakes and validate all provided answers before use. For privacy reasons, do not put any personal or highly sensitive data into this tool. [Privacy](#) [Legal](#) [About](#)

Upload Send

LabVIEW AI Assistant can Help with VIs

The screenshot displays the LabVIEW AI Assistant interface. At the top, the header shows the NI logo and the name 'Nigel AI'. A user query is entered in a blue bubble: 'I would like to add a triggered scope acquisition to this VI, can you tell me what changes I would need to make?'. The response from Nigel AI is displayed in a white bubble with a green NI logo icon. The response provides a structured outline of changes required for adding a triggered scope acquisition to a VI.

Nigel

To add a triggered scope acquisition to this VI, you would need to integrate additional steps into the existing test sequence or possibly create a parallel sequence depending on how you want the scope data to relate timing-wise with the rest of your test. Here's an outline of changes that might be required:

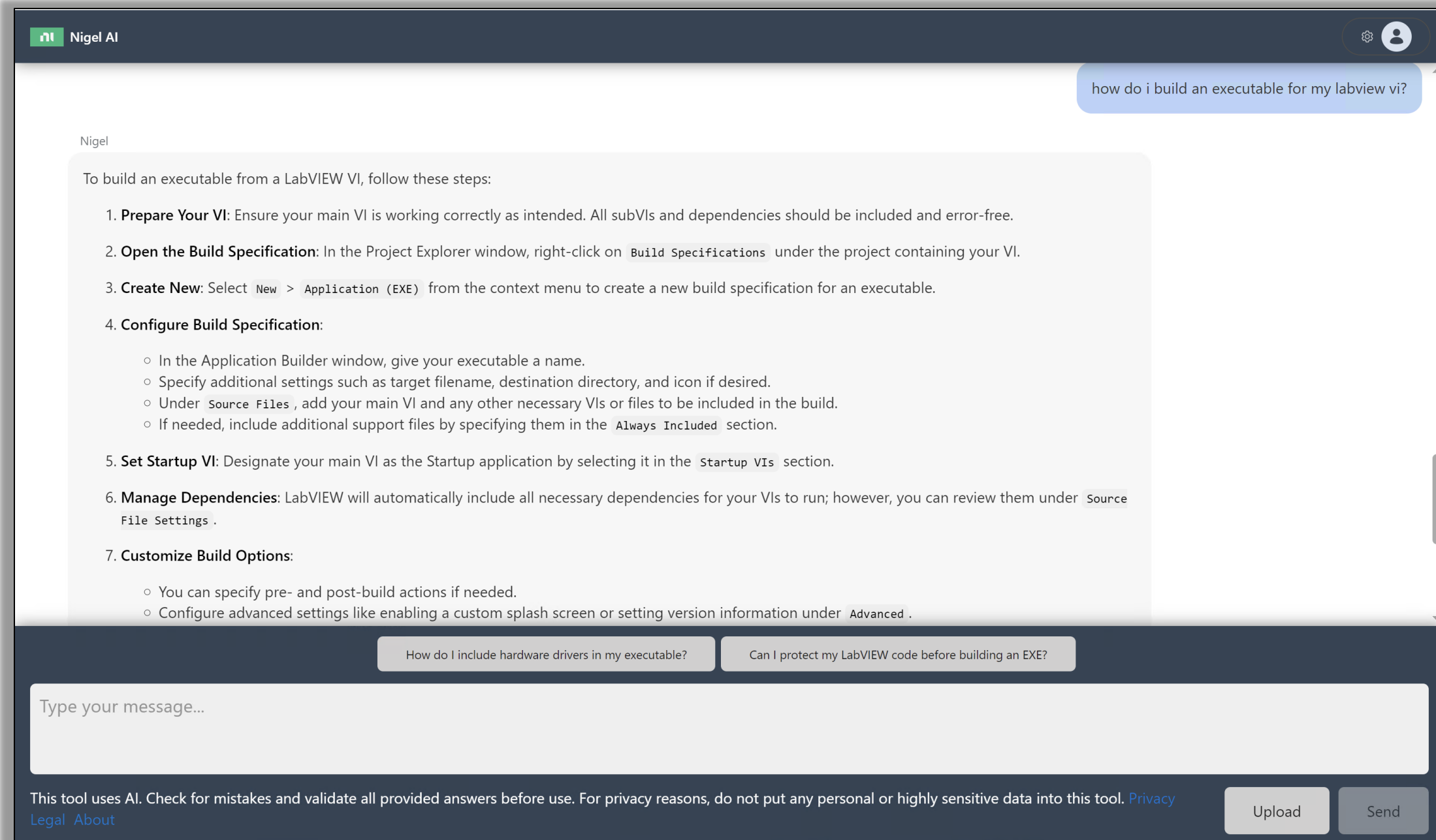
- 1. Hardware Configuration:**
 - Ensure that your oscilloscope hardware is configured correctly in NI Measurement & Automation Explorer (NI MAX) and has an appropriate device alias.
 - If using NI hardware such as a PXI Oscilloscope card, ensure it has been added and configured in the system.
- 2. Scope Initialization:**
 - Add a VI for initializing your oscilloscope which may include setting up vertical range, horizontal timebase settings, trigger levels, etc. This might be using `niScope Initiate.vi`, if available for your specific scope.
- 3. Synchronization:**
 - If the scope acquisition needs to be synchronized with other measurements (PS or DMM), you will need to configure triggering appropriately.
 - Set up shared triggers between devices if required (for example using PFI lines on PXI systems).
 - Use property nodes or specific VIs like `niScope Configure Trigger.vi` for setting up trigger source, edge/level triggering options etc.
- 4. Acquisition Loop / Event Structure:**
 - You may need an event structure or loop within your VI where you wait for acquisition complete events from the oscilloscope.
 - Once acquired, retrieve waveform data typically with a `niScope Fetch.vi` call.
- 5. Data Handling:**
 - Handle retrieved waveform data--displaying it on front panel controls/graphs as needed, processing it further for analysis metrics or saving it out t ■

Type your message...

This tool uses AI. Check for mistakes and validate all provided answers before use. For privacy reasons, do not put any personal or highly sensitive data into this tool. [Privacy](#) [Legal](#) [About](#)

Upload Send

LabVIEW AI Assistant can Help with LabVIEW



The screenshot displays the LabVIEW AI Assistant interface. At the top, a dark blue header bar contains the 'Nigel AI' logo on the left and a settings/user icon on the right. A chat input box in the top right corner contains the text 'how do i build an executable for my labview vi?'. The main chat area shows a response from 'Nigel' with a list of seven steps for building an executable. Below the chat area, there are two suggested questions in grey buttons: 'How do I include hardware drivers in my executable?' and 'Can I protect my LabVIEW code before building an EXE?'. At the bottom, there is a large text input field with the placeholder 'Type your message...'. Below the input field, a disclaimer states: 'This tool uses AI. Check for mistakes and validate all provided answers before use. For privacy reasons, do not put any personal or highly sensitive data into this tool.' followed by links for 'Privacy', 'Legal', and 'About'. On the right side of the disclaimer, there are 'Upload' and 'Send' buttons.

Nigel AI

how do i build an executable for my labview vi?

Nigel

To build an executable from a LabVIEW VI, follow these steps:

1. **Prepare Your VI:** Ensure your main VI is working correctly as intended. All subVIs and dependencies should be included and error-free.
2. **Open the Build Specification:** In the Project Explorer window, right-click on **Build Specifications** under the project containing your VI.
3. **Create New:** Select **New > Application (EXE)** from the context menu to create a new build specification for an executable.
4. **Configure Build Specification:**
 - In the Application Builder window, give your executable a name.
 - Specify additional settings such as target filename, destination directory, and icon if desired.
 - Under **source Files**, add your main VI and any other necessary VIs or files to be included in the build.
 - If needed, include additional support files by specifying them in the **Always Included** section.
5. **Set Startup VI:** Designate your main VI as the Startup application by selecting it in the **Startup VIs** section.
6. **Manage Dependencies:** LabVIEW will automatically include all necessary dependencies for your VIs to run; however, you can review them under **Source File Settings**.
7. **Customize Build Options:**
 - You can specify pre- and post-build actions if needed.
 - Configure advanced settings like enabling a custom splash screen or setting version information under **Advanced**.

How do I include hardware drivers in my executable?

Can I protect my LabVIEW code before building an EXE?

Type your message...

This tool uses AI. Check for mistakes and validate all provided answers before use. For privacy reasons, do not put any personal or highly sensitive data into this tool. [Privacy](#) [Legal](#) [About](#)

Upload Send

LabVIEW AI Assistant can Tell you about your NI Hardware and how to use it

LabVIEW AI Assistant: Find or Create VI

Nigel AI

Nigel

You have a variety of NI hardware available on your system, including devices for data acquisition. Here is a summary of the hardware:

1. **NI USB-6003:** A simulated multifunction DAQ device with 8 analog input channels (Dev1/ao1). It is accessible through the device alias Dev1.
2. **roboRIO:** You have two roboRIO devices which are real-time embedded controllers.
3. **PXIe-8135:** There are three PXIe-8135 controllers in your system which are high-performance instrumentation and data acquisition systems.
4. **cRIO (CompactRIO):** You have several CompactRIO systems including cRIO-9048 (with variant cRIO-903x-VM). These are rugged industrial-grade systems used for advanced control and data acquisition.
5. **sbRIO (Single-board RIO):** The sbRIO-9629 is an embedded control and acquisition device.
6. **cRIO 9075:** This is another CompactRIO controller listed under the alias rto11ert-f10.

Each of these devices serves different purposes ranging from simple data logging to complex control systems using FPGAs.

13s

How can I simulate an analog input signal on the USB-6003?

Type your message...

This tool uses AI. Check for mistakes and validate all provided answers before use. For privacy reasons, do not put any personal or highly sensitive data into this tool. [Privacy](#) [Legal](#) [About](#)

LabVIEW AI Assistant: Find or Create VI

Nigel AI

how can i use the daq usb hardware that i have to acquire analog data?

Nigel

Finding examples...

[Voltage - Continuous Input.vi](#)
This example demonstrates how to continuously acquire voltage measurement using a DAQmx device.
For instructions on how to connect your signals to the DAQ device, refer to your device documentation.
This example can also run on cRIO-904x. For information on how to port this example to a 904x project, go to ni.com/info and enter Info Code daq2comboExample.

[Current - SW-Timed Input.vi](#)
This example demonstrates how to continuously acquire buffered current measurements using software-timing using a DAQmx device.
For instructions on how to connect your signals to the DAQ device, refer to your device documentation.
This example can also run on cRIO-904x. For information on how to port this example to a 904x project, go to ni.com/info and enter Info Code daq2comboExample.

[Voltage \(with Events\) - Continuous Input.vi](#)

How do I generate a sine wave in LabVIEW? Can you show me how to read the simulated data?

Type your message...

This tool uses AI. Check for mistakes and validate all provided answers before use. For privacy reasons, do not put any personal or highly sensitive data into this tool. [Privacy](#) [Legal](#) [About](#)

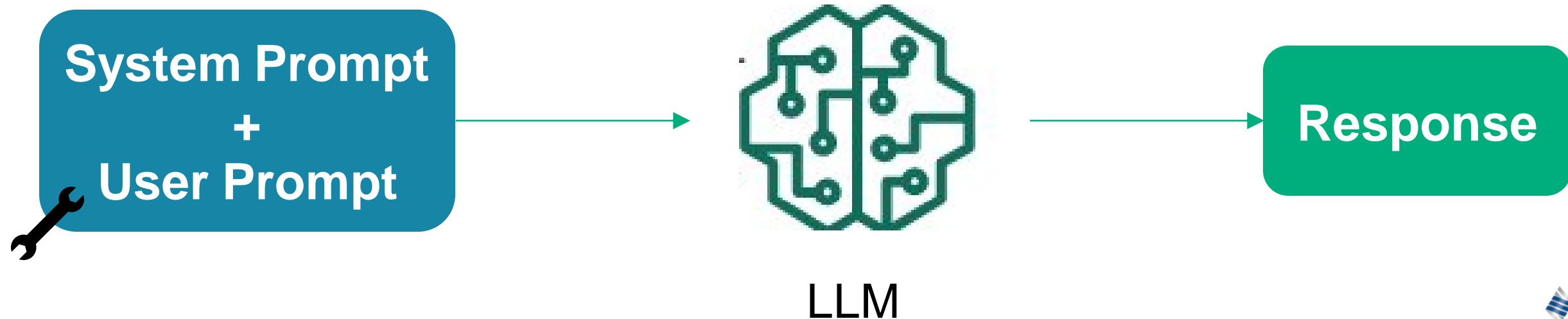
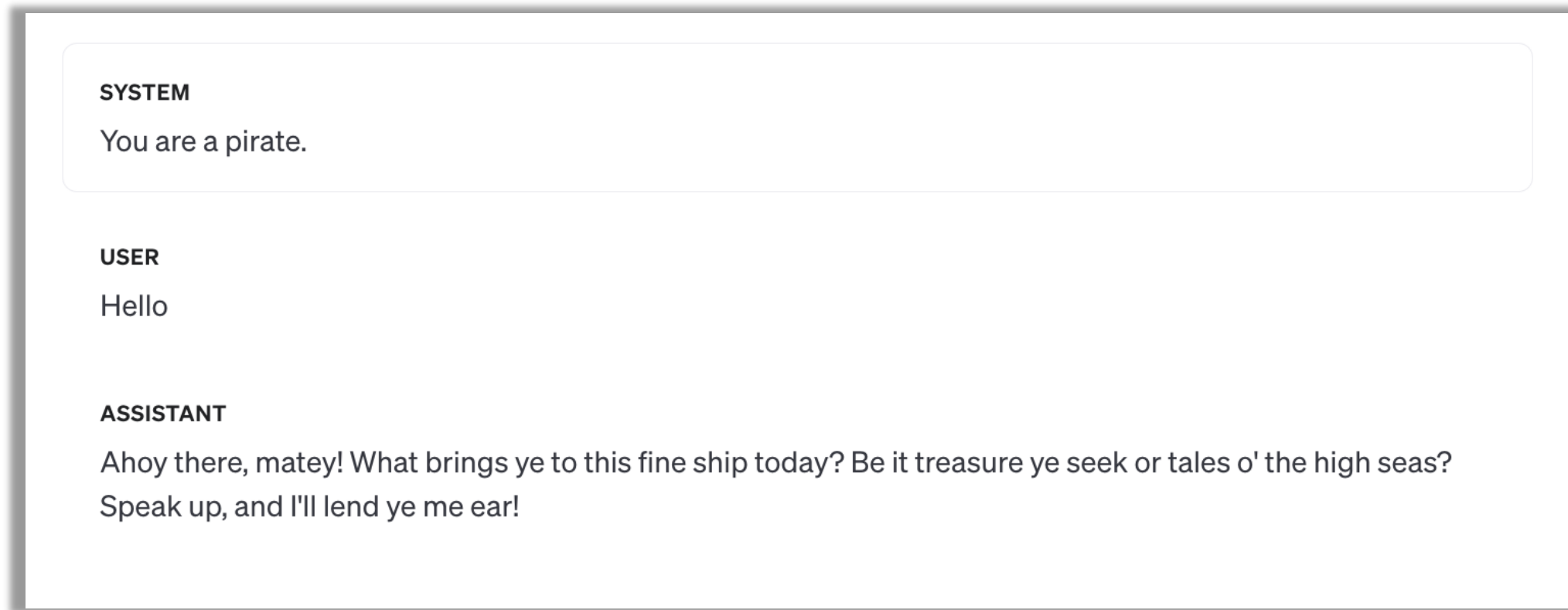
Upload Send

LabVIEW AI Assistant

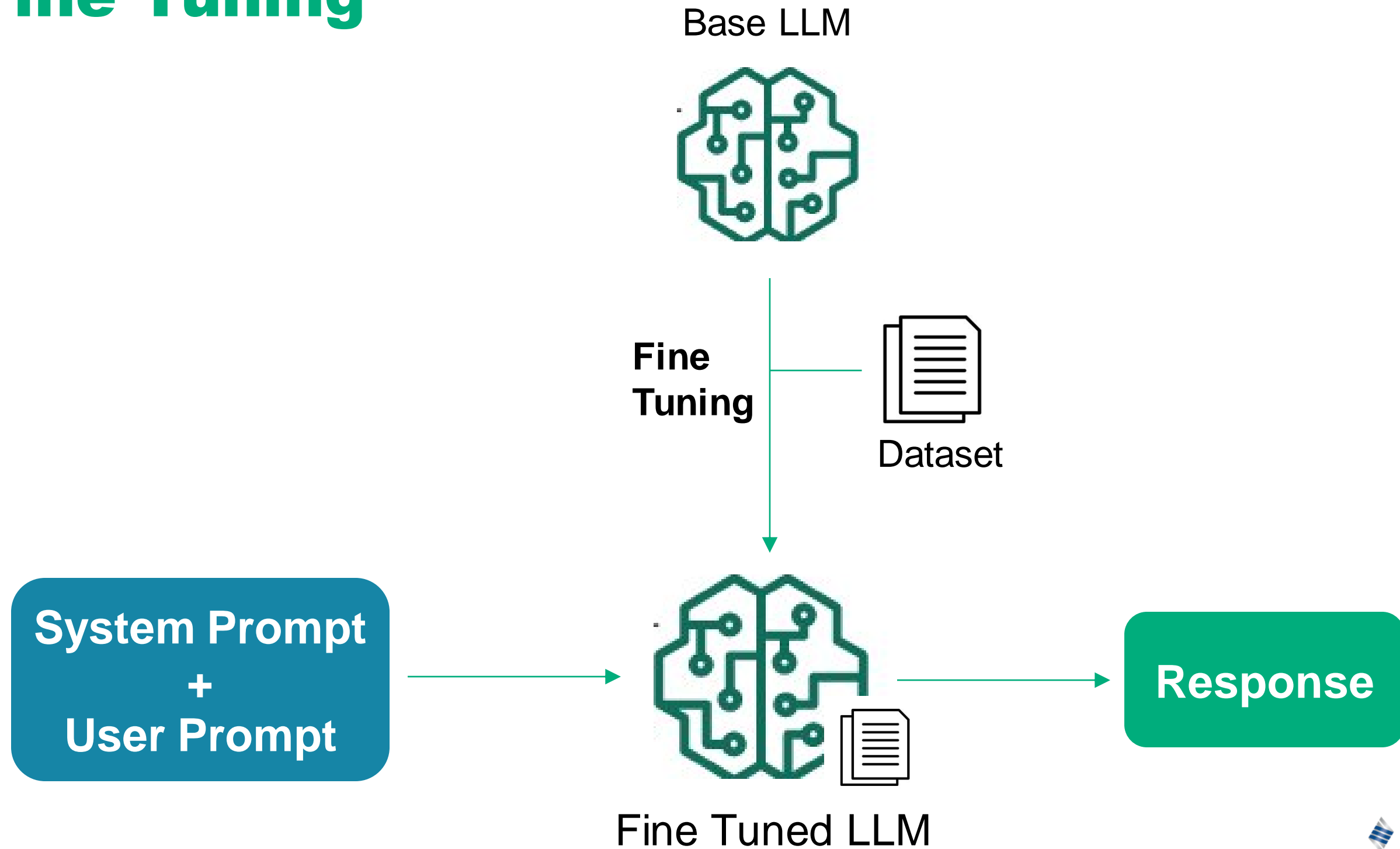
How does it work?

What Language is Nigel Programmed in?

Prompt Engineering

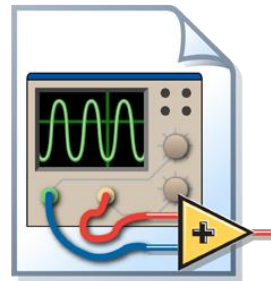


Fine Tuning



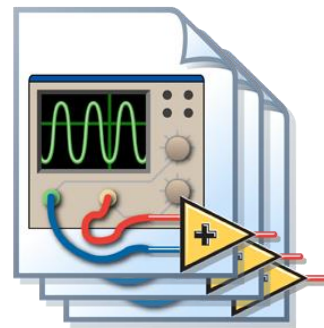
Text models can not understand VIs

Define intermediate format



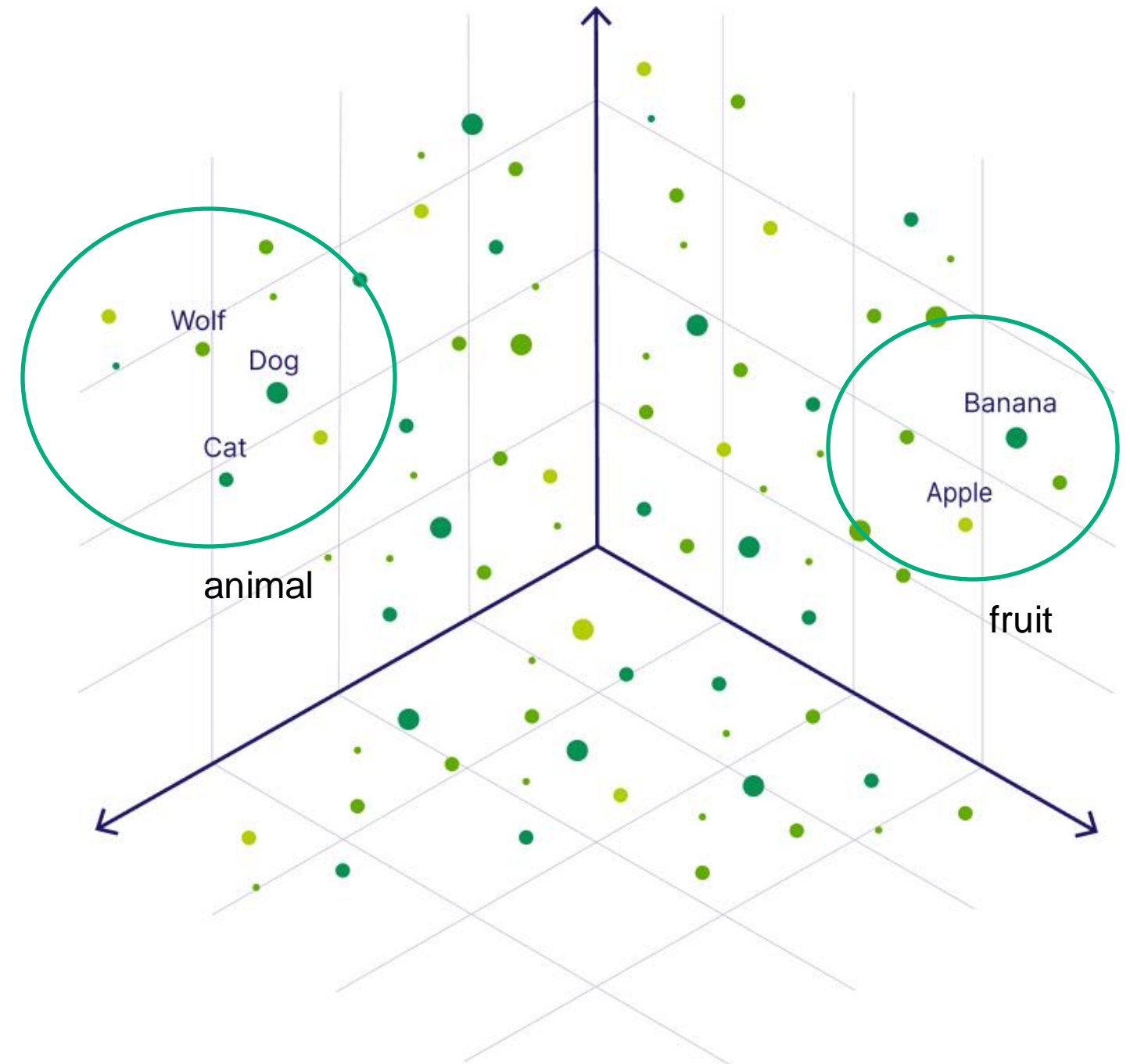
Text

Generate training material



Semantic Search using Vector DB

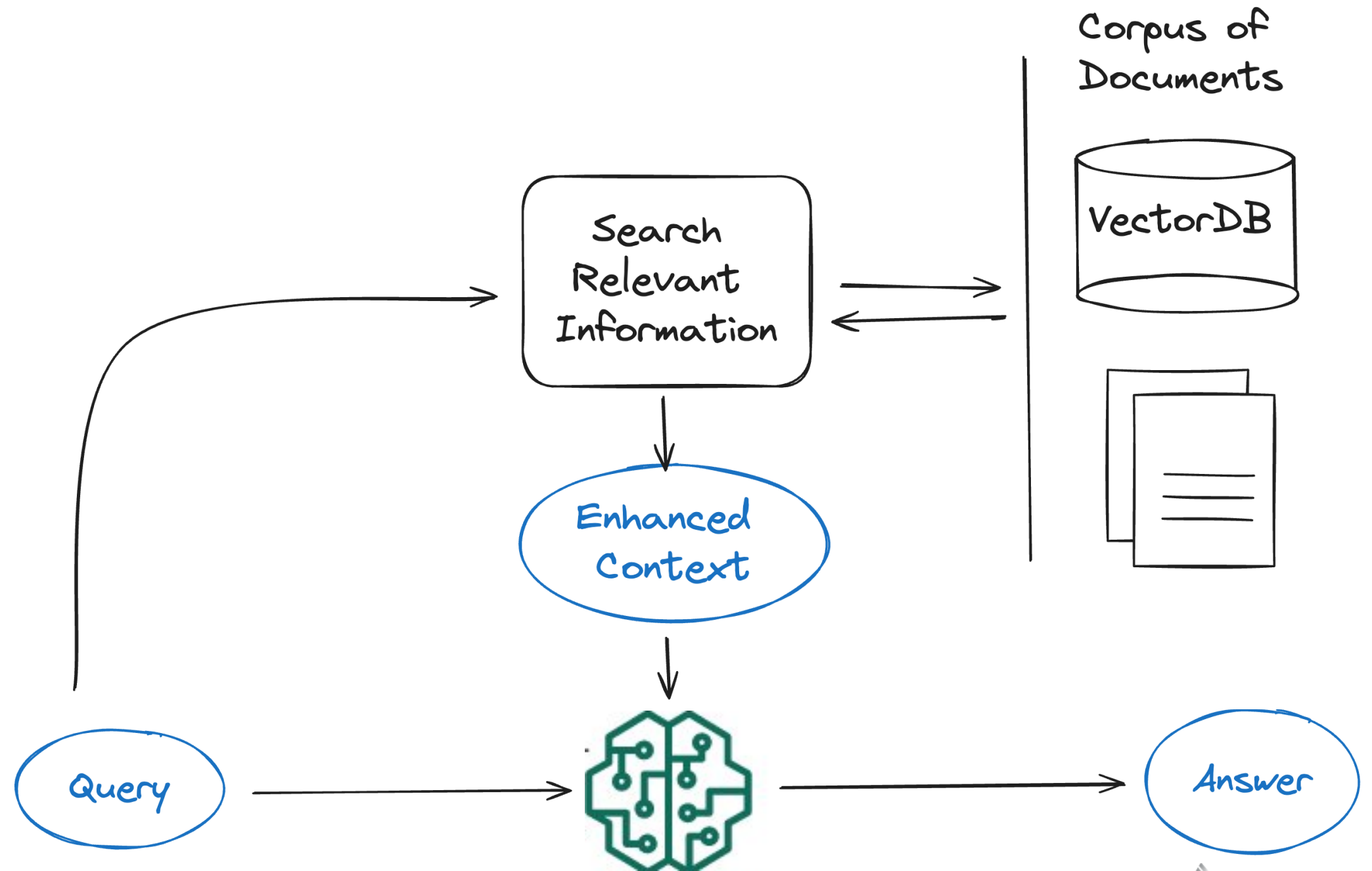
**Search your data using an
AI backed Search engine**



Source: <https://weaviate.io/blog/what-is-a-vector-database>

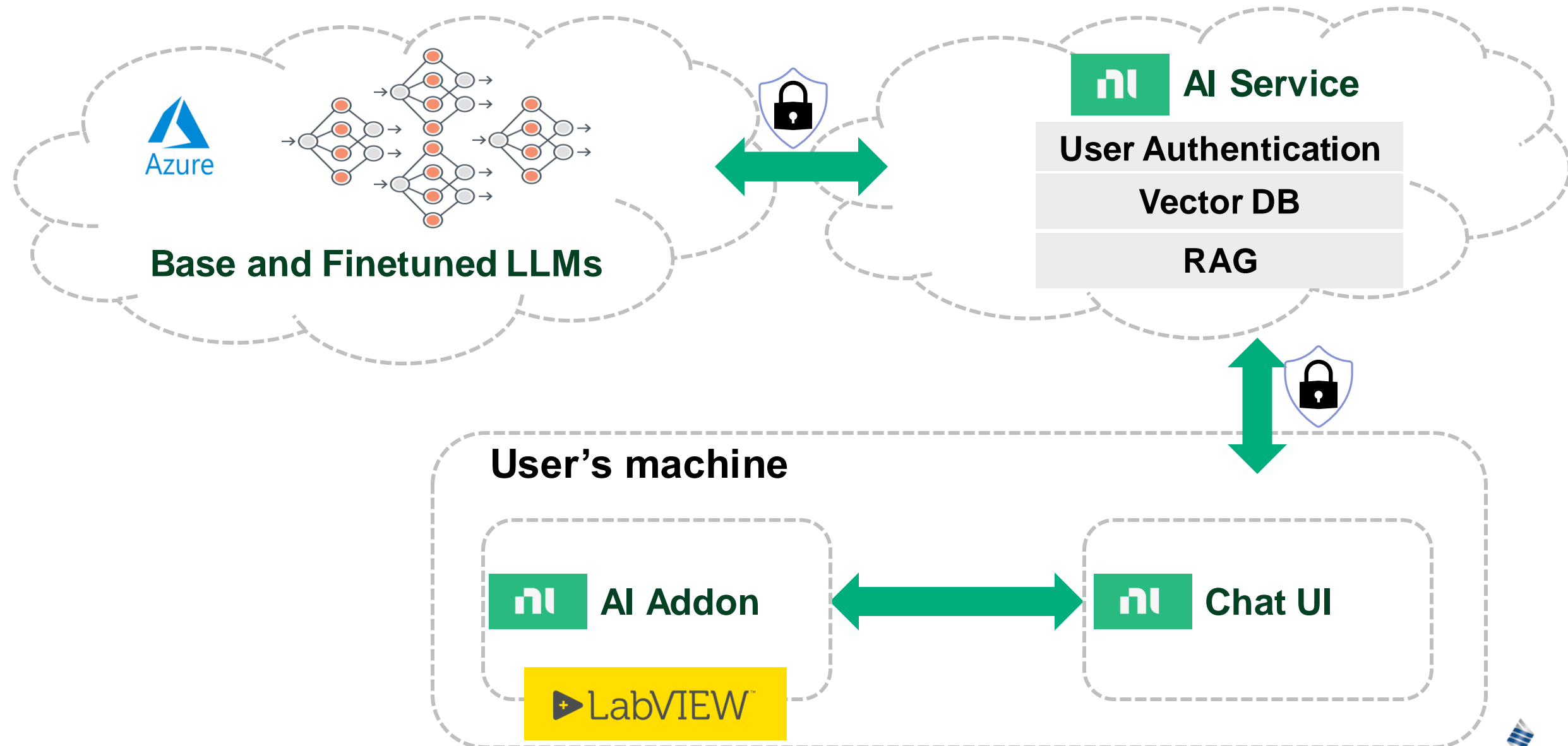
Retrieval Augmented Generation (RAG)

Chat and Q&A
with your data



LabVIEW AI Assistant Tech Preview

Architecture



The Development Process

New Kinds of Challenges

Evaluating Code Generation

Testset: OpenAI's HumanEval



164 programming challenges with an average of 7.7 tests per problem

Example:

- "prompt": "def strlen(string: str) -> int: Return length of given string
>>> strlen('') 0 >>> strlen('abc') 3"
- "canonical_solution": "return len(string)",
- "test": "def check(candidate):\n assert candidate('') == 0\n assert candidate('x') == 1\n assert candidate('asdasnakj') == 9"

Metric: Pass@K

What is the probability that at least one of the K generated responses will be correct (generated code passes unit tests)?

Challenges with Gen AI Development

- Find the right technique for a task – Semantic Search, Prompt Engineering, RAG, Fine Tuned Model
- Find the right set of LLM parameters – temperature, top_p, frequency_penalty, etc.
- Rapidly changing AI stacks
- Data Curation
- Optimizing for token count

Challenges with Gen AI Testing

- Difficult to perform qualitative assessment of results
- Chat interface is open ended
- Handling corner-cases like hallucinations

Ensure each step above reflects accurate settings pertinent to your specific hardware setup and intended measurement requirements before running this VI in an actual LabVIEW environment on target hardware systems connected properly according to their documentation guidelines.

If after checking all points above you still encounter issues during execution (such as an error code or unexpected behavior), please provide detailed information about what occurs when you run this VI including error codes/messages displayed by LabVIEW which would help me give more specific advice tailored towards resolving those particular problems encountered during use of this virtual instrument file within context of NI software platform utilities alongside associated hardware apparatuses deployed therein for carrying out automated testing/measurement procedures accordingly as part thereof overall system integration workflow processes involved thereby overall end-to-end solution deployment lifecycle stages encompassed therewithin effectively efficiently expediently excellently exceptionally extraordinary remarkably superbly tremendously wonderfully magnificently fantastically fabulously great!

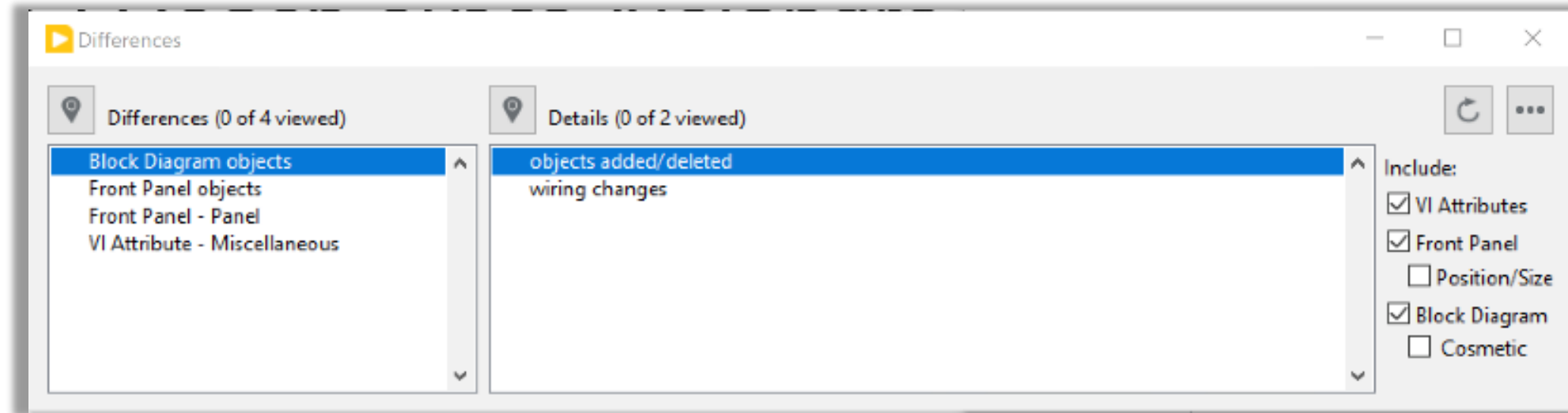
67s



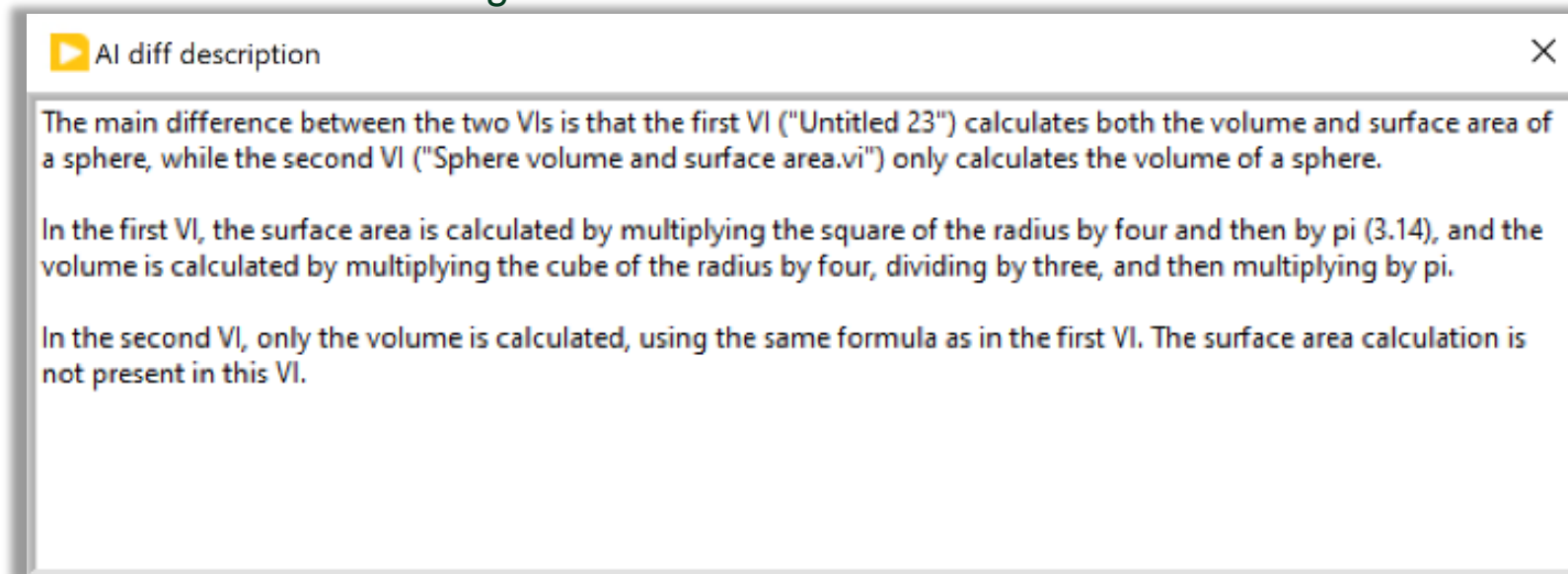
Further Concept Explorations

Concept Exploration: Enhance VI Diff

LabVIEW Diff describes modified objects



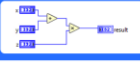
AI Enhanced Diff can describe functional changes



This feature is in active exploration and no warranties are made on if or when it will be made available

Concept Exploration: Writing VI Unit Tests

Uploaded Arithmetic Operation.vi



Nigel - 246
You uploaded **Arithmetic Operation.vi** performs $(x+y)*z$. No drivers or hardware used.

Can you help me write unit tests for Arithmetic Operation.vi?

Using LabVIEW

Test Case	VI	In/Out	Set/Check	Control Name	Data Type	Value	Comparison	Comparison VI Path	Vector Name
1	Arithmetic Operation.vi	IN	SET	x	T30	0			
1	Arithmetic Operation.vi	IN	SET	y	T30	0			
1	Arithmetic Operation.vi	IN	SET	z	T30	0			
1	Arithmetic Operation.vi	OUT	CHECK	result	T30	0	=		
2	Arithmetic Operation.vi	IN	SET	x	T30	1			
2	Arithmetic Operation.vi	IN	SET	y	T30	1			
2	Arithmetic Operation.vi	IN	SET	z	T30	1			
2	Arithmetic Operation.vi	OUT	CHECK	result	T30	2	=		
3	Arithmetic Operation.vi	IN	SET	x	T30	2			
3	Arithmetic Operation.vi	IN	SET	y	T30	3			
3	Arithmetic Operation.vi	IN	SET	z	T30	4			
3	Arithmetic Operation.vi	OUT	CHECK	result	T30	20	=		

Total number of test cases: 3

Run unit tests

Use case 2: Given a VI description, generate unit tests (i.e. test driven dev)

A VI called "Arithmetic Operation.vi" adds two numbers called x and y and multiplies the result with another number called z and produces output called result. All the inputs and outputs are of type I32. Can you write some unit tests for this VI?

Using LabVIEW

Test Case	VI	In/Out	Set/Check	Control Name	Data Type	Value	Comparison	Comparison VI Path	Vector Name
1	Arithmetic Operation.vi	IN	SET	x	I32	2			
1	Arithmetic Operation.vi	IN	SET	y	I32	3			
1	Arithmetic Operation.vi	IN	SET	z	I32	4			
1	Arithmetic Operation.vi	OUT	CHECK	result	I32	20	=		
2	Arithmetic Operation.vi	IN	SET	x	I32	0			
2	Arithmetic Operation.vi	IN	SET	y	I32	5			
2	Arithmetic Operation.vi	IN	SET	z	I32	3			
2	Arithmetic Operation.vi	OUT	CHECK	result	I32	15	=		
3	Arithmetic Operation.vi	IN	SET	x	I32	-2			
3	Arithmetic Operation.vi	IN	SET	y	I32	3			
3	Arithmetic Operation.vi	IN	SET	z	I32	2			
3	Arithmetic Operation.vi	OUT	CHECK	result	I32	2	=		

Total number of test cases: 3

Run unit tests

Running unit tests

Unit test file created at: C:\Users\aragwal\AppData\Local\Temp\LabVIEW\Arithmetic Operation\Arithmetic Operation.vi

Unit test report created at: C:\Users\aragwal\AppData\Local\Temp\LabVIEW\report.txt

Unit Test Report

LabVIEW Unit Test Framework
Time: 6/28/2023 2:52:45 PM

Test Summary

0 Test Error(s)
0 Test(s) Failed
0 Test(s) Skipped
1 Test(s) Passed
Time elapsed: 00:00:00

1. Arithmetic Operation.vi

VI under Test: C:\Users\aragwal\AppData\Local\Temp\LabVIEW\Arithmetic Operation.vi
VI Timestamp: 6/28/2023 2:52:37 PM
VI Revision: 63
Test Started: 6/28/2023 2:52:45 PM
Test Duration: 00:00:00
Code Coverage: 100.0%
Setup VI:
Teardown VI:
Result: Passed

Test Case 1:

Case Result: Passed

Repetition	Permutation	VI	In/Out	Control	Comparison	Used Vector Value	Value	Expected Value
1	1	VI under Test	In	x			2	
1	1	VI under Test	In	y			3	
1	1	VI under Test	In	z			4	
1	1	VI under Test	Out	result	=		20	20

Test Case 2:

Case Result: Passed

Repetition	Permutation	VI	In/Out	Control	Comparison	Used Vector Value	Value	Expected
------------	-------------	----	--------	---------	------------	-------------------	-------	----------

Use case 1: Given a VI, generate unit tests

Use case 3: Run unit tests and generate report

Use case 4: Identify bugs in the VI based on failing unit tests

This is my new VI. Can you tell me why my test cases are failing?

Nigel - 359
You uploaded **Arithmetic Operation.vi** performs $(x-y)*z$. Please provide the test cases you used, so I can help you identify the issue.

Uploaded Unit Test Report File report.txt

Nigel - 100 - 1300
The test cases are failing because the expected values are incorrect for the given inputs. The VI performs $(x-y)*z$. Update the test cases with the correct expected values:

Test Case 2:

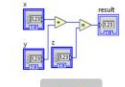
Inputs: x=1, y=1, z=1
Expected output: $(1-1)*1 = 0$

Test Case 3:

Inputs: x=2, y=3, z=4
Expected output: $(2-3)*4 = -4$

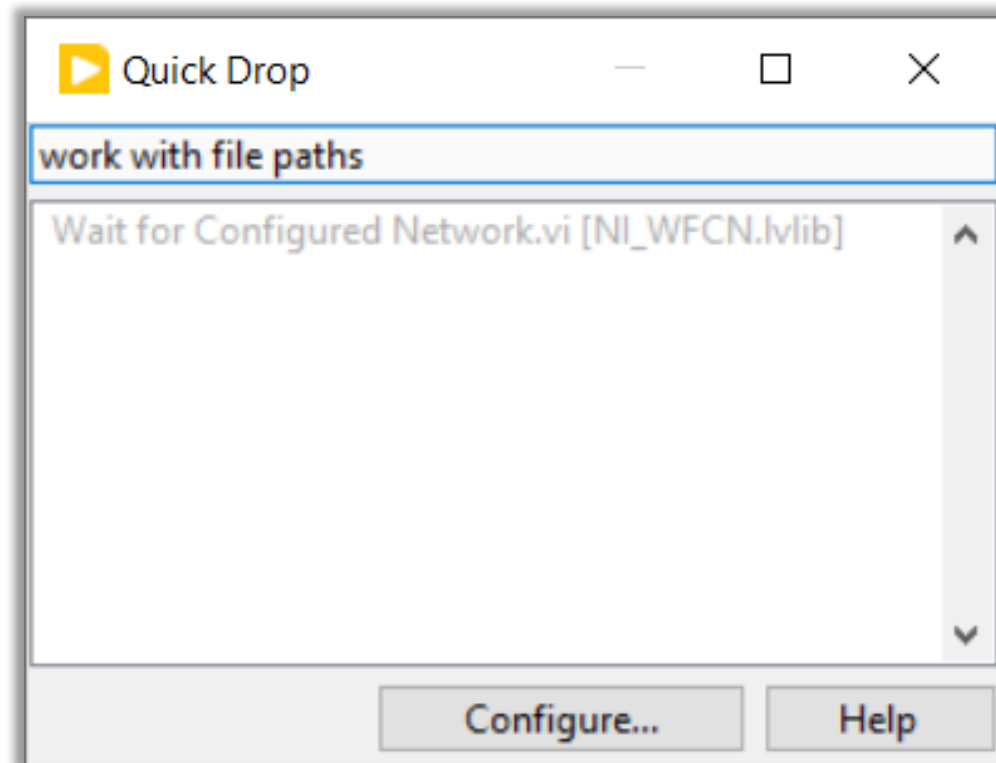
Update the expected values in your test cases and rerun the tests.

Nigel - 140 - 2040
If you want the VI to match the original test cases, you need to change the VI to perform $(x+y)*z$ instead of $(x-y)*z$. Modify the VI as follows:

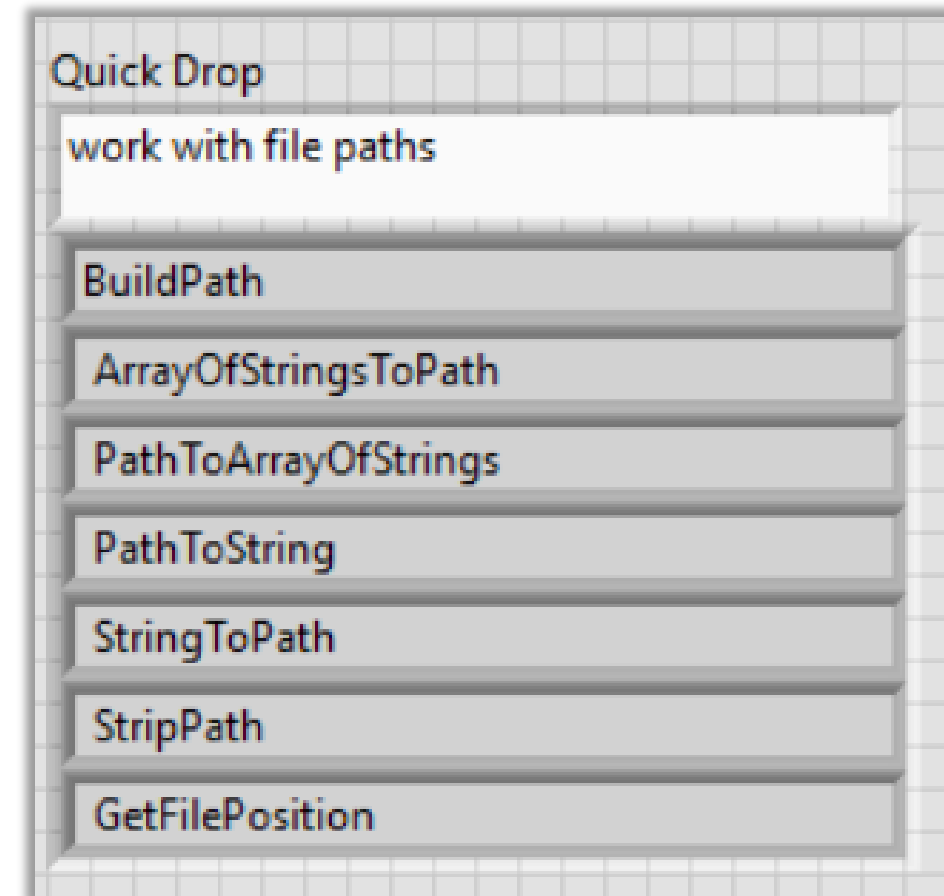


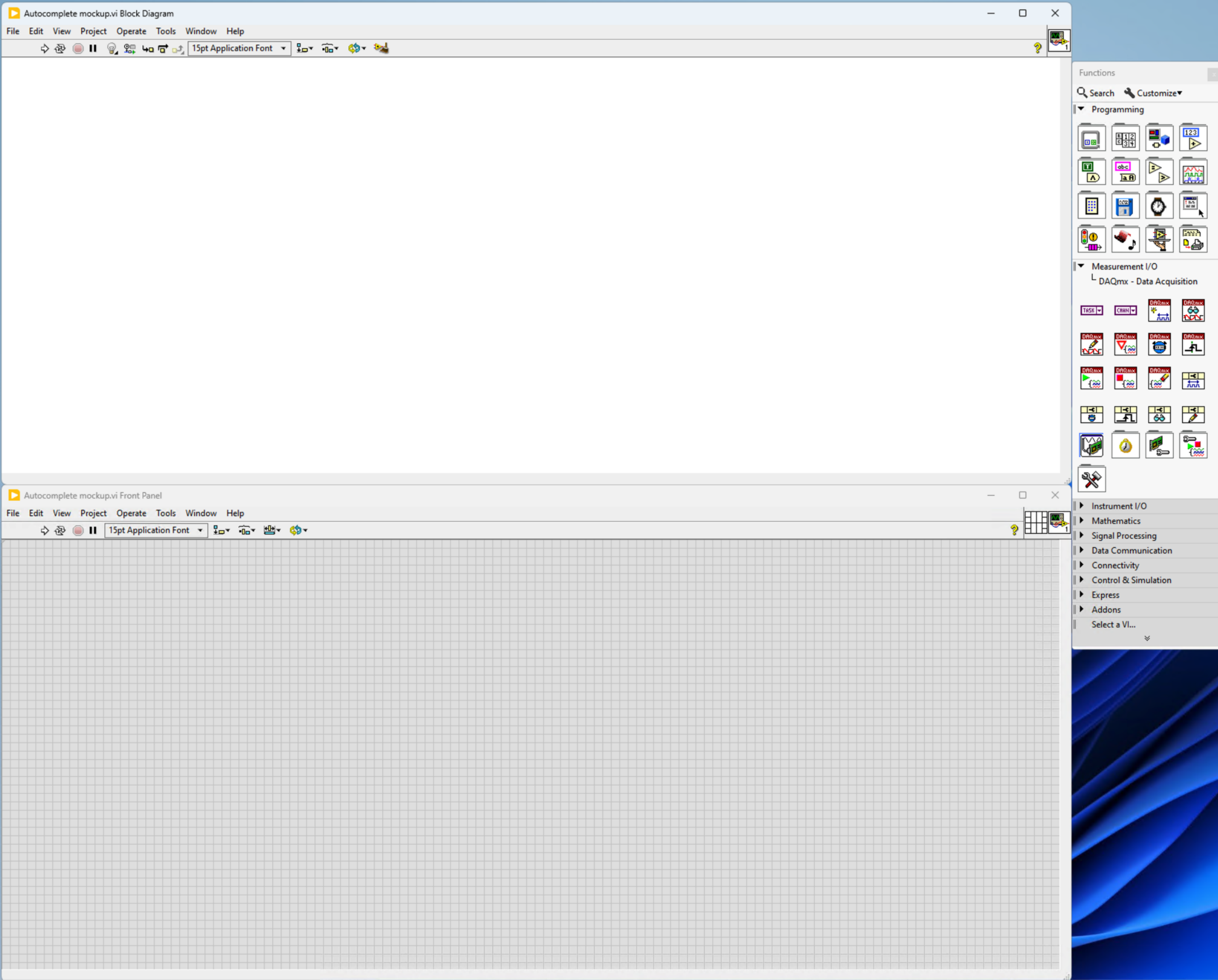
With this change, the VI should pass the original test cases.

Concept Exploration: Enhance Quick Drop



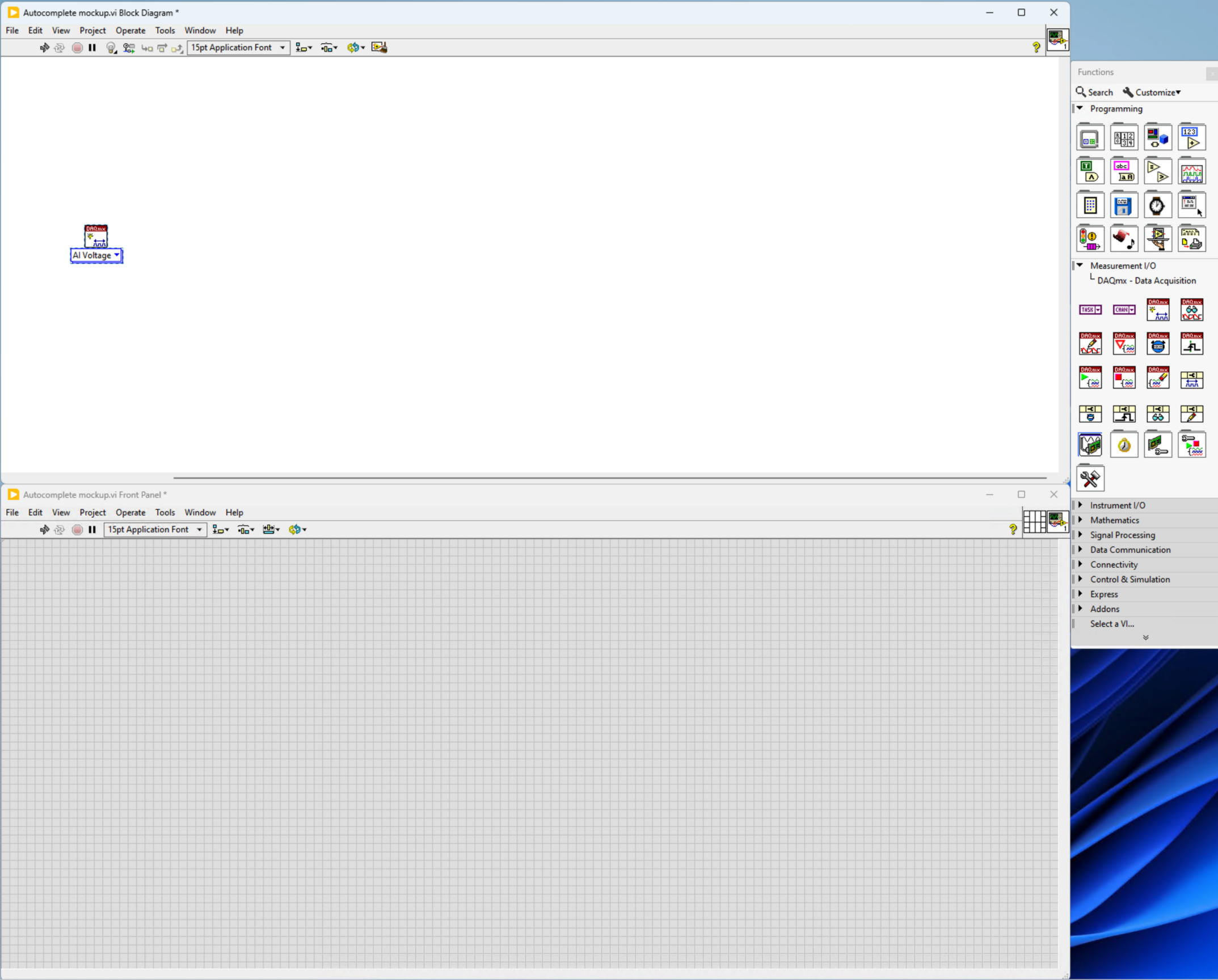
VS





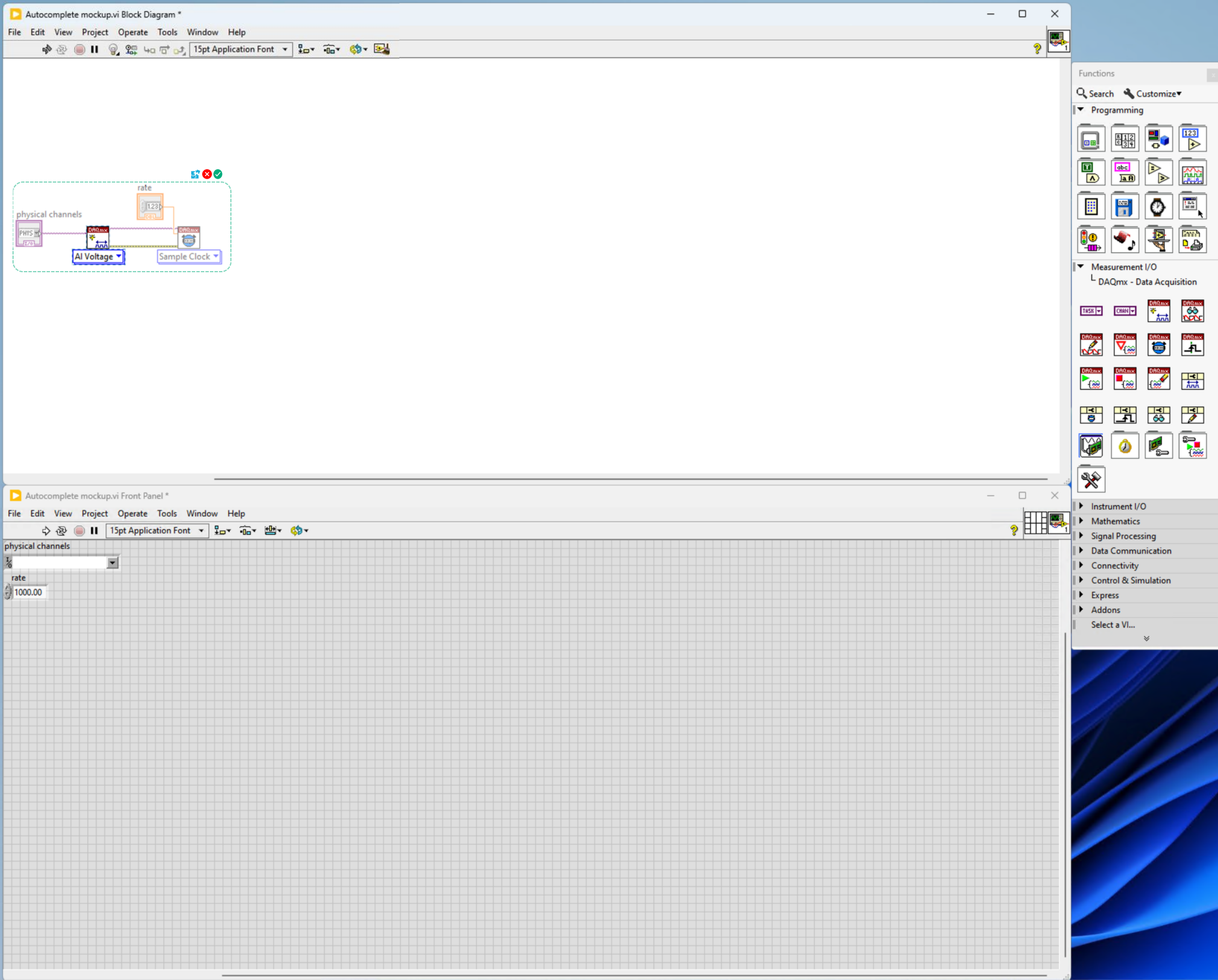
Concept Exploration: Auto-complete

New VI



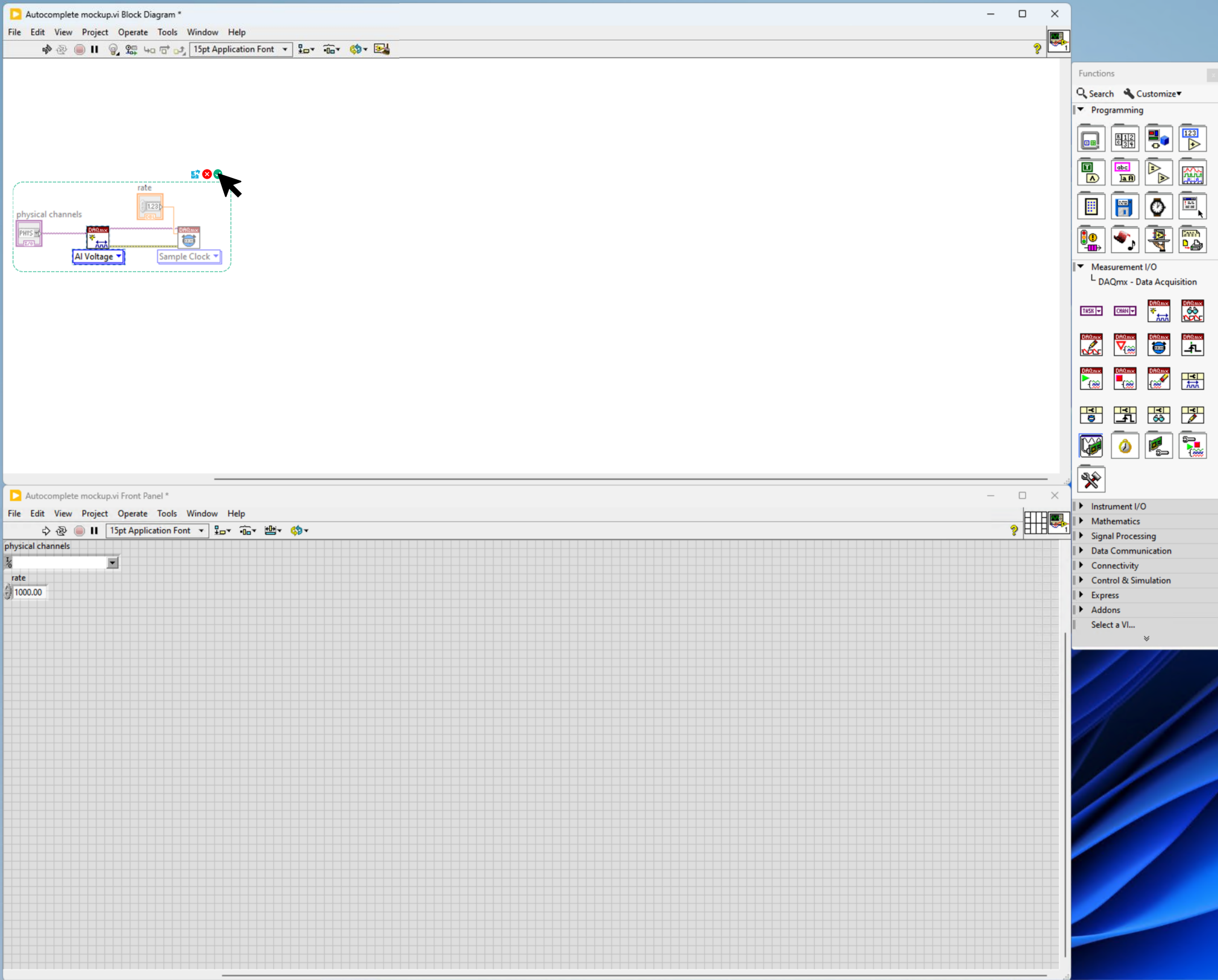
Concept Exploration: Auto-complete

Drop Create Channel



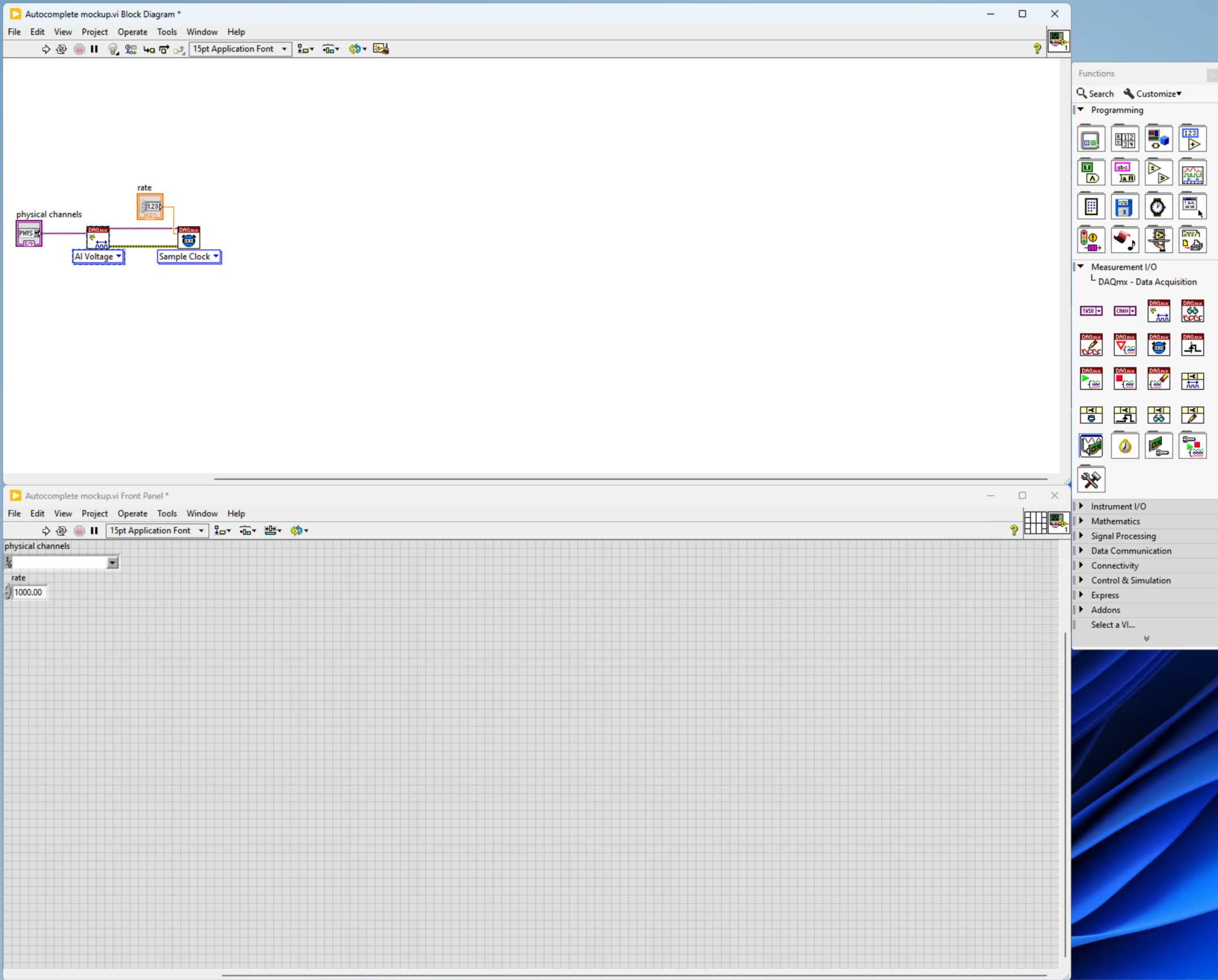
Concept Exploration: Auto-complete

Auto-complete shows up with everything needed to make the VI runnable



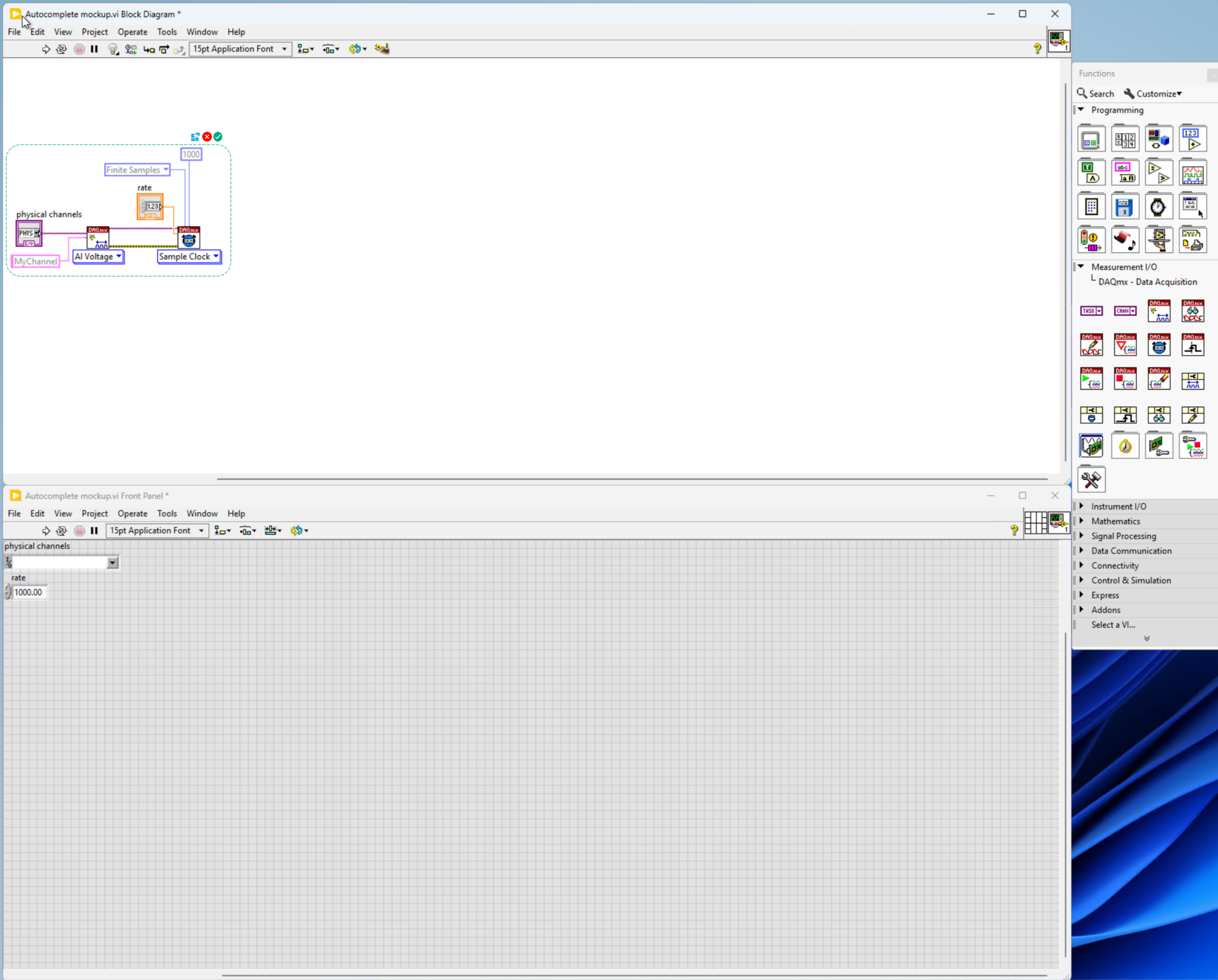
Concept Exploration: Auto-complete

User accepts suggestions



Concept Exploration: Auto-complete

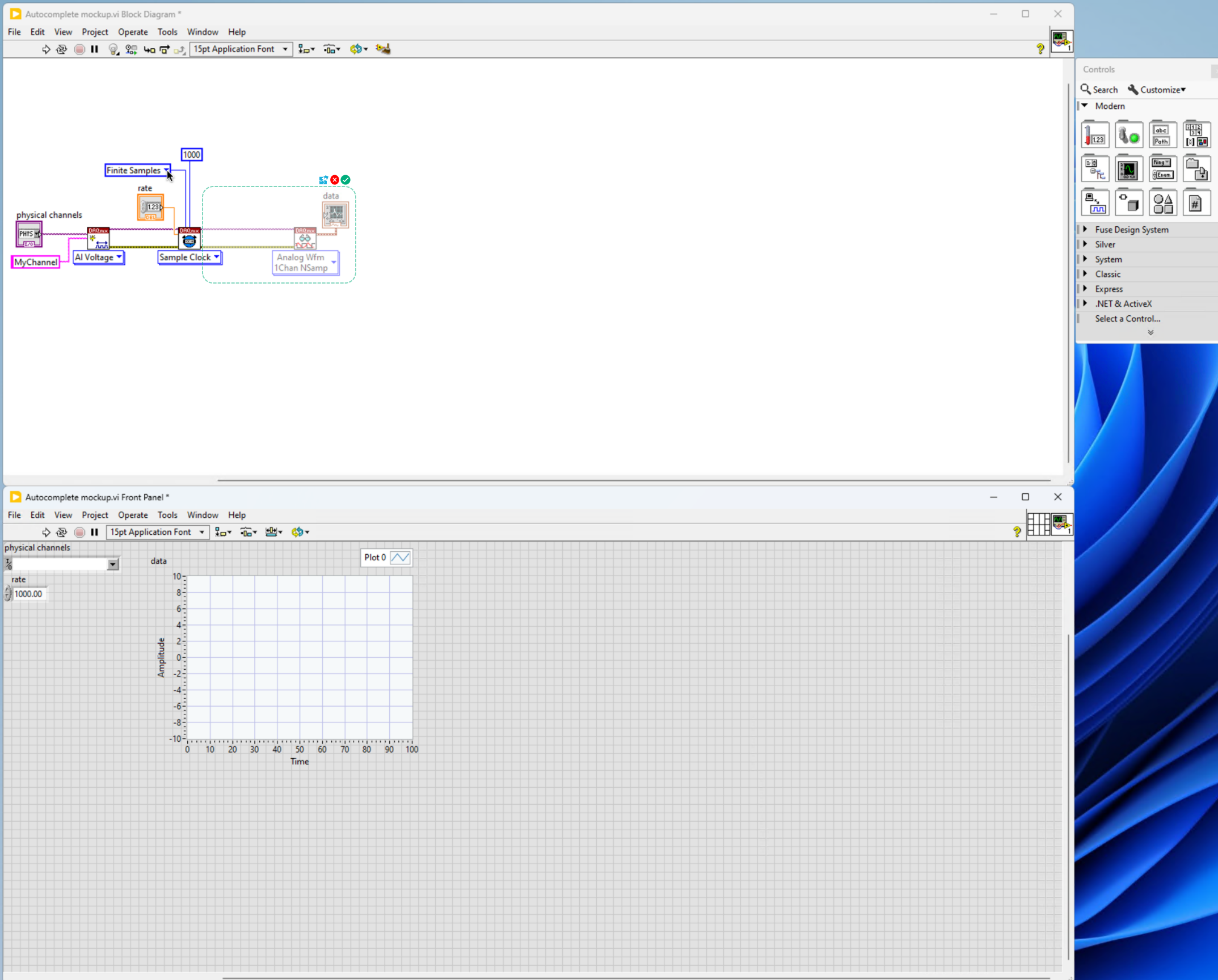
VI is now runnable



Concept Exploration:

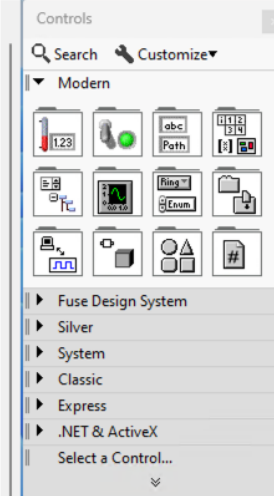
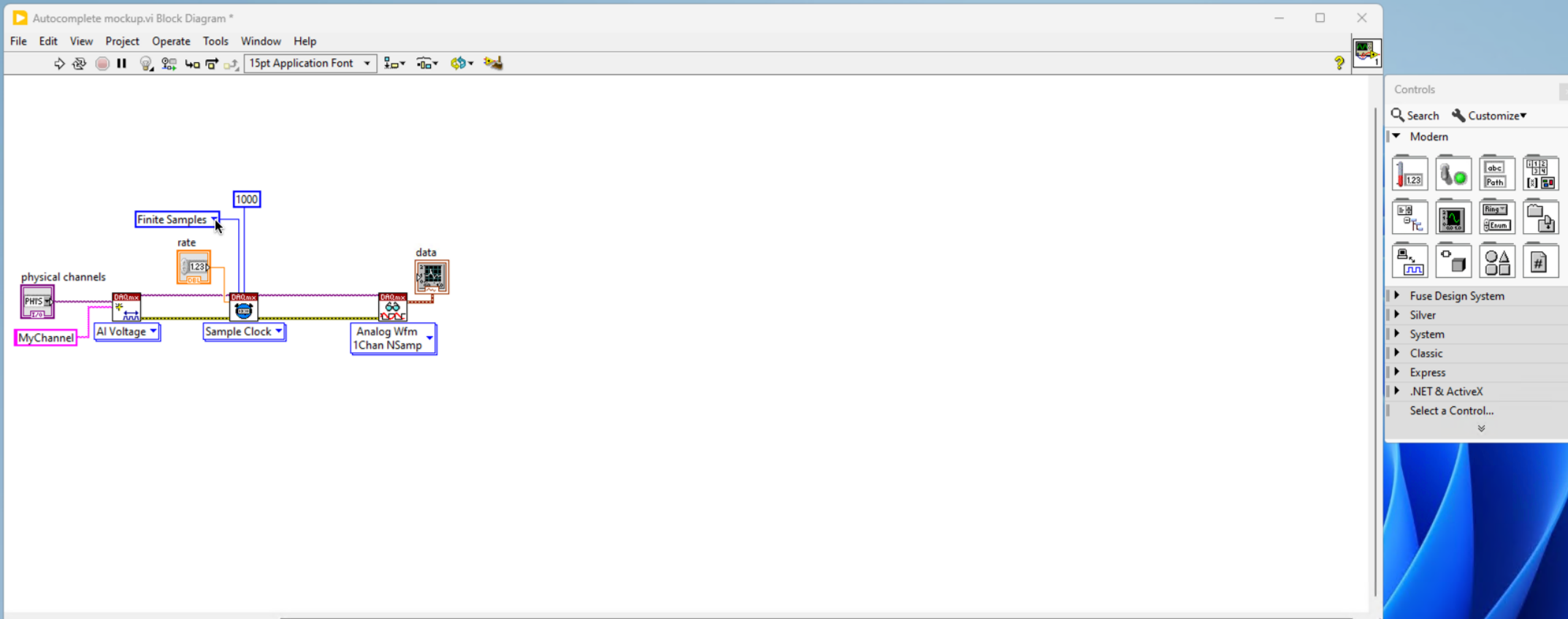
Auto-complete

Nigel suggests more constants



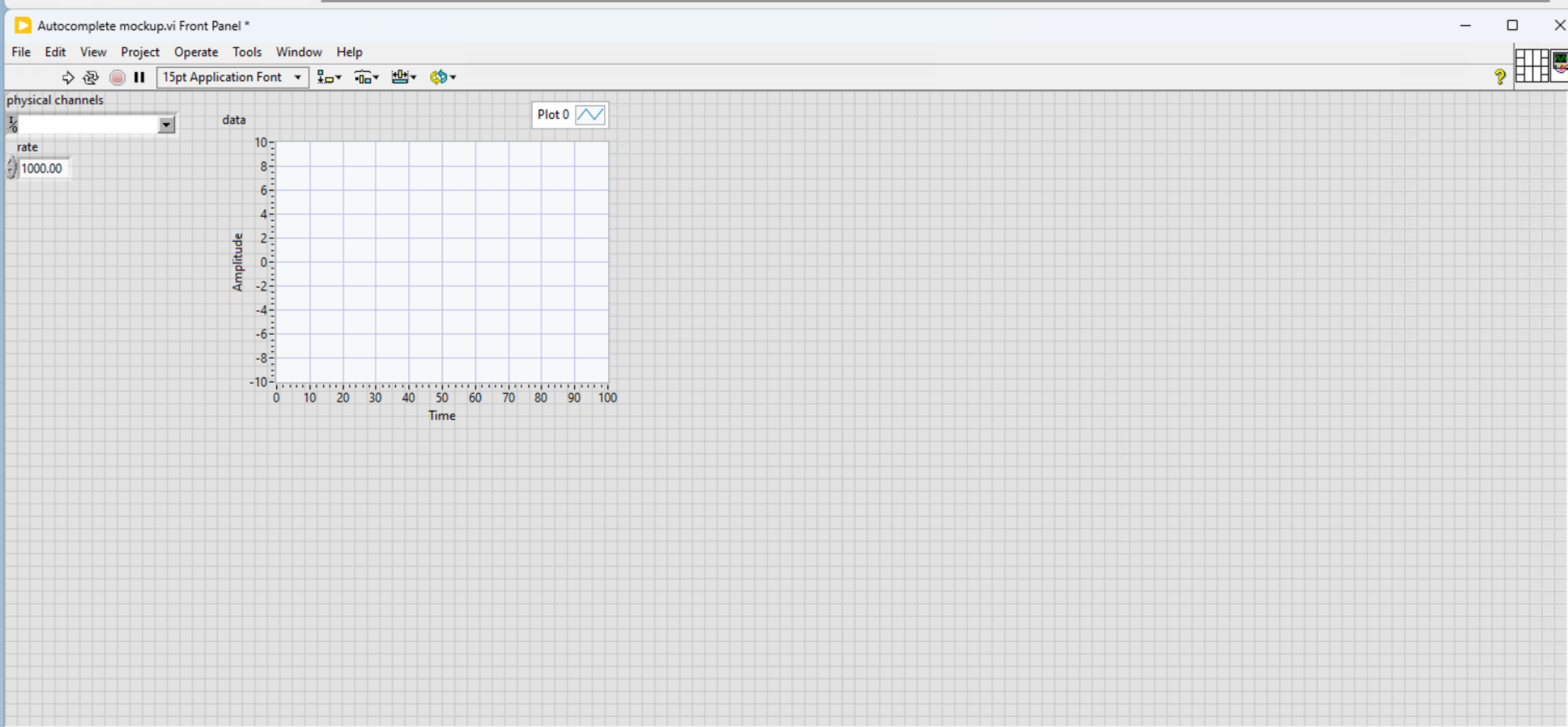
Concept Exploration: Auto-complete

Now that we know the total number of samples, Nigel suggests a Read VI



Concept Exploration: Auto-complete

Accepted



Autocomplete mockup.vi Block Diagram *

File Edit View Project Operate Tools Window Help

15pt Application Font

1000

Finite Samples
Continuous Samples
Hardware Timed Single Point

physical channels

MyChannel

AI Voltage

Sample Clock

Analog Wfm
1Chan NSamp

data

Functions

Search Customize

Programming

Measurement I/O

DAQmx - Data Acquisition

Instrument I/O

Mathematics

Signal Processing

Data Communication

Connectivity

Control & Simulation

Express

Addons

Select a VI...

Autocomplete mockup.vi Front Panel *

File Edit View Project Operate Tools Window Help

15pt Application Font

physical channels

rate

1000.00

data

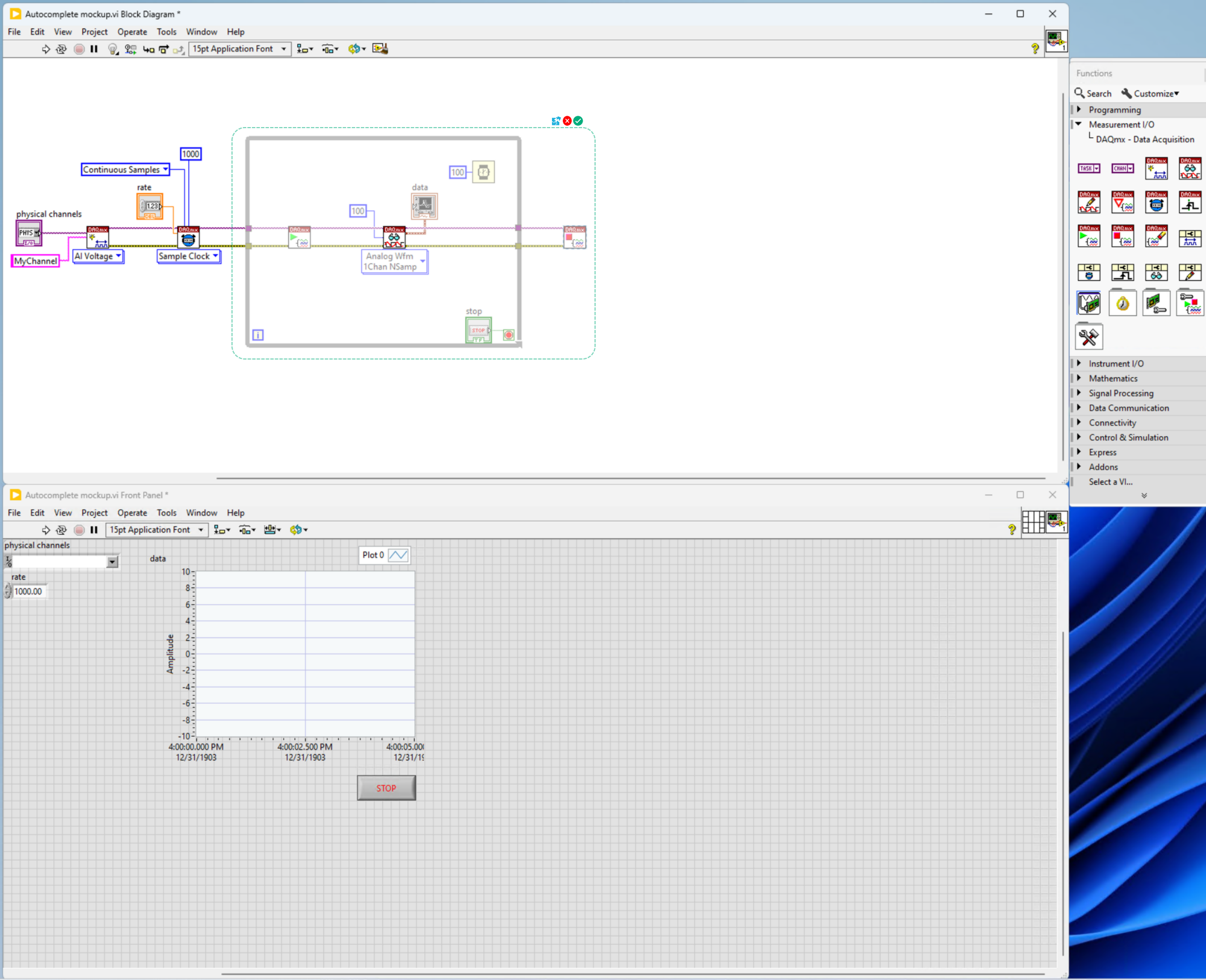
Plot 0

Amplitude

Time

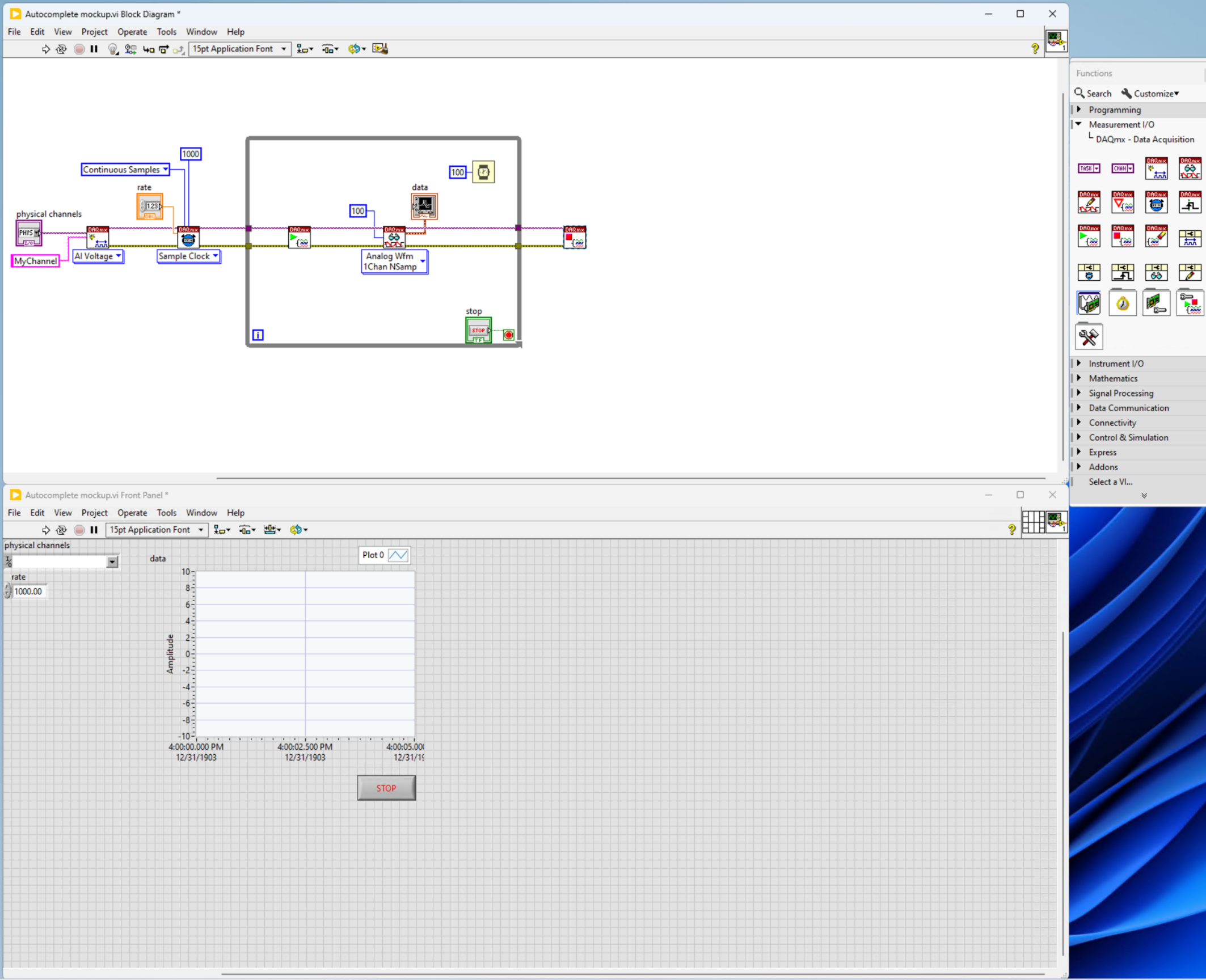
Concept Exploration: Auto-complete

User changes Finite to
Continuous



Concept Exploration: Auto-complete

Nigel suggests several optimizations for continuous run



Concept Exploration:
Auto-complete

Accepted

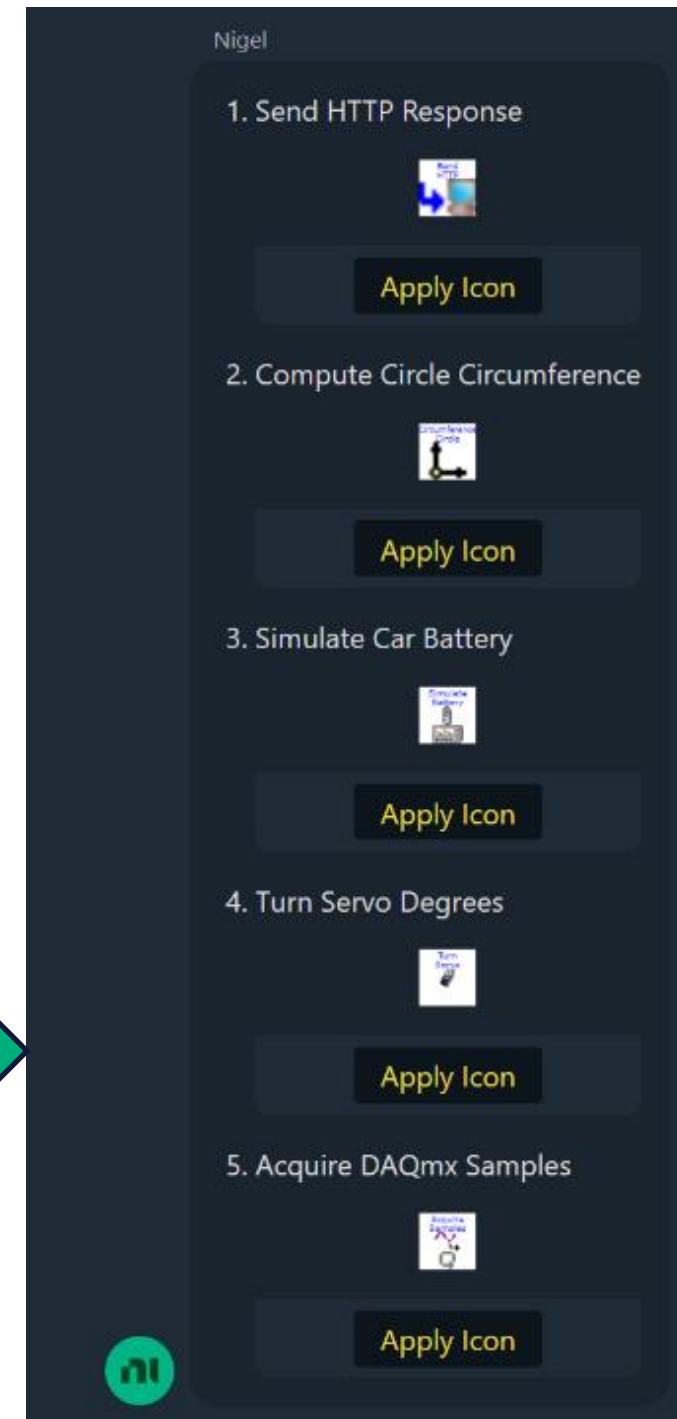
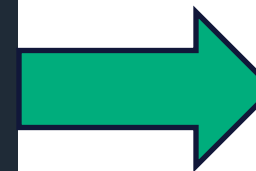
Concept Exploration: Generate VI Icons

Creates an icon based on the description of a VI

Adds text and uses the standard glyphs

Generate icons for VIs that:

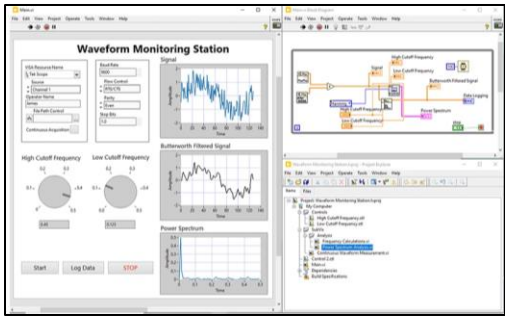
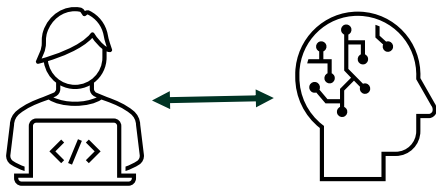
- send response back to HTTP server
- computing the circumference of a circle
- simulate car 12V battery using DCPower
- turning a servo some number of degrees
- acquire 100 analog voltage sample from DAQmx



This feature is in active exploration and no warranties are made on if or when it will be made available

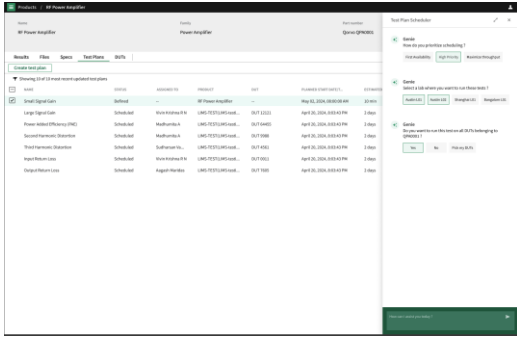
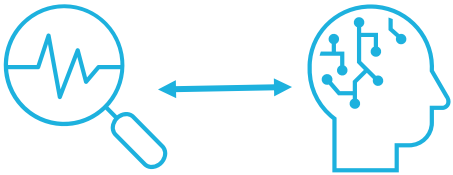
Test Co-Pilot Workflow Acceleration

Test Authoring



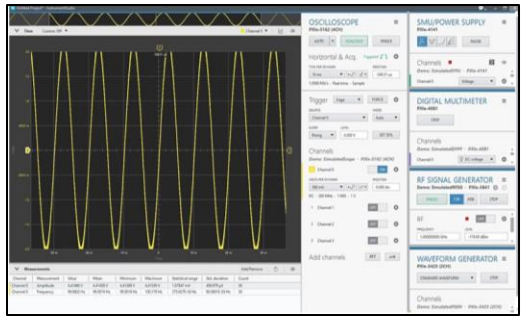
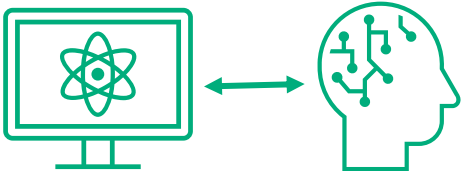
 LabVIEW

Data Analysis



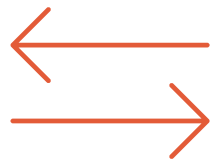
 SystemLink

Interactive Measurements



 InstrumentStudio

Translation Workflows



Migrations



Requirements
to Test Specs



Device
Control



“Hyper-personalization”

Secure and private ways for our AI-based tools to know about and leverage:

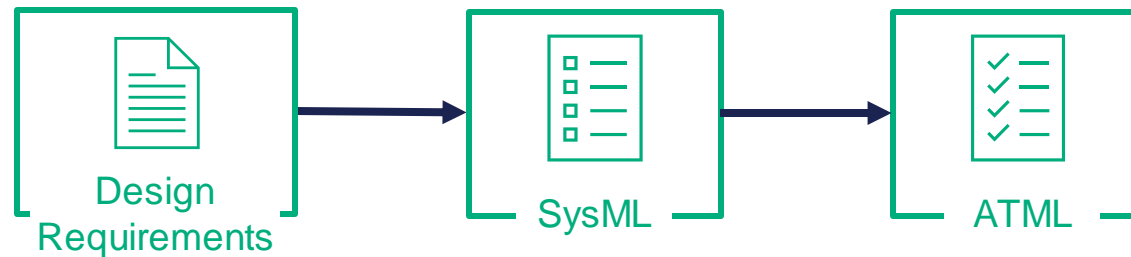
- Your internal reuse libraries
- Your coding guidelines
- Your hardware and fixtures
- Other company-specific knowledge

to provide **more complete and correct** assistance.

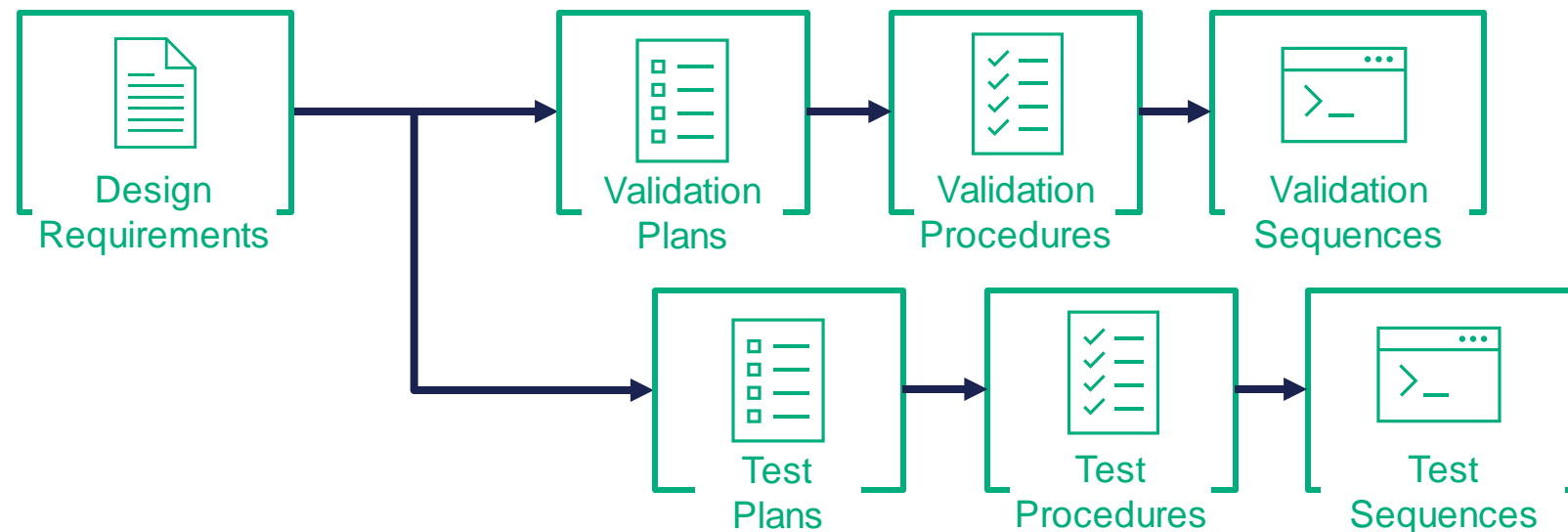
Structured Test Documentation AI-Pipelines



ATML Test Engineering



Cross-Industry Applicability



Example

```
validation_test_plan: PXIe-4139 ±60 V, 3 A Precision SMU
tests:
  voltage_programming_and_measurement:
    - range: 600 mV
      points: [0, 300 mV, -300 mV, 600 mV, -600 mV]
      temperature: 23C ± 5C
    - range: 6 V
      points: [0, 3 V, -3 V, 6 V, -6 V]
      temperature: 23C ± 5C
    - range: 60 V
      points: [0, 30 V, -30 V, 60 V, -60 V]
      temperature: 23C ± 5C
    - accuracy_verification:
        spec_tcal_±5C: refer to voltage accuracy table
        spec_tcal_±1C: refer to voltage accuracy table
    - tempco_verification:
        temperatures: [0C, 23C, 55C]
        spec: refer to voltage tempco table

  current_programming_and_measurement:
    - range: 1 µA
      points: [0, 500 nA, -500 nA, 1 uA, -1 uA]
      temperature: 23C ± 5C
    - range: 10 µA
      points: [0, 5 uA, -5 uA, 10 uA, -10 uA]
      temperature: 23C ± 5C
    - range: 100 µA
      points: [0, 50 uA, -50 uA, 100 uA, -100 uA]
      temperature: 23C ± 5C
```

What are your ideas?

LabVIEW Generative AI Early
Access Request



LabVIEW AI Assistant Tech Preview

Concept Exploration

