SEKANINA APS

CI/CD and AI for LabVIEW and TestStand

What actually works right now

Anders Sekanina

T&M days · Copenhagen · 3 November 2026

WHILE YOU SIT DOWN

Where does AI save you the most time in your daily work?

sekanina.dk/TM26

Meanwhile, in mainstream software

83%

use AI tools

~½

trust AI output when they can verify it easily

?

of this room — who is still standing?

Source: Stack Overflow Developer Survey 2026, 30,000+ respondents

Sekanina ApS

HIRING
Talk to me today



Anders Sekanina

Founder and consultant · Hørsholm

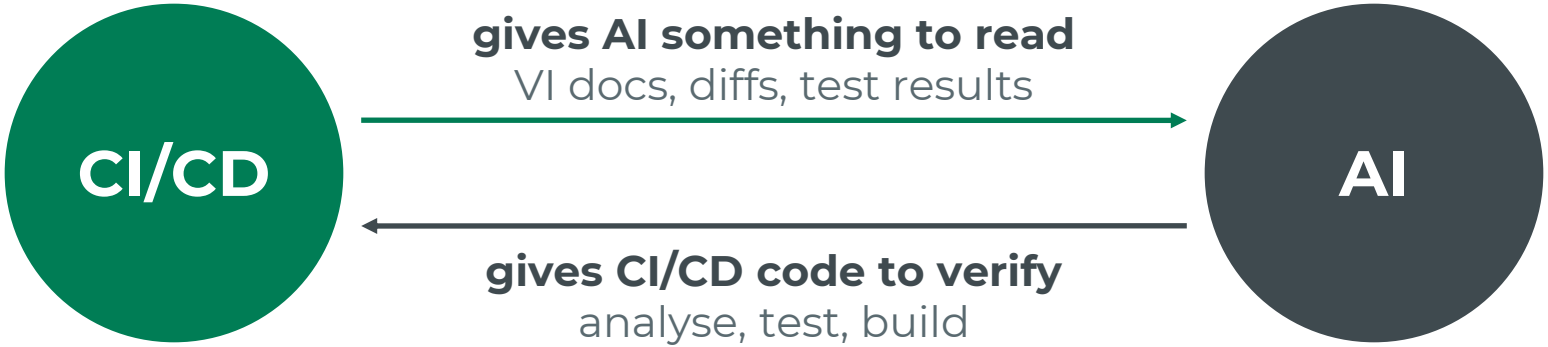
We help companies make better products — through automated test and production.

- LabVIEW and TestStand
- FPGA and Real-Time
- Hardware-in-the-Loop test
- CI/CD for LabVIEW
- Courses

Where does AI save you the most time?

The idea of this talk: a loop

CI/CD: every push to Git → a build machine analyses, tests and builds your code. No human involved.



What works today. It may look different by the time you try it.

01

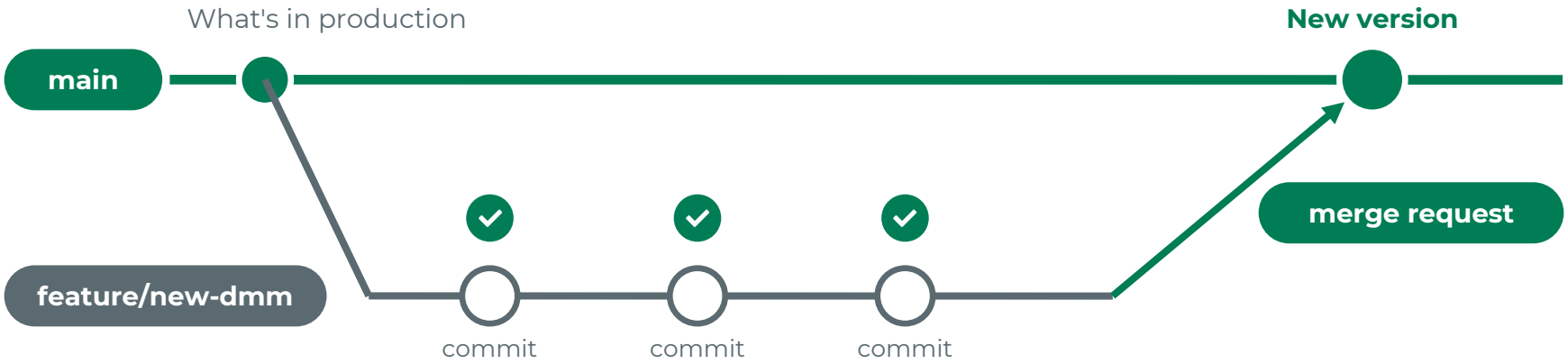
CI/CD

What runs when — six steps, one by one

Hands up: LabVIEW code in Git? · An automated build?

Git: branches, commits and merge requests

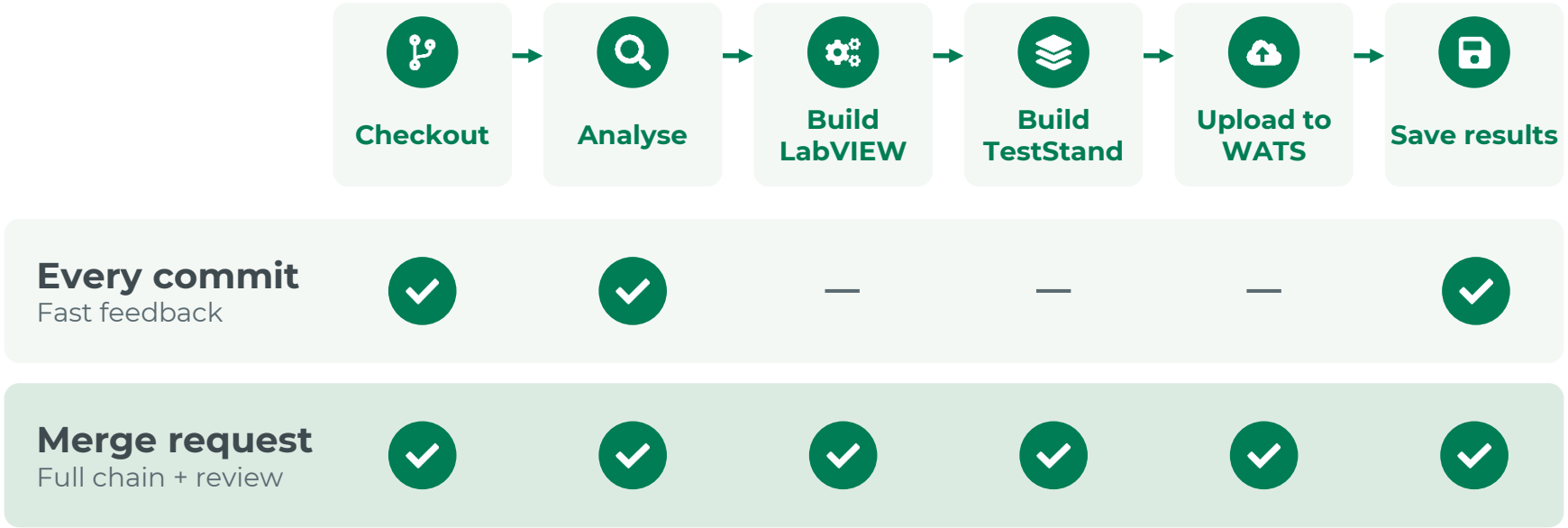
You change the code on a branch. A merge request brings the change into main.



Every commit
Quick checks — checkout, analyse, save results. Feedback in minutes.

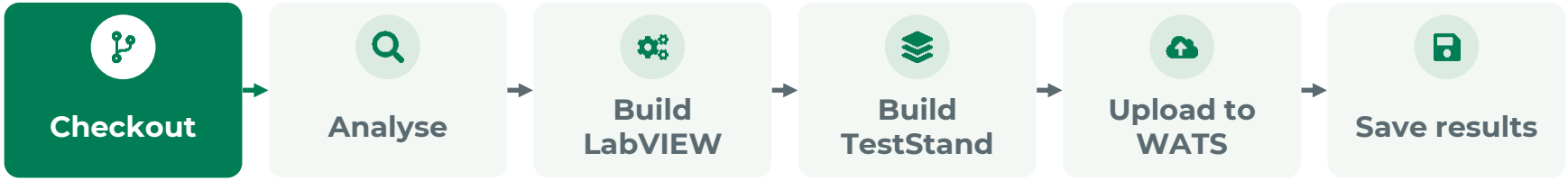
Merge request
A colleague reviews it, and the full chain runs:.

CI/CD: what runs when



Every commit gets checked. Every merge request ends in WATS.

1 · Checkout



Git checkout on a VM
Self-hosted GitLab runner on Windows

LabVIEW and TestStand on the VM
No developer PC involved

Started by a commit
Nobody logs in to a build PC

LESSON LEARNED

Keep the path short

GitLab's build folders plus deep LabVIEW paths pass 260 characters. The errors say nothing about paths.
Fix: builds_dir = C:\b

2 · Analyse

THANK YOU
Elijah Kerry



Checkout → **Analyse** → Build LabVIEW → Build TestStand → Upload to WATS → Save results

Documentation + diff report
HTML VI docs, and what changed in the merge request

VI Analyzer
Checks every VI on every commit

Unit tests
Results on the dashboard

Could be next: SBOM
A list of every component in the build

FROM NI

LabVIEW CI/CD Toolkit

Everything on the left, with results on its CI Dashboard.

Scan for the toolkit
ni.github.io/LabVIEW-CI-CD-Toolkit

3 · Build LabVIEW

THANK YOU
Anton Sundqvist



Checkout → Analyse → **Build LabVIEW** → Build TestStand → Upload to WATS → Save results

LabVIEW → packed libraries
Built in dependency order

LabVIEW CLI operations
Built into LabVIEW — see right

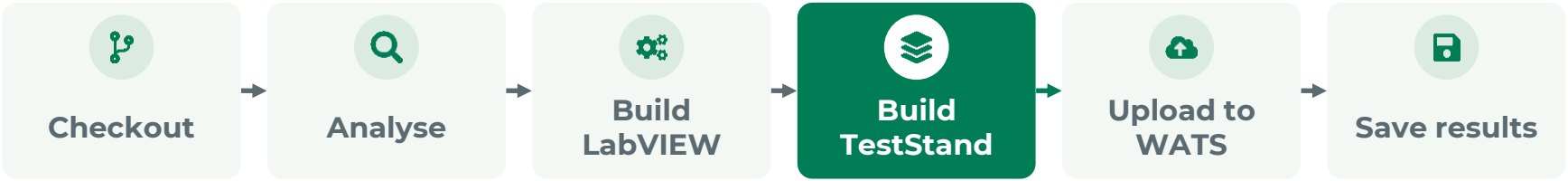
Plus custom operations
A LabVIEW class you drop in, called by name

DID YOU KNOW

LabVIEW CLI has built-in operations

`MassCompile` Recompile a folder of VIs	**`ExecuteBuildSpec`** Run a build specification
`RunVIAnalyzer` VI Analyzer, report to a file	**`RunUnitTests`** Unit tests, report for CI
`CreateComparisonReport` VI diff as an HTML report	**`RunVI`** Run your own VI

4 · Build TestStand



Update the sequence file
Relink .lvlib → .lvlibp, and paths to firmware and other files

Today: TestStand engine API from LabVIEW
A LabVIEW step opens the sequence and updates the paths

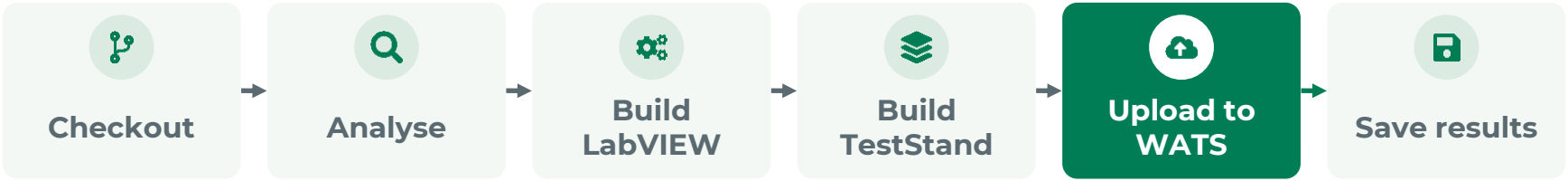
Maybe easier: the XML sequence file
Save the sequence as XML — then it's a plain text replace

LESSON LEARNED

Native DLLs not found

Works in development, fails under the TestStand RTE: OpenCV DLLs not found.
Fix: copy the helper DLLs into the deployment

5 · Upload to WATS



1. **Draft package in WATS**
   WATS gives the package number
2. **Version into the .seq file**
   Merge request + package number
3. **Tags in WATS**
   Installs on the right machines — branch builds are special
4. **Upload the package**
   The test stations take it from WATS

THE VERSION

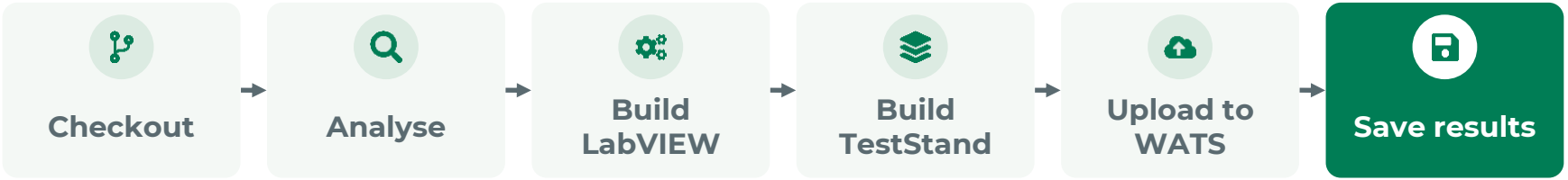
`3.2.418.1207`

418 merge request !418
1207 WATS package number

Every unit traces back to a merge request.

Example numbers

6 · Save results



Results saved
Kept with the pipeline run

Relevant emails sent
To the people who need to know

Traceable
Back to the merge request via the version in step 5

WHAT IS KEPT

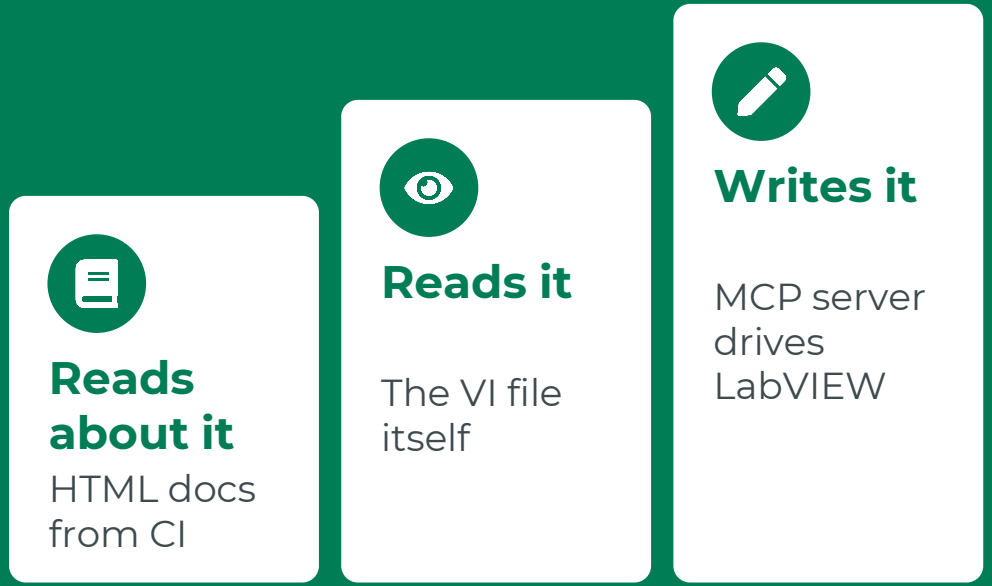
- HTML VI docs
- Diff reports
- VI Analyzer report
- Unit test results
- Build logs

Start on Monday: Git → LabVIEW CLI → VI Analyzer
All three run on your own PC — no build server needed.

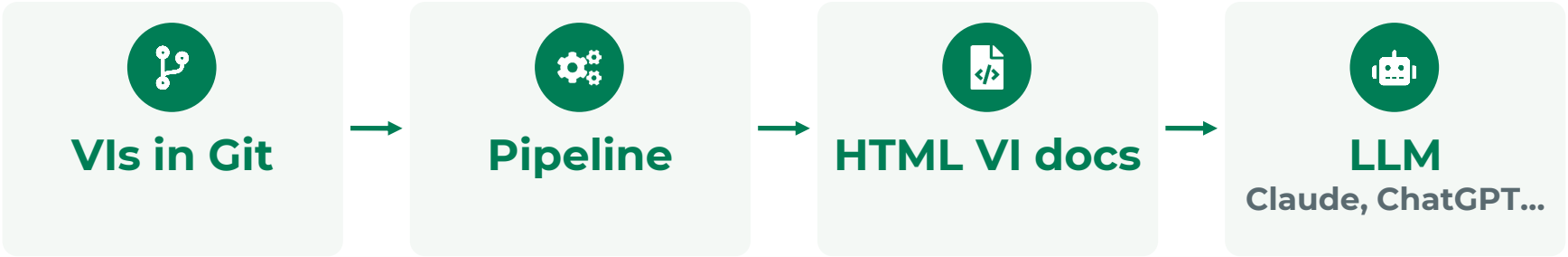
02

AI

LabVIEW first, then TestStand



My original plan: docs from CI, read by an LLM



Then you can discuss your own code

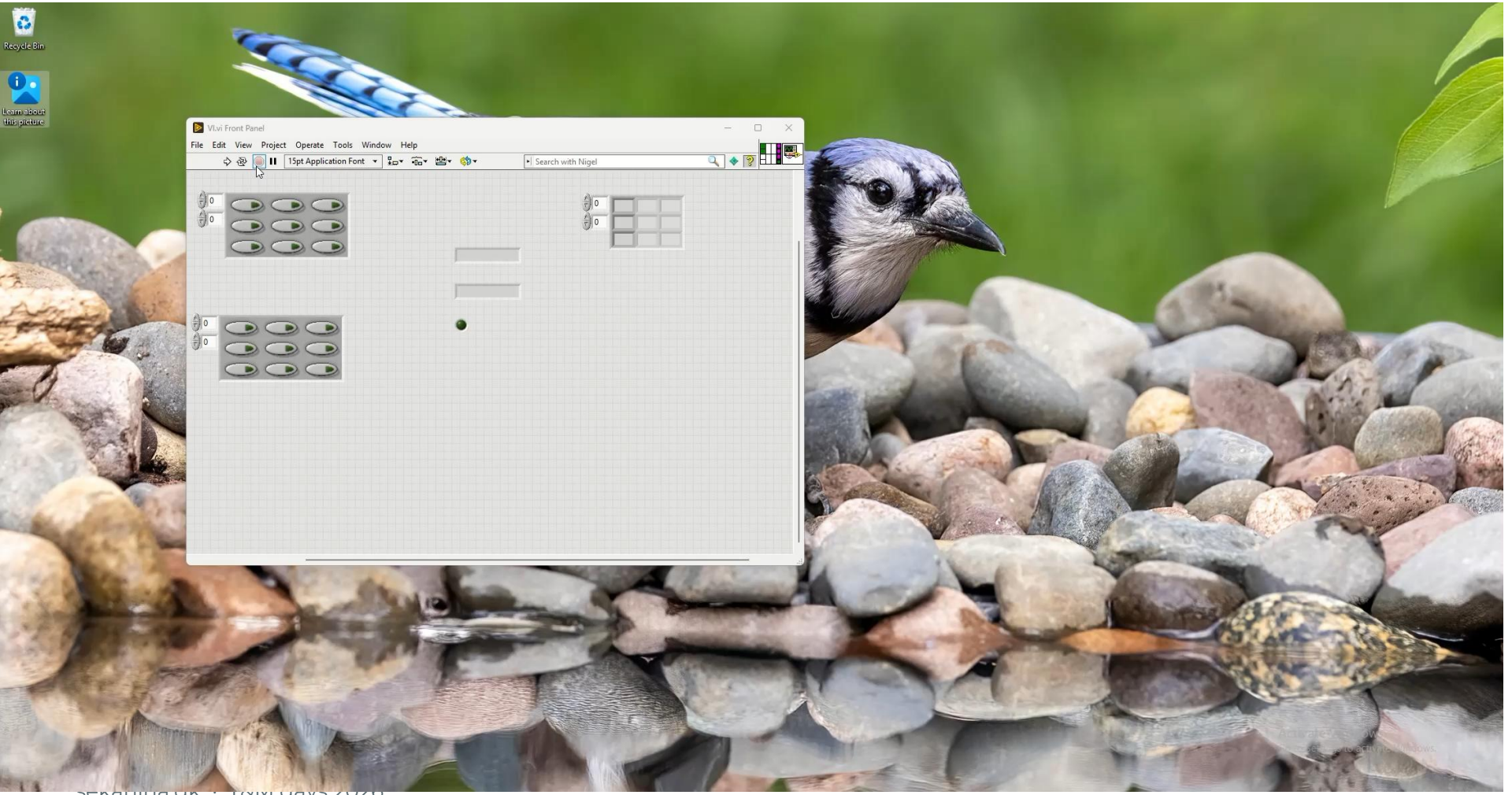
“I need to support a new DMM — which VIs are affected?”

“What does this merge request change — and what calls it?”



Tic-tac-toe, made by Nigel

No documentation. No comments.


Recycle Bin
Learn about this picture
VI.vi Front Panel
File Edit View Project Operate Tools Window Help
15pt Application Font
Search with Nigel

Can AI read that code?

00:00

Back at it, Anders

VI.vi

VI

For this session forget everything you know from earlier.
Read this VI and extract all information you can get, both what the function are and all information about the VI you can get.
Do not use local installed tools, only what is default included in Claude
Draw also the front panel and block diagram

Project or folder

Opus 5.5 High Manual

DID YOU KNOW #1 | TWO WEEKS AGO

THANK YOU
Jonathan Graesser



What does Claude use to read my VIs?

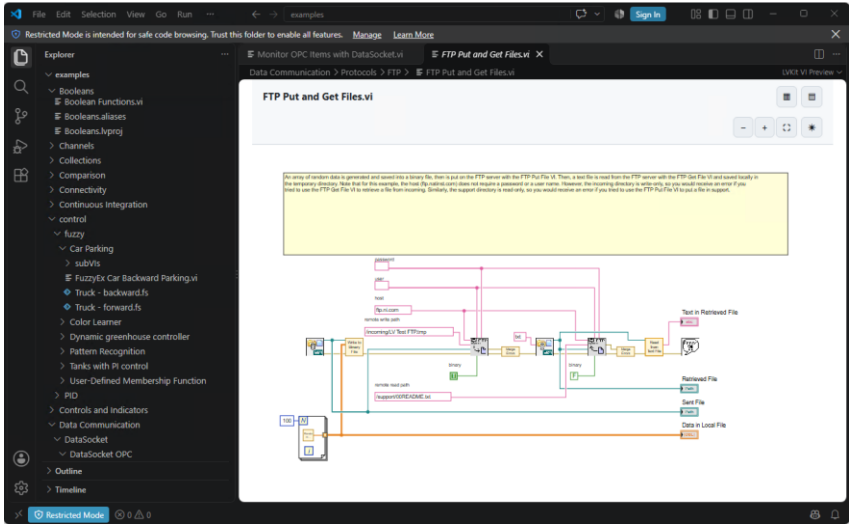
pylabview

Open-source Python parser for .vi files.
Claude cloned it and ran readRSRC → XML.

LVkit built on pylabview

- ✔ Your VIs in VS Code
- ✔ Describe, diff, document
- ✔ MCP server for AI agents

pragmatest.com/lvkit



LVkit showing a VI in VS Code

DID YOU KNOW #2

NOW

The way Claude reads VIs keeps changing

TWO WEEKS AGO

Claude used pylabview

Commands it ran:

```
git clone github.com/mefistotelis/pylabview
```

```
python3 -m pylabview.readRSRC -x -i Main.vi
```

.vi → XML → Claude reads the XML

→

NOW

Its own RSRC parser

Command it ran:

```
python3 read_rsrc.py Main.vi
```

```
28 blocks: LVSR vers CONP STRG icl8
FPHb BDHb VCTP … → labels, comments
```

About 40 lines of Python, written on the spot

Same VI, different tool. Claude picks what works best right now.

Can an LLM modify my LabVIEW code?



THANK YOU
Jean-Claude Mengisen

Yes — when it can use LabVIEW itself.

LLM
Claude, ChatGPT…

→

MCP server
from Zühlke

→

LabVIEW
the real IDE

→

VI
a normal VI file

MCP = Model Context Protocol. An open standard that lets an AI use tools. This MCP server gives it LabVIEW — the same functions Nigel uses, and more.

github.com/Zuehlke/labview-mcp

Can an LLM modify my LabVIEW code?



THANK YOU
Jean-Claude Mengisen

Yes — when it can use LabVIEW itself

NOW

That was a week ago.

Now the LabVIEW MCP server uses pylabview: it reads and writes VIs directly — also older LabVIEW versions.

LL
Claude, Cl

VI
al VI file

MCP = M standard that lets an AI use tools. This MCP server gives it LabVIEW — the same functions Nigel uses, and more.

github.com/Zuehlke/labview-mcp

Can an LLM modify my TestStand code?

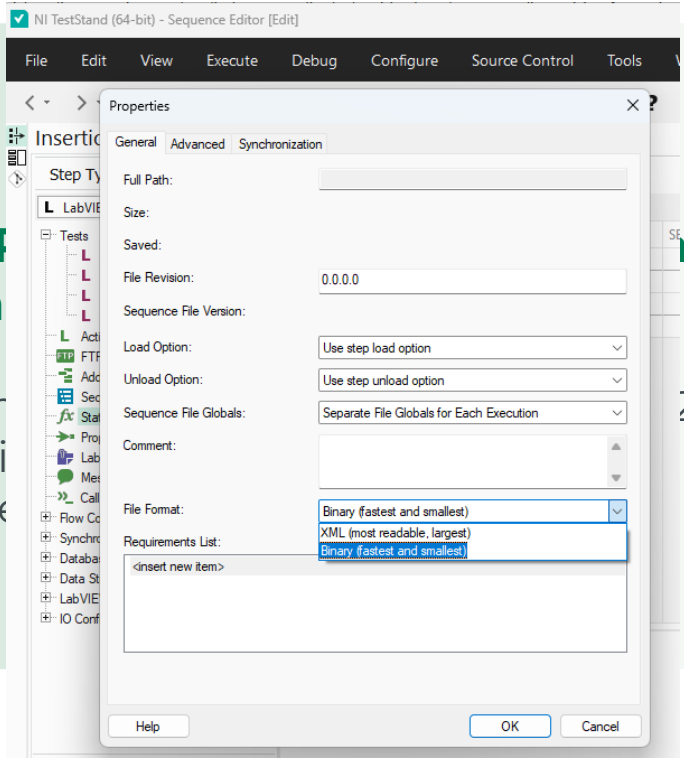
Three ways — from the file to the engine

Edit the file

Save sequences as XML. The LLM edits the text directly.

TestStand AP… from Python

The AI drives th… TestStand engi… TestStand write… file.

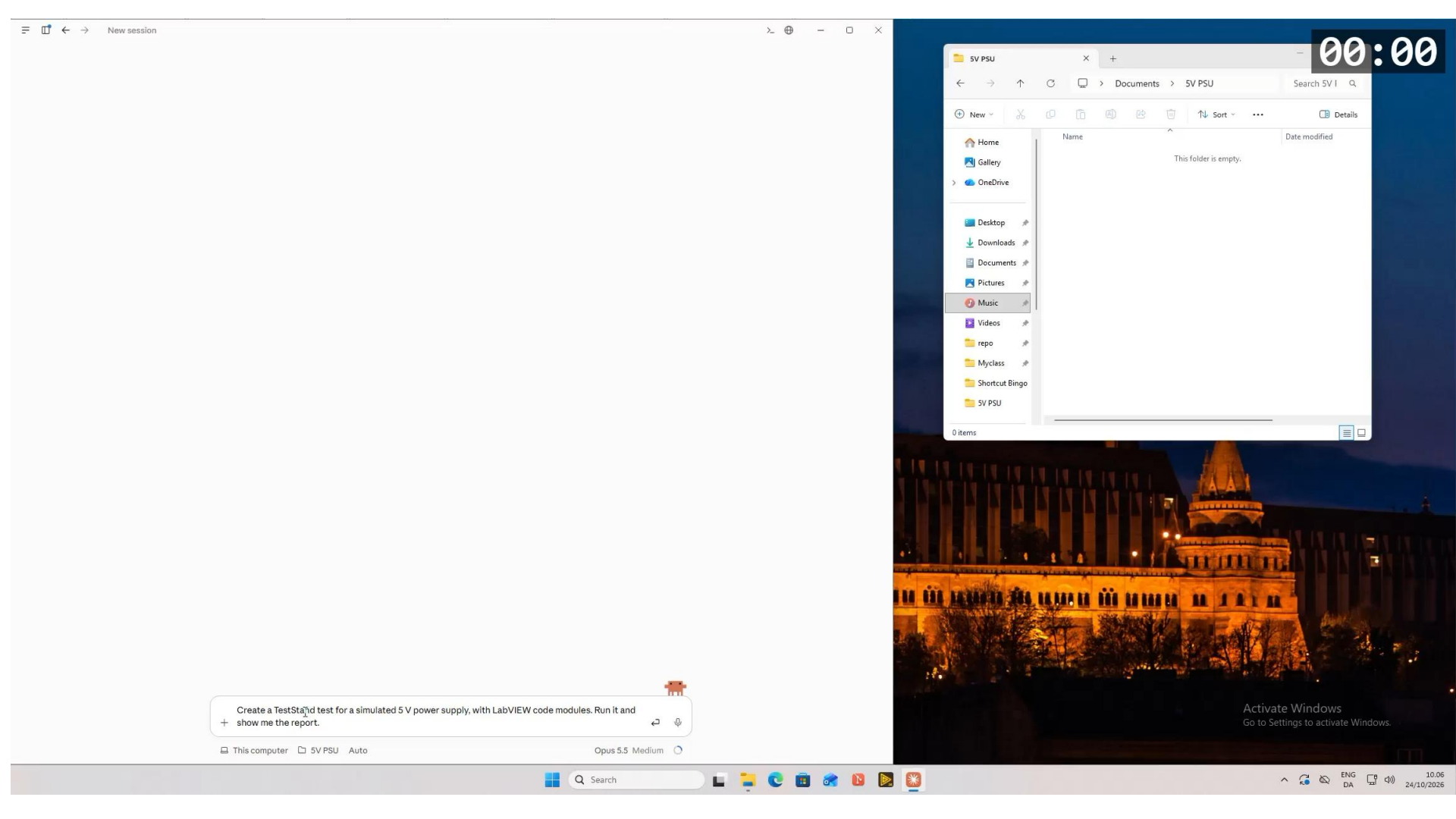


One prompt. A complete test.

What is possible with AI, LabVIEW and TestStand today

PROMPT

“Create a TestStand test for a simulated 5 V power supply, with LabVIEW code modules. Run it and show me the report.”


New session
00:00
5V PSU
Documents
5V PSU
Search 5V I
New
Sort
Details
Home
Gallery
OneDrive
Desktop
Downloads
Documents
Pictures
Music
Videos
repo
Myclass
Shortcut Bingo
5V PSU
Name
Date modified
This folder is empty.
0 items
Create a TestStand test for a simulated 5 V power supply, with LabVIEW code modules. Run it and show me the report.
This computer
5V PSU
Auto
Opus 5.5 Medium
Activate Windows
Go to Settings to activate Windows.
Search
ENG
DA
10.06
24/10/2026

LabVIEW and TestStand are close to working with AI like any text-based language.

A REAL CUSTOMER CHANGE

Planned in the car. Implemented by the time I arrived.
Reviewed, tested, in production.

AI on customer projects: I ask first

I'm always open about it. The customer decides what I'm allowed to do:

“It's test code, not our core business.”

I use the full toolbox.

“Use our own subscription.”

Their account. They control what is shared, and where.

“No AI on our code.”

No AI on their code. General questions only.

Which of the three is your company? Worth knowing before you paste code into a chatbot.

What works — status 3 November 2026

Works

- CI Dashboard: docs, diffs, VI Analyzer, unit tests
- Claude reads, writes and modifies LabVIEW and TestStand code
- One prompt → a complete, running TestStand test or LabVIEW application

Half-works

- Modifying large, complex Vis.
- Nigel writing a LabVIEW code.
- Style and cleanup could be improved.

Not yet

- LLM writing .seq files from scratch as text
- Nigel with event structures, subvi, etc

NORDLUG — the next dates

NEXT

NORDLUG #9

Tuesday 8 December 2026

15:00–17:30 · Online (Teams)

Agenda out 20 November

Scan to register →

luma.com/ahm1cwgj

SAVE THE DATE

NORDLUG #10

Tuesday 9 February 2027

15:00–18:30

In person in Oslo, Stockholm and Copenhagen — and online.

Ends with food and a chance to talk, thanks to NI.

There's never been a better time to be a LabVIEW developer than now.



We're hiring. Talk to me today — anders@sekanina.dk