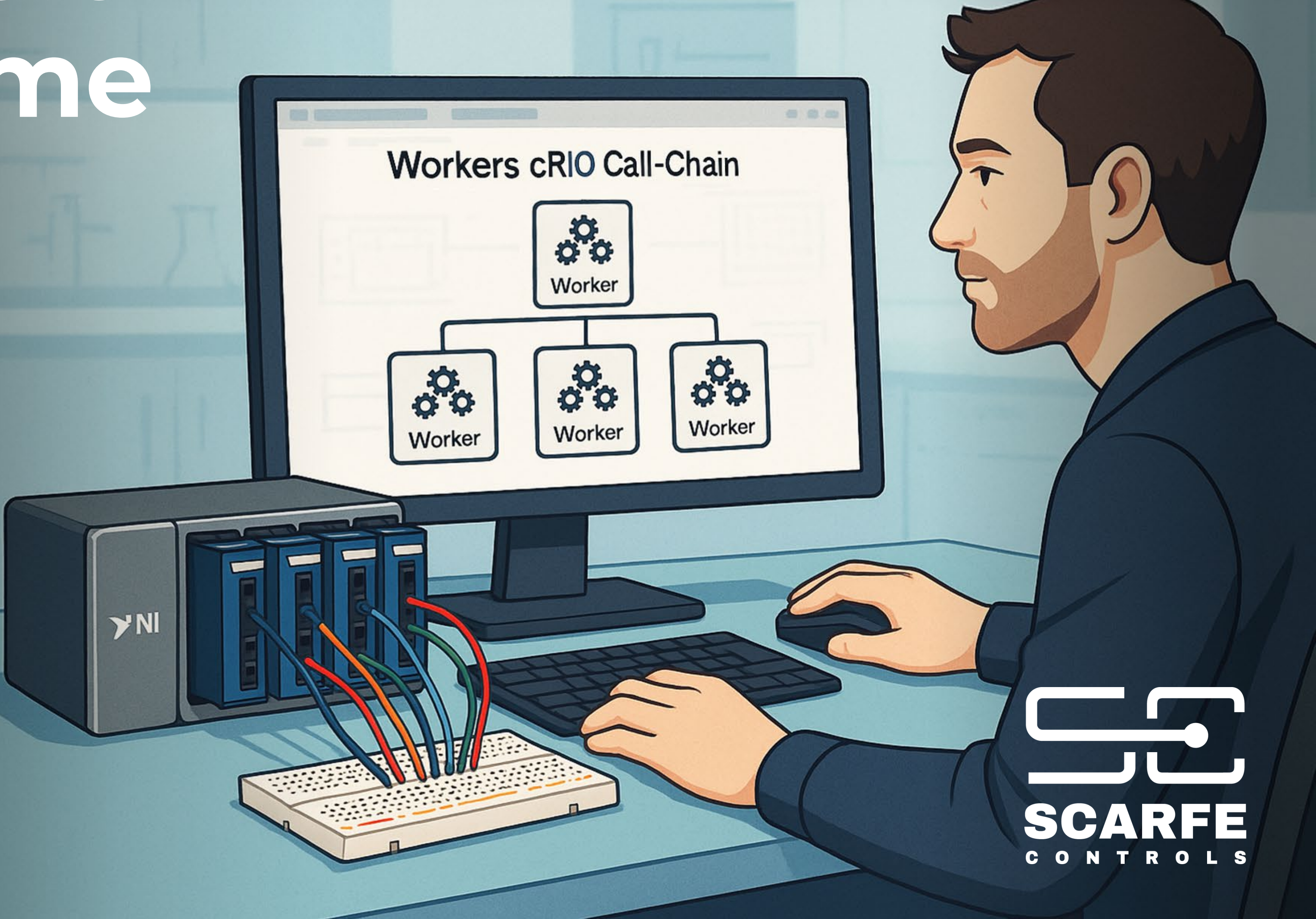
Workers 5.0 on NI **Real-time** Targets

Peter Scarfe, PhD
LabVIEW Champion



GLA Summit 2025

SCARFE CONTROLS

What is...

Workers for LabVIEW ?

Workers is an asynchronous modular framework **for LabVIEW**

Workers for LabVIEW – the Four Pillars

NI QMH Development Style

Built on the familiar NI QMH. Easy to develop, read and debug. Developer friendly, even at the Core3/CLD level.

LVOOP Modules

Uses self-contained, reusable modules, while providing the power of abstraction, inheritance and dynamic overriding.

Hierarchy Structure

Workers are organized into a structured call-chain, providing scalability while reducing spaghetti architecture.

Plug-in Architecture

Workers are decoupled from their Callers. Promotes reuse of modules. Allows for dynamic injection on HALs

Workers for LabVIEW – Main Features

Scripted Dev Tools

The tools do the heavy lifting for you, providing quick, easy, structured development. Allows you to stay focused on your logic.

Worker APIs

Create clearly defined APIs for your Workers. Clean interfaces. Clean communication. Built for concurrency.

API Abstractions and HALs

Create HALs easily with the scripted dev tools. Mock, test, and replace Workers without breaking your architecture.

Workers Debug Server

Know exactly what Workers are running on your local network. Visualize every message, error, and status live.

Workers for LabVIEW

- with -

NI Real-Time

NI CompactRIO Evolution – Then vs Now

Modern improvements compared to cRIOs 20 years ago

Feature	Early cRIOs (2004–2008)	Modern cRIOs (2020–2025)	Improvement Factor
CPU Power	~500 DMIPS	~10,000–25,000 DMIPS	~20x to 50x
CPU Cores	Single-core only	Dual-core and Quad-core available	2x to 4x
RAM	64–128 MB	512 MB – 4 GB	8x to 32x
Storage	128–256 MB Flash	4 – 64+ GB eMMC / SSD	16x to 256x
FPGA Fabric	~200K logic gates (Virtex-II)	1–2M logic elements (Zynq-7020, Kintex-7)	5x to 10x
Ethernet	10/100 Mbps	Gigabit Ethernet, Time-Sensitive Networking	10x
Application Size	Small, monolithic apps only	Large modular apps with HALs, dynamic loading	10x capability

Can use **Workers for LabVIEW !** ☺

Why use Workers on NI Real-time ?

Modern Hardware Allows it

Today’s cRIOs are powerful enough to support structured, message-driven frameworks like Workers. No need to compromise on architecture.

One Architecture, Two Platforms

Use the same architecture and codebase on both Windows and RT targets. Less rework, better consistency, easier testing.

Modular & Scalable Architecture

Workers naturally break your app into isolated, reusable components. Ideal for growing, complex systems.

Tools fully support RT Development

All dev tools in the framework are also designed for RT development. From creating new Workers, Public APIs and HALs to debugging.

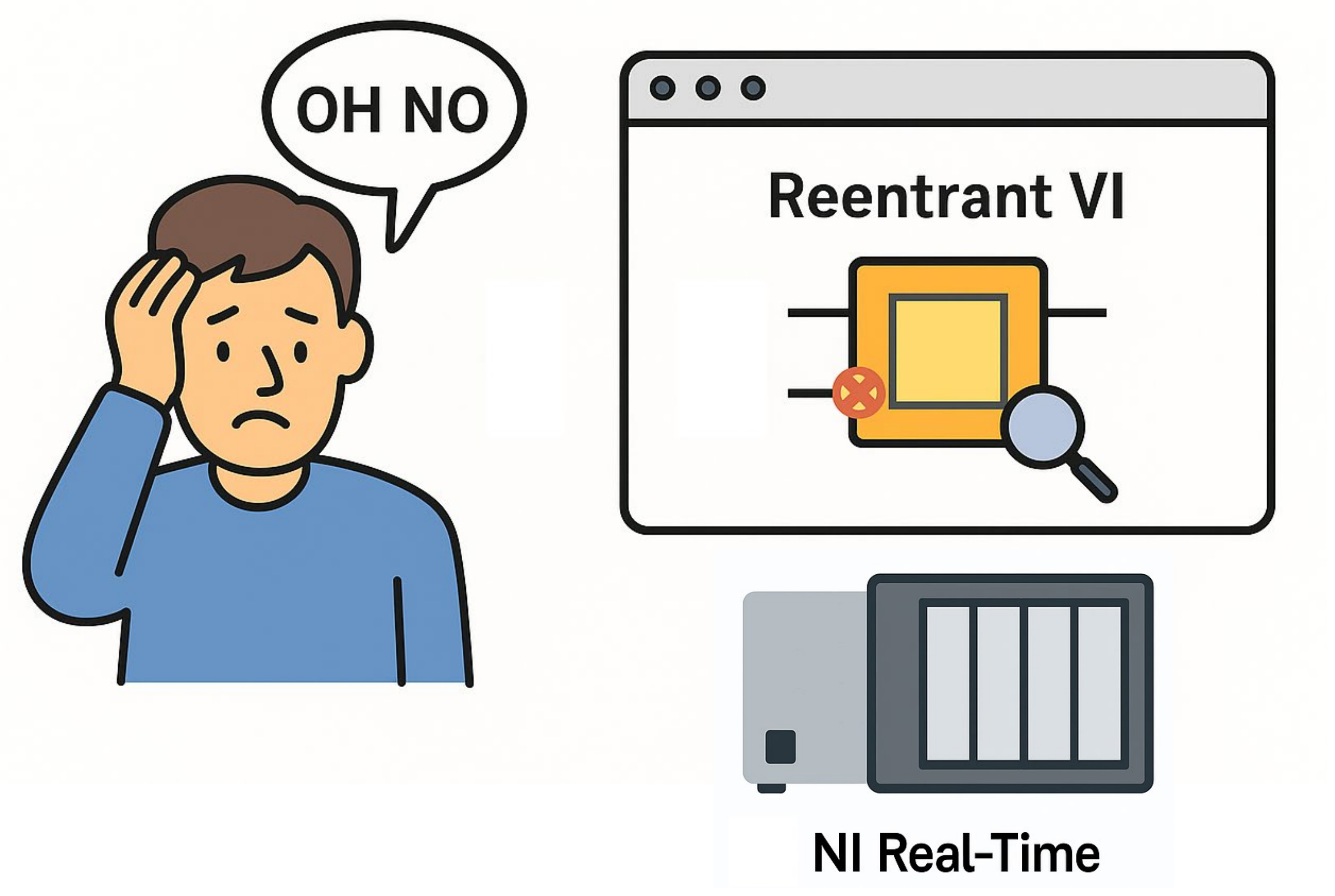
Problem #1

Debugging Reentrant VIs on RT Targets

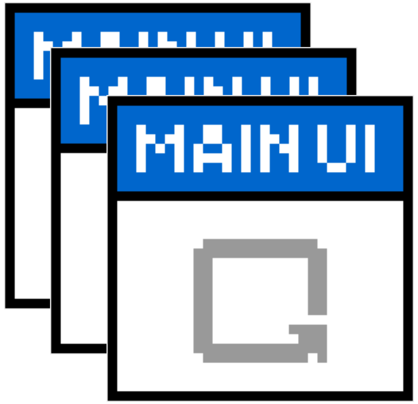
Debugging Reentrant VIs on Real-Time targets

Reentrant VIs on RT

- ❌ Can’t pause execution
- ❌ Can’t probe wires
- ❌ Can’t add breakpoints
- ❌ Painful to debug !



Reentrancy of a Worker’s Main VI



Main UI : **Main.vi**

Reentrant VI

Multiple instances allowed, but *not* RT debug-friendly

Default for all new Windows Workers

Main UI : **Main NR.vi**

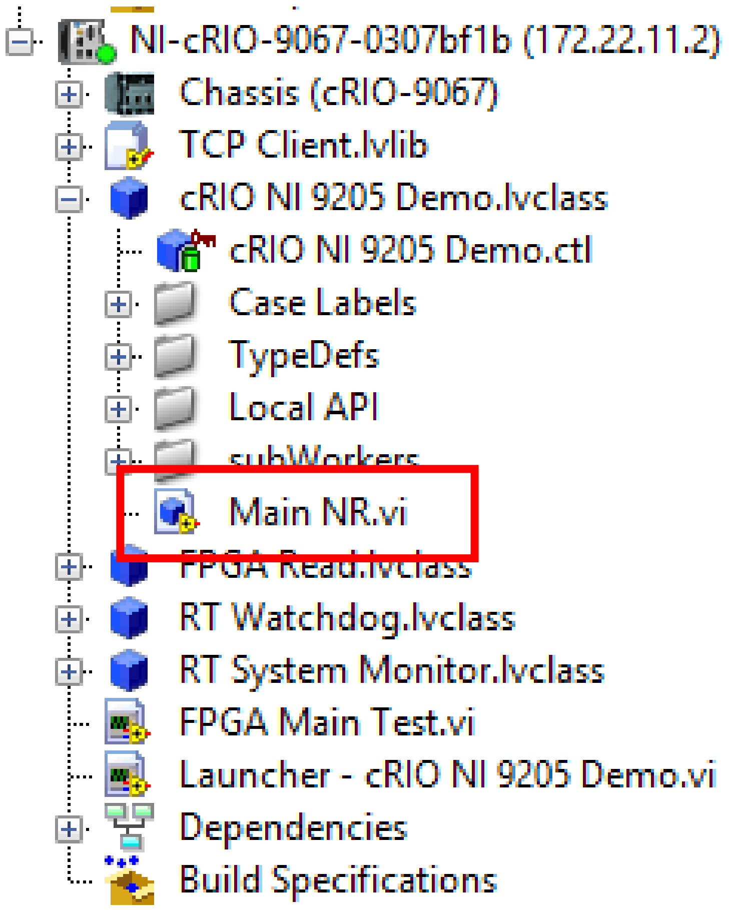
Non-reentrant VI

Single instance only, RT debug-friendly

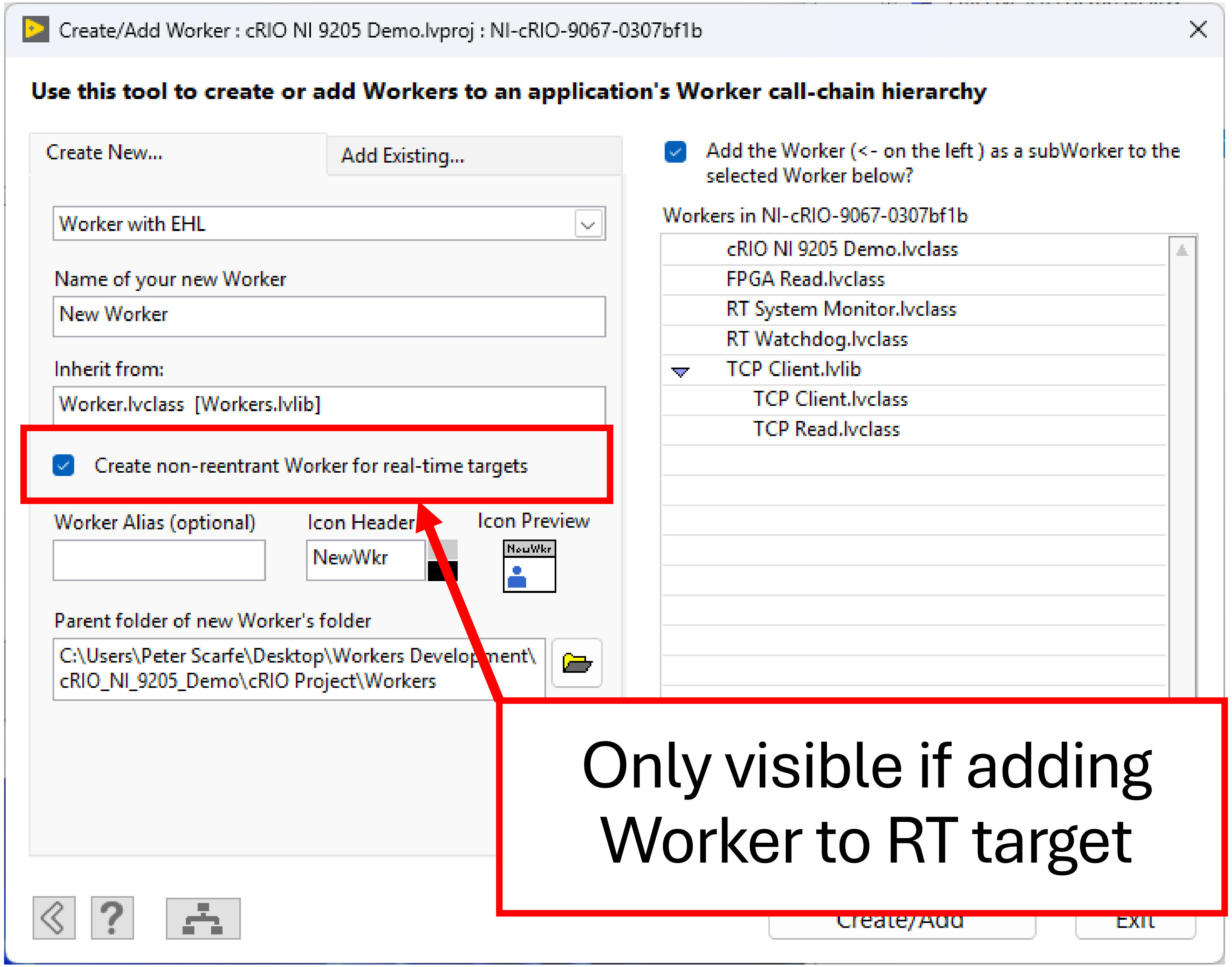
Default for all new real-time Workers

Creating a new Worker

With a **N**on-**R**eentrant Main VI



- NI-cRIO-9067-0307bf1b (172.22.11.2)
  - Chassis (cRIO-9067)
  - TCP Client.lvlib
  - cRIO NI 9205 Demo.lvclass
    - cRIO NI 9205 Demo.ctl
    - Case Labels
    - TypeDefs
    - Local API
    - subWorkers
    - Main NR.vi
  - FPGA Read.lvclass
  - RT Watchdog.lvclass
  - RT System Monitor.lvclass
  - FPGA Main Test.vi
  - Launcher - cRIO NI 9205 Demo.vi
  - Dependencies
  - Build Specifications



Create/Add Worker : cRIO NI 9205 Demo.lvproj : NI-cRIO-9067-0307bf1b

Use this tool to create or add Workers to an application's Worker call-chain hierarchy

Worker with EHL

Name of your new Worker: New Worker

Inherit from: Worker.lvclass [Workers.lvlib]

☑ Create non-reentrant Worker for real-time targets

Worker Alias (optional):

Icon Header: NewWkr

Parent folder of new Worker's folder: C:\Users\Peter Scarfe\Desktop\Workers Development\cRIO_NI_9205_Demo\cRIO Project\Workers

☑ Add the Worker (<- on the left) as a subWorker to the selected Worker below?

Workers in NI-cRIO-9067-0307bf1b

- cRIO NI 9205 Demo.lvclass
- FPGA Read.lvclass
- RT System Monitor.lvclass
- RT Watchdog.lvclass
- TCP Client.lvlib
  - TCP Client.lvclass
  - TCP Read.lvclass

Only visible if adding Worker to RT target

RT Worker Convert tool

Convert between Main.vi <-> Main NR.vi

🔄 Converts Main.vi to Main NR.vi

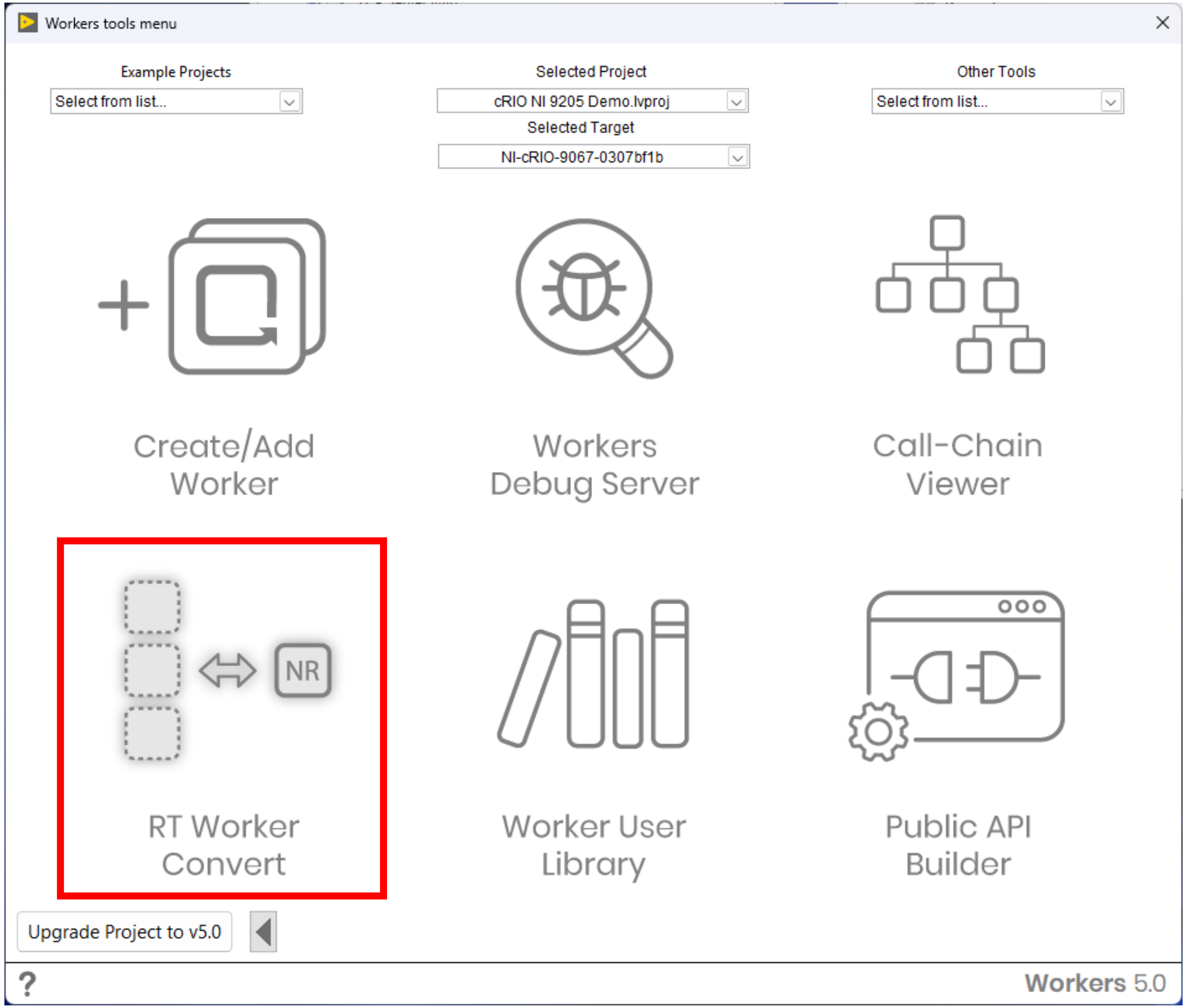
Use breakpoints, step through logic, probe wires.

🛠️ Switch Reentrancy Forms

The flexibility to both debug and run multiple instances of the Main VI.

✅ Automates Safe Conversion

Converts and replaces the VI throughout your code



Workers tools menu

Example Projects: Select from list...

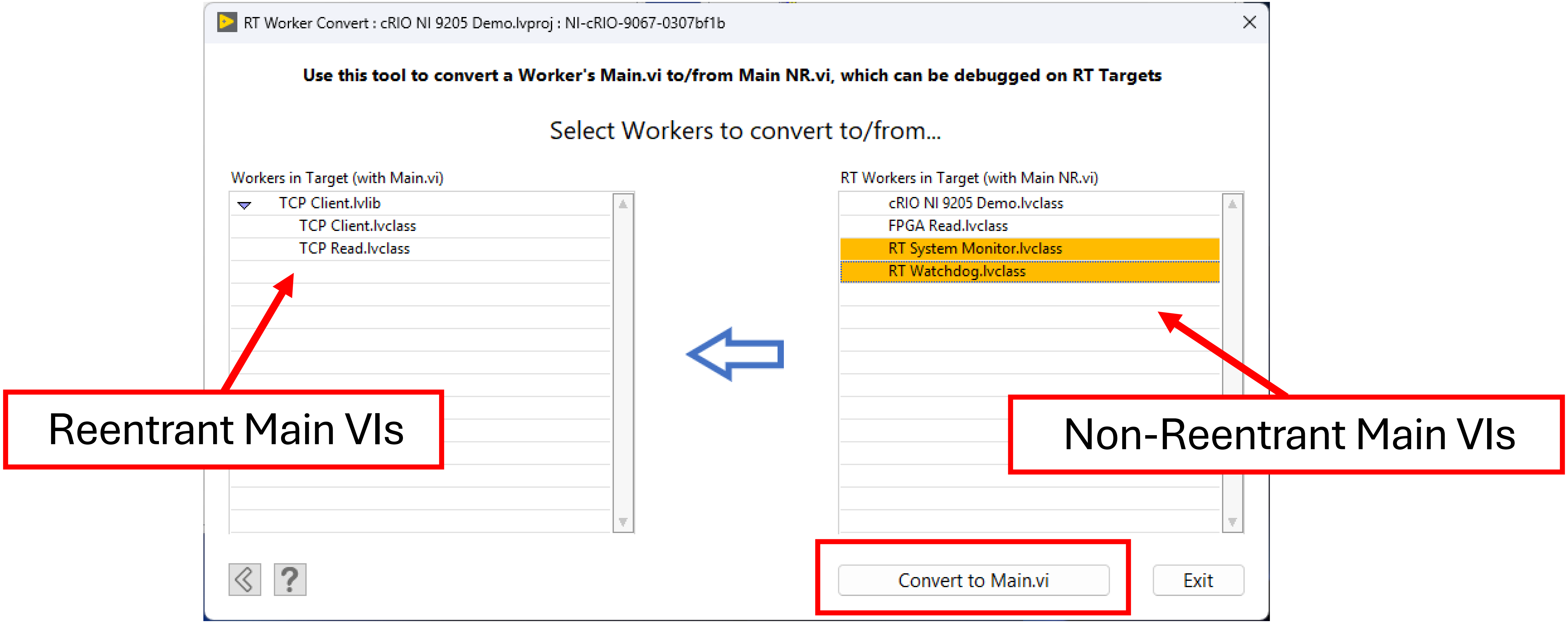
Selected Project: cRIO NI 9205 Demo.lvproj

Selected Target: NI-cRIO-9067-0307bf1b

Other Tools: Select from list...

RT Worker Convert tool : Main NR.vi to Main.vi

Convert between Main.vi <-> Main NR.vi



RT Worker Convert : cRIO NI 9205 Demo.lvproj : NI-cRIO-9067-0307bf1b

Use this tool to convert a Worker's Main.vi to/from Main NR.vi, which can be debugged on RT Targets

Select Workers to convert to/from...

Workers in Target (with Main.vi)

- TCP Client.lvlib
  - TCP Client.lvclass
  - TCP Read.lvclass

RT Workers in Target (with Main NR.vi)

- cRIO NI 9205 Demo.lvclass
- FPGA Read.lvclass
- RT System Monitor.lvclass
- RT Watchdog.lvclass

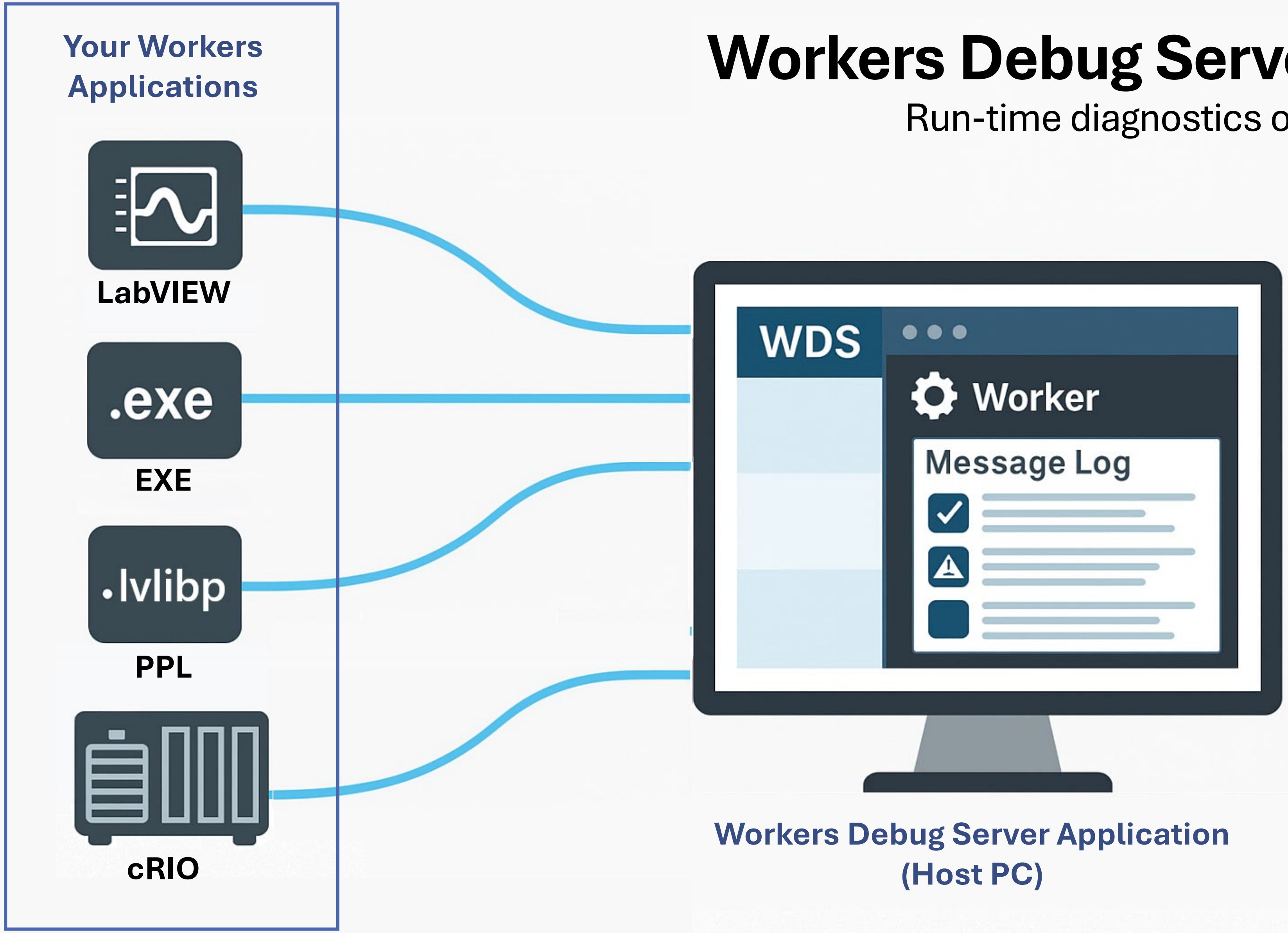
Problem #2

Run-time transparency of RT apps

Flashing LEDs on your cRIO are not enough !
It’s similar to a crying baby… you know there is something wrong… but what exactly ??

Workers Debug Server Application

Run-time diagnostics over your Local Network


Your Workers Applications
LabVIEW
.exe
EXE
.lvlibp
PPL
cRIO
WDS
Worker
Message Log
Workers Debug Server Application (Host PC)

Workers Application Manager

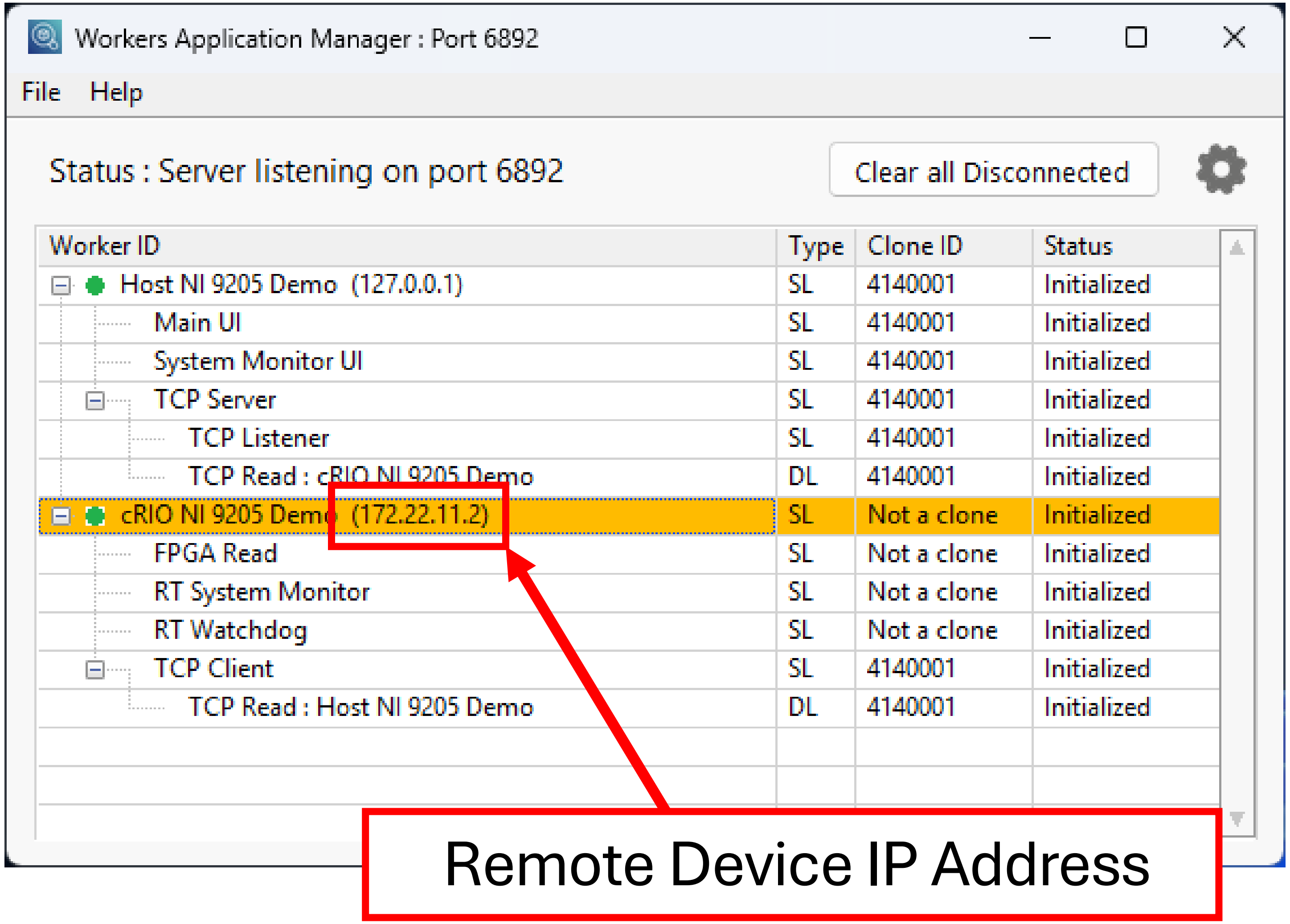
Running Workers on your Local Network

Application Manager for RT

❌ Can’t open Main VI’s FP/BD

✅ Can view Message Log

✅ Can shutdown remotely



Workers Application Manager : Port 6892

Status : Server listening on port 6892

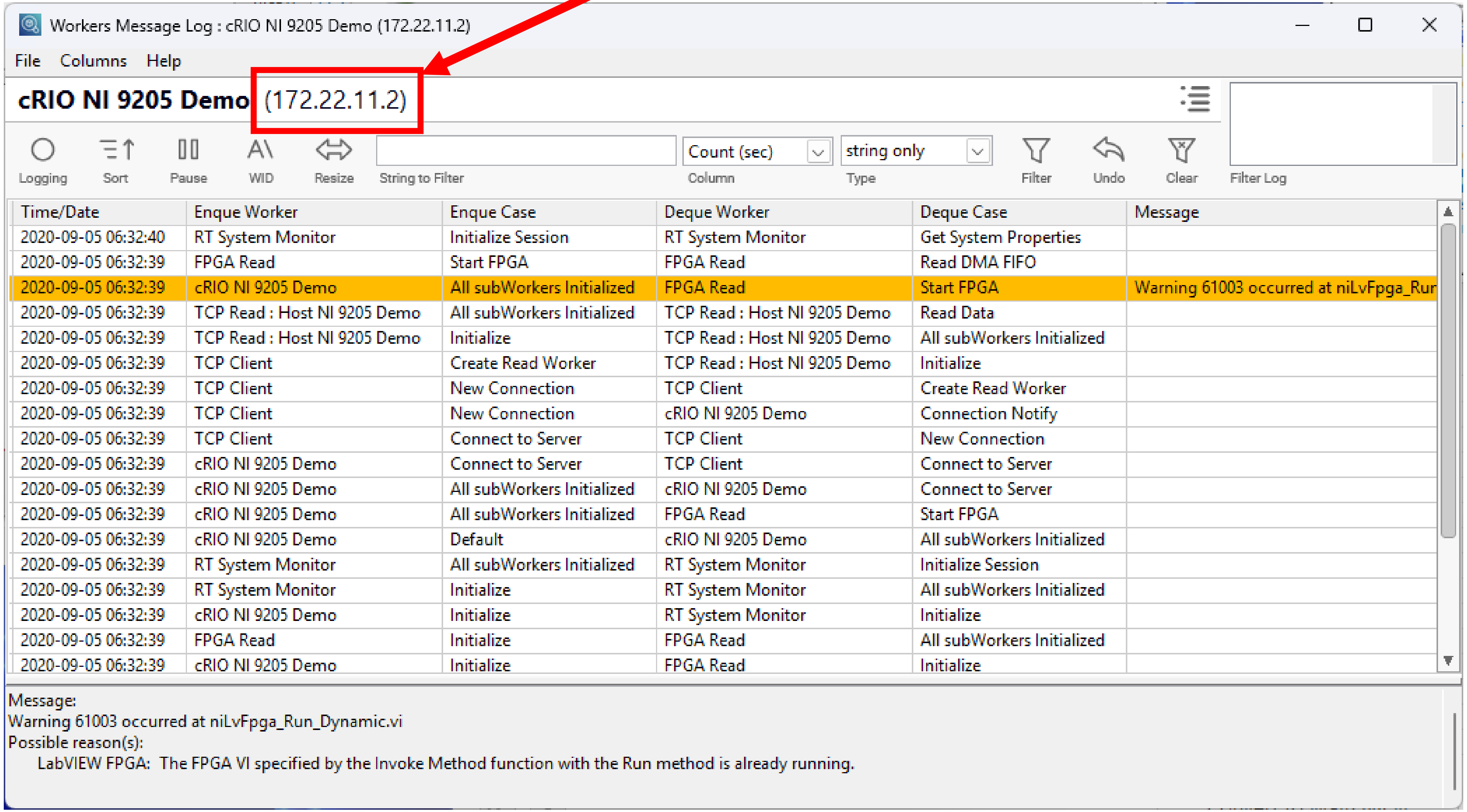
Worker ID	Type	Clone ID	Status
⊟ ● Host NI 9205 Demo (127.0.0.1)	SL	4140001	Initialized
Main UI	SL	4140001	Initialized
System Monitor UI	SL	4140001	Initialized
⊟ TCP Server	SL	4140001	Initialized
TCP Listener	SL	4140001	Initialized
TCP Read : cRIO NI 9205 Demo	DL	4140001	Initialized
⊟ ● cRIO NI 9205 Demo (172.22.11.2)	SL	Not a clone	Initialized
FPGA Read	SL	Not a clone	Initialized
RT System Monitor	SL	Not a clone	Initialized
RT Watchdog	SL	Not a clone	Initialized
⊟ TCP Client	SL	4140001	Initialized
TCP Read : Host NI 9205 Demo	DL	4140001	Initialized

Remote Device IP Address

Workers Message Log

See Worker message flow and errors on cRIO

Remote Device IP Address



Time/Date	Enque Worker	Enque Case	Deque Worker	Deque Case	Message
2020-09-05 06:32:40	RT System Monitor	Initialize Session	RT System Monitor	Get System Properties	
2020-09-05 06:32:39	FPGA Read	Start FPGA	FPGA Read	Read DMA FIFO	
2020-09-05 06:32:39	cRIO NI 9205 Demo	All subWorkers Initialized	FPGA Read	Start FPGA	Warning 61003 occurred at niLvFpga_Rur
2020-09-05 06:32:39	TCP Read : Host NI 9205 Demo	All subWorkers Initialized	TCP Read : Host NI 9205 Demo	Read Data	
2020-09-05 06:32:39	TCP Read : Host NI 9205 Demo	Initialize	TCP Read : Host NI 9205 Demo	All subWorkers Initialized	
2020-09-05 06:32:39	TCP Client	Create Read Worker	TCP Read : Host NI 9205 Demo	Initialize	
2020-09-05 06:32:39	TCP Client	New Connection	TCP Client	Create Read Worker	
2020-09-05 06:32:39	TCP Client	New Connection	cRIO NI 9205 Demo	Connection Notify	
2020-09-05 06:32:39	TCP Client	Connect to Server	TCP Client	New Connection	
2020-09-05 06:32:39	cRIO NI 9205 Demo	Connect to Server	TCP Client	Connect to Server	
2020-09-05 06:32:39	cRIO NI 9205 Demo	All subWorkers Initialized	cRIO NI 9205 Demo	Connect to Server	
2020-09-05 06:32:39	cRIO NI 9205 Demo	All subWorkers Initialized	FPGA Read	Start FPGA	
2020-09-05 06:32:39	cRIO NI 9205 Demo	Default	cRIO NI 9205 Demo	All subWorkers Initialized	
2020-09-05 06:32:39	RT System Monitor	All subWorkers Initialized	RT System Monitor	Initialize Session	
2020-09-05 06:32:39	RT System Monitor	Initialize	RT System Monitor	All subWorkers Initialized	
2020-09-05 06:32:39	cRIO NI 9205 Demo	Initialize	RT System Monitor	Initialize	
2020-09-05 06:32:39	FPGA Read	Initialize	FPGA Read	All subWorkers Initialized	
2020-09-05 06:32:39	cRIO NI 9205 Demo	Initialize	FPGA Read	Initialize	

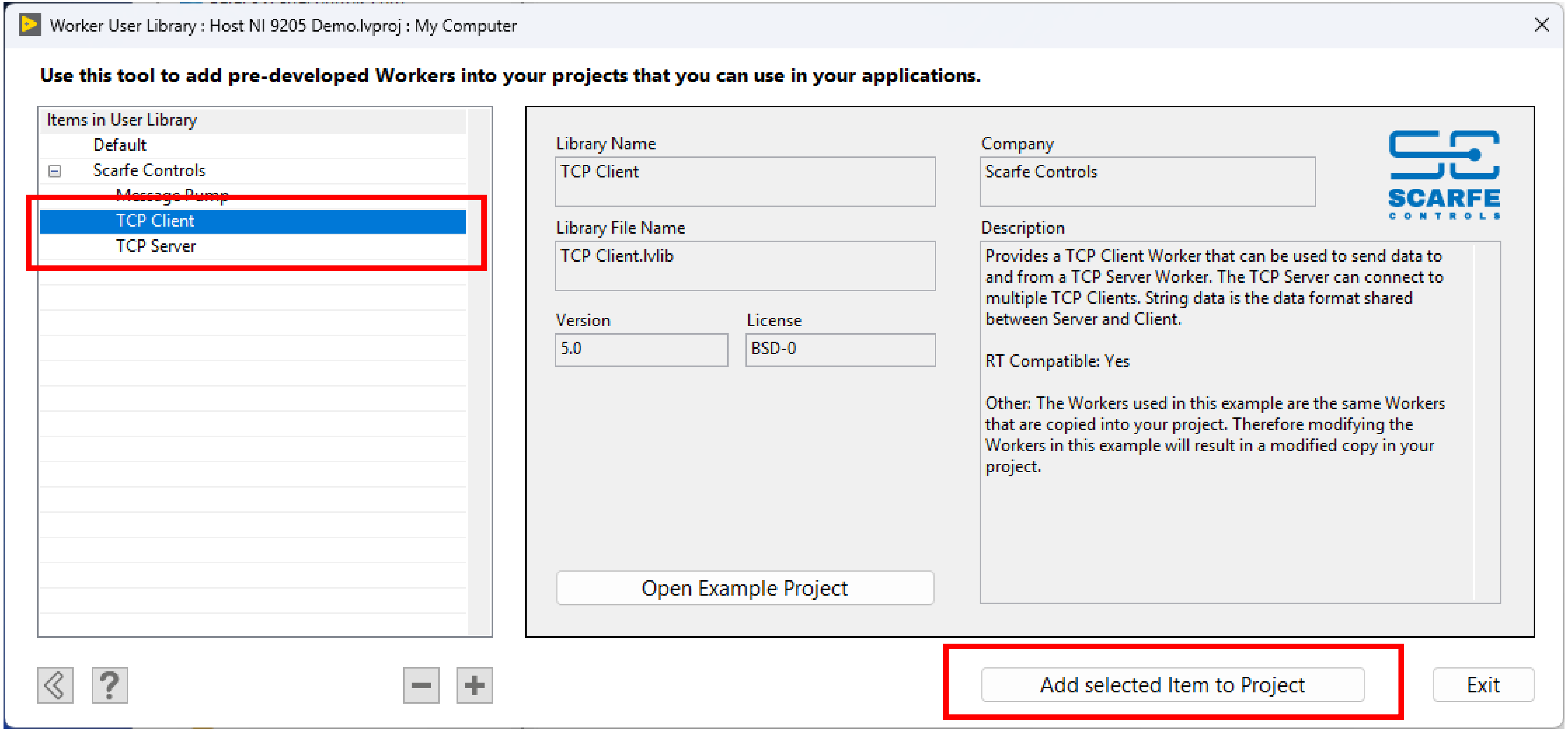
```
Message:
Warning 61003 occurred at niLvFpga_Run_Dynamic.vi
Possible reason(s):
    LabVIEW FPGA:  The FPGA VI specified by the Invoke Method function with the Run method is already running.
```

Problem #3

Streaming between Host and RT targets

TCP Server and Client Network Endpoint Workers

Supplied by default with the framework. Add to your project with **Worker User Library** tool



Worker User Library : Host NI 9205 Demo.lvproj : My Computer

Use this tool to add pre-developed Workers into your projects that you can use in your applications.

Library Name: TCP Client

Company: Scarfe Controls

Library File Name: TCP Client.lvlib

Version: 5.0

License: BSD-0

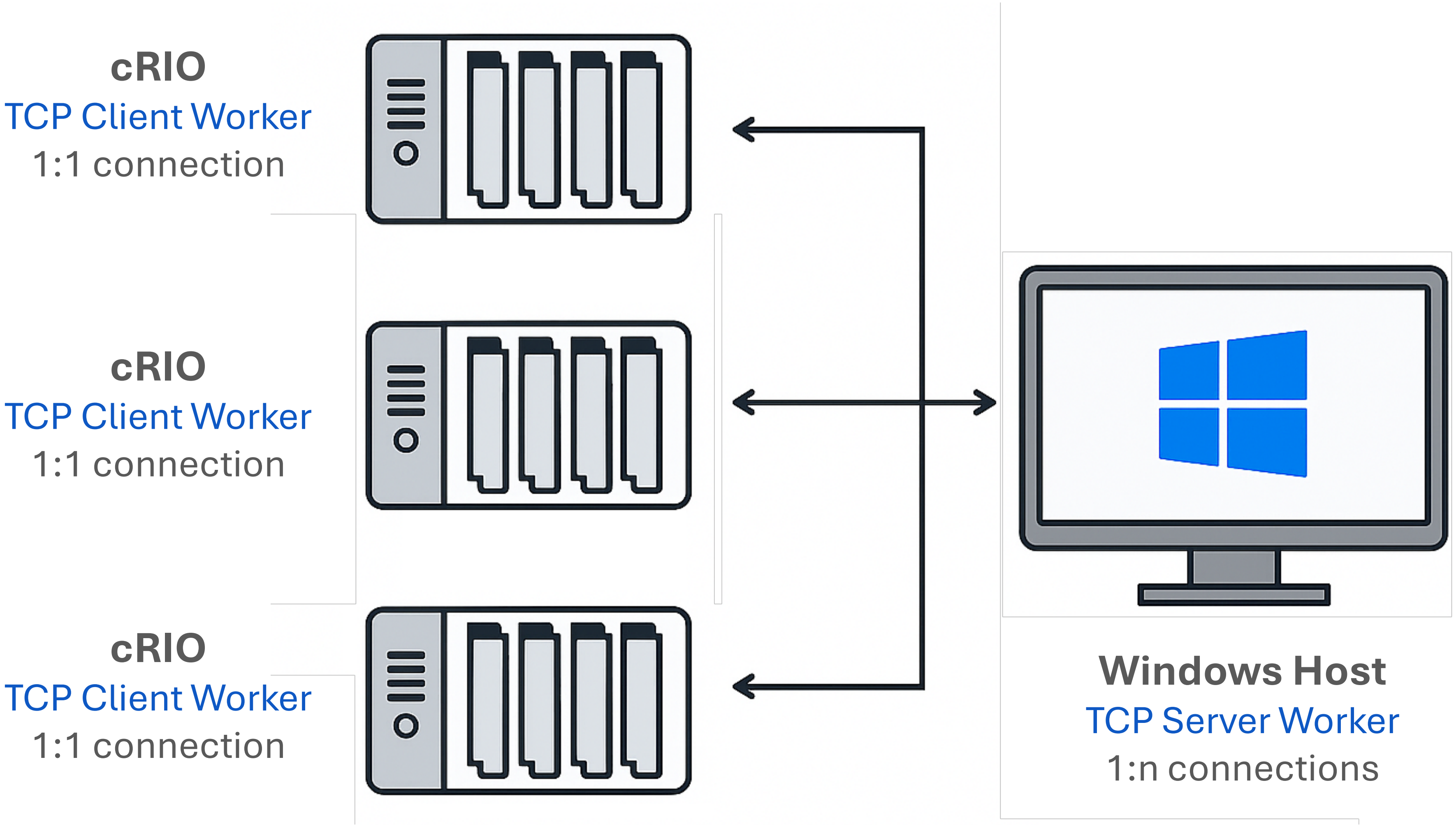
Description: Provides a TCP Client Worker that can be used to send data to and from a TCP Server Worker. The TCP Server can connect to multiple TCP Clients. String data is the data format shared between Server and Client.

RT Compatible: Yes

Other: The Workers used in this example are the same Workers that are copied into your project. Therefore modifying the Workers in this example will result in a modified copy in your project.

TCP Server and Client Workers

Supplied with the framework (Worker User Library)



cRIO
TCP Client Worker
1:1 connection

cRIO
TCP Client Worker
1:1 connection

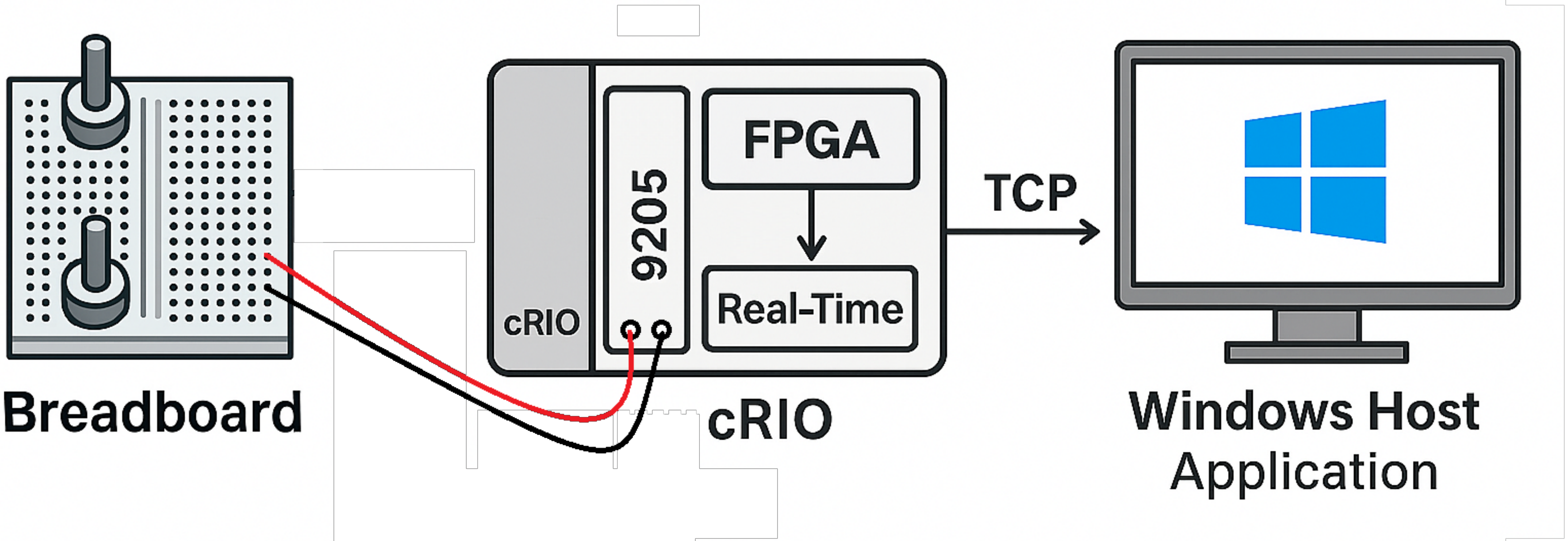
cRIO
TCP Client Worker
1:1 connection

Windows Host
TCP Server Worker
1:n connections

Demo Application on a cRIO

cRIO NI 9205 Workers Demo Setup

From potentiometer to Windows host application



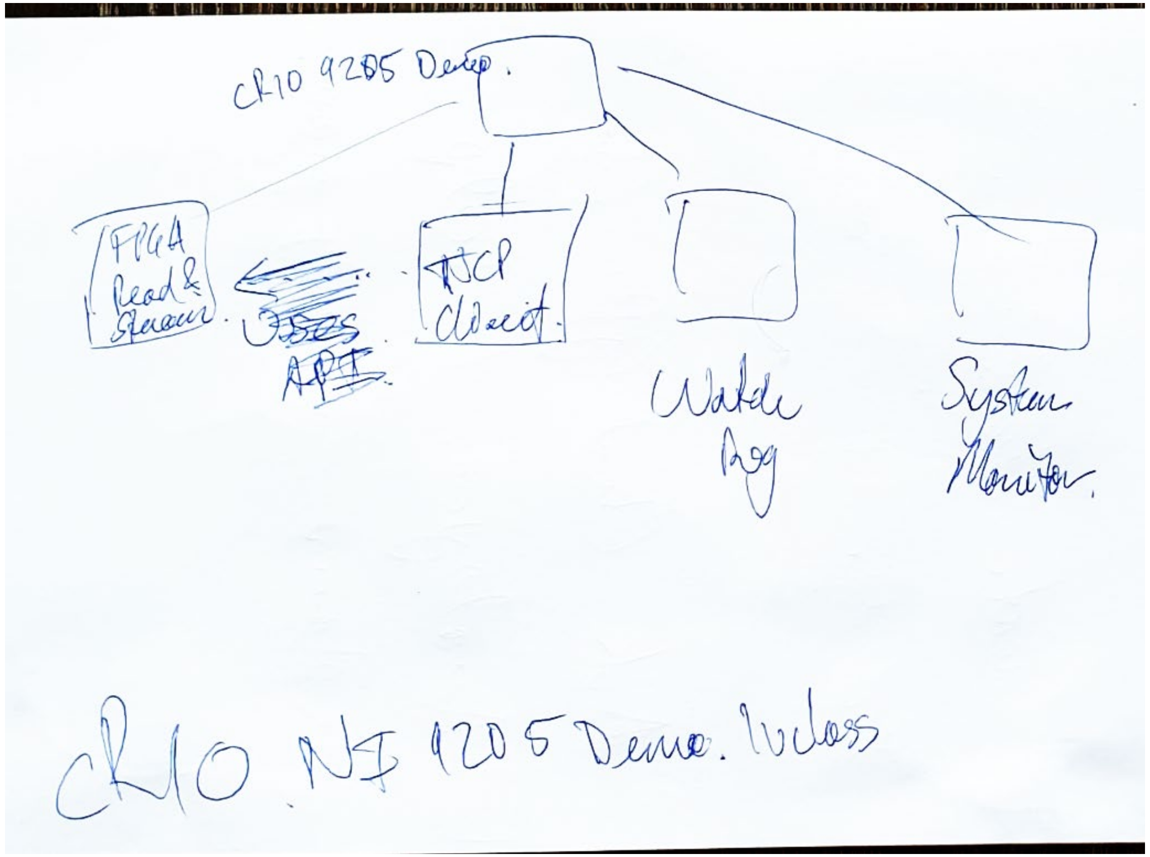
- 2 Potentiometers
  - 0 to 5V DC

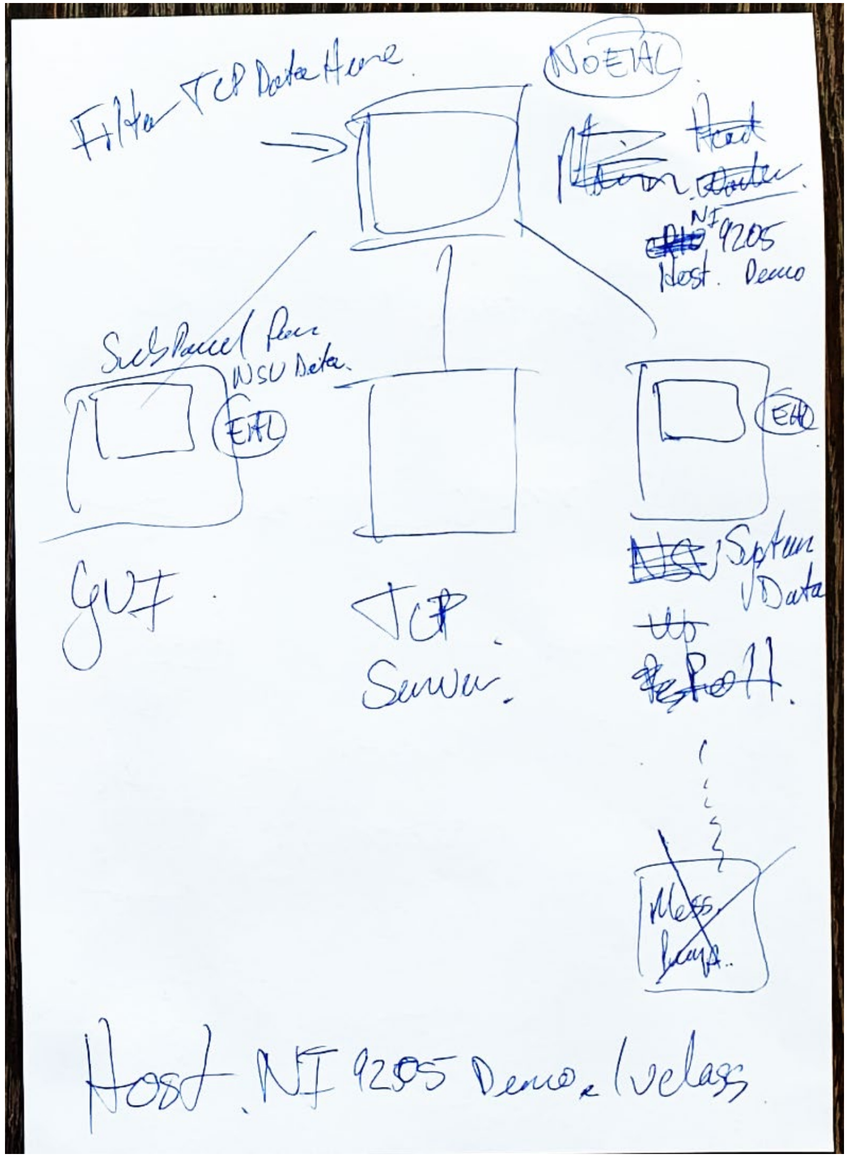
- NI 9205 DAQ module
- FPGA 10K sample rate
- Workers application on RT

- Workers application
- Displays Potentiometer Voltages

Start with a Simple Architecture Diagram

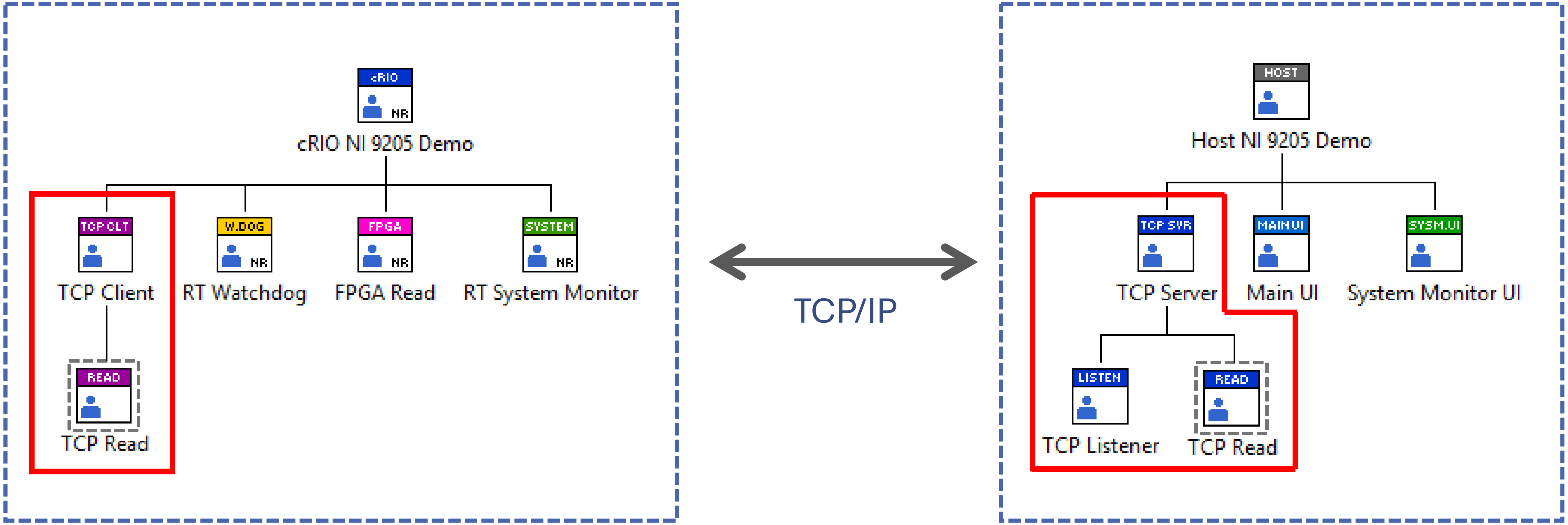
How will you will assemble your Workers ?





Use Create/Add Worker tool to build Worker call-chains

Plug Workers together to form applications



cRIO Application

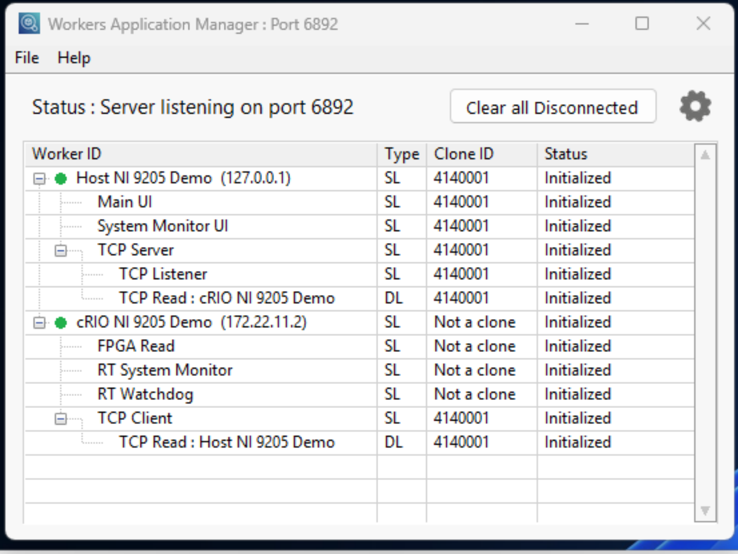
Windows Host Application

□ **TCP Server and Client Worker** libraries added with Worker User Library tool

Live Demo

Source Code for cRIO – HostPC Demo Available Here :

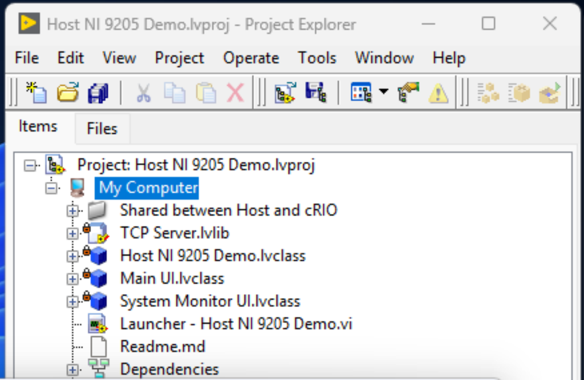
https://github.com/w4lv-community/Additional-Example-Projects/tree/Workers5.0/LabVIEW_2017



Workers Application Manager : Port 6892

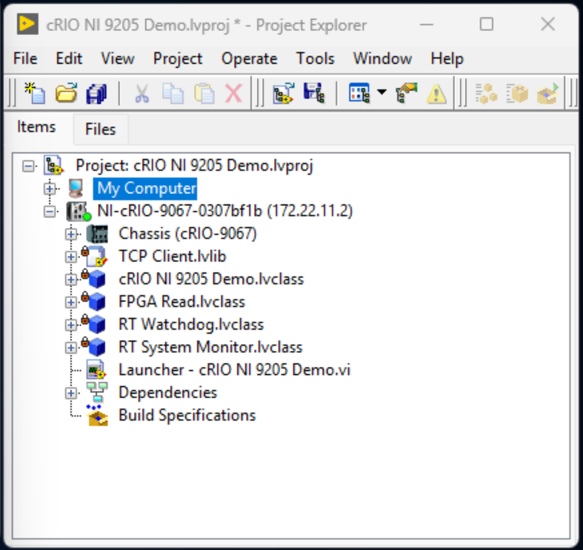
Status : Server listening on port 6892

Worker ID	Type	Clone ID	Status
Host NI 9205 Demo (127.0.0.1)	SL	4140001	Initialized
Main UI	SL	4140001	Initialized
System Monitor UI	SL	4140001	Initialized
TCP Server	SL	4140001	Initialized
TCP Listener	SL	4140001	Initialized
TCP Read : cRIO NI 9205 Demo	DL	4140001	Initialized
cRIO NI 9205 Demo (172.22.11.2)	SL	Not a clone	Initialized
FPGA Read	SL	Not a clone	Initialized
RT System Monitor	SL	Not a clone	Initialized
RT Watchdog	SL	Not a clone	Initialized
TCP Client	SL	4140001	Initialized
TCP Read : Host NI 9205 Demo	DL	4140001	Initialized



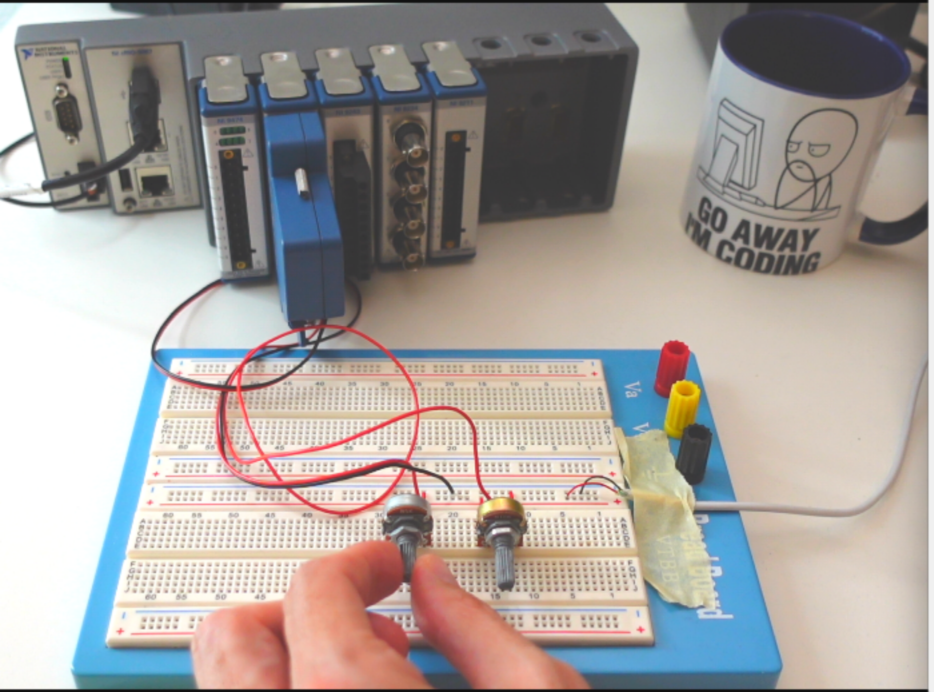
Host NI 9205 Demo.lvproj - Project Explorer

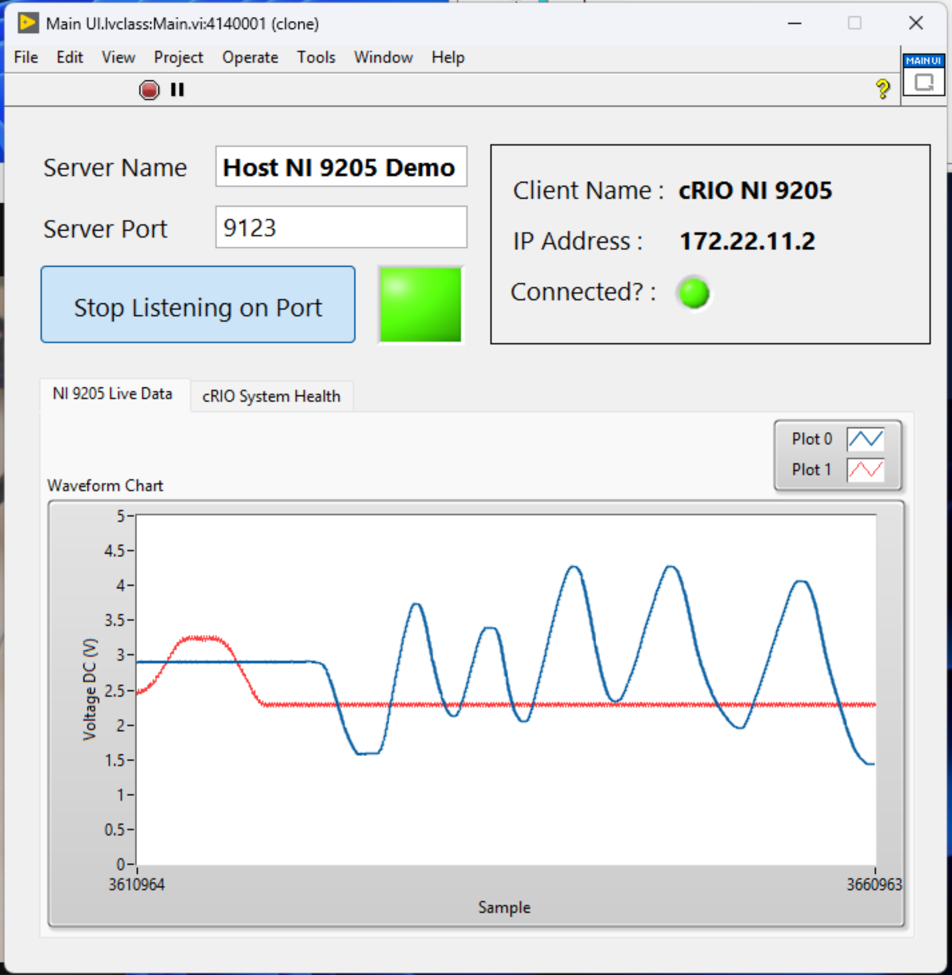
Project: Host NI 9205 Demo.lvproj
My Computer
Shared between Host and cRIO
TCP Server.lvlib
Host NI 9205 Demo.lvclass
Main UI.lvclass
System Monitor UI.lvclass
Launcher - Host NI 9205 Demo.vi
Readme.md
Dependencies



cRIO NI 9205 Demo.lvproj * - Project Explorer

Project: cRIO NI 9205 Demo.lvproj
My Computer
NI-cRIO-9067-0307bf1b (172.22.11.2)
Chassis (cRIO-9067)
TCP Client.lvlib
cRIO NI 9205 Demo.lvclass
FPGA Read.lvclass
RT Watchdog.lvclass
RT System Monitor.lvclass
Launcher - cRIO NI 9205 Demo.vi
Dependencies
Build Specifications





Main UI.lvclass:Main.vi:4140001 (clone)

Server Name: Host NI 9205 Demo

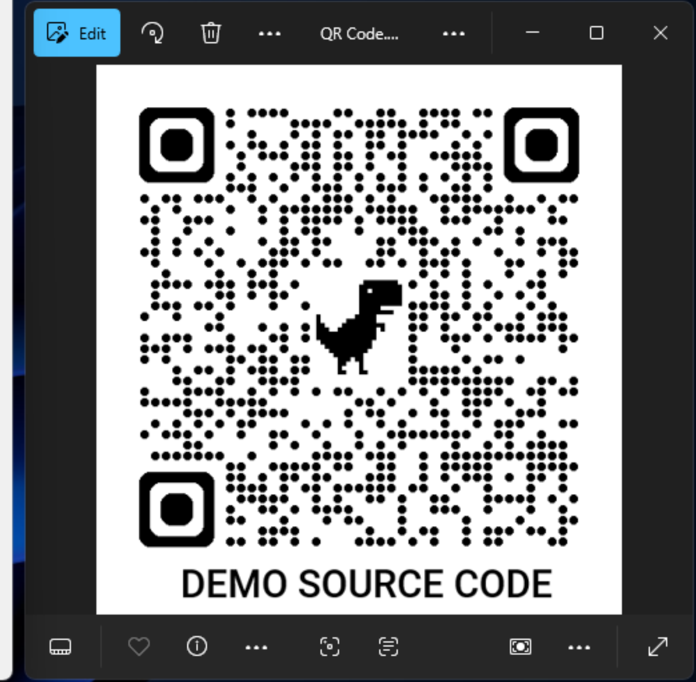
Server Port: 9123

Stop Listening on Port

Client Name : cRIO NI 9205

IP Address : 172.22.11.2

Connected? : ●



DEMO SOURCE CODE

wőrkers v5.0

⚠️ Two things to Avoid if you want to stay sane

1. ⚠️ **Avoid “Separate compiled code from source” flag on Classes for RT Deployment**

   - Deployment may fail on NI RT targets when this setting is enabled, especially when using LabVIEW classes . E.g. When deploying a Workers application !

   ✅ **Best practice**: Uncheck this flag for **Workers applications** that deploy to RT targets

2. ⚠️ **Avoid creating RT application and host application in same LabVIEW project.**

   - Workers tools don’t like it if you have multiple targets open that contain Workers when adding new Workers to the target.

   ✅ **Best practice**: Create one project for Host application and another for RT application.

workers for LabVIEW

Thank YOU 🙏

GLA Summit - 23 / 24th June 2025

SCARFE CONTROLS

Workers for LabVIEW Community
https://community.workersforlabview.io