Oslo - Stockholm - Utrecht - Brussels - Copenhagen - Helsinki

ni.com/nidays

NATIONAL INSTRUMENTS

PORTING YOUR WINDOWS APPLICATION TO REAL-TIME

Agenda

- General OS vs. RT OS, terminology
- NI Real-Time Solutions
- Porting from Windows Example
- Where Do I Learn More?

General Purpose Operating Systems

- Windows, Linux, MacOS, Unix
  - Processor time shared between programs
  - OS can preempt high priority threads
  - Service interrupts – keyboard, mouse, Ethernet…
  - Cannot ensure that code finish within specified time limits
- Extensions that add real-time cores are available

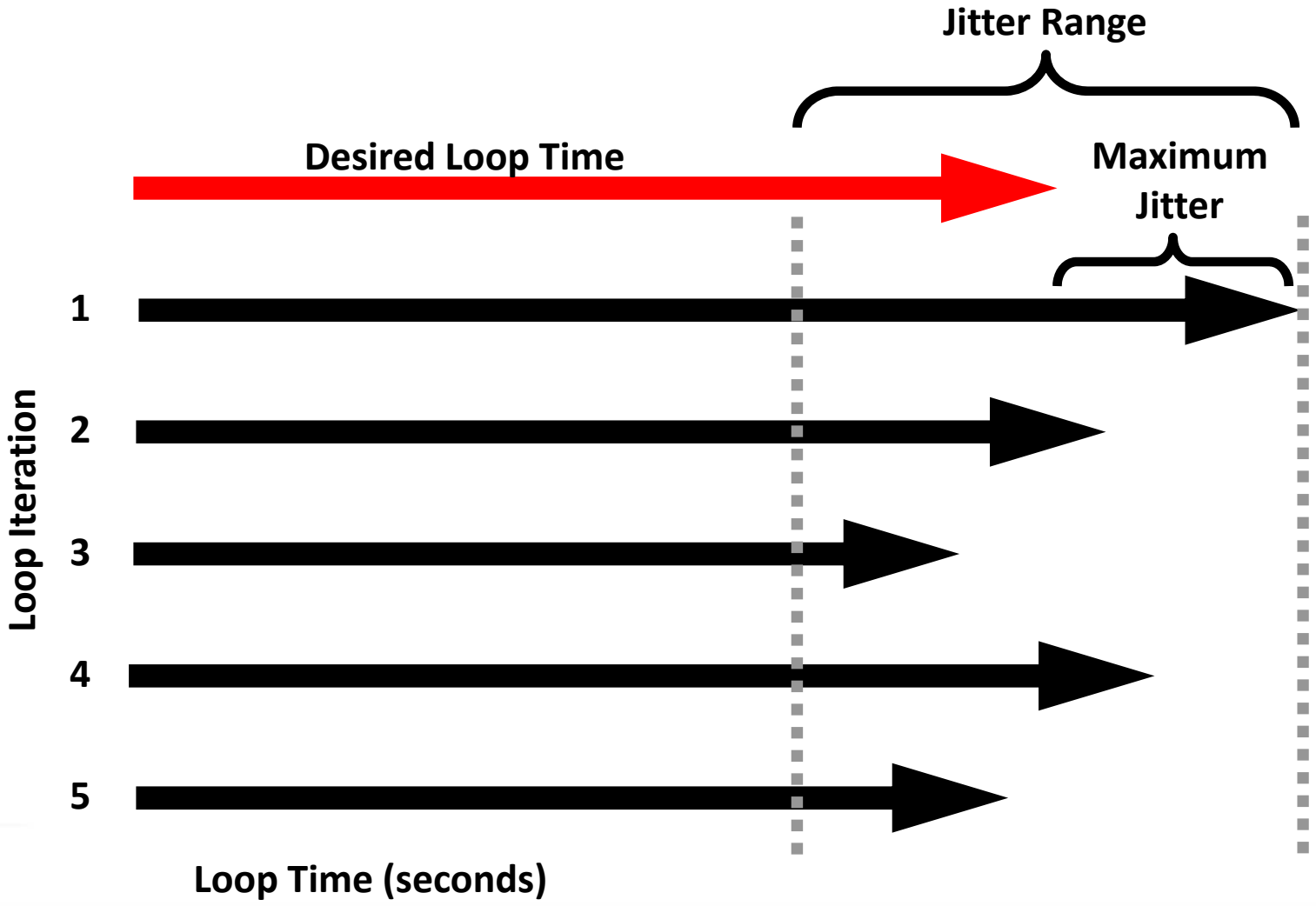
What is Real-Time?

- Common misconception:
  - Real-Time equals quick/fast
- Real-Time refers to ‘in-time’
  - Responses occur in time
  - Responses occur on time
  - Meet timing constraints

Common Real-Time Terminology

- Determinism
- Loop Cycle Time (T)
- Sample Rate (1/T)
- Jitter
- Embedded

Maximum Jitter

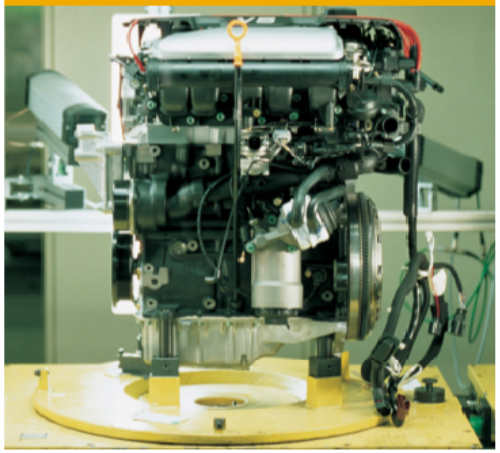


Spectrum of Real-Time Applications

Deterministic Performance

Wind Tunnel Control

Maximum Reliability



Endurance Testing

Autonomous Operation

Safety Monitoring

Selecting an Operating System

General Purpose OS

- Features
  - User interface
  - Enterprise connectivity
  - Peripheral interrupts
  - Background applications
  - OS that controls all scheduling
- Applications
  - Buffered data acquisition
  - Offline analysis
  - Data presentation

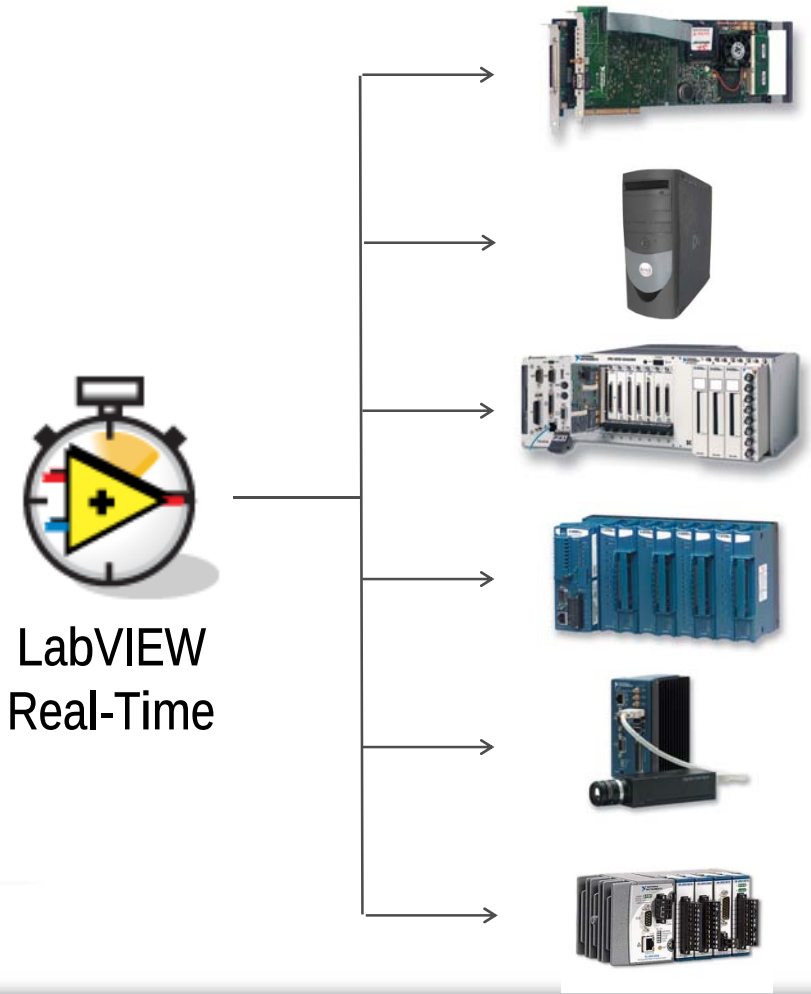
Real-Time OS

- Features
  - Embedded
  - Deterministic
  - Control over OS
  - Schedule that ensure that high-priority tasks execute first
- Applications
  - Closed Loop Control
  - Time-critical decision making
  - Extended run time
  - Increased reliability
  - Standalone operation

What is LabVIEW Real-Time?

- Add-on module to LabVIEW
  - Same graphical programming paradigm as LabVIEW
  - Same environment for Windows and RT applications
  - Rapid development of deterministic applications
  - Wide variety of analysis and control routines
  - Architect distributed control and monitoring systems
  - Integrate diverse I/O using full-featured HW drivers

LabVIEW Real-Time Targets



PCI Plug-In Device

Real-time component for PCs

Desktop PCs

Determinism for PCI systems

Real-Time PXI Embedded Controllers

High speed, high channel density, I/O variety

RT Compact FieldPoint

Small footprint, harsh environments

RT Compact Vision System

Compact and distributed machine vision

CompactRIO

Reconfigurable Embedded System

What is LabWindows/CVI Real-Time?

- Add-on module to LabWindows/CVI
  - Create deterministic ANSI-C applications
  - Same environment for Windows and RT applications
  - Reuse existing C-Code
  - Many analysis routines available
  - Architect distributed control and monitoring systems
  - Integrate diverse I/O using full-featured HW drivers

LabWindows/CVI Real-Time Targets

Desktop PCs

Determinism for PCI systems

Real-Time PXI Embedded Controllers

High speed, high channel density, I/O variety

LabVIEW RT Development

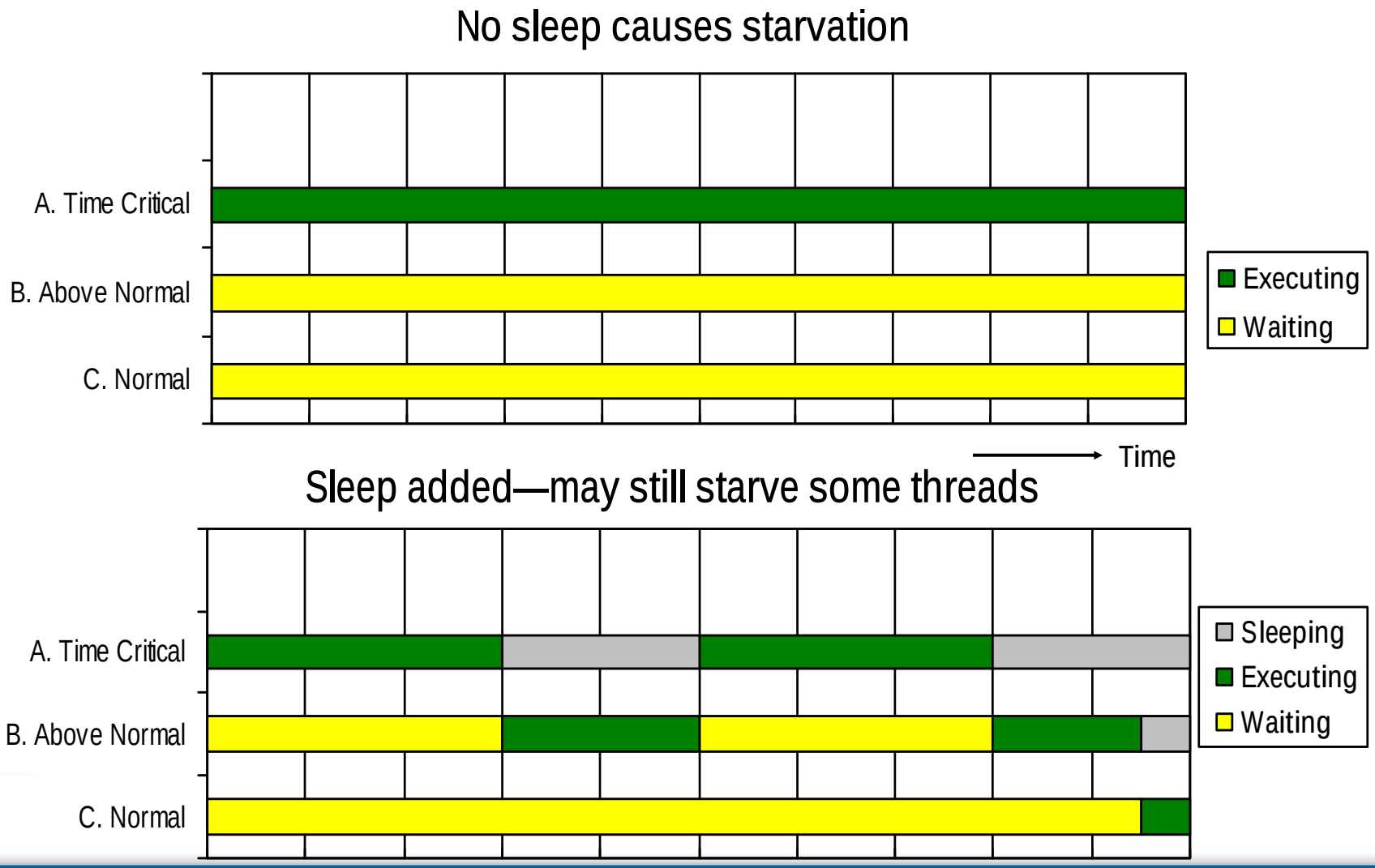
Host Computer

RT Target

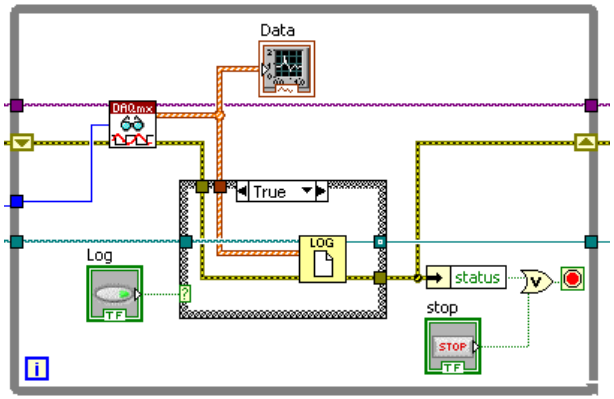
How Does LabVIEW RT Guarantee Determinism?

- Assign priorities:
  - Assign time-critical priority to code
  - Preemptive schedule guarantees processor bandwidth for time-critical code
- Risks:
  - Unresponsive systems due to starvation

What is starvation?



Porting from Windows Example



- Existing application
  - Acquires voltage data
  - Logs it to file
  - Displays data on user interface
- New requirement
  - If voltage reaches threshold, output digital signal
  - Signal must be sent within 1 ms

When Porting to LabVIEW Real-Time ...

1. What do I do with the user interface?
2. How do I communicate with other systems?
3. How do I achieve determinism?
4. What should I avoid?
5. How do I debug real-time applications?

What do I do with the User Interface?

- Control / Indicators can be used for debugging
  - Should be removed during final deployment
  - SubVIs with TCL priority should NEVER have UI elements inside
- How to exchange data between loops?
  - Shared Variables with RT FIFOs enabled
  - RT FIFOs
- How to present the data?
  - Send data through network to host application
  - Make use of web server capabilities.

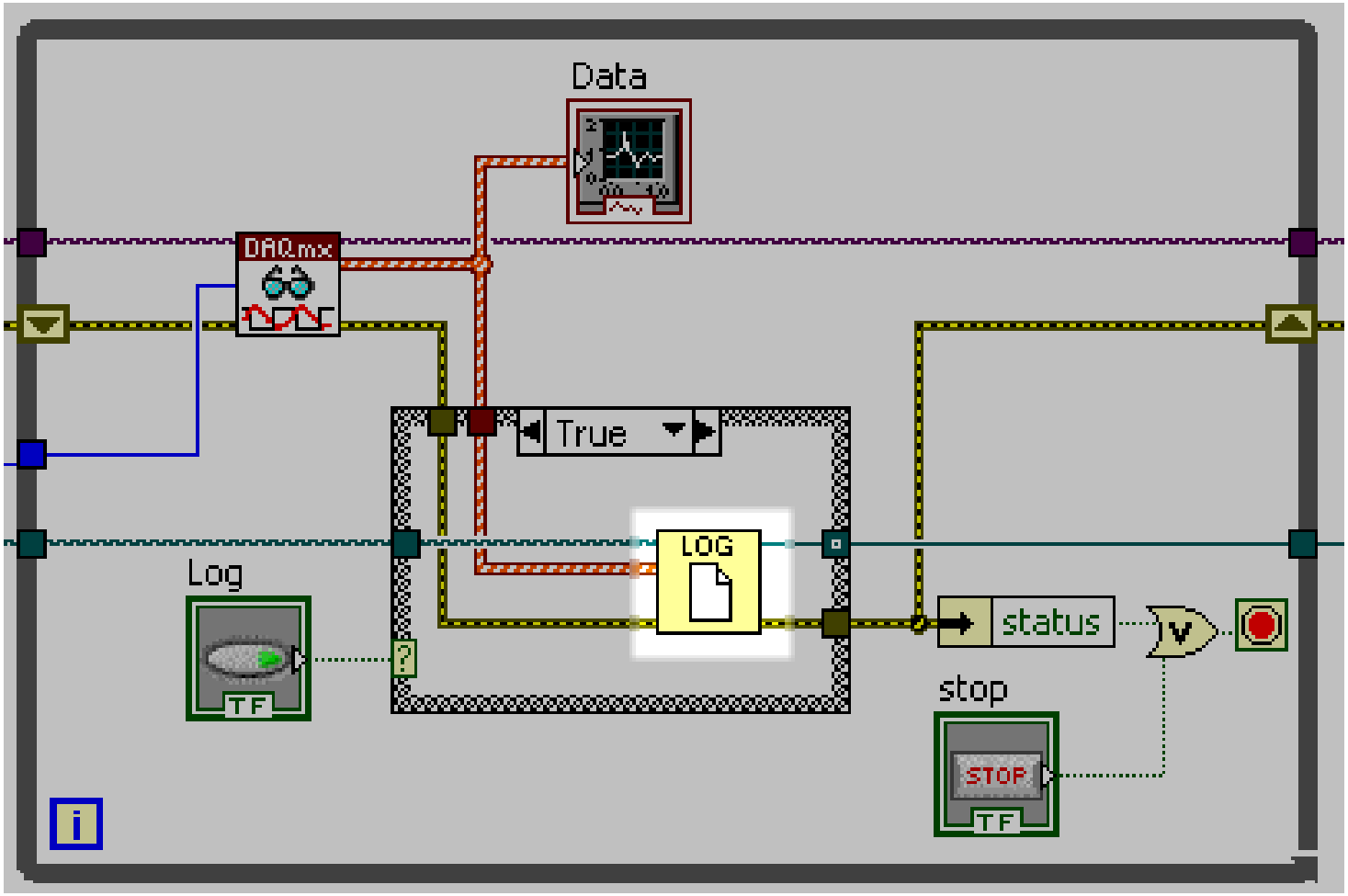
How Do I Communicate with Other Systems?

- Network Data Transfer
  - LabVIEW shared variable
  - TCP/UDP

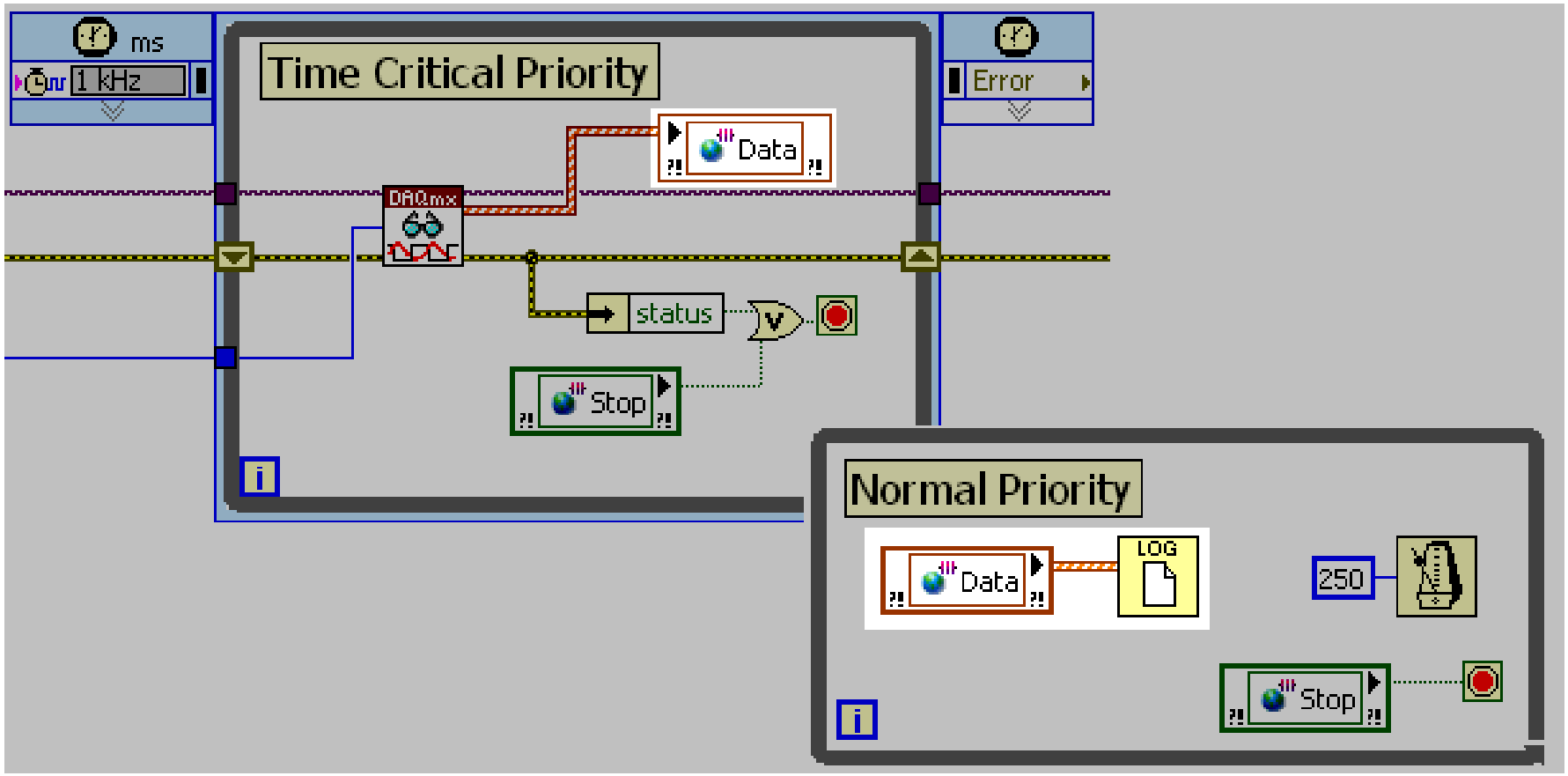
- Remote Application Control
  - VI server
  - Remote panels

- I/O Buses
  - Serial, GPIB, USB storage, IEEE 1394
  - CAN, DeviceNet, Modbus, etc.
  - Reflective memory

How do I achieve determinism?



How do I achieve determinism?

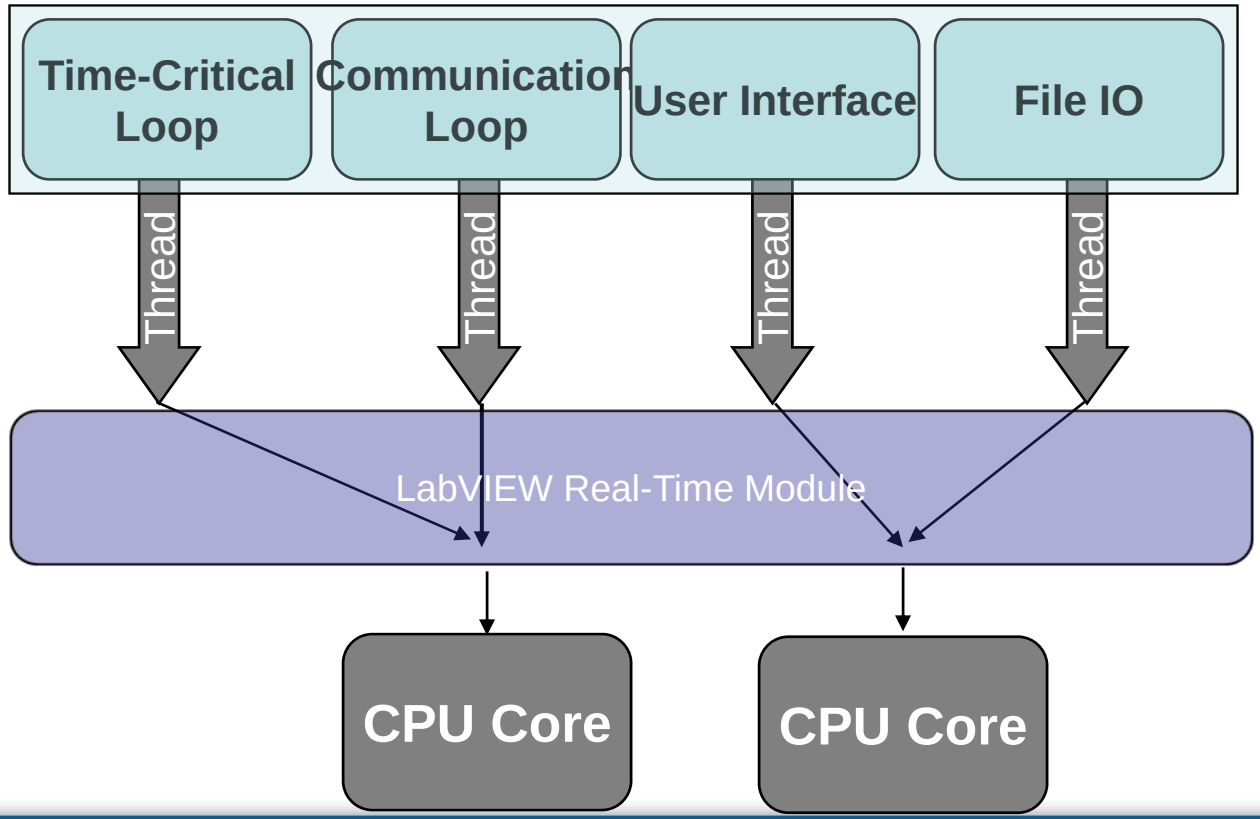


How do I achieve determinism?

- LabVIEW Real-Time support for multicore processors
- Two common implementations:
  - LabVIEW RTcan automatically load-balance applications i.e. automatic thread load-balancing
  - Timed loops can be used to assign a specific task to a processor core also referred to as assigning processor affinity
- Benefits for Real-Time Applications:
  - For applications that require many different tasks, SMP helps you do more tasks simultaneously (multitasking)
  - For applications that demand high-end performance, SMP offers better utilization of hardware resources

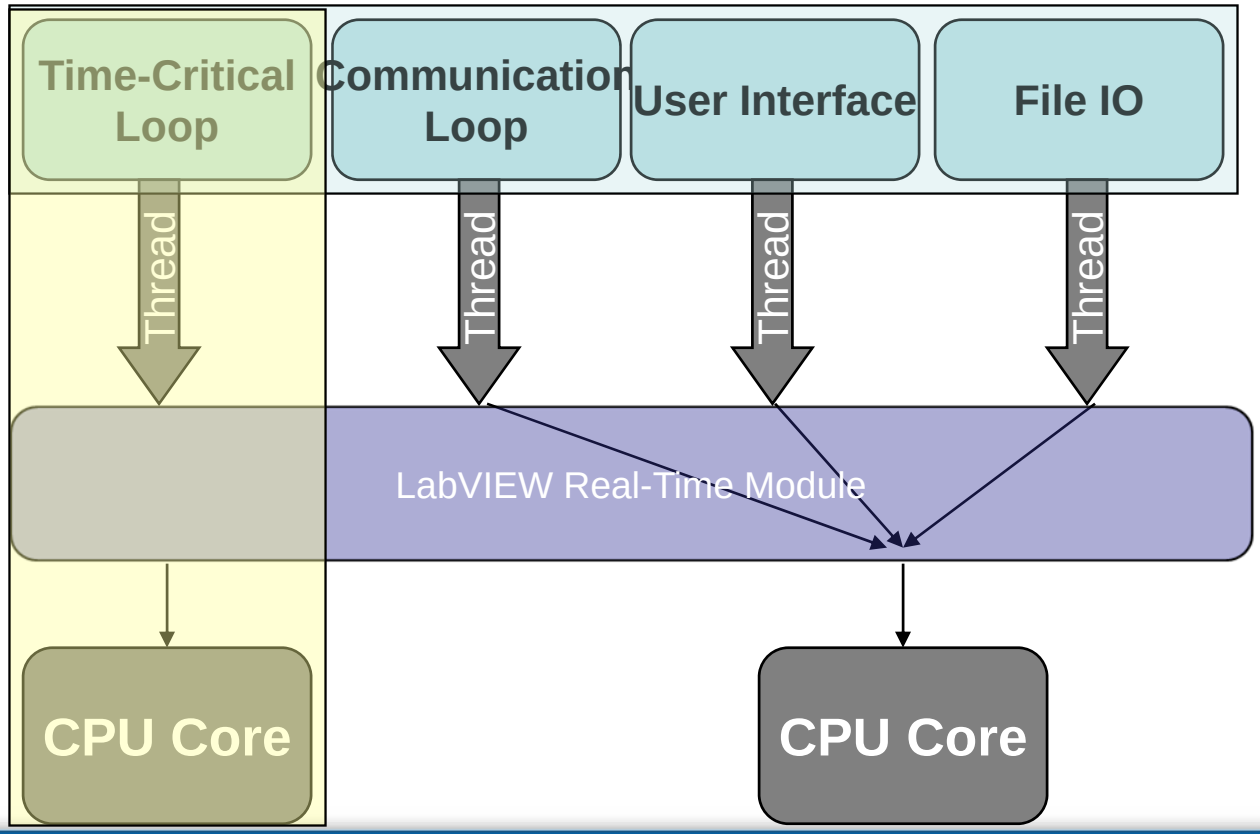
How do I achieve determinism?

- LabVIEW RT 8.5 adds automatic thread load-balancing (symmetric multiprocessing – SMP) to embedded systems.



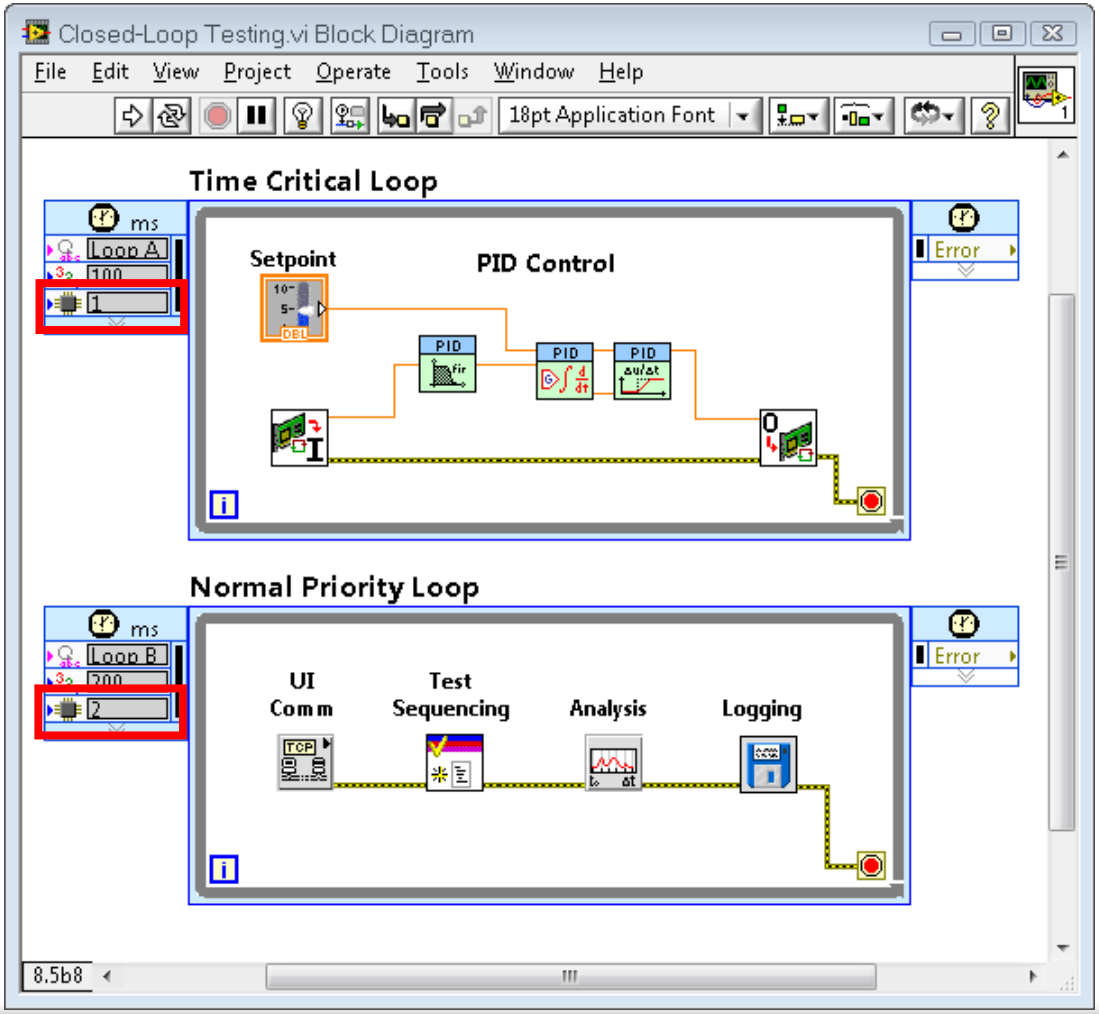
How do I achieve determinism?

- In LabVIEW 8.5, you can assign code to specific CPU cores using the LabVIEW timed loop.



How do I achieve determinism?

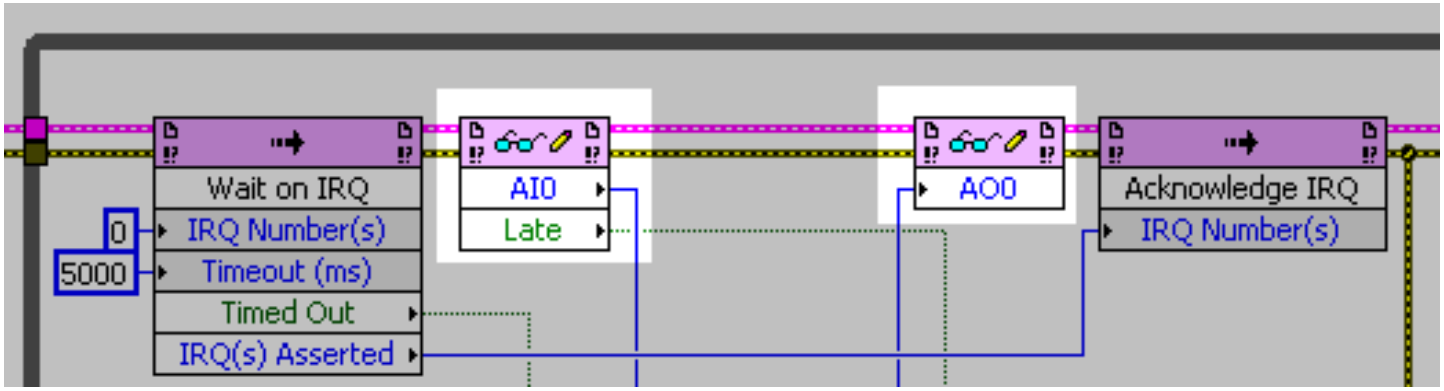
Assign timed loops to specific processor cores.



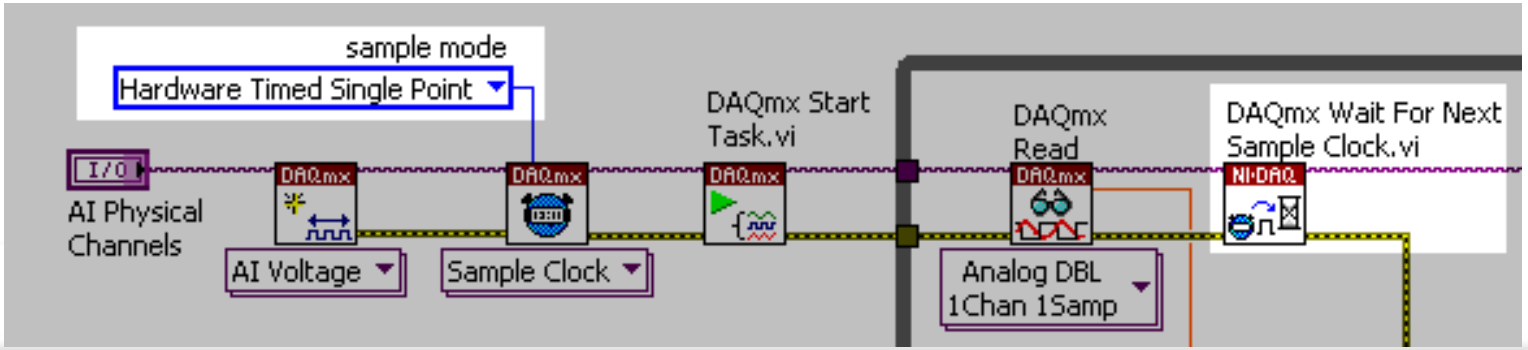
How do I achieve determinism?

Learn Real-Time I/O Optimizations

NI-RIO



NI-DAQmx



What to avoid in high-priority code?

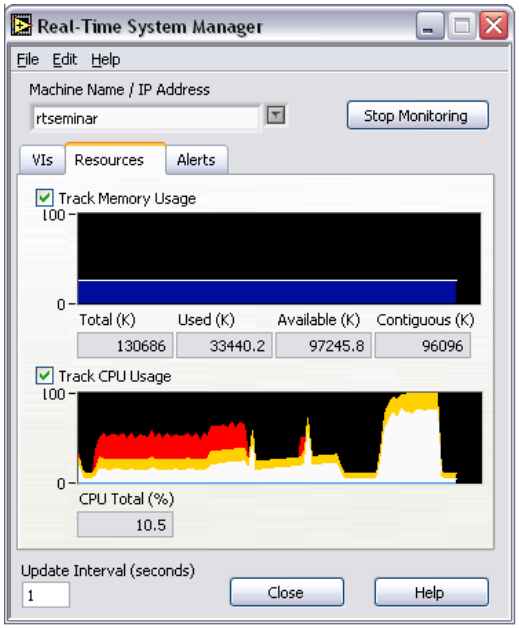
- Operations that allocated memory:
  - Array functions such as Build array, Append array
  - String manipulation
- Non-deterministic functions:
  - File I/O Operations
  - Networking functions
  - Some I/O Driver calls

What to avoid in high-priority code?

- Shared Resources:
  - Global variables
  - Real-Time Module Memory Manager
  - Non-reentrant subVIs
  - Single-threaded DLLs
  - Semaphores*
  - Networking code (TCP/IP, UDP, VI Server)*
  - File I/O*

How to debug RT applications?

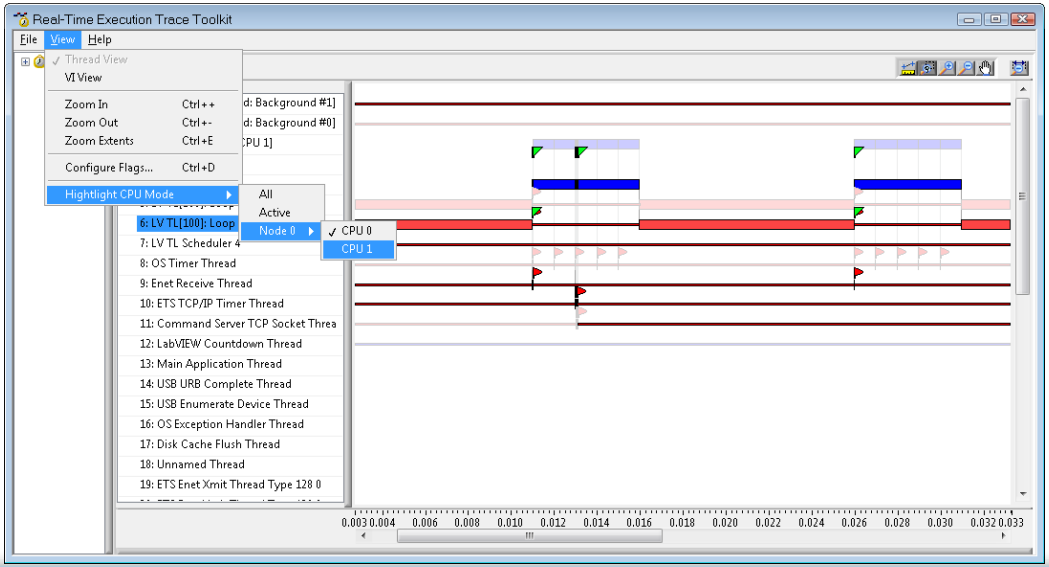
- Standard LabVIEW Debugging features:
  - Breakpoints
  - Highlight Execution
  - Single Stepping
- Benchmarking:
  - VI Profiler
  - Benchmarking project template
- System Monitor
  - Real-Time System Manager



How to debug RT applications?

Execution Trace Toolkit 2.0

- Visually Analyze Real-Time code
- Improvements to version 2.0 include:
  - Faster environment response time
  - Multicore debugging
  - More flags for detecting problems in code



Agenda

- General OS vs. RT OS, terminology
- NI Real-Time Solutions
- Porting from Windows Example
- Where Do I Learn More?

Where Do I Learn More?

- NI Developer Zone
- Web-based presentations on RT topics
- LabVIEW RT
- LabWindows/CVI RT