

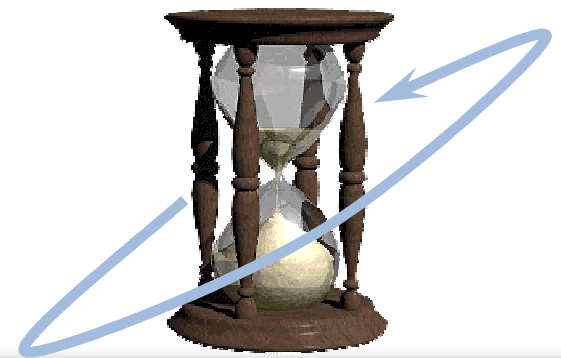
Migrando sus Aplicaciones de Windows a Tiempo Real

Gustavo Valdés

Ingeniero de Mercadotecnia - México

¿Qué es Tiempo Real?

- Tiempo real significa **determinismo**
- Determinismo es la habilidad de completar tareas en una cantidad fija de tiempo
- Tiempo real **no significa** siempre alta velocidad
- Los sistemas de tiempo real tienen restricciones de tiempo que deben ser respetadas para evitar fallas



Gama de Aplicaciones de Tiempo Real

**Desempeño
Determinístico**



Control de Túnel de Viento

**Máxima
Confiabilidad**



Prueba de Confiabilidad

**Operación
Autónoma**



Monitoreo de Seguridad

Seleccionando un Sistema Operativo

SO de Propósito General

- Funciones

- Interfaz de usuario
- Conectividad a sistemas empresariales
- Interrupciones de periféricos
- Aplicaciones silenciosas
- El SO controla toda la calendarización

- Aplicaciones

- Adquisición de datos
- Análisis fuera de línea
- Presentación de datos

SO de Tiempo Real

- Funciones

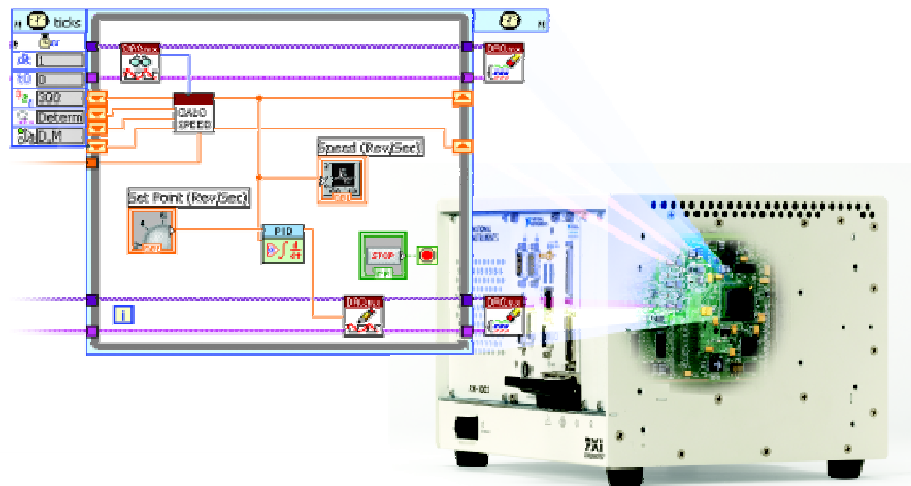
- Embebido
- Control directo sobre el SO
- El calendarizado asegura de ejecutar primero las tareas de alta prioridad

- Aplicaciones

- Control de lazo cerrado
- Decisiones críticas en tiempo
- Operación extendida
- Operación independiente
- Confiabilidad extendida

¿Qué es LabVIEW Real-Time?

- Desarrolle rápidamente aplicaciones determinísticas con programación gráfica
- Diseñe fácilmente sistemas de control y monitoreo distribuido
- Elimine tiempo utilizado en integrar E/S diversas



LabWindows/CVI Real-Time



Cree aplicaciones determinísticas y confiables con ANSI C

- Integración transparente con E/S
- Despliegue con un “click” a objetivos PXI y PC de tiempo real
- Una sola herramienta de desarrollo para tiempo real y Windows
- Reutilice su código existente de C y su experiencia en programación
 - Los mismos “APIs” y hardware que en Windows.

```
while (!RTIsShuttingDown ())  
{  
    DAQmxErrChk (DAQmxReadAnalogScalarF64  
        (aiTaskHandle, 10.0,  
        &processVariable, NULL));  
  
    GetNextPidOutput (processVariable,  
        &output);  
  
    DAQmxErrChk (DAQmxWriteAnalogScalarF64  
        (aoTaskHandle, 1, 10.0, output, NULL));  
}
```



RTOS



Objetivos de Hardware para LabVIEW Real-Time



LabVIEW Real-Time



CompactRIO



PXI



Compact
FieldPoint

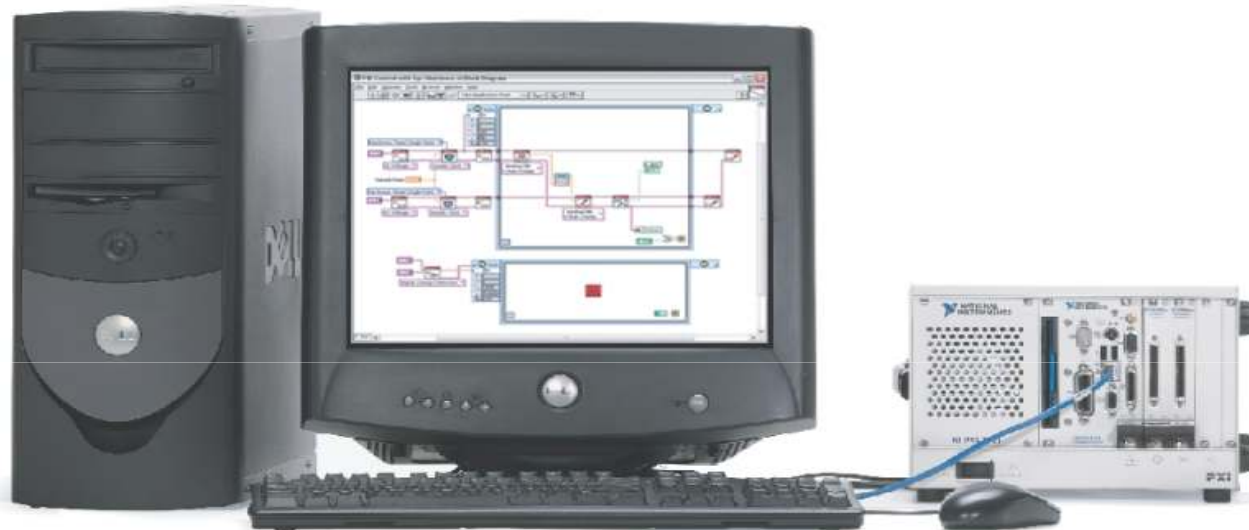


PC Industrial o
de Escritorio



Compact Vision
System

Demo 1: Sistema de LabVIEW Real-Time



Develop on
Host Computer



Execute on
RT Target

¿Cómo Garantiza el Determinismo LabVIEW Real-Time?

- El usuario puede asignar prioridad a código crítico en tiempo
- Se garantiza el ancho de banda del procesador para código crítico en tiempo

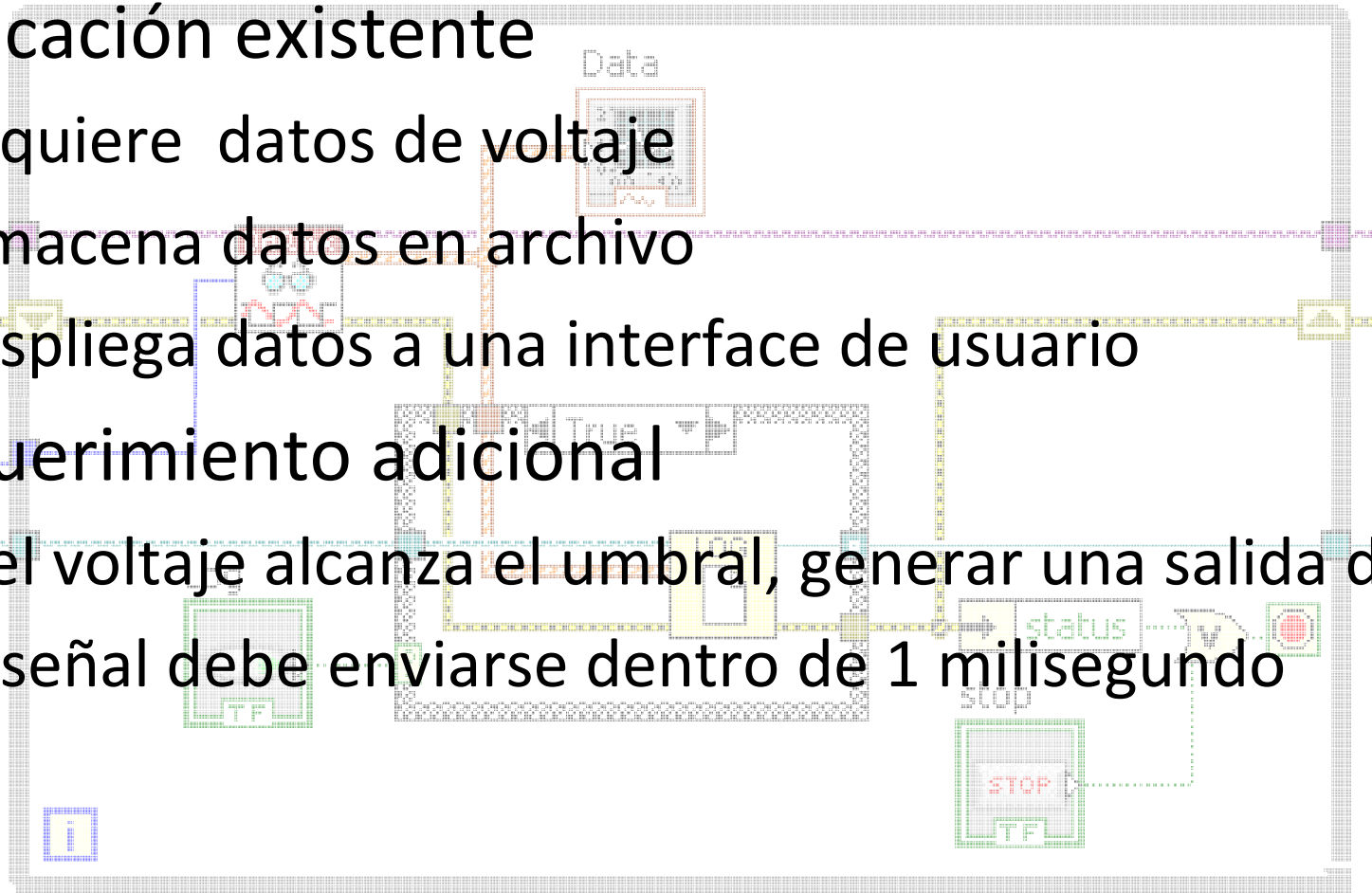
Un gran poder conlleva a una gran responsabilidad – Hombre Araña

- Se debe asegurar que el código crítico en tiempo no consuma todo el ancho de banda del procesador

Demo 2: Determinismo

Ejemplo de Migrar desde Windows

- Aplicación existente
 - Adquiere datos de voltaje
 - Almacena datos en archivo
 - Despliega datos a una interface de usuario
- Requerimiento adicional
 - Si el voltaje alcanza el umbral, generar una salida digital
 - La señal debe enviarse dentro de 1 milisegundo



Cuando se Migra a LabVIEW Real-Time...

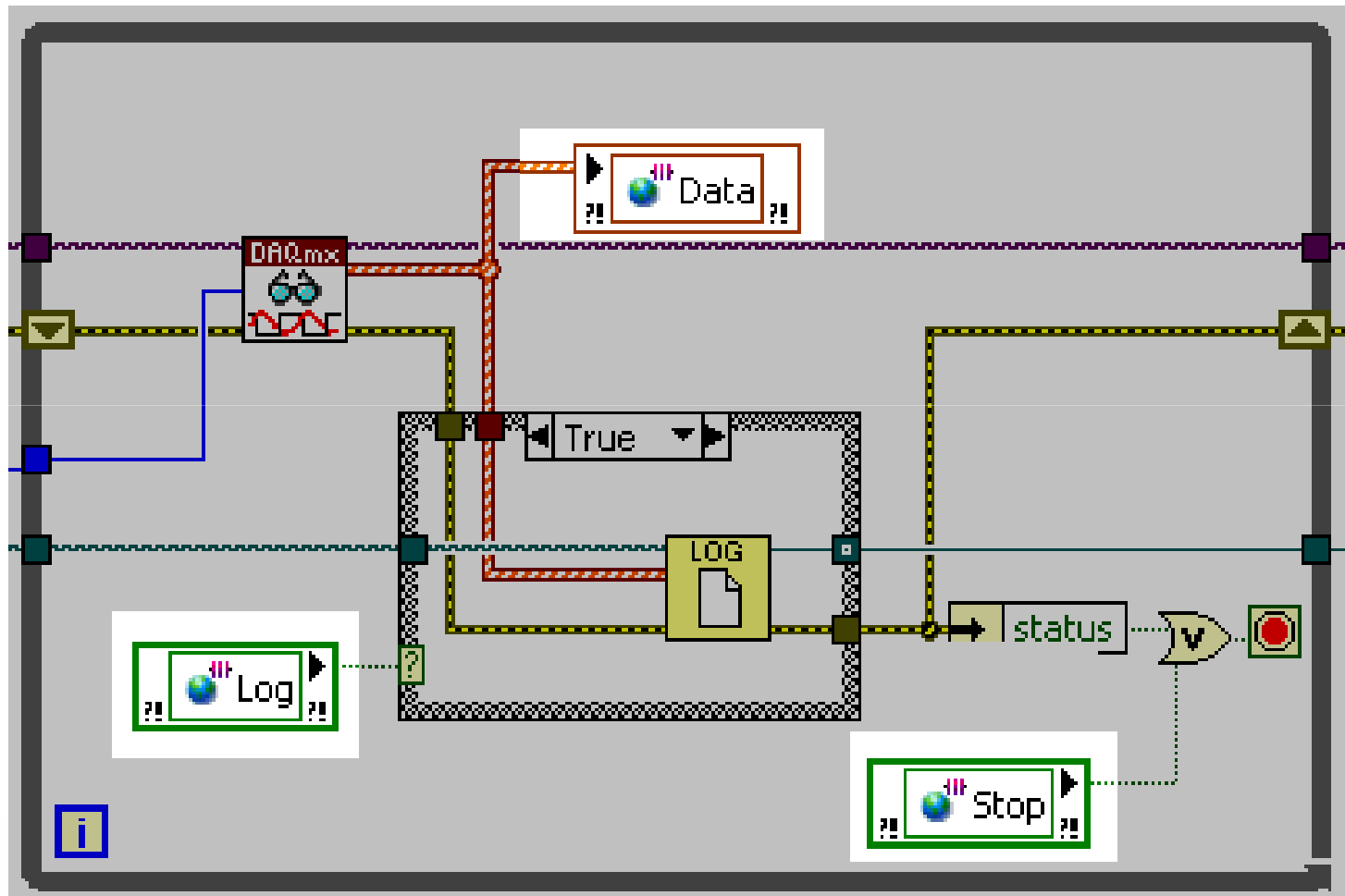
1. *¿Qué hago con la interfaz de usuario?*
2. *¿Cómo me comunico con otros sistemas?*
3. *¿Cómo logro el determinismo?*
4. *¿Qué debo evitar?*
5. *¿Cómo depuro aplicaciones de tiempo real?*

Evitar Elementos de la UI en el Código Crítico en el Tiempo



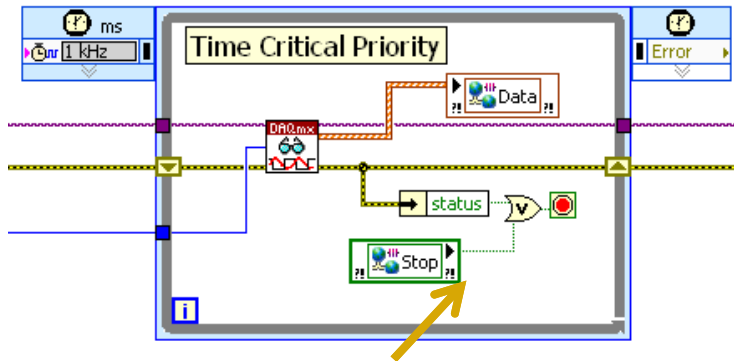
¿Qué Hago con la Interfaz de Usuario (IU)?

Reemplazar los Controles e Indicadores con “RTFIFOs”

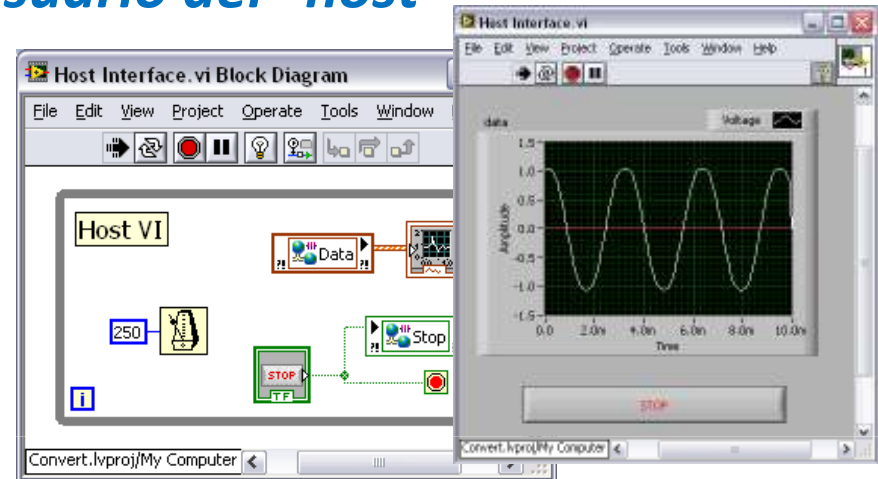


¿Qué Hago con la Interfaz de Usuario (IU)?

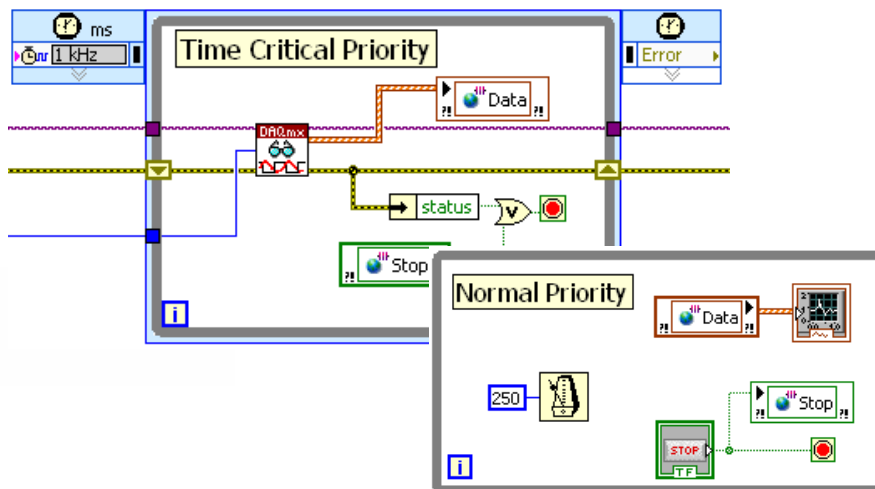
Transferir Datos a la Interfaz de Usuario del “host”



Variables publicadas en red



Utilizar Ciclo de Baja Prioridad para IU en Web



¿Cómo Me Comunico con Otros Sistemas?

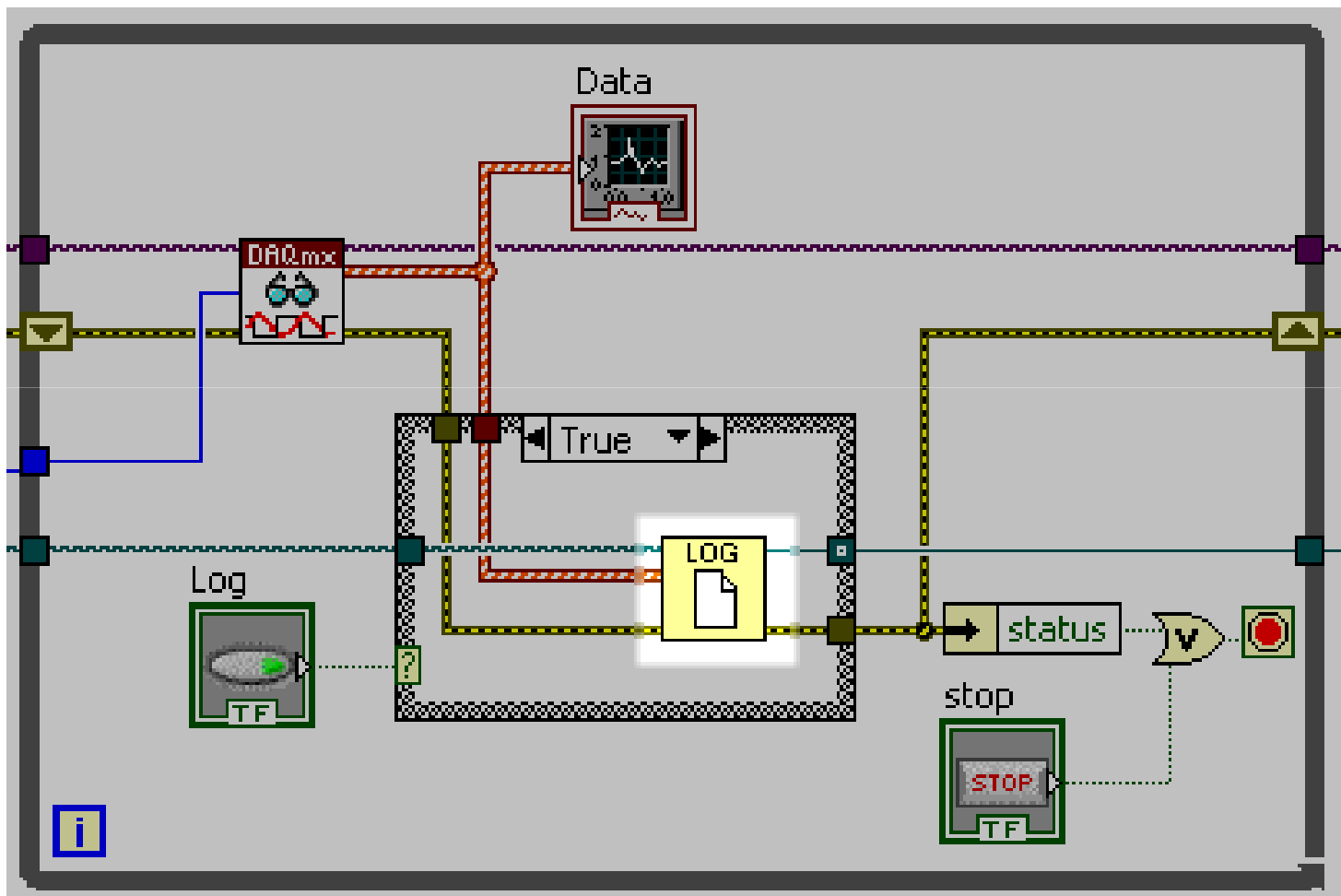
Utilice Funciones de Red o Buses de E/S

- Transferencia de Datos por Red
 - Variable Compartida de LabVIEW (Sólo Punto)
 - TCP/UDP (Personalizada)
- Control Remoto de Aplicación
 - VI Server
 - Paneles Remotos
- Buses de E/S
 - Serial, GPIB, USB, IEEE 1394
 - CAN, DeviceNET, Modbus, etc.
 - Memoria Reflectiva



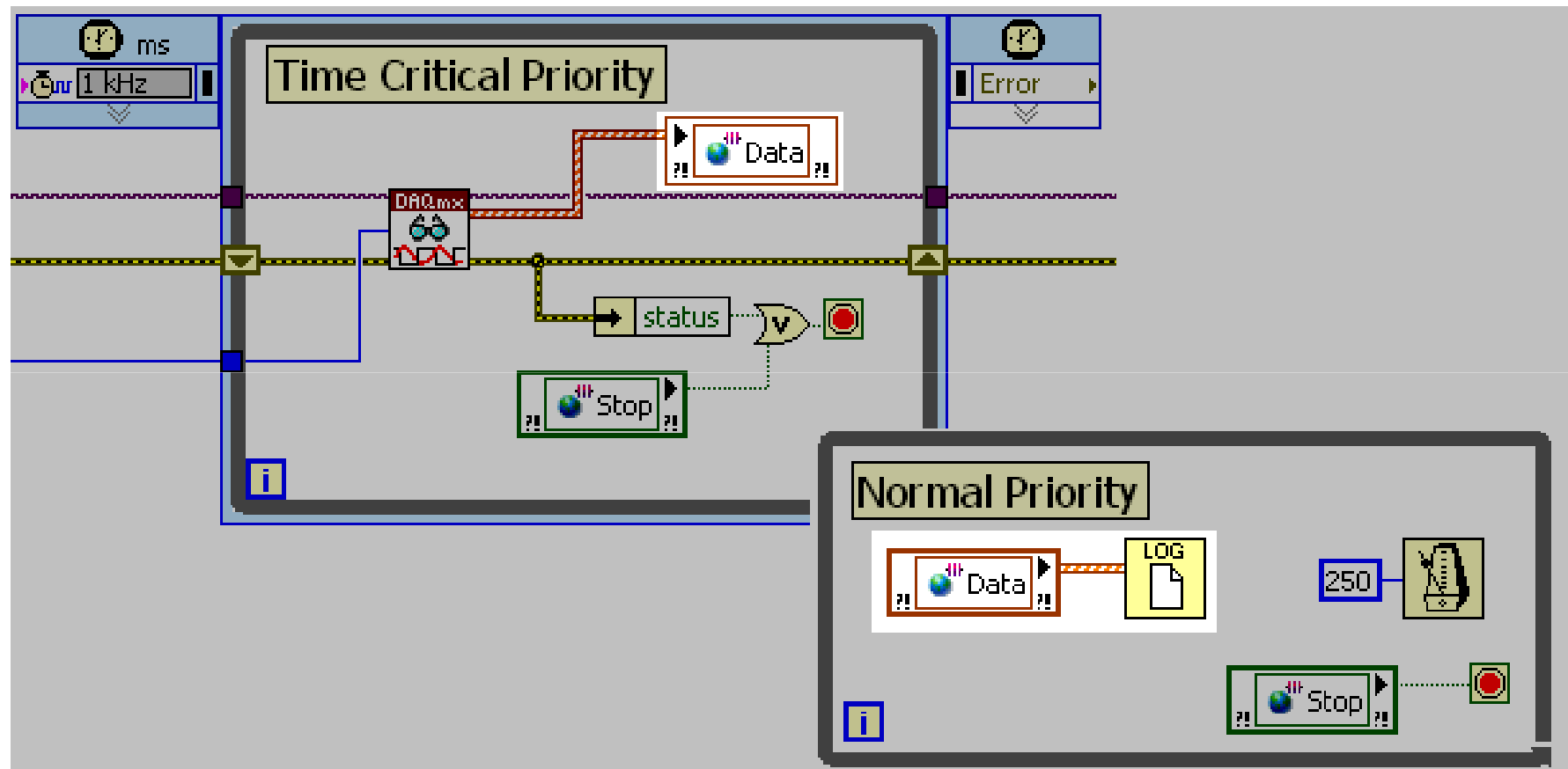
¿Cómo Logro el Determinismo?

Mueva funciones no determinísticas a ciclos de baja prioridad



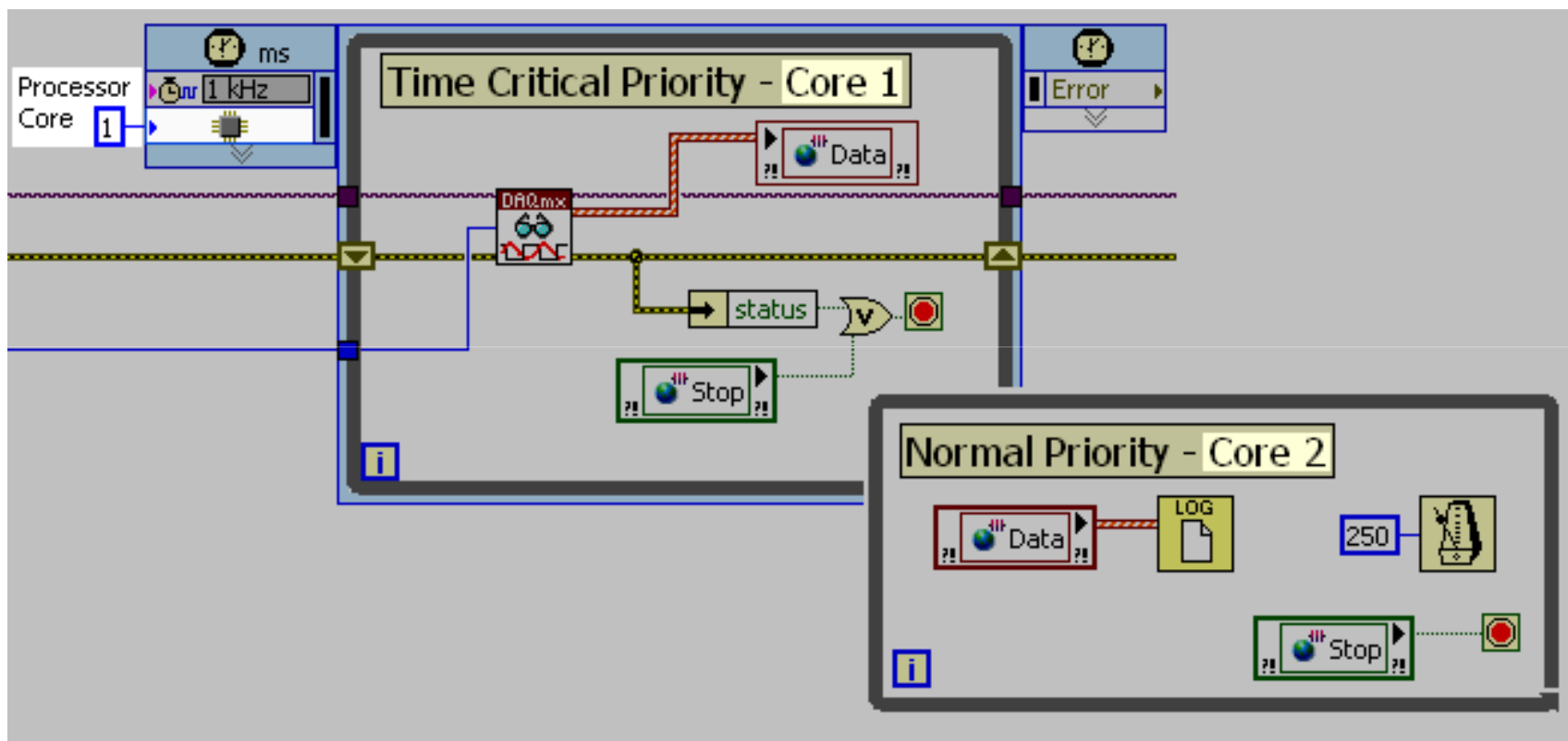
¿Cómo Logro el Determinismo?

Mueva funciones no determinísticas a ciclos de baja prioridad



¿Cómo Logro el Determinismo?

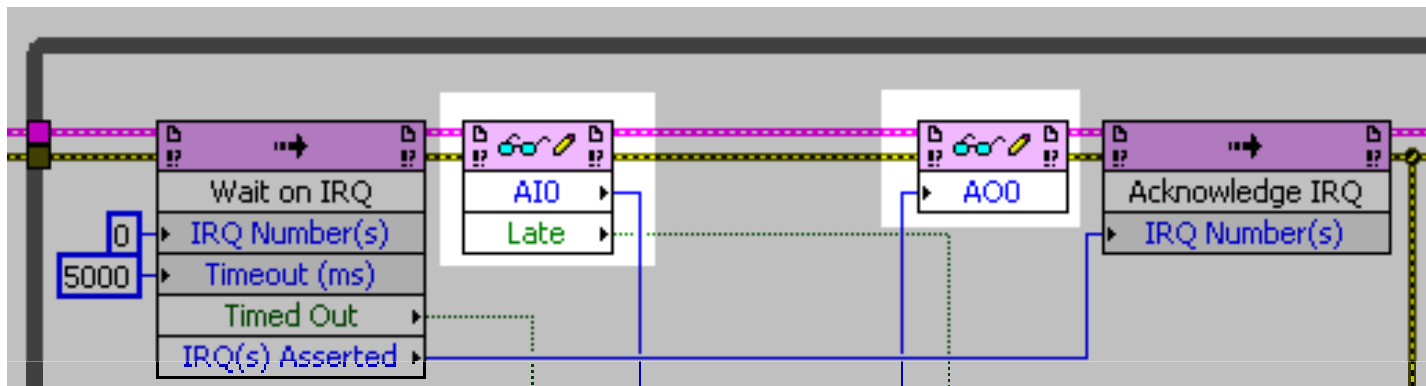
Asigne el código crítico a un procesador dedicado (PXI/PC)



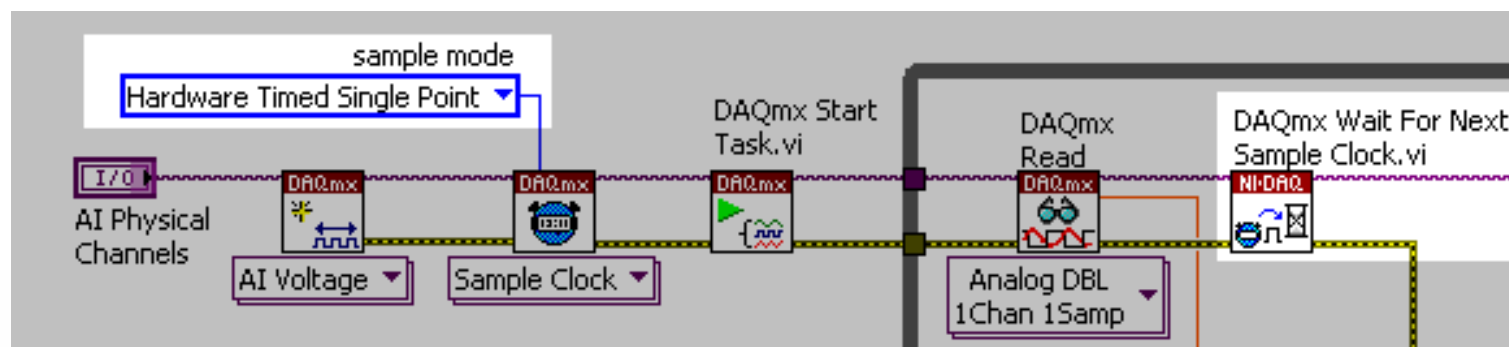
¿Cómo Logro el Determinismo?

Aprenda optimizaciones de E/S de tiempo real

NI-RIO



NI-DAQmx



Demo: Arquitectura de LabVIEW Real-Time

¿Qué Debo Evitar en Código Crítico en Tiempo?

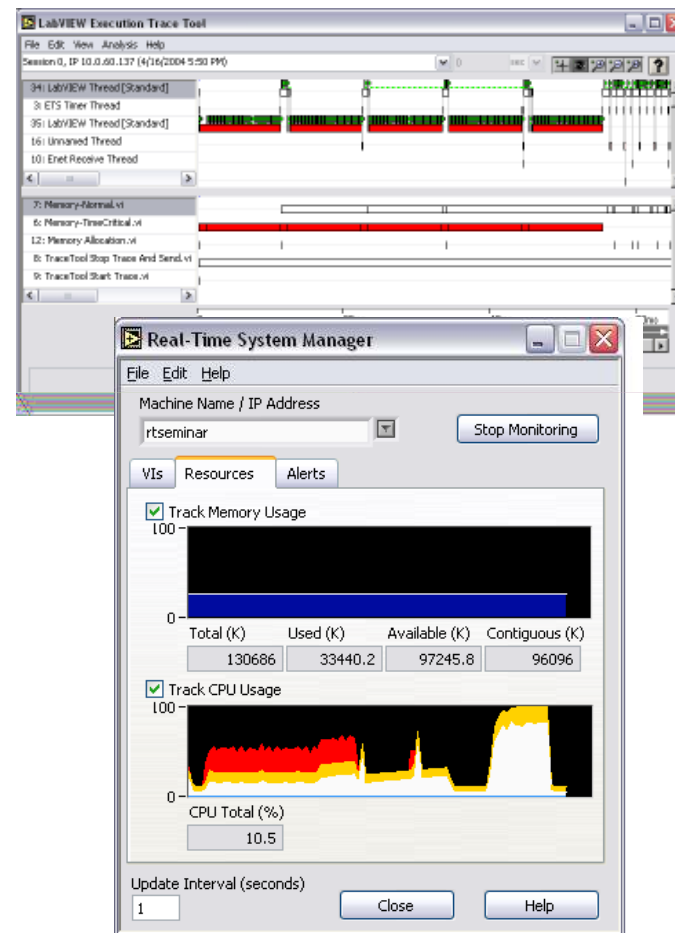
Asignación de memoria y recursos compartidos

- Cualquier operación que asigne memoria
 - Construir arreglo, anexar a arreglos, etc.
 - Manipulación de cadenas de caracteres (*strings*)
- Llamadas no determinísticas
 - E/S de Archivo
 - Funciones de red
 - Algunas llamadas de manejadores de E/S
- Recursos compartidos
 - SubVIs no reentrantes
 - Recursos de E/S

¿Cómo Depuro Aplicaciones de Tiempo Real?

Utilice depuración estándar, comparaciones, y rastreo

- Depuración estándar de LabVIEW
 - Ejecución pausada, paso a paso.
- Comparaciones
 - “VI Profiler”
 - “Benchmarking Project Template”
- Optimización
 - LabVIEW Execution Trace Toolkit
- Monitor del Sistema
 - Real-Time System Manager
- Aplicación Ejecutándose
 - VI de “RT Debug String (printf)”



Demo: Depurando Aplicaciones de Tiempo Real

¿Dónde Aprendo Más?

- NI Developer Zone
- Seminario Práctico de LabVIEW Real-Time
- Curso de Desarrollo de Aplicaciones de Tiempo Real

