


Arquiteturas comprovadas para o projeto de máquinas inteligentes

Nome do Apresentador
Cargo

Agenda

- Máquinas inteligentes (*Smart Machines*)
- Principais tecnologias
- Arquiteturas comprovadas
 - Controle com FPGA
 - Sequenciador RT
 - Integração de Motion e Vision

O mundo da construção de máquinas



Complexidade
e flexibilidade

Economia e
qualidade

- Produtos altamente integrados
- Menor tamanho de lote
- Variações específicas do cliente
- Menores ciclos de vida dos produtos

- Maior rendimento
- Maior qualidade
- Maior eficiência energética
- Maior grau de automação

Máquinas inteligentes

Operação autônoma

- Equipamento de produção modular
- Alto grau de flexibilidade
- Percepção do ambiente

Prevenir e corrigir erros no processo

- Habilidades de auto-análise e auto-reparação
- Modificações em tempo real nos planos do processo

Aprender e antecipar

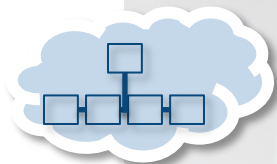
- Controle baseado em modelos, controle adaptativo
- Simulação

Interação com outras máquinas e sistemas

- Sistemas interconectados – fábrica inteligente
- Estruturas de dados comumente compartilhadas

As máquinas de produção modernas

Integração com a
planta, SCADA



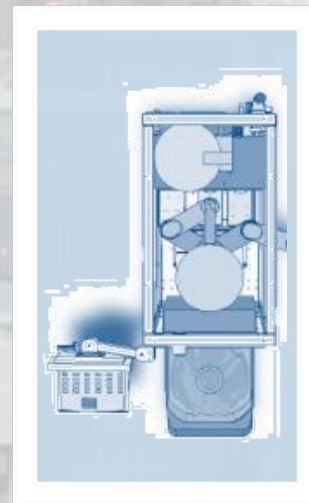
Cabeça de
usinagem
inteligente



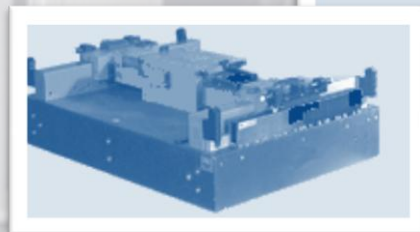
Controlador para
movimentação customizada



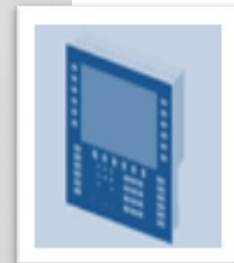
Manipulador



Sistemas de
segurança

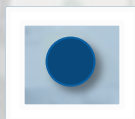


Controlador de
movimento multieixo



IHM local

Monitoramento de
condição de
máquina (MCM)



Transferência de
posição para posição

Lidando com o crescente nível de complexidade

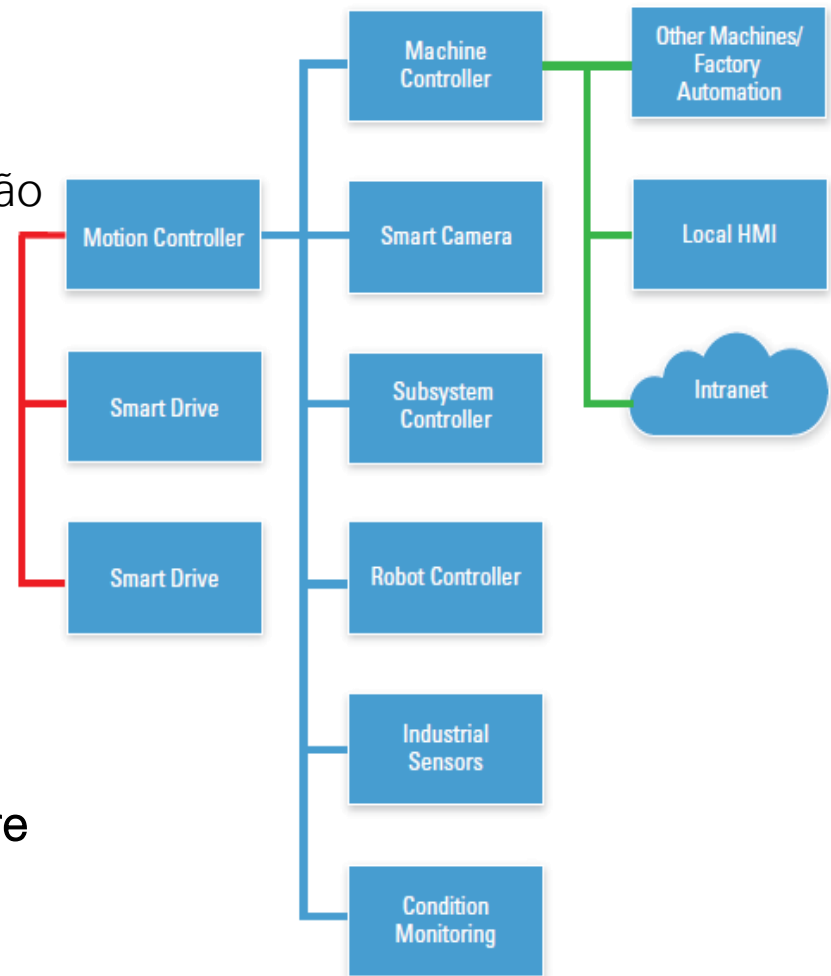
Sistemas embarcados

- Computação heterogênea
- Integra diferentes tarefas de automação
- E/S modulares
- Customização de baixo custo

Comunicação

- Comunicação de tempo crítico e não crítico
- Protocolos industriais
- Integração com TI
- Computação na nuvem

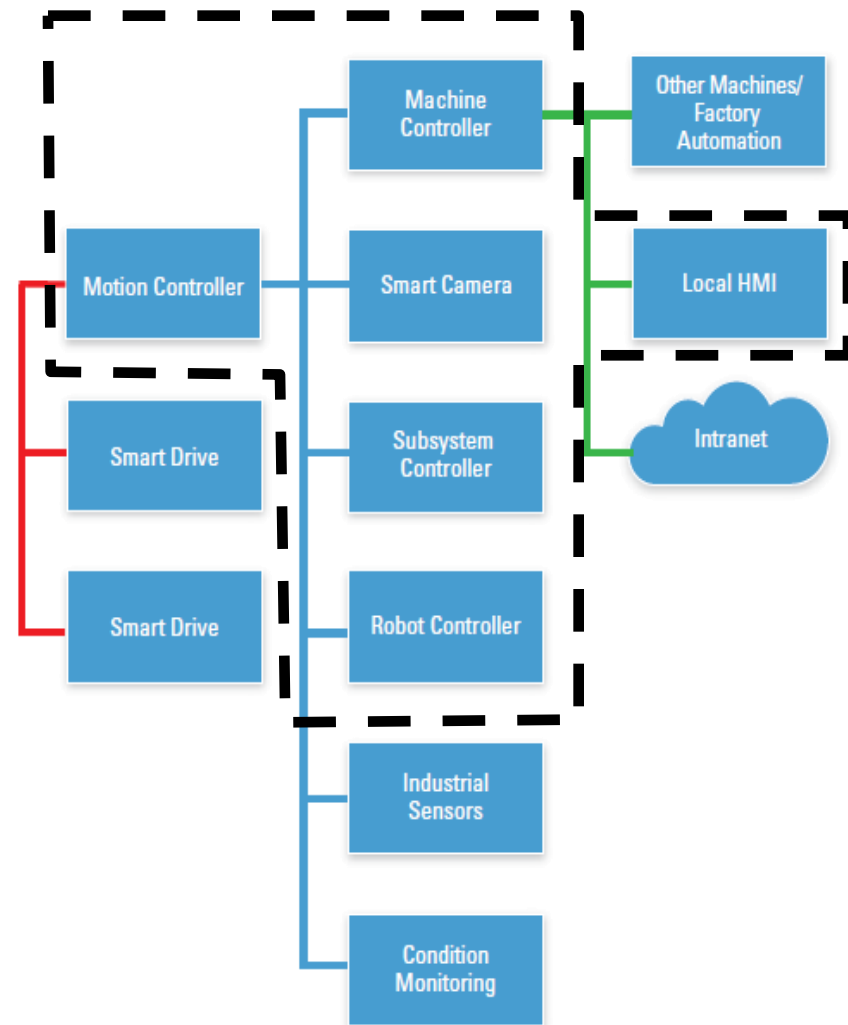
Abordagem de projeto centrado em software



Lidando com o crescente nível de complexidade

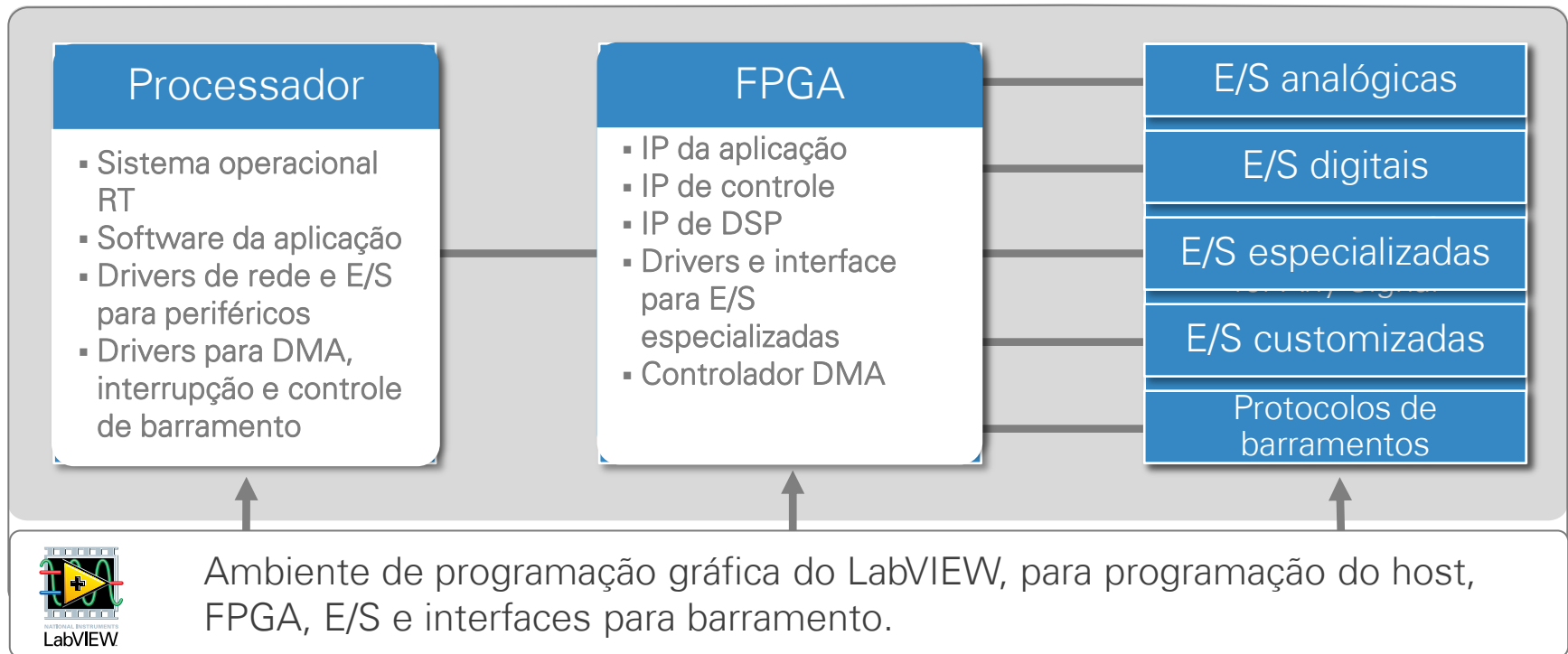
Consolidação

Abstração de software



Computação heterogênea

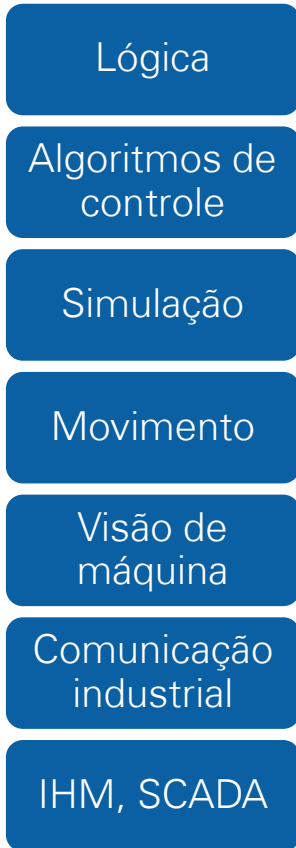
Arquitetura RIO do LabVIEW



A tecnologia da NI para máquinas inteligentes e produção inteligente



Projeto de máquinas centrado em software



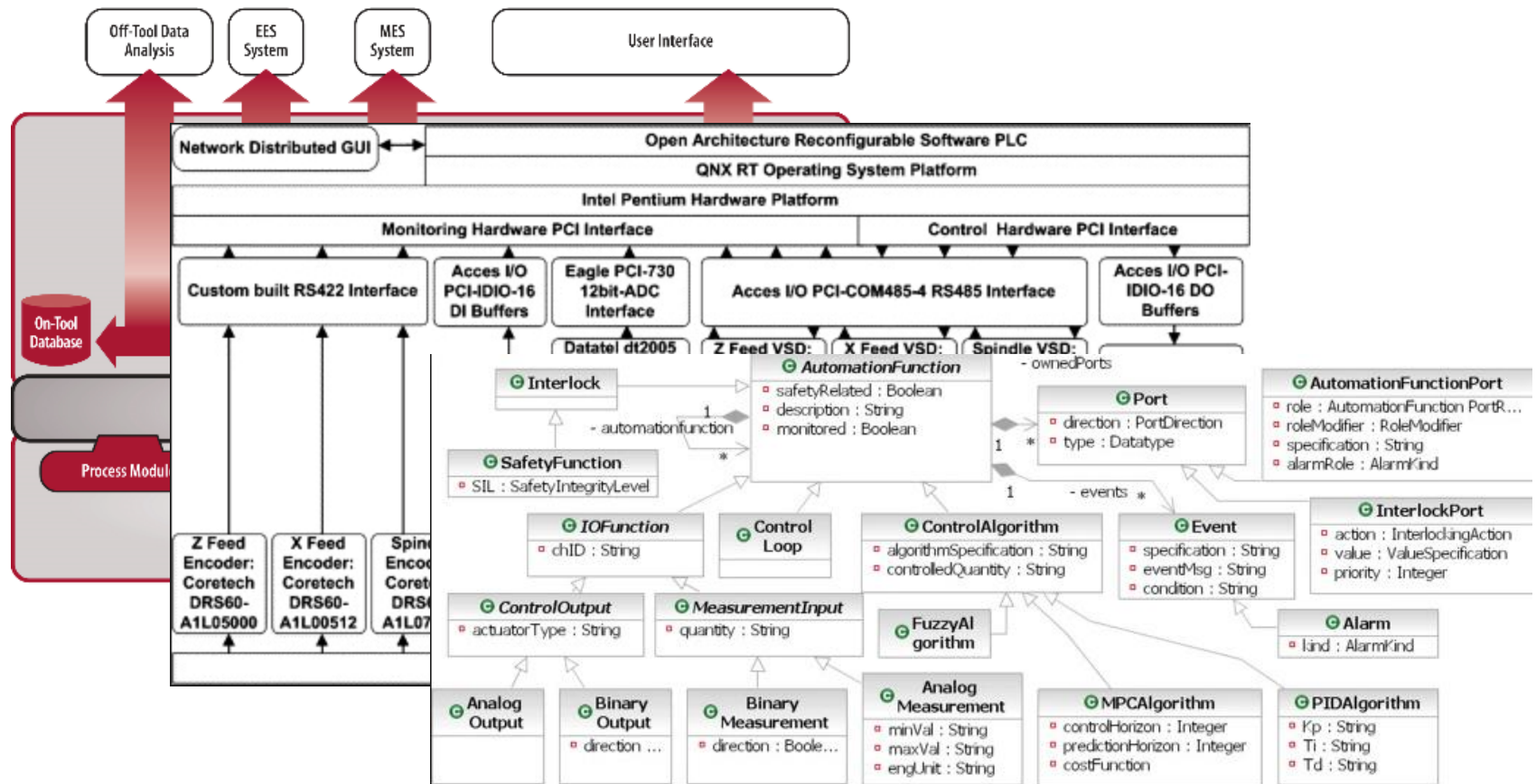
Módulos de Add-on do LabVIEW

- Real-Time
- FPGA
- Módulo MathScript RT
- Módulo Control Design & Simulation
- NI SoftMotion
- Módulo Vision Development
- Módulo LV DSC

Toolkits e ferramentas de terceiros para LabVIEW

- PID and Fuzzy Logic Toolkit
- Signal Processing Toolkit
- Sound & Vibration Toolkit
- Adaptive Filter Toolkit
- Database Connectivity Toolkit
- Mais....
- Expert Control – ferramenta para parametrização do controlador
- Bibliotecas Digimetrix Robotics
- Mais....

Esses sistemas são complexos!



Funcionalidades principais

Operação determinística, sem inteligência

- Execução determinística
- Sincronização do sistema sem inteligência
- Comunicação com os sistemas host (SCADA, IHM, CLP)

Varredura de E/S

- Varredura de E/S
- Controle de alarme
- Data logging

Sequenciamento de tarefas

- Máquina de estados/ lógica de estados / sequenciadores
- Transições/ triggers / gerenciamento de eventos

Controle de movimento

- Controle supervisão
- Geração de trajetória
- Controle dos motores

Projeto para uma operação segura

- Watchdogs
- Estado de segurança
- Arquiteturas de redundância

Integração com outros componentes de automação

- Comunicação industrial

Interface para E/S e processamento de sinais

- Processamento de imagem
- Data logging / aquisição e análise de formas de onda

Funcionalidades principais

Operação determinística, sem inteligência

- **Execução determinística**
- Sincronização do sistema sem inteligência
- **Comunicação com os sistemas de host (SCADA, IHM, CLP)**

Varredura de E/S

- Varredura de E/S
- Controle de alarme
- Data logging

Sequenciamento de tarefas

- **Máquina de estados/ lógica de estados / sequenciadores**
- Transições/ triggers / gerenciamento de eventos

Controle de movimento

- Controle supervisão
- Geração de trajetória
- Controle dos motores

Projeto para uma operação segura

- **Watchdogs**
- **Estado de segurança**
- Arquiteturas de redundância

Integração com outros componentes de automação

- Comunicação industrial

Interface para E/S e processamento de sinais

- Processamento de imagem
- Data logging / aquisição e análise de formas de onda

Funcionalidades principais

Operação determinística, sem inteligência

- Execução determinística
- Sincronização do sistema sem inteligência
- Comunicação com os sistemas de host (SCADA, IHM, CLP)

Varredura de E/S

- Varredura de E/S
- Controle de alarme
- Data logging

Sequenciamento de tarefas

- Máquina de estados/ lógica de estados / sequenciadores
- Transições/ triggers / gerenciamento de eventos

Controle de movimento

- Controle supervisão
- Geração de trajetória
- Controle dos motores

Projeto para uma operação segura

- Watchdogs
- Estado de segurança
- Arquiteturas de redundância

Integração com outros componentes de automação

- Comunicação industrial

Interface para E/S e processamento de sinais

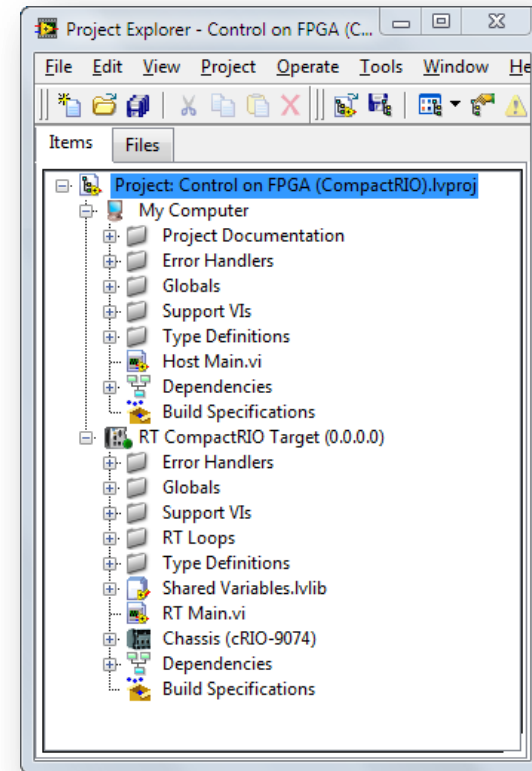
- Processamento de imagem
- Data logging / aquisição e análise de formas de onda

Considerações do projeto

- Escalabilidade
- Flexibilidade
- Manutenção
- Reuso
- Desempenho
- Esforços para o desenvolvimento

O que são os exemplos de projeto do LabVIEW?

- Pontos de partida para sua aplicação, que demonstram o uso de uma estrutura escalável
- Construídos a partir dos templates de projeto do LabVIEW
- Benefícios
 - Gaste menos tempo com desenvolvimento, começando a partir de uma aplicação pré-construída
 - Garanta que seu projeto é confiável, começando a partir de arquiteturas comprovadas e recomendadas pela NI

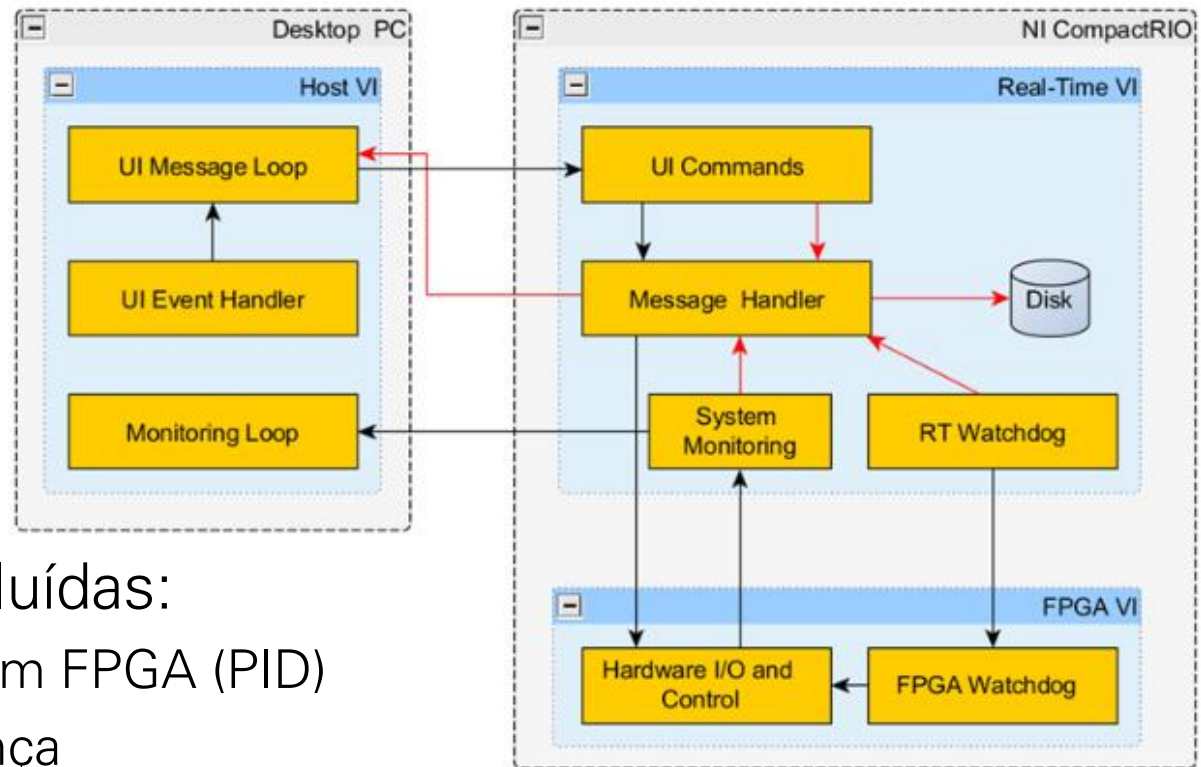


Exemplos de projeto do LabVIEW

- Templates para controle
 - LabVIEW FPGA Control
 - LabVIEW Real-Time Control
 - LabVIEW Real-Time Sequencer
- Templates para aquisição de dados
 - LabVIEW FPGA Waveform Acquisition and Data Logging
 - LabVIEW Real-Time Control (NI-DAQmx)
 - LabVIEW Real-Time Waveform Acquisition and Logging (NI-DAQmx)
- Templates para SCADA
 - LabVIEW SCADA with NI Datalogging and Supervisory Control (DSC)



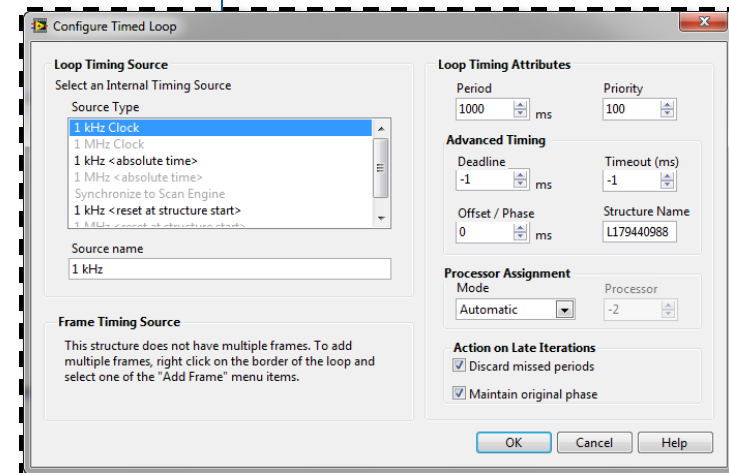
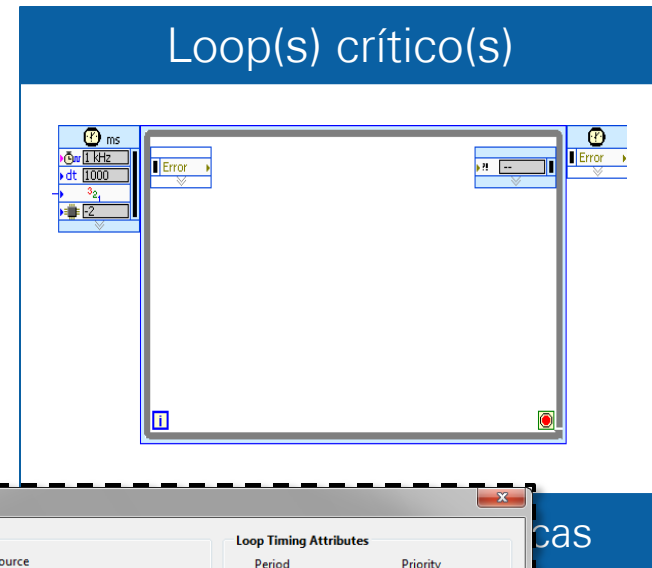
Exemplo de projeto: FPGA Control on CompactRIO



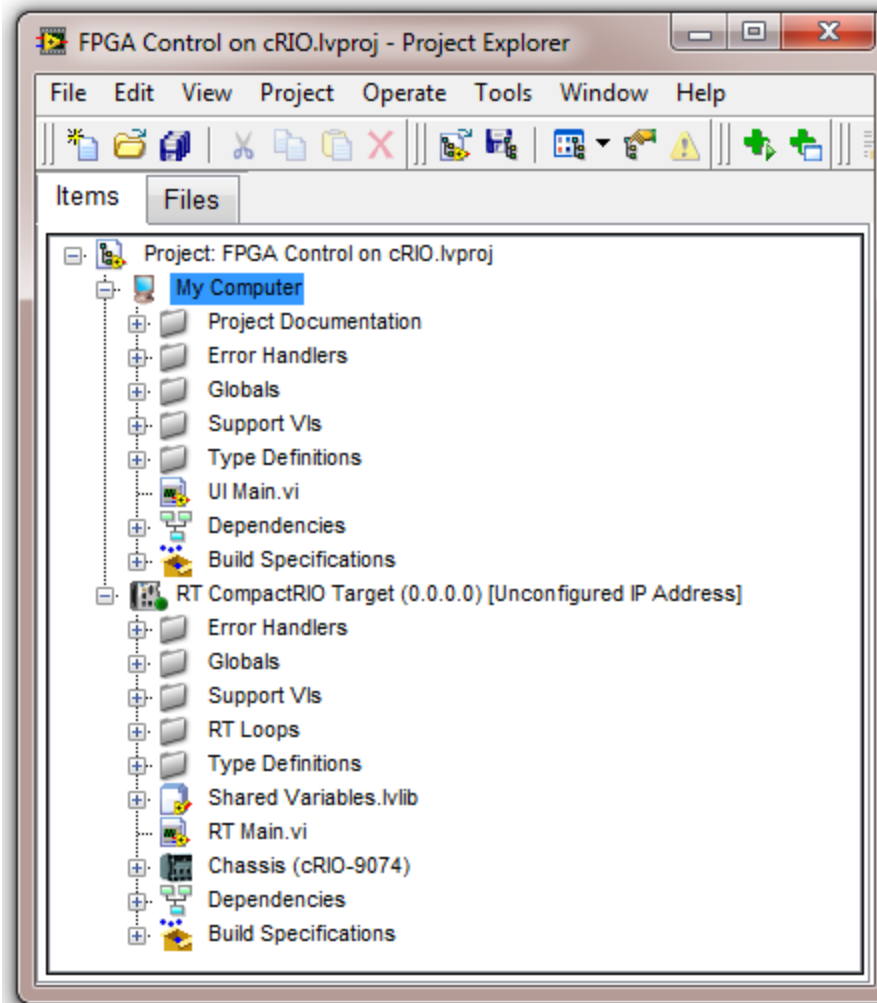
- Funcionalidades incluídas:
 - Controle baseado em FPGA (PID)
 - Estados de segurança
 - Operação sem inteligência
 - Interface de comunicação RT para IU
 - IU remota para visualização e mudanças na configuração

Execução determinística

- Temporização precisa
- Maior confiabilidade
- Priorização
- Sistema operacional RT versus FPGA



Explicação: FPGA Control on CompactRIO

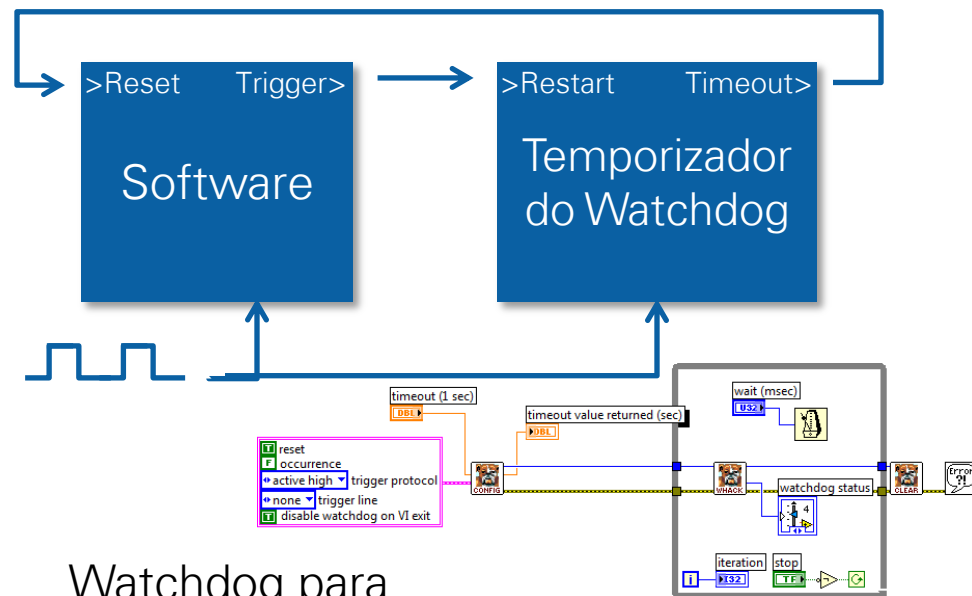


Temporização Watchdog

- Utilize a funcionalidade do hardware para verificar a execução do programa
- Recupera-se com segurança de quedas no programa



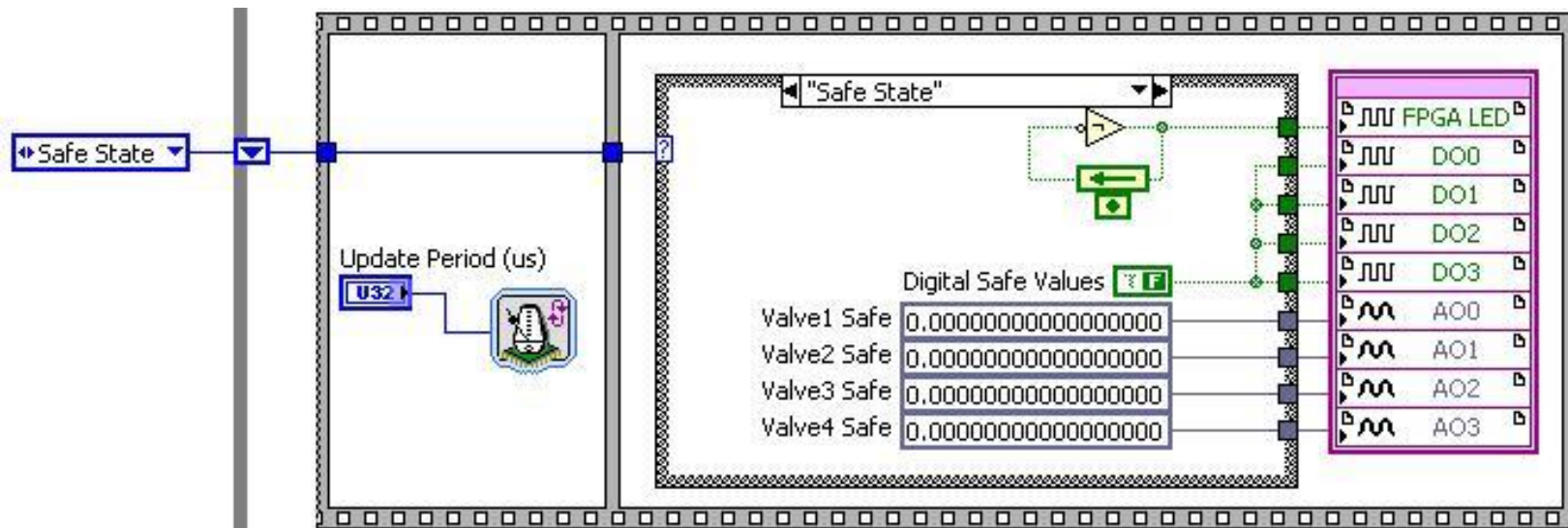
Watchdog para controlador humano



Watchdog para controlador de software

Estados de segurança

- Define os estados das saídas durante o processo de boot e inicialização
- Define os estados das saídas em caso de falha
- Referência de projeto para controle à prova de falhas para cRIO

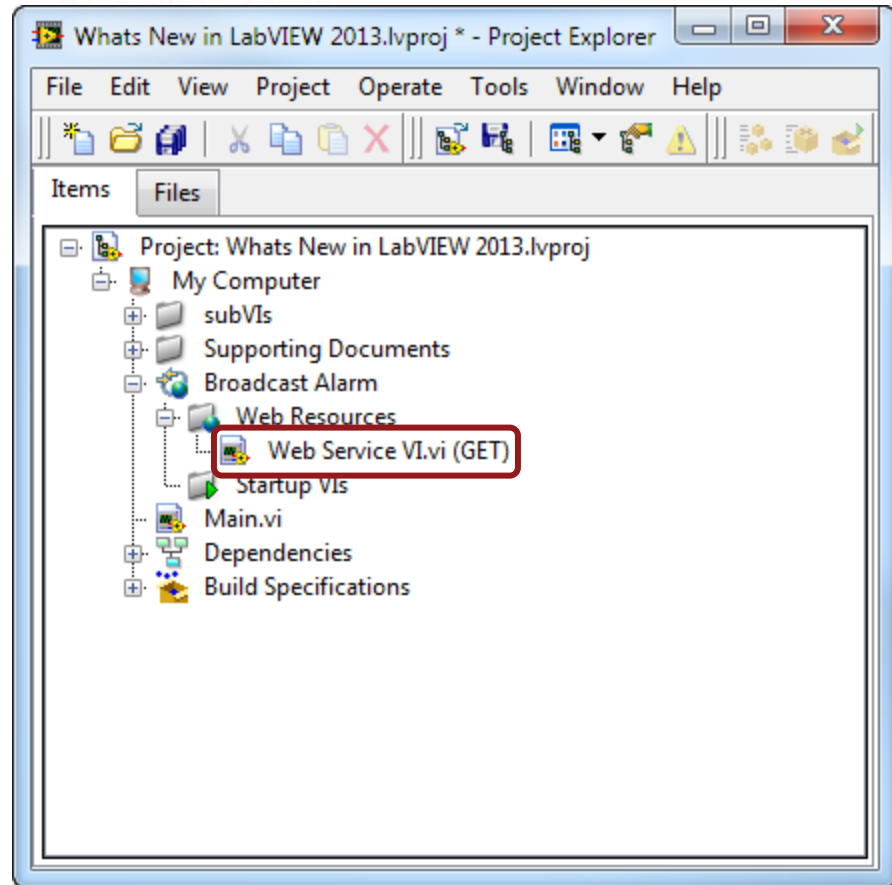


Comunicação com sistemas de host

- Há muitas opções diferentes para comunicação
 - Determinística versus não determinística
 - Mestre-escravo versus publicar-subscriver
 - Ponto a ponto versus bufferizada
 - Web services

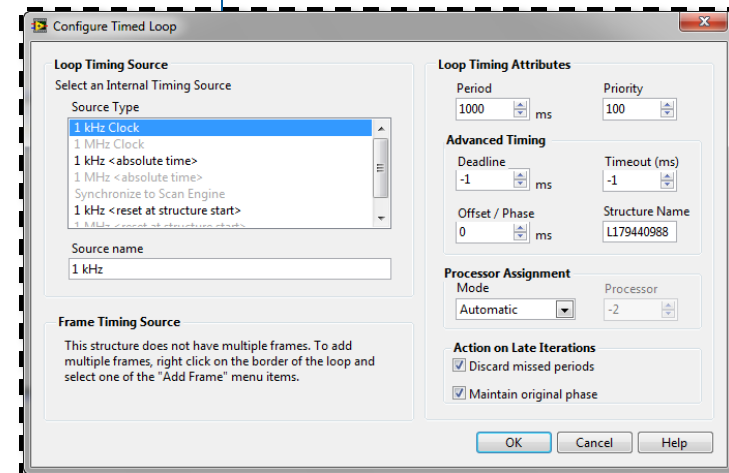
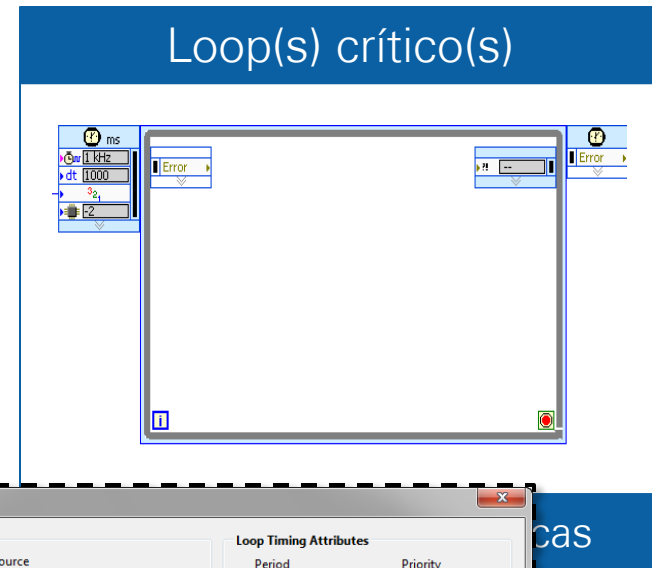
Nova experiência com Web Service

- Fornece acesso remoto às aplicações do LabVIEW
- Antigamente era um item pertencente às *Build Specifications*
- Agora é um item de projeto – edição e implementação mais rápidas

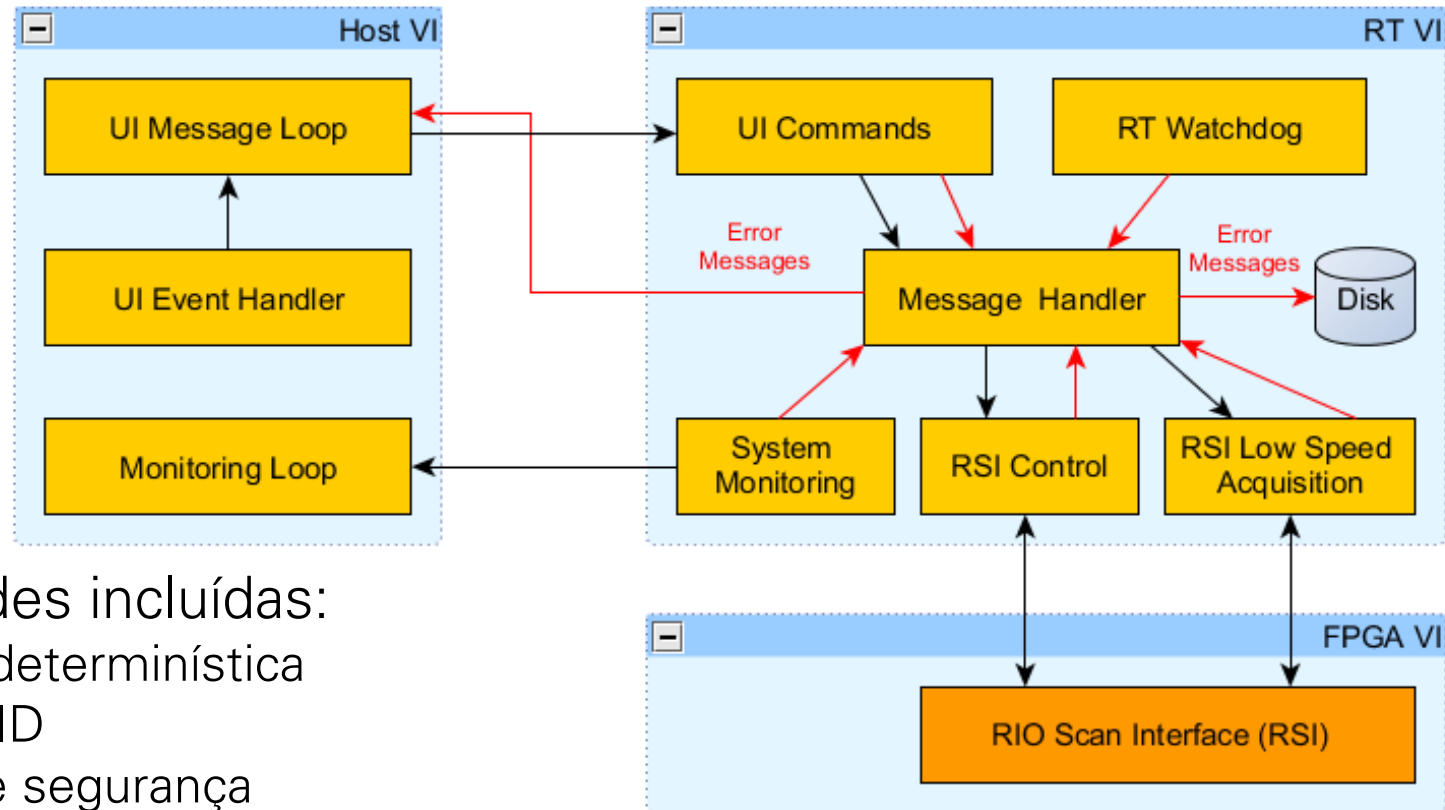


Execução determinística

- Temporização precisa
- Maior confiabilidade
- Priorização
- Sistema operacional RT versus FPGA

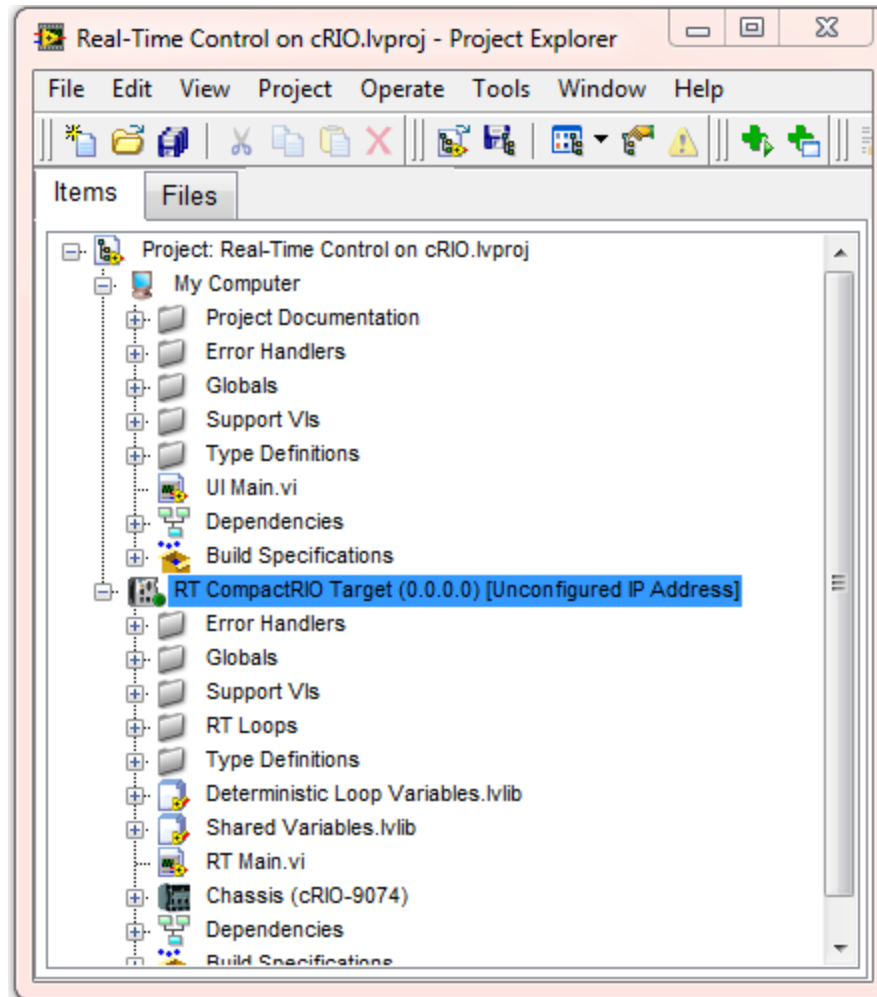


Exemplo de projeto: LabVIEW RT Control on cRIO



- Funcionalidades incluídas:
 - Execução determinística
 - Controle PID
 - Estados de segurança
 - Gerenciamento de erros
 - Operação sem inteligência
 - Interface de comunicação de RT para IU
 - IU remota para mudanças de visualização e configuração

Explicação: LabVIEW RT Control on cRIO

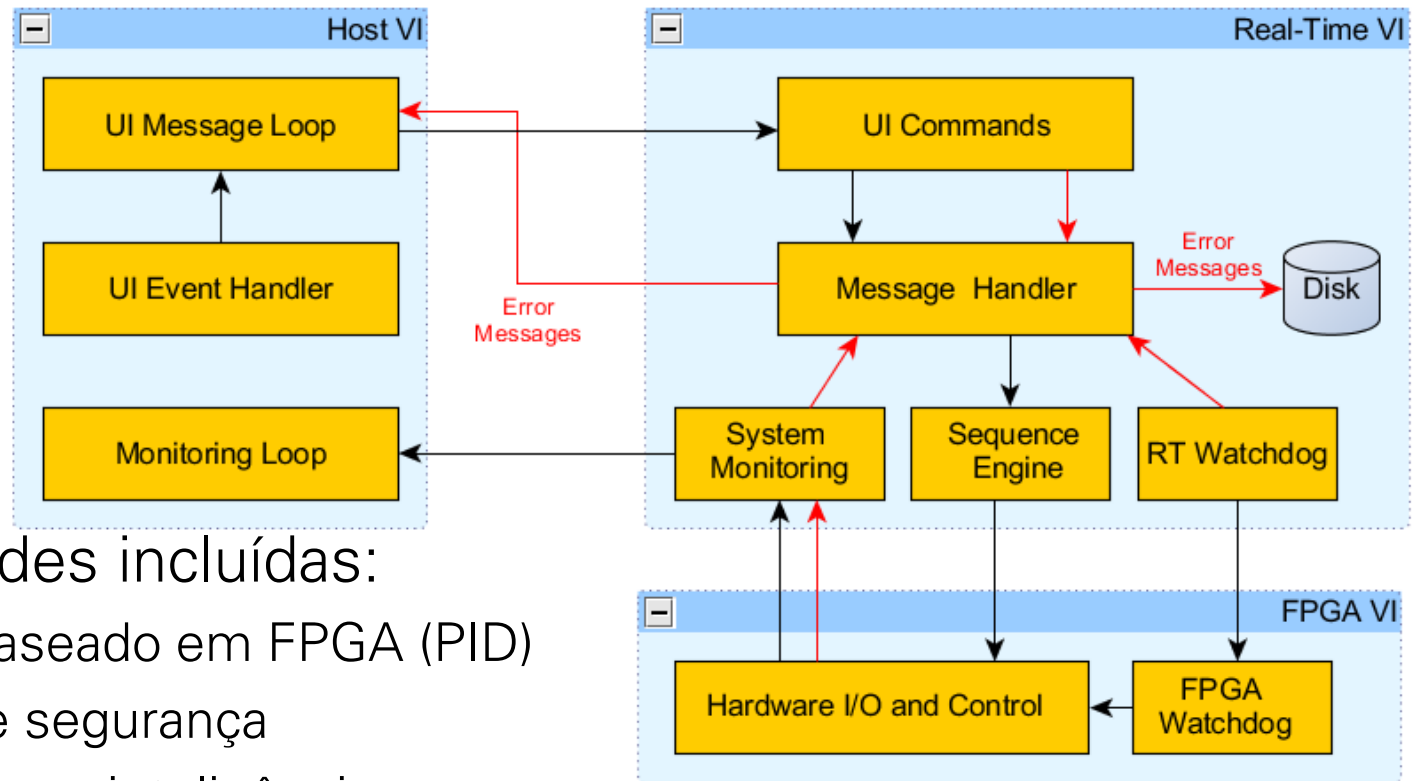


Máquinas de estados/ lógicas de estados / sequenciadores

- As máquinas de estados são amplamente utilizadas para controlar, sequenciar e coordenar o comportamento de outros subsistemas digitais
- Diagramas de estados são traduzidos facilmente para o LabVIEW
- Há muitas maneiras de implementar máquinas de estados



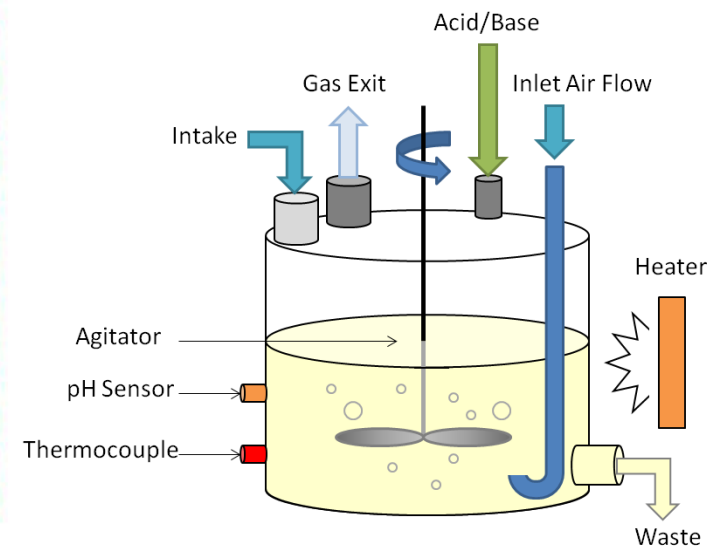
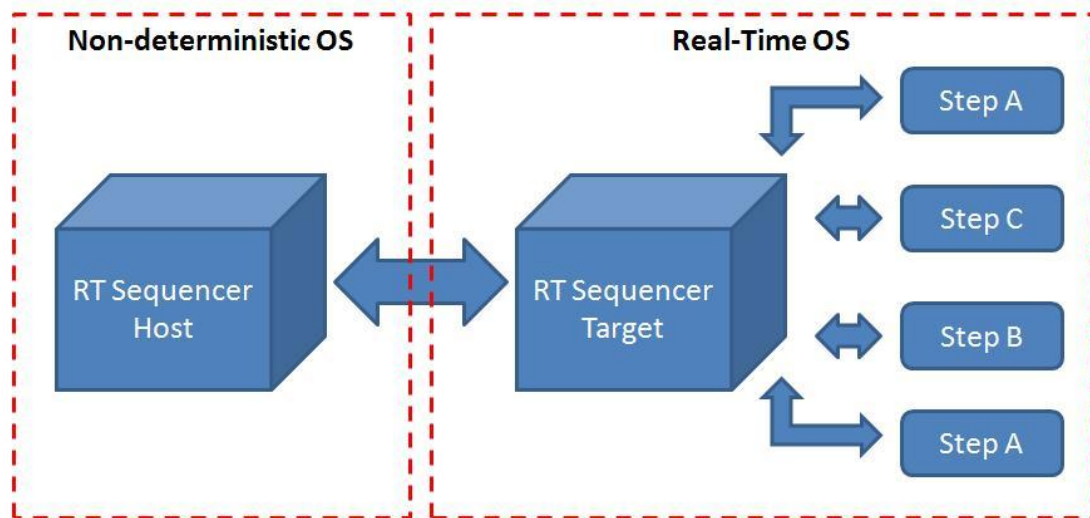
Motivos para utilizar o exemplo de projeto Real-Time Sequencer

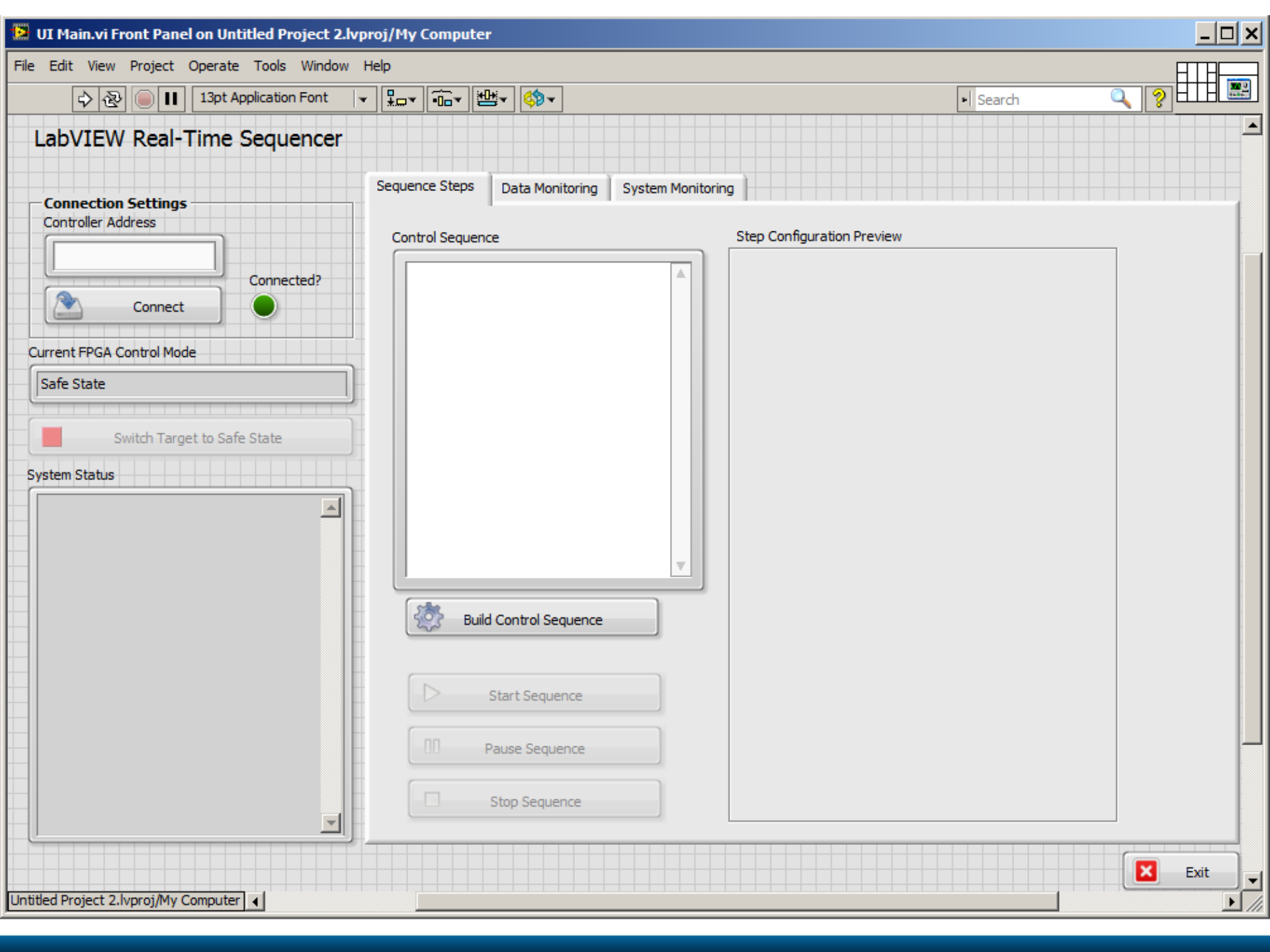


- Funcionalidades incluídas:
 - Controle baseado em FPGA (PID)
 - Estados de segurança
 - Operação sem inteligência
 - Interface de comunicação de RT para IU
 - IU remota para mudanças de visualização e configuração
 - **Sequenciador Real-Time**

O exemplo de projeto Real-Time Sequencer do LabVIEW 2013

- Disponível no LabVIEW 2013
- A IU permite que o usuário defina e sequeencie as etapas
- As etapas são executadas em um mecanismo de sequenciamento RT
- O FPGA é utilizado para E/S de hardware e controle





Connection Settings

Controller Address



Connect

Connected?



Current FPGA Control Mode

Safe State



Switch Target to Safe State

System Status

Sequence Steps

Data Monitoring

System Monitoring

Control Sequence

Step Configuration Preview



Build Control Sequence



Start Sequence



Pause Sequence



Stop Sequence



Exit

The Real-Time VI contains five parallel loops:

UI Command Loop - Receives messages from the UI Main VI.

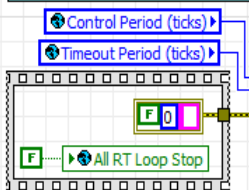
Message Handling Loop - Handles all messages coming from the UI Command Loop, itself, the Watchdog Loop, and the Monitoring Loop.

Sequence Engine Loop - Executes the sequence sent from the UI Main VI.

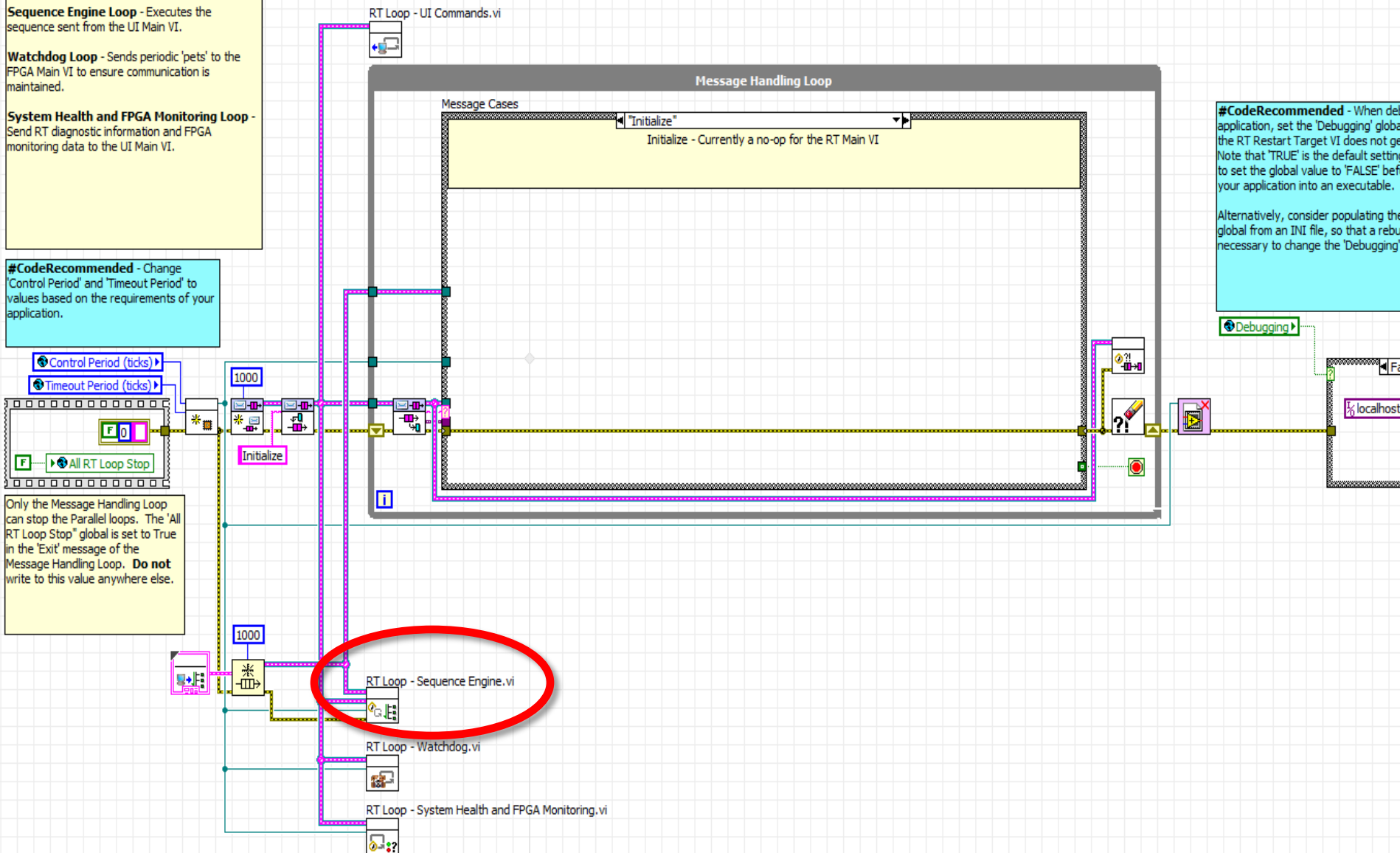
Watchdog Loop - Sends periodic 'pets' to the FPGA Main VI to ensure communication is maintained.

System Health and FPGA Monitoring Loop - Send RT diagnostic information and FPGA monitoring data to the UI Main VI.

#CodeRecommended - Change 'Control Period' and 'Timeout Period' to values based on the requirements of your application.



Only the Message Handling Loop can stop the Parallel loops. The 'All RT Loop Stop' global is set to True in the 'Exit' message of the Message Handling Loop. **Do not** write to this value anywhere else.

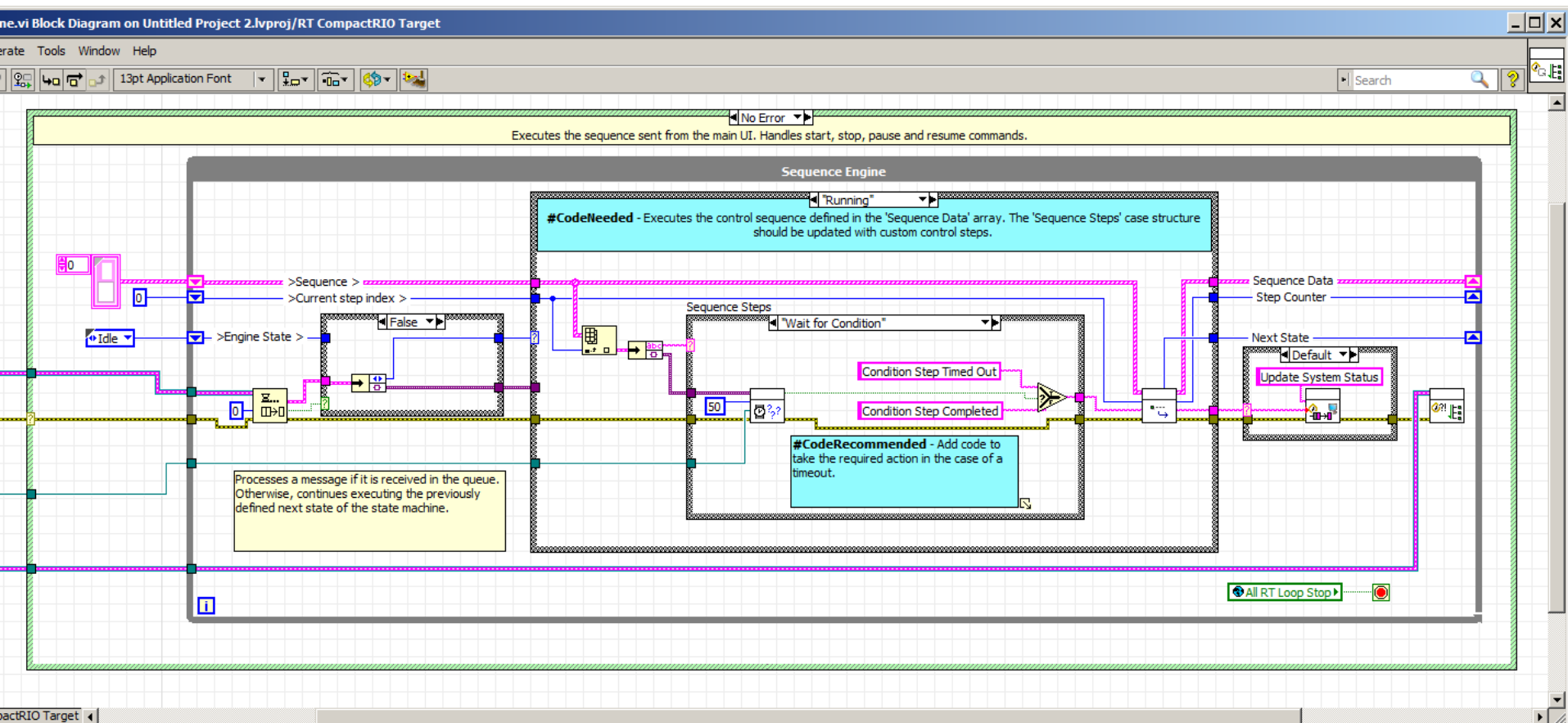


#CodeRecommended - When debugging an application, set the 'Debugging' global to 'TRUE'. Note that 'TRUE' is the default setting to set the global value to 'FALSE' before your application into an executable.

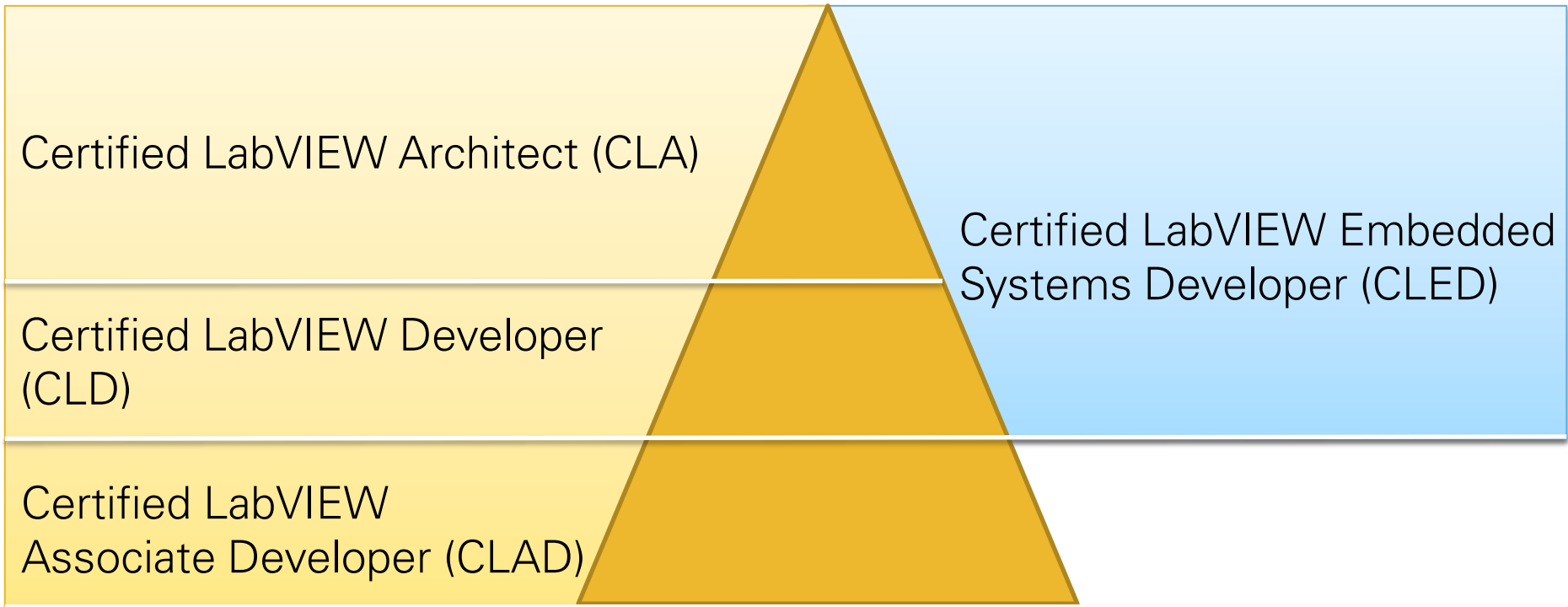
Alternatively, consider populating the global from an INI file, so that a reboot is necessary to change the 'Debugging' global.

Debugging

Mecanismo do Sequenciador

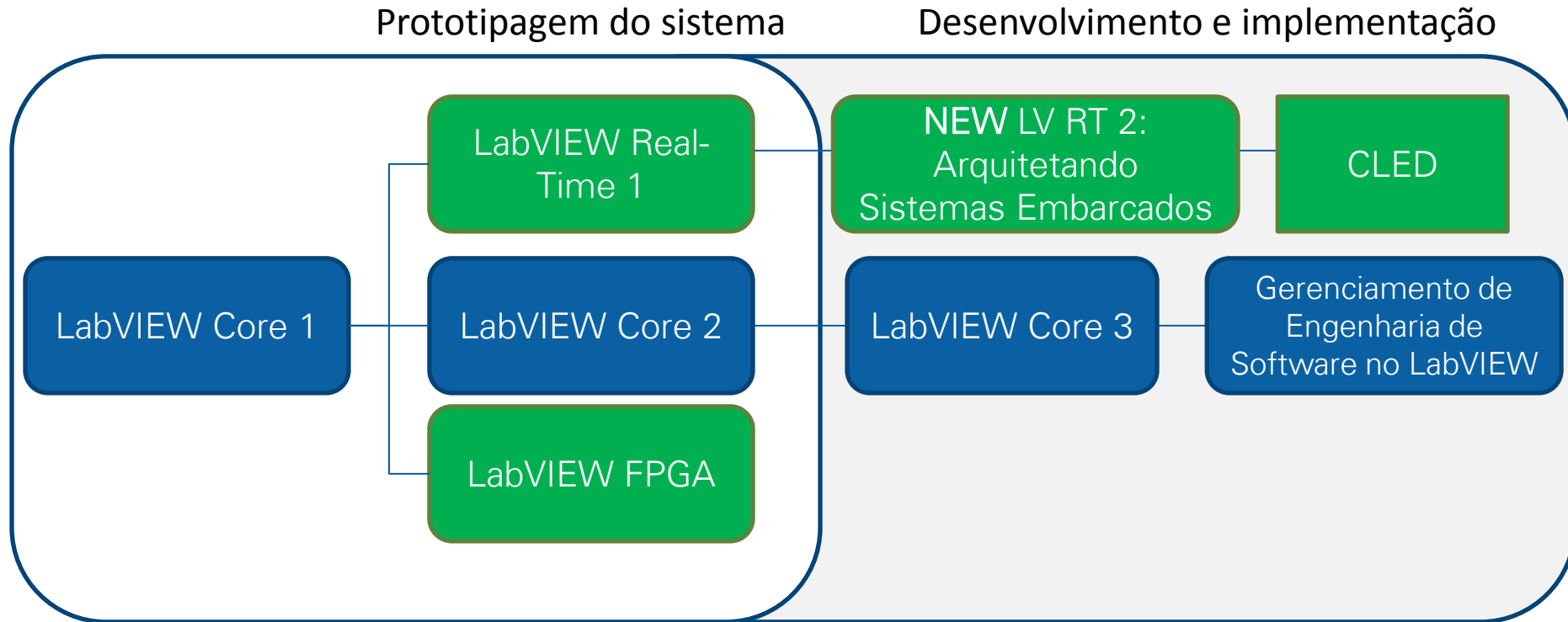


Nova certificação: Certified LabVIEW Embedded Systems Developer (CLED)



Projete seus sistemas embarcados com confiança
ni.com/CLED

O caminho mais rápido para uma melhor proficiência



Todos os cursos estão agora disponíveis no treinamento
online autoguiado
ni.com/self-paced-training

Perguntas e respostas...