

# Software Packages for the Development of Powerful Drive Systems

Erik van Hilten

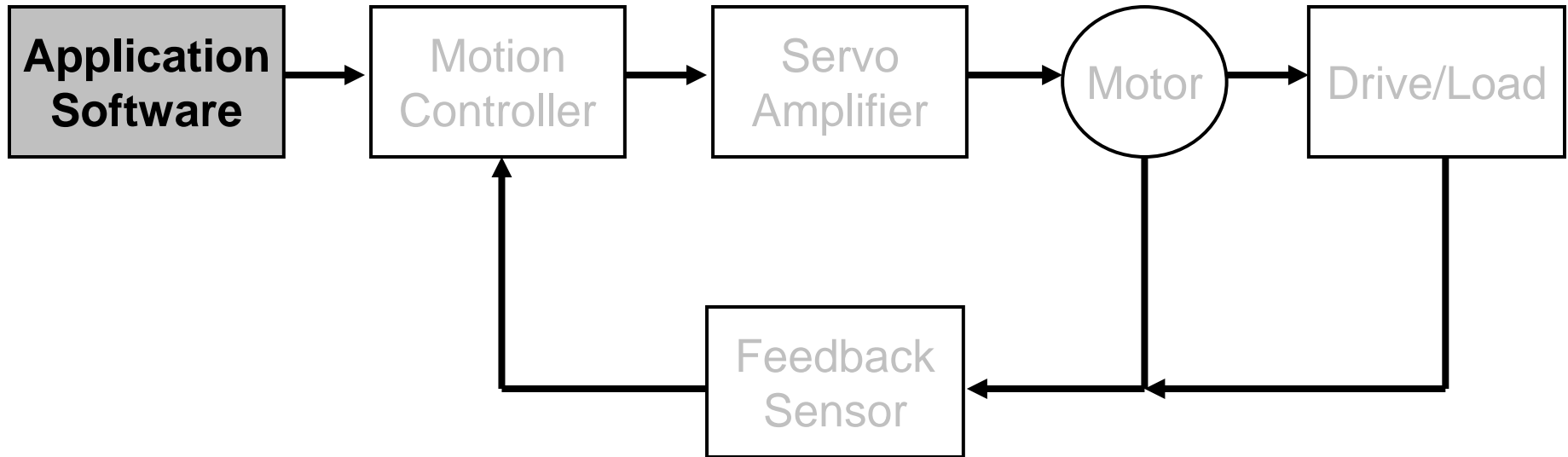
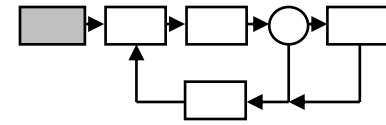
Rik Prins

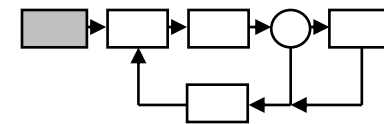
National Instruments

# Agenda

- Step by step to the application
- Configuration
- Prototype creation
- Application development
  - NI-Motion
  - Interfaces (CANopen, Serial)
  - NI-SoftMotion

# Drive System





# Application Development

## Test and Data Management Software

NI TestStand, DIAdem

### Interactive Tools

**SignalExpress**  
**Vision Builder AI**  
**Motion Assistant**

### Application Development Software

**LabVIEW**

Graphical Development Environment

**LabWindows/  
CVI**

ANSI-C development

**Measurement  
Studio**

Visual Studio Components

## Measurement and Control Services



GPIB/Serial  
and VXI



Data Acquisition and  
Signal Conditioning



Modular  
Instrumentation



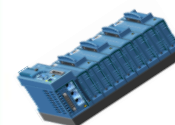
PXI/CompactPCI



Motion



Vision



Distributed I/O



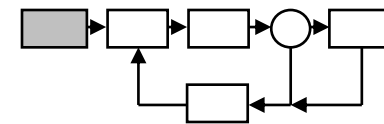
PLCs

**maxon motor**

driven by precision

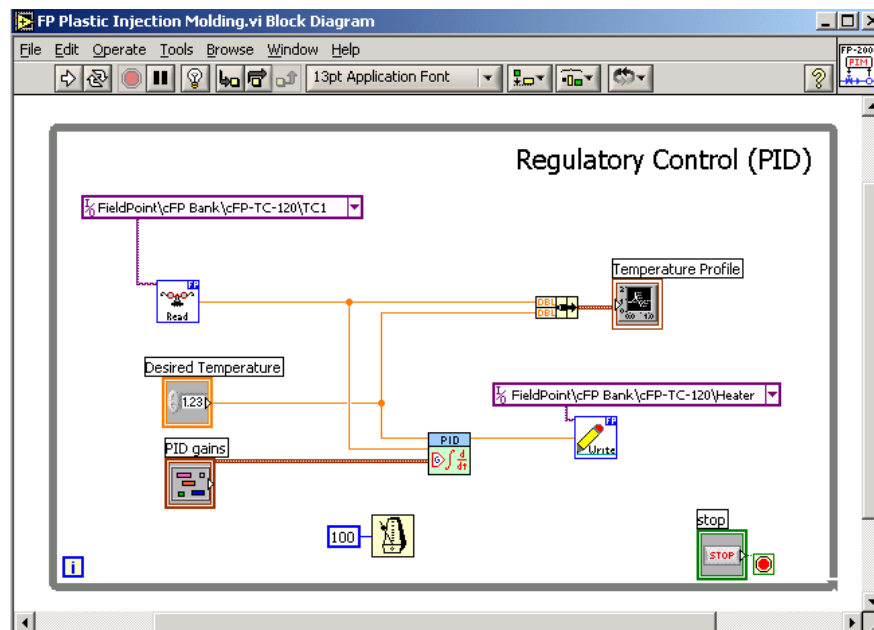
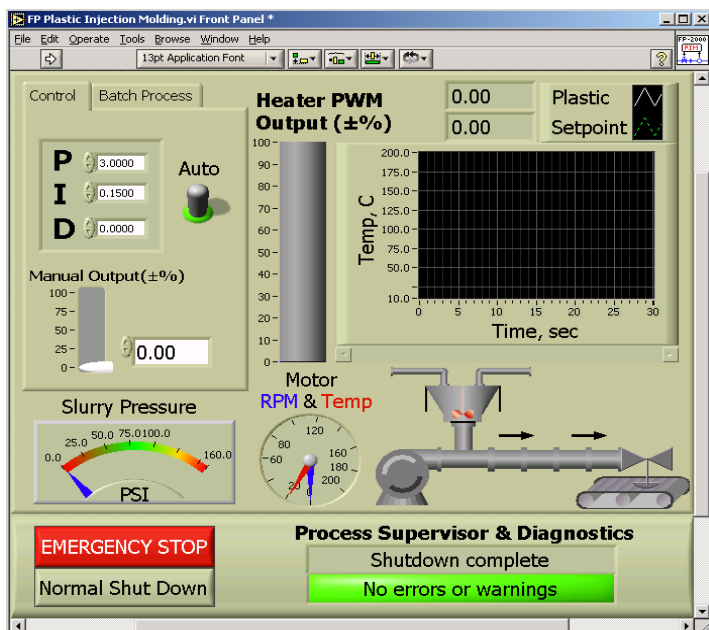
Motion under Control





# NI LabVIEW

## Graphical Programming



- Graphical programming language for engineers
- More than 650 control, regulating and analytical functions
- Simple data entry, motion and vision integration
- 20 years, tested industry standard

## LabVIEW

C\*

The image displays two identical screenshots of a LabVIEW block diagram, likely representing a data acquisition system. The diagram is contained within a main frame with a title bar showing a clock icon and the text "ms".

On the left side of the frame, there is a control panel with several settings:

- A clock icon followed by "ms".
- A "Sample Rate" control set to "1 kHz".
- A "Time Delay" control set to "1000".
- A "Start Time" control set to "0".
- A "Stop Time" control set to "100".
- A "DAQ Assistant" control set to "L7660".
- A "DAQ Assistant" control set to "D,M".

The main area of the diagram contains the following components:

- A "DAQ Assistant" block (represented by a book icon) with the text "data" below it.
- A "Waveform Graph" block (represented by a graph icon) connected to the "DAQ Assistant" block by a dashed blue line.
- A "Stop" button (a green rectangle with a red "STOP" label) located below the "Waveform Graph".
- A red LED indicator (a red circle with a green border) connected to the "Stop" button by a dashed green line.
- A small "i" icon (information) in the bottom left corner of the main frame.

[illegible]

\* does it

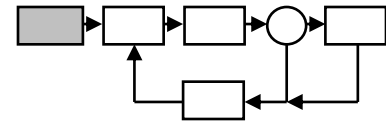
**driven by precision**

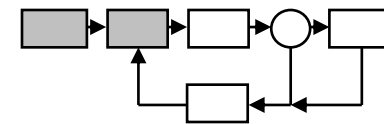
## Motion under Control



# Demo

## LabVIEW Development System





# Application Development – Motion

## Test and Data Management Software

NI TestStand, DIAdem

### Interactive Tools

**Motion Assistant**

### Application Development Software

**LabVIEW**

NI-Motion, NI-SoftMotion, ...

**LabWindows /  
CVI**

Motion device driver

**Measurement  
Studio**

Well documented DLLs

**NI-Motion (driver, API, DLLs, examples), NI-SoftMotion, NI-CAN**



PCI



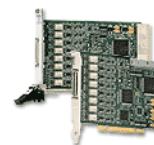
PXI



RIO



FP



FPGA



CANopen

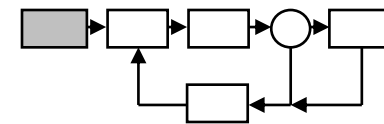
**maxon motor**

driven by precision

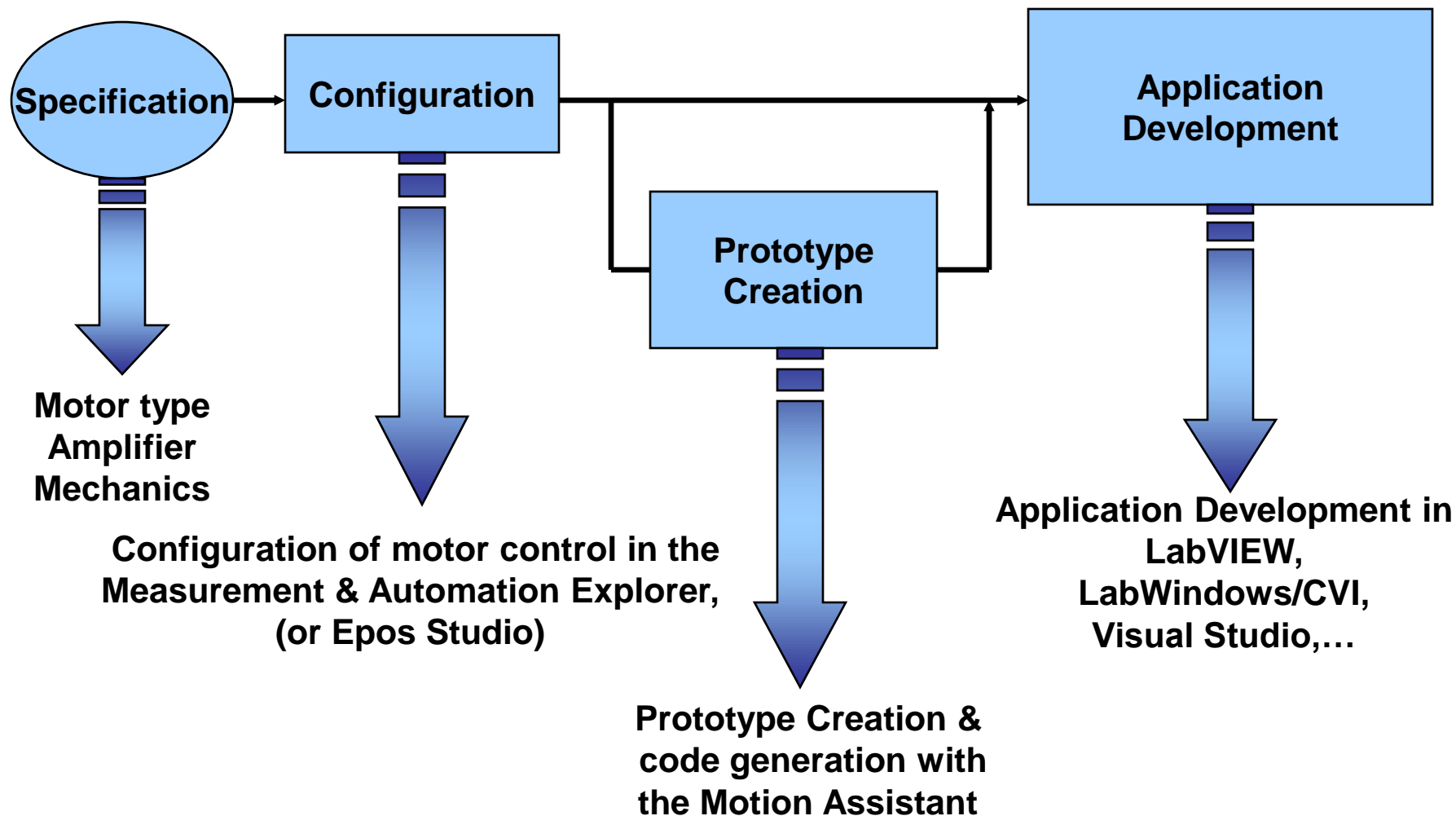
Motion under Control







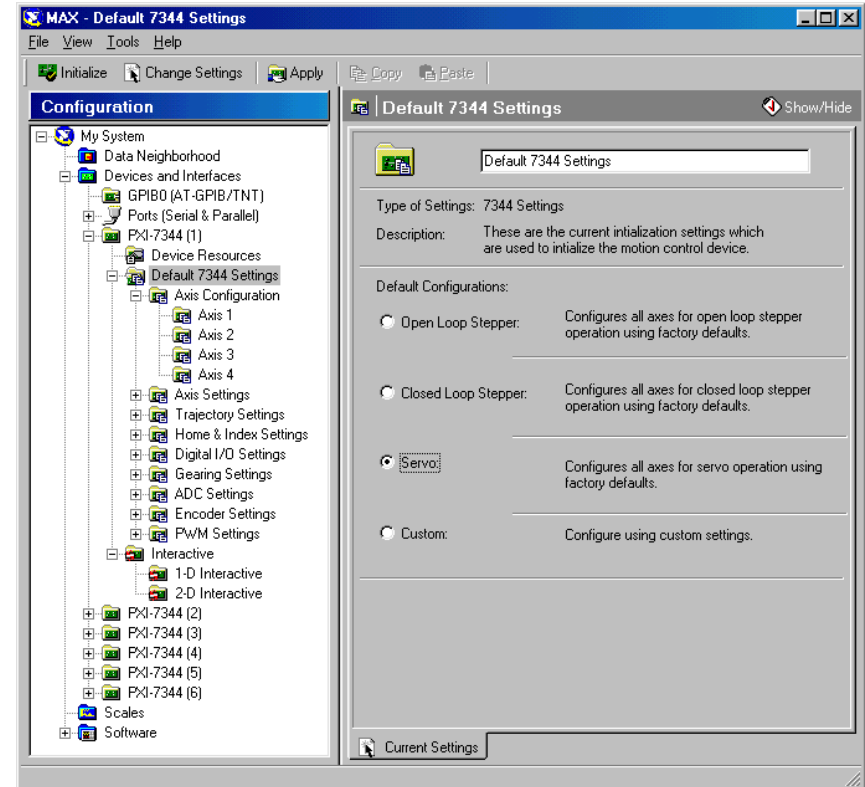
# Step by step to the application

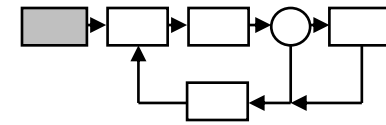


# Configuration

## Measurement & Automation Explorer

- is used to configure all NI hardware
- configuration of motion card
- configuration of axis and limit switches
- expanded motion settings (trajectory, ...)
- manual test (position, velocity, ...)
- tuning control parameters for servo motors



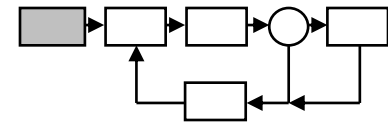


# Demo

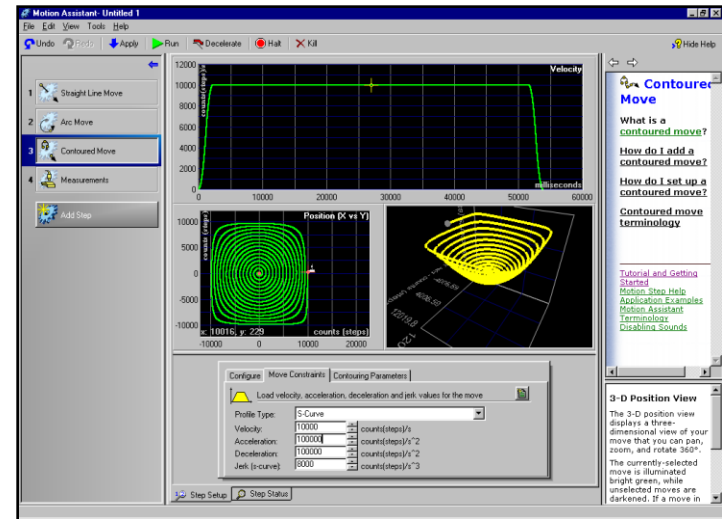
## Measurement & Automation Explorer General

# Prototype Creation

## NI Motion Assistant

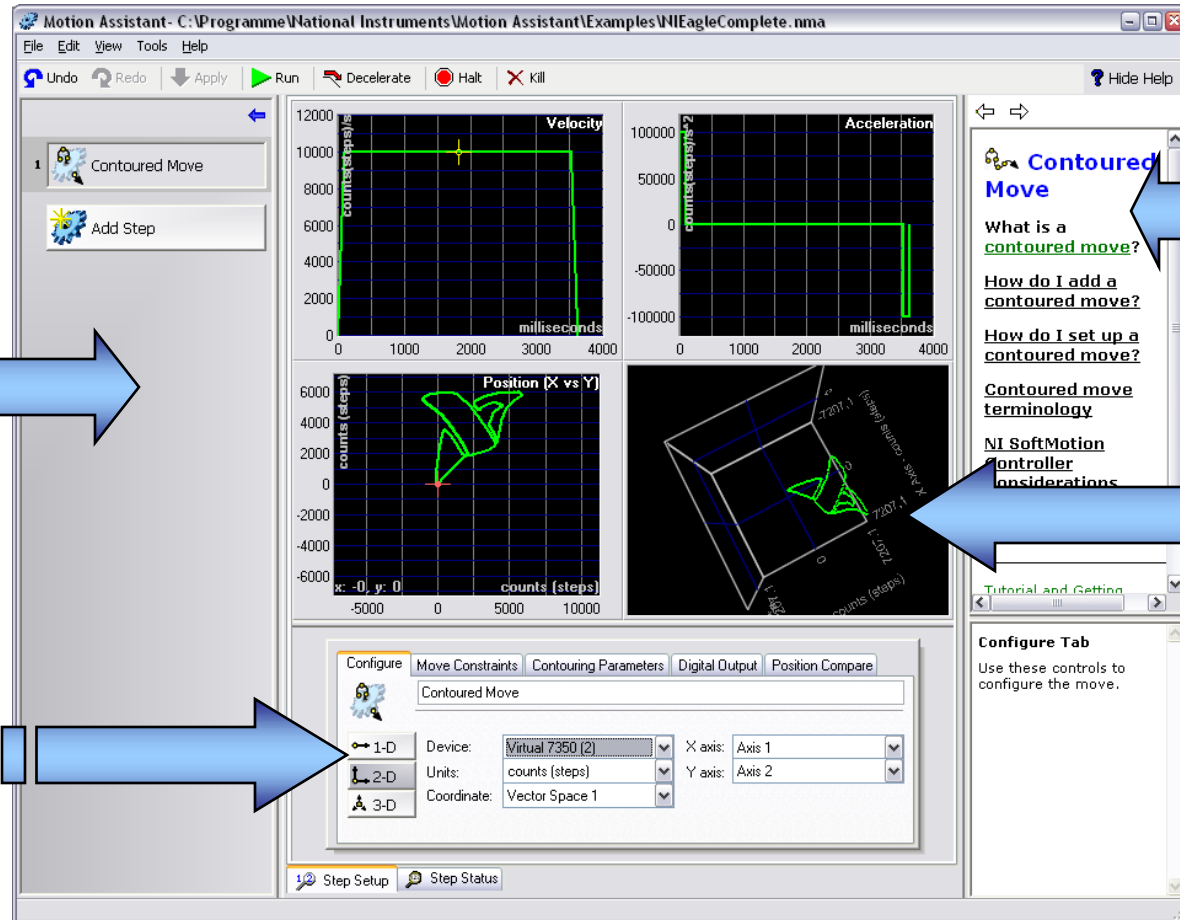
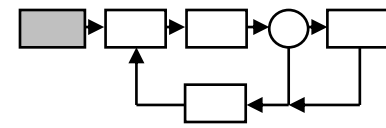


- interactive environment with 2D and 3D visualization
- simple trapezoids and S-curve profile velocity implementation
- CAD file import
- LabVIEW and C-code generation



# Motion Assistant

## User Interface



Script Bar

Step Help

Preview

Properties

**maxon motor**

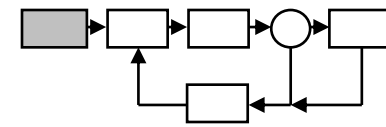
driven by precision

Motion under Control



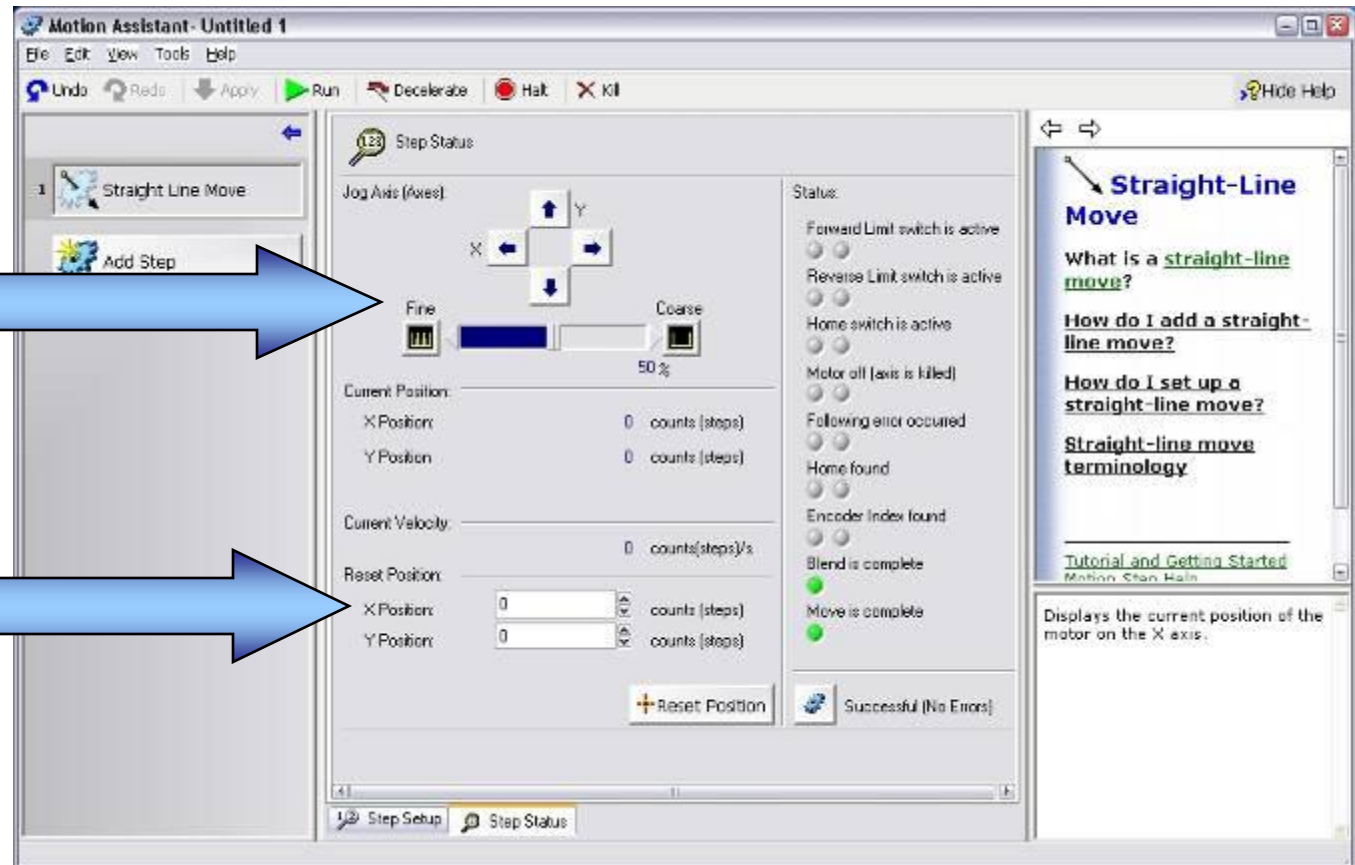
# Motion Assistant

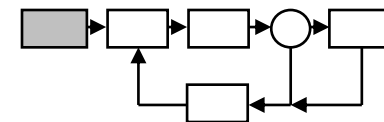
## User Interface



Manual  
Positioning


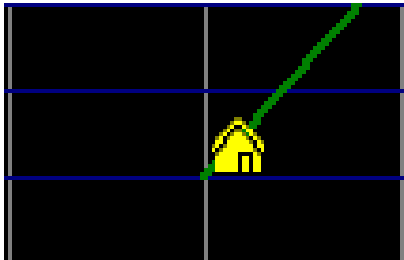

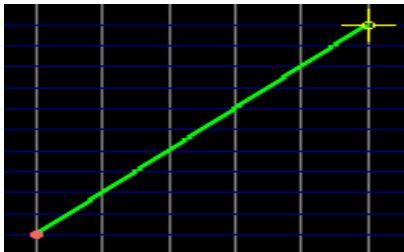

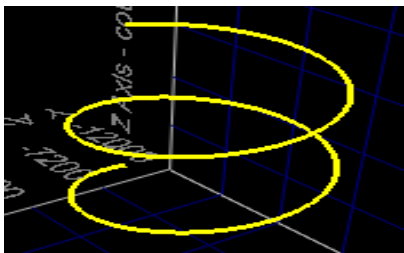
Status  
Indicator





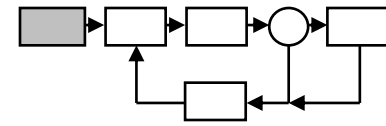
# Motion Assistant


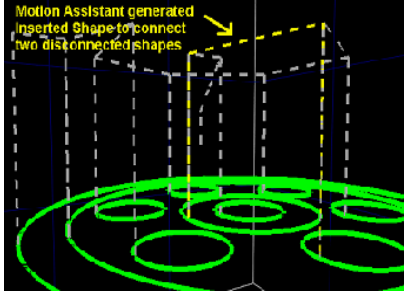

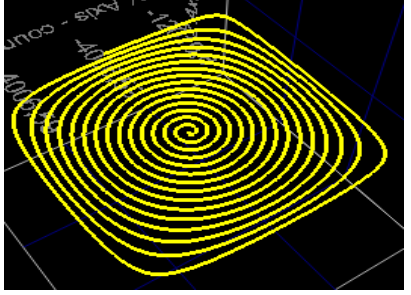

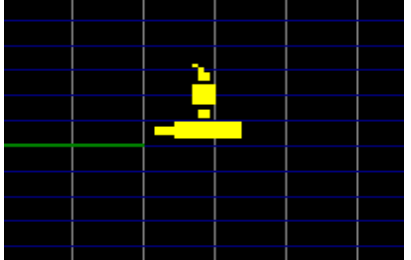
## Script Blocks

|   |                   |   |  |
|---|-------------------|---|--|
|  Reference Move     | Reference run     | Finds the reference switch or the index position of the encoder |   |
|  Straight Line Move | Linear motion     | Executes a linear run in 1, 2 or 3 dimensions                   |   |
|  Arc Move           | Arc-shaped motion | Executes an arc-, circle or spiral-shaped motion (2D or 3D)     |  |

# Motion Assistant

## Script Blocks



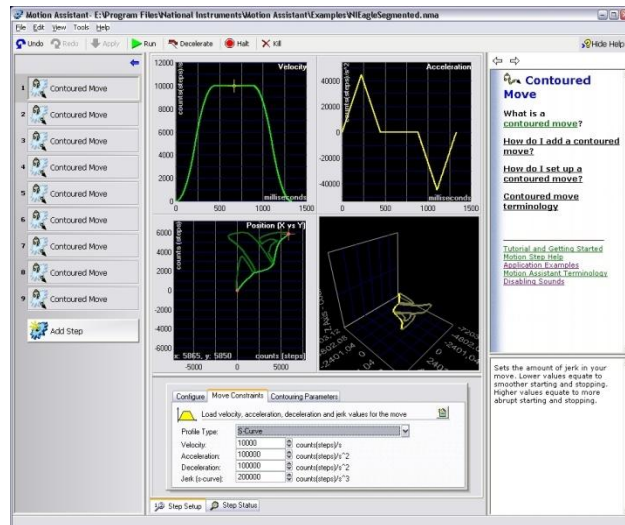
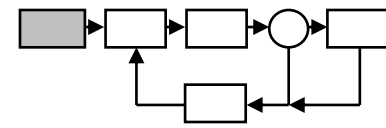
|   |                |  |   |
|---|----------------|--|---|
|  CAD Move       | CAD Move       | Import of CAD data from DXF File   |    |
|  Contoured Move | Contour motion | Executes any contour motions that are given by support points (ASCII, Excel or Teach In) |    |
|  Measurements | Measurement    | Placeholder for technical measurement functions  |  |



# Motion Assistant

## Code Generation

ASCII plain text file with documentation and description of all motion functions used and their parameter



Pseudo-code



LabVIEW-VI



Structured, documented graphical code

```
mynew.vi - Notepad
File Edit Format View Help

In the NI-Motion API, vector spaces are numbered starting with 17, or Axis 1. Therefore in this pseudocode, when "vector space 1 (17)" is referenced, the number in parenthesis is the actual value of the vector space resource.

Information about the NI-Motion functions or VIs can be found in the software reference manual or the VIs online help.

Configure
Configure vector space 1 (17) on board ID 1. Map the x axis to axis 1, the y axis to axis 2, and the z axis to axis 0. Mapping the z axis to 0 indicates a two-axis move, while a non-zero number indicates a three axis move.

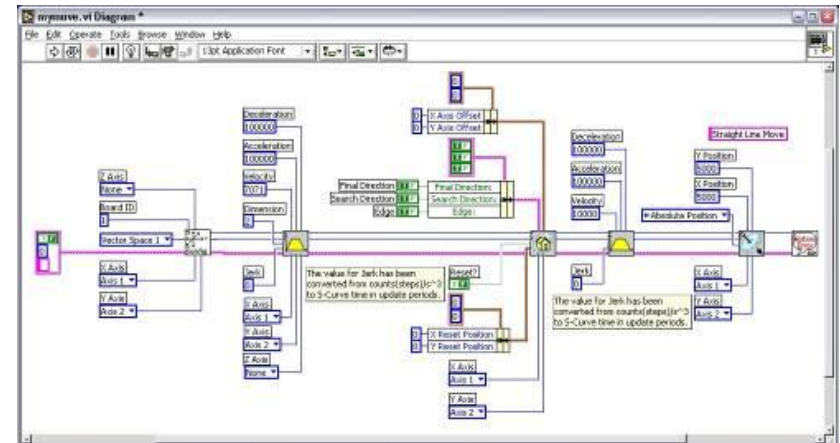
The NI-Motion function that should be used is:
Configure vector space function.

Move Constraints
Load the move constraints for the 2 axes. The x axis is axis 1, the y axis is axis 2, and the z axis is axis 0. Any axis that is zero can be ignored, because less than three axes are being used. Set the velocity to 2071.087038048 counts/sec, the acceleration to 10000 counts/sec/sec, the deceleration to 10000 counts/sec/sec, and the s-curve time to 0 update periods on each axis.

The NI-Motion functions that should be used are:
Load Acceleration-Deceleration
Load Velocity
Load S-curve Time

Find move on 2 axes - "Reference Move"
Set the operation mode for axes 1 and 2 to Absolute position mode. Enable home inputs and limits on all the axes. Start the find home on both the axes. Wait 1.000 s for the axes to start finding home. After the following until both axes have either found home or timed out, following error is exceeded on any axis, a motor is off, or there is a function error:
Wait 100 milliseconds.
Read the status of both the axes.
If an axis is move complete, do the following:
If the home input is found, record the current time for this axis.
If 1.000 s have elapsed since you recorded the time for this axis, then this axis has timed out.
When the above loop has terminated, disable the home inputs on all the axes.

The NI-Motion functions that should be used are:
Set operation mode
Enable home inputs
Enable Limits
```



maxon motor

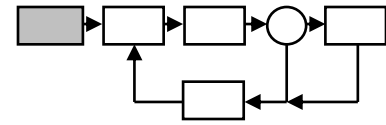
driven by precision

Motion under Control

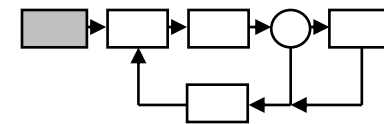


# Demo

- XY table

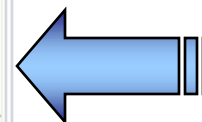
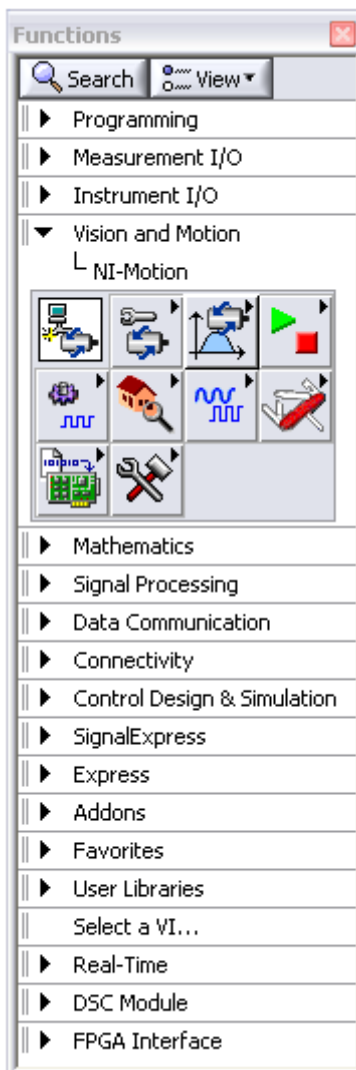






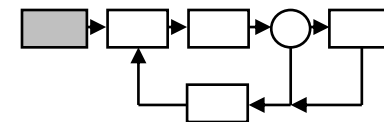
# NI LabVIEW Motion-VI Library

**NI-Motion  
Function Palette**



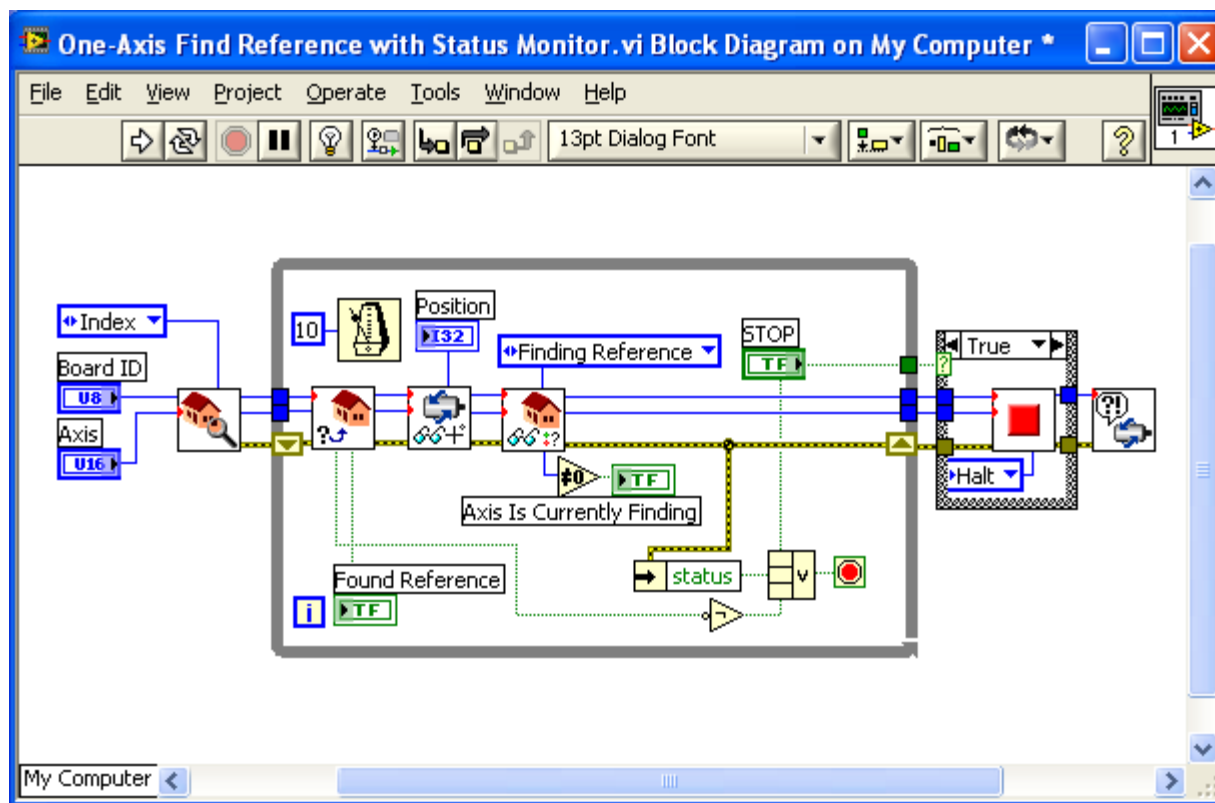
## **NI-Motion**

- Initializing
- Axis configuration
- Trajectory control
- Start & Stop Motion
- Motion I/O
- References
- Analog & digital I/O
- Error & Utility
- Onboard programming
- Advanced functions



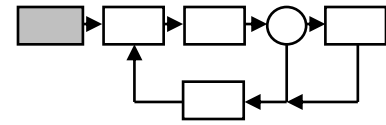
# NI LabVIEW

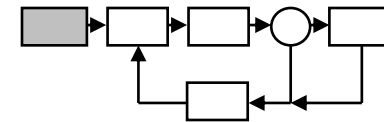
## Programming Example



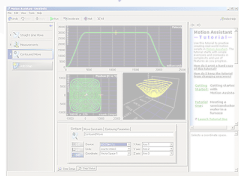
# Demo

## 3-Axis Demo





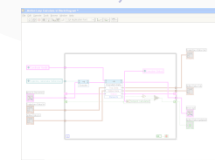
**PC based (Plug-In)**  
PCI, PXI



**Distributed**  
CANopen



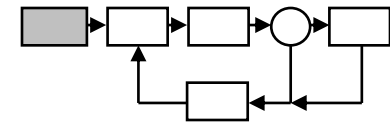
**Custom**  
RTOS, FPGA



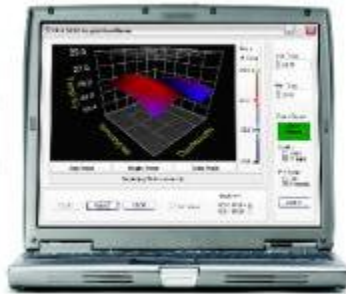
# Application Development LabVIEW and Decentralized Controller

# Application Development

## LabVIEW and Interfaces



Computer with CAN card



VI  
download



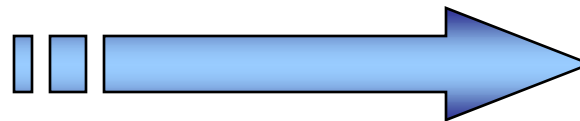
RT Target with CAN card

Programming with:

- CANopen Library
- EPOS library



Next EPOS



Next EPOS

**maxon motor**

driven by precision

Motion under Control





# Configuration with EPOS Studio

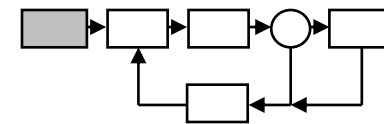


**maxon motor**

driven by precision

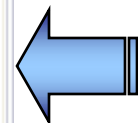
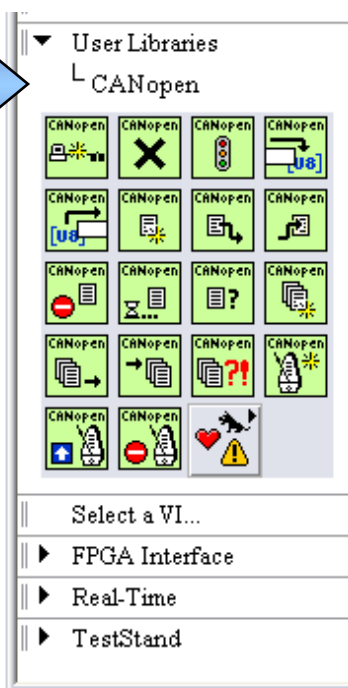
Motion under Control





# LabVIEW and CANopen Library

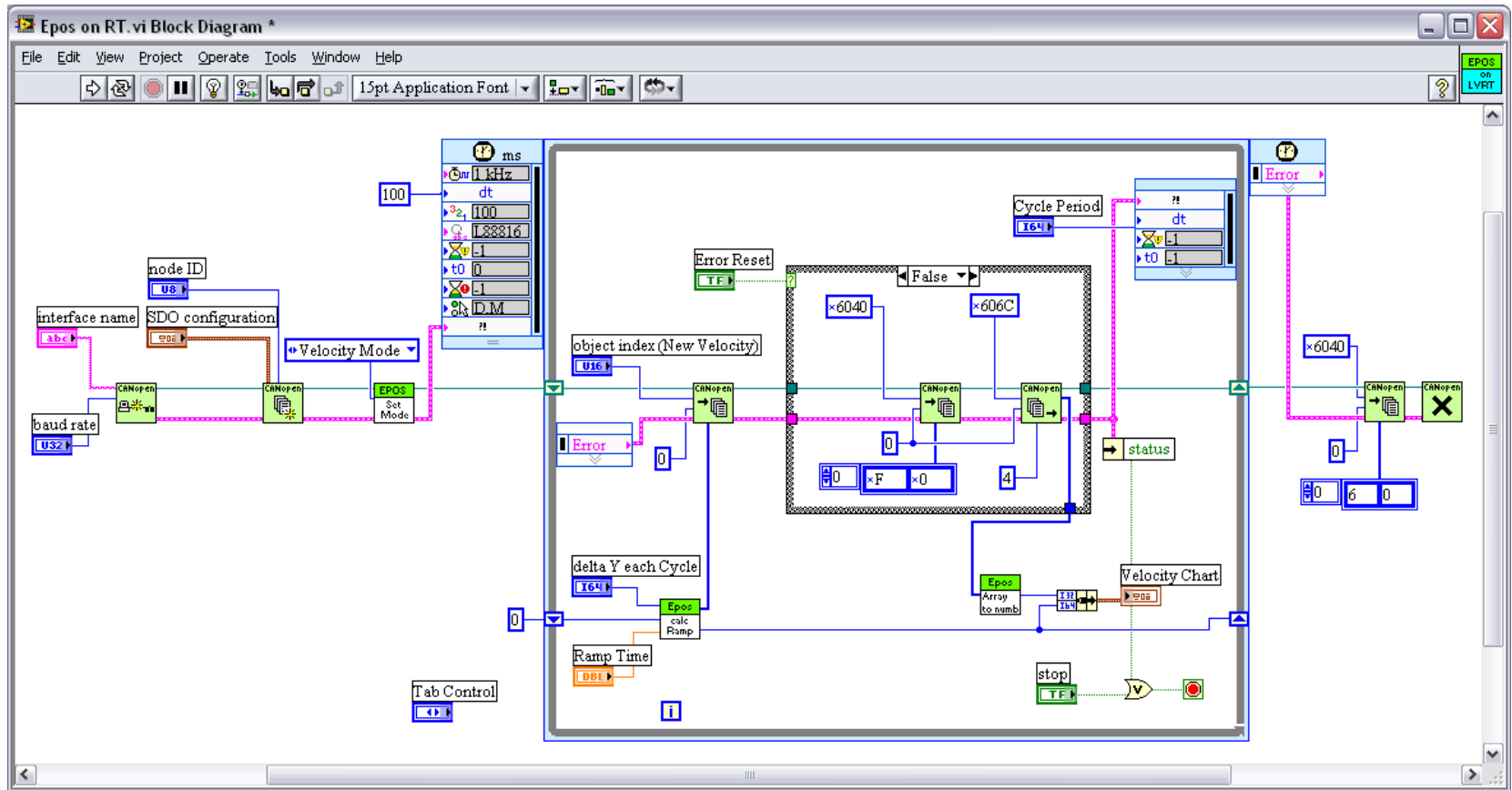
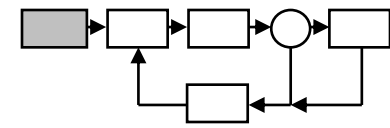
**CANopen  
Functions Palette**



## **CANopen**

- Initializing
- Sending and receiving service data objects (BDO)
- Sending and receiving process data objects (PDO)
- Network management
- Synchronization functions

# LabVIEW and NI-CANopen Programming Example

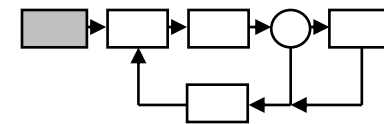


**maxon motor**

driven by precision

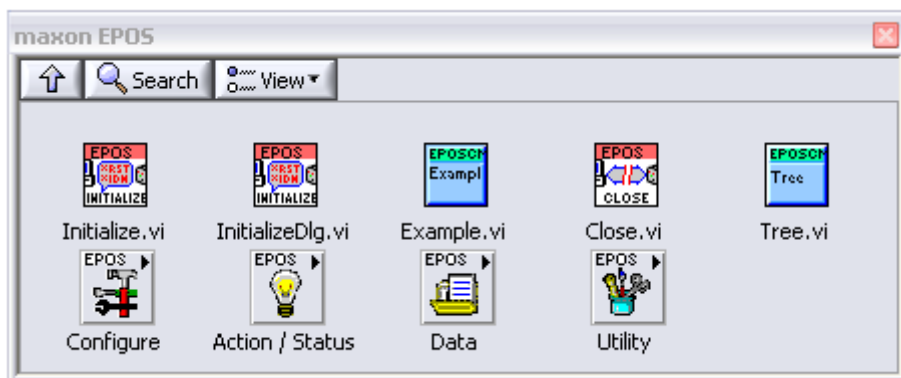
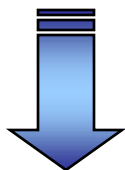
Motion under Control





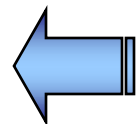
# LabVIEW and EPOS Driver

maxon EPOS functions palette

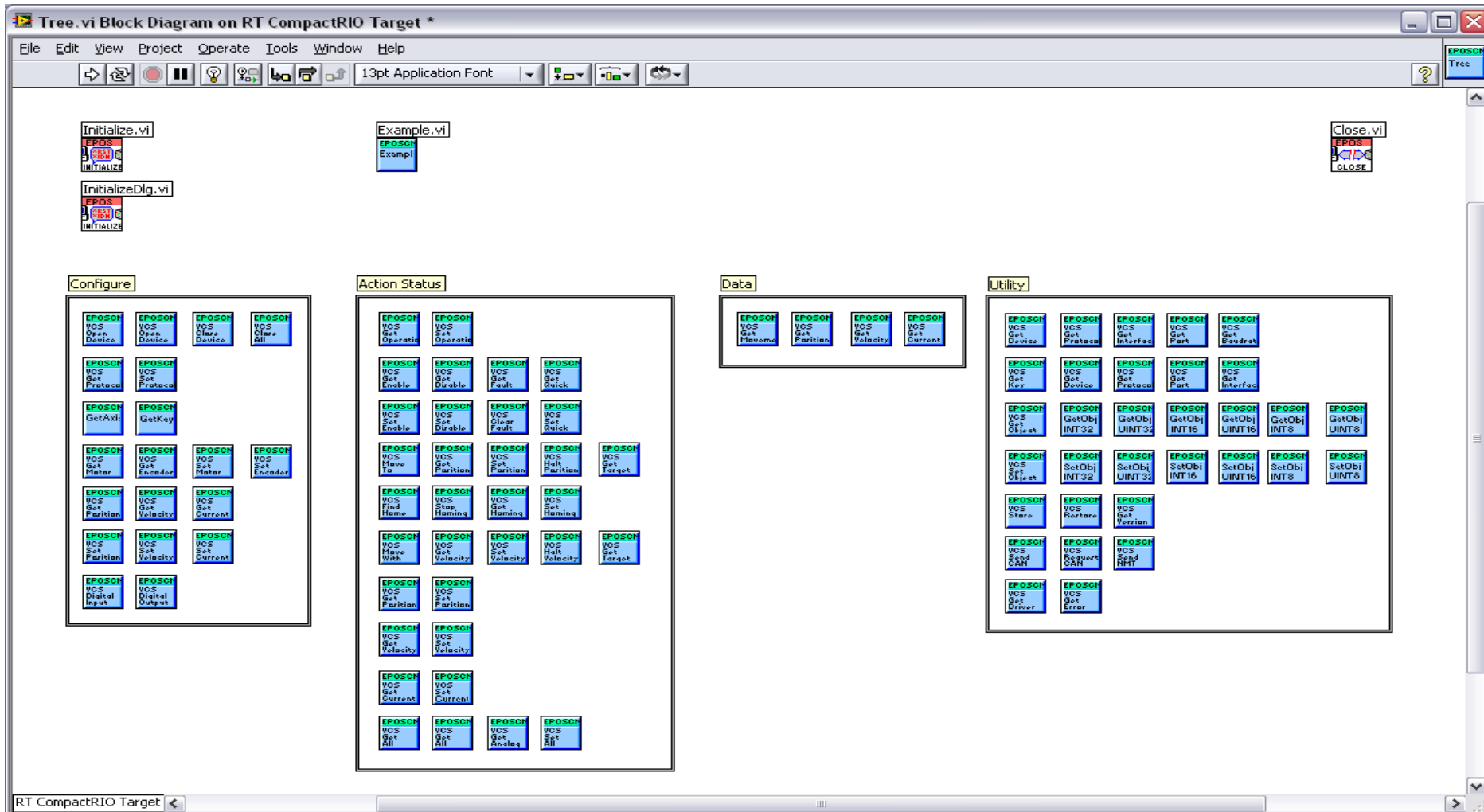
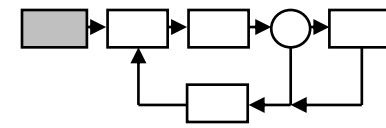


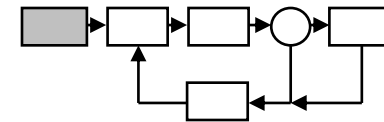
## EPOS driver

- Initializing
- Configuration
- Data exchange
- Settings
- Example



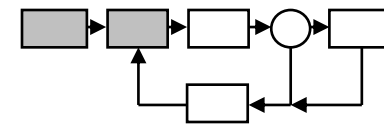
# LabVIEW and EPOS Driver Programming Possibilities



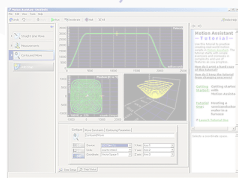


# Demo

- EPOS with EPOS Library



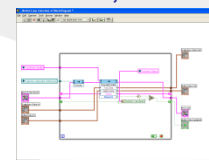
PC based (Plug-In)  
PCI, PXI



Distributed  
CANopen



Custom  
RTOS, FPGA



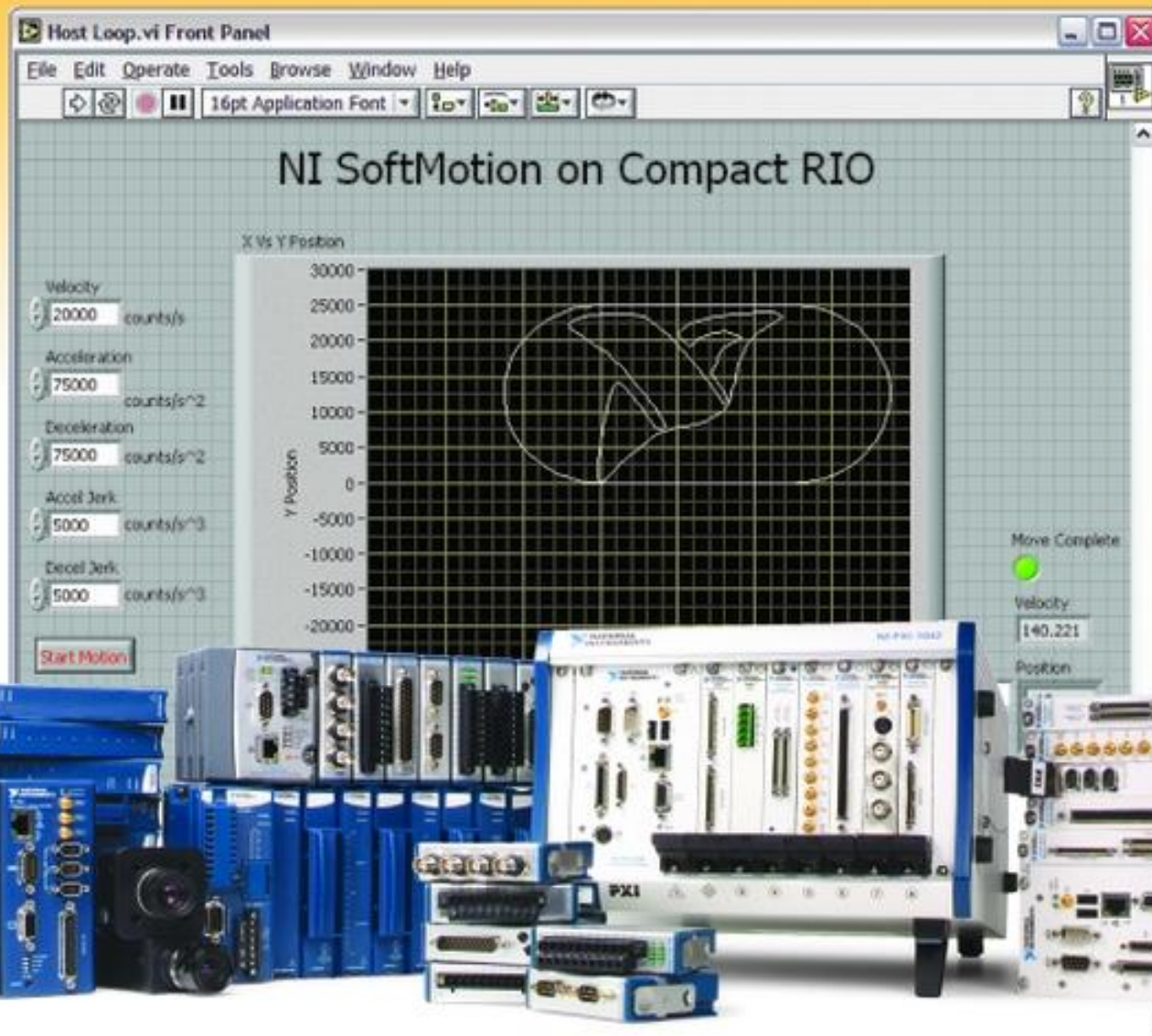
# Application Development with LabVIEW RT, FPGA and NI-SoftMotion

**maxon motor**

driven by precision

Motion under Control





**maxon motor**

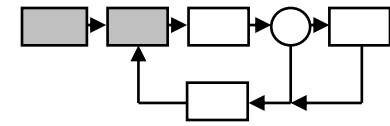
driven by precision

Motion under Control





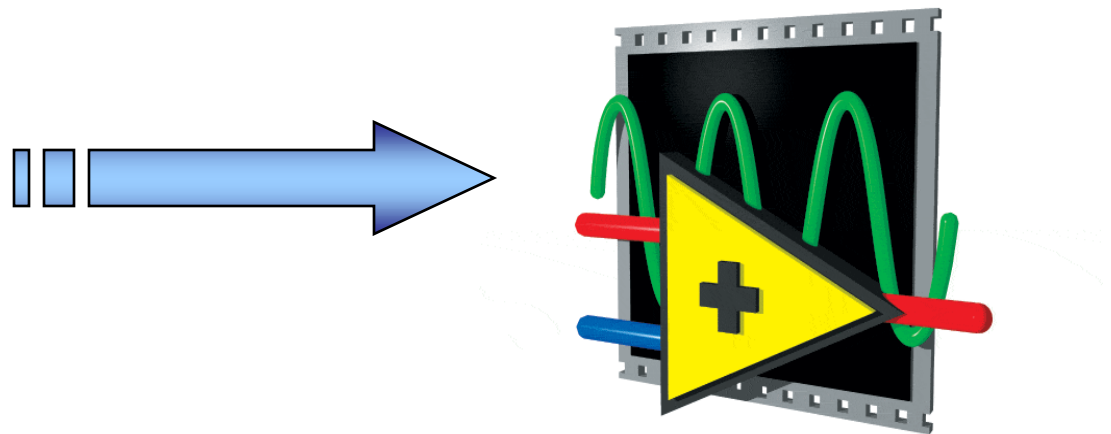
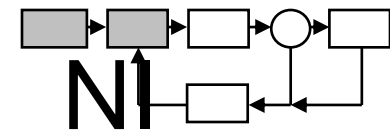
# LabVIEW and NI-SoftMotion



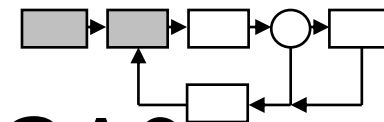
- 5 microseconds / 200 kHz servo update rates with CompactRIO and NI SoftMotion Development module
  - 40 kHz in DSP-based controllers
- synchronization with I/O

# Requirements for a NI-"Soft Motion" System

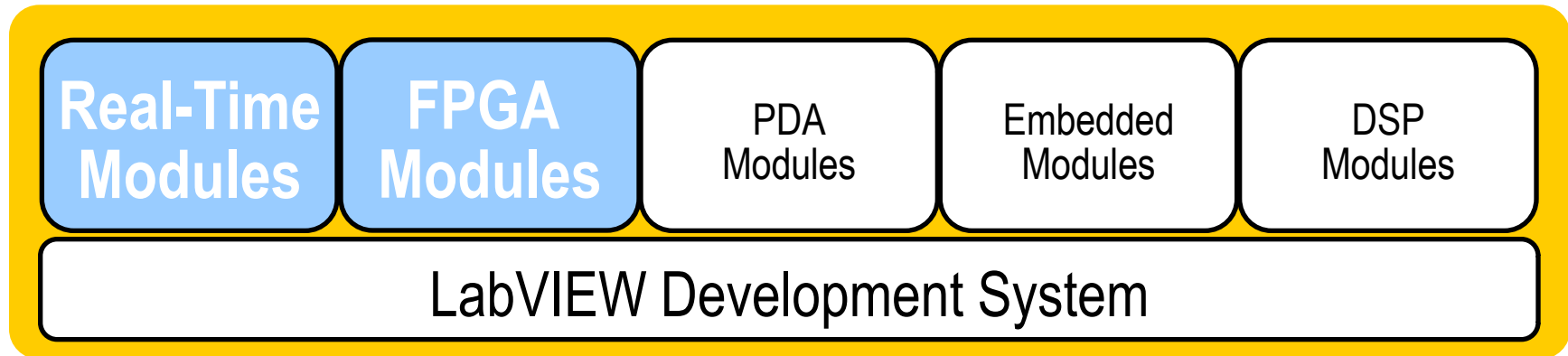
- deterministic cycle times
- minor jitter
- stability



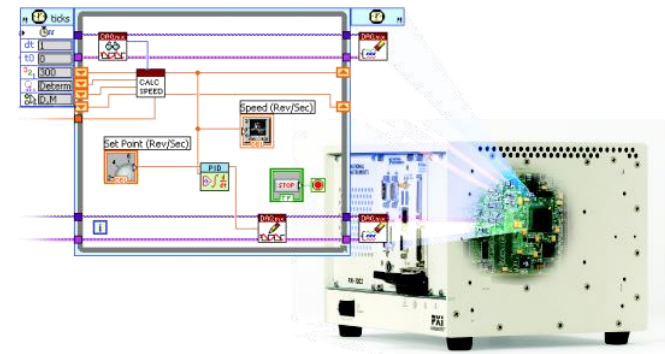
## LabVIEW Real-Time & FPGA

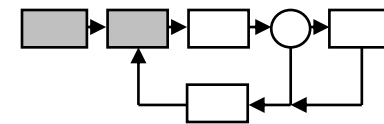


# What is LabVIEW Real-Time & FPGA?



- Development of deterministic applications with graphical programming
- Simple construction of distributed applications in the field of control, regulation and monitoring
- Use of COTS hardware



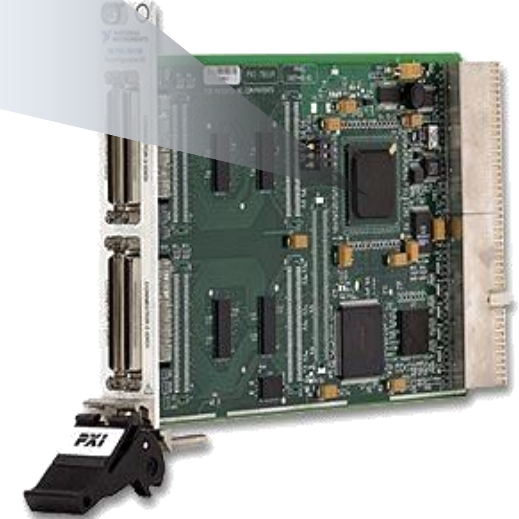
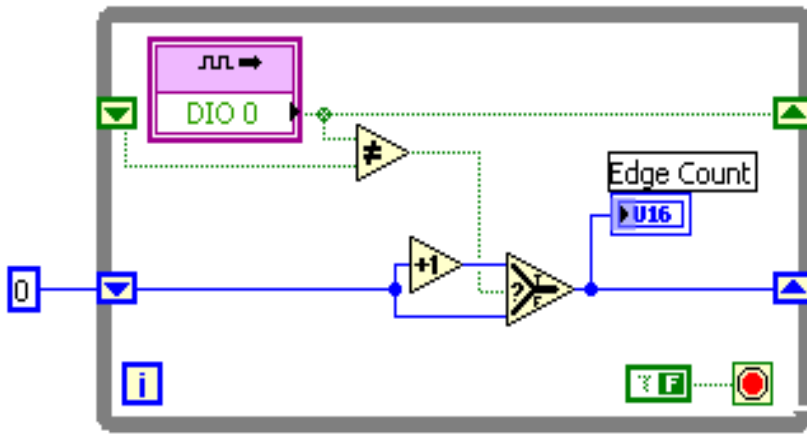
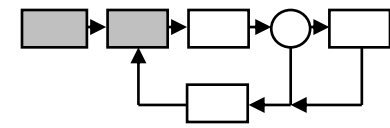


# What is LabVIEW Real-Time?

- Real-time **does not** always mean real fast
- Real-time means **absolute reliability**
- Real-time systems have timing constraints that must be met to avoid failure
- Determinism is the ability to complete a task within a fixed amount of time
- Develop LabVIEW code and run on dedicated reliable OS

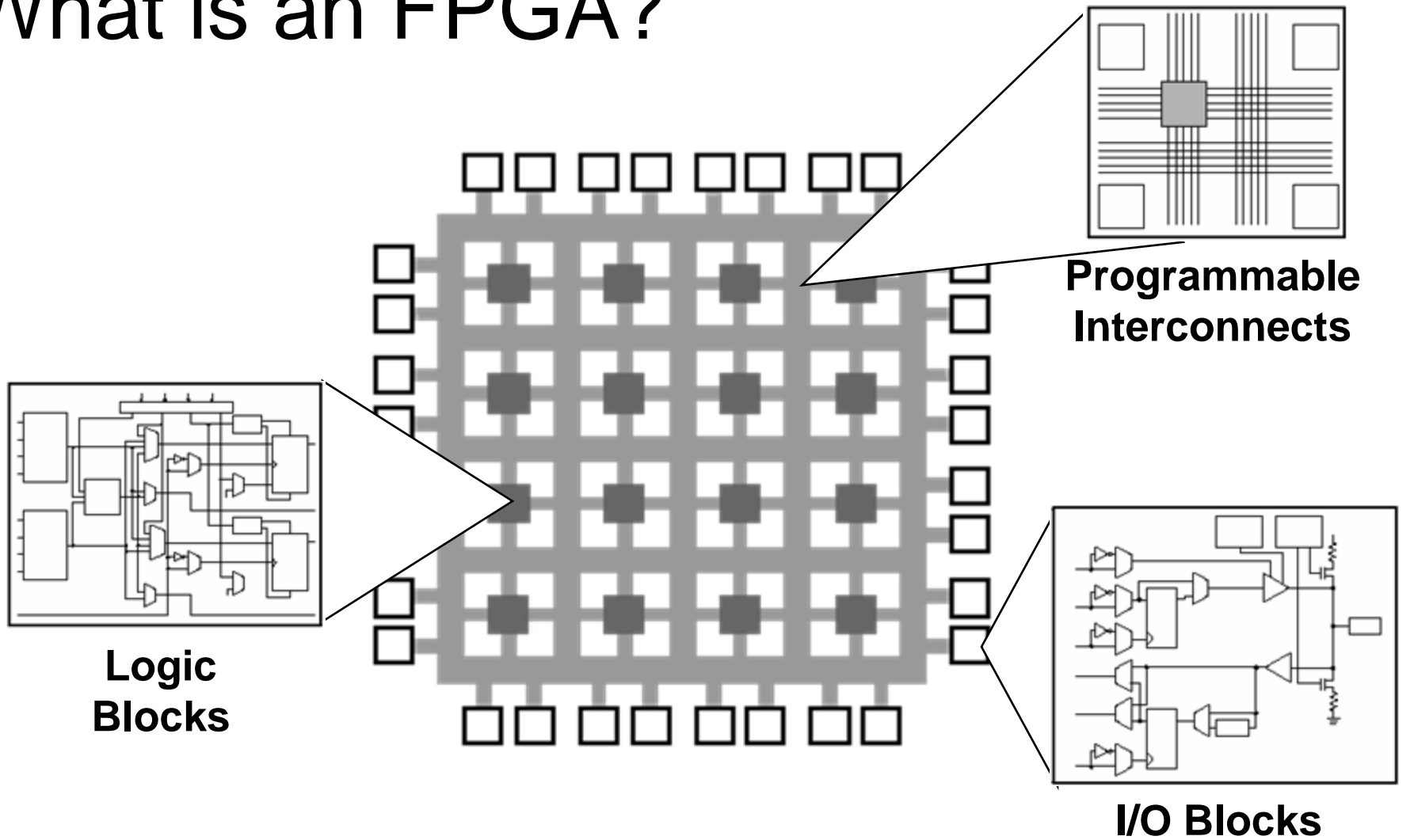


# LabVIEW FPGA module



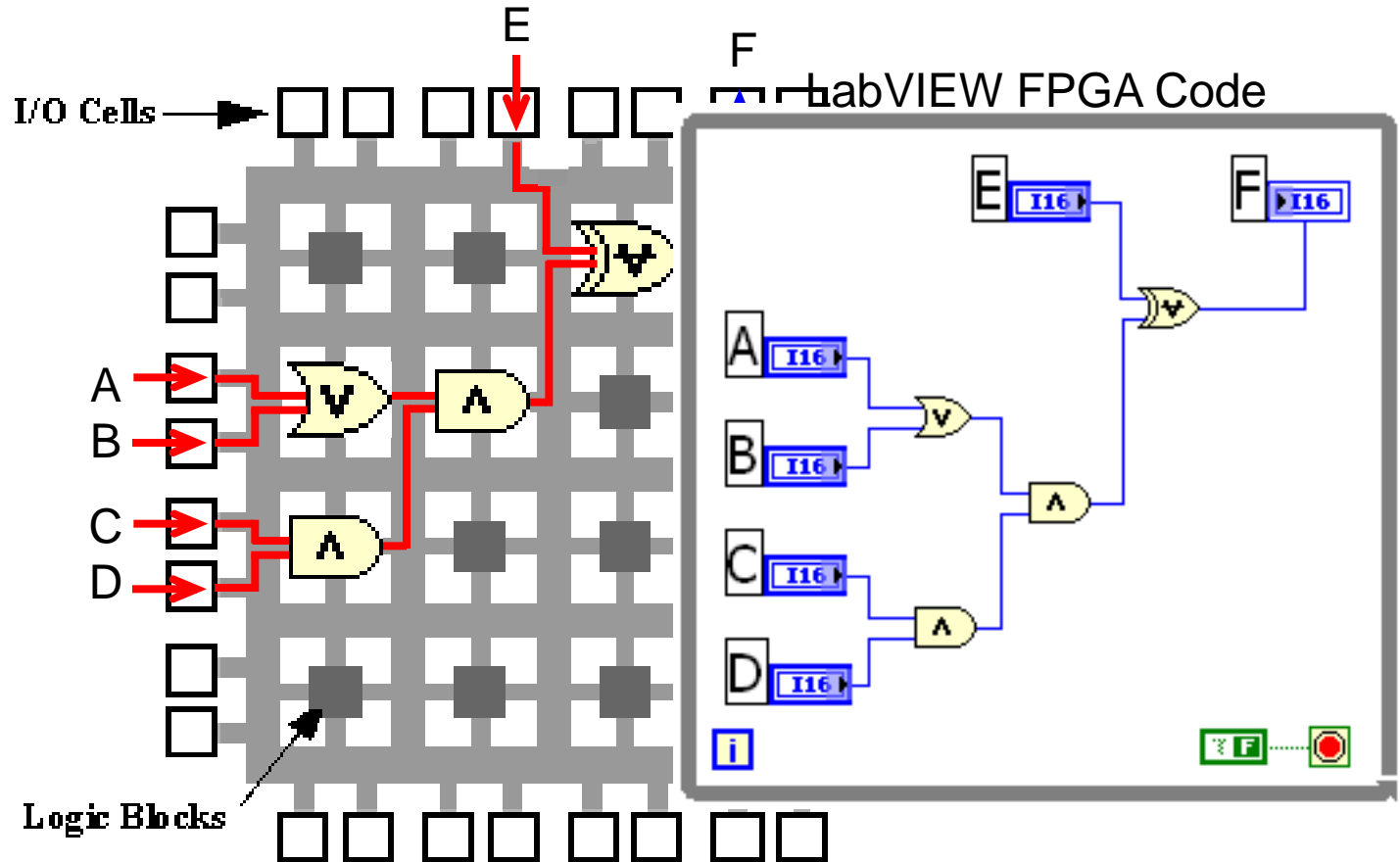
- Run LabVIEW Diagrams on an FPGA
- HDL knowledge not required
- Hardware speed, reliability and determinism
- Visualize and implement dataflow and parallelism
- Tight analog and digital I/O integration
- Intuitive programming for both embedded engineers and domain experts

# What is an FPGA?

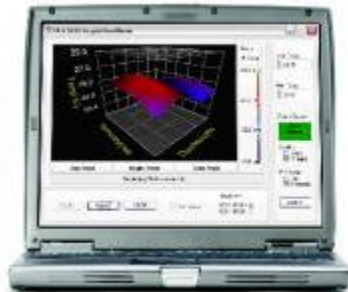
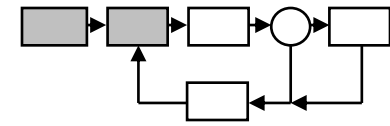


# Simplified FPGA Example

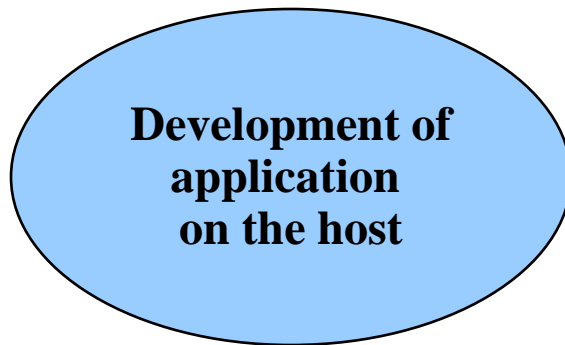
Implementing Logic on FPGA:  $F = \{(A+B)CD\} \oplus E$



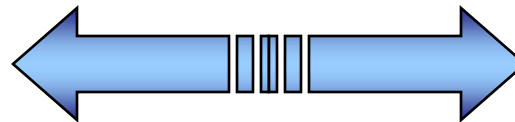
# LabVIEW Real-Time & FPGA



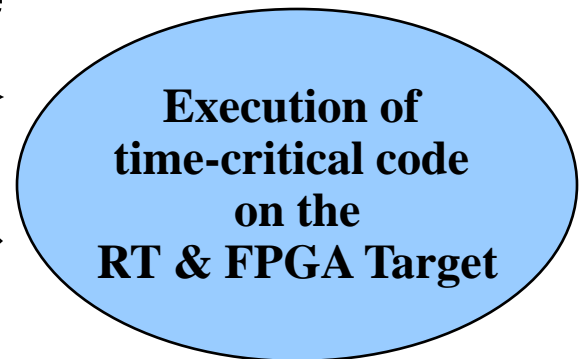
PAC



Download of RT code

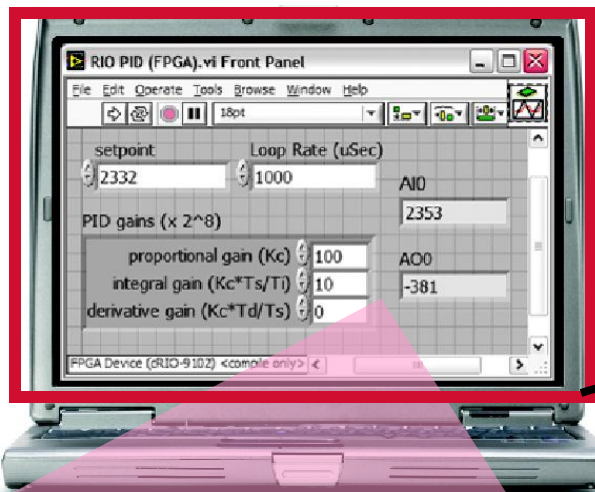
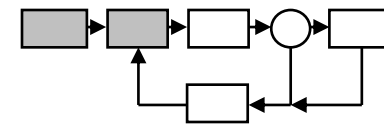


Data Exchange





# Realization on PXI



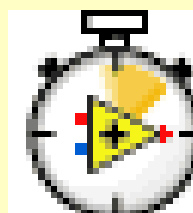
User Interface



LabVIEW

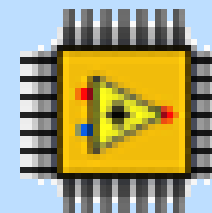


Real-Time Processor



LabVIEW Real-Time

Reconfigurable FPGA



LabVIEW FPGA

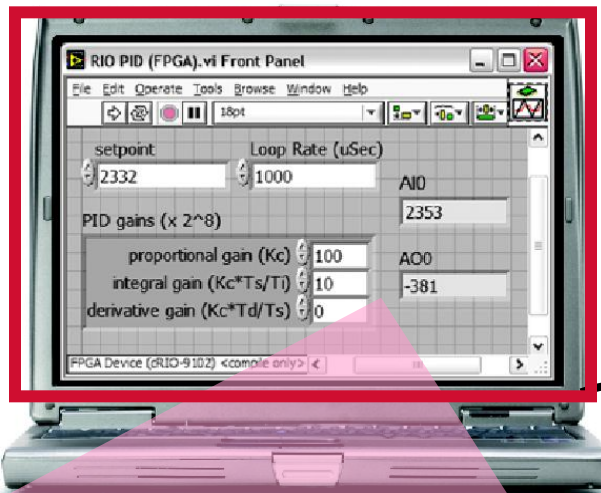
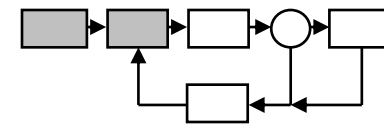
**maxon motor**

driven by precision

Motion under Control



# Realization on CompactRIO



User Interface



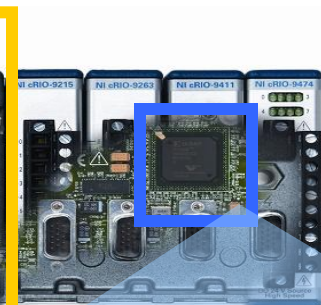
LabVIEW



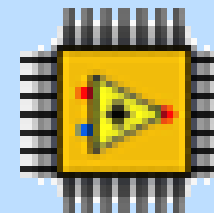
Real-Time Processor



LabVIEW Real-Time



Reconfigurable FPGA



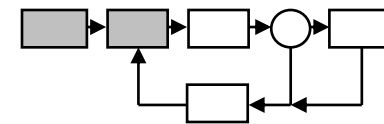
LabVIEW FPGA

**maxon motor**

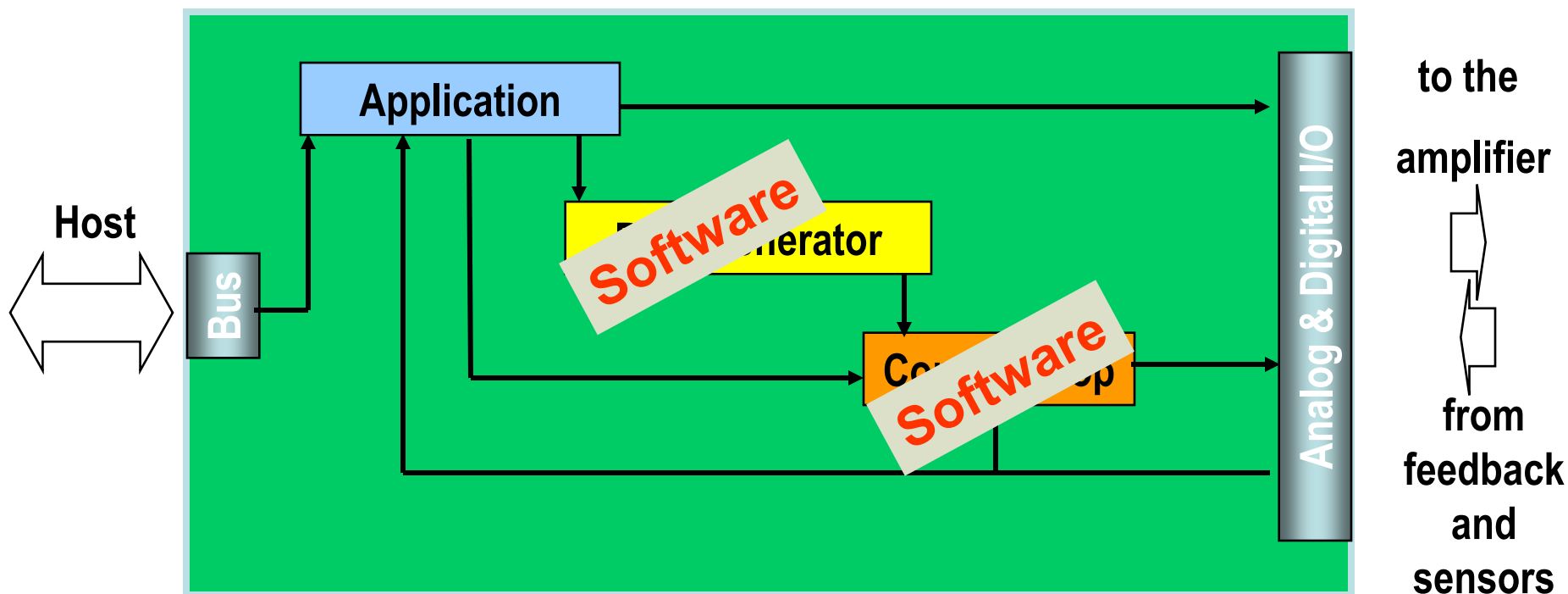
driven by precision

Motion under Control



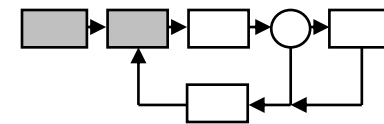


# Typical Motion Controller Architecture



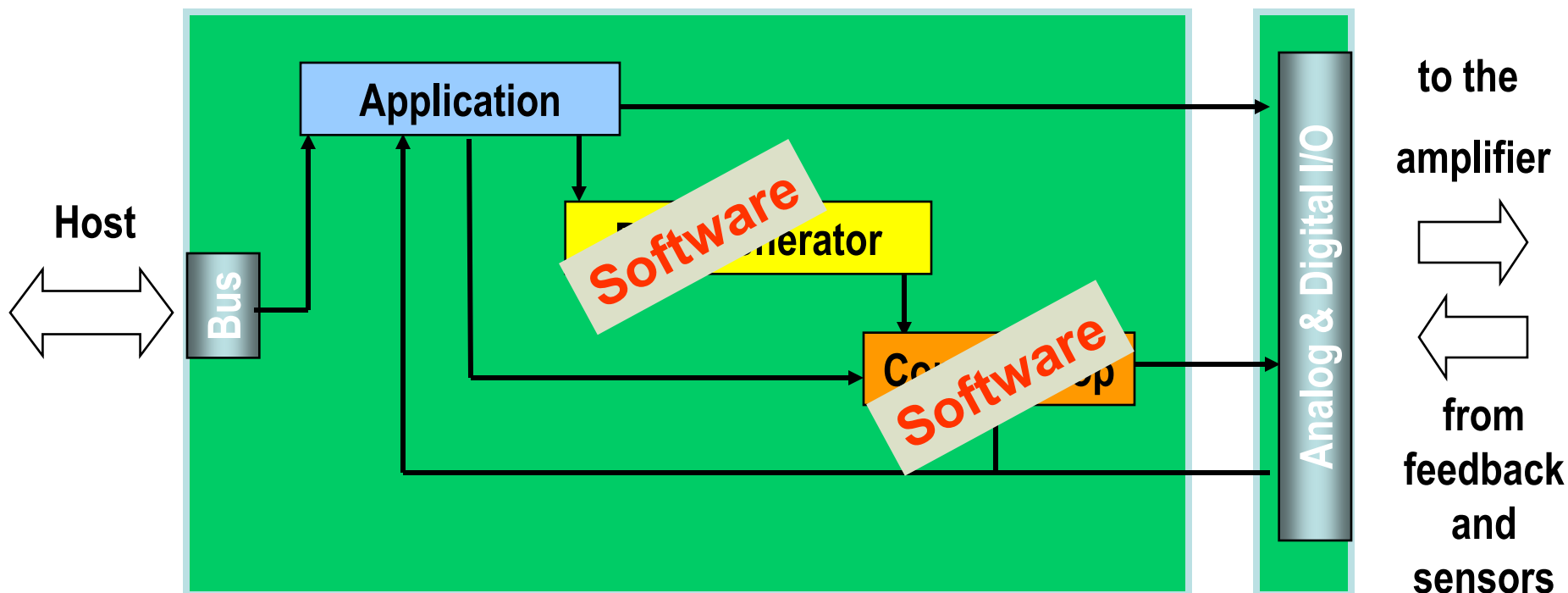
Requirement:

Microcontroller with RTOS, DSPs, FPGA



# NI SoftMotion Controller Architecture

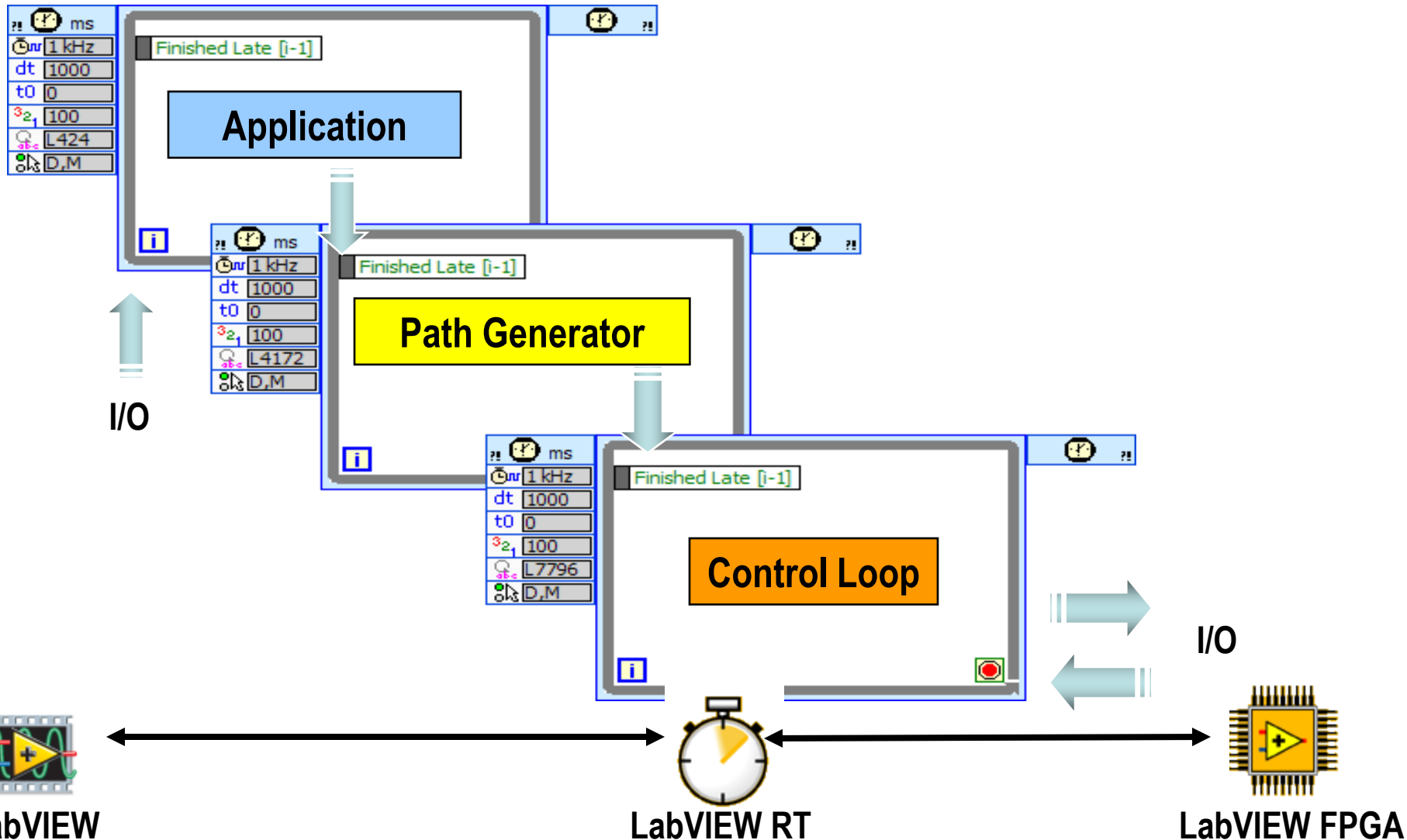
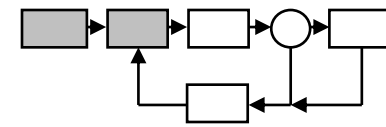
Separates the software from the I/Os



Requirement:

Any CPU with Real-Time Environment

# Motion Loops in LabVIEW



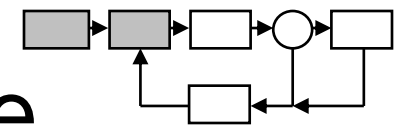
**maxon motor**

driven by precision

Motion under Control

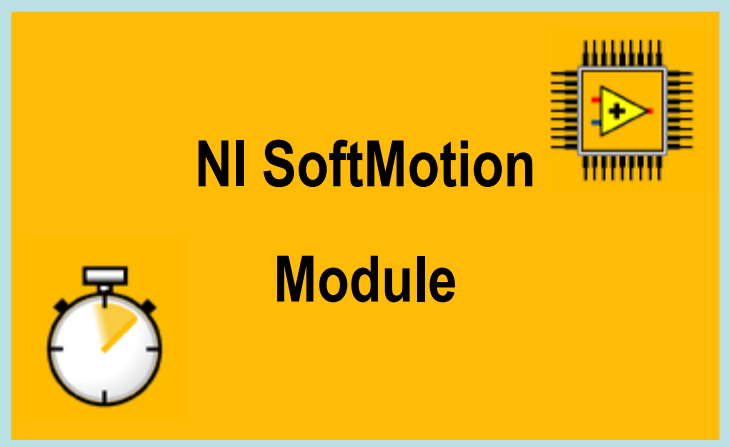
**NATIONAL  
INSTRUMENTS**

# NI SoftMotion Development Module



Software

Hardware



RIO



DAQ



Compact  
FieldPoint

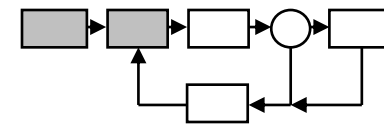
Motion Controller = LabVIEW + NI SoftMotion + I/O

**maxon motor**

driven by precision

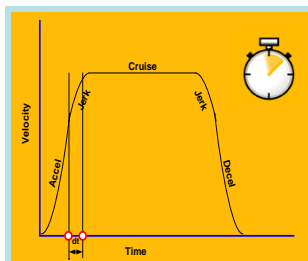
Motion under Control



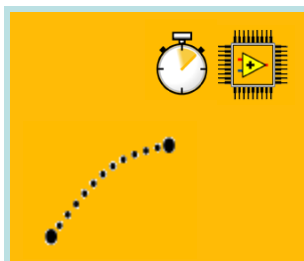


# NI SoftMotion Development Module

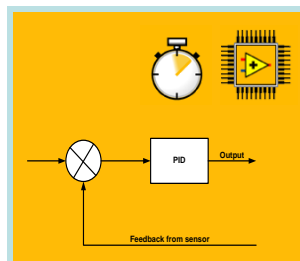
Path Generator



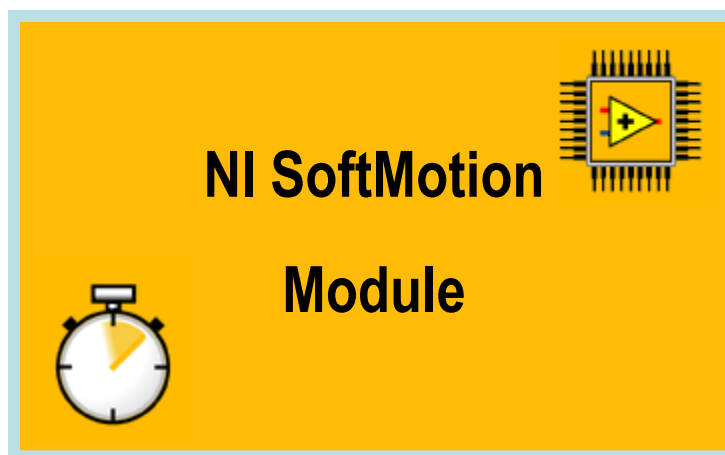
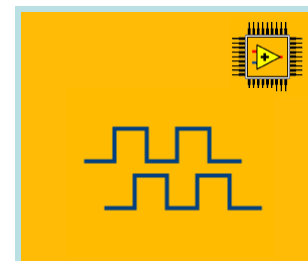
Spline Engine



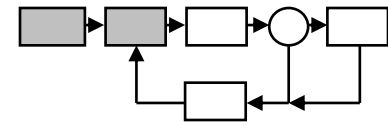
Control Loop\*



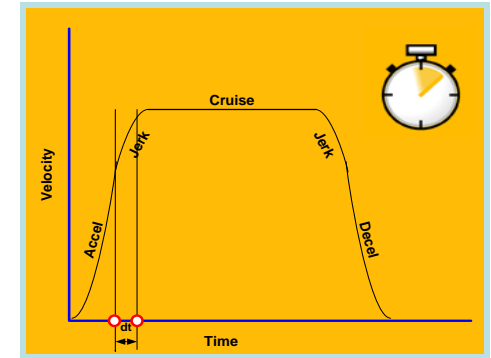
Encoder Decoder\*



# Path Generator



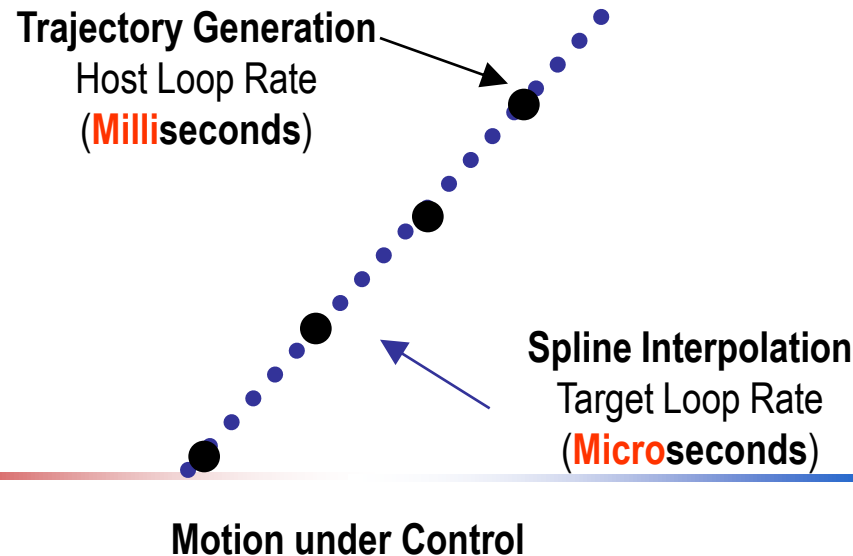
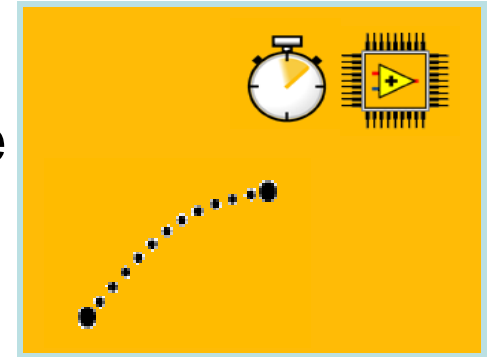
- preemptive 64-bit floating point unit, supports "n" axes (only limited by CPU performance)
- fully supports 64-bit floating point values for position, velocity, acceleration and soft start and braking parameters
- supports circular interpolation (2D, 3D, helix-shaped)
- supports contour mode (start of any geometries)
- supports cubic splining and makes coefficients available for the spline engine





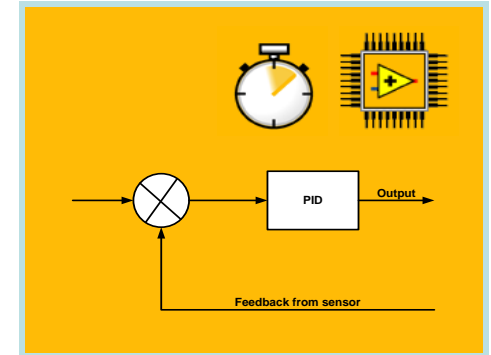
# Spline Engine

- Interpolation routine which uses the spline coefficients of the trajectory generator
- Interpolates missing setpoints for the controller when the loop runs quicker than the trajectory generator
- Can run on a real-time controller or in an FPGA

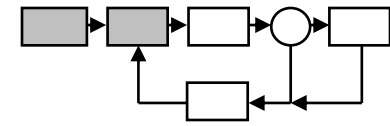


# Loop

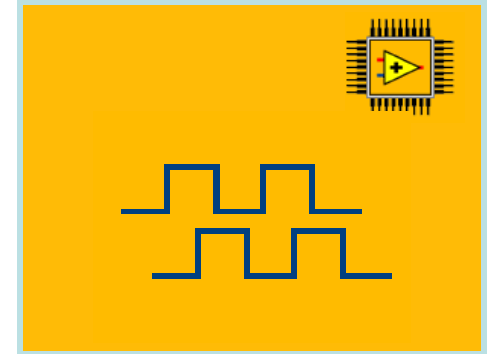
- fixed point upgraded PID algorithm
- supports dual feedback
- supports position and speed control
- supports velocity feedforward and acceleration feedforward to minimize contouring errors
- can run on a real-time controller or in an FPGA
- can run up to 200 kHz (on FPGA hardware)
- source code is open

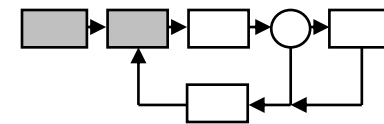


# Rotary Encoder Evaluation

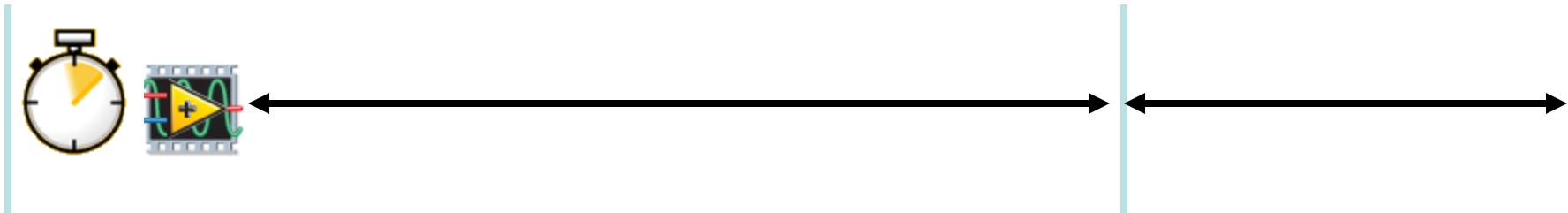
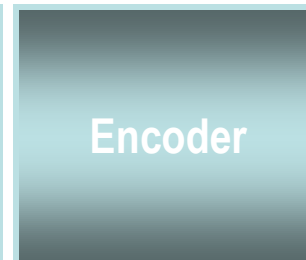
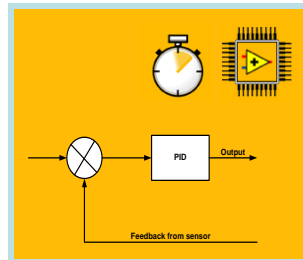
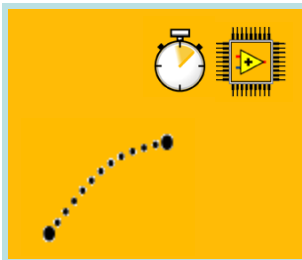
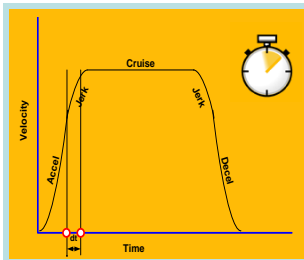


- Code for the evaluation of rotary encoder signals for the application-specific implementation on FPGA hardware
- Calculates velocity and direction (can be executed with up to 20 MHz)
- Source code is open
- With NI SoftMotion basically any feedback sources can be used, e.g. analog signals (LVDTs, Hall sensors, ...), absolute rotary encoder





# NI SoftMotion with Compact FieldPoint



• Compact FieldPoint Controller

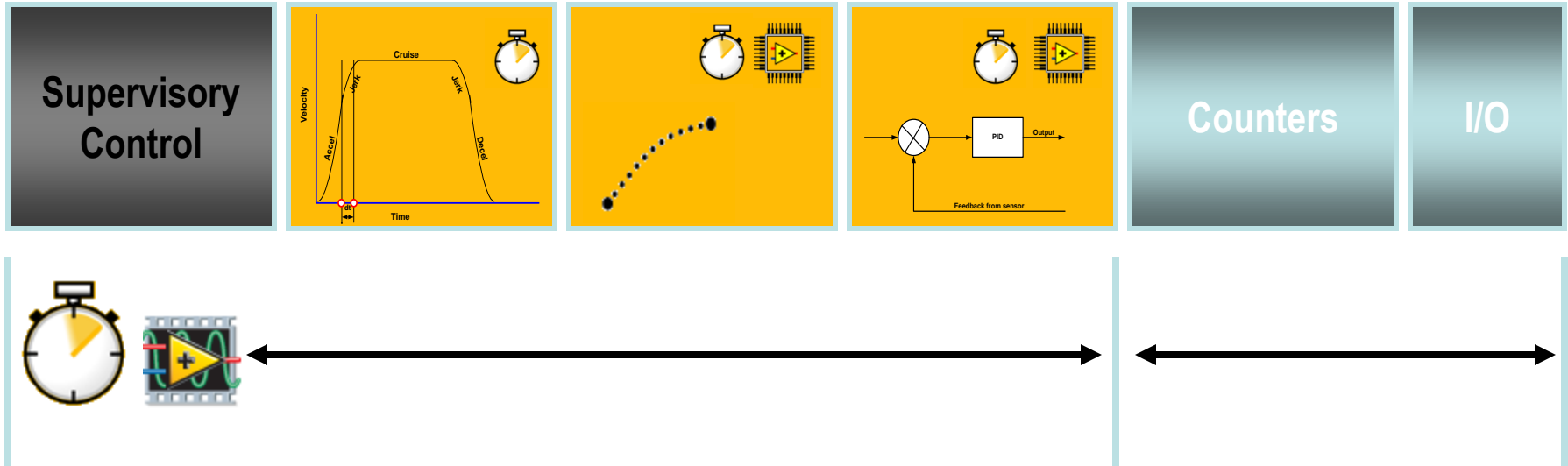
• Compact FieldPoint Modules

Typical loop frequencies:

- *Trajectory Generation* – 100 Hz
- *Control Loop* – 100 Hz



# NI SoftMotion with Data Acquisition Devices



- PXI Chassis/PC

- Data Acquisition M series
- TIO + analog I/O data entry

Typical loop frequencies:

- *Trajectory Generation* – 1 kHz
- *Control Loop* – 1 kHz

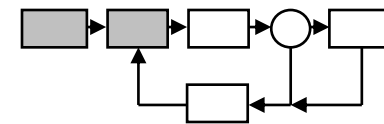


**maxon motor**

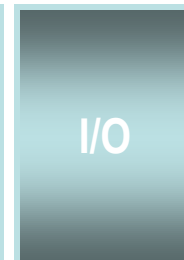
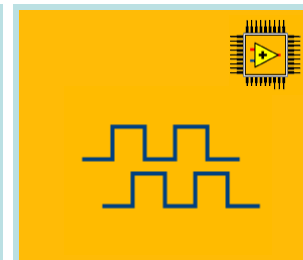
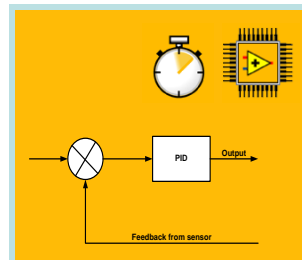
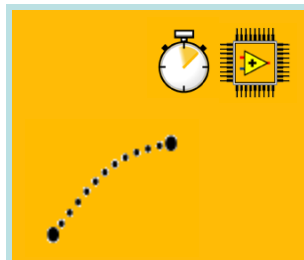
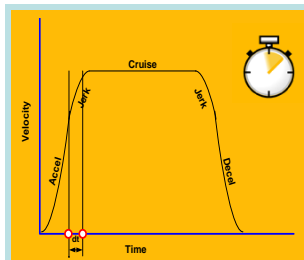
driven by precision

Motion under Control

**NATIONAL  
INSTRUMENTS™**



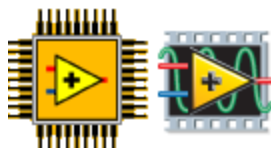
# NI SoftMotion with Reconfigurable I/O



- PXI-Chassis/PC
- CompactRIO Controller

Typical loop frequencies:

- *Trajectory Generation* – 1 kHz
- *Control Loop* – 100 kHz

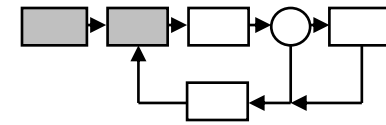


- CompactRIO (I/O modules)
- PXI-7831R/PCI-7831R
- PXI-7811R + CompactRIO of R-series Expansion (I/O modules)

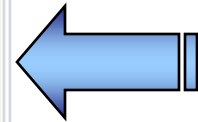
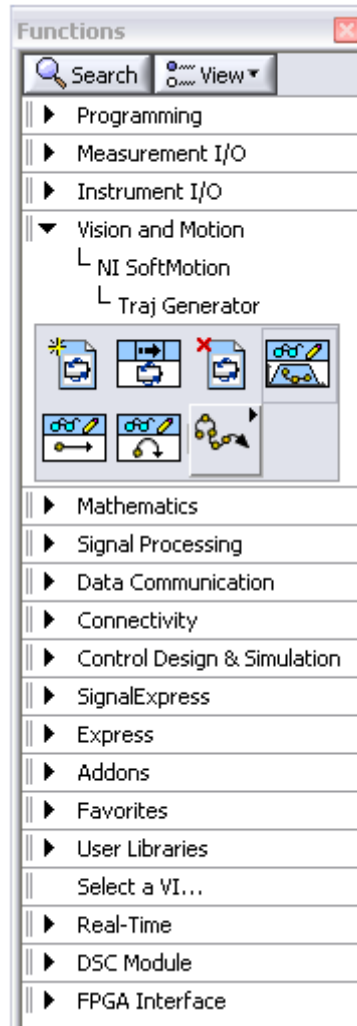
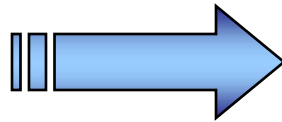


# NI LabVIEW

SoftMotion Library



**NI-SoftMotion  
Functions Palette**



## NI-SoftMotion

- Initializing
- Axis configuration
- Trajectory control
- Start & Stop Motion
- Contouring
- Straight motion
- Circular motion

**maxon motor**

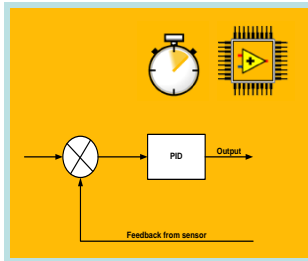
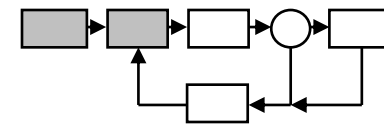
driven by precision

**Motion under Control**

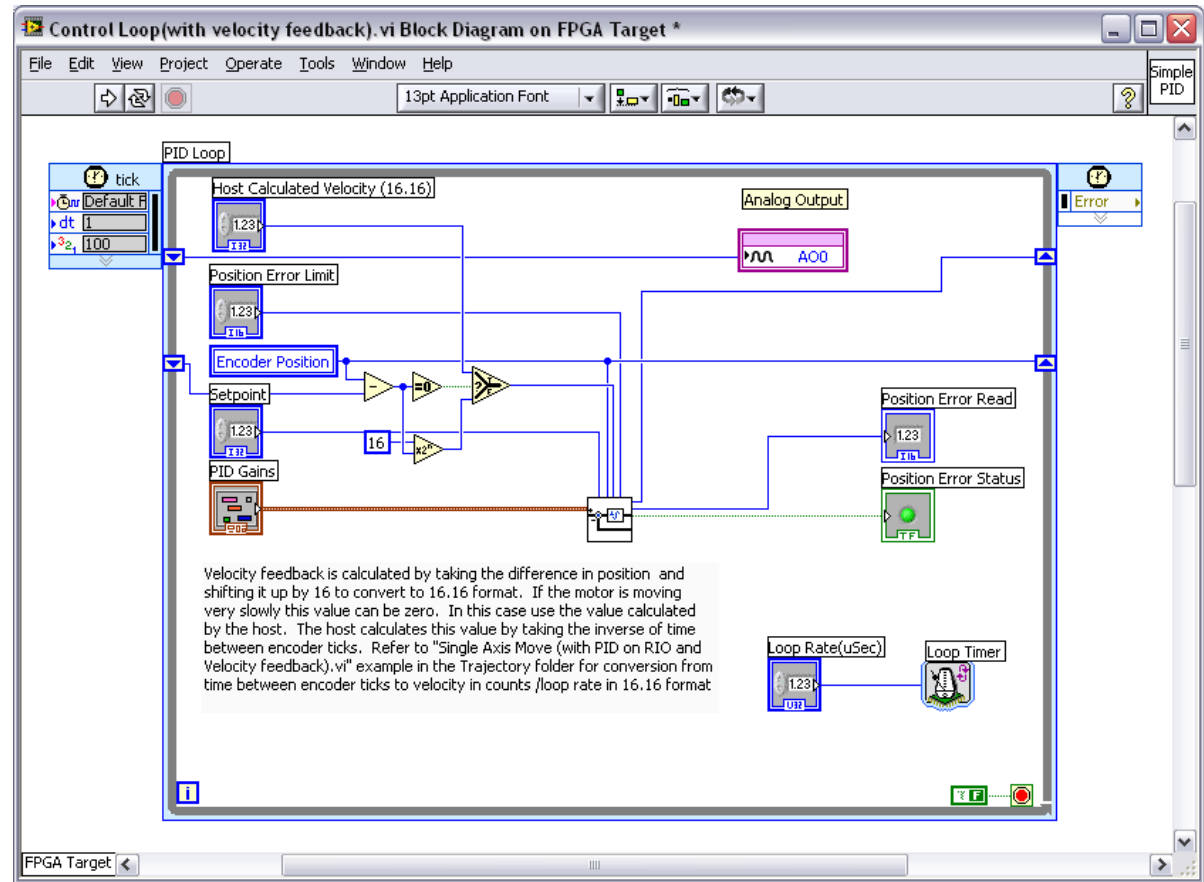


# NI LabVIEW

## Programming Example



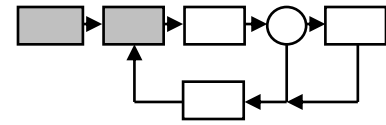
Control Loop\*





# Demo

- CompactRIO Demo



# Comparison

|               | +   | -   |
|---------------|---|---|
| NI-Motion     | <ul style="list-style-type: none"><li>▪ Simple handling</li><li>▪ Cost-effective</li><li>▪ Comfortable tools to start</li><li>▪ Optimized for standard tasks</li></ul>  | <ul style="list-style-type: none"><li>▪ Control encapsulated in firmware</li><li>▪ Only incremental rotary encoder and analog feedback signals</li><li>▪ Control frequency max. 8 kHz</li></ul> |
| NI-CANopen    | <ul style="list-style-type: none"><li>▪ Simple handling</li><li>▪ Cost-effective</li><li>▪ Hardware programming</li></ul>   | <ul style="list-style-type: none"><li>▪ CAN KnowHow needed</li><li>▪ Real-time behavior</li></ul>   |
| NI-SoftMotion | <ul style="list-style-type: none"><li>▪ Hardware-independent architecture</li><li>▪ Free programmability for controls and feedback signals</li><li>▪ Depending on HW very high control frequencies possible</li></ul> | <ul style="list-style-type: none"><li>▪ More complex programming</li><li>▪ Expensive depending on system assembly</li><li>▪ Until now only available for LabVIEW</li></ul>                      |