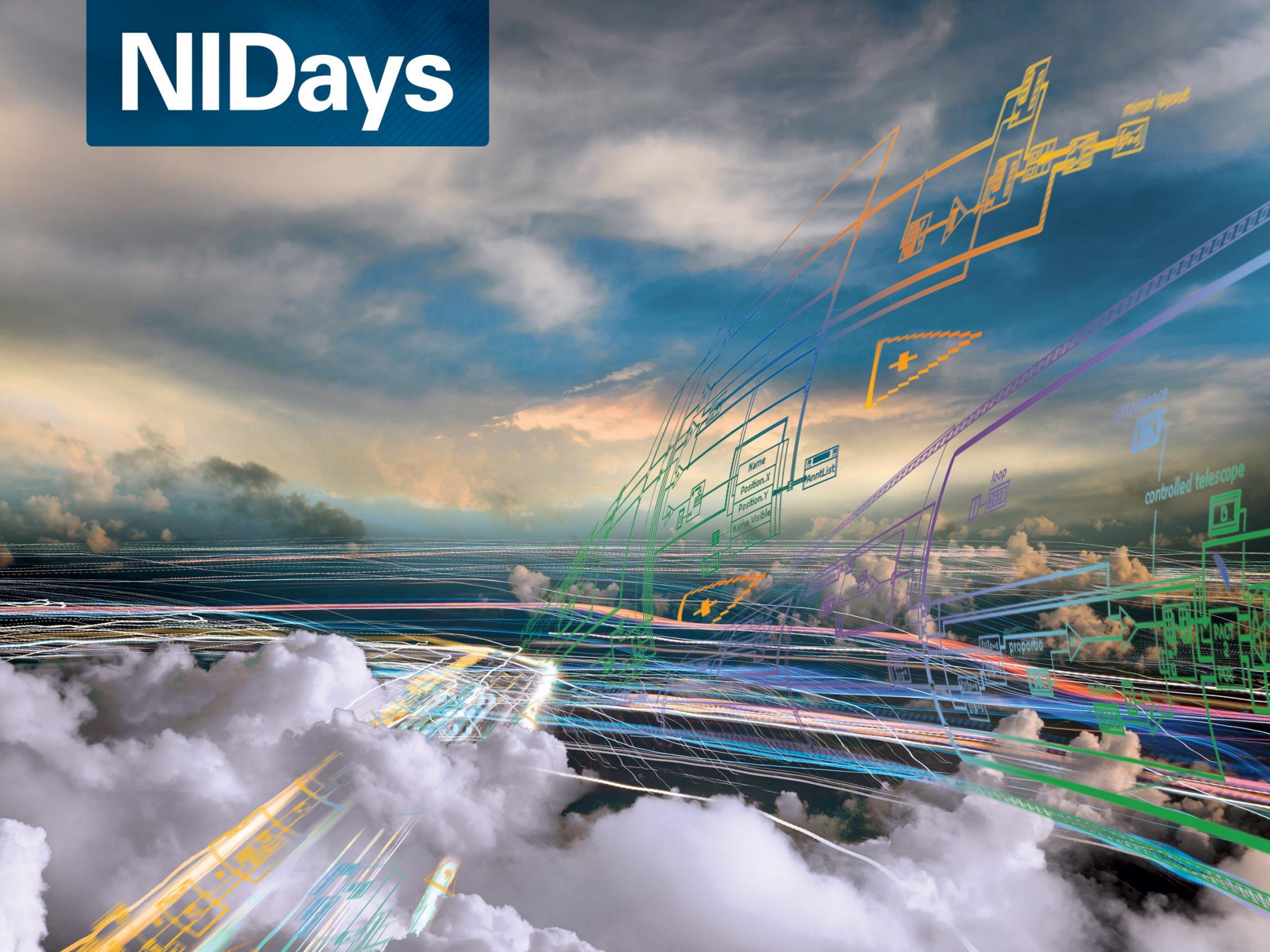
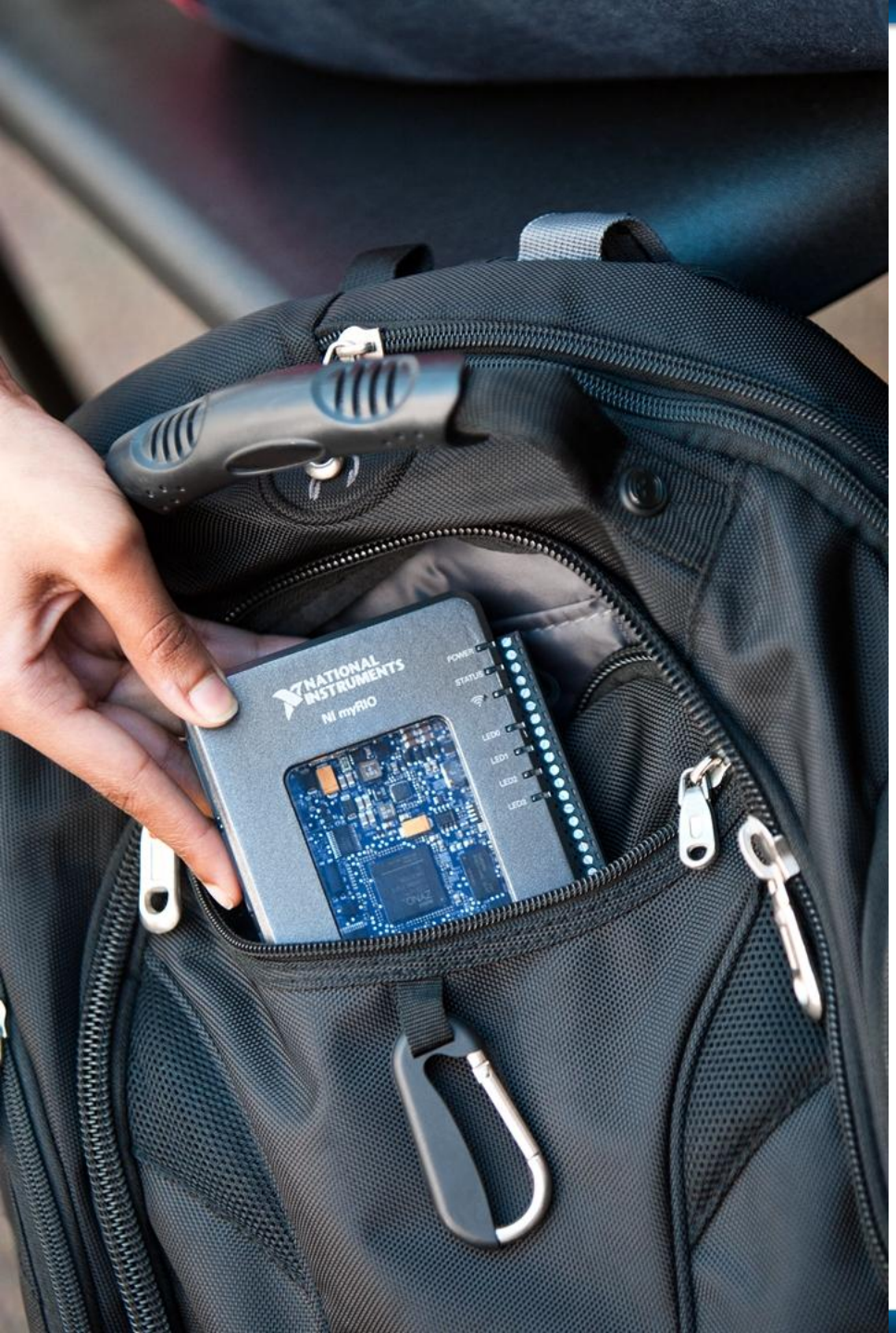

NIDays
Name
Position.X
Position.Y
Name Visible
AnntList
loop
controlled telescope


NATIONAL
INSTRUMENTS™

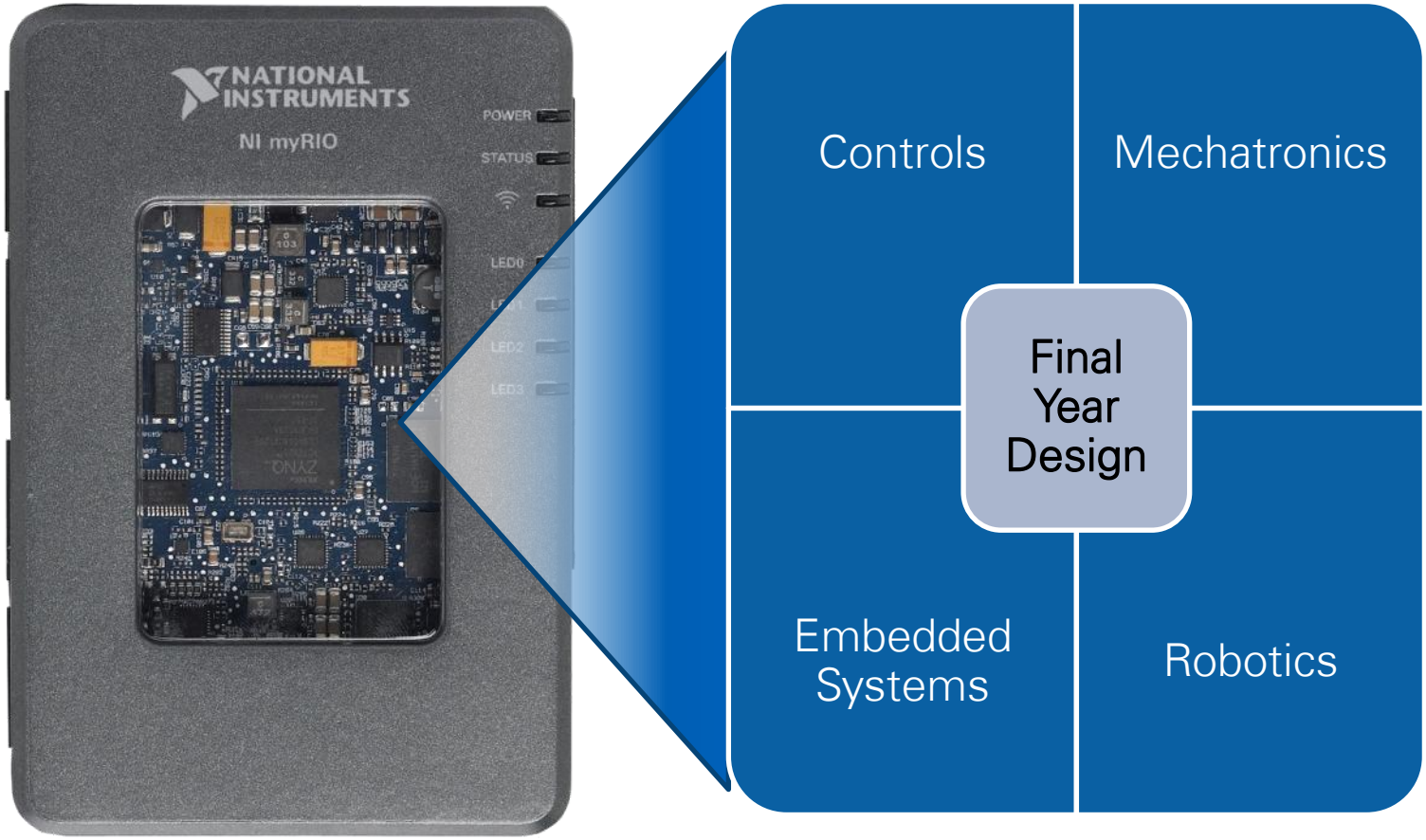


NI myRIO

Design Real Systems, Fast

NATIONAL INSTRUMENTS

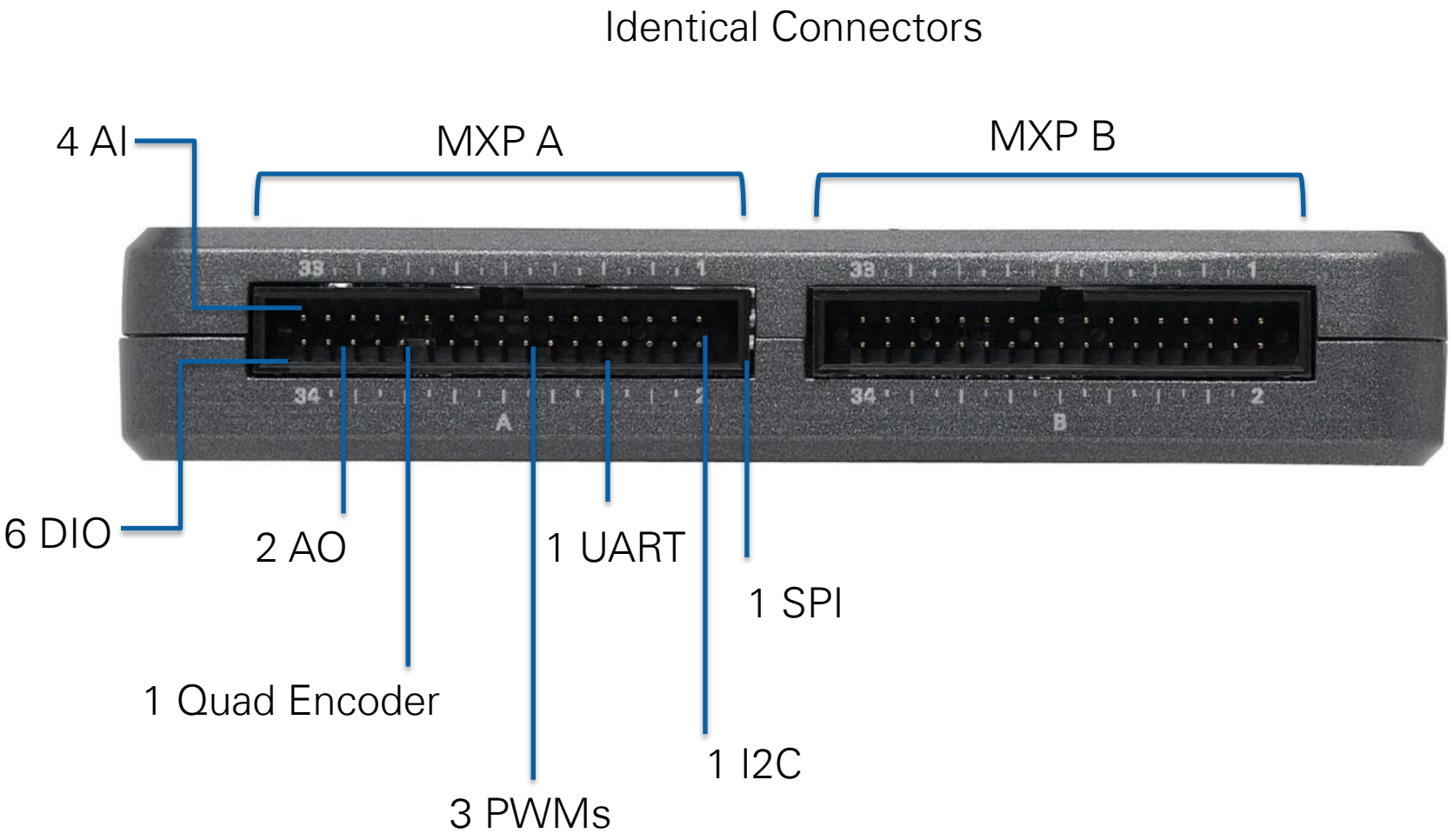
From Core Concepts to System Design



Why Zynq Matters in Education

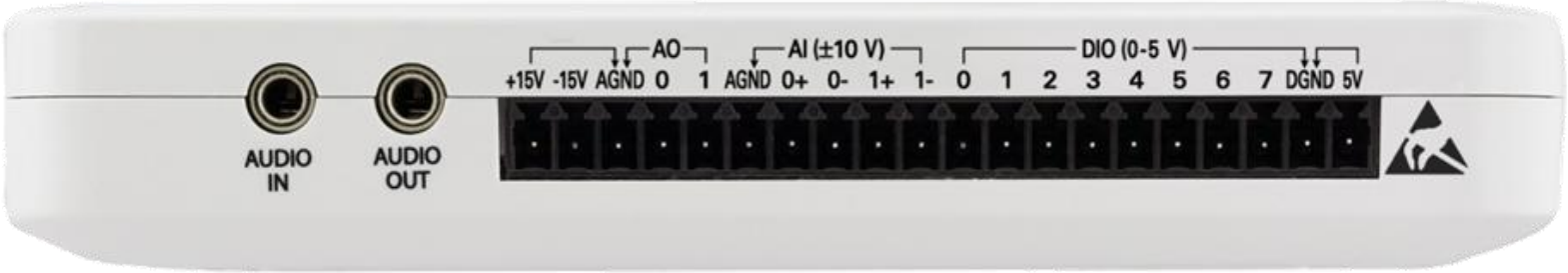
- Smaller Size, Lower Power
- 667 MHz Dual-Core ARM Cortex-A9 Processor
- Artix-7 FPGA, 28k logic cells
- 16 DMA Channels

NI myRIO Expansion Port (MXP)



NI miniSystems Port (MSP)

NI Multisim



What is 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 timing reliability of the system

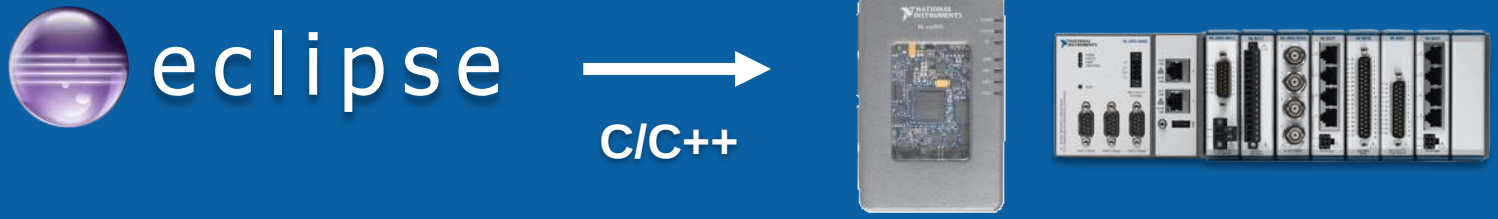
NI Linux Real-Time

- Unlock the vast Linux ecosystem

Database	Security	Code Reuse	Connectivity
Raima	OpenVPN	C/C++	Isshd
MySQL	IP Tables	Shell Scripting	IPv6
SQLite	System Logging	Python	SNMP
MongoDB	fail2ban	Ruby	NTP
CouchDB	denyhost	Perl	netstat

NI Linux Real-Time

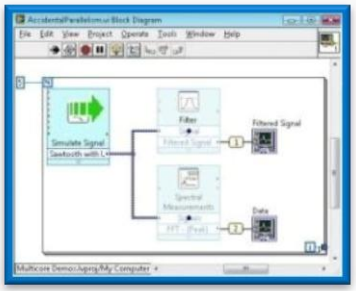
- Reuse C/C++ code in and alongside Real-Time applications built with LabVIEW



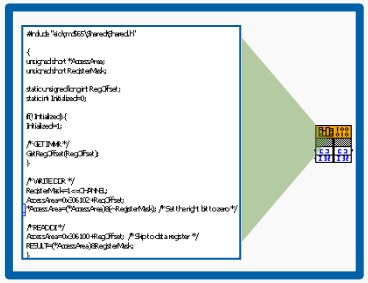
- Program the processor in C/C++ using Eclipse IDE
- A custom distribution of Eclipse is available for FREE download from ni.com
- Examples for NI myRIO provided with download

Models of Computation

Data Flow



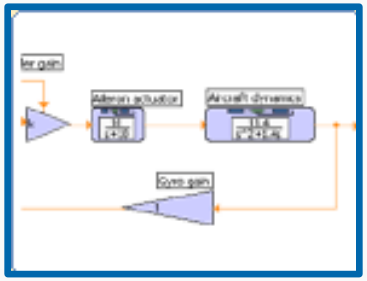
C Code



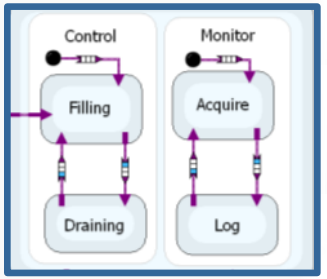
Textual Math

```
1 c = 0.285 + 0.013i;
2 [X Y] = meshgrid(x, y);
3 z = X + i*Y;
4 for k=1:30
5     z = z.^2 + c;
6 end
```

Simulation



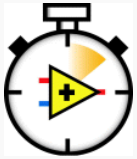
Statechart



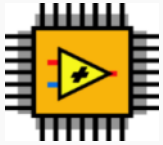
NATIONAL INSTRUMENTS LabVIEW™

Desktop

Real-Time



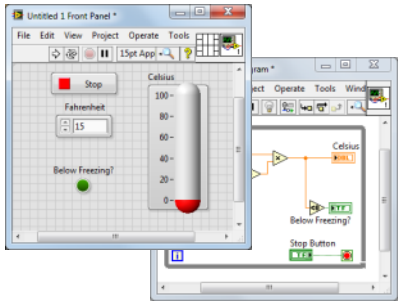
FPGA



Microprocessors

Your Turn

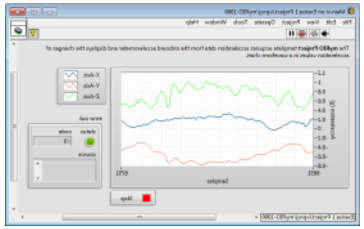
Exercise



(20mins) Optional Exercise 0

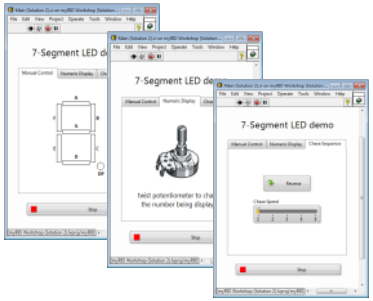
Building a LabVIEW VI in Windows

Optional, but recommended for new LabVIEW users



(10mins) Exercise 1

Create Your First myRIO Project



(30mins) Exercise 2

Create RT code for NI myRIO

Includes challenge exercise for experienced LabVIEW users

Your Turn

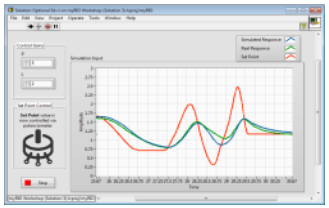
Exercise

Exercise 3

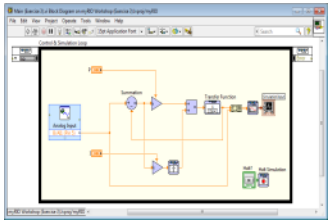
Control System Design with NI myRIO

We have provided a simple PI control algorithm. You will…

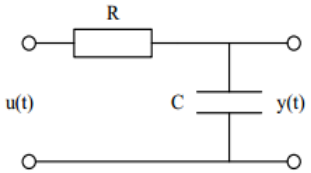
a. [20mins] Create a transfer function to simulate a simple 1st order plant (RC circuit)



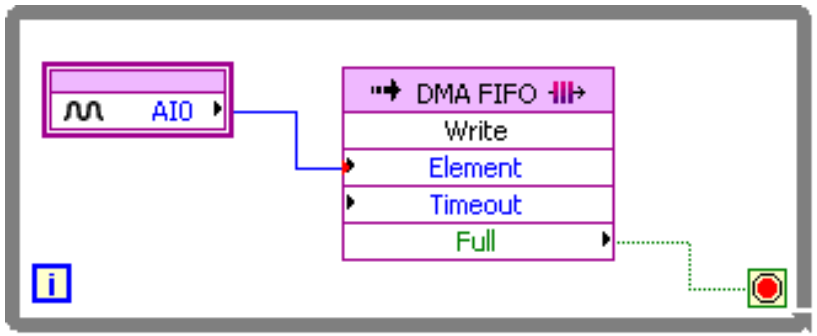
b. [5mins] Deploy the control algorithm and transfer function to myRIO

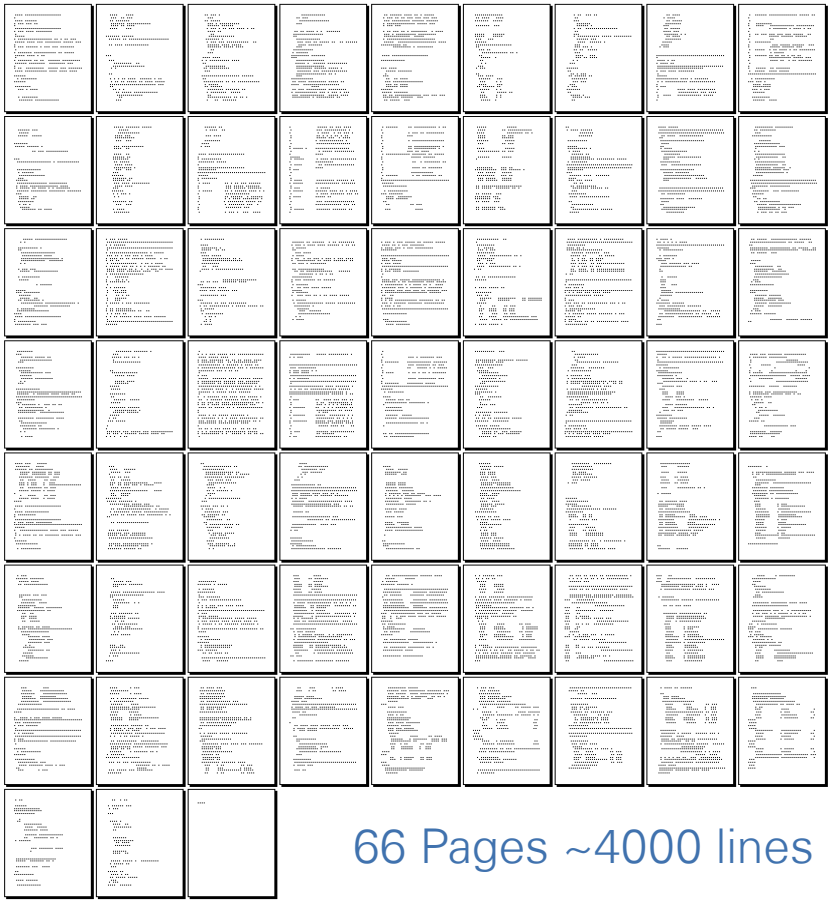


c. [30mins] Move from simulation to the real world, to control a physical RC circuit



LabVIEW FPGA vs. VHDL

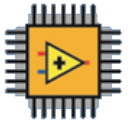




66 Pages ~4000 lines

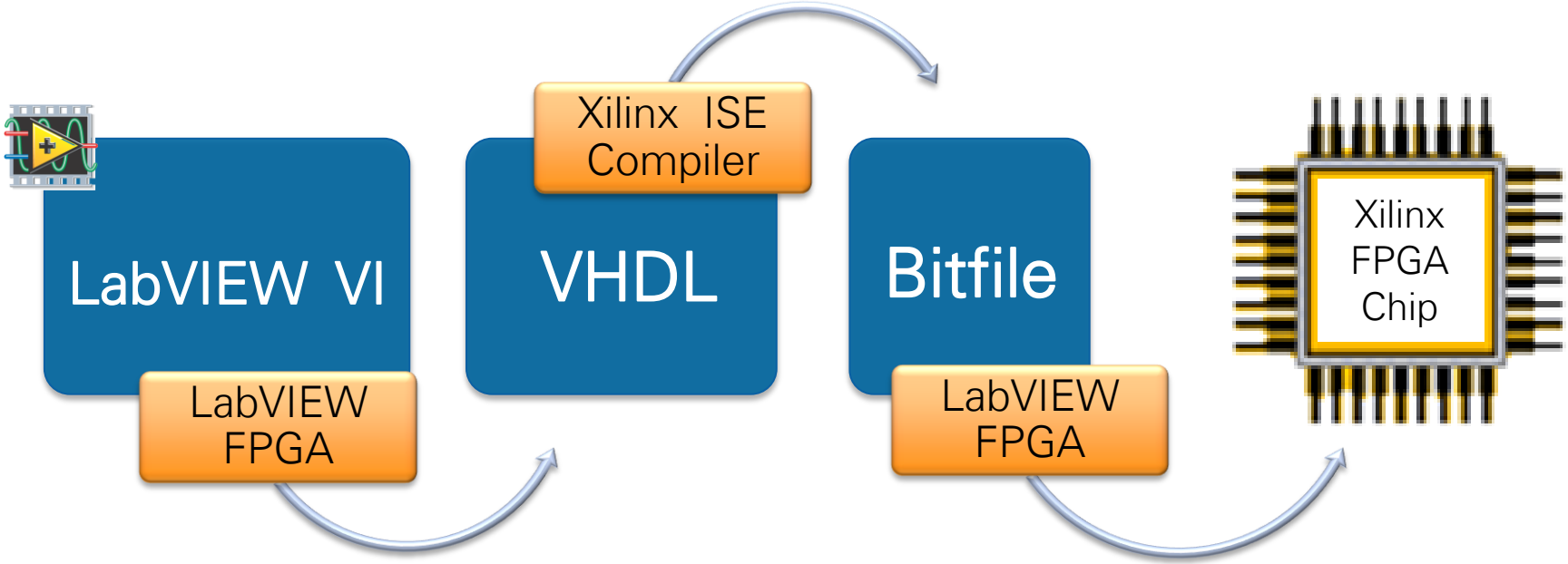
I/O with DMA

Why Are FPGAs Useful?



- **True Parallelism**
  Provides parallel tasks and pipelining
- **High Reliability**
  Designs become a custom circuit
- **High Determinism**
  Runs algorithms at deterministic rates down to 25 ns (faster in many cases)
- **Reconfigurable**
  Create new and alter existing task-specific personalities

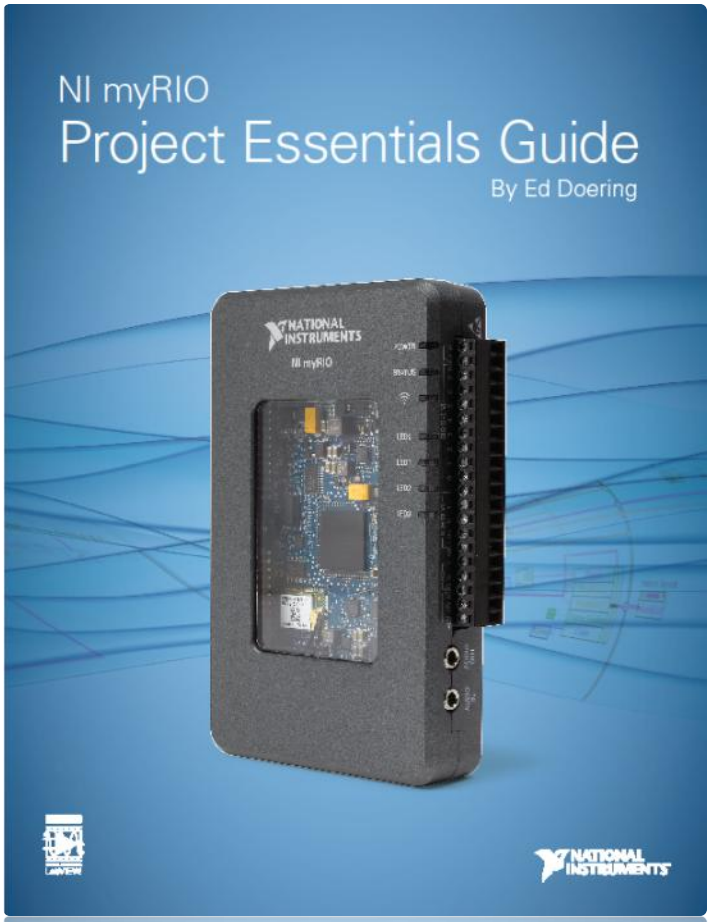
LabVIEW FPGA: How does it work?



Learn More About Programming NI myRIO

ni.com/learn-myrio

NI myRIO | Courseware



2 Discrete LED

LEDs, or light-emitting diodes, provide simple yet essential visual indicators for system status and error conditions. Figure 2.1 shows the four types of LEDs included in the SparkFun "LED Mixed Bag (5mm)" kit `http://www.sparkfun.com/products/9881`.

Learning Objectives: In this module you will create a standard interface circuit to verify correct operation of the LED, learn interface circuit design principles and related LabVIEW programming techniques, make some basic modifications to extend your understanding of the interface, and then challenge yourself to design a system that integrates the discrete LED with additional components or devices.

2.1 Component Verification

Follow these steps to verify correct operation of the discrete LED component.

Select these parts:

- Resistor, 220 ohm
- "Basic Red" LED from Sparkfun 9881
- Breadboard
- Connecting wires [need details]

ownload the LabVIEW project: Download the roject `Discrete LED demo.lvproj` from eed details].

2.3. BASIC MODIFICATIONS

Figure 2.2: Discrete LED verification circuit: schematic diagram, recommended breadboard layout, and connection to NI myRIO MXP Connector B.

NATIONAL INSTRUMENTS

NI myRIO Kits | ni.com/myrio

Starter

LEDs & switches
7-segment display
Potentiometer
Thermistor
Photo resistor
Hall effect
Microphone/Speaker
Battery holder
DC motor

Mechatronics

DC gear motors/encoders
H-bridge driver
Accelerometer
Triple-axis gyro
Infrared proximity sensor
Ambient light sensor
Ultrasonic range finder
Compass
Hobby servo motors

Embedded

RFID reader kit
Numeric keypad
LED matrix
Digital potentiometer
Character LCD
Digital temp sensor
EEPROM

C Support for NI myRIO – ni.com/myrio/c-support

NATIONAL INSTRUMENTS

Contact Us | United States

NIWeek

Products & Services | Solutions | Support | Developer Zone | Academic | Events | Company | Search

My Profile | MyNI | Notifications | Parts List | Cart

Home > Developer Zone > Tutorial > C Support for NI myRIO

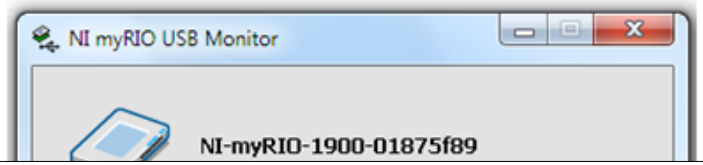
C Support for NI myRIO

Publish Date: Jun 18, 2013 | 0 Ratings | **0.00** out of 5 | PDF | Submit your review |

Overview

NI myRIO is an embedded hardware device designed specifically to help students design re
quickly and affordably than ever before. NI myRIO is based on NI reconfigurable I/O (RIO) tec
program both a processor running a real-time OS and a customizable FPGA. In addition to N
processor is fully programmable in C or C++ using the default shipping personality placed o
using only the LabVIEW FPGA Module. Follow the steps below to start programming the NI n

1. Install the Eclipse integrated development environment (IDE) and compile tools.
2. Download and install the LabVIEW for myRIO Module. This module provides you with th
well as some software utilities, which offer useful information about your device.
3. Download the NI myRIO C Examples and Documentation. The NI myRIO C examples ar
supported if you use the tools provided in Step 1.
4. Connect power to the NI myRIO device.
5. Connect the USB cable from NI myRIO to your development computer.
6. The **NI myRIO USB Monitor** appears and displays the IP address of the NI myRIO device



- Processor Programming
- Eclipse IDE
- C/C++ Support
- Free
- One Installer from ni.com

Thank you!

Where to go next?

- Get additional resources on LabVIEW
  www.ni.com/labview
- Learn about NI and Data Acquisition
  www.ni.com/daq
- Enroll for a instructor-led LabVIEW class
  www.ni.com/training
- Attend other free seminars and workshops
  http://serbia.ni.com/dogadjaji
- Contact our technical consultant and get a free consultation of your project and potential solutions!
  Toll-free number: 06 80 204 704

Contact us

National Instruments,
Instrumentacija, avtomatizacija in upravljanje procesov d.o.o.
Kosovelova 15
SI 3000 Celje, Slovenia

Tel: + 386 3 425 4270
Fax: +386 3 425 4212
Email: ni.serbia@ni.com
Web: http://serbia.ni.com