

Reconfigurable measurement systems based on real time and FPGA

Ján Šaliga
Technical university of Košice, Slovakia
(based on different sources mainly from National Instruments)

Modern instrumentation and control – nowadays systems

- **Typical nowadays system:**
 - Low cost instruments (system): microcontroller with fix firmware and fix hardware – no flexibility
 - Quality instruments: more-less PC with signal inputs and output (signal front – end: analog preprocessing circuit AD and DA converters, digital/timing IO = fixed hardware)
 - Flexibility only in software: General purpose OS (e.g., W7) + flexible application software (measurement functions)
 - Large, high energy consumption, limited flexibility given by software and fixed hardware, problematic in real time signal processing, exact and fast timing (multitask in windows), and similar applications
 - Easy maintained (software development and installation), multitask, data presentation and archiving.
- **Typical convenient applications:**
 - Single purpose instrumentation (oscilloscope, basic spectrum analyzers, logic and network analyzers, etc.)
- **Typical inconvenient applications:**
 - Distributed systems (sensor network), control systems of machines, robust and reliable systems continuously working for a long time, fast signal processing in real time, reconfigurable measurement system (reconfigurable signal and pattern generation, signal analyzers, etc. that cannot be processed by software in **real time**)

Summer School on Marine Technology Instrumentation, UPC
Barcelona, Spain, July 2014

2

What is a Real-Time System?

Real Time Response - The ability to reliably and, without fail, respond to an event or perform an operation within a guaranteed time period.

Loop Cycle Time - Execution time of one cycle of a loop.

Jitter - Variation from the desired loop cycle time.

Summer School on Marine Technology Instrumentation, UPC
Barcelona, Spain, July 2014

3

Real-Time Terms

Determinism – Describes how consistently the system responds to external events or performs operations within a given time limit.

Time Critical Code - Code that must execute on a specific schedule to function as desired

Priority - Defines when a VI or loop should execute relative to other VIs and loops

Summer School on Marine Technology Instrumentation, UPC
Barcelona, Spain, July 2014

4

Real time operational systems

- In general, real time systems are based on real time OS (RTOS)
- What is RTOS?
 - Any OS is responsible for managing the hardware resources of a computer and hosting applications that run on the computer
 - RTOS is specially designed to run applications with very precise timing and a high degree of reliability
 - To be considered "real-time", an operating system must have a known maximum time for each of the critical operations that it performs



Summer School on Marine Technology Instrumentation, UPC
Barcelona, Spain, July 2014

5

General Purpose Operating Systems

- Multitasking = Processor time is shared between programs
- Can preempt high priority programs
 - Many programs run in the background—screen savers, disk utilities, antivirus software, etc.
 - Must service interrupts—keyboard, mouse, Ethernet, etc.
- High amount of jitter
- Cannot guarantee determinism

Summer School on Marine Technology Instrumentation, UPC
Barcelona, Spain, July 2014

6

Real-Time Operating Systems

- * Multitasking - Ensure that high-priority tasks execute first
- * Generally do not require user input from peripherals

LabVIEW Real-Time Module executes VIs on the following real-time operating systems:

- * NI ETS (Embedded Tool Suite)
- * Wind River VxWorks
- * Linux

Summer School on Marine Technology Instrumentation, UPC
Barcelona, Spain, July 2014

7

RTOS vs. general purpose OS (GPOS)

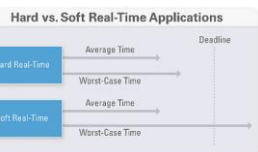
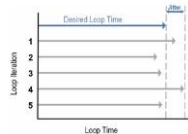
- * GPOS such as Windows is excellent platform for developing and running your non-critical measurement and control applications but not the ideal platform for running applications that require precise timing or extended up-time.
- * **Priorities:**
 - * GPOS is fair – all running tasks (threads) will achieve a processor time
 - * RTOS behavior is given by programmer decision on tasks' priorities (static, dynamic) and on environment requirements
- * **Interrupts latency:** - measured as the amount of time between when a device generates an interrupt and when that device is serviced:
 - * GPOS: variable amount of time
 - * RTOS must guarantee that all interrupts will be serviced within a certain maximum amount of time (interrupt latency must be bounded)
- * **Performance** – actual performance is given mostly by hardware
 - * GPOS – virtually decreased by multitasking
 - * RTOS – can provide much more precise and predictable timing characteristics

Summer School on Marine Technology Instrumentation, UPC
Barcelona, Spain, July 2014

8

Hard and Soft RTOS

- * RTOS that can absolutely guarantee a maximum time for operations are commonly referred to as “hard real-time”
- * RTOS that can only guarantee a maximum most of the time are referred to as “soft real-time”



Summer School on Marine Technology Instrumentation, UPC
Barcelona, Spain, July 2014

9

RT system components

- * Components of RT system:

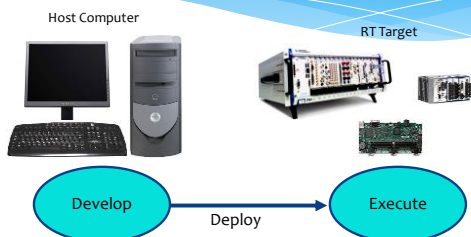


- * When I need RT system?
 - * If I need a precise timing:
 - * Perform tasks within a guaranteed worst-case timeframe
 - * Carefully prioritize different sections of my program
 - * Run loops with nearly the same timing each iteration (typically within microseconds)
 - * Detect if a loop missed its timing goal
 - * If I need a high reliability:
 - * System should run reliably for days, months, or years without stopping (watchdog system may be useful)

Summer School on Marine Technology Instrumentation, UPC
Barcelona, Spain, July 2014

10

LabVIEW Real-Time System



Summer School on Marine Technology Instrumentation, UPC
Barcelona, Spain, July 2014

11

Real-Time System Components: Real-Time Targets



NI RT PXI Embedded Controllers
High speed, high channel density, I/O variety



NI CompactRIO
Reconfigurable Embedded System



NI Compact Vision System & NI Smart Camera
Machine vision



NI Industrial Controller
Rugged, fanless, high-performance



NI Single-Board RIO
Reconfigurable single circuit board form factor



Desktop PC running RTOS
Determinism for PCI systems

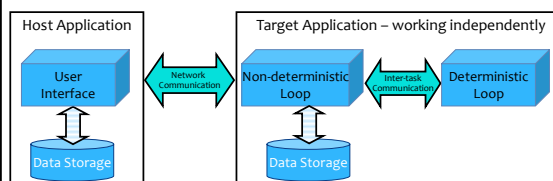


NI myRIO Education

Summer School on Marine Technology Instrumentation, UPC
Barcelona, Spain, July 2014

12

Host and Target Application Architecture



Summer School on Marine Technology Instrumentation, UPC
Barcelona, Spain, July 2014

13

Host Application

Runs on the host computer

Handles non-deterministic tasks

- * Communicates with the target application
 - user interface parameters and data retrieval
- * Data logging
- * Data analysis
- * Data broadcast to other systems

Host Application

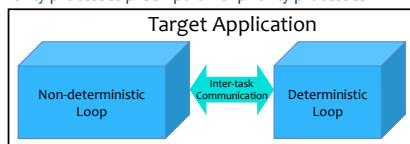


Summer School on Marine Technology Instrumentation, UPC
Barcelona, Spain, July 2014

14

Target Application

- * Processes that requires determinism are time-critical processes—set all other processes to a lower priority
- * Use multithreading to set the priority of a process
- * Higher priority processes preempt lower priority processes



Summer School on Marine Technology Instrumentation, UPC
Barcelona, Spain, July 2014

15

FPGA in instrumentation

Summer School on Marine Technology Instrumentation, UPC
Barcelona, Spain, July 2014

16

FPGA

- * Field-programmable gate arrays (FPGAs) are reprogrammable silicon chips.
- * In contrast to processors that you find in your PC, programming an FPGA rewires the chip itself to implement your functionality rather than run a software application.
- * Ross Freeman, the cofounder of Xilinx, invented the first FPGA in 1985.



Summer School on Marine Technology Instrumentation, UPC
Barcelona, Spain, July 2014

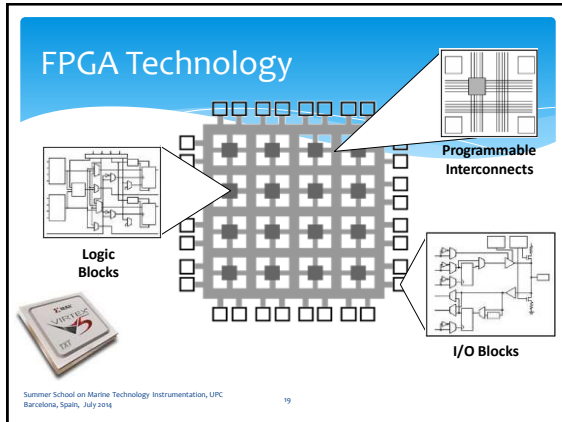
17

Top 5 Benefits of Using FPGAs

- * FPGA chip adoption across all industries is driven by the fact that FPGAs combine the best parts of application-specific integrated circuits (ASICs) and processor-based systems.
- * These benefits include the following:
 - * Faster I/O response times and specialized functionality
 - * Exceeding the computing power of digital signal processors
 - * Rapid prototyping and verification without the fabrication process of custom ASIC design
 - * Implementing custom functionality with the reliability of dedicated deterministic hardware
 - * Field-upgradable eliminating the expense of custom ASIC re-design and maintenance

Summer School on Marine Technology Instrumentation, UPC
Barcelona, Spain, July 2014

18



Configurable logic blocks (CLBs)

- * Basic blocks
 - * flip-flops
 - * lookup tables (LUTs)
- * The basic block can be combine in different FPGA into more complex blocks:
 - * Digital signal processors (DSP) slices
 - * Embedded block of RAM
 - * Multipliers
 - * Etc.

Note: Most applications do not require detailed knowledge of these components

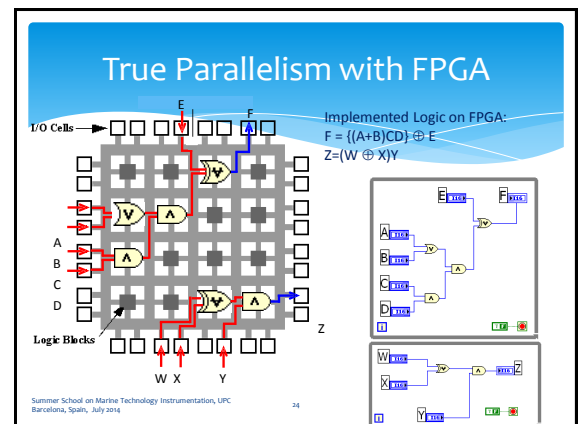
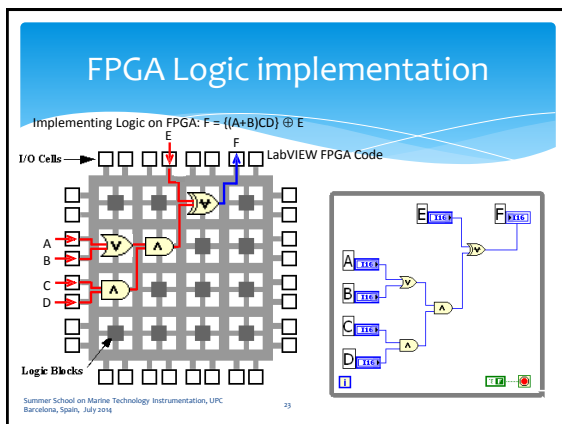
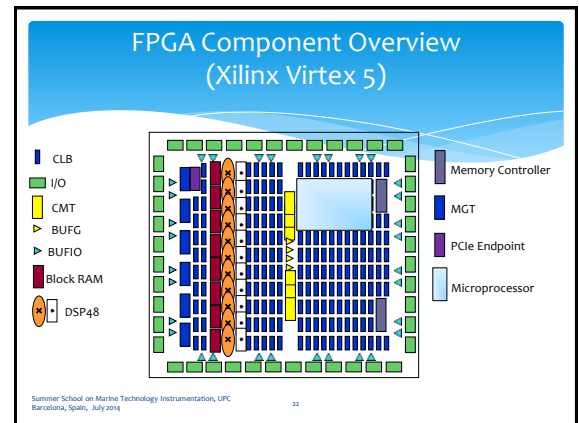
Summer School on Marine Technology Instrumentation, UPC
Barcelona, Spain, July 2014

Look Up Table (LUT)

- * Basic FPGA Building Block
- * Common uses
 - * Logic functions
 - * Distributed memory
 - * Shift registers

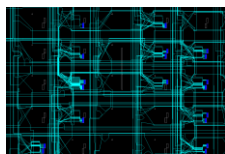
Inputs	And2
0000	0
0001	0
0010	0
0011	0
0100	0
0101	0
0110	0
0111	0
1000	0
1001	0
1010	0
1011	0
1100	0
1101	0
1110	0
1111	1

Summer School on Marine Technology Instrumentation, UPC
Barcelona, Spain, July 2014

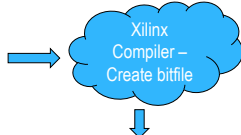


FPGA Hardware to Bitfile

Function Nodes – Case Study



Schematic



Bitfile

```
001010011010001010
10101001001010101
001001010101011101
001010101010001010
101001010101010101
010010010010010000
```

Summer School on Marine Technology Instrumentation, UPC
Barcelona, Spain, July 2014

31

High-Level Synthesis Design Tools

- * Simplification of design – no HDL is needed
- * Verification in user friendly environment known from software development and debugging.
- * Example for projects: LabVIEW FPGA



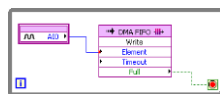
Summer School on Marine Technology Instrumentation, UPC
Barcelona, Spain, July 2014

32

Abstraction to the Pin



VHDL



LabVIEW FPGA

Summer School on Marine Technology Instrumentation, UPC
Barcelona, Spain, July 2014

33

LabVIEW, LabVIEW Real Time, LabVIEW FPGA

Summer School on Marine Technology Instrumentation, UPC
Barcelona, Spain, July 2014

34

Graphical System Design

A Platform-Based Approach

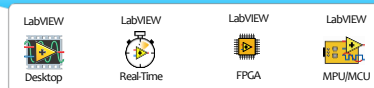
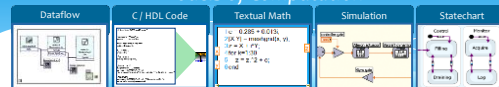


Summer School on Marine Technology Instrumentation, UPC
Barcelona, Spain, July 2014

35

System Design to Deployment

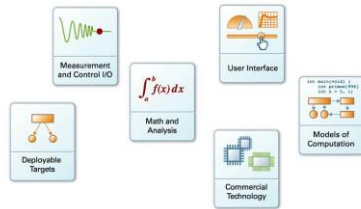
Models of Computation



Summer School on Marine Technology Instrumentation, UPC
Barcelona, Spain, July 2014

36

Integrating Elements



Summer School on Marine Technology Instrumentation, UPC
Barcelona, Spain, July 2014

37

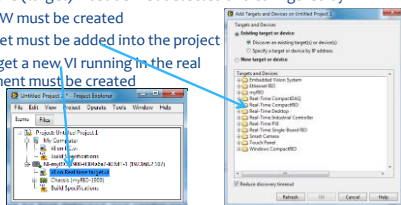
LabVIEW Real time notes

Summer School on Marine Technology Instrumentation, UPC
Barcelona, Spain, July 2014

38

Preparation

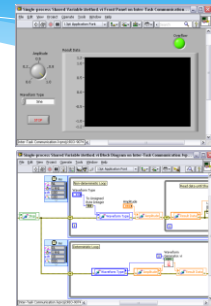
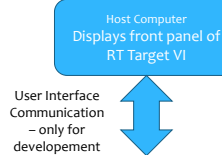
- Real time application is always developed on PC
- Real time hardware (target) must be first detected and configured by MAX
- Project in LabVIEW must be created
- Real time target must be added into the project
- Under the target a new VI running in the real time environment must be created



Summer School on Marine Technology Instrumentation, UPC
Barcelona, Spain, July 2014

39

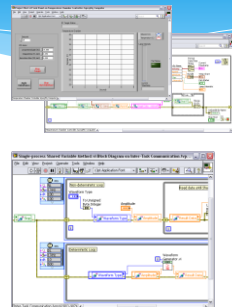
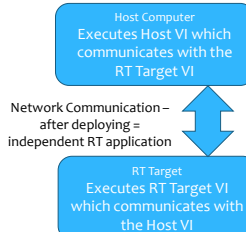
A. Front Panel Communication



Summer School on Marine Technology Instrumentation, UPC
Barcelona, Spain, July 2014

40

B. Network Communication



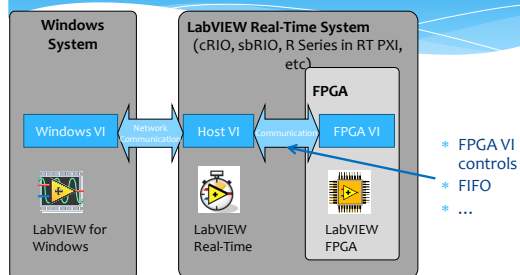
Summer School on Marine Technology Instrumentation, UPC
Barcelona, Spain, July 2014

41

Network Communication Comparison

Protocol	Common Use	Speed	Deterministic Data Transfer	Deterministic Data Transfer	Advantages	Caveats
Network-Published Shared Variable	Latest value, host interface	Fast	With RT FIFO enabled	No	Ease of programming	LabVIEW only
Network Streams	Data streaming	Faster	No	No	Built-in functions	LabVIEW only
TCP	Data streaming	Fastest	No	No	High transfer rates, standard protocol	String data
UDP	Broadcast latest values	Fastest	No	No	High transfer rates, standard protocol	String data, Lossy

FPGA – Real-Time



Summer School on Marine Technology Instrumentation, UPC
Barcelona, Spain, July 2014

49

Example: cRIO, sbRIOmyRIO

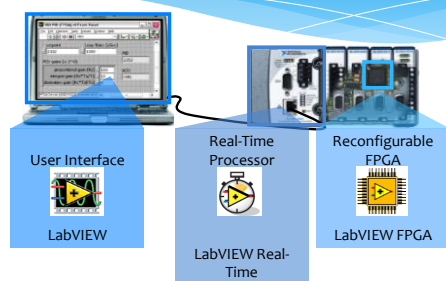
- * cRIO: Chassis with FPGA + RT processor module or embedded + slots for DAQ modules
- * sbRIO: single board cRIO with integrated DAQ modules
- * myRIO: educational module similar to sbRIO



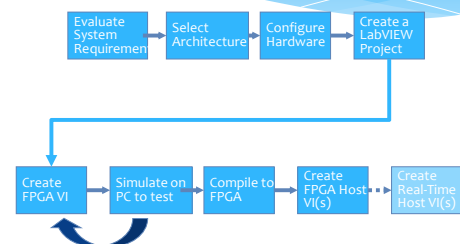
Summer School on Marine Technology Instrumentation, UPC
Barcelona, Spain, July 2014

50

LabVIEW Programming & CompactRIO or myRIO



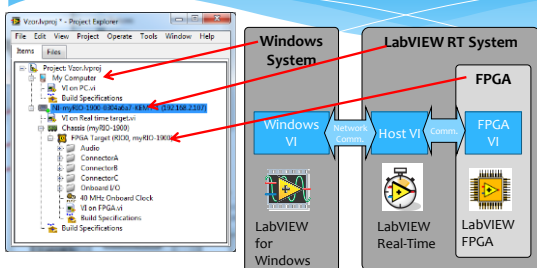
FPGA Development Flow



Summer School on Marine Technology Instrumentation, UPC
Barcelona, Spain, July 2014

51

Project with CompactRIO Controller



Summer School on Marine Technology Instrumentation, UPC
Barcelona, Spain, July 2014

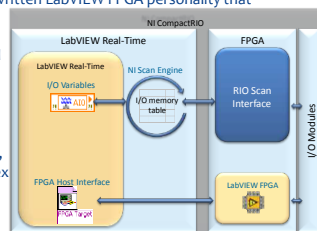
53

IO access

IO are connected to processor across FPGA

IO access:

1. The „Scan Engine” is a prewritten LabVIEW FPGA personality that automatically reads values from I/O modules (myRIO up to 10kHz rate) – no need to program FPGA.
2. Own personality (VI configuring FPGA and optionally including digital signal preprocessing, control, timing, fast complex calculations, etc.).



Summer School on Marine Technology Instrumentation, UPC
Barcelona, Spain, July 2014

54

Data transfer FPGA – RT application

- * Direct writing/reading from virtual front panel of FPGA VI
- * Simple data transfer without memory
- * FIFO – DMA transfer with buffered data transfer
- * Register
- * Handshake
- * Memory
- * FPGA I/O variable

* See the demonstration later for more details

Summer School on Marine Technology Instrumentation, UPC
Barcelona, Spain, July 2014

55

www.ni.com/myrio

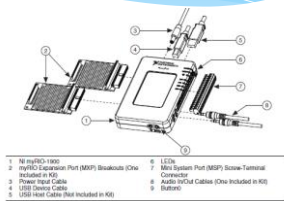
myRIO

Summer School on Marine Technology Instrumentation, UPC
Barcelona, Spain, July 2014

56

What is myRIO

- * National Instruments myRIO-1900 is a portable reconfigurable I/O (RIO) device that students can use to design control, robotics, and mechatronics systems

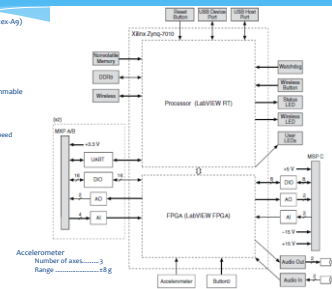


Summer School on Marine Technology Instrumentation, UPC
Barcelona, Spain, July 2014

57

What is inside?

Processor typeXilinx Zynq (ARM Cortex-A9)
Processor speed400 MHz
Processor cores3
Memory
Nonvolatile memory256 MB
DDR3 memory256 MB
DDR3 clock frequency533 MHz
DDR3 data bus width64 bits
FPGAXilinx Zynq (28,000 programmable logic cells)
Wireless Characteristics
Radio modeIEEE 802.11b/g/n
USB Ports
Host port + USB device port...USB 2.0 Hi-Speed
Analog Input
Aggregate sample rate500 kS/s
Resolution12 bits
Overvoltage protection+16 V
Analog Output
Aggregate max. update rates340 kS/s
Resolution12 bits
Overvoltage protection+16 V
Digital I/O
GPIO I/O348
2xUART230,400 bps
2xSPI40 MHz
2xPWM100 kHz
2xQuadr. encoder input100 kHz
2xI²C400 kHz

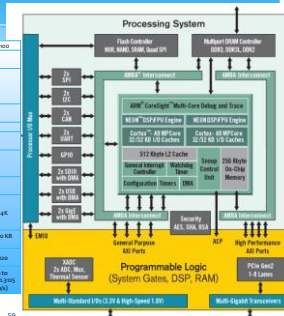


Summer School on Marine Technology Instrumentation, UPC
Barcelona, Spain, July 2014

58

Xilinx Zynq 7000 family

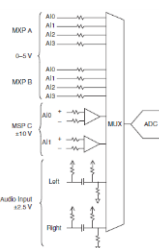
	Z-7010	Z-7015	Z-7020	Z-7025	Z-7030
Processor Core	Dual ARMv7 Cortex™-A9 MPCore™ with CoreLink™				
Processor Extensions	NEON™ & Single / Double Precision Floating Point for each processor				
L1 Cache	32 KB Instruction, 32 KB Data per processor				
L2 Cache	512 KB				
On-Chip Memory	256 KB				
Memory Interfaces	DDR3, DDR3L, DDR3L/DDR3L, 1x Quad SPI, NAND, NOR				
Peripherals	2x USB 2.0 (OTG), 2x Tri-mode Gigabit Ethernet, 1x SDIO				
Logic Cells	28K	74K	84K	105K	144K
BlockRAM (Kb)	144	384	512	640	864
DSP Slices	80	160	200	250	320
Transceiver Count	4 (6.5 Gb/s)	16 (6.5 Gb/s)	16 (6.5 Gb/s)	16 (6.5 Gb/s)	16 (6.5 Gb/s)



Summer School on Marine Technology Instrumentation, UPC
Barcelona, Spain, July 2014

59

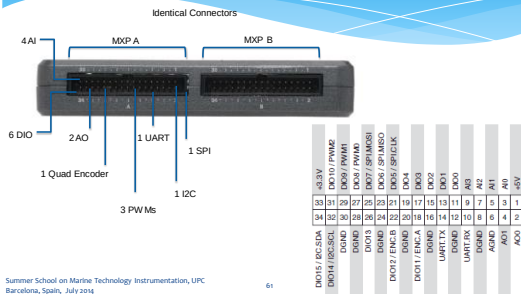
Analog inputs and outputs



Summer School on Marine Technology Instrumentation, UPC
Barcelona, Spain, July 2014

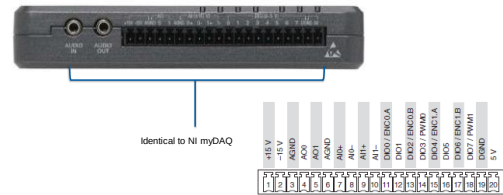
60

MXP connectors



61

MSP connector



62

NI myRIO Palette



63

Links

- * www.ni.com/myrio
 - * www.ni.com/labview
 - * www.ni.com/crio
 - * www.ni.com/flexrio
 - * www.ni.com/...
- Summer School on Marine Technology Instrumentation, UPC
Barcelona, Spain, July 2014

64