



Developing Custom CompactRIO Modules

An Introduction to the Module Development Kit 2.0

Background



William Johnston

- Certified LabVIEW Architect
- Certified Instructor
- Project Engineer (Embedded Systems)



Bloomy Controls

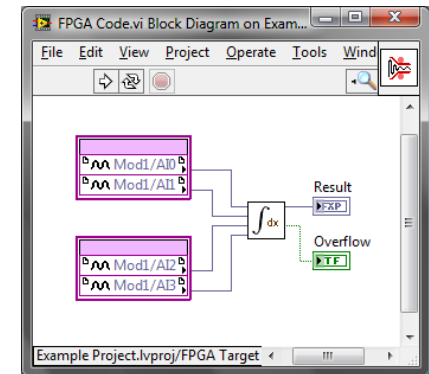
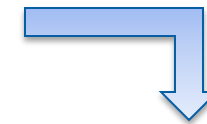
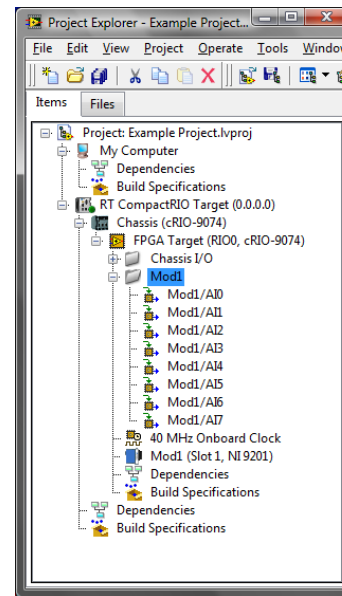
- Platinum Alliance Partner of National Instruments (NI)
- Automated test, data acquisition, and control specialists since 1991
- Turnkey test systems, systems integration, software development, and training services

Agenda

- NI cRIO platform and module development kit
- Basic hardware design
- Basic software design
- Development and testing
- Lessons learned
- Utilities and resources
- Feedback

Overview – CompactRIO

- Range of NI c-series modules available
- Simple user experience
 - Plug-in module
 - Add to project
 - Drag nodes into code



Overview – Module Development Kit 2.0

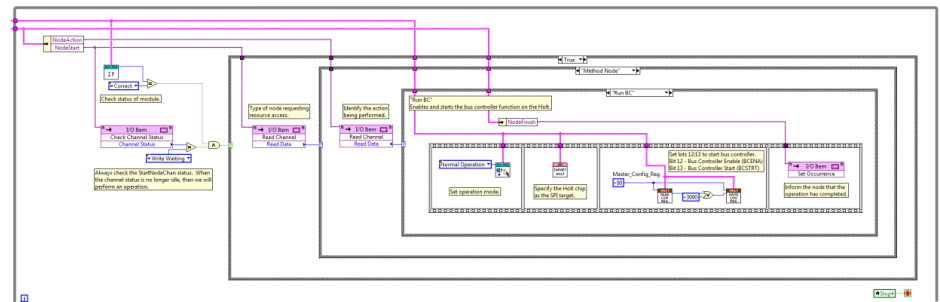
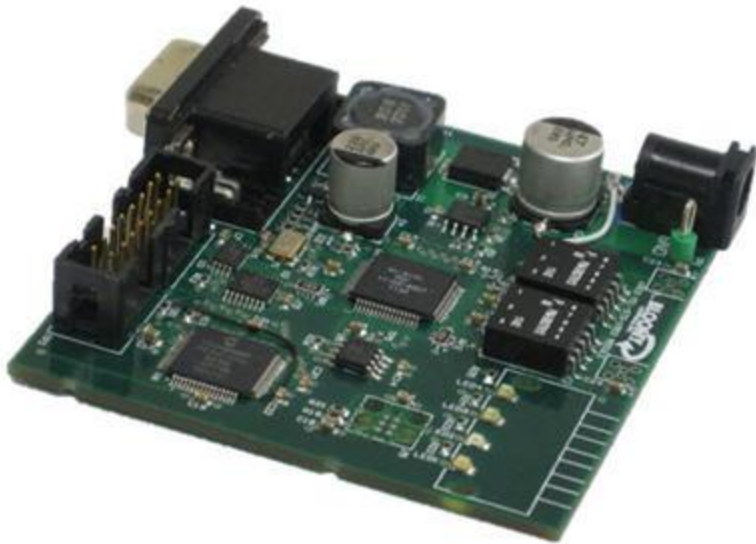
- Third-party modules for specialized applications
- Kit (NI cRIO-9951) includes:
 - Hardware development manual
 - Software development manual
 - Module housing and connectors
 - PCB insert
 - Information on support forum
- Improved development features
 - MDK API palette for developers
 - Ambiguous chassis slot installation
 - End-user API nodes identical to nodes for NI modules



Example Application

Specialized Communication Module

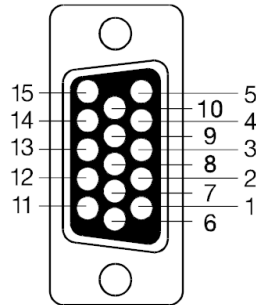
- MIL-STD-1553 (aeronautical applications)
- Ruggedized application environments
- Driven by customer interest
- Custom features/functions



Hardware Design – Backplane Interface

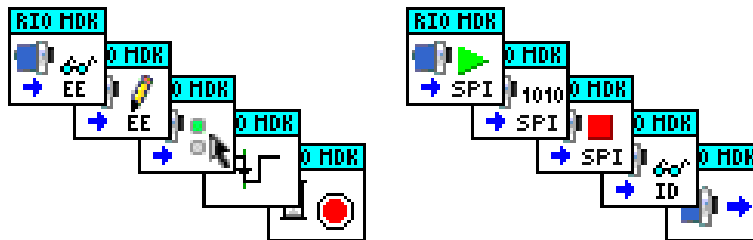
- HD15 Connector

- FPGA → Module
- 8x DIO
- 5x Dedicated Purpose



- SW Wrappers

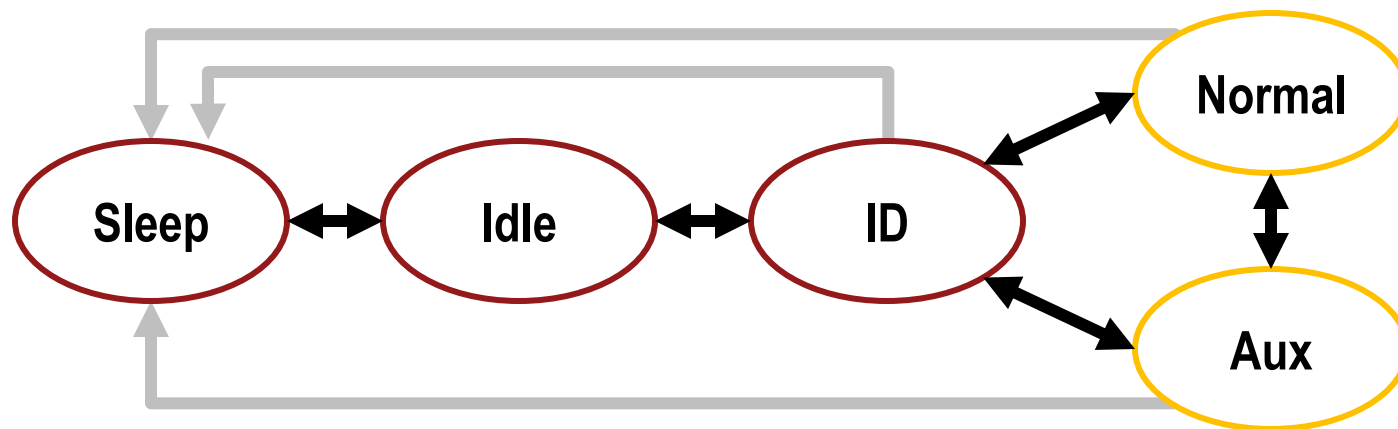
- SPI communication
- Timing and synchronization
- Custom communication schemes



HD15 Pin	Function
1	Special (ID_SELECT)
2	GPIO / SPI
3	Special (Reserved)
4	Special (Power)
5	GPIO
6	GPIO / SPI
7	GPIO / SPI
8	Special (SLEEP)
9	GPIO
10	GPIO
11	Special (SPI Clock)
12	GPIO / SPI
13	Special (Reserved)
14	Special (GND)
15	GPIO

Hardware Design – Module Modes

Module Mode	Purpose	I/O Configuration
Sleep	Standby	SLEEP line only
Idle	Detect module	SLEEP and ~ID_SELECT
ID	Identify module and access EEPROM	SPI comm lines
Normal	Primary functions	All comm/DIO lines
Auxiliary	Additional functions	All comm/DIO lines



Hardware Design – EEPROM Data

- Identification block
 - Module information
- Calibration table
 - Custom data

Table 4-10. Calibration Table Header

Address	Short Description	Size in Bytes
0x0030	Table Type Major	1

Table 4-12. Table Types 1.2 Calibration Blocks

Relative Address	Short Description	Size in Bytes
0x0000	Calibration Timestamp	8

Table 4-11. Table Type 1.1 Calibration Blocks

Relative Address	Short Description	Size in Bytes
0x0000	Calibration Timestamp	8
0x0008	Calibration Temperature	1
0x0009	Calibration CRC-16	2
0x000B	Constant List	Varies
Varies	Constant List CRC-16	1

Table 4-8. ID EEPROM Identification Block Format

Address	Short Description	Size in Bytes
0x0000–0x0001	EEPROM Size	2
0x0002–0x0003	Start Sentinel	2
0x0004–0x0005	Reserved (0x0000)	2
0x0006–0x0007	Product ID	2
0x0008–0x0009	Vendor ID	2
0x000A–0x000B	Module Model Code	2
0x000C–0x000F	Extended Serial Number (for future use)	4
0x0010–0x0013	Serial Number	4
0x0014	Configuration Version Major	1
0x0015	Configuration Version Minor	1
0x0016–0x0017	Module Descriptor Address	2
0x0018–0x002F	Reserved (0x00)	24

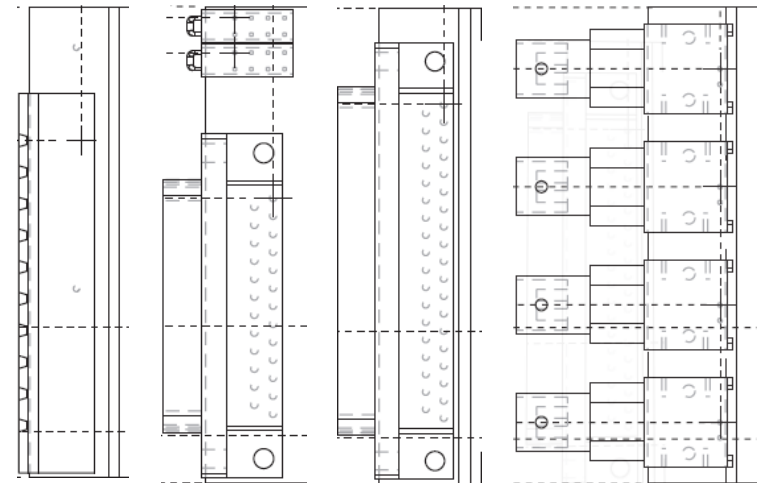
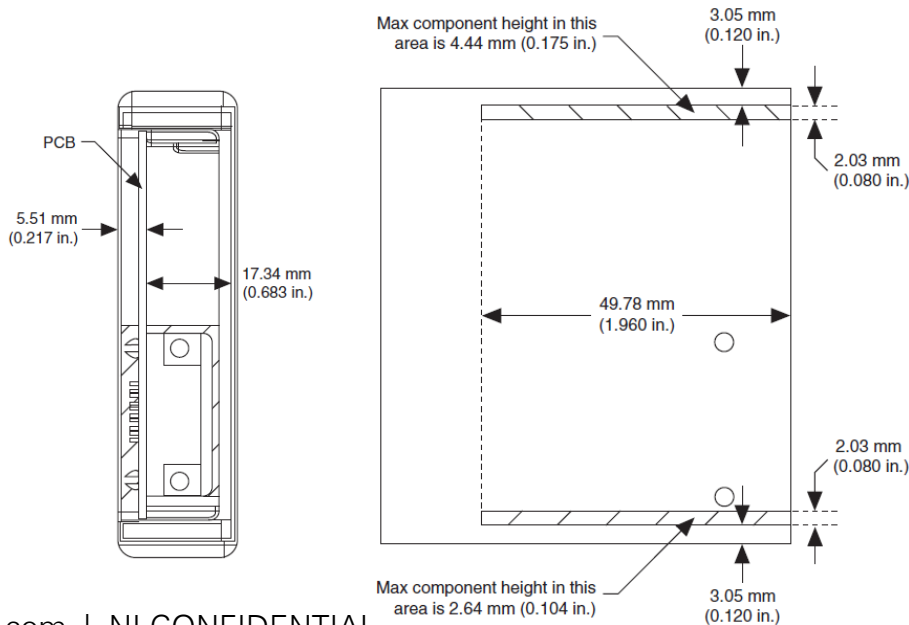
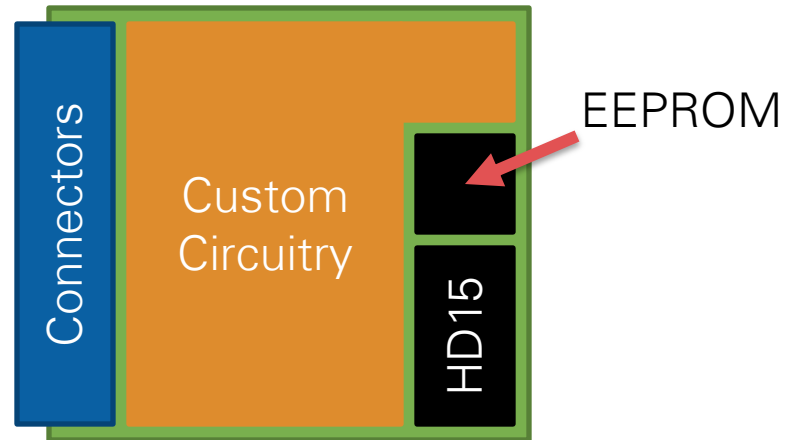
Table 4-9. Calibration Table Format

Address	Short Description	Size in Bytes
0x0030–0x0057	Calibration Table Header	40
0x0058–end	Factory Calibration Block	Varies
	External Calibration Block	Varies

*From MDK Hardware User Manual

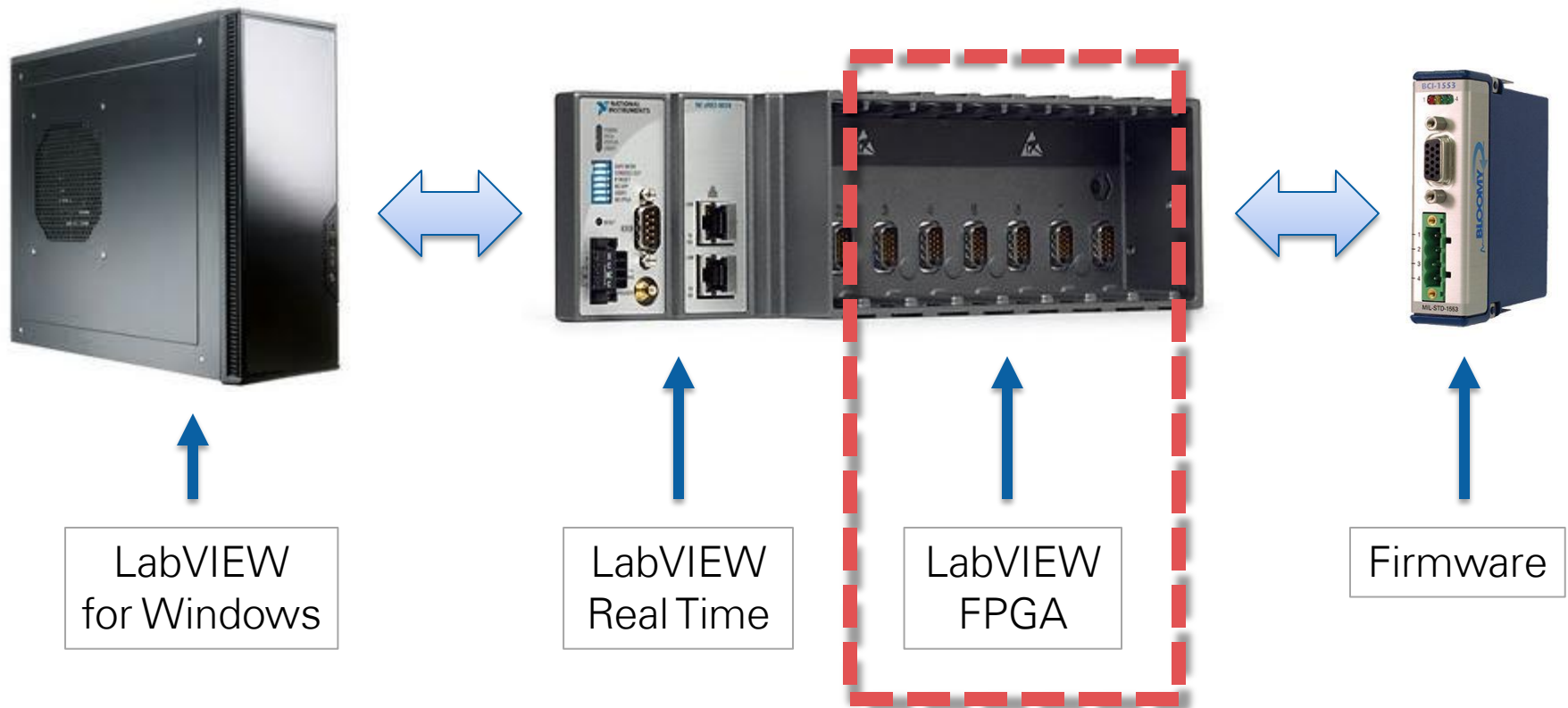
Hardware Design – Physical Constraints

- Space
 - PCB: 2.59" x 2.89"
 - Connectors: 0.68" x 2.89"
- Electrical (backplane)
 - Voltage: 5V
 - Current: 200mA

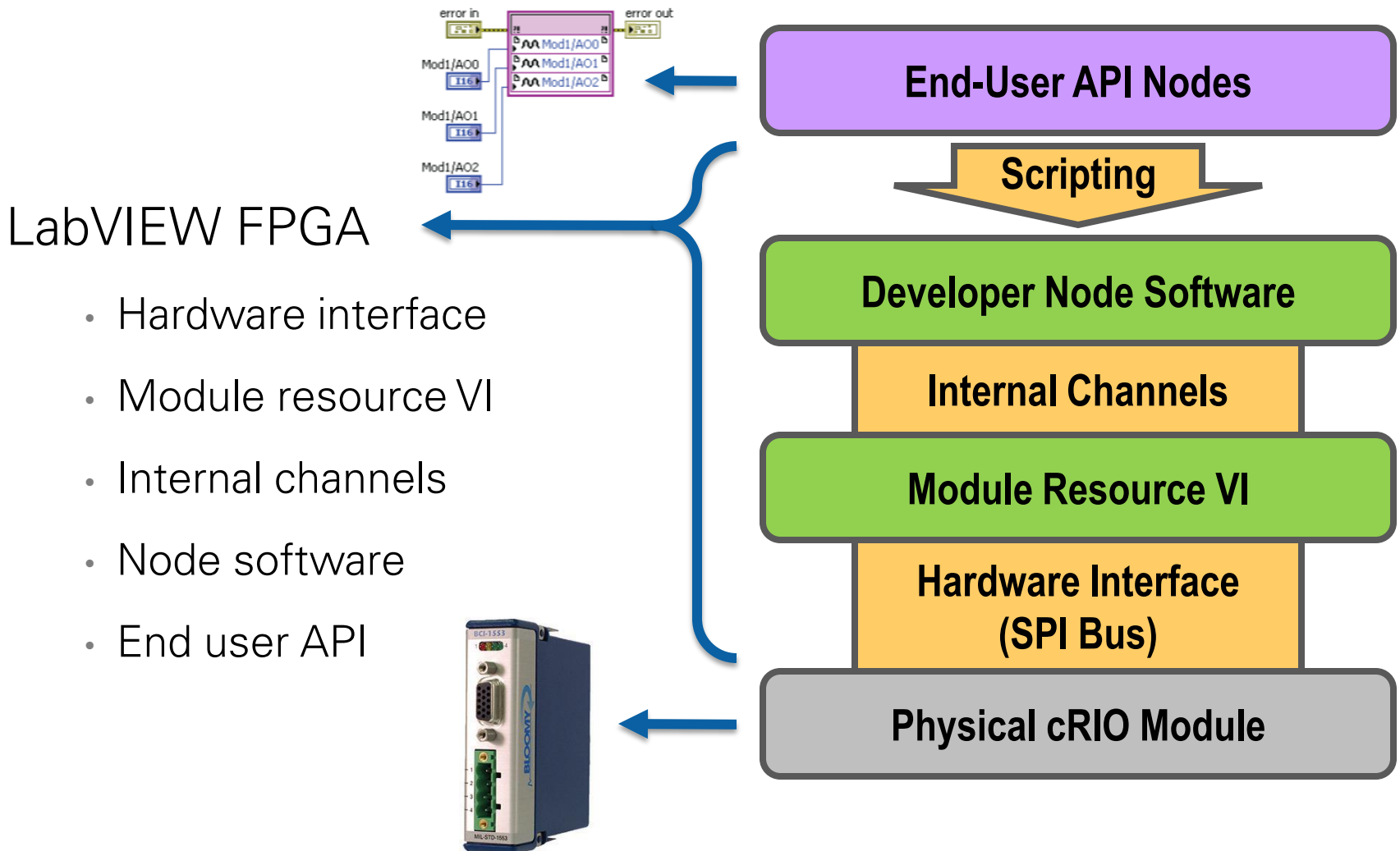


*From MDK Hardware User Manual

Software Design – Execution Locations

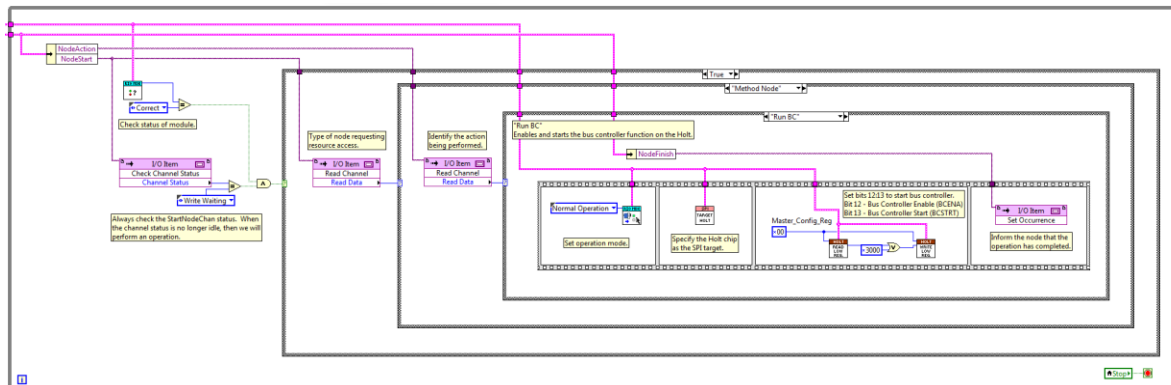
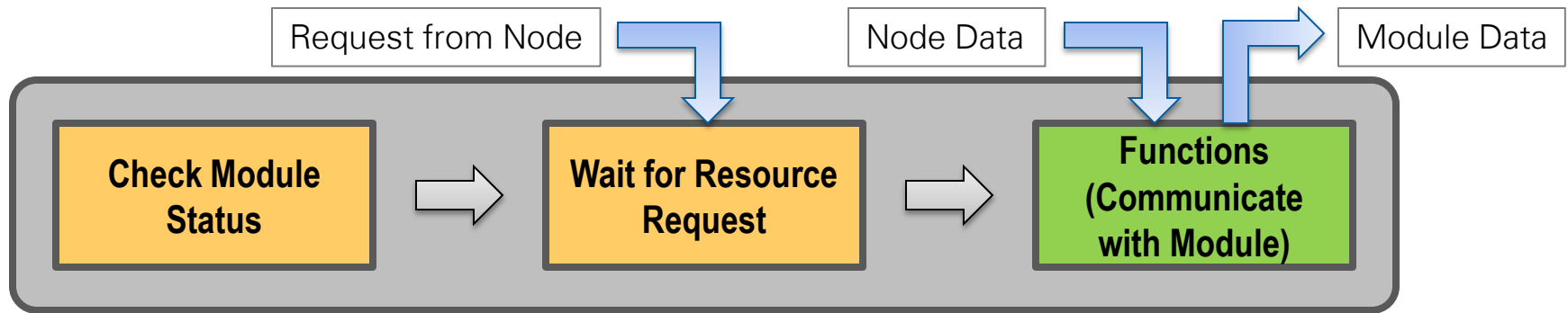


Software Design – Layer Map



Software Design – Module Resource VI

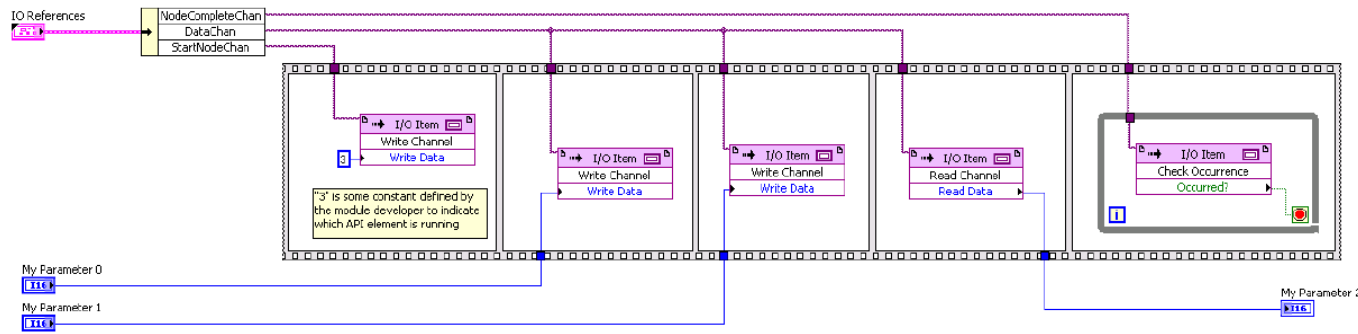
- Monitors module status
 - Unknown, Correct, Incorrect, No Module, Invalid
- Regulates access to communication bus



Software Design – Internal Channels

- Transfer data between nodes and module resource VI
- Regulate and synchronize data and actions

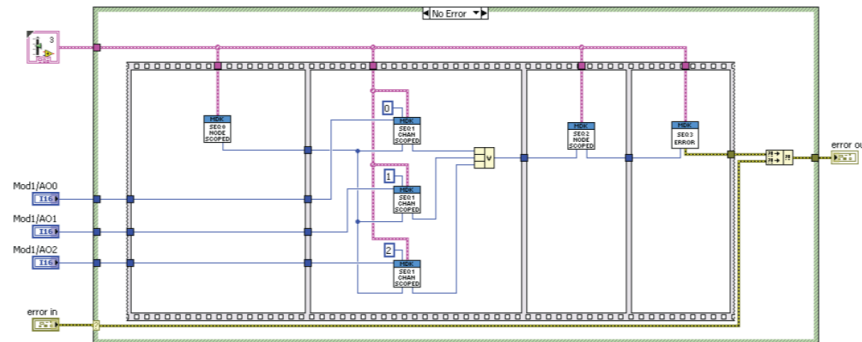
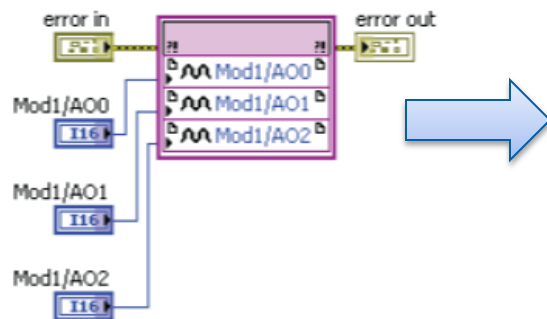
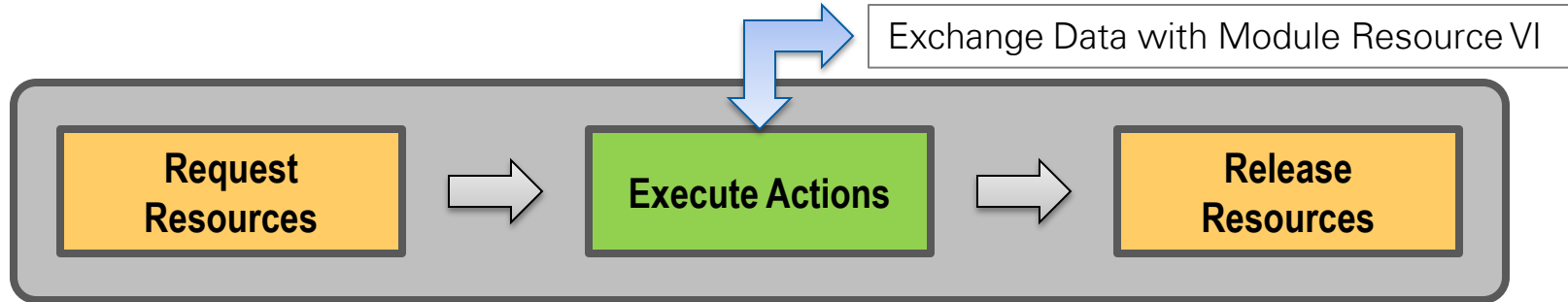
Channel Type	Write Action	Read Action	Uses
Asynchronous	<ul style="list-style-type: none"> • Replaces data • Returns immediately 	<ul style="list-style-type: none"> • Reads most recent data • Returns immediately 	Publishing data without feedback
Blocking	<ul style="list-style-type: none"> • Completes once data is read 	<ul style="list-style-type: none"> • Completes once data is available 	Synchronized data transfers
Occurrence	<ul style="list-style-type: none"> • Sets flag • Returns immediately 	<ul style="list-style-type: none"> • Reads and clears flag • Returns immediately 	Triggering actions



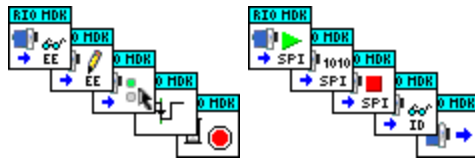
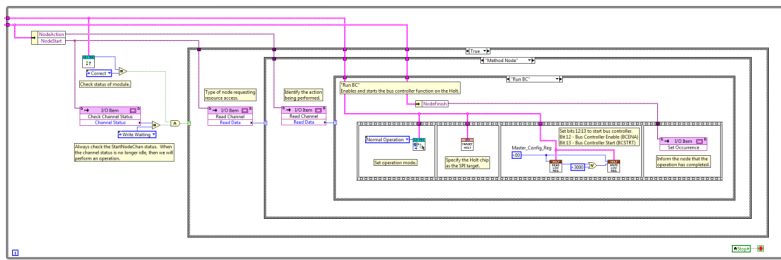
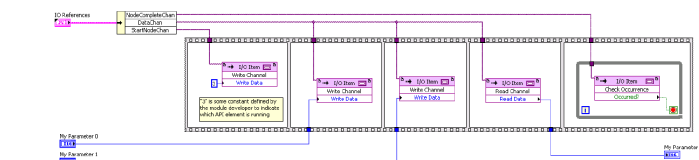
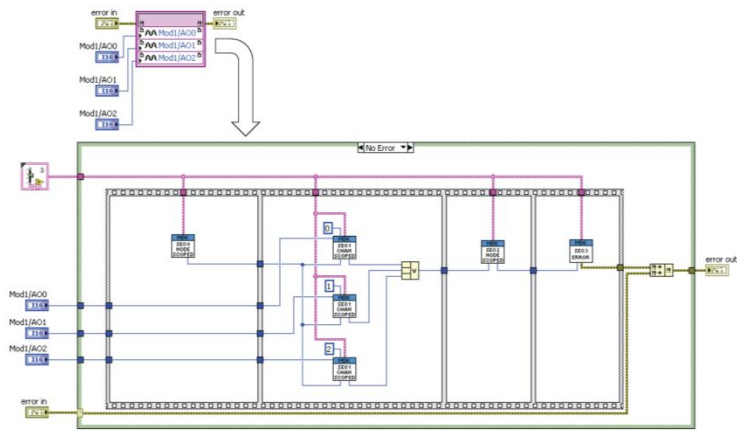
*From MDK Software User Manual

Software Design – Node Software

- I/O nodes – ‘direct’ reflection/control of hardware states
- Property Nodes – configuration and monitoring
- Method Nodes – triggers and actions



*From MDK Software User Manual



End-User API Nodes

Scripting

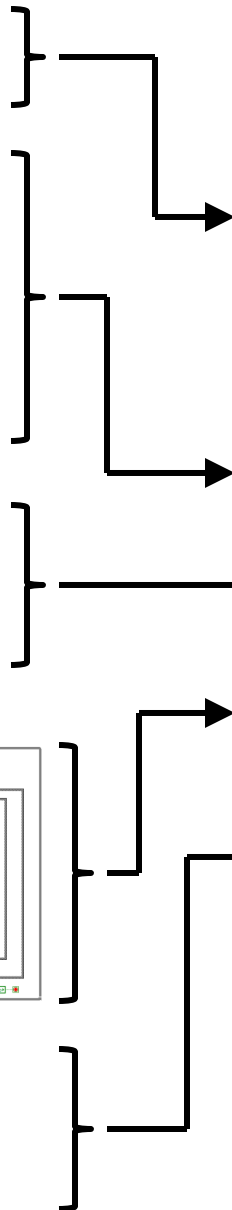
Developer Node Software

Internal Channels

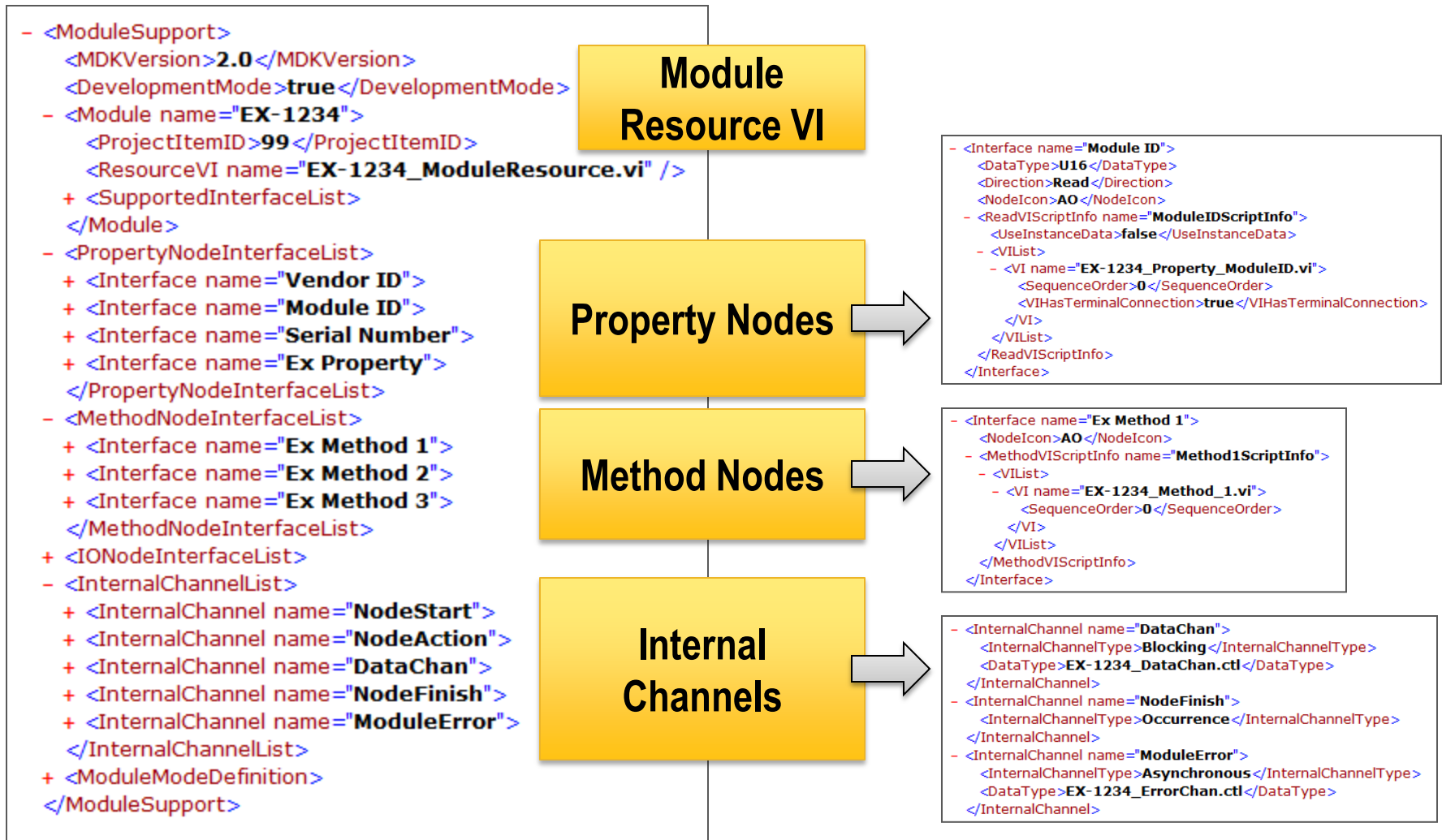
Module Resource VI

Hardware Interface (SPI Bus)

Physical cRIO Module



Software Design – Module Support XML File

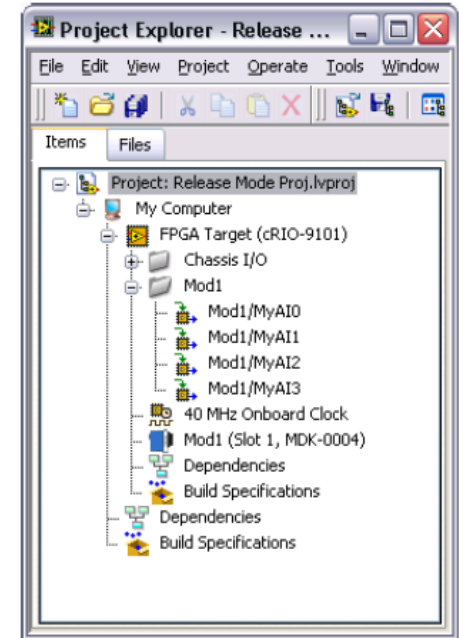
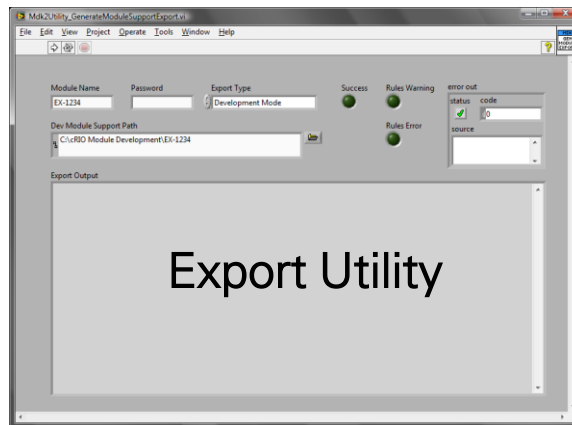
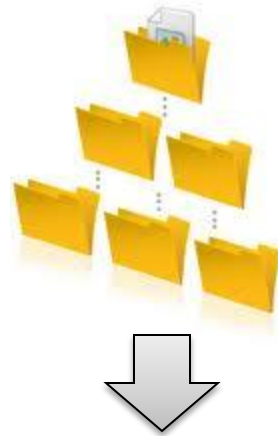
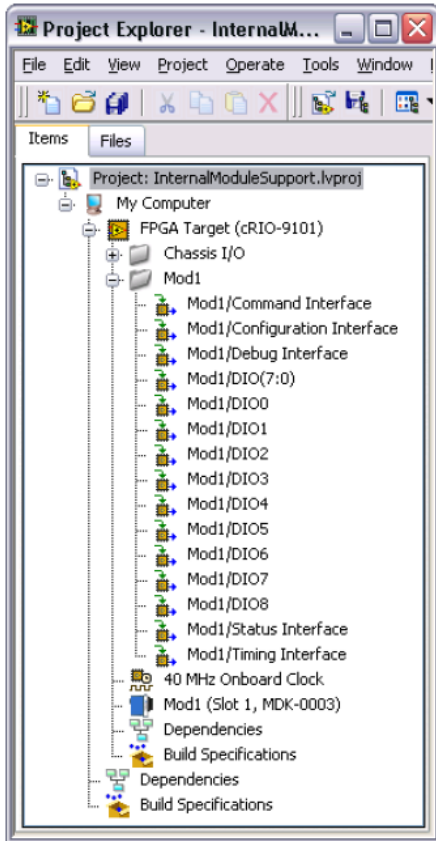


Development and Testing – Export Modes

Development Folder

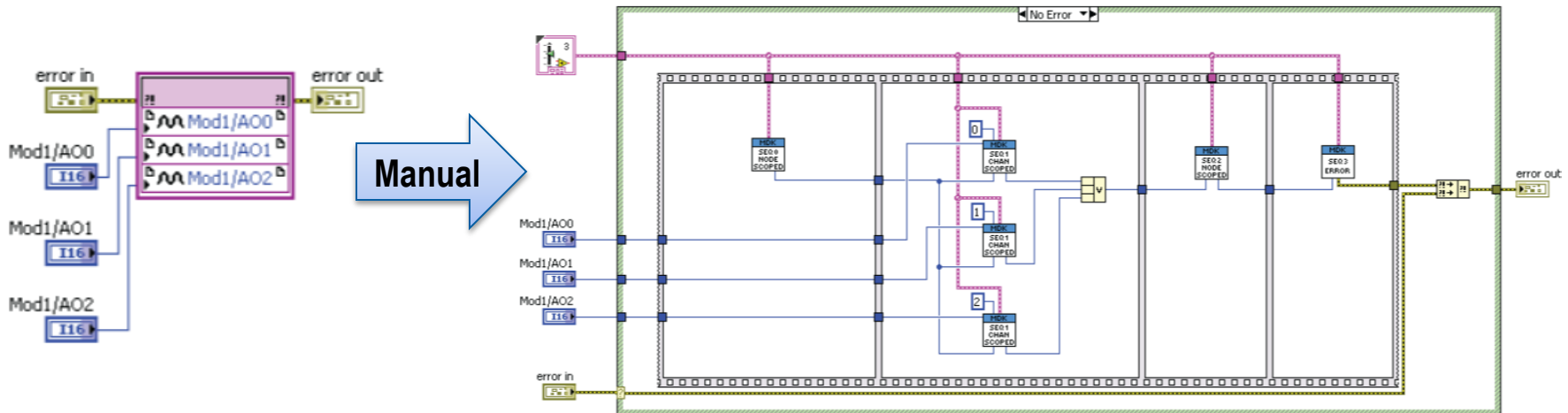
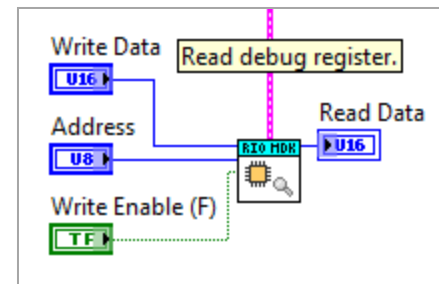
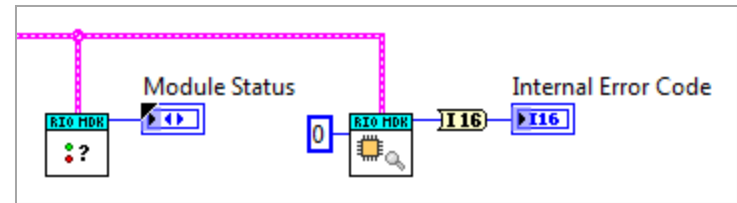
Development Mode

Release Mode



Development and Testing – Framework

- Module status
- Internal error codes
- Debug / EEPROM register
- Simulated node scripting
- Example projects



Lessons Learned

- Error terminals (consumes FPGA resources)
- Internal channels (changing data)
- Recommended property nodes
 - Module ID
 - Vendor ID
 - Serial Number
- Folder export ('extra' files will cause errors)
- Standard cutout for hardware connectors

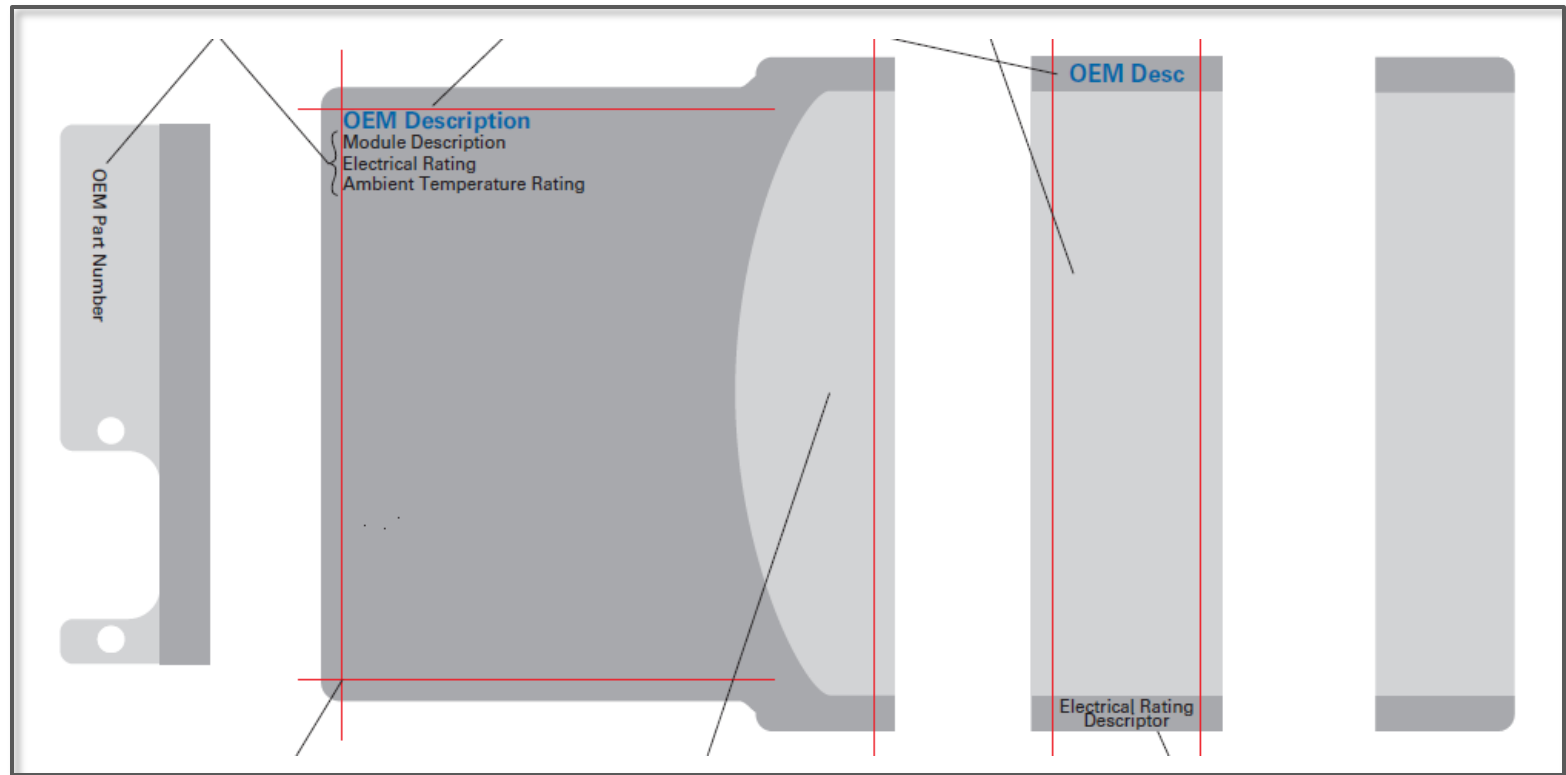
Utilities and Resources

- Example projects
 - MDK-9901, MDK-9902, MDK-9903, MDK-MFG
- EEPROM loader / reader
 - MDK-MFG Write EEPROM Contents to Module (Host).vi
 - MDK-MFG Read EEPROM Contents from Module (Host).vi
- Export utility
 - Mdk2Utility_GenerateModuleSupportExport.vi
- MDK forum
 - [Private] – Email NI for access
- Software / Hardware design manuals

Module Enclosures



- Templates available in forum
- Northern Engraving (www.norcorp.com)





Questions?

Visit our booth (#519)

or

Check out our latest projects at

www.bloomy.com

will.johnston@bloomy.com



Stay **Connected** During and After NIWeek



ni.com/niweekcommunity



facebook.com/NIWEEK



twitter.com/#!/niweek



<http://linkd.in/ljfwyB>



youtube.com/niglobal