Advanced NI-DAQmx Programming Techniques with LabVIEW
Agenda

• Understanding Your Hardware
  – Data Acquisition Systems
  – Data Acquisition Device Subsystems

• Advanced Programming with NI-DAQmx
  – Understanding Your Timing Engine
  – Multiple Device Synchronization
  – Optimizing the Data Transfer Path (DMA, buffers, data streaming)
Understanding Your Hardware
What is a DAQ System?

Input Signals

Output Signals

Application Software (LabVIEW)

Measurement Services Software (NI-DAQmx)

Signal Conditioning

A/D

D/A

DIO

TIO
## Transducers Options

<table>
<thead>
<tr>
<th>Phenomena</th>
<th>Transducer</th>
</tr>
</thead>
<tbody>
<tr>
<td>Temperature</td>
<td>Thermocouples</td>
</tr>
<tr>
<td></td>
<td>Resistive Temperature Devices (RTDs)</td>
</tr>
<tr>
<td></td>
<td>Thermistors</td>
</tr>
<tr>
<td>Light</td>
<td>Vacuum tube</td>
</tr>
<tr>
<td></td>
<td>Photo sensors</td>
</tr>
<tr>
<td>Sound</td>
<td>Microphone</td>
</tr>
<tr>
<td>Force and Pressure</td>
<td>Strain gauges</td>
</tr>
<tr>
<td></td>
<td>Piezoelectric transducers</td>
</tr>
<tr>
<td>Position and Displacement</td>
<td>Potentiometers</td>
</tr>
<tr>
<td></td>
<td>Linear voltage differential transformer</td>
</tr>
<tr>
<td></td>
<td>Optical encoder</td>
</tr>
<tr>
<td>Fluid</td>
<td>Head meters</td>
</tr>
<tr>
<td></td>
<td>Rotational flowmeters</td>
</tr>
<tr>
<td>pH</td>
<td>pH electrodes</td>
</tr>
</tbody>
</table>
Signal Conditioning

High voltage signals and most sensors require signal conditioning to properly read the signal.

**Sensors/Signals**
- Thermocouples
- RTDs
- Strain Gauges
- Common Mode or High Voltages

**Signal Conditioning**
- Amplification, Linearization, and Cold-Junction Compensation, Filtering
- Current Excitation, Linearization, Filtering
- Voltage Excitation, Bridge Configuration, Linearization, Filtering
- Isolation

DAQ Device
NI Signal Conditioning Hardware Options

Modular Signal Conditioning
- SCC
- SCXI
- FieldPoint

Integrated Signal Conditioning
- PXI Instruments
- SC Series
- USB-9200 Series

Visit ni.com for more information.
NI DAQ Hardware Options

- Ethernet, Serial, or Wireless
- Distributed
- PXI
- Desktop
- Portable/Handheld

ni.com
DAQ Device – Subsystems

• Most DAQ devices have:
  – Analog Input
  – Analog Output
  – Digital I/O
  – Counters

• Applications specific devices for:
  – High speed digital I/O
  – High speed waveform generation
  – Dynamic Signal Acquisition (vibration, sonar)

• Compatible with a variety of bus protocols:
  – PCI, PXI/CompactPCI, PCI Express, PCMCIA, USB, 1394/Firewire®
Analog Input Subsystem

• Uses an ADC (analog to digital converter)
  – Converts analog signal to digital bits that a computer can manipulate

• Allows acquisition of “real world” analog signals
Analog Output Subsystem

- Architecture
- Accuracy
  - Resolution
  - Absolute Accuracy
- Range
  - Adjustable versus Fixed

- Waveform Frequency
  - Update Rate
  - Settling Time
  - Number of Cycles in the Buffer
Digital I/O Subsystem

• General Terminology
  – **Bit**: The smallest unit of data. Each bit is either a 1 or a 0.
  – **Line**: One individual signal in a port. Bit refers to the data transferred. Line refers to the hardware.
  – **Port**: A collection of digital lines
Two basic functions:
1) To “count” based on the comparison of input signals (Gate, Source…)
2) To generate pulses based upon inputs and register value
NI-DAQmx and Measurement Services

- **Streamlined API**
  - Polymorphic functions
  - Automatic code generation

- **Driver Architecture**
  - Multithreaded measurements
  - Instant calibration
  - Fast single-point operations
  - Code generation

NI-DAQmx Driver Software and Measurement Services
NI-DAQmx Fundamentals

NI-DAQmx Task

Virtual Channel

Scale

y=mx+b

Units

Physical channel

Terminal configuration

Input range

Calibration

Acquisition mode

Triggering

Timing

Clock Source
Programming NI-DAQmx and LabVIEW

Flexible programming options:

• **DAQ Assistant**
  - Creating Tasks and Channels in MAX
  - DAQ Assistant Express VI
  - DAQmx Task Name Constant

• **Automatic code generation**

• **NI-DAQmx API**
  - DAQmx VIs
  - DAQmx Property Nodes
Demo – Express VIs and NI-DAQmx VIs

Implementing Pause Triggering
Understanding Your Timing Engine
Configuring Analog Input Clocks

- Sample clock and convert clock
- Multi-channel scanning considerations
- Using an internal clock
- Using an external clock
- Programmable functions input (PFI) pins
Sample and Convert Clock

- **Sample Clock** controls when a scan begins
- **Convert Clock** controls when each channel is sampled
Interval Scanning

• When only the sample rate is set, LabVIEW requests for NI-DAQmx to select the convert clock rate.

• NI-DAQmx
  – Selects fastest convert clock rate possible, then adds 10 μs to the interchannel delay.
Demo – Change Convert Clock Rate

DAQmx Timing property node
Settling Time

- Time required for signal being amplified by PGIA to reach specified accuracy range
- To acquire accurate data, signal must settle within accuracy range before A/D conversion takes place
- Usually specified in LSBs (Least Significant Bits)
  - 1 LSB = smallest voltage change detectable by the ADC
Programmable Gain Instrumentation Amplifier (PGIA)

- Applies gain to signal
- Can apply different gains to different channels
- As you multiplex between different channels, you must let the amplified signal settle before sampling it with the ADC
Factors That Affect Settling Time

• Multiplexing
• Characteristics of input signal
  – Higher gains increase settling time
  – Large voltage swings between channels increase settling time
• Source impedance
  – High source impedance increases settling time due to charge injection
  – Built up charge on mux from last channel must dissipate through the next channel
  – The higher the source impedance, the more time is required for the charge to dissipate
• Transmission line resistance and capacitance
Source Impedance

- Source impedance $> 1 \, \text{k}\Omega$ may cause ghosting
Improve Accuracy and Reduce Ghosting

- Select transducer with low source impedance
- Reduce sampling rate and/or convert clock rate
- Implement a voltage follower circuit to decrease source impedance if $> 1 \, \text{k}\Omega$
- Arrange signals to minimize voltage swings between channels
- Insert grounded channels between signal channels
- Avoid multiplexing – sample one channel at a time
Internal Clocks

• Default clocking source
• NI-DAQ divides down onboard timebases to achieve sampling clocks
• Different boards offer different timebases
  – DAQ-STC2: 20 MHz (default) or 100 kHz
M Series – Timebase and Clock Sources

- Onboard 80 MHz Oscillator
- External Reference Clock
- PLL
- 10 MHz RefClk
- RTSI <0..7>
- PXI_CLK10
- PXI_STAR

Reference Clock

- 80 MHz Timebase
- 20 MHz Timebase
- 100 kHz Timebase

(To RTSI <0..7> Output Selectors)

ni.com
M Series – Clock Sources (Cont.)

• Reference clock
  – The source of the internal timebase

• By default, Reference clock is set to “none”
  – This means that it will use the onboard oscillator as the source of the internal timebase

• Setting the Reference clock to “On-board Clock”
  – 10 MHz clock is passed to the Master device’s Reference Clock source through RTSI so Master device has the same delay as slaves
External Clocks – Analog Input

• Use a TTL compatible signal
• Analog Input
  – External Sample Clock
    • Input on any unused PFI pin (rising or falling edge)
  – External Convert Clock
    • Input on any unused PFI pin (rising or falling edge)
Programmable Function Input (PFI) Pins

• Serve as connections to virtually all internal timing signals
• Can input timing signal on any PFI pin
  – Example: import digital trigger signal on PFI0
• Can output timing signal on any PFI pin (M Series)
  – Example: export AI sample clock on PFI7
Input Signal on Any PFI Pin

- NI-DAQmx:
  - Select any PFI pin as source of signal
  - Can specify active edge (rising/falling)
Output Signal on Dedicated PFI Pin

• Must turn on output driver to output signal
• NI-DAQmx:
  – Use LabVIEW **Export Signal** VI
  – Use C API **Export Signal** function
External Convert Clock (AI)

• NI-DAQmx does not allow disabling the sample clock
• Therefore, it is not possible for an external convert clock to solely control all of the A/D conversions
• The alternative is to provide both an external sample and convert clock
External Sample Clock (AI)

- If the source input of the DAQmx Timing VI is left unwired, the internal clock is used by default.
- A rate must always be specified when using an external clock.
Both Sample and Convert Clock (AI)
Timing Tool

- M and E Series diagrams
- Interactive
- Available on ni.com
Multiple Device Synchronization
Demo – Synchronization
AI/AI Shared Sample Clock

Minimum Value
Maximum Value
Physical Channel

AI Voltage
Sample Clock

Rate
Samples per Channel

Minimum Value 2
Maximum Value 2
Physical Channel 2

/Dev1/ai/SampleClock

Master device data

Slave device data

ni.com
Sharing a Timebase and Start Trigger
PLL Circuit

- Circuit that adjusts an oscillator so that its frequency is locked to a reference clock
- Can replace the native 10 MHz PXI clock with a more accurate clock to improve accuracy
Synchronizing PCI M Series Devices

• Recommended Method:
  – Slave pulls Master’s 10 MHz reference clock as its reference clock and sets the rate to 10 MHz
  – Master sets the source of its reference clock to “onboard” clock
  - By default, the reference clock is set to “none” which means that it uses the on-board oscillator
  - Onboard clock causes the master’s reference clock to be routed through RTSI to have the same delay as the slave
PCI M Series Synchronization
Reducing Clock Error in PXI

• PXI chassis has 10 MHz backplane clock
  – Common reference clock provided to each slot
  – Trace lengths matched to minimize skew
  – May be replaced by another 10 MHz clock

• Devices phase-locked to the backplane clock
  – Clock edges occur simultaneously
  – Frequencies are derived from common 10 MHz backplane clock

• Replace backplane clock with high-stability 10 MHz clock to reduce clock error in PXI
Synchronizing PXI M Series Devices

• PXI M Series devices do not automatically phase lock to the PXI Clock

• Recommended Method:
  – Set the source of all device reference clocks to PXI Clock 10 and set the reference clock rate to 10 MHz
  – This causes the PLL circuit to output a clock that is phase locked to PXI Clock 10
  – Share a common start trigger between all devices and set all sample clocks to a common rate
PXI M Series Synchronization
Optimizing the Data Transfer Path
Data Transfer Mechanisms

• Programmed I/O
  – Each read or write call in the application initiates the transfer of data
  – Used for software timed operations

• Interrupt Request (IRQ)
  – Relies on the CPU to service data transfer requests
  – Transfer speed limited by the PC
  – Used when point by point data is needed

• Direct Memory Access (DMA)
  – Data is transferred directly to memory, bypassing the CPU
  – Faster method of transfer than IRQ
  – Typical (default) method of data transfer
  – Limited number of DMA channels per board
Components of Streaming

DMA Controller

FIFO

Buffer

Samples to Read
Measurement

Analog 1D Wfm
NChan NSamp

status

timecut

10.00

stop

stop
Components of Streaming

- DMA Controller
- FIFO
- FIFO Buffer

Diagram showing the components and connections in a streaming system.
Buffer Overflow
Run Error Demo
Solutions to Buffer Overflow

- Configure Task to Ignore Overflows
- Reduce Number of Concurrent Applications
- Increase Buffer Size, Tweak Number of Samples Per Read, Reduce Sample Rate
- Move to Off Line Analysis
- Use Unscaled or Raw Reads
Show Fixes to Error Demo
FIFO Overflow

DMA Controller → PCI 32 → Bus → Buffer

Bus Unavailable

Local Bus

FIFO

ni.com

NATIONAL INSTRUMENTS
Utilizing Onboard Memory
Utilizing Onboard Memory

DMA Controller → PCI → Local Bus → FIFO → Bus → Buffer → DMA Controller

- DMA Controller: Direct Memory Access controller
- PCI: Peripheral Component Interconnect
- Local Bus: Local computer bus
- FIFO: First-In, First-Out buffer
- Buffer: Memory buffer

NI.com

National Instruments
Utilizing Onboard Memory Summary

• For Devices With Large Onboard Memory
  – Reference triggered acquisitions are no longer limited by PCI bandwidth
  – Finite Acquisitions
    • Effective Buffer = Host Buffer + Onboard Memory

\[
\begin{align*}
33 \text{ MS} & \quad = \quad 1 \text{ MS} \quad + \quad 32 \text{ MS} \\
\text{Effective Buffer} & \quad \text{Buffer} \quad \text{FIFO}
\end{align*}
\]

• No longer require large buffers
• Fewer page locking issues
Scaled, Unscaled, and Raw Data

Waveform Data

Timing And Attributes

Scaled Data

Channel 1

Channel 2

Scaling

Unscaled Data

Raw Data

Raw ID I16

Reordering

Buffer

Channel 2

Channel 1

Analog 1D Wfm
NChan NSamp

Analog 2D DBL
NChan NSamp

Analog 2D I16
NChan NSamp

Raw ID I16

ni.com
Scaled, Unscaled, and Raw Data

Waveform Data
Scaled Data
Unscaled Data
Raw Data

Performance / Throughput
Ease of Use

ni.com
Streaming to Disk Enhancements

0000100101101011

Unused Resolution

DMA Controller

PCI Bus 32

Local Bus

NI PCI-6110

FIFO

DAQmx Channel

AI.RawDataCompressionType
AI.LossyLSBRemoval CompressedSampleSize

FIFO Buffer

Samples to Read

status

timeout

stop

Ram 1D USB
Resources

• **ni.com/support**
  – Examples
  – Knowledgebase
  – Discussion forums
  – Hardware/software manuals

• **MAX**
  – Hardware manuals
  – NI-DAQmx help documents

• **Start menu**
  – Getting started guide
  – Readme
  – C Function reference manual