

Synchronizing Analog Input C Series Modules with NI-DAQmx in LabVIEW

Publish Date: Dec 17, 2018

Overview

CompactDAQ controllers and chassis offer support for a wide variety of C Series I/O modules with diverse measurement types such as acceleration, temperature, and strain. When applications require tight synchronization between measurements, the CompactDAQ hardware in conjunction with the NI-DAQmx API allow user's flexible configuration options to achieve the desired time correlation across multiple C Series I/O modules. This document highlights some of the diverse options available to achieve synchronization between C Series I/O modules.

Table of Contents

1. C Series Analog Input Modules
2. Start Trigger Synchronization
3. Sample Clock Synchronization
4. Multidevice Task
5. Multirate Synchronization
6. MultiChassis Tasks
7. Conclusion
8. Additional Resources

1. C Series Analog Input Modules

C Series Analog Input Modules can be categorized into four groups based on their synchronization: Reference Clocked, Delta Sigma, Slow Sampled, and Successive Approximation Register (SAR). Reference Clocked modules convert using an internal onboard clock. This onboard clock is disciplined using a phase lock loop to another internal clock or can be disciplined to clocks external to the module. Delta Sigma Modules use a 24-bit ADC and convert using a clock internal to the module. Slow Sampled Modules use a 24-bit ADC and convert using an external clock that originates from the CompactDAQ controller or chassis. Lastly, SAR Modules use a Successive Approximation ADC (12- or 16-bit) and convert using an external clock. The method of implementing synchronization may depend on the group that the module falls into. A list of module groupings can be found in the Resources section of this tutorial. In this tutorial, synchronization is defined as the steps or techniques used directly or indirectly via the DAQmx driver to start measurements at the same time and to remove drift between subsequent measurements. While the result of these steps increases the sample to sample alignment, the focus is to prevent drift.

2. Start Trigger Synchronization

This section outlines different methods to control when acquisition begins across multiple modules. Because the focus is to simply align the start time across multiple tasks, this method is the most rudimentary way to achieve synchronization.

The first method uses a software function that, when executed, starts acquisition as defined by one DAQmx task. When this function executes in each task at relatively the same time, a similar level of synchronization is seen in the start of the data.

When hardware timing is used and no triggers are configured, executing the DAQmx Start Task VI will initiate the acquisition. This VI causes the CompactDAQ Controller and Chassis to send a digital pulse that begins the acquisition. If the DAQmx Start Task VIs of two independent tasks are placed in the same frame of a flat sequence structure, then the two tasks begin at the same relative time. However, the separation between the exact start times of the two tasks depends on the software time delay between the executions of the DAQmx Start Task VIs and will therefore vary due to the Operating System. This method is seen in figure 1 where the start of two analog input tasks is controlled during runtime with the DAQmx Start Task VIs inside the same frame of a flat sequence structure.

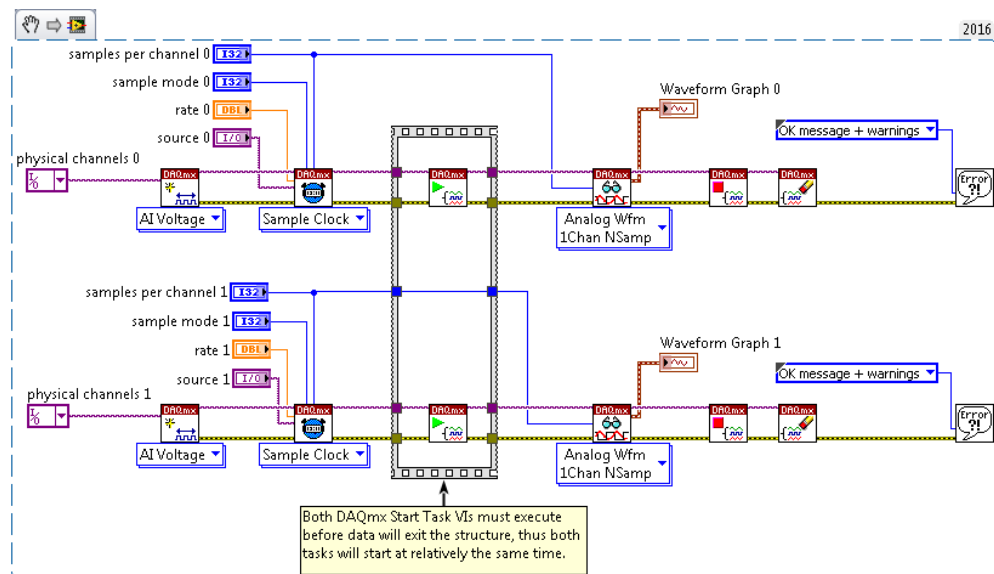


Figure 1: Start of Each Task Occurs Inside Flat Sequence Structure

To avoid the delay, the DAQmx Trigger VI can be used. This method can configure tasks to start upon a hardware trigger rather than by calling a software function. When using this feature to synchronize tasks, one task will be sending the trigger, and the other task will be receiving the trigger. The task receiving the trigger will wait for the trigger sent from the master task to begin acquisition or generation. When the master task starts, a digital trigger is sent from the master task and starts any task configured to start based upon this signal. This method is seen in the Figure 2 where the DAQmx Control Task VI is called before the task accepting the ai/Start Trigger ensuring that the Master Task reserves the ai/Start Trigger.

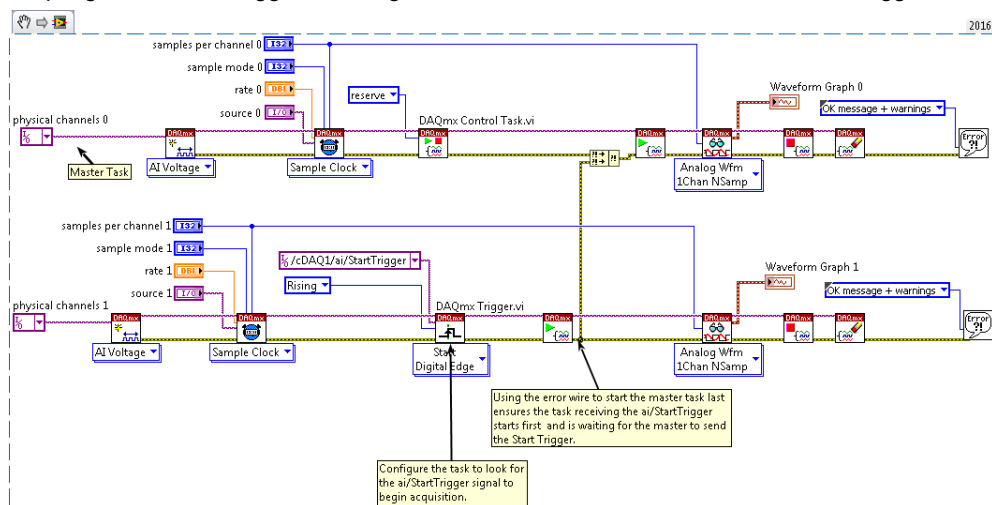


Figure 2: Trigger Configuration to Start Both Tasks

To understand trigger and timing nomenclature, you must know which timing engine is reserved for a task. CompactDAQ controllers and chassis have three analog input timing engines (<http://zone.ni.com/reference/en-XX/help/370466AD-01/mxdevconsid/timingenginemiio/>): ai, te0, and te1. Each concurrently run task (<http://digital.ni.com/public.nsf/allkb/5E0B829E50ADE1BC86257AC50062B2D2>) is assigned to a different timing engine to control the sample clock and triggers. For example, the ai/StartTrigger is assigned to the ai timing engine. The NI-DAQmx driver automatically assigns each analog input task to a timing engine starting with ai, then te0, and finally te1. It is important to know which task is using which timing engine, so we set tasks receiving the start trigger to use the signal from the same master task's timing engine. By reserving the master task first, we are assigning this task specifically to the ai timing engine. Now both tasks can be set to use the master task's start trigger (ai/StartTrigger).

3. Sample Clock Synchronization

Using the above technique, tasks are started at the same time by sharing a start trigger. However, this method will not guarantee that the timing of subsequent measurements between the two tasks will not drift away from each other. A sample clock can be shared between both tasks, so that any task that uses the shared clock can take samples at the same time. Therefore, the two tasks will not drift from each other. As discussed earlier, Compact DAQ controllers and chassis have three analog input timing engines, so the order in which the tasks are reserved will determine the timing engines used by each task. Figure 3 below details how to share the sample clock between tasks.

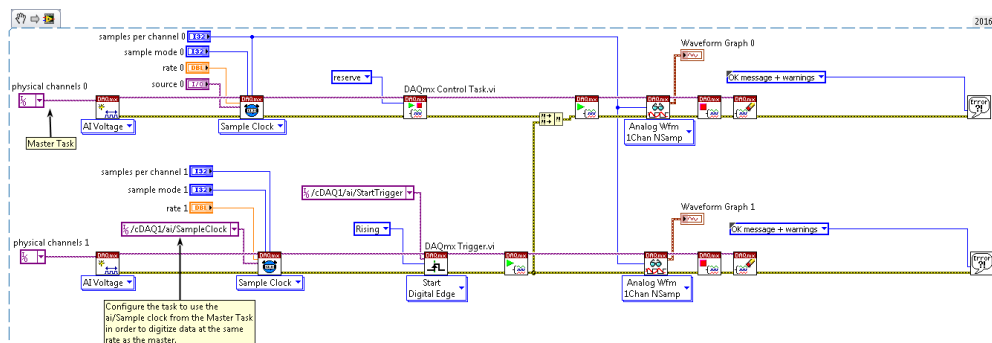


Figure 3: Sample Clock and Trigger is Shared

Since each task shares the same sample clock, this technique requires that the two tasks operate at the same sample rate. This method of synchronization will not work for Delta Sigma or Reference Clocked Modules. These modules use a clock onboard the module to acquire their measurements. The following sections (Multidevice Task and Multirate Tasks) will discuss how to synchronize Delta Sigma Modules.

4. Multidevice Task

To help simplify synchronization configuration, NI-DAQmx allows multiple Analog Input channels from various modules to be placed in the same task (<http://zone.ni.com/reference/en-XX/help/370466AD-01/mxdevconsid/cdaqmultidevice/>). To place various modules in the same task, specify multiple channels across different modules in the Physical Channel input to the Create Virtual Channel VI. Channels from different modules should be included in the same DAQmx Create Channel if they are intended to have the same measurement configuration (voltage range, and units) and the same timing and triggering information.

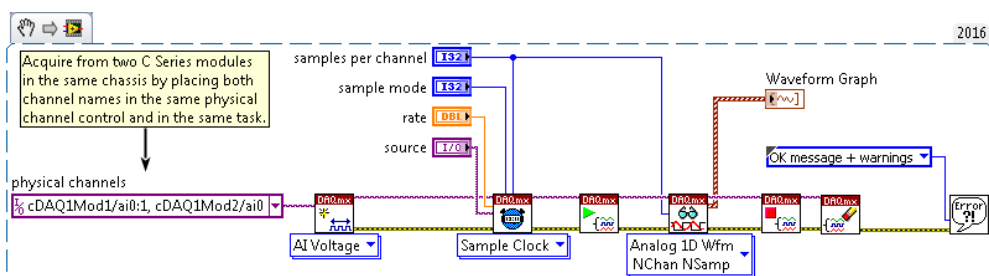


Figure 4: Using different modules in the same task

If the modules are taking different measurements (such as voltage or current), require different measurement ranges, or need to use different measurement units then use multiple instances of the DAQmx Create Virtual Channel VI as seen below.

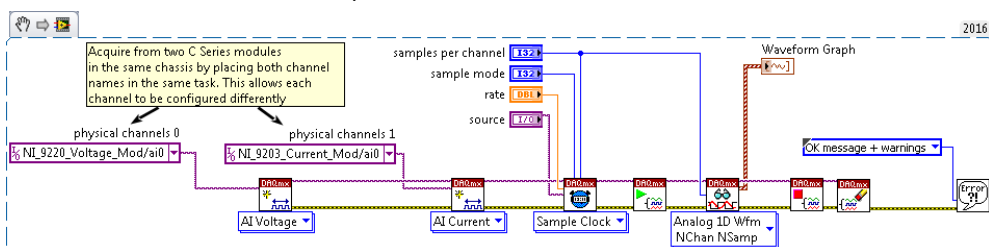


Figure 5: Multiple DAQmx Create Virtual Channel VIs used for one task to specify different measurement types

You can mix different types of analog input modules (SAR, Delta Sigma, Slow Sampled, and Reference Clocked) in the same task and allow the NI-DAQmx Driver to synchronize them for you. With Multidevice Tasks, DAQmx chooses the best possible method to synchronize the modules. An explanation is provided below of different scenarios and how Multidevice Task synchronizes the measurements.

Two SAR/Slow Sampled Modules - When two SAR modules, SAR and Slow Sampled modules, or two Slow Sampled Modules are placed in the same hardware timed task, the DAQmx driver shares the same chassis or controller's sample clock and start trigger between both modules. For the SAR and Slow Sampled Module configuration, the maximum rate that data is returned is not determined by the Slow Sampled Module. The driver will return duplicate samples for the Slow Sampled module to allow you to run the SAR module at the fastest allowable rate.

Two Delta Sigma Modules - When two Delta Sigma Modules are placed in the same hardware timed AI task, the DAQmx driver shares the fastest master timebase between all the modules, resets their ADCs based on the same synchronization pulse, shares a start trigger, and adjusts the reset time. The master timebase is the internal 12.8 MHz or 13.1 MHz clock that Delta Sigma Modules use to acquire measurements. The default behavior for an cDAQ-9185 or cDAQ-9189 (time-aware chassis) is that the master timebase originates from the chassis instead of the module with the fastest master timebase.

Two Reference Clocked Modules - When two Reference Clocked Modules are placed in the same hardware timed AI task, the DAQmx driver shares the reference clock from the master module to the other module, resets their ADCs based on the same synchronization pulse, shares a start trigger, and adjusts the reset time. The reference clock is the internal master timebase that the subsequent module locks onto. The default behavior for a time-aware chassis is that the reference clock originates from the chassis instead of the master module.

Reference Clocked Module and Delta Sigma Module - When a Reference Clocked Module and Delta Sigma Module are placed in the same hardware-timed AI task, the DAQmx driver routes the Reference Clocked Module's reference clock to the Delta Sigma Module's master timebase. Both modules receive a synchronization pulse from the chassis/controller that resets their respective ADCs and share a start trigger, and each module's reset time is adjusted. In a time-aware chassis, the Reference Clocked Module's reference clock is not shared with the Delta Sigma Module's master timebase rather the chassis routes a reference clock to the Reference Clocked Module and a master timebase to the Delta Sigma Module. The Reference Clocked Module (NI 9775) cannot be in record mode when placed in the same task as a Delta Sigma Module.

Delta Sigma/Reference Clock and SAR/Slow Sampled - When a SAR or Slow Sampled Module and a Delta Sigma or Reference Clocked Module are placed in the same hardware timed AI task, the DAQmx driver configures the Delta Sigma Module or Reference Clocked Module to export its internal sample clock to be used by the SAR or Slow Sampled Module. The internal sample clock is equivalent to the data rate configured for the Delta Sigma Module

$$fs = \left(\frac{fm}{(256)} \right) / n$$

or

$$fs = \frac{fm}{(4 * a * b)}$$

where

fs is the data rate

fm is the internal master timebase specified in each modules specification sheet

n/a/b are divisors.

The internal sample clock for the Reference Clocked Module can be found in the device's Datasheet. The driver will return duplicate samples for the Slow Sampled Module, which allows the Delta Sigma or Reference Clocked Module to run at a faster rate. The Reference Clocked Module (NI 9775) cannot be in record mode when placed in the same task as a SAR or Slow Sampled module.

5. Multirate Synchronization

When referencing multiple modules in one task, the same sampling rate will apply to all channels. However, some applications may require synchronization at different rates.

Two SAR Modules - You do not need to do any additional work to synchronize two SAR Modules or Slow Sampled modules acquiring at different rates in a CompactDAQ chassis or controllers besides sharing a Start Trigger. The sample clock each module uses is derived from the same signal. Thus, the measurements will be synchronized if the modules are in the same chassis.

Two DSA Modules - To synchronize two Delta Sigma Modules acquiring at different rates, set the tasks to share a common master timebase, synchronization pulse (sync pulse), and start trigger. The master timebase clock is routed from one module to another in the Figure 6. Note that the NI DAQmx Timing Property node SampleClock.Timebase.Src refers to the same 12.8 MHz or 13.1 MHz signal as the master timebase listed in the Operating Instructions and Specifications/Datasheet. This clock is also known as an oversample clock because there are many ticks of this clock required to produce one sample.

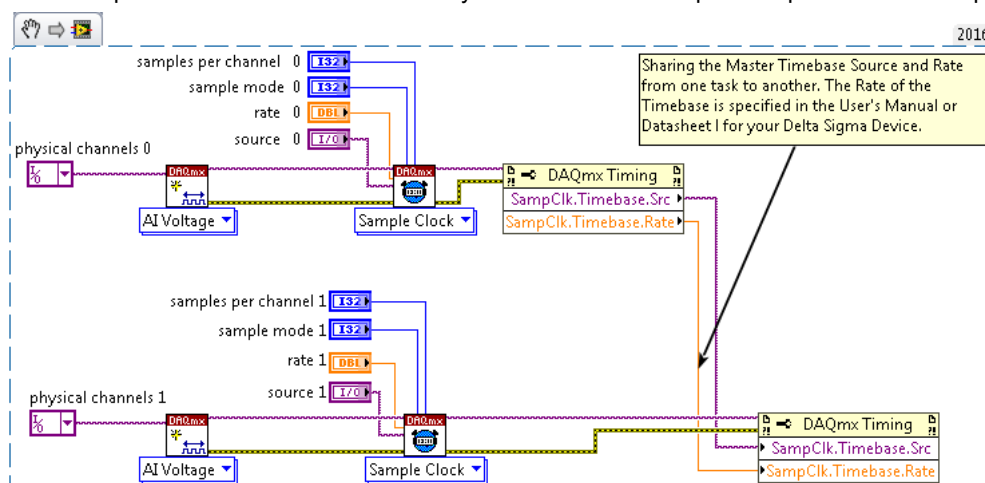


Figure 6: Sharing the over sample clock

A device that uses a Delta Sigma Module has a free-running ADC. Sharing the master timebase will ensure that measurements are derived from the same clock, but there will be an unknown phase shift due to the exact start of the ADC. The next step is sharing the synchronization pulse. The sync pulse is used to reset the ADC/DAC before it begins acquiring or generating samples. Without this signal to reset the ADC/DAC, measurements taken will not be synchronized. Below, the property nodes are shown that allow the sharing of an oversample clock and a sync pulse.

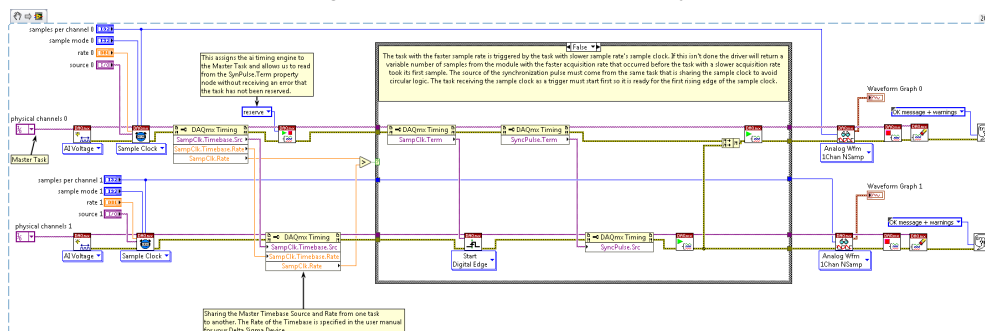


Figure 7: Sharing over sample clock, start trigger and sync pulse.

The above configuration routes the synchronization pulse from one task to another depending on which task is running at a faster rate. For multirate configuration, we use the sample clock of the slower running task to trigger the faster running task. The reasoning for this is explained in the example.

One more consideration to study is the reset time required by each module. The reset time is a delay required by a Delta Sigma Modules ADC after it has received the synchronization pulse until it begins acquiring data. Different model Delta Sigma Modules may have different reset times. You need to account for this reset time to ensure each device has gone through the reset sequence and begins to acquire or output samples at the same time. To properly account for potentially different reset times, the reset time required for each module is checked and, if necessary, a delay added to modules with a shorter reset time. Therefore, the sum of the reset time and the delay should be the same for each module.

The time needed to reset the ADC can be found using the DAQmx Timing Property Node SyncPulse.ResetTime. To ensure all ADCs end their reset cycle coincidentally, a comparison is made between both tasks' reset time. We take the task with the largest Reset Time and subtract the Reset Time of the task with the smaller Reset Time. The result is the time where the one task has begun to acquire data while the other task is still going through its reset cycle. To account for this delay we use another NI-DAQmx property called SyncPulse.ResetDelay. This property allows a delay in the sync pulse that will be sent to each module. As previously stated the reset time is the delay between when the ADC receives a sync pulse and begins to acquire data. To best use this property, you should set this equal to the difference between the module that has the greatest reset time and the reset time of the module. This will shift the synchronization pulse of a module that takes less time to reset thus ensuring that both modules with a slow and fast reset time end their reset sequence at the same time. The code below details how to get the reset time and determine the appropriate reset delay.

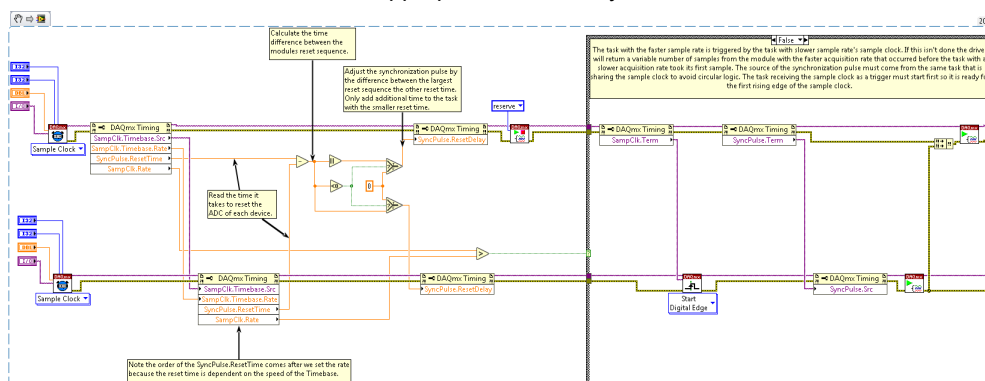


Figure 8: Sync Pulse reset time query and calculating the reset delay

The NI-DAQmx property node SyncPulse.ResetDelay has a maximum input of .013 seconds, which may not be suitable for large differences in the Reset Delay values of differing modules. Note the order of the property node when polling for the SyncPulse.ResetTime on the task receiving the master timebase. The value will change depending on a 13.1 MHz Timebase or 12.8 MHz Timebase. Thus, it is important to call the SyncPulse.ResetTime on this task after the module has been informed of the Master Timebase's rate.

When searching for the properties SyncPulse.ResetDelay and SyncPulse.ResetTime, you will also encounter SyncPulse.SyncTime and SyncPulse.MinDelaytoStart. Most developers will not use this property but may encounter them due to their similarity in name and proximity in location. SyncPulse.SyncTime is the SyncPulse.ResetTime y plus the value you enter for SyncPulse.ResetDelay. The value of SyncPulse.SyncTime is rounded to the nearest millisecond. SyncPulse.MinDelaytoStart is the time between when the Synchronization Pulse is issued and master task issues the start trigger. This property is equal to SyncPulse.SyncTime unless explicitly set.

Two Reference Clocked Modules -To synchronize two Reference Clocked Modules at different rates (like two DSA Modules), distribute the master task's reference clock, synchronization pulse, and sample clock to other tasks. Note that the NI DAQmx Timing Property node RefClick.Src refers to the same Internal master timebase listed in the Datasheet.

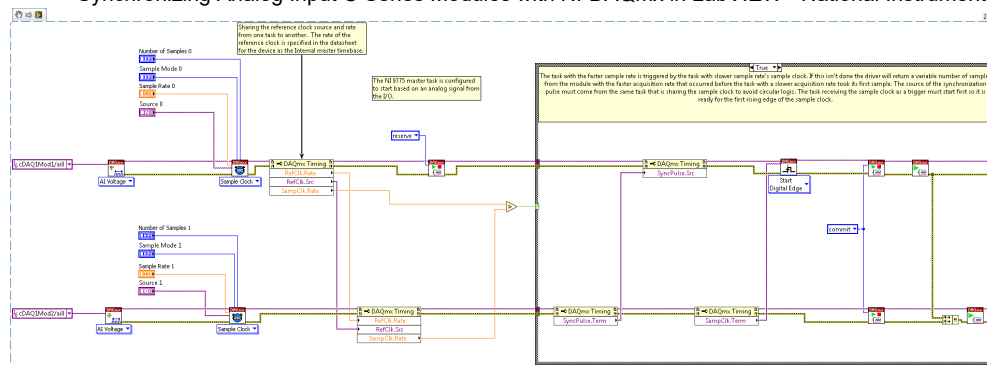


Figure 9: Synchronizing Two Reference Clocked Modules

Delta Sigma Module and SAR/Slow Sampled Module – To synchronize a Delta Sigma Module and SAR/Slow Sampled Module, share the Delta Sigma Module's master timebase and sample clock with the SAR/Slow Sampled Module. Since the SAR/Slow Sampled Module cannot acquire at 12.8 MHz or 13.1 MHz the Delta Sigma's master timebase is shared with the master timebase of the SAR/Slow Sampled Module as depicted in figure 10. The sample clock of a SAR/Slow Sampled Module is typically derived from the CompactDAQ Chassis or Controller's onboard clock. This method replaces the chassis or controller clock with the 12.8 MHz or 13.1 MHz clock from the Delta Sigma Module.

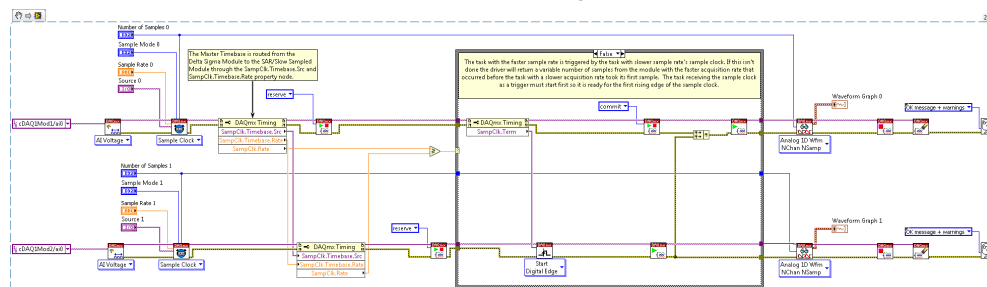


Figure 10: Synchronizing a Delta Sigma Module and SAR/Slow Sampled Module

Reference Clocked Module and Delta Sigma Module– To synchronize a Reference Clocked Module and Delta Sigma Module, share the Reference Clocked Module's reference clock, synchronization pulse, and sample clock with the Delta Sigma module. Like the section on multirate synchronization of two Delta Sigma Modules, you must consider the difference between the reset delays.

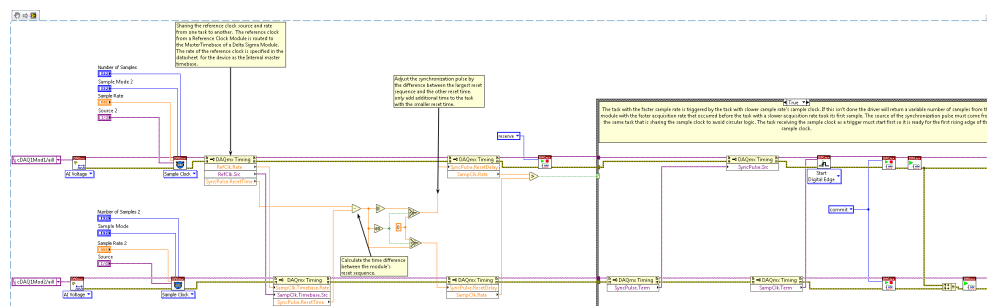


Figure 11: Synchronizing a Reference Clocked Module and Delta Sigma Module

Reference Clocked Module and SAR/Slow Sampled Module – To synchronize a Reference Clocked Module and SAR/Slow Sampled Module, share the Reference Clocked Module's reference clock and sample clock with the SAR/Slow Sampled Module. Since the SAR/Slow Sampled Module cannot acquire at 12.8 MHz, the Reference Clocked Module's reference clock is shared with the master timebase of the SAR/Slow Sampled Module. The sample clock of a SAR/Slow Sampled Module is typically derived from the CompactDAQ Chassis or Controller's onboard clock. This method replaces the chassis or controller clock routed to the SAR/Slow Sampled Module with the Internal Master Timebase from the Reference Clocked Module.

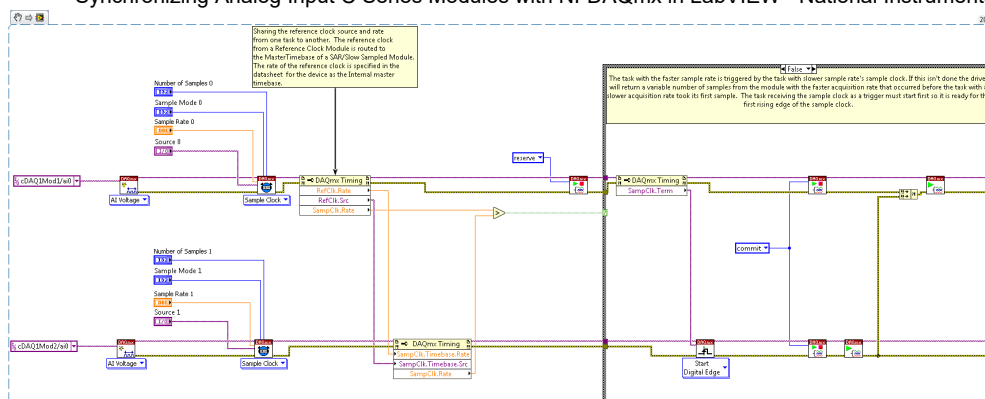


Figure 12: Synchronizing a Reference Clocked Module with a SAR/Slow Sampled Module

6. MultiChassis Tasks

Some measurement applications require mixed measurements and distributed systems, which might require more than one CompactDAQ chassis or controller. There are three main methods of multichassis synchronization.

Multi Chassis Synchronization Using BNC/SMB Connectors or Digital Modules - One method to achieve synchronization between chassis is to use the two BNC connectors located on the cDAQ-9178, cDAQ-9188, cDAQ-9188XT, and cDAQ-9179 chassis or one SMB Connector on the cDAQ-9132, cDAQ-9133, cDAQ-9134, cDAQ-9135, cDAQ-9136, cDAQ-9137, cDAQ-9185, and cDAQ-9189 to import and export timing signals.



Figure 13: NI cDAQ-9178 with two BNC ports can be used to import or export timing and trigger signals.

The BNC or SMB terminals of different chassis can be connected to route triggers and timing signals. Using the DAQmx driver, signals can be exported from a master task by using the NI-DAQmx Export Signal Property Node. On tasks (in another chassis), the Sample Clock and Start Trigger are configured to use the BNC or SMB terminals to accept signals.

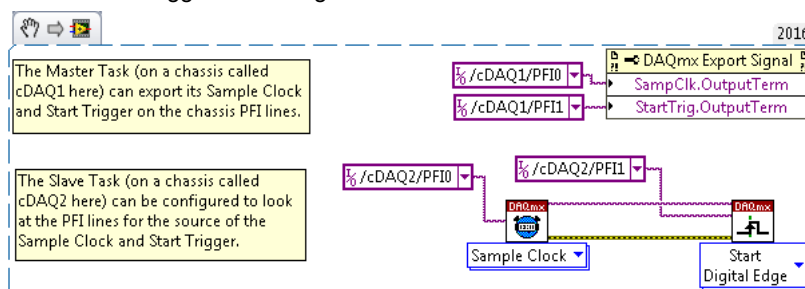


Figure 14: Configuring the BNC Ports on a CompactDAQ Chassis

Parallel/Correlated Digital IO Modules (<http://digital.ni.com/public.nsf/allkb/3114EE73A39F1300862578BE0059A895>) could also be used to receive or send trigger and timing signals across the chassis. These Digital I/O modules are preferred when there isn't sufficient BNC or SMB ports on the NI Compact DAQ Controllers and Chassis or the signal bandwidths exceeds that of the BNC or SMB connector (1MHz).

Multichassis Synchronization Using the NI 9469 - The above method to configure timing and trigger signals across the chassis BNC connectors or a Parallel/Correlated Digital IO Modules may work for many applications. However, an alternative method using the NI 9469 is preferable in the following circumstances:

- Long distance between chassis
- High bandwidth of the exported signals
- Complex topology of synchronization

The NI 9469 allows simplified signal routing and coding and can offer synchronization to SAR, Slow-Sampled, Delta Sigma, Digital Input and Output, Analog Output modules in differing chassis topologies. The connections between the NI 9469 cards can be configured in Measurement & Automation Explorer (MAX) as shown in the following figure. Please see the related link section at the end of this document for more details on how to configure the NI 9469.

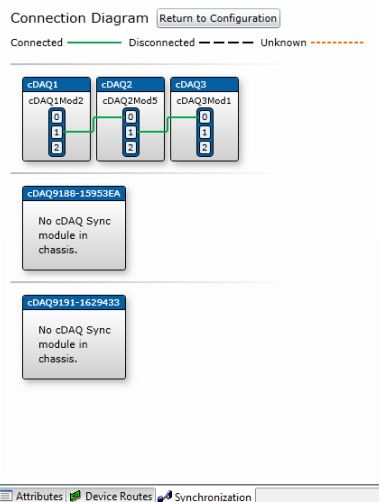


Figure 15: Configuration of NI 9469 in Measurement & Automation Explorer

Once the NI 9469 is configured in MAX (using NI-DAQmx version 9.9 or newer) any module in separate chassis can be added to the same task to greatly simplify the code required for synchronization. The following figure shows how this method of multichassis synchronization.

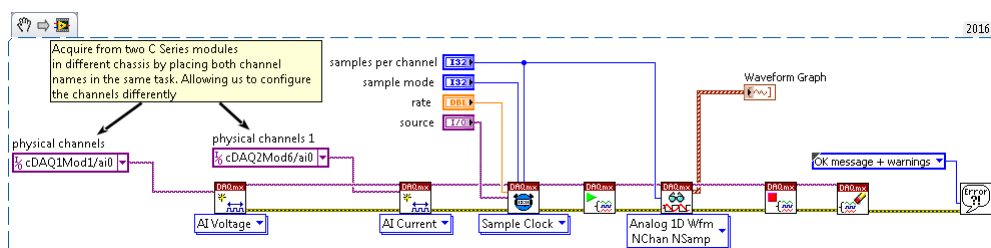


Figure 16: Multichassis Synchronization Code with NI 9469 Configured in MAX

The background process for how the modules are synchronized in different chassis is like the stated methods in the Multirate Tasks and Multidevice Tasks Sections. The limitation for which modules in multiple chassis can be synchronized automatically using this approach exists and is documented (<http://zone.ni.com/reference/en-XX/help/370466AD-01/mxdevconsid/cdaqmultidevice/#guid-f0f5745c-ecd4-4d8b-b347-7a831729d3c1>).

Multichassis Synchronization with a Time Aware Chassis – The above methods were all based on signal based synchronization. With a time-aware chassis like the cDAQ-9185 and cDAQ-9189, a different and simpler approach is needed to synchronize modules in different chassis. This alternative method is preferable in the following circumstances:

- Long distance between chassis
- High bandwidth of the exported signals
- Complex synchronization topology
- Synchronizing to external time sources

The cDAQ-9185 and cDAQ-9189 offer three distinct clocks (10 MHz, 12.8 MHz, and 13.1 MHz) to Delta Sigma and Reference Clocked Modules. When using Delta Sigma or Reference Clocked Modules in these chassis DAQmx will default to using the chassis clocks to acquire or generate signals instead of the clocks onboard the modules.

When the time-aware chassis share the same grand master, all clocks onboard the chassis are synchronized to this grand master. Thus, clocks across multiple chassis are synchronized to each other. This process is done automatically and without user input. Since each module in the chassis will default to using the chassis backplane clocks for generation or acquisition, clocks used by each module (master timebase, sample clock, and/or reference clock) will all be synchronized. To synchronize the module acquisitions or generations between chassis, ensure that the Start Trigger is sent at the same moment.

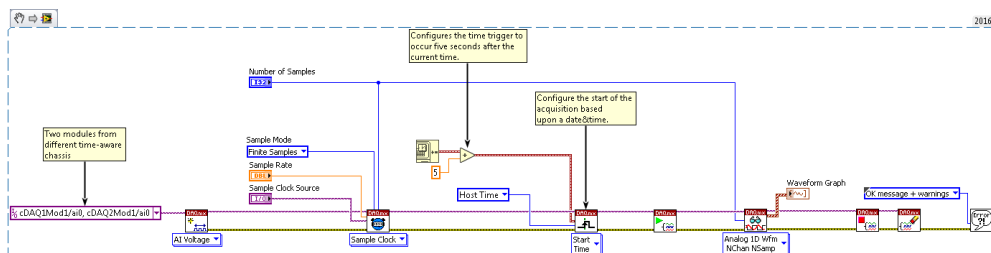


Figure 17: Multichassis Synchronization with the cDAQ-9185 or cDAQ-9189

Placing all the modules from various chassis in one task to achieve synchronization is not a feasible solution for certain modules (<http://zone.ni.com/reference/en-XX/help/370466AD-01/mxdevconsid/cdaqmultidevice/#guid-f0f5745c-ecd4-4d8b-b347-7a831729d3c1>) and locks all the modules to a single sample rate. To work around this issue, use separate tasks, configure a start trigger for each task, and set the start trigger to begin at the same time.

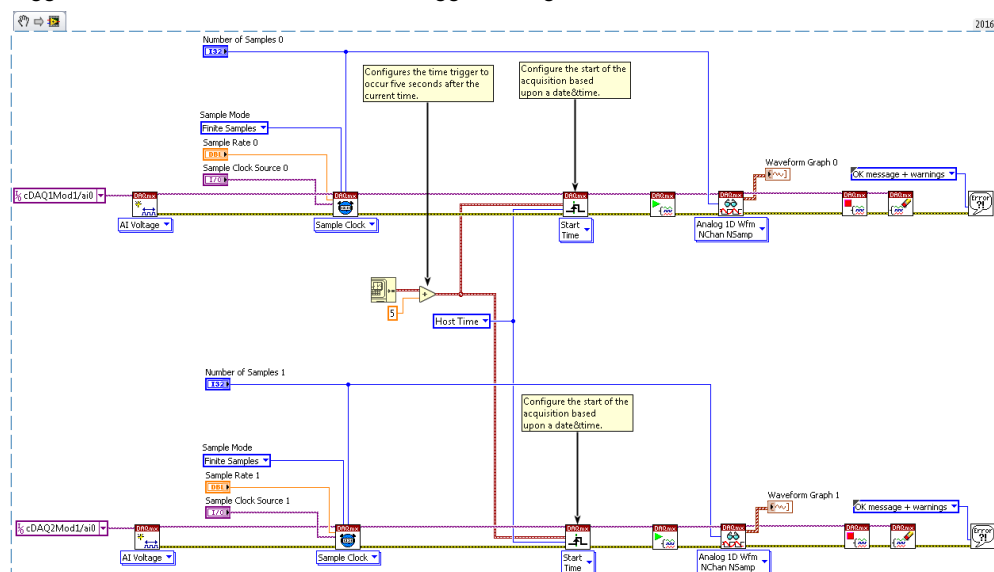


Figure 18: Multichassis and Multirate Synchronization with the cDAQ-9185 or cDAQ-9189

When Delta Sigma and/or Reference Clocked Module Modules are running at different rates, we must account for the difference in reset times and the variable number of samples that occur due to the difference in sample rates. The difference in reset times are accounted for by sending the synchronization pulse x seconds before the start trigger, where x is equal to the longer reset time. The variable number of samples is accounted for by analyzing the ratio between the two acquisition periods and adjusting the synchronization pulse based on this analysis. The subvis necessary for this example are attached to this document in the top right hand corner.

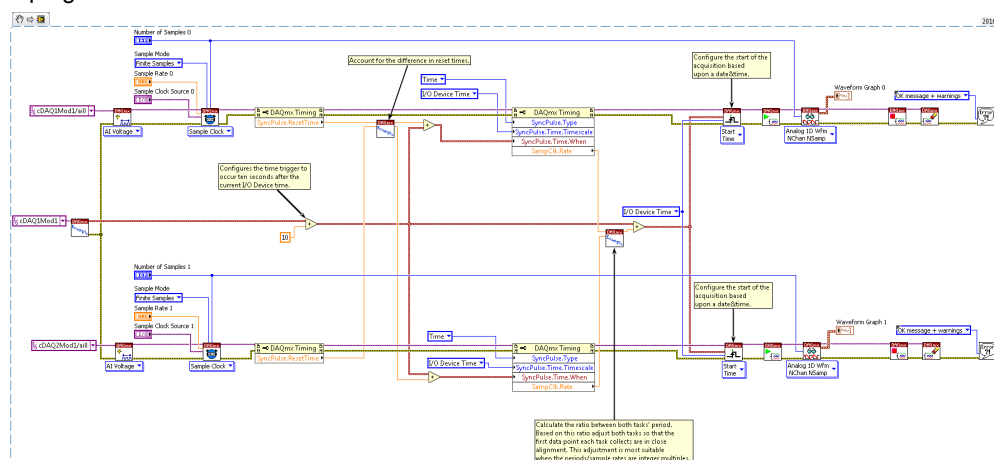
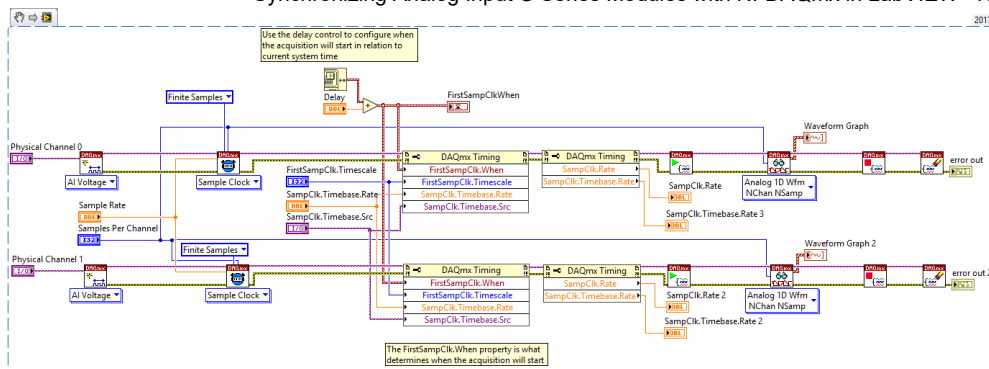


Figure 19: Multichassis and Multirate Synchronization with the cDAQ-9185 or cDAQ-9189 for Delta Sigma and Reference Clocked Modules

I/O Device time is used in the above example due to the accuracy in translation between Host Time and I/O Device time (<http://zone.ni.com/reference/en-XX/help/370466AD-01/TOC96.htm>).

As mentioned earlier, Delta Sigma and Reference Clocked Modules will default to the available onboard (13.1 MHz, 12.8 MHz, and 10 MHz) chassis clocks when in time-aware chassis such as the cDAQ-9185 or cDAQ-9189. Therefore, when attempting to perform a multirate synchronization in a single cDAQ-9185 or cDAQ-9189, you only need to share a Start Trigger and/or synchronization pulse to synchronize the two tasks. The Sample Clocks, Master Timebase, and Reference Clocks originate from the chassis and are synchronized, thus measurements using these clocks will be synchronized.

Starting in DAQmx 18.6 a new DAQmx Timing Property is available called First Sample Clock When. This property allows you to set the time when the first sample clock will be generated in terms of System Time or I/O Device Time. This property performs the calculations demonstrated in Figure 19 internally, including start trigger time, difference in reset times, and first sample alignment when using multiple rates. The First Sample Clock When property can be used to align acquisition start times on any combination of module types including DSA, SAR, Slow Sampled and FieldDAQ devices. The example below demonstrates how to use the property.



7. Conclusion

Throughout this tutorial several synchronization options have been demonstrated. Initially, tasks were started with software timed commands. The level of synchronization can then be improved by sharing a hardware trigger between tasks. Drift between the tasks is prevented by sharing a sample clock from one task to another. When synchronizing at the same rate, programming can be simplified by referring to several modules within the same task. If different rates are required, special considerations depend on what type of module is being used. Finally, in a multiple chassis scenario, options include routing the timing and trigger manually, a more automated process involving the NI 9469, or using a time aware chassis. The options outlined above represent an overview of several different ways to accomplish synchronization using the C Series platform applicable for the timing requirements in many application areas.

8. Additional Resources

- C Series Modules:
 - Delta Sigma Modules: 9202 (<http://www.ni.com/en-us/support/model.ni-9202.html>), 9218 (<http://sine.ni.com/nips/cds/view/p/lang/en/nid/212730>), 9225 (<http://sine.ni.com/nips/cds/view/p/lang/en/nid/208795>), 9227 (<http://sine.ni.com/nips/cds/view/p/lang/en/nid/208794>), 9229 (<http://sine.ni.com/nips/cds/view/p/lang/en/nid/208796>), 9230 (<http://sine.ni.com/nips/cds/view/p/lang/en/nid/213063>), 9232 (<http://sine.ni.com/nips/cds/view/p/lang/en/nid/210281>), 9234 (<http://sine.ni.com/nips/cds/view/p/lang/en/nid/208802>), 9235 (<http://sine.ni.com/nips/cds/view/p/lang/en/nid/208790>), 9236 (<http://sine.ni.com/nips/cds/view/p/lang/en/nid/208790>), 9237 (<http://sine.ni.com/nips/cds/view/p/lang/en/nid/208791>), 9238 (<http://sine.ni.com/nips/cds/view/p/lang/en/nid/212512>), 9239 (<http://sine.ni.com/nips/cds/view/p/lang/en/nid/208797>), 9242 (<http://sine.ni.com/nips/cds/view/p/lang/en/nid/212305>), 9244 (<http://sine.ni.com/nips/cds/view/p/lang/en/nid/212307>), 9246 (<http://sine.ni.com/nips/cds/view/p/lang/en/nid/213202>), 9247 (<http://sine.ni.com/nips/cds/view/p/lang/en/nid/213199>), 9250 (<http://www.ni.com/en-us/support/model.ni-9250.html>), 9251 (<http://sine.ni.com/nips/cds/view/p/lang/en/nid/213580>)
 - Slow Sampled Modules: 9207 (<http://sine.ni.com/nips/cds/view/p/lang/en/nid/208168>), 9208 (<http://sine.ni.com/nips/cds/view/p/lang/en/nid/207770>), 9209 (<http://sine.ni.com/nips/cds/view/p/lang/en/nid/213127>), 9210 (<http://www.ni.com/en-us/shop/select/c-series-temperature-input-module?modelId=148563>), 9211 (<http://sine.ni.com/nips/cds/view/p/lang/en/nid/208787>), 9212 (<http://sine.ni.com/nips/cds/view/p/lang/en/nid/212727>), 9213 (<http://sine.ni.com/nips/cds/view/p/lang/en/nid/208788>), 9214 (<http://sine.ni.com/nips/cds/view/p/lang/en/nid/209414>), 9216 (<http://sine.ni.com/nips/cds/view/p/lang/en/nid/213130>), 9217 (<http://sine.ni.com/nips/cds/view/p/lang/en/nid/208804>), 9219 (<http://sine.ni.com/nips/cds/view/p/lang/en/nid/208789>), 9224 (<http://www.ni.com/en-us/support/model.ni-9224.html>), 9226 (<http://sine.ni.com/nips/cds/view/p/lang/en/nid/213138>), 9228 (<http://www.ni.com/en-us/support/model.ni-9228.html>)
 - SAR Modules: 9201 (<http://sine.ni.com/nips/cds/view/p/lang/en/nid/208798>), 9203 (<http://sine.ni.com/nips/cds/view/p/lang/en/nid/208805>), 9205 (<http://sine.ni.com/nips/cds/view/p/lang/en/nid/208800>), 9206 (<http://sine.ni.com/nips/cds/view/p/lang/en/nid/208801>), 9215 (<http://sine.ni.com/nips/cds/view/p/lang/en/nid/208793>), 9220 (<http://sine.ni.com/nips/cds/view/p/lang/en/nid/210836>), 9221 (<http://sine.ni.com/nips/cds/view/p/lang/en/nid/208799>), 9222 (<http://sine.ni.com/nips/cds/view/p/lang/en/nid/209142>), 9223 (<http://sine.ni.com/nips/cds/view/p/lang/en/nid/209139>)
 - Reference Clocked Module: 9775 (<http://www.ni.com/en-us/support/model.ni-9775.html>)
 - Chassis Synchronization Module: 9469 (<http://sine.ni.com/nips/cds/view/p/lang/en/nid/210832>)
- Time Aware CompactDAQ Chassis: cDAQ-9185 (<http://www.ni.com/en-us/support/model.cdaq-9185.html>) and cDAQ-9189 (<http://www.ni.com/en-us/support/model.cdaq-9189.html>)
- Configuring the NI 9469 in Measurement & Automation Explorer (<http://www.ni.com/tutorial/52807/en/>)
- Programmatically Configuring the NI 9469 in LabVIEW (<http://www.ni.com/product-documentation/52809/en/>)
- Learn 10 Functions in NI-DAQmx and Handle 80 Percent of Your Data Acquisition Applications (<http://www.ni.com/white-paper/2835/en/>)
- NI-DAQmx-Based TSN Synchronization Accuracy Explained (<http://www.ni.com/white-paper/53851/en/>)

- How to Achieve High-Accuracy Measurements With NI-DAQmx-Based TSN Devices (<http://www.ni.com/white-paper/53852/en/>)
- Designing Distributed TSN Ethernet-Based Measurement Systems (<http://www.ni.com/white-paper/53850/en/>)
- NI CompactDAQ Advisor (<http://ohm.ni.com/advisors/compactdaq/pages/common/intro.xhtml>)