SETUP GUIDE

PathWave Test Sync Executive 2022

System Setup Guide

PATHWAVE

This guide describes how to set up a Multi-Chassis system using Keysight PXIe Chassis and PXIe System Synchronization Modules. It also describes how to set a reference clock and how to configure everything in your software code by using the HVI API (Application Programming Interface) delivered by PathWave Test Sync Executive.

KEYSIGHT TECHNOLOGIES

Table of Contents

System Setup Guide

This guide describes how to set up a single or multi-chassis system using Keysight PXIe Chassis and PXIe System Synchronization Modules. It also describes how to set a reference clock and how to configure everything in your software code by using the HVI API delivered by PathWave Test Sync Executive.

Keysight System Synchronization Modules (SSM)

KS2201A 2022 release introduces new multi-chassis topologies that use the Keysight M9032A/M9033A PXIe *System Synchronization Modules* (SSMs). The previous means of interconnecting multiple PXI chassis using M9031A modules is discontinued starting from the KS2201A 2022 release. Compared to the discontinued M9031A module, the SSM has a much wider range of functions including:

- Distribution of a precise reference clock.
- Management of *Fast Data Sharing* (FDS).
- Chassis interconnectivity.
- Synchronization of all the PXI instruments in the multi-chassis.

M9032A and M9033A PXIe SSM Overview

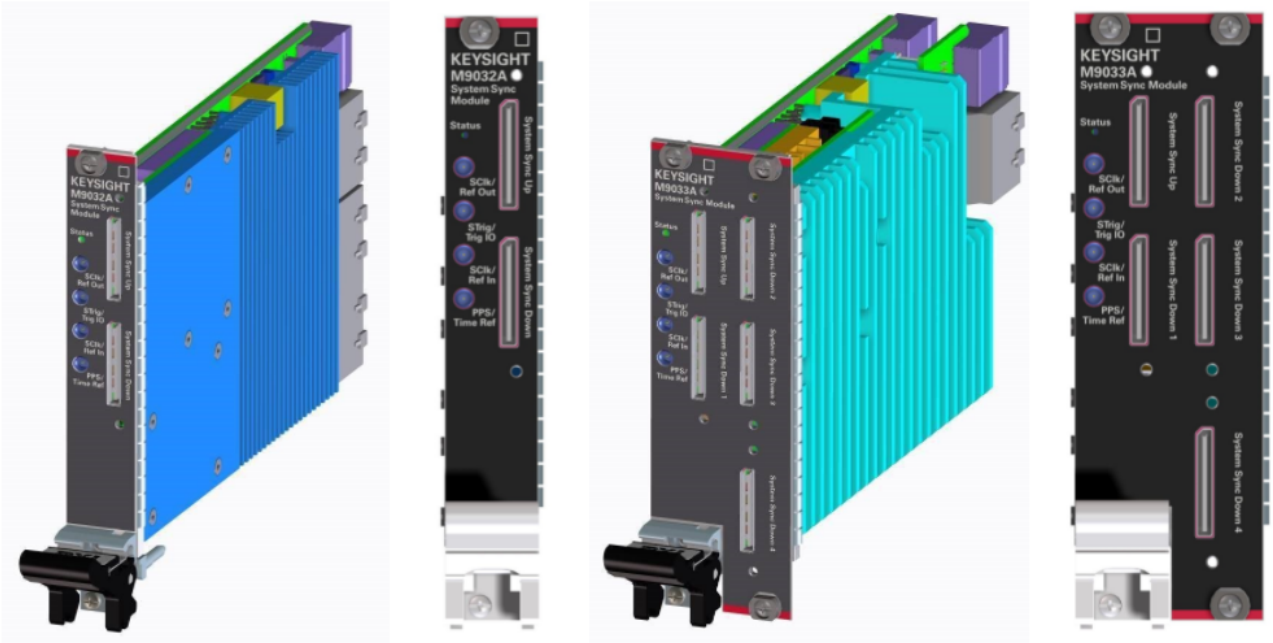
The M9032A/M9033A are PXIe *System Synchronization Modules* (SSM). These include an onboard high-quality 10MHz Oven Controlled Crystal Oscillator (OCXO) to achieve a very precise synchronization among various measurement instruments distributed across different chassis. The M9032A/M9033A System Synchronization Module functionalities can only be successfully deployed on chassis compliant with the *PXI-Express* (PXIe) standard. The SSM must be inserted in the timing slot of the PXIe chassis.

Keysight PXIe System Synchronization Module is available in two form factors, which only differ in their connectivity capabilities:

- M9032A is a one-slot PXIe System Synchronization Module with 1 System Sync Upstream and 1 System Sync Downstream ports.
- M9033A is a two-slot PXIe System Synchronization Module with 1 System Sync Upstream and 4 System Sync Downstream ports.

For further information about these SSMs including detailed performance specifications, see the M9032A/M9033A User's Guide, available at Keysight PXI Products.

The following image shows the physical M9032A and M9033A SSMs:


KEYSIGHT
M9032A
System Sync Module
Status
SClk/ Ref Out
STrig/ Trig IO
SClk/ Ref In
PPS/ Time Ref
System Sync Up
System Sync Down
KEYSIGHT
M9033A
System Sync Module
System Sync Down 2
System Sync Down 3
System Sync Down 1
System Sync Down 4

M9032A and M9033A Connectivity

Front Panel

The M9032A and M9033A Front Panel contains various connectors that can be used for both multi-chassis interconnection and configuration of the reference clock source.

Front Panel SMP IOs

FP (Front Panel) SMP (Sub Miniature Push-on) connectors are:

SClk/Ref Out:

Outputs a copy of the system clock or a reference clock signal.

STrig/Trig IO:

Receives an arbitrary trigger signal.

SClk/Ref In:

Receives the reference clock signal.

PPS/Time Ref:

Receives a Pulse Per Second (PPS) signal.

The front panel SMP connectors can be used to share input and output reference clocks.

System Sync ports

System Sync ports use PCIe *Optical Copper Link* (OCuLink) connectors. System Sync ports are used for chassis interconnection and synchronization in the multi-chassis system. The signals in the System Sync include:

- Clocking (System Sync only).
- Triggering.
- Data.

The different SSM models have the following front panel System Sync ports:

The M9032A has 2 System Sync ports:

- 1 System Sync Upstream.
- 1 System Sync Downstream.

The M9033A has 5 System Sync ports:

- 1 System Sync Upstream.
- 4 System Sync Downstream.

Each System Sync Downstream port can connect to the System Sync Upstream port of another SSM placed in a different chassis. For more information, see the section below about Inter/Intra-chassis Connectivity.

PXIe Backplane DSTAR Connectivity

The M9032A and M9033A are placed in the Timing Slot of a PXIe chassis which enables them to support the DSTAR connectivity built-in the chassis.

DSTARA/B/C are multi-instrument point to point connections inside a chassis. DSTARA is used to carry the clock signal. DSTARB and DSTARC carry trigger or data signals.

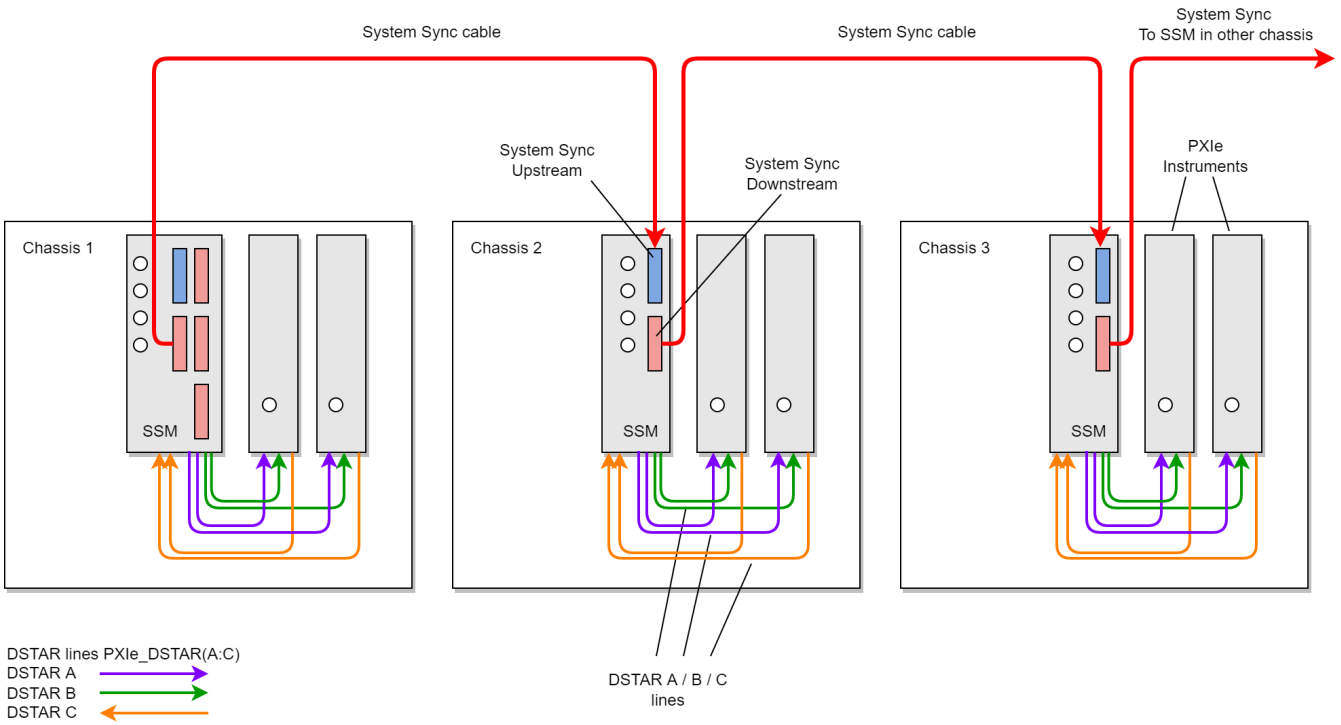
Inter/Intra-chassis connectivity, Synchronization and Data-Sharing Functionalities

An SSM can enable both multi-chassis and multi-instrument interconnections. With these connections, SSMs enable synchronization and data sharing across all the instruments in a multi-chassis system.

- Multi-chassis interconnections are made with System Sync connections using their capability to interconnect two SSMs together through their System Sync Downstream/Upstream ports.
- Intra-chassis, multi-instrument interconnections are made with PXIe DSTARA/B/C connections. The SSM can share the precise reference clock over the DSTARA signal.

The following diagram shows a 3 chassis system connected with System Sync cables and DSTARA/B/C signals in each chassis:

3 Chassis system with System Sync



Data can be shared across System Sync and DSTAR connections in several different ways:

- The reference clock can be shared between two interconnected SSMs using the System Sync connection between System Sync Downstream/Upstream ports.

- The System Sync connection can share the signals sent over the PXI_TRIG[0:7] trigger buses, from one SSM to the next. This enables the SSMs to share PXI sync resources used by PathWave Test Sync Executive for the *Hard Virtual Instrument* (HVI) across the different chassis.
- System Sync connections can route data shared using *Fast Data Sharing* (FDS) between PXIe instruments.
- The SSM can send the data between two modules located in the same chassis using the DSTARB/C signals.
- Data can be sent through the System Sync connections to route it to instruments located in a different chassis.

Configuring a System with SSMs and System Sync Connectivity

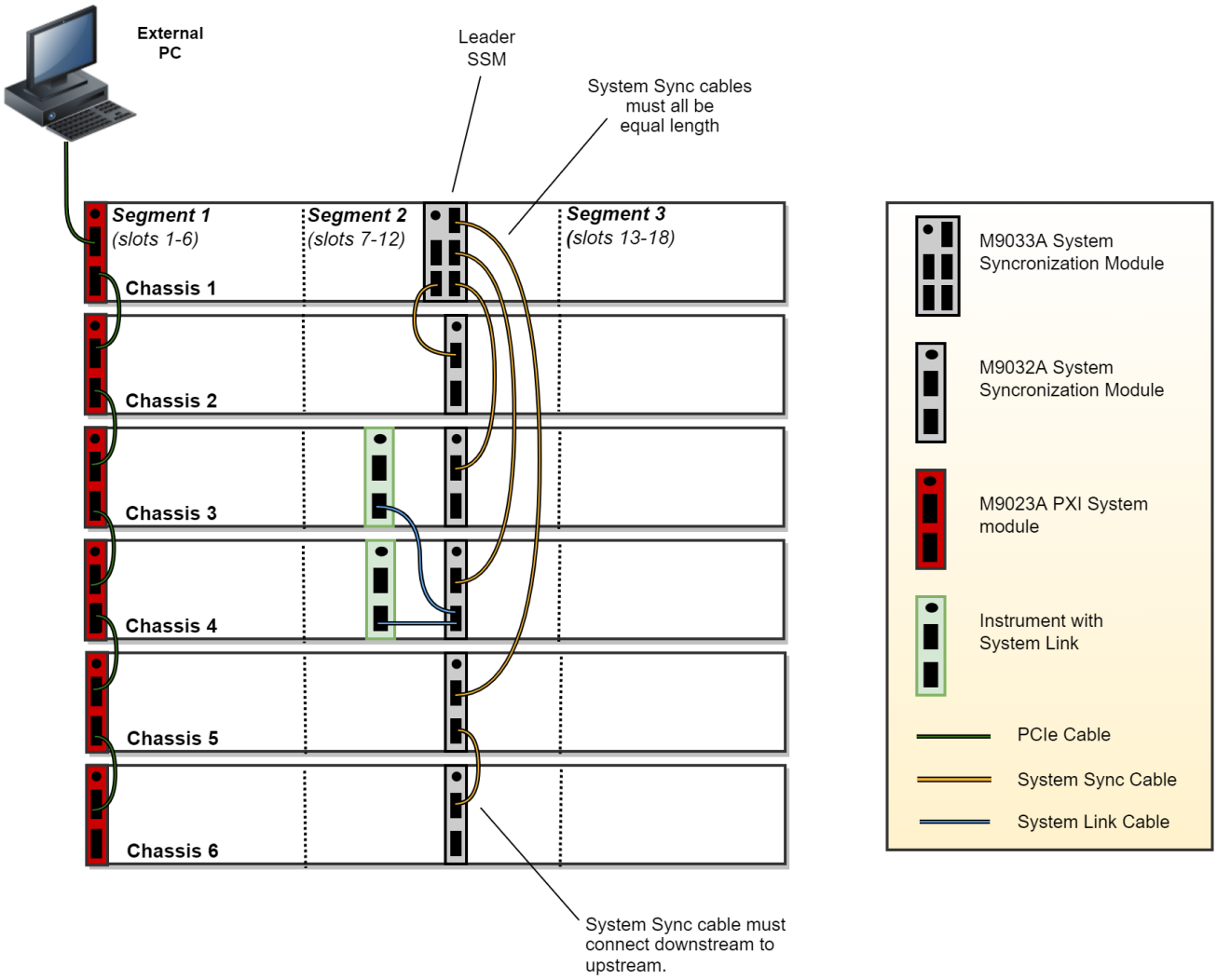
In a multi-chassis system connected with Keysight PXIe System Synchronization Modules, you must include one SSM in each chassis that is part of the system. Each SSM must be inserted in the **timing slot** of your chassis. This is typically slot 10 in Keysight 18-slot chassis, but it can be a different slot number in different chassis models. The SSMs are connected to each other with System Sync cables.

One SSM is automatically chosen as a leader and it is used to synchronize all the instruments in the multi-chassis system. The SSM chosen as leader is the SSM that has no incoming connection to its System Sync Upstream port. The leader SSM distributes a replica of the reference clock signal to the SSMs located in the other chassis. It does this through point-to-point connections between System Sync Downstream/Upstream ports. In the example multi-chassis system shown in the following diagram, the leader SSM is in Chassis 1.

A multi-chassis PXIe system may be configured to use many different reference options. For a list of those options and descriptions of how to configure them, see the section *Clocking* in this document. For one of those reference options, an SSM is chosen as a leader and uses its internal Oven Controlled Crystal Oscillator (OCXO) clock to synchronize all the instruments included in the multi-chassis system.

NOTE A Multi-chassis system based on the older M9031A modules required an external reference clock generator to distribute the precise common 10 MHz reference clock signal across different chassis. In the new multi-chassis topology delivered by PathWave Test Sync Executive 2022, the SSM assumes the function of the **reference clock signal generator/distributor**, by sharing a reference clock generated by an internal PLL. This PLL can be fed by different sources (as explained later in this document) including the OCXO inside the SSM, which generates a 10 MHz sine wave. An external 10 or 100 MHz reference signal can still be connected to the SSM SClk / Ref Input port, to sync it together with other clusters.

The following diagram shows an example of a 6 chassis system connected with SSMs. In this system an M9033A SSM in chassis 1 distributes the reference clock to four M9032A SSMs located in each of the other chassis. The SSM in chassis 5 also forwards the clock to a sixth chassis.



This following code shows how to use the HVI Python API to define and use the SSMs in the multi-chassis system shown in the diagram. Each System Sync Downstream port connects to the System Sync Upstream port of another System Sync Module in a different chassis.

The first step is to define the SSMs placed in each of the chassis during the system definition phase.

```
# Create system definition object
my_system = kthvi.SystemDefinition("MySystem")
#
# Define System Sync Modules
resource_id_ssm_1 = 'PXI0::CHASSIS1::SLOT10::INSTR'
resource_id_ssm_2 = 'PXI0::CHASSIS2::SLOT10::INSTR'
resource_id_ssm_3 = 'PXI0::CHASSIS3::SLOT10::INSTR'
resource_id_ssm_4 = 'PXI0::CHASSIS4::SLOT10::INSTR'
resource_id_ssm_5 = 'PXI0::CHASSIS5::SLOT10::INSTR'
resource_id_ssm_6 = 'PXI0::CHASSIS6::SLOT10::INSTR'
#
# In the options, SSMs are set to be simulated with Simulate=true and there are a number of
parameters.
# For the hardware SSM instruments, set options to an empty string.
options1 = "Simulate=true,DriverSetup=Model=M9033A"
options2 = "Simulate=true,DriverSetup=Model=M9032A"
options3 = "Simulate=true,DriverSetup=Model=M9032A"
options4 = "Simulate=true,DriverSetup=Model=M9032A"
options5 = "Simulate=true,DriverSetup=Model=M9032A"
options6 = "Simulate=true,DriverSetup=Model=M9032A"
#
sync_module_1 = my_system.interconnects.add_sync_module(resource_id_ssm_1, options1)
sync_module_2 = my_system.interconnects.add_sync_module(resource_id_ssm_2, options2)
sync_module_3 = my_system.interconnects.add_sync_module(resource_id_ssm_3, options3)
sync_module_4 = my_system.interconnects.add_sync_module(resource_id_ssm_4, options4)
sync_module_5 = my_system.interconnects.add_sync_module(resource_id_ssm_5, options5)
sync_module_6 = my_system.interconnects.add_sync_module(resource_id_ssm_6, options6)
```

NOTE In the HVI System Definition phase, the SSMs are added to the interconnects collection by using their resource ID and options. Same as for the chassis, it is not necessary to open objects representing the System Sync Modules (SSMs) that are included in the multi-chassis system.

The next step is to define the interconnections among the System Sync Downstream/Upstream ports of each pair of SSMs. The SSM System Sync ports can only be connected Downstream to Upstream.

```
# Define connections among System Sync connectors of the SSMs
#
# Connect SSM 1 to SSM 2
ssm1_downstream_sync1 = sync_module_1.connectivity.systemsync_downstream[0]
ssm2_upstream_sync    = sync_module_2.connectivity.systemsync_upstream[0]
ssm1_downstream_sync1.set_connection(ssm2_upstream_sync)
#
# Connect SSM 1 to SSM 3
ssm1_downstream_sync2 = sync_module_1.connectivity.systemsync_downstream[1]
ssm3_upstream_sync    = sync_module_3.connectivity.systemsync_upstream[0]
ssm1_downstream_sync2.set_connection(ssm3_upstream_sync)
#
# Connect SSM 1 to SSM 4
ssm1_downstream_sync3 = sync_module_1.connectivity.systemsync_downstream[2]
ssm4_upstream_sync    = sync_module_4.connectivity.systemsync_upstream[0]
ssm1_downstream_sync3.set_connection(ssm4_upstream_sync)
#
# Connect SSM 1 to SSM 5
ssm1_downstream_sync4 = sync_module_1.connectivity.systemsync_downstream[3]
ssm5_upstream_sync    = sync_module_5.connectivity.systemsync_upstream[0]
ssm1_downstream_sync4.set_connection(ssm5_upstream_sync)
#
# Connect SSM 5 to SSM 6
ssm5_downstream_sync = sync_module_5.connectivity.systemsync_downstream[0]
ssm6_upstream_sync    = sync_module_6.connectivity.systemsync_upstream[0]
ssm5_downstream_sync.set_connection(ssm6_upstream_sync)
```

Chassis Supported for Multi-Chassis Systems

The following Keysight chassis models are supported:

- M9018B
- M9019A
- M9046A

Software and firmware version requirements are listed on-line here: Chassis Software and Firmware Requirements for KS2201A .

NOTE If you mix different chassis models in your multi-chassis setup, you may observe some skew across the different chassis and different performance depending on the different chassis characteristics.

Non Keysight chassis are not supported for multi-chassis systems.

Clocking

Clock Types

In a single or multi-chassis system there are 4 types of clocks used for synchronization and instrument-related tasks:

- Reference clock.
- System clocks.
- Analog clocks.
- Sample clocks.

All these clocks are synchronous with one another, but are used for different purposes and can be configured in different ways trading off performance and complexity/cost.

Reference Clock

The Reference clock determines the absolute frequency and lowest-frequency offset phase noise performance of the analog instrumentation's inputs and outputs. That is because all of the other clocks are phase-locked to the Reference Clock. A PXIe system can either use its own internal reference clock or phase-lock to an external reference clock. It can also provide external reference clock outputs for other instrumentation to phase-lock to.

System Clocks

The System clocks synchronize all the digital operation of all instruments and the PXIe platform. These clocks are derived from the Reference Clock and are used by, for example, the PathWave FPGAs Sandbox logic, the HVI Engine core clock, Fast Data Sharing and other digital capabilities in the instruments. Basically, a system clock is clock that is neither the reference clock nor an analog clock.

Analog Clocks

The Analog Clocks are intermediate frequency clocks from which the instrument's Sample Clocks are derived. Like the Sample clocks, the Analog Clocks affect the overall phase noise performance and skew drift of the instrument analog inputs and outputs. In the simplest clock configurations, each peripheral module generates it's own independent Analog Clock. In the highest fidelity clock configuration, a single common Analog clock is generated by the *High Performance Reference Clock Source* (HPRCS) and is distributed to all the individual peripheral modules though external cables and power dividers.

Sample Clocks

The instrument's ADCs and DACs that digitize analog input signals and generate analog output signals are clocked by their own internal Sample Clocks. The various types of peripheral modules use different sample clock frequencies even though they are ultimately derived from the same Reference clock. These sample clocks determine the overall phase noise performance and skew drift of the analog inputs and outputs because they directly clock the instrument's ADCs and DACs.

System Clock Distribution using SSM and System Sync connectivity

In a multi-chassis system based on the Keysight PXIe SSMs and chassis, the SSM with no other SSM connected to its System Sync Upstream port acts as the leader. This leader SSM forwards a copy of the system clock to other SSMs using System Sync cables. In turn, each SSM shares the forwarded system clock with the instruments located in their respective chassis using the PXIe DSTARA backplane signal.

NOTE You are not required to set the the leader in the HVI API. The leader SSM is determined by the hardware connections. That is, the leader role is automatically taken by the SSM that has no System Sync cable connected to its System Sync Upstream port.

Overview of Supported Clocking Schemes

There are several possible different clocking configurations, the one you should use depends on the hardware and the application requirements. Some of the key aspects to consider when selecting a clocking scheme are:

1. System and Analog clock sources. The source for the System and intermediate-frequency analog clocks is a critical element that determines the system synchronization, phase noise and drift performance. The clock sources covered in this section include:
   a. PXIe chassis.
   b. System Sync Module.
   c. PXIe Chassis with *High Performance Reference Clock Source* (HPRCS). This is only available on Keysight PXIe chassis models M904xA.

2. Internal/external Reference clock. The clock that serves as reference for the System/Analog clocks can be generated internally by the selected source, or externally provided by the user, generated by a clock source external to the PXIe system. In systems that include the *High Performance Reference Clock Source* (HPRCS), and other external instrumentation that you wish to share a common Reference Clock, the best overall jitter performance will usually be achieved by phase-locking the other external instrumentation to the HPRCS Reference Clock instead of the other way around. If the overall system needs to be phase-locked to a GPS or atomic standard reference, you should phase-lock the HPRCS to the GPS or atomic standard and phase-lock all the other instrumentation to the HPRCS Reference Clock.
3. Instruments internal/external Analog Clock. Most instruments can either use an external Analog Clock or generate their own Analog Clock internally for convenience, however, using a common external Analog Clock will always provide the best performance because all peripheral module sample clocks will jitter and drift together.

The following table shows the different supported/recommended clocking schemes:

Clocking Scheme	Reference Clock Source	Reference Clock Mode	Description	Performance
A.1 Single-chassis, no SSM.	Chassis	Internal 10MHz	An OCXO inside the chassis generates a 10 MHz reference clock. Independent Analog clocks are generated in each peripheral module.	See the chassis datasheet for exact phase noise performance. See the M5xxx PXIe instrument documentation for exact performance of channel to channel skew, jitter, and drift.
	External	External 10MHz	The external reference clock must have a frequency of 10 MHz. As an example, it can come from a *Device Under Test* (DUT), another instrument that is part of the setup, etc. Independent Analog clocks are generated in each peripheral module.	-
A.2 Single/multiple chassis with at least one M904x, Analog clocks, SSMs.	Chassis	Internal 10MHz	An OCXO inside the chassis generates a 10 MHz reference clock. A common Analog clock is externally distributed to each peripheral module.	See the chassis datasheet for exact phase noise performance. See the M5xxx PXIe instrument documentation for exact performance of channel to channel skew, jitter, and drift.
	External	External 10MHz	The external reference clock must have a frequency of 10 MHz. As an example, it can come from a DUT, another instrument that is part of the setup, etc. A common Analog clock is externally distributed to each peripheral module.	-

Clocking Scheme	Reference Clock Source	Reference Clock Mode	Description	Performance
B No external Analog clocks, SSMs.	SSM	Internal 10MHz	An OCXO inside the SSM generates a 10 MHz reference clock. Independent Analog clocks are generated in each peripheral module.	See the SSM datasheet for exact phase noise performance. See the M5xxx PXIe instrument documentation for exact performance of channel to channel skew, jitter, and drift.
	External	External 10/100MHz	The external reference clock can have a 10 or 100 MHz frequency. As an example, it can come from a DUT, from another instrument that is part of the setup, etc. Independent Analog clocks are generated in each peripheral module.	-
C external Analog clocks, SSMs, HPRCS.	HPRCS	Internal 10MHz	The HPRCS generates a 2.4 GHz sine wave that gets divided in frequency to generate a 100 MHz reference clock signal. A common Analog clock is externally distributed to each peripheral module.	This option provides the best performance in terms of phase noise. For more information, see the *Keysight PXIe Chassis M9046A Datasheet*, available at Keysight PXI chassis.
	External	External 10/100MHz	The external reference clock for the HPRCS can have a 10 or 100 MHz frequency. As an example, it can come from a DUT or another instrument that is part of the setup, etc. A common Analog clock is externally distributed to each peripheral module.	-

Depending on the chosen clocking scheme, additional connections to the SSM may be required:

- A.1 or B Internal: No extra connections are required.
- B External: Attach the external reference clock source to the SSM **SClk / Ref In** input.
- A.2 or C Internal: The chassis Ref1 output must be connected to the SSM **SClk / Ref In** input.
- A.2 or C External: The chassis Ref1 clock output must be connected to the SSM **SClk / Ref In** input. The external reference clock must be connected to the chassis external **Ref In** input.

The precise connections are described in the configuration sections later in this guide.

You may also phase-lock external instrumentation to the PXI system using either 10 MHz or 100 MHz external reference clock outputs.

Some instruments also require an analog clock. For more information see the later section titled *Configuring Analog Clock Source for Instruments*.

Configuring Clocking Scheme A.1

(Single-chassis, no SSM, no external analog clocks)

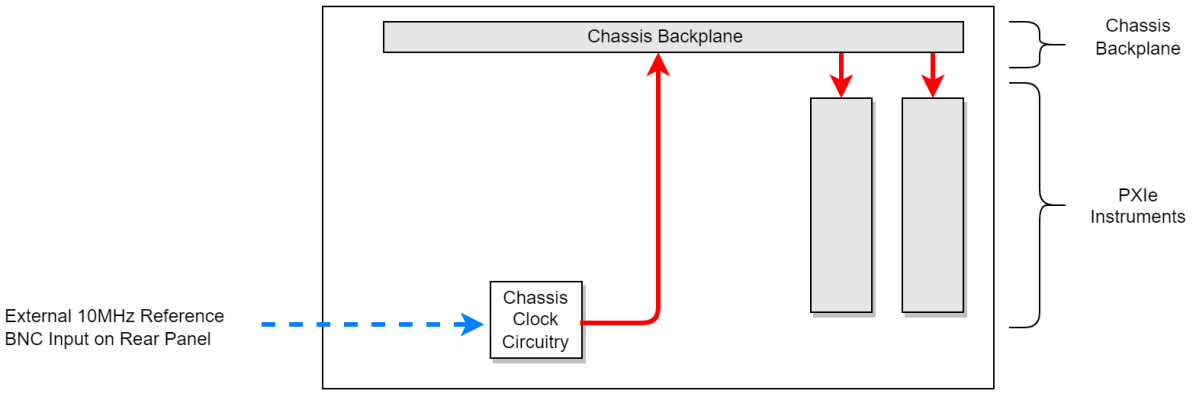
This is the simplest configuration and is the default if you have not specified another.

The chassis is the clock source. For the reference clock, there are two options:

1. **Internal (default):** This is the 10MHz clock built into the chassis (VCXO for the M9019A or OCXO for the M904xA).
2. **External:** A 10MHz signal connected to the 10MHz Ref BNC input located on the chassis rear panel.

The following diagram shows a chassis with an internal clock source (chassis clock) or an external clock source (blue):

Single chassis, no SSM
Internal or External Reference clock



All chassis have on their rear panel a 10MHz reference BNC input and a 10MHz reference BNC output. In the case of the M904x chassis, there are two Reference clock SMA outputs on the front panel.

This clocking scheme is rather constrained in terms of features because it only allows for a single chassis and, given that there is no SSM, advanced features like Fast Data Sharing are not available.

The following snippet shows how to configure the chassis as the clock source:

```
# Create system definition object
my_system = kthvi.SystemDefinition("MySystem")
#
# Define chassis
chassis = my_system.add_chassis(1)
#
# Select the chassis as ref. clock source
clockSource = chassis.clock_source
#
# Set the chassis as clock source
systemDefinition.clocking.reference_source = clockSource
#
# Explicitly set the clock source to use the internal OCXO as the reference clock (this is the
default)
clockSource.set_mode(keysight_hvi.ClockingReferenceMode.INTERNAL)
#
# Alternatively you can configure the chassis to use the external clock reference with the 10Mhz
frequency value in Hz
clockSource.set_mode(keysight_hvi.ClockingReferenceMode.EXTERNAL, 10e6)
```

Configuring Clocking Scheme B

(Single/multi-chassis system with SSM as clock source and no external analog clocks)

Each SSM is equipped with an onboard high-quality 10MHz Oven Controlled Crystal Oscillator (OCXO) that can be used as the reference clock.

Alternatively, the chassis backplane reference clock output (Scheme A.2) or the optional *High Performance Reference Clock Source* (HPRCS) output (Scheme C) can be used as reference clock. The HPRCS option requires a Keysight PXIe Chassis model M9046A. Clocking scheme B assumes the external reference clock is neither output by the chassis nor by the HPRCS because otherwise further configurations would be required for proper operation.

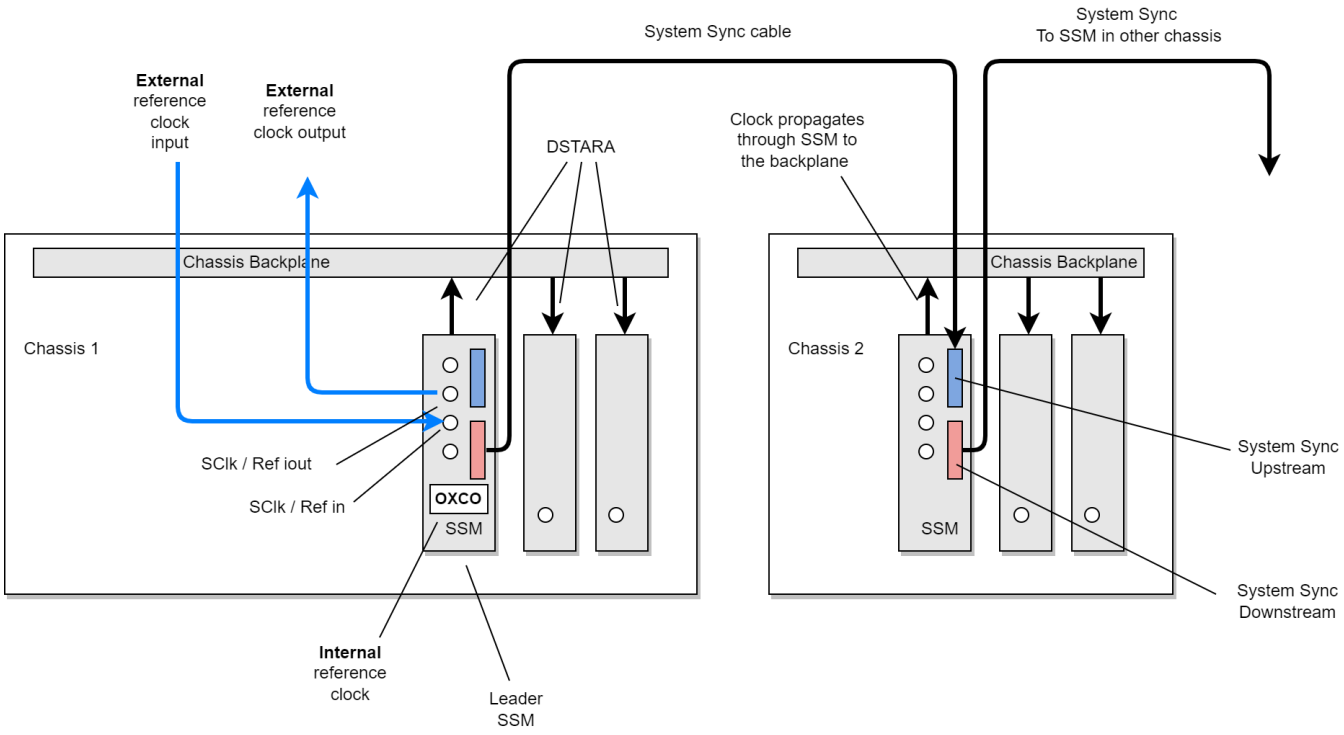
The reference clock can be chosen from two options:

1. **Internal:** This is the default mode. The internal OCXO of the leader SSM is used as the reference clock.
2. **External:** An 10 MHz or 100 MHz external reference clock is connected to the SSM's front-panel **SClk/Ref In** SMP input.

The reference clock gets propagated to all the PXIe instruments within the same chassis through the DSTARA signal path. It gets propagated to the next SSM through the System Sync cable from the downstream connection on leader SSM to the upstream connection on the follower.

The following diagram shows the operation of the Clocking Scheme B. The clock is generated in the SSM in chassis 1 and is passed to the other instruments in the chassis via the DSTARA signal path in the backplane (red arrows). It is also passed to the next chassis via the System Sync cable (in black) where it propagates via the SSM in that chassis. The internal reference is the SSM's OCXO, and the external reference is shown in blue:

Multiple chassis with SSMs
SSM as clock source with internal or external reference



Configuring the SSM as the System Clock source

By default, if you do not specify anything, PathWave Test Sync Executive configures the leader SSM as the reference clock source using its internal OCXO clock. The leader SSM is defined by the hardware connections. In the HVI API, no additional definition other than the connections between SSM is required to identify the leader SSM. You must ensure the connections you define in software match the physical hardware connections between SSMs.

The following code shows how to configure a pair of chassis with SSMs where the OCXO clock is the reference clock source, `options` is set to an empty string:

```
# Create system definition object
my_system = kthvi.SystemDefinition("MySystem")
#
# Define all necessary follower SSMs depending the number of chassis
leader_ssm = my_system.interconnects.add_sync_module(SSM_1, options)
my_system.interconnects.add_sync_module(SSM_2, options)
#
# Define chassis
my_system.add_chassis(1)
my_system.add_chassis(2)
#
#  Select the leader SSM as ref. clock source
clockSource = interconnects[0].clock_source
#
# Set the SSM clock source
systemDefinition.clocking.reference_source = clockSource
#
# Explicitly set the clock source to use the internal OCXO as the reference clock (this is the
default)
clockSource.set_mode(keysight_hvi.ClockingReferenceMode.INTERNAL)
```

Configuring the SSM to explicitly use internal OCXO or external reference clock

The SSM leading the synchronization by default with its internal reference clock, can optionally be connected to an external reference clock. The external reference can come from, for example, a DUT or another source such as a PXIe frequency reference.

To use an external reference clock, you must:

- Connect the external reference source to the SSM's SClk / Ref in port.
- In the HVI API you must set the SSM to synchronize to an external reference clock. To do this, set the mode to `EXTERNAL` and set the frequency in Hz.

To use the external reference, change the final line in the previous code snippet to:

```
# Set clock mode to EXTERNAL and set frequency to 10MHz
clockSource.set_mode(keysight_hvi.ClockingReferenceMode.EXTERNAL, 10e6)
```

Configuring Clocking Schemes A.2 and C

(Single/multi-chassis with Keysight M904xA chassis, external Analog clocks and chassis or HPRCS as clock source)

Some instruments such as the analog ones in the Keysight M5xxx PXIe family derive their Sample Clocks from an Analog Clock source and perform best when configured to use an externally distributed Analog Clock. This section explains how to distribute the analog clocks to these and similar instruments.

The analog clock can be generated from either the M9546A HPRCS or from the M9046A chassis backplane board. The preferred choice for the analog clock source is the M9546A HPRCS inside a Keysight M9046A PXIe chassis because of it's superior phase noise. The HPRCS can generate a sine wave with frequencies of 2.4, 4.8, 9.6, or 19.2 GHz. In this section we assume the analog clock source being set to generate 2.4 GHz, because this is the frequency required by the Keysight M5xxx PXIe family. The frequency can be configured at purchase by choosing the corresponding option for the Keysight M9046A PXIe chassis. For more information, see the *Keysight PXIe Chassis M9046A User Manual* available at Keysight PXI chassis.

Analog clock configuration options

the following table lists the options available:

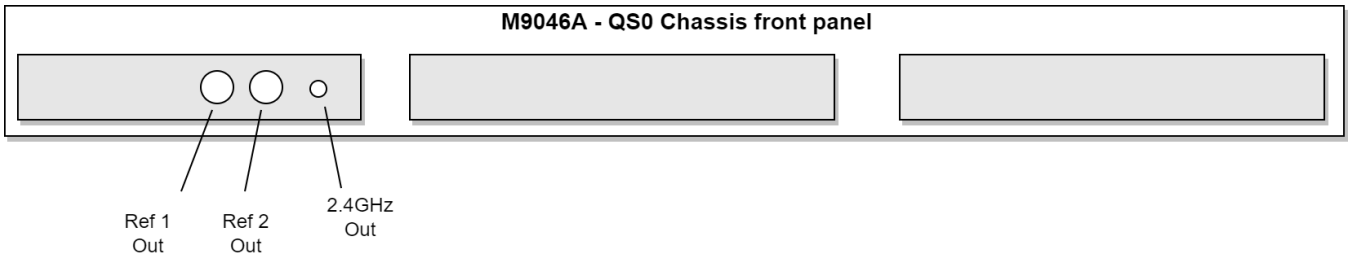
Source	Analog clock, locked to the reference clock	Performance
M9046A chassis with M9546A HPRCS	2.4, 4.8, 9.6, or 19.2 GHz	Best
M9046A chassis without M9546A HPRCS	2.4 GHz	Medium

M9046A Front Panel Clocking IO overview

The following diagrams show the M9046A chassis front panels and how they are connected in different configurations. The type and number of front panel connectors depend on the purchased hardware option for splitters and HPRCS: (-QS0, -QS1/3, -QS2). In the diagrams a frequency of 2.4 GHz is assumed to have been selected for the analog clock. The analog clock frequency is also chosen as hardware option at purchase time. More info in the *Keysight PXIe Chassis M9046A User Manual*, available at Keysight PXI chassis.

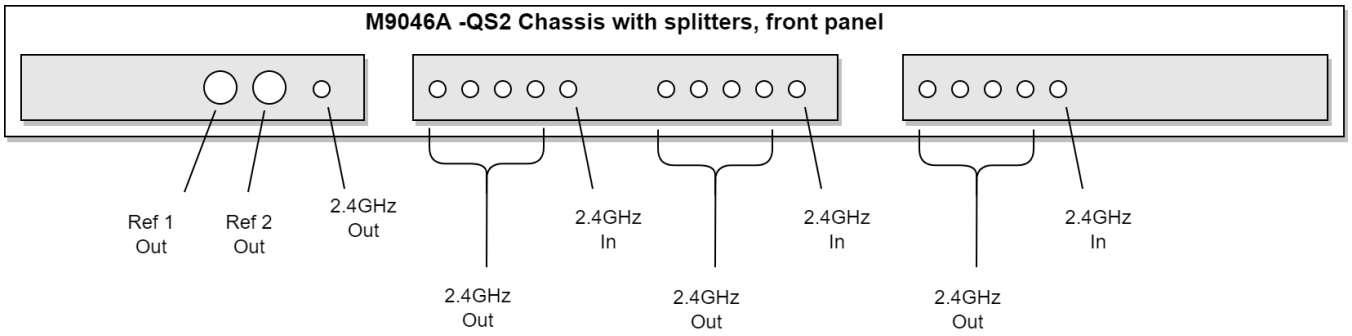
M9046A -QS0 Chassis (no HPRCS)

The following diagram shows the front panel of an M9046A -QS0 chassis.



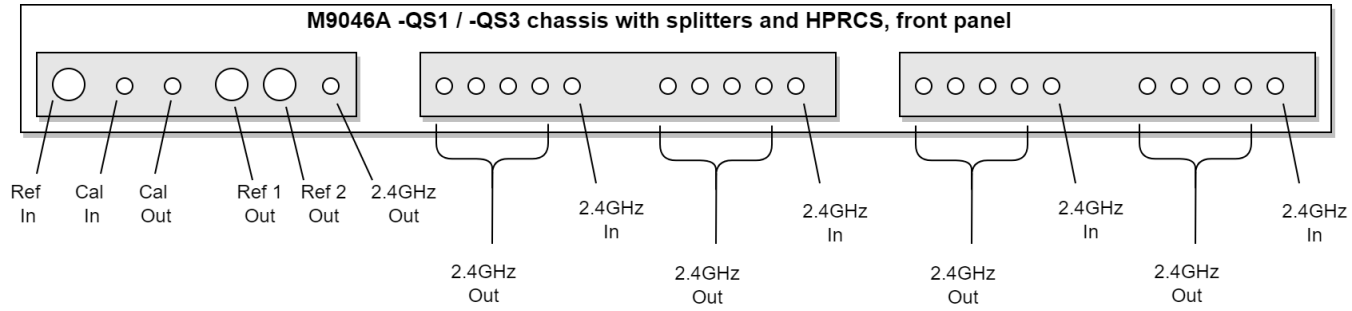
M9046A -QS2 Chassis with Analog clock splitters

The following diagram shows the front panel of a M9046A chassis with -QS2 option including the front panel analog clock splitters to ease the distribution of the analog clocks to all modules.



M9046A -QS1/3 Chassis with Analog clock splitters and HPRCS

The following diagram shows the front panel of an M9046A -QS1/3 chassis with analog clock splitters and Ref In, Cal In and Cal Out for the M9546A HPRCS.



Using the M9046A analog clock source

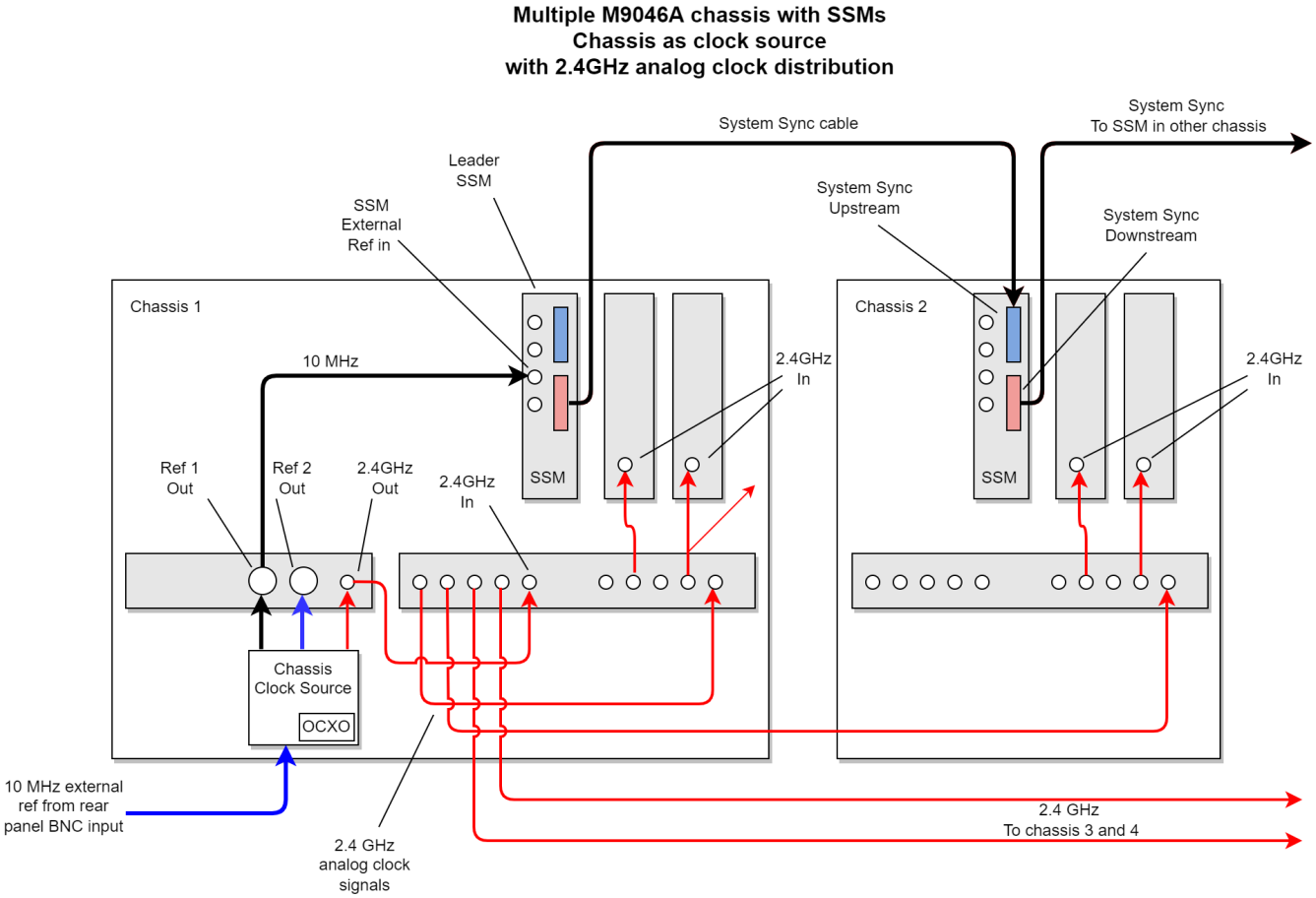
(Clocking Scheme A.2)

This configuration is for single or multiple chassis with SSMs. This configuration is compatible with PXIe Chassis model M9046A with hardware options -QS0, -QS1/3 or -QS2. The internal chassis clock is used as the clock source and this configuration must be defined in the HVI API. The chassis clock must be taken out from the **Ref 1 Out** port on the PXIe M9046 Chassis front panel and must be connected to the **SClk / Ref** In port of the PXIe SSM (see diagram below).

You can use the chassis as the reference clock source with its reference clock set to:

1. **Internal**: Use the chassis internal OCXO.
2. **External**: Using an external reference clock connected to the chassis rear panel 10MHz Ref BNC input.

The following diagram depicts the SSM using the chassis clock (indicated in red) as the clock source. The chassis external reference is indicated by the dotted blue arrow:



Configuring the M9046A as the system and analog clock source

To use the internal chassis clock, you must:

- Connect the Chassis Ref 1 Out output to the SSM's SClk / Ref In located in the same chassis.
- In the HVI API you must instruct the SSM to use the chassis clock.

By default, and if it is not specified otherwise, the chassis clock circuitry uses its internal OCXO as the reference clock.

The following code shows how to configure a pair of chassis with SSMs using the chassis clock as the reference clock, options is set to an empty string:

```
# Create system definition object
ktHvi.SystemDefinition definition("Name")
#
# You must add all necessary follower SSMs depending on the number of chassis
syncModuleLeader = definition.interconnects.add_sync_module(SSM_1, options)
syncModuleFollower = definition.interconnects.add_sync_module(SSM_2, options)
#
# Add chassis
chassis1 = definition.add_chassis(1)
definition.add_chassis(2)
#
# Get the chassis clock
clock_source = chassis1.clock_source
#
# Set as reference
definition.clocking.reference_source = clockSource
#
# Enable the chassis analog clock
clock_output_2_4GHz = ref_chassis.clock_outputs["FP2.4GHzOut"]
clock_output_2_4GHz.set_enabled(True)
```

Configuring the M9046A to use the external reference clock

To use the chassis clock with an external reference clock, you must:

- Connect the external reference clock to the Chassis rear panel's 10 MHz Ref BNC input.
- Connect the Chassis Ref 1 Out to the SSM SClk / Ref In of the SSM in the same chassis.
- In the HVI API you must instruct the chassis to use the external reference clock and set the frequency in Hz.

The following code shows how to set the external reference:

```
# Set the reference mode to use an external reference
# Set clock mode to EXTERNAL and set frequency to 10MHz
clockSource.set_mode(keysight_hvi.ClockingReferenceMode.EXTERNAL, 10e6)
```

Using the M9046A's M9546A High Performance Reference Clock Source

(Clocking Scheme C)

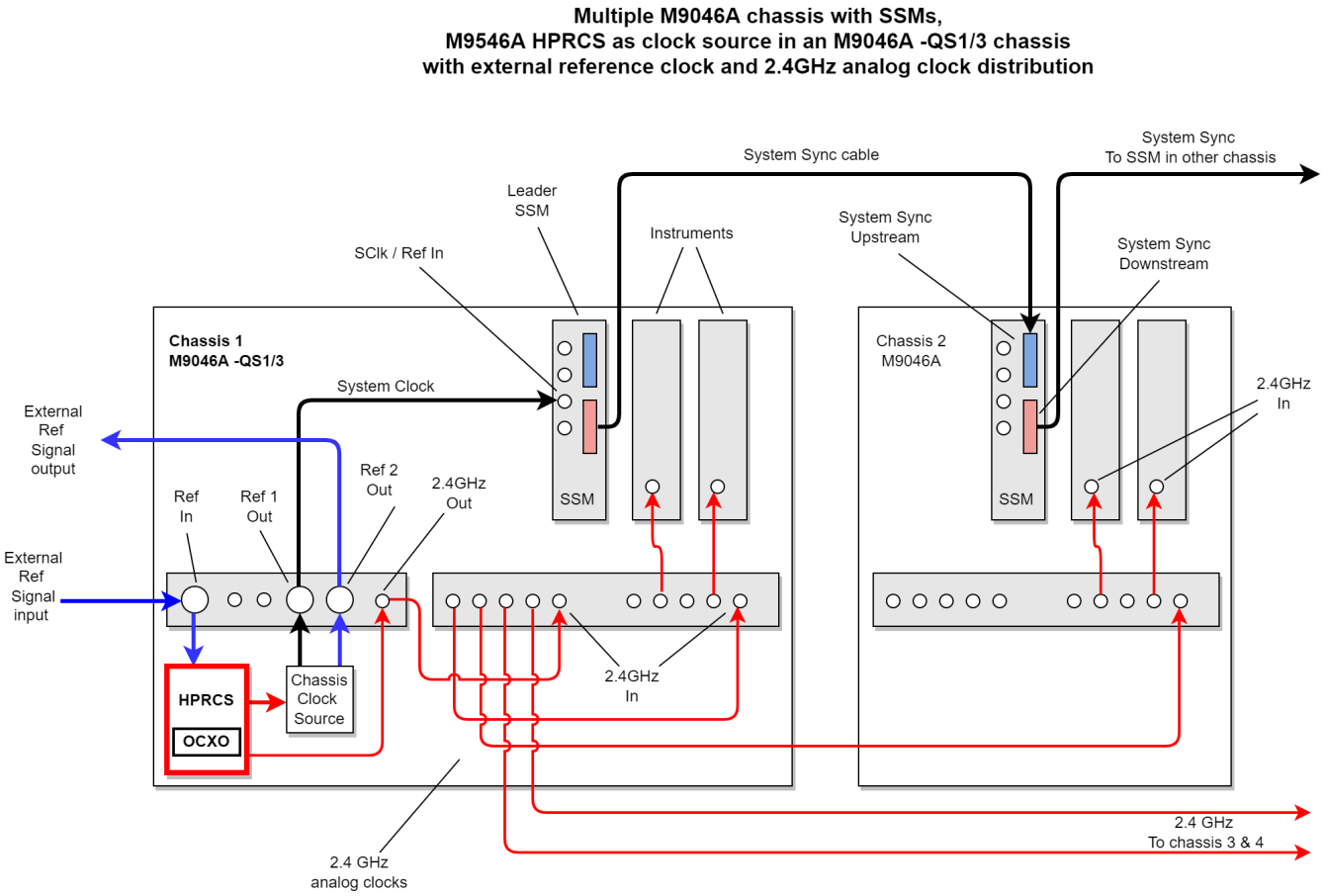
This configuration is for single or multiple chassis with SSMs. For this clocking scheme to be used, the first SSM must be in a Keysight M9046A chassis containing an M9546A *High Performance Reference Clock Source* (HPRCS).

The HPRCS is used as the clock source and this configuration must be specified in the HVI API.

We can use the HPRCS as a clock source with its reference clock set to:

- Internal: Use the HPRCS internal OCXO.
- External: Use an external reference clock connected to the chassis front panel Ref In input.

The following diagram shows the leader SSM using the M9546A HPRCS (indicated in red) as the clock source in chassis 1 M9046A -QS1/3. The HPRCS external reference is indicated by the blue arrow. The distribution of the 2.4 GHz analog clock to up to 4 chassis is also shown. The connection topology and cables used are critical to achieving the optimal channel skew drift performance. For information about how to connect the analog clock to more than 4 chassis, see *Keysight PXIe Chassis M9046A User Manual* available at Keysight PXI chassis.



Configuring the M9046A + HPRCS as the system and analog clock source

To use the HPRCS as clock source, you must:

- Connect the chassis **Ref 1 Out** output to the **SClk / Ref In** of the SSM located in this chassis
- In the HVI API you must:
  - Add the M9046A -QS1/3 chassis with HPRCS to the system definition.
  - Set the HPRCS to be the clock source.

When no external reference clock for the HPRCS is specified, its internal OCXO is used.

The following code shows how to configure a pair of chassis with SSMs using the HPRCS as clock source, options is set to an empty string:

```
# Create system definition object
my_system = kthvi.SystemDefinition("MySystem")
#
# Define all necessary SSMs depending on the number of chassis
```

```
my_system.interconnects.add_sync_module(SSM_1, options)
my_system.interconnects.add_sync_module(SSM_2, options)
#
# Define chassis
hprcs_chassis = my_system.add_chassis(1)
my_system.add_chassis(2)
#
# Create HPRCS object
clockSource = hprcs_chassis.high_performance_clock_source
#
# Set the HPRCS as the reference clock
my_system.clocking.reference_source = clockSource
#
# Enable the chassis analog clock
clock_output_2_4GHz = ref_chassis.clock_outputs["FP2.4GHzOut"]
clock_output_2_4GHz.set_enabled(True)
```

Configuring the M9046A + HPRCS to use an external reference clock

To use the HPRCS with an external reference clock, you must:

- Connect the external reference clock to the chassis' front panel Ref In input.
- Connect the chassis' front panel Ref 1 Out output to the SClk / Ref In input of the SSM located in this chassis.
- In the HVI API you must:
  - Add the M9046A -QS1/3 chassis to the system definition.
  - Set the HPRCS to be the clock source.
  - Instruct the HPRCS to use an external reference clock and the desired frequency in Hz.

To use the external reference, set the reference clock mode in the previous code snippet to (defaults to internal):

```
# Set the reference clock mode. Set the HPRCS to use an external reference @10Mhz
clockSource.set_mode(keysight_hvi.ClockingReferenceMode.EXTERNAL, 10e6 )
```

Enabling chassis clock outputs

If you are using a clock output from a chassis you can enable it in the HVI API.

The chassis clock outputs are available in the chassis and you can access them by their name as follows:

```
# Get the Clock configuration for the Rear Panel 10MHz output port from the Chassis
ktHvi.SystemDefinition definition("Name")
#
chassis = definition.add_chassis(1)
#
```

```
clockOutputRp10Mhz = chassis.clock_outputs["RP10MHzOut"]
clockOutputRP10Mhz.set_enabled(true/false)
```

Some clock outputs support one single frequency and others support multiple frequencies. For the outputs supporting only one frequency, no frequency must be provided when enabling/disabling them. If the clock outputs do support multiple frequencies, you must specify what frequency (in Hz) you want to enable.

When you disable the clock, the frequency argument is ignored.

The following code shows some examples and error cases:

```
clockOutputRp10Mhz = chassis.clock_outputs["RP10MHzOut"]
clockOutputRP10Mhz.set_enabled(true) # Ok
#
clockOutputFpRef2Out = chassis.clock_outputs["FPRef2Out"]
clockOutputFpRef2Out.set_enabled(true, 10e6) # Ok
```

Enabling the chassis Analog clock

If you are using an analog clock output from a chassis you must enable it.

The following code shows how to enable a 2.4GHz analog clock output from an M9046A chassis.

```
clock_output_2_4GHz = ref_chassis.clock_outputs["FP2.4GHzOut"]
clock_output_2_4GHz.set_enabled(True)
```

Configuring the Analog Clock Source in Instruments

For instruments that require an analog clock, you must set the source and frequency of the analog clock in your system definition.

You can set parameters for the analog clock:

- The source as internal or external.
- The frequencies of the sources, in Hz.

For external sources, the source selected depends on the analog clock frequencies that the instrument supports.

- If you indicate multiple frequencies, the first external frequency supported by the instrument is selected.
- If none of the external frequencies are supported, and the instrument has an internal clock, the internal clock is selected.
- If none of the external frequencies are supported, and the instrument does not have an internal clock, an error is generated.

The code is:

```
my_system.clocking.enable_external_analog_clocks(frequencies)
```

For example, if you are using a M9046A chassis with a M9546A HPRCS with the analog clock set to 2.4GHz (Clocking Scheme C), add the following line:

```
my_system.clocking.enable_external_analog_clocks([2400e6])
```

Instruments that support an external analog clock are set to use this clock. Instruments that do not support this external frequency are set to use an internal clock. If the instrument does not support the frequency and does not have an internal clock, an error is generated.

Selecting the best analog clock source for instruments

While it is often convenient for instruments to use their own internally generated analog clocks, the best jitter and drift performance is achieved by using a single common analog clock source generated within the Leader chassis (with or without the HPRCS) and distributing it using the chassis amplified power splitters in a balanced star configuration. This ensures that any low-frequency jitter skew drift is common across the system, minimizing the inter-channel jitter and drift.

In some cases with high channel count configurations, there may not be enough individual copies of the the Analog Clock available from a full balanced star distribution to connect to every instrument. In those cases, a single daisy-chain connection of the Analog Clock between instrument pairs can be used. Noting that the downstream instrument of the daisy-chained pair will have slightly higher skew drift than the non-daisy-chained instrument. Daisy-chained instruments shall have slightly higher skew drift, so these instruments should be the ones in the system which have the lowest bandwidth. For example, in systems which employ the M5201A Downconverter and the M5200 Digitizer, which are typically used in pairs, it is best practice to route the Analog Clock to the downconverter first and then daisy-chain the downconverter's Analog Clock output to the digitizer's Analog Clock input. This is because the 2 GHz digitizer is less sensitive to the same amount of channel skew than the 16 GHz downconverter.

The Keysight MCX cables are made of a special material that minimizes their propagation delay change with temperature. Matching the total propagation delay from their common clock source to each instrument causes the propagation delay drifts of the clocks to cancel out between instruments.

The balanced-star configuration of external Analog clocks uses custom 4:1 amplified power dividers built into some chassis. These power dividers are designed specifically for minimizing phase noise, temperature drift, and to maintain the Analog clocks amplitude as it is divided many times. Substituting other power dividers to distribute the Analog clocks will degrade jitter and drift performance, so this is not recommended.

Small spurious oscillations can occur within the amplified power divider when any of the outputs are loaded with certain reflective loads. For this reason, terminating unused outputs with 50 ohm loads is recommended. It is only necessary to terminate unused outputs of power dividers that are currently being used to distribute the Analog clocks.

System Initialization

The HVI API enables you to control the process of system initialization and clock alignment.

The following describes the steps you take to initialize the system for a number of different scenarios. For more information about initialization options available in the HVI API, see the section *System Initialization* in the *PathWave Test Sync Executive User Manual* available here.

The system must be initialized after every power cycle, a change in the system hardware configuration, or a change in the clock configuration. The initialization procedure you need to perform depends on:

- The instruments in the system.
- Your required channel skew accuracy.
- If the hardware or clock configuration has changed.
- If a system initialization has been performed.

Example of System Initialization and Operation

To use the HVI API to initialize and run real-time operation in your system, there are two main procedures that you must follow:

1. System Warm-up and Calibration
2. Normal Operation

There are also a number of use cases that are variations on these main procedures. The following text describes these procedures along with the use case variations.

NOTE The initialization process requires access and control of all of the hardware resources, so it is important that these resources are not already in use by another application or HVI instance already loaded to hardware. An exception is thrown if any of the hardware resources are already in use.

System Warm-up and Calibration

The system warm-up must be performed every time the system is turned on or the hardware configuration is changed. This is to enable all of the components to reach a stable and repeatable operating temperature. Once the system is warmed-up, the system can be initialized using the stored System Calibration data.

The System Calibration must performed in these cases:

1. The very first time that the system is put together and powered-on.
2. When relevant hardware changes are made that require a new system calibration. These hardware changes include:

a. Adding/removing a chassis in your SystemDefinition object.
b. Adding/removing any instrument that requires clock alignment calibration data, such as an M5300 or M5201, or changes the operating temperature of the system.
c. Changing the cable connections between System Synchronization Modules, even replacing a cable with a similar one with a different serial number.
d. Changing any of the external System Clock or Analog Clock cable connections, even replacing a cable with a similar one with a different serial number.
e. Making any change to the clock configuration, even if it is only from the HVI API. This is because this triggers the usage of different clock sources or signal paths.

3. Other situations where the system calibration should be updated.
4. On rare occasions, a component in the system can move into an invalid state and a reset of the calibration might be required. For more information, see *System Troubleshooting* in the ***System Setup Guide*** .

NOTE Warning : Resetting the system calibration shall in turn require you to recalculate the User Calibration for some instruments. Observe extreme caution when doing this to avoid costly time-consuming recalibration.

Procedure steps:

1. Power-on the system
   a. Power-on all of the chassis. After this is complete, if you are using an external chassis controller, power it on.

2. Connect to all the instruments
   a. For example: `instrument = ktm5300.KtM5300x(resource_id, query, reset, options)`

3. Create a System Definition using the HVI API and the instrument drivers:
   a. Create a SystemDefinition object that we refer to here as *my_system*. Use the *my_system* object to define all the hardware resources in your system: chassis, SSMs, instruments, clocking configuration, reference clock source, etc.
      For example: `my_system.chassis.add(1)`, `my_system.clocking.reference_source = chassis.clock_source`
   b. Add the HVI Engines of each instrument to the SystemDefinition object.
      For example: `my_system.engines.add(instrument.hvi.engines.main_engine, "MyEngine")`

4. System Initialization for Warm-Up
   a. Execute `my_system .initialize(keysight_hvi.AlignmentModes.FULL | keysight_hvi.AlignmentModes.PRE_CALIBRATION)`. The `PRE_CALIBRATION` flag indicates there is no need to apply any previously stored system calibration values because the system is warming-up. This enables the system to execute code without calibration related errors. After this step, instruments may present channel skew errors which are compensated by the next steps.

5. Wait for System Warm-Up
   a. Wait for the required warm-up time, this can range from a few minutes to about 30 minutes. The actual time typically depends on the type and number of instruments in the system, clocking configuration, etc.
   b. For detailed warm-up time information, see your instrument documentation, for example: *M5300 RF AWG User's Manual* .

6. System initialization to perform System Calibration
   a. Using the SystemDefintion created in step 3, run `my_system.initialize(keysight_hvi.AlignmentModes.FULL | keysight_hvi.AlignmentModes.RESET_CALIBRATION)` to generate internal system calibration data. At first system turn-on, no previous calibration data is expected to be available.

7. Calculate User Calibration or channel deskew (Optional)
   a. This operation is optional and consists of correcting analog channel skews introduced by cable and signal path delays. Note that in some instruments, the User Calibration must be re-calculated when a System Calibration is executed. For information about how to do this, see your instrument documentation.

8. Ready for Normal Operation

Use Cases

Use Case Scenario	Description
First system start-up and calibration	The very first time that the system is put together and powered-on, you must execute a full warm-up and calibration procedure to achieve the best system performance and repeatability: • Execute all steps #1 to #7 above.
System start-up using existing calibration	If the system has already been calibrated for the current hardware configuration, then, to reuse the existing calibration to configure the system, wait for the system temperature to stabilize then apply the existing calibration: • Execute steps #1 to #5 above. • Skip steps #6 and #7 *System initialization to perform System Calibration* and *Calculate user calibration or channel deskew* , and run `my_system.initialize (keysight_hvi.AlignmentModes.FULL)`.
Simplified uncalibrated system start-up	If you want to use the system for test development, or you can tolerate analog channel drift of up to 50ps across reboots/power-cycles: • Execute steps #1 to #4 above. • Skip steps #5 to #7 *Wait for System Warm-Up* , *System initialization to perform System Calibration* and *Calculate user calibration or channel deskew* .

NOTE **System hot boot-up:** If the system is already warmed-up to the calibration operating conditions, for example after a system restart, you can skip the steps #4 and #5 *System Initialization for Warm Up* and *Wait for System Warm-Up* .

Normal Operation

Once the system is warmed-up and the system calibration has been done, users can use the the HVI API to execute real-time operations:

NOTE Note that if it is the first system start-up or you have introduced any of the HW changes that require new System/User Calibration you must execute the *First system start-up and calibration* use case described in the *System Warm-up and Calibration* procedure.

Procedure steps:

1. Connect to all the instruments, if not already connected.
   a. For example: `instrument = ktm5300.KtM5300x(resource_id, query, reset, options)`

2. Apply user calibration to instruments, You only need to do this if it is required, the user calibration data is available, and it has not been applied already.
   a. The user calibration is calculated during the *System Warm-up and Calibration* process. For information about how to apply existing calibration, see your instrument documentation, for example: *M5300 RF AWG User's Manual* .

3. Create a SystemDefinition object, or reuse an existing one.
4. Initialize the SystemDefinition object (Optional)
   a. Run `my_system.initialize()`. This call executes the minimal or default initialization, provided a Full Initialization has been executed already as described in the *System Warm-up and Calibration* procedure. If the full initialization has not been executed, this step requires calibration data. If the calibration data is not available this operation will fail. To run the system initialization without calibration you can specify the `PRE_CALIBRATION` flag: `my_system.initialize(keysight_hvi.AlignmentModes.PRE_CALIBRATION)`
   b. Note that you can skip the call to `my_system.initialize()` because the minimal or default initialization happens implicitly in steps #5 and #7 described below.

5. Create a Sequencer object
   a. For example: `sequencer = keysight_hviy.Sequencer("MySequencer", my_system)`
   b. Note that the sequencer creation operation implicitly executes a default initialization, this is equivalent to calling `SystemDefinition:Initialize()`.

6. Create an HVI object
   a. For example: `hvi = sequencer.compile()`
   b. The HVI object is created by compiling the Sequencer object after all the HVI sequences have been programmed.

7. Load HVI to HW
   a. For example: `hvi.load_to_hw()`
   b. Note that the load_to_hw() operation implicitly executes a default initialization, this is equivalent to calling `SystemDefinition:Initialize()`.

8. Run HVI
   a. For example: `hvi.run(hvi.no_timeout)`

9. Release HW
   a. For example: `hvi.release_hw()`

NOTE **Forcing a full initialization.** You can optionally force a full initialization. Forcing the full initialization can be useful to unblock a system if it is in a bad state, when some temporary hardware changes in the system are done such as reconnecting cabling using the same cables, or in general when it is useful to ensure the system is fully initialized to discard any previous state. To force the full initialization run:

1. `my_system.initialize(keysight_hvi.AlignmentModes.FULL)`.
2. Or if you are using the system without calibration, add the `PRE_CALIBRATION` flag: `my_system.initialize(keysight_hvi.AlignmentModes.FULL | keysight_hvi.AlignmentModes.PRE_CALIBRATION)`

NOTE **User Calibration not required or already applied:** If user calibration is not required or has already been applied to the instruments, you can skip step #2 *Apply user calibration to instruments.* For more information on how to handle User Calibration in instruments, see your instrument documentation.

System Troubleshooting

When you are using more than 1 chassis, you must:

- Use the latest chassis driver and firmware.
- Specify the connections between the chassis in the HVI API.

Chassis numbering

- Ensure your chassis are numbered from 1 upwards.

  The PXI standard does not permit chassis to be numbered as 0. If this happens, it indicates there has been an incorrect installation of the firmware, PXI chassis driver, software, or PXI resource manager.

Ensure you are using correct firmware and software components

- For PathWave Test Sync Executive to work correctly, the PXI chassis, firmware, driver, software, and PXI Resource Manager must be all be installed correctly, regardless of the chassis vendor.

  Compatibility requirements for PathWave test Sync Executive are listed at Instrument Software and Firmware Requirements for KS2201A.

Using non-Keysight chassis with PathWave Test Sync Executive

- Keysight recommends you use PathWave Test Sync Executive with Keysight chassis. It is It is possible to use non-Keysight chassis with the following limitations:

  Only a single PXIe chassis is supported if you are using a non-Keysight chassis. Multi-chassis operation requires the recommended Keysight PXIe Chassis.

  The proper PXIe resource manager and chassis VISA driver installation is required.

  PathWave Test Sync Executive has not been validated with non-Keysight PXIe chassis, if you encounter any issues, contact support.

Using non-Keysight chassis with Keysight Instruments and PathWave Test Sync Executive

- Check the documentation of each PXI instrument that you are using with PathWave Test Sync Executive, to ensure they comply with the instrument limitations on compatibility with non-Keysight chassis or controllers.

Error messages and troubleshooting guide

If an error occurs while you are running HVI, an error message is typically displayed on the console. This message can originate in either:

- HVI itself.
- One of the components that HVI controls.

In the second case, HVI outputs a message identifying the instrument and process that generated the error, prefixing the error itself by the string **`Product error`** for easy identification.

NOTE If an error message includes **`Product Error`**, to ensure the best and fastest service, report the problem directly to the support representatives for the relevant product, for example Chassis, HPRCS, SSM or Instruments.

Common courses of action to resolve errors are described in the following tables. If the error is not listed or the proposed action does not solve the problem, follow these steps:

1. Rerun the system initialization forcing `FULL` initialization.
2. If the error persists, power cycle the complete hardware setup and run the system initialization forcing `FULL` initialization.
3. If the error still persists, and as a very last resort, consider running a `FULL` initialization with `RESET_CALIBRATION` => `my_system.initialize(keysight_hvi.AlignmentModes.FULL | keysight_hvi.AlignmentModes.RESET_CALIBRATION)`
   WARNING: Resetting the system calibration in this way will in turn also force you to rerun the *System Warm-up and Calibration* procedure described above, which for some instruments requires to redo the User calibration. So observe extreme caution when doing this to avoid costly and time-consuming recalibration.

System Setup Errors

Error message	Explanation	Common causes and possible fixes
`Chassis with number n not found`	Adding chassis `n` to the `SystemDefinition` using `add_chassis` fails.	PXIe cable to chassis `n` is not plugged in. Chassis `n` is not powered up. Chassis `n` was not discovered during PC host boot-up.
`IVI ERROR: library:KtM9032x_64.dll, error_code:x - Invalid session ID (VI_NULL).` `No error message could be retrieved. Please check the documentation`	Adding an SSM to the `SystemDefinition` using `add_sync_module` fails.	There is no SSM in the SSM slot of at least one chassis The SFP of at least one SSM is open when it shouldn't be.
`ERROR: Only leader Sync Module can be configured as clock source; leader cellini is connected to chassis n`	Setting the clock source to be an SSM other than the SSM in leader chassis `n.`	Set the leader SSM (with no SSM connected to its upstream System Sync port) to be the clock source.
`ERROR: Chassis Clock Source can only be used in the chassis where the leader Sync Module is connected; clock source chassis is n and leader Sync Module chassis is n`	Setting the clock source to be a chassis other than the leader chassis `n.`	Only the leader chassis hosting the leader SSM can be set to be a clock source
`Chassis only supports 10MHz external reference. Please set an external reference frequency of 10MHz when using a chassis clock source as a reference`	The frequency of the external reference for the chassis is incorrectly set using `set_mode()` when using the chassis as a clock source.	Set the frequency of the external reference clock to 10 MHz.
`Chassis n doesn't support High Performance Clock Source`	Leader chassis `n` is not an M9046A with HPRCS. This is required to set the clock source to HPRCS.	Replace leader chassis `n` with a chassis containing an HPRCS.
`Kt9546x: wait reference clock set mode fails.: Please check the physical connection between Chronos and its clock source`	Incorrect external reference clock for the HPRCS in the leader chassis.	Reference clock set to the incorrect frequency. M9546 SMA Ref In input not connected. Double check the external reference clock source going into the HPRCS.
`Hardware Error: Kt9546 Pll is not locked`	M9546 fails to lock to the external reference clock when using HPRCS as clock source.	

Initialization errors

Error message	Explanation	Common causes and possible fixes
`operation "Link Initialization" failed: FdsConnectorAdapter::initializeConnectivity: Module info: Chassis n, Slot 10, HVI version x, Firmware version y. Product error: The SystemSyncUp_x8 port is not hooked up. FdsConnectorAdapter::initializeConnectivity: Module info: Chassis m, Slot 10, HVI version x, Firmware version y. Product error: The SystemSyncDown_x8 port(s) is not hooked up`	FDS System Sync link initialization failed.	System Sync cable connecting the SSMs in chassis `m` and `n` is missing (downstream `s` of SSM `m` to upstream of SSM `n`).
`operation "LinkAlignment" failed: Could not align link CHASSIS#n:SLOT#10::PxiBackplane#0:Port#s:Tx -> CHASSIS#n:SLOT#m::PxiBackplane#0:Port#0:Rx. FdsConnectorAdapter::finishAlignment: Module info: Chassis n, Slot m, HVI version x, Firmware version y. Product error: Required Pipe Select for DstarB of -5 exceeds expected max value of 7. Latency of bank 10. Latency count 15 Could not align link CHASSIS#n:SLOT#m::PxiBackplane#0:Port#0:Tx -> CHASSIS#n:SLOT#10::PxiBackplane#0:Port#s:Rx. FdsConnectorAdapter::finishAlignment: Module info: Chassis n, Slot 10, HVI version x, Firmware version y. Product error: FDS measured latency count larger than expected fixed latency count`	FDS DSTARB/C link initialization between SSM `n` and the instrument in chassis `n` slot `m` failed.	Power cycling the hardware setup should fix this. If the error persists then it might be a hardware error that you should report to Keysight.
`Hardware Error: Chassis n is not locked to SyncModule clock`	–	Power cycling the hardware setup should fix this. If the error persists then it might be a hardware error that you should report to Keysight.

Initialization errors (continued)

The following SSM messages are of similar nature and are all preceded by the chassis number the SSM is in, the failing function call and the HVI and FW versions:

Error message	Explanation	Common causes and possible fixes
`Product error: PLL unlocked after MMCM input phase adjustment`	The leader SSM in chassis n fails to lock to the Ref1Out reference from the chassis front panel. The leader SSM in chassis n fails to lock to the external reference clock.	Cable is not connected, or it is connected to the incorrect SMA output. The external reference clock is missing.
`Product error: Unable to lock to REF_IN. Verify clock configuration and cabling and re-run the test.  Refer to the KS2201A PathWave Test Sync Executive System Setup Guide for further guidance`	Rear 10 MHz chassis n REF IN BNC input is not present. Leader SSM n Ref In input is not connected.	The external reference clock has the incorrect frequency. This is usually because of missing external clocks, cabling issues, or an incorrect frequency setting. Check all of these.
`Product error: Unable to lock to System Sync FWD_CLK. (Same as above ...)`	-	Follower SSM n upstream System Sync connection fails.
`Product error: Unable to lock the internal OCXO source. (Same as above ...)`	-	This indicates a hardware issue because we should always be able to lock to the internal OCXO.
`Product error: Unsupported reference frequency. Only 10MHz and 100MHz supported`	-	Incorrect reference frequency specified.

Initialization errors (continued)

Error message	Explanation	Common causes and possible fixes
`Product error: LMKCLKout11 phase measurement out of range. Verify clock configuration and cabling and re-run the test.  Refer to the KS2201A PathWave Test Sync Executive System Setup Guide for further guidance`	-	Power cycling the hardware is required. If the error persists then it might indicate a hardware issue.
`Product error: PLL unlocked after LMKCLKout11 phase adjustment. (Same as above ...)`	-	
`Product error: PLL unlocked after DSTARA/CK100_ STM phase adjustment. (Same as above ...)`	-	
`Product error: FPGA_10M phase alignment to CLK100_STM failed. (Same as above ...)`	-	
`Product error: PXIe_SYNC100 alignment to Sync100_ Base failed. (Same as above ...)`	-	

Rare SSM errors (preceded by the chassis number the SSM is in, the failing function call and the HVI and FW versions)

Error message	Explanation	Common causes and possible fixes

`Product error: PXIe_CLK10 phase measurement out of range. Verify clock configuration and cabling and re-run the test.  Refer to the KS2201A PathWave Test Sync Executive System Setup Guide for further guidance`
 	
Other similar messages:
 	
`Product error: FPGA_10M phase detection failed. (Same as above ...)`
`Product error: FPGA_10M phase unstable. (Same as above ...)`
`Product error: PXIe_CLK10 phase out of range. (Same as above ...)`
`Product error: PXIe_CLK10 phase adjustment failed. (Same as above ...)`
	-	These errors are usually resolved by rerunning the test. If the error persist, try to power cycle the hardware setup. If it still persists, please contact Keysight.

Edition 2022_U0_00, June, 2022
Keysight Technologies, USA

KS2201-90009
www.keysight.com