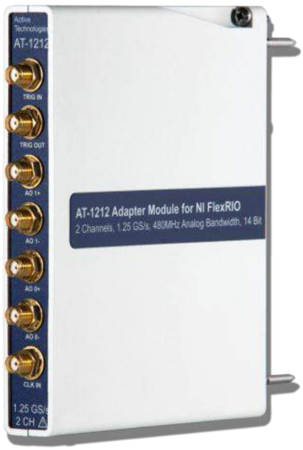
Active Technologies

AT-1212

2 Channels, 1.25 GS/s, 480 MHz Analog BW , 14 Bit

High Speed Signal Generator Adapter Module for NI FlexRIO



User Manual

January, 2013 Rev. 1.3

Active Technologies S.r.l
Via Bela Bartok 29/B
44124 Ferrara, ITALY
Tel +39 0532 91456, Fax: +39 0532 970134
Internet: www.activetechnologies.it
E-mail: info@activetechnologies.it

AT-1212 User Guide and Specifications

The AT-1212 is a two-channel, 1.25 GS/s, 14-bit, 480 MHz Analog Bandwidth, High-Speed Signal Generator adapter module designed to work in conjunction with your NI FlexRIO™ FPGA module. This document contains signal information and specifications for the AT-1212R, which is composed of an NI FlexRIO FPGA module and the AT-1212. This document contains tutorial sections that demonstrate how to generate signals using a LabVIEW FPGA example VI and how to create and run your own LabVIEW project with the AT-1212R.

Note Before configuring your AT-1212, you must install the appropriate software and hardware. Refer to the *NI FlexRIO FPGA Module Installation Guide and Specifications* for installation instructions. Figure 1 shows an example of a properly connected NI FlexRIO device.

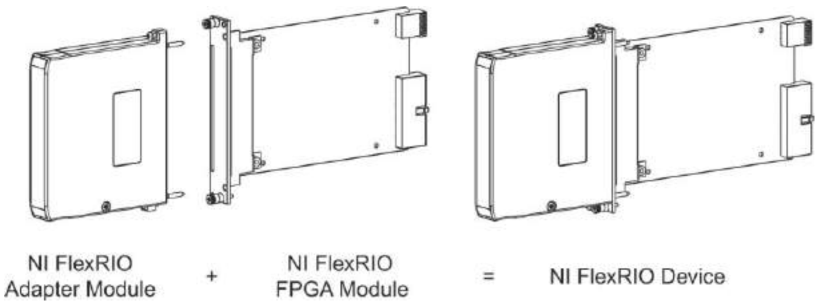


Figure 1. NI FlexRIO Device

Note *AT-1212R* refers to the combination of your AT-1212 adapter module and your NI FlexRIO FPGA module. *AT-1212* refers to your AT-1212 adapter module only.

Electromagnetic Compatibility Guidelines

This product was tested and complies with the regulatory requirements and limits for electromagnetic compatibility (EMC) as stated in the product specifications. These requirements and limits are designed to provide reasonable protection against harmful interference when the product is operated in its intended operational electromagnetic environment.

This product is intended for use in industrial locations. There is no guarantee that harmful interference will not occur in a particular installation, when the product is connected to a test object, or if the product is used in residential areas. To minimize the potential for the product to cause interference to radio and television reception or to experience unacceptable performance degradation, install and use this product in strict accordance with instructions in the product documentation.

Furthermore, any changes or modifications to the product not expressly approved by Active Technologies could void your authority to operate it under your local regulatory rules.

⚠ **Caution** To ensure the specified EMC performance, you must install PXI EMC Filler Panels (National Instruments part number 778700-01) in adjacent chassis slots. For more information about installing PXI EMC filler panels on your NI 5762R, refer to the *Appendix: Installing EMI Controls* section of this document.

⚠ **Caution** To ensure the specified EMC performance, operate this product only with shielded cables and accessories.

⚠ **Caution** This product is sensitive to electrostatic discharge (ESD).

How to Use Your NI FlexRIO Documentation Set

Refer to Figure 2 and Table 1 for information about how to use your NI FlexRIO documentation set.

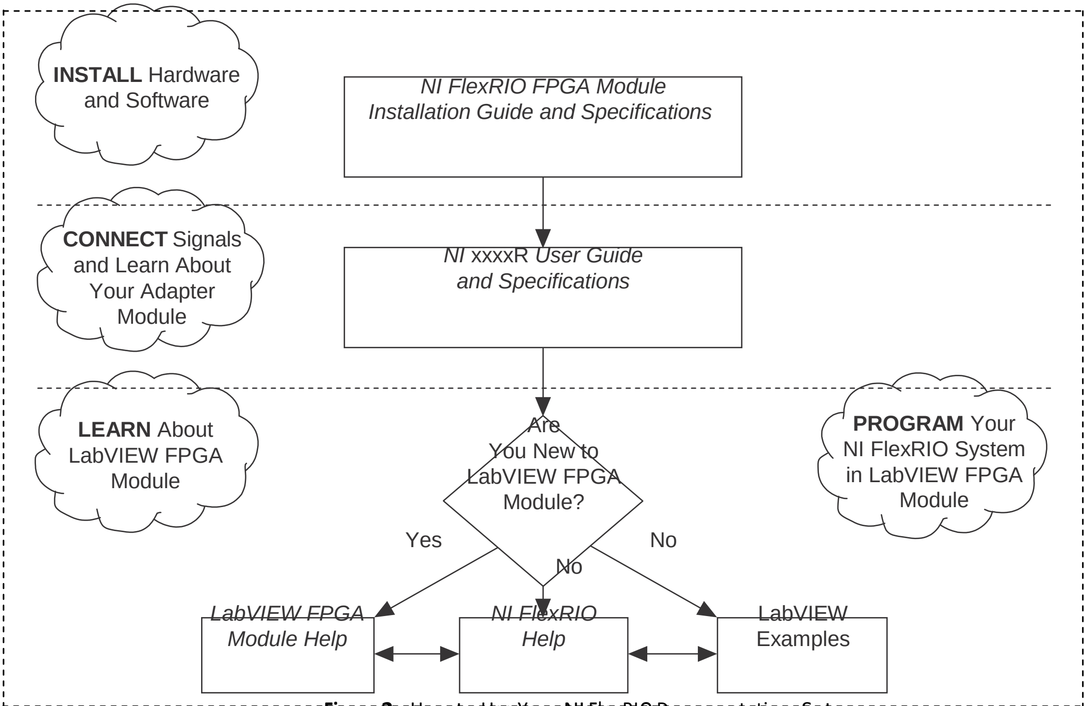


Figure 2. How to Use Your NI FlexRIO Documentation Set

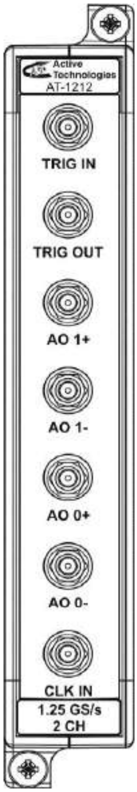
Table 1. NI FlexRIO Documentation Locations and Descriptions

Document	Location	Description
*NI FlexRIO FPGA Module Installation Guide and Specifications**	Available in your FPGA module hardware kit and from the Start Menu.	Contains installation instructions for your NI FlexRIO system and specifications for your FPGA module.
NI Adapter Module User Guide and Specifications *	Available in your adapter module hardware kit and from the Start Menu.	Contains signal information, examples, and specifications for your adapter module.
LabVIEW FPGA Module Help*	Embedded in *LabVIEW Help*.	Contains information about the basic functionality of LabVIEW FPGA Module.
*NI FlexRIO Help**	Embedded in *LabVIEW FPGA Module Help*.	Contains FPGA module, adapter module, and CLIP configuration information.
LabVIEW Examples	Available in NI Example Finder.	Contains examples of how to run FPGA VIs and Host VIs on your device.
Other Useful Information on ni.com		
ni.com/ipnet	Contains LabVIEW FPGA functions and intellectual property to share.	
ni.com/flexrio	Contains product information and data sheets for NI FlexRIO devices.	
* These documents are also available at ni.com/manuals.		

Front Panel and Connector Pinouts

Table 2 shows the front panel connectors and signal descriptions for AT-1212. Refer to the *Specifications* sheet for additional signal information.

Caution To avoid permanent damage to the AT-1212, disconnect all signals connected to the AT-1212 before powering down the module, and connect signals only after the adapter module has been powered on by the NI FlexRIO FPGA module.

Device Front Panel	Connector	Signal Description
	TRIG IN	SMA input connector for Trigger IN
	TRIG OUT	SMA output connector for Trigger Out
	AO 1+	Differential analog output+ channel 1
	AO 1-	Differential analog output- channel 1
	AO 0+	Differential analog output+ channel 0
	AO 0-	Differential analog output- channel 0
	CLK IN	SMA input connector for external clock

* When using single-ended output, terminate the unused output with a 50 Ω terminator.

Table 2. AT-1212 Front Panel Connectors

Caution Connections that exceed any of the maximum ratings of any connector on the AT-1212R can damage the device and the chassis. Active Technologies is *not* liable for any damage resulting from such signal connections. For the maximum input and output ratings for each signal, refer to the *Specifications* sheet.

AT-1212 Component-Level Intellectual Property (CLIP)

The LabVIEW FPGA Module includes a feature for HDL IP integration called CLIP. NI FlexRIO devices support two types of CLIP: user-defined and socketed.

•*User-defined CLIP* allows users to insert HDL IP into an FPGA target, enabling VHDL code to communicate directly with an FPGA VI.

•*Socketed CLIP* provides the same IP integration functionality of the user-defined CLIP, but it also allows the CLIP to communicate directly with circuitry external to the FPGA. Adapter module socketed CLIP allows your IP to communicate directly with both the FPGA VI and the external adapter module connector interface.

Figure 4 shows the relationship between an FPGA VI and CLIP.

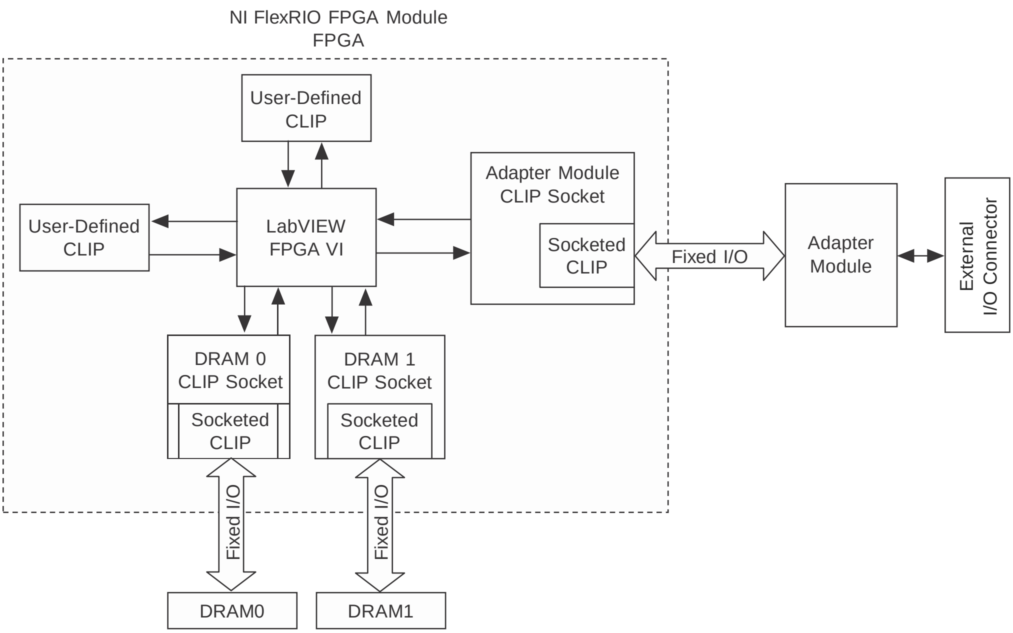


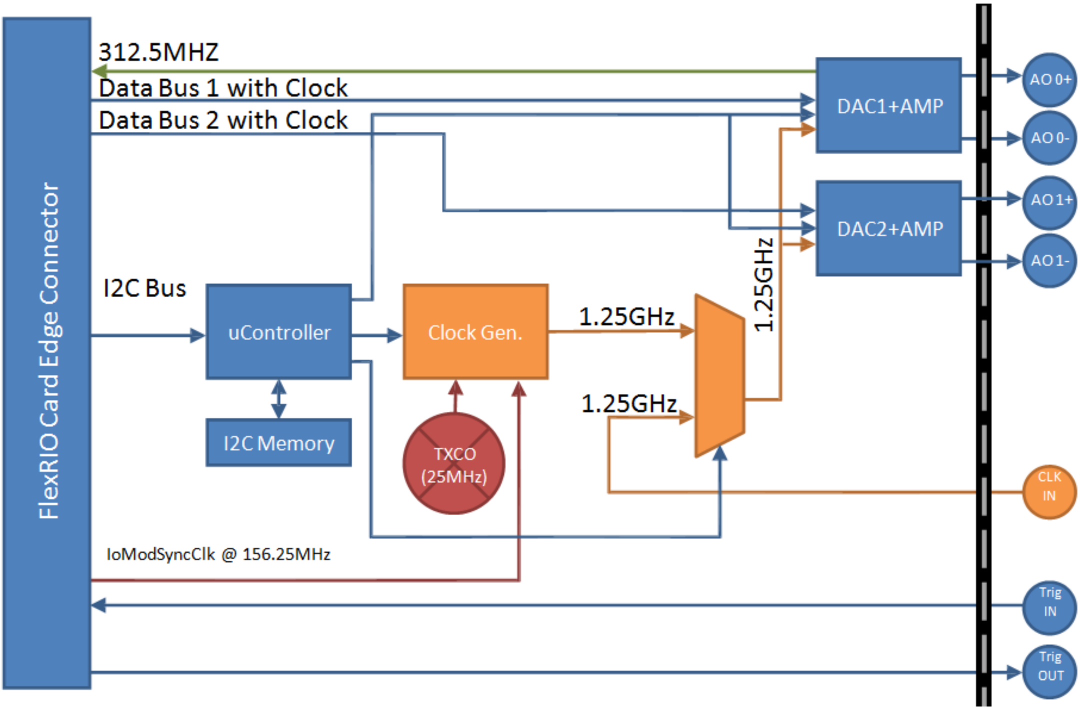
Figure 4. CLIP Relationship

The AT-1212 ships with socketed CLIP items that are used to add module I/O to the LabVIEW project. The AT-1212ships with the following CLIP item: *AT_1212_IOModule_CLIP.*

AT-1212 CLIP DESIGN

There are two DACs installed on the board that control the two available analog output channels. The DAC contains two parallel LVDS input ports consisting of 14 differential LVDS signals DB[13:0]. Each data port runs internally at half the speed of the DAC Clock (1.25 GHz), so the user needs to guarantee them the correct data rate.

The installed multiplexer selects the DAC clock source: it can be internal (from Clock Generator circuit output) or external (from SMA connector).
The adapter module has a Clock Generator circuit that provides the 1.25 GHz clock to the DAC; the clock generator source clock can be provided by 25 MHz onboard TCXO or by PXI_e DSTARA backplane connection.
The uController write/read the DAC parameters, the Clock Gen registers and control the multiplexer output. It performs also the Power up sequencing.



This CLIP provides read/write access to all FPGA look-up tables that are included in the project.
The data read from the look up tables are serialized by the CLIP to provide the requested data rate to the DAC.

Clip files: AT_1212_IOModule_CLIP.xml; AT_1212_IOModule_CLIP.vhd; Inst_PLL_156M.vhd; i2c_ctrl.vhd; OSerdes4_1.vhd; AT1212_Constraints.ucf; AT1212_IOModule.tbc

FPGA I/O Interface

datadac_in0_ch0,datadac_in2_ch0, datadac_in4_ch0, datadac_in6_ch0	Input[13..0]	Data for DAC Port 1 CH0 The user needs to send data to the Clip Inputs at IO Module Clock rate (156.25 MHz); the datadac_in0..datadac_in6 are internally serialized by 4 to obtain a data rate of 625 MS/s.
datadac_in1_ch0,datadac_in3_ch0, datadac_in5_ch0, datadac_in7_ch0	Input[13..0]	Data for DAC Port 2 CH0 The user needs to send data to the Clip Inputs at IO Module Clock rate (156.25 MHz); the datadac_in1..datadac_in7 are internally serialized by 4 to obtain a data rate of 625 MS/s.
datadac_in0_ch1,datadac_in2_ch1, datadac_in4_ch1, datadac_in6_ch1	Input[13..0]	Data for DAC Port 1 CH1 The user needs to send data to the Clip Inputs at IO Module Clock rate (156.25 MHz); the datadac_in0..datadac_in6 are internally serialized by 4 to obtain a data rate of 625 MS/s.
datadac_in1_ch1,datadac_in3_ch1, datadac_in5_ch1, datadac_in7_ch1	Input[13..0]	Data for DAC Port 2 CH1 The user needs to send data to the Clip Inputs at IO Module Clock rate (156.25 MHz); the datadac_in1..datadac_in7 are internally serialized by 4 to obtain a data rate of 625 MS/s.
datai2c0..datai2c5	Input[7..0]	Data transferred by I2C write
readdataI2C	Output[7..0]	Data retrieved by I2C read
startRDI2C	Input	Start I2C bus read
startWRI2C	Input	Start I2C bus write
reset_in	Input	Reset the DCMs and FAM
lockedfast	Output	Check if the FPGA DCM has locked or not

trig_in	Output	Check if the SMA Trigger IN signal is high or low
trig_out	Input	Send the Trigger Output to the FAM
clkenable	Input	Enable/Disable the FPGA DCM clock outputs
clocksel	Input	Select the clock source for DAC/OSerdes. The user should set it to false if he needs to use the TCXO onboard clock or set it to true if he needs to use the clock coming from the Timing Board (PXIe backplane).

Clocking Scheme

The DAC clock can be sourced from the internal Clock Generator circuit on the adapter module or from the outside using the SMA CLK IN (external clock).
The Clock Generator clock can be sourced from the internal TCXO oscillator (25 MHz) or from the PXI Express backplane clock.
In software, use the *ClockSelection1212.vi* to select from the different clock sources.

If the user selects **Internal** and **From Oscillator**, the 25 MHz onboard TCXO will provide the clock to the Clock Generator input. The clock generator generates 1.25GHz, the DAC divides it by 4 and it provides 312.5 MHz clock to FlexRIO FPGA.
The BUFR and PLL_156M instances divide the 312.5 MHz clock by 2 and it provide the 312.5 MHz and 156.25 MHz clocks to the OSerdes and FlexRIO interface.

If the user selects **Internal** and **From FPGA Clock**, the FAM takes the 156.25MHz from the PXI Express backplane, so the user should provide the clock through DSTARA clock by using the NI timing board.

If the user selects **External**, the DAC clock (1.25 GHz) has to be provided by outside using the SMA CLK IN bypassing the clock generator circuit.

Component Level IP Clocks

Name	Connection
Clockin	DSTARA Clock (156.25 MHz): see above for clocking scheme usage
Clockin40m	40 MHz FlexRIO Clock: used on the LabView interface of the clip (ReadWriteTablesTest.vi)
IO Module Clock	156.25 MHz clock from FAM

Cables

Use any shielded 50 Ω coaxial cable with an SMA plug end to connect to the AO 0+, AO 0-, AO 1+, AO 1-, TRIG IN, TRIG OUT and CLK IN connectors on the AT-1212 front panel.
For more information about connecting I/O signals on your device, refer to the *Specifications* sheet.

LabView Interface

Target Side - ReadWriteTablesTest1212.vi

There are two DACs installed on the board that control the two available analog output channels. The DAC contains two parallel LVDS input ports consisting of 14 differential LVDS signals DB[13:0]. Each data port runs internally at half the speed of the DAC Clock (1.25GHz), so the user needs to guarantee them the correct data rate.

PORT 1 DATA - CH0
The LabView interface of the CLIP reads data from the look up tables (Memory1,Memory3,Memory5,Memory7) at IO Module Clock rate (156.25 MHz); the datadac_in0_ch0/ datadac_in2_ch0/datadac_in4_ch0/datadac_in6_ch0 are internally serialized by 4 to obtain a data rate of 625MS/s.

PORT 2 DATA - CH0
The LabView interface of the CLIP reads data from the look up tables (Memory2,Memory4,Memory6,Memory8) at IO Module Clock rate (156.25 MHz); the datadac_in1_ch0/ datadac_in3_ch0/datadac_in5_ch0/datadac_in7_ch0 are internally serialized by 4 to obtain a data rate of 625MS/s.

PORT 1 DATA - CH1
The LabView interface of the CLIP reads data from the look up tables (Memory9,Memory11,Memory13,Memory15) at IO Module Clock rate (156.25 MHz); the datadac_in0_ch1/ datadac_in2_ch1/datadac_in4_ch1/datadac_in6_ch1 are internally serialized by 4 to obtain a data rate of 625MS/s.

PORT 2 DATA - CH1
The LabView interface of the CLIP reads data from the look up tables (Memory10,Memory12,Memory14,Memory16) at IO Module Clock rate (156.25 MHz); the datadac_in1_ch1/ datadac_in3_ch1/datadac_in5_ch1/datadac_in7_ch1 are internally serialized by 4 to obtain a data rate of 625MS/s.

The Clip provides also access to DDS/Arbitrary mode selection,Clock Selection, Trigger In and Trigger Out resources, START_CH Generation command and EN_CH0/EN_CH1 to enable/disable channels. The user has to load each look-up tables with the same waveform and the CLIP reads them updating the Start/Stop Address and Increment parameters to convey parallel data stream at the converter at high speeds.

Control/Indicator	Description
START_CH	Start the waveform generation
EN_CH0	Enable/disable the start on channel 1
EN_CH1	Enable/disable the start on channel 2
Trig Selection	If *Trig Selection* is true, the global software trigger signal is incoming from PXIe backplane (PXI_Trig0). It generates the synchronized RUN for multiple modules. If Trig Selection is false, the software trigger is incoming from START_CH signal.
Trig Reset	If *Trig Reset* is true, it resets the trigger(start) condition.
SysReset	System reset
IO Module\lockedfast (indicator)	If the indicator is true, the FPGA DCM has locked
IO Module\trig_out	It sets the Trigger Out value
IO Module\trig_in(indicator)	It reads the Trigger In value.
Write MEM CH0 / Write MEM CH1	If true, it writes the data contained into Write DATA control at the address specified by Write Address into CH0/CH1
Write DATA	Memory data
Write Address	Memory address
ClockSEL	Select the clock source for DAC/OSerdes. If false, it selects the Inst_DCM_CLOCK_DIV4 outputs; if true it selects the Inst_PLL DCM clock outputs. The user should set it to false if he needs to use the TCXO onboard clock or set it to true if he needs to use the clock coming from the Timing Board (PXIe backplane).
ClkEnable	It enables the FPGA DCM clock output.
StartAddress1_CH0..StartAddress8_CH0 StartAddress1_CH1..StartAddress8_CH1	Start Address represents the starting pointer address of the look-up tables.

	StartAddress is 1..8 depending on the look-up table
StopAddress1_CH0..StopAddress8_CH0 StopAddress1_CH1..StopAddress8_CH1	Stop Address represents the stopping pointer address of the look-up tables. StopAddressN = (NumSamples-Increment)+(N-1)
Increment_CH0 / Increment_CH1	In Arbitrary mode the Increment is fixed at 16; in DDS it depends on the DAC sampling rate, the output frequency and on the number of samples.
Data1..Data16	Retrieved data from memories
DDS/ARB CH0 DDS/ARB CH1	DDS/ARB: set to false enters in DDS mode waveform generation; set to true enters in Arbitrary mode.[1] In Arbitrary mode the Increment is fixed at 16 and Start/Stop Address represent the start/end point of the look-up tables.
DataI2c0..Datai2c5	Data transferred by I2C write transfer
ReadDataI2c	Data retrieved by I2C read transfer
StartRDI2c	Start I2c read transfer
StartWRI2c	Start I2c write transfer

[1] In DDS mode the lookup table contains one cycle of the waveform to be generated and typically contains 1024 to 8192 sample points which represent the waveform. For our examples we are using a 2048-sample reference waveform, representing one cycle of our repetitive waveform.
The core component of a DDS waveform generator is the accumulator. The accumulator is a running counter which stores the current phase value of the generated waveform.
Increment parameter represents the value the accumulator is updated determining the frequency of the generated waveform.

LabView Interface

Host Side

Host VI - CLIP	**Description**
InitModule1212(Host).vi	Initialize the 1212 adapter module. If the adapter has been correctly initialized, DAC A Aligned / DAC B Aligned and FPGA DCM Locked indicators must be true. To enter in Debug mode, set DEBUG as true. **Input Parameters** *FPGA VI Reference IN* : FPGA reference *RIO Device* *CLOCK SELECTION*:DAC Clock source selection *INTERNAL CLOCK*:Clock Generator input source *Error in* **Output Parameters** *FPGA DCM LOCKED*: if true, the FPGA DCM has locked *DAC A Aligned*: if true the first DAC has been correctly initialized and aligned *DAC B Aligned*: if true the second DAC has been correctly initialized and aligned *Error out* **IMPORTANT NOTE**: the user **should NOT** hit the **ABORT** button during the initialization
SetVocm.vi	Set the Vocm voltage

Host VI – Target Example	Description
WriteTables1212(Host).vi	The WavefArray contains the waveform samples that need to be stored into the look-up tables. The VI stores the samples into the tables on the selected channel.
ReadTables1212(Host).vi	The vi calculates the Start/Stop Addresses and the Increment parameter and sends them to the target side. Those parameters are necessary for look-up tables reading (see section above). **Input parameters** NumSamples: the waveform length in samples DDS/ARB: DDS (false) or Arbitrary(true) generation mode FSampleDAC: DAC sampling rate FoutDDS: DDS waveform frequency requested by the user. The vi calculates the Increment parameter using FSampleDAC and FoutDDS. resource name: FlexRIO resource name error in **Output parameter** error out
StartGeneration1212(Host).vi	Start the waveform generation on the selected channels. If Sync is true, the adapter modules waits for global software trigger (PXIe-DSTARB) to start the generation.
StopGeneration1212(Host).vi	Stop the waveform generation on the selected channels

Software Prerequisites

1. Open the installation folder \XML_VHDL
2. Copy the \XML_VHDL\AT 1120 and \XML_VHDL\AT 1212 folders into the following paths:

   64-Bit Paths

   C:\Program Files (x86)\National Instruments\Shared\FlexRIO\IO Modules\AT 1120

   C:\Program Files (x86)\National Instruments\Shared\FlexRIO\IO Modules\AT 1212

   32-Bit Paths

   C:\Program Files \National Instruments\Shared\FlexRIO\IO Modules\AT 1120

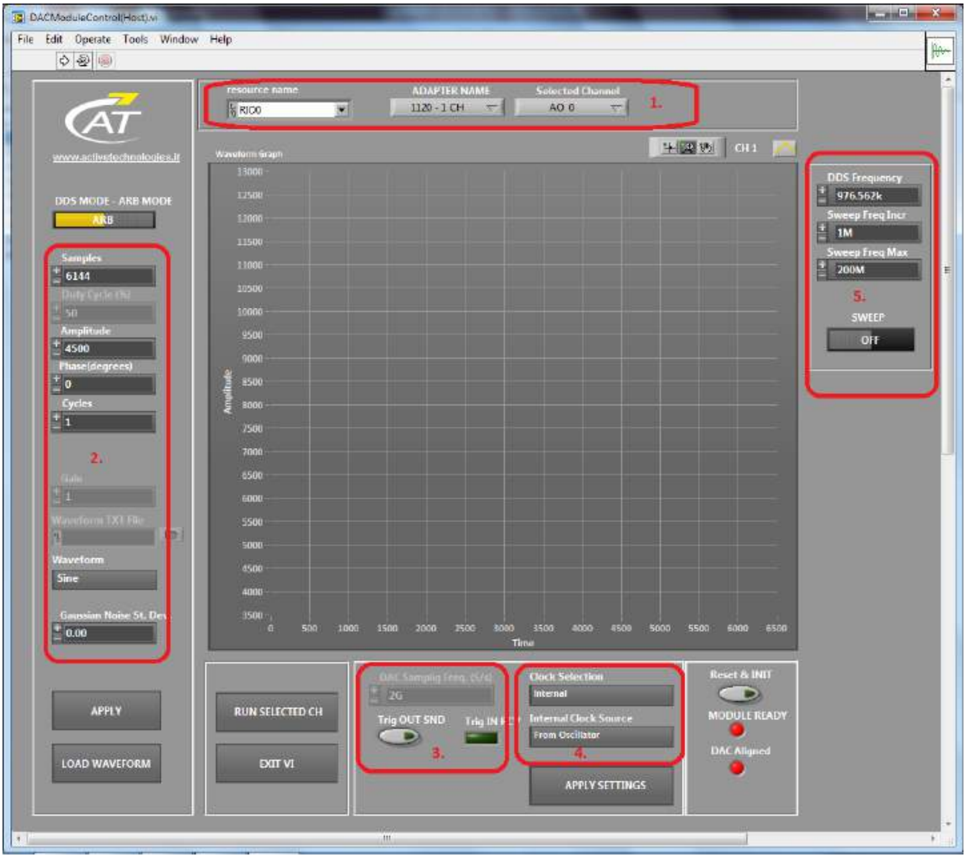
   C:\Program Files \National Instruments\Shared\FlexRIO\IO Modules\AT 1212

Example Code Review for AT-1212 adapter module

DACModuleControl(Host).exe

Software requirements: Labview 2012 + Modulation Toolkit

This LabView application gives the user full access to all the main 1120/1212 features. The control panel provides a *waveform generator* style approach to the FAM, allowing to set the parameters of the signals that will be load and generated by the FAM.
The source file **DACModuleControl(Host).vi** is located into the folder *LabViewDesigns\AT_HS_Signal_GeneratorLV2012* and it needs the **NI Modulation** toolkit (http://sine.ni.com/nips/cds/view/p/lang/en/nid/210568)to run properly; otherwise it is possible to make some modifications on the code to exclude the QAM, BPSK modulations and run it without the toolkit.



- Select the *resource name*, the *adapter name* (1120 or 1212), the *RIO DEVICE* and the analog output channel to control (AO 0 for 1120 FAM, AO 0 / AO1 for 1212 FAM).

- Set the parameters of the most common waveforms selectable by *Waveform enum control* :

  a) samples number (must be multiple of 16), duty cycle, amplitude (from 0 to 8191)
  b) phase and number of cycles (related to the waveform output frequency)
  c) Waveform type and the standard deviation of the gaussian noise that it is possible to add to the waveform.
  The user can also import a waveform from file or generate a Chirp pattern, BPSK signals and QAM modulation adding AWGN, IQ Impairments, Phase Noise to the output pattern.

- Press the *APPLY* button to display the changes on the graph .

- Press the *LOAD WAVEFORM* button to upload the data into the FPGA look-up tables.

- Press *RUN SELECTED CH* button to start the waveform generation.

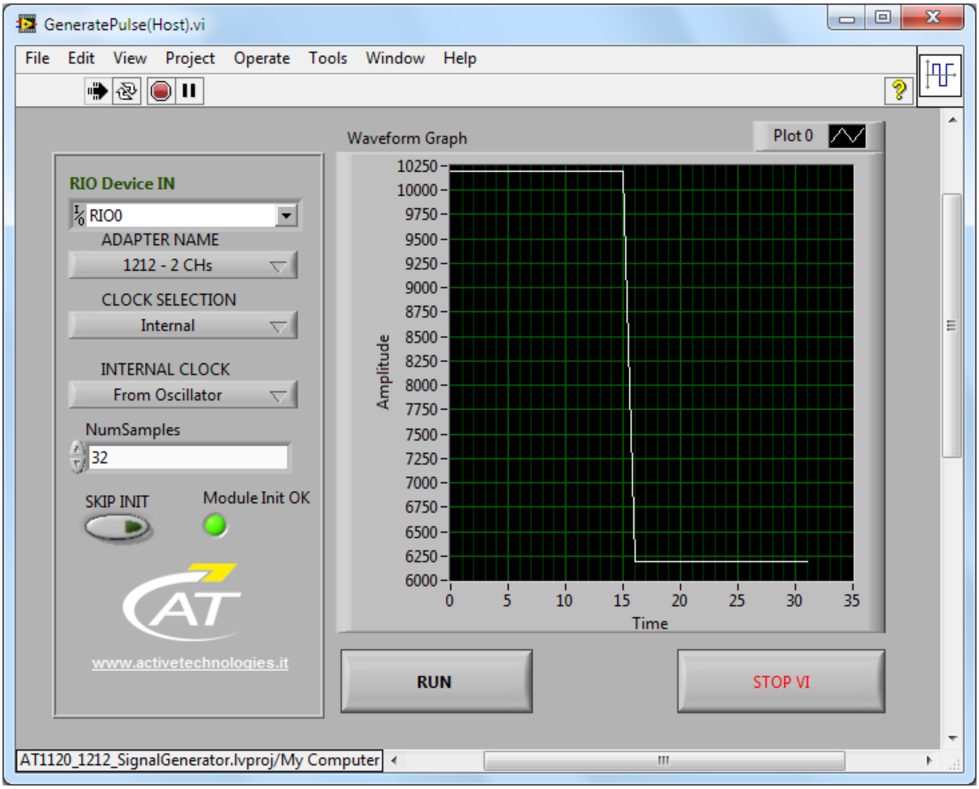
Clock Selection: select the DAC clock source. It can be Internal (from Clock Generator circuit) or External from SMA connector (1.25GHz).
Internal Clock Source: select the source clock for the Clock Generator circuit.
It can be From Oscillator (onboard TCXO) or From FPGA Clock (from PXIe Backplane - DSTARA Timing board).

DDS mode waveform generation parameters.
When the FAM works in DDS mode, the user should upload a 2048 samples waveform and set its parameters using DDS frequency, Sweep Freq.Incr and Sweep Freq.Max controls.
It is also possible to generate a frequency sweep, setting the increment and the stop frequency and then pressing the SWEEP ON/OFF button.

GeneratePulse(Host).vi

Software requirements: LabView 2012

This VI is an user friendly example that generates pulse signals. On AT-1212 adapter modules it starts/stops both channels.



- Select the *resource name* and the *adapter name* (1120 or 1212).
- *Clock Selection*: select the DAC clock source. It can be Internal (from Clock Generator circuit) or External from SMA connector (1.25GHz).
  Internal Clock Source: select the source clock for the Clock Generator circuit. It can be From Oscillator (onboard TCXO) or From FPGA Clock (from PXIe Backplane - DSTARA Timing board).
- Set the NumSamples: the samples number of the pulse waveform. It must be multiple of 16.

Module Init OK: indicator that states if the module is properly initialized or not.
SKIP INIT: if true, the FAM skips the initialization phase.

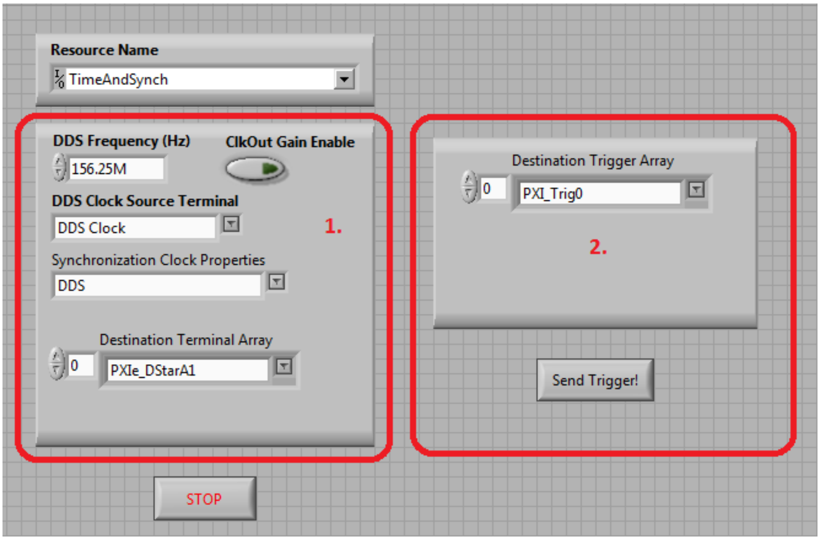
Once the VI has started, the pulse waveform will be uploaded into the FPGA look-up tables on all available channels

- Press the *RUN* button to start the waveform generation

DDS and Trigger Route.vi

Software requirements: LabView 2012

This VI gives the user access to the clock generated by PXIe-6674T NI Timing board and to the DSTARA/DSTARB clock/global trigger routing.



- Set *the DDS frequency* to 156.25 MHz and fill the *Destination Terminal Array.* The Destination Terminal Array should contain all the PXIe_DSTARA connection to the FlexRIO boards that will provide the same reference clock to the adapter module (refer to the chassis manual).

- Fill the *Destination Trigger Array* with the PXIe_DSTARB global trigger routing connections. The FlexRIO boards will receive now the same trigger signal.

- Press the *Send Trigger* button to send it to all connected FAMs.

Using Your AT-1212 with a LabVIEW FPGA Example VI

Note You must install the software before running this example. Refer to the NI FlexRIO FPGA Module Installation Guide and Specifications for more information about installing your software.

You must copy the following files (tbc,xml and vhdl) in the FlexRIO working paths

a. Open the installation folder \XML_VHDL
b. Copy the \XML_VHDL\AT 1120 and \XML_VHDL\AT 1212 folders into the following paths:
64-Bit Paths
C:\Program Files (x86)\National Instruments\Shared\FlexRIO\IO Modules\AT 1120
C:\Program Files (x86)\National Instruments\Shared\FlexRIO\IO Modules\AT 1212
32-Bit Paths
C:\Program Files \National Instruments\Shared\FlexRIO\IO Modules\AT 1120
C:\Program Files \National Instruments\Shared\FlexRIO\IO Modules\AT 1212

The NI FlexRIO Adapter Module Support software includes a variety of example projects to help get you started creating your LabVIEW FPGA application. This section explains how to use an existing LabVIEW FPGA example project to generate signals with the AT-1212. This example requires at least one SMA cable for connecting signals to your AT-1212.

Note The examples available for your device are dependent on the version of the software and driver you are using.

Each AT-1212 example project includes the following components:

• A LabVIEW FPGA VI that can be compiled and run on the FPGA embedded in the hardware. This VI is referred to as the FPGA Target VI.

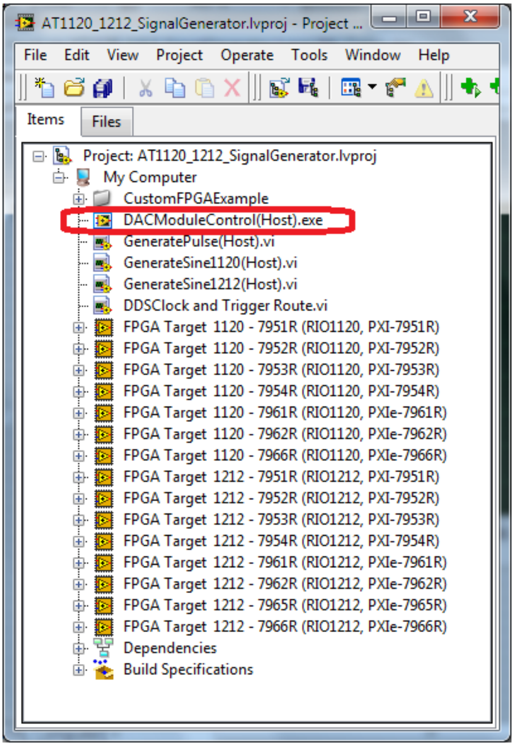
• A VI that runs on Windows that interacts with the LabVIEW FPGA VI. This VI is referred to as the Host VI.

Complete the following steps to run an example that generates a waveform on AO 0+ of the AT-1212.

1. Connect one end of an SMA cable to AO 0+ on the front panel of the AT-1212 and the other end of the cable to your oscilloscope input (50 Ω). Tap the unused output (AO 0-) with a 50 Ω load.

2. Launch LabVIEW.

3. Select and open AT1120_1212_SignalGenerator.lvproj LabView project.



4. In the **Project Explorer** window, open **DACModuleControl (Host).vi** under **My Computer**. The host VI opens. The Open FPGA VI Reference function in this VI uses the NI 7962R as the FPGA target by default. If you are using an NI FlexRIO FPGA module other than the NI 7962R, select the FPGA target by using RIO Device *Menu Ring*.

5. On the front panel, in the **resource** pull-down menu, select an AT-1212 resource that corresponds with the target configured in step 4.

6. Select **1212- 2CHs** in the *Adapter Name* pull-down menu.

7. Select **AO 0** in the *Selected Channel* pull-down menu.

8. Select **ARB** in the *DDS-ARB* control

9. Select **Internal** in the *Clock Selection* pull-down menu and **From Oscillator** in the *Internal Clock Source* pull-down menu.

10. Click the Run button to run the VI.

11. Wait until Module Ready and DAC Aligned leds become green

12. Select the **Waveform** you want to generate by using Waveform pull-down menu and choose its parameters

13. Press the *Apply button* to update the Waveform Graph

14. Press the *Load Waveform* button to load the samples into the FAM.

15. Press the *RUN SELECTED CH.* button to start the waveform generation

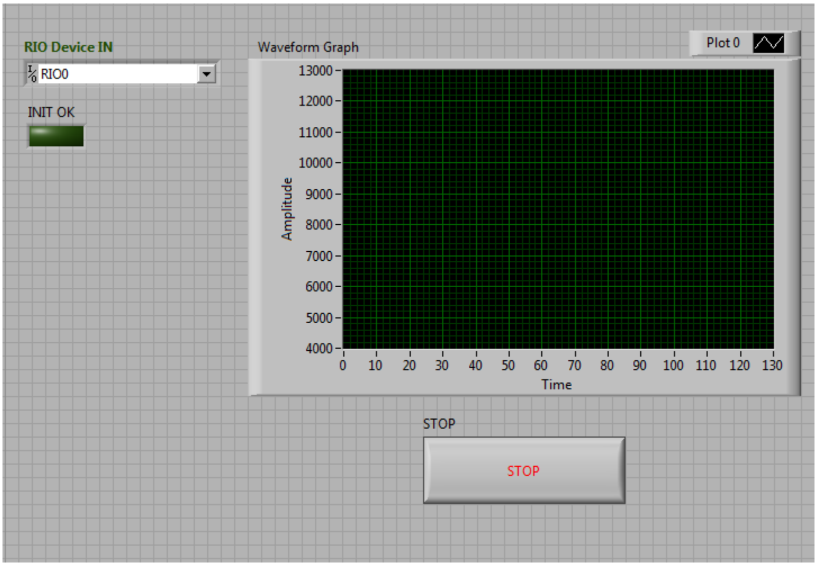
16. Press the *STOP* button to stop the generation

17. Press the EXIT VI button to stop the VI.

Creating a Host VI on an FPGA Target

This section explains how to set up your target and create an host VI for waveform generation.
Note: the following tutorial can be found under the directory \AT_HS_SignalGenerator\GenerateSine1212(Host).vi

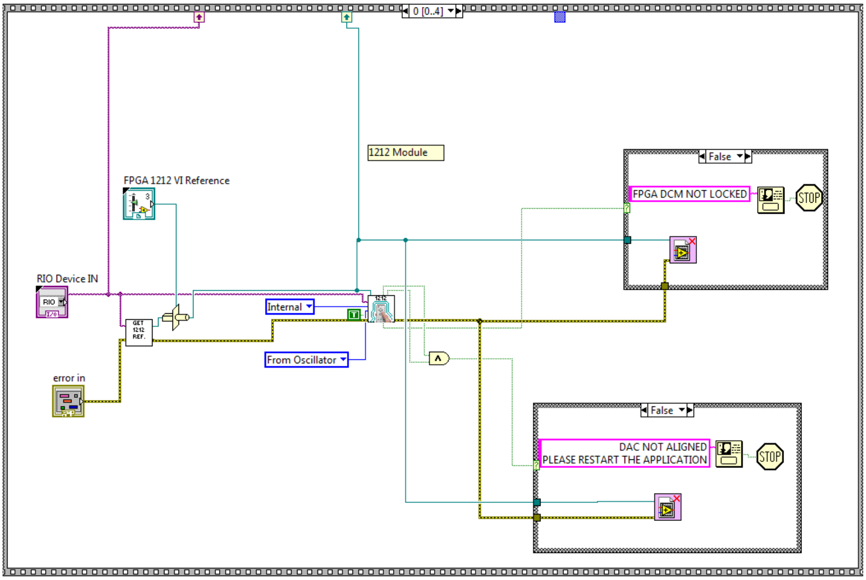
1. In the **Project Explorer** window, right-click **My Computer** and select **New»VI**. A blank VI opens.
2. Add a **Waveform Graph** indicator in the front panel
3. Add the RIO Device control, located on Modern → I/O palette
4. Add a STOP button, located on Express→Buttons & Switches
5. Add a LED indicator (INIT OK), located on Express→LEDs



6. Select **Window»Show Block Diagram** to open the VI block diagram
7. Add a **Stacked Sequence Structure**
8. Place **the GetFPGA1212Reference.vi** to get the reference to the 1212 FPGA Target *(AT_HS_Signal_Generator\FPGA Bitfiles\AT1212SignalGenerator_FPGATarget.lvbitx)*
9. *Connect resource name →RIO Device IN* and *error in →error in* control

10. Place the **InitModule1212(Host).vi**, located on *\AT_HS_Signal_Generator\1212 Module* folder

11. Wire **InitModule1212(Host).vi** controls and indicators like the picture below



RIO Device → RIO Device IN, CLOCK SELECTION → Internal, INTERNAL CLOCK → From Oscillator, Master → TRUE,DAC A Aligned and DAC B Aligned connect to AND operator, FPGA DCM Locked → case structure.

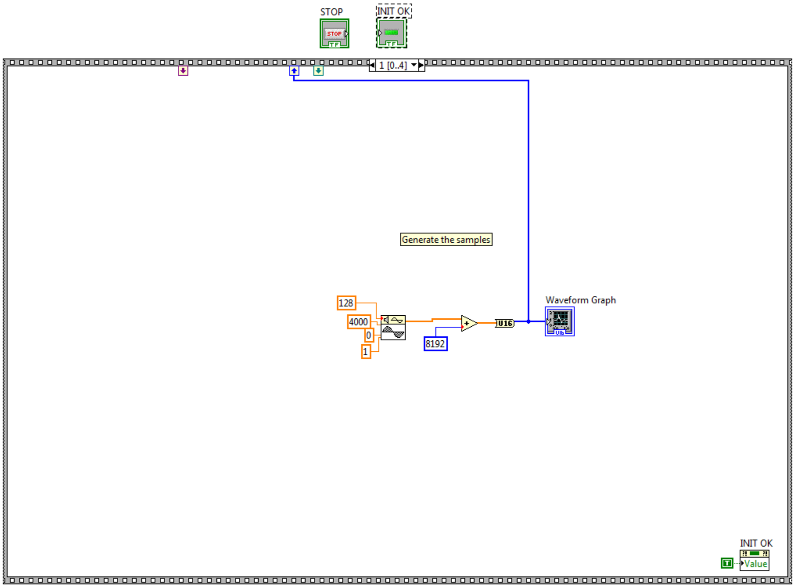
The FPGA DCM Locked and DAC A Aligned/DAC B Aligned monitor if the initialization stage has been correctly terminated.

12. Add a **Frame After** in the stacked sequence: we will generate the waveform samples

13. Add **Sine Pattern.vi** from *Signal Generation Palette*

14. Samples →128, Amplitude →4000, Cycle →1, Phase →0, add 8192 to the waveform array and connect the samples array to the Waveform Graph.
    If the VI execution arrives at the second frame, it means that the adapter module has been

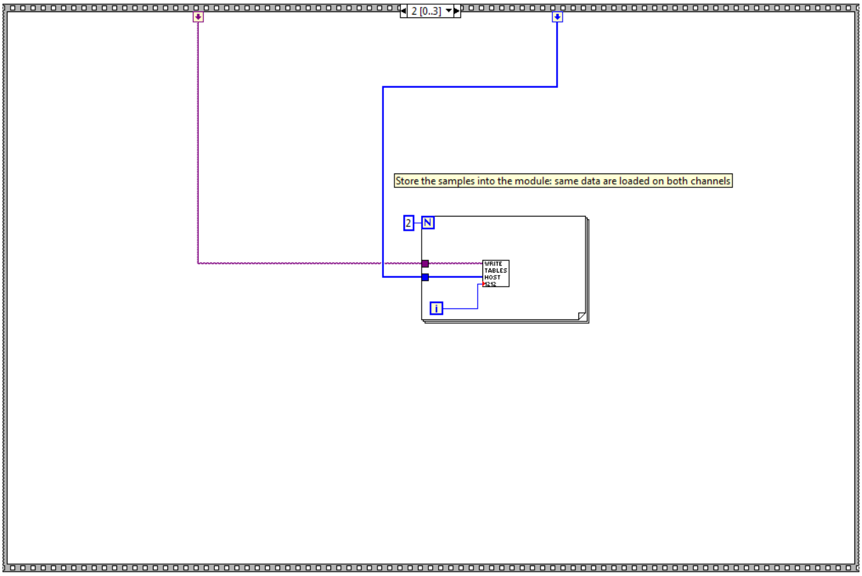
correctly initialized, so connect True to the INIT OK led indicator.



15. Add a **Frame After** in the stacked sequence: we will load the generated samples into the channels module

16. Add a **For Loop** structure and wire 2 to the Loop count.

17. Place the **WriteTables1212(Host).vi**, located on *\AT_HS_Signal_Generator\1212 Module* folder, into the For Loop structure. Connect the RIO Device, WavefArray and Select Channel controls like in the picture below



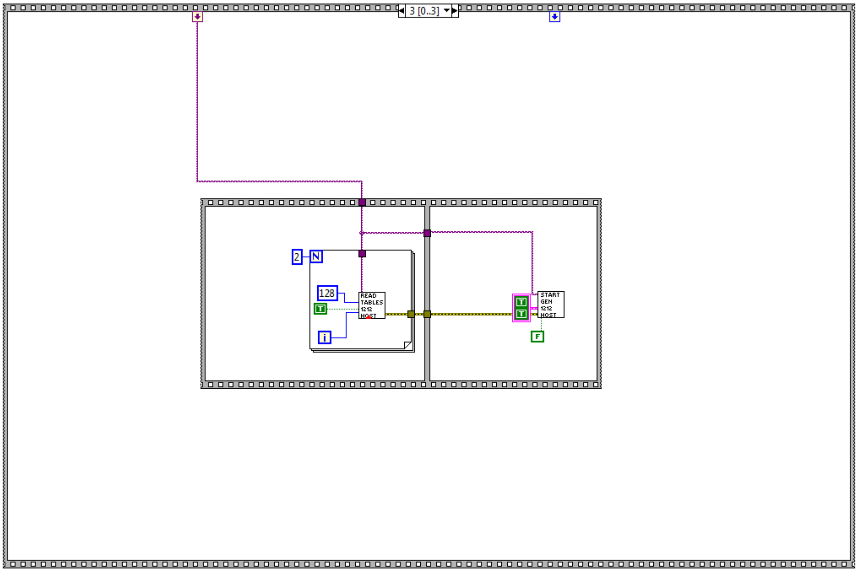
18. Add a **Frame After** in the stacked sequence: we will start the waveform generation

19. Add a **Flat Sequence Structure**

20. In the first frame add a **For Loop** structure and wire 2 to the Loop count.

21. Place the **ReadTables1212(Host).vi**, located on *\AT_HS_Signal_Generator\1212 Module* folder, into the For Loop structure. Connect the *NumSamples →128, DDS/ARB → True and SelectChannel → index i of the For Loop*.

22. In the second frame place the **StartGeneration1212(Host).vi**, located on *\AT_HS_Signal_Generator\1212 Module* folder. Connect the Channel Cluster → TRUE/TRUE and SYNC → False.

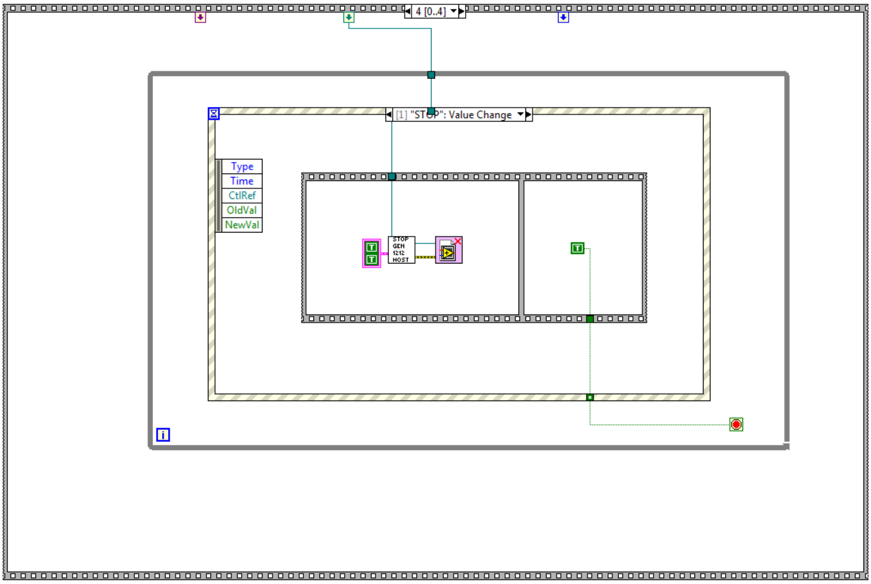


23. Add a **Frame After** in the stacked sequence

24. Add a **While Loop** structure and a **Case** structure inside the loop

25. Place an Event Structure inside a While Loop Connect and *Add an Event Case...* on the Stop button value change.
    Place the **StopGeneration1212 (Host).vi**, located on *\AT_HS_Signal_Generator\1212 Module* folder. Connect the FPGA VI Reference IN → FPGA Reference and place *Close FPGA VI Reference* block located on FPGA Interface wiring it like in the picture below.
    The second frame contains the True constant to stop the VI.

26.



Running the Host VI

1. Connect one end of an SMA cable to AO 0+ on the front panel of the AT 1212 and the other end to the oscilloscope (50 Ω input).

2. Connect one end of an SMA cable to AO 1+ on the front panel of the AT 1212 and the other end to the oscilloscope (50 Ω input).

3. Tap the unused inputs (AO 0- / A1 -) with a 50Ω load.

4. Open the front panel of **GenerateSine1212(Host).vi.**

5. Click the Run button to run the VI.

6. Wait for module initialization.

7. The AT-1212 generates two 128 points sine waveforms.

8. Click the STOP button on the front panel to stop the module and the VI.

Creating a Custom FPGA Target

This section explains how to create your custom target and setup an host VI for waveform generation.

PLEASE NOTE THE FOLLOWING:

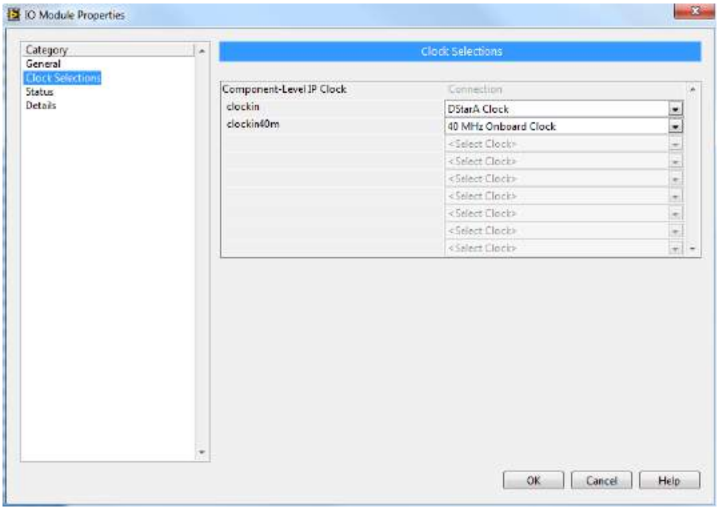
- The custom FPGA Target of this tutorial reads a sine waveform data from the Memory1 lookup table (2048 samples) and send the same data to both channels and to the OSerdes.
- The data rate for the DAC has not been respected to simplify the code implementation (the same data have been sent to all the serdes), so the sine output rate will be divided by 4.
- ***Note****: the following tutorial can be found under the directory \AT_HS_SignalGenerator\FPGA_Tutorial_1212.vi* and *\AT_HS_SignalGenerator\CustomFPGAExample1212(Host).vi.*

1. In the Project Explorer window, right-click My Computer and select **New»Targets and Devices**.
2. In the **Add Targets and Devices on My Computer** dialog box, select the **Existing Target** or Device option button and expand FPGA Target. The target is displayed.
3. Select your device and click OK. The target and target properties are loaded into the Project Explorer window.
4. Right-click **FPGA Target()** in the Project Explorer window and select **New>>FPGA-Base Clock**. Select the **IO Module Clock 0 resource**, Compile for single frequency and set the frequency at **156.25 MHZ**. Click **OK**.
5. Right-click **FPGA Target()** in the Project Explorer window and select **New>>FPGA-Base Clock**. Select the **DStarA Clock**, Compile for single frequency and set the frequency at **156.25 MHZ**. Click **OK**.
6. Click on the "+" button to add the Component-Level IP File Path: browse the **AT_1212_IOModule_CLIP.xml** file in the \XML_VHDL folder. Click **OK** to confirm and exit from the **FPGA Target Properties** panel.
7. Right-click **IO Module** in the **Project Explorer** window and select **Properties**.
8. Select the **Active Technologies: AT-1212** from the IO Module list. The available CLIP items for the AT-1212 are displayed in the **General** category of the Component Level IP panel. If

the information in the **General** category is dimmed, select the **Enable IO Module** checkbox.

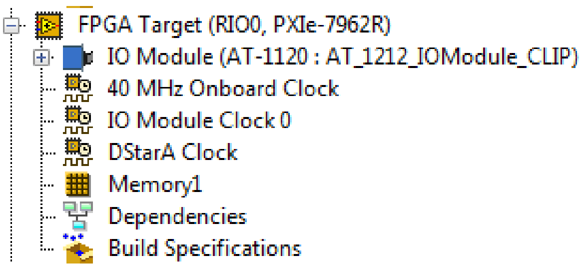
9. Select **Active Technologies: AT-1212** to use the connector-based CLIP.

10. Click on the **Clock Selections** category and select **DStarA Clock** as clockin and **40 MHz Onboard Clock** as clockin40m.



Click **OK**.

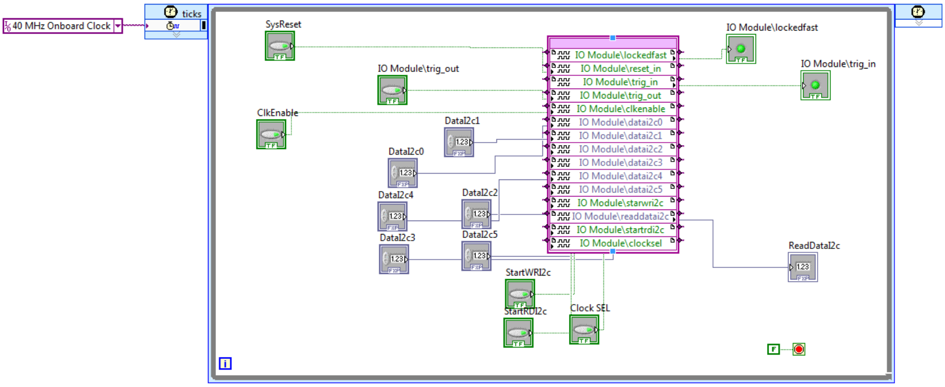
11. Copy the **Memory1** element in the FPGA Target 1212 and paste it in the FPGA Target() you have just created.



12. Right-click the **FPGA Target** and select **New»VI** to create a VI for the FPGA.

13. In the **Project Explorer** window, expand the **IO Module** tree view.

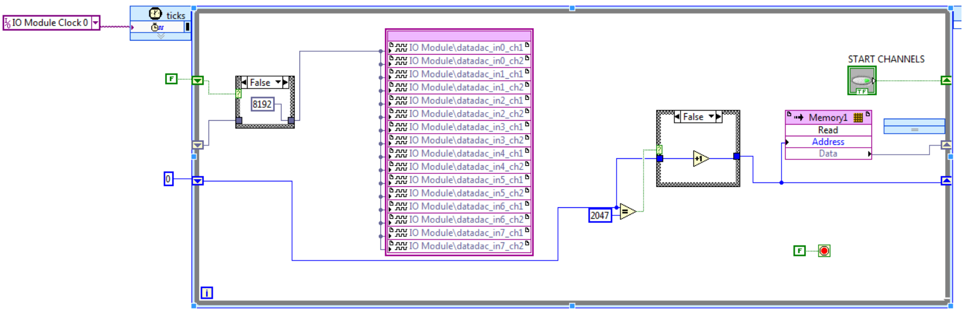
14. Add a Timed Loop as shown in the figure below and right-click to open the LabView palette. Select the **FPGA I/O palette** and add an **I/O node**.



15. Wire controls and indicators from the input/output terminals **datai2c0**, **datai2c1, datai2c2, datai2c3, datai2c4, datai2c5, readdataI2c,startrdi2c, startwri2c, reset_in, trig_in, trig_out, clkenable, clocksel** and **lockedfast .**

16. Wire the 40MHz Onboard Clock to the clock loop.

17. Add another Timed Loop as shown in the figure below and right-click to open the LabView palette. Select the **FPGA I/O palette** and add an **I/O node**.



18. Wire the datadac_in0_ch0..datadac_in7_ch0 and datadac_in0_ch1..datadac_in7_ch1 connectors as in the picture above.

19. Wire the **IO Module Clock 0** to the clock loop.

20. Right-click to open the LabView palette. Select the **Memory & FIFO palette** and add a **Memory Method node.** Right-click on the Memory method, Select **Memory>>Memory1**.

21. Left-click on the Memory1 Memory Method and select the **Read** method. Insert a new shift register in the Timed Loop and wire the connectors like in the picture above.

22. Add the **START CHANNELS** control, insert it into the Timed Loop and wire it to a new shift register. Set true the START CHANNELS control to start the waveforms generation.

23. Add two case structures and wire them as in the picture above. The case on the left controls the data values (8192) that have to be send to the serializers when the module is stopped. The second case checks if the last address of the Memory1 (2047) has been reached. If true, it puts the memory address to the initial value (0).

24. Save the VI.

Running the Host VI

1. Connect one end of an SMA cable to AO 0+ on the front panel of the AT 1212 and the other end to the oscilloscope (50 Ω input).

2. Connect one end of an SMA cable to AO 1+ on the front panel of the AT 1212 and the other end to the oscilloscope (50 Ω input).

3. Tap the unused inputs (AO 0- / A1 -) with a 50Ω load.

4. Open the front panel of **CustomFPGAExample1212(Host).vi**.

5. Click the Run button to run the VI.

6. Wait for module initialization.

7. Click the START CHANNELS button to start the waveform generation.

8. The AT-1212 generates two 2048 points sine waveforms.

Filter Options

If necessary, images and sample clock feed-through can be largely removed using a low-pass filter; Active Technologies suggests the following filters depending on the application and the user needs:

- Mini Circuit RLP-470+
- Mini Circuit SBPL-467 (Maximally Flat Group Delay)
- Mini Circuit SCLF-550
- Mini Circuit SLP-550+
- Mini Circuit VLF-530
- Mini Circuit VLFX-500
- Mini Circuit SBPL-933+ (Maximally Flat Group Delay)

Calibration Service for AT-1212 (OPTIONAL)

- Perform the calibration procedure.
- Store the calibration parameters into the EEPROM module.
- Verification of measurements according to specification.
- Adjustment of performance if found outside specification.

Specifications

Note: Specifications are subject to change without notice.
Please refer to the Active Technologies website at www.activetechnologies.it for the most current specification information regarding your product.

Caution To avoid permanent damage to the AT-1212, disconnect all signals connected to the AT-1212 before powering down the module, and only connect signals after the module has been powered on by the NI FlexRIO FPGA module.

Note All numeric specifications are typical unless otherwise noted.

Typical values describe useful product performance that are not covered by warranty. Typical values cover the expected performance of units over ambient temperature ranges of 23 ±5 °C with an 95% confidence level, based on measurements taken during development or production.

Configuration EEPROM Map

Byte Address	Size (Bytes)	Field Name
0x0	2	Vendor ID
0x2	2	Product ID
0x4	4	Serial Number
0x20	14	Calibration Parameters

Total power, typical operation...............................6W
Weight………………………………………………………………..330g (11.7 oz)
Front panel connectors……………………………………..SMA

Electromagnetic Compatibility

This product meets the requirements of the following EMC standards for electrical equipment for measurement, control, and laboratory use:

- CEI EN 61326-1 (2007-01) - Radiated field strength emission from enclosure port, class A equipment, QPK limit line according to CEI EN 55011 standard corrected for 3m measurement distance.

- CEI EN 61326-1 (2007-01) - Conducted voltage emission from AC mains input port, class A equipment, QPK & AVG limit lines according to CEI EN 55011 standard.

- CEI EN 61000-4-2 – ESD Immunity Test. ±4.0kV contact direct method; ±4.0kV HCP, VCP indirect method.

- CEI EN 61000-4-3 – Radiated RF electro-magnetic field immunity test.
  Level: 10V/m mod. 1kHz AM 80% at 3m of distance - frequency range 80 ÷ 1000MHz.
  Level: 3V/m mod. 1kHz AM 80% at 3m of distance - frequency range 1.4 ÷2.0 GHz.
  Level: 1V/m mod. 1kHz AM 80% at 3m of distance - frequency range 2.0 ÷2.7 GHz.

CE Compliance CE

This product meets the essential requirements of applicable European Directives as follows:

- 2006/95/EC; Low-Voltage Directive (safety)

- 2004/108/EC; Electromagnetic Compatibility Directive (EMC)

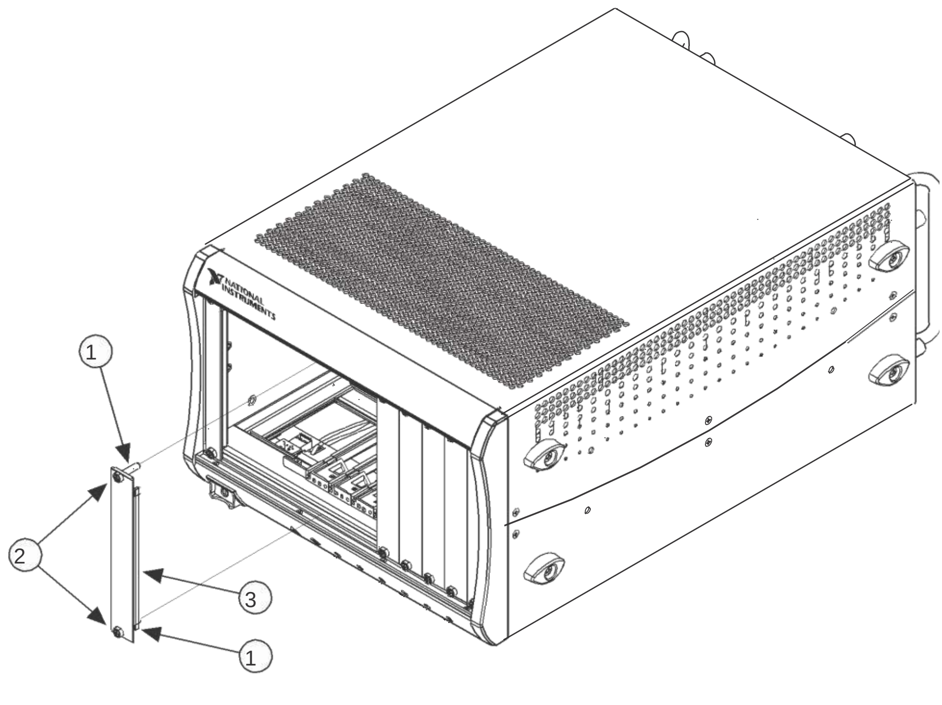
Appendix: Installing EMI Controls

To ensure specified EMC performance, PXI EMC filler panels must be properly installed on your NI system. PXI EMC filler panels (National Instruments part number 778700-01) must be purchased separately. For more installation information, refer to the *NI FlexRIO FPGA Module Installation Guide and Specifications*.

Installing PXI EMC Filler Panels

Complete the following instructions to install PXI EMC filler panels (National Instruments part number 778700-01) in your PXI chassis:

1. Remove the captive screw covers.
2. Install the PXI EMC filler panels by securing the captive mounting screws to the chassis, as shown in Figure 15. Make sure that the EMC gasket is on the right side of the PXI EMC filler panel.



1	Captive Screw Covers	2	Captive Mounting Screws	3	EMC Gasket

Figure 15. PXI EMC Filler Panels and Chassis

Note You must populate all slots with a module or a PXI EMC filler panel to ensure proper module cooling. Do not over tighten screws (2.5 lb · in. maximum). For additional information about the use of PXI EMC filler panels in your PXI system, visit ni.com/info and enter emcpanel