

ARTISAN®
TECHNOLOGY GROUP

Your **definitive** source for quality pre-owned equipment.

Artisan Technology Group
(217) 352-9330 | sales@artisantg.com | artisantg.com

Full-service, independent repair center
with experienced engineers and technicians on staff.

We buy your excess, underutilized, and idle equipment
along with credit for buybacks and trade-ins.

Custom engineering
so your equipment works exactly as you specify.

- Critical and expedited services
- In stock / Ready-to-ship
- Leasing / Rentals / Demos
- ITAR-certified secure asset solutions

Expert team | Trust guarantee | 100% satisfaction

All trademarks, brand names, and brands appearing herein are the property of their respective owners.

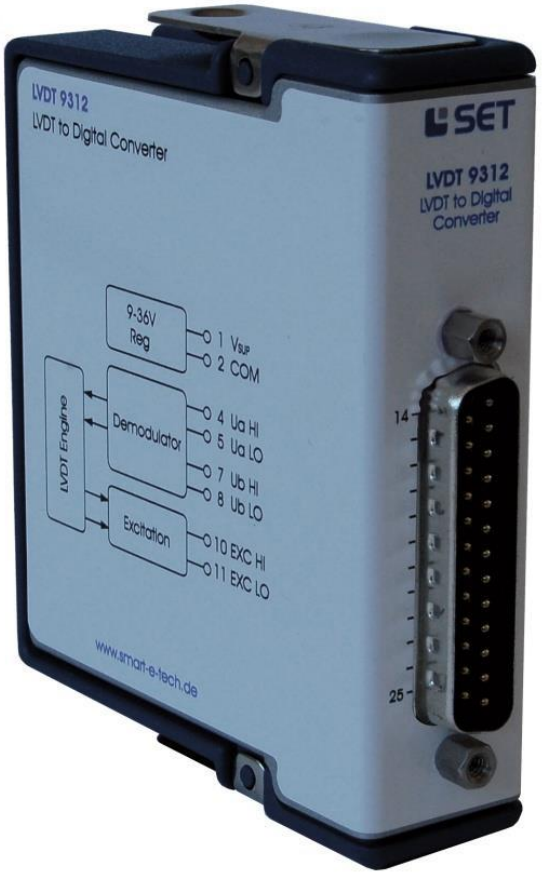
Visit our website - *Click HERE*

TECHNICAL DESCRIPTION

cRIO LVDT 9312 Module

Compact RIO

Art. Nr. 89299



This document is a technical description of the cRIO LVDT 9312 Module

Note Before you begin, complete the Software and Hardware installation procedures applicable to your application.

Note The guidelines in this document are specific to the cRIO LVDT 9312 Module. The other components in the system might not meet the same safety ratings. Refer to the documentation of each component in the system to determine the safety and EMC ratings for the entire system.

MORE INFORMATION ON OUR WEBSITE:

www.smart-e-tech.de/slsc

Copyrights:
SET GmbH
All rights reserved.
Information contained herein is the property of SET GmbH and shall not be duplicated, copied, used or disclosed in whole or in part of any purpose. The right of duplication, use or disclosure is permitted only by written agreement of SET GmbH, August Braun Str. 1, 88239 Wangen/Germany.
LabVIEW, CompactRIO, TestStand are Trademarks of National Instruments

Table Of Contents

1. Important Instructions 4
1.1 Initial Inspection 4
1.2 Safety Instructions 4
1.2.1 Module Failure 4
1.2.2 Impermissible Applications 4
1.2.3 Module Installation And Removal 4
1.2.4 Electrical Connections 4
2. Module Overview 5
3. Getting Started 5
3.1 Software-Requirements 5
3.2 Driver-Installation 6
3.3 LVDT9312 Compact RIO LabVIEW Drivers 7
3.4 NI PCI/PXI FPGA-Card Driver Application 10
3.4.1.1 Creation of a new Project 10
3.4.1.2 Creating a New FPGA-Target 11
3.4.1.3 Adding An R-Series Expansion Chassis 13
3.4.1.4 Adding The LVDT9312 Driver Components To A Project 13
3.4.1.5 Adding LVDT9312 Drivers To The Project Via Drag-And-Drop Function 15
3.4.1.6 Adding LVDT9312 Modules To The Project Via The Discovery-Function 16
3.4.1.7 Adding FPGA Example-Applications 17
3.4.1.8 Compiling And Running The Example Application 19
3.5 Using The Driver Together With A NI RealTime CompactRIO 20
3.5.1.1 Compiling And Running The Example Application 23
4. Connecting The LVDT Demodulator 24
4.1 Running The Application Example 25
5. Application Development 26
5.1.1 LVDT9312 Driver And Communication 26
5.1.1.1 Driver-VI Implementation 26
5.1.1.2 Triggering LVDT9312 Functions Via Functions-VIs 27
5.1.1.3 Initialising And Clearing The FIFO Buffers 28
5.1.1.4 Application To Module Communication Diagram 29
5.2 LVDT9312 Driver-Instructions 31
5.3 LVDT9312 Driver Error Codes 34
5.4 Module Identification Under LabVIEW 34
5.5 Saving The Setup 35
6. Technical Specification 36
6.1 Housing 36

6.2 External Power Supply 36
6.3 Excitation Output 36
6.4 Ua and Ub Signal Inputs 37
6.5 Position Processing 37
6.6 Environmental Conditions 37
6.7 Connector Pinout 38
7. Module Calibration 38
8. Module Maintenance 38
9. Service Address 38

1. Important Instructions

Please read all instructions carefully before the LVDT9312 is installed into a cRIO chassis or connected to other equipment!

1.1 Initial Inspection

Check that the shipment is complete and note whether any damage has occurred during transport. If the contents are incomplete or there is damage, fill a claim with the carrier immediately, and notify the SET GmbH service contact to facilitate the repair or replacement of the module. The address is listed in the back of the manual.
The following parts should be included in the shipment:

- ✓ LVDT9312 cRIO module
- ✓ CD-ROM with driver software, application examples and manual

1.2 Safety Instructions

1.2.1 Module Failure



Do not install the LVDT9312 module into a cRIO chassis when the module is obviously damaged:

- physical damage
- lose parts inside the module

1.2.2 Impermissible Applications



The module is designed for laboratory use. Installing or operating the module in explosive or hazardous environments is not permissible and may result in serious injury or death!

1.2.3 Module Installation And Removal



Hot Surface!
The module may be hot. Touching the module may result in body injury!

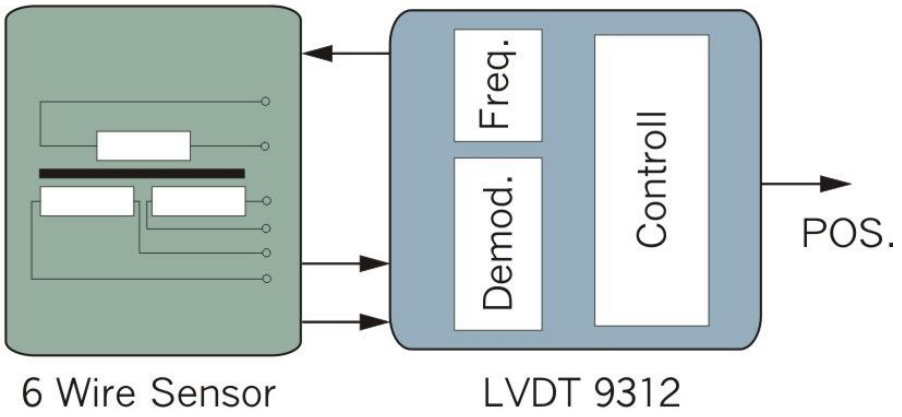
1.2.4 Electrical Connections



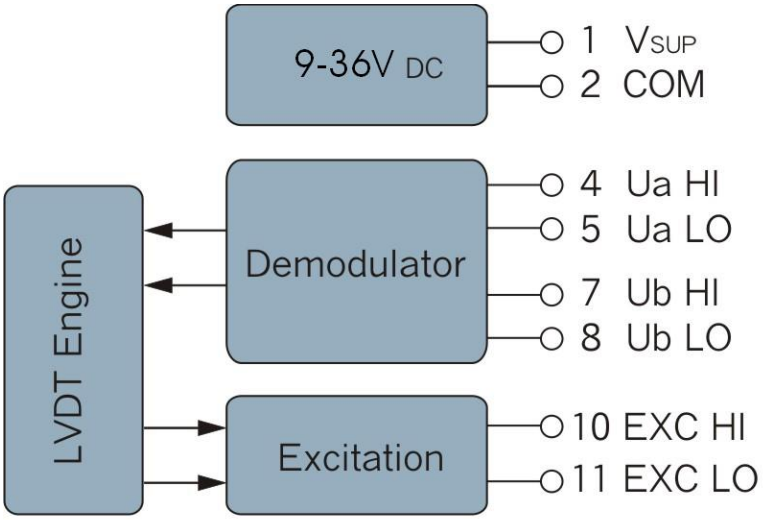
The module is not designed to isolate voltage levels of more than $50V_{DC}$. Do not exceed the voltage levels according to the technical specifications. Not following this instruction may result in module damage.

2. MODULE OVERVIEW

LVDT9312 is a compact, reliable and highly versatile LVDT-to-digital conversion module. Most commonly used LVDTs can easily be used with the module without any additional signal adaption. The module has a built-in excitation oscillator inclusive power stage which can drive most LVDT's directly without the necessity of an external power booster. All module parameter are interactively adjustable via software.



The LVDT9312 accepts a wide voltage supply range from 9V to 36V and provides galvanic isolation between the cRIO and the demodulator interface. The module is useable within the NI cRIO real time environment and can also be plugged into a NI R-series expansion chassis with PCI / PXI FPGA card. We provide all drivers required for the cRIO system and LabVIEW integration examples.



LVDT9312 applications include industrial and military position control systems such as motor control, robotics and many kinds of servo loops which use LVDT.

3. GETTING STARTED

3.1 SOFTWARE-REQUIREMENTS

To use the LVDT FPGA-Driver the following NI Software Components must be installed first:

- LabVIEW version 2012 (12.0) or later
- LabVIEW FPGA Module
- SET C-Series Module Drivers („Setup.exe" included on the SET LVDT Driver-CD)

3.2 DRIVER-INSTALLATION

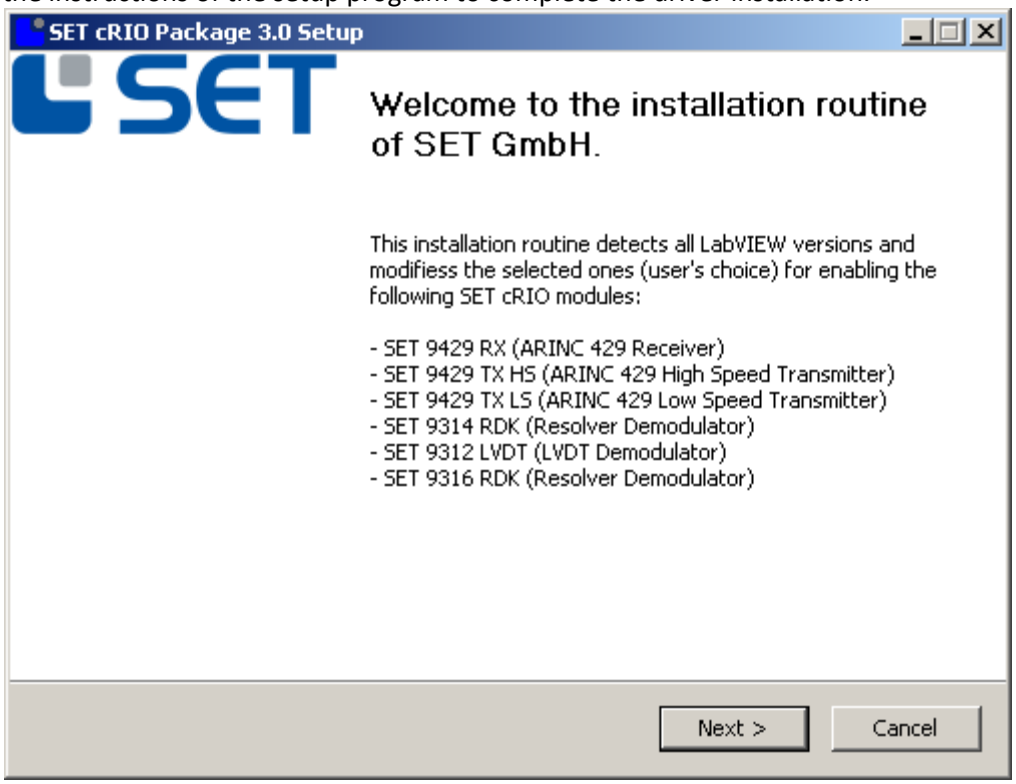
After installation of the NI Software Components, the LVDT driver software must be installed. To do this use the LVDT9312 CD-ROM and follow instructions.

To start the installation process, execute the „setup.exe" program on the CD-ROM.



Setup.exe

Please follow the instructions of the setup program to complete the driver installation.



When the installation is completed successfully, the host system is ready to use LVDT9312 modules in a LabVIEW project.

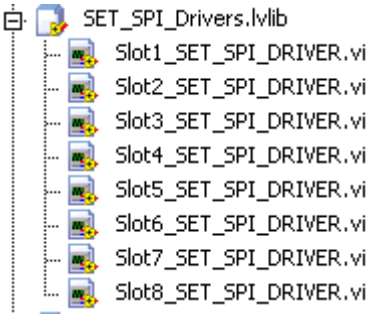
3.3 LVDT9312 COMPACT RIO LABVIEW DRIVERS DEFINITION

The driver application is demonstrated within a LabVIEW example project. To copy the driver components into a new project, use the drag & drop function. (see 3.4.1.4)



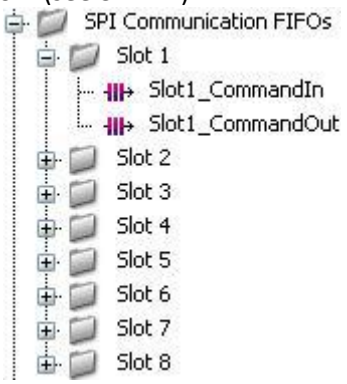
LabVIEW driver components

The LabVIEW FPGA software driver for the LVDT9312 module includes eight driver VIs which control the serial data transfer between the modules and the FPGA backplane. (Refer to the VI-library „SET_SPI_Drivers.lvlib“). For each possible cRIO chassis slot a dedicated driver VI is provided.



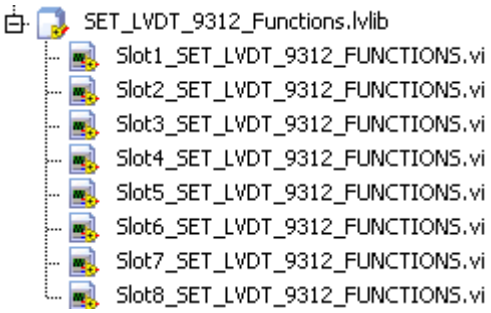
Each module slot applies to its own driver-VI

Each driver VI uses 2 FIFO buffers to communicate with the LabVIEW application. The FIFO buffers are organized in CommandIN-FIFOs and CommandOUT-FIFOs for bidirectional data flow from the application to the module and vice versa. Note that these buffers are pre-defined in the example project in terms of name and size and must not be modified to ensure correct operation. (see 3.4.1.4)



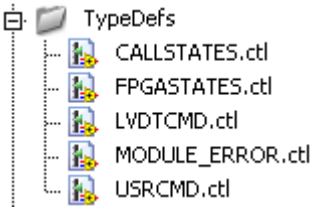
Each driver-VI uses two command FIFOs

The user interface is formed from eight slot specific „Functions-VIs“, which perform the driver calls by use of the communication-FIFOs.
(Refer to the VI-library „SET_LVDT_9312_Functions.lvlib“).
To make the driver VIs and its dedicated FIFOs for the application accessible, a high level function VI is provided. For each possible cRIO slot one function VI. These VIs control the correct data flow and the organisation between the application and the drivers. Again these VIs are predefined and must not be modified to ensure proper operation.



Each slot applies a specific functions-VI

The folder „TypeDefs“ contains data type definitions used within he previous VIs.



LabVIEW driver data type definitions

- USRCMD.ctl

Defines the applicable LVDT9312 commands

- MODULE_ERROR.ctl

Defines the driver error codes

- LVDTCMD.ctl

For driver internal use

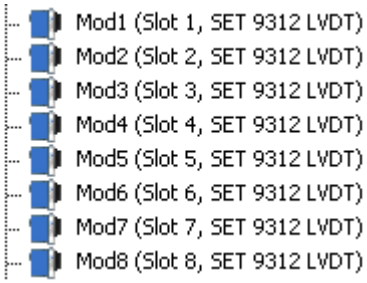
- CALLSTATES.ctl

For driver internal use

- FPGASTATES.ctl

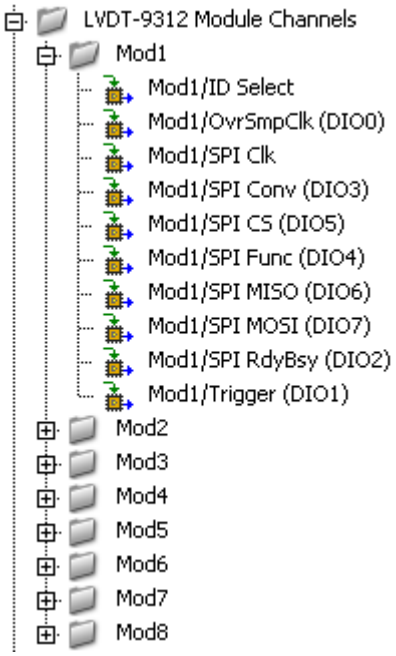
For driver internal use

The example project already covers all module slots with the maximum number of modules, which is four, when an R-Series Expansion Chassis is applied and up to eight modules for a cRIO-Chassis. The names of the modules in the project and the correlating modules I/O channels are pre-defined and must not be changed for correct operation.



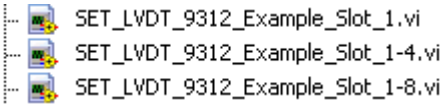
The example project includes all module slots

The example project also defines all required data signals:



Example project data signal definitions

The example project modules can be copied to a new project. Alternatively, when the automatic LabVIEW module detection mechanism “Discover C-Series Modules” is used, attention on identifier assignment should be paid to avoid naming conflicts
Three example files are included to demonstrate the driver-VI application:



Tree example programmes for the R-Series Expansion Chassis and cRIO-Chassis

- SET_LVDT_9312_Example_Slot_1.vi

Program example which demonstrates the application of a single LVDT9312 module connected to slot 1 of a (PCI) FPGA R-Series Expansion-Chassis or a cRIO-Chassis.

- SET_LVDT_9312_Example_Slot_1-4.vi

Program example which demonstrates the application of four LVDT9312 modules connected to a (PCI) FPGA R-Series Expansion-Chassis or a cRIO-Chassis.

- SET_LVDT_9312_Example_Slot_1-8.vi

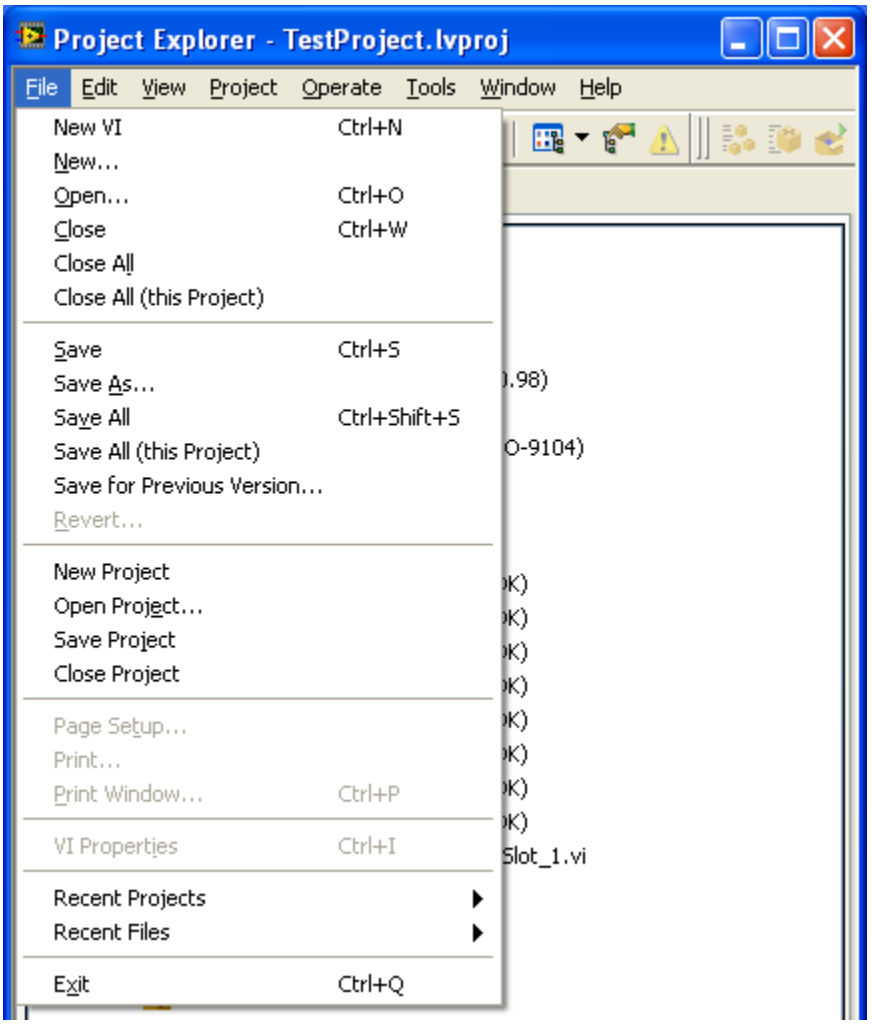
Program example which demonstrates the application of eight LVDT9312 modules connected to a cRIO-Chassis.

3.4 NI PCI/PXI FPGA-CARD DRIVER APPLICATION

The example below demonstrates the driver integration for a NI PCI-FPGA card together with an R-Series Expansion Chassis.

3.4.1.1 CREATION OF A NEW PROJECT

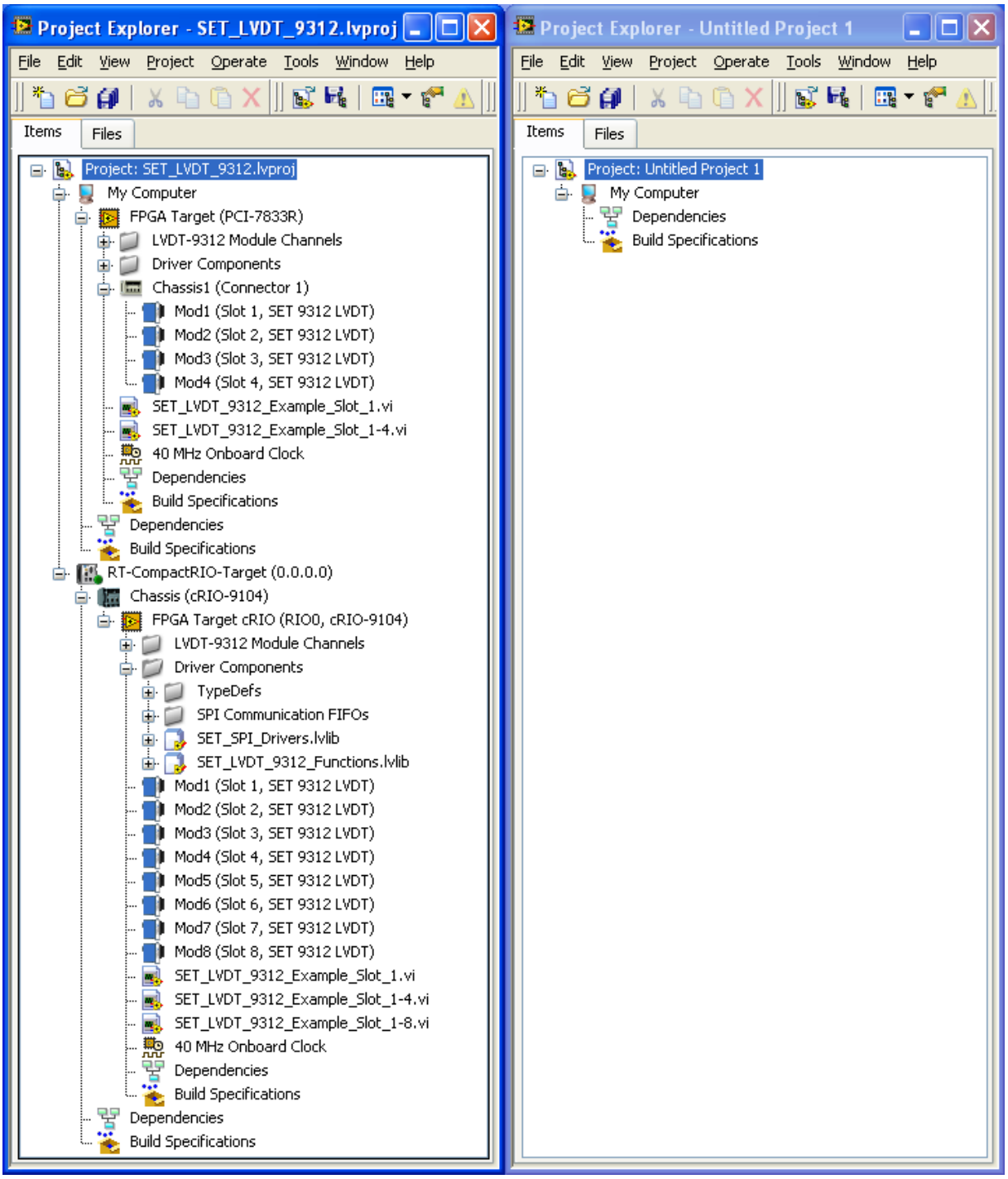
Copy all example project files included in …/LabVIEW on your local disk. We recommend to copy the complete Folder in your LabVIEW project stricter. To integrate the driver into an new LabVIEW project, the target project must be opened subsequently. Alternatively, to start a new project „File → New Project“ must be clicked.



Opening the target project or creating a new project

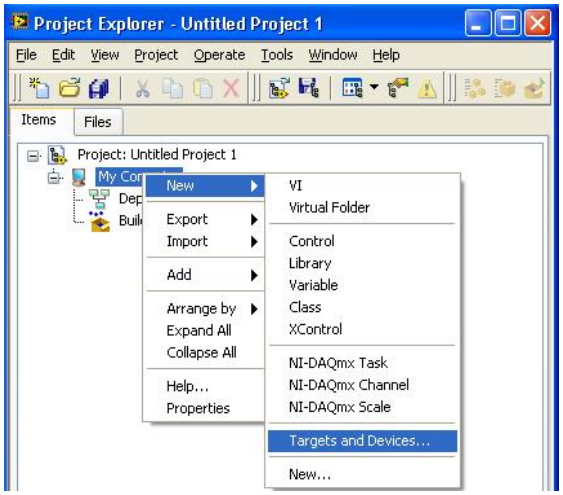
3.4.1.2 CREATING A NEW FPGA-TARGET

Both project windows can be aligned as shown below. The driver components can be copied into a new project via drag-and-drop function Note that the STRG-key must be pressed during drag-and-drop to copy the element. Otherwise the element will be removed from the example project.

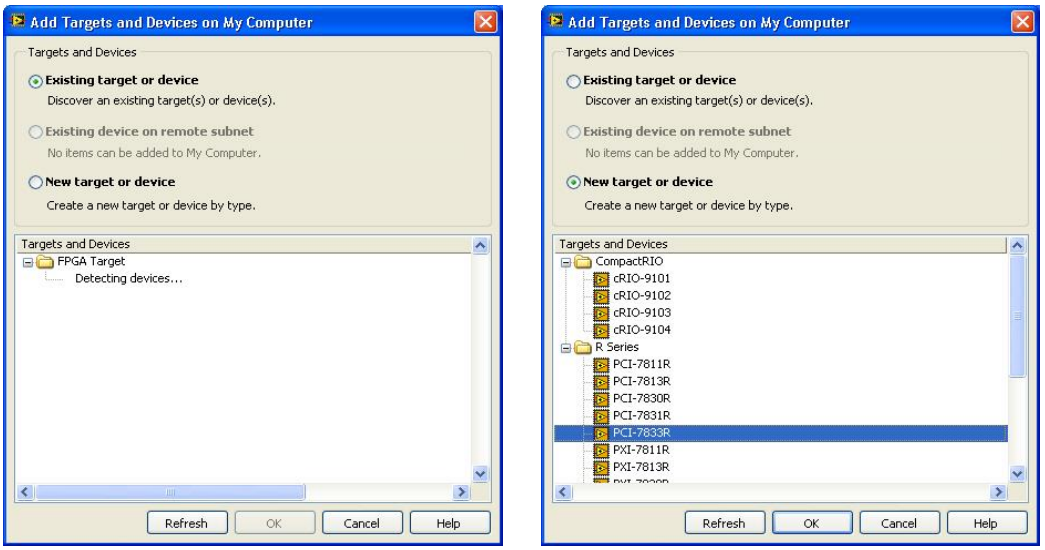


The driver components can be copied by use of the drag-and-drop function

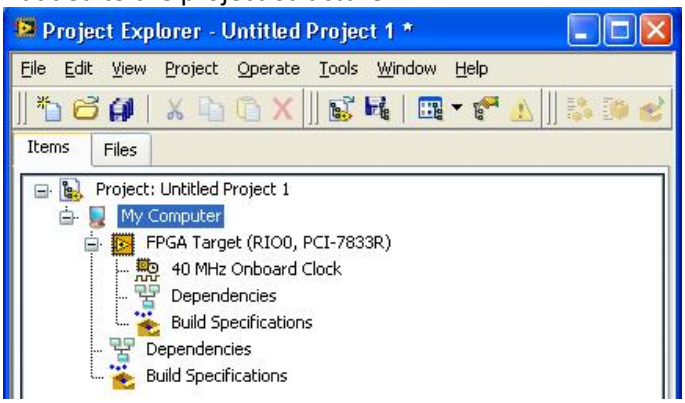
To add a new FPGA-Target use the mouse and click (right button) onto the “my computer” symbol and select: „New → Targets and Devices“.



Select an existing FPGA-Target or define a new device.

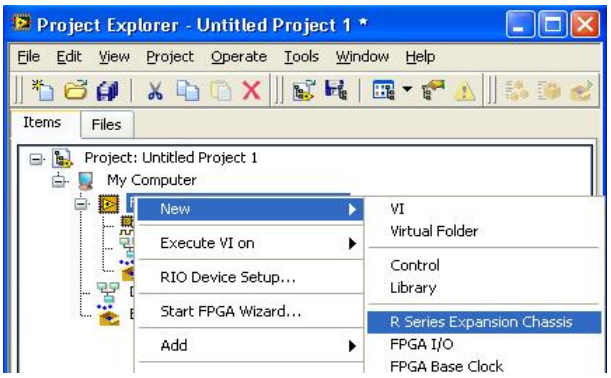


The new FGPA-Target is now added to the project-structure.

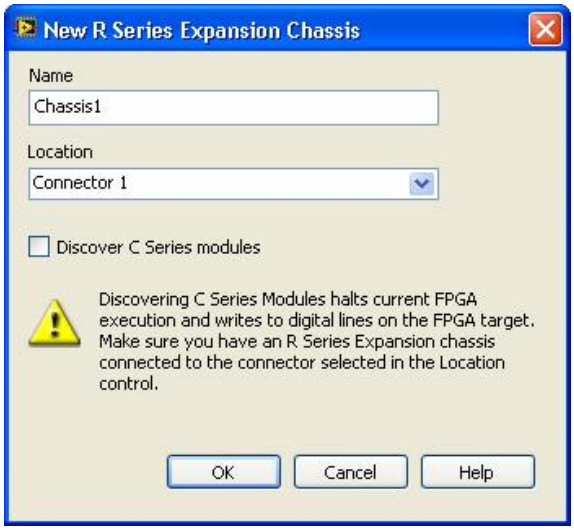


3.4.1.3 ADDING AN R-SERIES EXPANSION CHASSIS

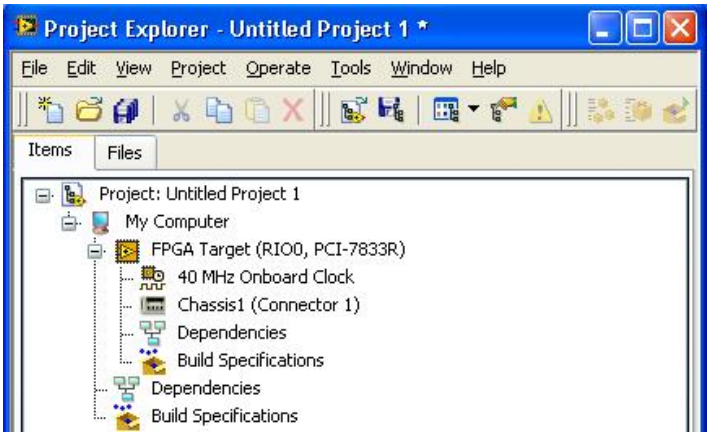
To use an LVDT9312 module together with a FPGA chassis, an „R-Series Expansion Chassis“ must be added to the actual project structure. Use the mouse and click with the right button onto the newly added FPGA-Target and select „New → R Series Expansion Chassis“.



Confirm the dialog with „OK“.

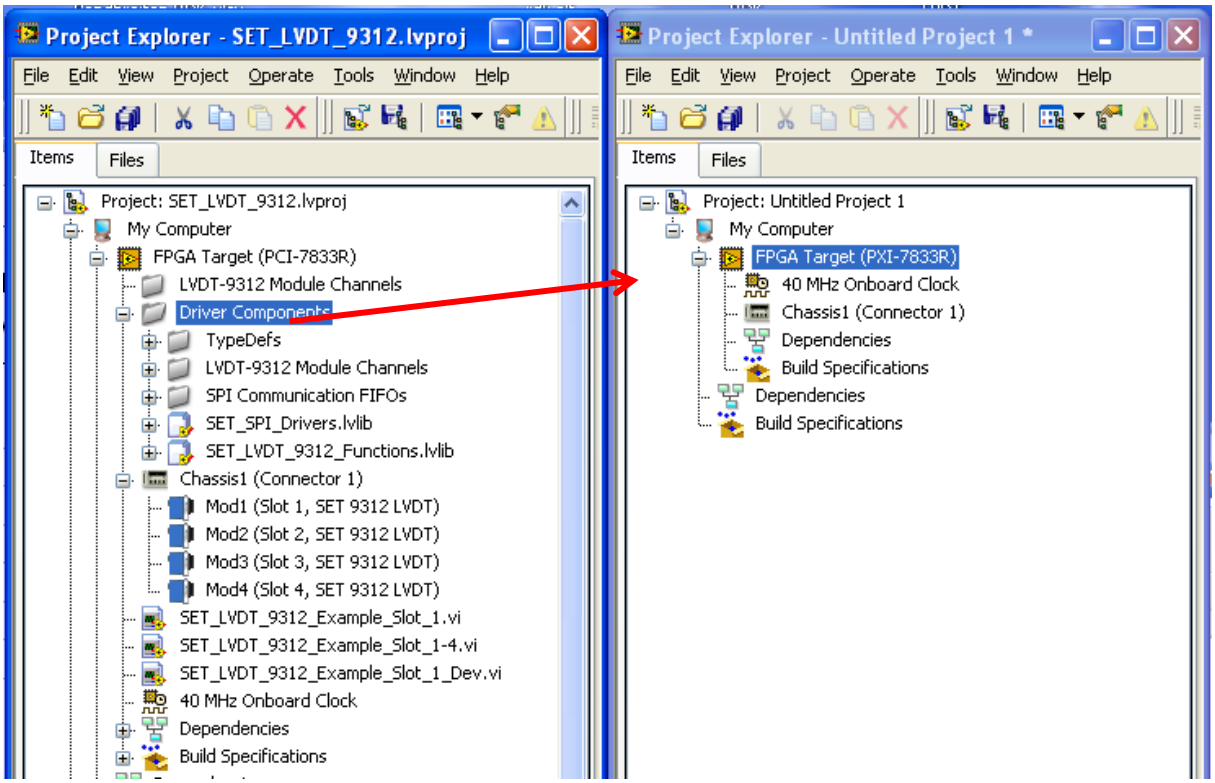


The newly added R-Series Expansion Chassis is now included in the project structure:



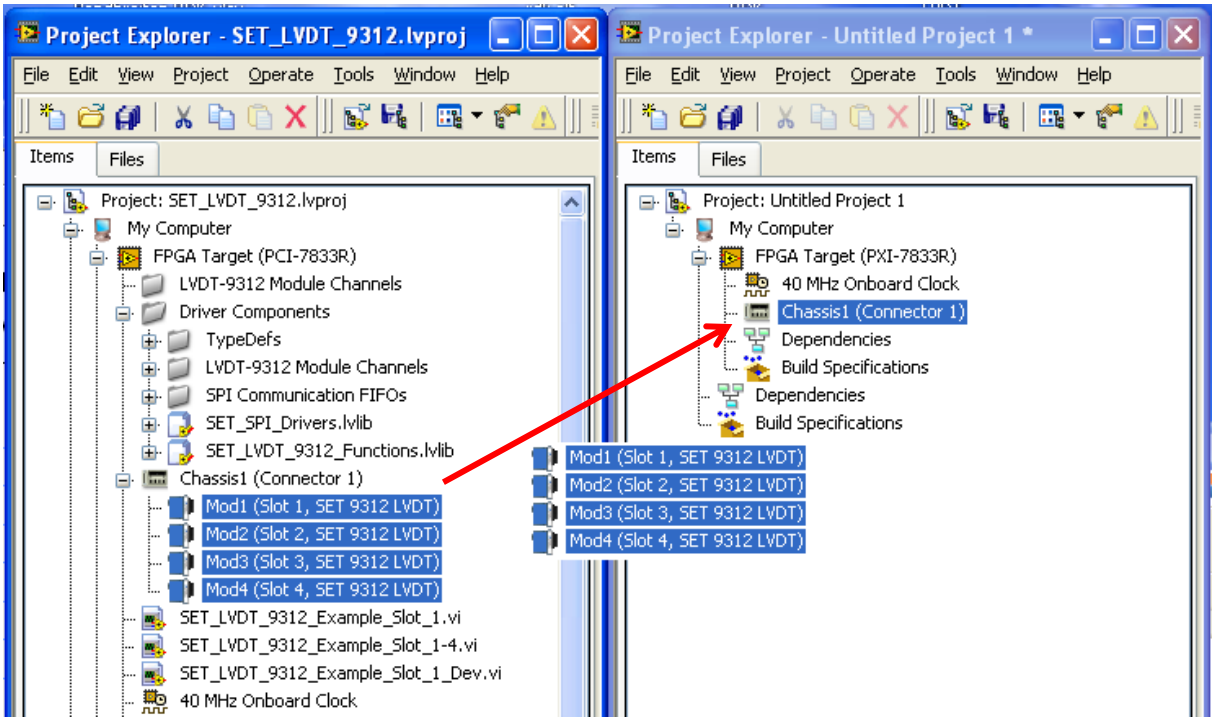
3.4.1.4 ADDING THE LVDT9312 DRIVER COMPONENTS TO A PROJECT

To include the driver components to the new project the files must be copied. Using drag & drop



Copying the Driver Components via Drag'n Drop

3.4.1.5 ADDING LVDT9312 DRIVERS TO THE PROJECT VIA DRAG-AND-DROP FUNCTION



As all driver components for all slots are already defined within the example project they can be copied with the drag-and-drop function. Components which are not used in the new application can be removed from the new project.

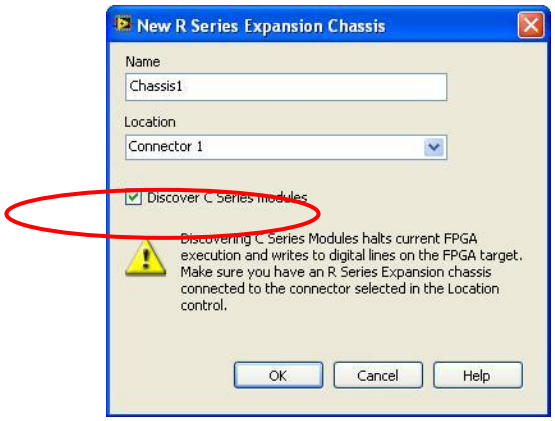
Caution:

It is important to Name the Modules Mod1(...) and not Chassis1/Mod1(...) like the default naming of the "Discovery Function" suggests.

Do not erase the contents of the libraries "SET_SPI_Drivers.lvlib" and "SET_LVDT_9312_Functions.lvlib".

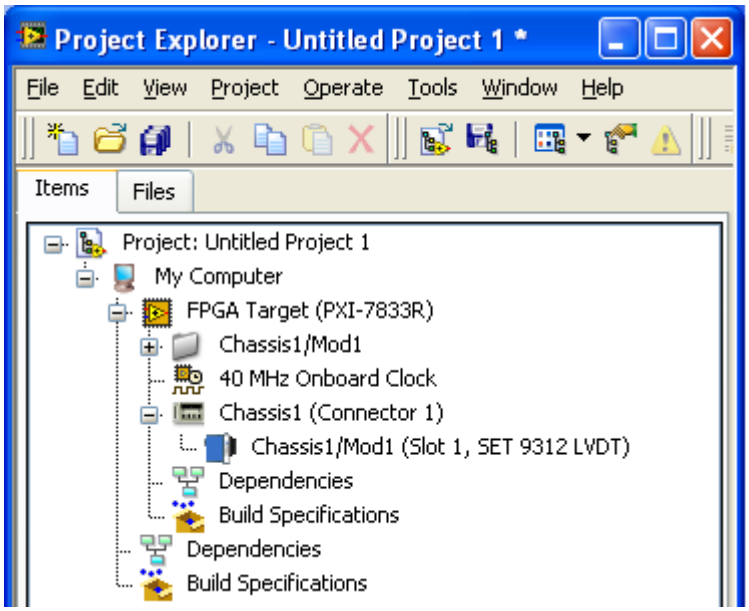
3.4.1.6 ADDING LVDT9312 MODULES TO THE PROJECT VIA THE DISCOVERY-FUNCTION (NOT RECOMMENDED)

The LabVIEW function “Discover C-Series Modules” automatically detects and integrates cRIO modules which are plugged into an R-Series Expansion Chassis or a cRIO-Chassis.



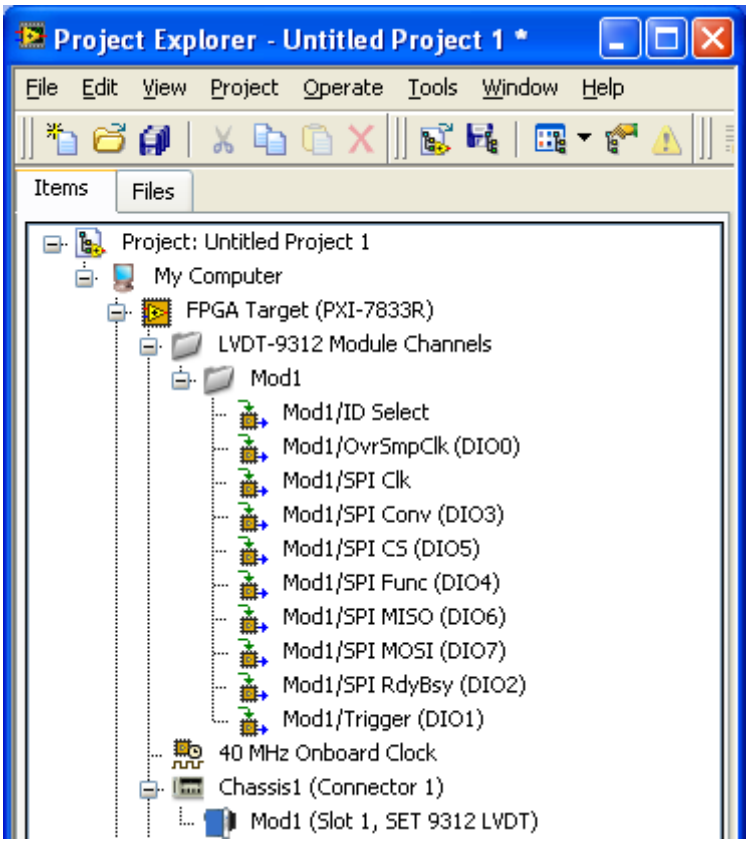
This initiates LabVIEW to check the chassis for cRIO modules.

When the search procedure is complete, LabVIEW adds an module item for each detected module and a virtual folder including the module IOs.



Automatik Detektion, with wrong name for Module

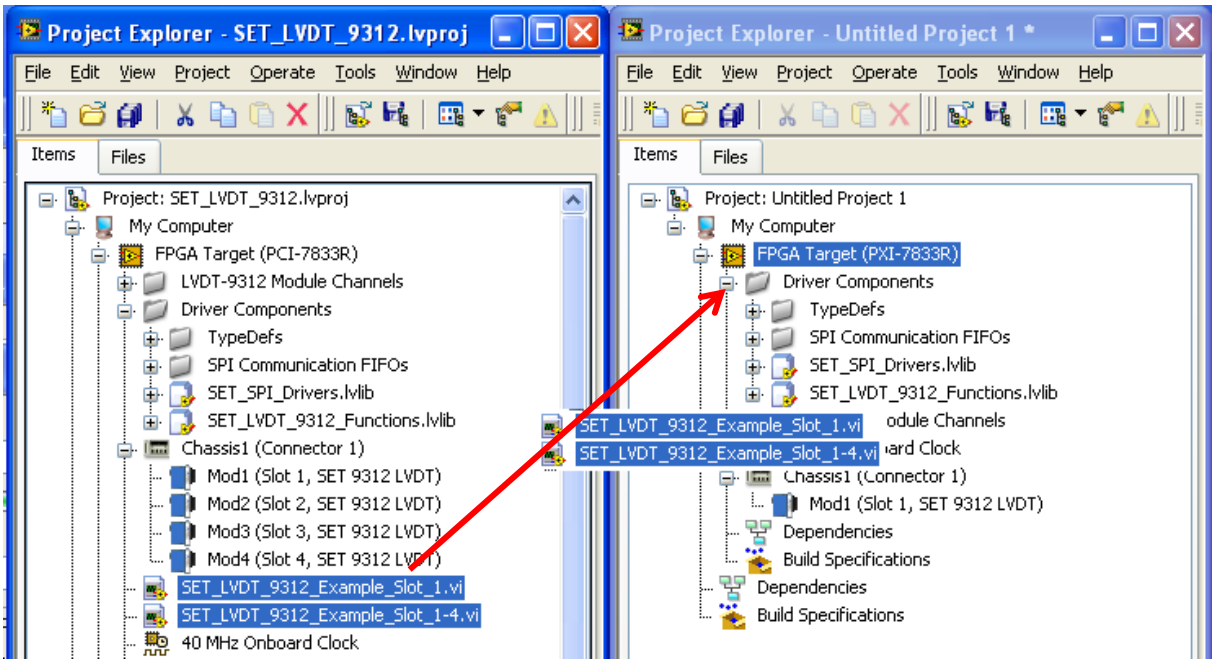
It is important that the I/O names of the Module match the names expected by the driver. Normally the correct I/O name are created by LabVIEW within the discovery function. However on some LabVIEW versions the I/O Names are different. In this case you need to change the names of the I/O’s to the names used in the example project, or copy the I/Os and module from the example project and delete the automatically created ones.



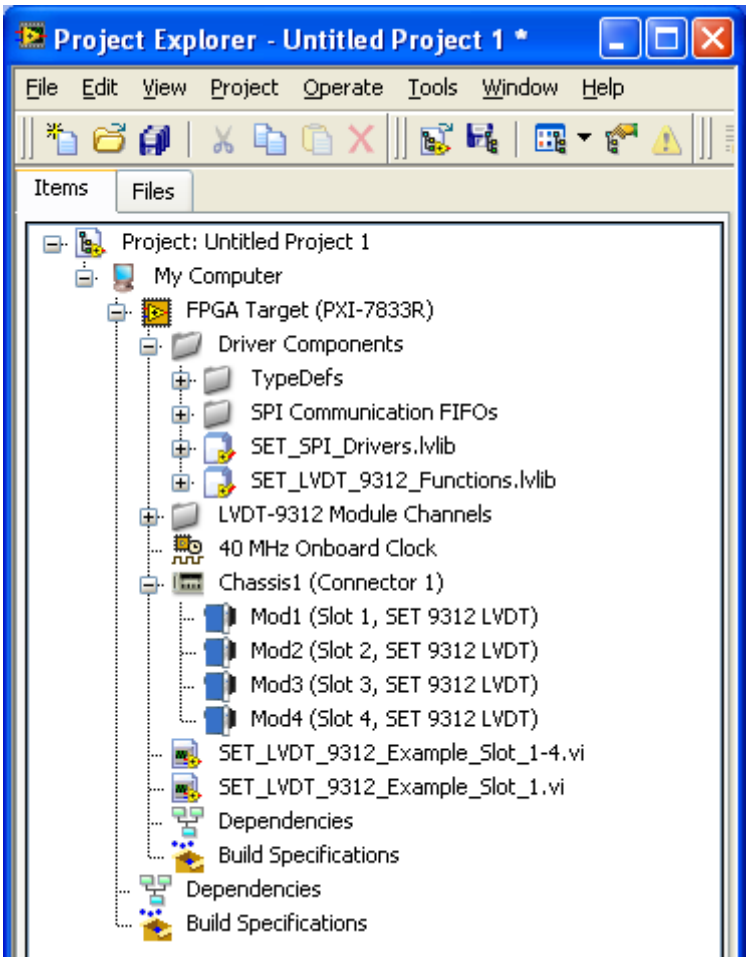
Renamed Module with copied Module Channels IOs

3.4.1.7 ADDING FPGA EXAMPLE-APPLICATIONS

To illustrate the driver use, small example programs are provided with the module on the CD. Use the copy & paste function to copy these examples into the new application.

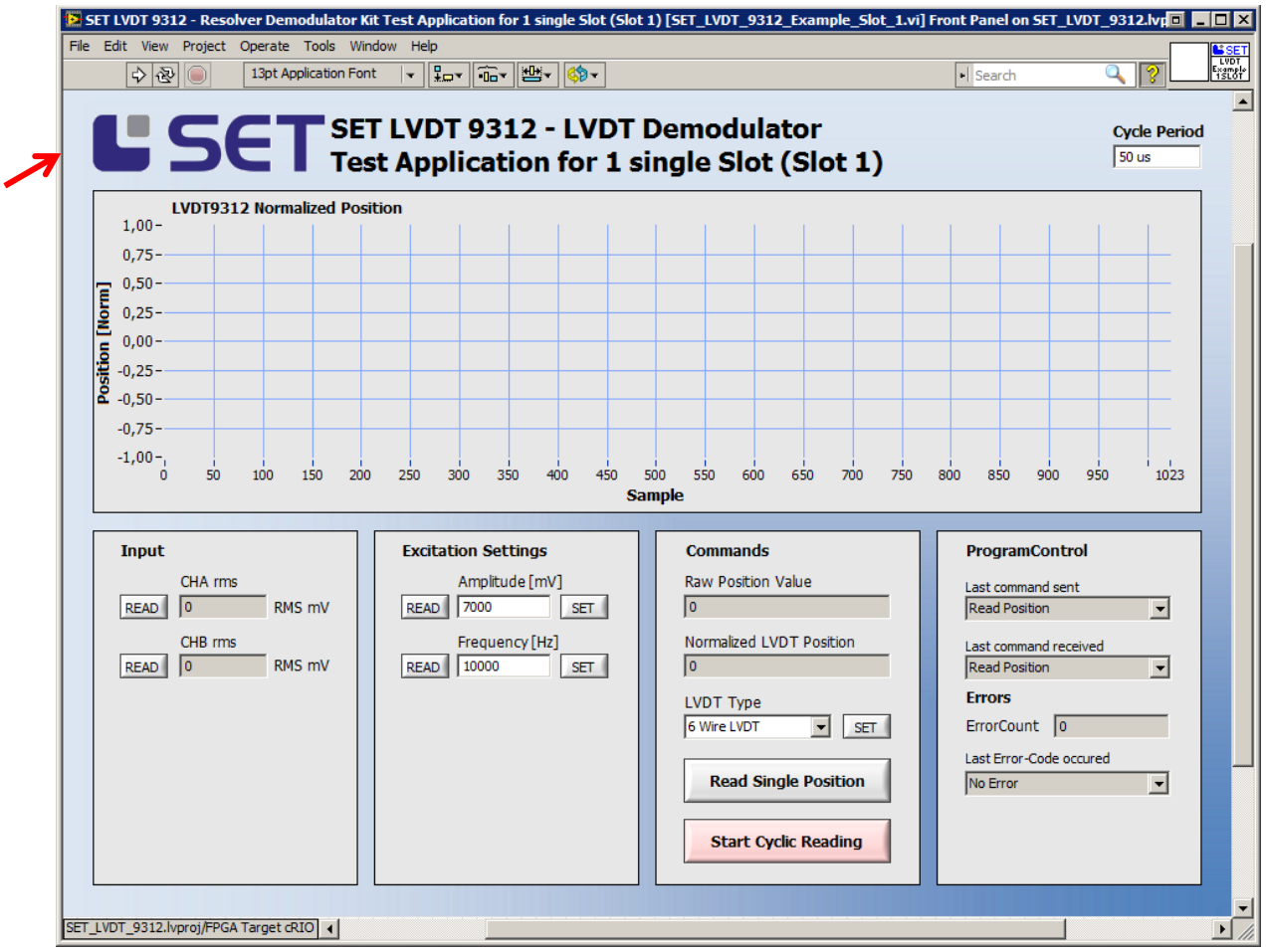


All driver components and the example application are now added to the new project.



3.4.1.8 Compiling And Running The Example Application

Open the „SET_LVDT_9312_Example_Slot_1.vi“ file within the project structure to start the slot 1 single module application example. Then click the run button.



Now the compilation process starts and may take up to 60 minutes. Note that the number of modules used in the application has an impact on the compilation time as every module uses separate driver-VIs, function-VIs and FIFOs. Compilation time depends on the used PC Hard- and Software.

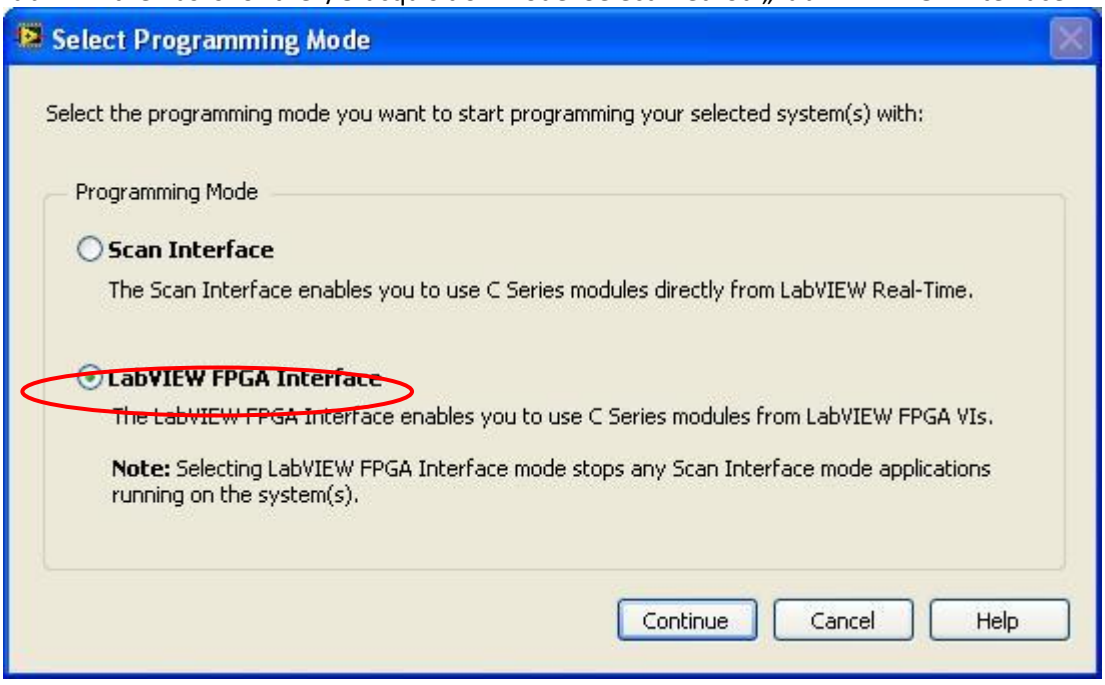
3.5 USING THE DRIVER TOGETHER WITH A NI REALTIME COMPACTRIO

The procedure illustrated before (see 3.4) refers to a PCI-/PXI FPGA-Target, but all driver components are fully compatible with a cRIO target. The only exception is the eight-slot example application which is only operable on an eight-slot cRIO chassis.

cRIO-Chassis (with FPGA-Target):

When the cRIO chassis and FPGA target does not exist in the new application, it must be installed as shown in chapter 3.4.1.2: select „New → Targets and Devices“ device type „cRIO Realtime“ by clicking the “+” button. LabVIEW now detects the connected cRIO chassis. On detection of the chassis it must be selected and “OK” must be checked.

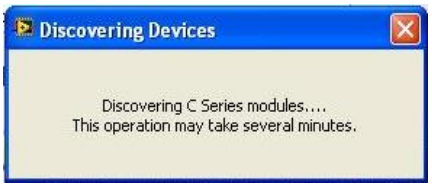
LabVIEW then asks for the I/O acquisition mode. Select method „LabVIEW FPGA Interface“.



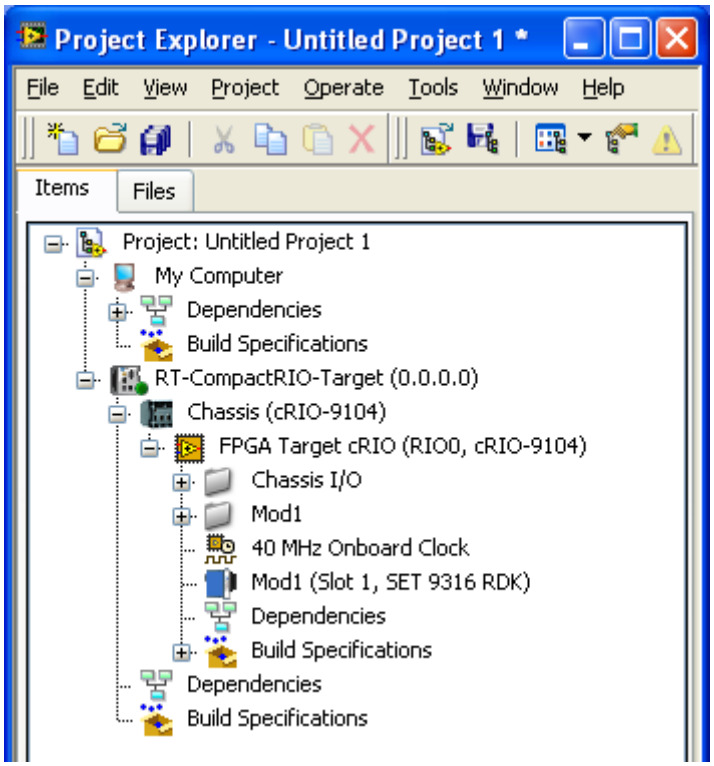
When a new chassis is added, LabVIEW automatically triggers the „Discover C-Series Modules“ procedure.



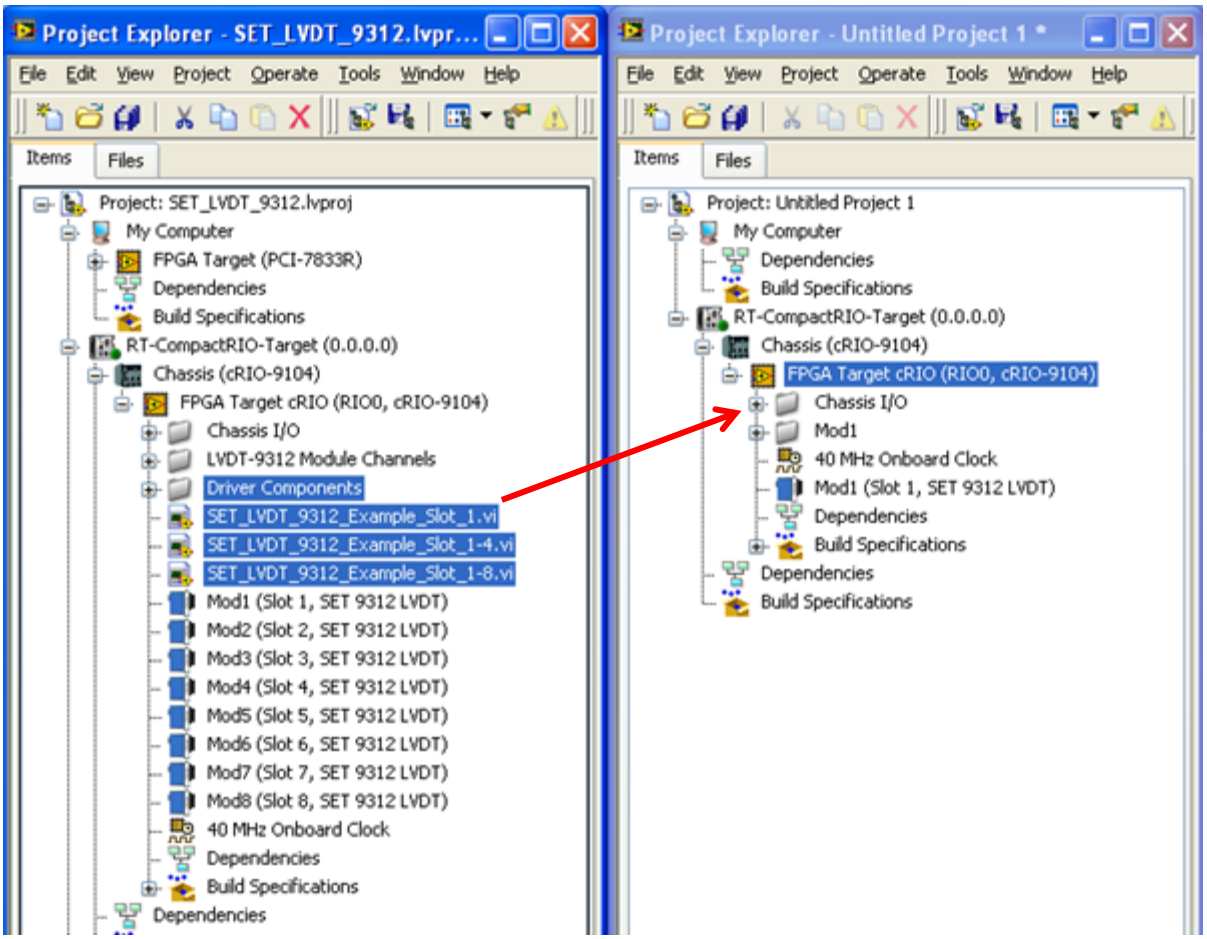
LabVIEW now detects newly installed cRIO modules.
The external supply is not required for module detection.



On completion of the detection process, LabVIEW adds the detected modules to the project structure as illustrated below. Additionally a virtual I/O-folder for every new module is automatically installed within the project.

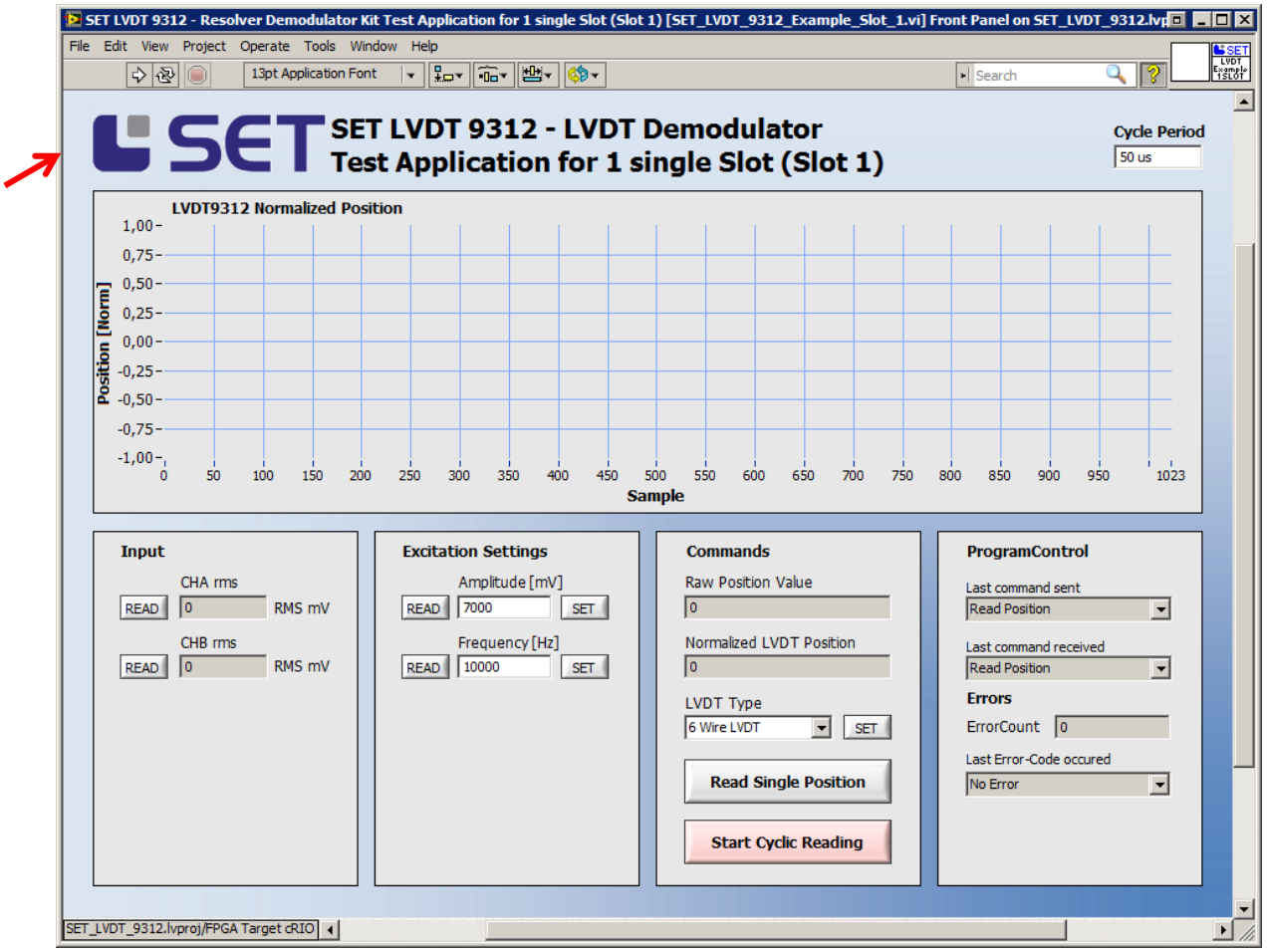


Now the virtual folder „Driver Components“ must be copied from the example project to the new FPGA target within the LabVIEW project.



3.5.1.1 COMPILING AND RUNNING THE EXAMPLE APPLICATION

Open the „SET_LVDT_9312_Example_Slot_1.vi“ file within the project structure to start the slot 1 single module application example. Then click the run button.
Note: This example is designed for an LVDT9312 Module in Slot 1. However it is easy to alter the example to work for a different slot.

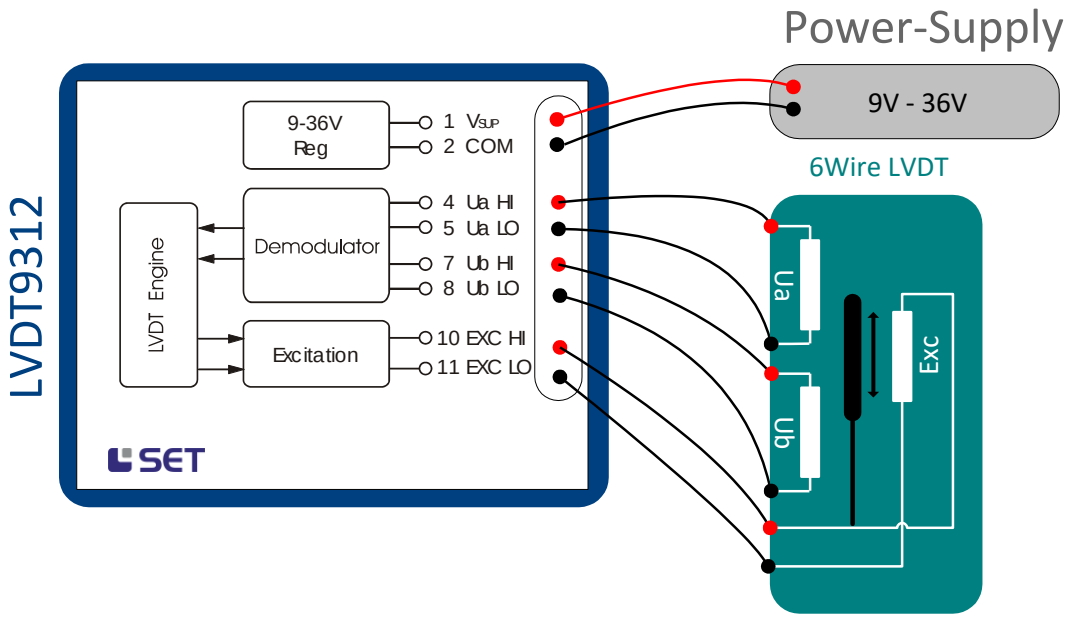


Now the compilation process starts and may take up to 60 minutes. Note that the number of modules used in the application has an impact on the compilation time as each module uses separate driver-VIs, function-VIs and FIFOs.

4. CONNECTING THE LVDT DEMODULATOR

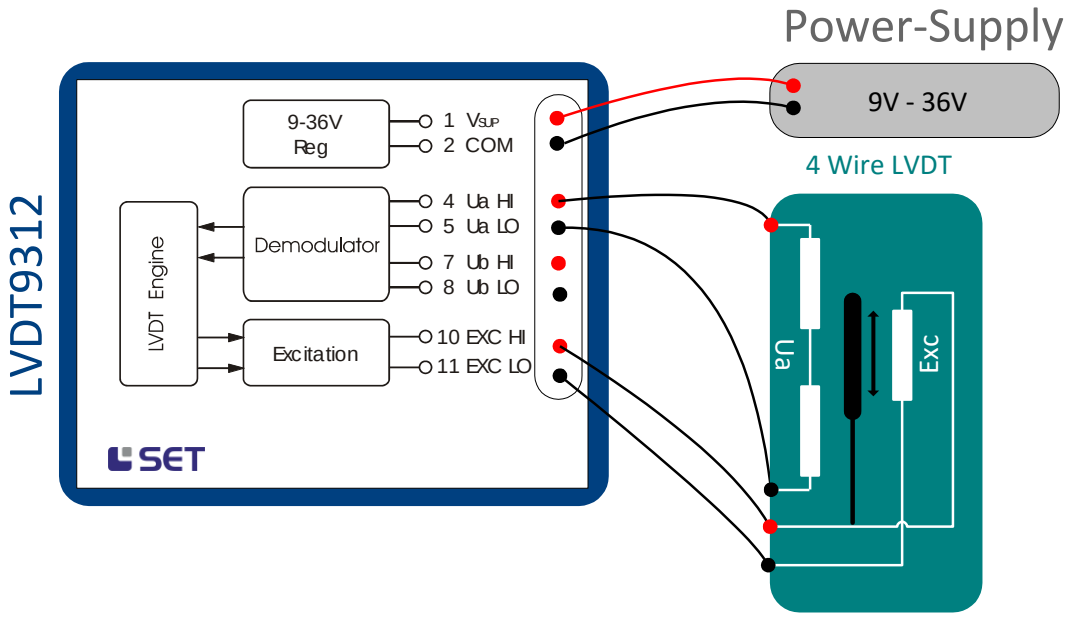
To make the LVDT9312 ready for use plug the module into the correct slot on the cRIO Chassis. Make sure to select the correct slot according to the LabVIEW project definition. Then connect the module external power supply ($9V_{DC}$ – $36V_{DC}$) as shown below. Connect the LVDT interface as shown below (see also paragraph 6.7 Connector Pinout).

Connecting a 6 Wire LVDT:



Connection a 6 Wire LVDT

Connecting a 4 Wire LVDT:



Connecting a 4 Wire LVDT

4.1 RUNNING THE APPLICATION EXAMPLE

Make sure that:

- ✓ Application compiling process is complete
- ✓ The module is plugged into the correct slot according to the application
- ✓ The Resolver and the external supply is connected

The application example can now be started by clicking the “Run”-Button in LabVIEW.
Use parameters *Amplitude* [mV] *and Frequency* [Hz] to adjust the excitation signal to their correct levels.

After that specify the LVDT Type for the demodulation Stage.
The transfer Function for 6 Wire LVDT mode is $Pos = \frac{Ua-Ub}{Ua+Ub}$
The transfer Function for 4Wire LVDT mode is $= \frac{Ua}{exc}$,
(the sign in 4W mode is determined by a separate phase detection).

Note: All settings are stored in non-volatile memory within the LVDT module and restored after each power cycle.

5. APPLICATION DEVELOPMENT

5.1.1 LVDT9312 DRIVER AND COMMUNICATION

The LVDT driver package includes eight driver-VIs, eight functions-VIs and 16 communication-FIFOs which are used internally. The driver- and function-VIs are located in two different libraries, „SET_LVDT_9312_Drivers.lvlib“ and „SET_LVDT_9312_Functions.lvlib”. All driver components are located in the project structure of the shipped example projects (folder “Driver Components”).

LVDT9312 Driver-VI’s:

The driver VI controls the communication between the FPGA and the LVDT9312 modules and operates as server. This VI waits for an application call. On reception of an instruction The VI communicates with the module and returns the response data to the calling process.

LVDT9312 Functions-VI’s:

The Functions-VI provide the user interface for the application and transfer instructions and data to the driver VI. These VIs act as clients for the transfer instructions and data to the driver-VIs.

5.1.1.1 DRIVER-VI IMPLEMENTATION

The implementation of the LVDT9312 drivers into a LabVIEW application takes place by simply locating the relevant VI’s on the main programs block diagram. Each chassis slot uses its own driver-VI.



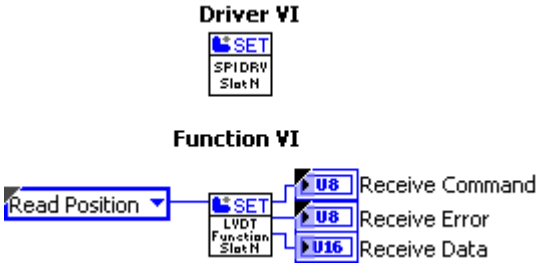
When the LabVIEW application is started, all involved driver VIs are also processed. The driver VIs act as server and wait for instructions from the application.

Important notice: The driver VIs run cautiously. Therefore the Drivers may block the application. We recommend placing them separately on the block diagram[1]. When the driver-VI’s are used inside a data flow or a sequence they will block the program.

[1] Not added tot he data flow

5.1.1.2 TRIGGERING LVDT9312 FUNCTIONS VIA FUNCTIONS-VIS

The Functions-VI's are located in the „SET_LVDT_9312_Functions.lvlib". Note that each chassis slot requires a specific functions-VI. (e.g. slot 1, slot 2 …)



Calling a LVDT9312 function via the Functions-VI

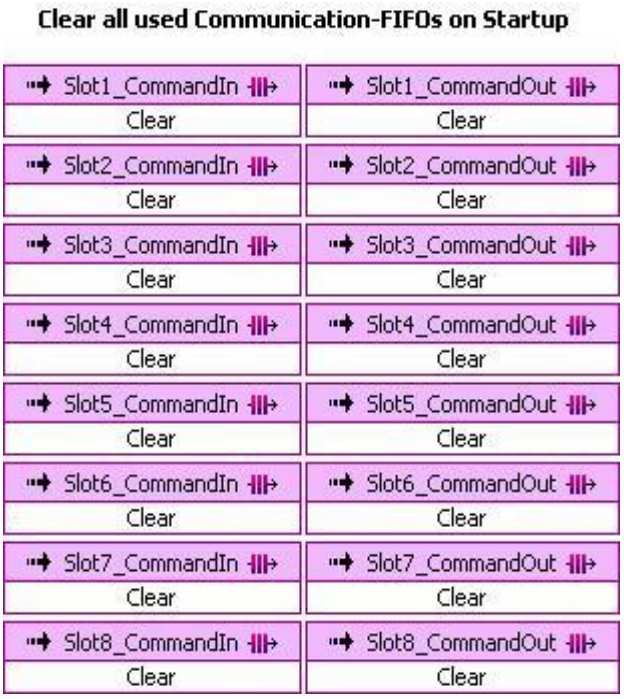
When a Function-VI is executed, instructions and data are transferred to the corresponding FIFOs „SlotN_CommandIN". The applicable driver-VI then communicates with the LVDT9312 module via the driver internal FIFO „SlotN_CommandOUT".



Driver internal FIFOs for data transfer between Function-VI and Driver-VI (Slot1 example)

5.1.1.3 INITIALISING AND CLEARING THE FIFO BUFFERS

To ensure a correct data transfer it is essential to clear the FIFO contents during the programs initialisation sequence. This is done by executing the “Clear” function after program start.
Important notice: Execute the “Clear” instruction before any other VI call.
The example below illustrates the FIFO buffer clearing of an application which uses eight modules:

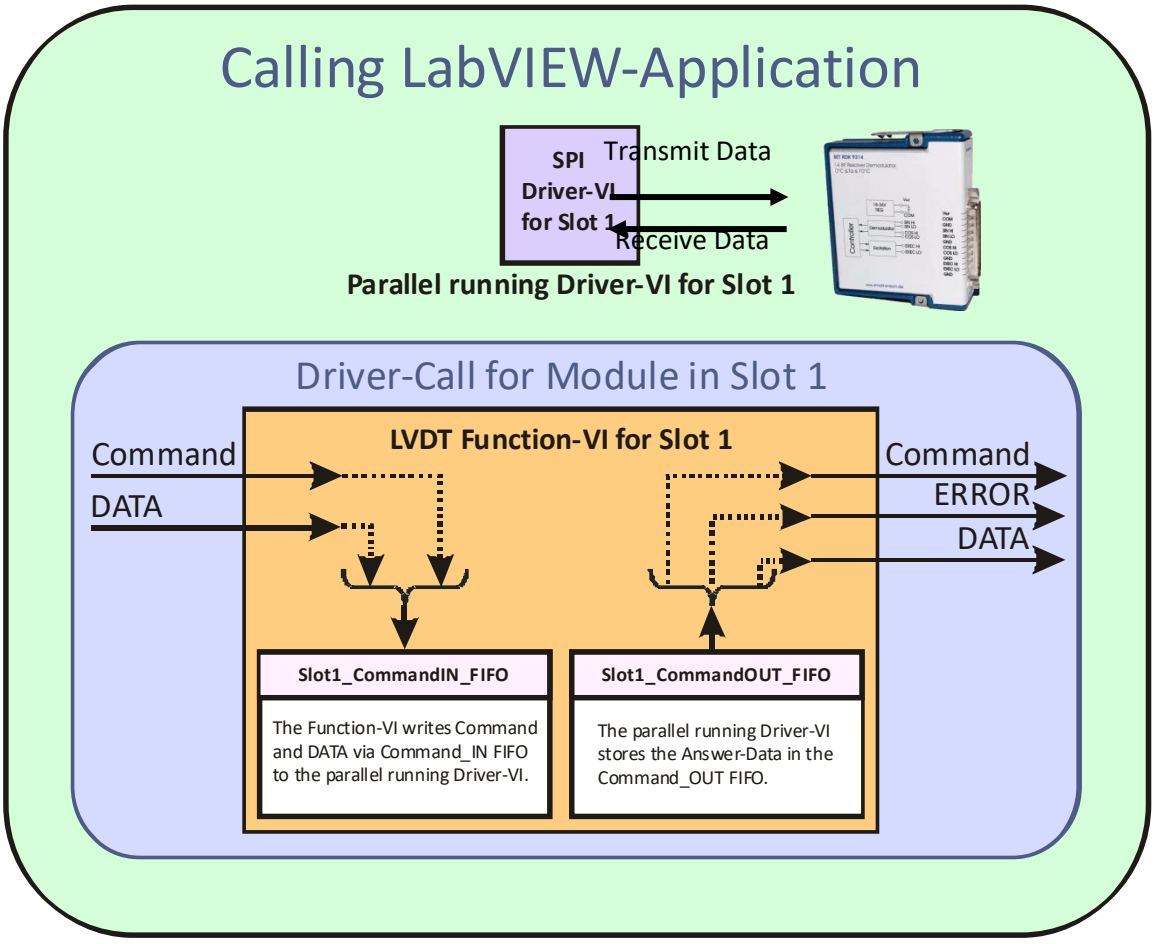


Clearing all (used) communication FIFOs during program start

5.1.1.4 APPLICATION TO MODULE COMMUNICATION DIAGRAM

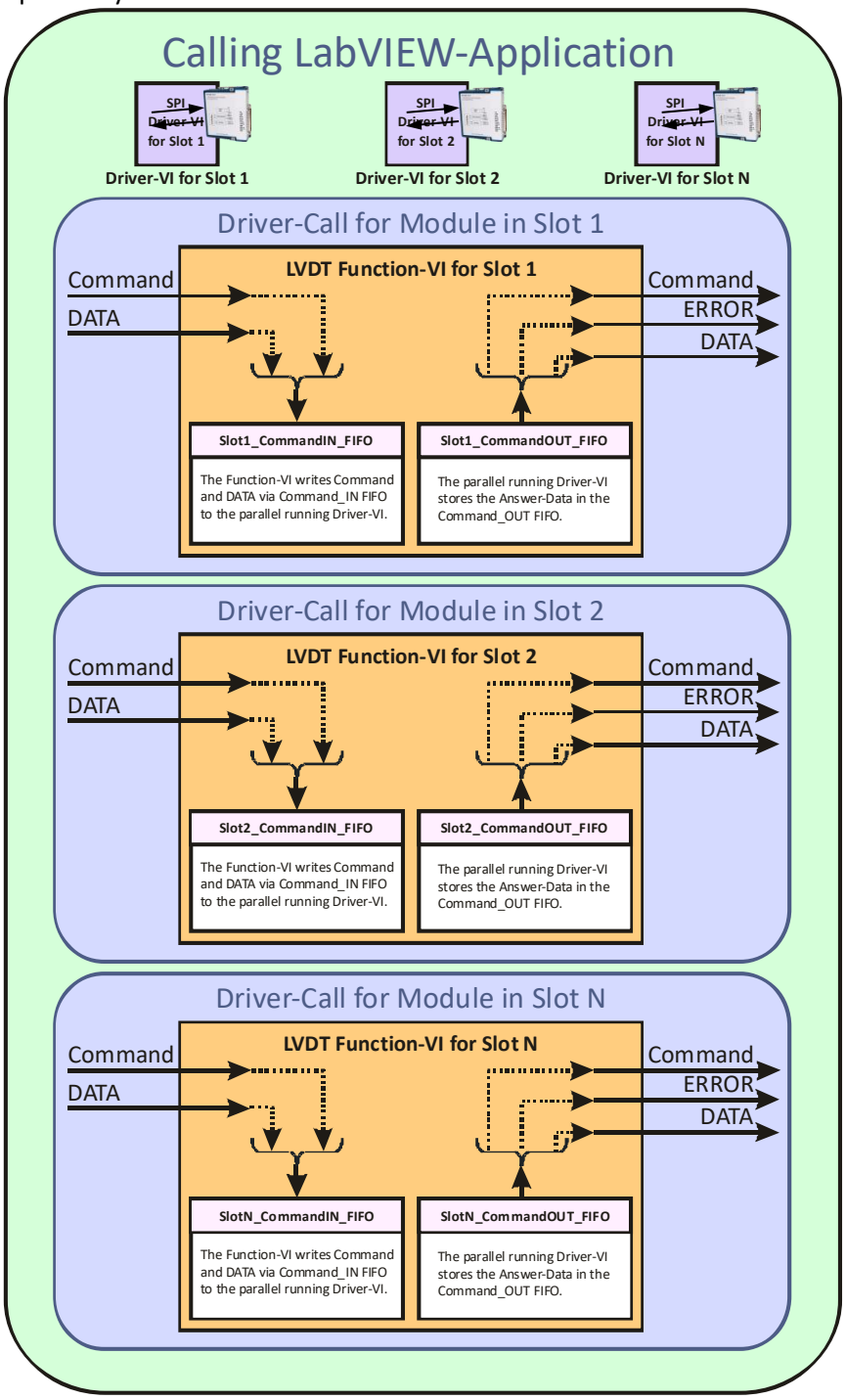
Communication with a single LVDT9312 module:

The diagram below illustrates the communication scheme between application, Function-VI, Driver-VI and LVDT9312 module (slot 1). From the diagram it can be noticed that the Driver-VI of the FPGA driver (slot 1) is located outside and parallel to the code section which triggers the instructions. However, the Function-VI may be used within any LabVIEW structure like loops, cases or sequences. Furthermore, multiple calls are allowed which are operated sequentially. Note that the Function-VIs use non-re-entrant structures.



Communication between application and LVDT9312 module

Communication with multiple LVDT9312 modules:
The diagram below illustrates the communication scheme between application, Function-VI, Driver-VI and multiple LVDT9312 modules. From the diagram it can be noticed that the Driver-VIs of the FPGA drivers are located outside and parallel to the code section which triggers the instructions. However, the Function-VI may be used within any LabVIEW structure like loops, cases or sequences. Furthermore, multiple calls are allowed which then are operated sequentially. Note that the Function-VIs use non-re-entrant structures.



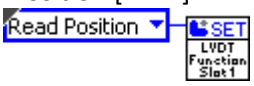
Communication between application and multiple LVDT9312 modules

5.2 LVDT9312 DRIVER-INSTRUCTIONS

The LVDT9312 driver instructions are defined within the type definition file „USRCMD.ctl“. The instruction names correlate with an identification number which is given in brackets (), below.

Command execution requires a specific execution time and varies due to asynchronous processing principles. Hence a timeout should be defined for an instruction call which by default is 40.000.000 system ticks (equal to 1 second for a 40MHz FPGA). When no timeout is connected to the input terminal of the Function-VI, the default timeout value is active. To specify a timeout other than the default value, the input must be connected to a new timeout value. Note that specifying a timeout less than 40.000.000 has no effect because the default value cannot be reduced. If an instruction takes longer than the specified timeout, a fault is generated from the driver (refer to the drivers fault codes).

Instruction [U8]:	Tx-Data [U8]:	Receive-Data [I16]:	Data Sink:
Read Position (1)	no data	LVDT Position [RAW]	LVDT Controller

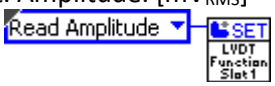


This instruction reads the actual LVDT position. The data scaling is binary data within the range of -32768 to 32767 equal to -1 to 1. It takes 24us (+/-1us) to receive the position data.
The maximum position sampling rate for cyclic reading is 40kHz. To calculate the resolver position in engineering units use the following equation:

$$Position = DATA \cdot \frac{1}{2^{15}}$$

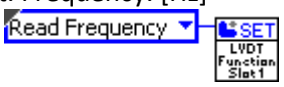
Note that dividing by 2^{15} equals a logical shift of 15.

Instruction [U8]:	Tx-Data [U8]:	Receive-Data [U16]:	Data Sink:
Read Amplitude (3)	no data	Excit. Amplitude. [mV_{RMS}]	LVDT Controller



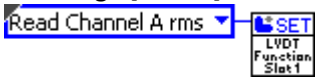
This instruction reads the programmed amplitude of the excitation voltage. The return data format is a 16bit engineering value in [mV_{RMS}].

Instruction [U8]:	Tx-Data [U8]:	Receive-Data [U16]:	Data Sink:
Read Frequency (4)	no data	Excit. Frequency. [Hz]	LVDT Controller



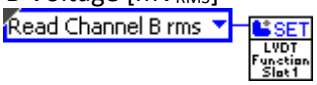
This instruction reads the programmed frequency of the excitation voltage. The return data format is a 16bit engineering value in [Hz].

Instruction [U8]:	Tx-Data [U8]:	Receive-Data [U16]:	Data Sink:
Read Channel A (6)	no data	Ch A Voltage [mV_{RMS}]	LVDT Controller



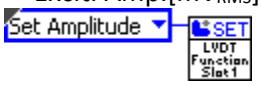
This instruction reads input channel A voltage amplitude. The return data format is a 16bit engineering value with dimension [mV_{RMS}]. This value is NOT for Position calculation and updated only 9 times per Second

Instruction [U8]:	Tx-Data [U8]:	Receive-Data [U16]:	Data Sink:
Read Channel B (7)	no data	Ch B Voltage [mV_{RMS}]	LVDT Controller



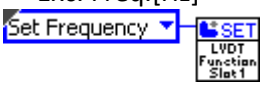
This instruction reads input channel B voltage amplitude. The return data format is a 16bit engineering value with dimension [mV_{RMS}]. This value is NOT for Position calculation and updated only 9 times per Second

Instruction [U8]:	Tx-Data [U8]:	Receive-Data [U16]:	Data Sink:
Set Amplitude (9)	Excit. Amp.[mV_{RMS}]	Excit. Amp.[mV_{RMS}]	LVDT Controller



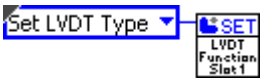
This instruction programs the excitation amplitude. The data format is a 16bit engineering value with dimension [mV_{RMS}].
Valid Amplitude Range: 2000 [mV_{RMS}].. 7000 [mV_{RMS}]

Instruction [U8]:	Tx-Data [U8]:	Receive-Data [U16]:	Data Sink:
Set Frequency (10)	Exc. Freq.[Hz]	Exc. Freq.[Hz]	LVDT Controller



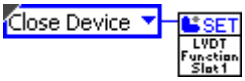
This instruction programs the excitation frequency. The data format is a 16bit engineering value with dimension [Hz].
Valid Frequency Range: 1000 [Hz].. 10000 [Hz]

Instruction [U8]:	Tx-Data [U8]:	Receive-Data [U16]:	Data Sink:
Set LVDT Type (8)	Res.Mode	Res.Mode	LVDT Controller



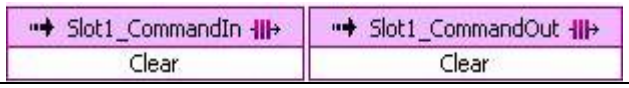
This instruction programs the demodulation stage. The data format is enumeration according to the following table.
Valid position resolution data: Data = 0, 6 Wire LVDT demodulation
Data = 1, 4 Wire LVDT demodulation

Instruction [U8]:	Tx-Data [U8]:	Receive-Data [U16]:	Data Sink:
Close Device (16)	No Data	No Data	FPGA Driver-VI



This instruction stops and terminates the applicable FPGA driver-VI.

5.3 LVDT9312 Driver Error Codes

Error Code [U8]	Error Type	Error Detection	Description
0	No Error	none	If no error has appeared during communication with the LVDT9312 module, the received telegram has error code „0“.
1	Unknown Command	LVDT Module	This error occurs when the LVDT Module receives an unknown instruction code.
2	Communication Error	LVDT Module	This error occurs when the FPGA aborts a telegram transmission to the LVDT Module or when an internal communication error of the module occured
3	Invalid Parameter	LVDT Module	This error occurs when the parameter “Data” is out of range.
8	Excitation ShortCircuit	LVDT Module	This error occurs if the Excitation is in Short Circuit and deactivated due to thermal limits. The Excitation will be disabled for 5 seconds and the Error will remain 10 more seconds. Regularly driving the Excitation in its thermal limit will degrade performance!
100	No Module	FPGA Driver-VI	This error occurs when no LVDT Module is connected to the applicable slot.
101	Module Busy	Functions-VI	The busy signal to the FPGA indicates module readiness. When a time consuming instruction is processed by the module, the busy signal becomes valid for a certain amount of time. However, if a new instruction is sent to the module while the module still processes a previous instruction (and therefore the busy line still is active), the FPGA driver waits for the busy line to return to the ready state. The maximum idle time for the FPGA driver is a default value and can be increased from the user. When the specified timeout is exceeded, the error “Module Busy” is generated by the functions-VI.
103	Transmit Incomplete	FPGA Driver-VI	This error occurs when the LVDT Module aborts telegrams or exceeds the driver timing constraints.
104	Wrong Driver Answer	Functions-VI	This error occurs when the replied telegram does not match with the instruction telegram. Note: This error may occur subsequently to a module power interrupt while the application is still running. However, this is only a singular error.
105	FIFO Timeout	Functions-VI	This error occurs if a FIFO fault takes place during data transfer from the functions-VI to the applicable driver-VI. Note: To reset this error scenario, both communication FIFOs (SlotN_CommandIN und SlotNCommandOUT) of the applicable module must be cleared with the „Clear“ instruction 
106	Power Error	Functions VI	This Error occurs if the external Supply of the Module is not in it’s limits.
107	Loose of Tracking	LVDT Module	This error occurs if both signals (Ua/Ub) are below 350mVrms. - In 4 Wire mode this error occurs only if Excitation and Ua are below 350mVRms.

5.4 Module Identification Under LabVIEW

LabVIEW automatically detects the cRIO module type (Discover C-Series Modules),

even if the external Supply is not applied.

5.5 SAVING THE SETUP

The LVDT9312 automatically saves the setup (Excitation Amplitude, Excitation Frequency and LVDT demodulation mode) in a non-volatile memory. After a power cycle the last set Parameters are active when the Module is switched on.

6. TECHNICAL SPECIFICATION

6.1 HOUSING

- NI CompactRIO 9952 Standard Housing
- Dimensions: appr. 71mm x 72mm x 23mm



6.2 EXTERNAL POWER SUPPLY

- Supply Voltage: $9V_{DC}$ – $36V_{DC}$
- Power Consumption: max. 1.5 Watt without Resolver

Caution:
The external power supply input is isolated from the cRIO chassis signals with a maximum isolation voltage of $500V_{DC}$. Note that the excitation output and the Signals Ua and Ub are galvanically connected to the external power supply.

6.3 EXCITATION OUTPUT

- Frequency: 1kHz – 10kHz, adjustable via software
- Amplitude: $2V_{RMS}$ – 7 V_{RMS}, adjustable via software
- Current: max. $120mA_{RMS}$

Caution:
The excitation output can be damaged when a power source is connected to its terminals. Driving the excitation regularly in short circuit condition will degrade performance!

6.4 UA AND UB SIGNAL INPUTS

- Voltage Range: $0V_{RMS}$ – 7 V_{RMS}
- Zin (differential): 108kOhm
- Zin (single-ended): 27.5kOhm

Caution:
The signal inputs can be damaged, when the input voltage exceeds a limit of $40V_{peak}$.

6.5 POSITION PROCESSING

- Resolution: 16Bit
- Bandwidth ½ Excitation Frequency
- Accuracy: 0.1%
  (accuracy of the module, the Position accuracy depends on the xVDT, the wiring and the Position of the xVDT)

6.6 ENVIRONMENTAL CONDITIONS

- Temperature Range: 0°C – 70°C
- Humidity: 10% – 90% relative, non-condensing

6.7 CONNECTOR PINOUT

User Interface Connector (25-pin Sub-D / male):

Pin	Function
1	V_{SUP}
2	COM
3	GND
4	A+
5	A-
6	GND
7	B+
8	B-
9	GND
10	+EXC
11	-EXC
12 - 25	n.c.

7. MODULE CALIBRATION

The LVDT9312 is calibrated when shipped. A recalibration can be done by SET GmbH.

8. MODULE MAINTENANCE

No maintenance is required for the LVDT Module.

9. SERVICE ADDRESS

For technical support contact the SET service address:

SET GmbH
August-Braun-Str. 1
88239 Wangen/Allgäu
Germany
Tel.: +49 (0)7522 91687-600
Fax: +49 (0)7522 91687-899
e-Mail: support.crio@smart-e-tech.de

Artisan Technology Group is an independent supplier of quality pre-owned equipment

Gold-standard solutions

Extend the life of your critical industrial, commercial, and military systems with our superior service and support.

We buy equipment

Planning to upgrade your current equipment? Have surplus equipment taking up shelf space? We'll give it a new home.

Learn more!

Visit us at **artisantg.com** for more info on price quotes, drivers, technical specifications, manuals, and documentation.

Artisan Scientific Corporation dba Artisan Technology Group is not an affiliate, representative, or authorized distributor for any manufacturer listed herein.

We're here to make your life easier. How can we help you today?

(217) 352-9330 | sales@artisantg.com | artisantg.com

ARTISAN® TECHNOLOGY GROUP