

# 6088 SERIES (14 BIT)

PUBLICATION NO. 980910

EADS North America Defense Test and Services, Inc.  
4 Goodyear, Irvine, CA 92618  
Tel: (800) 722-2528, (949) 859-8999; Fax: (949) 859-7139

[info@eads-nadefense.com](mailto:info@eads-nadefense.com)  
[sales@eads-nadefense.com](mailto:sales@eads-nadefense.com)  
[helpdesk@eads-nadefense.com](mailto:helpdesk@eads-nadefense.com)  
<http://www.eads-nadefense.com>



---

**PUBLICATION DATE: March 03, 2005**

Copyright 2004 by EADS North America Defense Test and Services, Inc. Printed in the United States of America. All rights reserved. This book or parts thereof may not be reproduced in any form without written permission of the publisher.

---

---

**THANK YOU FOR PURCHASING THIS  
EADS NORTH AMERICA DEFENSE TEST AND SERVICES PRODUCT**

---

---

For this product, or any other EADS North America Defense Test and Services, Inc. product that incorporates software drivers, you may access our web site to verify and/or download the latest driver versions. The web address for driver downloads is:

<http://www.eads-nadefense.com/downloads>

If you have any questions about software driver downloads or our privacy policy, please contact us at:

[info@eads-nadefense.com](mailto:info@eads-nadefense.com)

---

---

**WARRANTY STATEMENT**

---

---

All EADS North America Defense Test and Services, Inc. products are designed and manufactured to exacting standards and in full conformance to EADS ISO 9001:2000 processes.

For the specific terms of your standard warranty, or optional extended warranty or service agreement, contact your EADS North America Defense Test and Services, Inc. customer service advisor. Please have the following information available to facilitate service.

1. Product serial number
2. Product model number
3. Your company and contact information

You may contact your customer service advisor by:

E-Mail:	<a href="mailto:Helpdesk@eads-nadefense.com">Helpdesk@eads-nadefense.com</a>	
Telephone:	+1 800 722 3262	(USA)
Fax:	+1 949 859 7309	(USA)

---

---

**RETURN of PRODUCT**

---

---

Authorization is required from EADS North America Defense Test and Services, Inc. before you send us your product for service or calibration. Call or contact the Customer Support Department at 1-800-722-3262 or 1-949-859-8999 or via fax at 1-949-859-7139. We can be reached at: [helpdesk@eads-nadefense.com](mailto:helpdesk@eads-nadefense.com).

---

---

## **PROPRIETARY NOTICE**

---

---

This document and the technical data herein disclosed, are proprietary to EADS North America Defense Test and Services, Inc., and shall not, without express written permission of EADS North America Defense Test and Services, Inc., be used, in whole or in part to solicit quotations from a competitive source or used for manufacture by anyone other than EADS North America Defense Test and Services, Inc. The information herein has been developed at private expense, and may only be used for operation and maintenance reference purposes or for purposes of engineering evaluation and incorporation into technical specifications and other documents which specify procurement of products from EADS North America Defense Test and Services, Inc.

---

---

## **DISCLAIMER**

---

---

Buyer acknowledges and agrees that it is responsible for the operation of the goods purchased and should ensure that they are used properly and in accordance with this handbook and any other instructions provided by Seller. EADS North America Defense Test and Services, Inc. products are not specifically designed, manufactured or intended to be used as parts, assemblies or components in planning, construction, maintenance or operation of a nuclear facility, or in life support or safety critical applications in which the failure of the EADS North America Defense Test and Services, Inc. product could create a situation where personal injury or death could occur. Should Buyer purchase EADS North America Defense Test and Services, Inc. product for such unintended application, Buyer shall indemnify and hold EADS North America Defense Test and Services, Inc., its officers, employees, subsidiaries, affiliates and distributors harmless against all claims arising out of a claim for personal injury or death associated with such unintended use.

---

# FOR YOUR SAFETY

---

Before undertaking any troubleshooting, maintenance or exploratory procedure, read carefully the **WARNINGS** and **CAUTION** notices.



**CAUTION**  
RISK OF ELECTRICAL SHOCK  
DO NOT OPEN



This equipment contains voltage hazardous to human life and safety, and is capable of inflicting personal injury.



If this instrument is to be powered from the AC line (mains) through an autotransformer, ensure the common connector is connected to the neutral (earth pole) of the power supply.



Before operating the unit, ensure the conductor (green wire) is connected to the ground (earth) conductor of the power outlet. Do not use a two-conductor extension cord or a three-prong/two-prong adapter. This will defeat the protective feature of the third conductor in the power cord.



Maintenance and calibration procedures sometimes call for operation of the unit with power applied and protective covers removed. Read the procedures and heed warnings to avoid “live” circuit points.

Before operating this instrument:

1. Ensure the proper fuse is in place for the power source to operate.
2. Ensure all other devices connected to or in proximity to this instrument are properly grounded or connected to the protective third-wire earth ground.

If the instrument:

- fails to operate satisfactorily
- shows visible damage
- has been stored under unfavorable conditions
- has sustained stress

Do not operate until, performance is checked by qualified personnel.

# Racal Instruments

## EC Declaration of Conformity

We

Racal Instruments Inc.  
4 Goodyear Street  
Irvine, CA 92618

declare under sole responsibility that the

**6088-14 Series, P/N 407989-XXX**

conforms to the following Product Specifications:

**EMC:** EN61326:1997 + A1:1998 + A2:2001, Class A

**Safety:** EN61010-1:1993 + A2:1995

**Supplementary Information:**

The above specifications are met when the product is installed in a Racal Instruments certified mainframe with faceplates installed over all unused slots, as applicable

The product herewith complies with the requirements of the Low Voltage Directive 73/23/EEC and the EMC Directive 89/336/EEC.

Irvine, CA, January 26, 2005:

  
Director of Engineering

This page was left intentionally blank.

**Table Of Contents**

**Chapter 1** ..... 1-1

GENERAL DESCRIPTION ..... 1-1

    Introduction..... 1-1

    General Description..... 1-2

    6088-14 Series Specifications..... 1-4

Functional Specifications ..... 1-4

Electrical Specifications ..... 1-5

    Digitizer Specifications ..... 1-5

Power Requirements ..... 1-7

Mechanical Specifications..... 1-7

    DAC Specifications ..... 1-8

Environmental Specifications..... 1-10

EMC Specifications..... 1-10

**Chapter 2** ..... 2-1

INSTALLATION INSTRUCTIONS..... 2-1

    Introduction..... 2-1

    Unpacking and Inspection..... 2-1

    Address Switch Settings for Configuration Control ..... 2-1

    6088-14 Series to VXI plug & play Main Frame Installation ..... 2-2

    Power-up Initialization ..... 2-3

    Configuration Defaults..... 2-3

---

Chapter 3 .....	3-1
DIGITIZER .....	3-1
Introduction .....	3-1
Analog Circuit .....	3-3
Input Coupling .....	3-3
Input Ranges .....	3-4
Input Offset .....	3-5
Filters .....	3-6
Triggering .....	3-7
Digital Input Triggering .....	3-7
Software Triggering .....	3-8
Analog Triggering .....	3-9
Trigger Timing .....	3-11
Clocking .....	3-12
Onboard Memory .....	3-14
Calibration .....	3-16
VXI Interface .....	3-16
Register Assignment .....	3-17
Channel A .....	3-17
Channel B .....	3-18



Chapter 4 .....	4-1
2165 DIGITIZER SOFTWARE .....	4-1
Introduction.....	4-1
2165 Driver Package.....	4-1
Installation .....	4-1
2165 DLL Functions .....	4-2
ri2165_AutoCalibrate (instrumentHandle, mode, displayTime).....	4-2
ri2165_CheckTestStatus (instrumentHandle, testStatus) .....	4-3
ri2165_Close (instrumentHandle).....	4-3
ri2165_CodeToVoltage (instrumentHandle, ADCCode, voltage) .....	4-3
ri2165_ConnectCard (instrumentHandle, positiveInputConnection, negativeInputConnection ) .....	4-4
ri2165_GetActive Channel (instrumentHandle, channel ) .....	4-4
ri2165_GetAddress Counter(instrument Handle, addresscounter ) .....	4-5
ri2165_GetCardAddress (instrumentHandle, address ) .....	4-5
ri2165_GetCard Connection (instrument Handle, positiveInput Connection, negativeInput Connection ) .....	4-5
ri2165_GetCard List (count, buslist, devicelist) .....	4-6
ri2165_GetClock Divider (instrumentHandle, clockDivider) .....	4-6
ri2165_GetClock Source (instrumentHandle, clockSource).....	4-7
ri2165_GetDCOffset LimitVoltages (instrumentHandle, positiveVoltage, negativeVoltage).....	4-7
ri2165_GetDCOffset Voltage(instrument Handle, voltage ) .....	4-7
ri2165_GetError Message(code, message) .....	4-8
ri2165_GetFilter (instrumentHandle, filterStatus).....	4-8
ri2165_GetGainCalCode (instrumentHandle, gainCalibrationCode).....	4-8
ri2165_GetInputVoltage (instrumentHandle, voltage, code, averages, timeOut) .....	4-9
ri2165_GetLoopMode (instrumentHandle, loopMode) .....	4-9

---

ri2165_GetOffsetCalDacCode(instrument Handle, code) .....	4-9
ri2165_GetRange (instrumentHandle, range) .....	4-10
ri2165_GetRevision (revision).....	4-10
ri2165_GetSample Divider (instrumentHandle, sampleDivider).....	4-10
ri2165_GetSoftware TriggerStatus(instrumentHandle, trigger status ) .....	4-11
ri2165_GetTriggerInput (instrumentHandle, triggerSource, triggerMode).....	4-11
ri2165_Init( bus, device, instrumentHandle ).....	4-12
ri2165_MemoryTest (instrumentHandle, level, errorAddress).....	4-12
ri2165_ReadAdcResults (instrumentHandle, startPosition, samples, buffer, include Status Bits)..	4-13
ri2165_ReadEEPROM (instrumentHandle, eeaddress, data).....	4-13
ri2165_ReadId (instrumentHandle, id ) .....	4-14
ri2165_ReadRam (instrumentHandle, data ).....	4-14
ri2165_ReadRam Buffer(instrumentHandle, length, buf32, data Interpretation ).....	4-14
ri2165_SetActive Channel(instrument Handle, channel ).....	4-15
ri2165_SetAddress Counter (instrument Handle, address CounterPosition).....	4-15
ri2165_SetClock Divider(instrumentHandle, clock divider ) .....	4-16
ri2165_SetClock Source(instrumentHandle, clock source ).....	4-16
ri2165_SetDCOffset Code(instrument Handle, code, connect ).....	4-16
ri2165_SetDCOffset Voltage(instrumentHandle, voltage, connect ) .....	4-17
ri2165_SetDC OffsetLimitVoltages(instrumentHandle, posvolt, negvolt ).....	4-17
ri2165_SetFilter (instrumentHandle, filter ).....	4-18
ri2165_SetGainCal Code(instrument Handle, code).....	4-19
ri2165_SetLock Mode(instrument Handle, lock ) .....	4-19
ri2165_SetLoop Mode(instrument Handle, loopMode).....	4-19
ri2165_SetOffset CalCode(instrumentHandle, code).....	4-20
ri2165_SetRange (instrumentHandle, range ).....	4-20
ri2165_SetSample Divider(instrument Handle, sample DividerValue) .....	4-20

ri2165_Set Software TriggerStatus (instrumentHandle, triggerstatus ).....	4-21
ri2165_Set Trigger Input(instrument Handle,trigger source,triggermode ).....	4-21
ri2165_Store CalibrationData (instrumentHandle ) .....	4-21
ri2165_WriteEEPROM (instrumentHandle, eeaddress, data ).....	4-22
ri2165_WriteId (instrumentHandle, id ).....	4-22
ri2165_WriteRam (instrumentHandle, data ).....	4-23
ri2165_WriteRam Buffer(instrument Handle, length, buffer ) .....	4-23
Status Codes.....	4-24

<b>Chapter 5</b> .....	<b>5-1</b>
2165 DIGITIZER SOFTWARE UTILITY.....	5-1
Introduction.....	5-1
Adjusting the Instrument.....	5-3
Block Diagram .....	5-3
The Instrument Settings Panel .....	5-4
Performing a Test.....	5-5
Analyzing the Data .....	5-6
General.....	5-6
Time Domain .....	5-7
Frequency Domain .....	5-7
Frequency Domain Settings .....	5-10
Parameter Calculation .....	5-12

**Chapter 6** ..... 6-1

2165 DIGITIZER CALIBRATION ..... 6-1

    Introduction ..... 6-1

    Calibration Procedure ..... 6-1

    Checking the Calibration Values ..... 6-2

**Chapter 7** ..... 7-1

DIGITAL TO ANALOG CONVERTER (DAC)..... 7-1

    Introduction ..... 7-1

    Device Operation..... 7-3

    Memory Operation..... 7-3

    Trigger Processing ..... 7-5

    Clock Select ..... 7-7

    Filter Select ..... 7-8

    Output Stage ..... 7-9

    DAC Range ..... 7-10

    Attenuator..... 7-10

    DAC Offset ..... 7-11

    Output Offset Calibration DAC ..... 7-12

    Offset Correction DAC..... 7-14

    Memory Segmentation ..... 7-15

    Serial EEPROM..... 7-16

    Front Panel..... 7-17

        OUT A+ and OUT A- ..... 7-17

        OUT B+ and OUT B- ..... 7-17

CLK in/out.....	7-17
TRIG in .....	7-17
VXI Interface.....	7-17
Register Assignment:.....	7-18
Channel-A.....	7-18
Channel-B.....	7-19

**Chapter 8** ..... 8-1

3165 SOFTWARE.....	8-1
Introduction.....	8-1
3165 Driver Package.....	8-1
Installation .....	8-1
3165 DLL Functions .....	8-2
ri3165_Close (instrumentHandle).....	8-2
ri3165_ConnectCard (instrumentHandle, cc ) .....	8-3
ri3165_GetActive Channel(instrumentHandle, channel ) .....	8-3
ri3165_GetAddress Counter(instrumentHandle, address counter ) .....	8-3
ri3165_GetAttenuator (instrumentHandle, attenuation ).....	8-4
ri3165_GetCard Address(instrument Handle, address ) .....	8-4
ri3165_GetCard Connection(instrument Handle, cc ) .....	8-4
ri3165_GetCard List(count, buslist, devicelist).....	8-5
ri3165_GetCard Number(instrument Handle, card ) .....	8-5
ri3165_GetDC OffsetVoltage (instrumentHandle, voltage ) .....	8-5
ri3165_GetError Message(code, message) .....	8-6
ri3165_Get Number OfCards( cards ) .....	8-6
ri3165_Get OffsetCalDacCode (instrumentHandle, code) .....	8-6
ri3165_Get OutputOffsetCalDac Code(instrument Handle, dac, code).....	8-7

---

ri3165_Get Revision(revision).....	8-7
ri3165_Get TriggerStatus (instrumentHandle, triggerstatus ).....	8-7
ri3165_GetStart Address(instrument Handle, start address ) .....	8-8
ri3165_GetStop Address(instrument Handle, stop address).....	8-8
ri3165_Init( bus, device, instrument Handle).....	8-8
ri3165_Load ArbitraryWaveForm (instrumentHandle, startaddress, length, waveform) .....	8-9
ri3165_Read (instrumentHandle, offset, data) .....	8-9
ri3165_Read EEPROM(instrument Handle, eeaddress, data).....	8-9
ri3165_ReadId (instrumentHandle, id ) .....	8-10
ri3165_ReadRam (instrumentHandle, data ).....	8-10
ri3165_Read RamBuffer(instrumentHandle, length, buf32 ) .....	8-10
ri3165_Reset ToStartAddress (instrumentHandle) .....	8-11
ri3165_Set ActiveChannel (instrumentHandle, channel ).....	8-11
ri3165_Set Attenuator(instrumentHandle, attenuation ).....	8-12
ri3165_Set ClockDivider (instrumentHandle, clockdivider ) .....	8-12
ri3165_Set ClockSource (instrumentHandle, clocksource ).....	8-12
ri3165_Set DacCode(instrument Handle, code ).....	8-13
ri3165_SetDC OffsetDacCode (instrumentHandle, code ).....	8-13
ri3165_Set DCOffsetVoltage (instrumentHandle, voltage ).....	8-14
ri3165_SetDC OffsetLimitVoltages (instrumentHandle, posvolt, negvolt ) .....	8-14
ri3165_SetFilter (instrumentHandle, filter ).....	8-15
ri3165_SetLock Mode(instrument Handle, lock ) .....	8-15
ri3165_SetOffset CalDacCode (instrumentHandle, code).....	8-15
ri3165_Set OutputOffsetCalDac Code(instrument Handle, dac,code).....	8-16
ri3165_SetRange DacCode(instrument Handle, code ).....	8-16
ri3165_SetRange LimitVoltages (instrumentHandle, maxvolt, minvolt ) .....	8-16
ri3165_Set SoftwareTriggerStatus (instrumentHandle, triggerstatus ).....	8-17

ri3165_Set StartAddress (instrumentHandle, startaddress ) .....	8-18
ri3165_Set StopAddress (instrumentHandle, stopaddress ) .....	8-18
ri3165_Set TriggerMode (instrumentHandle, triggersource, triggermode ) .....	8-18
ri3165_Signal Add(instrumentHandle, type, amplitude, periods, phase, symmetry) .....	8-19
ri3165_Signal Clear(instrument Handle) .....	8-20
ri3165_Signal ToRam(instrument Handle) .....	8-20
ri3165_Store CalibrationData (instrumentHandle).....	8-21
ri3165_Write (instrumentHandle, offset , data ).....	8-21
ri3165_Write EEPROM(instrument Handle, eeaddress, data ) .....	8-21
ri3165_WriteId (instrumentHandle,id ).....	8-22
ri3165_Write Ram(instrument Handle, data ).....	8-22
ri3165_Write RamBuffer (instrumentHandle, length, buffer ) .....	8-22
Status Codes .....	8-23
General.....	8-23
Programming example .....	8-24

<b>Chapter 9</b> .....	9-1
<b>3165 SOFTWARE UTILITY</b> .....	9-1
General.....	9-1
Load and Save Instrument Configuration .....	9-2
Adjusting the Instrument.....	9-2
Standard Signals .....	9-5
Import Signals .....	9-6
Decimal Import Data.....	9-6
Hexadecimal Import Data (instrument main-DAC codes).....	9-6
Updating The Instrument.....	9-7

**Chapter 10**..... 10-1

3165 CALIBRATION..... 10-1

    The Calibration Procedure ..... 10-1

**Chapter 11**..... 11-1

SOFTWARE DRIVER..... 11-1

    Overview ..... 11-1

**Chapter 12**..... 12-1

PRODUCT SUPPORT ..... 12-1

    Product Support ..... 12-1

    Warranty..... 12-1



**List Of Figures**

Figure 1-1, 6088-14 Series Front Panel (407989-003 shown)..... 1-1

Figure 1-2, 6088-14 System Hardware Architecture..... 1-3

Figure 2-1, Module Configuration Switch..... 2-3

Figure 3-1, 2165 Digitizer Functional Block Diagram..... 3-2

Figure 3-2, Analog Input Circuit ..... 3-3

Figure 3-3, Input Offset..... 3-5

Figure 3-4, Typical Frequency Response ..... 3-6

Figure 3-5, 2165 Trigger Capabilities..... 3-8

Figure 3-6, Analog Positive Level Triggering ..... 3-9

Figure 3-7, Analog Positive Edge Triggering ..... 3-10

Figure 3-8, Analog Positive Continuous Edge Trigger..... 3-10

Figure 3-9, External Trigger Timing Start..... 3-11

Figure 3-10, External Trigger Timing Stop..... 3-12

Figure 3-11, Clock Circuit ..... 3-13

Figure 3-12, Function of the Loopmode-Bit ..... 3-15

Figure 7-1, 3165 Functional Block Diagram..... 7-2

Figure 7-2, Pattern Generation Mode Block Diagram..... 7-4

Figure 7-3, Trigger Processing ..... 7-5

Figure 7-4, Clock Select..... 7-7

Figure 7-5, Filter Select..... 7-8

Figure 7-6, Output Stage ..... 7-9

Figure 7-7, Serial DAC Timing Diagram ..... 7-12

Figure 7-8, Output Offset Calibration DAC .....7-12

Figure 7-9, Calibration DAC Addressing.....7-13

Figure 7-10, Memory Segmentation .....7-15

Figure 7-11, 3165 Front Panel.....7-17

**List of Tables**

Table 3-1, Input Ranges 2165 Digitizer .....3-4

Table 7-1, Serial DAC Control Line Assignment.....7-11

Table 7-2, Serial Calibration DAC Control Line Assignment.....7-14

Table 7-3, Serial EEPROM Control Line Assignment.....7-16

# Chapter 1

## GENERAL DESCRIPTION

---

### Introduction

This manual contains information on how to install and use the 6088-14 Series 14-bit products in a VXI plug & play environment.



Figure 1-1, 6088-14 Series Front Panel (407989-003 shown)

The 6088-14 Series 14-bit high speed digitizers & DACs are register or message based VXI Modules. They come in 5 variations as follows:

Model	Description	Part Number
6088-14-DIG	2-Channel 14-bit, 70MHz Digitizer	407989-001
6088-14-DIG-DIG	4- Channel 14-bit, 70MHz Digitizer	407989-002
6088-14-DIG-DAC	2-Channel 14-bit Digitizer, 2 Ch. DAC	407989-003
6088-14-DAC	2-Channel 14-bit, 70/100MHz DAC	407989-011
6088-14-DAC-DAC	4- Channel 14-bit, 70/100MHz DAC	407989-012

## General Description

The 6088-14 Series interface logic maps the internal registers of the carrier modules to the VXI A24 Address space. All transfers to and from the carrier modules are full 32 bit data transfers. Refer to the module address memory map in Chapter 3 for specific register information.

The carrier modules are referred to as Module "0" and Module "1". Each of the modules inputs and/or outputs is brought out to the VXI front panel.

An external front panel clock has been implemented to drive the CLK signals of each of the carrier modules. An internal clock can be generated either by selecting one of the module internal clocks, the external clock or CLK10 (for DAC).

An external front panel trigger has been implemented.

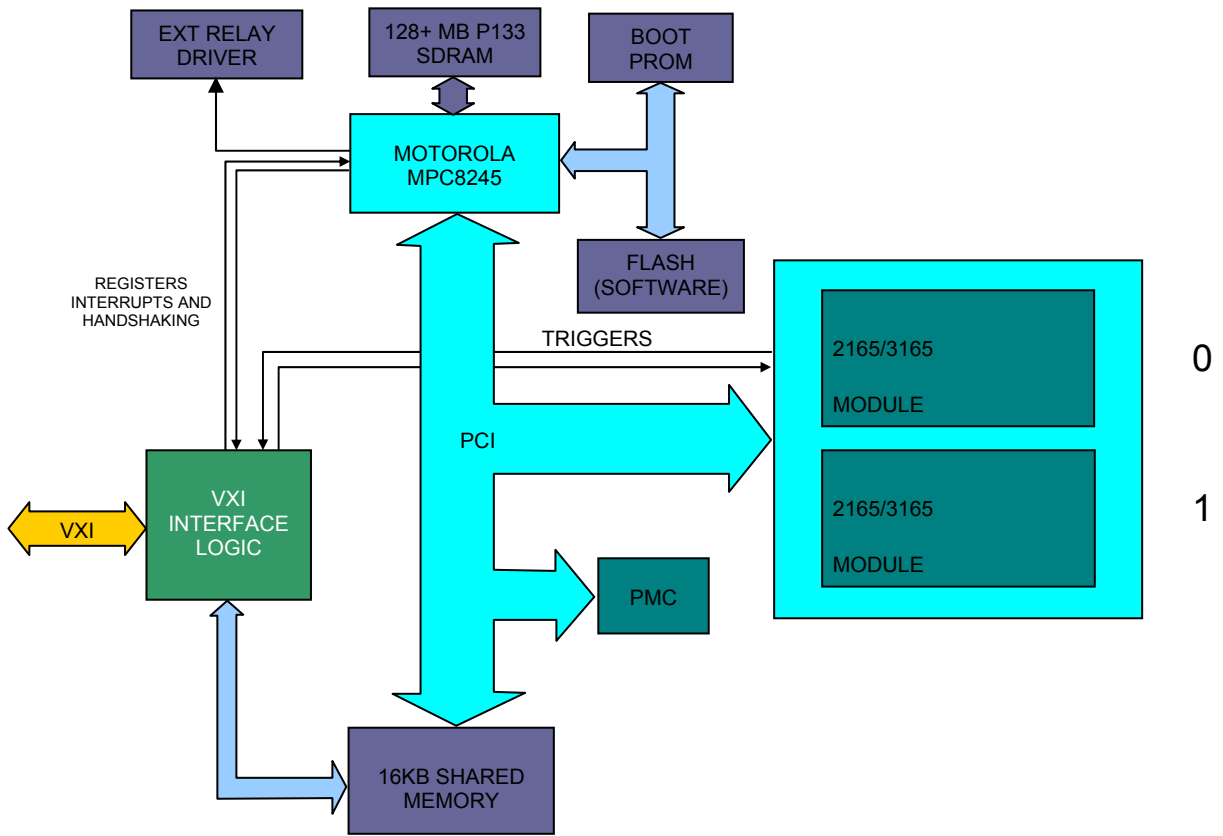


Figure 1-2, 6088-14 System Hardware Architecture

## 6088-14 Series Specifications

### FUNCTIONAL SPECIFICATIONS:

<b>Modes of Operation:</b>	Register-based or message based VXI Servant  Capture waveform to memory, generate waveform from memory
<b>Front panel connectors:</b>	Digitizer: IN A+, IN A-, External Clock, Trigger, IN B+, IN B-  DAC: OUT A+, OUT A-, External Clock, Trigger, OUT B+, OUT B-  All are SMB connectors
<b>Front panel controls:</b>	None
<b>Front panel indicators:</b>	None
<b>Rear panel connectors:</b>	VXI connectors P1 & P2
<b>Rear panel controls:</b>	None
<b>Rear panel indicators:</b>	None
<b>Other major components:</b>	2165 Digitizer modules, 3165 DAC modules
<b>Interface characteristics</b>	VXI-1, Rev. 1.4
<b>Manufacturer ID no:</b>	FC1 hex (can also be set by power PC)
<b>Model No:</b>	FE4 hex
<b>Logical Addressing:</b>	Static 1-254, Switch Selectable
<b>Address Space:</b>	A16/A32 support
<b>Data Transfer Bus:</b>	D16/D32 support
<b>Interrupt Levels:</b>	None
<b>VXI plug &amp; play Interface Drivers:</b>	LabWindows/CVI Instrument Panel LabVIEW Digitizer Utility, Labview DAC utility
<b>Sample Memory:</b>	512k words

## ELECTRICAL SPECIFICATIONS:

### Digitizer Specifications:

Typical at 22 °C

#### General:

Number of channels	2
ADC resolution	14 bit
Sample rate	500kHz.....70 MHz
Memory depth	512k per channel

#### Analog:

Input configuration	Differential or single ended
Input ranges (span)	1Vpp, 2Vpp, 4Vpp. With attenuator on: 5Vpp, 10Vpp, 20Vpp
Input impedance	50Ω AC, 50Ω DC and 10kΩ DC
Input coupling	DC
DC Input offset range	-5Volt...+5V at 1V,2V and 4V range -25Volt...+25Volt at 5V, 10V and 20V range
Input filters.	None, 30MHz, 15MHz, 6MHz (3-pole Butterworth)
Bandwidth (-3dB) no filter	70MHz
Channel crosstalk	<70dB at 1MHz
Frequency response, referenced at 500kHz (no filter)	± 0.5dB DC.....20MHz ± 2.0dB 20MHz...50MHz
Absolute accuracy	±(500μV+0.1% of range), with attenuator on: ±(2.5mV+0.2% of range)
Relative accuracy (INL)	±0.025% of range
SFDR (fs = 50MHz/ Vin = 2Vpp)	80dB @ f-in = 1MHz 72dB @ f-in = 10MHz
SINAD (fs = 50MHz/ Vin = 2Vpp)	68dB @ f-in = 1MHz 64dB @ f-in = 10MHz

#### Sample clock:

<b>Clock sources</b>	Internal, external
----------------------	--------------------

#### Internal clock:

Internal clock frequency	50MHz and 70MHz (dividable by 1 to 256)
Clock accuracy	50MHz and 70MHz ±100ppm

**External clock input:**

External clock source SMB connector front  
 Maximum external clock frequency 100 MHz  
 External clock levels Vlow < 0.6V Vhigh > 1.4Volt  
 External clock impedance 50Ω DC

**External clock output:**

Clock output impedance 50Ω DC  
 Clock output levels Vlow < 0.6V Vhigh > 4.5Volt (no load)

**Triggering:**

**Trigger sources**

Front Panel (External), VXI triggers, software trigger, analog trigger

Trigger modes

Positive level, negative level, positive edge, negative edge, positive edge continuous and negative edge continuous

**Front trigger impedance**

10kΩ DC

Front trigger levels

Vlow < 0.6V Vhigh > 2.4Volt

**Power Requirements:**

+5V	+12V	-12V
950mA	40mA	40mA



## POWER REQUIREMENTS:

<b>Power consumption:</b>	12 W + 10W per DAC+ 5.71W per digitizer
<b>Peak Current (<math>I_{MP}</math>):</b>	4.3A @ +5V, 0.3A @ -12V, 0.3 A @ +12V

**NOTE 1:** All VXI DC power voltages used are specified per VXI plug & play Rev. 1.4.

## MECHANICAL SPECIFICATIONS:

<b>Enclosure Style:</b>	VXI "C" Size Module
<b>Enclosure Dimensions:</b>	14L x 10.3W x 1.2D
<b>Module weight:</b>	3 lbs. + 0.5 lbs per digitizer + 0.4 lbs per DAC
<b>Cooling:</b>	2.0 Liters/sec at approx. .25mm H <sub>2</sub> O for a 10°C Rise
<b>User Maintenance Items:</b>	None

**DAC SPECIFICATIONS:**

Typical 22 °C

**General:**

Number of channels	2
DAC resolution	14 bit
Update rate	DC- 100MHz
Memory depth	512k per channel

**Analog:**

Output configuration	Differential or single ended
Output ranges	350mVpp - 5Vpp proportional into open circuit
Output impedance	50Ω
Output coupling	DC
DC output offset range	-2.5 Volt to +2.5Volt
Output filters	None, 6MHz, 15MHz, 30MHz (3-pole Butterworth)
Bandwidth (-3dB) no filter	
Channel crosstalk	<80dB at 1MHz
Frequency response, referenced at 500kHz (no filter)	
Absolute accuracy	±(500μV+0.1% of range)
Relative accuracy (INL)	±0.025% of range
SFDR (fs = 100MHz/ Vout = 2Vpp)	78dB @ f-out = 1MHz 64dB @ f-out = 10MHz
SINAD (fs = 100MHz/ Vout = 2Vpp)	65dB @ f-out = 1MHz 62dB @ f-out = 10MHz

**Sample clock:**

<b>Clock sources</b>	Internal, external
----------------------	--------------------

**Internal clock:**

Internal clock frequency	50MHz and 70MHz dividable by 1 to 256
clock accuracy	50MHz and 70MHz ±100ppm

**External clock input:**

External clock source	SMB connector front
Maximum external clock frequency	100 MHz dividable by 1 to 256
External clock levels	Vlow < 0.6V Vhigh > 1.4Volt
External clock impedance	50Ω DC

**External clock output:**

Clock output impedance  
Clock output levels

50Ω DC  
Vlow < 0.6V Vhigh > 4.5Volt (no load)

**Triggering:**

**Trigger sources**

Front Panel, VXI triggers, software trigger

Trigger modes

Positive level, negative level, positive edge, negative edge, positive edge continuous and negative edge continuous.

**Front trigger impedance**

10kΩ DC

Front trigger levels

Vlow < 0.6V Vhigh > 2.4Volt

**Power Requirements:**

maximum power consumption

+5V	+12V	-12V
950mA	240mA	240mA

**Mechanical**

Size  
Weight

Single slot, 3U high (100mm x 160mm)  
210 gram

## ENVIRONMENTAL SPECIFICATIONS:

<b>Temperature, operating:</b>	0°C to +50°C
<b>Temperature, non-operating:</b>	0°C to +70°C
<b>Relative Humidity (%):</b>	10 to 80% RH non condensing $\leq$ 30°C; 10 to 70% RH above 30°C; 10 to 40% RH above 40°C

## EMC SPECIFICATIONS:

### Emissions / Immunity:

<b>FCC/IC</b>	FCC Part 18 & Industry Canada ICES-003
<b>CE Marking</b>	EN61326/EN55011 & EN61000-3-2/EN61000-3-3
<b>Conducted emissions:</b>	VXI plug & play Rev 1.4, B.8.7.3
<b>Conducted susceptibility:</b>	VXI plug & play Rev 1.4, B.8.7.4
<b>Radiated emissions (Closed Field):</b>	VXI plug & play Rev 1.4, B.8.6.3

## INSTALLATION INSTRUCTIONS

---

### Introduction

This section describes the unpacking, inspection, set-up and installation of the 6088-14 Series Register Based modules in a C-size mainframe.

### Unpacking and Inspection



1. Remove the 6088-14 Series module and inspect it for damage. If any damage is apparent, inform the carrier immediately. Retain shipping carton and packing material for the carrier's inspection.
2. Verify that the pieces in the package you received contain the correct 6088-14 Series module option and the 6088-14 Series Users Manual. Notify EADS North America Defense Test and Services, Inc. if the module appears damaged in any way. Do not attempt to install a damaged module into a VXI chassis.
3. The 6088-14 Series module is shipped in an anti-static bag to prevent electrostatic damage to the module. Do not remove the module from the anti-static bag unless it is in a static-controlled area.

### Address Switch Settings for Configuration Control

The 6088-14 Series Module has an internal 8-position address dip switch, LOG ADDR, used to determine the static configuration. It is located on the top of the module, accessible through the case.

In the ON position, the switch is set to logical 1, and in the OFF position to a logical 0.

The user can select any logical address from 1 to 254 for static configuration.

---

**NOTE:**

Logical address 0 is not allowed. Logical address 255 is not allowed because it is used for modules that support dynamic configuration.

---

---

**NOTE:**

The 6088-14 Series Modules are shipped with the logical address set for 2. Refer to the Resource Manager's manual for details about addressing methods.

---

## **6088-14 Series to VXI plug & play Main Frame Installation**

The 6088-14 Series Register Based Modules are ready for operation when shipped. The address switch is set to 2.

To install the 6088-14 Series module in a C-size VXI chassis, use the following instructions:

1. Ensure power is OFF.
2. Configure the interrupt daisy chain on the backplane to bypass empty slots, per VXI plug & play specifications.
3. Remove the front cover of the VXI chassis, and slide the 6088-14 Series module into the appropriate slot.

---

**NOTE:**

There is no need to configure the VXI Backplane Slot used for the 6088-14 Series Register Based Module, since all BUSGRANT and IACK signal lines are passed to the next slot.

---

## Power-up Initialization

Before turning on the VXI plug & play mainframe, make sure a Slot 0 with a Resource Manager is present. Upon power-up of the system, the 6088-14 Series modules goes through its power-up sequence.

## Configuration Defaults

The Logical address is determined by the Logical Address Switch. The default state of the Logical Address Switch is "36". The user can set the address to any unused Logical Address from 1-254.

Make sure that the module configuration switch and Power PC switches are set as shown below:

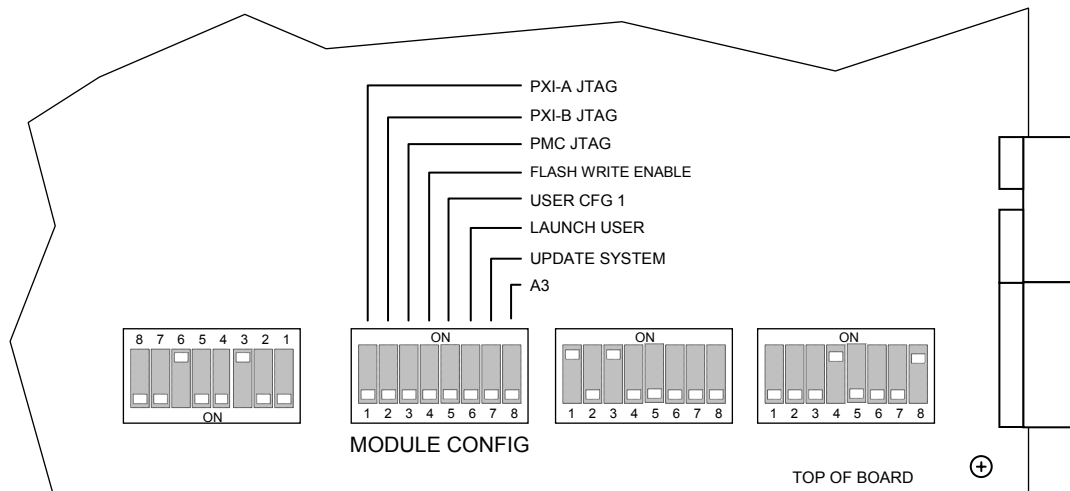


Figure 2-1, Module Configuration Switch

This page was left intentionally blank.



## Chapter 3

# DIGITIZER

---

### Introduction

The 6088-14 can accommodate one or two 2165 dual-channel high speed Waveform Digitizer card(s) installed in a dual VXI carrier. The two channels are completely separate and each channel has it's own 14 bit, 70 MS/s Analog to Digital Converter (ADC). This structure ensures low crosstalk and no phase error between the channels. **Figure 3-1** shows the functional block diagram for the 2165 digitizer.

On the left side of the diagram are the differential analog inputs, the external clock input /output and the external trigger input. The analog input is a differential input. By switching one input to ground or to the programmable DC Offset source it becomes a single ended input with programmable DC mid level. The input has six range steps. If desired the analog signal can be filtered to remove out of band noise. The EXTERNAL CLOCK is the sample clock in- or output. The two channels can sample simultaneously or at different, independent speeds. The TRIGGER input allows the user to control the start of data capture with external triggering.

After the 14 bit analog to digital conversion there is the 512k RAM capture memory.

When the 2165 digitizer is triggered, the DAC capture memory starts running from its start address.

During the measurement an internal or external clock increments the memory counter. When the counter reaches the value of the stop address it jumps back to the start address, or stops if the loop mode is programmed off.

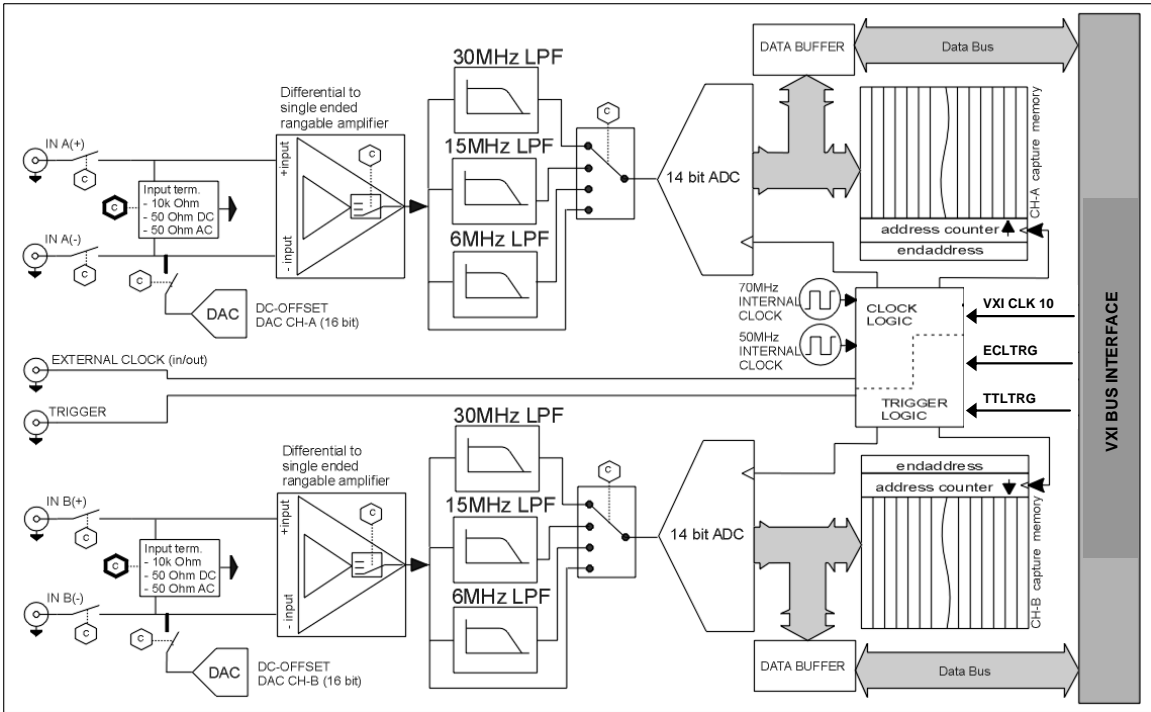


Figure 3-1, 2165 Digitizer Functional Block Diagram

## Analog Circuit

The SMB connectors labeled A+, A-, B+ and B-, on the front of the 2165 digitizer, are the analog differential inputs for Channel A and Channel B. If a channel is set in the single input mode the input signal must be connected to the + input. The negative input is connected to ground by the software.

## Input Coupling

Figure 3-2 shows the analog input circuit from one input. The analog input impedance is selectable, 50Ω AC, 50Ω DC or 10kΩ DC. If the analog input is disconnected, the internal circuit is disconnected from the input by a mechanical switch. The settings for the input mode are separate for each channel, so Channel A can be a 50Ω AC differential input, while channel B is a 10kΩ DC single-ended input.

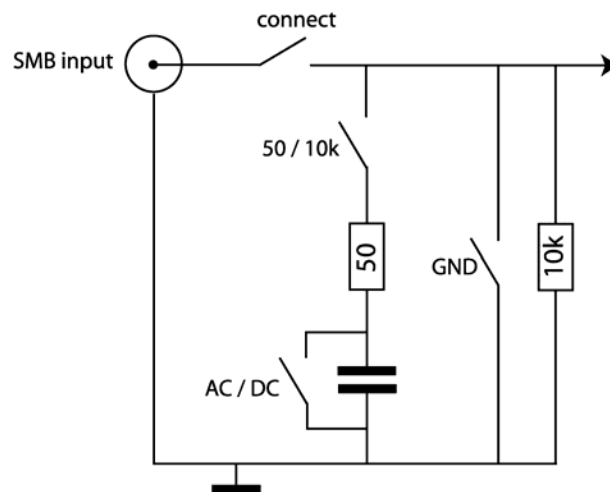


Figure 3-2, Analog Input Circuit

# Input Ranges

The 2165 digitizer offers six ranges to optimize the ADC resolution to the input signal.

**Table 3-1, Input Ranges 2165 Digitizer**

<b>Input Range</b>	<b>Input DC offset</b>
1Vpp	+5V...-5V
2Vpp	+5V...-5V
4Vpp	+5V...-5V
5Vpp	+25V...-25V
10Vpp	+25V...-25V
20Vpp	+25V...-25V

## Input Offset

At the input of the 2165 digitizer there is a 16 bit offset DAC. The input offset positions an input signal around a DC value. Depending on the range, the input offset span is +5Volt to -5Volt or +25Volt to -25Volt (**Table 3-1**). When the input offset is used the negative connector from the differential input is disabled.

Example: Examine a signal (1Vpp) on a 1 Volt DC offset, see **Figure 3-3**. The top of the input signal is 1Volt offset + 0.5V signal = 1.5Vpeak input voltage. Without using input offset, you would need to specify a range of 4Vpp to capture the waveform. In this case a large range from the ADC will not be used because of the DC voltage on the signal. However, with the input offset set to 1Volt, the signal centers around 0V and a range of 1Vpp is enough to capture the signal. This improves the accuracy of the measurement.

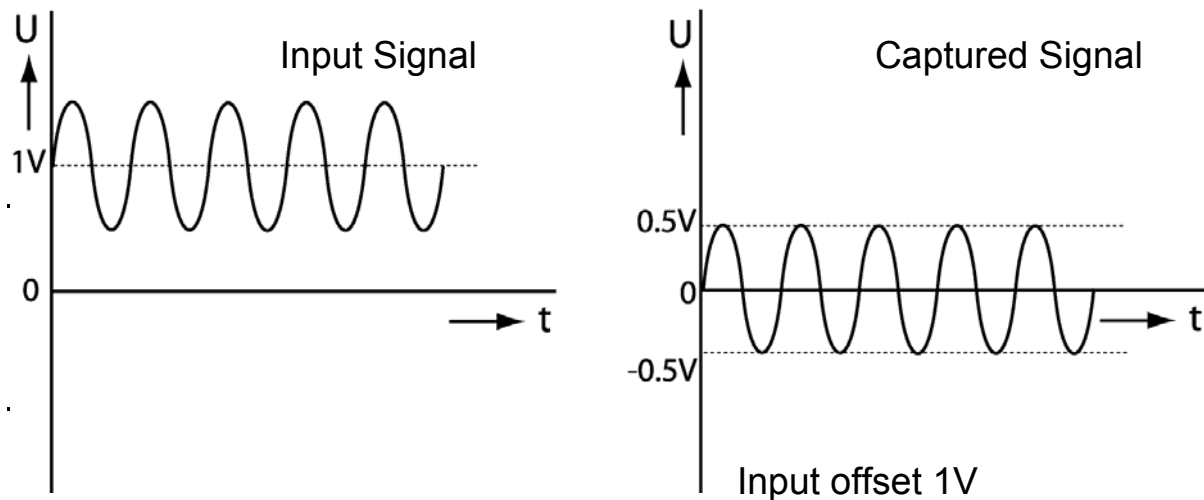


Figure 3-3, Input Offset

### Filters

Each channel has three selectable 3-pole low pass filters. The filters limit the bandwidth of the signal path and are useful for rejecting out of band noise. The selectable filters have cutoff frequencies of: 6 MHz, 15 MHz, and 30 MHz. The filters can also be bypassed for wideband signals. **Figure 3-4** has the typical frequency response of the 2165 digitizer.

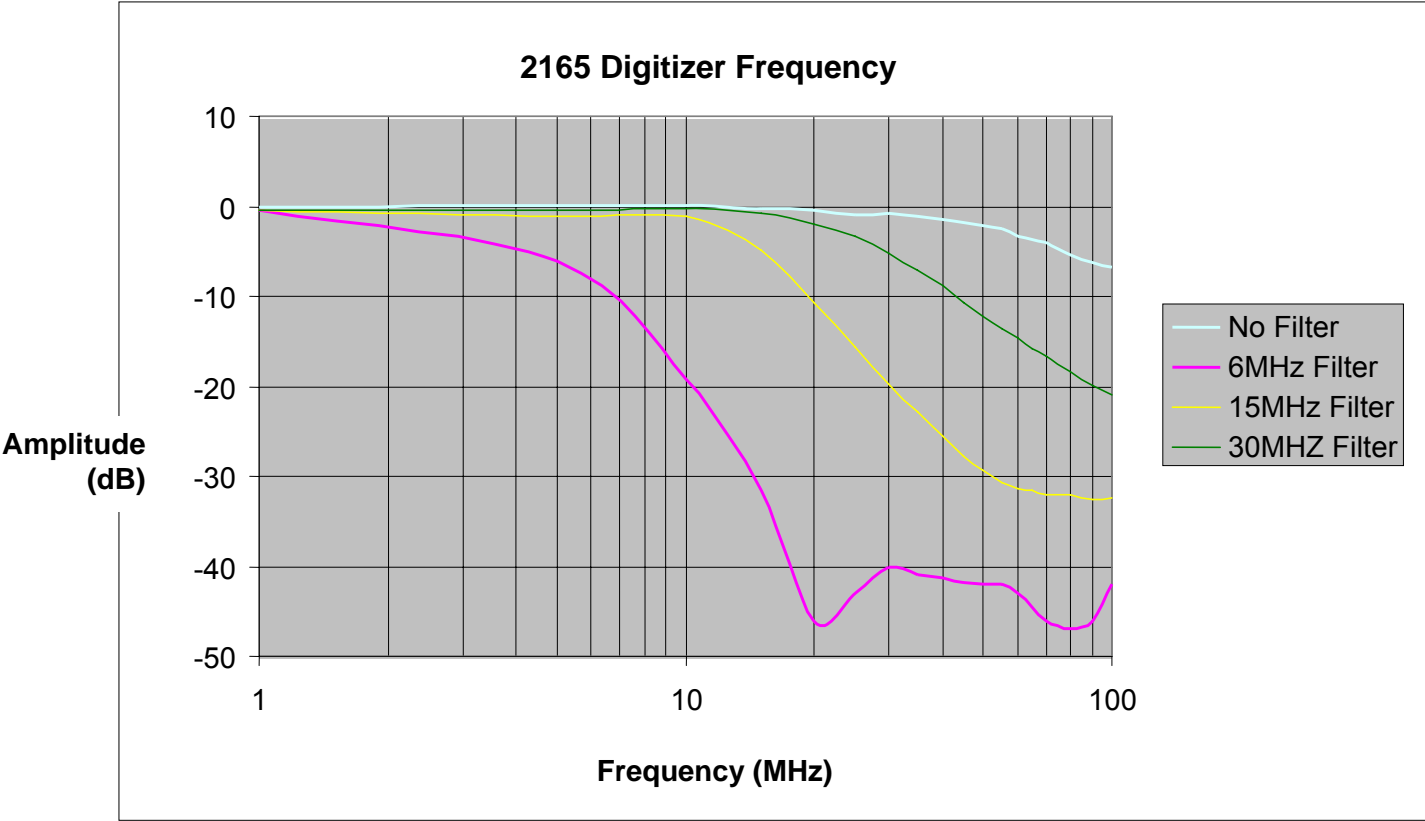


Figure 3-4, Typical Frequency Response

## Triggering

The 2165 digitizer starts data capture on a triggered signal. The separate trigger circuits from Channel-A and Channel-B allow each channel to start on different trigger sources and on different trigger edges or levels. The 2165 digitizer can trigger on three trigger sources: Digital input trigger, Software trigger and Analog trigger. **Figure 3-5** shows the trigger capabilities. To prevent triggering during connection or initialization there is a lock bit. The 2165 digitizer can be triggered once the lock bit is set.

## Digital Input Triggering

The digital trigger accepts triggers from the front panel trigger and the VXI back plane trigger sources. The front panel trigger input uses normal TTL logic levels, with a 0.5V nominal threshold for a low level and a 2V nominal threshold for a high level.

All the trigger inputs except the software trigger can handle different trigger signals. It is possible to trigger on positive or negative level or positive and negative edges.

For settings see the “SET TRIGGER” command in the DLL functions.

In level trigger mode the capturing starts when the trigger goes active and stops when the trigger goes inactive. Edge trigger mode has two options, normal or continuous. In normal mode the data capture starts at a trigger edge and stops on the next trigger edge. In continuous mode the measurement runs until stopped by the software. Every measurement can be stopped with the software by forcing the channel out of “lock mode”. When the module is out of lock mode the trigger register will be cleared.

### Software Triggering

If trigger timing is not a issue, there is a software initiated trigger. By writing the trigger register the data capture can be started or stopped.

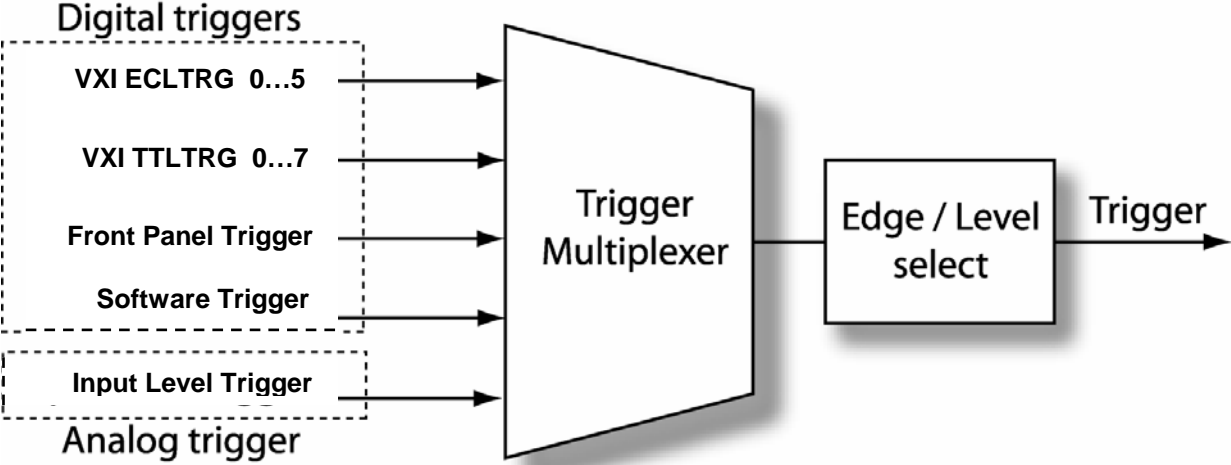


Figure 3-5, 2165 Trigger Capabilities



## Analog Triggering

In the analog trigger mode the analog input signal starts and stops the data capture. The trigger signal is extracted from the zero crossing from the input signal. (ADC code 4000 Hex).

If the analog input voltage is higher than 0 volt the trigger signal is high and if the analog input voltage is lower than 0 the trigger signal is low. Trigger at another analog level by adding an input offset voltage with the input offset DAC. With the level and edge trigger settings it is possible to capture different parts from the analog input signal. In the positive or negative level trigger mode data capture starts when the analog input signal is higher or lower than 0 volt. In this mode only the upper parts or the lower parts from the analog signal will be captured. (**Figure 3-6**) In edge trigger mode data capture starts at a trigger edge and stops on the next trigger edge. This edge can be negative or positive according to the setting. (**Figure 3-7**)

In continuous trigger mode the capturing starts on a positive or negative edge and runs until the memory stop address is reached (when not in loop mode) or stopped by the user in loop mode. (**Figure 3-8**)

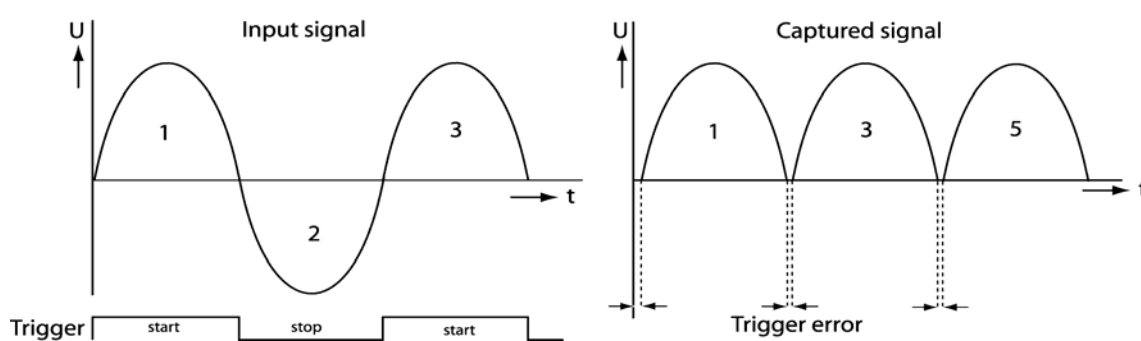


Figure 3-6, Analog Positive Level Triggering

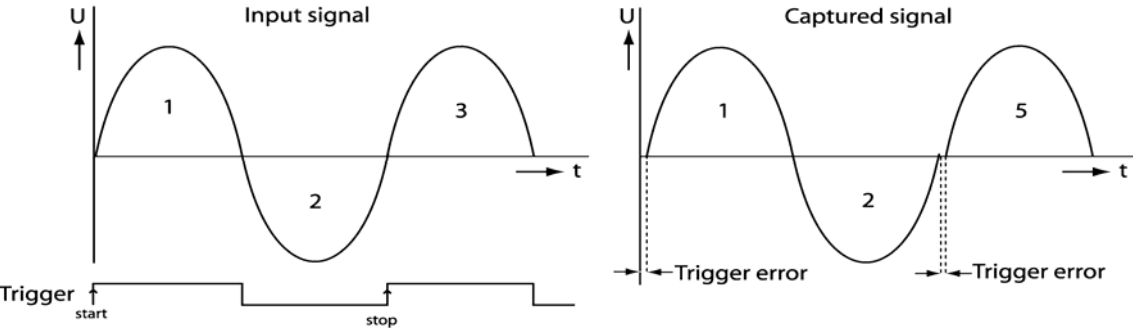


Figure 3-7, Analog Positive Edge Triggering

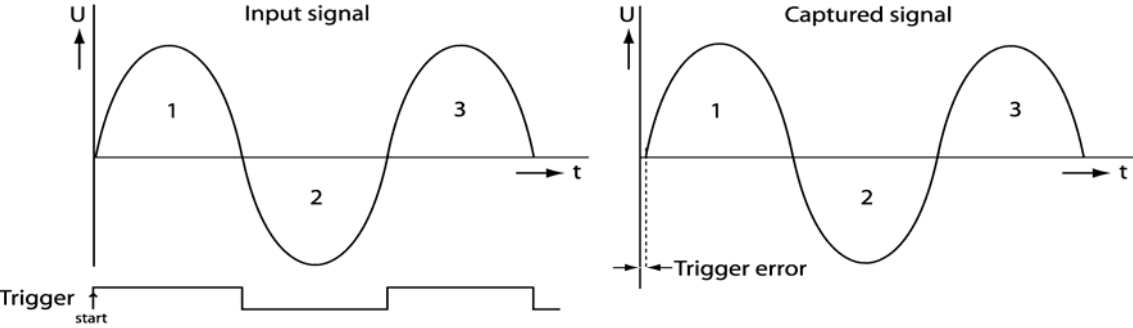


Figure 3-8, Analog Positive Continuous Edge Trigger

## Trigger Timing

The external trigger can come from the front panel trigger input or the VXI back plane trigger sources. The external trigger input uses normal TTL logic levels, with a 0.5V nominal threshold for a low level and a 2V nominal threshold for a high level. The trigger signal is related to the sample clock, and needs one sample clock cycle to clock into the trigger register. The trigger must be high 6ns before the rising edge from the sample clock to start or stop the measurement (See **Figure 3-9** and **Figure 3-10**). The ADC has a latency of 8 sample clock pulses, the first sample that will be saved after triggering is the sample taken 8 sample clocks before triggering. When the measurement is stopped with the trigger signal or by reaching the stop-address the last sample that will be saved is a sample taken eight sample clocks before the stop condition (See **Figure 3-9** and **Figure 3-10**).

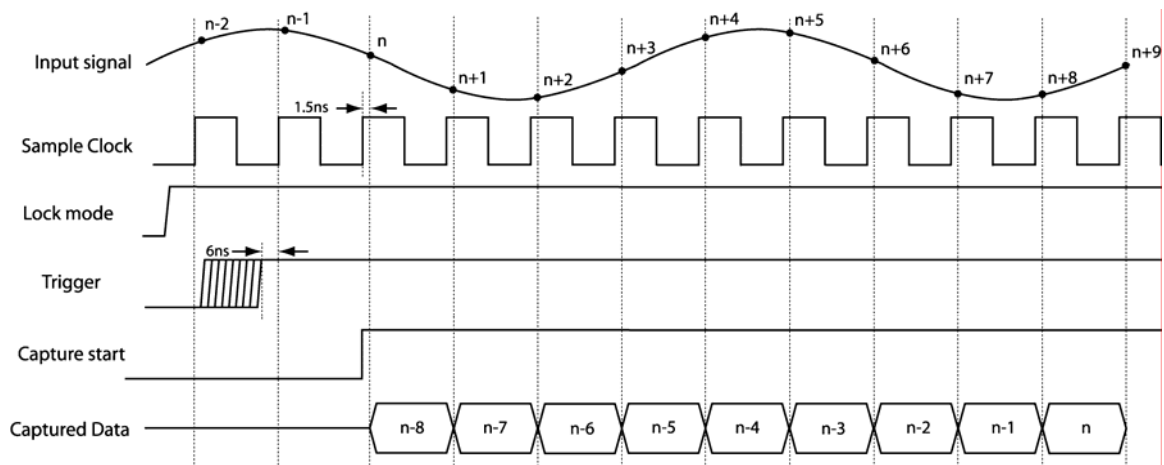


Figure 3-9, External Trigger Timing Start

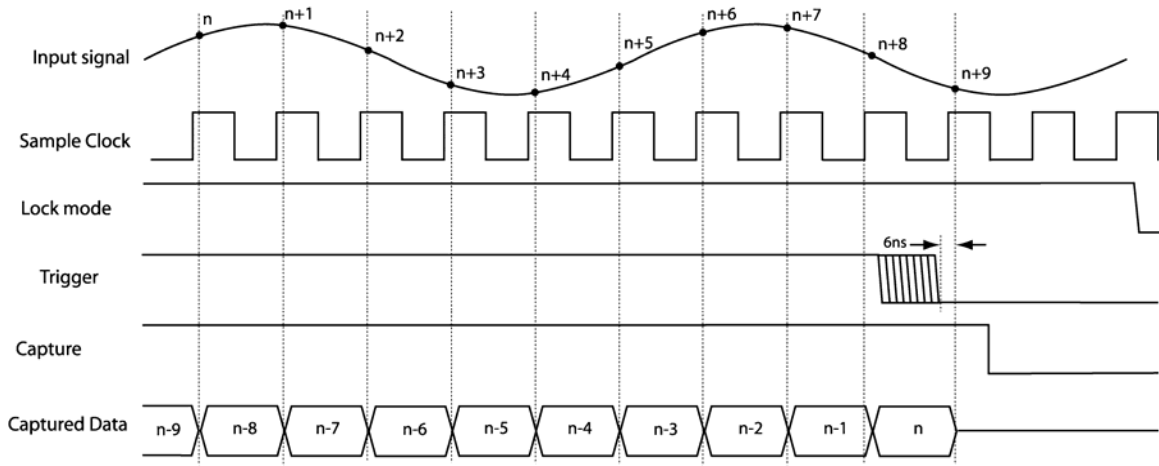


Figure 3-10, External Trigger Timing Stop

### Clcking

The 2165 digitizer can sample on different clock signals. The module has two internal clock sources, 70MHz and 50MHz and an external clock source input. To sample lower clock frequencies it is possible to divide the clock source to a lower frequency with the onboard divider. The divider can divide by a maximum of 256. Both channels have their own clock-divider, so they can capture data using different frequencies.

If an application requires sampling at specific intervals that cannot be achieved by using the internal 70MHz or 50 MHz clock, an external sample clock can be used. The external clock can be input via the SMB connector. When the front panel clock connector output is enabled, the sample clock from channel-A is present on this connector. This clock can be used to synchronize other devices in a measurement system via the 2165 digitizer. The front panel clock input and output are both 50-Ohm termination.

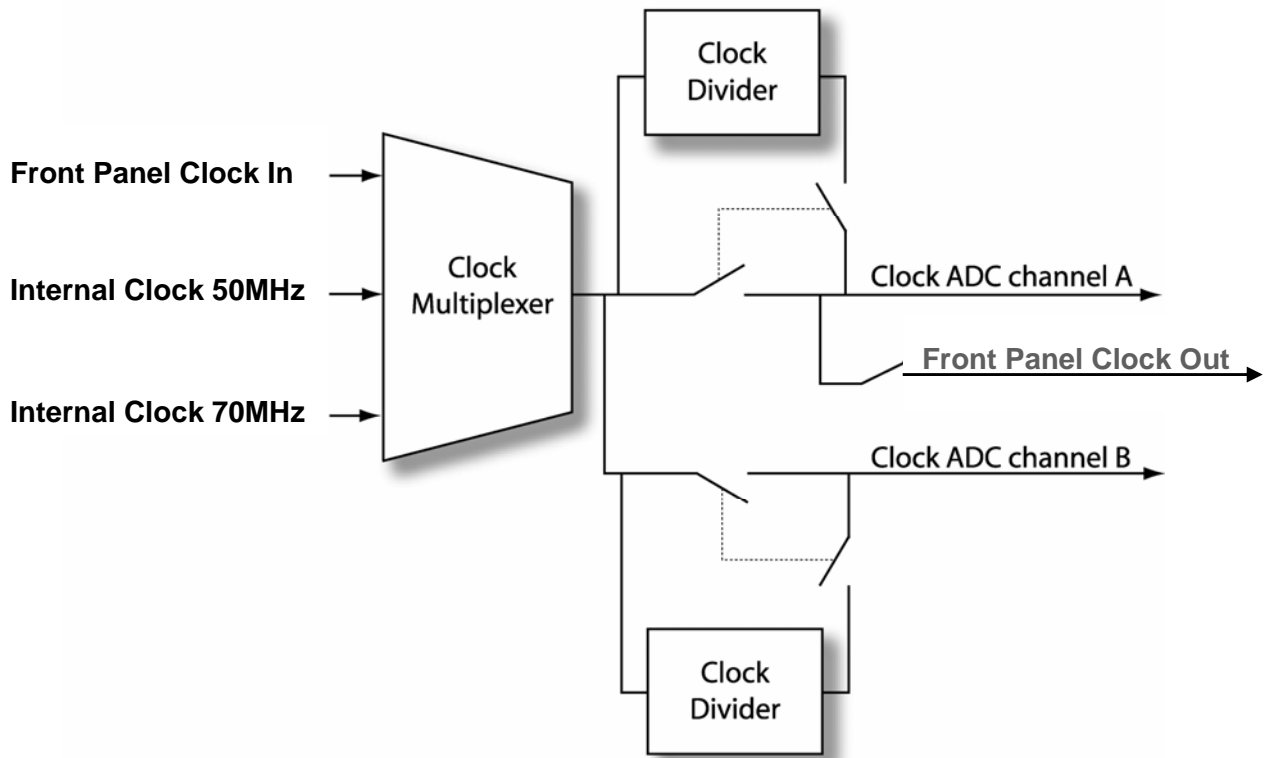


Figure 3-11, Clock Circuit

## Onboard Memory

The onboard memory is for captured data storage before transfer to the computer. This capture memory has 512k words available for each channel. The capture memory of the 2165 digitizer is addressed through a counter. This counter is active during data capture and during reading or writing the memory via the bus. In capture mode the counter starts counting from the start and increments on each sample clock until it reaches the end address from the memory ( 7FFFF Hex ). At the end of the memory the counter stops, or jumps to address 00000 Hex and counts up again. This selection is set with the loop mode-bit in the 2165 digitizer register, see (**Figure 3-12**). In this register there is also an address to see if the channel has reached the end address, the loop status register. See the **Register Assignment** for details. The start address should be written to the appropriate registers before the pattern starts. When writing the start address, this value is also loaded into the counter. In bus access mode the same mechanism is active except that the clock is now the read or write signal. To write a memory section the start address should be set first. The counter is now pointing to the start address and the content of this address can be read or written. After each read or write the counter increments to the next address allowing burst read or write actions. Note that the counter will stop or jump back to the start address when it reaches the end address, depending on the loop mode-bit in the PD register.

Since the memory may not be read or written during pattern generation, there is a lock bit that should be set to allow pattern generation. The memory is then locked for reading or writing. After setting the lock bit to unlock the memory is accessible again. A measurement that was running at that moment will be aborted.

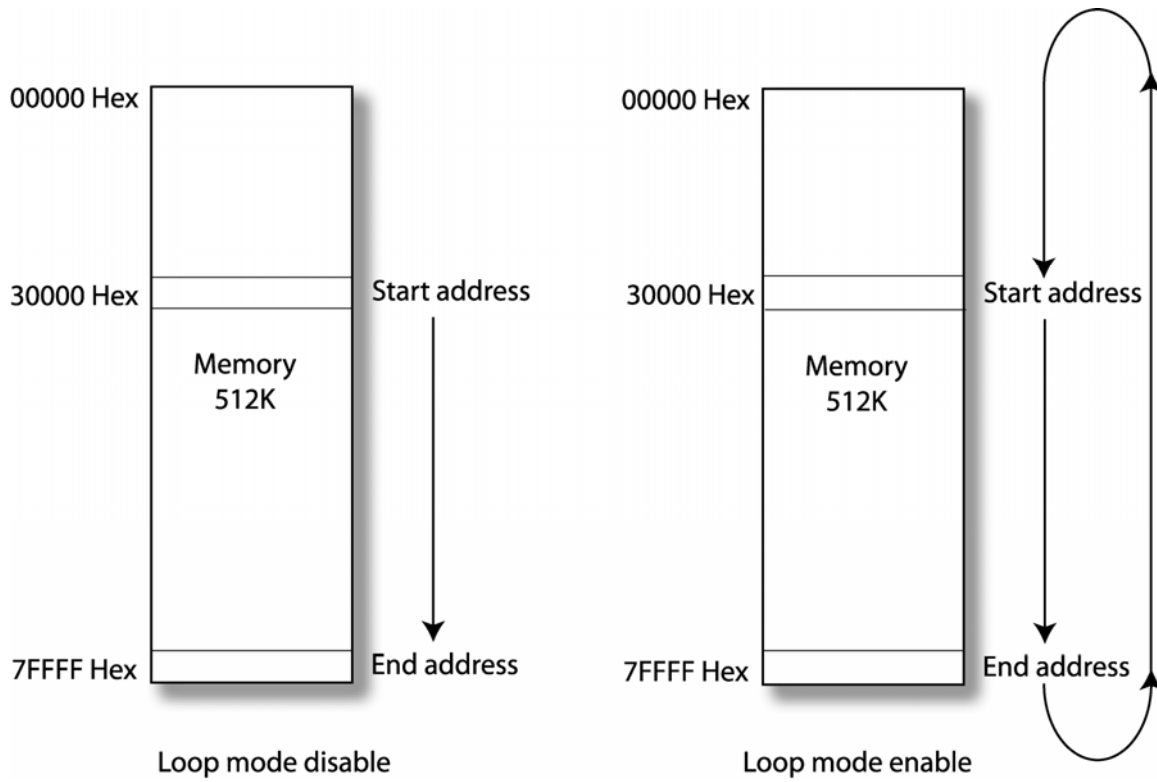


Figure 3-12, Function of the Loopmode-Bit

## Calibration

Calibration is a test that compares the values indicated by the 2165 digitizer with an external reference source. The result of a calibration is used to determine the gain and offset error so the 2165 digitizer can correct the error with the trim DAC's.

For optimum performance use self-calibration when the digitizer is placed in a new system or if the temperature changes more than 5°C from the previous calibration.

The maximum recommended amount of time between two calibrations is six months.

The calibration can be done with the software tool **2165 digitizer calibration** or with the DLL function **ri2165 digitizer\_AutoCalibrate**.

The input offset DAC is used as reference voltage source. Before the auto calibration is started the input offset DAC must be calibrated first, using a calibrated high-precision voltmeter connected to the negative input.

The 2165 digitizer uses trim DAC's to calibrate the offset and gain errors of the analog input channels. Once the calibration process is done, the calibration constant will be stored in EEPROM. These values can be loaded by the software, and used as needed by the board.

The software tool leads you step for step through the calibration process, and for details of the DLL function see the **2165 digitizer Software Manual**.

## VXI Interface

Supplied with the 2165 is a PC software package that consists of a windows driver, Labview driver and a demo program. Details on how to use this software are given in Chapter 4, 2165 Digitizer Software.



## Register Assignment

### Channel A

ADDRESS	Operation	DATA	Description
01 Hex	R and W	00000 : 3FFFF Hex	Read or write address counter
02 Hex	R and W	0000 : FFFF (auto increment)	Read or write to memory
04 Hex	W	00 : FF Hex	Write clock source divider (ADC)
05 Hex	R	00 Hex No channel has looped 01 Hex Channel A has looped 02 Hex Channel B has looped 03 Hex Channel A and B looped	Read loop status. Displays if the memory counter has looped one time.
06 Hex	R and W	00 Hex No trigger 01 Hex Trigger channel A 02 Hex Trigger channel B 03 Hex Trigger channel A and B	Write software trigger start. Read current trigger status.
07 Hex	W	00 Hex Front clock input 01 Hex Internal clock 70MHz 02 Hex Internal clock 50MHz 10 Hex Front clock out enable	Select Clock source and Clock output function.
08 Hex	R and W	01 Hex No filter 02 Hex 6MHz filter 04 Hex 15MHz filter 08 Hex 30MHz filter	Select Filter (Combine with Analog output select code)
08 Hex	R and W	10 Hex range 1 input = 1Vpp (x0.2 = 5Vpp) 20 Hex range 2 input = 2Vpp (x0.2 = 10Vpp) 40 Hex range 3 input = 4Vpp (x0.2 = 20Vpp)	Select Attenuator (Combine code with Filter select and input offset)
08 Hex	R and W	80 Hex Input offset on	Input offset on / off (combine with ATT)
09 Hex	R and W	01 Hex Connect input A+ 02 Hex Connect input A- 04 Hex Input A+ 50 Ohm 08 Hex Input A- 50 Ohm 10 Hex Input A+ to GND 20 Hex Input A- to GND 40 Hex input attenuation x 0.2 80 Hex Both inputs AC coupling	Analog output select
0A Hex	W	00 Hex Front trigger 08 Hex Software triggering 09 Hex Analog level Triggering	Select Trigger source (Combine with select Edge code)
0A Hex	W	00 Hex positive level triggering 10 Hex negative level triggering 20 Hex positive edge triggering 30 Hex negative edge triggering 60 Hex positive edge, continuous triggering 70 Hex negative edge, continuous triggering	Select trigger edge
0A Hex	W	80 Loop counter	Counter loop (0x) or one time (8x)
0B Hex	R and W	(bit0=Do/Di bit1=CLK bit3=CS)	Read Write EEPROM
0C Hex	W	(bit0=DO bit1=CLK bit3=CS)	Write Input offset DAC (16bit)
0D Hex	W	(bit0=DO bit1=CLK bit3=CS) 07 Hex cal offset channel A 08 Hex cal. gain channel A	Write Callibration offset and gain DAC (10bit)
0F Hex	W	00 Hex Lock off 01 Hex ready for trigger	Lock on or off

## Channel B

ADDRESS	Operation	DATA	Description
81 Hex	R and W	00000 : 3FFFF Hex	Read or write address counter
82 Hex	R and W	0000 : FFFF (auto increment)	Read or write to memory
84 Hex	W	00 : FF Hex	Write clock source divider (ADC)
85 Hex	R	00 Hex No channel has looped 01 Hex Channel A has looped 02 Hex Channel B has looped 03 Hex Channel A and B looped	Read loop status. Displays if the memory counter has looped one time.
86 Hex	R and W	00 Hex No trigger 01 Hex Trigger channel A 02 Hex Trigger channel B 03 Hex Trigger channel A and B	Write software trigger start. Read current trigger status.
87 Hex	W	00 Hex Front clock input 01 Hex Internal clock 70MHz 02 Hex Internal clock 50MHz 10 Hex Front clock out enable	Select Clock source and Clock output function.
88 Hex	R and W	01 Hex No filter 02 Hex 6MHz filter 04 Hex 15MHz filter 08 Hex 30MHz filter	Select Filter (Combine with Analog output select code)
88 Hex	R and W	10 Hex range 1 input = 1Vpp (x0.2 = 5Vpp) 20 Hex range 2 input = 2Vpp (x0.2 = 10Vpp) 40 Hex range 3 input = 4Vpp (x0.2 = 20Vpp)	Select Attenuator (Combine code with Filter select and input offset)
88 Hex	R and W	80 Hex Input offset on	Input offset on / off (combine with ATT)
89 Hex	R and W	01 Hex Connect input A+ 02 Hex Connect input A- 04 Hex Input A+ 50 Ohm 08 Hex Input A- 50 Ohm 10 Hex Input A+ to GND 20 Hex Input A- to GND 40 Hex input attenuation x 0.2 80 Hex Both inputs AC coupling	Analog output select
8A Hex	W	00 Hex Front trigger 08 Hex Software triggering 09 Hex Analog level Triggering	Select Trigger source (Combine with select Edge code)
8A Hex	W	00 Hex positive level triggering 10 Hex negative level triggering 20 Hex positive edge triggering 30 Hex negative edge triggering 60 Hex positive edge, continuous triggering 70 Hex negative edge, continuous triggering	Select trigger edge
8A Hex	W	80 Loop counter	Counter loop (0x) or one time (8x)
8B Hex	R and W	(bit0=Do/Di bit1=CLK bit3=CS)	Read Write EEPROM
8C Hex	W	(bit0=DO bit1=CLK bit3=CS)	Write Input offset DAC (16bit)
8D Hex	W	(bit0=DO bit1=CLK bit3=CS) 07 Hex cal offset channel A 08 Hex cal. gain channel A	Write Calibration offset and gain DAC (10bit)
8F Hex	W	00 Hex Lock off 01 Hex ready for trigger	Lock on or off

## 2165 DIGITIZER SOFTWARE

---

### Introduction

The 2165 is a 70MHz dual channel Waveform Digitizer. For software development and integration in a VXI system, the card is provided with a software driver, utility software, a demo program for LabView and a calibration tool.

The main part of the 2165 driver consists of a Windows dynamic link library, the ri2165\_32.dll

For Labview users the driver includes also a Labview library with the ri2165\_32.dll driver functions.

This chapter describes the functions of the ri2165\_32.dll.

### 2165 Driver Package

The driver includes the following items:

- 1) A low level driver for direct communication;
- 2) The user mode driver, ri2165\_32.dll;
- 3) A Labview library, ri2165.llb
- 4) A LabWindows driver (function tree) ri2165.fp

The following driver can be installed:

VISA from National Instruments.

### Installation

Install the 2165 Driver Software before the hardware is placed in the system.

Place the installation CD in the CD-ROM. If the installation program does not start automatically, run the program setup.exe (placed in the root of the CD-ROM).

If the software is installed on a Windows NT based operating system (Win2000, WinNT, WinXP), you should have Administrator rights.

After installation shutdown the computer and place the 2165 in the system. After turning on the computer the operation system should automatically detect the new hardware.

## 2165 DLL Functions

This chapter describes the functions of the dll. After the description follows a table with the necessary parameters belonging to the function. The following parameter types are used:

Type	Details
unsigned long	4-byte (32 bit) unsigned long
double	8-byte floating point
unsigned long*	reference variable (pointer) to a 4-byte (32 bit) unsigned long
double*	reference variable (pointer) to a 8-byte floating point

Every function returns the ri2165\_status (type: 32-bit integer). A negative value corresponds to an error. After a successful completion the return status is ri2165\_SUCCESS, which corresponds to a 0. The possible status codes can be found at the end of this Chapter

### ri2165\_AutoCalibrate (instrumentHandle, mode, displayTime)

#### Description:

This function performs an auto calibration procedure. The input DC-offset should already be calibrated. The calibration voltages can be monitored on the negative input of the module.

#### Parameters:

Name	Type	Direction	Description	Value
instrumentHandle	unsigned long	in	Instrument Handle	0 to 2 <sup>32</sup>
mode	unsigned long	in	mode	0 no monitoring 1 monitor at neg. input
displayTime	double	in	display time	time to display voltage at negative input

**ri2165\_CheckTestStatus (instrumentHandle, testStatus)****Description:**

This function returns the test status. It will return a 1 if the memory address counter passes the maximum address counter value (ready). Bit 0 represents channel A, bit 1 channel B.

**Parameters:**

Name	Type	Direction	Description	Value
instrumentHandle	unsigned long	in	Instrument Handle	0 to 2 <sup>32</sup>
testStatus	unsigned long*	out	Test Status	0 A & B not ready 1 Channel A ready 2 Channel B ready 3 A & B ready

**ri2165\_Close (instrumentHandle)****Description:**

Close the card session. All resources belonging to the card will be released.

**Parameters:**

Name	Type	Direction	Description	Value
instrumentHandle	unsigned long	in	card identifier	0 to 2 <sup>32</sup>

**ri2165\_CodeToVoltage (instrumentHandle, ADCCode, voltage)****Description:**

This function converts an ADC Code to the corresponding voltage.

**Parameters:**

Name	Type	Direction	Description	Value
instrumentHandle	unsigned long	in	Instrument Handle	0 to 2 <sup>32</sup>
ADCCode	unsigned long	in	mode	0 to 2 <sup>32</sup>
voltage	double*	out	voltage	corresponding voltage

**ri2165 \_ConnectCard  
(instrumentHandle,  
positiveInputConnection,  
negativeInputConnection )**

**Description:**

This function connects/disconnects the input relays. There are 4 relays for each input. Each relay can be controlled with a bit:

- Bit 1 (value 1 Hex) : Input relay;
- Bit 2 (value 2 Hex) : 50 Ohm relay;
- Bit 3 (value 4 Hex) : 50 Ohm DC relay, this bit controls the positive and negative channel simultaneously;
- Bit 4 (value 8 Hex) : Ground relay, connect input to ground.

**Parameters:**

Name	Type	Direction	Description	Value
instrumentHandle	unsigned long	in	Instrument Handle	0 to 2 <sup>32</sup>
positiveInputConnection	unsigned long	in	positive input	see description
negativeInputConnection	unsigned long	in	negative input	see description

**ri2165 \_GetActive Channel  
(instrumentHandle,  
channel )**

**Description:**

This function returns the active channel. The active channel is the channel, which can be configured with the corresponding driver functions. The channel can be set active with the function ri2165\_SetActiveChannel().

**Parameters:**

Name	Type	Direction	Description	Value
instrumentHandle	unsigned long	in	Instrument Handle	0 to 2 <sup>32</sup>
channel	unsigned long*	out	channel	1 = Channel A 2 = Channel B

**ri2165\_GetAddress  
Counter(instrument  
Handle, addresscounter )****Description:**

This function reads the current position of the memory address counter from the active channel. If the loopmode is active and the card is stopped capturing data, use this function to get the last captured data position.

**Parameters:**

Name	Type	Direction	Description	Value
instrumentHandle	unsigned long	in	Instrument Handle	0 to 2 <sup>32</sup>
addresscounter	unsigned long*	out	Position of the address-counter	0 to 2 <sup>19</sup>

**ri2165\_GetCardAddress  
(instrumentHandle,  
address )****Description:**

This address can be found in the Device Manager of Windows (under Resources). The card is installed under VISA, this function will return 0.

**Parameters:**

Name	Type	Direction	Description	Value
instrumentHandle	unsigned long	in	Instrument Handle	0 to 2 <sup>32</sup>
address	unsigned long *	out	address of card	0 to 2 <sup>32</sup>

**ri2165\_GetCard  
Connection (instrument  
Handle, positiveInput  
Connection, negativeInput  
Connection )****Description:**

Get the status of the input relays. This routine returns the connection status from the active channel. There are 4 relays for each input. Each relay can be controlled with a bit:

- Bit 1 (value 1 Hex) : Input relay;
- Bit 2 (value 2 Hex) : 50 Ohm relay;
- Bit 3 (value 4 Hex) : 50 Ohm DC relay, this bit controls the positive and negative channel simultaneously;
- Bit 4 (value 8 Hex) : Ground relay, connect input to ground.

**Parameters:**

Name	Type	Direction	Description	Value
instrumentHandle	unsigned long	in	Instrument Handle	0 to 2 <sup>32</sup>
positiveInputConnection	unsigned long*	out	positive input	see description
negativeInputConnection	unsigned long*	out	negative input	see description

**ri2165\_GetCard List  
(count, buslist, devicelist)**

**Description:**  
Use this function to retrieve the available arbitrary waveform generators.

**Parameters:**

Name	Type	Direction	Description	Value
count	unsigned long*	in	available card	0 to 255
buslist	long*	out	list with the bus numbers	0 to 255
devicelist	long*	out	list with the device numbers	0 to 255

**ri2165\_GetClock Divider  
(instrumentHandle,  
clockDivider)**

**Description:**  
This function returns the clock divider value.

**Parameters:**

Name	Type	Direction	Description	Value
instrumentHandle	unsigned long	in	Instrument Handle	0 to 2 <sup>32</sup>
clockDivider	unsigned long *	out	clock divider	1 to 16



**ri2165\_GetClock Source  
(instrumentHandle,  
clockSource)****Description:**

This function returns the clock source.

**Parameters:**

Name	Type	Direction	Description	Value
instrumentHandle	unsigned long	in	Instrument Handle	0 to 2 <sup>32</sup>
clockSource	unsigned long *	out	clock source	0: Extern clock front 1: Intern clock 70 MHz 2: Intern clock 50 MHz 3: VXI clock 10 MHz

**ri2165\_GetDCOffset  
LimitVoltages  
(instrumentHandle,  
positiveVoltage,  
negativeVoltage)****Description:**

Returns the DC Offset Limit Voltages. See function ri2165\_SetDCOffsetLimitVoltages(..).

**Parameters:**

Name	Type	Direction	Description	Value
instrumentHandle	unsigned long	in	Instrument Handle	0 to 2 <sup>32</sup>
positiveVoltage	double*	out	positive limit voltage	5 to 6
negativeVoltage	double*	out	negative limit voltage	-5 to - 6

**ri2165\_GetDCOffset  
Voltage(instrument  
Handle, voltage )****Description:**

This function will returns the current voltage of the DC-offset DAC (of the active channel). This voltage is previously set with the function ri2165\_SetDCOffsetVoltage().

**Parameters:**

Name	Type	Direction	Description	Value
instrumentHandle	unsigned long	in	Instrument Handle	0 to 2 <sup>32</sup>
voltage	double*	out	voltage of offset DAC	-5 to +5 volt

**ri2165\_GetError  
Message(code,  
message)**

**Description:**

This function translates an error code to an error message. The message buffer should be at least 256 bytes long.

**Parameters:**

Name	Type	Direction	Description	Value
code	unsigned long	in	error code	see chapter
message	char*	out	error message	see chapter

**ri2165\_GetFilter  
(instrumentHandle,  
filterStatus)**

**Description:**

This function returns the filter position.

**Parameters:**

Name	Type	Direction	Description	Value
instrumentHandle	unsigned long	in	Instrument Handle	0 to 2 <sup>32</sup>
filterStatus	unsigned long*	out	filter status	0: Disconnect 1: Bypass filter 2: 6 MHz filter 3: 15 MHz filter 4: 30 MHz filter

**ri2165\_GetGainCalCode  
(instrumentHandle,  
gainCalibrationCode)**

**Description:**

This function returns the calibration code for gain calibration DAC.

**Parameters:**

Name	Type	Direction	Description	Value
instrumentHandle	unsigned long	in	Instrument Handle	0 to 2 <sup>32</sup>
gainCalibrationCode	unsigned long*	out	gain cal. code	0 to 2 <sup>10</sup>

**ri2165\_GetInputVoltage  
(instrumentHandle,  
voltage, code,  
averages, timeOut)****Description:**

The function measures the input voltage Averages times and averages the voltage. It returns the voltage and the corresponding ADC code.

**Parameters:**

Name	Type	Direction	Description	Value
instrumentHandle	unsigned long	in	Instrument Handle	0 to 2 <sup>32</sup>
voltage	double*	out	measured voltage	depends on range
code	unsigned long*	out	adc code	0 to 2 <sup>14</sup>
average	unsigned long	in	averages	1 to 2 <sup>19</sup>
timeOut	unsigned long	in	time out	0 to 2 <sup>32</sup> no timeout it time out is 0

**ri2165\_GetLoopMod  
e (instrumentHandle,  
loopMode)****Description:**

This function returns the Loop Mode status.

**Parameters:**

Name	Type	Direction	Description	Value
instrumentHandle	unsigned long	in	Instrument Handle	0 to 2 <sup>32</sup>
loopMode	unsigned long*	out	loop mode status	0 or 1

**ri2165\_GetOffsetCalDac  
Code(instrument  
Handle, code)****Description:**

This function returns the offset calibration dac code.

**Parameters:**

Name	Type	Direction	Description	Value
instrumentHandle	unsigned long	in	Instrument Handle	0 to 2 <sup>32</sup>
code	unsigned long*	out	code	0 to 2 <sup>10</sup>

**ri2165\_GetRange**  
**(instrumentHandle,**  
**range)**

**Description:**

This function returns the range setting.

**Parameters:**

Name	Type	Direction	Description	Value
instrumentHandle	unsigned long	in	Instrument Handle	0 to 2 <sup>32</sup>
range	unsigned long*	out	code	1 to 6

**ri2165\_GetRevision**  
**(revision)**

**Description:**

This function returns the driver revision.

**Parameters:**

Name	Type	Direction	Description	Value
revision	unsigned long*	out	driver revision	1 to 2 <sup>32</sup>

**ri2165\_GetSample**  
**Divider**  
**(instrumentHandle,**  
**sampleDivider)**

**Description:**

The function returns the sample divider value.

**Parameters:**

Name	Type	Direction	Description	Value
instrumentHandle	unsigned long	in	Instrument Handle	0 to 2 <sup>32</sup>
sampleDivider	unsigned long*	out	sample divider	1 to 65536

**ri2165\_GetSoftware  
TriggerStatus(instrument  
Handle, trigger status )****Description:**

This function returns the trigger status.

**Parameters:**

Name	Type	Direction	Description	Value
instrumentHandle	unsigned long	in	Instrument Handle	0 to 2 <sup>32</sup>
triggerstatus	unsigned long *	out	current trigger status	0 = no channels triggered 1 = channel A triggered 2 = channel B triggered 3 = both channels triggered

**ri2165\_GetTriggerInput  
(instrumentHandle,  
triggerSource,  
triggerMode)****Description:**

This function returns the trigger source and trigger mode settings.

**Parameters:**

Name	Type	Direction	Description	Value
instrumentHandle	unsigned long	in	Instrument Handle	0 to 2 <sup>32</sup>
triggerSource	unsigned long*	out	trigger source	0 Front 8 Software 9 Level
triggerMode	unsigned long*	out	trigger mode	0 positive level 1 negative level 2 positive edge retrigger 3 negative edge retrigger 4 positive edge continuous 5 negative edge continuous

**ri2165\_Init( bus, device,  
instrumentHandle )****Description:**

Init a card with a VISA session. This function starts a card session. Call this routine if the card is installed with VISA. The bus and device number can be determined with the Measurement and Automation eXplorer (MAX) from National Instruments. This function returns a instrument handle (reference number), which is needed in most of the other functions to control the card. The function will return the same instrument handle if the function is called more than once, without calling ri2165\_Close() in between.

**Parameters:**

Name	Type	Direction	Description	Value
instrumentHandle	unsigned long*	out	Instrument Handle	0 to 2 <sup>32</sup>

**ri2165\_MemoryTest  
(instrumentHandle, level,  
errorAddress)****Description:**

This function performs a memory test.

**Parameters:**

Name	Type	Direction	Description	Value
instrumentHandle	unsigned long	in	Instrument Handle	0 to 2 <sup>32</sup>
level	unsigned long	in	level	0 = 1/10 of memory is tested 1 = total memory tested
errorAddress	unsigned long*	out	error address	if error, this parameter will contain the address which contains invalid data

**ri2165\_ReadAdcResults  
(instrumentHandle,  
startPosition, samples,  
buffer, include Status  
Bits)****Description:**

This function reads the latest test results from the 2165. The results are a 14 bit hexadecimal code. If the status bits are included the results are 16 bit words, where the two upper bits contain the status bits. Bit 14 represents the overflow bit and bit 15 (MSB) an extra capture bit.

**Parameters:**

Name	Type	Direction	Description	Value
instrumentHandle	unsigned long	in	Instrument Handle	0 to 2 <sup>32</sup>
startPosition	unsigned long	in	start position	0 to 2 <sup>19</sup>
samples	unsigned long	in	nr. of samples	1 to 2 <sup>19</sup>
buffer	unsigned long*	out	buffer for adc codes	
includeStatusBits	unsigned long	in	include the status bits	0 not included 1 included

**ri2165\_ReadEEPROM  
(instrumentHandle,  
eaddress, data)****Description:**

This function reads a 16-bit word from the serial EEPROM at a serial EEPROM address determined by eaddress. If the EEPROM address is previously written with the function ri2165\_WriteEEPROM(), the lower byte will correspond to the byte written with the function ri2165\_WriteEEPROM(). The upper byte will be the complement of the lower byte. This byte can be used for verifying purposes. One on board EEPROM is used for both channels.

**Parameters:**

Name	Type	Direction	Description	Value
instrumentHandle	unsigned long	in	Instrument Handle	0 to 2 <sup>32</sup>
eaddress	unsigned long	in	EEPROM address	0 to 255
data	unsigned long*	out	read data	0 to 2 <sup>16</sup>

### ri2165\_ReadId (instrumentHandle, id )

#### Description:

This function reads the card ID. A card ID can be set with the function ri2165\_WriteId(). The card ID is placed in the on board serial EEPROM.

#### Parameters:

Name	Type	Direction	Description	Value
instrumentHandle	unsigned long	in	Instrument Handle	0 to 2 <sup>32</sup>
id	unsigned long*	out	card id	0 to 2 <sup>32</sup>

### ri2165\_ReadRam (instrumentHandle, data )

#### Description:

Read from the (stimuli) ram of the active channel. The ram address is determined by the address-counter. The address-counter can be initialized with the function ri2165\_SetAddressCounter(). After this function call the address-counter is incremented with one step.

#### Parameters:

Name	Type	Direction	Description	Value
instrumentHandle	unsigned long	in	Instrument Handle	0 to 2 <sup>32</sup>
data	unsigned long*	out	read data	0 to 2 <sup>16</sup>

### ri2165\_ReadRam Buffer(instrumentHandle, length, buf32, data Interpretation )

#### Description:

Read length ram-places from the stimuli ram (of the active channel), starting from the current address-counter value. The ram address-counter can be initialized with the function ri2165\_SetAddressCounter(). After this function call the address-counter is incremented with "length" steps. Ram data can be interpret as test (adc) results. In this case the ram data will be shifted 2 places (right shift) and the upper bit will be inverted (adc digital data is 2's complement).



**Parameters:**

<b>Name</b>	<b>Type</b>	<b>Direction</b>	<b>Description</b>	<b>Value</b>
instrumentHandle	unsigned long	in	Instrument Handle	0 to 2 <sup>32</sup>
length	unsigned long	in	number of data elements to be read	0 to 2 <sup>18</sup>
buf32	unsigned long*	out	reference to a buffer for read data	0 to 2 <sup>32</sup>
dataInterpretation	unsigned long	in	data interpretation	0 raw data from ram 1 adc codes only

**ri2165\_SetActiveChannel(instrument Handle, channel )****Description:**

Select the active channel. This function selects the channel to be updated.

**Parameters:**

<b>Name</b>	<b>Type</b>	<b>Direction</b>	<b>Description</b>	<b>Value</b>
instrumentHandle	unsigned long	in	Instrument Handle	0 to 2 <sup>32</sup>
channel	unsigned long	in	active channel	1 Channel-A 2 Channel-B

**ri2165\_SetAddress Counter (instrument Handle, address CounterPosition)****Description:**

This function programs the address counter. Use this function to read from or write to a specified address.

**Parameters:**

<b>Name</b>	<b>Type</b>	<b>Direction</b>	<b>Description</b>	<b>Value</b>
instrumentHandle	unsigned long	in	Instrument Handle	0 to 2 <sup>32</sup>
addressCounterPosition	unsigned long	in	address counter position	0 to 2 <sup>19</sup>

**ri2165\_SetClock  
Divider(instrument  
Handle, clock  
divider )**

**Description:**

Set the clock-divider. Programs the divider for the sample clock of the active channel. The clock divider divides the clock selected with ri2165\_SetClockSource() and determines the sample rate.

**Parameters:**

Name	Type	Direction	Description	Value
instrumentHandle	unsigned long	in	Instrument Handle	0 to 2 <sup>32</sup>
clockdivider	unsigned long	in	clock divider value	1 to 16

**ri2165\_SetClock  
Source(instrument  
Handle, clock  
source )**

**Description:**

Select the desired clock source (for the selected channel). The clock source and the clock divider determine the update rate of the output signal. See also ri2165\_SetClockDivider().

**Parameters:**

Name	Type	Direction	Description	Value
instrumentHandle	unsigned long	in	Instrument Handle	0 to 2 <sup>32</sup>
clocksource	unsigned long	in	select clock source	0 = FRONT PANEL CLK 1 = INT CLK 1 (100MHz) 2 = INT CLK 2 (70MHz) 3 = VXI 10MHz CLK

**ri2165\_SetDCOffset  
Code(instrument  
Handle, code,  
connect )**

**Description:**

Write a code to the DC offset DAC (of the selected channel). This functions programs the 16-bit offset DAC with the desired code. This function can be used for calibration purposes. In normal operation the function ri2165\_SetDCOffsetVoltage() will program the DC offset voltage to a desired level. If connect is 1 offset voltage is connected to the negative input.

**Parameters:**

Name	Type	Direction	Description	Value
instrumentHandle	unsigned long	in	Instrument Handle	0 to 2 <sup>32</sup>
code	unsigned long	in	code for DC offset DAC	0 to 2 <sup>16</sup>
connection	unsigned long	in	connect	0 disconnect from negative input 1 connect to negative input

**ri2165\_SetDCOffsetVoltage(instrumentHandle, voltage, connect )**

**Description:**

Program the DC offset voltage DAC (of the selected channel) with the desired voltage. The DC offset voltage can be a voltage between the -5V and +5V.

**Parameters:**

Name	Type	Direction	Description	Value
instrumentHandle	unsigned long	in	Instrument Handle	0 to 2 <sup>32</sup>
voltage	double	in	voltage for offset DAC	-5 to +5V
connect	unsigned long	in	connect	0 disconnect 1 connect dc offset, disconnect negative input 2 connect dc offset & negative input

**ri2165\_SetDCOffsetLimitVoltages(instrumentHandle, posvolt, negvolt )**

**Description:**

Set limit voltages of dc offset DAC (of the selected channel). These voltages are necessary for a calibrated offset voltage.

Procedure to determine limit voltages:

- Disconnect card: ri2165\_ConnectCard(instrumentHandle, 0, 0)
- Filter bypass: ri2165\_SetFilter(instrumentHandle, 0)
- Program DC offset dac at maximal voltage and connect offset voltage with input: ri2165\_SetDCOffsetCode (instrumentHandle, 0xFFFF,1)

- Measure DC offset voltage with accurate voltage meter
- Program DC offset dac at minimal voltage and connect offset voltage with input:  
ri2165\_SetDCOffsetCode(instrumentHandle,0x0,1)
- Measure DC offset voltage with accurate voltage meter
- Disconnect offset voltage: ri2165\_SetDCOffsetCode (instrumentHandle, 0x8000,0)
- Call this routine with measured voltages
- Call ri2165\_StoreCalibration for storing data in EEPROM

This routine can be called to store just one of the limit voltages. Fill in a voltage < 5V for the PositiveVoltage or a voltage > -5V for the Negative Voltage and the corresponding voltage will not be stored.

**Parameters:**

Name	Type	Direction	Description	Value
instrumentHandle	unsigned long	in	Instrument Handle	0 to 2 <sup>32</sup>
posvolt	double	in	measured positive voltage	> 5V
negvolt	double	in	measured negative voltage	< -5V

**ri2165\_SetFilter  
(instrumentHandle,  
filter )**

**Description:**  
Select a desired filter path for the active channel.

**Parameters:**

Name	Type	Direction	Description	Value
instrumentHandle	unsigned long	in	Instrument Handle	0 to 2 <sup>32</sup>
filter	unsigned long	in	filter select	0 = Disconnect 1 = Bypass 2 = 6 MHz filter 3 = 15 MHz filter 4 = 30 MHz filter

**ri2165\_SetGainCal  
Code(instrument  
Handle, code)****Description:**

Write a code to the gain calibration dac.

**Parameters:**

Name	Type	Direction	Description	Value
instrumentHandle	unsigned long	in	Instrument Handle	0 to 2 <sup>32</sup>
code	unsigned long	in	code for gain cal. dac	0 to 2 <sup>10</sup>

**ri2165\_SetLock  
Mode(instrument  
Handle, lock )****Description:**

Lock or unlock the memory access for active channel. A channel should be locked before the channel can be used to capture a signal. The memory cannot be accessed (by a controller) and the channel waits for a trigger in this mode. If the channel is unlocked the channel does not respond to a trigger signal and the memory can be accessing by a controller.

**Parameters:**

Name	Type	Direction	Description	Value
instrumentHandle	unsigned long	in	Instrument Handle	0 to 2 <sup>32</sup>
lock	unsigned long	in	lock/unlock active channel	1 = lock 0 = unlock

**ri2165\_SetLoop  
Mode(instrument  
Handle, loopMode)****Description:**

This function sets the loop mode. If not in loop mode the capturing of data stops if address counter is at the end.

**Parameters:**

Name	Type	Direction	Description	Value
instrumentHandle	unsigned long	in	Instrument Handle	0 to 2 <sup>32</sup>
loopmode	unsigned long	in	loopmode	0 = not in loopmode 1 = loopmode

**ri2165\_SetOffset**      **Description:**  
**CalCode(instrument**  
**Handle, code)**      This function programs the offset calibration dac.

Parameters:

Name	Type	Direction	Description	Value
instrumentHandle	unsigned long	in	Instrument Handle	0 to 2 <sup>32</sup>
code	unsigned long	in	code	0 to 2 <sup>10</sup>

**ri2165\_SetRange**      **Description:**  
**(instrumentHandle,**  
**range )**      This function will set the input voltage range.

Parameters:

Name	Type	Direction	Description	Value
instrumentHandle	unsigned long	in	Instrument Handle	0 to 2 <sup>32</sup>
range	unsigned long	in	input range	1: -0.5 to + 0.5V 2: -1 to + 1V 3: -2 to + 2V 4: -2.5 to + 2.5V 5: -5 to + 5V 6: -10 to + 10V

**ri2165\_SetSample**      **Description:**  
**Divider(instrument**  
**Handle, sample**  
**DividerValue)**      This function will program the sample divider. The sample divider determines how many samples will be stored. For example, if the sample divider is 2, 1 of the 2 samples will be stored in the capture ram. Programming the sample divider will actually lower the sample rate, while the adc sample clock is not lowered.

Parameters:

Name	Type	Direction	Description	Value
instrumentHandle	unsigned long	in	Instrument Handle	0 to 2 <sup>32</sup>
sampleDividerValue	unsigned long	in	sample divider	1 to 65536

**ri2165\_Set Software  
TriggerStatus  
(instrumentHandle,  
triggerstatus )**

**Description:**

Trigger (start capturing) or stop the channel(s). This function enables the software to trigger and stop the channel(s). Select Software Trigger with the function ri2165\_SetTriggerMode() to enable the software trigger mode.

**Parameters:**

Name	Type	Direction	Description	Value
instrumentHandle	unsigned long	in	Instrument Handle	0 to 2 <sup>32</sup>
triggerstatus	unsigned long	in	trigger status	0 inactive trigger 1 trigger channel A 2 trigger channel B 3 trigger both channels

**ri2165\_Set Trigger  
Input(instrument  
Handle,trigger  
source,triggermode )**

**Description:**

Select trigger source and trigger mode. In lock mode (Set with the function ri2165\_SetLockMode() ) the signal capturing can be started (triggered) by the selected trigger source.

**Parameters:**

Name	Type	Direction	Description	Value
instrumentHandle	unsigned long	in	Instrument Handle	0 to 2 <sup>32</sup>
triggersource	unsigned long	in	select trigger source	0 = FRONT PANEL TRIG 8 = SOFTWARE TRIG 9 = Level
triggermode	unsigned long	in	select trigger mode	0 = positive level 1 = negative level 2 = positive edge (re-trigger) 3 = negative edge (re-trigger) 4 = positive edge (continuous) 5 = negative edge (continuous)

**ri2165\_Store  
CalibrationData  
(instrumentHandle )**

**Description:**

Store calibration data (of both channels) in serial EEPROM. This function should be called after a calibration procedure to store the calibration data in the on board serial EEPROM.

**Parameters:**

Name	Type	Direction	Description	Value
instrumentHandle	unsigned long	in	Instrument Handle	0 to 2 <sup>32</sup>

**ri2165\_WriteEPRO** Description:**M****(instrumentHandle,  
eaddress, data )**

Write a data byte to the EEPROM address (defined with eaddress) of the serial EEPROM. An EEPROM address has place for 2 bytes (16 bits). With this function the upper byte will be filled with the complement of the lower byte. This byte can be used for verifying purposes during reading. With this function ALL EEPROM addresses can be written! So the calibration data and module ID can be changed with this function! Till EEPROM address 161 are reserved for calibration data and the module ID.

**Parameters:**

Name	Type	Direction	Description	Value
instrumentHandle	unsigned long	in	Instrument Handle	0 to 2 <sup>32</sup>
eaddress	unsigned long	in	EEPROM address	0 to 255
data	unsigned long	in	byte to be written	0 to 255

**ri2165\_Writelid** Description:**(instrumentHandle,  
id )**

This function writes a card ID in the serial EEPROM. The card ID may be any 32 bit value. The card ID can be read with the function ri2165\_Readlid().

**Parameters:**

Name	Type	Direction	Description	Value
instrumentHandle	unsigned long	in	Instrument Handle	0 to 2 <sup>32</sup>
id	unsigned long	in	card id	0 to 2 <sup>32</sup>



**ri2165\_WriteRam  
(instrumentHandle,  
data )****Description:**

Write to the (capture) ram of the active channel. The ram address is determined by the address-counter. The address-counter can be initialized with the function ri2165\_SetAddressCounter(). After this function call the address-counter is incremented with one step.

**Parameters:**

Name	Type	Direction	Description	Value
instrumentHandle	unsigned long	in	Instrument Handle	0 to 2 <sup>32</sup>
data	unsigned long	in	data to write	0 to 2 <sup>14</sup>

**ri2165\_WriteRam  
Buffer(instrument  
Handle, length,  
buffer )****Description:**

Write a buffer with length (32 bit) words to the (capture) ram of the active channel, starting at the current address counter position. The address-counter can be initialized with the function ri2165\_SetAddressCounter(). After this function call the address-counter is incremented with "length" steps.

**Parameters:**

Name	Type	Direction	Description	Value
instrumentHandle	unsigned long	in	Instrument Handle	0 to 2 <sup>32</sup>
length	unsigned long	in	number of data elements to be written	0 to 2 <sup>18</sup>
buffer	unsigned long*	in	reference to buffer with data to be written	0 to 2 <sup>32</sup>

## Status Codes

This chapter will give an overview of the possible status codes that can be returned by the dll-functions.

Completion without error:

Constant name	Value	Description
ri2165_SUCCESS	0x0	No error(s)

General error codes:

Constant name	Value	Description
ri2165_ERROR_INVALID_CHANNEL	0xBFFE0004	Invalid channel
ri2165_ERROR_INVALID_PARAMETER	0xBFFE0005	Invalid parameter
ri2165_ERROR_MEMORY	0xBFFE0006	Could not allocate memory
ri2165_ERROR_NO_SIGNALDEF	0xBFFE0007	No signal defined
ri2165_ERROR_EEPROMCHECK	0xBFFE0008	EEPROM verify error
ri2165_ERROR_MEMTEST	0xBFFE000B	Memory test failed
ri2165_ERROR_CALIBRATION	0xBFFE000C	Auto calibration error
ri2165_ERROR_TIMEOUT	0xBFFE000D	Time-out expired

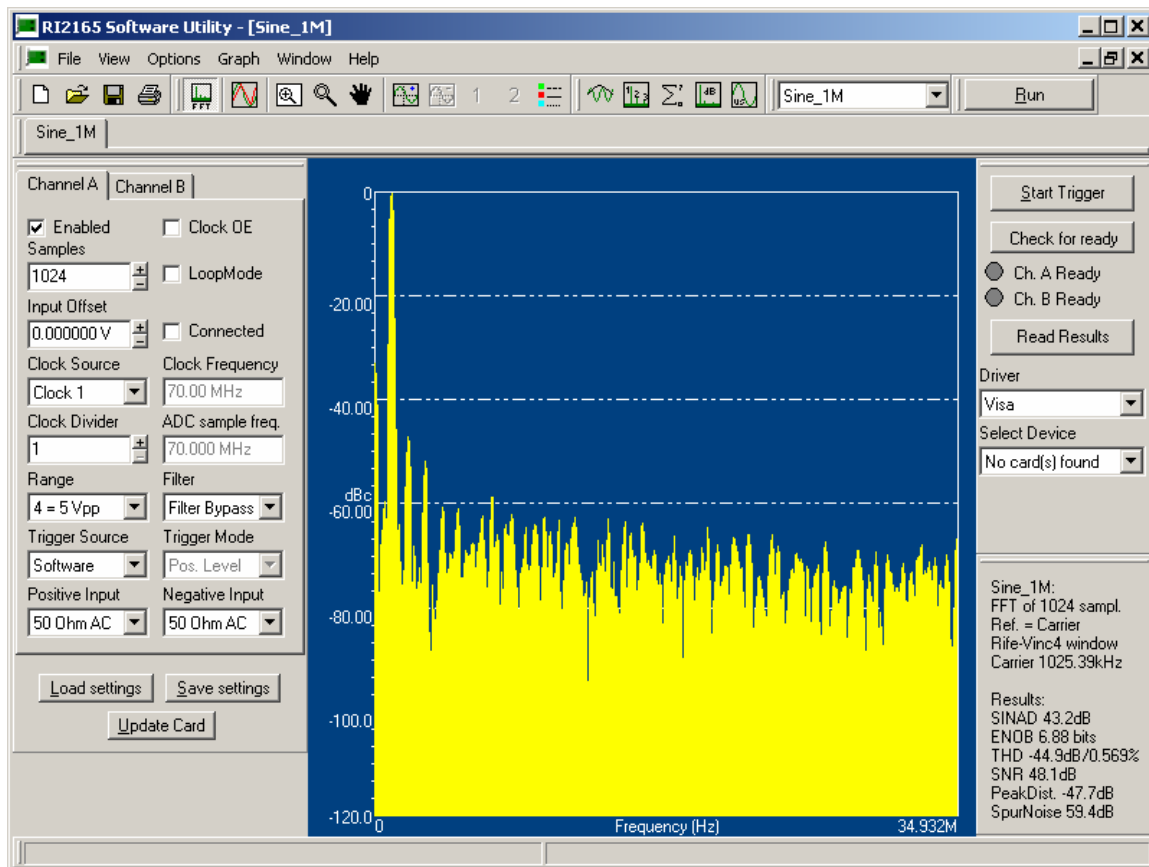
VISA error codes:

For the VISA completion codes and error codes, please read the NI-VISA programmer reference.

# 2165 DIGITIZER SOFTWARE UTILITY

## Introduction

The 2165 Software Utility can be used to adjust the 2165 digitizer and analyze the captured data.



The Software Utility includes the following features:

- Adjust the 2165 instrument;
- Do a software trigger to start data capture;
- Check the test status;
- Analyzing the results by using the Fast Fourier Transform or Discrete Fourier Transform algorithm.

Some analyzing features:

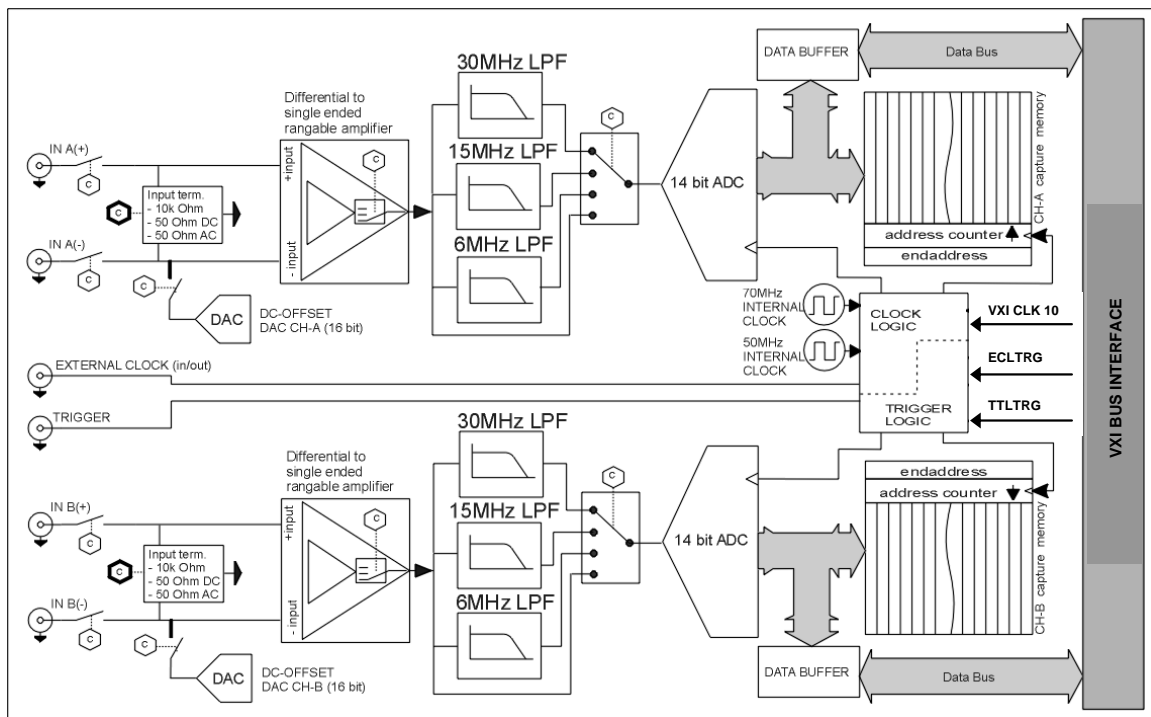
- 8 different windowing types (e.g. Hanning, Hamming, Flat-Top, Blackman-Harris and Rife Vincent);
- Summary of Harmonics;
- Adjustable harmonics level and count level for parameter calculations;
- Adjustable noise level for parameter calculations;

Fixed reference level (dBc) or adjustable reference level (dB);

# Adjusting the Instrument

## Block Diagram

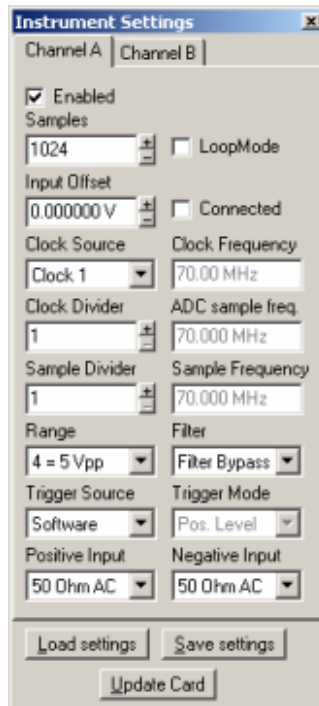
It is easier to understand the instrument settings if the hardware structure is known:



For details, please read the Hardware User Manual 2165

## The Instrument Settings Panel

The instrument can be adjusted with the following panel:



The instrument can be configured by adjusting the following parameters:

- **Enabled:** Determines if the channel will be updated and set in lockmode.
- **Samples:** Number of samples that will be read and can be analyzed.
- **LoopMode:** Set channel in loopmode. If channel is in loopmode the data will be captured as long as the channel is triggered and in lockmode. When the data is read, the software starts reading “samples” before the last captured sample. If the channel is not in loopmode data capture stops at the end of the capture memory. When the data is read, the reading starts at the first captured sample after the trigger.
- **Input offset:** Set the DC input offset voltage at a specified level.
- **Connected:** Connect or disconnect the DC input offset.

- **Clock source:** Select the clock source.
- **Clock frequency:** Can only be adjusted if the External clock source is selected. Please fill in the supplied clock frequency for a correct time and frequency axis.
- **Clock divider:** The clock divider for the desired (ADC) sample frequency.
- **Sample divider:** The sample divider for the desired sample frequency. The sample divider will not change the ADC sample frequency. E.g. a sample divider of 2: 1 of the 2 samples will be stored in the capture memory.
- **Range:** Select the input voltage range.
- **Filter:** Select a filter or bypass the filter.
- **Trigger source:** Select the trigger source. If using the “Start” button of the software utility, select Software-trigger.
- **Trigger mode:** Select if the trigger mode.
- **Positive and Negative input:** Select the input connection.

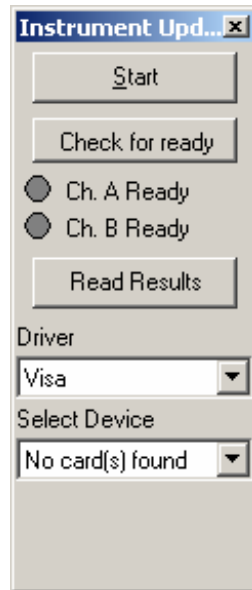
The load and save buttons can be used to load and store predefined settings. Use the “Update Card” button to update the Instrument.

## Performing a Test

Perform the following steps before the test:

1. Install the VISA driver;
2. Select the Device;
3. Adjust the instrument settings;
4. Press button “Update Card”;
5. Press the “Start” button in case of software trigger;
6. Press the “Check for ready” button and wait till the enabled channel(s) is/are ready;

Press the “Read Results” button. The result will be displayed in the active window.



It is possible to run a test by pressing the “Run” button on the toolbar. This will perform the steps 4 till 7 automatically.

## Analyzing the Data

### General

Some general tools can be found in the Graph Toolbar. These options can also be found in the Menu Graph.



**FFT:** Do a Fast Fourier analysis (if samples are a power of 2) or discrete Fourier analyses (in all other cases).

**Overlay:** Place more than one graph in the same window.

**Undo zoom:** Graph will be displayed in original position.

**Zoom:** Zoom in/out or area. For an accurate zoom use the Options->Time Domain Settings dialog.

**Scroll:** Scroll the graph.

**Add cursor:** Add a cursor to the active line.



**Remove cursor:** Remove the active cursor.

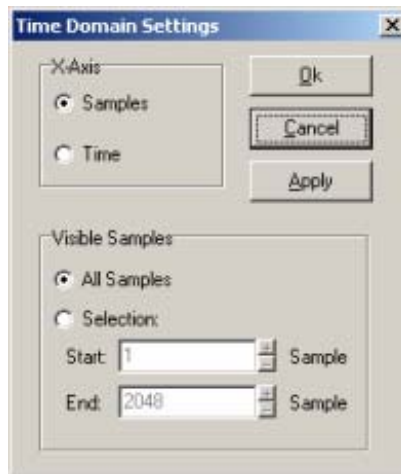
**Activate cursor 1 or 2:** If the cursor is the active cursor, the cursor can also be move with the arrow keys.

**Graph colors:** Change the graph colors.

The graph can be saved as a bitmap (Graph->Save Graph As Bitmap) and can be copied to the Clipboard (Graph->Copy Graph To Clipboard).

## Time Domain

Select “Time Domain Settings” from the Option menu or toolbar. The X-axis can display samples or a time axis. For an accurate zoom select the option “Selection” of Visible Samples.

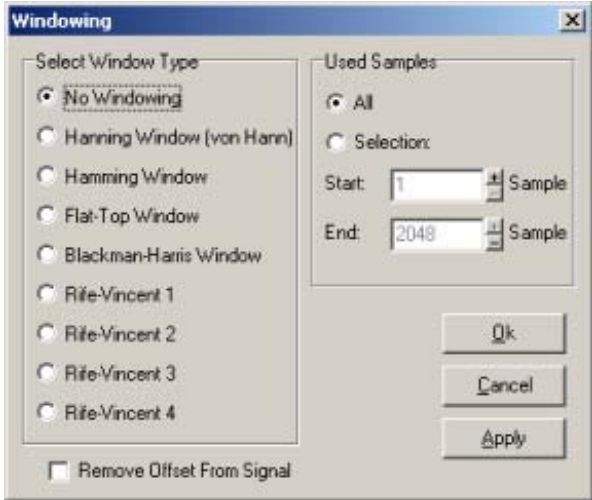


## Frequency Domain

### Windowing

There are two reason why you would apply a window:

1. Your signal has a non-integer number of periods. Analyzing the signal in the frequency domain will result in spectral leakage. Applying a window reduces the spectral leakage.
2. You want to analyze a part of the signal.



The general equation of the window-function is:

$$Wk = \sum_{m=0}^{m=M-1} a_m \cos\left(\frac{2\pi k}{N} m\right) \quad 0 \leq k < N$$

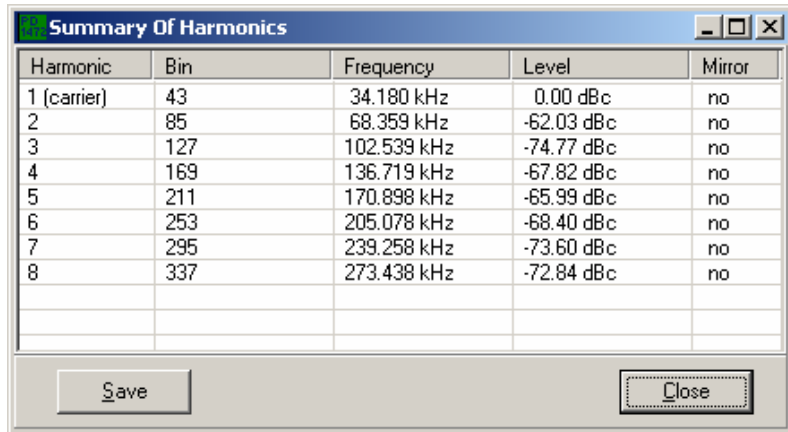
The coefficients for each window are:

Window type	a0	a1	a2	a3	a4
No window (also Rectangular)	1.0	0.0	0.0	0.0	0.0
Hanning	0.5	-0.5	0.0	0.0	0.0
Hamming	0.54	-0.46	0.0	0.0	0.0
Flat-Top	0.281	-0.521	0.198	0.0	0.0
Blackman-Harris	0.432	-0.497	0.079	0.0	0.0
Rife Vincent 1	0.5	-0.5	0.0	0.0	0.0
Rife Vincent 2	0.375	-0.5	0.125	0.0	0.0
Rife Vincent 3	0.3125	-0.46875	0.1875	-0.03125	0.0
Rife Vincent 4	0.2734375	-0.4375	0.21875	-0.0625	0.0078125

In the "Used samples" section you can select a part of the signal to be analyzed.

**Summary of Harmonics**

This data for the summary of harmonics is only available if the signal is displayed in the frequency domain.



Harmonic	Bin	Frequency	Level	Mirror
1 (carrier)	43	34.180 kHz	0.00 dBc	no
2	85	68.359 kHz	-62.03 dBc	no
3	127	102.539 kHz	-74.77 dBc	no
4	169	136.719 kHz	-67.82 dBc	no
5	211	170.898 kHz	-65.99 dBc	no
6	253	205.078 kHz	-68.40 dBc	no
7	295	239.258 kHz	-73.60 dBc	no
8	337	273.438 kHz	-72.84 dBc	no

Save Close

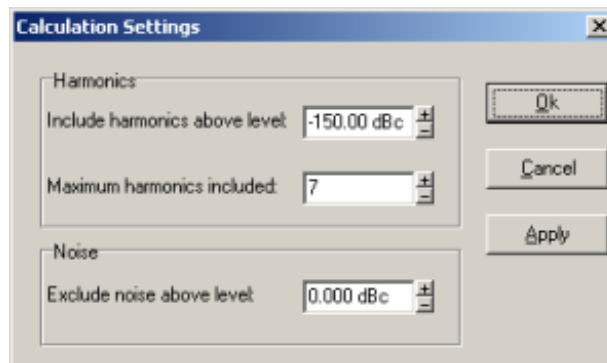
The dialog window shows an overview with the frequency bins that are multiples of the carrier. The carrier is the frequency bin with the highest amplitude. The number of harmonics is determined in the dialog box "Calculation options", described below. Calculation of the harmonic bin:

$$Bin_n = (Carrierbin - 1) * n - 1$$

### Calculation Settings

The parameters SINAD, ENOB, THD, SNR, Peak Distortion and Spurfree can be determined after defining the function of each bin (noise, distortion or carrier). The carrier is determined by the software (frequency bin with the highest amplitude). Multiples of the carrier are the harmonics. All frequency bins that are not harmonic or carrier are assumed to be noise elements.

With the option "Calculation settings" the user can define some selection criteria, which can determine for each frequency bin to be distortion or not and noise or not.



Calculation Settings

Harmonics

Include harmonics above level: -150.00 dBc

Maximum harmonics included: 7

Noise

Exclude noise above level: 0.000 dBc

Ok Cancel Apply

Harmonics:

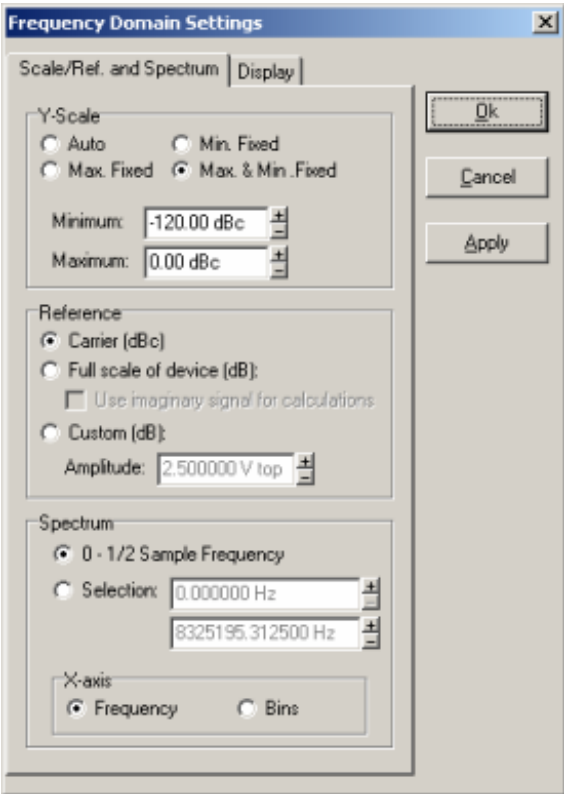
The "Maximum included harmonics" parameter determines the maximum number of frequency bins to be distortion. With the field "Include harmonics above level" it is possible to exclude bins below a certain level.

Noise:

All frequency bins that are not distortion or carrier and are below the level-parameter are assumed to be noise.

**Frequency Domain Settings**

With the "Frequency domain settings" option you can configure how to display the data in the frequency domain.



Y-scale:

- Auto: Maximum and minimum y-scale values are determined by the software. The minimal scale value is limited to minimum calculated level, maximum scale is limited to the maximum calculated level;
- Min. fixed: The minimum y-scale value is fixed to the "Minimum" parameter;
- Max. fixed: The maximum scale y-value is fixed to the "Maximum" parameter;
- Max. and min. fixed: The maximum scale y-value is fixed to the "Maximum" and the minimum y-scale value is fixed to the "Minimum".

Reference:

- Carrier (dBc): All frequency bins are related to the carrier (frequency bin with the highest amplitude). So the carrier itself is 0 dBc;
- Full scale of device (dB): All frequency bins are related to the maximum amplitude of the range (determined by the instrument range setting).
- Custom (dB): All frequency bins are related to the "Amplitude (top)" parameter.

Spectrum:

- Display all bins of the spectrum. All bins are taken into account for the parameter calculations.
- Display a selection of the spectrum. Only the displayed bins are taken into account for the parameter calculations.

Display:

- Amplitude plot: Displays the amplitude of the frequency spectrum with a logarithmic y-scale (dB/dBc) or linear y-scale (top-values or rms-values);
- Phase plot: Displays the phase of the frequency spectrum in radians or degrees;
- Imaginary parts only: Shows only the imaginary results of the Fourier transform;
- Real parts only: Shows only the real results of the Fourier transform.

There are three possible draw styles:

- Line: A line from data point to data point is drawn;
- Filled: The area below all data points is filled up;
- Small bars: A line (small bar) from the minimal y-scale to the data point is drawn.

## Parameter Calculation

The definitions of dynamic result parameters are:

$$SINAD = \frac{c}{n+d} \quad ENOB = \frac{SINAD - 1.8}{6.02} \quad SNR = \frac{c+d}{n}$$

$$THD = \frac{d}{c}$$

Where:

$$c = \sqrt{\text{Re}^2_{\text{carrier}} + \text{Im}^2_{\text{carrier}}} \quad d = \sqrt{\sum_{\text{distortion}} \text{Re}^2_{\text{dist.}} + \text{Im}^2_{\text{dist.}}}$$

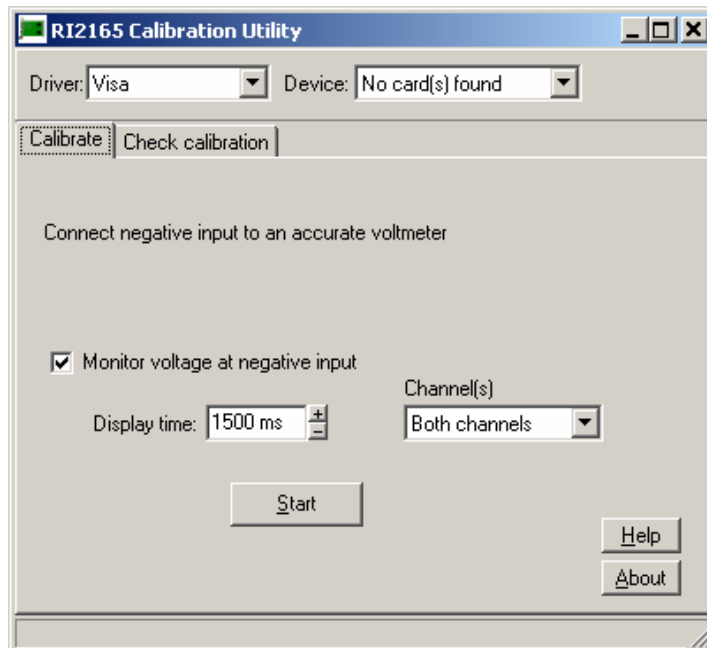
$$n = \sqrt{\sum_{\text{noise}} \text{Re}^2_{\text{noise}} + \text{Im}^2_{\text{noise}}}$$

The Peak distortion is the largest harmonic. The Spurfree is the absolute value of the largest noise element.

## 2165 DIGITIZER CALIBRATION

### Introduction

The 2165 Calibration Software can be used to calibrate and verify the 2165 Digitizer.



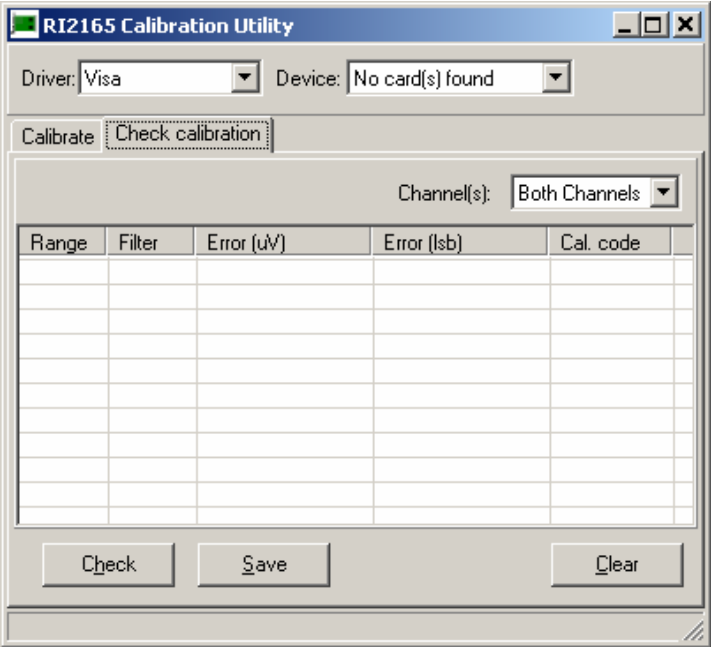
### Calibration Procedure

1. Perform Calibration on 6088-14-DIG or 6088-14-DIG-DIG when it's plugged into the VXI chassis.
2. Select the driver and card to calibrate;
3. Select Tab "Calibrate";
4. Connect a voltmeter (6.5 digits or better) to the negative input of the channel which should be calibrated;
5. Press the "Start" button and follow the instructions;
6. Fill in the measured voltages;

- 7. After the DC offset voltage is calibrated the program starts the auto-calibration procedure. The calibration voltages can be monitored during the calibration procedure;
- 8. If the auto-calibration procedure is finished, press the "Store" button for storing the new calibration values in the on board EEPROM.

### Checking the Calibration Values

One can check the calibration errors. Select the Tab "Check calibration" and press the button "Check" to start the check.





## DIGITAL TO ANALOG CONVERTER (DAC)

---

### Introduction

The 6088-14 can accommodate one or two 3165 dual-channel Digital to Analog Converter (DAC) card(s) installed in a dual VXI carrier. The DAC is based on dual 14-bit high speed digital to analog converters. **Figure 7-1** shows the functional block diagram for the 3165. As shown in the diagram channel-B is an exact copy of channel-A except for the addition of the signal path from channel-A.

On the left of the diagram is the VXI Bus interface with all the required signals. The onboard stimulus generator contains an address counter and a 14bit x 512k RAM memory.

When the 3165 is triggered, the stimulus memory starts from its start address.

During the measurement an internal or external clock increments the memory counter. When the counter reaches the value of the stop address it jumps back to the start address. This allows the generation of continues analog patterns.

If desired the generated analog signal can be filtered to remove quantization noise and improve the dynamic performance.

The attenuator has 7 steps of 3dB each. Since the Main Digital to Analog Converter (DAC) has a programmable reference voltage that allows a proportional output level variation of 3 dB, the Main DAC output level is proportionally adjustable over a range of 24dB.

A DC-offset voltage can be added to the signal before it is output via the differential 50 Ohm output stage.

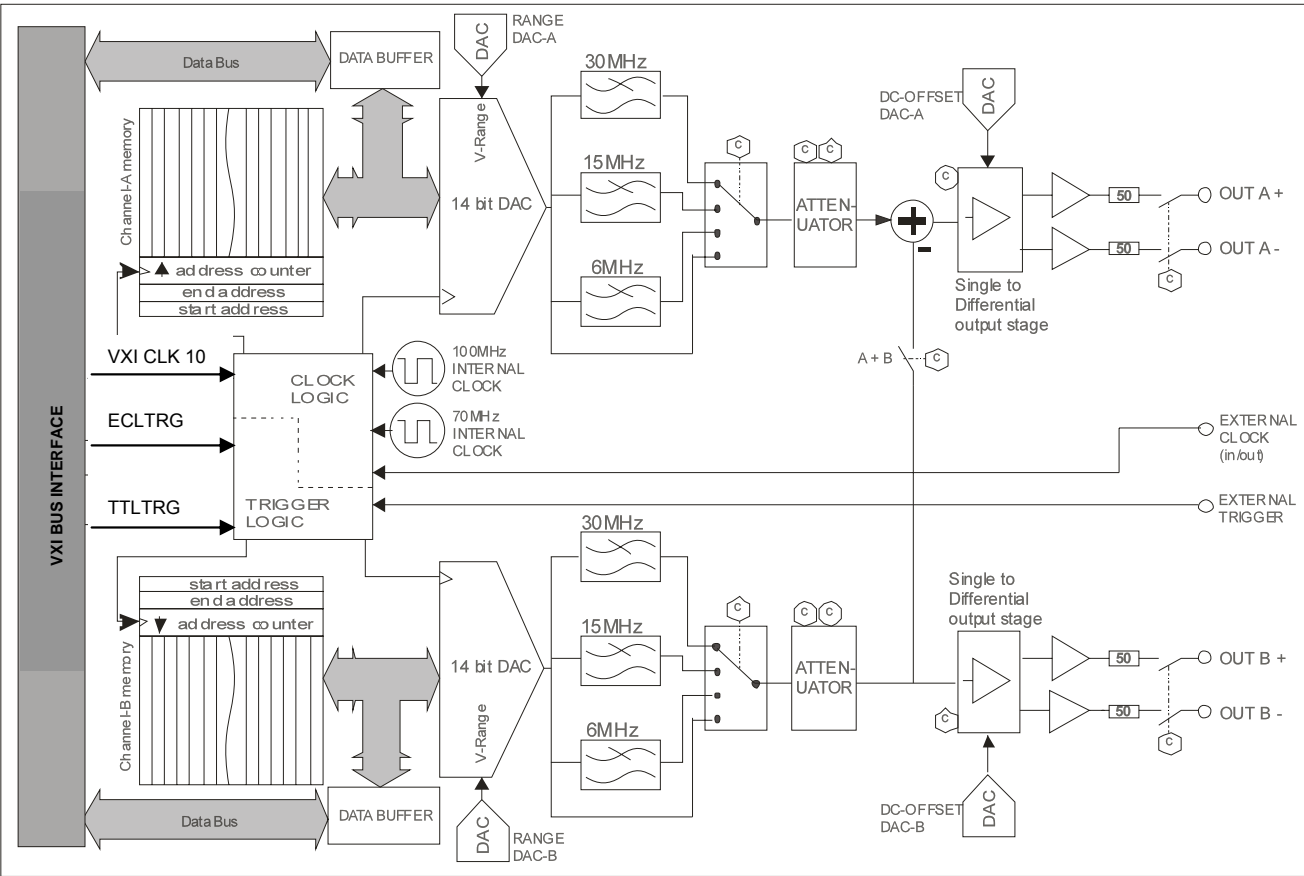


Figure 7-1, 3165 Functional Block Diagram

## Device Operation

### Memory Operation

The stimulus memory of the 3165 is addressed through a counter. This counter is active during pattern generation as during reading or writing the memory via the bus. In Pattern generation mode the counter starts counting from the start (See **Figure 7-2**) and increments on each sample clock until it reaches the end address. Then it jumps back to the start address and repeats the same memory segment. The start address and end address values should be written to the appropriate registers before the pattern starts. When writing the start address, this value is also loaded into the counter.

In bus access mode the same mechanism is active except that the clock is now a read or write signal. To write a memory section the start address and stop address should be set first. The counter is now pointing to the start address and the content of this address can be read or written via register 04 Hex (84 Hex). After each read or write the counter increments to the next address allowing burst read or write actions. Note that the counter will jump back to the start address when it reaches the stop address.

Since the memory may not be read or written during pattern generation, there is a Lock bit that should be set to allow pattern generation. The memory is then locked for reading or writing. This is possible again after setting the lock bit to unlock. A pattern that was still running at that moment will be aborted.

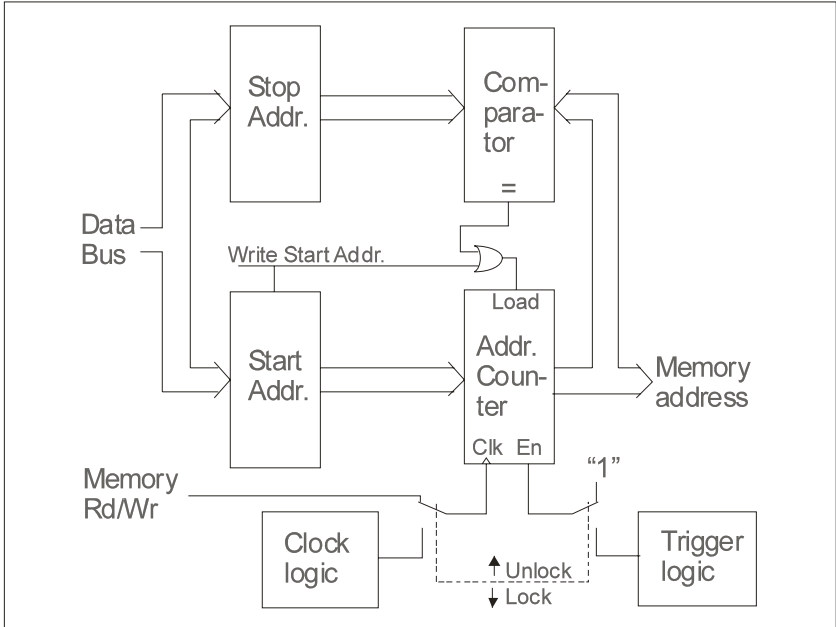


Figure 7-2, Pattern Generation Mode Block Diagram

Register assignment:

ADDRESS Channel A (B)	Operation (R/W )	Data	Description
01 (81) Hex	W	00000 : 3FFFF Hex	Write stop address
02 (82) Hex	W	00000 : 3FFFF Hex	Write start address
04 (84) Hex	R and W	0000 : FFFF Hex (auto increment)	Read or write to memory
05 (85) Hex	R	00000 : 3FFFF Hex	Read counter
0F (8F) Hex	W	00 Hex Unlock 01 Hex Lock, ready for trigger	Lock / Unlock (Measurement mode /Memory access mode)

## Trigger Processing

The 3165 starts generating a signal after it is triggered. The separate trigger circuits for Channel-A and Channel-B allows each channel to start on different sources and on different edges. The 3165 accepts triggers from the VXI back plane trigger sources including VXI\_ECLTRG 0..5 and VXI TTLTRG 0..7. In addition, the 3165 allows selection of the front panel trigger input and a software initiated trigger. The front panel trigger input uses normal TTL logic levels, with a 0.5V nominal threshold for a low level and a 2V nominal threshold for a high level. The 3165 can handle positive and negative, edge and level trigger signals, see the (SELECT TRIGGER EDGE) command. In Level trigger mode the pattern generator starts when Trigger goes active and stops when trigger goes inactive. In edge trigger mode the pattern generator starts at a trigger edge and either stops on the next trigger edge or runs continuously until stopped by the software. **Figure 7-3** shows a diagram of the 3165 trigger processing.

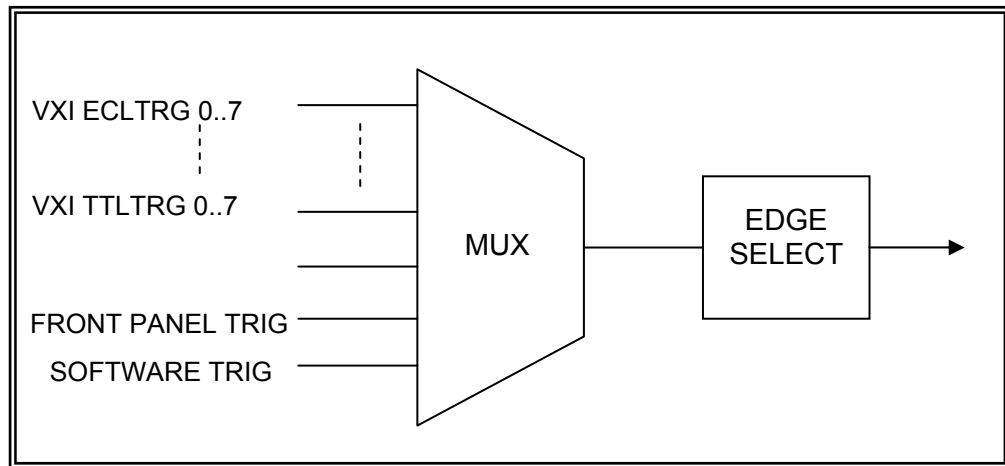


Figure 7-3, Trigger Processing

**Register assignment:**

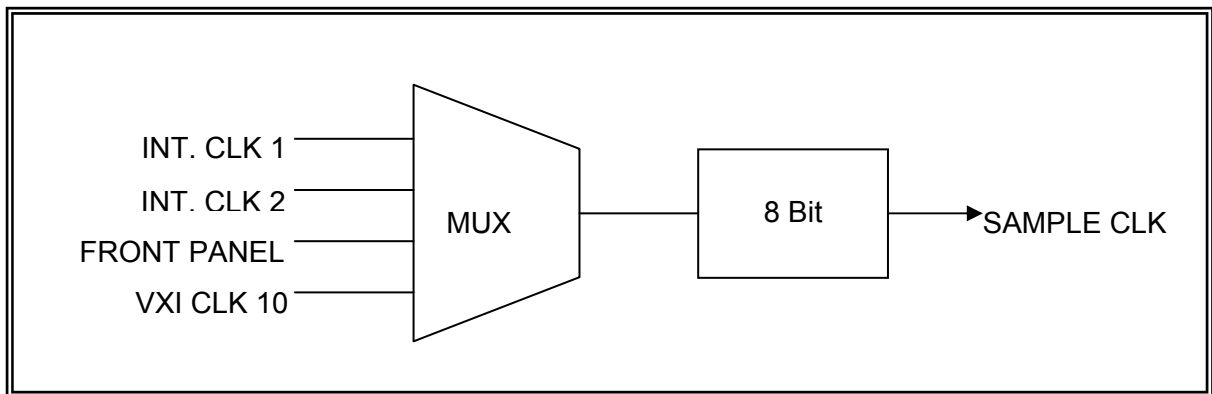
ADDRESS Channel A (B)	Operation (R/W )	Data	Description
0B (8B) HEX	W	x0 Hex FRONT PANEL TRIG x8 Hex SOFTWARE TRIG	Select Trigger source
0B (8B) HEX	W	0x Hex positive level 1x Hex negative level 2x Hex positive edge, toggle on/off 3x Hex negative edge, toggle on/off 6x Hex positive edge, continuous 7x Hex negative edge, continuous	Select Edge

ADDRESS Channel A (B)	Operation (R/W )	Data	Description
07 (87) Hex	R and W	00 Hex No trigger 01 Hex Software trigger channel A 02 Hex Software trigger channel B 03 Hex Software trigger channel A & B	Write = software trigger start. Read = hardware and software trigger status.

## Clock Select

After triggering, the memory counter runs on an internal or external clock signal. The external clock can be the front panel clock or the 10 MHz VXI clock. The two internal clock sources are either 100MHz and 70MHz. To run on lower clock frequencies it is possible to divide the clock source to a lower frequency with the onboard divider. The divider is 8 bit so it can divide with a maximum of 256. Both channels can be set to a different clock source and a different divider ratio. If the front panel clock input is not used by one of the two channels, the sample clock from channel-A is present on this connector (See **Figure 7-4**).

The front panel clock input has a 50 Ohm termination to ground.



**Figure 7-4, Clock Select**

### Register assignment:

ADDRESS Channel A (B)	Operation (R/W )	Data	Description
08 (88) HEX	W	00 Hex FRONT PANEL CLK 01 Hex INT CLK 1 (100MHz) 02 Hex INT CLK 2 (70MHz) 03 Hex VXI 10MHz CLK	Select Clock source
06 (86) HEX	W	00 : FF HEX	Write clock divider

### Filter Select

The 3165 has a bank of selectable 3-pole low pass filters for each channel. These filters allows improved reconstruction of the analog output signal. The cutoff frequencies are: 6 MHz, 15 MHz, and 30 MHz. The filters can also be bypassed for reproduction of wide band signals (See **Figure 7-5**).

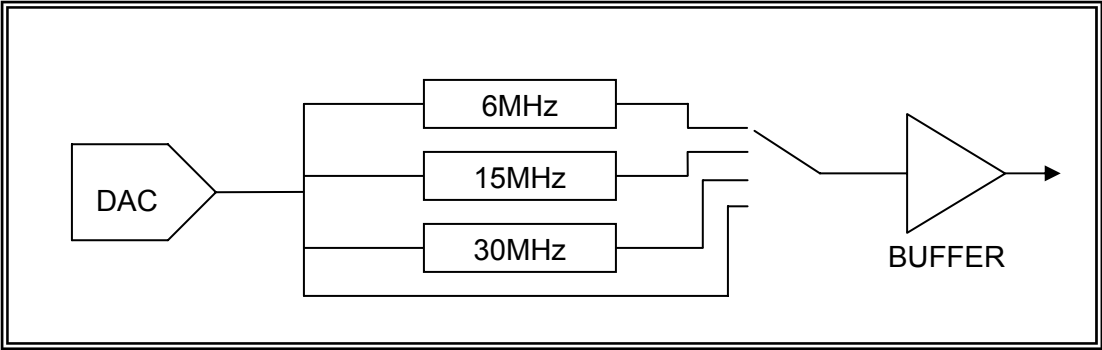


Figure 7-5, Filter Select

Register assignment:

ADDRESS	Operation (R/W )	Data	Description
Channel A (B)			
0A (8A) HEX	R and W	01 Hex No filter 02 Hex 6MHz filter 04 Hex 15MHz filter 08 Hex 30MHz filter	Select Filter DAC A



## Output Stage

Both channels have two different connect modes, single and differential output. If the analog output is disconnected, the internal circuit is disconnected from the output by a mechanical switch. The output includes a series 50Ω 0.1% tolerance resistor as back termination.

It is also possible to add both channels together.

**Pay attention**, the addition is  $A + (B \text{ inverted})$ .

If the signal from DAC B is added to Channel A, the signal from DAC B is still present on channel B (See **Figure 7-6**).

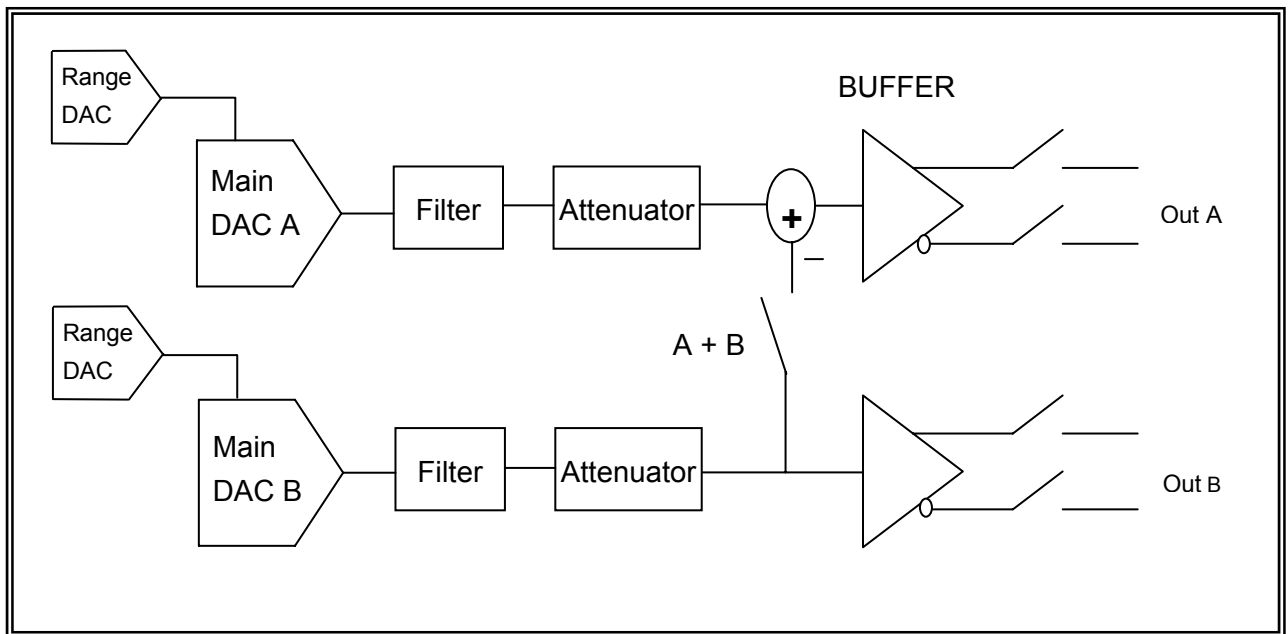


Figure 7-6, Output Stage

### Register assignment:

ADDRESS Channel A (B)	Operation (R/W )	Data (*)	Description
09 (89) HEX	R and W	40 Hex single out (OUT+) 60 Hex differential out C0 Hex Add B to A single out E0 Hex Add B to A diff out	Output single/ differential Output add (only for channel-A)

(\*) bits should be OR-ed with the attenuator code to obtain the Hex code to be written

## DAC Range

The output signal level is a combination of the DAC, who sets the reference from the main DAC, and the attenuator (described on page 7-10). The attenuator has steps of  $-3\text{dB}$ , and the range DAC covers the range values in-between. The DAC is serially programmed through a 3-wire serial bus that accepts 16-bit serial words. Data is loaded Most Significant Bit (MSB) first. Data is shifted into the serial input on the rising edge of the clock. After transferring data the chip select must be taken low so that the DAC reacts on the data. The serial data must be written through the parallel data lines from the module. For connections and a timing diagram see "Offset DAC" description.

### Register assignment:

ADDRESS Channel A (B)	Operation (R/W )	Data	Description
0D Hex	W	0000 : FFFF ( bit0=DO bit1=CLK bit3=OFFCS bit4=RANGECS )	Write range DAC & offset DAC

## Attenuator

The output level can be attenuated from 0 to  $-21\text{dB}$  in steps of  $-3\text{dB}$ .

For ranging steps less then 3dB, see DAC Range description.

### Register assignment:

ADDRESS Channel A (B)	Operation (R/W )	Data (*)	Description
09 (89) HEX	W	01 Hex 0dB 11 Hex $-3\text{dB}$ 02 Hex $-6\text{dB}$ 12 Hex $-9\text{dB}$ 04 Hex $-12\text{dB}$ 14 Hex $-15\text{dB}$ 08 Hex $-18\text{dB}$ 18 Hex $-21\text{dB}$	Select Attenuation (bits should be OR-ed with the output mode bit to obtain the Hex code to be written)

(\*) bits should be OR-ed with the output stage configuration to obtain the Hex code to be written

## DAC Offset

It is possible to add a DC offset to the output signal with the offset DAC.

The offset DAC is serially programmed through a 3-wire serial bus that accepts 16-bit serial words. Data is loaded MSB first. Data is shifted into the serial input on the rising edge of the clock. After transferring data the chip select must be taken low so that the DAC reacts on the data. The serial data must be written through the parallel data lines from the module. For connections see **Table 7-1**.

### Register assignment:

ADDRESS Channel A (B)	Operation (R/W )	Data	Description
0D Hex	W	0000 : FFFF ( bit0=DO bit1=CLK bit3=OFFCS bit4=RANGECS )	Write range DAC & offset DAC

Data line	Serial function
D0	Serial data in
D1	clock
D3	chip select offset DAC
D4	chip select range DAC

**Table 7-1, Serial DAC Control Line Assignment**

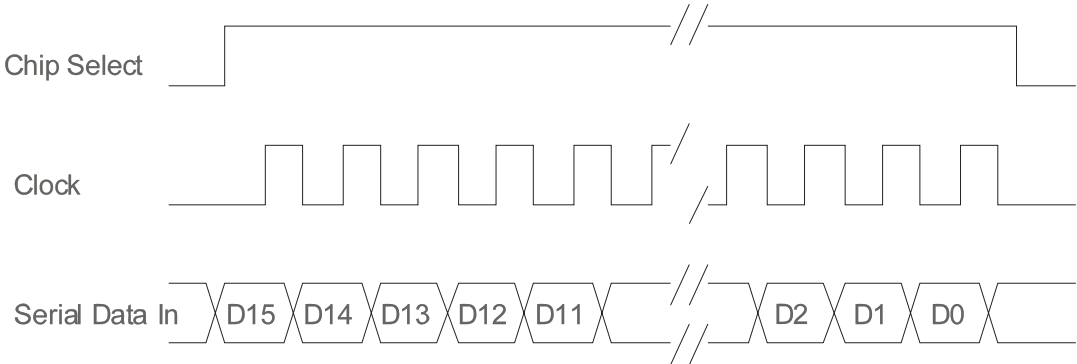


Figure 7-7, Serial DAC Timing Diagram

### Output Offset Calibration DAC

To minimize the total output offset and differential offset, two calibration DACs are added. These 10 bit DACs insert a calibration voltage to the output buffer. (Figure 7-8) The calibration can be done with calibration software. After calibration the settings are stored in the on-board EEPROM.

It is recommended to start a calibration after a system warm up period of half an hour.

The calibration details and the EEPROM organization can be found in Chapter 10, 3165 Calibration.

For connections and register assignment see "Offset correction DAC" description.

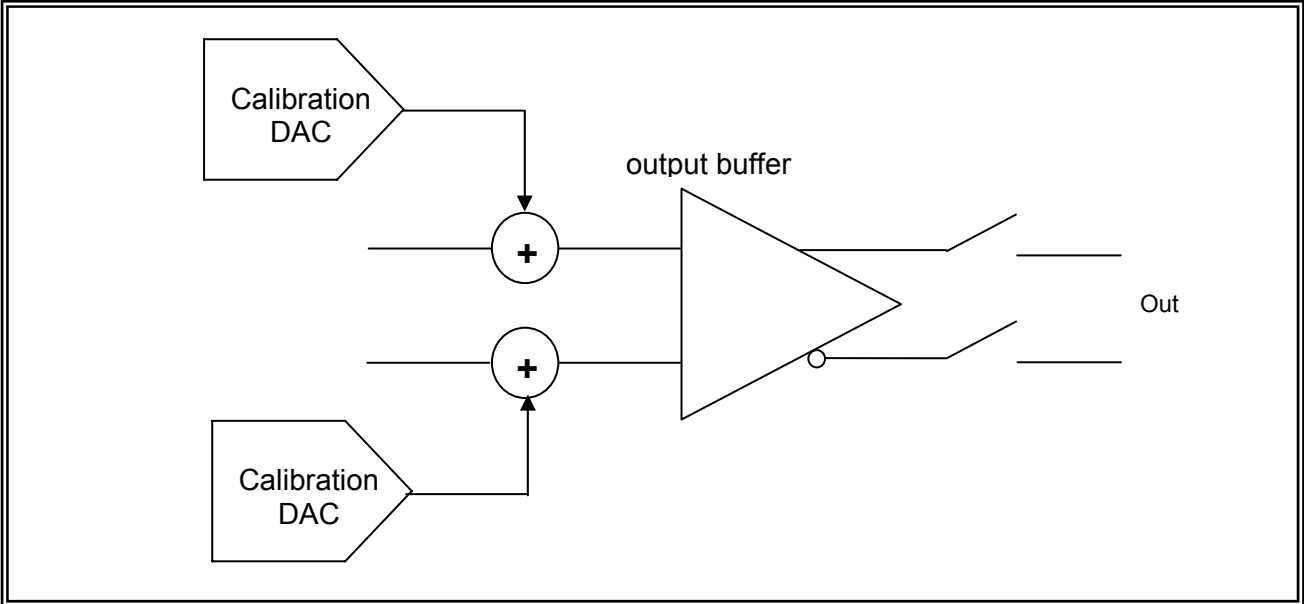
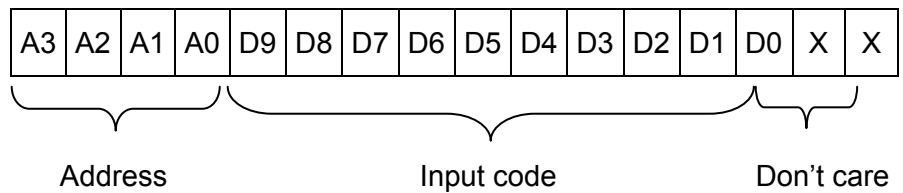


Figure 7-8, Output Offset Calibration DAC

The output calibration DAC's are all on a single chip. To select the offset calibration DAC for each output, an address is added to the input code (See **Figure 7-9**). For connections see "Offset Correction DAC" description.

offset calibration	Address
+ out Channel A	8 hex
- out Channel A	7 hex
+ out Channel B	1 hex
- out channel B	0 hex



**Figure 7-9, Calibration DAC Addressing**

**Offset Correction DAC**

This calibration DAC is an offset correction for the main DAC. If this DAC is calibrated the output must stay stable with all settings from the range DAC.

To calibrate this DAC, the signal DAC should be programmed midscale (code 8000 hex). The offset correction DAC should now be trimmed in such a way that the voltage on the output does not change when the range DAC is changed over the full range.

The calibration software automatically sets all the DACs in the right position. All details can be found in Chapter 10, 3165 Calibration.

**Register assignment:**

ADDRESS Channel A (B)	Operation (R/W )	Data	Description
0E Hex	W	0000 : FFFF (bit0=DO bit1=CLK bit3=cal-offcorCS bit4=cal-offoutCS )	Write Calibration offset Zero scale from DAC and output offset (10bit)

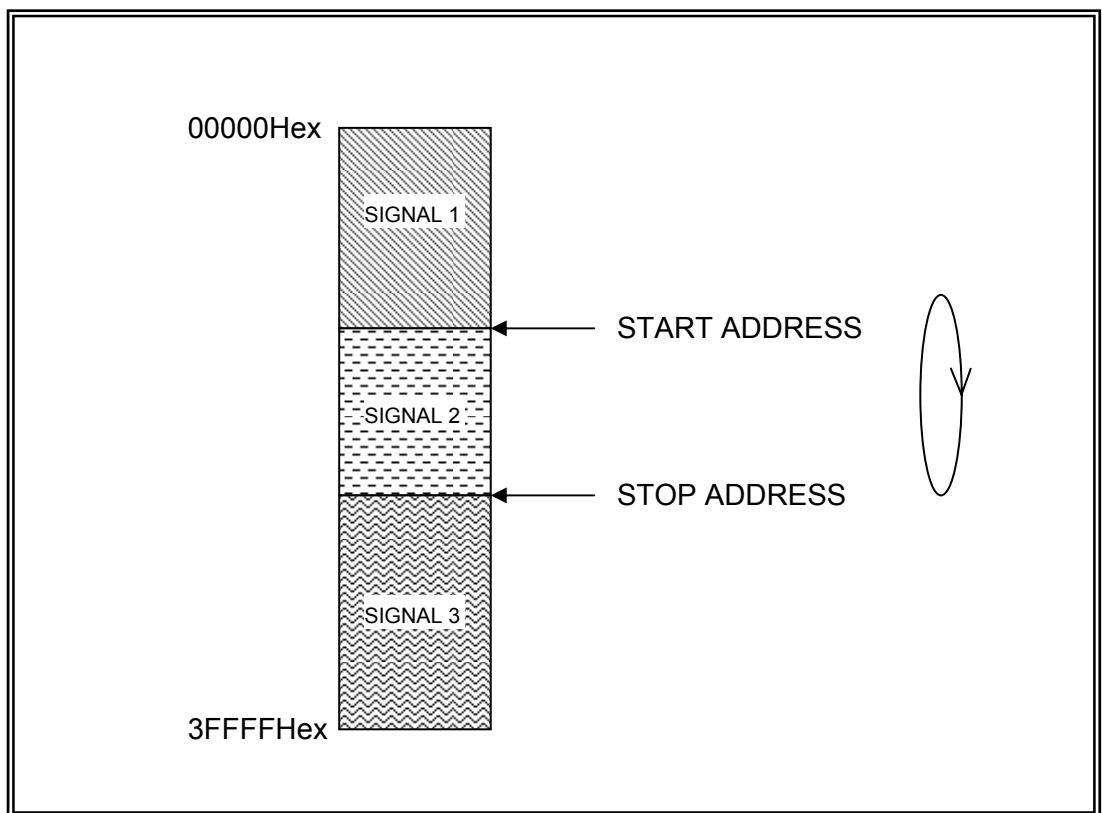
Data line	Serial function
D0	Serial data in
D1	clock
D3	chip select offset correction DAC
D4	chip select output offset calibration DAC

**Table 7-2, Serial Calibration DAC Control Line Assignment**

## Memory Segmentation

On the 3165 there is a 512k word stimulus memory for each channel.

In this memory the wave forms can be stored. It is also possible to split up the memory into smaller segments, and load different waveforms into individual segments. The desired signal can be selected with the start and stop address. When the 3165 is running the waveform stored in that memory segment is converted by the DAC to the analog output signal. It is not possible to change from memory segment while the DAC is running (See **Figure 7-10**).



**Figure 7-10, Memory Segmentation**

### Register assignment:

ADDRESS Channel A (B)	Operation (R/W )	Range	Description
02 (82) HEX	W	00000 : 3FFFF HEX	Write start address
01 (81) HEX	W	00000 : 3FFFF HEX	Write stop address

### Serial EEPROM

The 3165 has a 93C66 serial EEPROM onboard (256 locations of 16 bit each). This EEPROM is used to store calibration values and other module specific information.

The read and write sequence for the EEPROM are according the Microwire® standard. **Table 7-3** shows the bit assignment. There is only one EEPROM for both channels! Access to the EEPROM is via the Channel-A register bank only.

**Register assignment:**

ADDRESS	Operation (R/W )	Range	Description
0C Hex	R and W	00 : FF Hex	Read Write EEPROM

Data line	Serial function
D0	Serial data in/out
D1	clock
D3	chip select

**Table 7-3, Serial EEPROM Control Line Assignment**



## Front Panel

A diagram of the 3165 front panel is shown in **Figure 7-11**. On the front panel, there are six SMB coaxial connectors. Four connectors are used for the differential output from channel A and B. The two other connectors are for the trigger input and the clock. The function of these connectors is described below.

### OUT A+ and OUT A-

This is the output from channel A. In single ended mode the connector A+ is used. In differential mode the output signal is present between A+ and A-. The output impedance is always 50Ω.

### OUT B+ and OUT B-

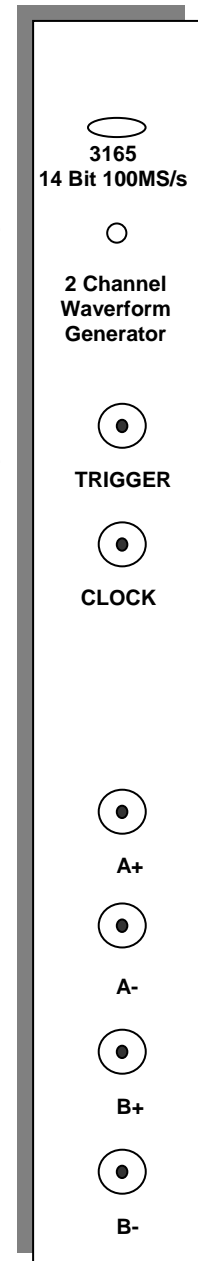
This is the output from channel B. In single ended mode the connector B+ is used. In differential mode the output signal is present between B+ and B-. The output impedance is always 50Ω.

### CLK in/out

This connector is bi-directional. If the clock source selector is set to FRONT this connector is an input for the sample clock, in all other clock source modes this connector is an output for the sample clock. This signal can be used to clock or synchronize other modules. The input impedance of this input is 50Ω.

### TRIG in

If the trigger selector is set to FRONT this input accepts signals that starts generation of output waveforms. The trigger input can be set to level or negative level trigger and edge or negative edge trigger. Trigger level for this input is TTL.



**Figure 7-11, 3165 Front Panel**

## VXI Interface

Supplied with the 3165 is a PC software package that consists of a windows driver, Labview driver and a demo program. Details on how to use this software are given in Chapter 8, 3165 Software User Manual.

# Register Assignment:

## Channel-A

ADDRESS	Operation (R/W )	DATA	Description
00 Hex	W	XXXX	Set counter to start address
01 Hex	W	00000 : 7FFFF Hex	Write stop address
02 Hex	W	00000 : 7FFFF Hex	Write start address
03 Hex	W	0000 : 3FFF Hex	Write direct to DAC A
04 Hex	R and W	0000 : FFFF Hex (auto increment)	Read or write to memory
05 Hex	R	00000 : 7FFFF Hex	Read counter
06 Hex	W	00 : FF	Write clock divider
07 Hex	R and W	00 Hex No trigger 01 Hex Software trigger channel A 02 Hex Software trigger channel B 03 Hex Software trigger channel A & B	Write = software trigger start. Read = hardware and software trigger status.
08 Hex	W	00 Hex FRONT PANEL CLK 01 Hex INT CLK 1 (100MHz) 02 Hex INT CLK 2 (70MHz) 03 Hex VXI 10MHz CLK 10 Hex Front clock out enable	Select Clock source
09 Hex	R and W	01 Hex 0dB 11 Hex - 3dB 02 Hex - 6dB 12 Hex - 9dB 04 Hex -12dB 14 Hex -15dB 08 Hex -18dB 18 Hex -21dB	Select Attenuator (Combine with Analog output select code)
09 Hex	R and W	40 Hex Single output (OUT+) 60 Hex Differential output C0 Hex add B to A single out E0 Hex add B to A differential out	Analog output select (Combine with Attenuator select code)
0A Hex	R and W	01 Hex No filter 02 Hex 6MHz filter 04 Hex 15MHz filter 08 Hex 30MHz filter	Select Filter
0B Hex	W	x0 Hex FRONT PANEL TRIG x8 Hex SOFTWARE TRIG	Select Trigger source (Combine with select Edge code)
0B Hex	W	0x Hex positive level 1x Hex negative level 2x Hex positive edge, toggle on/off 3x Hex negative edge, toggle on/off 6x Hex positive edge, continuous 7x Hex negative edge, continuous	Select Edge
0C Hex	R and W	00 : FF Hex bit0=DO/Di bit1=CLK bit3=CS	Read / Write EEPROM
0D Hex	W	0000 : FFFF ( bit0=DO bit1=CLK bit3=OFFCS bit4=RANGECS )	Write range DAC & output offset DAC (both 16 bit)
0E Hex	W	bit0=DO bit1=CLK bit3=cal-offcorCS bit4=cal-offoutCS 7 Hex = - output 8 Hex = + output	Write offset correction DAC(12bit) and differential output offset calibration DAC(10bit)
0F Hex	W	00 Hex Unlock 01 Hex Lock, ready for trigger	Lock / Unlock (Measurement mode / Memory access mode)

## Channel-B

ADDRESS	Operation (R/W )	DATA	Description
80 Hex	W	XXXX	Set counter to start address
81 Hex	W	00000 : 3FFFF Hex	Write stop address
82 Hex	W	00000 : 3FFFF Hex	Write start address
83 Hex	W	0000 : 3FFF Hex	Write direct to DAC A
84 Hex	R and W	0000 : FFFF Hex (auto increment)	Read or write to memory
85 Hex	R and W	00000 : 3FFFF Hex	Read counter & Set counter to start address
86 Hex	W	00 : FF	Write clock divider
87 Hex	R and W	00 Hex No trigger 01 Hex Software trigger channel A 02 Hex Software trigger channel B 03 Hex Software trigger channel A & B	Write = software trigger start. Read = hardware and software trigger status.
88 Hex	W	00 Hex FRONT PANEL CLK 01 Hex INT CLK 1 (100MHz) 02 Hex INT CLK 2 (70MHz) 03 Hex VXI 10MHz CLK 10 Hex Front clock out enable	Select Clock source
89 Hex	R and W	01 Hex 0dB 11 Hex - 3dB 02 Hex - 6dB 12 Hex - 9dB 04 Hex -12dB 14 Hex -15dB 08 Hex -18dB 18 Hex -21dB	Select Attenuator (Combine with Analog output select code)
89 Hex	R and W	40 Hex Single output (OUT+) 60 Hex Differential output C0 Hex add B to A single out E0 Hex add B to A differential out	Analog output select (Combine with Attenuator select code)
8A Hex	R and W	01 Hex No filter 02 Hex 6MHz filter 04 Hex 15MHz filter 08 Hex 30MHz filter	Select Filter
8B Hex	W	x0 Hex FRONT PANEL TRIG x8 Hex SOFTWARE TRIG	Select Trigger source (Combine with select Edge code)
8B Hex	W	0x Hex positive level 1x Hex negative level 2x Hex positive edge, toggle on/off 3x Hex negative edge, toggle on/off 6x Hex positive edge, continuous 7x Hex negative edge, continuous	Select Edge
8C Hex	R and W	00 : FF Hex bit0=DO/Di bit1=CLK bit3=CS	Read / Write EEPROM
8D Hex	W	0000 : FFFF ( bit0=DO bit1=CLK bit3=OFFCS bit4=RANGECS )	Write range DAC & output offset DAC (both 16 bit)
8E Hex	W	bit0=DO bit1=CLK bit3=cal-offcorCS bit4=cal-offoutCS 1Hex = - output 2 Hex = + output	Write offset correction DAC(12bit) and differential output offset calibration DAC(10bit)
8F Hex	W	00 Hex Unlock 01 Hex Lock, ready for trigger	Lock / Unlock (Measurement mode / Memory access mode)

This page was left intentionally blank.

## Chapter 8

# 3165 SOFTWARE

---

### Introduction

The 3165 is a 100MHz dual-channel Digital to Analog Converter (DAC). The card is provided with a software driver, utility software, a demo program for LabView and a calibration tool.

The main part of the 3165 driver consists of a Windows® dynamic link library, the ri3165\_32.DLL

For Labview users the driver includes also a Labview® library with the ri3165\_32.dll driver functions.

This chapter describes the functions of the ri3165\_32.dll.

### 3165 Driver Package

The driver includes the following items:

- 1) A low level driver for direct communication;
- 2) The user mode driver, ri3165\_32.dll;
- 3) A Labview® library, ri3165.llb
- 4) A LabWindows® driver (function tree) ri3165.fp

The following low level drivers can be installed:

- 1) VISA from National Instruments.

### Installation

Install the 3165 Driver Software before the hardware is placed in the system.

Place the installation CD in the CD-ROM. If the installation program does not start automatically, run the program setup.exe (placed in the root of the CD-ROM).

If the software is installed on a Windows® NT based operating system (Win2000, WinNT, WinXP), you should have Administrator rights.

After installation shutdown the computer and place the 3165 in the system. After turning on the computer the operation system should automatically detect the new hardware.

## 3165 DLL Functions

This chapter describes the functions of the dll. After the description follows a table with the necessary parameters belonging to the function. The following parameter types are used:

Type	Details
unsigned long	4-byte (32 bit) unsigned long
double	8-byte floating point
unsigned long*	reference variable (pointer) to a 4-byte (32 bit) unsigned long
double*	reference variable (pointer) to a 8-byte floating point

Every function returns the 3165\_status (type: 32-bit integer). A negative value corresponds to an error. After a successful completion the return status is ri3165\_SUCCESS, which corresponds to a 0. The possible status codes can be found at the end of this Chapter.

### ri3165\_Close (instrumentHandle)

#### Description:

Close the card session. All resources belonging to the card will be released.

#### Parameters:

Name	Type	Direction	Description	Value
instrumentHandle	unsigned long	in	card identifier	0 to 2 <sup>32</sup>

### ri3165\_ConnectCard (instrumentHandle, cc )

#### Description:

Connect the card output relays. This function connects the output of the active channel in a desired status.

#### Parameters:

Name	Type	Direction	Description	Value
instrumentHandle	unsigned long	in	card identifier	0 to 2 <sup>32</sup>
cc	unsigned long	in	card connect status	0 disconnect 1 single ended 2 differential 3 A + B inverted single ended 4 A + B inverted differential

### ri3165\_GetActive Channel(instrumentHandle, channel )

#### Description:

This function returns the active channel. The channel settings can only be modified if it is the active channel. The channel can be set active with the function ri3165\_SetActiveChannel().

#### Parameters:

Name	Type	Direction	Description	Value
instrumentHandle	unsigned long	in	card identifier	0 to 2 <sup>32</sup>
channel	unsigned long*	out	active channel	1 Channel-A 2 Channel-B

### ri3165\_GetAddress Counter(instrumentHandle, address counter )

#### Description:

This function reads the current position of the memory address counter from the active channel.

#### Parameters:

Name	Type	Direction	Description	Value
instrumentHandle	unsigned long	in	card identifier	0 to 2 <sup>32</sup>
addresscounter	unsigned long*	out	position of the address-counter	0 to 2 <sup>18</sup>

### ri3165\_GetAttenuator (instrumentHandle, attenuation )

#### Description:

This function returns the position of the attenuator from the active channel. The attenuator can be programmed with the function ri3165\_SetAttenuator().

#### Parameters:

Name	Type	Direction	Description	Value
instrumentHandle	unsigned long	in	card identifier	0 to 2 <sup>32</sup>
attenuation	double*	out	current attenuation value	0 to -24 (dB)

### ri3165\_GetCard Address(instrument Handle, address )

#### Description:

When the card is installed under VISA, this function will return 0.

#### Parameters:

Name	Type	Direction	Description	Value
instrumentHandle	unsigned long	in	card identifier	0 to 2 <sup>32</sup>
address	unsigned long *	out	address of card	0 to 2 <sup>32</sup>

### ri3165\_GetCard Connection(instrument Handle, cc )

#### Description:

Get the status of the output relays. This routine returns the connection status from the active channel.

#### Parameters:

Name	Type	Direction	Description	Value
instrumentHandle	unsigned long	in	card identifier	0 to 2 <sup>32</sup>
cc	unsigned long *	out	card connect status	0 disconnect 1 single ended 2 differential 3 A + B inverted single ended 4 A + B inverted differential



**ri3165\_GetCard  
List(count, buslist,  
devicelist)****Description:**

Use this function to retrieve the available arbitrary waveform generators (installed with the VISA driver).

**Parameters:**

Name	Type	Direction	Description	Value
count	unsigned long*	in	available card	0 to 255
buslist	long*	out	list with the bus numbers	0 to 255
devicelist	long*	out	list with the device numbers	0 to 255

**ri3165\_GetCard  
Number(instrument  
Handle, card )****Description:**

The card number can be used as a reference in your program.

**Parameters:**

Name	Type	Direction	Description	Value
instrumentHandle	unsigned long	in	card identifier	0 to 2 <sup>32</sup>
card	unsigned long *	out	card number	0 to 256

**ri3165\_GetDC  
OffsetVoltage  
(instrumentHandle,  
voltage )****Description:**

This function will return the current voltage of the DC-offset DAC (of the active channel). This voltage is previously set with the function ri3165\_SetDCOffsetVoltage().

**Parameters:**

Name	Type	Direction	Description	Value
instrumentHandle	unsigned long	in	card identifier	0 to 2 <sup>32</sup>
voltage	double *	out	voltage of offset DAC	-2.5 to +2.5 volt

**ri3165\_GetError  
Message(code, message)**

**Description:**

This function translates an error code to an error message. The message buffer should be at least 256 bytes long.

**Parameters:**

Name	Type	Direction	Description	Value
code	unsigned long	in	error code	see chapter
message	char*	out	error message	see chapter

**ri3165\_Get Number  
OfCards( cards )**

**Description:**

This function returns the number of available cards in the system. This function will only return a of 0 if there are cards installed with the VISA driver.

**Parameters:**

Name	Type	Direction	Description	Value
cards	unsigned long *	in	number of available cards	0 to 256

**ri3165\_Get  
OffsetCalDacCode  
(instrumentHandle,  
code)**

**Description:**

This function returns the (general) offset calibration dac code.

**Parameters:**

Name	Type	Direction	Description	Value
instrumentHandle	unsigned long	in	card identifier	0 to 2 <sup>32</sup>
code	unsigned long*	out	code	0 to 2 <sup>12</sup>

**ri3165\_Get  
OutputOffsetCalDac  
Code(instrument  
Handle, dac, code)****Description:**

This function returns the code of an output offset calibration dac.

**Parameters:**

Name	Type	Direction	Description	Value
instrumentHandle	unsigned long	in	card identifier	0 to 2 <sup>32</sup>
dac	unsigned long	in	calibration dac	1 to 8
code	unsigned long*	out	code	0 to 2 <sup>10</sup>

**ri3165\_Get  
Revision(revision)****Description:**

This function returns the driver revision.

**Parameters:**

Name	Type	Direction	Description	Value
revision	unsigned long*	out	driver revision	1 to 2 <sup>32</sup>

**ri3165\_Get  
TriggerStatus  
(instrumentHandle,  
triggerstatus )****Description:**

This function returns the trigger status (for the active channel).

**Parameters:**

Name	Type	Direction	Description	Value
instrumentHandle	unsigned long	in	card identifier	0 to 2 <sup>32</sup>
triggerstatus	unsigned long *	out	current trigger status	0 = no channels triggered 1 = channel A triggered 2 = channel B triggered 3 = both channels triggered

**ri3165\_GetStart  
Address(instrument  
Handle, start address )**

**Description:**

This function returns the start address (for the active channel). The start address determines the start position of waveform.

**Parameters:**

Name	Type	Direction	Description	Value
instrumentHandle	unsigned long	In	card identifier	0 to 2 <sup>32</sup>
startaddress	unsigned long*	Out	start address	0 to 2 <sup>18</sup>

**ri3165\_GetStop  
Address(instrument  
Handle, stop address)**

**Description:**

This function returns the stop address of the waveform (for the active channel).

**Parameters:**

Name	Type	Direction	Description	Value
instrumentHandle	unsigned long	in	card identifier	0 to 2 <sup>32</sup>
stopaddress	unsigned long*	out	stop address	0 to 2 <sup>19</sup>

**ri3165\_Init( bus, device,  
instrument Handle)**

**Description:**

Initiate a card with a VISA session. This function starts a card session. Call this routine if the card is installed with VISA. The bus and device number can be determined with the Measurement and Automation eXplorer (MAX) from National Instruments. This function returns a card identifier (reference number), which is needed in most of the other functions to control the card. The function will return the same card identifier if the function is called more than once, without calling ri3165\_Close() in between.

**Parameters:**

Name	Type	Direction	Description	Value
instrumentHandle	unsigned long*	in	card identifier	0 to 2 <sup>32</sup>

**ri3165\_Load  
ArbitraryWaveForm  
(instrumentHandle,  
startaddress, length,  
waveform)****Description:**

This function will load the generator with the Waveform Data. Waveform data should have at least length elements. Data is loaded starting from the parameter startaddress.

**Parameters:**

Name	Type	Direction	Description	Value
instrumentHandle	unsigned long	in	card identifier	0 to 2 <sup>32</sup>
startaddress	unsigned long	in	start address	0 to 2 <sup>19</sup>
length	unsigned long	in	length of waveform	0 to 2 <sup>19</sup>
waveform	double*	in	reference to waveform data	0 to 2 <sup>32</sup>

**ri3165\_Read  
(instrumentHandle,  
offset, data)****Description:**

This function reads from the card offset (register of the card) address. For the available card registers, read the hardware manual.

**Parameters:**

Name	Type	Direction	Description	Value
InstrumentHandle	unsigned long	in	card identifier	0 to 2 <sup>32</sup>
offset	unsigned long	in	card offset address (card register)	see hardware manual
data	unsigned long*	out	read data	register dependent

**ri3165\_Read  
EEPROM(instrument  
Handle, eaddress,  
data)****Description:**

This function reads a 16-bit word from the serial EEPROM at a serial EEPROM address determined by eaddress. If the EEPROM address is previously written with the function ri3165\_WriteEEPROM(), the lower byte will correspond to the byte written with the function ri3165\_WriteEEPROM(). The upper byte will be the complement of the lower byte. This byte can be used for verifying purposes. One on board EEPROM is used for both channels.

**Parameters:**

Name	Type	Direction	Description	Value
instrumentHandle	unsigned long	in	card identifier	0 to 2 <sup>32</sup>
eeaddress	unsigned long	in	EEPROM address	0 to 255
data	unsigned long*	out	read data	0 to 2 <sup>16</sup>

**ri3165\_ReadId  
(instrumentHandle, id )**
**Description:**

This function reads the card ID. A card ID can be set with the function ri3165\_WriteId(). The card ID is placed in the on board serial EEPROM.

**Parameters:**

Name	Type	Direction	Description	Value
instrumentHandle	unsigned long	in	card identifier	0 to 2 <sup>32</sup>
id	unsigned long*	out	card id	0 to 2 <sup>32</sup>

**ri3165\_ReadRam  
(instrumentHandle,  
data )**
**Description:**

Read from the (stimuli) ram of the active channel. The ram address is determined by the address-counter. The address-counter can be initialized with the function ri3165\_SetStartAddress(). After this function call the address-counter is incremented with one step.

**Parameters:**

Name	Type	Direction	Description	Value
instrumentHandle	unsigned long	in	card identifier	0 to 2 <sup>32</sup>
data	unsigned long*	out	read data	0 to 2 <sup>16</sup>

**ri3165\_Read  
RamBuffer(instrument  
Handle, length, buf32 )**
**Description:**

Read length ram-places from the stimuli ram (of the active channel), starting from the current address-counter value. The ram address-counter can be initialized with the function ri3165\_SetStartAddress(). After this function call the address-counter is incremented with "length" steps.

**Parameters:**

Name	Type	Direction	Description	Value
instrumentHandle	unsigned long	in	card identifier	0 to 2 <sup>32</sup>
length	unsigned long	in	number of data elements to be read	0 to 2 <sup>18</sup>
buf32	unsigned long*	out	reference to a buffer for read data	0 to 2 <sup>32</sup>

**ri3165\_Reset  
ToStartAddress  
(instrumentHandle)**

**Description:**

Reset address counter (of the active channel) to the start address. After calling this function the address-counter is back to the last programmed start address. The start address can be programmed with the function ri3165\_SetStartAddress().

**Parameters:**

Name	Type	Direction	Description	Value
instrumentHandle	unsigned long	in	card identifier	0 to 2 <sup>32</sup>

**ri3165\_Set  
ActiveChannel  
(instrumentHandle,  
channel )**

**Description:**

Select the active channel. This function selects the channel to be updated.

**Parameters:**

Name	Type	Direction	Description	Value
instrumentHandle	unsigned long	in	card identifier	0 to 2 <sup>32</sup>
channel	unsigned long	in	active channel	1 Channel-A 2 Channel-B

### ri3165\_Set Attenuator(instrument Handle, attenuation )

#### Description:

Set attenuator(of the active channel) to a desired value. The attenuator can attenuate the programmed sine wave with a value between 0 and 24 dB. In hardware the attenuator consist of a attenuator with 3 dB steps and a programmable gain DAC for all steps between 0 and 3 dB. For more information see the hardware manual. The amplitude defined with the function ri3165\_SignalAdd() will be attenuated with the attenuator value.

#### Parameters:

Name	Type	Direction	Description	Value
instrumentHandle	unsigned long	in	card identifier	0 to 2 <sup>32</sup>
attenuation	double	in	attenuator value	0 to -24 (dB)

### ri3165\_Set ClockDivider (instrumentHandle, clockdivider )

#### Description:

Set the clock-divider. Programs the divider for the sample clock of the active channel. The clock divider divides the clock selected with ri3165\_SetClockSource() and determines the update sample rate of the signal (and consequently the signal frequency).

#### Parameters:

Name	Type	Direction	Description	Value
instrumentHandle	unsigned long	in	card identifier	0 to 2 <sup>32</sup>
clockdivider	unsigned long	in	clock divider value	1 to 256

### ri3165\_Set ClockSource (instrumentHandle, clocksource )

#### Description:

Select the desired clock source (for the selected channel). The clock source and the clock divider determines the update rate of the output signal. See also ri3165\_SetClockDivider().



**Parameters:**

Name	Type	Direction	Description	Value
instrumentHandle	unsigned long	in	card identifier	0 to 2 <sup>32</sup>
clocksource	unsigned long	in	select clock source	0 = FRONT PANEL CLK 1 = INT CLK 1 (100MHz) 2 = INT CLK 2 (70MHz) 3 = VXI 10MHz CLK

**ri3165\_Set  
DacCode(instrument  
Handle, code )****Description:**

Write a code directly to the main-DAC (of the selected channel). This function will program the main-DAC directly with the desired code.

**Parameters:**

Name	Type	Direction	Description	Value
instrumentHandle	unsigned long	in	card identifier	0 to 2 <sup>32</sup>
code	unsigned long	in	code for main-DAC	0 to 2 <sup>14</sup>

**ri3165\_SetDC  
OffsetDacCode  
(instrumentHandle,  
code )****Description:**

Write a code to the DC offset DAC (of the selected channel). This functions programs the 16-bit offset DAC with the desired code. This function can be used for calibration purposes. In normal operation the function ri3165\_SetDCOffsetVoltage() will program the DC offset voltage to a desired level.

**Parameters:**

Name	Type	Direction	Description	Value
instrumentHandle	unsigned long	in	card identifier	0 to 2 <sup>32</sup>
code	unsigned long	in	code for DC offset DAC	0 to 2 <sup>16</sup>

### ri3165\_Set DCOffsetVoltage (instrumentHandle, voltage )

#### Description:

Program the DC offset voltage DAC (of the selected channel) with the desired voltage. The DC offset voltage can be a voltage between the -2.5V and +2.5V.

#### Parameters:

Name	Type	Direction	Description	Value
instrumentHandle	unsigned long	in	card identifier	0 to 2 <sup>32</sup>
voltage	double	in	voltage for offset DAC	-2.5 to +2.5V

### ri3165\_SetDC OffsetLimitVoltages (instrumentHandle, posvolt, negvolt )

#### Description:

Set limit voltages of dc offset DAC (of the selected channel). These voltages are necessary for a calibrated offset voltage.

- Select Channel to calibrate (ri3165\_SelectChannel(..) )
- Be sure channel is not running (ri3165\_SetLockMode (0) )
- Bypass filter (ri3165\_SetFilter(1) )
- Connect card single ended (ri3165\_ConnectCard(1) )
- Set attenuator at 0dB (ri3165\_SetAttenuator (0.0) )
- Program DC offset DAC at maximum  
(ri3165\_SetDCOffsetDacCode(instrumentHandle, 0xFFFF) )
- Measure voltage at output with accurate voltage meter
- Program DC offset DAC at minimum  
(ri3165\_SetDCOffsetDacCode(instrumentHandle, 0x0) )
- Measure voltage at output with accurate voltage meter
- Call this routine with measured voltages
- Repeat steps with next channel
- Call StoreCalibration for storing data in EEPROM

#### Parameters:

Name	Type	Direction	Description	Value
instrumentHandle	unsigned long	in	card identifier	0 to 2 <sup>32</sup>
posvolt	double	in	measured positive voltage	> 2.5V
negvolt	double	in	measured negative voltage	< -2.5V

**ri3165\_SetFilter  
(instrumentHandle,  
filter )****Description:**

Select a desired filter path for the active channel.

**Parameters:**

Name	Type	Direction	Description	Value
instrumentHandle	unsigned long	in	card identifier	0 to 2 <sup>32</sup>
filter	unsigned long	in	filter select	0 = Disconnect 1 = Bypass 2 = 6 MHz filter 3 = 15 MHz filter 4 = 30 MHz filter

**ri3165\_SetLock  
Mode(instrument  
Handle, lock )****Description:**

Lock or unlock the memory access for active channel. A channel should be locked before the channel can be used to generate a signal. The memory cannot be accessed (by a controller) and the channel waits for a trigger in this mode. If the channel is unlocked the channel does not respond to a trigger signal and the memory can be accessing by a controller.

**Parameters:**

Name	Type	Direction	Description	Value
instrumentHandle	unsigned long	in	card identifier	0 to 2 <sup>32</sup>
lock	unsigned long	in	lock/unlock active channel	1 = lock 0 = unlock

**ri3165\_SetOffset  
CalDacCode  
(instrumentHandle,  
code)****Description:**

This function programs the (general) offset calibration dac.

**Parameters:**

Name	Type	Direction	Description	Value
instrumentHandle	unsigned long	in	card identifier	0 to 2 <sup>32</sup>
code	unsigned long	in	code	0 to 2 <sup>12</sup>

**ri3165\_Set  
OutputOffsetCalDac  
Code(instrument  
Handle, dac,code)**

**Description:**

This function programs the code of an output offset calibration dac.

**Parameters:**

Name	Type	Direction	Description	Value
instrumentHandle	unsigned long	in	card identifier	0 to 2 <sup>32</sup>
dac	unsigned long	in	calibration dac	1 to 8
code	unsigned long	in	code	0 to 2 <sup>10</sup>

**ri3165\_SetRange  
DacCode(instrument  
Handle, code )**

**Description:**

Write a code to the range DAC (of the active channel). This function will program the range DAC directly. Normally the range DAC is programmed with the function ri3165\_SetAttenuator().

**Parameters:**

Name	Type	Direction	Description	Value
instrumentHandle	unsigned long	in	card identifier	0 to 2 <sup>32</sup>
code	unsigned long	in	code for DC range DAC	0 to 2 <sup>16</sup>

**ri3165\_SetRange  
LimitVoltages  
(instrumentHandle,  
maxvolt, minvolt )**

**Description:**

Set limit voltages of range DAC. These voltages are necessary for a calibrated range voltage. Calibrate the DC offset DAC before range DAC!

- Procedure to determine limit voltages:
- Select Channel to calibrate (ri3165\_SelectChannel(..) )
- Be sure channel is not running (ri3165\_SetLockMode(0) )

- Bypass filter (ri3165\_SetFilter(1) )
- Set DC offset at 0 V (ri3165\_SetDCOffsetVoltage (0.0))
- Connect card single ended (ri3165\_ConnectCard(1) )
- Set DAC code at maximum (ri3165\_SetDacCode(0x3FFF) )
- Set attenuator at 0dB (ri3165\_SetAttenuator (0.0) )
- Program range DAC at maximum (ri3165\_SetRangeDacCode(instrumentHandle, 0xFFFF) )
- Measure output voltage with accurate voltage meter
- Program range DAC at minimum (ri3165\_SetRangeDacCode(instrumentHandle, 0x0) )
- Measure output voltage with accurate voltage meter
- Call this routine with measured voltages

**Parameters:**

Name	Type	Direction	Description	Value
instrumentHandle	unsigned long	in	card identifier	0 to 2 <sup>32</sup>
maxvolt	double	in	measured maximum voltage	> 2.5V
minvolt	double	in	measured minimum voltage	< 1,768 V (-3dB)

**ri3165\_Set  
SoftwareTriggerStatus  
(instrumentHandle,  
triggerstatus )**

**Description:**

Trigger (start) or stop the channel(s). This function enables the software to trigger and stop the channel(s). Select Software Trigger with the function ri3165\_SetTriggerMode() to enable the software trigger mode.

**Parameters:**

Name	Type	Direction	Description	Value
instrumentHandle	unsigned long	in	card identifier	0 to 2 <sup>32</sup>
triggerstatus	unsigned long	in	start or run	0 inactive trigger 1 trigger channel A 2 trigger channel B 3 trigger both channels

**ri3165\_Set  
StartAddress  
(instrumentHandle,  
startaddress )**

**Description:**

This function writes the start address. If the start address is written the counter is also loaded with this address. To read from a specific memory address use ri3165\_SetStartAddress to jump to this memory location.

**Parameters:**

Name	Type	Direction	Description	Value
instrumentHandle	unsigned long	in	card identifier	0 to 2 <sup>32</sup>
startaddress	unsigned long	in	desired start address	0 to 2 <sup>19</sup>

**ri3165\_Set  
StopAddress  
(instrumentHandle,  
stopaddress )**

**Description:**

Set stop address. The stop address is the last memory address during signal generation. After the stop address the address counter returns to the start address (set with ri3165\_SetStartAddress()). The number of samples of the signal is determined by the start address and stop address.

**Parameters:**

Name	Type	Direction	Description	Value
instrumentHandle	unsigned long	in	card identifier	0 to 2 <sup>32</sup>
stopaddress	unsigned long	in	desired stop address	0 to 2 <sup>19</sup>

**ri3165\_Set TriggerMode  
(instrumentHandle,  
triggersource,  
triggermode )**

**Description:**

Select trigger source and trigger mode. In lock mode (Set with the function ri3165\_SetLockMode() ) the signal generation can be started (triggered) by the selected trigger source.

**Parameters:**

Name	Type	Direction	Description	Value
instrumentHandle	unsigned long	in	card identifier	0 to 2 <sup>32</sup>
triggersource	unsigned long	in	select trigger source	0 = FRONT PANEL TRIG 8 = SOFTWARE TRIG
triggermode	unsigned long	in	select trigger mode	0 = positive level 1 = negative level 2 = positive edge(retrigger) 3 = negative edge(retrigger) 4 = positive edge(continuous) 5 = negative edge (continuous)

**ri3165\_Signal**

**Add(instrumentHandle, type, amplitude, periods, phase, symmetry)**

**Description:**

Add a signal definition for the active channel.

Call this function to add a new signal definition. It is possible to add many signal definitions. For a sine the symmetry parameter has no effect. For a square the symmetry determines the duty cycle (position of the negative edge) of the square. 0% and 100% will result in a dc-offset voltage. For a triangle the top of the triangle will be moved left (less than 50%) or right (more than 50%). 0% percent will result in a ramp starting at the top, 100% results in a ramp starting at the bottom. The amplitude will be attenuated with the attenuator value defined with ri3165\_SetAttenuator(). Call ri3165\_SignalClear() to clear all defined signal definitions. Call the function ri3165\_SignalToRam() to fill the stimuli memory with the defined signals (once, after all signals are defined).

**Parameters:**

Name	Type	Direction	Description	Value
instrumentHandle	unsigned long	in	card identifier	0 to 2 <sup>32</sup>
type	unsigned long	in	type of signal	0 = sine 1 = square 2 = triangle
amplitude	double	in	amplitude of the signal	0 – 2.5 V
periods	unsigned long	in	periods of the signal	1 – 130,000
phase	double	in	phase of the signal	0 – 360 (degrees)
symmetry	double	in	symmetry of the signal in case of square (duty cyle)	0 – 100 (%)

### ri3165\_Signal Clear(instrument Handle)

**Description:**

Clear all signal definitions for the active channel. Before starting to define a new signal definition call this function to clear all previously defined signal definitions. Call ri3165\_SignalAdd() to add a new signal definition.

**Parameters:**

Name	Type	Direction	Description	Value
instrumentHandle	unsigned long	in	card identifier	0 to 2 <sup>32</sup>

### ri3165\_Signal ToRam(instrument Handle)

**Description:**

Write the signal definition(s) for the active channel defined with the function ri3165\_SignalAdd() to stimuli memory. Call this function once, after all signal definitions are defined with ri3165\_SignalAdd().

**Parameters:**

Name	Type	Direction	Description	Value
instrumentHandle	unsigned long	in	card identifier	0 to 2 <sup>32</sup>



**ri3165\_Store  
CalibrationData  
(instrumentHandle)****Description:**

Store calibration data (of both channels) in serial EEPROM. This function should be called after a calibration procedure to store the calibration data in the on board serial EEPROM.

**Parameters:**

Name	Type	Direction	Description	Value
instrumentHandle	unsigned long	in	card identifier	0 to 2 <sup>32</sup>

**ri3165\_Write  
(instrumentHandle,  
offset , data )****Description:**

Write data to an offset address (register) on the card. For the available card registers, read the hardware Chapter 7.

**Parameters:**

Name	Type	Direction	Description	Value
instrumentHandle	unsigned long	in	card identifier	0 to 2 <sup>32</sup>
offset	unsigned long	in	offset address (register)	see hardware manual
data	unsigned long	in	data to write	see hardware manual

**ri3165\_Write  
EEPROM(instrument  
Handle, eaddress,  
data )****Description:**

Write a data byte to the EEPROM address (defined with eaddress) of the serial EEPROM. An EEPROM address has place for 2 bytes (16 bits). With this function the upper byte will be filled with the complement of the lower byte. This byte can be used for verifying purposes during reading. With this function ALL EEPROM addresses can be written! So the calibration data and module ID can be changed with this function! Till EEPROM address 89 are reserved for calibration data and the module ID.

**Parameters:**

Name	Type	Direction	Description	Value
instrumentHandle	unsigned long	in	card identifier	0 to 2 <sup>32</sup>
eaddress	unsigned long	in	EEPROM address	0 to 255
data	unsigned long	in	byte to be written	0 to 255

**ri3165\_WriteId  
(instrumentHandle,  
id )**

**Description:**

This function writes a card ID in the serial EEPROM. The card ID may be any 32 bit value. The card ID can be read with the function ri3165\_ReadId().

**Parameters:**

Name	Type	Direction	Description	Value
instrumentHandle	unsigned long	in	card identifier	0 to 2 <sup>32</sup>
id	unsigned long	in	card id	0 to 2 <sup>32</sup>

**ri3165\_Write  
Ram(instrument  
Handle, data )**

**Description:**

Write to the (stimuli) ram of the active channel. The ram address is determined by the address-counter. The address-counter can be initialized with the function ri3165\_SetStartAddress(). After this function call the address-counter is incremented with one step.

**Parameters:**

Name	Type	Direction	Description	Value
instrumentHandle	unsigned long	in	card identifier	0 to 2 <sup>32</sup>
data	unsigned long	in	data to write	0 to 2 <sup>14</sup>

**ri3165\_Write  
RamBuffer  
(instrumentHandle,  
length, buffer )**

**Description:**

Write a buffer with length (32 bit) words to the (stimuli) ram of the active channel, starting at the current address counter position. The address-counter can be initialized with the function ri3165\_SetStartAddress(). After this function call the address-counter is incremented with "length" steps.

**Parameters:**

Name	Type	Direction	Description	Value
instrumentHandle	unsigned long	in	card identifier	0 to 2 <sup>32</sup>
length	unsigned long	in	number of data elements to be written	0 to 2 <sup>18</sup>
buffer	unsigned long*	in	reference to buffer with data to be written	0 to 2 <sup>32</sup>

## Status Codes

This chapter will give an overview of the possible status codes that can be returned by the dll-functions.

## General

These codes can be return by the functions in both cases: VISA and non-VISA driver.

Completion without error:

Constant name	Value	Description
ri3165_SUCCESS	0x0	No error(s)

General error codes:

Constant name	Value	Description
ri3165_ERROR_INVALID_CHANNEL	0xBFFE0004	Invalid channel
ri3165_ERROR_INVALID_PARAMETER	0xBFFE0005	Invalid parameter
ri3165_ERROR_MEMORY	0xBFFE0006	Could not allocate memory
ri3165_ERROR_NO_SIGNALDEF	0xBFFE0007	No signal defined
ri3165_ERROR_EEPROMCHECK	0xBFFE0008	EEPROM verify error

VISA Error Codes:

For the VISA completion codes and error codes, please read the NI-VISA programmer reference.

## Programming example

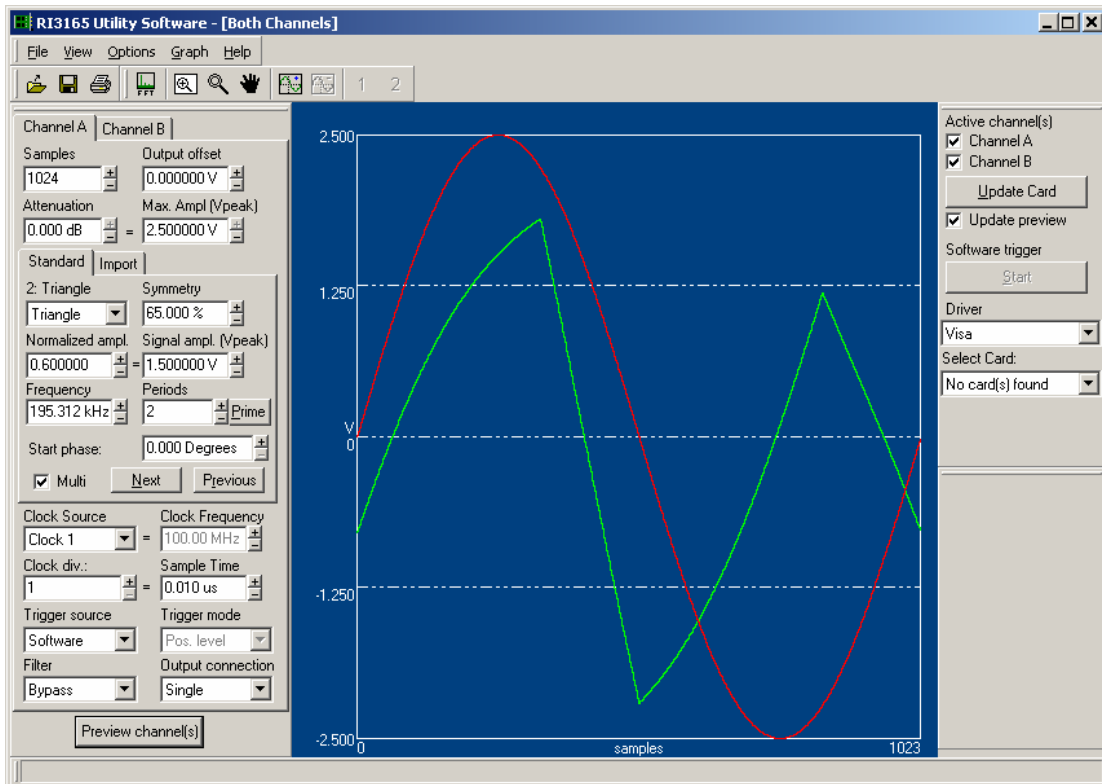
Calling the functions in the following sequence, will generate a sine wave of 1.074 MHz with an amplitude of 1 Volt (2Vpp) on the output of channel 1:

- `ri3165_Open(0, & instrumentHandle)`: Get a card reference number (`instrumentHandle`, card identifier). `&` = reference to the variable;
- `ri3165_SetActiveChannel(instrumentHandle, 1)`: Select channel 1 to be updated;
- `ri3165_SetLockMode(instrumentHandle, 0)`: Unlock the memory of channel 1 to make updating of stimuli ram possible;
- `ri3165_SignalClear(instrumentHandle)`: Clear all previous declared signals;
- `ri3165_SignalAdd(instrumentHandle, 0, 2, 11, 0, 0)`: Define a sine: 2V amplitude (4Vpp), 11 periods, 0 degrees phase.
- `ri3165_SetStartAddress(instrumentHandle, 0)`: Set start address to 0;
- `ri3165_SetStopAddress(instrumentHandle, 1023)`: Set stop address at 1023.  $\text{StopAddress} - \text{StartAddress} + 1 = \text{number of samples}$ ;
- `ri3165_SignalToRam(instrumentHandle)`: Fill stimuli memory with the sine;
- `ri3165_SetClockDivider(instrumentHandle, 1)`: Divide clock by 1;
- `ri3165_SetAttenuator(instrumentHandle, -6)`: Attenuate signal 6 dB
- `ri3165_SetDCOffsetVoltage(instrumentHandle, 0)`: Set the dc offset voltage at 0V;
- `ri3165_SetFilter(instrumentHandle, 2)`: Connect 6 MHz filter path;
- `ri3165_SetClockSource(instrumentHandle, 1)`: Clock 1 (100 MHz) clock source;
- `ri3165_SetTriggerMode(instrumentHandle, 8, 0)`: Software trigger, positive edge;
- `ri3165_ConnectCard(instrumentHandle, 1)`: connect the output single ended;
- `ri3165_ResetToStartAddress(instrumentHandle)`: Reset address counter to start position (start address);
- `ri3165_SetLockMode(instrumentHandle, 1)`: Set channel in lock mode;
- `ri3165_SetSoftwareTriggerStatus(instrumentHandle, 1)`: Trigger channel 1;

## 3165 SOFTWARE UTILITY

## General

The 3165 Software Utility default has the following layout:



The instrument's configuration can be changed with the settings in the left panel. By pressing the button "Preview channel(s)" the signal configuration will be displayed in the graph. Depending on the setting "Active channel(s)" in the right panel, the graph will display channel A, B or both (or none).

For more details about the Instrument Settings see "**Adjusting the Instrument**". For more detail about the right panel (Instrument Update) see "**Updating the Instrument**".

The default unit for the X-axis is samples. For a comparison in time domain the X-axis unit can be set to time. Select Options->Time Domain Settings to configure this setting.

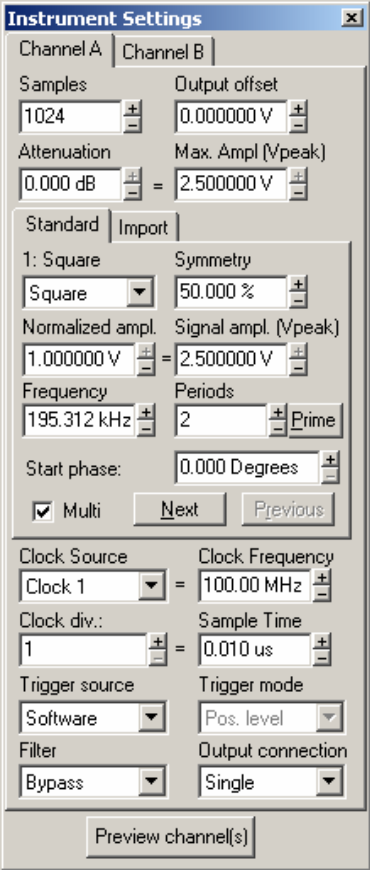
### Load and Save Instrument Configuration

The instrument configuration can be saved to disk. Select File->Save Settings. For loading a previous saved file select File->Load Settings.

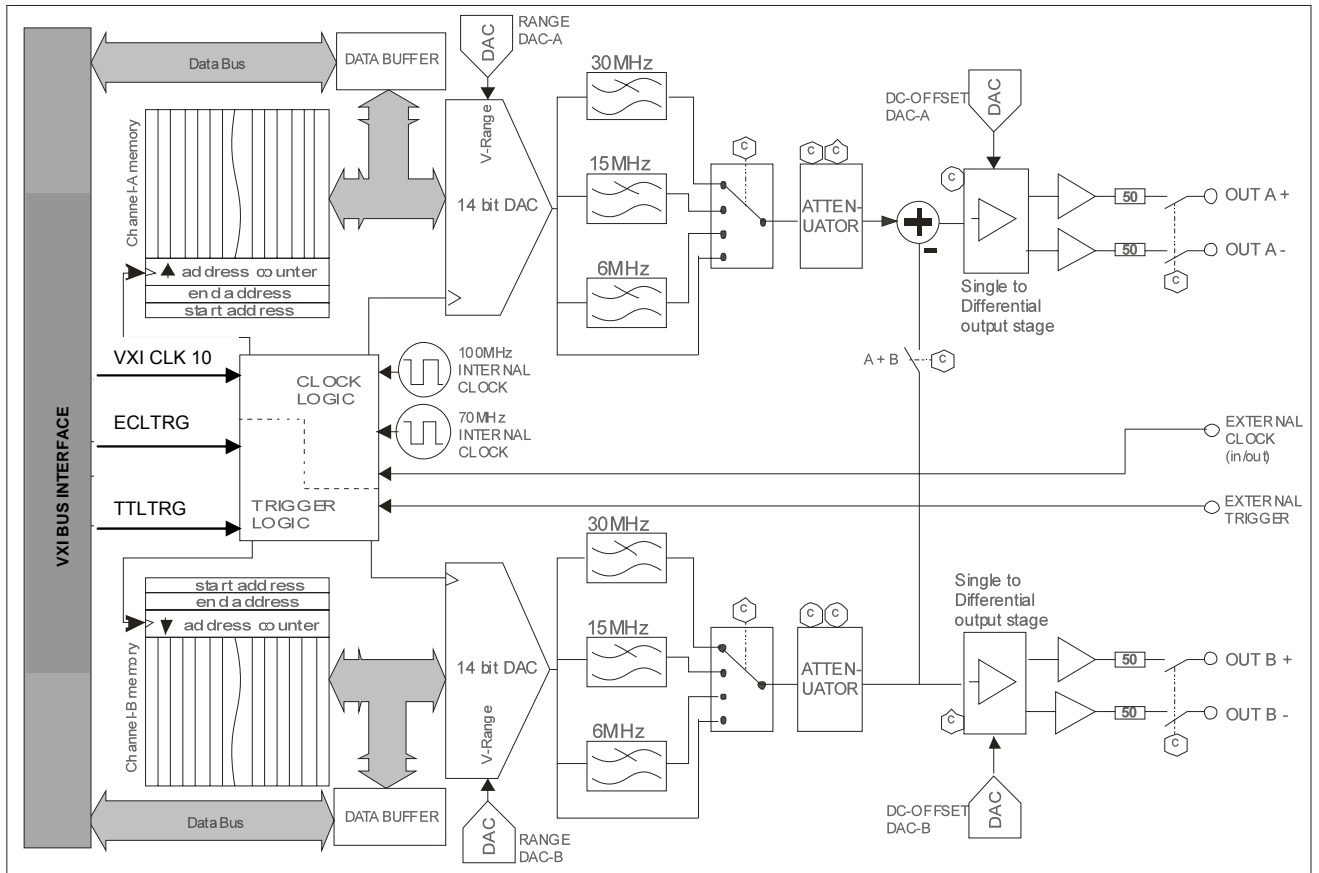
The default settings are save in the file "3165.set". Do not overwrite this file.

### Adjusting the Instrument

The instrument settings can be adjusted with the following panel:



It is easier to understand the settings if the hardware structure is known:



3165 Functional Block Diagram

The instrument can be configured by adjusting the following parameters:

- **Samples:** Number samples for the signal. If the signal is imported from a file the number of samples is determined by the file and cannot be altered.
- **Output offset:** This parameter will program the DC-Offset dac (see block diagram).
- **Attenuation:** The attenuator does change the following hardware elements:
  - The attenuator for the 3 dB step;
  - The Range DAC (close to the 14 bit main dac) for the attenuation between the 3 dB steps

- **Max. Ampl. (Vpeak):** This indicates the maximum possible output voltage. The attenuator determines the maximum amplitude (Vpeak). Changing this parameter will change the attenuator and determine the signal amplitude (see also “Standard Signals”).
- **Standard and Import:** The signal definition. See “Standard Signals” and “Import Signals” for more information.
- **Clock Source:** Select the desired clock source. If the front clock source is selected, the clock frequency can be altered. This is useful for a correctly calculated frequency and sample time.
- **Clock Frequency:** Only adjustable if the front clock is selected. This value will only influence the calculated frequency.
- **Clock div.:** The clock divider divides the Clock Source and determines the sample frequency.
- **Sample time:** The sample time can be adjusted by changing the clock divider. If this value is changed press “enter” to update the clock divider value (and the corresponding sample frequency).
- **Trigger Source:** Select the desired trigger source. See “Updating the Instrument” for how to send a software-trigger if “Software” is selected as source.
- **Trigger Mode:** The trigger mode is adjustable if the trigger source isn’t set to software. The following modes are available:
  - Positive level: Signal will run if trigger source has a logic high level
  - Negative level: Signal will run if trigger source has a logic low level
  - Positive edge re-trigger: Signal will run after the first positive edge. The next positive edge will stop the signal running.
  - Negative edge re-trigger: Signal will run after the first negative edge. The next negative edge will stop the signal running.
  - Positive edge continuous: Signal will run after a positive edge
  - Negative edge continuous: Signal will run after a negative edge
- **Filter:** Select a filter or just bypass the filters.



- **Output connection:** Connect the output single ended or differential. Channel B can be added (inversed) to channel A (single ended or differential).

Press “Preview channel(s)” to update the graph.

## Standard Signals

The standard signal can be configured by the following parameter:

- **Signal shape:** There are three standard shapes:
  - Sine
  - Square
  - Triangle
- **Symmetry:** For a square wave the symmetry parameter will change the duty cycle. For a triangle the symmetry will change the top of the triangle. 0% will result in a falling ramp, 100% in a rising ramp.
- **Normalize amplitude:** A value between –1 and +1. This parameter will adjust the signal amplitude. A value of 1 means a (main) DAC utilization of 100% (the DAC will get the maximum possible code) and the signal amplitude is equal to the settings “Max. amplitude”. A value of 0.5 corresponds with 50% DAC utilization.
- **Signal Ampl. (Vpeak):** The amplitude of the signal (Vpeak). The signal amplitude is influenced by the normalized amplitude and the attenuator setting. The maximum possible value of the signal amplitude is equal to the setting “Max. amplitude”. A value below “Max. amplitude” means the (main) DAC utilization is less than 100%.
- **Frequency:** The frequency of the signal. The frequency is calculated with:

$$f = p/N * fs$$

where :

f = signal frequency

p = number of periods

N = number of samples

fs = sample frequency

- **Periods:** Number of period for the signal. Changing the periods will change the frequency.
- **Start phase:** The start phase of the signal.

- **Multi:** Check multi for adding signals. This will allow you to create shapes other than the three standard shapes. Press the next button to define a new signal that will be added to the already configured signal(s). Press the previous button to go back to a previous declared signal.

## Import Signals

For more flexibility the software can import signals created in other applications. Signal files are text files with one column containing decimal or hexadecimal data, see next paragraphs for more details. The number of lines will determine the number of samples. Press the Browse button to select a signal file. The folder "Signals" of the application folder contains several examples of signal files.

## Decimal Import Data

The decimal import data has the following format:

0.123  
1.0  
-0.3  
etc.

A value of 1 will result in an amplitude of "Max. amplitude" (determined by the attenuator).

The imported data can be multiplied with any value.

## Hexadecimal Import Data (instrument main-DAC codes)

The hexadecimal import data has the following format:

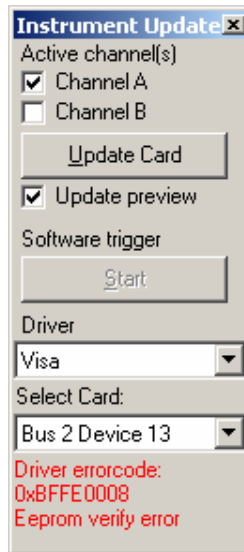
3000  
4F5  
5e2  
100a  
etc.

The instrument does have a 14 bit DAC (see block diagram). The import data will program this DAC, so the data can be in a range from 0 to 3FFF.

The imported data can be multiplied with any value.

Note: left shifting is equal to multiplying with 2, 4, 8 etc, for right shifting multiply with 0.5 , 0.25 , 0,125 etc.

## Updating The Instrument



This panel is used to update and select the card. It does have the following items:

- **Active channel(s):** Select the channel(s) to be updated.
- **Update Card:** Send the (changed) Instruments settings to the hardware. Only the active channel(s) will be updated.
- **Update preview:** The graph will also be updated with a card update if this item is checked.
- **Software trigger:** Send a software trigger to the instrument. Button is only enabled is card is updated and “Software” is selected to be the trigger source.
- **Driver:** Select the VISA driver that is to be used
- **Select card:** Select the card to be updated. When using the VISA driver, use NI-MAX to locate the card.
- **Status:** Display the driver error code if any.

This page was left intentionally blank.

## Chapter 10

# 3165 CALIBRATION

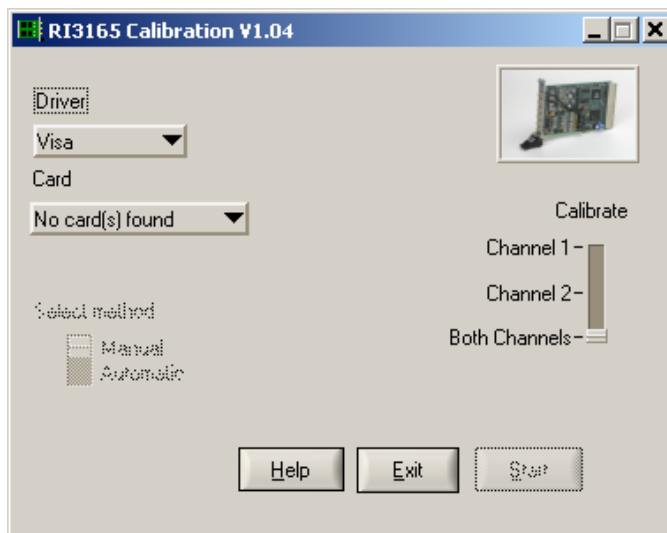
---

### The Calibration Procedure

Description:

Close the card session. All resources belonging to the card will be released.

#### 1) Start the Calibration Utility

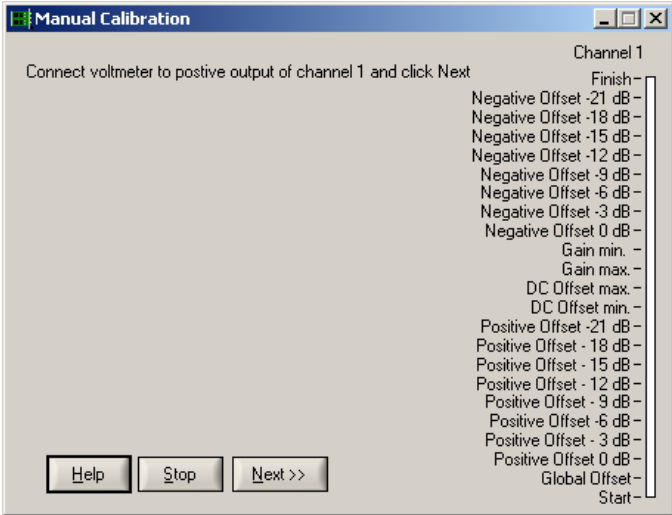


#### 2) Select the card

When using VISA, use NI-MAX to locate the card.

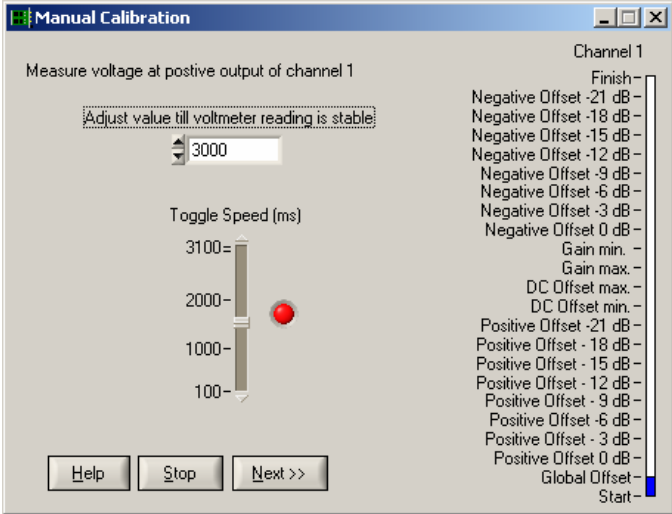
### 3) Select channel(s)

Select the channel(s) to be calibrated and press the Start button.



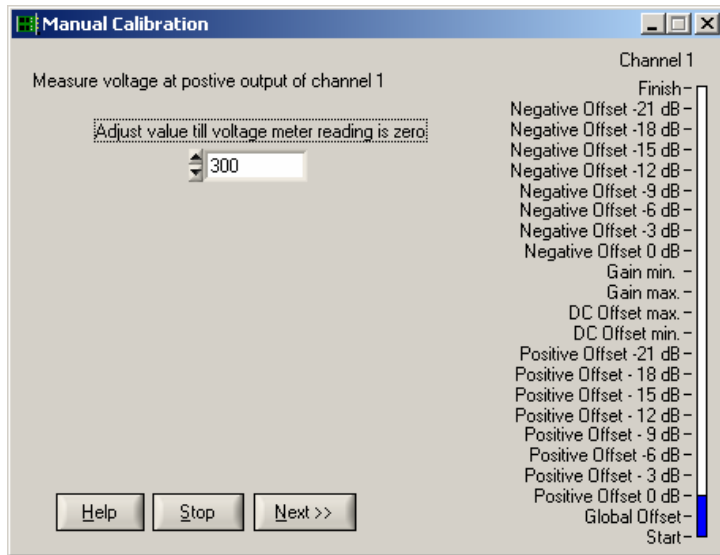
### 4) Connect Voltmeter

Connect the positive output of the channel to be calibrated to a(n) (accurate) voltmeter (e.g. Keithley 2000 or Agilent 3458A). Press the Next button to continue.



### 5) Offset balance calibration

The output voltage should toggle between two voltages. The toggle speed can be adjusted with the slide “Toggle Speed (ms)”. Change the calibration value till the voltage at the output varies less than 300 uV. Press the Next button to continue.



### 6) Offset Calibration at Positive Output

Change the calibration value till the output voltage is 0 volt (as close as possible). Performed the offset calibration for the 8 steps attenuator positions (0 dB ... 21 dB, 3 dB step).

### 7) DC Offset min. and max. voltages

Fill in the measured voltages. Press enter after changing the values!

### 8) Gain min. and max. voltages

Fill in the measured voltages. Press enter after changing the values!

### **9) Offset Calibration at Negative Output**

Connect the voltmeter to the negative output. Perform the same calibration steps as performed at step 6.

### **10) Finish**

After calibrating the channel(s) press finish to store the new calibration value in the on board EEPROM of the 3165. If the Stop button is pressed during the installation procedure, the new calibration values are lost.



## Chapter 11

# SOFTWARE DRIVER

---

**Overview**      There are two drivers for this unit and they are as follows:

<b>Model</b>	<b>Description</b>	<b>Part Number</b>
2165 Driver	Digitizer Software Drive	922029-001
3165 Driver	DAC Software Drive	922030-001

This page was left intentionally blank.

## Chapter 12

# PRODUCT SUPPORT

---

### Product Support

EADS North America Defense Test and Services, Inc. has a complete Service and Parts Department. If you need technical assistance or should it be necessary to return your product for repair or calibration, call 1-800-722-3262. If parts are required to repair the product at your facility, call 1-949-859-8999 and ask for the Parts Department.

When sending your instrument in for repair, complete the form in the back of this manual.

For worldwide support and the office closest to your facility, refer to the website for the most complete information <http://www.eads-nadefense.com>.

### Warranty

Use the original packing material when returning the 6088-14 Series to EADS North America Defense Test and Services, Inc. for calibration or servicing. The original shipping container and associated packaging material will provide the necessary protection for safe reshipment.

If the original packing material is unavailable, contact EADS North America Defense Test and Services, Inc. Customer Service at 1-800-722-3262 for information.

**REPAIR AND CALIBRATION REQUEST FORM**

To allow us to better understand your repair requests, we suggest you use the following outline when calling and include a copy with your instrument to be sent to the EADS North America Defense Test and Service, Inc. Repair Facility.

Model \_\_\_\_\_ Serial No. \_\_\_\_\_ Date \_\_\_\_\_

Company Name \_\_\_\_\_ Purchase Order # \_\_\_\_\_

Billing Address \_\_\_\_\_

City

State/Province

Zip/Postal Code

Country

Shipping Address \_\_\_\_\_

City

State/Province

Zip/Postal Code

Country

Technical Contact \_\_\_\_\_ Phone Number ( ) \_\_\_\_\_

Purchasing Contact \_\_\_\_\_ Phone Number ( ) \_\_\_\_\_

1. Describe, in detail, the problem and symptoms you are having. Please include all set up details, such as input/output levels, frequencies, waveform details, etc.

\_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_

2. If problem is occurring when unit is in remote, please list the program strings used and the controller type.

\_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_

3. Please give any additional information you feel would be beneficial in facilitating a faster repair time (i.e., modifications, etc.)

\_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_

4. Is calibration data required?      Yes    No    (please circle one)

Call before shipping      Ship instruments to nearest support office.

Note: We do not accept "collect" shipments.