SOFTWARE
USER MANUAL

MODEL 2165

PUBLICATION NO. 980898-002

RACAL INSTRUMENTS

United States

(Corporate Headquarters and Service Center)
4 Goodyear Street, Irvine, CA 92618
Tel: (800) 722-2528, (949) 859-8999; Fax: (949) 859-7139

5730 Northwest Parkway Suite 700, San Antonio, TX 78249
Tel: (210) 699-6799; Fax: (210) 699-8857

Europe

(European Headquarters and Service Center)
18 Avenue Dutartre, 78150 LeChesnay, France
Tel: +33 (0)1 39 23 22 22; Fax: +33 (0)1 39 23 22 25

29-31 Cobham Road, Wimborne, Dorset BH21 7PF, United Kingdom
Tel: +44 (0) 1202 872800; Fax: +44 (0) 1202 870810

Via Milazzo 25, 20092 Cinisello B, Milan, Italy
Tel: +39 (0)2 6123 901; Fax: +39 (0)2 6129 3606

Racal Instruments Group Limited, Technologie Park, D-51429 Bergisch Gladbach, Germany
Tel: +49 2204 844205; Fax: +49 2204 844219

info@racalinstruments.com
sales@racalinstruments.com
helpdesk@racalinstruments.com
http://www.racalinstruments.com
info@racalinstruments.de
www.racalinstruments.de

PUBLICATION DATE: August 24, 2004

THANK YOU FOR PURCHASING THIS RACAL INSTRUMENTS PRODUCT

For this product, or any other Racal Instruments 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.racalinstruments.com/downloads

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

info@racalinstruments.com.

WARRANTY STATEMENT

All Racal Instruments, Inc. products are designed and manufactured to exacting standards and in full conformance to Racal Instruments ISO 9000/2000 procedures.

For the specific terms of your standard warranty, or optional extended warranty or service agreement, contact your Racal Instruments 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:	Helpdesk@racalinstruments.com	
Telephone:	+1 800 722 3262	(USA)
	+44(0) 8706 080134	(UK)
Fax:	+1 949 859 7309	(USA)
	+44(0) 1628 662017	(UK)

RETURN of PRODUCT

Authorization is required from Racal Instruments before you send us your product for service or calibration. Call your nearest Racal Instruments support facility. A list is located on the last page of this manual. If you are unsure where to call, contact Racal Instruments, Inc. Customer Support Department in Irvine, California, USA at 1-800-722-3262 or 1-949-859-8999 or via fax at 1-949-859-7139. We can be reached at: helpdesk@racalinstruments.com.

PROPRIETARY NOTICE

This document and the technical data herein disclosed, are proprietary to Racal Instruments, and shall not, without express written permission of Racal Instruments, be used, in whole or in part to solicit quotations from a competitive source or used for manufacture by anyone other than Racal Instruments. 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 Racal Instruments.

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. Racal Instruments 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 Racal Instruments product could create a situation where personal injury or death could occur. Should Buyer purchase Racal Instruments product for such unintended application, Buyer shall indemnify and hold Racal Instruments, 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.

CAUTION
SENSITIVE ELECTRONIC DEVICES
DO NOT SHIP OR STORE NEAR
STRONG ELECTROSTATIC,
ELECTROMAGNETIC, MAGNETIC OR
RADIOACTIVE FIELDS

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.

TABLE OF CONTENTS

Introduction

The Racal Instruments' Model 2165 is a 70MHz dual channel PXI Waveform Digitizer. For software development and integration in a PXI 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 RI 2165 driver consist 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 manual describes the functions of the RI2165_32.dll.

1 RI 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, RI 2165.llb
4) A LabWindows driver (function tree) RI 2165.fp

The following low level drivers can be installed:

1) A kernel mode pxi driver, appl_pxi.sys;
2) VISA from National Instruments.

1.1 Installation

Install the RI 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.

The PXI Kernel Driver cannot be installed on Windows95 or Windows NT. This selection will be disabled if one of these operating systems is detected.

After installation shutdown the computer and place the RI 2165 in the system. After turning on the computer the operation system should automatically detect the new hardware and install the low level driver.

1.2 Uninstalling The Low Level Driver

Before switching from low-level driver (Visa <-> Kernel Driver), uninstall the current low-level driver.

For uninstalling the low level driver, perform the following steps:

1. Start the Device Manager
2. Select the “ RI 2165 PXI driver” and uninstall the driver;
3. Go to the Windows inf-directory (e.g. WinNT\inf, hidden directory).
4. Delete the RI 2165_xxxx.inf file. The addition xxxx indicates the Windows Operation System version.
5. If installed with the pxi kernel mode driver, go the Windows sub-directory System\Drivers (e.g. C:\WinNT\System\Drivers).
6. Delete the appl_pxi.sys file. Do NOT delete this file if other cards (other than the RI 2165 cards) need the pxi kernel driver!

1.3 “Manual” Installation of the Low Level Driver

1. Copy the corresponding inf-file to the Windows Inf-subdirectory (e.g. C:\WinNT\inf). The inf-files can be found in de following directories of the CD-ROM:

   For Visa installation:

   Directory : \Driver\Visa

Windows9x	“RI 2165_9x.inf”
WindowsNT4	“RI 2165_nt4.inf”
Windows2000	“RI 2165_nt5.inf”
WindowsXP	“RI 2165_nt5.inf”

   For the PXI kernel mode driver installation:

   Directory: \Driver\Kernel

Windows98	“RI 2165_98.inf”
Windows2000	“RI 2165_2000.inf”
WindowsXP	“RI 2165_xp.inf”

   The inf-subdirectory is a hidden directory.

2. If the pxi kernel mode driver is installed, copy also the appl_pxi.sys file. This file can be found in the directory \Drivers\Kernel\Winxx, where xx indicates the operating system. Copy this file to the Windows sub-directory \System\Drivers (e.g. C:\WinNT\System\Drivers);
3. Turn the system off and place the RI 2165 in the system;
4. Turn the system on and reboot the host computer;
5. The operation system should detect new hardware;

Be sure driver signing is set to Ignore or Warn, when installing the Racal pxi kernel driver on a Win2000 or WinXP system.

If the “Add new hardware” wizard doesn't start or something went wrong during installation, start the Device Manager. Select the device (normally marked with a question mark, if the driver could not be loaded) and select properties. Then install/reinstall the driver.

2 RI 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

All functions use the standard calling conventions (stdcall or WINAPI). Every function returns the RI 2165_status (type: 32-bit integer). A negative value corresponds to an error. After a successful completion the return status is RI 2165_SUCCESS, which corresponds to a 0. The possible status codes can be found in Chapter 3.

2.1 RI 2165_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 2^32
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

2.2 RI 2165_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 2^32
testStatus	unsigned long*	out	Test Status	0 A & B not ready 1 Channel A ready 2 Channel B ready 3 A & B ready

2.3 RI 2165_Close(ci)

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

Parameters:

Name	Type	Direction	Description	Value
ci	unsigned long	in	card identifier	0 to 2^32

2.4 RI 2165_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 2^32
ADCCode	unsigned long	in	mode	0 to 2^32
voltage	double*	out	voltage	corresponding voltage

2.5 RI 2165_ConnectCard(instrumentHandle , positveInputConnection, 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 2^32
positiveInputConnection	unsigned long	in	positive input	see description
negativeInputConnection	unsigned long	in	negative input	see description

2.6 RI 2165_GetActiveChannel(ci , 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 SetActiveChannel().

Parameters:

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

2.7 RI 2165_GetAddressCounter(instrumentHandle, 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 2^32
addresscounter	unsigned long*	out	Position of the address-counter	0 to 2^19

2.8 RI 2165_GetCardAddress(instrumentHandle, address)

Description:
Returns the physical address of the card. This function can be used to determine the physical (start) address of the card when the kernel mode (non-visa) driver is installed. This address can also be found in the Device Manager of Windows (under Resources). If 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 2^32
address	unsigned long *	out	address of card	0 to 2^32

2.9 RI 2165_GetCardConnection(instrumentHandle, positiveInputConnection, negativeInputConnection)

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 2^32
positiveInputConnection	unsigned long*	out	positive input	see description
negativeInputConnection	unsigned long*	out	negative input	see description

2.10 RI 2165_GetCardList(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

2.11 RI 2165_GetCardNumber(instrumentHandle , cardNumber)

Description:
This function returns the card number. This function is only useful if the card is installed with the kernel mode (non-visa) driver. The card number can be used as a reference in your program. The number of available cards (installed with the non-visa driver) can be determined with the function GetNumberOfCards().

Parameters:

Name	Type	Direction	Description	Value
instrumentHandle	unsigned long	in	Instrument Handle	0 to 2^32
cardNumber	unsigned long *	out	card number	0 to 256

2.12 RI 2165_GetClockDivider (instrumentHandle, clockDivider)

Description:
This function returns the clock divider value.

Parameters:

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

2.13 RI 2165_GetClockSource (instrumentHandle, clockSource)

Description:
This function returns the clock source.

Parameters:

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

2.14 RI 2165_GetDCOffsetLimitVoltages (instrumentHandle, positiveVoltage, negativeVoltage)

Description:
Returns the DC Offset Limit Voltages. See function SetDCOffsetLimitVoltages(..).

Parameters:

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

2.15 RI 2165_GetDCOffsetVoltage(instrumentHandle , 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 SetDCOffsetVoltage().

Parameters:

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

2.16 RI 2165_GetErrorMessage(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 3
message	char*	out	error message	see chapter 3

2.17 RI 2165_GetFilter (instrumentHandle, filterStatus)

Description:
This function returns the filter position.

Parameters:

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

2.18 RI 2165_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 2^32
gainCalibrationCode	unsigned long*	out	gain cal. code	0 to 2^10

2.19 RI 2165_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 2^32
voltage	double*	out	measured voltage	depends on range
code	unsigned long*	out	adc code	0 to 2^14
average	unsigned long	in	averages	1 to 2^19
timeOut	unsigned long	in	time out	0 to 2^32 no timeout it time out is 0

2.20 RI 2165_GetLoopMode (instrumentHandle, loopMode)

Description:
This function returns the Loop Mode status.

Parameters:

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

2.21 RI 2165_GetNumberOfCards(cards)

Description:
This function returns the number of available cards in the system. This function will only return a value above 0 if there are cards installed with the kernel mode (non-visa) driver.

Parameters:

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

2.22 RI 2165_GetOffsetCalDacCode(ci, code)

Description:
This function returns the offset calibration dac code.

Parameters:

Name	Type	Direction	Description	Value
instrumentHandle	unsigned long	in	Instrument Handle	0 to 2^32
code	unsigned long*	out	code	0 to 2^10

2.23 RI 2165_GetRange (instrumentHandle, range)

Description:
This function returns the range setting.

Parameters:

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

2.24 RI 2165_GetRevision(revision)

Description:
This function returns the driver revision.

Parameters:

Name	Type	Direction	Description	Value
revision	unsigned long*	out	driver revision	1 to 2^32

2.25 RI 2165_GetSampleDivider (instrumentHandle, sampleDivider)

Description:
The function returns the sample divider value.

Parameters:

Name	Type	Direction	Description	Value
instrumentHandle	unsigned long	in	Instrument Handle	0 to 2^32
sampleDivider	unsigned long*	out	sample dividider	1 to 65536

2.26 RI 2165_GetSoftwareTriggerStatus(instrumentHandle, triggerstatus)

Description:
This function returns the trigger status.

Parameters:

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

2.27 RI 2165_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 2^32
triggerSource	unsigned long*	out	trigger source	0 Front 1 PXI 0 2 PXI 1 3 PXI 2 4 PXI 3 5 PXI 4 6 PXI 5 7 PXI Star 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

2.28 RI 2165_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 Close() in between.

Parameters:

Name	Type	Direction	Description	Value
Bus	unsigned long	in	PXI bus number	0 to 2^32
Device	unsigned long	in	PXI device number	0 to 2^32
instrumentHandle	unsigned long*	out	Instrument Handle	0 to 2^32

2.29 RI 2165_InitCard(card , instrumentHandle)

Description:
Init a card with the kernel driver (non-visa driver). This function starts a card session. Call this routine if the card is installed with the kernel driver (non-visa) driver. To determine the number of available cards call GetNumberOfCards(). To determine the physical address of the card call GetCardAddress(). The card addresses can also be found in the Windows Device Manager. 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 Close() in between.

Parameters:

Name	Type	Direction	Description	Value
card	unsigned long	in	card to open	0 to 2^32
instrumentHandle	unsigned long*	in	pointer to the variable for the instrument handle	0 to 2^32

2.30 RI 2165_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 2^32
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

2.31 RI 2165_ReadAdcResults (instrumentHandle, startPosition, samples, buffer, includeStatusBits)

Description:
This function reads the latest test results from the pd172. 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 2^32
startPosition	unsigned long	in	start position	0 to 2^19
samples	unsigned long	in	nr. of samples	1 to 2^19
buffer	unsigned long*	out	buffer for adc codes	
includeStatusBits	unsigned long	in	include the status bits	0 not included 1 included

2.32 RI 2165_ReadEeprom(instrumentHandle , eeaddress , data)

Description:
This function reads a 16-bit word from the serial eeprom at a serial eeprom address determined by eeaddress. If the eeprom address is previously written with the function WriteEeprom(), the lower byte will correspond to the byte written with the function 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 2^32
eeadress	unsigned long	in	Eeprom address	0 to 255
data	unsigned long*	out	read data	0 to 2^16

2.33 RI 2165_ReadId(instrumentHandle , id)

Description:
This function reads the card ID. A card ID can be set with the function 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 2^32
id	unsigned long*	out	card id	0 to 2^32

2.34 RI 2165_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 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 2^32
data	unsigned long*	out	read data	0 to 2^16

2.35 RI 2165_ReadRamBuffer(instrumentHandle , length , buf32, dataInterpretation)

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 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:

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

2.36 RI 2165_SetActiveChannel(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	Instrument Handle	0 to 2^32
channel	unsigned long	in	active channel	1 Channel-A 2 Channel-B

2.37 RI 2165_SetAddressCounter (instrumentHandle, addressCounterPosition)

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

Parameters:

Name	Type	Direction	Description	Value
instrumentHandle	unsigned long	in	Instrument Handle	0 to 2^32
addressCounterPosition	unsigned long	in	address counter position	0 to 2^19

2.38 RI 2165_SetClockDivider(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 SetClockSource() and determines the sample rate.

Parameters:

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

2.39 RI 2165_SetClockSource(instrumentHandle, clocksource)

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 SetClockDivider().

Parameters:

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

2.40 RI 2165_SetDCOffsetCode(instrumentHandle , 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 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 2^32
code	unsigned long	in	code for DC offset DAC	0 to 2^16
connection	unsigned long	in	connect	0 disconnect from negative input 1 connect to negative input

2.41 RI 2165_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 2^32
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

2.42 RI 2165_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: ConnectCard(ci, 0, 0)
- Filter bypass: SetFilter(ci, 0)
- Program DC offset dac at maximal voltage and connect offset voltage with input: SetDCOffsetCode(ci,0xFFFF,1)
- Measure DC offset voltage with accurate voltage meter
- Program DC offset dac at minimal voltage and connect offset voltage with input: SetDCOffsetCode(ci,0x0,1)
- Measure DC offset voltage with accurate voltage meter
- Disconnect offset voltage: SetDCOffsetCode(ci,0x8000,0)
- Call this routine with measured voltages
- Call 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 PostiveVoltage 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 2^32
posvolt	double	in	measured positive voltage	> 5V
negvolt	double	in	measured negative voltage	< -5V

2.43 RI 2165_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 2^32
filter	unsigned long	in	filter select	0 = Disconnect 1 = Bypass 2 = 6 MHz filter 3 = 15 MHz filter 4 = 30 MHz filter

2.44 RI 2165_SetGainCalCode (instrumentHandle, code)

Description:
Write a code to the gain calibration dac.

Parameters:

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

2.45 RI 2165_SetLockMode(instrumentHandle , 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 2^32
lock	unsigned long	in	lock/unlock active channel	1 = lock 0 = unlock

2.46 RI 2165_SetLoopMode (instrumentHandle, 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 2^32
loopmode	unsigned long	in	loopmode	0 = not in loopmode 1 = loopmode

2.47 RI 2165_SetOffsetCalCode(instrumentHandle, code)

Description:
This function programs the offset calibration dac.

Parameters:

Name	Type	Direction	Description	Value
instrumentHandle	unsigned long	in	Instrument Handle	0 to 2^32
code	unsigned long	in	code	0 to 2^10

2.48 RI 2165_SetRange(instrumentHandle , range)

Description:
This function will set the input voltage range.

Parameters:

Name	Type	Direction	Description	Value
instrumentHandle	unsigned long	in	Instrument Handle	0 to 2^32
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

2.49 RI 2165_SetSampleDivider (instrumentHandle, sampleDividerValue)

Description:
This function will program the sample divider. The sample divider determines how many samples will be stored. E.g. 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 2^32
sampleDividerValue	unsigned long	in	sample divider	1 to 65536

2.50 RI 2165_SetSoftwareTriggerStatus(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 SetTriggerMode() to enable the software trigger mode.

Parameters:

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

2.51 RI 2165_SetTriggerInput(instrumentHandle , triggersource , triggermode)

Description:
Select trigger source and trigger mode. In lock mode (Set with the function 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 2^32
triggersource	unsigned long	in	select trigger source	0 = FRONT PANEL TRIG 1 = PXI TRIG 0 2 = PXI TRIG 1 3 = PXI TRIG 2 4 = PXI TRIG 3 5 = PXI TRIG 4 6 = PXI TRIG 5 7 = PXI STAR 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)

2.52 RI 2165_StoreCalibrationData(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 2^32

2.53 RI 2165_WriteEeprom(instrumentHandle , eeaddress , data)

Description:
Write a data byte to the eeprom address (defined with eeadress) 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 2^32
eeaddress	unsigned long	in	eeprom address	0 to 255
data	unsigned long	in	byte to be written	0 to 255

2.54 RI 2165_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 ReadId().

Parameters:

Name	Type	Direction	Description	Value
instrumentHandle	unsigned long	in	Instrument Handle	0 to 2^32
id	unsigned long	in	card id	0 to 2^32

2.55 RI 2165_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 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 2^32
data	unsigned long	in	data to write	0 to 2^14

2.56 RI 2165_WriteRamBuffer(instrumentHandle , 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 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 2^32
length	unsigned long	in	number of data elements to be written	0 to 2^18
buffer	unsigned long*	in	reference to buffer with data to be written	0 to 2^32

3 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
RI 2165_SUCCESS	0x0	No error(s)

General error codes:

Constant name	Value	Description
RI 2165_ERROR_INVALID_CHANNEL	0xBFFE0004	Invalid channel
RI 2165_ERROR_INVALID_PARAMETER	0xBFFE0005	Invalid parameter
RI 2165_ERROR_MEMORY	0xBFFE0006	Could not allocate memory
RI 2165_ERROR_NO_SIGNALDEF	0xBFFE0007	No signal defined
RI 2165_ERROR_EEPROMCHECK	0xBFFE0008	Eeprom verify error
RI 2165_ERROR_MEMTEST	0xBFFE000B	Memory test failed
RI 2165_ERROR_CALIBRATION	0xBFFE000C	Auto calibration error
RI 2165_ERROR_TIMEOUT	0xBFFE000D	Time-out expired

Kernel mode driver (non-visa) errors codes

Constant name	Value	Description
RI 2165_ERROR_INV_OBJECT	0xBFFF000E	invalid card reference
RI 2165_ERROR_ALLOC	0xBFFF003C	Insufficient system resources
RI 2165_ERROR_INV_RSRC_NAME	0xBFFF0012	invalid resource name
RI 2165_ERROR_OPEN_FAILURE	0xBFFE0001	Could not open card
RI 2165_ERROR_READ_FAILURE	0xBFFE0002	Error during reading
RI 2165_ERROR_WRITE_FAILURE	0xBFFE0003	Error during writing

VISA error codes

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

Product Support

Racal Instruments 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 closes to your facility, refer to the Support Offices section on the following page.

Warranty

Use the original packing material when returning the 2165 to Racal Instruments 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 Racal Instruments Customer Service for information.

Support Offices

RACAL INSTRUMENTS

United States

(Corporate Headquarters and Service Center)
4 Goodyear Street, Irvine, CA 92618
Tel: (800) 722-2528, (949) 859-8999; Fax: (949) 859-7139

5730 Northwest Parkway Suite 700, San Antonio, TX 78249
Tel: (210) 699-6799; Fax: (210) 699-8857

Europe

(European Headquarters and Service Center)
18 Avenue Dutartre, 78150 LeChesnay, France
Tel: +33 (0)1 39 23 22 22; Fax: +33 (0)1 39 23 22 25

29-31 Cobham Road, Wimborne, Dorset BH21 7PF, United Kingdom
Tel: +44 (0) 1202 872800; Fax: +44 (0) 1202 870810

Via Milazzo 25, 20092 Cinisello B, Milan, Italy
Tel: +39 (0)2 6123 901; Fax: +39 (0)2 6129 3606

Racal Instruments Group Limited, Technologie Park,
D-51429 Bergisch Gladbach, Germany
Tel: +49 2204 844205; Fax: +49 2204 844219

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 Racal Instruments 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