SOFTWARE
USER MANUAL

MODEL 3165

PUBLICATION NO. 980899-002

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

Technologie Park, Friedrich Ebert Strasse, 51429 Bergisch Gladbach, Germany
Tel: +49 (0) 2204 844200; Fax: +49 (0) 2204 844219

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

RACAL INSTRUMENTS

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 3165 (980899-002) is a 100MHz dual-channel PXI based Arbitrary Waveform Generator (AWG). 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 3165 driver consists of a Windows® dynamic link library, the RI3165 14102_32.DLL

For Labview users the driver includes also a Labview® library with the RI3165 14102_32.dll driver functions.

This manual describes the functions of the RI3165 14102_32.dll.

1 3165 Driver Package

The driver includes the following items:

1) A low level driver for direct communication;
2) The user mode driver, RI3165 14102_32.dll;
3) A Labview® library, RI3165 14102.llb
4) A LabWindows® driver (function tree) RI3165 14102.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 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.

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 3165 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 “Racal 3165 PXI driver” and uninstall the driver;
3. Go to the Windows inf-directory (e.g. WinNT\inf, hidden directory).
4. Delete the RI3165 14102_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 RI3165 14102 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	“RI3165 14102_9x.inf”
WindowsNT4	“RI3165 14102_nt4.inf”
Windows2000	“RI3165 14102_nt5.inf”
WindowsXP	“RI3165 14102_nt5.inf”

For the PXI kernel mode driver installation:

Directory: \Driver\Kernel

Windows98	“RI3165 14102_98.inf”
Windows2000	“RI3165 14102_2000.inf”
WindowsXP	“RI3165 14102_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 3165 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 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

All functions use the standard calling conventions (stdcall or WINAPI). Every function returns the RI3165 14102_status (type: 32-bit integer). A negative value corresponds to an error. After a successful completion the return status is 3165_SUCCESS, which corresponds to a 0. The possible status codes can be found in Chapter 3.

2.1 RI3165 14102_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

2.2 RI3165 14102_ConnectCard(ci , 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
ci	unsigned long	in	card identifier	0 to 2^32
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

2.3 RI3165 14102_GetActiveChannel(ci , 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 SetActiveChannel().

Parameters:

Name	Type	Direction	Description	Value
ci	unsigned long	in	card identifier	0 to 2^32
channel	unsigned long*	out	active channel	1 Channel-A 2 Channel-B

2.4 RI3165 14102_GetAddressCounter(ci , addresscounter)

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

Parameters:

Name	Type	Direction	Description	Value
ci	unsigned long	in	card identifier	0 to 2^32
addresscounter	unsigned long*	out	position of the address-counter	0 to 2^18

2.5 RI3165 14102_GetAttenuator(ci , attenuation)

Description:
This function returns the position of the attenuator from the active channel. The attenuator can be programmed with the function SetAttenuator().

Parameters:

Name	Type	Direction	Description	Value
Ci	unsigned long	in	card identifier	0 to 2^32
attenuation	double*	out	current attenuation value	0 to –24 (dB)

2.6 RI3165 14102_GetCardAddress(ci , 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
Ci	unsigned long	in	card identifier	0 to 2^32
address	unsigned long *	out	address of card	0 to 2^32

2.7 RI3165 14102_GetCardConnection(ci , 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
ci	unsigned long	in	card identifier	0 to 2^32
cc	unsigned long *	out	card connect status	0 disconnect 1 single ended 2 differential 3 A + B inverterted single ended 4 A + B inverterted differential

2.8 RI3165 14102_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

2.9 RI3165 14102_GetCardNumber(ci , card)

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
ci	unsigned long	in	card identifier	0 to 2^32
card	unsigned long *	out	card number	0 to 256

2.10 RI3165 14102_GetDCOffsetVoltage(ci , 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 SetDCOffsetVoltage().

Parameters:

Name	Type	Direction	Description	Value
ci	unsigned long	in	card identifier	0 to 2^32
voltage	double *	out	voltage of offset DAC	-2.5 to +2.5 volt

2.11 RI3165 14102_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

2.12 RI3165 14102_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

2.13 RI3165 14102_GetOffsetCalDacCode(ci, code)

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

Parameters:

Name	Type	Direction	Description	Value
ci	unsigned long	in	card identifier	0 to 2^32
code	unsigned long*	out	code	0 to 2^12

2.14 RI3165 14102_GetOutputOffsetCalDacCode(ci, dac,code)

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

Parameters:

Name	Type	Direction	Description	Value
ci	unsigned long	in	card identifier	0 to 2^32
dac	unsigned long	in	calibration dac	1 to 8
code	unsigned long*	out	code	0 to 2^10

2.15 RI3165 14102_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

2.16 RI3165 14102_GetTriggerStatus(ci , triggerstatus)

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

Parameters:

Name	Type	Direction	Description	Value
ci	unsigned long	in	card identifier	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

2.17 RI3165 14102_GetStartAddress(ci , startaddress)

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
ci	unsigned long	In	card identifier	0 to 2^32
startaddress	unsigned long*	Out	start address	0 to 2^18

2.18 RI3165 14102_GetStopAddress(ci , stopaddress)

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

Parameters:

Name	Type	Direction	Description	Value
ci	unsigned long	in	card identifier	0 to 2^32
stopaddress	unsigned long*	out	stop address	0 to 2^19

2.19 RI3165 14102_Init(bus , device , ci)

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 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
Ci	unsigned long*	in	card identifier	0 to 2^32

2.20 RI3165 14102_InitCard(card , ci)

Description:
Initiate 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 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 Close() in between.

Parameters:

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

2.21 RI3165 14102_LoadArbitraryWaveForm(ci, 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
ci	unsigned long	in	card identifier	0 to 2^32
startaddress	unsigned long	in	start address	0 to 2^19
length	unsigned long	in	length of waveform	0 to 2^19
waveform	double*	in	reference to waveform data	0 to 2^32

2.22 RI3165 14102_Read(ci , 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
ci	unsigned long	in	card identifier	0 to 2^32
offset	unsigned long	in	card offset address (card register)	see hardware manual
data	unsigned long*	out	read data	register dependent

2.23 RI3165 14102_ReadEeprom(ci , 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
ci	unsigned long	in	card identifier	0 to 2^32
eeadress	unsigned long	in	Eeprom address	0 to 255
data	unsigned long*	out	read data	0 to 2^16

2.24 RI3165 14102_ReadId(ci , 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
ci	unsigned long	in	card identifier	0 to 2^32
id	unsigned long*	out	card id	0 to 2^32

2.25 RI3165 14102_ReadRam(ci, 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 SetStartAddress(). After this function call the address-counter is incremented with one step.

Parameters:

Name	Type	Direction	Description	Value
ci	unsigned long	in	card identifier	0 to 2^32
data	unsigned long*	out	read data	0 to 2^16

2.26 RI3165 14102_ReadRamBuffer(ci , 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 SetStartAddress(). After this function call the address-counter is incremented with “length” steps.

Parameters:

Name	Type	Direction	Description	Value
ci	unsigned long	in	card identifier	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

2.27 RI3165 14102_ResetToStartAddress(ci)

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 SetStartAddress().

Parameters:

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

2.28 RI3165 14102_SetActiveChannel(ci , channel)

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

Parameters:

Name	Type	Direction	Description	Value
ci	unsigned long	in	card identifier	0 to 2^32
channel	unsigned long	in	active channel	1 Channel-A 2 Channel-B

2.29 RI3165 14102_SetAttenuator(ci , 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 SignalAdd() will be attenuated with the attenuator value.

Parameters:

Name	Type	Direction	Description	Value
ci	unsigned long	in	card identifier	0 to 2^32
attenuation	double	in	attenuator value	0 to –24 (dB)

2.30 RI3165 14102_SetClockDivider(ci , 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 update sample rate of the signal (and consequently the signal frequency).

Parameters:

Name	Type	Direction	Description	Value
ci	unsigned long	in	card identifier	0 to 2^32
clockdevider	unsigned long	in	clock divider value	1 to 256

2.31 RI3165 14102_SetClockSource(ci , 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 SetClockDivider().

Parameters:

Name	Type	Direction	Description	Value
ci	unsigned long	in	card identifier	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

2.32 RI3165 14102_SetDacCode(ci , 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
ci	unsigned long	in	card identifier	0 to 2^32
code	unsigned long	in	code for main-DAC	0 to 2^14

2.33 RI3165 14102_SetDCOffsetDacCode(ci , 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 SetDCOffsetVoltage() will program the DC offset voltage to a desired level.

Parameters:

Name	Type	Direction	Description	Value
ci	unsigned long	in	card identifier	0 to 2^32
code	unsigned long	in	code for DC offset DAC	0 to 2^16

2.34 RI3165 14102_SetDCOffsetVoltage(ci , 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
ci	unsigned long	in	card identifier	0 to 2^32
voltage	double	in	voltage for offset DAC	-2.5 to +2.5V

2.35 RI3165 14102_SetDCOffsetLimitVoltages(ci , 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:

- Select Channel to calibrate (SelectChannel(..))
- Be sure channel is not running (SetLockMode (0))
- Bypass filter (SetFilter(1))
- Connect card single ended (ConnectCard(1))
- Set attenuator at 0dB (SetAttenuator (0.0))
- Program DC offset DAC at maximum (SetDCOffsetDacCode(ci, 0xFFFF))
- Measure voltage at output with accurate voltage meter
- Program DC offset DAC at minimum (SetDCOffsetDacCode(ci, 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
ci	unsigned long	in	card identifier	0 to 2^32
posvolt	double	in	measured positive voltage	> 2.5V
negvolt	double	in	measured negative voltage	< -2.5V

2.36 RI3165 14102_SetFilter(ci , filter)

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

Parameters:

Name	Type	Direction	Description	Value
ci	unsigned long	in	card identifier	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

2.37 RI3165 14102_SetLockMode(ci , 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
ci	unsigned long	in	card identifier	0 to 2^32
lock	unsigned long	in	lock/unlock active channel	1 = lock 0 = unlock

2.38 RI3165 14102_SetOffsetCalDacCode(ci, code)

Description:
This function programs the (general) offset calibration dac.

Parameters:

Name	Type	Direction	Description	Value
ci	unsigned long	in	card identifier	0 to 2^32
code	unsigned long	in	code	0 to 2^12

2.39 RI3165 14102_SetOutputOffsetCalDacCode(ci, dac,code)

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

Parameters:

Name	Type	Direction	Description	Value
ci	unsigned long	in	card identifier	0 to 2^32
dac	unsigned long	in	calibration dac	1 to 8
code	unsigned long	in	code	0 to 2^10

2.40 RI3165 14102_SetRangeDacCode(ci , 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 SetAttenuator().

Parameters:

Name	Type	Direction	Description	Value
ci	unsigned long	in	card identifier	0 to 2^32
code	unsigned long	in	code for DC range DAC	0 to 2^16

2.41 RI3165 14102_SetRangeLimitVoltages(ci , 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 (SelectChannel(..))
- Be sure channel is not running (SetLockMode (0))
- Bypass filter (SetFilter(1))
- Set DC offset at 0 V (SetDCOffsetVoltage (0.0))
- Connect card single ended (ConnectCard(1))
- Set DAC code at maximum (SetDacCode(0x3FFF))
- Set attenuator at 0dB (SetAttenuator (0.0))
- Program range DAC at maximum (SetRangeDacCode(ci, 0xFFFF))
- Measure output voltage with accurate voltage meter
- Program range DAC at minimum (SetRangeDacCode(ci, 0x0))
- Measure output voltage with accurate voltage meter
- Call this routine with measured voltages

Parameters:

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

2.42 RI3165 14102_SetSoftwareTriggerStatus(ci , 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 SetTriggerMode() to enable the software trigger mode.

Parameters:

Name	Type	Direction	Description	Value
ci	unsigned long	in	card identifier	0 to 2^32
triggerstatus	unsigned long	in	start or run	0 inactive trigger 1 trigger channel A 2 trigger channel B 3 trigger both channels

2.43 RI3165 14102_SetStartAddress(ci , 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 SetStartAddress to jump to this memory location.

Parameters:

Name	Type	Direction	Description	Value
ci	unsigned long	in	card identifier	0 to 2^32
startaddress	unsigned long	in	desired start address	0 to 2^19

2.44 RI3165 14102_SetStopAddress(ci , 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 SetStartAddress()). The number of samples of the signal is determined by the start address and stop address.

Parameters:

Name	Type	Direction	Description	Value
ci	unsigned long	in	card identifier	0 to 2^32
stopaddress	unsigned long	in	desired stop address	0 to 2^19

2.45 RI3165 14102_SetTriggerMode(ci , triggersource , triggermode)

Description:

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

Parameters:

Name	Type	Direction	Description	Value
ci	unsigned long	in	card identifier	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
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)

2.46 RI3165 14102_SignalAdd(ci, 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 SetAttenuator(). Call SignalClear() to clear all defined signal definitions. Call the function SignalToRam() to fill the stimuli memory with the defined signals (once, after all signals are defined).

Parameters:

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

2.47 RI3165 14102_SignalClear(ci)

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 SignalAdd() to add a new signal definition.

Parameters:

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

2.48 RI3165 14102_SignalToRam(ci)

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

Parameters:

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

2.49 RI3165 14102_StoreCalibrationData(ci)

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
ci	unsigned long	in	card identifier	0 to 2^32

2.50 RI3165 14102_Write(ci , offset , data)

Description:

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

Parameters:

Name	Type	Direction	Description	Value
ci	unsigned long	in	card identifier	0 to 2^32
offset	unsigned long	in	offset address (register)	see hardware manual
data	unsigned long	in	data to write	see hardware manual

2.51 RI3165 14102_WriteEeprom(ci , 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 89 are reserved for calibration data and the module ID.

Parameters:

Name	Type	Direction	Description	Value
ci	unsigned long	in	card identifier	0 to 2^32
eeaddress	unsigned long	in	eeprom address	0 to 255
data	unsigned long	in	byte to be written	0 to 255

2.52 RI3165 14102_WriteId(ci , 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
ci	unsigned long	in	card identifier	0 to 2^32
id	unsigned long	in	card id	0 to 2^32

2.53 RI3165 14102_WriteRam(ci , 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 SetStartAddress(). After this function call the address-counter is incremented with one step.

Parameters:

Name	Type	Direction	Description	Value
ci	unsigned long	in	card identifier	0 to 2^32
data	unsigned long	in	data to write	0 to 2^14

2.54 RI3165 14102_WriteRamBuffer(ci , 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 SetStartAddress(). After this function call the address-counter is incremented with “length” steps.

Parameters:

Name	**Type**	**Direction**	**Description**	**Value**
ci	unsigned long	in	card identifier	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
3165_SUCCESS	0x0	No error(s)

General error codes:

Constant name	Value	Description
3165_ERROR_INVALID_CHANNEL	0xBFFE0004	Invalid channel
3165_ERROR_INVALID_PARAMETER	0xBFFE0005	Invalid parameter
3165_ERROR_MEMORY	0xBFFE0006	Could not allocate memory
3165_ERROR_NO_SIGNALDEF	0xBFFE0007	No signal defined
3165_ERROR_EEPROMCHECK	0xBFFE0008	Eeprom verify error

Kernel Mode Driver (non-visa) Errors Codes

Constant name	Value	Description
3165_ERROR_INV_OBJECT	0xBFFF000E	invalid card reference
3165_ERROR_ALLOC	0xBFFF003C	Insufficient system resources
3165_ERROR_INV_RSRC_NAME	0xBFFF0012	invalid resource name
3165_ERROR_OPEN_FAILURE	0xBFFE0001	Could not open card
3165_ERROR_READ_FAILURE	0xBFFE0002	Error during reading
3165_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.

4 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:

- Open(0, &ci) or OpenCard(bus, device, &ci): Get a card reference number (ci, card identifier). & = reference to the variable;
- SetActiveChannel(ci, 1)): Select channel 1 to be updated;
- SetLockMode(ci, 0): Unlock the memory of channel 1 to make updating of stimuli ram possible;
- SignalClear(ci): Clear all previous declared signals;
- SignalAdd(ci, 0, 2, 11, 0, 0): Define a sine: 2V amplitude (4Vpp), 11 periods, 0 degrees phase.
- SetStartAddress(ci, 0): Set start address to 0;
- SetStopAddress(ci, 1023): Set stop address at 1023. StopAddress - StartAddress + 1 = number of samples;
- SignalToRam(ci): Fill stimuli memory with the sine;
- SetClockDivider(ci, 1): Divide clock by 1;
- SetAttenuator(ci, -6): Attenuate signal 6 dB
- SetDCOffsetVoltage(ci, 0): Set the dc offset voltage at 0V;
- SetFilter(ci, 2): Connect 6 MHz filter path;
- SetClockSource(ci, 1): Clock 1 (100 MHz) clock source;
- SetTriggerMode(ci, 8, 0): Software trigger, positive edge;
- ConnectCard(ci, 1): connect the output single ended;
- ResetToStartAddress(ci): Reset address counter to start position (start address);
- SetLockMode(ci, 1): Set channel in lock mode;
- SetSoftwareTriggerStatus(ci, 1): Trigger channel 1;

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 3165 (980899-002) 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

Technologie Park, Friedrich Ebert Strasse, 51429 Bergisch Gladbach, Germany
Tel: +49 (0) 2204 844200; Fax: +49 (0) 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