USER MANUAL

MODEL 9215
DUAL OUTPUT PROGRAMMABLE 0-30 VDC/2A POWER SUPPLY

PUBLICATION NO. 980894

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/

PUBLICATION DATE: June 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.

This warranty does not apply to defects resulting from any modification(s) of any product or part without Racal Instruments express written consent, or misuse of any product or part. The warranty also does not apply to fuses, software, non-rechargeable batteries, damage from battery leakage, or problems arising from normal wear, such as mechanical relay life, or failure to follow instructions.

This warranty is in lieu of all other warranties, expressed or implied, including any implied warranty of merchantability or fitness for a particular use. The remedies provided herein are buyer's sole and exclusive remedies.

For the specific terms of your standard warranty, or optional extended warranty or service agreement, contact your Racal 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)
	+852 2405 5500	(Hong Kong)
Fax:	+1 949 859 7309	(USA)
	+44(0) 1628 662017	(UK)
	+852 2416 4335	(Hong Kong)

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 in the Product Support section 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. RII 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 RII product could create a situation where personal injury or death could occur. Should Buyer purchase RII product for such unintended application, Buyer shall indemnify and hold RII, 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.

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 92718

declare under sole responsibility that the

9215 PXI Programmable DC Dual P/S, P/N 407884

The 9215 conforms to the following Product Specifications:

Safety: EN61010-1:1993+A2:1995

EMC: EN61326:1997+A1:1998

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 (modified by 93/68/EEC).

Irvine, CA, December 9, 2003

Engineering Director for Ken Evenson

This page was left intentionally blank.

Table of Contents

This page was left intentionally blank.

Chapter 1 – General Information

1.1 Overview

This 9215 User Manual provides all the information needed to install, program, and use the Racal Instruments 9215 Dual Output, Programmable, 0-30VDC/2A Power Supply. This manual assumes that the user has a general knowledge of personal computers, Windows 9x/Me/2000/NT operating systems, and an electronics background. Some knowledge of programming and development tools will permit computer program control of the programmable DC power supply.

Model 9215 is a dual output power supply on a 3U-size PXI card format. The instrument is a message-based device and includes software drivers. External power may connect to Model 9215 through a front-panel connector. This approach avoids the need for either special backplane modifications to route DC power to the module (although available as an option) or excessively large backplane power supplies. Also available is a side connector for a 48 VDC supply from an optional, adjacent PXI module.

Model 9215 power supply is packaged in a standard PXI, single-slot, 3U card. It is comprised of a motherboard PCB containing the PXI interface and communications circuitry. The motherboard PCB provides the necessary control circuitry for communications over the PXI backplane, and storage of calibration data and default settings in on-board EEPROM.

1.1.1 OUTPUT STAGES

Designed for applications requiring higher power levels and signal quality, Model 9215 provides output power by two independent, isolated output stages. The outputs are rated at 0-30 VDC, 2 Amps and are limited to 30 Watts per channel. Efficiency is >80%.

Model 9215 is covered and shielded to eliminate EMI/RFI interference and to provide for controlled airflow. Each stage is joined to a cooling plate sized to provide adequate cooling up to 30 W per channel with rated airflow (see PXI specification) even in instances where adjacent PXI card slots are vacant. To provide for maximum flexibility in meeting a broad range of UUT (Unit Under Test) power requirements, output stages may be used in parallel or series under software control. For example, an instrument configured with two 30 VDC output stages may be used as a dual output module at 30 W per output, or may be programmed as a single 0-30 VDC/60 W supply in parallel mode, or as a single 0-60 VDC/30 W supply in series mode. To cover larger power requirements, additional 9215 supplies may also be paralleled with some de-rating in performance specifications.

The Model 9215 supply can be purchased with either an optional AC to 48 VDC desktop style converter or a separate AC to 48 VDC module to allow the units to operate from worldwide AC mains.

1.2 Firmware Capability

1.2.1 SOFTWARE CALIBRATION

Model 9215 contains internal non-volatile EEPROM memory for storage of calibration and configuration data. Calibration is accomplished via backplane communications without the need

for manual adjustments. It is not necessary to remove the instrument from the PXI card cage to perform calibration.

1.3 Graphical User Interface

Software drivers and examples for Visual Basic and Visual C are included on the CD. The included graphical (GUI) provides quick and easy control. No knowledge of specific 9215 instrument drivers or commands is needed when using the (GUI).

1.4 Applicable Specification

Model 9215 is designed to meet the following standards and specifications:

MIL-STD-28800D Type III, Class 3

MIL-STD-45208 Quality Standards

MIL-STD-461C EMI Part 7 (For commercial test equipment)

VDE

UL

CE

CSA

IEC 435 Safety and Isolation

IEC 380 Safety Requirements

IEEE-488.2

PXI Specification Revision 2.0

Chapter 2 – PXI Description

2.1 Overview

The goal of PXI is to define a technically sound modular instrument standard based on the PCI standard, which is open to all manufacturers and is compatible with present industry standards.

The PXI specification details the technical requirements of PXI compatible components such as mainframes, backplanes, and logic power supplies. The specification also provides for interconnecting and operating different manufacturers' products with the same card chassis. Model 9215 has been designed to be compliant with the current revision of the PXI specification. You may download the latest revision of the PXI specification from http://www.pxisa.org.

2.2 PXI Capabilities

NOTE: Model 9215 does not implement the following PXI capabilities, as no J2 connector is installed:

Local Bus - This daisy-chained bus connects each peripheral slot with the adjacent peripheral slots to the left and right. Thus, the right local bus of a given peripheral slot connects to the left local bus of the adjacent slot, and so on. Each local bus is 13 lines wide and may be used to pass analog signals between modules or to provide a high-speed side-band digital communication path that does not affect the PXI bandwidth.

Star Trigger - The local bus lines for the leftmost peripheral slot of a PXI backplane are used for the star trigger. The star trigger bus implements a dedicated trigger line between the first peripheral slot (adjacent to the system slot) and the other peripheral slots.

Trigger Bus – Up to eight triggers may be passed from one module to another, allowing precisely timed responses to asynchronous external events that are being monitored or controlled.

System Reference Clock - The PXI 10 MHz system clock (PXI_CLK10) may be used for synchronization of PXI instruments.

This page left intentionally blank.

Chapter 3 –Unpacking, Inspection, System Requirements

3.1 Unpacking and Inspection

Unpacking and Inspection

Caution: Static-sensitive instrument! Ground yourself to discharge static.

1. Remove the 9215 from the static bag, handling only the metal portions.
2. Check the contents of the shipping carton to verify that the items match the packing list.
3. Inspect the 9215 for possible damage. If there is any sign of damage, return the 9215 immediately. Please refer to the warranty information at the beginning of this manual.

Discharge Static Electricity

To reduce the risk of damaging the 9215, observe the following precautions:

- Leave the 9215 in the anti-static bag until you are ready to install it. The anti-static bag protects the unit from harmful static electricity.
- Save the anti-static bag in case the 9215 is removed in the future.
- Carefully unpack and install the 9215. Do not drop the 9215 or handle it roughly.

Packing List

Model 9215 is shipped from the factory with the following:

- 9215 User Manual and driver (on CD-ROM)
- 9215 Dual-Output Programmable power supply, 0 to 30 VDC/2 A
- Optional: Two 48 VDC desktop power supplies (for 115/220 VAC operation). AC cords included. Racal Instruments part number 407884-901.
- Optional: Dual Isolated 48 VDC output module, PXI 3U x 4HP (1-slot) and interface module. Racal Instruments part number 407884-902.

Model 9215 was carefully tested and inspected for mechanical and electrical defects prior to shipment. Inspect the unit for any visible damage that may have occurred in transit. If the shipping container is damaged or other damage is apparent, it is recommended to a) report damage immediately to the carrier and to the factory, b) retain the shipping container, and c) photograph any damage to the container or product.

3.2 Environmental and System Requirements

In order for the 9215 to meet its specifications, the operating environment must be within the following limits:

Temperature	0 to +40 degrees C
Relative Humidity	20 to 95% (non-condensing)

The non-operating temperature specification is from –40 degrees C to +85 degrees C.

Model 9215 is shipped from the factory pre-wired for the following front panel input power:

Two isolated +48 VDC/1.25 A/60 W inputs are needed for dual isolated output.

Model 9215 requires the following input power from the PXI backplane:

+5 VDC @ 1.0 A

+12 VDC @ 0.2 A

Chapter 4 – Installing the Hardware and Software

4.1 Installing the 9215

Caution: Turn the chassis power OFF before attempting installation. Do not attempt to insert or remove the unit with chassis power on.

1. Turn off the power to the PC.
2. Unplug the PXI chassis from the AC power outlet to avoid possible electrical shock.
3. Locate an available PXI slot.
4. Remove the slot cover plate.

WARNING: Inserting the 9215 in a PXI slot with the PXI chassis power on may damage the 9215, the PXI chassis, or both.

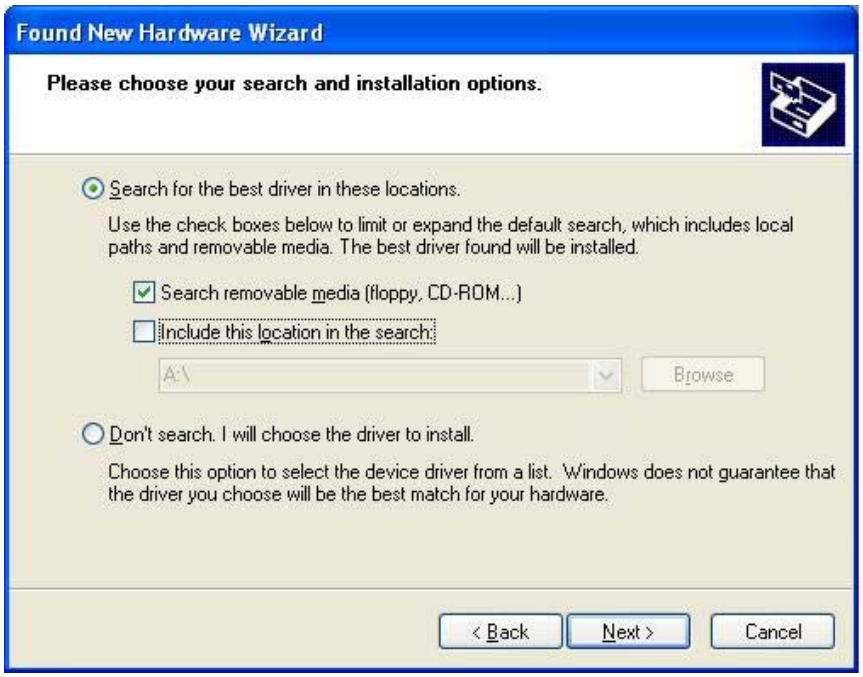
5. Insert the 9215 board into the PXI slot. It should insert with minimal amount of force. Press in firmly, yet gently, on the top edge of the front bracket while lifting the bottom insertion lever, applying equal force.
6. Secure the 9215 board by tightening the captive screws at the top and bottom of the front panel.
7. Connect the power cord to the back of the PXI chassis and plug it back into the AC wall outlet.

4.2 Installing the Software Drivers for Windows 9x/Me/2000/XP

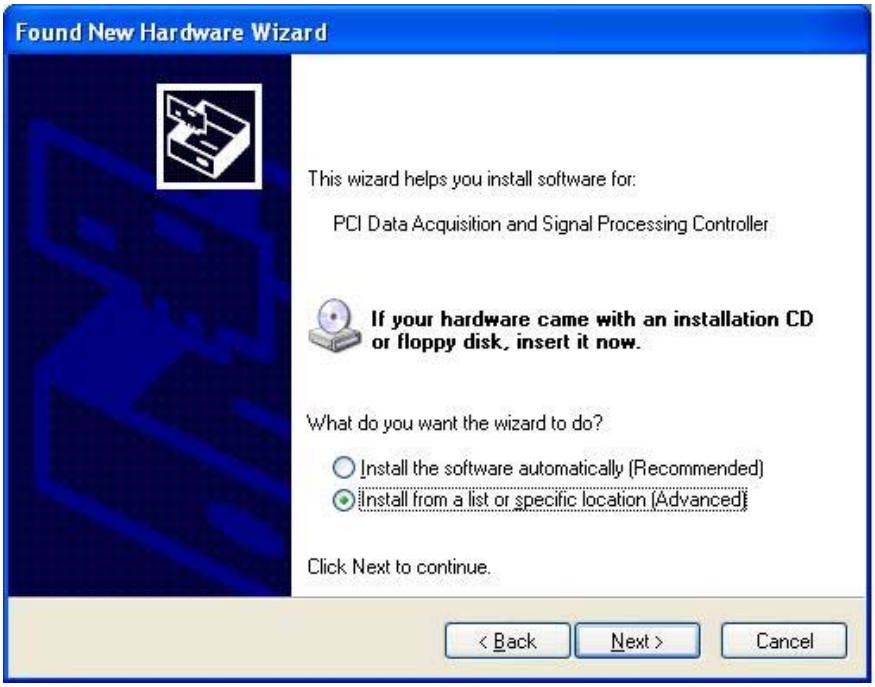
When you install the 9215 for the first time, Windows 9x/Me/2000/XP will automatically detect that you have a new device, and will request the software drivers. Please follow the step-by-step instructions below to properly install the required software drivers.

1. Below is a picture of the dialog box that will notify you that Windows has detected your new hardware. Select the last choice, “No, not this time” and click “Next.”

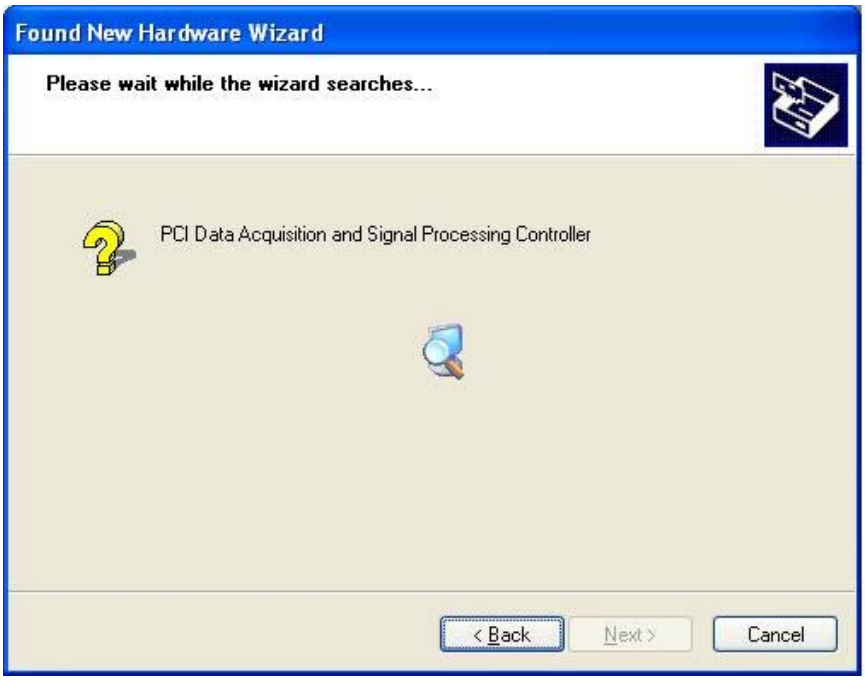
2. Windows will now automatically run the “Add New Hardware Wizard.” Select the first option “Search for the best driver…” and click the “Next” button.



3. Windows will now ask where you would like to search for the driver. Make sure you have the installation CD in the CD-ROM drive, then, make the selection shown below. Then, click on the “Next” button.



4. Please be patient while Windows locates your new driver on the CD. During this process the display will appear as follows:



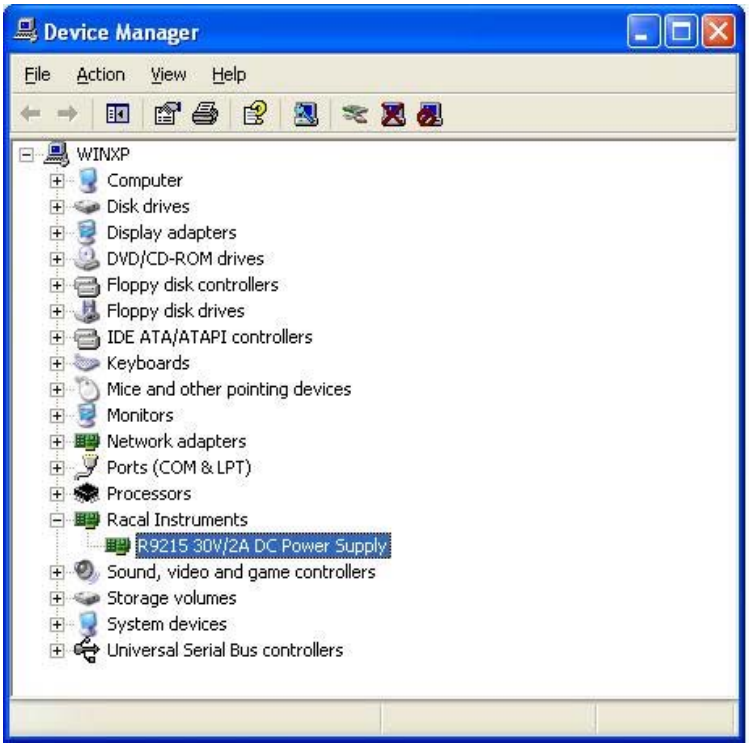
5. Click on the “Next” button once Windows finds the correct driver information (r9215pxi.inf). The software will then be installed.

6. Once driver installation is complete, click the “Finish” button.

7. The Hardware Wizard will now detect the Windriver Virtual device. Click “Next” once it finds the driver.

8. Click “Finish” to complete the installation of the WinDriver Virtual device.

9. To check your installation, run "Device Manager". In Windows 2000, right click on “My Computer,” select Properties, then the Hardware tab, then select the Device Manager button. You should see the following screen once you open the Racal Instruments branch:



10. Double-click the Racal Instruments icon. You should see the following screen.

4.3 Installing the Application Software

To install the application software, insert the supplied installation CD and run the setup program. Assuming that your CD drive is device "D:" the setup program is “D:\ setup_r9215.exe” for Windows 9x/Me/NT/2K/XP. Follow the on-screen instructions. The installation program will create a shortcut on the Windows desktop and a program group in the Start menu.

This page was left intentionally blank

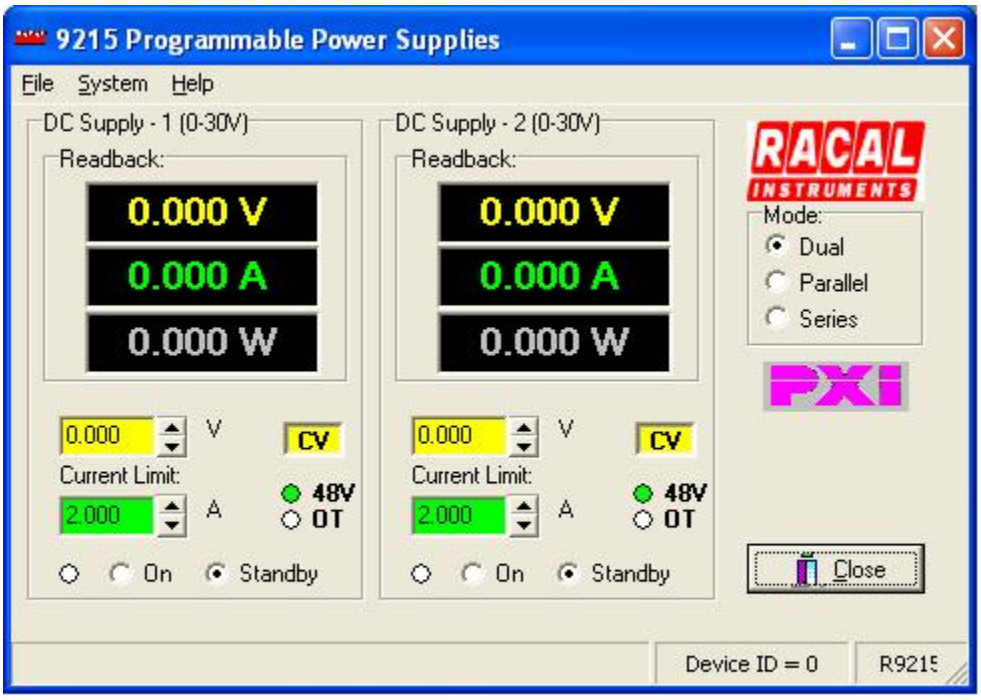
Chapter 5 – Operation and Programming

5.1 Overview

Model 9215 is a PXI/Compact PCI, dual output, programmable, 0-30 VDC/2 A power supply. Up to six 9215 instruments may be deployed in the same chassis, limited only by the environmental cooling, available slots, and available system resources. All installed units are controlled through a single Windows dynamic link library (r9215ps.dll). The DLL provides the interface between the hardware and the end-user's application program.

As an alternative to writing custom code, a Windows Graphical User Interface (GUI) is provided, which may be used for interactive control of the 9215. This chapter describes the functions contained in r9215ps.dll and the function declarations. Examples are provided for calling the DLL from VB and VC.

5.2 9215 Graphical User Interface



The main GUI screen consists of two identical frames, one for each power supply output. If the supplies are operated in either Parallel or Series mode, only the left-hand frame (DC Supply - 1) is visible. In these modes, both supplies are controlled by the DC Supply - 1 frame controls. The mode can be selected on the right hand side of the GUI window.

The top three fields with black backgrounds read out the voltage, current and power measurements for each supply as long as the output relay for the supply is ON. To change

settings, use the Voltage (Yellow) or Current (Green) spin controls below the measurement readout. You may also type in a value directly and press the “Enter” key, or move to a different control to change a setting.

The status of each supply is shown directly to the right of the Voltage and Current controls. If the 48 V supply feeding the PWM circuit is missing, the 48 V indicator lights up and an error message is visible in the status bar at the bottom. A Front Panel LED also shows an error if 48 V is not supplied. If an over-temperature condition occurs, the supply shuts down and the OT indicator is on.

To open or close the output relay, use the On or Standby controls at the bottom of each frame.

To exit the program, click on the “Close” button or select File/Exit from the menu bar.

5.3 DLL Configuration File

Each time the DLL is initiated, it looks for the configuration settings file located in the 9215 application directory. This file is called r9215ps.ini and contains the following information and formatting:

```
[board_settings]
base0 = 0
Base1 = 8
Base2 = 16
Base3 = 24
```

Values shown here are defaults and are used if one or more entries or the entire ini file is omitted. Base numbers correspond to board IDs, thus “base0=0” indicates that the board with ID 0 will be the first board found. Offset values may be assigned to the installed boards as desired as long as there are no duplicates and the values are 8 apart. For most situations, there is no need to set the base values, and the defaults may be used.

A unique board ID number ranging from 0 to *n* identifies each board. Every function or procedure call in the DLL has a board ID parameter. It is always a 32-bit integer type (**long** in Visual BASIC).

The Windows 32-bit r9215ps.dll file can be used for applications that are developed for Windows 9x/Me/NT/2000/XP.

To use the DLL driver from within an application, the DLL must reside in one of the following directories:

- Application directory
- Windows directory (e.g., \Windows or \WinNT)
- Windows system directory (e.g., \Windows\System or WinNT\System32)
- One of the directories specified in the PATH statement

The r9215ps.dll file may be distributed with the 9215 power supply and any associated applications.

5.4 Constant Declarations

The following constants may be useful when developing application programs for the 9215.

```
define EPROM_OK = $AA55;                // EPROM good value
//=============================================================
 // EPROM Address Locations stored scaled to 16 bit integers
//=============================================================
define EE_ZERO = $0;
define EE_MODEL = 1;                    // Board model number 9 character length
define EE_SNUMBER = $0B;                // Serial number
define EE_REVISION = $0C;               // Hardware revision
define EE_MFG_MONTH = $0D;              // Manufacturing month
define EE_MFG_YEAR = $0E;               // Manufacturing year
define EE_CAL_DAY = $1D;
define EE_CAL_MONTH = $1E;
define EE_CAL_YEAR = $1F;
define EE_CHECK = $0F;
//=============================================================
// Control State Constants
//=============================================================
define RELAY_CLOSED = 0;
define RELAY_OPEN = 1;
//=============================================================
// Offsets shown for PSU1.
// PSU 2 offset = PSU1 offset + $10     Scaling                Resolution
//=============================================================
define EE_CAL_VGAIN = $10;              // Scaled x 10000
define EE_CAL_VOFFS = $11;              // Scaled x 1000          1 mV
define EE_CAL_IGAIN = $12               // Scaled x 10000
define EE_CAL_IOFFS = $13               // Scaled x 1000          1 mA
define EE_CAL_VMGAIN = $14              // Scaled x 10000
define EE_CAL_VMOFFS = $15              // Scaled x 1000          1 mV
define EE_CAL_IMGAIN = $16              // Scaled x 10000
define EE_CAL_IMOFFS = $17;             // Scaled x 1000          1 mA
define EE_LIM_VOLT = $18;               // Scaled x 100           10 mV
define EE_LIM_CURR = $19;               // Scaled x 100           10 mA
define EE_LIM_POWER = $1A;              // Scaled x 10            100 mW
define EE_LIM_VGAIN = $1B;              // Scaled x 1000000
define EE_LIM_IGAIN = $1C;              // Scaled x 1000000
 //=============================================================
 // Cal Parameter Constants
 //=============================================================
define CAL_VGAIN = 1;
define CAL_VOFFS = 2;
define CAL_IGAIN = 3;
define CAL_IOFFS = 4;
define CAL_VMGAIN = 5;
define CAL_VMOFFS = 6;
define CAL_IMGAIN = 7;
define CAL_IMOFFS = 8;
```

Programming with C/C++ Tools

The following steps are required to use the 9215 driver with C/C++ development tools:

- Include the r9215.h header file in the C/C++ source file that uses the 9215 function. This header file is used for all driver formats. The file contains function prototypes and constant declarations to be used by the compiler for the application.
- For Windows applications, make sure the DLL is installed in the proper directory (see previous sections that describe how to use the DLL).
- Add the required .LIB file to the project. This can be the import library R921532.LIB for 32-bit Microsoft applications, or R9215.LIB for 32-bit Borland C++ applications. Windows based applications that explicitly load the DLL by calling the Windows **LoadLibrary** API need not include the .LIB file in the project.
- Add code to call the 9215 as required by the application.
- Build the project.
- Run, test, and debug the application.

Programming with Visual Basic

The r921532.bas file contains function declarations for the r9215ps.dll driver. The BAS file must be loaded using **Load File** from the Visual Basic File menu before the functions can be used.

5.5 DLL Functions and Procedures

The following functions and procedures are provided by the DLL:

General System functions

r9215_Close_all
r9215_DLLVersion
r9215_WinDriverVersion
r9215_BoardPresent
r9215_Model_Number
r9215_xWaitFor
r9215_Model_Number
r9215_Reset

Power Supply Control functions

r9215_FlashLed
r9215_OutputRelay
r9215_PowerSate
r9215_set_voltage
r9215_set_current

Measurement functions

r9215_Measure_Voltage
r9215_Measure_Current

Status functions

r9215_Status

EPROM functions

r9215_EEProm_Read
r9215_GetConfigValue
r9215_GetCalibration

Details on parameters passed and results returned are provided in the following paragraphs.

This page left intentionally blank.

Chapter 6 –Functions Reference

6.1 General System Functions

This chapter discusses all of the 9215 driver functions. For each function, a syntax example is followed by a short description of the function, parameters, and type.

r9215_Close_all

procedure r9215_Close_all;
This function closes all device handles still open to any 9215 board. ***This function must be called upon exiting your application program***. If you fail to do so, the board cannot be accessed again, as the drivers lock access to the board while board handles are open.

If you are using Visual BASIC, place this call in the Form, QueryUnload event handler.
This function has no parameters and returns a void. (no function result.)

r9215_DLLVersion

function r9215_DLLVersion(iBrdID:integer; Var strRevision:PChar):integer;
This function returns a null-terminated string containing the version of the R9215ps.dll. This string is passed by reference and is the second parameter in the function list. The board ID must be passed by value as the first parameter. The function result returns a 32-bit integer (int32), which contains the DLL version multiplied by 100. Thus, a returned value of 100 indicates version 1.00.

Future versions of the DLL may contain additional functions. By having the ability to determine the DLL version, your application program can be designed to handle new features when they become available.

r9215_WinDriverVersion

function r9215_WinDriverVersion(iBrdID:integer; Var strRevision:PChar):integer;
This function returns a null-terminated string containing the version of the PCI card device driver. This version string is passed by reference and is the second parameter in the function list. The board ID must be passed by value as the first parameter. The function result returns a 32-bit integer (int32), which contains the WinDriver version multiplied by 100. Thus, version 1.00 would be returned as 100.

r9215_BoardPresent

function r9215_BoardPresent(iBrdID:integer):integer;
This function returns 0 if the board with the board ID passed as the parameter is not present. If the board is present, -1 is returned instead. The function type is a 32-bit integer (int32).

This function must be used any time a new device is being accessed to make sure a handle to the board is available. If a board is not present, and access to that board is attempted using any of the other DLL functions (except the r9215_DLLVersion call), a run-time error will occur.

r9215_Model_Number

procedure r9215_Model_Number(iBrdID:integer):integer;

This function returns the model number for the board ID specified as the first and only parameter. The board model for a 9215 is 3200 hex or 12800 decimal.

r9215_xWaitFor

procedure r9215_xWaitFor(period:integer);
This procedure generates a time delay specified in milliseconds. The resolution and accuracy of this function is about 1 millisecond for delays under 2 seconds. Delays above 2 seconds (2000 msec) will result in relinquishing control to other windows applications, which may affect accuracy. The delay time is specified as a 32-bit integer (int32). The function result for this call is void.

r9215_Reset

procedure r9215_Reset(iBrdID:integer);
This procedure generates a time delay specified in milliseconds. The resolution and accuracy of this function is about 1 millisecond for times under 2 seconds. Times above 2 seconds (2000 msec) will result in relinquishing control to other windows applications, which may affect accuracy. The time delay period is specified as a 32-bit integer (int32). The function result for this call is void.

6.2 9215 Control Functions

r9215_FlashLed

procedure r9215_FlashLed(iBrdID:integer; iIndex:integer);
This function causes one of the LED's on the PSU front panel to flash momentarily. The iIndex parameter (Int32) determines which of the two output module LEDs will flash.
This function has no parameters and returns a void. (no function result.)

r9215_OutputRelay

procedure r9215_OutputRelay(iBrdID:integer; iIndex:integer; iValue:integer);
This function controls the state of the output relay and PWM power for the selected output module (iIndex). The iValue passed determines whether the output relay is opened or closed. The state of the output relay and 9215 power is returned as part of the Status query (see r9215_Status, paragraph 6.4).
Note: To prevent damage to the output relay, the 9215 power is disengaged before the output relay is opened and re-engaged before the output relay is closed. This process requires approximately 250 msec to complete.
Valid values for iValue are:

RELAY_CLOSED = 0;
RELAY_OPEN = 1;

r9215_ PowerState

procedure r9215_PowerState(iBrdID:integer; iIndex:integer; iValue:integer);
This function controls the state of the PWM power for the selected PSU output (iIndex). The iValue passed determines whether the PWM power relay (48 Volt Relay State) is applied or not. The state of the PWM power is returned as part of the Status query (48 Volt Relay State), see paragraph 0.
Note: This process requires approximately 50 msec to complete to allow for relay de-bounce.
Valid values for iValue are:

RELAY_CLOSED = 0;
RELAY_OPEN = 1;

r9215_Set_voltage

function r9215_Set_Voltage(iBrdID:integer; iIndex:integer; sVoltage:double):integer;
This function programs the output voltage in VDC for the selected output channel (1 or 2). The desired output voltage is passed as a double precision floating point number.
If a problem occurs during this query, the function result may be one of the following error values:

Error Codes:
-200 // V set below lower limit
-201 // V set above upper limit
-202 // V set requested exceeds total power available. Note that V * I must be < Max. Power.
-203 // V set exceeds DAC upper limit

r9215_Get_voltage

function r9215_Get_Voltage(iBrdID:integer; iIndex:integer; var sVoltage:double):integer;
This function returns the programmed voltage value for the requested output module number.
The programmed value is passed by reference.
If a problem occurs during this query, the function result may be one of the following error values:

Error Codes:
-220 // // Output module number out of range
-230 // Board not present

r9215_Set_current

function r9215_Set_Voltage(iBrdID:integer; iIndex:integer; sVoltage:double):integer;
This function programs the output current limit in ADC for the selected output channel (1 or 2).
The desired current limit is passed as a double precision floating point number.
If a problem occurs during this query, the function result may be one of the following error values:

Error Codes:
-210 // I set below lower limit
-211 // I set above upper limit
-202 // V set requested exceeds total power available. Note that V * I must be < Max. Power
-213 // I set exceeds DAC upper limit

r9215_Get_current

function r9215_Get_Current(iBrdID:integer; iIndex:integer; var sCurrent:double):integer;
This function returns the programmed current limit value for the requested output module number.
The programmed value is passed by reference.
If a problem occurs during this query, the function result may be one of the following error values:

Error Codes:
-220 // // Output module number out of range
-230 // Board not present

6.3 Measurement Functions

r9215_Measure_Voltage

function r9215_Measure_Voltage(iBrdID:integer; iIndex:integer; iAverage:integer; var sVoltage:double):integer;
This function call measures the output voltage for the select power supply. If no measurement data is available, the function result will be non-zero. The iAverage parameter can range from 1 to 255 and sets the number of measurements to be averaged before returning the average measurement result.
If a problem occurs during this query, the function result may be one of the following error values:

Error Codes:
-240 // Measurement Average requested less than 1

r9215_Measure_Current

function r9215_Measure_Current(iBrdID:integer; iIndex:integer; iAverage:integer; var sCurrent:double):integer;
This function call measures the output current for the select power supply. If no measurement data is available, the function result will be non-zero. The iAverage parameter can range from 1 to 255 and sets the number of measurements to be averaged before returning the average measurement result.
If a problem occurs during this query, the function result may be one of the following error values:

Error Codes:
-240 // Measurement Average requested less than 1

6.4 Status Functions

r9215_Status

function r9215_Status(iBrdID:integer; iIndex:integer):integer;
This function call returns the status for the requested output module number, iIndex (1 or 2). The status bits can be decoded using the table below:

Bit number	Weight	Description	
b31-b5		Reserved	
b4	16	48 Volt Relay State:	0 = Closed, 1 = open
b3	8	Output Relay State:	0 = Closed, 1 = open
b2	4	Fault	0 = Good, 1 = Fault
b1	2	48 Volt	0 = 48 V OK, 1 = 48 V Bad
b0	1	Mode	0 = CC mode, 1 = CV mode

6.5 Calibration and Configuration Functions (EPROM access)

Model 9215 contains non-volatile memory to store configuration and calibration values. The following two functions may be used to query EPROM configuration. The third function is used to query calibration data.

r9215_EEProm_Read

function r9215_EEProm_Read(iBrdID:integer; iAddress:integer; var iData:integer):integer;
This function may be used to read from any EPROM location. The EPROM address offsets for relevant data fields are shown in the Constant declaration section. Note that while the data type of the variable returned (iData) is a 32-bit integer, each EPROM location only stores 16 bits so the results should be interpreted as a signed short integer or character depending on the location read. For configuration value (limits), use the te_GetConfigValue function instead as it returns a floating-point number. For calibration coefficients, use the te_GetCalibration function instead as it returns a floating-point number.
If a problem occurs during this function call, the function result may be one of the following error values:

Error Codes:
-250 // EPROM read error

r9215_GetConfigValue

function r9215_GetConfigValue(iBrdID, iIndex, iParam:integer; var dbValue:double):integer;
This function may query a specific configuration (hardware limit) value for the specified board and output module number (iIndex). The configuration value returned depends on the iParam value passed. The following values are valid for iParam:

Parameter value	Limit Value Returned
1	Max Voltage
2	Max Current
3	Max. Power

If a problem occurs during this function call, the function result may be one of the following error values:

Error Codes:
-220 // Output module number out of range
-222 // Parameter value out of range

r9215_GetCalibration

r9215_GetCalibration(iBrdID, iIndex, iParam:integer; var dbCalData:double):integer;
This function returns the selected Cal coefficient for the specified board and output module (iIndex).in the dbCalData value. The cal coefficient value returned is determined by the value of the iParam parameter per the table shown below.

Parameter value	Calibration Coefficient Value Returned	
1	CAL_VGAIN.	Voltage output gain
2	CAL_VOFFS	Voltage output offset
3	CAL_IGAIN	Current output gain
4	CAL_IOFFS	Current output offset
5	CAL_VMGAIN	Voltage measurement gain
6	CAL_VMOFFS	Voltage measurement offset
7	CAL_IMGAIN	Current measurement gain
8	CAL_IMOFFS	Current measurement offset

If a problem occurs during this function call, the function result may be one of the following error values:

Error Codes:
-220 // Output module number out of range
-222 // Cal parameter requested out of range

6.6 DLL Error Codes

All DLL function calls return 32-bit integer error results. If no error occurs during the call, the error result will be zero. Any negative result indicates an error. Note that procedure calls have a VOID result and return no error codes.

Below is a list of all error codes by category and their descriptions. The application programmer is responsible for error checking.

Error Code	Description
Device Driver Errors	
-100	PXI board handler error
Range Errors - Voltage	
-200	V set below lower limit. Requested voltage setting is less than zero.
-201	V set above upper limit. Requested voltage setting is exceeds maximum voltage of output module.
-202	V set exceeds power limit. Requested voltage setting would result in power above limit. Try lowering current limit first.
-203	V DAC value out of range. Programmed value exceeds hardware limit.
Range Errors - Current	
-210	I set below lower limit. Requested current setting is less than zero.
-211	I set above upper limit. Requested current setting is exceeds maximum current limit.
-212	I set exceeds power limit. Requested current setting would result in power above the limit. Try lowering
-213	I DAC value out of range. Programmed value exceeds hardware limit.
Range Errors – Other	
-220	Output module number out of range. Index range is 1 to 2.
-221	Cal parameter requested out of range. Unknown calibration parameter index specified.
-222	Limit parameter requested out of range. Unknown limit parameter index specified.

Error Code	Description
Misc. Errors	
-230	Board not present. No board found at board index location.
-240	Measurement average requested less than 1. Minimum value allowed is 1.
EPROM Errors	
-250	EPROM read error. Could not read from EPROM at requested address.
-251	EPROM write error. EPROM is write-protected.

6.7 Visual Basic Applications

The r9215ps.dll may be called from a Visual Basic 6 application program. The required function declarations are provided in "DLL Declaration Module", below. A sample VB6 application GUI (r9215GuiSample.vbp) is included on the CD (D:\SampleCode\ r9215VB6_Sample.zip).

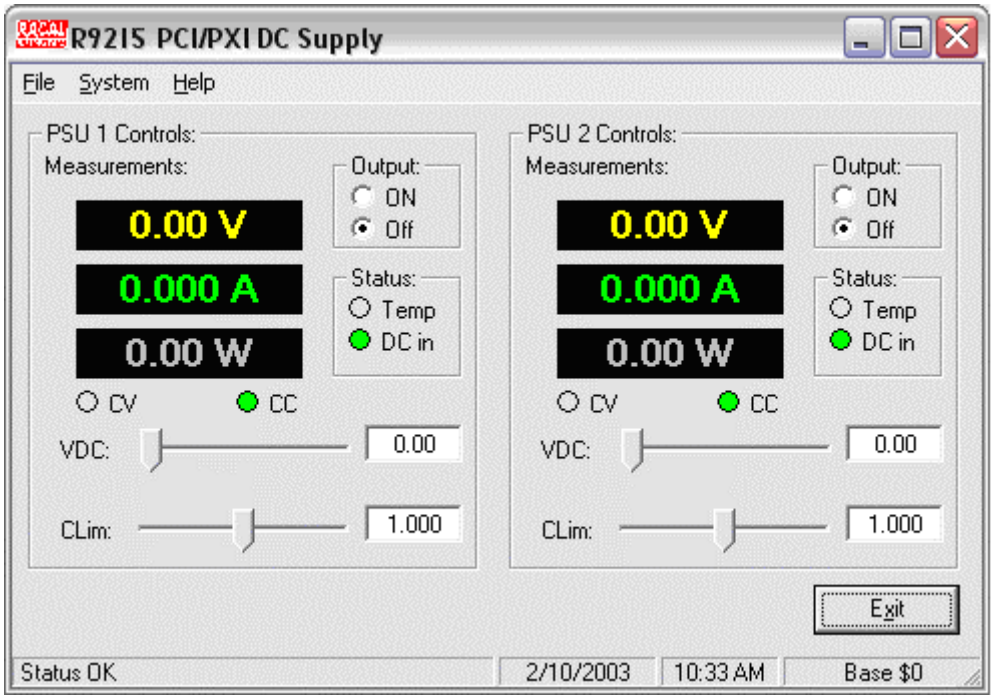


Figure 1: R9215 VB6 GUI Sample

Programming tips

When developing a VB Application, make sure to check for the presence of each of up to four boards using the r9215_BoardPresent function call.

```
'Initialize all boards
Dim iBrdID As Long
  For iBrdID = 0 To MAX_BOARDS
    If r9215_BoardPresent(iBrdID) = BOARD_PRESENT Then
      R9215_Initialize_DDS iBrdID
    End If
  Next iBrdID
```

Make sure to put the r9215_Close_all procedure call in the QueryUnload event handler of your main application form.

```
Private Sub Form_QueryUnload(Cancel As Integer, UnloadMode As Integer)
'=============================================================================
'Close Device handles to 9215 Board(s)
'=============================================================================
  r9215_Close_all
End Sub
```

DLL Declaration Module

```
Attribute VB_Name = "r9215_DLL_Declarations"
'================================================================================
' Module:       9215 DLL declarations
' Copyright:    © 2004 Racal Instruments, Inc.
' Date:         06/21/2004
' Revised:      06/21/2004
'================================================================================
'String Passing:
'===============
'String arguments should always be passed using the ByVal keyword to ensure
'they are passed as null terminated strings. Otherwise a string descriptor
'is passed which the r9215ps.dll does not know how to handle as it does
'not know VB.
'Integers
'========
'Pass integers by value using the ByVal keyword
'Arrays
'======
'Arrays cannot be passed using the ByVal keyword and should always be passed by
reference.
'================================================================================
'r9215_Close_all
Declare Sub r9215_Close_all Lib "r9215ps.dll" ()
'r9215_DLLVersion
Declare Function r9215_DLLVersion Lib " r9215ps.dll" (ByVal BrdID As Long, ByRef strRevision As
String) As Long
'r9215_WinDriverVersion
Declare Function r9215_WinDriverVersion Lib " r9215ps.dll" (ByVal BrdID As Long, ByRef
strRevision As String) As Long
'r9215_BoardPresent
Declare Function r9215_BoardPresent Lib " r9215ps.dll" (ByVal iBrdID As Long) As Long
'r9215_xWaitFor
Declare Sub r9215_xWaitFor Lib " r9215ps.dll" (ByVal period As Long)
'r9215_Model_Number
Declare Function r9215_Model_Number Lib " r9215ps.dll" (ByVal iBrdID As Long) As Long
'r9215_Reset
Declare Sub r9215_Reset Lib " r9215ps.dll" (ByVal iBrdID As Long)
'PSU Programming Functions
'r9215_Output_Power
Declare Sub r9215_OutputRelay Lib " r9215ps.dll" (ByVal iBrdID As Long, ByVal iIndex As Long,
ByVal iValue As Long)

Declare Sub r9215_FlashLed Lib " r9215ps.dll" (ByVal iBrdID As Long, ByVal iIndex As
Long)
Declare Function r9215_Set_Voltage Lib " r9215ps.dll" (ByVal iBrdID As Long, ByVal iIndex
As Long, _
                                                       ByVal sVoltage As Double) As Long
Declare Function r9215_Set_Current Lib " r9215ps.dll" (ByVal iBrdID As Long, ByVal iIndex
As Long, _
                                                       ByVal sCurrent As Double) As Long
Declare Function r9215_Get_Voltage Lib " r9215ps.dll" (ByVal iBrdID As Long, ByVal iIndex
As Long, _
                                                       ByRef sVoltage As Double) As Long
Declare Function r9215_Get_Current Lib " r9215ps.dll" (ByVal iBrdID As Long, ByVal iIndex
As Long, _
                                                       ByRef sCurrent As Double) As Long
Declare Function r9215_Measure_Voltage Lib " r9215ps.dll" (ByVal iBrdID As Long, ByVal
iIndex As Long, _
                                                           ByVal iAverage As Long, ByRef
sVoltage As Double) As Long
Declare Function r9215_Measure_Current Lib " r9215ps.dll" (ByVal iBrdID As Long, ByVal
iIndex As Long, _
                                                           ByVal iAverage As Long, ByRef
sCurrent As Double) As Long
Declare Function r9215_Status Lib " r9215ps.dll" (ByVal iBrdID As Long, ByVal iIndex As
Long) As Long
'PSU Eprom Storage Routines
```

```
Declare Function r9215_EEProm_Read Lib " r9215ps.dll" (ByVal iBrdID As Long, ByVal
iAddress As Long, ByRef iData As Long) As Long
Declare Function r9215_GetConfigValue Lib " r9215ps.dll" (ByVal iBrdID As Long, ByVal
iIndex As Long, _
                                                          ByVal iParam As Long, ByRef
dbValue As Double) As Long
'Calibration Routines
Declare Function r9215_GetCalibration Lib " r9215ps.dll" (ByVal iBrdID As Long, ByVal
iIndex As Long, _
                                                 ByVal iParam As Long, ByRef dbCalData As
Double) As Long
```

Constant Declaration Module

```
Attribute VB_Name = "9215_Constants"
'====================================================================
' Model 9215 CONSTANTS
'====================================================================
' Module:       R9215 Constant declarations
' Copyright:    © 2004 Racal Instruments, Inc.
' Date:         06/21/2004
' Revised:      06/21/2004
'====================================================================
  '9215 GUI specific constants
  Global Const PROG_VERSION = "1.0 - 06/21/2004"     'Program Version
  Global Const PROG_NAME = "R9215 Dual DC Power Supply"
  Global Const MAX_BOARDS = 3
  Global Const BOARD_PRESENT = -1

  'Operating modes
  Global Const MODE_DUAL = 0
  Global Const MODE_PARALLEL = 1
  Global Const MODE_SERIES = 2
```

This page was left intentionally blank

Chapter 7 –Product Support

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.

Reshipment Instructions

Use the original packing material when returning Model 9215 to Racal Instruments for calibration or servicing. The original shipping crate 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
Note: We do not accept "collect" shipments.

Ship instruments to nearest support office.

This page left intentionally blank.

Appendix A – Calibration Tolerances

Calibration Tolerances

CAL FACTOR	DESCRIPTION	TOLERANCE
VOFS	Voltage calibration offset	0.1% of FS
VGAIN	Voltage calibration gain	
IOFS	Current calibration offset	0.1% of FS
IGAIN	Current calibration gain	
VRBOFS	Voltage read-back calibration offset	0.1% of FS
VRBGAIN	Voltage read-back calibration gain	
IRBOFS	Current read-back calibration offset	0.1% of FS
IRBGAIN	Current read-back calibration gain	
VTRPOFS	Over-voltage trip calibration offset	1% of FS
VTRPGAIN	Over-voltage trip calibration gain	
ITRPOFS	Over-current trip calibration offset	1% of FS
ITRPGAIN	Over-current trip calibration gain	

This page was left intentionally blank

Appendix B – Specifications

General

Type	Programmable output
Voltage	0-30 V
Power	30 W
Max Current	2 A
Programming Accuracy	14 mV
Load Regulation	150 mV for full load

Outputs

Modes of Operation:	Constant Voltage, Constant Current
Current Regulation:	Current Limit
Programming Resolution:	12 bits
Read-back Resolution:	12 bits
Remote/Local Sense:	Up to 2 VDC may be dropped across sense loads. This drop reduces the voltage available at the load.

Input Power

Configuration:	2 separate isolated DC Sources
Voltage:	48 VDC/1.25 A/60 W
Efficiency:	>80% (depends on programmed output voltage and current)

Protection Circuitry

Over Voltage:	100% of full scale.
Current Limit:	Programmable from 0 to FS. Output will automatically switch to constant current mode when limit is reached.
Current Trip:	100% of full scale (fixed).
Over Temperature:	Automatically disables output if maximum allowable temperature is exceeded. (80°C)
UUT Discharge:	Active when output is down-programmed. Circuit will discharge at a constant rate.
Short Circuit:	Outputs are protected in the event of a short across output terminals.
Isolation Relays:	Independently controllable for each output. Circuitry is provided to protect relays from current switching.

Physical

Format:	PXI single-slot
Size:	Single-width 3U PXI standard
Approximate Weight:	13 oz.
Operating Temperature:	0 to +40 °C
Non-operating Temperature:	-40 °C to +85 °C
Humidity:	20% - 95% non-condensing
Cooling:	PXI/Compact PCI Chassis airflow