

# LabView Interface For EDR Enhanced

## Third Party Interface Driver User's Manual

LabView Version 6*i* with 32-bit support

Eagle Technology – Cape Town, South Africa

Copyright © 2002

[www.eagle.co.za](http://www.eagle.co.za)

---

# Third Party Interface

## Data Acquisition and Process Control

---

© Eagle Technology  
31-35 Hout Street • Cape Town • South Africa  
Phone +27 21 423 4943 • Fax +27 21 424 4637  
Email [eagle@eagle.co.za](mailto:eagle@eagle.co.za)

**Copyright**

All rights reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or any means, electronic, mechanical, by photographing, recording, or otherwise without prior written permission.

Copyright © Eagle Technology, South Africa  
October 2003  
Revision 1.1

Information furnished in this manual is believed to be accurate and reliable; however no responsibility is assumed for its use, or any infringements of patents or other rights of third parties, which may result from its use.

Trademarks and Logos in this manual are the property of their respective owners.

**Product Warranty**

Eagle Technology, South Africa, warrants its products/software from defect in material and workmanship from confirmed date of purchase for a period of one year if the conditions listed below are met. The product warranty will call the Eagle Technology Data Acquisition Device short as **ETDAQD**.

- The warranty does not apply to an **ETDAQD** that has been previously repaired, altered, extended by any other company or individual outside the premises of Eagle Technology.
- That a qualified person configure and install the **ETDAQD**, and damages caused to a device during installation shall make the warranty void and null.
- The warranty will not apply to conditions where the **ETDAQD** has been operated in a manner exceeding its specifications.

Eagle Technology, South Africa, does not take responsibility or liability of consequential damages, project delays, damaging of equipment or capital loss as result of its products.

Eagle Technology, South Africa, holds the option and final decision to repair or replace any **ETDAQD**. Proof of purchase must be supplied when requesting a repair.

<b>TABLE OF CONTENTS</b>
--------------------------

<b>1</b>	<b>INTRODUCTION .....</b>	<b>4</b>
1.1	Architecture Overview .....	4
1.2	Supported Operating Systems .....	1
1.3	Features .....	1
1.4	Contact Details .....	1
<b>2</b>	<b>INSTALLATION .....</b>	<b>2</b>
2.1	Installation Instructions .....	2
2.2	Location of Files .....	2
<b>3</b>	<b>LABVIEW VI FUNCTIONS .....</b>	<b>3</b>
3.1	The Utility Functions .....	3
3.1.1	EDRE_Query .....	3
3.2	The Digital Input/Output Functions .....	3
3.2.1	EDRE_DioWrite .....	3
3.2.2	EDRE_DioRead .....	3
3.2.3	EDRE_MioConfig .....	4
3.3	The Counter/Timer Functions .....	4
3.3.1	EDRE_CTWrite .....	4
3.3.2	EDRE_CTRead .....	4
3.3.3	EDRE_CTConfig .....	4
3.3.4	EDRE_CTSofGate .....	4
3.4	The Interrupt Functions .....	5
3.4.1	EDRE_IntEnable .....	5
3.4.2	EDRE_IntDisable .....	5
3.4.3	EDRE_IntConfigure .....	5
3.4.4	EDRE_WaitOnInterrupt .....	5
3.4.5	EDRE_ReleaseInterrupt .....	5
3.5	The Analog Output Functions .....	5
3.5.1	EDRE_DAWrite .....	5
3.5.2	EDRE_DAConfig .....	6
3.5.3	EDRE_DAControl .....	6
3.5.4	EDRE_DAUpdateData .....	6
3.6	The Analog Input Functions .....	6
3.6.1	EDRE_ADSingle .....	6
3.6.2	EDRE_ADConfig .....	7
3.6.3	EDRE_ADConfig703 .....	7
3.6.4	EDRE_ADStart .....	7
3.6.5	EDRE_ADStop .....	8
3.6.6	EDRE_ADGetData .....	8
3.6.7	EDRE_ADGetDataRaw .....	8
3.6.8	EDRE_ADOpenStreamFile .....	8
3.6.9	EDRE_ADCLoseStreamFile .....	9
3.6.10	EDRE_ADGetDataFromFile .....	9

3.6.11	EDRE_ADGetDataRawFromFile .....	9
<b>3.7</b>	<b>The Temperature Functions.....</b>	<b>9</b>
3.7.1	EDRE_CalcCJmC .....	9
3.7.2	EDRE_CalcRTDmC.....	9
3.7.3	EDRE_CalcTCmC.....	10
<b>3.8</b>	<b>The String Functions.....</b>	<b>10</b>
3.8.1	EDRE_StrBoardName .....	10
3.8.2	EDRE_StrError.....	10
<b>4</b>	<b>EASY DAQ VI FUNCTIONS.....</b>	<b>11</b>
<b>4.1</b>	<b>Easy Digital I/O.....</b>	<b>11</b>
4.1.1	Read Digital Line.....	11
4.1.2	Read Digital Port.....	11
4.1.3	Write Digital Line.....	11
4.1.4	Write Digital Port.....	11
<b>4.2</b>	<b>Easy Analog Output.....</b>	<b>12</b>
4.2.1	DA Write Channel .....	12
4.2.2	DA Write Channels.....	12
4.2.3	DA Generate Waveform.....	12
4.2.4	766 DA Generate Waveforms.....	12
4.2.5	DA Config Waveform .....	12
4.2.6	DA Update Channel .....	13
4.2.7	DA Start.....	13
4.2.8	DA Stop.....	13
<b>4.3</b>	<b>Easy Analog Input.....</b>	<b>13</b>
4.3.1	AD Acquire Waveform.....	13
4.3.2	AD Acquire Waveforms.....	13
4.3.3	AD Sample Channel.....	14
4.3.4	AD Sample Channels.....	14
4.3.5	AD Config Waveform .....	14
4.3.6	AD Get Data .....	14
4.3.7	AD Start.....	14
4.3.8	AD Stop.....	15
<b>4.4</b>	<b>Easy Temperature.....</b>	<b>15</b>
4.4.1	Temp Sample Channel .....	15
<b>A.</b>	<b>CONFIGURATION CONSTANTS.....</b>	<b>16</b>
<b>1.</b>	<b>Gain.....</b>	<b>16</b>
1	PCI703-16/32/64/A.....	16
2	PCI703S-8/16/A .....	16
3	PCI-725/726/730.....	16
4	PCI-30F/G .....	16
5	PCI-773T/R 16.....	16
6	USB-26/30 & SRL-26/30.....	17
<b>2.</b>	<b>Clock Source.....</b>	<b>17</b>
1	PCI703/S-8/16/32/64/A.....	17
2	PCI-725/726/730.....	17
3	PCI-30F/G .....	17
4	USB-26/30 & SRL-26/30.....	17
<b>3.</b>	<b>Trigger Source .....</b>	<b>17</b>
1	PCI703/S-8/16/64/A .....	17
2	PCI-725/726/730.....	17
3	PCI-30FG .....	17
4	USB-26/30 & SRL-26/30.....	17

<b>4.</b>	<b>Trigger Mode</b> .....	<b>17</b>
1	PCI703/S-8/16/64/A .....	18
2	PCI-725/726/730 .....	18
3	PCI-30FG .....	18
4	USB-26/30 & SRL-26/30 .....	18
<b>5.</b>	<b>Range</b> .....	<b>18</b>
1	PCI703/S-8/16/32/64/A .....	18
2	PCI-725/726/730 .....	18
3	PCI-30F/G .....	18
4	USB-26/30 & SRL-26/30 .....	18
<b>6.</b>	<b>Reference</b> .....	<b>18</b>
1	PCI703/S-8/16/64/A .....	18
2	PCI-725/726/730 .....	18
3	PCI-30FG .....	19
<b>7.</b>	<b>Burst</b> .....	<b>19</b>
1	PCI703/S-8/16/64/A .....	19
2	PCI-725/726/730 .....	19
3	PCI-30F/G .....	19
4	USB-26/30 & SRL-26/30 .....	19
<b>8.</b>	<b>Thermocouple Type Codes</b> .....	<b>19</b>
1	PCI-773T/R 16 & USB-73 & SRL-73 .....	19
<b>B.</b>	<b>ADDITIONAL INFORMATION</b> .....	<b>20</b>



# 1 Introduction

EDR Enhanced (EDRE) is a powerful application program interface (API) between your data acquisition and control application and Eagle Technologies line of plug in boards for PCs. It is a software development kit (SDK) designed to simplify the programming of the Eagle data acquisition cards but not sacrificing any power in its functionality. An extension to EDR Enhanced is the third party support for LabVIEW through the interface VI's.

## 1.1 Architecture Overview

The EDRE API consists of three layers of software, Namely the device driver, DLL and the interface VI's. Each layer has got unique futures and is designed to perform a specific task.

At the lowest level you will find a device driver that is specific for each operating system. The device drivers are easy to install and support Plug and Play. A control panel applet supplies information of the current hardware that is installed. General information is also available like serial number, manufacturing date, etc.

The middle layer is implemented in a Windows dynamic link library and contains a lot of the intelligence to isolate the difficulty of communicating to the driver from the COM control or application. The DLL is platform independent and contains a database of all the drivers that is currently supported.

The third layer of software, which also serves as the API, is implemented in the VI's which interface/translates between LabVIEW and the EDRE API.

The figure below shows the different parts of EDR Enhanced.

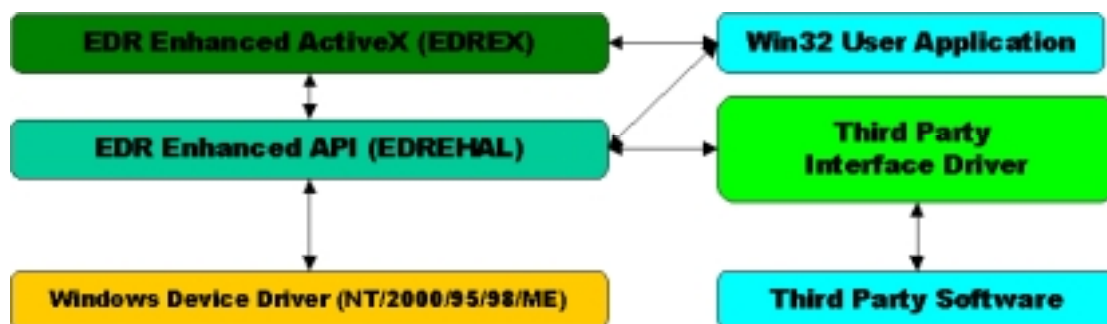


Figure 1-1 EDR Enhanced Architecture

---

## 1.2 Supported Operating Systems

The EDR Enhanced SDK supports the following operating systems.

- Windows 95.
- Windows 98.
- Windows 98 SE
- Windows 2000 Professional Edition
- Windows 2000 Server Edition
- Windows ME.
- Windows XP Home Edition
- Windows XP Professional Edition

---

## 1.3 Features

- Easy to install.
- Support all LabView functions
- Quick to learn.

---

## 1.4 Contact Details

Below are the contact details of Eagle Technology.

### **Eagle Technology**

PO Box 4376

Cape Town

8000

South Africa

Telephone +27 (021) 423 4943

Fax +27 (021) 424 4637

E-Mail [eagle@eagle.co.za](mailto:eagle@eagle.co.za)

Website <http://www.eagle.co.za>



## 2 Installation

This chapter explains how to install the TestPoint interface driver for EDR Enhanced. Follow the instruction closely to get your driver going quickly.

---

### 2.1 Installation Instructions

- Step 1** Run *EDRE\_LABVIEW.exe* file found in the <EAGLECD>\EDRE\DRIVERS3P\LABVIEW\SETUP directory.
- Step 2** On the Functions Palette select Options.  
In the Palette Set box, select EDREDAQ to get to Eagle EDREDAQ Functions Palette.
- Step 3** Install the EDR Enhanced API, *EDREAPI.EXE*. The installation file can be found in the following directory, <EAGLECD>\EDREAPI.

---

### 2.2 Location of Files

The installation files can be downloaded from the web or can be found in the Eagle CD-Rom.

#### Eagle Technology CD-Rom

<EAGLECD>\EDRE\DRIVERS3P\LABVIEW

#### Eagle Technology Website

<http://www.eagle.co.za>

<http://www.eagledaq.com>



## 3 LabView VI Functions

The chapter **LabView VI Functions** discusses the various functions that are exported by the EDR Enhanced Windows 32-bit DLL, **EDRAPI.DLL**. The chapter is broken up in sections dealing with each hardware sub-system.

### 3.1 The Utility Functions

#### 3.1.1 EDRE\_Query

Parameter	Type	Description
SerialNumber	Unsigned long	Board's serial number.
QueryCode	Unsigned long	Query code. See appendix for a list of query code.
Param	Unsigned long	Extra query parameter.
Return	Long	Indicates success or failure

**EDRE\_Query** is used to for many functions; to the status of a board, driver version, DLL version and many more functions.

### 3.2 The Digital Input/Output Functions

#### 3.2.1 EDRE\_DioWrite

Parameter	Type	Description
SerialNumber	Unsigned long	Board's serial number.
Port	Unsigned long	Port number
Value	Unsigned long	Value to write to port
Return	Long	Indicates success or failure

**EDRE\_DioWrite** is used to write digital value to a specific output port. If the port is configurable, the driver will automatically configure the port for output. The returned error code will indicate if the operation was successful.

#### 3.2.2 EDRE\_DioRead

Parameter	Type	Description
SerialNumber	Unsigned long	Board's serial number.
Port	Unsigned long	Port number
*Value	Pointer to a unsigned long	Value to read from port
Return	Long	Indicates success or failure

**EDRE\_DioRead** is used to read a digital value from a specific digital input port. If the port is configurable, the driver will configure it for input. The returned error code will indicate if the operation was successful.

### 3.2.3 EDRE\_MioConfig

Parameter	Type	Description
SerialNumber	Unsigned long	Board's serial number.
Port	Unsigned long	Port number
Value	Unsigned long	Value to write to port
Return	Long	Indicates success or failure

**EDRE\_DioConfig** is used to configure multiplexed digital I/O ports. The returned error code will indicate if the operation was successful.

## 3.3 The Counter/Timer Functions

### 3.3.1 EDRE\_CTWrite

Parameter	Type	Description
SerialNumber	Unsigned long	Board's serial number.
Ct	Unsigned long	Counter number
Value	Unsigned long	Value to write to counter
Return	Long	Indicates success or failure

**EDRE\_CtWrite** is used to write a value to a counter timer. The returned error code will indicate if the operation was successful.

### 3.3.2 EDRE\_CTRead

Parameter	Type	Description
SerialNumber	Unsigned long	Board's serial number.
Ct	Unsigned long	Counter number
*Value	Pointer to a unsigned long	Value to read from counter
Return	Long	Indicates success or failure

**EDRE\_CtRead** is used to read a value from a counter. The returned error code will indicate if the operation was successful.

### 3.3.3 EDRE\_CTConfig

Parameter	Type	Description
SerialNumber	Unsigned long	Board's serial number.
Ct	Unsigned long	Counter number
Mode	Unsigned long	Counter mode
Type	Unsigned long	Counter type
ClkSrc	Unsigned long	Clock source
GateSrc	Unsigned long	Gate source
Return	Long	Indicates success or failure

**EDRE\_CtConfig** is used to configure a counter timer. The returned error code will indicate if the operation was successful.

### 3.3.4 EDRE\_CTSoftGate

Parameter	Type	Description
SerialNumber	Unsigned long	Board's serial number.
Ct	Unsigned long	Counter number
Gate	Unsigned long	Gate value
Return	Long	Indicates success or failure

**EDRE\_CtConfig** is used to control the software gate, if configured for software gating. The returned error code will indicate if the operation was successful.

## 3.4 The Interrupt Functions

### 3.4.1 EDRE\_IntEnable

Parameter	Type	Description
SerialNumber	Unsigned long	Board's serial number.
Return	Long	Indicates success or failure

**EDRE\_IntEnable** is used to connect the device interrupt system to the bus. A returned error code will indicate if the operation was successful.

### 3.4.2 EDRE\_IntDisable

Parameter	Type	Description
SerialNumber	Unsigned long	Board's serial number.
Return	Long	Indicates success or failure

**EDRE\_IntDisable** is used to disconnect the device interrupt system from the bus. A returned error code will indicate if the operation was successful.

### 3.4.3 EDRE\_IntConfigure

Parameter	Type	Description
SerialNumber	Unsigned long	Board's serial number.
Src	Unsigned long	Interrupt source
Mode	Unsigned long	Interrupt mode
Type	Unsigned long	Interrupt type
Return	Long	Indicates success or failure

**EDRE\_IntConfigure** is used to configure the interrupt system. This function is only used where interrupts needs to be transferred to user space. A returned error code will indicate if the operation was successful.

### 3.4.4 EDRE\_WaitOnInterrupt

Parameter	Type	Description
SerialNumber	Unsigned long	Board's serial number.
Return	Long	Indicates success or failure

**EDRE\_WaitOnInterrupt** is used to wait for an interrupt to occur. The function makes use of overlapped I/O, so it will hang until an interrupt occur. If it needs to be release early use the interrupt release function. The returned code will indicate which interrupt was triggered. Please consult the hardware manual for the specific equipment.

### 3.4.5 EDRE\_ReleaseInterrupt

Parameter	Type	Description
SerialNumber	Unsigned long	Board's serial number.
Return	Long	Indicates success or failure

**EDRE\_ReleaseInterrupt** is used to release a call on **EDRE\_WaitOnInterrupt**. A returned error code will indicate if the operation was successful.

## 3.5 The Analog Output Functions

### 3.5.1 EDRE\_DAWrite

Parameter	Type	Description
SerialNumber	Unsigned long	Board's serial number.
Channel	Unsigned long	DAC channel
UVoltage	Long	Microvoltage
Return	Long	Indicates success or failure

**EDRE\_DAWrite** is used to output a voltage on a DAC channel. A returned error code will indicate if the operation was successful.

### 3.5.2 EDRE\_DAConfig

Parameter	Type	Description
SerialNumber	Unsigned long	Board's serial number.
Channel	Unsigned long	DAC channel
Frequency	Unsigned long	Update frequency
ClkSrc	Unsigned long	Clock source
GateSrc	Unsigned long	Gate source
Continuous	Unsigned long	DAC mode
Length	Unsigned long	Size in samples of buffer
*uVoltage	Pointer to a long buffer	Microvoltage buffer
Return	Long	Indicates success or failure

**EDRE\_DAConfig** is used to configure the ADC system. If the hardware supports independent, block or combined channels, the hardware will be configured in that way. Meaning the implementation of the channel parameter might change according to the hardware. Please the hardware manual for a particular piece of equipment. The entire configuration is done in one function call. The initial data must also be supplied. The returned error code will indicate if any errors occurred.

### 3.5.3 EDRE\_DAControl

Parameter	Type	Description
SerialNumber	Unsigned long	Board's serial number.
Channel	Unsigned long	DAC channel
Command	Unsigned long	1 : Start 2 : Stop
Return	Long	Indicates success or failure

**EDRE\_DAControl** is used to control a specific, block or combined set of channels depending on the hardware. A returned error code will indicate if the operation was successful.

### 3.5.4 EDRE\_DAUpdateData

Parameter	Type	Description
SerialNumber	Unsigned long	Board's serial number.
Channel	Unsigned long	DAC channel
Length	Unsigned long	Size in samples of buffer
*uVoltage	Pointer to a long buffer	Microvoltage buffer
Return	Long	Indicates success or failure

**EDRE\_DAUpdateData** is used to update the driver buffer for a specific or group of channels. A query can be made to check the space available in the driver buffer. Also query for any errors that might have occurred A returned error code will indicate if the operation was successful.

---

## 3.6 The Analog Input Functions

This section deals with functions relating to analog input functions.

### 3.6.1 EDRE\_ADSSingle

Parameter	Type	Description
SerialNumber	Unsigned long	Board's serial number.
Channel	Unsigned long	ADC channel.
Gain	Unsigned long	ADC gain.
Range	Unsigned long	ADC range.
*uVoltage	Long variable to hold microvolt value	Converted voltage
Return	Long	Indicates success or failure

**EDRE\_ADSSingle** is used to read a single ADC channel with specific settings specified by *Gain* and *Range*. The method used to read the voltage is programmed I/O. An error code is return to indicate the successfulness of the called function.

### 3.6.2 EDRE\_ADConfig

Parameter	Type	Description
SerialNumber	Unsigned long	Board's serial number.
*Frequency	Pointer to a unsigned long	Sampling frequency
ClkSrc	Unsigned long	ADC clock source.
Burst	Unsigned long	0: Disable burst mode 1: Enable burst mode
Range	Unsigned long	ADC range
*Channel List	Pointer to an array of unsigned long	ADC channel list
*Gain List	Pointer to an array of unsigned long	ADC gain list
List Size	Unsigned long	Size of list.
Return	Long	Indicates success or failure

**EDRE\_ADConfig** is used to configure the ADC sub-system for channel list mode for the **PCI-30** boards. When called the hardware gets configured and ready to start scanning through the channel list. The use of the parameters is different for each series of boards. Make sure that you consult the specific hardware manual for the card that is being programmed. The returned error code will indicate success or failure. Also don't try and configure the board while it is busy. This will also result in a failed configuration call.

### 3.6.3 EDRE\_ADConfig703

Parameter	Type	Description
SerialNumber	32-bit unsigned long	Board's serial number.
*Frequency	Pointer to an unsigned long	Sampling frequency
Clock Source	Unsigned long	ADC clock source.
Trigger Source	Unsigned long	ADC Trigger Source
Trigger Mode	Unsigned long	ADC Trigger Mode
Burst	Not Used	
Range	Not Used	
*Channel List	Pointer to an array of unsigned long	ADC channel list
*Gain List G	Pointer to an array of unsigned long	Specifies the gain of the channel. See table on ADC gain codes.
*Gain List R	Pointer to an array of unsigned long	Specifies the range of the channel. See table on ADC range codes.
*Gain List F	Pointer to an array of unsigned long	Specifies the analog reference. 0: analog in sense pin. 1: analog ground.
List Size	Unsigned long	Size of list.
Return	Long	Indicates success or failure

**EDRE\_ADConfig703** is used to configure the ADC sub-system for channel list mode for the **PCI703** boards. When called the hardware gets configured and ready to start scanning through the channel list. The use of the parameters is different for each series of boards. Make sure that you consult the specific hardware manual for the card that is being programmed. The returned error code will indicate success or failure. Also don't try and configure the board while it is busy. This will also result in a failed configuration call.

### 3.6.4 EDRE\_ADStart

Parameter	Type	Description
SerialNumber	Unsigned long	Board's serial number.
Return	Long	Indicates success or failure

**EDRE\_ADStart** is used to start the ADC channel list process. The board will be enabled to start running through the channel list. Before the operation can be started, the board must first be configured. If not the operation will fail.

### 3.6.5 EDRE\_ADStop

Parameter	Type	Description
SerialNumber	Unsigned long	Board's serial number.
Return	Long	Indicates success or failure

**EDRE\_ADStop** is used to stop the ADC process.

### 3.6.6 EDRE\_ADGetData

Parameter	Type	Description
SerialNumber	Unsigned long	Board's serial number.
*Buf	Pointer to an array of long	Buffer to hold microvolt values returned.
*BufSize	Pointer to an array of unsigned long	Variable to hold the value of requested samples and the actual number of samples returned.
Return	Long	Indicates success or failure

**EDRE\_ADGetData** is used to retrieve data from the driver buffer when sampling data. Use **EDRE\_Query** to check if any data is available. The query code must **ADUNREAD**. The buffer size parameter must contain the number of samples requested. However the driver can change this number. So even less samples can be returned, but not more. Make sure to check this number when called. A returned error code will indicate if the function call was successful. Please note that the buffer contains data in microvolts. The buffer is organized in the same order as the channel list.

### 3.6.7 EDRE\_ADGetDataRaw

Parameter	Type	Description
SerialNumber	Unsigned long	Board's serial number.
*Buffer	Pointer to an array of unsigned integer	Buffer to hold binary values returned.
*BufSize	Pointer to an array of unsigned long	Variable to hold the value of requested samples and the actual number of samples returned.
Return	Long	Indicates success or failure

**EDRE\_ADGetDataRaw** is used to retrieve data from the driver buffer when sampling data. Use **EDRE\_Query** to check if any data is available. The query code must **ADUNREAD**. The buffer size parameter must contain the number of samples requested. However the driver can change this number. So even less samples can be returned, but not more. Make sure to check this number when called. A returned error code will indicate if the function call was successful. . Please note that the buffer contains data in raw binary format. The buffer is organized in the same order as the channel list.

### 3.6.8 EDRE\_ADOpenStreamFile

Parameter	Type	Description
SerialNumber	Unsigned long	Board's serial number.
Mode	Unsigned long	File Mode 0: Overwrite 1: Append
*Char	Pointer to an array of characters or string.	Specify file name and full path of streaming file.
Return	Long	Indicates success or failure

**EDRE\_ADOpenStreamFile** is used to stream data to file. If a streaming file is opened and **EDRE\_ADGetData(Raw)** get called, all data supplied by the driver gets written to the specified file.

### 3.6.9 EDRE\_ADCloseStreamFile

Parameter	Type	Description
SerialNumber	Unsigned long	Board's serial number.

**EDRE\_ADCloseStreamFile** is used to close and save a file that was opened by **EDRE\_ADOpenStreamFile**.

### 3.6.10 EDRE\_ADGetDataFromFile

Parameter	Type	Description
SerialNumber	Unsigned long	Board's serial number.
Start	Unsigned long	Start position in file.
*Buf	Pointer to an array of unsigned long	Buffer to hold microvolt values returned.
*BufSize	Pointer to an array of unsigned long	Variable to hold the value of requested samples and the actual number of samples returned.
Return	Long	Indicates success or failure

**EDRE\_ADGetDataFromFile** is used to retrieve data from a file. The starting position and number of samples can be specified. The *BufSize* parameter will specify the number of actual samples copied to the buffer.

### 3.6.11 EDRE\_ADGetDataRawFromFile

Parameter	Type	Description
SerialNumber	Unsigned long	Board's serial number.
Start	Unsigned long	Start position in file.
*Buffer	Pointer to an array of unsigned integer	Buffer to hold binary values returned.
*BufSize	Pointer to an array of unsigned long	Variable to hold the value of requested samples and the actual number of samples returned.
Return	Long	Indicates success or failure

**EDRE\_ADGetDataRawFromFile** is used to retrieve binary data from a file. The starting position and number of samples can be specified. The *BufSize* parameter will specify the number of actual samples copied to the buffer.

---

## 3.7 The Temperature Functions

### 3.7.1 EDRE\_CalcCJCmC

Parameter	Type	Description
Cjcuv	Long	Cold junction compensation channel microvoltage.
Return	Long	Returns cold junction temp in milli degrees.

**EDRE\_CalcCJCmC** is used to calculate the temperature of the cold junction channel.

### 3.7.2 EDRE\_CalcRTDmC

Parameter	Type	Description
Rtduv	Long	RTD microvolt
Return	Long	Returns RTD in milli degrees.

*EDRE\_CalcCJmC* is used to calculate the temperature of a RTD channel.

### 3.7.3 EDRE\_CalcTCmC

Parameter	Type	Description
Tctype	Long	Thermocouple type
Tcuv	Long	Thermocouple temp
Ambientmc	Long	Ambient temp
Return	Long	Returns thermo temp in milli degrees

*EDRE\_CalcTCmC* is used to calculate the temperature of a thermocouple channel.

---

## 3.8 The String Functions

### 3.8.1 EDRE\_StrBoardName

Parameter	Type	Description
SerialNumber	Unsigned long	Board's serial number.
*StrBoardName	Pointer to a character array or string	Variable to hold a string
Return	Long	Indicates success or failure

*EDRE\_StrBoardName* is used to retrieve a board's name or text description. The returned error code will indicate if the operation was successful.

### 3.8.2 EDRE\_StrError

Parameter	Type	Description
Error	Unsigned long	Error number
*StrBoardName	Pointer to a character array or string	Variable to hold a string
Return	Long	Indicates success or failure

*EDRE\_StrError* is used to retrieve a text description of a specific error code. The returned error code will indicate if the operation was successful.



## 4 Easy DAQ VI Functions

The chapter **Easy DAQ VI Functions** discusses various easy to use VI's. These VI's were written to make the world of data acquisition simple, quick and easy. The chapter is broken up in sections dealing with each hardware sub-system.

### 4.1 Easy Digital I/O

#### 4.1.1 Read Digital Line

Parameter	Type	Description
Device	Unsigned long	Device number.
Channel	String	Port number.
Line	Unsigned long	Line number.
Status	Boolean	Return state of line.

**Read Digital Line** is used to read a Boolean value from a specific digital input line. If the port is configurable, the driver will configure it for input.

#### 4.1.2 Read Digital Port

Parameter	Type	Description
Device	Unsigned long	Device number.
Channel	String	Port number.
Pattern	Unsigned short	Return binary value.

**Read Digital Port** is used to read a digital value from a specific digital input port. If the port is configurable, the driver will configure it for input.

#### 4.1.3 Write Digital Line

Parameter	Type	Description
Device	Unsigned long	Device number.
Channel	String	Port number.
Digital Line	Unsigned long	Line number.
Line State	Boolean	State of line.

**Write Digital Line** is used to write digital value to a specific output line. If the port is configurable, the driver will automatically configure the port for output.

#### 4.1.4 Write Digital Port

Parameter	Type	Description
Device	Unsigned long	Device number.
Channel	String	Port number.
Binary Value	Unsigned short	Binary value.

**Write Digital Port** is used to write digital value to a specific output port. If the port is configurable, the driver will automatically configure the port for output.

## 4.2 Easy Analog Output

This section deals with functions relating to analog output functions.

### 4.2.1 DA Write Channel

Parameter	Type	Description
Device	Unsigned long	Device number.
Channel	String	DAC channel.
Sample	Voltage	Voltage.

**DA Write Channel** is used to output a Voltage on a DAC channel.

### 4.2.2 DA Write Channels

Parameter	Type	Description
Device	Unsigned long	Device number.
Channels	String	Array of DAC channels
Values	Array of Voltages	Array of Voltage

**DA Write Channels** is used to output a Voltage on DAC channels.

### 4.2.3 DA Generate Waveform

Parameter	Type	Description
Device	Unsigned long	Device number.
Channel	String	DAC channel.
Samples per Second	Unsigned long	Updates/second.
Waveform	Waveform	Waveform of Voltages.

**DA Generate Waveform** is used to output a Waveform on a DAC channel. A returned error code will indicate if the operation was successful.

### 4.2.4 766 DA Generate Waveforms

Parameter	Type	Description
Device	Unsigned long	Device number.
Channel	Unsigned long	DAC channel to start from.
Update rate	Unsigned long	Updates per second.
Waveform	Pointer to a long buffer	Array of Micro Voltage.
Return	Long	Indicates success or failure .

**766 DA Generate Waveforms** is used to output Waveforms on the DAC channels. A returned error code will indicate if the operation was successful.

### 4.2.5 DA Config Waveform

Parameter	Type	Description
Device	Unsigned long	Device number.
Channel	String	DAC channel.
Counter Value	Unsigned long	Counter Value written to counter.
Waveform	Waveform	Waveform of Voltages.
Clock Source	Unsigned long	Clock Source
Gate Source	Unsigned long	Gate Source
Continuous	Boolean	Continuous Streaming or FIFO loop Mode.

Error	Long	Return error code.
-------	------	--------------------

**DA Config Waveform** is used to configure to board to stream data out.

#### 4.2.6 DA Update Channel

Parameter	Type	Description
Device	Unsigned long	Device number.
Channel	Unsigned long	DAC channel.
Update Waveform	Waveform	Sample values written to driver buffer.
Error	Long	Return error code.

**DA Update Channel** is used to output a Voltage on a DAC channel.

#### 4.2.7 DA Start

Parameter	Type	Description
Device	Unsigned long	Device number.
Channel	Unsigned long	DAC channel to start.
Error	Long	Return error code.

**DA Start** is used to start the streaming process.

#### 4.2.8 DA Stop

Parameter	Type	Description
Device	Unsigned long	Device number.
Channel	Unsigned long	DAC channel to start.
Error	Long	Return error code.

**DA Stop** is used to stop the streaming process.

---

### 4.3 Easy Analog Input

This section deals with functions relating to analog input functions.

#### 4.3.1 AD Acquire Waveform

Parameter	Type	Description
Device	Unsigned long	Device number.
Channel	String	ADC channel.
Samples	Unsigned long	Number of Samples.
Frequency	Unsigned long	Samples per Second.
High limit	Double	High limit of waveform to acquire.
Low limit	Double	Low limit of waveform to acquire.
Differential	Boolean	Set Differential mode
Waveform	Waveform	Return type waveform.

**AD Acquire Waveform** is used to acquire a number of analog samples from an ADC channel.

#### 4.3.2 AD Acquire Waveforms

Parameter	Type	Description
Device	Unsigned long	Device number.
Channels	String	String of channels.
Samples	Unsigned long	Number of Samples.
Frequency	Unsigned long	Samples per Second.
High limit	Double	High limit of waveform to acquire.
Low limit	Double	Low limit of waveform to

Differential Waveform	Boolean Array of Waveforms	acquire. Set Differential mode Return array of Waveform type.
-----------------------	-------------------------------	---

**AD Acquire Waveforms** is used to acquire a number of analog samples from the ADC channels.

### 4.3.3 AD Sample Channel

Parameter	Type	Description
Device	Unsigned long	Device number.
Channel	String	ADC channel.
High limit	Double	High limit of waveform to acquire.
Low limit	Double	Low limit of waveform to acquire.
Sample	Double	Return Sample Voltage.

**AD Sample Channel** is used to get an analog sample from the ADC channel.

### 4.3.4 AD Sample Channels

Parameter	Type	Description
Device	Unsigned long	Device number.
Channels	String	String of channels.
High limit	Double	High limit of waveform to acquire.
Low limit	Double	Low limit of waveform to acquire.
Samples	Pointer to a Double buffer	Return array of Sample Voltages.

**AD Sample Channels** is used to get analog samples from ADC channels.

### 4.3.5 AD Config Waveform

Parameter	Type	Description
Device	Unsigned long	Device number.
Channels	String	String of channels.
Samples	Unsigned long	Number of Samples.
Frequency	Unsigned long	Samples per Second.
High limit	Double	High limit of waveform to acquire.
Low limit	Double	Low limit of waveform to acquire.
Differential	Boolean	Set Differential mode

**AD Config Waveform** is used to configure the board before streaming in data.

### 4.3.6 AD Get Data

Parameter	Type	Description
Device	Unsigned long	Device number.
Channels	String	String of channels.
Samples	Unsigned long	Number of Samples.
Waveform	Array of Waveforms	Return array of Waveform type.

**AD Get Data** is used to get blocks of data from the driver buffer.

### 4.3.7 AD Start

Parameter	Type	Description
-----------	------	-------------

Device	Unsigned long	Device number.
--------	---------------	----------------

**AD Start** is used to start streaming.

#### 4.3.8 AD Stop

Parameter	Type	Description
Device	Unsigned long	Device number.

**AD Stop** is used to stop streaming.

---

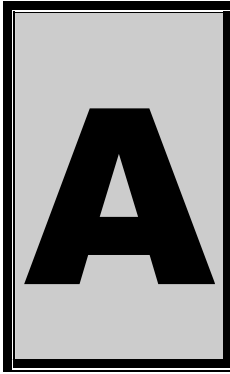
## 4.4 Easy Temperature

This section deals with functions relating to temperature input functions.

### 4.4.1 Temp Sample Channel

Parameter	Type	Description
Device	Unsigned long	Device number.
Channel	String	Temperature channel.
TC Type	Unsigned long	A number value to specify the Thermocouple Type find in appendices A.
Temp	Double	Return Temperature.

**Temp Sample Channel** is used to get the temperature from a Thermocouple channel.



## A. Configuration Constants

### 1. Gain

#### 1 PCI703-16/32/64/A

Gain	Value	Description
Gain 0.25	0	Gain of $\frac{1}{4}$ (+/- 10V, NU)
Gain 0.50	1	Gain of $\frac{1}{2}$ (+/- 5V, 0-10V)
Gain 1.00	2	Gain of 1 (+/- 2.5V, 0-5V)
Gain 2.50	3	Gain of 2.5(+/- 1V, 0-2V)
Gain 5.00	4	Gain of 5 (+/- 500mV, 0-1V)
Gain 10.0	5	Gain of 10 (+/- 250mV, 0-500mV)
Gain 25.0	6	Gain of 25 (+/- 100mV, 0-200mV)
Gain 50.0	7	Gain of 50 (+/- 50mV, 0-100mV)

#### 2 PCI703S-8/16/A

Gain	Value	Description
Gain 0.50	0	Gain of $\frac{1}{2}$ (+/- 5V)
Gain 1.00	1	Gain of 1 (+/- 2.5V)
Gain 2.50	2	Gain of 2.5 (+/- 1V)
Gain 5.00	3	Gain of 5 (+/- 500mV)
Gain 10.0	4	Gain of 10 (+/- 250mV)
Gain 25.0	5	Gain of 25 (+/- 100mV)
Gain 50.0	6	Gain of 50 (+/- 50mV)
Gain 100.0	7	Gain of 100(+/- 25mV)

#### 3 PCI-725/726/730

Gain	Value	Description
Gain 1.00	0	Gain of 1 (+/-2.5V)
Gain 0.50	1	Gain of 0.5 (+/-5V)
Gain 0.25	2	Gain of 0.25 (+/-10V)

#### 4 PCI-30F/G

Gain	Value	Description
Gain 1.00	0	Gain of 1
Gain 10.0	1	Gain of 10
Gain 100.0	2	Gain of 100
Gain 1000.0	3	Gain of 1000

#### 5 PCI-773T/R 16

Gain	Value	Description
Gain 1.00	0	Gain of 1 ( $\pm 2.5V$ )
Gain 10.0	1	Gain of 10 ( $\pm 250mV$ ) (Only available on Rev. 2 Board)
Gain 100.0	2	Gain of 100 ( $\pm 25mV$ ) (Only available on Rev. 2 Board)

SET BY HARDWARE (default 30.0)	3	Gain of 30 ( $\pm 70\text{mV}$ )
--------------------------------	---	----------------------------------

## 6 USB-26/30 & SRL-26/30

Gain	Value	Description
Gain 0.25	0	Gain of $\frac{1}{4}$ (+/- 10V, NU)
Gain 0.50	1	Gain of $\frac{1}{2}$ (+/- 5V, 0-10V)
Gain 1.00	2	Gain of 1 (+/- 2.5V, 0-5V)
Gain 2.50	3	Gain of 2.5 (+/- 1V, 0-2V)
Gain 5.00	4	Gain of 5 (+/- 500mV, 0-1V)
Gain 10.0	5	Gain of 10 (+/- 250mV, 0-500mV)
Gain 25.0	6	Gain of 25 (+/- 100mV, 0-200mV)
Gain 50.0	7	Gain of 50 (+/- 50mV, 0-100mV)
Gain 100.0	8	Gain of 100 (+/- 25mV, 0-50mV)

## 2. Clock Source

### 1 PCI703/S-8/16/32/64/A

Value	Description
0	Undefined
1	Internal
2	External

### 2 PCI-725/726/730

Value	Description
0	Not Used

### 3 PCI-30F/G

Value	Description
0	Internal clock to AD and external trigger
1	Internal clock to AD
2	External trigger to AD
3	Internal clock to AD with external trigger as gate

### 4 USB-26/30 & SRL-26/30

See USB/SRL manual.

## 3. Trigger Source

### 1 PCI703/S-8/16/64/A

Value	Description
0	Internal
1	Reference (NOT SUPPORTED BY PCI 703S)
2	External

### 2 PCI-725/726/730

Value	Description
0	Internal
1	External

### 3 PCI-30FG

Value	Description
0	Not Used

### 4 USB-26/30 & SRL-26/30

See USB/SRL manual.

## 4. Trigger Mode

## 1 PCI703/S-8/16/64/A

Value	Description
0	Positive
1	Negative
2	Rising
3	Falling

## 2 PCI-725/726/730

Value	Description
0	Not Used

## 3 PCI-30FG

Value	Description
0	Not Used

## 4 USB-26/30 & SRL-26/30

See USB/SRL manual.

## 5. Range

### 1 PCI703/S-8/16/32/64/A

Name	Value	Description
UNIPOLAR, SINGLE ENDED	0	Channel is single ended unipolar input.
BIPOLAR, SINGLE ENDED	1	Channel is single ended bipolar input.
UNIPOLAR, DIFFERENTIAL	2	Channel is differential unipolar input.
BIPOLAR, DIFFERENTIAL	3	Channel is differential bipolar input.

### 2 PCI-725/726/730

Value	Description
0	SINGLE ENDED
1	DIFFERENTIAL

### 3 PCI-30F/G

Name	Value	Description
Single Ended	0	-5V to 5V
Single Ended	1	0V to 10V
Single Ended	2	-10V to 10V
Differential	0	-5V to 5V
Differential	1	0V to 10V
Differential	2	-10V to 10V

### 4 USB-26/30 & SRL-26/30

Name	Value	Description
UNIPOLAR, SINGLE ENDED	0	Channel is single ended unipolar input.
BIPOLAR, SINGLE ENDED	1	Channel is single ended bipolar input.
UNIPOLAR, DIFFERENTIAL	2	Channel is differential unipolar input.
BIPOLAR, DIFFERENTIAL	3	Channel is differential bipolar input.

## 6. Reference

### 1 PCI703/S-8/16/64/A

Value	Description
0	Analog in sense pin.
1	Analog ground.

### 2 PCI-725/726/730

Value	Description
0	Not Used

### 3 PCI-30FG

Value	Description
0	Not Used

## 7. Burst

### 1 PCI703/S-8/16/64/A

Value	Description
0	Not Used

### 2 PCI-725/726/730

Value	Description
0	Not Used

### 3 PCI-30F/G

Value	Description
0	Disable
1	Enable

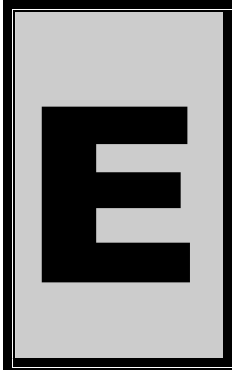
### 4 USB-26/30 & SRL-26/30

Value	Description
0	Not Used

## 8. Thermocouple Type Codes

### 1 PCI-773T/R 16 & USB-73 & SRL-73

Value	Thermocouple Type
0	Type J
1	Type K
2	Type E
3	Type T
4	Type S
5	Type R
6	Type B
7	Type N



## B. Additional Information

For more information please contact Eagle Technology directly or visit our website [www.eagle.co.za](http://www.eagle.co.za). They can also be emailed at [eagle@eagle.co.za](mailto:eagle@eagle.co.za).