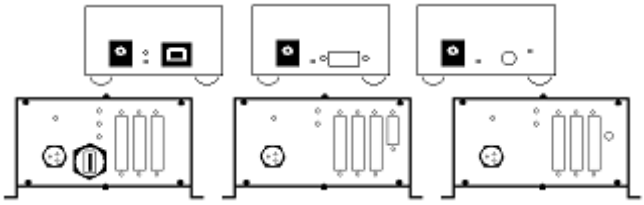
µDAQ & Rugged µDAQ Digital I/O, Analog I/O & Temperature

User's Manual for

USB µDAQ, USB Rugged µDAQ
SERIAL µDAQ, SERIAL Rugged µDAQ
WIRELESS µDAQ, WIRELESS Rugged µDAQ
☺ Preliminary Release ☺



WaveView for WINDOWS

POWERED BY EDR ENHANCED

POWERED BY RUGGED µDAQ MicroDAQ Data Acquisition

POWERED BY µDAQ MicroDAQ Data Acquisition

Windows Mobile

Microsoft Windows

Linux ™

USB 1.1
USB 2.0

Wireless with Bluetooth® Technology

Serial
RS232 & RS485

Eagle Technology – Cape Town, South Africa

www.eagledaq.com www.eagle.co.za

µDAQ & Rugged µDAQ Remote Devices

Data Acquisition and Process Control

31-35 Hout Street • Cape Town • South Africa
Phone +27 21 423 4943 • Fax +27 21 424 4637
Email eagle@eagle.co.za

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

Liability Notice
Eagle Technology, South Africa, does not take responsibility or liability of consequential damages, project delays, damaging of equipment or capital loss as a result of its products, hardware and software. Eagle Technology does not authorize its products for use as critical instruments in life support systems where malfunction or failure may result in significant injury or loss of life. The inclusion of Eagle Technology instruments in such systems implies the user assumes all risk of such use and in doing so indemnifies Eagle Technology against all charges.

Copyright, Product & Liability Agreement
By using the Eagle Technology software and hardware it binds you to the above Copyright notice, Product Warranty and Liability notice.

Third Party Trademarks and Copyrights
Windows™ is a trademark of the Microsoft® Corporation
Pocket PC is a trademark of the Microsoft® Corporation
The Windows Logo is a copyright of the Microsoft™ Corporation
The USB Logo is the copyright of the USB Implementers Forum
Linux® is a registered trademark of Linus Torvalds.
Bluetooth® is a registered trademark of the Bluetooth Special Interest Group.
VEE Pro™ is a trademark of Agilent Technologies.
Labview™ is a trademark of National Instruments.
TestPoint™ is a trademark of Capital Equipment.
Intel® is a registered trademark of the Intel Corporation
Any other unmentioned trademarks are the property of their respective owners.

Table of Contents

List of Figures

List of Tables

1 Introduction

The μDAQ products consist of two main types, the commercial μDAQ series and the industrial Rugged μDAQ series. They currently support three communication protocols, USB, serial communications and wireless communications. The μDAQ & Rugged μDAQ is digital input/output, counter-timer, analog I/O and temperature data acquisition devices. All the devices come with a common software interface and all communications are handled for the programmer or application. No detailed knowledge is necessary of how to do serial communications or to control USB devices. The devices are also interchangeable with minor common properties that are different among units.

The μDAQ Digital I/O series has support for 24 to 120 digital input/output lines and 6 counters. They are available in five basic models, 24, 48, 72, 96 and a 120 digital input/output channels. The models support USB, serial and wireless communication protocols.

The Rugged μDAQ Digital I/O series has support for 24 to 168 digital input/output lines and 6 counters. They are available seven basic models, 24, 48, 72, 96, 120, 144 and 168 digital input/output channels. The models support USB, serial and wireless communication protocols.

The μDAQ Analog Input series support 16 analog inputs and 4 analog outputs. It's available in two models, the 26 that excludes analog outputs and the 30 that include the analog outputs. It has full support for channel & gain list scanning. The USB has a much faster transfer rate than serial and wireless.

The Rugged μDAQ Analog Input series support 16/32 analog inputs and 4/8 analog outputs. The number of channels depends on the version. The 26-model exclude analog outputs where the 30-model does have analog outputs. The 26A16-model has sixteen analog input channels and the 26A32-model has 32 analog input channels. The 30A16 version has an additional four analog output channels and the 30A32 eight channels. Currently the 32-channel unit only supports channel and gain list scanning from 16 channels at a time. The serial and wireless units has a much slower transfer rate that the USB version.

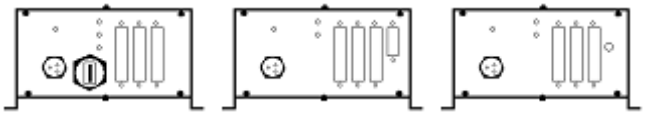
The μDAQ Temperature Input series supports 8, 16 and 32 temperature inputs. The two basic models are the 73T and 73R, which support thermocouple and RTD. The units come with external adaptors that serve as the connection point for the sensors.

The Rugged μDAQ Temperature Input series supports 16 and 32 temperature inputs. The two basic models are the 73T and 73R, which supports thermocouple and RTD. As with the μDAQ the units comes with external adaptors that serves as the connection point for the sensors.

1.1 What is µDAQ?

µDAQ is a commercial product that can be used in various everyday data acquisition and process control applications. It is typically used indoors in a protected environment. They must be protected against harsh conditions and will function well inside a lab or office space. The µDAQ units can be used in a factory environment but care must be taken when installing the unit. The unit is an ABS plastic type with standard DB25 connectors and bus and power connector. The unit runs off a DC power supply and depending on the version it comes in a two tier and three tier form.

1.2 What is Rugged µDAQ?



Rugged µDAQ is an industrial product that was engineered to withstand moisture, dust and vibration. The unit can operate within reasonable harsh conditions. The unit has a built-in power supply (AC & DC) and can be installed with an optional battery pack. All connectors and cables connect firmly to the unit and will not simply unplug. Care was taken during the design process to only use connectors that screw-on to prohibit unwanted unplugging of cables. The units support the industry standard DINRAIL mounting and can also be mounted via an anti-vibration kit. As with the µDAQ series the unit also use standard DB25 to make all external connections. Status lights will indicate the presents of power, charger activity and bus connectivity.

1.3 Features - µDAQ Commercial Version

The USB µDAQ series has some very unique features and are listed below:

- USB Revision 1.1 compliant @ full speed
- Powered externally or internally
- Intel 8255 compatible digital I/O ports
- Intel 8254 compatible counter-timer
- Easy to use interrupt system
- Channel list and voltage range scanning @ 250 KHz
- 14-bit Analog I/O resolution
- Temperature inputs
- Quick and effortlessly to install
- ABS plastic housing
- Supported by vast range of application modules

1.4 Features – Rugged DAQ Industrial Version

The USB Rugged DAQ series has some very unique features and are listed below:

- USB Revision 1.1 compliant @ full speed
- Rugged and moisture proof housing
- Externally Powered either AC/DC with screw-in secure connector
- Screw-in industrial type USB cable and connector
- Intel 8255 compatible digital I/O ports
- Intel 8254 compatible counter-timer.
- Channel list and voltage range scanning @ 250 KHz.

- 14-bit Analog I/O resolution.
- Temperature inputs.
- Vibration mounting kit
- DINRAIL mountable kit
- Supported by a vast range of application modules

1.5 Communication Protocols

The μDAQ and Rugged μDAQ devices support various protocols. This section will list the current protocols supported.

1.5.1 USB – Universal Serial Bus

μDAQ and Rugged μDAQ has full support for the universal serial bus protocol. All units come with high-speed USB cables. The μDAQ ships with a commercial peripheral USB cable where the Rugged μDAQ has an industrial type screw on connector to connect to the unit.

1.5.2 SRL – Serial Communication Protocol

μDAQ and Rugged μDAQ has support for the RS232 and RS485 communication protocol. The device is preset while manufactured for the specific protocol. The connector is a standard DB9 male connector.

1.5.3 BT – Wireless Protocol

The μDAQ and Rugged μDAQ use Bluetooth technology for it's wireless connections. The connections are transparent and the host computer uses a Bluetooth dongle to connect to the remote unit. The unit is supplied with an aerial to increase the connectivity range. No knowledge is necessary of how to operate Bluetooth.

1.5.4 Which protocol to use?

The type of protocol to use depends on the application. Each protocol has its own unique advantages. The rules below are a simple outline that can be followed to determine the best protocol for the application.

- For high speed applications up to five (5) meters from the controlling PC use USB. USB hubs can also be used to increase this distance, but distance will always be a problem. USB is easy to use and install. It offers a real plug and play solution.
- For applications where the unit needs to be very remote serial is a good option. For distances up to forty (40) meters use RS232 and greater than that use RS485. The RS485 protocol can cover long distances, even further than 500 meter, but the BAUD/transfer rate will drop the longer the cable. In such an application it is advisable to determine the best BAUD rate by doing a field tests.
- Wireless offers a unique solution where the connection is truly wireless. All connections are done automatically with no user intervention. The current model can cover distances up to thirty (30) meters. The range depends severely on the location and obstacles. The best range will be achieved in open air with no obstructions.

1.6 Operating Systems and API Software

The EDR Enhanced Software Development Kit supports both the Rugged μDAQ and μDAQ series. This kit contains a driver for Microsoft® Windows™ and the Linux kernel. Please consult Eagle Technology Software support for the latest information on which specific operating systems are supported.

Current Supported Operating System:

- Microsoft® Windows™ 2000

- Microsoft® Windows™ Millennium Edition
- Microsoft® Windows™ XP
- Microsoft® Windows™ Pocket PC 2003
- Linux Kernel 2.4 and later

The EDR Enhanced SDK serves as a common application and programming interface for all the units, no matter what the communications protocol. This single property makes the units easy to program because no knowledge is needed about the specific type of interface. It also means that the units can be controlled from the same application without any redevelopment when installing a different device. It also increases the life expectancy of the software application. The EDR Enhanced SDK comes with complete documentation and examples programs. For custom software the API is easy to learn shortening the learning curve. It also means that it's quicker to go into production.

1.7 µDAQ and Rugged µDAQ Part Numbers

Care was taken when part numbers were assigned to different products. This section will explain the format of µDAQ and Rugged µDAQ part numbers.

1.7.1 R-CCC AA-TT-DDD

The µDAQ and Rugged µDAQ part numbers are built-up in sections, depending if it supports analog I/O, temperature, digital I/O. The part number will also indicate if it's a normal µDAQ or Rugged µDAQ.

R	-CCC	-DDD	-AA	-TT	-RSXXX	-BNC	-PWR	-BAT

R This is a Rugged µDAQ if prefixed to the part number. If omitted it's a normal µDAQ.

CCC This is the connection protocol and can be **USB**, **SRL**, **BT**

DDD DDD is digital I/O units. The assigned part number can be **24**, **48**, **72**, **96**, **120**, **144** and **168**. The appended letter **A** means normal digital I/O and **C** mean with counter-timers.

AA This is analog if installed. It can be 26 or 30.
A postfix of **A** means 14-bit @ 250 KHz
A postfix of **B** means 14-bit @ 400 KHz (future product)
A postfix of **C** means 16-bit @ 100 KHz (future product)
The number of channels follows the postfix type.

TT This is the temperature if installed. The temperature units have a part number of **73 R** or **T**, where R is RTD and T is thermocouple. The number of channels follows the postfix type.

DDD DDD is digital I/O units. The assigned part number can be **24**, **48**, **72**, **96**, **120**, **144** and **168**. The appended letter **A** means normal digital I/O and **C** mean with counter-timers.

RSXXX If serial the part number will have the serial interface type appended. RS232 or RS485

BNC If a µDAQ BNC unit

PWR If a Rugged µDAQ it will indicate the power input type

BAT If a Rugged µDAQ it will indicate if backup battery pack is installed

1.7.2 Universal Serial Bus

This device type is denoted with the characters **USB**.

1.7.3 Wireless Connection

This device type is denoted with the characters **BT**.

1.7.4 Serial Communication Protocol

This device type is denoted with the characters **SRL**. The serial devices have a specific postfix code that indicates the serial protocol type. The interface can either be **RS232** or **RS485**.

1.7.5 μDAQ BNC Connector Model

The commercial μDAQ unit has a model with BNC connectors. This model appends **BNC** to the end of the part number.

1.7.6 Rugged μDAQ Version

The Rugged version appends the character **R-** to the front of the part number. The Rugged μDAQ also has a built-in power supply and optional battery pack. The postfix part number as **AC** or **DC** for the power supply type, and **BAT** for battery pack and charger if installed.

1.7.7 Rugged μDAQ Power Interface

The Rugged μDAQ has either an **AC** or **DC** power interface. The end of part number will indicate the type. **AC** denotes and AC power interface and **DC** a DC power interface.

1.7.8 Rugged μDAQ Optional Battery Pack

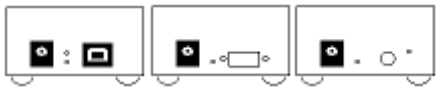
An optional battery pack with built-in charger can be installed. The part number will be appended with a **BAT** at the end.

1.7.9 Example Part Numbers

USB 144C	USB device with 144 digital I/O channels and counter-timers
USB 30-BNC	USB 30 device with BNC connectors.
R-SRL 30A16-73T16-AC-BAT	Rugged μDAQ Device with 16 analog inputs, 4 analog outputs and sixteen thermocouple channels. AC power Interface and battery-charger pack installed.

1.8 µDAQ Versions

The tables below list the various versions that are available.



1.8.1 Digital I/O Series

Feature	USB 24 SRL 24 BT 24	USB 48 SLR 48 BT 48	USB 72 SRL 72 BT 72	USB 96 SRL 96 BT 96	USB 120 SRL 120 BT 120
Number of digital channels – A/C Version	24	48	48	96	120
Number of counters – C Version	6	6	6	6	6
Number of 8255 compatible ports (8-bit)	3	6	9	12	15
Number of interrupt sources - A Version	0	0	0	0	0
Number of interrupt sources - C Version	6	6	6	6	6

Feature	USB SRL BT 62-16	USB SLR BT 62-32	USB SRL BT 63-16	USB SRL BT 63-32	USB SRL BT 69-16	USB SRL BT 69-32
Number of digital channels	40	56	40	56	40	56
Number of Opto isolated input ports (8-bit)	2	4	0	0	1	2
Number of Reed Relay output ports (8-bit)	0	0	2	4	1	2
Number of 8255 compatible ports (8-bit)	3	3	3	3	3	3

Table 1-1 µDAQ Digital I/O Versions

1.8.2 Analog Input Series

Feature	USB 26/30/A	USB 26/30/B	SRL 26/30 BT 26/30	USB 26/30/C
Number of digital I/O channels	24	24	24	24
Number of 8255 compatible ports (8-bit)	3	3	3	3
Number of analog input channels	16	16	16	16
Number of analog output channels	4 (30 only)	4 (30 only)	4 (30 only)	4 (30 only)
Analog input resolution	14-bit	14-bit	14-bit	16-bit
Maximum sampling speed	250 KHz	400 KHz	3 KHz	250 KHz
Analog output resolution	14-bit	14-bit	14-bit	16-bit

Table 1-2 µDAQ Analog Input Versions

1.8.3 Analog Input Series – BNC model

Feature	USB 26/30/A	USB 26/30/B	SRL 26/30 BT 26/30	USB 26/30/C
Number of digital I/O channels	24	24	24	24
Number of 8255 compatible ports (8-bit)	3	3	3	3
Number of analog input channels	8	8	8	8
Number of analog output channels	4 (30)	4 (30)	4 (30)	4 (30)
Analog input resolution	14-bit	14-bit	14-bit	16-bit
Maximum sampling speed	250 KHz	400 KHz	3 KHz	250 KHz
Analog output resolution	14-bit	14-bit	14-bit	16-bit

Table 1-3 µDAQ Analog Input Versions

1.8.4 Temperature Input Series

- R is RTD
- T is Thermocouple

Feature	USB 73R/T8 SRL 73R/T8 BT 73R/T8	USB 73R/T16 SRL 73R/T16 BT 73R/T16	USB 73R/T 32 SRL 73R/T32 BT 73R/T32
Number of digital I/O channels	24	24	24
Number of 8255 compatible ports (8-bit)	3	3	3
Number of temperature channels	8	16	32
Number of CJC channels	1	2	4

Analog input resolution	14-bit	14-bit	14-bit
Maximum sampling speed	100 Hz	100 Hz	100 Hz

Table 1-4 µDAQ Temperature Versions

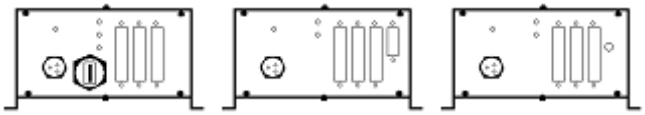
1.8.5 Hybrid Models

Feature	USB 30A16-73T16 SRL 30A16-73T16 BT 30A16-73T16	USB 30A32-73T16 STR 30A32-73T16 BT 30A32-73T16	USB 30A1673T32 SRL 30A1673T32 BT 30A1673T32
Number of digital I/O channels	24	24	24
Number of 8255 compatible ports (8-bit)	3	3	3
Number of temperature channels	16	16	32
Number of CJC channels	2	2	4
Number of analog inputs	16	32	16
Number of analog outputs	4	4	4

Table 1-5 µDAQ Hybrid Versions

1.9 Rugged DAQ Versions

The tables below list the various versions that are available.



1.9.1 Digital I/O Series

Feature	R-USB R-SRL R-BT 24	R-USB R-SRL R-SRL 48	R-USB R-SRL R-BT 72	R-USB R-SRL R-BT 96	R-USB R-SRL R-BT 120	R-USB R-SRL R-BT 144	R-USB R-SRL R-BT 168
Digital channels – A/C Version	24	48	48	96	120	144	168
Counters – C Version	6	6	6	6	6	6	6
8255 compatible ports (8-bit)	3	6	9	12	15	18	21
Interrupt sources – A Version	0	0	0	0	0	0	0
Interrupt sources – C Version	6	6	6	6	6	6	6

Feature	R-USB R-SRL R-BT 62-16	R-USB R-SRL R-SRL 62-32	R-USB R-SRL R-BT 63-16	R-USB R-SRL R-BT 63-32	R-USB R-SRL R-BT 69-16	R-USB R-SRL R-BT 69-32
Digital channels	40	56	40	56	40	56
Number of Opto isolated input ports (8-bit)	2	4	0	0	1	2
Number of Reed Relay output ports (8-bit)	0	0	2	4	1	2
Number of 8255 compatible ports (8-bit)	3	3	3	3	3	3

Table 1-6 Rugged µDAQ Digital I/O Versions

1.9.2 Analog Input Series

Feature	R-USB 26A16	R-USB 26A32	R-USB 30A16	R-USB 30A32	R-SRL R-BT 26A16	R-SRL R-BT 26A32	R-SRL R-BT 30A16	R-SRL R-BT 30A32
Digital I/O channels	24	24	24	24	24	24	24	24
8255 compatible ports (8-bit)	3	3	3	3	3	3	3	3
Analog input channels	16	16	32	32	16	16	32	32
Analog output channels	0	0	4	4	0	0	4	4
Analog input resolution	14-bit	14-bit	14-bit	14-bit	14-bit	14-bit	14-bit	14-bit
Maximum sampling speed	250 KHz	250 KHz	250 KHz	250 KHz	3 KHz	3 KHz	3 KHz	3 KHz
Analog output resolution	0	0	14-bit	14-bit	0	0	14-bit	14-bit

Table 1-7 Rugged µDAQ Analog Input Versions

1.9.3 Temperature Series

Feature	R-USB 73R/T16 R-SRL 73R/T16 R-BT 73R/T16	USB 73R/T 32 SRL 73R/T32 BT 73R/T32
Number of digital I/O channels	24	24
Number of 8255 compatible ports (8-bit)	3	3
Number of temperature channels	8	32
Number of CJC channels	1	4
Analog input resolution	14-bit	14-bit
Maximum sampling speed	100 Hz	100 Hz

Table 1-8 Rugged μDAQ Temperature Versions

1.9.4 Hybrids

Currently there are only a few hybrids available. Please consult with Eagle Technology for a model that can be adopted for a specific application.

Feature	R-USB 30A16-73R/T16	R-USB 30A32-73R/T16	R-USB 30A316-73R/T32
Digital I/O channels	24	24	24
8255 compatible ports (8-bit)	3	3	3
Analog input channels	16	32	16
Analog output channels	4	4	4
Analog input resolution	14-bit	14-bit	14-bit
Maximum sampling speed	250 KHz	250 KHz	250 KHz
Analog output resolution	14-bit	14-bit	14-bit
A Number of temperature channels	16	16	32
Number of CJC channels	2	2	4

Table 1-9 μRugged μDAQ Hybrids

1.10 Applications

The USB μDAQ & Rugged DAQ series can be used in the following applications:

- Automation test equipment.
- TTL compatible status monitoring.
- Plant/Factory process control.
- Pulse counting.
- Frequency measurement.
- Frequency generation.
- Controlling and monitoring of any TTL compatible equipment.
- Mobile computing.
- Laboratory applications

1.11 Key Specifications

- 3,6,12 or 15 x 8-bit ports.
- 6 x 16-bit counters.
- Fully programmable digital input/output system.
- Fully programmable counter-timer system.
- Fully programmable interrupt support.
- 14-bit Resolution analog input system with a max range of ±10 volt.
- Fully programmable channel/gain list @ 250 KHz for USB
- 14-bit Resolution analog output system with a range of ±10 volt.

1.12 Software Support

The USB µDAQ series is supported by EDR Enhanced and has an extensive range of examples. The software will help you to get your hardware going very quickly. It also makes it easy to develop complicated control applications. All operating system drivers, utility and test software are supplied on the EDR Enhanced CD-Rom. The latest drivers can also be downloaded from the Eagle Technology website. For further support information see the Contact Details section.

1.13 Contact Details

Below are the contact details of Eagle Technology, South Africa.

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
Website http://www.eagledaq.com

2

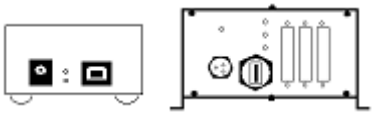
2 Getting Started

This chapter describes how to install and configure the µDAQ and Rugged µDAQ device for the first time. Minimal configuration is necessary; almost all settings are done through software. The chapter will deal with each type of protocol separately.

2.1 Package Items

The package of items differs by device type. Depending on the device type accessories will be included like power supplies, USB or serial cable and third party devices.

2.1.1 USB Devices



2.1.1.1 USB µDAQ Digital I/O and Analog I/O Device

1. µDAQ unit
2. USB 2.0 compliant peripheral cable
3. Universal power supply if listed
4. Software CD Rom

2.1.1.2 USB µDAQ Temperature Device

1. µDAQ unit
2. USB 2.0 compliant peripheral cable
3. Universal power supply
4. Software CD Rom
5. RTD or Thermocouple adaptor unit

2.1.1.3 USB Rugged µDAQ Digital I/O and Analog I/O Device

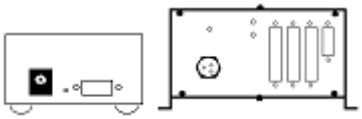
1. Rugged µDAQ unit
2. Industrial power cable
3. USB 2.0 compliant industrial peripheral cable
4. Rubber mounting vibration kit
5. DINRAIL mounting kit
6. Software CD Rom

2.1.1.4 USB Rugged µDAQ Temperature Device

1. Rugged µDAQ unit
2. RTD or Thermocouple adaptor unit

3. Industrial power cable
4. USB 2.0 compliant industrial peripheral cable
5. Rubber mounting vibration kit
6. DINRAIL mounting kit
7. Software CD Rom

2.1.2 Serial Devices



2.1.2.1 Serial µDAQ Digital I/O and Analog Device

1. µDAQ unit
2. Serial cable for RS232 or a DB9 screw terminal board for RS485
3. Universal power supply
4. Software CD Rom

2.1.2.2 Serial µDAQ Temperature Device

1. µDAQ unit
2. Serial cable for RS232 or a DB9 screw terminal board for RS485
3. Universal power supply
4. Thermocouple or RTD adaptor
5. Software CD Rom

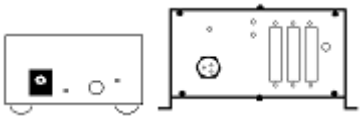
2.1.2.3 Serial Rugged µDAQ Digital I/O and Analog Device

1. Rugged µDAQ unit
2. Serial cable for RS232 or a DB9 screw terminal board for RS485
3. Industrial power cable
4. Rubber vibration mounting kit
5. DINRAIL mounting kit
6. Software CD Rom

2.1.2.4 Serial Rugged µDAQ Temperature Device

1. Rugged µDAQ unit
2. Serial cable for RS232 or a DB9 screw terminal board for RS485
3. Industrial power cable
4. Rubber vibration mounting kit
5. DINRAIL mounting kit
6. Thermocouple or RTD adaptor
7. Software CD Rom

2.1.3 Wireless Devices



2.1.3.1 Wireless µDAQ Digital I/O and Analog Device

1. µDAQ unit
2. 2.4 GHz antenna
3. Bluetooth USB dongle
4. Universal power supply
5. Software CD Rom

2.1.3.2 Wireless µDAQ Temperature Device

1. µDAQ unit
2. 2.4 GHz antenna
3. Bluetooth USB dongle
4. Universal power supply
5. Thermocouple or RTD adaptor
6. Software CD Rom

2.1.3.3 Wireless Rugged µDAQ Digital I/O and Analog Device

1. Rugged µDAQ unit
2. 2.4 GHz antenna
3. Bluetooth USB dongle
4. Industrial power cable
5. Rubber vibration mounting kit
6. DINRAIL mounting kit
7. Software CD Rom

2.1.3.4 Wireless Rugged µDAQ Temperature Device

1. Rugged µDAQ unit
2. 2.4 GHz antenna
3. Bluetooth USB dongle
4. Industrial power cable
5. Rubber vibration mounting kit
6. DINRAIL mounting kit
7. Thermocouple or RTD adaptor
8. Software CD Rom

2.2 Operating Systems for Specific Devices

The µDAQ and Rugged µDAQ series support various bus/communication interfaces. Each interface type will be listed separately.

2.2.1 USB - Universal Serial Bus Type Devices

The USB driver for Windows is a Windows Driver Model (WDM) type that will run on all modern Windows platforms. Linux has it's own driver which is exported as a character device.

OS Type	Driver Type	Devices
Windows ME/2000/XP	WDM Plug and Play	USB 24A/C USB 48A/C USB 96A/C USB 120A R-USB 24A/C R-USB 48A/C R-USB 96A/C R-USB 120A/C R-USB 144 A/C R-USB 168A USB 30/26 USB 73R/T R-USB 30/26 R-USB 73R/T
Pocket PC 2003	Not Supported	None
Linux 2.4 and later	Linux Character Device	USB 24A/C USB 48A/C USB 96A/C USB 120A R-USB 24A/C R-USB 48A/C R-USB 96A/C R-USB 120A/C R-USB 144 A/C R-USB 168A USB 30/26 USB 73R/T R-USB 30/26 R-USB 73R/T

Table 2-1 USB Operating System Support

2.2.2 SRL/BT - Serial Communication Protocol and Wireless Driver

The serial and wireless communications driver for Windows is embedded into the EDR Enhanced API. Currently there is no support for Linux yet. This should be available soon by the end of 2004.

OS Type	Driver Type	Devices
Windows 98/ME/2000/XP	Dynamic Link Library	SRL 24A/C SRL 48A/C SRL 96A/C SRL 120A R-SRL 24A/C R-SRL 48A/C R-SRL 96A/C R-SRL 120A/C R-SRL 144 A/C R-SRL 168A SRL 30/26 SRL 73R/T R-SRL 20/26 R-SRL 73R/T
Windows 2000/XP	Dynamic Link Library	BT 24A/C BT 48A/C BT 96A/C BT 120A R-BT 24A/C R-BT 48A/C R-BT 96A/C R-BT 120A/C R-BT 144 A/C R-BT 168A BT 30/26 BT

		73R/T R-BT 20/26 R-BT 73R/T
Pocket PC 2003	Embedded Dynamic Link Library	SRL 24A/C SRL 48A/C SRL 96A/C SRL 120A R-SRL 24A/C R-SRL 48A/C R-SRL 96A/C R-SRL 120A/C R-SRL 144 A/C R-SRL 168A SRL 30/26 SRL 73R/T R-SRL 20/26 R-SRL 73R/T
Pocket PC 2003	Embedded Dynamic Link Library	BT 24A/C BT 48A/C BT 96A/C BT 120A R-BT 24A/C R-BT 48A/C R-BT 96A/C R-BT 120A/C R-BT 144 A/C R-BT 168A BT 30/26 BT 73R/T R-BT 20/26 R-BT 73R/T
Linux 2.4 and later	Not available	None

Table 2-2 Serial and Wireless Operating System Support

2.3 Installation

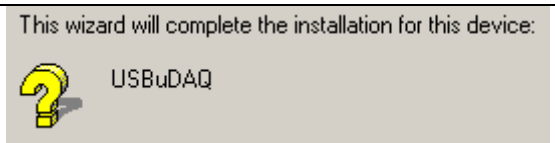
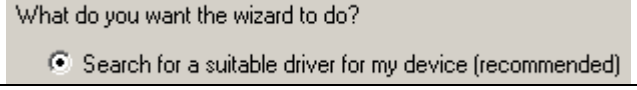
The installation section will discuss the installation procedure for each communication type individually. Each communication type has its own way to be installed.

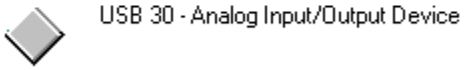
2.3.1 USB Installation

The USB installation is different on each operating system type. The installation procedure for each operating system will be discussed separately for Windows Desktop, Windows Mobile and Linux.

2.3.1.1 Microsoft Windows PnP Installation (Windows ME, 2000, XP)

For the Windows PnP installation you will need a PC that can accept a USB device and that is configured to work with USB devices. Depending on the version you will need either USB 1.1 or USB 2.0.

1		Find an open USB port and connect your device with the provide USB cable. Also provide power to your device if it is externally powered. Only use the provide power cable or power supply.
2		Windows will now detect that a new USB device was attached and request driver to be supplied.
3		Click on the ***next*** button to start the process.
4		The wizard will now need to be supplied with driver files to for your USB device.
5		Select the option as indicated and then the next button.
6		Only select to specification the location of the driver files.
7		The driver is located at ***<EAGLECD>\EDRE\Drivers\WDM\USB***. Select the next button to install the driver.

8		Depending on the model the last screen will show that the installation was completed successfully for your particular device.

2.3.1.2 Linux Installation (Kernel 2.4 and later) - Overview

The Linux installation can be a bit tricky sometimes. Make sure that the kernel supports USB devices and that the necessary modules are already loaded or can load on demand. Also make sure that all kernel source and header files are installed. The Linux driver is available with complete source and it will be best if it is recompiled to be compatible with the current kernel version. The source listing on the <EAGLECD>\EDRE\Linux should match the directory structure on the target system. Drivers can be copied manually to */usr/src/<linux kernel source>/drivers/edredaq*. The USB driver will also be located here. Copy the driver and read the documentation in the same directory to install the driver.

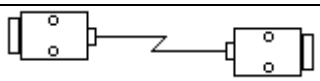
Also copy the EDR Enhanced shared object and examples. This should be located at */usr/src/edre.* The EDRE source directory contains the source for the API and examples. The examples directory has samples program specifically for each device. Make sure to install the EDR Enhanced before trying to build the source code. The header files are located at *usr/include/edre.* The header files is also necessary when building the drivers. Please consult the Linux documentation for complete instructions.

2.3.2 Serial Device Device Installation

The serial devices require a PC with a compatible serial port, either RS232 or RS485 depending on the µDAQ or Rugged µDAQ version. The pin connections are listed in the hardware interface chapter.

2.3.2.1 Microsoft Windows Installation (Windows 98/ME/2000/XP)

The serial and wireless communications driver is embedded into the EDR Enhanced API. To be able to communicate to your device you have to add it as a serial server.

1		Attach you serial cable to your device and computer. Apply power to the unit.
2		Install the EDR Enhanced API by running <EAGLCD>\EDRE\API\EDREAPI.EXE. This will install all required operating specific interface software.
3		Open the ***EDR Enhanced Setup*** in the Windows Control Panel Start>Settings>Control Panel>EDR Enhanced Setup
4		Select the ***Serial*** Tab
5		Click on the ***Add*** button
6		Select the correct ***Comm Port***, If it is a wireless device set the ***BAUD Rate*** to 115200, Set ***Connection Type*** to permanent and Select the ***Auto*** button. This will automatically detect the serial number of the device. If this fails enter the serial number

<table>
<tr><td></td><td></td><td>manually.</td></tr>
<tr><td>7</td><td> 
</td><td>Select the OK button.</td></tr>
<tr><td>8</td><td> 
</td><td>Select the Host Setup button.
Confirm the following values
<table>
<tr><td>Host ID</td><td>1000</td></tr>
<tr><td>Read Interval Timeout</td><td>100</td></tr>
<tr><td>Read Total Timeout Multiplier</td><td>1000</td></tr>
<tr><td>Read Total Timeout Constant</td><td>500</td></tr>
<tr><td>Write Total Timeout Multiplier</td><td>100</td></tr>
<tr><td>Write Total Timeout Constant</td><td>500</td></tr>
</table></td></tr>
<tr><td>9</td><td> 
</td><td>Select the OK button to close and apply the new settings.</td></tr>
<tr><td>10</td><td> 

EDR Enhanced Setup</td><td>To verify the installation reopen EDR Enhanced Setup and select Devices Tab.</td></tr>
<tr><td>11</td><td>  </td><td>Select the device serial number to display information about the newly installed serial device.
If the device name is SRL Device Unavailable, it means the device was no detected properly. Go to the troubleshooting section to resolve this problem or try entering the serial number and BAUD rate manually.</td></tr>
</table>

2.3.2.2 Microsoft Pocket PC 2003 Installation

Information was not available at time of publication.

2.3.2.3 Linux Installation

Not available on Linux platform.

2.3.3 Wireless Installation

2.3.3.1 Microsoft Windows (Windows 2000/XP)

The wireless devices use Bluetooth® technology for communications. The wireless devices work in the same manner as serial. After the Bluetooth® installation has been completed use the serial installation section to complete the device setup.

Follow the instructions below to prepare your wireless device for serial installation.

<table>
<tr><td>1</td><td> 
</td><td>Install Bluetooth OEM software found at <EAGLECD>\OEM\Bluetooth Dongle\setup.exe.
During this step you will be required to install the USB Bluetooth Dongle.</td></tr>
<tr><td>2</td><td> 
My Bluetooth Places</td><td>Do initial Bluetooth setup by selecting the My Bluetooth Places on the Windows Desktop. Skip any specific device setup. Install only serial port services.</td></tr>
</table>

3	 Find Bluetooth Devices	Right click on ***Find Bluetooth Devices*** and select properties. Select the ***Client Applications*** Tab. Select the ***Bluetooth Serial Port*** properties. Deselect ***Secure Connection***. Select Ok to end.
4	 Find Bluetooth Devices	Open ***Find Bluetooth Devices***.
5	 BSC110-DCE	Select **BSC110-DCE**
6	 Serial Port on BSC110-DCE	Select **Serial Port on BSC110-DCE**
7	 Serial Port on BSC110-DCE	A dialog will pop-up with the assigned COM port. The icon will turn green if it connected to the device. Select to return to previous folder.
8	 BSC110-DCE	The device will now be ready for use. Use the serial setup section to configure your device for EDR Enhanced. Make sure to set the BAUD rate to 115200 to Auto detect the device properly.

2.3.3.2 Microsoft Pocket PC 2003

Information was not available at time of publication..

2.3.3.3 Linux Installation

Not available on Linux platform.

2.4 Application Software

The EDR Enhanced Software Development Kit CD-Rom comes with WaveView for Windows™. WaveView has support for Analog Inputs, Analog Outputs, Digital I/O and Counter-Timers. It has an oscilloscope function to continuously display incoming analog data, a signal generator, a power supply, temperature logger and a multifunctional chart recorder to sample and control signals, analog and digital, at preset intervals.

WaveView can be found on the EDR Enhanced CD-Rom at *<EAGLECD>\EDRE\APPS\WVFW*.

3

3 Hardware Interface

The Hardware Interface chapter will discuss all connectors located on the µDAQ and Rugged µDAQ products. It will list all the different external power supplies, internal power supplies and batter packs with onboard chargers. The pin assignments for each connector will also be listed.

The µDAQ & Rugged µDAQ series has connectors for digital I/O, counter-timers and analog I/O. The µDAQs make use of only one connector type, a DB25 male. To inter-connect to application modules there are adapters modules available. A cable is used to connect to these modules. Screw terminal modules are also available for quick installations. This chapter will also discuss method of hardware operation, optional accessories and connectable application modules.

3.1 µDAQ External Application Connectors

The µDAQ is fitted with various DB25 male connectors. The illustrations below show the different box types. There are two-tier and three-tier models. Depending on the number of connectors required the box would be either one. The USB µDAQ devices use a standard USB type B connector and serial devices a DB9 male. The DC power connector is not always used because limited power can be derived from the USB bus. In the case of the serial and wireless devices there is always a power supply connector present. The side with the power connector is the rear side of the device. The pin assignments are the same for all models.

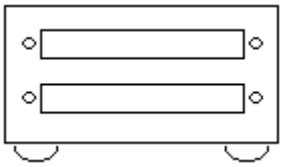


Figure 3-1 µDAQ Front Side 2-Tier

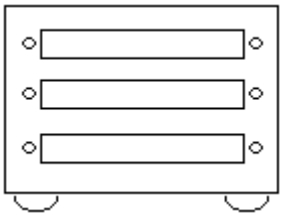


Figure 3-2 µDAQ Front Side 3-Tier

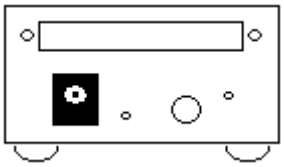


Figure 3-3 USB µDAQ Rear Side 2-Tier

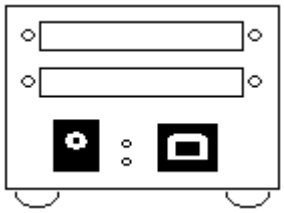


Figure 3-4 USB µDAQ Rear Side 3-Tier

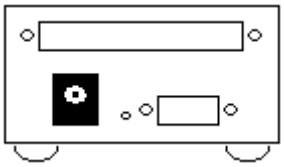


Figure 3-5 Serial µDAQ Rear Side 2-Tier

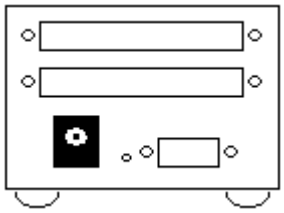


Figure 3-6 Serial µDAQ Rear Side 3-Tier

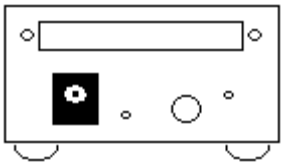


Figure 3-7 Wireless µDAQ Rear Side 2-Tier

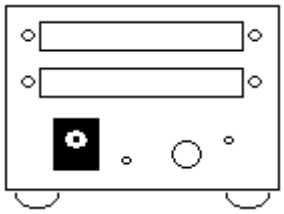


Figure 3-8 Wireless µDAQ Rear Side 3-Tier

XXX can be USB, SRL or BT
B = Bottom, M = Middle, T = Top
F = Front, R = Rear

Device Type	DB25 (M) Digital I/O Port Assignment	DB25(M) Counter Port Assignment	DB25(M) Analog I/O	Box Type
XXX 24A	FB (0-2)			
XXX 24C	FB (0-2)	RT (0-5)		
XXX 48A	FB (0-2) FT (3-5)			
XXX 48C	FB (0-2) FT (3-5)	RT (0-5)		
XXX 72A	FB (0-2) FT (3-5) RT (6-8)			
XXX 72C	FB (0-2) FT (3-5) RT (6-8)	RT (0-5)		

XXX 96A	FB (0-2) FM (3-5) RM (6-8) FT (9-11)			
XXX 96C	FB (0-2) FM (3-5) RM (6-8) FT (9-11)	RT (0-5)		
XXX 120A	FB (0-2) FM (3-5) RM (6-8) FT (9-11) RT (12-15)			
XXX 26A16	FB (0-2)		FT (A/D 0-15)	
XXX 30A16	FB (0-2)		FT (A/D 0-15, D/A 0-3)	
XXX 26B16	FB (0-2)		FT (A/D 0-15)	
XXX 30B16	FB (0-2)		FT (A/D 0-15, D/A 0-3)	
XXX 26C16	FB (0-2)		FT (A/D 0-15)	
XXX 30C16	FB (0-2)		FT (A/D 0-15, D/A 0-3)	
XXX 26C32	FB (0-2)		FM (A/D 0-15) FT (A/D 16-31)	
XXX 30C32	FB (0-2)		FM (A/D 0-15, D/A 0-3) FT (A/D 16-31)	
XXX 73T/R8	FB (0-2)		FT (CH0-7, CJC0)	
XXX 73T/R16	FB (0-2)		FT (CH0-7, CJC0) RT (CH8-15, CJC1)	
XXX 73T/R32	FB (0-2)		FM (CH0-7, CJC0) RM (CH8-15, CJC1) FT (CH16-23, CJC2) RT (CH24-31, CJC3)	

Table 3-1 µDAQ Connectors

3.2 Rugged µDAQ Application Connectors

Rugged DAQ unit's connectors are not pre-assigned; therefore there exist a predefined order in which the connectors are assigned to a type of functionality. The connector names are prefixed with an **F** and **R**, which is short for **front** and **rear**. The power connector is located on the rear side of the Rugged µDAQ device. The connectors are numbered from zero onwards.

The device is populated from connector zero to three, where zero is the motherboard and the other daughterboards. Board zero always has 24 digital I/O lines located at connector **F0**. The rest are populated in the order as below if installed.

1) Digital I/O. This can be located front and rear.
2) Digital I/O with counter-timers. The digital I/O is located at the front (F1, F2 etc.) and the counter-timers will be found on the rear (R1, R2 etc.) connector.
3) Analog I/O is the next device in the stack if installed. The analog device has only one connector and will always be installed in front (F1, F2, etc).
4) The last interface is the temperature, either RTD or thermocouple. There will be 8 channels located on a connector, found both in the front and on the rear side.

3.2.1 Rugged µDAQ Connector Locations

The figures below show the location of the front and rear connectors located on the Rugged µDAQ extrusion. The connectors are labeled F0-3 and R1-3. The table below shows the location of each type of connector for a particular type of device.

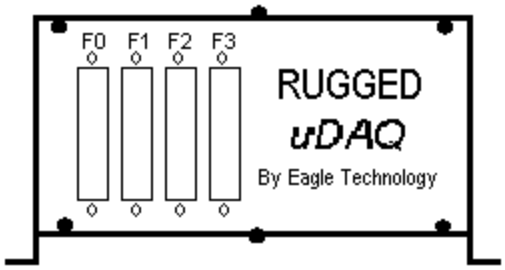


Figure 3-9 Rugged µDAQ Front Panel

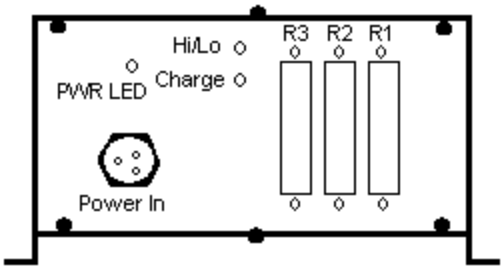


Figure 3-10 Rugged µDAQ Rear Panel

XXX can be either USB, SRL or BT

Device Type	DB25 (M) Digital I/O Port Assignment	DB25(M) Counter Assignment	DB25(M) Analog I/O Assignment	DB25(M) Temperature I/O Assignment
R-XXX 24A	F0 (0,1,2)			
R-XXX 48C	F0 (0,1,2) F1 (3,4,5)	R1 (0-5)		
R-XXX 72A	F0 (0,1,2) F1 (3,4,5) R1 (6,7,8)			
R-XXX 96C	F0 (0,1,2) F1 (3,4,5) R1 (6,7,8) F2 (9,10,11)	R2 (0-5)		
R-XXX 120A	F0 (0,1,2) F1 (3,4,5) R1 (6,7,8) F2 (9,10,11) R2 (12,13,14)			
R-XXX 144C	F0 (0,1,2) F1 (3,4,5) R1 (6,7,8) F2 (9,10,11) R2 (12,13,14) F3 (15,16,17)	R3 (0-5)		
R-XXX 168A	F0 (0,1,2) F1 (3,4,5) R1 (6,7,8) F2 (9,10,11) R2 (12,13,14) F3 (15,16,17) R3 (18,19,20)			
R-XXX 26A/B/C16	F0 (0,1,2)		F1 (A/D 0-15)	
R-XXX 30A/B/C16	F0 (0,1,2)		F1 (A/D 0-15, D/A 0-3)	
R-XXX 26A/B/C32	F0 (0,1,2)		F1 (A/D 0-15) F2 (A/D 16-31)	
R-XXX 30A/B32	F0 (0,1,2)		F1 (A/D 0-15, D/A 0-3) F2 (A/D 16-31, D/A 4-7)	
R-XXX 30C32	F0 (0,1,2)		F1 (A/D 0-15, D/A 0-3) F2 (A/D 16-31)	
R-XXX 73T/R16	F0 (0,1,2)			F1 (CH0-7, CJC 0) R1 (CH8-15, CJC 1)
R-XXX 73T/R32	F0 (0,1,2)			F1 (CH0-7, CJC 0) R1 (CH8-15, CJC 1) F2 (CH16-23, CJC 2) R2 (CH24-31, CJC 3)

Table 3-2 Rugged µDAQ Connector Assignment

3.3 Pin Assignments

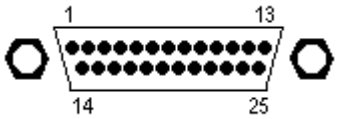


Figure 3-11 µDAQ and Rugged µDAQ DB25 Male Connector

3.3.1 µDAQ & Rugged µDAQ DIO Connector – DB25 (M)

The table below shows the pin assignments for the DB25(M) digital I/O connectors found on the µDAQ and Rugged µDAQ devices.

Pin	Name	Pin	Name
1	PA0	14	PA1
2	PA2	15	PA3
3	PA4	16	PA5
4	PA6	17	PA7
5	PB0	18	PB1
6	PB2	19	PB3
7	PB4	20	PB5
8	PB6	21	PB7
9	PC0	22	PC1
10	PC2	23	PC3
11	PC4	24	PC5
12	PC6	25	PC7
13	DGND		

Table 3-3 µDAQ & Rugged µDAQ DIO Connector – DB25 (M)

3.3.2 µDAQ & Rugged µDAQ Opto-isolated Connector – DB25 (M)

The table below shows the pin assignments for the DB25(M) Opto-isolated input connectors found on the µDAQ and Rugged µDAQ devices.

Pin	Name	Pin	Name
1	Opto 0+	14	Opto 0-
2	Opto 1+	15	Opto 1-
3	Opto 2+	16	Opto 2-
4	Opto 3+	17	Opto 3-
5	Opto 4+	18	Opto 4-
6	Opto 5+	19	Opto 5-
7	Opto 6+	20	Opto 6-
8	Opto 7+	21	Opto 7-
9	NC	22	COM -
10	NC	23	NC
11	NC	24	NC
12	NC	25	NC
13	NC		

Table 3-4 µDAQ & Rugged µDAQ Opto-isolated Connector – DB25 (M)

3.3.3 µDAQ & Rugged µDAQ Reed Relay Connector – DB25 (M)

The table below shows the pin assignments for the DB25(M) Reed Relay output connectors found on the µDAQ and Rugged µDAQ devices.

Pin	Name	Pin	Name
1	Relay 0	14	Relay 0 RET
2	Relay 1	15	Relay 1 RET
3	Relay 2	16	Relay 2 RET
4	Relay 3	17	Relay 3 RET
5	Relay 4	18	Relay 4 RET
6	Relay 5	19	Relay 5 RET
7	Relay 6	20	Relay 6 RET
8	Relay 7	21	Relay 7 RET
9	NC	22	NC
10	NC	23	NC
11	NC	24	NC
12	NC	25	NC
13	NC		

Table 3-5 µDAQ & Rugged µDAQ Reed Relay Connector – DB25 (M)

3.3.4 µDAQ & Rugged µDAQ Counter-Timer Connector – DB25 (M)

The table below shows the pin assignments for the DB25(M) counter-timer connector found on the µDAQ and Rugged µDAQ devices.

Pin	Name	Pin	Name
1	NC	14	NC
2	NC	15	OUT5
3	GATE_EXT5	16	CLK_EXT5
4	CLK_EXT0	17	OUT4
5	GATE_EXT0	18	OUT0
6	OUT2	19	CLK_EXT2
7	CLK_EXT1	20	GATE_EXT2
8	OUT1	21	GATE_EXT1
9	DGND	22	GATE_EXT4
10	NC	23	CLK_EXT4
11	NC	24	OUT3
12	NC	25	GATE_EXT3
13	CLK_EXT3		

Table 3-6 µDAQ CT Connector – DB25 (M)

3.3.5 µDAQ & Rugged µDAQ Analog I/O – DB25 (M)

The table below shows the pin assignments for the DB25(M) analog I/O connectors found on the µDAQ and Rugged µDAQ Analog I/O.

Pin	Name	Pin	Name
1	ACH0	14	ACH1
2	ACH2	15	ACH3
3	ACH4	16	ACH5
4	ACH6	17	ACH7
5	ACH8	18	ACH9
6	ACH10	19	ACH11
7	ACH12	20	ACH13
8	ACH14	21	ACH15
9	AGND	22	DAC0
10	DAC1	23	DAC2
11	DAC3	24	10V_REFCAL
12	NC	25	EXT_GATE
13	EXT_CLK	26	SHELL – DGND

Table 3-7 µDAQ and Rugged µDAQ Analog I/O Connector – DB25 (M)

3.3.6 µDAQ & Rugged µDAQ Analog I/O – DB25 (M)

The table below shows the pin assignment for the DB25(M) analog inputs 16 – 31. The connector is fount on the 32-channel analog C-version µDAQ and Rugged µDAQ devices.

Pin	Name	Pin	Name
1	ACH16	14	ACH17
2	ACH18	15	ACH19
3	ACH20	16	ACH21
4	ACH22	17	ACH23
5	ACH24	18	ACH25
6	ACH26	19	ACH27
7	ACH28	20	ACH29
8	ACH30	21	ACH31
9	AGND	22	NC
10	NC	23	NC
11	NC	24	NC
12	NC	25	NC
13	NC	26	SHELL – DGND

Table 3-8 µDAQ and Rugged µDAQ Analog Inputs 16-31 Connector – DB25 (M)

3.3.7 µDAQ & Rugged µDAQ Temperature Input – DB25 (M)

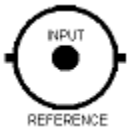
The table below shows the pin assignments for the DB25(M) temperature input connectors found on the USB µDAQ Temperature device.

Pin	Name	Pin	Name
1	AGND	14	+8.4V
2	AGND	15	+12V
3	AGND	16	CJC
4	AGND	17	-12V
5	AGND	18	TCH7-
6	TCH7+	19	TCH6-
7	TCH6+	20	TCH5-
8	TCH5+	21	TCH4-
9	TCH4+	22	TCH3-
10	TCH3+	23	TCH2-
11	TCH2+	24	TCH1-
12	TCH1+	25	TCH0-
13	TCH0+	26	SHELL – DGND

Table 3-9 USB µDAQ Temperature Input – DB25 (M)

3.3.8 µDAQ Analog I/O BNC – 8 x BNC Connectors

The BNC connectors are paired in differential input form. Each connector will contain the differential channel (on the middle pin) and the reference on the outside. The unit can also be used in single-ended mode. In this case each connector will have two inputs.

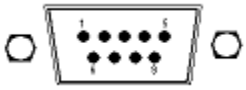


BNC Inside Pin	Channel	Pair Outside Shell	Channel Reference
1+	ACH0	1-	ACH8
2+	ACH1	2-	ACH9
3+	ACH2	3-	ACH10
4+	ACH3	4-	ACH11
5+	ACH4	5-	ACH12
6+	ACH5	6-	ACH13
7+	ACH6	7-	ACH14
8+	ACH7	8-	ACH15

Table 3-10 µDAQ BNC Connectors

3.3.9 µDAQ Analog I/O BNC – DB9 (F)

The analog output connector of the BNC unit can be found on the front. The pin assignments are below.



Pin	Channel	Pin	
1	DAC0	6	DAC1
2	DAC2	7	DAC3
3	10V_REFCAL	8	AGND
4	DGND	9	EXT_TRIG
5	EXT_CLK	Shell	NC

Table 3-10 DAQ BNC Analog Output

3.3.10 Signal Definitions

This sections deal with all the signals abbreviations.

Signal	Description
PA0-7	8255 PPI Port A
PB0-7	8255 PPI Port B
PC0-7	8255 PPI Port C
GATE_EXT0-5	Counter External Gate
CLK_EXT0-5	Counter External Clock
OUT0-5	Counter Output
DGND	Digital ground.
AGND	Analog Ground
ACH0-31	Analog Input Channels 0 –15
TCH (0-)-(7-)	Thermo couple negative input
TCH (0+)-(7+)	Thermo couple positive input
DAC0-3	Analog Outputs Channels 0 – 3
EXT_CLK	External Clock/Convert
EXT_GATE	External Gate
10V_REFCAL	10 Volt Reference for Calibration
NC	Not Connected

Table 3-12 Signal definitions

3.4 Pin Descriptions

3.4.1 Digital Inputs/Outputs (PA0-7, PB0-7, PC0-7)

These lines are connected to the 3 ports of the 8255 PPI. Each port can be configured as either an input or an output.

3.4.2 Opto-isolated Inputs positive (Opto 0 - 7+)

Optically isolated positive input. (0 – 28V)

3.4.3 Opto-isolated Inputs negative (Opto 0 - 7-)

Optically isolated negative input.

3.4.4 Opto-isolated (COM-)

During manufacturing some or all of the negative opto lines can be connected together. This COM- pin can then be used for these optically isolated input negative lines to cut down on wiring.

3.4.5 Reed Relay (Relay 0 – 7)

3.4.6 Reed Relay return (Relay RET 0-7)

Normally open reed relay contacts.

3.4.7 External Gate (GATE_EXT)

These lines are used to externally control the gate of the counters.

3.4.8 External Clock (CLK_EXT)

These lines are used to externally clock the counters.

3.4.9 Output (OUT0-5)

These are the outputs of each counter-timer.

3.4.10 Digital Ground (DGND)

All digital ground signals should be connected to this pin.

3.4.11 Analog Ground (AGND)

All analog inputs should be referenced to AGND. Do not connect AGND and DGND together. This will create ground loops and instability in the hardware.

3.4.12 Analog Inputs (ACH0-15)

The analog input channels are connected to the analog input sub-system and are used to measure analog voltages. These signals are referenced to analog ground (AGND).

3.4.13 Analog Outputs (DAC0-3)

The analog output channels are used to output analog voltages. They are referenced to analog ground (AGND).

3.4.14 Temperature Inputs (TCH0-7)+-

Each temperature channel has two input lines for positive and negative. However in all temperature applications there are always and external modules for thermocouple or RTD inputs. Their user never connects directly to these pins.

3.4.15 External Clock/Convert (EXT_CLK)

This pin is the external clock/convert input. It is used to control the convert timing of the analog to digital converter (ADC). This signal is synchronized with a master clock of 20MHz. This signal must be referenced to digital ground (DGND), which is the connector shell.

3.4.16 External Trigger (EXT_GATE)

This signal is used as a control gate for the analog input scanning process. When selected by software and set high it will enable the process. A low voltage will disable the process.

3.4.17 10 Volt Reference Calibration (10V_REFCAL)

This pin is used to measure the 10-volt reference for the analog circuit. It is only used during calibration and should not be used externally. If used it can affect the performance of the analog I/O.

Do not connect to the 10Volt Reference Pin. It's only used for calibration purposes.

3.5 Bus Connectors

The µDAQ and Rugged µDAQ use commercial and industrial type bus connectors. The commercial type connector is a normal plug-in type where the industrial must be screwed on.

This is to stop connectors from being pulled off accidentally. Only a connector specific to a bus topology is used as a bus interface connector.

3.5.1 µDAQ USB Interface Connector

The µDAQ USB uses a standard USB Type B connector. The device is supplied with a USB 2.0 compliant cable. When viewing the USB µDAQ device from the rear the USB B Type connector is located on the right side. In the middle are the indicator light and the power connector on the left. When the USB device is connected to the PC the USB indicator light will light up, indicating that USB bus power is present.

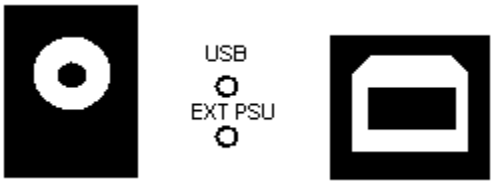


Figure 3-12 USB µDAQ Connectors

3.5.2 Rugged µDAQ USB Interface Connector

The Rugged µDAQ has a moister and splash proof type USB connector. This has rubber seals to prevent unwanted moister to enter the unit. The connector is located just to the right of the power input connector on the rear side of the Rugged µDAQ.

Figure 3-13 Rugged µDAQ USB Connector

3.5.3 µDAQ Serial Interface Connector

The µDAQ serial device uses a standard DB9 male connector for both RS232 and RS485. For RS232 a normal serial crossover cable can be used. The unit is supplied with a serial cable. The RS485 model is supplied with an ADPT-9 adaptor. The table below shows the pin assignments for the different serial interfaces.

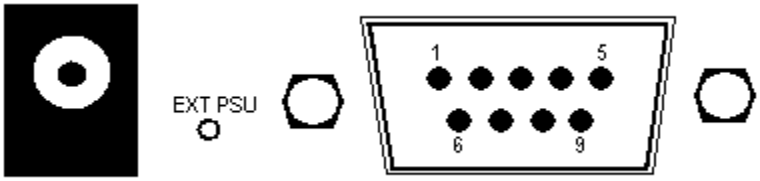


Figure 3-14 Serial DAQ Connectors

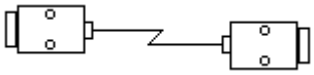


Figure 3-15 RS 232 9-way cable

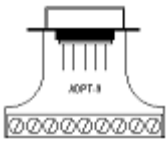


Figure 3-16 RS 485 - ADPT-9

3.6 Power Supplies, Power Connectors and Battery Chargers

The µDAQ and Rugged µDAQ support a wide variety of power supplies. Depending on the model there can be an external power supply, internal power supply, DC or AC input. The µDAQ always has an external power supply and the Rugged DAQ has and internal power supply. Special Rugged µDAQ models can be build which uses and external power supply. The Rugged µDAQ model can also contain a battery pack and build-in charging circuit. These units can operate without battery for some times. The battery operation time depends on the model type.

3.6.1 µDAQ Power Supply Units (PSU)

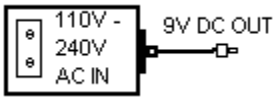


Figure 3-17 DAQ AC PSU

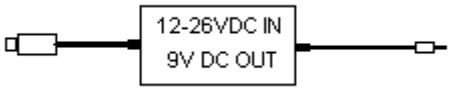


Figure 3-18 DAQ DC In Car PSU

The commercial µDAQ units are supplied with two types of power supplies; an AC wide range input power supply and a mobile DC power supply. The output voltage of both power supplies is 9V DC, which is required by the µDAQ unit. The polarity of the output connector must be positive on the outside and negative in the middle.

Figure 3-19 µDAQ Power Connector Pin Assignment

The USB, serial and wireless units have the same power connector. There is also an indicator light to show of the power is present. Please note that there are some USB models that are bus powered and don't ship with power supplies. Although the power socket is present it is disconnected from the internal circuits.

Figure 3-20 µDAQ Power Socket

The AC power supply is a wide AC input voltage range type, 110V to 250V. The AC power supply also contains a universal socket kit, which makes it possible to connect to any international power socket. The output voltage is 9V @ 1A maximum.

The DC power supply is an in car type. The power connecter plugs into a normal car cigarette lighter socket. The input voltage range is 12V to 26V DC. This makes it ideal for use within 12V or 24V vehicles. The output is 9V @ 2.3A.

3.6.2 Rugged µDAQ Internal Power Supplies

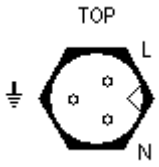


Table 3-11 Rugged µDAQ AC PWR Connector

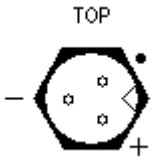


Table 3-12 Rugged µDAQ DC PWR Connector

The Rugged µDAQ has two types of internal power supplies, an AC wide range input type and a DC input type. The part number of the Rugged µDAQ is either appended with an **AC** or **DC** to indicate the power interface type.

The AC type is a wide input voltage range switch mode power supply. The specification is 90-264V AC, 47 to 63 Hz @ 60W maximum.

The DC type is a wide voltage range regulated power supply designed for industrial applications. The input range is 18 to 40V DC @ 5A maximum.

The Rugged µDAQ has a green power present indicator. The illustration below shows the power indicator. It's located above the power connector. The indicator will be illuminated when the power is connected and present.

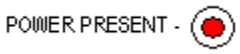


Figure 3-21 Rugged µDAQ Power Indicator

3.6.3 Rugged µDAQ Battery Pack and Charger

The Rugged µDAQ has an optional nickel metal hydride (NiMH) battery pack and charger. The battery pack serves as an uninterruptible power supply (UPS). The part number of the Rugged DAQ will have the name **BAT** appended. This will indicate that the unit has the battery pack and charger installed. The capacity of the battery pack is 3700mAh. The indicators, as illustrated below, found on the rear of the unit indicate the charge status.

Please note that the battery pack needs to be charged for sixteen (16) hours before being used for the first time. The battery pack must also be recharged after long storage or when used in the case of a power interruption. The purpose of the battery pack is to provide power to the Rugged µDAQ unit during a power failure. However the battery pack can be used as a power pack, but the operation time depends on the unit type.

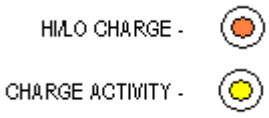


Figure 3-22 Rugged µDAQ Battery Charger Indicators

3.6.3.1 Charger Operation – High & Low Charge

The amber **HI/LO** indicator will be illuminated when the charger is in fast charge mode. This is a fast charge mode charging the battery pack as quickly as possible. When the indicator is off the charger is in a trickle charge mode, meaning the battery pack is fully charged and ready to use.

3.6.3.2 Charger Operation – Charge Activity

The yellow **charge activity** indicator shows the status of the current charge mode (HI or LO). When flickering fast the charger is more active for the current mode. The battery pack is fully empty when the **HI/LO** indicator is **on** and the charge activity is flickering fast. If the **HI/LO** indicator is **off** and the **charge activity** is flickering slowly the battery pack is at its fullest.

3.6.3.3 Battery Pack Operation Time

The operation time of the battery pack differ drastically from a unit type to another. The table below shows an estimate of how long the battery pack should last if fully charged. The battery life also depends on the number output digital I/O connections. All times are listed as typical maximum and typical minimum.

XXX can be USB, SRL(RS232,RS485) or BT

Device Type	Battery Operation Time (Hours) Minimum - Maximum
Digital I/O Models	
R-XXX-24A-AC/DC-BAT	20-25
R-XXX-48C-AC/DC-BAT	9-12
R-XXX-72A-AC/DC-BAT	16-25
R-XXX-96C-AC/DC-BAT	7-11
R-XXX-120A-AC/DC-BAT	11-25
R-XXX-144C-AC/DC-BAT	6-9
R-XXX-168A-AC/DC-BAT	10-25
Analog I/O Models	
R-XXX-26A16-AC/DC-BAT	4.9-6.0
R-XXX-30A16-AC/DC-BAT	4.5-5.5
R-XXX-26A32-AC/DC-BAT	2.8-3.1
R-XXX-30A32-AC/DC-BAT	2.55-3.0
Temperature Models	
R-XXX-73R/T16-AC/DC-BAT	4.9-6.0
R-XXX-73R/T32-AC/DC-BAT	2.8-3.1
Hybrid Models	
R-XXX-30A16-73R/T16-AC/DC-BAT	2.55-3.0
R-XXX-30A32-73R/T16-AC/DC-BAT	1.80-2.0
R-XXX-30A16-73R/T32-AC/DC-BAT	1.80-2.0

Table 3-13 Rugged μDAQ Estimate Battery Life

3.7 Application Modules & Accessories

The μDAQ and Rugged μDAQ devices support a wide range of standard applications modules. These application modules can help to simply or easily duplicate installations that can save allot of time. Application modules and accessories come in many forms. It has support for digital output control and digital input monitoring for AC and DC. Analog signal conditioners, analog amplifiers and optical-isolation are also available.

3.7.1 Digital I/O Adapter Module – PC43A2

The digital I/O adapter module is used to map digital I/O ports to a usable form. The PC43A2 adapter module makes it possible to connect to application modules, such as solid relays modules and optical-isolated modules. The PC43A2 output the 3 x 8-bit ports (A,B,C) into a combination of port A and B and three separate ports A,B and C. To connect to the PC43A2 use a standard DB25M/F cable. Port AB is a standard IDC20 connecter and port A,B and C standard IDC10 connecters. The PC43A2 can also be used to provide external power to

application modules. Connect +5V to the external power screw terminal and it will distribute it to all child models through the various connecters. The table below shows the pin mappings for the module. When externally powered the power indicator will light up.

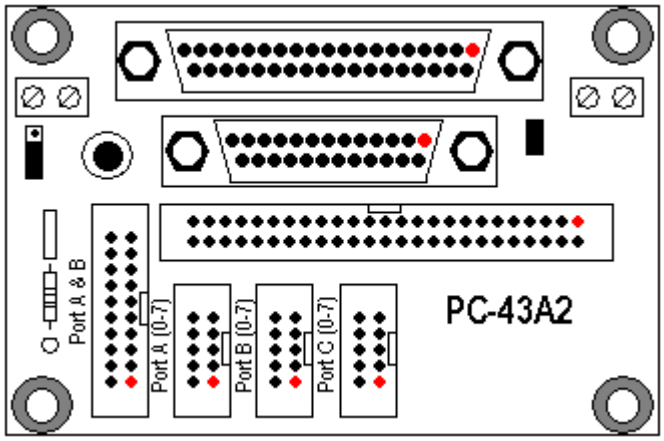


Figure 3-23 PC43A2 Digital I/O Adapter

Pin No Signal Name	DB25 (Incoming) CON3	IDC10 Port A CON7	IDC10 Port B CON8	IDC10 Port C CON9	IDC20 Port AB CON6
PA0	1	1			1
PA2	2	3			3
PA4	3	5			5
PA6	4	7			7
PB0	5		1		9
PB2	6		3		11
PB4	7		5		13
PB6	8		7		15
PC0	9			1	
PC2	10			3	
PC4	11			5	
PC6	12			7	
GND	13	9	9	9	17,18
PA1	14	2			2
PA3	15	4			4
PA5	16	6			6
PA7	17	8			8
PB1	18		2		10
PB3	19		4		12
PB5	20		6		14
PB7	21		8		16
PC1	22			2	
PC3	23			4	
PC5	24			6	
PC7	25			8	
SCREEN	SHELL				
+5V		10	10	10	19,20

Table 3-14 PC43A2 Pin Assignment

3.7.2 Analog I/O Adapter Modules

The PC52A2 analog I/O module maps the analog connector of the µDAQ and Rugged µDAQ into a usable form. The module is used to interconnect to application modules such PC52-4, which is a 5B module carrier board. The output pin assignment is compatible with all other analog application modules. The module connects with a standard DB25M/F cable to the µDAQ and Rugged µDAQ units. The headers (H1, H2) can be used either to terminate a channel or to install a current to voltage converter module. The sixteen input channels maps into either four or eight channel connectors. The analog output channels are available on a standard output connector. The module contains a screw terminal to connect to external trigger, external clock and digital ground. The table below shows the pin mappings of the module.

EC=EXT CLK
DG=DIGITAL GROUND
ET=EXTERNAL TRIGGER

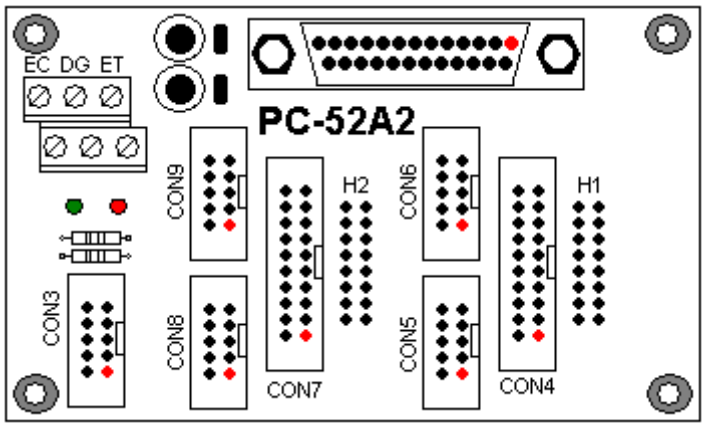


Figure 3-24 PC52A2 Analog I/O module

Pin No Signal Name	DB25 (Incoming) CON3	IDC10 CH0-3 CON5	IDC10 CH4-7 CON6	IDC10 CH8-11 CON8	IDC10 CH12-15 CON9	IDC20 CH0-7 CON4	IDC20 CH8-15 CON7	IDC10 DAC0-3 CON3
CH0	1	3				3		
CH2	2	7				7		
CH4	3		3			11		
CH6	4		7			15		
CH8	5			3			3	
CH10	6			7			7	
CH12	7				3		11	
CH14	8				7		15	
AGND	9	1,2,4,6 8,10	1,2,4,6 8,10	1,2,4,6 8,10	1,2,4,6 8,10	1,2,4,6,8 10,12,14 16,18,19,20	1,2,4,6,8 10,12,14 16,18,19,20	1,2,4,6 8,10
DAC1	10							5
DAC3	11							9
-12V	12							
EXT CLK	13							
CH1	14	5				5		
CH3	15	9				9		
CH5	16		5			13		
CH7	17		9			17		
CH9	18			5			5	
CH11	19			9			9	
CH13	20				5		13	
CH15	21				9		17	
DAC0	22							3
DAC2	23							7
+12V	24							
EXT TRIG	25							
DGND	SHELL							

Table 3-15 PC52A2 Pin Assignment

3.7.3 Thermocouple Adapter

The thermocouple adapter is used to interconnect the µDAQ and Rugged µDAQ unit and thermocouple temperature sensors. The adapter has cold junction compensation (CJC) built-on, which is necessary to make thermocouple connections. The CJC needs to be calibrated before being used. The illustration below shows the thermocouple adapter. The adapter has DB25 female connector that connects directly to the µDAQ or Rugged µDAQ unit. Care must be taken to only connect to a temperature input connector.

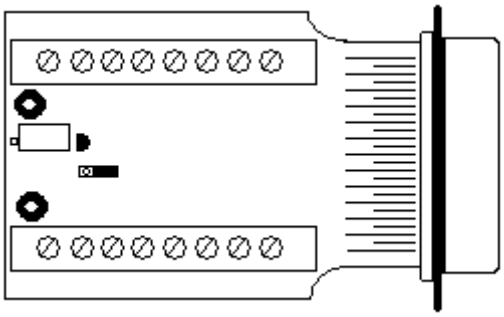


Figure 3-25 Thermocouple Adapter

4

4 Software

The µDAQ and Rugged µDAQ series are supplied with a complete software package called the EDR Enhanced Software Development Kit (EDRE SDK). The SDK is supplied with many operating system drivers, an application program interface (EDRE API), complete documentation and programming examples for most programming languages.

The software development kit also contains WaveView for Windows™ to provide a generic application solution for data capturing and simple process control. Third party Interface drivers are also provided for Labview, TestPoint and Agilent VEE Pro.

The Software chapter serves as reference for the EDR Enhanced API exported functions. It explains how to do common operations to program and control the µDAQ and Rugged µDAQ hardware. Each supported function is listed for each type of interface. The parameters for each function are also listed. There are only a few functions, so make sure to have a thorough understanding of how to use them.

4.1 EDR Enhanced Application Program Interface (EDRE API)

The EDR Enhanced API consist of operating system drivers, Windows dynamic link libraries, Windows ActiveX controls, Windows .Net components and Linux shared objects. EDRE Enhanced API has support for Windows 2000/XP/.Net/CE/Pocket PC2003 and the Linux kernel.

The EDR Enhanced SDK contains example code to use as reference or starting point for a custom application. The examples covers topics such as digital I/O reads and writes, counter-timers programming, analog outputs and reading and scanning the analog inputs. It shows how to build channel lists for scanning analog inputs or doing simple analog reads or writes.

The EDRE API hides the complexity of the hardware and makes it really easy to program the µDAQ and Rugged µDAQ devices. It has functions for each basic sub-system and is real easy to learn.

The EDR Enhanced Application Programming Interface currently support four interface types:

1. Shared Object with exported functions via dynamic link library and shared object
2. EDR Enhanced ActiveX package for Windows platform
3. EDR Enhanced Object interface for Linux and .Net platform
4. EDR Enhanced Device Object interface for Linux and .Net platform

The illustration below shows the EDR Enhanced Application Program Interface design. The next sections will discuss each sub-system's functions separately. It will show which functions each type of interface supports and what parameters are used.

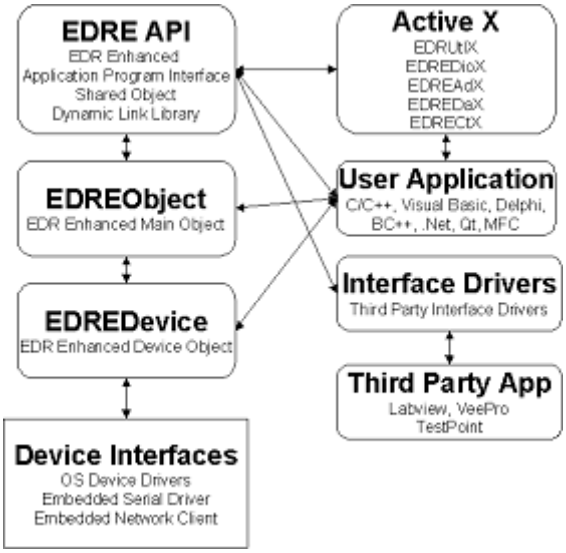


Figure 4-1 EDR Enhanced API Design

4.2 The Query Command – EDRE_Query

The query API function is used to retrieve useful information about your device. Appendix C has a list of query codes that will operate on the µDAQ and Rugged µ DAQ devices. The query function is the most powerful function of the EDR Enhanced API and can tell a lot about your device, like manufacturing date, the serial number, bus type, revision and device driver version. The query function can also supply information about the API itself like release version and number of devices installed.

4.2.1.1 Shared Object and Linked Library Interface

Function Name	EDRE_Query		
Object	Edrapi.dll / edrapi.so		
Platform	Linux, Microsoft Windows, Pocket PC 2003		
Parameters	32-bit unsigned integer 32-bit unsigned integer 32-bit unsigned integer	Serial Number Query Code Parameter	
Return Error Error Codes	32-bit signed integer		
	>=0	Query return value	No error
	-1	EDRE_FAIL	General failure
	-3	EDRE_BAD_SN	Device with serial number does not exist
	-18	EDRE_BAD_QUERY	Invalid query code

4.2.1.2 ActiveX Interface

Function Name	Query		
Object	EDREUTLX		
Platform	Microsoft Windows		
Related Property	SerialNumber		
Parameters	32-bit unsigned integer 32-bit unsigned integer	Query Code Parameter	
Return Value Error Codes	32-bit signed long		
	>=0	Query return value	No error
	-1	EDRE_FAIL	General failure
	-3	EDRE_BAD_SN	Device with serial number does not exist
	-18	EDRE_BAD_QUERY	Invalid query code

4.2.1.3 EDR Enhanced Main Object Interface

Function Name	Query		
Object	EDREObject		
Platform	Linux		
Parameters	32-bit unsigned integer 32-bit unsigned integer 32-bit unsigned integer	Serial Number Query Code Parameter	
Return Value Error Codes	32-bit signed integer		
	>=0	Query return value	No error
	-1	EDRE_FAIL	General failure
	-3	EDRE_BAD_SN	Device with serial number does not exist
	-18	EDRE_BAD_QUERY	Invalid query code

4.2.1.4 EDR Enhanced Device Object Interface

Function Name	Query
Object	EDREDevice
Platform	Linux
Parameters	32-bit unsigned integer — Query Code 32-bit unsigned integer — Parameter
Return Value	32-bit signed integer
Error Codes	>=0 — Query return value — No error -1 — EDRE_FAIL — General failure -3 — EDRE_BAD_SN — Device with serial number does not exist -18 — EDRE_BAD_QUERY — Invalid query code

4.2.1.5 Query example

This example queries the number of devices installed and the serial number of each device.

PSEUDO START

```
Define APINUMDEV = 5   "See appendix C for list of query codes"
NumDevices of type 32-bit integer
SerialNumber of type 32-bit integer
  NumDevices = EDRE_Query(0, APINUMDEV,0)
  For d = 0 to NumDevices-1 do
    SerialNumber = EDRE_Query(0, APIDEVSN,d)
  Next d
```

PSEUDO STOP

4.3 Digital Inputs/Outputs

Depending on the version the µDAQ and Rugged µDAQ device can have up to 168 digital lines. Please refer to your particular version for specific details. The table below shows all the assigned ports to each device. The device can be either µDAQ or Rugged µDAQ. The digital port makes out an important part of any automated system. The digital I/O ports are used to control external equipment logically or to monitor logical conditions. The digital I/O port can be used to switch high voltage external equipment through controlling external relays and contactors. It can also be connected external monitoring circuit that indicates conditions logically.

XXX = USB, SRL, BT or R-USB, R-SRL, R-BT
T=TOP, M=MIDDLE, B=BOTTOM
F=FRONT, R=REAR
2 = 2 TIER BOX, 3 = 3 TIER BOX

Port	PPI No	Assigned Number	Port Width	µDAQ Connector	Rugged µDAQ Connector	Description
XXX 24A/CA						
A	0	0	8-bits	FB2	F0	Port A
B	0	1	8-bits	FB2	F0	Port B
C	0	2	8-bits	FB2	F0	Port C
XXX 48A/C						
A	0	0	8-bits	FB2	F0	Port A
B	0	1	8-bits	FB2	F0	Port B
C	0	2	8-bits	FB2	F0	Port C
A	1	3	8-bits	FT2	F1	Port A
B	1	4	8-bits	FT2	F1	Port B
C	1	5	8-bits	FT2	F1	Port C
XXX 72A						
A	0	0	8-bits	FB2	F0	Port A
B	0	1	8-bits	FB2	F0	Port B
C	0	2	8-bits	FB2	F0	Port C
A	1	3	8-bits	FT2	F1	Port A
B	1	4	8-bits	FT2	F1	Port B
C	1	5	8-bits	FT2	F1	Port C
A	2	6	8-bits	RT2	R1	Port A
B	2	7	8-bits	RT2	R1	Port B
C	2	8	8-bits	RT2	R1	Port C
XXX 96A/C						
A	0	0	8-bits	FB3	F0	Port A
B	0	1	8-bits	FB3	F0	Port B
C	0	2	8-bits	FB3	F0	Port C
A	1	3	8-bits	FM3	F1	Port A
B	1	4	8-bits	FM3	F1	Port B
C	1	5	8-bits	FM3	F1	Port C
A	2	6	8-bits	RM3	R1	Port A
B	2	7	8-bits	RM3	R1	Port B
C	2	8	8-bits	RM3	R1	Port C
A	3	9	8-bits	FT3	F2	Port A
B	3	10	8-bits	FT3	F2	Port B
C	3	11	8-bits	FT3	F2	Port C
XXX 120A						
A	0	0	8-bits	FB3	F0	Port A
B	0	1	8-bits	FB3	F0	Port B
C	0	2	8-bits	FB3	F0	Port C
A	1	3	8-bits	FM3	F1	Port A
B	1	4	8-bits	FM3	F1	Port B
C	1	5	8-bits	FM3	F1	Port C
A	2	6	8-bits	RM3	R1	Port A
B	2	7	8-bits	RM3	R1	Port B
C	2	8	8-bits	RM3	R1	Port C
A	3	9	8-bits	FT3	F2	Port A
B	3	10	8-bits	FT3	F2	Port B
C	3	11	8-bits	FT3	F2	Port C
A	4	12	8-bits	RT3	R2	Port A
B	4	13	8-bits	RT3	R2	Port B
C	4	14	8-bits	RT3	R2	Port C

R-XXX 144A/C						
A	0	0	8-bits		F0	Port A
B	0	1	8-bits		F0	Port B
C	0	2	8-bits		F0	Port C
A	1	3	8-bits		F1	Port A
B	1	4	8-bits		F1	Port B
C	1	5	8-bits		F1	Port C
A	2	6	8-bits		R1	Port A
B	2	7	8-bits		R1	Port B
C	2	8	8-bits		R1	Port C
A	3	9	8-bits		F2	Port A
B	3	10	8-bits		F2	Port B
C	3	11	8-bits		F2	Port C
A	4	12	8-bits		R2	Port A
B	4	13	8-bits		R2	Port B
C	4	14	8-bits		R2	Port C
A	5	15	8-bits		F3	Port A
B	5	16	8-bits		F3	Port B
C	5	17	8-bits		F3	Port C
R-XXX 168A						
A	0	0	8-bits		F0	Port A
B	0	1	8-bits		F0	Port B
C	0	2	8-bits		F0	Port C
A	1	3	8-bits		F1	Port A
B	1	4	8-bits		F1	Port B
C	1	5	8-bits		F1	Port C
A	2	6	8-bits		R1	Port A
B	2	7	8-bits		R1	Port B
C	2	8	8-bits		R1	Port C
A	3	9	8-bits		F2	Port A
B	3	10	8-bits		F2	Port B
C	3	11	8-bits		F2	Port C
A	4	12	8-bits		R2	Port A
B	4	13	8-bits		R2	Port B
C	4	14	8-bits		R2	Port C
A	5	15	8-bits		F3	Port A
B	5	16	8-bits		F3	Port B
C	5	17	8-bits		F3	Port C
A	6	18	8-bits		R3	Port A
B	6	19	8-bits		R3	Port B
C	6	20	8-bits		R3	Port C
XXX 62-16						
A	0	0	8-bits	FB2	F0	Port A
B	0	1	8-bits	FB2	F0	Port B
C	0	2	8-bits	FB2	F0	Port C
		3	8-bits	FT2	F1	Opto Port
		4	8-bits	BT2	R1	Opto Port
XXX 62-32						
A	0	0	8-bits	FB3	F0	Port A
B	0	1	8-bits	FB3	F0	Port B
C	0	2	8-bits	FB3	F0	Port C
		3	8-bits	FM3	F1	Opto Port
		4	8-bits	BM3	R1	Opto Port
		5	8-bits	FT3	F2	Opto Port
		6	8-bits	BT3	R2	Opto Port
XXX 63-16						
A	0	0	8-bits	FB2	F0	Port A
B	0	1	8-bits	FB2	F0	Port B
C	0	2	8-bits	FB2	F0	Port C
		3	8-bits	FT2	F1	Relay Port
		4	8-bits	BT2	R1	Relay Port
XXX 63-32						
A	0	0	8-bits	FB3	F0	Port A
B	0	1	8-bits	FB3	F0	Port B
C	0	2	8-bits	FB3	F0	Port C
		3	8-bits	FM3	F1	Relay Port
		4	8-bits	BM3	R1	Relay Port
		5	8-bits	FT3	F2	Relay Port
		6	8-bits	BT3	R2	Relay Port
XXX 69-16						
A	0	0	8-bits	FB2	F0	Port A
B	0	1	8-bits	FB2	F0	Port B
C	0	2	8-bits	FB2	F0	Port C
		3	8-bits	FT2	F1	Relay Port
		4	8-bits	BT2	R1	Opto Port

XXX 69-32						
A	0	0	8-bits	FB3	F0	Port A
B	0	1	8-bits	FB3	F0	Port B
C	0	2	8-bits	FB3	F0	Port C
		3	8-bits	FM3	F1	Relay Port
		4	8-bits	BM3	R1	OptoPort
		5	8-bits	FT3	F2	Relay Port
		6	8-bits	BT3	R2	OptoPort
XXX 26/30						
A	0	0	8-bits	FB2	F0	Port A
B	0	1	8-bits	FB2	F0	Port B
C	0	2	8-bits	FB2	F0	Port C
XXX 73R/T						
A	0	0	8-bits	FB2	F0	Port A
B	0	1	8-bits	FB2	F0	Port B
C	0	2	8-bits	FB2	F0	Port C

Table 4-1 Digital I/O Assigned Ports

4.3.1 Reading the Digital Inputs – EDRE_DioRead

A single call is necessary to read a digital I/O port. Depending on the interface the result is either passed through a reference parameter or by the returned code. The digital I/O ports a self managed, meaning it will automatically configure as inputs when read from. Please note that due to the identity of the digital I/O devices the output ports will looses its output value when the a port is configured differently. It's very important to do the appropriate reads and writes to ensure the device is in proper configuration before using it. Each interface function is showed separately in the next section.

4.3.1.1 Shared Object and Linked Library Interface

Function Name	EDRE_DioRead
Object	Edrapi.dll / edrapi.so
Platform	Linux, Microsoft Windows, Pocket PC 2003
Parameters	32-bit unsigned integer – Serial Number 32-bit unsigned integer – Port Pointer to 32-bit unsigned integer – Value
Return Error	32-bit signed integer
Error Codes	0 – EDRE_OK – No error -1 – EDRE_FAIL – General failure -3 – EDRE_BAD_SN – Device with serial number does not exist -14 – EDRE_BAD_PARAMTER – Port value is incorrect

4.3.1.2 ActiveX Interface

Function Name	Read
Object	EDREDioX
Platform	Microsoft Windows
Related Property	SerialNumber
Parameters	32-bit signed integer – Port
Return Value	32-bit signed long
Error Codes	>=0 – Value read from port – No error -1 – EDRE_FAIL – General failure -3 – EDRE_BAD_SN – Device with serial number does not exist -14 – EDRE_BAD_PARAMTER – Port value is incorrect

4.3.1.3 EDR Enhanced Main Object Interface

Function Name	DioRead
Object	EDREObject
Platform	Linux

Parameters	32-bit unsigned integer 32-bit unsigned integer Pointer to 32-bit unsigned integer	Serial Number Port Value
Return Value	32-bit signed integer	
Error Codes		

0	Value read from port	No error
-1	EDRE_FAIL	General failure
-3	EDRE_BAD_SN	Device with serial number does not exist
-14	EDRE_BAD_PARAMTER	Port value is incorrect

4.3.1.4 EDR Enhanced Device Object Interface

Function Name	DioRead	
Object	EDREDevice	
Platform	Linux	
Parameters	32-bit unsigned integer Pointer to 32-bit unsigned integer	Port Value
Return Value	32-bit signed integer	
Error Codes		

0	Value read from port	No error
-1	EDRE_FAIL	General failure
-3	EDRE_BAD_SN	Device with serial number does not exist
-14	EDRE_BAD_PARAMTER	Port value is incorrect

4.3.2 Writing to the Digital Outputs – EDRE_DioWrite

A single call is necessary to write to a digital I/O port. Depending on the interface type a port number and value is passed to the function. A returned error code will tell if the function passed or not. . Please note that due to the identity of the digital I/O devices the output ports will looses its output value when the a port is configured differently. It's very important to do the appropriate reads and writes to ensure the device is in proper configuration before using it. Each interface function is showed separately in the next section.

4.3.2.1 Shared Object and Linked Library Interface

Function Name	EDRE_DioWrite	
Object	Edrapi.dll / edrapi.so	
Platform	Linux, Microsoft Windows, Pocket PC 2003	
Parameters	32-bit unsigned integer 32-bit unsigned integer 32-bit unsigned integer	Serial Number Port Value
Return Error	32-bit signed integer	
Error Codes		

0	EDRE_OK	No error
-1	EDRE_FAIL	General failure
-3	EDRE_BAD_SN	Device with serial number does not exist
-14	EDRE_BAD_PARAMTER	Either the port or value is incorrect

4.3.2.2 ActiveX Interface

Function Name	Write	
Object	EDREDioX	
Platform	Microsoft Windows	
Related Property	SerialNumber	
Parameters	32-bit signed integer	Port

	32-bit signed integer	Value	
Return Value	32-bit signed integer		
Error Codes	0	EDRE_OK	No error
	-1	EDRE_FAIL	General failure
	-3	EDRE_BAD_SN	Device with serial number does not exist
	-14	EDRE_BAD_PARAMTER	Either the port or value is incorrect

4.3.2.3 EDR Enhanced Main Object Interface

Function Name	DioWrite		
Object	EDREObject		
Platform	Linux		
Parameters	32-bit unsigned integer	Serial Number	
	32-bit unsigned integer	Port	
	32-bit unsigned integer	Value	
Return Value	32-bit signed integer		
Error Codes	0	EDRE_OK	No error
	-1	EDRE_FAIL	General failure
	-3	EDRE_BAD_SN	Device with serial number does not exist
	-14	EDRE_BAD_PARAMTER	Either the port or value is incorrect

4.3.2.4 EDR Enhanced Device Object Interface

Function Name	DioWrite		
Object	EDREDevice		
Platform	Linux		
Parameters	32-bit unsigned integer	Port	
	32-bit unsigned integer	Value	
Return Value	32-bit signed integer		
Error Codes	0	EDRE_OK	No error
	-1	EDRE_FAIL	General failure
	-3	EDRE_BAD_SN	Device with serial number does not exist
	-14	EDRE_BAD_PARAMTER	Either the port or value is incorrect

4.3.3 Query Codes

The digital I/O sub-system can be queried for the number of ports available and the property of each port. The following query codes are relevant to the digital I/O system.

Name	Value	Description
DIONUMPORT	400	Query the number of digital I/O ports installed
DIOQRYPORT	401	Query a specific port for its properties. Use PARAM to specify the port. 0 — DIOOUTPUT — Port is output only 1 — DIOINPUT — Port is input only 2 — DIOINOROUT — Port is either in or out 3 — DIOINANDOUT — Port is in and out
DIOPORTWIDTH	402	Query a specific port width. Use PARAM to specify the port. The value returned is the bit width of the port.

4.3.3.1 Query number of ports

PSEUDO START
Dioports of type 32-bit integer
  Dioport = EDRE_Query(serialnumber, DIONUMPORT,0)
PSEUDO STOP

4.3.3.2 Query port type

“Port type can be output, input, in-and-out, in-or-out
PSEUDO START
Port_type of type 32-bit integer
  For p = 0 to Dioports-1 do
    Port_type = EDRE_Query(serialnumber, DIOQRYPORT,p)
  Next p
PSEUDO STOP

4.3.3.3 Query port width in bits

PSEUDO START
Port_width of type 32-bit integer
  For p = 0 to Dioports-1 do
    Port_width = EDRE_Query(serialnumber, DIOPORTWIDTH,p)
  Next p
PSEUDO STOP

4.4 Counters

The counter-timers sub-system is supported by functions to Write, Read, Configure and controlling the internal gate. There are 6 counters on the digital I/O C-version models. The table below shows the counter-timer number assignment. All the counter-timers are Intel 8254 compatible. For further reference on the operation please refer to the datasheet of the 8254 device. The counter-timers are mainly used to generate pulses or to count them. One of the most useful functions of the counter-timers is to generate digital square waves. It can also generate timed events through interrupts. Please note that only USB has support for interrupts.

Counter	Assigned Number	Description
CT0	0	Counter 0
CT1	1	Counter 1
CT2	2	Counter 2
CT3	3	Counter 3
CT4	4	Counter 4
CT5	5	Counter 5

Table 4-2 Counter Assignment

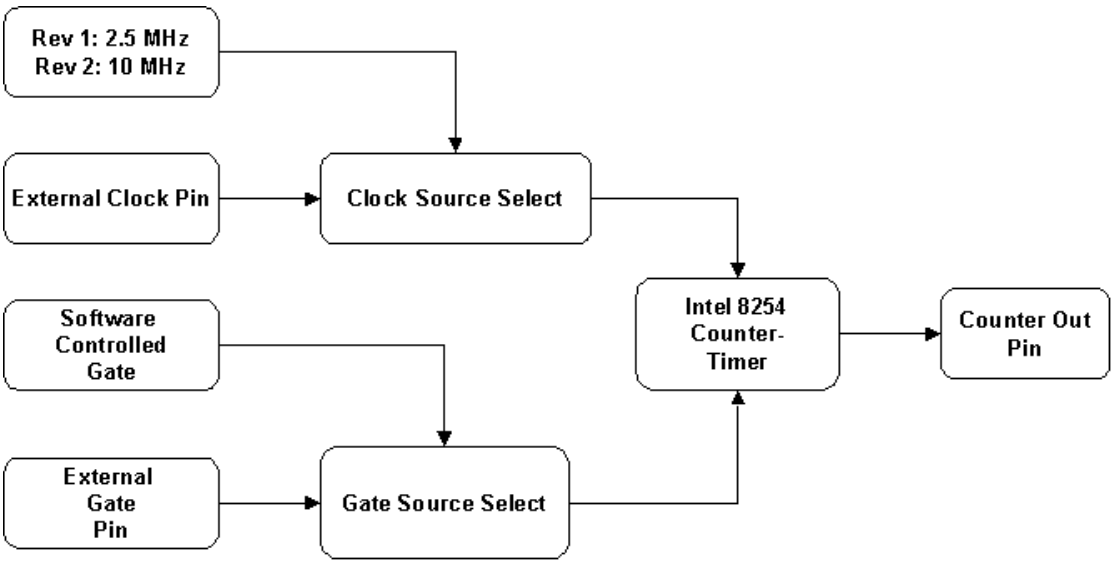


Figure 4-2 Counter-Timer Architecture

4.4.1 Architecture

All the counters have 16-bit count values and all always count down. When the initial value is loaded it will decrement on each clock pulse. The clock source and gate source can be selected via software. The clock source can either be internal or external. The gate can also be either internal or external. The internal gate is controlled via software. The external gate pin has a pull-up resistor that allows the gate to be enabled by default when set to external.

4.4.2 Writing the initial counter value – EDRE_CTWrite

A single call is necessary to write a counter's initial load value. The initial load value will only be loaded into the counter on the first clock pulse. This will affect the read value when no clock is applied to the counter before a read.

4.4.2.1 Shared Object and Linked Library Interface

Function Name	EDRE_CTWrite
Object	Edrapi.dll / edrapi.so
Platform	Linux, Microsoft Windows, Pocket PC 2003

Parameters	32-bit unsigned integer 32-bit unsigned integer 32-bit unsigned integer	Sn Port Value	
Return Error	32-bit signed integer		
Error Codes	0	EDRE_OK	No error
	-1	EDRE_FAIL	General failure
	-3	EDRE_BAD_SN	Device with serial number does not exist
	-14	EDRE_BAD_PARAMTER	Either the counter or value is incorrect

4.4.2.2 ActiveX Interface

Function Name	Write		
Object	EDRECtX		
Platform	Microsoft Windows		
Related Property	SerialNumber		
Parameters	32-bit signed integer 32-bit signed integer	Counter Value	
Return Value	32-bit signed integer		
Error Codes	0	EDRE_OK	No error
	-1	EDRE_FAIL	General failure
	-3	EDRE_BAD_SN	Device with serial number does not exist
	-14	EDRE_BAD_PARAMTER	Either the counter or value is incorrect

4.4.2.3 EDR Enhanced Main Object Interface

Function Name	CTWrite		
Object	EDREObject		
Platform	Linux		
Parameters	32-bit unsigned integer 32-bit unsigned integer 32-bit unsigned integer	Sn Port Value	
Return Value	32-bit signed integer		
Error Codes	0	EDRE_OK	No error
	-1	EDRE_FAIL	General failure
	-3	EDRE_BAD_SN	Device with serial number does not exist
	-14	EDRE_BAD_PARAMTER	Either the counter or value is incorrect

4.4.2.4 EDR Enhanced Device Object Interface

Function Name	CTWrite		
Object	EDREDevice		
Platform	Linux		
Parameters	32-bit unsigned integer 32-bit unsigned integer	Port Value	
Return Value	32-bit signed integer		
Error Codes	0	EDRE_OK	No error
	-1	EDRE_FAIL	General failure
	-3	EDRE_BAD_SN	Device with serial number does not exist
	-14	EDRE_BAD_PARAMTER	Either the counter or value is incorrect

4.4.3 Reading the counter value – EDRE_CTRead

A single call is necessary to read a counter. The value returned will be the current register value of the counter. Please note that when written a value and now clock was applied to value read will be the last count value and not the written value.

4.4.3.1 Shared Object and Linked Library Interface

Function Name	EDRE_CTRead		
Object	Edrapi.dll / edrapi.so		
Platform	Linux, Microsoft Windows, Pocket PC 2003		
Parameters	32-bit unsigned integer	Sn	
	32-bit unsigned integer	Port	
	Pointer to 32-bit unsigned integer	Value	
Return Error	32-bit signed integer		
Error Codes	0	EDRE_OK	No error
	-1	EDRE_FAIL	General failure
	-3	EDRE_BAD_SN	Device with serial number does not exist
	-14	EDRE_BAD_PARAMTER	Port is incorrect

4.4.3.2 ActiveX Interface

Function Name	Read		
Object	EDRECtX		
Platform	Microsoft Windows		
Related Property	SerialNumber		
Parameters	32-bit signed integer	Counter	
Return Value	32-bit signed integer		
Error Codes	0	Value read from counter	No error
	-1	EDRE_FAIL	General failure
	-3	EDRE_BAD_SN	Device with serial number does not exist
	-14	EDRE_BAD_PARAMTER	Counter is incorrect

4.4.3.3 EDR Enhanced Main Object Interface

Function Name	CTRead		
Object	EDREObject		
Platform	Linux		
Parameters	32-bit unsigned integer	Sn	
	32-bit unsigned integer	Ct	
	Pointer to 32-bit unsigned integer	Value	
Return Value	32-bit signed integer		
Error Codes	0	EDRE_OK	No error
	-1	EDRE_FAIL	General failure
	-3	EDRE_BAD_SN	Device with serial number does not exist
	-14	EDRE_BAD_PARAMTER	Port is incorrect

4.4.3.4 EDR Enhanced Device Object Interface

Function Name	CTRead
Object	EDREDevice
Platform	Linux

Parameters	32-bit unsigned integer	Ct
	Pointer to 32-bit unsigned integer	Value
Return Value	32-bit signed integer	
Error Codes	0 EDRE_OK	No error
	-1 EDRE_FAIL	General failure
	-3 EDRE_BAD_SN	Device with serial number does not exist
	-14 EDRE_BAD_PARAMTER	Port is incorrect

4.4.4 Configuring a counter-timer – EDRE_CTConfig

A single call is necessary to configure a counter. Please refer to the counter-timer modes section. It is important to select the relevant mode for it to operate correctly.

4.4.4.1 Shared Object and Linked Library Interface

Function Name	EDRE_CTConfig	
Object	Edrapi.dll / edrapi.so	
Platform	Linux, Microsoft Windows, Pocket PC 2003	
Parameters	32-bit unsigned integer	Serial Number
	32-bit unsigned integer	Ct
	32-bit unsigned integer	Mode
	32-bit unsigned integer	Type
	32-bit unsigned integer	ClkSrc
	32-bit unsigned integer	GateSrc
Return Error	32-bit signed integer	
Error Codes	0 EDRE_OK	No error
	-1 EDRE_FAIL	General failure
	-3 EDRE_BAD_SN	Device with serial number does not exist
	-14 EDRE_BAD_PARAMTER	Either the counter, mode, type, clksrc or gatesrc is incorrect

4.4.4.2 ActiveX Interface

Function Name	Configure	
Object	EDRECtX	
Platform	Microsoft Windows	
Related Property	SerialNumber	
Parameters	32-bit unsigned integer	Counter
	32-bit unsigned integer	Mode
	32-bit unsigned integer	Type
	32-bit unsigned integer	ClkSrc
	32-bit unsigned integer	GateSrc
Return Value	32-bit signed integer	
Error Codes	0 EDRE_OK	No error
	-1 EDRE_FAIL	General failure
	-3 EDRE_BAD_SN	Device with serial number does not exist
	-14 EDRE_BAD_PARAMTER	Either the counter, mode, type, clksrc or gatesrc is incorrect

4.4.4.3 EDR Enhanced Main Object Interface

Function Name	CTConfig
Object	EDREObject
Platform	Linux
Parameters	32-bit unsigned integer — Serial Number 32-bit unsigned integer — Ct 32-bit unsigned integer — Mode 32-bit unsigned integer — Type 32-bit unsigned integer — ClkSrc 32-bit unsigned integer — GateSrc
Return Value	32-bit signed integer
Error Codes	0 — EDRE_OK — No error -1 — EDRE_FAIL — General failure -3 — EDRE_BAD_SN — Device with serial number does not exist -14 — EDRE_BAD_PARAMTER — Either the counter, mode, type, clksrc or gatesrc is incorrect

4.4.4.4 EDR Enhanced Device Object Interface

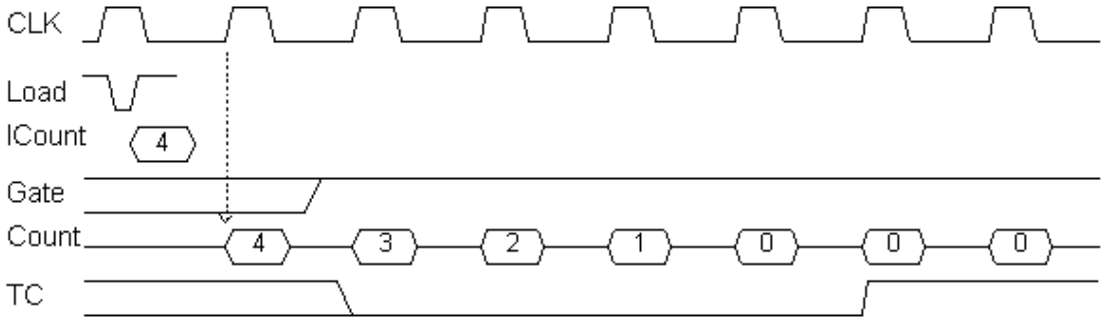
Function Name	CTConfig
Object	EDREDevice
Platform	Linux
Parameters	32-bit unsigned integer — Ct 32-bit unsigned integer — Mode 32-bit unsigned integer — Type 32-bit unsigned integer — ClkSrc 32-bit unsigned integer — GateSrc
Return Value	32-bit signed integer
Error Codes	0 — EDRE_OK — No error -1 — EDRE_FAIL — General failure -3 — EDRE_BAD_SN — Device with serial number does not exist -14 — EDRE_BAD_PARAMTER — Either the counter, mode, type, clksrc or gatesrc is incorrect

4.4.4.5 Valid parameter values for counter-timer configuration

Parameter	Description
Serial Number	Device serial number. This device should exist
Counter	Counter Number: 0 : Counter 0 1 : Counter 1 2 : Counter 2 3 : Counter 3 4 : Counter 4 5 : Counter 5
Mode	8254 Mode: See counter-timer mode section
Type	0 : Binary Count (16-bit binary) 1 : BCD count (4 decade binary coded decimal)
Source	0 : Internal (2.5/10MHz) 1 : External (External connector)
Gate	0 : Internal 1 : External (External connector)

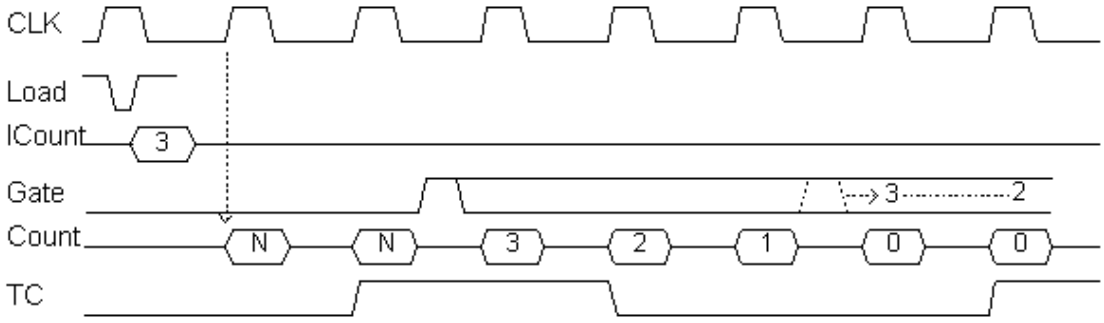
Table 4-3 Counter Configuration Parameters

4.4.4.6 Counter-Timer Mode 0



Application	Event counter
Output operation	The output is set low by the control word setting and kept low until the counter value becomes 0.
Gate function	High level enables counting and low disables counting. The gate does not affect the output.
Count value load timing	After writing the control word and initial count value, the count value is loaded into the counting register on the next falling edge of the clock pulse. The first clock pulse does not cause the count value to decrement. The output will stay low for initial count clock and go high on initial clock plus one.
Count value writing during timing	The count value is loaded on the next falling edge of the clock pulse and counting with the new value continues.
Count value writing when gate is low	The count value is also load on the falling edge of the next clock pulse. When the gate goes high the count value start decrementing and will the output will go high when he counter reaches zero after initial-count number of clocks

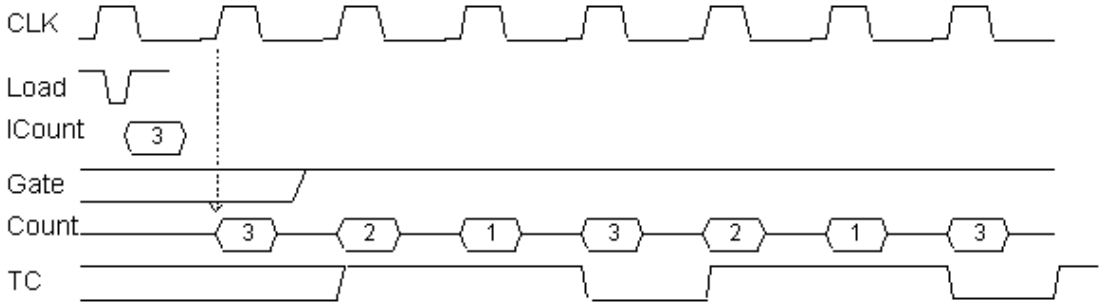
4.4.4.7 Counter-Timer Mode 1



Application	Digital one-shot
Output operation	The control word setting sets the output to high. It is set low by the clock falling edge following a gate trigger, and kept low until the counter becomes zero. Once the output is set high, it is kept high until the clock pulse succeeding a gate trigger.
Gate function	The gate is used as a trigger. A rising edge will trigger the operation. If it goes low before the end it does not affect the operation.
Count value load timing	After the control word and initial count value are written, the count value is loaded into the count register on the next clock falling edge following a gate trigger. The output is then set low. The one-shot start in this manner. The output stays low for the interval of initial-count.
Count value writing during timing	This does not affect the process until it triggers again.

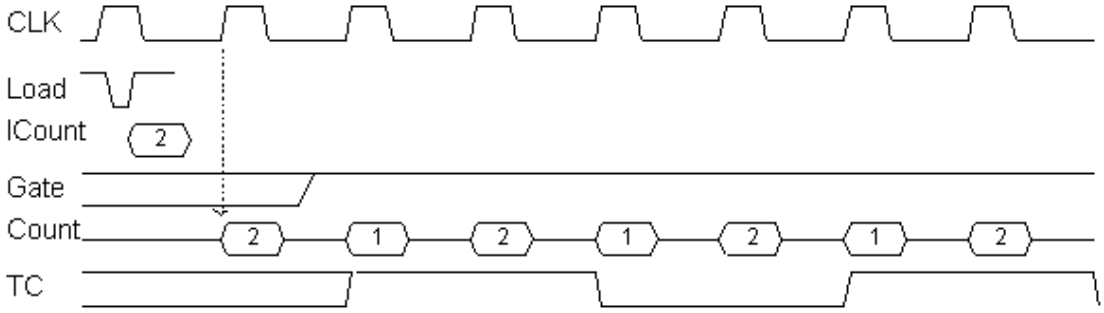
Count value writing when gate is low	This does not affect the process until it triggers again.

4.4.4.8 Counter-Timer Mode 2



Application	Rate generator, real-time interrupt clock
Output operation	The output is set to high by writing the control word. When the count value reaches one, the output goes low for one clock period. The initial count is reloaded and the process starts again.
Gate function	When high it enables counting, when low it disables counting. On a rising edge the process starts. If set low when he output is low, the output is set high immediately. The gate can be used as for counter synchronization.
Count value load timing	After the control word and initial count is written, the count value is loaded on the next clock falling edge. The output is set low after the initial-count number of clocks.
Count value writing during timing	Writing during operation does not affect the process. The new count values will be loaded when the counter laps. A gate trigger can load a new count value.
Count value writing when gate is low	A new count value will be loaded when the gate is set high again.

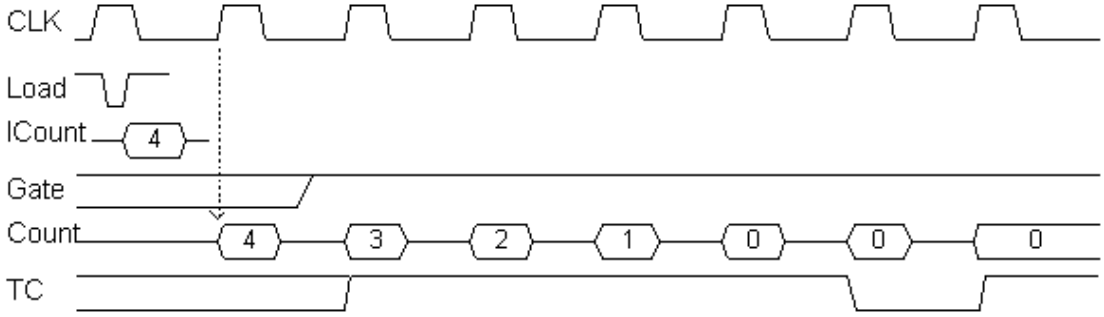
4.4.4.9 Counter-Timer Mode 3



Application	Baud rate generator, square wave generator
Output operation	Same as mode two except the output duty is different. The output is set high by writing the control word. When the count value reaches half of the initial-count, the output goes low for the remainder of the count. This mode repeats the above sequence periodically.
Gate function	A high level enables counting and low level stop counting. If the gate is set low when the output is low, the output is set high immediately. The count value is reloaded on the next clock pulse following a gate trigger.
Count value load timing	The count value is loaded after the control word and initial count value was written.
Count value	Writing during counting does not affect the process. The new value is

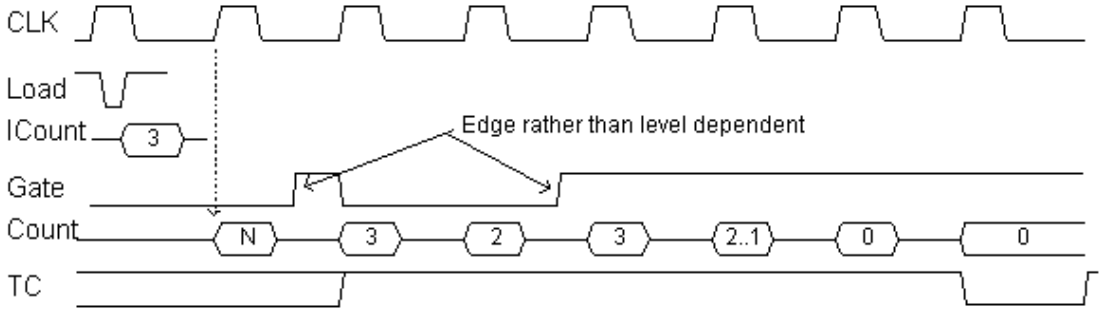
writing during timing	loaded when the counter laps. A gate trigger during counting will reload the count value.
Even number counting	The output is initially set high. The initial count value is loaded on the next falling clock pulse, and is decremented by two on every clock pulse. When the counter reaches two, the output is set low and the initial-count is reloaded. The process the repeats.
Odd number counting	The output is initially set high. The initial-count value minus one is loaded on the falling clock pulse, and then decremented by two on each clock pulse. When the count value becomes zero, the output is set low. The initial count value minus one is reloaded again, and the counter decrement by two. The output is set high when the count value is two. The output is set high again and the process repeats. The high level period is (initial-count + 1) / 2 and the low period is (initial-count - 1) / 2.

4.4.4.10 Counter-Timer Mode 4



Application	Software triggered strobe
Output operation	The output is initially set high. The count value is loaded on the next falling clock pulse. The output goes low for one clock when the count value becomes zero. The process restart when a new initial-count is written.
Gate function	The gate does not affect the output. When set high it enables counting. When set low it disables counting.
Count value load timing	After the control word and initial-count value is written, the counter register is loaded on he following clock pulse. The strobe is output initial-count plus one after the initial-count was written.
Count value writing during timing	The new count value is loaded immediately on the next clock falling edge. The counter then continues with the new count. This means the counting is triggered by software.

4.4.4.11 Counter-Timer Mode 5



Application	Hardware triggered strobe
Output operation	The output is initially set high. When the counter value becomes zero after triggered by a rising edge on the gate, the output goes low for one clock period. The output restores to high again.

Gate function	The gate does not affect the output. The initial-count value is loaded on the next clock falling edge succeeding a gate rising edge trigger pulse. The gate triggers the count sequence.
Count value load timing	Even after the control word and initial-count was written, the process only start on the following clock falling edge succeeding a gate trigger. The output will go low initial-count plus one after the trigger occurred.
Count value writing during timing	Writing during while counting does not affect the current sequence. The new value will only be loaded on a new trigger.

4.4.4.12 Counter-Timer Mode Summary

Mode \ Gate	Low level falling edge	Rising Edge	High Level
0	Counting not possible		Counting possible
1		Start counting Retriggering	
2	Counting not possible Output forced high	Start counting	Counting possible
3	Counting not possible Output forced high	Start counting	Counting possible
4	Counting not possible		Counting possible
5		Start counting Retriggering	

4.4.5 Controlling the counter-timer gate – EDRE_CTGate

A single call is necessary to control a counter's gate. A gate value of one enables the gate and zero disables the gate.

4.4.5.1 Shared Object and Linked Library Interface

Function Name	EDRE_CTSoftGate
Object	Edrapi.dll / edrapi.so
Platform	Linux, Microsoft Windows, Pocket PC 2003
Parameters	32-bit unsigned integer – Sn 32-bit unsigned integer – Port 32-bit unsigned integer – Gate
Return Error	32-bit signed integer
Error Codes	0 – EDRE_OK – No error -1 – EDRE_FAIL – General failure -3 – EDRE_BAD_SN – Device with serial number does not exist -14 – EDRE_BAD_PARAMTER – Port is incorrect

4.4.5.2 ActiveX Interface

Function Name	SoftGate
Object	EDRECtX
Platform	Microsoft Windows
Related Property	SerialNumber
Parameters	32-bit signed integer – Counter 32-bit signed integer – Value
Return Value	32-bit signed integer
Error Codes	0 – EDRE_OK – No error

	-1	EDRE_FAIL	General failure
	-3	EDRE_BAD_SN	Device with serial number does not exist
	-14	EDRE_BAD_PARAMTER	Either the counter or value is incorrect

4.4.5.3 EDR Enhanced Main Object Interface

Function Name	CTSoftGate		
Object	EDREObject		
Platform	Linux		
Parameters	32-bit unsigned integer 32-bit unsigned integer 32-bit unsigned integer	Sn Port Gate	
Return Value	32-bit signed integer		
Error Codes	0	EDRE_OK	No error
	-1	EDRE_FAIL	General failure
	-3	EDRE_BAD_SN	Device with serial number does not exist
	-14	EDRE_BAD_PARAMTER	Port value incorrect

4.4.5.4 EDR Enhanced Device Object Interface

Function Name	CTWrite		
Object	EDREDevice		
Platform	Linux		
Parameters	32-bit unsigned integer 32-bit unsigned integer	Port Gate	
Return Value	32-bit signed integer		
Error Codes	0	EDRE_OK	No error
	-1	EDRE_FAIL	General failure
	-3	EDRE_BAD_SN	Device with serial number does not exist
	-14	EDRE_BAD_PARAMTER	Port value incorrect

4.4.6 Internal Oscillator Frequency (2.5MHz or 10MHz)

The internal oscillator frequency is different from the first revision to the second revision of the counter timer enabled units. The revision 1 units have a frequency of 2.5 MHz. For the second revision the frequency is 10 MHz.

4.5 Programming Interrupts (USB only)

The interrupt sub-system is totally programmable and includes functions to configure, enable and disable interrupts. The EDR Enhanced API implements the interrupt service routine notification through block I/O. However the EDR Enhanced ActiveX control for Windows make use of event notification. The interrupt interface has a release function to release a blocked call prematurely.

4.5.1 Procedure

1	Disable Interrupts by default
2	Configure each interrupt source by making the appropriate configure calls.
3	Enable interrupts
4	Call WaitOnInterrupt for API implementation or implement event handler for ActiveX
5	When triggered WaitOnInterrupt will release or the ActiveX event handler will be called. The interrupt source will each bit set of the source of the interrupt. The WaitOnInterrupt function call will return an error code that contains the success of the operation. The ActiveX implementation will have a negative source if it failed.
6	Call Release interrupt in the case of the API.
7	Always disable interrupts when done.

4.5.2 Configuring the Interrupt sub-system – EDRE_IntConfigure

A single call is necessary to configure the interrupt sub-system.

4.5.2.1 Shared Object and Linked Library Interface

Function Name	EDRE_IntConfigure
Object	Edrapi.dll / edrapi.so
Platform	Linux, Microsoft Windows
Parameters	32-bit unsigned integer – Serial Number 32-bit unsigned integer – Source 32-bit unsigned integer – Mode 32-bit unsigned integer – Type
Return Error	32-bit signed integer
Error Codes	0 – EDRE_OK – No error -1 – EDRE_FAIL – General failure -3 – EDRE_BAD_SN – Device with serial number does not exist -14 – EDRE_BAD_PARAMTER – Either source, mode or type is incorrect.

4.5.2.2 ActiveX Interface

Function Name	Configure
Object	EDREIntX
Platform	Microsoft Windows
Related Property	SerialNumber
Parameters	32-bit signed integer – Source 32-bit signed integer – Mode 32-bit signed integer – Type
Return Value	32-bit signed integer
Error Codes	0 – EDRE_OK – No error

-1	EDRE_FAIL	General failure
-3	EDRE_BAD_SN	Device with serial number does not exist
-14	EDRE_BAD_PARAMTER	Either source, mode or type is incorrect.

4.5.2.3 EDR Enhanced Main Object Interface

Function Name	IntConfigure
Object	EDREObject
Platform	Linux
Parameters	32-bit unsigned integer – Serial Number 32-bit unsigned integer – Source 32-bit unsigned integer – Mode 32-bit unsigned integer – Type
Return Value	32-bit signed integer
Error Codes	0 EDRE_OK No error -1 EDRE_FAIL General failure -3 EDRE_BAD_SN Device with serial number does not exist -14 EDRE_BAD_PARAMTER Either source, mode or type is incorrect.

4.5.2.4 EDR Enhanced Device Object Interface

Function Name	IntConfigure
Object	EDREDevice
Platform	Linux
Parameters	32-bit unsigned integer – Source 32-bit unsigned integer – Mode 32-bit unsigned integer – Type
Return Value	32-bit signed integer
Error Codes	0 EDRE_OK No error -1 EDRE_FAIL General failure -3 EDRE_BAD_SN Device with serial number does not exist -14 EDRE_BAD_PARAMTER Either source, mode or type is incorrect.

4.5.2.5 Interrupt Configuration Parameters

The table below shows the possible values to configure the interrupt sub-system.

Parameter	Type	Description
Source	long	0 Counter 0 1 Counter 1 2 Counter 2 3 Counter 3 4 Counter 4 5 Counter 5
Mode	long	Disable or Enable a source 0 : Disable 1 : Enable
Type	long	Set the type of trigger for the interrupt 0 Rising Edge 1 Falling Edge

Table 4-4 Interrupt Configuration Parameters

4.5.3 Enabling Interrupts – EDRE_IntEnable

A single call is necessary to enable the interrupt sub-system.

4.5.3.1 Shared Object and Linked Library Interface

Function Name	EDRE_IntEnable		
Object	Edrapi.dll / edrapi.so		
Platform	Linux, Microsoft Windows		
Parameters	32-bit unsigned integer	Serial Number	
Return Error	32-bit signed integer		
Error Codes	0	EDRE_OK	No error
	-1	EDRE_FAIL	General failure
	-3	EDRE_BAD_SN	Device with serial number does not exist

4.5.3.2 ActiveX Interface

Function Name	Enable		
Object	EDREIntX		
Platform	Microsoft Windows		
Related Property	SerialNumber		
Parameters	None		
Return Value	32-bit signed integer		
Error Codes	0	EDRE_OK	No error
	-1	EDRE_FAIL	General failure
	-3	EDRE_BAD_SN	Device with serial number does not exist

4.5.3.3 EDR Enhanced Main Object Interface

Function Name	IntEnable		
Object	EDREObject		
Platform	Linux		
Parameters	32-bit unsigned integer	Serial Number	
Return Value	32-bit signed integer		
Error Codes	0	EDRE_OK	No error
	-1	EDRE_FAIL	General failure
	-3	EDRE_BAD_SN	Device with serial number does not exist

4.5.3.4 EDR Enhanced Device Object Interface

Function Name	IntEnable		
Object	EDREDevice		
Platform	Linux		
Parameters	None		
Return Value	32-bit signed integer		
Error Codes	0	EDRE_OK	No error
	-1	EDRE_FAIL	General failure
	-3	EDRE_BAD_SN	Device with serial number does not exist

4.5.4 Disabling Interrupts – EDRE_IntDisable

A single call is necessary to disable interrupts.

⚠ **If interrupts are disabled, the interrupt sources need to be reconfigured. The interrupt mask register inside the µDAQ and Rugged µDAQ will be reset.**

4.5.4.1 Shared Object and Linked Library Interface

Function Name	EDRE_IntDisable
Object	Edrapi.dll / edrapi.so
Platform	Linux, Microsoft Windows
Parameters	32-bit unsigned integer — Serial Number
Return Error	32-bit signed integer
Error Codes	0 — EDRE_OK — No error -1 — EDRE_FAIL — General failure -3 — EDRE_BAD_SN — Device with serial number does not exist

4.5.4.2 ActiveX Interface

Function Name	Disable
Object	EDREIntX
Platform	Microsoft Windows
Related Property	SerialNumber
Parameters	None
Return Value	32-bit signed integer
Error Codes	0 — EDRE_OK — No error -1 — EDRE_FAIL — General failure -3 — EDRE_BAD_SN — Device with serial number does not exist

4.5.4.3 EDR Enhanced Main Object Interface

Function Name	IntDisable
Object	EDREObject
Platform	Linux
Parameters	32-bit unsigned integer — Serial Number
Return Value	32-bit signed integer
Error Codes	0 — EDRE_OK — No error -1 — EDRE_FAIL — General failure -3 — EDRE_BAD_SN — Device with serial number does not exist

4.5.4.4 EDR Enhanced Device Object Interface

Function Name	IntDisable
Object	EDREDevice
Platform	Linux
Parameters	None
Return Value	32-bit signed integer
Error Codes	0 — EDRE_OK — No error -1 — EDRE_FAIL — General failure -3 — EDRE_BAD_SN — Device with serial number does not exist

4.5.5 Releasing an Interrupt Prematurely – EDRE_ReleaseInterrupt

Sometimes it's required to release a blocked *WaitOnInterrupt* call. The API has a function to do this. The *WaitOnInterrupt* source will in such a case returned with an error code.

4.5.5.1 Shared Object and Linked Library Interface

Function Name	EDRE_ReleaseInterrupt
Object	Edrapi.dll / edrapi.so

Platform	Linux, Microsoft Windows		
Parameters	32-bit unsigned integer	Serial Number	
Return Error	32-bit signed integer		
Error Codes	0	EDRE_OK	No error
	-1	EDRE_FAIL	General failure
	-3	EDRE_BAD_SN	Device with serial number does not exist

4.5.5.2 ActiveX Interface

Function Name	ClearEvent		
Object	EDREIntX		
Platform	Microsoft Windows		
Related Property	SerialNumber		
Parameters	None		
Return Value	32-bit signed integer		
Error Codes	0	EDRE_OK	No error
	-1	EDRE_FAIL	General failure
	-3	EDRE_BAD_SN	Device with serial number does not exist

4.5.5.3 EDR Enhanced Main Object Interface

Function Name	ReleaseInterrupt		
Object	EDREObject		
Platform	Linux		
Parameters	32-bit unsigned integer	Serial Number	
Return Value	32-bit signed integer		
Error Codes	0	EDRE_OK	No error
	-1	EDRE_FAIL	General failure
	-3	EDRE_BAD_SN	Device with serial number does not exist

4.5.5.4 EDR Enhanced Device Object Interface

Function Name	ReleaseInterrupt		
Object	EDREDevice		
Platform	Linux		
Parameters	None		
Return Value	32-bit signed integer		
Error Codes	0	EDRE_OK	No error
	-1	EDRE_FAIL	General failure
	-3	EDRE_BAD_SN	Device with serial number does not exist

4.5.6 ActiveX Interrupt Event

If interrupts are enabled an event will occur on each interrupt. The interrupt control's interrupt event will be triggered. The source of the interrupt will also be passed to the event handler. If the source is negative, it means an error occurred. If the source is a positive integer it, each bit set will indicate the interrupt source.

4.5.6.1 ActiveX Event

Function Name	Interrupt
Object	EDREIntX
Platform	Windows

Parameters	Source	If >0 the interrupt was triggered by hardware, else it will have a negative error code.	
Errors	-1	EDRE_FAIL	General failure
	-6	EDRE_EVENT_FAILED	Event was release prematurely.

4.5.7 WaitOnInterrupt API Function – EDRE_WaitOnInterrupt

The WaitOnInterrupt function is a blocked I/O call that will release on a hardware interrupt. It will contain a returned error and interrupt source. Each bit set will indicate the source of the interrupt. This means that more than one source could have triggered the interrupt.

4.5.7.1 Shared Object and Linked Library Interface

Function Name	EDRE_WaitOnInterrupt		
Object	Edrapi.dll / edrapi.so		
Platform	Linux, Microsoft Windows		
Parameters	32-bit unsigned integer	Serial Number	
Return Error	32-bit signed integer		
Error Codes	0	EDRE_OK	No error
	-1	EDRE_FAIL	General failure
	-15	EDRE_BUSY	General failure
	-3	EDRE_BAD_SN	Device with serial number does not exist

4.5.7.2 EDR Enhanced Main Object Interface

Function Name	WaitOnInterrupt		
Object	EDREObject		
Platform	Linux		
Parameters	32-bit unsigned integer	Serial Number	
Return Value	32-bit signed integer		
Error Codes	0	EDRE_OK	No error
	-1	EDRE_FAIL	General failure
	-15	EDRE_BUSY	General failure
	-3	EDRE_BAD_SN	Device with serial number does not exist

4.5.7.3 EDR Enhanced Device Object Interface

Function Name	WaitOnInterrupt		
Object	EDREDevice		
Platform	Linux		
Parameters	None		
Return Value	32-bit signed integer		
Error Codes	0	EDRE_OK	No error
	-1	EDRE_FAIL	General failure
	-15	EDRE_BUSY	General failure
	-3	EDRE_BAD_SN	Device with serial number does not exist

4.6 Analog Outputs

The µDAQ 30 and Rugged µDAQ 30 have up to 8 digital to analog channels. The output voltage range is ±10V @ 14-bit resolution. The output voltages are simple to set, you simply have to specify the channel and microvolt value.

4.6.1 Writing to a DAC channel – EDRE_DAWrite

A single call is necessary to set a voltage on a DAC channel. The table below shows the relation between the software channel and the channel on the connector.

Assigned Software Channel	Assigned Connector Pin	µDAQ30A16 Connector Location	µDAQ30A32 Connector Location	Rugged µDAQ Connector Location
0	DAC0	FRONT TOP 2-TIER	FRONT MIDDLE 3-TIER	F1
1	DAC1	FRONT TOP 2-TIER	FRONT MIDDLE 3-TIER	F1
2	DAC2	FRONT TOP 2-TIER	FRONT MIDDLE 3-TIER	F1
3	DAC3	FRONT TOP 2-TIER	FRONT MIDDLE 3-TIER	F1
4	DAC4	Not installed	FRONT TOP 3-TIER	F2
5	DAC5	Not installed	FRONT TOP 3-TIER	F2
6	DAC6	Not installed	FRONT TOP 3-TIER	F2
7	DAC7	Not installed	FRONT TOP 3-TIER	F2

Table 4-5 Assigned DAC channels and

4.6.1.1 Shared Object and Linked Library Interface

Function Name	EDRE_DAWrite	
Object	Edrapi.dll / edrapi.so	
Platform	Linux, Microsoft Windows, Pocket PC 2003	
Parameters	32-bit unsigned integer	Sn
	32-bit unsigned integer	Channel
	32-bit unsigned integer	Microvolt
Return Error	32-bit signed integer	
Error Codes	0	EDRE_OK	No error
	-1	EDRE_FAIL	General failure
	-3	EDRE_BAD_SN	Device with serial number does not exist
	-14	EDRE_BAD_PARAMTER	Either the channel or voltage is incorrect

4.6.1.2 ActiveX Interface

Function Name	Write	
Object	EDREDAX	
Platform	Microsoft Windows	
Related Property	SerialNumber	
Parameters	32-bit signed integer	Channel
	32-bit signed integer	Microvolt
Return Value	32-bit signed integer	

Error Codes	0	EDRE_OK	No error
	-1	EDRE_FAIL	General failure
	-3	EDRE_BAD_SN	Device with serial number does not exist
	-14	EDRE_BAD_PARAMTER	Either the channel or voltage is incorrect

4.6.1.3 EDR Enhanced Main Object Interface

Function Name	DAWrite	
Object	EDREObject	
Platform	Linux	
Parameters	32-bit unsigned integer 32-bit unsigned integer 32-bit unsigned integer	Serial Number Channel Microvolt
Return Value	32-bit signed integer	
Error Codes	0	EDRE_OK	No error
	-1	EDRE_FAIL	General failure
	-3	EDRE_BAD_SN	Device with serial number does not exist
	-14	EDRE_BAD_PARAMTER	Either the channel or voltage is incorrect

4.6.1.4 EDR Enhanced Device Object Interface

Function Name	DAWrite		
Object	EDREDevice		
Platform	Linux		
Parameters	32-bit unsigned integer 32-bit unsigned integer	Channel Microvolt	
Return Value	32-bit signed integer		
Error Codes	0	EDRE_OK	No error
	-1	EDRE_FAIL	General failure
	-3	EDRE_BAD_SN	Device with serial number does not exist
	-14	EDRE_BAD_PARAMTER	Either the channel or voltage is incorrect

4.7 Analog Inputs – USB 26/30 SRL 26/30 BT 26/30 (A-Version)

The µDAQ-26/30A and Rugged µDAQ-26/30A analog input models have either 16 or 32 input channels. The analog input sub-system support two modes of operation. The first is simply to read an analog channel with a list of specific settings. The second mode is to supply a channel list with each channel's individual settings, and to scan the list continuously at a specific frequency sequentially.

The current 14-bit model only supports channel scanning on sixteen channels. In the case of the 32-channel model only the first sixteen or the second sixteen channels can be scanned. The channels cannot be mixed. This limitation will be addressed in future products.

The maximum sampling speed of the USB model is 250 KHz for USB 1.1 and 400 KHz for USB 2.0. The USB model also support automatic over-sampling at low frequencies. The serial and wireless model has a maximum sampling speed of 3 KHz and makes use of over-sampling to run the analog to digital converter at its optimum conversion speed. The serial model's maximum sampling frequency decreases with lower BAUD rates. 3KHz is rated at 115200 BAUD. Consult the serial A/D product specification appendix for the rated sampling rate for a specific BAUD rate.

4.7.1 Configuration Constants for A/D Inputs (A-Version)

The A/D input has various voltage ranges and gains that need to be configured before reading a voltage. It is important to select the most optimum range and also to make sure not to drive the analog inputs outside its operating range. The input circuit can either operate as a single-ended input or a differential input. In the single-ended mode analog ground is the reference, and in the case of differential mode the channel's pair is the reference. The differential pairing table will show each channel's pair mate. On the 32-channel models the first and second sixteen channels operate independently, so care must be taken to completely understand the setup of the channel list.

4.7.1.1 Channel Configuration Constants – Range and Gain

4.7.1.1.1 A/B Versions

The analog input circuit has nine gains and four ranges. The table below shows the valid combinations.

<table>
<tr><th>Voltage Range Constant</th><th>Voltage Range Name</th><th>Possible Voltage Gains</th></tr>
<tr><td>0</td><td>UNIPOLAR SINGLE-ENDED</td><td>
<table>
<tr><th>Value</th><th>Gain</th><th>Voltage Range</th></tr>
<tr><td>1</td><td>X ½</td><td>0-10V</td></tr>
<tr><td>2</td><td>X 1</td><td>0-5V</td></tr>
<tr><td>3</td><td>X 2.5</td><td>0-2V</td></tr>
<tr><td>4</td><td>X 5</td><td>0-1V</td></tr>
<tr><td>5</td><td>X 10</td><td>0-500mV</td></tr>
<tr><td>6</td><td>X 25</td><td>0-200mV</td></tr>
<tr><td>7</td><td>X 50</td><td>0-100mV</td></tr>
<tr><td>8</td><td>X 100</td><td>0-50mV</td></tr>
</table>
</td></tr>
<tr><td>1</td><td>BIPOLAR SINGLE-ENDED</td><td>
<table>
<tr><th>Value</th><th>Gain</th><th>Voltage Range</th></tr>
<tr><td>0</td><td>X ¼</td><td>±10V</td></tr>
<tr><td>1</td><td>X ½</td><td>±5V</td></tr>
<tr><td>2</td><td>X 1</td><td>±2.5V</td></tr>
<tr><td>3</td><td>X 2.5</td><td>±1V</td></tr>
<tr><td>4</td><td>X 5</td><td>±500mV</td></tr>
<tr><td>5</td><td>X 10</td><td>±250mV</td></tr>
<tr><td>6</td><td>X 25</td><td>±100mV</td></tr>
<tr><td>7</td><td>X 50</td><td>±50mV</td></tr>
</table>
</td></tr>
</table>

		8	X 100	±25mV
2	UNIPOLAR DIFFERENTIAL	Value	Gain	Voltage Range
		1	X ½	0-10V
		2	X 1	0-5V
		3	X 2.5	0-2V
		4	X 5	0-1V
		5	X 10	0-500mV
		6	X 25	0-200mV
		7	X 50	0-100mV
		8	X 100	0-50mV
3	BIPOLAR DIFFERENTIAL	Value	Gain	Voltage Range
		0	X ¼	±10V
		1	X ½	±5V
		2	X 1	±2.5V
		3	X 2.5	±1V
		4	X 5	±500mV
		5	X 10	±250mV
		6	X 25	±100mV
		7	X 50	±50mV
		8	X 100	±25mV

Table 4-6 A/D Channel Configuration Constants

4.7.1.1.2 C Version

The analog input circuit has only one gain and two ranges. The table below shows the valid combinations.

Voltage Range Constant	Voltage Range Name	Possible Voltage Gains		
0	BIPOLAR SINGLE-ENDED	Value	Gain	Voltage Range
		0	X 1	±10V
1	BIPOLAR DIFFERENTIAL-ENDED	Value	Gain	Voltage Range
		0	X 1	±10V

Table 4-7 A/D Channel Configuration Constants

4.7.1.2 A/D Single Ended and Differential Channel Pairing (A/B/C-Version)

The analog input system uses analog ground as the voltage reference in single ended mode and another channel in differential mode. When in differential mode the channel's pair becomes unavailable for A/D operations. Because of this only half of the channels can be used when all are configured as differential. The first sixteen channels are grouped together and the next sixteen on the 32-channel models.

Channels from the first sixteen cannot be mixed with the next sixteen channels when scanning the inputs. This is true for differential inputs as well.

Assigned Software Channel	Input Type	Input Pin	Reference Pin
0	Single	ACH0	AGND
…	…	…	…
15	Single	ACH15	AGND
0	Differential	ACH0	ACH8
…	…	…	…
7	Differential	ACH7	ACH15
16	Single	ACH0	AGND
…	…	…	…
31	Single	ACH15	AGND
16	Differential	ACH16	ACH24

...	...	...	...
23	Differential	ACH23	ACH31

Table 4-8 A/D Channel Pairing

4.7.1.3 Clock Source and Gate Source

When scanning the analog inputs the clock source (or convert source) can be selected. This can be either external or internal. If internal the frequency can also be set. The internal clock source is 10MHz and has a resolution of 16-bits. When set for external the signal enters on the EXT_CLK pin.

Selecting the gate as external can also gate the A/D convert signal. Selecting the gate as internal means the convert pulse is always enabled. The external gate signal enters on the EXT_GATE pin. When set high the convert pulses are enabled, if set low the convert process is disabled.

The table below shows the assigned constants and the format of the clock source parameter.

Value	Description
0	Internal 10MHz clock
1	External clock/convert on EXT_CLK pin
256	External Gate and internal 10MHz clock
257	External gate and external clock/convert Gate high = enabled Gate low = disabled

Table 4-9 Clock/Gate Source Constants

4.7.1.4 The scanning frequency

A frequency needs to be specified when scanning the channel list. The channel list is executed synchronously, which means the frequency is the time between to consecutive executions. The µDAQ and Rugged µDAQ analog input units automatically make use of over sampling for frequencies below 1000 Hz. This is to make sure the A/D converter uses sample and hold technology and running to slow will degrade the performance. The frequency range is 1 Hz to maximum depending on the model. The internal clock frequency is 10 MHz with a resolution of 16-bits.

4.7.2 Reading a single voltage from a channel – EDRE_ADSingle

To read a single ADC channel you need to specify the channel, voltage range and gain. Gain is specified via the gain parameter or the gain property and the range via the range parameter or property. The microvolt buffer parameter will hold the voltage read if no error occurred. In the case of the AxtiveX control the voltage will be in the return parameter. The voltage is in micro volt form.

4.7.2.1 Shared Object and Linked Library Interface

Function Name	EDRE_ADSingle		
Object	Edrapi.dll / edrapi.so		
Platform	Linux, Microsoft Windows, Pocket PC 2003		
Parameters	32-bit unsigned integer	Sn	
	32-bit unsigned integer	Channel	
	32-bit unsigned integer	Gain	
	32-bit unsigned integer	Range	
	Pointer to 32-bit unsigned integer	Microvolt buffer	
Return Error	32-bit signed integer		
Error Codes	0	EDRE_OK	No error
	-1	EDRE_FAIL	General failure
	-3	EDRE_BAD_SN	Device with serial number does not exist
	-14	EDRE_BAD_PARAMTER	Either the channel, gain

			or range is incorrect

4.7.2.2 ActiveX Interface

Function Name	SingleRead
Object	EDREADX
Platform	Microsoft Windows
Related Property	SerialNumber, Range, Gain
	Have you set the range and gain before reading a channel?
Parameters	32-bit signed integer: Channel
Return Value	32-bit signed integer
Error Codes	The returned value will hold the voltage read as a micro voltage. If the method fails it will create an exception error.

4.7.2.3 EDR Enhanced Main Object Interface

Function Name	ADSingle		
Object	EDREObject		
Platform	Linux		
Parameters	32-bit unsigned integer		Sn
	32-bit unsigned integer		Channel
	32-bit unsigned integer		Gain
	32-bit unsigned integer		Range
	Pointer to 32-bit unsigned integer		Microvolt buffer
Return Value	32-bit signed integer		
Error Codes	0	EDRE_OK	No error
	-1	EDRE_FAIL	General failure
	-3	EDRE_BAD_SN	Device with serial number does not exist
	-14	EDRE_BAD_PARAMTER	Either the channel, range or gain is incorrect

4.7.2.4 EDR Enhanced Device Object Interface

Function Name	ADSingle		
Object	EDREDevice		
Platform	Linux		
Parameters	32-bit unsigned integer		Channel
	32-bit unsigned integer		Gain
	32-bit unsigned integer		Range
	Pointer to 32-bit unsigned integer		Microvolt buffer
Return Value	32-bit signed integer		
Error Codes	0	EDRE_OK	No error
	-1	EDRE_FAIL	General failure
	-3	EDRE_BAD_SN	Device with serial number does not exist
	-14	EDRE_BAD_PARAMETER	Either the channel, gain or range is incorrect

4.7.3 Configuring the ADC subsystem for scanning – EDRE_ADConfig

This is the most complicated part of configuring the USB 26/30 for auto scanning. Make sure that you use the correct format when applying the channel list configuration. There are many loopholes and care should be taken when implementing code to configure the USB 26/30. The most important part is to build the channel list array and the gain/range list array. The functions below are used for this purpose.

The channel list array is simply an array filled with channel numbers. The channels are zero indexed, so the first channel is zero. The maximum channel length is equal to the number of channels available on the device.

The gain and range list array is an array filled gain and range settings corresponding to the same index in the channel list array. In the gain and range list array the gain is the first eight bits of each value and the range the next eight bits.

4.7.3.1 Shared Object and Linked Library Interface

Function Name	EDRE_ADConfig		
Object	Edrapi.dll / edrapi.so		
Platform	Linux, Microsoft Windows, Pocket PC 2003		
Parameters	32-bit unsigned integer		Sn
	Point to 32-bit unsigned integer		Frequency
	32-bit unsigned integer		ClockSource
	32-bit unsigned integer		Burst Mode: Unused
	32-bit unsigned integer		Range: Unused, the range is specified with the gain list
	Pointer to 32-bit unsigned integer array		ChanList buffer
	Pointer to 32-bit unsigned integer array		GainList buffer
	32-bit unsigned integer		ListLength
Return Error	32-bit signed integer		
Error Codes	0	EDRE_OK	No error
	-1	EDRE_FAIL	General failure
	-3	EDRE_BAD_SN	Device with serial number does not exist
	-14	EDRE_BAD_PARAMETER	Frequency or ClockSource out of range
	-22	EDRE_BAD_CHANLIST	Either the channel list or gain and range list is incorrect

4.7.3.2 ActiveX Interface

Function Name	Configure		
Object	EDREADX		
Platform	Microsoft Windows		
Related Property	SerialNumber, ClockSource, Frequency		
Parameters	Pointer to 32-bit unsigned integer array		ChanList buffer
	Pointer to 32-bit unsigned integer array		GainList buffer
	32-bit unsigned integer		ListLength
Return Value	32-bit signed integer		
Error Codes	0	EDRE_OK	No error

-1	EDRE_FAIL	General failure
-3	EDRE_BAD_SN	Device with serial number does not exist
-14	EDRE_BAD_PARAMETER	Frequency or ClockSource out of range
-22	EDRE_BAD_CHANLIST	Either the channel list or gain and range list is incorrect

4.7.3.3 EDR Enhanced Main Object Interface

Function Name	ADConfig
Object	EDREObject
Platform	Linux
Parameters	32-bit unsigned integer — Sn Point to 32-bit unsigned integer — Frequency 32-bit unsigned integer — ClockSource 32-bit unsigned integer — Burst Mode: Unused 32-bit unsigned integer — Range: Unused, the range is specified with the gain list Pointer to 32-bit unsigned integer array — ChanList buffer Pointer to 32-bit unsigned integer array — GainList buffer 32-bit unsigned integer — ListLength
Return Value	32-bit signed integer
Error Codes	0 EDRE_OK No error -1 EDRE_FAIL General failure -3 EDRE_BAD_SN Device with serial number does not exist -14 EDRE_BAD_PARAMETER Frequency or ClockSource out of range -22 EDRE_BAD_CHANLIST Either the channel list or gain and range list is incorrect

4.7.3.4 EDR Enhanced Device Object Interface

Function Name	ADConfig
Object	EDREDevice
Platform	Linux
Parameters	Point to 32-bit unsigned integer — Frequency 32-bit unsigned integer — ClockSource 32-bit unsigned integer — Burst Mode: Unused 32-bit unsigned integer — Range: Unused, the range is specified with the gain list Pointer to 32-bit unsigned integer array — ChanList buffer Pointer to 32-bit unsigned integer array — GainList buffer 32-bit unsigned integer — ListLength
Return Value	32-bit signed integer
Error Codes	0 EDRE_OK No error -1 EDRE_FAIL General failure -3 EDRE_BAD_SN Device with serial number does not exist

	-3	EDRE_BAD_SN	Device with serial number does not exist
	-14	EDRE_BAD_PARAMETER	Frequency or ClockSource out of range
	-22	EDRE_BAD_CHANLIST	Either the channel list or gain and range list is incorrect

4.7.3.5 Channel, Gain and Range list example

PSEUDO START
Channel_list_array[0..15] of type unsigned 32-bit integer
Gain_list_array[0..15] of type unsigned 32-bit integer
Gain_Range of type unsigned 32-bit ineteger
Gain of type unsigned 32-bit integer
Range of type unsigned 32-bit integer

For c = 0 to 15
  Channel_list_array[c] = c "populate each channel array position with a channel number"
   Gain = 0 "Gain of x ¼"
   Range = 1 "Range of bipolar single ended"
  Gain_Range = Gain logic or with (Range shift left by 8) "Calculate value"
  Gain_list_array[c] = Gain_Range "write to channel list"
Next c
PSEUDO STOP

Result of array build process

Array Index	Channel_list_array	Gain_list_array
0	0	256
1	1	256
2	2	256
3	3	256
4	4	256
5	5	256
6	6	256
7	7	256
8	8	256
9	9	256
10	10	256
11	11	256
12	12	256
13	13	256
14	14	256
15	15	256

This will result in channels 0-15 be scanned bipolar, single ended and ±10V.

4.7.4 Starting the A/D process – EDRE_ADStart

After the A/D sub-system has been successfully configured the process can be started. A single API call is necessary to start sampling data. If no error is returned the data will immediately start moving to the driver buffer. Use the **ADUNREAD** query code to monitor the number available samples in the driver buffer.

4.7.4.1 Shared Object and Linked Library Interface

Function Name	EDRE_ADStart	
Object	Edrapi.dll / edrapi.so	
Platform	Linux, Microsoft Windows	
Parameters	32-bit unsigned integer	Serial Number

Return Error	32-bit signed integer		
Error Codes	0	EDRE_OK	No error
	-1	EDRE_FAIL	General failure
	-3	EDRE_BAD_SN	Device with serial number does not exist
	-15	EDRE_BUSY	Device is already busy sampling

4.7.4.2 ActiveX Interface

Function Name	Start		
Object	EDREADX		
Platform	Microsoft Windows		
Related Property	SerialNumber		
Parameters	None		
Return Value	32-bit signed integer		
Error Codes	0	EDRE_OK	No error
	-1	EDRE_FAIL	General failure
	-15	EDRE_BUSY	Device is already busy sampling
	-3	EDRE_BAD_SN	Device with serial number does not exist

4.7.4.3 EDR Enhanced Main Object Interface

Function Name	ADStart		
Object	EDREObject		
Platform	Linux		
Parameters	32-bit unsigned integer	Serial Number	
Return Value	32-bit signed integer		
Error Codes	0	EDRE_OK	No error
	-1	EDRE_FAIL	General failure
	-15	EDRE_BUSY	Device is already busy sampling
	-3	EDRE_BAD_SN	Device with serial number does not exist

4.7.4.4 EDR Enhanced Device Object Interface

Function Name	ADStart		
Object	EDREDevice		
Platform	Linux		
Parameters	None		
Return Value	32-bit signed integer		
Error Codes	0	EDRE_OK	No error
	-1	EDRE_FAIL	General failure
	-15	EDRE_BUSY	Device is already busy sampling
	-3	EDRE_BAD_SN	Device with serial number does not exist

4.7.5 Stopping the A/D process – EDRE_ADStop

When enough data was collected or the A/D sub-system needs to be reconfigured the process can be stopped. The functions below can be used to execute this command.

4.7.5.1 Shared Object and Linked Library Interface

Function Name	EDRE_ADStop
Object	Edrapi.dll / edrapi.so
Platform	Linux, Microsoft Windows
Parameters	32-bit unsigned integer — Serial Number
Return Error	32-bit signed integer
Error Codes	0 — EDRE_OK — No error -1 — EDRE_FAIL — General failure -3 — EDRE_BAD_SN — Device with serial number does not exist

4.7.5.2 ActiveX Interface

Function Name	Stop
Object	EDREADX
Platform	Microsoft Windows
Related Property	SerialNumber
Parameters	None
Return Value	32-bit signed integer
Error Codes	0 — EDRE_OK — No error -1 — EDRE_FAIL — General failure -3 — EDRE_BAD_SN — Device with serial number does not exist

4.7.5.3 EDR Enhanced Main Object Interface

Function Name	ADStop
Object	EDREObject
Platform	Linux
Parameters	32-bit unsigned integer — Serial Number
Return Value	32-bit signed integer
Error Codes	0 — EDRE_OK — No error -1 — EDRE_FAIL — General failure -3 — EDRE_BAD_SN — Device with serial number does not exist

4.7.5.4 EDR Enhanced Device Object Interface

Function Name	ADStop
Object	EDREDevice
Platform	Linux
Parameters	None
Return Value	32-bit signed integer
Error Codes	0 — EDRE_OK — No error -1 — EDRE_FAIL — General failure -3 — EDRE_BAD_SN — Device with serial number does not exist

4.7.6 Driver buffer functions – EDRE_ADGetData

The device driver contains a large circular buffer to store data. The driver contains a buffer manager and operates on a first-in-first-out (FIFO) basis. The size of the buffer can be queried by making use of the **ADBUFFSIZE** query code. The **ADUNREAD** query code is used to determine the number of available samples in the driver buffer. If the driver buffer is

not emptied regularly, the buffer will overrun generating an error condition. On an overrun error the process will stop automatically, but the sampled data will still be available in the driver buffer.

A user space buffer is needed to be able to transfer the data from the driver. The user needs to allocate memory for this. The data in the buffer is organized in the same order as the channel list. For this reason it is best practice to always request data as a multiple of the channel list size. Always check the buffer size parameter on return because the driver can copy less data than requested.

4.7.6.1 Shared Object and Linked Library Interface

Function Name	EDRE_ADGetData
Object	Edrapi.dll / edrapi.so
Platform	Linux, Microsoft Windows, Pocket PC 2003
Parameters	32-bit unsigned integer — Sn Pointer to 32-bit integer array — Data buffer Pointer to 32-bit unsigned integer array — Buffer size Always check this value on return. It holds the number of samples copied to the buffer
Return Error	32-bit signed integer
Error Codes	0 — EDRE_OK — No error -1 — EDRE_FAIL — General failure -3 — EDRE_BAD_SN — Device with serial number does not exist

4.7.6.2 ActiveX Interface

Function Name	GetData
Object	EDREADX
Platform	Microsoft Windows
Related Property	SerialNumber, ClockSource, Frequency
Parameters	32-bit unsigned integer — Sn Pointer to 32-bit integer array — Data buffer Pointer to 32-bit integer array — Buffer size Always check this value on return. It holds the number of samples copied to the buffer
Return Value	32-bit signed integer
Error Codes	

4.7.6.3 EDR Enhanced Main Object Interface

Function Name	ADGetData
Object	EDREObject
Platform	Linux
Parameters	32-bit unsigned integer — Sn Pointer to 32-bit integer array — Data buffer Pointer to 32-bit unsigned integer array — Buffer size Always check this value on return. It holds the number of samples copied to the buffer
Return Value	32-bit signed integer

Error Codes			
	0	EDRE_OK	No error
	-1	EDRE_FAIL	General failure
	-3	EDRE_BAD_SN	Device with serial number does not exist

4.7.6.4 EDR Enhanced Device Object Interface

Function Name	ADGetData	
Object	EDREDevice	
Platform	Linux	
Parameters	Pointer to 32-bit integer array Pointer to 32-bit unsigned integer array	Data buffer Buffer size Always check this value on return. It holds the number of samples copied to the buffer
Return Value	32-bit signed integer	
Error Codes	0 EDRE_OK	No error
	-1 EDRE_FAIL	General failure
	-3 EDRE_BAD_SN	Device with serial number does not exist

4.7.7 Querying the ADC subsystem

The driver can be queried to check the status of the ADC subsystem. The number of unread samples is one example. The table below shows a list of query codes that are relevant to the analog input sub-system.

Name	Value	Description
ADNUMCHAN	100	Query the number of channels installed
ADMAXFREQ	102	This is the maximum sampling frequency
ADBUSY	103	Query if the analog input system is busy sampling data. 0 = IDLE 1 = BUSY
ADBUFFSIZE	106	Query the number of samples that the driver buffer cam hold.
ADBUFFOVER	107	Query if the driver buffer has overrun.
ADUNREAD	109	Query the number of samples available in the driver buffer

4.7.7.1 Query the number of A/D channels

PSEUDO START
Channels of type 32-bit integer
  Channels = EDRE_Query(serialnumber, ADNUMCHAN,0)
PSEUDO STOP

4.7.7.2 Query the maximum sampling frequency

PSEUDO START
MaxFrequency of type 32-bit integer
  MaxFrequency = EDRE_Query(serialnumber, ADMAXFREQ,0)
PSEUDO STOP

4.7.7.3 Query if A/D system is busy

PSEUDO START
  If EDRE_Query(serialnumber, ADBUSY,0) = 1 then busy
  Else not busy
PSEUDO STOP

4.7.7.4 Query the driver buffer size

PSEUDO START
BufferSize of type 32-bit integer
  BufferSize = ERE_Query(serialnumber, ADBUFFSIZE,0)
PSEUDO STOP

4.7.7.5 Query for buffer overrun

PSEUDO START
  If EDRE_Query(serialnumber, ADBUFFOVER,0) = 1 then overrun
  Else not overrun
PSEUDO STOP

4.7.7.6 Query unread data

PSEUDO START
Unread of type 32-bit integer
  Unread = EDRE_Query(serialnumber, ADUNREAD,0)
PSEUDO STOP

4.8 Temperature Input – 73 Model

The µDAQ has three different basic temperature models and the Rugged µDAQ has two. The µDAQ is available in an eight, sixteen and 32-channel model and the Rugged µDAQ in a sixteen and 32-channel model.

There are eight channels available on a single DB25 male connector and one cold junction compensation (CJC) channel. Only a genuine Eagle Technology thermocouple adapter can be used to interface to the thermocouple inputs. This adapter has a build CJC device and is necessary to read thermocouple temperatures.

The µDAQ and Rugged µDAQ temperature units also support RTD inputs. If RTDs are used a RTD adapter must be connected. The RTD adapter has build in support for a 1mA and 10mA current source. It supports both the two wire and four wire system.

⚠ Note that the µDAQ and Rugged µDAQ units are manufactured for either thermocouples or RTD and cannot be changed by swapping the adapter modules.

4.8.1 Thermocouple Procedure

To read a thermocouple channel a sequence needs to be followed. The steps below shows what steps to follow the read a thermocouple channel.

1. Read CJC channel
2. Calculate ambient temperature
3. Read thermocouple channel
4. Calculate temperature

4.8.2 Reading CJC Channel

To be able to read and calculate a thermocouple voltage the CJC channel need to be read. This value is used to calculate the cold-junction-compensation for a thermocouple channel. A thermocouple adapter's CJC channel has a circuit that will supply a voltage of 10mV per 1 degree Celsius. The CJC channel is read in the same way as a normal analog channel. The index of the channel is very important as for a unit can have more than one CJC channel. The CJC channel must always be read before reading a number of channels on a thermocouple adapter. The table below shows the CJC channel index for a particular type of device.

XXX = USB, SRL, BT

Devices	CJC Channels	
XXX 73T8	CJC	Value Assigned
	0	8
XXX 73T16 R-XXX 73T16	CJC	Value Assigned
	0	16
	1	17
XXX 73T32 R-XXX 73T32	CJC	Value Assigned
	0	32
	1	33
	2	34
	3	35

Table 4-10 CJC Channels Assigned

4.8.2.1 Query the CJC channel index

The index of each CJC channel can be queried

PSEUDO START
CJC_channel of type 32-bit integer
  Do

```
   CJC_channel = EDRE_Query(serialnumber, ADAMBCHAN,0)
  Until CJC_channel < 0
PSEUDO STOP
```

4.8.2.2 Read CJC channel

To read a CJC channel use the EDRE_ADSingle command. For further reference on this command see the analog input section. The channel must be the CJC channel index. The Gain and Range parameter are not used.

```
PSEUDO START
CJC_value of type 32-bit signed integer
CJC_channel of type 32-bit unsigned integer
  CJC_channel = 8
  error = EDRE_ADSingle(serialnumber, CJC_channel,0,0,pointer of CJC_value)
  if error = 0 print CJC_value
  else  print “Error reading CJC channel”
PSEUDO STOP
```

4.8.3 Types of Thermocouples

The EDR Enhanced application program interface supports the conversion of many thermocouple types. The table below lists all the supported types. Each type has its purpose like wide temperature range or very accurate in a certain range. The constant value is used when calculating the temperature.

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
8	Type C

Table 4-11 Thermocouple Type Table

4.8.4 Reading a Thermocouple Channel

Reading the thermocouple channel is the same as reading an analog channel or CJC channel. The channels on the 73 models are zero indexed. If the device is a hybrid that has normal analog channels and temperature channels, the temperature channels will follow the analog channels. As with the reading of the CJC channel the range and gain parameter of the EDRE_ADSingle command is not used.

The example below will read thermocouple channel 0.

```
PSEUDO START
TC_value of type 32-bit signed integer
  error = EDRE_ADSingle(serialnumber, 0,0,0,pointer of TC_value)
  if error = 0 print TC_value
  else print “Error reading termocpuple channel”
PSEUDO STOP
```

4.8.5 Calculating Ambient Temperature

After reading the CJC channel the ambient temperature can be calculated. The voltage read from the CJC channel must be passed to this conversion function. The returned value will be the temperature in millidegrees.

4.8.5.1 Shared Object and Linked Library Interface

Function Name	EDRE_CalcCJCmC	
Object	Edrapi.dll	
Platform	Microsoft Windows	
Parameters	32-bit signed integer	CJC microvolt
Return Value	32-bit signed integer	
Value	Millidegrees	

4.8.5.2 ActiveX Interface

Function Name	CalcCJCmC	
Object	EDREADX	
Platform	Microsoft Windows	
Related Property	None	
Parameters	32-bit signed integer	CJC microvolt
Return Value	32-bit signed integer	
Value	Millidegrees	

4.8.5.3 EDR Enhanced Main Object Interface

Function Name	CalcCJCmC	
Object	EDREObject	
Platform	No support	
Parameters	32-bit signed integer	CJC microvolt
Return Value	32-bit signed integer	
Value	Millidegrees	

4.8.5.4 EDR Enhanced Device Object Interface

Function Name	CalcCJCmC	
Object	EDREDevice	
Platform	No support	
Parameters	32-bit signed integer	CJC microvolt
Return Value	32-bit signed integer	
Value	Millidegrees	

4.8.6 Calculating Temperature for Thermocouples

After the CJC channel and thermocouple channel was read and the CJC value converted to millidegrees, the thermocouple temperature can be calculated. The thermocouple type, thermocouple channel voltage and ambient temperature need to be supplied in order to do this. The returned value will be in millidegrees.

4.8.6.1 Shared Object and Linked Library Interface

Function Name	EDRE_CalcTCmC	
Object	Edrapi.dll	
Platform	Microsoft Windows	
Parameters	32-bit signed integer array	Thermocouple type
	32-bit signed integer array	Thermocouple channel voltage
	32-bit signed integer array	Ambient temperature in millidegrees
Return Value	32-bit signed integer	

Value	Millidegrees

4.8.6.2 ActiveX Interface

Function Name	CalcTCmC	
Object	EDREADX	
Platform	Microsoft Windows	
Related Property	None	
Parameters	32-bit signed integer	Thermocouple type
	32-bit signed integer	Thermocouple channel voltage
	32-bit signed integer	Ambient temperature in millidegrees
Return Value	32-bit signed integer	
Value	Millidegrees	

4.8.6.3 EDR Enhanced Main Object Interface

Function Name	CalcTCmC	
Object	EDREObject	
Platform	No support	
Parameters	32-bit signed integer	Thermocouple type
	32-bit signed integer	Thermocouple channel voltage
	32-bit signed integer	Ambient temperature in millidegrees
Return Value	32-bit signed integer	
Value	Millidegrees	

4.8.6.4 EDR Enhanced Device Object Interface

Function Name	CalcTCmC	
Object	EDREDevice	
Platform	No support	
Parameters	32-bit signed integer	Thermocouple type
	32-bit signed integer	Thermocouple channel voltage
	32-bit signed integer	Ambient temperature in millidegrees
Return Value	32-bit signed integer	
Value	Millidegrees	

4.8.6.5 Example of calculating a thermocouple temperature

The tempera

PSEUDO START

```
Define TYPE_K 1
CJC_millideg of type 32 signed integer
TC_value of type 32-bit signed integer
TC_millideg of type 32-bit signed integer

  TC_millideg = EDRE_CalcTCmC(TYPE_K, TC_value, CJC_millideg)
  Print TC_millideg
```

PSEUDO STOP

5

5 Calibration Procedure

This chapter deals with calibrating your hardware. Please follow the instruction carefully to properly calibrate your device.

5.1 Calibrating the USB 26/30, SRL 26/30, BT 26/30 and Rugged μDAQ

The analog μDAQ and Rugged μDAQ devices need to be calibrated to ensure accurate operation. This section describes the calibration process, requirements and procedure. Specialized equipment is needed to perform this procedure. Please make sure that your equipment conforms to the preset specification. The two main items are required: a precision multimeter and voltage source.

5.1.1 Equipment

The following calibration equipment is required to calibrate the analog μDAQ and Rugged μDAQ device. If the calibration equipment does conform to these specifications it will not be possible to calibrate the device accurately.

5.1.1.1 High Precision Multimeter

A high precision multimeter is required to measure output analog voltages. The HP3478A digital multimeter is an example of such a device. This device is used as standard test equipment to calibrate the analog μDAQ and Rugged μDAQ device. Make sure the device conform to its own calibration requirements and that it is serviced regularly. The device requirements are the following.

Item	Specification
Voltage Range	-10V to 10V
Type	Analog Input
Relative Accuracy	0.1 % of 1 bit in 16384
Accuracy	< 1.2 μV

Table 5-1 Analog Multimeter Requirements

5.1.1.2 High Precision Voltage Source

A high precision voltage source is required to generate input analog voltages. The Burster Digistant Typ 4405 is am example of such a device. The device is used as standard test equipment to calibrate the μDAQ and Rugged μDAQ device. Make sure the device conform to its own calibration requirements and that it is serviced regularly. The device requirements are the following.

Item	Specification

Voltage Range	0V to 10V
Type	Analog Output
Relative Accuracy	0.1 % of 1 bit in 16384
Accuracy	< 1.2 µV

Table 5-2 Analog Source Requirements

5.1.1.3 Coaxial Calibration Cable

The µDAQ and Rugged µDAQ devices are only calibrated on one analog input channel and all analog output channels. A specialized cable is needed to connect the calibration equipment to the device. The connection points are the following.

Source	µDAQ and Rugged µDAQ Pin	Destination
Analog Ground	AGND – 9	1. Voltage Generator Reference
		2. Voltage Meter Reference
Analog Input Channel 0	ACH0 - 1	Voltage Generator Positive
Analog Output Channel 0	DAC0 - 22	Voltage Meter Positive
Analog Output Channel 1	DAC1 – 10	Voltage Meter Positive
Analog Output Channel 2	DAC2 – 23	Voltage Meter Positive
Analog Output Channel 3	DAC3 - 11	Voltage Meter Positive
10V Reference	10V_REFCAL - 24	Voltage Meter Positive

Table 5-3 Calibration Cable Connections

The diagram below shows a typical connection cable. To reduce external noise effects on the process only use coaxial cables. Banana type plugs can be used to connect to the calibration instruments. Use a shielded DB25 connector to connect to the µDAQ and Rugged µDAQ unit.

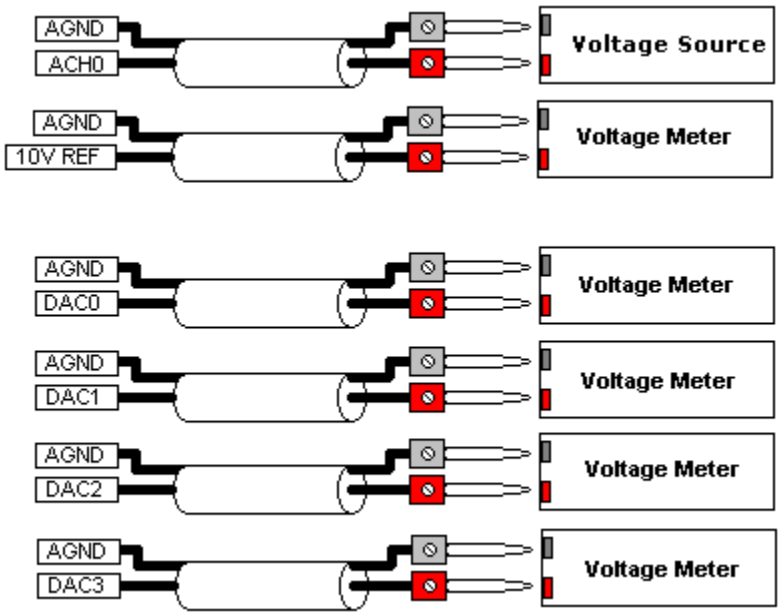


Figure 5-1 Calibration Cable

5.1.2 Software

The analog µDAQ and Rugged µDAQ require that the device software must be installed and operational. If not see the **Getting Started** chapter to setup the device. The next step is to install the calibration software for the µDAQ and Rugged µDAQ device.

5.1.2.1 Software Installation

The software to calibrate the A and B (14-bit) versions can be found on the Eagle Technology CDROM at *<EAGLECD>\EDRE\APPS\udaq_cal_26_30*. The software for the C (16-bit) version can be found on the Eagle Technology CDROM at *<EAGLECD>\EDRE\APPS\udaq_cal_26C_30C.*

The installation application will place a short cut under Eagle Technology on the Windows Menu System.

5.1.2.2 Operation

The software application will indicate the current step and a description of what to do. Use the buttons at the bottom to navigate. The slider bar is used to adjust the setting. The mouse roller button or the keyboard arrow keys can be used for fine adjustment. The indicator box will show any readings if they are relevant to the current step. If the device does not respond to the adjustments it can be caused by the wrong wiring connection.

5.1.3 Procedure and Methodology

The table below indicates the purpose in calibrating the analog device. Follow the steps as below. The software will also show a short description of the process and what to do. End the end make sure to save the new settings to the device.

Step	Type	Sub-System	Calibration Device	Description
1	Getting Ready	None	None	
2	10V Reference	Analog Output	Meter	Use 10V reference cable. This step is to set the reference voltage to the DACs. Ideally this should 10V, but DAC0 can be used to verify the voltage range. If DAC0 is below 9.9982V, increase this voltage.
3	A/D Unipolar Offset	Analog Input	None	Use analog input cable. Set this value so that the A/D voltage flickers between 0V and 306μV.
4	A/D Bipolar Offset	Analog Input	None	Use analog input cable. Set this value so that the A/D voltage flickers around 0V ±306μV.
5	A/D Range	Analog Input	Voltage Source	Use analog input cable. Input a voltage of 2.45V. Use the slider to set to the same voltage.
6	D/A 0 Offset	Analog Output	Meter	Use analog output cable. Set offset to 0V.
7	D/A 1 Offset	Analog Output	Meter	Use analog output cable. Set offset to 0V.
8	D/A 2 Offset	Analog Output	Meter	Use analog output cable. Set offset to 0V.
9	D/A 3 Offset	Analog Output	Meter	Use analog output cable. Set offset to 0V.
--	Second 16 channels	Goto step 2		For the A-version 32 channel device repeat steps 2 to 9
10	Save Calibration Values	All	All	Save data to device

Table 5-4 Calibration Procedure

5.1.4 Calibration Validity and Operating Conditions

The analog μDAQ and Rugged μDAQ device will perform as specified when operating under normal conditions as set in the specification appendix. However there are conditions where the device can behave outside these preset specifications. The following has an effect on the accuracy of the device. It would be good practice to recalibrate the device specifically for this environment.

No	Condition	Remedy
1	One year since last calibration.	It is advisable that the device be recalibrated every year if highest accuracy is required.
2	Harsh operating conditions.	If the unit operates in a harsh area, like factories, it is advisable that the unit recalibrated each year (1) and service every five (5) years
3	High/Low temperatures	Extreme temperature can effect to operation of the device. The identities of the analog circuit will certainly changes under extreme temperatures. The solution would be to recalibrate the unit within these conditions
4	Above average humidity	If the device operates in high humidity it can cause the unit to degrade in performance over time. The device needs to be calibrated and serviced more frequently.
5	Exceeding analog input/output specifications	If the device was driven outside its operating region it can affect the accuracy of the device. It would be best practice to recalibrate or in severe case to service the device.
6	Lightning strike	In the case of such an event the device need to be checked and tested by the manufacturer to prevent costly secondary damage.

7	Exceeding power input	If the power input was to high or of the wrong type the unit can be severely damaged
8	Extensive long storage	Make sure the device is stored in a static free environment inside the original packaging. If the device was stored for an extensive period it will need to be recalibrated.

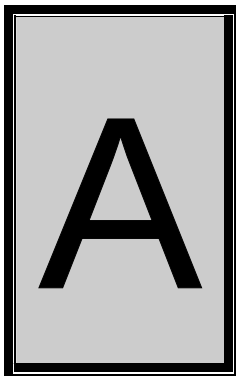
Table 5-5 Operating Conditions Voiding Calibration

5.2 Calibration - USB 73

If the USB device needs to be calibrated, the software can be found on the EDR Enhanced SDK CD-Rom. This application provides step-by-step information of how to calibrate your device. Make sure that you have a high precision calibration voltage source. This will help to configure your device more accurately.

5.2.1 Calibration Procedure – USB 73

1. Install the USB Calibration Software <EDRECD>\EDRE\APPS\USB-73 CAL
2. Run the USB Calibration Software.
3. Follow the step-by-step information on screen to tune your device.
4. Make sure to save the data to your device.



A Specifications

A.1 Digital Input/Output Characteristics

Number of Digital Channels:
Models are: USB, SRL, BT and Rugged

Device	Channels
24A	24
24C	24
48A	48
48C	48
72A	72
72C	72
96A	96
96C	96
120A	120
144C	144
168A	168
26	24
30	24
73	24

Number of Grouped Channels:
Models are: USB, SRL, BT and Rugged

Device	PPI Channels
24A	3
24C	3
48A	6
48C	6
72A	9
72C	9
96A	12
96C	12
120A	15
144C	18
168A	21
26	3
30	3
73	3

Compatibility: Intel® 82C55/TTL

D.C Characteristics – PPI 8255 Compatible Ports

Level	Min	Max
Input Low Voltage	-0.5V	0.8V
Input High Voltage	2.0V	5.0V
Output High Voltage	2.4V	
Output Low Voltage		0.45V
Output Current		2mA

A.2 Counter-Timer Characteristics

Number of Counter-Timer Channels:
Models are: USB, SRL, BT and Rugged

Device	Channels
24A	0
24C	6
48A	0
48C	6
72A	0
72C	6
96A	0
96C	6
120A	0
144C	6
168A	0
26	0
30	0
73	0

Resolution: 16-bits

Compatibility Intel® 82C54 / TTL

Clock Source
Software Selectable
1. Internal 2.5/10 MHz
2. External

Rev 1 – 2.5 MHz
Rev 2 – 10 MHz

Gate Source
Software Selectable
1. Software Controlled
2. External – Internal pull-up

Interrupt Source 6 x Terminal Count (TC).

I/O Characteristics

Level	Min	Max
Input Low Voltage	0V	0.8V
Input High Voltage	2.0V	5.25V
Low Level Input Current		- 100 uA
High Level Input Current		100 uA
Output High Voltage	2.4V	
Output Low Voltage		0.6V
Low Level Output Current		-24 mA
High Level Output Current		4 mA

A.3 Analog Output Characteristics - USB, SRL, BT & Rugged 30

A.3.1 A/B-Version

Number of Channels:	4
Resolution:	14-bits
Settling Time	2 microseconds
Maximum Update Rate – USB 1.1, USB 2.0	2 milliseconds
Maximum Update Rate – SRL @ 115K BAUD	4 milliseconds
Maximum Update Rate – BT @ 115K AUD	4 milliseconds
Data Transfer	Programmed I/O
Full Scale Error	±1 LSB (1.220 millivolts)
Zero Offset Error	±1/4 LSB (0.306 millivolts)
Output Drive	±5 milliamp
Load Characteristics	2 KΩ || 10 nF
Power On State	0 Volt

A.3.2 C-Version

Number of Channels:	4
Resolution:	16-bits
Settling Time	2 microseconds
Maximum Update Rate – USB 1.1, USB 2.0	2 milliseconds
Maximum Update Rate – SRL @ 115K BAUD	4 milliseconds
Maximum Update Rate – BT @ 115K AUD	4 milliseconds
Data Transfer	Programmed I/O
Full Scale Error	±1 LSB (1.220 millivolts)
Zero Offset Error	±1/4 LSB (0.306 millivolts)
Output Drive	±5 milliamp
Load Characteristics	2 KΩ || 10 nF
Power On State	0 Volt

A.4 Analog Input Characteristics – USB, SRL, BT & Rugged 26/30A/B

A.4.1 Input Characteristics

Number of Channels
16 Single Ended or 8 Differential – A/B/C16 Version
32 Single Ended or 16 Differential – A/B/C32 Version

Resolution
14-bits – A/B Version
16-bits – C Version

Maximum Update Rate – USB 1.1 (A-Version)
250 000 samples per second (S/s)

Maximum Update Rate – USB 2.0 (B-Version)
400 000 samples per second (S/s)

Maximum Update Rate – USB 1.1/2.0 (C-Version)
250 000 samples per second (S/s)

Maximum Update Rate – SRL @ 115K BAUD
3 000 samples per second (S/s)

Maximum Update Rate – BT @ 115K BAUD
3 000 samples per second (S/s)

Data Transfer – USB 1.1 (A-Version)
USB full speed isochronous transfers @ 512Kb/s – A Version

Data Transfer – USB 2.0 (B-Version)
USB high speed isochronous transfers @ 1024Kb/s

Data Transfer – SRL
EDR Enhanced Packet Transfer

Data Transfer – BT
EDR Enhanced Packet Transfer

Input Programmable Ranges – A/B-Version

Channel Gain	Unipolar Range	Bipolar Range
0.25	0-10V	±10V
0.50	0-10V	± 5V
1.00	0-5V	± 2.5V
2.50	0-2V	± 1V
5.00	0-1V	± 500 mV
10.00	0-500mV	± 250 mV
25.00	0-200mV	±100 mV
50.00	0-100mV	± 50 mV
100.00	0-50mV	± 25 mV

Input Programmable Ranges – C-Version

Channel Gain	Unipolar Range	Bipolar Range
1.0	NA	±10V

Input Coupling
DC

Maximum Working Voltage
±10V relative to analog ground

Over voltage protection
Power On State: -25V to +40V
Power Off State: -40V to +55V

Maximum Channel List Size
16

A.4.2 Conversion Characteristics

Maximum Conversion Rate
250 000 Samples per second – A/C Version
400 000 Samples per second – B Version

Converter Type
Successive approximation with sample and hold capacitors

Resolution
14-bits – A/B Version
16-bits – C Version

Relative Accuracy
±1 LSB

Gain x 1 Offset Error
±0.4 millivolts – A/B Version
±1.3 millivolts – C Version

Gain x 10 Offset Error
±0.6 millivolts – A/B Version

Gain x 100 Offset Error	±0.1 millivolts – A/B Version
Scale x 1/4 Offset Error	± 1 millivolts – A/B Version
Scale x 1/2 Offset Error	±0.5 millivolts – A/B Version

A.4.3 External Clock – EXT_CLK pin

Maximum Rate	250 000 Hz – A/C Version 400 000 Hz – B Version
Synchronization	Internal 20 MHz clock
Conversion	Falling Edge

A.4.4 External Gate – EXT_GATE pin

Enable Process	High Input (>2.4V DC)
Disabled Process	Low Input (<1.2V DC)

A.5 Thermocouple Input Characteristics

Number of Channels
Models are: USB, SRL, BT and Rugged

Device	Differential Channels
73T8	8
73T16	16
73T32	32

Resolution 14-bits

Maximum Update Rate – USB 1.1 & USB 2.0 2 milliseconds

Maximum Update Rate – SRL & BT 4 milliseconds

Data Transfer – USB 1.1 & USB 2.0 Programmed I/O – USB Bulk Transfer

Data Transfer – SRL Programmed I/O – EDR Enhanced Packet Transfer

Data Transfer - BT Programmed I/O – EDR Enhanced Packet Transfer

Input Programmable Ranges

Channel Gain	Bipolar Range
30	±83mV

Input Coupling DC

Relative Accuracy ±1 LSB

Gain x 30 Offset Error ±0.011 millivolts

A.6 Bus Interface

A.6.1 USB 1.1

Bus Type	Universal Serial Bus Revision 1.1
Bus Speed	USB Full Speed – 12 Mega bit per second.
Controller	USB Serial Interface Endpoint Compliant
Voltage	5V
Endpoints	1 x Control Transfer Endpoint 1 x Bulk Transfer Endpoint 1 x Interrupt Transfer Endpoint 1 x Isochronous Transfer Endpoint @ 512kB/s

A.6.2 USB 2.0

Bus Type	Universal Serial Bus Revision 2.0
Bus Speed	USB High Speed – 480 Mega bit per second.
Controller	USB Serial Interface Endpoint Compliant
Voltage	5V
Endpoints	1 x Control Transfer Endpoint 1 x Bulk Transfer Endpoint 1 x Interrupt Transfer Endpoint 1 x Isochronous Transfer Endpoint @ 512kB/s

A.6.3 Serial

Bus Type	Universal Asynchronous Receive Transmit (UART)
Bus Speed	300, 600, 1200, 4800, 9600, 19200, 28800, 38400, 57600, 115200 BAUD
Controller	UART compatible
Protocols	RS 232 & RS 485

A.6.4 Wireless Interface

Connection Type	Wireless Radio
Bus Speed	300, 600, 1200, 4800, 9600, 19200, 28800, 38400, 57600, 115200 BAUD
Interface	UART HCI Master Protocol Stack
Compliancy	Bluetooth V1.1 Compliant
Operation Type	Class 2
Reception Range	More than ten (10) meter
Antenna Type	External 2.4 GHz
Frequency Range	2.4 GHz – 2.4835 GHz
Transmitter Average Output Power f=2.441 GHz	0.5 dBm
Receiver Sensitivity (DH1/3/5) at 0.1% BER	-78 dBm

A.7 Power Requirements

A.7.1 USB Devices

Device	Minimum	Typical	Maximum	Power Source
USB 24A		50 mA	100 mA	USB Power
USB 24C		295 mA	345 mA	External PSU
USB 48A		55 mA	155 mA	USB Power
USB 48C		300 mA	400 mA	External PSU
USB 72A		60 mA	210 mA	USB Power
USB 72C		310 mA	460 mA	External PSU
USB 96A		65 mA	265 mA	External PSU
USB 96C		315 mA	515 mA	External PSU
USB 120A		70 mA	320 mA	External PSU
USB-62-16	60 mA		100 mA	External PSU
USB-62-32	60 mA		100 mA	External PSU
USB-63-16	60 mA		220 mA	External PSU
USB-63-32	60 mA		380 mA	External PSU
USB-69-16	60 mA		150 mA	External PSU
USB-69-16	60 mA		220 mA	External PSU
USB 26A16		450 mA	500 mA	External PSU
USB 26A32		850 mA	900 mA	External PSU
USB 30A16		480 mA	530 mA	External PSU
USB 30A32		880 mA	930 mA	External PSU
USB 26B16		450 mA	500 mA	External PSU
USB 26B32		850 mA	900 mA	External PSU
USB 30B16		480 mA	530 mA	External PSU
USB 30B32		880 mA	930 mA	External PSU
USB 26C16		450 mA	500 mA	External PSU
USB 26C32		500 mA	550 mA	External PSU
USB 30C16		480 mA	530 mA	External PSU
USB 30C32		530 mA	580 mA	External PSU
USB 73T8		400 mA	450 mA	External PSU
USB 73T16		420 mA	470 mA	External PSU
USB 73T32		820 mA	870 mA	External PSU

A.7.2 SRL Devices

Device	Minimum	Typical	Maximum	Power Source
SRL 24A		60 mA	110 mA	USB Power
SRL 24C		305 mA	355 mA	External PSU
SRL 48A		65 mA	165 mA	USB Power
SRL 48C		310 mA	410 mA	External PSU
SRL 72A		70 mA	220 mA	USB Power
SRL 72C		320 mA	470 mA	External PSU
SRL 96A		75 mA	275 mA	External PSU
SRL 96C		325 mA	525 mA	External PSU
SRL 120A		80 mA	330 mA	External PSU
SRL-62-16	70 mA		110 mA	External PSU
SRL-62-32	70 mA		110 mA	External PSU
SRL-63-16	70 mA		230 mA	External PSU
SRL-63-32	70 mA		380 mA	External PSU
SRL-69-16	70 mA		160 mA	External PSU
SRL-69-16	70 mA		230 mA	External PSU
SRL 26A16		460 mA	510 mA	External PSU
SRL 26A32		860 mA	910 mA	External PSU
SRL 30A16		490 mA	540 mA	External PSU
SRL 30A32		890 mA	940 mA	External PSU
SRL 73T8		410 mA	460 mA	External PSU
SRL 73T16		430 mA	480 mA	External PSU
SRL 73T32		830 mA	880 mA	External PSU

A.7.3 BT Devices

Device	Minimum	Typical	Maximum	Power Source
BT 24A		75 mA	125 mA	USB Power
BT 24C		320 mA	370 mA	External PSU
BT 48A		80 mA	180 mA	USB Power
BT 48C		325 mA	425 mA	External PSU
BT 72A		85 mA	235 mA	USB Power
BT 72C		335 mA	485 mA	External PSU
BT 96A		90 mA	290 mA	External PSU
BT 96C		340 mA	540 mA	External PSU

BT 120A		95 mA	345 mA	External PSU
BT-62-16	85 mA		125 mA	External PSU
BT-62-32	85 mA		125 mA	External PSU
BT-63-16	85 mA		245 mA	External PSU
BT-63-32	85 mA		405 mA	External PSU
BT-69-16	85 mA		175 mA	External PSU
BT-69-16	85 mA		245 mA	External PSU
BT 26A16		475 mA	525 mA	External PSU
BT 26A32		875 mA	925 mA	External PSU
BT 30A16		505 mA	555 mA	External PSU
BT 30A32		905 mA	955 mA	External PSU
BT 73T8		425 mA	475 mA	External PSU
BT 73T16		445 mA	495 mA	External PSU
BT 73T32		845 mA	895 mA	External PSU

A.8 Environmental / Physical

A.8.1 µDAQ (Commercial)

Relative Humidity	0% to 90% (non-condensing)
Operating Temperature	0°C to 70°C
Housing Type	Plastic Casing
Dimensions – 2 Tier Box	Height: 45mm Width: 80mm Length: 148mm
Dimensions – 3 Tier Box	Height: 60mm Width: 80mm Length: 148mm

A.8.2 Rugged µDAQ (Industrial)

Relative Humidity	0% to 90% (non-condensing)
Operating Temperature	0°C to 70°C
Housing Type	Powder Coated Aluminium Extrusion
Mounting Type	Rubber vibration feet or DINRAIL
Dimensions	Height: 107mm Width: 204mm Length: 154mm

A.9 Power Supplies and Battery Packs

A.9.1 Battery Pack & Charger– Rugged µDAQ (Industrial)

Type	Nickel metal hydride (NiMH)
Capacity	3700 mAh
Charge Time – Fast	3 Hours
Charge Time – Slow	13 Hours
Location	Built-In
Indicators	Fast/Slow Charge & Activity

A.9.2 AC Power Supply Unit (PSU) – µDAQ (Commercial)

Input Type	AC
Input Range	110V – 240V
Input Connector Type	Multiple to comply with all international wall sockets
Power	DC
Output Type	DC
Output Voltage	9V
Maximum Output Current	1A
Output Connector Type	5mm DC plug
Regulator Type	Switch Mode
Location	External/Separate
Indicators	Power

A.9.3 DC Power Supply Unit (PSU) – µDAQ (Commercial)

Input Type	DC
Input Range	12V-26V
Input Connector Type	In car lighter plug
Output Type	DC
Output Voltage	9V
Maximum Output Current	2.3A
Output Connector Type	5mm DC plug
Regulator Type	Analog
Location	External/Separate
Indicators	Power

A.9.4 AC Power Supply Unit (PSU) – Rugged µDAQ (Industrial)

Input Type	AC
Input Range	90V-264V
Input Connector Type	3 Pin Industrial
Maximum Power	60W
Output Type	DC
Output Voltage	15V
Output Connector Type	µDAQ Bus
Regulator Type	Switch mode
Location	Internal
Indicators	Power

A.9.5 DC Power Supply Unit (PSU) – Rugged µDAQ (Industrial)

Input Type	DC
Input Range	18V-40V
Input Connector Type	3 Pin Industrial
Output Type	DC
Output Voltage	15V
Maximum Output Current	4A
Output Connector Type	µDAQ Bus
Regulator Type	Analog
Location	Internal
Indicators	Power

B

B Related Products and Accessories

A whole range of external connectable accessories and application modules supports the μDAQ & Rugged μDAQ series. These modules are easy to connect too and support DINRAIL installations. Below is a list of products that are compatible with the μDAQ and Rugged μDAQ devices. They are broken up sub-sections to cover each type of application.

B.1 General Adapters

There are general adapters that can be used with the μDAQ and Rugged μDAQ devices. These modules simple map a cable-connector to a screw terminal block.

Adapter	Entry Connector	Cable	Possible Usage
ADPT-2526	DB25 Female	DB25M/F	To map the main connector to a screw terminal clock. This adapter is used to make simple connections to the unit.
ADPT-20	IDC 20	IDC 20	To connect to digital I/O port A-B on the PC43A2. To connect 8 channels on the PC52A2
ADPT-910	IDC 10 DB9 Female	IDC 10 DB9 Female	For connecting to digital I/O ports or analog channels on the PC43A2 and PC52A2. Also for connecting to the DB9 on the serial units.

Table 5-6 General Adapters

B.2 Analog I/O Application Modules

All analog I/O application modules are DINRAIL mountable and use standard connection cables. These modules cover a wide application range. Care must be taken to select the appropriate module to suite the application. The application modules are also fitted with screw terminals for easy installation.

Adapter	Entry Connector	Cable	Application
PC-52-2	IDC10	IDC10	2CH 5B carrier module
PC-52-4	IDC10	IDC10	4CH 5B carrier module
PC-52-8	IDC10	IDC10	8CH 5B carrier module

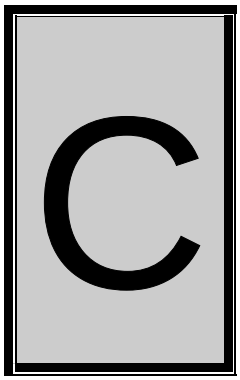
Table 5-7 Analog I/O Application Modules

B.3 Digital I/O Application Modules

All Digital I/O application modules are DINRAIL mountable. They are supplied with standard cables and are easy to connect too. Wide variety is available and care must be taken to select the correct module for the application.

Adapter	Entry Connector	Cable	Application
PC-37 Series	IDC10	IDC10	Opto 22 solid state relay modules
	IDC20	IDC20	
	DB25	DB25M/F	
PC-51-8	IDC10	IDC10	8CH Opto 22 G4 Carrier module
PC-38G	IDC10	IDC10	8CH Electro-Mechanical relay module
PC-38H	IDC10	IDC10	8CH Reed relay module
PC-38V	IDC10	IDC10	8CH TTL Driver module
PC-38W	IDC20	IDC20	16CH TTL Driver module
PC-38X	DB25	DB25M/F	24CH TTL Driver module
PC-43B	IDC20	IDC20	16CH Opto-Isolated inputs
PC-43C	DB25	DB25F/M	24CH Opto-Isolated inputs
PC-43E	IDC10	IDC10	8CH Opto-Isolated inputs

Table 5-8 Digital I/O Application Modules



C Configuration Constants

C.1 Query Codes

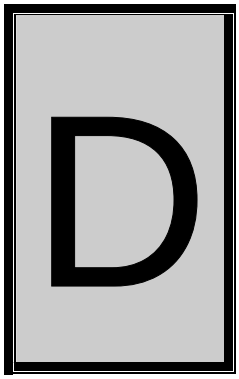
Name	Value	Description
APIMAJOR	1	Query EDRE API major version number.
APIMINOR	2	Query EDRE API minor version number.
APIBUILD	3	Query EDRE API build version number.
APIOS	4	Query EDRE API OS type.
APINUMDEV	5	Query number of devices installed.
BRDTYPE	10	Query a board's type.
BRDREV	11	Query a board's revision.
BRDYEAR	12	Query a board's manufactured year.
BRDMONTH	13	Query a board's manufactured month.
BRDDAY	14	Query a board's manufactured day.
BRDSERIALNO	15	Query a board's serial number.
DRVMAJOR	20	Query a driver's major version number.
DRVMINOR	21	Query a driver's minor version number.
DRVBUILD	22	Query a driver's build version number.
ADNUMCHAN	100	Query number of ADC channel.
ADNUMSH	101	Query number of samples-and-hold channels.
ADMAXFREQ	102	Query maximum sampling frequency.
ADBUSY	103	Check if ADC system is busy.
ADFIFOSIZE	104	Get ADC hardware FIFO size.
ADFIFOOVER	105	Check for FIFO overrun condition.
ADBUFFSIZE	106	Check software buffer size.
ADBUFFOVER	107	Check for circular buffer overrun.
ADBUFFALLOC	108	Check if software buffer is allocated.
ADUNREAD	109	Get number of samples available.
ADEXTCLK	110	Get status of external clock line – PCI30FG.
ADEXTTRIG	111	Get status of external trigger line – PCI30FG.
ADBURST	112	Check if burst mode is enabled.
ADRANGE	113	Get ADC range.
DANUMCHAN	200	Query number of DAC channels.
DAMAXFREQ	201	Query maximum DAC output frequency.
DABUSY	202	Check if DAC system is busy.
DAFIFOSZ	203	Get DAC FIFO size.
CTNUM	300	Query number of counter-timer channels.
CTBUSY	301	Check if counter-timer system is busy.
DIONUMPORT	400	Query number of digital I/O ports.
DIOQRYPORT	401	Query a specific port for capabilities.
DIOPORTWIDTH	402	Get a specific port's width.
INTNUMSRC	500	Query number of interrupts sources.
INTSTATUS	501	Queries interrupt system's status.
INTBUSCONNECT	502	Connect interrupt system to bus.
INTISAVAILABLE	503	Check if an interrupt is available.
INTNUMTRIG	504	Check number times interrupted

C.2 Error Codes

Name	Value	Description
EDRE_OK	0	Function successfully.
EDRE_FAIL	-1	Function call failed.
EDRE_BAD_FN	-2	Invalid function call.
EDRE_BAD_SN	-3	Invalid serial number.
EDRE_BAD_DEVICE	-4	Invalid device.
EDRE_BAD_OS	-5	Function not supported by operating system.
EDRE_EVENT_FAILED	-6	Wait on event failed.
EDRE_EVENT_TIMEOUT	-7	Event timed out.
EDRE_INT_SET	-8	Interrupt in use.
EDRE_DA_BAD_RANGE	-9	DAC value out of range.
EDRE_AD_BAD_CHANLIST	-10	Channel list size out of range.
EDRE_BAD_FREQUECY	-11	Frequency out of range.
EDRE_BAD_BUFFER_SIZE	-12	Data passed by buffer incorrectly sized
EDRE_BAD_PORT	-13	Port value out of range.
EDRE_BAD_PARAMETER	-14	Invalid parameter value specified.
EDRE_BUSY	-15	System busy.
EDRE_IO_FAIL	-16	IO call failed.
EDRE_BAD_ADGAIN	-17	ADC-gain out of range.
EDRE_BAD_QUERY	-18	Query value not supported.
EDRE_BAD_CHAN	-19	Channel number out of range.
EDRE_BAD_VALUE	-20	Configuration value specified out of range.
EDRE_BAD_CT	-21	Counter-timer channel out of range.
EDRE_BAD_CHANLIST	-22	Channel list invalid.
EDRE_BAD_CONFIG	-23	Configuration invalid.
EDRE_BAD_MODE	-24	Mode not valid.
EDRE_HW_ERROR	-25	Hardware error occurred.
EDRE_HW_BUSY	-26	Hardware busy.
EDRE_BAD_BUFFER	-27	Buffer invalid.
EDRE_REG_ERROR	-28	Registry error occurred.
EDRE_OUT_RES	-29	Out of resources.
EDRE_IO_PENDING	-30	Waiting on I/O completion

C.3 Digital I/O Return Query Codes

Name	Value	Description
DIOOUT	0	Port is an output.
DIOIN	1	Port is an input.
DIOINOROUT	2	Port can be configured as in or out.
DIOINANDOUT	3	Port is an input and an output.



D Ordering Information

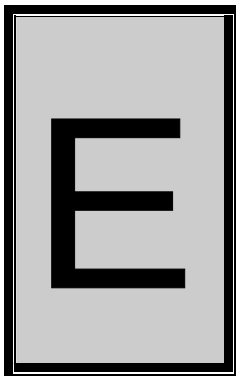
For ordering information please contact Eagle Technology directly or visit our website www.eagledaq.com. They can also be emailed at eagle@eagle.co.za.

Device	USB 1.1	USB 2.0	SRL	BT	µDAQ	Rugged µDAQ	Description
24A	✓		✓	✓	✓	✓	24 channel digital I/O device
48C	✓		✓	✓	✓	✓	24 channel digital I/O & counter-timer device
72A	✓		✓	✓	✓	✓	72 channel digital I/O device
96C	✓		✓	✓	✓	✓	96 channel digital I/O & counter-timer device
120A	✓		✓	✓	✓	✓	120 channel digital I/O device
120C							120 channel digital I/O device & counter-timer device
144A	✓		✓	✓		✓	144 channel digital I/O device
144C	✓		✓	✓		✓	144 channel digital I/O device & counter-timer device
168A	✓		✓	✓		✓	168 channel digital I/O device
26A16	✓		✓	✓	✓	✓	16 channel analog input device @ 250 KS/s
26A16-BNC	✓		✓	✓	❶	❶	16 channel analog input device @ 250 KS/s
26A32	✓		✓	✓	✓	✓	32 channel analog input device @ 250 KS/s
30A16	✓		✓	✓	✓	✓	16 channel & 4 channel analog input/output device @ 250 KS/s
30A16-BNC	✓		✓	✓	❶	❶	16 channel & 4 channel analog input/output device @ 250 KS/s
30A32	✓		✓	✓	✓	✓	32 channel & 8 channel analog input/output device @ 250 KS/s
26B16		✓			✓	✓	16 channel analog input device @ 400 KS/s
26B16-BNC		✓			❶	❶	16 channel analog input device @ 400 KS/s
26B32		✓			✓	✓	32 channel analog input device @ 400 KS/s
30B16		✓			✓	✓	16 channel & 4 channel analog input/output device @ 400 KS/s
30B16-BNC		✓			❶	❶	16 channel & 4 channel analog input/output device @ 400 KS/s
30B32		✓			✓	✓	32 channel & 8 channel analog input/output device @ 400 KS/s
26C16	✓	✓			✓	✓	16 channel analog input device @ 250 KS/s
26C16-BNC	✓	✓			❶	❶	16 channel analog input device @ 250 KS/s
26C32	✓	✓			✓	✓	32 channel analog input device @ 250 KS/s
30C16	✓	✓			✓	✓	16 channel & 4 channel analog input/output device @ 250 KS/s
30C16-BNC	✓	✓			❶	❶	16 channel & 4 channel analog input/output device @ 250 KS/s
30C32	✓	✓			✓	✓	32 channel & 8 channel analog input/output device @ 250 KS/s
73T8	✓		✓	✓	✓		8 Channel Thermocouple input device
73T16	✓		✓	✓	✓	✓	16 Channel Thermocouple input device
73T32	✓		✓	✓	✓	✓	32 Channel Thermocouple input device
73R8					❶	❶	8 Channel RTD input device
73R16					❶	❶	16 Channel RTD input device
73R32					❶	❶	32 Channel RTD input device
30A16-73T16	✓		✓	✓	✓	✓	16 channel & 4 channel analog input/output device and 16 Channel Thermocouple input device
30A16-73T32	✓		✓	✓		✓	16 channel & 4 channel analog input/output device and 32 Channel Thermocouple input device
30A32-73T16	✓		✓	✓		✓	32 channel & 8 channel analog input/output device and 16 Channel Thermocouple input device
30B16-73T16		✓					16 channel & 4 channel analog input/output device and

30B16-73T32		✓					16 Channel Thermocouple input device 16 channel & 4 channel analog input/output device and 32 Channel Thermocouple input device
30B32-73T16		✓					32 channel & 8 channel analog input/output device and 16 Channel Thermocouple input device

Table D-1 µDAQ & Rugged µDAQ Ordering Information

❶ Future product



E Troubleshooting

This section deals with common errors that can be experienced with your USB unit. It does not list all errors, but will be updated to contain the most recent information. Please browse through the different cases and try the solution before contacting customer support. However if the problem still occurs email you local supplier or Eagle Technology eagle@eagle.co.za.

E.1 Case 1 - USB Streaming Errors

E.1.1 Devices Affected

USB 26/30 – Usb 1.1

E.1.2 Operating Systems

Microsoft® Windows™ 2000, Microsoft® Windows™ XP

E.1.3 Description

USB transfer errors occur when data is transferred to and from the unit, but the resulting data is wrong. This occurs when streaming analog data from the unit, in particular the high-speed analog I/O units. The slave USB controller inside the µDAQ unit makes use of Isochronous transfers which contains the least amount of error checking by the USB host and slave connection.

E.1.4 Symptoms

To detect the problem a constant voltage needs to be supplied to one of the channels. Start scanning that particular channel by making use a software program such as WaveView for Windows™. The constant voltage level will be broken every so often. This indicates an error in the data, but the USB sub-system did not notify the USB µDAQ driver of this condition.

E.1.5 Solution

There can be three possible sources to the problem. The first is on older PCs, or even recent motherboards the BIOS needs to be updated. The second is the driver for the USB host on the motherboard can be problematic. Please contact your local supplier or visit the particular brand's World Wide Web site for the latest updates. Normally these web sites have a list of errors that the BIOS update or the driver will fix. The last problem is with the OS. Visit the Microsoft® Windows™ operating system web site for the latest updates and service packs for your particular version.

Document History

The table below lists the document history. A minor revision change will indicate document errors that are edited. A major revision change will indicate an update or change to the document contents or structure.

Revision	Date	Comments
1.0	30/07/2004	Original Release. Most data got copied from the USB & Serial μDAQ manual.
1.2	15/03/2005	Add support for the Opto-isolated and Reed relay devices.
1.3	13/04/2005	Add support for the C (16-bit) analog I/O devices.