NET µDAQ Series Digital, Counter, & Analog I/O

User's Manual

Eagle Technology – Cape Town, South Africa

www.eagledaq.com

Net µDAQ series 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, does not take responsibility or liability of consequential damages, project delays, damaging of equipment or capital loss as a result of its products.

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

TABLE OF CONTENTS

Table of Figures

Table of Tables

1. Introduction

The NET μDAQ series are Ethernet type data acquisition devices. It makes use of TCP / IP over Ethernet and the EDR Enhanced Remote Device protocol connecting at 10 mega bits per second or 100 mega bits per second network. The protocols are built into the EDR Enhanced application interface and the user requires no knowledge of network programming. The NET μDAQ series has built-in Ethernet networking capabilities and TCP / IP stack. This means that the device can be connected to any existing Ethernet network.

Features

- **Ethernet 10/100 Mbps compatible.**
- **Automatic link speed detection.**
- **Support remote setup and firmware upgrade.**
- **No knowledge of network programming is needed**
- **Powered by an external power supply**
- **TTL compatible digital I/O ports.**
- **Quick and effortlessly to install.**

Applications

The NET μDAQ series can be used in the following applications:

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

Key Specifications

- **Support for up to 32 16-bit resolution analog input channels, ±10 volt**
- **Support for up to 4 16-bit resolution analog outputs channels, ±10 volt**
- **Support for up to 120 TTL compatible digital I/O channels**
- **Analog input sampling @ 250Khz via channel list**

Software Support

The NET µDAQ series is supported by EDRE SDK 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.

Contact Details

Below are the contact details of Eagle Technology.

Eagle Technology
PO Box 4376
Cape Town
8000
South Africa
Telephone +27 (021) 423 4943
Fax +27 (021) 424 4637
E-Mail eagle@eagle.co.za
Website http://www.eagledaq.com

2

2. Installation

This chapter describes how to install and configure the NET μDAQ device for the first time. Minimal configuration is necessary; almost all settings are done through software. The operating system will take care of all resource assignments.

Package

μDAQ package will contain the following:

- **NET μDAQ device.**
- **9V External power supply.**
- **Crossover Ethernet cable.**
- **Software CD-Rom.**

Operating System Support

The NET μDAQ makes use of TCP / IP over Ethernet and the EDR Enhanced Remote Device protocol connecting at 10 mega bits per second or 100 mega bits per second network. The protocols are built into the EDR Enhanced application interface (API).

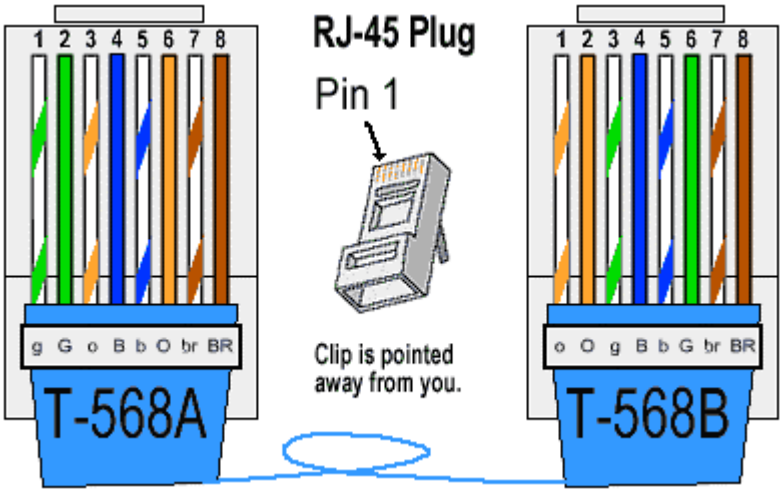
Board Type	Revision	Operating Systems	Driver Type
NET μDAQ	Revision 1	Windows 2000/XP/Vista/Linux	API

Table 2-1 Operating System Support

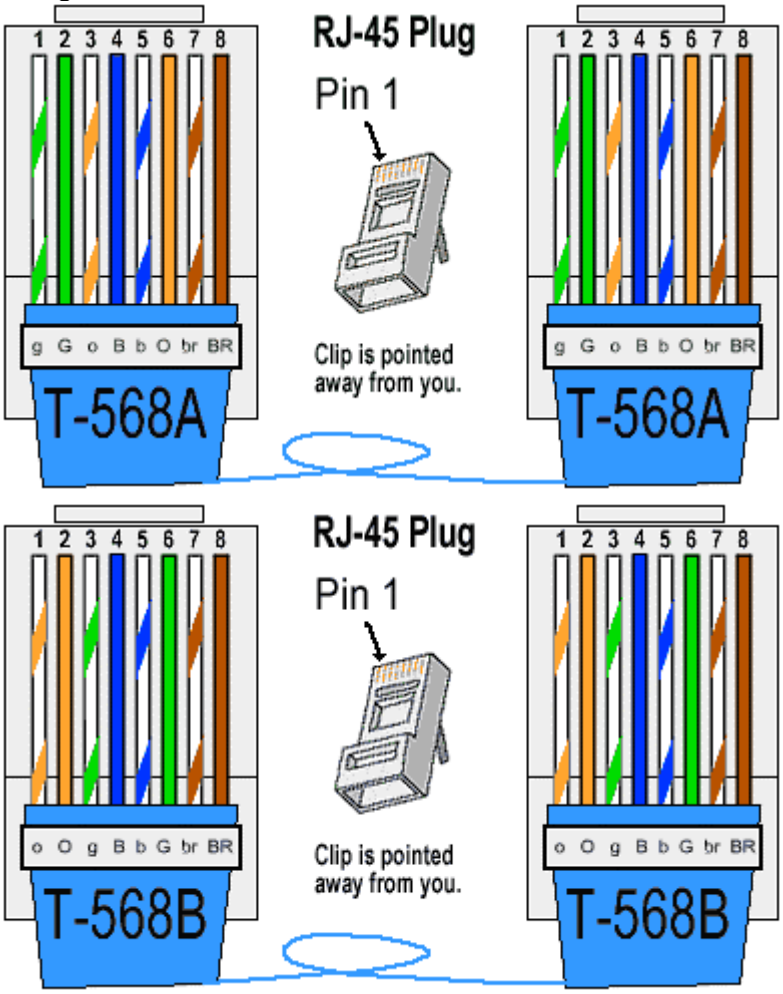
Installation

This section will describe how to connect your device to your computer.

- **When connecting the device directly to a computer a RJ-45 crossover Ethernet cable should be used.**



- **When connecting the device through a hub or switch a T-568A or T568B straight-through Ethernet cable should be used.**



Post Connection Installation

Run *edreapi.exe* found on the Eagle CD-Rom and follow the on screen instructions. Edreapi.exe will install activex controls and libraries needed by applications controlling the hardware. (Normally located in <CD-ROM>:\EDRE\API directory).

To add the NET µDAQ device to your system open the *EDR Enhanced Setup* applet (found in control panel) that was installed when you installed *edreapi.exe*. Select the TCP/IP tab and click on add. The default IP address of the device is *192.168.0.1* (yours may have been changed) and should be entered into the *IP Address* box. The port used will always be *7070*. You can now click on the *OK* button and select *OK* to exit the application.

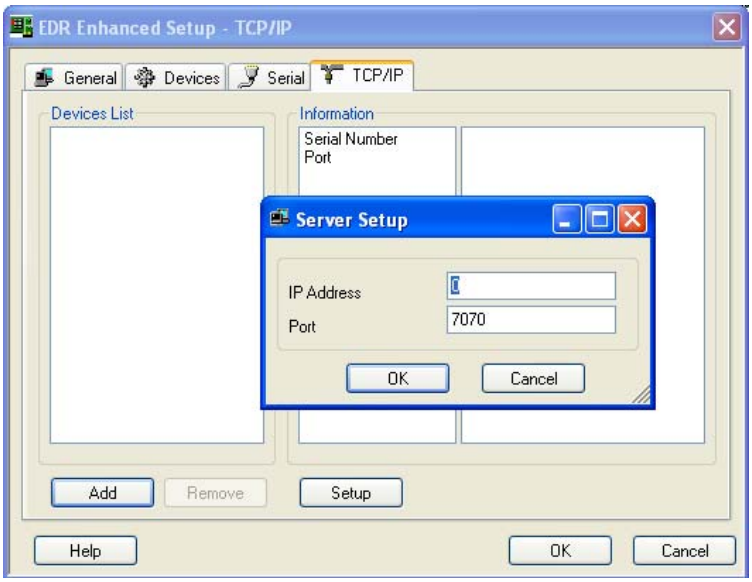


Figure 2-1 EDR Enhanced Setup - Ethernet

You might have to add the device IP address to the *hosts* file. See *<WINDOWS_DIR>\system32\drivers\etc* directory and add the NET µDAQ device IP address and name to the list.

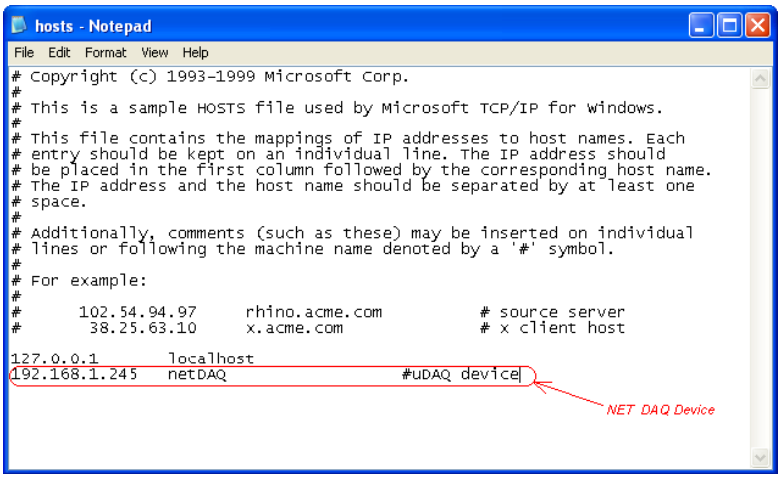


Figure 2-2 *hosts* file

You can now verify that installation was successful by opening the *EDR Enhanced Setup* applet again. Select the *Devices* tab. This dialog should list all installed hardware. Verify your device properties on this dialog. See picture below

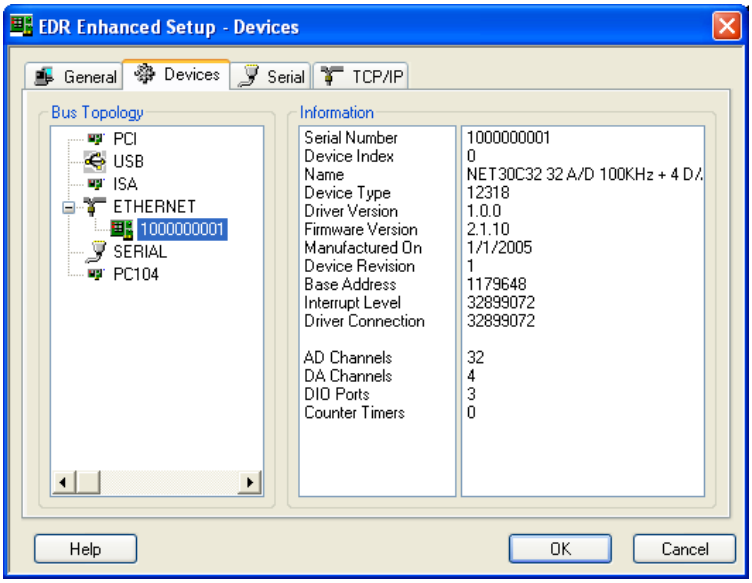


Figure 2-3 EDR Enhanced Setup - Devices

If the NET µDAQ device is not in the list please refer to *Figure 2-2* and add the IP address in the *hosts* file.

Reset IP Address

If you have forgotten what the IP address of the device is you can do an IP reset. This will reset the IP address back to *192.168.0.1*.

Reset procedure

- **Disconnect power to the unit.**
- **Use a thin, round object like a pen to press the button at the bottom of the device.**
- **Do not press the button before the power is connected.**
- **You need to press the button within 0.5 seconds after connecting the 9V power supply that came with the device.**
- **You will see the active LED flash twice and then stay on.**
- **Keep pressing the button for 3 seconds from when the power was applied.**
- **Remove the power and start the installation as explained in the previous, *installation*, section.**

Rugged Unit

- **Disconnect power to the unit.**
- **See Rugged Net μDAQ CAN bus connector PIN 1 & 4.**
- **Do not short circuit the pins before the power is connected.**
- **You need short circuit the pins within 0.5 seconds after connecting the 9V power supply that came with the device.**
- **You will see the active LED flash twice and then stay on.**
- **Keep the short circuit for 3 seconds from when the power was applied.**
- **Remove the power and start the installation as explained in the previous, *installation*, section.**

Configure IP address

Before you can configure the IP address first complete the installation as explained in the *installation* section and install the *Net µDAQ Setup Tools* (see *Application Software* section). If this has been done successfully, open the *NET-uDAQ Setup* application.

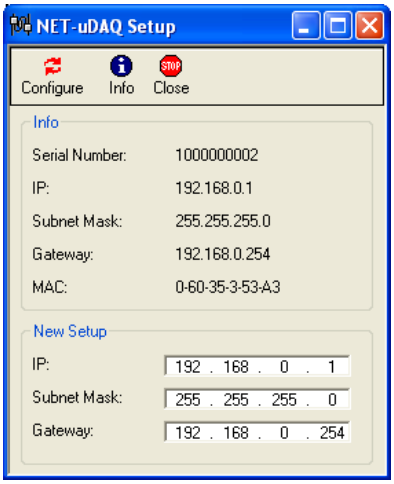


Figure 2-4 NET-uDAQ Setup Application

Configuration procedure

- **You can now edit the appropriate parameters and *click "Configure"* and close the application**
- **Before disconnection the device you should open the *"EDR Enhanced Setup"* in the control panel and remove the device from the list of Ethernet devices.**
- **Disconnect the power.**
- **Connect the power.**
- **Add the device to the *"EDR Enhanced Setup"* applet as explained in the *Post Connection Installation* using the new IP address.**

Application Software

NET µDAQ Setup Tools

Software tool for the NET µDAQ is also supplied, run *setup.exe* found on the CD-ROM and is normally located in the "<CD-ROM>:\Edre\apps\NET_uDAQ_Tools" directory.

The tools include two applications:

1. ***NET-uDAQ Setup* will allow you to configure the device's IP address.**
2. ***NET-uDAQ Firmware Programmer* is used to update the device firmware.**

Wave View for Windows

The EDR Enhanced Software Development Kit CD-Rom comes with WaveView for Windows™. WaveView has support for Analog Inputs, Analog Outputs, Digital I/O.

WaveView can be found on the EDR Enhanced CD-Rom. (Normally located in <CD-ROM>:\EDRE\APPS\WVFW2 directory)

3

3. Interconnections

The NET μDAQ series has connectors for digital I/O, counter-timers and analog I/O. The NET μDAQ make use of only one connector type; a DB25 male. To inter-connect to application modules there are adapters available. A cable is used to connect to these modules. Screw terminals are also available for quick installations.

Pin Assignments

NET μDAQ DIO – DB25 (M)

The table below shows the pin assignments for the DB25(M) digital I/O connectors found on the NET μDAQ DIO A and C versions.

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 3-1 NET μDAQ DIO Connector – DB25 (M)

NET μDAQ CT – DB25 (M)

The table below shows the pin assignments for the DB25(M) counter timer connector found on the NET μDAQ CT.

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 3-2 NET μDAQ CT Connector – DB25 (M)

NET μDAQ Analog I/O – DB25 (M)

The table below shows the pin assignments for the DB25(M) analog I/O connectors found on the NET 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 3-3 NET μDAQ Analog I/O (CH0 – CH15) – DB25 (M)

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 3-4 NET μDAQ Analog I/O (CH16 – CH31) – DB25

NET μDAQ Temperature Input – DB25 (M)

The table below shows the pin assignments for the DB25(M) temperature input connectors found on the NET 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 3-5 NET μDAQ Temperature Input – DB25 (M)

Rugged μNet DAQ CAN bus connector

The table below show the pin assignment for the DB9(M) CAN bus connector that is only available on the rugged version of the NET μDAQ device.

Pin	Name
1	IP_DEFAULT
2	CAN_L
3	CAN_GND
4	IP_GND
5	N/C
6	N/C
7	CAN_H
8	N/C
9	CAN_V+

Table 3-6 Rugged NET μDAQ CAN bus - DB9 (M)

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
EXT_GATE	External Gate
10V_REFCAL	10 Volt Reference Calibration
NC	Not Connected
IP_DEFAULT	Pins used to reset IP address on the rugged units
IP_GND	
CAN_L	Low bus output
CAN_GND	CAN bus ground
CAN_H	High bus output
CAN_V+	3.3V CAN bus voltage

Table 3-7 Signal definitions

Pin Descriptions

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.

External Gate (GATE_EXT)

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

External Clock (CLK_EXT)

These lines are used to externally clock the counters.

Output (Out)

These are the outputs of each counter-timer.

Digital Ground (DGND)

All digital ground signals should be connected to this pin.

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.

Analog Inputs (ACH0-31)

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).

Analog Outputs (DAC0-3)

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

External Clock (EXT_CLK)

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

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.

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.

WARNING: Do not connect to the 10Volt Reference Pin.

Application Modules & Accessories

The Ethernet devices support a wide range of standard applications modules. These application modules can help to simply or easily duplicate installation that can save plenty 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 optically isolation are also available.

4. Programming Guide

The NET μDAQ series is supplied with a complete software development kit. EDR Enhanced (EDRE SDK) comes with drivers for many operating systems and a common application program interface (API). The API serves as a interface layer between the control application and the TPC/IP protocol. The EDRE API makes it possible to write an application that can be used on all hardware with common sub-systems.

EDR Enhanced API

The EDR Enhanced SDK comes with both ActiveX controls and a Windows DLL API. Examples are provided in many different languages and serve as tutorials. EDRE is also supplied with a software manual and user's guide.

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

Digital Inputs/Outputs

Depending on the version that you have the NET μDAQ device can have up to 120 digital lines. Please refer to your particular version for specific details.
Please refer to 82C55 datasheet for DIO port characteristics.

Port	82C55 PPI No	Software assigned number	Port width	Description
NET 24A/CA				
A	0	0	8-bits	Port A
B	0	1	8-bits	Port B
C	0	2	8-bits	Port C
NET 48A/C				
A	0	0	8-bits	Port A
B	0	1	8-bits	Port B
C	0	2	8-bits	Port C
A	1	3	8-bits	Port A
B	1	4	8-bits	Port B
C	1	5	8-bits	Port C
NET 72A/C				
A	0	0	8-bits	Port A
B	0	1	8-bits	Port B
C	0	2	8-bits	Port C
A	1	3	8-bits	Port A
B	1	4	8-bits	Port B
C	1	5	8-bits	Port C
A	2	6	8-bits	Port A
B	2	7	8-bits	Port B
C	2	8	8-bits	Port C
NET 96A/C				
A	0	0	8-bits	Port A
B	0	1	8-bits	Port B
C	0	2	8-bits	Port C
A	1	3	8-bits	Port A
B	1	4	8-bits	Port B
C	1	5	8-bits	Port C
A	2	6	8-bits	Port A
B	2	7	8-bits	Port B
C	2	8	8-bits	Port C
A	3	9	8-bits	Port A
B	3	10	8-bits	Port B
C	3	11	8-bits	Port C
NET 120A				
A	0	0	8-bits	Port A
B	0	1	8-bits	Port B
C	0	2	8-bits	Port C
A	1	3	8-bits	Port A
B	1	4	8-bits	Port B
C	1	5	8-bits	Port C
A	2	6	8-bits	Port A
B	2	7	8-bits	Port B
C	2	8	8-bits	Port C
A	3	9	8-bits	Port A
B	3	10	8-bits	Port B
C	3	11	8-bits	Port C
A	4	12	8-bits	Port A
B	4	13	8-bits	Port B
C	4	14	8-bits	Port C
NET 26/30				
A	0	0	8-bits	Port A
B	0	1	8-bits	Port B
C	0	2	8-bits	Port C
NET 73R/T				
A	0	0	8-bits	Port A
B	0	1	8-bits	Port B
C	0	2	8-bits	Port C
NET 30C16/32-73T/R				
A	0	0	8-bits	Port A
B	0	1	8-bits	Port B
C	0	2	8-bits	Port C

Table 4-1 NET μDAQ Digital I/O Port Assignments

Reading the Digital Inputs

A single call is necessary to read a digital I/O port.

API-CALL
Long EDRE_DioRead(ulng Sn, ulng Port, ulng *Value)
The serial number, port, and a pointer to variable to hold the result must be passed by the calling function. A return code will indicate if any errors occurred.

ACTIVEX CALL
Long EDREDioX.Read(long Port)
Only the port-number needs to be passed and the returned value will either hold an error or the value read. If the value is negative an error did occur.

Writing to the Digital Outputs

A single call is necessary to write to a digital I/O port.

API-CALL
Long EDRE_DioWrite(ulng Sn, ulng Port, ulng Value)
The serial number, port, and a value must be passed by the calling function. A return code will indicate if any errors occurred.

ACTIVEX CALL
Long EDREDioX.Write(long Port, long Value)
The port number and value to be written needs to be passed and the returned value holds an error or the value read. If the value is negative an error did occur.

Counter

The counter sub-system is supported by functions to Write, Read, Configure and controlling the gate. There are 6 counters. Counter-timers are only supported by the NET 24C, NET 48C, NET 72C and NET 96C. The table below shows the relation of the counters and their software assigned numbers.
Please refer to 82c54 datasheet for more information regarding the counter-timers.

Counter	Software 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

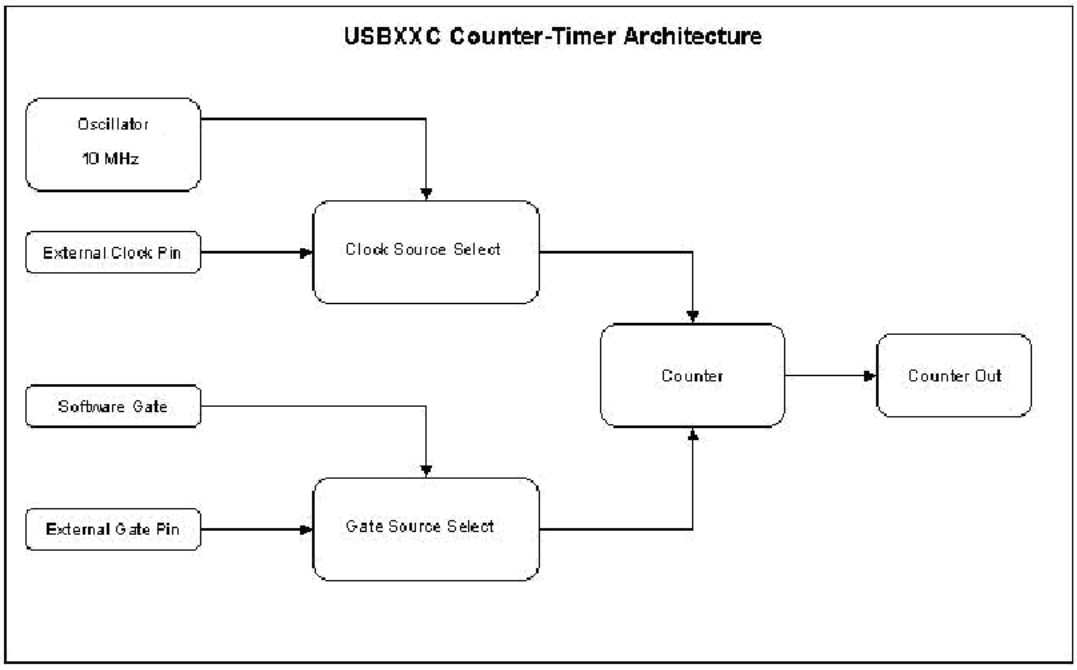


Figure 4-1 Counter-Timer Architecture

Architecture

The clock source and gate 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 as well. The external gate pin has a pull-up resistor, which allows the gate to be enabled when configured to external.

Writing the initial counter value

A single call is necessary to write a counter's initial load value.

API-CALL
Long EDRE_CTWrite(ulng Sn, ulng Ct, ulng Value)
The serial number, counter-number, and a value must be passed by the calling function. A return code will indicate if any errors occurred.

ACTIVEX CALL
Long EDRECTX.Write(long Port, ulng Value)

The counter-number and a value must be passed by the calling function. A return code will indicate if any errors occurred.

Reading the counter value

A single call is necessary to read a counter.

API-CALL
Long EDRE_CTRead(ulng Sn, ulng Ct, pulng Value)
The serial number, counter-number, and a reference parameter must be passed by the calling function. A return code will indicate if any errors occurred.

ACTIVEX CALL
Long EDRECTX.Read(long Port)
The counter number must be passed by the calling function. If the return code is negative it means an error occurred, otherwise it will be the value read from the counter.

Counter	Software assigned number	Resolution
CT0	0	16-bits
CT1	1	16-bits
CT2	2	16-bits
CT3	3	16-bits
CT4	4	16-bits
CT5	5	16-bits

Table 4-3 Counter Resolution

Configuring a counter

A single call is necessary to configure a counter.

API-CALL
Long EDRE_CTConfig(ulng Sn, ulng Ct, ulng Mode, ulng Type, ulng ClkSrc, ulng GateSrc)
The serial number, counter-number, mode, type, clock source and gate source is needed to specify a counter's configuration. A return code will indicate if any errors occurred.

ACTIVEX CALL
Long EDRECTX.Configure(long ct, long mode, long type, ulng source, ulng gate)
The counter-number, mode, type, clock source and gate source is needed to specify a counter's configuration. A return code will indicate if any errors occurred.

Only the counter mode, clock source and type parameters are used by the NETμDAQ. The table below shows the options for each parameter.

Parameter	Description
Sn	Serial Number
Ct	Counter Number: 0 : Counter 0 1 : Counter 1 2 : Counter 2 3 : Counter 3 4 : Counter 4 5 : Counter 5
Mode	82c54 Mode See 82c54 datasheet
Type	Interrupt on TC: 0 : Disabled 1 : Enabled This bit will only generate a interrupt at the interrupt sub-system. The interrupt sub-system must also be setup to generate a interrupt.
Source	0 : Internal (10MHz)

Gate	1 : External (External connector) 0 : Internal 1 : External (External connector)

Table 4-4 Counter Configuration

Controlling the counter gate

A single call is necessary to control a counter's gate.

API-CALL
Long EDRE_CTSoftGate(ulng Sn, ulng Ct, ulng Gate)
The serial number, counter-number and gate are needed to control a counter's gate. A return code will indicate if any errors occurred.

ACTIVEX CALL
Long EDRECTX.SoftGate(ulng Ct, ulng Gate)
The counter-number and mode is needed to control a counter's gate. A return code will indicate if any errors occurred.

These values are acceptable as a gate source.

Value	Description
0	Gate disabled
1	Gate enabled

Table 4-5 Gate Configuration

Internal Oscillator Frequency (10MHz)

The internal oscillator frequency is 10 MHz.

Interrupt

Not included in current versions

Analog Output

The NET 30 has 4 x 16-bit analog output channels with a range of ±10 volt. These channels are very easy to program. A single command is used to write to them.

Writing to a DAC channel

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.

Software assigned channel	Assigned connector Pin
0	DAC0
1	DAC1
2	DAC2
3	DAC3

Table 4-6 Assigned DAC Channels

API-CALL
Long EDRE_DAWrite (ulng Sn, ulng Channel, long uVoltage)
The serial number, DAC channel and micro-voltage is needed to set a DAC channel's voltage. A return code will indicate if any errors occurred.

ACTIVEX CALL
Long EDREDAX.Write (ulng Channel, long uVoltage)
The DAC channel and micro-voltage is needed to set a DAC channel's voltage. A return code will indicate if any errors occurred.

Analog Input

The NET 26/30C has a 16-bit analog to digital converter that supports data conversion up to 250kHz. The device only supports ±10V inputs. Channels can be configured for single ended or differential ended mode

The analog inputs can operate in two modes, single read or scanning. Only one mode can be used at a single moment. The table below shows the relation between the software assigned channels and the connector.

Software assigned channel	Input type	Input pin	Reference Pin
0	Single	ACH0	AGND
…	…	…	…
31	Single	ACH31	AGND
0	Differential	ACH0	ACH8
…	…	…	…
7	Differential	ACH7	ACH15
16	Differential	ACH16	ACH24
…	…	…	…
23	Differential	ACH23	ACH31

Table 4-7 Assigned Analog Input Channels

Reading a single voltage from a channel

To read a single ADC channel you need to specify the channel, voltage range and gain.

API-CALL

Long EDRE_ADSingle (ulng Sn, ulng Channel, ulng Gain, ulng Range, plong uVoltage)

Parameter	Type	Description
Sn	Unsigned long	Device serial number
Channel	Unsigned long	ADC channel to read
Gain	Unsigned long	Gain Codes Device only support one input gain, ±10V Value: 0 — Gain: X 1
Range	Unsigned long	Range Codes Value: 0 — Range: BIPOLAR, SINGLE ENDED Value: 1 — Range: BIPOLAR, DIFFERENTIAL ENDED
uVoltage	Pointer to long	Returned micro voltage
Return	long	Error code

ACTIVEX CALL

Long EDREADX.SingleRead (long Channel)

Make sure to set the *Gain* and *Range* properties of the ADC ActiveX control. This will in turn set the range and gain when reading the ADC channel.

Configuring the ADC subsystem for scanning

This is the most complicated part of configuring the NET 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 NET 26/30.

API-CALL

Long EDRE_ADConfig (ulng Sn, pulng Freq, ulng ClkSrc, ulng Burst, ulng Range, pulng ChanList, pulng GainList, ulng ListSize)

The following parameters must be specified when configuring the ADC sub-system.

<table>
<tr><th>Parameter</th><th>Type</th><th>Description</th></tr>
<tr><td>Sn</td><td>Unsigned long</td><td>Device serial number</td></tr>
<tr><td>Frequency</td><td>Pointer to an unsigned long</td><td>ADC Sampling frequency</td></tr>
<tr><td>ClkSrc</td><td>Unsigned long</td><td>This parameter is used to configure the clock/convert source of the ADC sub-system.
<table>
<tr><th>Offset (bits)</th><th>Description</th></tr>
<tr><td>0</td><td>Clock Source (C0-C7)
0: Internal 10 MHz clock
1: External Convert (EXT_CLK)</td></tr>
<tr><td>8</td><td>Gate Source (G0-G7)
0: Disable (software controlled)
1: External Gate (EXT_GATE)</td></tr>
</table>
Example Layout:
<table>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td>G7</td><td>G6</td><td>G5</td><td>G4</td><td>G3</td><td>G2</td><td>G1</td><td>G0</td><td>C7</td><td>C6</td><td>C5</td><td>C4</td><td>C3</td><td>C2</td><td>C1</td><td>C0</td></tr>
</table></td></tr>
<tr><td>Burst</td><td>Unsigned long</td><td>Not Used</td></tr>
<tr><td>Range</td><td>Unsigned long</td><td>Not used</td></tr>
<tr><td>ChanList</td><td>Pointer to an unsigned long</td><td>This is a pointer to an array that contains the list of channels to be scanned. The array length should be the same length as the value of ListSize</td></tr>
<tr><td>GainList</td><td>Pointer to an unsigned long</td><td>GainList is an array that contains the gain/range settings for each channel in the scan list. The array length should the same as the ListSize value.
<table>
<tr><th>Offset (bits)</th><th>Description</th></tr>
<tr><td>0</td><td>Specifies the gain of the channel.(G) Only value of 0 supported.
<table>
<tr><th>Value</th><th>Gain</th></tr>
<tr><td>0</td><td>X 1</td></tr>
</table></td></tr>
<tr><td>8</td><td>Specifies the range of the channel. (R)
<table>
<tr><th>Value</th><th>Range</th></tr>
<tr><td>0</td><td>BIPOLAR, SINGLE ENDED</td></tr>
<tr><td>1</td><td>BIPOLAR, DIFFERENTIAL</td></tr>
</table></td></tr>
</table>
Example Layout:
<table>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td>R7</td><td>R6</td><td>R5</td><td>R4</td><td>R3</td><td>R2</td><td>R1</td><td>R0</td><td>G7</td><td>G6</td><td>G5</td><td>G4</td><td>G3</td><td>G2</td><td>G1</td><td>G0</td></tr>
</table></td></tr>
<tr><td>ListSize</td><td>unsigned long</td><td>This is the length of the channel list.</td></tr>
<tr><td>Return</td><td>long</td><td>Error code</td></tr>
</table>

ACTIVEX CALL

Long EDREADX.Configure (plong Channels, plong Gains, long ListSize)

The *Frequency* and *ClockSource* ADC ActiveX control must be setup before calling the configure function. See the above table for the layout of the Channels and Gains lists.

EDREADX.Frequency

This is the sampling frequency of the ADC process. This parameter must be set before calling the Configure method. After calling the Configure method the Frequency property will be set to the actual sampling frequency.

Please Note!!

- **On the NET 26/30 the frequency is the update rate of the A/D converter. This means that the board will convert the channels at a period of equal to the frequency and the channels in the sequence of the channel list. The end result is that the time between samples is equal to 1/Frequency.**

EDREADX.ClockSource
The clock source property is used to specify the clock settings for the ADC process.

Offset (bits)	Description
0	Clock Source (C0-C7) 0: Internal 10 MHz clock 1: External Convert (EXT_CLK)
8	Gate Source (G0-G7) 0: Disable (software controlled) 1: External Gate (EXT_TRIG)

Example Layout

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
G7	G6	G5	G4	G3	G2	G1	G0	C7	C6	C5	C4	C3	C2	C1	C0

Starting and Stopping the ADC process

A single call is necessary to start or stop the ADC process

API-CALL
Long EDRE_ADStart (ulng Sn)
A serial number needs to be specified to start the ADC process. A returned error code will indicate if the function succeeded.

ACTIVEX CALL
Long EDREADX.Start ()
A call to the start method will start the ADC process of the device too which the ActiveX control is linked. A returned error code will indicate if the function succeeded.

API-CALL
Long EDRE_ADStop (ulng Sn)
A serial number needs to be specified to stop an ADC process. A returned error code will indicate if the function succeeded.

ACTIVEX CALL
Long EDREADX.Stop ()
A call to the start method will stop the ADC process of the device too which the ActiveX control is linked. A returned error code will indicate if the function succeeded.

Device buffer process and functions

The device doesn't support continuous uninterrupted sampling. The device has a 500k-byte onboard memory that is used to buffer the sampled data. The size of the buffer can be configured, using the *EDRE_Query* function and the *ADIRQLEVEL* query code. The query code is defined in the *"query.h"* file found on the CD-ROM supplied. The size that the buffer value is set to is called the "*water level"*.
The value that the water level is set to is very important to give your application a good steady supply of data. The device will sample data until the water level are reached before it will indicate to the EDRE API that data is available. The device will then automatically stop the sampling process while the EDRE API downloads the data from

the device. Once all the data has been downloaded the device will configure the ADC hardware and start the sampling process. This will repeat until the *EDRE_ADStop* function is called.

A single call is necessary copy data from the EDRE API buffer to a user buffer. The buffer is a large circular buffer that can hold 1mega samples of data. This buffer needs to be emptied regularly to make sure it does not overrun. The buffer can be queried with number of samples available and other status issues as well. There are two functions available to copy data, one for copying voltages, another to copy the raw data. The raw data is significantly faster as for the data does not have to be converted to voltages before copying it to the user buffer. The raw data also occupies less space than the micro voltage buffer. There are also functions to write data to disk as the user buffer get copied. Refer to the EDR Enhanced programming manual for a reference to these functions.

API-CALL
Long EDRE_ADGetData (ulng Sn, plong Buf, pulng BufSize)
ACTIVEX CALL
Long EDREADX.GetData (plong Buffer, plong Size)
To retrieve data from the driver buffer the serial number need to be supplied, a buffer to hold the data and the size of the buffer or requested number of samples. The driver will only copy the number of available samples in multiple of the channel list. For the ActiveX call only the buffer and size need to be supplied.

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 appendix has a list of all possible query codes.

API-CALL
Long EDRE_Query (ulng Sn, ulng QueryCode, ulng Param)
A serial number, query code and parameter must be specified when doing a query. To query the number of samples available in the EDRE API buffer use the *ADUNREAD* code define in the *"query.h"* file found on the CD-ROM supplied. You will notice that the unread samples will increment in multiples of the water level.

ACTIVEX CALL
Long EDREADX.GetUnread ()
This function automatically queries the ADC driver buffer for the number of available samples.

Temperature Input

The NET 73T8/16/32 has a basic 8 differential channel configuration (See table 3-5), accessible via a DB25M connector. The channels on one of these DB25M connectors will be referred to as a channels set. Each NET 73 is supplied with is set of *μDAQ temp T/C adapters*.

Temperature measurements can be done with as little as four lines of code.

Reading CJC Channel

Each channels set has a CJC channel. The CJC channels is use in software when calculating the temperature for Cold Junction Compensation, hence CJC. Each of the *μDAQ temp T/C adapters* has a circuit that will supply a voltage of 10mVolt per 1 degree C. Reading the CJC channel is as easy as reading an analog channel. Each of the CJC channel is allocated a channel number witch are always the last channels of any device.
E.g. if your device has two CJC channels and 34 analog channels, channel 0-31 will be analog inputs and channels 32 and 33 will be the CJC channels.

The table shows the normal CJC assignment for the NET 73.

Device	CJC Channels	
NET 73T/R8	CJC	Value Assigned
	0	8
NET 73T/R16	CJC	Value Assigned
	0	16
	1	17
NET 73T/R32	CJC	Value Assigned
	0	32
	1	33
	2	34
	3	35
NET 30C16-73T/R16	CJC	Value Assigned
	0	32
	1	33
NET 30C32-73T/R16	CJC	Value Assigned
	0	48
	1	49

Table 4-8 CJC Channels Assigned

The value that is assigned to a CJC channel can be queried with software as well.

API-CALL
Long EDRE_Query (ulng Sn, ulng Code, long Param)
Serial number, Query code *ADAMBCHAN or 141* and the param that represent the CJC channel 0-3. A return code will indicate the channel assigned or if any errors occurred.

ACTIVEX CALL
Long EDREUtlX.Query (ulng Channel, long uVoltage)
Query code *ADAMBCHAN or 141* and the param that represent the CJC channel 0-3. A return code will indicate the channel assigned or if any errors occurred.

Reading the CJC channel

API-CALL
Long EDRE_ADSingle (ulng Sn, ulng Channel, ulng Gain, ulng Range, plong uVoltage)

Parameter	Type	Description
Sn	Unsigned long	Device serial number
Channel	Unsigned long	Assigned Channel
Gain	Unsigned long	NULL
Range	Unsigned long	NULL
uVoltage	Pointer to long	Returned micro voltage
Return	long	Error code

ACTIVEX CALL
Long EDREADX.SingleRead (long Channel)
Only the assigned channel value is needed. A return code will indicate the voltage in microvolt.

Read Thermo Couple Channel

Reading the thermocouple channel is the same as reading the CJC channels. The thermo couple channels for the NET 73 will always start at channel 0. For a device with both normal ADC channels and Temperature channels the channels value assigned to the temperature channels will start after the normal ADC channels.

Device	Channels and assigned software value	
NET 73T/R8	Cannel	Value Assigned
	0	0
	…	…
	7	7
NET 73T/R16	Cannel	Value Assigned
	0	0
	…	…
	15	7
NET 73T/R32	Cannel	Value Assigned
	0	0
	…	…
	31	31
NET 30C16-73T/R16	Cannel	Value Assigned
	0	16
	…	…
	15	31
NET 30C32-73T/R16	Cannel	Value Assigned
	0	32
	…	…
	15	47

Table 4-9 Temperature Channels Assigned

API-CALL

Long EDRE_ADSingle (ulng Sn, ulng Channel, ulng Gain, ulng Range, plong uVoltage)

Parameter	Type	Description
Sn	Unsigned long	Device serial number
Channel	Unsigned long	Assigned Channel
Gain	Unsigned long	NULL
Range	Unsigned long	NULL
uVoltage	Pointer to long	Returned micro voltage
Return	long	Error code

ACTIVEX CALL

Long EDREADX.SingleRead (long Channel)

Only the assigned channel value is needed. A return code will indicate the voltage in microvolt.

Calculating Ambient Temperature

API-CALL
Long EDRE_CalcCJCmC(long cjcuv)

ACTIVEX CALL
Long EDREADX.CalcCJCmC(long cjcuv)

Parameter	Type	Description
cjcuv	Long	CJCTemp channel uVolts
Return	Long	Milli Degrees Celsius

Calculating Temperature for Thermocouples

API-CALL
Long EDRE_CalcTCmC(long tctype, long tcuv, long ambientmc)

ACTIVEX CALL
Long EDREADX.CalcTCmC(long tctype, long tcuv, long ambientmc)

Parameter	Type	Description
tctype	Long	Type Thermocouple used. (See appendix for details)
tcuv	Long	Voltage read from channel uVolts
ambientmc	Long	Ambient temperature mille Degrees
Return	Long	Milli Degrees Celsius

5

Calibration - NET 26/30C

If the Ethernet 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 voltmeter and calibration voltage source. This will help to configure your device more accurately.

Calibration Procedure – NET 26/30C

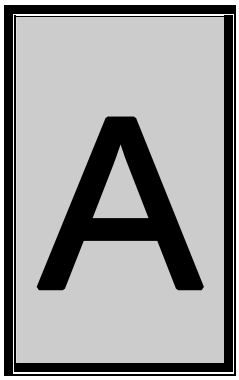
1. **Install the USB Calibration Software <CD-Rom>\EDRE\APPS\uDAQ_cal_36C_30C\uDAQ30C_Calibration.exe**
2. **Run the Calibration Software.**
3. **Follow the step-by-step information on screen to calibrate your device.**
4. **Make sure to save the data to your device.**

Calibration - NET 73

If the Ethernet 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.

Calibration Procedure – NET 73

1. **Install the NET Calibration Software <CD-Rom>\EDRE\APPS\ udaq_cal_73 \uDAQ73CalibrationSoftware.exe**
2. **Run the Calibration Software.**
3. **Follow the step-by-step information on screen to calibrate your device.**
4. **Make sure to save the data to your device.**



A.Specifications

Digital Input/Output Characteristics

Number of Digital Channels:

Device	Channels
NET 24A	24
NET 24C	24
NET 48A	48
NET 48C	48
NET 72A	72
NET 72C	72
NET 96A	96
NET 96C	96
NET 120A	120
NET 26C16/32	24
NET 30C16/32	24
NET 73	24

Number of Grouped Channels:

Device	PPI Channels
NET 24A	3
NET 24C	3
NET 48A	6
NET 48C	6
NET 72A	9
NET 72C	9
NET 96A	12
NET 96C	12
NET 120A	15
NET 26C16/32	3
NET 30C16/32	3
NET 73	3

Compatibility: 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

Counter-Timer Characteristics

Number of Counter-Timer Channels:

Device	Channels
NET 24A	0
NET 24C	6
NET 48A	0
NET 48C	6
NET 72A	0
NET 72C	6
NET 96A	0
NET 96C	6
NET 120A	0
NET 26C16/32	0
NET 30C16/32	0
NET 73	0

Resolution: 16-bits

Compatibility 82C54 / TTL

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

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

Analog Output Characteristics (NET 30C16/32)

Number of Channels:	**4**
Resolution:	**16-bits**
Maximum Update Rate	**2 milliseconds**
Data Transfer	**Ethernet TCP/IP 10/100Mbps**
Full Scale Error	**±2 LSB**
Zero Offset Error	**±2 LSB**
Output Drive	**±5 milliamp**
Load Characteristics	**2 KΩ || 10 nF**
Power On State	**0 Volt**

Analog Input Characteristics (NET 26/30C16/32)

Input Characteristics

Number of Channels

Device Type	Single Ended	Differential Ended
NET 26/30C16	16	8
NET 26/30C32	32	16
NET 30C16-73T/R16	16	8
NET 30C32-73T/R16	32	16

Resolution 16-bits

Maximum Update Rate 250 KS/s

Data Transfer Ethernet TCP/IP 10/100Mbps

Input Programmable Ranges

Channel Gain	Bipolar Range
1	±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 32

Conversion Characteristics

Maximum Conversion Rate 250 000 Samples per second

Converter Type Successive approximation

Resolution 16-bits

Relative Accuracy ±2 LSB

External Clock – EXT_CLK pin

Maximum Rate 250 000 Hz

Synchronization Internal 20 MHz clock

Conversion Falling Edge

External Gate – EXT_GATE pin

Enable Process High Input (>2.4V DC)

Disabled Process Low Input (<1.2V DC)

Thermo Couple Input Characteristics

Number of Channels

Device	Differential Channels
NET 73T/R8	8
NET 73T/R16	16
NET 73T/R32	32
NET 30C16/32-73T/R16	16

Resolution 14-bits

Maximum Update Rate 2 milliseconds

Data Transfer Ethernet TCP/IP 10/100Mbps

Input Programmable Ranges

Channel Gain	Bipolar Range
30	±83mV

Input Coupling DC

Relative Accuracy ±1 LSB

Gain x 30 Offset Error ±0.011 millivolts

Bus Interface

Bus Type	**Ethernet**
Bus Speed	**10/100Mbps**
Protocol	**TCP/IP**

Power Requirements

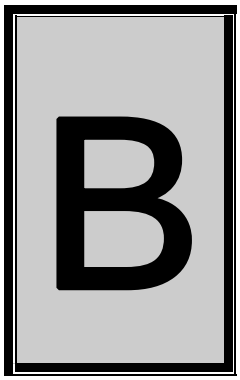
Device	Minimum	Typical	Maximum	Power Source
NET 24A		50 mA	100 mA	External PSU
NET 24C		295 mA	345 mA	External PSU
NET 48A		55 mA	155 mA	External PSU
NET 48C		300 mA	400 mA	External PSU
NET 72A		60 mA	210 mA	External PSU
NET 72C		310 mA	460 mA	External PSU
NET 96A		65 mA	265 mA	External PSU
NET 96C		315 mA	515 mA	External PSU
NET 120A		70 mA	320 mA	External PSU
NET 26C16/32		450 mA	500 mA	External PSU
NET 30C16/32		480 mA	530 mA	External PSU
NET 73T/R16		450 mA	500 mA	External PSU

Environmental / Physical

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

Connectors

NET 24A	**DB25 (M)**
NET 24C	**2 x DB25 (M)**
NET 48A	**2 x DB25 (M)**
NET 48C	**3 x DB25 (M)**
NET 72A	**3 x DB25 (M)**
NET 72C	**4 x DB25 (M)**
NET 96A	**4 x DB25 (M)**
NET 96C	**5 x DB25 (M)**
NET 120A	**5 x DB25 (M)**
NET 26/30C16	**2 x DB25 (M)**
NET 26/30C32	**3 x DB25 (M)**
NET 73T/R8	**2 x DB25 (M)**
NET 73T/R16	**3 x DB25 (M)**
NET 73T/R32	**5 x DB25 (M)**



B.Configuration Constants

Query Codes

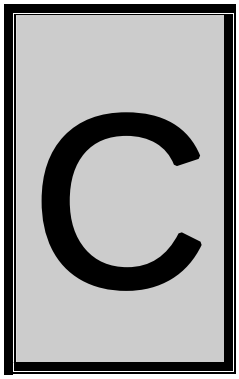
Name	Value	Description
APIMAJOR	1	Query EDRE API major version number.
APIMINOR	2	Query EDRE API minor version number.
APIBUILD	3	Query EDRE API build version number.
APIOS	4	Query EDRE API OS type.
APINUMDEV	5	Query number of devices installed.
BRDTYPE	10	Query a board's type.
BRDREV	11	Query a board's 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

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

Digital I/O Return Query Codes 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.



C.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. Currently only the following version are available.

Board	Description
NET 24A	24 channel digital I/O Ethernet device
NET 24C	24 channel digital I/O & counter-timer ETHERNET device
NET 48A	48 channel digital I/O ETHERNET device
NET 48C	24 channel digital I/O & counter-timer ETHERNET device
NET 72A	72 channel digital I/O ETHERNET device
NET 72C	24 channel digital I/O & counter-timer ETHERNET device
NET 96A	96 channel digital I/O ETHERNET device
NET 96C	24 channel digital I/O & counter-timer ETHERNET device
NET 120A	120 channel digital I/O ETHERNET device
NET 26C16/32	16/32 channel analog input ETHERNET device
NET 30C16/32	16/32 channel & 4 channel analog input/output ETHERNET device
NET 73T8	8 Channel Thermo Couple input ETHERNET device
NET 73T16	16 Channel Thermo Couple input ETHERNET device
NET 73T32	32 Channel Thermo Couple input ETHERNET device
NET 73R8	8 Channel RTD input ETHERNET device
NET 73R16	16 Channel RTD input ETHERNET device
NET 73R32	32 Channel RTD input ETHERNET device
NET 30C16-73T16	16 channel & 4 channel analog input/output + 16 Channel Thermo Couple input ETHERNET device
NET 30C16-73R16	16 channel & 4 channel analog input/output + 16 RTD input ETHERNET device
NET 30C32-73T16	32 channel & 4 channel analog input/output + 16 Channel Thermo Couple input ETHERNET device
NET 30C32-73R16	32 channel & 4 channel analog input/output + 16 RTD input ETHERNET device

Table C-1 NET μDAQ Ordering Information