Errata Sheet

This Errata Sheet contains corrections or changes made after the publication of this manual.

Product Family:	Terminator I/O	**Date**:	June 2023
Manual Number	T1H-EBC-M		
Revision and Date	2nd Edition, Rev. B; May, 2014		

Changes to Chapter 3. Configuring the T1H-EBC(100) Using NetEdit3

Page 3-8. Using NetEdit3

Add the following NOTE to the top of page 3-8:

NOTE: T1H-EBC100 modules have a DIP switch located under the cover on the left side. Turning this switch on will reset the IP Address, subnet and gateway to the factory defaults. (T1H-EBC modules do not have a DIP switch.)

Changes to Chapter 4. MODBUS® TCP/IP for T1H-EBC100

Page 4-8. T1H-EBC100 System Memory

I*n the fourth column of the table, replace the existing text for "4 - Flags:" with the following:*

4 - Flags:

Bit 0: REBOOT BIT (R/W) - This bit is a status bit indicating if any module has rebooted. In other words it is OFF after a proper power up. It is ON only if a module has disappeared from the base and has reappeared. This normally might occur if a module in the base has been hot swapped. To clear this bit write ANY value to this word.

Bit 1: RE-SCAN BIT (WO) - Turning this bit ON will cause the EBC100 to re-scan its base. This should be done after a hot swap of a module.

Bit 2: CLEAR OUTPUTS BIT (WO) - Setting this bit ON prior to, or at the same time as Bit 1 will cause all discrete outputs to go OFF and all analog putputs to go to zero during the rescan. Leaving this bit OFF will cause al discrete and analog outputs to remain at their current value after the rescan.

Bit 3-15: Reserved

Added Note for Appendix A . Using T1H-EBC(100) with Think & Do

Note: Think & Do software (PC-TD8-USB & PC-TD8-WEB4-USB) has been retired and no longer supported. *Please consider Do-more software as a consideration for your project.*

AUTOMATIONDIRECT.com

Terminator I/O
Ethernet Base Controller
Manual

Manual Number T1H-EBC-M

WARNING

Thank you for purchasing automation equipment from ***Automationdirect.com***™, doing business as **AutomationDirect**. We want your new automation equipment to operate safely. Anyone who installs or uses this equipment should read this publication (and any other relevant publications) before installing or operating the equipment.

To minimize the risk of potential safety problems, you should follow all applicable local and national codes that regulate the installation and operation of your equipment. These codes vary from area to area and usually change with time. It is your responsibility to determine which codes should be followed, and to verify that the equipment, installation, and operation are in compliance with the latest revision of these codes.

At a minimum, you should follow all applicable sections of the National Fire Code, National Electrical Code, and the codes of the National Electrical Manufacturer's Association (NEMA). There may be local regulatory or government offices that can also help determine which codes and standards are necessary for safe installation and operation.

Equipment damage or serious injury to personnel can result from the failure to follow all applicable codes and standards. We do not guarantee the products described in this publication are suitable for your particular application, nor do we assume any responsibility for your product design, installation, or operation.

Our products are not fault-tolerant and are not designed, manufactured or intended for use or resale as on-line control equipment in hazardous environments requiring fail-safe performance, such as in the operation of nuclear facilities, aircraft navigation or communication systems, air traffic control, direct life support machines, or weapons systems, in which the failure of the product could lead directly to death, personal injury, or severe physical or environmental damage ("High Risk Activities"). **AutomationDirect** *specifically disclaims any expressed or implied warranty of fitness for High Risk Activities.*

For additional warranty and safety information, see the Terms and Conditions section of our Desk Reference. If you have any questions concerning the installation or operation of this equipment, or if you need additional information, please call us at 770-844-4200.

This publication is based on information that was available at the time it was printed. At **AutomationDirect** we constantly strive to improve our products and services, so we reserve the right to make changes to the products and/or publications at any time without notice and without any obligation. This publication may also discuss features that may not be available in certain revisions of the product.

Trademarks

This publication may contain references to products produced and/or offered by other companies. The product and company names may be trademarked and are the sole property of their respective owners. **AutomationDirect** disclaims any proprietary interest in the marks and names of others.

AVERTISSEMENT

Nous vous remercions d'avoir acheté l'équipement d'automatisation de **Automationdirect.comMC**, en faisant des affaires comme **AutomationDirect**. Nous tenons à ce que votre nouvel équipement d'automatisation fonctionne en toute sécurité. Toute personne qui installe ou utilise cet équipement doit lire la présente publication (et toutes les autres publications pertinentes) avant de l'installer ou de l'utiliser.

Afin de réduire au minimum le risque d'éventuels problèmes de sécurité, vous devez respecter tous les codes locaux et nationaux applicables régissant l'installation et le fonctionnement de votre équipement. Ces codes diffèrent d'une région à l'autre et, habituellement, évoluent au fil du temps. Il vous incombe de déterminer les codes à respecter et de vous assurer que l'équipement, l'installation et le fonctionnement sont conformes aux exigences de la version la plus récente de ces codes.

Vous devez, à tout le moins, respecter toutes les sections applicables du Code national de prévention des incendies, du Code national de l'électricité et des codes de la National Electrical Manufacturer's Association (NEMA). Des organismes de réglementation ou des services gouvernementaux locaux peuvent également vous aider à déterminer les codes ainsi que les normes à respecter pour assurer une installation et un fonctionnement sûrs.

L'omission de respecter la totalité des codes et des normes applicables peut entraîner des dommages à l'équipement ou causer de graves blessures au personnel. Nous ne garantissons pas que les produits décrits dans cette publication conviennent à votre application particulière et nous n'assumons aucune responsabilité à l'égard de la conception, de l'installation ou du fonctionnement de votre produit.

Nos produits ne sont pas insensibles aux défaillances et ne sont ni conçus ni fabriqués pour l'utilisation ou la revente en tant qu'équipement de commande en ligne dans des environnements dangereux nécessitant une sécurité absolue, par exemple, l'exploitation d'installations nucléaires, les systèmes de navigation aérienne ou de communication, le contrôle de la circulation aérienne, les équipements de survie ou les systèmes d'armes, pour lesquels la défaillance du produit peut provoquer la mort, des blessures corporelles ou de graves dommages matériels ou environnementaux ("activités à risque élevé"). La société **AutomationDirect** nie toute garantie expresse ou implicite d'aptitude à l'emploi en ce qui a trait aux activités à risque élevé.

Pour des renseignements additionnels touchant la garantie et la sécurité, veuillez consulter la section Modalités et conditions de notre documentation. Si vous avez des questions au sujet de l'installation ou du fonctionnement de cet équipement, ou encore si vous avez besoin de renseignements supplémentaires, n'hésitez pas à nous téléphoner au 770-844-4200.

Cette publication s'appuie sur l'information qui était disponible au moment de l'impression. À la société **AutomationDirect**, nous nous efforçons constamment d'améliorer nos produits et services. C'est pourquoi nous nous réservons le droit d'apporter des modifications aux produits ou aux publications en tout temps, sans préavis ni quelque obligation que ce soit. La présente publication peut aussi porter sur des caractéristiques susceptibles de ne pas être offertes dans certaines versions révisées du produit.

Marques de commerc

La présente publication peut contenir des références à des produits fabriqués ou offerts par d'autres entreprises. Les désignations des produits et des entreprises peuvent être des marques de commerce et appartiennent exclusivement à leurs propriétaires respectifs. **AutomationDirect** nie tout intérêt dans les autres marques et désignations.

Manual Revisions

If you contact us in reference to this manual, be sure and include the revision number.

Title: Terminator Installation and I/O Manual
Manual Number: T1H-EBC-M

Edition/Rev	Date	Description of Changes
Original	11/01	original issue
2nd Edition	08/04	added T1H-EBC100, NetEdit3, HTML Configuration
2nd Edition, Rev A	01/05	added Holding Register Mirror Image addressing for Modbus Function Code 3 Clients
2nd Edition, Rev B	05/14	added Appendix D and Appendix E (T1H-EBC100 addressing)

Table of Contents

Chapter 1: Introduction

Chapter 2: Installation and Setup

Chapter 3: Configuring the T1H-EBC(100) Using NetEdit3

Chapter 4: MODBUS® TCP/IP for T1H-EBC100

Chapter 5: T1H-EBC100 DHCP & HTML Configuration

Appendix A: Using the T1H-EBC(100) with Think & DO

Appendix B: Using the T1H-EBC(100) with KEPDirect OPC Server

Appendix C: Using the KEPDirect OPC Quick Client

Appendix D: T1H-EBC100 Analog Module Addressing - Modbus TCP

Appendix E: T1H-EBC(100) Analog Module Addressing - H2/4-ERM(100)

1

Introduction

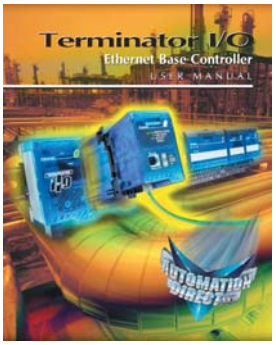
In This Chapter. . . .

- Manual Overview
- Ethernet Base Controller Overview
- Ethernet Standards

Manual Overview

Overview of this Manual

This manual describes the installation and operation of the **Ethernet Base Controller (EBC)**. You will find the necessary information for configuring the T1H-EBC or T1H-EBC100, installing the module, and connecting the EBC to a 10Base-T or 100BaseT Ethernet network. In this manual, the EBC designation is used when the subject applies to both the T1H-EBC and T1H-EBC100. Otherwise, the specific part number will be listed.



Other Reference Materials

You may find other technical manuals useful for your application. For technical information related to your PC-based control software, your PC or other network masters, please refer to the appropriate manual for that product.

- Terminator I/O Installation and I/O Manual (T1K-INST-M)

Who Should Read This Manual

You will find this manual helpful for setup and installation if you have chosen to use the following:

- Network master – PC-based Control with embedded Ethernet I/O drivers, KEPDirect EBC I/O Server or ***Direct***Logic PLCs/WinPLC using the Ethernet Remote Master (ERM) module
- Automationdirect Terminator I/O products

A familiarity with Ethernet communications and with the setup and installation of industrial controls is helpful. An understanding of electrical codes is essential.

Technical Support

We strive to make our manuals the best in the industry. We rely on your feedback to let us know if we are reaching our goal. If you cannot find the solution to your particular application, or, if for any reason you need additional technical assistance, please call us at

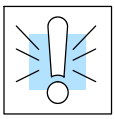
770-844-4200.

Our technical support group is glad to work with you in answering your questions. They are available **weekdays from 9:00 a.m. to 6:00 p.m. Eastern Time**. We encourage you to visit our site on the worldwide web where you can find technical and nontechnical information about our products and our company.

http://www.automationdirect.com.

If you have a comment or question about any of our products, services, or manuals, please fill out and return the 'Suggestions' card that was shipped with this manual.

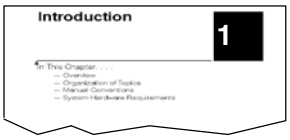
When you see the “notepad” icon in the left-hand margin, the paragraph to its immediate right will be a **special note**.



When you see the “exclamation mark” icon in the left-hand margin, the paragraph to its immediate right will be a **warning**. This information could prevent injury, loss of property, or even death.

Key Topics for Each Chapter

The beginning of each chapter will list the key topics that can be found in that chapter.



Ethernet Base Controller Overview

The Ethernet Base Controllers provide a low-cost, high-performance Ethernet link between a network master controller and an Automationdirect Terminator I/O slave system. Network masters include the DL205, DL405 ***Direct***Logic PLCs and WinPLCs using the Ethernet Remote Master module (ERM), and PCs using PC-based control software that includes embedded Ethernet I/O drivers or through a compatible OPC server. The T1H-EBC100 also supports the MODBUS TCP/IP protocol.

The Ethernet Base Controller serves as an interface between the master control system and the Terminator I/O modules. The control function is performed by the master controller, not the EBC slave. The EBC is positioned immediately to the right of the first power supply and communicates across the backplane to input and output modules. The function of the EBC is to:

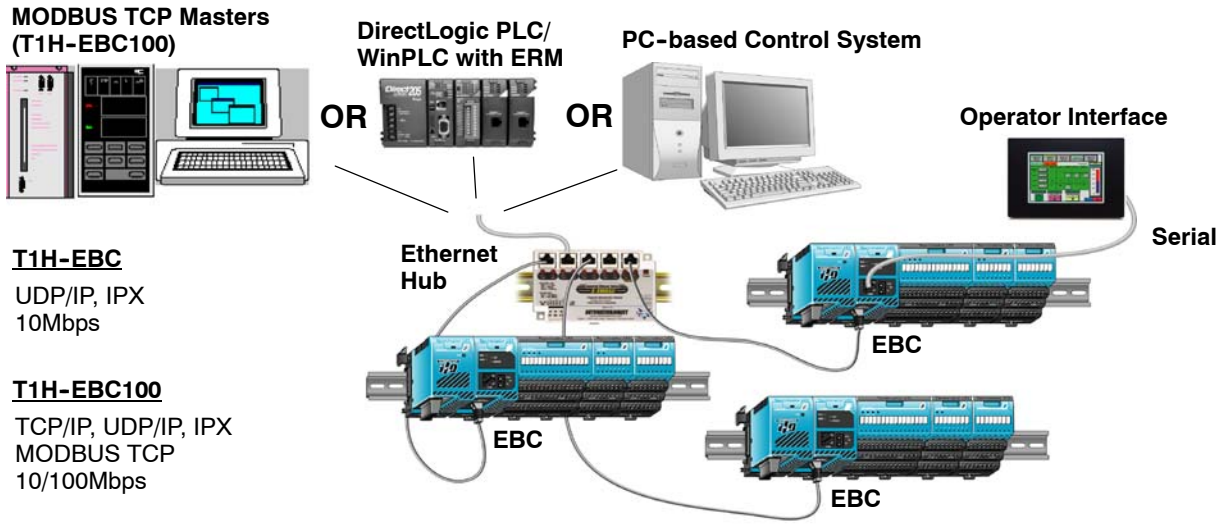
- process analog and digital input signals
- format the I/O signals to conform to the Ethernet standard
- transmit the signals to the network master
- receive and translate output signals from the network master
- distribute the output signals to the appropriate output module in the base

I/O Values Stored in Cache Memory

The EBC module continually scans all I/O and stores the most recent values in cache memory. The cache memory contents are available to the master controller as a block of data or by individual slot location. The EBC reads all channels of digital and analog modules on each scan.

Typically, the network master will request *all* input and output values at the same time from the EBC. The EBC passes the cache memory values for all channels of all input and output modules. By using this method, very fast response times can be achieved by the network master control system. Various master controllers with EBC slaves are shown below.

Example EBC Systems: Various Masters with EBC Slaves



Industry Standard Ethernet

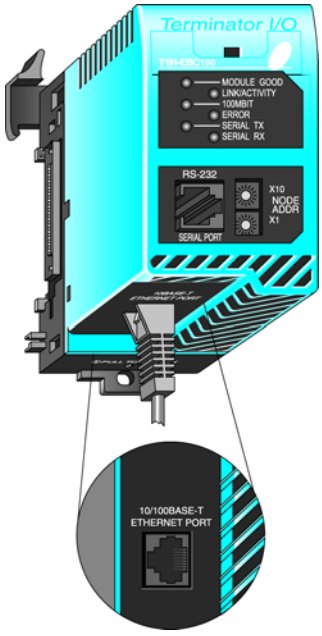
The T1H-EBC module supports industry standard 10Base-T Ethernet communications. It allows up to10Mbps transfer rates between your master controller and and your I/O.

The T1H-EBC100 module supports industry standard 10/100Base-T Ethernet communications. It allows up to100Mbps transfer rates between your master controller and your I/O.

WARNING: For deterministic Ethernet communication you must use a dedicated network of EBC modules connected to your master control system. The EBC modules and the master controller must be the only devices on the network.

T1H-EBC/ T1H-EBC100

The EBC installs to the right of the first power supply (see chapter 2 for basic installation steps). For further information about installing power supplies and I/O modules, consult the Terminator I/O Installation and I/O Manual (T1K-INST-M).



RS-232 Serial Port

An RJ12 RS-232 serial port on-board the EBC module allows serial communication to an operator interface device or other serial device. See your master controller documentation to determine whether this EBC feature is supported.

Important Note: The T1H-EBC100 is configured at the factory to look for a DHCP (Dynamic Host Configuration Protocol) server at power up. Refer to Chapter 5 for information on DHCP.

2

Installation and Setup

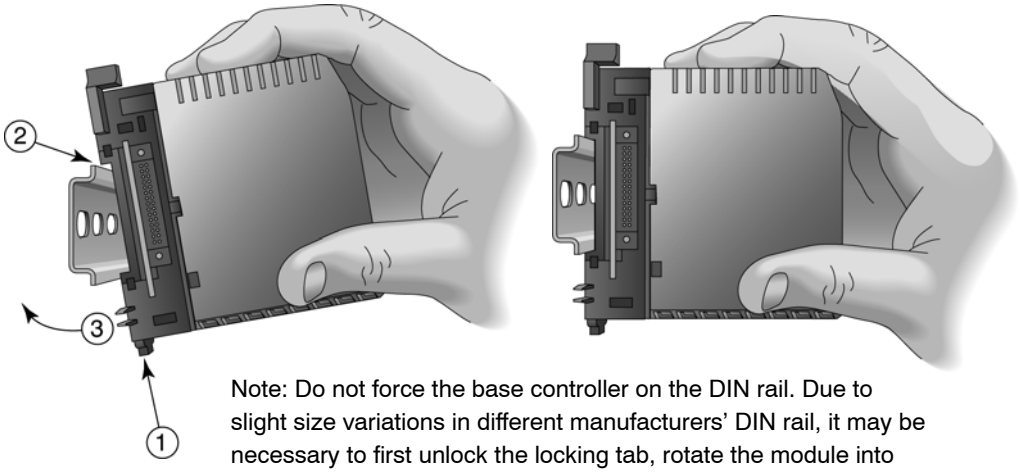
In This Chapter. . . .

- Installing the Ethernet Base Controller
- EBC Network Identifiers
- Setting the Node Address
- 10BaseT / 100BaseT Network Cabling
- Maximum 10BaseT / 100BaseT Cable Length
- Specifications
- LED Indicators and Hot Swapping I/O Modules
- Ethernet Standards

Installing the Ethernet Base Controller

The EBC installs to the *right* of the first power supply. To mount the module on the DIN rail, follow steps 1 through 3 below.

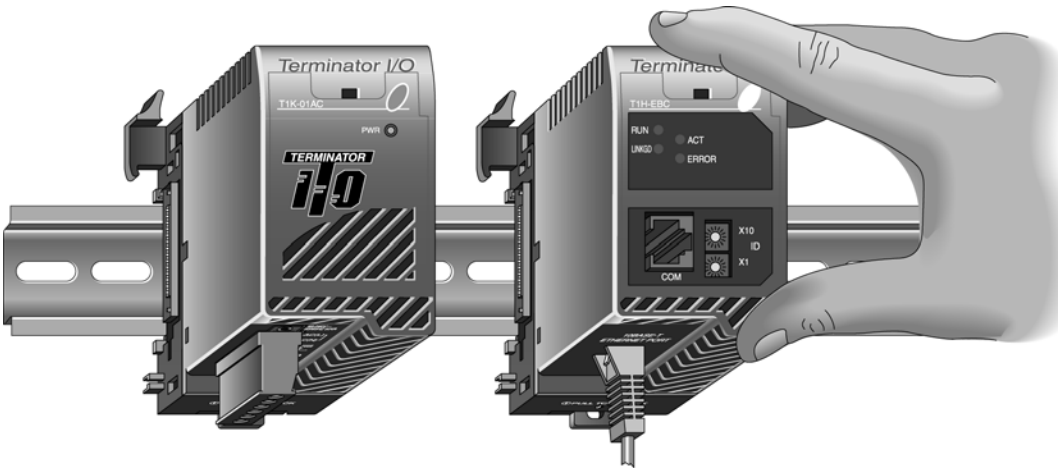
1. Push in the locking tab on the bottom of the module.
2. Hook the upper tab over the upper flange of the DIN rail.
3. Tilt the module toward the DIN rail until it snaps securely into place.



Note: Do not force the base controller on the DIN rail. Due to slight size variations in different manufacturers' DIN rail, it may be necessary to first unlock the locking tab, rotate the module into place, then latch the locking tab.

Assure that power wiring is not connected.

When the module is securely attached to the DIN rail, push the module toward the power supply until the connectors are joined and the release arm of the EBC has clamped the two modules together.



Continue to add I/O modules to the right of the EBC as necessary for your application. More information about power wiring and power budgeting is available in the Terminator I/O Installation Manual, T1K-INST-M.

EBC Network Identifiers

Each Ethernet Base Controller module must be assigned at least one unique identifier to make it possible for PCs or other clients (masters) to recognize it on the network. Two methods of identifying the EBC module give it the flexibility to fit most networking schemes.

The identifiers are:

- Module ID (IPX protocol only)
- IP Address (for TCP/IP and MODBUS TCP protocols); see Chapter 3

Setting the Node Address

Each Ethernet Base Controller residing on a network must have a Node Address, and each Node Address must be unique. **Duplicate Node Addresses** on the same network will cause unpredictable results and **must be avoided**.

Several Methods for Setting Node Address

There are several methods for setting the Node Address:

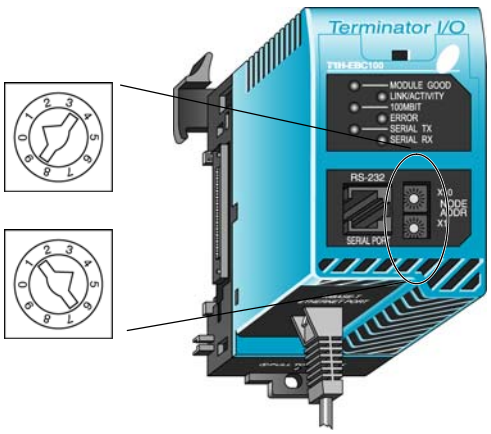
- The **rotary switches** on the face of the EBC module
- The **NetEdit3** software utility (described in Chapter 3)
- HTML Confguration (after IP address is assigned to module using NetEdit3; described in Chapter 5)
- The software utility in your **PC-based Control software** (if a utility is provided)

Setting the Node Address Using the Rotary Switches

Two rotary switches, each labeled 0 – 9, are located on the face of the EBC. Use a small screwdriver to set the switches to a two-digit number.

The upper switch sets the *tens* position and the lower switch sets the *ones* position. Setting the upper switch to 1 and the lower switch to 4 yields a Node Address of 14.

Do not use Node Address “0” *for communications*. Node Address “0” is used only to allow communications with a PC while changing the Node Address (Module ID) in software. If the rotary switches are set to a number greater than 0, the software tools are disabled from setting the Node Address.



NOTE: The rotary switch settings are read only at powerup. You must cycle power if you change the rotary switches. If you change the rotary switches and cycle power, the rotary switch setting will override any previous software setting.

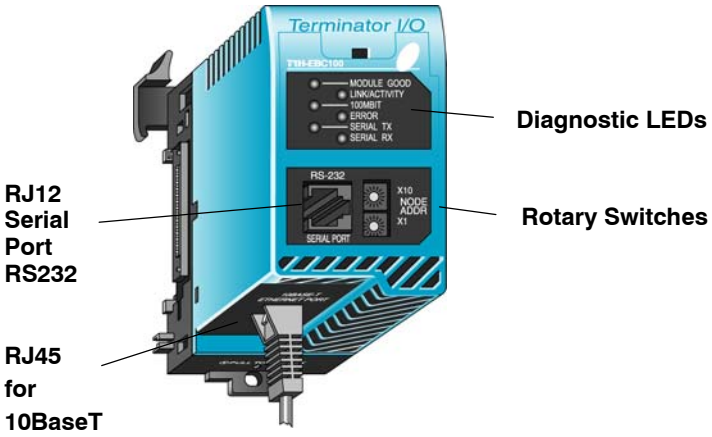
Setting Node Address with Software Tool

Software changes to the Node Address do not require cycling power. To set the Node Address using one of the available software tools, do the following:

- Check to be sure both rotary switches are set to the “0” position
- Connect module to the Ethernet network
- Apply power
- Link to the module and change the Node Address using the software of your choice. Remember to “update” the module before exiting the software.

10BaseT / 100BaseT Network Cabling

The T1H-EBC module supports the Ethernet 10BaseT standard. The T1H-EBC100 module supports the Ethernet 10/100BaseT standard. The standards call for twisted pairs of copper wire conductors.

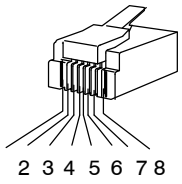


10BaseT/100BaseT connections

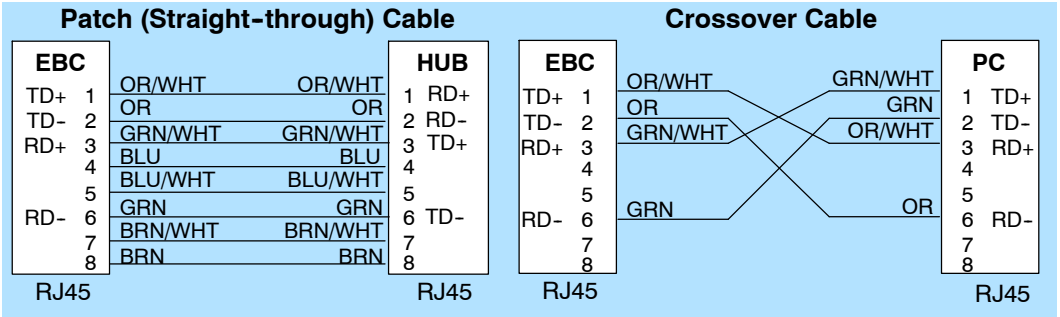
The EBC has an eight-pin modular jack that accepts RJ45 connector plugs. UTP (Unshielded Twisted-Pair) cable is rated according to its data-carrying ability (bandwidth) and is given a "category" number. We strongly recommend using a Category 5 (CAT5) cable for all Ethernet 10BaseT/100BaseT connections. For convenient and reliable networking, we recommend that you purchase commercially manufactured cables (cables with connectors already attached).

To connect an EBC (or PC) to a hub or repeater, use a **patch cable** (sometimes called a straight-through cable). The cable used to connect a PC *directly* to an EBC or to connect two hubs is referred to as a **crossover cable**. Some hubs provide a crossover port which eliminates the need for a crossover cable.

10BaseT/100BaseT



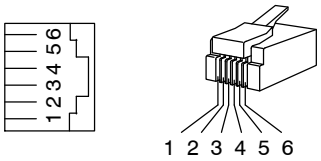
8-pin RJ45 Connector (8P8C)



This diagram illustrates the standard wire positions in the RJ45 connector. We recommend that you use only **Category 5**, UTP cable.

Serial Port (RS-232)

Serial Port



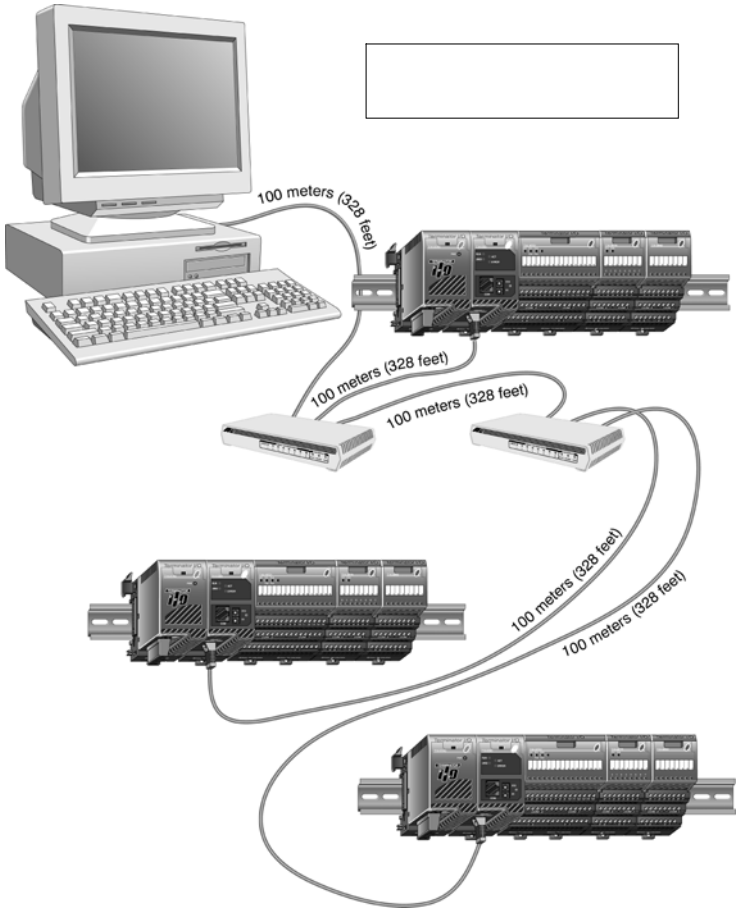
The Serial Port on the EBC can be used to communicate with operator interfaces or ASCII devices. The T1H-EBC100 also supports MODBUS RTU serial protocol.

Use Automationdirect.com cable Part Number D2-DSCBL to connect your PC to the RJ12 Serial Port.

Serial Port Pinout	
Pin	**Signal**
1	0V
2	+5V
3	RXD
4	TXD
5	RTS
6	CTS

Maximum 10BaseT / 100BaseT Cable Length

The **maximum distance** per **10BaseT**/**100BaseT** cable segment is **100 meters** or **328 feet**. Repeaters extend the distance. Each cable segment attached to a repeater can be 100 meters long. Two repeaters connected together extend the total range to 300 meters.



T1H-EBC Specifications

T1H-EBC Ethernet Base Controller

Module Type	Ethernet slave
Communications	10BaseT
Auto-configuring	I/O type/position automatically identified during power-up
Ethernet Protocols	UDP/IP, IPX
Ethernet Port	RJ45
Node Address	1 to 99 (decimal) set by rotary switches or software (0 used for setting address via software only)
Link Distance	100 meters (328 feet)
Data Transfer Rate	10Mbps
LED Indicators Note: All indicators re-initialize during power-up.	MODULE GOOD (green): On = module passed diagnostic check during last power-up Fast blink = configured I/O module no longer reporting (see auto-configuring, above) Slow blink = unconfigured I/O module added to system (see auto-configuring, above) LINK GOOD (green): On = 10Base-T link pulses are being received ACTIVITY (red): On= Ethernet network activity detected ERROR (red): On = watchdog timer timeout represents hardware, communications, or network fault; power-on reset or reset within master device software
Serial Communications Port	RJ12, RS232C K-Sequence protocol, ASCII (not functional when used with H2-ERM / H4-ERM)
Base Power Requirement	350mA @ 5VDC

General Specifications

Installation Requirements	mounts to right of first power supply
Operating Temperature	32° F to 131° F (0° C to 55° C)
Storage Temperature	-4° F to 158° F (-20° C to 70° C)
Relative Humidity	5 to 95% (non-condensing)
Environmental Air	No corrosive gases, pollution level = 2 (UL 840)
Vibration	MIL STD 810C 514.2
Shock	MIL STD 810C 516.2
Noise Immunity	NEMA ICS3-304 Impulse noise 1us, 1000V FCC class A RFI (144MHz, 430MHz, 10W, 10cm)

T1H-EBC100 Specifications

T1H-EBC100 Ethernet Base Controller

Module Type	Ethernet slave
Communications	10/100BaseT
Auto-configuration	I/O type/position automatically identified during power-up
IP Configuration	-Obtain an IP address from a DHCP Server automatically at power-up (Default); -Dedicated IP address using NetEdit3 or HTML configuration
Ethernet Protocols	TCP/IP, UDP/IP, IPX, MODBUS TCP/IP
Ethernet Port	RJ45
Node Address	1 to 99 (decimal) set by rotary switches or software (0 used for setting address via software only)
Link Distance	100 meters (328 feet)
Data Transfer Rate	10/100Mbps
LED Indicators Note: All indicators re-initialize during power-up.	MODULE GOOD (green): On = module passed diagnostic check during last power-up Fast blink = configured I/O module no longer reporting (see auto-configuring, above) Slow blink = unconfigured I/O module added to system (see auto-configuring, above) LINK/ACTIVITY (green): On= Ethernet network activity detected 100MBIT (green): On= Ethernet activity is auto-detected at 100bps Off = (with LINK/ACTIVITY On) Ethernet activity is auto-detected at 10Mbps ERROR (red): On = watchdog timer timeout represents hardware, communications, or network fault; power-on reset or reset within master device software SERIAL TX (green): On= EBC RJ12 serial port is transmitting SERIAL RX (green): On= EBC RJ12 serial port is receiving
Serial Communications Port	RJ12, RS232C K-Sequence protocol, ASCII, MODBUS RTU serial (not functional when used with H2-ERM / H4-ERM at this time)
Base Power Requirement	350mA @ 5VDC

General Specifications

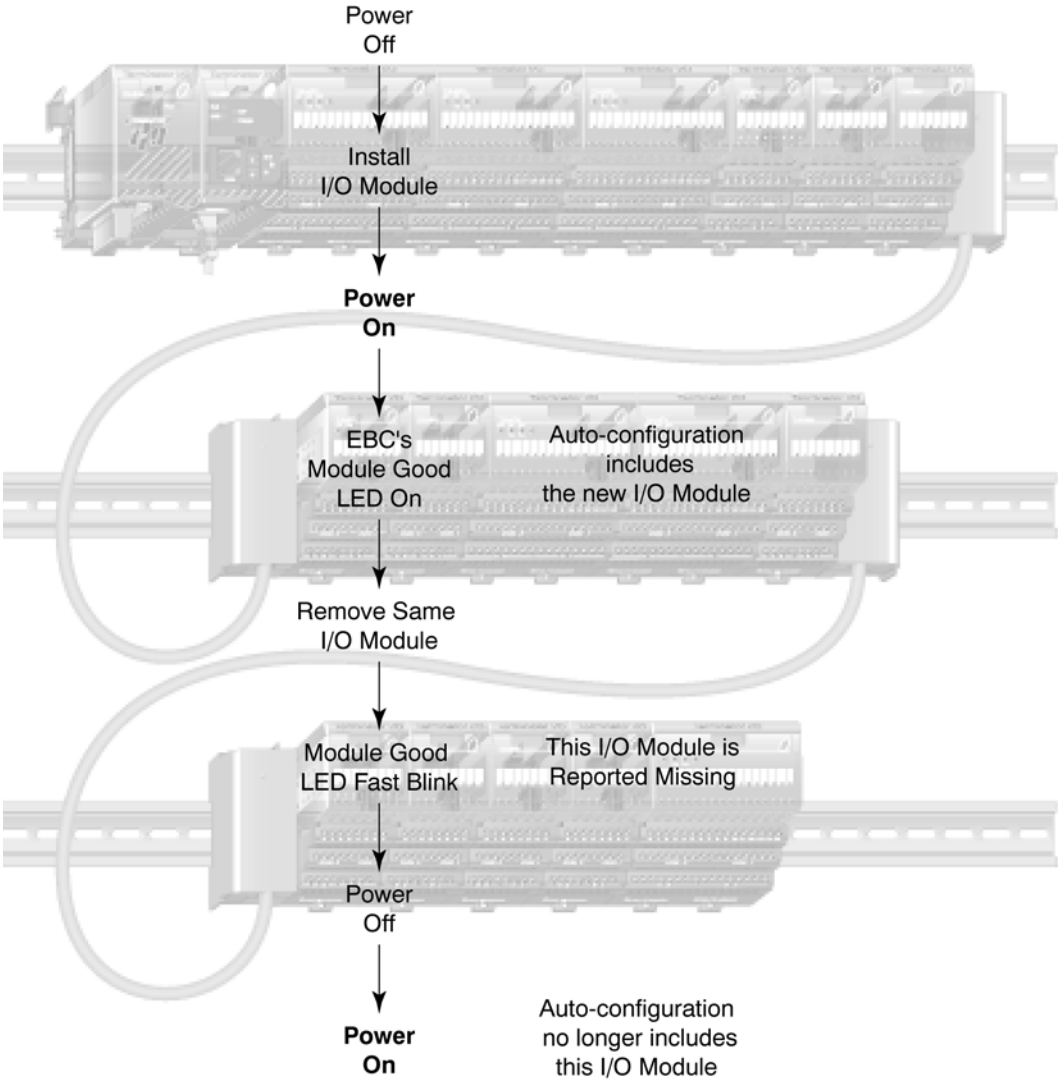
Installation Requirements	mounts to right of first power supply
Operating Temperature	32° F to 131° F (0° C to 55° C)
Storage Temperature	-4° F to 158° F (-20° C to 70° C)
Relative Humidity	5 to 95% (non-condensing)
Environmental Air	No corrosive gases, pollution level = 2 (UL 840)
Vibration	MIL STD 810C 514.2
Shock	MIL STD 810C 516.2
Noise Immunity	NEMA ICS3-304 Impulse noise 1us, 1000V FCC class A RFI (144MHz, 430MHz, 10W, 10cm)

LED Indicators and Hot Swapping I/O Modules

The “Hot Swap” feature allows Terminator I/O modules to be replaced with Terminator I/O system power ON. Be careful not to touch the terminals with your hands or any conductive material to avoid the risk of personal injury or equipment damaged. *Always remove power if it is equally convenient to do so.*

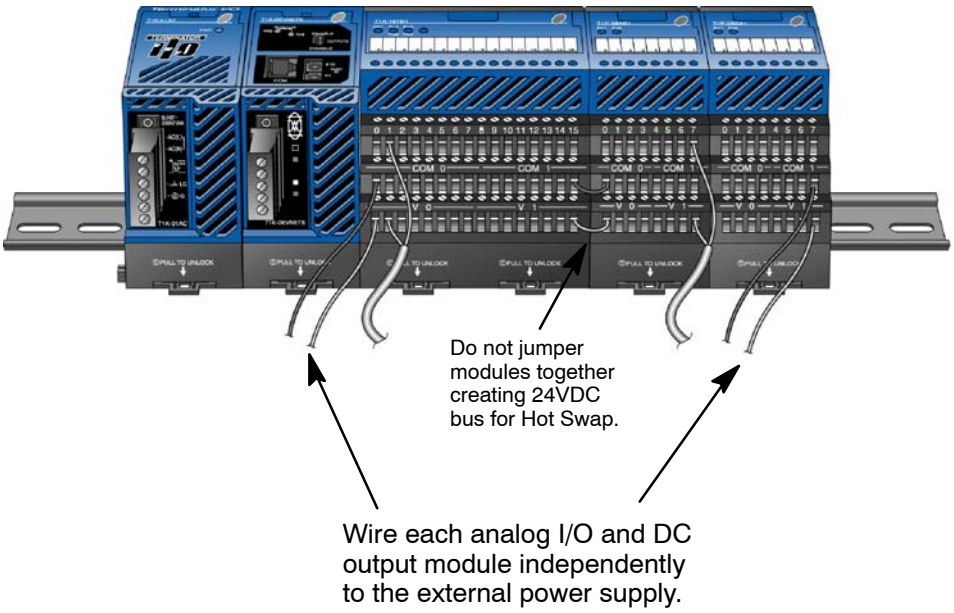
WARNING: Only authorized personnel fully familiar with all aspects of the application should replace an I/O module with system power ON.

LED Indicators when Hot Swapping an I/O Module



Check External 24VDC Wiring Before Hot Swapping

Before "Hot Swapping" an analog I/O module or a DC output module in a Terminator I/O system, make sure that each of the analog I/O and DC output module's 24VDC and 0VDC base terminals are wired directly to the external power supply individually (see diagram below). If the external 24VDC / 0VDC is jumpered from base to base in a daisy chain fashion, and an analog I/O or DC output module is removed from its base, the risk of disconnecting the external 24VDC to the subsequent I/O modules exists.



Hot Swap: I/O Module Replacement

The following steps explain how to "Hot Swap" an I/O module.

1. Remove I/O module from base. (If necessary, refer to the Terminator I/O Installation & I/O Manual for steps on removing an I/O module).
2. The EBC Module Good LED will begin to *fast blink*.
3. Install a new I/O module with the **exactly the same part number**.
4. Verify that the EBC LEDs have returned to normal.

Ethernet Standards

Various institutes and committees have been involved in establishing Ethernet data communication standards. These specification standards assure Ethernet network compatibility for products from a broad variety of manufacturers.

The EBC module complies with American National Standards Institute (ANSI) and Institute of Electrical and Electronic Engineers standard ANSI/IEEE 802.3, Carrier Sense Multiple Access with Collision Detection (CSMA/CD) Access Methods and Physical Layer Specifications. This standard has been adopted by the International Organization for Standardization (ISO) as document ISO/IEC 8802-3.

The Electronic Industries Association (EIA) and Telecommunications Industries Commercial Building Telecommunications Wiring Standard designated EIA/TIA-568A defines implementation of 10Base-T (twisted pair) Ethernet communications.

The same two organizations produced EIA/TIA TSB40-Additional Transmission Specifications for Unshielded Twisted-Pair Connecting Hardware. The purpose of this document is to specify transmission performance requirements and connecting hardware requirements.

3

Configuring the T1H-EBC(100) Using NetEdit3

In This Chapter. . . .

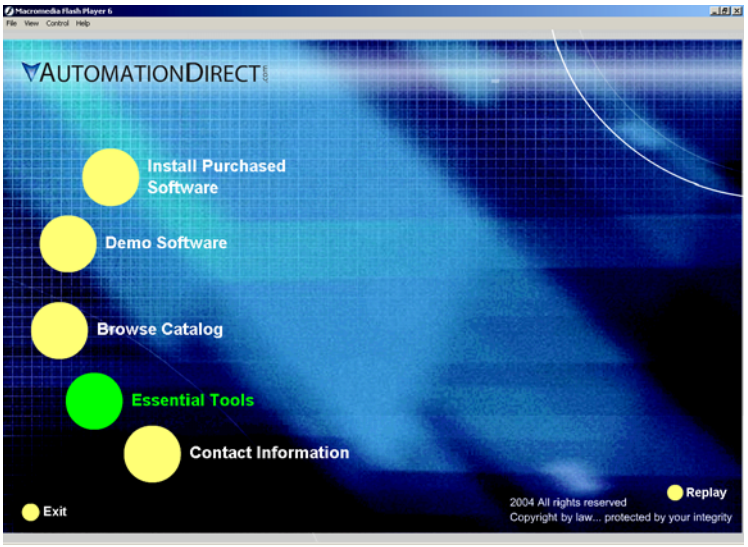
- NetEdit3 Software
- Using NetEdit3
- Locating the MAC Address Label

NetEdit3 Software

NetEdit3 is a software utility which can be used to set network identifiers (Module ID or IP Address), configure the EBC serial port, perform diagnostic and troubleshooting tasks and upgrade the firmware in the EBC module if necessary. The T1H-EBC100 requires NetEdit 3.x or later.

Installing NetEdit3

You can install NetEdit3 on Windows98/ME/2000/XP™ or Windows NT4™. NetEdit3 is included with this manual on the AutomationDirect Software Product Showcase CD (also available online at www.automationdirect.com). After inserting the CD into the drive, the following window will appear.



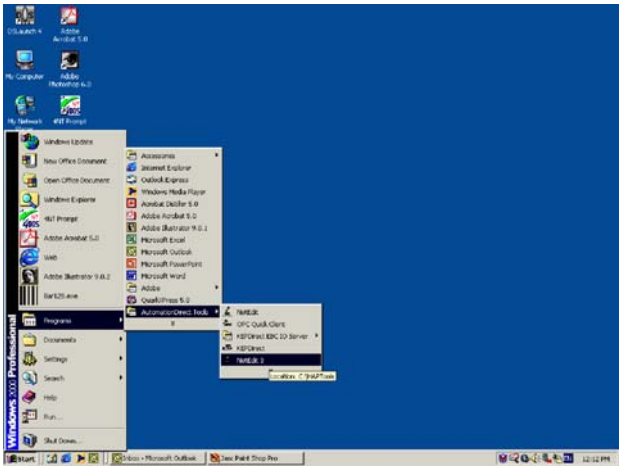
Click on the Essential Tools button. The following window will be displayed.

Click on Install NetEdit3. A series of windows will step you through the installation process. Fill in the necessary information as the installation wizard prompts through the install. In the Setup Type window, select Typical setup. This setup type is recommended for most users. The installation process places NetEdit3 in the C:\HAPTools directory (default).

Launching NetEdit3

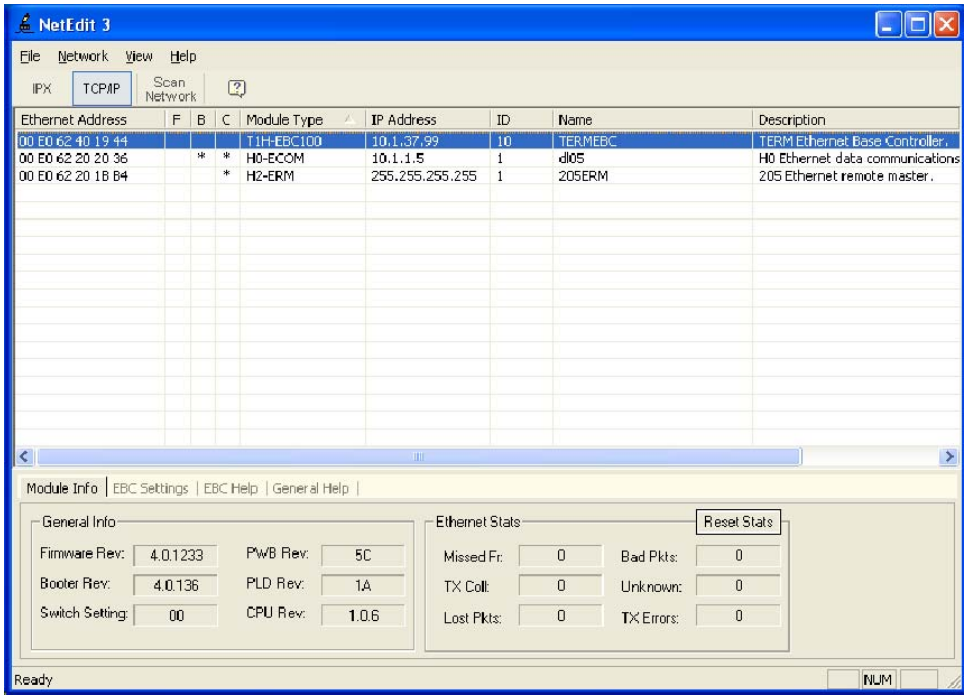
There are three methods to launch NetEdit3.
The three methods are:

- using the Windows Start menu Programs>AutomationDirect Tools> NetEdit3 as shown below
- launching ***Direct***Soft32 (if installed), from the programming window, select PLC>Tools>NetEdit3
- launching ***Direct***Soft32 (if installed), then select Utilities>NetEdit3



The NetEdit3 Screen

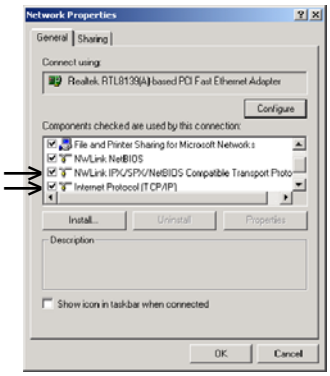
Starting NetEdit brings up the screen below. All NetEdit3 functions are accessed from this screen.



Adding Network Protocol Support to the NetEdit3 PC

You may have already set up your PC with selected networking protocols for Ethernet communications. If not, you will need to select the protocols now for communication with the Ethernet modules. We strongly recommend that you include the IPX protocol. For Windows 2000, go from My Computer on your Windows desktop to Control Panel. Double click on Network and Dial-up Connections, then double click on the desired Network Device to see the installed Protocols. If IPX is not listed among the protocols already loaded, add it now by clicking on the Install button. For Windows XP, go from Start>Settings>Control Panel. The steps are the same as Windows 2000 from this point.

Add the TCP/IP protocol if it is necessary for your application. The TCP/IP selection will give you support for the UDP/IP protocol. Also, add the IPX protocol if it is not already active.



NOTE: We strongly recommend you load IPX protocol on your PC for the connection from your PC to the ethernet modules. Use UDP/IP in your application, if required, but also add IPX to your list of active protocols. Having IPX loaded on your PC gives you a backup for troubleshooting communication problems.

Using NetEdit3

This section steps through the features and uses of NetEdit3. We will describe the individual segments of the NetEdit3 screen and the function of each.

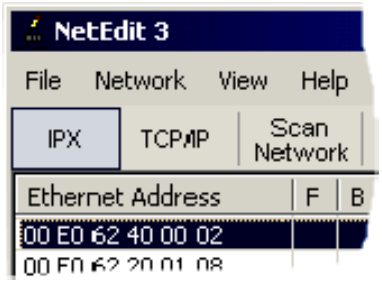
NOTE: Your PC-based Control software may be capable of configuring the EBC module. If so, please refer to the appropriate documentation for that software product to determine the best method to configure the EBC. Depending on which software you are using, it may not be necessary to use NetEdit3.

Ethernet Communication Protocol

In the upper left corner of the NetEdit3 screen, you will find buttons labeled *IPX* and *TCP/IP*. The EBC module understands these protocols. Both protocols are *permanently resident* in the firmware of the module.

When you click on one of these buttons, you are selecting the protocol you want your PC to use to communicate with the EBC module. You are not telling the module which protocol to use, because it is using both protocols all the time. IPX is a Novell standard in widespread use, and UDP/IP is a popular protocol supported by the TCP/IP suite of protocols in your PC.

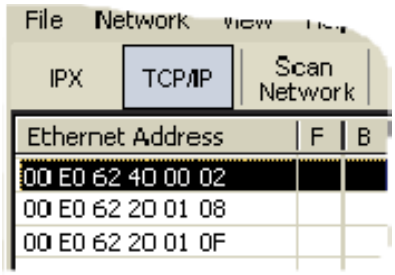
The figure to the right shows the Protocol selection buttons in the upper left corner of the NetEdit3 screen. The choice you make here tells **your PC** which protocol to send to the EBC to link NetEdit3 to the module.



Some PC-based control software products may support only one of these protocols. Read the documentation for your software to be sure the protocol you select is supported.

Ethernet Address

The upper left section of the NetEdit3 screen displays the *Ethernet Address* of the modules currently on the network.

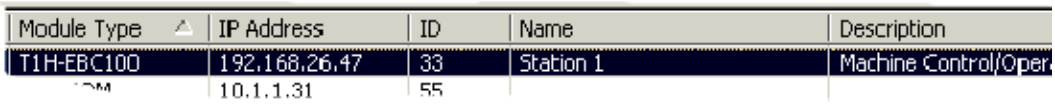


If modules are added or removed from the network, click on the *Scan Network* button to update the list. Notice that the MAC Address is the factory-assigned address that is on the permanent label on the module.

Select a specific module here by clicking on the MAC Address or by using the arrow keys. The selected module is highlighted.

NOTE: The Module window may list the MAC Addresses of devices not covered by this manual.

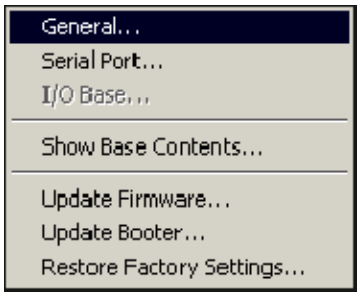
Module Type, IP Address and ID



The upper mid section of the NetEdit3 screen displays the *Module Type, IP Address,* module *ID, Name* and *Description* of the modules currently on the network.

A new EBC will have an IP Address of 0.0.0.0, a Module ID of 0 (zero), and no Module Name or Description. To assign or change a module an IP address, ID, name or description refer to the EBC Settings>General Information description later in this section.

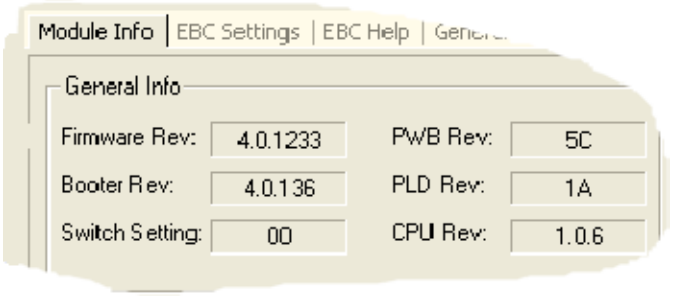
Right clicking on an EBC module listed on the NetEdit3 screen will display the window to the right. This is an alternative to using the Module Info or EBC settings tabs (shown below) to access the module's configuration settings. The settings are discussed later in this section.



Module Info> General Information

When the Module Info tab is selected, the *General Info* box lists the selected module's Firmware Revision, Booter Revision, DIP Switch Setting, PWB Revision, PLD Revision and CPU Revision.

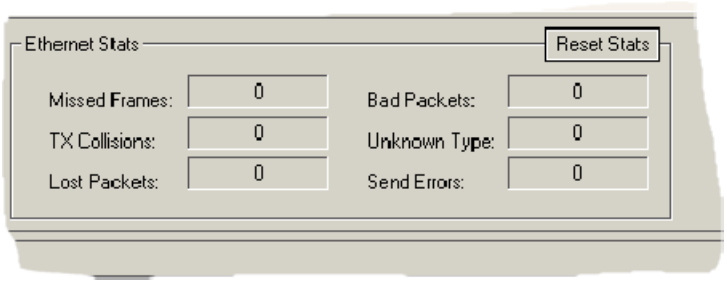
This box is in the lower left section of the NetEdit3 screen.



Module Info> Ethernet Stats

When the Module Info tab is selected, the *Ethernet Stats* box displays statistics related to the selected module's communication errors. Click on the Reset Stats button to reset all categories to 0 (zero).

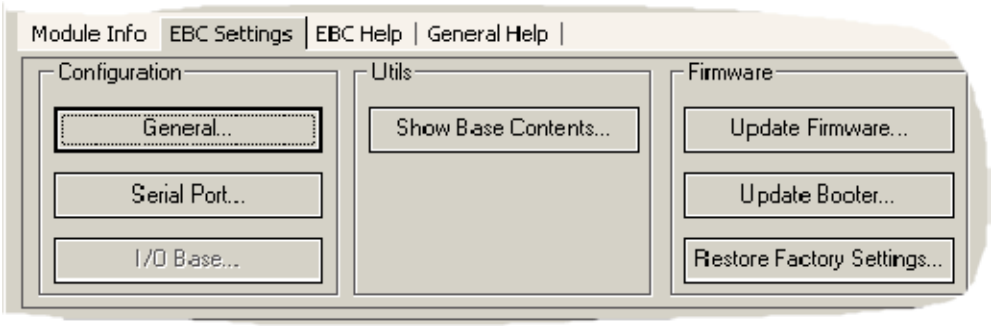
This box is in the lower middle section of the NetEdit3 screen.



EBC Settings

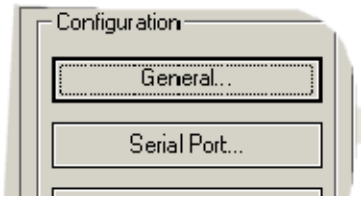
When the EBC Settings tab is selected, the selected module's Configuration, Utilities and Firmware tools can be accessed.

This box is in the lower middle section of the NetEdit3 screen.

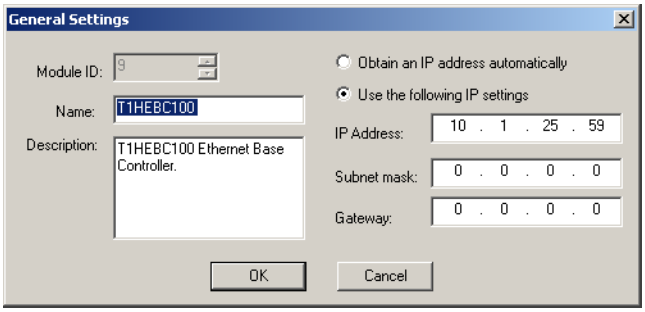


EBC Settings> Configuration> General

Clicking the General button in the EBC Settings>Configuration box brings up the General Settings window below.



See Errata Sheet at the beginning of this file. Add this note:
NOTE: T1H-EBC100 modules have a DIP switch located under the cover on the left side. Turning this switch on will reset the IP Address, subnet and gateway to the factory defaults. (T1H-EBC modules do not have a DIP switch.)



The General Settings box allows you to assign a **Module ID**. Module IDs must be unique for each EBC, but they do not have to be in sequence. The module's Node Address rotary switches must both be set to zero to allow NetEdit3 to set a Module ID. Do not use address zero for communications.

The **Name** field and **Description** field are optional.

To set an **IP Address**, highlight the number in each of the four boxes, and overwrite the number. Use the twelve-digit number assigned to the EBC module by your network administrator. If you change the IP Address, do not use the number "255" in any field. Doing so will cause communication problems.

The OK button sends all the entries to the module's flash memory.

IP Address

An **IP Address** is assigned to the EBC module if your network will be using the TCP/IP or MODBUS TCP/IP (T1H-EBC100 only) protocols. If you have a separate dedicated network for your EBCs, you may be able to use the Module ID identifier (IPX protocol) for communications instead of an IP address. To set an IP Address, use the twelve-digit number assigned to the EBC module by your network administrator. If you change the IP address, do not use the number "0" or "255" in any field. Doing so will cause communications problems. The **valid settings are 1 through 254.** The module ships from the factory with an IP Address of 0.0.0.0. This is not a usable IP Address for communications. Click on *Use the following IP settings* radio button before clicking on the OK button to write the updated settings to the module's flash memory. It is extremely important not to have duplicate IP Addresses on your network.

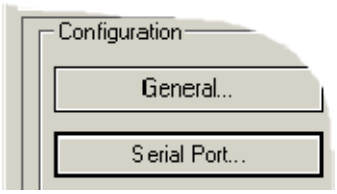
Example

Client Subnet Mask:	255.255.0.0	
Valid Client IP Address	192.168.50.2	**1-254 Valid settings for Bold number fields (Do not duplicate)**
Valid EBC IP Address	192.168.55.5	
Valid EBC IP Address	192.168.70.15	

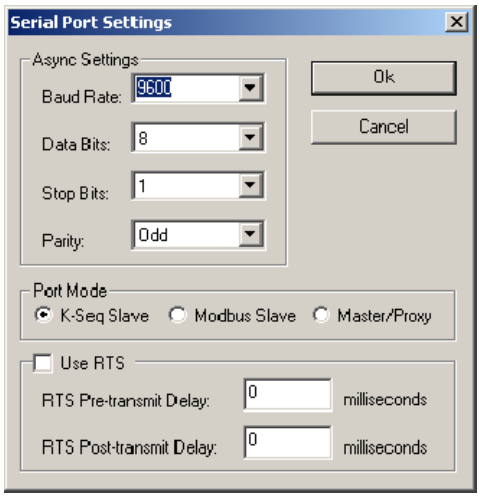
WARNING: If you are using the T1H-EBC100, be sure to read Chapter 5 concering DHCP issues.

EBC Settings> Configuration> Serial Port

Clicking the Serial Port button in the EBC Settings>Configuration box brings up the Serial Port Settings window below.



On the *Serial Port Settings* window, make any necessary changes to the serial communication parameters. After making changes, be sure to click on the OK button. Also, Be sure these parameters match the parameters of the serial device with which you are communicating.

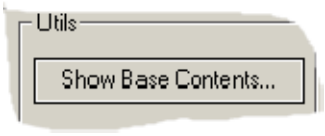


Note: Some PC-based Control software packages may automatically overwrite settings selected here. Refer to the documentation for your PC-based Control software.

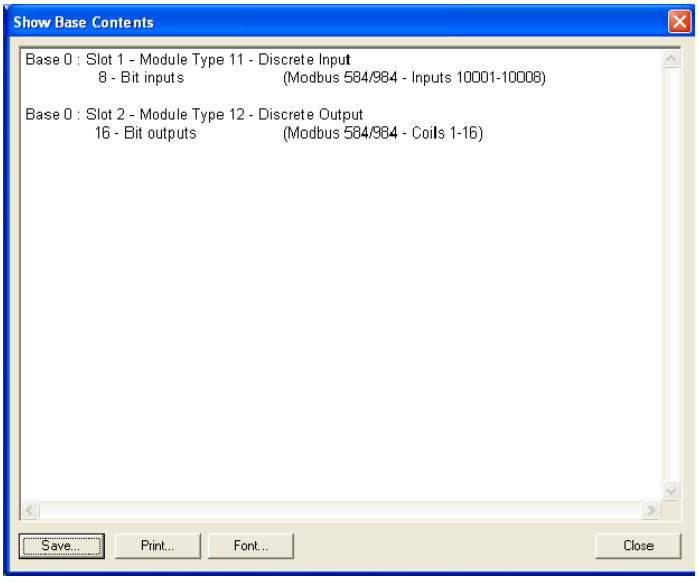
The OK button sends all the entries to the module's flash memory.

EBC Settings> Utils>Show Base Contents

Clicking the Show Base Contents button in the EBC Settings>Utilities box brings up the Show Base Contents Window shown below.

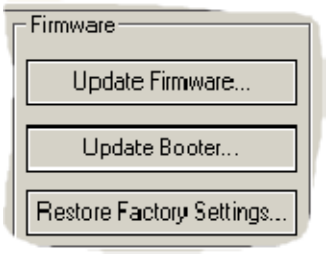


This function queries the EBC for a list of I/O modules it has in its base. The Module Type and the MODBUS 584/984 addressing will be listed as well. This will help confirm that the EBC can recognize all the I/O modules connected to the EBC controller. The Show Base Contents information can be saved as a (*.txt) file or printed for reference or future use.

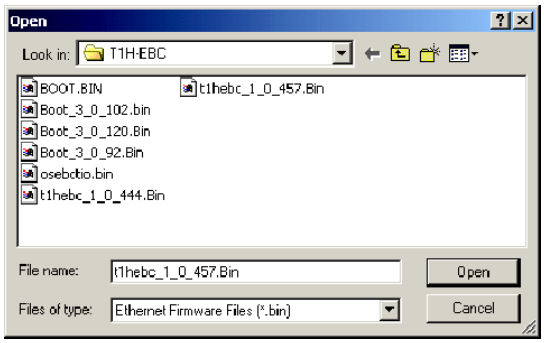


EBC Settings> Firmware

The functions in the Firmware box are used to update the selected module's Firmware and Booter versions. The Restore Factory Settings buttons resets the selected module's IP address, ID, Name and Description to factory defaults. Refer the sections on the next page to determine if updates are necessary.

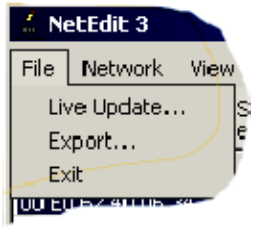


Clicking on the either of the Update buttons opens the appropriate EBC folder within the Images folder, which is created during the install of NetEdit3. The Images folder is located in the same folder as NetEdit3.exe. Each module folder contains the module's firmware and boot loader files. The next section discusses keeping the firmware files up to date.

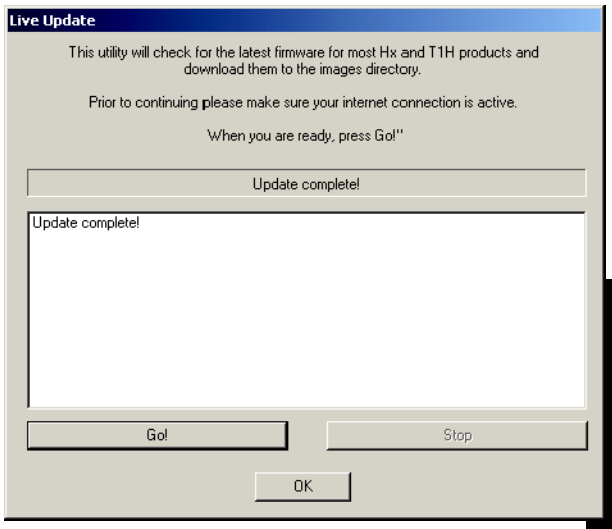


FileMenu>
Live Update

The Live Update will retrieve the latest firmware and boot loader files from the Host Engineering web site and place them in the NetEdit3 Images folder that was created during the install of NetEdit3. The feature requires that you have a functional Internet connection (dial-up or broadband). If the Images folder does not exist on your PC, it will be created as part of the retrieval process.

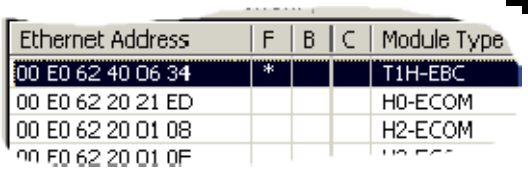


When you click the Go! button on the Live Update window, NetEdit3 will compare the version information of the files on the Host Engineering web site against the files you have locally on your PC, and it will download any newer files. Once this process is complete, NetEdit3 will rescan the devices on your network and refresh the "F" and "B" columns next to the listed devices.



F / B / C
Columns

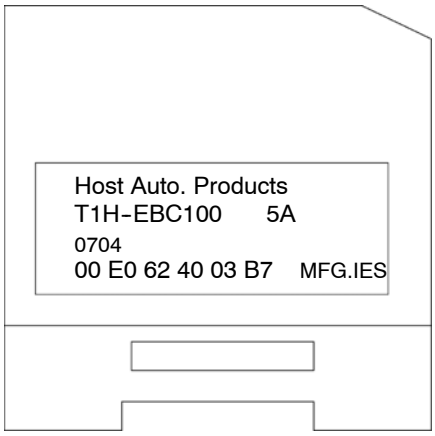
The F, B, and C columns are provided to signify potential issues with devices on the network.



The "F" column will display an asterick beside any device whose firmware is older than its firmware file in your Images folder.
The "B" column will display an asterick beside any device whose boot loader is older than its boot loader file in your Images folder.
The "C" column will display an asterick beside any device that has a configuration conflict with another device on the network. Duplicate module IDs (that are non-zero) and duplicate IP Addresses (that are not 255.255.255.255) will report as conflicts.

Locating the MAC Address Label

Factory-assigned MAC Address



MAC Address

A unique Media Access Control (MAC) Address is assigned to each module at the factory and cannot be changed. It is a twelve digit number, and it is printed on a label permanently attached to the EBC circuit board. The MAC address label can be viewed through the translucent module cover. NetEdit3 and HTML Configuration can be used to read the MAC address.

4

Modbus® TCP for T1H-EBC100

In This Chapter. . . .

- Modbus TCP/IP
- Supported Modbus Function Codes
- Modbus 584/984 Addressing
- Modbus 584/984 Addressing for Function 3 Clients
- T1H-EBC100 System Memory
- Current / Last State Error Codes
- Extended Error Codes
- Analog Input Module Configuration
- Analog Output Module Configuration

Modbus TCP

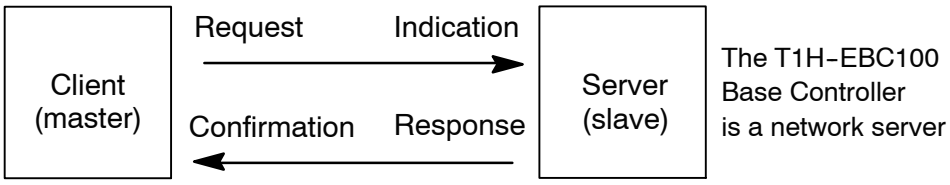
Modbus TCP is essentially the serial MODBUS RTU protocol encapsulated in a TCP/IP wrapper. Modbus RTU is used for serial communications between a master and slave(s) devices. Modbus TCP is used for TCP communications between client and server devices on an Ethernet network. The TCP version of Modbus follows the OSI Network Reference Model.

Client / Server Model

The Modbus messaging service provides a Client/Server communication between devices connected on an Ethernet TCP network. This client / server model is based on four type of messages:

- Modbus Request – the message sent on the network by the Client to initiate a transaction
- Modbus Confirmation – the Response Message received on the Client side
- Modbus Indication – the Request message received on the Server side
- Modbus Response – the Response message sent by the Server

Client / Server Model



Protocol Description

A typical Modbus TCP/IP frame consists of the following fields:

The **MBAP header** (MODBUS Application Protocol header) is seven bytes long. It consists of the following fields.

- Transaction Identifier – It is used for transaction pairing, the Modbus server copies in the response the transaction identifier of the request. (2 bytes)
- Protocol Identifier – It is used for intra-system multiplexing. The Modbus protocol is identified by the value 0. (2 bytes)
- Length – The length field is a byte count of the following fields, including the Unit Identifier and data fields. (2 bytes)
- Unit Identifier – This field is used for intra-system routing purpose. It is typically used to communicate to a Modbus or a Modbus+ serial line slave through a gateway between an Ethernet TCP/IP network and a Modbus serial line. This field is set by the Modbus Client in the request and must be returned with the same value in the response by the server. (1 byte)

The MBAP header provides some differences compared to the Modbus RTU application data unit used on serial line.

- The Modbus “slave address” field usually used on Modbus Serial Line is replaced by a single byte “Unit Identifier” within the MBAP Header. The “Unit Identifier” is used to communicate via devices such as bridges, routers and gateways that use a single IP address to support multiple independent Modbus end units.
- All Modbus requests and responses are designed in such a way that the recipient can verify that a message is finished. For function codes where the Modbus PDU has a fixed length, the function code alone is sufficient. For function codes carrying a variable amount of data in the request or response, the data field includes a byte count.
- Protocol Identifier – It is used for intra-system multiplexing. The Modbus protocol is identified by the value 0. (2 bytes)

The **function code field** of a message contains 8 bits. Valid function codes are in the range of 1 – 255 decimal. The function code instructs the slave what kind of action to take. Some examples are to read the status of a group of discrete inputs; to read the data in a group of registers; to write to an output coil or a group of registers; or to read the diagnostic status of a slave.
When a slave responds to the master, it uses the function code field to indicate either a normal response or that some type of error has occurred. For a normal response, the slave echoes the original function code. In an error condition, the slave echoes the original function code with its MSB set to a logic 1.

The **data field** is constructed using sets of two hexadecimal digits in the range of 00 to FF. According to the network’s serial transmission mode, these digits can be made of a pair of ASCII characters or from one RTU character.
The data field also contains additional information that the slave uses to execute the action defined by the function code. This can include internal addresses, quantity of items to be handled, etc.
The data field of a response from a slave to a master contains the data requested if no error occurs. If an error occurs, the field contains an exception code that the master uses to determine the next action to be taken. The data field can be nonexistent in certain types of messages.

ModScan32 is a Windows based application program that can be used as a Modbus master (client) to access and change data points in a connected slave (server) device (i.e. T1H-EBC100).
The utility is ideally suited for quick and easy testing of Modbus TCP network server devices. Visit www.win-tech.com to download a free ModScan32 trial demo and for more information on ModScan32.

Supported Modbus Function Codes

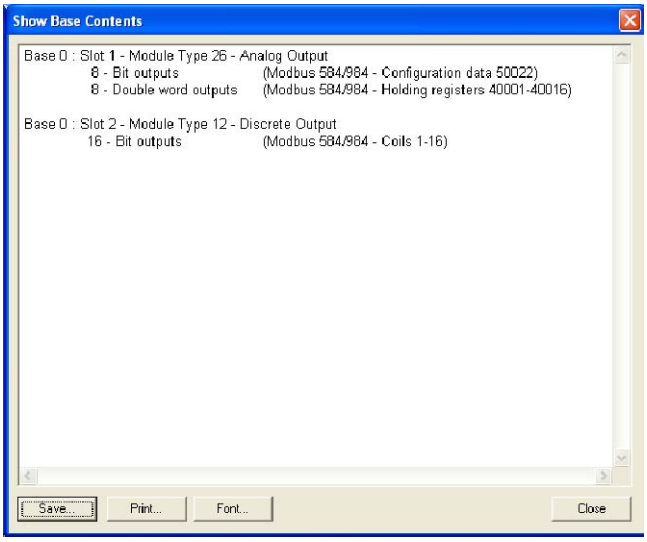
The following Modbus function codes are supported by the T1H-EBC100 base controller.

Modbus Function Code	Function
01	Read Output Table
02	Read Input Table
03	Read Holding Registers (when addressing mode is 584/984, this function is used to access analog output registers)
04	Read Input Registers (when addressing mode is 584/984, this function is used to access analog input registers)
05	Force Single Output
06	Preset Single Registers
08	Loop back / Maintenance
15	Force Multiple Outputs
16	Preset Multiple Registers

Modbus 584/984 Addressing

Modbus Data Type		T1H-EBC100				
		Range (Decimal)	Points		Memory Type	Access
Coil		**1 - 1024**	1024		**Discrete Output**	Read / Write
		1025 - 10000	-		Reserved	-
Input		**10001 - 11024**	1024		**Discrete Input**	Read only
		11025 - 20000	-		Reserved	
Modbus Data Type						
		Range (Decimal)	**Words (16-bit)**	**Channel (32-bit)**	**Memory Type**	
Input Register	**Analog Input** (see table below)	**30001 - 30512**	512	256	Analog Input Register	Read only
	Input Register	30513 - 32000	-	-	Reserved	-
	Bit Input Register	**32001 - 32064**	64	32	Discrete Input Bit Register	Read only
	Input Register	32065 - 37000	-	-	Reserved	-
Hold Register	**Analog output** (see table below)	**40001 - 40512**	512	256	Analog Output Register	Read / Write
	Hold Register	40513 - 42000	-	-	Reserved	-
	Bit Output Register	**42001 - 42064**	64	32	Discrete Output Bit Register	Read / Write
	Hold Register	42065 - 44000	-	-	Reserved	-

Note: NetEdit3 Show Base Contents function will list the MODBUS addressing for each I/O module on the base. For the analog I/O, the module Configuration Data registers are also listed. Refer to Chapter 3 for information on NetEdit3.



Modbus 584/984 Addressing for Function Code 3 Clients

The following memory map offers duplicate registers from the 30001 range and Bit memory data type into the 411000 range for clients/masters that only support function code 3. These ranges are word level data only.

Modbus Word Data Type		T1H-EBC100				
		Range (Decimal)	**Words**		**Memory Type**	**Access**
Coil		**411000 - 411063**	64		**Discrete Output**	Read / Write
		411064 – 411124	–		Reserved	–
Input		**411625 - 411688**	64		**Discrete Input**	Read only
		411689 – 412062	–		Reserved	
Modbus Word Data Type		**Range (Decimal)**	**Words (16-bit)**	**Channel (32-bit)**	**Memory Type**	
Input Register	**Analog Input (see table below)**	**412251 - 412762**	512	256	Analog Input Register	Read only
	Input Register	412763 – 414250	–	–	Reserved	–
	Bit Input Register	**414251 - 414314**	64	32	Discrete Input Bit Register	Read only
	Input Register	414315 – 419250	–	–	Reserved	–
Hold Register	**Analog output (see table below)**	**40001 - 40512**	512	256	Analog Output Register	Read / Write
	Hold Register	40513 – 42000	–	–	Reserved	–
	Bit Output Register	**42001 - 42064**	64	32	Discrete Output Bit Register	Read / Write
	Hold Register	42065 – 44000	–	–	Reserved	–

T1H-EBC100 System Memory

	T1H-EBC100			
	Modbus Addressing Range (Decimal)	**Words (16-bit)**	**Word Descriptions**	**Access**
Module Version Information	**37001 - 37006 (419251 - 419256)***	6	1 - OS Major Version 2 - OS Minor Version 3 - OS Build Version 4 - Booter Major Version 5 - Booter Minor Version 6 - Booter Build Version	R only
	37007 - 37010 (419257 - 419260)	-	Reserved	-
Device Data	**37011 - 37100 (419261 - 419350)***	90	1 - Version of Device 2 - Family 3 - Processor 4 - Module Type 5 - Status Code (6-8) - Ethernet Address 9 - RAM Size 10 - Flash Size 11 - Batt Switch 12 - DIP Settings 13 - Media Type (14-15) - Reserved 16 - Reserved 17 - Reserved 18 - Model Number 19 - Ethernet Speed 20 - Reserved 21 - IO Total Byte Count 22 - Bit Input Byte Count 23 - Bit Output Byte Count 24 - Non-bit Input Byte Count 25 - Non-bit Output Byte Count (26-90) - Reserved	R only
I/O Module ID's	**37101 - 37132 (419351 - 419382)***	32 (1 word per slot)	I/O module ID numbers per slot location	R only
	37133 - 37200 (419383 - 419450)	-	Reserved	-
Module Information	**37201 - 37328 (419451 - 419578)***	128 (4 words per slot)	1 - Bit Input Count 2 - Bit Output Count 3 - Non-bit Input Count 4 - Non-bit Output Count	R only
	37329 - 37400 (419579 - 419650)	-	Reserved	-

*For clients that only support function code 3 to read word data.

(continued)

T1H-EBC100 System Memory (continued)

	T1H-EBC100			
	Modbus Addressing Range (Decimal)	**Words (16-bit)**	**Word Descriptions**	**Access**
EBC Dynamic Module Data	**410001 - 410020**	20	1 - See Error Codes on p. 4-9. 2 - Error bit-per-slot for first 16 slots If any bit is set, see extended error info of Module Status data for specific problem 3 - Error bit-per-slot for second 16 slots (if present) If any bit is set, see extended error info of Module Status data for specific problem NOTE: Any write to [1], [2], or [3] above will clear the module / slot errors. 4 - Flags: Bit 0: If 1, module has rebooted since this bit was cleared, a write to the Flags word with this bit set will clear this reboot bit. Bit 1: Write Only Bit - A write to the Flags word with this bit set will cause the base to be re-scanned. Bit 2: Write Only Bit - If Bit 1 is set to do rescan, this bit is used to indicated if RESCAN_LEAVE_IMAGE_RAM or RESCAN_CLEAR_IMAGE_RAM Bit 3-7: Reserved 5 - Reboot Count (LSW) - Read Only 6 - Reboot Count (MSW) - Read Only 7 - Link Monitor Timeout - 0 to disable 8-20 - Reserved	R / W
Configuration Data	**410021 - 410052**	32	1 word per slot to read/write module configuration data (See Analog I/O Module Configuration tables at the end of this chapter for bit definitions)	R / W
	410053 - 425536	-	Reserved	-

See Errata Sheet at the beginning of this file. This information for "4 - Flags:" has been revised.

(continued)

T1H-EBC100 System Memory (continued)

	T1H-EBC100			
	Modbus Addressing Range (Decimal)	**Words (16-bit)**	**Word Descriptions**	**Access**
I/O Module Status	**37401 - 38040 (419651 - 420290)***	640 (20 words per slot)	1 - Flags with bits indicating presence of Error, Warning, Info Values Bit 0: If set, indicates that Error Value is non-zero Bit 1: If set, indicates that Warning Value is non-zero Bit 2: If set, indicates that Info Value is non-zero Bit 3: Reserved Bit 4: If set, indicates that Extended error info is present Bit 5: Reserved Bit 6: Reserved Bit 7: Reserved For Words 2-4, refer to Current/Last State Error Codes Table (p.4-9) 2 - Error Code 3 - Warning Code 4 - Info Code For Words 5-20, refer to Extended Error Codes Table (p.4-9) 5 - Extended Error Code 1 (i.e channel 1 of an analog module) 6 - Extended Error Code 2 (i.e channel 2 of an analog module) 7 - Extended Error Code 3 (i.e channel 3 of an analog module) 8 - Extended Error Code 4 (i.e channel 4 of an analog module) 9 - Extended Error Code 5 (i.e channel 5 of an analog module) 10 - Extended Error Code 6 (i.e channel 6 of an analog module) 11 - Extended Error Code 7 (i.e channel 7 of an analog module) 12 - Extended Error Code 8 (i.e channel 8 of an analog module) 13 - Extended Error Code 9 (i.e channel 9 of an analog module) 14 - Extended Error Code 10 (i.e channel 10 of an analog module) 15 - Extended Error Code 11 (i.e channel 11 of an analog module) 16 - Extended Error Code 12 (i.e channel 12 of an analog module) 17 - Extended Error Code 13 (i.e channel 13 of an analog module) 18 - Extended Error Code 14 (i.e channel 14 of an analog module) 19 - Extended Error Code 15 (i.e channel 15 of an analog module) 20 - Extended Error Code 16 (i.e. channel 16 of an analog module)	R only
	38041 - 40000 (420291 - 422250)	-	Reserved	-

*For clients that only support function code 3 to read word data.

(continued)

Current / Last State Error Codes

The following table lists the error codes for Words 2-4 in the **Module Status** System Memory area.

Error Code (Decimal)	Description
E0	No error.
E121	Channel failure.
E122	Unused analog input channels exist.
E139	Broken transmitter on one of the analog input channels (if supported by analog module)
E142	Multiple channels failed.
E153	The module which was in this slot is no longer responding. User has removed a module in a Terminator I/O slave system. If Automatic Reset (default) is enabled for this slave, it will reset itself once the replacement module is inserted. If Manual Reset is enabled for this slave, the user must 1) SET the slave disable flag for that slave in the first diagnostic output word, 2) wait for bits 8-15 in second diagnostic input word to equal 1, then 3) RESET the slave disable flag in the first diagnostic output word.
E154	I/O configuration has changed. See E153 for reset methods.
E200-E216	Unused analog input channels exist at channel xx (1-16), where xx = Value -200. (example: E212 indicates unused analog channel exists at channel 12.

Extended Error Codes

The following table lists the error codes for Words 5-20 in the **Module Status** System Memory area.

Error Code (Decimal)	Description
E32-E63	Bitwise error where bit 5 is always SET. Look at bit 0 thru bit 4 to get a possible list of errors. Example 34 decimal =22 hexadecimal (Bit 5 SET and Bit 1 SET). BIT — Type of Error 0 — Terminal block off 1 — External P/S voltage low 2 — Fuse blown 3 — Bus error 4 — Module initialization error (intelligent module) 5 — Fault exists in module (this bit is SET if any of the above bits are SET)
E117	Write attempt to an invalid analog channel.
E119	Data not valid. Subnet mask or IP address not allowed // EBC SDK data packet not constructed properly.
E121	Analog input channel error.
E122	Unused analog input channels exist.
E139	Broken transmitter on one of the analog input channels.
E142	Channel failure.
E146	Communications failure. Hitachi drive on-board relay set.
E153	The module which was in this slot is no longer responding. User has removed a module in a Terminator I/O slave system. If Automatic Reset is enabled for this slave, it will reset itself once the replacement module is inserted. If Manual Reset is enabled for this slave, the user must 1) SET the slave disable flag for that slave in the first diagnostic output word, 2) wait for bits 12-15 in second diagnostic input word to equal 1, then 3) RESET the slave disable flag in the first diagnostic output word.
E154	One or more new modules has been inserted into the base. See E153 for reset methods.
E155	Terminator module status error. One or more of the modules in the T1H-EBC100 base has an error. For more detail check extended errors
E200-E216	Unused analog input channels exist at channel xx (1-16), where xx = Value -200.

Analog Input Module Configuration

The Terminator I/O analog input modules are configured using the following bit definitions located in the Configuration Data memory area of the T1H-EBC100's System Memory. Only the T1F-08AD1 and T1F-08AD2 support Fast Response mode.

Analog Input Module Configuration Bits		
Bit 0-4	**Input Enable** 0 = All Channels Enabled 1 - 31 = Number of Channels Enabled Starting With Channel 1	Write
Bit 5-6	Reserved	-
Bit 7	0 = Normal Response 1 = Fast Response **(T1F-08AD1 and T1F-08AD2 only)**	Write
Bit 8-15	Reserved	-

Analog Output Module Configuration

The Terminator I/O analog output modules are configured using the following bit definitions located in the Configuration Data memory area of the T1H-EBC100's System Memory.

Analog Output Module Configuration Bits		
Bit 0	**Outputs Enable** 0 = All outputs OFF 1 = All outputs Enabled	Write
Bit 1	**Unipolar** / **Bipolar** 0 = Unipolar selected 1 = Bipolar selected	Write
Bit 2	**5V** / **10V Range** 0 = 5V range 1 = 10V range	Write
Bit 3	**0 - 20mA** / **4-20mA Range** 0 = 0 - 20mA range 1 = 4 - 20mA range	Write
Bit 4-15	Reserved	-

5

T1H-EBC100 DHCP & HTML Configuration

In This Chapter. . . .

- T1H-EBC100 DHCP
- Disabling DHCP and Assigning a Static IP Address
- Using HTML Configuration

T1H-EBC100 DHCP

DHCP Issues

The T1H-EBC100 is configured at the factory to look for a DHCP (Dynamic Host Configuration Protocol) server at power up. DHCP provides a way to allocate IP address dynamically to devices on a local area network (LAN). A system or network administrator configures a DHCP server with a range of IP addresses that can be assigned to DHCP enabled clients (i.e. T1H-EBC100).

In addition to an IP address, a DHCP server can provide other information such as DNS domain or a gateway IP address.

DHCP uses the concept of a "lease" or amount of time that an assigned IP address will be valid for a client. The lease time can vary depending on how long a user is likely to require the network connection at a particular location. Since the TCP/IP configuration is "leased" to the client, that is, it's not a permanent configuration. This information can change from one power up session to the next. While this is an acceptable solution for the initial testing and setup of your T1H-EBC100 device, we **do not** recommend that you use DHCP to assign IP addresses for your runtime operation. Use NetEdit3 or the T1H-EBC100's HTML Configuration page to assign a static IP address to the module (shown below).

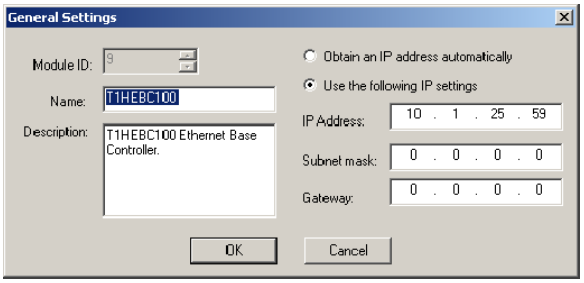
NetEdit3 can be used to connect to a T1H-EBC100 using the IPX protocol, regardless of the IP address that was assigned to it by a DHCP server.

Disabling DHCP and Assigning a Static IP Address

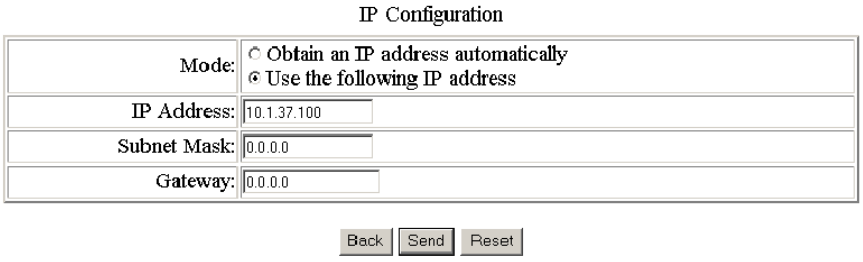
You can use NetEdit3 or the T1H-EBC100's HTML Configuration page to disable DHCP and assign a static IP address to the module. Click on the *Use the following IP Address* button and enter a valid IP address for your network application.

NetEdit3 (refer to chapter 3)

.



HTML Configuration

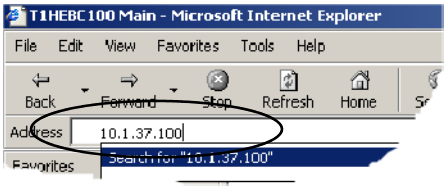


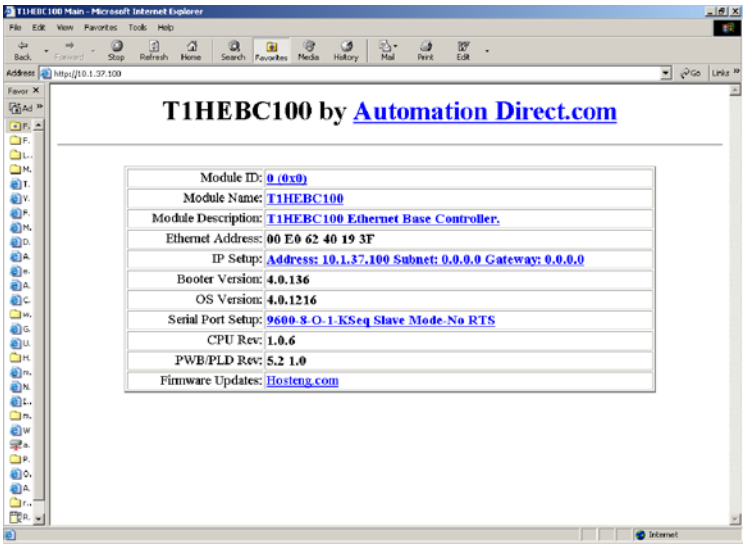
Using HTML Configuration

The T1H-EBC100 can be configured by using your PC's internet browser to access the module's HTML configuration page. This method of configuration uses the TCP/IP protocol, so you must know T1H-EBC100's IP address to establish communications. The IP address may have been assigned by a DHCP server (default) or may have been set by using NetEdit3.

Connecting to the T1H-EBC100

Enter the module's IP address in your browsers Address field. Connecting to the module's HTML Configuration utility brings up the window below

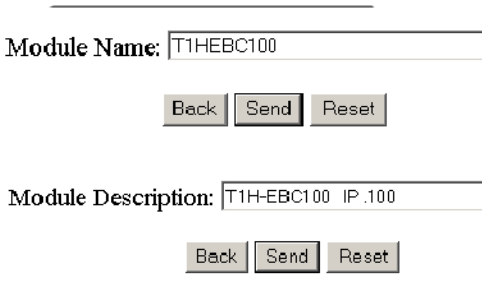




To configure the module, click on the desired parameter field. A new window will open, which are all described below and on the following page. Clicking the Back button will take you back to the main configuration screen shown above. Clicking the Send button writes the entry or change to the module's flash memory and clicking the Reset button reads the module's flash memory.

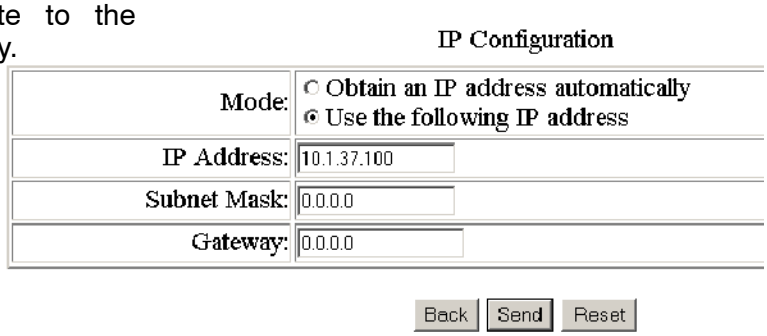
Module ID: Module IDs must be unique for each EBC, but they do not have to be in sequence. The module's Node Address rotary switches must both be set to allow the HTML configuration tool to set a Module ID. Do not use address zero for communications.

Module Name field and **Module Description** fields are optional to identify the module. Click the Send button to write to the module's flash memory.



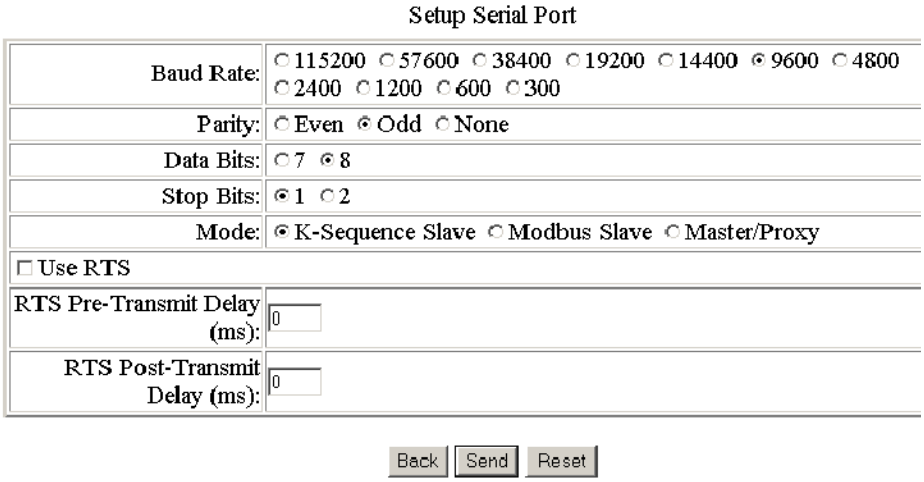
Ethernet Address: this is the MAC Address. It is a factory-assigned address that is on the permanent label on the module.

IP Configuration: Set IP Address, Subnet Mask and Gateway addresses. Click the Send button to write to the module's flash memory.



The module's current **Booter Version** and **OS Version** are listed. The latest versions can be found by clicking Hosteng.com in the Firmware Updates field.

Serial Port Setup: configure or make necessary changes to the serial port communication parameters. Click the Send button to write to the module's flash memory.



The module's current **CPU Rev** and **PWB/PLD Rev** are listed. The latest versions can be found by clicking Hosteng.com in the Firmware Updates field.

Firmware Updates: If your PC is connected to the internet, clicking on Hosteng.com will take you to Host Engineering's web site where the most current firmware files are available for downloading to your PC. You must use NetEdit3 to upgrade the module.

A

Using the T1H-EBC(100) with Think & Do

In This Appendix. . . .

- Mapping T1H-EBC(100) I/O Points
- Hot Swap Setup
- Analog I/O Module Configuration
- I/O Module Status Words / Bits
- Using EZTouch/EZText Panel with the RJ-12 Serial Port

NOTE: Think & Do Software has been retired and therefore is no longer supported.

Mapping T1H-EBC(100) I/O Points

We recommend that you are familiar with the "Getting Started" and "Creating a Project" chapters in the Think & Do Studio Learning Guide before attempting to map the EBC I/O points/channels to Data Items using ConnectivityCenter.

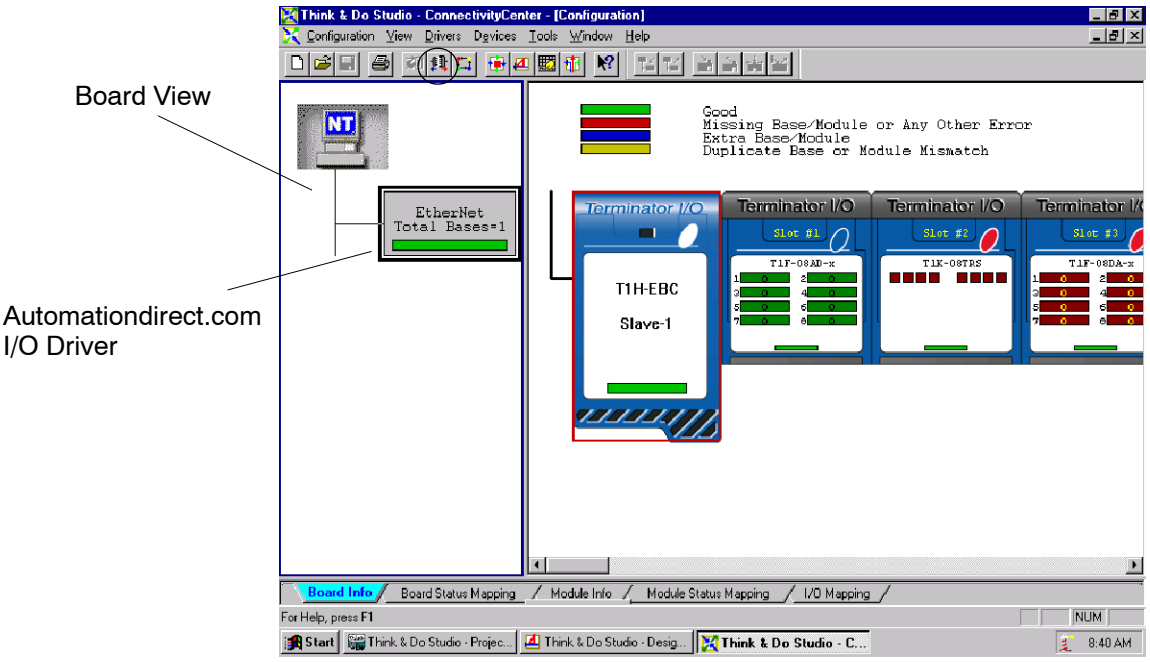
Launching Connectivity Center Tool

To launch ConnectivityCenter:

1) Launch Think & Do Studio ProjectCenter from the Windows desktop by clicking on Start > Programs > Think & Do Studio > ProjectCenter. Or, click on the ProjectCenter icon to start.
2) Click on the File Menu and either Open your Think & Do Project or select New.
3) Within ProjectCenter select Windows 2000 or NT Certified PC as the Runtime Target.
4) Then click Tools > ConnectivityCenter to launch ConnectivityCenter. Or, click on the ConnectivityCenter shortcut in the Project Explorer.
5) Once in ConnectivityCenter click on Drivers > Add and select Automationdirect.com Ethernet I/O Driver.
5)Then click on Configuration > Connect or click on the Connect toolbar button.

Connecting to the EBC

ConnectivityCenter will draw a picture of your EBC I/O system.



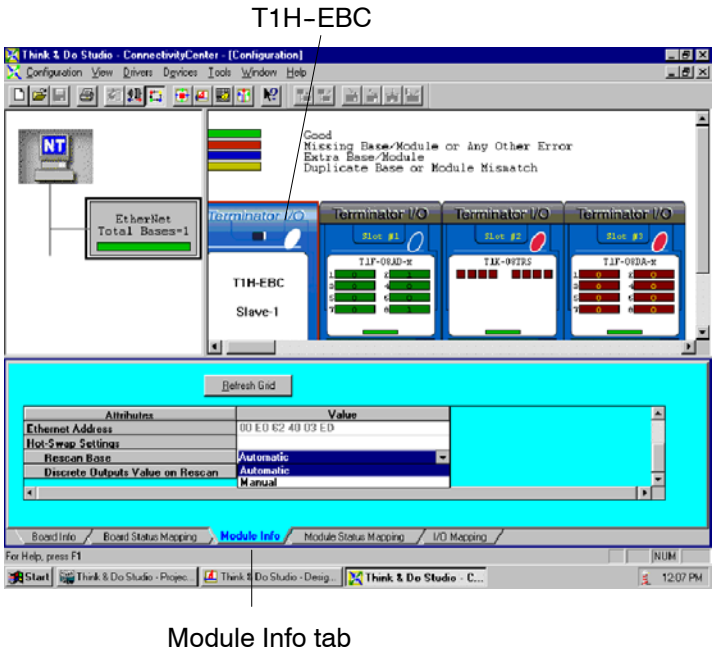
Mapping I/O Points to Data Items

This procedure is discussed in detail in the "Creating a Project" chapter in the Think & Do Studio Learning Guide. This will map your real world I/O to Data Items.

Hot Swap Setup

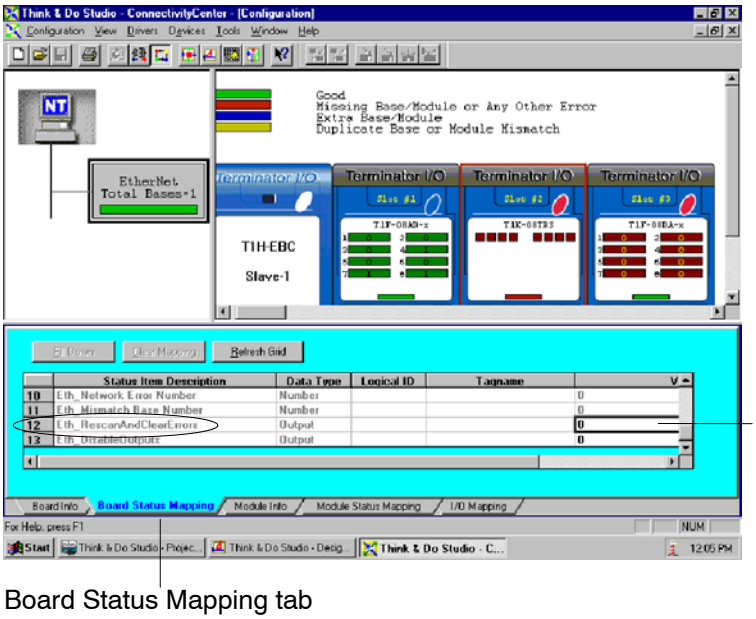
Hot Swap: Automatic Mode

Click on the T1H-EBC(100) graphic and the Module Info tab in ConnectivityCenter. The Hot Swap Settings can be set to either Automatic or Manual mode. If Automatic mode is selected, the base will automatically rescan once a module has been "Hot Swapped".



Hot Swap: Manual Mode Reset

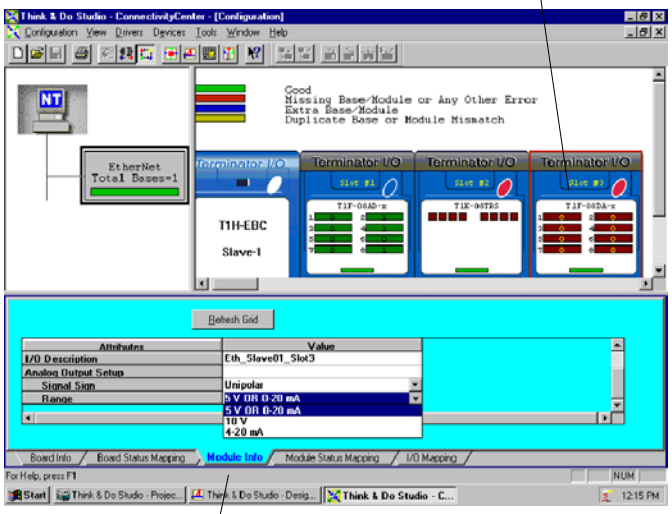
If Manual mode is selected, you will need to write a logical "1" to the Eth_Rescan And Clear Errors Status Item Value to force the base rescan once the module Hot Swap has been performed. This information is listed under the Board Status Mapping tab.



Analog I/O Module Configuration

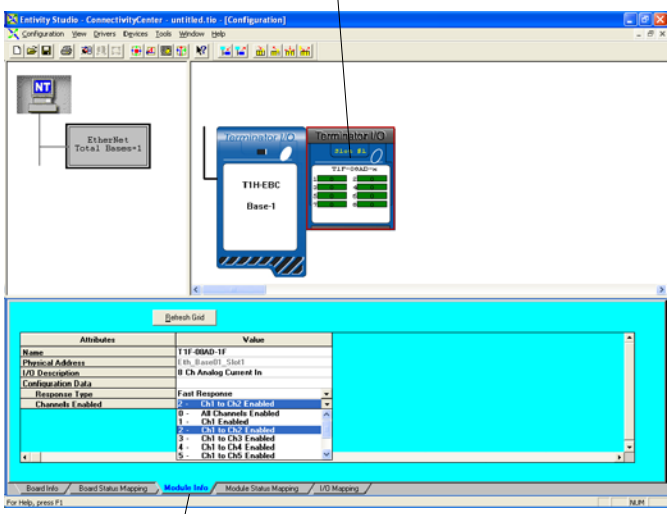
Click on the analog I/O module graphic to be configured and the Module Info tab in ConnectivityCenter. For the analog output modules, select the module Signal Sign and Range. For the analog input modules (T1F-08AD1 and T1F-08AD2 only), select either Fast Response or Normal Response (default) mode.

Analog Output Module



Module Info Tab

Analog Input Module

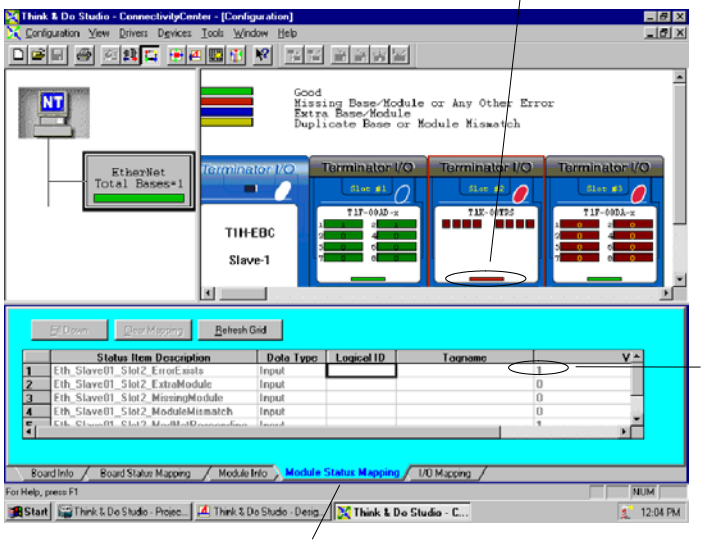


Module Info Tab

I/O Module Status Word / Bits

I/O Module diagnostic information is listed for each I/O module under the Module Status Mapping tab. Click on a module graphic to display its Status Item Descriptions.

Status Indicator



1 = Error

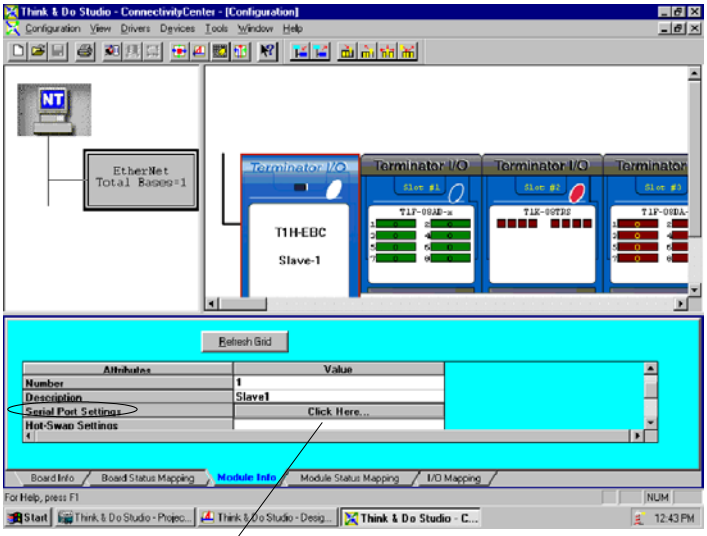
Module Status Mapping Tab

Using EZTouch/EZText Panel with the RJ-12 Serial Port

The T1H-EBC has a built-in RS232C serial port that can be used to connect to an operator interface panel. Use ConnectivityCenter to configure the connection from the T1H-EBC to the EZTouch or EXText panel. The “HMI Options for Remote Base Controllers” section in the “Operator Screen Techniques” chapter in the Think & Do Studio Learning Guide discusses configuring and using Optimate Panels with the EBC.

Adding Operator Interface Device

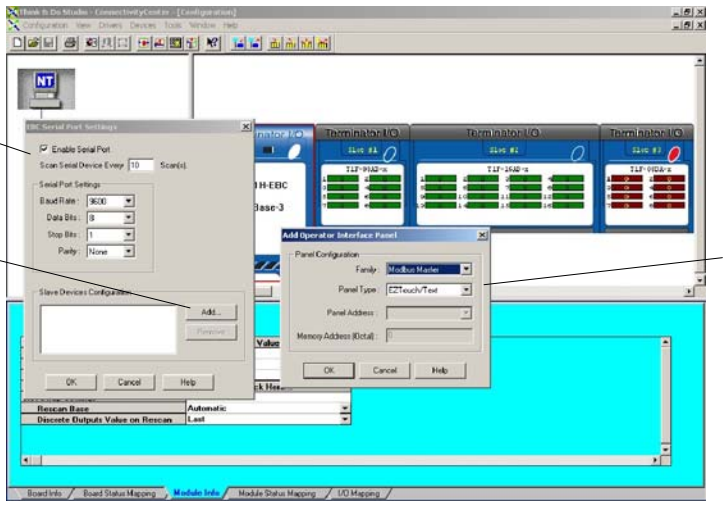
Click on the T1H-EBC graphic and Module Info tab in ConnectivityCenter. The Serial Port Settings attribute is only visible in ConnectivityCenter when the I/O is disconnected. Following the steps below will configure the EBC’s RJ12 serial port to be used with the EZTouch or EZText panels.



1. Click to access port settings

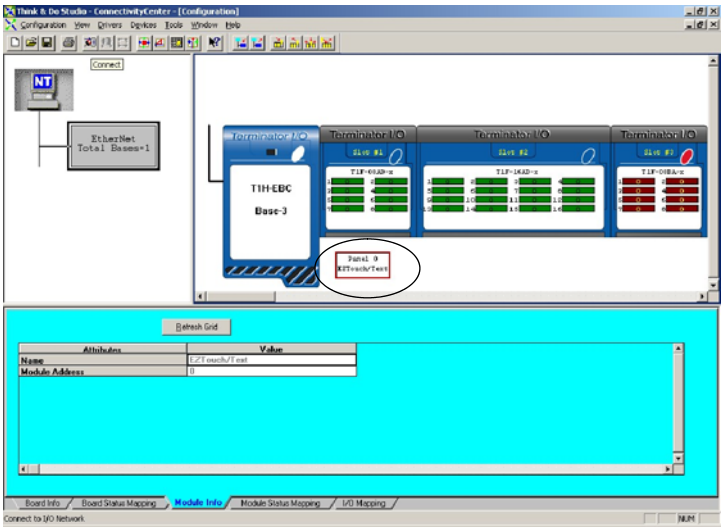
2. Select to Enable the serial port. These settings must match the port configuration of the EZ panel.

3. Click to Add Panel



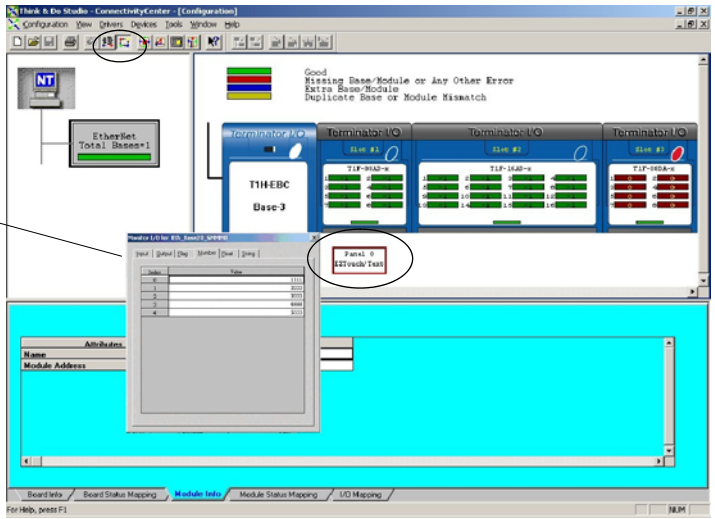
4. Set the Family to Modbus Master and the Panel Type to EZTouch/Text.

Once the EZTouch or EZText panel has been added, it will show up in the list of the configured devices, and an EZTouch/Text panel graphic symbol will be located under the I/O base next to the EBC.



Using Monitor I/O to Verify Panel Operation

Re-connect to the I/O in ConnectivityCenter by clicking on the Configuration menu > Connect or by clicking on the Connect toolbar button. Then Scan the I/O by clicking on the Configuration menu > Scan or by clicking on the Scan toolbar button. Doubleclick on the EZ panel box graphic to launch the Monitor I/O Dialog Box. The Monitor I/O tool allows the user to update the fields at any moment, but the panel continuously updates the fields with any changes as well. All of the “Value” fields in the Monitor I/O Dialog Box are read/write and updates from the the Monitor I/O Dialog box take precedence over updates from the panel.
The user can update bit values (Input, Output and Flag) immediately by one mouse click or by pressing the “space bar”.
When typing in numbers, the grid enters the “edit mode”, which blocks off any conflicting updates from the panel. The “edit mode” entry completes after pressing “Enter”, any arrow key or selecting a new line.



B

Using the T1H-EBC(100) with *KEPDirect* OPC Server

In This Appendix. . . .

Introduction to *KEPDirect*

Introduction to OPC

OPC, OLE (**O**bject **L**inking and **E**mbedding) for Process Control, is an industry standard created by a number of worldwide leading hardware and software suppliers in cooperation with Microsoft. The OPC Data Access specification, as maintained by the OPC Foundation, is a non-proprietary technical specification that defines a set of standard interfaces based upon Microsoft's OLE/COM technology. An OPC server (driver) allows items such as distributed control systems, programmable logic controllers, I/O systems and smart field devices to communicate with a wide range of HMI/SCADA (client) software packages residing on a PC. Traditionally, each software or application developer was required to write a custom interface, or server/driver, to exchange information with hardware field devices. OPC eliminates this requirement allowing manufacturing customers true plug and play connectivity and the freedom to choose products based on their automation requirements.

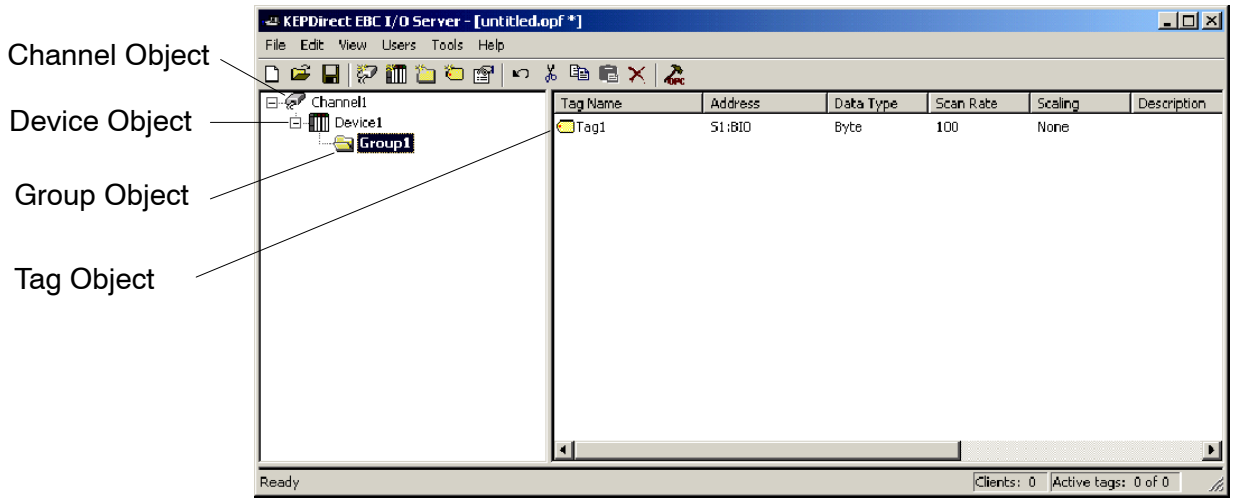
DDE Support

While *KEPDirect* is first and foremost an OPC server, KEPware recognized that a number of legacy applications still depend upon DDE for their underlying client server technology. Early in the development of Windows, Microsoft provided a generic client server technology called DDE (Dynamic Data Exchange). DDE did provide a basic architecture that would allow many windows applications from a wide range of vendors to share data. But there was one problem, DDE was not designed for the industrial market lacking much of the speed and robustness desired in an industrial setting. However, this did not stop DDE from becoming a dominant client/server architecture, largely due to its availability in most windows applications.

KEPDirect

*KEPDirec*t Enhanced OPC/DDE Server is a 32 bit windows application that provides a means of bringing data and information from a wide range of industrial devices and systems into client applications on your Windows PC. *KEPDirect* falls under the category of a "Server" application. It is very common to hear the term "client/server application" in use across many software disciplines and business segments. In the industrial market, it has usually come to mean the sharing of manufacturing or production data between a variety of applications ranging from human machine interface software and data historians, to large MES and ERP applications.

At a high level, the *KEPDirect* OPC Server is comprised of several objects that are described on the next page.



Channel Object: Each protocol or driver used in a *KEPDirect* project is referred to as a channel. A channel refers to a specific communications driver. A *KEPDirect* project can consist of many channels each with unique communications drivers or each with the same communications driver.
Each channel name must be unique in a *KEPDirect* application. The channel name entered here will be part of the OPC browser information.

Device Object: Unlike the channel name, "Device names" can be the same from one channel to the next. The device name is a user defined logical name for the device. The device name and channel name will be part of the OPC browser information as well as a DDE item name. Within an OPC client the combination of channel name and device name would appear "ChannelName.DeviceName".

Group Object: *KEPDirect* allows tag groups to be added to your project. Tag groups allow you to tailor the layout of OPC data in logical groupings that fit the needs of your application. Using tag groups allows multiple sets of identical tags to be added under the same device. This can be very convenient when a single device handles a number of similar machine segments. From an OPC client standpoint, the use of tag grouping allows you to segregate your OPC data into smaller tag lists, which can make finding a specific tag easier when browsing the server.

Tag Object: *KEPDirect* allows both dynamic tags, (tag entered directly at the OPC client that specify device data) and user defined tags. User defined tags have the benefit of allowing the tag to be browsed from an OPC client that supports tag browsing. User defined tags also support tag scaling. Unlike many of the dialogs you will find in *KEPDirect*, the tag properties dialog has a number of features that are driven by icons. The tag name is part of the OPC browse data. Tag names must be unique within a given device branch or tag group branch. If your application is best suited by using blocks of tags with the same names, use tag groups to segregate the tags.

KEPDirect Project: Adding and Configuring a Channel

Running the Server

KEPDirect, like any OPC server, can be started a number of ways. One of the benefits of OPC technology is that your OPC client can automatically invoke the server when it attempts to connect and collect data from it. In order for this automatic mode of operation to occur you must first create and configure a project. Once you have created a project, *KEPDirect* will automatically select the most recently used project when it is invoked by an OPC client.

Initially however, you need to manually invoke *KEPDirect* using either the desktop icon, if you chose to install it, or by selecting *KEPDirect* from the windows start menu. Depending on any changes you may have made to the appearance of *KEPDirect*, once invoked you should be presented with the following interface. To learn more about the various elements of the user interface see (Basic *KEPDirect* Components).

While discussing how to start *KEPDirect* its important to understand what the system requirements are for running the server. *KEPDirect* has been designed to place as little strain on your system as possible.

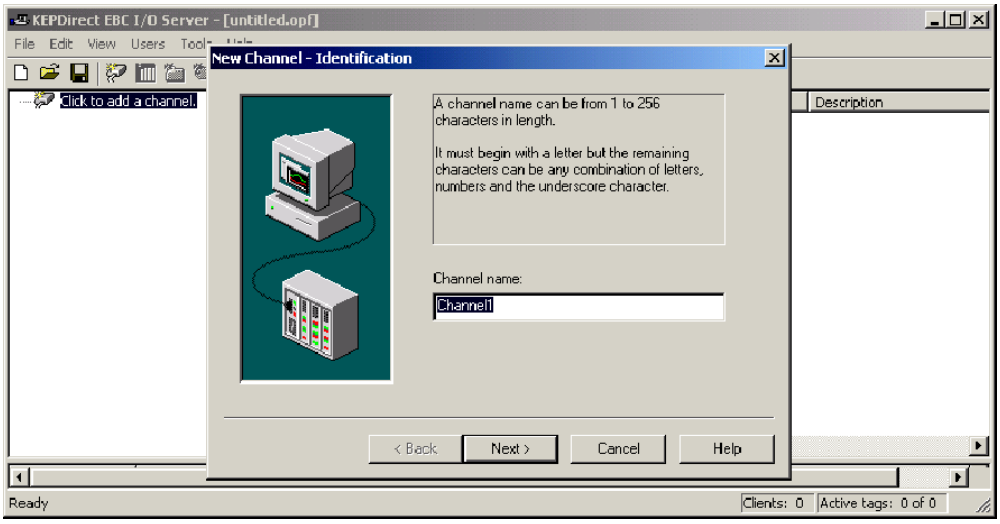
Recommended System Requirements:

400Mhz Pentium
64 Megs of Ram
10 Megs of Hard Disk Space
Windows NT(SP6a)/2000 (Strongly recommended for industrial settings)
Available Ethernet Card

Adding a Channel

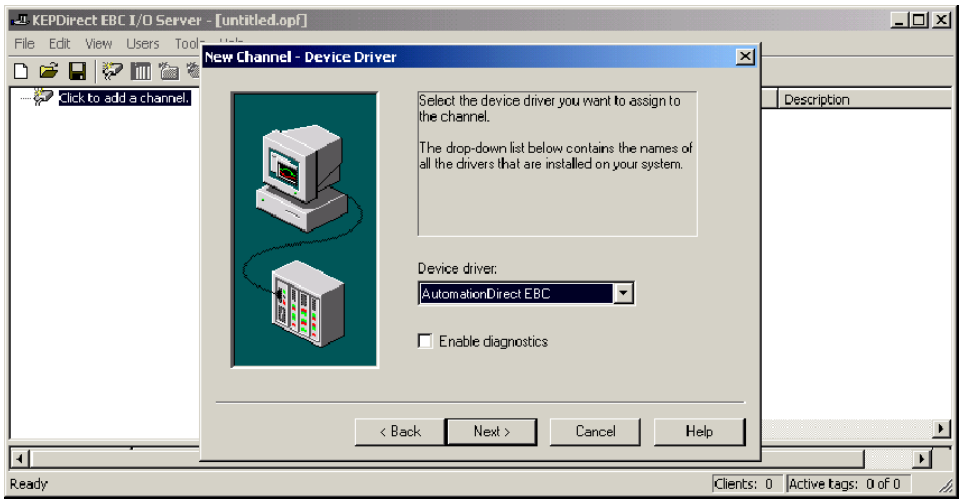
A channel refers to a specific communications driver. A *KEPDirect* project can consist of many channels each with unique communications drivers or each with the same communications driver. Depending on the driver or drivers you have installed you can define a number of channels within a single project. A channel acts as the basic building block of an OPC link. Properties like communications port, baud rate, and parity are contained at the channel level. Each channel name must be unique in a *KEPDirect* project. The channel name can be up to 31 characters long.

To add a new channel to your project you can use the Edit menu > New Channel, the Toolbar Add Channel, or the “Click to add a channel” dialog.



Selecting the Device Driver

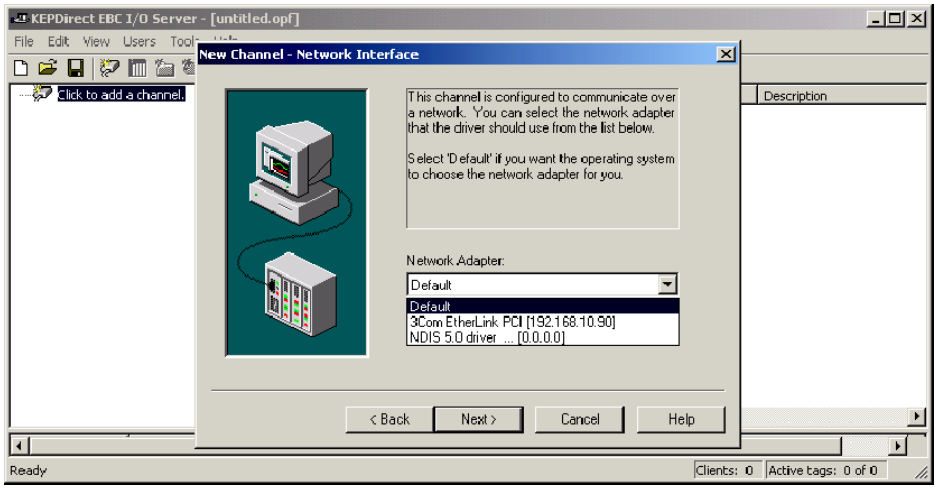
Select the device driver you want to assign to the channel. A driver list will be presented displaying all of the device drivers that are installed in your system.



Selecting the ”Enable diagnostics” check box will enable diagnostic information to be available to your OPC application for this channel. With diagnostic functions enabled, diagnostic tags are available for use within client applications. In addition to diagnostic tags, a diagnostic window is also available when this feature is enabled. The diagnostic features of *KEPDirect* do require a minimal amount of overhead processing. For this reason it is recommended that you only use the diagnostic features when needed and disable them when not in use which is the default case.

Selecting the Network Adapter

The Network Interface selection allows you to select a specific NIC card for the Automationdirect EBC Ethernet driver to use based on the NIC name or its assigned IP address. By selecting a specific NIC interface you will be able to force the driver to send all Ethernet communication through the specified NIC. If you do not know which NIC you should use, select the ”Default” condition.



Setting the Server Writes Optimizations

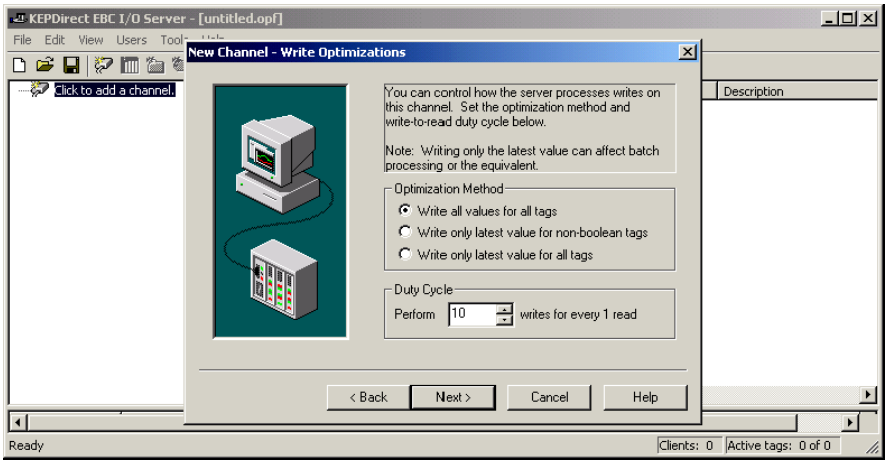
As with any OPC server, writing data to your device may be the most important aspect of your application. Insuring that the data written from your OPC client application gets to the device in a timely manners is the goal of the server. *KEPDirect* provides a number of optimization settings that can be used to tailor the server to meet the needs, and improve the responsiveness of your application.

There are currently three write optimization modes. The following is a brief description of the modes. For a detailed explanation, refer to the “Channel Properties - Write Optimizations” section in the *KEPDirect* on-line help file.

NOTE: We strongly suggest that you characterize your application for compatibility with these write optimization enhancements before using them in a production environment.

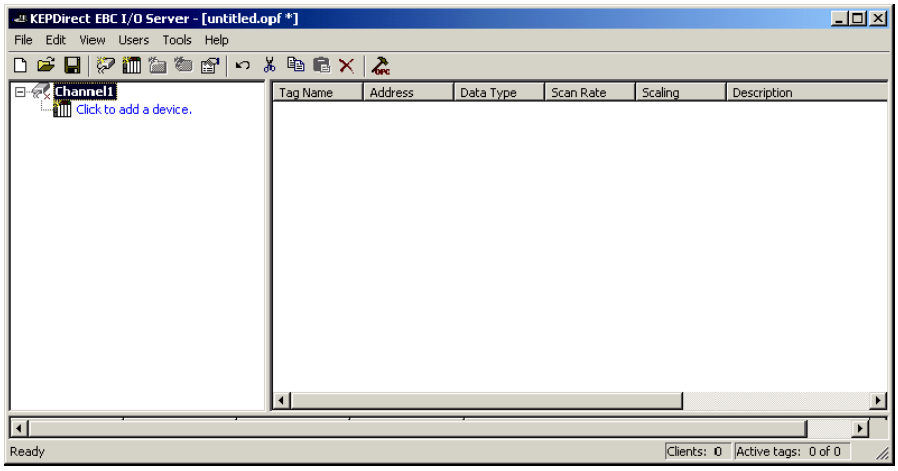
The default mode, ”**Write all values for all tags**” will force the server to attempt to write every value to the controller. This mode insures that everything written from your OPC client applications will be sent to the target device. While writing every value to the device may seem like the best course of action, there are a number of applications where writing every value, many of which may be the same value, over and over may be simply a waste of communications bandwidth.
The ”**Write only latest value for non-boolean tags**” allows any value that is not a boolean value to be updated in the server’s internal write queue and will then be sent to the device at the next possible opportunity. This can dramatically improve the overall performance of your application. This feature must be used with a clear understanding of how it will affect the operation of your application.
The final write optimization mode, ”**Write only the latest value for all tags**”, takes the operation described for the second mode and applies it to all tags.

The **Duty Cycle** selection allows you to control the ratio of write operations to read operations. By default the duty cycle is set to ten. This means that ten writes will occur for each read operation. If your application is doing a large number of continuous writes but you need to insure that read data is still given time to process, you may want to reduce the Duty Cycle. A setting of one will result in one read operation for every write operation. In all cases if there are no write operations to perform, reads will be processed continuously.



Saving the New Channel Settings

With “Channel1” channel added to the server, the *KEPDirect* window will appear as follows:



Note that the channel is shown using the channel name given, but it is also has a small red ”x” below the channel icon. The red ”x” indicates that the channel does not contain a valid configuration. “Channel1” is not valid because a device has not yet been added to the channel.

Using Multiple Channels in a Project

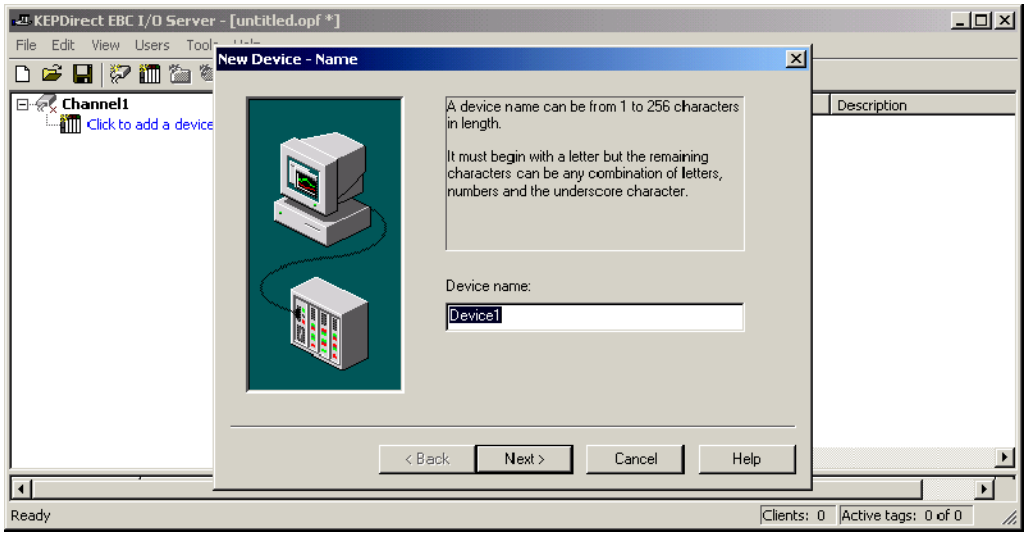
KEPDirect supports the use of multiple channels. As you add channels to your project you can specify either the same communications driver or different communications drivers. Most communication drivers offered by KEPware support operation on up to 16 communications ports or ethernet network connections simultaneously. By defining multiple channels you can improve the overall performance of you application. In the case of either a serial driver or Ethernet driver using multiple channels allows you to spread large communications loads across the multiple channels. A good example of this would be a serial driver that is being used to communicate with eight devices on the serial line. Normally the communications driver used in this application would be responsible for gathering data from all eight devices in a round robin fashion. If this same application is reconfigured to use multiple channels assigned to multiple communications ports, the device load can be divided across the channels. The end result is reduce work load on each channel and dramatic improvements in the responsiveness of your application. The need to use multiple channels is dependent solely on the needs of your application. In either case there is no additional cost involved to use a licensed driver on multiple communications or Ethernet ports.

KEPDirect Project: Adding and Configuring a Device

Adding a Device

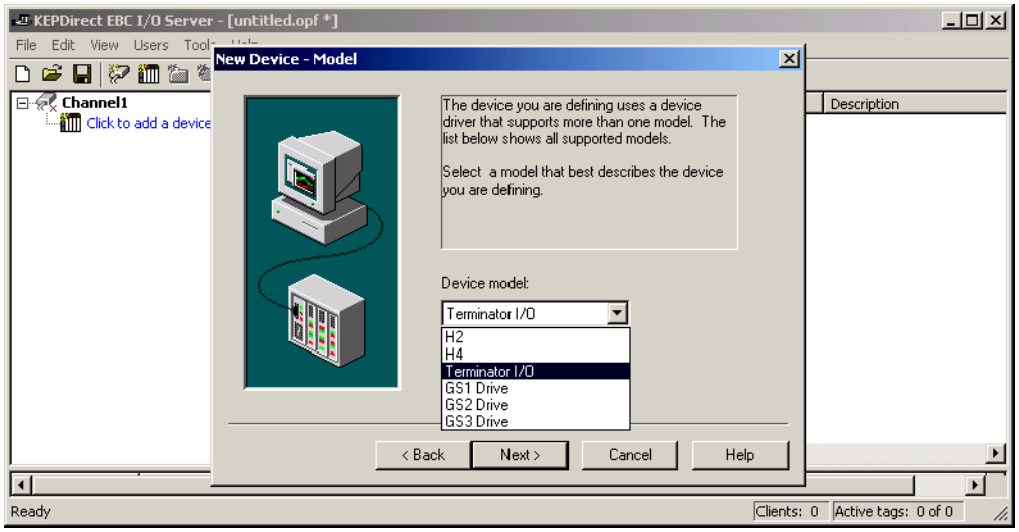
Once a channel has been configured in a *KEPDirect* project, a device must be added to the channel. Devices represent PLCs, I/O devices or other hardware that the server will communicate with. Device selection is restricted by the device driver the channel is using.

To add a device to a channel, select the desired channel and use the Edit menu > New Device, the Toolbar Add Device, or the “Click to add a device” dialog.



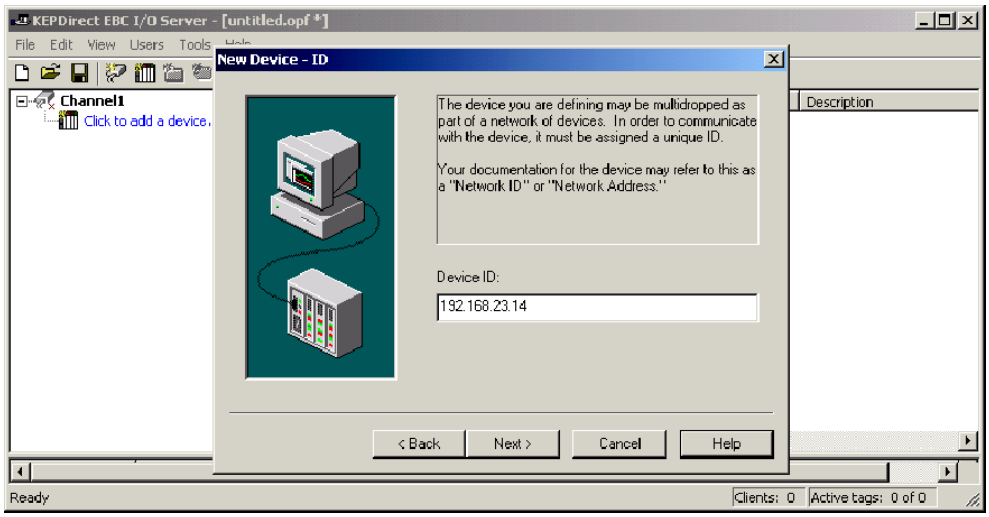
Selecting the Device Model

The ”Model” parameter allows you to select the specific type of the device associated with a device ID. The contents of the model selection drop down will vary depending on the chosen communication driver.



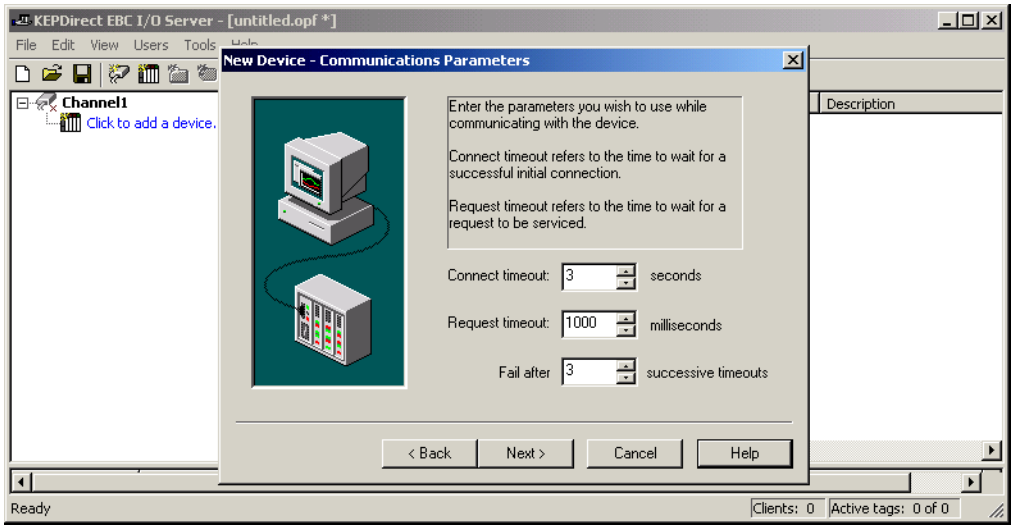
Selecting the Device Model

The "Device ID" parameter allows you to specify the driver specific station or node address for a given device. Since the Automationdirect EBC driver is an Ethernet based driver, a unique and valid TCP/IP address must be entered. IPX protocol is not supported.



Setting the Device Timeout Properties

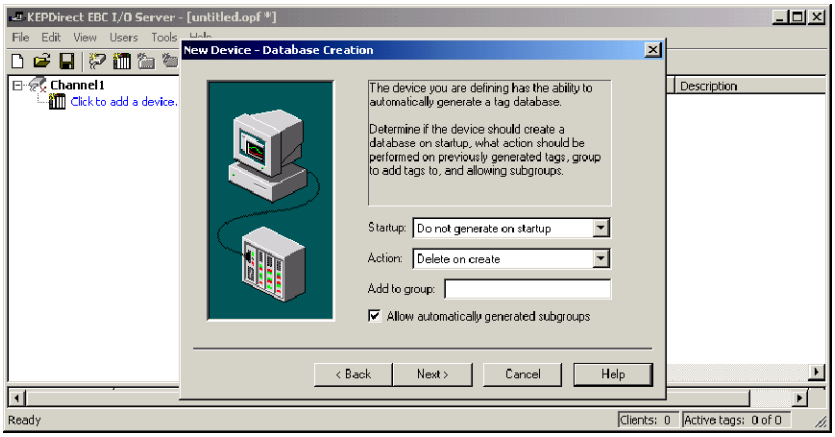
Device timeout parameters allow a driver's response to error conditions to be tailored to the needs of your application. The timeout parameters are specific to each device you configure. Each of the field parameters is defined in detail in the "Device Properties - Timeout" section in the *KEPDirect* on-line help file.



The "**Connection timeout**" allows the time required to establish a socket connection to a remote device to be adjusted. The "**Request timeout**" is used by all drivers to determine how long the driver will wait for a response from the target device. The "**Fail after**" parameter is used to determine how many times the driver will retry a communications request before considering the request to have failed. If your environment is prone to noise induced communications failures you may want to increase the number of **retries** the driver performs.

Automatic OPC Tag Database Generation

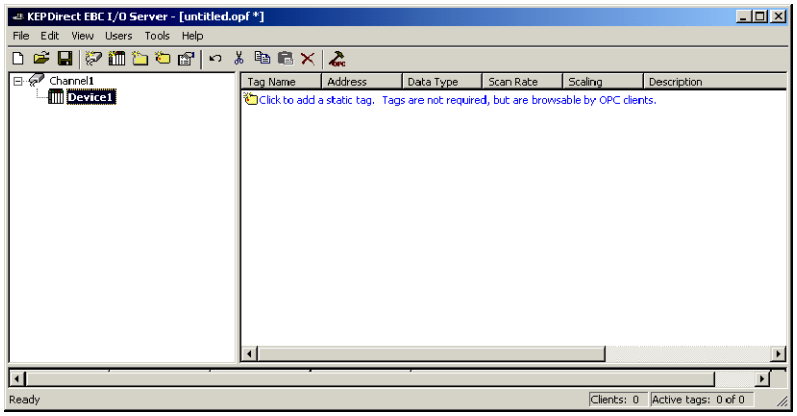
The automatic OPC tag database generation features of *KEPDirect* have been designed to make the setup of your OPC application a Plug and Play operation. Since the Automationdirect EBC communication driver supports this feature, you can configure it to automatically build a list of OPC tags within *KEPDirect* that correspond to device specific data. The automatically generated OPC tags are then browsable from your OPC client. The OPC tags that are generated are dependent upon the nature of the supporting driver. Each field selection is defined in detail in the "Automated OPC Tag Base Generation" section in the *KEPDirect* on-line help file.



The "Automatic tag database generation on device startup" selection allows you to configure when OPC tags will be automatically generated. There are three possible selections. The default condition, "Do not generate on startup", will prevent the driver from adding any OPC tags to tag space of *KEPDirect*. The selection "Always generate on startup", will cause the driver to always evaluate the device for tag information and to add OPC tags to the tag space of the server each time the server is launched. The final selection "Generate on first startup" will cause the driver to evaluate the target device for tag information the first time this *KEPDirect* project is run and to add any OPC tags to the server tag space as needed. When the automatic generation of OPC tags is selected, any tags that are added to the server's tag space must be saved with the project. You can configure your *KEPDirect* project to auto save from the Tools > Options menu.

Saving the New Device Settings

With "Device1" added to "Channel1", the *KEPDirect* window will appear as follows:



KEPDirect Project: Adding Tags to the Project

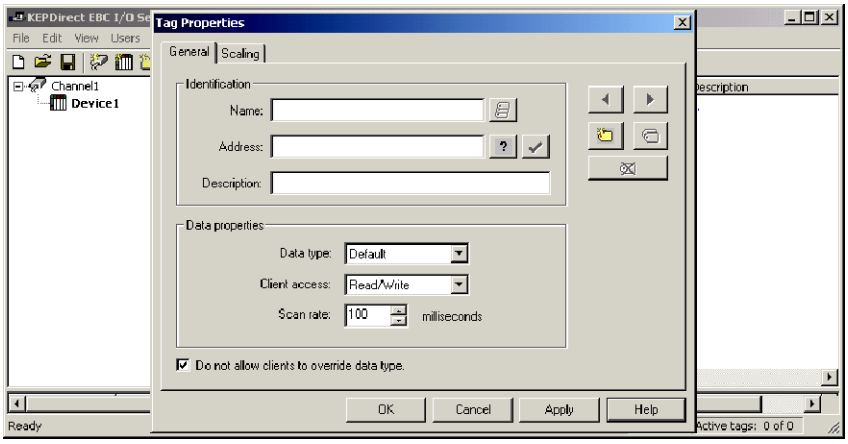
There are two ways to get data from a device to your client application using *KEPDirect*. The first method and most common method of defining tags is called **User Defined Tags**. This requires that you define a set of tags in the server project and then use the name you assigned to each tag as the item of each OPC/DDE link between the client and the server. The primary benefit to this method is that all user defined tags are available for browsing within OPC clients. Additionally, user defined tags also support scaling.

The second method of defining tags is called **Dynamic Tags**. Dynamic tags allow you to define tags in the client application. Instead of providing the server with a tag name as the OPC/DDE item, you would provide the device address (and optionally a data type). The server will create a tag for that location and start scanning for data automatically. *KEPDirect* allows tag groups to be added to your project.

Tag groups allow you to tailor the layout of OPC data in logical groupings that fit the needs of your application. Using tag groups allows multiple sets of identical tags to be added under the same device. This can be very convenient when a single device handles a number of similar machine segments. From an OPC client standpoint, the use of tag grouping allows you to segregate your OPC data into smaller tag lists, which can make finding a specific tag easier when browsing the server.

User Defined Tags

Each field selection is defined in detail in the “Tag Properties” section in the *KEPDirect* on-line help file. A brief description of each is listed below.



The tag **”Name”** parameter allows you to enter the string that will represent the data available from this tag. The tag name can be up to 31 characters in length. While using long descriptive names is generally a good idea, keep in mind that some OPC client applications may have a limited display window when browsing the tag space of an OPC server. The tag name is part of the OPC browse data. Tag names must be unique within a given device branch or tag group branch. If your application is best suited by using blocks of tags with the same names, use tag groups to segregate the tags.

The **"Address**" parameter allows you to enter the desired driver address for this tag. To determine how an address should be entered, you can use the **Hints button** next to the address parameter. Hints provide a quick reference guide to the address format of the driver. Once you have entered an address you can test it using the check address button. When pressed, the check address button attempts to validate the address with the driver. If the driver accepts the address as entered no message will be displayed. If an error is detected a pop-up will inform you of the error. Keep in mind that some errors will be related to the data type selection and not the address string.

The **"Description"** parameter allows you to attach a comment to this tag. A string of up to 64 characters can be entered for the description. If you are using an OPC client that supports Data Access 2.0 Tag Properties, the description parameter will be accessible from the Item Description property of the tag.

The **"Data Type"** selection allows you to specify the format of the tag's data as it is found in the physical device. The data type setting is an important part of how a communication driver reads and writes data to a device. For many drivers the data type of a particular piece of data is rigidly fixed.
The available data type selections are:

- **Default** - This type allows the driver to choose its default data type see the specific driver help for details
- **Boolean** - Single bit data On or Off
- **Char** - Signed 8 bit data
- **Byte** - Unsigned 8 bit data
- **Short** - Signed 16 bit data
- **Word** - Unsigned 16 bit data
- **Long** - Signed 32 bit data
- **Dword** - Unsigned 32 bit data
- **Float** - 32 bit Real value IEEE format
- **String** - Null terminated ASCII string
- **Double** - 64 bit Real value IEEE format
- **BCD** - Two byte packed BCD value range is 0 - 9999
- **LBCD** - Four byte packed BCD value range is 0 - 99999999

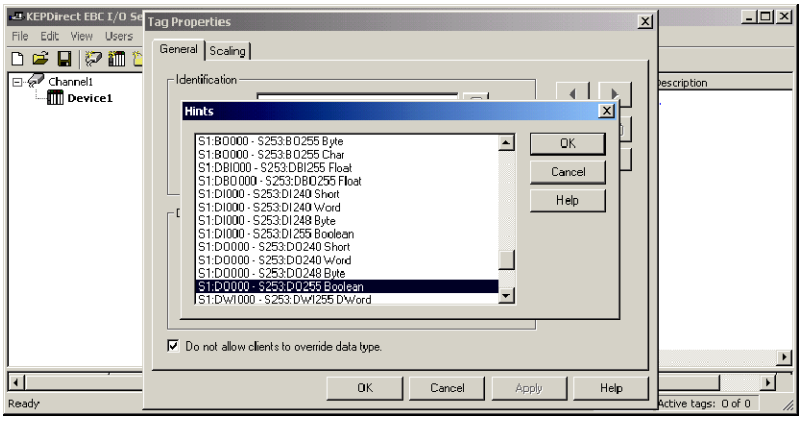
The **"Client access"** selection allows you to specify whether this tag is Read Only or Read/Write. By selecting Read Only you can prevent client applications from changing the data contained in this tag. By selecting Read/Write you are allowing client applications to change this tag's value as needed.

The **"DDE scan rate"** parameter allows to you specify the the update interval for this tag when used in a DDE client. OPC clients can control the rate at which data is scanned by using the update rate that is part of all OPC groups.

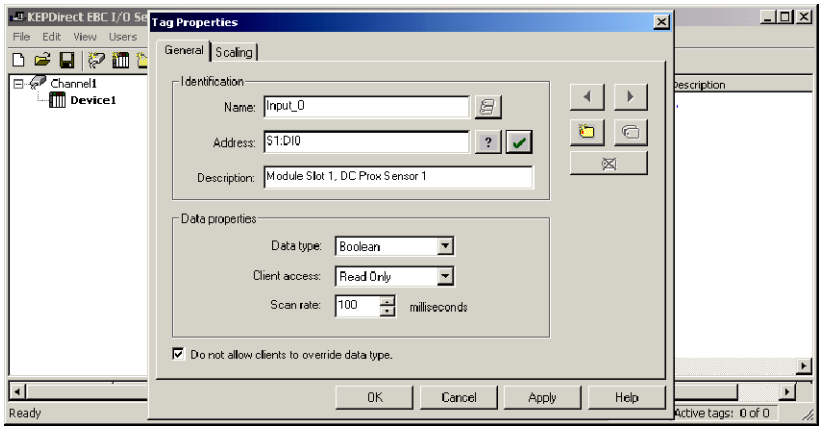
The **"Allow client to override data type"** selection allows you force OPC clients to use the data type you have specified for this tag. OPC clients can specify how they desire to view the data from a particular tag.

Creating a User Define Tag

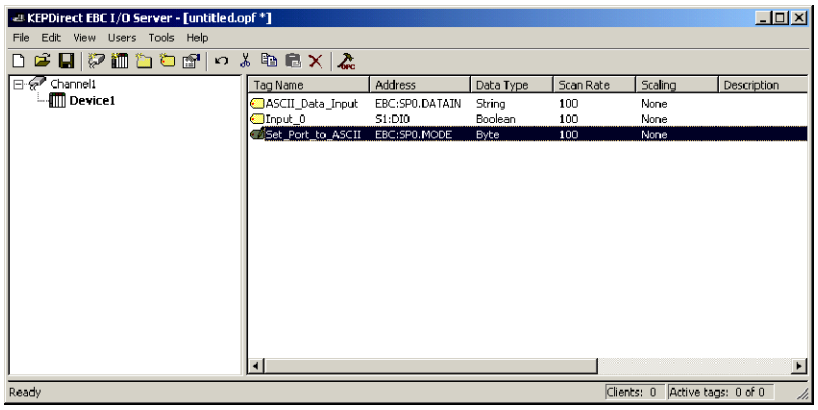
To determine how an address should be entered, use the Hints button "?" to the right of the address field. Hints provide a quick reference guide to the address format of the driver.



Once you have entered an address you can test it using the check address "✓" button. When pressed, the check address button attempts to validate the address with the driver. If the driver accepts the address as entered no message will be displayed. If an error is detected a pop-up will inform you of the error. Keep in mind that some errors will be related to the data type selection and not the address string. Below is an example of a valid tag properties.



The window below shows a valid configured channel, device and several user defined tags.



T1H-EBC(100) I/O Addressing

I/O slots must be individually addressed in the following form: S<ss>:<t><nn> where ss is the slot number (1 to 93), t is the address type (DI, DO, WI, WO, etc.), and nn is the address. The address ranges from 0 to an upper limit determined by the module occupying the slot.

I/O Type	Syntax	Data Type
Discrete Inputs	DI<nn> nn = Bit Number (decimal)	**Boolean**, Byte, Char, Word, Short, DWord, Long
Discrete Outputs	DO<nn> nn = Bit Number (decimal)	**Boolean**, Byte, Char, Word, Short, DWord, Long
Byte Inputs	BI<nn> nn = Bit Number (decimal)	**Byte**, Char
Byte Outputs	BO<nn> nn = Bit Number (decimal)	**Byte**, Char
Word Inputs	WI<nn> nn = Bit Number (decimal)	**Word**, Short
Word Outputs	WO<nn> nn = Bit Number (decimal)	**Word**, Short
DWord Inputs	DWI<nn> nn = Bit Number (decimal)	**DWord**, Long
DWord Outputs	DWO<nn> nn = Bit Number (decimal)	**DWord**, Long
Float Inputs	FI<nn> nn = Bit Number (decimal)	**Float**
Float Outputs	FO<nn> nn = Bit Number (decimal)	**Float**
Double Inputs	DBI<nn> nn = Bit Number (decimal)	**Float**
Double Outputs	DBO<nn> nn = Bit Number (decimal)	**Float**

T1H-EBC(100) I/O Addressing Example

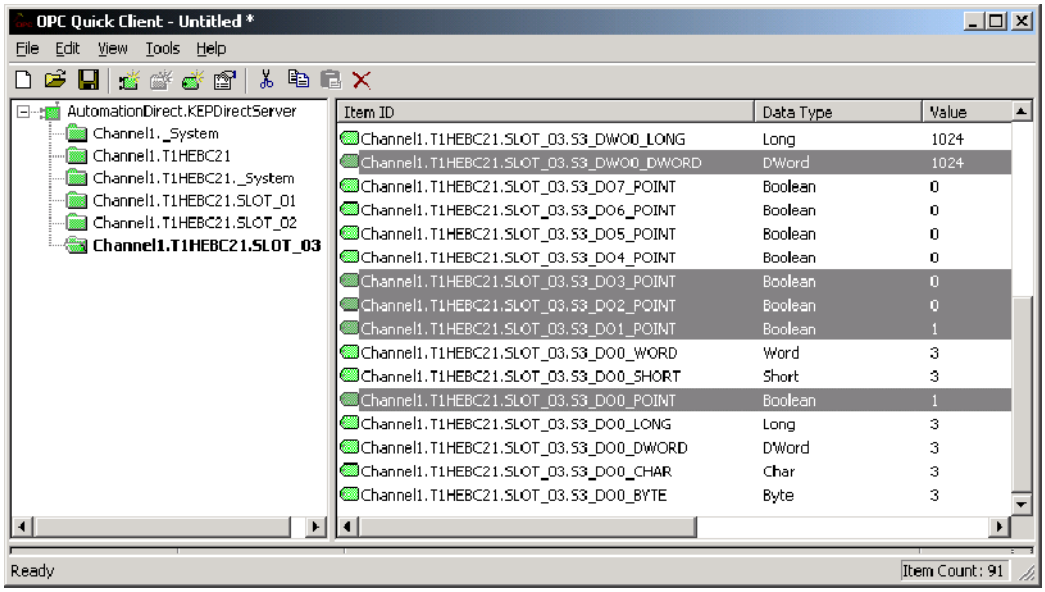
Each field selection is defined in detail in the "Tag Properties" section in the *KEPDirect* on-line help file.

Terminator I/O EBC Module Slot 0 Serial I/O Port EBC:SP0.item	Slot 1 8 Digital Input Addresses S1:DI0 to S1:DI7	Slot 2 16 Digital Input Addresses S2:DI0 to S2:DI15	Slot 3 8 Digit Output Addresses S3:DO0 to S3:DO7	Slot 4 16 Digital Output Addresses S4:DO0 to S4:DO15	Slot 5 8 Analog Input Addresses S5:DWI0 to S5:DWI7	Slot 6 16 Analog Output Addresses S6:DWO0 to S6:DWO15

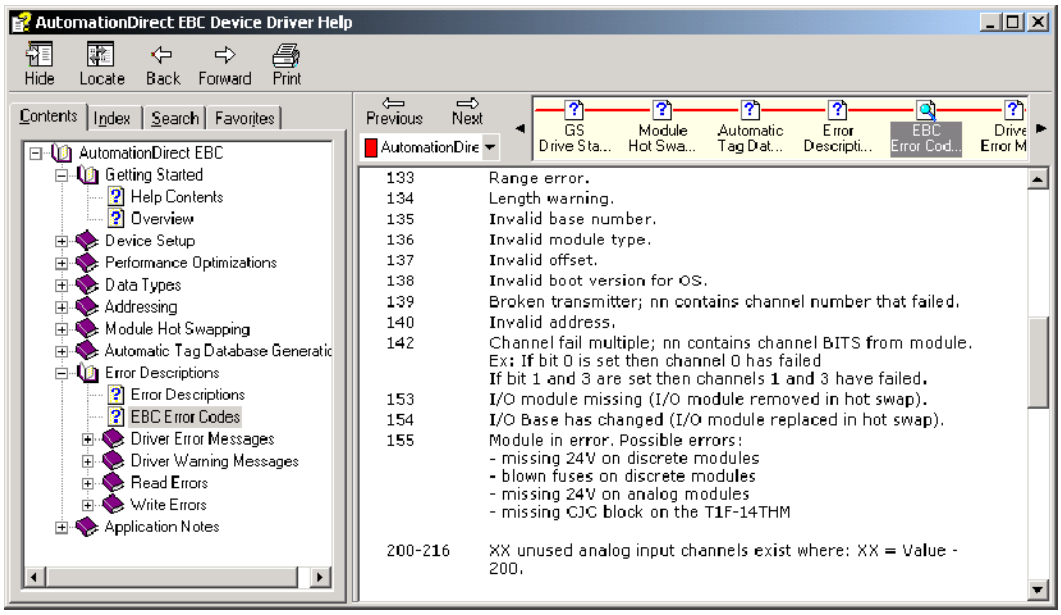
Analog Output Module Configuration

Analog Output Module Configuration Byte			
Module Control Byte	**KEPDirect Byte**	**Description**	
Bit 24	DO0_POINT	**Outputs Enable** 0 = All outputs OFF 1 = All outputs Enabled	Write
Bit 25	DO1_POINT	**Unipolar / Bipolar** 0 = Unipolar selected 1 = Bipolar selected	Write
Bit 26	DO2_POINT	**5V / 10V Range** 0 = 5V range 1 = 10V range	Write
Bit 27	DO3_POINT	**0 - 20mA / 4-20mA Range** 0 = 0 - 20mA range 1 = 4 - 20mA range	Write
Bit 28-31	DO4_POINT - DO7_POINT	Reserved	-

The following example shows the KEPDirect OPC Quick Client (discussed in Appendix C) used to setup a Terminator I/O analog output voltage module in slot 3. The highlighted selections are configured for Output Enabled (DO0_Point=1), BiPolar (DO1_Point=1), and 5V (DO2_Point=0). The analog output data value is 1024 decimal and results in a voltage output of -2.5V.



Diagnostic bits for Terminator I/O family of analog I/O are supported differently on each module but will present themselves as error bits/values or messages to the KEPDirect EBC I/O server using a common convention. A complete definition of the error information, and it's format convention, is available in the AutomationDirect EBC Help file. This can be accessed either from the Start Menu > Program > KEPDirect EBC I/O Server > Help
Documentation or through the Help menu from within the server. The example below shows the list of error codes supported by the EBC I/O server. The most common errors for analog I/O are 139, 142, 155, and 200-216 depending on the features supported in the specific analog module.



C

Using the *KEPDirect* OPC Quick Client

In This Appendix. . . .

— Creating a *KEPDirect* Quick Client Project

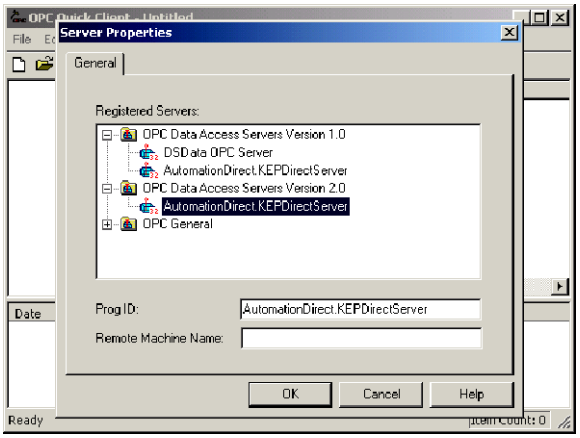
— Using the RJ12 Serial Port in ASCII Mode

Creating a *KEPDirect* Quick Client Project

KEPDirect Quick Client can be used to assist in the test and development of KEPware's OPC Data Access 1.0 and 2.0 Servers.

Connecting the Client to the OPC Server

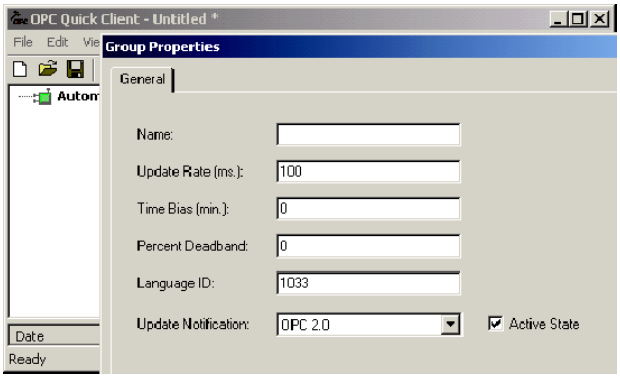
A server connection provides a link between the Quick Client and the *KEPDirect* OPC server. To add a server connection to the Quick Client, you can use the Edit menu>New Server Connection or click on the New Server icon in the toolbar menu.



Specify the Prog ID of the OPC Server the client should connect to. You can browse for registered servers by expanding any of the branches. Double-clicking on any registered server will automatically update the Prog ID field. For more information on the registered servers, click on the "Help" button to display the "Server Connection" section of the on-line help file. Once a connection to the OPC server has been established, additional "Server Operations" can be accessed by right clicking on the highlighted server in the right window column or by using the Tools menu>Server selection.

Creating a Client Group

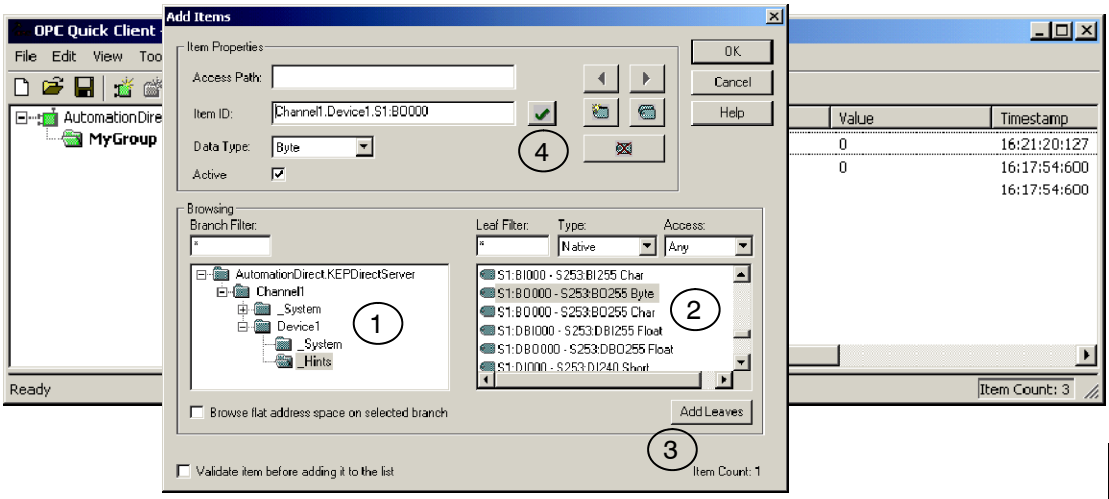
A group is used to organize a collection of items with a common set of properties. To add a Group to the Quick Client, you can use the Edit menu>New Group or click on the New Group icon in the toolbar menu.



The group specifies the following properties: group name, update rate, time bias, percent deadband, language ID, active state, and the type of data connection that should be made to the server. For detailed information on the group properties, click on the "Help" button to display the "Group" section of the on-line help file. Once a Group has been created, additional "Group Operations" can be accessed by right clicking on the highlighted branch Group or by using the Tools menu>Group selection.

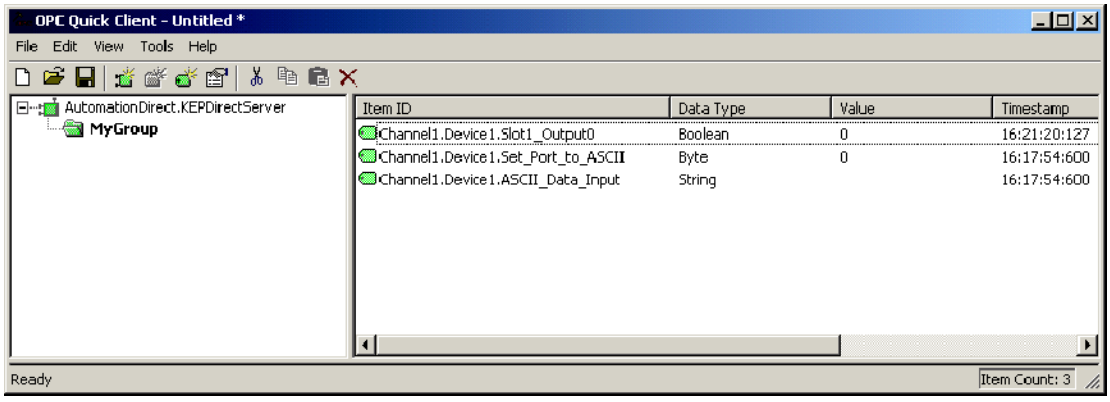
Selecting a Group Item

Items represent data that may be accessed via the OPC server. An item specifies the following properties: item ID, access path, requested data type and active state. For detailed information these properties, click on the Help button to display the “Item” section of the on-line help file. To add an Item to the Quick Client Group, you can use the Edit menu>New Item or click on the New Item icon in the toolbar menu.



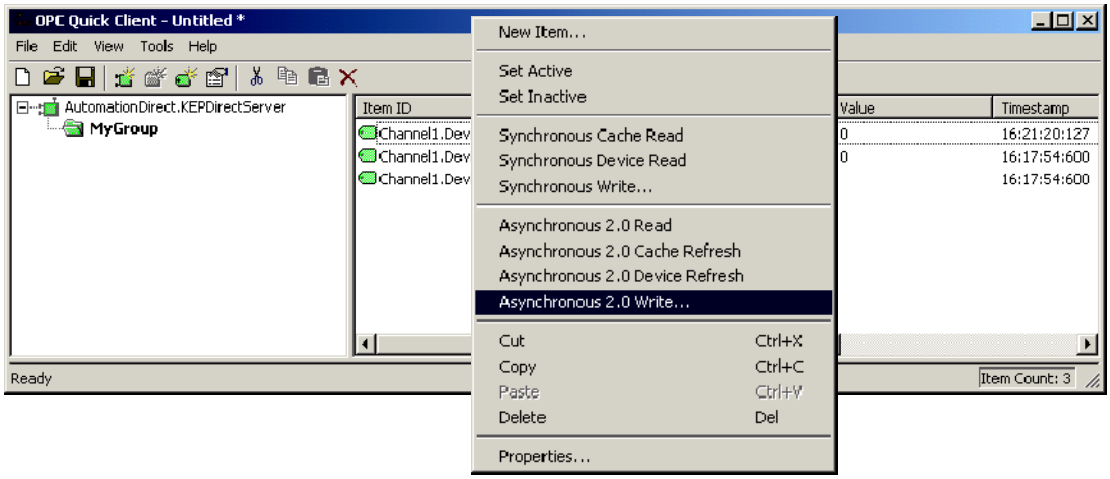
If the OPC Server was configured to automatically generate OPC tags, the generated tags would be browsable from the OPC client. If automatic tag generation was not selected, create an item by:
1) browsing the OPC Server branch tags
2) highlighting the desired tag in the right column
3) clicking on the “Add Leaves” button
4) clicking on the “Green Check Mark” button to validate the item
5) and clicking on the “OK” button.

After clicking on the “OK” button, the following window will be display the created items.

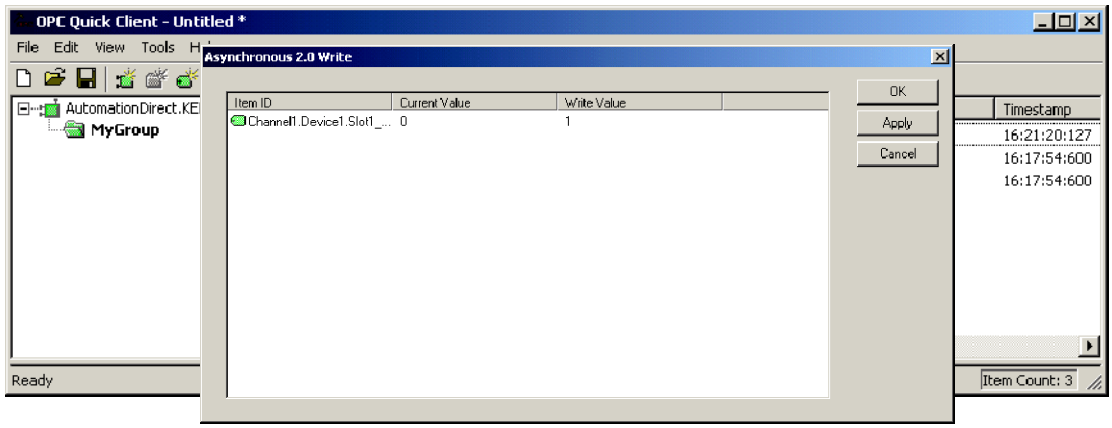


Item Operations

Item operations can be accessed by right clicking on the desired item or by using the Tools menu>Group selection.

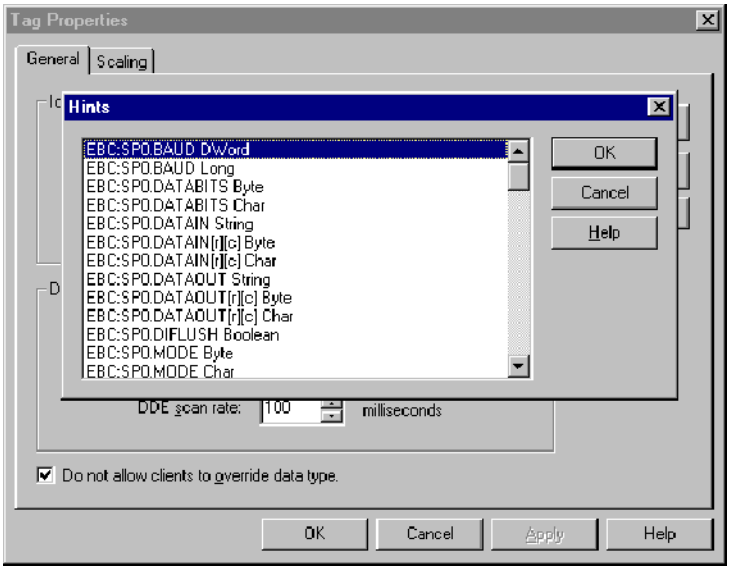


After clicking on the desired item operation, a window similar to the following will be displayed. In this example, a logical "1" (Boolean data type) is being written to a discrete output to turn it on. The item operations can be used to read discrete/analog inputs and write to discrete/analog outputs, etc.



Using the RJ12 Serial Port in ASCII Mode

The EBC RJ12 serial port can be configured for generic ASCII communications (refer to the “Advanced Settings” section in the “Using NetEdit” chapter to confirm or change the RJ12 serial port settings). Both the transmit buffer and receive buffer of the driver are 127 bytes in size. Thus, the corresponding tags can be a maximum of 127 bytes. Incoming bytes are appended to the receive buffer.
Port specifiers precede the serial port address. It defines which port the serial port address corresponds to. To define an EBC address the mnemonic ”EBC” is used and the mnemonic SP0 specifies serial port 0. For addressing the EBC serial port, no base or slot information is needed.
As shown below in the Hints dialog, there are several port address parameters. In many cases the default values can be used. A detailed list explaining the parameters are found by clicking on the “Help” button in the Hints window. Then click on the “Index” button in the Terminator I/O, I/O Addressing window. Then locate the “H2, H4, Terminator I/O Serial Port Addressing” help section.



The communication parameter defaults are:

9600 baud

8 data bits (7 may be selected)

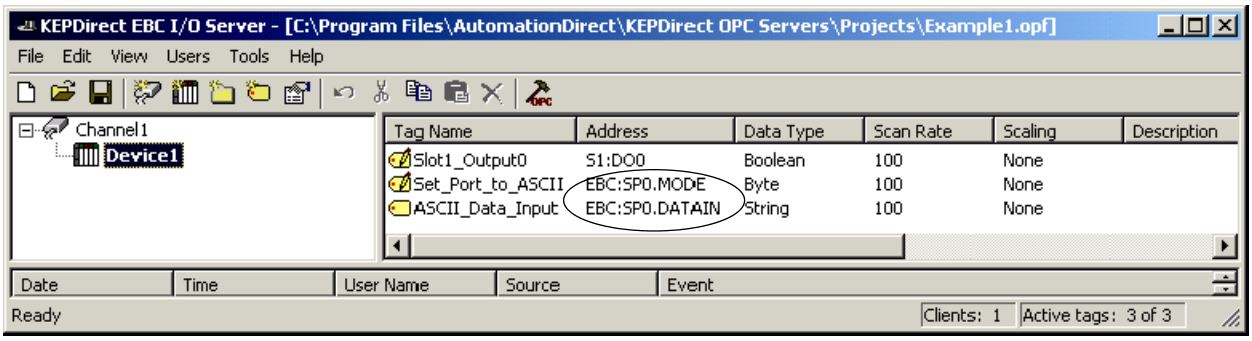
no parity (odd or even may be selected)

1 stop bits (2 may be selected)

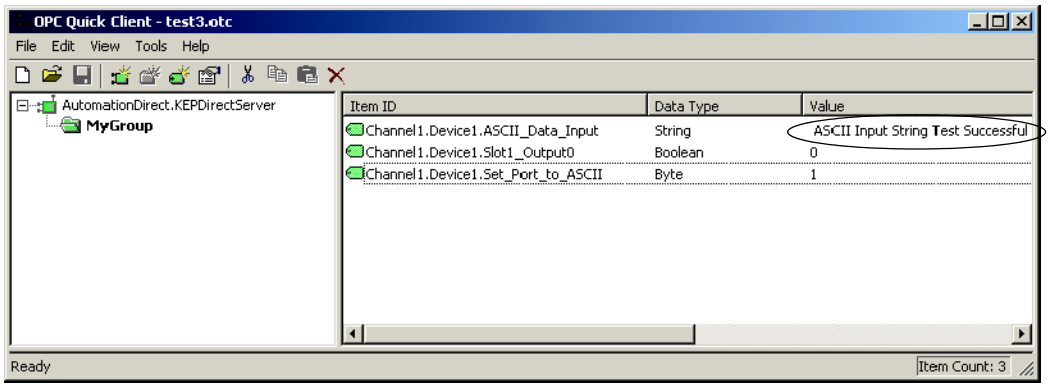
The following tags were created in the *KEPDirect* OPC server for this example.

EBC:SP0.MODE

EBC:SP0.DATAIN



The tags created above were browsed and selected as items within the Quick Client as shown below. The EBC.SP0.MODE address must be set to a value of 1 to select the ASCII communications mode. The ASCII string “ASCII String Input Test Successful” was entered via the RJ12 serial port. The ASCII Sting displays in the “ASCII Data Input “ Item ID’s Value column.

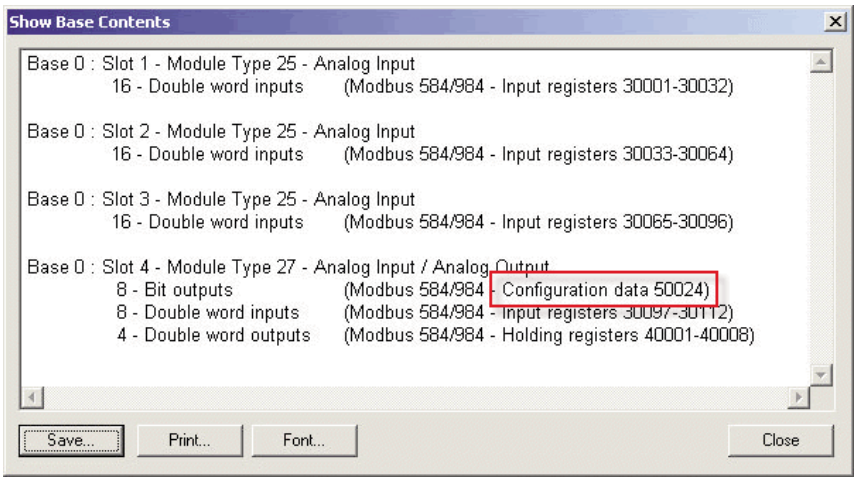


T1H-EBC100 Analog Module Addressing - Modbus TCP

In This Appendix...

T1H-EBC100 Analog Module Addressing - Modbus TCP

Using the NetEdit3 utility, find and select the IP address of the desired T1H-EBC(100). Then select the 'EBC Settings' tab and the 'Show Base Contents' button to see the I/O modules in the T1H-EBC100 base and the Modbus addressing for those modules. You should see something similar to the following:



Use the addresses shown in the 'Show Base Contents' section of NetEdit3 along with the following table to access the analog I/O with your Modbus TCP master.

For example, to read the current temperature detected by Channel 1 of the T1F-16RTD module in Slot 2, Modbus addresses 30033 and 30034 are required. For analog output modules with a Module Control Byte, find the required address in the Show Base Contents table. 'Configuration data' is the address needed (shown in the red box above). Then use the bit definitions below to determine how to configure the module. Be aware that the address could be displayed as a six digit or five digit address. In Modbus, 50024 and 410024 are equivalent. The screen capture shows it as five digit. DirectSOFT MRX and MWX instructions require six digit in the upper ranges.

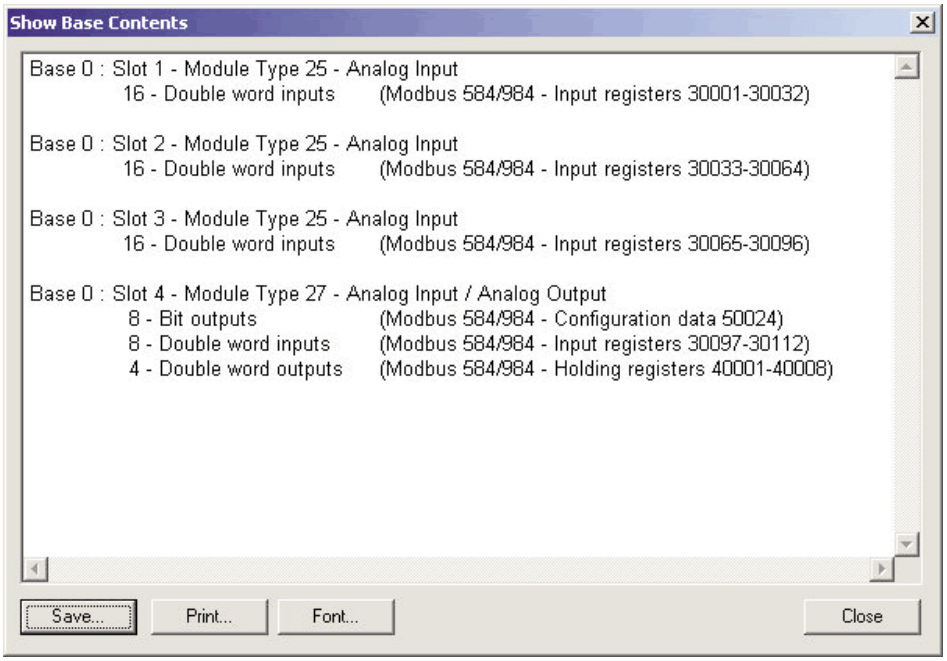
T1H-EBC100 Analog Module Addressing - Modbus TCP			
Part Number	**Channel Data**	**Module Configuration Data**	**Diagnostics Data**
T1F-08AD-1	Input Registers Word 1 = Ch1 Word 3 = Ch2 Word 5 = Ch3 Word 7 = Ch4 Word 9 = Ch5 Word 11 = Ch6 Word 13 = Ch7 Word 15 = Ch8	No Software Configuration Input Range Depends on Input Signal: -20 to 20mA = -8192 to 8191 0 to 20mA = 0 to 8191 4 to 20mA = 1638 to 8191	No Built-In Broken Transmitter Detection Monitor for counts less than 1638 when using 4 to 20mA
T1F-08AD-2	Input Registers Word 1 = Ch1 Word 3 = Ch2 Word 5 = Ch3 Word 7 = Ch4 Word 9 = Ch5 Word 11 = Ch6 Word 13 = Ch7 Word 15 = Ch8	No Software Configuration Input Range Depends on Input Signal: 0 to 5V = 0 to 4095 0 to 10V = 0 to 8191 +/-5V = -4095 to 4095 +/-10V = -8192 to 8191	No Broken Transmitter Detection (N/A for Voltage)
T1F-16AD-1	Input Registers Word 1 = Ch1 Word 3 = Ch2 Word 5 = Ch3 ... Word 27 = Ch14 Word 29 = Ch15 Word 31 = Ch16	No Software Configuration Input Range Depends on Input Signal: -20 to 20mA = -8192 to 8191 0 to 20mA = 0 to 8191 4 to 20mA = 1638 to 8191	No Built-In Broken Transmitter Detection Monitor for counts less than 1638 when using 4 to 20mA
T1F-16AD-2	Input Registers Word 1 = Ch1 Word 3 = Ch2 Word 5 = Ch3 ... Word 27 = Ch14 Word 29 = Ch15 Word 31 = Ch16	No Software Configuration Input Range Depends on Input Signal: 0 to 5V = 0 to 4095 0 to 10V = 0 to 8191 +/-5V = -4095 to 4095 +/-10V = -8192 to 8191	No Broken Transmitter Detection (N/A for Voltage)
T1F-14THM	Input Registers Word 1 = Ch1 Word 3 = Ch2 Word 5 = Ch3 ... Word 25 = Ch13 Word 27 = Ch14 Word 29 = Status Word 31 = Status	No Software Configuration THM Type Set by Jumpers	I/O Module Status: 37401: Indicates if an error is present 37405 to 37420: One word per channel gives 1 if broken transmitter, 0 if OK Once error condition is corrected write 0 (zero) to 410001 to clear errors
T1F-16RTD	Input Registers Word 1 = Ch1 Word 3 = Ch2 Word 5 = Ch3 ... Word 27 = Ch14 Word 29 = Ch15 Word 31 = Ch16	No Software Configuration RTD Type Set by Jumpers	I/O Module Status: 37401: Indicates if an error is present 37405 to 37420: One word per channel gives 1 if broken transmitter, 0 if OK Once error condition is corrected write 0 (zero) to 410001 to clear errors

T1H-EBC100 Analog Module Addressing - Modbus TCP				
Part Number	**Channel Data**	**Module Configuration Data**		**Diagnostics Data**
T1F-8AD4DA-1	Input Registers Word 1 = Ch1 Word 3 = Ch2 Word 5 = Ch3 Word 7 = Ch4 Word 9 = Ch5 Word 11 = Ch6 Word 13 = Ch7 Word 15 = Ch8 Holding Registers Word 1 = Ch1 Word 3 = Ch2 Word 5 = Ch3 Word 7 = Ch4	Analog Output Configuration (T1F-8AD4DA-1)		No Built-In Broken Transmitter Detection Monitor for counts less than 1638 when using 4 to 20mA
		Bit 0	Output Enable 0: Outputs OFF 1: Outputs Enabled	
		Bit 1	N/A	
		Bit 2	N/A	
		Bit 3	0-20mA or 4-20mA 0: 0-20mA range 1: 4-20mA range	
		Bits 4-15	Reserved	
		Input Range Depends on Input Signal: -20 to 20mA = -8192 to 8191 0 to 20mA = 0 to 8191 4 to 20mA = 1638 to 8191		
T1F-8AD4DA-2	Input Registers Word 1 = Ch1 Word 3 = Ch2 Word 5 = Ch3 Word 7 = Ch4 Word 9 = Ch5 Word 11 = Ch6 Word 13 = Ch7 Word 15 = Ch8 Holding Registers Word 1 = Ch1 Word 3 = Ch2 Word 5 = Ch3 Word 7 = Ch4	Analog Output Configuration (T1F-8AD4DA-2)		No Broken Transmitter Detection (N/A for Voltage)
		Bit 0	Output Enable 0: Outputs OFF 1: Outputs Enabled	
		Bit 1	Unipolar/Bipolar 0: Unipolar 1: Bipolar	
		Bit 2	5V/10V Range 0: 5V Range 1: 10V Range	
		Bit 3	N/A	
		Bits 4-15	Reserved	
		Input Range Depends on Input Signal: 0 to 5V = 0 to 4095 0 to 10V = 0 to 8191 +/-5V = -4095 to 4095 +/-10V = -8192 to 8191		
T1F-08DA-1	Holding Registers Word 1 = Ch1 Word 3 = Ch2 Word 5 = Ch3 Word 7 = Ch4 Word 9 = Ch5 Word 11 = Ch6 Word 13 = Ch7 Word 15 = Ch8	Analog Output Configuration (T1F-08DA-1)		None
		Bit 0	Output Enable 0: Outputs OFF 1: Outputs Enabled	
		Bit 1	N/A	
		Bit 2	N/A	
		Bit 3	0-20mA or 4-20mA 0: 0-20mA range 1: 4-20mA range	
		Bits 4-15	Reserved	

T1H-EBC100 Analog Module Addressing - Modbus TCP				
Part Number	**Channel Data**	**Module Configuration Data**		**Diagnostics Data**
T1F-08DA-2	Holding Registers Word 1 = Ch1 Word 3 = Ch2 Word 5 = Ch3 Word 7 = Ch4 Word 9 = Ch5 Word 11 = Ch6 Word 13 = Ch7 Word 15 = Ch8	Analog Output Configuration (T1F-08DA-2)		None
		Bit 0	Output Enable 0: Outputs OFF 1: Outputs Enabled	
		Bit 1	Unipolar/Bipolar 0: Unipolar 1: Bipolar	
		Bit 2	5V/10V Range 0: 5V Range 1: 10V Range	
		Bit 3	N/A	
		Bits 4-15	Reserved	
T1F-16DA-1	Holding Registers Word 1 = Ch1 Word 3 = Ch2 Word 5 = Ch3 ... Word 27 = Ch14 Word 29 = Ch15 Word 31 = Ch16	Analog Output Configuration (T1F-16DA-1)		None
		Bit 0	Output Enable 0: Outputs OFF 1: Outputs Enabled	
		Bit 1	N/A	
		Bit 2	N/A	
		Bit 3	0-20mA or 4-20mA 0: 0-20mA range 1: 4-20mA range	
		Bits 4-15	Reserved	
T1F-16DA-2	Holding Registers Word 1 = Ch1 Word 3 = Ch2 Word 5 = Ch3 ... Word 27 = Ch14 Word 29 = Ch15 Word 31 = Ch16	Analog Output Configuration (T1F-16DA-2)		None
		Bit 0	Output Enable 0: Outputs OFF 1: Outputs Enabled	
		Bit 1	Unipolar/Bipolar 0: Unipolar 1: Bipolar	
		Bit 2	5V/10V Range 0: 5V Range 1: 10V Range	
		Bit 3	N/A	
		Bits 4-15	Reserved	

T1F-8AD4DA-2 Example (Module in Slot 4)

Using the 'Show Base Contents' dialog below and the 'T1H-EBC100 Analog Module Addressing - Modbus TCP' chart above, we can find all of the addresses associated with the T1F-8AD4DA-2 module in Slot 4. The addresses are in the following chart. Be aware that the configuration data address could be displayed as a six digit or five digit address. In Modbus, 50024 and 410024 are equivalent. The screen capture below shows it as five digit.



Input Channel	Address
Channel 1	30097 - 30098
Channel 2	30099 - 30100
Channel 3	30101 - 30102
Channel 4	30103 - 30104
Channel 5	30105 - 30106
Channel 6	30107 - 30108
Channel 7	30109 - 30110
Channel 8	30111 - 30112
Output Channel	**Address**
Channel 1	40001 - 40002
Channel 2	40003 - 40004
Channel 3	40005 - 40006
Channel 4	40007 - 40008

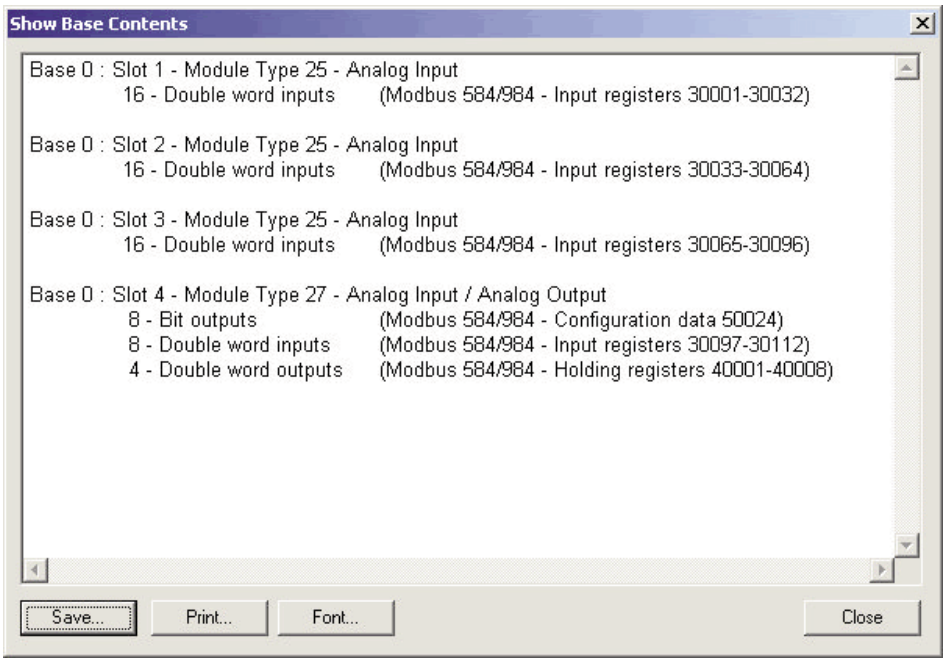
Control Byte (Address 50024)	
Bit	**Function**
Bit 0	Output Enable 0: Outputs OFF 1: Outputs Enabled
Bit 1	Unipolar/Bipolar 0: Unipolar 1: Bipolar
Bit 2	5V/10V Range 0: 5V Range 1: 10V Range
Bit 3	N/A
Bit 4	Reserved

The chart below, lists all of the possible Control Byte combinations and their resulting configuration.

Control Byte (Address 50024)	Corresponding Configuration		
	Output Enable/Disable	Unipolar/Bipolar	5V/10V Range
0	Disabled	Unipolar	5V
1	Enabled		
2	Disabled	Bipolar	
3	Enabled		
4	Disabled	Unipolar	10V
5	Enabled		
6	Disabled	Bipolar	
7	Enabled		

T1F-16RTD Example (Module in Slot 2)

Using the 'Show Base Contents' dialog below and the 'T1H-EBC100 Analog Module Addressing - Modbus TCP' chart, we can find all of the addresses associated with the T1F-16RTD module in Slot 2. The addresses are in the following chart.



Input Channel	Address	Error Words
Channel 1	30033 - 30034	37421 is the Error Flag Word: For Words 2 to 4, refer to the Current/Last State Error Codes Table (Chapter 4) Extended Error Codes (0 = Transmitter OK, 1 = Broken Transmitter) 37425 is the Extended Error Code for Ch1 37426 is the Extended Error Code for Ch2 37427 is the Extended Error Code for Ch3 ... 37438 is the Extended Error Code for Ch14 37439 is the Extended Error Code for Ch15 37440 is the Extended Error Code for Ch16 410001 is the EBC Dynamic Module Data Error Code (Write zero to clear errors)
Channel 2	30035 - 30036	
Channel 3	30037 - 30038	
Channel 4	30039 - 30040	
Channel 5	30041 - 30042	
Channel 6	30043 - 30044	
Channel 7	30045 - 30046	
Channel 8	30047 - 30048	
Channel 9	30049 - 30050	
Channel 10	30051 - 30052	
Channel 11	30053 - 30054	
Channel 12	30055 - 30056	
Channel 13	30057 - 30058	
Channel 14	30059 - 30060	
Channel 15	30061 - 30062	
Channel 16	30063 - 30064	

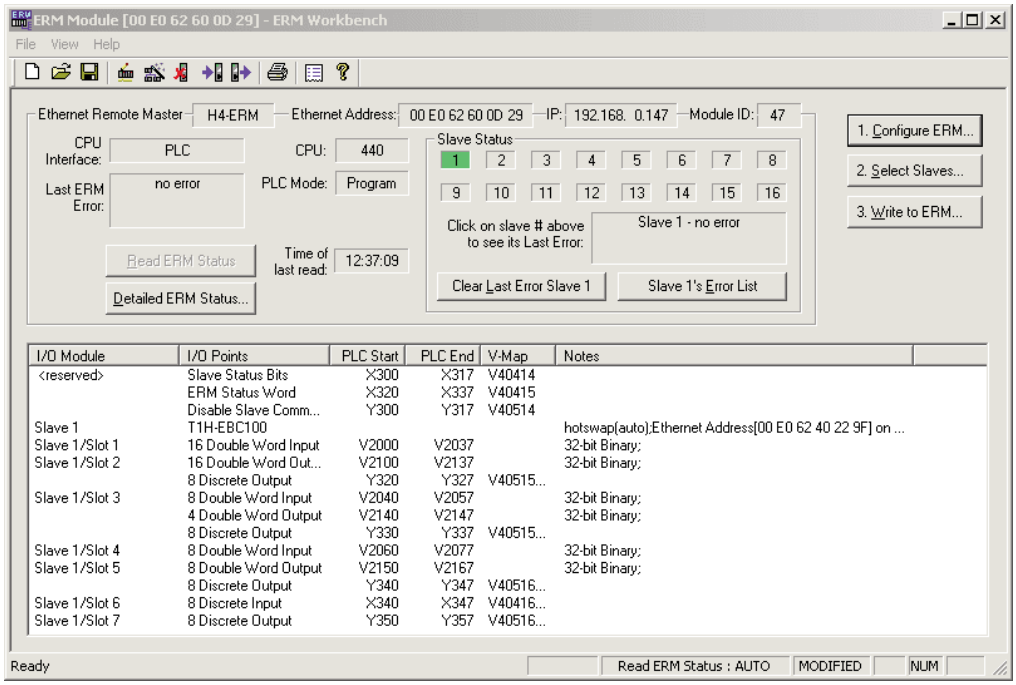
T1H-EBC(100) Analog Module Addressing - H2/4-ERM(100)

In This Appendix...

T1H-EBC(100) Analog Module Addressing - H2/4-ERM(100)

When using an H2-ERM(100) or H4-ERM(100) to T1H-EBC(100) configuration, the analog module data in the EBC base is mapped to V-memory or Discrete I/O.

The ERM Workbench software will tell you what the mapping is for each I/O module in the T1H-EBC(100) base. Once you have configured the ERM using ERM Workbench you will get a screen similar to the following:



For the example above, the I/O configuration for Slave 1 is:

Slot 1 = T1F-14THM
Slot 2 = T1F-16DA-2
Slot 3 = T1F-8AD2DA-2
Slot 4 = T1F-08AD-2
Slot 5 = T1F-08DA-2
Slot 6 = T1K-08NA-1
Slot 7 = T1K-08TR

Use the addresses shown in the ERM Workbench along with the following table to access the analog I/O with your ERM master.

<table>
<tr><th colspan="4">T1H-EBC(100) Analog Module Addressing - H2/4-ERM(100)</th></tr>
<tr><th>Part Number</th><th>Channel Data</th><th>Module Configuration Data</th><th>Diagnostics Data</th></tr>
<tr><td>T1F-08AD-1</td><td>Input Data
V+0 = Ch1
V+2 = Ch2
V+4 = Ch3
V+6 = Ch4
V+10 = Ch5
V+12 = Ch6
V+14 = Ch7
V+16 = Ch8</td><td>No Software Configuration

Input Range Depends on Input Signal:
-20 to 20mA = -8192 to 8191
0 to 20mA = 0 to 8191
4 to 20mA = 1638 to 8191</td><td>No Built-In Broken Transmitter Detection
Monitor for counts less than 1638 when using 4 to 20mA</td></tr>
<tr><td>T1F-08AD-2</td><td>Input Data
V+0 = Ch1
V+2 = Ch2
V+4 = Ch3
V+6 = Ch4
V+10 = Ch5
V+12 = Ch6
V+14 = Ch7
V+16 = Ch8</td><td>No Software Configuration

Input Range Depends on Input Signal:
0 to 5V = 0 to 4095
0 to 10V = 0 to 8191
+/-5V = -4095 to 4095
+/-10V = -8192 to 8191</td><td>No Broken Transmitter Detection
(N/A for Voltage)</td></tr>
<tr><td>T1F-16AD-1</td><td>Input Data
V+0 = Ch1
V+2 = Ch2
V+4 = Ch3
...
V+34 = Ch15
V+36 = Ch16</td><td>No Software Configuration

Input Range Depends on Input Signal:
-20 to 20mA = -8192 to 8191
0 to 20mA = 0 to 8191
4 to 20mA = 1638 to 8191</td><td>No Built-In Broken Transmitter Detection
Monitor for counts less than 1638 when using 4 to 20mA</td></tr>
<tr><td>T1F-16AD-2</td><td>Input Data
V+0 = Ch1
V+2 = Ch2
V+4 = Ch3
...
V+34 = Ch15
V+36 = Ch16</td><td>No Software Configuration

Input Range Depends on Input Signal:
0 to 5V = 0 to 4095
0 to 10V = 0 to 8191
+/-5V = -4095 to 4095
+/-10V = -8192 to 8191</td><td>No Broken Transmitter Detection
(N/A for Voltage)</td></tr>
<tr><td>T1F-14THM</td><td>Input Data
V+0 = Ch1
V+2 = Ch2
V+4 = Ch3
...
V+32 = Ch14
V+34 = Status 1
V+36 = Status 2

Status info is only available if T1F- 14THM is date code 1205 or later</td><td>No Software Configuration

THM Type Set by Jumpers

<table><tr><th colspan="2">Status 1 Data</th></tr><tr><td>Bits 0-3</td><td>Number of Channels Enabled (Inverted)
0001 = All Channels
1110 = One Channel</td></tr><tr><td>Bit 4</td><td>T/C Type Jumper 0
0=installed, 1=removed</td></tr><tr><td>Bit 5</td><td>T/C Type Jumper 1</td></tr><tr><td>Bit 6</td><td>T/C Type Jumper 2</td></tr><tr><td>Bit 7</td><td>T/C Type Jumper 3</td></tr><tr><td>Bit 8</td><td>Units 0 Jumper</td></tr><tr><td>Bit 9</td><td>Units 1 Jumper</td></tr><tr><td>Bit 10</td><td>Calibrate Enable Jumper</td></tr><tr><td>Bit 11</td><td>CJC Installed 0=Yes, 1=No</td></tr><tr><td>Bits 12,13</td><td>Always ON</td></tr><tr><td>Bits 14, 15</td><td>Always OFF</td></tr></table>
Status 2 Data is the Temperature of the CJC with one implied decimal place.</td><td>Broken Thermocouple Indication.
The channel data goes to zero and ERM Workbench ‘Slave Error List’ shows error in ‘Extended Error column.</td></tr>
</table>

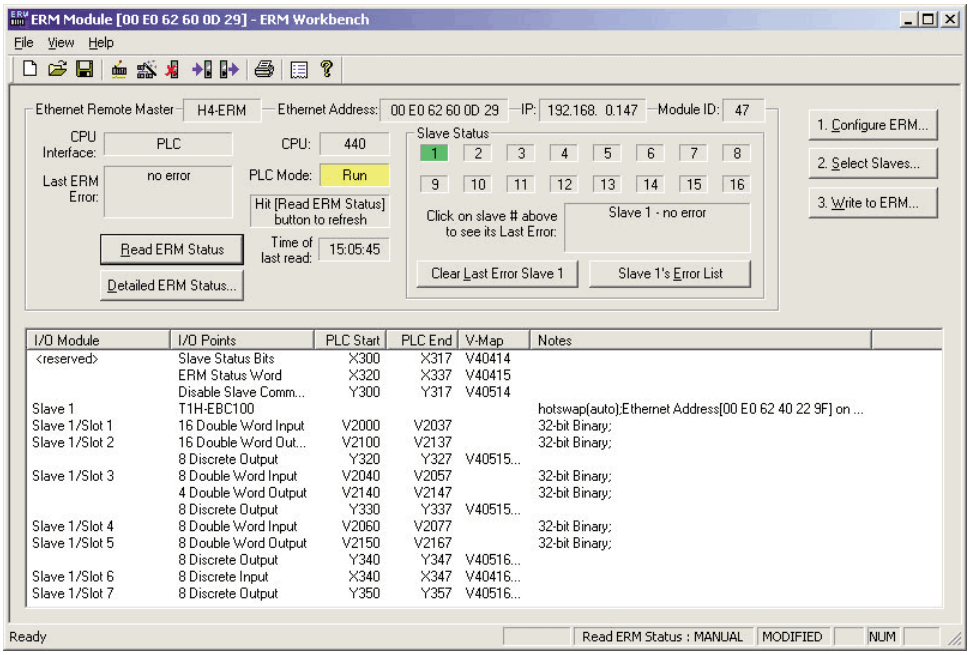
T1H-EBC(100) Analog Module Addressing - H2/4-ERM(100)

Part Number	Channel Data	Module Configuration Data		Diagnostics Data
T1F-16RTD	Input Data V+0 = Ch1 V+2 = Ch2 V+4 = Ch3 ... V+34 = Ch15 V+36 = Ch16	No Software Configuration RTD Type Set by Jumpers		Broken RTD Indication. The channel data goes to zero and ERM Workbench 'Slave Error List' shows error in 'Extended Error' column.
T1F-8AD4DA-1	Input Data V+0 = Ch1 V+2 = Ch2 V+4 = Ch3 V+6 = Ch4 V+10 = Ch5 V+12 = Ch6 V+14 = Ch7 V+16 = Ch8 Output Data V+0 = Ch1 V+2 = Ch2 V+4 = Ch3 V+6 = Ch4	Analog Output Configuration (T1F-8AD4DA-1)		No Built-In Broken Transmitter Detection Monitor for counts less than 1638 when using 4 to 20mA
		Y+0	Output Enable 0: Outputs OFF 1: Outputs Enabled	
		Y+1	N/A	
		Y+2	N/A	
		Y+3	0-20mA or 4-20mA 0: 0-20mA range 1: 4-20mA range	
		Y+4 to Y+7	Reserved	
		Input Range Depends on Input Signal: -20 to 20mA = -8192 to 8191 0 to 20mA = 0 to 8191 4 to 20mA = 1638 to 8191		
T1F-8AD4DA-2	Input Data V+0 = Ch1 V+2 = Ch2 V+4 = Ch3 V+6 = Ch4 V+10 = Ch5 V+12 = Ch6 V+14 = Ch7 V+16 = Ch8 Output Data V+0 = Ch1 V+2 = Ch2 V+4 = Ch3 V+6 = Ch4	Analog Output Configuration (T1F-8AD4DA-2)		No Broken Transmitter Detection (N/A for Voltage)
		Y+0	Output Enable 0: Outputs OFF 1: Outputs Enabled	
		Y+1	Unipolar/Bipolar 0: Unipolar 1: Bipolar	
		Y+2	5V/10V Range 0: 5V Range 1: 10V Range	
		Y+3	N/A	
		Y+4 to Y+7	Reserved	
		Input Range Depends on Input Signal: 0 to 5V = 0 to 4095 0 to 10V = 0 to 8191 +/-5V = -4095 to 4095 +/-10V = -8192 to 8191		
T1F-08DA-1	Output Data V+0 = Ch1 V+2 = Ch2 V+4 = Ch3 V+6 = Ch4 V+10 = Ch5 V+12 = Ch6 V+14 = Ch7 V+16 = Ch8	Analog Output Configuration (T1F-08DA-1)		None
		Y+0	Output Enable 0: Outputs OFF 1: Outputs Enabled	
		Y+1	N/A	
		Y+2	N/A	
		Y+3	0-20mA or 4-20mA 0: 0-20mA range 1: 4-20mA range	
		Y+4 to Y+7	Reserved	

T1H-EBC(100) Analog Module Addressing - H2/4-ERM(100)				
Part Number	**Channel Data**	**Module Configuration Data**		**Diagnostics Data**
T1F-08DA-2	Output Data V+0 = Ch1 V+2 = Ch2 V+4 = Ch3 V+6 = Ch4 V+10 = Ch5 V+12 = Ch6 V+14 = Ch7 V+16 = Ch8	Analog Output Configuration (T1F-08DA-2)		None
		Y+0	Output Enable 0: Outputs OFF 1: Outputs Enabled	
		Y+1	Unipolar/Bipolar 0: Unipolar 1: Bipolar	
		Y+2	5V/10V Range 0: 5V Range 1: 10V Range	
		Y+3	N/A	
		Y+4 to Y+7	Reserved	
T1F-16DA-1	Output Data V+0 = Ch1 V+2 = Ch2 V+4 = Ch3 ... V+34 = Ch15 V+36 = Ch16	Analog Output Configuration (T1F-16DA-1)		None
		Y+0	Output Enable 0: Outputs OFF 1: Outputs Enabled	
		Y+1	N/A	
		Y+2	N/A	
		Y+3	0-20mA or 4-20mA 0: 0-20mA range 1: 4-20mA range	
		Y+4 to Y+7	Reserved	
T1F-16DA-2	Output Data V+0 = Ch1 V+2 = Ch2 V+4 = Ch3 ... V+34 = Ch15 V+36 = Ch16	Analog Output Configuration (T1F-16DA-2)		None
		Y+0	Output Enable 0: Outputs OFF 1: Outputs Enabled	
		Y+1	Unipolar/Bipolar 0: Unipolar 1: Bipolar	
		Y+2	5V/10V Range 0: 5V Range 1: 10V Range	
		Y+3	N/A	
		Y+4 to Y+7	Reserved	

T1F-14THM Example (Module in Slot 1)

Using ERM Workbench (below) and the 'T1H-EBC(100) Analog Module Addressing - H2/4-ERM(100) chart above, we can find all of the addresses associated with the T1F-14THM module in Slot 1. The addresses are listed in the tables below.



Input Channel	Address
Channel 1 Temperature	V2000
Channel 2 Temperature	V2002
Channel 3 Temperature	V2004
Channel 4 Temperature	V2006
Channel 5 Temperature	V2010
Channel 6 Temperature	V2012
Channel 7 Temperature	V2014
Channel 8 Temperature	V2016
Channel 9 Temperature	V2020
Channel 10 Temperature	V2022
Channel 11 Temperature	V2024
Channel 12 Temperature	V2026
Channel 13 Temperature	V2030
Channel 14 Temperature	V2032
Status Words	**Address**
Status1	V2034
CJC Temperature	V2036

In this example, 24VDC is applied to the T1F-14THM in Slave 1 Slot 1 and all channels are shorted CH+ to CH-. As seen in the DirectSoft Data View window below, all channels will read the terminal block ambient temperature when shorted (degrees F in this configuration). All V-memory values in this DirectSoft Data View window are displayed as Decimal DWORDs except V2034 which is displayed as a Binary WORD.

V2036 is the CJC temperature reading in degrees C with one implied decimal place. So 27.2°C = 80.9°F.

Data1

E↓ Decimal DWORD

	Element	Status
1		
2	V2000	806
3	V2002	809
4	V2004	811
5	V2006	815
6	V2010	811
7	V2012	819
8	V2014	821
9	V2016	813
10	V2020	799
11	V2022	803
12	V2024	805
13	V2026	809
14	V2030	806
15	V2032	788
16	V2034	0011010000000001
17	V2036	272

V2034 Status 1 Word	
Bits 0-3	All Channels Enabled (0001)
Bit 4	T/C Type Jumper 0 Installed (0)
Bit 5	T/C Type Jumper 1 Installed (0)
Bit 6	T/C Type Jumper 2 Installed (0)
Bit 7	T/C Type Jumper 3 Installed (0)
Bit 8	Units 0 Jumper Installed (0)
Bit 9	Units 1 Jumper Installed (0)
Bit 10	Calibrate Enable Jumper Removed (1)
Bit 11	CJC Installed Yes (0)
Bits 12,13	Always ON
Bits 14, 15	Always OFF

T1F-14THM Example (Broken Thermocouple)

In this example, 24VDC is applied to the T1F-14THM in Slave 1 Slot 1 and all channels are shorted CH+ to CH- except Channel 8 which is open.

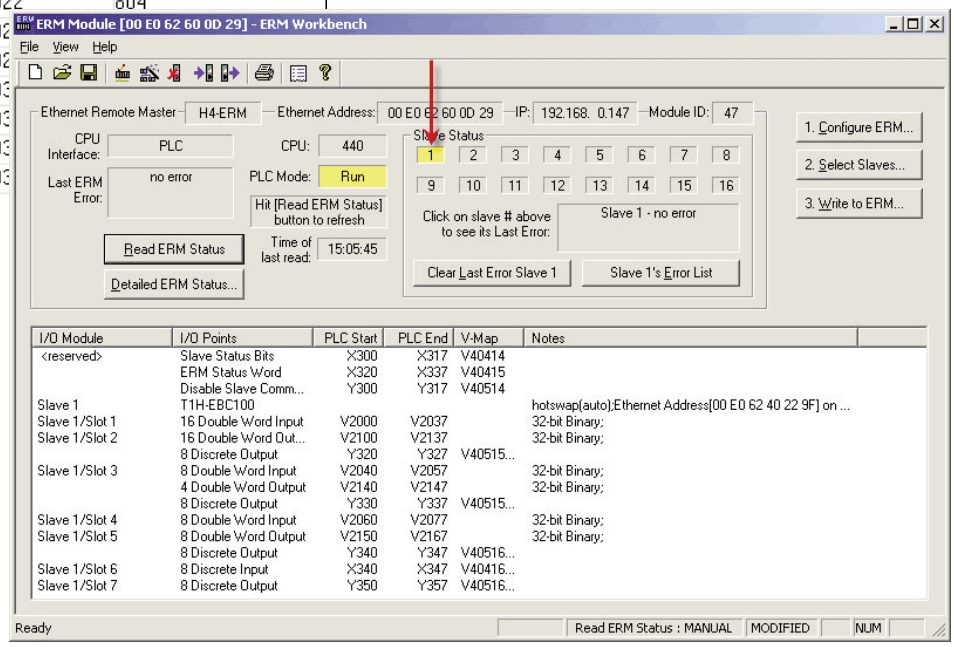
Data1

Binary | WORD

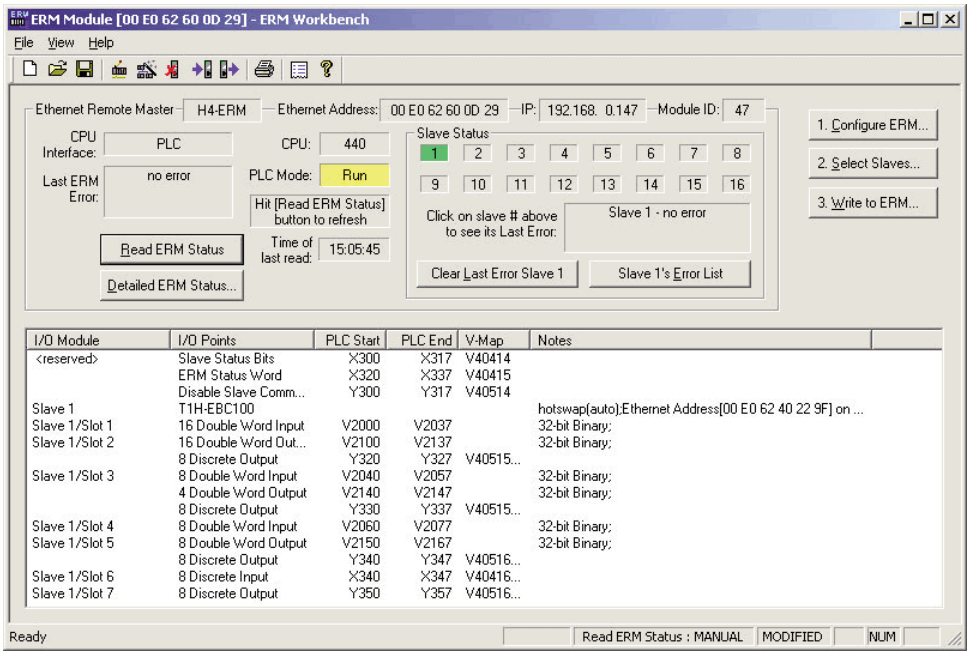
	Element	Status
1		
2	V2000	807
3	V2002	810
4	V2004	812
5	V2006	815
6	V2010	812
7	V2012	820
8	V2014	821
9	V2016	0
10	V2020	801
11	V2022	804
12	V202	
13	V202	
14	V203	
15	V203	
16	V203	
17	V203	

As seen in the DirectSoft Data View window, all channels will read the terminal block ambient temperature when shorted (degrees F in this configuration) except the open Channel 8 which reads 0.

The ERM Workbench page will also indicate an error on Slave 1 as seen below.



T1F-16DA-2 Example (Module in Slot 2)



In this example, 24VDC is applied to the T1F-16DA-2 in Slave 1 Slot 2 and a multi-meter is used to measure the output. The outputs are enabled and configured for -5 to +5V range.

All V-memory values in this DirectSoft Data View window are displayed as Decimal DWORDs.

Data1

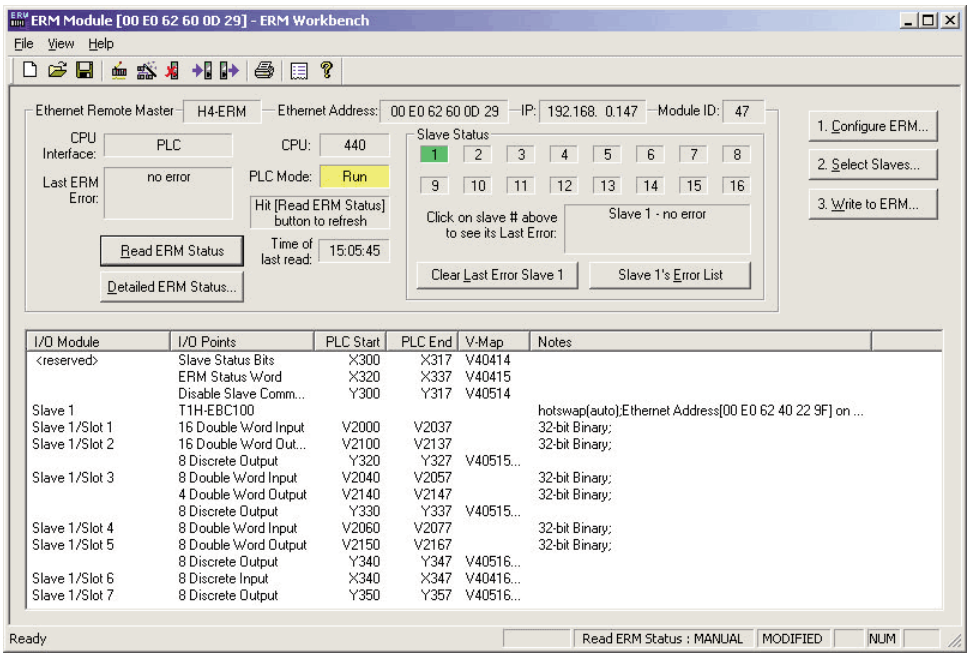
Decimal | DWORD

	Element	Status	Edits
1	V2100	0	0
2	V2102	270	270
3	V2104	525	525
4	V2106	780	780
5	V2110	1035	1035
6	V2112	1545	1545
7	V2114	1800	1800
8	V2116	2055	2055
9	V2120	2310	2310
10	V2122	2565	2565
11	V2124	2820	2820
12	V2126	3075	3075
13	V2130	3330	3330
14	V2132	3585	3585
15	V2134	3840	3840
16	V2136	4095	4095
17	Y320	ON	ON OFF
18	Y321	ON	ON OFF
19	Y322	OFF	ON OFF
20	Y323	OFF	ON OFF
21	Y324	OFF	ON OFF
22	Y325	OFF	ON OFF
23	Y326	OFF	ON OFF
24	Y327	OFF	ON OFF

Output Channel	Address	Value
Channel 1	V2100	0 = -5V
Channel 2	V2102	270 = -4.34V
Channel 3	V2104	525 = -3.71V
Channel 4	V2106	780 = -3.09V
Channel 5	V2110	1035 = -2.47V
Channel 6	V2112	1545 = -1.22V
Channel 7	V2114	1800 = -0.60V
Channel 8	V2116	2055 = 0.01V
Channel 9	V2120	2310 = 0.64V
Channel 10	V2122	2565 = 1.26V
Channel 11	V2124	2820 = 1.88V
Channel 12	V2126	3075 = 2.50V
Channel 13	V2130	3330 = 3.13V
Channel 14	V2132	3585 = 3.75V
Channel 15	V2134	3840 = 4.37V
Channel 16	V2136	4095 = 5V

Discrete Bits	Value
Y320	ON for Output Enable
Y321	ON selects Bipolar output
Y322	OFF selects 5V output range
Y323 to Y327	N/A

T1F-08AD-2 Example (Module in Slot 4)



In this example, 24VDC is applied to the T1F-08AD-2 in Slave 1 Slot 4. Voltage is applied to all eight channels.

V2060 and V2062 are displayed as both Signed Decimal DWORD and BCD/Hex DWORD in this DirectSoft Data View. V2064-V2076 are displayed as Signed Decimal DWORD.

Data3

E↓ BCD/Hex DWORD

	Element	Status
1	V2060	-4097
2	V2060	FFFFEFFF
3	V2062	-4097
4	V2062	FFFFEFFF
5	V2064	1
6	V2066	1
7	V2070	4097
8	V2072	4097
9	V2074	8190
10	V2076	8191

Input Channel	Address	Value
Channel 1	V2060	-4097 = -5V
Channel 2	V2062	-4097 = -5V
Channel 3	V2064	1 = 0V
Channel 4	V2066	1 = 0V
Channel 5	V2070	4097 = 5V
Channel 6	V2072	4097 = 5V
Channel 7	V2074	8190 = 10V
Channel 8	V2076	8190 = 10V

Notes: