

## Using IDENT M System T with Ethernet/IP

### ***Introduction***

The Pepperl+Fuchs IDENT M System T consists of two models [MTT3000-F180-B12-V45-MON](#), which is a read only unit and the [MTT6000-F120-B12-V45](#) which is a read/write unit. Tags that can be used are MTO-xx which have an 8 byte read only number on them and the MTM-C2 which have an 8 byte read only number and 71 bytes of read/write data.

### ***Ethernet/IP***

The IDENT M System T has RS232, RS485 and Ethernet TCP/IP ports. It does not however directly support Ethernet/IP. In order to talk Ethernet/IP use the converter RTS-UP-1 unit to either convert the TCP/IP or serial data to Ethernet/IP.

[RTS-UP-1](#) – Converts one serial and one Ethernet device to Ethernet/IP

[RTS-UP-4](#) – Converts four serial and four Ethernet devices to Ethernet/IP

This document will show you step by step how to read and write to the MTT devices using Ethernet/IP

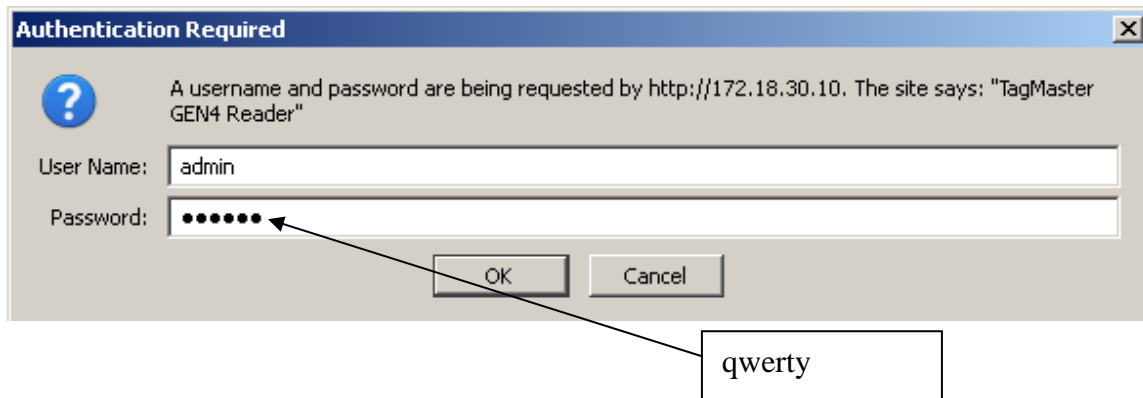
## ***Configure the MTT... devices***

### ***Set IP Address***

The first thing to do is to configure the Ethernet settings of your RFID system. Here are the default parameters.

Default Ethernet Settings: IP Address: 192.168.0.2  
Subnet Mask: 255.255.255.0

Put this IP address into your web browser and logon to the MTT... device. Make sure the IP address of your PC is close. For example set it to 192.168.0.1.



Go to the Settings > System > Network tab and set the network parameters correctly as required by your network administrator.



**Start**  
Information...  
Settings...  
System...  
  Passwords  
  Date & Time  
  **Network**  
  TAGD  
  Options  
Applications...  
Clone  
Web Tools...  
Log Files  
Reboot

### Network Settings

DHCP:    
Bonjour:    
Hostname:   
DynDNS username:   
DynDNS password:   
DynDNS hostname:

*Values below are used when DHCP is off or no DHCP server is available.*

IP address:   
Netmask:   
Gateway:   
Primary DNS:   
Secondary DNS:

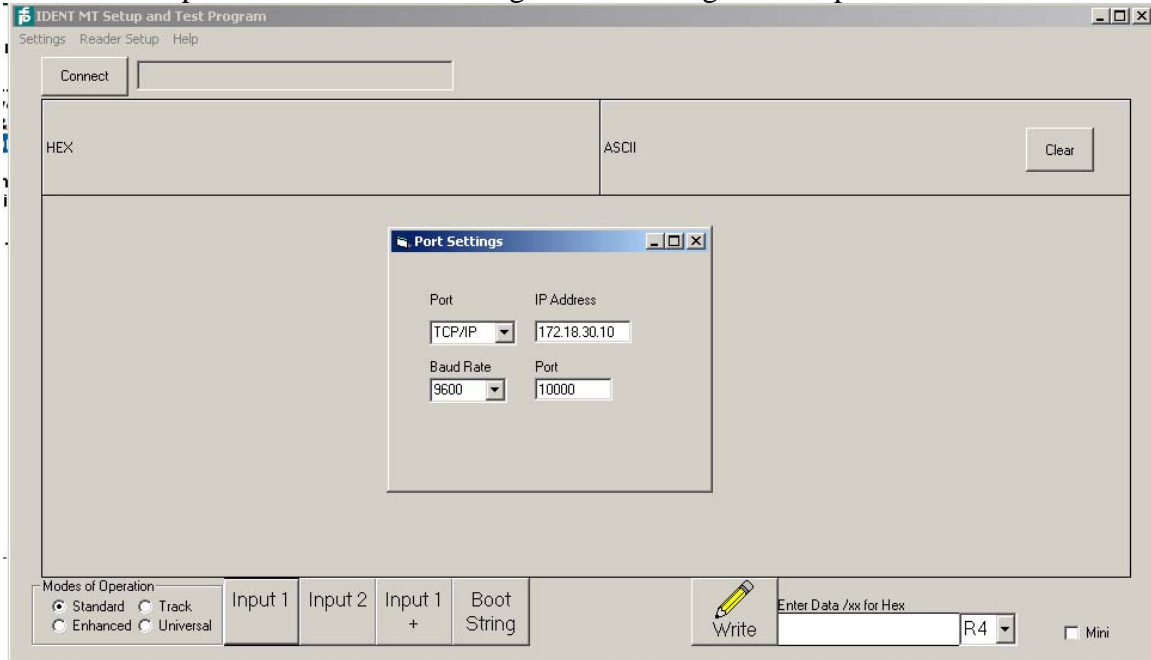
*Note: A reboot is required for these settings to take effect!*

Network configuration screen for the MTT...

After you change the IP address reboot the hardware so the settings will take affect. Reset the IP address of your PC to reconnect to it.

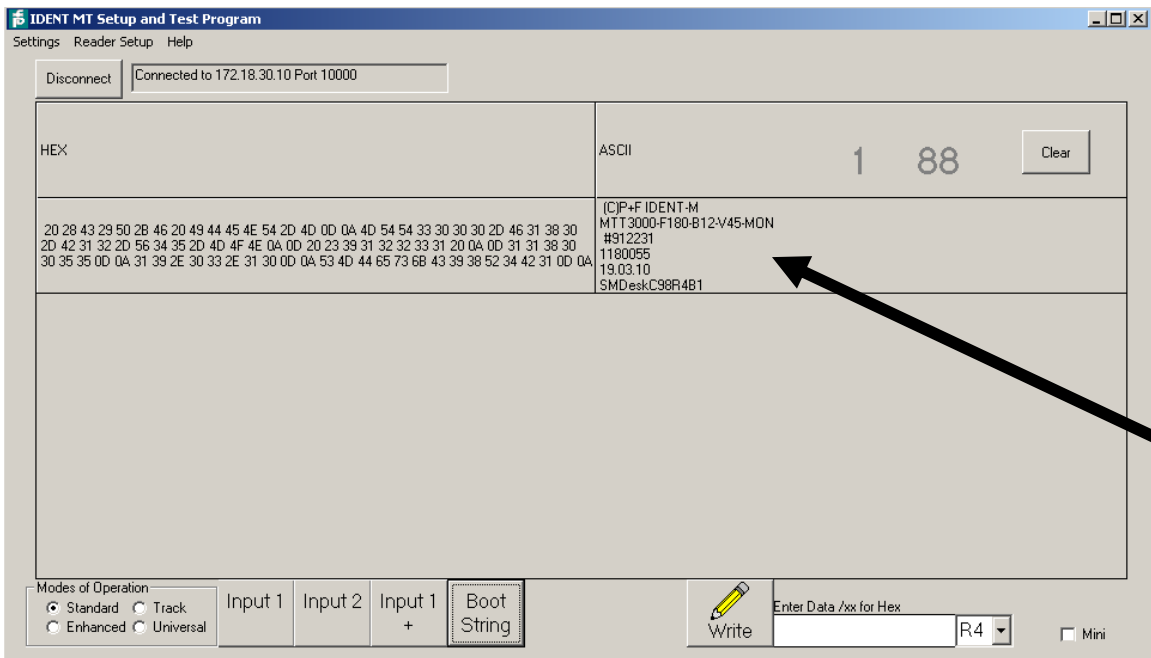
## Configure reader

Download the [Configuration and diagnostic software](#) from the web site. This software will connect to the serial or Ethernet ports so that a configuration can be made. Put your new Ethernet parameters into the Settings > Port settings menu option.



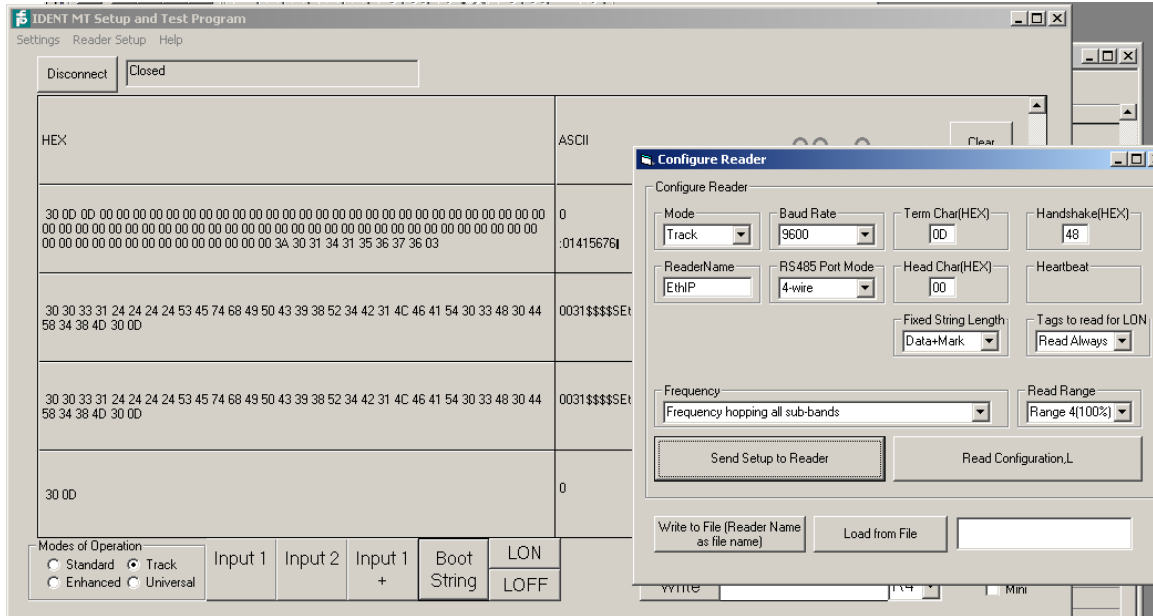
Setting the Network parameters to connect to an MTT... device

Close the port settings dialog box and press connect. It should say connected at the top. Press the Boot String button at bottom to verify that you have a Pepperl+Fuchs ID system connected.



## Reading the version information of an MTT... reader

Go to the Reader Setup > Configure Reader menu option. Configure the reader like I have suggested. Many other options are possible. Press **“Send Setup to Reader”** and look for a 0 on the previous screen. Close the window and reconnect to the reader and verify the configuration.



Configuring an MTT... reader

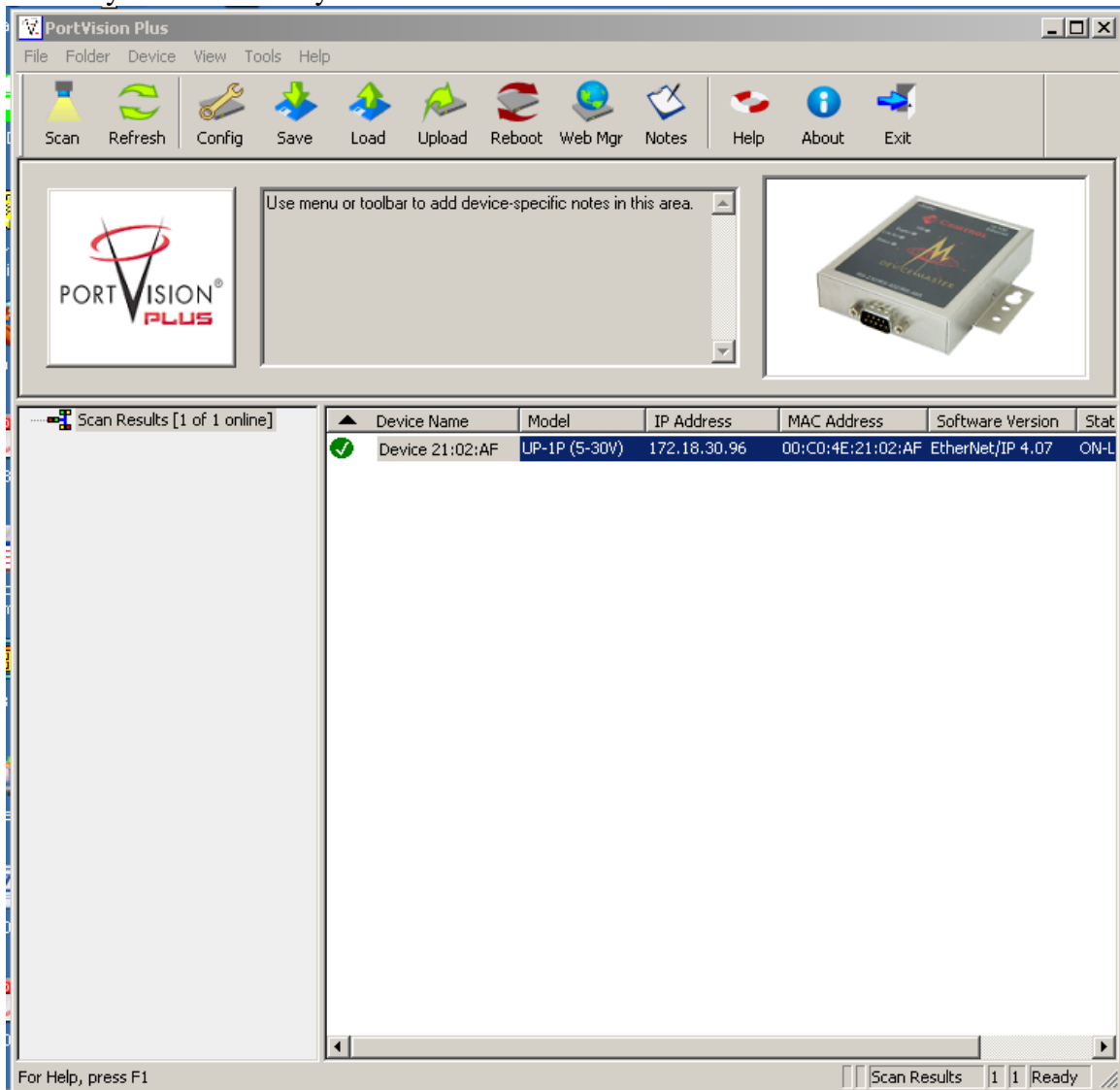
## Configure RTS-UP-... Ethernet/IP adapter

### Load Ethernet/IP firmware

The RTS-UP unit comes with socket server firmware. If you want other firmware for industrial busses like Ethernet/IP, PROFINET, or Modbus/TCP then download this firmware from our web site and send the firmware to the unit using PortVision.

[Download and install Portvision](#)

You may have to reboot your PC to see the RTS unit. Click “Scan”.



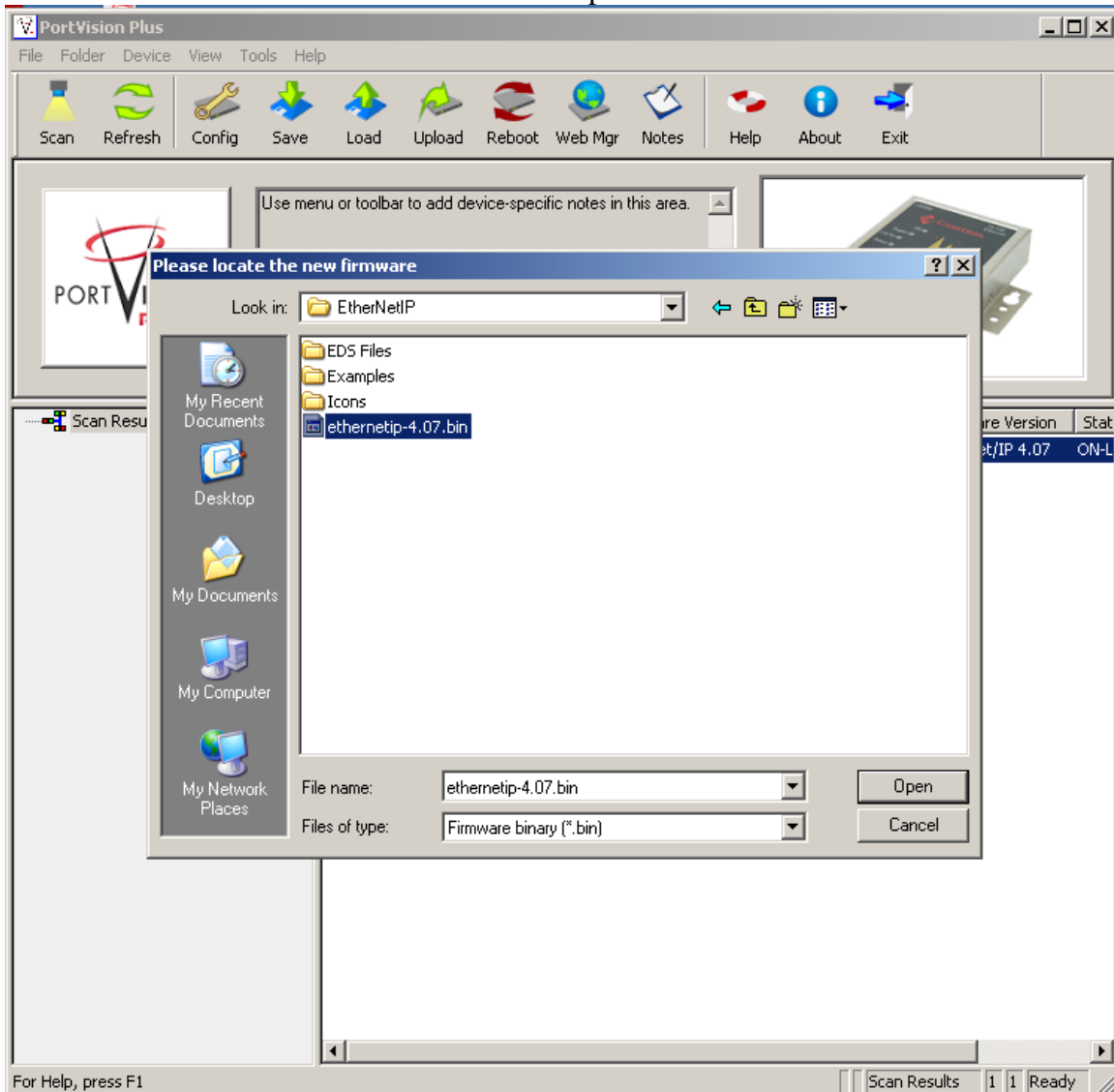
The screenshot shows the PortVision Plus software interface. The window title is "PortVision Plus". The menu bar includes "File", "Folder", "Device", "View", "Tools", and "Help". The toolbar contains icons for "Scan", "Refresh", "Config", "Save", "Load", "Upload", "Reboot", "Web Mgr", "Notes", "Help", "About", and "Exit". The main area is divided into three sections: a logo for "PORTVISION PLUS" on the left, a text area in the center with the instruction "Use menu or toolbar to add device-specific notes in this area.", and a photograph of the RTS-UP device on the right. Below these sections is a table of scan results. The table has columns for "Device Name", "Model", "IP Address", "MAC Address", "Software Version", and "Stat". One device is listed: "Device 21:02:AF" with model "UP-1P (5-30V)", IP address "172.18.30.96", MAC address "00:C0:4E:21:02:AF", software version "EtherNet/IP 4.07", and status "ON-L". The status bar at the bottom indicates "Scan Results 1 1 Ready".

Device Name	Model	IP Address	MAC Address	Software Version	Stat
Device 21:02:AF	UP-1P (5-30V)	172.18.30.96	00:C0:4E:21:02:AF	EtherNet/IP 4.07	ON-L

Scanning for RTS-UP... devices

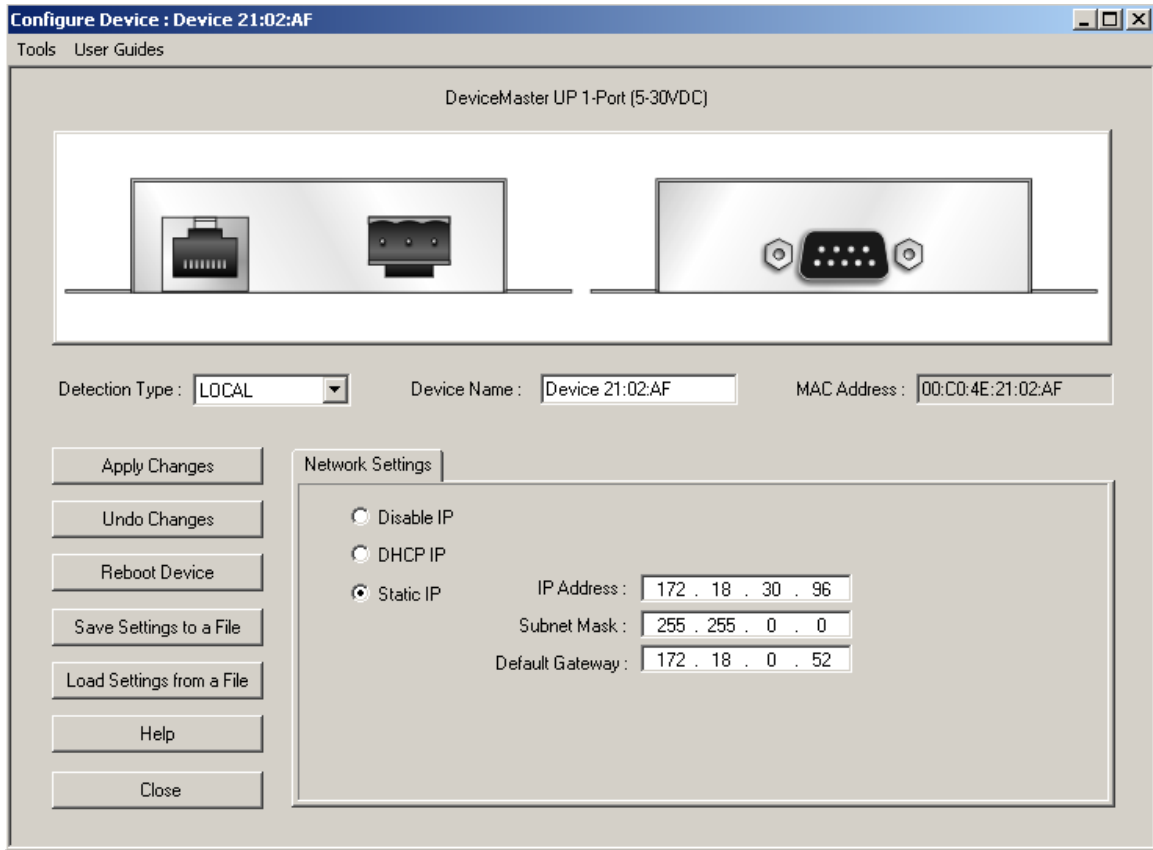
[Download the Ethernet/IP firmware](#)

If the Scan Results do not show a device with Ethernet/IP firmware; then highlight the device and go to the menu “Device > Upload Firmware” and update the RTS unit with the right firmware. When you install the Ethernet/IP firmware above the .bin file will be in the folder Control > Ethernet/IP > Ethernetip-x.xx.bin



Loading the Ethernet/IP firmware into the RTS-UP...

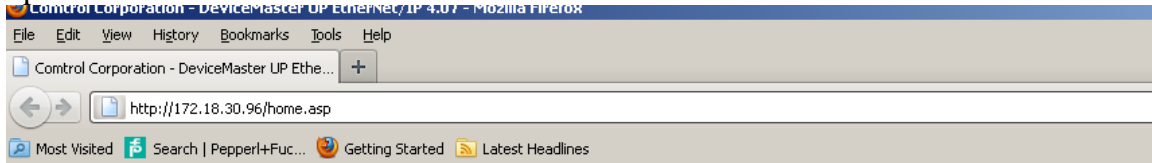
Using PortVision you can also double click on the scanned unit and configure the IP Address, subnet mask, and gateway.



IP address configuration screen for RTS-UP...

## Configure the RTS and MTT to work together

Put the IP address of the RTS-UP... unit in a web browser. You will configure the rts-up...unit here.



### Server Configuration

**Software:** EtherNet/IP 4.07  
**Serial Number:** 9011 - 518  
**IP Config:** Static  
**IP Address:** 172.18.30.96  
**IP Netmask:** 255.255.0.0  
**IP Gateway:** 172.18.0.52

[Serial Device Configuration](#)  
[Ethernet Device Configuration](#)  
[Communication Statistics](#)  
[Display Serial Logs](#)  
[Display Ethernet Device Logs](#)  
[PLC Interface Diagnostics](#)  
[Configure Network](#)

Reboot



Go to Ethernet Device Configuration and open up socket 1. Make the configuration changes you see below. Some settings you will have to customize yourself.

Things you will need to know to make these settings:

PLC IP address

PLC controller slot number(Usually 0)

Control tag variable, SINT array, where the read data will be placed.



## Edit Socket Port 1 Configuration

**Device TCP Connection Configuration**

Enable:

Listen:

Listen Port: 8000

Connect To Mode: Connect-Always

Connect Port: 10000

Connect IP Address: 172.18.30.10

Disconnect Mode: Never

Idle Timer: 0 (msec)

---

**Socket Packet ID Settings**

Rx Timeout Between Packets: 100 (ms)

STX (Start of Transmission) Rx Detect: none Byte 1: Byte 2: (dec)

ETX (End of Transmission) Rx Detect: one byte Byte 1: 13 Byte 2: (dec)

**PLC Specific Settings**

STX (Start of Transmission) Tx Append: none Byte 1: Byte 2: (dec)

ETX (End of Transmission) Tx Append: none Byte 1: Byte 2: (dec)

Strip Rx STX/ETX:

**Application Specific Settings**

STX (Start of Transmission) Tx Append: none Byte 1: Byte 2: (dec)

ETX (End of Transmission) Tx Append: none Byte 1: Byte 2: (dec)

---

**EtherNet/IP Settings**

Rx (To PLC) Ethernet Transfer Method: Write-to-Tag/File

PLC IP Address: 172.18.30.7

PLC Controller Slot Number (ControlLogix Family): 0

Maximum PLC Update Rate (Write-To-Tag/File): 40 (msec)

Maximum Rx Data Packet Size: 500 (bytes)

Oversized Rx Packet Handling: Truncate

Rx (To PLC) Produced Data Tag/File Name: MTTData

Note: File names for SLC/PLC-5 must begin with a "\$" (i.e. \$N10:0).  
 Note: File names for MicroLogix must begin with a "#" (i.e. #N10:0).

Tx Sequence Number Checking:

Disable Non-Filtered To PLC Rx Queue:

(PLC-5/SLC) Rx MS Byte First:

(PLC-5/SLC) Tx MS Byte First:

Connect

Connect always

MTT port 10000

MTT IP address

Disconnect never

Timeout 100ms

Choose one byte suffix, carriage return is 13 = 0D in MTT software

This option writes to PLC memory automatically

PLC IP address

Variable the data will be written to (SINT array)

Port configuration screen for RTS-UP...

Once all settings have been made, choose "Submit" at bottom of page. Wait for unit to reboot.

## ***How will you know it is working?***

Go back to the web configuration screen of the RTS-UP... unit and now go to the PLC Interface Diagnostics screen. Place tag over the MTT...reader. Refresh screen and the “Messages/Responses Sent to PLC” will count up on the PLC Interface Diagnostics screen. Also at the bottom you will see “No Error Detected”

## **PLC Interface Diagnostics**

[Server Configuration Home](#)

[Serial Device Configuration](#)

[Ethernet Device Configuration](#)

[Communication Statistics](#)

[Display Serial Logs](#)

---

<b>EtherNet/IP Interface Statistics</b>	<b>Reset Statistics</b>
<b>Messages/Responses Received From PLC:</b>	3
<b>Broadcasts Received From PLC:</b>	0
<b>Messages/Responses Sent To PLC:</b>	3
<b>Request Messages From PLC:</b>	0
<b>Bad Responses to Msgs Sent To PLC:</b>	0
<b>Invalid Network Path Errors:</b>	0
<b>No Response From PLC Errors:</b>	0
<b>Pending Request Limit Errors:</b>	0
<b>Unexpected Event Errors:</b>	0
<b>Unsupported CIP Request Instance Errors:</b>	0
<b>Unsupported CIP Request Service Errors:</b>	0
<b>Unsupported CIP Request Class Errors:</b>	0
<b>Unsupported CIP Request Attribute Errors:</b>	0
<b>Improper Configuration Errors:</b>	0
<b>Invalid Message Data Errors:</b>	0
<b>System Resource Errors:</b>	0
<b>Oversized Received Data Packet Errors:</b>	0
<b>Writes To Offline Ethernet Device on Socket1:</b>	0
<b>First Error Description:</b>	No Error Detected

In PLC the data will also appear. (If you have trouble make sure the variable is a controller tag and an SINT array large enough to hold all of your data.)

Name	Value	Force Mask	Style	Data Type
MTTData	{...}	{...}	ASCII	SINT[200]
MTTData[0]	'\$1D'		ASCII	SINT
MTTData[1]	'\$00'		ASCII	SINT
MTTData[2]	81		Decimal	SINT
MTTData[3]	'\$00'		ASCII	SINT
MTTData[4]	'\$00'		ASCII	SINT
MTTData[5]	'\$00'		ASCII	SINT
MTTData[6]	'\$00'		ASCII	SINT
MTTData[7]	'\$00'		ASCII	SINT
MTTData[8]	'\$00'		ASCII	SINT
MTTData[9]	'\$00'		ASCII	SINT
MTTData[10]	'\$00'		ASCII	SINT
MTTData[11]	'\$00'		ASCII	SINT
MTTData[12]	'\$00'		ASCII	SINT
MTTData[13]	'\$00'		ASCII	SINT
MTTData[14]	'\$00'		ASCII	SINT
MTTData[15]	'\$00'		ASCII	SINT
MTTData[16]	'\$00'		ASCII	SINT

Two byte Counter increments on every new packet

Two byte Length shows how many MTT bytes to follow

Data

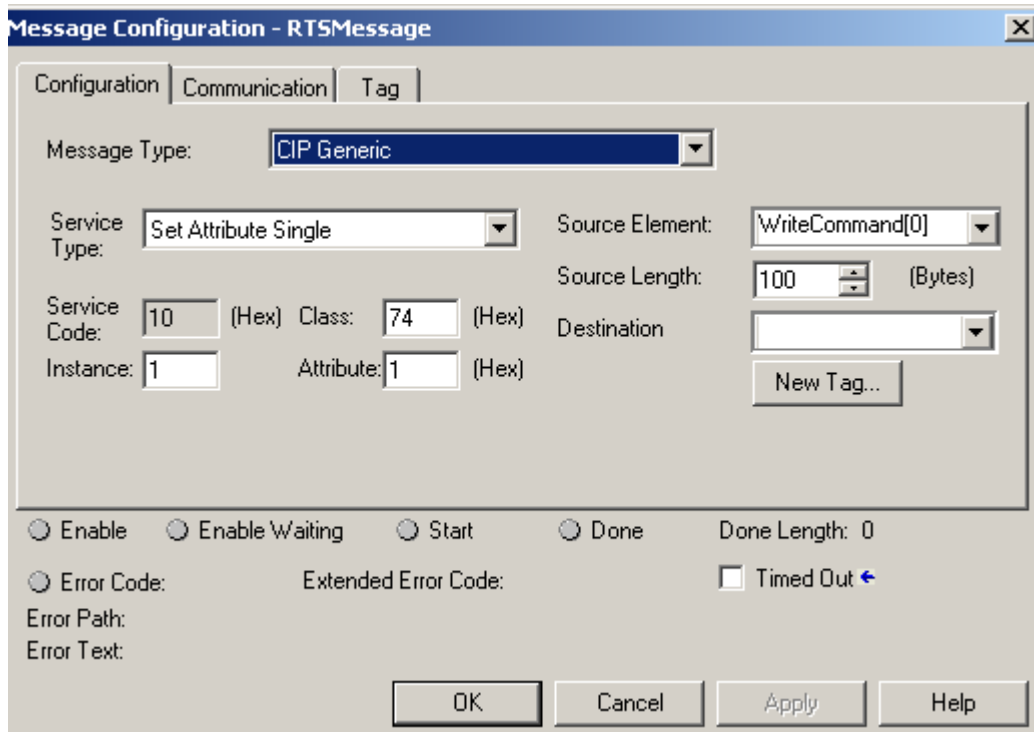
...				
MTTData[67]	'\$00'		ASCII	SINT
MTTData[68]	'\$00'		ASCII	SINT
MTTData[69]	'\$00'		ASCII	SINT
MTTData[70]	'\$00'		ASCII	SINT
MTTData[71]	'\$00'		ASCII	SINT
MTTData[72]	'\$00'		ASCII	SINT
MTTData[73]	'\$00'		ASCII	SINT
MTTData[74]	'\$00'		ASCII	SINT
MTTData[75]	':'		ASCII	SINT
MTTData[76]	'0'		ASCII	SINT
MTTData[77]	'1'		ASCII	SINT
MTTData[78]	'4'		ASCII	SINT
MTTData[79]	'1'		ASCII	SINT
MTTData[80]	'5'		ASCII	SINT
MTTData[81]	'6'		ASCII	SINT
MTTData[82]	'7'		ASCII	SINT
MTTData[83]	'6'		ASCII	SINT
MTTData[84]	'\$r'		ASCII	SINT
MTTData[85]	'\$00'		ASCII	SINT
MTTData[86]	'\$00'		ASCII	SINT

Suffix

The amount of data will depend on the format of the tag and the “Fixed String Length” parameter that was used when you configured the MTT... unit. In this example I used the “Data+Mark” option. This will display all the read/write data on the tag first, then a colon(:) then the 8 byte fixed code or MARK. If an MTO read only tag was used only the :Mark would be shown because there is no read/write data available.

## Writing to tags

Reading the tags is automatic. If you want to send a command to the MTT unit or write data with the MTT6000-F120-B12-V45 unit then a message instruction will be used. Use Class 74h, Instance 1(Four port units have one instance for each port), attribute 1. The length of the string should be longer than you need. The send string in this example is an SINT array with length 100.



Before writing to the tags: verify the following.

How much data do you want to write?

What is the tag format you want to use?(My example uses C0, one tag in field)

The string to send to the RTS... then on to the MTT unit needs to have a specific format. This is an example of the format when you want to write 71 bytes to an MTM tag using format C0.

Bytes 0,1 = any number you like

Bytes 2,3 = Length of bytes to follow. Always 7 more bytes than the amount of data to write to the tag

Byte 4 = write command “w”

Bytes 5,6,7 = Length of data to write up to 071

Up to 019 bytes = Quarter memory

From 020 to 071 bytes is FULL memory

(To write mini memory tags with data from 0 to 16383 use different command)

Byte 8-78 = data

Byte 79-80 = Format

C0 = fastest but cannot read multiple tags in field

R4 = Longer battery life and a couple tags n field

Other options also available

Byte 81 = Carriage return or decimal 13

Name	Value	Force Mask	Style	Data Type
+ WriteCommand[0]	1		Decimal	SINT
+ WriteCommand[1]	0		Decimal	SINT
+ WriteCommand[2]	78		Decimal	SINT
+ WriteCommand[3]	0		Decimal	SINT
+ WriteCommand[4]	'w'		ASCII	SINT
+ WriteCommand[5]	'0'		ASCII	SINT
+ WriteCommand[6]	'7'		ASCII	SINT
+ WriteCommand[7]	'1'		ASCII	SINT
+ WriteCommand[8]	'\$00'		ASCII	SINT
+ WriteCommand[9]	'\$00'		ASCII	SINT
+ WriteCommand[10]	'\$00'		ASCII	SINT
+ WriteCommand[11]	'\$00'		ASCII	SINT
+ WriteCommand[12]	'\$00'		ASCII	SINT
+ WriteCommand[13]	'\$00'		ASCII	SINT
+ WriteCommand[14]	'\$00'		ASCII	SINT
+ WriteCommand[15]	'\$00'		ASCII	SINT
+ WriteCommand[16]	'\$00'		ASCII	SINT
+ WriteCommand[17]	'\$00'		ASCII	SINT
+ WriteCommand[18]	'\$00'		ASCII	SINT
+ WriteCommand[19]	'\$00'		ASCII	SINT
+ WriteCommand[20]	'\$00'		ASCII	SINT
+ WriteCommand[21]	'\$00'		ASCII	SINT
+ WriteCommand[22]	'\$00'		ASCII	SINT
+ WriteCommand[23]	'\$00'		ASCII	SINT
+ WriteCommand[24]	'\$00'		ASCII	SINT
+ WriteCommand[25]	'\$00'		ASCII	SINT
+ WriteCommand[26]	'\$00'		ASCII	SINT
+ WriteCommand[27]	'\$00'		ASCII	SINT
+ WriteCommand[28]	'\$00'		ASCII	SINT
+ WriteCommand[29]	'\$00'		ASCII	SINT
+ WriteCommand[30]	'\$00'		ASCII	SINT
+ WriteCommand[31]	'\$00'		ASCII	SINT
+ WriteCommand[32]	'\$00'		ASCII	SINT

Sequence number

Length to follow, two bytes

Write command

Length, 3 bytes

Data

+ WriteCommand[33]	'\$00'	ASCII	SINT
+ WriteCommand[34]	'\$00'	ASCII	SINT
+ WriteCommand[35]	'\$00'	ASCII	SINT
+ WriteCommand[36]	'\$00'	ASCII	SINT
+ WriteCommand[37]	'\$00'	ASCII	SINT
+ WriteCommand[38]	'\$00'	ASCII	SINT
+ WriteCommand[39]	'\$00'	ASCII	SINT
+ WriteCommand[40]	'\$00'	ASCII	SINT
+ WriteCommand[41]	'\$00'	ASCII	SINT
+ WriteCommand[42]	'\$00'	ASCII	SINT
+ WriteCommand[43]	'\$00'	ASCII	SINT
+ WriteCommand[44]	'\$00'	ASCII	SINT
+ WriteCommand[45]	'\$00'	ASCII	SINT
+ WriteCommand[46]	'\$00'	ASCII	SINT
+ WriteCommand[47]	'\$00'	ASCII	SINT
+ WriteCommand[48]	'\$00'	ASCII	SINT
+ WriteCommand[49]	'\$00'	ASCII	SINT
+ WriteCommand[50]	'\$00'	ASCII	SINT
+ WriteCommand[51]	'\$00'	ASCII	SINT
+ WriteCommand[52]	'\$00'	ASCII	SINT
+ WriteCommand[53]	'\$00'	ASCII	SINT
+ WriteCommand[54]	'\$00'	ASCII	SINT
+ WriteCommand[55]	'\$00'	ASCII	SINT
+ WriteCommand[56]	'\$00'	ASCII	SINT
+ WriteCommand[57]	'\$00'	ASCII	SINT
+ WriteCommand[58]	'\$00'	ASCII	SINT
+ WriteCommand[59]	'\$00'	ASCII	SINT
+ WriteCommand[60]	'\$00'	ASCII	SINT
+ WriteCommand[61]	'\$00'	ASCII	SINT
+ WriteCommand[62]	'\$00'	ASCII	SINT
+ WriteCommand[63]	'\$00'	ASCII	SINT
+ WriteCommand[64]	'\$00'	ASCII	SINT
+ WriteCommand[65]	'\$00'	ASCII	SINT
+ WriteCommand[66]	'\$00'	ASCII	SINT
+ WriteCommand[67]	'\$00'	ASCII	SINT
+ WriteCommand[68]	'\$00'	ASCII	SINT
+ WriteCommand[69]	'\$00'	ASCII	SINT
+ WriteCommand[70]	'\$00'	ASCII	SINT
+ WriteCommand[71]	'\$00'	ASCII	SINT
+ WriteCommand[72]	'\$00'	ASCII	SINT
+ WriteCommand[73]	'\$00'	ASCII	SINT
+ WriteCommand[74]	'\$00'	ASCII	SINT
+ WriteCommand[75]	'\$00'	ASCII	SINT
+ WriteCommand[76]	'\$00'	ASCII	SINT
+ WriteCommand[77]	'\$00'	ASCII	SINT
+ WriteCommand[78]	'\$00'	ASCII	SINT
+ WriteCommand[79]	'C'	ASCII	SINT
+ WriteCommand[80]	'0'	ASCII	SINT
+ WriteCommand[81]	'\$r'	ASCII	SINT

Tag Format

Carriage return

If you want to format the tag to mini memory because you only need data from 0 up to 16383 and you want the data returned as fast as possible use the command `w5xxxxx<CR>` where the length is 5 and the data would be 00000 up to 16383

It may take up to 15s to write to a tag. If after 15seconds you don't see your data returned in the PLC you can retry.