O & M Manual

MetriNet
Ethernet/IP Communications Manual

Home Office

Analytical Technology, Inc.
6 Iron Bridge Drive
Collegeville, PA 19426
Phone: 800-959-0299
610-917-0991
Fax: 610-917-0992
Email: sales@analyticaltechnology.com

European Office

ATI (UK) Limited
Unit 1 & 2 Gatehead Business Park
Delph New Road, Delph
Saddleworth OL3 5DE
Phone: +44 (0)1457-873-318
Fax: + 44 (0)1457-874-468
Email: sales@atiuk.com

PRODUCT WARRANTY

Analytical Technology, Inc. (Manufacturer) warrants to the Customer that if any part(s) of the Manufacturer's equipment proves to be defective in materials or workmanship within the earlier of 18 months of the date of shipment or 12 months of the date of start-up, such defective parts will be repaired or replaced free of charge. Inspection and repairs to products thought to be defective within the warranty period will be completed at the Manufacturer's facilities in Collegeville, PA. Products on which warranty repairs are required shall be shipped freight prepaid to the Manufacturer. The product(s) will be returned freight prepaid and allowed if it is determined by the manufacturer that the part(s) failed due to defective materials or workmanship.

This warranty does not cover consumable items, batteries, or wear items subject to periodic replacement including lamps and fuses.

Gas sensors carry a 12 months from date of shipment warranty and are subject to inspection for evidence of misuse, abuse, alteration, improper storage, or extended exposure to excessive gas concentrations. Should inspection indicate that sensors have failed due to any of the above, the warranty shall not apply.

The Manufacturer assumes no liability for consequential damages of any kind, and the buyer by acceptance of this equipment will assume all liability for the consequences of its use or misuse by the Customer, his employees, or others. A defect within the meaning of this warranty is any part of any piece of a Manufacturer's product which shall, when such part is capable of being renewed, repaired, or replaced, operate to condemn such piece of equipment.

This warranty is in lieu of all other warranties (including without limiting the generality of the foregoing warranties of merchantability and fitness for a particular purpose), guarantees, obligations or liabilities expressed or implied by the Manufacturer or its representatives and by statute or rule of law.

This warranty is void if the Manufacturer's product(s) has been subject to misuse or abuse, or has not been operated or stored in accordance with instructions, or if the serial number has been removed.

Analytical Technology, Inc. makes no other warranty expressed or implied except as stated above.

Table of Contents

Table of Figures

Ethernet/IP Option

1.1 General

MetriNet Monitors are available with three digital communication options: Cellular, Modbus-RTU, Modbus-TCP/IP, Ethernet/IP. This manual applies only to instruments supplied with the Ethernet/IP communication option. It is critical to note that Ethernet/IP devices only communicate with other systems running the “Industrial Protocol” of Ethernet. You cannot plug a MetriNet into your office Ethernet and expect to have the instrument talk to your desktop computer.

The discussion of standard Ethernet and Ethernet/IP is vast, and well beyond the ability to discuss here in great detail. The documentation for this option assumes working network knowledge by the user.

1.2 Ethernet/IP Communication

Ethernet/IP was first introduced in 2001, and is a highly developed and complete protocol for an industrial Ethernet network solution. Intended for automation applications, EtherNet/IP is a member of a family of networks that implement the Common Industrial Protocol (CIP) at its upper layers of the standard ISO/OSI 7-layer model. CIP defines the object structure, specifies the message transfer, provides users with a unified communication architecture, and encompasses a comprehensive suite of services for a variety of manufacturing automation applications which include data, control, safety, and configuration.

In the CIP Protocol, every network device represents itself as a series of objects. Each object is simply a grouping of the related data values in a device. The identity object contains related identity data values called attributes. Attributes for the identity object include the vendor ID, date of manufacture, device serial number and other identity data.

1.3 Ethernet/IP Transmission Details

It should be understood that Ethernet/IP is simply an application layer protocol that is transferred over an Ethernet hardware link. The word "Ethernet" simply refers to the common physical cable, perhaps running to an office PC.

In the OSI model, “Ethernet” is the lower part of the model, the physical transfer method or the hardware. It says nothing about the way information is transferred, which is specified near the top of the OSI model. In the common office network, many different standard communication protocols are operating during normal office use, like IP, TCP, etc. None of these are designed to handle the CIP industrial format, so that interface must be handled by a specific program that recognizes the format. Because of this, an Ethernet/IP device cannot be directly connected to your office network for transferring information.

The data for the protocol is constructed into a specific structure inside a standard TCP/IP Packet. A user application program simply decodes the structure inside the received TCP or UDP packet. The data structure consists of a grouping of data values called Attributes inside other sets of containment data called Objects. There are Ethernet/IP required Objects and application Objects. Required objects are typically - Identity, TCP, and Router. Application objects consist of some defined data types, like a Motor Object, and some user specific data types.

1.4 Cable Connection

The cable used for Ethernet/IP communication should meet the CAT5 standard defined by the Electronic Industries Association and Telecommunications Industry Association. It is readily available in lengths up to 100 ft. (30 m) with plugs on each end.

To install an Ethernet cable in the MetriNet, pass the unterminated cable through the cable gland nearest the location of the RJ45 connector on the Ethernet/IP option board. Provide a slight service loop amount of slack so that the terminated cable will not be pulling directly on the RJ45 connector once installed. Termination of Cat5/5e/6 cables is very easy and can be completed quickly with the commonly available RJ45 crimp tool.

Note that there a reconnection LEDs present on the face of the RJ45 connector, which can be used to verify activity on the interface. Rather than detail their specific use, it is sufficient to just very flashing on these LEDS when the unit is powered up or actively communicating with the network client.

Figure 1 – Ethernet Cable Routing on MetriNet

Figure 2 - RJ45 Crimp Tool

While professional results are optimized with the use of the termination tool and custom cable lengths, patch cord style connection can be completed in some cases by passing the finished RJ45 connectors through the enclosure cable gland. The connector will fit through the plastic part of the cable gland, and the rubber grommet can be slit (some RJ45s may be too big for this.) Once the connector is inside the enclosure, simply plug it into the jack provided on the Ethernet/IP communication board. Be sure to adjust the rubber insert in the cable gland so that the slit is on the bottom and then tighten the gland to seal around the wire.

1.5 Configuring MetriNet for Ethernet Option

Once the proper power is applied to the MetriNet monitor, select **“Ethr**” in the **^Host Comms** listing of the **OPTIONS** menu to enable Ethernet/IP functionality.

1.6 Configuring fixed IP with BOOTP

The next step in the use of the MetriNet is to sort out how an IP network address will be assigned to the instrument. The MetriNet acquires a dynamic network IP address from the network client/master through the BOOTP service, which is enabled by default for ODVA EIP-CIP conformance.

BOOTP is a very simple software server application that waits for a specific MAC address to appear on the network, and then assigns it a pre-determined IP address that has been set-up by the user. In the most actual applications, BOOTP runs right on the PLC alongside tools like Rockwell’s Logix/Studio 5000. For manual fixed settings, this same BOOTP server tool can be easily run on a PC. The current Rockwell version of this tool (recommended) is –
“**BootP-DHCP Ethernet/IP Commissioning Tool Version 3.02.00**”

This tool is available from the Rockwell Automation website for free. For optimum security, only download this tool directly from Rockwell Automation.

While this tool is normally used for dynamic assignment, the MetriNet will remember the last successful assignment made by BOOTP, and it will retain that assignment after cycling instrument power. Therefore, the tool can be used to lock in a fixed IP address assignment, and then the BOOTP service will not actually be required after that first assignment.

In order to use BOOTP properly on a PC, some careful NIC adjustment settings must be made to the PC Ethernet port prior to running the BOOTP service. The PC can only reach IP addresses at the predetermined IP/subnet level, so we must force the PC to a subnet that can reach the existing MetriNet.

1) Turn off MetriNet. Connect MetriNet to laptop with direct cable. Set laptop NIC port on a locked IP and subnet to stop DHCP service. In IPV4 properties on your PC’s Ethernet connection port, select "Use the following IP Address" and set the PC to fixed IP of -
   IP Address = 192.168.0.10
   Subnet Mask = 255.255.255.0

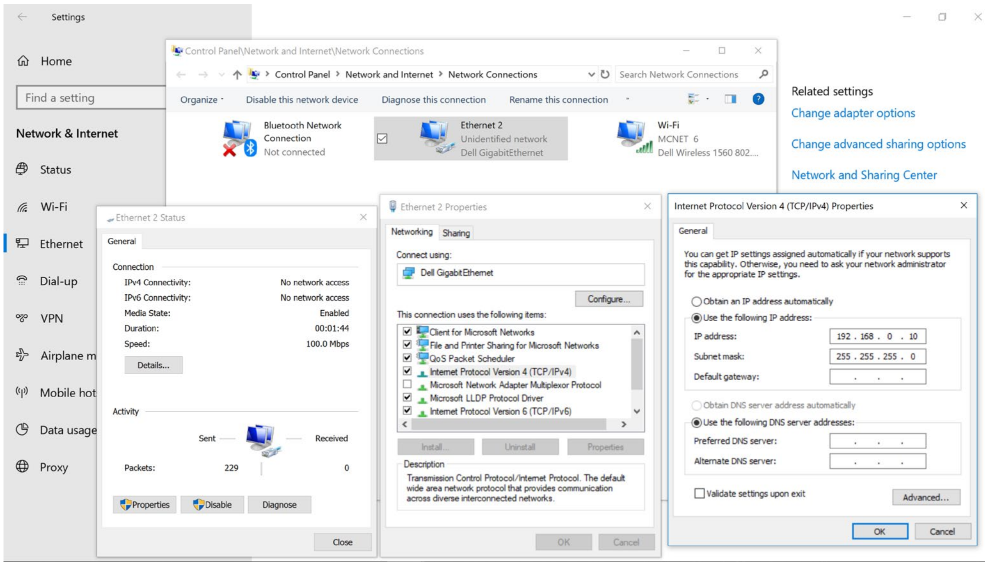

Figure 3 – Force PC to Specific Network Subnet Prior to Running BOOTP

2) Once this IP/subnet is set, click OK to save and then close it all out. Turn PC off then back on to place it on the new IP address set in step 1. The PC will now be able to access any network device on 192.168.0.X, as the subnet mask 255.255.255 locks in the first three variables. The default value for the MetriNet is 192.168.0.254, so it can now be reached by the PC.

3) Find the Rockwell BOOTP PC application and launch it. Select your Ethernet Adapter as the network interface.

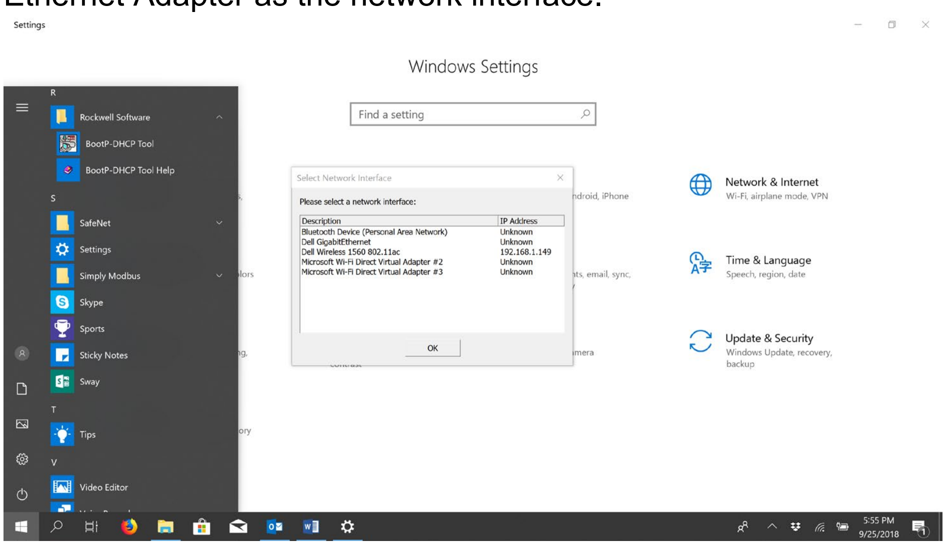

Figure 4 – Launching BOOTP server

4) On next screen, you must create an assignment relationship to match a known MAC ID with a desired IP address. The MAC ID for your MetriNet is written on a label inside the enclosure. For this example, we will use the MAC ID - **00:80:A3:9B:BE:B6** and assign it an IP of **192.168.0.30**. Hit OK after this information is filled in, and you will see this entry appear in the bottom half of the window under "Entered Relations."

Once the MAC assignment has been entered, it is ready for a BOOTP IP request to come in for assignment. Requests for assignment will appear in the upper window as they come in.

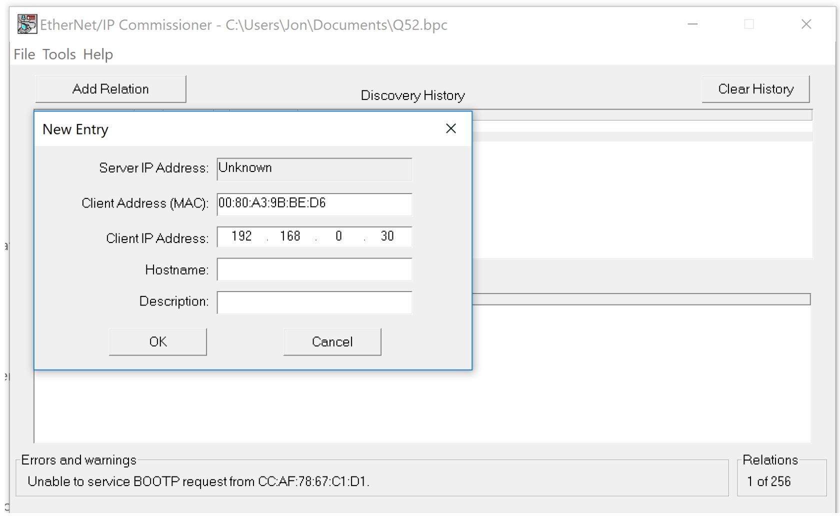

Figure 5 – Setting IP/mask Assignment in BOOTP Server

5) Now, plug the cable from MetriNet into the PC Ethernet port, turn on power to the MetriNet, and wait for BOOTP to detect the request from the MetriNet for an IP address.

The upper "Discovery History Window" will show new IP requests coming in on the Ethernet port, and when the specific MAC ID in our relation shows up, the IP will be automatically assigned per our relation entry and sent to the MetriNet. Note that you may see requests from other devices as they come in, but only the specific MAC ID from the assignment will be acknowledged.

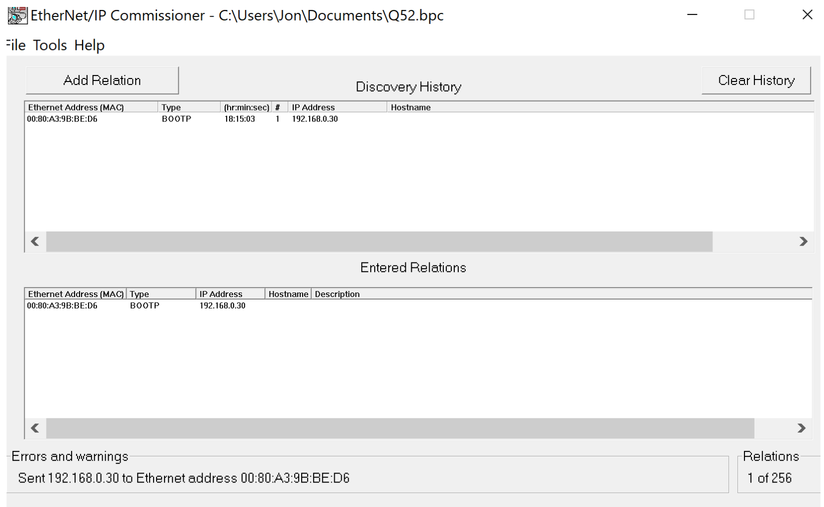

Figure 6 – Successful Detection of MAC ID and IP Address Assignment in BOOTP Server

Once the assignment has been made and the IP has been sent to the MetriNet, disconnect power to the MetriNet and that IP will be stored in the instrument. The MetriNet stores the very last successful IP address assignment. Then close out all BOOTP server windows.

6) Now, to check that the IP address has been properly saved, apply power to MetriNet again and wait one 1 minute before connecting it to the PC port. Do not launch BOOTP server this time, as you don't need it. During the 1 minute wait, MetriNet will search for a BOOTP server on power up and won't find the service because it is not used here now, and the MetriNet is also not connected to the PC. It will then revert to the stored default – the last successful BOOTP assignment. Connect cable from MetriNet to the PC after that one minute. You can now check the IP connection to the MetriNet at that address by entering a ping command at the Windows OS command prompt, for example **">ping 192.168.0.30."** You should see a response as shown below. In addition, you can try an **">arp -a"** command to show all current connections and associated MAC addresses. You should see the new IP address of 192.168.0.30 showing up.

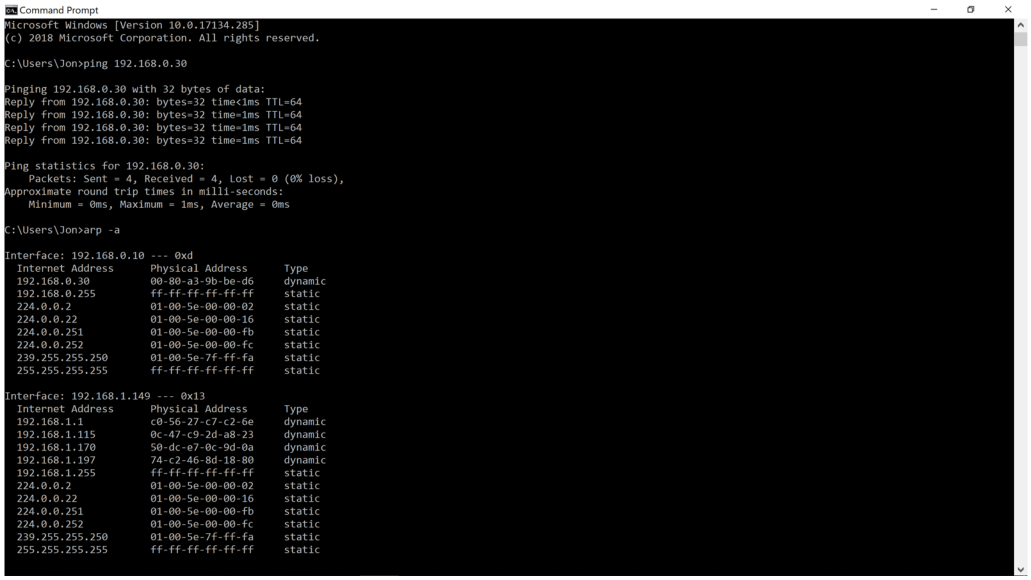

Figure 7 – PING and ARP Command Tests for IP Address Setting.

Note that the “arp -a” listing still shows this IP as a “dynamic” assignment, as this is the fall-back IP that has been stored in the MetriNet. On power up, the MetriNet will still try and find a BOOTP server for 30 seconds, and then default to this stored value as the fallback. If you do not see your new connection, two troubleshooting notes –

1-You must always be careful to place the new IP address on the proper subnet level so the PC can still reach it with the settings made in step 1. If the PC is on 192.168.0.X, then any assigned IP must be on the X level – and don’t place the MetriNet on the same IP as the PC. During BOOTP server assignment, if you place the MetriNet on a vastly different IP like 10.20.120.100, you must repeat step 1 before step 6 above to place the PC at something like 10.20.120.10. This can get tricky, as you can make assignments anywhere, but then the PC must be re-established on that new net to find the new assignment.

2-Remember to wait a minute before the ping test if you verify a connection. The MetriNet runs BOOTP by default, so it will be searching for that server for about 30 seconds on power up, and then will default to the stored IP.

1.7 Loading EDS File

The EDS file will specify many of the parameters for reading and writing the proper registers in the MetriNet. It must first be loaded into server in control of the network. This can vary widely depending on the tools, but here is an example using Rockwell's library "EDS Hardware Installation Tool."

1-Launch the EDS tool and select "Add" to start the EDS Wizard for file addition.

2-Select button for "Register a single file," and browse to the file location for the EDS file "ATI Q52x Transmitter EDS File.eds"

3-Follow though the screens to complete the addition, then exit the tool.

Now, the EDS file that is specific to the MetriNet is part of the Rockwell EDS RSLinx library, and devices can be recognized when connected to the network.

1.8 PLC Object Generic Settings

PLCs handle this Ethernet/IP exchange in a wide variety of configurations, and the basic configuration of a class 1 connection in every PLCs software is different, but there should be a place to enter the following basic information.

Data Type: "INT" or "16-Bit"
Input T->O Assembly Instance: 0x65 or 101 decimal
Input T->O Size: 250 words (16-bit) or 500 bytes (8-bit)
Output O->T Assembly Instance: 0x66 or 102 decimal
Output O->T Size: 5 word (16-bit) or 10 bytes (8-bit)
Configuration Assembly Instance: 0x80 or 128 decimal
Configuration Size: 0

RPI Setting = 1000 mS or higher

This description represents the raw block of data in and out of the MetriNet, and will take everything in 16-bit words (so you must parse out the 32-bit values shown earlier.) The data typing and setting up of variables must all be done manually, based on the values shown below in section 1.9. For more basic tools, like Molex-EIP, that there may be an endian byte-swap occurring on each set of registers, so you may have to byte swap each 16-bit register to recombine the data.

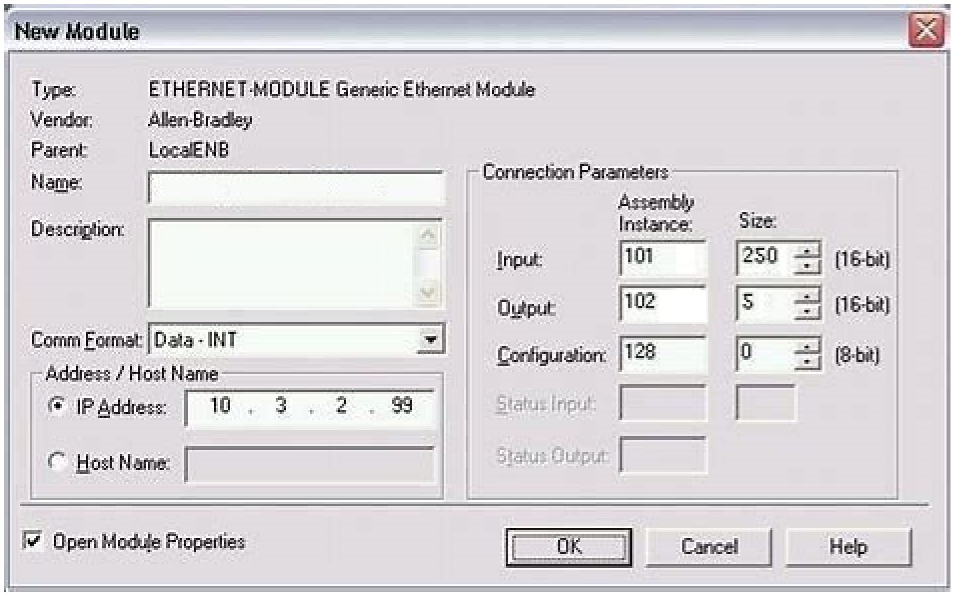

Figure 8 – Generic Module Input/Output Settings

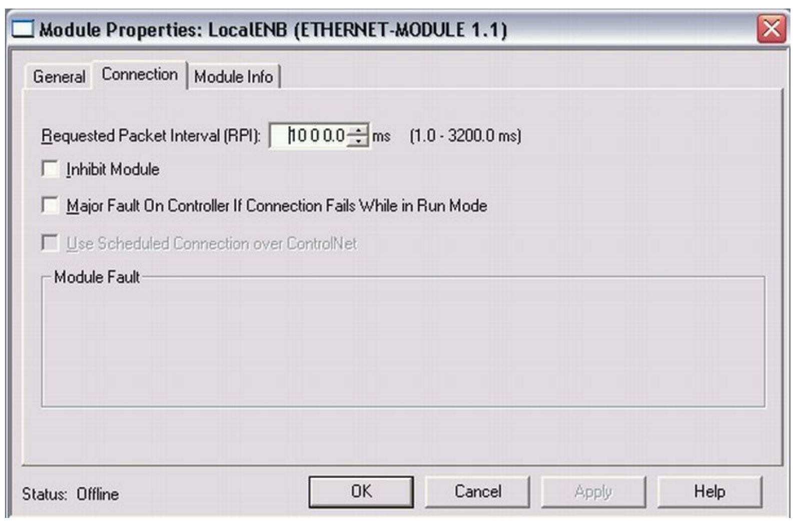

Figure 9 – Connection/RPI Settings

1.9 PLC Object Settings Parameter Detail, and EDS Files

The following table describes all of the data types used

UINT	Unsigned Integer (16-bit)
UDINT	Unsigned Double Integer (32-bit)
INT	Signed Integer (16-bit)
DINT	Signed Integer (32-bit)

BYTE	Bit String (8-bits)
WORD	Bit String (16-bits)
DWORD	Bit String (32-bits)
LWORD	Bit String (64-bits)

ODVA CIP conforming objects implemented include –
0x01 Identity Object
0x04 Assembly Object, Input
0x04 Assembly Object, Output
0xF5 TCP Object
0xF6 Ethernet Link Object
*These are all broken down below in sections below.

Identity Object (01_{HEX} – 1 Instance)

The following tables contain the attribute, status, and common services information for the Identity Object. The Identity Object lists specific information on the actual product and the vendor.

Class Attributes (Instance 0)

Attribute ID	Name	Data Type	Data Value	Access Rule
1	Revision	UINT	1	Get

Identity Instance Attributes (Instance 1)

Attribute ID	Name	Data Type	Data Value	Access Rule
1	Vendor Number	UINT	1530	Get
2	Device Type	UINT	$2B_{HEX}$	Get
3	Product Code Number	UINT	101	Get
4	Major Revision Minor Revision	USINT USINT	01 01	Get
5	Status	WORD	See Below	Get
6	Serial Number	UDINT	NU	Get
7	Product Name	SHORT STRING32	ATI Q52x	Get

Identity Instance Common Services

Service Code	Implemented for		Service Name
	Class Level	Instance Level	
05_{HEX}	No	Yes	Reset
$0E_{HEX}$	Yes	Yes	Get_Attribute_Single

Input Assembly Object (04HEX – 1 Instance)

The Input Assembly Object is a collection of all the measurement information that can be read from the MetriNet. This includes all data from any attached sensor(s.) This is "read only" data that the MetriNet can send out. The data arrangement is detailed below, including the byte size of various data fields.

Class Attributes (Instance 0)

Attribute ID	Name	Data Type	Data Value	Access Rule
1	Revision	UINT	2	Get
2	Max Instance	UINT	0x66 (102 decimal)	Get

Input Instance Attributes (Instance 101)

Attribute ID	Name	Data Type	Default Data Value	Access Rule
3	Input Data	VARIES	0	Get

Input Instance Common Services

Service Code	Implemented for		Service Name
	Class Level	Instance Level	
$0E_{HEX}$	Yes	Yes	Get_Attribute_Single

Input Instance 101 – 500 Bytes

System IINFO

Bytes	Data Type	Description
0–1	WORD	Interface Status Flags[1]
2–3	WORD	System Status 1 Flags[2]
4–5	WORD	System Status 2 Flags[2]
6–7	UINT	Number of Sensors
8–9	UINT	IP Address High
10–11	UINT	IP Address Low
12–13	UINT	Subnet Mask High
14–15	UINT	Subnet Mask Low
16–17	UINT	Gateway High
18–19	UINT	Gateway Low

Notes on System INFO flag data –

[1,2,3]Flag status as follows for each bit.

Bit	Interface Status Flags	System Status 1 Flags	System Status 2 Flags
0 (LSB)	MB Bridge Status	Sensor 1 Comm Err	Flow Solenoid, 0=close,1=flow
1	MB Bridge Timeout	Sensor 2 Comm Err	Digital Input, 0=OFF,1=ON
2	MB Bridge Fail	Sensor 3 Comm Err	Digital Output, 0=OFF,1=ON
3	NU	Sensor 4 Comm Err	
4	NU	Sensor 5 Comm Err	
5	NU	Sensor 6 Comm Err	
6	NU	Sensor 7 Comm Err	
7	NU	Sensor 8 Comm Err	
8	Active	NU	
9	New Config	NU	

10	NU	NU
11	NU	NU
12	NU	NU
13	NU	NU
14	NU	NU
15	NU	NU

Sensor 1 MEASURE

Bytes	Data Type	Description
20-23	DINT	S1 Main Value
24-27	DINT	S1 Units
28-31	DINT	S1 Raw Value
32-35	DINT	S1 Temperature
36-37	INT	S1 Output Value
38-39	WORD	S1 Status 1
40-41	WORD	S1 Status 2
42-43	INT	S1 ID

Sensor 2 MEASURE

Bytes	Data Type	Description
44-47	DINT	S2 Main Value
48-51	DINT	S2 Units
52–55	DINT	S2 Raw Value
56–59	DINT	S2 Temperature
60-61	INT	S2 Output Value
62-63	WORD	S2 Status 1
64-65	WORD	S2 Status 2
66-67	INT	S2 ID

Sensor 3 MEASURE

Bytes	Data Type	Description
68-71	DINT	S3 Main Value
72-75	DINT	S3 Units
76–79	DINT	S3 Raw Value
80–83	DINT	S3 Temperature
84-85	INT	S3 Output Value
86-87	WORD	S3 Status 1
88-89	WORD	S3 Status 2
90-91	INT	S3 ID

Sensor 4 MEASURE

Bytes	Data Type	Description
92-95	DINT	S4 Main Value
96-99	DINT	S4 Units
100–103	DINT	S4 Raw Value
104–107	DINT	S4 Temperature
108-109	INT	S4 Output Value
110-111	WORD	S4 Status 1
112-113	WORD	S4 Status 2
114-115	INT	S4 ID

Sensor 5 MEASURE

Bytes	Data Type	Description
116-119	DINT	S5 Main Value
120-123	DINT	S5 Units
124–127	DINT	S5 Raw Value
128–131	DINT	S5 Temperature
132-133	INT	S5 Output Value
134-135	WORD	S5 Status 1
136-137	WORD	S5 Status 2
138-139	INT	S5 ID

Sensor 6 MEASURE

Bytes	Data Type	Description
140-143	DINT	S6 Main Value
144-147	DINT	S6 Units
148–151	DINT	S6 Raw Value
152–155	DINT	S6 Temperature
156-157	INT	S6 Output Value
158-159	WORD	S6 Status 1
160-161	WORD	S6 Status 2
162-163	INT	S6 ID

Sensor 7 MEASURE

Bytes	Data Type	Description
164-167	DINT	S7 Main Value
168-171	DINT	S7 Units
172–175	DINT	S7 Raw Value
176–179	DINT	S7 Temperature
180-181	INT	S7 Output Value
182-183	WORD	S7 Status 1
184-185	WORD	S7 Status 2
186-187	INT	S7 ID

Sensor 8 MEASURE

Bytes	Data Type	Description
188-191	DINT	S8 Main Value[1]
192-195	DINT	S8 Units[2]
196–199	DINT	S8 Raw Value[1]
200–203	DINT	S8 Temperature[1]
204-205	INT	S8 Output Value[3]
206-207	WORD	S8 Status 1[5]
208-209	WORD	S8 Status 2[5]
210-211	INT	S8 ID[4]

Notes on MEASURE data –

[1]Main Value, Raw Value, are Temperature are all 32 bit integers, displayed with an implied fixed-decimal point 0.000 resolution.

So, 14.00 pH will show as 14000, which means 14.000 pH. 2000 uS conductivity will show as 2000000, meaning 2000.000 uS. 25C will show as 25000 meaning 25.000.

[2]Units are stored as packed ASCII hi/lo bytes.

[3] Output Value is also displayed with implied 0.000 resolution. The MetriNet has no actual hardware level output for these voltages. So, these are virtual values, like the alarm setpoints, that could be used to feed some other process digitally.

[4] ID not implemented at this time.

[5] Status Flags are as follows -

Bit	Status 1	Status 2
0 (LSB)	ALARM_A	EE_INIT_FAIL
1	ALARM_B	MAIN_UNITS_HI
2	ALARM_C	MAIN_UNITS_LO
3	ALARM_D	MAIN_INPUT_ERR
4	ALARM_E	TC_UNITS_HI
5	ENTRY_OUT_OF_RANGE	TC_UNITS_LO
6	ENTRY_ACCEPTED	TC_INPUT_ERR
7	ENTRY_FAIL	CAL_MAIN_SLOPE_HI
8	MAIN_CAL_PASS	CAL_MAIN_SLOPE_LO
9	MAIN_CAL_FAIL	CAL_MAIN_ZERO_HI
10	TC_CAL_PASS	CAL_MAIN_OFFSET_HI
11	TC_CAL_FAIL	MAIN_UNSTABLE
12	TC_F	CAL_TC_OFFSET_HI
13	SENSOR_LOCK	TC_UNSTABLE
14	NU	NU
15	NU	NU

Sensor 1 INFO

Bytes	Data Type	Description
212-213	INT	S1 Slope
214-215	INT	S1 Offset
216–217	INT	S1 Delay
218–219	INT	S1 Alarm A
220-221	INT	S1 Alarm B
222-223	INT	S1 Slope Alarm
224-225	INT	S1 Timer Limit
226-227	INT	S1 Vout HI
228-229	INT	S1 Vout LO
230-231	INT	S1 TC Mode
232-239	LWORD	S1 Tag1
240-247	LWORD	S1 Tag2

Sensor 2 INFO

Bytes	Data Type	Description
248-249	INT	S2 Slope
250-251	INT	S2 Offset
252–253	INT	S2 Delay
254–255	INT	S2 Alarm A
256-257	INT	S2 Alarm B
258-259	INT	S2 Slope Alarm
260-261	INT	S2 Timer Limit
262-263	INT	S2 Vout HI
264-265	INT	S2 Vout LO
266-267	INT	S2 TC Mode
268-275	LWORD	S2 Tag1
276-283	LWORD	S2 Tag2

Sensor 3 INFO

Bytes	Data Type	Description
284-285	INT	S3 Slope
286-287	INT	S3 Offset
288-289	INT	S3 Delay
290-291	INT	S3 Alarm A
292-293	INT	S3 Alarm B
294-295	INT	S3 Slope Alarm
296-297	INT	S3 Timer Limit
298-299	INT	S3 Vout HI
300-301	INT	S3 Vout LO
302-303	INT	S3 TC Mode
304-311	LWORD	S3 Tag1
312-319	LWORD	S3 Tag2

Sensor 4 INFO

Bytes	Data Type	Description
320-321	INT	S4 Slope
322-323	INT	S4 Offset
324-325	INT	S4 Delay
326-327	INT	S4 Alarm A
328-329	INT	S4 Alarm B
330-331	INT	S4 Slope Alarm
332-333	INT	S4 Timer Limit
334-335	INT	S4 Vout HI
336-337	INT	S4 Vout LO
338-339	INT	S4 TC Mode
340-347	LWORD	S4 Tag1
348-355	LWORD	S4 Tag2

Sensor 5 INFO

Bytes	Data Type	Description
356-357	INT	S5 Slope
358-359	INT	S5 Offset
360-361	INT	S5 Delay
362-363	INT	S5 Alarm A
364-365	INT	S5 Alarm B
366-367	INT	S5 Slope Alarm
368-369	INT	S5 Timer Limit
370-371	INT	S5 Vout HI
372-373	INT	S5 Vout LO
374-375	INT	S5 TC Mode
376-383	LWORD	S5 Tag1
384-391	LWORD	S5 Tag2

Sensor 6 INFO

Bytes	Data Type	Description
392-393	INT	S6 Slope
394-395	INT	S6 Offset
396-397	INT	S6 Delay
398-399	INT	S6 Alarm A
400-401	INT	S6 Alarm B
402-403	INT	S6 Slope Alarm
404-405	INT	S6 Timer Limit
406-407	INT	S6 Vout HI
408-409	INT	S6 Vout LO
410-411	INT	S6 TC Mode
412-419	LWORD	S6 Tag1
420-427	LWORD	S6 Tag2

Sensor 7 INFO

Bytes	Data Type	Description
428-429	INT	S7 Slope
430-431	INT	S7 Offset
432-433	INT	S7 Delay
434-435	INT	S7 Alarm A
436-437	INT	S7Alarm B
438-439	INT	S7 Slope Alarm
440-441	INT	S7 Timer Limit
442-443	INT	S7 Vout HI
444-445	INT	S7 Vout LO
446-447	INT	S7 TC Mode
448-455	LWORD	S7 Tag1
456-463	LWORD	S7 Tag2

Sensor 8 INFO

Bytes	Data Type	Description
464-465	INT	S8 Slope
466-467	INT	S8 Offset
468-469	INT	S8 Delay
470-471	INT	S8 Alarm A
472-473	INT	S8Alarm B
474-475	INT	S8 Slope Alarm
476-477	INT	S8 Timer Limit
478-479	INT	S8 Vout HI
480-481	INT	S8 Vout LO
482-483	INT	S8 TC Mode
484-491	LWORD	S8 Tag1
492-499	LWORD	S8 Tag2

Notes on INFO data –

Format for INFO data is as follows (assumes Q32P pH sensor) -

Value	Field	Meaning
1000	Slope	100.0%
50	Offset	5.0 mVDC
10	Delay	1.0 minutes
700	Alarm A	7.00 pH setpoint
800	Alarm B	8.00 pH setpoint
800	Slope Alarm	80.0% low alarm
1000	Timer Limit	?
1400	Vout HI	14.00 pH = 2.5 VDC (virtual)
0	Vout LO	0.00 pH = 0 VDC (virtual)
0	TC Mode	?
10	Tag1	
11	Tag2	

Output Assembly Object (04HEX – 1 Instance)

The Output Assembly Object is a collection of all the information that can be written to the MetriNet. As the MetriNet has a fully interactive user interface with an LCD and keypad, it is expected that the vast majority of changes made to the sensors and system will be performed by trained technicians at that user interface on the actual MetriNet. While making changes digitally at the network level are possible, these "blind" calibration attempts over the network are not recommended. Network changes to the system ignore the typical sensor visual inspection process that occurs during typical calibration or adjustment. For example, sensors should always be removed, inspected, and then cleaned before calibration in solutions. This is pretty standard GMP procedure.

Even so, a secure password-protected write-window is provided to allow the user to change the majority of the parameters that are seen in the Input Assembly Object. To enter data to change these values, all values are entered through a 5-register window with specific codes to tell the system which sensor the operation is targeting So, this is the "write only" data window. The data arrangement is detailed below, including byte size of various fields. Using the EDS file for the MetriNet, all of the data information can be immediately setup automatically in the network master/client.

Class Attributes (Instance 0)

Attribute ID	Name	Data Type	Data Value	Access Rule
1	Revision	UINT	2	Get
2	Max Instance	UINT	0x66 (102 decimal)	Get

Output Instance Attributes (Instance 102)

Attribute ID	Name	Data Type	Default Data Value	Access Rule
3	Output Data	UINT	0	Get/Set

Output Instance Common Services

Service Code	Implemented for		Service Name
	Class Level	Instance Level	
$0E_{HEX}$	Yes	Yes	Get_Attribute_Single
10_{HEX}	No	Yes	Set_Attribute_Single

Output Instance 102 – 10 Bytes

Bytes	Data Type	Description
0-1	UNIT	Run/Idle Code (1)
2–3	UINT	Sensor number 1-8
4–5	UINT	Function Code
6–7	UINT	Data Entry
8–9	UINT	Lock Code

Output (Write to MetriNet) Example Using Molex-EIP Tool –
To change MetriNet delay setting in S1 to 2.0, send the following data in the EIP message.

Code = 0x16 Set_Attribute_Single.
Class = 0x04
Instance = 0x66, 102
Attribute = 3
Data = **01 00 01 00 13 A4 14 00 00 00**

Where…
00 01 = 1, Run Code is 1, always
00 01 = 1, Sensor #1*
A4 13 = **42003***, Special location code for delay setting on sensor #1
00 14 = Data value of 20, decimal (per M-Node manual, interpreted as 2.0)
00 00 = Lock code. If lock disabled, enter 0.

See screenshot using Molex EIP tool below.

*Writing data to the MetriNet system is done by referencing a coded table that controls the variable identified fro that change. All writable data is passed though the same “window,” which can provide added security for any changes by utilizing the user lock codes on the MetriNet if desired.

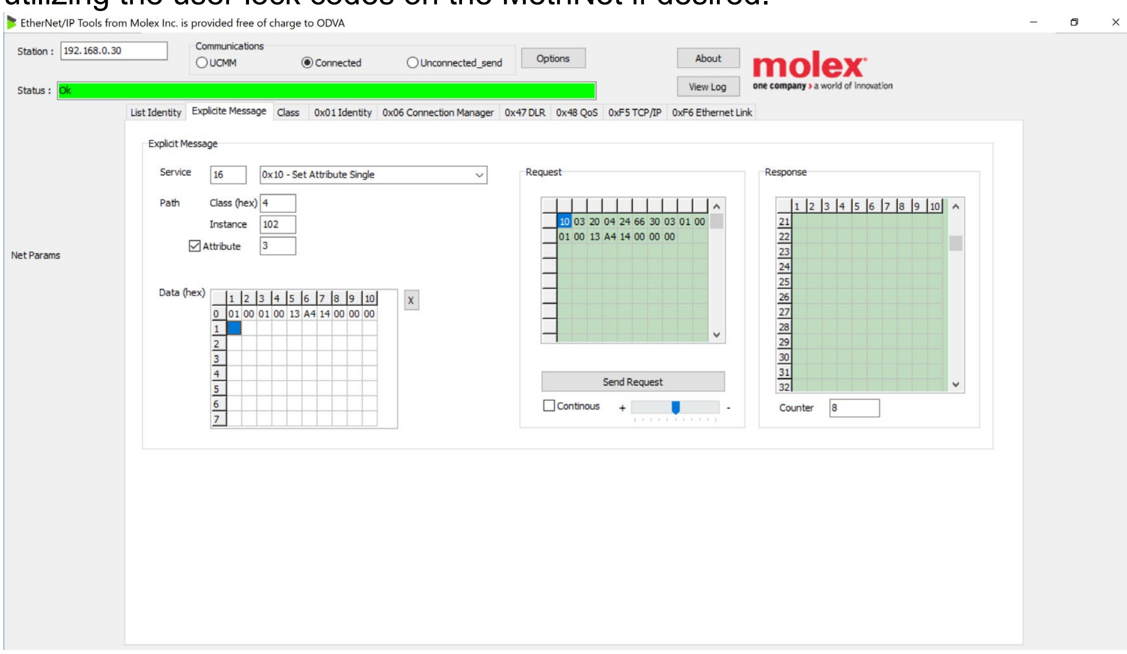

Figure 10 –Output Assembly Object Command Example with EIP_Tool

Once this message is sent, the S1 delay setting will be updated to 2.0 on the MetriNet.

The special codes for writing individual registers in the MetriNet come from specific Modbus register locations in the unit.

Snsr 1 Reg	Snsr 2 Reg	Snsr 3 Reg	Snsr 4 Reg	Snsr 5 Reg	Snsr 6 Reg	Snsr 7 Reg	Snsr 8 Reg	Sensor Data	Data Format
42003	42021	42039	42057	42075	42093	42111	42129	Delay	10=1.0min
42004	42022	42040	42058	42076	42094	42112	42130	[1]Alarm A	(sensor dependent)
42005	42023	42041	42059	42077	42095	42113	42131	[1]Alarm B	(sensor dependent)
42006	42024	42042	42060	42078	42096	42114	42132	Slp Alarm	80=80%
42007	42025	42043	42061	42079	42097	42115	42133	Tmr Limit	90=90 days
42008	42026	42044	42062	42080	42098	42116	42134	[1,2]VoutHI	(sensor dependent)
42009	42027	42045	42063	42081	42099	42117	42135	[1,2]VoutLO	(sensor dependent)
42010	42028	42046	42064	42082	42100	42118	42136	TcMode	0 = F, 1 = C
42011	42029	42047	42065	42083	42101	42119	42137	[3]Tag1	0x70,0x48="p","H"
42012	42030	42048	42066	42084	42102	42120	42138	[3]Tag2	…
42013	42031	42049	42067	42085	42103	42121	42139	[3]Tag3	…
42014	42032	42050	42068	42086	42104	42122	42140	[3]Tag4	…
42015	42033	42051	42069	42087	42105	42123	42141	[3]Tag5	…
42016	42034	42052	42070	42088	42106	42124	42142	[3]Tag6	…
42017	42035	42053	42071	42089	42107	42125	42143	[3]Tag7	…
42018	42036	42054	42072	42090	42108	42126	42144	[3]Tag8	…

Figure 11 –Special Write-Codes for MetriNet Variable Changes

Notes on Table data –

[1] *Sensor dependent variable. The formatting of these variables are based on the specific data value from that sensor. See the M-Node sensor manual for details.*

[2] *There are no analog voltage outputs of the bussed MetriNet system. However, the scaled 0-2.5V value from the sensor can be used to simplify the creation of the scale value for other purposes.*

[3] *The Tag values are compressed ASCII characters stored in the sensor, and together they create a 16 character string for unique sensor identification. The user may change these to whatever they desire. For a Tag entry of 0x70 0x48 (hex 70, 48,) you would store the characters "pH."*

TCP Object (F5$_{HEX}$ – 1 Instance)

The following tables contain the attribute and common services information for the TCP Object

Class Attributes (Instance 0)

Attribute ID	Name	Data Type	Data Value	Access Rule
1	Revision	UINT	1	Get

TCP Instance Attributes (Instance 1)

Attribute ID	Name	Data Type	Default Data Value	Access Rule
1	Status[1]	DWORD	0x00010000	Get
2	Configuration Capability [2]	DWORD	0x00110000	Get
3	Configuration Control[3]	DWORD	0x00010000	Get
4	Physical Link Object[4] **Structure of:** Path Size Path	 UINT Array Of WORD	2 0xF620 0x0124	Get
5	Interface Configuration[5] **Structure of:** IP Address Network Mask Gateway Address Name Server Name Server 2 Domain Name Size Domain Name	 UDINT UDINT UDINT UDINT UDINT UINT STRING	 0x00FEC0A8 0xFF00FFFF 0x00000000 0x00000000 0x00000000 0x0000	Get
6	Host Name[6] **Structure of:** Host Name Size Host Name	 UINT STRING	0x0000	Get

TCP Instance Common Services

Service Code	Implemented for		Instance Level
	Class Level	Instance Level	
$0E_{HEX}$	Yes	Yes	Get_Attribute_Single

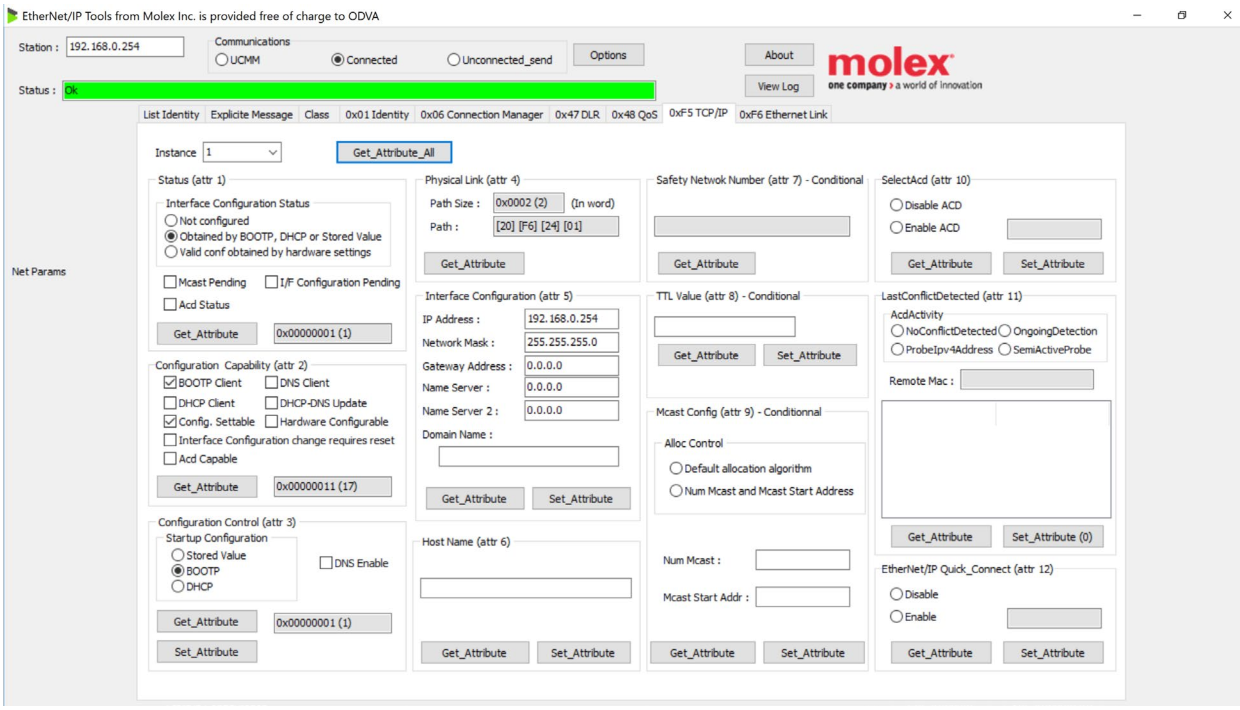

Figure 12 – TCP Object Query Example with EIP_Tool

Ethernet Link Object (F6HEX – 1 Instance)

The following tables contain the attribute and common services information for the Ethernet Link Object

Class Attributes

Attribute ID	Name	Data Type	Data Value	Access Rule
1	Revision	UINT	3	Get

Link Instance Attributes

Attribute ID	Name	Data Type	Default Data Value	Access Rule
1	Interface Speed[7]	UDINT	0x00640000	Get
2	Interface Flags[8]	DWORD	0x000F0000	Get
3	Physical Address[9]	USINT Array[6]	0x80009BA3D6BE	Get

Link Instance Common Services

Service Code	Implemented for		Service Name
	Class Level	Instance Level	
$0E_{HEX}$	Yes	Yes	Get_Attribute_Single

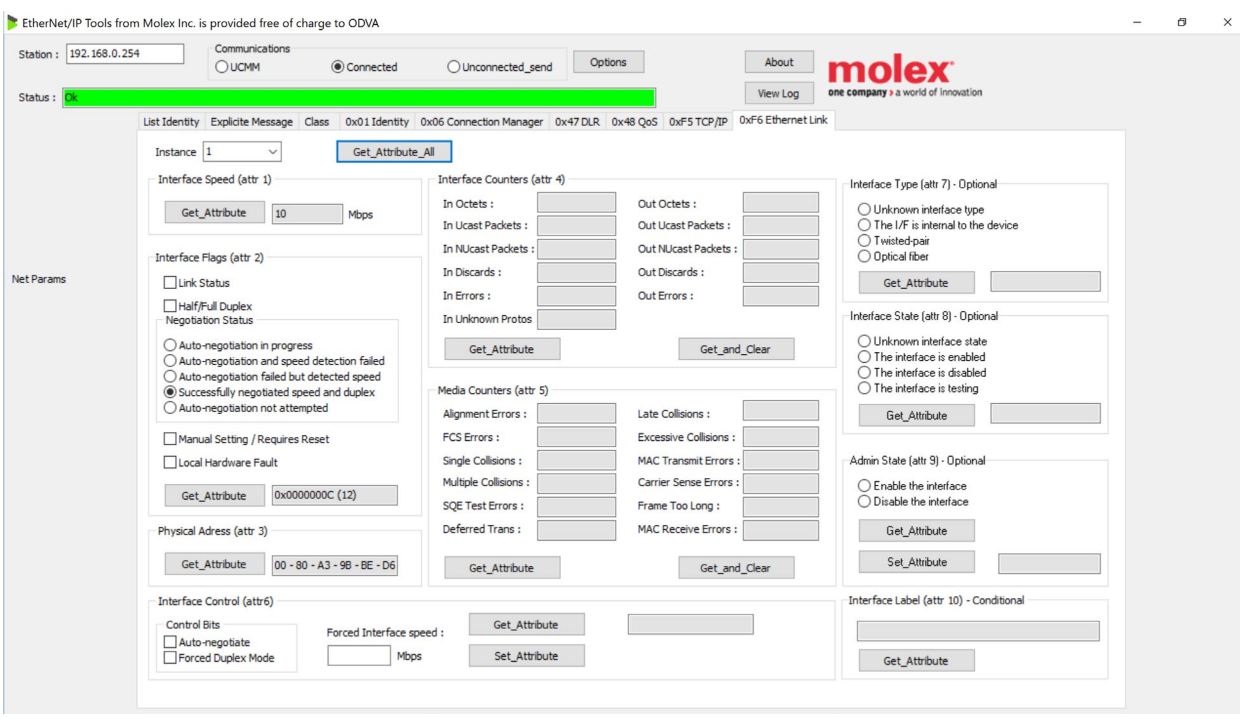

Figure 13 – Ethernet Link Object Query Example with EIP_Tool

Message Router Object (02HEX)

No supported services or attributes

Connection Manager Object (06HEX)

No supported services or attributes

1.10 Optional Tool Examples: RSLinx Classic

The primary method of setup for the MetriNet at this time is a Genetic Module" and is simply entered into most PLCs using the I/O settings found in section 1.8. Those main T->0 and O->T settings are then byte parsed into the needed final values. The EDS file if often not even needed for this type of installation.

A brief setup with RSLinx Classic using an OPC server is shown next, which will verify operation of the interface. While these screenshots are specific to Rockwell tools, the same set-up steps are part of other platforms that support Ethernet/IP. RSLinx Classic is chosen here as its very common for Ethernet/IP, and it is the foundation communication driver component for RSLogix Studio 5000. In addition, using the OPC server function allows all of this testing to be done without a PLC. This all runs on a single computer, with only the MetriNet connected to the PC Ethernet port. So, this test can also be used as a quick field test of Ethernet/IP adapter devices off the main network.

Before running this step, confirm that –

1-The proper EDS file has been properly loaded.

2-The MetriNet has already been configured with the proper IP address to match the network requirements. Turn on MetriNet but do not connect it to PC port yet.

3-For RSLinx Classic, this example assumes the user has a version other than the "Lite" version, as that version does not allow an OPC server connection. The "single node", "OEM", and "gateway" versions all can run an OPC server.

To begin, launch RSLinx Classic, and configure/add the Ethernet/IP Driver.

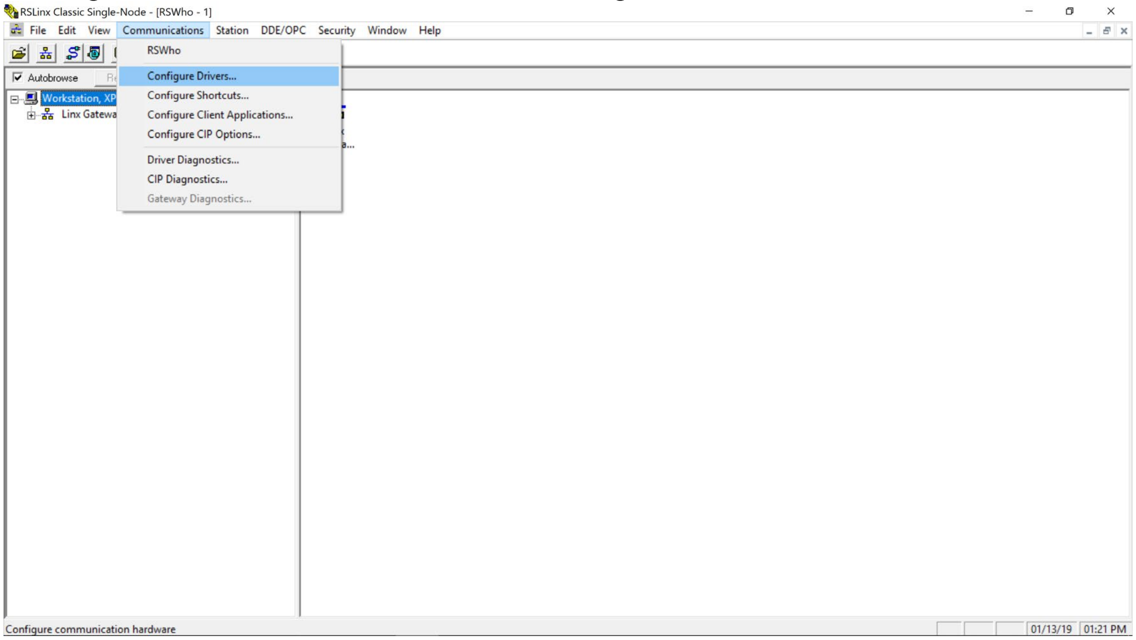

Figure 14 – RSLinx Classic

Add the driver to the system. You will need to select the active PC Ethernet port as part of the process. It will immediately begin running. Close this window.

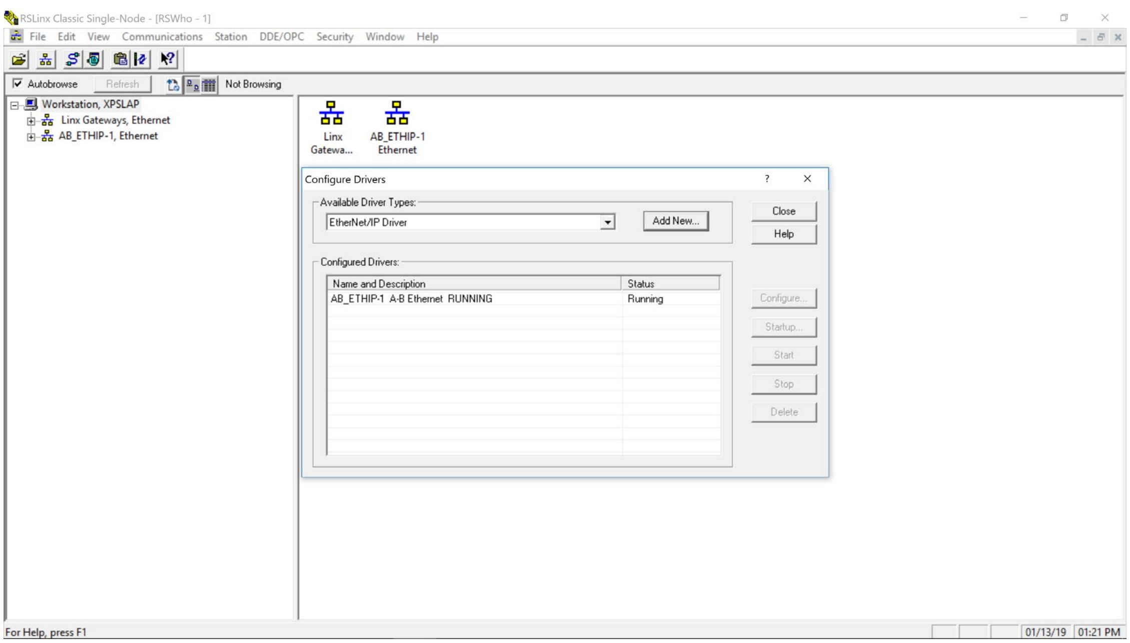

Figure 15 – Adding Ethernet/IP Driver

Connect MetriNet to the Ethernet port, and you should quickly see the node appear under the drive that was just added.

The icon is part of the MetriNet EDS file. Right clicking on node on right side of screen and selecting "Device Properties" will show the information from the node. The node is active now.

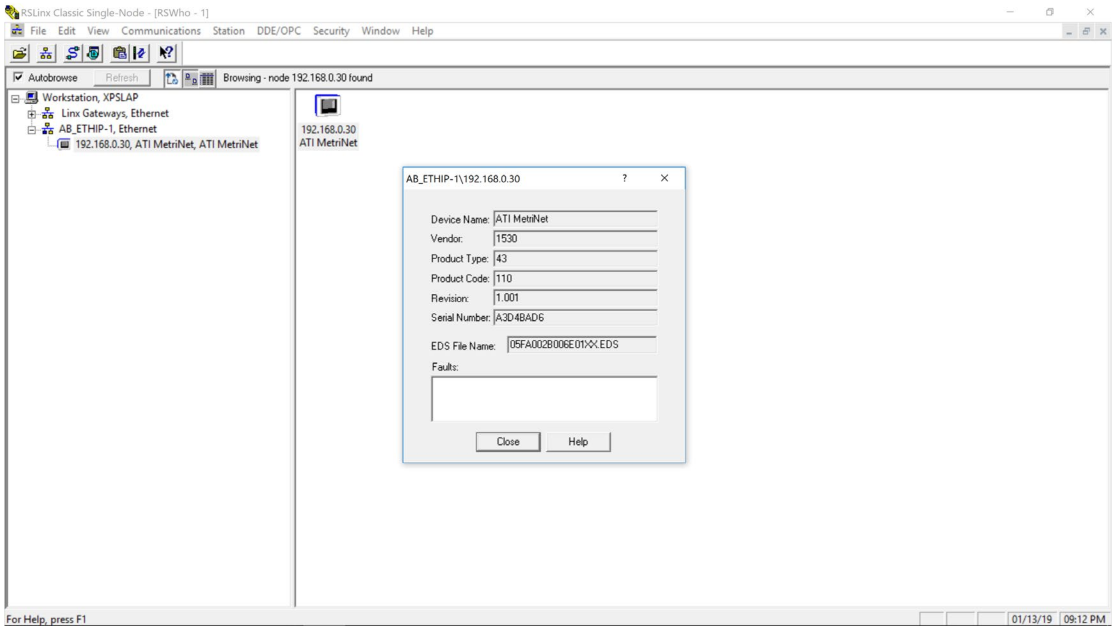

Figure 16 – New Node Added Successfully to Driver Application.

Close Properties pop-up. Now, a Topic must be added to give the OPC server access to the node. Select DDE/OPC and then "Add Topic." Select node from below Ethernet/IP driver on right, and create new Topic called Q46.

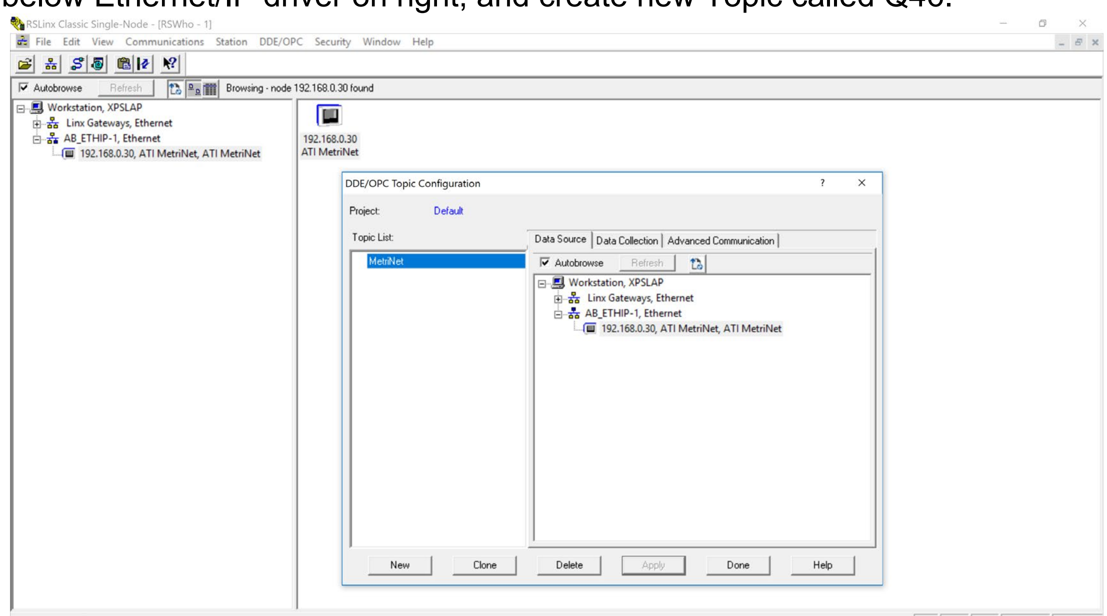

Figure 17 – Creating DDE/OPC Topic

On Data Collection tab, select “Device with EDS Parameters” and hit “apply” to update Q46 topic. Then select Done.

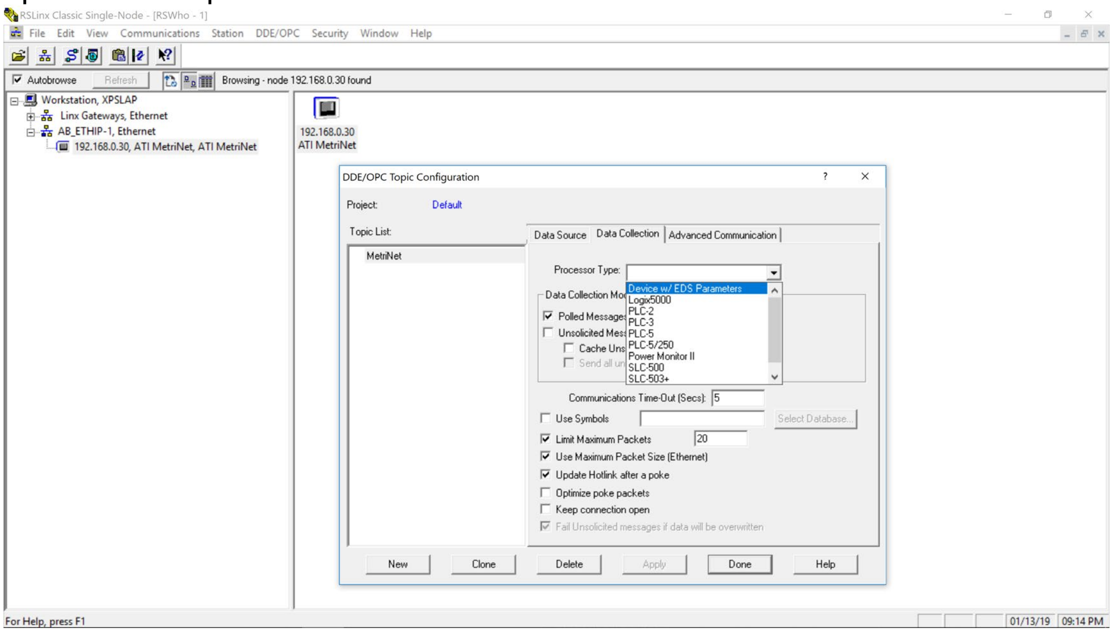

Figure 18 – Processor Selection for PC-based OPC Topic

Minimize this RSlinx window now (don’t close it though) and launch the Rockwell OPC Test Client utility, which will start the OPC test server. First, via the menus, connect server to RSLinx OPC server, and add a simple Group called “Test.” Set up an item next.

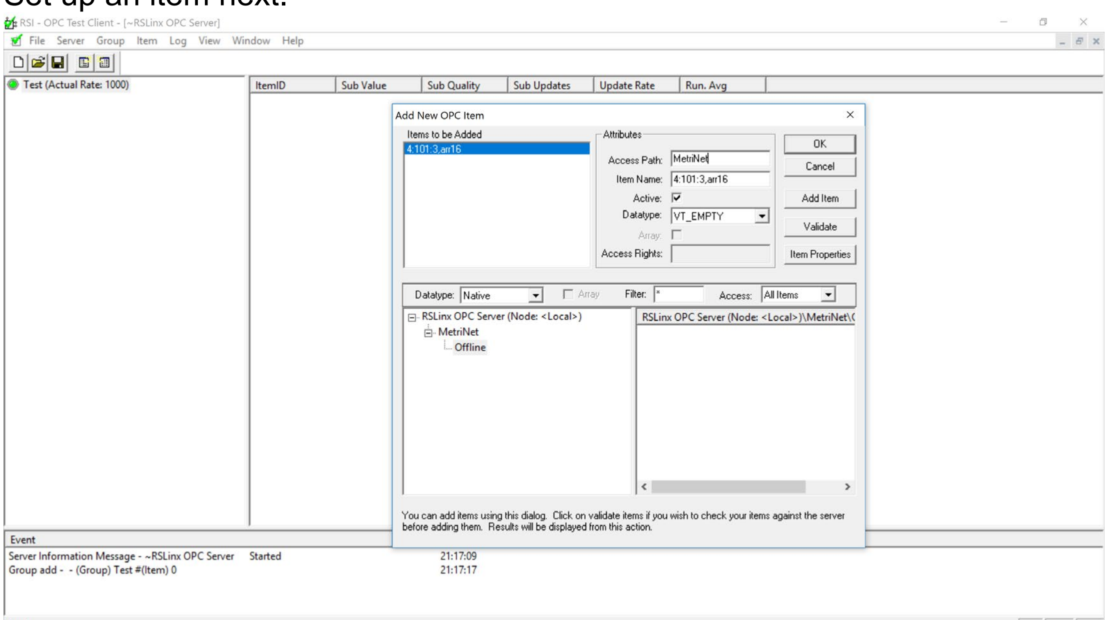

Figure 19 – RSLinx OPC Test Client

Click to highlight “Offline” under Q46, add the item as shown. The entered “Item Name” is the CIP direct path to Class 4, Instance 101, Attribute 3, and the added “arr16” indicates that we expect an array of 16-bit values. Select “Add Item” then “OK” and the following data should appear on the screen.

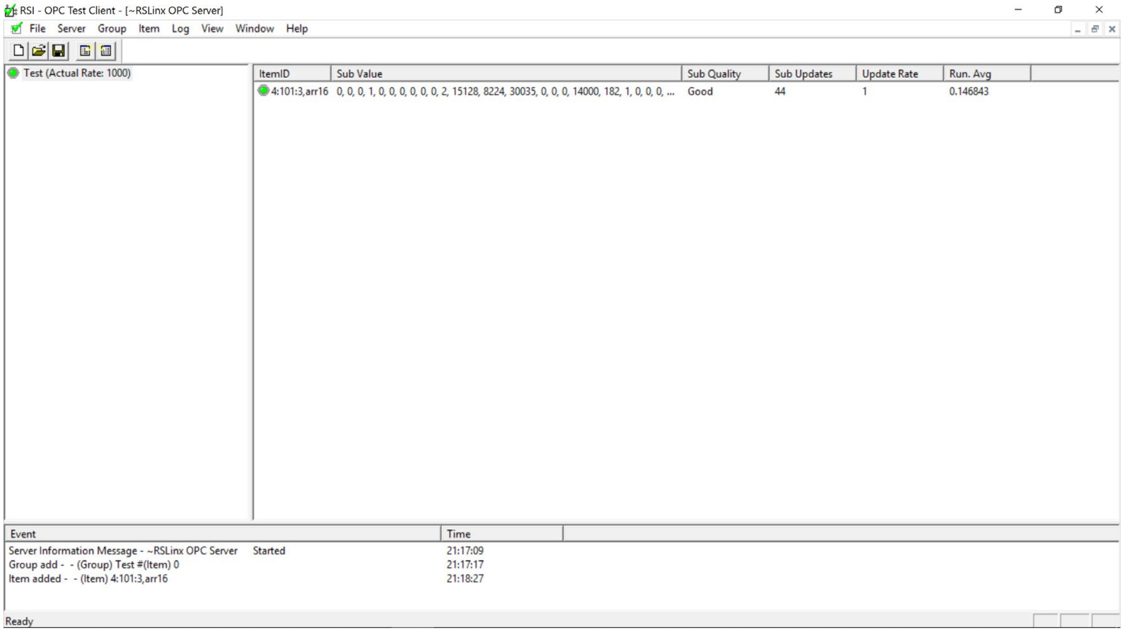

Figure 20 – OPC Running On Active Node Data

Note the data, this example is run with one single S1 Q32C2 M-Node attached, so all other sensor data beyond it will be zero. This is an array of 16-bit words.

The data is a string of bytes starting with the info structure (Input Assembly Object) which was shown on page 16-17. Here, we will look only at the System info, and then the info for S1 –

System Info

Bytes	Data Type	Description
0–1	WORD	Interface Status Flags
2–3	WORD	System Status 1 Flags
4–5	WORD	System Status 2 Flags
6–7	UINT	Number of Sensors
8–9	UINT	IP Address High
10–11	UINT	IP Address Low
12–13	UINT	Subnet Mask High
14–15	UINT	Subnet Mask Low
16–17	UINT	Gateway High
18–19	UINT	Gateway Low

The data in Figure 20 is then starts with the following string of decimal values - 0,0,0,1,0,0,0,0,0,0,2,15128,8224,30035,0,0,0,14000,182,1,0,0,…

Parsing SYSTEM INFO Data -

0 = Byte 0-1, Q52 diagnostic bits
0 = Byte 2-3, System Status 1 Flags
0 = Byte 4-5, System Status 2 Flags
1 = Byte 6-7, Number of sensors connected
0 = Byte 8-9, NU
0 = Byte 10-11, NU
0 = Byte 12-13, NU
0 = Byte 14-15, NU
0 = Byte 16-17, NU
0 = Byte 18-19, NU

Sensor 1 MEASURE

Bytes	Data Type	Description
20-23	DINT	S1 Main Value
24-27	DINT	S1 Units
28-31	DINT	S1 Raw Value
32-35	DINT	S1 Temperature
36-37	INT	S1 Output Value
38-39	WORD	S1 Status 1
40-41	WORD	S1 Status 2
42-43	INT	S2 ID

Parsing S1 Sensor MEASURE Data -

2 = Byte 20-21, S1 MainValue high byte (0x0002)
15128 = Byte 22-23, S1 MainValue low byte (0x3b18)
8224 = Byte 24-25, S1 Units high byte (0x2020)
30035 = Byte 26-27, S1 Units low bye (0x7553)
0 = Byte 28-29, S1 RawValue high byte (0x0000)
0 = Byte 30-31, S1 RawValue low byte (0x0000)
0 = Byte 32-33, S1 Temperature high byte (0x0000)
14000 = Byte 34-35, S1 Temperature low byte (0x36b0)
182 = Byte 36-37, S1 Output Value (0.182 VDC)
1 = Byte 38-39, S1 Status Flags 1 (0x0001)
0 = Byte 40-41, S1 Status Flags 2 (0x0000)
0 = Byte 42-43, S1 ID (0x0000) NU

The values above in red are 32 bit INT type. So, combine them to get total value.

MainValue = 0x00023b18 = 1462000, or 146.200 uS (implied 0.000 dp)
Units = 0x20207553 = _ _ u S (ASCII text)
RawValue = 0x00000000 = 0, or 0.000 nA
Temperature = 0x000036b0 = 14000, or 14.000C

WATER QUALITY MONITORS

Dissolved Oxygen
Free Chlorine
Combined Chlorine
Total Chlorine
Residual Chlorine Dioxide
Potassium Permanganate
Dissolved Ozone
pH/ORP
Conductivity
Hydrogen Peroxide
Peracetic Acid
Dissolved Sulfide
Residual Sulfite
Fluoride
Dissolved Ammonia
Turbidity
Suspended Solids
Sludge Blanket Level
MetriNet Distribution Monitor

GAS DETECTION PRODUCTS

NH_3	Ammonia
CO	Carbon Monoxide
H_2	Hydrogen
NO	Nitric Oxide
O_2	Oxygen
CO	Cl2 Phosgene
Br_2	Bromine
Cl_2	Chlorine
ClO_2	Chlorine Dioxide
F_2	Fluorine
I_2	Iodine
H_X	Acid Gases
C_2H_4O	Ethylene Oxide
C_2H_6O	Alcohol
O_3	Ozone
CH_4	Methane (Combustible Gas)
H_2O_2	Hydrogen Peroxide
HCl	Hydrogen Chloride
HCN	Hydrogen Cyanide
HF	Hydrogen Fluoride
H_2S	Hydrogen Sulfide
NO_2	Nitrogen Dioxide
NO_x	Oxides of Nitrogen
SO_2	Sulfur Dioxide
H_2Se	Hydrogen Selenide
B_2H_6	Diborane
GeH_4	Germane
AsH_3	Arsine
PH_3	Phosphine
SiH_4	Silane
HCHO	Formaldehyde
$C_2H_4O_3$	Peracetic Acid
DMA	Dimethylamine