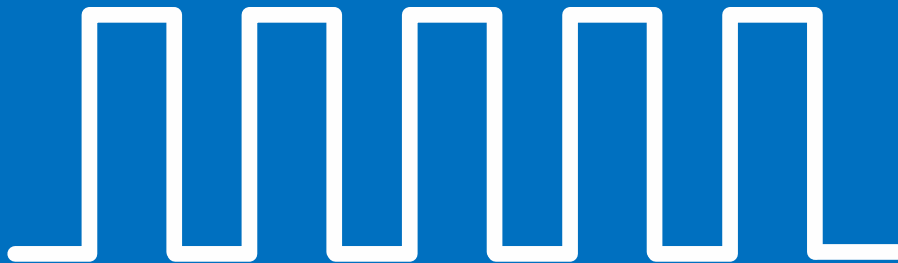




O & M Manual



Model Q46 Modbus TCP 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

Table of Contents

Part 1 – Modbus TCP Description	3
1.1 General	3
1.2 Modbus TCP Communication	3
1.3 Modbus TCP Transmission Details	4
1.4 Modbus TCP Card Installation Instructions	6
1.5 Cable Connection	7
1.6 Interface Operation	8
1.7 Setting Manual IP Address	9
1.8 Q46 Modbus TCP Register Assignment	12

Table of Figures

Figure 1 – Modbus TCP Card Installation	6
Figure 2 - RJ45 Crimp Tool	7
Figure 3 – Simply Modbus TCP 7.0 Tool Example	8
Figure 4 - DeviceInstaller Identifies Q46 Interface Connection	9
Figure 5 - DeviceInstaller Default Port Details	9
Figure 6 - DeviceInstaller Manual IP Settings	10
Figure 7 - DeviceInstaller Completed Port Assignment	10
Figure 8 - DeviceInstaller Updated Port Details at New IP	11
Figure 9 - SimplyModbus Client Scan at New IP	11
Figure 10 - Q46 Modbus Input Base Register Data	12
Figure 11 – Base Status Register Bitfield Flags	12
Figure 12 – Parameter Specific Data Formats	13

Part 1 – Modbus TCP Description

1.1 General

Q46 Analyzers are available with four digital communication options. Profibus-DP, Modbus-RTU, Modbus-TCP, or Ethernet-IP. This manual applies only to instruments supplied with the Modbus TCP communication option.

It is important critical to note that Modbus TCP devices only communicate with other systems running that application protocol on Ethernet. You cannot plug a Q46 into your office Ethernet and expect to have the Q46 talk to your computer, unless you have a Modbus protocol application on that computer.

The discussion of standard Modbus and Ethernet are 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 Modbus TCP Communication

It should be understood that Modbus TCP 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.

Per the *Modbus Messaging on TCP/IP Implementation Guide V1.0b*, authored by Modbus.org, a 5-layer Internet model is used for Modbus TCP instead of the familiar three layer model for RTU/ASCII over serial line. This new standard encapsulates standard Modbus function code and data contents into a higher level TCP protocol. The physical layer is not specifically mentioned in the guide, therefore, the wiring infrastructure generally just follows standard Ethernet wiring practices.

The Modbus "bus" structure in this case is an IP bus, and slave addresses become IP addresses. Also, the master/slave relationship in RTU/ASCII becomes more of a client/server relationship in Modbus TCP – where multiple clients initiate requests to servers (slaves) for information. So, unlike the serial implementation of bussed RTU/ASCII, the TCP implementation becomes more "multi-master" in nature. However, even with all these variations, Modbus TCP data looks exactly like Modbus RTU/ASCII data in data format and structure.

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 industrial protocol formats, so that interface must be handled by a specific program that recognizes the format. Because of this, a Modbus TCP device cannot be directly connected to your office network for transferring information.

The Q46 utilizes a gateway device inside the analyzer that maps an internal Modbus data structure into the required Modbus TCP data object. This is done to maintain plug-and-play compatibility to all Q46 digital communication options. Any communication option can be plugged into any Q46 for immediate conversion to a new protocol. Because this gateway device is part of the Q46 interface, the ID and device type show up on the interface as the gateway device. However, the Q46 variation type is mapped into the application data object as the last byte.

1.3 Modbus TCP Transmission Details

The data for the protocol is constructed into a specific structure inside a standard TCP Packet. A user application program simply decodes the structure inside the received TCP or UDP packet. To create the Modbus TCP frame, the standard Modbus data frame PDU (Protocol Data Unit) is appended by a new MBAP header (Modbus Application Protocol) to make a larger transmission frame called the ADU (Application Data Unit.) In order to send the new ADU over TCP, the registered port number 502 is used by Modbus.org.

MODBUS PDU

Modbus protocol is a messaging structure, widely used to establish master-slave communication between intelligent devices. In serial format, a message sent from a master to a slave contains a one-byte slave address, a one-byte command, data bytes (depending on command), and a two byte CRC. The protocol is independent of the underlying physical layer and is traditionally implemented using RS232, RS422, or RS485 over a variety of media (e.g. fiber, radio, cellular, etc.) The TCP format strips out the slave address and CRC and just uses the function code and data. So, the basic structure of the PDU frame section is simply:

|<- Modbus PDU ->|
[FUNCTION][DATA]

The function code field of a message frame contains an eight-bit code in the range of 1 ... 255 decimal. When a query message is sent from the master, the function code field tells the slave device what kind of action to perform. Examples include reading the contents of a group of registers, writing to a single register, writing to a group of registers, and reading the exception status.

When the slave device responds to the master, it uses the function code field to indicate either a normal (error-free) response or that some kind of error occurred (called an exception response). For a normal response, the slave simply echoes the original function code. For an exception response, the slave returns a code that is equivalent to the original function code with its most significant bit set to logic 1.

The data field is constructed of one or more bytes and contains additional information, which the slave must use to take the action defined by the function code. This can include items like discrete and register addresses, the quantity of items to be handled, and the count of actual data bytes in the field. If no error occurs, the data field of a response from a slave to a master contains the data requested. If an error occurs, the field contains an exception code that the master application can use to determine the next action to be taken.

The data field can be nonexistent (of zero length) in certain kinds of messages. For example, in a request from a master device for a slave to respond with its communications event log (function code 0B hexadecimal), the slave does not require any additional information. The function code alone specifies the action.

MODBUS ADU

Modbus TCP adds a header to the PDU to get the full ADU transmission frame –



Where:

Transaction ID = 2 bytes set by client to uniquely identify the request

Protocol ID = 2 bytes set by client (always 00 00)

Length = 2 bytes identifying the number of bytes in the message

Unit ID = 1 byte set by client to ID a slave connected on a remote serial line

MODBUS REGISTERS AND COILS

The PDU data portion of the Modbus TCP follows the exact same register format as the serial form of the protocol. Modbus protocol was originally designed to transfer data to and from PLCs (Programmable Logic Controllers), which organize data into groups of registers and coils. PLC registers containing i/o information are called input registers and are numbered 30001 to 39999, while registers containing data or the results of calculations are known as holding registers and are numbered from 40001 to 49999. The term coils, on the other hand, refers to discrete (0 or 1) inputs and outputs. Traditionally, these are inputs from such things as switch closures and outputs to the coils of relays, which are under the control of the PLC.

All registers are 16 bit values, which may be read or written to individually, or in blocks by using specific functions. Likewise for coils, which are one bit values. Since register functions transfer 16 bits and discrete (coil) functions transfer only one, it is usually more efficient to use register functions, which reduces the number of messages required to transfer data. For this reason, the Q46 Series transmitter organizes all of its data into input registers only, or more specifically, data is organized into the holding registers starting at 30001.

The protocol specifies which registers to access by the value of the function code embedded into the message. For example, to read one or more holding registers in a slave device, the master must use function 3 – “Read Holding Register”. Similarly, the master must use function 4 – “Read Input Register” to read one or more of the input registers. The Q46 only responds to request for reading input registers (Function 4).

For more information on the protocol, please refer to the “Modicon Modbus Protocol Reference Guide” at -

<http://www.modicon.com/techpubs/toc7.html>

or, “Modbus Protocol Specification”, available for download at

<http://www.modbus-ida.org/specs.php>

Deviations from this guide are noted in the appropriate section. More information regarding Modbus, in general, may be viewed at:

<http://www.modbus-ida.org/>

1.4 Modbus TCP Card Installation Instructions

Use the following instructions to install the Modbus TCP Communications Option into an existing Q46 Analyzer. Monitors ordered with the Ethernet option will be supplied with the board already installed.

1. Disconnect power to the Q46 prior to installation of the board.
2. Remove the connector cover on the power supply circuit board located next to the fuse.
3. Carefully plug the Modbus TCP circuit board into the connector on the power supply board.
4. Align the cover board holes with the holes in the circuit board and use the screws supplied to secure the assembly. Refer to Figure 1 below for the proper assembly.

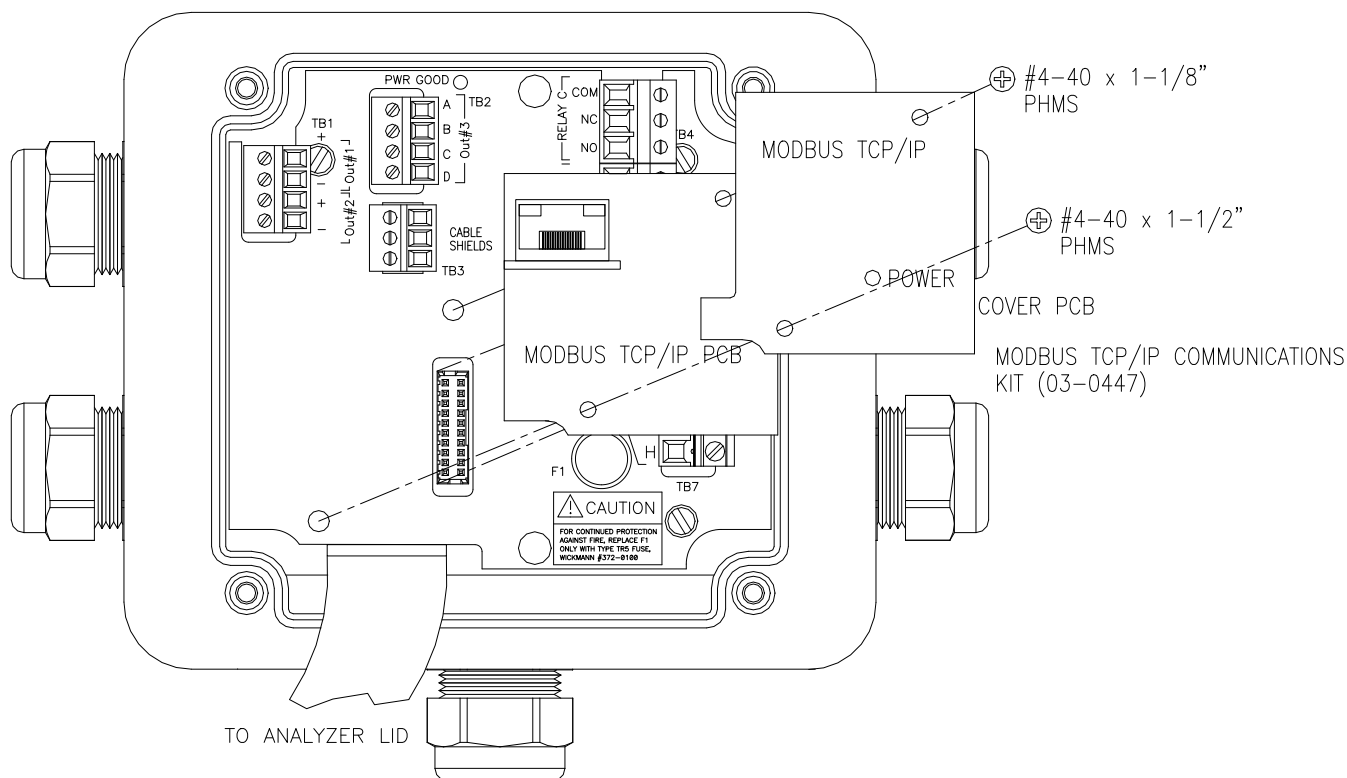


Figure 1 – Modbus TCP Card Installation

1.5 Cable Connection

The cable used for Modbus TCP 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 Q46, pass the unterminated cable through the cable gland nearest the location of the RJ45 connector on the Modbus TCP option board. Termination of Cat5/5e/6 cables is very easy and can be completed quickly with the commonly available RJ45 crimp tool.



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 Q46 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 Modbus TCP 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.

1.6 Interface Operation

On power-up, the Q46 looks for a DHCP server to automatically assign an IP address on the network. Once the IP is set, Modbus TCP data can be transferred across the network just like a standard Modbus RTU bus system.

As an example of the data structure, a completed Q46 data set is displayed below in a common diagnostic PC software tool - **Simply Modbus TCP**. The user simply enters the IP address established by the client PC, and sets up the standard Modbus PDU data set details.

To run this demo tool on a PC for check-out of the Q46 interface, the user must first be sure to set the IP address of the PC Ethernet port to the proper subnet - and so the user must be able to see the interface on the network first. There are various tools available for viewing that defined IP address, but it should be noted that the name which will be “**X-PORT**.”

Any TCP test software supporting Modbus will work. The “Simple Modbus TCP Client” test software can be found at – <http://www.simplymodbus.ca/TCPclient.htm>

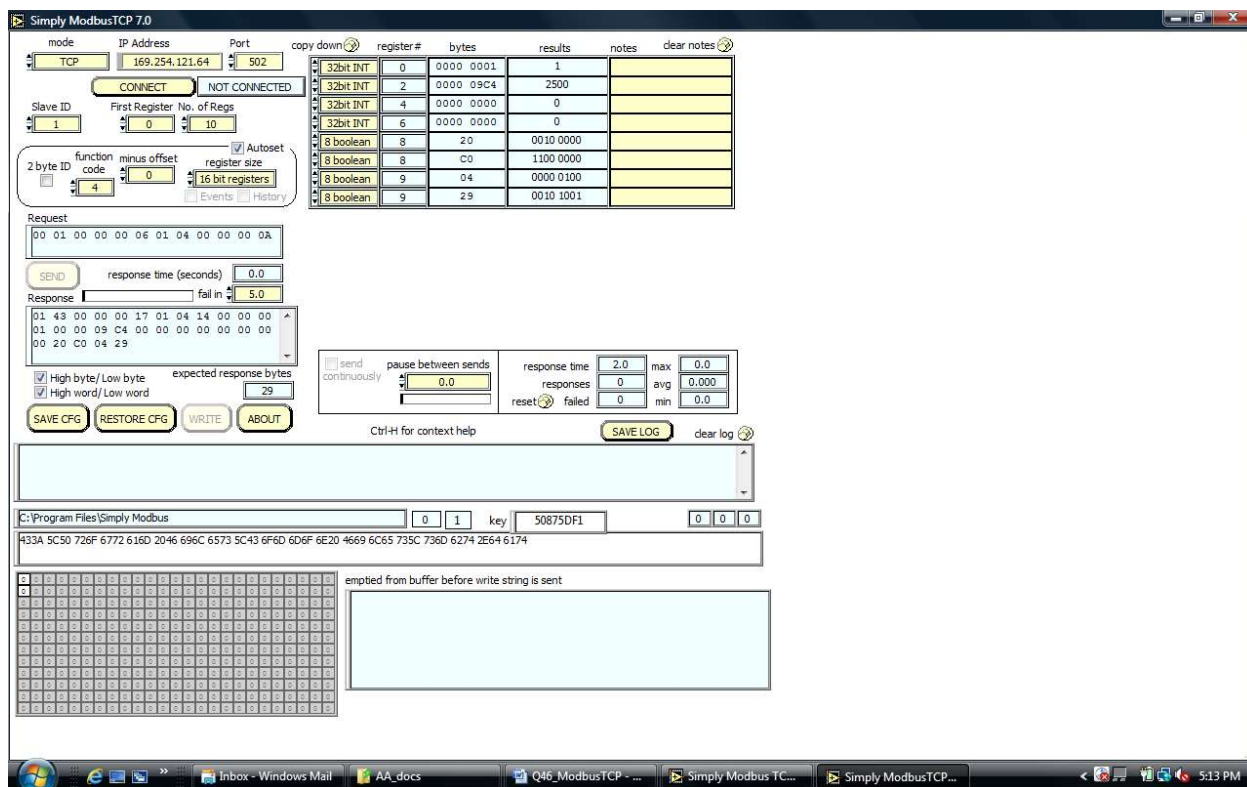


Figure 3 – Simply Modbus TCP 7.0 Tool Example

1.7 Setting Manual IP Address

The Modbus TCP interface on the Q46 utilizes a Lantronix interface gateway, with a default setting for auto-IP assignment with DHCP. As an alternative to using DHCP to set up the interface IP address, the IP may be set manually and stored in the Q46.

The communications interface option ships from the factory with DHCP enabled, and the default IP for the actual port setting can be located outside the subnet/gateway reach of the PC connected. Because of this, it can be difficult to access quickly on a PC with preset subnet/gateway settings that are different than the default Q46 port (would mean that you might have to shift the subnet or gateway to reach that default address manually.) Therefore, the easiest way to set this IP manually is to use a program called “DeviceInstaller” from Lantronix.

This free program is located here - <http://www.lantronix.com/products/deviceinstaller/>

Once this program has been installed on a PC, plug in the Q46 and connect an Ethernet cable from the Q46 communications port to an Ethernet port of the PC running the program. Then, launch the DeviceInstaller program.

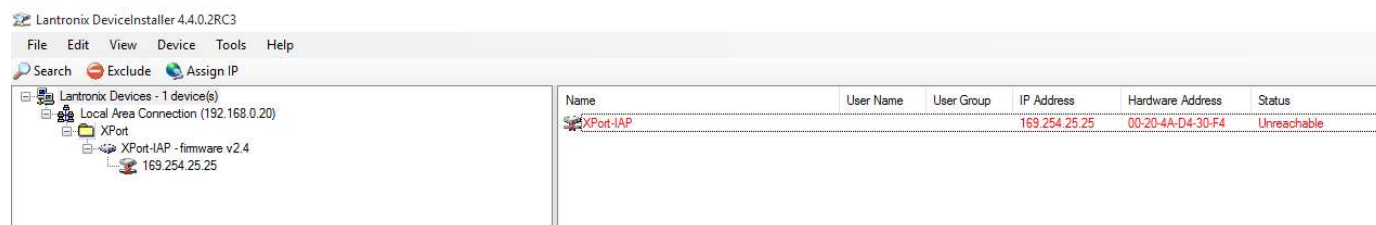


Figure 4 - DeviceInstaller Identifies Q46 Interface Connection

In this case, the default IP for the actual interface is outside this PC’s reach at 169.254, so we will manually set it in the 192.168.0 range. For this Q46 port, we will randomly choose 192.168.0.21. To do this, simply click on the actual IP address of the part now, “169.254.25.25” above. A window will pop up indicating “The configuration could not be retrieved from the device”, and just hit “OK.” Now the following detail screen will be shown –

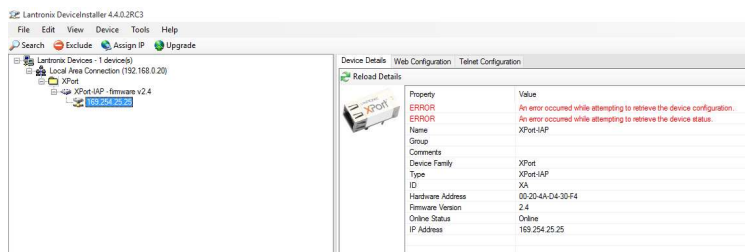


Figure 5 - DeviceInstaller Default Port Details

At this point, select “**Assign IP**” from the upper bar. A window will pop up asking if you want to set the IP automatically, or to set it manually. Choose “**Assign a Specific IP Address.**” The following screen will appear with entry fields for IP, subnet, and gateway. In this example we chose to configure as follows (your settings depend on your own subnet/gateway requirements) –

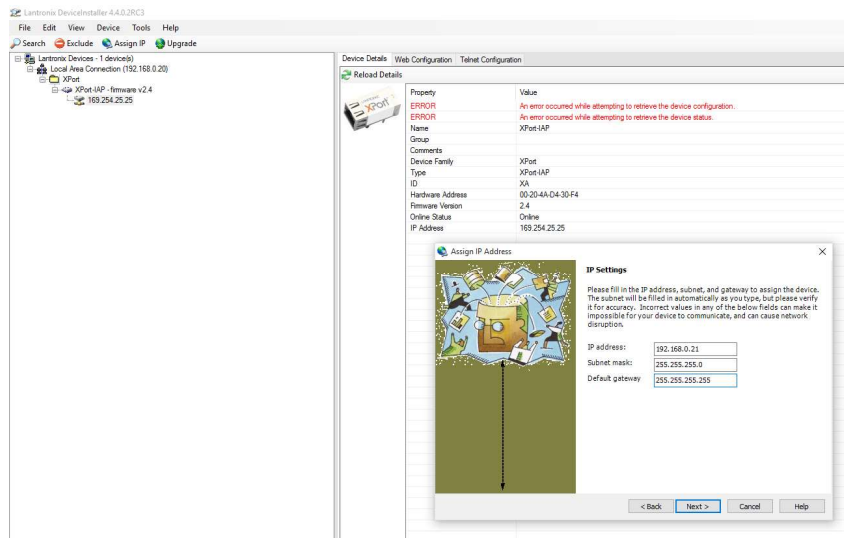


Figure 6 - DeviceInstaller Manual IP Settings

After the IP/subnet/gateway info is entered, hit “**Next**”, then click on “**Assign.**” Wait for “progress of task bar” to complete the operation. You will now get a pop-up screen again that shows “**The configuration could not be retrieved from the device**”, just click OK. Then click “**Finish**” to clear window -

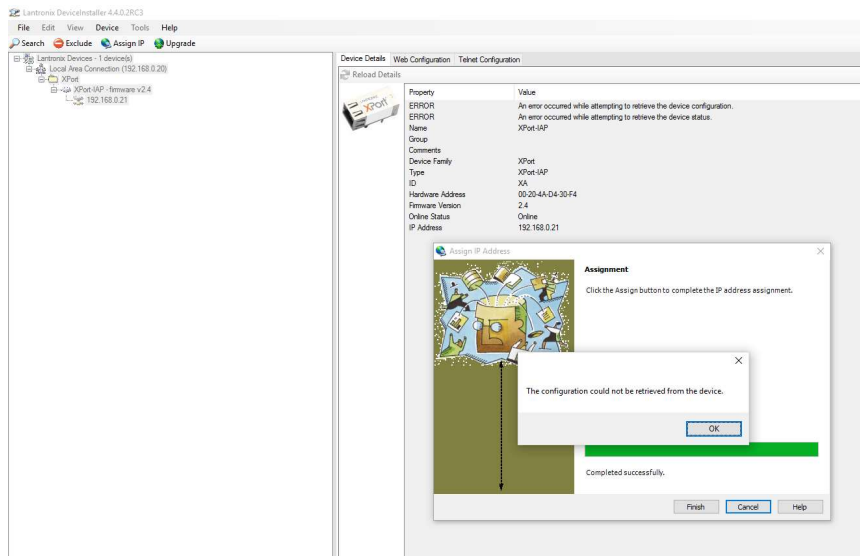


Figure 7 - DeviceInstaller Completed Port Assignment

The updated IP screen detail for the port will show the following new information

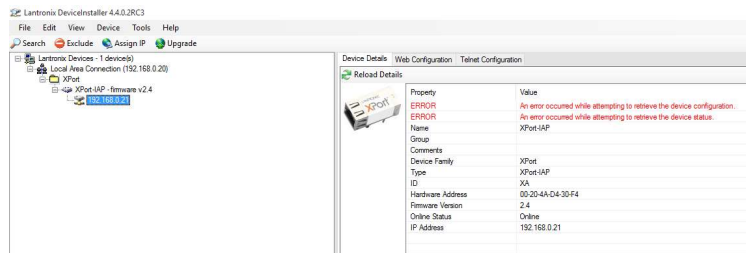


Figure 8 - DeviceInstaller Updated Port Details at New IP

You can now close this program, and the Q46 is now fixed on these IP settings. You must cycle power to the Q46 for this new IP to take effect. In our SimplyModbus program, setting this IP and hitting "connect" now connects the PC to the Q46 port. See screen shot of it below –

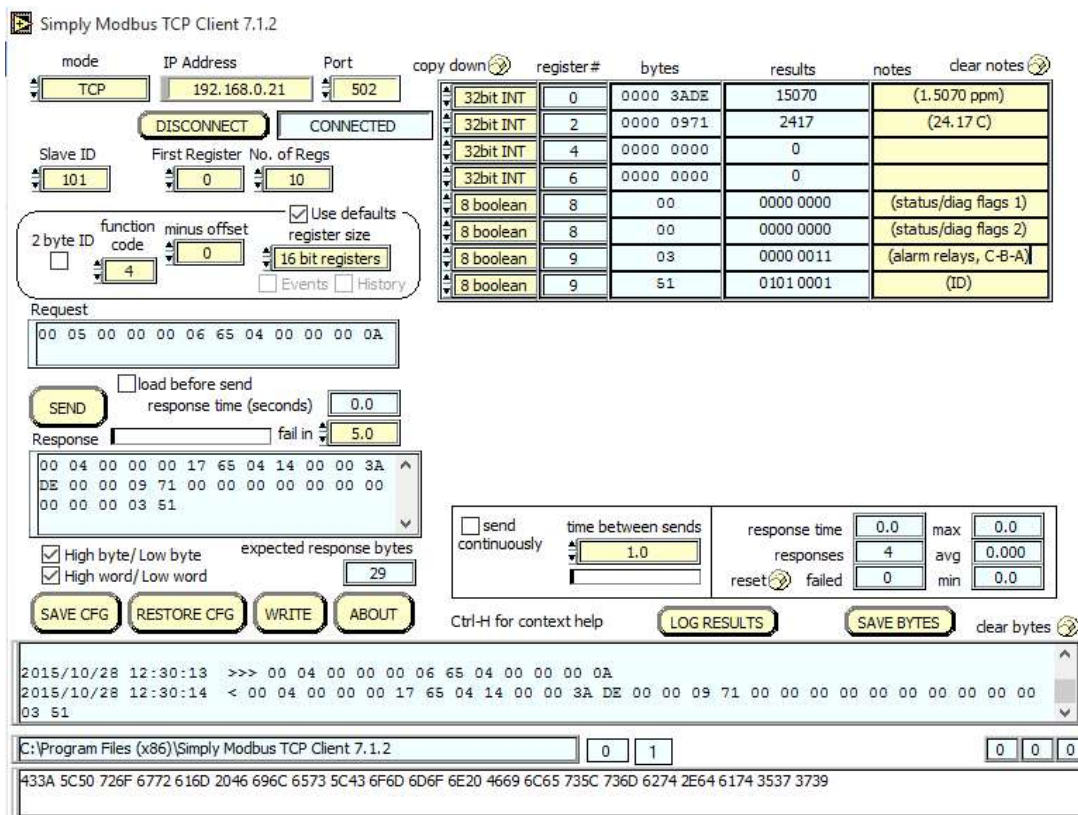


Figure 9 - SimplyModbus Client Scan at New IP

1.8 Q46 Modbus TCP Register Assignment

The following tables summarize the registers used for the Modbus data in all Q46 communication boards. Specific parameter values for all variations are listed in figure 12. As mentioned previously, the Q46 only supports the “04 – Read Input Register” function.

Register	Byte	Data Type	Sensor	Description	Data Format
30001 (HI) 30002 (LO)	1 to 4	long(32)	Main	Measured Main	See Figure 12
30003 (HI) 30004 (LO)	5 to 8	long(32)	Main	Measured Temperature	25.00° C = 2500
30005 (HI) 30006 (LO)	9 to 12	long(32)	Aux1	Measured Aux1	See Figure 12
30007 (HI) 30008 (LO)	13 to 16	long(32)	Aux2	Measured Aux2	See Figure 12
30009	17	Char(8)	NA	System Status 1	(binary) 00000000
	18	Char(8)	NA	System Status 2	(binary) 00000000
30010	19	Char(8)	NA	Alarm Status	(binary) 00000000
	20	Char(8)	NA	Instrument ID	See Figure 5

Figure 10 - Q46 Modbus Input Base Register Data

Register	Bitfield	Description
30009	0 (lsb)	Raw Hi ¹
	1	Raw Lo ¹
	2	Main Hi
	3	Main Lo
	4	Temp Hi
	5	Temp Lo
	6	Aux1 Hi ²
	7	Aux1 Lo
	0	NU
	1	NU
	2	LCD Controller Fail
	3	Cal Main Fail
	4	PID Controller Fail
	5	Cal TC Fail
	6	TC Error
7	Acknowledge Fail (global)	
30010	0	Alarm 1, Relay A
	1	Alarm 2, Relay B
	2	Alarm 3, Relay C
	3	Alarm 4, Relay D (optional)
	4	Alarm 5, Relay E (optional)
	5	Alarm 6, Relay F (optional)
	6	NU
	7	NU

Figure 11 – Base Status Register Bitfield Flags

Instrument	Main	Aux1	Aux2	Main Data Format	Aux1 Data Format	Aux2 Data Format
Q46H/62/63 /79PR ID=41	Chlorine	pH	PID	14920= 1.4920 ppm	700= 7.00pH	475= 47.5%
Q46S/66 ID=81	Sulfite	None	PID	14920= 1.4920 ppm	None	475= 47.5%
Q46S/81 ID=81	Sulfide	None	PID	14920= 1.4920 ppm	None	475= 47.5%
Q46H/64 ID=41	Ozone	pH	PID	14920= 1.4920 ppm	700= 7.00pH	475= 47.5%
Q46H/65 ID=41	Chlor Diox	pH	PID	14920= 1.4920 ppm	700= 7.00pH	475= 47.5%
Q46D ID=49	D.O.	Sat	PID	840= 8.40 ppm	980= 98.0% sat	475= 47.5%
Q46P ID=53	pH	None	PID	700= 7.00 pH	None	475= 47.5%
Q46R ID=57	ORP	None	PID	-137= -137 mV	None	475= 47.5%
Q46N ID=45	Total Amm	Mono Chloramine	Free Amm	100= 1.00 ppm	51= 0.51 ppm	321= 3.21 ppm
Q46C2 ID=69	Conductivity	³ Concentration	³ TDS	100123= 100.123 uS	13= 1.3%	2235= 223.5 mg/L
Q46C4 ID=65	Conductivity	³ Concentration	³ TDS	2238= 2.238 mS	13= 1.3%	2235= 223.5 mg/L
Q46CT ID=61	Conductivity	³ Concentration	³ TDS	2238= 2.238 mS	13= 1.3%	2235= 223.5 mg/L
Q46F ID=73	Fluoride	None	PID	1000= 1.000 ppm	None	475= 47.5%
Q46T ID=85	Turbidity	None	PID	16= 0.16 NTU	None	475= 47.5%

Figure 12 – Parameter Specific Data Formats

¹ Raw Hi/Lo is an internal sensor flag signal that indicates the raw signal is measuring too high or low, and can indicate sensor service is required or it may simply mean the sensor is presently out of the process liquid. This signal does not apply to the Q46T/C2/C4/CT.

² This bit flag is the glass break diagnostic for pH.

³ For Conductivity units, concentration data shows for concentration instruments, and TDS shows for standard conductivity units (not both.)

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

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	Cl ₂ Phosgene
Br ₂	Bromine
Cl ₂	Chlorine
ClO ₂	Chlorine Dioxide
F ₂	Fluorine
I ₂	Iodine
H _x	Acid Gases
C ₂ H ₄ O	Ethylene Oxide
C ₂ H ₆ O	Alcohol
O ₃	Ozone
CH ₄	Methane (Combustible Gas)
H ₂ O ₂	Hydrogen Peroxide
HCl	Hydrogen Chloride
HCN	Hydrogen Cyanide
HF	Hydrogen Fluoride
H ₂ S	Hydrogen Sulfide
NO ₂	Nitrogen Dioxide
NO _x	Oxides of Nitrogen
SO ₂	Sulfur Dioxide
H ₂ Se	Hydrogen Selenide
B ₂ H ₆	Diborane
GeH ₄	Germane
AsH ₃	Arsine
PH ₃	Phosphine
SiH ₄	Silane
HCHO	Formaldehyde
C ₂ H ₄ O ₃	Peracetic Acid
DMA	Dimethylamine