BC9180™

Base Station/Charger
Ethernet Base

Software Reference Guide

Datalogic ADC S.r.l.
Via S. Vitalino, 13
40012 Lippo di Calderara di Reno
Bologna - Italy
Tel. +39 051 3147011
Fax +39 051 3147205

©2013-2016 Datalogic ADC S.r.l.

An Unpublished Work - All rights reserved. No part of the contents of this documentation or the procedures described therein may be reproduced or transmitted in any form or by any means without prior written permission of Datalogic ADC S.r.l., or its subsidiaries or affiliates ("Datalogic" or "Datalogic ADC"). Owners of Datalogic products are hereby granted a non-exclusive, revocable license to reproduce and transmit this documentation for the purchaser's own internal business purposes. Purchaser shall not remove or alter any proprietary notices, including copyright notices, contained in this documentation and shall ensure that all notices appear on any reproductions of the documentation.
Should future revisions of this manual be published, you can acquire printed versions by contacting your Datalogic representative. Electronic versions may either be downloadable from the Datalogic website (www.datalogic.com) or provided on appropriate media. If you visit our website and would like to make comments or suggestions about this or other Datalogic publications, please let us know via the "Contact Datalogic" page.

Disclaimer

Datalogic has taken reasonable measures to provide information in this manual that is complete and accurate, however, Datalogic reserves the right to change any specification at any time without prior notice.
Datalogic and the Datalogic logo are registered trademarks of Datalogic S.p.A. in many countries, including the U.S.A. and the E.U. PowerScan is a trademark of Datalogic S.p.A. or of Datalogic Group companies, registered in many countries, including the U.S. and the E.U. PowerScan is a trademark of Datalogic S.p.A. or of Datalogic Group companies, registered in many countries, including the U.S. and the E.U.

All other brand and product names may be trademarks of their respective owners.

Patents

See www.patents.datalogic.com for patent list.

Table of Contents

INTRODUCTION 1
About this Manual 1
Manual Conventions 1
The BC9180™ Base Station/Charger 2
Datalogic Aladdin™ 3
Technical Support 3
Datalogic Website Support 3
Reseller Technical Support 3
Telephone Technical Support 3
..... 3

SETUP 5
BC9180™ Startup and Connection to Ethernet Network 5
BC9180 Configuration 6
Selecting a Non-Ethernet interface 7

ETHERNET HOST INTERFACE OPERATION 9
Service: Telnet 9
Service: Data Socket 9
Service: Webserver 10
XML Web Service 12
ETH Standard 16
DHCP Client 17
Static IP Address 18
Subnet Mask 19
Gateway Address 20
DNS1 Address 21
DNS2 Address 22
Device Name 23
Data Socket Port 24
Web Server Security 25
Web Server Password 26

ETHERNET/IP™: INDUSTRIAL PROTOCOL 27
ETHERNET/IP™ Overview 28
BACKGROUND 28
ETHERNET/IP™ 28
CIP OVERVIEW 29
REQUIRED OBJECTS 29
APPLICATION OBJECTS 29
VENDOR SPECIFIC OBJECTS 30
USER CHALLENGES 30
ETHERNET/IP™ LED Standard 31
ETHERNET/IP™ Object Model 31
Modbus TCP Mapping 38
Fragmentation Example 39
Configuring a CompactLogix to Communicate via ETHERNET/IP™ 40

STANDARD DEFAULTS 47

KEYPAD 49

Chapter 1
Introduction

About this Manual

This Ethernet Software Reference Guide (SRG) is provided for users seeking advanced technical information related to the BC9180™ Ethernet Base Station/Charger, including connection, programming, maintenance and specifications. The Quick Reference Guide (QRG) or other publications associated with this product are downloadable free of charge from the website listed on the back cover of this manual.

Manual Conventions

The following conventions are used in this document:

The symbols listed below are used in this manual to notify the reader of key issues or procedures that must be observed when using the BC9180™ base station/charger:

NOTE

Notes contain information necessary for properly diagnosing, repairing and operating the BC9180™ base station/charger.

CAUTION

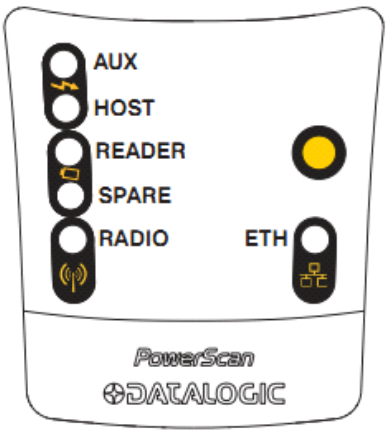
The CAUTION symbol advises you of actions that could damage equipment or property.

The BC9180™ Base Station/Charger

The BC9180™ base station, when paired with one or more PowerScan™ 9X00 readers, builds a Cordless Reading System for the collection, decoding and transmission of bar code data. It can be connected to a Host PC via Ethernet Host Interface. The BC91x0 models also provide a spare battery charger slot.

The label on the cradle contains LED indicators and a multi-function button. When the button is pressed for less than 5 seconds, the cradle will transmit a "broadcast" message." When the broadcast is sent, all properly configured scanners that are linked to that base and within radio range coverage will emit a beep and blink within 5 seconds. This functionality is useful for:

- Verifying which scanners are linked to a certain base station
- Paging (detect the position of linked scanners)



The LEDs signal the BC9180 status, as shown in Table 1.

Table 1. LED Status

LED	STATUS
Aux	Yellow On = BC9180 is powered through an external power supply.
Host	Yellow On = BC9180 is powered by the Host.
Reader	Green On = the reader battery is completely charged. Red On = the reader battery is charging. Red / Green Alternatively Blinking = charging error. Off = reader not in the cradle or not properly inserted.
Spare	Green On = the spare battery is completely charged. Red/Green Alternatively Blinking = charging error. Off = no spare battery in the housing or battery not fully inserted.
Radio	Yellow Blinking = radio activity.
Ethernet	Green Blinking = Ethernet activity.

Datalogic Aladdin™

Datalogic Aladdin™ is a multi-platform utility program providing a quick and user-friendly configuration method via the Ethernet interface. Aladdin allows you to program the cradle and the reader by selecting configuration commands through a user-friendly graphical interface running on a PC. These commands are sent to the device over the selected communication interface, or they can be printed as bar codes to be scanned. Aladdin also facilitates image capturing.

In addition, Aladdin makes it easy to upgrade the cradle's and handheld's firmware, to attain the benefits of new reader features. Reference the Datalogic Aladdin™ Online Help for more details.

Aladdin is available for download free of charge on the Datalogic website.

Technical Support

Datalogic Website Support

The Datalogic website (www.datalogic.com) is the complete source for technical support and information for Datalogic products. The site offers product support, warranty information, product manuals, product tech notes, software updates, demos, and instructions for returning products for repair.

Reseller Technical Support

An excellent source for technical assistance and information is an authorized Datalogic reseller. A reseller is acquainted with specific types of businesses, application software, and computer systems and can provide individualized assistance.

Telephone Technical Support

If you do not have internet or email access, you may contact Datalogic technical support at (541) 349-8283 or check the back cover of your manual for more contact information.

NOTES

Chapter 2
Setup

BC9180™ Startup and Connection to Ethernet Network

Ensure the BC9180 is correctly plugged into your system (see the BC91XX Quick Reference Guide or Product Reference Guide for information. Power on the BC9180™ cradle.

When the cradle powers up, the Ethernet Green LED will be OFF. When the Ethernet cable is plugged in, the cradle will attempt to connect to the network. Upon a successful connection, the Ethernet Green LED will be ON and the cradle will be ready to work as a receiver for the wireless HandHeld scanner linked to it.

When the DHCP Client configuration is enabled (default), the BC9180™ cradle will use the dynamic IP Address supplied by the DHCP server in the network. If the DHCP Client configuration is disabled, you can use a fixed static IP address (the default value is 192.168.187.31). To disable the DHCP client, go to page 17. To change the default fixed IP address, see page 18.

After connecting the BC9180™ to the network, Datalogic Aladdin™ software (version 1.6.3 or newer) can be used to discover the IP address of the Base using the Discovery feature.

Figure 1. Datalogic Aladdin™ Startup Screen

BC9180 Configuration

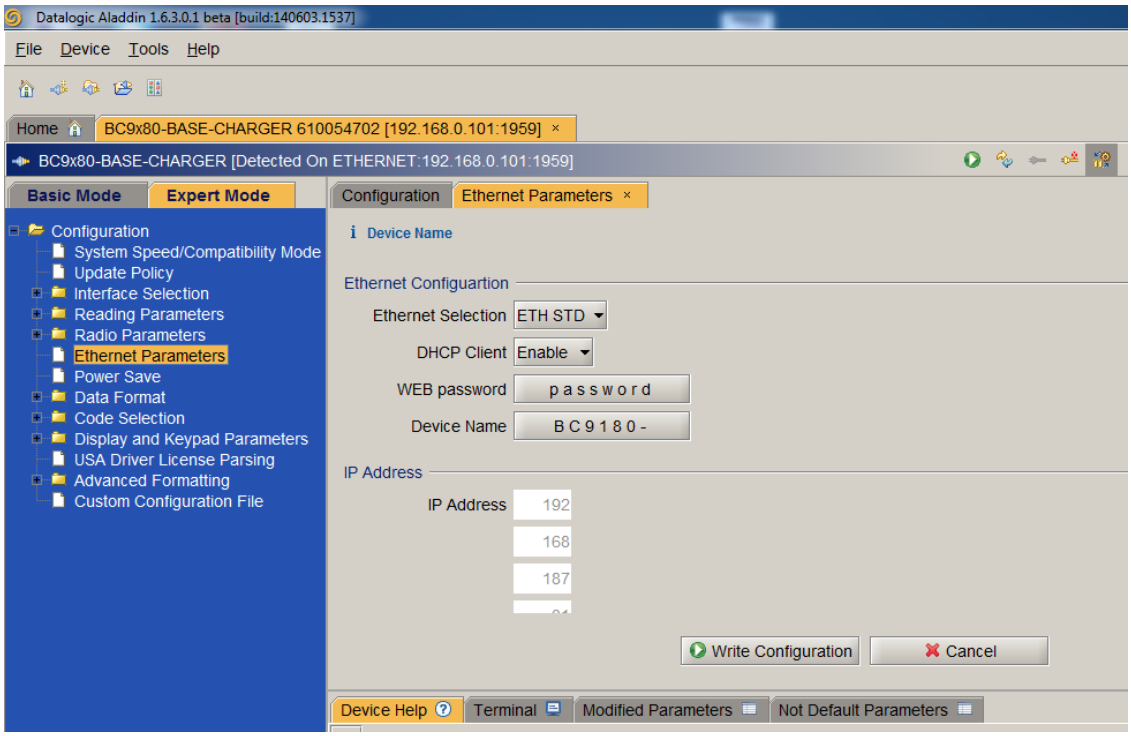
The BC9180 configuration can be performed in several ways, by using:

- Datalogic Aladdin software
- Configuration Bar Codes
- Webserver Configuration

Datalogic Aladdin™ software

After connection, the BC9180 can be configured by sending configuration strings from the Datalogic Aladdin™ via the Ethernet interface.

Figure 2. Datalogic Aladdin™ Configuration Screen



Configuration Bar Codes

Link the cradle and the reader using the procedures described in the PowerScan™ PM9X00 or PBT9X00 Quick Reference Guide (QRG). Once the pairing is complete, you can configure the BC9xx0 cradle by scanning configuration bar codes from this manual. See Ethernet Host Interface Parameters Only, starting on page 15.

To configure the BC9180 using the PowerScan™ 9X00 reader (paired to the cradle with the Bind command), follow the procedure for the interface selected.

Webserver Configuration

After successful connection to the Ethernet network, you can configure the BC9180™ cradle parameter by using any commercial Web Browser accessing the URL: http://<BC9180™_IP_Address>.

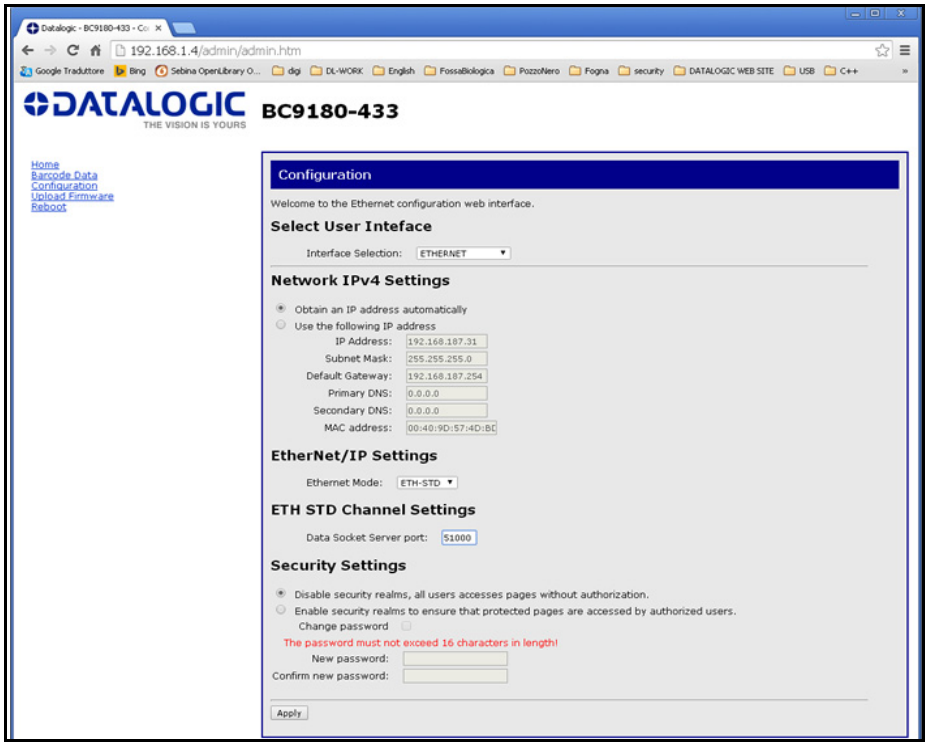
Then select Configuration from the menu on the left.

NOTE

You can also use Aladdin software to obtain the IP address of the cradle after the cradle is connected to the network.

Web Server configuration is always available when the BC9180™ is connected to the network, REGARDLESS of the interface selection.

Figure 3. Webserver configuration screen



After selecting the desired parameter, click Apply to finish the configuration. The cradle will be reset and then start up with the new configured parameter.

Selecting a Non-Ethernet interface

Selecting a non-Ethernet interface should not be needed in any Ethernet application, but if you want do it for service purposes, download the Quick Reference Guide (QRG) from the Datalogic website.

To restore the Ethernet interface, scan the bar code below.

NOTE

Unlike some programming features and options, interface selections require that you scan only one programming bar code label. DO NOT scan an ENTER/EXIT bar code prior to scanning an interface selection bar code.

ETHERNET		FEATURES
Ethernet Host interface	Select Ethernet Standard Interface	Set Ethernet Standard Features starting on page 16

For other interfaces, you can download the Product Reference Guide (PRG) or Quick Reference Guide (QRG) on the Datalogic website: www.datalogic.com.

NOTE

The default interface for the BC9180™ base station/charger is Ethernet Standard.

NOTES

Chapter 3
Ethernet Host Interface Operation

When Ethernet Host Interface is selected as the active interface and the system startup is completed, a label received from the Handheld wireless reader will be transmitted to the Host on multiple TCP/IP-based services concurrently. This will occur **only to the ones having** at least one client connected to them in order to preserve resources (see below).

Service: Telnet

Telnet is a bidirectional interactive text-oriented communication protocol. The host can receive label data with WINTERM (or other TELNET-ready application) and send Datalogic Service Port Commands/ Host Commands.

NOTE

BC9180™ only supports 1 Telnet client running at the same time.

Service: Data Socket

Data Socket is a point-to-point bidirectional communication channel available for Ethernet communication; it is the BC9180™ internal instance of a standard TCP-IP socket.

NOTE

BC9180™ only supports 1 Data Socket client running at the same time.

Service: Webserver

The Host can receive a dynamic page that is updated with all newly received labels from the BC9180™ by using any commercial Web Browser and accessing it at the URL:

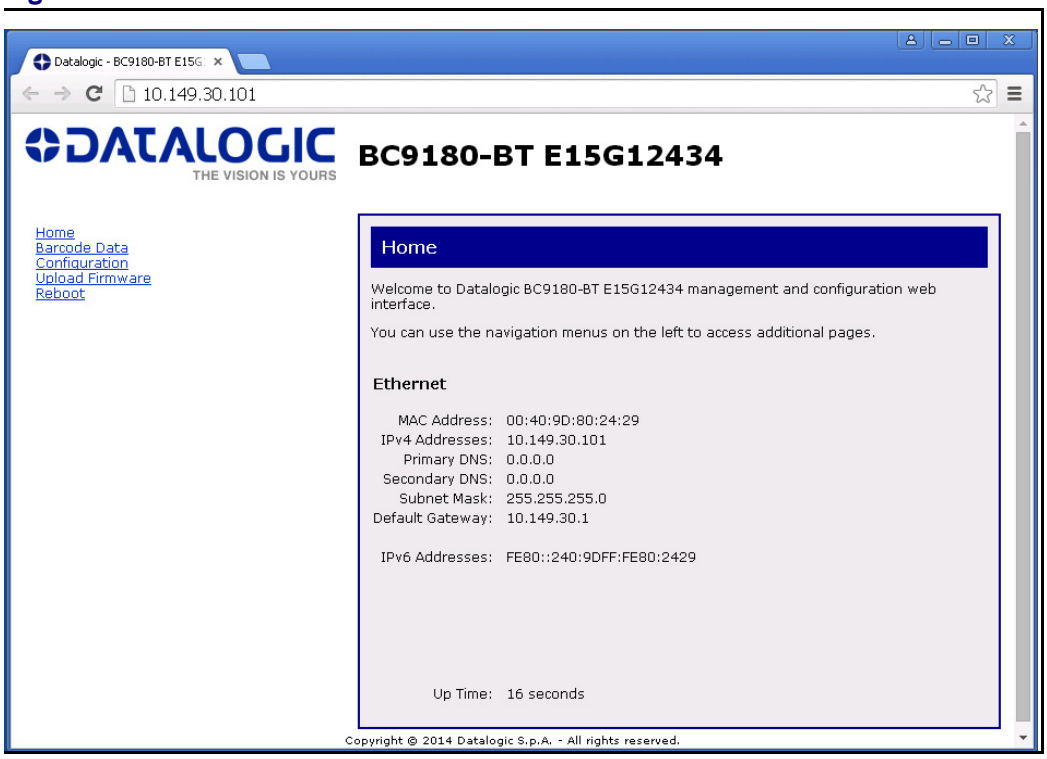
http://<BC9180™_IP_Address>

NOTE

You can also use Aladdin software to obtain the IP address of the cradle, after the cradle is connected to the network.

The BC9180™ cradle can support multiple clients accessing the Web Server at the same time.

Figure 4. Webserver Service



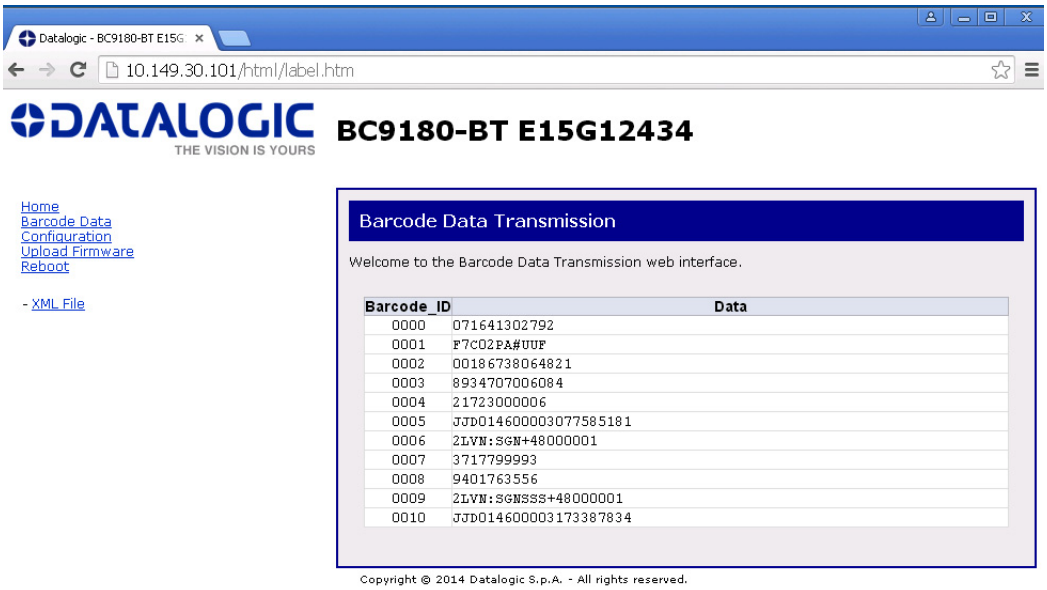
Example 1.

BC9180™ is connected to the network with IP address 10.149.30.101.

1. Select **Barcode Data** from the left menu.

All the transmitted labels will be shown in a table format with two column fields.

- **Data** – contains the received barcode data
- **Barcode_ID** – a progressive counter updated at every barcode data transmission



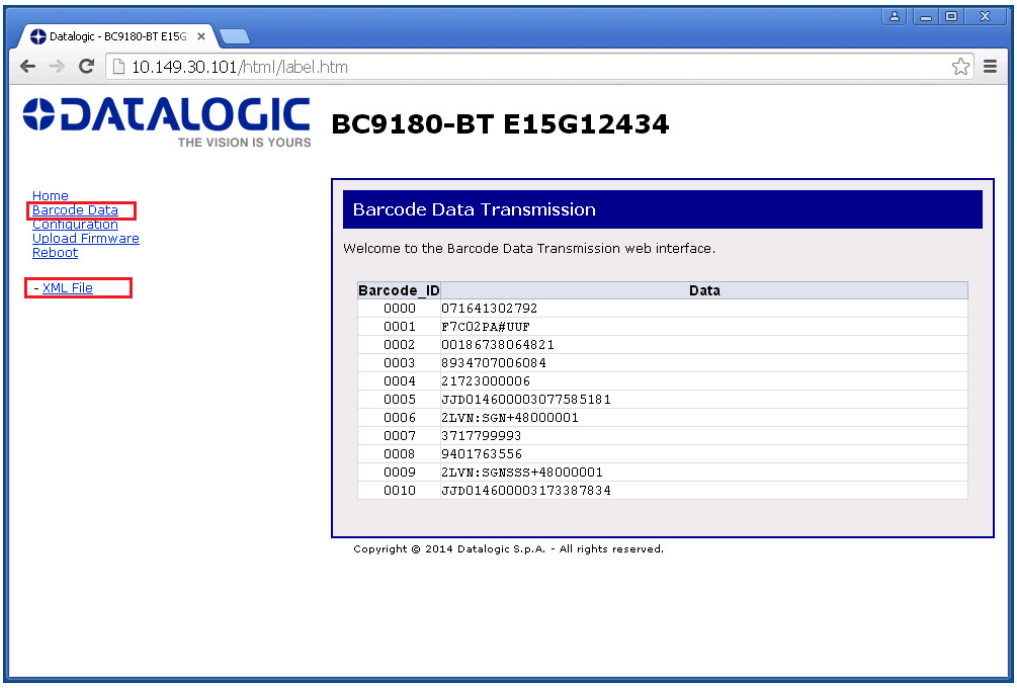
2. When the connection is lost, a warning message is shown “There was a problem retrieving the data... try later”.

XML Web Service

The BC9180™ also offers the possibility of accessing data in a simplified way for a Machine-to -Machine communication using an XML-based data representation.All labels transmitted to Host can be accessed at the URL:

http://Cradle_IP_Address>/xml_file.xml

or select **XML** from the menu on the left, as shown below:



The XML file has the format as shown below:

```
<Datalogic><CH1>Barcode_ID</
CH1><CH2>Data</CH2>
<Label_>
      <counter>0000</counter>
      <data>Data0 </data>
</Label>_
<Label_>
      <counter>0001</counter>
      <data>Data1 </data>
</Label>_
...
</Datalogic>
```

The XML file will contain the data for a maximum of 50 barcode labels with progressive counters; it is up to the Client to timely poll the BC9180™ in order to prevent data loss. In order to provide a Client-controllable data flow, an XML file at the maximum 50-barcode labels is only upgraded using a rolling buffering scheme, copying forward the last 10 labels received to the newly updated file.

The Barcode ID counters should be checked from the Client to avoid checking in duplicated data.

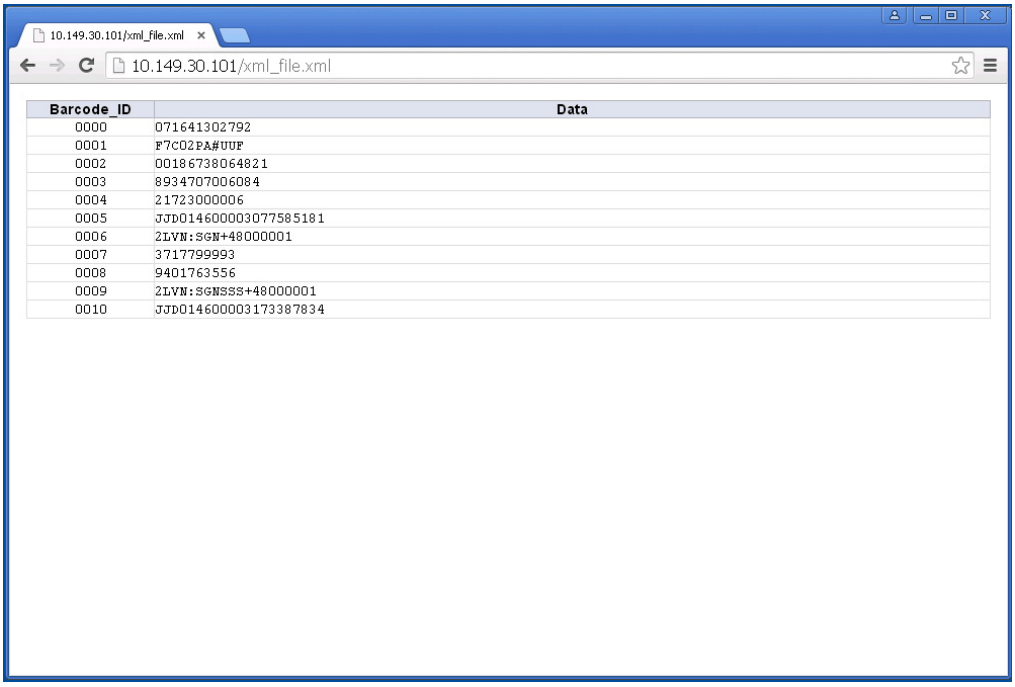
Poll request	XML file content	Notes
Poll #a	$Data_{m0}, \ldots, Data_{m48}$	Starting 49 labels at time of Poll #a
Poll #a+1	$Data_{m0}, \ldots, Data_{m48}$	No new data before Poll #a+1
…		
Poll #b	$Data_{m0}, \ldots, Data_{m48}, Data_{m49}$	50 labels in the XML file
Poll #b+1	$Data_{m40}, \ldots, Data_{m49}, Data_{m50}, Data_{m51}$	XML file updated: last 10 labels rolled in the newly updated file and labels number 50 and 51 added

Example 2.

BC9180™ connected to the network with IP address is 10.149.30.101

Even if this kind of service is set up for M2M communication, it is possible to visualize the XML data formatted using a simple CSS, by opening in a browser and pointing to the aforementioned URL.

The simplified XML file will display as shown below:



Barcode_ID	Data
0000	071641302792
0001	F7C02PA#UUF
0002	00186738064821
0003	89347070006084
0004	21723000006
0005	JJD014600003077585181
0006	2LVN:SGN+48000001
0007	3717799993
0008	9401763556
0009	2LVN:SGNSSS+48000001
0010	JJD014600003173387834

NOTES

Ethernet Host Interface Parameters Only

ETH STANDARD on page 16
DHCP CLIENT on page 17
STATIC IP ADDRESS on page 18
SUBNET MASK on page 19
GATEWAY ADDRESS on page 20
DNS1 ADDRESS on page 21
DNS2 ADDRESS on page 22
DEVICE NAME on page 23
DATA SOCKET PORT on page 24
WEB SERVER SECURITY on page 25
WEB SERVER PASSWORD on page 26

This section contains explanations of selected Ethernet Host Interface Parameters. Refer to Appendix A, Standard Defaults for initial configuration when the cradle is set with the default values.

ETH Standard

Configure the Ethernet interface as a Standard Ethernet or an Industrial ETHERNET/IP™ protocol or Industrial Ethernet using Modbus™ protocol

	Ethernet network = ETHERNET/IP™
Ethernet network = Standard Ethernet	DEFAULT
	Ethernet network = Ethernet Modbus™

DHCP Client

When DHCP Client is Enabled, the Ethernet Module dynamically obtains an Internet Protocol (IP) address from a Dynamic Host Configuration Protocol (DHCP) server. DHCP server also provides Subnet Masks, Gateway address and network addresses of DNS servers.

When DHCP Client is Disabled, the Ethernet Module uses a manually specified Internet Protocol (IP) address (also known as a static IP address). You can use the default address provided, or you can use the configuration items that follow to specify an IP address in Static IP address, a subnet mask in Subnet mask, a gateway address in Gateway address and the network addresses of DNS servers.

	DHCP Client = Disable
DHCP Client = Enable	DEFAULT

Static IP Address

This feature sets a Static Internet Protocol (IP) address for the BC9180™. An IP address is a 32-bit number that is notated by using four numbers from 0 through 255, separated by periods. For example: 192.168.1.100

To add the Cradle to an existing LAN you must specify a unique IP address that is not used elsewhere in the network.

Set Static IP Address	To configure this feature, scan the ENTER/EXIT bar code above, then the bar code at left followed by 8 digits (in hex) from the Alphanumeric characters in Appendix B, Keypad representing your desired character(s). Exit programming mode by scanning the ENTER/EXIT barcode again.
Make a mistake? Scan the CANCEL bar code to abort and not save the entry string. You can then start again at the beginning.	CANCEL

DEFAULT

Default IP Address = C0A8BB1F, hexadecimal representation of 192.168.187.31

Subnet Mask

The subnet mask number is combined with the IP address number to identify which network segment the BC9180™ is on.

A subnet mask is a 32-bit number that is composed by using four numbers from 0 through 255, separated by periods. Typically, default subnet mask numbers use either 0 or 255 as values (such as 255.255.255.0), but other number values can appear.

To connect the Cradle to an existing LAN, specify the Subnet Mask that the LAN uses.

Set Subnet mask	To configure this feature, scan the ENTER/EXIT bar code above, then the bar code at left followed by 8 digits (in hex) from the Alphanumeric characters in Appendix B, Keypad representing your desired character(s). Exit programming mode by scanning the ENTER/EXIT barcode again.
Make a mistake? Scan the CANCEL bar code to abort and not save the entry string. You can then start again at the beginning.	CANCEL

DEFAULT

Default Subnet mask = FFFFFF00, hexadecimal representation of 255.255.255.0

Gateway Address

A gateway is a router that connects separate IP network segments. For example, a network segment might need a gateway to connect it to another network segment, a wide area network (WAN), or to the Internet.

This feature specifies the address of a local IP router on the same network as the BC9180™, used to forward traffic to destinations beyond the local network. The value in each field must be a number from 0 through 255.

Set Gateway address	To configure this feature, scan the ENTER/EXIT bar code above, then the bar code at left followed by 8 digits (in hex) from the Alphanumeric characters in Appendix B, Keypad representing your desired character(s). Exit programming mode by scanning the ENTER/EXIT barcode again.
Make a mistake? Scan the CANCEL bar code to abort and not save the entry string. You can then start again at the beginning.	CANCEL

DEFAULT

Default Gateway address = C0A8BBFE, hexadecimal representation of 192.168.187.254

DNS1 Address

This is the address of the preferred or primary DNS server for the BC9180™. This server is used first, to resolve DNS names to IP addresses for DNS names queried by this computer that cannot be resolved by using local name resolution information (such as cached DNS names or names contained in a Hosts file).

Set DNS1 address	To configure this feature, scan the ENTER/EXIT bar code above, then the bar code at left followed by 8 digits (in hex) from the Alphanumeric characters in Appendix B, Keypad representing your desired character(s). Exit programming mode by scanning the ENTER/EXIT barcode again.
Make a mistake? Scan the CANCEL bar code to abort and not save the entry string. You can then start again at the beginning.	CANCEL

DEFAULT

Default DNS1 address = 00000000, hexadecimal representation of 0.0.0.0

DNS2 Address

This is the address of the alternate or secondary DNS server for BC9180™. This server is used if the DNS server specified as the preferred DNS server (DNS1) is unreachable or cannot resolve DNS names to IP addresses for DNS names queried by this computer.

Set DNS2 address	To configure this feature, scan the ENTER/EXIT bar code above, then the bar code at left followed by 8 digits (in hex) from the Alphanumeric characters in Appendix B, Keypad representing your desired character(s). Exit programming mode by scanning the ENTER/EXIT barcode again.
Make a mistake? Scan the CANCEL bar code to abort and not save the entry string. You can then start again at the beginning.	CANCEL

DEFAULT

Default DNS2 address = 00000000, hexadecimal representation of 0.0.0.0

Device Name

This parameter is used to uniquely identify one BC9180™ from other Datalogic devices during the Product Discovery service initiated by Aladdin. This feature will identify a Hostname and discovery name to correspond to the address of a device connected to a network. .

Set Device Name	To configure this feature, scan the ENTER/EXIT bar code above, then the bar code at left followed by the digits (in hex) from the Alphanumeric characters in Appendix B, Keypad representing your desired character(s). If less than the expected string of 32 characters is selected, scan the ENTER/ EXIT bar code to terminate the string. Exit programming mode by scanning the ENTER/EXIT barcode again.
Make a mistake? Scan the CANCEL bar code to abort and not save the entry string. You can then start again at the beginning.	CANCEL

DEFAULT

BC9180-433(910)(BT) [BASE_SERIAL_NUMBER]

Data Socket Port

This parameter will allow the identification of a network service on an IP network (the Internet). It is a mapping to the underlying TCP–IP socket port number.

Set Data Socket Port	To configure this feature, scan the ENTER/EXIT bar code above, then the bar code at left followed by 5 digits from the Alphanumeric characters in Appendix B, Keypad representing your desired character(s). Exit programming mode by scanning the ENTER/EXIT barcode again.
Make a mistake? Scan the CANCEL bar code to abort and not save the entry string. You can then start again at the beginning.	CANCEL

DEFAULT

Default Data Socket Port = 51000

NOTE

Port numbers 59180, 00080, 00023 are reserved for the Datalogic Aladdin™ configuration tool and other default services ports. It is strongly recommend not to use these numbers for the Data Socket network service.

Web Server Security

Enabling this feature will apply the root account security used when connected to the Web Server embedded into the BC9180™. This account has all privileges and can be used to protect the login for key services such as Configuration, Firmware Upgrade or Forcing Reboot.

DEFAULT	Web Server security = Disable
Web Server security = Enable	

Web Server Password

This feature will allow you to specify the root account password to be used when Web Server Security is enabled. This account has all privileges and can be used to protect the login for key services such as Configuration, Firmware Upgrade or Forcing Reboot.

Set Web Password	To configure this feature, scan the ENTER/EXIT bar code above, then the bar code at left followed by 32 digits (in hex) from the Alphanumeric characters in Appendix B, Keypad representing your desired character(s). Exit programming mode by scanning the ENTER/EXIT barcode again.
Make a mistake? Scan the CANCEL bar code to abort and not save the entry string. You can then start again at the beginning.	CANCEL

DEFAULT

Default Web Password = password

Chapter 5
ETHERNET/IP™: Industrial Protocol

This section contains explanations and examples of selected bar code features. See Ethernet Host Interface Operation, starting on page 9 for the actual bar code labels used to configure the reader.

ETHERNET/IP™ OVERVIEW on page 28
- BACKGROUND on page 28
- ETHERNET/IP™ on page 28

CIP OVERVIEW on page 29
- REQUIRED OBJECTS on page 29
- APPLICATION OBJECTS on page 29
- VENDOR SPECIFIC OBJECTS on page 30
- USER CHALLENGES on page 30

ETHERNET/IP™ LED STANDARD on page 31

ETHERNET/IP™ OBJECT MODEL on page 31
- Identity Object (01HEX – 1 Instance) on page 32
- Message Router Object (02HEX – 1 Instance) on page 32
- Assembly Object (04HEX – 4 Instances) on page 33
- Connection Manager Object (06HEX) on page 34
- TCP Object (F5HEX – 1 Instance) on page 35
- Ethernet Link Object (ETHERNET/IP™ only) (F6HEX – 1 Instance) on page 36
- Barcode Scanner Object (64HEX – 1 Instance) on page 37
- Modbus TCP Mapping on page 38
- Fragmentation Example on page 39
- Configuring a CompactLogix to Communicate via ETHERNET/IP™ on page 40

ETHERNET/IP™ Overview

BACKGROUND

Most people who work in an office associate the term “Ethernet” with the physical cable behind their desk. This cable connects their office PC to the printers and servers of the local network and the infinite websites on the Internet. This cable is only the physical part of Ethernet, the media carrying Ethernet messages to your PC. On this wire there is a whole series of communication protocols such as IP, the Internet Protocol; TCP, the Transport Control Protocol; and various Microsoft protocols such as NetBEUI. This suite of protocols works well for the office environment. It allows users to share files, access printers, send email, search the Internet and perform all the other communications used in the office environment.

The needs of the factory floor are much different, with some very special requirements. Instead of accessing files and printers, factory floor controllers must access data embedded in drive systems, operator workstations and I/O devices. Instead of making a user wait while a task is being performed, factory floor data communications needs are real-time or very close to real time. Terminating the fill operation on a bottle requires much more time-precise communications than accessing the next page of an Internet site.

Traditionally, Ethernet had only limited acceptance in Industrial Automation. Until recently the expense, lack of intelligent switches and routers and the domination of large vendors with proprietary protocols prevented the wide acceptance of Ethernet on the factory floor. Now with prices falling, PCs with inherent Ethernet capability moving in droves onto the factory floor and intelligent switches and routers, Ethernet is gaining acceptance. Only the lack of a widely accepted, flexible application layer targeted to Industrial Automation has prevented its complete acceptance.

ETHERNET/IP™

ETHERNET/IP™ is the application layer protocol that can meet this challenge. Four independent groups have joined forces to develop and promote EIP as a public domain Ethernet application layer for Industrial Automation. These groups include the Open DeviceNet Vendor Association (ODVA), the Industrial Open Ethernet Association (IOANA), Control Net International (CI) and the Industrial Ethernet Association (IEA).

The goals of this effort illustrate how EIP provides a wide-ranging, comprehensive, certifiable standard suitable to a wide variety of automation devices: ETHERNET/IP™ uses all the transport and control protocols used in traditional Ethernet including the Transport Control Protocol (TCP), the Internet Protocol (IP) and the media access and signalling technologies found in off-the-shelf Ethernet interface cards.

Building on these standard PC technologies means that EIP works transparently with all the standard off-the-shelf Ethernet devices found in today’s marketplace. It also means that EIP can be easily supported on standard PCs and all their derivatives. Even more importantly, basing EIP on a standard technology platform ensures that EIP will move forward as the base technologies evolve in the future.

CIP OVERVIEW

The Communications and Information Protocol (CIP) is a communications protocol for transferring automation data between two devices. 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. For example, every CIP device is required to make an Identity object available to the network. 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. CIP does not specify at all how this object data is implemented, only what data values or attributes must be supported and that these attributes must be available to other CIP devices.

The Identity object is an example of a required object. There are three types of objects defined by the CIP protocol:

REQUIRED OBJECTS

Required objects are required by the specification to be included in every CIP device. These objects include the Identity object, a Message Router object and a Network object.

The identity object contains related identity data values called attributes. Attributes for the identity object include the vendor ID, date of manufacturer, device serial number and other identity data.

A Network object contains the physical connection data for the object. For a CIP device on DeviceNet the network object contains the MacID and other data describing the interface to the CAN network. For EIP devices, the network object contains the IP address and other data describing the interface to the Ethernet port on the device.

APPLICATION OBJECTS

Application objects are the objects that define the data encapsulated by the device. These objects are specific to the device type and function. For example, a Motor object on a Drive System has attributes describing the frequency, current rating and motor size. An Analog Input object on an I/O device has attributes that define the type, resolution and current value for the analog input.

These application layer objects are predefined for a large number of common device types. All CIP devices with the same device type (Drive Systems, Motion Control, Valve Transducer…etc.) must contain the identical series of application objects. The series of application objects for a particular device type is known as the device profile. A large number of profiles for many device types have been defined. Supporting a device profile allows a user to easily understand and switch from a vendor of one device type to another vendor with that same device type.

A device vendor can also group Application Layer Objects into assembly objects. These super objects contain attributes of one or more Application Layer Objects. Assembly objects form a convenient package for transporting data between devices. For example, a vendor of a Temperature Controller with multiple temperature loops may define assemblies for each of the temperature loops and an assembly with data from both temperature loops. The user can than pick the assembly that is most suited for the application and how often to access each assembly.

For example, one temperature assembly may be configured to report every time it changes state while the second may be configured to report every one-second regardless of a change in state.

Assemblies are usually predefined by the vendor but CIP also defines a mechanism in which the user can dynamically create an assembly from application layer object attributes.

VENDOR SPECIFIC OBJECTS

Objects not found in the profile for a device class are termed Vendor Specific. These objects are included by the vendor as additional features of the device. The CIP protocol provides access to these vendor extension objects in exactly the same method as either application or required objects. This data is strictly of the vendors choosing and is organized in whatever method makes sense to the device vendor.

In addition to specifying how device data is represented to the network, the CIP protocol specifies a number of different ways in which that data can be accessed such as cyclic, polled and change-of-state.

USER CHALLENGES

EIP implementation is not without challenges. Two of the most important challenges to the first time user include training and network configuration. One common problem is the lack of trained staff who understands both the IT fundamentals and the automation network. A collaborative effort between the IT and Automation staffs is required to successfully implement the first ETHERNET/IP™ system. A second challenge is proper network configuration. Planning your Ethernet factory automation infrastructure is essential. Careful identification of all your control loops, choosing the correct routers, switches and paths and documenting your network properly are requisites for a communications network which meets your production goals and requires little ongoing maintenance.

ETHERNET/IP™ LED Standard

Table 2 below shows the LEDs supported by ETHERNET/IP™.

Table 2. ETHERNET/IP™ supported LEDs

LED	Color	State	Indicates
Module Status	None	Off	No Power
	Red	Solid	Unrecoverable fault
		Flashing	Recoverable fault
	Green	Solid	Normal runtime operation
		Flashing	N/A
	Red/Green	Alternating	Self test
Network Status	None	Off	No power / No Ethernet Link
	Red	Solid	Unrecoverable fault
		Flashing	Recoverable fault or I/O connection timed out
	Green	Solid	Normal runtime operation (I/O Connection Allocated)
		Flashing	Device is idle or is not allocated to a Client (PLC)
	Red/Green	Alternating	Self test

ETHERNET/IP™ Object Model

Table 3 describes data types used in this Object Model.

Table 3. Object Model data types

Data Type	Description
USINT	Unsigned Short Integer (8-bit)
UINT	Unsigned Integer (16-bit)
UDINT	Unsigned Double Integer (32-bit)
SINT	Signed Integer (8-bit)
INT	Signed Integer (16-bit)
DINT	Signed Integer (32-bit)
STRING	Character String (1 byte per character)
SHORT STRING*nn*	Character String (1st byte is length; up to *nn* characters)
STRINGI	International String format
BYTE	Bit String (8-bits)
WORD	Bit String (16-bits)
DWORD	Bit String (32-bits)
REAL	IEEE 32-bit Single Precision Floating Point

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

The following tables contain the attribute, status, and common services information for the Identity Object.

Table 4.Identity Object (01_{HEX} - 1 Instance)

Instance	Attribute ID	Name	CIP Data Type	Data Value	Access Rule
Class (Instance 0)	1	Revision	UINT	1	Get
Instance 1	1	Vendor number	UINT	850	Get
	2	Device type	UINT	43	Get
	3	Product code number	UINT	9X00	Get
	4	Product major revision Product minor revision	USINT USINT	1.01	Get
	5	Status	WORD	See	Get
	6	Serial number	UDINT	Unique 32 bit value (Last 4 of MAC ID)	Get
	7	Product name	SHORT STRING32	Wireless Barcode Reader	Get
	100	Product model number	SHORT STRING32	“Product Model Number”	Get

Table 5. Identity Object’s common services

Service code	Implemented for		Service name
	Class level	Instance level	
05_{Hex}	No	Yes	Reset
$0E_{Hex}$	Yes	Yes	Get_Attribute_Single
10_{Hex}	No	Yes	Set_Attribute_Single

Message Router Object (02HEX - 1 Instance)

No supported services or attributes

Assembly Object (04HEX - 4 Instances)

The following tables contain the attribute, instance, data mapping, and common services information for the Assembly Object.

Table 6. Assembly Object (04_{HEX} - 2 Instances)

Instance	Attribute ID	Name	CIP Data Type	Data Value	Access Rule
Class (Instance 0)	1	Revision	UINT	2	Get
	2	Max instance	UINT	129	Get
100 (0x64)	3	T2O (Input) Assembly (see below)	SINT[472]	Varies	Get
112 (0x70)	3	O2T (Output) Assembly (see below)	SINT[468]	Varies	Get
128 (0x80)	N/A	Input only heartbeat[1]	Heartbeat	0	n/a
129 (0x81)	N/A	Listen only heartbeat[2]	Heartbeat	0	n/a
Unused (n)	N/A	Configuration[3]			

1. This instance allows clients (PLCs) to monitor input data without providing output data.
2. This instance allows clients (PLCs) to monitor input data without providing output data. To use this connection type, an owning connection must exist from a second client and the configuration of the connection must match exactly.
3. Configuration data is not required, but it must match if supplied. Contents of the configuration instance are yet to be determined.

Table 7. T2O (Input) Assembly Data Format

Byte	Description	Data Type	Class ID	Inst ID	Attr ID
0 – 3	Discrete Inputs	DWORD	0x64	0x01	1
4 – 7	Barcode Scanner Status	DWORD	0x64	0x01	2
8 – 9	Item Sequence Number	UINT	0x64	0x01	3
10 – 11	Item Total Size	UINT	0x64	0x01	4
12 – 13	Fragment Sequence Number	UINT	0x64	0x01	5
14 – 15	Fragment Size	UINT	0x64	0x01	6
16 – 465	Fragment Data	USINT[450]	0x64	0x01	7
466 – 467	Reserved (for 32-bit alignment)	UINT	N/A	N/A	N/A
468 – 469	Last Output Item Sequence Number	UINT	0x64	0x01	15
470 – 471	Reserved (for 32-bit alignment)	UINT	N/A	N/A	N/A

Table 8. O2T (Output) Assembly Data Format

Byte	Description	Data Type	Class ID	Inst ID	Attr ID
0 – 3	Discrete Outputs	DWORD	0x64	0x01	8
4 – 5	Last Item Sequence Number	UINT	0x64	0x01	9
6 – 7	Last Fragment Sequence Number	UINT	0x64	0x01	10
8 – 11	Output Status	DWORD	0x64	0x01	11
12 – 13	Output Item Sequence Number	UINT	0x64	0x01	12
14 – 15	Output Item Total Size	UINT	0x64	0x01	13
16 – 465	Output Data	USINT[450]	0x64	0x01	14
466 – 467	Reserved (for 32-bit alignment)	UINT	N/A	N/A	N/A

Table 9. Assembly Object's common services

Service code	Implemented for		Service name
	Class level	Instance level	
$0E_{Hex}$	Yes	Yes	Get_Attribute_Single
10_{Hex}	Yes	Yes	Set_Attribute_Single

Connection Manager Object (06HEX)

No supported services or attributes

TCP Object (F5HEX - 1 Instance)

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

Table 10. TCP Object ($F5_{HEX}$ - 1 Instance)

Instance	Attribute ID	Name	Data Type	Data Value	Access Rule
Class (Instance 0)	1	Revision	UINT	1	Get
Instance 1	1	Status*	DWORD	Varies	Get
	2	Configuration capability*	DWORD	Varies	Get
	3	Configuration control*	DWORD	Varies	Get
	4	Physical Link Object * Structure of Path Size Path	 UINT Array of Word	 Varies Varies	Get
	5	Interface configuration* 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	 Varies Varies Varies Varies Varies Varies Varies	Get
	6	Host name* Structure of Host Name Size Host Name	 UINT STRING	 Varies Varies	Get

* For more details on these attributes, see *Volume 2: ETHERNET/IP™ Adaptation of CIP*, Section 5-3.2 from ODVA.

Table 11. TCP Object's 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

Ethernet Link Object (ETHERNET/IP™ only) ($F6_{HEX}$ - 1 Instance)

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

Table 12. Ethernet Link Object ($F6_{HEX}$ - 1 Instance)

Instance	Attribute ID	Name	Data Type	Data Value	Access Rule
Class (Instance 0)	1	Revision	UINT	1	Get
Instance 1	1	Interface speed*	UDINT	Varies	Get
	2	Interface flags*	DWORD	Varies	Get
	3	Physical address	USINT Array (6)	Varies	Get

* For more details on these attributes, see *Volume 2: ETHERNET/IP™ Adaptation of CIP*, Section 5-4.2 from ODVA.

Table 13. Ethernet Link Object's common services

Service code	Implemented for		Service name
	Class level	Instance level	
$0E_{Hex}$	Yes	Yes	Get_Attribute_Single

Barcode Scanner Object (64_{HEX} - 1 Instance)

The following tables contain the attribute and common services information for the Barcode Item Object.

Table 14. Unit Object (64_{HEX} - 1 Instance)

Instance	Attribute ID	Name	Data Type	Data Value	Access Rule
Class (Instance 0)	1	Revision	UINT	1	Get
	100	Max Item Data Buffer Size	UINT	65535	Get
	101	Max Fragment Data Buffer Size	UINT	450	Get
Instance 1	1	Discrete Inputs	DWORD	Varies	Get
	2	Barcode Scanner Status	DWORD	Varies	Get
	3	Item Sequence Number	UINT	Varies	Get
	4	Item Total Size	UINT	Varies	Get
	5	Fragment Sequence Number	UINT	Varies	Get
	6	Fragment Size	UINT	Varies	Get
	7	Fragment Data	USINT[450]	Varies	Get
	8	Discrete Outputs	DWORD	Varies	Get / Set
	9	Last Item Sequence Number	UINT	Varies	Get / Set
	10	Last Fragment Sequence Number	UINT	Varies	Get / Set
	11	Output Status	DWORD	Varies	Get / Set
	12	Output Item Sequence Number	UINT	Varies	Get / Set
	13	Output Item Total Size	UINT	Varies	Get / Set
	14	Output Data	USINT[450]	Varies	Get / Set
	15	Last Item Sequence Number	UINT	Varies	Get

Table 15. Barcode Scanner Object's 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

Modbus TCP Mapping

Holding Register (4x)	Description	Data Type	Class ID	Inst ID	Attr ID
1 – 2	Discrete Inputs	DWORD	0x64	0x01	1
3 – 4	Barcode Scanner Status	DWORD	0x64	0x01	2
5	Item Sequence Number	UINT	0x64	0x01	3
6	Item Total Size	UINT	0x64	0x01	4
7	Fragment Sequence Number	UINT	0x64	0x01	5
8	Fragment Size	UINT	0x64	0x01	6
9 – 233	Fragment Data	USINT[450]	0x64	0x01	7
234 – 235	Discrete Outputs	DWORD	0x64	0x01	8
236	Last Item Sequence Number	UINT	0x64	0x01	9
237	Last Fragment Sequence Num–ber	UINT	0x64	0x01	10
238 – 239	Output Status	DWORD	0x64	0x01	11
240	Output Item Sequence Number	UINT	0x64	0x01	12
241	Output Item Total Size	UINT	0x64	0x01	13
242 – 466	Output Data	USINT[450]	0x64	0x01	14
467	Last Output Item Sequence Number	UINT	0x64	0x01	15

Fragmentation Example

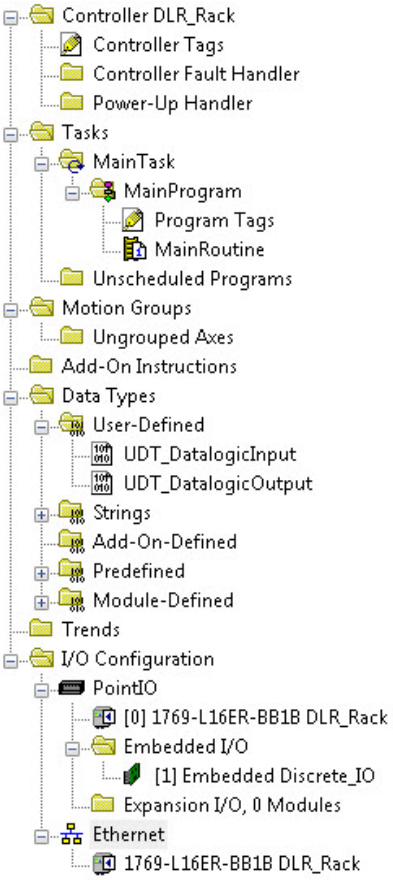
To Datalogic barcode scanner from EIP Client		To ETHERNET/IP™ Client from Datalogic Barcode Scanner					
Last Item Sequence Number	Last Fragment Sequence Number	Item Sequence Number	Fragment Sequence Number	Item Size	Fragment Size	Fragment Data Buffer	Description
0	0	0	0	0	0	NULL	Power Up
		1	1	2000	480	[0-479]	Datalogic sends fragment 1
0	1						EIP Client acknowledges fragment 1
		1	2	2000	480	[480-959]	Datalogic sends fragment 2
0	2						EIP Client acknowledges fragment 2
		1	3	2000	480	[960-1439]	Datalogic sends fragment 3
0	3						EIP Client acknowledges fragment 3
		1	4	2000	480	[1440-1919]	Datalogic sends fragment 4
0	4						EIP Client acknowledges fragment 4
		1	5	2000	80	[1920-1999]	Datalogic sends fragment 5
1	5						EIP Client acknowledges whole Item Data Buffer

Configuring a CompactLogix to Communicate via ETHERNET/IP™

This section is meant as a quick start for adding your gateway to a CompactLogix system. The process is applicable for all Rockwell PLCs that support I/O messaging to ETHERNET/IP™ Adapter devices.

This document assumes a certain level of knowledge of the Rockwell Automation environment and starts from a properly configured base project for your PLC. It also assumes you have a valid connection to the PLC (via RSLinx) to download the program.

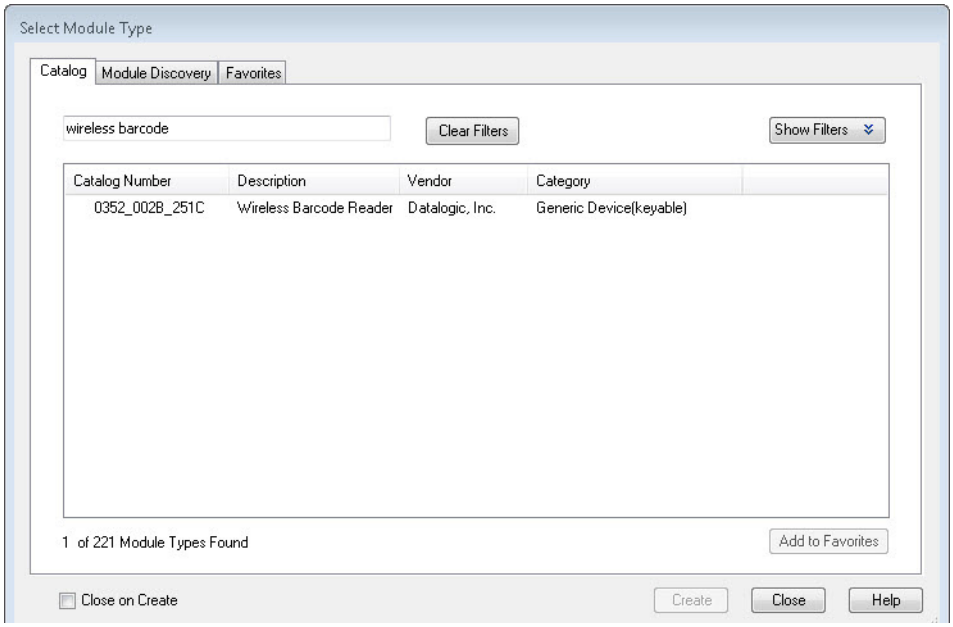
This document also assumes the EDS file was properly imported into RSLogix5000 using the EDS Hardware Installation Tool from Rockwell Automation.



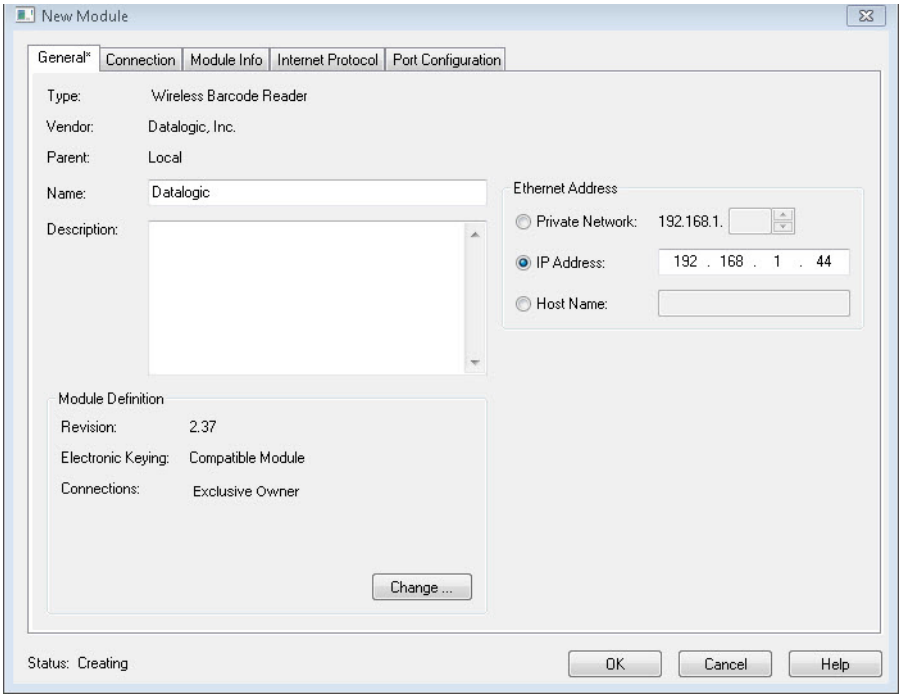
1. Browse to the I/O Configuration tree for your Ethernet network within RSLogix5000.

2. Right click on the “Ethernet” network and select “**New Module...**”.

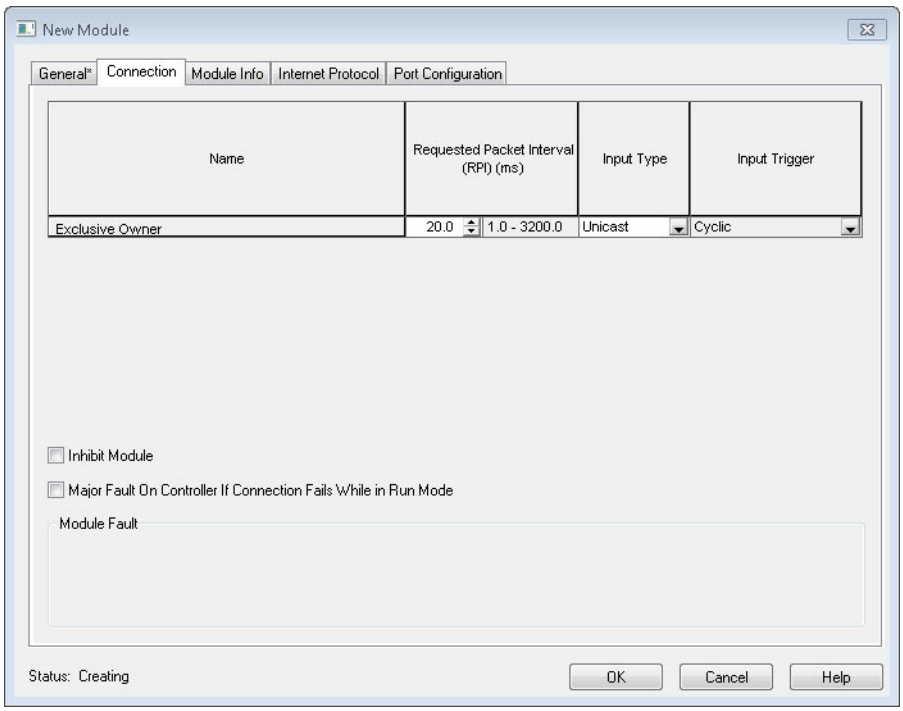
3. Type “**`wireless barcode`**” into the filter box. Double click on the entry.



4. Enter the configuration shown below. **Name** is the value used to add the tag in the controller variable list, and can be any value. Match the IP address to your barcode reader.

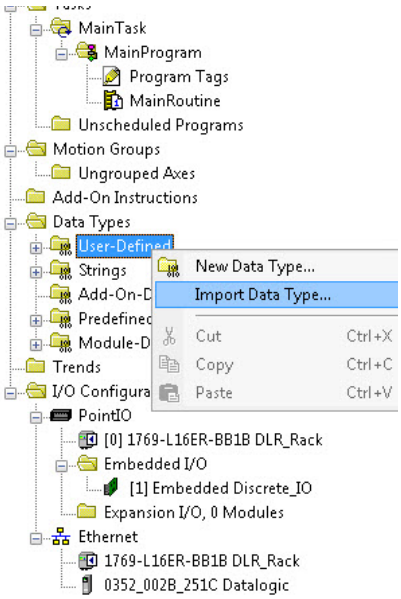


5. Select the **Connection** tab and change the RPI. There is usually no need to go faster than 20ms, since scanning a barcode has a certain amount of overhead for processing. The other settings should match the below settings. Press OK.

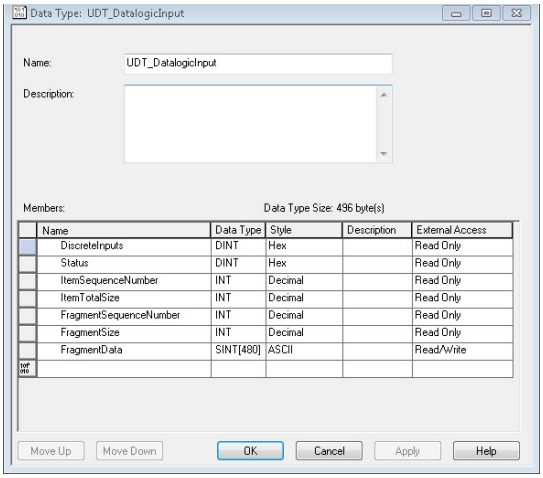


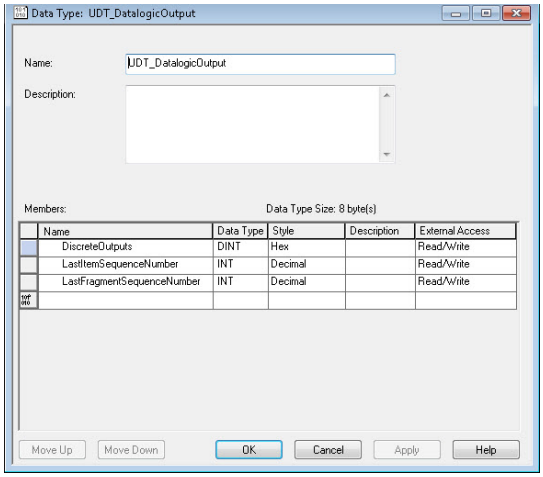
The default data to/from the barcode reader is a block of bytes. To make PLC coding easier, custom User Defined Types are provided.

1. Expand the Data Types folder and right click on the User Defined folder.
2. Select **Import Data Type...** and browse for the **UDT_DatalogicInput.L5X** and **UDT_DatalogicOutput.L5X** files provided.

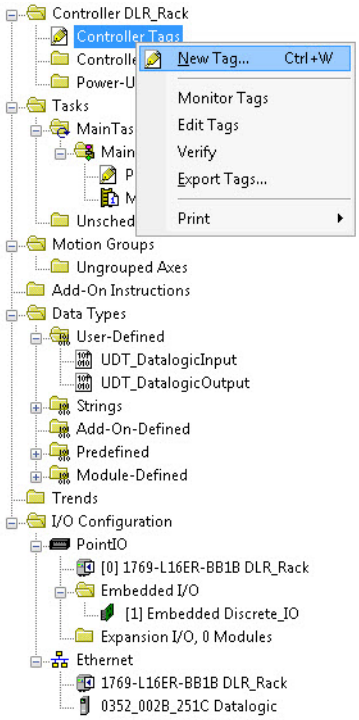


The new data types should look as follows:

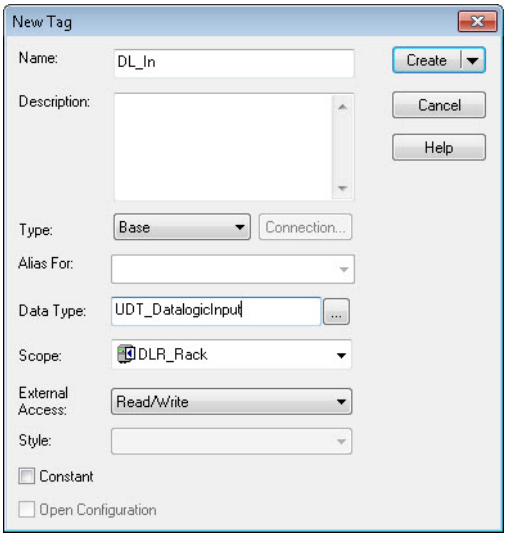


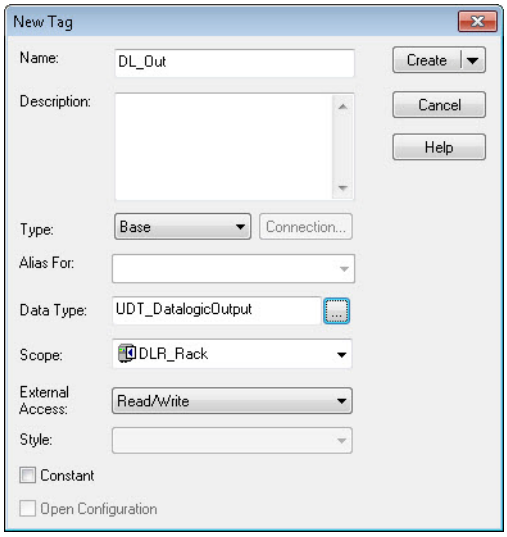


3. Next, create a new tag by right-clicking on the Controller Tags list in the Controller tree and selecting **New Tag...**

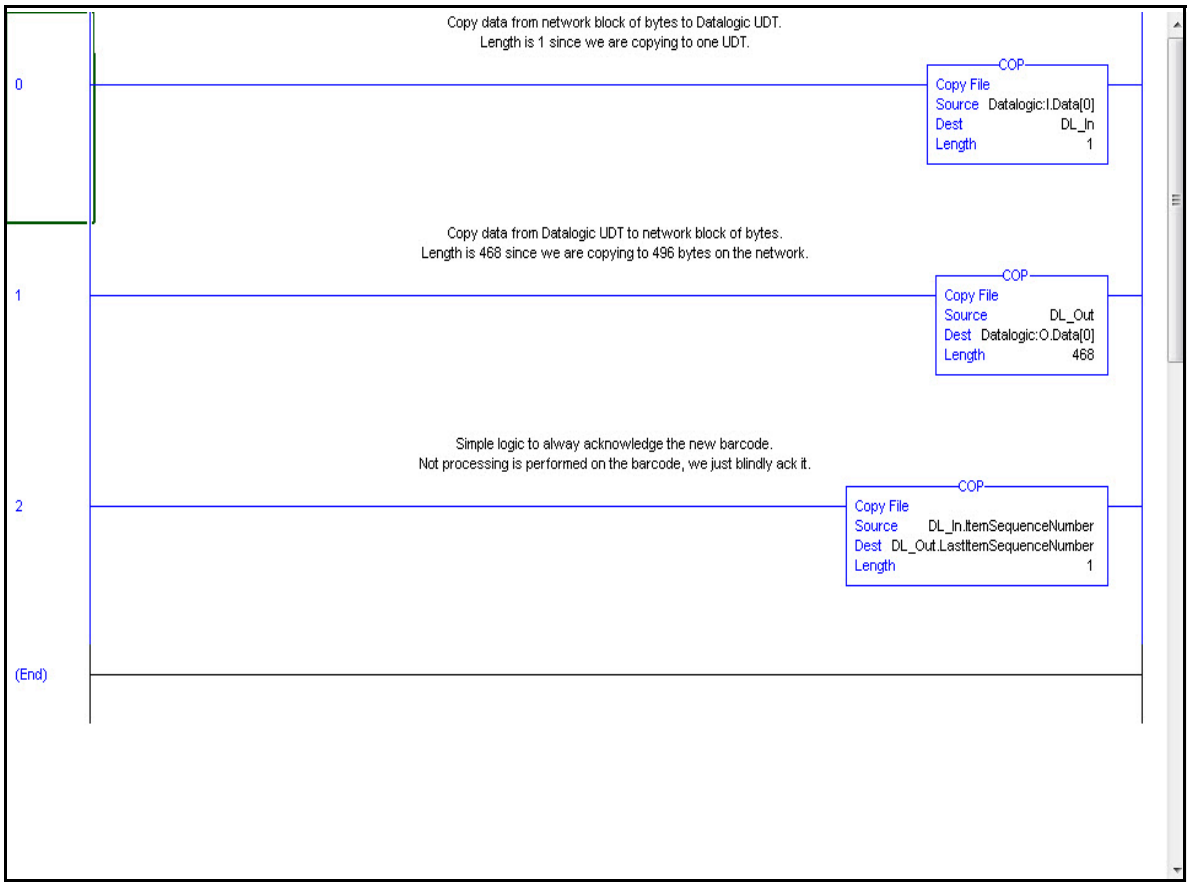


4. For this example, two new tags called **DL_In** and **DL_Out** are created with the newly imported UDT.

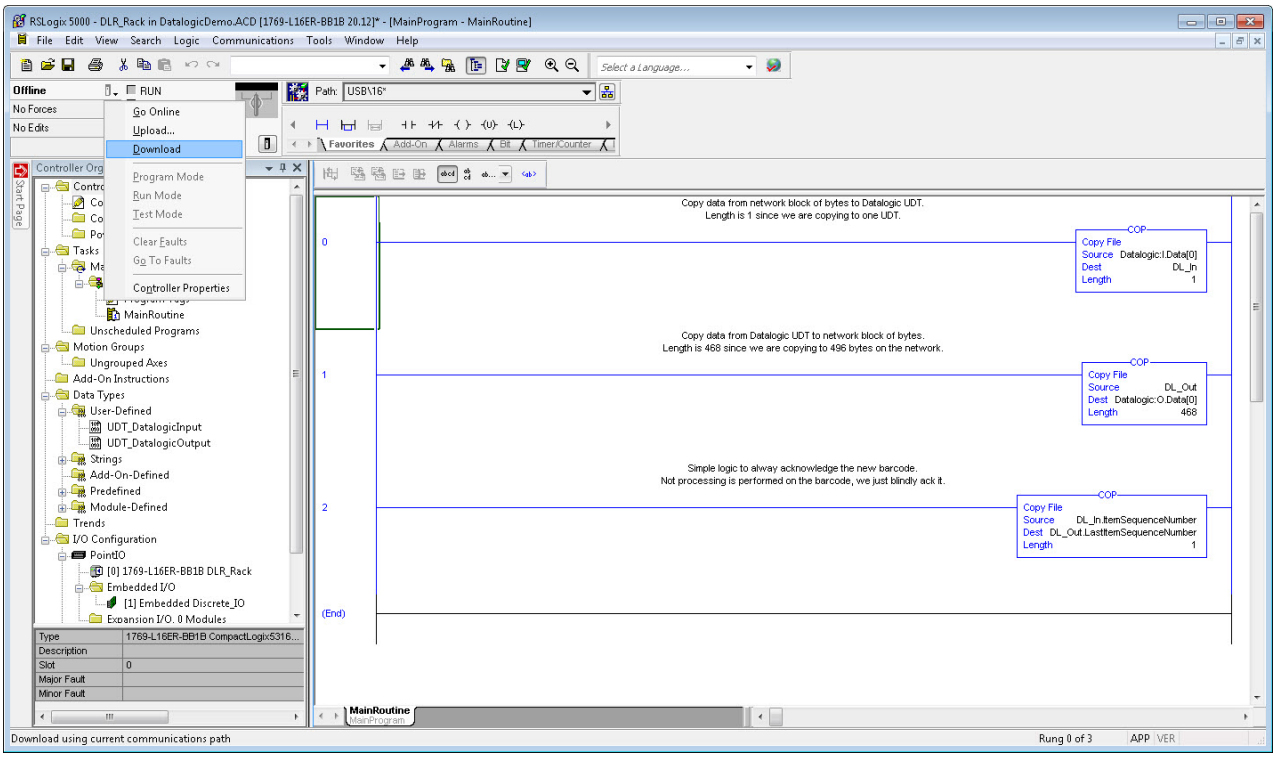




5. Create the ladder logic to copy the data between the UDT structures and the bytes of data on the network.



6. The final step is to download the program to your PLC and go online. You must transition to Run mode for the new instruction run to execute.



7. Once the program is downloaded, you are online, and the PLC is in Run mode, you can view the data using the “Monitor Tags” tab in the Controller Tags window.

Name	Value	Style	Description
Datalogic:I	{...}		
Datalogic:O	{...}		
DL_In	{...}		
DL_In.DiscreteInputs	16#0000_0000	Hex	
DL_In.Status	16#0000_0286	Hex	
DL_In.ItemSequenceNumber	599	Decimal	
DL_In.ItemTotalSize	12	Decimal	
DL_In.FragmentSequenceNumber	1	Decimal	
DL_In.FragmentSize	12	Decimal	
DL_In.FragmentData	{...}	ASCII	
DL_In.LastOutputSequenceNumber	0	Decimal	
DL_In.Reserved	0	Decimal	
DL_Out	{...}		
DL_Out.DiscreteOutputs	16#0000_0000	Hex	
DL_Out.LastItemSequenceNumber	599	Decimal	
DL_Out.LastFragmentSequenceNumber	0	Decimal	
DL_Out.OutputStatus	16#0000_0000	Hex	
DL_Out.OutputItemSequenceNumber	0	Decimal	
DL_Out.OutputItemTotalSize	0	Decimal	
DL_Out.OutputFragmentData	{...}	ASCII	
Local:1:C	{...}		
Local:1:I	{...}		
Local:1:O	{...}		

NOTES

Appendix A
Standard Defaults

The most common configuration settings are listed in the “Default” column of the table below. Page references are also provided for feature descriptions and programming bar codes for each parameter. A column has also been provided for recording of your preferred default settings for these same configurable features.

Table 16. Standard Defaults

Parameter	Default	Your Setting
ETHERNET HOST INTERFACE ONLY		
ETH Standard	Host Interface	
DHCP Client	Enable	
Static IP Address	192.168.187.31	
Subnet Mask	255.255.255.0	
Gateway Address	192.168.187.254	
DNS1 Address	0.0.0.0	
DNS2 Address	0.0.0.0	
Device Name	BC9180-433(910)(BT) [BASE_SERIAL_NUMBER]	
Data Socket Port	51000	
Web Server Security	Disable	
Web Server Password	password	

NOTES

Appendix B
Keypad

Use the bar codes in this appendix to enter numbers as you would select digits/characters from a keypad.

0

1

2

3

4

5

6

7

8

9

Keypad (continued)

A

B

C

D

E

F

Make a mistake? Scan the CANCEL bar code to abort and not save the entry string. You can then start again at the beginning.

CANCEL

ASCII Chart

ASCII Char.	Hex No.	ASCII Char.	Hex No.	ASCII Char.	Hex No.	ASCII Char.	Hex No.
NUL	00	SP	20	@	40	’	60
SOH	01	!	21	A	41	a	61
STX	02	’	22	B	42	b	62
ETX	03	#	23	C	43	c	63
EOT	04	$	24	D	44	d	64
ENQ	05	%	25	E	45	e	65
ACK	06	&	26	F	46	f	66
BEL	07	’	27	G	47	g	67
BS	08	(	28	H	48	h	68
HT	09	)	29	I	49	i	69
LF	0A	*	2A	J	4A	j	6A
VT	0B	+	2B	K	4B	k	6B
FF	0C	,	2C	L	4C	l	6C
CR	0D	-	2D	M	4D	m	6D
SO	0E	.	2E	N	4E	n	6E
SI	0F	/	2F	O	4F	o	6F
DLE	10	0	30	P	50	p	70
DC1	11	1	31	Q	51	q	71
DC2	12	2	32	R	52	r	72
DC3	13	3	33	S	53	s	73
DC4	14	4	34	T	54	t	74
NAK	15	5	35	U	55	u	75
SYN	16	6	36	V	56	v	76
ETB	17	7	37	W	57	w	77
CAN	18	8	38	X	58	x	78
EM	19	9	39	Y	59	y	79
SUB	1A	:	3A	Z	5A	z	7A
ESC	1B	;	3B	[	5B	{	7B
FS	1C	<	3C	\	5C	|	7C
GS	1D	=	3D	]	5D	}	7D
RS	1E	>	3E	^	5E	~	7E
US	1F	?	3F	_	5F	DEL	7F

www.datalogic.com

©2013-2016 Datalogic ADC S.r.l. All rights reserved. Datalogic and the Datalogic logo are registered trademarks of Datalogic S.p.A. in many countries, including the U.S.A. and the E.U.
PowerScan is a trademark of Datalogic S.p.A. or of Datalogic Group companies, registered in many countries, including the U.S. and the E.U.

Datalogic ADC S.r.l.
Via S. Vitalino, 13 | Lippo di Calderara di Reno | 40012 | Bologna - Italy
Telephone: +39 051 3147011 | Fax: +39 051 3147205

820080614 (Rev A) Sept. 2016