Modicon M340 Using Unity Pro S

Architectures and Communication Services
Reference Manual

07/2011

Schneider Electric

The information provided in this documentation contains general descriptions and/or technical characteristics of the performance of the products contained herein. This documentation is not intended as a substitute for and is not to be used for determining suitability or reliability of these products for specific user applications. It is the duty of any such user or integrator to perform the appropriate and complete risk analysis, evaluation and testing of the products with respect to the relevant specific application or use thereof. Neither Schneider Electric nor any of its affiliates or subsidiaries shall be responsible or liable for misuse of the information contained herein. If you have any suggestions for improvements or amendments or have found errors in this publication, please notify us.

No part of this document may be reproduced in any form or by any means, electronic or mechanical, including photocopying, without express written permission of Schneider Electric.

All pertinent state, regional, and local safety regulations must be observed when installing and using this product. For reasons of safety and to help ensure compliance with documented system data, only the manufacturer should perform repairs to components.

When devices are used for applications with technical safety requirements, the relevant instructions must be followed.

Failure to use Schneider Electric software or approved software with our hardware products may result in injury, harm, or improper operating results.

Failure to observe this information can result in injury or equipment damage.

Table of Contents

Safety Information

Important Information

NOTICE

Read these instructions carefully, and look at the equipment to become familiar with the device before trying to install, operate, or maintain it. The following special messages may appear throughout this documentation or on the equipment to warn of potential hazards or to call attention to information that clarifies or simplifies a procedure.

The addition of this symbol to a Danger or Warning safety label indicates that an electrical hazard exists, which will result in personal injury if the instructions are not followed.

This is the safety alert symbol. It is used to alert you to potential personal injury hazards. Obey all safety messages that follow this symbol to avoid possible injury or death.

⚠ **DANGER**

DANGER indicates an imminently hazardous situation which, if not avoided, **will result in** death or serious injury.

⚠ **WARNING**

WARNING indicates a potentially hazardous situation which, if not avoided, **can result in** death or serious injury.

⚠ CAUTION

CAUTION indicates a potentially hazardous situation which, if not avoided, **can result in** minor or moderate injury.

CAUTION

CAUTION, used without the safety alert symbol, indicates a potentially hazardous situation which, if not avoided, **can result in** equipment damage.

PLEASE NOTE

Electrical equipment should be installed, operated, serviced, and maintained only by qualified personnel. No responsibility is assumed by Schneider Electric for any consequences arising out of the use of this material.

A qualified person is one who has skills and knowledge related to the construction and operation of electrical equipment and its installation, and has received safety training to recognize and avoid the hazards involved.

About the Book
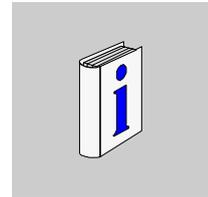
At a Glance

Document Scope

This manual presents the architectures and communication services associated with Schneider PLCs programmed using Unity Pro S.

Validity Note

This documentation is valid from Unity Pro 6.0

User Comments

We welcome your comments about this document. You can reach us by e-mail at techcomm@schneider-electric.com.

Introduction to the Communication Application

<div style="border:1px solid #000; display:inline-block; padding:0.5em 1em;">**I**</div>
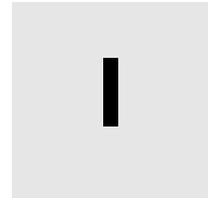
Subject of this Part

This part gives an overview of the communication application: the types of networks and buses, services and architectures available.

What's in this Part?

This part contains the following chapters:

General

1

Subject of this Chapter

This chapter gives an overview of the different characteristics of the communication application.

What's in this Chapter?

This chapter contains the following topics:

Introduction to the Application-Specific Communication Function

At a Glance

The application-specific communication function makes it possible to exchange data between different devices connected to a bus or a network.

This function applies to:

- Processors with an Ethernet, Modbus, or CANopen link,
- Specific rack-mounted communication modules,
- The terminal port of a processor.

Communication Types

The different types of communication are:

- Ethernet TCP/IP,
- Modbus,
- Character mode serial link,
- CANopen field bus,
- USB-standard fast terminal port

Available Services

The available services can be classified into three categories:

- Explicit messaging *(see page 22)* services:
    - Modbus messaging
    - Telegrams
- Implicit database access services:
    - Global data *(see Modicon M340, Premium, Atrium and Quantum Using Unity Pro, Communication Services and Architectures, Reference Manual)*
- Implicit Input/Output management services:
    - I/O scanning *(see Modicon M340, Premium, Atrium and Quantum Using Unity Pro, Communication Services and Architectures, Reference Manual)*

⚠ **WARNING**

UNEXPECTED APPLICATION BEHAVIOR - DATA COMPATIBILITY

Verify that the data exchanged are compatible data as the structure alignments are not the same for Premium/Quantum and M340 PLCs. See the page DDT: Mapping rules *(see Unity Pro, Program Languages and Structure, Reference Manual)* for more information.

Failure to follow these instructions can result in death, serious injury, or equipment damage.

Characteristics of the Different Service Types

The following table gives an overview of the main characteristics of the types of services mentioned above:

Type of service	These services make it possible...	They are used...
Messaging services	for a device (Client) to send a message to another device (Server) and obtain a response without having to program anything into the server device.	to access data from time to time.
Implicit database access services	to share data which is refreshed automatically and on a regular basis.	to synchronize applications or to transparently obtain real time images of a system on several remote PLCs.
Implicit I/O management services	to transparently and automatically manage remote I/Os on a network.	to monitor a set of distributed systems across a network.

Summary of Communication Solutions

At a Glance

The services presented earlier in this chapter are available for certain types of communication.

For example, for messaging services, certain communication functions apply to networks, others to buses and others to serial links in character mode *(see Modicon M340, Premium, Atrium and Quantum Using Unity Pro, Communication Services and Architectures, Reference Manual)*.

Summary

The following table gives an overview of the different services available according to the types of communication:

Function	Character mode	Modbus	Ethernet TCP/IP	CANopen	USB
Messaging services					
Communication functions	The communication functions that can be used depend closely on the type of communication for which they are applied *(see Modicon M340, Premium, Atrium and Quantum Using Unity Pro, Communication Services and Architectures, Reference Manual)*.				
Implicit database access services					
Global Data	-	-	X	-	-
Implicit I/O management services					
I/O Scanning	-	-	X	-	-
Legend:					
X	Yes				
-	No				

Services Available on Networks and Buses

2

Subject of this Section

This section describes the main services available on the communication buses and networks.

What's in this Chapter?

This chapter contains the following sections:

2.1   Global Data Service

Global Data

Introduction

Global Data, which is supported by the BMX NOE 0100 module, provides an automatic data exchange for the coordination of PLC applications.

NOTE: Global Data is not supported by the Ethernet port of BMX P34 20•• CPUs.

Global Data

Key features of Global Data:
- Data exchanges are based on a standard producer/consumer protocol, giving optimal performance while maintaining a minimum network load.
- The Global Data service provides real-time exchanges between stations in the same distribution group, which share variables used for PLC coordination.
- Global Data can synchronize remote applications or share a common database among several distributed applications.
- A maximum of 64 stations can participate in Global Data within the same distribution group.

Operation

An application variable is a group of contiguous words from a PLC. Communication module stations can publish or subscribe to application variables:
- **publish:** Communication modules can publish one 1024-byte local application variable for other communication modules in the time-based distribution group. The publication period can be configured from 10 ms to 15000 ms (in 10 ms increments).
- **subscribe:** A communication module can subscribe to between 1 and 64 application variables that are published by other modules in their distribution group, regardless of their location. The validity for each variable is controlled by health status bits, linked to a refresh timeout configurable between 50 ms and 15 s (15000 ms). Access to a variable element is not possible. The maximum size of the subscribed variables is 4 Kbytes.

Through Global Data configuration you can define:
- the number of valid published and subscribed variables
- the group with which these variables are associated for the communications module

Once the module is configured, exchanges between the communication modules sharing the same Distribution Group are automatically carried out when the PLC is in RUN mode.
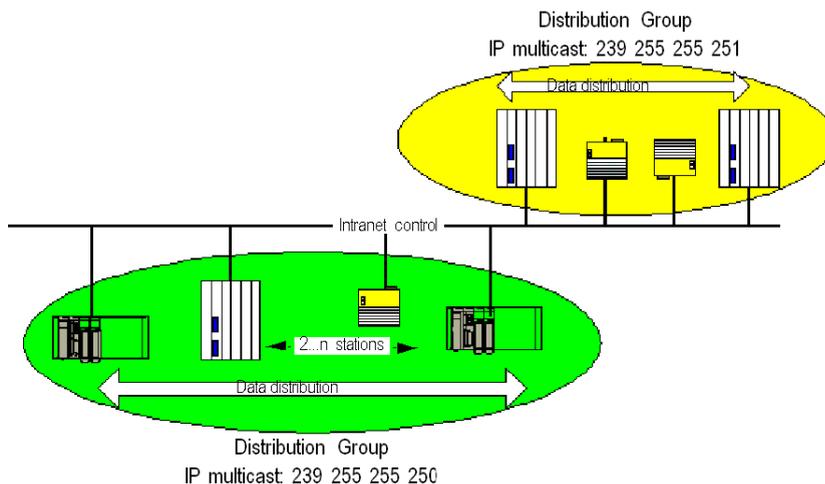
A distribution group is a group of communication modules identified by the same multicast IP address. Multicast exchanges *(see Modicon M340 for Ethernet, Communications Modules and Processors, User Manual)* distribute Global Data. Several independent distribution groups can co-exist on the same subnetwork with different multicast addresses.

Important points:

- A publish/subscribe protocol on UDP/IP is used for data distribution.
- The content of the publication variable is synchronized during the out section of the MAST task.
- The content of the subscribed variables are copied to the PLC application memory during the in section of the MAST task.
- The PLC memory zones that receive the various subscription variables should not overlap. Do not use these variables for any other purpose.

Health Bits

A health bit (status bit) is associated with each application variable.

This bit indicates the validity of each subscription variable. It is 1 if the variable has been published and the subscriber has received it in the configured validity time. Otherwise, it is 0.

Multicast Filtering

Global Data uses multicast addressing. In accordance with the nature of broadcast messages, the multicast frames transmitted by a module are repeated on all switch ports, generating network congestion.

When the switches possess this function, multicast filtering allows multicast frame spreading to be limited to only the ports that require it.

The GMRP protocol establishes the list of ports involved in data traffic.

Operating Modes

The operating modes are as follows:
- Stopping the PLC stops Global Data exchanges.
- The use of the I/O forcing system bits (%S9, %SW8, %SW9) does not stop Global Data exchanges.

Limits

There are no theoretical limits for the number of stations that share a distribution group. There is a limit on the number of variables (64) that can be exchanged within the distribution group.

NOTE: We recommend that you use fewer than 200 modules in any single distribution group.

2.2   IO Scanning Service

Description of the IO Scanning Service

At a Glance

The I/O Scanner is used to periodically:
- reads remote inputs
- writes remote outputs
- reads/writes remote outputs

The I/O Scanner is used to periodically read or write remote inputs and outputs on the Ethernet network without specific programming.

Configure the I/O Scanner with Unity Pro *(see Modicon M340 for Ethernet, Communications Modules and Processors, User Manual)*.

NOTE: I/O scanning is not supported by the Ethernet port of BMX P34 20•• CPUs.

Characteristics

This service comprises the following essential elements:
- **read zone**: the values of remote inputs
- **write zone**: the values of remote outputs
- **scanning periods**: independent of the PLC cycle and specific to each remote device

During operation, the module:
- manages the TCP/IP connections with each remote device
- scans inputs and copies their values into the configured %MW word zone
- scans outputs and copies their values from the configured %MW word zone
- reports status words so that correct operation of the service can be monitored by the PLC application
- applies the preconfigured fallback values (in the event of a communication problem)
- enables or disables each entry in the I/O scanner table in accordance with its application

Recommended Use

Scanning is performed only when the PLC is in Run mode.

This service operates with all devices that support Modbus TCP/IP server mode.

The exchange mechanism, which is transparent to the user, is executed with these request types (as long as the remote devices support them):

● read
● write
● read and write

I/O Scanner Operation

This figure illustrates the scanning of remote inputs and outputs:
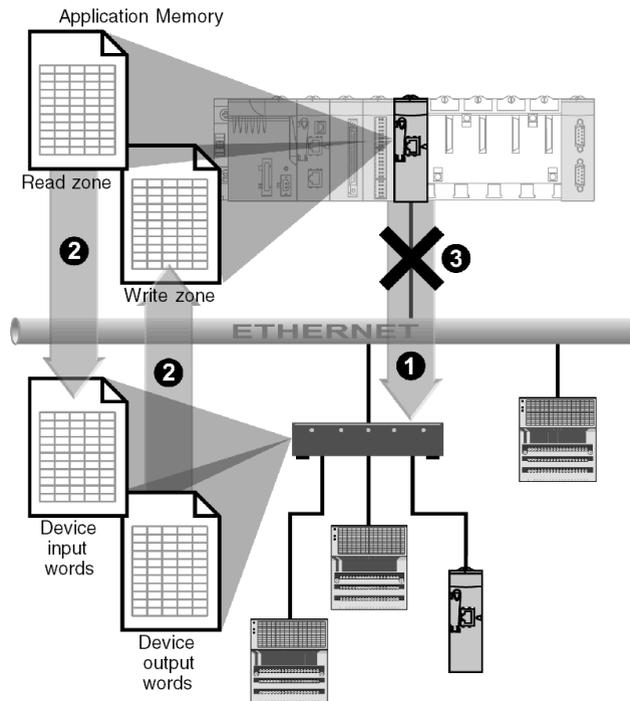
1   As soon as the PLC switches to Run mode, the module successfully opens a connection for each scanned device. (One connection is made for each line entered in the table of scanned devices.)
2   Then the module periodically reads input words and reads/writes output words for each device.
3   If the PLC switches to Stop mode, the connections to all devices are closed.

Summary of Functionality

The I/O scanning service:

- manages the connection with each remote device (one connection per remote device)
- scans the inputs/outputs of the device using Modbus read/write requests on the TCP/IP server mode.
- updates the read and write zones in the application memory
- refreshes the status bits for each remote device

Each I/O Scanner device can be enabled/disabled *(see Modicon M340 for Ethernet, Communications Modules and Processors, User Manual)*.

NOTE: These status bits show whether the module's input/output words have been refreshed.

2.3 Messaging Service

Subject of this Section

This section gives an overview of the messaging service available on Schneider PLCs.

What's in this Section?

This section contains the following topics:

Topic	Page
Messaging Service	23
Characteristics of the Messaging Service Communication Functions	24

Messaging Service

At a Glance

The messaging service makes it possible to perform inter-PLC data exchanges using communication functions.

The standard type of messaging is used on Modicon M340 installed base.

The destination entities of an exchange can either be located in a local station or in a remote station on a communication channel or directly in the CPU.

The communication functions provide an interface that is independent of the location of the destination entity. Furthermore, they mask the coding of the communication requests from the user. They thus guarantee compatibility of communication between Modicon M340 PLCs and the other platforms.

Synchronous/Asynchronous Communication

A communication function is said to be synchronous when it is wholly executed during the PLC task which activated it.

A communication function is said to be asynchronous when it is executed during one or more PLC tasks after the one which activated it.

Characteristics of the Messaging Service Communication Functions

At a Glance

These functions *(see Unity Pro, Communication, Block Library)* enable communication between one device and another. Certain functions are common to several types of communication channel. Others may be specific to one communication function.

NOTE: Processing of communication functions is asynchronous in relation to the processing of the application task which allowed them to be activated. The send/receive telegram and stop operation functions are the only exceptions, as their execution is totally synchronous with the execution of the activation task.

NOTE: It is recommended that asynchronous functions be triggered on edge and not on state so as to avoid sending several identical requests in quick succession, thus saturating the communication buffers.

Communication Functions on Modicon M340

The following table gives an overview of Modicon M340 communication functions:

Function	Its role is...
ADDM	To convert a character string into an address that can be used directly by the communication functions READ_VAR, WRITE_VAR, INPUT_CHAR and PRINT_CHAR.
INPUT_BYTE	To send an array of read request bytes.
READ_VAR	To read standard language objects: words and bits using UNI-TE or Modbus.
WRITE_VAR	To write standard language objects: words and bits using UNI-TE or Modbus.
PRINT_CHAR	To write a character string.
INPUT_CHAR	To read a character string.

Availability of Functions According to Protocols

The following table lists the protocols that support the communication functions:

Function	Uni-Telway	Character mode	Modbus	Ethernet TCP/IP	CANopen
Modicon M340					
READ_VAR	X	X	X	X	X
WRITE_VAR	X	X	X	X	X
ADDM	X	X	X	X	X
INPUT_BYTE	-	X	-	-	-

Function	Uni-Telway	Character mode	Modbus	Ethernet TCP/IP	CANopen
PRINT_CHAR	-	X	-	-	-
INPUT_CHAR	-	X	-	-	-
Legend:					
X	Yes				
-	No				

Communication Architectures

<div style="text-align: right">**3**</div>

Subject of this Chapter

This chapter gives an overview of the different communication architectures.

What's in this Chapter?

This chapter contains the following topics:

Global Architecture

At a Glance

Schneider has a communications strategy based on **open standards (core of the range)** such as:

- Ethernet Modbus TCP/IP
- CANOpen
- Modbus Link series

The possible and recommended communication architectures are presented in the following pages, according to the type of PLC used:

- At level 2: Inter-PLC network,
- At level 1: Field Bus.

The communication solutions for existing installations, from the Télémécanique or Modicon ranges, are then presented.

Network Architecture

The network architectures available for Modicon M340 PLCs are summarized below:

Modicon M340 — Ethernet TCP/IP

Field Bus Architecture

The field bus architectures available for Modicon M340, Premium and Quantum PLCs are summarized below:

Modicon M340
- Ethernet I/O Scanning
- CANopen
- Modbus
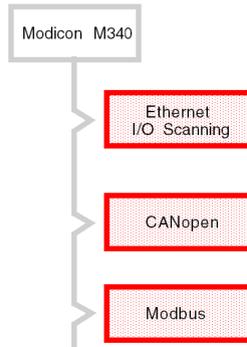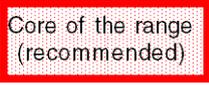
Legend:
Core of the range (recommended)

Preferred Network Architecture

At a Glance

The Ethernet network architecture presented below is Schneider's so-called "preferred"
solution.
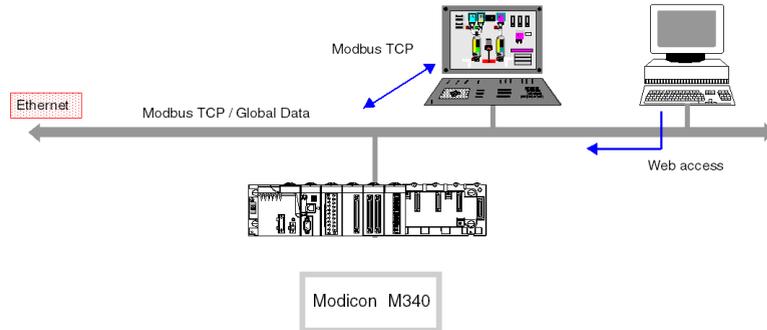
Main services:

- Global Data: inter-PLC synchronization.
- Modbus TCP/IP: client/server device for accessing automation variables.
- Web Access: access to the variables and diagnostics from a standard workstation.

Depending on the type of device, other services may also operate simultaneously:

- SMTP: e-mail
- MTP: time distribution
- SNMP: network management
- FDR: faulty device replacement

IP Routers

As PLCs do not have IP routers, standard routers must be used to connect two Ethernet TCP/IP networks. The following diagram shows the communication between two Ethernet networks:
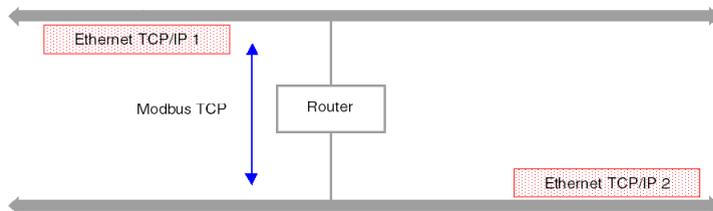
Communication with the Installed Base

"Télémécanique" Architecture

The Ethway, Xip, Fipway, Uni-telway and Fipio installed bases use the Télémécanique Uni-te private messaging protocol on a network layer called X-way.

This layer guarantees transparent routing of Uni-te messages between each of these networks. Only Premium and Micro PLCs support this protocol.

In the architecture below, transparency may be achieved by configuring the Premium 2 PLC as a Bridge, and provided the Premium 1 PLC or the Unity terminal uses the Xip protocol (X-way on TCP/IP).
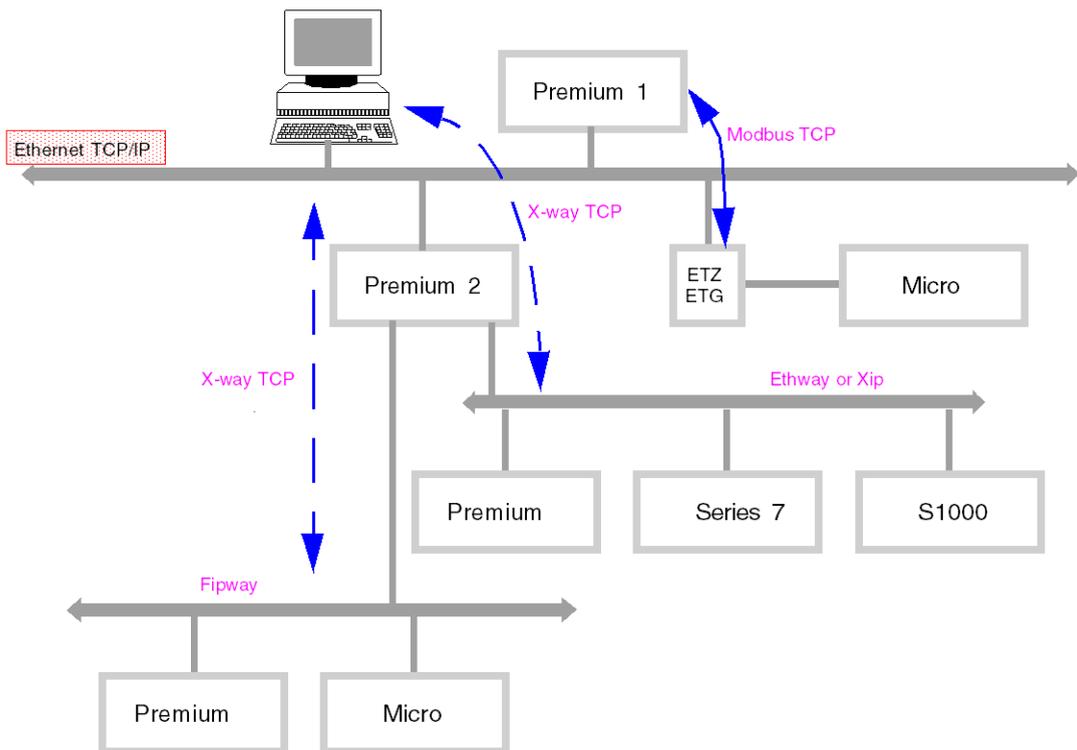
"Modicon" Architecture

The Modicon installed base uses the standard Modbus protocol on serial link or token bus.

It is not possible to perform routing using this protocol.

This is, however, possible using gateways or bridges.
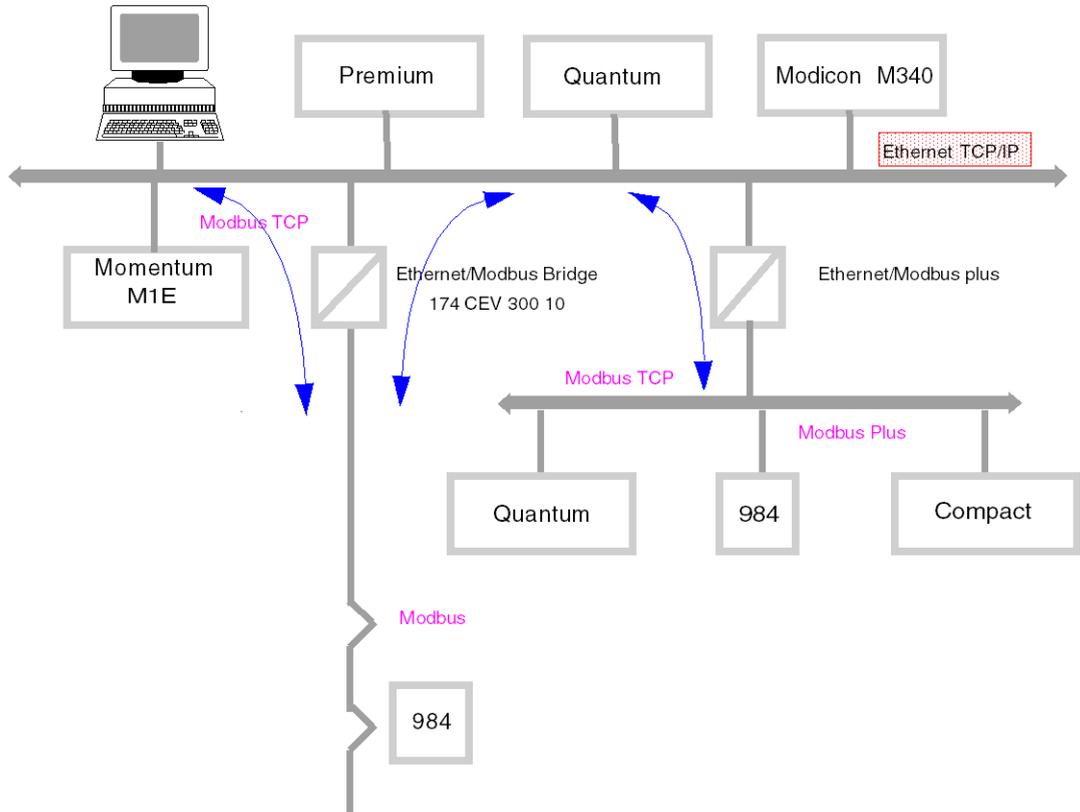
Interoperability

4

List of Modbus Function Codes

At a Glance

M340 PLCs have communication server kernels that accept the common Modbus function codes. These are listed in the table on this page.

As servers, M340 PLCs recognize all **Class 0** and **Class 1** Modbus function codes, as stipulated in the Modbus specifications available at `http://www.Modbus.org`. Their server kernel also includes the function code `23` for reading/writing of consecutive variables.

List of Modbus Requests Recognized When Connected as a Server

The following table lists the function codes and the address of the Modbus function codes, recognized by M340 platform:

Function code	M340 memory address	Meaning
1	%M	Read output bits
2	%M	Read input bits
3	%MW	Read consecutive integer values
4	%MW	Read consecutive input integer values
5	%M	Write single output bit
6	%MW	Write single integer value
15	%M	Write n output bits
16	%MW	Write consecutive integer values
23	%MW	Read/write consecutive integer values

Use of Modbus Function Codes as a Client on M340

The table below lists the Modbus function codes and their use as a client on M340 PLCs:

Function code	M340 memory address	Modbus request	Communication function
1	%M	Read output bits	READ_VAR
2	%I	Read input bits	READ_VAR
3	%MW	Read consecutive integer values	READ_VAR
4	%IW	Read consecutive input integer values	READ_VAR
15	%M	Write n output bits	WRITE_VAR
16	%MW	Write consecutive integer values	WRITE_VAR

The way in which to use function codes with communication functions is described in the Modbus manual *(see Premium and Atrium using Unity Pro, Asynchronous Serial Link, User Manual)*.

NOTE: Interoperability with Windows applications is provided in 2 ways:

● Access to the PLC variables can be easily achieved with OFS software.
● The application download function, import/export source format function and access to operating modes (RUN/STOP/INIT) functions can be performed using the UDE (Unity Development Edition) range.

⚠ **WARNING**

UNEXPECTED APPLICATION BEHAVIOR - DATA COMPATIBILITY

Verify that the data exchanged are compatible as the data structure alignments are not the same for Premium/Quantum and M340 PLCs.

See the page DDT: Mapping rules *(see Unity Pro, Program Languages and Structure, Reference Manual)* for more information.

Failure to follow these instructions can result in death, serious injury, or equipment damage.

Addressing

II
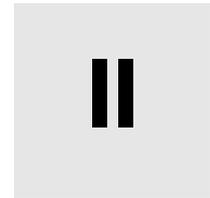
Subject of this Part

This part describes the different addressing solutions for devices on a communication bus or network.

What's in this Part?

This part contains the following chapters:

General Points Concerning Addressing

<div style="text-align: right; font-size: 2em; font-weight: bold;">5</div>

5.1 General

Description

At a Glance

In a communication architecture, each device must be identified by an address. This address is specific to each device, and enables the device initiating communication to determine the destination precisely. Similarly, for the configuration of services such as Global Data on Ethernet, the Peer Cop service on Modbus Plus or common words and shared tables on Fipway, these addresses make it possible to identify the stations that own different shared information.

Modicon M340 PLCs support 2 types of addressing depending on the type of device, network or bus used:

- IP addressing *(see page 39)*
- Modicon M340 PLCs addressing *(see page 43)*

IP Addressing

6

Reminder about IP Addressing

IP Address

On a TCP/IP Ethernet network, each device must have a **unique IP address**. This address is made up of two identifiers, one of which identifies the network, while the other identifies the connected machine.

The uniqueness of the addresses is managed as follows:

- If the type of network environment is open, the uniqueness of the address is guaranteed by the attribution of a network identifier by the relevant authority in the country where the network is located.
- If the type of environment is closed, the uniqueness of the address is managed by the company's network manager.

An IP address is defined as 32 bits. It consists of 4 numbers, one for each byte of the address.

NOTE: Standardized and made common largely thanks to the Internet, IP addressing is described in detail in RFCs (Request For Comment) 1340 and 791 which stipulate the Internet standards as well as in computing manuals describing networks. You can refer to these sources for further information.

Example

Depending on the size of the network, three classes of address can be used:
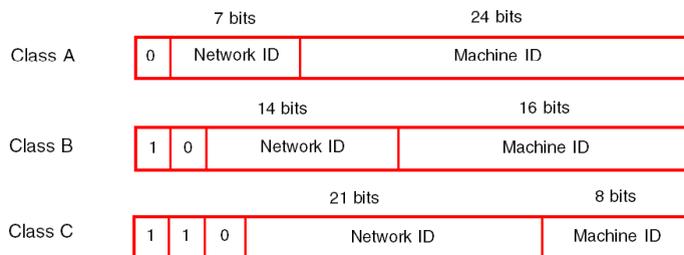
Spaces reserved for the different classes of IP address:

Class	Range
A	**0**.0.0.0 to **127**.255.255.255
B	**128**.0.0.0 to **191**.255.255.255
C	**192**.0.0.0 to **223**.255.255.255

- Class A addresses are intended for large-scale networks which have a large number of connected sites.
- Class B addresses are intended for medium-scale networks which have fewer connected sites.
- Class C addresses are intended for small-scale networks which have a small number of connected sites.

Sub-Addressing and Subnet Mask

An IP address is composed of two identifiers, one identifies the network while the other identifies the connected machine. In reality, the machine identifier can also hold a subnet identifier.

In an open environment, having received a network identifier from the relevant authority, the local system administrator has the possibility of managing many networks. This means that local networks can be installed without having any effect on the outside world which still sees only one network, the one designated by the network identifier.

The subnet mask makes it possible to see the number of bits attributed respectively to the network identifier and to the subnet identifier (bits at 1), and then to the machine identifier (bits at 0).

Example

Example: 140.186.90.3

	16 bits	8 bits	8 bits
Class B	Network identifier = 140.186	Sub-network identifier = 90	Machine identifier = 3

Subnet mask	24 bits at 1	8 bits at 0
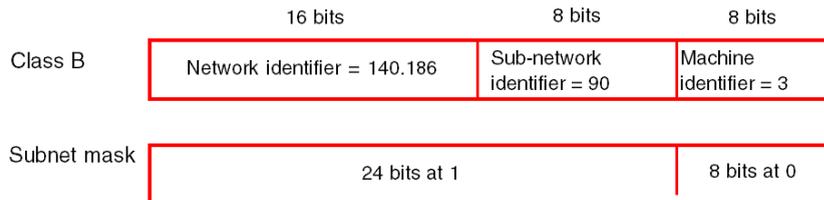
The segmentation allows for 254 possible sub-networks, each with 254 sub-network machines.

The value of the subnet mask should be chosen so that it is consistent with the IP address class.

The subnet mask will have the following value:

- For a class A address: 255.xxx.xxx.xxx
- For a class B address: 255.255.xxx.xxx
- For a class C address: 255.255.255.xxx

Where xxx is an arbitrary value which can be chosen by the user.

Gateway

The term Gateway is used in this manual in the sense of "router". If the target machine is not connected to the local network, the message will be sent to the "default gateway" connected to the local network which will manage the routing to another gateway or towards its final destination.

35013341 07/2011

Modicon M340 PLCs Addressing

7

Purpose of this Chapter

This chapter describes Modicon M340 PLCs addressing and indicates its fields of application.

What's in this Chapter?

This chapter contains the following topics:

Modicon M340 Types of Communication Entities

At a Glance

There are different types of communication entities.

These exchanges are performed by the communication functions described in the Communication EF library.

It is possible to class addresses into 3 types:

- local addresses, identified by `r.m.c.SYS`, or more simply, `r.m.c`,
- remote addresses, to address a device (Modbus, CANopen or Ethernet) directly connected to the channel,
- broadcast addresses, depend on the network. For Modbus communication, broadcast address is obtained with the slave number set to 0. Note that a broadcast address can be used for all networks but requires that the communication channel supports broadcasting. This is not always the case.

SYS Keyword

`SYS` gives access to a local module or a channel server. `SYS` is used for character mode and can be ommitted.

Broadcast Addresses

Broadcast addresses depend on the destination devices:

Destination	Broadcast address
Broadcast to all Modbus slaves (the slave number equals 0)	rack.module.channel.0

Modicon M340 Addressing for a Communication Entity

At a Glance

With Modicon M340 PLCs, it is possible to address any Modicon M340 PLC communication channel and any device directly connected to a Modicon M340 PLC communication channel.

Each device is identified by a unique address, which consists of a device number or an IP address. The addresses then differ according to the protocol:

- Ethernet TCP/IP
- Modbus or CANopen
- Character Mode

Within a station, each communication entity is characterized by a topological address (access path) and a target entity.

NOTE: An address is expressed in the form of a character string. However, it can only be used in conjunction with the function `ADDM`, which is why the following notation will be used to describe an address: `ADDM('address string')`.

Modicon M340 addressing uses 3 concepts:

- The target entity depends on the communication EF and is chosen implicitly:
  - `MBS` for addressing a Modbus server,
  - `TCP.MBS` for addressing a TCP Modbus server,
  - `SYS` for addressing a channel server on Character mode. `SYS` can be ommitted.
- The communication channel is explicit (processor's or module's position and communication channel number) or symbolized with the Netlink name for Ethernet communication.
- The node address depends on the communication protocol:
  - IP address with Ethernet,
  - node address with CANopen,
  - slave address with Modbus.

Addressing a Station on a Ethernet

The address of a station on Ethernet takes the form:

- `ADDM('Netlink{hostAddr}')`
- `ADDM('Netlink{hostAddr}TCP.MBS')`
- `ADDM('Netlink{hostAddr}node')`
- `ADDM('r.m.c{hostAddr}')`
- `ADDM('r.m.c{hostAddr}TCP.MBS')`
- `ADDM('r.m.c{hostAddr}node')`
- `ADDM('{hostAddr}')`
- `ADDM('{hostAddr}TCP.MBS')`
- `ADDM('{hostAddr}node')`

Where:

- Netlink: network name set in the Net Link field of Ethernet channel
- hostAddr: IP address of device
- r: rack number (rack)
- c: channel number (channel)
- node: Modbus or CANopen node behind a gateway (gateway identified with hostAddr)

NOTE: If the netlink name is ommitted the system takes the default netlink connection which is the closest link to the processor (usually the processor Ethernet channel).

Addressing of a Device on a CANopen Bus

The address of a device on a CANopen bus takes the form `ADDM('r.m.c.e')`, where:

- r: rack number (rack)
- m: rack module position
- c: channel number (channel) of CANopen port (2)
- e: CANopen slave node (equipment) (range 1 to 127)

Addressing a Device on a Modbus

The address of a device on a Modbus bus takes the form `ADDM('r.m.c.e.MBS')`, where:

- r: rack number (rack)
- m: rack module position
- c: channel number (channel) of Modbus port (0)
- e: Modbus slave number (equipment) (range 1 to 247)

Addressing a Device on Character mode

To send or receive a character string, you can use `ADDM('r.m.c')` or `ADDM('r.m.c.SYS')`, where:

- r: rack number (rack)
- m: rack module position
- c: channel number (channel) of Character mode port (0)
- SYS: keyword used to stipulate the station server system *(see page 44)*. `SYS` can be ommitted.

Example

The figure below describes the address of the servodrive. The example here shows slave 14 on channel 2 (CANopen) of the module in rack 0, slot 0:
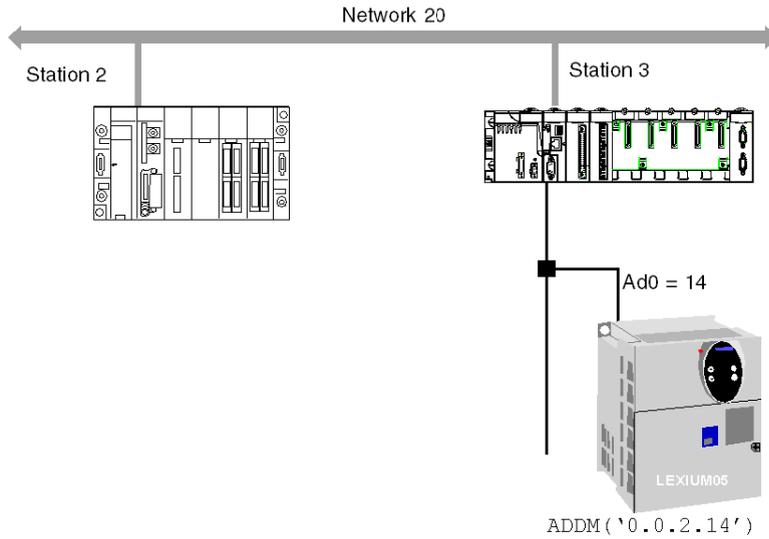
Processor Communication Channels Addressing

At a Glance

Following are examples of the different types of addressing for a processor's communication channels.

The examples are based on a Modicon M340 type processor.

The modules have a topological address that is a function of the module's position in the rack.

The first two slots of the rack (marked PS and 00) are reserved for the rack's power supply module (BMX CPS ••••) and the processor (BMX P34 •••••) respectively.

Available communication channels

The available communication channels vary depending on the processor:

Processor	Integrated Modbus Connection	Integrated CANopen Master Connection	Integrated Ethernet Connection
BMX P34 1000	X	-	-
BMX P34 2000	X	-	-
BMX P34 2010/20102	X	X	-
BMX P34 2020	X	-	X
BMX P34 2030/20302	-	X	X
Key			
X Available			
- Not available			

Processor Communication Channels Addressing

The diagram below shows an example of Modicon M340 configuration including a BMX P34 2010 processor and the addresses of the processor communication channels:
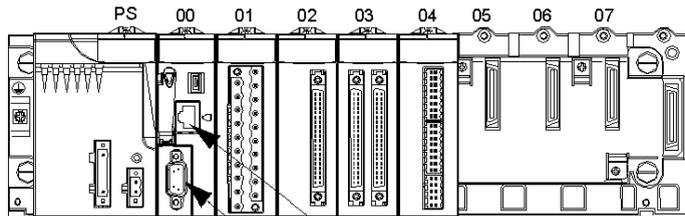
Rack 0. Module 0. Channel 0: Serial port
(available on BMX P34 1000/2000/2010/20102/2020 processors)

Rack 0. Module 0. Channel 2: CANopen port
(available on BMX P34 2010/20102/2030/20302 processors)

The diagram below shows an example of Modicon M340 configuration including a BMX P34 2030 processor and the addresses of the processor communication channels:
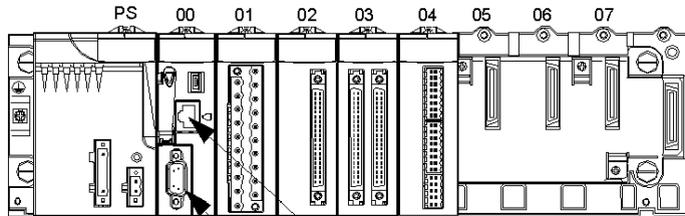
Rack 0. Module 0. Channel 3: Ethernet port
(available on BMX P34 2020/2030/20302 processors)

Rack 0. Module 0. Channel 2: CANopen port
(available on BMX P34 2010/20102/2030/20302 processors)

Example of Modicon M340 Ethernet Addressing

At a Glance

With this type of addressing, a station can access different station connected to logical network.

Connection via CPU Ethernet port

A device with the IP address 139.180.204.2 is connected to the Ethernet network. It is the processor Ethernet port configured with Netlink name `Ethernet_1`.

hostAddr: 139.180.204.1

Ethernet
Netlink: Ethernet_1

hostAddr: 139.180.204.2
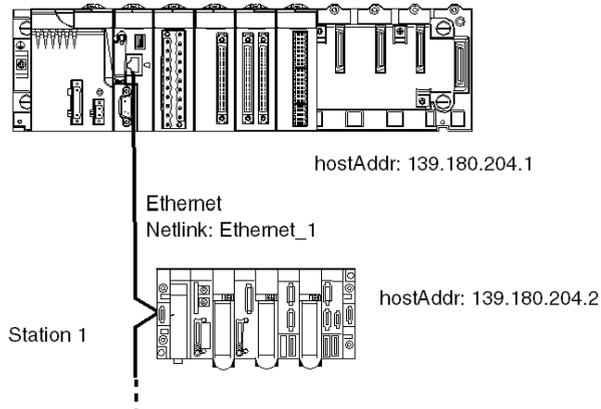
Station 1

Address settings station 1: ADDM('0.0.3{139.180.204.2}')

or Address settings station 1: ADDM('Ethernet_1{139.180.204.2}')

Example of Modicon M340 CANopen Addressing

At a Glance

With this type of addressing, a master station can access different slaves connected to CANopen bus.

Addressing Rules

The syntax of CANopen addressing is `ADDM ('r.m.c.node')`. The meaning of the string parameter is as follows:

- r: rack address. The processor's rack address is always 0.
- m: module address. The Modicon M340 processor's slot number in the rack is always 0.
- c: channel address. The Modicon M340 CANopen port is always channel 2.
- node: slave number to which the request is being sent. The range for configured slave numbers is from 1 to 127.

Example

In the following example, the Modicon M340 processor's bus manager addresses the Lexium 05 device at connection point 28:
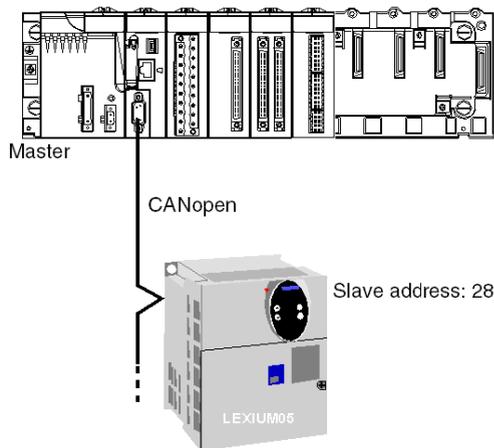
Address settings of slave 28: `ADDM('0.0.2.28')`.

NOTE: In addition to the address defined by `ADDM`, the `READ_VAR` and `WRITE_VAR` functions use another parameter `NUM`, which must be defined to address the SDO to be read or written.

Examples of Modicon M340 Modbus and Character Mode Addressing

At a Glance

The following examples deal with:

- Modbus addressing
- Character mode addressing.

Modbus Addressing Rules

The syntax of Modbus addressing is `ADDM ('r.m.c.node')`. The meaning of the string parameter is as follows:

- r: rack address. The processor's rack address is always 0.
- m: module address. The Modicon M340 processor's slot number in the rack is always 0.
- c: channel address. The Modicon M340 processor's serial port is always channel 0.
- node: slave number to which the request is being sent. The range for configured slave numbers is from 1 to 247.

NOTE: In a Modbus Slave configuration, an additional address, number 248, is used for a point-to-point serial communication.

Serial Link Using Modbus Protocol

The diagram below shows two Modicon M340 processors connected via a serial link and using Modbus protocol:

Modbus Master
BMX P34 2010 Processor

Modbus Slave
BMX P34 2010 Processor
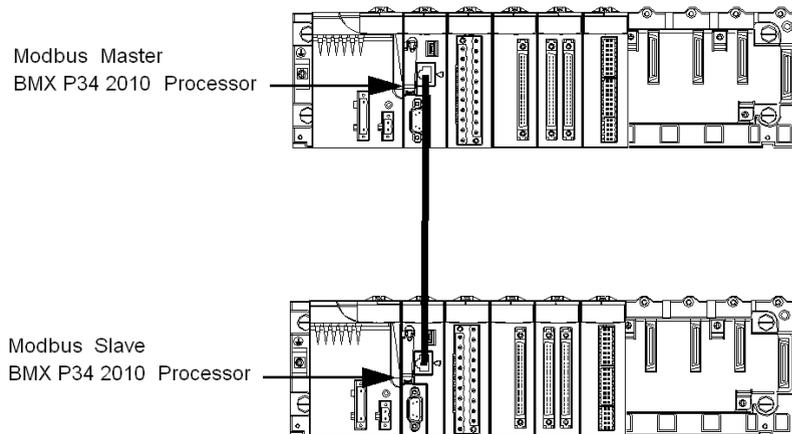
The address settings of the slave processor number 8 are `ADDM('0.0.0.8')`.

Character Mode Addressing Rules

The syntax of Character mode addressing is ADDM ('r.m.c') or ADDM ('r.m.c.SYS') (SYS can be omitted). The meaning of the string parameter is as follows:

- r: rack address of the connected device.
- m: module address of the connected device.
- c: channel address of the connected device.
- SYS: keyword used to stipulate the station server system. SYS can be omitted.

Serial Link Using Character Mode Protocol

The diagram below shows a Modicon M340 processor linked to a data entry/display terminal TM8501:
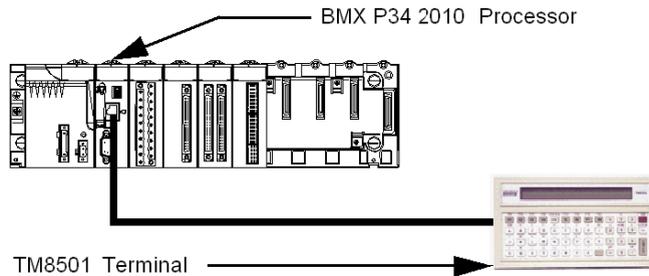
The address settings of the TM8501 terminal are ADDM('0.0.0') or ADDM('0.0.0.SYS').

Examples of Modicon M340 Communication EFs Addressing

At a Glance

The multi-network addressing available on Modicon M340 PLCs is described below.

Example 1

The first example is a multi-network configuration as follows:
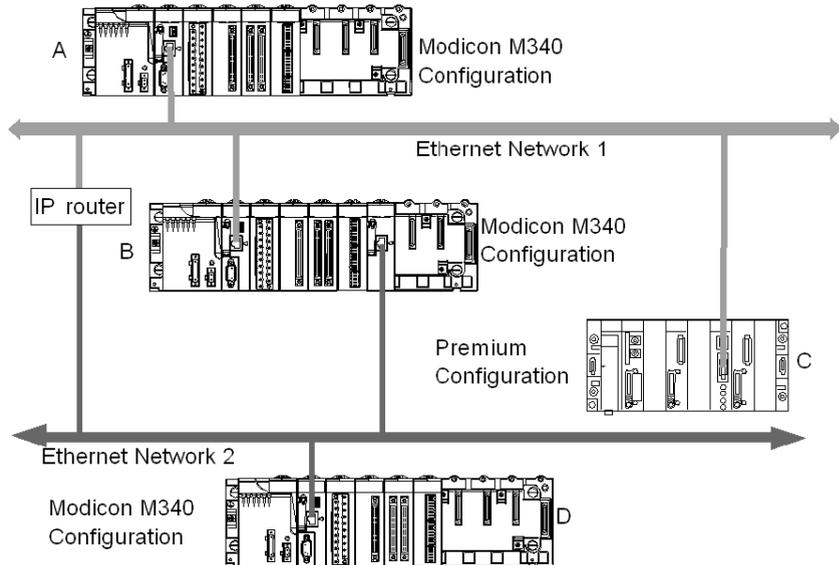
In the diagram above there are the following configurations:

- Three Modicon M340 configurations called A, B and D
- One Premium configuration called C

All the configurations can communicate because of the following statements:

- A and B: communication between two Modicon M340 PLCs on an Ethernet network is possible.
- A and C: communication between a Modicon M340 PLC and a Premium PLC is possible on an Ethernet network.
- A or C, and D: communication between two Modicon M340 PLCs or between a Modicon M340 PLC and a Premium PLC on Ethernet multi-network is possible. An IP router is required.

Example 2

The second example is a multi-network configuration as follows:
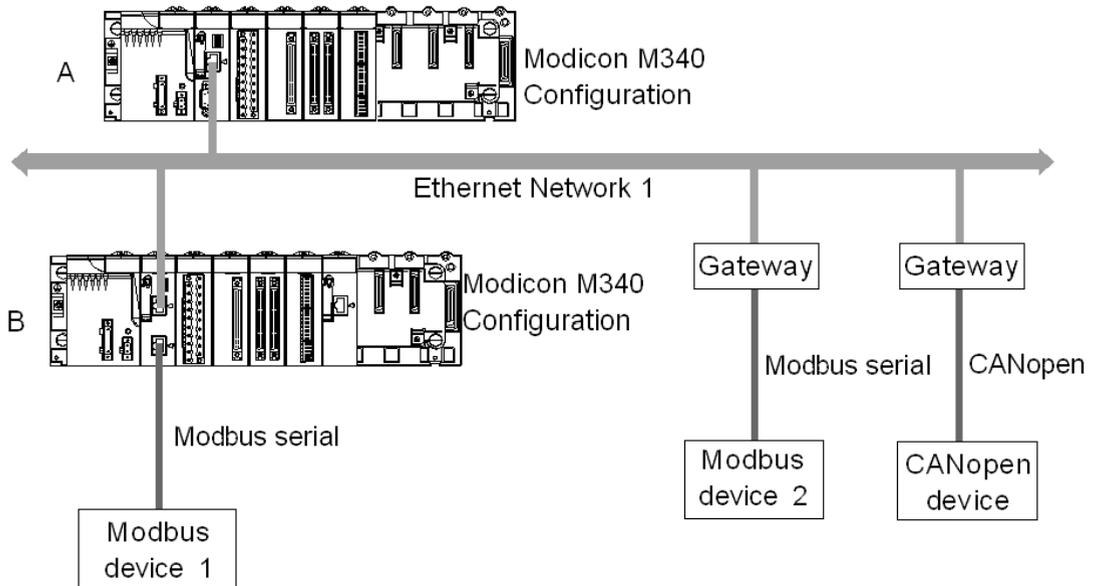
In the diagram above there are two Modicon M340 configurations which are called A and B. The configuration B is directly connected to the Modbus device 1 via Modbus communication channel.

Communication between the two Modicon M340 PLCs is possible because the configurations are linked to the same Ethernet network.

Communication between the configuration A and the Modbus device 2 is possible only if you use an Ethernet/Modbus gateway. In case of it is a CANopen device, an Ethernet/CANopen gateway is required.

NOTE: To address the CANopen device or the Modbus device 2 on the configuration A you must use the following syntax :
`ADDM('Netlink{hostAddr}node')`, the gateway being identified with `hostAddr` field. For example, if the Netlink is set to `Ethernet_1`, the gateway address is `139.160.234.64` and the slave number of the device is set to 247, the syntax of the ADDM function is as follows:
`ADDM('Ethernet_1{139.160.230.64}247')`

General points concerning bridging

8

Subject of this Chapter

This chapter gives an overview of the different bridging solutions for devices in a communication architecture.

What's in this Chapter?

This chapter contains the following topics:

Bridging Description

At a Glance

There are two available connection modes:

- Direct PLC access: Unity Pro connects directly to the PLC.
- Transparent PLC access or bridging: Unity Pro connects to a PLC via a Modicon M340 PLC.

Set Address

The direct PLC access and transparent PLC access features are available via the **Set Address** screen. In this screen, you must enter the PLC address.

To access the **Set Address** screen, use the **Set Address** command on the **PLC** menu.

The **Set Address** screen is as follows:
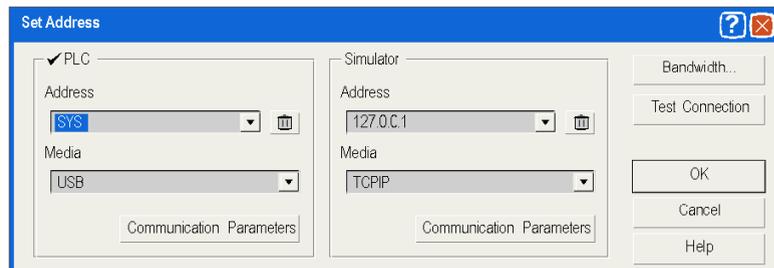
Direct PLC Access Syntax

The available syntaxes for a direct PLC access are described below:

Connection used	Address
USB	SYS or empty
Ethernet	IP address: 139.169.3.4
Modbus	Slave number

Transparent PLC Access Syntax

The bridging address string consists of two parts:

- First part: The "via address" (optional).
- Second part: The "remote PLC address".

The address parameter syntax is:

via address\\remote PLC address

The syntax for the "remote PLC address" depends on the network link type:

Network link	Remote PLC address
Modbus slave	Link_address.Modbus Slave Number
Ethernet	Link_address {IP address}
Ethernet device	Link_address.UnitID

"Link_address" is a r.m.c-type topological address where:

- r: rack address.
- m: module address.
- c: channel address.

The "via address" is a classical address depending on the media:

Media	Via address
Modbus slave	Slave_nbr
USB	SYS or empty
Ethernet	IP address

Online Service Limitations of Transparent PLC Access

The transparent PLC access or bridging offers:

- full online services if the remote PLC is a Modicon M340 or a Quantum PLC.
- restricted online services if the remote PLC is a Unity Premium PLC (not all option module screens work).
- no online services for the modules ETY 4103, ETY 5103, WMY 100 and ETY PORT (except embedded Ethernet ports of Premium PLCs P57 4634, P57 5634 and P57 6634).

Bridging Example

At a glance

The following pages present an example of PLC configurations bridging and its transparent PLC adresses.

Bridging Example

The following example consists of the following PLC configurations:

- Configuration 3: this Modicon M340 configuration consists of the following communication modules:
  - An Ethernet-Modbus processor with IP address 139.160.235.34 and Modbus Master. The processor is in slot 0 of the configuration so that the topological address of this processor's Ethernet channel is 0.0.3 and the topological address of this processor's Modbus channel is 0.0.0.
  - An Ethernet module BMX NOE 0100 with IP address 118.159.35.2. The Ethernet module is in slot 5 of the configuration so that the topological address of this Ethernet module's channel is 0.5.0.
- Configuration 1: this configuration consists of a remote PLC linked to processor's Ethernet channel of the configuration 3. The IP address of this remote PLC is 139.160.235.16.
- Configuration 2: this configuration consists of a remote PLC linked to Ethernet module's channel of the configuration 3. The IP address of this remote PLC is 118.159.35.45.
- Configuration 4: this configuration consists of a remote PLC linked to processor's Modbus channel of the configuration 3. The Modbus slave address of this remote PLC is 66.

This diagram presents the bridging example:
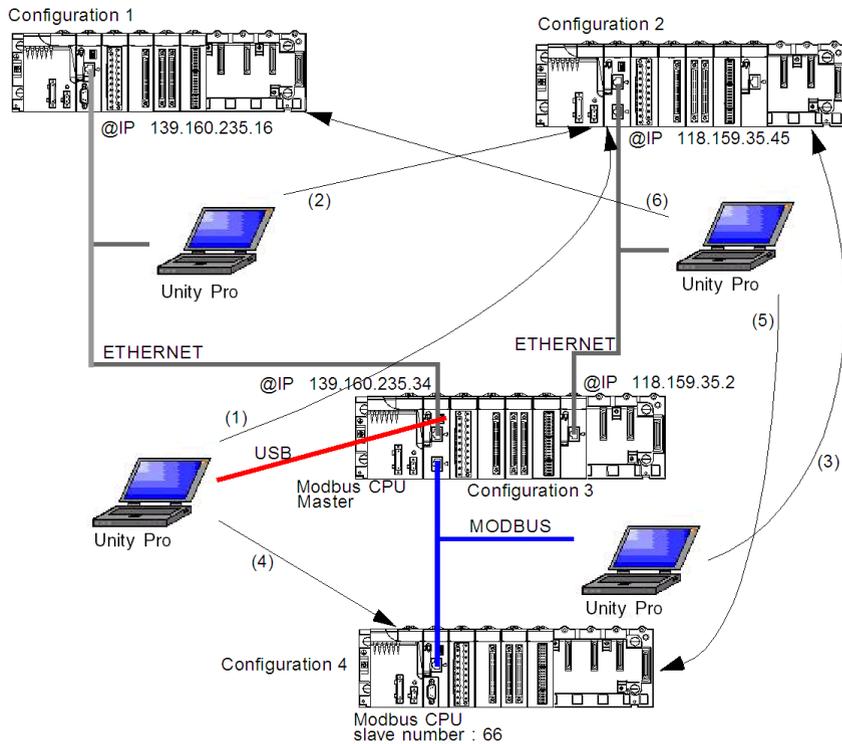
The transparent PLC addresses are as follows:

Bridging configuration	Transparent PLC address
(1) USB connection to remote PLC, which is linked to an Ethernet module	SYS\\0.5.0.{118.159.35.45}
(2) processor's Ethernet channel to remote PLC, which is linked to an Ethernet module	139.160.235.34\\0.5.0{118.159.35.45}
(3) processor's Modbus channel to remote PLC linked, which is linked to an Ethernet module	5\\0.5.0{118.159.35.45}
(4) USB connection to remote PLC, which is linked to processor's Modbus channel	SYS\\0.0.0.66
(5) Ethernet module connection to remote PLC, which is linked to processor's Modbus channel	118.159.35.2\\0.0.0.66
(6) Ethernet module connection to remote PLC, which is linked to processor's Ethernet channel	118.159.35.2\\0.0.3{139.160.235.16}

Operating Modes

III
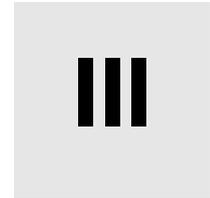
Subject of this Part

This part describes the operating modes associated with expert communication.

What's in this Part?

This part contains the following chapters:

35013341 07/2011

Network Configuration

9

Subject of this Chapter

This chapter presents the tools for configuring a network at the global level and at the station level.

What's in this Chapter?

This chapter contains the following topics:

Network Configuration Principle Using Unity Pro

At a Glance

With Unity Pro, the installation of a network takes place using the project browser and the hardware configuration editor.

The method involves the following four steps:

- Creation of a logic network,
- Configuration of the logic network,
- Declaration of the module,
- Association of the module with the logic network.

These four steps are presented further on in this documentation.

NOTE: The advantage of this method is that from the second step onwards, you can design your communication application (you do not need to have the hardware to start work) and use the simulator to test its operation.

NOTE: The first two steps are performed using the project browser and the next two using the hardware configuration editor.

This manual introduces the method. For detailed information on how to configure the various networks, please refer to the specific documentation:

- Ethernet configuration *(see Modicon M340 for Ethernet, Communications Modules and Processors, User Manual)*

Creating a Logic Network

At a Glance

The first step in implementing a communication network is to create a logic network.

Creating a Logic Network

The following table describes how to create a network using the project browser.

Step	Action
1	Expand the *Communication* directory in the **project browser**. **Result**: Communication Networks
2	Right-click in the *Networks* sub-directory and select the **New network** option. **Result**: Add network Network	Comment List of available networks: None selected Change name: OK Cancel Help
3	Select the network that you wish to create from the list of available networks and give it a meaningful name. **Result**: Example of an Ethernet network. Add network Network	Comment List of available networks: Ethernet Change name: Ethernet factory OK Cancel Help **Note**: You can also add a comment, if you so wish, by clicking on the **Comment** tab.
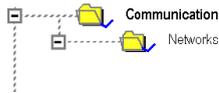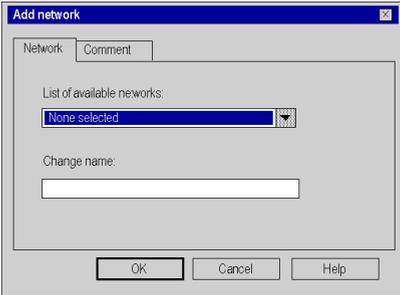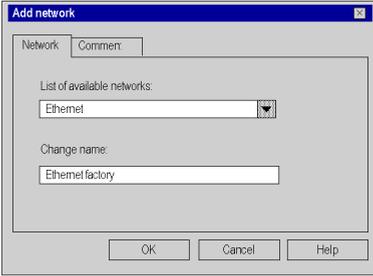
Step	Action
4	Click OK and a new logic network is created. **Result**: We have just created the Ethernet network that appears in the project browser. Communication Networks Ethernet factory **Note**: As you can see, a small icon indicates that the logic network is not associated with any PLC hardware. Furthermore, the small blue "v" sign indicates that the project needs to be rebuilt before it can be used in the PLC.
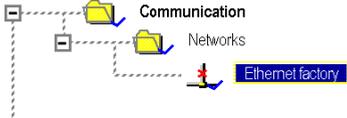
Configuring a Logic Network

At a Glance

The second step in implementing a communication network consists of configuring a logic network.

This manual introduces the access to network configuration. For further information on configuring the various networks please refer to the specific documentation:

- Ethernet configuration *(see Modicon M340 for Ethernet, Communications Modules and Processors, User Manual)*

Configuring a Logic Network

The table below describes how to access the configuration of a network using the project browser.

Step	Action
1	In the project browser, expand the **Networks** sub-tab located in the **Communication** tab of the tree directory to display all the networks of the project. **Example**:
2	Double-click the network you wish to configure to obtain the network's configuration window. **Note**: The windows differ according to the network family selected. However, for all networks it is from this window that you can configure the Global Data, IO scanning etc. **Note**: For Ethernet networks an intermediate step is necessary which involves selecting the family of the module that will be used in the hardware configuration.
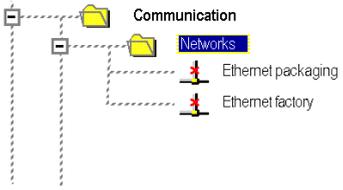
Associating a Logic Network with Network Hardware

At a Glance

The final step in implementing a communication network is to associate a logic network with a network module. Although the screens differ, the procedure is the same for each network device.

How to Associate a Logic Network

The following table describes how to associate a logic network to a network device declared in the hardware configuration editor.

Step	Action
1	Open the hardware configuration editor.
2	Right-click the device (Ethernet module) that you wish to associate with a logical network.
3	Select the channel and function. **Result:** For a BMX NOE 0100 module:
4	In the **Network link** field, select the network to be associated with the module. **Result:**
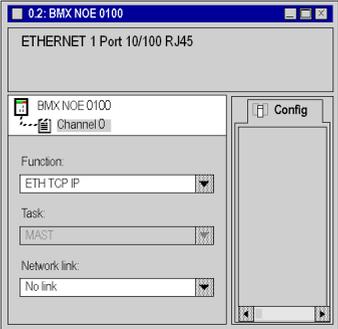
Step	Action
5	Confirm your choice and close the window. **Result:** The logic network is associated with the device. The icon associated with this logic network changes and indicates the existence of a link with a PLC. Furthermore, the rack, module and channel numbers are updated in the logic network configuration screen. In our example we obtain the following project browser:
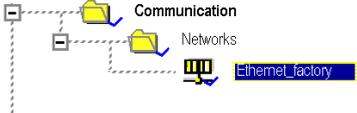
Debugging

10

Description of the Communication Debug Screens

At a Glance

The debug screen dedicated to the application-specific communication function may be accessed via the **Debug** tab. It is broken down into two distinct sections:

- The top and left-hand part of the screen, common to all types of debug screen, is dedicated to module and communication channel information.
- The right-hand part of the screen is dedicated to debugging data and parameters. This area, specific to the type of communication selected, is detailed in the documentation relating to the various types of communication.

How to Access the Screen

The debug mode may only be accessed in online mode.

Step	Action
1	Access the configuration screen.
2	Select the **Debug** mode by clicking on the corresponding tab.

Illustration

This area is used to access diagnostics for a communication channel.
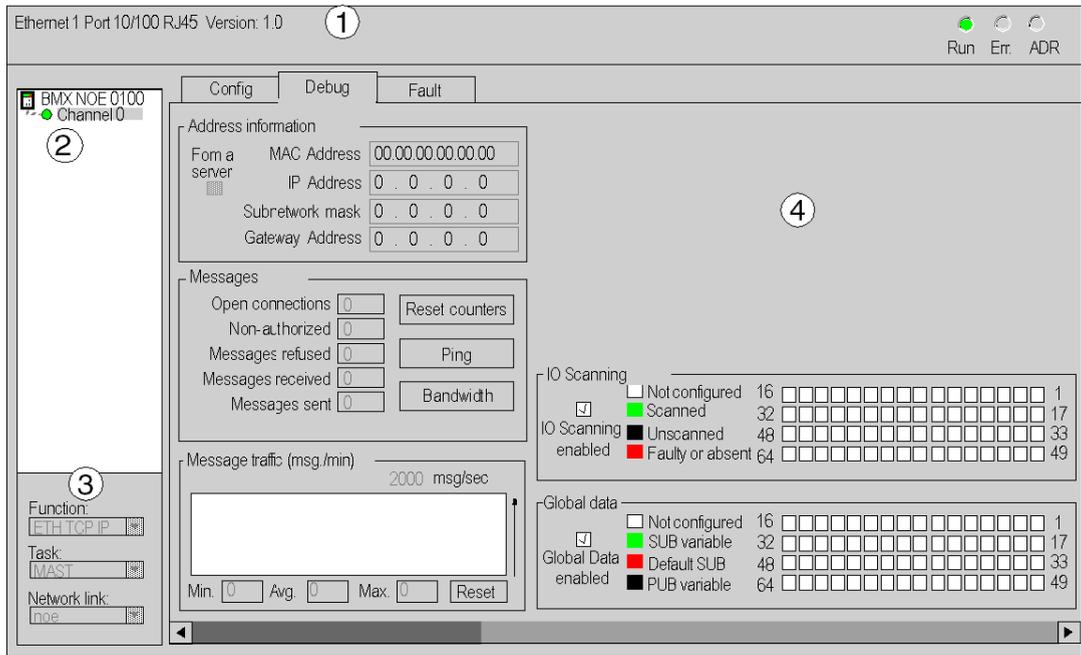
Description

The table below shows the various elements of the debug screen and their functions.

Zone	Function	
1: **Module**	module description zone	
2: **Channel**	channel selection zone	
3: parameters	general parameters zone	
4: **Debug** tab	**Address information**	• displays TCP/IP utility configuration • tests communication of the TCP/IP profile
	Messages	displays of the number of unacknowledged or refused messages
	Message traffic	displays the number of messages processed by the module per minute
	IO Scanning	display of the status for each remote input/output module
	Global Data	displays the status of Global Data variables

Communication Function Programming and Entry Help

11

How to Access a Function, Function Block or DFB-Type Specific Instruction

At a Glance

The application-specific function may be accessed:

- by direct entry of the instruction and its parameters in an operate block
- via the entry help function that may be accessed in the program editors (FBD, LD, IL, ST).

How to Call a Function

Step	Action
1	Access the required editor.
2	Depending on the editor, select one of the following methods to open the function library: ● Select the function to enter with the data editor. Once in the editor, right-click on the function (LD, FBD editors). ● Right-click in the program editor and select the option **FFB Input Assistant**. **Note:** The function entry help window appears.
3	Select the type of FFB required (if it is not already entered).
4	Then select the name of the instance (where necessary and if available).
5	Enter each parameter of the instruction (each instruction is expanded upon in the relevant application-specific documentation): ● In the **Entry field** field, in the **Prototype** area.
6	Validate with **Ok**.
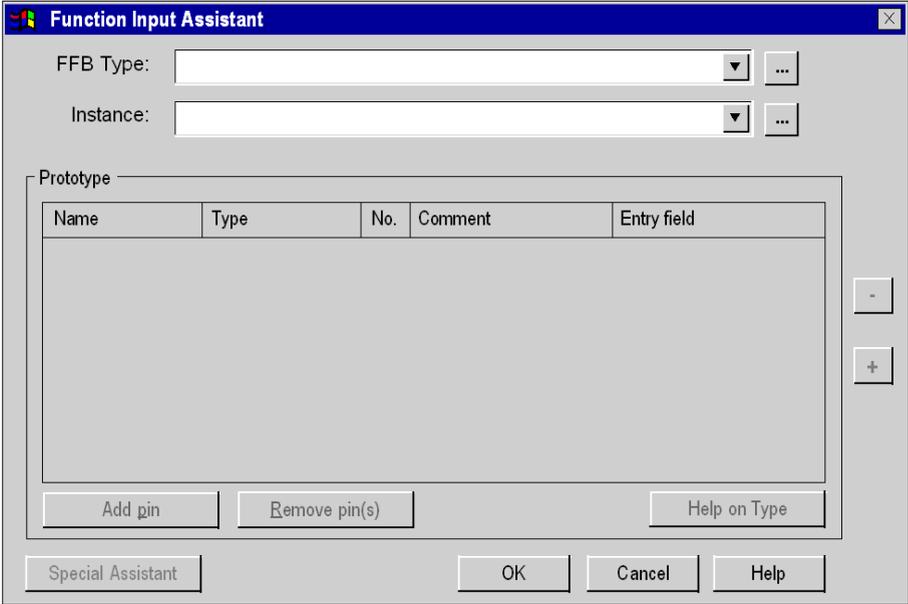
Index

A

addressing
  assistant, *75*
  IP, *39*
  Modicon M340, *43*
architectures, *27*

B

brigde, *57*
broadcast
  Modicon M340, *44*

C

configuring networks, *65*

F

function codes
  Modbus, *33*

G

Global Data, *15*

I

IO Scanning, *15*

M

messaging, *15*

T

topologies, *27*