# EtherNet/IP Fieldbus

Lexium MDrive Ethernet TCP/IP Products

V1.00, 08.2014


EtherNet/IP™ EtherNet/IP™ conformance tested

CE RoHS REACH

Schneider Electric

| EtherNet/IP Fieldbus Manual For Lexium MDrive | | |
|---|---|---|
| **Date** | **Revision** | **Changes** |
| 02/21/2013 | V1.00, 03.2013 | Initial release |
| 08/15/2014 | V1.00, 08.2014 | Changed Max Velocity (VM) from 5000000 to 2560000 steps/sec |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |

# Important information

This manual is part of the product.

Carefully read this manual and observe all instructions.

Keep this manual for future reference.

Hand this manual and all other pertinent product documentation over to all users of the product.

Carefully read and observe all safety instructions and the chapter “Before you begin - safety information”.

Some products are not available in all countries.

For information on the availability of products, please consult the catalog.

Subject to technical modifications without notice.

All details provided are technical data which do not constitute warranted qualities.

Most of the product designations are registered trademarks of their respective owners, even if this is not explicitly indicated.

This page intentionally left blank

# Table of Contents

# List of Figures



# List of Tables

# About this manual

The information provided in this manual supplements the product hardware manual.

*Source manuals*

The latest versions of the manuals can be downloaded from the Internet at:

http://motion.schneider-electric.com

Applicable manuals for Lexium Lexium MDrive Ethernet products are:

- MCode Programming and Software Reference manual
- MODBUS/TCP Fieldbus manual
- EtherNet/IP Fieldbus manual

*Graphic User Interface software*

For easier prototyping and development, a Graphic User Interface (GUI) is available for use with Lexium Lexium MDrive products. This software is available for download from the Internet at:

http://motion.schneider-electric.com

## Further reading

Recommended literature for further reading.

*Reference documents*

- The CIP Networks Library Volume 1 Common Industrial Protocol
- The CIP Networks Library Volume 3 DeviceNet Adaptation of CIP
- The CIP Networks Library Volume 2 DeviceNet Adaptation of CIP
- DeviceNet terms of Usage Agreement http://www.odva.org

*User Association*

- Open DeviceNet Vendor Association (ODVA) http://www.odva.org

Page intentionally left blank

# 1 Introduction

1

EtherNet/IP is a fieldbus based on TCP and UDP. EtherNet/IP extends Ethernet by an advanced industrial protocol (CIP, Common Industrial Protocol) as an application layer for automation applications - this way, Ethernet is excellently suited for industrial control . Products from different manufacturers can be networked without the need for special interface adaptation. The majority of the required network components correspond to the Ethernet components used in the PC world.

## 1.1 Fieldbus devices on the EtherNet/IP network

Different products with an EtherNet/IP interface can be operated in the same fieldbus. EtherNet/IP provides a common basis for interchanging commands and data between the network devices.

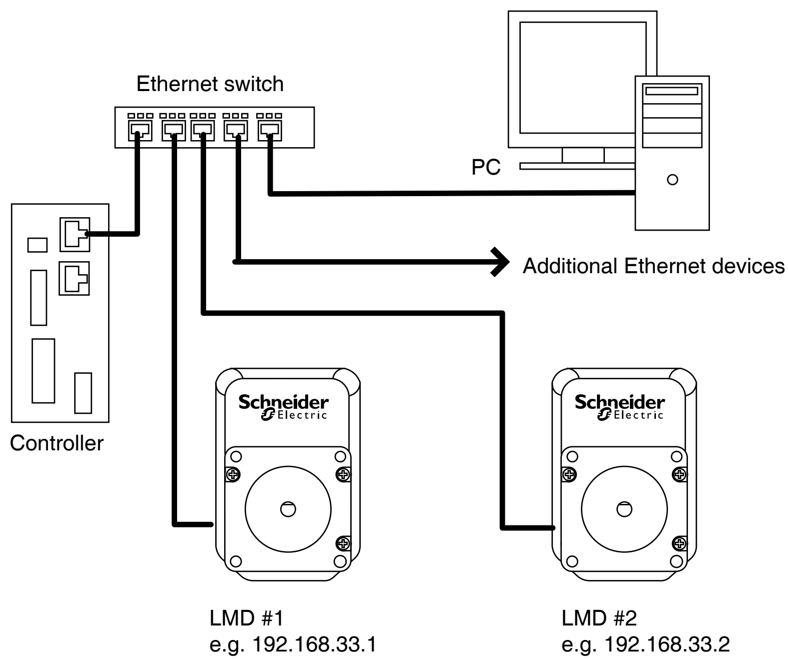


Figure 1.1: Example TCP/IP network with Lexium MDrive products.

V1.00, 08.2014

Page intentionally left blank

# 2 Before you begin - safety information

2

The information provided in this manual supplements the product manual. Carefully read the product manual before using the product.

## 2.1 Qualification of personnel

Only appropriately trained persons who are familiar with and understand the contents of this manual and all other pertinent product documentation are authorized to work on and with this product. In addition, these persons must have received safety training to recognize and avoid hazards involved. These persons must have sufficient technical training, knowledge and experience and be able to foresee and detect potential hazards that may be caused by using the product, by changing the settings and by the mechanical, electrical and electronic equipment of the entire system in which the product is used.

All persons working on and with the product must be fully familiar with all applicable standards, directives, and accident prevention regulations when performing such work.

## 2.2 Intended use

The functions described in this manual are only intended for use with the basic product; you must read and understand the appropriate product manual.

The product may only be used in compliance with all applicable safety regulations and directives, the specified requirements and the technical data.

Prior to using the product, you must perform a risk assessment in view of the planned application. Based on the results, the appropriate safety measures must be implemented.

Since the product is used as a component in an entire system, you must ensure the safety of persons by means of the design of this entire system (for example, machine design).

Operate the product only with the specified cables and accessories. Use only genuine accessories and spare parts.

Any use other than the use explicitly permitted is prohibited and can result in hazards.

Electrical equipment should be installed, operated, serviced, and maintained only by qualified personnel.

The product must NEVER be operated in explosive atmospheres (hazardous locations, Ex areas).

## 2.3 Hazard categories

Safety instructions to the user are highlighted by safety alert symbols in the manual. In addition, labels with symbols and/or instructions are attached to the product that alert you to potential hazards.

Depending on the seriousness of the hazard, the safety instructions are divided into 4 hazard categories.

| ⚠ 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, serious injury, or equipment damage. |

| ⚠ CAUTION |
| --- |
| CAUTION indicates a potentially hazardous situation, which, if not avoided, **can result** in injury or equipment damage. |

| CAUTION |
| --- |
| CAUTION used without the safety alert symbol, is used to address practices not related to personal injury (e.g. **can result** in equipment damage). |

## 2.4 Basic information

**⚠ DANGER**

**UNINTENDED CONSEQUENCES OF EQUIPMENT OPERATION**

When the system is started, the drives are usually out of the operator's view and cannot be visually monitored.

- Only start the system if there are no persons in the hazardous area.

**Failure to follow these instructions will result in death or serious injury.**

**⚠ WARNING**

**LOSS OF CONTROL**

- The designer of any control scheme must consider the potential failure modes of control paths and, for certain critical functions, provide a means to achieve a safe state during and after a path failure. Examples of critical control functions are emergency stop, overtravel stop, power outage and restart.
- Separate or redundant control paths must be provided for critical functions.
- System control paths may include communication links. Consideration must be given to the implication of unanticipated transmission delays or failures of the link.
- Observe all accident prevention regulations and local safety guidelines. 1)
- Each implementation of the product must be individually and thoroughly tested for proper operation before being placed into service.

**Failure to follow these instructions can result in death or serious injury.**

1) For USA: Additional information, refer to NEMA ICS 1.1 (latest edition), "Safety Guidelines for the Application, Installation, and Maintenance of Solid State Control" and to NEMA ICS 7.1 (latest edition), "Safety Standards for Construction and Guide for Selection, Installation and Operation of Adjustable-Speed Drive Systems".

## 2.5 Standards and terminology

Technical terms, terminology and the corresponding descriptions in this manual are intended to use the terms or definitions of the pertinent standards.

In the area of drive systems, this includes, but is not limited to, terms such as “safety function”, “safe state”, “fault”, “fault reset”, “failure”, “error”, “error message”, “warning”, “warning message”, etc.

Among others, these standards include:

- IEC 61800 series: “Adjustable speed electrical power drive systems”
- IEC 61158 series: “Industrial communication networks - Fieldbus specifications”
- IEC 61784 series: “Industrial communication networks - Profiles”
- IEC 61508 series: “Functional safety of electrical/electronic/ programmable electronic safety-related systems”

# 3 Basics

3

## 3.1 EtherNet/IP technology

EtherNet/IP devices are based on the same technology as devices on Ethernet networks in the computer world. For many applications, components from the existing company network can be used. However, it is recommended to use industrial switches for time-critical applications.

This manual is not intended to convey basic knowledge, for example in terms of network topology, data security or address assignment.

### 3.1.1 Data security

The larger the network into which the product is integrated, the greater the risk of unauthorized external access. The operator of the local network must take appropriate measures to prevent unauthorized access. Contact your network administrator prior to commissioning the product.

### 3.1.2 Basics

The ODVA ( is in charge of the specifications for the EtherNet/IP network and EtherNet/IP data terminal equipment. For more information on the ODVA see:

http://www.odva.org

*Number of nodes* The number of nodes in an EtherNet/IP network is theoretically unlimited; it depends on the subnet size and on whether or not a CIP router is used. For example, 254 nodes are possible in a class C subnet.

*Cable length* The maximum cable length is 100 m between EtherNet/IP terminal points and 90 m between infrastructure components. However, interference in industrial environments may require you to use shorter cables.

*Drive profiles* The product supports the following drive profiles:

- “Drive Profile Lexium MDrive”

*Communication means*

The product supports the following communication means:

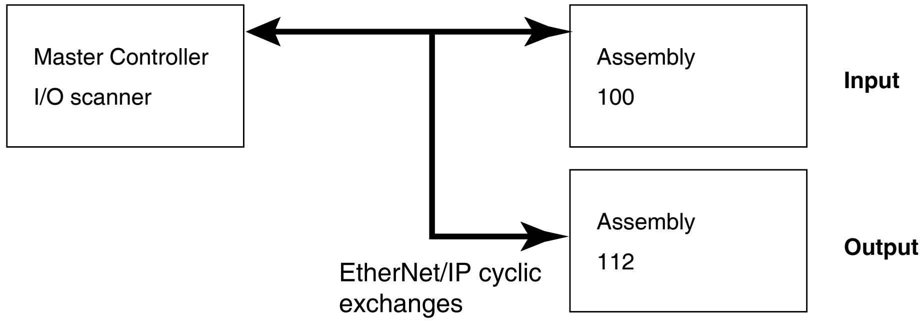


Figure 3.1: Overview of communication means

The product identifies itself as CIP “Generic Device” (Device Type = 0h).

*Data link layer*

The EtherNet/IP data link layer uses the transmission mechanisms as per IEEE 802.3 Standard Ethernet specification (edition 2002). This makes it possible to use a wide selection of available Ethernet components.

*Physical layer*

Industrial EtherNet/IP specifies minimum requirements in terms of ambient conditions, cabling and connectors, based on IEC, ANSI, TIA and EIA standards.

The connectors required for Industrial EtherNet/IP include RJ-45 connectors. Use shielded or unshielded CAT5 or CAT6 cables for Industrial EtherNet/IP.

Copper media may be used only for distances up to 100 m.

*Terms: Object class, instance, attribute, service*

The EtherNet/IP approach is object-oriented. CIP defines object classes; one or more instances (objects) can be derived from such object classes. The attributes of an object class or the instance derived from it contain the various parameters. Services are actions that are possible with these attributes.

*Example*

| Class | Instance | Attribute ID | Description | MCode |
|---|---|---|---|---|
| 0x64 | 0x01 | 0x01 | Reset enable | CE |

*CIP object model*

See section 7.

*Communication model*

EtherNet/IP uses the producer-consumer communication model. All nodes check the bus as to whether a data packet with the Identifier they support is available. Data packets that are sent by producers can only be received by the consumers of these packets.

*Groups of connections*

EtherNet/IP is a connection-oriented network. Connections must be established and managed between two nodes. There are 4 connection groups with different priorities:

| | |
|---|---|
| Group 1 | Top-priority process data (highest priority) |
| Group 2 | For simple master-slave connections |
| Group 3 | For Explicit Messages |
| Group 4 | Reserved group (lowest priority) |

Table 3.1: Connection groups

*Electronic Data Sheet*

An EDS file is a file in ASCII format. This file contains device-specific and vendor-specific descriptions of all parameters for a device. The EDS file also contains the fieldbus-specific communication parameters. The EDS file is required for commissioning.

### 3.1.3 Encapsulation

EtherNet/IP is based entirely on existing TCP/IP and UDP/IP technologies that are used without any modification. TCP/IP is used for the transmission of Explicit Messages while UDP/IP is used for I/O Messaging.

### 3.1.4 Messaging and message types

EtherNet/IP uses two types of messaging: Explicity Messaging and Implicit (I/O data) Messaging. EtherNet/IP defines several message types for communication. The product described here uses the “Explicit Message” and “I/O Message” message types.

*Explicit messages*

Explicit Messaging connections are point-to-point connections between two network nodes that are used for transactions of the type request - response. These connections are used to address parts of a device which are accessible via the network. The data field of Explicit Messages contains both protocol data and application-specific commands.

*I/O messages*

I/O Messages, also referred to as Implicit Messages, are transmitted via UDP/IP. I/O Message connections are often established as One-to- Many relationships in the producer-consumer Multicast model of Ether-Net/IP. The data fields of I/O Messages contain no protocol information, but only time-critical I/O data. I/O Messages are a lot smaller than Explicit Messages, thus allowing for much faster processing. These messages are used to to transport application-specific I/O data over the network at regular intervals. The meaning of the data is pre-defined at the time the connection is established. I/O Messages can contain socalled Assemblies of several parameters that can be transmitted with a single message. The parameters for configuring EtherNet/IP communication are described in chapter 5 “Commissioning”.

*Command processing: Transmit data and receive data*

The master sends a command to the drive system (slave) to execute a motion command, activate functions or request information from the slave. The slave executes the command and acknowledges it with a response message that may contain an error message if an error occurred.

The master device can send new commands as soon as it has received acknowledgement concerning the current command. Acknowledgement information and error messages are included in the transmitted data in bit-coded form.

The master must then continuously monitor for completion of the processing command by evaluating the acknowledgment from the slave. I/O messages are a special case. I/O messages are not acknowledged from the slave.

### 3.1.5 Communication via Explicit Message

An Explicit Message (EtherNet/IP-specific or vendor-specific) is used to read or write an individual parameter. See the product manual for an overview of all parameters.

The parameter is accessed by means of Class.Instance.Attribute as per CIP.

### 3.1.6 Communication via I/O Messages

An I/O Message is used for realtime exchange of process data. I/O messages lend themselves for motion commands. Transmission is very fast because the data is sent without administration data and a transmission acknowledgement from the recipient is not required.

The master can control the operating states of the slave by means of I/O Message, for example, enable and disable the power stage, trigger a “Quick Stop”, reset errors and activate operating modes.

Changing operating states and activating operating modes must be done separately. An operating mode can usually only be activated if the operating state is already “Operation Enabled”.

A new operating mode can only be activated when the motor is at a standstill.

*Output, Input*

Output and Input refer to the direction of data transmission from the perspective of the master.

- Output: Commands from the master to the slave, or originator to target.
- Input: Status messages from the slave to the master, or target to originator.

*Assembly*

I/O Messages contain a collection (Assembly) of different parameters that are transmitted with a single message.

The following Assemblies are defined for EtherNet/IP:

- Output Assembly, instance 112
- Input Assembly, instance 100

*Polled I/O Connection*

The Assemblies are used in a Polled I/O Connection . A Polled I/O Connection is initiated by the master with a Poll Command. The Slave responds with a Poll Response.

# 4 Installation

4

**⚠ WARNING**

**SIGNAL AND DEVICE INTERFERENCE**

Signal interference can cause unexpected responses of device.

- Install the wiring in accordance with the EMC requirements.
- Verify compliance with the EMC requirements.

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

For information on installation of the device and connecting the device to the fieldbus see the product hardware manual.

Page intentionally left blank

# 5 Configuration

5

Configuring the Lexium MDrive Ethernet devices is accmplished using the TCP/IP Configuration utility.

Please refer to the Lexium MDrive Software Suite manual.

Page intentionally left blank

# 6 Diagnostics and troubleshooting

6

## 6.1 Fieldbus communication error diagnostics

A properly operating fieldbus is essential for evaluating operating and error messages.

*Connections for fieldbus mode*

If the product cannot be addressed via the fieldbus, first check the connections. The product manual contains the technical data of the device and information on network and device installation. Check the following:

- Power connections to the device
- Fieldbus cable and fieldbus wiring
- Network connection to the device

You can also use the TCP/IP Configuration Utility for troubleshooting.

*Fieldbus function test*

If the connections are correct, check the settings for the fieldbus addresses. After correct configuration of the transmission data, test fieldbus mode.

1) In addition to the master that knows the product via the EDS file and addressing, activate a bus monitor that, as a passive device, displays messages.
2) Switch the supply voltage off and on.
3) Observe the network messages that are generated briefly after the supply voltage is switched on. A bus monitor can be used to record the elapsed time between

*Addressing, parameterization*

If it is impossible to connect to a device, check the following:

1) Addressing: Each network device must have a unique IP address. and the correct subnet mask.
2) Parameterization: “Vendor ID” and “Product Code” must match the values stored in the EDS file.

## 6.2 Status LEDs

The Lexium MDrive with Ethernet has two dual-color (red/green) LEDs visible from the back of the drive to give status and error indication of the EtherNet/IP connection.

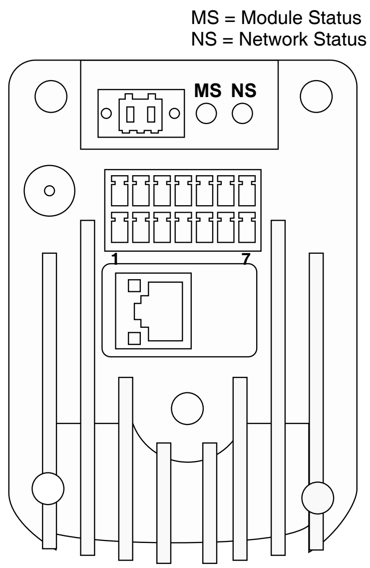


Figure 6.1: Status indicator LEDs

| Color | State | Description |
|---|---|---|
| ***LED 1 – Network Status (NS)*** | | |
| None | Off | No power |
| | Flashing | Recoverable fault or I/O connection timed out |
| Green | Solid | Normal runtime operation (I/O connection allocated) |
| | Flashing | Device is idle or not allocated to a client (PLC) |
| Red/green | Alternating | Power-up self test in progress |
| ***LED 2 – Module Status (MS)*** | | |
| None | Off | No power |
| Red | Solid | Unrecoverable fault |
| | Flashing | Minor, recoverable fault |
| Green | Solid | Device operational |
| | Flashing | Standby, device has not been configured |
| Red/green | Alternating | Power-up self test in progress |

NOTE: The term I/O, in this context refers to EtherNet/IP communications protocol and is unrelated to the hardware Input - output points.

Table 6.1: Status indicator LEDs

V1.00, 08.2014

# 7 Object model

7

## 7.1 Data types used in this object model

| Data Type | Description |
|---|---|
| SINT | Signed 8-bit integer |
| INT | Signed 16-bit integer |
| DINT | Signed 32-bit integer |
| USINT | Unsigned 8-bit integer |
| UINT | Unsigned 16-bit integer |
| UDINT | Unsigned 32-bit integer |
| STRING | Character string (1 byte per character) |
| SHORTSTRING*nn* | Character string (1 byte in length; up to *nn* characters) |
| BOOL | 1 bit |
| BYTE | Bit string (8-bits) |
| WORD | Bit string (16-bits) |
| DWORD | Bit string (32-bits) |
| REAL | IEEE 32-bit single precision floating point |

Table 7.1: Data types identification

## 7.2 Identity object ($01_h$ – 1 instance)

The following tables contain the attribute, status, and common services information for the identity object.

| Instance | Attribute ID | Name | Data type | Data value | Access rule |
|---|---|---|---|---|---|
| 0x00 | 0x01 | Revision | UINT | 1 | Get |
| 0x01 | 0x01 | Vendor number | UINT | 243 | Get |
| | 0x02 | Device type | UINT | 43 | Get |
| | 0x03 | Product code number | UINT | 15362 | Get |
| | 0x04 | Product major revision<br>Product minor revision | USINT<br>USINT | 02<br>22 | Get |
| | 0x05 | Status | WORD | See Table 7.4 | Get |
| | 0x06 | Serial number | UDINT | Unique 32 bit value | Get |
| | 0x07 | Product name | SHORTSTRING32 | Lexium MDrive | Get |

Table 7.2: Identity object (01h – 1 instance)

| Service code | Implemented for | | Service name |
|---|---|---|---|
| | Class level | Instance levels | |
| 0x05 | No | Yes | Reset |
| 0x0E | Yes | Yes | Get_Attribute_Single |
| 0x10 | No | Yes | Set_Attribute Single |

Table 7.3: Identity object's common services

### 7.2.1 Identity object attribute 0x05 (Status)

This attribute represents the current status of the entire device. Its value changes as the state of the device changes. The status attribute is a WORD, with the following bit definitions:

| 15 ... 12 | 11 | 10 | 9 | 8 | 7 ... 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|
| Rsrvd | Maj UF | Maj RF | Min.UF | Min. RF | Ext Stat | Rsrvd | Config | Rsrvd | Owned |

| Bit(s) | Name | Meaning |
|---|---|---|
| 0 | | 1 indicates the device (or an object within the device) has an owner. Within the Master/ Slave paradigm the setting of this bit means that the Predefined Master/Slave Connection Set has been allocated to a master. |
| 1 | | Reserved, always 0 |
| 2 | Configured | 1 indicates the application of the device has been configured to do something different than the “out–of–box” default. This shall not include configuration of the communications. |
| 3 | | Reserved, always 0 |
| 4 – 7 | Extended Device Status | Vendor–specific or as defined by table below. The EDS shall indicate if the device follows the public definition for these bits using the DeviceStatusAssembly keyword in the [Device] section of the EDS. If these bits are vendor specific then they shall be enumerated in the EDS using the Assembly and Parameter sections. |
| 8 | Minor Recoverable Fault | 1 indicates the device detected a problem with itself, which is thought to be recoverable. The problem does not cause the device to go into one of the faulted states. |
| 9 | Minor Unrecoverable Fault | 1 indicates the device detected a problem with itself, which is thought to be unrecoverable. The problem does not cause the device to go into one of the faulted states. |
| 10 | Major Recoverable Fault | TRUE indicates the device detected a problem with itself, which caused the device to go into the “Major Recoverable Fault” state. |
| 11 | Major Unrecoverable Fault | TRUE indicates the device detected a problem with itself, which caused the device to go into the “Major Unrecoverable Fault” state. |
| 12 – 15 | | Reserved, always 0 |

Table 7.4: Bit definitions for status instance attribute of identity object

## 7.3 Assembly object ($04_h$ – 4 instances)

Note that instance attributes show the default mapping of parameters to an attribute. This mapping may be changed to any MCode mnemonic of the same datatype using the TCP/IP Configuration Utility. See Section 5: Commissioning.

<table>
<tr><th>Instance</th><th>Attribute ID</th><th>Name</th><th>Data type</th><th>Data value</th><th>Access rule</th></tr>
<tr><td>Class<br>(Instance 0)</td><td>1</td><td>Revision</td><td>UINT</td><td>2</td><td>Get</td></tr>
<tr><td></td><td>2</td><td>Max instance</td><td>UINT</td><td>0xFF</td><td>Get</td></tr>
<tr><td>Input (T→O)<br>Instance 100</td><td>3</td><td colspan="3">
<table>
<tr><th>Bytes</th><th>MCode mnemonic</th><th colspan="2">Description</th></tr>
<tr><td>BOOL</td><td>EF</td><td>Error flag</td><td rowspan="2">Same byte</td></tr>
<tr><td>BOOL</td><td>MV</td><td>Moving flag</td></tr>
<tr><td>USINT</td><td>IN</td><td colspan="2">Read inputs as a group</td></tr>
<tr><td>USINT</td><td>Pad</td><td colspan="2"></td></tr>
<tr><td>USINT</td><td>Pad</td><td colspan="2"></td></tr>
<tr><td>DINT</td><td>C1</td><td colspan="2">Position counter</td></tr>
<tr><td>DINT</td><td>C2</td><td colspan="2">Encoder counter</td></tr>
<tr><td>DINT</td><td>LL</td><td colspan="2">Lead/Lag</td></tr>
</table>
</td><td>Get</td></tr>
<tr><td>Output<br>(O→T)<br>Instance 112</td><td>3</td><td colspan="3">
<table>
<tr><th>Bytes</th><th>MCode mnemonic</th><th>Description</th></tr>
<tr><td>UDINT</td><td>A</td><td>Acceleration</td></tr>
<tr><td>UDINT</td><td>D</td><td>Deceleration</td></tr>
<tr><td>DINT</td><td>MA</td><td>Move to absolute position</td></tr>
<tr><td>DINT</td><td>MR</td><td>Move to relative position</td></tr>
<tr><td>USINT</td><td>RC</td><td>Run current percent</td></tr>
<tr><td>USINT</td><td>HC</td><td>Hold current percent</td></tr>
<tr><td>BYTE</td><td>OT</td><td>Write digital outputs</td></tr>
<tr><td>USINT</td><td>Pad</td><td></td></tr>
<tr><td>DINT</td><td>SL</td><td>Slew</td></tr>
<tr><td>UDINT</td><td>VI</td><td>Initial (starting) velocity</td></tr>
<tr><td>UDINT</td><td>VM</td><td>Maximum (final) velocity</td></tr>
<tr><td>USINT</td><td>TQ</td><td>Torque percent</td></tr>
<tr><td>USINT</td><td>EF</td><td>Error Flag</td></tr>
</table>
</td><td>Get/set</td></tr>
<tr><td>128 (0x80)</td><td></td><td>Input only heartbeat[1]</td><td>Heartbeat</td><td>0</td><td>n/a</td></tr>
<tr><td>129 (0x81)</td><td></td><td>Listen only heartbeat[2]</td><td>Heartbeat</td><td>0</td><td>n/a</td></tr>
<tr><td>Unused (n)</td><td></td><td>Configuration[3]</td><td></td><td></td><td></td></tr>
</table>

1. This instance allows clients 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 7.5: Assembly object (04h – 4 instances)

| Service code | Implemented for | | Service name |
|---|---|---|---|
| | Class level | Instance levels | |
| 0x0E | Yes | Yes | Get_Attribute_Single |
| 0x10 | Yes | Yes | Set_Attribute Single |

Table 7.6: Assembly object’s common services

## 7.4 TCP object ($F5_h$ – 1 instance)

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

| Instance | Attribute ID | Name | Data type | Data value | Access rule |
|---|---|---|---|---|---|
| Class (Instance 0) | 1 | Revision | UINT | 1 | Get |
| Instance 1 | 1 | Status* | DWORD | See Table 5-3.2* | Get |
| | 2 | Configuration capability* | DWORD | See Table 5-3.4* | Get |
| | 3 | Configuration control* | DWORD | 0x0 | Get/set |
| | 4 | Physical Link Object * Structure of | | Below are defaults | Get |
| | | Path Size | UINT | 0x2 | |
| | | Path | Array of Word | 0x20 0xF6 0x24 0x01 | |
| | 5 | Interface configuration* Structure of | | Below are defaults | Get |
| | | IP Address | UDINT | 192.168.33.1 | |
| | | Network Mask | UDINT | 255.255.0.0 | |
| | | Gateway Address | UDINT | 192.168.1.200 | |
| | | Name Server | UDINT | 0x0 | |
| | | Name Server 2 | UDINT | 0x0 | |
| | | Domain Name Size | UINT | 0x0 | |
| | | Domain Name | STRING | 0x0 | |
| | 6 | Host name* Structure of | | Below are defaults | Get |
| | | Host Name Size | UINT | 0x0 | |
| | | Host Name | STRING | 0x0 | |

* For more details on these attributes, see ***Volume 2: EtherNet/IP Adaptation of CIP,*** Section 5-3.2 from ODVA. Tables reference in the data value column are located in that document.

Table 7.7: TCP object (05h – 1 instance)

| Service code | Implemented for | | Service name |
|---|---|---|---|
| | Class level | Instance levels | |
| 0x0E | Yes | Yes | Get_Attribute_Single |
| 0x10 | No | Yes | Set_Attribute Single |

Table 7.8: Assembly object’s common services

## 7.5 Ethernet Link object ($F6_h$ – 1 instance)

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

| Instance | Attribute ID | Name | Data type | Data value | Access rule |
|---|---|---|---|---|---|
| Class (Instance 0) | 1 | Revision | UINT | 1 | Get |
| Instance 1 | 1 | Interface speed* | DWORD | 0 | Get |
| | 2 | Interface flag* | DWORD | See Table 5-4.44* | Get |
| | 3 | Physical address | ARRAY | MAC Address | Get |

*For more details on these attributes, see ***Volume 2: EtherNet/IP Adaptation of CIP,*** Section 5-3.2 from ODVA. Tables reference in the data value column are located in that document.

Table 7.9: Ethernet object (0xF6 – 1 instance)

| Service code | Implemented for | | Service name |
|---|---|---|---|
| | Class level | Instance levels | |
| 0x0E | Yes | Yes | Get_Attribute_Single |

Table 7.10: Assembly object's common services

Page intentionally left blank

# 8 Manufacturer specific objects

8

## 8.1 Preliminary

This section contains the objects specific to the Lexium Lexium MDrive Motion Control devices.

It is divided into Object classes by functional grouping, these are:

| Object | Description |
|---|---|
| 0x64 | Setup instructions |
| 0x65 | Miscellaneous instructions and flags |
| 0x66 | Motion instructions and flags |
| 0x67 | I/O instruction variables and flags |
| 0x68 | Position related instructions and flags |
| 0x69 | Encoder related instructions and flags |
| 0x6A | Hybrid specific instructions and flags |

Table 8.1: Object class groupings

| Access | Description |
|---|---|
| RO | Readable only |
| WO | Writable only |
| WONE | Writable only , no equal sign |
| RW | Readable and writable, unconditional |
| RW_IO | Readable always, Writable only if I/O connection present |

Table 8.2: Access types identification

NOTE: References in the section to “I/O” refer to the control of the hardware input/output points on the device.

### 8.1.1 MCode compatibility

Each attribute on the object class grouping references a 1 or 2 character MCode mnemonic.

Please reference the ***Lexium MCode Programming and Reference Manual,*** which is located on the web site at http://www.motion.schneider-electric.com for detailed usage instructions, value ranges and restrictions.

## 8.2 Setup object ($64_h$ – 1 instance)

The Setup object contains attributes configuring basic setup parameters.

| Instance | Attribute ID | Description | Data type | MCode | Access |
|---|---|---|---|---|---|
| Class (Instance 0) | 0x01 | Revision | UINT | — | RO |
| Instance 1 | 0x01 | Software reset enable | USINT | CE | RW |
| | 0x02 | Output clock width | USINT | CW | RW |
| | 0x03 | Drive enable | BOOL | DE | RW |
| | 0x04 | Trip on relative position | USINT | TR | RW |
| | 0x05 | Warning temperature | USINT | WT | RW |

Table 8.3: Setup object (64h – 1 instance)

## 8.3 Miscellaneous object ($65_h$ – 1 instance)

The Miscellaneous object contains attributes for reading and/or setting miscellaneous variables and flags.

| Instance | Attribute ID | Description | Data type | MCode | Access |
|---|---|---|---|---|---|
| Class (Instance 0) | 0x01 | Revision | UINT | — | RO |
| | 0x01 | Output clock width | USINT | CW | RW |
| | 0x02 | Error flag | BOOL | EF | RO |
| | 0x03 | Read/clear error condition | UNIT16 | ER | RW |
| | 0x04 | Read internal temperature | INT | IT | RO |
| | 0x05 | Part Number | STRING | PN | RO |
| | 0x06 | Read/set user register 1 value | DINT | R1 | RW |
| | 0x07 | Read/set user register 2 value | DINT | R2 | RW |
| | 0x08 | Read/set user register 3 value | DINT | R3 | RW |
| | 0x09 | Read/set user register 4 value | DINT | R4 | RW |
| | 0x0A | Read device serial number | STRING | SN | RO |
| | 0x0B | Read hardware/firmware version number | STRING | VR | RO |
| | 0x0C | Read device voltage | STRING | VT | RO |

Table 8.4: Miscellaneous object ($65_h$ – 1 instance)

## 8.4 Motion object ($66_h$ – 1 instance)

The Motion object contains attributes for issuing motion commands and reading and/or writing motion related variables or flags.

| Instance | Attribute ID | Description | Data type | MCode | Access |
|---|---|---|---|---|---|
| Class (Instance 0) | 0x01 | Revision | UINT | — | RO |
| Instance 1 | 0x01 | Read/set acceleration | UDINT | A | RW_IO |
| | 0x02 | Read/set deceleration | UDINT | D | RW_IO |
| | 0x03 | Read/set motor holding current percent | USINT | HC | RW_IO |
| | 0x04 | Read/set holding current time delay | UINT | HT | RW |
| | 0x05 | Read/set jog mode enable/disable | BOOL | JE | RW |
| | 0x06 | Read/set limit switch response mode | USINT | LM | RW |
| | 0x07 | Command absolute move | DINT | MA | RW_IO |
| | 0x08 | Moving to position flag | BOOL | MT | RO |
| | 0x09 | Command relative move | DINT | MR | RW_IO |
| | 0x0A | Read/set microstep resolution | UINT | MS | RW |
| | 0x0B | Read/set motor settling delay time | UINT | MT | RW |
| | 0x0C | Read axis in motion flag | BOOL | MV | RO |
| | 0x0D | Read/set motor run current percent | USINT | RC | RW_IO |
| | 0x0E | Command slew at constant velocity | DINT | SL | RW_IO |
| | 0x0E | Read current velocity | DINT | V | RO |
| | 0x10 | Read velocity changing flag | BOOL | VC | RO |
| | 0x11 | Read/set initial (starting) velocity | UDINT | VI | RW_IO |
| | 0x12 | Read/set maximum (terminal) velocity | UDINT | VM | RW_IO |

Table 8.5: Motion object ($65_h$ – 1 instance)

V1.00, 08.2014

## 8.5 I/O object ($67_h$ – 1 instance)

The I/O object contains attributes for issuing I/O commands and reading and/or writing I/O related variables or flags.

**NOTE: Attributes in this object relate solely to the configuration and control of the hardware Input–Output points and have no bearing on EtherNet/IP communications protocol I/O messaging.**

| Instance | Attribute ID | Description | Data type | MCode | Access |
|---|---|---|---|---|---|
| Class (Instance 0) | 0x01 | Revision | UINT | — | RO |
| Instance 1 | 0x01 | Read/set attention output mask | USINT | AO | RW |
| | 0x02 | Read/set digital filtering for input 1 | USINT | D1 | |
| | 0x03 | Read/set digital filtering for input 2 | USINT | D2 | RW |
| | 0x04 | Read/set digital filtering for input 3 | USINT | D3 | RW |
| | 0x05 | Read/set digital filtering for input 4 | USINT | D4 | RW |
| | 0x06 | Read/set digital filtering for analog input | USINT | D5 | RW |
| | 0x07 | Read/set filtering for the capture input | USINT | FC | RW |
| | 0x08 | Read/set filtering for the motion inputs | USINT | FM | RW |
| | 0x09 | Read the state of input 1 | BOOL | I1 | RO |
| | 0x0A | Read the state of input 2 | BOOL | I2 | RO |
| | 0x0B | Read the state of input 3 | BOOL | I3 | RO |
| | 0x0C | Read the state of input 4 | BOOL | I4 | RO |
| | 0x0D | Read the value of the analog input | UINT | I5 | RO |
| | 0x0E | Read inputs as a group | BOOL | IN | RO |
| | 0x0F | Setup input funtion | UINT | IS | RW |
| | 0x10 | Set the state of output 1 | BOOL | O1 | WONE |
| | 0x11 | Set the state of output 2 | BOOL | O2 | WONE |
| | 0x12 | Set the state of output 3 | BOOL | O3 | WONE |
| | 0x13 | Read the output fault flag | BOOL | OF | RO |
| | 0x14 | Setup output function | UINT | OS | RW |
| | 0x15 | Write the binary state of the outputs as a group (NEMA 23 and 34 only) | UINT | OT | WONE |

Table 8.6: I/O object ($67_h$ – 1 instance)

V1.00, 08.2014

## 8.6 Position object ($68_h$ – 1 instance)

The position object contains attributes for reading and/or writing position related variables or flags.

| Instance | Attribute ID | Description | Data type | MCode | Access |
|---|---|---|---|---|---|
| Class (Instance 0) | 0x01 | Revision | UINT | — | RO |
| Instance 1 | 0x01 | Read/set counter 1 (position) | DINT | C1 | RW |
| | 0x02 | Read/set homing mode | USINT | HM | |
| | 0x03 | Read/set position | DINT | P | RW |
| | 0x04 | Read position capture at trip | DINT | PC | RO |
| | 0x05 | Read/set trip enable | USINT | TE | RW |

Table 8.7: Position object ($68_h$ – 1 instance)

## 8.7 Encoder object ($69_h$ – 1 instance)

The encoder object contains attributes for reading and/or writing encoder related variables or flags.

| Instance | Attribute ID | Description | Data type | MCode | Access |
|---|---|---|---|---|---|
| Class (Instance 0) | 0x01 | Revision | UINT | — | RO |
| Instance 1 | 0x01 | Read/set counter 2 (encoder) | DINT | C1 | RW |
| | 0x02 | Read/set encoder deadband | UINT | DB | RW |
| | 0x03 | Enable/disable encoder functions | BOOL | EE | RW |
| | 0x04 | Read/set home to index mode | USINT | HI | RW |
| | 0x05 | Read encoder index mark | BOOL | I6 | RO |
| | 0x06 | Enable/disable position maintenance (non-hybrid only) | BOOL | PM | RW |
| | 0x07 | Read/set stall factor (non-hybrid only) | UINT | SF | RW |
| | 0x08 | Read/set stall detect mode (non-hybrid only) | BOOL | SM | RW |
| | 0x09 | Read stall flag (non-hybrid only) | BOOL | ST | RO |

Table 8.8: Encoder object ($69_h$ – 1 instance)

## 8.8 Hybrid specific object ($6A_h$ – 1 instance)

The hybrid specific object contains attributes for issuing hybrid commands and reading and/or writing hybrid related variables or flags.

| Instance | Attribute ID | Description | Data type | MCode | Access |
|---|---|---|---|---|---|
| Class (Instance 0) | 0x01 | Revision | UINT | — | RO |
| Instance 1 | 0x01 | Read hybrid status | USINT | AF | RO |
| | 0x02 | Read/set hybrid operating mode | USINT | AS | RW |
| | 0x03 | Read/set control bounds | USINT | CB | RW |
| | 0x04 | Clear locked rotor flag | BOOL | CF | CMD |
| | 0x05 | Read/set remote encoder line count | UINT | EL | RW |
| | 0x06 | Read/set lead limit | UDINT | LD | RW |
| | 0x07 | Read lead/lag position error | DINT | LL | RO |
| | 0x08 | Read/set lag limit | UDINT | LG | RW |
| | 0x09 | Read state of rotor locked/unlocked | BOOL | LR | RO |
| | 0x0A | Read set locked rotor timeout time | UINT | LT | RW |
| | 0x0B | Read set make-up speed | UDINT | MF | RW |
| | 0x0C | Read/set make-up mode | STRING | MU | RW |
| | 0x0D | Command calibration start | USINT | SC | WONE |
| | 0x0E | Read/set torque direction | BOOL | TD | RW |
| | 0x0F | Read/set torque current percent | USINT | TQ | RW |
| | 0x10 | Read/set torque speed | USINT | TS | RW |

Table 8.9: Hybrid specific object ($6A_h$ – 1 instance)

# 9 Glossary

9

## 9.1 Units and conversion tables

The value in the specified unit (left column) is calculated for the desired unit (top row) with the formula (in the field).

Example: conversion of 5 meters [m] to yards [yd]
5 m / 0.9144 = 5.468 yd

### 9.1.1 Length

| | in | ft | yd | m | cm | mm |
|---|---|---|---|---|---|---|
| **in** | — | / 12 | / 36 | * 0.0254 | * 2.54 | * 25.4 |
| **ft** | * 12 | — | / 3 | * 0.30479 | * 30.479 | * 304.79 |
| **yd** | * 36 | * 3 | — | * 0.9144 | * 91.44 | * 914.4 |
| **m** | / 0.0254 | / 0.30479 | / 0.9144 | — | * 100 | * 1000 |
| **cm** | / 2.54 | / 30.479 | / 91.44 | / 100 | — | * 10 |
| **mm** | / 25.4 | / 304.79 | / 914.4 | / 1000 | / 10 | — |

### 9.1.2 Mass

| | lb | oz | slug | kg | g |
|---|---|---|---|---|---|
| **lb** | — | * 16 | * 0.03108095 | * 0.4535924 | * 453.5924 |
| **oz** | / 16 | — | * $1.942559*10^{-3}$ | * 0.02834952 | * 28.34952 |
| **slug** | / 0.03108095 | * $1.942559*10^{-3}$ | — | * 14.5939 | * 14593.9 |
| **kg** | / 0.453592370 | / 0.02834952 | / 14.5939 | — | * 1000 |
| **g** | / 453.592370 | / 28.34952 | / 14593.9 | / 1000 | — |

### 9.1.3 Force

| | lb | oz | p | dyne | N |
|---|---|---|---|---|---|
| **lb** | — | * 16 | * 453.55358 | * 444822.2 | * 4.448222 |
| **oz** | / 16 | — | * 28.349524 | * 27801 | * 0.27801 |
| **p** | / 453.55358 | / 28.349524 | — | * 980.7 | * $9.807*10^{-3}$ |
| **dyne** | / 444822.2 | / 27801 | / 980.7 | — | / $100*10^{3}$ |
| **N** | / 4.448222 | / 0.27801 | / $9.807*10^{-3}$ | * $100*10^{3}$ | — |

V1.00, 08.2014

### 9.1.4 Power

| | HP | W |
|---|---|---|
| **HP** | — | * 745.72218 |
| **W** | / 745.72218 | — |

### 9.1.5 Rotation

| | $min^{-1}$ (RPM) | rad/s | deg./s |
|---|---|---|---|
| **$min^{-1}$ (RPM)** | — | * π / 30 | * 6 |
| **rad/s** | * 30 / π | — | * 57.295 |
| **deg./s** | / 6 | / 57.295 | — |

### 9.1.6 Torque

| | lb·in | lb·ft | oz·in | Nm | kp·m | kp·cm | dyne·cm |
|---|---|---|---|---|---|---|---|
| **lb·in** | — | / 12 | * 16 | * 0.112985 | * 0.011521 | * 1.1521 | * $1.129*10^6$ |
| **lb·ft** | * 12 | — | * 192 | * 1.355822 | * 0.138255 | * 13.8255 | * $13.558*10^6$ |
| **oz·in** | / 16 | / 192 | — | * $7.0616*10^{-3}$ | * $720.07*10^{-6}$ | * $72.007*10^{-3}$ | * 70615.5 |
| **Nm** | / 0.112985 | / 1.355822 | / $7.0616*10^{-3}$ | — | * 0.101972 | * 10.1972 | * $10*10^6$ |
| **kp·m** | / 0.011521 | / 0.138255 | / $720.07*10^{-6}$ | / 0.101972 | — | * 100 | * $98.066*10^6$ |
| **kp·cm** | / 1.1521 | / 13.8255 | / $72.007*10^{-3}$ | / 10.1972 | / 100 | — | * $0.9806*10^6$ |
| **dyne·cm** | / $1.129*10^6$ | / $13.558*10^6$ | / 70615.5 | / $10*10^6$ | / $98.066*10^6$ | / $0.9806*10^6$ | — |

### 9.1.7 Moment of inertia

| | $lb·in^2$ | $lb·ft^2$ | $kg·m^2$ | $kg·cm^2$ | $kp·cm·s^2$ | $oz·in^2$ |
|---|---|---|---|---|---|---|
| **$lb·in^2$** | — | / 144 | / 3417.16 | / 0.341716 | / 335.109 | * 16 |
| **$lb·ft^2$** | * 144 | — | * 0.04214 | * 421.4 | * 0.429711 | * 2304 |
| **$kg·m^2$** | * 3417.16 | / 0.04214 | — | * $10*10^3$ | * 10.1972 | * 54674 |
| **$kg·cm^2$** | * 0.341716 | / 421.4 | / $10*10^3$ | — | / 980.665 | * 5.46 |
| **$kp·cm·s^2$** | * 335.109 | / 0.429711 | / 10.1972 | * 980.665 | — | * 5361.74 |
| **$oz·in^2$** | / 16 | / 2304 | / 54674 | / 5.46 | / 5361.74 | — |

### 9.1.8 Temperature

| | °F | °C | K |
|---|---|---|---|
| **°F** | — | (°F - 32) * 5/9 | (°F - 32) * 5/9 + 273.15 |
| **°C** | °C * 9/5 + 32 | — | °C + 273,15 |
| **K** | (K - 273.15) * 9/5 + 32 | K - 273.15 | — |

### 9.1.9 Conductor cross section

| AWG | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $mm^2$ | 42.4 | 33.6 | 26.7 | 21.2 | 16.8 | 13.3 | 10.5 | 8.4 | 6.6 | 5.3 | 4.2 | 3.3 | 2.6 |

| AWG | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $mm^2$ | 2.1 | 1.7 | 1.3 | 1.0 | 0.82 | 0.65 | 0.52 | 0.41 | 0.33 | 0.26 | 0.20 | 0.16 | 0.13 |

## 9.2 Terms and Abbreviations

*AC* Alternating current

*Acceleration* The time rate of change of velocity with respect to a fixed reference frame. The commanded step rate is started at a base velocity and accelerated at a slew velocity at a defined and controlled rate or rate of changes.

*ASCII* American Standard Code for Information Interchange. Standard for coding of characters.

*Back Electro-Motive Force (Back EMF)* Also known as regeneration current, the reversed bias generated by rotation of the magnetic field across a stator's windings. Sometimes referred to as counter EMF.

*CAN* (Controller Area Network), standardized open fieldbus as per ISO 11898, allows drives and other devices from different manufacturers to communicate.

*CANopen* CANopen is a CAN-based higher layer protocol. It was developed as a standardized embedded network with highly flexible configuration capabilities. CANopen was designed motion oriented machine control networks, such as handling systems. It is used in many various fields, such as medical equipment, off-road vehicles, maritime electronics, public transportation, building automation, etc

*Closed Loop System* In motion control, this term describes a system wherein a velocity or position (or both) sensor is used to generate signals for comparison to desired parameters. For cases where loads are not predictable, the closed loop feedback from an external encoder to the controller may be used for stall detection, position maintenance or position verification.

*Daisy Chain* This term is used to describe the linking of several devices in sequence, such that a single signal stream flows through one device and on to another

*DC* Direct current

*Deadband* A range of input signals for which there is no system response.

*Default value* Factory setting.

*Detent Torque* The periodic torque ripple resulting from the tendency of the magnetic rotor and stator poles to align themselves to positions of minimal reluctance. The measurement is taken with all phases de-energized.

*Direction of rotation* Rotation of the motor shaft in a clockwise or counterclockwise direction of rotation. Clockwise rotation is when the motor shaft rotates clockwise as you look at the end of the protruding motor shaft.

*DOM* The Date of manufacturing on the nameplate of the device is shown in the format DD.MM.YY,
e.g. 31.12.06 (December 31, 2006).

*Duty Cycle* For a repetitive cycle, the ratio of on time to total cycle time.

*EMC* Electromagnetic compatibility

*Encoder* Sensor for detection of the angular position of a rotating component. The motor encoder shows the angular position of the rotor.

*Error class* Classification of errors into groups. The different error classes allow for specific responses to faults, e.g. by severity.

*Fatal error* In the case of fatal error, the drive is not longer able to control the motor, so that an immediate switch-off of the drive is necessary.

*Fault* Operating state of the drive caused as a result of a discrepancy between a detected (computed, measured or signaled) value or condition and the specified or theoretically correct value or condition.

*Fault reset* A function used to restore the drive to an operational state after a detected fault is cleared by removing the cause of the fault so that the fault is no longer active (transition from state “Fault” to state “Operation Enable”).

*Forcing* Forcing switching states of inputs/outputs. Forcing switching states of inputs/outputs.

*Full Duplex* The transmission of data in two directions simultaneously. For example, a telephone is a full-duplex device because both parties can talk at the same time.

*Ground Loop* A ground loop is any part of the DC return path (ground) that has more than one possible path between any two points.

*Half Duplex* The transmission of data in just one direction at a time. For example, a walkie-talkie is a half-duplex device because only one party can talk at a time.

*Half Step* This term means that the motor shaft will move a distance of 0.9 degree (400 steps per shaft revolution) instead of moving 1.8 degree per digital pulse.

*Hybrid Motion Technology™ (HMT)* A motor control technology representing a new paradigm in brushless motor control. By bridging the gap between stepper and servo performance, HMT offers system integrators a third choice in motion system design.

*Hybrid Motors* Hybrid stepper motors feature the best characteristics of PM and VR motors. Hybrid steppers are best suited for industrial applications because of high static and run torque, a standard low step angle of 1.8°, and the ability to Microstep. Hybrid stepper motors offer the ability to precisely position a load without using a closed-loop feedback device such as an encoder.

*Holding Torque* The maximum torque or force that can be externally applied to a stopped, energized motor without causing the rotor to rotate continuously. This is also called “static torque”.

*I/O* Inputs/outputs

*Inc* Increments

*Index pulse* Signal of an encoder to reference the rotor position in the motor. The encoder returns one index pulse per revolution.

*Inertia* A measure of an object’s resistance to a change in velocity. The larger an object’s inertia, the greater the torque required to accelerate or decelerate it. Inertia is a function of an object’s mass and shape. For the most efficient operation, the system-coupling ratio should be selected so that the reflected inertia of the load is equal to or no greater than 10 times the rotor inertia of the stepper motor.

*Inertia (Reflected)* Inertia as seen by the stepper motor when driving through a speed change, reducer or gear train.

*Lag* The amount (in full motor steps) that the rotor lags the stator. Lag conditions are caused by loading on the motor shaft, as during transient loading or rapid acceleration.

*Lead* The amount (in full motor steps) that the rotor leads the stator. Lead conditions are caused by an overhauling load, as during periods of rapid deceleration.

*Limit switch* Switch that signals overtravel of the permissible range of travel.

*Load* Any external resistance (static or dynamic) to motion that is applied to the motor.

*Locked rotor* When the lag/lead limit is reached, a timer starts a countdown that is determined by the user. The locked rotor will assert itself by triggering a flag and, depending on the selected mode, by disabling the output bridge.

*Loss of synchronization* In traditional stepper systems, when the lead/lag relationship of the rotor and stator reaches two full motor steps, the alignment of the magnetic fields is broken and the motor will stall in a freewheeling state. Hybrid Motion Technology eliminates this.

*Microstepping* A control electronic technique that proportions the current in a stepper motor's windings to provide additional intermediate positions between poles. Produces smooth rotation over a wide range and high positional resolution. Typically, step resolutions range from 400 to 51,200 steps per shaft revolution.

*Motor phase current* The available torque of a stepper motor is determined by the motor phase current. The higher the motor phase current the higher the torque.

*Multidrop* A communications configuration in which several devices share the same transmission line, although generally only one may transmit at a time. This configuration usually uses some kind of polling mechanism to address each connected device with a unique address code.

*NEMA* The acronym for the National Electrical Manufacturer's Association, an organization that sets standards for motors and other industrial electrical equipment.

*Node guarding* Monitoring of the connection with the slave at an interface for cyclic data traffic.

*Open Loop System* An open loop motion control system is where no external sensors are used to provide position or velocity feedback signals, such as encoder feedback of position.

*Opto-Isolated* A method of sending a signal from one piece of equipment to another without the usual requirement of common ground potentials. The signal is transmitted optically with a light source (usually a Light Emitting Diode) and a light sensor (usually a photo-sensitive transistor). These optical components provide electrical isolation.

*Parameter* Device data and values that can be set by the user.

*Persistent* Indicates whether the value of the parameter remains in the memory after the device is switched off.

*PLC* Programmable logic controller

*Position lead/lag* The HMT circuitry continually tracks the position lead or lag error, and may use it to correct position.

*Position make-up* When active, the position make-up can correct for position errors occurring due to transient loads. The lost steps may be interleaved with incoming steps, or reinserted into the profile at the end of a move.

*Power stage* The power stage controls the motor. The power stage generates currents for controlling the motor on the basis of the positioning signals from the controller.

*Pull-In Torque* This is the maximum torque the stepper motor can develop when instantaneously started at that speed.

*Pull-Out Torque* This is the maximum torque that the stepper can develop once an acceleration profile has been used to “ramp” it to the target speed.

*Quick Stop* Function used to enable fast deceleration of the motor via a command or in the event of a malfunction.

*Resolution* The smallest positioning increment that can be achieved.

*Resonance* The frequency that a stepper motor system may begin to oscillate. Primary resonance frequency occurs at about one revolution per second. This oscillation will cause a loss of effective torque and may result in loss of synchronism. The designer should consider reducing or shifting the resonance frequency by utilizing half step or micro-step techniques or work outside the primary resonance frequency.

*Rotor* The moving part of the motor, consisting of the shaft and the magnets. These magnets are similar to the field winding of a brush type DC motor

*Rotor Inertia* The rotational inertia of the rotor and shaft.

*RS485* Fieldbus interface as per EIA-485 which enables serial data transmission with multiple devices.

*Sinking Current* Refers to the current flowing into the output of the chip. This means that a device connected between the positive supply and the chip output will be switched on when the output is low.

*Slew* The position of a move profile where the motor is operating at a constant velocity

*Sourcing Current* Refers to the current flowing out of the output of the chip. This means that a device connected between the chip output and the negative supply will be switched on when the output is high.

*Stall detection* Stall detection monitors whether the index pulse is always correctly triggered at the same angle position of the motor shaft.

*Stator* The stationary part of the motor. Specifically, it is the iron core with the wire winding in it that is pressed into the shell of the frame. The winding pattern determines the voltage constant of the motor.

*Torque ramp* Deceleration of the motor with the maximum possible deceleration, which is only limited by the maximum permissible current. The higher the permissible braking current, the stronger the deceleration. Because energy is recovered up depending on the coupled load, the voltage may increase to excessively high values. In this case the maximum permissible current must be reduced.

*Variable current control* When active, variable current control will control the motor current as such to maintain the torque and speed on the load to what is required by the profile. This leads to reduced motor heating and greater system efficiency.

*Warning* If not used within the context of safety instructions, a warning alerts to a potential problem detected by a monitoring function. A warning is not a fault and does not cause a transition of the operating state. Warnings belong to error class 0.

*Watchdog* Unit that monitors cyclic basic functions in the product. Power stage and outputs are switched off in the event of faults.

*Zero crossing* The point in a stepper motor where one phase is at 100% current and the other is at 0% current.

# A Setting up an Lexium MDrive using RS Logix 5000

A

## A.1 Adding the Lexium MDrive

This appendix shows an example of adding an Lexium MDrive EtherNet/IP unit to an RS Logix 5000 project. The PLC used in this example was an Rockwell Automation Compact Logix L23E.

*Step 1*

1) Open a new project
2) Click the [+] next to I/O Configuration, Under your [PLC Name] right-click "Ethernet"
3) Select "New Module"

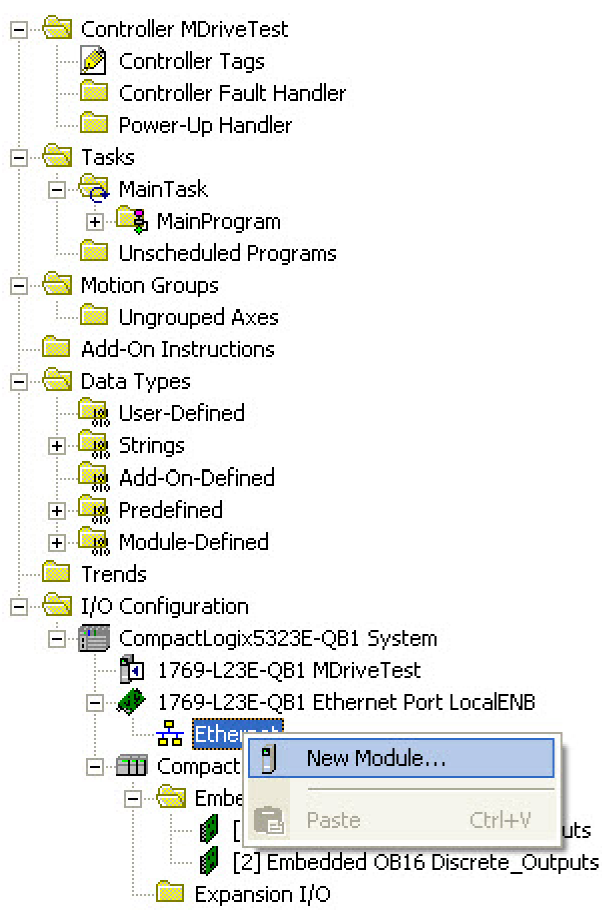


Figure A.1: Adding a new module

V1.00, 08.2014

4) Under “communications” of the Select Module” dialog Select ‘ETHERNET-MODULE Generic Ethernet Module’

5) Click OK

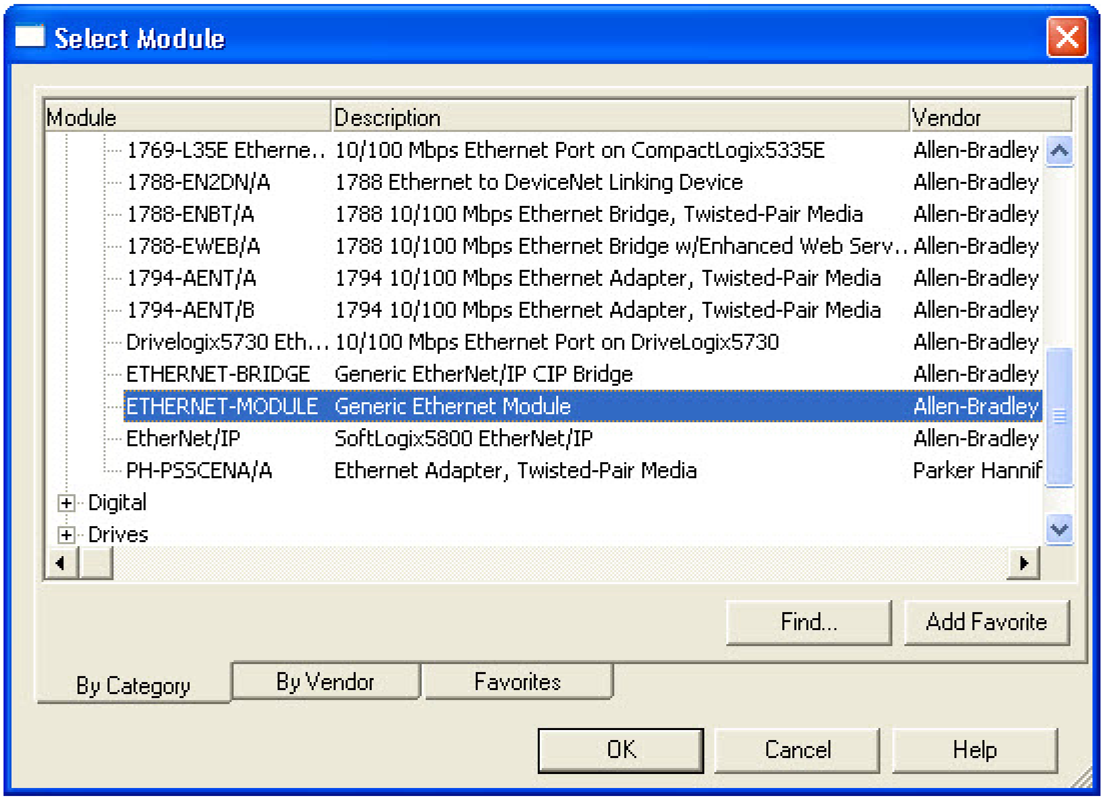


Figure A.2: Generic Ethernet Module

*Step 2* A “New Module” dialog will appear. Fill in the following information as follows:

**Name:** Lexium MDrive (IMPORTANT!)

**Description:** Lexium MDrive (Desc. is at user discretion)

**Comm Format:** Data - SINT

**IP Address:** 192.168.33.1

**Connection Parameters:**

| | Assembly Instance | Size |
|---|---|---|
| **Input** | 100 | 16 |
| **Output** | 112 | 36 |
| **Configuration** | 1 | 0 |

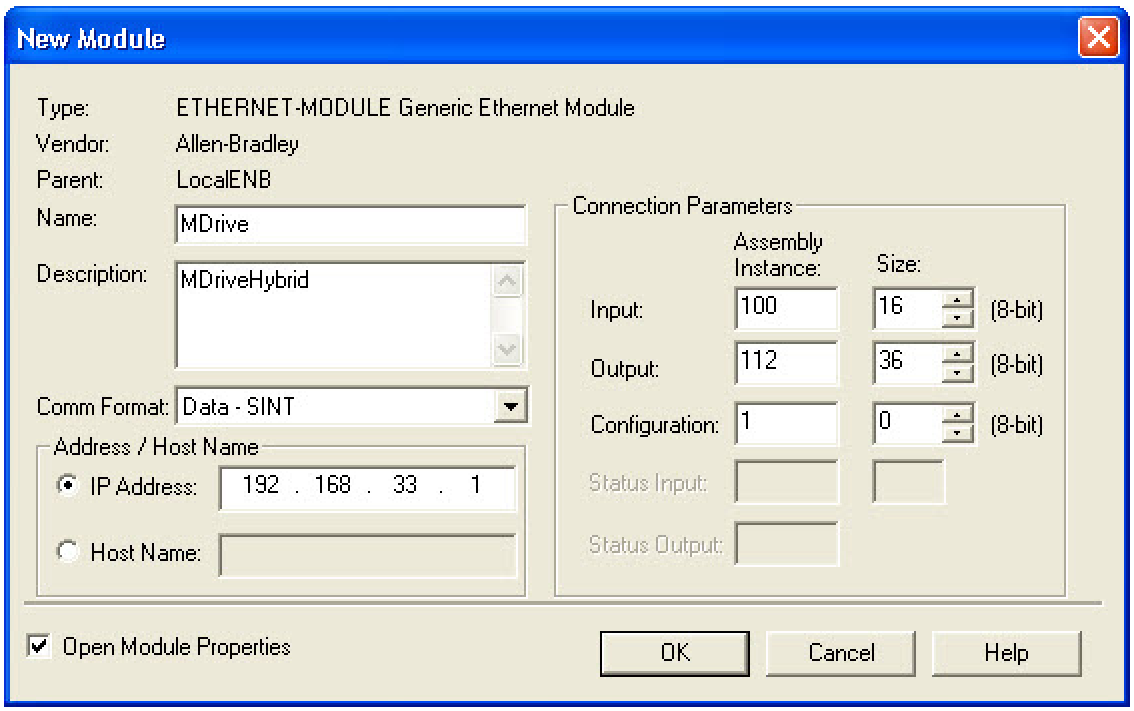


Figure A.3: New module setup

V1.00, 08.2014

*Step 3* Set “Request Packet Interval (RPI)” under the connection tab to 20ms.

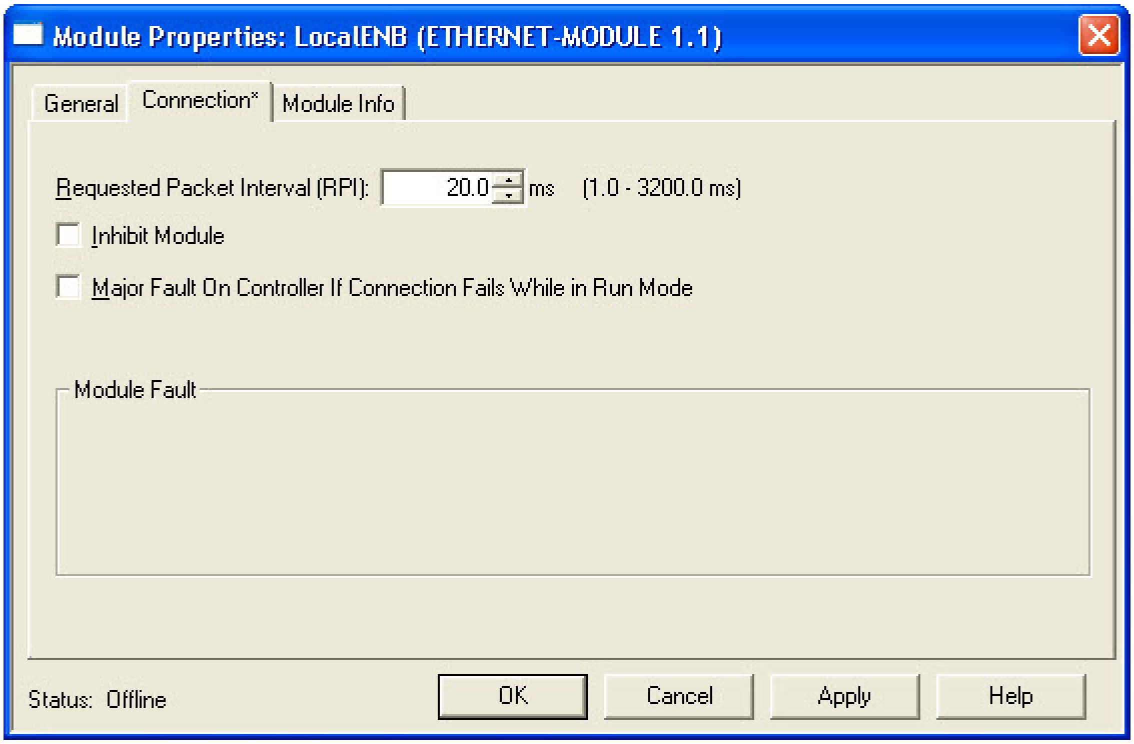


Figure A.4: Request Packet Interval setting

*Step 4* Under “Data Types”, right-click on “User-Defined and select “Import Data Type”.

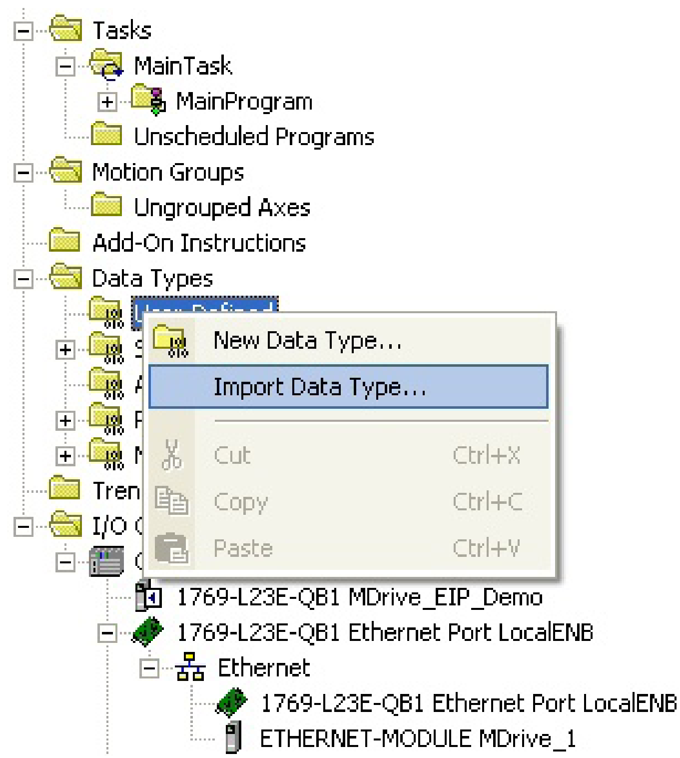


Figure A.5: Import routine

Select and import both Lexium MDrive_Inputs_T2O.L5X and Lexium MDrive_Outputs_O2T.L5X . These files are created using the TCP/IP Configuration Utility and exported as shown in Section 5.1.2 of this document.

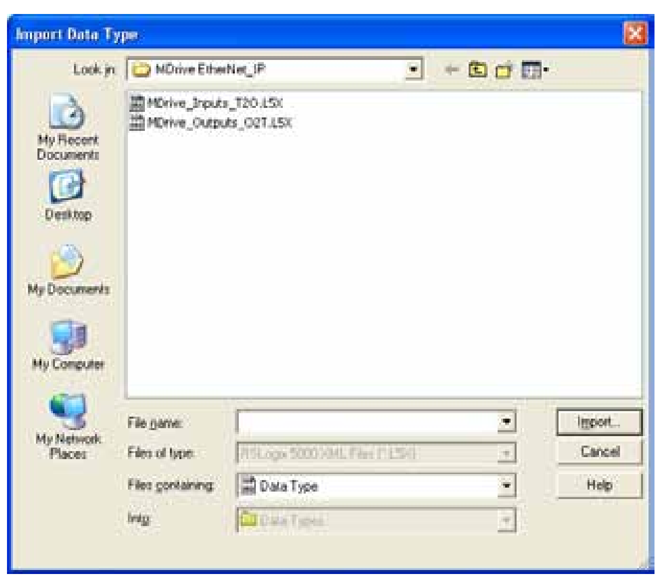


Figure A.6: I/O Data type files

In the Program Window, select the Synchronous Copy File Function (CPS). For the Source, The Lexium MDrive Input Data [0] tag.

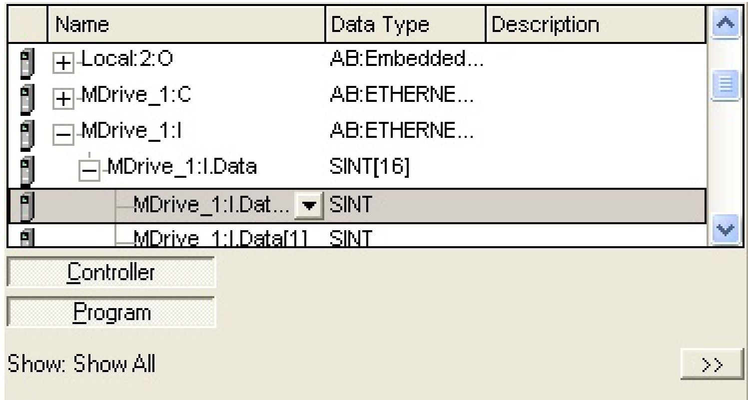


Figure A.7: Synchronous Copy File Function

For the Destination, create a Tag using the User Defined data type Lexium MDrive_Inputs_T2O as the Data Type.

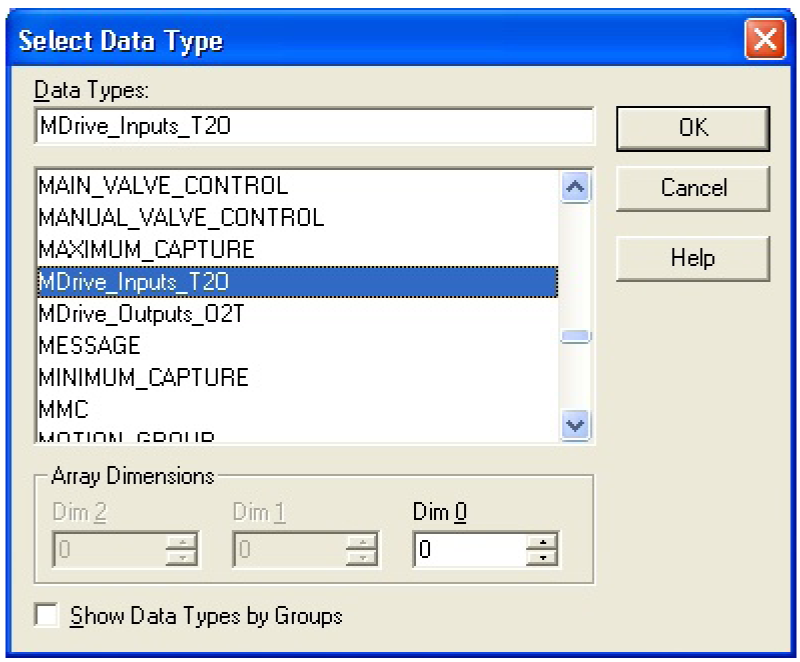


Figure A.8: Select data type

Set the Length to 16.

On the following rung, repeat the above process using a created Tag using the Lexium MDrive_Outputs_T2O data type as the source, The Lexium MDrive Output Data [0] as the destination and 36 as the length

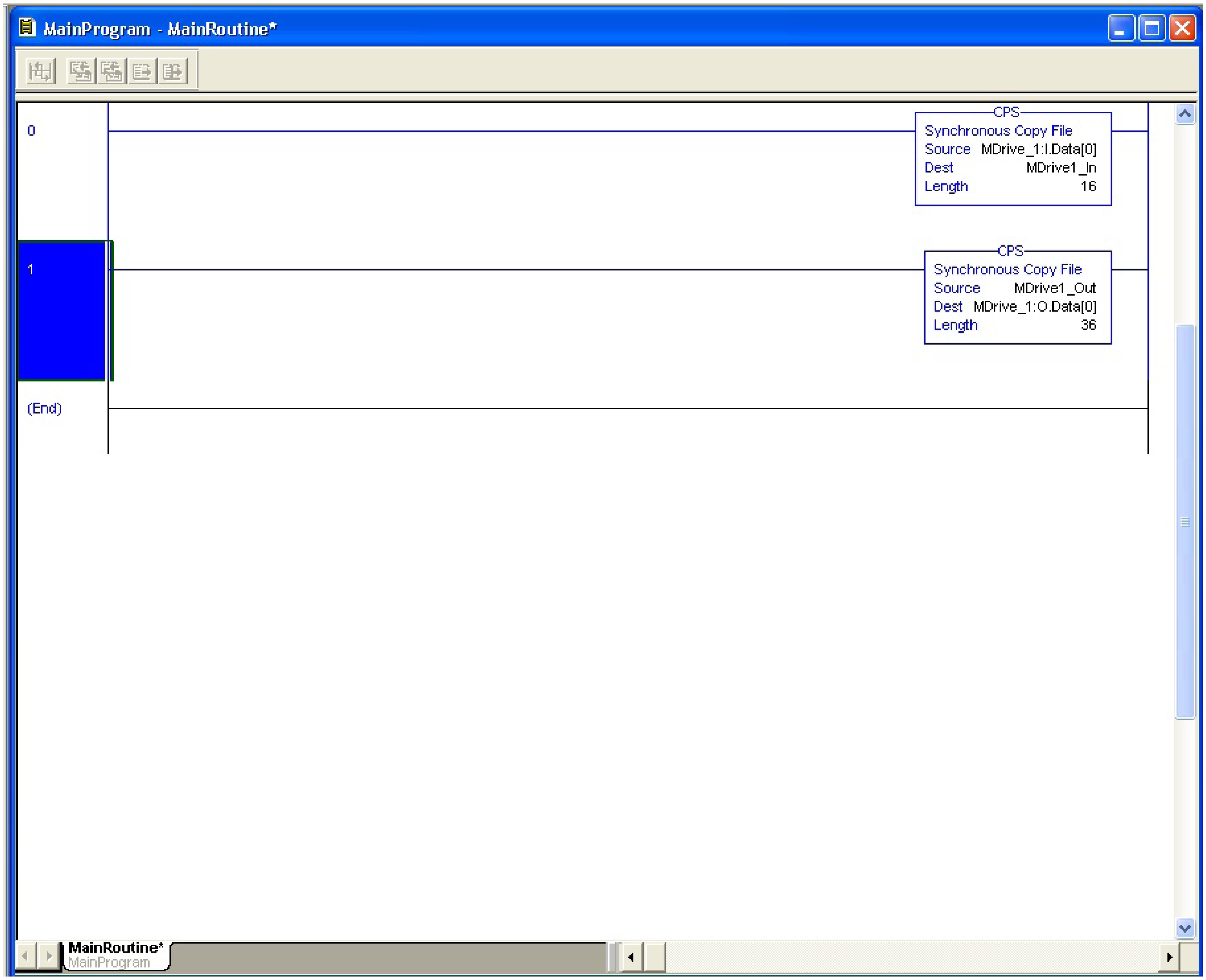


Figure A.9: Main program window

This associates the Tags created with the import of the user defined data types with the data in the implicit data object in the Lexium MDrive.

*Step 5* Add another rung to the routine created above. This rung should include a N/O contact as well and a Move (MOV) command. For the Contact, create a Tag called “Jog”. In the Move, select the Source to be a Jog speed in micosteps/sec. Set the Destination to be the “SL” parameter within the Lexium MDrive Out Tag created in Step 4. Copy this rung, changing the contact Tag to “Step”, and set the source to be 0.

Download the application and go online with the project. Toggling the “Jog” contact will cause the connected Lexium MDrive to Slew at the requested speed. Toggling the “Stop” contact will make it stop.

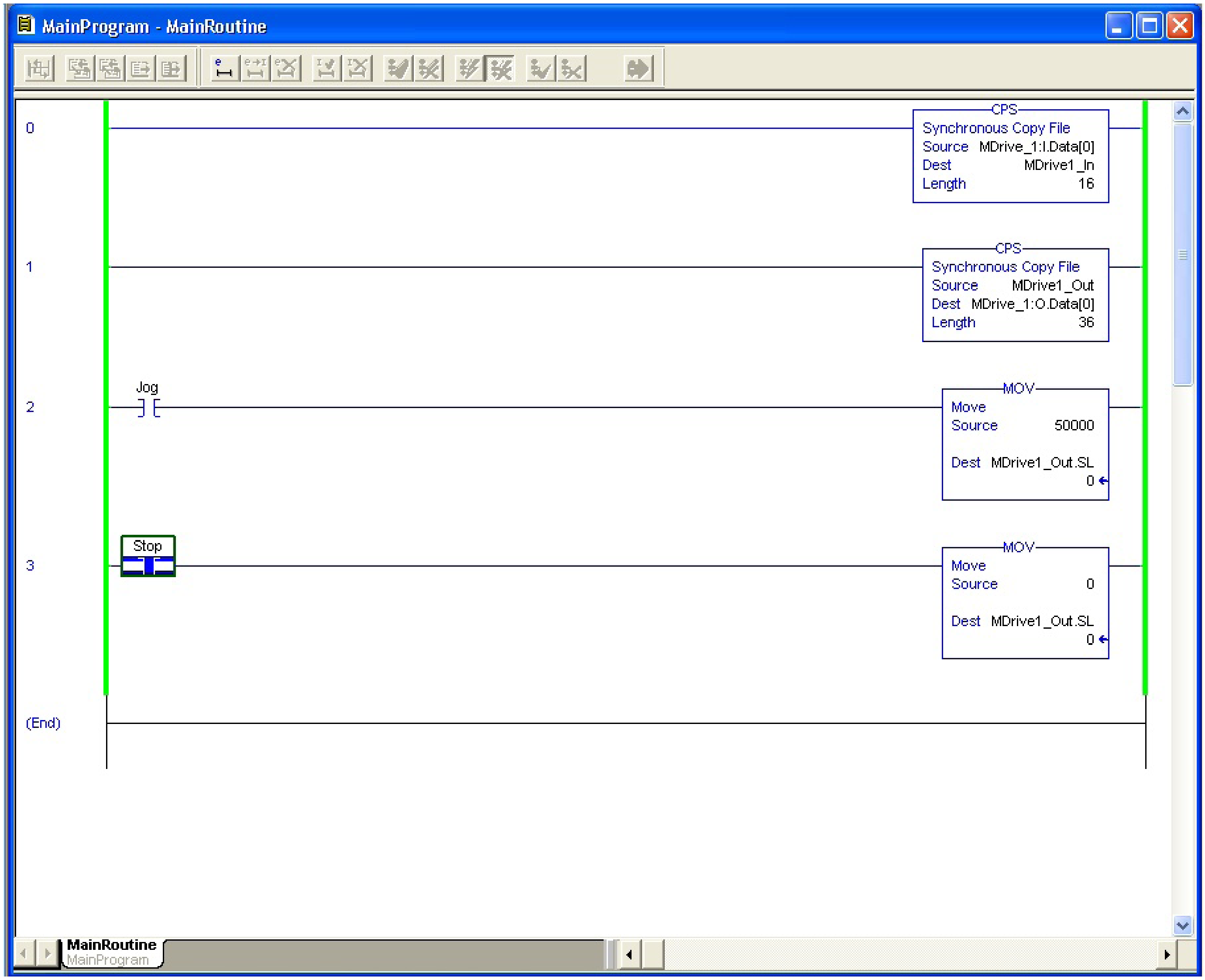


Figure A.10: Main program

V1.00, 08.2014

## A.2 Explicit messaging

Explicit messaging is used to transfer data that does not require continual updates. All Lexium MDrive parameters may be accessed via explicit messaging

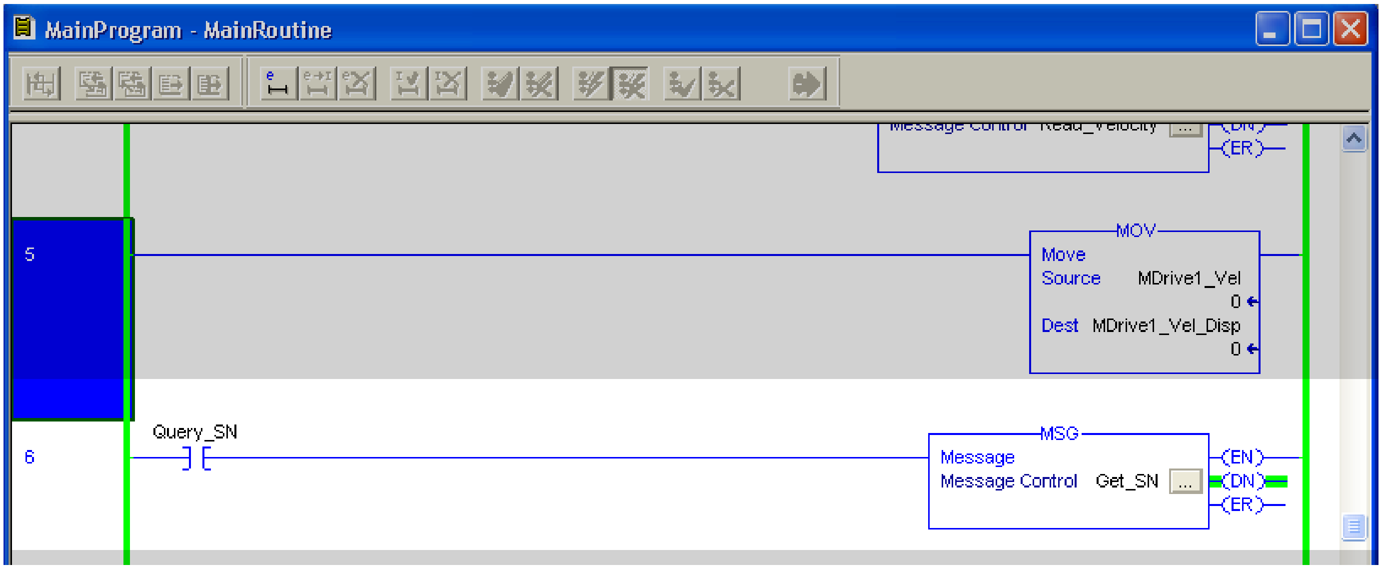


Figure A.11: Explicit message Query_SN

The message instruction must be configured to read or write to a specific Attribute (parameter) in the Lexium MDrive.

In this case, the Message is set to read the Serial Number from the Lexium MDrive and move it into a Tag labeled “Ser_Num”. This is done by chosing the “Get Attribute Single” from the Service Type pull-down and selecting the appropriate Class, Instance and Attribute. This information can be found in Section 8 of the Ethernet/IP Fieldbus Manual.

### A.2.1 Formatting the message

1) Add a message instruction (MSG), create a new tag for the message Get_SN (properties, base tag type, message data typt, controller scope) and click the configure button.

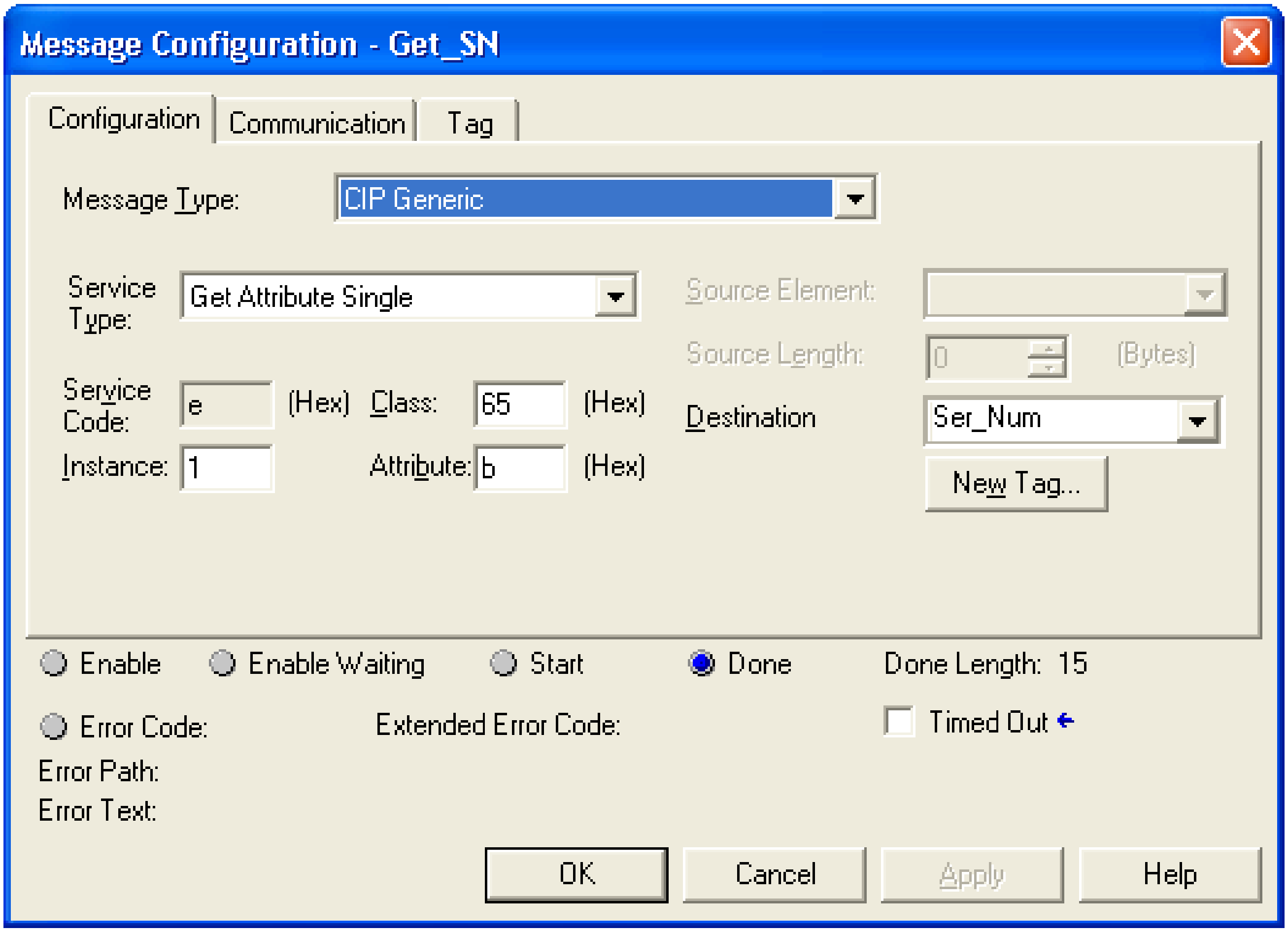


Figure A.12: Message configuration dialog

2) Set the message parameters as shown in the screen capture above in accordance with Table A.1,

| Field | Description |
|---|---|
| **Message Type** | The message type for Lexium MDrive parameters will be `CIP Generic` |
| **Service Type** | The service type for Lexium MDrive, in this case will be `Get Attribute Single`. If setting a parameter the service type would be `Set Attribute Single`. Available services depend on the class and instance being read or written. |
| **Service code** | This field will be read only when `Set Attribute Single` or `Get Attribute Single` is the service type. |
| **Class** | This is the EtherNet/IP class. For this exercise it is $64_h$ Miscellaneous. Refer to Section 8: Manufacturer specific objects, for a listing of supported classes, instances and attributes. |
| **Instance** | This is the EtherNet/IP instance or object. Refer to Section 8: Manufacturer specific objects, for a listing of supported classes, instances and attributes. |
| **Attribute** | The attribute represents, in this exercise, the hex number (0x0B), of the instance assigned to the `Read_Serial_Number` command. Refer to Section 8: Manufacturer specific objects, for a listing of supported classes, instances and attributes. |

Table A.1: Message configuration

Page intentionally left blank

# WARRANTY

Reference the web site at www.motion.schneider-electric.com for the latest warranty and product information.

**USA SALES OFFICES**
**East Region**
Tel. 610-573-9655
e-mail: e.region@imshome.com
**Northeast Region**
Tel. 860-368-9703
e-mail: n.region@imshome.com
**Central Region**
Tel. 630-267-3302
e-mail: c.region@imshome.com
**Western Region**
Tel. 602-578-7201
e-mail: w.region@imshome.com

**EUROPEAN SALES MANAGEMENT**
Tel. +33/4 7256 5113 – Fax +33/4 7838 1537
e-mail: europe.sales@imshome.com

**TECHNICAL SUPPORT**
Tel. +00 (1) 860-295-6102 – Fax +00 (1) 860-295-6107
e-mail: etech@imshome.com

**Schneider Electric Motion USA**
370 N. Main Street
Marlborough, CT 06447 USA

www.motion.schneider-electric.com

Owing to changes in standards and equipment, the characteristics given in the text and images in this document are not binding until they have been confirmed with us.
Print: Schneider Electric Motion USA
Photos: Schneider Electric Motion USA

Date : 08 / 2014