Programming and Software Reference for:

- **Lexium MCode**
- **Lexium MCode/TCP**

Lexium Lexium MDrive
Motion Control
Products
V1.00, 12.2014

Schneider Electric

MCode Programming and Software Reference		
Date	**Revision**	**Changes**
05/02/2013	V1.00, 02.2013	Initial Release
01/10/2013	1.00, 10.2013	Lexium MDrive Motion Control and Ethernet support release.
04/07/2014	1.00, 01.2014	Support for firmware release 5.007 - Added status marker for the read voltage level (VT) command.
04/24/2014	1.00, 04.2014	Support for firmware release 5.009
08/15/2014	1.00, 08.2014	Support for firmware release 5.010, minor corrections and additions throughout.
12/15/2014	1.00, 12.2015	Support for firmware release 5.013, minor corrections and additions throughout.

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

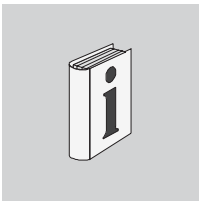
Table of Contents

List of Figures

List of Tables

Page intentionally left blank

Writing conventions and symbols



Feature definitions

Icons are used to differentiate between versions of the Lexium MDrive Motion Control models with or without an encoder.

C **O** Indicates a feature on both open and closed loop models

C Used to indicate a feature found only on closed loop models with an internal encoder

Work steps

If work steps must be performed consecutively, this sequence of steps is represented as follows:

- ■ Special prerequisites for the following work steps
- ▶ Step 1
- ◁ Specific response to this work step
- ▶ Step 2

If a response to a work step is indicated, this allows you to verify that the work step has been performed correctly.

Unless otherwise stated, the individual steps must be performed in the specified sequence.

Bulleted lists

The items in bulleted lists are sorted alphanumerical or by priority. Bulleted lists are structured as follows:

- Item 1 of bulleted list
- Item 2 of bulleted list
  - Subitem for 2
  - Subitem for 2
- Item 3 of bulleted list

Making work easier

Information on making work easier is highlighted by this symbol:

Sections highlighted this way provide supplementary information on making work easier.

Draft V0.900.08.2011

Parameters

Parameters are shown as follows

```
RC      Motor Run Current
```

Units of measure

Measurements are given US units, metric values are given in SI units in parenthesis.

Examples:

1.00 in (25.4 mm)
100 oz-in (70 N-cm)

1 Introduction

1.1 About this manual

This manual is for devices utilizing the Lexium MCode programming language.

1.2 Lexium MCode versions

1.2.1 Lexium MCode

Lexium MCode is the programming and control language for products communicating over serial RS-422/485.

1.2.2 Lexium MCode/TCP

Lexium MCode/TCP is the Lexium MCode language adapted to communicate over Ethernet TCP/IP networks. The function and usage is identical with the exception that the commands related to RS-422/485 communication are disabled. Lexium MCode/TCP will connect to TCP or UDP port 503.

1.3 Lexium MCode device overview

The following product lines use Lexium MCode for configuration and programming:

- Lexium MDrive Motion Control (closed loop)
- Lexium MDrive Motion Control (open loop)
- Lexium MDrive Ethernet (closed loop)
- Lexium MDrive Ethernet (open loop)

Many of the Lexium MCode commands are device-specific based on encoder availability and communications interface. In the command detail (Section 5), each command is listed with device compatibility. Ensure that the command will function with your device before using.

1.4 Documentation reference

The following user's manuals are available for the Lexium MCode devices:

- Product hardware manual, describes the technical data and installation of the product.
- Product software manual, describes the configuration and programming of the product.

This documentation is also available for download from our web site at: http://motion.schneider-electric.com

V1.00, 12.2014

1.4.1 Related documents

MODBUS/TCP The MODBUS/TCP Fieldbus manual describes the implementation of the MODBUS protocol for Ethernet equipped MDrive products. When connected to TCP port 502, the MDrive Ethernet will only respond to MODBUS commands.

EtherNet/IP The EtherNet/IP Fieldbus manual describes the implementation of the EtherNet/IP protocol for Ethernet equipped MDrive products.

Lexium MDrive Software Suite The Lexium MDrive Software Suite Manual documents the installation and use of the programming tool for the Lexium MDrive Motion Control and Ethernet products.

1.5 Product software

The Lexium MDrive Software Suite is the program used to commission, program and operate the Lexium MDrive Motion Control. The Lexium MDrive Software Suite may be downloaded from the web site at:

http://motion.schneider-electric.com/

Instructions for installation and use of this software are to be found in the Lexium MDrive Software Suite product manual which may be downloaded at:

http://motion.schneider-electric.com/

2 Safety

2.1 Qualification of personnel

Only technicians who are familiar with and understand the contents of this manual and the other relevant documentation are authorized to work on and with this drive system. The technicians must be able to detect potential dangers that may be caused by setting parameters, changing parameter values and generally by the operation of mechanical, electrical and electronic equipment.

The technicians must have sufficient technical training, knowledge and experience to recognise and avoid dangers.

The technicians must be familiar with the relevant standards, regulations and safety regulations that must be observed when working on the drive system.

2.2 Intended Use

The drive systems described here are products for general use that conform to the state of the art in technology and are designed to prevent any dangers. However, drives and drive controllers that are not specifically designed for safety functions are not approved for applications where the functioning of the drive could endanger persons. The possibility of unexpected or unbraked movements can never be totally excluded without additional safety equipment.

For this reason personnel must never be in the danger zone of the drives unless additional suitable safety equipment prevents any personal danger. This applies to operation of the machine during production and also to all service and maintenance work on drives and the machine. The machine design must ensure personal safety. Suitable measures for prevention of property damage are also required.

In all cases the applicable safety regulations and the specified operating conditions, such as environmental conditions and specified technical data, must be observed.

The drive system must not be commissioned and operated until completion of installation in accordance with the EMC regulations and the specifications in this manual. To prevent personal injury and damage to property damaged drive systems must not be installed or operated.

Changes and modifications of the drive systems are not permitted and if made no warranty and liability will be accepted.

The drive system must be operated only with the specified wiring and approved accessories. In general, use only original accessories and spare parts.

The drive systems must not be operated in an environment subject to explosion hazard (ex area).

2.3 Hazard Categories

Safety notes and general information are indicated by hazard messages in the manual. In addition there are symbols and instructions affixed to the product that warn of possible hazards and help to operate the product safely.

Depending on the seriousness of the hazard, the messages are divided into three 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 General safety instructions

⚠ 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.

⚠ DANGER

EXPOSED SIGNALS

Hazardous voltage levels may be present if using an open frame power supply to power the product.

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".

Page intentionally left blank

3 Introduction to Lexium MCode programming

This section will acquaint the user with basics of Lexium MCode programming and the simple 1 and 2 character mnemonics which make up the Lexium MCode programming language.

- Operational modes.
- Basic components of the Lexium MCode programming language.

3.1 Operational modes

There are two operational modes for the Lexium MCode compatible products: Immediate and Program.

1) **Immediate**: Commands are issued and executed directly to the controller by user input into the terminal window.
2) **Program:** Program Mode is used to input user programs into the motion controller.

3.2 Basic components of Lexium MCode

There are five basic components of the Lexium MCode Programming Language, they are:

1) Instructions
2) Variables
3) Flags
4) Keywords
5) Math functions

3.2.1 Instructions

An instruction results in an action. There are four types of instructions:

Motion Motion instructions are those that result in the movement of a motor. The syntax for these commands is as follows: type the command followed by a space, and then the velocity or position data. For example: MA 2000 will move the motor to an absolute position of 2000.

I/O An I/O instruction results in the change of parameters or the state of an input or output. The syntax for these commands are as follows: type the command then a space or an equal sign, then the data. Example: O2=0 or O2 0 will set output 2 to 0.

Program A program instruction allows program manipulation. The syntax of these vary due to the nature of the command. Some command examples would be: PG 100, which toggles the system into program mode starting at address 100; BR LP, I1=1, which will branch to a program labeled LP if input 1 is true.

System A system instruction is an instruction that can only be used in immediate mode to perform a system operation such as program execution (EX) or listing the contents of program memory (L). For example: EX 100 will execute a program located at address 100 of program memory space, or EX K1 will execute a program labeled K1.

3.2.2 Variables

A Variable is identified by a mnemonic and allows the user to define or manipulate data. These can also be used with the math functions to manipulate data. There are two classes of variables: factory-defined and user-defined. There are 192 user program labels and variables available. The syntax for each variable may differ.

Factory defined variables These variables are predefined at the factory. They cannot be deleted. When an FD (Factory Default) instruction is given, these variables will be reset to their factory default values. There are two types of factory defined variables:

- Read/Writable: These factory defined variables can have their value altered by the user to affect events inside or outside of a program. For example A (Acceleration variable) can be used to set the Acceleration, or P (Position variable) can be used to set the position counter.
- Read Only: These factory defined variables cannot be changed by the user. They contain data that can be viewed or used to affect events inside a program. For example, V (Velocity variable) registers the current velocity of the motor in steps per second.

User defined variables The VA instruction allows the user to create a 2 or 3 character name to a user defined variable (32 bit value).

The restrictions for this command are:

1) A variable cannot be named after an Lexium MCode Instruction, Variable or Flag.
2) The first character must be alpha, the second character may be alphanumeric.
3) A variable is limited to two characters.

With these the user can define a variable to store and retrieve data and perform math functions. When the FD (Factory Defaults) instruction is given, these variables will be deleted! There are two types of user defined variables:

- **Global variables:** global variables are variables that are defined outside of a program. The benefit to using a global variable is that no user program memory is required. For example, the user can define a variable called SP for speed by entering VA SP into the terminal. The user can then set that variable equal to the value of a read only variable V (velocity) by entering SP = V into the terminal.
- **Local variables:** this type of user defined variable is defined within a program and can only affect events within that program. It is stored in RAM. Note a local variable is not static, but is erased and declared again each time a program is executed.

3.2.3 Flags

Flags show the status of an event or condition. A flag will only have one of two possible states: either 1 or 0. Unlike variables, there are only factory defined flags.

Factory defined flags

Factory defined flags are predefined at the factory and cannot be deleted. When a FD (Factory Defaults) instruction is given, these flags will be returned to their factory default state. There are two types of factory defined flags:

- **Read/Writable:** This type of flag is user alterable. They are typically used to set a condition or mode of operation for device. For example EE = 1 would enable encoder operation, or EE = 0 would disable the encoder functions.
- **Read Only:** Read Only flags cannot be changed by the user. They only give the status of an event or condition. Typically this type of flag would be used in a program in conjunction with the BR (Branch Instruction) to generate an if/then event based upon a condition. For example the following line of code in a program BR SP, MV = 0 would cause a program to branch to a subroutine named “SP” when the MV, the read only moving flag, is false.

3.2.4 Keywords

Keywords are used in conjunction with the PR and IP instructions to indicate or control variables and flags. For instance, PR UV would print the state of all the user-defined variables to the screen. IP would restore all the factory variables from the NVM.

3.2.5 Math functions

Math functions are used to perform various arithmetic functions on numeric data stored in registers or variables. Supported functions are +, -, x, ÷, >, <, =, <=, >=, <>, AND, OR, XOR, NOT.

Example

Addition .....K2†=P+R2

Subtraction.....K3†=R1-P

Multiplication .....A=A*2

Division ..... A=A/2

†User-defined variable used as an example.

3.3 Program structuring

Proper structuring of your Lexium MCode program will ensure your ability to work efficiently and will aid in trouble shooting your program. The figure below illustrates how your program can be blocked out to group the global system declarations, the main program body and the subroutines.

```
'[VARIABLES]
Is = 1,0,0
Ms=256
A=1000000
D=A
Hc=2
Rc=75
P=0
```

Block 1: System configuration
Contains the global variables and flags that define the operating parameters of the device during the execution of the program.

```
'[PROGRAMS]
PG 1
  LB X1
   P=0
  LB X2
   CL Z1, I1-1
   H 10
   BR X1
```

Block 2: Main program
The first line places the device in program mode starting at address 1. This will contain the main program label, as well as trigger events which will call subroutines to perform some action.

```
'Subroutine from
"trigger event
 LB Z1
   MA 51200
   H
   MA 0
   H
   RT
```

Block 3: Subroutines
Subroutines are called from the main program block based upon a condition such as an input state, a trip, or a numeric value stored in a register or variable. They may also be called unconditionally.

```
E
PG
```

Block 4: End program
E - designates the end of the program
PG - exits program mode

Figure 3.1 Recommended Lexium MCode program structure

3.3.1 Programming aids

Lexium MDrive Programmer

One of the most powerful tools available to you is the Lexium MDrive Programmer module of the Lexium MDrive Software Suite. The Lexium MDrive Programmer is visual IDE (Integrated Development Environment) for developing, debugging, simulating and deploying Lexium MDrive programs written in Lexium MCode.

It features a program text editor, terminal emulation, program simulation and graphing. Program development may be accomplished by direct entry or by selecting an action and filling out a dialog.

User Labels

The Lexium MCode programming language allows for 192 user labels for your programs, subroutines, and user variables and flags. A label consists of 2 characters, the first of which must be a letter, the second may be alphanumeric. A label cannot use the same character combination as any of the mnemonics used in the Lexium MCode programming language.

For purpose of this manual we have used the following example labels:

Program label (G).....Example: G1, G8, Ga

Subroutine label (K) ..... Example: K7, K2, Ks

User variable label (Q) ..... Example: Q3, Q9, Qz

Example labeling

```
VA Q1       'Create user variable Q1

PG 100      'Enter Program mode
 LB G1      'Label Program G1
 CL K1, I2=1       'Call Subroutine K1 if Input 2 is HIGH
 BR G1      'Unconditional Branch to G1

K1          'Declare Subroutine K1
```

Comments

Lexium MCode allows for comments to be inserted in your program code. The comment character for the Lexium MCode language is the Apostrophe (‘). The device will ignore the text string following the apostrophe. Please note that the maximum length of a single line of program code is 64 characters, this includes program text, spaces and comments.

Using comments will be of assistance in trouble shooting your program.

Programming reference

Another powerful tool is this manual. Section 3 contains detailed explanations and usage examples of each mnemonic in the Lexium MCode Programming Language. In Section 8 there are a number of fully commented example programs that can be used to learn the basics of programming and using the various functions of your Lexium MCode compatible device.

3.4 Commonly used variables and instructions

3.4.1 Variables

MS (Microstep resolution)

MS (Microsteps Select) defines the resolution of the stepping motor.

Motor rotation:	1.8° per step (200 steps/rev.)
Microsteps/step:	MS
Microsteps/rev:	MS * 200
MS default:	256 microsteps/step or 256 * 200 = 51200 microsteps/rev
To read:	PR MS
To write:	MS=<integer>
Notes:	MS values are predefined to 20 resolutions. See command details

As we continue you will see that all motion variables use this value.

P (Position)

P indicates the position in either steps or encoder counts depending upon the enable/disable state of encoder functions.

Open loop:	Position from Counter 1 (C1) in motor steps
Encoder enabled:	Position from Counter 2 (C2) in encoder counts
To read:	PR P
To write:	P=0 will clear the position
Notes:	MS values are predefined to 20 resolutions. See command details

VI (Initial velocity)

Initial velocity in steps per second.

Default:	1000 steps/sec
To read:	PR VI
To write:	VI=<integer>
Notes:	VI will return an error is set to a value greater than VM. The size of the step is a function of MS

VM (Maximum velocity)

Maximum or final velocity in steps per second. (Step size is a function of the value of ms).

Default:	768000 steps/sec
To read:	PR VM
To write:	VM=<integer>
Notes:	VM will return an error if set to a value less than VI. The size of the step is a function of MS

A (Acceleration)

Acceleration in steps per second^2.

Default:	1000000 steps/sec^2
To read:	PR A
To write:	A=<integer>
Notes:	The velocity of the motor increases by <A> every second until VM, or the velocity commanded by a slew (SL)

D (Deceleration)

Deceleration in steps per second^2.

Default:	1000000 steps/sec^2
To read:	PR D
To write:	D=<integer>
Notes:	The velocity of the motor decreases by <D> every second until VI, or the velocity commanded by a slew (SL)

3.4.2 Motion instructions

Motion instructions are those that cause the motor to move or affect the movement of the motor. There are a few factors that must be considered when programming motion commands. Linear distances, number of revolutions, degrees of rotation and timed moves can be calculated and programmed from these factors.

- All motion is programmed either microsteps per second or (when the encoder is enabled) encoder counts (pulses) per second.
- All motion is directly affected by the motion command and the program variables.
- There are a number of factors impacting motion instructions. These are addressed in detail in Section 7: Application and programming notes.

MA (Move absolute)

Move to an absolute position relative to a defined zero position.

For example, type the following commands followed by pressing enter:

```
P=0        'set the current position to 0 (zero)
MA 20000   'move 20000 steps from 0 in the plus direction
PR P       'the terminal screen will read 20000
MA 3000    'move 3000 steps from 0 in the plus direction
PR P       'the terminal screen will read 3000
```

Absolute moves are always relative to 0 (zero).

You may program moves in the minus direction by typing the minus sign (-) before the value.

MR (Move relative)

Move the number of steps programmed relative to current position.

For example, type the following commands followed by pressing enter:

```
P=0        'set the current position to 0 (zero)
MR 20000   'move 20000 steps from the current position in
           'the plus direction
PR P       'the terminal screen will read 20000
MR 3000    'move 3000 steps from the current position in
           'the plus direction
PR P       'notice the position read is 23000 and not 3000
```

Relative moves are cumulative and are either added to or subtracted from the current position.

You may program moves in the minus direction by typing the minus sign (-) before the value.

SL (Slew axis)

Move at a constant velocity.

```
SL 200000  'the motor moves at a constant velocity 200000
           'steps per second
```

- The slew command overrides the VM (maximum velocity) parameter.
- The value of the slew command may be changed “on the fly”.
- You may program moves in the minus direction by typing the minus sign (-) before the value.

H (Hold)

An H (hold command) should typically follow any MA or MR commands in a program so that program execution is suspended until the motion is complete.

Below is a usage example.

```
PG 100     'enter program mode at address 100
LB M1              'label program M1
MR 20000   'set mode to relative, move relative 20000
steps
H          'hold until motion completes
MR -20000  'move relative -20000 steps
H          'hold until motion completes
E          'end program
PG         'exit program mode
```

A delay time value (1 to 65000 milliseconds) may be programed with the hold command.

(Note: There are circumstances where you may not want to hold up program execution.)

3.4.3 I/O instructions

Is (Set input function)

This command configures the Line, Type and Active state of inputs 1-4.

Type	Function	Description
0	General purpose	Typical usage: to trigger events within a program
1	Home	When active, will trigger the set homing routine as set by the homing variable (HM)
2	Limit plus (+)	will function as specified by the limit variable (LM). Will trigger and error 83 when a + limit is reached.
3	Limit minus (—)	will function as specified by the limit variable (LM). Will trigger and error 84 when a — limit is reached.
4	G0	G0 input, will run program at address 1 upon activation.
5	Soft stop	Soft stop, stops motion with deceleration and halts program execution. If program is paused (PS), input is ignored.
6	Pause	Pause, pause/resume program with motion.
7	Jog plus (+)	When active, will jog the motor in the positive direction at max. velocity (VM). The jog enable (JE) flag must be set for this to function.
8	Jog minus (—)	When active, will jog the motor in the minus direction at max. velocity (VM). The jog enable (JE) flag must be set for this to function.
11	Reset	When set as RESET input, then the action is equivalent to a ^C entered into a terminal. Note: If the input is in a sourcing configuration, active when high, ground the input first or a reset will occur.
12	Capture (INPUT 1 Only)	When set as a capture input is a momentary high speed input that operates with the Trip Capture (TC) variable to run a subroutine upon the trip. It feature variable input filtering ranging from 50 nS to 12.9 µS.

The syntax for setting up an input is

```
Is = <input #>, <type>, <active>
```

Set input 1 as general purpose active low	`Is =1,0,0`
Set input 2 as jog+ active high	`Is =2,7,1`
Set inputs 3 and 4 as Limit +/Limit —, active low	`Is =3,2,0` `Is =4,3,0`
Set input 1 as a capture input active high	`Is =1,12,1`

- Only input 1 may be set to the Capture function
- Inputs may be set globally or locally (inside a program)
- The syntax to read the settings of the inputs is PR Is

I<1-4> (Read input state)

Used to read the state of an individual input.

PR I1 will read the state of input 1 and display it to the terminal window.

BR K5, I2=0 will branch to the program address tabled K5 when Input 2 is LOW

IN (Read all inputs as decimal)

Used to read the decimal equivalent of the 4 bit binary nibble represented by all inputs collectively. Note the Input 4 is the Most Significant Bit.

PR IN will print the decimal value of the inputs.

IN will only read inputs 1-4 on devices equipped with only the standard I/O Set!

0s (Set output function)

Used to set the function of an output.

Type	Function	Description
16	General purpose	Sets the output as a general purpose user output.
17	Moving	The output will be active when the axis is in motion
18	Error	Indicates a software error condition occurred.
19	Stall	**C** Indicates a stalled condition exists. Stall detect mode (SM) must be enabled and HMT must be off (AS=0), and encoder functions must be enabled (EE=1)
20	Velocity changing	Will be active when the axis is accelerating or decelerating
21	Locked rotor	**C** Indicates a locked rotor condition exists.
23	Moving to position	Will indicate when the axis is moving to a specified position.
24	HMT active	**C** Will be active when HMT is on.
25	Make-up active	**C** Will indicate when steps are being made up/
26	Encoder A	**C** Sets as the encoder channel A output
27	Encoder B	**C** Sets as the encoder channel B output
28	Trip	Indicates a trip condition (Output 3 ONLY, active low only)
29	Attention	Can be set to trigger on an attention event as defined by the Ao variable.

The syntax for setting up an output is

```
Os = <output #>, <type>, <active>
```

Set output 1 as general purpose active low	`Os =1,16,0`
Set output 2 as moving active high	`Os =2,17,1`
Set input 3 as a trip output active high	`Os =3,28,1`

- Only output 3 may be set to the trip function
- Outputs may be set globally or locally (inside a program)
- The syntax to read the settings of the inputs is PR Os

O <1 - 3> (Set output)

Used to set the state of an output point.

O2=1 will set Output 2 TRUE

OT (Set all outputs as BCD)

Used to set the 3 bit binary equivalent of the decimal number represented by all 3 outputs collectively. Note the output 3 is the most significant bit.

OT=5 will set the outputs to 101

3.4.4 System instructions

The following system instructions will be used frequently.

CP (Clear program memory)

The CP instruction is used to clear program memory space. CP must be followed by a save commend S.

FD (Restore facory defaults)

The FD instruction is used to return the device to its factory default state.

<esc> (Stop motion and program)

<esc> The ESCAPE key will stop the user program and stop the motor with no decel rate.

<CTRL+C > (Software reser)

CTRL+C will reboot the unit. This includes reloading of the programs stored in nonvolatile memory into RAM and executing any programs residing at label SU (Start Up).

3.4.5 Program instructions

PG (Begin program mode)

This instruction toggles the device into or out of program mode.

```
PG 200      'Switch to program mode at address 200
 xxxxx      'Program starting at address 200
 xxxxx      '     |
 xxxxx      '     |
PG          'Switch out of program mode
```

LB (Label program, subroutine or branch)

Lexium MCode also offers the user the convenience of naming programs, subroutines and processes to ease in branching from one part of a program to another, or calling a subroutine.

These labels, once set, will act as pointers to locations in program memory space.

The LB, or label instruction, allows the user to assign a 2 character name to a program or branch process within a program or subroutine.

The restrictions for this command are:

1) A label cannot be named after an instruction, variable or flag.
2) The first character must be alpha, the second character may be alpha-numeric.
3) A label is limited to two characters.
4) A program labeled SU will run on power-up

Please Note: Any program labeled “SU” will execute on power-up.

```
PG 200     'Switch to program mode at address 200
LB k1              'Label command will name the program K1
 xxxxx     'Program named by LB command  xxxxx
 xxxxx     '
PG         'Switch out of program mode
```

BR (Branch) Used to branch conditionally or unconditionally to a routine.

```
PG 200     'Switch to program mode at address 200
LB K1              'Label command will name the program
 xxxxx
 xxxxx     'Program named by LB command
 xxxxx
 BR K1     'Unconditional branch to Program Label K1
PG         'Switch out of program mode
```

CL (Call subroutine) Used to call a subroutine conditionally or unconditionally to a routine.

```
PG 200     'Switch to program mode at address 200
LB K1              'Label command will name the program
 xxxxx
 xxxxx     'Program named by LB command
 xxxxx
 CL X1     'Unconditional call to subroutine label X1
E          'End program
PG         'Switch out of program mode

'[SUBROUTINES]
LB X1      'Label subroutine X1
 xxxxx     'Subroutine named by LB command
RT         'Return from subroutine
```

E (End program) Designates the end of a program.

```
PG 200     'Switch to program mode at address 200
LB K1              'Label command will name the program
 xxxxx
 xxxxx     'Program named by LB command
 xxxxx
 BR K1     'Unconditional branch to Program Label K1
E          'End Program
PG         'Switch out of program mode
```

H (Hold program execution) Delays program execution in milliseconds.

```
PG 200     'Switch to program mode at address 200
LB K1              'Label command will name the program
 xxxxx
 xxxxx     'Program named by LB command
 xxxxx
 H 2000    'Hold 2 seconds before execution of program
 BR K1     'Unconditional branch to Program Label K1
E          'End Program
PG         'Switch out of program mode
```

PR (Print) Outputs specified text and parameter values to a terminal or terminal software on a host PC.

```
PG 200            'Switch to program mode at address 200
LB K1             'Label command will name the program
 xxxxx
 xxxxx            'Program named by LB command
 xxxxx
 H 2000           'Hold 2 seconds before execution.
 PR "Position =", P       'Print position
 BR K1            'Uncond branch to Program Label K1
E                 'End Program
PG                'Switch out of program mode
```

RT (Return from subroutine) Required to return from a subroutine to the program.

```
PG 200     'Switch to program mode at address 200
LB K1              'Label command will name the program
 xxxxx
 xxxxx     'Program named by LB command
 xxxxx
 CL X1     'Unconditional call to subroutine label X1
E          'End program
PG         'Switch out of program mode

'[SUBROUTINES]
LB X1      'Label subroutine X1
 xxxxx     'Subroutine named by LB command
RT         'Return from subroutine
```

VA (Create user variable) Command used to define a user variable consisting of 2 alphanumeric characters.

```
PG 200            'Switch to program mode at address 200
VA Q1             'Define user variable Q1
LB K1             'Label command will name the program
 xxxxx
 xxxxx            'Program named by LB command
 xxxxx
 H 2000           'Hold 2 seconds before execution
 PR "Position =", P       'Print position
 BR K1, Q1<10     'Cond branch to K1 if Q1 less than 10
E                 'End Program
PG                'Switch out of program mode
```

Page intentionally left blank

4 Command summary

Lexium MCode supports multiple families of motion control devices. Not all instructions, variables and flags apply to all motion control products.

See the detailed descriptions for each command in Section 5 of this document for device compatibility

4.1 Setup instructions, variables and flags

Mnemonic	Function	Unit	Range	Syntax Example
AL	Retrieve all parameters	–	–	PR AL
BD	Communications BAUD rate	BAUD	48, 96, 19, 38, 11	BD=<baud>
CE	Control C software reset	–	–	CE=<0/1>
CK	Check sum enable	–	–	CK=<1/0>
CW	Trip output pulse width	–	0 to 255	CW=100
DE	Enable/disable drive	–	1/0	DE=<1/0>
DG	Disable global response	–	1/0	DG=<1/0>
DN	Device name	Character	a–z, A–Z, 0–9	DN=<char>
EM	Echo mode 0 (def)=full duplex, 1=half duplex	Mode	<0 to 3>	EM=<mode>
FD	Restore factory defaults	–	–	FD
IP	Initial parameters from NVM	–	–	IP
PY	Enable/disable party mode	Mode	1/0	PY=<mode>
QD	Queued	–	1/0	QD=<1/0>
UG	Upgrade firmware	Code	2956102	IMS Term. Upgrader
WT	Warning temperature	Degrees C	0 to 84	WT=<0 – 84>

Shaded commands apply only to units with and RS-422/485 interface.

4.2 Misc instructions, variables and flags

Mnemonic	Function	Unit	Range	Syntax Example
AL	All parameters, used with PR (Print)	–	–	PR AL
BY	BSY flag 1=prog. running	–	0/1	PR BY
CW	Trip output pulse width	–	0 to 255	CW=100
EF	Error flag	–	0/1	PR EF
ER	Error number variable	Number	–	PR ER
ES	Escape	–	–	ES=0
FD	Restore factory defaults	–	–	FDy
IF	Input to variable pending flag	–	–	PR IF
IT	Internal temperature	C°	-20 to 100	PR IT
IV	Input into variable	Number	–	IV <var>
PN	Part number	–	–	PR PN
PR	Print selected data and/or text	–	–	PR <data/text string>
R1	User register 1	Number	Signed 32 bit	R1=<number>
R2	User register 2	Number	Signed 32 bit	R2=<number>
R3	User register 3	Number	Signed 32 bit	R3=<number>
R4	User register 4	Number	Signed 32 bit	R4=<number>
SN	Serial number	–	–	PR SN
TM	Trip on main power loss	–	–	TM=<addr/label>
VR	Firmware version	Number	–	PR VR
UV	Read user variables	–	–	PR UV

4.3 Motion instructions, variables and flags

Mnemonic	Function	Unit	Range	Syntax Example
(–)	Move previously commanded move *MA/MR) mode at this value.	per mode		–<number>
A	Set acceleration	Steps/Sec2	1000000000	A=<accel>
D	Set deceleration	Steps/Sec2	1000000000	D=<decel>
HC	Set hold current	% (Percent)	0 to 100	HC=<percent>
HT	Set hold current delay time	milliseconds	0–65000	HT=<msec>
JE	Jog enable flag	–	0/1	JE<0/1>
LM	Limit stop mode	–	1–6	LM=<number>
MA	Move to abs. position	±Position	Signed 32 bit	MA <±pos>
MD	Motion mode setting	–	–	–
MR	Move to relative position	±Distance	Signed 32 bit	MR <±dist>
MS	Set microstep resolution	µsteps/step	MSEL Table	MS=<param>
MT	Motor settling delay time	milliseconds	0–65000	MT=<msec>
MV	Moving flag	–	–	PR MV
NE	Numeric enable	–	0/1	NE<0/1>
RC	Set run current	% (Percent)	1 to 100	RC=<percent>
RD	Reverse direction	–	–	RD=<0/1>
SL	Slew axis at velocity	Steps/sec	±5000000	SL <velocity>
V	Read current velocity	Steps/sec	±5000000	PR V
VC	Velocity changing flag	–	–	BR<addr>, VC
VI	Set initial velocity	Steps/sec	1–5000000	VI=<velocity>
VM	Set maximum velocity	Steps/sec	1–5000000	VM=<velocity>

4.4 Program instructions, variables and flags

Mnemonic	Function	Unit	Range	Syntax Example
BP	Break point	–	–	BP <addr>, <mode>
BR	Branch (conditional/unconditional)	–	–	BR <addr>, <cond>
BY	Program executing	–	–	PR BY
CL	Call subroutine (conditional/ unconditional)	–	–	CL <addr>, <cond>
CP	Clear [rogram	Address	1-767	CP <addr>
DC	Decrement variable	–	–	DC <var/ureg>
E	End program execution	–	–	E
ES	Set escape mode	–	0 – 3	ES=<0–3>
EX	Execute program at address using selected trace mode	1 to 767		EX <addr>, <mode>
H	Hold prog. execution blank/0=motion stops	milliseconds	Blank(0) 1-65000	H=<msec>
IC	Increment variable	–	–	IC <var>
L	List program	Address	1-767	L <addr>
LB	Create a program address label name			
LK	Lock user program		0/1	LK=<0/1>
OE	On error handler 0=Disabled	Address	0/1-767	OE <addr>
PG	Start program entry at specified address	–	Blank/1-767	PG <addr>
PS	Pause program execution	–	–	PS
RT	Return from subroutine	–	–	RT
RS	Resume paused program	–	–	RS
S	Save to NVM	–	–	S
SU	Start program on power on or reset	–	–	LB SU
VA	Create a user variable name			Mnemonic

4.5 I/O instructions, variables and flags

Mnemonic	Function	Unit	Range	Syntax Example
AO	Set attention output variable	–	0 – 4,294,967,295	Ao=<mask>
D1	Set input 1 digital filtering	Milliseconds	0–255	D1=<time>
D2	Set input 2 digital filtering	Milliseconds	0–255	D2=<time>
D3	Set input 3 digital filtering	Milliseconds	0–255	D3=<time>
D4	Set input 4 digital filtering	Milliseconds	0–255	D4=<time>
D5	Analog input digital filter	counts	0–1000	D5=<counts>
FC	Filter capture	–	0-9	FC=3
I1 -I4	Read input 1-4	–	0/1	PR Ix, BR Ix,<cond>
I5	Read input 5 (analog)	–	0–1024	PR I5, BR I5,<cond>
I6	Read encoder index mark low true	–	–	PR I6
IN	Read inputs 1–4 as one value	data	0–15	PR IN
IS	Setup input function (inputs 1-4)	–	–	Is=<line>, <type>, <active>
	Setup input function (analog input)	–	–	Is=5, <type>, <active>
	Setup input function (Encoder index)			Is=6, 0, <active>
IT	Internal temperature	C°	-20 to 100	PR IT
O1-O3	Set output x to logic state	–	0/1	Ox=<1/0>
OF	Output fault	—	0-3	PR OF
OS	Setup output functions			Os=<line>, <type>, <active>
OT	Write aata to outputs 1–3 as one value	data	0–255	OT=<data>
TI	Trip on input	–	–	TI <input>,<addr>
TE	Trip enable	See Table	<1–4>	TE=<num>

4.6 Position related instructions, variables and flags

Mnemonic	Function	Unit	Range	Syntax Example
C1	Set Counter 1	Motor Counts	Signed 32 bit	C1=<counts>
HM	Home to home switch	Type	1 to 4	HM <type>
P	Set/read position	Motor/Encoder Counts	Signed 32 bit	P=<counts>
PC	Read captured position at trip	Motor/Encoder Counts	Signed 32 bit	PR PC
TP	Trip on position	Position	–	<pos>, <addr>
TR	Trip on relative position	Position	–	TR <pos>, <addr>
TE	Trip enable	See Table	<0-3>	TE=<num>

4.7 Encoder related instructions, variables and flags C

Mnemonic	Function	Unit	Range	Syntax Example
C2	Set counter 2	Encoder counts	Signed 32 bit	C2=<counts>
DB	Set encoder deadband	Encoder Counts	0-65000	DB=<counts>
EE	Enable/disable encoder functions	–	1/0	EE=<1/0>
HI	Home to encoder index	Type	1 to 4	HI=<type>
I6	Read encoder index mark	–	–	I6
PM	Position maintenance enable flag		0/1	PM=<0/1>
SF	Set stall factor	Encoder Counts	0-65000	SF=<counts>
SM	Set stall mode	0=Stop Motor/1=Don't Stop	1/0	SM=<mode>
ST	Stall flag	–	0/1	PR ST

4.8 Hybrid specific instructions, variables and flags C

Mnemonic	Function	Unit	Range	Syntax Example
AF	Read hybrid status	—	128 bit binary	PR AF BR <address>, AF=x CL <address>, AF=x
AO	Attention output line mask settings	—	0 - 4,294,967,295	PR Ao Ao=<mask>
AS	Set hybrid mode	—	0 = Off 1 = Fixed current (RC, HC) 2 = Variable current (RC max) (default) 3 = Torque	AS=0
CB	Set control bounds	—	0 = 1.1 Full step (torque) 1 = 1.3 Full step (default) 2 = 1.5 Full step 3 = 1.7 Full step (speed)	CB=2
CF	Clear locked rotor fault	—	—	CF
EL	Remote encoder line count	Lines	—	EL=512
LD	Set rotor lead limit	step	0 to 2147483647 (102400 default)	LD=256000
LG	Set rotor lag	step	0 to 2147483647 (102400 default)	LG=256000
LL	Read position lead/lag limit	steps	-2147483647 to +2147483647	PR LL BR <address>, LL=x CL <address>, LL=x
LR	Locked rotor flag	—	0/1	PR LR BR <address>, LR=1 CL <address>, LR=1
LT	Set locked rotor timeout	mS	2 to 65535 (2000 default)	LT=1000
MF	Set make-up frequency	Hz	306 to 5000000 (768000 default)	MF=51200
MU	Set make-up mode	—	0 = Off (default) 1 = Use MF (make-up freq) 2 = Use system speed Clear parameter (not saved) 1 = Use LL 2 = Clear LL	MU=1,1

SC	Start configuration test	—	1 - Configuration test	SC=1
TA	Trip on Hybrid status		1 = Calibration done 2 = Hybrid active 4 = Locked rotor 8 = Lag limit 16 = Lead limit	TA=4,<address,label>
TD	Set torque direction	—	0 = Minus 1 = Plus (default)	TD=1
TQ	Set torque percent	%	1 to 100 (25 default)	TQ-50
TS	Set torque mode speed	Steps/ sec	46512 — 2560000	TS=50000
VF	Torque velocity filter	Counts	0 to 1000 (0 default)	VF=100

4.9 Math functions

Symbol	Function
+	Add Two Variables and/or Flags
-	Subtract Two Variables and/or Flags
*	Multiply Two Variables and/or Flags
/	Divide Two Variables and/or Flags
<>	Not Equal
=	Equal
<	Less Than
<=	Less Than and/or Equal
>	Greater Than
>=	Greater Than and/or Equal
&	AND (Bitwise)
|	OR (Bitwise)
^	XOR (Bitwise)
!	NOT (Bitwise)

5 Command details

A (Acceleration)

Compatible with Lexium MDrive products:
Open Loop (RS-422/485)
Closed Loop (RS-422/485)
Open Loop (Ethernet)
Closed Loop (Ethernet)

Mnemonic	A
Function	Set acceleration
Type	Variable
Description	The A Variable sets the acceleration rate when changing velocity in steps per second^2. If the A was set at 76800 per second^2 the motor would accelerate at a rate of 76800 counts per second, every second. If the maximum velocity was set at 768000 microsteps per second it would take 10 seconds to reach maximum speed if VI=0.
Syntax	A=<mode>
Units	Steps/sec^2 (EE=0)/Counts/sec^2 (EE=1)
Range	91 to 1525878997 (steps), 91 to 61035160 (encoder counts)
Default	Default: 1000000 (EE=0), 40000 (EE=1)
Usage	Program/Immediate, Read/Write

Code Example	`A=20000  'Set Accel to 20000 steps/sec2`
	`PR A  'Print acceleration`
	`A=Q1  'Set Accel to user var Q1`

Equivalents	***MODBUS/TCP:*** 0x0000 ***EtherNet/IP:*** class: 0x66 instance: 1 attribute: 0x01
Related	D

AJ (Reserved)

Mnemonic	AJ
Function	Reserved for internal use
Description	Reserved for future use, will return an error if used. Do not use as a user variable, flag or program label.

AL (Retrieve all parameters)

Compatible with Lexium MDrive products:
Open Loop (RS-422/485)
Closed Loop (RS-422/485)
Open Loop (Ethernet)
Closed Loop (Ethernet)

Mnemonic	AL
Function	Retrieve all parameters
Type	Variable
Description	The AL variable is used with the PR (PRINT) instruction to print the value/state of all variables and flags to the terminal program.
Syntax	PR AL
Usage	Immediate, Read
Code Example	`PR AL`
Equivalents	***MODBUS/TCP:*** — ***EtherNet/IP:*** —

AF (HMT flags)

Compatible with Lexium MDrive products:
Closed Loop (RS-422/485)
Closed Loop (Ethernet)

Mnemonic	AF	
Function	Read Hybrid status	
Type	Read-only status flag	
Description	The AF flag will print the status of the Hybrid logic.	
Syntax	PR AF BR <address/label>, AF=<condition> CL <address/label>, AF=<condition>	
Units	—	
Range	128-bit binary	
Conditions	1	Lead limit reached
	2	Lag limit reached
	4	Maximum lead/lag limit reached
	8	Locked rotor
	16	Hybrid mode is active
	32	Hardware fault condition exists
	64	At zero
	128	Calibration is complete
	256	Calibration fault
Notes	If multiple conditions exist the result is additive. i.e. At zero (64) and Calibration complete (128) AF=192	
Usage	Program/Immediate, Read only	
Code Example	`PR AF  'print AF to the terminal` `BR XY, AF=128 'branch to XY when calibration complete`	
Equivalents	***MODBUS/TCP:*** 0x008F ***EtherNet/IP:*** class: 0x6A instance: 1 attribute: 0x01	
Related	AS, TA	

AO (Attention output mask settings)

Compatible with Lexium MDrive products:
Open Loop (RS-422/485)
Closed Loop (RS-422/485)
Open Loop (Ethernet)
Closed Loop (Ethernet)

Mnemonic	AO
Function	Set Attention output mask
Type	Variable
Description	The AO flag will set the condition on which the attention output will trigger LED 2 and, if configured, to an output assigned to the Attention output type..
Syntax	PR AO AO = <value>
Units	—
Range	0 - 4,294,967,295

Conditions

mask	description	LED color (Green = default)
1	Error flag	Red
2	Locked rotor	Red
4	Lead limit	Red
8	Lag limit	Red
16	hMT active	Red
32	Calibration active	Red
64	Over temperature	Red
128	Software error	Red
256	At zero cross	Red
512	Current reduction active	Red
1024	Make-up active	Red
2048	Drive disabled (DE=0)	Red
4096	Warning temperature	Red
8192	Voltage warning	Red
16384	Moving flag	Red
32768	Stall flag	Red

Notes	If multiple conditions need to trigger the output the result is additive. i.e. Lead limit (4) and Lag limit (8) AO=12, Moving flag (16384) and Stall Flag (32768) AO=49152
Usage	Program/Immediate, Read only

Code Example	`AO=512  'Activate attn out (LED 2) when`
	`'current reduction is active`
	
	`PR AO  'print AO to the terminal`

Equivalents	***MODBUS/TCP:*** TBD ***EtherNet/IP:*** TBD
Related	OS

NOTE: Highlighted function not available on open loop models.

AS (hMTechnology mode)

Compatible with Lexium MDrive products:
Closed Loop (RS-422/485)
Closed Loop (Ethernet)

⚠WARNING

EXECUTION OF MOTION

Changing hMT mode to torque mode (AS=3) will result in immediate motion at the velocity specified by the torque speed (TS).

- Motion will occur immediately on AS=3

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

Mnemonic	AS	
Function	Set Hybrid operational mode	
Type	Variable	
Description	The AS variable will set the operating mode for thehMTechnology device to one of four modes, which are detailed below. These modes will determine the active state of the hMT circuitry and the behavior characteristics of the closed ;loop Lexium MDrive based upon motor current or motor torque.	
Syntax	AS=<mode>	
Units	—	
Range	0-3	
Modes	0	Hybrid circuitry off.
	1	Fixed current mode, motor current will be as specified by the run current (RC) and hold current (HC) variables.
	2	Variable current mode, motor current will vary as needed to move/position the load with a maximum current level established by the run current (RC) variable..
	3	Torque mode, motor torque and speed will vary as needed to move/position the load at the maximum torque specified by the set torque percent variable (TQ) at the maximum speed as specified by the set torque speed variable (TS).
Default	0 (HMT Off)	
Usage	Program/Immediate, Read/Write	
Code Example	`AS=1  'set hybrid mode to fixed current`	
Equivalents	***MODBUS/TCP:*** 0x008E ***EtherNet/IP:*** class: 0x6A instance: 1 attribute: 0x02	
Related	RC, HC, HT, TD, TQ, TS	

AT (Reserved)

Mnemonic	AT
Function	Reserved for internal use
Description	Reserved for internal use, will return an error if used. Do not use as a user variable, flag or program label.

BD (Communications BAUD rate)

Compatible with Lexium MDrive products:
Open Loop (RS-422/485)
Closed Loop (RS-422/485)

Mnemonic	BD
Function	Set communications BAUD rate
Type	Variable
Description	This variable sets the baud rate for serial communications with the Lexium MCode device. It sets the rate for the RS-422/485 interface. The baud rate is set by indicating the first two digits of the desired rate as shown in the range section below. In order for the new BAUD rate to take effect, the user must issue the S (SAVE) instruction and then reset the device. When the Lexium MCode device is reset, it will communicate at the new BAUD rate.
Syntax	BD=<mode>
Units	Bits/second
Range	48, 96, 19, 38, 11
Modes	48 4800 bps 96 9600 bps (default) 19 19200 bps 38 38400 bps 11 115200 bps .
Default	96 (9600 bps)
Usage	Program/Immediate, Read/Write
Code Example	`BD=19 'set BAUD rate to 19200 bps` `S 'save to NVM`
Equivalents	***MODBUS/TCP:*** — ***EtherNet/IP:*** —
Notes	If you change the Baud Rate in the Lexium MCode device it must be matched in Lexium MDrive Programmer. A delay time between the command requests to the device must be considered to allow it time to interpret a command and answer the host before a subsequent command can be sent. The time between requests is dependent on the command and the corresponding response from the device.
Related	CK

BE (Reserved)

Mnemonic	BE
Function	Reserved for internal use
Description	Reserved for future use, will return an error if used. Do not use as a user variable, flag or program label.

BL (Reserved)

Mnemonic	BL
Function	Reserved for internal use
Description	Reserved for future use, will return an error if used. Do not use as a user variable, flag or program label.

BM (Reserved)

Mnemonic	BM
Function	Reserved for internal use
Description	Reserved for future use, will return an error if used. Do not use as a user variable, flag or program label.

BP (Break point)

Compatible with Lexium MDrive products:
Open Loop (RS-422/485)
Closed Loop (RS-422/485)
Open Loop (Ethernet)
Closed Loop (Ethernet)

Mnemonic	BP
Function	Set break point/execution mode
Type	Instruction
Description	The BP, or Break Point Instruction allows the user to set break points within an Lexium MCode program to help in debugging the program. To use the BP instruction the program must be executed in either trace or single-step mode. The program will then run normally the number of times specified by the count, then go into single-step mode at the address or label specified by BP. Press the spacebar to step through the program if in single step mode. To disable the break point, set BP=0.
Syntax	BP <address/label>,<count>
Usage	Program/Immediate

Code Example	`BP X1, 3  'Break  at label X1 after 3 cycles`
	
	`EX P1, 1  'Execute  P1 in trace mode`
	`or`
	`EX P1,2  'Execute P1 in single step mode`

Equivalents	***MODBUS/TCP:*** — ***EtherNet/IP:*** —
Related	`EX`

BR (Program branch)

Compatible with Lexium MDrive products:
Open Loop (RS-422/485)
Closed Loop (RS-422/485)
Open Loop (Ethernet)
Closed Loop (Ethernet)

Mnemonic	BR
Function	Conditional or unconditional program branch.
Type	Instruction
Description	The branch instruction can be used to perform a conditional or unconditional branch to a routine in a Lexium MCode device program. It can also be used to perform loops and IF THEN logic within a program. There are two parameters to a branch instruction. These are used to perform two types of branches: **Conditional Branch** This type of branch first specifies an address or user label where program execution should continue if the second parameter, the condition, is true. The condition parameter may include flags as well as logical functions that are to be evaluated. Only one condition may exist. **Unconditional Branch** In this type of branch the second parameter is not specified, then the execution will continue at the label or address specified by the first parameter.
Syntax	BR <address/label>,<condition>
Usage	Program

Code Example	`BR 256, I2=1 'Cond. branch to address 256 'if input 1 = ACTIVE`
	`BR G1 'Unconditional branch to 'program label G1`
	`BR G2, Q4<10 'Cond branch to program G2 if 'user var Q4 is less than 10`

Equivalents	***MODBUS/TCP:*** — ***EtherNet/IP:*** —
Related	`EX`

BY (Program executing)

Compatible with Lexium MDrive products:
Open Loop (RS-422/485)
Closed Loop (RS-422/485)
Open Loop (Ethernet)
Closed Loop (Ethernet)

Mnemonic	BY
Function	Busy (program executing)
Type	Status flag
Description	The BY flag will indicate the status of program execution,
Syntax	PR BY
Response	0 - No programs running, 1 - program executing
Default	0
Usage	Immediate, Read only
Code Example	`PR BY`
Equivalents	***MODBUS/TCP:*** — ***EtherNet/IP:*** —
Related	`PR`

C1 (Counter 1)

Compatible with Lexium MDrive products:
Open Loop (RS-422/485)
Closed Loop (RS-422/485)
Open Loop (Ethernet)
Closed Loop (Ethernet)

Mnemonic	C1
Function	Counter 1 (position counter)
Type	Variable
Description	This variable contains the count of the clock pulses generated by the Lexium MCode compatible device. Counter 1 may be preset if necessary
Syntax	C1=<steps>
Units	Motor steps
Range	-2147483648 to +2147483647
Default	0
Usage	Program/Immediate, Read/Write

Code Example	`C1=20000 'Set counter 1 to 20000 motor steps`
	`PR C1 'Print the value of C1 to the terminal screen`
	`CL K5,C1>2100000 'Call subroutine K5 if C1>2100000`

Equivalents	***MODBUS/TCP:*** 0x0005 ***EtherNet/IP:*** class: 0x68 instance: 1 attribute: 0x01
Related	C2, P

C2 (Counter 2)

Compatible with Lexium MDrive products:
Closed Loop (RS-422/485)
Closed Loop (Ethernet)

Mnemonic	C2
Function	Counter 2 (encoder counter)
Type	Variable
Description	This variable contains the encoder edge count received by the Lexium MCode compatible device. Counter 1 may be preset if necessary
Syntax	C2=<counts>
Units	Encoder counts
Range	-2147483648 to +2147483647
Default	0
Usage	Program/Immediate, Read/Write

Code Example	`C2=512 'Set counter 2 to 512 counts`
	`PR C2 'Print the value of C2 to the terminal screen`
	`CL K5,C2>2100000 'Call subroutine K5 if C2>2100000`

Equivalents	***MODBUS/TCP:*** 0x0007 ***EtherNet/IP:*** class: 0x69 instance: 1 attribute: 0x01
Related	C1, P

CB (Control bounds)

Compatible with Lexium MDrive products:
Closed Loop (RS-422/485)
Closed Loop (Ethernet)

Mnemonic	CB
Function	Set control bounds
Type	Variable
Description	There are four limits, or control bounds that can be selected. They are 1.1, 1.3, 1.5 and 1.7 full motor steps. Bounds of 1.1 will produce greater torque performance, though maximum speed will be reduced. Bounds of 1.7 will produce greater speed performance, though transient response is decreased. Best overall torque-speed performance is achieved with bounds set at 1.3 or 1.5.
Syntax	CB=<bounds>
Units	full motor step
Range	0 — 3
Bounds	0 1.1 full motor steps — best torque performance 1 1.3 full motor steps 2 1.5 full motor steps (best overall performance) 3 1.7 full motor steps — best speed performance
Default	1 (1.3 full motor steps)
Usage	Program/Immediate, Read/Write
Code Example	`CB=2 'set control bounds to 1.5 steps`
Equivalents	***MODBUS/TCP:*** 0x0091 ***EtherNet/IP:*** class: 0x6A instance: 1 attribute: 0x04
Notes	For torque mode operation the bounds are preset.
Related	

CE (Software reset enable)

Compatible with Lexium MDrive products:
Open Loop (RS-422/485)
Closed Loop (RS-422/485)
Open Loop (Ethernet)
Closed Loop (Ethernet)

Mnemonic	CE	
Function	Software reset enable (CTRL+C)	
Type	Setup flag	
Description	This setup flag will configure the device to respond or not respond to a CTRL+C software reset.	
Syntax	CE=<mode>	
Range	0 - 2	
Modes	0	Disables CTRL+C response
	1	Enables CTRL+C response (default)
	2	Is addressable in party mode (PY=1), CTRL+C will respond the same as CE=1 when not in party mode.
Default	1	
Usage	Program,/Immediate, Read/Write	
Code Example	`CE=0 'disable CTRL+C response`	
Equivalents	***MODBUS/TCP:*** 0x0009 ***EtherNet/IP:*** class: 0x64 instance: 1 attribute: 0x01	

CF (Clear locked rotor fault)

Compatible with Lexium MDrive products:
Closed Loop (RS-422/485)
Closed Loop (Ethernet)

Mnemonic	CF
Function	Clear locked rotor flault
Type	Instruction
Description	The CF instruction will clear a locked rotor fault then re-enable the output bridge. A locked rotor will be indicated by both the Locked Rotor flag (LR=1) and by an error condition (error 104).
Syntax	CF
Usage	Program/Immediate, Read/Write
Code Example	`CF 'clear locked rotor fault`
Related	LR
Equivalents	***MODBUS/TCP:*** 0x0093 ***EtherNet/IP:*** class: 0x6A instance: 1 attribute: 0x08

CK (Checksum enable)

Compatible with Lexium MDrive products:
Open Loop (RS-422/485)
Closed Loop (RS-422/485)

Mnemonic	CK	
Function	Check sum enable	
Type	Flag	
Description	This setup flag will configure the device operate in check sum mode where a check sum is required following the command.	
Syntax	CK=<mode>	
Range	0 - 2	
Modes	0	Checksum mode disabled (default)
	1	Puts the device into Check Sum Mode. When enabled, all communications with the device require a Check Sum to follow the commands. The Check Sum is the 2's complement of the 7 bit sum of the ASCII value of all the characters in the command "OR"ed with 128 (hex = 0x80). The command will be acknowledged with a NAK (0x15) if the Check Sum is incorrect or an ACK (0x06) when the command is correctly processed (no error).
	2	CK=2 will enable check sum mode, however NAK only sent for bad check sum. "ACK" is not echoed if a program is running. Only a NAK is echoed if an error occurs. In immediate mode both ACK or NAK characters are echoed.
Default	0	
Usage	Program,/Immediate, Read/Write	
Code Example		

```
MR 1                          'Lexium MCode Command
77  82  32  49               'Decimal Value
4D  52  20  31                'Hex
77 + 82 + 32 + 49 = 240       'Add decimal values together
1111 0000  240                'Change 240 decimal to binary
0000 1111                     '1's complement
0001 0000                     'Add 1 (results in the 2's
1000 0000                     'complement)
1001 0000  144                'OR result with 128
                              'result Check Sum value in
                              'decimal
```

Equivalents	***MODBUS/TCP:*** — ***EtherNet/IP:*** —
Notes	To Send the checksum, in IMS terminal use ALT+ Checksum (In the example ALT+0144) The Response will be 06
Related	BD

CL (Call subroutine)

Compatible with Lexium MDrive products:
Open Loop (RS-422/485)
Closed Loop (RS-422/485)
Open Loop (Ethernet)
Closed Loop (Ethernet)

Mnemonic	CL
Function	Call subroutine
Type	Instruction
Description	This function can be used to invoke a subroutine within a program. This allows the user to segment code and call a subroutine from a number of places rather than repeating code within a program. There are two parameters to the CL instruction. The first specifies the program address or label of the subroutine to be invoked if the second parameter, the condition, is true. If the second parameter is not specified, the subroutine specified by the first parameter is always invoked. The condition parameter can include flags as well as logical functions that are to be evaluated. There can only be one condition. The subroutine should end with a RT (Return) instruction. The RT instruction will cause program execution to return to the line following the CL instruction.
Syntax	CL=<address/label>,<condition>
Usage	Program

Code Example	`CL 256, I5<512  'Call Sub at 256, analog in < than 512`
	`CL K5  'Unconditional call to subroutine label K5`
	`CL K8, I4=0  'Call sub k8 if input 4 is INACTIVE`

Equivalents	***MODBUS/TCP:*** *—* ***EtherNet/IP:*** *—*
Notes	The called subroutine should end with a return command (RT)
Related	RT

CP (Clear program)

Compatible with Lexium MDrive products:
Open Loop (RS-422/485)
Closed Loop (RS-422/485)
Open Loop (Ethernet)
Closed Loop (Ethernet)

Mnemonic	CP
Function	Clear program
Type	Instruction
Description	This instruction will clear the program space in the NVM as specified by the instruction parameter. Programs are stored directly to the NVM and executed from there. Will clear program addresses only. Will not clear globally declared user variable or flags. FD will clear program memory as well. A save (S) command must be issued following a CP to permanently clear the program.
Syntax	CP <address/label>
Usage	Immediate

Code Example	`CP 256  'Clear prog. space beginning at addr. 256`
	`CP G3  'Clear program space beginning at label G3`
	`CP  'Clear all of program space`

Equivalents	***MODBUS/TCP:*** — ***EtherNet/IP:*** —
Related	FD, IP

CW (Clock width)

Compatible with Lexium MDrive products:
Open Loop (RS-422/485)
Closed Loop (RS-422/485)
Open Loop (Ethernet)
Closed Loop (Ethernet)

Mnemonic	CW
Function	Set clock width
Type	Variable
Description	Sets the pulse width for the trip output only in 50 nano second increments
Syntax	CW=<time>
Units	Nanoseconds
Range	0 - 255
Default	500 nS
Usage	Program/Immediate, Read/Write
Code Example	`CW=100  'output pulse width to 5µS (100 x 50nS)`
Equivalents	***MODBUS/TCP:*** — ***EtherNet/IP:*** class: 0x64 instance: 1 attribute: 0x05
Related	CM, PC

D (Deceleration)

Compatible with Lexium MDrive products:
Open Loop (RS-422/485)
Closed Loop (RS-422/485)
Open Loop (Ethernet)
Closed Loop (Ethernet)

Mnemonic	D
Function	Set deceleration
Type	Variable
Description	The D variable sets the deceleration of the device in steps per second^2. If the D was set at 76800 per second^2 the motor would decelerate at a rate of 76800 per second, every second. If the device was running at a maximum velocity of 768000 microsteps per second it would take 10 seconds to decelerate if VI=0.
Syntax	D=<steps/counts>
Units	Steps/Sec^2 (EE=0)/Counts/Sec^2 (EE=1)
Range	91 to 1525878997 (Steps EE=0), 91 to 61035160 (Counts EE=1)
Default	1000000(EE=0), 40000 (EE=1)
Usage	Program/Immediate, Read/Write
Code Example	`D=20000 'set acceleration to 20000 step/sec2` `D=A 'set deceleration equal to acceleration`
Equivalents	***MODBUS/TCP:*** 0x0018 ***EtherNet/IP:*** class: 0x66 instance: 1 attribute: 0x02
Related	A, C1, C2, P

D1-D4 (Input switch debounce)

Compatible with Lexium MDrive products:
Open Loop (RS-422/485)
Closed Loop (RS-422/485)
Open Loop (Ethernet)
Closed Loop (Ethernet)

Mnemonic	D1-D4
Function	Set input switch debounce
Type	Variable
Description	This variable will set the digital filtering to be applied to the selected input 1 - 4. The input must be stable for “time” amount of milliseconds before a change in state is detected.
Syntax	D1-4=<time>
Units	Milliseconds
Range	0 to 255
Default	0
Usage	Program/Immediate, Read/Write
Code Example	`D1=0 'No debounce` `D1=150 'Set filtering to 150 msec`
Equivalents	***MODBUS/TCP:*** 0x000F, 0x0010, 0x0011, 0x0012 ***EtherNet/IP:*** class: 0x67 instance: 1 attribute: 0x01-0x04
Related	I1 - I4

D5 (Analog input filter)

Compatible with Lexium MDrive products:
Open Loop (RS-422/485)
Closed Loop (RS-422/485)
Open Loop (Ethernet)
Closed Loop (Ethernet)

Mnemonic	D5
Function	Set analog input filter
Type	Variable
Description	D5 is a continuous filtering process. It does a running average by computing: (((X-1)/X)*current reading) + (1 / X) If X = 10, then: ((current averaged value * 9)/10) + (new reading / 10) == NEW current averaged value.
Syntax	D5=<X>
Units	Counts
Range	0 to 255
Default	0
Usage	Program/Immediate, Read/Write
Code Example	`D5=0 'No debounce` `D5=100 'Set filtering to 100 counts`
Equivalents	***MODBUS/TCP:*** 0x0013 ***EtherNet/IP:*** class: 0x67 instance: 1 attribute: 0x05

DB (Encoder deadband)

Compatible with Lexium MDrive products:
Closed Loop (RS-422/485)
Closed Loop (Ethernet)

Mnemonic	DB
Function	Set encoder deadband
Type	Variable
Description	This variable defines the plus (+) and minus (-) length of the encoder deadband in encoder counts. When the encoder is enabled, a move is not completed until motion stops within DB. If PM=1 device will correct if pushed outside of DB value once in position.
Syntax	DB=<counts>
Units	Encoder counts
Range	0 to 65000
Default	1
Usage	Program/Immediate, Read/Write
Code Example	`DB=10  'Set encoder deadband to ±10 counts`
Equivalents	***MODBUS/TCP:*** 0x001A ***EtherNet/IP:*** class: 0x69 instance: 1 attribute: 0x02
Related	EE, C2, SF, SM, ST, PM

DC (Decrement variable)

Compatible with Lexium MDrive products:
Open Loop (RS-422/485)
Closed Loop (RS-422/485)
Open Loop (Ethernet)
Closed Loop (Ethernet)

Mnemonic	DC
Function	Decrement variable
Type	Instruction
Description	The DC instruction will decrement the specified variable by one.
Syntax	DC <variable>
Usage	Program/Immediate, Read/Write
Code Example	`DC R1  'Decrement register r1` `DC K5  'Decrement user variable K5`
Equivalents	***MODBUS/TCP:*** — ***EtherNet/IP:*** —
Related	IC

DE (Drive enable)

Compatible with Lexium MDrive products:
Open Loop (RS-422/485)
Closed Loop (RS-422/485)
Open Loop (Ethernet)
Closed Loop (Ethernet)

Mnemonic	DE	
Function	Set drive enabled/disabled	
Type	Variable	
Description	The DE flag enables or disables the drive portion of the Lexium MCode compatible device.	
Syntax	DE=<0/1>	
Range	0/1	
States	0	Drive output bridge disabled.
	1	Drive output bridge enabled (default).
Default	1	
Usage	Program/Immediate, Read/Write	
Code Example	`DE=0 'Disable the motor driver section` `DE=1 'Enable the motor driver section`	
Equivalents	***MODBUS/TCP:*** 0x001C ***EtherNet/IP:*** class: 0x64 instance: 1 attribute: 0x05	

DG (Disable global response)

Compatible with Lexium MDrive products:
Open Loop (RS-422/485)
Closed Loop (RS-422/485)

Mnemonic	DG	
Function	Enable/disable global response in party mode	
Type	Flag	
Description	The DG flag enables or disables device response to global commands made while in Party Mode. In the default state (DG=1) the device will not respond but will execute global commands. By setting the the DG flag to 0, that device will respond to global commands.	
Syntax	DG=<0/1>	
Range	0/1	
States	0	Response to global commands enabled.
	1	Response to global commands disabled (default).
Default	1 (disabled)	
Usage	Program/Immediate, Read/Write	
Code Example	`DG=0 'Enable response to global commands` `DG=1 'Disable Response to global commands no echo`	
Equivalents	***MODBUS/TCP:*** — ***EtherNet/IP:*** —	
Related	DN, PY	

DN (Device name)

Compatible with Lexium MDrive products:
Open Loop (RS-422/485)
Closed Loop (RS-422/485)

Mnemonic	DN
Function	Set device name for party mode.
Type	Variable
Description	DN sets the name of the device for party mode communications. The acceptable range of characters is a-z, A-Z, 0-9. The factory default is "!" Once named, the device name must precede the instruction to that drive. When assigning a device name, the character MUST be within quotation marks. The name is case sensitive. Refer to Section 2.2 in the Hardware Reference for specific Party Mode configuration and use instructions. Once the default character has been changed it can not be reset to the default character (!) except on a FD (Factory Default Reset).
Syntax	DN=<"ascii character">
Units	ASCII Characters
Range	a-z, A-Z, 0-9
Default	!
Usage	Program/Immediate, Read/Write
Code Example	`DN="A"  'Set the device name to the character A` `DN="65"  'Set the device name to the character A*`
Equivalents	***MODBUS/TCP:*** — ***EtherNet/IP:*** —
Notes	Note: Must enter S (Save) prior to any reset of the DN will be lost.
Related	PY, S

E (End program)

Compatible with Lexium MDrive products:
Open Loop (RS-422/485)
Closed Loop (RS-422/485)
Open Loop (Ethernet)
Closed Loop (Ethernet)

Mnemonic	E
Function	End program
Type	Instruction
Description	Stops the execution of a program. Used in program mode to designate the end of the program
Syntax	E
Usage	Program
Code Example	`E  'End program`
Equivalents	***MODBUS/TCP:*** — ***EtherNet/IP:*** —
Related	PG, EX

EE (Encoder enable)

Compatible with Lexium MDrive products:
Closed Loop (RS-422/485)
Closed Loop (Ethernet)

Mnemonic	EE
Function	Enable/disable encoder
Type	Flag
Description	The EE flag enables or disables the optional encoder mode of the Lexium MCode compatible device. When in Encoder Mode, all moves are done by Encoder Counts. The 1000 line Encoder generates counts in a Quadrature format which results in 4000 counts per revolution.
Syntax	EE=<0/1>
Range	0/1
States	0 Disable the encoder (default state) 1 Enable the encoder
Default	0 (disabled)
Usage	Program/Immediate, Read/Write
Code Example	`EE=0 'Disable the encoder` `EE=1 'Enable encoder mode`
Equivalents	***MODBUS/TCP:*** 0x001E ***EtherNet/IP:*** class: 0x69 instance: 1 attribute: 0x03
Related	DB, C2, SF, SM, ST, PM, FM

EF (Error flag)

Compatible with Lexium MDrive products:
Open Loop (RS-422/485)
Closed Loop (RS-422/485)
Open Loop (Ethernet)
Closed Loop (Ethernet)

Mnemonic	EF
Function	Error condition exists
Type	Flag
Description	The Error flag will indicate whether or not an error condition exists. It is automatically cleared when a new program is executed. The only way to manually clear the EF flag is to read the value of the ER variable or set ER=0. There is an instruction, OE, which allows the user to specify the execution of a subroutine in the program memory when an error occurs. The subroutine might contain instructions to read the ER variable which would clear the EF flag.
Syntax	PR EF
Response	0/1
States	0 No error exists 1 Error condition exists
Default	0 (no error exists)
Usage	Program/Immediate, Read

Code Example	`PR EF 'Read the state of the error flag`
	`'Response = 0: No error exists`
	`'Response = 1: Error condition exists,`
	`'Error value exists`

Equivalents	***MODBUS/TCP:*** 0x001F ***EtherNet/IP:*** class: 0x65 instance: 1 attribute: 0x04
Related	ER, OE

EL (Reserved)

Mnemonic	EL
Function	Reserved for internal use
Description	Reserved for future use, will return an error if used. Do not use as a user variable, flag or program label.

EM (Echo mode)

Compatible with Lexium MDrive products:
Open Loop (RS-422/485)
Closed Loop (RS-422/485)

Mnemonic	EM	
Function	Set echo mode	
Type	Flag	
Description	The Echo Mode Flag will set the full/half duplex configuration of the RS-422/485 channel.	
Syntax	EM=<0-3>	
Range	0 - 3	
States	0	Echo all information back over communications line. CR/LF indicates command accepted (full duplex) (default).
	1	Don't echo the information, only send back prompt. CR/LF indicates command accepted (half duplex).
	2	Does not send prompt, only responds to PRINT (PR) and LIST (L) commands.
	3	Saves Echo in Print Queue then executes. Prints after command is terminated.
Default	0 (disabled)	
Usage	Program/Immediate, Read	
Code Example	`EM=1 'Don't echo the information,`	
Equivalents	***MODBUS/TCP:*** — ***EtherNet/IP:*** —	
Related	ER, OE	

ER (Error)

Compatible with Lexium MDrive products:
Open Loop (RS-422/485)
Closed Loop (RS-422/485)
Open Loop (Ethernet)
Closed Loop (Ethernet)

Mnemonic	ER
Function	Display error code
Type	Variable
Description	The ER variable indicates the Lexium MCode device error code for the most recent error that has occurred. The ER variable must be read or set to zero to clear the EF flag. A Question Mark <?> in place of the normal cursor indicates an ERROR. A list of Error codes are located at the end of this section..
Syntax	PR ER
Units	Numeric error code
Default	0
Usage	Program/Immediate, Read/Write
Code Example	`PR ER  'Read the error number, result = <Value>` `ER 0  'Set the error value to zero`
Equivalents	***MODBUS/TCP:*** 0x00121 ***EtherNet/IP:*** class: 0x65 instance: 1 attribute: 0x05
Notes	Note: See Error code section for full listing of error codes.
Related	EF, OE

ES (Escape)

Compatible with Lexium MDrive products:
Open Loop (RS-422/485)
Closed Loop (RS-422/485)

Mnemonic	ES	
Function	Set escape mode	
Type	Flag	
Description	ESC flag to switch between ESC and CTRL+E. An Escape will stop both the program and the motion.	
Syntax	ES=<0-3>	
Range	0 - 3	
Modes	0	Escape Flag set to respond to CTRL+E
	1	Escape Flag set to respond to ESC keypress (default)
	2	Escape Flag set to respond addressable CTRL+E (party mode)
	3	Escape Flag set to respond to addressable ESC keypress (party mode)
Default	1	
Usage	Program/Immediate, Read/Write	
Equivalents	***MODBUS/TCP:*** — ***EtherNet/IP:*** —	
Code Example	`ESC=0  'ESC responds to CTRL+E`	

EX (Execute program)

Compatible with Lexium MDrive products:
Open Loop (RS-422/485)
Closed Loop (RS-422/485)
Open Loop (Ethernet)
Closed Loop (Ethernet)

Mnemonic	EX	
Function	Execute program	
Type	Instruction	
Description	Execute program at a specified address or label using a selected trace mode. Used in immediate mode. There are three modes of program execution.	
Syntax	EX <address/label>,<mode>	
Modes	0	Normal execution, is specified by a mode of 0 (or simply leaving the mode blank).
	1	Trace mode is specified by a mode of 1. This means that the program executes continuously until the program E is encountered, but the instructions are “traced” to the communications port so the user can see what instructions have been executed.
	2	Single step mode is specified by a mode of 2. In this mode, the user can step through the program using the space bar to execute the next line of the program. The program can be resumed at normal speed in this mode by pressing the enter key.
Usage	Immediate	

Code Example	`EX 1 'Execute program at address 1 normally`
	`EX G2,1 'Execute program G2 in trace mode`
	`EX 200,2 'Execute program at address 200 in 'single-step mode`	

Equivalents	***MODBUS/TCP:*** 0x0041, 0x0024 ***EtherNet/IP:*** —	

F1 - F4 (Reserved)

Mnemonic	F1 - F4
Function	Reserved for internal use
Description	Reserved for future use, will return an error if used. Do not use as a user variable, flag or program label.

FC (Filter capture)

Compatible with Lexium MDrive products:
Open Loop (RS-422/485)
Closed Loop (RS-422/485)
Open Loop (Ethernet)
Closed Loop (Ethernet)

Mnemonic	FC
Function	Set filtering for the capture input.
Type	Variable
Description	This variable will set the digital filtering to be applied to Input 1 when configured as a Capture input (type = 12). The input must be stable for “time” amount of milliseconds before a change in state is detected.

Input 13 Filter Capture Settings

Range	Min Pulse	Cutoff Frequency
0	50 nS	10 MHz
1	150 nS	3.3 MHz
2	200 nS	2.5 MHz
3	300 nS	1.67 MHz
4	500 nS	1.0 MHz
5	900 nS	555 kHz
6	1.7 µS	294.1 kHz
7	3.3 µS	151 kHz
8	6.5 µS	76.9 kHz
9	12.9 µS	38.8 kHz

Syntax	FC=<0-9>
Range	0 to 9
Units	Numeric
Default	0 (min pulse 50 nS, cutoff of 10 MHz)
Usage	Program/Immediate, Read/Write
Code Example	`FC=2 'Set filtering for Input 13 min pulse to 200 'nS/CO 2.5 MHz`
Equivalents	***MODBUS/TCP:*** — ***EtherNet/IP:*** class: 0x67 instance: 1 attribute: 0x06
Related	I13, TC, S13

FD (Restore factory defaults)

Compatible with Lexium MDrive products:
Open Loop (RS-422/485)
Closed Loop (RS-422/485)
Open Loop (Ethernet)
Closed Loop (Ethernet)

Mnemonic	FD
Function	Restore factory default settings
Type	Instruction
Description	FD will clear all program memory and return the Lexium MCode compatible device to factory default settings. The response will be the sign on message. FD will clear programs and initialize parameters.
Syntax	FD <carriage return>
Usage	Immediate
Code Example	`FD 'Restore device to factory default state Response "Copyright 2001-2012 by Intelligent Motion Systems, Inc."`
Equivalents	***MODBUS/TCP:*** — ***EtherNet/IP:*** —
Notes	FD will generate an error 73 if the unit is in motion.
Related	CP, IP

FT (Internal use only)

Mnemonic	FT
Function	Factory use only
Description	Will return an error if used. Do not use for a user program label, variable or flag..

H (Hold program execution)

Compatible with Lexium MDrive products:
Open Loop (RS-422/485)
Closed Loop (RS-422/485)
Open Loop (Ethernet)
Closed Loop (Ethernet)

Mnemonic	H
Function	Hold program execution
Type	Instruction
Description	The hold instruction is used in a program to suspend program execution. If no parameter is specified the execution of the program will be suspended while motion is in progress. This will typically be used following a MA, MR, HI or HM instruction. A time in milliseconds may be placed as a parameter to the hold instruction, This will suspend program execution for the specified number of milliseconds.
Syntax	H <time>
Units	Milliseconds
Range	1 - 65000
Usage	Program/Immediate

Code Example	`MA 1000  ' Move Absolute 1000 steps`
	`H  'Suspend prog ex. until motion completes`
	`'(used after a move command)`
	
	`H 2000  'Suspend program execution for 2 seconds`

Equivalents	***MODBUS/TCP:*** — ***EtherNet/IP:*** —
Related	PG, E, EX

HC (Motor holding current)

Compatible with Lexium MDrive products:
Open Loop (RS-422/485)
Closed Loop (RS-422/485)
Open Loop (Ethernet)
Closed Loop (Ethernet)

Mnemonic	HC
Function	Motor holding current
Type	Variable
Description	This variable defines the motor holding current in percent.
Syntax	HC=<percent>
Units	Percent
Range	0 to 100
Default	5%
Usage	Program/Immediate, Read/Write
Code Example	`HC=25  'set holding current to 25%`
Equivalents	***MODBUS/TCP:*** 0x0029 ***EtherNet/IP:*** class: 0x66 instance: 1 attribute: 0x03
Related	RC, HT

HI (Home to index mark)

Compatible with Lexium MDrive products:
Closed Loop (RS-422/485)
Closed Loop (Ethernet)

Mnemonic	HI
Function	Home to index mark
Type	Instruction
Description	This instruction will find the the encoder index mark. There are four combinations for this command. (See Use below.) When HI is executed, the axis moves in the direction specified by the (S) at VM until it reaches the index mark. It then creeps off of the index in the direction specified by the sign of (C) at VI. Motion is stopped as soon as the index changes state. Note that Speed and Creep is set by the VM and VI commands. 1. Speed: Specifies the direction and speed that the axis will move until the switch is activated (VM). 2. Creep: Specifies the direction and speed that the axis will move off the switch until it becomes inactive again (VI).
Syntax	HI <type>
Range	1 - 4
Modes	1 Slew at VM in the minus direction and Creep at VI in the plus direction. 2 Slew at VM in the minus direction and Creep at VI in the minus direction. 3 Slew at VM in the plus direction and Creep at VI in the minus direction. 4 Slew at VM in the plus direction and Creep at VI in the plus direction.
Usage	Program/Immediate
Code Example	`HI 2 'Slew at VM minus direction and Creep at VI in the 'minus direction`
Equivalents	***MODBUS/TCP:*** 0x002A ***EtherNet/IP:*** class: 0x69 instance: 1 attribute: 0x04
Related	VM, VI, EE, I6, HM

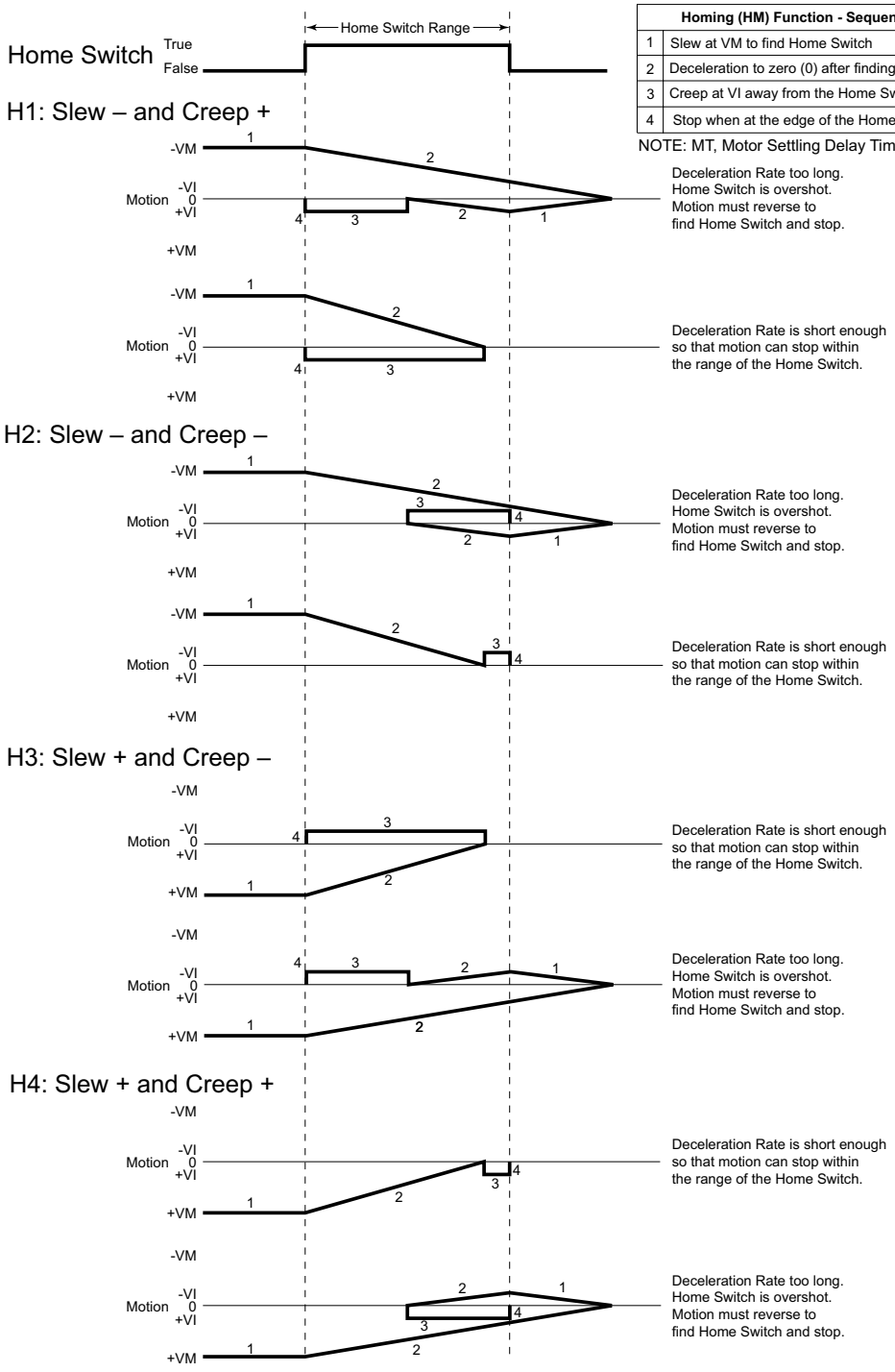
HM (Home to home switch)

Compatible with Lexium MDrive products:
Open Loop (RS-422/485)
Closed Loop (RS-422/485)
Open Loop (Ethernet)
Closed Loop (Ethernet)

Mnemonic	HM
Function	Home to home switch
Type	Instruction
Description	This instruction will find the selected I/O switch assigned to “Home”. 1. Speed: Specifies the direction and speed that the axis will move until the switch is activated (VM). 2. Creep: Specifies the direction and speed that the axis will move off the switch until it becomes inactive again (VI). When HM is executed, the axis moves at VM in the direction specified by the sign of speed. It then creeps off of the switch at VI in the direction specified by the sign of creep. Motion is stopped as soon as the switch becomes deactivated. Note that Speed and Creep is set by the VM and VI commands. The diagram on the following page illustrates the different scenarios possible during the Homing (HM) sequence. The diagrams represent the four HM commands. Below are the four combinations of the HM command.
Syntax	HM <type>
Range	1 - 4
Modes	1 Slew at VM in the minus direction and Creep at VI in the plus direction. 2 Slew at VM in the minus direction and Creep at VI in the minus direction. 3 Slew at VM in the plus direction and Creep at VI in the minus direction. 4 Slew at VM in the plus direction and Creep at VI in the plus direction.
Usage	Program/Immediate
Code Example	`HM 2 'Slew at VM minus direction and Creep at VI in the 'minus direction`
Equivalents	***MODBUS/TCP:*** 0x002B ***EtherNet/IP:*** class: 0x68 instance: 1 attribute: 0x02
Related	VM, VI, EE, I6, HI, LM, IS

The key to the diagrams is as follows.

1. Slew at VM to find the Index Mark.
2. Decelerate to zero (0) after finding the Index Mark.
3. Creep at VI away from the Index Mark.
4. Stop when at the edge of the Index Mark.



Homing (HM) Function - Sequence of Events	
1	Slew at VM to find Home Switch
2	Deceleration to zero (0) after finding the Home Switch
3	Creep at VI away from the Home Switch
4	Stop when at the edge of the Home Switch

NOTE: MT, Motor Settling Delay Time is still in effect

Figure 5.1 Homing functions sequence of events

HT (Hold current delay time)

Compatible with Lexium MDrive products:
Open Loop (RS-422/485)
Closed Loop (RS-422/485)
Open Loop (Ethernet)
Closed Loop (Ethernet)

Mnemonic	HT
Function	Holding current delay time
Type	Variable
Description	The HT variable sets the delay time in milliseconds between the MV=0 and when the device shifts to the holding current level specified by the HC (Motor Holding Current) variable. The delay time is also effected by the MT (Motor Settling Delay Time) variable in that the total time to current change is represented by the sum of MT + HT. The total of MT+HT cannot add up to more than 65535, thus the value of MT is included in the HT range. Thus the Maximum setting for HT=(65535-MT). ***If HT=0, the current will not reduce.***
Syntax	HT=<time>
Units	Milliseconds
Range	0 (no current reduction) or 1 - 65535
Default	500 ms
Usage	Program/Immediate. Read/Write
Code Example	`HT=1500  'set hold current delay time to 1.5 seconds`
Equivalents	***MODBUS/TCP:*** 0x002C ***EtherNet/IP:*** class: 0x66 instance: 1 attribute: 0x04
Related	HC, MT, RC

I1, I2, I3 and I4 (Read inputs 1, 2 ,3 or 4)

Compatible with Lexium MDrive products:
Open Loop (RS-422/485)
Closed Loop (RS-422/485)
Open Loop (Ethernet)
Closed Loop (Ethernet)

NOTE: On LMDxM42x or LMDxE42x (NEMA 17) Input 1 is not present. Use of this command will return an Error 37: Command, variable or flag not available.

Mnemonic	I1 - I4
Function	Read the state of inputs 1 - 4
Type	Variable
Description	This variable will read the state of the specified input 1 - 4. Can be used with PR (Print), BR (Branch) and CL (Call Subroutine) instructions. Can also be used with R1 - R4 and User Variables. The value of the bit state will be dependant on active (low/high) state of the input.
Syntax	PR I<1-4> BR <addr/lbl>, I<1-4>=<1/0> CL<addr/lbl>, I<1-4>=<1/0>
Units	Logic 0 or 1
Range	0/1
Usage	Program/Immediate. Read

Code Example	`PR I2  'Prints the logic state of input 2`
	`BR 128,I3=1  'Cond branch to address 128, Input 3 ACTIVE`
	`CL K9, I4=0  'Call subroutine K9, Input 4 INACTIVE`

Equivalents	***MODBUS/TCP:*** 0x002D, 0x002E, 0x002F, 0x0030 ***EtherNet/IP:*** —
Notes	Lexium MDrive models in the NEMA 17 frame size only have 3 inputs, thus I4 is invalid.
Related	IL, IN, O1-O4, IS

I5 (Read analog input)

Compatible with Lexium MDrive products:
Open Loop (RS-422/485)
Closed Loop (RS-422/485)
Open Loop (Ethernet)
Closed Loop (Ethernet)

Mnemonic	I5
Function	Read the value of the analog input
Type	Variable
Description	This variable will read the value of the correlating bit value seen on the Analog Input. Can be used with PR (Print), BR (Branch) and CL (Call Subroutine) instructions. The value read will between 0 and 4096. This value represents a voltage or current being seen on the analog input. For example, if in current mode (0 - 20mA range) 4096 would represent 20 mA, 2048 would represent 10 mA.
Syntax	PR I5 BR <addr/lbl>,I5=<0-4096> CL<addr/lbl>, I5<0-4096>
Units	Numeric value
Range	0 to 4096
Usage	Program/Immediate. Read

Code Example	`PR I5 'Print the value of I5`
	`BR G3,I5>512 'Branch to prog G3 if I5 is > 512`
	`CL 423,I5<220 'Call sub at address 423 if I5 is < 220`

Equivalents	***MODBUS/TCP:*** 0x0031 ***EtherNet/IP:*** class: 0x67 instance: 1 attribute: 0x08
Related	IS

I6 (Read encoder index mark)

Compatible with Lexium MDrive products:
Open Loop (RS-422/485)
Closed Loop (RS-422/485)
Open Loop (Ethernet)
Closed Loop (Ethernet)

Mnemonic	I6
Function	Read the state of the encoder index mark
Type	Variable
Description	This variable will read the on/off state of the Encoder Index Mark. Can be used with PR (Print), BR (Branch) and CL (Call Subroutine) instructions. The value read will be 0 (off mark) or 1 (on mark).
Syntax	PR I6 BR <addr/lbl>,I6=<0/1> CL<addr/lbl>, I6<0/1>
Units	Logic state 0 or 1
Range	0/1
Usage	Program/Immediate. Read

Code Example	`PR I6 'Print state of the encoder index mark`
	`BR 324,I6=1 'Branch to address 324 if I6 is ACTIVE`
	`CL K3,I6=1 'Call subroutine K3 if I6 is ACTIVE`

Equivalents	***MODBUS/TCP:*** 0x0037 ***EtherNet/IP:*** class: 0x67 instance: 1 attribute: 0x09
Related	IS

I7, I8 (Reserved)

Mnemonic	I7, I8
Function	Reserved, do not use as a user variable, flag or label.

I13 (Reserved)

Mnemonic	I13
Function	Reserved, do not use as a user variable, flag or label.

IC (Increment variable)

Compatible with Lexium MDrive products:
Open Loop (RS-422/485)
Closed Loop (RS-422/485)
Open Loop (Ethernet)
Closed Loop (Ethernet)

Mnemonic	IC
Function	Increment variable
Type	Instruction
Description	The IC instruction will increment the specified variable by one.
Syntax	IC <variable>
Usage	Program/Immediate, Read/Write
Code Example	`IC R1  'Increment register r1` `IC K5  'Increment user variable K5`
Equivalents	***MODBUS/TCP:*** — ***EtherNet/IP:*** —
Related	DC

IF (Input variable pending)

Compatible with Lexium MDrive products:
Open Loop (RS-422/485)
Closed Loop (RS-422/485)
Open Loop (Ethernet)
Closed Loop (Ethernet)

Mnemonic	IF	
Function	Input variable pending	
Type	Flag	
Description	The IF instruction is automatically set to 1 when IV command is executed. The IF flag reflects an input value from serial port is pending, not that one has been received. IF will be cleared to zero (0) with a carriage return or can be reset manually.	
Syntax	<user input><CR> IF=<0/1>	
States	0	No variable input pending
	1	Variable input pending
Default	0	
Usage	Program/Immediate, Read	
Code Example	`No Usage Example, Flag set automatically by IV`	
Equivalents	***MODBUS/TCP:*** — ***EtherNet/IP:*** —	

IN (Read inputs as a group)

Compatible with Lexium MDrive products:
Open Loop (RS-422/485)
Closed Loop (RS-422/485)
Open Loop (Ethernet)
Closed Loop (Ethernet)

Mnemonic	IN
Function	Read the state of inputs as one value
Type	Variable
Description	This keyword will read the binary state of the inputs and print them as a decimal value. When used thus, Input 1 is the Least Significant Bit (LSb) and Input 4 is the Most Significant Bit (MSb). It may be used in conjunction with PR (Print), BR (Branch) and CL (Call Subroutine) instructions. The value is a function of the actual state of the IO where 1 = +V and 0 = Ground. (Not a function of the active state defined in IS variable). The debounce function has no effect on the data read.
Syntax	PR IN BR <addr/lbl>,IN=<0-15> CL <addr/lbl>, IN<0-15>
Units	Decimal
Range	0 —15
Usage	Program/Immediate. Read

Code Example	`PR IN  'Print value of IO4-IO1`
	`BR 324,IN=07  'Branch to address 324 if IN=07`
	`CL K3,IN=13  'Call subroutine K3 if IN=13`

Equivalents	***MODBUS/TCP:*** 0x003B ***EtherNet/IP:*** class: 0x67 instance: 1 attribute: 0x0B
Notes	Lexium MDrive NEM 17 products have only three inputs. In these products input 3 will be the MSb and the total range is IN=<0-7>
Related	OT, IS, OS

IP (Initialize parameters)

Compatible with Lexium MDrive products:
Open Loop (RS-422/485)
Closed Loop (RS-422/485)
Open Loop (Ethernet)
Closed Loop (Ethernet)

Mnemonic	IP
Function	Initialize parameters
Type	Instruction
Description	The IP instruction will return all of the device variable and flag parameters to their stored values. If IP is used while the motor is moving an Error 74: Tried to Initialize Parameters of Clear Program while moving will be issued.
Syntax	IP
Usage	Program/Immediate
Code Example	`IP  'initialize parameters`
Equivalents	***MODBUS/TCP:*** — ***EtherNet/IP:*** —
Related	S, FD

IS (Setup inputs 1 - 4, 5 [analog] amd 6 [encoder index])

Compatible with Lexium MDrive products:
Open Loop (RS-422/485)
Closed Loop (RS-422/485)
Open Loop (Ethernet)
Closed Loop (Ethernet)

NOTE: The Inputs may be set up as sinking or sourcing. The sink or source condition of the inputs is defined by the bias of the input reference hardware input. See The Lexium MDrive Motion Control Hardware Manual, Section 6: Installation for circuit examples.

NOTE: On LMDxM42x or LMDxE42x (NEMA 17) Input 1 is not present. Use of Is=1,x,x command will return an Error 10: Illegal I/O number.

Mnemonic	IS
Function	Set up Input
Type	Instruction
Description	This instruction is used to setup the input line type and active state. There are three parameters attached to this instruction: 1. The line specifies the input line being configured 1 - 4 for logic inputs, 5 for analog, and 6 for encoder index. 2. The second parameter sets the type, or input function, as shown in the table below, the default is 0, or general purpose 3. The third parameter specifies whether the point will be active when high or active when low.
Syntax	IS=<line number>, <type>,<0(low active)/1(high active)>
Usage	Program/Immediate, Read

Code Example	`IS=1,1,1  'Set in 1 as homing active high`
	`IS=5,9,0  'Set analog to voltage mode, 0-5V`
	`IS=6,0,1  'set encoder index to be active high`

Equivalents	***MODBUS/TCP:*** TBD ***EtherNet/IP:*** TBD
Related	IN, I1-6, D1-4, OS, OT, O1-3

Logic input functions

These functions will apply to the 0 to 24 VDC inputs 1 — 4 (1 — 3 on NEMA 17 size Lexium MDrives).

The syntax for configuring them is:

Is = <1-4>, <type>, <1/0>

Input Type Table	
0	General purpose input
1	Homing function, will function as specified by the homing command (HM).
2	Limit +, will function as specified by the limit command (LM).
3	Limit —, will function as specified by the limit command (LM).
4	G0 input, will run program at address 1 upon activation.
5	Soft stop, stops motion with deceleration and halts program execution. If program is paused (PS), input is ignored.
6	Pause, pause/resume program with motion
7	Jog +, Will Jog motor in the positive direction at Max. Velocity (VM). The Jog Enable (JE) Flag must be set for this to function.
8	Jog —, Will Jog motor in thenegative direction at max. velocity (VM). The jog enable (JE) flag must be set for this to function.
11	Reset, When set as RESET input, then the action is equivalent to a ^C entered into a terminal. Note: If setting the input to sourcing, active true, ground the input first or a reset will occur.
12	Capture input will operate with the Trip Capture (TC) trip to run a subroutine when active ***NOTE: THIS FUNCTION APPLIES TO INPUT 1 ONLY!***

Analog input functions

These functions will apply to the analog input.

The syntax for configuring it is:

ls = 5, <mode>, <level>

Analog input mode table	
9	Voltage mode
10	Current mode
Analog level table	
0	0 to +5V (voltage mode), 0 to 20 mA (current mode)
1	0 to 10V (voltage mode), 4 to 20 mA (current mode)

Encoder index

This applies strictly to the encoder index mark. The only user setting is the active state.

The syntax for configuring it is:

ls = 6, 0, <1/0>

IT (Read internal temperature)

Compatible with Lexium MDrive products:
Open Loop (RS-422/485)
Closed Loop (RS-422/485)
Open Loop (Ethernet)
Closed Loop (Ethernet)

Mnemonic	IT
Function	Read the internal temperature of the device
Type	Variable
Description	Will read the internal temperature of the driver module of the Lexium MDrive in two locations: 1. The output bridge 2. The circuit board near the microprocessor When polled it will return both readings, with the bridge temperature first.
Syntax	PR IT
Units	Degrees
Range	-20°C to 100°C
Usage	Program/Immediate. Read

Code Example	`PR IT  'Print the internal temperature to the terminal`
	`  'screen. two values will return`
	`> 67, 58`

Equivalents	***MODBUS/TCP:*** — ***EtherNet/IP:*** class: 0x65 instance: 1 attribute: 0x06
Related	WT

IV (Input to variable)

Compatible with Lexium MDrive products:
Open Loop (RS-422/485)
Closed Loop (RS-422/485)
Open Loop (Ethernet)
Closed Loop (Ethernet)

Mnemonic	IV
Function	Input data into variable
Type	Instruction
Description	With the IV command, a user may input new variable values. These values must be numeric and will be input into the variable specified in the IV command. The variable used for the IV may be a system or USER Variable. A USER Variable must be declared prior to the IV command. When waiting for user input, there must be a conditional program loop based upon the state of IF (Input Pending) until the variable is input by the user.
Syntax	IV <user variable/R1-R4>
Usage	Program/Immediate

Code Example	`IV R1  'Input data into R1`
	`VA K5  'Create user variable K5`
	`IV K5  'Input data into user variable K5`
	
	`IV R1  'input value into register 1`
	`  LB k1  'label program loop k1`
	`  BR k1,If=1 'branch to k1 awaiting user input`

Equivalents	***MODBUS/TCP:*** — ***EtherNet/IP:*** —
Related	IF

JE (Jog enable)

Compatible with Lexium MDrive products:
Open Loop (RS-422/485)
Closed Loop (RS-422/485)
Open Loop (Ethernet)
Closed Loop (Ethernet)

Mnemonic	JE
Function	Enable/disable jog functions
Type	Flag
Description	This command will enable Jog Mode if I/O are set for Jog Plus and/or Jog Minus.
Syntax	JE=<0/1>
States	0 Jog functions disabled 1 Jog functions enabled
Default	0
Usage	Program/Immediate, Read/Write
Code Example	`JE=0  'Disable Jog mode` `JE=1  'Enable Jog mode`
Equivalents	***MODBUS/TCP:*** 0x003F ***EtherNet/IP:*** class: 0x66 instance: 1 attribute: 0x05

L (List program space)

Compatible with Lexium MDrive products:
Open Loop (RS-422/485)
Closed Loop (RS-422/485)
Open Loop (Ethernet)
Closed Loop (Ethernet)

Mnemonic	L
Function	List contents of program space
Type	Instruction
Description	The L instruction will print the contents of program space beginning at the specified address to the end. If no address is specified it will list beginning at address 1.
Syntax	L <address/label>
Usage	Immediate
Code Example	`L  'List program space contents from address 1` `L G5  'List program space contents from label G5`
Equivalents	***MODBUS/TCP:*** — ***EtherNet/IP:*** —
Related	CP, FD

LB (Label program or subroutine)

Compatible with Lexium MDrive products:
Open Loop (RS-422/485)
Closed Loop (RS-422/485)
Open Loop (Ethernet)
Closed Loop (Ethernet)

Mnemonic	LB
Function	Label program or subroutine
Type	Instruction
Description	The LB, or Label Instruction, allows the user to assign a 2 character name to a program, (BR) branch process or (CL) call subroutine. There is a limit of 192 labels. The restrictions for this command are: 1. A label cannot be named after an Lexium MCode Instruction, Variable or Flag or Keyword. 2. The first character must be alpha, the second character may be alpha-numeric. 3. A label is limited to two characters. 4. A program labeled SU will run on power-up 5. Labels ARE NOT case sensitive.
Syntax	LB <label>
Usage	Program

Code Example	`PG 100  'Start Program at address 100`
	`LB G1  'Name program G1`
	
	`PG 1`
	`LB SU  'Label Program to execute on power up.`

Equivalents	***MODBUS/TCP:*** — ***EtherNet/IP:*** —
Related	BR, CL, EX, TI, TP, L, CP

LD (Lead limits)

Compatible with Lexium MDrive products:
Closed Loop (RS-422/485)
Closed Loop (Ethernet)

Mnemonic	LD
Function	Set rotor lead limit
Type	Variable
Description	LD will set the lead limit in motor steps.
Syntax	LD=<steps>
Units	steps
Range	0 — 2147483647
Default	102400
Usage	Program/Immediate, Read/Write
Code Example	`LD=52100 'set lead limit to 52100`
Equivalents	***MODBUS/TCP:*** 0x0095 ***EtherNet/IP:*** class: 0x6A instance: 1 attribute: 0x09
Notes	When the lead limit is reached, error 106, lead limit reached will be active
Related	LG, LL

LG (Lag limits)

Compatible with Lexium MDrive products:
Closed Loop (RS-422/485)
Closed Loop (Ethernet)

Mnemonic	LG
Function	Set rotor lag limit
Type	Variable
Description	LG will set the lag limit in motor steps.
Syntax	LG=<steps>
Units	steps
Range	0 — 2147483647
Default	102400
Usage	Program/Immediate, Read/Write
Code Example	`LG=52100 'set lag limit to 52100`
Equivalents	***MODBUS/TCP:*** 0x0097 ***EtherNet/IP:*** class: 0x6A instance: 1 attribute: 0x0B
Notes	When the lag limit is reached, error 107, lag limit reached will be active
Related	LD, LL

LL (Position lead/lag)

Compatible with Lexium MDrive products:
Closed Loop (RS-422/485)
Closed Loop (Ethernet)

Mnemonic	LL
Function	Read rotor lead/lag
Type	Variable
Description	Represent the number of counts that the rotor leads or lags the stator. A positive value indicates position lag. A negative value indicates position lead
Syntax	PR LL CL <address/label>, LL <condition><steps> BR <address/label>, LL <condition><steps>
Units	steps
Range	-2147483647 to +2147483647
Usage	Program/Immediate, Read only
Code Example	`PR LL  'read rotor lead/lag` `CL k5, LL>102500 'Call subroutine k5 if 'LL is greater than '102500 steps`
Equivalents	***MODBUS/TCP:*** 0x0099 ***EtherNet/IP:*** class: 0x6A instance: 1 attribute: 0x0A
Related	LD, LG

LK (Lock user program)

Compatible with Lexium MDrive products:
Open Loop (RS-422/485)
Closed Loop (RS-422/485)
Open Loop (Ethernet)
Closed Loop (Ethernet)

Mnemonic	LK	
Function	Lock access to user program space	
Type	Flag	
Description	This flag allows the user to lock the program from being listed or modified. It can only be reset by clearing the entire program space: CP (no address). If CP (address,/label), L (address/label) or PG (address/label) are entered, then error 44 (Program Locked) will be set and nothing else will happen. To clear LK, don't save (S) then do a Ctrl-C or Cycle Power and the LK will be reset to previous unlocked state. (Program is automatically stored in NVM as it is entered.) Or you may clear program (CP or FD to list or reprogram the device.). This will clear the program and reset LK to 0	
Syntax	LK=<0/1>	
States	0	User programs may be modified or listed
	1	User programs may not be modified or listed
Default	0 (disabled)	
Usage	Immediate, Read/Write	
Code Example	`LK=1  'Lock programs from being listed or changed`	
Equivalents	***MODBUS/TCP:*** — ***EtherNet/IP:*** —	

LM (Limit stop modes)

Compatible with Lexium MDrive products:
Open Loop (RS-422/485)
Closed Loop (RS-422/485)
Open Loop (Ethernet)
Closed Loop (Ethernet)

Mnemonic	LM	
Function	Limit stop modes	
Type	Variable	
Description	The LM variable specifies the Limit Stop Mode for the Lexium MCode compatible device. There are six LM modes.	
Syntax	LM=<mode>	
Range	1 - 6	
Modes	1	Normal Limit function with a decel ramp. The I/O must be set for Limits (IS command). If the limit switch in the direction of travel is reached, the motion will decelerate to a stop. That is, the plus limit works only in the plus direction of travel and the minus limit works only in the minus direction of travel. In Figure 5.2, the Limit is activated at a given position but because of the deceleration rate the motion continues for the duration of the deceleration time. This position may be beyond the trip point of the limit and a subsequent move in the same direction will not stop. A crash may be imminent. If the limit is activated and maintained the software will allow motion only in the opposite direction. If Homing (HM) is active and a limit is reached, the motion will decel to a stop and then reverse direction and seek the Homing Switch. If the Homing Switch is not activated on the reverse and the opposite limit is reached all motion will stop with a decel ramp. (See HM)
	2	A Limit stops all motion with a deceleration ramp including Homing.
	3	A Limit will stop all motion with a deceleration ramp and stop program execution.
	4	Functions as LM=1 but with no deceleration ramp
	5	Functions as LM=2 but with no deceleration ramp.
	6	Functions as LM=3 but with no deceleration ramp.
Usage	Program/Immediate, Read/Write	
Code Example	`LM=2 'Set Limit stop with a decel ramp, no homing.`	
Equivalents	***MODBUS/TCP:*** 0x0042 ***EtherNet/IP:*** class: 0x66 instance: 1 attribute: 0x06	
Related	H,I, HM, JE, MA, MR, SL	

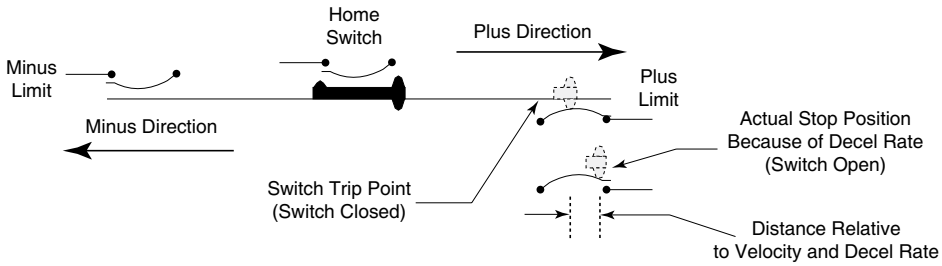


Figure 5.2 Limit modes of operation

LR (Locked rotor)

Compatible with Lexium MDrive products:
Closed Loop (RS-422/485)
Closed Loop (Ethernet)

A locked rotor is defined as no rotor movement while at the maximum allowed lag for a specified period of time. When lag becomes equal to the bounds, a timer starts to count down. Upon reaching zero, a locked rotor will be indicated by the assertion of a status flag. The timer reloads on any encoder movement. The timer timeout period is user selectable from 2mS to 65.5 seconds.

When HMT is configured AS=1 or 2, a locked rotor will also cause an internal fault (LR) disabling the output bridge. The bridges may be re-enabled by cycling power, or via software command CF: Clear Locked Rotor Fault. A locked rotor condition will assert an error 104 as well.

In torque mode, a locked rotor does not disable the bridges. The locked rotor flag (LR) can be used to indicate the rotor has been stopped at the specified torque for a preset amount of time.

Mnemonic	LR
Function	Locked rotor
Type	Read only status flag
Description	Indicates the state of the rotor as locked (1) or unlocked (0).
Syntax	PR LR
Range	0/1
Usage	Program/Immediate, Read only
Code Example	`PR LR  'read rotor lock status` `TA 2,k3 'trip to sub k3 on locked rotor`
Equivalents	***MODBUS/TCP:*** 0x009B ***EtherNet/IP:*** class: 0x6A instance: 1 attribute: 0x0C
Related	CF, LT, TA

LT (Locked rotor timeout)

Compatible with Lexium MDrive products:
Closed Loop (RS-422/485)
Closed Loop (Ethernet)

Mnemonic	LT
Function	Set locked rotor timeout
Type	Variable
Description	The LT variable will set the locked rotor timeout in milliseconds. This setting will determine the time in milliseconds that the output bridge will disable following the locked rotor (LR) flag being active.
Syntax	LT=<time>
Units	Milliseconds
Range	2 — 65535
Default	2000 mS
Usage	Program/Immediate, Read/Write
Code Example	`LT=1000  'set locked rotor time to 1S`
Equivalents	***MODBUS/TCP:*** 0x009C ***EtherNet/IP:*** class: 0x6A instance: 1 attribute: 0x0D
Notes	The output bridge will not disable if the Hybrid is in torque mode (AS=3)
Related	AS, CB, LR

MA (Move to an absolute position)

Compatible with Lexium MDrive products:
Open Loop (RS-422/485)
Closed Loop (RS-422/485)
Open Loop (Ethernet)
Closed Loop (Ethernet)

Mnemonic	MA
Function	Move to an absolute position
Type	Instruction
Description	Set mode for absolute move and move to an absolute position relative to (0) zero. MD (Motion Mode) will be set to MA. The time required to calculate the move is 20 µSec. MA command will not operate during a homing sequence. An in progress MA can be stopped with and "esc" entry or an SL 0 command.
Syntax	MA <±position>,<parameter 2>, <parameter 3>
Parameter 2	0 (or blank) no action taken 1 Device name will be sent out of the serial port on move completion.
Parameter 3	0 (or blank) no action taken. 1 Motor will continue moving after reaching position.
Usage	Program/Immediate

Code Example	`MA 200000 'Move to absolute position 200000`
	`MA 100000,1 'Move to abs. pos. 100000, send DN when 'complete`
	`MA 512000,0,1 'Move to abs. pos. 512000, continue 'motion after position is reached.`

Equivalents	***MODBUS/TCP:*** 0x0043 ***EtherNet/IP:*** class: 0x66 instance: 1 attribute: 0x07
Related	MD, MR, MS, P, SL

MD (Motion mode)

Compatible with Lexium MDrive products:
Open Loop (RS-422/485)
Closed Loop (RS-422/485)
Open Loop (Ethernet)
Closed Loop (Ethernet)

Mnemonic	MD
Function	Motion mode
Type	Variable
Description	Indicates what the last motion command was. When just a number is entered, then it will execute the move type according to the previous move type entered. This allows the user to apply numeric data to the last motion command without having to enter the command itself.
Syntax	PR MD
Units	Motor steps/Encoder counts
Usage	Program/Immediate. Read

Code Example	`PR MD 'Return the last motion command used to the 'terminal screen. Response will be the last 'motion command ie. MR`
	
	`MR 10000 'Move Relative 10000 steps`
	`5000 'Motor will move relative 5000 steps (if NE=1)`

Equivalents	***MODBUS/TCP:*** — ***EtherNet/IP:*** —
Notes	Note that if the IF flag is pending, numeric entry will be applied to the IV (Input Variable).
Related	MA, MR, MS, P, PR, SL

MF (Make up speed)

Compatible with Lexium MDrive products:
Closed Loop (RS-422/485)
Closed Loop (Ethernet)

Mnemonic	MF
Function	Set make up speed in steps/sec
Type	Variable
Description	Defines the frequency at which missed steps are re-inserted into the move profile.
Syntax	MF=<steps/sec>
Units	Steps per second
Range	92 to VM
Default	768000
Usage	Program/Immediate, Read/Write
Code Example	`MF=256000  'set make up steps to 256000`
Equivalents	***MODBUS/TCP:*** 0x009E ***EtherNet/IP:*** class: 0x6A instance: 1 attribute: 0x0E
Notes	MF will be used for make up if MU=1
Related	MU, VM

MP (Moving to position)

Compatible with Lexium MDrive products:
Open Loop (RS-422/485)
Closed Loop (RS-422/485)
Open Loop (Ethernet)
Closed Loop (Ethernet)

Mnemonic	MP
Function	Moving to position
Type	Flag
Description	This flag will at a 1 state when the axis is moving to a position following a MA or MR instruction until MT expires.
Syntax	PR MP
Range	0/1
Status Response	0 Not moving
	1 Moving to a position
Usage	Program/Immediate, Read
Code Example	`PR MP  'read the state of the MP flag to terminal`
Equivalents	***MODBUS/TCP:*** 0x0045 ***EtherNet/IP:*** —
Related	MT

MR (Move to a relative position)

Compatible with Lexium MDrive products:
Open Loop (RS-422/485)
Closed Loop (RS-422/485)
Open Loop (Ethernet)
Closed Loop (Ethernet)

Mnemonic	MR
Function	Move to a relative position
Type	Instruction
Description	Set mode for relative move and move a relative distance. MD (Motion Mode) will be set to MR. The time required to calculate the move is 20 μSec. MR command will not operate during a homing sequence. An in progress MR can be stopped with and "esc" entry or an SL 0 command.
Syntax	MR <±position>,<parameter 2>, <parameter 3>
Parameter 2	0 (or blank) no action taken 1 Device name will be sent out of the serial port on move completion. ***(RS-422/485 models only)***
Parameter 3	0 (or blank) no action taken. 1 Motor will continue moving after reaching position.
Usage	Program/Immediate

Code Example	`MR 200000 'Move to relative position 200000`
	`MR 100000,1 'Move to rel. pos. 100000, send DN when 'complete`
	`MR 512000,0,1 'Move to rel. pos. 512000, continue 'motion after position is reached.`

Equivalents	***MODBUS/TCP:*** 0x0046 ***EtherNet/IP:*** class: 0x66 instance: 1 attribute: 0x08
Related	MD, MR, MS, P, SL

MS (Microstep resolution select)

Compatible with Lexium MDrive products:
Open Loop (RS-422/485)
Closed Loop (RS-422/485)
Open Loop (Ethernet)
Closed Loop (Ethernet)

Mnemonic	MS
Function	Microstep resolution select
Type	Variable
Description	The MS variable controls the microstep resolution of the Lexium MCode compatible device. There are 20 different microstep resolutions that can be used with the device. The table below illustrates the parameter settings and their associated resolutions for the 1.8° stepping motor used with the Lexium MCode compatible device. The MS parameters given in the table below are the only valid parameters that will be accepted by the device. Note that the setting of MS will impact the maximum value of VM (Max Velocity) to MS*10000.
Syntax	MS=<parameter>
Units	Microsteps per full motor step
Default	256
Parameters	

Binary resolution parameters

per step	1	2	4	8*	16	32	64	128	256
per rev,	200	400	800	1600	3200	6400	12800	25600	51200

Decimal resolution parameters

per step	5*	10	25	50	100	125	200	250
per rev,	1000	2000	5000	10000	20000	25000	40000	50000

Additional resolution parameters

per step	108	127	180
per rev,	21600 (1 Arc Minute/µStep)	25400 (0.001mm/µStep)	36000 (0.01°/µStep)

*Do not use with hMT active

Usage	Program/Immediate. Read/Write
Code Example	MS=5 'microstep resolution to 1000 steps/rev
Equivalents	***MODBUS/TCP:*** 0x0048 ***EtherNet/IP:*** class: 0x66 instance: 1 attribute: 0x09
Related	MA, MR, MS, P, PR, SL, C1

MT (Motor settling delay time)

Compatible with Lexium MDrive products:
Open Loop (RS-422/485)
Closed Loop (RS-422/485)
Open Loop (Ethernet)
Closed Loop (Ethernet)

Mnemonic	MT
Function	Motor settling delay time
Type	Variable
Description	Specifies the motor settling delay time in milliseconds. MT allows the motor to settle following a move. This is the time between moves if consecutive motions are executed. the MV flag will be active during this time.
Syntax	PR MT
Units	Milliseconds
Range	0 to 65000
Default	0
Usage	Program/Immediate. Read/Write
Code Example	`MT=50  'Set motor settling delay time to 50 milliseconds`
Equivalents	***MODBUS/TCP:*** 0x0049 ***EtherNet/IP:*** class: 0x66 instance: 1 attribute: 0x0A
Notes	MT is added into HT (Hold Current Delay Time). The total of the two cannot exceed 65535. Thus the maximum setting for MT=(65535-HT). MT should be at least 50 mS when Encoder function is enabled (EE=1)
Related	HC, HT, RC

MU (Make up mode)

Compatible with Lexium MDrive products:
Closed Loop (RS-422/485)
Closed Loop (Ethernet)

Make-up mode is active when ever hMTechnology is on in fixed or variable current mode (AS=1/AS=2). MU will compensate for position errors due to a disturbance during a move by reinserting missed steps into a motion profile as conditions allow. The MU mode selected defines how that compensation occurs.

MU=0: Position maintenance happens without regard to time. In this mode missed steps are added to the motion profile to end the move at the commanded position. The speed at which the error compensation occurs is determined by the point in which the disturbance leading to the error occurs.

Should the disturbance occur at during acceleration or at velocity, steps are added at the set maximum velocity (VM). Should the disturbance occur during deceleration, the axis will creep into position at the set initial velocity (VI).

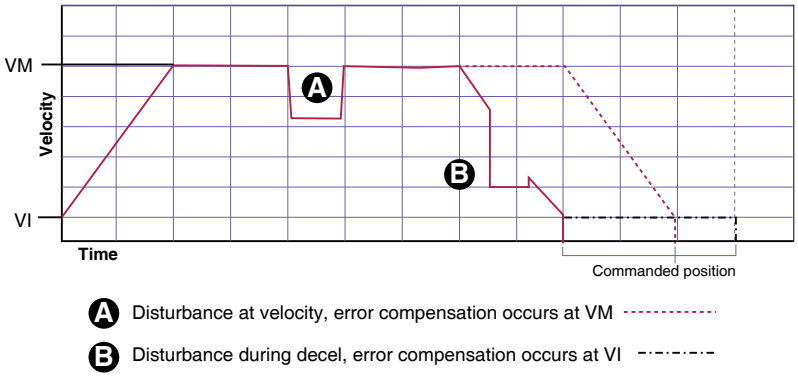


Figure 5.3: Make up mode MU=0

MU=1: Position maintenance occurs as the load allows with regard to the timing of the move. In this mode error compensation occurs by missed steps being inserted into the profile. The hMT algorithm will interleave steps into the move attempting to complete the motion profile on time. Missed steps are reinserted when the lead/lag relationship of the rotor and stator is <=1.1 motor full steps.

During make-up active in mode 1, the steps will be generated at a rate (frequency) that is a composite of the maximum velocity (VM) or commanded slew rate (SL) and the set make-up (MU) frequency. This frequency will be the greater of 2 X (VM or SL) or MU.

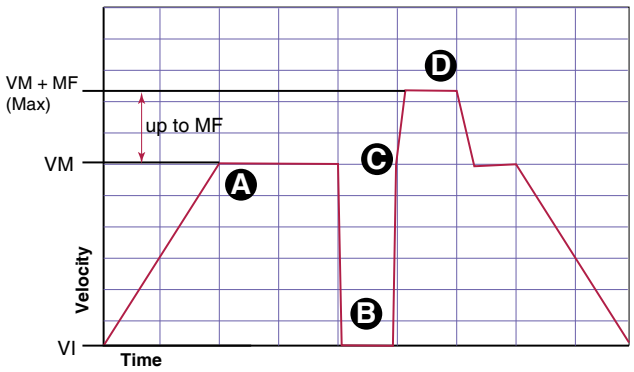


Figure 5.4: Make up mode MU=1

MU=2: In mode 2 error compensation will occur similar to mode 1 at the highest velocity the load will allow without regard to VM, but at a velocity not exceeding 2560000 steps/sec (3000 RPM).

Note that when the the mtor shaft is torqued out of position without any commanded motion, make up will occur at ≤ MF (MU=1) or at ≤ 3000 RPM (MU=2).

Acceleration during make-up

Make-up acceleration occurs at 16,763,806 steps/sec2. The VI setting for make-up is 916 steps/sec. These are fixed values that cannot be changed by the user.

Mnemonic	MU	
Function	Set make up mode	
Type	Variable	
Description	Determines the mode for position maintenance	
Syntax	MU=<mode>	
Range	0 — 2	
Modes	0	Make up position without regard to time
	1	Use make up frequency (MF) as make up frequency
	2	Use system speed , an internally defined velocity limited to 2560000 steps/sec (3000 RPM) as make up frequency
Default	0 (Make up position without regard to time)	
Usage	Program/Immediate, Read/Write	
Code Example	`MU=1  'use make-up frequency (MF)`	
Equivalents	***MODBUS/TCP:*** 0x00A0 ***EtherNet/IP:*** class: 0x6A instance: 1 attribute: 0x0F	
Notes	Position maintenance will only occur when motor lag/lead is within 1.1 motor steps. Make up steps may be interleaved with motion steps and made after a move has completed. Where position maintenance occurs is dependant on motor lag/lead, motion frequency and selected make up speed. Make up mode will be cleared when bridges are disabled and Hybrid is enabled.	
Related	MF	

MV (Moving)

Compatible with Lexium MDrive products:
Open Loop (RS-422/485)
Closed Loop (RS-422/485)
Open Loop (Ethernet)
Closed Loop (Ethernet)

Mnemonic	MV
Function	Axis in motion
Type	Flag
Description	Moving flag will be in a logic 1 state when a motion is occurring. This flag will set an output ACTIVE if the OS (Output setup command) is configured to output type 17.
Syntax	PR MV
Range	0/1
Status Response	0 Not moving
	1 Moving
Usage	Program/Immediate, Read
Code Example	`PR MV 'read the state of the MV flag to terminal`
Equivalents	***MODBUS/TCP:*** 0x004A ***EtherNet/IP:*** class: 0x66 instance: 1 attribute: 0x0B
Related	VC, OS

O1, O2 and O3 (Set outputs 1, 2 or 3)

Compatible with Lexium MDrive products:
Open Loop (RS-422/485)
Closed Loop (RS-422/485)
Open Loop (Ethernet)
Closed Loop (Ethernet)

NOTE: On LMDxM42x or LMDxE42x (NEMA 17) Outputs 1 and 2 are not present. Use of this command will return an Error 37: Command, variable or flag not available.

Mnemonic	O1 - O3
Function	Set output logic state
Type	Variable
Description	This variable will set the logic state of the specified output to 1 or 0. The voltage level will be dependant on the active (low/high) state of the output, specified by the OS variable.
Syntax	Syntax:O<1-4>=<0/1>
Units	Logic 0 or 1
Range	0/1
Usage	Program/Immediate. Write
Code Example	`O2=1 'Set output 2 ACTIVE`
Equivalents	***MODBUS/TCP:*** 0x004B, 0x004C, 0x004D ***EtherNet/IP:*** class: 0x67, instance: 1, attribute: 0x10, 0x11, 0x12
NOTES	The Lexium MDrive 42mm (NEMA 17) has only one output. THis output can be set to a trip function.
Related	OS, OT

OE (On error handler)

Compatible with Lexium MDrive products:
Open Loop (RS-422/485)
Closed Loop (RS-422/485)
Open Loop (Ethernet)
Closed Loop (Ethernet)

Mnemonic	OE
Function	On error handler
Type	Instruction
Description	When an error occurs, the specified subroutine is called. If a program was running when the fault occurs, once the error routine completes, program execution continues with the instruction after the one that caused the error. After OE, the command is executed. A program need not be running for the subroutine specified by OE to run. The ON ERROR function is disabled by setting the address parameter to 0 or resetting the device with an FD or CP OE Subroutine MUST have an RT at the end. OE will not execute during programming.
Usage	Program
Code Example	`OE K1 'run subroutine K1 on an error`
Equivalents	***MODBUS/TCP:*** — ***EtherNet/IP:*** —
Related	EF, ER

OF (Output Fault)

Compatible with Lexium MDrive NEMA 23 (57 mm) and NEMA 34 (85 mm) products:
Open Loop (RS-422/485)
Closed Loop (RS-422/485)
Open Loop (Ethernet)
Closed Loop (Ethernet)

Mnemonic	OF	
Function	Output fault	
Type	Variable	
Description	Read-only status variable indicates an over-current fault condition on the power outputs (Outputs 1 and 2). An Error code 1 or 2 will also be asserted	
Usage	PR OF	
Status Response	0	No fault
	1	output 1 fault
	2	output 2 fault
	3	both output 1 and 2 fault
Code Example	`PR OF  'print the output fault status`	
Equivalents	***MODBUS/TCP:*** 0x004E ***EtherNet/IP:*** class: 0x67, instance: 1, attribute: 0x13	
Related	Os	

OS (Setup outputs 1 — 3)

Compatible with Lexium MDrive products:
Open Loop (RS-422/485)
Closed Loop (RS-422/485)
Open Loop (Ethernet)
Closed Loop (Ethernet)

NOTE: On LMDxM42x or LMDxE42x (NEMA 17) Outputs 1 and 2 are not present. Use of this command will return an Error 10: Illegal I/O number.

Mnemonic	OS
Function	Setup output
Type	Instruction
Description	This instruction is used to setup the output line type and active state. There are three parameters attached to this instruction: 1. The line specifies the output line being configured 1 - 3. 2. The second parameter sets the type, or output function, as shown in the table below, the default is 16, or general purpose 3. The third parameter specifies whether the point will be active when high or active when low.
Syntax	OS=<line number>, <type>,<0(low active)/1(high active)>
Usage	Program/Immediate, Read
Code Example	`OS=1,16,1  'Set out 1 as gen. purpose active high` `OS=2,17,0  'Set out 2 as moving, active low`
Equivalents	***MODBUS/TCP:*** 0x41 / 0x42 ***EtherNet/IP:*** class: 0x67, instance: 1, attribute: 0x14
Related	IS, OT, O1-3

Logic output functions

These functions will apply to outputs 1 — 3 (3 on 42mm [NEMA 17] size Lexium MDrives).

The syntax for configuring them is:

Os = <1-3>, <type>, <1/0>

Output Type Table	
16	USER: General purpose user output.
17	MOVING: output will be active while the axis is in motion.
18	ERROR: will be active when a software error occurs.
19	STALL: Indicates a stall has been detected ***[Closed loop models only]***.
20	VCHANGE: velocity changing indicates the axis is accelerating or decelerating.
21	LOCKED ROTOR: indicates the rotor is in a locked state ***[Closed loop models only]***.
23	MOVINGPOS: indicates the axis is moving to a specified position.
24	HMT ACTIVE: indicates when HMT is active ***[Closed loop models only]***.
25	MAKE UP ACTIVE: indicates when HMT is making up steps *[Closed loop models only]*.
28	TRIP OUT, applies to output 3 only, active low only. Os=3,28,0
29	ATTENTION: indicates a status or statuses as configured by the AO variable.

OT (Set outputs 1— 3 as a group)

Compatible with Lexium MDrive products:
57 mm (NEMA23) and 85 mm (NEMA 34) only.
Open Loop (RS-422/485)
Closed Loop (RS-422/485)
Open Loop (Ethernet)
Closed Loop (Ethernet)

Mnemonic	OT
Function	Set the state of outputs 1 - 3 as one value
Type	Variable
Description	The OT variable allows the user to set outputs 1-3 as one 3 bit binary value. The value is entered in decimal, with a range of 0-7 in binary where Output 1 will be the LSb and Output 3 will be the MSb.
Syntax	OT=<0-7>
Units	Decimal
Range	0 — 7
Usage	Program/Immediate. Write
Code Example	`OT=5 'set the standard output group to 101`
Equivalents	***MODBUS/TCP:*** 0x0056 ***EtherNet/IP:*** class: 0x67 instance: 1 attribute: 0x15
Notes	NOTE: The 42mm (NEMA 17) model of the Lexium MDrive only has a single output, thus cannot be used for BCD functions.
Related	IN, IS, OS

P (Position counter)

Compatible with Lexium MDrive products:
Open Loop (RS-422/485)
Closed Loop (RS-422/485)
Open Loop (Ethernet)
Closed Loop (Ethernet)

Mnemonic	P
Function	Position counter
Type	Variable
Description	This instruction is used to set or print the value of the Lexium MCode compatible device position counter. The position will read in Motor Steps from C1 (Counter 1) by default, if encoder functions are enabled on closed loop models, the position counter will read in Encoder Counts from C2 (Counter 2). Modifying P in essence changes the frame of reference for the axis for Move Absolute (MA) instructions. P will probably be set once during system set up to reference or “home” for the system.
Syntax	P=<±position>, PR P
Units	Motor Steps (EE=0)/Encoder Counts (EE=1
Range	-2147483648 to +2147483647
Default	0
Usage	Program/Immediate. Read/Write
Code Example	`PR P 'sprint the position to the terminal window`
Equivalents	***MODBUS/TCP:*** 0x0057 ***EtherNet/IP:*** class: 0x68 instance: 1 attribute: 0x03
Related	C1, C2

PC (Position capture at trip)

Compatible with Lexium MDrive products:
Open Loop (RS-422/485)
Closed Loop (RS-422/485)
Open Loop (Ethernet)
Closed Loop (Ethernet)

Mnemonic	PC
Function	Position capture at trip
Type	Instruction
Description	Captures motor or encoder position during a trip event. Activation will occur upon any trip function EXCEPT a position trip (TP or TR). Will display in either motor steps (EE=0) or encoder counts (EE=1)
Syntax	PR PC
Usage	Immediate
Code Example	`PR PC    'Display captured position`
Equivalents	***MODBUS/TCP:*** 0x0059 ***EtherNet/IP:*** class: 0x68 instance: 1 attribute: 0x04
Related	TE, TI, TC, TT, IS

PG (Program mode)

Compatible with Lexium MDrive products:
Open Loop (RS-422/485)
Closed Loop (RS-422/485)
Open Loop (Ethernet)
Closed Loop (Ethernet)

Mnemonic	PG
Function	Enter/exit program mode
Type	Instruction
Description	This instruction toggles the device into or out of program mode.
Syntax	PG <address>
Usage	Immediate/Program

Code Example	`PG 100 'enter program at address 100`
	`...`
	`...  'program body`
	`...`
	`PG   'end program return to immediate mode`

Equivalents	***MODBUS/TCP:*** — ***EtherNet/IP:*** —
Related	CP

PM (Position maintenance enable)

Compatible with Lexium MDrive products:
Closed Loop (RS-422/485)
Closed Loop (Ethernet)

Mnemonic	PM
Function	Position maintenance enable
Type	Flag
Description	This flag will enable the position maintenance functions of an Lexium MCode compatible device with encoder. The position maintenance velocity will be at the setting for VI (Initial Velocity). If moved beyond the value of DB (DeadBand), unit will correct If SM = 0 and PM = 1, Position Maintenance will take place provided the position does not exceed the Stall Factor (SF). If SM = 1 and PM = 1, Position Maintenance will take place even if the Stall Factor (SF) is exceeded, unless VI is set too high causing the motor to stall.
Syntax	PM=<1/0>
Range	0/1
Default	0 (disabled)
Status Response	0 Disabled (Default) 1 Enabled
Usage	Program/Immediate, Read/Write
Code Example	`PM=1 'Enable position maintenance`
Equivalents	***MODBUS/TCP:*** 0x005C ***EtherNet/IP:*** class: 0x69 instance: 1 attribute: 0x07
Related	VI, EE, SM, DB, C2, SF

PN (Part number)

Compatible with Lexium MDrive products:
Open Loop (RS-422/485)
Closed Loop (RS-422/485)
Open Loop (Ethernet)
Closed Loop (Ethernet)

Mnemonic	PN
Function	Read device part number
Type	Variable
Syntax	PR PN
Usage	Immediate, Read
Code Example	`PR PN 'read the device part number`
Equivalents	***MODBUS/TCP:*** 0x2B/0xoE ***EtherNet/IP:*** class: 0x01 instance: 1 attribute: 0x03

PR (Print specified data or text)

Compatible with Lexium MDrive products:
Open Loop (RS-422/485)
Closed Loop (RS-422/485)
Open Loop (Ethernet)
Closed Loop (Ethernet)

Mnemonic	PR
Function	Print specified data or text
Type	Instruction
Description	This instruction is used to output text and parameter value(s) to the host PC. Text should be enclosed in quotation marks while parameters (variables and flags) should not. Text strings and parameters which are to be output by the same PR instruction should be separated by commas. The information being output is followed by a carriage return unless a semicolon (;) is included at the end of the PR instruction to indicate that the cursor should remain on the same line. It is important to note that the receive buffer for the Lexium MCode device is 64 characters, this includes the PR instruction itself, any spaces, text characters, etc. If the buffer length is exceeded a CR/LF will ocur and set error 20.
Syntax	PR <text>, <data>
Usage	Immediate
Code Example	`PR  "Position =", P  'print axis position` `PR MS  'Print the µStep Resolution setting`
Equivalents	***MODBUS/TCP:*** — ***EtherNet/IP:*** —
Notes	A delay time between the print requests to the device must be considered to allow the device time to interpret a command and answer the host before a subsequent command can be sent.

ASCII control codes

There are a number of ASCII control codes which can be used to enhace the performance of the Print command.

ASCII control codes	
(semicolon) ;	PR "data/text"; The semicolon will suppress CR/LF at the end of a line.
\b	Backspace
\c	CTRL-C
\e	ESC
\g	Bell/beep
\n	Line Feed
\r	Carriage Return
\t	Tab
Additionally most ASCII escape codes used with terminal emulators may be used.	

PS (Pause program execution)

Compatible with Lexium MDrive products:
Open Loop (RS-422/485)
Closed Loop (RS-422/485)
Open Loop (Ethernet)
Closed Loop (Ethernet)

Mnemonic	PS
Function	Pause program execution
Type	Instruction
Description	This instruction is used to pause an executing program and invoke normal deceleration of any motion being executed to Zero. Immediate mode instructions are allowed while a program is in a paused state. To resume the program the RS instruction is used.
Syntax	PS
Usage	Immediate
Code Example	`PS  'Pause running program`
Equivalents	***MODBUS/TCP:*** 0x41/0x42 ***EtherNet/IP:*** —
Related	RS, SL, MA, MR, HI, HM

PY (Party mode enable)

Compatible with Lexium MDrive products:
Open Loop (RS-422/485)
Closed Loop (RS-422/485)

Mnemonic	PY
Function	Party mode enable
Type	Flag
Description	The party flag must be set to 1 if the device is being used in a multidrop communication system. When Party Mode is enabled, each device in the system must be addressed by the host computer by using the device name specified by the DN instruction. This name will precede any command given to a specified unit in the system and be terminated with a Control J (CTRL + J). One CTRL + J must be issued after power up or entering the Party Mode to activate the Party Mode. By default the DN assigned at the factory is the exclamation character (!) . The global Drive Name is the asterisk character (*). Commands preceded by this character will be recognized by every Lexium MCode compatible device in the system. After the Party Mode is enabled, send CTRL + J (^J) to activate it. Type commands with Device Name (DN) and use CTRL + J as the Terminator.
Syntax	PY=<0/1>
Range	0/1
Default	0 (disabled)
Status	0 Disabled (Default) 1 Enabled
Usage	Program/Immediate, Read/Write
Code Example	`PY=1  'Enable Party Mode Communications`
Notes	Note: A delay time between the command requests to the device must be considered to allow the device time to interpret a command and answer the host before a subsequent command can be sent. The time between requests is dependent on the command and the corresponding response from the Device.
Related	DN, DG

QD (Queued)

Compatible with Lexium MDrive products:
Open Loop (RS-422/485)
Closed Loop (RS-422/485)

Mnemonic	QD
Function	Queued
Type	Flag
Description	Function is to queue drives on party lines. Works similar to uLynx QUED flag. If a drive or drives are Queued, then, when they see the address "^", they will respond to it. All other, non-queued drives will ignore the command
Syntax	QD=<0/1>
Range	0/1
Default	0 (disabled)
Status	0 Disabled (Default) 1 Enabled
Usage	Immediate, Read/Write
Code Example	`QD=1 'Queue Drive`
Equivalents	***MODBUS/TCP:*** — ***EtherNet/IP:*** —
Related	PY, DN

R1, R2, R3 and R4 (User registers)

Compatible with Lexium MDrive products:
Open Loop (RS-422/485)
Closed Loop (RS-422/485)
Open Loop (Ethernet)
Closed Loop (Ethernet)

Mnemonic	R1 - R4
Function	User registers
Type	Variable
Description	The Lexium MCode compatible device has four 32 bit user registers to contain numerical data. These registers may contain up to 11 digits including the sign and may be used to store and retrieve data to set variables, perform math functions, store and retrieve moves and set conditions for branches and call subroutine.
Syntax	R<1-4>=<number>
Range	-2147483647 to 2147483647
Usage	Immediate, Read/Write
Code Example	`R1=50000 'Set Register 1 to 50000` `R2=Q2 'Set Reg. 2 to the value of User Variable Q2`
Equivalents	***MODBUS/TCP:*** 0x005F, 0.0061, 0x0062, 0x0063 ***EtherNet/IP:*** class: 0x65 instance: 1 attribute: 0x07-0A

RC (Motor run current)

Compatible with Lexium MDrive products:
Open Loop (RS-422/485)
Closed Loop (RS-422/485)
Open Loop (Ethernet)
Closed Loop (Ethernet)

Mnemonic	RC
Function	Motor run current
Type	Variable
Description	This variable defines the motor run current in percent.
Syntax	RC= <percent>
Units	Percent
Range	1 to 100
Default	25%
Usage	Program/Immediate
Code Example	`RC=50  'set run current to 50%`
Equivalents	***MODBUS/TCP:*** 0x0067 ***EtherNet/IP:*** class: 0x66 instance: 1 attribute: 0x0C
Related	HC

RD (Rotation direction)

Compatible with Lexium MDrive products:
Open Loop (RS-422/485)
Closed Loop (RS-422/485)
Open Loop (Ethernet)
Closed Loop (Ethernet)

⚠WARNING

UNINTENDED MOTION

Use of the RD command in Lexium MDrive Motion Control or Ethernet (Closed Loop models) with firmware versions 5.007 or earlier may, under certain conditions, result in unintended motion.

- Upgrade the device firmware to 5.009 or greater.

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

Mnemonic	RD
Function	Rotation of direction
Type	Variable
Description	This variable, when TRUE will reverse the default +/- motor direction. Cannot be issued when the axis is in motion or error 95 will be asserted.
Syntax	RD= <0/1>
Range	0 or 1
Default	0
Usage	Program/Immediate
Code Example	`RD=1  'switch the direction of rotation`

RS (Resume program execution)

Compatible with Lexium MDrive products:
Open Loop (RS-422/485)
Closed Loop (RS-422/485)
Open Loop (Ethernet)
Closed Loop (Ethernet)

Mnemonic	RS
Function	Resume program execution
Type	Instruction
Description	This instruction is used to resume a program that has been paused using the PS instruction. Any move that was paused will resume as well. Motion will resume using the normal acceleration profiles.
Syntax	RS
Usage	Immediate
Code Example	`RS  'Resume paused program`
Equivalents	***MODBUS/TCP:*** 0x41/0x42 ***EtherNet/IP:*** —
Related	PS, SL, MA, MR, HI, HM

RT (Return from subroutine)

Compatible with Lexium MDrive products:
Open Loop (RS-422/485)
Closed Loop (RS-422/485)
Open Loop (Ethernet)
Closed Loop (Ethernet)

Mnemonic	RT
Function	Return from called or on-error subroutine
Type	Instruction
Description	This instruction defines the end of a subroutine. This instruction is required and will be the final instruction in the subroutine executed by the CL or OE instruction. When used, it will return to the program address immediately following the instruction which executed the subroutine.
Syntax	RT
Usage	Program

Code Example	`CL K8  'Call Subroutine K8`
	`LB K8`
	`*****SUBROUTINE K8*****`
	`RT     'Go back to main program`

Equivalents	***MODBUS/TCP:*** — ***EtherNet/IP:*** —
Related	CL, OE

S (Save to NVM)

Compatible with Lexium MDrive products:
Open Loop (RS-422/485)
Closed Loop (RS-422/485)
Open Loop (Ethernet)
Closed Loop (Ethernet)

Mnemonic	S
Function	Save to NVM
Type	Instruction
Description	Saves all variables and flags currently in working memory (RAM) to nonvolatile memory (NVM). The previous values in NVM are completely overwritten with the new values. When the user modifies variables and flags, they are changed in working memory (RAM) only. If the S instruction is not executed before power is removed from the control module, all modifications to variables & flags since the last S will be lost.
Syntax	S
Usage	Program/Immediate
Code Example	`S    'Save all variable and flag states to NVM`
Equivalents	***MODBUS/TCP:*** — ***EtherNet/IP:*** —
Notes	Note: Communications during a Save could corrupt communications. If a Save is performed during the execution of a motion command, trips may be delayed. Use of the S command during a move (MA or MR) will generate an error 73, the save will not occur.

SC (System configuration test)

Compatible with Lexium MDrive products:
Open Loop (RS-422/485)
Closed Loop (RS-422/485)
Open Loop (Ethernet)
Closed Loop (Ethernet)

Mnemonic	SC
Function	System configuration test
Type	Instruction
Description	Tests encoder resolution and direction
Syntax	SC
Units	—
Range	—
Default	—
Usage	Program/Immediate, Write
Code Example	`SC 1  'start configuration test`
Equivalents	***MODBUS/TCP:*** 0x00A1 ***EtherNet/IP:*** class: 0x6A instance: 1 attribute: 0x01
Notes	The motor will rotate one half of a revolution during configuration test. Ensure the motor is safe to move before running SC. An error will be returned if the encoder is mis-configured. SC or SC 0 will return an error 24: illegal data entered.
Related	

SF (Stall factor)

Compatible with Lexium MDrive products:
Closed Loop (RS-422/485)
Closed Loop (Ethernet)

Mnemonic	SF
Function	Stall factor
Type	Variable
Description	If the encoder is enabled (EE = 1) and the encoder differs from the commanded position by more than the specified factor, a STALL is indicated. If SM is set to 0, then the motor will be stopped when a STALL is detected. If SM=1, the motor will not be stopped upon detection of a stall. ST will return an ER=86 on stall.
Syntax	SF=<counts>
Units	Encoder counts
Range	0 to 65000
Default	15
Usage	Program/Immediate, Read/Write
Code Example	`SF=20  'Set the stall factor for 20 counts`
Equivalents	***MODBUS/TCP:*** 0x0077 ***EtherNet/IP:*** class: 0x69 instance: 1 attribute: 0x07
Related	EE, SM, ST, PM

SL (Slew axis)

Compatible with Lexium MDrive products:
Open Loop (RS-422/485)
Closed Loop (RS-422/485)
Open Loop (Ethernet)
Closed Loop (Ethernet)

Mnemonic	SL
Function	Slew Axis
Type	Instruction
Description	The SL instruction will slew the axis at the specified velocity in steps per second. The axis will accelerate at the rate specified by the A (Acceleration) variable. Note that the maximum slew velocity is independent of the maximum velocity specified by the VM variable.
Syntax	SL <velocity>
Units	Motor Steps (EE=0)/Encoder Counts (EE=1)
Range	±5000000 (EE=0)/±200000 (EE=1)
Usage	Program/Immediate, Read/Write
Code Example	`SL 20000  'Slew at a rate of 20000 steps/sec`
Equivalents	***MODBUS/TCP:*** 0x0078 ***EtherNet/IP:*** class: 0x66 instance: 1 attribute: 0x0D
Notes	If 'SL 0' is issued after a MA/MR, motion has to come to a stop before issuing another motion command. This can be accomplished automatically with an 'H', <HOLD>, in user program mode.
Related	MA, MR, VI

SM (Stall detect mode)

Compatible with Lexium MDrive products:
Closed Loop (RS-422/485)
Closed Loop (Ethernet)

Mnemonic	SM
Function	Stall detection mode
Type	Variable
Description	The SM variable specifies the action which will be taken by the device when a stall is detected. When set to 0 (default) the motion will be stopped upon a stall detection. When SM=1, the motor will try to continue the move. In either case ST (Stall Flag) will be set. The functionality of SM when used with Position Maintenance (PM) is listed below: If SM = 0 and PM = 1, Position Maintenance will take place provided the position does not exceed the Stall Factor (SF). If SM = 1 and PM = 1, Position Maintenance will take place even if the Stall Factor (SF) is exceeded, unless VI is set too high causing the motor to stall.
Syntax	SM <mode>
Modes	0 Motion stops on stall detect (default) 1 Motion will attempt to continue
Default	0
Usage	Program/Immediate, Read/Write
Code Example	`SM=1  'Change Stall mode that the motor doesn't 'stop on stall detect`
Equivalents	***MODBUS/TCP:*** 0x007A ***EtherNet/IP:*** class: 0x69 instance: 1 attribute: 0x08
Related	EE, SM, ST, PM, SF

SN (Serial number)

Compatible with Lexium MDrive products:
Open Loop (RS-422/485)
Closed Loop (RS-422/485)
Open Loop (Ethernet)
Closed Loop (Ethernet)

Mnemonic	SN
Function	Serial number
Type	Variable
Description	Keyword allows the user to read the device serial number.
Syntax	PR SN
Usage	Read
Code Example	`PR SN  'read the device serial number`
Equivalents	***MODBUS/TCP:*** 0x2B/0x0E ***EtherNet/IP:*** class: 0x65 instance: 1 attribute: 0x0B

ST (Stall flag)

Compatible with Lexium MDrive products:
Closed Loop (RS-422/485)
Closed Loop (Ethernet)

Mnemonic	ST	
Function	Stall flag	
Type	Flag	
Description	The ST flag will be set to 1 when a stall is detected. It is the responsibility of the user to reset it to zero (0). Encoder function must be enabled (EE=1) in order for a stall flag to set.	
Syntax	PR ST BR <addr>, ST=1 CL <addr>, ST=1	
Status	0	Not stalled (default)
	1	Stalled
Default	0	
Usage	Program/Immediate, Read/Write	
Code Example	`CL K5,ST=1 'Call subroutine K5 if motor stalls` `ST=0 'Clear stall flag`	
Equivalents	***MODBUS/TCP:*** 0x00A2 ***EtherNet/IP:*** class: 0x6A instance: 1 attribute: 0x11	
Related	EE, SF, OE	

SU (Execute program on power up)

Compatible with Lexium MDrive products:
Open Loop (RS-422/485)
Closed Loop (RS-422/485)
Open Loop (Ethernet)
Closed Loop (Ethernet)

⚠ DANGER

UNINTENDED CONSEQUENCES OF EQUIPMENT OPERATION

Programs labeled with the SU label will execute on sytem power application or software reset. Depending on the program structure this could result in immediate motion on power application or system restart.

- Only use the SU label in instances or applications where operation does not represent a hazard to personnel or equipment.

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

Mnemonic	SU
Function	Execute program labeled SU on power up
Type	Predefined label
Description	The Start up label will cause any program labeled SU to automatically execute on power-up.
Syntax	LB SU
Usage	Program
Code Example	`LB SU 'Label program to executes on power up.`
Equivalents	***MODBUS/TCP:*** — ***EtherNet/IP:*** —

TA (Trip on HMT status)

Compatible with Lexium MDrive products:
Open Loop (RS-422/485)
Closed Loop (RS-422/485)
Open Loop (Ethernet)
Closed Loop (Ethernet)

Mnemonic	TA
Function	Trip on hybrid status
Type	Variable
Description	Will execute a subroutine address or label on the trip. The trip can be set to occure on any or all of three conditions: calibration done, hybrid active, locked rotor or lead/lag limit reached conditions.
Syntax	TA <flag>,<address/label>
Units	—
Range	0 — 7
Trip Condition	1 Calibration done
	2 Hybrid active
	4 Locked rotor
	8 Lag limit reached
	16 Lead limit reached
Default	0 (Off)
Usage	Program/Immediate, Read/Write
Code Example	`TA=4,k6 'execute subroutine k6 on locked rotor condition`
Equivalents	***MODBUS/TCP:*** — ***EtherNet/IP:*** —
Notes	The conditions are additive, eg. TA=3 will trip on calibration complete and hybrid active status. There is no error generation when enabling trip on locked rotor, lag limit or lead limit.

TC (Trip capture)

Compatible with Lexium MDrive products:
Open Loop (RS-422/485)
Closed Loop (RS-422/485)
Open Loop (Ethernet)
Closed Loop (Ethernet)

Mnemonic	TC
Function	Trip capture
Type	Variable
Description	Sets the Capture input trip for input 1. Sets one parameter for trip address. The TE command (Trip Enable/Disable TC) is reset when trip occurs. TE must be re-enabled in the main program prior to the next trip if it is to be repeated. The Trip subroutine must use a RETURN (RT) to exit the subroutine, use of a BRANCH will cause stack errors.
Syntax	TC=<address/label>
Usage	Program/Immediate, Read/Write
Code Example	`TC=K1 'Run subroutine K1 on Input Trip` `TE=4 'Re-enable Trip`
Equivalents	***MODBUS/TCP:*** — ***EtherNet/IP:*** —
Related	TE, IS

TD (Torque direction)

Compatible with Lexium MDrive products:
Closed Loop (RS-422/485)
Closed Loop (Ethernet)

Mnemonic	TD	
Function	Set torque direction	
Type	Variable	
Description	Sets torque direction to + or –	
Syntax	TD=<dir>	
Units	—	
Range	0 — 1	
Modes	0	Minus (CCW facing shaft)
	1	Plus (CW facing shaft)
Default	1	
Usage	Program/Immediate,Read/Write	
Code Example	`TD=0 'set torque direction to minus`	
Equivalents	***MODBUS/TCP:*** 0x00A5 ***EtherNet/IP:*** class: 0x6A instance: 1 attribute: 0x12	
Notes	Takes effect when torque mode starts	
Related	AS, TQ, TS	

TE (Trip enable)

Compatible with Lexium MDrive products:
Open Loop (RS-422/485)
Closed Loop (RS-422/485)
Open Loop (Ethernet)
Closed Loop (Ethernet)

Mnemonic	TE	
Function	Trip enable	
Type	Flag	
Description	The trip functions may be combined by adding trip numbers. For example TE=3 will trip on input or on position, TE=127 enables all trips. When multiple trips are used only the activated trip function needs to be re-enabled, the other trips will still be enabled.	
Syntax	TE=<trip>	
Range	0 — 127	
Trip	0	Disabled (default)
	1	Trip on input enabled
	2	Trip on position enabled
	4	Trip on capture (I/O 13) enabled
	8	Trip on time enabled
	16	Trip on relative position
	32	Trip on HMT status
	64	Trip on main power loss
Default	0 (Off)	
Usage	Program/Immediate, Read/Write	

Code Example	`TE=1 'Enable trip on input function`
	`TE=8 'Enable trip on time function`
	`TE=6 'Trip on Position or Capture input.`	

Equivalents	***MODBUS/TCP:*** — ***EtherNet/IP:*** —	
Related	I1-I4, P, IS, TA, TC, TI, TP, TR, TT	

TI (Trip on input)

Compatible with Lexium MDrive products:
Open Loop (RS-422/485)
Closed Loop (RS-422/485)
Open Loop (Ethernet)
Closed Loop (Ethernet)

Mnemonic	TI
Function	Trip on input
Type	Variable
Description	Sets up an input event (Trip) for the specified input. There are two parameters for the TI variable. The first specifies which input line to monitor. The second specifies the subroutine that should be executed when the input goes to true. The Trip subroutine must use a RETURN (RT) to exit the subroutine, use of a BRANCH will cause stack errors The TE is reset when a Trip occurs. TE must be re-enabled prior to the next Trip if it is to be repeated.
Syntax	TI=<input>,<address/label>
Usage	Program/Immediate, Read/Write
Code Example	`TI=2,K3  'execute subroutine K3 when input 2 active` `TE=1  'Re-enable Trip`
Equivalents	***MODBUS/TCP:*** — ***EtherNet/IP:*** —
Related	I1-4, IS, TE, TA, TC, TP, TR, TT

TM (Trip on main power loss)

Compatible with Lexium MDrive products:
Open Loop (RS-422/485)
Closed Loop (RS-422/485)
Open Loop (Ethernet)
Closed Loop (Ethernet)

Mnemonic	TM
Function	Trip on main power loss
Type	Variable
Description	Sets up an event (trip) to run a subroutine if main power is lost. In order for this to be used the Auxiliary power supply must be powered and connected. The TE (Trip Enable which Enables/Disables TP) is reset when a Trip occurs. TE must be re-enabled in the main program prior to the next Trip if it is to be repeated. The Trip subroutine must use a RETURN (RT) to exit the subroutine, use of a BRANCH will cause stack errors. Trips should be set BEFORE motion commands in the program.
Syntax	TM=<address/label>
Usage	Program/Immediate, Read/Write
Code Example	`TM=K9  'exe sub K9 when main power drops` `TE=64  'Re-enable trip`
Equivalents	***MODBUS/TCP:*** — ***EtherNet/IP:*** —
Notes	Note that Auxiliary power must be connected and active for TM to function.
Related	P, TI, PC, IS, CW, TR

TP (Trip on position)

Compatible with Lexium MDrive products:
Open Loop (RS-422/485)
Closed Loop (RS-422/485)
Open Loop (Ethernet)
Closed Loop (Ethernet)

Mnemonic	TP
Function	Trip on position
Type	Variable
Description	Sets up an event (trip) for the specified position. There are two parameters for the TP variable. The first specifies the position which will cause the event. The second specifies the subroutine that should be executed when the position is detected. The TE (Trip Enable which Enables/Disables TP) is reset when a Trip occurs. TE must be re-enabled in the main program prior to the next Trip if it is to be repeated. The Trip subroutine must use a RETURN (RT) to exit the subroutine, use of a BRANCH will cause stack errors. Trips should be set BEFORE motion commands in the program.
Syntax	TP=<position>,<address/label>
Usage	Program/Immediate, Read/Write
Code Example	`TP=650000,K9  'exe sub K9 when position = 650000` `TE=2  'Re-enable trip`
Equivalents	***MODBUS/TCP:*** — ***EtherNet/IP:*** —
Notes	Note that TP will always use motor counts regardless of the encoder enabled state (EE=1) Note that only a single position trip type may be used at a time. TP cannot be used simultaneously with TR.
Related	P, TI, PC, IS, CW, TR

TQ (Set torque)

Compatible with Lexium MDrive products:
Closed Loop (RS-422/485)
Closed Loop (Ethernet)

Mnemonic	TQ
Function	Set torque
Type	Variable
Description	Sets the maximum out put torque of the motor to a percentage.
Syntax	TQ=<percent>
Units	Percent
Range	1 — 100
Default	25%
Usage	Program/Immediate, Read/Write
Code Example	`TQ=50  'set torque to 50%`
Equivalents	***MODBUS/TCP:*** 0x00A6 ***EtherNet/IP:*** class: 0x6A instance: 1 attribute: 0x13
Related	AS, TS

TR (Trip on relative position)

Compatible with Lexium MDrive products:
Open Loop (RS-422/485)
Closed Loop (RS-422/485)
Open Loop (Ethernet)
Closed Loop (Ethernet)

Mnemonic	TR
Function	Trip on relative position
Type	Variable
Description	Sets up an event (trip) for the specified relative position. There are three parameters for the TR variable. The first specifies the position which will cause the event. The second specifies the subroutine that should be executed when the position is detected, if no subroutine address or label is specified then the High Speed Trip Output will activate. The Trip subroutine must use a RETURN (RT) to exit the subroutine, use of a BRANCH will cause stack errors The third parameter specifies the number of times the trip will repeat. If 0 (default) the trip will repeat infinite times, other wise the range is 1- 65000 The TE (Trip Enable which Enables/Disables TR) is reset after repeating the number of relative trips specified. TE must be re-enabled in the main program prior to the next series of Trip on Relative if it is to be repeated. For exampl, if TR=10000,0,25, the Output (S13) will trip 25 times in succession at 100,000 counts relative to the last position. Following these 25 trips the trip must be re-enabled (TE=16). Trips should be set BEFORE motion commands in the program.
Syntax	TR=<±dist>,<address/label>, <repeat>
Usage	Program/Immediate, Read/Write

Code Example	`TR=650000,0,0  'Activate trip out when at 650000`
	`'counts relative`
	`TE=16  'Re-enable trip`
	
	`TR=512000,K2,27  'Run Sub K2 when at 512000 rel,`
	`'repeat 27 times`
	`TE=16  'Re-enable trip`

Equivalents	***MODBUS/TCP:*** — ***EtherNet/IP:*** —
Notes	Note: Output S13 must be configured as a trip output (S13=61,1/0) Note that TR will always use motor counts unless the encoder is enabled (EE=1). Note: The maximum rate of trip is 20 kHz. Exceeding this may cause communications errors Note that only a single position trip type may be used at a time. TR cannot be used simultaneously with TP.
Related	CW, P, TI, PC, IS, TE, TP

TS (Set torque speed)

Compatible with Lexium MDrive products:
Closed Loop (RS-422/485)
Closed Loop (Ethernet)

Mnemonic	TS
Function	Set torque speed
Type	Instruction
Description	Determines the system speed for torque mode (AS=3) Hybrid will perform the following calculation based upon the value of TS:
Syntax	TS=<integer>
Units	steps/second
Range	46,512 — 2560000
Default	0
Usage	Program/Immediate, Read/Write
Code Example	`TS=50000  'set spd to 50000 steps/sec`
Equivalents	***MODBUS/TCP:*** 0x00A7 ***EtherNet/IP:*** class: 0x6A instance: 1 attribute: 0x14
Notes	TS is only available when in Torque Mode (AS=3).
Related	AS, MS

TT (Trip on time)

Compatible with Lexium MDrive products:
Open Loop (RS-422/485)
Closed Loop (RS-422/485)
Open Loop (Ethernet)
Closed Loop (Ethernet)

Mnemonic	TT
Function	Trip on time
Type	Variable
Description	Sets up a trip based on time. The first parameter is time in mSec. The second parameter specifies the subroutine that should be executed when the time is expired. The Trip subroutine must use a RETURN (RT) to exit the subroutine, use of a BRANCH will cause stack errors
Syntax	TT=<time>,<address/label>
Umits	Milliseconds
Range	1 to 65535
Usage	Program/Immediate, Read/Write
Code Example	`TT=2000,K3  'Trip on time 2000 mS, run- sub K3` `TE=8  'Re-enable trip`
Equivalents	***MODBUS/TCP:*** — ***EtherNet/IP:*** —
Related	TC, TE, TI, TP, TR

UG (Upgrade firmware)

Compatible with Lexium MDrive products:
Open Loop (RS-422/485)
Closed Loop (RS-422/485)
Open Loop (Ethernet)
Closed Loop (Ethernet)

Mnemonic	UG
Function	Upgrade firmware
Type	Instruction
Description	The upgrade command and code will be automatically entered by the Upgrader Utility in the Motion Control Programmer or SEM Terminal software programs. Once initiated, the firmware Upgrade MUST be completed.
Dependencies	The Motion Control Programmer module of the Lexium MDrive Software Suite or SEM Terminal are required to upgrade firmware.
Usage	Use Upgrade Firmware Utility in GUI.

UV (Read user variables)

Compatible with Lexium MDrive products:
Open Loop (RS-422/485)
Closed Loop (RS-422/485)
Open Loop (Ethernet)
Closed Loop (Ethernet)

Mnemonic	UV
Function	Read user variables
Type	Variable
Description	Read User Variables is used with the PR (Print) Instruction to read the value of all user variables.
Syntax	PR UV
Usage	Program/Immediate, Read
Code Example	`PR UV    'Read the value of all user variables`
Equivalents	***MODBUS/TCP:*** — ***EtherNet/IP:*** —
Related	PR, VA

V (Read velocity)

Compatible with Lexium MDrive products:
Open Loop (RS-422/485)
Closed Loop (RS-422/485)
Open Loop (Ethernet)
Closed Loop (Ethernet)

Mnemonic	V
Function	Read axis velocity
Type	Variable
Description	Used to read or act upon the current velocity of the axis in counts per second.
Syntax	PR V BR <address/label>, V=<velocity> BR <address/label>, V=<velocity>
Usage	Program/Immediate, Read

Code Example	`PR V            'Read the velocity`
	`CL Ka, V=20000  'Execute sub Ka when velocity is`
	`'20000/steps sec in the + direction`
	`CL Kb, V=-20000 'Execute sub Kb when velocity is`
	`'-20000/steps sec in the - irection`

Equivalents	***MODBUS/TCP:*** 0x0085 ***EtherNet/IP:*** class: 0x66 instance: 1 attribute: 0x0E
Notes	Note that V is signed. When using an Hybrid product V will not return an accurate value when Hybrid make up is active. In torque mode V will return 0.
Related	VI, VM, SL, MA, MR

VA (Create user variable)

Compatible with Lexium MDrive products:
Open Loop (RS-422/485)
Closed Loop (RS-422/485)
Open Loop (Ethernet)
Closed Loop (Ethernet)

Mnemonic	VA
Function	Create user variable name
Type	Instruction
Description	The VA instruction creates a user variable with a 1 or 2 character name. Can optionally set value assigned to that variable. The restrictions for this command are: 1. A variable cannot be named after a Lexium MCode Instruction, Variable or Flag or Keyword 2. The first character must be alpha, the second character may be alpha-numeric. 3. A variable is limited to two characters. 4. Limited to 192 variables and labels.
Syntax	VA <char><char>=<value>
Usage	Program/Immediate, Read/Write
Code Example	`VA Q3 'Create user Variable Q3` `VA Q3=20000 'Create user Variable Q3, set to 20000`
Equivalents	***MODBUS/TCP:*** — ***EtherNet/IP:*** —
Related	UV

VC (Velocity changing)

Compatible with Lexium MDrive products:
Open Loop (RS-422/485)
Closed Loop (RS-422/485)
Open Loop (Ethernet)
Closed Loop (Ethernet)

Mnemonic	VC
Function	Velocity changing
Type	Flag
Description	The read-only motion flag will be at an active state (1) when the velocity of the motor is changing, either accelerating or decelerating.
Syntax	PR VC BR <addr>, VC=1 CL <addr>, VC=1
Range	0/1
Trip	0 Motor stopped or at constant velocity (default) 1 Velocity is changing
Default	0 (not active)
Usage	Program/Immediate, Read/Write
Code Example	`CL K5,VC=1 'Call sub K5 if velocity is changing` `PR VC 'Print the state of the VC Flag`
Equivalents	***MODBUS/TCP:*** 0x0088 ***EtherNet/IP:*** class: 0x66 instance: 1 attribute: 0x0F
Related	MV, VI, VM, OS

VF (Torque velocity filter)

Compatible with Lexium MDrive products:
Closed Loop (RS-422/485)
Closed Loop (Ethernet)

Mnemonic	VF
Function	Set torque velocity filter
Type	Variable
Description	VF takes a value of 0 to 1000. It can be defined as 0 = no filtering and 1000 = most filtering. Because the Torque Velocity is computed and the encoder is sampled every mSec there can be fluctuation in the result. The filtering compensates for this fluctuation.
Syntax	VF=<integer>
Units	Counts
Range	0 to 1000
Default	0
Usage	Program/Immediate, Read/Write
Code Example	`VF=100 'Set filtering to 100 counts`
Equivalents	***MODBUS/TCP:*** — ***EtherNet/IP:*** —

VI (Initial velocity)

Compatible with Lexium MDrive products:
Open Loop (RS-422/485)
Closed Loop (RS-422/485)
Open Loop (Ethernet)
Closed Loop (Ethernet)

Mnemonic	VI
Function	Initial velocity
Type	Variable
Description	Initial velocity for all motion commands. The factory default value is 1000 clock pulses (steps) per second. The initial velocity for a stepper should be set to avoid the low speed resonance frequency and must be set lower than the pull in torque of the motor. It must also be set to a value lower than VM (Max. Velocity).
Syntax	VI=<velocity>
Units	Motor Steps (EE=0)/Encoder Counts (EE=1)
Range	1 to (VM –1)
Default	1000/40
Usage	Program/Immediate, Read/Write
Code Example	`VI=10000 'Set initial vel to 10000 steps/sec.`
Equivalents	***MODBUS/TCP:*** 0x0089 ***EtherNet/IP:*** class: 0x66 instance: 1 attribute: 0x10
Related	VM, MR, MA, HI, HM, JE

VM (Maximum velocity)

Compatible with Lexium MDrive products:
Open Loop (RS-422/485)
Closed Loop (RS-422/485)
Open Loop (Ethernet)
Closed Loop (Ethernet)

Mnemonic	VM
Function	Maximum velocity
Type	Variable
Description	The VM variable specifies the maximum velocity in steps/counts per second that the axis will reach during a move command. The maximum setting of VM is dependant on the setting of the Microstep Resolution and is equal to MS*10000. VM must be greater than VI. Changes to VM made during motion will not take effect until the current move completes.
Syntax	VM=<velocity>
Units	Motor Steps (EE=0)/Encoder Counts (EE=1)
Range	(VI + 1) to (MS*10000) (EE=0)/200000 (EE=1)
Default	768000/30720
Usage	Program/Immediate, Read/Write
Code Example	`VM=500000  'Set max vel to 500000 steps/counts sec.`
Equivalents	***MODBUS/TCP:*** 0x008B ***EtherNet/IP:*** class: 0x66 instance: 1 attribute: 0x11
Related	VI, MR, MA, HI, HM, JE

VR (Firmware version)

Compatible with Lexium MDrive products:
Open Loop (RS-422/485)
Closed Loop (RS-422/485)
Open Loop (Ethernet)
Closed Loop (Ethernet)

Mnemonic	VR
Function	Firmware version
Type	Variable
Description	This variable is used in conjunction with the PR instruction to read the version of the firmware installed at the factory.
Syntax	PR VR
Usage	Read
Code Example	`PR VR    'Read the firmware version installed`
Equivalents	***MODBUS/TCP:*** 0x2B/0x0E ***EtherNet/IP:*** class: 0x65 instance: 1 attribute: 0x0C
Notes	Hybrid products will return a second value for the hardware version.
Related	UG

VT (Read voltage)

Compatible with Lexium MDrive products:
Open Loop (RS-422/485)
Closed Loop (RS-422/485)
Open Loop (Ethernet)
Closed Loop (Ethernet)

Mnemonic	VT
Function	Read voltage
Type	Variable
Description	This read only variable will return the in/out of range status and the voltage seen at the Aux power and V+ voltage inputs.
Syntax	PR VT
Usage	Read
Code Example	`PR VT  'Read the Aux and V+ voltage`
Equivalents	***MODBUS/TCP:***TBD ***EtherNet/IP:*** TBD
Notes	PR VT will return three values: the power status, the Aux Voltage first, followed by the V+ voltage. Status reading: 0 = +V and Aux both in range 1= +V in range, Aux out of range 2= +V out of range, Aux in range 3= +V and Aux out of range ***NOTE:*** Firmware 5.004 and below will only display the Aux and +V voltage levels, not the status.
Related	IT

WT (Warning temperature)

Mnemonic	WT
Function	Warning temperature
Type	Variable
Description	The Warning Temperature variable allows the user to set a threshold temperature at which the device will print an error 71 to the terminal screen if the set temperature is exceeded.
Syntax	WT=<temp>
Units	Degrees C
Range	0 to 84
Default	80
Usage	Program/Immediate, Read/Write
Code Example	`WT=75  'set the warning temperature to 75 deg. C`
Equivalents	***MODBUS/TCP:*** — ***EtherNet/IP:*** class: 0x64 instance: 1 attribute: 0x06
Notes	NOTE: This is a single setting for both temp sensors. If Either reach the limit set by WT and error will be generated.
Related	IT

Page intentionally left blank

6 Supporting Software

The software associated with Lexium MDrive products is contained within the Lexium MDrive Software Suite. This software package is available for download at the Schneider Electric Motion USA web site at http://motion.schneider-electric.com.

The modules applicable are:

1. **Motion Control Programmer:**
   - Graphic User Interface (GUI) for developing and simulating Lexium MCode programs.
   - ANSI Terminal emulation with the ability for multiple terminal tabs to be open on different COM ports.
   - Program editor tabs with color coding.
   - Programmable function keys
   - Program simulator allows for quick test and debugging of Lexium MCode programs.
   - For RS-422/485 and Ethernet Lexium MDrive products
   - Motion Control Firmware upgrade utility.
2. **Ethernet Configuration Utility:**
   - For Lexium MDrive Ethernet products
   - Configure basic TCP/IP parameters such as:
     - IP address
     - Subnet mask
     - Gateway address
   - Firmware upgrades to Ethernet controller firmware

These modules are documented in separate manuals. The Manual for the module be used may be down loaded at:

http://motion.schneider-electric.com

This page intentionally left blank

7 Programming and application notes

This section will cover the following areas of Lexium MCode programming and applications in detail.

- Party mode communications
- Programming the I/O
- Factors impacting motion commands

7.1 Party mode communications

The following communication formats, used by Lexium MCode compatible devices.

{} ..... The contents between the {} symbols are transmitted.

{0D} ..... Hex equivalent for a CR (Carriage Return).

{0A} ..... Hex equivalent for a LF (Line Feed).

{DN} ..... Represents the Device Name being sent.

{CS} ..... Check Sum; {ACK} 06 Hex; {NAK} 15 Hex

EM = Echo Mode; PY = PartY Mode; CK= CheckK sum

The word {command} represents the immediate command sent to the device.

Command execution time (CET) is the time the device takes to execute a command. This varies from command to command and usually is in the 1-5 millisecond range.

7.1.1 Response to Echo Mode

Dependent on how the echo mode (EM) is set in conjunction with party mode (PY) and check sum (CK), the device will respond differently. The following tables illustrate the various responses based on how the EM, PY and CK parameters are set.

Parameter Setting	Transmission	Initial Response	Final Response	Notes
EM=0 & PY=0 CK=0	(command) (D)	(command) Echoed back one character at a time as the character is entered.	CET (0D) (0A)>	The last character sent is the prompt >
EM=1 & PY=0 CK=0	(command) (0D)	–	CET (0D) (0A)	The last character sent is LF
EM=2 & PY=0 CK=0	(command) (0D)	–	–	No response except to PR and L commands
EM=3 & PY=0 CK=0	(command) (0D)	–	CET command (0D) (0A)	Queued response. The last character sent is the LF

Table 7.1 Response to echo mode - party and check sum are zero (0)

Parameter Setting	Transmission	Initial Response	Final Response	Notes
EM=0 & PY=1 CK=0	(DN) (command) (0A)	(command) Echoed back one character at a time as the character is entered.	CET (0D) (0A)>	The last character sent is the prompt >
EM=1 & PY=1 CK=0	(DN) (command) (0A)	–	CET (0D) (0A)	The last character sent is LF
EM=2 & PY=1 CK=0	(DN) (command) (0A)	–	–	No response except to PR and L commands

EM=3 & PY=1 CK=0	(DN) (command) (0A)	–	CET command (0D) (0A)	Queued response. The last character sent is the LF

Table 7.2 Response to echo mode - party is one (1) and check sum is zero (0)

Parameter Setting	Transmission	Initial Response	Final Response	Notes
EM=0 & PY=0 CK=1	(DN) (command) (0A)	(command) Echoed back one character at a time as the character is entered.	CET (0D) (0A)>	The last character sent is the prompt >
EM=1 & PY=0 CK=1	(DN) (command) (0A)	–	CET (0D) (0A)	The last character sent is LF
EM=2 & PY=0 CK=1	(DN) (command) (0A)	–	–	No response except to PR and L commands
EM=3 & PY=0 CK=1	(DN) (command) (0A)	–	CET command (0D) (0A)	Queued response. The last character sent is the LF

Table 7.3 Response to echo mode - party is zero (0) and check sum is one (1)

Parameter Setting	Transmission	Initial Response	Final Response	Notes
EM=0 & PY=1 CK=1	(DN) (command) (CS) (0A)	(command) Echoed back one character at a time as the character is entered.	CET (ACK) or (NAK)>	The last character sent is the prompt >
EM=1 & PY=1 CK=1	(DN) (command) (CS) (0A)	–	CET (ACK) or (NAK)>	The last character sent is ACK or NAK
EM=2 & PY=1 CK=1	(DN) (command) (CS) (0A)	–	–	No response except to PR and L commands
EM=3 & PY=1 CK=1	(DN) (command) (CS) (0A)	–	CET command (CS) (ACK) (NAK)	Queued response. The last character sent is ACK or NAK

Table 7.4: Response to echo mode - party and check sum are one (1)

7.1.2 Using Check Sum

For communication using check sum, the following 2 commands demonstrate sending and receiving.

1) Check sum set to zero before first character is sent.
2) All characters (ascii values) are added to check sum, including the device name DN (if PY=1), to the end of the command, but not including terminator.
3) Check sum is 2's complement, then "or" ed with hex 80 (prevents check sum from being seen as command terminator).
4) Terminator sent.

Note: Any combination of upper/lower case may be used. In this example, if a lower case <mr> were to be used, the decimal values will change to 109 and 114. Subsequently the result check sum value will change. (Possible entries: MR, mr, Mr, mR.) (M = 77, R =82, m = 109, r = 114) (See ASCII table in Section 9 of this document.)

```
77  82  32  49    Decimal value of M, R, <space> and 1
4D  52  20  31    Hex
77+82+32+49 = 240        Add decimal values together
1111  0000 = 240  Change 240 decimal to binary
0000  1111 1's complement (invert binary)
0001 0000  Add 1 [2's complement]
1000  0000 OR result with 128 (Hex 80)
1001 0000 144     Result Check Sum value
```

Once the result is reached, add the check sum value (144 in this example) to your string by typing: MRr 1(alt key + 0144) (use the symbol of 0144 in your string by holding down the alt key and typing 0144). You must type the numbers from the numlock key pad to the right of the keyboard. The numbers at the top of the keyboard will not work.

1) Check sum set to zero.
2) All characters are added to check sum.
3) When receiving a command terminator, the lower 7 bits of the check sum should be equal to zero.

   A) if not zero, the command is ignored and NAK echoed.

   B) if zero, ACK is sent instead of CR/LF pair.
4) Responses to PR commands will be check summed as above, but the receiving device should not respond with ACK or NAK.

7.1.3 Immediate party mode sample codes

Once party mode has been defined and set up as previously described under the heading "multiple devices (party mode)", you may enter commands in the immediate mode in the ims terminal window. Some examples follow.

Move device A, B or C 10000 steps

Assuming there are three devices set up in party mode as shown in the sample codes above.

- To move mdrive unit “a”, press CTRL+J and then type: aMR∧10000 and press CTRL+J. device “a” will move 10000 steps.
- To print the position type: aPR p and press CTRL+J. The position of device “a” will be printed.
- To move device “b” type: bMR 10000 and press CTRL+J. Device “b” will move 10000 steps.
- To move all three devices at the same time type: *MR 10000 and press CTRL+J. All devices will move 10000 steps.
- To change a variable in the “c” unit type: c<variable name><number> and press CTRL+J. The variable will be changed. To verify the change type: cPR <variable name> and press CTRL+J. The new value will be displayed.
- All commands and variables may be programmed in this manner.
- To take a device out of party mode type: <device name>PY=0 and press CTRL+J. That unit will be taken out of party mode. To take all units out of party mode type: *PY=0 and press CTRL+J. All units will be taken out of party mode.

7.2 Programming the I/O

7.2.1 I/O availability per device type

The product families using the Lexium MCode language may have different sets of I/O points and functions. These are

NEMA size 17

- 3 — +5 to +24 VDC isolated input points. Programmable to multiple functions. Sink or source.
- 1 — analog input.
- 1 — high speed isolated output. Programmable to multiple functions including Trip.

NEMA Size 23 and 34

- 4 — +5 to +24 VDC isolated input points. Programmable to multiple functions. Sink or source.
- 1 — analog input.
- 2 — +5 to +24 VDC isolated outputs, dry contact configuration. Programmable to multiple functions
- 1 — high speed isolated output. Programmable to multiple functions including Trip.

7.2.2 Active states defined

The active state determines at what voltage level the input will be active.

Active HIGH: the input will be active when +5 to +24 VDC is applied to the input.

Active LOW: The input will be active when it is grounded (0 VDC).

Examples

Input 1 is to be configured as a Jog– input which will activate when a switch is toggled to ground:

```
IS=1,8,0 'set input 1 to jog-, active low
```

Input 4 is to be configured as a home input which will activate when instructed by a PLC (+24VDC sourcing input):

```
IS=4,1,1     'set input 4 to home, active high
```

7.2.3 Digital input functions

The inputs may be interfaced to a variety of sinking or sourcing devices. An input may be programmed to be a general purpose user input, or to one of 11 dedicated input functions. These may then be programmed to have an active state of either high or low.

The inputs are configured using the “IS” variable (see Section 5: Command details). The command is entered into the ims terminal or program file as:

```
IS=<line number>,<type>,<active low/high>
```

Example:

```
IS=3,3,0   'set input 3 = limit-, active low
IS=2,0,1   'set input 2 = gen. purpose, active high
```

NOTE: The Sink/Source Function is defined by the bias of the Input Reference input.

Connecting the input to a +5 to +24 VDC supply will provide for sinking inputs.

Connecting the input to Ground will provide for sourcing inputs.

Refer to Section 6 of the Lexium MDrive Motion Control Hardware Manual for examples.

Input Functions

The following table lists the programmable input functions.

Function	Description	Line	Type	Active
General Purpose	General purpose input function used to control program branches, subroutine calls or bcd functions when input bank is used as a group	1 — 4	0	0/1
Home	Homing input. Will function as specified by the home (hm) command.	1 — 4	1	0/1
Limit +	Positive limit input. Will function as specified by the limit (lm) command.	1 — 4	2	0/1
Limit –	Negative limit input. Will function as specified by the limit (lm) command.	1 — 4	3	0/1
G0	G0 input. Will run program located at address 1 on activation.	1 — 4	4	0/1
Soft Stop	Soft stop input. Stops motion with deceleration and stops program execution.	1 — 4	5	0/1
Pause	Pause/resume program with motion.	1 — 4	6	0/1
Jog +	Will jog motor in the positive direction at max. Velocity (vm). The jog enable (je) flag must be set for this to function.	1 — 4	7	0/1
Jog –	Will jog motor in the negative direction at max. velocity (VM). The jog enable (JE) flag must be set for this to function.	1 — 4	8	0/1
Reset	When set as reset input, then the action is equivalent to a ^c entered into a terminal.	1 — 4	11	0/1
Capture	Capture input will operate with the Trip Capture (TC) trip to run a subroutine when active. Only applicable to input 1. Capture function not available on the NEMA 17 (42 mm) Lexium MDrive models.	1	12	0/1

Table 7.5 Digital input functions

7.2.4 Digital output functions

The outputs may be configured as general purpose or set to dedicated functions, such as fault or moving. These outputs will sink up to 600 mA (one channel of two banks) and may be connected to an external VDC source.

The outputs are set using the “Os” command (see Section 5 of this document for precise details on this command). The command is entered into the terminal or program file as:

```
OS=<line>,<type><active low/high>
```

Examples

```
OS=1,17,0  'set output 3 to moving, active high
OS=3,0,0   'set output 3 to be error, active low
```

Output Functions

Output functions may be programmed to be a general purpose user output with the following functions. Shaded areas apply only to units with an internal encoder installed.

Function	Description	Line	Type	Active
General Purpose User	A general purpose output can be set in a program or in immediate mode to trigger external events. When used as a group they can be a BCD output.	1 — 3	16	0/1
Moving	Will be in the active state when the motor is moving.	1 — 3	17	0/1
Software error	Will be in the Active State when a error occurs. .	1 — 3	18	0/1
Stall	Will be in the active state when a stall is detected. Encoder required, stall detect mode (SM) must be enabled.	1 — 3	19	0/1
Velocity Changing	Will be in the active state when the velocity is changing. Example: during acceleration and deceleration.	1 — 3	20	0/1
Locked Rotor	Will be in an active state when the rotor is locked on MDrive Hybrid products	1 — 3	21	0/1
Moving to Position	Will be active when the motor is indexing to a commanded position.	1 — 3	23	0/1
Hybrid Active	Will be active when the Hybrid control circuitry is engaged.	1 — 3	24	0/1
Make Up Active	Will be active when the Hybrid is correcting lead/lag conditions.	1 — 3	25	0/1
Trip	Trip output applies to output 3 only, active low only	3	28	0
Attention	When active, indicates a status or statuses as configured by the AO variable.	1 — 3	29	0/1

Table 7.6 Digital output functions

7.2.5 Programmable input usage examples

The code examples below illustrate possible interface examples for using the digital I/O.

Reference the hardware manual of your device for connection and wiring information

Input Interface Example - Switch Input

The following example shows a switch connected between an I/O point and power ground.

Code Sample

For the code sample, this switch will be set up as a G0 sinking input, active when low. When pressed, the switch will launch the program beginning at address 1 in device memory:

```
***Setup Variables***
IS=4,4,0      'set input 4 to be a G0 input, active LOW

****Program***
PG1
MR 20000      'Move +20000 steps relative to current pos.
H             'Hold program execution until motion completes
MR -20000     'Move -20000 steps
H             'Hold program execution until motion completes
E

PG            'End program, exit program mode
```

Input interface example - switch input

The following circuit example shows a switch connected between an I/O point and a voltage supply which will source the input to perform a function.

Code Sample

For the code sample, the switch will be set up as a soft stop sourcing input, active when high. When pressed, the switches will stop the motor.

```
IS=1,5,1      'set input 1 to Soft Stop, active when HIGH

SL 200000     'slew the motor at 200000 µsteps/sec
```

When the switch is depressed the motor will decelerate to a stop.

Output interface example

The following circuit example shows a load connected to an I/O point that will be configured as a sinking output.

Code Sample

For the code sample, the load will be an LED. The motor is configured such that the LED will be lit while the motor is at constant velocity. Set input 1 up to be a soft stop input using a switch in a sinking configuration this will soft stop the motor.

```
IS=1,5,1      'set input 1 to Soft Stop, active when HIGH
OS=1,20,0     'set output 1 as a Velocity Changing activeLOW
SL 2000000    'slew the motor at 200000 µsteps/sec
```

While the motor is accelerating the LED will be dark, but will light up when the motor reaches a constant velocity. When the Soft Stop switch is depressed the motor will begin to decelerate, the LED will go dark again while velocity is changing.

Output interface example

The following circuit example shows a load connected to an I/O point that will be configured as a sourcing output.

Code Sample

For the code sample, the load will be a relay. The output will be configured to be a general purpose user output that will be set active when a range of motion completes.

```
******Setup Variables******
OS=1,16,1      'set IO 1 = user output, active HIGH

******Program******
PG 100         'Enter program at address 100
MR 2000000     'Move x in the positive direction
H              'Hold execution until motion completes
MR -1000000    'Move x distance negative direction
H              'Hold execution until motion completes
O1=1           'Set output 1 HIGH
```

Enter EX 100 to execute the program, the motion will occur and the output will set high.

Reading inputs as a group example

The inputs may read as a group using the IN keyword. This will display as a decimal between 0 to 15 representing the 4 bit binary number (The IN keyword will function on the 42mm (NEMA 17) devices but will only read inputs 1 - 3. Inputs should be configured as user inputs (IS = <line>,0).

```
PR IN          'Reads Inputs 4(MSB) – 1(LSB)
```

Interfacing outputs as a group example

Outputs may be written to as a group using the OT keywords\ . This will set the outputs as a binary number representing the decimal between 0 to 7 representing the 3 bit binary number on 57 mm (NEMA 23) and 85 mm (NEMA 34) devices but will have no practical use on 42 mm (NEMA 17) devices. The outputs should be configured to the general purpose user type (S=<line>,16).

```
OT=5  'set the binary state of the combined I/O to 101
```

7.2.6 Analog input usage

The analog input is configured from the factory as a 0 to 5V, 12 bit resolution input (IS = 5,9,0). This offers the user the ability to receive input from temperature, pressure, or other forms of sensors, and then control events based upon the input.

The value of this input will be read using the I5 instruction, which has a range of 0 to 4095, where 0 = 0 volts and 4095 = 5.0 volts. The anlog input may also be configured as 0 to 10 vols (IS = 5,9,1) for a 4 to 20 mA (IS = 5,10,0) or 0 to 20 mA Analog Input (IS = 5,10,1). If used as a 4 to 20mA input the range is 0 to 3200 units.

Sample Usage

```
'***********Main Program*************
IS=5,9,0              'set analog to read voltage (0 to +5VDC)
PG 100                'start prog. address 100
LB A1                 'label program A1
CL A2, I5<500         'Call Sub A2, If I5 is less than 500
CL A3, I5>524         'Call Sub A3, If I5 is greater than 524
BR A1                 'loop to A1

'***********Subroutines*************

LB A2         'label subroutine A2
MA 2000       'Move Absolute 2000 steps
H             'Hold program execution until motion ceases
RT            'return from subroutine

LB A3         'label subroutine A3
MA -2000      'Move Absolute -2000 steps
H             'Hold program execution until motion ceases
RT            'return from subroutine
E             'End

PG            'Exit program
```

7.3 Factors impacting motion commands

7.3.1 Motor steps

All Lexium MCode examples assume 200 step motors. They rotate at 1.8° per clock pulse. 200 steps would equal 1 revolution.

Microsteps divide the 200 motor steps into smaller steps to improve smoothness and resolution of the Lexium MCode compatible device. Using the default setting of 256 for MS, the 200 motor steps are increased to 51200 microsteps. One motor revolution requires 51200 microsteps with the ms set at 256. If you were to set MS to 128, one revolution of the motor would now require 25600 microsteps.

7.3.3 Move Command

The move absolute (MA) and the move relative (MR) commands are programmed in microsteps or if the encoder is enabled, encoder counts. If the ms was set at 256 and you were to program a move of 51200 microsteps, the motor would turn one full revolution. If the ms was set to 128, one full revolution of the motor would be 25600 microsteps (128 x 200). If you programmed a move of 51200, the motor would turn 2 full revolutions.

7.3.4 Closed loop control with an encoder

If the encoder is enabled the move commands use different values. The encoder has 1000 lines and yields 4000 counts or counts per revolution. Therefore, the MR and MA command values are programmed in encoder counts. One full revolution would be programmed as mr or ma 4000.

When the encoder is enabled, the MS value is defaulted to 256. It cannot be changed.

Knowing these factors you can program a multitude of different movements, speeds, and time intervals.

7.3.5 Linear movement

You have a rack and pinion or a ball screw to move a linear axis. The rack and pinion or ball screw moves the linear axis 0.1 inches for each revolution. You need to move 7.5 inches.

7.5 inches divided by 0.1 inches = 75 motor revolutions.

Assuming an MS of 256 (51200 Microsteps) is programmed, 51200 Microsteps x 75 revolutions requires a move of 3840000 microsteps.

Knowing the values of the variables as well as the required move, you can calculate the actual time it takes to move the axis the required distance. This is done with a trapezoidal profile as shown below.

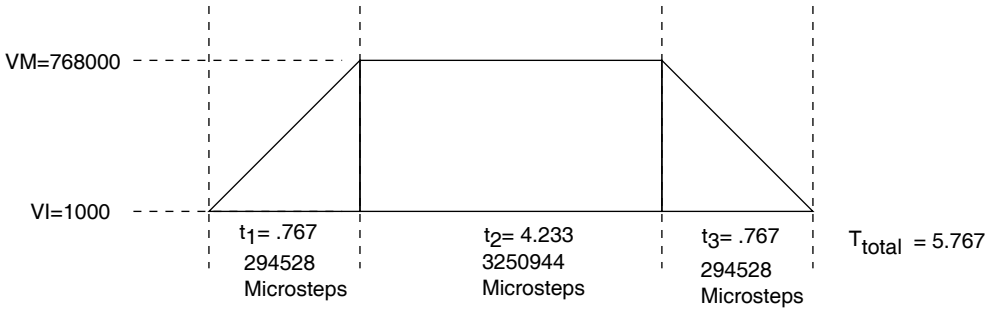


Figure 7.4 Trapezoidal move profile

Calculating axis speed (velocity)

There are several steps required to determine the actual axis speed. They are all based on the Trapezoidal Profile above.

Known Values and Parameters:

VM.....768000 Steps/Sec.

VI.....1000 Steps/Sec.

A.....1000000 Steps/Sec2.

D .....1000000 Steps/Sec2.

MA/MR.....3840000 Microsteps

Determine the Acceleration (A) and Deceleration (D) times (t1 and t3). Since the Deceleration (D) value is also 1000000 Steps/Sec. the Deceleration time (t3) will be the same as the Acceleration time (t1).

$$(t1 \text{ and } t3) = \frac{VM - VI}{A} \textbf{ OR } \frac{768000 - 1000}{1000000} \textbf{ = 0.767 Seconds}$$

Determine the distance (Steps) traveled in t1 or t3.

$$\text{Distance} = \frac{VM + VI}{2} \times t1 \textbf{ OR } \frac{768000 + 1000}{2} \times 0.767$$

$$\textbf{= 294911 steps}$$

Determine the t2 time.

The t2 time is calculated by dividing the remainder of MA/MR by VM.

The remainder of MA/MR = MA/MR - (t1 steps + t3 steps) or 3840000 - 589056 = 3250944.

$$t2 = \frac{3250944}{768000} \textbf{ = 4.233 Seconds}$$

Determine the total time. (t1 + t2 + t3) or (0.767 + 4.233 + 0.767) = 5.767 Seconds

The linear axis took 5.767 seconds to move 7.5 inches or an average speed of 78 inches/minute.

Note that the average speed includes the Acceleration and Deceleration. The maximum axis speed attained is approximately 90 inches/minute.

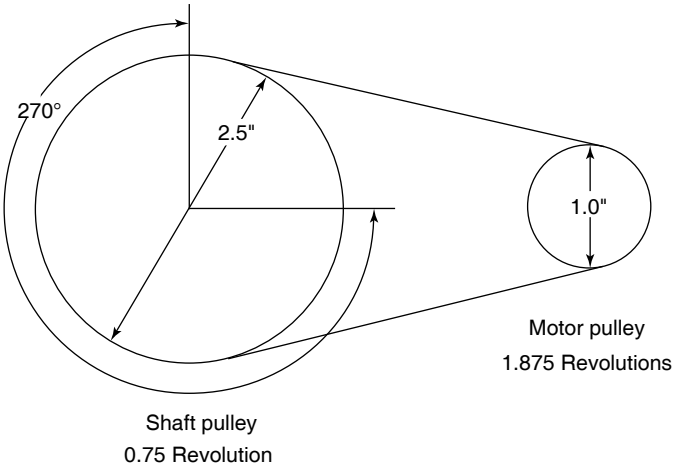
$$\frac{768000}{51200} \text{ x } 0.1 \text{ x } 60 \textbf{ = 90 IPM}$$

7.3.6 Calculating rotary movement

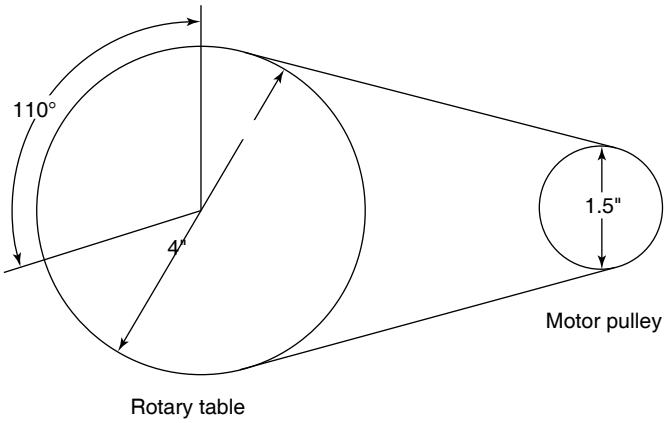
Assume that MS is set to 256. You are using the motor to drive a shaft with a timing belt and pulley arrangement. As shown below, the pulley is 1” in diameter and the shaft pulley is 2.5” in diameter. You must turn the shaft 270°.

- The shaft will rotate 1 full revolution for every 2.5 revolutions of the motor.
- 270° is 0.75 of a revolution.
- 0.75 x 2.5 = 1.875 motor revolutions to turn the shaft 270°.
- If 51200 Microsteps is 1 motor revolution, then the device must be programmed to move 96000 Microsteps (51200 x 1.875).

You may also do many of the calculations in reverse to calculate motor moves to meet a required move of your device. A linear or rotational move as well as speed may be translated into an Lexium MCode command.



Rotary drive example 1



Rotary drive example 2

Figure 7.5 Rotary examples

In the example above, the belt driven rotary table must be turned 110° at 3 RPM. How should the device be set up?

Bear in mind that all the numbers are approximate due to rounding.

Mechanical ratio between the motor and the rotary table is 2.666:1. That is, the motor must rotate 2.666 revolutions for the table to rotate 1 revolution and the table will rotate 2.666 times slower than the motor.

In order to move the table 110° the motor must move 293.3°.

110 x 2.66 **= 293.3°**

If 51200 steps = 1 revolution then 1° = 142.222 steps.

$$\frac{51200}{360} = \textbf{142.222 steps}$$

The Lexium MCode device must be programmed to move 41713 steps to rotate 293.3°.

142.222 steps x 293.3° **= 41713 steps**

In order to rotate the table at 3 RPM the motor must turn at 8 RPM.

3 RPM x 2.666 **= 8 RPM**

If you were to set VM at 51200 and MS set at 256 the motor will rotate 1 full revolution (51200 steps) in 1 second or 1 RPS. In order to rotate at 8 RPM, the motor must rotate at 0.13333 RPS.

$$\frac{8}{60} = \textbf{0.133333 RPS}$$

In order to rotate at 0.13333 RPS the VM must be set at 6827 steps/sec.

51200 x 0.133333 **= 6827**

VM = 6827

Note: These numbers will vary slightly depending on Acceleration and Deceleration rates.

7.3.7 Programming with the optional encoder enabled

Compatible with Lexium MDrive products:
Closed Loop (RS-422/485)
Closed Loop (Ethernet)

An optional 1000 line magnetic encoder is available. When the Encoder is enabled (EE=1) the programming also changes. All motion must now be programmed by the encoder counts. The Encoder operates in the "Quadrature" format. That is, there are four Encoder counts for each Encoder line or 4000 counts per revolution (1000 × 4 = 4000). (See Figure below.) If you were to program motion using the MR (Move Relative) or MA (Move Absolute) commands the motor would rotate a distance equal to the encoder counts.

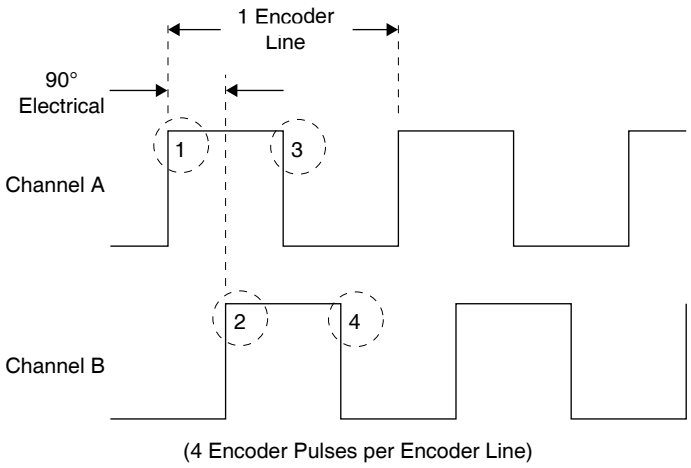


Figure 7.6 Encoder waveform

Example: A programmed move of 14000 counts would result in the motor rotating 3.5 revolutions at a velocity controlled by VM.

```
14000 ÷ 4000 = 3.5 revolutions
```

If you were to program motion using the SL (Slew) command the motor would rotate at a "counts per second" rate based on the programmed value.

Example: An SL (Slew) rate of 14000 counts was programed. The motor will rotate at 14000 counts/sec., 3.5 RPS, or 210 RPM.

```
14000 ÷ 4000 = 3.5 RPS × 60 = 210 RPM
```

When the Encoder is enabled, the parameters are also changed to be compatible with the 4000 counts.

The Encoder Enabled defaults are:

VM.....60000 Counts/Sec.

VI.....78 Counts/Sec.

A.....78125 Counts/Sec

D.....78125 Counts/Sec.

MS.....256 (default for encoder mode.)

To enable the encoder the program syntax is <EE=n> where n is a zero (0) or a one (1). The default is zero (0) which is encoder disabled. To enable the encoder, program EE=1.

Any motion will now be programmed in encoder counts. You can calculate the distance or velocity you need in a similar manner as done previously only with different factors.

Note: The microstep select is defaulted and locked at 256 in the encoder mode to ensure stable, high resolution.

Several Variables work in conjunction with Encoder Enable (EE). They are:

DB .....Encoder Deadband

SF.....The Stall Factor Variable

SM.....The Stall Detection Mode

ST.....Stall Flag

PM.....Position Maintenance

EE .....Encoder Enabled

When the encoder is enabled, all motion is “closed loop”. That is, motion steps are delivered from the Lexium MCode device to the motor which turns the encoder. The encoder sends counts back to the drive to complete the motion. If you programmed a move of 2048 counts, the device would output an appropriate number of microsteps provided the stall factor (SF) value or other fault is not encountered. If no faults were encountered, the device would output the full amount of microsteps. Depending on which variables were set, the driver would then wait until the position (plus or minus the encoder deadband) was read and confirmed.

DB - Encoder Deadband

The Encoder Deadband is a Variable that is set in Encoder Counts. Motion will be deemed complete when the Encoder Counts are within ± the Deadband variable. With DB=5 the motion of 2048 counts would be complete between 2043 and 2053 counts.

SF - Stall Factor

The Stall Factor is a Variable which is entered in Encoder Counts. The Stall Factor is active only in the EE=1 mode. The Stall Factor might be compared to the “following error” or “lag error” of a servo drive. The Stall Factor is triggered by the number of steps output from the device to the motor as compared to the number of counts returned by the encoder. The comparison should always be within the value of the Stall Factor, otherwise a fault will occur and the Stall Flag (ST) will be set. If the Stall Detection Mode is active (SM=0), the motion will be stopped.

Example:

A Stall Factor of 30 counts (SF=30) is programmed. A motion command of 2048 counts is programmed. The device reaches a mechanical bind at 2000 counts. The device will keep outputting steps equivalent to 2030 counts (present position plus the SF value) and then the Stall Flag (ST) will be set. The motor will be stopped if the Stall Detection Mode (SM=0) is active.

SM - Stall Detection Mode

The Stall Detection Mode can be programmed to stop the device (SM=0) or to allow the device to continue (SM=1) when the Stall Factor (SF) is reached. Whether SM is active or not, the Stall Flag will always be set when the SF is encountered.

ST - Stall Flag

The Stall Flag will be set any time the SF is reached regardless of the state of the Stall Detection Mode (SM). If the Stall Flag is set, the user must reset it to zero (0).

PM - Position Maintenance

Position maintenance (PM) is active only after the motion has completed. Position maintenance is used to maintain position when there might be an external force on the drive. If position maintenance is enabled (PM=1) and the stall detection mode is enabled (SM=0), the motor will be driven back to its final position if it was forced out of position provided the stall factor (SF) was not reached.

If position maintenance is enabled (PM=1) and the stall detection mode is disabled (SM=1), the motor will be driven back to its final position if it was forced out of position regardless of whether the stall factor (SF) was reached or not.

There are three other variables, although not directly conned to EE, that do affect the overall operation when in encoder mode, they are:

HC..... Motor Hold Current

HT..... Motor Hold Current Delay Time

MT.....Motor Settling Delay Time

HC.....Hold Current

When motion is complete, the device will switch from motor run current (RC) to motor hold current (HC). The hold current is set at a lower percentage than the run current (rc). However, the hold current must be sufficient to overcome an outside force such as driving a vertical slide which maintains a load on the motor at all times. Actual hold current

values will vary depending on the application and the load on the motor when it is at rest.

HT - Motor Hold Current Delay Time

The motor hold current delay time (HT) is a variable that delays the change from run current (RC) to hold current (HC) at the end of a move. The end of the move is triggered by the device when it has completed outputting the correct number of steps. Depending on the application, including velocity, deceleration, load and inertia, the device may lag behind a few counts. The ht will allow the device to finish its move before applying the lower HC.

MT - Motor Settling Delay Time

A stepping motor may ring or oscillate in minuscule amounts at the completion of a move until it satisfies the target position. The amount of this “ringing” is dependent on the application including velocity, deceleration, inertia, friction and load. The motor settling delay time (MT) allows the motor to stop “ringing” before checking the position count. If the device tried to check the position count during this ringing, it would assume a position error and try to correct an already moving motor and possibly cause ringing of a larger magnitude and longevity. Typically, the MT is set between 50 and 100 milliseconds. It is recommended that there is always a Motor Settling Time programmed any time you are in EE=1 mode.

Note: If MT has no value, the motor may hunt and never satisfy the position check.

8 Sample programs

This section is made up of several example programs designed to aid the user in discovering the Lexium MCode programming language.

Download sample programs

All the sample programs from this section may be downloaded from the web site at http://motion.schneider-electric.com in *.ixt ((Motion Control Programmer) format.

> **DOWNLOAD NOW**

8.1 Move on an input

Note: Do not copy/paste this code from the PDF into the Editor tab of the Motion Control programmer. Use the file move_on_input.ixt in the **sample_programs.zip** file

```
'[VARIABLES]
'This block contains the global variable and system
'configuration information.

Is=1,0,0
Ms=256
Vi=200000
Vm=2500000
A=1000000
D=A
Hc=2
Rc=75
P=0


'[PROGRAMS]
'The program block for this application sets the event
'that triggers the subroutine call when input 1 is active
'and loops when I1=inactive

PG 1
LB Ga      'Program execution label
  P=0
LB G1      'Loop back label
  CL Kb,I1=1
  H 10
  BR G1
E

'Subroutine from trigger event will execute a ten
'revolution positive move, hold, then return to 0 in the
'negative direction and repeat as long I1=1

LB Kb      'subroutine label
  MA 512000
  H
  MA 0
  H
  RT
PG                 'exit program
S
'Keep this line to save program on load
'[END]
```

Enter EX Ga or EX 1 in the termianl tab to run

8.2 Change velocity during a move

This program will demonstrate ability to change speed during move. The device does not have abiltity to change speed during point to point move, so we use the slew command with position trips. End position trip, decel and slew speed determine actual ending position. Program is written to print ending position to serial port 10 times for averaging, expected end position = 102400.

Use the file change_speed_during_a_move.ixt in the **sample_programs.zip** file

```
'[VARIABLES]
'This block contains the global variable and system
'configuration information.
Hc=20
Rc=100

'[PROGRAMS]
PG 1
'Program label Ga sets local variables and register
'values. These are re-initialized each time the program
'is executed.
LB Ga
  Vi=20000
  Vm=500000
  A=500000
  D=800000000
  R1=0
  R2=0

'Label Gx sets the trip response and Performs Register
'Math to print final position
LB Gx
  P=0
  Tp=51200,Kb
  Te=2
  SL 101200
  H
  H 250
  IC R1
  R2=R2+P
  BR Gx,R1<10
  R2=R2/100
  PR "Average end pos = ",R2
  E

'[SUBROUTINES]
'Subroutine Kb, when called by Tp=51200 increases the
'axis velocity by 50%
LB Kb
  SL 202400
  Tp=102290,Kc
  Te=2
  RT

'Subroutine Kc, when called from Kb ends the motion
'sequence
LB Kc
  SL 0
  H
  RT

PG
S
'Keep this line to save program on load
'[END]
```

Enter EX Ga or EX 1 in the terminal tab to run

8.3 Binary mask

This program will demonstrate ability to execute various subroutines depending on the binary value of inputs 1-3 while masking all i/o above input 3.

Use the file binary_mask.ixt in the **sample_programs.zip** file

```
'[VARIABLES]
'Define I/O configuration
Is=1,0,0
Is=2,0,0
Is=3,0,0
Is=4,0,0
Os=1,16,0

'Set up system variables
Vi=20000
Vm=1000000
A=500000
D=A
Hc=20
Rc=75

'[PROGRAMS]
'The main program block is labeled SU 'a keyword which
'will execute the program on power up.

PG 1
LB Su
  P=0

'The block G1 will cally various subroutines based upon
'the wieght of the inputs which is stored in register R1

LB G1
  R1=In
  R1=R1 & 7
  O1=0
  CL K0,R1 = 0
  CL K1,R1 = 1
  CL K2,R1 = 2
  CL K3,R1 = 3
  CL K4,R1 = 4
  CL K5,R1 = 5
  CL K6,R1 = 6
  CL K7,R1 = 7
  H 10
  BR G1
  E

'[SUBROUTINES]
'These 8 routines will rotate the motor
'1 time for each input bit and repeat
'the input wieght changes

LB K0
  PR "Logic 000"
  MR R1*51200
  H
  O1=1
  H 2000
  RT

LB K1
  PR "Logic 001"
  MR R1*51200
  H
  H 200
  RT
```

```
LB K2
  PR "logic 010"
  MR R1*51200
  H
  H 200
  RT

LB K3
  PR "Logic 011"
  MR R1*51200
  H
  H 200
  RT

LB K4
  PR "Logic 100"
  MR R1*51200
  H
  H 200
  RT

LB K5
  PR "Logic 101"
  MR R1*51200
  H
  H 200
  RT

LB K6
  PR "Logic 110"
  MR R1*51200
  H
  H 200
  RT

LB K7
  PR "Logic 111"
  MR R1*51200
  H
  H 200
  RT

PG
S
'Keep this line to save program on load

'[END]
```

Program will execute on power on or software reset (CTRL+C)

8.4 Closed Loop

This program illustrates closed loop control with an On Error (OE) routine which will perform math functions on the counters to display the position error.

Use the file closed_loop_on_error.ixt in the **sample_programs.zip** file

```
'[VARIABLES]
Rc=80
Mt=50

'HMT off and encoder functions enabled and configured
As = 0
Ee=1
Sf=15
Sm=0

'motion variables are scaled to encoder counts instead of
'microsteps
A=20000
D=A
Vi=2048
Vm=15000

'user variable created to hold move count
VA Q1

'[PROGRAMS]
'program block Ga sets the on error handle routine to
'call K1
PG 1
LB Ga
  OE K1
  P=0

'program block Gb contains the motion loop which will run
'100 times
LB Gb
  MR 51200
  H
  H 500
  MR -51200
  H
  H 500
  IC Q1
  BR Gb,Q1<100
  CL K1
E

'[SUBROUTINES]
'Subroutine K1 sets the response for the on-error
'handler. It will perform some math to 'determine the
'position error in encoder counts, as well as display the
'error # if one occurs.
LB k1
  R3=C1/25
  R1=R3 - C2
  PR "Counts error = ",R1
  PR "Error = ",Er
  Er=0
  H 20
  RT

PG
S
'Keep this line to save program on load
'[END]
```

8.5 User input into variables

This program demonstrates the ability to hold up program execution while the user enters multiple variables. Uses variable K1 and K2 to enter the amount and direction of motor rotation.

Use the file user_input_into_variables.ixt in the **sample_programs.zip** file

```
'[VARIABLES]
'System configuration variables
Ms=256
Vi=200000
Vm=2500000
A=1000000
D=A
Hc=10
Rc=75

'Globally defined user variables to contain
'input data
VA K1=0
VA K2=0
VA K3=51200
VA K4=0

'[PROGRAMS]
'Program labeled Su will start on power on
'or software reset. Will zero the position
'counter and wait 2 sec before dropping to
'program block Z1
PG 1
    LB Su

    P=-0
    PR "At Home Position"
    H 2000

 'Block will request a number of desired
 'revolutions and insert the number into
 'variable K1
 LB Z1
     PR "Enter the number of revolutions in whole numbers"
     IV K1
     LB X1
     BR X1, If=1
     H 50

'Block will request a direction of
 'rotation and insert the number into
 'variable K2, then call the appropriate
 'subroutine with error checking for
 'invalid entries

 LB X4
     PR "Enter rotation direction (0) neg. (1) pos."
     IV K2
     LB X2
     BR X2, If=1
     H 50
     BR Y1,K2=0
     BR Y2, K2=1
     PR "Invalid Entry"
     BR X4
```

```
‘X6 will orint the final position of the axis
‘to the terminal screen
LB X6
     VA K5
     K5=P/K3
     PR “Axis position is “, K5, “ absolute from home”
     H 3000

‘Block X5 will initiate following the commanded
‘move with an option to re-run or quit
LB X5
     PR “Repeat program (1) or quit (0)”
     IV K4
     LB X3
     BR X3, If=1
     BR Z1, K4=1
     BR Z2, K4=0
     PR “Invalid Entry”
     BR X5

‘[SUBROUTINES]
‘The following branch routines will
‘calculate the move distance and
‘direction and execute the move
LB Y1
     MR -K3*K1
     H
     BR X6


LB Y2
     MR K3*K1
     H
     BR X6

‘[END]
LB Z2
PR “Program Ended”
E
PG
S
‘Keep this line to save program on load
```

8.6 Closed loop with homing

This program demonstrates the use of the home to home switch instruction (HM) in closed loop, also there is a move on input routine.

The Homing method used is HM1, which will slew at VM (Max Velocity) in the negative direction, when input 1 is activated, the axis will creep in the plus direction at VI (Initial Velocity). See the MCode Home to home switch command and change the homing method to experiment with different methods of homing. Output 1 is set to activate when the axis is moving. Stalling the motor will generate an error, activating output 2.

Use the file closed_loop_with_homing.ixt in the **sample_programs.zip** file.

```
'[VARIABLES]
'Global variable declarations
Ee=1
Vm=4096
Vi=Vm/50
A=20480
D=A
Hc=50
Rc=50
Mt=50

'Encoder setup

Sf=20
Sm=0
Db=5

'I/O setup
Is = 1, 1, 0  'Homing input
Is = 2, 0, 1  'General purpose input
Os = 1, 17, 1  'Moving output
Os = 2, 18, 1  'Error output
D1=100

'[PROGRAMS]
'Main program will home in mode 1 Slew minus @ VM until
'to find home switch then creep plus @ VI

PG 1
LB G1
   H 1000
   PR C1 ,C1
   PR C2 ,C2
   Pm=1
   PR "Position counter: " C1
   PR "Encoder counter: " C2
   H 5000
   HM 1
   H
   P=0

'After homing, motor will move @ 7186 steps each move
'printing position each time
 LB G2
   BR G2,I2=1
   MR 7186
   H
   PR "Position: " P
   BR G2
E
PG
S
'Keep this line to save program on load on load
```

8.7 Input trip

This program demonstrates the use input trips. The Lexium MDrive will perform a short 1 revolution move in each direction repeating four times when input 1 is toggled.

When using a mechanical switch, remember to set the input filtering to avoid erroneous trips.

IMPORTANT! Trip Rules:

1. Trip must be enabled using Te=<num> following the trip definition.
2. Only a single input trip may be defined in a program.
3. Trip must be re-enabled to re-execute trip.

Use the file trip_on_input.ixt in the **sample_programs.zip** file.

```
'LEXIUM MDRIVE DEMO PROGRAM
'Last modified 12/13/12

'[VARIABLES]
VA Q1
D1=255

'[PROGRAMS]
'Program will run a motion
'profile on an input toggle
PG 1
LB G1
  Ti = 1, X1
  Te = 1
  LB G2
  Q1 = 0
  BR G2
E

'Motion profile
LB X1
  IC Q1
  MR 51200
  H
  MR -51200
  H
  BR X1, Q1 < 4
  Te = 1
  RT
  E
PG      ' End of Program
S
'Keep this line to save program on load
```

8.8 Position teach (encoder required)

This program allows the user to “teach” the Lexium MDrive a +/- move profile based on manually positioning the motor shaft. The shaft is manually moved to a position, then an input is toggled to store that position in encoder counts to a user variable. The shaft is moved to second position, the input is again toggled to store the second position in a second variable.

The motor will then move between the two stored positions.

Use the file position_teach.ixt in the **sample_programs.zip** file.

```
'LEXIUM MDRIVE DEMO PROGRAM
'Last modified 12/14/12

'[VARIABLES]
VA Q1 = 0
VA Q2 = 0
D1 = 255
D2 = 255

'HMT off, encoder enabled
As=0
Ee=1

'[PROGRAMS]
'Program stores a +/- move p profile based on encoder
'counts set by manually positioning the motor shaft
'An input toggle stores the encoder counts to a user
'variable.

PG 1
LB Su
   Er = 0
   C2=0
   Q1 = 0
   O2 = 0
   PR "Move motor to position 1"
   PR "Toggle switch 1 when ready"
LB X1
   BR X1, I1 = 0
   Q1 = C2
   PR Q1
LB X2
   BR X2, I1 = 1
   PR "Move motor to position 2"
   PR "Toggle switch 1 when ready"
LB X3
   BR X3, I1 = 0
   Q2 = C2
   PR Q2
LB X4
   BR X4,I1=1
   PR "Toggle Sw 2 to start cycle"
LB X5
   BR X5, I2 = 0
   LB X6
   MA Q1
   H
   PR P
   H 250
   MA Q2
   H
   PR P
   H 250
   BR X5
E
PG                ' End of Program
S
'Keep this line to save program on load
```

8.9 Analog speed control

This program demonstrates the use of the analog input in a speed control application.

The program subroutine performs calculations using the user registers R1-R4 and slews the axis bi-directionally based upon the value seen on the analog input.

Hardware requirement: 10kΩ potentiometer connected to the Analog input.

Use the file analog_speed_control.ixt in the **sample_programs.zip** file

```
'LEXIUM MDRIVE DEMO PROGRAM
'Last modified 12/14/12

'[VARIABLES]
Os = 1, 20, 1 'Velocity changing output
A=2000000
D=2000000
R4=80

'[PROGRAMS]
'The main program block calls
'subroutine to calculate a slew rate
'based on the value of I5
PG 1
LB G1
   R1 = I5
   CL Z1
   SL R3
   H 10
   BR G1
E

'Subroutine performs calculation
'to vary the velocity based upon
'the analog input
LB Z1
   R1 = R1-2032
   R2 = 1
   BR Z2, R1>=0
   R2 = -1
   R1 = R1 * R2

LB Z2
   BR Z3,R1<R4
   R1 = R1 * 625
   R3 = R1 * R2
RT

LB Z3
   R3=0
RT

E
PG
S
'Keep this line to save program on load
```

8.10 Analog slew with stall detect

This program will use the analog input reading to ram the velocity until the motor stalls. When the stall occurs, an error is generated.

A subroutine is triggered by the error to:

Print the Error number and stalled sate of the motor,

Use the file analog_slew_with_stall_detect.ixt in the **sample_programs.zip** file.

```
'LEXIUM MDRIVE DEMO PROGRAM
'Last modified 12/13/12

'[VARIABLES]
As=0
Ee=1
Sf=30

'[PROGRAMS]
'Main program will assign
'a register to do math on the value of the analog
'input and slew the register value. An on-error event
'calls a subroutine to register stall
PG 1
LB Su
   OE X1
   Er=0
   R1=I5
   R1=R1*50
   SL R1
   PR V
   H 250
   BR Su
E

'[SUBROUTINES]
'on error routine
LB X1
   PR "Error! " Er
LB Y1
   BR Y2,Er <> 86
   PR "Stall"
LB Y2
   Er=0
E
PG                    ' End of Program
S
'Keep this line to save program on load
```

8.11 Multiple position trips

This program will use the position trip function multiple times to change position and velocity, each time printing the position and velocity to the terminal screen.

Use the file multiple_position_trips.ixt in the **sample_programs.zip** file.

```
'LEXIUM MDRIVE DEMO PROGRAM
'Last modified 02/21/2013

'Distance traveled is V * 12 sec.
' 33 RPM * 0.2 min =  6.6 revs. X 51200 uSteps =  337920
uStp.
' 66 RPM * 0.2 min = 13.2 revs. X 51200 uSteps =  675840
uStp.
'100 RPM * 0.2 min = 19.8 revs. X 51200 uSteps = 1013760
uStp.
' Slew Rate is V/60 * 51200 uSteps/sec.
' 33 RPM /60 Sec.  X 51200 uSteps =  28160 uStp./Sec.
' 66 RPM /60 Sec.  X 51200 uSteps =  56320 uStp./Sec.
'100 RPM /60 Sec.  X 51200 uSteps =  85333 uStp./Sec.

'Step 1  V=33RPM
'Step 2  V=66RPM
'Step 3  V=100RPM
'Step 4  V=66RPM
'Step 5  V=33RPM
'Step 6  V=-33RPM
'Step 7  V=-66RPM
'Step 8  V=-100RPM
'Step 9  V=-66RPM
'Step 10 V=-33RPM

'STEPS
'   1   2    3    4   5    6    7   8    9    10
'
'          ____
'       ___|    |____
'   ___|             |___
'_|_____________________|_____________________________
__Ds
'                      |___                  ____|
'                          |____      ____|
'                               |____|


'[VARIABLES]
VA Sp=33*51200/60 'Step 1 speed 33 RPM * 51200 Stp/rev
/60 s/m
VA Ds=50000  'Length of first step.

'[PROGRAMS]
PG 1
  R1=0
  P=0       'set position counter to 0
  LB Aa
    PR ""
    PR "Starting Step 1   P=",P,"    V=",V
    Tp Ds,X1
    Te=2
    Vm=Sp       'Step 1 speed 33 RPM * 51200 Stp/rev /60
s/m
    MA Ds*9,0,1
    LB G1
      BR G1,R1=0
      R1=0
      BR Aa
    E
```

```
'[SUBROUTINES]
'Each sub will move a dist at a velocity
'then redefine and re-enable the trip

'Step 2 speed 66 RPM * 51200 Stp/rev /60 s/m
  LB X1
    PR " Starting Step 2   P=",P,"   V=",V
    Vm= Sp*2
    MA Ds*9,0,1
    Tp Ds*3,X2
    Te=2
    RT

'Step 3 speed 100 RPM * 51200 Stp/rev /60 s/m
  LB X2
    PR "  Starting Step 3   P=",P,"  V=",V
    Vm= Sp*3
    MA Ds*9,0,1
    Tp  Ds*6,X3
    Te=2
    RT

'Step 4 speed 66 RPM * 51200 Stp/rev /60 s/m
  LB X3
    PR "   Starting Step 4   P=",P,"  V=",V
    Vm= Sp*2
    MA Ds*9,0,1
    Tp Ds*8,X4
    Te=2
    RT

'Step 5 speed 33 RPM * 51200 Stp/rev /60 s/m
  LB X4
    PR "    Starting Step 5   P=",P,"  V=",V
    Vm= Sp
    MA Ds*9 ',0,1
    Tp Ds*9,X5
    Te=2
    RT

'Step 6 speed 33 RPM * 51200 Stp/rev /60 s/m
  LB X5
    PR "     Starting Step 6   P=",P,"  V=",V
    Vm= Sp
    MA 0,0,1
    Tp Ds*8,X6
    Te=2
    RT

'Step 7 speed 66 RPM * 51200 Stp/rev /60 s/m
  LB X6
    PR "      Starting Step 7   P=",P,"  V=",V
    Vm= Sp*2
    MA 0,0,1
    Tp Ds*6,X7
    Te=2
    RT

'Step 8 speed 100 RPM * 51200 Stp/rev /60 s/m
  LB X7
    PR "       Starting Step 8   P=",P,"  V=",V
    Vm= Sp*3
    MA 0,0,1
    Tp Ds*3,X8
    Te=2
    RT
```

```
'Step 9 speed 66 RPM * 51200 Stp/rev /60 s/m
  LB X8
    PR "        Starting Step 9   P=",P,"  V=",V
    Vm= Sp*2
    MA 0,0,1
    Tp Ds,X9
    Te=2
    RT
'Step 10 speed 33 RPM * 51200 Stp/rev /60 s/m
  LB X9
    PR "         Starting Step 10  P=",P,"   V=",V
    Vm= Sp
    MA 0
    H
    PR "          Back at Start     P=",P,"   V=",V
    R1=1
    RT

  PG
S
'Keep this line.
```

Page intentionally left blank

9 Error codes

A question mark <?> displayed as a cursor indicates an error. To determine what the error is, type <pr er> in the terminal window. The device will respond with an error number displayed in the terminal window. The error number may then be referenced to this list.

0	No Error

9.1 I/O errors

1	Over-current condition on output 1
2	Over-current condition on output 2
6	An I/O is already set to this type. Applies to non-General Purpose I/O
8	Tried to SET IO to an incorrect I/O type
9	Tried to write to I/O set as Input or is “TYPED”
10	Illegal I/O number
11	Incorrect CLOCK type
12	Input 1 not defined as a capture input

9.2 Data errors

20	Tried to set unknown variable or flag. Trying to set an undefined variable of flag. Also could be a typo.
21	Tried to set an incorrect value. Many variables have a range such as the Run Current (RC) which is 1 to 100%. As an example, you cannot set the RC to 110%.
22	VI is set greater than or equal to VM. The Initial Velocity is set equal to, or higher than the Maximum Velocity. VI must be less than VM.
23	VM is set less than or equal to VI. The Maximum Velocity is set equal to, or lower than the Initial Velocity. VM must be greater than VI.
24	Illegal data entered. Data has been entered that the device does not understand.
25	Variable or flag is read only. Read only flags and variables cannot be set.
26	Variable or flag is not allowed to be incremented or decremented. IC and DC cannot be used on variables or flags such as Baud and Version.
27	Trip not defined. Trying to enable a trip that has not yet been defined.
28	WARNING! Trying to redefine a program label or variable. This can be caused when you download a program over a program already saved. Before downloading a new or edited program, type <FD> and press ENTER to return the device to the Factory Defaults. You may also type <CP> and press ENTER to Clear the Program.
29	Trying to redefine a built in command, variable or flag.
30	Unknown label or user variable. Trying to Call or Branch to a Label or Variable that has not yet been defined.

31	Program label or user variable table is full. The table has a maximum capacity of 22 labels and/or user variables.
32	Trying to set a label (LB). You cannot name a label and then try to set it to a value. Example: Lable P1 (LB P1). The P1 cannot be used to set a variable such as P1=1000.
33	Trying to SET an Instruction.
34	Trying to Execute a Variable or Flag
35	Trying to Print Illegal Variable or Flag
36	Illegal Motor Count to Encoder Count Ratio
37	Command, Variable or Flag Not Available in Drive
38	Missing parameter separator
39	Trip on Position and Trip on Relative Distance not allowed together

9.3 Program errors

40	Program not running. If HOLD (H) is entered in Immediate Mode and a program is not running.
41	Stack overflow
42	Illegal program address. Tried to Clear, List, Execute, etc. an incorrect Program address.
43	Tried to overflow program stack. Calling a Sub-Routine or Trip Routine with no Return.
44	Program locked. User Programs can be Locked with the <LK> command. Once Locked, the program cannot be listed or edited in any way.
45	Trying to Overflow Program Space.
46	Not in Program Mode.
47	Tried to Write to Illegal Flash Address
48	Program Execution stopped by I/O set as Stop.

9.4 Communications errors

60	Tried to enter an unknown command
61	Trying to set illegal BAUD rate. The only Baud Rates accepted are those listed on the Properties Page of IMS Terminal. (4,800, 9,600, 19,200, 38,400, 115,200)
62	IV already pending or IF Flag already TRUE.
63	Character over-run. Character was received. Processor did not have time to process it and it was over-written by the next character.

9.5 System errors

70	FLASH Check Sum Fault
71	Internal Temperature Warning
72	Internal Over Temperature disabling drive
73	Tried to SAVE while moving
74	Tried to Initialize Parameters (IP) or Clear Program (CP) while Moving
76	Microstep resolution set to low, must be greater than min sys. speed.
77	VM, VI, or SL too great for selected microstep resolution
78	Aux Power out of range
79	V+ out of range

9.6 Motion errors

80	HOME switch not defined. Attempting to do a HOME (H) sequence but the Home Switch has not yet been defined.
81	HOME type not defined. The HOME (HM or HI) Command has been programmed but with no type or an illegal type. (Types = 1, 2, 3, or 4)
82	Went to both LIMITS and did not find home. The motion encroached both limits but did not trip the Home switch. Indicates a possible bad switch or a bad circuit.
83	Reached plus LIMIT switch. The LIMIT switch in the plus direction was tripped.
84	Reached minus LIMIT switch. The LIMIT switch in the minus direction was tripped.
85	MA or MR isn't allowed during a HOME and a HOME isn't allowed while the device is in motion.
86	Stall detected. The Stall Flag (ST) has been set to 1.
87	Not allowed to change AS mode while moving
88	Moves not allowed while calibration is in progress.
89	Calibration not allowed while motion is in progress.
90	Motion Variables are too low switching to EE=1
91	Motion stopped by I/O set as Stop.
92	Position Error in Closed loop. motor will attempt to position the shaft within the dead-band, After failing 3 attempts Error 92 will be generated. Axis will continue to function normally.
93	MR or MA not allowed while correcting position at end of previous MR or MA.
94	Motion commanded while drive disabled (DE=0)
95	Rotation of direction (RD) attempted while axis is in motion
96	Motion attempted when +V is out of range

9.7 Hybrid errors

100	Configuration test done, encoder resolution mismatch
101	Configuration test done, encoder direction incorrect
102	Configuration test done, encoder resolution and direction incorrect
103	Configuration not done, drive not enabled
104	Locked rotor. The Locked Rotor flag will also be active (LR=1). Clear by issuing a CF (Clear Locked Rotor Fault).
105	Maximum position count reached
106	Lead limit reached
107	Lag limit reached
108	Lead/lag not zero at the end of a move
109	Calibration failed because drive not enabled.
110	Make up disabled.
111	Factory calibration failed

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 : 12 / 2014