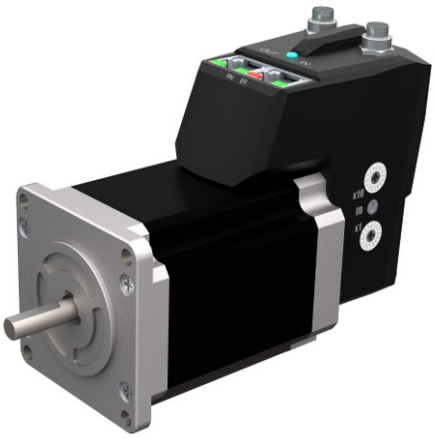
MYOSTAT MOTION CONTROL INC.

CM1-T User Guide

Version: 3.00
Date: 12/08/2025



TCP
UDP

EtherNet/IP™

Modbus TCP/IP

1 Table of Contents

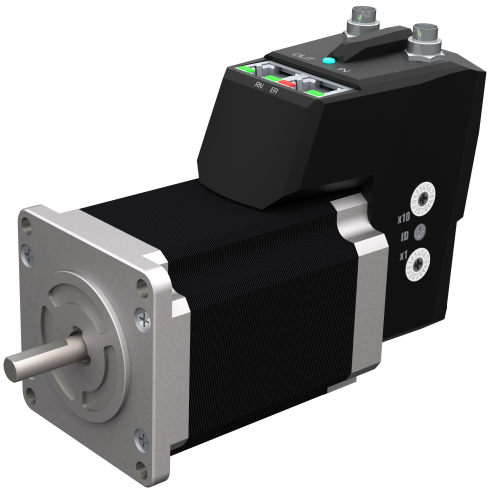
2 CM1-T DESIGN REFERENCES

Not all links are enabled in the pdf user guide. Go to https://docs.myostat.ca/cm1-t/user-guide/ to access all downloads.

The intelligence and efficiency of the Cool Muscle servos, combined with an EtherNet/IP interface and separate power for control and drive.

The CM1-T Cool Muscle Ethernet servos provide EtherNet/IP, MODBUS TCP, and TCP/IP interface options.

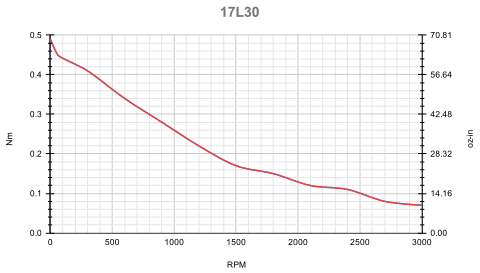
The CM1-T motor is a certified EtherNet/IP device, including a range of add-on instructions that speed time to installation.



2.1 USER GUIDE

Use the search bar or links on the left to find the content you are looking for. Alternatively download a copy of the user guide to pdf.

2.3 TECHNICAL SPECIFICATIONS



2.4 DATASHEETS

- CM1-T-17X30D.PDF
- CM1-T-23XX0D.PDF

2.5 CAD FILES

- 3D Models
- DXF Files

2.6 PLC Support

2.6.1 EtherNet/IP Support Files

- Add-on Instructions, Data Types, Examples

2.2 Product Brochure

2.6.2 IEC 61131-3 Library

- CM1-T Library

2.7 SOFTWARE

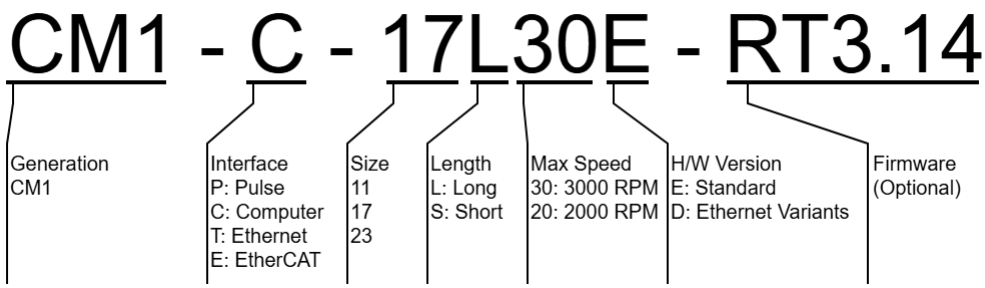
2.7.1 CM1-T Configuration Tool

- CM1-T Configuration Tool

2.7.2 Programming IDE

- Control Room IDE

3 Motor Specifications



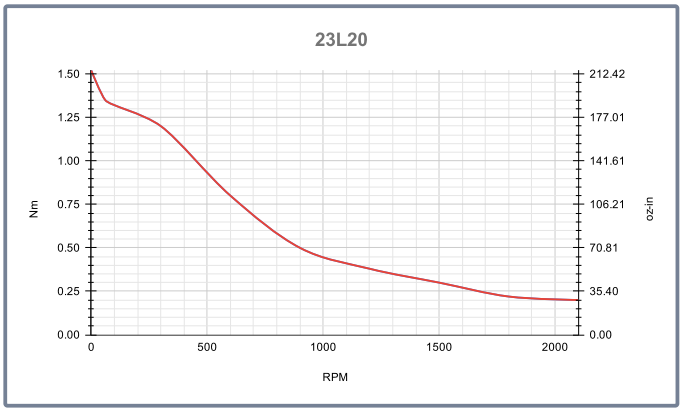
3.1 Specifications

Model	11L30	11S30	17L30	17S30	23L30	23S30	Unit
Max Speed	3000	3000	3000	3000	2000	3000	**RPM**
Continuous Torque	0.055 (7.78)	0.027 (3.8)	0.36 (50.98)	0.084 (11.89)	0.89 (126)	0.3 (42.48)	**NM (oz•in)**
Peak Torque	0.078 (11.1)	0.039 (5.5)	0.53 (75)	0.12 (16.56)	1.24 (175.6)	0.42 (65.14)	**NM (oz•in)**
Max Load Inertia	180 (2.5 x 10^{-3})	80 (1.1 x 10^{-3})	760 (1.07 x 10^{-2})	380 (5.38 x 10^{-3})	4.6 x 10^{3} (6.5 x 10^{-2})	1.4 x 10^{3} (1.9 x 10^{-2})	**g•cm^2 (oz•in•s^2)**
Motor Inertia	18 (2.5 x 10^{-4})	8 (1.1 x 10^{-3})	74 (1.04 x 10^{-3})	36 (5.09 x 10^{-4})	3.6 x 10^{2} (5.09 x 10^{-3})	180 (2.5 x 10^{-3})	**g•cm^2 (oz•in•s^2)**
Input Supply Voltage	24 ± 10%						**VDC**
Continuous Current	1.2	0.8	1.5	0.8	2.6	3.9	**A**
Peak Current	1.5	1.0	1.8	1.0	3.4	5.1	**A**
Operating Temp.	0 - 40						**°C**
Storage Temperature	-20 - 60						**°C**
Operational Humidity	<90						**% Relative Humidity**
Shock	<10						**G**

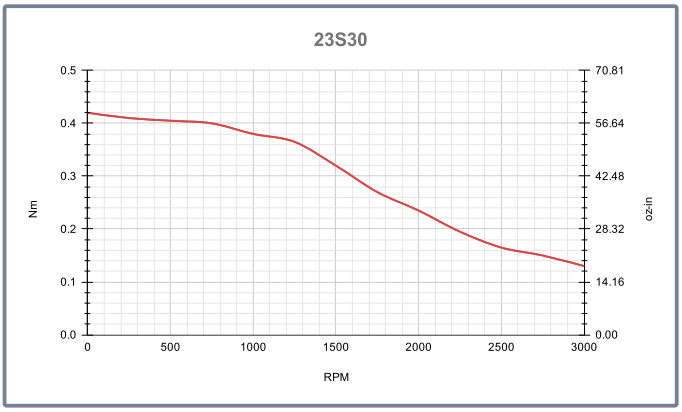
Vibration	<1						**G**
Mass	295	234	470	323	1110	552	**g**

3.2 Torque-Speed Curves

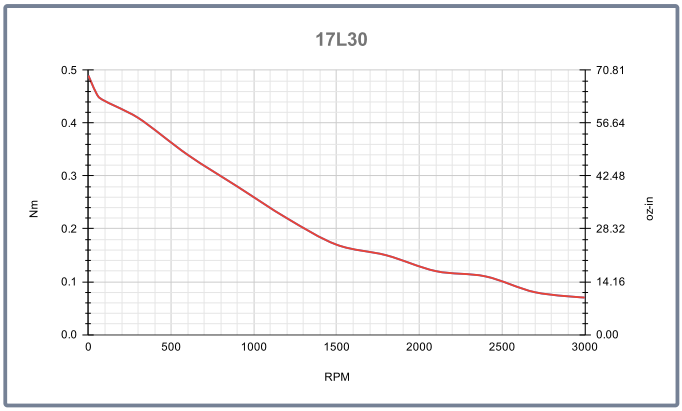
3.2.1 23L20



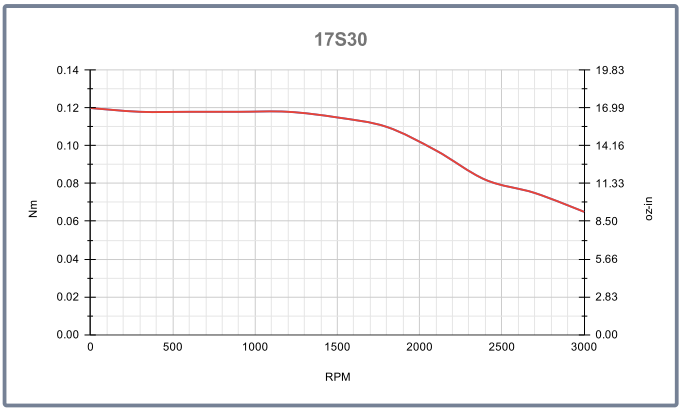
3.2.4 23S30



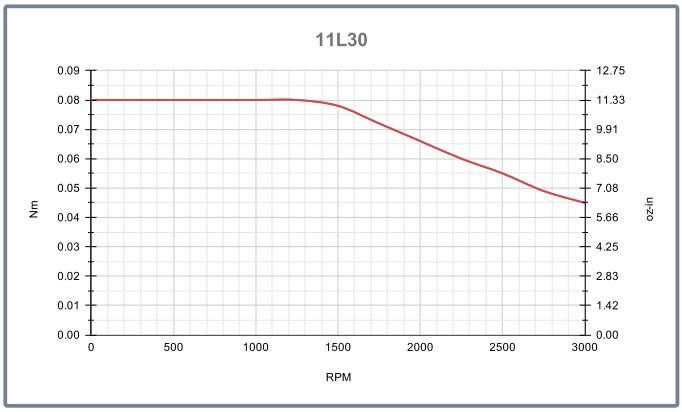
3.2.2 17L30



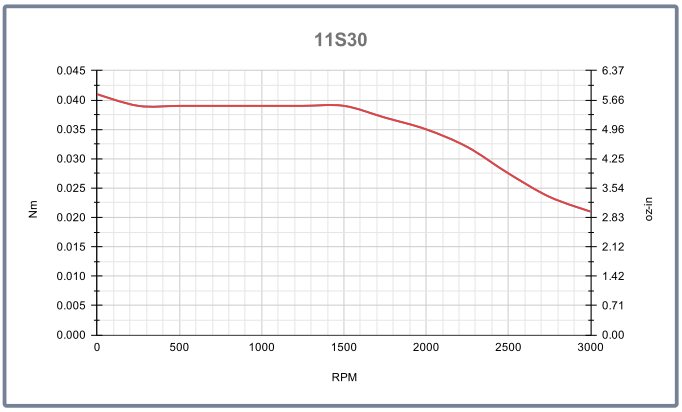
3.2.5 17L30



3.2.3 11L30



3.2.6 11S30



4 Using the CM1-T

⚠ This manual covers all Ethernet functionality of the CM1-T. For CML programming see the CM1-C User Guide.

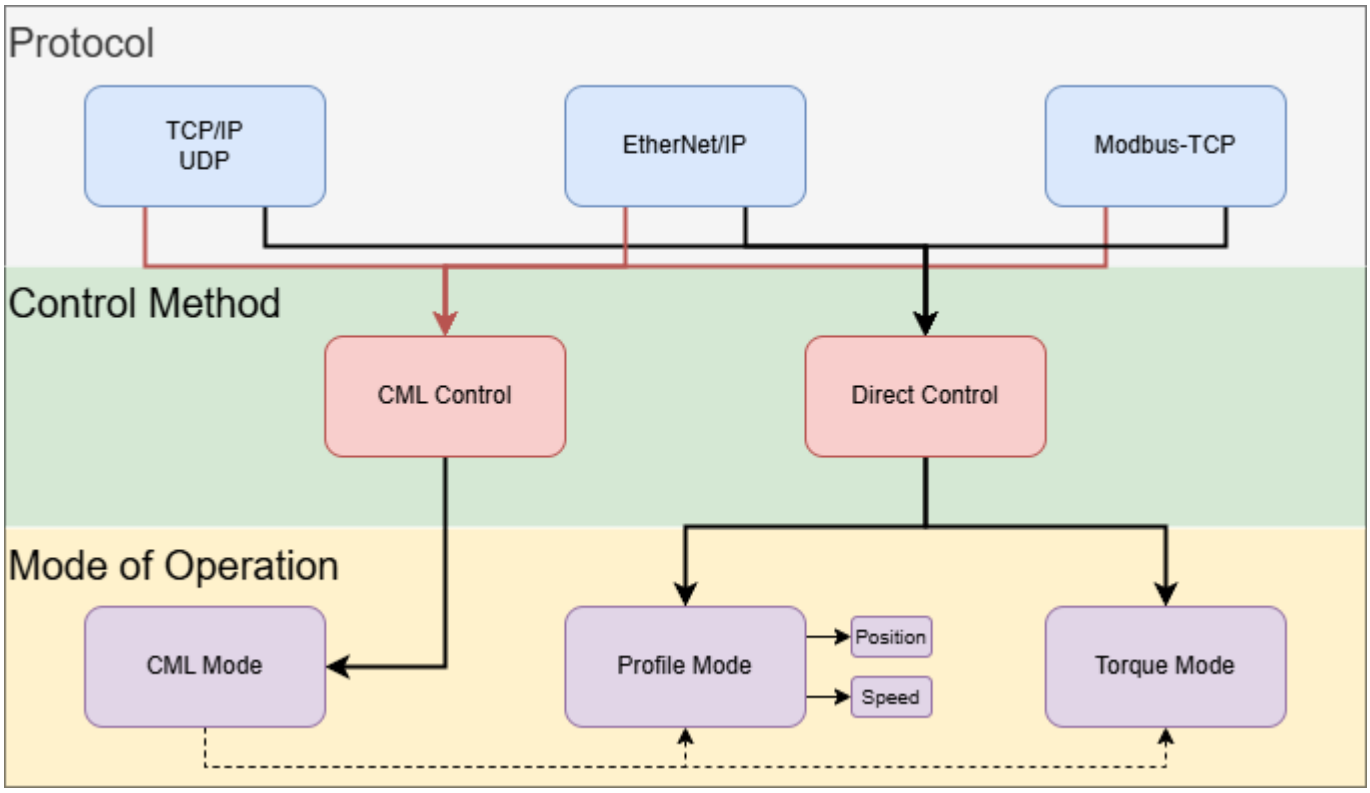
4.1 Introduction

The CM1-T is an Ethernet based Cool Muscle. It offers all the functionality of the standard Cool Muscle but enables connections over TCP/IP, UDP, Modbus-TCP and EtherNet/IP. This variant of the Cool Muscle cannot be daisy chained.

The following terminology is important to understand how to understand this document and how to operate the motor.

Term	Description
Protocols	The protocols are the different Ethernet languages the user can implement to communicate to the controls methods. They are • TCP/IP and UDP • Modbus-TCP • EtherNet/IP
Control method	The control method indicates how the motor controller is communicated with, updated and controlled. Control methods are • CML- the motor runs logic and program banks • Direct Control - the motor receives position, speed, acceleration/deceleration and torque values to dynamically move between targets
Mode of operation	The mode of operation is how the motor controller operates in Direct Control. This is how the motor's motion controller generates a set of outputs to the motor driver. These are typically • CML mode - traditional operation of the motor • Profile mode - operate the motor by in position and/or speed control • Torque mode - operate the motor by limiting torque and speed.
TCP/UDP	This refers to any port or protocol that can be connected using TCP/IP and/or UDP.

The following diagram shows how the protocols break down to the different modes of operation



4.2 Connections and Port

Any TCP/IP enabled client device can connect and communicate with the CM1-T. The table below lists the ports and protocols they are associated with

Protocol	Port #	# TCP	# UDP	Description
CML	10001	1	1	Protocol used by the standard motor. Program the motor to run from IO, etc.
Direct Control Port	10002	1	1	Directly control the motor in a number of modes such as position, speed and torque
Modbus-TCP	502	1	0	Modbus access to all motor registers including CML and Direct Control.
Motor Information Port	30718	0	1	1. Query the motors configuration (Network configuration, serial number, etc) 2. Qeury/Stream the motors information (status, position, speed, etc)

Protocol	Port #	# TCP	# UDP	Description
EtherNet/IP	44818 2222	2 0	1 1	EtherNet/IP CIP EtherNet/IP IO

Additional connection information can be found in Network Configuration.

4.3 General Usage and Operation

4.3.1 CML Programming

Refer to the CM1-C User Guide for details on using CML. You will find information on the following in this manual:

- K parameters (setup parameters)
- H parameters (postion controller gain parameters)
- Motor registers (position, speed, acceleration, torque, etc)
- Variable registers
- Program banks
- Logic banks
- Inputs and Outputs

As stated in the introduction the CM1-T is a standalone motor and cannot be used in a daisy-chain configuration. As such there are features in CML related to multiple motor networks which will not function if programmed.

Features that cannot be used in the CM1-T:

- Executing motion on another motor
    - I.e Program and logic banks can only reference motor ID 1.
- Reading input and setting outputs on another motor
- Coordinated motion with another motor

4.3.2 Modbus-TCP

Modbus-TCP reads and writes to all motor registers.

- CML Control - Typical usage would be executing logic and program banks with Modbus registers reading and/or writing to get motor information as well as intervening in operation as required.
- Direct Control - A set of registers can run the motor directly (see Direct Control).

4.3.3 EtherNet/IP

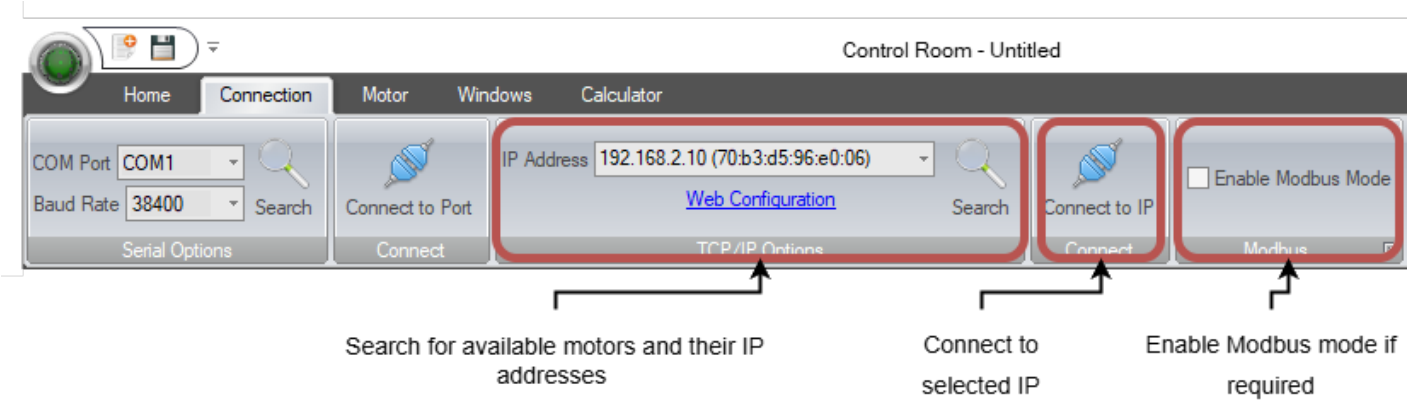
See the EtherNet/IP section of this document for details on using the CM1-T with EtherNet/IP

4.3.4 Connecting in Control Room

Control Room is the Cool Muscle Windows application to program the motors. It is typically not used for run-time operation but this can depend on the user's requirements. Control Room can open up a connection in either standard CML mode or by Modbus-TCP. Two instances of Control Room can be opened if the user needs to use Modbus-TCP and CML simultaneously.

Note: You cannot program in CML mode if the connection is opened in Modbus Mode and you cannot use the Modbus Window if the connection is opened not in Modbus Mode.

To connect to a motor use the TCP/IP options, Connect to IP and Enable Modbus Mode functions on the Connection tab as shown below.



Search Button

- If any motors are available they will be displayed by IP with their MAC address

Connect to IP Button

- Once the correct motor is selected in the IP Address drop down list click the Connect to IP button to connect to the motor

Enable Modbus Mode Checkbox

- If the motor is set to modbus check the Enable Modbus Mode before opening the connection. This will allow the Control Room modbus window to operate.

Note: The Web Configuration is not available on the CM1-T. Use the Network Configuration Tool to setup network related parameters.

5 Connectors and LED Information

5.1 Overview

The following diagram shows an overview of all connectors and LEDs.

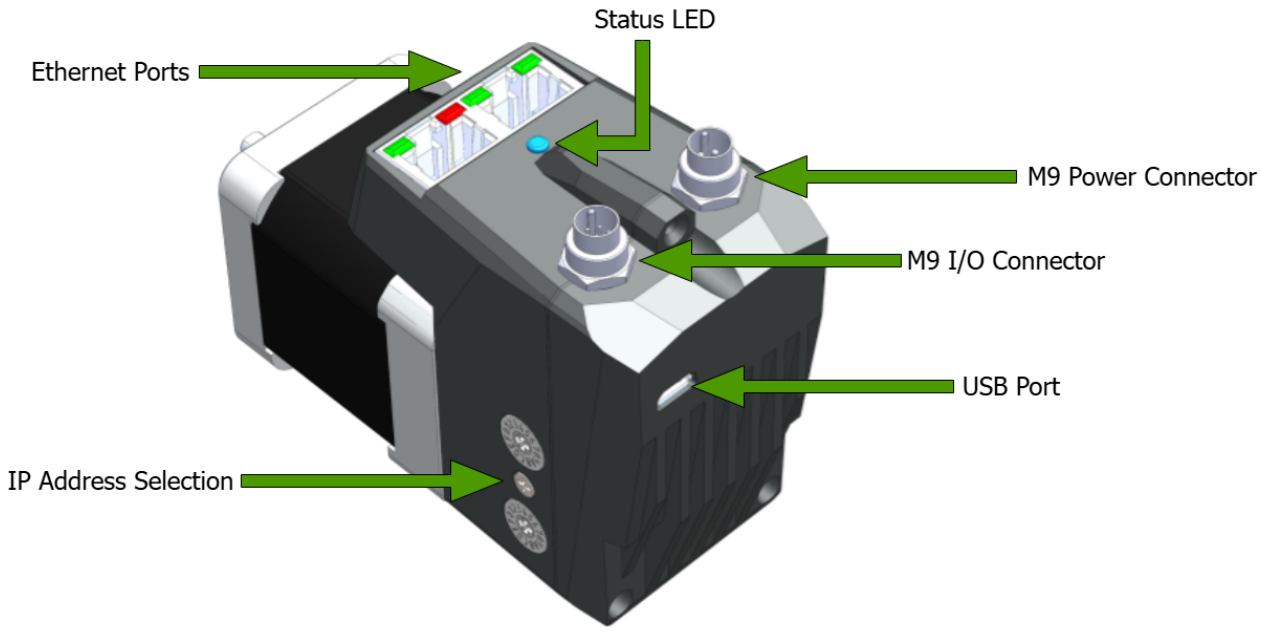
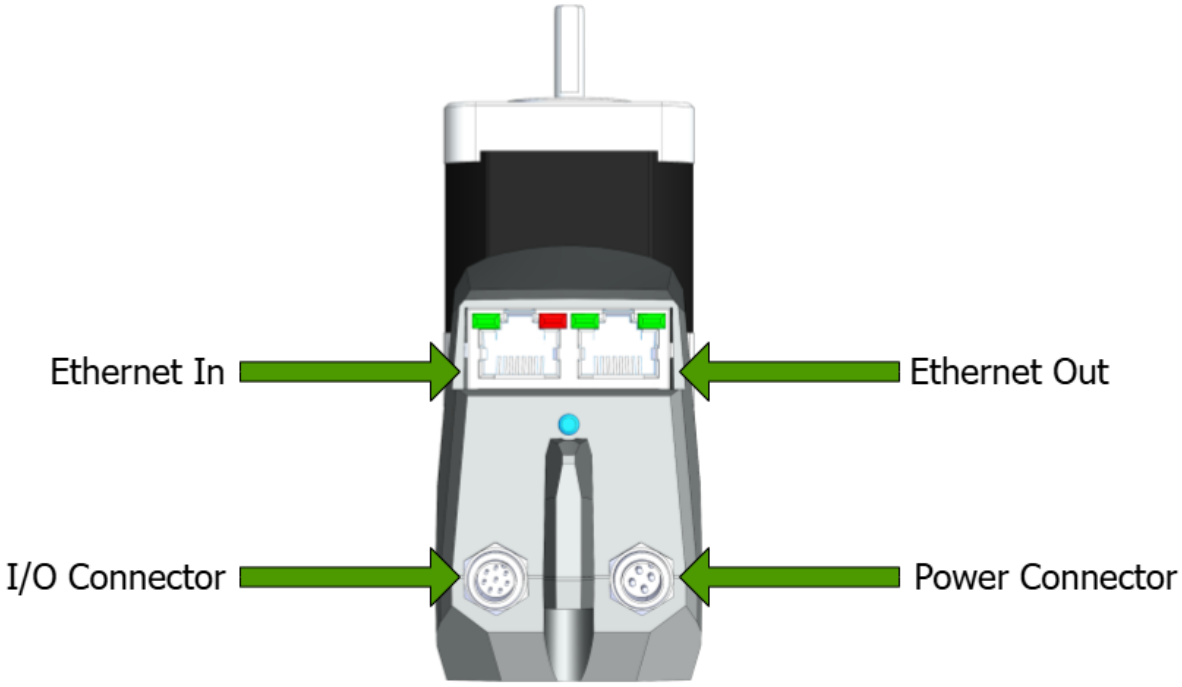


Figure: Overview of Connections and status LEDs

5.2 Motor Connections

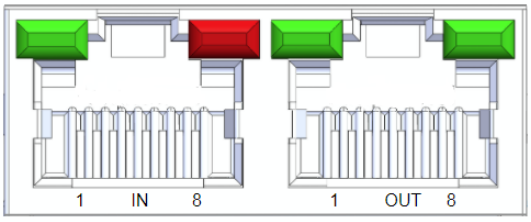


5.2.1 Ethernet Connectors

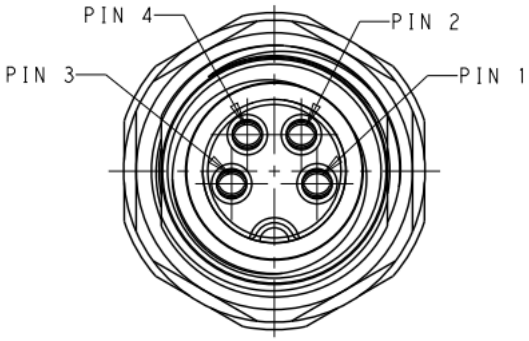
The Ethernet ports use standard Ethernet RJ45 CAT5e, M8-A or M8-D connectors depending on the motor variant. They are labeled IN and OUT but either port may be used. The CM1-T Ethernet connections will act as a TCP/IP switch. Multiple motors or other devices can be connected in a daisy chain configuration without the need to run back to a central switch for each ethernet device.

Connector Options

- RJ45



  - Amphenol - RJHSE538B02
- M8-A Female



  - TE Connectivity - T4041017041-000
- M8-D Female

Pinout

	RJ45[(1)]	M8-A	M8-D
Tx+	1	1	1
Rx+	3	2	2
Rx-	6	3	4
Tx-	2	4	3

(1) Pins 4,5,7 and 8 are connected to GND.

Example Cables

All ethernet cables are standard pinouts and are available from a variety of online suppliers or cable houses. Here are a few example cables from the Phoenix Contact NBC series

Image	Part Number	Description	Digikey Link
	1407353	M8-A male to RJ45 Phoenix Contact - 1407353	Digikey - 1407353
	1407349	M8-A male to M8A male Phoenix Contact - 1407349	Digikey - 1407349
	1227562	RJ45 to RJ45 Phoenix Contact - 1227562	Digikey - 1227562

5.2.2 Power Connector

The power connector supplies 24V to the TCP/IP slave and the motor separately. These two can be tied together so both are off the same power supply.

The power connector is an M9 circular connector from Binder. The relevant parts numbers are

Connector	Part Number	Supplier
Motor connector	09 0081 20 04	Binder
Female cable side mating connector	99 0080 102 04	Binder
4m power cable	CM1M9-4F-4000	Myostat

- The CM1M9-4F-4000 is 24AWG with conductor resistance of 97.5Ω/km
- The HF version with EXT-3D cable has a resistance of 91.1Ω/km

Pin #	Description	Voltage	Current
1 - yellow	Motor Control Power	24V ±10%	125mA max
2 - white	0V	-	
3 - grey	0V	-	
4 - orange	Motor Drive Power	24V ±10%	See individual motor ratings

*Colors indicated are for the standard **CM1M9-4F-4000** power cable.*

There is no reverse polarity protection. Ensure the 24V power is connected correctly before powering the unit.

Maintaining Control power and switching off motor drive power will
- retain motor position
- remove any ability for the motor to be driven (as power to the motor drive has been removed).

5.2.3 I/O Connector

Connecting a digital input to GND will produce a logical high on the device.

Connector	Part Number	Supplier
Motor connector	09 0481 22 08	Binder
Female cable side mating connector	99 0480 102 08	Binder
4m I/O cable	CM1M9-8F-4000	Myostat
Custom length I/O Cable	CM1M9-8F-xxxxS • xxxx = length in mm	Myostat

Pin Functions

Pin #	Name	Function	Specifications			
Digital Inputs - Sourcing (supply 0V to trigger)			**Parameter**	**Min**	**Max**	**Unit**
1 - orange 2 - brown 3 - green 4 - yellow	IN1 IN2 IN3 IN4	Digital input 1 Digital input 2 Digital input 3 Digital input 4	Voltage Range	0	36	V
			Input ON level	0	1.4	V
			Input OFF level	1.4	36	V
			Continuous Current	-	30	mA
			Peak Current	-	0.5	A
			Pulse Width	-	1	ms
Analog Input (0-5V)			**Parameter**	**Min**	**Max**	**Unit**
5 - purple	A-IN5	Analog Input	Voltage Range	0	5	V
			Resolution	10 bit		
Digital Outputs - Sinking (output supplies 0V when asserted)			**Parameter**	**Min**	**Max**	**Unit**
6 - blue 7 - black	OUT1 OUT2	Digital Output 1 Digital Output 2	Voltage Range	0	-	V
			Continuous Current	-	1	A
			Inductive Load Peak Rev Current	-	0.2	A
			Inductive Load Peak Rev Voltage	-	70	V
Signal Ground (0V)			**Parameter**	**Min**	**Max**	**Unit**
8 - red	0V	0V	Voltage Range	0	0	V

- *Colors indicated are for the standard **CM1M9-8F-4000** I/O cable.*
- *For custom cable length pin-out and colors see CM1M9-8F-xxxxS.PDF*

CM1M9-8F-4000 Specifications

Specification	Value
Length	4.0m
Wire Gauge	26 AWG
Colour Code	1 - orange 2 - brown 3 - green 4 - yellow 5 - purple 6 - blue 7 - black 8 - red

CM1M9-8F-xxxxS Specifications

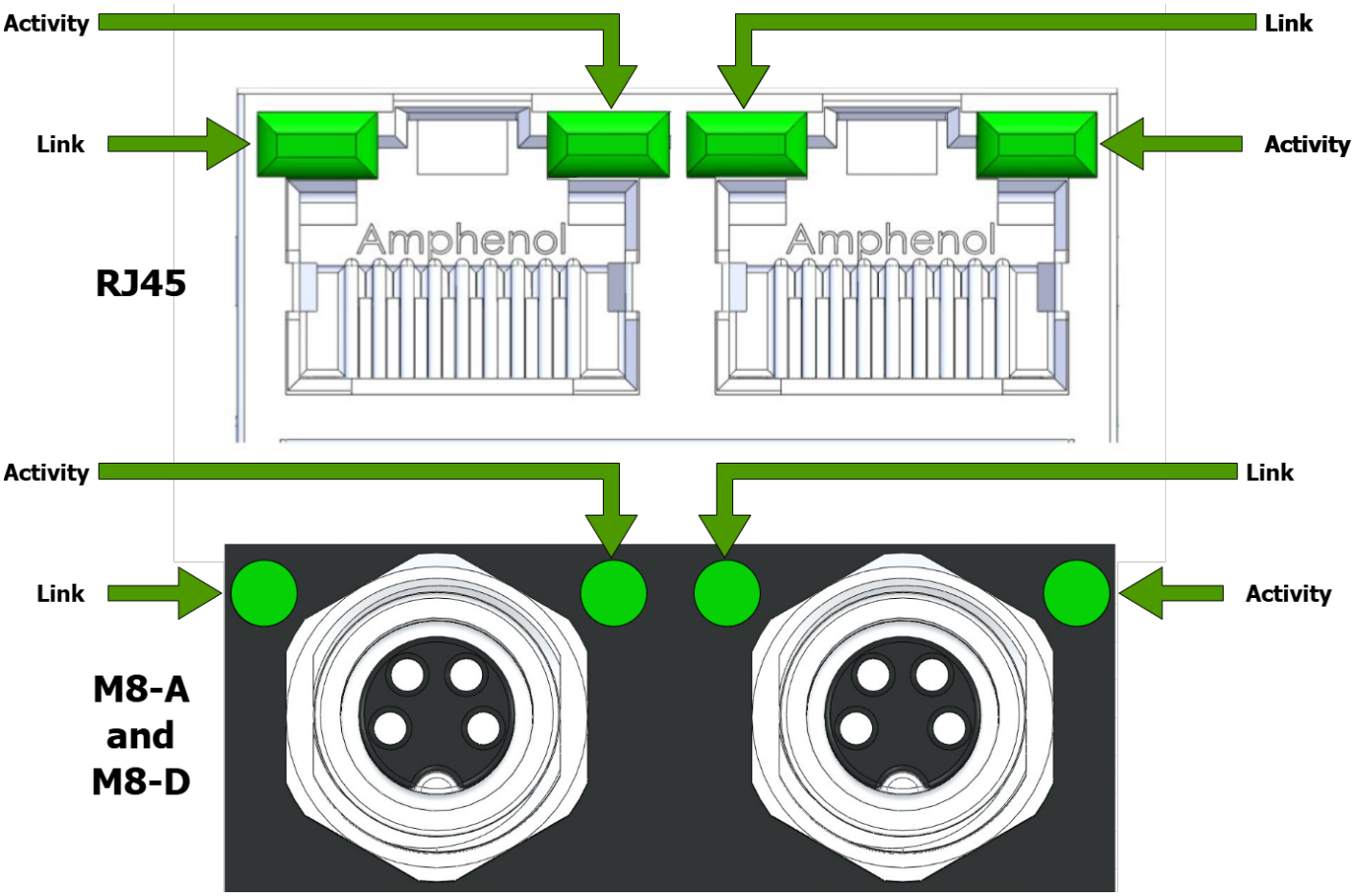
Specification	Value
Length	Defined by value in xxxx in millimeters (mm)
Wire Gauge	24 AWG
Colour Code	1 - orange/Black 2 - orange 3 - Grey/Black 4 - Grey/Red 5 - White/Black 6 - White/Red 7 - Yellow 8 - Yellow/Red
Drawing	CM1M9-8F-xxxxS.PDF

5.2.4 USB Connection

The USB connector is a standard micro USB and is used to update firmware. When it is plugged into a computer it will create a virtual serial port.

5.3 LEDs

5.3.1 Link and Activity LEDs



The Link LED will be lit as long as there is a network connection. The activity LED will blink when there is data transfer to/from the motor.

5.3.2 System and Motor Status LED

The system status LED provides feedback to the user on overall system status. The following states exist

GREEN	State	Resolution

Solid	No Error	N/A
Blinking	No IP Address given	• Investigate why no DHCP server found • Program a static IP
YELLOW	**State**	**Resolution**
Solid	Motor in disabled state (no torque)	Enable motor
RED	**State**	**Resolution**
Solid	24V motor drive power not present	Apply 24V drive power
Blinking	Module error	Unrecoverable - cycle power
Flashing	1 - Position error 2 - Over speed error 3 - Over temperature error 4 - Over torque error	Clear error/fault

5.3.3 IP Address Selector

The IP address selector switches will only be active if not using DHCP. If a static IP is programmed with the final value as 0, the rotary selectors will choose the final octet of the IP address.

For example:

Static IP	x10 Value	x1 Value	IP Address used
192.168.1.10	2	3	192.168.1.10
192.168.1.10	0	0	192.168.1.10
192.168.1.0	2	3	192.168.1.35
192.168.1.0	A	5	192.168.1.165
192.168.1.0	E	3	192.168.1.227


x10
ID
x1
Rotary
Selector
switches

6 Network Configuration

6.1 Introduction

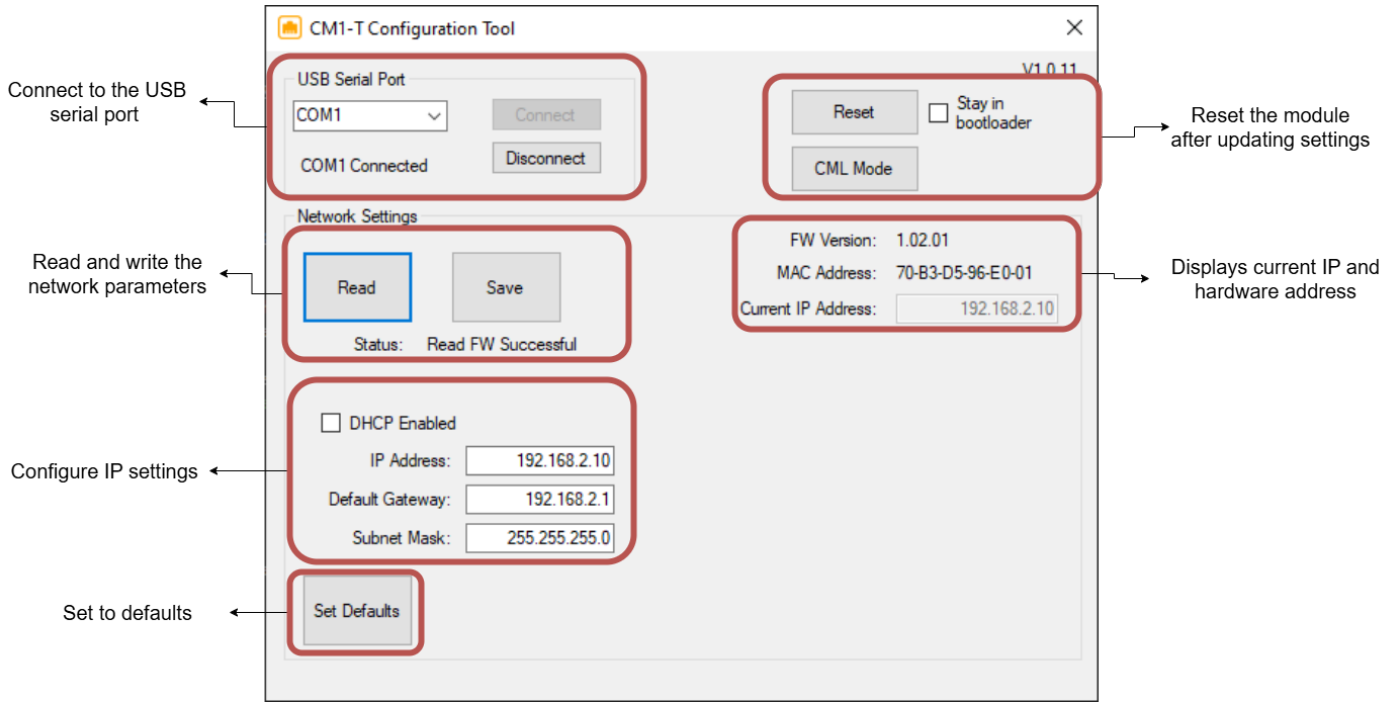
The network settings are configured through the CM1-T Configuration Tool application which connects to the USB on the back of the motor. This application also allows you to read the currently configured settings.

6.2 Parameter Table

The following network parameters are available in the tool:

Parameter	Read/Write	Description
Firmware Version	Read	Current CM1-T firmware version
MAC Address	Read	MAC address of the CM1-T
Current IP Address	Read	The IP address that the CM1-T is currently using
DHCP Enabled	Read/Write	Indicates whether the CM1-T should use DHCP or a static IP
IP Address	Read/Write	IP address if DHCP is not enabled
Default Gateway	Read/Write	Default gateway if DHCP is not enabled
Subnet Mask	Read/Write	Subnet mask if DHCP is not enabled

6.2.1 CM1-T Configuration Tool



6.3 Setting IP Properties

The IP on the Cool Muscle can be set as static or configured to use DHCP to assign the IP.

6.3.1 DHCP

The CM1-T must be on the same network as a DHCP server to use the DHCP option. The CM1-T will be assigned an IP automatically by the server. This IP can be seen in the "Current IP Address" box. In addition Control Room can search for the CM1-T in the Connection tab of Control room.

To set to DHCP:

1. Check the DHCP Enabled checkbox in the CM1-T Configuration Tool
2. Click the Save button
3. Power cycle or click the Reset button

The main status LED on the motor will flash green while it is waiting to be assigned an IP address. It will be solid green if an IP has been assigned.

THE DHCP search will not time out.

6.3.2 Static IP

The CM1-T can be assigned an IP address, default gateway and subnet mask. In addition the final octet in the IP can be set by the rotary switches on the side of the motor.

To set to static IP:

1. Uncheck the DHCP Enabled checkbox
2. Enter the required IP address
3. Enter the required gateway address
4. Enter the required Subnet mask
5. Click the Save button
6. Power cycle the motor or click the reset button

Using the Rotary Switches for assigning IP

The rotary switches can be used to assign the last value (octet) of the IP. Set the final number of the static IP to 0 to enable the rotary switches.

Examples:

Static IP	x10 Value	x1 Value	Final IP address assigned
192.168.1.10	2	3	192.168.1.10
192.168.1.10	0	0	192.168.1.10
192.168.1.0	2	3	192.168.1.35
192.168.1.0	A	5	192.168.1.165
192.168.1.0	E	3	192.168.1.227

7 Direct Control

7.1 Introduction

All the protocols can use the Direct Control method. Using Direct Control allows the user to directly control the motors position, torque and speed. All values can be updated dynamically which allows for very flexible operation. Not only does Direct Control allow you to control the motor but the read registers give a full array of motor information for program use and diagnostics.The CML control method has a special provision to run Direct Control by writing to specific CML registers. The documentation below describes how to use Direct Control. For details on each of the protocols please see their specific sections

- Modbus-TCP
- Direct Control Port
- EtherNet/IP - EtherNet/IP includes AOIs that setup all required mappings.
- CML - The CML registers used are detailed below in the Direct Control documentation

Though all the protocols utilize the same set of variables they can work concurrently. If multiple protocols are to be used concurrently then the user should be aware of how the write operations function:

- EtherNet/IP - Cyclical (1ms)
- Modbus-TCP - On write
- TCP/UDP - On write

As can be seen above, if EtherNet/IP is used it will overwrite any of the other protocols in a write operation. The other protocols can still be used for read operations and diagnostics.

7.1.1 Read Registers

Register	Description
CPUTime	CPU clock time in 50us counts
ActualPosition	The actual position of the motor in encoder counts
ActualTargetPosition	The actual instantaneous position the motor is currently moving to.
MotorStatus	The motors status (error, homing, in position, etc)
ActualRatedCurrent	Percentage (0.1%) of rated current. 1000 = rated, 1100 = peak.
ActualOverloadTorque	Percentage (0.1%) of overload torque.
AnalogIN	10 bit analog input value (0-1023)
DigitalIO	Digital IO status. • B3-B0 = IN4-IN1 • B5-B4 = OUT2 - OUT1

Register	Description
Temperature	Drive temperature in °C
DCVoltage	24V DC bus voltage in 0.1V
DigitalOUT	Digital OUT status. B0-B1 = OUT1-OUT2
ModeOfOperationDisplay	Indicates the mode of operation currently set
ActualSpeed	Actual speed of the motor in encoder counts/s

7.1.2 Write Registers

Register	Description
TargetPosition	Final target position
TargetSpeed	Maximum speed
TargetTorque	Maximum torque
TargetAcceleration	Acceleration (used when accelerating to target speed)
TargetDeceleration	Deceleration (used when stopping)
Controlword	Control the motor operation
ModeOfOperation	Set the required mode of operation
DigitalOUT	Set the 2 digital outputs. Requires K34=44.

7.2 Setting Mode Of Operation

There are a number of modes of operation the motor can be set to.

1. CML mode - this is the standard and default mode. If no mode is selected, this is the mode the motor Cool Muscle operates in.
2. Profile mode - run position or speed profiles.
3. Torque mode - run the motor by adjusting the torque and maximum speed values.

The control registers in the mode selected use either standard CML registers or registers set directly such as through Modbus-TCP. The mode of operation selected indicates where the control registers will be received

The following table lists the mode of operation and the associated registers

Mode of Operation	Value	Description
CML Mode	0	Standard CML mode. No control registers associated
Profile Mode	2	Profile mode using P1, S1, A1, A2, R1 and N1
	3	Profile mode using • Modbus control registers (see registers starting 41029). • EtherNet/IP IO messaging • TCP/UDP port 10002
Dynamic Position Mode	4	Dynamic position mode using P1, A1, R1 and N1
	5	Dynamic position mode using • Modbus control registers (see registers starting 41029). • EtherNet/IP IO messaging • TCP/UDP port 10002
Torque Mode	10	Torque mode using N1, S1 and R1
	11	Torque mode using • Modbus control registers (see registers starting 41029). • EtherNet/IP IO messaging • TCP/UDP port 10002

- Bit0 of the mode of operation selects the register source. I.e. P1 to N1 or the direct control registers.

Set the mode of operation by the following

1. CML
    a. Send "_mop=x" on the CML port when x is the value of the mode of operation
    b. Assign a variable to "mop" and write the variable in a logic bank. E.g. V0="mop".
2. Direct
    a. Set modbus register 41041 to the value of the required mode of operation.
    b. Set Byte[16] in EtherNet/IP
    c. Set Byte[16] in TCP/UDP Port 10002

7.3 Registers

The following registers are used in the different Direct Control modes of operation.

7.3.1 Controlword

The Controlword is used to control the motor. The usage does depend on the mode that is selected. See the mode for bit usage. B4-B7 are common to all modes.

Register Name	Profile Mode = 2	Profile Mode = 3			Unit	Description
	CML Registers	Modbus Registers	TCP/UPD Bytes	Ethernet/IP O2T Bytes		
Controlword	R1	41039	14-15	14-15	N/A	See below

Bit Descriptions:

BIT	Name	Value	Description
B0	Operation mode specific		
B1	Operation mode specific		
B2	Operation mode specific		
B3	Operation mode specific		
B4	Disable	0 1 1 → 0	Enable the motor Disable the motor Reset any error
B5	Reset Error	0 → 1	Reset any error
B6	Set to Zero	0 → 1	Set the current position to 0
B7	Home	0 → 1	Start a home search(1)

(1) Set the home search parameters with K parameters K42-K48

7.3.2 Statusword

The Statusword returns information on the status of the motor.

Register Name	Profile Mode = 2	Profile Mode = 3			Unit	Description
	CML Registers	Modbus Registers	TCP/UPD Bytes	Ethernet/IP O2T Bytes		
Statusword	Query ?99	40009	16-17	16-17	N/A	See below

Bit Descriptions:

BIT	Name
B0	Position overflow alarm
B1	Overspeed alarm
B2	Overload alarm
B3	Inposition
B4	Disable
B5	Pushmode limit
B6	Communication error alarm
B7	Over temperature alarm
B8	Pushmode Timeout warning
B9	E-Stop active
B10	N/A
B11	N/A
B12	N/A
B13	N/A
B14	New start required
B15	Home achieved

7.3.3 TargetPosition

The TargetPosition sets a new target position for the motor.

Register Name	Profile Mode = 2	Profile Mode = 3			Unit	Range
	CML Registers	Modbus Registers	TCP/UPD Bytes	Ethernet/IP O2T Bytes		
TargetPosition	P1	41029	0-3	0-3	pulses (counts)	$-2^{\wedge 31}$ to $+2^{\wedge 31}$

The motor has a maximum resolution of 50,000 counts per revolution. Regardless of the resolution defined in K37 the motor internally will convert the target position to a position based on 50,000. As such the maximum range depends on the motor

resolution. For example if K37=3 and the resolution is 1000 pulses/revolution, the maximum target position is $2^{\wedge 31} \div (50{,}000/1{,}000) = 2^{\wedge 31} \div 50 = 42{,}949{,}672$.

7.3.4 TargetSpeed

The TargetSpeed sets a new maximum speed.

Register Name	Profile Mode = 2	Profile Mode = 3			Unit	Range
	CML Registers	Modbus Registers	TCP/UPD Bytes	Ethernet/IP O2T Bytes		
TargetSpeed	S1	41031	4-7	4-7	pulses/ second 10 pulses/ second 100 pulses/ second	$-2^{\wedge 31}$ to $+2^{\wedge 31}$

The unit depends on the motor resolution (K37) value selected. See K37 for more details.

7.3.5 TargetTorque

The TargetTorque sets the maximum torque used by the motor.

Register Name	Profile Mode = 2	Profile Mode = 3			Unit	Range
	CML Registers	Modbus Registers	TCP/UPD Bytes	Ethernet/IP O2T Bytes		
TargetTorque	N1	41033	8-9	8-9	0.1% rated torque	0-1100 (±1100 in Torque mode)

The torque is set in a percentage of rated torque. The Cool Muscle will only use the torque required so typically this value should be set to 1100 unless the application requires that the torque is limited.

7.3.6 TargetAcceleration

The TargetAcceleration sets the acceleration used by the motion profile.

Register Name	Profile Mode = 2	Profile Mode = 3			Unit	Range
	CML Registers	Modbus Registers	TCP/UPD Bytes	Ethernet/IP IO Bytes		
TargetAcceleration	A1	41035	10-11	10-11	Kpulses/ second^2	± 32767

If the TargetAcceleration=0 the profile shall be undefined and may result in an unexpected motion profile.

7.3.7 TargetDeceleration

Register Name	Profile Mode = 2	Profile Mode = 3			Unit	Range
	CML Registers	Modbus Registers	TCP/UPD Bytes	Ethernet/IP O2T Bytes		
TargetDeceleration	A2	41037	12-13	12-13	Kpulses/ second^2	± 32767

If the TargetDeceleration=0 the TargetAcceleration shall be used. This allows the user to only change the acceleration if a trapezoid type profile is required.

7.4 Modes of Operation

7.4.1 CML Mode

CML mode is the standard mode that the Cool Muscle motor typically operates in. There are no specific control registers. See the CM1 documentation for CML mode usage.

7.4.2 Profile Mode (Position and Speed)

Profile mode would be a typical mode used in point-to-point motion and/or continuous velocity motion. The following registers are used depending on the profile mode selected.

Register Name	Profile Mode = 2	Profile Mode = 3			Description
	CML Registers	Modbus Registers[1]	TCP/UPD Bytes[2]	Ethernet/IP O2T Bytes[3]	
TargetPosition	P1	41029	0-3	0-3	The target position the motor will move to. • Used as an absolute position • Used as an incremental position in a relative control[1] move • Not used in speed control[1] (1) See the Controlword for switching to speed or relative control.
TargetSpeed	S1	41031	4-7	4-7	The maximum speed of the profile trajectory. • the sign of S1 is ignored in a position control move. • +ve S1 in speed control will run the motor in a +ve direction (typically CW) • -ve S1 in speed control will run the motor in a -ve direction (typically CCW)
TargetTorque	N1	41033	8-9	8-9	The maximum torque the motor will use.
TargetAcceleration	A1	41035	10-11	10-11	Acceleration is used when the profile is accelerating to the target speed. This is regardless of whether the target speed will be reached or not
TargetDeceleration	A2	41037	12-13	12-13	Deceleration is used when the profile is decelerating to the target position.
Controlword	R1	41039	14-15	14-15	Bits in the Controlword dictate profile execution. A more detailed description can be found below.
ModeOfOperation	Set with "_mop"	41041	16	16	Sets the mode of operation see Direct Control#Setting Mode of Operation
DigitalOUT	N/A	41043	17	17	Binary value to set the digital output (Register K34=44 to enable this function)

(1) See Modbus-TCP for additional details
(2) See Direct Control#TCP/UDP Port 10002 for additional details
(3) See EtherNet/IP for additional details. EtherNet/IP includes AOIs that setup all required mappings.

Controlword Usage

Profile mode uses some specific bits in the Controlword

BIT	Name	Value	Description
B0	Start/New Set Point	0	• Do not execute any changes in profile registers • Do not start a new profile • Any profile currently in motion will be completed

		0 → 1	• Start a profile move defined by the profile registers
			• 0 → 1 transition is required to start a profile after
			  • an error has been cleared
			  • a home has been completed
			  • a stop command has been given from CML or an input.
			• In relative mode transition is required to start the next incremental move.
			  • The move is incremented from the current target position not the actual position.

		1	Execute any changes in any of the profile registers immediately.
B1	Halt	0	Do nothing
		1	• Halt the current profile being executed • Prohibit execution of a profile if no profile is being run
B2	Relative mode	0	TargetPosition is an absolute position
		1	TargetPosition is a relative incremental position
B3	Speed control	0	Position control mode
		1	Speed control mode
B7-B4	See Controlword		

Profile Mode Examples

The following example uses the standard registers to run an absolute move. This could be run from a Control Room CML script.

Absolute Move Example

```
R1=0        //clear the controlword before setting the mode
_mop=2      //set the mode of operation to 2. Profile mode
P1=10000    //set target position
S1=100      //set target speed
A1=10       //set target acceleration
A2=10       //set target deceleration
N1=1100     //set the target torque to peak
R1=1        //start the profile move

//While R1=1 any change in the profile registers will execute an immediate change
```

The following example uses the standard registers to run a speed move. This could be run from a Control Room CML script.

Speed Move Example

```
R1=0        //clear the controlword before setting the mode
_mop=2      //set the mode of operation to 2. Profile mode
S1=100      //set target speed
A1=10       //set target acceleration
A2=10       //set target deceleration
N1=1100     //set the target torque to peak
R1=9        //start the profile move with the speed mode bit set.

//While R1=9 any change in the profile registers will execute an immediate change
```

Speed Move Example using Logic and Inputs

```
/* Two Inputs - Two Speeds

This example will run the motor in a speed control mode.
 * The speed selected will be based on which input is active
 * The speed in this example switches motor direction between CW and CCW
 * If both inputs are ON or both inputs are OFF the motor will stop

IN3 - set speed to 20
IN4 - set speed to -20

The example could be extended for other applications
```

```
 1) by setting 4 different speeds based on the binary combination of the inputs
 2) Having the speeds not change direction but adjust speed only. This could be used on something like
a conveyor

------------
This example uses direct control.
See https://docs.myostat.ca/cm1-t/user-guide/direct-control#DirectControl-ProfileMode(PositionandSpeed)
------------
This example uses the IN and INFL variables to easily read the input status
IN --> outputs the binary combination of inputs
INMK --> sets and mask on IN to only return inputs required.
See https://docs.myostat.ca/cm1-c/user-guide/internal-state-variables#InternalStateVariables-"IN"
------------
*/

K85=1       //Start L1 on power up
K87=1       //Logic scan time = 1ms


var ModeOfOperation V1  //name V1 as ModeOfOperation
V1="mop"                //assign to internal mop variable

var INMask V2
V2="INMK"

var INValue V3
V3="IN"

V4=0

B100        //clear all program banks
L100        //clear all logic banks

//L1 will be used to init all variables
L1.1
R1=0        //clear the controlword before setting the mode
INMask = 12 //Mask only IN3 and IN4 (IN4=8, IN3=4, IN2=2, IN1=1 -> bit combination)
ModeOfOperation=2       //set the mode of operation to 2. Profile mode
S1=0        //init target speed
A1=10       //init target acceleration
A2=10       //init target deceleration
N1=1100     //init the target torque to peak (110% of rated)
R1=9        //start the profile move with the speed mode bit set.
JL2         //jump to L2
END.1

//L2 will monitor the input status and change speed accordingly
L2.1
```

```
S2=0;
INValue == 4, S2=20, T0
INValue == 8, S2=-20, T0
S1=S2;      //set direct control speed
END.1

$.1
```

7.4.3 Dynamic Position Mode

This mode can be used to dynamically stream a target position only and have the motor track it. There is a filter that filters the responsiveness of the position change as no speed or acceleration is defined. For best performance try used a fixed time interval on the position update.

Register Name	Profile Mode = 4	Profile Mode = 5			Description
	CML Registers	Modbus Registers(1)	TCP/UPD Bytes(2)	Ethernet/IP O2T Bytes(3)	
TargetPosition	P1	41029	0-3	0-3	The target position the motor will move to. • Used as an absolute position
TargetTorque	N1	41033	8-9	8-9	The maximum torque the motor will use.
Filter Gain	A1	41035	10-11	10-11	The A1 register is used to store the filter gain.
Controlword	R1	41039	14-15	14-15	Bits in the Controlword dictate the execution. A more detailed description can be found below.
ModeOfOperation	Set with "_mop"	41041	16	16	Sets the mode of operation see Direct Control#Setting Mode of Operation
DigitalOUT	N/A	41043	17	17	Binary value to set the digital output (Register K34=44 to enable this function)

(1) See Modbus-TCP for additional details
(2) See Direct Control#TCP/UDP Port 10002 for additional details
(3) See EtherNet/IP for additional details. EtherNet/IP includes AOIs that setup all required mappings.

Controlword Usage

Profile mode uses some specific bits in the Controlword

BIT	Name	Value	Description
B0	Start/New Set Point	0	• Do not execute any changes in target position register • Do not start a new mode • Any active move will be completed

		0 → 1	• Start a profile move defined by the profile registers
			• 0 → 1 transition is required to start a profile after
			  • an error has been cleared
			  • a home has been completed
			  • a stop command has been given from CML or an input.

		1	Execute any changes in any of the profile registers immediately.
B1	Halt	0	Do nothing
		1	• Halt the tracking position executed • Prohibit execution of a profile if no profile is being run
B3-B2	N/A		
B7-B4	See Controlword		

Dynamic Position Mode Example

The following example uses the standard registers to continuously update the motor position

Dynamic Position Mode Example

```
K87=10
V1="mop"    //assign V1 to set the mode of operation

a1=10       //set the filter gain to 10 (1-1024)
p1=0        //target position
p2=10       //amount to increment each iteration of the logic scan
N1=1100     //run motor at full torque

|2          //reset position to 0


L1.1        //Logic bank 1 to init all parameters
R1=0        //clear control word
V1=4        //set mode of operation to 4
|2          //set motor position to 0
P1=0        //set target position to 0
R1=1        //set control word to start running dynamic position
```

```
JL2.1       //jump to L2
END.1

L2.1        //L2 will continuously loop every #ms set in K87.
p1=p1+p2;   //increment target position by P2
end.1
```

7.4.4 Torque Mode

Torque mode would be used where there is no target position but the motor is required to continuously hold or move at a required maximum torque. In this mode the toque set is a maximum torque and the torque will only be reached if required.

Register Name	Profile Mode = 10	Profile Mode = 11			Description
	CML Registers	Modbus Registers(1)	TCP/UPD Bytes(2)	Ethernet/IP O2T Bytes(3)	
TargetSpeed	S1	41031	4-7	4-7	The maximum speed of the profile
TargetTorque	N1	41033	8-9	8-9	The maximum torque the motor will use while moving. • +ve N1 will rotate the motor in a +ve direction (typically CW) • -ve N1 will rotate the motor in a -ve direction (typically CCW)
Controlword	R1	41039	14-15	14-15	Bits in the Controlword dictate torque execution. A more detailed description can be found below.
ModeOfOperation	Set with "mop"	41041	16	16	Sets the mode of operation see Direct Control#Setting Mode of Operation
DigitalOUT	N/A	41043	17	17	Binary value to set the digital output (Register K34=44 to enable this function)

(1) See Modbus-TCP for additional details
(2) See Direct Control#TCP/UDP Port 10002 for additional details
(3) See EtherNet/IP for additional details. EtherNet/IP includes AOIs that setup all required mappings.

Controlword Usage

Torque mode uses some specific bits in the Controlword

BIT	Name	Value	Description
B0	Start/New Set Point	0	• Do not follow the N1 target torque • Target Speed S1 is still executed • Do not start a new torque move

		0 → 1	• Start a torque move defined by the torque registers
			• 0 → 1 transition is required to start a torque move after
			  • an error has been cleared
			  • a home has been completed
			  • a stop command has been given from CML or an input.

		1	Following N1 target torque
B1	Halt	0	Do nothing
		1	• Halt all motion and inhibit further motion • Halt all torque related parameters (torque & speed)
B3-B2	N/A		
B7-B4	See Controlword		

The interaction of the halt bit and the new set point (nsp) bit is quite specific. See the table below for additional important interaction.

B0 (nsp)	B1 (Halt)	Description
1	0	Motion profile follows torque and speed values
1	0 → 1	All motion stops
1	1	Motion is inhibited • Changes in speed are NOT followed • Changes in torque are NOT followed
0	1	Operation is the same as above.
0	0	Motion may resume based on values • Torque will NOT be updated • Motion will continue if speed > 0

It is important to understand that the halt bit temporarily halts operation. It does not interrupt motion. This means that a rising edge (0 → 1) is not required on B0 to continue operation. To keep prohibiting motion the following could be use

- Keep the halt bit at 1
- Set target velocity = 0
- Move out of torque mode

Torque Mode Examples

The following example uses the standard registers to run a torque move. This could be run from a Control Room CML script.

Torque Mode Example

```
R1=0        //clear the controlword before setting the mode
_mop=10     //set the mode of operation to 10. Torque mode
S1=100      //set target speed
N1=300      //set the target torque to 30.0% of rated
R1=1        //start the profile move

//While R1=1 any change in the torque register will execute an immediate change
```

8 EtherNet/IP

8.1 Introduction

This section explains how to use the CM1-T Cool Muscle motor with EtherNet/IP. It is assumed there is a basic knowledge of EtherNet/IP and PLC programming. The implementation makes use of the different Direct Control in the motor. It does not use CML logic and bank programming. However, setup (K) and controller gain (H) parameters are used for things like motor resolution, s-curve, software limits, etc. These are all accessible to be setup with Control Room or over explicit messages.

8.1.1 Device Profile

The CM1-T uses a generic device profile. This is to allow it to be flexible and operate in a number of operation modes.

Direct Control Operation

This mode enables the PLC to directly control the motor. See Direct Control on how to use the I/O messages to control the motor. Add on instructions are supplied in the downloads section.

CML Operation

This mode lets to motor control itself through CML programming. The PLC reads and writes to registers to monitor and make changes how the motor operates.

8.2 I/O Messages

There are two distinct I/O Assemblies based on the operation mode required.

8.2.1 Direct Control Operation

I/O	Assembly Instance	Size
Input	100	36
Output	112	28
Configuration	1	0

T2O (Input) Message

- Instance = 100

- Size = 36 bytes

Byte	Name	Data Type
0-3	us50Counter	DINT
4-7	ActualPosition	DINT
8-11	MotionTarget	DINT
12-15	ActualSpeed	DINT
16-17	MotorStatus	INT
18-19	PercentActualCurrent	INT
20-21	PercentOverloadTorque	INT
22-23	AnalogIN	INT
24-25	DCVoltage	INT
26	DigitalIO	SINT
27	Temperature	SINT
28	ModeOfOperationDisplay	SINT
29	ErrorCode(1)	SINT
30-31	ReadAddress1	INT
32-35	ReadValue1	DINT

(1) - ErrorCode is currently reserved and returns 0. Use MotorStatus to get motor error information

O2T (Output) Message

- Instance = 112
- Size = 28

Byte	Name	Data Type
0-3	TargetPosition	DINT
4-7	TargetSpeed	DINT
8-9	TargetTorque	INT

Byte	Name	Data Type
10-11	TargetAccelertion	INT
12-13	TargetDeceleration	INT
14-15	Controlword	INT
16	ModeOfOperation	SINT
17	DigitalOUT	SINT
18-19	WriteAddress1	INT
20-23	WriteValue1	DINT
24-25	ReadAddress1	INT
26-27	PAD	INT

8.2.2 CML Operation

I/O	Assembly Instance	Size
Input	101	60
Output	113	44
Configuration	1	0

T2O (Input) Message

- Instance = 101
- Size = 60 bytes

Byte	Name	Data Type
0-3	us50Counter	DINT
4-7	ActualPosition	DINT
8-11	MotionTarget	DINT
12-15	ActualSpeed	DINT
16-17	MotorStatus	INT

Byte	Name	Data Type
18-19	PercentActualCurrent	INT
20-21	PercentOverloadTorque	INT
22-23	AnalogIN	INT
24-25	DCVoltage	INT
26	DigitalIO	SINT
27	Temperature	SINT
28	ModeOfOperationDisplay	SINT
29	ErrorCode(1)	SINT
30-31	ReadAddress1	INT
32-35	ReadValue1	DINT
36-37	ReadAddress2	INT
38-39	ReadAddress3	INT
40-41	ReadAddress4	INT
42-43	ReadAddress5	INT
44-47	ReadValue2	DINT
48-51	RadValue3	DINT
52-55	ReadValue4	DINT
56-59	ReadValue5	DINT

(1) - ErrorCode is currently reserved and returns 0. Use MotorStatus to get motor error information

O2T (Output) Message

- Instance = 113
- Size = 44

Byte	Name	Data Type
0-3	WriteValue1	DINT

Byte	Name	Data Type
4-7	WriteValue2	DINT
8-11	WriteValue3	DINT
12-15	WriteValue4	DINT
16-19	WriteValue5	DINT
20-21	WriteAddress1	INT
22-23	WriteAddress2	INT
24-25	WriteAddress3	INT
26-27	WriteAddress4	INT
28-29	WriteAddress5	INT
30-31	ReadAddress1	INT
32-33	ReadAddress2	INT
34-35	ReadAddress3	INT
36-37	ReadAddress4	INT
38-39	ReadAddress5	INT
40	DigitalOUT	SINT
41	PAD	SINT
42-43	PAD	INT

8.2.3 Read/Write Motor Registers

In the I/O messages there is the ability to read and write to internal motor registers. These registers could be configured as flags, input counters, or other custom operations.

1. Direct Operation has
    - 1 x read register
    - 1 x write register
2. CML Operation has
    - 5 x read registers
    - 5 x write registers

The register addresses can be changed after a cycle to read/write to more than 1 register.

Register Address

The register address is coded into a 2 byte (UINT16) format as follows

B1	B0
Register Type (P, S, A, V, etc)	Register Number (0 to max)

Register Address = Register Type Register Number

The following list maps the register type and the applicable number range

B1 Value (hex)	Register	Register No. Range	Example		
			Register	Address	
				Hex	Decimal
16#00	N/A	N/A	-	-	-
16#01	reserved	reserved	-	-	-
16#02	reserved	reserved	-	-	-
16#03	P	0-25	P0	16#0300	768
16#04	S	0-15	S1	16#0x0401	1025
16#05	A	0-8	A5	16#0x0505	1285
16#06	T	0-8	T2	16#0x0602	1538
16#07	M	0-8	M0	16#0x0700	1792
16#08	R	0-25	R16	16#0x0810	2064
16#09	N	0-25	N15	16#0x090F	2319
16#0A	V	0-15	V10	16#0x0A0A	2570

For a complete list go to EtherNet/IP#Complete Register Address Map

8.3 Explicit Messages

8.3.1 K Parameters - Setup Parameters

- Get Attribute Single - Service Code 0x0E

- Set Attribute Single - Service Code 0x10

Class	0x64
Instance	1
Attribute	The attribute represents the K number. example: K37 → 37_d → 0x25

8.3.2 H Parameters - Gain Parameters

- Get Attribute Single - Service Code 0x0E
- Set Attribute Single - Service Code 0x10

Class	0x65
Instance	1
Attribute	H0 → 0x01 H1 → 0x02 H2 → 0x03 H3 → 0x04 H4 → 0x05 H5 → 0x06 H6 → 0x07 H7 → 0x08

8.4 Direct Control Operation

8.4.1 Control Modes

This section focuses on controlling the motor with the provided Add on Instructions (AOIs). These will satisfy most applications, however, if required, the motor can be controlled by setting the CM1-T objects directly (see I/O Messages). Understanding the Direct Control will be important in this case.

There are currently 3 main control modes available while running the motor over EtherNet/IP.

1. Position - this mode sets the motor to run to a target position. The position could be absolute or relative
2. Speed - this mode allows the motors speed to be controlled for continuous motion
3. Torque - this mode allows the motors torque to be controlled.

The control modes are mapped to the ModeOfOperation value from the CM1T_ControlMode AOI:

Control Mode	CM1T_ControlMode AOI value	O2T ModeOfOperation Value
Position	0	3

Control Mode	CM1T_ControlMode AOI value	O2T ModeOfOperation Value
Speed	1	3
Torque	2	11

In addition:

- Homing is an additional mode that can be executed at any point. Homing should start from a known safe state to avoid unexpected changes in motion
- All modes can be switched to during motion. It is recommended the state is switched during a known safe state.
- Torque can be limited in all modes. Running in torque mode will disable errors such as position error which can still occur in other modes if the set torque is too low. The torque is set in 0.1% of rated torque. The range is [0,1100]. By default the torque is set to 1100 when using the AOIs supplied.

⚠ All modes use common objects such as position, speed, torque, acceleration and controlword. Switching parameters in another mode will change the value in the existing mode.

✖ Setting the Start bit to 0 will NOT halt motion. The HALT bit must be used to interrupt motion.

Position Mode

Absolute position and relative position both use the position mode. The key difference in operation between the two is the usage of the start bit in the CM1T_Controlword. An absolute move will accept a new position immediately if the start bit is 1.

Start Bit Usage

BIT	Name	Value	Description
B0	Start/New Set Point	0	• Do not execute any changes in profile registers • Do not start a new profile • Any profile currently in motion will be completed

		0 → 1	• Start a profile move defined by the profile registers
			• 0 → 1 transition is required to start a profile after
			  • an error has been cleared
			  • a home has been completed
			  • a stop command has been given from CML or an input.
			• In relative mode transition is required to start the next incremental move.
			  • The move is incremented from the current 1ms target position not the actual position.
			  • This will be the same if there is no position error
			  • This is to account for possible drift in a repetitive motion.

		1	Execute any changes in any of the profile registers immediately.

Motion Parameters

Position moves will accept the following motion parameters. These will be processed immediately if the start bit is high.

Name	Unit	Data Type	Access
PositionTarget	counts	DINT	AOI(2) or Axis tag
SpeedTarget	counts/s	DINT	AOI or Axis tag
AccelerationTarget	1000 counts/s^2	INT	AOI or Axis tag
DecelerationTarget(1)	1000 counts/s^2	INT	AOI or Axis tag
TorqueTarget	0.1% rated torque	INT	Axis Tag

(1) If the deceleration is set to 0 the acceleration value will be used
(2) MoveAbsolute or MoveRelative AOI

Speed Mode

A speed move has no target position and will continue to run at the target speed. Setting a -ve speed value will reverse direction.

Start Bit Usage

BIT	Name	Value	Description
B0	Start/New Set Point	0	• Do not execute any changes in profile registers • Do not start a new profile • Any profile currently in motion will be completed

		0 → 1	• Start a profile move defined by the profile registers
			• 0 → 1 transition is required to start a profile after
			  • an error has been cleared
			  • a home has been completed
			  • a stop command has been given from CML or an input.

		1	Execute any changes in any of the profile registers immediately.

Motion Parameters

Speed moves will accept the following motion parameters. These will be processed immediately if the start bit is high.

Name	Unit	Data Type	Access
SpeedTarget	counts/s	DINT	AOI(2) or Axis tag
AccelerationTarget	1000 counts/s^2	INT	AOI or Axis tag
DecelerationTarget(1)	1000 counts/s^2	INT	AOI or Axis tag
TorqueTarget	0.1% rated torque (1)	INT	Axis Tag

(1) If the deceleration is set to 0 the acceleration value will be used
(2) MoveSpeed AOI

Torque Mode

A torque move only uses a target torque and maximum speed. Acceleration of the motor should be limited by adjusting the target torque or speed.

Setting a -ve torque value will reverse direction.

Start Bit Usage

BIT	Name	Value	Description
B0	Start/New Set Point	0	• Do not execute any changes in profile registers • Do not start a new profile • Any profile currently in motion will be completed

		0 → 1	• Start a profile move defined by the profile registers
			• 0 → 1 transition is required to start a profile after
			  • an error has been cleared
			  • a home has been completed
			  • a stop command has been given from CML or an input.

		1	Execute any changes in any of the profile registers immediately.

Motion Parameters

Torque moves will accept the following motion parameters. These will be processed immediately if the start bit is high.

Name	Unit	Data Type	Access
SpeedTarget	counts/s	DINT	AOI(1) or Axis tag
TorqueTarget	0.1% rated torque Range: ± 1100 Where: • 1100 = peak torque • 1000 = rated torque	INT	AOI or Axis Tag

(1) MoveTorque AOI

8.5 General Resources

8.5.1 EDS File

The CM1-T EtherNet/IP eds file can be found on the Datasheets and Downloads page.

8.5.2 CM1-T Library

The CM1-T Library for use with Codesys, Schneider, Beckhoff, Wago and other IEC 61131-3 PLCs can be found on the Datasheets and Downloads page.

8.6 RSLogix Resources

See Datasheets and Downloads to download the UDTs, AOIs and sample program.

8.6.1 User Data Types

Direct Control Operation

CM1T_Axis

The CM1-T_Axis data type defines how the bytes are assigned withing the IO message.

Name	Data Type	Description
Input	SINT[36]	Data coming from the CM1-T

Name	Data Type	Description
Output	SINT[28]	Data going to the CM1-T
us50Counter	DINT	Motor 50us counter
ModeOfOperation	SINT	Select control mode
ModeOfOperationDisplay	SINT	Current control mode
Controlword	CM1T_Controlword	Control operation
DriveStatus	CM1T_DriveStatus	Drive status
PositionTarget	DINT	Target position for position move
SpeedTarget	DINT	Max speed for target move
TorqueTarget	INT	Max torque for target move
AccelerationTarget	INT	Acceleration for target move
DecelerationTarget	INT	Deceleration for target move
PositionActual	DINT	Actual position of the motor
TargetActual	DINT	1ms target position of the motor
PositionError	DINT	Error between actual and target
SpeedActual	DINT	Actual speed of the motor
CurrentActual	INT	Actual current percentage of the motor (0.1% rated)
OverloadActual	INT	0.1% of overload torque
DCVoltage	INT	24V DC input value at 0.1V
AnalogIN	INT	10bit analog input (0-1023)
DigitalIN	SINT	Read digital inputs
DigitalOUT	SINT	Set digital outputs
Temperature	SINT	Drive temperature in °C
ErrorCode	SINT	ErrorCode is currently reserved and returns 0. Use MotorStatus to get motor error information

Name	Data Type	Description
ReadAddress1	INT	Address of register to read
ReadValue1	DINT	Value of read register
ReadValueAddress1	INT	Address of the current read register
WriteValue1	DINT	Register write value
WriteRegister1	INT	Write register address

CM1T_Controlword

Name	Data Type	Description
Start	BOOL	Start/New set point
Halt	BOOL	Halt current motion
Relative_Move	BOOL	Execute position move as relative
Speed_Control	BOOL	Execute move as a speed only move
Disable	BOOL	Disable the drive
Reset_Error	BOOL	Reset error with rising edge
Set_to_Zero	BOOL	Set the current position to zero with rising edge
Home	BOOL	Home the motor with rising edge

CM1T_DriveStatus

Name	Data Type	Description
StatusValue	SINT	Motor status and error type
StatusValue1	SINT	Additional motor status
In_Position	BOOL	Motor is in position and ready to move
Moving	BOOL	Motor is currently moving
Disabled	BOOL	Drive is disabled
Error	BOOL	Drive is in error state

Name	Data Type	Description
MoveComplete	BOOL	Move is completed
NewStartRequired	BOOL	Rising edge required to restart motion
HomeAchieved	BOOL	Homing has been achieved

CML Operation

CM1T_CMLAxis

The CM1-T_CMLAxis data type defines how the bytes are assigned withing the IO message.

Name	Data Type	Description
Input	SINT[60]	Data coming from the CM1-T
Output	SINT[44]	Data going to the CM1-T
us50Counter	DINT	Motor 50us counter
DriveStatus	CM1T_DriveStatus	Drive status
PositionActual	DINT	Actual position of the motor
TargetActual	DINT	1ms target position of the motor
PositionError	DINT	Error between actual and target
SpeedActual	DINT	Actual speed of the motor
CurrentActual	INT	Actual current percentage of the motor (0.1% rated)
OverloadActual	INT	0.1% of overload torque
DCVoltage	INT	24V DC input value at 0.1V
AnalogIN	INT	10bit analog input (0-1023)
DigitalIN	SINT	Read digital inputs
DigitalOUT	SINT	Set digital outputs
Temperature	SINT	Drive temperature in °C

Name	Data Type	Description
ErrorCode	SINT	(1) - ErrorCode is currently unused and returns 0. Use MotorStatus to get motor error information
WriteValue	DINT[5]	Register write value (array) Address of register to read
ReadValue	DINT[5]	Value of read register (array)
WriteAddress	INT[5]	Write register address (array)
ReadAddress	INT[5]	Read address register (array)
ReadValueAddress	INT[5]	Address of the current returned read register (array)

CM1T_CMLDriveStatus

Name	Data Type	Description
StatusValue	SINT	Motor status and error type
StatusValue1	SINT	Additional motor status
In_Position	BOOL	Motor is in position and ready to move
Moving	BOOL	Motor is currently moving
Disabled	BOOL	Drive is disabled
Error	BOOL	Drive is in error state
HomeAchieved	BOOL	Homing has been achieved

8.6.2 Add-On Instructions

Direct Control Operation

The following tables outlines the available add-on instructions

Name	Description	Required
CM1T_DataExchange	Exchanges data between the drive and the drive variable	Yes
CM1T_Enable	Enables the motor	No

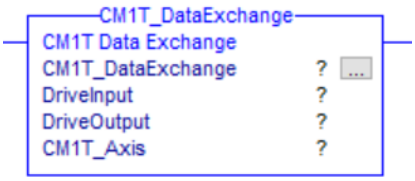
Name	Description	Required
CM1T_Disable	Disables the motor	No
CM1T_ClearFault	Clears any faults on the drive	No
CM1_TControlMode	Set the type of control mode required	No
CM1T_MoveAbsolute	Executes an absolute position move	No
CM1T_MoveRelative	Executes a relative position move	No
CM1T_MoveSpeed	Executes a speed move	No
CM1T_MoveTorque	Executes a torque move	No
CM1T_MoveHome	Executes a home routine	No
CM1T_ReadRegister	Read a motor register value	No
CM1T_WriteRegister	Write a value to a motor register	No

CM1T_DataExchange

This AOI is required if any other AOIs are used. One instance is required for each CM1-T axis being controlled.

It has a number of functions

1. It exchanges data between the motor connections and the motor/drive tag.
2. It places the Controlword bits into the Controlword data type
3. It retrieves the drive status and places it into the DriveStatus variable.



Parameters

Operand	Type	Format	Description
CM1T_DataExchange	CM1T_DataExchange	Tag	Instance tag for the AOI
DriveInput	AB:ETHERNET_MODULE_SINT_32Bytes:I:0	Tag	Module defined connection input for the CM1-T

Operand	Type	Format	Description
DriveOutput	AB:ETHERNET_MODULE_SINT_20Bytes:O:0	Tag	Module defined connection output for the CM1-T
CM1T_Axis	CM1T_Axis	Tag	Working tag for the CM1-T axis

Execution

Condition	Action
Prescan	• Initialize tag and local variables • rung-condition-out is set to false
Rung Condition - TRUE	• Process DataExchange • rung-condition-out is set to true
Rung Condition - FALSE	N/A - rung-condition-out is set to false

CM1T_Enable

This AOI enables the motor and drive. Torque will be present.

Parameters

Operand	Type	Format	Description
CM1T_Enable	CM1T_Enable	Tag	Instance tag for the AOI
CM1T_Axis	CM1T_Axis	Tag	Working tag for the CM1-T axis

Execution

Condition	Action
prescan	N/A - rung-condition-out is set to false
rung-condition-in is true	1. Drive disable bit is set to false (drive is enabled) 2. rung-condition-out is set a. true - when drive status bit shows drive enabled b. false - when drive status bit show drive disabled

Condition	Action
rung-condition-in is false	N/A - rung-condition-out is set to false

CM1T_Disable

This AOI disables the motor and drive. Torque will not be present

Parameters

Operand	Type	Format	Description
CM1T_Disable	CM1T_Disable	Tag	Instance tag for the AOI
CM1T_Axis	CM1T_Axis	Tag	Working tag for the CM1-T axis

Execution

Condition	Action
prescan	N/A - rung-condition-out is set to false
rung-condition-in is true	1. Drive disable bit is set to true (drive is disabled) 2. rung-condition-out is set a. true - when drive status bit shows drive disabled b. false - when drive status bit show drive enabled
rung-condition-in is false	N/A - rung-condition-out is set to false

CM1T_ClearFault

This AOI clears any drive fault/error condition. This will also enable the motor. The rung-out-condition indicates the drive error state (true or false).

Parameters

Operand	Type	Format	Description
CM1T_ClearFault	CM1T_ClearFault	Tag	Instance tag for the AOI
CM1T_Axis	CM1T_Axis	Tag	Working tag for the CM1-T axis

Execution

Condition	Action
Prescan	N/A

rung-condition-in is true	• Assert clear error bit
	• rung-condition-out is set
	  • true - when drive status bit shows drive error
	  • false - when drive status bit shows no drive error

rung-condition-in is false	• Deassert clear error bit
	• rung-condition-out is set
	  • true - when drive status bit shows drive error
	  • false - when drive status bit shows no drive error

CM1T_ControlMode

This AOI set the motion control mode.

Parameters

Operand	Type	Format	Description
CM1T_Disable	CM1T_ControlMode	Tag	Instance tag for the AOI
CM1T_Axis	CM1T_Axis	Tag	Working tag for the CM1-T axis
ControlMode_Value	DINT	Immediate/Tag	A value that sets the type of control mode required 0 - POSITION 1 - SPEED 2 - TORQUE

Execution

Condition	Action
prescan	N/A - rung-condition-out is set to false

Condition	Action
rung-condition-in is true	• Sets ModeOfOperation value and Controlword bits to set the control mode required • rung-condition-out is true
rung-condition-in is false	N/A - rung-condition-out is set to false

CM1T_MoveAbsolute

This AOI will execute an absolute position move.

Parameters

Operand	Type	Format	Description
CM1T_MoveAbsolute	CM1T_MoveAbsolute	Tag	Instance tag for the AOI
CM1T_Axis	CM1T_Axis	Tag	Working tag for the CM1-T axis
Position	DINT	Immediate/ Tag	Target absolute position
Speed	DINT	Immediate/ Tag	Target speed
Acceleration	INT	Immediate/ Tag	Target acceleration
Deceleration (optional)	INT	Immediate/ Tag	Target deceleration (uses acceleration if set to 0)

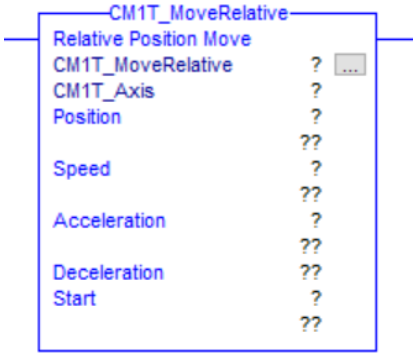
Execution

Condition	Action
prescan	N/A - rung-condition-out is set to false

Condition	Action
rung-condition-in is true	• Processes values and sets start bit • rung-condition-out is true
rung-condition-in is false	N/A - rung-condition-out is set to false

CM1T_MoveRelative

This AOI will execute a relative position move.



Parameters

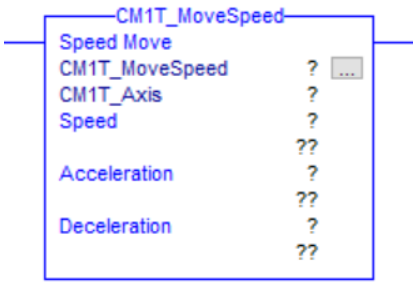
Operand	Type	Format	Description
CM1T_MoveRelative	CM1T_MoveRelative	Tag	Instance tag for the AOI
CM1T_Axis	CM1T_Axis	Tag	Working tag for the CM1-T axis
Position	DINT	Immediate/ Tag	Target absolute position
Speed	DINT	Immediate/ Tag	Target speed
Acceleration	INT	Immediate/ Tag	Target acceleration
Deceleration (optional)	INT	Immediate/ Tag	Target deceleration (uses acceleration if set to 0)
Start	BOOL	Immediate/ Tag	0 → 1 transition will execute a new relative move. If a motion is currently in process it will increment relative to the current position

Execution

Condition	Action
prescan	N/A - rung-condition-out is set to false
rung-condition-in is true	• Processes values and sets start bit as per Start input variable • rung-condition-out is true
rung-condition-in is false	N/A - rung-condition-out is set to false

CM1T_MoveSpeed

This AOI will execute a speed move.



Parameters

Operand	Type	Format	Description
CM1T_MoveSpeed	CM1T_MoveSpeed	Tag	Instance tag for the AOI
CM1T_Axis	CM1T_Axis	Tag	Working tag for the CM1-T axis
Speed	DINT	Immediate/ Tag	Target speed
Acceleration	INT	Immediate/ Tag	Target acceleration
Deceleration (optional)	INT	Immediate/ Tag	Target deceleration (uses acceleration if set to 0)

Execution

Condition	Action
prescan	N/A - rung-condition-out is set to false

Condition	Action
rung-condition-in is true	• Processes values and start the speed move • rung-condition-out is true
rung-condition-in is false	N/A - rung-condition-out is set to false

CM1T_MoveTorque

This AOI will execute a torque move.

Parameters

Operand	Type	Format	Description
CM1T_MoveSpeed	CM1T_MoveSpeed	Tag	Instance tag for the AOI
CM1T_Axis	CM1T_Axis	Tag	Working tag for the CM1-T axis
Torque	INT	Immediate/ Tag	Target torque
MaxSpeed	DINT	Immediate/ Tag	Target speed

Execution

Condition	Action
prescan	N/A - rung-condition-out is set to false
rung-condition-in is true	• Processes values and start the torque move • rung-condition-out is true
rung-condition-in is false	N/A - rung-condition-out is set to false

CM1T_ReadRegister

This AOI loads a register address to read. It returns the register value. The value is returned when the ReadRegister (output) address and the ReadValueAddress (input) address are equal.

Parameters

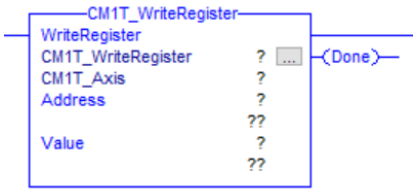
Operand	Type	Format	Description
CM1T_ReadRegister	CM1T_ReadRegister	Tag	Instance tag for the AOI
CM1T_Axis	CM1T_Axis	Tag	Working tag for the CM1-T axis
Address	INT	Immediate/ Tag	Address of register to read
Value	DINT	Tag	Tag to load the returned value
Done	BOOL	Tag	Optional output indicating the value has been returned.

Execution

Condition	Action
prescan	N/A - rung-condition-out is set to false
rung-condition-in is true	• Processes values and start the torque move • rung-condition-out is true
rung-condition-in is false	N/A - rung-condition-out is set to false

CM1T_WriteRegister

This AOI loads a register address and value to write. Once the write cycle is complete the done output is set.



Parameters

Operand	Type	Format	Description
CM1TWriteRegister	CM1T_WriteRegister	Tag	Instance tag for the AOI
CM1T_Axis	CM1T_Axis	Tag	Working tag for the CM1-T axis
Address	INT	Immediate/ Tag	Address of register to write
Value	DINT	Immediate/ Tag	Tag/value to load into the write value
Done	BOOL	Tag	Optional output indicating the value has been written.

Execution

Condition	Action
prescan	N/A - rung-condition-out is set to false
rung-condition-in is true	• Processes values and start the torque move • rung-condition-out is true
rung-condition-in is false	N/A - rung-condition-out is set to false

CML Operation

The following tables outlines the available add-on instructions

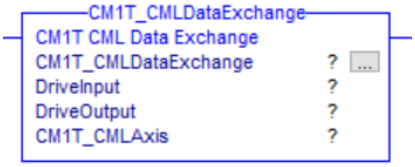
Name	Description	Required
CM1T_CMLDataExchange	Exchanges data between the drive and the drive variable	Yes
CM1T_CMLReadRegister	Reads a motor register	No
CM1T_CMLWriteRegister	Writes to a motor register	No

CM1T_CMLDataExchange

This AOI is required if any other CML AOIs are used. One instance is required for each CM1-T axis being controlled.

It has a number of functions

1. It exchanges data between the motor connections and the motor/drive tag.
2. It retrieves the drive status and places it into the DriveStatus variable.



Parameters

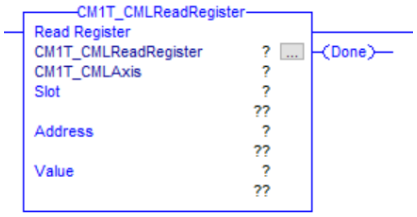
Operand	Type	Format	Description
CM1T_DataExchange	CM1T_DataExchange	Tag	Instance tag for the AOI
DriveInput	AB:ETHERNET_MODULE_SINT_32Bytes:I:0	Tag	Module defined connection input for the CM1-T
DriveOutput	AB:ETHERNET_MODULE_SINT_20Bytes:O:0	Tag	Module defined connection output for the CM1-T
CM1T_Axis	CM1T_Axis	Tag	Working tag for the CM1-T axis

Execution

Condition	Action
Prescan	N/A - rung-condition-out is set to false
Rung Condition - TRUE	• Process DataExchange • rung-condition-out is set to true
Rung Condition - FALSE	N/A - rung-condition-out is set to false

CM1T_CMLReadRegister

This AOI loads a register address to read. It returns the register value. The value is returned when the ReadRegister (output) address and the ReadValueAddress (input) address are equal.



Parameters

Operand	Type	Format	Description
CM1T_ReadRegister	CM1T_ReadRegister	Tag	Instance tag for the AOI
CM1T_Axis	CM1T_Axis	Tag	Working tag for the CM1-T axis
Slot	SINT	Immediate/ Tag	There are 5 slots available. Enter 1-5.
Address	INT	Immediate/ Tag	Address of register to read
Value	DINT	Tag	Tag to load the returned value
Done	BOOL	Tag	Optional output indicating the value has been returned.

Execution

Condition	Action
prescan	N/A - rung-condition-out is set to false
rung-condition-in is true	• Processes values and start the torque move • rung-condition-out is true
rung-condition-in is false	N/A - rung-condition-out is set to false

CM1T_CMLWriteRegister

This AOI loads a register address and value to write. Once the write cycle is complete the done output is set.

Parameters

Operand	Type	Format	Description
CM1TWriteRegister	CM1T_WriteRegister	Tag	Instance tag for the AOI
CM1T_Axis	CM1T_Axis	Tag	Working tag for the CM1-T axis
Slot	SINT	Immediate/ Tag	There are 5 slots available. Enter 1-5.
Address	INT	Immediate/ Tag	Address of register to write
Value	DINT	Immediate/ Tag	Tag/value to load into the write value
Done	BOOL	Tag	Optional output indicating the value has been written.

Execution

Condition	Action
prescan	N/A - rung-condition-out is set to false
rung-condition-in is true	• Processes values and start the torque move • rung-condition-out is true
rung-condition-in is false	N/A - rung-condition-out is set to false

8.6.3 Adding CM1-T and AOIs to Project

Download and extract the EtherNet/IP support file.

The file includes the following

1. Direct Control Operation
    a. CM1T_SampleApplication.ACD - This a sample application built RSLogix. It shows how to
        i. Run the motors in each of the modes
        ii. Reset from faults
        iii. Enable/Disable the motor
    b. User Data Types (UDT)
    c. Add-On Instructions (AOI)
2. CML Operation
    a. CM1T_CMLSampleApplication.ACD - RSLogix application to read/write motor registers
    b. CM1T_CMLSampleApplication.crp - Control Room project with CML sample program.
        i. Executes incremental move if variable is set (Run, distance, speed and acceleration set from PLC)
        ii. Increments counter variable (Variable read in PLC)

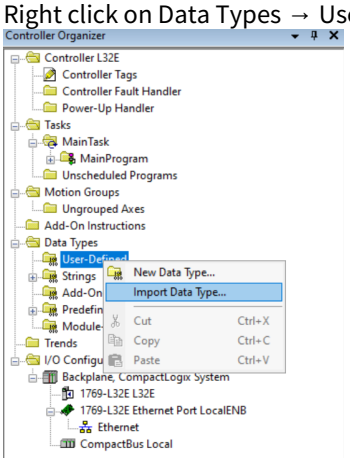
c. User Data Types (UDT)
d. Add-On Instructions (AOI)

Direct Control Operation

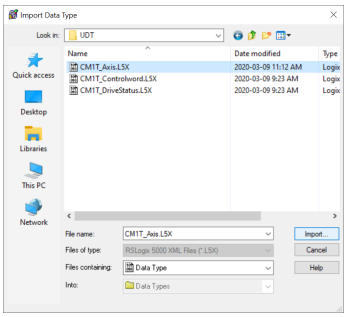
Import User Data Types (UDTs)

The following steps show how to import the UDTs used by the CM1-T module.

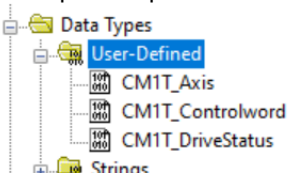
1. Right click on Data Types → User-Defined and select "Import Data Type..."



2. Browse to the location where the CM1T_AOI support file was extracted. Select the CM1T_Axis and click import.



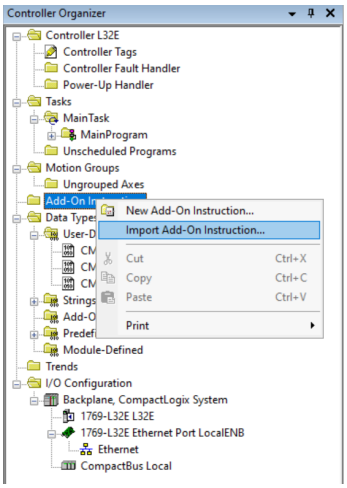
3. On the Import Configuration window click OK
4. The CM1T_Axis, CM1T_Controlword and CM1T_DriveStatus should all have been import as shown below. If they were not repeat steps 1-3 with the CM1T_Controlword and CM1T_DriveStatus.



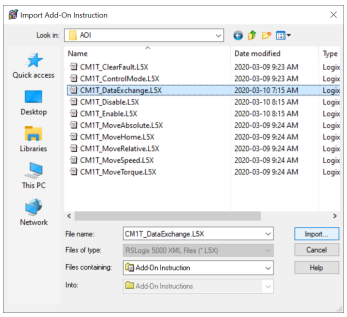
Import Add-On Instructions (AOIs)

The following steps show how to import the CM1-T modules AOIs. The only required AOI is CM1T_DataExchange. All the other AOIs depend on this one to populate the data type.

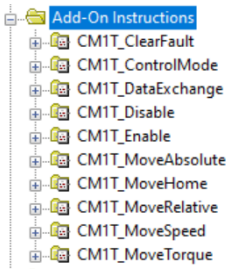
1. Right click on Add-On Instructions and select "Import Add-On Instruction..."



2. Browse to the extracted AOI folder and select CM1T_DataExchange. Click Import



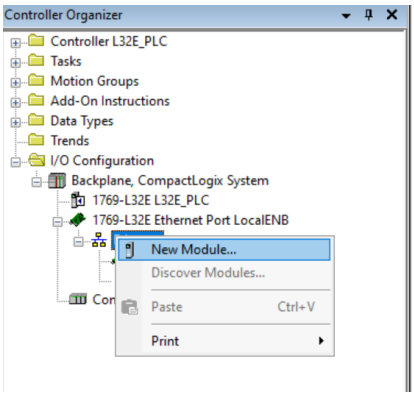
3. On the Configuration Import window click OK.
4. Repeat steps 1-3 until all required AOIs are imported
5. The Add-On Instruction tree will look as follows once all AOIs are imported



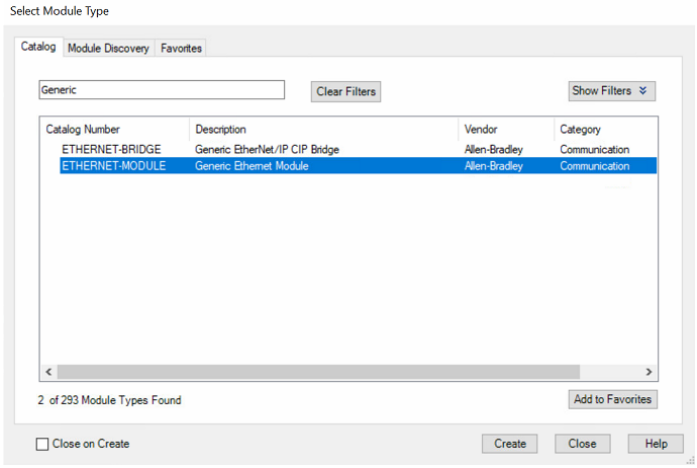
Add CM1-T Module

The following steps show how to add an CM1-T module in RSLogix. At present the EtheNet/IP is in testing and the CM1-T is added a a generic module.

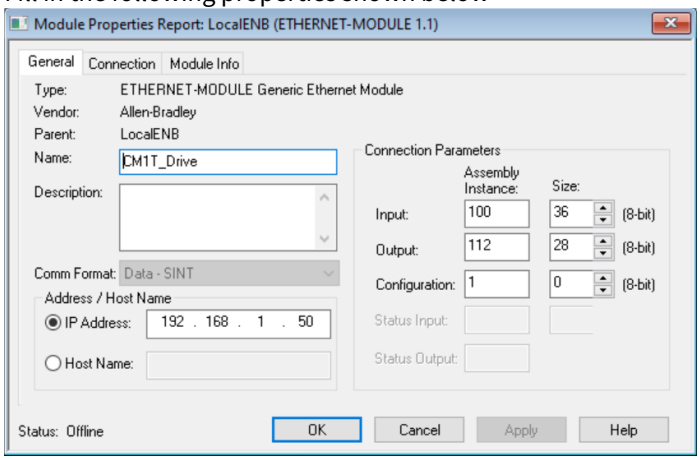
1. Open an existing project or start a new one
2. In the Controller Organizer expand I/O Configuration and right click on "Ethernet". Select "New Module"



3. Search "Generic", Select "ETHERNET-MODULE" and click Create



4. Fill in the following properties shown below

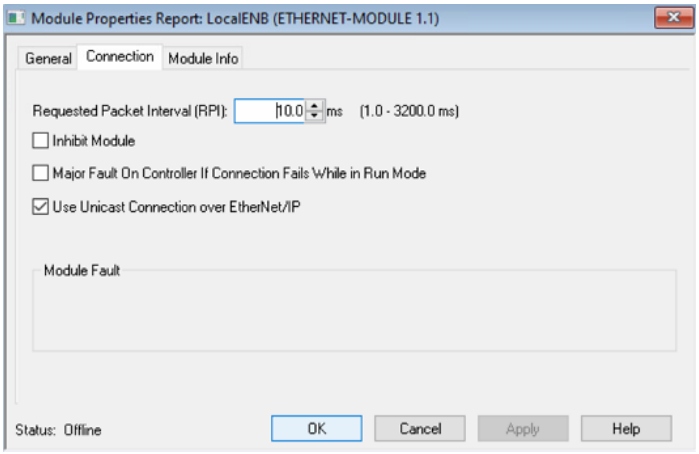


a. Name: Give the drive a name. E.g. CM1T_Drive, Axis1, LabelXAxis, etc
b. Comm Format: Data - SINT
c. IP Address: Enter the IP Address of the CM1-T you are adding.
d. Connection Parameters
    i. Input: Assembly Instance = 100, Size = 36
    ii. Output: Assembly Instance = 112, Size = 28

iii. Configuration: Assembly Instance = 1, Size = 0

e. Click OK

5. The Module Properties window should open. If it does not double click on the newly created module inside the Ethernet heading.
   Select the Connection Tab

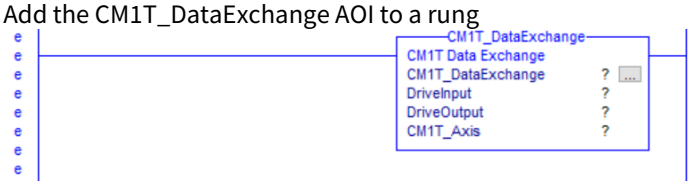


a. Set the packet interval as required. Typically this would be 10ms.

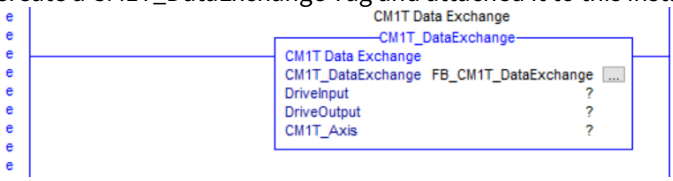
b. Check the boxes as shown in the picture above

The final set of steps below shows how to add the CM1T_DataExchange AOI and configure it to exchange data between the tag created for the CM1-T and the CM1-T module. A single instance of the CM1T_DataExchange should be created per module. There should never be more than one per module.

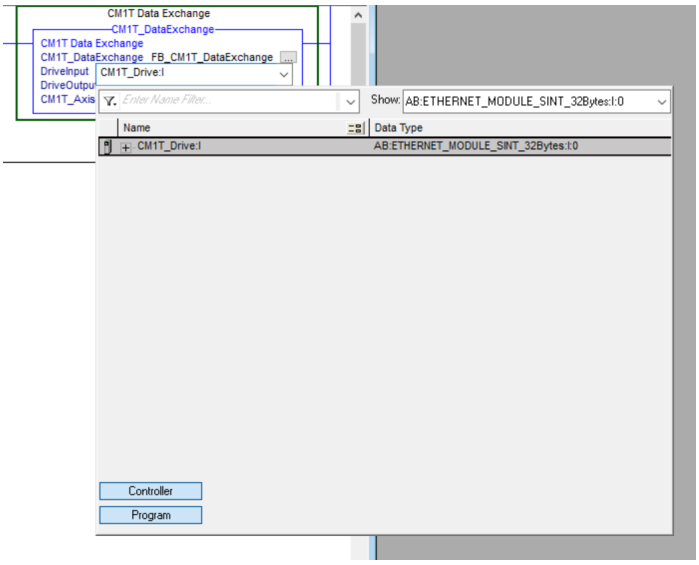
1. Add the CM1T_DataExchange AOI to a rung



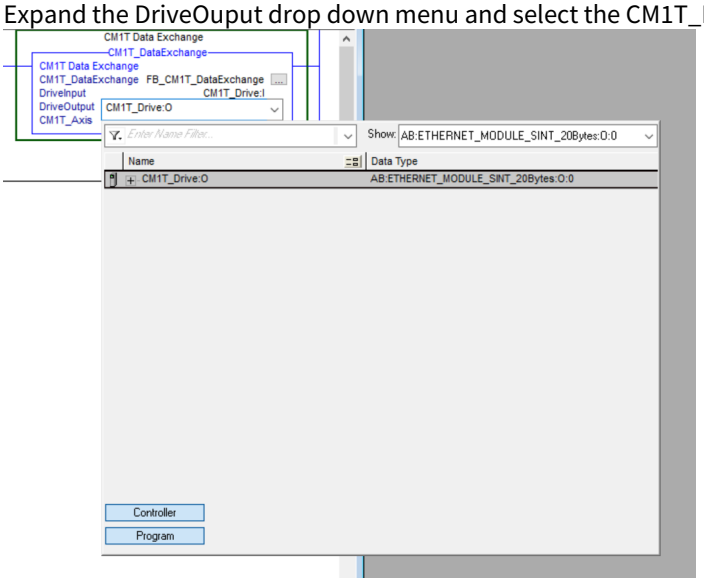
2. Create a CM1T_DataExchange Tag and attached it to this instance. In this example it is called FB_CM1T_DataExchnage



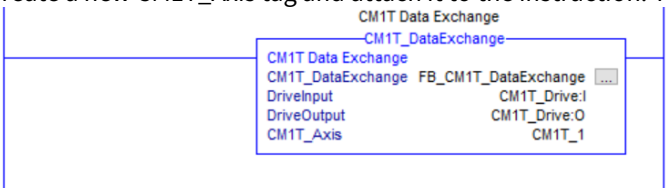
3. Expand the DriveInput drop down menu and select the CM1T_Drive module input create earlier



4. Expand the DriveOuput drop down menu and select the CM1T_Drive module output create earlier



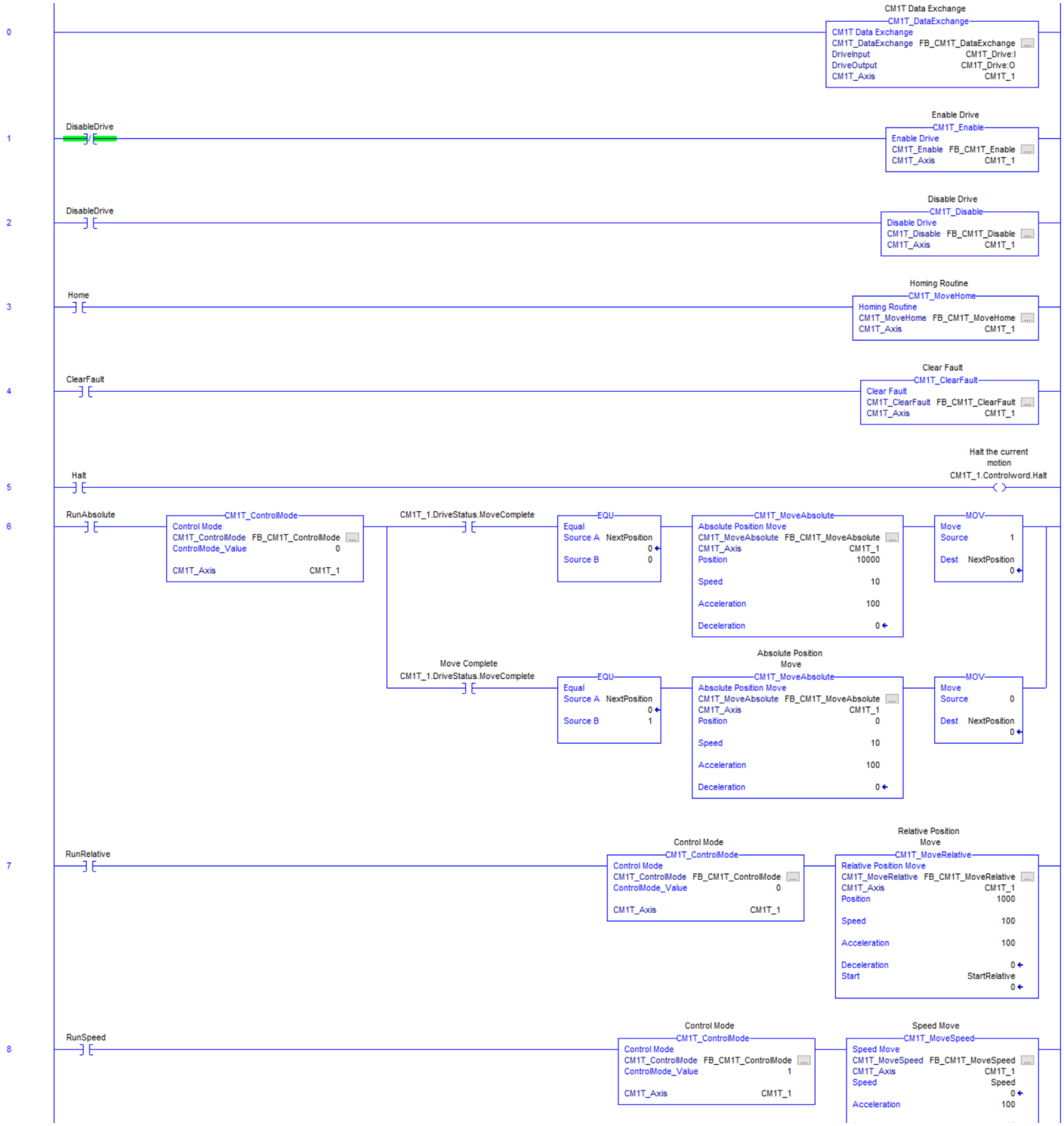
5. Create a new CM1T_Axis tag and attach it to the instruction. This is the tag that will be used in all other AOIs

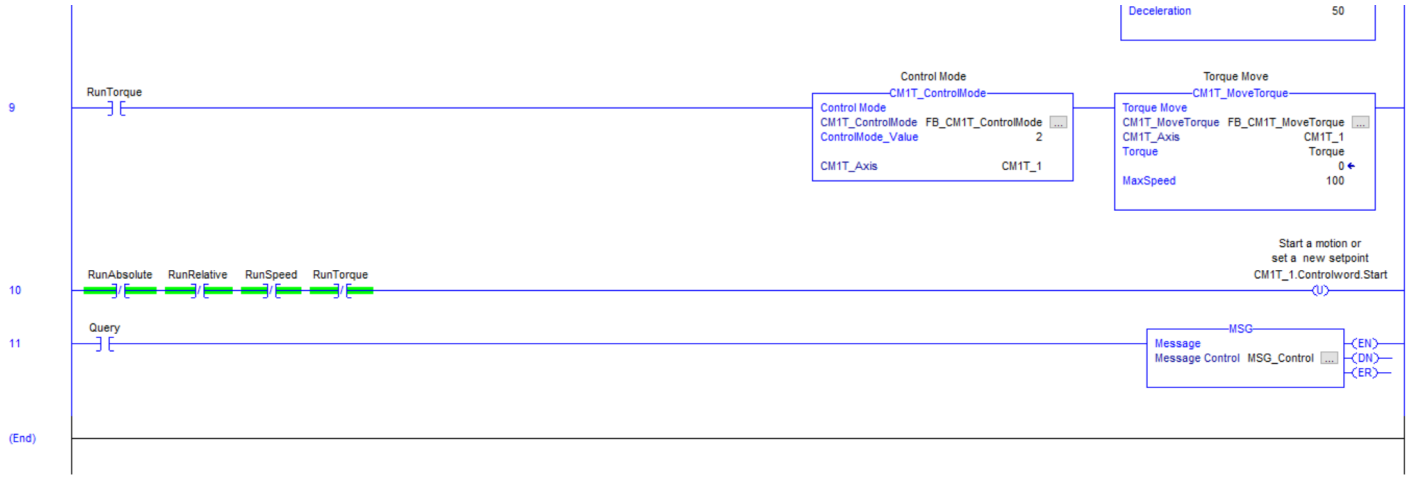


All CM1-T AOIs are now ready to be used. See the example application for more details on running the motor.

Example Logic - Direct Control Operation Sample

The following logic is from the Direct operation sample application. The example shows how to run the motor in each of the modes, enable, disable, halt, and recover from an error.





CML Operation

Import User Data Types (UDTs)

Following the instructions for Direct Control Operation except import

- CM1T_CMLAxis
- CM1T_CMLDriveStatus

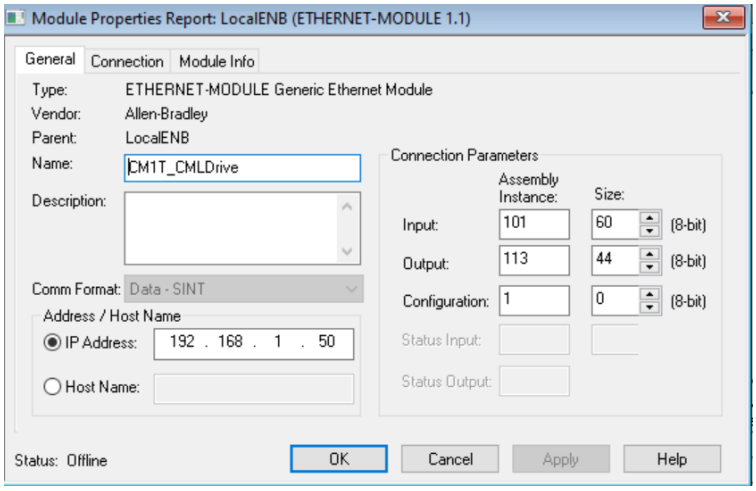
Import Add-On Instructions (AOIs)

Follow the instructions for Direct Control Operation except import

- CM1T_CMLDataExchange
- CM1T_CMLReadRegister
- CM1T_CMLWriteRegister

Add CM1-T Module

Follow the instructions for Direct Control Operation except on step 4 enter the following configuration data

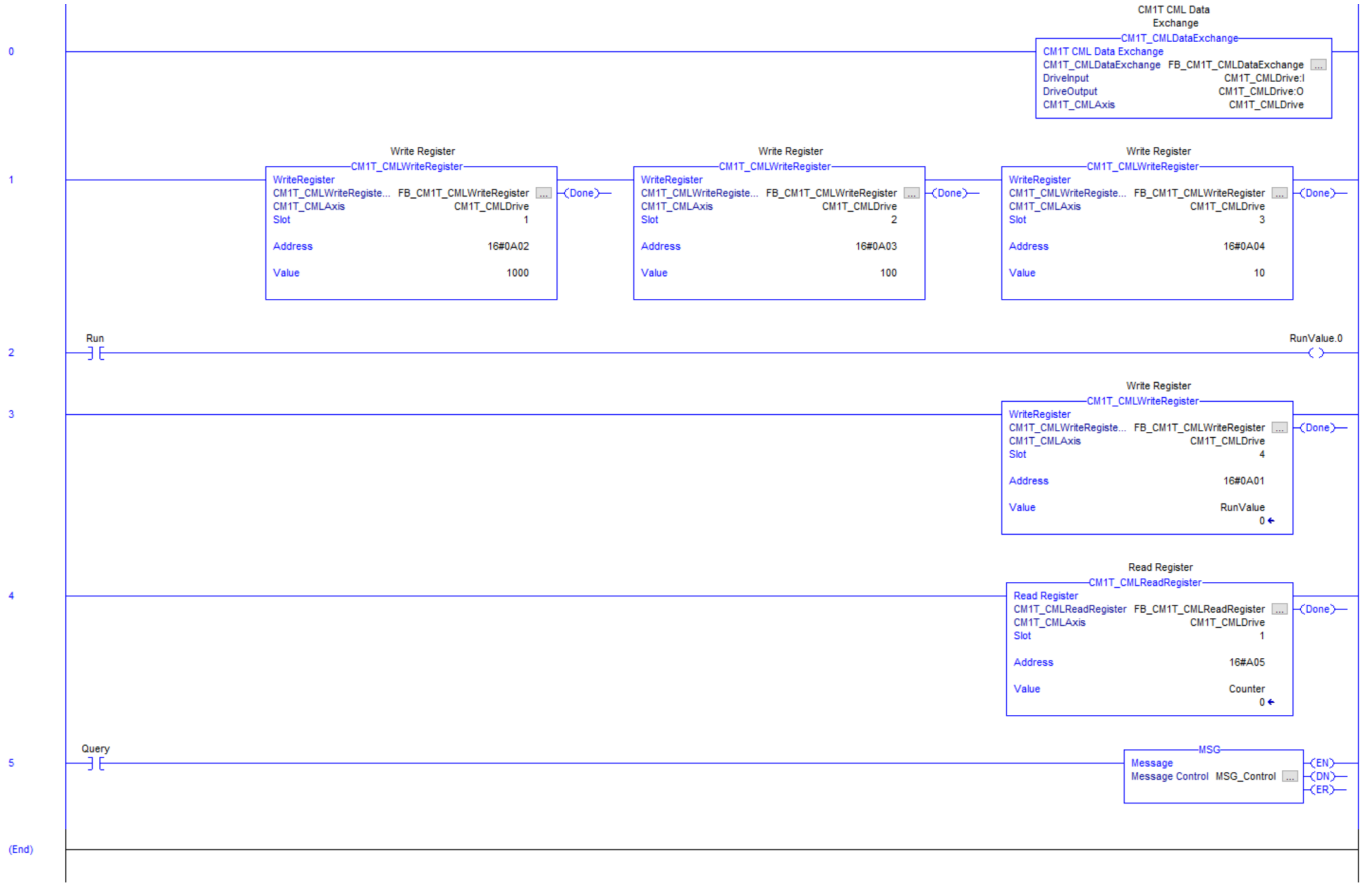


1. Name: Give the drive a name. E.g. CM1T_Drive, Axis1, LabelXAxis, etc
2. Comm Format: Data - SINT
3. IP Address: Enter the IP Address of the CM1-T you are adding.
4. Connection Parameters
    a. Input: Assembly Instance = 101, Size = 60
    b. Output: Assembly Instance = 113, Size = 44
    c. Configuration: Assembly Instance = 1, Size = 0
5. Click OK

Example Logic - CML Operation Sample

The following sample logic is supplied with the AOIs. It requires the motor to be programmed with a CML program. The program does the following

1. Motor indexes continuously by the distance amount defined by PLC
2. Motor indexes with speed and acceleration defined by PLC
3. Indexing is started and stopped by PLC
4. A counter showing number of indexes is read by the PLC



8.7 Appendix

8.7.1 A - Support Files

See Datasheets and Downloads.

8.7.2 B - Complete Register Address Map

K Registers

Register	Address	
	Hex	Decimal
K21	16#0115	277
K22	16#0116	278
K23	16#0117	279
K24	16#0118	280
K25	16#0119	281
K26	16#011A	282
K27	16#011B	283
K28	16#011C	284
K29	16#011D	285
K30	16#011E	286
K31	16#011F	287
K32	16#0120	288
K33	16#0121	289
K34	16#0122	290
K35	16#0123	291

Register	Address	
	Hex	Decimal
K47	16#012F	303
K48	16#0130	304
K49	16#0131	305
K50	16#0132	306
K51	16#0133	307
K52	16#0134	308
K53	16#0135	309
K54	16#0136	310
K55	16#0137	311
K56	16#0138	312
K57	16#0139	313
K58	16#013A	314
K59	16#013B	315
K60	16#013C	316
K61	16#013D	317

Register	Address	
	Hex	Decimal
K73	16#0149	329
K74	16#014A	330
K75	16#014B	331
K76	16#014C	332
K77	16#014D	333
K78	16#014E	334
K79	16#014F	335
K80	16#0150	336
K81	16#0151	337
K82	16#0152	338
K83	16#0153	339
K84	16#0154	340
K85	16#0155	341
K86	16#0156	342
K87	16#0157	343

Register	Address	
	Hex	Decimal
K36	16#0124	292
K37	16#0125	293
K38	16#0126	294
K39	16#0127	295
K40	16#0128	296
K41	16#0129	297
K42	16#012A	298
K43	16#012B	299
K44	16#012C	300
K45	16#012D	301
K46	16#012E	302

Register	Address	
	Hex	Decimal
K62	16#013E	318
K63	16#013F	319
K64	16#0140	320
K65	16#0141	321
K66	16#0142	322
K67	16#0143	323
K68	16#0144	324
K69	16#0145	325
K70	16#0146	326
K71	16#0147	327
K72	16#0148	328

Register	Address	
	Hex	Decimal
K88	16#0158	344
K89	16#0159	345
K90	16#015A	346

H Registers

Register	Address	
	Hex	Decimal
H0	16#0200	512
H1	16#0201	513
H2	16#0202	514
H3	16#0203	515
H4	16#0204	516
H5	16#0205	517
H6	16#0206	518

Register	Address	
	Hex	Decimal
H7	16#0207	519

P Registers

Register	Address	
	Hex	Decimal
P0	16#0300	768
P1	16#0301	769
P2	16#0302	770
P3	16#0303	771
P4	16#0304	772
P5	16#0305	773
P6	16#0306	774
P7	16#0307	775
P8	16#0308	776
P9	16#0309	777
P10	16#030A	778
P11	16#030B	779
P12	16#030C	780
P13	16#030D	781
P14	16#030E	782
P15	16#030F	783
P16	16#0310	784

S Registers

Register	Address	
	Hex	Decimal
S0	16#0400	1024
S1	16#0401	1025
S2	16#0402	1026
S3	16#0403	1027
S4	16#0404	1028
S5	16#0405	1029
S6	16#0406	1030
S7	16#0407	1031
S8	16#0408	1032
S9	16#0409	1033
S10	16#040A	1034
S11	16#040B	1035
S12	16#040C	1036
S13	16#040D	1037
S14	16#040E	1038
S15	16#040F	1039

Register	Address	
	Hex	Decimal
P17	16#0311	785
P18	16#0312	786
P19	16#0313	787
P20	16#0314	788
P21	16#0315	789
P22	16#0316	790
P23	16#0317	791
P24	16#0318	792
P25	16#0319	793

A Registers

Register	Address	
	Hex	Decimal
A0	16#0500	1280
A1	16#0501	1281
A2	16#0502	1282
A3	16#0503	1283
A4	16#0504	1284
A5	16#0505	1285
A6	16#0506	1286
A7	16#0507	1287
A8	16#0508	1288

T Registers

Register	Address	
	Hex	Decimal
T1	16#0601	1537
T2	16#0602	1538
T3	16#0603	1539
T4	16#0604	1540
T5	16#0605	1541
T6	16#0606	1542
T7	16#0607	1543
T8	16#0608	1544

M Registers

Register	Address	
	Hex	Decimal
M0	16#0700	1792
M1	16#0701	1793
M2	16#0702	1794
M3	16#0703	1795
M4	16#0704	1796
M5	16#0705	1797
M6	16#0706	1798
M7	16#0707	1799

R Registers

Register	Address	
	Hex	Decimal
R0	16#0800	2048
R1	16#0801	2049
R2	16#0802	2050
R3	16#0803	2051
R4	16#0804	2052
R5	16#0805	2053
R6	16#0806	2054
R7	16#0807	2055
R8	16#0808	2056
R9	16#0809	2057
R10	16#080A	2058
R11	16#080B	2059
R12	16#080C	2060
R13	16#080D	2061
R14	16#080E	2062
R15	16#080F	2063
R16	16#0810	2064
R17	16#0811	2065
R18	16#0812	2066
R19	16#0813	2067
R20	16#0814	2068

N Registers

Register	Address	
	Hex	Decimal
N0	16#0900	2304
N1	16#0901	2305
N2	16#0902	2306
N3	16#0903	2307
N4	16#0904	2308
N5	16#0905	2309
N6	16#0906	2310
N7	16#0907	2311
N8	16#0908	2312
N9	16#0909	2313
N10	16#090A	2314
N11	16#090B	2315
N12	16#090C	2316
N13	16#090D	2317
N14	16#090E	2318
N15	16#090F	2319
N16	16#0910	2320
N17	16#0911	2321
N18	16#0912	2322
N19	16#0913	2323
N20	16#0914	2324

V Registers

Register	Address	
	Hex	Decimal
V0	16#0A00	2560
V1	16#0A01	2561
V2	16#0A02	2562
V3	16#0A03	2563
V4	16#0A04	2564
V5	16#0A05	2565
V6	16#0A06	2566
V7	16#0A07	2567
V8	16#0A08	2568
V9	16#0A09	2569
V10	16#0A0A	2570
V11	16#0A0B	2571
V12	16#0A0C	2572
V13	16#0A0D	2573
V14	16#0A0E	2574
V15	16#0A0F	2575

Register	Address	
	Hex	Decimal
R21	16#0815	2069
R22	16#0816	2070
R23	16#0817	2071
R24	16#0818	2072
R25	16#0819	2073

Register	Address	
	Hex	Decimal
N21	16#0915	2325
N22	16#0916	2326
N23	16#0917	2327
N24	16#0918	2328
N25	16#0919	2329

9 Modbus TCP

9.1 Setup Information

The following list indicates important setup information for running in Modbus TCP. Ensure that the master is setup with the same configuration.

- Port 502 - Modbus-TCP uses port 502. This is fixed.
- Holding register addressing depends on the system used. Refer to your modbus master documentation to understand the address format.
    - Additional information is below to list the most common address types
- Max holding registers for read and write is 16
- Use function code 0x10 for write single register. Function code 0x06 is not supported.
- All registers are 32-bit
    - Exceptions are the registers used used directly in the CM1-T Library (IEC 61131-3).
- 32-bit word mapping is little endian

9.2 Holding Register List

Included below is a table which details the location and function of the CM1 Modbus registers.
The addresses included below are referenced to the Modbus data model (PLC address for read/write to holding registers)

Note: Even if Direct Control is not used the Direct Control feedback registers can be useful for reading back motor information.

Motor Information

Holding Register Address		Motor Parameter	Read Access	Write Access
Base Address	Modicon Address			
0	40001	Position Error (?95)	Yes	No
2	40003	Motor Position (?96)	Yes	No
4	40005	Motor Speed (?97)	Yes	No
6	40007	Motor Torque (?98)	Yes	No
8	40009	Motor Status (?99)	Yes	No

9.2.1 Variables

Holding Register Address		Motor Parameter	Read Access	Write Access
Base Address	Modicon Address			
10	40011	V0	Yes	Yes
12	40013	V1	Yes	Yes
14	40015	V2	Yes	Yes
16	40017	V3	Yes	Yes
18	40019	V4	Yes	Yes
20	40021	V5	Yes	Yes
22	40023	V6	Yes	Yes
24	40025	V7	Yes	Yes
26	40027	V8	Yes	Yes
28	40029	V9	Yes	Yes
30	40031	V10	Yes	Yes
32	40033	V11	Yes	Yes
34	40035	V12	Yes	Yes
36	40037	V13	Yes	Yes
38	40039	V14	Yes	Yes
40	40041	V15	Yes	Yes

9.2.2 Direct Registers

Holding Register Address		Motor Parameter	Read Access	Write Access
Base Address	Modicon Address			
42	40043	P0	Yes	Yes

Holding Register Address		Motor Parameter	Read Access	Write Access
Base Address	Modicon Address			
44	40045	S0	Yes	Yes
46	40047	A0	Yes	Yes
48	40049	V0	Yes	Yes
50	40051	R0	Yes	Yes
52	40053	M0	Yes	Yes

9.2.3 Inputs/Outputs

Holding Register Address		Motor Parameter	Read Access	Write Access
Base Address	Modicon Address			
54	40055	Analog Output	Yes	Yes
56	40057	Input Status (?70)	Yes	No
58	40059	Output Status (?50)	Yes	Yes

9.2.4 Positions

Holding Register Address		Motor Parameter	Read Access	Write Access
Base Address	Modicon Address			
200	40201	P0	Yes	Yes
202	40203	P1	Yes	Yes
204	40205	P2	Yes	Yes
206	40207	P3	Yes	Yes
208	40209	P4	Yes	Yes
210	40211	P5	Yes	Yes
212	40213	P6	Yes	Yes

Holding Register Address		Motor Parameter	Read Access	Write Access
Base Address	Modicon Address			
214	40215	P7	Yes	Yes
216	40217	P8	Yes	Yes
218	40219	P9	Yes	Yes
220	40221	P10	Yes	Yes
222	40223	P11	Yes	Yes
224	40225	P12	Yes	Yes
226	40227	P13	Yes	Yes
228	40229	P14	Yes	Yes
230	40231	P15	Yes	Yes
232	40233	P16	Yes	Yes
234	40235	P17	Yes	Yes
236	40237	P18	Yes	Yes
238	40239	P19	Yes	Yes
240	40241	P20	Yes	Yes
242	40243	P21	Yes	Yes
244	40245	P22	Yes	Yes
246	40247	P23	Yes	Yes
248	40249	P24	Yes	Yes
250	40251	P25	Yes	Yes

9.2.5 **R Registers**

Holding Register Address		Motor Parameter	Read Access	Write Access
Base Address	Modicon Address			
300	40301	R0	Yes	Yes
302	40303	R1	Yes	Yes
304	40305	R2	Yes	Yes
306	40307	R3	Yes	Yes
308	40309	R4	Yes	Yes
310	40311	R5	Yes	Yes
312	40313	R6	Yes	Yes
314	40315	R7	Yes	Yes
316	40317	P8	Yes	Yes
318	40319	P9	Yes	Yes
320	40321	R10	Yes	Yes
322	40323	R11	Yes	Yes
324	40325	R12	Yes	Yes
326	40327	R13	Yes	Yes
328	40329	R14	Yes	Yes
330	40331	R15	Yes	Yes
332	40333	R16	Yes	Yes
334	40335	R17	Yes	Yes
336	40337	R18	Yes	Yes
338	40339	R19	Yes	Yes
340	40341	R20	Yes	Yes

Holding Register Address		Motor Parameter	Read Access	Write Access
Base Address	Modicon Address			
342	40343	R21	Yes	Yes
344	40345	R22	Yes	Yes
346	40347	R23	Yes	Yes
348	40349	R24	Yes	Yes
350	40351	R25	Yes	Yes

9.2.6 **N Registers**

Holding Register Address		Motor Parameter	Read Access	Write Access
Base Address	Modicon Address			
400	40401	N0	Yes	Yes
402	40403	N1	Yes	Yes
404	40405	N2	Yes	Yes
406	40407	N3	Yes	Yes
408	40409	N4	Yes	Yes
410	40411	N5	Yes	Yes
412	40413	N6	Yes	Yes
414	40415	N7	Yes	Yes
416	40417	N8	Yes	Yes
418	40419	N9	Yes	Yes
420	40421	N10	Yes	Yes
422	40423	N11	Yes	Yes
424	40425	N12	Yes	Yes

Holding Register Address		Motor Parameter	Read Access	Write Access
Base Address	Modicon Address			
426	40427	N13	Yes	Yes
428	40429	N14	Yes	Yes
430	40431	N15	Yes	Yes
432	40433	N16	Yes	Yes
434	40435	N17	Yes	Yes
436	40437	N18	Yes	Yes
438	40439	N19	Yes	Yes
440	40441	N20	Yes	Yes
442	40443	N21	Yes	Yes
444	40445	N22	Yes	Yes
446	40447	N23	Yes	Yes
448	40449	N24	Yes	Yes
450	40451	N25	Yes	Yes

9.2.7 **Speeds**

Holding Register Address		Motor Parameter	Read Access	Write Access
Base Address	Modicon Address			
602	40603	S0	Yes	Yes
604	40605	S1	Yes	Yes
606	40607	S2	Yes	Yes
608	40609	S3	Yes	Yes
610	40611	S4	Yes	Yes

Holding Register Address		Motor Parameter	Read Access	Write Access
Base Address	Modicon Address			
612	40613	S5	Yes	Yes
614	40615	S6	Yes	Yes
616	40617	S7	Yes	Yes
618	40619	S8	Yes	Yes
620	40621	S9	Yes	Yes
622	40623	S10	Yes	Yes
624	40625	S11	Yes	Yes
626	40627	S12	Yes	Yes
628	40629	S13	Yes	Yes
630	40631	S14	Yes	Yes
632	40633	S15	Yes	Yes

9.2.8 Accelerations

Holding Register Address		Motor Parameter	Read Access	Write Access
Base Address	Modicon Address			
634	40635	A0	Yes	Yes
636	40637	A1	Yes	Yes
638	40639	A2	Yes	Yes
640	40641	A3	Yes	Yes
642	40643	A4	Yes	Yes
644	40645	A5	Yes	Yes
646	40647	A6	Yes	Yes

Holding Register Address		Motor Parameter	Read Access	Write Access
Base Address	Modicon Address			
648	40649	A7	Yes	Yes
650	40651	A8	Yes	Yes

9.2.9 Torques

Holding Register Address		Motor Parameter	Read Access	Write Access
Base Address	Modicon Address			
652	40653	M0	Yes	Yes
654	40655	M1	Yes	Yes
656	40657	M2	Yes	Yes
658	40659	M3	Yes	Yes
660	40661	M4	Yes	Yes
662	40663	M5	Yes	Yes
664	40665	M6	Yes	Yes
666	40667	M7	Yes	Yes
668	40669	M8	Yes	Yes

9.2.10 K Parameters

Holding Register Address		Motor Parameter	Read Access	Write Access
Base Address	Modicon Address			
670	40671	RESERVED	Yes	No
672	40673	K1	Yes	No
674	40675	K2	Yes	No
676	40677	K3	Yes	No

Holding Register Address		Motor Parameter	Read Access	Write Access
Base Address	Modicon Address			
678	40679	K4	Yes	No
680	40681	K5	Yes	No
682	40683	K6	Yes	No
684	40685	K7	Yes	No
686	40687	K8	Yes	No
688	40689	K9	Yes	No
690	40691	K10	Yes	No
692	40693	K11	Yes	No
694	40695	K12	Yes	No
696	40697	K13	Yes	No
698	40699	K14	Yes	No
700	40701	K15	Yes	No
702	40703	K16	Yes	No
704	40705	K17	Yes	No
706	40707	K18	Yes	No
708	40709	K19	Yes	No
710	40711	K20	Yes	Yes
712	40713	K21	Yes	Yes
714	40715	K22	Yes	Yes
716	40717	K23	Yes	Yes
718	40719	K24	Yes	Yes
720	40721	K25	Yes	Yes

Holding Register Address		Motor Parameter	Read Access	Write Access
Base Address	Modicon Address			
722	40723	K26	Yes	Yes
724	40725	K27	Yes	Yes
726	40727	K28	Yes	Yes
728	40729	K29	Yes	Yes
730	40731	K30	Yes	Yes
732	40733	K31	Yes	Yes
734	40735	K32	Yes	Yes
736	40737	K33	Yes	Yes
738	40739	K34	Yes	Yes
740	40741	K35	Yes	Yes
742	40743	K36	Yes	Yes
744	40745	K37	Yes	Yes
746	40747	K38	Yes	Yes
748	40749	K39	Yes	Yes
750	40751	K40	Yes	Yes
752	40753	K41	Yes	Yes
754	40755	K42	Yes	Yes
756	40757	K43	Yes	Yes
758	40759	K44	Yes	Yes
760	40761	K45	Yes	Yes
762	40763	K46	Yes	Yes
764	40765	K47	Yes	Yes

Holding Register Address		Motor Parameter	Read Access	Write Access
Base Address	Modicon Address			
766	40767	K48	Yes	Yes
768	40769	K49	Yes	Yes
770	40771	K50	Yes	Yes
772	40773	K51	Yes	Yes
774	40775	K52	Yes	Yes
776	40777	K53	Yes	Yes
778	40779	K54	Yes	Yes
780	40781	K55	Yes	Yes
782	40783	K56	Yes	Yes
784	40785	K57	Yes	Yes
786	40787	K58	Yes	Yes
788	40789	K59	Yes	Yes
790	40791	K60	Yes	Yes
792	40793	K61	Yes	Yes
794	40795	K62	Yes	Yes
796	40797	K63	Yes	Yes
798	40799	K64	Yes	Yes
800	40801	K65	Yes	Yes
802	40803	K66	Yes	Yes
804	40805	K67	Yes	Yes
806	40807	K68	Yes	Yes
808	40809	K69	Yes	Yes

Holding Register Address		Motor Parameter	Read Access	Write Access
Base Address	Modicon Address			
810	40811	K70	Yes	Yes
812	40813	K71	Yes	Yes
814	40815	K72	Yes	Yes
816	40817	K73	Yes	Yes
818	40819	K74	Yes	Yes
820	40821	K75	Yes	Yes
822	40823	K76	Yes	Yes
824	40825	K77	Yes	Yes
826	40827	K78	Yes	Yes
828	40829	K79	Yes	Yes
830	40831	K80	Yes	Yes
832	40833	K81	Yes	Yes
834	40835	K82	Yes	Yes
836	40837	K83	Yes	Yes
838	40839	K84	Yes	Yes
840	40841	K85	Yes	Yes
842	40843	K86	Yes	Yes
844	40845	K87	Yes	Yes
846	40847	K88	Yes	Yes
848	40849	K89	Yes	Yes
874	40875	RESERVED	No	Yes

9.2.11 H Gain Parameters

Holding Register Address		Motor Parameter	Read Access	Write Access
Base Address	Modicon Address			
900	40901	H0	Yes	Yes
902	40903	H1	Yes	Yes
904	40905	H2	Yes	Yes
906	40907	H3	Yes	Yes
908	40909	H4	Yes	Yes
910	40911	H5	Yes	Yes
912	40913	H6	Yes	Yes
914	40915	H7	Yes	Yes

9.2.12 Timers

Holding Register Address		Motor Parameter	Read Access	Write Access
Base Address	Modicon Address			
950	40951	T0	Yes	No
952	40953	T1	Yes	Yes
954	40955	T2	Yes	Yes
956	40957	T3	Yes	Yes
958	40959	T4	Yes	Yes
960	40961	T5	Yes	Yes
962	40963	T6	Yes	Yes
964	40965	T7	Yes	Yes
966	40967	T8	Yes	Yes

9.2.13 Direct Control Registers

Holding Register Address		Motor Parameter	Read Access	Write Access
Base Address	Modicon Address			
1000	41001	CPUTime	Yes	No
1002	41003	ActualPosition	Yes	No
1004	41005	ActualTargetPosition	Yes	No
1006	41007	MotorStatus	Yes	No
1008	41009	ActualRatedCurrent	Yes	No
1010	41011	ActualOverloadTorque	Yes	No
1012	41013	AnalogIn	Yes	No
1014	41015	DigitalIn	Yes	No
1016	41017	Temperature	Yes	No
1018	41019	DCVoltage	Yes	No
1020	41021	N/A	Yes	No
1022	41023	ModeOfOperationDisplay	Yes	No
1024	41025	N/A	Yes	No
1026	41027	ActualSpeed	Yes	No
1028	41029	TargetPosition	Yes	Yes
1030	41031	TargetSpeed	Yes	Yes
1032	41033	TargetTorque	Yes	Yes
1034	41035	TargetAcceleration	Yes	Yes
1036	41037	TargetDeceleration	Yes	Yes
1038	41039	Controlword	Yes	Yes
1040	41041	ModeOfOperation	Yes	Yes

Holding Register Address		Motor Parameter	Read Access	Write Access
Base Address	Modicon Address			
1042	41043	DigitalOut	Yes	Yes

9.2.14 CM1-T Library Mapped Registers

The set of registers below are typically used when mapping the Modbus read and write messages directly to a data structure.

- This is often achieved using function code 0x23 (read/write in single message)
- The data structure should break the DINT register into the respective variables.
- The CM1-T Library has been developed to be used with IEC 61131-3 IDEs such as Codesys, Machine Expert and TwinCAT. See Examples and Demos.

Holding Register Address		Motor Parameter	Read Access	Write Access
Base Address	Modicon Address			
Read Registers				
1100	41101	us50Counter (DINT)	Yes	No
1102	41103	ActualPosition (DINT)	Yes	No
1104	41105	MotionTarget (DINT)	Yes	No
1106	41107	ActualVelocity (DINT)	Yes	No
1108	41109	MotorStatus (INT) PercentActualCurrent (INT)	Yes	No
1110	41111	PercentOverloadTorque (INT) AnalogIN (INT)	Yes	No
1112	41113	DCVoltage (INT) DigitalIO (SINT) Temperature (SINT)	Yes	No
1114	41115	ModeOfOperationDisplay (SINT) ErrorCode (SINT) ReadAddress1 (INT)	Yes	No
1116	41117	ReadValue1 (DINT)	Yes	No

Write Registers				
1200	41201	TargetPosition (DINT)	Yes	Yes
1202	41203	TargetVelocity (DINT)	Yes	Yes
1204	41205	TargetTorque (INT) TargetAcceleration (INT)	Yes	Yes
1206	41207	TargetDeceleration (INT) ControlWord (INT)	Yes	Yes
1208	41209	ModeOfOperation (SINT) DigitalOUT (SINT) WriteAddress1 (INT)	Yes	Yes
1210	41211	WriteValue1 (DINT)	Yes	Yes
1212	41213	ReadAddress1 (INT) PAD (INT)	Yes	Yes

10 TCP/UDP

10.1 Connections and Ports

Any TCP/IP enabled client device can connect and communicate with the CM1-T. The table below lists the ports and protocols they are associated with

Protocol	Port #	# TCP	# UDP	Description
CML	10001	1	1	Protocol used by the standard motor. Program the motor to run from IO, etc.
Direct Control Port	10002	1	1	Directly control the motor in a number of modes such as position, speed and torque • The CM1-T Library can be used with the Direct Control port
Modbus TCP	502	1	0	Modbus access to all motor registers including CML and Direct Control. • The CM1-T Library can be used with the Modbus TCP port
Motor Information Port	30718	0	1	1. Query the motors configuration (Network configuration, serial number, etc) 2. Query/Stream the motors information (status, position, speed, etc)
EtherNet/IP	44818 2222	2 0	1 1	EtherNet/IP CIP EtherNet/IP IO • The CM1-T Library can be used with the EtherNet/IP port

This section focus on the Direct Control Port and the Motor Information port. For information on the other protocols go to their specific section.

10.2 Direct Control Port

Port 10002 can be used to read and write the Direct Control registers. TCP/IP or UDP can be used.

10.2.1 Write Data Packet

This is the data received by the motor on a write from the TCP/UDP Client.

- TCP/IP or UDP
- Port 10002

- Length = 28 bytes

Data

Byte	Register	Size/Type	Description
0-3	TargetPosition	DINT	Final target position
4-7	TargetSpeed	DINT	Maximum targetspeed
8-9	TargetTorque	INT	Maximum torque
10-11	TargetAccelertion	INT	Acceleration (used when accelerating to target speed)
12-13	TargetDeceleration	INT	Deceleration (used when stopping)
14-15	Controlword	INT	Control the motor operation
16	ModeOfOperation	SINT	Set the required mode of operation
17	DigitalOUT	SINT	Set the 2 digital outputs. Requires K34=44.
18-19	WriteAddress1	INT	Address of the motor register to write to
20-23	WriteValue1	DINT	Value to be written to the motor register
24-25	ReadAddress1	INT	Address of the motor register to read
26-27	PAD	INT	16-bit padding not used

10.2.2 Read Data Packet

This is the data transmitted by the motor to the TCP/UDP Client when a write packet is received

- TCP/IP or UDP
- Port 10002
- Length = 36 bytes

Data

Byte	Register	Size/Type	Description
0-3	us50Counter	DINT	CPU clock time in 50us counts
4-7	ActualPosition	DINT	The actual position of the motor in encoder counts

Byte	Register	Size/Type	Description
8-11	MotionTarget	DINT	The actual instantaneous position the motor is currently moving to.
12-15	ActualSpeed	DINT	Actual speed of the motor in encoder counts/s
16-17	MotorStatus	INT	The motors status (error, homing, in position, etc)
18-19	PercentActualCurrent	INT	Percentage (0.1%) of rated current. 1000 = rated, 1100 = peak.
20-21	PercentOverloadTorque	INT	Percentage (0.1%) of overload torque.
22-23	AnalogIN	INT	10 bit analog input value (0-1023)
24-25	DCVoltage	INT	24V DC bus voltage in 0.1V
26	DigitalIN	SINT	Digital IN status. B0-B3 = IN1-IN4
27	Temperature	SINT	Drive temperature in °C
28	ModeOfOperationDisplay	SINT	Indicates the mode of operation currently set
29	ErrorCode	SINT	Error code on communication error
30-31	ReadAddress1	INT	Address of the motor register for ReadValue1
32-35	ReadValue1	DINT	The value in the motor register displayed in ReadAddress1

10.3 Motor Information Port

The motor information port is used to query the motor configuration data as well as all status data. The status data can also be set to stream out at defined time intervals.

- Port 30718
- UDP only

The motor receives a 4 byte data packet which indicates the type of request. The response includes a 4 byte header. The request and expected response header are defined as such:

	Command	B3 (Command ID)	B2	B1	B0
1	Configuration Request	0xF6	0x00	0x00	some data
	Configuration Response	0xF7	0x00	0x00	echo some data
2	Information Request	0xF4	Time ms high byte	Time ms low by	some data

	Information Response	0xF5	Counter high byte	Counter low byte	echo some data

10.3.1 Configuration Data

The configuration data responds with info regarding the motor's network configuration, firmware versions and serial number

To request the configuration data send the following data packet

Byte	Value	Description
0	0x00-0xFF	Any value. This will be echoed back on the response
1	0x00	Ignored
2	0x00	Ignored
3	0xF6	Command ID for configuration request

The response packet is returned as follows

Bytes	Description
0	Echo of some data
1-2	0x0000
3	0xF7 (Command ID)
4-11	N/A
12-15	Current IP Address
16-19	Subnet Mask
20-23	Gateway IP
24-29	MAC Address
30	DHCP Enabled
31	N/A
32-35	CM1-T Interface Firmware Version
36-39	CM1 Drive Firmware Version
40-99	Part Number (ascii)

Bytes	Description
100-103	Reserved
104-105	Production Date
106-107	Product ID
108-109	Serial Number
110-118	CM1-T Interface Hardware Version
119	Reserved

10.3.2 Information Data

The information request returns a complete set of the motor's state data including, position, speed, drive status, IO, etc

Send Data

To request the information data send the following data packet:

length = 4 bytes

Byte	Value	Description
0	0x00-0xFF	Any value. This will be echoed back on the response
1	0x00	Interval Time in ms low byte
2	0x00	Interval Time in ms high byte
3	0xF4	Command ID for information request

Interval Time

The Interval Time is a 16 bit value that sets the response interval in ms. It has the following characteristics

- Value range: [0-65536]
- If value = 0 then the query sends a single response
- If value > 0 then a response is sent at the interval time

Receive Data

The following data is received on a request or on an interval response

length = 33 bytes

Byte	Name	Description
0	Data echo	Echo the data sent in B0 of the request
1-2	Counter Value	Each response increments the counter value. A query resets the counter.
3	0xF5 - Command ID	Response command ID is 0xF5
4-7	CPUTime	CPU clock time in 50us counts
8-11	ActualPosition	The actual position of the motor in encoder counts
12-15	ActualTargetPosition	The actual instantaneous position the motor is currently moving to.
16-17	MotorStatus	The motors status (error, homing, in position, etc)
18-19	ActualRatedCurrent	Percentage (0.1%) of rated current. 1000 = rated, 1100 = peak.
20-21	ActualOverloadTorque	Percentage (0.1%) of overload torque.
22-23	AnalogIN	10 bit analog input value (0-1023)
24	DigitalIN	Digital IN status. B0-B3 = IN1-IN4
25	Temperature	Drive temperature in °C
26-27	DCVoltage	24V DC bus voltage in 0.1V
28	DigitalOUT	Digital OUT status. B0-B1 = OUT1-OUT2
29	Reserved	-
30	ModeOfOperationDisplay	Indicates the mode of operation currently set
31	ActualSpeed	Actual speed of the motor in encoder counts/s

10.4 Example Packets

The following example uses Packet Sender to execute some direct moves, query configuration and query information.

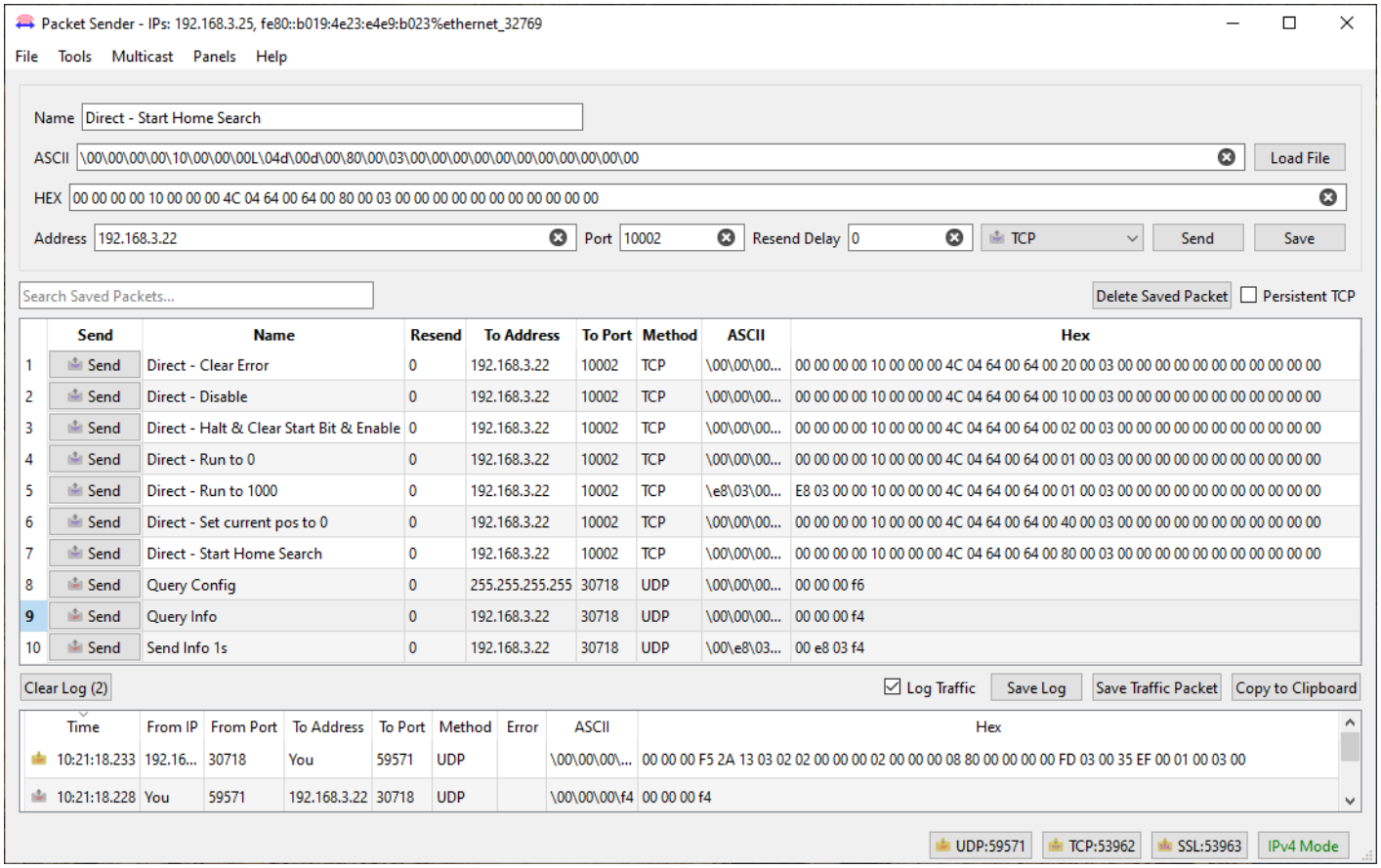
Please be aware of any firewall restrictions especially when using UDP.

Name	Send Data Bytes
Direct - Clear Error	00 00 00 00 10 00 00 00 4C 04 64 00 64 00 20 00 03 00 00 00 00 00 00 00 00 00 00 00
Direct - Disable	00 00 00 00 10 00 00 00 4C 04 64 00 64 00 10 00 03 00 00 00 00 00 00 00 00 00 00 00

Name	Send Data Bytes
Direct - Halt & Clear Start Bit & Enable	00 00 00 00 10 00 00 00 4C 04 64 00 64 00 02 00 03 00 00 00 00 00 00 00 00 00 00 00
Direct - Run to 0	00 00 00 00 10 00 00 00 4C 04 64 00 64 00 01 00 03 00 00 00 00 00 00 00 00 00 00 00
Direct - Run to 1000	E8 03 00 00 10 00 00 00 4C 04 64 00 64 00 01 00 03 00 00 00 00 00 00 00 00 00 00 00
Direct - Set current pos to 0	00 00 00 00 10 00 00 00 4C 04 64 00 64 00 40 00 03 00 00 00 00 00 00 00 00 00 00 00
Direct - Start Home Search	00 00 00 00 10 00 00 00 4C 04 64 00 64 00 80 00 03 00 00 00 00 00 00 00 00 00 00 00
Query Config	00 00 00 f6
Query Info	00 00 00 f4
Query Info 1s	00 e8 03 f4

- *Direct - Run* uses a speed = 10, acceleration = 100, deceleration = 1000 and Torque = 1100.

The image below shows Packet Sender with the response to Query Info



The Packet Sender database for the above example can downlaoded here - CM1-T Example.ini

11 CM1-T Library

11.1 Download

See Datasheets and Downloads

11.2 Introduction

The CM1-T Library is an add-on library that helps the user easily deploy, manage and run a CM1-T integrated motor. The CM1-T Library can be used with development environments based on the IEC 61131-3 standard. These IDEs will include Codesys™, Schneider EcoStruxure Machine Expert™, WAGO's **e!**COCKPIT™, Beckhoff's TwinCAT™ and more.

The CM1-T Library can be used with the following protocols

- EtherNet/IP
- Modbus/TCP
- TCP/IP
- UDP

The library conforms to the PLCopen Motion Control specification and implements the following function blocks:

Administration	Motion	Information	Homing	Digital I/O
• MC_Power • MC_Reset • MC_Stop	• MC_MoveAbsolute • MC_MoveRelative • MC_MoveAdditive • MC_MoveVelocity • MC_Halt	• MC_ReadActualPosition • MC_ReadActualVelocity • MC_ReadAcutalTorque • MC_ReadInfo • MC_ReadAxisError • MC_ReadParameter • MC_WriteParameter	• MC_Home • MC_SetPosition	• MC_ReadDigitalInput • MC_ReadDigitalOutput • MC_WriteDigitalOutput

11.3 Appendix

11.3.1 ErrorID

The ErrorID available on most function blocks has the following values

Value	Description
0	No error
1	Invalid mode of operation
2	Invalid input parameter

12 Datasheets and Downloads

12.1 CM1-T Configuration Tool

CM1-T Configuration Tool.exe

12.2 EtherNet/IP Support Files

CM1-T EIP_V13.eds

12.2.1 Allen-Bradley Specific

CM1-T_AOIs.zip

- Add-On Instructions
- User Data Types
- Example Application

12.3 CM1-T Library

CM1T.compiled-library

See the CM1-T Library documentation page for additional information.

The CM1-T library can be use in all IEC 61131-3 development environments such as Codesys, Schneider's Machine Expert, WAGO's **e!**COCKPIT, Beckhoff's TwinCAT and more.

The CM1-T Library can be used with the following protocols

- EtherNet/IP
- Modbus/TCP
- TCP/IP
- UDP

Version History

Version	Changes

V1.0.1.0	• Function blocks added
	  • MC_ReadDigitalInput
	  • MC_ReadDigitalOutput
	  • MC_WriteDigitalOutput

V1.0.0.2	• Updated WORD structure in DATA_IN and DATA_OUT for simplified Modbus TCP mapping
V1.0.0.1	• Initial release

12.4 MCAD

12.4.1 3D Models

For additional 3D formats please contact Myostat support (support@myostat.ca).

CM1-T-17L30D MCAD.zip

CM1-T-17S30D MCAD.zip

CM1-T-23S30D MCAD.zip

CM1-T-23L20D MCAD.zip

12.4.2 DXF Files

CM1-T-17X30D rev 1.1.DXF

CM1-T-23XX0D rev 1.1.DXF

12.5 Datasheets

12.5.1 CM1-T-17S30D / CM1-T-17L30D

CM1-T-17X30D rev 1.1.PDF

12.5.2 CM1-T-23S30D / CM1-T-23L20D

CM1-T-23XX0D rev 1.1.PDF

13 Examples and Demos

The following examples demonstrate and show how to use the CM1-T on a few different PLC platforms. All examples include application source code so they can be further explored.

The controllers use Direct Control to run the motor in position, speed or torque modes.

- Codesys™
  - EtherNet/IP
  - Modbus TCP
  - TCP/IP
- Schneider EcoStruxure Machine Expert™
  - EtherNet/IP
  - Modbus TCP
- Automation Direct - Do-more! Designer
  - Modbus TCP
- IDEC FL1F Smart Relay - WindLGC
  - Modbus TCP

13.1 Codesys™

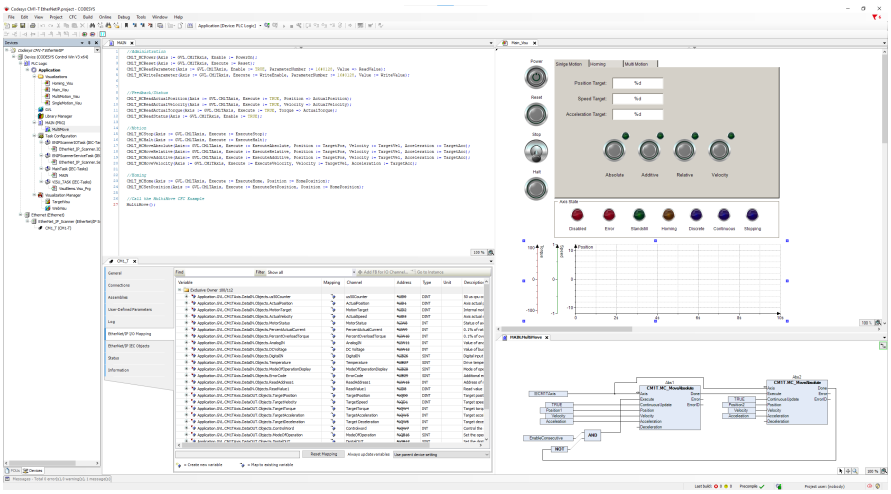
The CM1-T supports multiple protocols in Codesys.

- EtherNet/IP
- Modbus TCP
- TCP/IP

13.1.1 EtherNet/IP

CM1-T on Codesys PLC using EtherNet/IP

Codesys Control Win PLC controlling a CM1-T integrated servo using EtherNet/IP and the CM1T library



Downloads

Codesys Project - V1.0.1.0

- Codesys CM1-T EtherNetIP.zip

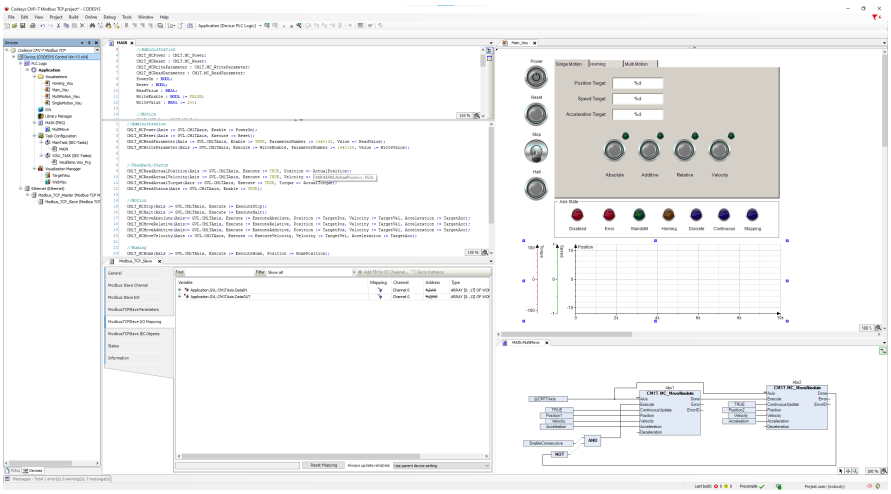
CM1-T library

- CM1-T Library

13.1.2 Modbus TCP

CM1-T on Codesys PLC using Modbus TCP

Codesys Control Win PLC controlling a CM1-T integrated servo motor using Modbus TCP and the CM1T library



Downloads

Codesys Project - V1.0.1.0

- CM1T_Demo.projectarchive

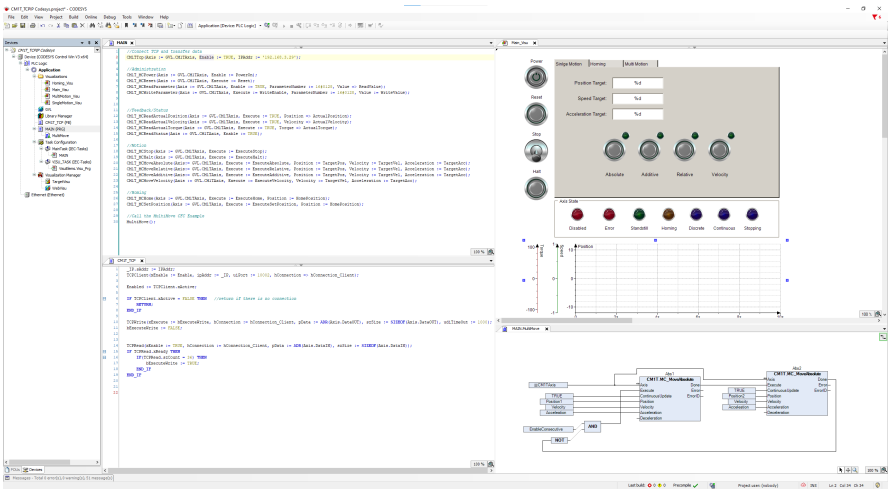
CM1-T library

- CM1-T Library

13.1.3 TCP/IP

CM1-T on Codesys PLC using TCP/IP

Codesys Control Win PLC controlling a CM1-T integrated servo motor using TCP/IP and the CM1T library



Downloads

Codesys Project - V1.0.1.0

- CM1T_TCPIP Codesys.projectarchive

CM1-T library

- CM1-T Library

13.2 Schneider EcoStruxure Machine Expert™

Schneider supports multiple protocols that can be used to control the CM1-T.

- EtherNet/IP
- Modbus TCP

13.2.1 EtherNet/IP

CM1-T connected to Schneider M241 Series PLC using EtherNet/IP

Sorry, the widget is not supported in this export.
But you can reach it using the following URL:

https://www.youtube.com/watch?v=TMNibrxGp2Y

This demonstration and example video shows how to connect a CM1-T Cool Muscle motor to a Schneider PLC over EtherNet/IP. The video further demonstrates using the CM1-T library to control and monitor the motor with standard PLCopen (MC) function blocks. The demonstration walks you through the following steps:

- Starting a Machine Expert project
- Adding the CM1-T EtherNet/IP device to the Device Repository
- Adding the CM1-T IEC 61131-3 PLCopen Motion Control library to the Library Repository
- Adding the Industrial Ethernet Manager Device which in this instance acts as the EtherNet/IP scanner
- Adding an EtherNet/IP Remote Adapter (CM1-T)
- Mapping the adapter remote IO to the CM1-T library instance
- Calling multiple library functions to run and control the CM1-T through visualizations.

Machine Expert Source Code

- CM1T_Demo.zip (Machine Expert V1.2)
- CM1T_Demo_V2.zip (Machine Expert V2.0)

Library Files

The CM1-T device and CM1-T library are required. The files can be downloaded here:

- CM1T.compiled-library
- CM1-T EIP_V13.eds

13.2.2 Modbus TCP

CM1-T connected to Schneider M241 Series PLC using Modbus TCP

Sorry, the widget is not supported in this export.
But you can reach it using the following URL:

https://www.youtube.com/watch?v=DwP_ZrjOeQs

This demonstration and example video shows how to connect a CM1-T Cool Muscle motor to a Schneider PLC using Modbus TCP. The video further demonstrates using the CM1-T library to control and monitor the motor with standard PLCopen (MC) function blocks. The demonstration walks you through the following steps

- Starting a Machine Expert project
- Adding the CM1-T IEC 61131-3 PLCopen Motion Control library to the Library Repository
- Adding the Industrial Ethernet Manager Device which in this instance acts as the Modbus TCP Master
- Adding a Generic Modbus TCP Slave to connect to the CM1-T as a Modbus Slave
- Creating and setting up the Modbus channels with Read/Write Multiple Registers (Function Code 23)
- Mapping the Modbus channel data to the CM1-T variable instance
- Calling multiple library functions to run and control the CM1-T through visualizations.

Machine Expert Source Code

- Schneider - CM1-T MDB_TCP.zip (Machine Expert V2.0)

Library Files

The CM1-T device and CM1-T library are required. The files can be downloaded here:

- CM1T.compiled-library

13.3 Automation Direct - Do-more! Designer

Automation Direct BRX PLCs can use the following protocols to control the CM1-T.

BRX PLCs programmed with Do More! Designer can be used to directly control a CM1-T.

13.3.1 Modbus TCP

CM1-T connected to BRX PLC using Modbus TCP

Sorry, the widget is not supported in this export.
But you can reach it using the following URL:

https://www.youtube.com/watch?v=zw63hCQSEM8

This demonstration and example video shows how to connect a CM1-T Cool Muscle motor to a BRX PLC over Modbus-TCP using the Do-more! Designer development environment. The video demonstrates using the CM1-T Modbus user profile and the supplied subroutine to simplify development and control of the CM1-T. The demonstration walks you through the following steps

- Saving the Modbus TCP user profile for use in all Do-more! Designer projects
- Importing an existing project to copy the CM1-T subroutine
- Adding a Modbus TCP module. In this case the CM1-T integrated Cool Muscle motor.

- Adding and calling a subroutine in Do-more! Designer including passing parameters in and out of the subroutine.
- Creating logic to power and run the CM1-T integrated Cool Muscle motor
- Running a simulation to validate and test the developed logic

Do-more! Designer Files

- CM1-T Do-more Designer Files.zip

13.4 IDEC FL1F Smart Relay - WindLGC

IDEC FL1F Smart Relay using WindLGC and the Siemens LOGO! PLC using LOGO! can control a CM1-T. The demos can be opened in WindLGC and LOGO!

13.4.1 Modbus TCP

IDEC FL1F Smart Relay - WindLGC

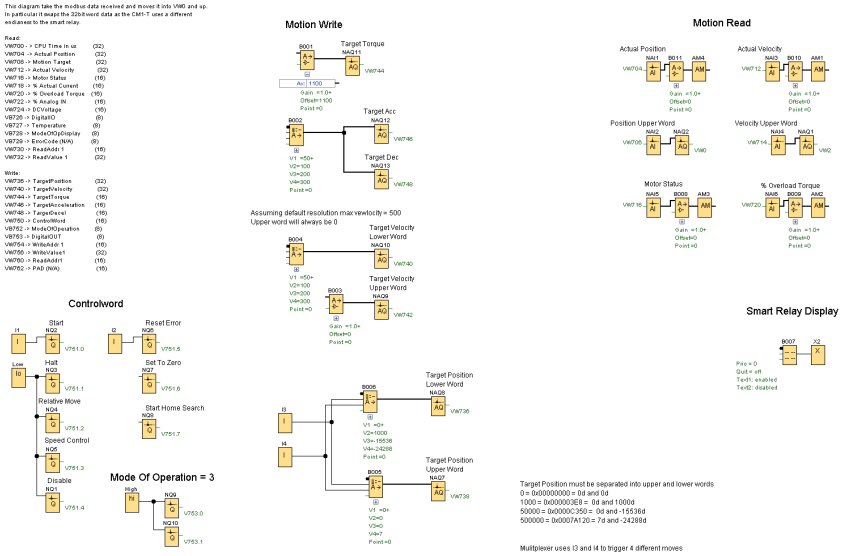
(Siemens LOGO! PLC - LOGO!)

Modbus TCP - CM1-T connected to FL1F using Modbus TCP

The IDEC FL1F smart relay can be setup as a Modbus TCP Client (master). The Smart Relay is able to control the motor by directly writing to motion registers. The attached example sets the following

- I1 - start
- I2 - Reset from error state (over torque, etc)
- I3 + I4 - input combination executes any of 4 absolute position moves
- Display Output
    - Position (16bit as per PLC restrictions)
    - Speed
    - Torque
    - Motor Status

The FL1F can be setup to execute the motor using position, velocity or torque modes.



WindLGC Source Files

- CM1-T and Smart Relay Example.mnp

14 CM1-T Certifications

14.1 CE Declaration

- CM1-E / CM1-T CE Declaration of Conformity
- CM1M9 CE Declaration of Conformity