www.jvl.dk

Industrial Ethernet

User Manual

MAC and MIS Motors generation III

Integrated Servo and Stepper Motors

EtherCAT, EtherNet/IP and PROFINET — POWERLINK (beta)

Preliminary Edition



MAC G3 (left, right) and MIS G3 (centre) — same Ethernet interface, two motor families

Litt. no.: LB0077-10GB - Revised 10. th. July 2026 firmware v3.1.1.0

Important
User Information



Warning



The MAC series of products are used to control electrical and mechanical components of motion control systems.
You should test your motion system for safety under all potential conditions. Failure to do so can result in damage to equipment and/or serious injury to personnel.

Please contact your nearest JVL representative in case of technical assistance. Your nearest contact can be found on our web site www.jvl.dk

Copyright 1998-2026, JVL A/S. All rights reserved.
This user manual must not be reproduced in any form without prior written permission of JVL A/S.
JVL A/S reserves the right to make changes to information contained in this manual without prior notice.
Similarly JVL A/S assumes no liability for printing errors or other omissions or discrepancies in this user manual.

MacTalk and MotoWare are registered trademarks

JVL A/S
Bregnerødvej 127
DK-3460 Birkerød
Denmark
Tlf. +45 45 82 44 40
Fax. +45 45 82 55 50
e-mail: jvl@jvl.dk
Internet: http://www.jvl.dk

1 Introduction 3
1.1 Introduction 4
1.2 Ethernet on generation III motors 5
1.3 Selecting the protocol 6
1.4 Finding the FW/HW version 7

2 Common setup and functions 9
2.1 Introduction 10
2.2 Status indicators (LEDs) 11
2.3 Register numbering 14
2.4 Diagnosis 16
2.5 IP configuration 17
2.6 MacTalk over Ethernet 20
2.7 Firmware update over Ethernet 21
2.8 Motor reaction on lost connection 22
2.9 Web server 23
2.10 I/O junction box PA0190 24
2.11 Terms and abbreviations 27

3 EtherCAT (beta) 29
3.1 Introduction 30
3.2 Protocol specifications 32
3.3 Status indicators 37
3.4 Diagnosis 38
3.5 Profiles 40
3.6 CoE object dictionary 41
3.7 JVL register profile 43
3.8 CiA402 drive profile (beta) 45
3.9 Distributed Clocks 55
3.10 ESI file 57
3.11 Ethernet over EtherCAT 58
3.12 Commissioning 59
3.13 Examples 61

4 EtherNet/IP 63
4.1 Introduction 64
4.2 Status indicators 66
4.3 Cyclic data 67
4.4 CIP objects 70
4.5 Explicit messaging 72
4.6 EDS file and AOI 74
4.7 Implementation guidelines 75
4.8 Setup via explicit messages 77
4.9 Ethernet switches 78
4.10 Commissioning 79
4.11 Examples 81
4.12 ODVA conformance 83

5 Modbus TCP/UDP 85
5.1 Introduction 86
5.2 Status indicators 87
5.3 Connection 88
5.4 Modbus address 89
5.5 Register access 90
5.6 Function codes 92
5.7 Exception responses 95
5.8 Response time 96
5.9 Quick start 97
5.10 Examples 99

6 POWERLINK (beta) 103
6.1 Introduction 104
6.2 Protocol specifications 106
6.3 Status indicators 109
6.4 Commissioning 110

Litt. no.: LB0077-10GB - Revised 10. th. July 2026 firmware v3.1.1.0

6.5 POWERLINK objects 112
6.6 NMT services 116
6.7 XDD file 117
6.8 Troubleshooting 118
6.9 Examples 119

7 PROFINET (beta) 121

7.1 Introduction 122
7.2 Status indicators 124
7.3 Cyclic data 125
7.4 Acyclic data 127
7.5 PROFIdrive 130
7.6 Diagnosis 132
7.7 Conformance and timing 133
7.8 I&M functions 134
7.9 GSDML file 135
7.10 Ethernet switches 136
7.11 Byte order 137
7.12 Name and IP address 138
7.13 Commissioning 139
7.14 Quick start with TIA Portal 141
7.15 Examples 143

8 Using MacTalk over Ethernet 145

8.1 Introduction 146
8.2 Setting up the PC 148
8.3 Connecting from MacTalk 149
8.4 The MacTalk TCP tab 151
8.5 Ethernet over EtherCAT 152
8.6 Firmware update over Ethernet 155
8.7 Troubleshooting 157

9 Examples common to all protocols 159

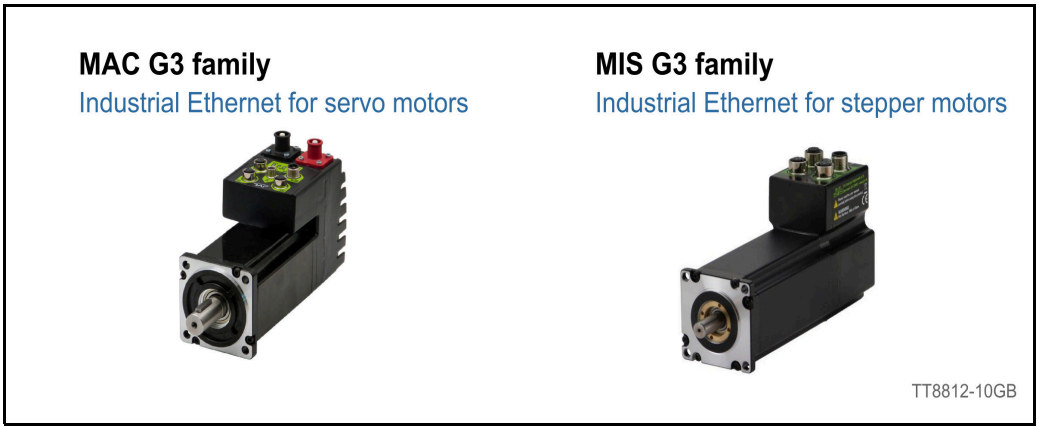
9.1 Introduction 160
9.2 ePLC and the Ethernet interface 161
9.3 Reset errors from the master 162
9.4 ePLC control from the master 164

10 Technical data 165

10.1 Introduction 166
10.2 Ethernet interface 167
10.3 Protocol data 169
10.4 User I/O 172

1 Introduction

1.1 Introduction



Industrial Ethernet is becoming more and more popular as it offers

- Very fast response time
- Predictable delay times (deterministic protocols)
- Safe transmission of data

Compared with most of the classic, non Ethernet based protocols, Industrial Ethernet offers state of the art performance.

On generation III motors the interface is built into the motor electronics and cannot be removed. The protocol is selected via the Interface option in the Product Part Number Builder (PPNB) when the motor is ordered, and each motor runs exactly one real-time protocol:

- EtherCAT - JVL register profile and CiA402 (DSP402) drive profile (beta)
- EtherNet/IP - cyclic register I/O
- PROFINET IO - cyclic register I/O and PROFIdrive (beta)
- Ethernet POWERLINK - JVL register profile and CANopen DS 301 (beta, MAC G3 only in this release)

Modbus TCP/UDP, MacTalk over Ethernet and a built-in web server are available on all variants, alongside the selected real-time protocol.

Main Features:

- Industrial Ethernet built into the motor electronics
- Two M12 4-pin D-coded Ethernet ports, In and Out, for daisy chaining
- One protocol per motor, selected at order time in the PPNB
- Two independent IP addresses: the protocol stack, and MacTalk / Modbus TCP
- Modbus TCP/UDP and MacTalk over Ethernet (port 47100) alongside the protocol
- Built-in web server ("MacTalk Light") on HTTP port 80
- Fully user configurable cyclic process data, one register list per direction
- Access to all motor and Ethernet interface registers. No pre-setup of the motor needed
- Five status LEDs for power, error and the two Ethernet ports
- Configurable reaction on lost connection: passive, controlled stop or ignore
- Firmware update over Ethernet (PROFINET, EtherNet/IP) or over RS485
- 8 digital I/O points (17-pin) or 3 inputs and 1 output (8-pin), by type number

Preliminary / beta: The Industrial Ethernet functionality described in this manual is released as a beta version. In particular the EtherCAT® CiA402 drive profile, PROFINET® and POWERLINK® are still being finalised, and some error situations may currently be visible only through MacTalk and the power LED. Verify critical behaviour in your own installation before using it in production.

1.2 Ethernet on generation III motors

The Ethernet interface consists of a CPU A communication processor and a flexible print that carries the Ethernet signals from the controller to the motor connectors. CPU A runs the selected real-time protocol stack and exchanges data with the motor's main processor over an internal link.
This is different from previous MAC motors, where the Ethernet interface was a separate MAC00-Ex unit that could be exchanged. On generation III nothing is exchangeable: the interface, and therefore the protocol, is a fixed part of the ordered motor.

1.3 Selecting the protocol

The protocol is selected via the Interface option in the Product Part Number Builder (PPNB) when the motor is ordered. The motor is delivered with the firmware for that protocol installed, and the type number and the label on the motor state which protocol it is.

The protocol is not a setting in the motor: each protocol is a separate firmware image for the Ethernet interface, and register 558 (16433 in MacTalk on MAC G3) only reports which one is installed. The firmware file that JVL supplies contains the images for all protocols, so MacTalk can install a different protocol on a motor that has already been delivered - over Ethernet when the motor uses PROFINET or EtherNet/IP®, or over the RS485 connection (Modbus® RTU, 115200 baud by default) on any protocol.

Note that the motor then no longer matches its own type number and label, and that the configuration files for the new protocol (ESI, GSDML or EDS) must be used instead. For a permanent change of protocol, order a motor for that protocol.

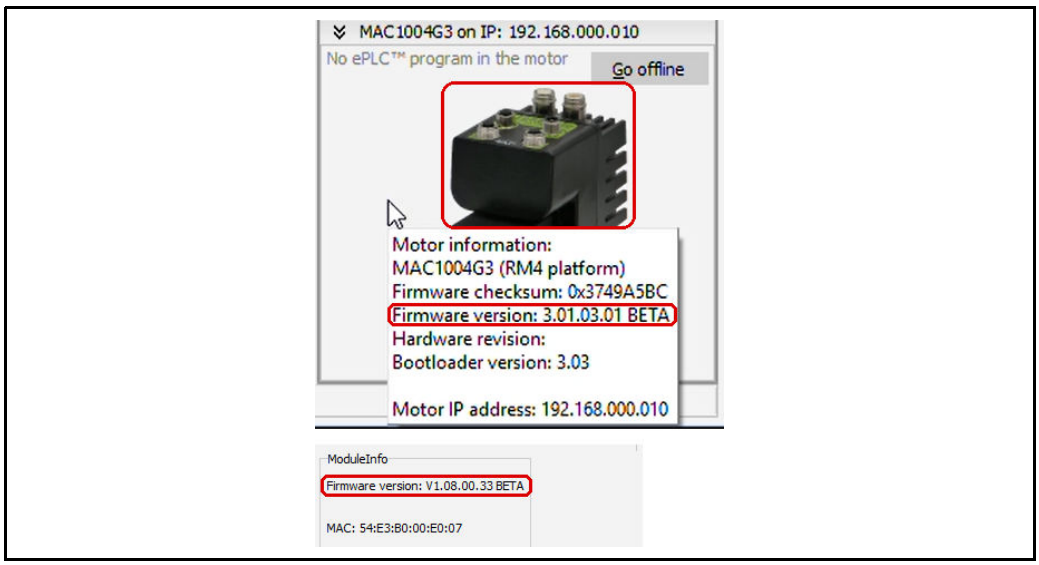
1.4 Finding the FW/HW version

Quote the firmware version in any support request - on MAC G3 both the motor firmware version and the firmware version of the Ethernet interface. The versions can be read in MacTalk, and they are also available as registers.

1.4.1 See in MacTalk

Connect MacTalk to the motor and hold the mouse pointer over the motor picture at the bottom right of the MacTalk window. The Motor information box shows the motor firmware version. On MAC G3 the Ethernet interface runs its own firmware; its version is shown on the tab of the protocol, under ModuleInfo (Firmware version) - quote both in a support request. On MIS G3 the motor firmware includes the Ethernet interface.

MAC G3:



MIS G3:



1.4.2 See via motor registers

When MacTalk is not available - over the selected real-time protocol, over Modbus TCP/UDP or from a controller - the same information is in the registers. The registers differ

between the two products.

MAC G3:

- Register 543 (16418 in MacTalk) - the firmware version of the Ethernet interface. It is an Ethernet interface register, so read it over Ethernet (the real-time protocol or Modbus TCP/UDP), see Register numbering in Common setup and functions. Over the RS485 connection register 543 is a register of the motor controller and reads 0.
- Register 250 - the motor firmware version.
- Register 205 - the hardware revision. The revision is also printed on the motor type label.

MIS G3:

- Register 250 - the firmware version, which includes the Ethernet interface.
- Register 156 - the hardware revision: bits 31-24 hold the PCB major revision, bits 23-16 the PCB minor revision and bits 15-8 the component revision. The label shows them as x.yz, so 0x01020000 is revision 1.20.

Both products:

- Register 558 (16433 in MacTalk on MAC G3) - which protocol is installed.

The layout of each register is described in the online register overview:
MAC G3: www.jvl.dk/register.html?register=MACmotor
MIS G3: www.jvl.dk/register.html?register=Registerlist_SMC67_MASTER

[EDITORIAL NOTE - NOT FOR RELEASE] ADO #2108 KT

This manual describes the Ethernet firmware released as 1.8.0.2 together with motor firmware 3.1.2.0.

2 Common setup and functions

2.1 Introduction

This chapter describes the setup and the functions that are common to all the Industrial Ethernet protocols: the status indicators, register numbering, diagnosis, IP configuration, MacTalk over Ethernet, firmware update over Ethernet, the motor reaction on lost connection, the web server, the I/O junction box PA0190, and the terms and abbreviations used in this manual.

2.2 Status indicators (LEDs)

The LED indications follow JVL's shared LED definition for MAC G3 and MIS G3. The motor has five status indicators, and all five are present on every motor. The two product families name them differently: on MAC G3 they are PWR, ERR, RUN, LA1 and LA2, and on MIS G3 they are PWR, ERR, L1, L2 and L3. The names map one to one in that order - RUN and L1 are the same indicator, as are LA1 and L2, and LA2 and L3. Their physical order and what each one reports are shown in figure TT8800-10GB, which lists both sets of names. What the four communication LEDs show depends on the protocol in use. The states for each protocol, and the blink patterns that belong to them, are given under Status indicators in that protocol's own chapter.

Status indicators - all five LEDs TT8800-10GB

All five indicators are on every motor. MAC G3 and MIS G3 name them differently.

MAC / MIS	Colours	What it reports
PWR	green, orange, red	Power and overall motor status
ERR	red	Protocol error. Off = no error
RUN / L1	green	Protocol state
LA1 / L2	green, yellow	Link and activity, port 1
LA2 / L3	green, yellow	Link and activity, port 2

RUN and L1 are the same indicator, as are LA1/L2 and LA2/L3.
Source: LED_description_MACG3_MISG3 v0.2 (2026-08-12), Common_MIS_MAC.

The motor signals its operating state through the LEDs on the motor itself. Reading them is the quickest way to establish what the motor is doing before a PC or a controller is connected.

The PWR LED shows the overall state of the motor and Ethernet interface; the highest-priority active condition is shown (see figure TT4084-10GB).

PWR (power / status) LED TT4084-10GB

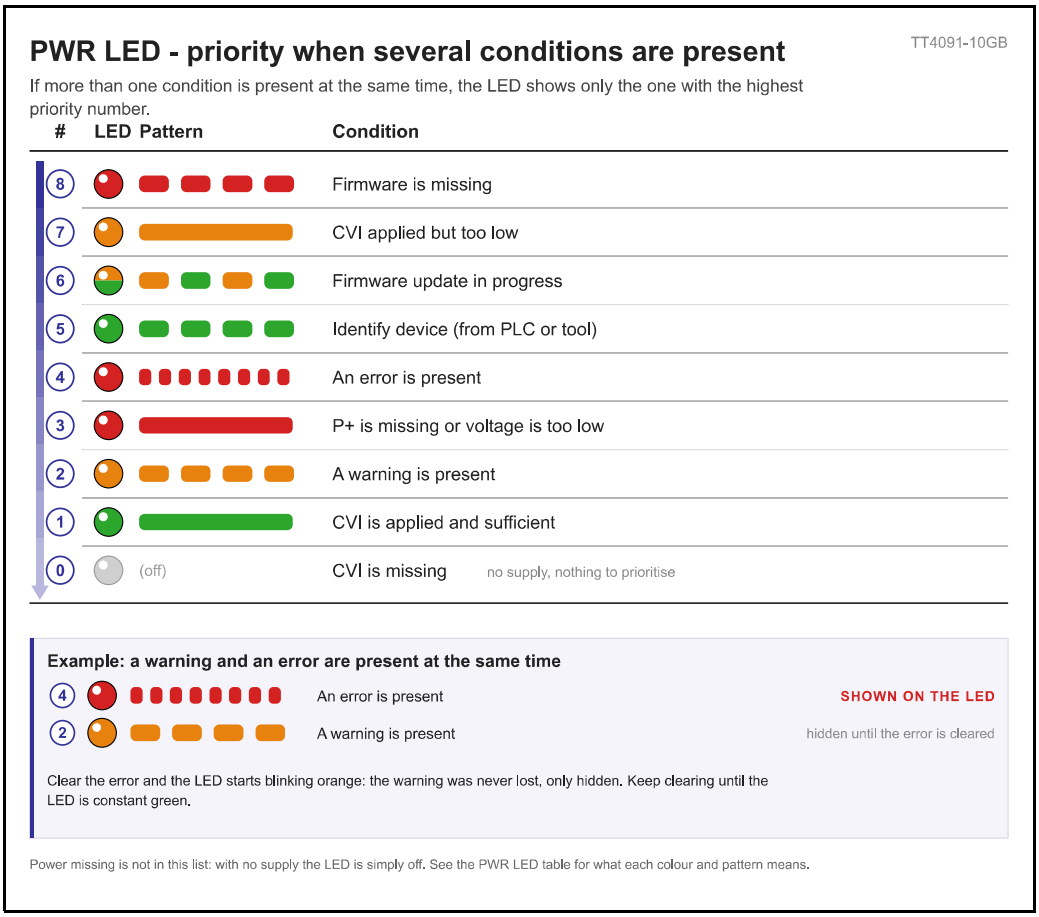
LED	Pattern	Description	Action
	(off)	CVI is missing	Connect CVI
		CVI is applied and sufficient	-
		P+ is missing or voltage is too low	Connect or increase P+
		CVI applied but too low	Increase CVI voltage
		Identify device (from PLC or tool)	-
		Firmware is missing	Connect MacTalk, update firmware
		A warning is present	-
		Firmware update in progress	-
		An error is present	Clear the error

Pattern key: solid = constant | dashes = blinking (200 ms on/off, 2.5 Hz) | dots = flickering (approx. 50 ms, 10 Hz).

A warning shown here excludes P+ and CVI warnings, which have their own constant-colour rows above. When several conditions are present at once, only one is shown - see the priority figure. Source: LED_description_MACG3_MISG3 v0.2.

2.2 Status indicators (LEDs)

When more than one condition is present at the same time, the PWR LED shows only the one with the highest priority. The priority order, and what each combination means, is shown in figure TT4091-10GB.



The LED tables in this chapter use the blink patterns below. Where a protocol chapter gives its own pattern definitions, those apply to that protocol:

- Constant: the LED is on continuously.
- Blinking: 200 ms on, 200 ms off (2.5 Hz).
- Single flash: 200 ms on, followed by 1 s off.
- Double flash: two pulses of 200 ms, followed by 1 s off.
- Flickering: rapid on/off at approximately 50 ms (10 Hz).

Only the combinations listed in the tables are used. The PWR LED can physically produce other combinations of colour and pattern, but they are not used by the current firmware and carry no meaning. If a pattern is observed that is not in the tables, note exactly what was seen and contact JVL rather than trying to interpret it.

Source: shared LED description LED_description_MACG3_MISG3 v0.3 (2026-09-15), Common_MIS_MAC; the PWR states in figure TT4084-10GB follow that document. It lists only the MIS G3 names; the MAC G3 names RUN, LA1 and LA2 given here come from the products themselves.

The PWR LED is a multi-colour indicator that reports the overall condition of the motor. As a rule:

- Green indicates power applied and normal operation.
- Orange indicates information or a warning, and the pattern decides which: a

constant orange LED means the control voltage (CVI) is applied but too low, whereas a blinking orange LED means a warning is present.

- Red indicates a fault that prevents normal operation.

2.2.1 See in MacTalk

MacTalk shows the same information in words. When the motor is connected, the protocol state and any error or warning are shown directly, so the LEDs do not have to be read at all - which is the usual way to work during commissioning.

2.2.2 See via motor registers

From a controller, or over Modbus® TCP/UDP, the underlying state is in the registers. Note that there is no register that reports the LEDs themselves: the registers report the state that the LEDs also show.

- Register 552 (16427 in MacTalk on MAC G3), STATE_MACHINE_STATUS (read only) - the state of the protocol state machine. The meaning of the value depends on the protocol, so read it together with the state description in the chapter for the protocol in use.
- Register 560 (16435 in MacTalk on MAC G3), ETHERNET_DIAGNOSTICS (read only) - the diagnostic code the interface records when it logs an error or a warning.
- Register 558 (16433 in MacTalk on MAC G3), PROTOCOL (read only) - which protocol is installed, which decides how the four communication LEDs are to be read in the first place.

2.3 Register numbering

The motor has two groups of registers. The motor registers hold the motion parameters and the state of the motor itself. The Ethernet interface registers hold the settings of the interface: the IP addresses, the protocol setup and the cyclic data mapping. The two groups are not numbered the same way on the two products, and that is the thing that most often goes wrong when a program written for one product is pointed at the other.
On the MIS G3 there is one register list and one numbering. The motor registers are 0 to 508, and the Ethernet interface registers follow from 509 upwards in the same list. A register number is the address, and nothing has to be converted.
On the MAC G3 the two groups are in different places: the motor registers are in the motor controller and the interface registers are in the Ethernet interface. They are addressed separately, and the interface registers carry an offset of 509, because the interface register that a protocol calls number 0 is number 509 in the shared register list.
In the two tables below, N is the register number in the shared list, which is the number this manual uses. Each table gives the address or the object that reaches register N over each protocol. The first table is the MAC G3, where the two groups are addressed separately. The second is the MIS G3, where one list serves both.

Table 1: MAC G3 - the two register groups

Protocol	Motor register N	Ethernet interface register N
Modbus TCP/UDP	address 2N	address 2(N + 15875)
EtherCAT	0x2012 sub N	0x2011 sub N - 509
EtherNet/IP	class 0x64, instance N	class 0x65, instance N
POWERLINK	0x2012 sub N	0x2011 sub N - 509
PROFINET	0x12nn, nn = N	0x11nn, nn = N - 509

Two limits are left out of the first table to keep it short. Object 0x2012 and index 0x12nn reach the first motor registers only; the motor registers above those are reached through object 0x2013 and index 0x13nn, which continue the count from there. The exact ranges differ a little between the protocols, and the chapter for the protocol in use gives them.

Table 2: MIS G3 - the one register list

Protocol	Register N
Modbus TCP/UDP	address 2N
EtherCAT	0x2001 sub N; 0x2002 sub N - 254; 0x2003 sub N - 508
EtherNet/IP	class 0x64, instance N
PROFINET	index N

On the MIS G3 there is nothing to convert for Modbus TCP/UDP, EtherNet/IP® and PROFINET®: the address, the instance number or the index is the register number. EtherCAT® is the exception. It divides the one list across three objects, and they are not the same ones it uses on a MAC G3: it uses 0x2001, 0x2002 and 0x2003, which a MAC G3 does not have at all.
The offset on the MAC G3 is not applied the same way by every protocol, and this is worth reading twice. EtherCAT, POWERLINK and PROFINET count their interface registers from the first one, so the 509 is subtracted before the number is used: interface register 512 is sub-index 3 of object 0x2011 and index 0x1103. EtherNet/IP subtracts nothing: the instance number is the number in the shared list, so the first interface register is instance 509 and not instance 0. Check which of the two a protocol uses before converting an address by hand.
Over Modbus TCP/UDP on the MAC G3 the constant 15875 holds for the interface registers 509 to 799. The interface registers from 1016 upwards use 16384 instead. The numbers between those two windows are not registers, and an access there is refused.

2.3 Register numbering

On older interface firmware it was not refused but reached a motor register instead, so never write to an address in that range.
MacTalk also has its own numbers for the MAC G3 interface registers. They follow the Modbus rule without the factor two: interface register N is register N + 15875 in MacTalk from 509 upwards, and N + 16384 from 1016 upwards. The Modbus address, interface register 641, is therefore register 16516 in MacTalk, and the communication setup register 515 is 16390. Register 641 in MacTalk is a register of the motor controller, not the Modbus address. Type the MacTalk number in the Watch Registers list or the Register Workspace Pad. On the MIS G3 register N is register N in MacTalk as well.
Register numbers given in this manual are the numbers of the shared list. Where a protocol needs a different number, the chapter for that protocol says so.

2.4 Diagnosis

Every error and warning the motor records carries four things: the class, which is either an error or a warning; the origin, which is the part of the motor that reported it; the category, which says what kind of condition it is; and a number that identifies the exact condition. The same structure is used on every protocol, so a diagnosis read from a controller and the same event read in MacTalk describe one thing.
The origins are:

Table 3: Origin of a diagnosis

Origin	What it covers
Core	The shared firmware: start-up, stored settings and internal communication
Motor	The motor itself: supply voltage, temperature, encoder, driver and overload
Module	The Ethernet interface (MAC G3)
Slave	The motor controller behind the interface (MAC G3)
Functional safety	The safety channel
Protocol	The real-time protocol in use. One protocol is present at a time

MacTalk shows the active errors and warnings in words, which is the quickest way to read them during commissioning. From a controller, or over Modbus TCP/UDP, the last diagnostic code is in register 560 (16435 in MacTalk) on the MAC G3 and in register 153 on the MIS G3. Register 35 holds the status and error bits.
How a diagnosis reaches the controller depends on the protocol: see Diagnosis in the chapter for the protocol in use. Modbus TCP/UDP has no diagnosis mechanism of its own, so read the registers instead.

2.5 IP configuration

2.5.1 General description

The motor has two independent IP addresses: one for the Industrial Ethernet protocol stack (used by EtherNet/IP, POWERLINK and PROFINET; EtherCAT does not use it) and one for the MacTalk / Modbus TCP interface (used by MacTalk over Ethernet and Modbus TCP/UDP). Both are stored in the motor and are preserved across a firmware update. The motor does not check the two addresses against each other, so give them different addresses.

Static addressing is the default for both interfaces. The factory default addresses are derived from the motor's own MAC address: the last byte of the first MAC address, as printed on the motor label, becomes the last octet of both addresses. If that byte is zero, 1 is used instead. Writing N for that byte, a motor leaves the factory with:

- Protocol stack: 10.10.10.N with subnet mask 255.255.255.0. The gateway is 10.10.10.N on EtherNet/IP and PROFINET, and 0.0.0.0 on EtherCAT. On POWERLINK the address is always 192.168.100.<node ID>, see Node ID in POWERLINK.
- MacTalk and Modbus TCP/UDP: 192.168.0.N with subnet mask 255.255.255.0. The gateway is 192.168.0.N on EtherNet/IP, PROFINET and POWERLINK, and 0.0.0.0 on EtherCAT.

On EtherNet/IP only, the interface also checks the stored protocol stack configuration when it starts: if the address is 0.0.0.0, ends in .0 or is 224.0.0.0 or higher, or the subnet mask is 0.0.0.0 or 255.255.255.255, it substitutes 192.168.1.1 with subnet mask 255.255.255.0 and no gateway, and writes these values to registers 512-514 (16387-16389 in MacTalk on MAC G3). The gateway is not checked. EtherCAT and PROFINET have no such fallback.

DHCP is available for the protocol stack on EtherNet/IP only:

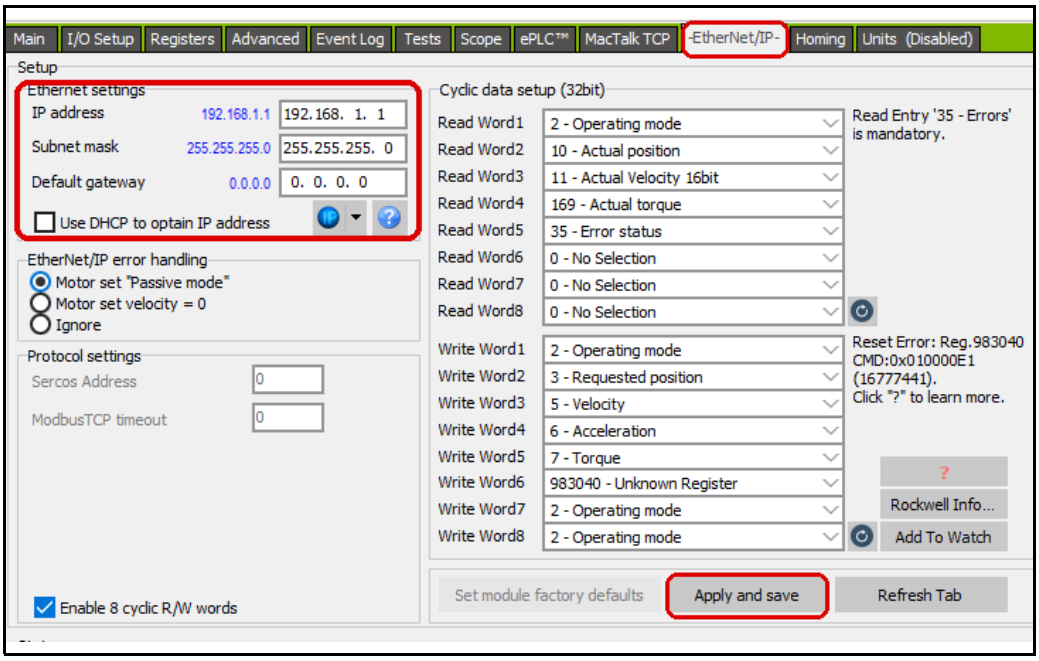
- EtherNet/IP - DHCP can be enabled, either by setting bit 10 of the communication setup register (515, 16390 in MacTalk on MAC G3) or by writing the Configuration Control attribute of the CIP TCP/IP Interface object (class 0xF5, attribute 3). Both routes end up in the same place: when the setting is changed over EtherNet/IP the motor writes it back into register 515 and saves it, so the choice survives a power cycle.
- PROFINET - DHCP is not available. The station name and the IP address of the protocol stack are normally assigned by the controller through the standard PROFINET DCP mechanism (see the PROFINET section).
- EtherCAT - DHCP is not available; use static addressing.

The MacTalk / Modbus TCP interface always uses static addressing. Register 515 (16390 in MacTalk on MAC G3) contains a separate DHCP-enable bit for it (bit 9), but that function is not implemented in this release and the bit has no effect.

2.5 IP configuration

2.5.2 Configuration via MacTalk

On the tab of the protocol in MacTalk (for example the EtherNet/IP tab), enter the IP address, subnet mask and default gateway under Ethernet settings, then press Apply and save. The address of the MacTalk / Modbus TCP interface is set on the MacTalk TCP tab, see The MacTalk TCP tab in Using MacTalk over Ethernet.



2.5.3 Configuration via motor registers

When the configuration is done by channels other than MacTalk, such as Ethernet, Profibus, CAN-open, etc., or just a simple interface connection to the basic motor, the following description must be followed.

Industrial Ethernet protocol stack (the address used by the selected real-time protocol):

- Register 512 (16387 in MacTalk on MAC G3) - IP address.
- Register 513 (16388 in MacTalk on MAC G3) - subnet mask.
- Register 514 (16389 in MacTalk on MAC G3) - gateway.

MacTalk / Modbus TCP interface (the address used by MacTalk over Ethernet and Modbus TCP/UDP):

- Register 553 (16428 in MacTalk on MAC G3) - IP address.
- Register 554 (16429 in MacTalk on MAC G3) - subnet mask.
- Register 555 (16430 in MacTalk on MAC G3) - gateway.

Write the values and save them to flash so they are retained after power-off.

Example

A motor is to sit at 192.168.10.50 on the protocol stack and at 192.168.10.51 for MacTalk and Modbus TCP, on a network with subnet mask 255.255.255.0 and gateway 192.168.10.1. Each address occupies one 32-bit register with the first octet in the most significant byte, so 192.168.10.50 is written as 0xC0A80A32. Six registers are written:

- Register 512 (16387 in MacTalk on MAC G3) = 0xC0A80A32 - protocol stack IP address 192.168.10.50.
- Register 513 (16388 in MacTalk on MAC G3) = 0xFFFFFF00 - protocol stack subnet mask 255.255.255.0.

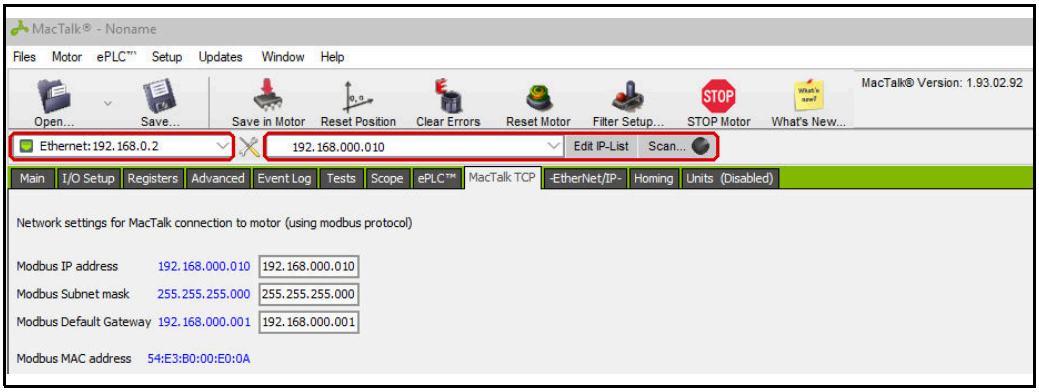
- Register 514 (16389 in MacTalk on MAC G3) = 0xC0A80A01 - protocol stack gateway 192.168.10.1.
- Register 553 (16428 in MacTalk on MAC G3) = 0xC0A80A33 - MacTalk and Modbus TCP/UDP IP address 192.168.10.51.
- Register 554 (16429 in MacTalk on MAC G3) = 0xFFFFFF00 - MacTalk and Modbus TCP/UDP subnet mask 255.255.255.0.
- Register 555 (16430 in MacTalk on MAC G3) = 0xC0A80A01 - MacTalk and Modbus TCP/UDP gateway 192.168.10.1.

Save the registers to flash afterwards. The protocol stack reads its IP configuration when it starts, so power-cycle the motor for the new address to take effect.

For PROFINET the IP address may instead be assigned by the IO controller via DCP, in which case the value written here is overwritten by the controller. See the PROFINET section.

2.6 MacTalk over Ethernet

MacTalk can connect to the motor over Ethernet: it finds the motor on UDP port 47100 and then works over Modbus UDP on port 502, see Using MacTalk over Ethernet. This allows full setup and monitoring over the same cable as the real-time protocol, and on EtherNet/IP and PROFINET also firmware update.



2.7 Firmware update over Ethernet

The motor and Ethernet firmware can be updated over Ethernet from MacTalk when the motor uses PROFINET or EtherNet/IP. Firmware update over EtherCAT is not available in this release, but MacTalk over the RS485 connection can update the firmware regardless of which Industrial Ethernet protocol is selected, including EtherCAT. That connection is Modbus RTU and runs at 115200 baud by default; the rate can be changed from MacTalk.

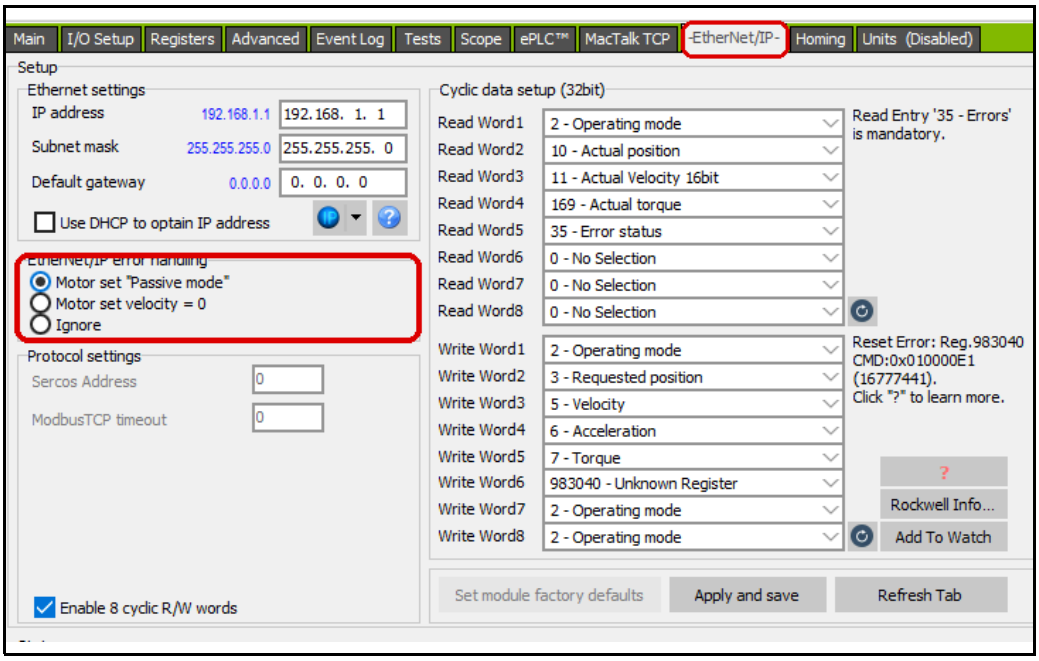
2.8 Motor reaction on lost connection

2.8.1 General description

When the cyclic connection to the PLC/master (the IO controller, scanner or EtherCAT master) is lost, the motor can be configured to react - typically by switching to passive mode or by setting the commanded velocity to zero. This prevents the motor from continuing an uncontrolled movement after the fieldbus connection has dropped.

2.8.2 Configuration via MacTalk

Select the reaction on the tab of the protocol in MacTalk, under the error handling setting (for example EtherNet/IP error handling on the EtherNet/IP tab): Motor set "Passive mode", Motor set velocity = 0, or Ignore. Then press Apply and save.



2.8.3 Configuration via motor registers

When the configuration is done by channels other than MacTalk, such as Ethernet, Profibus, CAN-open, etc., or just a simple interface connection to the basic motor, the following description must be followed.

The reaction is selected by the two lowest bits of the communication setup register (register 515, 16390 in MacTalk on MAC G3):

- Value 0 - the motor is set to passive mode.
- Value 1 - the motor is set to velocity mode with velocity 0 (a controlled stop).
- Value 2 - the reaction is ignored and the motor keeps its current settings.

The value is written into bits 0-1 of the communication setup register (515, 16390 in MacTalk on MAC G3). Value 3 behaves like value 2. The bit layout is given in the online register overview:
www.jvl.dk/register.html?register=MACmotor

[EDITORIAL NOTE - NOT FOR RELEASE] ADO #2056 CLF

MIS G3 firmware does not react on a lost connection yet (smc67 master b879a8ad39f: platform_set_ethernet_communication_lost() is never called in the motor build). The text above describes the intended behaviour; remove this note when #2056 is fixed.

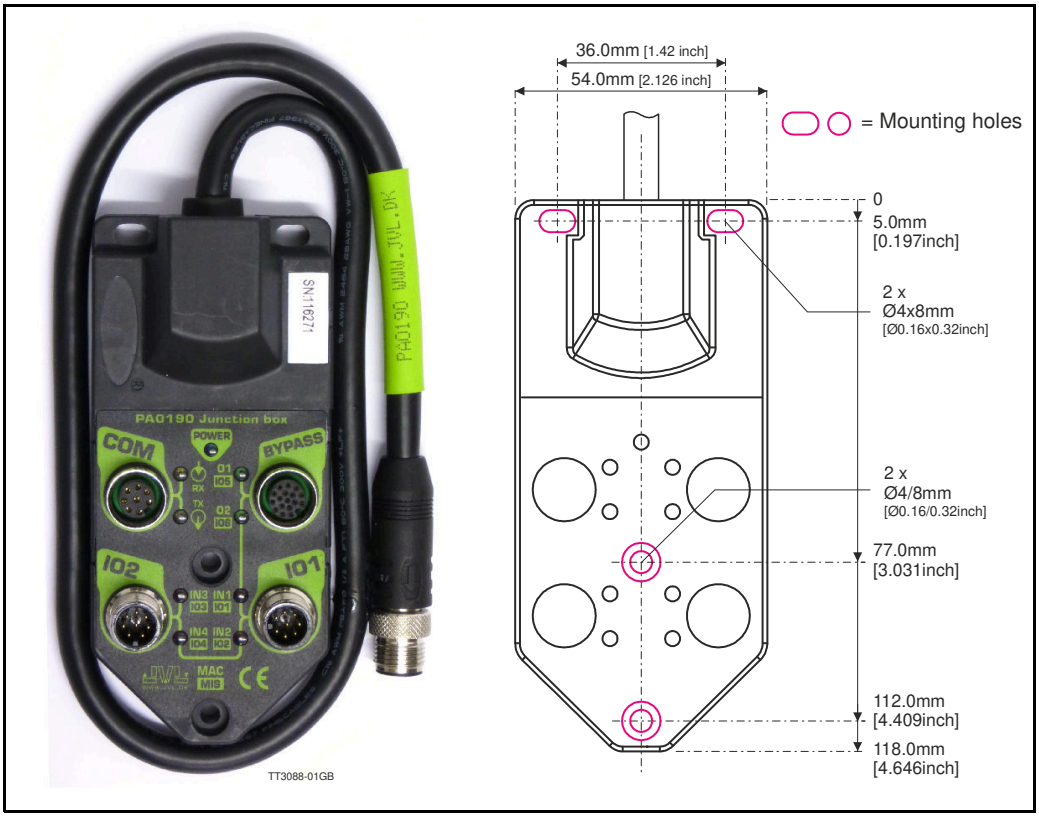
2.9 Web server

A built-in web server ("MacTalk Light") is available on HTTP port 80 and gives read/write access to the motor registers from a standard web browser.

2.10 I/O junction box PA0190

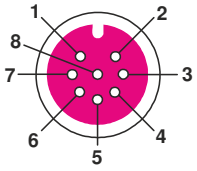
Only 17-pin I/O

The PA0190 junction box splits the signals in the motor's I/O connector into four individual connectors (COM, BYPASS, IO1 and IO2) for easier and more flexible installation.



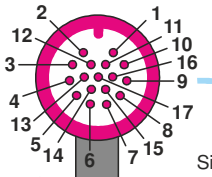
The pin assignment of the four connectors and the meaning of the indicator LEDs are shown in figure TT3089-01GB.

«COM»
8 pin Female



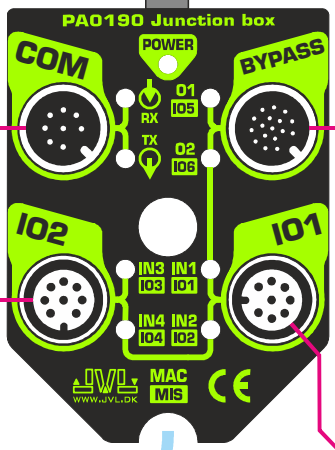
Pin no.	Func.	Color*
Pin 1	IN4	White
Pin 2	TX	Brown
Pin 3	RX	Green
Pin 4	GND	Yellow
Pin 5	RX	Grey
Pin 6	TX	Pink
Pin 7	AIN1	Blue
Pin 8	AIN2	Red
Body	GND	-

Cable from Module
with 17 pin Male connector

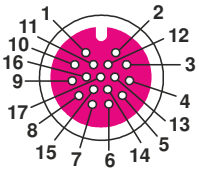


The screen is connected to «GND» (ground)

Signals and colors like the «BYPASS» cable

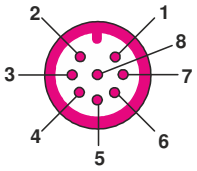


«BYPASS»
17 pin Female



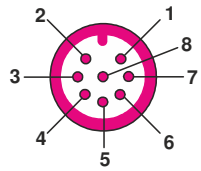
Pin no.	Funct.	Color*
Pin 1	IN1	Brown
Pin 2	GND	Blue
Pin 3	IN2	White
Pin 4	IN3	Green
Pin 5	B2-	Pink
Pin 6	IN4	Yellow
Pin 7	A2-	Black
Pin 8	B2+	Grey
Pin 9	O+	Red
Pin 10	A2+	Violet
Pin 11	O1	GY/PK
Pin 12	O2	RD/BU
Pin 13	AIN1	WH/GN
Pin 14	AIN2	BN/GN
Pin 15	RX	WH/YE
Pin 16	IO-	YE/BN
Pin 17	TX	WH/GY
Body	GND	-

«IO2»
8 pin Male



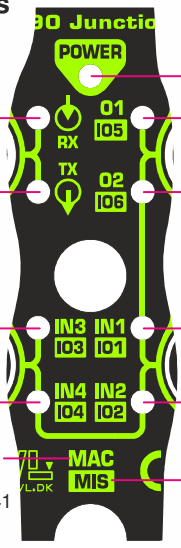
Pin no.	Funct.	Color*
Pin 1	IN4	White
Pin 2	AIN1	Brown
Pin 3	AIN2	Green
Pin 4	IO-	Yellow
Pin 5	O2	Grey
Pin 6	O1	Pink
Pin 7	GND	Blue
Pin 8	O+	Red
Body	GND	-

«IO1»
8 pin Male



Pin no.	Funct.	Color*
Pin 1	IN1	White
Pin 2	IN2	Brown
Pin 3	IN3	Green
Pin 4	IO-	Yellow
Pin 5	O2	Grey
Pin 6	O1	Pink
Pin 7	GND	Blue
Pin 8	O+	Red
Body	GND	-

LED explanations



POWER is lit if terminal *«OUT+»* is supplied

RX is lit if data is received at the RS232 line.

O1 is lit if output 1 is activated

TX is lit if data is transmitted at the RS232 line.

O2 is lit if output 2 is activated

IN3 is lit if input 3 is activated.

IN1 is lit if input 1 is activated

IN4 is lit if input 4 is activated.

IN2 is lit if input 2 is activated

Text in NON-inverse must be used when PA0190 is used with the MAC00-Ex41 Ethernet modules.

Text in inverse is NOT relevant when PA0190 is used with the MAC00-Ex41 Ethernet modules.

TT3089-01GB

*** Notes.**
Colors shown are based on the JVL standard cables type:

COM connection 8 Pin Male cable WI1000-M12M8TxxN

BYPASS connect. 17 Pin Male cable WI1009-M12M17TxxN

IO1 connection 8 Pin Female cable WI1000-M12F8TxxN

IO2 connection 8 Pin Female cable WI1000-M12F8TxxN

2.10 I/O junction box PA0190

Only 17-pin I/O

Use either the "BYPASS" connector or the "COM" connector - not both at the same time. If a cable is connected to "BYPASS", the communication pins and GND must be properly connected to valid signals on pins 2, 15 and 17.

The indicator LEDs on the PA0190 only work on motors where OUT+ and IO- are supplied from the Ethernet interface. See also the I/O description for the motor.

The PA0190 applies only to motor variants ordered with the 17-pin I/O connector - the connectivity options with 8 I/O points (type-number connectivity codes EC / EI / EP / EM, "2 channels + 8 I/O (17-pin)"). Variants with the 8-pin I/O connector (codes DC / DI / DP / DM, "3 inputs / 1 output (8-pin)") do not use the PA0190. Which variant a motor is can be read from its type number - see the type number nomenclature in the MAC G3 (LB0067) or MIS G3 (LB0076) user manual - or from the connector options in the product configurator at https://ppnb.jvl.dk.

2.11 Terms and abbreviations

These terms are used throughout this manual. Terms that belong to one protocol only are listed in that protocol's own chapter.

Table 4: Terms used throughout this manual

Term	Meaning
100BASE-TX	100 Mbit/s Ethernet on twisted pairs, which all the protocols in this manual use
DHCP	Dynamic Host Configuration Protocol. Available on EtherNet/IP only - see IP configuration
IP address	The logical address of the motor. A generation III motor has two - see IP configuration
MAC address	The hardware address, printed on the motor label
TCP	Transmission Control Protocol, the connection-oriented IP transport
UDP	User Datagram Protocol, the connectionless IP transport

2.11 Terms and abbreviations

3 EtherCAT (beta)

Only EtherCAT

3.1 Introduction

Only EtherCAT

EtherCAT® is a real-time Ethernet technology in which each slave processes the Ethernet frame "on the fly" as it passes through, giving very short cycle times. A generation III motor with EtherCAT is an EtherCAT slave using CANopen over EtherCAT (CoE).

The EtherCAT CiA402 drive profile is released as a beta version in this firmware.

3.1.1 Introduction to EtherCAT

EtherCAT is a real-time Ethernet technology that aims to make full use of the 100 Mbit/s full-duplex Ethernet bandwidth. It avoids the overhead normally associated with Ethernet by processing the frames on the fly, in hardware.

An EtherCAT network consists of one master and up to 65535 slave devices, connected with standard Ethernet cabling. Each slave processes the incoming frame directly - it extracts the data addressed to it and inserts its own - and passes the frame on to the next slave with a delay of approximately 4 us. The last slave in the segment returns the processed frame, so the master receives it back as a kind of response frame.

Several protocols can be used as the application layer. In CANopen over EtherCAT (CoE) the CANopen protocol is carried over EtherCAT: it defines Service Data Objects (SDO) for acyclic access, Process Data Objects (PDO) for cyclic data, and the object dictionary that holds the parameters. Both profiles that generation III motors offer - the JVL register profile and CiA402 - are reached over CoE.

On generation III motors the Industrial Ethernet interface is built into the motor and cannot be removed. The protocol is selected via the Interface option in the Product Part Number Builder (PPNB) when the motor is ordered. The two Ethernet ports are used to daisy chain the motors: in on LA-I (XG1) on MAC G3 or CN2 on MIS G3, out on LA-O (XG2) or CN3. All four ports are M12 4-pin D-coded.

EtherCAT does not use IP addressing itself. The IP addresses described in Common setup and functions are used by MacTalk and Modbus® TCP/UDP, which on EtherCAT are carried inside the Ethernet over EtherCAT (EoE) tunnel.

Addressing, status indicators, MacTalk over Ethernet and the reaction on a lost connection are the same on all protocols and are described in Common setup and functions.

Further information is available from the EtherCAT Technology Group, https://www.ethercat.org.

3.1.2 Abbreviations

These terms are used in this chapter and in the register descriptions.

Table 5: EtherCAT abbreviations

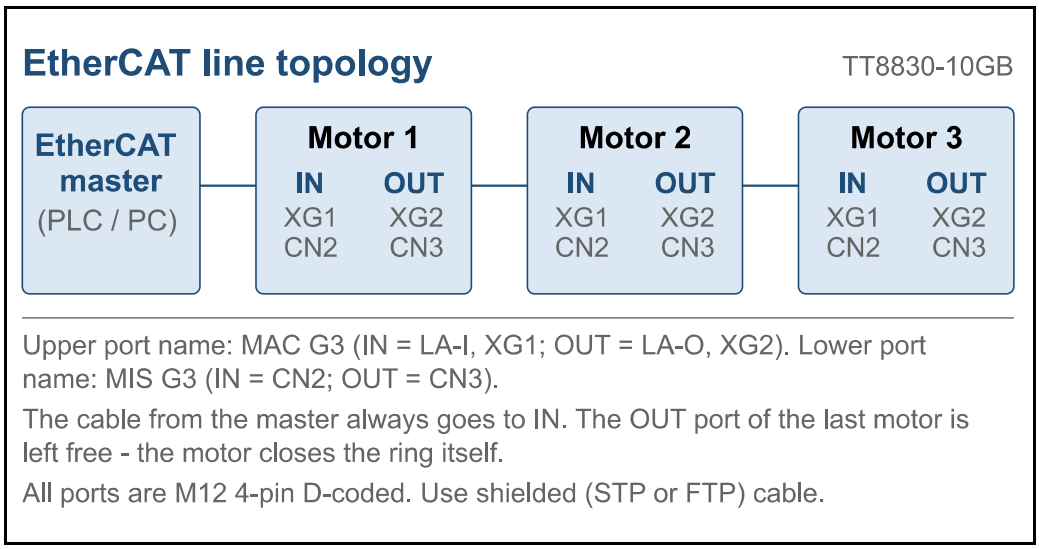
Term	Meaning
AL	Application Layer (EtherCAT state machine)
CAN	Controller Area Network
CANopen	Application layer protocol used in automation
CoE	CANopen over EtherCAT
CSP / CSV / CST	Cyclic Synchronous Position / Velocity / Torque (CiA402 modes)
DC	Distributed Clocks
EMCY	Emergency object
EoE	Ethernet over EtherCAT
ESC	EtherCAT Slave Controller
ESI	EtherCAT Slave Information (the XML device file)
ETG	EtherCAT Technology Group
PDO	Process Data Object, for cyclic data
PP / PV / HM	Profile Position / Profile Velocity / Homing (CiA402 modes)

Table 5: EtherCAT abbreviations (Continued)

Term	Meaning
SDO	Service Data Object, for acyclic data
SII	Slave Information Interface
SYNC0	The Distributed Clocks synchronisation event

3.1.3 Network topology

The motors are connected in a line, as shown in figure TT8830-10GB. The cable from the master goes to the IN port of the first motor, and each following motor is connected from the OUT port of the one before it. The OUT port of the last motor is left free: the EtherCAT slave controller closes the ring itself, so the frame returns to the master through the same cable.



Standard shielded Ethernet cable (STP or FTP, CAT5 or better) with M12 D-coded plugs is used. Unshielded cable is not recommended in an industrial environment.

3.2 Protocol specifications

Only EtherCAT

This section describes the parts of EtherCAT that the motor implements as an EtherCAT slave: how the master and the slaves communicate, how a frame is built, how the sync managers give access to the process data, and the state machine the master steps the motor through.

3.2.1 Communication and addressing

EtherCAT is standardised by the EtherCAT Technology Group (ETG). Communication is based on master/slave operation with exactly one master on the network. Because of the ring topology only one telegram is sent on the bus in each bus cycle, so the bus cycle time is the same in every cycle. How long that cycle is depends on the number of slaves, the amount of process data of the individual slaves, and the cycle time set in the master.

A slave can be addressed in two ways. With auto increment addressing the master scans the network for slaves and addresses them in the order in which they are physically present, so the position in the daisy chain decides the address. With fixed node addressing each node uses an address that has been programmed into it, and the address then follows the motor rather than its position in the chain.

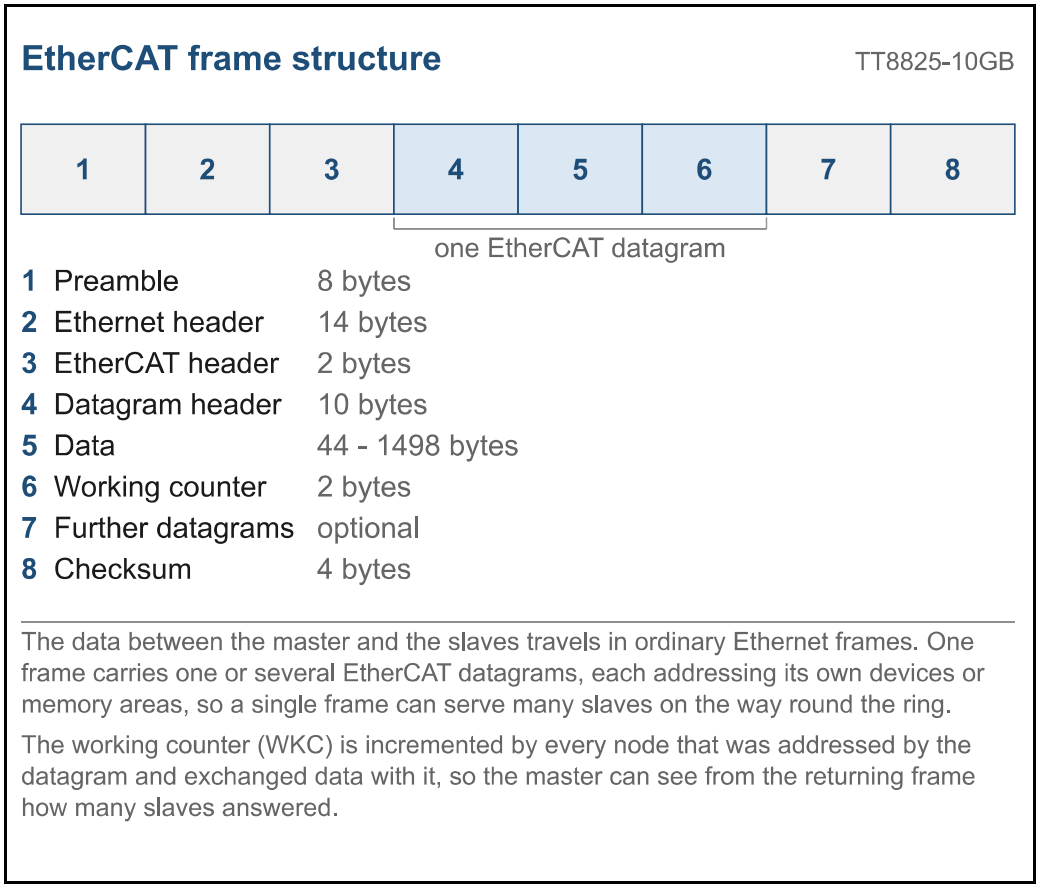
The station alias used for fixed node addressing is kept in the EEPROM of the EtherCAT slave controller (the SII) and is written there by the master's configuration tool; the motor firmware does not set it, and interface register 518 is not used for it. The interface registers are listed in the online register overview:
www.jvl.dk/register.html?register=MACmotor

3.2.2 EtherCAT frame structure

The data between the master and the slaves is carried in standard Ethernet frames. An EtherCAT frame holds one or several EtherCAT telegrams, each addressing individual devices or memory areas. The telegrams are carried either directly in the data area of the Ethernet frame, or inside the data section of a UDP datagram transported over IP.

Each telegram consists of an EtherCAT header, the data area and a working counter (WKC). The working counter is incremented by every node that is addressed by the telegram and has exchanged its data, so the master can see from the returned frame how many of the addressed nodes actually took part.

The frame structure is shown in figure TT8825-10GB. One Ethernet frame carries one or several EtherCAT datagrams, and each datagram addresses its own devices or memory areas, so a single frame can serve many motors on its way round the ring.



The working counter (WKC) at the end of each datagram is incremented by every node that was addressed and exchanged data. The master therefore sees from the returning frame how many slaves actually answered, which is what makes a missing or silent node visible without any extra diagnostics.

Table 6: EtherCAT frame structure

Field	Size
Preamble and start of frame	8 bytes
Ethernet header	14 bytes
EtherCAT header	2 bytes
Datagram header	10 bytes
Data area	variable
Working counter (WKC)	2 bytes
Frame check sequence	4 bytes

3.2.3 Sync managers

Sync managers control the access to the application memory. Each channel defines a consistent area of that memory, so the master and the motor never read and write the same area at the same time. A generation III motor has four sync manager channels.

Two of them carry the mailbox, which is where the acyclic SDO traffic runs, and they are 128 bytes in each direction. The other two carry the cyclic process data: the outputs the master writes to the motor, and the inputs the motor returns. The EtherCAT Slave Information (ESI) file declares room for up to 1024 bytes in each direction, but the motor uses much less: 32 bytes in each direction with the JVL register profile (eight 32-bit registers), and at most 32 bytes with the CiA402 profile.

3.2.4 Sync manager watchdog

The sync manager watchdog monitors the output sync manager. If the master does not update the output data within the time it has configured, the watchdog times out and the motor leaves the Operational state for Safe-Operational. If the cause is still present when the master brings the slave back to Operational, the slave will move between Operational and Safe-Operational repeatedly, and that pattern is the symptom to look for.

EtherCAT has been designed so that a slave has no way of monitoring the connection to the master as long as it receives no output data. The watchdog is therefore what detects a master that has stopped sending, and it acts on the communication rather than on the motion.

What the motor itself does when the connection is lost - stop, stay passive, or hold the last commanded value - is configured in the motor and is the same on all protocols. It is described in Common setup and functions.

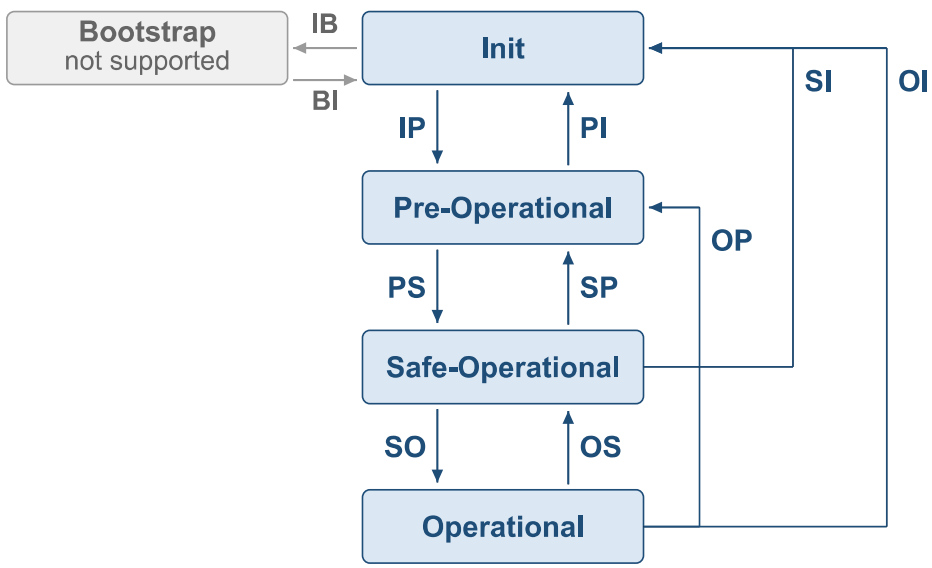
3.2.5 EtherCAT state machine

Both the master and the slave have a state machine. After power-up the motor is in the Init state, and from there it is the master that requests every transition; the motor never changes state on its own except when an error forces it down.

The Bootstrap state is not supported. The bootstrap mailbox is disabled in the firmware, and firmware update over EtherCAT (FoE) is therefore not available. Firmware can still be updated over the RS485 connection from MacTalk.

The PDO mapping is configured in the Pre-Operational state. The mapping entries are write-restricted to Pre-Operational, so a master that tries to change the mapping in Safe-Operational or Operational will be refused. If the mapped process data does not match the size the sync manager has been configured with, the transition from Pre-Operational to Safe-Operational fails, and the master reports an AL status code rather than reaching Safe-Operational.

The state machine is shown in figure TT8826-10GB, with the two-letter code of each transition. The motor enters Init at power-up, and from there it is the master that requests every transition.



The motor enters Init at power-up. From there it is the master that requests every transition; the motor never changes state on its own except to fall back when something is wrong.

Bootstrap is not supported on generation III: the bootstrap mailbox is disabled in the firmware, so firmware update over EtherCAT (FoE) is not available. Use MacTalk over the RS485 connection instead.

Mailbox communication (SDO) becomes available in Pre-Operational, which is also the only state in which the PDO mapping can be changed. In Safe-Operational the motor sends its input process data but does not act on the output data; only in Operational is the output data used. A motor that will not leave Safe-Operational has either a process data size that does not match the sync manager or, with Distributed Clocks enabled, a SYNC0 cycle time that does not match the motor.

Table 7: EtherCAT state machine

State	Description
Init	State after device initialisation. No application layer communication: neither SDO nor PDO.
Pre-Operational	SDO communication is possible. No PDO communication. The PDO mapping is configured here.
Safe-Operational	The motor sends its input PDO to the master. There is no output data from the master to the motor.
Operational	Fully operational. The motor also acts on the output PDO from the master.
Bootstrap	Not supported.

3.2.6 CANopen over EtherCAT

The application layer is CANopen over EtherCAT (CoE), which is based on the CANopen DS301 communication profile. It defines the object dictionary of the motor and

three kinds of messages:

- Process Data Objects (PDO) carry the cyclic process data.
- Service Data Objects (SDO) carry acyclic access to the object dictionary - identification, configuration and parameters.
- Emergency messages (EMCY) report errors in the motor, see Diagnosis.

Both profiles, the JVL register profile and CiA402, are reached over CoE.

3.3 Status indicators

Only EtherCAT

The red ERR indicator and the green RUN / L1 indicator report the EtherCAT state of the motor, and the two port indicators LA1/L2 and LA2/L3 report link and activity. The patterns are shown in figure TT8801-10GB.

Communication LEDs - EtherCAT states

TT8801-10GB

On the other protocols the same LEDs are used differently - see the protocol chapter.

Pattern		Meaning
ERR		(red)
off	(off)	No EtherCAT error
constant		Critical communication error
blinking		General configuration error
single flash		Local error
double flash		Process-data / watchdog timeout
flickering		Booting error
RUN / L1		(green)
off	(off)	Device state = INIT
constant		Device state = Operational
blinking		Device state = Pre-Operational
single flash		Device state = Safe-Operational
flickering		Booting - not yet in INIT
LA1, LA2 / L2, L3		(green/yellow)
off	(off)	No valid Ethernet connection
constant		Ethernet connected
flickering		Link / activity

blinking = 200 ms on/off (2.5 Hz) | single flash = 200 ms on + 1 s off
double flash = 2 x 200 ms + 1 s off | flickering = approx. 50 ms (10 Hz)

ERR is off as long as EtherCAT communication is healthy. A constant red indicator is a critical communication error, blinking is a general configuration error, a single flash is a local error, a double flash is a process-data or watchdog timeout, and flickering means the interface is still booting.

Errors and warnings are reported to the master through the CANopen mechanisms. An error produces an Emergency (EMCY) frame, and every entry is also written to the diagnosis history in object 0x10F3, which the master can read back at any time. A master that missed the Emergency can therefore still find out what happened.
Object 0x1001 is the CiA-301 error register and holds the active error flags. On the CiA402 drive profile, object 0x603F holds the error code of the most recent error as well. The texts that belong to the codes are in the EtherCAT Slave Information file, so the master shows the condition in words and not as a number.

3.4.1 Emergency messages

An Emergency message is sent once when an error occurs, and once more with error code 0x0000 when the error is no longer present. Emergency messages are sent in Pre-Operational, Safe-Operational and Operational. Only errors produce an Emergency message.

Table 8: Emergency message layout

Bytes	Content
0-1	CANopen error code (see the table below)
2	Error register, the value of object 0x1001
3	0
4-7	JVL diagnostic code, 32 bits, least significant byte first

The error codes the motor can send are listed below. The JVL diagnostic code in bytes 4-7 tells exactly which condition it is, and the ESI file holds the text for it.

Table 9: Emergency error codes

Code	Meaning	Product
0x0000	Error cleared (sent when an error is no longer present)	Both
0x1000	Too many errors at the same time	Both
0x2221	Peak current	MAC G3
0x2222	Power stage (PWM) locked	MAC G3
0x3120	Mains undervoltage	MAC G3
0x3210	Overvoltage	MAC G3
0x4210	Temperature too high	MAC G3
0x5000	Current loop error	MAC G3
0x5112	Control voltage (24 V) error	Both
0x5380	Initialisation error	MAC G3
0x5381	Safe torque off (STO)	MAC G3
0x5382	FPGA error	Both
0x5530	Non-volatile memory or encoder data corrupt	Both
0x5580	Flash write error	MAC G3
0x6010	Software, firmware image or protocol stack error	Both
0x6320	Parameter, production data or PDO configuration error	Both
0x7110	Regenerative overload or brake error	MAC G3
0x7305	Encoder error	Both
0x7500	Encoder communication error	MIS G3
0x8110	Internal communication error in the protocol stack	Both
0x8180	Communication with the motor controller lost	MAC G3
0x8181	Slave error	MAC G3
0x8311	Overload (I2T)	MAC G3
0x8331	Function error	MAC G3

Table 9: Emergency error codes (Continued)

Code	Meaning	Product
0x8480	Overspeed	MAC G3
0x8611	Follow error	MAC G3
0x8680	Software position limit exceeded	MAC G3
0xFF00	Homing timeout	MIS G3

3.4.2 AL status codes

When the motor refuses a state transition it reports the reason as an AL status code, which the master shows. Besides the codes the EtherCAT protocol stack sets by itself, the motor sets these:

Table 10: AL status codes set by the motor

Code	Cause	Product
0x0024	Operational refused: the read list (TxPDO) is invalid	MAC G3
0x0025	Operational refused: the write list (RxPDO) is invalid	MAC G3
0x0035	The SYNC0 cycle time does not match the motor cycle time (register 231)	MAC G3

3.5 Profiles

Only EtherCAT

Two profiles are available. The profile is chosen when the EtherCAT interface starts, from bit 3 (Enable drive profile) of register 515 (16390 in MacTalk) on MAC G3 or register 123 on MIS G3. In MacTalk this is the "Enable DSP402 drive profile" option. A change takes effect only when the setting has been saved and the motor restarted.

- JVL register profile (DS301) - eight motor or interface registers in each direction. See JVL register profile.
- CiA402 (DSP402) drive profile - the standardised drive profile, for MAC G3 only in this release. It is the default after a factory reset.

On MIS G3 the CiA402 profile is not supported in this release. A factory reset nevertheless selects it, so an MIS G3 must be switched to the JVL register profile after a factory reset: clear bit 3 in register 123 (or the MacTalk option), save to flash and restart the motor.

On MAC G3, selecting the CiA402 profile also sets the motor cycle time to 1 ms (bit 13 of register 236). The motor saves this and restarts by itself if it was set otherwise.

The profile also decides the device the master sees - the two profiles have different product codes, see ESI file.

3.6 CoE object dictionary

Only EtherCAT

Motor and interface data are available as CoE objects: object 0x2011 for interface parameters, object 0x2012 for motor registers 1-255, and object 0x2013 for the extended motor registers (256 and up). The cyclic process data (PDO) uses a single PDO in each direction whose mapping can be configured in the Pre-Operational state; the mapping entries appear as "Cyclic Write 1..8" and "Cyclic Read 1..8".
The sub-index is the register number, with one shift per object: 0x2012 sub-index N is motor register N, 0x2013 sub-index N is motor register N + 255, so sub-index 1 of 0x2013 is register 256. In object 0x2011 the sub-index counts from the first Ethernet interface register instead: sub-index 3 is register 512 in the shared register list. These three objects are the MAC G3 arrangement.
The MIS G3 uses three different objects, and objects 0x2011, 0x2012 and 0x2013 do not exist on it at all. Object 0x2001, Motor Parameters, holds the motor registers with the sub-index equal to the register number. Object 0x2002, Scope and User Parameters, continues from register 255: sub-index N is register N + 254. Object 0x2003, Ethernet Parameters, holds the Ethernet interface registers: sub-index N is register N + 508, so sub-index 4 is register 512. The three objects 0x2001 to 0x2003 in turn do not exist on a MAC G3. See Register numbering in Common setup and functions.
In this release a write to object 0x2003 on MIS G3 does not reach the right register. Set the MIS G3 interface registers from MacTalk until this is corrected; reading through 0x2003 is not affected.

3.6.1 Communication objects

The communication objects follow CiA301 and the EtherCAT specification. They are the same on both products unless the table says otherwise.

Table 11: Communication objects

Index	Name	Access	Value
0x1000	Device type	ro	0x012C1389 (JVL register profile), 0x00020192 (CiA402)
0x1001	Error register	ro	Bit 0 generic, 1 current, 2 voltage, 3 temperature, 4 communication, 7 manufacturer-specific
0x1008	Device name	ro	MACxxx G3 or MISxxx 67
0x1009	Hardware version	ro	1
0x100A	Software version	ro	Version of the EtherCAT protocol stack - not the motor firmware
0x1018	Identity	ro	Vendor ID 0x117 (JVL), product code (see ESI file), revision 1, serial number
0x10F3	Diagnosis history	ro	Up to 50 messages. Sub-index 3 and 5 are writable
0x1600	RxPDO mapping	rw	Cyclic Write 1..8. Writable in Pre-Operational only
0x1A00	TxPDO mapping	rw	Cyclic Read 1..8. Writable in Pre-Operational only
0x1C00	Sync manager type	ro	1, 2, 3, 4: mailbox out, mailbox in, outputs, inputs
0x1C12	SM2 assignment	ro	Fixed: 0x1600
0x1C13	SM3 assignment	ro	Fixed: 0x1A00
0x1C32	SM2 synchronisation	ro	See Distributed Clocks
0x1C33	SM3 synchronisation	ro	See Distributed Clocks

3.6.2 Commands

Commands such as save to flash are written to a command object. On MAC G3 it is object 0x2010, which writes the interface command register 524; the command status can be read in register 523. On MIS G3 it is object 0x2000 sub-index 1, which writes the motor's command register 24.

Table 12: Commands over EtherCAT

Command	MAC G3: object 0x2010	MIS G3: object 0x2000:1
Save parameters to flash	16 interface, 272 motor, 273 both and restart	269 (268: save and restart)
Restart	1 interface, 257 interface and motor	257
Clear errors	225 to 0x2012 sub-index 211	265
Factory default	18	275 (277 keeps the IP settings)
Re-initialise cyclic data	21	300

Parameters changed over EtherCAT are not saved automatically. Save them with the save command, otherwise they are lost at the next power cycle.

3.7 JVL register profile

Only EtherCAT

With the JVL register profile the process data is eight 32-bit registers in each direction: Cyclic Write 1..8 from the master to the motor (RxPDO 0x1600) and Cyclic Read 1..8 from the motor to the master (TxPDO 0x1A00). Which register goes in each slot is selected in the motor, so any motor or interface register can be exchanged cyclically.

3.7.1 Selecting the cyclic registers

The registers are selected in the motor, in MacTalk or by writing the interface registers directly: Cyclic Read 1..8 are registers 525-532, and Cyclic Write 1..8 are registers 533-540 (16400-16407 and 16408-16415 in MacTalk on MAC G3). The selection is kept in RAM until it is saved to flash, so a power cycle without a save brings back the saved selection.

MAC G3: a new selection is taken into use with command 21 (re-initialise cyclic data) or when it is saved with command 16, both through object 0x2010. The master may also change the mapping itself through objects 0x1600 and 0x1A00 in the Pre-Operational state; the motor takes it into use at the transition to Safe-Operational.

MIS G3: save the selection and restart the motor. The mapping cannot be changed from the master, and objects 0x1600 and 0x1A00 always show the mapping the motor started with.

3.7.2 Default cyclic registers

Out of the box the cyclic registers are the ones below. Register numbers are the product's own; see Register numbering in Common setup and functions.

Table 13: Default cyclic registers, JVL register profile

Slot	MAC G3 read	MAC G3 write	MIS G3 read	MIS G3 write
1	2 mode	2 mode	9 actual mode	2 mode
2	10 position	3 P_SOLL	10 position	4 position
3	11 velocity	5 V_SOLL	12 velocity	3 velocity
4	169 torque	6 A_SOLL	13 torque	6 acceleration
5	35 ERR_STAT	7 T_SOLL	153 diagnosis	7 torque
6	-	983040 command	-	24 command
7	-	-	-	-
8	-	-	-	-

The MAC G3 write list contains the torque limit T_SOLL (register 7), and the MIS G3 write list the torque (register 7). Write a sensible value there as well - with 0 the motor has no torque. 341 is 100 % on MAC G3, and 1000 is 100 % on MIS G3.

3.7.3 Rules for the cyclic registers

- MAC G3: registers 35, 36 and 211 cannot be written cyclically, and register 35 is always in the read list - the motor adds it if it is missing. Send motor commands and clear errors through the interface command register instead, see Examples.
- MAC G3: an interface register (512 and up) is entered as (register - 509) x 65536 and must come after all motor registers in the list. The interface command register 524 is therefore entered as 983040.
- MAC G3: if the selection is invalid, the list is emptied and the master is refused Operational with AL status code 0x0024 (read list) or 0x0025 (write list).
- MIS G3: read-only registers cannot be written cyclically.
- Both: a register can appear only once in each list. A command written cyclically is executed when the value changes to a new value other than 0; to repeat

the same command, write 0 for at least one cycle first.

- MIS G3: registers 4, 10, 20 and 330 (P7) are 64-bit. To read one cyclically, put it in the read list followed by the next register number, for example 20 and 21. The two words then carry the upper 32 bits first and the lower 32 bits second. If the slot after a 64-bit register holds another register, that register is sent in place of the upper 32 bits, and only the lower 32 bits of the 64-bit register are read.

[EDITORIAL NOTE - NOT FOR RELEASE] ADO #2015 DJO

3.7.4 Cycle time

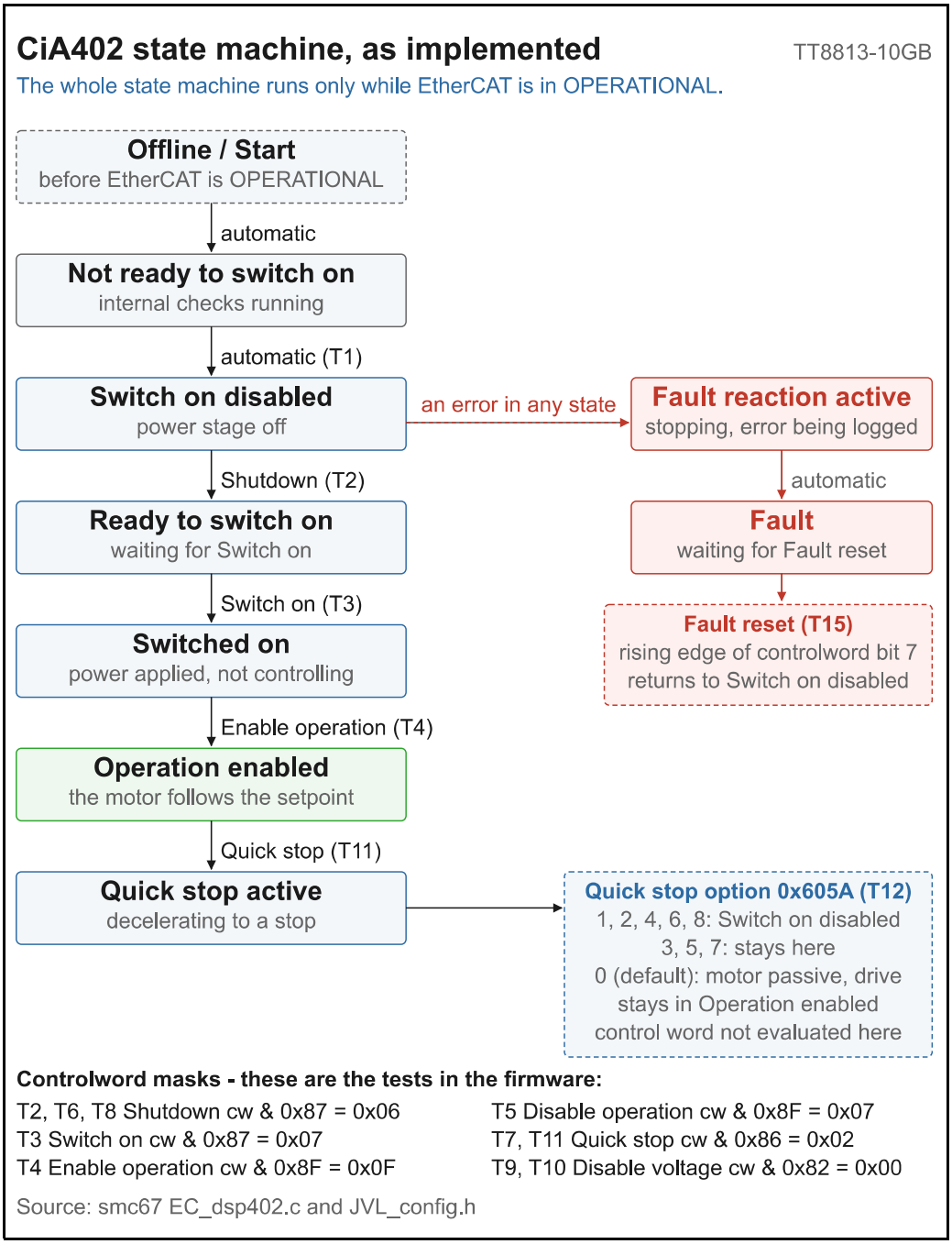
The JVL register profile runs free-running or synchronised to SYNC0, see Distributed Clocks. The minimum cycle time the motor reports in object 0x1C32 is 500 us.

3.8 CiA402 drive profile (beta) Only EtherCAT

On MAC G3 the CiA402 drive profile supports the operating modes Profile Position (PP), Profile Velocity (PV), Homing (HM), Cyclic Synchronous Position (CSP), Cyclic Synchronous Velocity (CSV) and Cyclic Synchronous Torque (CST). The control word (0x6040) and status word (0x6041) follow the CiA402 state machine, and the homing methods (0x6098) are mapped to the motor's own homing functions.
The CiA402 profile is not supported on MIS G3 in this release - use the JVL register profile. The rest of this section applies to MAC G3.

3.8.1 CiA402 state machine

The state machine that the motor implements is shown in figure TT8813-10GB. It is the CiA402 state machine with two additions that belong to this implementation: the Offline and Start states the motor passes through before EtherCAT reaches OPERATIONAL, and the fact that the state machine is only evaluated while the EtherCAT slave is in the OPERATIONAL state.



Commands are written to the control word (0x6040) and the resulting state is read back from the status word (0x6041). The motor tests the control word with the masks shown in the figure, so the bits that are not part of a mask do not affect that transition.

If the control word is written before the master has brought the slave to OPERATIONAL, the motor does not act on it and the state bits in the status word stay as they are. This is the most common reason for a motor that appears not to respond to Enable operation: the drive profile is not running yet, and the motor is still in the Offline or Start state.

3.8.2 Quick stop

Quick stop is entered from Operation enabled with the Quick stop command (transition 11). What happens then is decided by the quick stop option code (object 0x605A). With option codes 1, 2, 4, 6 and 8 the motor stops and the drive returns to Switch on disabled by itself (transition 12). With option codes 3, 5 and 7 it stops and stays in Quick stop active. Option code 0 is different: the motor is set passive at once, and the drive stays in Operation enabled - it does not enter Quick stop active at all.

Option codes 2, 4, 6 and 8 load the quick stop deceleration (object 0x6085) before the motor is stopped.

The control word is not evaluated while the drive is in Quick stop active, so in this release the state cannot be left by a command from the master; it is left only when an error occurs or the drive profile is initialised again. The default quick stop option code is 0. Set it to 2 if the motor should stop and return to Switch on disabled after a quick stop.

3.8.3 Control word

The state machine uses bits 0-3 and 7 with the masks shown in figure TT8813-10GB. The other bits are used as follows.

Table 14: Control word 0x6040 (MAC G3)

Bit	Name	Use in this firmware
0-3, 7	State machine	Switch on, Enable voltage, Quick stop, Enable operation, Fault reset - see figure TT8813-10GB
4	Mode specific	PP: new set-point (rising edge). HM: start homing (rising edge)
5	Change set immediately	PP: 1 = a new set-point interrupts the move, 0 = it waits until the target is reached
6	Absolute / relative	Not evaluated: target positions are always absolute
8	Halt	PP: stop and drop a waiting set-point. HM: abort homing. Not evaluated in PV, CSP, CSV, CST
9-15	-	Not evaluated

PP holds at most one waiting set-point: a new set-point while one is already waiting is not acknowledged.

3.8.4 Status word

Table 15: Status word 0x6041 (MAC G3)

Bit	Name	Meaning in this firmware
0-3, 5, 6	State	The CiA402 state, coded as in the standard
4	Voltage enabled	Mains voltage is present
7	Warning	Not set in this release
9	Remote	Always 1
10	Target reached	PP: position reached, also set on halt. Not set in the other modes
11	Internal limit active	Not set in this release
12	Mode specific	PP: set-point acknowledge. HM: homing attained. PV, CSP, CSV, CST: always 1
13	Mode specific	CSP: following error. HM: homing error. Otherwise 0
8, 14, 15	-	Not used

The following error bit in CSP is only active when a following error window is set in 0x6065; the default value 0xFFFFFFFF switches the check off. The bit is set while the following error (0x60F4) is larger than 0x6065 for longer than 0x6066 ms, and is not latched.

3.8 CiA402 drive profile (beta) Only EtherCAT

3.8.5 Operation modes

Table 16: Modes of operation (MAC G3)

0x6060	Mode	What the motor does
1	PP	Profile position: moves to 0x607A with 0x6081 as velocity and 0x6083 as acceleration
3	PV	Profile velocity: runs at 0x60FF with 0x6083 as acceleration
6	HM	Homing, see Homing
8	CSP	Cyclic synchronous position: follows 0x607A every cycle, limited by 0x6080, 0x60C5 and 0x6072
9	CSV	Cyclic synchronous velocity: follows 0x60FF every cycle
10	CST	Cyclic synchronous torque: follows 0x6071 every cycle

0x6502 Supported drive modes reports these six modes. The mode can be changed at any time, also in Operation enabled. A change aborts a homing in progress and drops a waiting PP set-point. A value that is not supported shows as 0 in 0x6061.

Masters usually set the mode with a start-up command when they configure the motor. In TwinCAT this is an entry on the Startup tab, see figure TT8833-10GB.

[SCREENSHOT NEEDED - CLF] TwinCAT Startup tab with a CoE start-up entry that writes 0x6060 = 8 (CSP).

Screenshot pending TT8833-10GB

ADO #1730 · Christian Lotz Felter (clf@jvl.dk)

LB0077 · section 3.8.5

The screenshot must show:

TwinCAT: Startup tab with a new CoE startup entry that writes 0x6060 Modes of operation = 8 (cyclic synchronous position) at the Pre-Operational to Safe-Operational transition

Placeholder - not for release. Replace this file with the screenshot under the same TT number; the manual then needs no change.

While the CiA402 profile is active, the motor mode and motor registers 5, 6 and 7 are written from the CiA402 objects every cycle - except in Homing. Values written to them from MacTalk or through object 0x2012 are overwritten.

3.8.6 Homing

Homing is started with a rising edge on control word bit 4 in mode 6 (HM). Status word bit 12 is set when homing is attained and bit 13 if it fails; a method the motor does not support sets bit 13 at once. Halt (control word bit 8) aborts the search. Object 0x60E3 lists the supported methods.

Table 17: Homing methods (MAC G3)

0x6098	Search for	Direction	Index pulse
-4	Mechanical stop	positive	no
-3	Mechanical stop	negative	no
-2	Mechanical stop	positive	yes
-1	Mechanical stop	negative	yes
3	Home sensor	positive	yes
4	Home sensor	positive	yes, zero 1 rev. lower
5	Home sensor	negative	yes, zero 1 rev. lower
6	Home sensor	negative	yes
19, 20	Home sensor	positive	no
21, 22	Home sensor	negative	no
33, 34	Index pulse only	same for both	-
35	None: position := 0	-	-
37	None: position := home offset	-	-

The search uses 0x6099 sub-index 1 as speed (sub-index 2 is not used), 0x609A as acceleration and 0x607C as home offset. Torque homing (methods -4 to -1) uses object 0x2100 Homing torque, in the same unit as 0x6072. Keep it low so the machine is not damaged, and use a home offset that moves the motor away from the mechanical stop.

The home sensor is the motor's analogue input used as a digital input, active above 4.5 V. There are no limit switch inputs for CiA402 homing.

3.8.7 Factors

Positions are in user units. The position factor converts them to encoder counts and is calculated from the gear ratio (0x6091) and the feed constant (0x6092) against the encoder resolution (0x608F):

position factor = (0x6091:1 x 0x6092:2 x 0x608F:1) / (0x6092:1 x 0x6091:2 x 0x608F:2)

The motor position in counts is the user position multiplied by the position factor. With the defaults the feed constant equals the encoder resolution and the gear ratio is 1:1, so one user unit is one encoder count. 0x608F reports the encoder resolution of the motor (4096, 8000 or 8192 counts per revolution, depending on the motor) and is read-only.

Example: a MAC G3 with 8192 counts per revolution drives a belt through a 3.5:1 gear. The drive wheel is 124 mm in diameter, so one revolution of the wheel is 389.56 mm, and the position is wanted in millimetres. Set 0x6091:1 = 35 and 0x6091:2 = 10 (3.5:1), and 0x6092:1 = 38956 and 0x6092:2 = 100 (389.56 mm per revolution). The position factor is (35 x 100 x 8192) / (38956 x 10 x 1) = 73.6 counts per millimetre.

Velocities are always in rpm, accelerations in rpm/s multiplied by the acceleration factor 0x6097, and torques in per mille of the maximum torque (on MAC G3, 1000 is 300 % of the rated torque). Polarity 0x607E bit 7 inverts the position and bit 6 the velocity.

3.8.8 PDO mapping

Table 18: Default PDO mapping, CiA402 profile (MAC G3)

PDO	Index	Object
RxPDO 0x1600	0x6040	Control word
RxPDO 0x1600	0x6060	Modes of operation
RxPDO 0x1600	0x607A	Target position
RxPDO 0x1600	0x60FF	Target velocity
RxPDO 0x1600	0x6071	Target torque
TxPDO 0x1A00	0x6041	Status word
TxPDO 0x1A00	0x6061	Modes of operation display
TxPDO 0x1A00	0x6064	Position actual value
TxPDO 0x1A00	0x60F4	Following error actual value
TxPDO 0x1A00	0x606C	Velocity actual value

The master can change the mapping in the Pre-Operational state, with up to eight objects and 32 bytes in each direction. Only objects marked RxPDO or TxPDO in the object tables can be mapped. The mapping is not stored in the motor - the master sends it every time it configures the motor. The sync manager assignment (0x1C12, 0x1C13) is fixed. In TwinCAT the mapping is edited on the Process Data tab, see figure TT8832-10GB.

[SCREENSHOT NEEDED - CLF] TwinCAT Process Data tab of a MAC G3 in the CiA402 profile, inserting 0x60FF into the RxPDO.

Screenshot pending TT8832-10GB

ADO #1730 · Christian Lotz Felter (clf@jvl.dk)

LB0077 · section 3.8.8

The screenshot must show:

TwinCAT: Process Data tab of a MAC G3 in the CiA402 profile - the default RxPDO/TxPDO content, and inserting 0x60FF Target velocity into the RxPDO for cyclic synchronous velocity

Placeholder - not for release. Replace this file with the screenshot under the same TT number; the manual then needs no change.

3.8.9 Object dictionary

The tables in this section list the CiA402 objects the motor implements. They are taken from the object dictionary in the firmware itself, which is what answers the master when it reads an object.

Access is ro for read-only and rw for read/write. The PDO column says whether the object can be mapped into the cyclic process data and in which direction: RxPDO carries data from the master to the motor, TxPDO from the motor to the master. An object with a dash can be reached with acyclic (SDO) access only.

Being listed here means the master can see and address the object. A few of them are present in the object dictionary without yet acting on the hardware - see Limitations in this release at the end of this section.

3.8 CiA402 drive profile (beta)

Only EtherCAT

Table 19: CiA402 objects: device control and status

Index	Object name	Type	Access	PDO
0x6040	Control word	UINT16	rw	RxPDO
0x6041	Status word	UINT16	ro	TxPDO
0x6060	Modes of operation	INT8	rw	RxPDO
0x6061	Modes of operation display	INT8	ro	TxPDO
0x605A	Quick stop option code	INT16	rw	-
0x6007	Abort connection option code	INT16	rw	-
0x603F	Error code	UINT16	ro	TxPDO
0x6502	Supported drive modes	UINT32	ro	-
0x60D9	Supported synchronisation functions	UINT32	ro	-

Table 20: CiA402 objects: set-points and actual values

Index	Object name	Type	Access	PDO
0x607A	Target position	INT32	rw	RxPDO
0x6064	Position actual value	INT32	ro	TxPDO
0x6065	Following error window	UINT32	rw	-
0x6066	Following error time out	UINT16	rw	-
0x6067	Position window	UINT32	rw	-
0x6068	Position window time	UINT16	rw	-
0x607B	Position range limit	Record	rw	-
0x607C	Home offset	INT32	rw	-
0x607D	Software position limit	Record	rw	-
0x60F4	Following error actual value	INT32	ro	TxPDO
0x60FF	Target velocity	INT32	rw	RxPDO
0x60B1	Velocity offset	INT32	ro	-
0x606B	Velocity demand value	INT32	ro	TxPDO
0x606C	Velocity actual value	INT32	ro	TxPDO
0x606D	Velocity window	UINT16	rw	-
0x606E	Velocity window time	UINT16	rw	-
0x6080	Max motor speed	UINT32	rw	-
0x6081	Profile velocity	UINT32	rw	RxPDO
0x6071	Target torque	INT16	rw	RxPDO
0x60B2	Torque offset	INT16	ro	-
0x6072	Max torque	UINT16	rw	RxPDO
0x6077	Torque actual value	INT16	ro	TxPDO
0x6087	Torque slope	UINT32	rw	-
0x60E0	Positive torque limit value	UINT16	rw	-
0x60E1	Negative torque limit value	UINT16	rw	-

Table 21: CiA402 objects: motion profile, homing and scaling

Index	Object name	Type	Access	PDO
0x6083	Profile acceleration	UINT32	rw	RxPDO
0x6085	Quick stop deceleration	UINT32	rw	-
0x6086	Motion profile type	INT16	rw	-
0x60C5	Max acceleration	UINT32	rw	-
0x607E	Polarity	UINT8	rw	-
0x6098	Homing method	INT8	rw	-
0x6099	Homing speeds	Record	rw	-
0x609A	Homing acceleration	UINT32	rw	-
0x60E3	Supported homing methods	Array	ro	-
0x60C2	Interpolation time period	Record	ro	-
0x608F	Position encoder resolution	Record	ro	-
0x6091	Gear ratio	Record	rw	-
0x6092	Feed constant	Record	rw	-
0x6097	Acceleration factor	Record	rw	-

Table 22: CiA402 objects: digital I/O, identification and JVL objects

Index	Object name	Type	Access	PDO
0x60FD	Digital inputs	UINT32	ro	TxPDO
0x60FE	Digital outputs	Record	rw	RxPDO
0x6402	Motor type	UINT16	rw	-
0x6403	Motor catalogue number	String	ro	-
0x6404	Motor manufacturer	String	rw	-
0x6405	http motor catalogue address	String	rw	-
0x6503	Drive catalogue number	String	rw	-
0x6505	http drive catalogue address	String	rw	-
0x2100	Homing torque	UINT16	rw	-
0x2101	Analogue input 1	INT16	ro	TxPDO
0x2103	Motor temperature	INT16	ro	TxPDO
0x2104	Motor load factor	UINT32	rw	-

The identification strings are not all read-only: 0x6403 Motor catalogue number is read-only, while 0x6404, 0x6405, 0x6503 and 0x6505 are read/write, so that a system integrator can record what is actually installed. The objects 0x2100 to 0x2104 are JVL objects: 0x2100 is the torque used by torque homing, 0x2101 the analogue input, 0x2103 the motor temperature in degrees Celsius, and 0x2104 the motor load factor.

3.8.10 Record objects

Nine of the objects are records with two sub-indices each. Sub-index 0 is the number of entries, as it is for every CiA402 record, and is not repeated in the table.

Table 23: CiA402 record objects and their sub-indices

Object	Sub-index	Name	Type	Access
0x607B	01	Min position range limit	INT32	rw
0x607B	02	Max position range limit	INT32	rw
0x607D	01	Min position limit	INT32	rw
0x607D	02	Max position limit	INT32	rw
0x608F	01	Encoder increments	UINT32	ro
0x608F	02	Motor revolutions	UINT32	ro
0x6091	01	Motor revolutions	UINT32	rw
0x6091	02	Shaft revolutions	UINT32	rw
0x6092	01	Feed	UINT32	rw
0x6092	02	Shaft revolutions	UINT32	rw
0x6097	01	Numerator	UINT32	rw
0x6097	02	Divisor	UINT32	rw
0x6099	01	Speed during search for switch	UINT32	rw
0x6099	02	Speed during search for zero	UINT32	rw
0x60C2	01	Interpolation time period value	UINT8	ro
0x60C2	02	Interpolation time index	INT8	ro
0x60FE	01	Physical outputs	UINT32	rw
0x60FE	02	Bit mask	UINT32	rw

Sub-index 0 of a record holds the number of entries and is always read-only; the access shown above applies to the value sub-indices. 0x608F Position encoder resolution is read-only as a whole, because the encoder resolution is a property of the motor rather than a setting. 0x607B accepts only 0 (no position range limit), and 0x60C2 reports the SYNC0 cycle, or 0 in free run.

3.8.11 Start values

The CiA402 objects are not saved in the motor. They get the values below each time the EtherCAT interface starts with the CiA402 profile. A master that needs other values writes them every time it configures the motor, for example as start-up commands, see Operation modes.

Where the table names a motor register, the object is read from that register at the start and converted to the CiA402 unit, so CoE and MacTalk show the same value. If the motor cannot be read at that moment, 0x6067 starts at 100, 0x6068 at 6 ms, 0x6085 at 50000 rpm/s, and 0x607D and 0x2100 at 0.

Table 24: CiA402 start values (MAC G3)

Index	Object	Start value
0x6080	Max motor speed	90 % of motor speed limit, rpm
0x6081	Profile velocity	100 rpm
0x6083	Profile acceleration	15000 rpm/s
0x6085	Quick stop deceleration	From register 74 (A2)
0x60C5	Max acceleration	500000 rpm/s
0x6072	Max torque	1000 (per mille)
0x6065	Following error window	0xFFFFFFFF (check off)
0x6066	Following error time out	0 ms
0x6067	Position window	From register 33
0x6068	Position window time	From register 34
0x606D	Velocity window	100
0x606E	Velocity window time	6 ms
0x607D	Software position limit	From registers 28 and 30
0x6099	Homing speeds	100 rpm (sub 1), 50 (sub 2)
0x609A	Homing acceleration	5000 rpm/s
0x2100	Homing torque	From register 41 (T_HOME)
0x608F:1	Encoder increments	Encoder counts per revolution
0x608F:2	Motor revolutions	1
0x6091	Gear ratio	1:1
0x6092:1	Feed	Encoder counts per revolution
0x6092:2	Shaft revolutions	1
0x6097	Acceleration factor	1:1

Objects that are not in the table, for example 0x605A, 0x6098 and 0x607C, start at 0.

3.8.12 Limitations in this release

The following are not implemented in this release and must not be relied on: the digital input/output objects 0x60FD/0x60FE; the analogue input object 0x2101, which always reads 0; the torque limit values 0x60E0/0x60E1 and the abort connection option code 0x6007, which are stored but not used; control word bit 6 (relative positioning); and status word bits 7 (warning) and 11 (internal limit active). The operating modes Velocity (VL), Profile Torque (TQ) and Interpolated Position (IP) are not supported, and the CiA402 profile as a whole is not supported on MIS G3.

3.9 Distributed Clocks

Only EtherCAT

The motor supports both of the synchronisation modes that the EtherCAT Slave Information (ESI) file offers: Free run and Distributed Clocks (DC synchron with SYNC0 event). Both modes are available with both profiles - the choice is not tied to the JVL register profile or to CiA402.

It is the master that decides. If the master activates SYNC0 with a cycle time other than zero, the motor synchronises its cyclic task to the SYNC0 event. If the SYNC0 cycle time is zero, the motor runs free-running and its cyclic task is not locked to the network.

On MAC G3 the SYNC0 cycle time configured in the master must match the motor's own cycle time in motor register 231 (1 ms by default). The motor compares the two when the master requests Safe-Operational or Operational. If they differ, the motor refuses the transition, stays in the state it is already in and reports AL status code 0x0035. MIS G3 makes no such check.

A MAC G3 that will not go to Operational while Distributed Clocks is enabled is therefore almost always a mismatch between the cycle time in the master and the cycle time set in the motor. Check the two against each other before looking anywhere else.

The synchronisation data the motor reports in object 0x1C32 is shown below; 0x1C33 has the same structure for the inputs.

Table 25: Sync manager synchronisation, object 0x1C32

Sub	Name	JVL register profile	CiA402 (MAC G3)
1	Synchronisation type	0 free run, 2 DC SYNC0	same
2	Cycle time	SYNC0 cycle time	same
4	Synchronisation types supported	5 (free run, DC SYNC0)	5
5	Minimum cycle time	500 us	1 ms
6	Calc and copy time	100 us	71 us
9	Delay time	1 us	450 us

With Distributed Clocks the master sends the process data some time before the SYNC0 event - the shift time, which is 30 % of the cycle time by default in TwinCAT - and the motor uses the new data at SYNC0. A master with a large timing jitter should use a larger shift time, so the data always arrives before SYNC0. The shift time is a master setting and affects every device on the network. In TwinCAT the mode and the shift time are set on the drive's DC tab, see figure TT8831-10GB.

[SCREENSHOT NEEDED - CLF] TwinCAT DC tab: Operation Mode DC-Synchron and the Advanced Settings with SYNC0 cycle time and shift time.

TT8831-10GB

Screenshot pending

ADO #1730 · Christian Lotz Felter (clf@jvl.dk)

LB0077 · section 3.9

The screenshot must show:

TwinCAT: the drive selected, DC tab, Operation Mode set to "DC-Synchron" (and the Free Run alternative), and the Advanced Settings dialog with the SYNC0 cycle time and shift time

Placeholder - not for release. Replace this file with the screenshot under the same TT number; the manual then needs no change.

3.10 ESI file

Only EtherCAT

Install the EtherCAT Slave Information (ESI) file in the master's engineering tool before configuring the motor. The two products have their own ESI file, and each file contains one device per profile, so there are four devices to choose between. Select the one that matches the product and the profile in use.

The device name in the master's catalogue is the name from the ESI file. For MIS G3 that name is SMC67, which is the name of the electronics inside the motor - so look for SMC67 rather than MIS in the device list. The product code distinguishes the two profiles as well as the name does, and a master configured against the wrong product code will report a mismatch on the device identity rather than anything that looks like a profile problem.

This manual corresponds to the ESI versions in the table. The version is part of the file name; the revision number inside the file is 1 for all four devices, so the two must not be confused. Both ESI files can be downloaded from www.jvl.dk.

Table 26: EtherCAT Slave Information files

Product	ESI version	Device name in the master
MAC G3	1.0.16	MAC G3 with EC, JVL-8/8 (0x4900)
MAC G3	1.0.16	MAC G3 with EC, Drive profile CiA402 (0x5900)
MIS G3	1.0.2	SMC67 with EC, JVL-8/8 (0x4800)
MIS G3	1.0.2	SMC67 with EC, CiA402 (0x5800)

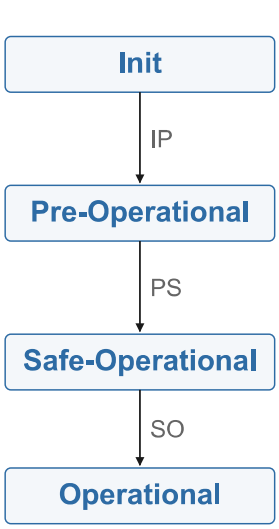
Use the CiA402 device only with a MAC G3. For MIS G3 select the JVL-8/8 device, since the CiA402 profile is not supported on MIS G3 in this release.

3.11 Ethernet over EtherCAT

Only EtherCAT

MacTalk, Modbus TCP and the web server are available over EtherCAT through Ethernet over EtherCAT (EoE); the master assigns the IP parameters. Firmware update over EtherCAT (FoE) is not supported, but MacTalk over the RS485 (Modbus RTU) connection can still update the firmware.

The commissioning sequence is the same in every EtherCAT master, and it is shown in figure TT8814-10GB: the four states the master takes the motor through, what has to be done in each of them, and the two points at which a transition can be refused. The screenshots below show the sequence in TwinCAT.



The motor is here after power-up. No application layer communication: neither SDO nor PDO. The master requests every transition from here on.

The mailbox is up, so SDO access works. Install the ESI file and select the device that matches the product and the profile. Configure the PDO mapping here - it cannot be changed later.

The motor sends its input PDO to the master. There is still no output data from the master, so the motor is not commanded yet.

The master sends output data as well. The CiA402 state machine is evaluated only from here, so a control word written earlier has no effect.

A transition can be refused:

PreOP -> SafeOP is refused with AL status 0x1E if the mapped process data does not match the size the sync manager is configured with.

With Distributed Clocks enabled, the transition is refused unless the SYNC0 cycle time equals the cycle time configured in the motor (motor register 231).

And Operational is not held unconditionally:

If the master stops updating the output data, the sync manager watchdog expires (AL status 0x1B) and the motor drops back to Safe-Operational. If the cause persists, the motor moves between the two repeatedly.

ESI devices: MAC G3 with EC, JVL-8/8 (0x4900) and Drive profile CiA402 (0x5900); SMC67 with EC, JVL-8/8 (0x4800) and CiA402 (0x5800).

The Bootstrap state is not supported - the bootstrap mailbox is disabled in the firmware, so firmware update over EtherCAT (FoE) is not available.

TT8814-10GB

Two things are worth settling before starting. The PDO mapping can only be changed in the Pre-Operational state, so decide the mapping before the master brings the slave further up. And if Distributed Clocks is to be used, the SYNC0 cycle time in the master must match the cycle time configured in the motor, or the motor will not reach Operational at all.
The steps below are the ones that have to be done in order.

- Install the ESI file for the product in the engineering tool of the master, see ESI file. The two products have their own file.
- Connect the master to the IN port of the first motor: LA-I (XG1) on MAC G3, CN2 on MIS G3. The next motor is connected to the OUT port. Check that the port indicator shows a link.
- Scan the network in the master. The motor appears with the device name from

the ESI file: "MAC G3 with EC" or "SMC67 with EC", followed by the profile. If the profile in the name is not the one wanted, change the profile in the motor, save and restart it, and scan again - see Profiles.

- JVL register profile: select the cyclic registers in the motor, see Selecting the cyclic registers. CiA402 profile (MAC G3): set the PDO mapping and the start-up objects, such as 0x6060 Modes of operation, in the master, see PDO mapping and Operation modes.
- Select Free run or Distributed Clocks in the master. With Distributed Clocks on MAC G3, set the SYNC0 cycle time to the motor cycle time in register 231, see Distributed Clocks.
- Take the motor to Operational. The RUN indicator shows the state reached, see Status indicators. The objects of the motor can now be read online over CoE, for example 0x1018 Identity and 0x1008 Device name.

A motor that stays in Safe-Operational, or that moves between Operational and Safe-Operational, is almost always one of those two: a process data size that does not match the sync manager, or a SYNC0 cycle time that does not match the motor. Check both before looking at the profile.

[SCREENSHOT NEEDED - CLF] Step-by-step TwinCAT commissioning - scan for the motor, select the ESI device, map the PDO, and run PP/PV/HM.

Screenshot pending TT8819-10GB

ADO #1730 · Christian Lotz Felter (clf@jvl.dk)

LB0077 · section 3.12

The screenshot must show:

Step-by-step TwinCAT commissioning - scan for the motor, select the ESI device, map the PDO, and run PP/PV/HM

Placeholder - not for release. Replace this file with the screenshot under the same TT number; the manual then needs no change.

A ready-made EtherCAT example for TwinCAT can be downloaded from www.jvl.dk.

3.13 Examples

Only EtherCAT

The examples use the JVL register profile with the default cyclic registers, unless they say otherwise. Registers are given by number; see the register overview for the full description.

3.13.1 Velocity control

In velocity mode the motor runs at the requested velocity with the requested acceleration. With the default cyclic registers everything needed is already in the process data.

MAC G3: set register 5 (V_SOLL) to the velocity and register 6 (A_SOLL) to the acceleration, then write 1 (velocity mode) to register 2. With the default motor cycle of 1 ms and 8192 counts per revolution, V_SOLL = rpm x 2.1845, and one unit of A_SOLL is 457.8 rpm/s. For 1200 rpm, write 2621 to register 5.

MIS G3: set register 3 to the velocity in 0.01 rpm and register 6 to the acceleration in rpm/s, then write 1 to register 2. For 1200 rpm, write 120000 to register 3.

Write 0 to register 2 to make the motor passive again.

3.13.2 Position control

In position mode the motor moves to the requested position with the velocity and acceleration as limits.

MAC G3: write the target to register 3 (P_SOLL, 8192 counts per revolution), the maximum velocity to register 5 and the acceleration to register 6, then write 2 (position mode) to register 2.

MIS G3: write the target to register 4 (409600 counts per revolution), the maximum velocity to register 3 and the acceleration to register 6, then write 2 to register 2.

Follow the move in register 10 (actual position) on both products. The move is finished when the in-position bit is set: bit 4 of register 35 on MAC G3, and bit 4 of register 25 on MIS G3. Register 25 is not in the default MIS G3 read list; add it if it is needed.

3.13.3 General considerations

Keep the error and status information in the read list and monitor it in the master: register 35 on MAC G3 (the motor adds it if it is missing), and on MIS G3 the status bits in register 25 and the error bits in registers 76-79. When an error occurs the motor stops, and the cause is shown there and in the diagnosis history (object 0x10F3).

The motion status is in the same place. MAC G3 register 35: bit 4 in position, bit 5 accelerating, bit 6 decelerating, bit 18 homing done. MIS G3 register 25: bit 4 in position, bit 5 accelerating, bit 6 decelerating, bit 9 homing done.

3.13.4 Homing using only cyclic I/O

Set up the homing in the motor once, in MacTalk, and save it to flash: the homing velocity and position and, on MAC G3, the homing torque. Set the start-up mode to position if the motor should hold its position after the homing - the motor switches to the start-up mode when the homing is done.

MAC G3: keep register 2 in both the read list and the write list. Write the homing mode to register 2: 12 torque homing, 13 sensor homing forward, 14 sensor homing forward and backward, 25 index homing, 26 quick index homing. While the homing runs, the motor does not accept the other cyclic writes. When bit 18 of register 35 is set, write the mode you want next to register 2 - this releases the cyclic writes again.

MIS G3: add register 25 to the read list, and write 13 (sensor homing, one direction) or 14 (both directions) to register 2. MIS G3 does not block the cyclic writes during the homing, so change register 2 to the next mode as soon as bit 9 of register 25 is set.

3.13.5 Relative positioning

The motor must be in position mode. The move is finished when the in-position bit is set.

MAC G3: the interface command register is in the write list by default (slot 6, entered as 983040). Write the distance to motor register 61 (P7) - put it in the write list, or write it once through object 0x2012 sub-index 61. Then write 0x010000F1 to the command register; the motor sets P_SOLL to the actual position plus P7. Write 0 to the command register before the next move.

MIS G3: register 24 is the command register and is in the write list by default. Write the distance to register 330 (P7), then write 113 to register 24. Write 0 to register 24 before the next move. In this release only positive distances are handled correctly.

3.13.6 Saving parameters to flash

MAC G3: write 16 to object 0x2010 to save the EtherCAT interface parameters, 272 to save the motor parameters, or 273 to save both and restart. MIS G3: write 269 to object 0x2000 sub-index 1 to save, or 268 to save and restart.

3.13.7 Clearing errors

MAC G3: write 0x010000E1 to the command register in the write list, or write 225 to object 0x2012 sub-index 211. MIS G3: write 265 to register 24 or to object 0x2000 sub-index 1. With the CiA402 profile, a rising edge on control word bit 7 (Fault reset) clears the error.

3.13.8 CiA402: profile position move

On a MAC G3 with the CiA402 profile and the default PDO mapping:

- Write 1 (PP) to 0x6060, and set 0x6081 profile velocity and 0x6083 profile acceleration.
- Write 0x06, then 0x07, then 0x0F to the control word. The status word shows Ready to switch on, Switched on and then Operation enabled.
- Write the target to 0x607A, then set bit 4 of the control word (0x1F).
- When status word bit 12 (set-point acknowledge) is set, clear bit 4 again (0x0F).
- Status word bit 10 (target reached) is set when the motor is in position.

3.13.9 CiA402: homing

- Write the method to 0x6098, the speed to 0x6099 sub-index 1, the acceleration to 0x609A and, if needed, the home offset to 0x607C.
- Write 6 (HM) to 0x6060 and bring the drive to Operation enabled (0x0F).
- Set bit 4 of the control word (0x1F) to start. Status word bit 12 is set when homing is attained, and bit 13 if it fails.
- Clear bit 4 and change 0x6060 to the mode needed next.

4.1 Introduction

Only EtherNet/IP

EtherNet/IP® is an industrial protocol based on the Common Industrial Protocol (CIP). A generation III motor with EtherNet/IP is an adapter-class (I/O) device that exchanges cyclic data with a scanner (typically a Rockwell PLC).

4.1.1 Introduction to EtherNet/IP

EtherNet/IP brings the CIP object model to standard Ethernet hardware. That gives fast data access using ordinary off-the-shelf equipment, with the availability and the prices that follow from it.

EtherNet/IP divides the traffic into two groups. Explicit messages are acyclic and are used for configuration and for data that is not time critical - motor temperature and setup parameters, for instance. I/O messages are cyclic and carry the time-critical data such as actual position, actual status, actual velocity and actual torque. Generation III motors support both, with up to eight freely configurable input and output words.

The protocol is object based, in the same way as DeviceNet, and follows the standards issued by ODVA. Alongside the standard classes, JVL implements manufacturer specific classes that give access to every register in the motor and in the interface - see CIP objects.

On generation III motors the Industrial Ethernet interface is built into the motor and cannot be removed. The protocol is selected via the Interface option in the Product Part Number Builder (PPNB) when the motor is ordered. The two Ethernet ports are used to daisy chain the motors: in on LA-I (XG1) on MAC G3 or CN2 on MIS G3, out on LA-O (XG2) or CN3. All four ports are M12 4-pin D-coded.

Up to 64 motors can be daisy chained from one master, which saves the switch and is often the simplest wiring. The cost is that the data is delayed slightly for each motor in the chain, and that the network load grows with the number of motors. A switch after the master, distributing to each motor separately, keeps the delay at a minimum.

Addressing, status indicators, MacTalk over Ethernet and the reaction on a lost connection are the same on all protocols and are described in Common setup and functions.

Further information, and the EtherNet/IP specification itself, is available from ODVA at https://www.odva.org.

4.1.2 Abbreviations

These terms are used in this chapter and in the register descriptions.

Table 27: EtherNet/IP abbreviations

Term	Meaning
ACD	Address Conflict Detection
AOI	Add-On Instruction, a reusable block in a Rockwell PLC
CIP	Common Industrial Protocol, the CIP object model
EDS	Electronic Data Sheet, the device description file
LLDP	Link Layer Discovery Protocol
MS / NS	Module Status and Network Status indications
O->T / T->O	Originator to target (scanner to motor) and back
ODVA	The organisation that issues the EtherNet/IP standard
QoS	Quality of Service
RPI	Requested Packet Interval (connection cycle time)

4.1.3 EtherNet/IP specification

The main data of the EtherNet/IP interface are the same on MAC G3 and MIS G3.

Table 28: EtherNet/IP specification

Item	Value
Protocol stack	netX90 EtherNet/IP stack V5.5.0.0
Vendor ID / device type	936 (JVL A/S) / 12 (Comm. Adapter)
Cyclic data	Eight 32-bit registers in each direction
I/O connections	Data In/Out, Listen Only, Input Only
Requested packet interval	1 ms to 1000 ms
Explicit messages	Get and Set Attribute Single
IP configuration	Stored settings or DHCP
Address conflict detection	ACD, enabled by default
Other network functions	QoS, LLDP

4.2 Status indicators

Only EtherNet/IP

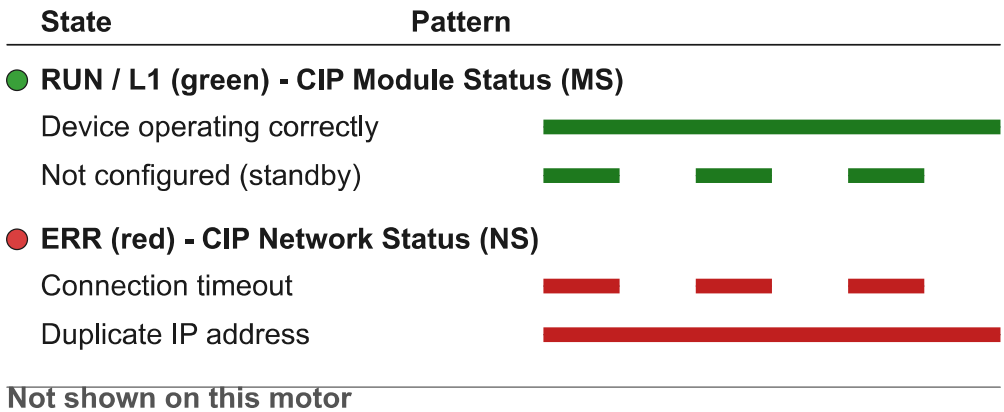
EtherNet/IP uses the two CIP indications Module Status (MS) and Network Status (NS). On the motor the green RUN / L1 indicator carries Module Status and the red ERR indicator carries Network Status; the two port indicators LA1/L2 and LA2/L3 report link and activity as on every other protocol. Both CIP indicators can be flashed from the controller to identify a particular motor in a cabinet.

The conditions the stack reports for each of the two indicators are shown in figure TT8828-10GB. Module Status covers the motor itself and Network Status covers the CIP connections, so the two together separate a motor that is unwell from a motor that simply has nobody talking to it.

Communication LEDs - EtherNet/IP status

TT8828-10GB

On the other protocols the same LEDs are used differently - see the protocol chapter.



Not shown on this motor

The CIP scheme uses two red/green indicators. This motor has one green and one red, so the red half of Module Status (major recoverable and unrecoverable fault) and the green half of Network Status (connected, and no connections) have no indicator to appear on.

Flashing (1 Hz): 500 ms on, 500 ms off. Both indicators also perform a self-test at power-on and can be made to flash from the controller to identify one motor, through Identity object 1.

Source: Hilscher netX 90 EtherNet/IP Adapter (EIS V5.5.0.0), and the CIP specification Volume 2. The stack drives the indicators; the motor firmware does not.

The patterns are produced by the netX90 EtherNet/IP stack (EIS V5.5.0.0) and not by the motor firmware, and they follow the CIP specification, Volume 2. The CIP scheme uses two red/green indicators, while the motor has one green and one red, so the red half of Module Status and the green half of Network Status have no indicator to appear on. A motor that is running normally therefore shows steady green and a dark ERR indicator.

4.3 Cyclic data

Only EtherNet/IP

The cyclic data is exchanged through two instances of the Assembly object (class 0x04): instance 100 carries the output data from the scanner to the motor, and instance 101 the input data from the motor to the scanner. Each is 32 bytes - eight 32-bit registers - and the size is fixed, so the connection must be set up with exactly 32 bytes in each direction. Instances 192 and 193 are the connection points of the Listen Only and Input Only connections, which carry no output data.

4.3.1 Assemblies and connections

Three connections are defined in the EDS file:

Table 29: Connections defined in the EDS file

Connection	Output O->T	Input T->O	Max.
Data, In/Out (owner)	100, 32 bytes	101, 32 bytes	1
Listen Only	192, none	101, 32 bytes	2
Input Only	193, none	101, 32 bytes	2

Data, In/Out is the normal connection and the only one that writes to the motor; one such connection is accepted at a time. Listen Only and Input Only receive the input data alone, up to two of each. A Listen Only connection can only be opened while a Data, In/Out or an Input Only connection is already running.

Only the Data, In/Out connection controls the motor. When it is closed or times out, the cyclic writes stop and the motor reacts as set under Motor reaction on lost connection in Common setup and functions.

The output data from the scanner carries the standard 32-bit run/idle header; the input data from the motor is sent without one.

A scanner that does not use the EDS file can open the Data, In/Out connection as a generic Ethernet device, for instance a Generic Ethernet Module in Studio 5000. Enter the values below, with the data as eight 32-bit words in each direction. The EDS file and the AOI remain the recommended way, see EDS file and AOI.

Table 30: Connection without the EDS file

Parameter	Value
Output (O->T)	Assembly instance 100, 32 bytes
Input (T->O)	Assembly instance 101, 32 bytes
Configuration	Instance 1, size 0
Run/idle header	On the output data only
RPI	1 ms to 1000 ms

4.3.2 Selecting the cyclic registers

Which register goes in each of the eight slots is selected in the motor, in MacTalk or by writing the interface registers directly: the read list (motor to scanner) is registers 525-532 and the write list (scanner to motor) is registers 533-540 (16400-16407 and 16408-16415 in MacTalk on MAC G3). Slot 1 is the first 32-bit word of the assembly, slot 2 the second, and so on. A slot set to 0 ends the list - the slots after it are not used.

A new selection is taken into use when the motor starts, when the interface registers are saved (MAC G3), or at once with the re-initialise command: 21 to interface register 524 (16399 in MacTalk) on MAC G3, or 300 to register 24 on MIS G3. The selection is kept in RAM until it is saved to flash.

4.3.3 Default cyclic registers

Out of the box the cyclic registers are the ones below. Register numbers are the product's own; see Register numbering in Common setup and functions.

Table 31: Default cyclic registers

Slot	MAC G3 read	MAC G3 write	MIS G3 read	MIS G3 write
1	2 mode	2 mode	9 actual mode	2 mode
2	10 position	3 P_SOLL	10 position	4 position
3	11 V_IST_16	5 V_SOLL	12 velocity	3 velocity
4	169 torque	6 A_SOLL	13 torque	6 acceleration
5	35 ERR_STAT	7 T_SOLL	153 diagnosis	7 torque
6	-	983040 command	-	24 command
7	-	-	-	-
8	-	-	-	-

The MAC G3 write list contains the torque limit T_SOLL (register 7), and the MIS G3 write list the torque (register 7). Write a sensible value there as well - with 0 the motor has no torque. 341 is 100 % on MAC G3, and 1000 is 100 % on MIS G3.

4.3.4 Rules for the cyclic registers

- MAC G3: registers 35, 36 and 211 cannot be written cyclically, and register 35 is always in the read list: if it is missing, the motor adds it in the next free slot, or, if the list is full or holds interface registers, puts it in place of the last motor register in the list. Send motor commands and clear errors through the interface command register instead, see Examples.
- MAC G3: an interface register is entered as (register - 509) x 65536 and must come after all motor registers in the list. The interface command register 524 is therefore entered as 983040.
- MIS G3: read-only registers cannot be written cyclically.
- Both: a register can appear only once in each list. A command written cyclically is executed when the value changes to a new value other than 0; to repeat the same command, write 0 for at least one cycle first.
- MIS G3: registers 4, 10, 20 and 330 (P7) are 64-bit. To read one cyclically, put it in the read list followed by the next register number, for example 20 and 21. The two words then carry the upper 32 bits first and the lower 32 bits second. If the slot after a 64-bit register holds another register, that register is sent in place of the upper 32 bits, and only the lower 32 bits of the 64-bit register are read.
- Both: if the selection breaks one of these rules, no registers are exchanged in either direction until it has been corrected and taken into use again, and the motor reports an error: on MAC G3 bits 29 and 24 of register 35 are set and the motor goes passive, on MIS G3 the error code is in registers 35 and 153. The connection itself is not refused.

[EDITORIAL NOTE - NOT FOR RELEASE] ADO #2015 DJO

4.3.5 Requested packet interval

The motor accepts a requested packet interval (RPI) from 1 ms to 1000 ms in both directions and sends its input data cyclically at that interval. The RPI is set in the scanner for each connection.

4.3.6 Cyclic data in the PLC

In the scanner the output and the input data each appear as eight 32-bit words. Word 0 is slot 1 of the selection, word 1 is slot 2 and so on, so the meaning of each word is set by the register selection in the motor, not by the scanner. In Studio 5000 the words are the Data members of the motor's output and input tags, see figure TT8834-10GB.

[SCREENSHOT NEEDED - HC] Studio 5000 controller tags of the motor while the Data, In/Out connection runs: the eight 32-bit output words and the eight input words, with the default MAC G3 cyclic registers.

Screenshot pending TT8834-10GB

ADO #1733 · Hakim Chebbah (HC@jvl.dk)

LB0077 · section 4.3

The screenshot must show:

Studio 5000 controller tags of the motor while the Data, In/Out connection runs: the eight 32-bit output words and the eight input words, with the default MAC G3 cyclic registers

Placeholder - not for release. Replace this file with the screenshot under the same TT number; the manual then needs no change.

The motor implements the standard CIP objects listed below, plus JVL vendor-specific objects that give access to the registers: class 0x64 for the motor registers and, on MAC G3, class 0x65 for the registers of the Ethernet interface.

Table 32: CIP objects in the motor

Class	Object	Remark
0x01	Identity	See Identity object
0x02	Message Router	-
0x04	Assembly	Instances 100, 101, 192, 193
0x06	Connection Manager	See Cyclic data
0x48	Quality of Service (QoS)	DSCP values can be set
0xF5	TCP/IP Interface	See TCP/IP Interface object
0xF6	Ethernet Link	One instance per port
0x109	LLDP Management	LLDP on, sent every 30 s
0x64	JVL motor registers	Both products
0x65	JVL interface registers	MAC G3 only

The instance number is the register number. Unlike EtherCAT and PROFINET no offset is applied here, so the first Ethernet interface register is instance 509 and not instance 0. Class 0x65 exists on the MAC G3 only; on the MIS G3 there is one register list, and every register is reached through class 0x64. See Register numbering in Common setup and functions.

A CIP drive profile is not supported on EtherNet/IP; register access is used instead.
If the drive profile option is set - bit 3 of register 515 (16390 in MacTalk) on MAC G3 or of register 123 on MIS G3 - the motor clears it when EtherNet/IP starts.

4.4.1 Identity object

The Identity object carries the values the scanner matches against the EDS file. They differ between the two products. Device type 12 is the CIP device type Communications Adapter.

Table 33: EtherNet/IP device identity

Identity attribute	MAC G3	MIS G3
1 Vendor ID	936 (JVL A/S)	936 (JVL A/S)
2 Device type	12 (Comm. Adapter)	12 (Comm. Adapter)
3 Product code	18688 (0x4900)	18432 (0x4800)
4 Revision	1.4	1.4
7 Product name	MACxxx G3	MISxxx

The revision is the one the motor reports, 1.4 on both products. The serial number, the status and the state are kept by the EtherNet/IP stack.

[EDITORIAL NOTE - NOT FOR RELEASE] ADO #2156 CLF

The Reset service of the Identity object with type 0 restarts the Ethernet interface on MAC G3 and the whole motor on MIS G3. Changes that have not been saved to flash are lost. Type 1 (return to factory defaults) is accepted and gives the same restart, but it does not restore the factory defaults: the settings saved in flash are loaded again at start-up.

4.4.2 TCP/IP Interface object

The TCP/IP Interface object (class 0xF5, instance 1) holds the IP settings. The motor fills it at start-up from its own IP configuration and DHCP setting, see IP configuration in Common setup and functions.

Table 34: TCP/IP Interface object, class 0xF5

Attr.	Name	In the motor
1	Status	Read only, kept by the stack
2	Configuration capability	BOOTP, DHCP, settable, ACD
3	Configuration control	0 stored settings, 2 DHCP
5	Interface configuration	IP address, mask, gateway
6	Host name	Up to 64 characters
8	TTL value	1 by default
9	Multicast configuration	Default allocation
10	Select ACD	1 (enabled) by default
11	Last conflict detected	Read by the scanner
13	Encapsulation timeout	120 s by default

Configuration control accepts 0 (use the stored IP settings) and 2 (DHCP). Setting it also changes the DHCP setting of the motor - bit 10 of register 515 (16390 in MacTalk) on MAC G3 or of register 123 on MIS G3 - which must then be saved to flash to be kept after a restart. If the stored IP settings are not valid (for example all zero), the interface starts with 192.168.1.1 and mask 255.255.255.0 and puts these values in registers 512-514; save to flash to keep them.

Set the IP address in MacTalk or in registers 512-514, not through the Interface configuration attribute (attribute 5). An address written there is used at once and kept after a restart, but registers 512-514 and MacTalk go on showing the old address.

[EDITORIAL NOTE - NOT FOR RELEASE] ADO #2157 CLF

4.4.3 Ethernet Link object

The Ethernet Link object (class 0xF6) has one instance for each port, labelled L/A IN (instance 1) and L/A OUT (instance 2). Both ports auto-negotiate by default. Interface control (attribute 6) can force the speed and duplex, and admin state (attribute 9) can switch a port off - note that switching a port off breaks a daisy chain at that motor.

Table 35: Ethernet Link object, class 0xF6

Attr.	Name	In the motor
1	Interface speed	Read only, kept by the stack
2	Interface flags	Read only, kept by the stack
3	Physical address	MAC address, read only
4	Interface counters	Read only, kept by the stack
5	Media counters	Read only, kept by the stack
6	Interface control	Auto-negotiate by default, settable
7	Interface type	2 (twisted pair)
8	Interface state	Read only, kept by the stack
9	Admin state	1 enabled (default), 2 disabled
10	Interface label	L/A IN (1), L/A OUT (2)
11	Interface capability	Read only, kept by the stack
768	MDI-X configuration	1 (automatic)

Interface control and admin state are held in RAM only. After a restart both ports auto-negotiate and are enabled again, whatever was written before.

4.5 Explicit messaging

Only EtherNet/IP

Explicit messages are acyclic. They are the way to read or write a motor register that is not in the cyclic data - during commissioning, for parameters that are set once, and for anything that is not time critical. The cyclic assemblies stay reserved for the few registers that have to be exchanged every cycle.

4.5.1 Addressing a register

JVL exposes the registers as vendor-specific CIP object classes. The class selects the register space, the instance is the register number, and the attribute is 1. A register is therefore addressed as class / instance / attribute, with no offset and no translation table.

The MAC G3 has two register spaces, because the motor and the built-in Ethernet interface each have their own. The MIS G3 has one.

Table 36: JVL register object classes

Class	Register space	Product
0x64	Motor registers	Both
0x65	Registers of the Ethernet interface	MAC G3

4.5.2 Services and data

Two services are used: Get Attribute Single to read and Set Attribute Single to write. Register values are 32-bit. On the MIS G3 the registers that are defined as 64-bit are read and written as 64-bit through the same service; on the MAC G3 every register on this path is 32-bit.

A request that cannot be served is answered with a CIP general status code rather than silently ignored: 0x05 (path destination unknown) for instance 0 or an instance above 65535; 0x08 (service not supported) for any other service than Get Attribute Single, Set Attribute Single and No Operation; 0x14 (attribute not supported) for an attribute number above 1; 0x0E (attribute not settable) for a write to a read-only register (on MAC G3 only in class 0x65 - a write to a motor register in class 0x64 is passed on to the motor without this check, see When a write takes effect); and 0x19 (store operation failure) for a write with too few data bytes on MIS G3, or with no data or more than four bytes on MAC G3. A class that the motor does not have (such as 0x65 on MIS G3) is refused with an error status from the EtherNet/IP stack.

Write exactly the size of the register: four bytes, or eight bytes for a 64-bit MIS G3 register. Register values are sent least significant byte first, the normal CIP byte order. A write whose value is outside the range of the register is not reported as an error, so read the register back if the value matters.

[EDITORIAL NOTE - NOT FOR RELEASE] ADO #2020 CLF

4.5.3 When a write takes effect

On the MAC G3 a write to a motor register (class 0x64) is queued to the motor over the internal Modbus® link and the CIP response is sent as soon as the request has been queued. A positive response therefore means that the write was accepted, not that the motor has applied it. Read the register back if the application needs to know that the value has taken effect. Writes to the interface registers (class 0x65) and all writes on the MIS G3 are applied directly.

4.5.4 Message examples

The examples set up and run a velocity move with explicit messages only. All writes are Set Attribute Single (service 0x10) and all reads Get Attribute Single (service 0x0E), always to class 0x64 and attribute 1. The value is the decimal register value, sent as four bytes.

Table 37: Explicit messages for a velocity move

Step	MAC G3	MIS G3
Set the velocity	5: 2185	3: 100000
Set the acceleration	6: 44	6: 20000
Set the torque limit	7: 85	7: 250
Start velocity mode	2: 1	2: 1
Read the actual velocity	11	12
Stop (passive mode)	2: 0	2: 0

The values are given as instance: value. The velocity 2185 on MAC G3 is 1000 rpm and is sent as the four bytes 89 08 00 00. On MIS G3, 100000 is 1000 rpm. The acceleration is 20000 rpm/s (on MAC G3 the nearest value, 44 x 457.8 rpm/s) and the torque limit 25 % on both products; see Implementation guidelines for the scaling. On MAC G3 read register 11 to compare with the value written: register 12 reads 1/16 of the V_SOLL scale.

Explicit messages are not suited to data that is needed often and fast - use the cyclic data for that. In Studio 5000 an explicit message is a MSG instruction of type CIP Generic, see figure TT8835-10GB.

[SCREENSHOT NEEDED - HC] Studio 5000 MSG instruction, Configuration tab: CIP Generic, service Get Attribute Single (0x0E), class 0x64, instance 12, attribute 1, and a 4-byte destination tag.

Screenshot pending TT8835-10GB

ADO #1733 · Hakim Chebbah (HC@jvl.dk)

LB0077 · section 4.5

The screenshot must show:

Studio 5000 MSG instruction, Configuration tab: CIP Generic, service Get Attribute Single (0x0E), class 0x64, instance 12, attribute 1, and a 4-byte destination tag

Placeholder - not for release. Replace this file with the screenshot under the same TT number; the manual then needs no change.

4.6 EDS file and AOI

Only EtherNet/IP

Install the Electronic Data Sheet (EDS) file in the scanner's tool. There is one EDS file for each product, because MAC G3 and MIS G3 report different product codes. For Rockwell Studio 5000, JVL provides an Add-On Instruction (AOI) package - "Add On Instructions for MAC-Motors G3", version 1.0 - which includes the AOI, a data type, an example routine and a separate manual. Both the EDS files and the AOI package can be downloaded from www.jvl.dk.

4.7 Implementation guidelines

Only EtherNet/IP

This section describes the typical use of a generation III motor over EtherNet/IP and which motor registers to put in the cyclic data for the most common applications. It is a guideline for getting started, not an exhaustive description of the register set.

Note that the motor becomes active and may start moving as soon as the mode register (motor register 2) is set to anything other than 0, which is passive mode. Make sure the shaft is free to turn and that nobody is inside the movement of the machine before the mode register is written.

4.7.1 Velocity control

Velocity control uses the mode, the requested velocity, the requested acceleration and the status information. On MAC G3 these are registers 2 (mode), 5 (V_SOLL), 6 (A_SOLL) and 35 (status and errors). On MIS G3 they are registers 2 (mode), 3 (velocity), 6 (acceleration) and 25 (status bits), and the actual mode is register 9. Put them in the cyclic data, so the scanner can both command the motion and see the result of it.

Write the velocity and the acceleration first, and set the mode last. If the mode is written before the two ramp registers, the motor starts with whatever values were left in them from before, which is rarely what is intended. With the ramp values in place, writing 1 to register 2 selects velocity mode and the motor starts using them.

The scaling differs between the two products. MAC G3, with the default motor cycle of 1 ms and 8192 counts per revolution: V_SOLL = rpm x 2.1845, and one unit of A_SOLL is 457.8 rpm/s. MIS G3: the velocity is in 0.01 rpm and the acceleration in rpm/s. For 1200 rpm, write 2621 to register 5 on MAC G3 or 120000 to register 3 on MIS G3.

4.7.2 Position control

Position control needs the mode register set to 2. MAC G3: the requested position is register 3 (P_SOLL, 8192 counts per revolution), and registers 5 and 6 set the velocity and the acceleration used to get there; the actual position is register 10 (P_IST) and the actual velocity register 11 (V_IST_16, on the same scale as V_SOLL). MIS G3: the requested position is register 4 (409600 counts per revolution), the maximum velocity register 3 and the acceleration register 6; the actual position is register 10 and the actual velocity register 12.

Register 20 holds the follow error on both products - the difference between the commanded and the actual position. Together with the status information it tells the controller whether the motion is keeping up, so it belongs in the cyclic input data rather than being read on demand.

4.7.3 Error and status handling

MAC G3: register 35 holds the status and error information, and it is the one register that is always in the cyclic input data - the motor adds it if it is missing (see Rules for the cyclic registers). If an error occurs the motor stops and the cause is left in register 35. MIS G3: the status bits are in register 25, the error code of the last error in register 35 and the error bits in registers 76-79. Put register 25 and the error information in the cyclic input data, so the controller can tell why the motor stopped.

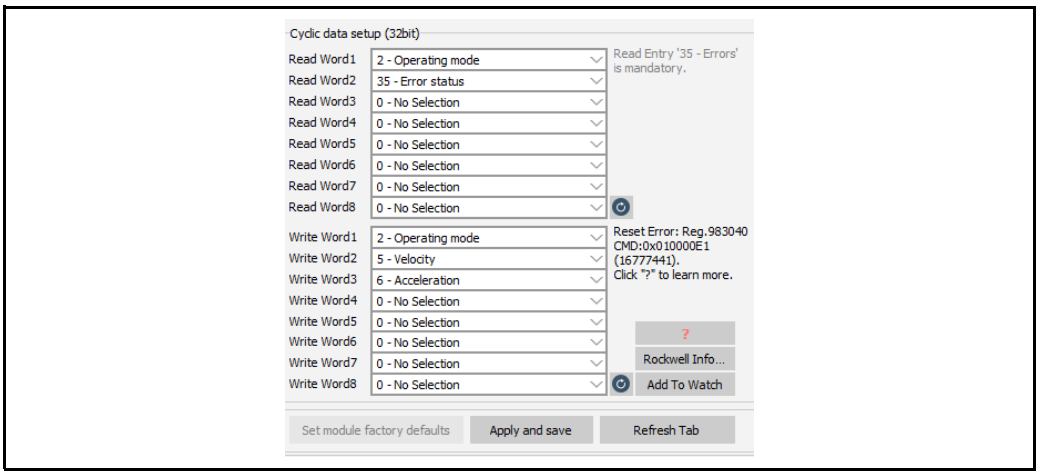
The same registers carry the motion status, so the controller can follow the movement without polling anything else. The most used bits are listed below.

Table 38: Status and error bits in MAC G3 register 35

Bit	Meaning
0	Overload (I2T) error
1	Follow error
2	Function error
3	Regenerative overload error
4	Motor is in position
5	Motor is accelerating
6	Motor is decelerating
7	Software position limit exceeded
8	Temperature too high
9	Undervoltage error
11	Overvoltage error
12	Peak current error
13	Overspeed error
17	Control voltage (24 V) error
18	Homing done
24	Any error (set while any error bit is set)
27	Safe torque off (STO) alarm
29	Error reported by the Ethernet interface

Table 39: Status bits in MIS G3 register 25

Bit	Meaning
3	Motor is at velocity
4	Motor is in position
5	Motor is accelerating
6	Motor is decelerating
7	Motor is rotating in the positive direction
8	Motor is rotating in the negative direction
9	Homing done



4.8 Setup via explicit messages

Only EtherNet/IP

A JVL motor loads its configuration from flash into RAM when it starts, and it holds one configuration. The configuration can be made in MacTalk, which can also save it to a file as a backup, or it can be sent from the scanner with explicit messages - either once, followed by a save to flash, or every time the PLC starts, in which case it need not be saved at all.

The IP settings are set up in MacTalk, see IP configuration in Common setup and functions.

Example: change the velocity, the acceleration and the torque limit and keep them. Send the first three messages of Message examples, and then the save command below. The values in the table are given as class / instance: value, always with Set Attribute Single to attribute 1.

Table 40: Save and clear commands by explicit message

Command	MAC G3	MIS G3
Save motor registers	0x65 / 524: 272	0x64 / 24: 269
Save interface registers	0x65 / 524: 16	-
Save all and restart	0x65 / 524: 273	0x64 / 24: 268
Clear errors	0x64 / 211: 225	0x64 / 24: 265

The new values take effect as soon as they are written; the save only makes them survive a power cycle, and no restart is needed. The same commands can be sent through the command register in the cyclic write list, see Examples.

In Studio 5000 each write is a MSG instruction with the service Set Attribute Single, see figure TT8836-10GB.

[SCREENSHOT NEEDED - HC] Studio 5000 MSG instruction, Configuration tab: CIP Generic, service Set Attribute Single (0x10), class 0x64, instance 5, attribute 1, and a 4-byte source tag with the new value.

Screenshot pending TT8836-10GB

ADO #1733 · Hakim Chebbah (HC@jvl.dk)

LB0077 · section 4.8

The screenshot must show:

Studio 5000 MSG instruction, Configuration tab: CIP Generic, service Set Attribute Single (0x10), class 0x64, instance 5, attribute 1, and a 4-byte source tag with the new value

Placeholder - not for release. Replace this file with the screenshot under the same TT number; the manual then needs no change.

The switch has to match the size of the network and the requested packet interval (RPI). A switch is also what connects several separate networks to each other. For a network of moderate size and a reasonable packet interval a 1 Gbit/s switch is normally adequate, provided it offers auto negotiation and full duplex at 100 Mbit/s.

Port mirroring is worth having. It routes the traffic out on a separate port where a network analyser can be connected, which is what makes troubleshooting and performance monitoring possible at all once the network is in a machine.

On the Data, In/Out and Input Only connections the input data from the motor can be sent multicast or point-to-point - the EDS file allows both, and the choice is made in the scanner. The Listen Only connection is multicast only. With multicast, a switch with IGMP snooping keeps the I/O traffic away from the ports that have not joined the group, and on a network with several motors and other multicast traffic that is the difference between a quiet network and one where every device sees every frame. The output data from the scanner is always point-to-point.

The motor has a built-in two-port switch, which is what makes the daisy chain possible without external hardware. It is useful for a limited number of motors on one line. The cost is that every motor forwards the traffic for the motors behind it, so the packet interval that can be achieved goes down as the chain gets longer. A separate switch, with shorter chains hanging off it, is the way out when the line gets long.

The commissioning sequence is the same in every EtherNet/IP scanner, and only the names of the dialogs change. The steps below are the ones that have to be done in order.

- Give the motor an IP address. Until it has one it cannot be found by the scanner. Addressing is common to all protocols and is described in Common setup and functions.
- Install the EDS file in the scanner tool, so that the tool can match the identity the motor reports. Check the product code against the table in CIP objects if the tool reports an identity mismatch: MAC G3 and MIS G3 have different codes and different EDS files.
- Add the motor to the I/O tree and select the connection. Data, In/Out is the normal choice and owns both directions; Input Only and Listen Only give the input assembly alone and are used when another controller owns the outputs.
- Set the requested packet interval (RPI). The motor accepts 1 ms to 1000 ms in each direction, but the RPI has to be achievable for the whole network, not only for one motor - see Ethernet switches.
- Decide which registers go in the cyclic data. The assemblies carry eight 32-bit registers in each direction and the choice is made in the motor, not in the scanner; see Implementation guidelines for the usual sets.
- Go online. The status information in the cyclic input data - register 35 on MAC G3, register 25 on MIS G3 - tells the controller whether the motor is ready, so put it in the selection before anything else.

  A connection that is refused, or that drops again shortly after it is established, is almost always an identity that does not match the EDS file, a connection size other than 32 bytes, or an RPI the network cannot keep. A cyclic register selection the motor cannot use does not stop the connection: the motor then exchanges no registers in either direction and reports an error (register 35 on MAC G3, registers 35 and 153 on MIS G3), so check the selection in MacTalk if the data does not change.

  [SCREENSHOT NEEDED - HC] Adding the motor in Studio 5000 using the EDS/AOI, setting the IP address, and exchanging the 8x32 cyclic registers.

  Screenshot pending TT8820-10GB

  ADO #1733 · Hakim Chebbah (HC@jvl.dk)

  LB0077 · section 4.10

  The screenshot must show:

  Adding the motor in Studio 5000 using the EDS/AOI, setting the IP address, and exchanging the 8x32 cyclic registers

  Placeholder - not for release. Replace this file with the screenshot under the same TT number; the manual then needs no change.

  The connection and the RPI are set in the properties of the motor in the scanner's tool. In Studio 5000 this is the Connection tab, see figure TT8837-10GB.

  [SCREENSHOT NEEDED - HC] Studio 5000 Module Properties of the motor, Connection tab: the Data, In/Out connection and the Requested Packet Inter-

val (RPI), which the motor accepts from 1 ms to 1000 ms.

TT8837-10GB

Screenshot pending

ADO #1733 · Hakim Chebbah (HC@jvl.dk)

LB0077 · section 4.10

The screenshot must show:

Studio 5000 Module Properties of the motor, Connection tab: the Data, In/Out connection and the Requested Packet Interval (RPI), which the motor accepts from 1 ms to 1000 ms

Placeholder - not for release. Replace this file with the screenshot under the same TT number; the manual then needs no change.

4.11 Examples

Only EtherNet/IP

The examples use the default cyclic registers, unless they say otherwise. Registers are given by number; see the register overview for the full description.

4.11.1 Velocity control

In velocity mode the motor runs at the requested velocity with the requested acceleration. With the default cyclic registers everything needed is already in the cyclic data.

MAC G3: set register 5 (V_SOLL) to the velocity and register 6 (A_SOLL) to the acceleration, then write 1 (velocity mode) to register 2. For 1200 rpm, write 2621 to register 5.

MIS G3: set register 3 to the velocity in 0.01 rpm and register 6 to the acceleration in rpm/s, then write 1 to register 2. For 1200 rpm, write 120000 to register 3.

Write 0 to register 2 to make the motor passive again.

4.11.2 Position control

In position mode the motor moves to the requested position with the velocity and acceleration as limits.

MAC G3: write the target to register 3 (P_SOLL, 8192 counts per revolution), the maximum velocity to register 5 and the acceleration to register 6, then write 2 (position mode) to register 2.

MIS G3: write the target to register 4 (409600 counts per revolution), the maximum velocity to register 3 and the acceleration to register 6, then write 2 to register 2.

Follow the move in register 10 (actual position) on both products. The move is finished when the in-position bit is set: bit 4 of register 35 on MAC G3, and bit 4 of register 25 on MIS G3. Register 25 is not in the default MIS G3 read list; add it if it is needed.

4.11.3 Homing using only cyclic I/O

Set up the homing in the motor once, in MacTalk, and save it to flash: the homing velocity and position and, on MAC G3, the homing torque. Set the start-up mode to position if the motor should hold its position after the homing - the motor switches to the start-up mode when the homing is done.

MAC G3: keep register 2 in both the read list and the write list. Write the homing mode to register 2: 12 torque homing, 13 sensor homing forward, 14 sensor homing forward and backward, 25 index homing, 26 quick index homing. While the homing runs, the motor does not accept the other cyclic writes. When bit 18 of register 35 is set, write the mode you want next to register 2 - this releases the cyclic writes again.

MIS G3: add register 25 to the read list, and write 13 (sensor homing, one direction) or 14 (both directions) to register 2. MIS G3 does not block the cyclic writes during the homing, so change register 2 to the next mode as soon as bit 9 of register 25 is set.

4.11.4 Relative positioning

The motor must be in position mode. The move is finished when the in-position bit is set.

MAC G3: the interface command register is in the write list by default (slot 6, entered as 983040). Write the distance to motor register 61 (P7) - put it in the write list, or write it once with an explicit message to class 0x64, instance 61. Then write 0x010000F1 to the command register; the motor sets P_SOLL to the actual position plus P7. Write 0 to the command register before the next move.

MIS G3: register 24 is the command register and is in the write list by default. Write the distance to register 330 (P7), then write 113 to register 24. Write 0 to register 24 before the next move. In this release only positive distances are handled correctly. Register 330 is 64-bit: an explicit message must carry eight bytes.

4.11.5 Saving parameters to flash

Use the save commands in Setup via explicit messages. They can be sent as an explicit message or written to the command register in the cyclic write list: register 24 on MIS G3, and the interface command register (983040) on MAC G3.

4.11.6 Clearing errors

MAC G3: write 0x010000E1 to the command register in the cyclic write list, or write 225 to class 0x64, instance 211 with an explicit message. MIS G3: write 265 to register 24, cyclically or with an explicit message.

4.12 ODVA conformance

Only EtherNet/IP

EtherNet/IP has not yet passed ODVA conformance testing for generation III; conformance is planned.

4.12 ODVA conformance

Only EtherNet/IP

5 Modbus TCP/UDP

5.1 Introduction

Modbus® TCP and Modbus UDP are available on all variants for register access, in parallel with the selected real-time protocol. (On EtherCAT®, Modbus TCP is carried inside the EoE tunnel.)

5.1.1 Introduction to Modbus TCP/UDP

Modbus TCP is the Modbus variant used over TCP/IP networks. The motor listens on port 502.

MAC G3 and MIS G3 also support Modbus UDP - the same protocol carried over UDP instead of TCP. It is faster, because there is no connection to set up and maintain, but it has no connection control. It is in effect Modbus RTU without the checksum, since the lower layers already provide checksum protection.

Modbus TCP and UDP are built on the standard TCP/IP protocols, so they work anywhere ordinary Ethernet is available and place no special requirements on the Ethernet hardware - unlike some of the other Industrial Ethernet protocols.

Modbus TCP/UDP is built into every MAC G3 and MIS G3 with Industrial Ethernet and runs in parallel with the selected real-time protocol, so a controller or a PC that does not speak the real-time protocol can still reach the registers. On EtherCAT it is carried inside the EoE tunnel.

Addressing, status indicators and MacTalk over Ethernet are described in Common set-up and functions. The reaction on lost connection described there applies to the cyclic connection of the real-time protocol: Modbus TCP/UDP has no connection monitoring of its own, see Connection.

Further information is available from the Modbus Organization, https://www.modbus.org.

5.1.2 Abbreviations

These terms are used in this chapter and in the register descriptions.

Table 41: Modbus abbreviations

Term	Meaning
Exception	The answer to a request the motor cannot carry out
Function code	The code in a request that says which operation is wanted
Holding register	The Modbus register type the motor exposes its registers as
MBAP	Modbus Application Protocol header, first 7 bytes of a frame
PDU	Protocol Data Unit: the function code and its data
RTU	Remote Terminal Unit, the serial Modbus variant
TCP	Transmission Control Protocol, connection-based
UDP	User Datagram Protocol, without connections
Unit id	Unit identifier, the Modbus address field in the MBAP header

5.2 Status indicators

Modbus TCP/UDP has no status indicators of its own. It runs alongside the real-time protocol the motor was ordered with, so the ERR and RUN / L1 indicators follow that protocol and not the Modbus connection. The PWR indicator reports the motor itself, and the two port indicators report link and activity. All of them are described under Status indicators in Common setup and functions.

A Modbus client that cannot reach the motor therefore gets no help from the indicators beyond the port LEDs, which at least show whether the cable is live. Check the IP address of the MacTalk / Modbus TCP interface and the port number (502) instead - the Modbus address plays no part over Ethernet, see Modbus address.

The motor listens for Modbus TCP and Modbus UDP on port 502, on the IP address of the MacTalk / Modbus TCP interface. That address is separate from the address of the real-time protocol.

5.3.1 Network configuration

The MacTalk / Modbus TCP interface always uses a static IP address. Set the IP address, subnet mask and gateway in MacTalk or in registers 553, 554 and 555 (16428-16430 in MacTalk on MAC G3), as described under IP configuration in Common setup and functions, and save them. The Modbus master must be on the same subnet, or reach the motor through the gateway. On EtherCAT the master can also set these three values through Ethernet over EtherCAT (EoE), see EtherCAT.

5.3.2 TCP and UDP

- Modbus TCP: open a TCP connection to port 502. Up to eight connections can be open at the same time. Each request must arrive in one TCP segment, which is what Modbus masters normally do.
- Modbus UDP: send each request as a UDP datagram to port 502. The answer goes back to the address and port it came from. Several UDP masters, and TCP masters, can use the motor at the same time.
- Requests sent to a broadcast address are not answered, and register reads and writes sent that way are not carried out.

The motor does not close a TCP connection by itself. A connection that is dropped without being closed - for example when the master is switched off - keeps its place, and when all eight places are taken a new connection is not accepted until the motor is restarted. Let the master close its connection before it stops.

Modbus TCP/UDP has no connection monitoring. If the Modbus master stops, the motor keeps the last values it received and goes on running. Supervise the motor from the master, or use the real-time protocol, whose lost-connection reaction is described in Common setup and functions.

Register 1838 (18222 in MacTalk on MAC G3) counts the Modbus TCP/UDP frames the motor has dropped as malformed. A value that climbs points to a master or a network problem.

5.3.3 Switching the Ethernet server off

Modbus TCP/UDP and MacTalk over Ethernet can be switched off together, for example where only the real-time protocol may be reachable on the network: set bit 14 of register 515 (16390 in MacTalk) on MAC G3, or bit 16 of register 123 on MIS G3. The setting is read when the motor starts, so save it and restart the motor. MacTalk over the RS485 connection is not affected.

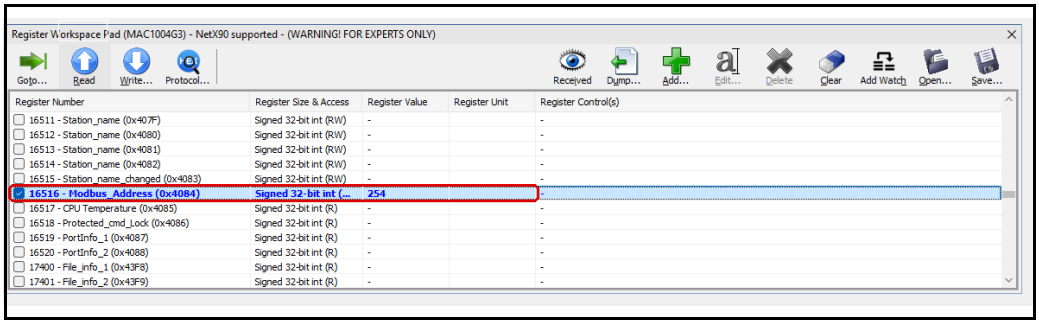
5.4 Modbus address

5.4.1 General description

The Modbus address (unit id), register 641 (16516 in MacTalk on MAC G3), is the address the motor puts in its answers on the serial Modbus RTU connection (RS485), and function code 0x11 reports it there. Over Modbus TCP/UDP the motor is found by its IP address alone: it answers a request whatever unit identifier the request carries, and returns the same unit identifier in the answer. The range is 1 to 254, and the default address is 254.

5.4.2 Configuration via MacTalk

MacTalk has no field for the Modbus address on its tabs. Set it as a register, either in the Watch Registers list at the right of the MacTalk window or in the Register Workspace Pad, then save it in the motor. On MAC G3 the register is 16516 in MacTalk (Modbus_Address), on MIS G3 it is 641.



5.4.3 Configuration via motor registers

The Modbus address can also be written directly to its register - from the register view in MacTalk, over Modbus itself or over the real-time protocol:

- Register 641 - Modbus address (unit id), 1 to 254, default 254. On MAC G3 it is an Ethernet interface register: in MacTalk it is register 16516, and over Modbus TCP/UDP it is at Modbus address 33032; on MIS G3 it is register 641 at address 1282. Save it to flash to keep it after a power cycle, see Commands under Examples.

5.5 Register access

Every register is a 32-bit value and occupies two consecutive Modbus holding registers, so both the address and the quantity must be even. The supported function codes are 0x03 (Read Holding Registers), 0x10 (Write Multiple Registers), 0x17 (Read/Write Multiple Registers) and 0x2B (Read Device Identification). 0x06 (Write Single Register) is accepted but must not be used in this release, see Function codes. On MAC G3 the Modbus address range is split in two halves: addresses below 32768 reach the motor registers in the motor controller, and addresses from 32768 (bit 15 set) reach the registers of the Ethernet interface.
The two halves are a MAC G3 arrangement. The Modbus address of a motor register is twice the register number. The address of an Ethernet interface register is twice the sum of the register number and 15875, so interface register 509 is at Modbus address 32768 and register 510 at 32770. From interface register 1016 the constant is 16384 instead, and the numbers in between are refused. On the MIS G3 there is one list and one numbering, so every register is simply at twice its own number. See Register numbering in Common setup and functions.

Not supported: function code 0x04 (Read Input Registers) and the coil and discrete input functions - use 0x03 instead. Function code 0x11 (Report Slave ID) is answered on the serial connection only; over Ethernet it gives exception 0x01. The motor also uses a few JVL function codes of its own (0x64 to 0x6B) for file transfer and firmware update from MacTalk; they are not meant for a controller.

5.5.1 Modbus addresses

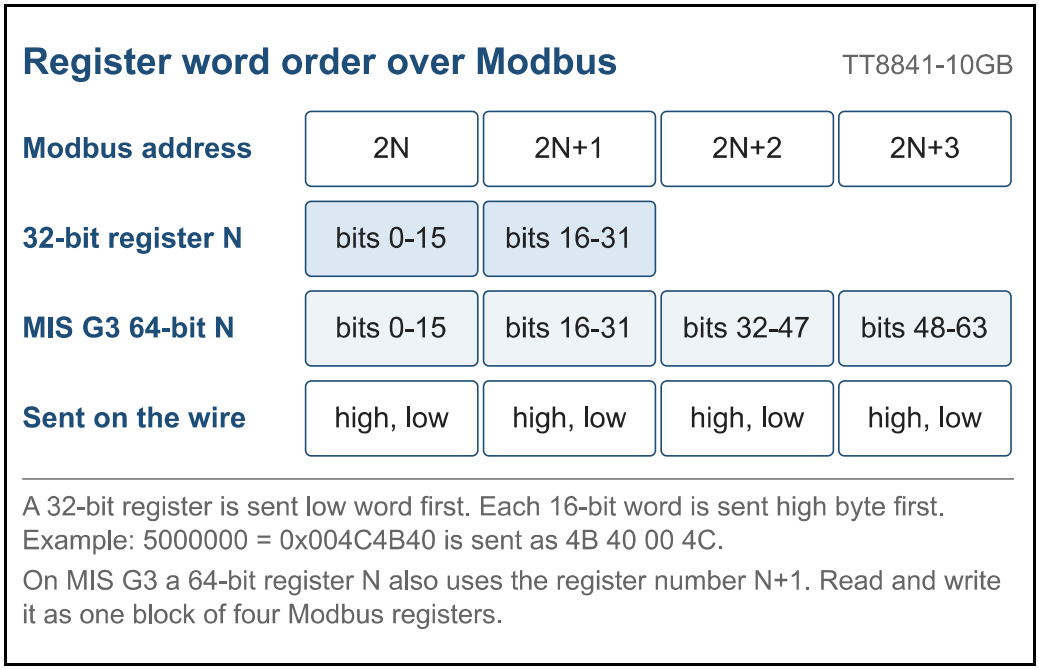
On MIS G3, and for the MAC G3 motor registers, the Modbus address of register N is 2 x N. The MAC G3 motor registers are 0 to 511, so their addresses are 0 to 1022; an address above that is refused. The MAC G3 interface registers use the constants described under Register numbering in Common setup and functions. The table gives the addresses of some registers that are often used; where two numbers are given, the first is the MAC G3 register and the second the MIS G3 register.

Table 42: Useful registers and their Modbus addresses

Register	Number	MAC G3 address	MIS G3 address
Motor command	211 / 24	422	48
Interface command	524	32798	-
IP address (Modbus)	553	32856	1106
Setup bits (server off)	515 / 123	32780	246
Modbus address	641	33032	1282
Bad frame counter	1838	36444	3676

5.5.2 Word order

Each Modbus register carries 16 bits and is sent high byte first, as the Modbus standard says. A 32-bit register is sent low word first: bits 0-15 at the even address 2N and bits 16-31 at the odd address 2N+1, as figure TT8841-10GB shows. Many masters call this word order "word swapped" or "CD AB"; select it in the master to see the values as 32-bit numbers.



Example: 5000000 is 0x004C4B40 and is sent as the bytes 4B 40 00 4C.

5.5.3 64-bit registers on MIS G3

Some MIS G3 registers are 64-bit: the positions and the values related to them, for example register 4 (requested position), 10 (actual position), 20 (follow error) and 330 (P7). A 64-bit register N also takes the register number N+1, and so it occupies four Modbus registers from address 2N, lowest 16 bits first.

Write a 64-bit register with one 0x10 request of all four Modbus registers, so that both halves are taken over together. Reading only the first two Modbus registers gives the lower 32 bits. MAC G3 has no 64-bit registers.

Table 43: MIS G3 register 4 = -409600 in one 0x10 request

Address	Bits of register 4	Bytes sent
8	0-15	C0 00
9	16-31	FF F9
10	32-47	FF FF
11	48-63	FF FF

[EDITORIAL NOTE - NOT FOR RELEASE] ADO #2017 CLF

5.6 Function codes

5.6.1 Frame layout

A Modbus TCP frame is the MBAP header followed by the function code and its data, the PDU, as figure TT8840-10GB shows. Modbus UDP carries exactly the same frame in a UDP datagram.

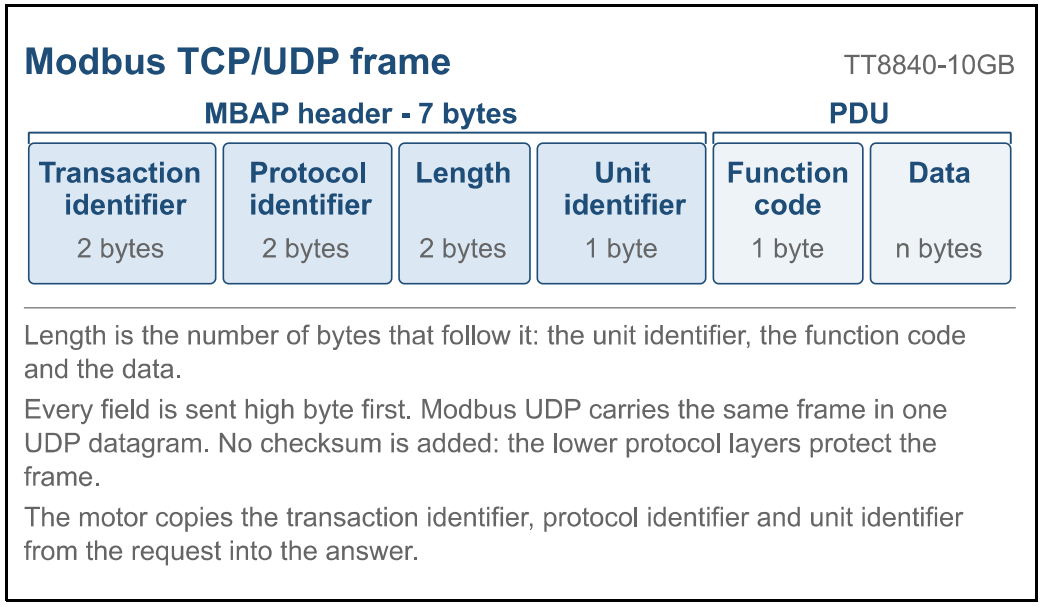


Table 44: MBAP header

Field	Bytes	Content
Transaction identifier	2	Chosen by the master, copied to the answer
Protocol identifier	2	0 for Modbus (not checked by the motor)
Length	2	Number of bytes that follow
Unit identifier	1	Any value, copied to the answer

The function codes below are supported. The examples use a MAC G3; on MIS G3 only the register numbers change, see Register access.

Table 45: Supported function codes

Code	Name	Limit
0x03	Read Holding Registers	2 to 124 Modbus registers, even
0x10	Write Multiple Registers	2 to 124 Modbus registers, even
0x17	Read/Write Multiple Registers	Read and write 2 to 124 each, even
0x2B	Read Device Identification	MEI type 0x0E
0x06	Write Single Register	Do not use, see below
0x11	Report Slave ID	Serial connection only

5.6.2 Read Holding Registers (0x03)

Reads 2 to 124 Modbus registers - up to 62 registers - from consecutive addresses. Use an even start address and an even quantity. An odd start address is not refused on a read: the answer is the register below it. The answer carries two bytes per Modbus register, in the word order described under Register access.

Table 46: Read Holding Registers (0x03): request

Byte	Field	Example: register 12
0-6	MBAP header	00 01 00 00 00 06 01
7	Function code	03
8-9	Start address (2 x register)	00 18
10-11	Quantity (2 per register)	00 02

Table 47: Read Holding Registers (0x03): answer

Byte	Field	Example: value 2621
0-6	MBAP header	00 01 00 00 00 07 01
7	Function code	03
8	Byte count	04
9-10	Bits 0-15 of the register	0A 3D
11-12	Bits 16-31 of the register	00 00

In the example register 12 of a MAC G3 holds 2621 (0x00000A3D), the actual velocity at 1200 rpm. An odd quantity above 1 is refused with exception 0x02. A quantity of 1 is accepted for MacTalk, but its answer is not a standard Modbus answer - do not use it.

5.6.3 Write Multiple Registers (0x10)

Writes 2 to 124 Modbus registers - 1 to 62 registers. The start address and the quantity must be even, the start address must not be 0, and the byte count must be twice the quantity. The answer repeats the start address and the quantity: 00 02 00 00 00 06 01 10 00 06 00 02 for the example below.

Table 48: Write Multiple Registers (0x10): request

Byte	Field	Example: 5000000 to register 3
0-6	MBAP header	00 02 00 00 00 0B 01
7	Function code	10
8-9	Start address	00 06
10-11	Quantity	00 02
12	Byte count (2 x quantity)	04
13-14	Bits 0-15 of the register	4B 40
15-16	Bits 16-31 of the register	00 4C

Registers are written in order from the start address. A read-only register in the block, or a value outside the range of a register, is skipped without an exception, see Exception responses.

5.6.4 Read/Write Multiple Registers (0x17)

Writes one block and reads another in the same request, which is useful for exchanging set-points and actual values in one poll. The write is carried out before the read. Each block follows the rules of 0x03 and 0x10, and neither start address may be 0. The example writes 5000000 to register 3 (P_SOLL) and reads register 10 (P_IST).

Table 49: Read/Write Multiple Registers (0x17): request

Byte	Field	Example
0-6	MBAP header	00 03 00 00 00 0F 01
7	Function code	17
8-9	Read start address	00 14 (register 10)
10-11	Read quantity	00 02
12-13	Write start address	00 06 (register 3)
14-15	Write quantity	00 02
16	Write byte count	04
17-20	Write data	4B 40 00 4C

The answer is 00 03 00 00 00 07 01 17 04 followed by the four bytes of register 10. A write in a 0x17 request that fails is not reported; check it with a read, or use 0x10 for writes that must be confirmed.

5.6.5 Write Single Register (0x06)

Function code 0x06 writes one 16-bit Modbus register, which is half of a register in the motor. In this release the motor does not put the two halves together correctly, so 0x06 must not be used to write registers. Use 0x10 with a quantity of 2 instead: it writes one whole register in one request.

[EDITORIAL NOTE - NOT FOR RELEASE] ADO #2016 CLF

5.6.6 Read Device Identification (0x2B)

Function code 0x2B with MEI type 0x0E returns identification strings. Read device ID code 1 returns the basic objects 0x00 to 0x02, code 2 the regular objects 0x03 to 0x06, and code 4 one object. The extended objects (code 3) are not supported and give exception 0x02; an MEI type other than 0x0E gives exception 0x03.

Table 50: Device identification objects (0x2B / 0x0E)

Object	Name	Content
0x00	VendorName	JVL A/S
0x01	ProductCode	JVL product code, hexadecimal
0x02	MajorMinorRevision	Firmware version, Vx.yy
0x03	VendorUrl	www.jvl.dk
0x04	ProductName	MIS
0x05	ModelName	Motor type, for example MIS23x
0x06	UserApplicationName	The real-time protocol in use

[EDITORIAL NOTE - NOT FOR RELEASE] ADO #2026 CLF

5.7 Exception responses

A request the motor cannot carry out is answered with a Modbus exception instead of a normal response. The function code in the answer has the highest bit set, and the data byte that follows is the exception code.

Table 51: Modbus exception codes

Code	Name	When the motor sends it
0x01	Illegal function	Unsupported function code, or 0x11 over Ethernet
0x02	Illegal data address	No such register or object, odd count, too many
0x03	Illegal data value	Byte count wrong, 0x06 refused, bad MEI type
0x04	Slave device failure	MAC G3: write did not reach the motor controller
0x06	Slave device busy	MAC G3: file transfer busy, repeat the request

Whether a refused write is reported depends on the function code. A 0x06 write to a read-only register, or of a value outside the range of the register, is answered 0x03. With 0x10 and 0x17 such a register is skipped and the answer is a normal one, so read the register back if the result matters. On MAC G3 a 0x06 or 0x10 write that does not reach the motor controller is answered 0x04; a value the motor controller itself refuses is not reported back.

5.8 Response time

The motor answers each request as soon as it has been carried out. How long that takes depends on the product and on where the registers are:

- MIS G3: all registers are in the motor itself, and the answer comes at once.
- MAC G3: the interface registers (from address 32768) are answered at once. The motor registers are in the motor controller and are read or written through the internal connection when the request arrives. A block of consecutive registers is moved in one exchange, so one request for ten registers is much quicker than ten requests for one.

The motor handles one request at a time. A master that waits for each answer before it sends the next request gets the shortest and most even poll cycle.

MAC G3: if the Ethernet interface has no contact with the motor controller, a read of motor registers is answered with 0 in every register (only the motor type in register 203 is filled in) rather than with an exception, and a write is answered with exception 0x04.

5.9 Quick start

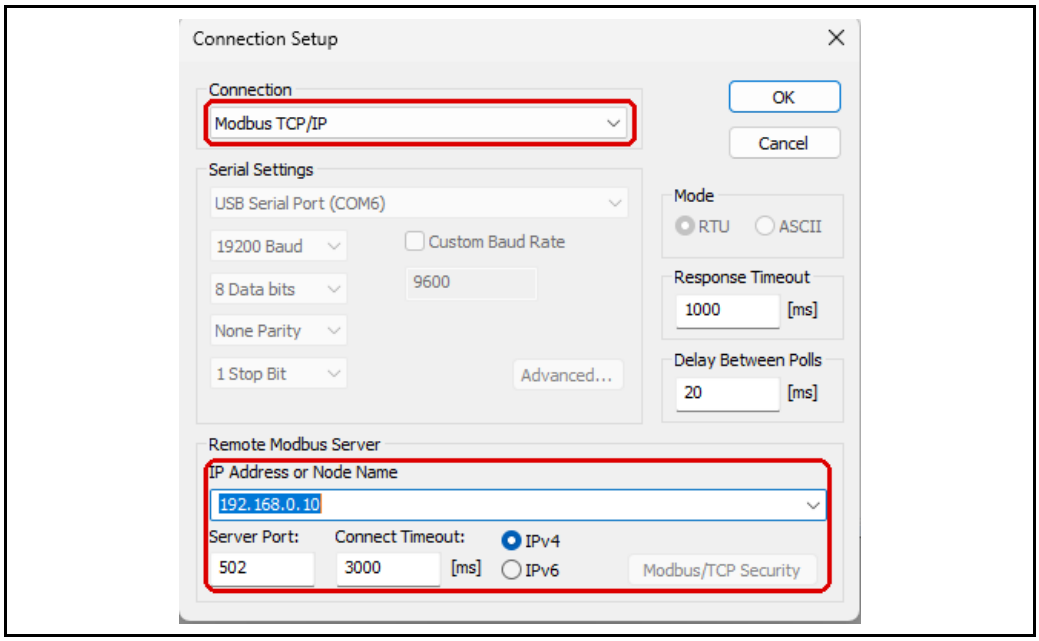
This section connects a PC to the motor with a general-purpose Modbus master utility and makes the motor turn. Modbus Poll (www.modbustools.com) is used as the example; any Modbus master works the same way.

5.9.1 Set the IP address

- Connect MacTalk to the motor and set the IP address, subnet mask and gateway of the MacTalk / Modbus TCP interface, see IP configuration in Common setup and functions. Save the settings and restart the motor.
- Connect the PC to one of the Ethernet ports of the motor, and give the PC an IP address on the same subnet.

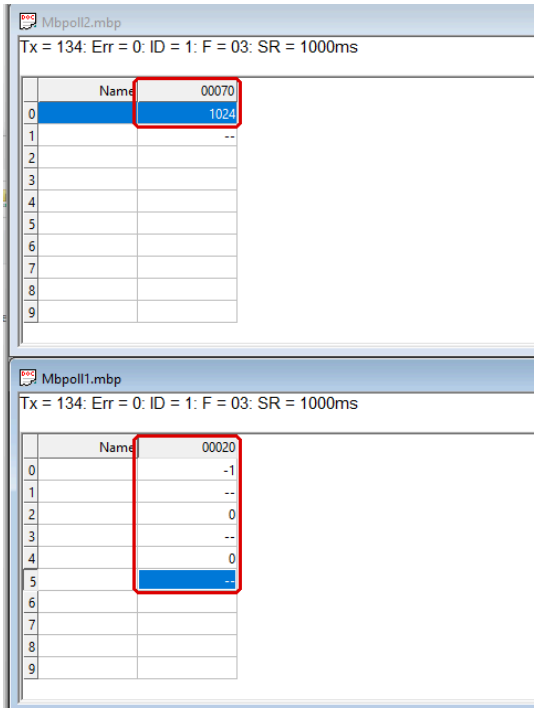
5.9.2 Connect

- In the Modbus master choose to connect, select Modbus TCP/IP or Modbus UDP/IP, and enter the IP address of the motor and port 502. The unit identifier (slave id) can be left at any value.



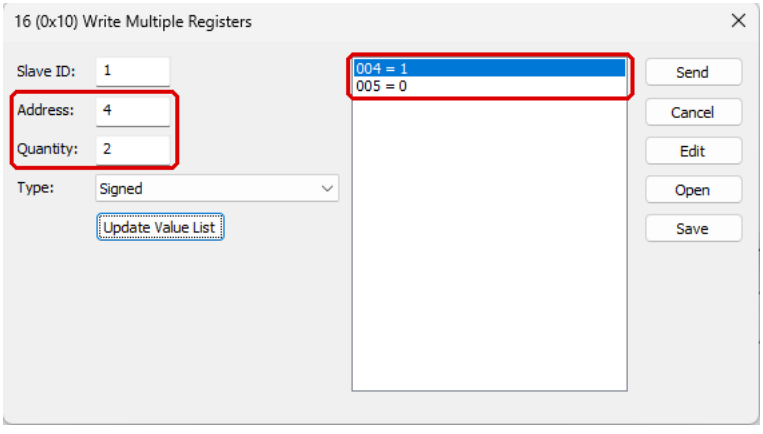
5.9.3 Poll registers

- Set up a poll with function 03, address 20 and quantity 6. That polls registers 10 to 12: the actual position and the actual velocity.
- Open a second poll with address 70 and quantity 2 on MAC G3 (register 35), or address 50 and quantity 2 on MIS G3 (register 25). Show the values of both polls in the 32-bit format with the low word first (in Modbus Poll: 32-bit Signed, Little-endian byte swap). A display format applies to the selected cells, so select all rows of the window before choosing it.



5.9.4 Write to the motor

- Use function 16 (Write Multiple Registers). Write the velocity and the acceleration, as in Velocity control under Examples, and make sure the torque in register 7 is not 0.
- Write 1 to register 2 (address 4, quantity 2). The motor starts in velocity mode. Write 0 to register 2 to make it passive again.



5.10 Examples

This section shows how the motor is controlled over Modbus TCP with a few worked examples. Any Modbus master can be used; for trying things out by hand a general-purpose Modbus master utility is usually the quickest way, see Quick start. Register numbers are each product's own, so the MAC G3 and the MIS G3 values are given separately.

The Modbus address of register N is twice N (the MAC G3 interface registers excepted, see Register access). Work it out once for the registers used here. A write to an odd address is refused with exception 0x02, but a read from an odd address returns the register below it without an error, so an address that is one out can go unnoticed on a read. On MIS G3 the positions are 64-bit and take four Modbus registers each.

Table 52: Modbus addresses of the registers used in the examples

Quantity	MAC G3 reg	MAC G3 addr	MIS G3 reg	MIS G3 addr
Mode	2	4	2	4
Requested position	3	6	4 (64-bit)	8
Requested velocity	5	10	3	6
Acceleration	6	12	6	12
Torque (limit)	7	14	7	14
Actual position	10	20	10 (64-bit)	20
Actual velocity	12	24	12	24
Follow error	20	40	20 (64-bit)	40
Status	35	70	25	50
Motor command	211	422	24	48
P7 (relative move)	61	122	330 (64-bit)	660
Interface command	524	32798	-	-

5.10.1 Velocity control

Set up polling of the actual velocity (register 12 on both products) and the status - register 35 on MAC G3, register 25 on MIS G3 - first, so the motor can be watched while it is commanded. With those on poll, the errors and the actual speed are visible throughout the rest of the example.

Then write the acceleration and the velocity. MAC G3: register 6 (A_SOLL) and register 5 (V_SOLL). MIS G3: register 6 in rpm/s and register 3 in 0.01 rpm. Both are written before the mode, because the motor uses whatever is in them the moment the mode is set.

Finally write 1 to register 2 (both products). That selects velocity mode and the motor starts running with the ramp values just written. Writing 0 to the same register returns the motor to passive mode and de-energises the shaft.

The scaling differs between the products. MAC G3, with the default motor cycle of 1 ms and 8192 counts per revolution: V_SOLL = rpm x 2.1845, and one unit of A_SOLL is 457.8 rpm/s - for 1200 rpm write 2621 to register 5 (address 10). MIS G3: for 1200 rpm write 120000 to register 3 (address 6).

5.10.2 Position control

Position control uses the same registers with two additions. Poll the actual position (register 10), the actual velocity (register 12), the follow error (register 20) and the status (register 35 on MAC G3, register 25 on MIS G3). One poll from address 20 covers registers 10 to 20: quantity 22 on MAC G3, and 24 on MIS G3, where registers 10 and 20 are 64-bit.

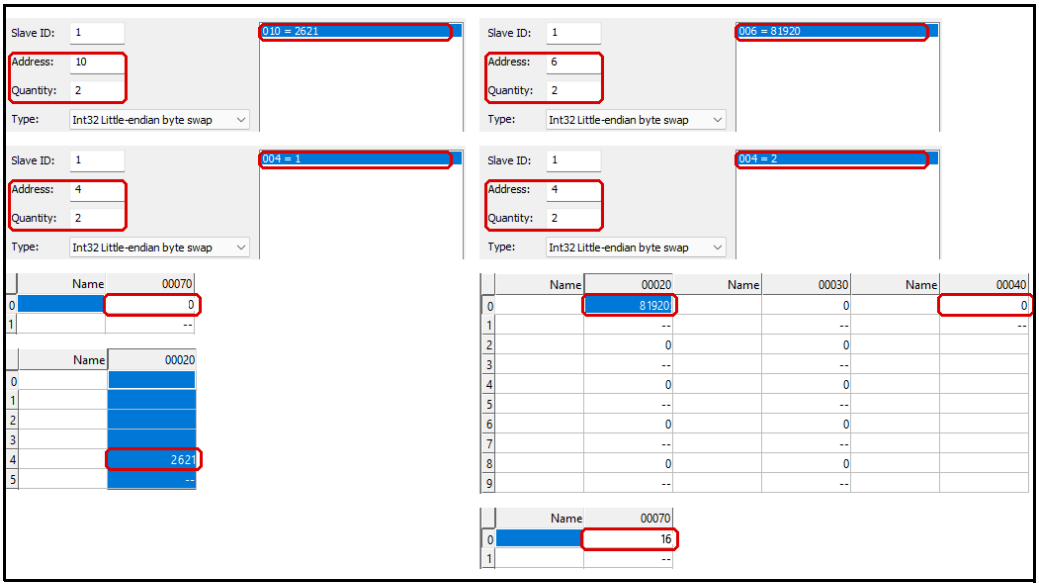
Write the ramp values as for velocity control, and the target position: register 3 (P_SOLL, 8192 counts per revolution) on MAC G3, register 4 (409600 counts per revolution, four Modbus registers from address 8) on MIS G3. Then write 2 to register 2 for position mode. The motor moves to the target and sets the in-position bit - bit 4 of register 35

on MAC G3, bit 4 of register 25 on MIS G3.

[EDITORIAL NOTE - NOT FOR RELEASE] ADO #2017 CLF

A new target position can be written while the motor is in position mode (register 3 on MAC G3, register 4 on MIS G3); the motor does not have to be taken back to passive first.

The write and poll frames of the two examples in a Modbus master, velocity control on the left and position control on the right (target position 81920 = 10 revolutions). The polls show the motor running at 1200 rpm and in position.



5.10.3 General considerations

Keep the status and error information on poll and monitor it in the master: register 35 on MAC G3, and on MIS G3 the status bits in register 25 and the error bits in registers 76 to 79. When an error occurs the motor stops, and the cause is shown there. The last diagnostic code is in register 560 (16435 in MacTalk) on MAC G3 and register 153 on MIS G3.

The motion status is in the same place. MAC G3 register 35: bit 4 in position, bit 5 accelerating, bit 6 decelerating, bit 18 homing done. MIS G3 register 25: bit 4 in position, bit 5 accelerating, bit 6 decelerating, bit 9 homing done.

Register 2 selects the mode: 0 is passive, where the shaft is de-energised, 1 velocity and 2 position. The other modes are described in the user manual of the motor.

5.10.4 Commands

Commands such as save to flash are written to a command register. On MAC G3 the motor commands go to motor register 211 (address 422) and the interface commands to interface register 524 (address 32798). On MIS G3 both kinds go to register 24 (address 48). A command is carried out every time it is written.

Table 53: Commands over Modbus

Command	MAC G3	MIS G3
Save to flash	524 = 272 (motor), 16	24 = 269
Save and restart	524 = 273	24 = 268
Restart	524 = 257	24 = 257
Clear errors	211 = 225	24 = 265
Relative move by P7	211 = 241, P7 in 61	24 = 113, P7 in 330

Parameters changed over Modbus are not saved automatically. Save them with the save command, otherwise they are lost at the next power cycle. On MAC G3, 272 saves the motor parameters and 16 the interface parameters.

5.10.5 Relative positioning

The motor must be in position mode. MAC G3: write the distance to register 61 (P7, address 122), then write 241 to register 211; the motor sets P_SOLL to the actual position plus P7. MIS G3: write the distance to register 330 (P7, four Modbus registers from address 660), then write 113 to register 24. In this release only positive distances are handled correctly on MIS G3. The move is finished when the in-position bit is set.

5.10 Examples

6 POWERLINK (beta)

Only MAC

6.1.1 Introduction to POWERLINK

[SCREENSHOT NEEDED - CK] Product photo: MAC G3 with POWERLINK, same angle and style as the other protocol photos in this manual.

Screenshot pending TT8806-10GB

ADO #2184 · Christian Kjærgaard (ck@jvl.dk)

LB0077 · section 6.1.1

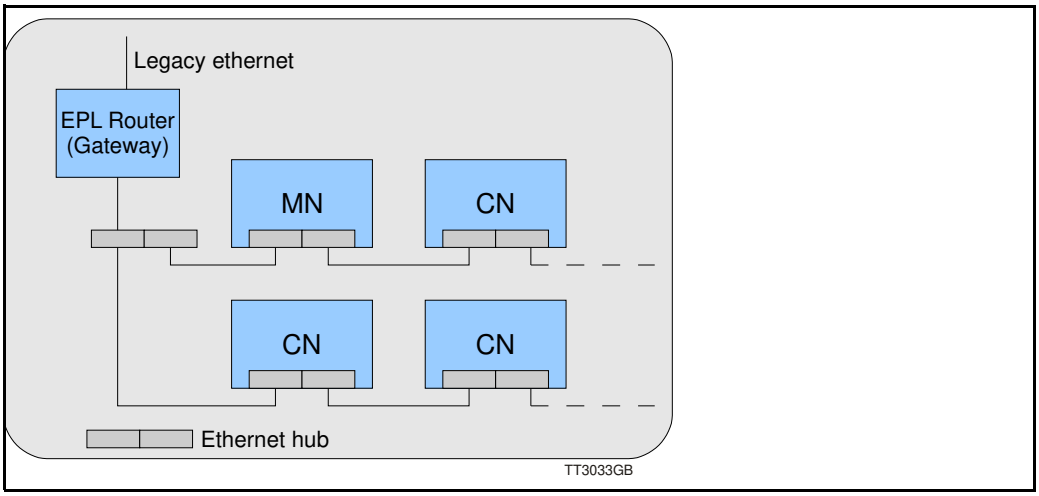
The screenshot must show:

Product photo: MAC G3 with POWERLINK, same angle and style as the other protocol photos in this manual. POWERLINK is MAC G3 only, so no MIS G3 photo

Placeholder - not for release. Replace this file with the screenshot under the same TT number; the manual then needs no change.

Ethernet POWERLINK® is a real-time Industrial Ethernet protocol, standardised and maintained by the Ethernet POWERLINK Standardization Group (EPSG). It uses standard Ethernet hardware and CAT5 cabling, and applies the same protocol technology as CANopen: it defines SDOs (Service Data Objects), PDOs (Process Data Objects) and the Object Dictionary structure to manage parameters.
In this release POWERLINK is available on MAC G3 only; support for MIS G3 will follow in a later release. On MAC G3 POWERLINK runs in the Ethernet interface of the motor, which passes the data on to the motor controller.

POWERLINK has no standardised drive profile (unlike PROFIdrive on PROFINET® or CiA 402 on EtherCAT®); the application layer uses the CANopen DS 301 communication profile - the same Object Dictionary / SDO / EMCY mechanics that EtherCAT's CoE (CANopen over EtherCAT) uses.



Unlike standard Ethernet, the Slot Communication Network Management (SCNM) ensures that only one node accesses the network at a time. The Managing Node (MN) grants access to the physical medium via dedicated poll request messages, so only one

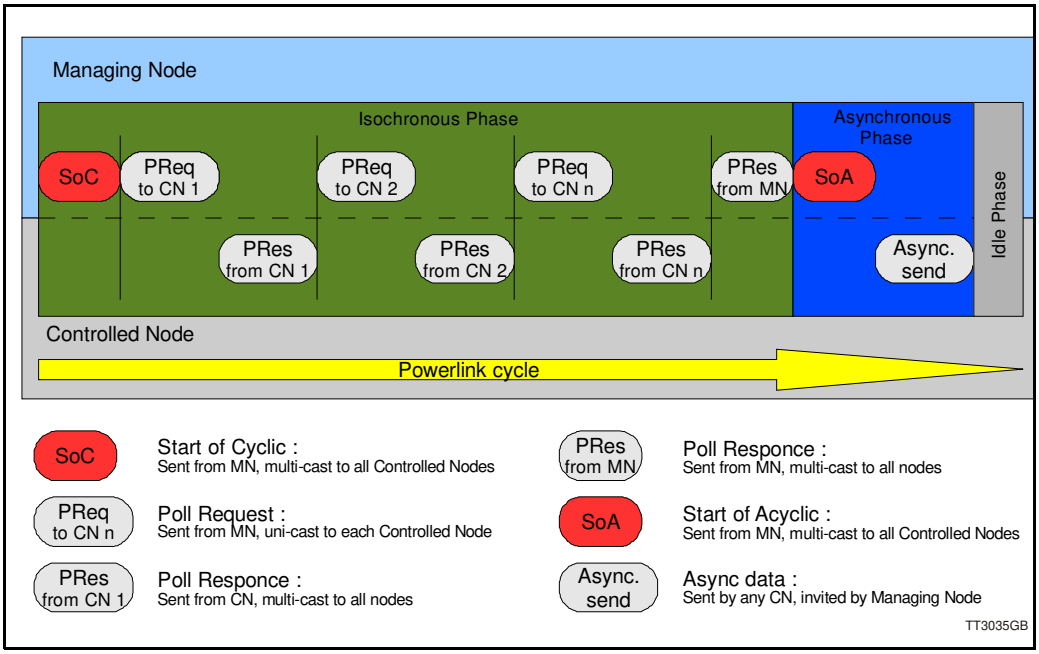
Controlled Node (CN) has access to the network at a time and no collisions occur. On generation III motors the Industrial Ethernet interface is built into the motor and cannot be removed. The protocol is selected via the Interface option in the Product Part Number Builder (PPNB) when the motor is ordered. The two Ethernet ports are used to daisy chain the motors: in on LA-I (XG1), out on LA-O (XG2). Both ports are M12 4-pin D-coded.

6.1.2 Abbreviations

Table 54: Abbreviations

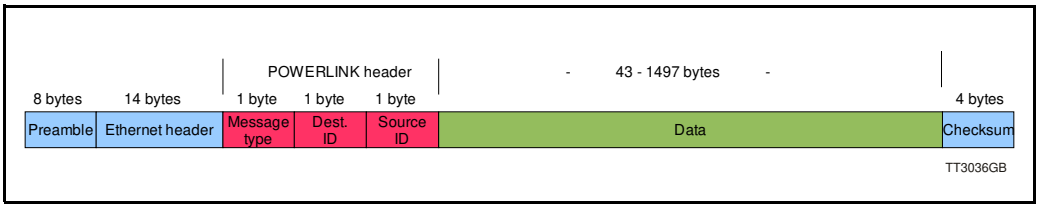
Term	Meaning
CN	Controlled Node (slave on the POWERLINK network)
EMCY	Emergency Object (CANopen error-reporting mechanism, part of DS 301)
EN	Error New (flag in the PRes frame, signals a new error to the MN)
MN	Managing Node (master on the POWERLINK network)
NMT	Network Management
PDO	Process Data Object (for cyclic data)
SDO	Service Data Object (for acyclic data)
SoC / SoA	Start of Cyclic / Start of Asynchronous (POWERLINK frame types)
PReq / PRes	Poll Request / Poll Response
SCNM	Slot Communication Network Management
XDD	XML Device Description - the file format that describes the device to the MN (POWERLINK's counterpart to CANopen's EDS)
DS 301	CANopen application-layer communication profile, reused by POWERLINK

6.2.1 Ethernet POWERLINK communication



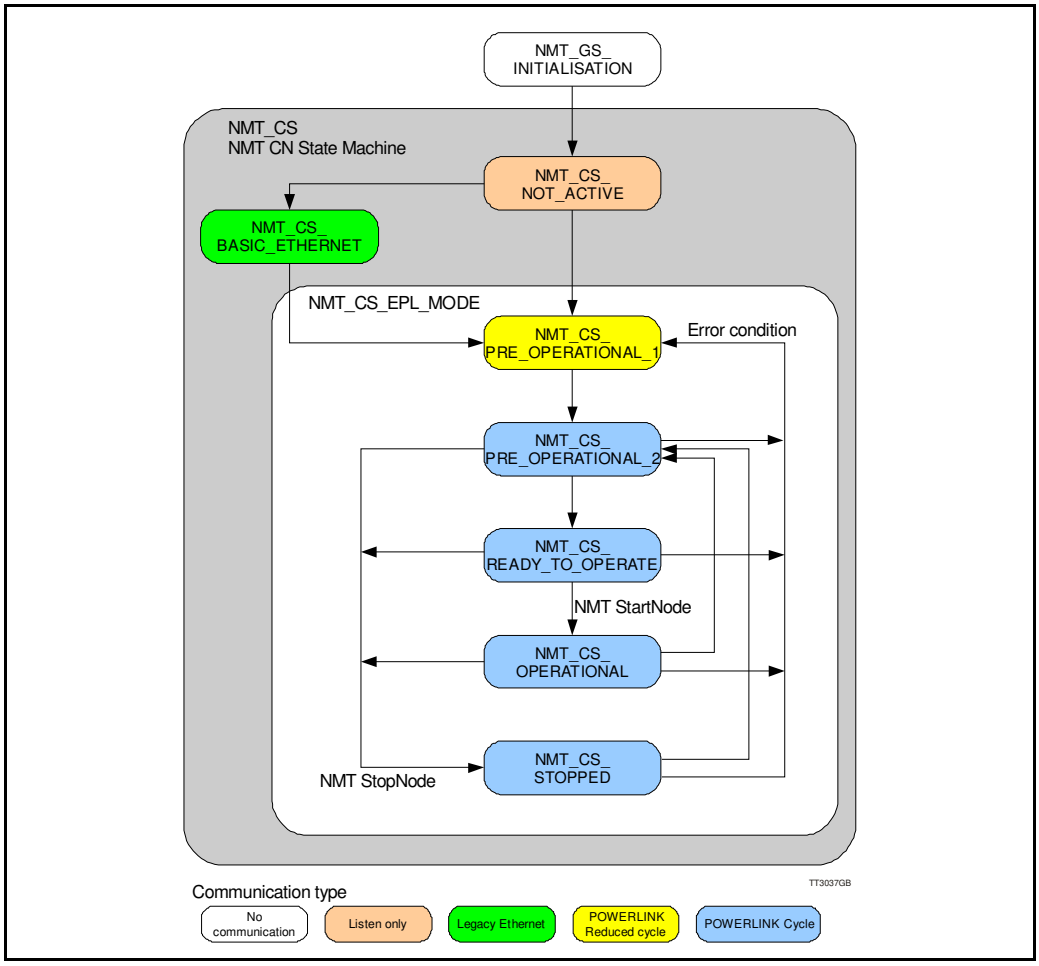
One cycle has an isochronous phase - the MN sends a Poll Request (PReq) to each CN in turn, and each CN replies immediately with a Poll Response (PRes) - an asynchronous phase (non-time-critical traffic, e.g. SDO or IP), and an idle phase.

6.2.2 Ethernet POWERLINK frame structure



POWERLINK messages are encapsulated in standard Ethernet II frames. Frame length is restricted to the configured cycle size; minimum 64 bytes, maximum 1518 bytes (exclusive of preamble). The POWERLINK header itself is only 3 bytes (message type, destination ID, source ID), leaving up to 1497 bytes of payload.

6.2.3 Ethernet POWERLINK CN State machine



All CN states are sub-states of NMT_GS_POWERED. After initialisation, the motor (a CN) enters the CN state machine in the NMT_GS_COMMUNICATING super-state:

- NMT_CS_NOT_ACTIVE - a temporary state that lets the node recognise the current network state. If no SoC, PReq, PRes or SoA frames are seen within a timeout, the node falls back to NMT_CS_BASIC_ETHERNET.
- NMT_CS_PRE_OPERATIONAL_1 - the first state within valid POWERLINK communication. The node moves on to NMT_CS_PRE_OPERATIONAL_2 once it detects a SoC frame.
- NMT_CS_PRE_OPERATIONAL_2 - entered once SoC frames are received cyclically. A communication error, for example a lost SoC or PReq, drops the node back to NMT_CS_PRE_OPERATIONAL_1.
- NMT_CS_READY_TO_OPERATE - the node confirms to the MN that it is ready, and waits for the NMT_StartNode command.
- NMT_CS_OPERATIONAL - the normal operating state. The node is configured and cyclic I/O data exchange starts.
- NMT_CS_STOPPED - entered from any active state on NMT_StopNode, for a controlled shutdown. The node keeps observing SoA frames but stops exchanging cyclic data.

6.2.4 Application layer communication

The application layer is the CANopen DS 301 communication profile - the same Object Dictionary / SDO / PDO / EMCY mechanics EtherCAT's CoE (CANopen over EtherCAT) uses. POWERLINK itself defines no drive profile: there are no 0x60xx CiA 402 objects. Message types used:

- Process Data Object (PDO) - cyclic I/O communication.
- Service Data Object (SDO) - slower, acyclic parameter access.
- NMT response services - identity and status signalling during start-up and at runtime.

6.3 Status indicators

Only MAC

The green RUN and the red ERR indicator are controlled by the POWERLINK stack. The two port indicators report link and activity, as on the other protocols.

6.3.1 RUN indicator

RUN follows the NMT state of the controlled node, as shown in figure TT8827-10GB.

Communication LEDs - POWERLINK states

TT8827-10GB

On the other protocols the same LEDs are used differently - see the protocol chapter.

NMT state	RUN / L1 (green)	Pattern
NMT_CS_NOT_ACTIVE	off	
NMT_CS_PRE_OPERATIONAL_1	single flash	
NMT_CS_PRE_OPERATIONAL_2	double flash	
NMT_CS_READY_TO_OPERATE	triple flash	
NMT_CS_OPERATIONAL	constant on	
NMT_CS_BASIC_ETHERNET	flickering	
NMT_CS_STOPPED	blinking	

The green RUN / L1 indicator follows the POWERLINK NMT state of the controlled node, and it is the quickest way to see how far the managing node has taken the motor without opening the engineering tool.

Blinking: 200 ms on and 200 ms off (2.5 Hz). Single flash: 200 ms on, then 1 s off. Double flash: two 200 ms flashes, then 1 s off. Triple flash: three 200 ms flashes, then 1 s off. Flickering: approximately 50 ms on and off (10 Hz).

6.3.2 ERR indicator

ERR is off when there is no error and lit solid red when an error is detected.

6.4.1 Mechanical installation

Connect the network cable(s) to the two Ethernet ports of the motor, L/A IN and L/A OUT. In a line topology, the next node is connected to the second port; in a redundant ring, the second port of the last node connects back to the MN. Standard CAT5 FTP or STP cables are recommended; UTP is not recommended in noisy industrial environments.

6.4.2 Node ID

The POWERLINK node ID is the last number of the motor's POWERLINK network address, the IP address in register 512 (16387 in MacTalk), and the motor reads it every time it starts. Valid node IDs are 1-239; if the last number is 0 or above 239, the motor uses node ID 1. With POWERLINK the first three numbers of the IP address are always 192.168.100: at start-up the motor sets register 512 to 192.168.100.<node ID>, and that is the address the motor has on the network. Out of the box the node ID is the last number of the MAC address on the motor label, or 1 if that number is 0 or above 239. Set the node ID by setting the IP address in MacTalk, see IP configuration in Common setup and functions, then save it in the motor and restart the motor.

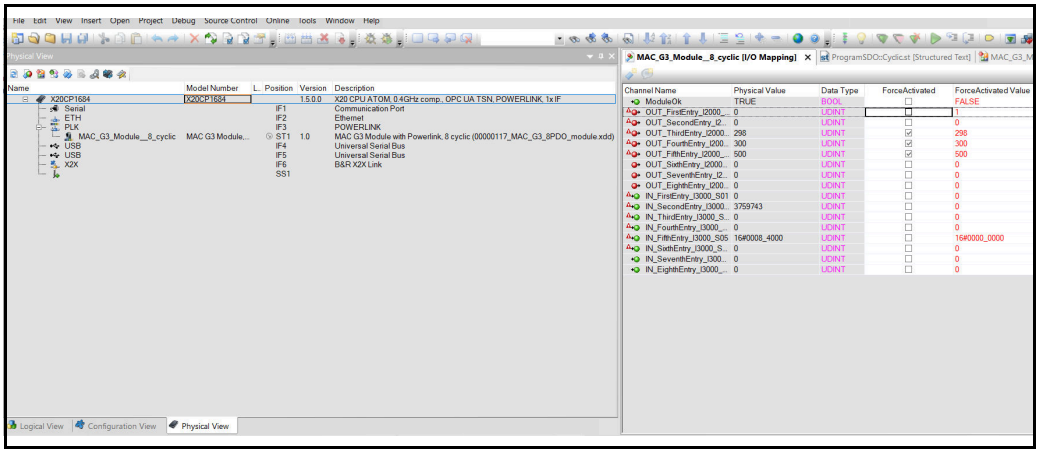
In addition to this, the motor has a second, separate address for MacTalk and Modbus UDP (register 553, 16428 in MacTalk on MAC G3). It is used only by MacTalk and Modbus UDP, not by POWERLINK itself; Modbus TCP is not available with POWERLINK. It is set on the MacTalk TCP tab, see IP configuration in Common setup and functions.

6.4.3 Quick start with Automation Studio

This example connects a motor to a B&R PLC with B&R Automation Studio and runs it in velocity mode with the default cyclic registers. The dialog names can differ between versions of Automation Studio; see the B&R documentation for details.

- Set the node ID: set the last number of the IP address in MacTalk, save it in the motor and restart the motor, see Node ID.
- Connect the POWERLINK interface of the PLC to one of the two Ethernet ports of the motor, L/A IN or L/A OUT, and apply power.
- In the Physical View, open the POWERLINK configuration of the PLC interface and make sure that POWERLINK communication is activated.
- Import the XDD file 00000117_MAC_G3_8PDO_module.xdd with Tools, Import fieldbus device. If Automation Studio asks whether to continue with a file that is not DS301 v1.3.0 conformant, continue.
- Insert the motor on the POWERLINK interface and enter the node ID that is set in the motor.
- Build the project and transfer it to the PLC. Use a full transfer: a change of the cyclic data or of the I/O mapping is not taken over by an online change.
- Open the I/O mapping of the motor and switch on the monitor. The write channels OUT_FirstEntry to OUT_EighthEntry are slots 1-8 of the write list, and the read channels IN_FirstEntry to IN_EighthEntry slots 1-8 of the read list. All channels are UDINT.

The figure shows the B&R Automation Studio I/O Mapping view with the cyclic register values.



- Force the write channels to the values below, the mode last. The motor then runs at 1000 rpm. Force OUT_FirstEntry to 0 to make the motor passive again.

Table 55: Quick start values, POWERLINK

Channel	Meaning	Register: value
OUT_FifthEntry	Torque limit	7: 85
OUT_FourthEntry	Acceleration	6: 44
OUT_ThirdEntry	Velocity	5: 2185
OUT_FirstEntry	Mode, velocity	2: 1

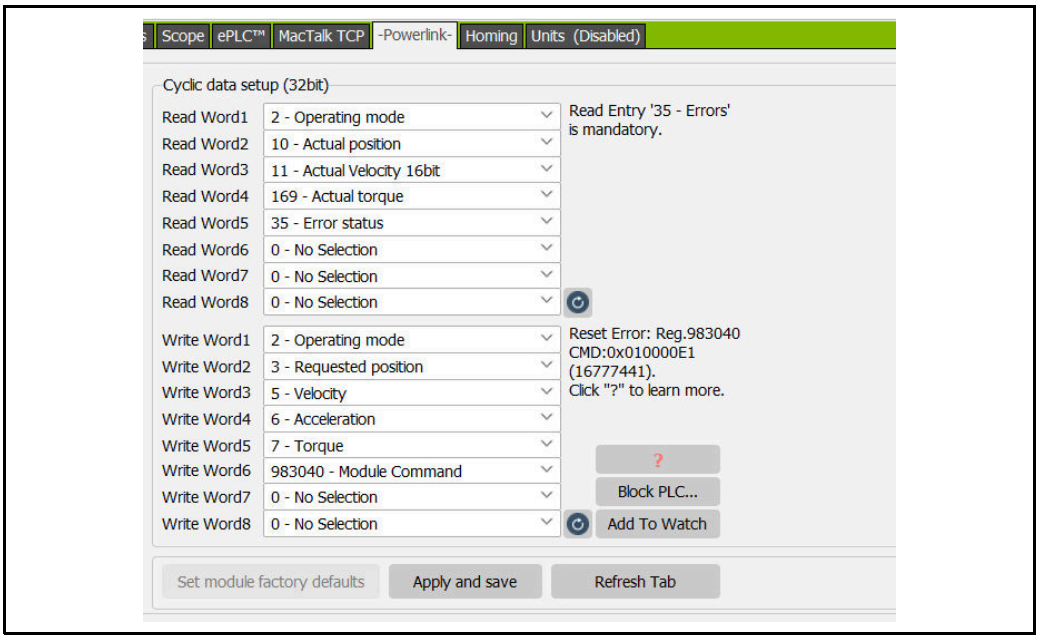
The values are given as register: value. They give 1000 rpm, an acceleration of about 20000 rpm/s and a torque limit of 25 %; see Examples for the scaling. IN_ThirdEntry shows the actual velocity.

6.5.1 Process data objects

PDOs carry the cyclic data: the receive PDO carries the data from the MN to the motor, and the transmit PDO carries the data from the motor to the MN. Each is 32 bytes - eight 32-bit registers - and the size is fixed. The eight write entries are object 0x2000, sub-indexes 1-8 (OUT_FirstEntry to OUT_EighthEntry), and the eight read entries object 0x3000, sub-indexes 1-8 (IN_FirstEntry to IN_EighthEntry). All entries are UNSIGNED32, and values are sent least significant byte first.
Out of the box the cyclic registers are the ones below, with the MAC G3 register numbers; see Register numbering in Common setup and functions. Slot 1 is OUT_FirstEntry or IN_FirstEntry, slot 2 the second entry, and so on.

Table 56: Default cyclic registers, POWERLINK

Slot	Read (motor to MN)	Write (MN to motor)
1	2 mode	2 mode
2	10 position	3 P_SOLL
3	11 V_IST_16	5 V_SOLL
4	169 torque	6 A_SOLL
5	35 ERR_STAT	7 T_SOLL
6	-	983040 command
7	-	-
8	-	-



The registers in the slots are selected in the motor, not by remapping the PDOs, see Selecting the cyclic registers.

6.5.2 Selecting the cyclic registers

Which register goes in each slot is selected in the motor, not in the MN: in MacTalk or by writing the interface registers directly. The read list (motor to MN) is registers 525-532 and the write list (MN to motor) is registers 533-540 (16400-16407 and 16408-16415 in MacTalk). A slot set to 0 ends the list - the slots after it are not used. In MacTalk, open the Powerlink tab shown above, choose the register for each slot under Cyclic data setup and press Apply and save; the new selection is taken into use as described below. From the MN the lists are written by SDO to object 0x2011: registers 525-532 are sub-indexes 16-23, and registers 533-540 sub-indexes 24-31. The value written is the register number.

A new selection is taken into use when the motor starts, or at once with the re-initialise command 21 or the save command 16 to object 0x2010. The selection is kept in RAM until it is saved, see Saving parameters to flash.

The PDO mapping objects 0x1600 and 0x1A00 are fixed. They always map the eight entries of object 0x2000 and 0x3000, and writing them does not change which registers are exchanged.

6.5.3 Rules for the cyclic registers

- Registers 35, 36 and 211 cannot be written cyclically, and register 35 is always in the read list: if it is missing, the motor adds it in the next free slot, or, if the list is full or holds interface registers, puts it in place of the last motor register in the list. An SDO write of 35, 36 or 211 to the write list (object 0x2011, sub-indexes 24-31) is refused with abort code 0x06040041.
- An interface register is entered as (register - 509) x 65536 and must come after all motor registers in the list. The interface command register 524 is therefore entered as 983040.
- A register can appear only once in each list. A command written cyclically is executed when the value changes to a new value other than 0; to repeat the same command, write 0 for at least one cycle first.
- If the selection breaks one of these rules, no registers are exchanged in either direction until it has been corrected and taken into use again, and the motor reports an error: bits 29 and 24 of register 35 are set. The node still goes to NMT_CS_OPERATIONAL.

6.5.4 Minimum cycle time

The cycle time is set in the MN, which writes it to object 0x1006 (NMT_CycleLen_U32, in microseconds) during start-up. The XDD file declares a default of 1000 us, a minimum of 400 us and a maximum of 1000000 us (1 s). The motor exchanges the cyclic registers with the Ethernet interface once per motor cycle (1 ms by default), so a network cycle below 1 ms does not update the motor data faster.

6.5.5 Service Data Objects

Service Data Objects (SDOs) are used for acyclic data - identification, configuration, and parameter access.

6.5.6 Object Dictionary

Each object in the Object Dictionary is addressed with a 16-bit index and a sub-index, and is read and written with SDO services. The standard communication objects (0x1000-0x1FFF) follow the POWERLINK communication profile, DS 301. The most important ones are listed below; the XDD file lists them all.

Table 57: Standard objects, POWERLINK

Index/sub	Name	Value
0x1000	Device type	0x000F0000, no drive profile
0x1001	Error register	Read only
0x1003	Error history	Read only entries
0x1006	Cycle length (us)	1000, written by the MN
0x1018/1	Vendor ID	0x00000117 (JVL)
0x1018/2	Product code	0x4900
0x1018/3	Revision number	Firmware major x 65536 + minor
0x1018/4	Serial number	Read only
0x1600	Receive PDO mapping	Fixed, 0x2000 sub-index 1-8
0x1A00	Transmit PDO mapping	Fixed, 0x3000 sub-index 1-8
0x1F93/1	Node ID	From the IP address
0x2000	PDO data, MN to motor	8 x UNSIGNED32
0x3000	PDO data, motor to MN	8 x UNSIGNED32

6.5.7 Manufacturer specific objects

The manufacturer specific objects give access to the registers of the motor and of the Ethernet interface, and hold the command object. Objects 0x2010, 0x2011 and 0x2012 keep the numbers used by the previous generation; 0x2013 is new. Every entry is UNSIGNED32. Changed registers are not saved automatically, see Saving parameters to flash. A positive SDO response to a write means that the write was received, not that the register accepted the value - read the register back if the value matters.

Table 58: Manufacturer specific objects

Index	Sub	Access	Description
0x2010	0	WO	Command object
0x2011	1-254	mostly RW	Interface parameters
0x2012	1-254	mostly RW	Motor parameters, sub-index = register number, as in the previous generation
0x2013	1-253	mostly RW	Extended motor parameters, new in G3

6.5.8 Object 0x2010 - Subindex 0

Write only. A value written to object 0x2010 is executed as a command to the interface command register 524, with the same command numbers - for example 16 (save the interface registers), 21 (re-initialise the cyclic registers) or 272 (save the motor registers). Bit 24 has no effect here: motor commands are written to motor register 211 through object 0x2012, sub-index 211. The command register is described in the online register overview:
www.jvl.dk/register.html?register=MACmotor

6.5.9 Object 0x2011

Interface registers are mapped to this object. The sub-index is not the register number: sub-index N is register N + 509, so sub-index 3 is register 512. See Register numbering in Common setup and functions.

6.5.10 Object 0x2012

Motor registers, for acyclic view or change.

6.5.11 Object 0x2013

Extended motor registers - not present in the legacy MAC00-ELx Object Dictionary; added to reach the full G3 register range. Sub-index N is motor register N + 255, so the object covers register 256 and up.

6.5.12 Acyclic access to motor registers

On MAC G3 the Ethernet interface does not hold the motor registers itself. Objects 0x2012 and 0x2013, and motor commands sent with bit 24 set in the command register, are passed on to the motor over an internal serial link, one register at a time. Object 0x2011 (Ethernet interface registers) is answered directly.
This has consequences for the master:

- Each access to a motor register takes up to about 30 ms, and can take longer while MacTalk is connected to the motor. Do not poll motor registers by SDO for fast data; place them in the cyclic registers instead. The cyclic data is not affected.
- If the motor does not answer in time, a read returns 0 and a write is not carried out, but the SDO is still answered without an abort code. Read an important value back after writing it.
- Motor register 255 cannot be reached by SDO: object 0x2012 ends at register 254, and object 0x2013 starts at register 256.

6.6 NMT services

Only MAC

Ethernet POWERLINK Network Management (NMT) is node-oriented and follows a master/slave relationship. The device is administered as an NMT slave by the MN. POWERLINK defines five categories of NMT services: NMT State Command Services, NMT Managing Command Services (not supported by a CN), NMT Response Services, NMT Info Services (not supported), NMT Guard Services (not supported).
Via the NMT State Response service the CN signals its state to the MN. The IdentResponse service is used by the MN to identify configured but unrecognised CNs at start-up or after a loss of communication. The StatusResponse service is used by the MN to query the current status of a CN that is not communicating isochronously - used for error signalling at runtime: the EN (Error New) flag in the PRes frame is toggled to notify the MN, which then polls the CN for a StatusResponse carrying the error information.

6.7 XDD file

Only MAC

XML Device Description files (XDD) specify a device's properties to the POWERLINK Managing Node - the communication objects it supports, its identity, and its default configuration. This is POWERLINK's counterpart to a CANopen EDS file.
The XDD file has a new file name, since the legacy 00000117_MAC00-ELx.xdd is replaced:

- 00000117_MAC_G3_8PDO_module.xdd - MAC G3

Re-importing the XDD into the MN project is unavoidable when upgrading from the previous generation (new hardware platform, new Product Code) - JVL's Vendor ID 0x117 is unchanged. XDD files are available through your local JVL representative and www.jvl.dk.

6.8 Troubleshooting

Only MAC

6.8.1 Motor not found, or stuck in Pre-operational 1

The MN does not find the motor, or the node stays in NMT_CS_PRE_OPERATIONAL_1 and no cyclic data is exchanged: check that POWERLINK is the protocol installed in the motor (register 558, 16433 in MacTalk, see Finding the FW/HW version in Introduction), and that the node ID entered in the MN is the node ID set in the motor, see Node ID.

6.8.2 No registers exchanged in Operational

The node reaches NMT_CS_OPERATIONAL, but no registers are exchanged: check the selected cyclic registers against Rules for the cyclic registers. A selection that breaks the rules stops the exchange in both directions, and the motor reports an error. If the selection is correct and the motor still exchanges no data with the MN, contact your local JVL representative.

6.8.3 Error reporting

Errors in the motor are reported to the MN as entries in the error history (object 0x1003), which also reach the MN in the StatusResponse frame. An error without a specific POWERLINK error code is reported as 0xFF00 (manufacturer specific). When the error is cleared, a new entry reports it as cleared.

6.8.4 Motor does not move

The motor does not move although the velocity and the mode are written in the cyclic data: check that write slot 5, the torque limit (register 7), is not 0.

6.8.5 DS301 conformance warning on XDD import

B&R Automation Studio asks on the XDD import whether to continue with a file that is not DS301 v1.3.0 conformant: continue. The import is not affected.

6.8.6 PDO channel type mismatch in Automation Studio

Automation Studio "Type mismatch" connecting a PDO channel to an ST variable: all 16 PDO channels are declared UDINT (Unsigned32) in the XDD, not DINT. Declare PLC-side variables as UDINT; use UDINT_TO_DINT / DINT_TO_UDINT for signed math or bitmasking on position/velocity/status-word values.

6.8.7 Motor does not stop on lost connection

The motor does not stop when the connection to the MN is lost: when the node leaves NMT_CS_OPERATIONAL, for example because the MN stops or a cable is disconnected, the motor reacts as selected in bits 0-1 of register 515 (16390 in MacTalk) - passive mode, velocity 0 or no reaction, see Motor reaction on lost connection in Common set-up and functions. Check that the value is not 2 or 3, which means no reaction.

6.9 Examples

Only MAC

6.9.1 Running Velocity control

In velocity mode the motor runs at the requested velocity with the requested acceleration. With the default cyclic registers everything needed is already in the PDO: OUT_FirstEntry is the mode (register 2), OUT_ThirdEntry the velocity (register 5, V_SOLL), OUT_FourthEntry the acceleration (register 6, A_SOLL) and OUT_FifthEntry the torque (register 7). V_SOLL = rpm x 2.1845, and one unit of A_SOLL is 457.8 rpm/s; for 1200 rpm write 2621. Write the velocity, the acceleration and a torque other than 0 first, and then 1 (velocity mode) to OUT_FirstEntry. Write 0 to make the motor passive again.

6.9.2 Running Position control

Position mode needs 2 in the mode register. With the default cyclic registers OUT_SecondEntry is the requested position, register 3 (P_SOLL, 8192 counts per revolution). OUT_ThirdEntry and OUT_FourthEntry set the maximum velocity and the acceleration used to get there. IN_SecondEntry is the actual position, register 10. The move is finished when the in-position bit, bit 4 of register 35 (IN_FifthEntry by default), is set. Register 20 (Follow error, the actual position error) and register 16 (Motor load mean, the average motor load) are also useful to monitor in position mode; neither is in the default cyclic list. Read them occasionally by SDO (object 0x2012, sub-index 20 and 16), or map them into an unused cyclic slot if continuous monitoring is needed.

6.9.3 General considerations

Register 35 holds the status and error information and is always in the cyclic read list - the motor adds it if it is missing. If an error occurs the motor stops and the cause is left in register 35. 0 in register 2 makes the motor passive (no current in the motor), and a mode such as 1 (velocity) or 2 (position) starts it. Register 2 can be written cyclically or by SDO to object 0x2012, sub-index 2.

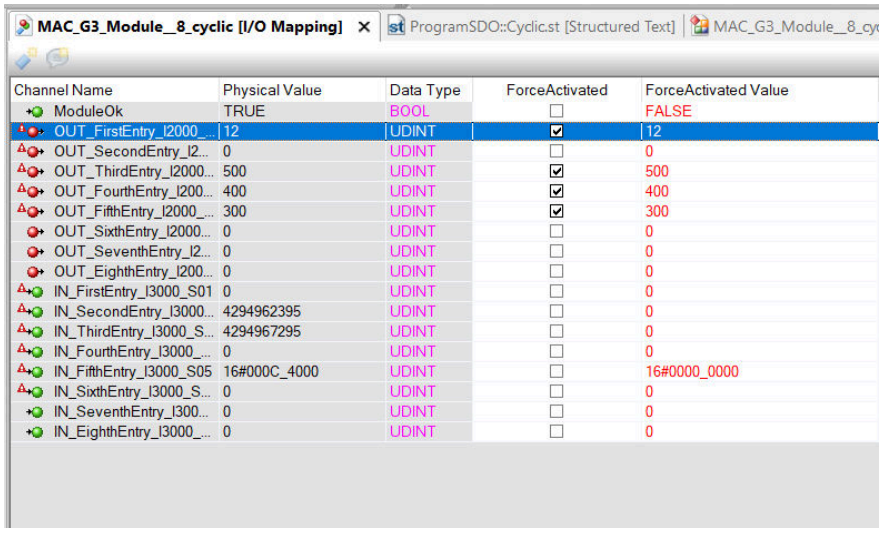
6.9.4 Homing using only cyclic I/O

Before requesting homing over cyclic I/O, set the home position (register 38), home velocity (register 40) and, for the torque-based homing mode, the home torque (register 41) in MacTalk and save them to the motor - they are read once at the start of the sequence, not updated while it runs. The motor start mode (register 37) should also be set to position and saved, so the motor stays in position once homing completes.

Set the "Requested_Mode" register to the appropriate homing mode value, then observe the motor stop and enter homing (cyclic writes are blocked to the motor during homing). On completion (register 35 bit 18 = homing done) change "Requested_Mode" to whatever is needed next; the motor then unblocks cyclic writes.

Homing can be aborted at any time by writing a different value to "Requested_Mode" - the motor detects the new value and applies it immediately, even while homing is still in progress. Register 35 and the rest of the cyclic read data keep updating throughout homing, so progress can be polled without any acyclic traffic.

The figure shows the B&R Automation Studio I/O Mapping view after torque homing (mode 12) has completed over cyclic I/O: OUT_FifthEntry (torque limit) is forced to 300, and IN_FifthEntry (register 35) reads 16#000C_4000 - bit 18 (homing done) is set.



6.9.5 Relative positioning

[EDITORIAL NOTE - NOT FOR RELEASE] ADO #2006 DJO

This moves the motor a fixed distance from wherever it currently is, without writing or knowing an absolute target position.
Place the command register (983040 in MacTalk) in the cyclic write list. Set register P7 to the requested relative offset, then issue command 0x010000F1 in the command register. Set the command register back to zero afterwards to prepare for the next command. Monitor register 35 bit 4 (in position) to know when the move is finished.

[EDITORIAL NOTE - NOT FOR RELEASE] ADO #2075 DJO

6.9.6 Saving parameters to flash

Registers written through the cyclic data or by SDO take effect at once, but they are kept in RAM and are lost at power-off unless they are saved. Use the save commands below. The same values can also be written to the interface command register (983040) in the cyclic write list. No restart is needed unless the command includes it.

Table 59: Save and clear commands, POWERLINK

Command	Object: value
Save motor registers	0x2010: 272
Save interface registers	0x2010: 16
Save all and restart	0x2010: 273
Clear errors	0x2012/211: 225

The values are given as object/sub-index: value. 0x2012/211 is motor register 211.

6.9.7 Clearing errors

Clear an error reported in register 35, or by the Ethernet interface itself, from the master - without a power cycle.
Write 16777441 (0x010000E1) to the interface command register in the cyclic write list - slot 6, OUT_SixthEntry, with the default registers - and write 0 afterwards to re-arm it. Without cyclic data, write 225 to motor register 211 by SDO (object 0x2012, sub-index 211). Both commands also clear the errors of the Ethernet interface. See Reset errors from the master in Examples common to all protocols.

7 PROFINET (beta)

Only PROFINET

7.1 Introduction

Only PROFINET

PROFINET® IO is an industrial Ethernet protocol in which the motor is an IO device controlled by an IO controller (PLC). A generation III motor with PROFINET supports cyclic register data and the PROFIdrive profile.

PROFINET is released as a beta version in this firmware.

7.1.1 Introduction to PROFINET IO

PROFINET IO is a fieldbus protocol that lets programmable controllers communicate with distributed field devices over Ethernet. It can be thought of as PROFIBUS on Ethernet: it uses ordinary Ethernet hardware and software to structure the exchange of data, alarms and diagnostics with controllers.

The protocol classifies devices as IO controllers, IO supervisors and IO devices, each with its own set of services. A generation III motor is an IO device.

PROFINET IO uses three communication channels. The standard UDP/IP and TCP/IP channel is used for parameterisation and configuration and for acyclic operations. The Real Time (RT) channel carries cyclic data and alarms. The Isochronous Real Time (IRT) channel is used in some motion control applications and is, in JVL motors, tied to the PROFIdrive profile.

PROFINET IO devices are structured in slots and sub-slots, which hold modules and submodules. Device specific data is in slot 0 and module data in the slots that follow. One of the benefits of PROFINET IO is its diagnostics and alarm mechanism: every module provides alarm data to the IO controller over the cyclic channel, and diagnostic data can be read acyclically as record data.

The properties and services of a device are described in a GSD file written in GSDML (General Station Description Markup Language) - see GSDML file.

On generation III motors the Industrial Ethernet interface is built into the motor and cannot be removed. The protocol is selected via the Interface option in the Product Part Number Builder (PPNB) when the motor is ordered. The two Ethernet ports are used to daisy chain the motors: in on LA-I (XG1) on MAC G3 or CN2 on MIS G3, out on LA-O (XG2) or CN3. All four ports are M12 4-pin D-coded.

The station name and the IP address of the protocol stack are normally assigned by the controller through the standard DCP mechanism, not set in the motor. See IP configuration in Common setup and functions.

There is no theoretical limit to the number of nodes in a PROFINET IO network; in practice the controller and the number of available addresses set the limit. The protocol is specified in IEC 61158 and IEC 61784. Further information is available from https://www.profinet.com.

7.1.2 Device structure

The motor uses slot 0 and slot 1. Slot 0 holds the Device Access Point (DAP), which represents the motor itself: the I&M data in subslot 1, the PROFINET interface in subslot 32768 and the two ports in subslots 32769 and 32770. Slot 1 holds one of two application modules:

- JVLdrive - the JVL register module. Its submodule, Cyclic data interface, carries eight 32-bit registers in each direction. Both products.
- PROFIdrive - MAC G3 only. The PROFIdrive access point (MAP) is in subslot 1, and standard telegram 1, 2 or 3 in subslot 2.

The motor offers cyclic data, acyclic access to its registers and to the PROFIdrive parameters through record data, diagnosis, and the Identification & Maintenance (I&M) functions. It supports one application relation (one IO controller) at a time.

7.1.3 Abbreviations

These terms are used in this chapter and in the register descriptions.

Table 60: PROFINET IO abbreviations

Term	Meaning
AC	Application Class
Acyclic	Communication where a message is sent once per request
Cyclic	Communication where process data is sent at predefined intervals
DAP	Device Access Point
DCP	Discovery and Configuration Protocol
DO	Drive Object
GSD	General Station Description, the device description file
GSDML	General Station Description Markup Language, the XML based language the GSD file is written in
I&M	Identification and Maintenance
IO controller	The control system with bus initiative, also called the master
IOPS / IOCS	IO Provider Status and IO Consumer Status
MAP	Module Access Point (PROFIdrive)
MRP	Media Redundancy Protocol
NSOLL / NIST	Speed set-point / actual speed (PROFIdrive)
PAP	Parameter Access Point
PNU	Parameter number (PROFIdrive)
RDREC/WRREC	Read record / write record (Siemens function blocks)
ST	Standard Telegram
STW / ZSW	Control word / status word (PROFIdrive)

7.2 Status indicators

Only PROFINET

PROFINET uses two indications, System Failure (SF) and Bus Failure (BF), and both of them are red. On the motor the red ERR indicator carries Bus Failure; the two port indicators LA1/L2 and LA2/L3 report link and activity.

What the indicator shows is given in figure TT8829-10GB. Bus Failure covers the connection to the IO controller: no configuration, no link, or a configured connection that has stopped exchanging data.

Communication LEDs - PROFINET status

TT8829-10GB

On the other protocols the same LEDs are used differently - see the protocol chapter.

State	Pattern
ERR (red) - Bus Failure (BF)	
No bus failure	
No data exchange	
No configuration, or no link	

Not shown on this motor

PROFINET uses two red indicators, System Failure and Bus Failure. This motor has one red indicator, so System Failure - a watchdog timeout or a pending diagnosis - has no indicator to appear on. The green RUN / L1 indicator is not used by PROFINET.

The flashing test of a device scan is a separate case: the stack passes the request to the motor firmware, which acknowledges it and lights nothing. Use the motor identification in MacTalk, which flashes PWR.

Flashing (2 Hz): 250 ms on, 250 ms off.

Source: Hilscher netX 90 PROFINET IO-Device (Stack_v5.7.0.1), and the PROFINET specification. The stack drives the indicators; the motor firmware does not.

The patterns are produced by the netX90 PROFINET stack (Stack_v5.7.0.1) and not by the motor firmware, and they follow the PROFINET specification. System Failure is the second red indication in that scheme, and since the motor has only one red indicator it has none of its own: a watchdog timeout and a pending diagnosis are not signalled on the motor, and the green RUN / L1 indicator is not used by PROFINET. The flashing test of a device scan is a separate case: the stack does not perform it itself but passes the request to the motor firmware, which acknowledges it and lights nothing. An engineering tool asking the motor to flash therefore receives a successful reply and no indicator changes. Use the motor identification in MacTalk instead, which flashes the PWR indicator.

7.3 Cyclic data

Only PROFINET

The motor exchanges eight 32-bit registers cyclically through the JVL register module, which is called JVLdrive in the GSDML file and which both products offer. The MAC G3 offers a PROFIdrive module as well, with standard telegrams 1, 2 and 3; the MIS G3 GSDML file does not contain that module, so on the MIS G3 the JVL register module is the only cyclic interface.

7.3.1 Selecting the cyclic registers

Which register goes in each of the eight slots is selected in the motor, so any motor or interface register can be exchanged cyclically. Cyclic read 1..8 (motor to controller) are registers 525-532, and cyclic write 1..8 (controller to motor) are registers 533-540, on both products (16400-16407 and 16408-16415 in MacTalk on MAC G3). In MacTalk the selection is made on the PROFINET tab, see figure TT8849-10GB. Remember to save it to flash.

[SCREENSHOT NEEDED - CLF] MacTalk PROFINET tab: the cyclic read and write register selection, Alt. byte order and the name of station setting.

Screenshot pending TT8849-10GB

ADO #1731 · Christian Lotz Felter (clf@jvl.dk)

LB0077 · section PROFINET - Selecting the cyclic registers

The screenshot must show:

MacTalk PROFINET tab of a MAC G3: the cyclic read and write register selection, the Alt. byte order option and the name of station setting

Placeholder - not for release. Replace this file with the screenshot under the same TT number; the manual then needs no change.

The motor takes the selection into use every time the controller establishes the connection. A change made while the connection is running is taken into use with the command that re-initialises the cyclic data (21 on MAC G3, 300 on MIS G3), see Commands. The selection can also be written by the controller itself, see Changing the cyclic setup in Examples.

7.3.2 Default cyclic registers

Out of the box the cyclic registers are the ones below. Register numbers are the product's own; see Register numbering in Common setup and functions.

Table 61: Default cyclic registers, JVLdrive module

Slot	MAC G3 read	MAC G3 write	MIS G3 read	MIS G3 write
1	2 mode	2 mode	9 actual mode	2 mode
2	10 position	3 P_SOLL	10 position	4 position
3	11 velocity	5 V_SOLL	12 velocity	3 velocity
4	169 torque	6 A_SOLL	13 torque	6 acceleration
5	35 ERR_STAT	7 T_SOLL	153 diagnosis	7 torque
6	-	983040 (524)	-	24 command
7	-	-	-	-
8	-	-	-	-

The MAC G3 write list contains the torque limit T_SOLL (register 7), and the MIS G3 write list the torque (register 7). Write a sensible value there as well - with 0 the motor has no torque. 341 is 100 % on MAC G3, and 1000 is 100 % on MIS G3.

7.3.3 Rules for the cyclic registers

- MAC G3: registers 35, 36 and 211 cannot be written cyclically, and register 35 is always in the read list. If it is missing, the motor puts it in the next free slot. If there is no free slot, or if the list contains interface registers, the motor puts it in place of the last motor register. Send motor commands and clear errors through the interface command register instead, see Examples.
- MAC G3: an interface register (512 and up) is entered as (register - 509) x 65536 and must come after all motor registers in the list. The interface command register 524 is therefore entered as 983040.
- MIS G3: read-only registers cannot be written cyclically.
- Both: a register can appear only once in each list, and the list ends at the first empty slot (0). A command written cyclically is executed when the value changes to a new value other than 0; to repeat the same command, write 0 for at least one cycle first.
- MIS G3: registers 4, 10, 20 and 330 (P7) are 64-bit. To read one cyclically, put it in the read list followed by the next register number, for example 20 and 21. The two words then carry the upper 32 bits first and the lower 32 bits second. If the slot after a 64-bit register holds another register, that register is sent in place of the upper 32 bits, and only the lower 32 bits of the 64-bit register are read.

The data of a register is transferred most significant byte first, unless the byte order is swapped, see Byte order.

[EDITORIAL NOTE - NOT FOR RELEASE] ADO #2015 DJO

[EDITORIAL NOTE - NOT FOR RELEASE] ADO #2013 DJO

7.4 Acyclic data

Only PROFINET

Besides the cyclic data, the IO controller can read and write single registers while the connection is running. It uses the PROFINET record services, which are called RDREC and WRREC in a Siemens controller. The index in the request says what is addressed.

Table 62: Record index map, MAC G3

Index	What it addresses
0x0000 - 0x0001	User area, reserved for the PROFINET Tester
0x1000	Command register of the Ethernet interface, write only
0x1100 - 0x11FF	Registers of the Ethernet interface, 0 to 255
0x1200 - 0x12FF	Motor registers 0 to 255
0x1300 - 0x13FF	Motor registers 256 to 511
0xB02E	PROFIdrive Base Mode Parameter Access, local
0xB02F	PROFIdrive Base Mode Parameter Access, global. Not implemented

In the three register windows the low byte of the index is the register number, so index 0x1223 is motor register 35 and index 0x1300 is motor register 256. One record is one 32-bit register. The byte order follows the setting described under Byte order.
The interface window counts from the first interface register, not from the shared list: index 0x1100 is register 509 and index 0x1101 is register 510. The two motor windows need no such conversion. The three windows are the MAC G3 arrangement; on the MIS G3 there are no windows at all, and the index is simply the register number over the whole list. See Register numbering in Common setup and functions.
The MIS G3 does not use the three windows. Its registers are reached directly in the range 0x0002 to 0x7FFF.
A write the motor cannot carry out is answered with a PNIO status and not with success:

Table 63: PNIO status of a refused write

The motor could not carry out the write because	PNIO status
the register is read only	PNIO_E_IOD_WRITE_ACCESS_DENIED
the value is outside the range of the register	PNIO_E_IOD_WRITE_ACCESS_INVALIDRANGE
there is no such register	PNIO_E_IOD_WRITE_ACCESS_INVALIDINDEX
the write did not reach the motor	PNIO_E_IOD_WRITE_APPL_BUSY

A read of an index the motor does not serve is answered PNIO_E_IOD_READ_ACCESS_INVALIDINDEX. A master may ask for any index, so this is an ordinary refusal and not a fault in the motor.

7.4.1 Data length

A register is read and written as 4 bytes. On MIS G3 the 64-bit registers, for example the positions in registers 4 and 10, are read and written as 8 bytes. On MIS G3, an 8-byte access to a 32-bit register is refused with an INVALIDRANGE status. On MAC G3, always use 4 bytes. The value is sent most significant byte first, unless the byte order is swapped, see Byte order - the setting applies to the record data as well as to the cyclic data.

7.4.2 Commands

Commands such as save to flash are written as a value to a command register. On MAC G3 it is index 0x1000, which writes the interface command register 524; the status of the last command can be read in register 523 (index 0x110E). On MIS G3 it is the motor command register 24 (index 0x0018).

Table 64: Commands over PROFINET

Command	MAC G3: index 0x1000	MIS G3: register 24
Save to flash	16 interface, 272 motor	269
Save to flash and restart	273 (both)	268
Restart	1 interface, 257 both	257
Factory default	18	275
Factory default, keep IP	24	277
Apply the cyclic setup	21	300
Clear errors	225 to register 211	265 (97 incl. warnings)

On MAC G3 a command written to index 0x1000 is always an interface command. The forwarding of a motor command with bit 24 set (for example 0x010000E1) only works from the cyclic data; acyclically, a motor command is written to motor register 211 (index 0x12D3) instead.

Parameters changed through record data are not saved automatically. Save them with the save command, otherwise they are lost at the next power cycle.

7.4.3 Record access from TIA Portal

In a Siemens controller the record services are the function blocks RDREC (read) and WRREC (write). The motor serves the register indices on any of its submodules, so the ID can be the hardware identifier of the DAP or of the cyclic data submodule.

Table 65: RDREC and WRREC parameters

Parameter	Value
REQ	A rising edge starts the request
ID	Hardware identifier of a submodule of the motor
INDEX	The record index, see the table above
MLEN / LEN	4 bytes, or 8 for a 64-bit MIS G3 register
RECORD	The value, MSB first (see Byte order)

The table "Record index examples" lists some of the indices used most often. Remember that the two products number their registers differently.

Table 66: Record index examples

What is addressed	MAC G3 index	MIS G3 index
Command register	0x1000	0x0018 (24)
Mode, register 2	0x1202	0x0002
Error and status bits	0x1223 (35)	0x0019 (25)
Actual position, register 10	0x120A	0x000A, 8 bytes
P7, distance for relative moves	0x123D (61)	0x014A (330)
Motor command register	0x12D3 (211)	-
Cyclic read 1-8 (525-532)	0x1110-0x1117	0x020D-0x0214
Cyclic write 1-8 (533-540)	0x1118-0x111F	0x0215-0x021C
IP address (512)	0x1103	0x0200
Setup bits	0x1106 (515)	0x007B (123)

Example: to save the motor registers of a MAC G3, write the value 272 with WRREC to index 16#1000. To read the error and status register of a MAC G3, read 4 bytes with RDREC from index 16#1223. See figure TT8850-10GB.

[SCREENSHOT NEEDED - CLF] TIA Portal program with WRREC writing 272 to index 16#1000 and RDREC reading index 16#1223 of a MAC G3.

TT8850-10GB

Screenshot pending

ADO #1731 · Christian Lotz Felter (clf@jvl.dk)

LB0077 · section PROFINET - Record access from TIA Portal

The screenshot must show:

TIA Portal program with WRREC writing 272 to index 16#1000 and RDREC reading index 16#1223 (register 35) of a MAC G3, with the parameters filled in

Placeholder - not for release. Replace this file with the screenshot under the same TT number; the manual then needs no change.

7.5 PROFIdrive

Only PROFINET

On the MAC G3, PROFIdrive is supported for application class 1 (AC1) in real time (RT) with standard telegrams 1, 2 and 3, and for application class 4 (AC4) in isochronous real time (IRT) with standard telegram 3. The MIS G3 GSDML file offers no PROFIdrive module, so these telegrams cannot be selected for a MIS G3 in the engineering tool.

[EDITORIAL NOTE - NOT FOR RELEASE] ADO #2183 CLF

7.5.1 Standard telegrams

The PROFIdrive module is placed in slot 1 instead of the JVLdrive module, with the telegram in subslot 2. The signals of the three telegrams are listed below. ST1 is 4 bytes in each direction, ST2 8 bytes, and ST3 10 bytes to the motor and 18 bytes from it.

Table 67: PROFIdrive standard telegrams (MAC G3)

Signal	Bits	Direction	Content
STW1	16	to motor	Control word 1 (ST1, ST2, ST3)
NSOLL_A	16	to motor	Speed set-point (ST1)
NSOLL_B	32	to motor	Speed set-point (ST2, ST3)
STW2	16	to motor	Control word 2, sign of life (ST2, ST3)
G1_STW	16	to motor	Encoder control word (ST3)
ZSW1	16	to PLC	Status word 1 (ST1, ST2, ST3)
NIST_A	16	to PLC	Actual speed (ST1)
NIST_B	32	to PLC	Actual speed (ST2, ST3)
ZSW2	16	to PLC	Status word 2, sign of life (ST2, ST3)
G1_ZSW	16	to PLC	Encoder status word (ST3)
G1_XIST1	32	to PLC	Actual position in encoder counts (ST3)
G1_XIST2	32	to PLC	Same value as G1_XIST1 (ST3)

The motor runs in velocity mode when the controller enables operation through STW1. The speed is scaled to the reference speed, which is 3000 rpm on the MAC G3 and can be read in parameter P60000: 0x4000 in NSOLL_A and NIST_A, and 0x40000000 in NSOLL_B and NIST_B, is the reference speed. 1500 rpm is therefore 0x2000 in NSOLL_A. G1_XIST1 is the actual position (register 10) in encoder counts. A fault is acknowledged with bit 7 of STW1.

A telegram uses the cyclic register selection of the Ethernet interface internally, and selecting it overwrites that selection. ST1 and ST2 set registers 525-526 and 533-534, ST3 sets registers 525-527 and 533-534, and the other slots are cleared. Set the JVL selection again before going back to the JVLdrive module, and do not save the interface registers while a telegram is in use if the JVL selection is to be kept.

7.5.2 Changing the configuration

The motor sets up its cyclic configuration - JVLdrive or a PROFIdrive telegram - the first time the controller connects after the motor has started. It cannot change to a different configuration while it is running. If the controller later requests a different configuration, the motor reports the warning CONFIGURATION NOT APPLIED, and the cyclic data must not be used for control until the motor has been restarted. Restart or power cycle the motor to take the new configuration into use. Changing the project back in the controller does not clear the warning.

7.5.3 PROFIdrive parameters

The PROFIdrive parameters are read and written with the Base Mode Parameter Access, record index 0xB02E, on the PROFIdrive module and also on the JVLdrive module. The global parameter access (index 0xB02F) is not implemented. The motor serves these parameters:

Table 68: PROFIdrive parameters

PNU	Name	Access
922	Telegram selection	ro
924	Pulses enabled (ZSW2 bit 11)	ro
925	Sign-of-life failures tolerated (0-256)	rw
930	Operating mode	ro
944	Fault message counter	ro
945-949	Fault code, list, number, time and value	ro
950	Fault buffer scaling (8 situations x 8)	ro
951	Fault number list, with texts	ro
952	Fault situation counter (0 clears)	rw
964	Drive unit identification	ro
965	Profile identification	ro
970	Load parameter set	rw
971	Save to non-volatile memory	rw
972	Drive reset	rw
974	Base mode parameter access identification	ro
975	Drive object identification	ro
977	Save drive unit to non-volatile memory	rw
979	Sensor format	ro
980	List of parameters	ro
60000	Velocity reference value, rpm	ro

P970, P971, P972 and P977 are commands: write 1 to carry out the action, and the parameter reads 0 again when the command has been queued. P970 = 1 loads the factory setting, and P970 = 2 loads it but keeps the IP settings. P971 and P977 save to flash; on MAC G3 they save the interface registers (command 16), and on MIS G3 all registers. P972 restarts the Ethernet interface on MAC G3, and the motor on MIS G3.

[EDITORIAL NOTE - NOT FOR RELEASE] ADO #2182 CLF

An error or a warning in the motor is reported to the IO controller as a channel diagnosis. The ChannelErrorType is the PROFIdrive fault class, and the ExtChannelErrorType is the JVL code that says which condition inside that class it is. With the PROFIdrive module (MAC G3) the diagnosis is sent on API 14848, which is the PROFIdrive API, with the fault classes as they are listed below.

Table 69: PROFIdrive fault classes the motor uses

ErrorType	PROFIdrive fault class	MAC G3	MIS G3
0x9000	01 Microcontroller hardware or software	75	81
0x9001	02 Mains supply	3	1
0x9002	03 Low voltage supply	1	1
0x9005	06 Overtemperature, electronic device	1	-
0x9007	08 Motor overload	4	-
0x9008	09 Fieldbus system	1	1
0x900A	11 Feedback	-	7
0x900B	12 Internal communication	11	1
0x900F	16 External	1	-
0x9010	17 Technology	3	8
0x9011	18 Engineering	16	8
0x9012	19 Other	4	8

With the JVLdrive module, and for a diagnosis that concerns the whole device, the diagnosis is sent on API 0 instead. The fault classes are then moved to the manufacturer range, so 0x9000 is reported as 0x0100, 0x9001 as 0x0101, and so on (0x9012 as 0x0112). The order of the classes and the ExtChannelErrorType are the same, and the GSDML file holds the texts for both ranges.

The GSDML file carries a name and a help text for every one of those codes, so the engineering tool shows the condition in words. The two products do not report the same things, which is why the counts differ: the MAC G3 has an Ethernet interface of its own to report on, and the MIS G3 reports more from the encoder.
A pending diagnosis is not shown on the motor indicators. PROFINET signals it on the System Failure indication, and a generation III motor does not have one - see Status indicators.

7.7 Conformance and timing

Only PROFINET

The MAC G3 is declared in its GSDML file as PROFINET Conformance Class C, with RT_CLASS_1 (RT) and RT_CLASS_3 (IRT) and netload class III. The MIS G3 is declared as Conformance Class B, with RT_CLASS_1 only and netload class III, so isochronous real time (IRT) is not available on the MIS G3. The minimum device interval is 1 ms on both.

The conformance class is the one declared in the GSDML file; the device has not yet passed PROFINET (PI) certification for generation III.

7.7.1 Update time

The update time is set in the controller. It is the send clock multiplied by the reduction ratio, within the limits the GSDML files declare. The update time cannot be shorter than the minimum device interval of 1 ms.

Table 70: Update time limits

	MAC G3	MIS G3
Minimum device interval	1 ms	1 ms
Send clock	1 or 2 ms	0.5, 1, 2 or 4 ms
Reduction ratio, RT	1, 2, 4 ... 512	1, 2, 4 ... 512
Reduction ratio, IRT	1, 2, 4, 8, 16	no IRT
Conformance class	C	B

In TIA Portal the update time is set in the properties of the motor, see Quick start with TIA Portal.

7.8 I&M functions

Only PROFINET

The motor supports the Identification & Maintenance (I&M) records I&M0 to I&M3; the records I&M1, I&M2 and I&M3 are writable.

The engineering tool matches the device it finds on the network against these values in the GSDML file.

Table 71: PROFINET device identity

GSDML value	MAC G3	MIS G3
Vendor ID	0x055E (JVL A/S)	0x055E (JVL A/S)
Device ID	0x0001	0x0002
Device family	Drives / Servo Motors	Drives / Stepper Motors
Default station name	mac-servo-motor	mis-stepper-motor
Cyclic modules	JVL register, PROFIdrive	JVL register module
Minimum device interval	1 ms	1 ms

Install the GSDML file for the product in the engineering tool (for example TIA Portal). The MAC G3 and the MIS G3 have separate files, GSDML-V2.45-JVL-MAC-<date>.xml and GSDML-V2.45-JVL-MIS-<date>.xml. Use the GSDML version published on www.jvl.dk. JVL also provides a PROFINET function-block library for TIA Portal (V14 and higher) and ready-made TIA example projects, all available from www.jvl.dk.

7.10 Ethernet switches

Only PROFINET

The switch has to match the topology and the size of the network, and a switch is also what connects several separate networks to each other. On PROFINET there is one requirement that comes before all the others.

Every switch, and every other device that acts as a switch, must comply with the PROFINET RT protocol and recognise LLDP and PN_PTCP frames. A switch that does not will broadcast frames that were never meant to be broadcast, and the network can congest as a result. This is not a recommendation but a requirement.

Beyond that, port mirroring is worth having. It routes the traffic out on a separate port where a network analyser can be connected, which is what makes troubleshooting and performance monitoring possible once the network is in a machine.

The motor has a built-in two-port switch which meets the requirement above, and it is what makes the daisy chain possible without external hardware. It is useful for a limited number of motors on one line.
The motor can be part of a ring with the Media Redundancy Protocol (MRP) as an MRP client. The MRP manager must be the controller or a switch.

7.11 Byte order

Only PROFINET

Some controllers expect the cyclic data in the opposite byte order from the one standard Siemens controllers use - Kuka robot controllers are the usual example. Without a change the data is still exchanged, but every value is interpreted with its bytes the wrong way round, so the symptom is plausible-looking nonsense rather than a communication error.

The motor can swap the byte order of the cyclic data to match. In MacTalk, tick "Alt. byte order" on the PROFINET tab and then use Apply and Save so the setting survives a power cycle.

The same setting can be made by writing the register directly. The bit is bit 12 (Swap_databytes) in both products, but not in the same register: register 515 (16390 in MacTalk) on MAC G3 and register 123 on MIS G3. Save to flash after writing it.

Table 72: Byte order setting

Product	Register	Bit
MAC G3	Register 515 (COM_SETUP_BITS), 16390 in MacTalk	12
MIS G3	Register 123 (GENERAL_SETUP_2)	12

7.12 Name and IP address

Only PROFINET

PROFINET addresses a device by its name of station. The controller finds the motor on the network with the Discovery and Configuration Protocol (DCP), gives it the name from the project and then the IP address, subnet mask and gateway. The IP settings shown in MacTalk are therefore only the start-up values: once the controller has connected, the motor may use the address the controller gave it.

The motor stores what the controller assigns. A new name of station is kept in the motor, and a new IP address, subnet mask or gateway is written to registers 512, 513 and 514 (16387-16389 in MacTalk on MAC G3) and saved to flash at once, so it is also what MacTalk shows. On MIS G3 this save stores all registers, so an unsaved change made from MacTalk is saved at the same time.

The motor starts with the stored name, unless bit 2 (clear station name at start-up) is set in register 515 (16390 in MacTalk) on MAC G3 or register 123 on MIS G3. With that bit set and saved, the motor always starts without a name, and the controller has to assign it again after every start. The controller can only do that by itself if the topology is configured in the project, with every port connection as it is wired.

Reset to factory settings from the engineering tool restores the factory defaults of the motor, including the IP settings. The motor does the same whichever option is chosen in the tool. The connection to the controller is lost, because the IP settings return to their factory values.

7.12.1 Mechanical installation

The cable from the IO controller can be connected to either of the two Ethernet ports, and the next device in the line to the other port. If the topology is configured in the project, the ports must be connected exactly as they are drawn in the topology view. Use shielded Ethernet cable (STP, CAT5 or better); unshielded cable is not recommended in an industrial environment.

The commissioning sequence is the same in every PROFINET engineering tool, and only the names of the dialogs change. The steps below are the ones that have to be done in order.

- Install the GSDML file in the engineering tool. Use the file that matches the product - the MAC G3 and the MIS G3 have different Device IDs and different files - and use the version published on www.jvl.dk.
- Assign the name of station. PROFINET addresses the device by name, not by IP address, and the IP address is handed out by the controller once the name matches. The engineering tool proposes the default name from the GSDML file, see the table above. See also Name of station and IP address.
- Add the device to the PROFINET topology and select the cyclic module. The JVL register module carries eight 32-bit registers in each direction; the PROFIdrive module (MAC G3) carries the standard telegram instead.
- Set the update time. The motor accepts a device interval down to 1 ms, but the achievable value depends on the whole network - see Ethernet switches.
- Decide which registers go in the cyclic data. The choice is made in the motor, not in the controller, see Selecting the cyclic registers. Keep the error and status information in the cyclic input data: register 35 on MAC G3 (the motor inserts it if it is missing, see Rules for the cyclic registers), and on MIS G3 register 25 or the diagnosis code in register 153.
- Download the configuration and go into cyclic exchange. If the device does not reach cyclic exchange, read the diagnosis first - the motor says which part of the configuration it rejected. A step-by-step example for TIA Portal follows in Quick start with TIA Portal.

7.13.1 If the device does not reach cyclic exchange

The motor reports a rejected cyclic configuration through the PROFINET diagnosis channel, so the reason is visible in the engineering tool without any extra tooling. These are the messages that concern the cyclic data mapping.

Table 73: Diagnosis messages for the cyclic data mapping

Diagnosis text	What it means
PDO TOO MANY REGISTERS	Too many registers are mapped.
PDO DUPLICATE REGISTER	The same register appears twice.
PDO REGISTER ORDER	The entries are in the wrong order.
PDO ERROR REGISTER	Register 35 cannot be written.
PDO COMMAND REGISTER	Register 211 cannot be written.

On MAC G3, registers 35, 211 and 36 are maintained by the motor itself and cannot be written from the cyclic data. A mapping is also rejected if a register appears twice, if a motor register comes after an interface register, or if an interface register is not entered as (register - 509) x 65536. On MIS G3, a mapping is rejected if it writes a read-only register or names a register twice. All of these are configuration errors in the cyclic mapping, not network faults, and none of them are cleared by restarting the controller. A register number that does not exist is not rejected. It simply reads as 0.

[SCREENSHOT NEEDED - CLF] Installing the GSDML, assigning the device name via DCP, mapping the 8x32 registers or a PROFIdrive telegram, and going into cyclic exchange.

TT8821-10GB

Screenshot pending

ADO #1731 · Christian Lotz Felter (clf@jvl.dk)

LB0077 · section 7.13.1

The screenshot must show:

Installing the GSDML, assigning the device name via DCP, mapping the 8x32 registers or a PROFIdrive telegram, and going into cyclic exchange

Placeholder - not for release. Replace this file with the screenshot under the same TT number; the manual then needs no change.

7.14 Quick start with TIA Portal

Only PROFINET

This section shows the steps in TIA Portal for a MAC G3 with the JVLdrive module. The MIS G3 is set up the same way, with its own GSDML file.

- Install the GSDML file: Options > Manage general station description files (GSD), select the file and install it.
- Open the hardware catalogue under Other field devices > PROFINET IO > Drives > JVL A/S, and drag the motor into the Network view, see figure TT8845-10GB.

[SCREENSHOT NEEDED - CLF] Manage GSD dialog with the MAC G3 and MIS G3 files, and the hardware catalogue with MAC and MIS.

Screenshot pending TT8845-10GB

ADO #1731 · Christian Lotz Felter (clf@jvl.dk)

LB0077 · section PROFINET - Quick start with TIA Portal

The screenshot must show:

TIA Portal: Manage general station description files with the MAC G3 and MIS G3 GSDML files installed, and the hardware catalogue Other field devices > PROFINET IO > Drives > JVL A/S with MAC and MIS

Placeholder - not for release. Replace this file with the screenshot under the same TT number; the manual then needs no change.

- Connect the motor to the PROFINET interface of the PLC, and assign the name of station with Assign device name, see figure TT8846-10GB.

[SCREENSHOT NEEDED - CLF] Network view with the PLC and a MAC G3, and the Assign device name dialog.

Screenshot pending TT8846-10GB

ADO #1731 · Christian Lotz Felter (clf@jvl.dk)

LB0077 · section PROFINET - Quick start with TIA Portal

The screenshot must show:

TIA Portal Network view with the PLC connected to a MAC G3, and the Assign device name dialog listing the motor as an accessible device

Placeholder - not for release. Replace this file with the screenshot under the same TT number; the manual then needs no change.

- Open the Device view and insert the JVLdrive module in slot 1 from the catalogue, see figure TT8847-10GB. On a MIS G3 the module is already inserted. For PROFIdrive on a MAC G3, insert the PROFIdrive module and select the telegram instead.

7.14 Quick start with TIA Portal

Only PROFINET

[SCREENSHOT NEEDED - CLF] Device overview of a MAC G3 with the JV-Ldrive module (Cyclic data interface) in slot 1.

Screenshot pending TT8847-10GB

ADO #1731 · Christian Lotz Felter (clf@jvl.dk)

LB0077 · section PROFINET - Quick start with TIA Portal

The screenshot must show:

TIA Portal Device overview of a MAC G3 with the JVLdrive module (Cyclic data interface, 8 x 32-bit input and output) in slot 1

Placeholder - not for release. Replace this file with the screenshot under the same TT number; the manual then needs no change.

- Set the update time under PROFINET interface > Advanced options > Real time settings > IO cycle, see figure TT8848-10GB.

[SCREENSHOT NEEDED - CLF] Real time settings > IO cycle with the update time set to 1 ms.

Screenshot pending TT8848-10GB

ADO #1731 · Christian Lotz Felter (clf@jvl.dk)

LB0077 · section PROFINET - Quick start with TIA Portal

The screenshot must show:

TIA Portal properties of the motor: PROFINET interface > Advanced options > Real time settings > IO cycle, with the update time set to 1 ms

Placeholder - not for release. Replace this file with the screenshot under the same TT number; the manual then needs no change.

- Compile and download the configuration. The eight input and output registers are now available in the PLC program; their meaning is set by the cyclic register selection in the motor, see Default cyclic registers.

7.15 Examples

Only PROFINET

The examples use the JVLdrive module with the default cyclic registers, unless they say otherwise. Registers are given by number and each product uses its own numbers; see Register numbering in Common setup and functions. The record indices are listed in Record access from TIA Portal.

7.15.1 Velocity control

MAC G3: set register 5 (V_SOLL) to the velocity and register 6 (A_SOLL) to the acceleration, then write 1 (velocity mode) to register 2. With the default motor cycle of 1 ms and 8192 counts per revolution, V_SOLL = rpm x 2.1845, and one unit of A_SOLL is 457.8 rpm/s. For 1200 rpm, write 2621 to register 5.

MIS G3: set register 3 to the velocity in 0.01 rpm and register 6 to the acceleration in rpm/s, then write 1 to register 2. For 1200 rpm, write 120000 to register 3.

Write 0 to register 2 to make the motor passive again.

7.15.2 Position control

MAC G3: write the target to register 3 (P_SOLL, 8192 counts per revolution), the maximum velocity to register 5 and the acceleration to register 6, then write 2 (position mode) to register 2.

MIS G3: write the target to register 4 (409600 counts per revolution), the maximum velocity to register 3 and the acceleration to register 6, then write 2 to register 2.

Follow the move in register 10 (actual position) on both products. The move is finished when the in-position bit is set: bit 4 of register 35 on MAC G3, and bit 4 of register 25 on MIS G3. Register 25 is not in the default MIS G3 read list; add it if it is needed.

7.15.3 General considerations

Keep the error and status information in the read list and monitor it in the controller: register 35 on MAC G3 (the motor inserts it if it is missing, see Rules for the cyclic registers), and on MIS G3 the status bits in register 25 and the diagnosis code in register 153. When an error occurs the motor stops, and the cause is shown there and reported as a diagnosis, see Diagnosis.

The motion status is in the same place. MAC G3 register 35: bit 4 in position, bit 5 accelerating, bit 6 decelerating, bit 18 homing done. MIS G3 register 25: bit 4 in position, bit 5 accelerating, bit 6 decelerating, bit 9 homing done.

7.15.4 Homing using cyclic data

Set up the homing in the motor once, in MacTalk, and save it to flash: the homing velocity and position and, on MAC G3, the homing torque. Set the start-up mode to position if the motor should hold its position after the homing - the motor switches to the start-up mode when the homing is done.

MAC G3: keep register 2 in both the read list and the write list. Write the homing mode to register 2: 12 torque homing, 13 sensor homing forward, 14 sensor homing forward and backward, 25 index homing, 26 quick index homing. While the homing runs, the motor does not accept the other cyclic writes. When bit 18 of register 35 is set, write the mode you want next to register 2 - this releases the cyclic writes again.

MIS G3: add register 25 to the read list, and write 13 (sensor homing, one direction) or 14 (both directions) to register 2. MIS G3 does not block the cyclic writes during the homing, so change register 2 to the next mode as soon as bit 9 of register 25 is set.

7.15.5 Relative positioning

The motor must be in position mode. The move is finished when the in-position bit is set.

MAC G3: the interface command register is in the write list by default (slot 6, entered as 983040). Write the distance to motor register 61 (P7) - put it in the write list, or write it once to index 0x123D. Then write 0x010000F1 (16777457) to the command register; the motor sets P_SOLL to the actual position plus P7. Write 0 to the command register before the next move.

MIS G3: register 24 is the command register and is in the write list by default. Write the distance to register 330 (P7, index 0x014A), then write 113 to register 24. Write 0 to register 24 before the next move. In this release only positive distances are handled correctly.

7.15.6 Saving parameters to flash

MAC G3: write 16 to index 0x1000 to save the interface registers, 272 to save the motor registers, or 273 to save both and restart. MIS G3: write 269 to register 24 (index 0x0018) to save, or 268 to save and restart.

7.15.7 Clearing errors

MAC G3: write 0x010000E1 (16777441) to the command register in the write list, and 0 again afterwards; or write 225 to motor register 211 (index 0x12D3). Both clear the errors of the motor and of the Ethernet interface. MIS G3: write 265 to register 24, cyclically or through index 0x0018. With a PROFIdrive telegram, use bit 7 of STW1.

7.15.8 Changing the cyclic setup

The controller can change the cyclic register selection itself, for example at start-up.

MAC G3: write the register numbers to cyclic read 1..8 through indices 0x1110-0x1117 and to cyclic write 1..8 through indices 0x1118-0x111F. Then write 21 to index 0x1000 to take the selection into use, and 16 to save it.

MIS G3: write the register numbers to registers 525-540 through indices 0x020D-0x021C. Then write 300 to register 24 (index 0x0018) to take the selection into use, and 269 to save it.

The rules in Rules for the cyclic registers apply. A selection the motor rejects is reported as a diagnosis, see If the device does not reach cyclic exchange.

8 Using MacTalk over Ethernet

8.1 Introduction

MacTalk can connect to a MAC G3 or MIS G3 motor over Ethernet as well as over the RS485 connection. On EtherNet/IP and PROFINET, and through the master on EtherCAT, MacTalk reaches the motor through the same cable as the real-time protocol, so the motor can be set up and monitored without a separate serial cable. This chapter describes how the PC and MacTalk are set up, how motors are found on the network, and what is different on EtherCAT.

8.1.1 Services on the Ethernet port

The Ethernet interface has its own IP address for MacTalk and Modbus TCP/UDP, separate from the address of the real-time protocol stack (see IP configuration in Common setup and functions). MacTalk must always be pointed at this MacTalk / Modbus address. On every protocol the factory default is 192.168.0.N with subnet mask 255.255.255.0, where N is the last byte of the first MAC address printed on the motor label (1 if that byte is 0).

The interface answers the services in the table below on the MacTalk / Modbus address. MacTalk first identifies the motor with the MacTalk protocol on UDP port 47100. When it finds a MAC G3 or MIS G3 it continues the session with Modbus over UDP on port 502, and the firmware update uses the same port. A MacTalk session therefore does not use one of the eight Modbus TCP connections that the interface can hold.

Table 74: Ethernet services used by MacTalk and Modbus

Service	Transport	Port	Used for
MacTalk protocol	UDP	47100	MacTalk identifies the motor
Modbus UDP	UDP	502	MacTalk session, firmware update
Modbus TCP	TCP	502	PLC, SCADA and other clients
ICMP echo	-	-	ping

The behaviour is the same on MAC G3 and MIS G3. On MAC G3 the Ethernet interface passes access to the motor registers on to the motor itself; this is invisible in MacTalk.

8.1.2 Availability per protocol

Whether MacTalk can reach the motor over Ethernet depends on the protocol installed in the Ethernet interface (register 558, 16433 in MacTalk on MAC G3):

Table 75: MacTalk over Ethernet per protocol

Protocol	MacTalk over Ethernet	Firmware update
EtherNet/IP	Yes, directly to the motor	Over Ethernet
PROFINET	Yes, directly to the motor	Over Ethernet
EtherCAT	Yes, through EoE in the master	Over RS485 only
POWERLINK	No	Over RS485 only

On EtherNet/IP, PROFINET and POWERLINK the PC can be connected directly to the motor or to the same switch. On EtherCAT the motor is reached only through the EtherCAT master, which forwards the MacTalk frames through Ethernet over EtherCAT (EoE); see Ethernet over EtherCAT below.

8.1.3 Limitations

- The motor answers MacTalk only when it is addressed directly (unicast), not a broadcast search. The Motor Scanner finds it with Full range in MacTalk 1.94 and later (see Finding motors on the network); with an older MacTalk, connect by entering the IP address of the motor (see Entering the IP address).
- MacTalk over Ethernet uses UDP. TCP port 47100 is not served.

8.1 Introduction

- The MacTalk / Modbus interface has no DHCP client. Its DHCP bit (bit 9 of register 515 (16390 in MacTalk) on MAC G3, bit 9 of register 123 on MIS G3) has no effect, so the address is always static.
- Firmware update over Ethernet is possible on EtherNet/IP, PROFINET and POWERLINK. On EtherCAT update the firmware over RS485.
- Use one connection at a time: MacTalk talks to the motor either over the serial port or over Ethernet, as selected in the port selector.

8.2 Setting up the PC

8.2.1 IP addresses

For IP communication, the PC and the motor must be on the same subnet, unless routing between their subnets is configured. With the factory defaults the MacTalk / Modbus address of the motor is 192.168.0.N with subnet mask 255.255.255.0, where N is the last byte of the first MAC address on the motor label, converted from hexadecimal to decimal (for example, a MAC address ending in 14 gives 192.168.0.20). This only applies while the motor still has its factory settings.

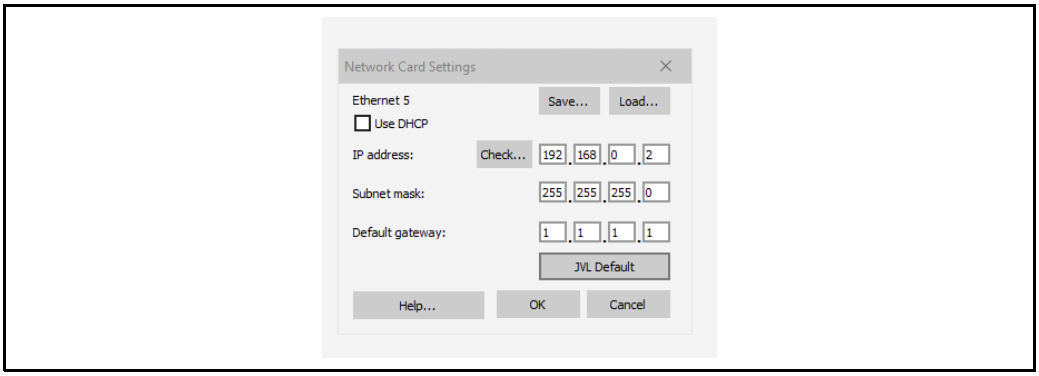
Give the PC an address in the range 192.168.0.1 to 192.168.0.254 with subnet mask 255.255.255.0 that no other device on the network uses, motors included, for example 192.168.0.2. If a DHCP server already gives the PC an address in this subnet, no manual setting is needed.

The motor does not act as a DHCP server. When connecting the PC directly to the motor, or through a network without a DHCP server, manually configure the PC's Ethernet adapter with an unused IP address in the motor's subnet.

8.2.2 Network Card Settings in MacTalk

MacTalk can set the address of the PC network adapter itself. Select the adapter in the port selector (see Selecting the Ethernet adapter) and press the button next to the port selector ("Setup the network adaptor card and IP settings for MacTalk"). The Network Card Settings dialog shows the IP address, subnet mask and default gateway of the adapter, and a Use DHCP option.

JVL Default fills in IP address 192.168.0.2, subnet mask 255.255.255.0 and default gateway 1.1.1.1. This puts the PC in the same subnet as the factory default MacTalk / Modbus address of the motor (192.168.0.N). Press OK to apply. MacTalk changes the adapter through the Windows netsh command, so Windows asks for administrator rights. Save stores the settings in a MacTalk IP configuration file.



8.2.3 Windows network settings

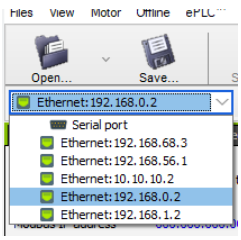
The same settings can be made in Windows: open the properties of the network adapter, select Internet Protocol Version 4 (TCP/IPv4), choose to use a fixed IP address and enter the address and subnet mask. The exact dialogs depend on the Windows version.

On a PC with more than one network adapter (for example a wireless one), it can be necessary to disable the unused adapter, or to make sure that the two adapters are on different subnets, so that Windows sends the frames to the motor through the right adapter.

8.3 Connecting from MacTalk

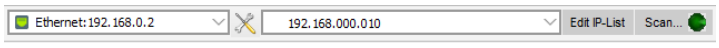
8.3.1 Selecting the Ethernet adapter

When MacTalk starts for the first time it is set up for the serial RS485 connection. The port selector at the left of the second toolbar row lists Serial port and one entry for each IP address of the network adapters in the PC, shown as the adapter type followed by the address, for example Ethernet:192.168.0.2. An adapter without an IP address is not listed. Select the entry of the adapter that is connected to the motor network. The COM port, baud rate and motor address selectors are then replaced by an IP address field and the Edit IP-List button.



8.3.2 Entering the IP address

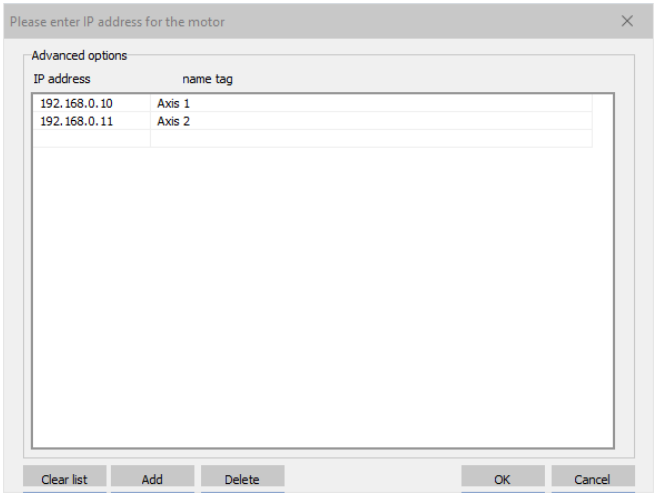
Enter the MacTalk / Modbus IP address of the motor in the IP address field, for example 192.168.0.20, and MacTalk connects. MacTalk remembers the last address used; G3 motors have an address field of their own, separate from the one used for older motors.



8.3.3 IP list and name tags

Edit IP-List opens the dialog "Please enter IP address for the motor". Each line holds an IP address and a name tag. The name tag is shown in the IP address field instead of the address, which makes it easier to keep several motors apart.

- Add adds a line, Delete removes the selected line and Clear list removes all lines.
- OK saves the list. It is loaded again every time MacTalk starts, and both the addresses and the name tags can be selected in the IP address field.



8.3 Connecting from MacTalk

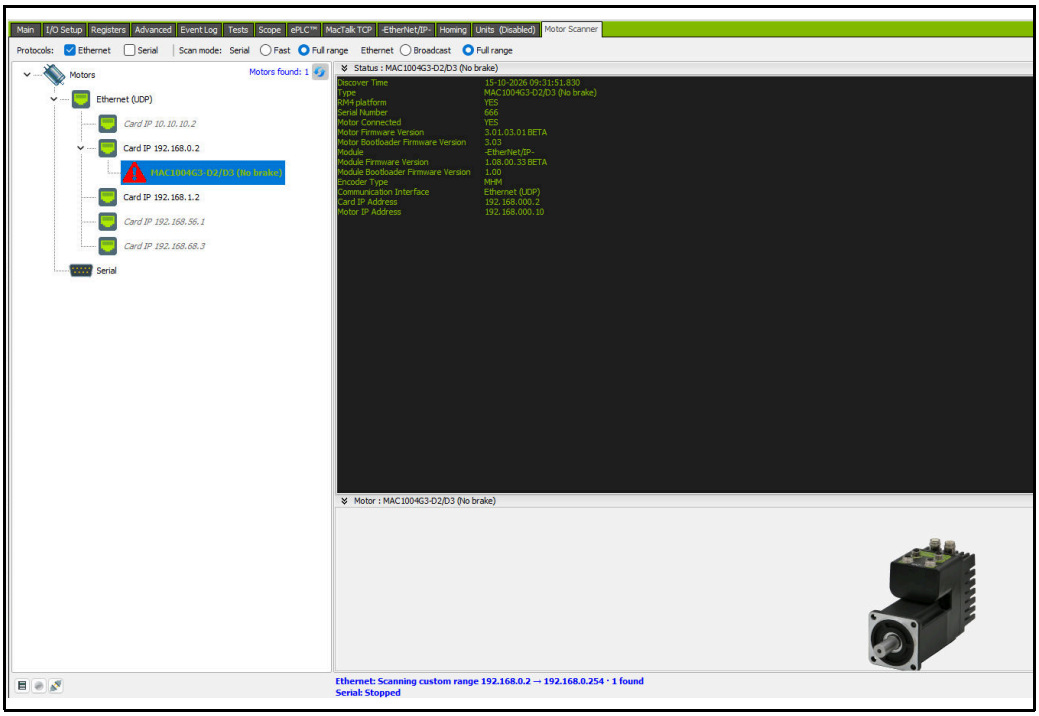
8.3.4 Finding motors on the network

Scan opens the Motor Scanner tab. The Motor Scanner searches the network for motors and lists each motor it finds with its type, serial number, firmware versions and IP address, without connecting to it first. MAC G3 and MIS G3 motors are found by MacTalk 1.94 and later. With an older MacTalk, connect by entering the IP address of the motor (see Entering the IP address).

To search the Ethernet network for G3 motors:

- Tick Ethernet under Protocols.
- In the Scan mode line, select Full range under Ethernet (the line has a Full range for Serial as well). MacTalk then asks every address in the range directly. MAC G3 and MIS G3 motors do not answer the faster Broadcast search in this release.
- Open More and, under Network adapters to scan, tick the adapter that the motors are connected to and clear the others. An adapter with several IP addresses is listed under its first address.
- Tick Use custom IP range and enter the subnet of the MacTalk / Modbus addresses of the motors, for example 192.168.0.1-254. MacTalk searches the range from the address of the adapter that is in the same subnet.
- Press Start. The status line at the bottom of the tab shows the range being searched and the number of motors found.

Each motor found is listed under the adapter address it was found from. Until MacTalk is connected to it, the motor is shown with a red warning triangle; this does not mean that the motor has an error. Holding the mouse pointer over the motor shows the tip "Motor found. Go online by double clicking on motor". Select the motor to see its details, and double-click it to connect MacTalk to it. The triangle then disappears.



8.4 The MacTalk TCP tab

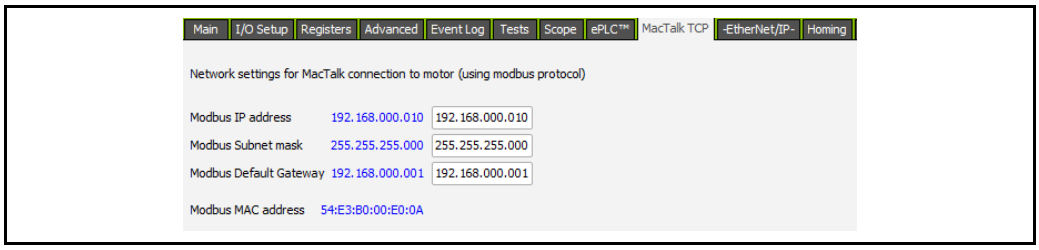
8.4.1 Configuration via MacTalk

On MAC G3 and MIS G3 the address of the MacTalk / Modbus interface is set on the MacTalk TCP tab ("Network settings for MacTalk connection to motor (using modbus protocol)"). The address of the real-time protocol stack is set on the tab of the protocol, see IP configuration in Common setup and functions.

Table 76: Fields on the MacTalk TCP tab

Field	Register	Content
Modbus IP address	553	IP address of the MacTalk interface
Modbus Subnet mask	554	Subnet mask
Modbus Default Gateway	555	Gateway
Modbus MAC address	8210	MAC address (EoE can set it)

Save the settings in the motor. A new address is used after the motor has been restarted, so reconnect MacTalk to the new address afterwards. The addresses are kept through a firmware update.



8.4.2 Configuration via motor registers

From a controller, or over Modbus TCP/UDP, the same settings are in these registers. Each address occupies one 32-bit register with the first byte in the most significant byte, so 192.168.0.20 is 0xC0A80014:

- Register 553 (16428 in MacTalk on MAC G3) - MacTalk / Modbus IP address.
- Register 554 (16429 in MacTalk on MAC G3) - MacTalk / Modbus subnet mask.
- Register 555 (16430 in MacTalk on MAC G3) - MacTalk / Modbus gateway.

Save the registers to flash and restart the motor for the new address to take effect.

8.4.3 Switching the Ethernet services off

MacTalk over Ethernet and Modbus TCP/UDP can be switched off together, for example where register access over the network is not wanted. The bit is in the communication setup register (515, 16390 in MacTalk) on MAC G3 and in general setup register 2 (123) on MIS G3:

Table 77: Bits that switch the Ethernet services off

Product	Register	Bit	When set to 1
MAC G3	515	14	MacTalk and Modbus over Ethernet off
MIS G3	123	16	MacTalk and Modbus over Ethernet off

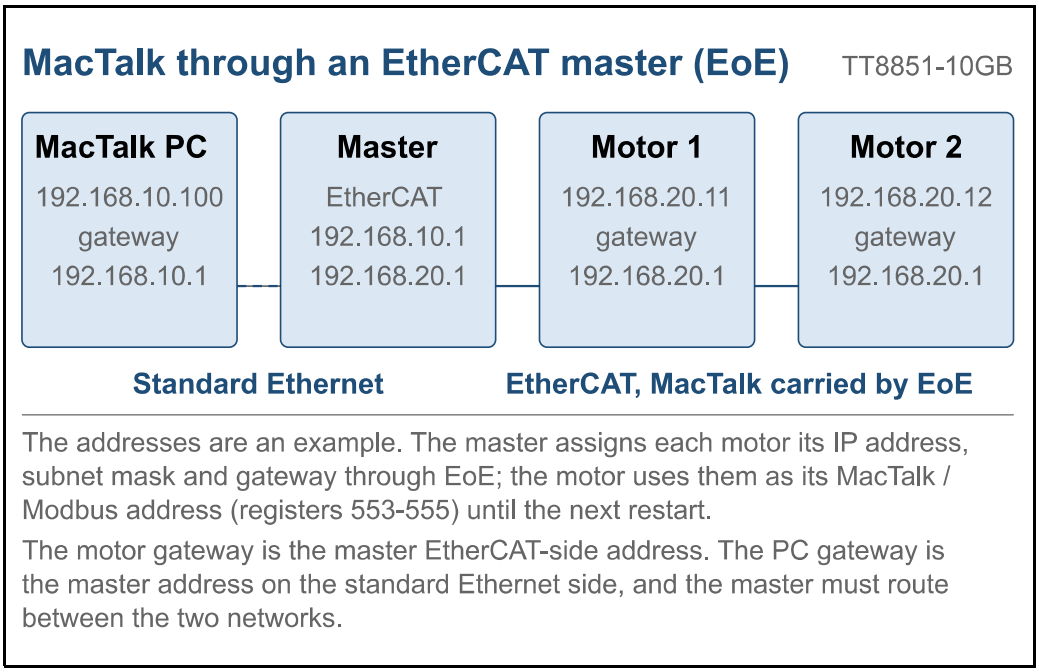
With the bit set, the interface does not answer on UDP port 47100 or on UDP and TCP port 502; ping is still answered. The bit is read when the interface starts, so save it and restart the motor. To switch the services on again, clear the bit over the RS485 connection or through the real-time protocol. On EtherCAT the bit has no effect in this release.

8.5 Ethernet over EtherCAT

8.5.1 How it works

On EtherCAT MacTalk is not connected to the motor directly. EtherCAT is a master-slave network, and the only way onto it is through the master. The master receives the MacTalk frames on its standard Ethernet port and forwards them to the motor through Ethernet over EtherCAT (EoE), a mailbox protocol that carries ordinary Ethernet frames between the cyclic EtherCAT frames. The motors declare EoE with IP and MAC support in their ESI file.

The master assigns the IP address, subnet mask and gateway of each motor through EoE, and can also assign its MAC address. The motor uses them as its MacTalk / Modbus address (registers 553-555, and 8210 for the MAC address; 16428-16430 and 24594 in MacTalk on MAC G3); the address set on the MacTalk TCP tab is replaced until the next restart. Until the master has sent the IP parameters, the motor cannot be reached from MacTalk. The same EoE path carries Modbus TCP/UDP.



The addresses are an example. The master assigns each motor its IP address, subnet mask and gateway through EoE; the motor uses them as its MacTalk / Modbus address (registers 553-555) until the next restart.

The motor gateway is the master EtherCAT-side address. The PC gateway is the master address on the standard Ethernet side, and the master must route between the two networks.

The master acts as a router between two subnets: the PC network (192.168.10.x in the example) and the EoE network of the motors (192.168.20.x). The steps below use a Beckhoff TwinCAT master; other masters have equivalent settings.

8.5.2 Setting up the master

- Select the EtherCAT device in the I/O tree, open the EtherCAT tab and press Advanced Settings.
- Under EoE Support, tick Connect to TCP/IP Stack and IP Enable Router. Changing the routing setting requires a restart of the master PC.
- Note the IP addresses of the two interfaces of the master: the standard Ethernet port that the MacTalk PC is connected to, and the EtherCAT port.

[SCREENSHOT NEEDED - CLF] TT8858-10GB: TwinCAT 3 EoE Support with IP Enable Router.

8.5 Ethernet over EtherCAT

Screenshot pending TT8858-10GB

ADO #2011 · Christian Lotz Felter (clf@jvl.dk)

LB0077 · section 8.5.2

The screenshot must show:

TwinCAT 3: the EtherCAT device, Advanced Settings, EoE Support, with Connect to TCP/IP Stack and IP Enable Router ticked

Placeholder - not for release. Replace this file with the screenshot under the same TT number; the manual then needs no change.

8.5.3 Setting up the motors

- Select the drive in the I/O tree, open the EtherCAT tab and press Advanced Settings.
- Under Mailbox, EoE, tick Virtual Ethernet Port, select IP Port and IP Address, and enter the IP address, subnet mask and default gateway of the motor.
- The IP address must be in the same subnet as the EtherCAT port of the master, and the default gateway must be exactly the address of that port.

[SCREENSHOT NEEDED - CLF] TT8859-10GB: TwinCAT 3 EoE settings of a MAC G3 or MIS G3 drive.

Screenshot pending TT8859-10GB

ADO #2011 · Christian Lotz Felter (clf@jvl.dk)

LB0077 · section 8.5.3

The screenshot must show:

TwinCAT 3: a MAC G3 or MIS G3 drive, Advanced Settings, Mailbox, EoE, with Virtual Ethernet Port, IP Port and IP Address set

Placeholder - not for release. Replace this file with the screenshot under the same TT number; the manual then needs no change.

8.5.4 Setting up the PC

Set up the PC as described in Setting up the PC, but with an address in the subnet of the standard Ethernet port of the master, and with the default gateway set to the address of that port. In the example: IP address 192.168.10.100, subnet mask 255.255.255.0, gateway 192.168.10.1. Then connect MacTalk to the address of the motor, 192.168.20.11.

8.5.5 Testing the connection

Test the path one step at a time from a command prompt on the MacTalk PC. If one step fails, check the settings of that step before going on:

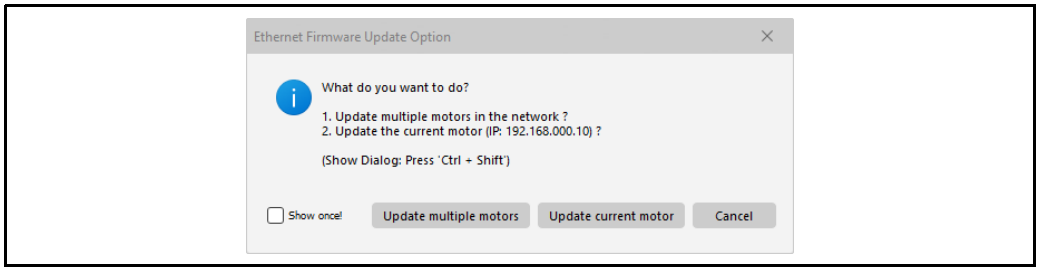
- ping 192.168.10.1 - the standard Ethernet port of the master.
- ping 192.168.20.1 - the EtherCAT port of the master.

- ping 192.168.20.11 - the motor.

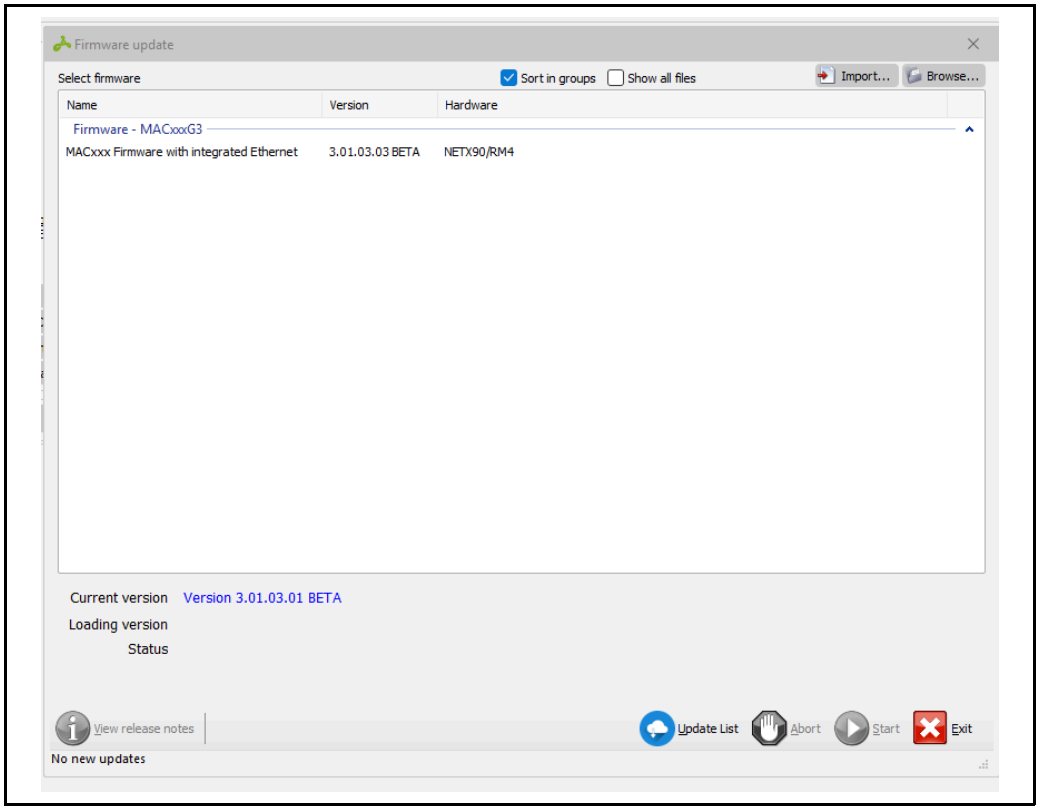
8.6 Firmware update over Ethernet

On EtherNet/IP, PROFINET and POWERLINK the motor firmware and the Ethernet interface firmware can be updated over Ethernet. Connect MacTalk to the motor over Ethernet and select Updates, Update Firmware. MacTalk transfers the firmware file over Modbus UDP, port 502, and the motor restarts when the new firmware has been written. The firmware file contains the images for all protocols, so a different protocol can be installed the same way (see Selecting the protocol in the Introduction chapter).

When MacTalk is connected over Ethernet, it first asks whether to update several motors on the network or only the motor it is connected to, shown with its IP address:



Press Update current motor. The Firmware update dialog then opens:



- Under Select firmware, select the firmware to install. The list shows the firmware files for the motor with name, version and hardware. The Current version field shows the firmware the motor has now.
- To install a firmware file (.maf) saved on the PC, for example one received from JVL, add it to the list with Import.
- View release notes shows what has changed in the selected version.

8.6 Firmware update over Ethernet

- Press Start. Loading version shows the firmware being written, and Status shows the progress.
- When Status shows Done, press Done. Check that Current version now shows the new version.

The IP addresses, the subnet masks and the gateways of both interfaces are kept through the update. Do not remove the power while the update is running.

On EtherCAT the firmware cannot be updated over Ethernet in this release. Use MacTalk over the RS485 connection instead. That connection is Modbus RTU at 115200 baud by default and works whichever protocol is installed.

- MacTalk cannot connect: check that the address entered is the MacTalk / Modbus address (register 553, 16428 in MacTalk on MAC G3, shown on the MacTalk TCP tab) and not the address of the protocol stack (register 512, 16387 in MacTalk on MAC G3). The two are different addresses.
- The motor does not answer ping: check the cable, the adapter selected in the port selector and that the PC and the motor are on the same subnet. On a PC with several adapters, disable the unused ones.
- The motor answers ping but MacTalk cannot connect: the Ethernet services may be switched off (see Switching the Ethernet services off), or a firewall on the PC blocks UDP ports 47100 and 502.
- The Motor Scanner finds no G3 motors: use MacTalk 1.94 or later, select Full range, and tick only the adapter that the motors are connected to (see Finding motors on the network). G3 motors do not answer a Broadcast search.
- EtherCAT: the motor cannot be reached until the master has assigned its EoE IP parameters. Check the EoE settings of the drive and the routing in the master, and test with ping as described in Testing the connection.

8.7 Troubleshooting

9 Examples common to all protocols

9.1 Introduction

The examples in this chapter work the same way whichever Industrial Ethernet protocol is selected, because they use only the motor registers and, on the protocols that have cyclic data, the cyclic data setup. Modbus TCP/UDP has no cyclic data: write the same values to the registers directly. Register numbers are each product's own; see Register numbering in Common setup and functions.

9.2 ePLC and the Ethernet interface

9.2.1 Where the ePLC runs

Both MAC G3 and MIS G3 have an embedded PLC (ePLC) that runs a program made in the MacTalk ePLC editor. The program runs alongside the Industrial Ethernet protocol, and the Ethernet master and the ePLC program can use the same motor registers.

- MAC G3: the ePLC runs in the motor processor. When a program is downloaded over Ethernet, the Ethernet interface only passes it on to the motor; the interface has no ePLC registers of its own.
- MIS G3: the ePLC runs in the same processor as the Ethernet interface. The program is stored in the motor, and the Ethernet master can start and stop it and read its status, see ePLC control from the master.

The ePLC instructions are described in the MAC G3 (LB0067) and MIS G3 (LB0076) user manuals.

9.2.2 Using the motor I/O in the ePLC

The ePLC uses the motor's own digital inputs and outputs directly. The I/O and its connection are described in the MAC G3 (LB0067) and MIS G3 (LB0076) user manuals.

The Ethernet master can follow the same I/O by placing these registers in the cyclic read list:

Table 78: Motor I/O and ePLC registers

Item	MAC G3	MIS G3
Digital inputs	106	18
Digital outputs	114	19
ePLC status	-	1861

MAC G3: register 114 shows the state actually written to the digital outputs. Bits 30 and 31 of register 35 also show outputs O1 and O2, and register 35 is always in the cyclic read list on MAC G3.

MIS G3: when the ePLC program starts, it switches all its outputs off. After that the ePLC writes the whole output register 19 each time the program sets or clears an output. Do not write register 19 from the Ethernet master while an ePLC program is running, as the ePLC overwrites the value at its next output instruction.

[EDITORIAL NOTE - NOT FOR RELEASE] ADO #2022 CLF

9.3 Reset errors from the master

Errors are often cleared with a command to the motor command register. A command must run only once, but the Ethernet master sends the cyclic data in every cycle. On MAC G3 a command in the cyclic write data is therefore executed only when its value changes. To send the same command again, first write another value - the re-arm value - to the same write word.

Table 79: Commands that clear errors

Item	MAC G3	MIS G3
Command register	983040 (cyclic)	24 (acyclic)
Clear errors	16777441	265
Clear errors and warnings	16777441	97
Re-arm value	0	-

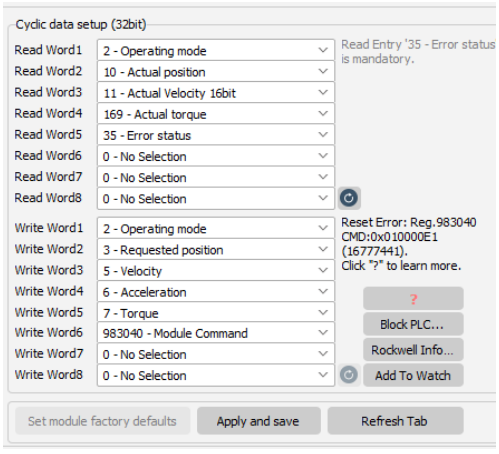
9.3.1 MAC G3

On MAC G3 the motor command register 211 cannot be placed in the cyclic write list; a setup that contains it is rejected. Commands are sent through the interface command register 524 instead, which is entered as 983040 in the cyclic list. A value with bit 24 set is passed on to register 211 with its lower 16 bits as the command, so 0x010000E1 (16777441) sends command 225, clear errors. It also clears the errors of the Ethernet interface.

Setup: in the cyclic data setup on the MacTalk Ethernet tab, select the general command register (983040) in a free write word. Place it after all motor registers - interface registers must come after the motor registers in the list, or the setup is rejected. Save the setup in the motor.

Use: to clear the errors, write 16777441 to that write word. When the errors are cleared, write 0 to re-arm it for the next command. The value 0 itself is never sent as a command.

Without cyclic data, for example with Modbus TCP/UDP, write 225 to motor register 211 instead. This also clears the errors of the Ethernet interface.



9.3.2 MIS G3

On MIS G3, write 265 to register 24 to clear the errors, or 97 to clear both the errors and the warnings. Write the command acyclically (see the chapter for the protocol in use). Register 24 is in the cyclic write list by default (register 538); remove it by writing 0 to register 538, save to flash and restart the motor. While register 24 is in the list, the

motor runs command 0 in every cycle in which the master does not change the value: the requested position is replaced by P1 (register 318), and register 2 reads passive.

[EDITORIAL NOTE - NOT FOR RELEASE] ADO #2014 DJO

9.4 ePLC control from the master

This section applies to MIS G3. The Ethernet master can start, stop and pause the ePLC program by writing a command to register 24. The commands are refused while the motor is password locked.

Table 80: ePLC commands, MIS G3 register 24

Value	Action
286	Start the program, or resume it after a pause
287	Stop the program; the next start begins from the top
288	Pause the program while it is running
291	Reset the program; accepted only when stopped or paused

[EDITORIAL NOTE - NOT FOR RELEASE] ADO #2029 CLF

The state of the program can be read in register 1861 (18245 in MacTalk on MAC G3), for example by placing it in the cyclic read list. The status code is in bits 24 to 31, so shift the value 24 bits to the right (or divide it by 16777216) to get it.

Table 81: ePLC status codes, register 1861 bits 24-31

Code	Meaning
0	Passive - the program has not been started
1	Running
2	Single step (MacTalk debugging)
3	Paused
7	Stopped
9	Starting
12	No program in the motor
Other	Error or MacTalk debug state - see MacTalk

On MAC G3 the ePLC runs in the motor processor, and the Ethernet interface has no ePLC status or control registers. See the ePLC chapter of the MAC G3 user manual (LB0067).

10 Technical data

10.1 Introduction

This chapter gives the technical data of the Ethernet interface that is built into the MAC G3 and MIS G3 motors, the data of each protocol, and the data of the user I/O. The data of the motor itself - power, torque, supply, mechanics and environment - is given under Technical data in the MAC G3 user manual (LB0067) and the MIS G3 user manual (LB0076).

The Ethernet interface is part of the control circuitry of the motor and has no supply connector of its own. See the control voltage (CVI) data in the user manual for the motor.

10.2 Ethernet interface

The Ethernet interface is the same hardware for every protocol. The protocol is given by the firmware installed on the interface - see Selecting the protocol in the Introduction.

Table 82: Ethernet interface, general data

Item	Value
Communication processor	Hilscher netX90
Physical layer	100BASE-TX, 100 Mbit/s
Ethernet ports	2 (in and out), for line topology
Connectors	M12, 4-pin, female, D-coded
Ports, MAC G3	XG1 (LA-I) in, XG2 (LA-O) out
Ports, MIS G3	CN2 in, CN3 out
Cable	Shielded, max. 100 m between devices
Termination	Not required
Galvanic isolation	Yes - see the text below the table
Tightening torque	0.6 Nm

Both Ethernet ports are galvanically isolated. On the MAC G3 the two ports belong to isolation groups 2 and 3, which are isolated from each other and from the rest of the motor. On the MIS G3 the Ethernet isolation withstands a potential difference of up to 500 VDC.

The M12 connectors must be tightened with 0.6 Nm to prevent ingress of fluids and particles, which can cause malfunction.

10.2.1 Connector pin-out

The two Ethernet connectors have the same pin-out. The in port (XG1 on the MAC G3, CN2 on the MIS G3) carries Ethernet channel 0, and the out port (XG2 or CN3) carries channel 1.

Table 83: Ethernet connector pin-out (XG1, XG2, CN2, CN3)

Pin	Signal	Description
1	Tx_P	Transmit, positive terminal
2	Rx_P	Receive, positive terminal
3	Tx_N	Transmit, negative terminal
4	Rx_N	Receive, negative terminal
Housing	Earth	Cable screen, on JVL cables

JVL supplies cables with an M12 male D-coded 4-pin connector: WI1046-M12M4S05R (5 m, loose wire ends), WI1046-M12M4S15R (15 m, loose wire ends) and WI1046-M12M4S05-NRJ45 (5 m, with an RJ45 connector). An M12 to M12 cable is available as WI1046-M12M4TM4TxxT, where xx is the length.

The cable WI1046-M12M4S05-NRJ45 is wired straight through from the M12 connector to the RJ45 connector as shown below. The RJ45 connector has an internal metal housing that is connected to the cable screen. Some cables use orange and green wires instead of brown and blue; the pins are the same.

Table 84: WI1046 cable, M12 to RJ45 wiring

M12 pin	Signal	RJ45 pin	Wire colour
1	Tx_P	1	Brown/White
3	Tx_N	2	Brown
2	Rx_P	3	Blue/White
4	Rx_N	6	Blue
Housing	Screen	Housing	Screen

10.2.2 Protocols

Each motor runs one real-time protocol. Modbus TCP/UDP and MacTalk over Ethernet are available alongside it; on EtherCAT they are carried through Ethernet over EtherCAT (EoE). On POWERLINK only Modbus UDP and MacTalk over Ethernet are available - Modbus TCP is not. Register 558 (read only; 16433 in MacTalk on MAC G3) reports which protocol is installed:

Table 85: Protocol reported in register 558

Protocol	Register 558
EtherNet/IP	0x34 (52)
EtherCAT	0x35 (53)
POWERLINK	0x36 (54)
PROFINET	0x37 (55)

10.2.3 Services

Modbus TCP/UDP and MacTalk use the MacTalk / Modbus IP address of the motor, not the address of the protocol stack - see IP configuration in Common setup and functions. On POWERLINK, Modbus TCP and the TCP connection limit do not apply.

Table 86: Services on the MacTalk / Modbus IP address

Service	Transport and port
Modbus TCP	TCP, port 502
Modbus UDP	UDP, port 502
MacTalk over Ethernet	UDP, port 47100
TCP connections	Max. 8 at the same time

Interface firmware later than 1.8.0.2 can switch Modbus TCP/UDP and MacTalk over Ethernet off: bit 14 of register 515 (16390 in MacTalk) on the MAC G3, and bit 16 of register 123 on the MIS G3. The bit is read when the motor starts, so save the setting and restart the motor. To switch the services on again, clear the bit through the real-time protocol (on the MIS G3 also with MacTalk over the RS485 connection), save it and restart the motor.

[EDITORIAL NOTE - NOT FOR RELEASE] ADO #2159 CLF

10.3 Protocol data

The tables below give the data for each protocol. With the JVL register profile the cyclic data is eight 32-bit registers in each direction on every protocol, selected in the cyclic register selection registers 525 to 540 (16400 to 16415 in MacTalk on MAC G3). The CiA402 drive profile on EtherCAT and the PROFIdrive module on PROFINET use their own cyclic data instead. The chapter for each protocol describes how the data is used.

10.3.1 EtherCAT

Table 87: EtherCAT data

Item	Value
Device type	EtherCAT slave, CANopen over EtherCAT
Profiles	JVL register profile, CiA402
CiA402 drive profile	MAC G3 only (beta)
Cyclic data	8 read + 8 write registers, 32-bit
Min. cycle time	500 us (object 0x1C32:05)
Min. cycle, CiA402	1 ms (MAC G3)
Synchronisation	Free run or Distributed Clocks (SYNC0)
Sync managers	4; SM0/SM1 mailbox, 128 bytes each
Mailbox protocols	CoE (SDO, emergency), EoE
FoE, bootstrap	Not supported
Vendor ID	0x00000117

The minimum cycle time is the value the motor reports in object 0x1C32. The CiA402 drive profile on the MAC G3 sets the motor cycle time to 1 ms. See Distributed Clocks in the EtherCAT chapter.

10.3.2 EtherNet/IP

Table 88: EtherNet/IP data

Item	Value
Device type	EtherNet/IP adapter (CIP)
Cyclic data	Assembly 100 (write), 101 (read)
Assembly size	8 registers of 32 bits each
I/O connection types	Exclusive owner, listen only, input only
RPI	1 ms to 1 s
I/O connections	1 exclusive owner, 2 listen only, 2 input only
IP configuration	Static or DHCP
Address conflict detect.	Supported, enabled by default
JVL objects	Class 0x64; 0x65 interface (MAC G3)

The requested packet interval (RPI) must be achievable for the whole network, not only for one motor - see Ethernet switches in the EtherNet/IP chapter.

[EDITORIAL NOTE - NOT FOR RELEASE] ADO #2158 CLF

10.3.3 Modbus TCP/UDP

Table 89: Modbus TCP/UDP data

Item	Value
Role	Server
Port	502, TCP and UDP
TCP connections	Max. 8
Function codes	0x03, 0x06, 0x10, 0x17, 0x2B
Registers per frame	Max. 124 Modbus registers (62 x 32-bit)
Addressing	Even address and even count
Default Modbus address	254

Each 32-bit motor register occupies two consecutive Modbus registers, so one frame carries up to 62 motor registers. See Register access in the Modbus TCP/UDP chapter.

10.3.4 POWERLINK

Table 90: POWERLINK data

Item	Value
Device type	Controlled node (CN), POWERLINK V2
Transmission	100 Mbit/s, half duplex
Cyclic data	8 read + 8 write registers, 32-bit
Default cycle time	1000 us (object 0x1006)
Min. cycle time	400 us (XDD)
Node ID	1 to 239
Acyclic data	SDO over ASnd and UDP (port 3819)
SDO connections	Max. 5

The Node ID is taken from the last octet of the IP address, as described in the POWERLINK chapter. The minimum cycle time is the value declared in the XDD file.

10.3.5 PROFINET

Table 91: PROFINET data

Item	Value
Device type	PROFINET IO device
RT classes, MAC G3	RT_CLASS_1 (RT), RT_CLASS_3 (IRT)
RT classes, MIS G3	RT_CLASS_1 (RT)
Conformance class	C (MAC G3), B (MIS G3)
Netload class	III
Min. device interval	1 ms
Send clock, MAC G3	1 ms, 2 ms
Send clock, MIS G3	0.5 ms, 1 ms, 2 ms, 4 ms
Cyclic data	JVL register module, 8 + 8 registers
PROFIdrive	Telegrams 1, 2 and 3 (MAC G3 only)
Max. I/O data	64 bytes in, 64 bytes out
Protocols	DCP, LLDP, SNMP, MRP client
Address assignment	DCP or local

The minimum device interval is 1 ms on both products. The data is taken from the GSDML files, which differ between the two products: only the MAC G3 file offers IRT

and the PROFIdrive module.

10.4 User I/O

The user I/O belongs to the motor, not to the Ethernet interface, and is read and written over any of the protocols through the motor registers. The data below is taken from the motor manuals: for the MAC G3 from User I/O and Technical Data in LB0067, and for the MIS G3 from User Inputs, User Outputs and Technical Data in LB0076.

Table 92: User I/O

Item	MAC G3	MIS G3
Number of I/O	Up to 8 (by type number)	8 (by type number)
I/O function	Input or output	Input, output; IO1-4 analogue
Analogue inputs	Up to 4, 0-10 VDC (0-5 VDC on older hardware)	4 (IO1-4), 0-5 VDC
ADC resolution	12 bit	12 bit
Input voltage	Up to 30 VDC	0-30 VDC
Input levels	Low max. 3.3 V, high min. 6.6 V	Low max. 1.2 V, high min. 3.0 V
Input impedance	30 kOhm to 10 V, 1 mA at 24 V	30 kOhm to 5 V, 2 mA at 24 V
Output type	Source driver	Source driver
Output supply	Approx. 24 VDC, regulated from CVI	8-30 VDC, from CVI
Output current	400 mA per group of 4, 0.8 A total	400 mA per group of 4, 0.8 A total
Short circuit	Current limit, then output off until switched off and on. Not reported. (OUT1 8-pin: restarts by itself)	Current limit, then output off until switched off and on. Not reported.
Galvanic isolation	No	No

On the MAC G3 the input levels apply to the hardware with the 0-10 V input divider (PCB version 0.71 or later with the internal flex cables WF0851-15 / WF0855-12); a 5 V TTL signal is not enough for logic high there. The 0-10 V analogue range of that hardware is not yet supported by the firmware. How to identify the hardware from Register 251 is described in the MAC G3 user manual LB0067. Which I/O a motor has depends on the connector option in the type number. On the MAC G3 the 8-pin I/O connector gives 3 inputs and 1 output, and the dedicated OUT1 on that connector is limited to 300 mA; the 17-pin connector gives up to 8 I/O. See How to see which I/O options exist in LB0067 and Connector overview for the MIS motors in LB0076.

On the MIS G3 the outputs are supplied from the control voltage (CVI). On the MAC G3 they are supplied from CVI via an internal regulator of approx. 24 VDC, which is fully overload protected. Do not connect more than 32 VDC (MAC G3) or 30 VDC (MIS G3) to the CVI terminal, as the output circuitry may be damaged. On both products each group of four outputs (IO1-4 and IO5-8) can supply max. 400 mA, and all outputs together max. 0.8 A; on the MIS G3 this is shared with the load on the CVO terminal.