Myostat Motion Control Inc

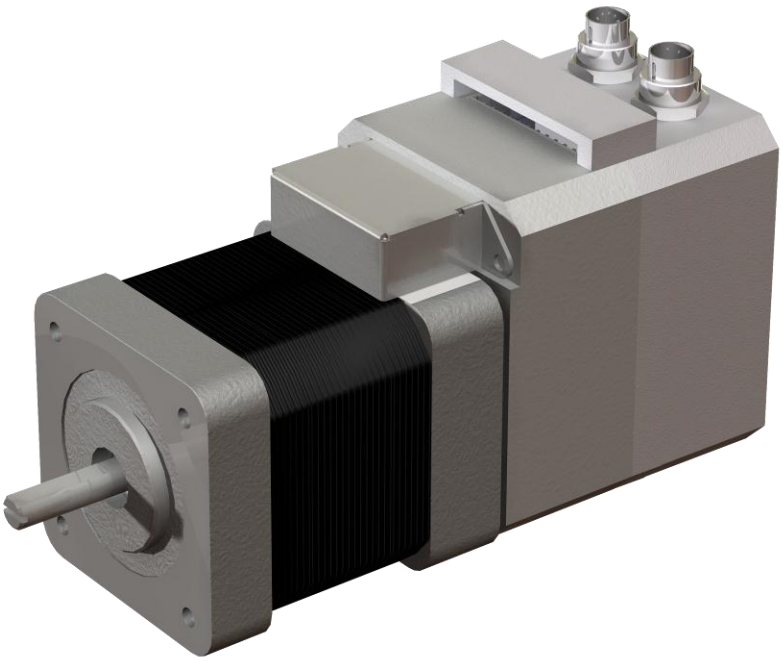
Cool Muscle

CANopen Interface Manual

CiA 301 – application layer and communication profile

CiA 402 – drives and motion control device profile

Document Version 1.07



Myostat Motion Control Inc
+1 905 836-4441
www.coolmuscle.com

1 Table of Contents

2 Introduction

This documentation assumes knowledge of the CANopen protocol or at least access to CiA 301 documentation. The CANopen model has the following general structure:

- Communication – this layer provides the basic function of transporting data on the underlying network structure.
- Object Dictionary – a collection of all the data items which have an influence on the behavior of the application objects, the communication objects and the state machine used on the device.
- Application – how the application interacts with its process environment

The Cool Muscle CANopen implementation follows the CiA 301 communication profile and the CiA 402 device profile for motors and control systems. A subset of Cia402 is implemented with all required dictionary objects. Please see the object list in this document listing all implemented objects. Further objects will be added. Please speak to your local distributor if you have an immediate request.

3 Object Dictionary

The following table lists all available object entries. Complete descriptions are available in their relevant sections

Table 1: Object dictionary overview

Address	Index	Name	Data Type	Value Description
1000_h	0	Device Type	UNSIGNED32	Device type and profile (402)
1001_h	0	Error Register	UNSIGNED 8	Predefined error message
1005_h	0	SYNC COB-ID	UNSIGNED32	Synchronization object identifier
1008_h	0	Manufacturer device name	CONST	Manufacturers application device name
1009_h	0	Manufacturer hardware version	CONST	Man. application hardware version
$100A_h$	0	Manufacturer software version	CONST	Man. application software version
$100C_h$	0	Guard Time	UNSIGNED16	Time span for node guarding
$100D_h$	0	Life Time Factor	UNSIGNED8	Life time factor for node guarding protocol
1010_h	1	Save all parameters	“save”	Save all the parameters to the EEPROM
	2	Save comm. parameters	“save”	Save only communication parameters
	3	Save application parameters	“save”	Save only application parameters
1011_h	1	Load all defaults	“load”	Load all parameters to the defaults
	2	Load comm. defaults	“load	Load only communication parameters
	3	Load application defaults	“load”	Load only application parameters
1017_h	0	Heartbeat	UNSIGNED16	Producer heartbeat in ms
1018_h	1	Vendor ID	CONST	Manufacturers name
	2	Product Code	CONST	Product name
	3	Rev Number	CONST	Revision number of the drive
	4	Serial Number	CONST	Serial number of the drive
1200_h	1	SDO1 rx COB-ID	UNSIGNED32	Identifier for receive SDO (R_SDO)
	2	SDO1 tx COB-ID	UNSIGNED32	Identifier for transmit SDO (T_SDO)
	3	Node ID	1h-7Fh	Node ID of the SDO Client
1400_h	0	1st receive PDO parameter	-	R_PDO1 settings
	1	PDO1 rx COB-ID	UNSIGNED32	Identifier for R_PDO1
	2	Transmission type	CONST	Transmission type for R_PDO1 (255)
1600_h	0	1st receive PDO mapping	-	R_PDO1 mapping settings
	1	1st PDO mapping	CONST	R_PDO1 Object ID map
1800_h	0	1st transmit PDO parameter	-	T_PDO1 settings
	1	PDO1 tx COB-ID	UNSIGNED32	Identifier for T_PDO1
	2	Transmit type	UNSIGNED8	Transmission type for R_PDO1
$1A00_h$	0	1st transmit PDO mapping	-	T_PDO1 mapping settings
	1	1st PDO tx mapping	UNSIGNED32	T_PDO1 Object ID map
2000_h	0	Homing direction	UNSIGNED8	Direction on homing profile
2001_h	1	Homing current	UNSIGNED8	Homing current limit for hard stop (%)
6040_h	0	Controlword	UNSIGNED16	Control for changing operating status
6041_h	0	Statusword	UNSIGNED16	Word to evaluate operating status
6060_h	0	Modes of Operation	Integer8	Set operating mode

$607A_h$	0	Target Position	Integer32	Position profile target position
$607C_h$	0	Home Offset	Integer32	Homing profile home offset
6081_h	0	Profile Velocity	Integer32	Maximum target velocity
6083_h	0	Profile Acceleration	Integer32	Maximum target acceleration
6098_h	0	Homing Method	Integer8	Type of homing method used
6099_h	0	Homing Speed Index	CONST 02H	Total indices in 6099
	1	Homing Speed	Integer32	Maximum homing speed
	2	Homing Speed return (N/A)	CONST 00H	Not applicable in the Cool Muscle
$609A_h$	0	Homing Acceleration	Integer32	Maximum homing acceleration
6402_h	0	Motor Type	UNSIGNED16	Sinusoidal PM BL motor (0xA)
6502_h	0	Supported drive modes	UNSIGNED32	Drive modes supported by the device

4 General Parameters

4.1 Object definitions

4.1.1 Object 1000_h: Device Type and Profile

Device type and profile detail is returned. Device type is indicated in the most significant 16 bits (0002) where 2 indicates a servo drive. The least significant 16 bits are set to 402 showing the profile implemented (CiA402).

Table 2: Object description

Attribute	Value
Index	1000_h
Name	Device Type
Object Code	Variable
Data Type	UNSIGNED32

Table 3: Entry description

Attribute	Value
Sub-Index	00_h
Access	ro
PDO mapping	None
Value range	UNSIGNED32
Default Value	$0002\ 0192_h$ ($131702_d + 402_d$)

4.1.2 Object 1001_h: Error Register

This object indicates whether the device is in an error state of not. Bit 0 of the Object shows the device is in a general error state if set to 1.

Table 4: Object description

Attribute	Value
Index	1001_h
Name	Error register
Object Code	Variable
Data Type	UNSIGNED8

Table 5: Entry description

Attribute	Value
Sub-Index	00_h
Access	r
PDO mapping	None
Value range	-
Default Value	-

4.1.3 Object 1005_h: Synchronization COB-ID

Object 1005_h defines the COB-ID of the synchronization (SYNC) object. Further it defines whether the object sends or receives messages.

	MSB				LSB
bits	31	30	29	28-11	10-0
11-bit ID	x	0/1	0	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	11-bit Identifier
29-bit-ID	x	0/1	1	29-bit Identifier	

Figure 1: SYNC COB-ID entry structure

Table 6: Read access values

Bit number	value	meaning
31 (MSB)	X	Do not care
30	0 1	Device does not generate SYNC message Device generates SYNC message
29	0 1	11-bit ID (CAN 2.0A) 29-bit ID (CAN 2.0B)
28-11	0 X	If bit 29=0 If bit 29=1: bits 28-11 of 29-bit-SYNC-COB-ID
10-0 (LSB)	X	Bits 10-0 of SYNC-COB-ID

Table 7: Object description

Attribute	Value
Index	1005_h
Name	COB-ID SYNC message
Object Code	Variable
Data Type	UNSIGNED32

Table 8: Entry description

Attribute	Value
Sub-Index	00_h
Access	rw
PDO mapping	None
Value range	-
Default Value	-

4.1.4 Object 1008_h: Manufacturer device name

Contains the Cool Muscle device name (i.e. CM1 or CM2).

Table 9: Object description

Attribute	Value
Index	1008_h
Name	Manufacturer device name
Object Code	Variable
Data Type	Visible String

Table 10: Entry description

Attribute	Value
Sub-Index	00_h
Access	const
PDO mapping	None
Value range	-
Default Value	“CM1”

4.1.5 Object 1009_h: Manufacturer hardware version

Contains the Cool Muscle motor’s hardware version hardware device name.

Table 11: Object description

Attribute	Value
Index	1009_h
Name	Manufacturer hardware version
Object Code	Variable
Data Type	Visible String

Table 12: Entry description

Attribute	Value
Sub-Index	00_h
Access	const
PDO mapping	None
Value range	-
Default Value	“200-6”

4.1.6 Object $100A_h$: Manufacturer software version

Contains the Cool Muscle motor’s firmware version

Table 13: Object description

Attribute	Value
Index	$100A_h$
Name	Manufacturer software version
Object Code	Variable
Data Type	Visible String

Table 14: Entry description

Attribute	Value
Sub-Index	00_h
Access	const
PDO mapping	None
Value range	-
Default Value	“3.08”

4.1.7 Object $100C_h$: Guard time

Objects $100C_h$ and $100D_h$ support the guard time and lifetime factors. The lifetime multiplied by the guard time gives the Life Guarding Protocol.

The guard time is given in milliseconds. It is set to 0 if not used. Either guarding or heartbeat (1017_h) must be implemented.

Table 15: Object description

Attribute	Value
Index	$100C_h$
Name	Guard Time
Object Code	Variable
Data Type	UNSIGNED16

Table 16: Entry description

Attribute	Value
Sub-Index	00_h
Access	rw
PDO mapping	None
Value range	UNSIGNED16
Default Value	0

4.1.8 Object $100D_h$: Life time factor

Objects $100C_h$ and $100D_h$ support the guard time and lifetime factors. The lifetime multiplied by the guard time gives the Life Guarding Protocol.

Table 17: Object description

Attribute	Value
Index	$100D_h$
Name	Life Time Factor
Object Code	Variable
Data Type	UNSIGNED8

Table 18: Entry description

Attribute	Value
Sub-Index	00_h
Access	rw
PDO mapping	None
Value range	UNSIGNED8
Default Value	0

4.1.9 Object 1010_h: Store parameters

This object supports the saving of parameters in non volatile memory. By read access the device provides information about its saving capabilities. Several parameter groups are distinguished:
Sub-Index 0 contains the largest Sub-Index that is supported.
Sub-Index 1 refers to all parameters that can be stored on the device.
Sub-Index 2 refers to communication related parameters (1000h - 1FFF).
Sub-Index 3 refers to application related parameters (Index 6000h - 9FFFh manufacturer specific application parameters).

In order to avoid storage of parameters by mistake, storage is only executed when a specific signature is written to the appropriate sub-index. The signature is “save“.

MSB			LSB
e	v	a	s
65_h	76_h	61_h	73_h

Figure 2: Storage write access signature

On read access the response to the appropriate sub-index is as follows

UNSIGNED 32		
MSB		LSB
31-2	1	0
reserved (=0)	0/1	0/1

Figure 3: Storage read access structure

Table 19: Read access values

Bit number	value	meaning
31-2	0	Reserved
1	0	Motor does not save parameters automatically
0	1	Motor save parameters on command

Table 20: object description

Attribute	Value
Index	1010_h
Name	Store parameters
Object Code	Array
Data Type	UNSIGNED32

Table 21: Entry description

Attribute	Value
Sub-Index	00_h
Description	Highest sub-index supported
Access	ro
PDO mapping	None
Value range	01_h – $7F_h$

Sub-Index	01_h
Description	Save all parameters
Access	rw
PDO Mapping	None
Value Range	UNSIGNED32 – see Figure 2 and Figure 3

Sub-Index	02_h
Description	Save communication parameters
Access	rw
PDO Mapping	None
Value Range	UNSIGNED32 – see Figure 2 and Figure 3

Sub-Index	03_h
Description	Save application parameters
Access	rw
PDO Mapping	none
Value Range	UNSIGNED32 – see Figure 2 and Figure 3

4.1.10 Object 1011_h: Restore default parameters

This object restores parameters back to their defaults. As with object 1010_h, depending on the sub-index referenced a different set of parameters is stored.

Sub-index 0: contains largest sub-index that is supported.
Sub-Index 1: restore all parameters to default.
Sub-index 2: restore communication parameters (1000_h - 1FFF).
Sub-index 3: restore application related parameters (6000_h-$9FFF_h$).

MSB			LSB
d	a	o	l
64_h	61_h	$6F_h$	$6C_h$

Figure 4: Restoring write access signature

To avoid restoring defaults by mistake, restoring is only executed when “load” is written to the appropriate sub-index. The default parameters are only valid when the motor is power-cycled.

On read access the response to the appropriate sub-index is as follows

UNSIGNED 32

MSB	LSB
31-1	0
reserved (=0)	0/1

Figure 5: Storage read access structure

Table 22: Read access values

Bit number	value	meaning
31-1	0	Reserved
0	1	Motor loads default parameters on command

Table 23: object description

Attribute	Value
Index	1011_h
Name	Restore default parameters
Object Code	Array
Data Type	UNSIGNED32

Table 24: Entry description

Attribute	Value
Sub-Index	00_h
Description	Highest sub-index supported
Access	ro
PDO mapping	None
Value range	01_h – $7F_h$

Sub-Index	01_h
Description	Restore all default parameters
Access	rw
PDO Mapping	None
Value Range	UNSIGNED32 – see Figure 4: Restoring write access signature and Figure 5

Sub-Index	02_h
Description	Restore default comm. parameters
Access	rw
PDO Mapping	None
Value Range	UNSIGNED32 – see Figure 4: Restoring write access signature and Figure 5

Sub-Index	03_h
Description	Restore default app. parameters
Access	rw

PDO Mapping	None
Value Range	UNSIGNED32 – see Figure 4: Restoring write access signature and Figure 5

4.1.11 Object 1017_h: Producer Heartbeat Time

Object 1017_h defines the cycle time of the heartbeat in milliseconds. It is set to 0 if not used. Either guarding ($100C_h$) or heartbeat must be implemented.

Table 25: Object description

Attribute	Value
Index	1017_h
Name	Producer Heartbeat Time
Object Code	Variable
Data Type	UNSIGNED16

Table 26: Entry description

Attribute	Value
Sub-Index	00_h
Access	rw
PDO mapping	None
Value range	UNSIGNED16
Default Value	0

4.1.12 Object 1018_h: Identity Object

The identity object contains general information about the device.

Sub-index 1_h: Vendor ID – a unique value allocated to each manufacturer
Sub-index 2_h: Manufacturer specific product code
Sub-index 3_h: Manufacturer specific revision number consisting of a major revision and a minor revision. The major revision number increments if functionality is expanded. The minor number increments for updates with the same CANopen behavior.
Sub-index 4h: Serial number.

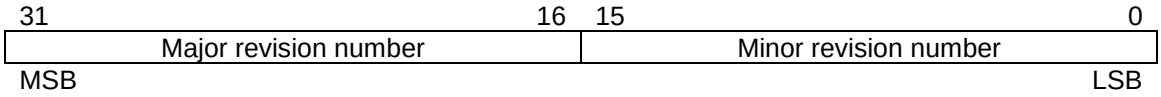


Figure 6: Revision number structure

Table 27: Object description

Attribute	Value
Index	1018_h
Name	Identity Object
Object Code	Record
Data Type	Identity

Table 28: Entry description

Attribute	Value
Sub-Index	00_h
Description	Number of entries
Access	ro
PDO mapping	None

Value range	4
Default Value	4

Sub-Index	01_h
Description	Vendor ID
Access	ro
PDO mapping	None
Value range	UNSIGNED32
Default Value	0

Sub-Index	02_h
Description	Product code
Access	ro
PDO mapping	None
Value range	UNSIGNED32
Default Value	No

Sub-Index	03_h
Description	Revision Number
Access	ro
PDO mapping	None
Value range	UNSIGNED32
Default Value	No

Sub-Index	04_h
Description	Serial number
Access	rw
PDO mapping	None
Value range	UNSIGNED32
Default Value	0

4.1.13 Object 6402_h: Motor Type

This object defines the motor type driven by the drive device. Please note that both CM1 and CM2 are closed loop fully integrated motors. Type $0A_h$ (AC Synchronous PM BL) is indicated by this object as it generally covers the functionality and performance of the Cool Muscle range.

Table 29: Object description

Attribute	Value
Index	6402_h
Name	Motor Type
Object Code	Variable
Data Type	UNSIGNED16

Table 30: Entry description

Attribute	Value
Sub-Index	00_h
Access	ro
PDO mapping	None
Value range	None
Default Value	$000A_h$

5 Communication Parameters

The communication objects define the CAN Object ID's and their mapping functions

5.1 Object definitions

5.1.1 Object 1200_h: Server SDO parameter

This object indicates the number of server SDOs implemented and the COB_ID's of R_SDO and T_SDO. The Node-ID is read/write and can be changed through this object (sub-index 03_h). Optionally the Node-ID is changeable through the CM1-CAN configuration utility.

	MSB				LSB
bits	31	30	29	28-11	10-0
11-bit ID	0/1	0	0	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	11-bit Identifier
29-bit-ID	0/1	0	1	29-bit Identifier	

Figure 7: SDO COB-ID entry structure

Table 31: SDO COB-ID entry description

Bit number	value	meaning
31 (MSB)	0 1	SDO is valid SDO is not valid
30	0	Reserved (always 0)
29	0 1	11-bit ID (CAN 2.0A) 29-bit ID (CAN 2.0B)
28-11	0 X	If bit 29=0 If bit 29=1: bits 28-11 of 29-bit-SYNC-COB-ID
10-0 (LSB)	X	Bits 10-0 of SYNC-COB-ID

Table 32: object description

Attribute	Value
Index	1200_h
Name	Server SDO Parameter
Object Code	RECORD
Data Type	SDO Parameter

Table 33: Entry description

Attribute	Value
Sub-Index	00_h
Description	Highest sub-index supported
Access	ro
PDO mapping	None
Value range	-
Default Value	03h

Sub-Index	01_h
Description	COB-ID Client→Server (rx)
Access	ro
PDO mapping	None
Value range	UNSIGNED32
Default Value	600_h + Node-ID

Sub-Index	02_h
Description	COB-ID Server→Client (tx)
Access	ro

PDO mapping	None
Value range	UNSIGNED32
Default Value	580_h + Node-ID

Sub-Index	03_h
Description	Node-ID of the SDO Client
Access	rw
PDO mapping	None
Value range	1_h – $7F_h$
Default Value	No

5.1.2 Object 1400_h: Receive PDO communication parameter

Contains the COB-ID of the first receive PDO (R_PDO1). R_PDO1 is mapped to object 6040_h, the ControlWord. This is shown in 1600_h, 1st receive PDO mapping.

The second entry is the transmission type and is always set to 255, asynchronous.

	MSB				LSB
bits	31	30	29	28-11	10-0
11-bit ID	0/1	0/1	0	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	11-bit Identifier
29-bit-ID	0/1	0/1	1	29-bit Identifier	

Figure 8: R_PDO COB-ID entry structure

Table 34: PDO COB-ID entry description

Bit number	value	meaning
31 (MSB)	0 1	PDO is valid PDO is not valid
30	0 1	RTR allowed on this PDO No RTR allowed on this PDO
29	0 1	11-bit ID (CAN 2.0A) 29-bit ID (CAN 2.0B)
28-11	0 X	If bit 29=0 If bit 29=1: bits 28-11 of 29-bit-SYNC-COB-ID
10-0 (LSB)	X	Bits 10-0 of SYNC-COB-ID

Table 35: object description

Attribute	Value
Index	1400_h
Name	Receive PDO Parameter
Object Code	RECORD
Data Type	PDO CommPar

Table 36: Entry description

Attribute	Value
Sub-Index	00_h
Description	Highest sub-index supported
Access	ro
PDO mapping	None
Value range	02_h

Sub-Index	01_h
Description	COB-ID used by PDO
Access	rw
PDO mapping	None

Value range	UNSIGNED32
Default Value	200_h + Node-ID

Sub-Index	02_h
Description	Transmission Type
Access	ro
PDO mapping	None
Value range	UNSIGNED8
Default value	255 (asynchronous only)

5.1.3 Object 1600_h: Receive PDO mapping parameter

This object shows which parameters are mapped to R_PDO1. For the CM-CAN interface this is always 6040_h, the ControlWord.

Table 37: object description

Attribute	Value
Index	1600_h
Name	1^{st} receive PDO mapping
Object Code	RECORD
Data Type	PDO Mapping

Table 38: Entry description

Attribute	Value
Sub-Index	00_h
Description	Highest sub-index supported
Access	ro
PDO mapping	None
Value range	01_h

Sub-Index	01_h
Description	PDO mapping for 1^{st} mapped object
Access	ro
PDO mapping	None
Value range	-
Default Value	60400010_h

Byte: MSB		LSB
Index (16bit)	Sub-index (8 bit)	Object length (8 bit)

Figure 9: Structure of the PDO mapping entry

5.1.4 Object 1800_h: Transmit PDO communication parameter

Contains the COB-ID of the first transmit PDO (T_PDO1). R_PDO1 is mapped to object 6041_h, the StatusWord. This is shown in $1A00_h$, 1^{st} transmit PDO mapping.

The second entry is the transmission type and is always set to 255, asynchronous.

	MSB				LSB
bits	31	30	29	28-11	10-0
11-bit ID	0/1	0/1	0	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	11-bit Identifier
29-bit-ID	0/1	0/1	1	29-bit Identifier	

Figure 10: T_PDO COB-ID entry structure

Table 39: T_PDO COB-ID entry description

Bit number	value	meaning
31 (MSB)	0 1	PDO is valid PDO is not valid
30	0 1	RTR allowed on this PDO No RTR allowed on this PDO
29	0 1	11-bit ID (CAN 2.0A) 29-bit ID (CAN 2.0B)
28-11	0 X	If bit 29=0 If bit 29=1: bits 28-11 of 29-bit-SYNC-COB-ID
10-0 (LSB)	X	Bits 10-0 of SYNC-COB-ID

Table 40: object description

Attribute	Value
Index	1800_h
Name	Transmit PDO Parameter
Object Code	RECORD
Data Type	PDO CommPar

Table 41: Entry description

Attribute	Value
Sub-Index	00_h
Description	Highest sub-index supported
Access	ro
PDO mapping	None
Value range	02_h

Sub-Index	01_h
Description	COB-ID used by PDO
Access	rw
PDO mapping	None
Value range	UNSIGNED32
Default Value	180_h + Node-ID

Sub-Index	02_h
Description	Transmission Type
Access	ro
PDO mapping	None
Value range	UNSIGNED8
Default value	255 (asynchronous only)

5.1.5 Object $1A00_h$: Transmit PDO mapping parameter

This object contains the transmit PDO mapping parameters. In the case of the CM-CAN interface, T_PDO1 transmits object 6041_h.

Table 42: object description

Attribute	Value
Index	$1A00_h$
Name	1^{st} transmit PDO mapping
Object Code	RECORD
Data Type	PDO Mapping

Table 43: Entry description

Attribute	Value

Sub-Index	00_h
Description	Highest sub-index supported
Access	ro
PDO mapping	None
Value range	01_h

Sub-Index	01_h
Description	PDO mapping for 1st mapped object
Access	ro
PDO mapping	None
Value range	-
Default Value	60410010_h

Byte: MSB LSB

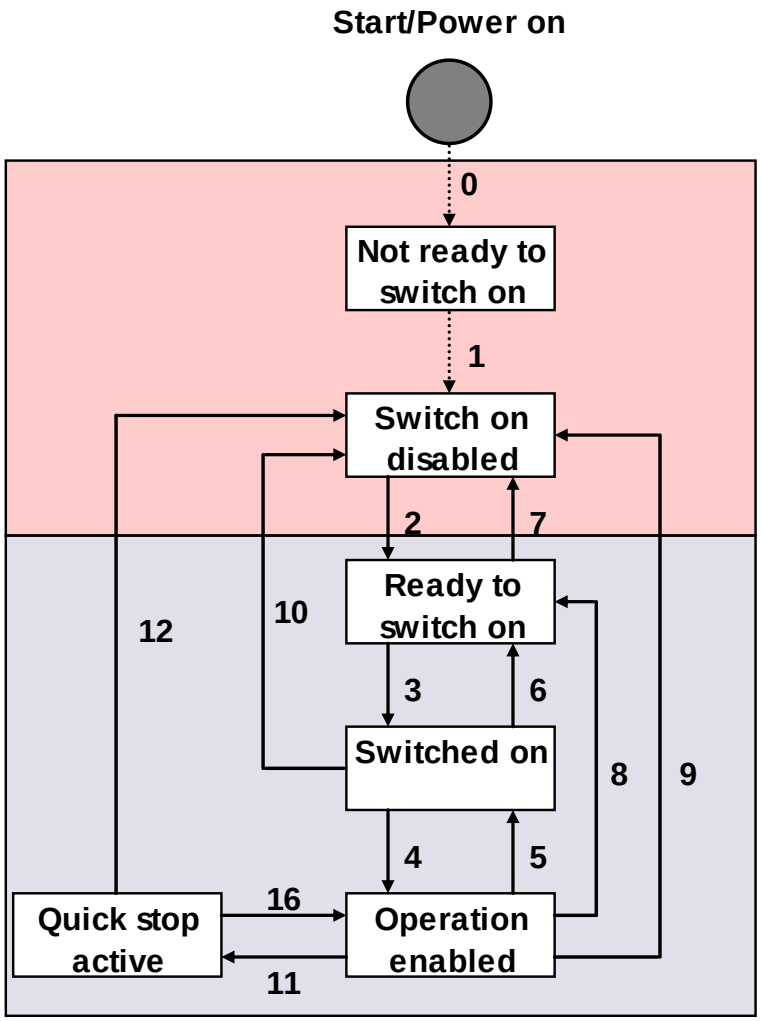
Index (16bit)	Sub-index (8 bit)	Object length (8 bit)

Figure 11: Structure of the PDO mapping entry

6 Controlling the power drive system

6.1 Finite State Automation

The following finite state machine defines how the drive system is to operate. The Control Word (6040_h) is used to transitions between states. The model is known as the Power Drive System Finite State Automation (PDS FSA).



Note:

- Drive power off
- Drive power on
- On Start/Power on transitions 0→1→2 are automatic and FSM will be in state “Ready to switch on”

6.2 Modes of operation

The current implementation of CAN open utilizes 2 modes of operation 1) Profile position mode and 2) Homing mode. The mode of operation is configured in object 6060_h.

6.3 Object definitions

6.3.1 Object 6040_h: Controlword

This object receives the command controlling the PDS FSA. Bits 3,2,1 and 0 control the state machine. Additional bits are used to start and halt the current mode of operation.

15 11	10	9	8	7	6 4	3	2	1	0
ms	r	oms	h	fr	oms	eo	qs	ev	so

ms = manufacturer-specific; r = reserved; oms = operation mode specific; h = halt; fr = fault reset; eo = enable operation; qs = quick stop; ev = enable voltage; so = switch on

Table 44: Command coding

PDS FSA State	Bits of the controlword				Transitions available from this state
	Bit 3 (eo)	Bit 2 (qs)	Bit 1 (ev)	Bit 0 (so)	
Switch on disabled	x	x	0	x	2
Ready to switch on	0	1	1	0	3,4*,7
Switched on	0	1	1	1	4,6,10
Operation Enabled	1	1	1	1	5,8,9,11
Quick stop active	x	0	1	x	12,16

*Note: The FSM can move from “Ready to switch on” to Operation enabled” directly.

Operation mode specific bits’ configurations can be found in the relative operation mode section e.g. when using profile position mode see section 7.2.

Table 45: Object description

Attribute	Value
Index	6040_h
Name	Controlword
Object Code	Variable
Data Type	UNSIGNED16

Table 46: Entry description

Attribute	Value
Sub-Index	00_h
Access	rw
PDO mapping	yes
Value range	See (see PDO mapping)
Default Value	0

6.3.2 Object 6041_h: Statusword

This object provides the status of the PDS FSA.

15 14	13 12	11	10	9	8	7	6	5	4	3	2	1	0
ms	oms	ila	tr	rm	ms	w	sod	qs	ve	f	oe	so	rtso

ms = manufacturer-specific; oms = operation mode specific; ila = internal limit active; tr = target reached; rm = remote; w = warning; sod = switch on disabled; qs = quick stop; ve = voltage enabled; f = fault; oe = operation enabled; so = switched on; rtso = ready to switch on

PDS FSA State	Status word (binary)
Switch on disabled	xxxx xxxx x1xx 0000_b
Ready to switch on	xxxx xxxx x01x 0001_b

Switched on	xxxx xxxx x011 0011_b
Operation enabled	xxxx xxxx x011 0111_b
Quick stop active	xxxx xxxx x001 0111_b

Bit 4 = 1 – indicates that high voltage is applied to the PDS
Bit 5 = 0 – indicates the PDS is reacting to a quick stop request
Bit 9 = 1 – indicates the controlword is processed. Bit 9 = 0 – indicates the controlword is not processed
Bit 10 = 1 – indicates the PDS has reached its set point/target position

Table 47: Object description

Attribute	Value
Index	6041_h
Name	Statusword
Object Code	Variable
Data Type	UNSIGNED16

Table 48: Entry description

Attribute	Value
Sub-Index	00_h
Access	rw
PDO mapping	Yes (see PDO Mapping)
Value range	See
Default Value	0

6.3.3 Object 6060_h: Modes of operation

Indicates the current mode of operation selected. The following modes are currently implemented in the Cool Muscle.

Table 49: value definition

Value	Definition
0	No mode change/no mode assigned
1	Profile position mode
6	Homing mode

Table 50: Object description

Attribute	Value
Index	6060_h
Name	Modes of operation
Object Code	Variable
Data Type	Integer8

Table 51: Entry description

Attribute	Value
Sub-Index	00_h
Access	rw
PDO mapping	none
Value range	See Table 49
Default Value	0

6.3.4 Object 6502_h: Supported drive modes

Object 6502_h indicates the supported drives modes. In the current version only Profile Positioning and Homing mode are available. Further modes will be supported. Please contact your distributor if you have drive mode requests. The figure below shows possible modes.

31 16	15 10	9	8	7	6	5	4	3	2	1	0
Manufacturer-specific	r(eserved)	cst	csv	csp	ip	hm	r	tq	pv	vl	pp
0	0	0	0	0	0	1	0	0	0	0	1
MSB											LSB

Figure 12: Value definition

r – reserved
cst – Cyclic synchronous torque mode
csv – Cyclic synchronous velocity mode
csp – Cyclic synchronous position mode
ip – Interpolated position mode
hm – Homing mode
tq – Profile torque mode
pv – Profile velocity
vl – Velocity mode
pp – Profile position mode

1 = mode is supported
0 = mode is not supported

Table 52: Object description

Attribute	Value
Index	6083_h
Name	Profile acceleration
Object Code	Variable
Data Type	UNSIGNED32

Table 53: Entry description

Attribute	Value
Sub-Index	00_h
Access	rw
PDO mapping	none
Value range	UNSIGNED32
Default Value	21_h

7 Profile position mode

7.1 General description

Profile position mode uses a set point to control the target position. The current implementation uses a single set point. The next set point can be set but only executed when the current set point is completed.

7.2 Use of controlword (6040_h) and statusword (6041_h)

7.2.1 Controlword

15 10	9	8	7	6	5	4	3 0
See 6.3.1	Change on set-point	Halt	See 6.3.1	Abs/rel	Change set immediately	New set-point	See 6.3.1

Figure 13: Controlword for profile position mode

Table 54: Definition of bit 4, bit 5, and bit 9

Bit 9	Bit 5	Bit 4	Definition
0	0	0 → 1	Positioning shall be completed (target reached) before the next one gets started

Table 55: Definition of bit 6 and bit 8

Bit	Value	Definition
6	0	Target position shall be an absolute value
8	0	Positioning shall be executed or continued
	1	Axis shall be stopped/halted

7.2.2 Statusword

15 14	13	12	11	10	9 0
See 6.3.2	Following error (not implemented)	Set-point acknowledged	See 6.3.2	Target reached	See 6.3.2

Figure 14: Statusword for profile position mode

Table 56: Definition of bit 10 and bit 12

Bit	Value	Definition
10	0	Halt (Bit 8 in controlword) = 0: Target position not reached Halt (Bit 8 in controlword) = 1: Axis is decelerating
	1	Halt (Bit 8 in controlword) = 0: Target position reached Halt (Bit 8 in controlword) = 1: Velocity of axis is 0
12	0	Positioning shall be executed or continued
	1	Axis shall be stopped/halted

7.3 Object definitions

7.3.1 Object $607A_h$: Target position

Indicates the commanded position that the drive should move to in position profile mode using the current settings of motion control parameters such as velocity, acceleration, deceleration, motion profile type etc.

Units are in pulses relative to the motor resolution which by default is 1000.

Table 57: Object description

Attribute	Value
Index	$607A_h$
Name	Target position
Object Code	Variable
Data Type	Integer32

Table 58: Entry description

Attribute	Value
Sub-Index	00_h
Access	rw
PDO mapping	none
Value range	Integer32
Default Value	0

7.3.2 Object 6081_h: Profile velocity

Indicates the configured velocity normally attained at the end of the acceleration ramp during a profiled motion and shall be valid for both directions of motion.

Units in 100p/s.

Table 59: Object description

Attribute	Value
Index	6081_h
Name	Profile velocity
Object Code	Variable
Data Type	UNSIGNED32

Table 60: Entry description

Attribute	Value
Sub-Index	00_h
Access	Rw
PDO mapping	None
Value range	UNSIGNED32
Default Value	10

7.3.3 Object 6083_h: Profile acceleration

Indicates the configured acceleration in Kp/s^2

Table 61: Object description

Attribute	Value
Index	6083_h

Name	Profile acceleration
Object Code	Variable
Data Type	UNSIGNED32

Table 62: Entry description

Attribute	Value
Sub-Index	00_h
Access	rw
PDO mapping	none
Value range	UNSIGNED32
Default Value	100

8 Homing Mode

8.1 General description

This mode describes how the motor is to seek home. The Cool Muscle implements 2 distinct methods 1) hard stopper and, 2) origin switch. They can be further configured to start on power up. The options to setup the methods are defined in the manufacturer specific region.

8.2 Use of controlword (6040_h) and statusword (6041_h)

8.2.1 Controlword

15 9	8	7	6 5	4	3 0
See 6.3.1	Halt	See 6.3.1	reserved (0)	Homing operation start	See 6.3.1

Figure 15: Controlword for homing mode

Table 63: Definition of bit 4 and bit 8

Bit	Value	Definition
4	0	Do not start homing procedure
	1	Start or continue homing mode
8	0	Enable bit 4
	1	Halt/stop homing mode

8.2.2 Statusword

15 14	13	12	11	10	9 0
See 6.3.2	Homing error	Home attained	See 6.3.2	Target reached	See 6.3.2

Figure 16: Statusword for profile position mode

Table 64: Definition of bit 10 and bit 12

Bit 13	Bit 12	Bit 10	Definition
0	0	0	Homing procedure is in progress
0	0	1	Homing procedure is interrupted or not started
0	1	0	Homing is attained, but offset is not reached
0	1	1	Homing procedure is completed successfully
1	x	x	Homing error occurred

8.3 Object definitions

8.3.1 Object $607C_h$: Home Offset

This object indicates the difference between the machine home position (found during homing) and the application zero position.

Table 65: object description

Attribute	Value
Index	$607C_h$
Name	Home Offset
Object Code	Variable
Data Type	Integer32

Table 66: Entry description

Attribute	Value
Sub-Index	00_h
Access	rw
PDO mapping	none
Value range	Integer32
Unit	100 pulses
Default Value	0_d

8.3.2 Object 6098_h: Homing method

Table 67: value definition

Value	Definition
0	No mode change/no mode assigned
-1	Hard stopper
-2	Hard stopper on power up
-3	Origin switch
-4	Origin switch on power up

Table 68: object description

Attribute	Value
Index	6098_h
Name	Homing method
Object Code	Variable
Data Type	Integer8

Table 69: Entry description

Attribute	Value
Sub-Index	00_h
Access	rw
PDO mapping	None
Value range	See Table 67
Default Value	0

8.3.3 Object 6099_h: Homing speed

Table 70: object description

Attribute	Value
Index	6099_h
Name	Homing speeds
Object Code	Array
Data Type	UNSIGNED32

Table 71: Entry description

Attribute	Value
Sub-Index	00_h
Description	Highest sub-index supported
Access	c
PDO mapping	none
Value range	01_h
Default Value	01_h
Sub-Index	01_h
Description	Speed during search for machine 0
Access	rw
PDO Mapping	none
Value Range	1 - 5000
Unit	100 p/s
Default Value	10

8.3.4 Object $609A_h$: Homing acceleration

This object indicates the configured homing acceleration and deceleration to be used during the homing operation.

Table 72: object description

Attribute	Value
Index	$609A_h$
Name	Homing acceleration
Object Code	Variable
Data Type	UNSIGNED32

Table 73: Entry description

Attribute	Value
Sub-Index	00_h
Access	rw
PDO mapping	none
Value range	1 - 5000
Unit	1 Kp/s^2
Default Value	100

8.3.5 Object 2000_h: Homing direction

This object indicates the direction the motor will initially start in when the home search is initiated.

Table 74: value definition

Value	Definition
0	Clockwise (CW)
1	Counter clock wise (CCW)

Table 75: object description

Attribute	Value
Index	2000_h
Name	Homing acceleration
Object Code	Variable
Data Type	Integer8

Table 76: Entry description

Attribute	Value
Sub-Index	00_h
Access	rw
PDO mapping	none
Value range	See Table 74
Default Value	0

8.3.6 Object 2001_h: Homing voltage level

This object specifies the rise in voltage to indicate a hard stopper has been reached when searching for a home position. It is defined as a percentage of the full voltage.

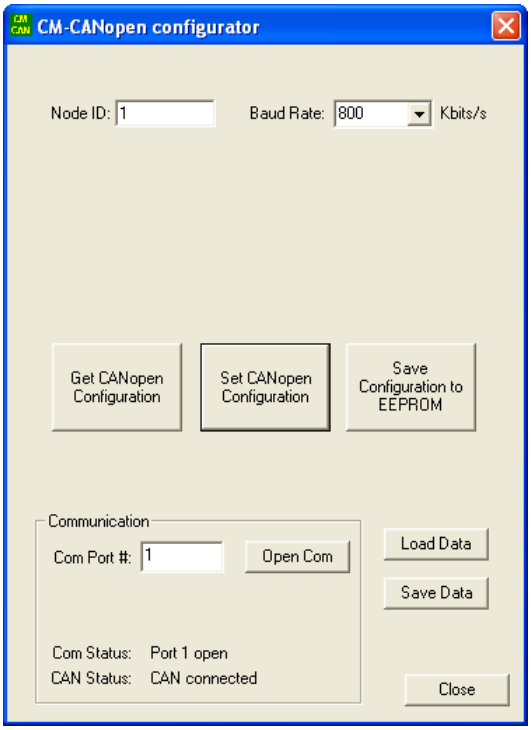
Table 77: object description

Attribute	Value
Index	2001_h
Name	Homing voltage level
Object Code	Variable
Data Type	Integer8

Table 78: Entry description

Attribute	Value
Sub-Index	00_h
Access	rw
PDO mapping	none
Value range	10-100 (%)
Default Value	30 (%)

A. Appendix A – CM-CANopen Configurator



The CM-CANopen Configurator is used for initial setup to specify certain details. Further version releases will allow most objects to initialize before the device is placed on the network.

Current configuration options:

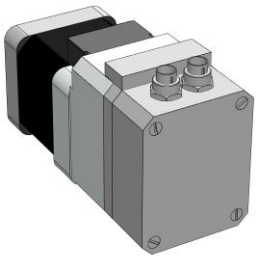
- Node ID: the CANopen Node ID of the device connected to the network
- Baud rate: the baud rate of the CANopen network

The configuration utility has a button to grab what is currently in the module or set new values in the module. If new values are set they can be saved to the EEPROM on the CANopen device so they are always initialized with the new value.

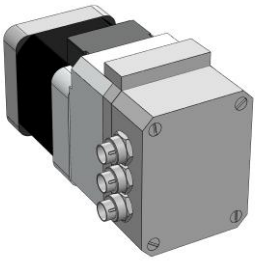
The setup data is optionally saved to/loaded from an .eds file.

The configuration utility can be downloaded from the CANopen page on www.coolmuscle.com.

B. Appendix B – Dimensioning and connector options



Option 1
M9 – Power
M9 – CANopen communication

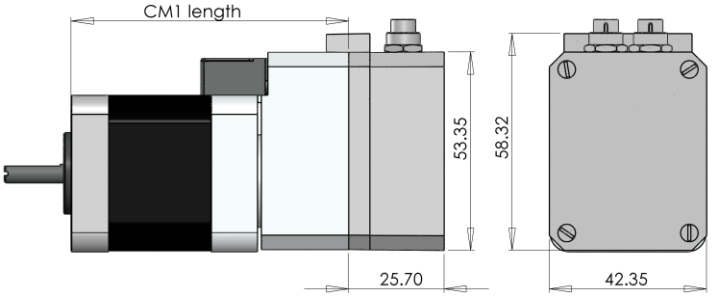


Option 2
M9 – Power
M9 – CANopen communication
M9 – I/O

Option 3
M9 – Power
DB9 – CANopen communication
DB9 – I/O

Complete dimensioning on the CANopen interface is dependent on the motor type. Dimensioning shown here indicates the additional length added to a CM1. CM1 dimensioning for the required motor can be found in the motor product brochure.

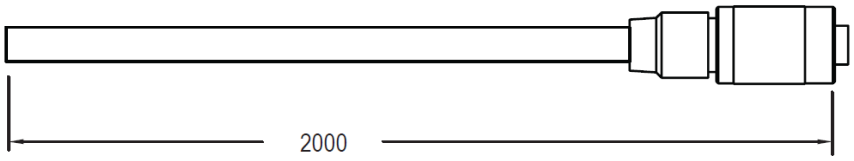
The integrated CANopen case is available on the CM1 Nema 23 and Nema 17 motors only.



C. Appendix C – Cable options

C.1 CM1M9-2F-2000

- 24VDC power cable
- 2 conductor M9 female plug
- 2 meters

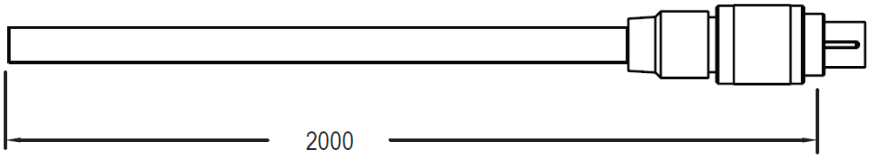


C.2 CM1M9-5F-2000

- I/O cable for CAN3 option
- 5 conductor M9 female plug
- 2 meters

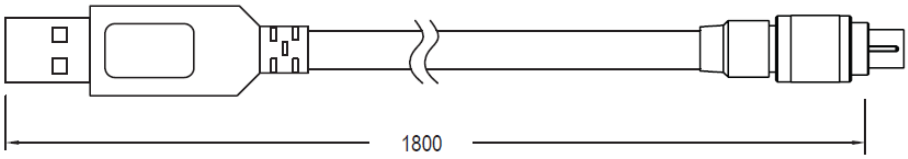
C.3 CM1M9-5M-2000

- CANopen communication cable
- 5 conductor M9 male plug cable
- 2 meters



C.4 CM1US3-1800

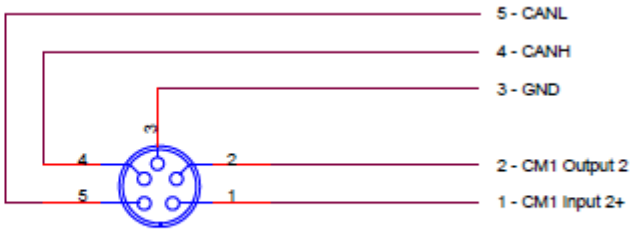
A. CANopen configuration cable
B. USB2.0 to 5 conductor M9
C. 1.8 meters



D. Connectors and Wiring

Motor side receptacles

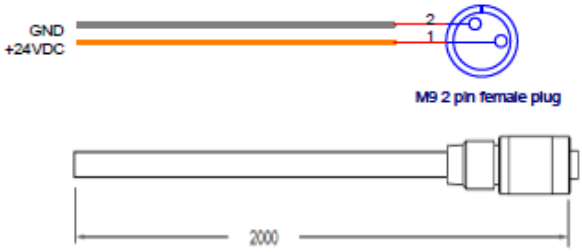
CN1 - 24VDC power
M9 2 pin male receptacle



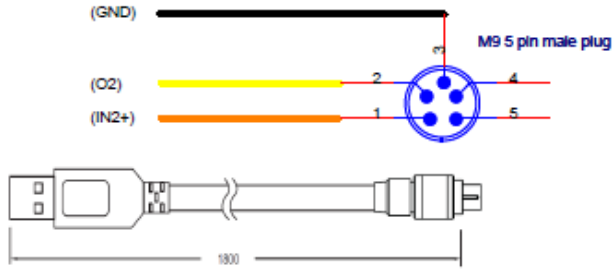
CN2 - CAN comm
M9 5 pin female receptacle

Cable side plugs

CM1M9-2F-2000 (Power cable)



CM1US3-1800 (CANopen USB configuration cable)



CM1M9-5M-2000 (CANopen communication cable)