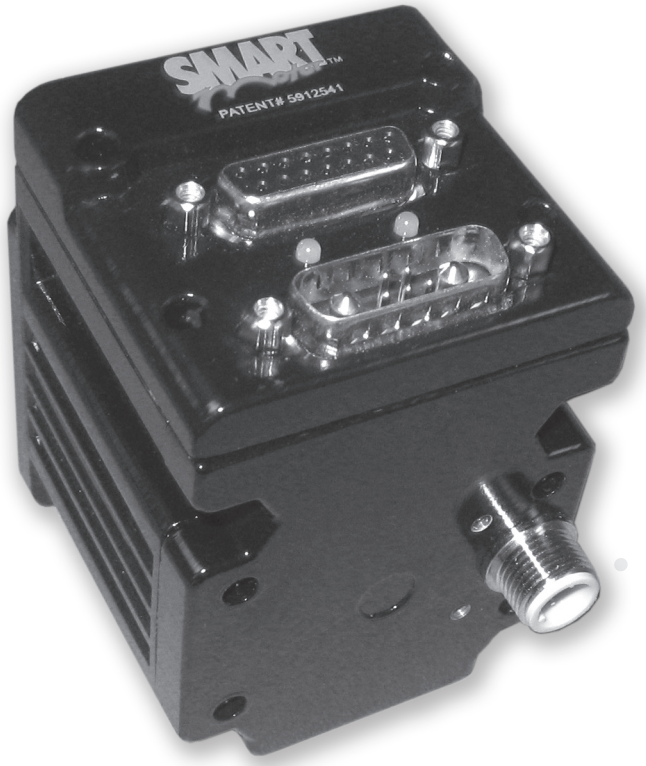
CANopen



ANIMATICS®

Motion Control Products

Defining the Future in Motion Control

About the CANopen SmartMotor™

CANopen Overview

CANopen is an open network standard that provides for reduced system complexity and significant reductions in wiring costs. CANopen allows a SmartMotor™ to respond to commands sent from a CANopen master.

CANopen is a broadcast-oriented, communications protocol developed by CiA (CAN in Automation). Please refer to http://www.can-cia.org/ for protocol specific information. With CAN, any node may transmit data if the bus is not busy. If two or more nodes begin transmitting at the same time, the message with the lowest CAN ID will complete the transmission.

A CANopen network may have up to 127 nodes, each with a unique address. CANopen supports baud rates from 20K to 1Mbps. As the baud rate increases, the maximum allowable distance of cable between any two devices decreases. The minimum cable length must be greater than 2 meters at higher baudrates.

Feature	Description
Network Size	Up to 127 nodes
Network Length	Selectable end-to-end network distance varies with speed **Baud Rate** — **Distance** 100 Kbps — 40 meters 500 Kbps — 100 meters 100 Kbps — 500 meters
Data Packets	0-8 bytes
Bus Topology	Linear (trunkline/dropline): power and signal on the same network cable
Bus Comms	Multi-Master and Master/Slave special case
System Features	Removal and replacement of devices from the network while energized

How CANopen works with our SmartMotor™

The CANopen SmartMotor™ is designed in a modular fashion, with the standard SmartMotor™ module adapted to work in conjunction with a CANopen gateway. The CANopen gateway uses a separate dedicated controller for CANopen operation, which means that varying network traffic demands will not affect the ability of the SmartMotor™ to handle motion and I/O tasks. The standard SM is equipped with two serial ports. These ports are configured such that one is RS-232 format and one is RS-485 format. On the CANopen version of the SmartMotor™ , the RS-485 port has been retained for use with the CANopen gateway, so the RS-485 port is no longer available for external use.

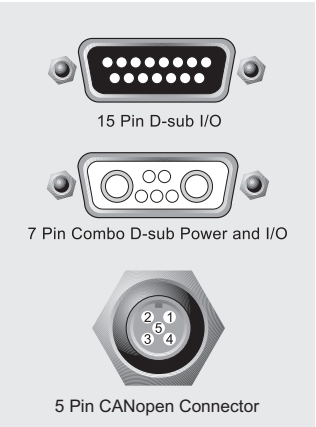
CANopen control of the SmartMotor™

To send and receive messages to and from the CANopen Smartmotor, the user must thoroughly understand the command structures used by CANopen.

The CANopen Gateway EDS and other reference files can be downloaded from the Animatics website.

Refer to the tables and examples and at the end of this document for quick reference.

CANopen



CANopen Pinout:

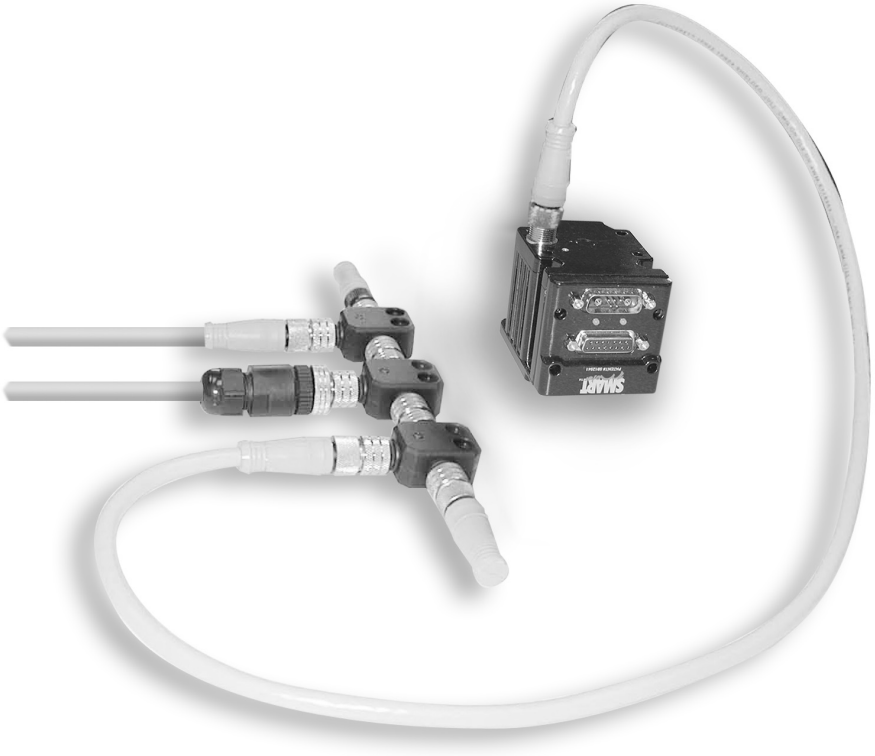
1 Not Connected
2 Not Connected
3 CAN ground
4 CAN H
5 CAN L

Note: This option DOES NOT apply to all Models

Animatics CANOpen SmartMotor™

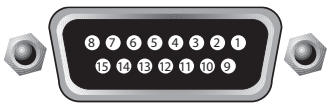
Features Include:

- All Basic Motion commands available via CiA V4.02 specification
- Ability to read/write all SmartMotor variables.
- Use of onboard I/O via CANOpen Gateway, SmartMotor program, or RS232 commands
- Ability to run 1000 SmartMotor subroutines via CANOpen
- Online diagnostics of the SmartMotors via SMI2 software and RS232 connection
- Up to 127 nodes
- 250 micro second interrupt driven subroutine with the -PLS firmware
- Gateway Baud Rates: 20K, 50K, 125K, 250K, 500K, 1Mpbs default 125Kbps

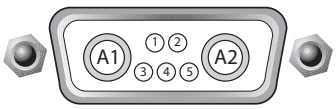


15 Pin D-Sub I/O:

1. I/O A
2. I/O B
3. I/O C
4. I/O D
5. I/O E
6. I/O F
7. I/O G
8. Encoder A Out
9. Encoder B Out
10. SM RS232 Transmit
11. SM RS232 Receive
12. +5V Out
13. Ground
14. Power Ground
15. Power



15 Pin D-sub I/O



Pin Combo D-sub Power and I/O

7 Pin Combo D-Sub Power and I/O:

A1 +20V to +48V DC
A2 Power Ground
1 Sync or I/O G
2 +5V Out
3 RS232 Transmit
4 RS232 Receive
5 RS232 Ground

Making a Velocity Move

- Send 0x3 to Index 0x6060 (Modes of Operation)
- Send desired velocity to index 0x60FF (Target Velocity)
- Send desired acceleration to Index 0x6083 (Profile Acceleration)
- Send 0x3F to Index 0x6040 (Controlword)
- Send 0x13F to Index 0x6040 (Controlword) to Halt the Move

Using the Quick Stop Option

- Send desired Deceleration to Index 0x6085 (Quick Stop Deceleration)
- Send 0xB to Index 0x6040 (Controlword)
- Note: The Quick Stop Deceleration value will be used for Accel and Decel until a value is sent to 0x6083 (Profile Acceleration).

Making an Absolute Position Move

- Send 0x1 to Index 0x6060 (Modes of Operation)
- Send desired velocity to Index 0x6081 (Profile Velocity)
- Send desired acceleration to Index 0x6083 (Profile Acceleration)
- Send absolute position value to Index 0x607A (Profiled Target Position)
- Send 0x3F to Index 0x6040 (Controlword)

Making a Relative Position Move

- Send 0x1 to Index 0x6060 (Modes of Operation)
- Send desired velocity to Index 0x6081 (Profile Velocity)
- Send desired acceleration to Index 0x6083 (Profile Acceleration)
- Send absolute position value to Index 0x607A (Profiled Target Position)
- Send 0x7F to Index 0x6040 (Controlword)

Using Torque Mode

- Send desired torque value to Index 0x6071 (Target Torque)
- Send 0x4 to Index 0x6060 (Modes of Operation)

Calling a Subroutine

- Send a value (0-999) to Index 0x2306 (Motor Subroutine Index)

Reading a SmartMotor Variable

- Send a value (0-76) to Index 0x2201,1 (User Variable Index)
- Receive a value from Index 0x2201,2 (User Variable Value)

Writing a SmartMotor Variable

- Send a value (0-76) to Index 0x2201,1 (User Variable Index)
- Send a value to Index 0x2201,2 (User Variable Value)
- Ex1. If 0x2201,1 is set to 76, the variable “yyy” can be written or read.
- Ex2. If 0x2201,1 is set to 0, the variable “a” can be written or read.

Configuring the CANopen Node through The SMI2 Software

- EPTR=32000 ‘set the EEprom pointer to 32000
- ab[0]=32 ‘load the node address into ab[0] (1-127)
- ab[1]=4 ‘set the bit rate index 4=125kbs
- VST(ab[0],2) ‘store the setup to EEprom

Bit rate index:

0	1000	K bits/sec
1	800	
2	500	
3	250	
4	125	
5	100	
6	50	
7	20	
8	10	
9	unused	

These Commands must be in the CANopen SmartMotor Program

```
OCHN(RS2,1,N,19200,1,8,C)     ‘open the RS485 port
PRINT1(“cog”,#13)             ‘signal the CANopen Gateway
                                to RUN
END
```

NOTE: Variables aaa-ggg are used by the CANopen Gateway. Do not use these variables in the SmartMotor program.

Index/SubIndex	Bytes	Read	Write	Object	Description
0x2100, 0	2	DCT	DCT	Port Configuration	Define which I/O points will be Outputs
0x2200, 1	4	DCT	DCT	Poll list entry #1	A value of -1 means this entry is blank.
0x2200, 2	4	DCT	DCT	Poll list entry #2	To add an item like Actual Velocity(0x606C)
0x2200, 3	4	DCT	DCT	Poll list entry #3	Enter 0x0000606C. This will cause Velocity
0x2200, 4	4	DCT	DCT	Poll list entry #4	To be polled constantly by the COG.
0x2201, 1	4	DCT	DCT	User Variable Index	0 is “a”, 1 is “b”… 26 is “aa” ..up thru “yyy”
0x2201, 2	4	MTR	MTR	User Variable	Value read or write to the indexed variable here
0x2202, 0	4	DCT	MTR	Set Actual Position	O=[value]
0x2300, 0	2	MTR	NAC	Bus Voltage	aaa=UJA Raaa
0x2301, 0	2	MTR	NAC	RMS Current	aaa=UIA Raaa
0x2302, 0	1	MTR	NAC	Internal Temperature	aaa=TEMP Raaa
0x2303, 0	4	MTR	MTR	Internal Clock	RCLK,CLK=[value]
0x2305, 0	2	DCT	MTR	Motor Control	See table on next page
0x2306, 0	2	DCT	MTR	Motor Subroutine Index	GOSUB[value]
0x6040, 0	2	DCT	MTR	Controlword	See table on next page
0x6041, 0	2	DCT	NAC	Statusword	See table on next page
0x6060, 0	1	DCT	DCT	Modes of operation	See table on next page
0x6063, 0	4	DCT	NAC	Actual position, internal units	RP
0x6064, 0	4	DCT	NAC	Actual position, user units	RP
0x6065, 0	4	DCT	MTR	Maximum following error	E=[value]
0x606c, 0	4	MTR	NAC	Actual velocity	RV
0x6071, 0	2	DCT	MTR	Target torque	T=[value]
0x6072, 0	2	DCT	MTR	Maximum torque	RAMPS and AMPS=[value]
0x6073, 0	2	DCT	MTR	Maximum current	AMPS=[value]
0x607a, 0	4	DCT	MTR	Profiled target position	P=[value]
0x607d, 1	4	DCT	MTR	Minimum software position limit	SLN=[value] (0 to -2147483648)
0x607d, 2	4	DCT	MTR	Maximum software position limit	SLP=[value] (0 to 2147483647)
0x607e, 0	1	DCT	DCT	Polarity	F=2 (only available in -PLS firmware)
0x607f, 0	4	DCT	MTR	Maximum profile velocity	V=[value]
0x6081, 0	4	DCT	MTR	Profile velocity	V=PVelocity
0x6083, 0	4	DCT	MTR	Profile acceleration	A=[value]
0x6085,0	2	DCT	MTR	Quick Stop Deceleration	A=[value] X
0x60f4, 0	4	MTR	NAC	Actual following error	RPE
0x60fb, 1	2	DCT	MTR	KP, Proportional Gain	KP=[value] F
0x60fb, 2	2	DCT	MTR	KI, Ingegral Gain	KI=[value] F
0x60fb, 3	2	DCT	MTR	KL, Integral Limit	KL=[value] F
0x60fb, 4	2	DCT	MTR	KD, Derivative Gain	KD=[value] F
0x60fb, 5	1	DCT	MTR	KS, Derivative Damping Sample Rate	KS=[value] F
0x60fb, 6	2	DCT	MTR	KV, Velocity Feedforward Gain	KV=[value] F
0x60fb, 7	2	DCT	MTR	KA, Acceleration Feedforward Gain	KA=[value] F
0x60fb, 8	4	DCT	MTR	KG, Gravitational Offset	KG=[value] F
0x60fd, 0	4	MTR	NAC	Digital inputs	0b01111111000000000000000 G is HiBit
0x60fe, 1	4	DCT	MTR	Digital outputs	0b01111111000000000000000 G is HiBit
0x60ff, 0	4	DCT	MTR	Target velocity	V=TVelocity
0x6401, 1	2	MTR	NAC	Port A analog value	10bit Analog (shifted 5 bits) *z = (*z)<<5;
0x6401, 2	2	MTR	NAC	Port B analog value	10bit Analog (shifted 5 bits) *z = (*z)<<5;
0x6401, 3	2	MTR	NAC	Port C analog value	10bit Analog (shifted 5 bits) *z = (*z)<<5;
0x6401, 4	2	MTR	NAC	Port D analog value	10bit Analog (shifted 5 bits) *z = (*z)<<5;
0x6401, 5	2	MTR	NAC	Port E analog value	10bit Analog (shifted 5 bits) *z = (*z)<<5;
0x6401, 6	2	MTR	NAC	Port F analog value	10bit Analog (shifted 5 bits) *z = (*z)<<5;
0x6401, 7	2	MTR	NAC	Port G analog value	10bit Analog (shifted 5 bits) *z = (*z)<<5;
0x6510, 1	2	DCT	MTR	Rms time before shutdown	default=12000 (~3 seconds)
0x6510, 2	1	DCT	MTR	Internal temperature limit	TH=[value]

NAC=no access, DCT=dictionary, POL=poll, MTR=motor

0x6040 Controlword
Bit 0 = Switch on
Bit 1 = Enable voltage
Bit 2 = Quick stop
Bit 3 = Enable operation
Bit 4 = New setpoint
Bit 5 = Change set immediately
Bit 6 = absolute mode / relative mode
Bit 7 = Fault reset
Bit 8 = Halt
Bit 9, 10 = Reserved
Bit 11-15 = Manufacturer specific

0x6060 Modes of operation
0=reserved
1=Profile Position Mode
2=Velocity Mode
3=Profile Velocity Mode
4=Torque Profile Mode
5=reserved
6=Homing Mode
7=Interpolated Position Mode
8-127=reserved

0x6041 Statusword
Bit 0=Ready to switch on
Bit 1=Switched on
Bit 2=Operation enabled
Bit 3=Fault
Bit 4=Voltage enabled
Bit 5=Quick stop
Bit 6=Switch on disabled
Bit 7=Warning
Bit 8=Manufacturer specific
Bit 9=Remote
Bit 10=Target Reached
Bit 11=Internal limit active
Bit 12-13=Operation mode specific
Bit 14-15=Manufacturer specific

Object 2201h: User Variable
This User Variable object accesses the user assigned variables in the motor defined by the associated index.

long value	index	name
uv[k]	[0,25]	a - z
uv[k]	[26,51]	aa - zz, al[0] - al[25]
uv[k]	[52,76]	aaa - yyy al[26] - al[50]

The variable uv[k] is either read directly from the motor or when the index k is written, however, is only writtten directly.

Object 2304h: Motor Status
Motor status bit field from RW command:

0	Trajectory in progress
1	Historical right (+) limit
2	Historical left (-) limit
3	Index report available
4	Wraparound occurred
5	Excessive position error
6	Excessive temperature
7	Motor is off
8	Index input asserted
9	Right (+) limit asserted
10	Left (-) limit asserted
11	User math overflow
12	User array index error
13	Syntax error
14	Overcurrent occurred
15	Program checksum error

Object 2305h: Motor Control
Motor control bit field:

0	Software limit enable
1	Program control, 1=run, 0=end

Object 60FDh: Digital Inputs
This object reflects the current state of the digital input signals. The user may apply any signals to these inputs for special purposes like limit or reference switches.

bit	function
0	negative limit switch
1	positive limit switch
2	home switch
3	interlock
4-15	reserved
16	Port A
17	Port B
18	Port C
19	Port D
20	Port E
21	Port F
22	Port G

Object 2100h: Port Configuration
This Object defines the I/O

0-3	Ports A and B 0 = B input, A input 1 = B input, A output 2 = B output, A input 3 = B output, A output
4-5	Port C 0 = C input 1 = C output 2 = C positive limit 3 = reserved 6-7 Port D 0 = D input 1 = D output 2 = D negative limit 3 = reserved
8-10	Ports E and F 0 = F input, E input 1 = F input, E output 2 = F output, E input 3 = F output, E output 4 = i2c
11-12	Port G 0 = G input 1 = G output 2 = G go 3 = reserved
13-15	reserved