Mac drives and DeviceNet control

Content

Description and test: March 2005, Allan Rex Pedersen, Automatik Partner ApS

In this example Mac drive number 1, has node adr. #5 and Mac drive number 2, has node adr. #6
For IO is used a Phoenix in-line IL DN BK3 node adr. #2 (1 byte out / 1 byte in).

List of project files

Omron-JVL.dnt	Devicenet project for Omron network 1 motor.
Omron-JVL2.dnt	Devicenet project for Omron network 2 motors and 1 phoenix I/O
CJ1-JVL.rss	PLC program for using 1 motor at device net.

List of software tools for this note

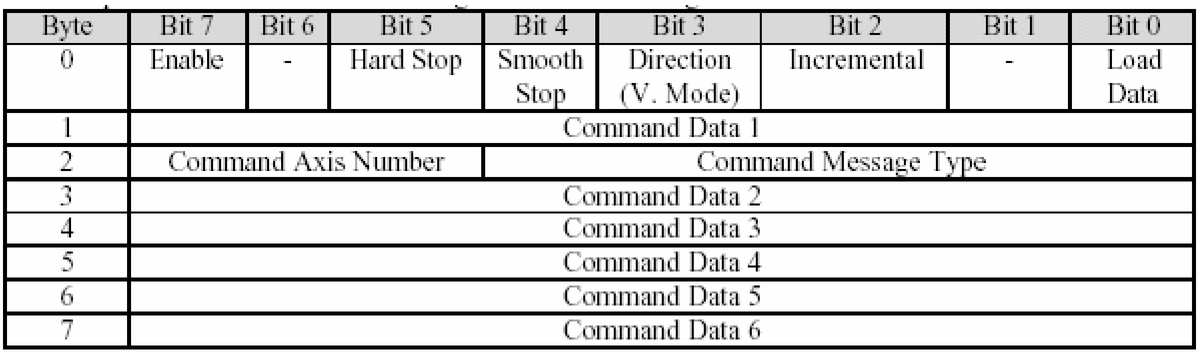
Controlled by: Omron CJ1 device net master module DRM-21
PLC program is made in CX-Programmer 3.2, which can be read by all newer version
Device net configurator software version 2.21, from Omron (for further info about devicenet configuration see omron manual W380).
MacTalk version 1.34 beta2, from JVL
MacRegIO version 1.14

List of equipment for making this note

Omron PLC: CPU, DeviceNet master master module DRM-21, digital input and digital output
Phoenix Contacts In-line block
2 pcs. Mac motor equipped with a device net interface: MAC140 + MAC00-FD4

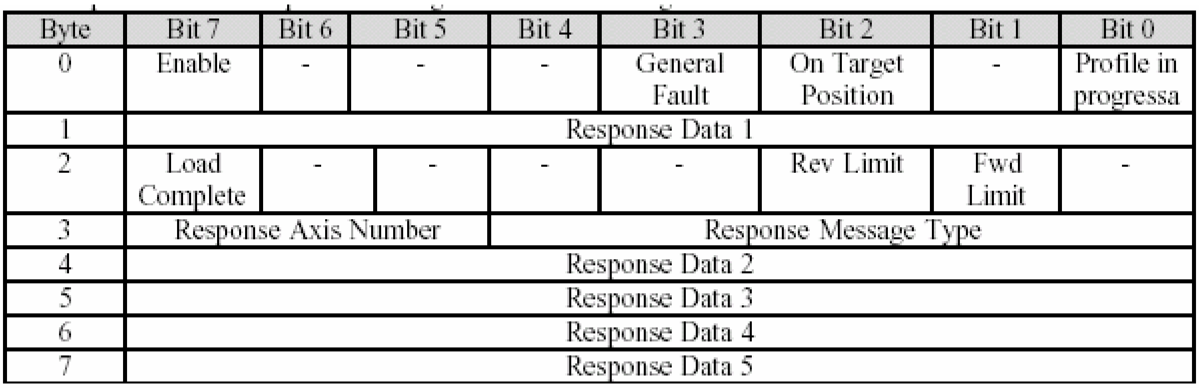
Short info about communication to mac motor

All communication is done via 8 bytes or 4 word, having a layout like a telegram. It is possible to set / get information to and from servo drive.
The communication is basically like setting specified values in the 4 word (see figure 1), and set the bit "LoadData", and wait for response at the input bit "LoadComplete" (see figure 2).
As the devicenet telegram layout is in byte and plc program in word, there will be a great job converting this information.



Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	Enable	-	Hard Stop	Smooth Stop	Direction (V. Mode)	Incremental	-	Load Data
1	Command Data 1							
2	Command Axis Number			Command Message Type				
3	Command Data 2							
4	Command Data 3							
5	Command Data 4							
6	Command Data 5							
7	Command Data 6							

Figure 1 Telegram layout for device net setting control bit and word. Startadr. 3201



Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	Enable	-	-	-	General Fault	On Target Position	-	Profile in progressa
1	Response Data 1							
2	Load Complete	-	-	-	-	Rev Limit	Fwd Limit	-
3	Response Axis Number			Response Message Type				
4	Response Data 2							
5	Response Data 3							
6	Response Data 4							
7	Response Data 5							

Figure 2 Telegram layout for device net getting status as bit and word. Startadr. 3301

How to use…

The idea about this note is to have an easy to use program, that may give You a possibility for fast setup and test of Your hardware and probably You may copy the subroutines to save time in Your programming.

Following steps is necessary:

1. set switches like described
2. Run configurator, create network and download to Your devicenet master
3. If needed change addresses, and correct parameter “Node addr.” In Subroutine call.
4. set enable flag for Your drive
5. set flag for MOVE, to see operation of Your drive positioning
6. Make modifications and finish Your job… Good luck

Good luck !

Configuration and setup Device net

Using the devicenet configurator software, You may put the wanted nodes in Your devicenet, and download to master module.
Check and correct switch setting on Your nodes. Node number shall match the configurator software.
Also the baudrate setting shall be the same for all units.
Resistors also called terminator should be either connected or activated by switch, in each physical end of the devicenet.

Configurator setup:

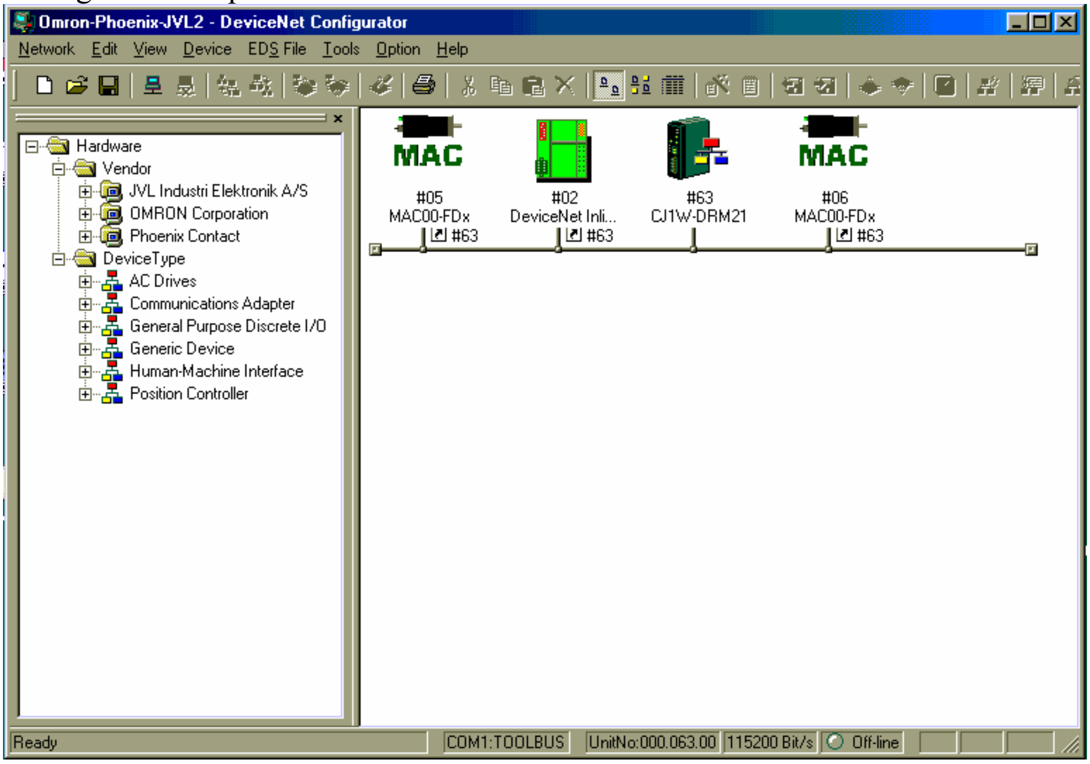


fig. 3 Device net configured with 2 drives. Filename: Omron-Phoenix-JVL2.npf

Omron PLC setup

CPU CJ1M-CPU13

Device net Master: DRM21
Nodenumber: 63
Unit number: 0
Baud rate: 125 kbit

PLC Status and control memory area:
CIO 1500 – 1524 (Unit: 0)
CIO 3200 – 3263 Reserved data area for IO communication of devicenet (control, write)
CIO 3300 - 3363 Reserved data area for IO communication of devicenet (status, read)
D 30000 – 30099

NB address area 3200-3363 is default, but can easily be changed in devicenet configurator. If You choose to change these addresses, You also need to change Base addresses in DIST and COLL instructions to make this program example work.

Device net I/O addresser (1.st device net):

Input adr.	Output adr
3200 dig. Input #2	3300 dig. Output #2
3201 Mac node #5 status	3301 Mac node #5 command
3202 Mac node #5	3302 Mac node #5

3203 Mac node #5	3303 Mac node #5
3204 Mac node #5	3304 Mac node #5
3201 Mac node #6 status	3301 Mac node #6 command
3202 Mac node #6	3302 Mac node #6
3203 Mac node #6	3303 Mac node #6
3204 Mac node #6	3304 Mac node #6

These CIO addresses will be valid for Omron series CJ1 and CS1.

Mac setup

Node number should be set at DIP switch on xx
Selection of termination for devicenet Switch 2 (if the drive is the last device physically in the devicenet wiring then ON, else OFF)
Selection of Baudrate for the devicenet

To set node: #5, and Baudrate: 125k

Switch no.	position
1	on
2	off
3	on
4	off
5	off
6	off
7	off
8	off
9	off
10	off

PLC program description: Easy-Mac

The PLC program is split in several sections to make it more simple to understand.
The program is constructed using subroutines for driver operation, The user should apply parameters and call the routines.

Section Main-test

Examples of how to call the different function to control Mac drives. The sections nessesarily can easily be copied into other User applications with the needed subroutines.
In this example the following control flags are used: enable, hardstop, incremental and load profile
Status flags: general fault, rev limit, fwd limit, in position, enabled, load completed.

Section UseMove

Move function is an “Easy to start” function for a simple positioning. This section shows how to set parameters. The section includes 2 call. One for position out at x100000, and one for position back to 0. The 2 call has different profiles. Out is going slow acceleration but high speed, and return is slow speed but fast acceleration and deceleration. See also Move below.

The only thing to do to position at 100.000 is to set flag (3080.14), the rest is turning out automatically. To return to 0, simply set (3080.13).

Section DeviceNet Common

Get status flags from drive and set command flags to drives, and check whether the call are.
Using Direct addressing, and shall be changed when devicenet addresses are changed.
Also containing status registers of Omron devicenet master, for simple trouble shooting.

Section Fault

Common section for indicating error and fault running devicenet, drives and communication functions.
The different timeout flags and other error indicators are collected into one flag (H30.00) which will indicate if an error should appear. Also an output will flash in the example.

Section DoCommand, and Section DoParameter

Basic commands (functions) are DoCommand and DoParameter doing transfers to and from drives.
The functions can handle one operation per call. Each call should be finished (correct) before another call can be done.
Each call will activate a small communication sequence for handshaking parameter to drives in a safe way.
The functions are relative addressing drives, which means that a drive is addressed using the first address in the devicenet, as the parameter NodeAddress (Interval 0..63).
DoParameter is also capable of handling DoPositionControllerAttribute, depending the value of parameter Messagetype (DoParameter messagetype hex 1f).

Section MoveSub

Move function is an “Easy to start”, kind of high level instruction. Using this function You only have to do one call in your part of the application to transfer a position profile to the drive and monitoring actual position until operation finished and motor is “In-position”.
This function is based upon several calls to basic command “DoCommand”.
See a call example in section UseMove. Set parameters and activate MOVE function

Section ZeroSearch

Sets the most important parameters for zerosearching.
Basically this function work as a 6 step sequence calling DoParameter 3 times one for each parameter using messagetype (hex 1b).

Section Initialize

Sets the most important parameters for drive control.
Basically this function work as a 6 step sequence calling DoParameter 3 times one for each parameter.

Description of Functions / Subroutines:

DoCommand

Ressources: word CIO 3091; Sub 1 DoCommand; call sub flag: 3091.15; Error 3070.00
Parameter layout

Address for parameter	**Calling parameter**	**Returning parameter**	**Comment**
D100	DINT		Parameter
D102	INT		Command Message type
D103	INT		Node address
D104	INT		Response message type
D110		DINT	Response value
D112		INT	Response msg. type

Example using DoCommand:
The function with example parameters will:
With node #5 (address 3201 to 3204), set target position= 10.000, Cmd. Messagetype=1 means “move to”, and Response message type = 1 means return actual position.

To activate example Set bit 3091.14
DoCommand(node: 1, Value: 10000, CmdMsg: 1, ResponsMsg: 1),
Return value = actual position and response msg. = 1.

Word scheme for PLC addresses writing DoCommand function at devicenet

3201	Block number	Control flags
3202	Response Axis no. and Message type	Control Axis no. and Message type
3203	Parameter value low order word	
3204	Parameter value high order word	

Word scheme for PLC addresses reading DoCommand respons at devicenet

3301	Block number	Status flags
3302	Response Axis no. and Message type	Status flags
3303	Parameter value low order word	
3304	Parameter value high order word	

Possible message types for command

Message number (hex)	**Description**	**Comments**
0	No operation	JVL standard
1	Target position	
2	Target velocity	
3	Acceleration	
5	Torque	
1B	PositionControllerAttribute	
1F	Parameter	

Possible message types for response

Message number	Description	Comments
0	No operation	
1	Actual position	
3	Actual velocity	
5	Torque	
14	Error code	N7:18 Lo order: general error code, and Hi order additional error code
1B	PositionControllerAttribute	
1F	Parameter	

NB some commands will only be valid when drive is enabled.

DoParameter
Ressources: word CIO 3092; Sub 2 DoParameter; call sub flag 3092.15; Error 3070.01
Messagetype: 1Fhex
Parameter layout

Address for parameter	Calling parameter	Returning parameter	Comment
D120	DINT		Parameter to set value
D122	INT		Messagetype: 1F
D123	INT		Node address
D124	INT		Attribute to set
D125	INT		Attribute to get
D130		DINT	Value of read parameter
D132		INT	Attribute of the read

NB: For normal position / velocity mode it's not necessary to use this function. Please notice that the units of the parameters is different from the devicenet units.

Example using DoParameter:
The function with example parameters will:
With node #5 (address 3201 to 3204), set value= 50, Messagetype=1f means parameter, Set attribute=6 (V_SOLL) and Get attribute=14 (GEARF1)

To activate example Set bit set 3092.14,
DoParameter(value: 50, Node: 1, Messagetype: #1F, Attribute set: #6, Attribute get: #14),
return value: 256, get attribute: 14 (means GEARF1 = 256)

Word scheme for PLC addresses writing DoParameter function at devicenet

3201	Parm Get Attribute	Control flags
3202	Parm Set Attribute	Axis no. and Message type
3203	Parameter value low order word	
3204	Parameter value high order word	

Word scheme for PLC addresses reading DoParameter function at devicenet

3301	Parameter read	Status flags
3302	Messagetype	Status flags
3303	Parameter value low order word	
3304	Parameter value high order word	

Some parameters are only possible to change when drive is enabled.

For further detail see “Technical Manual” for the motor available from JVL.

DoPositionControllerAttribute
Ressources: word CIO 3094; call DoParameter.; Error
Messagetype: 1Bhex
Parameter layout

Address for parameter	**Calling parameter**	**Returning parameter**	**Comment**
D120	DINT		Parameter to set value
D122	INT		Messagetype: 1F
D123	INT		Node address
D124	INT		Attribute to set
D125	INT		Attribute to get
D130		DINT	Value of read parameter
D132		INT	Attribute of the read

Example using DoPositionControllerAttribute:
The function with example parameters will:
With node #5 (address 3201 to 3204), set value= 50, Messagetype=#1B means messagetype, Set attribute=6 (homing torque) and Get attribute=102 (homing velocity).

Torque limit for zerosearch.
Notice: Torque zerosearch first activates at next rising enable. And is not visible.

To activate example Set bit set 3094.14,
DoPositionControllerAttr.(value: 50, Addressing Node: 1, Messagetype: #1B, Attribute set: 103, Attribute get: 102),
return value: -105, get attribute: 102 (value may be different)

LARS LISTE

ZeroSearch
Ressources: word CIO 3093; Sub 0; call DoParameter; call sub 3093.15; Error: 3070.
Messagetype: 1Bhex (basically DoPositionControllerAttribute)

D180	DINT	Type of zerosearch
D182	DINT	Zerosearch velocity
D184	DINT	Zerosearch Torque limit
D186	INT	Node address

This function set the "Enable flag" of the drive and function will bee active until the drive is "in-position".

Example using zerosearch: set torque limit.
NB: this example will turn motoraxis until torque limit is reached, then motor will change direction and turn on for 0,5 second.
3413 = 50 RPM, torque limit at the value 50 is stoppable using fingers

The function with example parameters will:
With node #5 (address 1.1 to 1.4), set Type of zerosearch=12, set Zerosearch velocity=3413, and set torque limit=50.

To activate example Set bit 3093.14,
Zerosearch (Type: 12, Velocity: 3413, Torque limit: 50)
No return value

Move
Ressources: word CIO 3080, 3088; Sub 3 DoCommand; call sub 309 . ; Error 3070.
Parameter layout

Address for parameter	**Calling parameter**	**Returning parameter**	**Comment**
D160	DINT		Target position
D162	DINT		Velocity
D164	DINT		Acceleration
D166	INT		Node address
D170		DINT	Response act. position

This function will only work if drive is not set into relative positioning.

Example using Use_Move:
The function with example parameters will:
To activate example Set 3080.13 and 3080.14,
Move1(Target pos.= 100000, Velocity = 50000 , Acc.= 10000)
Move2(Target pos.= 0, Velocity = 5000 , Acc.= 100000)
While the function is active it will continously return actual position of the drive.

Activation of MOVESUB function by setting 3080.15, the flag will be reset by the sub when drive becomes InPosition.

Parameter information:
Velocity: 5000 is slow, 100.000 is fast
Acceleration: 10.000 is slow, 100.000 is quicker (counts/second^2)

Name	Text	Devicenet	Skala	Value interval
Target velocity	Velocity during positioning	DoCommand Mess.type 2	(ingen)	0..273.000 (4000 RPM)
Acceleration	Acc. And Dec. during positioning	DoCommand Mess.type 3		1..27.306.666 counts / sek^2
Target position	Target position	DoCommand Mess.type 1	(ingen)	-67.000.000 .. +67.000.000

Figure 11. Table for values to apply

Initialize
Ressources: word CIO 3095; Sub 4 call DoCommand; call sub 3095.15; Error: 3070
Parameter layout

Address for parameter	**Calling parameter**	**Returning parameter**	**Comment**
D200	DINT		Maximum velocity
D202	DINT		Acceleration
D204	DINT		Torque limit
D206	INT		Node address

Sets the most important parameters for drive operation.
NB this function will only be valid when drive is enabled prior to call.

Example using Use_Move:
The function with example parameters will:
With node #5 (address 3201 to 3204),set maximum velocity = 135000, Acceleration = 270000, torque limit = 512,

To activate example Set 3094.14,

See table in figure 11, for further details.