

# Mac drives for Profibus control

## Content

List of software tools for this note .....	2
List of equipment for making this note .....	2
Short info about communication to mac motor .....	2
How to use... ..	3
PLC program description: Easy-Mac.....	3
Section Main-test, OB1 .....	3
Section Move, FB60 .....	4
Section Common Part of program, FB68.....	4
Section Fault, FB69.....	4
Description of Functions / Subroutines:.....	4
WriteParameter .....	4
ReadParameter .....	5
DoFastMacCommand .....	5
DoZeroSearch .....	6
Move .....	7
Configuration and setup of network using Step7.....	8
Siemens S7 PLC setup .....	9
Mac setup .....	10

In this example Mac drive number 1, has node adr. #5 and Mac drive number 2, has node adr. #6

NB Project is programmed using mnemonics in English.

An overview of registers and values are in the manual JVL lb0048-06GB. The manual is accessible at ([www.jvl.dk/downloads/JVL literature/User manuals](http://www.jvl.dk/downloads/JVL_literature/User_manuals))

For further detail about Registers, Parameters and dataformat, see lb048-06GB “JVL Technical Manual integrated motors MAC50...” for the motor available from JVL.

List of project files

- S7\_mac\_1. PLC program (.zip) for using 1 motor at Profibus.
- S7\_mac\_2. PLC program (.zip) for using 2 motors at Profibus.

### List of software tools for this note

Controlled by: S7-315-2DP incl. profibus interface  
PLC program is made in Step7 version 5.1  
MacTalk version 1.42, from JVL  
MacRegIO version 1.14, from JVL  
MAC motor using firmware FP1.7, Mac095 ver 7.5

### List of equipment for making this note

Siemens PLC: S7-315-2DP, digital input and digital output  
2 pcs. Of Mac motor equipped with each a Profibus interface: MAC140 + MAC00-FP4 and MAC 800-D1 + MAC00-FP4.

### Short info about communication to mac motor

All communication is done via 9 bytes output and 8 byte input, 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 bytes 0-3 (see figure 1), and toggle a command bit, and wait for response at the input command bit (see figure 2).

NB Resetting motor causes busfail at the profibus.

Address	Name	Description
0	Write data 3 (MSB)	Data to write to register
1	Write data 2	--- " ---
2	Write data 1	--- " ---
3	Write data 0 (LSB)	--- " ---
4	Write register selector	The register to write
5	Read register selector	The register to read
6	Direct register	Direct FlexMac command
7	Command	Bits for commanding reads/write
8	Input setup	Bits for input setup

Figure 1 Telegram layout for output data setting control bit and word. Startadr.: QB

Address	Name	Description
0	Read data 3 (MSB)	Data read from register register
1	Read data 2	--- " ---
2	Read data 1	--- " ---
3	Read data 0	--- " ---
4	Motor status	Status bits for the motor
5	Input status	Status of inputs
6	Last direct register	Last accepted direct FlexMac command
7	Command Status	Status bits for commands

Figure 2 Telegram layout for Input data getting status as bit and word. Startadr.: IB

Using profibus the units will be in counts. Positioning: counts, Acceleration: counts/sec.^2, velocity: counts/sec

## How to use...

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

Following steps are necessary:

1. set switches on MAC motor as described under configuration
2. If needed change addresses, and correct parameter "Node addr." In function calls.
3. Run DoZeroScr (M0.3) and try MOVE (M0.5 or M0.6), to see Your drive is positioning.
4. Make modifications and finish Your job...

Good luck !

## PLC program description: Easy-Mac

The PLC program is split in several sections to make it a more simple to understand.

The program is constructed using subroutines for driver operations, The user should apply new parameters and call the routines.

For a simple test of program, You may use the monitorlists "Variable table" in the project.

Common for the project is the word M99.1, collecting error flag from each part of the program.

### Section Main-test, OB1

To start a drive briefly, run the function DoZeroSrc, and thereafter positioning using the examples Move1 and Move2.

Examples of how to call the different function to control Mac drives. The sections necessarily can easily be copied into other User applications with the needed subroutines.

The section includes 2 calls for examples. One for positioning out at x100000, and one for positioning back to 0. The 2 calls has different positioning profiles. To position X using slow

acceleration but high speed, and return to position 0 using slow speed but fast acceleration and deceleration. See also Move below.

**Section Move, FB60**

Move function examples is an “Easy to start” function for a simple positioning. Drive should be enabled before call.

Using this function You only have to make 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”.

See call examples in section OB1. Set parameters and activate MOVE function

**Section Common Part of program, FB68**

Get status flags from drive and set command flags to drives, and check whether the calls are. Using Direct addressing, and shall be changed when net addresses are changed.

**Section Fault, FB69**

A common section for indicating of errors and faults using profibus, drives and communication functions.

The different timeout flags and other error indicators are collected into one flag (M99.1) which will indicate if an error should appear.

**Description of Functions / Subroutines:**

**WriteParameter**

Basic function WriteParameter makes transfers to drives using a relative addressing. The function can handle one operation per call.

The function are relatively addressing drives, which means that a drive is addressed using the first address for the drive in the Profibus, as the parameter NodeAddress (Interval 1..1021).

Ressources: FB62 and DB62; Sub WrParm; call function flag: M0.0;

Parameter layout

Address for parameter	Calling parameter	Returning Parm	Comment
DB62, 0.0	DINT		Parameter value
DB62, 0.4	INT		Motor register number
DB62, 0.5	BOOL		Set when 32 bit parameters
DB62, 0.6	INT		Node address, first number of motor adr.

Example using WriteParameter:

The function with example parameters will:

With node #5 (WrNodeAdr = 0, WrValue = 10.000, Wr32bitCmd = 0 , WrReg = 3 ).

Set acc. To 3000 counts / sec^2

To activate example Set bit M0.3  
WriteParameter(node: 1, Value: 10000,).

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

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

**ReadParameter**

Basic function ReadParameter makes transfers from drives using a relative addressing. The function can handle one operation per call.

The function are relatively addressing drives, which means that a drive is addressed using the first address for the drive in the Profibus, as the parameter NodeAddress (Interval 1..1021).

Resources: FB63 and DB63; Sub RdParm; call sub M0.1;  
Parameter layout

Address for parameter	Calling parameter	Returning parameter	Comment
DB63,		DINT	Parameter value
DB63,	INT		Motor parameter number
DB63,	BOOL		Set when 32 bit parameters
DB63,	INT		Node address, first number of motor adr.

**NB: For normal position / velocity mode it’s not necessary to use this function. Please notice that the units of the parameters are different from the other functions.**

Example using Read parameter:

The function with example parameters will:

With node #5, set value = 50, Set attribute = 6 (V\_SOLL) and Get attribute = 14 (GEARF1)

To activate example Set bit set M0.1,

ReadParameter( value: 50, Node: 1, Attribute set: #6, Attribute get: #14),  
return value: 256, get attribute: 14 (means GEARF1 = 256)

For further detail about Registers, Parameters and dataformat, see Ib048-06GB “JVL Technical Manual integrated motors MAC50...” for the motor available from JVL.

For example when problems in MAC, you may read ERR\_STAT register no. 35, dataformat is word.

**DoFastMacCommand**

In the Profibus interface there is added a facility, to send quick and more complicated commands, using only a single byte. The command is a value between 0..127.

See ....

Ressources: FB64 and DB64; call DoFastMac; call sub M0.2;  
Parameter layout

Address for parameter	Calling parameter	Returning parameter	Comment
DB64,	INT		Command
DB64,	INT		Node address, first number of motor adr.

A list of possible FastMac commands are available JVL User manual.....

Example using DoFastMacCommand:

The function with example parameters will:

With node #5, enable drive in position mode.

To activate example Set bit set M0.2,

DoFastMacCommand(command:64 nodeadr:0)

With node #5, disable drive.

DoFastMacCommand(command:0 nodeadr:0)

To repeat a command in FlexMac, You have to toggle bit 7 in the 6'th byte in command.

### DoZeroSearch

In the beginning the drive is disabled, then sends the parameters for zerosearching, and at last the drive is re enabled. The handling of Enable is caused in the fact that it is the positive transition of Enable which activates the transferred parameters.

The parameter "Zerosearch velocity", act now as a double word, using signed notation. Positive values start zerosearching CW, and negative values start zerosearching CCW.

Ressources: FB61 and DB61; Sub DoZeroSrc; call M0.3;

Parameter layout

DB61,	INT	Type of zerosearch
DB61,	DINT	PositionOffset
DB61,	DINT	Zerosearch Torque limit
DB61,	DINT	Zerosearch velocity
DB61,	BOOL	IsMAC800
DB61,	INT	Node address, first number of motor adr.

This function start up disabling the drive, and after parameter transfer, the drive is reenabled. The function call will be active while searching and until the drive is "in-position".

Example using zerosearch: set torque limit.

NB: this example will turn motor axis CW until torque limit is reached, then motor will change direction and turn on for 0,5 second. Finally an offset position will be done.

Speed 3413 = 50 RPM,

Torque limit at the value = 50 means that it is possible to stop motor using fingers on motor axis.

Startmode = 2, Register 37 => Position mode

Modereg. = 12..14, Register 2

The function with example parameters will:

With node #5, set Type of zereosrch = 12, set Zereosrch velocity = 3413, and set torque limit = 50.

To activate example Set bit M0.3

Zereosrch (Type: 12, Velocity: 75, Torque limit: 28 )

No return value

Torque limit for zereosrch.

Notice: Torque zereosrch first activates at the next rising enable. It is not possible to monitor this parameter in any way.

### Move

Move function is an “Easy to start” function for a simple positioning. This section shows how to set parameters.

Move function Calls WriteParameter 3 times for parameter transferring to driver sending first velocity, then acceleration and target position at last. Thereafter the actual position will be monitored, calling ReadParameter every 2. second requesting actual position until drive is “In-position” again.

Ressources: FB60 and DB60; Sub Move; call sub M0.4;

Parameter layout

Address for parameter	Calling parameter	Returning parameter	Comment
DB60,	DINT		Target position
DB60,	DINT		Velocity
DB60,	DINT		Acceleration
DB60,	BOOL		Set when 32 bit parameters
DB60,	INT		Node address, first number of motor adr.
DB60,		DINT	Response actual position, during run

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

Example using Move:

The function with example parameters will:

To activate example Set M0.5 and M0.6,

Move1(Target pos.= 100000, Velocity = 6291 , Acc.= 10)

Move2(Target pos.= 0, Velocity = 500 , Acc.= 500)

While the function is active it will continuously return actual position of the drive.

Activation of MOVE function by setting M0.4, the flag will be reset by the sub when drive becomes InPosition.

Parameter information:

Velocity: 50 is slow, 1.000 is fast

Acceleration: 10 is slow, 300 is quicker (counts/second<sup>2</sup>)

Name	Text	Skala	Value interval
Target velocity	Velocity during positioning	Counts / sec	0..8389 (4000 RPM)
Acceleration	Acc. And Dec. during positioning	Counts / sec <sup>2</sup>	1..399 counts / sek <sup>2</sup>
Target position	Target position	Counts	-67.000.000 .. +67.000.000

Figure 11. Table for values to apply

## Configuration and setup of network using Step7

Using the Step7 to configure your network. You may put the wanted nodes in Your Profibus, and download to network.

Check and correct switch setting on Your nodes. Nodenumbr shall match the configurator software.

Also the baudrate setting shall be the same for all units.

Resistors also called terminators should be either mounted or activated by switch, in each physical end of the Profibus.

To add a MAC00-xx drive to the profibus, view insert from hardware catalog, under “Additional Field devices, in the subpoint General.

Profibus configurator setup:



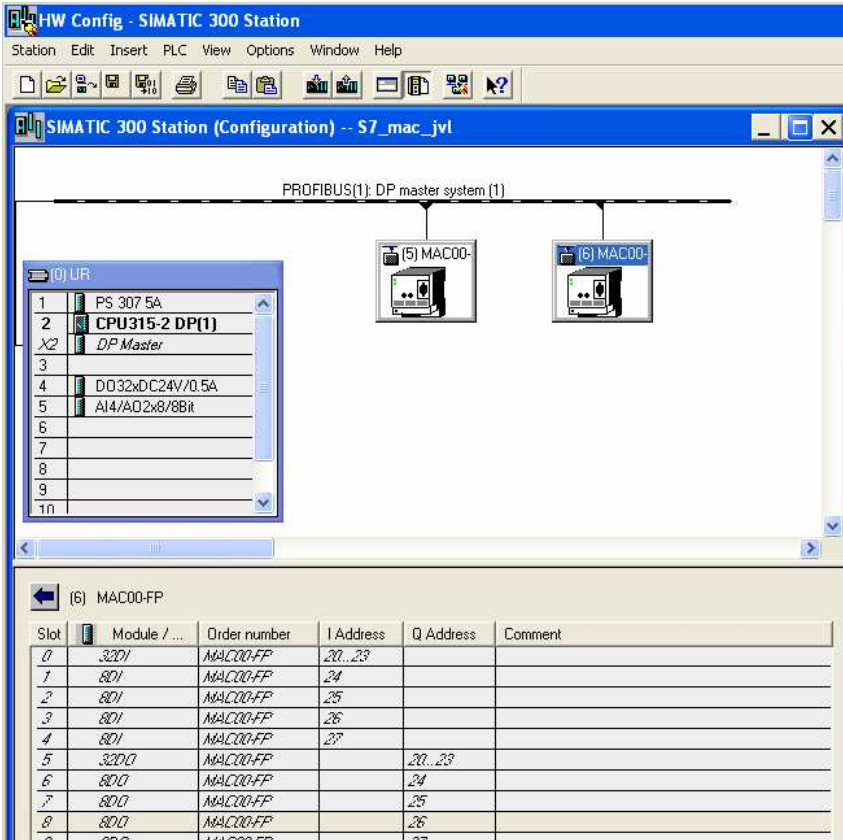


fig. 3 Profibus configured only with 2 drives and I/O, via hardware part of Step7 .

## Siemens S7 PLC setup

CPU S7-315-2DP

GSE/GSD file for setting up MAC to profibus in hardware section of project, is: JVLI06BC.GSD

PLC Status and control memory area:

QB Reserved data area for IO communication (control, write )

IB Reserved data area for IO communication (status, read)

**NB** address area depends on mounting position in rack. If You choose to change position You also have to choose the addresses in this example.

**NB** address of a servo motor must be continuous, for the indirect addressing to work, which means the ability to use these function described here. Also I and Q should start at the same number, for instance here both start at 0.

Addresses can be chosen manually using configurator in fig.3

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

Input adr.	Output adr
IB0..3 Mac node #5, Value from motor	QB0..3 Mac node #5, Value to motor

IB4 Mac node #5, motor status	QB4 Mac node #5, register to write
IB5 Mac node #5, Input status	QB5 Mac node #5, register to read
IB6 Mac node #5, Last direct register command	QB6 Mac node #5, Direct flex command
IB7 Mac node #5, Command status	QB7 Mac node #5, Command bits
	QB8 Mac node #5, Input setup bis
IB20 Mac node #6	QB20 Mac node #6
IB21 Mac node #6	QB21 Mac node #6

### Mac setup

Node number should be set at DIP switch SW1 at no. 1 to 7, addr. 0..127.

Selection of termination for profibus SW1 no. 9 and 10 (both ON = termination enabled, both OFF = termination disabled)

Baudrate: 1,5Mbit/s

Selection of Baudrate for the profibus

To set node: #5, and termination on (last bus module)

Switch no.    position

1	on
2	off
3	on
4	off
5	off
6	off
7	off
8	off
9	on
10	on

Normally Wiring the Profibus to 5 pole connector:

Green wire    to pole 2

Red wire      to pole 4

Hint, if motor not closed: when profibus is working well, use should be able to see 2 green LED on the print of the MAC-FP00