

**FB62 - <offline>**

"WrParm" Write prarmeter

Nom : wrparm

Famille : MAC

Auteur : arp

Version : 1.0

Version de bloc : 2

Horodatage Code : 16/06/2006 10:27:58

Interface : 06/06/2006 22:40:44

Longueur (bloc/code /données locales) : 00252 00134 00000

Nom	Type de données	Adresse	Valeur initiale	Commentaire
IN		0.0		
ParmValue	DInt	0.0	L#0	Value to be written to register
Is32Bit	Bool	4.0	FALSE	Value write as 16 or 32 bit (word or dword)
RegNumber	Int	6.0	0	Registernumber to write value
NodeAdr	Int	8.0	0	Start address of the servo node wanted
OUT		0.0		
IN_OUT		0.0		
STAT		0.0		
TEMP		0.0		

**Bloc : FB62 Writes parameter value to register number in servo**

Writes 1 parameter to a specific register in a MAC servo drive, per call.

Réseau : 1 relativ functionality

Create indirect address pointer

```

U      "WrOrderSent" //read wr order sent for handl  M99.5      -- WrParm
                        ing                          control flag
U      "WrOrderSent" //read wr order sent for handl  M99.5      -- WrParm
                        ing                          control flag
SPB   blok           //jump over order sending

L      #NodeAdr      //Get the nodeaddress number to write
LAR1   //load AR1 from accumulator

```

Réseau : 2 Send parameter

Depending on type of parameter, data is transferred to address Qd0 or QW2

```

L      #RegNumber
T      AB [AR1,P#4.0] //set registernumber to drive
U      #Is32Bit
U      #Is32Bit
SPB   ttb
L      #ParmValue    //This part is a 16 bit command
T      AW [AR1,P#2.0] //if parameter was an integer val
L      0
T      AW [AR1,P#0.0] //reset hi order part of DWord area

```

```

UN   #Is32Bit
R    A [AR1,P#7.5] //Reset bit for Write32bit
SPA  end

ttb: L   #ParmValue //This part is a 32 bit command
      T   AD [AR1,P#0.0] //if parameter was an double integer val

      U   #Is32Bit
      S   A [AR1,P#7.5] //Set bit for Write32bit

end:  U   "Dummy"           M1.0
      =   "Dummy"           M1.0

```

Réseau : 3	Toggle the WriteToggle flag in drive
------------	--------------------------------------

Toggle the write toggle flag in command byte
--

```

//toggle Q7.7 for servo starting at address 0
U    A [AR1,P#7.7] //read write toggle flag for level
U    A [AR1,P#7.7] //read write toggle flag for level
SPBN sete
R    A [AR1,P#7.7] //if off then set on
SPA  blok
sete: S   A [AR1,P#7.7] //if on then set off
blok: U   "WriteParmSub"           M0.0           -- Activa
                                   te sub
S     "WrOrderSent" //order has been sent M99.5           -- WrParm
                                   control flag

```

Réseau : 4	Reset calling bit at the end
------------	------------------------------

Finish function
-----------------

```

//afventer respons klar
U    A [AR1,P#7.7] //read read toggle flag for level
X    E [AR1,P#7.7] //read read status toggle flag for level
//compare
SPB  wait //if not ready jump over function end

U    "WriteParmSub" //final phase of WriteParamete M0.0           -- Activa
                                   r           te sub
R    "WrOrderSent" //reset sent info flag M99.5           -- WrParm
                                   control flag
R    "WriteParmSub" //reset call bit M0.0           -- Activa
                                   te sub
wait: U   "Dummy"           M1.0
      BE

```