FB60 - <offline>

"Move" Move command

Nom : move **Famille :** MAC
Auteur : arp **Version :** 1.0
Version de bloc : 2
Horodatage Code : 16/06/2006 11:18:55
Interface : 07/06/2006 15:15:37
Longueur (bloc/code /données locales) : 00384 00238 00002

Nom	Type de données	Adresse	Valeur initiale	Commentaire
IN		0.0		
TargetPos	DInt	0.0	L#0	Target position
Velocity	DInt	4.0	L#0	Velocity during positioning
Acceleration	DInt	8.0	L#0	Acc and Dec during positioning
IsMac800	Bool	12.0	FALSE	Type of motor, true = MAC800 motor
NodeAdr	Int	14.0	0	Start address of the node wanted
OUT		0.0		
ActualPos	DInt	16.0	L#0	Actuel position returning value
IN_OUT		0.0		
STAT		0.0		
TEMP		0.0		
inpos	Bool	0.0		

Bloc : FB60 Move function

Motor move, Including drive profile

Réseau : 1 Sequence last step

Step 4, reading actual position

```
      U     "WriteParmSub"  M0.0                -- Activate sub
      FN    "Flank3Move"    M95.2               -- Positive flank bit Move
      U     "Step3Move"     M97.2               -- Sequence for Move
      S     "Step4Move"     M97.3               -- Sequence for Move
      R     "Step3Move"     M97.2               -- Sequence for Move
```

Réseau : 2

Step 3, send target

```
      U     "WriteParmSub"  M0.0                -- Activate sub
      FN    "Flank2Move"    M95.1               -- Positive flank bit Move
      U     "Step2Move"     M97.1               -- Sequence for Move
      S     "Step3Move"     M97.2               -- Sequence for Move
      R     "Step2Move"     M97.1               -- Sequence for Move
```

Réseau : 3   Select register in servo for searchtype

Step 2, send velocity

```
      U     "WriteParmSub"  M0.0                -- Activate sub
      FN    "Flank1Move"    M95.0               -- Positive flank bit Move
      U     "Step1Move"     M97.0               -- Sequence for Move
      S     "Step2Move"     M97.1               -- Sequence for Move
      R     "Step1Move"     M97.0               -- Sequence for Move
```

Réseau : 4   Sequence first step, for stepping through parameter set

Step 1, send acceleration

```
      UN    "WriteParmSub"  //check not active            M0.0                -- Activa
                                                          te sub
      UN    "ReadParmSub"   //check not active            M0.1                -- Activa
                                                          te sub
      UN    "Moving"        //check not active, sequence l  M99.4             -- Mov fu
                            ock                           nction active
      S     "Step1Move"                                   M97.0               -- Sequen
                                                          ce for Move
      S     "Moving"                                      M99.4               -- Mov fu
                                                          nction active
```

Réseau : 5   Acc / Dec parm

First parameter to send

```
      U     "Step1Move"                                  M97.0               -- Sequen
                                                         ce for Move
      U     "Step1Move"                                  M97.0               -- Sequen
                                                         ce for Move
      SPBN  no1            //not in step1 => jump

      L     6
      T     "WrReg"        //Write to register no.6 Accel  MW104             -- Regist
                           eration                       er number to write
      L     #Acceleration
      T     "WrValue"                                    MD100               -- Value
                                                         to register write
      L     #NodeAdr
      T     "WrNodeAdr"                                  MW106               -- Start
                                                         address of the node at profi
                                                         bus

      U     "Step1Move"    //acceleration is only 16 bit  M97.0              -- Sequen
                           command                       ce for Move
      R     "Wr32bitCmd"   //reset cmd for 32 bit         M90.0              -- 32 bit
                                                          command handling
```

Réseau : 6   Velocity parm

Second parameter to send

```
no1:  U     "Step2Move"                                 M97.1               -- Sequenc
                                                        e for Move
      U     "Step2Move"                                 M97.1               -- Sequenc
                                                        e for Move
      SPBN  no2           //not in step2 => jump

      L     5
      T     "WrReg"       //Write to  register no.5 veloc  MW104             -- Registe
                          ity                           r number to write
      L     #Velocity
      T     "WrValue"                                         MD100              -- Value t
                                                              o register write
      L     #NodeAdr
      T     "WrNodeAdr"                                       MW106              -- Start a
                                                              ddress of the node at profibu
                                                              s

      U     "Step2Move"   //velocity is only 16 bit comma     M97.1              -- Sequenc
                          nd                                  e for Move
      R     "Wr32bitCmd"  //reset cmd for 32 bit              M90.0              -- 32 bit
                                                              command handling
```

Réseau : 7   Target position parm
Third parameter to send

```
no2:  U     "Step3Move"                                       M97.2              -- Sequenc
                                                              e for Move
      U     "Step3Move"                                       M97.2              -- Sequenc
                                                              e for Move
      SPBN  no3           //not in step3 => jump

      L     3
      T     "WrReg"       //Write to register no.3 P_SOLL     MW104              -- Registe
                                                              r number to write
      L     #TargetPos
      T     "WrValue"                                         MD100              -- Value t
                                                              o register write
      L     #NodeAdr
      T     "WrNodeAdr"                                       MW106              -- Start a
                                                              ddress of the node at profibu
                                                              s

      U     "Step3Move"   //position is a 32 bit command      M97.2              -- Sequenc
                                                              e for Move
      S     "Wr32bitCmd"  //set cmd for 32 bit                M90.0              -- 32 bit
                                                              command handling
```

Réseau : 8   First flank activate
Step 1 set "WriteParmSub"

```
no3:  U     "Step1Move"     M97.0              -- Sequence for Move
      S     "WriteParmSub"  M0.0               -- Activate sub
```

Réseau : 9   Second flank activate
Step 2 set "WriteParmSub"

```
      U     "Step2Move"     M97.1              -- Sequence for Move
      S     "WriteParmSub"  M0.0               -- Activate sub
```

Réseau : 10   Third flank activate
Step 3 set "WriteParmSub"

```
      U     "Step3Move"     M97.2              -- Sequence for Move
      S     "WriteParmSub"  M0.0               -- Activate sub
```

```
Réseau : 11     Read actual position from servo

Request drive actual position command via ReadParameter function call

      U     "Step4Move"                                     M97.3             -- Sequen
                                                            ce for Move
      U     "Step4Move"                                     M97.3             -- Sequen
                                                            ce for Move
      SPBN  pos1            //not in step4 => jump

      L     10              //register 10 = actual position
      T     "RdReg"                                         MW114             -- Regist
                                                            er number to read
      U     "Step4Move"                                     M97.3             -- Sequen
                                                            ce for Move
      S     "Rd32bitCmd"    //as 32 bit data, LongInt       M90.1             -- 32 bit
                                                             command handling
      UN    "ReadParmSub"   //when not requesting,          M0.1              -- Activa
                                                            te sub
      UN    "ReadParmSub"   //when not requesting, read     M0.1              -- Activa
                                                            te sub
      SPB   les
      L     ED [AR1,P#0.0]  //read actual pos
      T     #ActualPos      //write actual position to response parm

les:  UN    "ReadParmSub"   //when not requesting, active   M0.1              -- Activa
                            new request                     te sub
      S     "ReadParmSub"   //call read actual value        M0.1              -- Activa
                                                            te sub


Réseau : 12     make the in position signal

InPosition signal is created by sequence active step and the status flag of the
motor.

pos1: U     E [AR1,P#4.4]  //read In position flag for level
      U     "Step4Move"    //parameters are sent           M97.3             -- Sequen
                                                           ce for Move
      =     #inpos


Réseau : 13     Time delay before this function stops

Monitors the in position flag, and when it has been active for a while.

      U     #inpos
      L     S5T#2S
      SE    "DelayBeforeEnd"  T20               -- Delay for updating actual position b
                              efore end


Réseau : 14     Stop this function by reset Moving flag


      U     "DelayBeforeEnd"  T20               -- Delay for updating actual position b
                              efore end
      R     "Moving"          M99.4             -- Mov function active
```

Réseau : 15   Wait for Moving off and reset calling bit plus sequence

Respond from servo, comes after a while when read toggle in command status is
equal to read toggle in command.

```
      UN    "Moving"                                          M99.4              -- Mov func
                                                              tion active
      UN    "Moving"                                          M99.4              -- Mov func
                                                              tion active
      SPBN  wait       //if not ready jump over function end

//Function end after time delay
      U     "MoveSub"                                         M0.4               -- Activate
                                                               sub
      R     "MoveSub"  //reset call bit                       M0.4               -- Activate
                                                               sub
      R     "Moving"   //reset order sent work bit            M99.4              -- Mov func
                                                              tion active
      L     0          //reset sequence for transferring
      T     "MoveSeq"                                         MB97               -- Sequence
                                                               bits

wait: U     "Dummy"    //jump to here when not ending f       M1.0
                       unction
      =     "Dummy"                                           M1.0
      BE
```