FB64 - <offline>

"DoFastMAc"   Do fast commands to MAC
Nom : FastMac   **Famille :** MAC
Auteur : arp   **Version :** 1.0
                           Version de bloc : 2
Horodatage Code :   17/07/2006 01:11:05
          Interface : 07/06/2006 13:52:56
Longueur (bloc/code /données locales) : 00230   00112   00002

Nom	Type de données	Adresse	Valeur initiale	Commentaire
IN		0.0		
FastCmd	Int	0.0	0	Command number for DoFastCommand of a MAC motor
NodeAdr	Int	2.0	0	Start address of the servo node wanted
OUT		0.0		
IN_OUT		0.0		
STAT		0.0		
TEMP		0.0		
LastCmd	Int	0.0		Temporary for last command

Bloc : FB64   DoFastMac command

Réseau : 1	relativ functionality

Create indirect address pointer

```
      U     "FastMacSent"                          M99.2          -- FlexMa
                                                 c command is sent
      U     "FastMacSent"                          M99.2          -- FlexMa
                                                 c command is sent
      SPB   slu

      L     #NodeAdr
      LAR1                    //load AR1 from accumulator
```

Réseau : 2	Test for repeating command

If repeating command, it should be informed the MAC motor in a different way

```
      L     AB [AR1,P#6.0]  //read last sent command
      L     B#16#7F         //remove bit 7 because it is a flag
      UW
      T     #LastCmd        //save last sent command to flexmac

      L     #LastCmd        //Compare LastCommand and new FlexMac
      L     #FastCmd
      <>I
      SPB   opr             //if different go to simple cmd sending
```

```
┌────────────────────────────────────────────────────────────────────────────────────┐
│ Réseau : 3        Repeating command therefore Toggle FlexMac bit 7                   │
├────────────────────────────────────────────────────────────────────────────────────┤
│ This network is only for repeating commands                                          │
└────────────────────────────────────────────────────────────────────────────────────┘

        U     A [AR1,P#6.7]  //statusread of the toggle flag
        U     A [AR1,P#6.7]  //statusread of the toggle flag
        SPB   off            //Flag is on jump to reset part

        UN    "AllwaysOff"   //read statusread of the toggl    M1.7              -- Allway
                             e flag                            s off bit
        S     A [AR1,P#6.7]  //set the toggle flag
        SPA   slu

off:    UN    "AllwaysOff"   //read statusread of the toggl    M1.7              -- Allway
                             e flag                            s off bit
        R     A [AR1,P#6.7]  //reset the toggle flag
        SPA   slu


┌────────────────────────────────────────────────────────────────────────────────────┐
│ Réseau : 4        Sending new command                                                │
├────────────────────────────────────────────────────────────────────────────────────┤
│ Depending on type of parameter, data is transferred to address                       │
└────────────────────────────────────────────────────────────────────────────────────┘

//Simple version of command sending starts here
opr:    L     #FastCmd
        T     AB [AR1,P#6.0]  //Load Direct register with command


┌────────────────────────────────────────────────────────────────────────────────────┐
│ Réseau : 5        Transfer result and reset calling bit                              │
├────────────────────────────────────────────────────────────────────────────────────┤
│ Respond from servo, comes after a while when read toggle in command status is        │
│ equal to read toggle in command.                                                     │
└────────────────────────────────────────────────────────────────────────────────────┘

slu:    UN    "AllwaysOff"                                     M1.7              -- Allway
                                                               s off bit
        S     "FastMacSent"                                    M99.2             -- FlexMa
                                                               c command is sent

//Testing for accepted command
        L     AB [AR1,P#6.0]
        L     EB [AR1,P#6.0]
        <>I
        SPB   wait

//function end
        U     "DoFastMacSub"                                   M0.2              -- Activa
                                                               te sub
        R     "DoFastMacSub"  //reset call bit                 M0.2              -- Activa
                                                               te sub
        R     "FastMacSent"   //reset routine active bit       M99.2             -- FlexMa
                                                               c command is sent

wait:   U     "Dummy"                                          M1.0
        =     "Dummy"                                          M1.0
        BE
```