

Listing: DDPTP.PAS

```
{*****}
{*                D D P T P . P A S                *}
{*-----*}
{* Task          : Enables selective changing of individual *}
{*                values in the disk drive parameter table. *}
{*-----*}
{* Author         : Michael Tischer                        *}
{* Developed on   : 08/22/91                                *}
{* Last update    : 04/07/95                                *}
{*****}
```

program DDPTP;

Uses Crt, Dos; { Add Crt and Dos units }

type DDPT\_T = array[ 0..10 ] of byte; { Structure for the DDPT }  
DDPT\_PTR = ^DDPT\_T; { Pointer to the DDPT }

var DDPT : DDPT\_PTR; { Pointer to the DDPT }

```
{*****}
{* byte_hex : Changes a byte to a HEX number.                *}
{* Input    : Number to be changed                            *}
{* Output    : Number as a hex string                          *}
{*****}
```

function byte\_hex( rnum : byte ) : string;

{-- Change a numeral from 0 - 15 to 0H - FH -----}

function h\_numeral( numeral : byte ) : char;

```

begin
  if ( numeral >= 10 ) then          { Numeral >= 10 then A - F }
    h_numeral := chr( 55 + numeral )
  else
    { No, decimal numeral }
    h_numeral := chr( 48 + numeral );
end;

begin
  byte_hex := '$' + h_numeral( rnum div 16 ) + h_numeral( rnum mod 16 );
end;

{*****}
{ * hex_byte : Changes a hex string to a byte. * }
{ * Input    : Hex string to be changed      * }
{ * Output   : Number                        * }
{*****}

function hex_byte( hex : string ) : byte;

{-- Change hex numeral 0H - FH to 0 - 15 -----}

function d_numeral( numeral : char ) : byte;

begin
  if ( numeral >= 'A' ) and ( numeral <= 'F' ) then
    d_numeral := ord( numeral ) - 55
  else
    { No, decimal number }
    d_numeral := ord( numeral ) - 48;
end;

begin

```

```

if ( hex[ 1 ] = '$' ) then
  delete( hex, 1, 1 );
if length( hex ) = 1 then
  hex := '0' + hex;
hex_byte := d_numeral( hex[ 1 ] ) * 16 + d_numeral( hex[ 2 ] );
end;

```

```

{*****}
{ * RAM_DDPT : Test whether DDPT is in RAM or in ROM. * }
{ * Input      : None * }
{ * Output     : TRUE if DDPT is in RAM * }
{ * Info       : The function writes a value to the DDPT, reads it * }
{ *              out again, in this way determining whether the value * }
{ *              could be changed, since the DDPT is in RAM. * }
{*****}

```

```

function RAM_DDPT : boolean;

```

```

var buffer : byte;           { Memory for current value of the DDPT }

```

```

begin

```

```

  buffer := DDPT^[ 0 ];           { Save value of the DDPT }
  DDPT^[ 0 ] := not buffer;       { Invert value }
  RAM_DDPT := ( DDPT^[ 0 ] = not buffer ); { Evaluate write test }
  DDPT^[ 0 ] := buffer           { Restore old value }

```

```

end;

```

```

{*****}
{ * DisplayValues: Displays values of the DDPT. * }
{ * Input        : None * }
{ * Output       : None * }
{*****}

```

```
procedure DisplayValues;
```

```
begin
```

```
  writeln( 'Step rate           (SR): ',  
           byte_hex( DDPT^[ 0 ] shr 4 ) );  
  writeln( #13#10'Head unload time (HU): ',  
           byte_hex( DDPT^[ 0 ] and $F ) );  
  writeln( 'Head load time      (HL): ',  
           byte_hex( DDPT^[ 1 ] shr 1 ) );  
  writeln( 'Head settle time    (HS): ',  
           byte_hex( DDPT^[ 9 ] ) );  
  writeln( #13#10'Motor postrun time (MP): ',  
           byte_hex( DDPT^[ 2 ] ) );  
  writeln( 'Motor startup time (MS): ',  
           byte_hex( DDPT^[ 10 ] ) );
```

```
end;
```

```
{ *****  
* NewValues: Sets new values of the DDPT.                                     *  
* Input      : None                                                         *  
* Output     : None                                                         *  
* *****
```

```
procedure NewValues;
```

```
var i,j      : byte;                                { Loop counter }  
    PCh      : string[ 2 ];                        { Parameter to be changed }  
    NewV     : byte;                                { New value to be set }  
    AuxiValue : byte;                              { Auxiliary value to be saved }  
    CmdPar    : string[ 6 ];                       { Command line parameter }
```

```

begin
  {-- Loop      : Execute all parameters -----}

  for i := 1 to Paramcount do
    begin
      CmdPar := paramstr( i );                                { Get parameter }
      for j := 1 to length( CmdPar ) do                      { Command in upper-case }
        CmdPar[ j ] := upcase( CmdPar[ j ] );
      PCh := copy( CmdPar, 1, 2 );                            { Value to be changed }
      delete( CmdPar, 1, 3 );                                  { Determine new value }
      NewV := hex_byte( CmdPar );
      if ( PCh = 'SR' ) then                                    { Step rate? }
        begin
          NewV := NewV shl 4;                                  { Value in upper nibble }
          AuxiValue := DDPT^[ 0 ] and $0F;                    { Lower nibble }
          DDPT^[ 0 ] := NewV or AuxiValue;                     { Save value }
        end
      else if ( PCh = 'HU' ) then                                { Head unload time? }
        begin
          NewV := NewV and $0F;                                { Only value in lower nibble }
          AuxiValue := DDPT^[ 0 ] and $F0;                    { Upper nibble }
          DDPT^[ 0 ] := NewV or AuxiValue;                     { Save value }
        end
      else if ( PCh = 'HL' ) then                                { Head load time? }
        DDPT^[ 1 ] := NewV shl 1                               { Save balue in bit 1 - 7 }
      else if ( PCh = 'HS' ) then                                { Head settle time? }
        DDPT^[ 9 ] := NewV                                     { Save value }
      else if ( PCh = 'MP' ) then                                { Motor post run time? }
        DDPT^[ 2 ] := NewV                                     { Save value }
      else if ( PCh = 'MS' ) then                                { Motor starting time? }
        DDPT^[ 10 ] := NewV;                                   { Save value }
      end;
    end;
  end;

```

end;

```
{ ***** }
{ *                MAIN PROGRAM                * }
{ ***** }
```

begin

ClrScr; { Clear screen }

writeln( 'DDPTP - (c) 1992 by Michael Tischer' );

writeln( 'Allows user defined changes to current DDPT' );

GetIntVec( \$1E, pointer( DDPT ) ); { Get pointer to DDPT }

if ( RAM\_DDPT ) then { DDPT in RAM, can be changed? }

begin

if ( Paramcount > 0 ) then { Set values? }

begin

NewValues; { Yes, set new values }

writeln( #13#10#10'New DDPT contents:' );

DisplayValues; { Display new values of DDPT }

exit;

end;

end

else

writeln( 'Disk drive parameter table in ROM - cannot be changed' );

writeln( #13#10'DDPT contents:' );

DisplayValues; { Display old values of DDPT }

end.