

```

*****
*                                     T Y P M P                                     *
*****
* Task           : Sets the typematic rate on the MF II keyboard *
*                 according to user preferences.                  *
*****
* Author          : Michael Tischer                                *
* Developed on    : 08/27/88                                       *
* Last update    : 01/22/92                                       *
*****
program TYPMP;

*****
* SetTypm: Sends the key repeat rate to the keyboard controller. *
* Input  : RATE : the repeat rate to be set                      *
* Output : TRUE if the value was set                               *
*         FALSE if an error occurred                              *
* Info   : This function can be added from a UNIT or OBJ file.    *
*****
{$F+}                                { This function uses the FAR call model }

function SetTypm( Rate : byte ) : boolean;

begin
  inline(
    $32/$D2/$B4/$F3/$FA/$E8/$13/$00/$75/$0A/$8A/$66/$06/$E8/
    $0B/$00/$75/$02/$FE/$C2/$FB/$88/$56/$FF/$EB/$27/$90/$51/
    $53/$B3/$03/$33/$C9/$E4/$64/$A8/$02/$E0/$FA/$8A/$C4/$E6/
    $60/$E4/$64/$A8/$01/$E1/$FA/$E4/$60/$3C/$FA/$74/$07/$FE/
    $CB/$75/$E6/$80/$CB/$01/$5B/$59/$C3
  );
end;

{$F-}

*****
**                               MAIN PROGRAM                               **
*****

var Delay,                               { Stores the delay }
    Speed,                               { Stores the key repeat rate }
    Fpos1,
    FPos2 : integer;                    { Error position in string conversion }
    ParErr : boolean;                  { Error in parameter passing }

begin
  writeln(#13#10,'TYPMP - (c) 1988, 92 by MICHAEL TISCHER');
  ParErr := true;                      { Assume error in parameters }
  if ParamCount = 2 then                { Were 2 parameters passed? }
    begin                               { Yes }
      val(ParamStr(1), Delay, FPos1);    { First parameter to integer }
      val(ParamStr(2), Speed, FPos2);    { Second parameter to integer }
      if ((FPos1=0) and (FPos2=0)) then   { Error in conversion? }
        if ((Delay < 4) and (Speed < 32)) then { No --> Value O.K.? }
          ParErr := false;               { Yes --> Parameters are O.K. }
      end;
    if ( ParErr ) then                  { Are parameters O.K.? }
      begin                             { No }
        writeln('Syntax: TYPMP          delay      key_repeat_rate');
        writeln('          ',#30,'          ',#30);
        writeln('          3          3');
        writeln('          0 : 1/4 second 3 0 : 30.0 reps/sec 3');
        writeln('          1 : 1/2 second 3 1 : 26.7 reps/sec 3');
        writeln('          2 : 3/4 second 3 2 : 24.0 reps/sec 3');
        writeln('          3 : 1 second   3 3 : 21.8 reps/sec 3');
        writeln('          AAAAAAAAAAAAAAAAAA 3');
        writeln('          3 all values +-20% 3');
        writeln('          AAAAAAAAAAAAAAAAAA 3');
        writeln('          3 28 : 2.5 reps/sec 3');
        writeln('          3 29 : 2.3 reps/sec 3');
        writeln('          3 30 : 2.1 reps/sec 3');
        writeln('          3 31 : 2.0 reps/sec 3');
        writeln('          AAAAAAAAAAAAAAAAAA 3');
      end
    else                                { The parameters are O.K. }
      begin
        if (SetTypm( (Delay shl 5) + Speed )) then { Set key repeat rate }
          writeln('The keyboard repeat rate was set.')
        else
          writeln('ERROR accessing the keyboard controller.');
```

