

njen z 255 na 0, pomakne PEEK 23674 za 1 navzgor. To nam pojasnjuje zakaj zgornji izraz deluje.

Sedaj pazljivo pogledajte naslednje: predstavljajmo si, da so naša tri števila 0 za PEEK 23674, 255 za PEEK 23673 ter 255 za PEEK 23672. To pomeni, da bo 21 minut po vključitvi naš izraz dal rezultat:

$$(65536 \times 0 + 256 \times 255 + 255) / 50 = 1310.7$$

Vendar obstoja skrita nevarnost. Ko se naslednjič prešteje 1/50 sekunde, se bodo tri števila spremenila v 1, 0 in 0. To se bo dogajalo, ko boste na polovici izraza; računalnik bo ocenil PEEK 23674 kot 0 ter nato spremenil ostala dva tako, da jih bo postavil na 0 tik, preden bi jih pregledal. V tem primeru bi bil odgovor sledeč:

$$(65536 \times 0 + 256 \times 0 + 0) / 50 = 0$$

kar je popolnoma napačno. Enostavno pravilo, s katerim boste rešili ta problem je, da ocenite izraz dvakrat zaporedoma ter vzamete večji odgovor. Če natančneje pregledate zgornji program, boste videli, da on to počne sam od sebe.

Oglejmo si "trik" za uporabo pravila. Definirajte funkcije:

10 DEF FN m(x, y) = (x+y+ABS(x-y))/2: REM daje večjega izmed x in y

20 DEF FN u()=(65536*PEEK 23674+256*PEEK 23673+PEEK 23672)/50: REM čas, lahko je napačen

30 DEF FN t()=M(FN u(), FN u()): REM čas, točno

Spremenite lahko tri števila, ki merijo število preteklih petdeset^eing sekunde tako, da bodo merila pravi čas in ne tistega, ki nam pove koliko časa je računalnik vključen. Na primer, da bi nastavili čas na 10.00 dopoldne, morate izračunati koliko petdeset^eing sekunde je to (10x60x60x50 = 1800000 petdeset^eing) ter da je:

$$1800000 = 65536 * 27 + 256 * 119 + 64$$

Da bi nastavili tri števila na 27, 119 in 64, vtipkajte:

POKE 23674,27: POKE 23673,119: POKE 23672,64

V državah, kjer se uporabljajo frekvence iz 60 Hz, je v teh programih potrebno '50' zamenjati s '60', kjer je to potrebno.