

OKRESNÝ PODNIK VÝROBY A SLUŽIEB
DOLNÝ KUBÍN

ROM MODUL

pre personálny počítač SMEP
PP 01



Príručka používateľa

ROMCA

SPC pre nezávislé hodiny na PP01.

Tento SPC zaberá 1 kb pamäti EPROM a môže byť v ROM module zasunutý v ľubovoľnej pozícii. SPC, po spustení riadiacou časťou, prekopíruje svoj obsah z pamäte EPROM (zasunutej v ROM module) do pamäte RWM (prítomnej v PP01) a ROM modul je programovo odpojený t.j. môžeme prípadne ROM modul z konektora vytiahnuť.

Po prekopírovaní obsahu SPC je riadenie odovzdané do SPC. SPC spojí tabuľku príkazov a funkcií GBASICu s tabuľkou príkazov a funkcií SPC t.j. rozšíri programovací jazyk GBASICu o nové príkazy a funkcie. Pripojenie novej tabuľky príkazov a funkcií sa môže vykonať iba raz. T.j. je zakázané pripojiť dva a viackrát tú istú tabuľku príkazov a funkcií. Ak tento zákaz porušíme, potom dôjde k zacykleniu GBASICu a nepomôže nič inšie iba tvrdý reset (t.j. stlačenie kláves ekvivalentných vypnutiu a zapnutiu PP01).

SPC pre nezávislé hodiny (v ďalšom texte sa bude používať výraz "hodiny") je vytvorený v 2 verziách, pričom jediným rozdielom medzi nimi je, že sa obsadzujú inú oblasť RAM pamäti.

Jednotlivé SPC sa po vykonávaní príslušného príkazu ROM ohlásia výpisom :

HODINY NA PP01 /V1.x/

Ľáto verzia obsadí pamäť od adresy 8D00H po adresu 8FFFH.

HODINY NA PP01 /V2.x/

Ľáto verzia obsadí pamäť od adresy 8200H po adresu 84FFH.

SPC pre nezávislé hodiny na PP01 rozšíri programovací jazyk GBASIC o príkazy popísané v ďalšom.

Všetky nasledujúce príkazy môžu byť vykonávané v priamom i v príkazovom režime.

Pre vytváranie reálneho času v PP01 sa používa prerušovací systém PP01, ktorý je tvorený obvodom 8214. GBASIC nepoužíva prerušovací systém, a okrem toho SPC ho nepoužíva ani žiadny iný SPC.

Pre vlastné generovanie časových intervalov sa používajú čítač 0 a čítač 1 obvodu 8253, pričom výstup čítača 0 je zapojený do hodín čítača 1 a výstup čítača 1 je zapojený na prvú prerušovaciu úroveň.

Po spustení tohto SPC (príkazom ROM) sa nastaví oba obvody t.j. 8253 a 8214 a povolí sa prerušenie.

Užívateľský program nesmie zakazovať a povoľovať prerušenia!

Hodiny začnú bežať od hodnoty 00:00:00 a časový krok je nastavený na 5 sekúnd. Teda každých 5 sekúnd obvod 8253 vygeneruje žiadosť o prerušenie a táto žiadosť o prerušenie je prostredníctvom obvodu 8214 a mikroprocesora ošetrená ako prerušenie na prvej prerušovacej úrovni. To znamená, že je prerušená práca programovacieho jazyka GBASIC a vykoná sa obslužný podprogram pre prvú prerušovaciu úroveň.

Obslužný podprogram pre túto úroveň započíta ku časovému údaju časový krok a ak je povolené zobrazovanie času, potom aj zobrazí nový časový údaj. Ďalej sa kontroluje, či bol nastavený alarm a ak bol, potom sa testuje, či aktuálny čas neprekročil čas nastavený ako alarm. Ak áno potom sa spustí program od nastaveného riadku. V každom prípade, po skončení práce obslužného podprogramu pokračuje činnosť GBASICu.

(Ako nastaviť alarm, časový krok, povoliť zobrazenie - toto všetko bude popísané v ďalšom texte.)

Po stlačení kláviess RESET je časovač zastavený a spustí sa až po stlačení klávesy CR.

TIMESET - počiatočné nastavenie času.

syntax :

TIMESET výr1,výr2,výr3

kde :

výr1 po orezaní desatinnej časti musí byť z intervalu

0 až 23

výr2 po orezaní desatinnej časti musí byť z intervalu

0 až 59

výr3 po orezaní desatinnej časti musí byť z intervalu

0 až 59

Hodnota výrazu výr1 určuje hodiny, hodnota výrazu výr2 určuje minúty, hodnota výrazu výr3 určuje sekundy.

Po zavedení tohto SPC príkazom ROM sa ihneď spustia hodiny od hodnoty 0,0,0. Príkazom TIMESET môžeme nastaviť čas, od ktorého sa má bežať aktuálny čas. Po vykonaní príkazu TIMESET je znovu nastavené obvody 8214 a 8253 a do pracovných buniek sa uloží zadaný údaj pre hodiny, minúty a sekundy. Obvod 8253 je nastavený na aktuálny časový krok (po zavedení tohto SPC je časový krok 5 sekúnd).

TIMEEN - povolenie zobrazovania času.

syntax :

TIMEEN výr1, výr2

kde:

výr1, výr2 musia byť po orezaní desatinných častí z intervalu 0 až 31.

Výr1 určuje riadok a výr2 určuje stĺpec na zobrazovacej jednotke. Od takto zadanej pozície sa potom, po uplynutí každého časového kroku a zobrazí údaj o čase v nasledovnom tvare :

HH:MM:SS

kde HH sú hodiny, MM sú minúty a SS sú sekundy.

TIMESTEP - nastavenie kroku času.

syntax :

TIMESTEP výr1

kde výr1 je aritmetický výraz, ktorého hodnota je po orezaní desatinnej časti z intervalu 1 až 39.

Príkazom **TIMESTEP** nastavujeme časový krok v sekundách (nastavujeme vlastne interval v akom bude prerušovaný programovací jazyk **GBASIC**). Po vykonaní **TIMESTEP** sa znovu nastavia obvody 8253 a 8214 t.j. ak príkaz **TIMESTEP** vykonávame uprostred časového kroku, potom dôjde ku posunutiu aktuálneho času !

TIMEDIS - zákaz zobrazovania času.

syntax :

TIMEDIS

Tento príkaz zakáže zobrazovanie času. Hodiny na **PP01** bežia ďalej, ale informácia o čase sa nebude zobrazovať.

ALARM - nastavenie času pre alarm, pričom po alarme sa spustí program od zadaného riadku príkazom **RUN**.

syntax :

ALARM výr1, výr2, výr3, výr4

kde :

výr1 udáva po vyčíslení a orezaní desatinnej časti údaj pre hodiny

výr2 udáva po vyčíslení a orezaní desatinnej časti údaj pre minúty

výr3 udáva po vyčíslení a orezaní desatinnej časti údaj pre sekundy

výr4 udáva po vyčíslení a orezaní desatinnej časti číslo riadku, od ktorého má byť program spustený

ALARMS - nastavenie času pre alarm, pričom po alarme sa spustí program od zadaného riadku príkazom GOTO.

syntax :

ALARMS výr1, výr2, výr3, výr4

kde :

výr1 udáva po vyčíslení a orezaní desatinnej časti údaj pre hodiny

výr2 udáva po vyčíslení a orezaní desatinnej časti údaj pre minúty

výr3 udáva po vyčíslení a orezaní desatinnej časti údaj pre sekundy

výr4 udáva po vyčíslení a orezaní desatinnej časti číslo riadku, od ktorého má byť program spustený

Po vykonaní príkazu ALARM alebo ALARMS sa v obslužnom podprograme (pre obsluhu prerušenia od časovača) začne kontrolovať, či aktuálny čas nie je väčší alebo rovný času nastavenému v príkazoch ALARM alebo ALARMS. Ak je, potom je doteraz bežiaci program v Basicu prerušený a spustený program v Basicu od riadku, ktorého číslo udáva výr4.

Po vykonaní príkazu ALARM prebehne spustenie príkazom RUN (t.j. zruší sa tabuľka premenných atď).

Po vykonaní príkazu ALARMS prebehne spustenie príkazom GOTO (t.j. všetko, čo bolo vytvorené programom v Basicu zostane zachované, okrem informáciách o cykloch a podprogramoch).

Pri používaní príkazu ALARMS si treba uvedomiť, že prerušenie od časovača môže prísť napríklad aj počas nastavovania nejakej premennej a ak potom príkaz ALARMS spustí program od zadaného riadku, môže zostať nastavovanie premennej nedokončené. Preto je výhodnejšie používať príkaz ALARM.

Príkaz ALARMS môžeme potom používať na spúšťanie krátkych programov, ktorých činnosť sa skončí do najbližšieho alarmu.

Adresy, kde sú uložené jednotlivé časové údaje.

adresa 61H - údaj pre sekundy aktuálneho času
adresa 62H - údaj pre minúty aktuálneho času
adresa 63H - údaj pre hodiny aktuálneho času

adresa 64H - údaj pre sekundy alarmu
adresa 65H - údaj pre minúty alarmu
adresa 66H - údaj pre hodiny alarmu

adresa 69H - obsah rovný nule alarm nenastavený
 obsah rôzny od nuly alarm nastavený

adresa 6DH - obsah rovný nule po alarme spusti ako RUN výr4
 obsah rôzny od nuly po alarme spustí ako GOTO
 výr4

Údaje pre hodiny minúty a sekundy sa uchovávaajú v BCD formáte.

Príklad :

```
10 TIMESET 0,0,0 : ALARM 0,15,0,5000
```

```
.  
. nejaký program v Basicu
```

```
.  
4999 STOP
```

```
5000 TIMESET 0,0,0 : ALARM 0,15,0,5000
```

```
5010 PRINT "UPLYNULO 15 MINUT"
```

tento program vypíše každých 15 minút text :

```
UPLYNULO 15 MINUT
```

Stručný prehľad príkazov pre ovládanie hodín na PP01.

TIMESET výr1,výr2,výr3 - nastavenie času

TIMEEN výr1,výr2 - povolenie zobrazovacieho času

TIMESTEP výr1 - krok času

TIMEDIS - zákaz zobrazovania času

ALARM výr1,výr2,výr3,výr4 - po alarme ako RUN výr4

ALARMS výr1,výr2,výr3,výr4 - po alarme ako GOTO výr4

kde :

výr1, výr2, výr3, výr4 - sú ľubovoľné aritmetické výrazy, ktorých hodnoty sú orezávané o desatinnú časť a kontrolované podľa intervalov povolených pre jednotlivé príkazy.

Prepojky, ktoré je nutné urobiť na IO konektore, aby bežali hodiny na PP01.

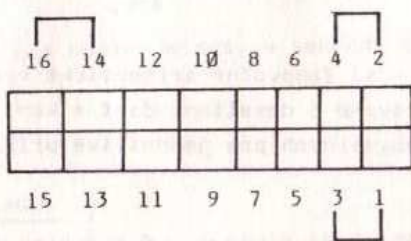
treba spojiť :

pin 2 s pinom 4

pin 1 s pinom 3

pin 14 s pinom 16

IO konektor :



POZOR !

Na pin 16 treba priviesť signál INTE z mikroprocesora. Pre napojenie sa na tento signál neexistuje ovijací trn ani iný prostriedok pre pripojenie, ale treba sa pripojiť priamo na plošný spoj !

Pin 16 IO konektora je voľný, preto bol zvolený na pripojenie signálu INTE. Signál inte pripájame zvnútra na pin 16 FRB konektora.