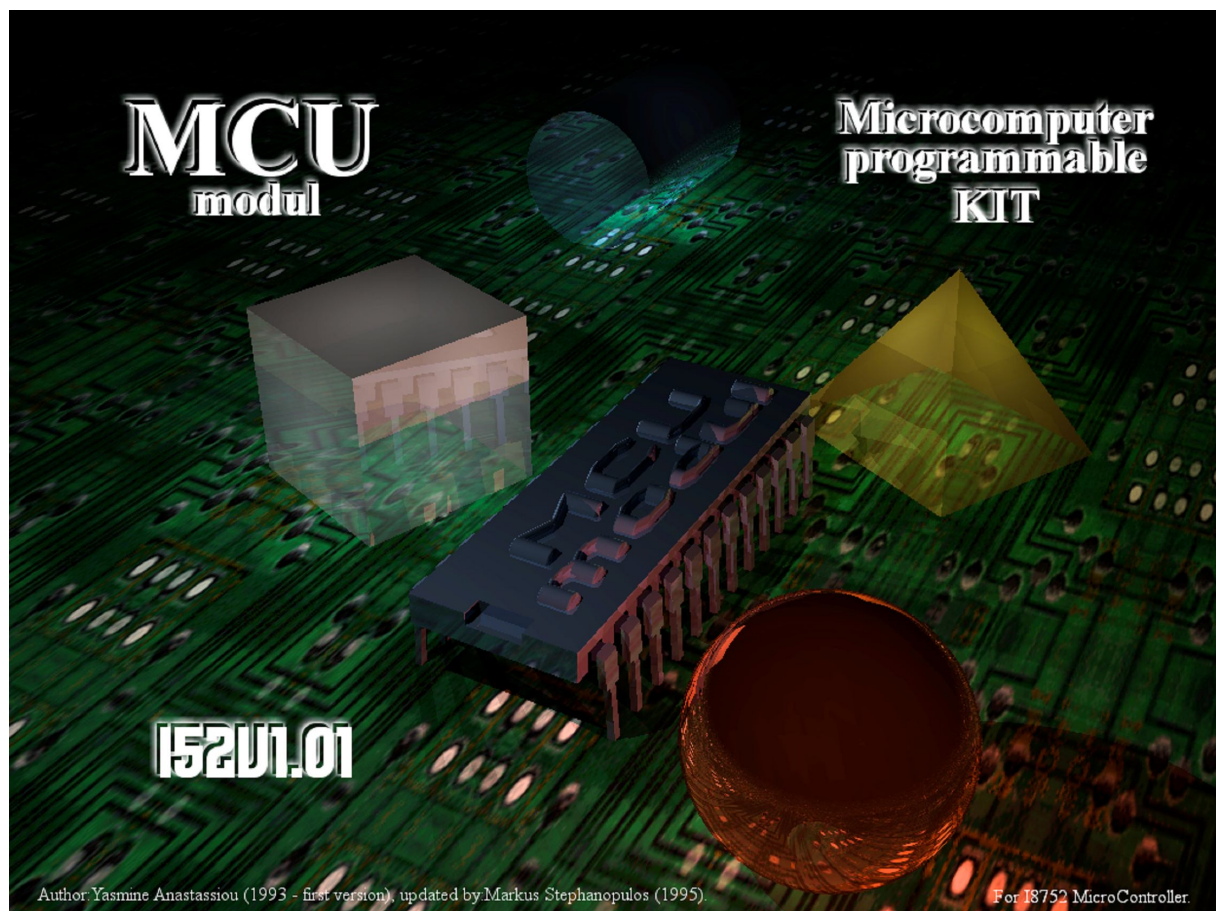


mcontrollers.com

„ Túto stavebnicu som vymyslela pre môjho 58-ročného otca. Rád stavia elektro-konštrukcie s MCU, no už sa nedokáže kôli veku naučiť assembler. Chce vidieť výsledky svojej práce, a to okamžite, napísaním niekoľko riadkov programu, čo v assembleri nieje možné. Preto je MCUmodul určený pre laikov, nie programátorov... “

Yasmine Anastassiou



UPOZORNENIE

Pôvodní autori nepovoľujú stavbu MCUmodulu pre žiadne komerčné účely, len pre individuálne použitie.

Nikto z autorov nenesie zodpovednosť za žiadne škody nijakého druhu spôsobené používaním MCUmodulu.

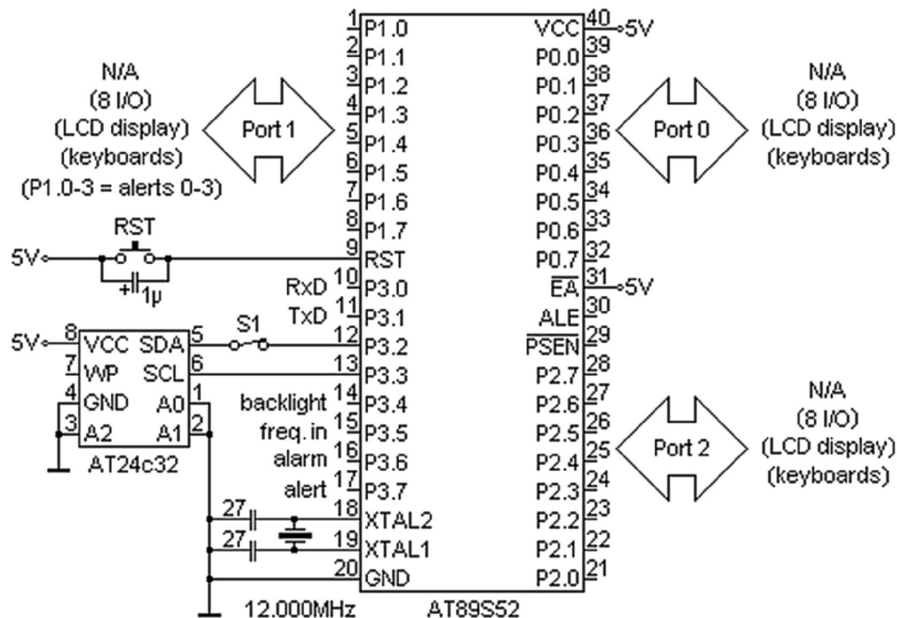
Díky patrí M.Stephanopulovi, za poskytnutie dokumentácie, obrázkov, popisu i samotného SW upraveného pre súčasné kontroléry rady-52.

MCU modul (S52v1.01)

*AUTOR: Yasmine Anastassiová (1993)
update: Markus Stephanopoulos (1995) & "Hannibal" (2010)*

Tento modul (pôvodne navrhnutý pre kontrolér - I8752) je zaujímavou alternatívou, ako možno zostaviť pomocou 8 súčiastok základ aplikácie s programom – a to bez nutnosti ovládania assembleru (či jazyka C), pričom k činnosti postačí len bežný terminálový program (bez napáľovačky čipov) a PC. Užívateľ ale musí mať skúsenosti s logickými funkciami a binárnou/hexa sústavou. Jeho použitie môže byť v rôznych zapojeniach s LED, rôzne blikáče, časové spínače, hodiny, alarmy, ovládanie osvetlení, ako jednoduché PLC, hračky, atd... Výhody/nevýhody tohoto modulu sú nasledovné:

- + veľmi jednoduchá, rýchlo zostaviteľná stavebnica s malým počtom súčiastok a malou spotrebou
- + pre použitie nie je potrebné ovládať assembler, nenáročnosť na vývojový čas a hardware
- + vstavaná obsluha riadenia základných užívateľských periférií: klávesnica, display, terminál, zvuk
- + simulované funkcie s prednastavenými reakciami, možnosť zálohovania dát do SEEPROM
- veľmi malá rýchlosť vykonávania inštrukcií (cca 30/sec pri 12Mhz), vhodné pouze pre riadenie časovo nenáročných činností a procesov
- veľmi jednoduchá forma prevedenia zvukových návěstí alarm a alert, nízka freq. zvuku



obr.1 – základné zapojenie MCUmodulu.

Schéma zapojenia

Schéma je na obr.1. Podľa pôvodného popisu zostavila Y.Anastassiová hardwarovo čo najjednoduchšie možné zapojenie. SW využíva len jadro 8052, dajú sa použiť typy Atmel AT89C52/S52, alebo Philips P89C52. Pre použitie rýchlejších MCU napr: AT89C51RD2, alebo P89C66x, je táto aplikácia z môjho pohľadu zbytočnosť – tieto obvody sú drahšie, tým pádom už je na mieste riešiť pre ne konkrétny program v asembleri, alebo C aby boli korektné využité a nakoniec – ani najrýchlejšie MCU rady-51 sa samo o sebe v úlohe simulácie nevyrovná možnostiam súčasných moderných PLC.

Je to ale iba môj názor – díky Y.Anastassiovej už v r.2002 existovali podstatne dokonalejšie modifikácie MCU modulov, určené pre vyššie spomínané obvody, umožňujúce ovládať grafické LCD displeje a pripojenú bežnú PC klávesnicu.

Funkcie

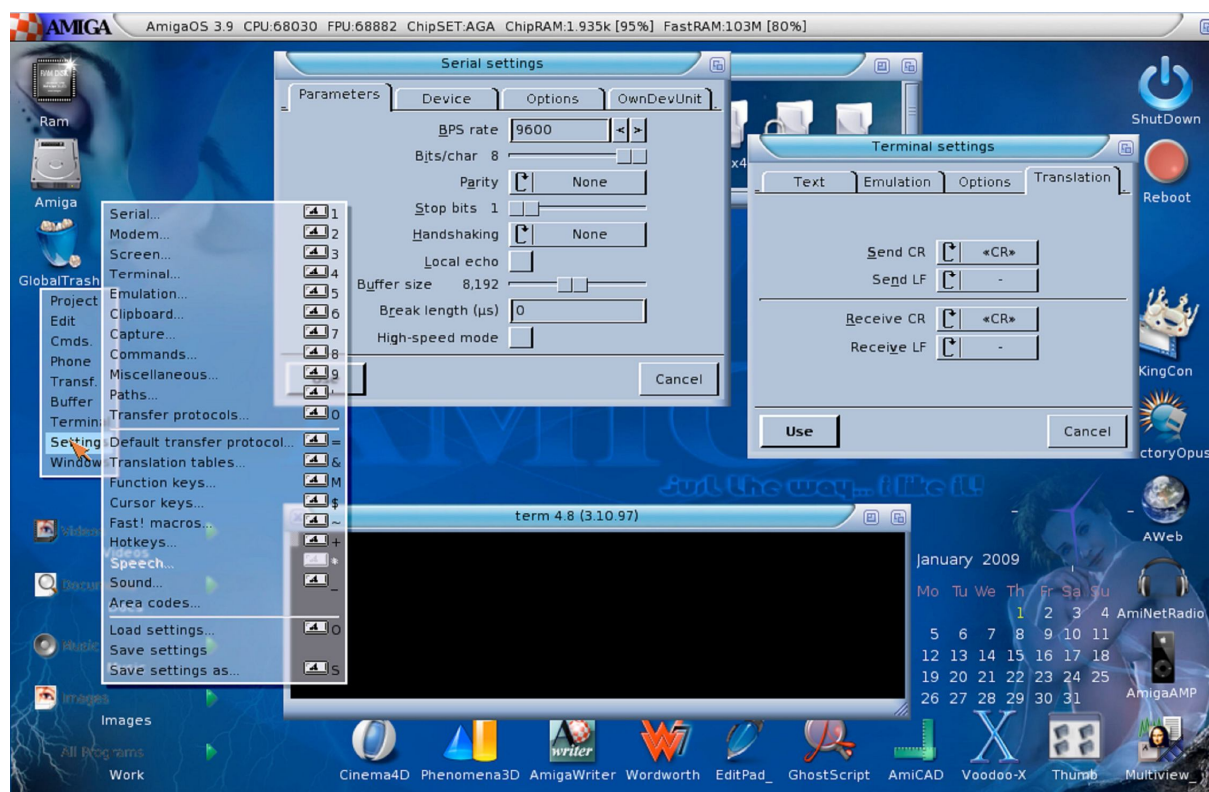
Modul simuluje (alebo obsahuje) nasledovné funkcie:

- Hodiny real. času (24 hod. cyklus) / dátum na 1 bežný (nepriestupný) rok.
- Nastavenie 7 spínacích / vypínacích časov s výberom dňa v týždni.
- 2 alarmy s výberom dňa v týždni.
- 4 linky jednoduchého alertu (poplach s nastaviteľnou prodlevou).
- 4 časovače do 24hod s auto-reloadom obsahu z predvolieb.
- Počítadlo impulzov.
- Nastaviteľný generátor*.
- Nastaviteľný sériový port (1k2 – 19k2).
- Program pre cca. 4k krokov.
- Jednoduché užívateľské heslo* (relatívna možnosť zapojiť viac MCU na spoločnú zbernicu).
- 3 paralelné porty.

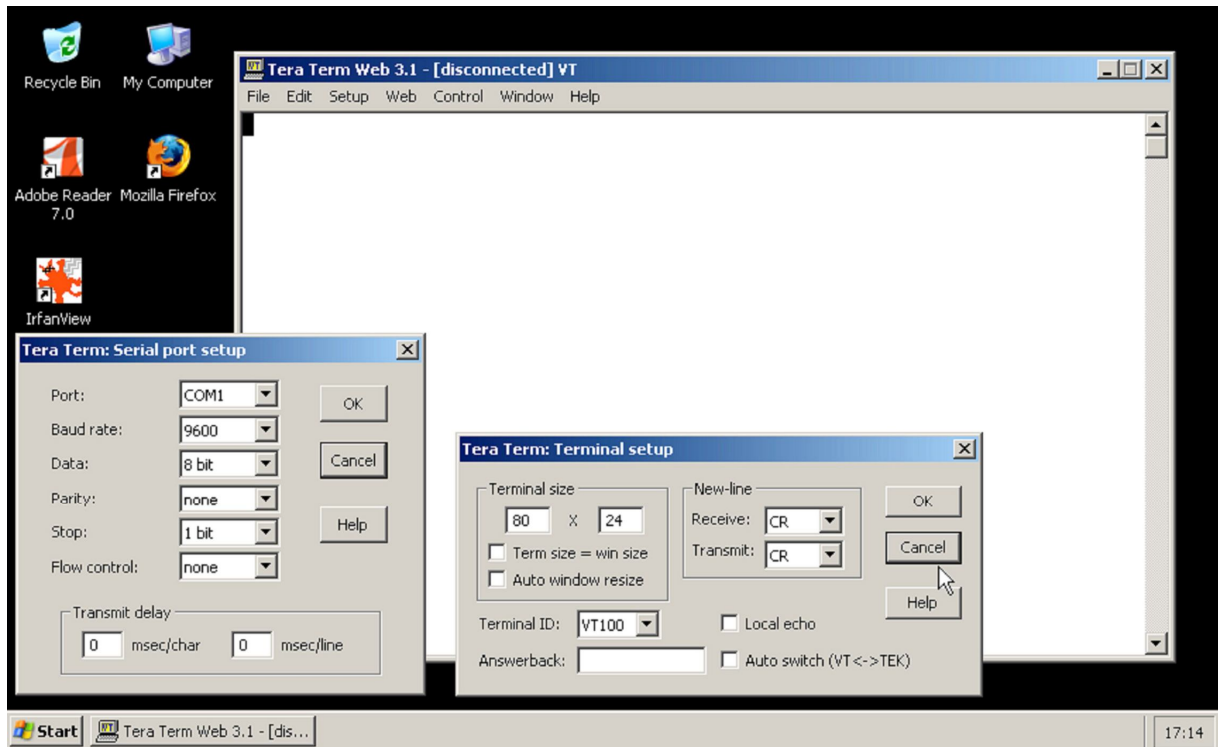
(*pridal M.Stephanopulos)

Pripojenie na počítač

Činnosť modulu je závislá od „programu v SEEPROM“, z ktorej beží. Ten vytvorí užívateľ na základe požadovanej činnosti zariadenia, na ktorého prvky bude modul pripojený. Modul programujeme pripojením do serial-portu počítača. Menu terminálu je relatívne zložitá a neviem ho presne opísať: pre potreby pripojenia stačí previesť „serial“ a „terminal“ nastavenia, viz. obr.2. Doporučená veľkosť bufferu terminálu je 1 – 4kB, aby sme mali k dispozícii aj tzv. „históriu“ prevádzaných akcií. Časové spoždenia medzi vysielanými / prijatými znakmi nie sú nutné. Defaultná rýchlosť serial-portu modulu je 9600BD.

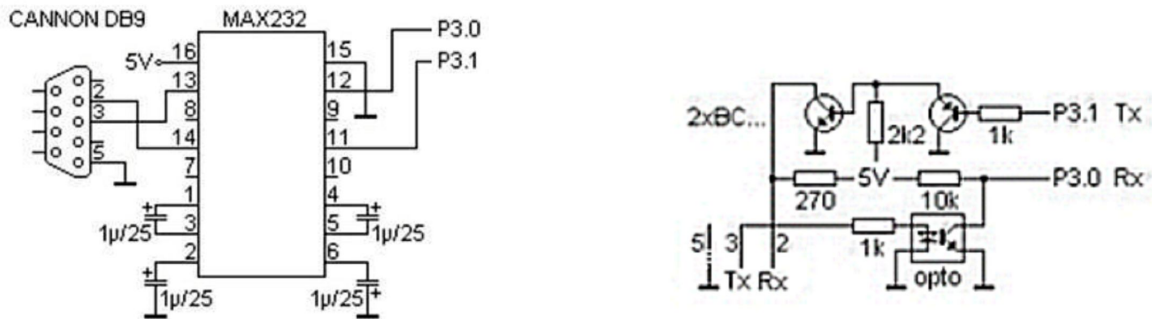


obr.2 – nastavenia terminálu: AmigaOS



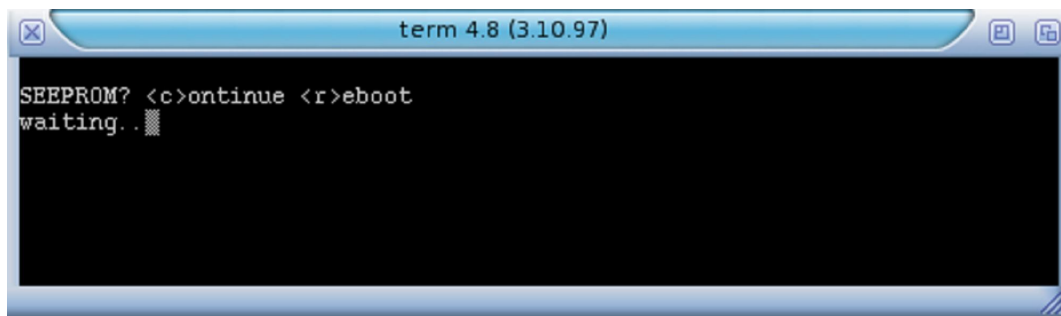
obr.2 – nastavenia terminálu: Windows

Vyskúšané to bolo na dvoch platformách: AmigaOS3.x (program DreamTerm, Term4.8 - <http://aminet.net>) a Windows (program TeraTerm - www.ayera.com). Pripojenie modulu na počítač je možné urobiť pomocou obvodu MAX232 (doporučené), alebo (len skúšobne) z bežných súčiastok – obr.3.

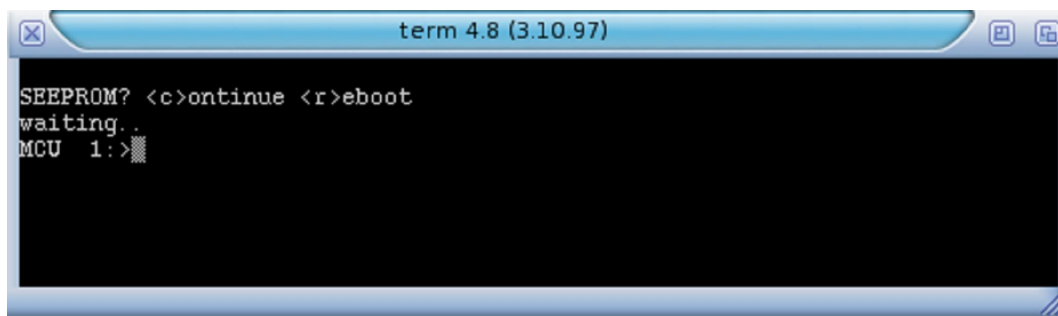


obr.3

Po prvom spustení MCUmodulu (alebo vloženej čistej SEEPROM, či rozopnutom spínači S1 - viz. schéma) sa objaví nasledujúce hlásenie:

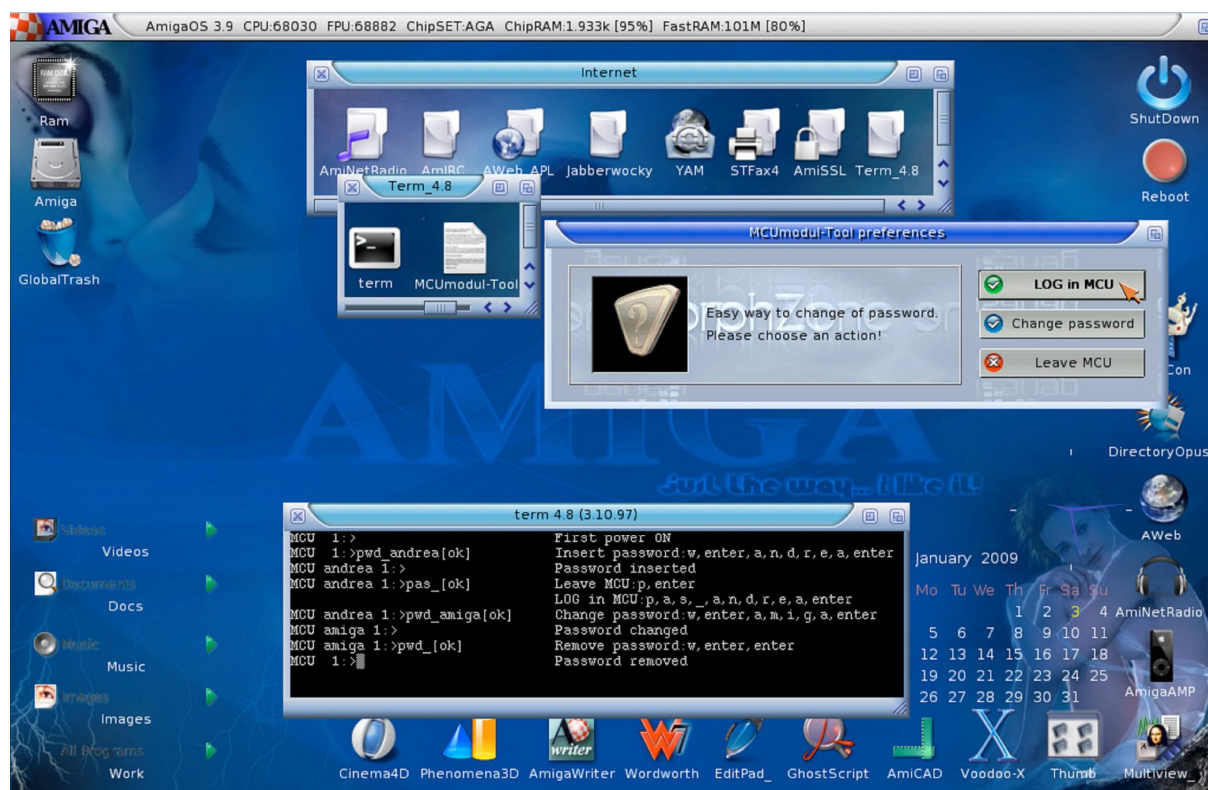


Opustíme ho dvomi spôsobmi: "c" znamená pokračovať, pričom MCU vytvorí a uloží do SEEPROM default-nú konfiguráciu čo trvá cca. 5sec (pri nasledujúcom spustení naskočí normálne), a "r": reset MCU v prípade, že nejaký záskmit pri zapnutí napájania spôsobil chybné načítanie už predtým správne používanej SEEPROM (stalo sa mi to pri 50 pokusoch dvakrát).



Menu

Menu vyvoláme stlačením „h“ (help), pričom sa objaví na termináli ponuka možností. Užívateľské heslo môže byť nastavené, ak chceme zabrániť prístupu dát do/z MCU modulu. Nemá vplyv na beh programu, no MCU "odpovedá" terminálu iba vtedy, ak je zhoda uloženého a prístupového hesla. Po prvom zapnutí je modul aktívny, reaguje na terminál a heslo zadané nieje. Screenshot ukážky komunikácie s počítačom je na obr.4. Na termináli je voľba nového hesla po prvom spustení (heslo v tomto prípade a teda aj meno MCU je "andrea"), odhlásenie, prihlásenie, zmena hesla na iné a odstránenie hesla (Stephanopulos to urobil i použitím AREXX-u).

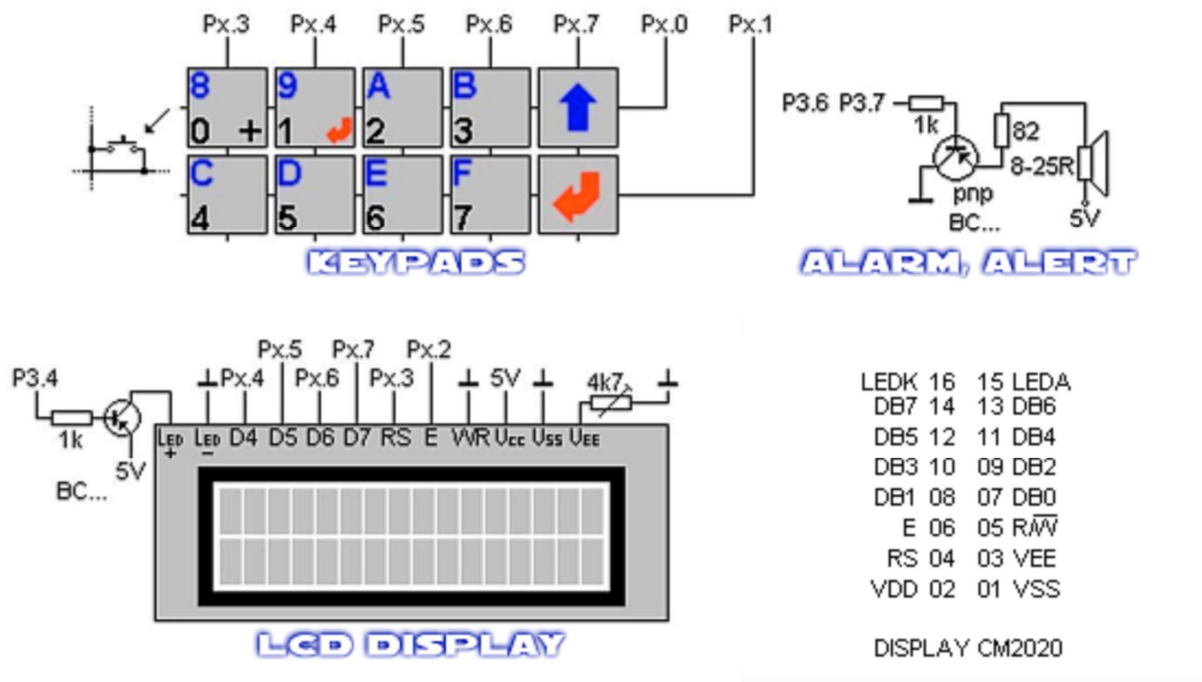


obr.4

Možné pripojenie periférií

Linky všetkých portov reagujú na log. nulu. K MCUmodulu (obr.5) môžu byť pripojené štyri linky alert (výstraha), dve akustické návestia, dva typy klávesníc a LCD display. Alarm / alert sú oddelené z dôvodu možnosti pripojenia výstrahy na externý zosilňovač. Klávesnica môže byť terminál (počítač), hexa (10 tlačidiel),

alebo inkrementálna (iba tlačidlá 0 a 1 = select/enter). Ako display používa typy alfa-numerických LCD (napr. CM2016, CM2020): 2 riadky po 16, alebo 20 znakov. Display a klávesnica **vždy používajú súčasne jeden port.**



obr.5 – Periférie k MCU modulu. „x“ je číslo portu: 0, 1, alebo 2

Písanie programu

Zapojenia s ukážkami programov spravil Markus Stephanopulos, pôvodne ich bolo oveľa viac - no z dôvodu nedostatku času som vybral iba niečo. Komentáre a úpravy textu v programoch som spravil pre lepší prehľad.



Do sequenceru skočíme zo základnej pozície. Sequencer sa zastaví a jeho ukazovateľ adries je nulový. V prvom rade sa uistíte že je pamäť čistá - najlepšie je ju zmazať celú, pretože niektoré bajty, ktoré MCU nepoužíva nie sú zobrazované na termináli - no program môže narazením na ne robiť hlúposti. Predpokladajme, že heslo (meno) MCU modulu nie je vložené. Takže urobíme tieto kroky:

```

MCU 1:> MCU modul zapnutý
MCU 1:>sequ[ok] s, enter (iná klávesa = zrušiť)
0000:cls_0000 c, 0, 0, enter, 0, 0, enter (zmazanie celej SEEPROM)
0000:cls_0000?.....[ok] enter
0000: SEEPROM zmazaná

0000: enter (editovanie)
0000>10 1, 0, enter (číslo príkazu)
0000>init_sdk enter
0000>init_sdk 2 2

0002> adresa je 0002, lebo príkaz "spotreboval" dva bajty pamäte SEEPROM
0002>00 0, 0, enter
0002>disptext enter
0002>disptext adr_04 0, 4, enter
0002>disptext adr_04 Hello!^ H, e, l, l, o, !, enter (znak ^ znamená koniec textu a vkladá ho automaticky)

000B> adresa je 000B, lebo príkaz "spotreboval" 9 bajtov pamäte SEEPROM
000B> enter
000B:
000B:list[ok] l, enter (list - výpis programu, začína vždy od adresy 0000)
    
```

| | |
|----------------------------|---|
| 0000:init_sdk port 2 | 1. riadok programu: inicializácia použitia displeja + klávesnice na porte 2 |
| 0002:disptext adr_04 Hello | 2. riadok programu: vypísanie textu od adresy 04 (horný riadok) |
| 000B | --- |
| 000C | --- |
| | |
| | |
| 00BA | --- |
| 00BB | stlačením ľubovolnej klávesy môžeme výpis prerušiť |
| [ok] | |
| 00BC: | |
| 00BC:quit[ok] | q, enter |
| MCU 1:> | štart programu (vždy od adresy 0000) |



Alfa-numerický displej 2x16 znakov, pripojený na porte 2 – pre program výpisu textu na určenej pozícii displeja stačili 2 riadky. Horný riadok u 2-riadkových LCD začína adresou 00, spodný adresou 40.

UPOZORNENIE:

Je treba si uvedomiť, že MCU modul je značne zastaraná aplikácia určená pre jednoduché použitie a nemožno zrovnávať editor programu s kompilátorom, ktorý automaticky kontroluje správnosť a upozorní na každú chybu. Pri úprave údajov v SEEPROM neexistuje žiadna kontrola správnosti, preto je nutné dodržať tieto zásady:

Pri mazaní príkazov pomocou **cls_xxxx** vkladajte len adresy zobrazujúce sa na termináli počítača po "vylistovaní", nie neexistujúce, inak môže dôjsť k odmazaniu len časti (nie celého) príkazu, alebo bloku príkazov a tým pádom k pomýleniu vykonávania programu zo SEEPROM. Tak isto – **insert/delete** používajte iba na vkladanie/mazanie prázdnych miest (bajtov), nie príkazov programu!

Rovnako pri používaní príkazov **obsahujúcich adresu skoku** vkladajte len adresy zobrazujúce sa na termináli po "vylistovaní", nie neexistujúce, inak sa vykonávanie programu môže pomýliť, tak isto pre príkaz **useflash** používajte len doporučené adresy rutin! Program ju nájde a v „liste“ sa objaví potom namiesto adresy jej názov.

Aplikácie

Jednoduché hodiny s LCD displejom a klávesničkou, pripojenými podľa obrázkov zapojení na port P1 MCU.

| | |
|-----------------------------------|---|
| 0000:init_sdk port 1 | inicializuje/zmaže display + klávesničku pre použitie na porte P1 |
| 0002:disptext adr_00 DEMO-hodiny^ | výpis textu |
| 0010:loadbyte 1B data 44 | vloženie začiatku zobrazenia pre FLASH-utilitu |
| 0013:useflash :dispCLCK | FLASH-util: zobrazí hodiny HH:MM:SS |
| 0016:loadbyte 1B data 28 | vloženie adresy skoku v stránke na adr.xx28 pre FLASH-utilitu |
| 0019:useflash :jumpKEYB | FLASH-util: skočí v stránke ak bude stlačené tlačidlo klávesničky |
| 001C:jump_adr :0010 | skok nazad do zobrazovacieho cyklu |
| 0028:disptext adr_00 Nastav cas:^ | výpis textu |
| 0036:loadbyte 1B data 44 | vloženie začiatku zobrazenia pre FLASH-utilitu |
| 0039:useflash :editCLCK | FLASH-util: edituje hodiny HH:MM:SS |
| 003C:jump_adr :0000 | skok nazad do cyklu |

DEMO-hodiny
10:34:17

Nastav cas:
00:34:17

Ak je 10 – tlačidlová klávesnica zbytočne veľká, stačí upraviť pôvodný program a pripojiť len tlačidlá 0 a 1.

```
MCU 1:>sequ[ok]
0000:adr_0039
0039:
0039:ins?[ok]
0039:ins?[ok]
0039:
0039>modekeyb 1      (0 = Hexa:10 tlačidiel, 1 = Duo: dve tlačidlá)
003B>
003B:list
...
...
0036:loadbyte 1B data 44
0039:modekeyb D
003B:useflash :editCLCK
003E:jump_ adr :0000
[ok]
```

Zvuková signalizácia po dopočítaní časovača 1 z predvoleného času na nulovú hodnotu. Vypína sa klávesničkou.

| | |
|--------------------------------|--|
| 0000:init_sdk port 1 | inicializuje/zmaže display + klávesničku pre použitie na porte P1 |
| 0002:disptext adr_05 TIMER1^ | výpis textu |
| 000B:disptext adr_42 cas ^ | výpis textu |
| 0012:loadbyte 1B data 46 | vloženie začiatku zobrazenia pre FLASH-utilitu |
| 0015:useflash :editTMRS | FLASH-util: edituje obsahy časovača 0,1,2, alebo 3 stlačením tlačidla |
| 0018:movebyte 90 byte 75 | kontrola, či sú vložené údaje do časovača 1 |
| 001B:orl_byte 90 byte 74 | kontrola, či sú vložené údaje do časovača 1 |
| 001E:if_ =byte 90 data 00:0000 | ak je časovač 1 aj po vložení údajov stále nulový, skok nazad na začiatok |
| 0023:loadbyte 73 data 00 | vynulovanie sekúnd časovača 1 |
| 0026:set__bit 5 byte 18 | štart časovača 1 |
| 0029:useflash :disp TM1 | FLASH-util: zobrazí stav časovača 1 (na tej istej pozícii, ako jeho editovnie) |
| 002C:if_0=bit 5 byte 19:0029 | ak časovač nedosiahol nulovú hodnotu, skok nazad do cyklu |
| 0031:set__bit 5 byte 17 | časovač je nulový, zapnutie zvuku na výstupe pre „alert“ |
| 0034:useflash :waitKEYB | zvuk písa, čaká sa na stlačenie nejakého tlačidla obsluhy... |
| 0037:clr__bit 5 byte 18 | vypnutie časovača 1 |
| 003A:clr__bit 5 byte 19 | vypnutie zvuku |
| 003D:jump_ adr :0000 | návrat nazad do cyklu |

| |
|----------------------------|
| TIMER1 cas 00:00 |
|----------------------------|

| |
|-------------------------------|
| TIMER1 cas 00:02:54 |
|-------------------------------|

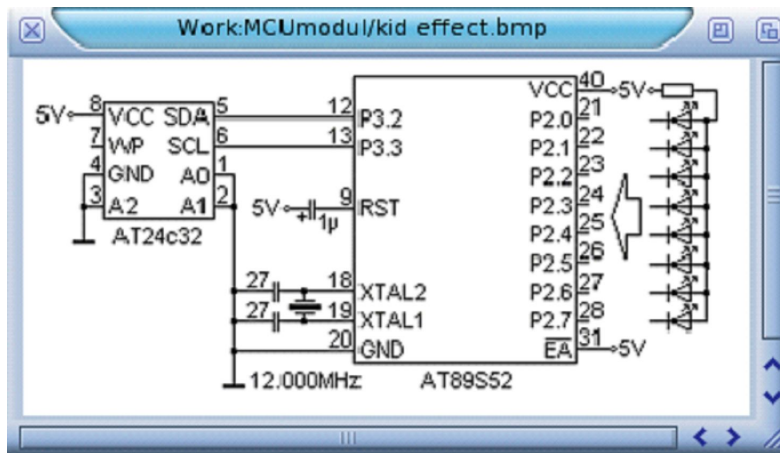
Časovač po dosiahnutí 00:00:00 „zdanlivo“ stojí. Dochádza však pri jeho vynulovaní k prednastaveniu obsahu predvoľby do časovača a tá je nulová. Ak by bola nastavená na určitý čas, mohol by sa časovač využívať na upozorňovanie opakujúcich sa časových úsekov.

LED efekt "kid raider" - pendlujúci svetelný bod z ľava do prava a nazad. LEDky sú pripojené na port P2.

| | |
|--------------------------|------------------------------|
| 0000:set__bit 6 byte 1A | nastav port 2 ako výstup |
| 0003:set__bit 7 byte 1A | nastav bit FLAG na 1 |
| 0006:loadbyte 3F data FE | rozsvietenie prvej LED diody |

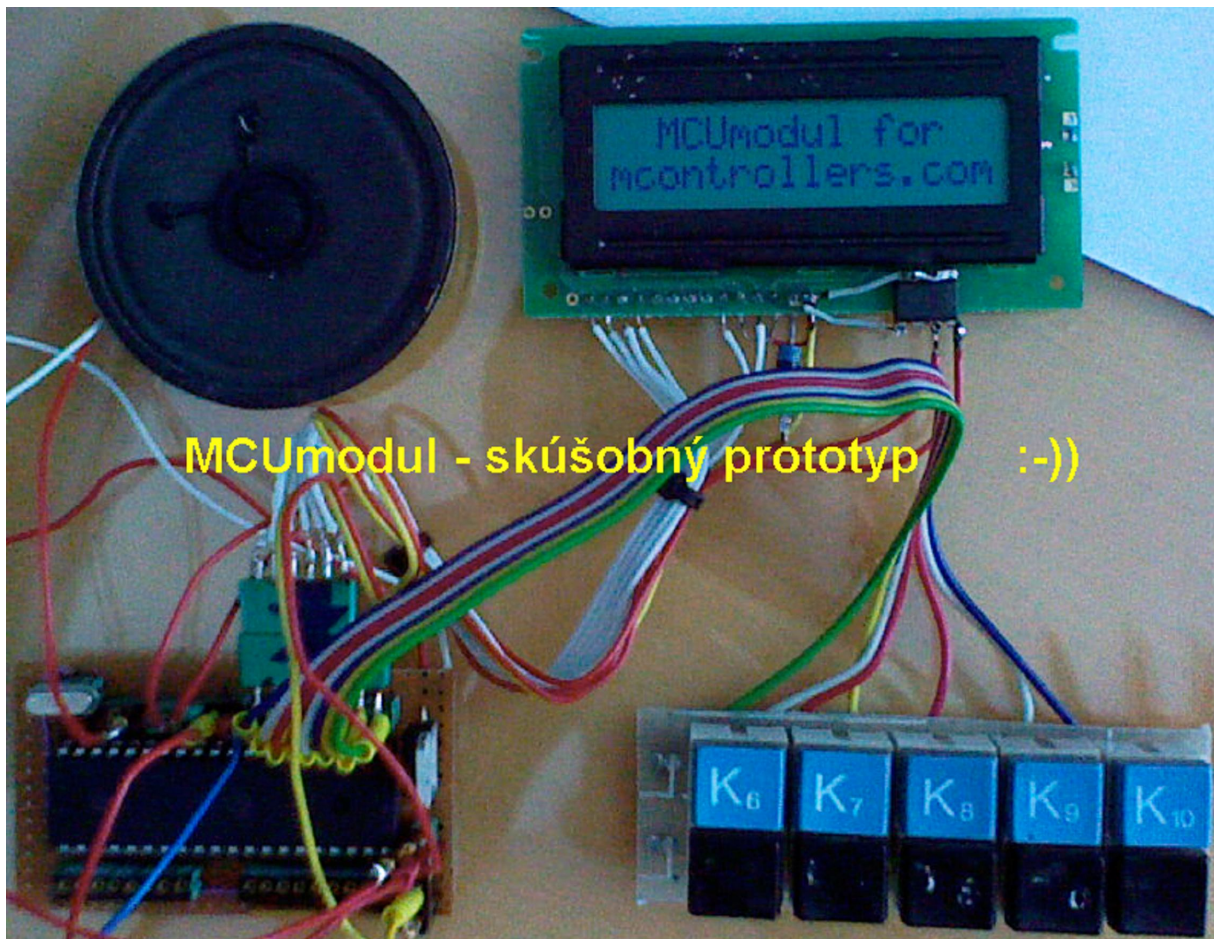


| | |
|------------------------------|--|
| 0009:rotlbyte 3F | rotácia portu 2 o 1 pozíciu vľavo |
| 000B:if_1=bit 7 byte 1A:0009 | ak nebola presiahnutá 8-sma dioda, vráť sa na 0009 |
| 0010:rotrbyte 3F | rotácia portu 2 o 1 pozíciu vpravo |
| 0012:if_1=bit 7 byte 1A:0010 | ak nebola presiahnutá prvá dioda, vráť sa na 0010 |
| 0017:jump_ adr :0006 | skok do východiskovej pozície rozsvietenia |



Pripojenie LED diod na port P2.

Z 15 aplikácií som vyskúšal len 3 uvedené (viac sa mi nechcelo ☺), k čomu som si zhotovil malý „prototyp“. V závere manuálu je uvedených niekoľko info pre prípadných záujemcov a tabuľka prílohy pre písanie programov.



Obrázok prototypu na univerzálnych doskách pl.spoja.

Info

Činnosti sequenceru v súvislosti s terminálom, ak je MCU: 1. odhlásený, 2. prihlásený a 3. prihlásený a vôjdem do niektorej položky – pas_ pwd_ ...atd:



1. MCU normálne pracuje, vykonáva program, ale prípadné reakcie na terminál sú blokované.
2. MCU normálne pracuje, vykonáva program a reakcie na terminál sú zobrazované na monitore.
3. Vojdením do akejkoľvek položky sa vykonávanie programu pozastaví, po opustení položky pokračuje ďalej. Pri vojení /vyskočení zo sequenceru sa program zastaví /reštartuje od adr. 0000H.

Je možné pripojiť i inú (väčšiu/menšiu) pamäť k MCU?

Vyhovujú všetky typy SEEPROM typu AT24Cxxx (32,64,128,256,512), žiaľ kapacita programovej - užívateľskej pamäte sa nad 4kB nezvýši. Pamäť vydrží cca. 100k zápisov s dobou pamätania okolo 40 rokov

Čo znamená jednotka za názvom (heslom) MCU: MCU xxxxxxxx 1:>

Je to indikácia oznamujúca, že prípadné dáta budú vysielané smerom k terminálu. Ak by sme použili príkaz zadania hodnoty opačne - teda z terminálu do MCU, indikátor sa zmení na dvojku - tj. MCU čaká a požaduje vloženie dát: "MCU xxxxxxxx 2:>". Zmena jednotky na dvojku nenastane iba v prípade použitia niektorých volaní "voľného" snímania useflash rutín, ktoré slúžia pre vetvenie programu.

Info o použití portov:

Každý port nakonfigurujeme ako vstupný / výstupný nastavením riadiaceho bitu v prípade, že je používaný ako bežný vstup / výstup. Príkaz "init_sdk" ho obsadzuje pre riadenie displaya / klávesničky, pričom mu budú presmerované všetky činnosti s nimi súvisiace – v tomto prípade dáta čítané /zapisované na adresu portu budú bez reakcie na fyzické piny puzdra obvodu. Port P1 má okrem toho ešte snímanie liniek "alert", ktoré sa aktivujú nastavením ich príslušných bitov. Ak port P1 používa display / klávesnička, linky alertu sú i napriek aktivovaniu ignorované. Pri používaní portu ako vstup / výstup sú však normálne brané v úvahu – preto ak je P1 nastavený ako výstupný, treba si dať pozor, aby nedošlo k vyvolaniu funkcie alert vlastným užívateľským programom. Port P0 je otvorený kolektor, preto ho treba ošetriť "pull-up" 10kΩ rezistormi na vetvu +5V. Ak je niektorý port používaný ako výstupný, pred zmenou na vstupný do neho treba vložiť data FFH. Dôvod = architektúra 8051.

Čo je konfigurácia (CFG):

Príkazy pracujúce s konfiguráciou ukladajú dáta z RAM MCU do SEEPROM: predvoľby podsvitu displaya, dočasovania, dáta spínacích /vypínacích /alarmových časov a dní ich použitia, predvoľby spoždení liniek funkcie alert a predvoľby timerov 0-3. Neukladá sa údaj o baudovej rýchlosti serial portu.

Koľko úrovní má simulovaný zásobník pre adresy volania (CALL)?

15 ponorení.

Je v aplikácii MCU modul počítané s možnosťou vloženia nezmyselných dát?

MCU prevádza automaticky kontrolu správnosti. Ak dátum, alebo mesiac presahujú logicky správnu hodnotu, budú nahradené údajom 01. Pre deň: 00-06 (po-ne). Iný údaj vložený na pozíciu dňa bude = 00. Hodiny: 00-23, minuty / sekundy: 00-59, časovania alarmov: 00-99. Iný údaj na týchto pozíciách bude nahradený hodnotou 00.

Tipy & triky

Program sa zacykloval a MCU nereaguje na terminál:



1. Stlačiť tlačidlo RST, rozopnúť spínač S1 a RST uvoľniť.
2. Po nabehnutí modulu stlačiť klávesy "s" a "enter".
3. Zopnúť spínač S1, a príkazmi terminálu pre sequencer problém v programe nájsť a odstrániť.

Čo robiť, ak zabudnem heslo:



1. Stlačiť tlačidlo RST, rozopnúť spínač S1 a RST uvoľniť.
2. Po nabehnutí modulu zopnúť spínač S1.
3. Stlačiť klávesy "w", "enter", "enter".

PRÍLOHA: nedotknuté bity

Sú obsiahnuté v 4 roch po sebe nasledujúcich bajtoch a reagujú, alebo ovplyvňujú činnosť simulovaných funkcií. Ich správne použitie je kľúčom k celému programovaniu MCU modulu.

170/bit-0 ON – časy
170/bit-1 OFF – časy
170/bit-2 a/n – ON/OFF
170/bit-3 zvuk alarm
170/bit-4 a/n – alarmové časy
170/bit-5 zvuk alert
170/bit-6 a/n – delay 0
170/bit-7 a/n – delay 1

nastavi MCU, ak je hociktorý spinací čas a deň zhodný s hodinami
nastavi MCU ak je hociktorý vypinací čas a deň zhodný s hodinami
nastavený = sú akceptované spinacie / vypinacie časy a dni
nastavi MCU ak je hociktorý alarmový čas a deň zhodný s hodinami
nastavený = sú akceptované navolené alarmové časy a dni
nastavi MCU po uplynutí času vzniku niektorého alertu
nastavený = port P1.0 spusti odpočítavanie času pre vznik alertu 0
nastavený = port P1.1 spusti odpočítavanie času pre vznik alertu 1

180/bit-0 a/n – delay 2
180/bit-1 a/n – delay 3
180/bit-2 čítanie CT
180/bit-3 nulový CT
180/bit-4 beh/stop TM 0
180/bit-5 beh/stop TM 1
180/bit-6 beh/stop TM 2
180/bit-7 beh/stop TM 3

nastavený = port P1.2 spusti odpočítavanie času pre vznik alertu 2
nastavený = port P1.0 spusti odpočítavanie času pre vznik alertu 3
nastavený = neprepisuje obsah počítadla do vyrovnávacej pamäte
nastavený = drží obsah počítadla impulzov vynulovaný
nastavený = beží časovač 0
nastavený = beží časovač 1
nastavený = beží časovač 2
nastavený = beží časovač 3

180/bit-0 –
180/bit-1 –
180/bit-2 –
180/bit-3 –
180/bit-4 nulový TM 0
180/bit-5 nulový TM 1
180/bit-6 nulový TM 2
180/bit-7 nulový TM 3

pre užívateľské použitie (rezerva)
pre užívateľské použitie (rezerva)
pre užívateľské použitie (rezerva)
pre užívateľské použitie (rezerva)
nastavi MCU, ak má počas behu nulovú hodnotu časovač 0
nastavi MCU, ak má počas behu nulovú hodnotu časovač 1
nastavi MCU, ak má počas behu nulovú hodnotu časovač 2
nastavi MCU, ak má počas behu nulovú hodnotu časovač 3

1A0/bit-0 signál alertu 0
1A0/bit-1 signál alertu 1
1A0/bit-2 signál alertu 2
1A0/bit-3 signál alertu 3
1A0/bit-4 smer portu 0
1A0/bit-5 smer portu 1
1A0/bit-6 smer portu 2
1A0/bit-7 FLAG

nastavi MCU, ak je alert 0 zaznamenaný ako aktívny na vstupe P1.0
nastavi MCU, ak je alert 1 zaznamenaný ako aktívny na vstupe P1.1
nastavi MCU, ak je alert 2 zaznamenaný ako aktívny na vstupe P1.2
nastavi MCU, ak je alert 3 zaznamenaný ako aktívny na vstupe P1.3
0=vstupný, 1=výstupný
0=vstupný, 1=výstupný
0=vstupný, 1=výstupný
nastavuje MCU použitím inkrementu, dekrementu, rotácií a porovnaní

PRÍLOHA: useflash rutiny

Sú to programové rutiny vo FLASH MCU, vykonávajúce časti programu, ktoré by boli buď zdlhavo realizovateľné inštrukciami MCU modulu, alebo by ich nimi nebolo možné previesť vôbec. Každá verzia MCU modulu má inú – Y. Anastassiou ich vytvorila pre verziu C668V1.5 - čo je MCU modul osadený s P89C664 asi 50. Pre základný MCU modul C52V1.01 sú k dispozícii adresy pre tieto:

0EFC: sendCLK
0EFC: sendDATE
0EFC: dispCLK
0EFC: dispDATE
0EFC: sendCLF
0EFC: jump TTY

vypíše na terminal hodiny HH:MM:SS
vypíše na terminal DEN/DAT.MES
vypíše hodiny HH:MM:SS na display. Pozícia zobrazenia = adr.1Bh
vypíše dátum DEN/DAT.MES na display. Pozícia zobrazenia = adr.1Bh
vyšie CR + LF na terminal
ak je klávesnica terminálu uvoľnená, tak návrat bez reakcie. Ak je prijatý serialom nejaký znak, tak data na adr.1Bh sú brané ako nižšia adresa stránky z ktorej useflash voláme a prevedie sa skok na túto adresu. Následne sa uloží prijatý znak z terminálu na adr.1Bh. Táto rutina reaguje pomalšie, takže treba podržať dlhšie stlačenú klávesu na počítači (cca 1 sec.), aby terminálový program poslal po sebe opakovane viac znakov.

0EFC: send TTY
0EE2: waiting
0EE3: jumpKEYB

vyšie data z adresy 1Bh cez serial port.
časová prodleva, závisiaca od veľkosti dat na adr.1Bh (00h=min,FFh=max - trvá viac ako minútu).
ak sú tlačidlá klávesníčky uvoľnené, tak návrat bez reakcie, inak je stav adresy 1Bh braný ako nižšia adresa stránky z ktorej useflash voláme a prevedie sa skok na túto adresu. Následne sa uloží prijatý znak z klávesnice do adr. 1Bh.
prijme bajt z klávesníčky a uloží na adr.1Bh. Ak je klávesnica nestlačená - uvoľnená, čaká...
prijme znak z terminálu a uloží na adr.1Bh. Ak nieje žiadna klávesa počítača stlačená, čaká...
editor hodín. Pozícia zobrazenia = adr.1Bh
editor dátumu. Pozícia zobrazenia = adr.1Bh
editor nastavení predvolieb časovačov T0 – T3. Pozícia zobrazenia = adr.1Bh
editor časovačov T0 – T3. Pozícia zobrazenia = adr.1Bh
editor spinacích časov + dní. Pozícia zobrazenia = adr.1Bh
editor vypinacích časov + dní. Pozícia zobrazenia = adr.1Bh
editor alarmových časov + dní. Pozícia zobrazenia = adr.1Bh
editor zpoždení liniek alert pre vstupy P1.0 – P1.3. Pozícia zobrazenia = adr.1Bh
zobrazí na displeji stav časovača T0. Pozícia zobrazenia = adr.1Bh
zobrazí na displeji stav časovača T1. Pozícia zobrazenia = adr.1Bh
zobrazí na displeji stav časovača T2. Pozícia zobrazenia = adr.1Bh
zobrazí na displeji stav časovača T3. Pozícia zobrazenia = adr.1Bh
zobrazí na displeji stav počítadla CT. Pozícia zobrazenia = adr.1Bh
ak port P1 nepoužíva klávesníčku / display, rutina zapne / vypne generátor – výstup má na P1.0, zároveň ale zastaví serial-port komunikáciu (pridal M.Stefanopoulos)

0EE3: waitKEYB
0EE3: wait TTY
0EE3: editCLK
0EE3: editDATE
0EE4: editPRES
0EE7: editTMS
0EEA: edit ON
0EEB: edit OFF
0EEE: editALAR
0EE3: editUNLAY
0EE3: disp TMO
0EE3: disp TM1
0EE3: disp TM2
0EE3: disp TMS
0EE3: disp CT
0EE3: ---

PRÍLOHA: RAM MCU

00 – 0F:
0C, 0B:
30, 3E, 3F:

užívateľské miesta RAM
H, L – deliaci pomer serial portu / „generátora“
porty P0, P1, P2

PRÍLOHA: popis inštrukcií

Riadia činnosť funkcií MCU a obsluhu "podmienkových" príznakov. Inštrukcie vkladáme terminálom - pod číselným kódom vojdenním do sekvenceru na jednotlivé adresy pamäte EEPROM.

| | |
|----------------------------------|---|
| 00(?)displaytext xx | abcd...^ zobrazí text na displayi od adr.xx, ukončený symbolom ^ |
| 01(?)terminaltext | abcd...^ zobrazí text na termináli, ukončený symbolom ^ |
| 02(3)displaybyte yy xx | zobrazí bajt yy na displayi od adr.xx |
| 03(2)terminalbyte yy | zobrazí bajt yy na termináli |
| 04(3)displaybits yy xx | zobrazí 8 bitov bajtu yy na displayi od adr.xx |
| 05(2)terminalbits yy | zobrazí 8 bitov bajtu yy na termináli |
| 06(3)displaychar yy xx | zobrazí bajt yy jako ascll znak na displayi od adr.xx |
| 07(2)terminalchar yy | zobrazí bajt yy jako ascll znak na termináli |
| 08(4)editbyte yy xx zz | edituje bajt yy na displayi od adr.xx v rozsahu 00-zz dekadicky |
| 09(4)editbyte yy xx zz | edituje bajt yy na displayi od adr.xx v rozsahu 00-zz hexa |
| 0A(3)editbits yy xx | edituje bity bajtu yy na displayi od adr.xx |
| 0B(3)editchar yy xx | edituje bajt yy zápisom hexa konštanty a jej ascll znaku na displ.adr.xx |
| 0C(3)editbyte yy zz | edituje bajt yy terminálom v rozsahu 00-zz dekadicky |
| 0D(3)editbyte yy zz | edituje bajt yy terminálom v rozsahu 00-zz hexa |
| 0E(2)editbits yy | edituje bity bajtu yy terminálom |
| 0F(2)editchar yy | edituje bajt yy zápisom hexa konštanty ascll znaku terminálom |
| 10(2)init_sck p | inicializuje EEPROM/Display/Keyboard pre použitie portu Pp a EEPROM na P3.2/P3.3 |
| 11(2)modekeyb m | voľba módu editovania užívateľskej klávesnice: klasicky/10tl., alebo inkrement/2tl. |
| 12(1)read_efg | načíta užívateľskú konfiguráciu do int.RAM MCU z najvyššej stránky EEPROM 24c32 |
| 13(1)save_efg | uloží užívateľskú konfiguráciu z int.RAM MCU do najvyššej stránky EEPROM 24c32 |
| 14(3)clr_bit b yy | vymaže bit b bajtu yy |
| 15(3)set_bit b yy | nastaví bit b bajtu yy |
| 16(3)opl_bit b yy | neguje bit b bajtu yy |
| 17(3)and_byte yy xx | log.and bajtu yy s bajtom xx |
| 18(3)ori_byte yy xx | log.ori bajtu yy s bajtom xx |
| 19(3)xor_byte yy xx | log.xor bajtu yy s bajtom xx |
| 1A(2)not_byte yy | neguje bajt yy |
| 1B(3)writeflash aaaa | vykoná ľubovoľný podprogram z FLASH MCU na adr.aaaa |
| 1C(2)rotlbyte yy | rotácia bajtu yy vľavo cez carry: bit 7 -> carry, carry -> bitu 0 |
| 1D(2)rotrbyte yy | rotácia bajtu yy vpravo cez carry: bit 0 -> carry, carry -> bitu 7 |
| 1E(3)inc_byte yy n | inkrement bajtu yy v hex/dec sústave |
| 1F(3)dec_byte yy n | dekrement bajtu yy v hex/dec sústave |
| 20(3)movbyte yy xx | presun obsahu bajtu xx do bajtu yy |
| 21(3)loaddata yy dd | zápis dát dd do bajtu yy |
| 22(5)if_0-bit b yy aaaa | ak je bit bajtu yy b=1, skok na adr.aaaa |
| 23(5)if_1-bit b yy aaaa | ak je bit bajtu yy b=0, skok na adr.aaaa |
| 24(5)if_=_byte yy xx aaaa | ak je bajt yy zhodný s bajtom xx, skok na adr.aaaa |
| 25(5)if_=_data yy dd aaaa | ak je bajt yy zhodný s dátami dd, skok na adr.aaaa |
| 26(3)jmp_adr aaaa | skok na adr.aaaa |
| 27(3)call_adr aaaa | volanie na adr.aaaa |
| 28(1)-return- | návrat z volania |
| 29(1)--- | prázdna operácia bez významu |

b= bit 0-7, p= port 0,1,2, m= mód činnosti klávesnice 0-hexa /1-inkrement, n= pre inštr.1E a 1F: 1-desiatková/0-hexa sústava, dd= datová konštanta 00-FFH, xx,yy= ak ide o adr.bajtu: 00-FFH* pričom yy je cieľový bajt, ak ide o adr.displaya, napr. CM2016, tak: 1.znak /1.riadok-00H, 2.znak / 1.riadok-01H... 0FH, 1.znak/2.riadok-40H, 2.znak/2.riadok-41H... 4FH, zz= pre dec. 00-99D, pre hexa 00-FFH, aaaa= adresa 0000-0EFFF

[* - určité obmedzenia, týkajúce sa zákazu prístupu na niektoré RAM adresy]