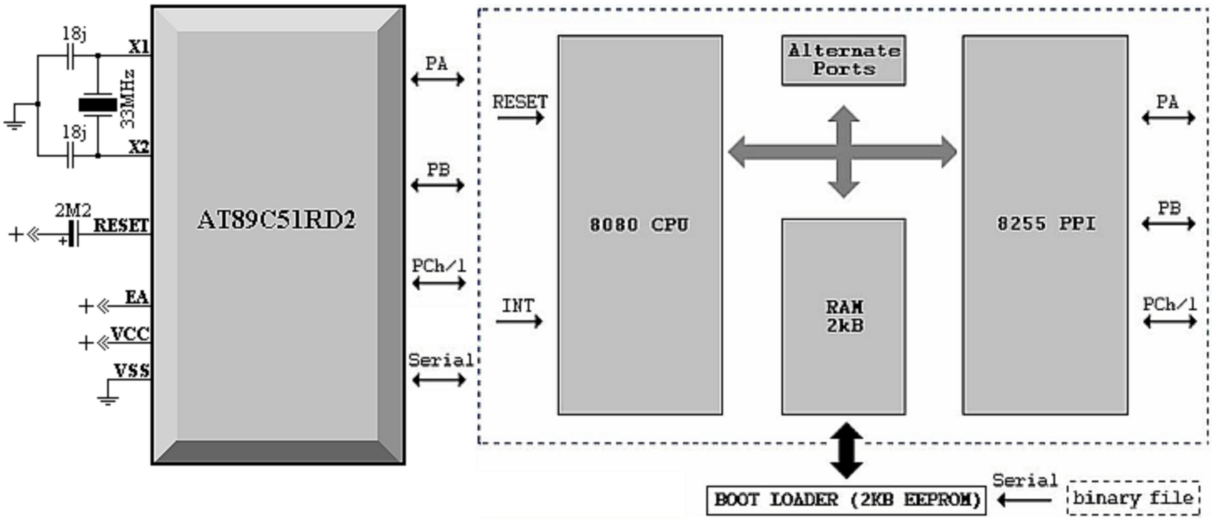

pmi-80
RE
I
EX
R
BR
C
D
DE
E
F
M
8
SP
9
HL
A
AF
B
BC
L
4
5
6
7
S
0
1
2
3
=
under emulation!

PMI-80 under emulation!

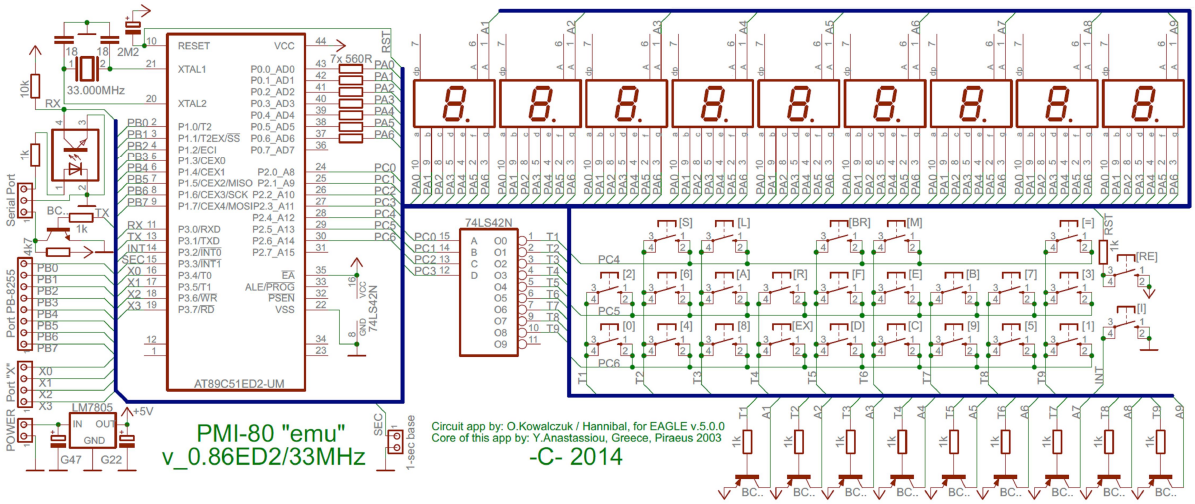
(Y.Anastassiou-2003, O.Kowalczuk / Hannibal-2014)

Áno - čítate správne, a fakt nejde o žart, ☺ hoci - keď mi fotky tejto veci O.Kowalczuk 1.4.2014 poslal, myslel som si, že si robí „prdel“. Za všetkým stojí Y.Anastassiová, známa z jednodeskových mikropočítačov **EMA-51**, **EMA-552**, a tiež **Hellena-51**. Uvolnila k stiahnutiu svoju aplikáciu (z r.2003!) s názvom „**MINIsystem-8080Emulator**“, pod ktorou **PMI-80** beží... Tu je zosumarizované, o čo ide:



obr.1 - schéma a interná hierarchia aplikácie „MINIsystem-8080Emulator“ (v0.86ED2)

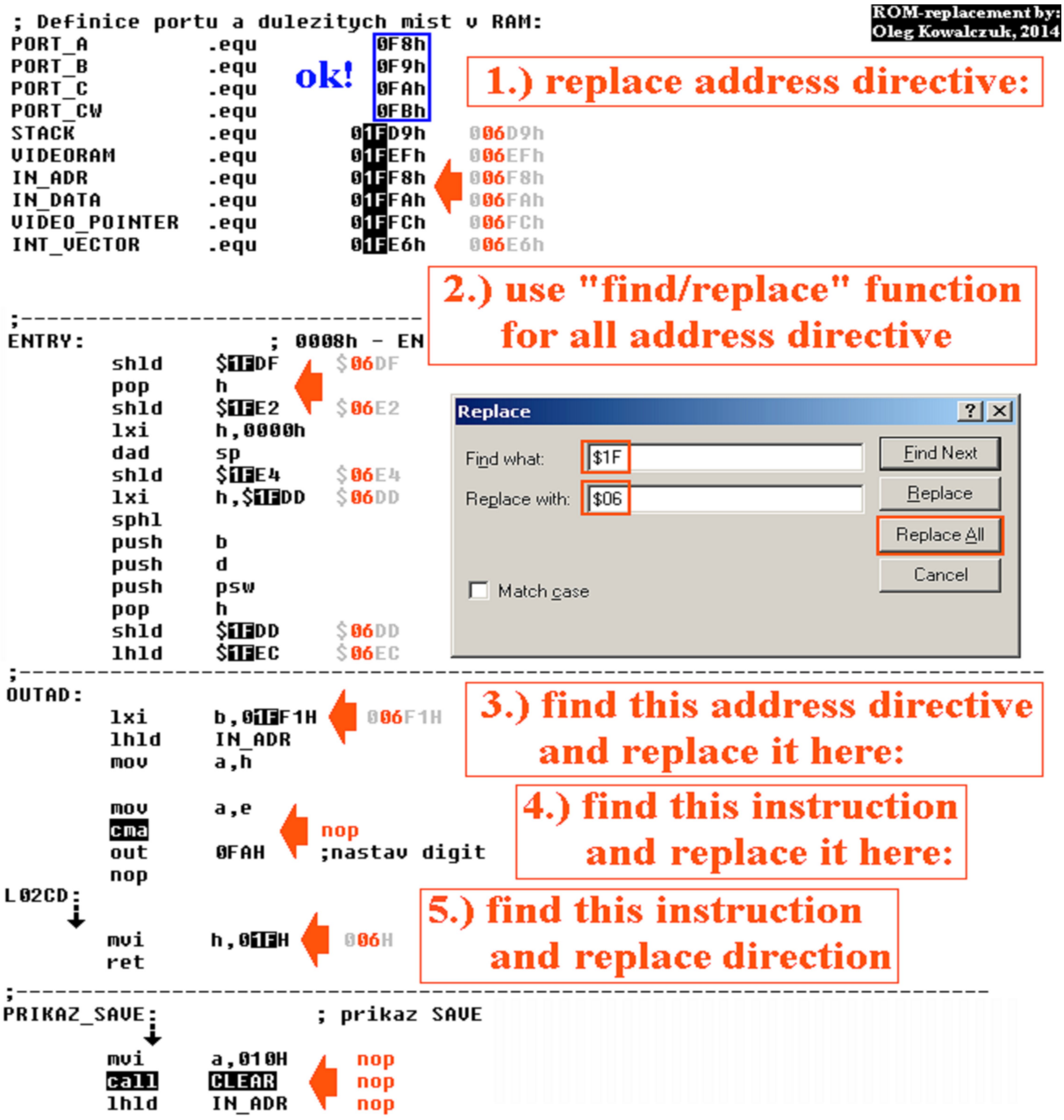
Yasmina derivátom **89C51ED2** emulovala **CPU-8080** s **1,75kB RAM** a obvodom **8255** v základnom móde-0: porty PA, PB, a PC pracujú iba ako paralelné. Obslužná rutina „boot-loader“ po zapnutí, či resete MCU, skopíruje obsah internej EEPROM do RAM, a zabezpečí automatické spustenie užívateľom vytvoreného programu. Plusom je i HW v MCU (serial, čítač, PWM,...), ktorý ako „alternatívne“ porty rozširuje možnosti emulátora. Je možné využiť i prerušenie „INT“ (RST 7), ktoré odovzdá emulovanému 8080 „call“ na 0038h. Emulátor má i malý „debuger“, kde v okne terminálu vidíte internú činnosť emulácie 8080 a portov, s možnosťou krokovania, pomalého behu, či editovania registrov a stránok pamäte. Spojiť to s PC môžete cez serial port, bližšie info - viz. text ďalej... Teraz k **PMI-80**:



obr.2 - schéma zapojenia PMI-80 „emu“ (v0.86ED2)

PMI-80 rom-replacement

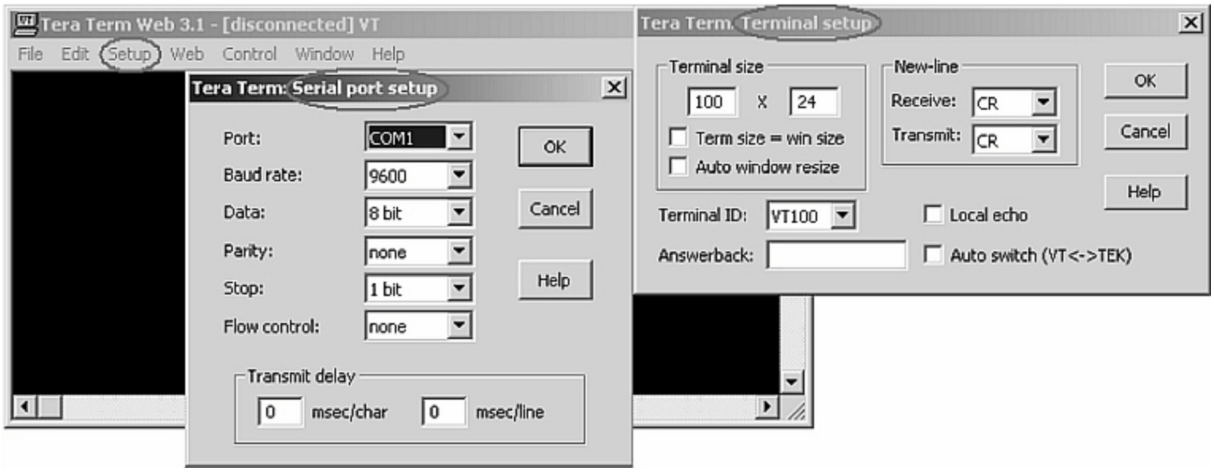
Pre beh monitora **PMI-80** na „**MINIsystem-8080Emulátore**“, boli nutné úpravy originál ROM pamäte, ktorej komentovaný Asm-výpis nájdete na www.nostalcomp.cz. Kowalczuk zmenil direktívy adries „RAM bufferu“, ktorý PMI-80 používa, z pôvodnej 1Fxxh na 06xxh (last-RAM-page MCU). Adresy emulovaného 8255 súhlasili s reálnym 8255 v PMI-80, a zvyšné zmeny sa týkali len inštrukcií „LXI“ a „OUT“. Potom stačilo použiť **asembler Tasm**, a hotovú **PMI-80 ROM** boot-loaderom prijať a uložiť do EEPROM od 0-3FFh. **Pozor**: pri posielaní „bin“ súborov, nezabudnite označiť na termináli status: „binary“**! Schéma**: piny „PA-7“ a „PC-7“ sú nezapojené, bola tam pôvodne logika a tvarovače signálu pre pripojenie magnetofónu, ako záznamového média. Tu je to nepoužité, lebo MCU má reálny UART. Zmena 74LS42N, za dekodér 74LS154, používaný veľakrát ako náhrada za MH1082: originál PMI-80 vysiela data na vstupy ABCD dekodéra MH1082 negovane: 1.znak displeja svieti pri ABCD=1111. Kowalczuk urobil v ROM úpravu, ktorá umožňuje použiť menší obvod 74LS42N. Tu sú úpravy ROM-ky:



„MINI system – 8080 Emulator“

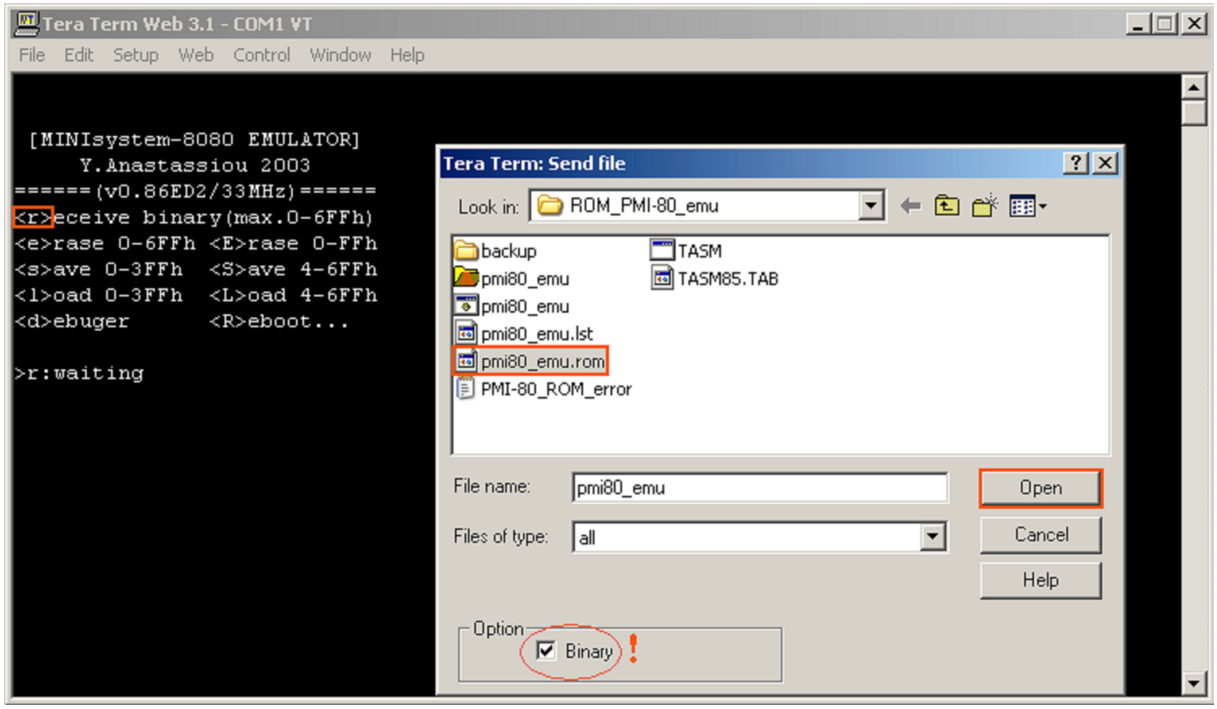
Je to aplikácia, na ktorej **PMI-80** beží. Emuluje **CPU-8080** s ext. **INT** vstupom a **RAM** od adr.0-6FFh (EEPROM 0-6FFh je len záloha pre užívateľský program). **Porty**: **00/01**=adresa/data EEPROM: prístup k stránke-7 (700-7FFh), **02/03/04**=H/L/SW 16bit.counter (in/out=PB0/PB1), **05/06/07**=baud/bufer/SW serial portu, **08/09/0A/0B**=PA/PB/PC/SW PPI-8255 (len mód-0), **0C**=port „X“, **0D/0E**=bufer/SW 8bit.PWM (out=PB4), **0F**=1-sec.báza. (`SW-8255: ---AC-Bc (ABCc=1/0-I/O), SW-counter:00/03=stop,01=počítadlo impulzov,02=generátor, SW-PWM:00=stop,01=run, SW-serial:0101000R, R=indikuje príjem`). Porty sa zrkadlia, zápis do PA-8255 urobíte rovnako na adrese 08, ako na adrese 18h, či B8h. Zápisom bajtov do CTh a CTl čítača, sa uložia data zároveň do jeho predvoľby, čo využíva mód generátor. Pre mód „counter“ (počítanie) sa musí zapísať CTh/l = 0000, aby počítal od nuly. Po pretečení FFFFh, sa opätovne nahrá do neho jeho predvoľba, čo je možné využiť pre počítanie v skrátenom cykle. Kanál PWM má fixnú freq: fosc/2 a dá sa iba zapnúť/vypnúť. Sériový port zápisom do buffera automaticky odvysiela bajt na 9600/N/8/1 a má fixný režim. Baudrate je variabilné zmenou konštanty TT pre čítač, ktorý ho časuje. Príjem bajtu indikuje bit R v jeho PSW porte (R = 1). Port „X“ je obojsmerný, t.j: čo do neho zapíšete, to i prečítate, a má 4 bity: 1111xxxx. Out 1-sec. bázy je pin P3.3, prečítanie jej portu vracia hodnotu 00/01h (pre L a H), zápisom ho - naopak na H či L úroveň môžete nastaviť. Je to využiteľné vtedy, ak používate 1-sec. výstup na generovanie prerušenia INT. Interná EEPROM stránka 07xx (xx=adresa EE), dovoľuje 8080tke zapisovať/čítať konštanty priamo v emulácii. Prerušovací systém je podobný, ako v reálnom 8080: po povolení inštrukciou „EI“, vzniká vynútený call:0038h (RST 7), pričom 8080 ďalšie prerušenie zakáže. Povoliť ho musí ďalšia „EI“, až tesne pred návratom (RET). **Rozšírenie PMI-80 nie je možné**: interná RAM končí adresou 6FFh, a jej prekročením od 700-FFFFh nastúpi externý access. Stratia sa porty PA, PC, a 2 bity portu „X“. No v inom **prípade použitia MINI system-8080Emulátora** je možné pomocou dekodérov pridať externé obvody PPI-8255 (alebo RAM), ku ktorým potom ale už pristupujete ako k RAM, nie ako k portom... **Štart „boot-loadera“** je urobený automaticky, ak je detekovaná interná EEPROM ako čistá, tj.celý jej rozsah (0-6FFh) = iba bajty FFh, alebo 00. Ak sú v nej zaznamenané nejaké iné bajty, boot-loader po resete MCU automaticky načíta do RAM celý jej rozsah 0-6FFh a spustí emuláciu.**Vynútiť spustenie boot-loadera** je potom možné napr. držaním stlačenej klávesy „space“ na termináli, a následný reset MCU. Menu (verzia v0.86ED2):

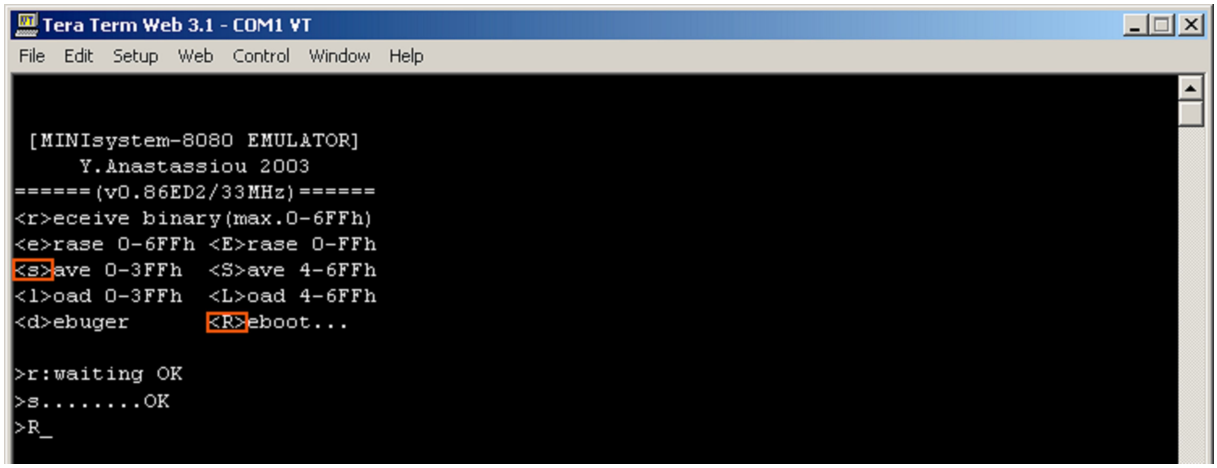
<e>, <E>	erase	- vymaže EEPROM 0-6FFh, alebo 700-7FFh naplnením bajtmi FFh
<r>	receive	- príjem binárneho súboru, ak je kratší ako 0-6FFh, tak samočinný návrat
<l>, <L>	load	- načíta (ak treba) EEPROM do RAM, rozsah: 0-3FFh, alebo 400-7FFh
<s>, <S>	save	- uloží (ak treba) RAM do EEPROM, rozsah: 0-3FFh, alebo 400-7FFh
<d>	debuger	- spustenie **debugeru** emulátora: krokovanie, editovanie, pages,...atd.
<enter>	„run“ mode	- emulácia inštrukcií je prerušovaná zobrazovaním statusu na termináli
<space>	„step“ mode	- emulácia inštrukcií je krokovaná so zobrazovaním statusu na termináli
<r>	register	- editovanie registrov emulovaného 8080 CPU (porty sa nedajú editovať)
<R>	reset registers	- vynulovanie všetkých registrov emulovaného 8080 CPU (nie portov)
<p>	page	- vypíše na terminál celú stránku pamäte RAM v hexa a ascii zobrazení
<a>, <l>	editor debugeru	- jednoduchý editor bajtov na adresách RAM stránok pamäte programu
<q>	quit	- návrat do boot-loadera...

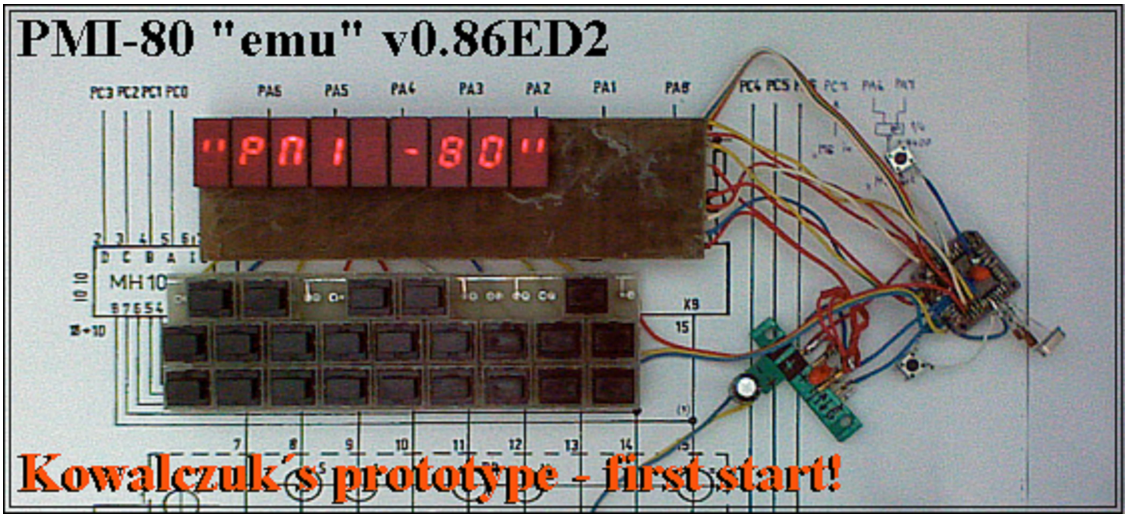


obr.3 - doporučené nastavenie terminálového programu Tera Term (www.ayera.com)

Ukážka činnosti **boot-loadera** pri príjme PMI-80.rom (v0.8.6ED2) a uložení do EEPROM 0-3FFh. Po následnom „reboote“, už po pár sekundách normálne naštartuje monitor **prototypu PMI-80**... (obvod dekodéru 74LS42N s budiacimi tranzistormi a odpormi má Oleg na spodu prototypu displeja).







Ešte **poznámka k debugeru**: je to malý „tool“, pre úpravu bajtov RAM (v hexa) a editovaní registrov 8080 počas emulácie. Môžete s ním urobiť napr. výpis ROM PMI-80 po jednotlivých stránkach, editovať registre, alebo klávesou „space“ krokovať činnosť inštrukcií PMI-80 ROM, ako je zobrazené tu:

```
  [MINIsystem-8080 EMULATOR]
     Y.Anastassiou 2003
======(v0.86ED2/33MHz)======
<r>eceive binary(max.0-6FFh)
<e>rase 0-6FFh <E>rase 0-FFh
<s>ave 0-3FFh  <S>ave 4-6FFh
<l>oad 0-3FFh  <L>oad 4-6FFh
<d>ebuger      <R>eboot...

>l........OK          (load obsahu z EEPROM do RAM, pretože boot-loader ju spustením vymaže)
>d                    (skok do debugeru)


<enter>go!  <space>step  <r>egisters  <R>eset_cpu  <p>ages  <l>ist  <a>dres_ram  <q>uit (menu)
==========================================================================================
[BC] [DE] [HL] [SP] [PC] A  SZ-A-P-C |PA|PB|PC|sw|X  TT|RX|sw  [CT]|sw  PW|sw  EE|dd  1s(port)
0000 0000 0000 0000 0000 00 00-0-0-0  FF FF FF 1B F  F3 64 50  0000 00  00 00  00 00  L (stav)

page:0                (klávesou „p“ a číslom page - sa vypíše jej obsah v hexa a ascII...)
adr  0  1  2  3  4  5  6  7  8  9  A  B  C  D  E  F
000:3E 8A D3 FB 00 C3 2E 00 22 DF 06 E1 22 E2 06 21  >.......".."..!
010:00 00 39 22 E4 06 21 DD 06 F9 C5 D5 F5 E1 22 DD  ..9"..!.......".
020:06 2A EC 06 3A EE 06 77 21 20 02 C3 40 00 21 D9  .*..:..w! ..@.!.
030:06 22 E4 06 C3 3D 00 FF C3 E6 06 FF FF 21 E7 01  ."...=.......!..
040:31 D9 06 22 FC 06 CD 16 01 21 EF 06 22 FC 06 3E  1.."....."..">
050:1D CD AB 00 CD 16 01 21 0B 02 06 06 BE 23 CA 6D  .......!.....#.m
060:00 23 23 05 C2 5C 00 21 02 02 C3 40 00 4E 23 66  .##..\.!...@.N#f
070:69 E9 3E 16 CD AB 00 CD D7 00 7E 32 FA 06 3E 18  i.>.......~2..>.
080:02 CD FB 00 2A F8 06 3A FA 06 77 23 22 F8 06 CD  ....*..:..w#"...
090:BB 00 C3 7A 00 1E 16 20 19 19 12 15 1B 1E 1E 16  ...z... ........
0A0:20 19 05 10 11 13 1E FF FF FF FF 11 08 00 2A FC   .............*.
0B0:06 19 36 19 1D C2 AE 00 2B 77 C9 01 F1 1F 2A F8  ..6.....+w....*.
0C0:06 7C CD C6 00 7D D5 57 0F 0F 0F 0F E6 0F 02 03  .|...}.W........
0D0:7A E6 0F 02 03 D1 C9 CD BB 00 CD 16 01 C8 D2 97  z...............
0E0:01 2A F8 06 E6 0F 29 29 29 29 85 6F 22 F8 06 C3  .*....))))).o"...
0F0:D7 00 01 F6 06 2A FA 06 C3 C5 00 CD F2 00 CD 16  .....*..........

page:1
adr  0  1  2  3  4  5  6  7  8  9  A  B  C  D  E  F
100:01 C8 D2 9D 01 00 00 00 E6 0F 29 29 29 29 85 6F  ..........)))).o
110:22 FA 06 C3 FB 00 CD 40 01 D2 16 01 0F 4F CD 40  "......@.....O.@
120:01 DA 1E 01 CD 40 01 79 FE 90 C9 08 09 0D 0B 0A  .....@.y........
130:13 14 0E 0C 0F 05 1A 0D 0B 0A E4 DF D9 DB DD FF  ................
140:E5 C5 D5 11 00 00 42 7A 32 FE 06 3E 7F D3 F8 00  ......Bz2..>□...
150:7B 2F D3 FA 00 2A FC 06 19 4E 21 BE 01 09 7E D3  {/...*...N!...~.
160:F8 00 3A FE 06 B7 C2 88 01 0E 09 21 9A 01 DB FA  ..:........!....
170:00 E6 70 07 07 D2 82 01 07 D2 81 01 07 DA 88 01  ..p.............
180:09 09 09 19 7E 32 FE 06 1C 3E 0A BB C2 4B 01 3A  ....~2...>...K.:
190:FE 06 07 D1 C1 E1 C9 21 F0 01 C3 40 00 21 F9 01  .......!...@.!..
1A0:C3 40 00 80 84 88 91 8D 8C 89 85 81 82 86 8A 9A  .@..............
1B0:8F 8E 8B 87 83 FF 94 93 FF 97 92 FF FF 90 40 79  ..............@y
1C0:24 30 19 12 02 78 00 18 08 03 46 21 06 0E 07 23  $0...x....F!...#
1D0:2F 0C 47 63 48 71 37 7F 09 2B 0B 2C 5D 3F 42 61  /.GcHq7□.+.,]?Ba
1E0:7B 11 FF FF FF FF FF 1E 13 16 01 19 1F 08 00 1E  {...............
1F0:0E 12 12 18 0A 0D 12 0E 05 0E 12 12 18 19 0D 0A  ................

page:2
adr  0  1  2  3  4  5  6  7  8  9  A  B  C  D  E  F
200:10 0A 1E 19 0E 12 12 11 12 19 1E 92 72 00 91 29  ............r..)
210:02 97 5A 02 9A 7E 02 94 4C 03 93 8C 03 FF FF FF  ..Z..~..L.......
220:1E 0B 12 1F 05 10 11 13 1E 3E 20 CD AB 00 2A E2  .........> ...*.
230:06 22 F8 06 CD D7 00 2A F8 06 22 E2 06 3E 06 D3  ."....*.."..>..
240:F8 00 3E 0F D3 FA 00 21 D9 06 F9 D1 C1 F1 2A E4  ..>....!......*.
250:06 F9 2A E2 06 E5 2A DF 06 C9 3E 0B CD AB 00 2A  ..*...*...>....*
260:EC 06 22 F8 06 CD D7 00 2A F8 06 22 EC 06 7E 32  .."....*.."..~2
270:EE 06 36 CF 2A E2 06 2B 22 E2 06 C3 29 02 3E 12  ..6.*..+"...).>.
280:CD AB 00 CD 16 01 D2 67 00 E6 0F 01 06 00 21 2A  .......g......!*
290:01 0B 09 0C 0D CA 4F 00 BE C2 8E 02 21 2F 01 CD  ......O.....!/..
2A0:CD 02 5D 21 34 01 CD CD 02 63 22 F6 06 C5 CD CA  ..]!4....c".....
2B0:02 E5 4E 23 66 69 22 F8 06 CD D7 00 D1 7D 12 13  ..N#fi".......}..
2C0:7C 12 C1 0D C2 9C 02 C3 4F 00 21 39 01 06 00 09  |.......O.!9....
2D0:6E 26 1F C9 06 09 3E C7 CD EE 02 79 1F 4F 3E 8F  n&....>....y.O>.
..
..
```

```
<enter>go!  <space>step  <r>egisters  <R>eset_cpu  <p>ages  <l>ist  <a>dres_ram  <q>uit
==========================================================================================
[BC] [DE] [HL] [SP] [PC] A  SZ-A-P-C |PA|PB|PC|sw|X  TT|RX|sw  [CT]|sw  PW|sw  EE|dd  1s
0000 0000 0000 0000 0000 00 00-0-0-0  FF FF FF 1B F  F3 33 50  0000 00  00 00  00 00  L

reg:b=12              (modifikujete klávesou „r", a písmenom registra - napr: b, c, h, l...)
reg:c=34
reg:a=A5
reg:

<enter>go!  <space>step  <r>egisters  <R>eset_cpu  <p>ages  <l>ist  <a>dres_ram  <q>uit
==========================================================================================
[BC] [DE] [HL] [SP] [PC] A  SZ-A-P-C |PA|PB|PC|sw|X  TT|RX|sw  [CT]|sw  PW|sw  EE|dd  1s
1234 0000 0000 0000 0000 A5 00-0-0-0  FF FF FF 1B F  F3 0D 50  0000 00  00 00  00 00  L
```

Najviac ma bavilo sledovať beh **ROM PMI-80** v krokovaní (space), či pomalom behu (enter: cca 10-15 inštrukcií/sec). Počas toho preblikávali segmentovky môjho „novo-osadeného" **PMI-80**, a stav emulácie prebiehal v riadku okna terminálu. Môžete vidieť výpis pod sebou, a sledovať zmeny, či robiť úpravy registrov. Spravil som malú ukážku. Myslím, že na „**beta**" verziu **z roku 2003**, urobila (vtedy iba 24-ročná) Y. Anastassiová vcelku solídnu prácu...

```
<enter>go!  <space>step  <r>egister_8080  <R>eset_8080_reg  <p>age  <a>dres_ram  <q>uit
==========================================================================================
[BC] [DE] [HL] [SP] [PC] A  SZ-A-P-C |PA|PB|PC|sw|X  TT|RX|sw  [CT]|sw  PW|sw  EE|dd  1
0000 0000 0000 0000 0000 00 00-0-0-0  FF FF FF 1B F  F3 64 50  0000 00  00 00  00 00  H  mvi a,#8A
0000 0000 0000 0000 0002 8A 00-0-0-0  FF FF FF 1B F  F3 20 50  0000 00  00 00  00 00  L  out $FB
0000 0000 0000 0000 0004 8A 00-0-0-0  FF FF FF 8A F  F3 20 50  0000 00  00 00  00 00  L  nop
0000 0000 0000 0000 0005 8A 00-0-0-0  FF FF FF 8A F  F3 20 50  0000 00  00 00  00 00  H  jmp $002e
0000 0000 0000 0000 002E 8A 00-0-0-0  FF FF FF 8A F  F3 20 50  0000 00  00 00  00 00  H  lxi h,#06d9
0000 0000 06D9 0000 0031 8A 00-0-0-0  FF FF FF 8A F  F3 20 50  0000 00  00 00  00 00  H  shld $06E4
0000 0000 06D9 0000 0034 8A 00-0-0-0  FF FF FF 8A F  F3 20 50  0000 00  00 00  00 00  L  jmp $003D
0000 0000 06D9 0000 003D 8A 00-0-0-0  FF FF FF 8A F  F3 20 50  0000 00  00 00  00 00  L  lxi h,#01E7
0000 0000 01E7 0000 0040 8A 00-0-0-0  FF FF FF 8A F  F3 20 50  0000 00  00 00  00 00  L  lxi sp,#06D9
0000 0000 01E7 06D9 0043 8A 00-0-0-0  FF FF FF 8A F  F3 20 50  0000 00  00 00  00 00  H  shld $06FC
0000 0000 01E7 06D9 0046 8A 00-0-0-0  FF FF FF 8A F  F3 20 50  0000 00  00 00  00 00  H  call $0116
0000 0000 01E7 06D7 0116 8A 00-0-0-0  FF FF FF 8A F  F3 20 50  0000 00  00 00  00 00  L  call $0140
0000 0000 01E7 06D5 0140 8A 00-0-0-0  FF FF FF 8A F  F3 20 50  0000 00  00 00  00 00  H  push h
0000 0000 01E7 06D3 0141 8A 00-0-0-0  FF FF FF 8A F  F3 20 50  0000 00  00 00  00 00  H  push b
..
..
```

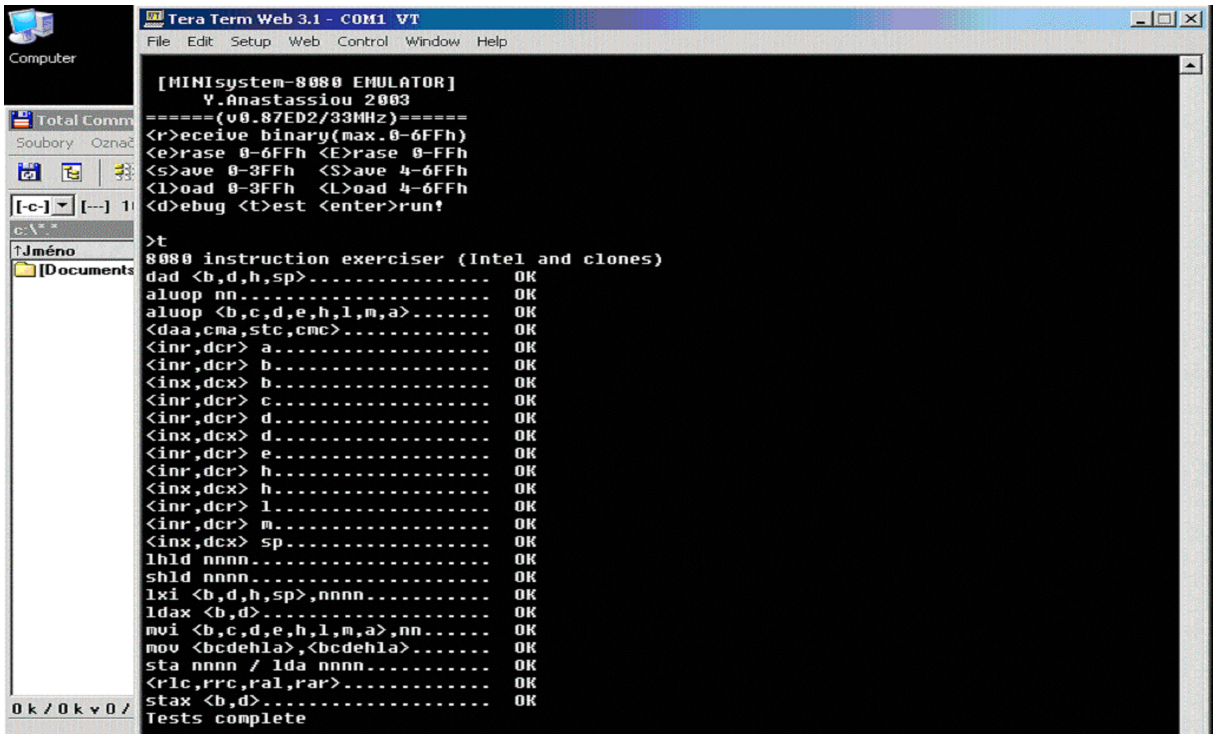
Debuger „**MINIsystem-8080Emulátora**" obsahuje aj jednoduchý disasembler, ako je vidieť z predošlého výpisu: príkaz **list** robí **výpis pamäte** RAM v preklade do inštrukcií, výpis pamäte začína vždy od adresy v PCh/l, a môžete ho prerušiť klávesou terminálu. Editovať môžete i v hexa, ak je potrebné náhodné upravenie nejakej konštanty. Pre tvorbu programu musíte použiť originál assembler (**Tasm**) a skompilovaný kód bez problémov poslať ako „bin" súbor cez sériový port do emulátora...

```
000:  mvi A,#8Ah      (klávesa „l" - list, výpis dissasemblovaných inštrukcií...)
002:  out $FBh
004:  nop
005:  jmp $002Eh
008:  shld $06DFh
00B:  pop H
00C:  shld $06E2h
00F:  lxi H,#0000h
012:  dad SP
013:  shld $06E4h
016:  lxi H,#06DDh
019:  sphl
01A:  push B
..
..

<enter>go!  <space>step  <r>egisters  <R>eset_cpu  <p>ages  <l>ist  <a>dres_ram  <q>uit
==========================================================================================
[BC] [DE] [HL] [SP] [PC] A  SZ-A-P-C |PA|PB|PC|sw|X  TT|RX|sw  [CT]|sw  PW|sw  EE|dd  1s
0000 0000 0000 0000 0000 00 00-0-0-0  FF FF FF 1B F  F3 6C 50  0000 00  00 00  00 00  H

adr:000               (klávesa „a"(editovanie adries RAM) - hexa editor...)
adr:000=3E<-FF
adr:001=8A<-FF
adr:002=D3<-12
adr:003=FB<-34
adr:004=00
```

MINIsystem-8080Emulátor bol vo viacerých verziách, **v0.89ED2** bola pre obvod **AT89C51ED2** v puzdre **PLCC-68**, a emulovala dva obvody PPI-8255 (v základnom móde-0). Ako však už Oleg zistil, existujú aj „**ULTRA**“ **verzie tohoto emulátora**, ktoré v roku 2013 upravili pre použitie na 1 cyklových jadrách x51 „**Children of TROY**“ (autori jednodeskáča „**Hellena-51**“). Sú v dvoch modifikáciách, a využívajú obvody **AT89LP51ED2** a **AT89LP6440**, obidva na frekvencii 24MHz (24-MIPS). Obidve verzie sú ale vyriešené inak, ako verzie určené pre AT89C51ED2: emulácia v AT89LP51ED2 nekopíruje obsah EEPROM do RAM, ale ju mapuje ako klasickú ROM, tj. od adr. 0-FFFh (4kB). Adresy 1000-17FFh už patria RAM-ke. Port „X“ má 6-bitov a pribudol i port „Y“, ktorý pracuje buď ako 4-bit I/O, alebo ako 4-vstupový A/D converter. Rozšírený je i počet PWM výstupov (na 4) a pridaná emulácia „2-wire“ rozhrania. U obvodu AT89LP6440 je k dispozícii až 8kB EEPROM a až 4kB RAM priamo na čipe MCU, tento obvod ale pracuje len s napätím 3,3V... Keďže je obvod **AT89LP51ED2** pinovo ekvivalentný s **AT89C51ED2**, zapožičal som si ho (od O. Kowalczuka) a „ultra“ emuláciu som vyskúšal - rýchlosť je neuveriteľná: pri kryštáli 16MHz sa už na **PMI-80** nedá písať, lebo emulovaný 8080 nedokáže ošetrovať zákmity klávesnice. Pri 24MHz má 8080 výkon asi 5 MIPS. Za zmienku stojí i presnosť emulácie: vyhodnotil ju (po pripojení externej RAM 64kB) test „**8080-Exerciser**“, ktorý je od verzie v0.87ED2 (LP2, LP64) „integrovaný“ a prispôsobený chalanmi **Children of TROY** v MCU, a dá sa spustiť: všetkých jeho 25 krokov, prešlo na emulátore **bez jedinej chyby**...



Musím sa priznať, že ma to najskôr všetko veľmi chytilo, no s odstupom času pri písaní tohoto článku, som pomaly menil názor. Teraz považujem stavbu takéhoto mikropočítača (či už originálu **PMI-80**, alebo tejto miniatúrnej repliky), ako aj stavbu **MINIsystem-8080Emulátora** (i v rýchlej 1-cyklovej modifikácii) za zbytočnosť. Arduino valcuje naprosto všetko a **PMI-80**, či **MINIsystem-8080 Emulátor,** sú iba nevyužiteľné hračky, vhodné i napriek svojím možnostiam iba na mrhanie času ☺. Postavil som to len preto, že mám na to čas a možnosti v práci vo firme. Doma totiž nemám ani pájkovačku...