

# Urob si sám PMD-85

Ing. Peter Chrenko

(Dokončenie)

## Emulácia procesora 8080

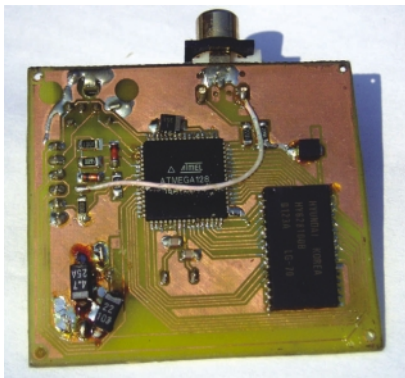
Keďže procesor ATmega128 je vyťažovaný generovaním videesignálu, vygenerovanie každého TV mikroriadku trvá asi 50  $\mu$ s zo 64  $\mu$ s. Zvyšný čas spätného chodu lúča asi 12  $\mu$ s je využitý práve na emuláciu správania procesora Intel 8080 a ostatných obvodov PMD.

## Doska s plošnými spojmi (DPS)

Motív plošných spojov bol navrhnutý tak, aby doska bola jednostranná, ako vidíme na obr. 2. Najpracnejšou úlohou bolo zabezpečiť nekříženie vodičov, čo sa takmer podarilo, až na jeden signál, ktorý je nutné viesť prepojkou medzi PAD1 a COMPOSITE\_VIDEO (CINCH) konektorom. Aby sa ostatné signály na DPS nekřížili, to sa podarilo najmä vhodným priradením signálov I/O portov procesora ATmega128, na schéme je to vidieť ako nelogické poprehadzovanie jednotlivých signálov adresnej a dátovej zbernice. Taktiež niektoré bity I/O portov sú použité iba na rozvod iných signálov alebo napájania, najmä v priestore pod procesorom.

Dosku som vyrobil fotocestou. Pri výrobe je veľmi nutné dodržať bezpečnosť pracoviska vzhľadom na šírku spojov (10 mil) a medzier.

Na vyčistenú vyvolanú, vyleptanú a vyvrátnú DPS odporúčam naniesť vrstvu roztoku kolofónie v liehu, nechať chvíľu odschnúť a nalepiť procesor na zodpovedajúce miesto na DPS. Doporučujem ho zafixovať spájkovačkou aspoň na krajných ôsmich pinoch puzdra TQFP 64. Podobne nalepiť do kolofónie a zafixovať aj pamäť a ostat-



Obr. 3. Pohľad na stranu súčiastok emulátora PMD-85

né súčiastky. Čiže začíname osádzaním súčiastok od najdrahších s najviac vývodmi. Je to tým, že na osadenie 64-pinového „švába“ budete potrebovať priestor na jeho posúvanie v kolofónii pinzetou. Taktiež je vhodné osadiť súčiastky s klasickými vývodmi, prv než kolofónia vytvrdne.

Na montáž súčiastok SMD stačí použiť klasickú transformátorovú spájkovačku, na ktorej hrot upevníme tenký drôtik na naberanie mikrokvapiek spájky. Kolofóniou dosiahneme, aby sa nezlepili piny súčiastok SMD.

Prvú vrstvu kolofónie odporúčam zotrieť liehom, nechať odschnúť a naniesť novú vrstvu roztoku kolofónie.

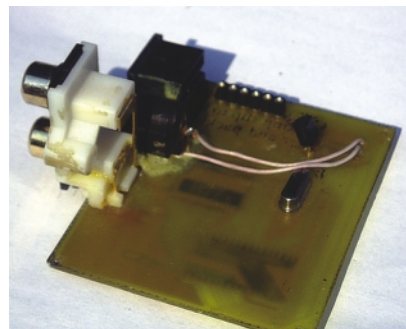
## Oživenie konštrukcie

Po správnom osadení všetkých súčiastok a konektorov vyskúšajte pripojiť prúdovo obmedzený zdroj napätia (napr. adaptér) asi 9 až 15 V jednosmerných na napájací konektor. Na napájacích pinoch procesora skontrolujte prítomnosť napájacieho napätia +5 V. Ak je napájanie procesora v poriadku, pripojte ISP programátor na zbernicu SPI a skontrolujte opäť napätie, či pripojený programátor nemá veľkú spotrebu. Otestujte funkčnosť ISP SPI rozhrania napríklad prečítaním FLASH/EEPROM/fuse bitov z procesora.

Následne je možné prikročiť k nastaveniu poistiek (fuses) procesora a nahratiu obsahu pamäte FLASH a EEPROM do procesora ATmega128.

## Download programu z PC

Na nahratie (upload) emulačného programu do pamäte FLASH a EEPROM a nastavenie poistiek (fu-



Obr. 4. Pohľad na stranu konektorov (CINCH - COMPOSITE\_VIDEO, CINCH - SOUND\_OUT, JACK - napájanie, PS/2 - klávesnica, SIP6 - ISP programátor)

ses) procesora ATmega128 využíva rozhranie SPI vyvedené na konektore JP1 (ISP). Pretože niektoré signály rozhrania sú používané aj rozhraním k PS/2 klávesnici, je bezpodmienečne nutné pred uploadom ju odpojiť, pretože klávesnica by prenos rušila. Počas uploadu môžeme na pripojenom TV sledovať horizontálne čiary, indikujúce prebiehajúci prenos – podobne, ako kedysi na legendárnom ZX Spectre pri nahrávaní programov z magnetofónovej pásky. Na upload súborov cez SPI rozhranie sa osvedčil program PonyProg a avr-dude z balíka utilít WinAVR.

## Nastavenie poistiek (fuses) procesora

Na prepnutie procesora do módu ATmega128 a použitie externého kryštálu 16 MHz je potrebné nastaviť poistky takto:

```
fuse_high_byte = 0x81
fuse_low_byte = 0xFE
fuse_ext_byte = 0xFF
```

To znamená:

```
EESAVE = 0
M103C = 1
BOOTSZ1 = 0
BOOTSZ0 = 0
SPIEN = 0
CKOPT = 0
CKSEL3 = 1
CKSEL2 = 1
CKSEL1 = 1
CKSEL0 = 0
```

Ostatné poistky nastaviť na hodnotu 1 (zakúpený procesor by mal mať tak nastavenú). Nula v tabuľke znamená naprogramovaný bit (prepálená poistka). Upozornenie - ATmega128 je od výroby prednastavená v režime kompatibility s ATmega103, je nutné preto nastaviť správne poistky podľa tejto tabuľky, inak obslužný program nebude fungovať správne.

## Upload obsahu FLASH a EEPROM do procesora ATmega128

Potrebné dátové súbory sú k dispozícii na [www.aradio.cz](http://www.aradio.cz). Po stiahnutí a rozbalení súboru *pmd-emu.zip* sú v adresári *code/* predkompilované obsahy EEPROM pamäte *pmd.eep* a FLASH pamäte *pmd.hex*. S ich uploadom by ste nemali mať žiadne problémy. V prípade úspešného nahratia dát do pamäti (a správnom nastavení poistiek) by ste mali na pripojenom TV prijímači káblom CINCH-SCART prečítať hlásenie:

**\*\* PMD-85 READY /1.0 \*\***

Následne stačí odpojiť ISP programátor a zasunúť PS/2 konektor AT klávesnice a môžete začať pracovať s vaším novým PMD. ISP programátor a klávesnica nesmú byť zapojené súčasne, pretože klávesnica zdieľa port spolu s ISP programátorom.

Ako do procesora napáliť programy, o ktoré máte záujem z pôvodného PMD, sa dozviete ďalej – je však potrebné mať možnosť prekompilovať zdrojové súbory. Z hľadiska elektromagnetickej kompatibility nie je vhodné mať súčasne pripojený TV prijímač a zasunutý ISP programátor v našom PMD + počítač PC, hrozí nebezpečie vytvorenia zemnej slučky pri nedokonalom oddelení zdrojov jednotlivých prístrojov od siete!

## Popis klávesnice

Ako vstupnú perifériu používalo PMD-85-1 neštandardnú klávesnicu QWERTZ, ktorej klávesy sú napávané na klávesnicu PS/2 nasledovne:

**ENTER = EOL** – ekvivalent ENTER na PMD,  
**CTRL = STOP** – kláves zastavenia programu,  
**ALT = MENU** – kombináciou ALT a iného klávesu sa vyvolávajú špeciálne funkcie:  
**ALT+1** alebo ALT+ESC vygeneruje RESET PMD-85/1,  
**ALT+S** – pozastavenie emulácie, výpis krátkeho infa,  
**ALT+C** – prepnutie grafického atribútu jas (negácia jasového bitu BRIGHTNESS).

Treba upozorniť, že pôvodné PMD-85/1 bolo osadené nekvalitnou klávesnicou a zákmity jej kontaktov boli softvérovo potlačené. Preto môžete mať niekedy pocit, že sa klávesnica zasekla, ale je to vlastnosť pôvodného algoritmu SCAN, ktorým sa zákmity ošetrovali. Príliš krátke stlačenie alebo príliš rýchle písanie na klávesnici je náchylné na výskyt stavu zablokovanej klávesnice, ale to je dané nedokonalosťou firmware PMD-85/1.

## Základné povely PMD-85/1

Medzi základné povely, ktoré by som dovolil pripomenúť, patria:  
**MGLD XY** (napríklad MGLD 02) spôsobí nahratie hry (alebo programu) z magnetofónovej pásky, ktorá je v našej konštrukcii realizovaná ako časť FLASH pamäte ATmega128. XY je dekadické dvojčiferné číslo, pod ktorým je daný program uložený na páske. Niektoré programy v strojovom kóde majú samospúšťanie (realizované nahraťím časti zásobníka z magnetofónovej pásky), niektoré treba spustiť povelom:  
**JUMP ADDR**, kde ADDR je 4-ciferná hexadecimálna adresa spustenia. Drvivá väčšina hier pre PMD používa na štartovanie tieto adresy: 0000, 0100, 1000, 1500 alebo 2000. Ak spúšťaciu adresu hry neuhádnete, zrejme spustíte náhodné dáta, ktoré Vám premažú RAM a preto je nutné hru alebo program nahráť povelom MGLD ešte raz. Veľmi užitočná je resetovacia kombi-

nácia kláves ALT+ESC (resp. ALT+1).  
**DUMP ADDR** – výpis obsahu pamäte od adresy ADDR.  
**BASIC G** – spustenie interpretera BASIC. Na nahrávanie programov pod BASICom treba použiť príkaz:  
**LOAD Y** (napríklad LOAD 10) na nahratie programu z magnetofónovej pásky (Y je dekadické číslo programu, pod ktorým je uložený na páske) a spustiť ho je možné príkazom **RUN**. Samozrejme, že PMD pozná mnoho viac príkazov, v BASIC G vám môžem odporúčiť nahráť si nejaký BASIC hru (napr. WURMI, BOMBAR- DER) a príkazom **LIST** si pozrieť zdrojový program. Určite sa na nejaké finesy o niekdajšom programovaní rozpamätáte.

## Hry

Hry a programy sú v strojovom kóde a v BASIC PMD-85/1. Keďže je limitovaný obsah internej pamäte FLASH (na hry a programy PMD zostáva asi 113 kB), nepodarilo sa ani po kompresii uložiť viacej PMD-85/1 programov. Ak chcete zaspomínať na iné hry, odporúčam pozrieť, či sa nenachádza v adresári rom1/ alebo rom2/. Popis hier nájdete v dokumentácii *index.html*, vid' ďalší popis.

## Softvérové vybavenie ATmega128

Emulačný softvér pre ATmega128 je napísaný v assembleri v kombinácii s preprocesorom - skriptovacím jazykom PHP. Na kompiláciu odporúčam mať nainštalované balíky WINAVR, AVRASSEMBLER a skriptovací jazyk PHP. Prípadné rozdiely v inštalácii nastavte v súbore code/Makefile. Jednotlivé zdrojové súbory sú uložené po adresároch:

### adresár code/

**8080.asm** – rozskoková tabuľka jednotlivých 256 kódov inštrukcií procesora 8080.  
**basic1.rom** – binárny obsah ROM modulu s BASIC-G.  
**monit1.rom** – binárny obsah ROM PMD 85/1 - firmware (MONITOR).  
**basic1.asm** – obsah ROM modulu BASIC-G vo formáte ASM, vygenerovaný skriptom games.php.  
**fuses\_18.txt** – textový popis nastavení konfiguračných fuses.  
**Makefile** – dávkový súbor pre skompilovanie zdrojových kódov. Nastavuje sa v ňom adresár \$(MGFDIR), z ktorého sa prikompilujú PMD-85/1 programy do vstavanej magnetofónovej pásky. Príkazom make all skompilujete všetky zdrojové kódy a nahráte obsah pamäti FLASH a EEPROM do mikroprocesora. Príkazom make pmd.hex preložíte zdrojové kódy, príkazom make eeprom preložíte zdrojové kódy a na-

hráte obsah pamäte EEPROM a napokon make flash preložíte zdrojové kódy a nahráte ich do pamäte FLASH.

**macro.asm** – definície priradenia jednotlivých I/O pinov procesora, jednoduché makrá.

**kbd.php** – prerošovací obslužná rutina prijímu znaku z USART0 od klávesnice PC-AT. Po prijímaču sa v internej bitovej mape vo formáte 16x5 bitov zaznačí informácia o stlačení alebo pustení jednotlivého klávesu. Taktiež je realizovaná OSD obrazovka po stlačení ALT+S, reset PMDčka po stlačení ALT+1 (ALT+ESC).

**kb\_lookup.asm** – mapovacia tabuľka SCAN kódov klávesnice na ich pozíciu v matici kláves PMD-85.

**monit1.asm** – obsah ROM pamäte PMD-85/1 vo formáte ASM vygenerovaný skriptom games.php. Je uložený do pamäte EEPROM.

**video.php** – generovanie TV obrazu, serializácia TV riadku, synchronizačných impulzov, zvuku.

**pmd.php** – hlavný zdrojový súbor, inicializácia, emulácia jednotlivých inštrukcií procesora 8080.

**pmd.hex** – obsah pamäte FLASH ATmega128, získaný kompiláciou.

**pmd.eep** – obsah pamäte EEPROM ATmega128, získaný kompiláciou.

**pmd.asm** – zdrojový assemblerovský súbor získaný po prejení pmd.php vykonaním PHP programu.

**pmd.lst** – výpis - listing po preložení pmd.asm.

**pmd.map** – výpis priradenia assemblerovských symbolov, získaný kompiláciou pmd.asm.

**games.php** – skript na konverziu binárnych ROM modulov do assemblerovskej formy. Taktiež je určený na zbalenie obsahu romXY adresárov, ich RLE kompresiu do jedného súboru.

**games\_rom.asm** – obsah magnetofónovej pásky PMD-85/1 vo formáte ASM vygenerovaný skriptom games.php. Dáta su komprimované RLE metódou - opakujúce bajty sú nahradené značkou, hodnotou a počtom opakovaní.

### adresár docs/

**index.html** – dokumentácia.

### adresár schema/

**schema128.sch** – schéma zapojenia emulátora PMD-85/1 vo formáte EAGLE 4.10.

**schema128.brd** – obrazec dosky s plošnými spojmi vo formáte EAGLE 4.10.

**plosak\_top\_mirrored.eps** – zrkadlový obrazec dosky s plošnými spojmi vo formáte EPS (postscript), vhodný pre tlač na fóliu pri výrobe DPS fotocestou. Opäť upozorňujem na nutnosť jednej prepajky vo vedení signálu COMPOSITE\_VIDEO.

#### adresár rom0/

Obsah magnetofónovej pásky – hry a programy v strojovom kóde (AUTO, TEHLÝ, FRED, HORACE, MANIC, PAMPUCH a ZABY) a v BASIC PMD-85/1 (BLACK, BOMBARDER, FARAON, KON, TAVA, TEST LOG, WURMI).

#### adresár rom1/

Obsah magnetofónovej pásky – ďalšie hry a programy v strojovom kóde PMD-85/1: BLUDISTE, CERES1, FLAPPY, KANKAN, KARE3D2, MANIC23, PUZZLE.

#### adresár rom2/

Obsah magnetofónovej pásky – ďalšie hry a programy v strojovom kóde PMD-85/1: HLIPA, HORAC, PEXESO, SABOTER, WILLY.

Popísaný potrebný softvér je k dispozícii na <http://www.aradio.cz>.

### Emulátor procesora 8080

Vlastná emulácia procesora 8080 je pekným príkladom extrémneho programovania za účelom vyžmýkania maximálnej priepustnosti programu. Vlastne je to dôsledok architektúry RISC, keď sú inštrukcie procesora optimalizované pre programátora a súčasne na rýchlость vykonávania. Spolu s 32 takmer voľne použiteľnými registrami je radosť programovať. Vôbec sa to nedá porovnávať s programovaním INTEL 8080, kde bol jeden univerzálny register a programovanie bolo ako valčík: 1. vlož do akumulátora jeden operand, 2. vykonaj operáciu, 3. ulož výsledok z akumulátora.

Emulátor PMD je komplexný program napísaný v assembleri, z ktorého asi najzaujímavejšie časti sú:

Dekódovanie inštrukcií - použitie IJMP a RJMP inštrukcií (v rozsahu 8 kB kódu), je to omnoho rýchlejšie ako klasická rozskoková tabuľka.

Často používané číselné konštanty – 0 a 255 sú uložené v registroch. Taktiež všetky často používané premenné a stav procesora 8080 sú uložené v registroch.

Rýchla ochrana oblastí ROM pred zápisom (emulácia chovania ROM v RAM) - umožnená vložením iba dvoch inštrukcií SBRC do sekvencie zápisu do RAM.

Načítanie a zobrazenie jednej šesťice pixelov - spracovanie jasového atribútu vďaka inštrukcii ASR. Spracovanie blikacieho atribútu je realizované cez prevodnú look-up tabuľku v internej pamäti RAM (Id video\_tmp,X). Taktiež rýchlostné pomery (6 Mpx/s) sú priaznivejšie pri našom usporiadaní externej pamäte RAM s možnosťou prístupu po fázach (vyšle sa horný bajt adresy, potom dolný bajt adresy, potom sa prečíta výsledok čítania, inkrementácia adresy a popri všetkom tom serializácia bajtu video-pamäte vrátane handlovania grafických atribútov).

Exaktné časovanie - aby sa obraz na obrazovke nevlnil v závislosti od dĺžky trvania inštrukcie, po ktorej nasledovalo prerušenie od časovača a začatie vykresľovania TV riadku, je zabudovaná automatická synchronizácia na 1, 2 a 3 cyklové inštrukcie, ktorým sa pri programovaní veľmi nedá

vyhnúť. Vtipné je aj generovanie TV synchronizačných signálov priamo cez PWM výstup.

### Záver

Záverom možno už iba dodať, že konštrukcia dokazuje pokrok v technológiach v priebehu približne 20-tich rokov (1985 až 2005) v kategórii tučtových mikroprocesorov - Intel 8080 a ATMEL MEGA128 a celkovú technickú úroveň dnešnej doby. Tiež konštrukcia niekdajšieho PMD-85/1 z produkcie TESLY Piešťany je veľmi vďačná v tom, že sa vlastne zobraza pôvodný softvér, ktorý tvorilo a ladilo množstvo šikovných ľudí. Efekt je takmer dokonalý – klasické hry (FLAPPY, BOULDER DASH, MANIC MINER, TETRIS...) v kombinácii jednoduchej konštrukcie s novým procesorom... Bez výborného softvéru by vlastne vznikol len ďalší generátor testovacích televíznych obrazcov. Len moje laborovanie, ako vtisnúť PMDčisko do iného procesora, je niekoľkoročné úsilie a na vývoj každej zložitejšej hry obetoval jej autor čas rátaný tiež na roky, suma sumárom – keby začnete vyvíjať takéto zariadenie (malú hernú konzolku) od nuly, bude Vám to trvať zrejme niekoľko desaťročí.

Na koniec by som chcel poďakovať Ing. Jurajovi Bulisčákovi, OM8ACE, za realizáciu 2 ks emulátora PMD-85/1 podľa tejto konštrukcie a za preštudovanie návodu a pripomienky pri nastavovaní fuses (poistiek)!

Forever 8 bits, forever PMD-85 ...