

ZX Spectrum a tlačiarne.

=====

Ing. Peter Koždoň

Pre každého majiteľa mikropočítača sa po určitom čase stáva nepostrádateľná tlačiarne, či už iba pre potrebu tlače výsledkov výpočtov, alebo ako nepostrádateľný pomocník pri ladení nových programov.

K mikropočítaču Sinclair je možné pripojiť ľubovoľnú tlačiarne, keďže však Sinclair nemá výstup pre žiadny typ tlačiarne musíme si pre pripojenie urobiť potrebné technické a programové doplnky. Každý majiteľ tohto mikropočítača si iste všimol, že na zadnej strane sú vyvedené signály zbernice, čo nám dáva pri ich znalosti možnosť ďalšieho rozširovania technických prostriedkov nášho počítača. Aj pre pripojenie tlačiarne využijeme niektoré z nich. O konkrétnych technických prostriedkoch prepojenia sa zmienim neskôr.

Aby sme si mohli urobiť potrebné doplnky (technické aj programové) potrebujeme dôkladne poznať spôsoby, akými prebieha komunikácia medzi tlačiarňou a počítačom. V tomto príspevku si postupne popíšeme spôsoby komunikácie, potrebné technické prostriedky a spôsob tvorby programového vybavenia pre pripojenie tlačiarne s paralelným rozhraním, ktorých je u nás čo do počtu prevaha.

Paralelných rozhraní, t.j. rozhraní ktoré umožňujú súčas-
tný prenos 1 Bytu = 8 bitov, je niekoľko typov:

1. Centronics,
2. IRPR,
3. LOSABAX/DZM 180,
4. FMD (pri tlačiarňe PRT 80 GS s prídavným CDI interfejsom).

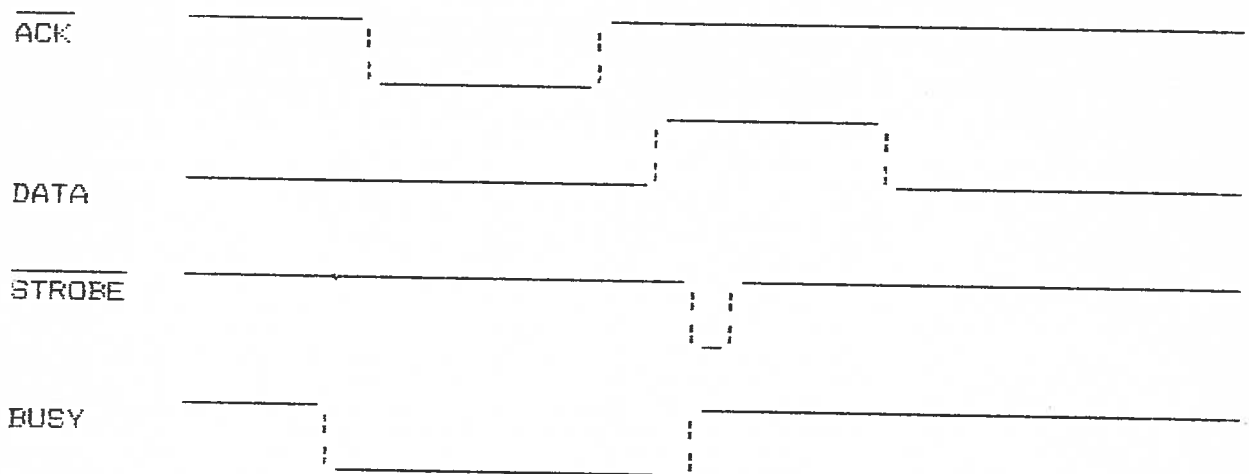
Aby sme si mohli vysvetliť funkciu jednotlivých typov rozhraní musíme si nakresliť časové priebehy rozhodujúcich signálov, ktoré riadia prenos. V tomto príspevku sa budeme zaoberať výlučne prenosom dát do tlačiarne bez využitia prerušenia, pričom celý prenos bude riadený prídavným programovým vybavením.

ROZHRANIE CENTRONICS

=====

Jedná sa o najčastejšie používané rozhranie u zahranič-
ných tlačiarň. Signály sú najčastejšie vyvedené na konektor
typu Amphenol 57-30360, alebo 871037 (ELTRA). Na tlačiarňe PRT
sú signály vyvedené na konektor 881025 (UNITRA). Pri popise
jednotlivých signálov budem uvádzať číslovanie kontaktov zhodne
s konektorom Amphenol, ktorý zodpovedá štandardu.

Diagram časových priebehov signálov:



Minimálna šírka signálu STROBE musí byť 0.5 mikrosekúnd.

Popis signálov:

=====

čís.	signál	smer	význam
1.	<u>STROBE</u>	VST	zmena tohto signálu z "1" do "0" zapisuje datovú informáciu do tlačiarne (strobuje vstupné data).
2.	DATA1	VST	informačný bit
3.	DATA2	VST	informačný bit
4.	DATA3	VST	informačný bit
5.	DATA4	VST	informačný bit
6.	DATA5	VST	informačný bit
7.	DATA6	VST	informačný bit
8.	DATA7	VST	informačný bit
9.	DATA8	VST	informačný bit
10.	<u>ACK</u> (ACKNLG)	VYS	potvrdenie prijatia vstupnej informácie.
11.	BUSY	VYS	logická hodnota "1" tohto signálu znamená nepripravenosť tlačiarne prevziať znak.
12.	PAPER OUT (PE)	VYS	logická hodnota "1" tohto signálu znamená, že v tlačiarne nie je papier.

13.	SELECTED (SLCT)	VYS	log.hod. "1" znamená pripravenosť tlačiarne (ON-LINE).
14.-15.	Nevyužité		
16.	SIGNAL GND		Signálová zem.
17.	CHASSIS GND		Prístrojová zem.
18.	+5VDC		Externý zdroj +5V.
19.-30.	GND		Logických OV.
31.	<u>RESET</u>	VST	log.hod. "0" tohto signálu nastavuje tlačiareň do základného stavu.
32.	<u>ERRCLR</u>	VYS	logická hodnota "0" tohto signálu znamená poruchu tlačiarne.
33.	EXT GND		Externá zem.
34.-36.	Nevyužité		

Pre realizáciu prenosu dát z počítača do tlačiarne potrebujeme z uvedených signálov minimálne 10 a zem, teda spojenie môžeme realizovať pomocou 11-tich vodičov. Komunikácia s použitím tohto rozhrania prebieha nasledovne:

1. Testujeme pripravenosť tlačiarne prijať znak (BUSY).
2. Ak je tlačiareň pripravená (BUSY="0"), zapíšeme na datovú zbernicu znak.
3. Platnosť znaku potvrdíme signálom STROBE.

Je možné taktiež využívať signály indikácie konca papiera, prípadne signál nastavenia tlačiarne do východzieho stavu. Test signálu konca papiera však nie je vždy vyhovujúci, nakoľko v prípade, že píšeme na sólo papier formátu A4, ku koncu stránky tlačiareň hlási koniec papiera, hoci ešte požadujeme písanie. Teda v prípade, že signál PAPER OUT použijeme, musíme mať v obslužnom programe tlače možnosť voľby zapnutia a vypnutia testu.

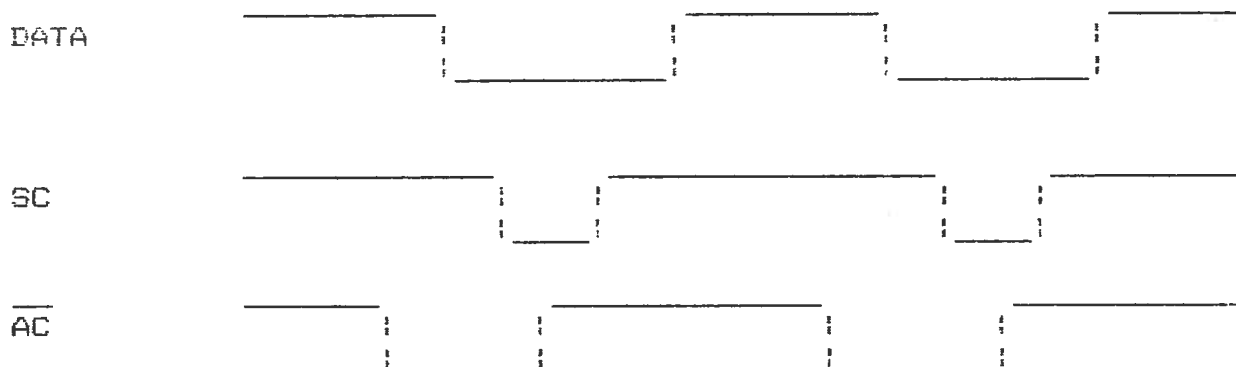
Signál RESET je obvykle možné vynechať a nahradiť zapísaním vhodnej postupnosti riadiacich znakov do tlačiarne, ktoré vykonajú tú istú činnosť - nastaví tlačiareň do východzieho stavu.

Uvedené čísla vývodov platia pre tlačiarne D-100, STAR, EPSON, IBM, VIDEOTON 21200 a pod. Pre tlačiareň PRT-80 GS uvedené čísla vývodov neplatia. Na tejto tlačiarňi sú signály vyvedené na 25 pinovom (kontaktnom) konektore a ich rozmiestnenie najdeme v manuále "Parallel GS-CDI Interface", ktorý je dodávaný spolu s touto tlačiarňou.

ROZHRANIE IRPR

=====

Pri tomto rozhraní je časový priebeh signálov nasledovný:



Uvedený časový diagram platí pre tlačiareň D-100. Pre tlačiareň PRT je priebeh rovnaký, iba s negovanými signálmi. Pre komunikáciu prostredníctvom rozhrania IRPR nám podľa časového priebehu postačujú signály AC, SC a samozrejme dátové signály, po ktorých posielame jednotlivé znaky pre tlač.

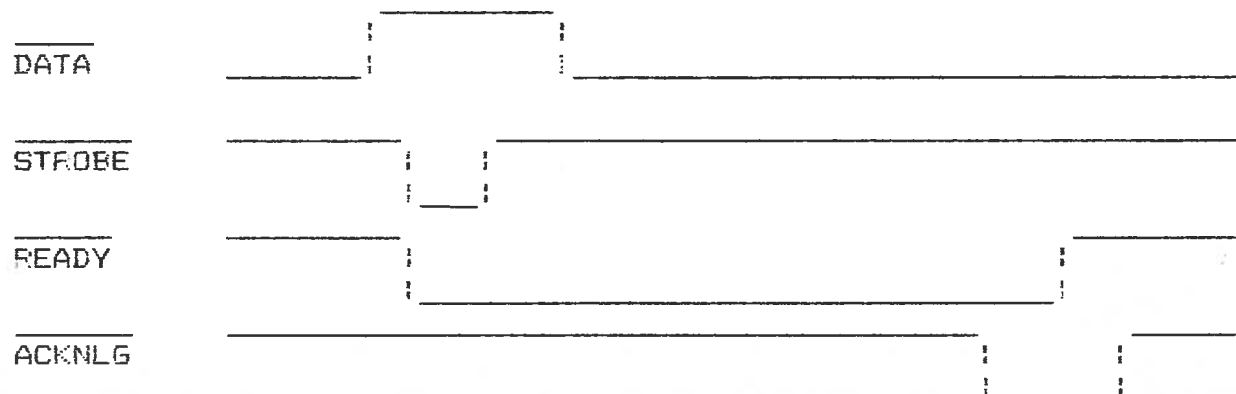
Signál SC strobuje (potvrďzuje) platnú informáciu na dátovej zbernici zmenou zo stavu "0" do stavu "1" pri stave signálu AC="1". Pri zmene signálu SC do stavu "1" sa nesmie dátová informácia (D0 - D7) meniť. Signál SC môže prejsť späť do stavu "0", až po zmene signálu AC do "0". Signál AC predstavuje potvrdenie prijatia informácie.

Pri komunikácii s rozhraním IRPR môžeme samozrejme využiť aj ďalšie signály, ako napr. koniec papiera, pripravenosť tlačiarne a pod. Toto však nie je nutné.

ROZHRANIE LOGABAX/DZM

=====

Pri tomto rozhraní je časový priebeh signálov nasledovný:



Pri komunikácii s týmto rozhraním bude táto prebiehať podobne ako s rozhraním Centronics. Signál READY má ten istý význam ako signál BUSY. Pri tomto type rozhrania je aktivnosť všetkých signálov v logickej hodnote "0".

Pri komunikácii môžeme obdobne ako v predchádzajúcich využiť signály konca papiera, pripravenosti tlačiarne (ON-LINE) a pod.

ROZHRANIE PMD

=====

Tento typ rozhrania je implementovaný pri tlačiarňi PRT-80 GS s paralelným interfejsom CDI. Komunikácia prebieha podobne ako pri rozhraní IRPR.

- 00 -

Popis signálov pre konkrétne rozhrania najčastejšie sa u nás vyskytujúcich tlačiarňi sú v prílohe aj s príslušným označením vývodov a typov konektorov.

Fripojenie tlačiarne k mikropočítaču.

=====

Fripojenie tlačiarne ku Spectru je možné realizovať rôznymi spôsobmi. V amatérskych podmienkach je však vhodné voliť typ pripojenia s minimálnou podporou Hardware (technických prostriedkov) a s maximálnou podporou Software (programových prostriedkov). Táto forma je maximálne zhodná aj s celou koncepciou mikropočítača Sinclair ZX Spectrum, ktorá je založená na minimálnom vybavení technickými prostriedkami.

Na základe uvedeného budeme pripojenie realizovať prostredníctvom programovateľného obvodu PPI (Programmable Peripheral Interface) 8255A, ktorého ekvivalent sa vyrába aj u nás pod označením MHB 8255A. Jedná sa o plne zlučiteľný obvod s logikou TTL, v púzdre DIL so 40-timi vývodmi a napájaním +5V.

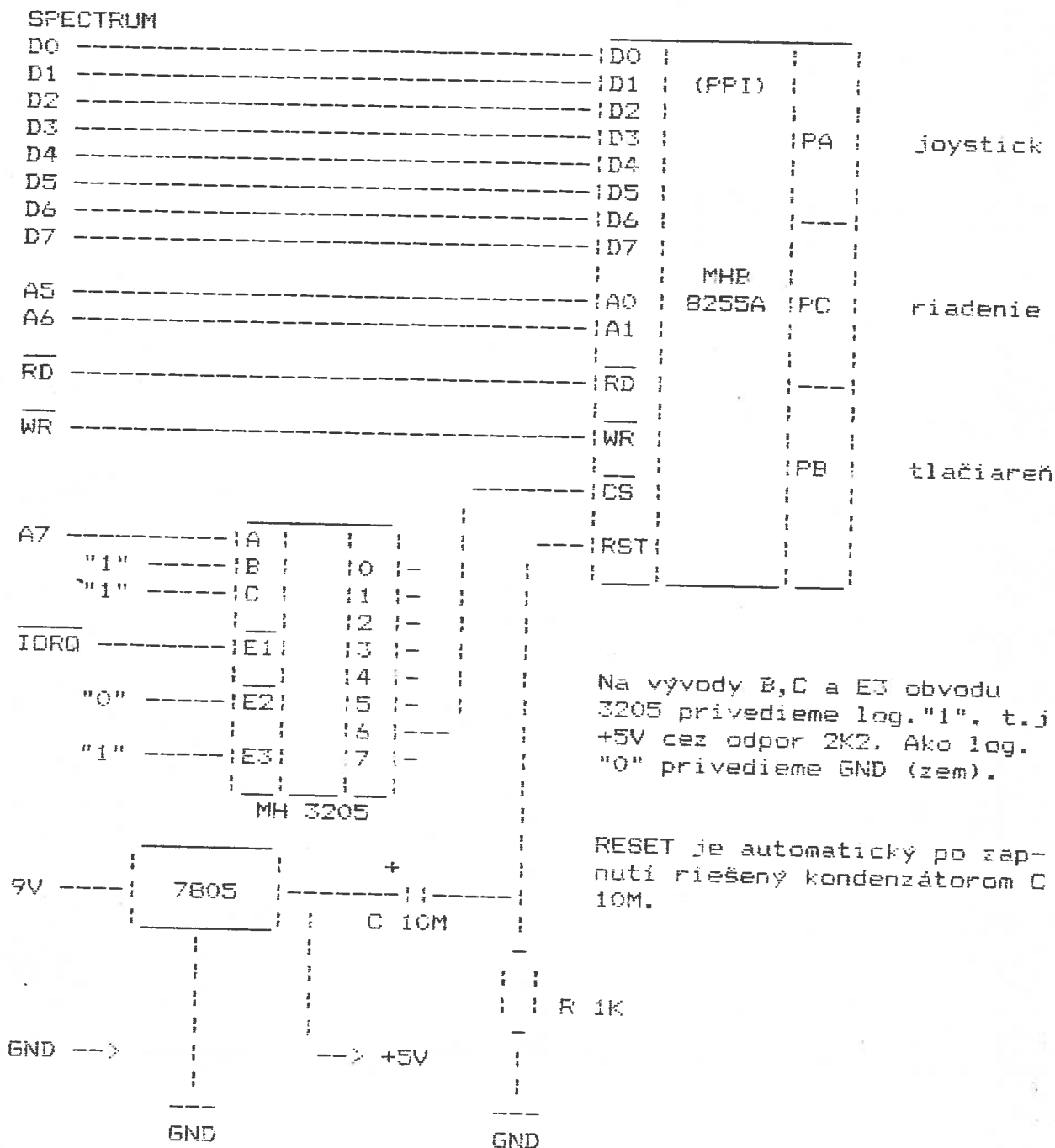
Bližším popisom tohto obvodu sa nebudeme zaoberať, nakoľko je ho možné nájsť v rôznej literatúre a v katalógových listoch.

Zapojenie tohto obvodu bolo uvedené v rôznych verziach aj v časopise Amatérske rádio. Nakoľko má obvod 8255A tri osem bitové vstupno-výstupné brány, je vhodné jeho súčasné využitie na viacej činností (nie iba pre komunikáciu s tlačiarňou).

Vhodná varianta využitia je napr.:

brána A joystick (podľa spravodaja č.III),
 brána B data pre tlačiareň,
 brána C riadenie komunikácie s tlačiarňou
 (BUSY, STROBE, PAPER OUT, SELECTED ...).

Schéma zapojenia môže byť nasledovná:



Zo Spectra nie je vhodné brať priamo napätie +5V, nakoľko sa tým zaťažuje stabilizátor 7805 v počítači a tento sa viac hreje. Vyhodnejšie je brať zo zbernice +9V a stabilizovať ich. Namiesto stabilizátora 7805 môžeme použiť obvod MAA 723. Obe verzie sú v praxi vyskúšané a fungujú spoľahlivo.

ZX Spectrum a tlačiarne.

Ing. Peter Koždoň

(pokračovanie)

Aby sme mohli pomocou interfejsu s 8255-kou tlačiť, musíme si pre to urobiť vhodný obslužný program vo forme tlačového podprogramu, ktorý umožní pre tlač používať priamo príkazy LPRINT a LLIST. Kopírovaním obsahu obrazovky (COPY) sa zatiaľ zaoberať nebudeme, nakoľko programové vybavenie pre túto činnosť je závislé priamo od typu použitej tlačiarne.

Tlačový podprogram musí zabezpečiť:

=====

1. Inicializáciu (naprogramovanie) obvodu PPI - 8255A. V tejto časti podprogramu môže byť zahrnutá aj inicializácia tlačiarne (jej uvedenie do východzieho stavu, naprogramovanie užívateľskej grafiky), prípadne test na prítomnosť a pripravenosť k tlači (stav tlačiarne ON-LINE).
2. Test kódu vysielaného znaku, pričom na základe jeho hodnoty nastane vetvenie podprogramu na:
 - prechod na novú stránku,
 - prechod na nový riadok,
 - tlač znaku,
 - tlač kľúčového slova Basicu (TOKENS).
3. Počítanie riadkov na stranu. Po naplnení strany zadaným počtom riadkov musí program zabezpečiť automatický prechod na novú stránku a vynulovanie počítadla riadkov. Počet riadkov na stranu musí byť nastaviteľný.
4. Počítanie vytlačených znakov na riadok. Po naplnení riadku (vytlačení) zadaným počtom znakov zabezpečiť automatický prechod na nový riadok a vynulovať počítadlo znakov na riadok. Počet znakov na riadok musí byť taktiež nastaviteľný.
5. Vyslanie kódu znaku do tlačiarne a obsluženie prenosu.
6. Testovanie tlačítka BREAK. Pri jeho stlačení sa musí tlač prerušiť. Tento test môžeme vynechať, avšak pri náhodnej poruche tlačiarne sa tlačový podprogram dostane do čakacej slučky a nebude ho možné prerušiť.

Pri obslužnom podprograme tlače pre Spectrum existuje jedna zvláštnosť. Ak chceme tlačiť kľúčové slová Basicu, používame pre túto činnosť podprogram z ROM, ktorý zabezpečí postupné vyslanie znakov kľúčového slova, teda "rozbiť" kľúčové slovo na jednotlivé znaky. Toto značne skrátí vlastný tlačový podprogram, ktorý by v opačnom prípade musel túto činnosť

zabezpečiť. Nakoľko kódy kľúčových slov sú od hodnoty 00A5H (165 dekadicky), nemáme možnosť pri použití tlačového podprogramu, ktorý "ovláda" kľúčové slová, využiť všetky vlastnosti tlačiarnie. Jedná sa hlavne o tlač semigrafických znakov, programovanie užívateľských znakov v tlačiarni, grafický režim a pod.

Aby sme neboli ochudobnení o tieto možnosti, musíme si urobiť iný druh obslužného podprogramu pre tlač, ktorý však nebude poznať kľúčové slová. Pre príkaz LPRINT to však nevadí, bude fungovať správne. V prípade, že pri takomto podprograme použijeme príkaz pre tlač listingu (výpisu) programu, tento nebude správne fungovať, nakoľko namiesto kľúčových slov bude vypisovať rôzne (napr. semigrafické) znaky implementované v danom type tlačiarnie na kódy od 165 a vyššie.

Ako vyplýva z uvedeného, môžeme si urobiť obslužný podprogram presne podľa požiadaviek. Napr. môžeme si priamo z užívateľského programu zapínať a vypínať automatické stránkovanie (pomocou POKE), používať zvolený znak pre príkaz na zastránkovanie, písať hlavičky a číslovať listy výpisov programov a pod.

Používanie zvoleného znaku pre zastránkovanie (prechod na začiatok novej strany) je vhodné pre textový editor TASWORD a rôzne jeho modifikácie. Takto si môžeme riadiť stránkovanie zapísaním zvoleného znaku priamo do textu. Podobne môžeme využívať aj ostatné vlastnosti tlačiarní (zväčšené znaky, kondenzované znaky, tlač semigrafických znakov, súčasné podčiarkovanie textu a pod.).

Konkrétny typ obslužného tlačového podprogramu závisí na type použitej tlačiarnie a mal by plne využívať jej možnosti.

Je samozrejmé, že celý podprogram musí byť urobený v strojovom kóde aby bol krátky a rýchly. Pre programovanie použijeme Jazyk symbolických adries procesora Z-80 (assembler). Vytvorený program uložíme na koniec pamäte RAM (nie je to nutné) a budeme ho chrániť posunutím RAMTOP-u na adresu a 1 nižšiu ako je začiatok podprogramu. RAMTOP presunieme príkazom CLEAR adresa. Takto máme podprogram pre tlač chránený proti premazaniu Basicovským programom, prípadne oblasťou pamäte pre uloženie premenných.

Ovládací podprogram.

=====

ZX Spectrum má v pamäti ROM podprogram, ktorý vysielá znak z registra A (akumulátora) buď na obrazovku, alebo na tlačiareň. Obslužný podprogram v ROM je však schopný spolupracovať iba s tlačiarnou, ktorá má vhodne prispôbené špeciálne rozhranie. Takáto tlačiareň je napr. SEIKOSHA GP 50 S, kde 'S' za typom znamená, že je určená pre Sinclair.

Chceme, aby náš ovládací podprogram pre tlač fungoval pri použití príkazov LPRINT a LLIST, čo znamená, že pre výstup na tlačiareň musíme nahradiť podprogram z ROM pre vyslanie znaku na tlačiareň novým podprogramom, ktorý umiestnime niekde v pamäti ROM nášho mikropočítača. Spôsobom vhodného umiestnenia sa budeme zaoberať neskôr. Ovládací podprogram pre vyslanie znaku v ROM-ke začína na adrese 09F4H. Ak si prezrieme oblasť systémových premenných nášho mikropočítača zistíme, že táto adresa je zapísaná v tejto oblasti na adresách 5CC5H a 5CC6H. Zmenou hodnoty na uvedených adresách môžeme ľubovoľne modifikovať adresu pre uloženie ovládacieho podprogramu pre výstup znaku na tlačiareň. Pripomeňme si, že na adrese 5CC5H je nižší byte a na adrese 5CC6H je vyšší byte adresy.

Nakoľko pre pripojenie tlačiarne použijeme stykovú jednotku s programovateľným obvodom PPI - MHB 8255 A, je potrebné v úvode (po nahratí) podprogramu nielen modifikovať adresu pre uloženie tlačového podprogramu, ale zároveň musíme aj tento obvod vhodne naprogramovať. Samotné naprogramovanie bude závisieť na spôsobe pripojenia. Pripojenie podľa obrázka uvedeného v predchádzajúcom bude nasledovné:

Port A - vstupný pre Joystick (PA0 až PA4). Zapojenie je možné realizovať podľa prílohy spravodaja č. 4.

Port B - výstupný. K nemu pripojíme datové vodiče z tlačiarne (PBO-D0, ..., PB7-D7).

Port C - Dolná polovica (PC0 až PC3) je výstupná a horná polovica (PC4 až PC7) je vstupná. Tento port nám bude slúžiť pre riadenie prenosu medzi počítačom a tlačiarňou. Pripojenie jednotlivých bitov bude nasledovné:

PC0 signál STROBE (alebo SC),
PC4 signál BUSY (alebo AC),

ďalej môžeme pripojiť:

PC1 RESET,
PC5 PAPER OUT (PE),
PC6 SELECTED (SLCT),
PC7 ERROR.

Samozrejme, že tieto ďalšie signály pripojíme iba v prípade, že ich naša tlačiareň má. Podľa pripojených signálov potom budeme modifikovať obslužný program.

Vieme, že obvod PPI obsahuje riadiaci register (CWR), do ktorého zapisujeme riadiace slovo a tri V/V brány (porty). CWR aj brány majú svoju adresu, prostredníctvom ktorej s nimi komunikujeme (do CWR je možné iba zapisovať). Pre zapojenie podľa uvedeného obrázka platia adresy:

Port A 1FH,
Port B 3FH,
Port C 5FH,
CWR 7FH.

Hodnoty adres sú dané spôsobom pripojenia adresových signálov A5, A6 a A7.

Obvod PFI naprogramujeme zapísaním vhodného riadiaceho slova (bytu) do CWR. Hodnota slova bude:

```
10011000B = 98H
|||||||
|||||||
|||||||+-----> nižšie bity brány C - výstupné,
||||||+-----> brána B - výstupná,
|||||+-----> brána B - režim 0,
||||+-----> vyššie bity brány C - vstupné,
|||+-----> brána A - vstupná,
|++-----> brána A - režim 0,
+-----> príznak riadiaceho slova.
```

Po zapísaní tohto slova na adresu CWR (7FH) bude obvod naprogramovaný podľa našich požiadaviek. Naprogramovanie dosiahneme krátkym podprogramom:

```
LD   A,98H      ; A <--- 98H
OUT  (7FH),A   ; pošli obsah A na periferiu s adr. 7F
RET                               ; návrat z podprogramu
```

V rámci tohto podprogramu však chceme okrem naprogramovania obvodu PFI uskutočniť aj niekoľko ďalších činností:

- inicializovať tlačiareň (ak poznáme riadiace kódy),
- vynulovať počítadlo riadkov, počítadlo znakov a nastaviť mód automatického stránkovania.
- naprogramovať tlačiareň (ak túto možnosť má). Touto variantou sa budeme zaoberať samostatne neskôr.

Úvodnú časť podprogramu budeme nazývať inicializačnou a označíme ju návestím INIC. V ďalšom budeme v podprogramoch používať mnemonické označenia:

```
RBYT .... riadiace slovo pre PFI,
CWR    .... adresa riadiaceho registra PFI,
LPRINT.... adresa začiatku nášho podprogramu pre tlač,
TLADR  .... adresa systémovej premennej (SCC5H),
ROUTA  .... adresa portu C,
DOUTA  .... adresa portu B (pre výstup dát),
TOKEN  .... adresa podprogramu v ROM pre 'rozbitie' kľúč. slov,
BRK    .... adresa podprogramu v ROM pre test klávesy BREAK,
KON    .... adresa pre ukončenie pri stlačení BREAK,
MAXZ   .... počet znakov na riadok,
MAXR   .... počet riadkov na stranu.
```

Uvedená mnemonika urobí program čitateľnejší a prehľadnejší, čo pomôže k ľahšiemu pochopeniu jeho činnosti.

Predpokladajme, že naša tlačiarneň má tzv. zvonček (BEL) a chceme, aby pri inicializácii zazvonila. Riadiaci kód pre BEL je 7. V takomto prípade bude podprogram pre inicializáciu vyzerat nasledovne:

```
INIC  LD  A,RBYT      ; A <--- RBYT
      OUT (CWR),A     ; zapíš obsah A do CWR
      LD  A,07       ; A <--- kód BEL
      CALL BYTOUT    ; obsah A vyšli do tlačiarne
      LD  HL,LPRINT  ; HL <--- adr. PPG pre tlač
      LD  (TLADR),HL ; a zapíš ju do sys. premennej
      LD  A,0        ; A <--- 0
      LD  (SKUTZ),A  ; nuluj počítadlo znakov
      LD  (SKUTR),A  ; nuluj počítadlo riadkov
      LD  (KLUC),A   ; nastav automatické stránkovanie
      RET           ; návrat z podprogramu
SKUTZ DS  1         ; počítadlo znakov
SKUTR DS  1         ; počítadlo riadkov
KLUC  DS  1         ; klúč pre stránkovanie
                        0=automatické stránkovanie
                        1=podprogram nestránkuje
```

Ak má tlačiarneň možnosť programového nastavovania do východzieho stavu, môžeme toto taktiež realizovať v inicializačnej časti, prípadne využiť signál RESET. Túto časť zaradíme pred inštrukciu LD A,07:

- uvedenie tlačiarne do východzieho stavu zapísaním sekvencie riadiacich slov, napr.:

```
LD  A,1BH      ; A <--- ESC
CALL BYTOUT    ; zápis obsahu A do tlačiarne
LD  A,40H      ; A <--- 40H
CALL BYTOUT    ; zápis obsahu A do tlačiarne
```

- uvedenie do východzieho stavu použitím signálu RESET. Signál predpokladáme pripojený na PC1 a je aktívny v logickej 0. Po naprogramovaní obvodu PPI bude PC1 výstupný a bude na ňom log. hodnota 0, čo spôsobí ihned RESET tlačiarne, nám teda zostáva iba zapísať do tohto bitu log. hodnotu 1:

```
LD  A,03      ; A <--- 3
OUT (ROUTA),A ; PC <--- obsah A
```

V prípade, že by bol signál RESET aktívny v log. 1, bola by inicializačná časť pre uvedenie tlačiarne do stavu počiatku nasledovná:

```
LD  A,02      ; A <--- 2
OUT (ROUTA),A ; PC <--- obsah A
LD  A,03      ; A <--- 3
OUT (ROUTA),A ; PC <--- obsah A
```

(pokračovanie)

Adresu premennej TLADR predpokladáme vždy 50C5, toto však nie je pravda. Konkrétna hodnota TLADR je závislá na pripojených perifériách. Keď máme cez Interface 1 pripojeny Microdrive, zmení sa hodnota 50C5 na 50F5. Túto skutočnosť musíme v ovládacích programoch tlačiarne rešpektovať. Hodnota adresy sa zmení aj pri pripojení diskovej jednotky (720 KByte). Chvilkou experimentovania sa nám však určite vždy podarí nájsť jej správne miesto. Pri pripojení diskovej jednotky navyše nepotrebujeme robiť ovládací program pre rozhranie Centronics, nakoľko je toto súčasťou tejto periférie (okrem iného).

V predchádzajúcom sme si ukázali, ako inicializujeme stykový obvod PPI, tlačiareň a ako si zapíšeme adresu nášho ovládacieho programu pre tlačiareň do oblasti systémových premenných.

Vieme, že náš program, ktorý začína na návěsti LPRINT musí zabezpečiť vyslanie znaku z registra A do tlačiarne. Znakov, ktoré prichádzajú do úvahy pre výpis je niekoľko druhov:

- riadiace znaky (CR, LF, ...), majú kód do 32,
- ASCII znaky, majú kód od 32 do 127,
- semigrafické znaky, majú kód od 128 do 164,
- kľúčové slová (tokens), majú kód od 165 do 255.

V prvej časti programu musíme teda zabezpečiť rozlíšenie znaku a na základe toho aj vhodnú činnosť programu. Uvedme si teraz význam jednotlivých znakov, ktoré môžu byť odovzdané nášmu ovládaciemu podprogramu, ktorý na základe ich významu musí zabezpečiť zodpovedajúcu vhodnú činnosť na tlačiarňi:

Odovzdaný znak		Vyslany znak		V ý z n a m
DEC	HEX	DEC	HEX	
12	0C	12	0C	Nová strana
13	0D	10	0A	Nový riadok
<128	<80	<128	<80	ASCII znak (>=32)
>164	>A4			Kódy kľúčových slov, vyššie sa postupnosť znakov

Nie vždy nám bude vyhovovať, aby bol kód počnúc od hodnoty 165 rozlíšený ako kľúčové slovo. Túto vlastnosť nebudeme využívať hlavne v oblasti programovania vlastných užívateľských znakov v tlačiarňi, prípadne pri vyžívaní grafických možností tlačiarne. Samozrejme, že takýto program nám neumožní vypisovať listingy programov na tlačiarňi, nakoľko nepozná kľúčové slová. Naopak, keď si urobíme ovládací program k Beta Basicu, musíme zabezpečiť aby rozpoznal okrem kľúčových slov Basicu aj kľúčové slová Beta Basicu.

Nacítame si teraz časť programu, ktorá realizuje rozpoznanie kódu a na jeho základe sa program vetví:

```
LPRINT DI          ; zákaz prerušenia
CP   OCH           ; porovnaj obsah A s hodnotou OCH
JR   Z,NOVS       ; ak sa rovnajú, skok na návěstie NOVS
CP   ODH           ; porovnaj obsah A s hodnotou ODH
JR   Z,NOVR       ; ak sa rovnajú, skok na návěstie NOVR
CALL RIADOK       ; kontrola počtu znakov na riadok
CALL STRANK       ; kontrola počtu riadkov na stranu
CP   BOH           ; porovnaj obsah A s hodnotou BOH
JR   C,VYST       ; ak je obsah A < BOH skok na VYST
SUB  OASH         ; odpočítaj od obsahu A hodnotu ASH
JP   TOKEN        ; skok na adresu TOKEN
```

Ovládací program má možnosť voľby počtu znakov na riadok a počtu riadkov na stránku, kontrolu zabezpečujú podprogramy RIADOK a STRANK. Skutočné hodnoty počtu riadkov a znakov sú uložené na adresách SKUTZ a SKUTR. V podprograme STRANK musíme navyše testovať hodnotu uloženú na adrese KLUC, ktorá určuje, či má program stránkovať automaticky, alebo nie. Oba podprogramy budú nasledovné:

```
RIADOK PUSH AF          ; uschovanie obsahu reg. A a príznakov
LD   A,(SKUTZ)         ; A ← skutočný počet znakov
CP   MAXZ              ; porovnaj obsah A s max.počtom znakov
JR   Z,RNUL           ; ak sú rovnaké, skok na RNUL
POP  AF                ; obnovenie obsahu reg. A a príznakov
RET                    ; návrat
RNUL  LD   A,0AH        ; A ← hodnotu 0AH
CALL BYTOUT            ; vyšli do tlačiarne
LD   A,0               ; A ← 0
LD   (SKUTZ),A         ; (SKUTZ) ← obsah A, t.j. nulu
POP  AF                ; obnovenie obsahu reg. A a príznakov
RET                    ; návrat
```

Nakoľko podprogram používa register A, čím by sa jeho obsah zničil, musíme si ho hneď na začiatku podprogramu uložiť do zásobníka, čím si hodnotu uchránime pred zničením. Keď sa pozrieme do časti programu za návěstím LPRINT vidíme, že obsahom registra A môže byť pri volaní tohto podprogramu iba hodnota ASCII znaku, alebo kód kľúčového slova. Znak na kód pre nový riadok a novú stránku testujeme ako prvé. Po uložení hodnoty z adresy SKUTZ do registra A, porovnáme túto hodnotu s hodnotou MAXZ, ktorú sme si v programe zadali. V prípade, že sú hodnoty rovnaké, t.j. počet vypísaných znakov sa rovná požadovanejmu počtu, vyšleme na tlačiarň kód pre nový riadok a na adresu SKUTZ si uložíme 0, čo znamená, že v riadku nie je ešte zapísaný žiaden znak. Ako vidieť, podprogram má dva body návratu, to ktorý z nich bude použitý závisí na počte vypísaných znakov. Vždy pred návratom z podprogramu si obnovíme obsah reg. A.

```

STRANK PUSH AF      ; uschovanie reg. A a príznakov
      LD  A, (KLUC)  ; A ← obsah adresy KLUC
      CP  0          ; stránkovanie dovolené ?
      JR  NZ, S11    ; nie, potom skok na S11
      LD  A, (SKUTR) ; A ← skutočný počet riadkov
      CP  MAXR       ; porovna s max. počtom riadkov
      JR  Z, SNUL    ; rovnajú sa, potom na SNUL
S11   POP  AF        ; obnovenie reg. A a príznakov
      RET           ; návrat
SNUL  LD  A, OAH     ; A ← kód pre novú stranu
      CALL BYTOUT    ; vyšli obsah A na tlačiareň
      LD  A, 0       ; A ← 0
      LD  (SKUTR), A ; ulož A na adresu SKUTR
      POP  AF        ; obnovenie reg. A a príznakov
      RET           ; návrat

```

V časti programu za návěstím LPRINT vidíme skoky na návestia NOVR, NOV5 a VYST. Tieto časti programu sú nasledovné:

```

NOVR  LD  A, 0       ; A ← 0
      LD  (SKUTZ), A ; na adr. SKUTZ ← A (nulu)
      LD  A, (SKUTR) ; A ← obsah SKUTR
      INC A          ; zvýš obsah A o 1
      LD  (SKUTR), A ; ulož späť
      LD  A, OAH     ; A ← OAH (kód pre nový riadok)
      JR  BYTOUT     ; vyslanie znaku na tlačiareň

```

```

NOV5  PUSH AF       ; uschovaj AF do zásobníka
      LD  A, (SKUTR) ; A ← obsah SKUTR
      INC A          ; zvýš obsah A o 1
      LD  (SKUTR), A ; ulož späť
      POP  AF        ; vráť AF zo zásobníka
      JR  BYTOUT     ; vyslanie znaku na tlačiareň

```

```

VYST  PUSH AF       ; uschovaj AF do zásobníka
      LD  A, (SKUTZ) ; A ← obsah SKUTZ
      INC A          ; zvýš obsah A o 1
      LD  (SKUTZ), A ; ulož späť
      POP  AF        ; vráť AF zo zásobníka
      JR  BYTOUT     ; vyslanie znaku na tlačiareň

```

Všetky tri uvedené časti programu upravujú počítadla riadkov na stránku, znakov na riadok a každý z nich skáče na časť programu s návěstím BYTOUT, ktorá zabezpečí vyslanie znaku na tlačiareň. Význam jednotlivých častí je zrejmý, teda ich nie je potrebné komentovať. Teraz už vlastne máme popísané všetky časti programu pre ovládanie tlačiarne, okrem časti ktorá realizuje konkrétne vyslanie znaku na tlačiareň. Táto časť začína na návěstí BYTOUT a je závislá na type rozhrania, s ktorým komunikujeme.

Napišeme si niekoľko typov "rutín" pre obsluhu najčastejšie sa vyskytujúcich rozhraní. Predpokladáme zapojenie podľa predchádzajúceho popisu.

Komunikácia s rozhraním Centronics:

```
BYTOUT OUT (DOUTA),A ; vyšli obsah A na adr. DOUTA
      LD  A,0          ; A ← 0
      OUT (ROUTA),A   ; vyšli obsah A na adr. ROUTA
      LD  A,OFFH      ; A ← OFFH
      OUT (ROUTA),A   ; vyšli obsah A na adr. ROUTA
SLUCKA CALL BRK      ; test klavesy BREAK
      JR  NC,KON      ; ak stlačená potom skok na KON
      IN  A,(RINAD)   ; ak nie čítaj stav tlačiarne
      BIT 4,A         ; testuj bit pripravenosti (BUSY)
      JR  NZ,SLUCKA   ; ak nepripravená skok na SLUCKA
      EI              ; prerušenie dovolené
      RET             ; návrat
```

Komunikácia s rozhraním IRPR: (komentár ako v predchádzajúcom)

```
BYTOUT PUSH AF
SL1  CALL BRK
      JR  NC,KON
      IN  A,(RINAD)
      BIT 5,A
      JR  NZ,SL1
SL2  CALL BRK
      JR  NC,KON
      IN  A,(RINAD)
      BIT 4,A
      JR  Z,SL2
      POP AF
      OUT (DOUTA),A
      LD  A,OFFH
      OUT (ROUTA),A
SL3  CALL BRK
      JR  NC,KON
      IN  A,(RINAD)
      BIT 4,A
      JR  NZ,SL3
      LD  A,0
      OUT (ROUTA),A
      EI
      RET
```

V rozhraní IRPR kontrolujeme okrem potvrdenia pripravenosti a prevzatia znaku (bit 4), aj celkovú pripravenosť tlačiarne komunikovať s počítačom (bit 5).

Komunikácia s rozhraním LOGABAX DZM 180:

```
      BYTOUT CF    60H
      JR    C,BYTOUT+6
      SBC  A,20H
      CPL
      PUSH AF
      LD   A,0FFH
      OUT (ROUTA),A
C2     CALL BRK
      JR   NC,KON
      IN  A,(RINAD)
      BIT 4,A
      JR  NZ,C2
      POP AF
      OUT (DOUTA),A
      LD  A,0
      OUT (ROUTA),A
C1     CALL BRK
      JR   NC,KON
      IN  A,(RINAD)
      BIT 4,A
      JR  Z,C1
      LD  A,0FFH
      OUT (ROUTA),A
      EI
      RET
```

Pri písaní časti podprogramov pre výstup znaku na tlačiareň sa musíme riadiť presne podľa časových priebehov signálov podľa údajov výrobcu. V rámci tohto podprogramu môžeme okrem testu pripravenosti tlačiarne testovať aj prítomnosť papiera (PE), hlásenie o chybovom stave a pod. Pri vyskytnutí chyby môžeme podprogram ukončiť vypísaním chybovej správy, podobne ako pri stlačení BREAK. Ak chyba papier, môžeme v slučke čakať na jeho založenie. Ukončenie programu pri stlačení BREAK bude nasledovné:

```
KON   RST 8
      DEFB 0CH
```

V závere si teraz uvedieme dva kompletne programy pre ovládanie tlačiarne. Prvý umožňuje aj výpis kľúčových slov, druhý kľúčové slová nepozná, čo znamená, že môžeme využívať pri tlači kódy znakov nad 164.


```
;>>> CENTRONICS <<<
;
; obsluzny program tlaciarne s rozhranim Centronics
; Prenos udajov zabezpeceny obvodom PPI
;
; port A ..... vstupny
; port B ..... vystupny ( DATA )
; port C ..... PC0 - STROBE
;                PC4 - BUSY
;
; Definicie
; =====
;
RBYT EQU 98H           ; riadiaci byt pre PPI
CWR EQU 7FH           ; adresa riad.reg. PPI
TLADR EQU 50C5H       ; alebo 50F5H pri MDRv
ROUTA EQU 5FH         ; adresa portu C
RINAD EQU 5FH
DOUTA EQU 3FH         ; adresa portu B
TOKEN EQU 0C10H       ; podprogr. v ROM
BPK EQU 1F54H         ; adr.ppg. v ROM-test BREAK
MAXZ EQU 3FH          ; pocet znakov na riadok
MAXR EQU 3BH          ; pocet riadkov na stranu
;
; inicializacia obvodu PPI a ulozenie adresy riadiaceho
; programu do systemovej premennej TLADR
;
;
; start. adresa 65001
INIC  ORG 0FDE9H      ; start. adresa 65001
      LD  A,RBYT      ; A <-- riadiace byt
      OUT (CWR),A     ; vysli do CWR PPI
      LD  A,18H       ; A <-- ESC
      CALL BYTOUT     ; zapis
      LD  A,7         ; A <-- 7 (BEL)
      CALL BYTOUT     ; zapis
      LD  HL,LPRINT   ; HL <-- adresu PPG pre tlac
      LD  (TLADR),HL  ; a uloz do sys.prem. TLADR
      LD  A,0
      LD  (SKUTZ),A   ; nulovanie pocitadiel
      LD  (SKUTR),A
      LD  (KLUC),A
      RET             ; navrat z podprogramu
;
; rezervovanie miesta pre pocitadla a kluc strankovania
;
SKUTZ DS 1
SKUTR DS 1
KLUC DS 1
;
; zaciatok tlacoveho programu, start programu je na
; prikazy LPRINT a LLIST
;
```

```
LPRINT DI ; zakaz prerusenia
CP OCH ; nova stranka ?
JR Z,NOVS ; ak ano, skok na NOVS
CP ODH ; novy riadok ?
JR Z,NOVR ; ak ano skok na NOVR
CALL RIADOK ; kontrola poctu znakov/riad.
CALL STRANK ; kontrola poctu riad./stranu
CP BOH ; ASCII znak ?
JR C,VYST ; ak ano, skok na VYST
SUB OASH ; nie, potom TOKEN
JP TOKEN ; skok do ROM

;
; obsluzna rutina pre novy riadok
;
NOVR LD A,0
LD (SKUTZ),A ; nulovanie pocitadla znakov
LD A,(SKUTR) ; zvyshenie poctu riadkov
INC A ; na stranu a ulozenie spat
LD (SKUTR),A
LD A,GAH ; A <-- kod pre novy riadok
JR BYTOUT ; vypis

;
; obsluzna rutina pre novu stranku
;
NOVS PUSH AF ; uschovaj AF
LD A,0
LD (SKUTR),A ; vynulovanie poc. riadkov
POP AF ; obnovenie AF
JR BYTOUT ; vypis

;
; obsluzna rutina pre vystup jedneho ASCII znaku
;
VYST PUSH AF ; uschovaj AF
LD A,(SKUTZ) ; zvyshenie pocitadla znakov
INC A ; na riadok o 1 a ulozenie
LD (SKUTZ),A ; spat
POP AF ; obnovenie AF

;
; obsluzna rutina pre vyslanie znaku do tlaciarne
; ktora kontroluje jej pripravenost a zaroven
; testuje aj klavesu BREAK. V pripade nepripravenosti
; tlaciarne program caka v slucke
;
BYTOUT OUT (DOUTA),A ; zapis byte na port B
LD A,0 ; realizacia signalu STROBE
OUT (ROUTA),A ; cez port C, bit 0
LD A,OFFH
OUT (ROUTA),A

SLUCKA CALL BRK ; test klavesy BREAK
JR NC,KON ; stlacena, potom skok na KON
IN A,(RINAD) ; test pripravenosti tlaciarne
JR NZ,SLUCKA ; nepripravena, potom skok
EI ; dovolenie prerusenia
RET ; navrat
```

```
; test na maximalny pocet znakov na riadok
; ak skutocny pocet znakov na riadok=pozadovany
; potom prechod na novy riadok
;
RIADOK PUSH AF                ; uschovaj AF
        LD  A,(SKUTZ)         ;
        CP  MAXZ              ; skutocny=pozadovany ?
        JR  Z,RNUL            ; ano, potom skok
        POP AF                ; nie, obnovenie AF
        RET                   ; navrat
RNUL   LD  A,OAH              ;
        CALL BYTOUT           ; novy riadok (vysle znak z A)
        LD  A,O
        LD  (SKUTZ),A        ; nuluj pocitadlo znakov
        LD  A,(SKUTR)
        INC A                  ; zvyš pocitadlo riadkov o 1
        LD  (SKUTR),A        ; a uloz spat
        POP AF                ; obnovenie AF
        RET                   ; navrat
;
; test na maximalny pocet riadkov na stranu
; ak skutocny pocet riadkov na stranu=pozadovany
; potom prechod na novu stranu
;
STRANK PUSH AF                ; uschovaj AF
        LD  A,(KLUC)          ;
        CP  0                 ; dovolene strankovanie ?
        JR  NZ,S11            ; nie potom skok
        LD  A,(SKUTR)
        CP  MAXR              ; skutocny=pozadovany ?
        JR  Z,SNUL            ; ano, potom skok
S11    POP  AF                ; nie, obnovenie AF
        RET                   ; navrat
SNUL   LD  A,OCH              ;
        CALL BYTOUT           ; nova strana (vysli znak)
        LD  A,O
        LD  (SKUTR),A        ; vynulovanie pocit.riadkov
        POP AF                ; obnovenie AF
        RET                   ; navrat
;
; prerusenie pri stlaceni BREAK
;
KON    RST  B                  ; restart
        DB  OCH                ; kod zdelenia BREAK .....
;
; koniec obsluzneho programu
;
        END
```

Uvedeny program je napisany preložený a odladený editorom a prekladačom EDIT/ASSEM ODP. V prípade, že použijeme assembler GENS, musíme namiesto DB písať DEFS a namiesto DB DEFB. Program preložíme, nahráme na pásku a po spätovnom nahratí do počítača

keď inicializujeme príkazom ~~RANDOMIZE~~ `USR 65001`. Pred nahratím programu do pamäte môžeme použiť príkaz `CLEAR 65000`, čím chránime oblasť pamäte v ktorej je program uložený pred prepísaním basicovským programom, alebo premennými.

Adresu, odkiaľ bude program uložený v pamäti určujeme pseudoinštrukciou `ORG`, teda uvedená adresa (65001) nie je predpísaná, ale pri preklade máme možnosť ju meniť.

Takýchto ovládacích programov si môžeme urobiť veľké množstvo, pričom všetky môžu byť prispôbolené pre naše účely. Ovládacie programy si môžeme dorobiť ku rôznym firemným programom, napr. Beta Basic, Pascal (HP4TM16,113 a pod.), Mega Basic, Tasword (aj slovensky, resp. český).

Teraz si ukážeme krátky príklad ako by sme počas inicializácie naprogramovali tlačiareň slovenskými znakmi, samozrejme ak tlačiareň má takúto možnosť (tlačiareň VT 21200 máme možnosť naprogramovať 10 znakmi a pod.). Uvádžam iba časti odlišné od predchádzajúceho:

```
INIC      .
          .
          .
          LD  B,200      ; B <-- počet kodov pre progr.
          LD  HL,ADDR-1
ZNAM      INC  HL
          LD  A,(HL)
          CALL BYTOUT
          DEC B
          JR  NZ,ZNAM
          RET
          .
          .
          .
LPRINT    DI
          CP  40H
          JR  Z,STRANK
          CALL RIADOK
          CP  0EH
          JR  Z,NOVS
          CP  0DH
          JR  Z,NOVR
          JP  VYST
          .
          .
          .
STRANK    PUSH AF
          CP  40H
          JR  Z,SNUL
          LD  A,(KLUC)
          CP  0
          JR  NZ,S11
```