



ZPRAVODAJ 4/88

ATARI **klub Praha**

INTERFACING



ROČNÍK II.

Vydává 487. ZO Svazarmu —

ATARI KLUB v Praze 4.

Šéfredaktor a vedoucí redakční rady

JUDr. Jan Hlaváček.

Zástupce šéfredaktora

ing. Stanislav Borský.

Obálku navrhl RNDr. J. Tamchyna.

Adresa redakce:

487. ZO Svazarmu — ATARI KLUB Praha

REDAKCE

poštovní příhrádka 51

100 00 Praha 10

Řídí redakční rada: V. Bílek, ing. J. Bis-

kup, RNDr. J. Bok, CSc., ing. S. Borský,

ing. V. Friedrich, ing. O. Hanuš, RNDr.

L. Hejna, CSc., Z. Lazar, prom. fyz., CSc.,

ing. M. Vavrda, F. Tvrdek.

Otisk povolen se souhlasem redakce při zachování autorských práv a s uvedením pramene. Rukopisy nevyžádané redakcí se nevracejí. Za původnost a věcnou správnost ručí autor.

Vychází šestkrát ročně. Neprodejné.

Členům klubu distribuováno zdarma.

Nepravidelné přílohy na objednávku jsou kompenzovány zvláštním klubovým příspěvkem.

Rozsah čísla 120 stran. Neprošlo jazykovou úpravou.

Tiskne PORS, závod 001,

reprografický provoz.

Do tisku předáno v IX/1988

Vydávání schváleno OV Svazarmu

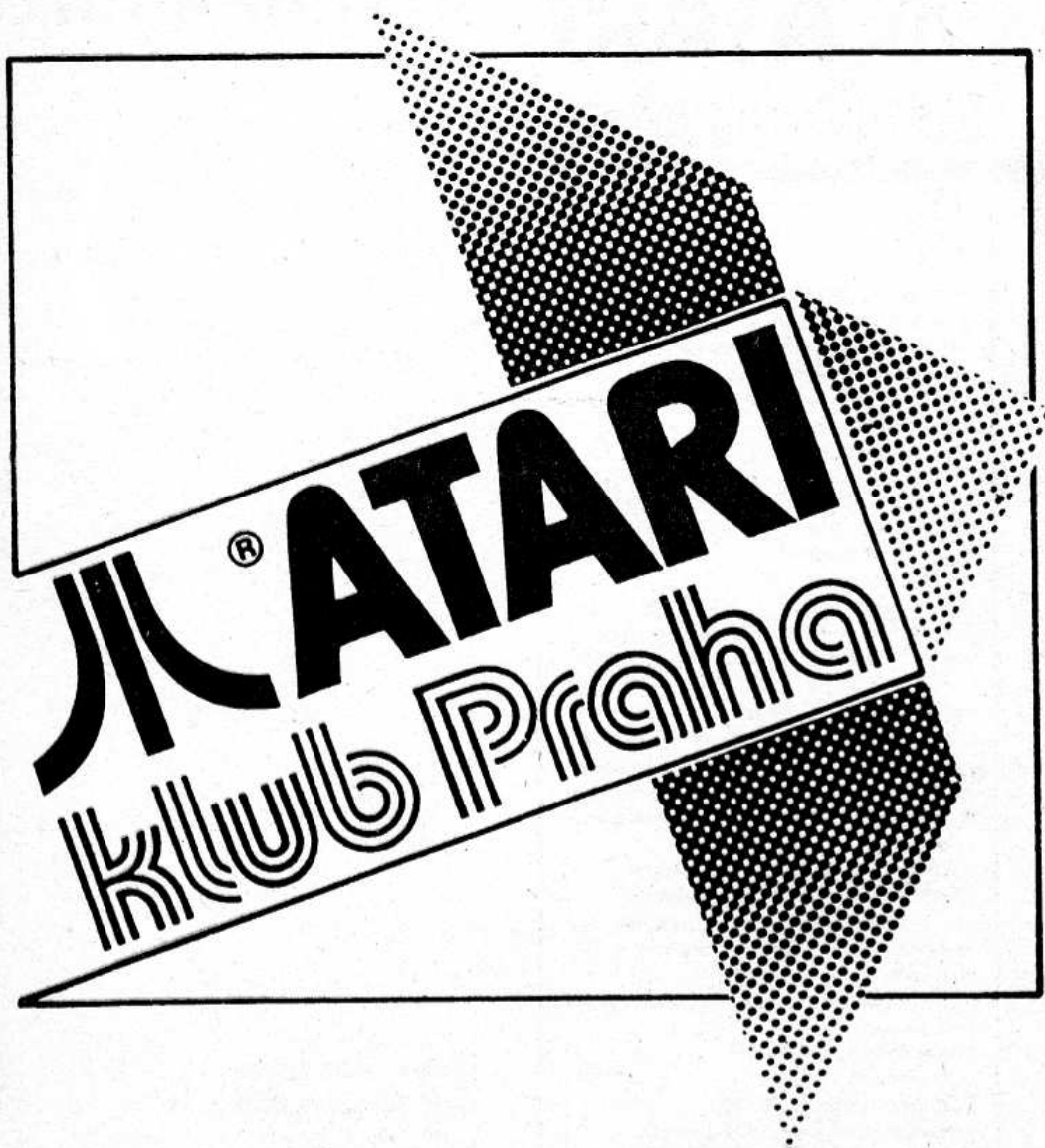
Praha 4 a OŠK ONV Praha 4.

Evidenční číslo ÚVTEI 86 042.

© ATARI KLUB Praha, 1988

Odborná redakce
Ing. Jaroslav Bisky

Technická redakce
Otilie Strnadová



Předmluva

Ve druhém letošním čísle ZAK jsme několika slovy uvedli připravovaný seriál, který jsme nazvali INTERFACING. Původně jsme předpokládali, že v každém následujícím čísle ZAK přineseme jeden či dva příspěvky získané od našich dopisovatelů, takže vytvoříme seriál, možná "bez konce". V závislosti na vyskytnuvších se vážných problémech s tiskem našeho zpravodaje jsme ale dospěli k závěru, že namísto seriálu nabídneme sborník, ve kterém budou soustředěny všechny dosud získané příspěvky. Vznikne sice úzce zaměřené číslo ZAK, ale nabídneme tím podklady k zamyšlení těm, kteří chtějí ATARI XL/XE využívat v co nejširším rozsahu (a také těm, kteří nás v anketě kritizovali, že přinášíme málo příspěvků v rubrice Koutek techniky).

Nechceme samozřejmě tvrdit, že tímto sborníkem jsou vyčerpány všechny možnosti. Naopak. Přivítáme další příspěvky na toto téma, které opět rádi otiskneme v některém dalším čísle ZAK.

* * *

Z příspěvků dopisovatelů sestavil, doplnil a odborně redigoval ing. Jaroslav Biskup.

Jmenný rejstřík autorů:

Bayer Miloš (kap. 5.1.1.)
 Bílek Karel (kap. 5.1.2.)
 Biskup Jaroslav (kap. Úvod, 1., 2., 3., 4. a odkaz v kap. 5.1.5.)
 Bok Jiří (odkaz v kap. 5.1.5.)
 Fajta Václav (odkaz v kap. 5.1.1.)
 Friml Jaromír (kap. 5.1.3.)
 Hájek Zdeněk (kap. 5.2.1.)
 Hána René (odkaz v kap. 5.1.5.)
 Chábera Jiří (kap. 5.1.5.)
 Chochlík Jaroslav (kap. 5.2.1., 5.2.2., 5.2.3.)
 Oškera Miroslav (kap. 5.1.1.)
 Pavlásek Jan (kap. 5.3.1.)
 Pavlíček Slavomír (kap. 5.1.1.)
 Pokorný Jan (kap. 5.1.1.)
 Pokorný Jiří st. (kap. 5.1.1.)
 Pokorný Jiří ml. (kap. 5.1.1.)
 Prunych Daniel (kap. 5.1.3.)
 Sládek Ján (kap. 5.2.1., 5.2.2., 5.2.3.)
 Štěrba J. (odkaz v kap. 5.1.5.)
 Štulpa Stanislav (odkaz v kap. 5.1.3.)
 Vavřda Miroslav (odkaz v kap. 5.1.5.)
 Vondruš František (odkaz v kap. 5.1.1.)
 Wagner H.C. (kap. 5.1.6.)

Přepis a technická redakce Otilie Strnadová. Obrázky překreslil ing. Petr Jedlička. Text byl pořízen v textovém editoru CAPEK 2.1, který pro potřeby redakce Zpravodaje ATARI KLUBU Praha poskytl jeho autor ing. Petr Jandík.

* * *

O b s a h :

1.	Úvod	5
2.	Popis standardních rozhraní	6
2.1.	Rozhraní CENTRONICS	6
2.2.	Rozhraní RS 232C, CCITT V.24	7
2.3.	Rozhraní IEC 652, IEEE 48	9
3.	Hradwarové možnosti počítače	10
3.1.	Paralelní porty (Joy-porty)	10
3.2.	Systémové sběrnice	11
3.3.	Sériový port	12
4.	Softwarové obslužení	16
4.1.	HATABS - tabulka zařízení	16
4.2.	Rozhraní CIO/handler	17
4.3.	Rozhraní handler/SIO	21
5.	Aplikace	23
5.1.	Paralelní porty (Joy-porty)	23
5.1.1.	Paralelní rozhraní pro tiskárnu	23
	- Řešení Miroslava Oškery a Slavomíra Pavlíčka (CVUT Praha)	23
	- Řešení Miloše Bayera (Rožtoky u Prahy)	25
	- Řešení Jana, Jiřího ml. a st. Pokorných (Praha)	29
	- Odkazy na řešení Václava Fajty a Františka Vondruše	32
5.1.2.	Sériové rozhraní pro tiskárnu	33
	- Řešení Karla Bílka	33
5.1.3.	Připojení dálnopisu	43
	- Odkaz na řešení Stanislava Štulpy	43
	- Řešení Daniela Prunycha (Hořovice)	43
	- Řešení Jaromíra Frimla (Jaroměř)	51
5.1.4.	Přídavná klávesnice (překlad z Happy Computer)	59
5.1.5.	Ovladače	61
	- Elektronický křížový ovladač (Jiří Chábera)	61
5.1.6.	Propojení počítačů (překlad z knihy H.C. Wagnera "Hofacker Book for your Atari Computer")	71
5.2.	Sériový port	74
5.2.1.	Připojení tiskárny (Zdeněk Hájek)	74
	- Modul pro připojení tiskárny pro k počítači ATARI (Jaroslav Chochlík, Ján Sládek)	85
5.2.2.	Připojení páskové jednotky (Jaroslav Chochlík, Ján Sládek)	88
5.2.3.	Sériová komunikační linka (Jaroslav Chochlík, Ján Sládek)	94
5.3.	Systémová sběrnice	104
5.3.1.	Programátor paměti EPROM (Jan Pavlásek)	104
	Použitá a doporučená literatura	120

* * *

Na stránkách ZAK jsme již uveřejnili několik příspěvků, zabývajících se problematikou připojování tiskáren k osmibitovým počítačům ATARI. Zájem o připojení počítače k různým periferním zařízením je stále značný. Proto věnujeme této otázce stálou pozornost.

Toto poněkud zvláštní číslo ZAK si klade za úkol zpřístupnit v obecnější formě širší veřejnosti ataristů řadu základních informací, z nichž ti zkušenější vytěží potřebné pro vlastní tvořivou práci. Uvedeme hardwarové i softwarové možnosti osmibitových počítačů ATARI z hlediska rozšiřování systému, popíšeme strukturu tvorby obslužných programů, tzv. handlerů a uvedeme řadu připravených konkrétních aplikací různé složitosti. Uvedeme zapojení adaptérů pro připojení tiskáren jak s rozhraním CENRONICS, tak i RS 232C včetně příslušných handlerů, uvedeme postup pro rozšiřování systému prostřednictvím konektorů pro připojení křížových ovladačů (dále jen "joy-portů") a systémové sběrnice. Uvedeme návod na stavbu programátoru paměti EPROM, způsob meziprocesorového propojení a další aplikace.

* * *

Připojení periferních jednotek k počítači ATARI je možno v zásadě buď sériově nebo paralelně. Pro kterou variantu se rozhodneme, to záleží na typu připojované jednotky a účelu použití.

U tiskáren a zapisovačů, které jsou základními perifériemi počítače, se nejčastěji setkáváme se dvěma typy standardních rozhraní: paralelním CENTRONICS a sériovým RS 232C.

Speciálním rozhraním, se kterým se můžeme u tohoto typu zařízení setkat, je rozhraní sérioparalelní IEC 625, označované rovněž jako IMS-2, IEEE 488, HP-IB, GP-IB. Toto rozhraní je používáno především v měřicích sítích, ale setkáváme se s ním u některých počítačů, kde je toto rozhraní použito jako standardní i pro periferní jednotky.

Při připojování periferních jednotek k počítači musíme řešit nejen otázku hardwarového propojení obou zařízení, ale musíme zajistit i příslušné softwarové obslužení, což znamená úpravu standardního handleru operačního systému počítače, nebo vytvoření handleru nového a jeho implementaci do systému.

Z hardwarového hlediska máme k připojení další periferní jednotky možnost využití joy-portů, systémové sběrnice nebo sériového portu. Konkrétní řešení má mimo jiné jinou přímou souvislost na profesionální softwarové produkty (grafické), které jsou zpravidla přizpůsobeny pro použití rozhraní ATARI/CENRONICS pro sériový port. Setkáme se však i s produkty, které využívají ke grafickému tisku joy-portů (např. STARTEXTER). Adaptace těchto profesionálních softwarových produktů pro použití s našimi jednoduchými adaptéry představuje vždy zásah do vlastního programu. To samo o sobě je odborně náročná práce, mnohdy i neřešitelná. Nicméně i s jednoduchým adaptérem můžeme zajistit grafický tisk. Instalujeme-li handler jako další I/O kanál, pak pomocí kódové tabulky pro řízení tiskárny a příkazů typu PUT jsme schopni zajistit grafický tisk potřebným způsobem pomocí vyšších programovacích jazyků,

případně můžeme vytvořit i kompletní grafický handler ve vyhovující formě. Označíme-li handler jménem tiskárny (P:), můžeme využívat i příkazů LPRINT apod.

Pro první pokusy je nejlepší začít s jednoduchým adaptérem připojeným prostřednictvím Joy-portů. Budeme se proto zabývat sériovým rozhraním typu RS 232C i paralelním rozhraním typu CENTRONICS. Univerzálním řešením je adaptér využívající systémové sběrnice, který umožňuje vytvoření libovolného typu rozhraní. Použijeme ji pro speciální aplikace jako univerzální port počítače.

Další možností je využití sériového portu, který však komunikuje nestandardním protokolem v úrovních TTL logiky. Profesionální adaptéry pro sériový port používají zákaznické obvody a jednočipové mikropočítače, zajišťující převod do standardu CENTRONICS.

Ve stručnosti se zmíníme o jednotlivých standardech. Jejich detailní popis přesahuje rámec našeho zpravodaje a proto se omezíme pouze na nejn nutnější informace.

* * * * *

2. POPIS STANDARDNÍCH ROZHRAŇÍ

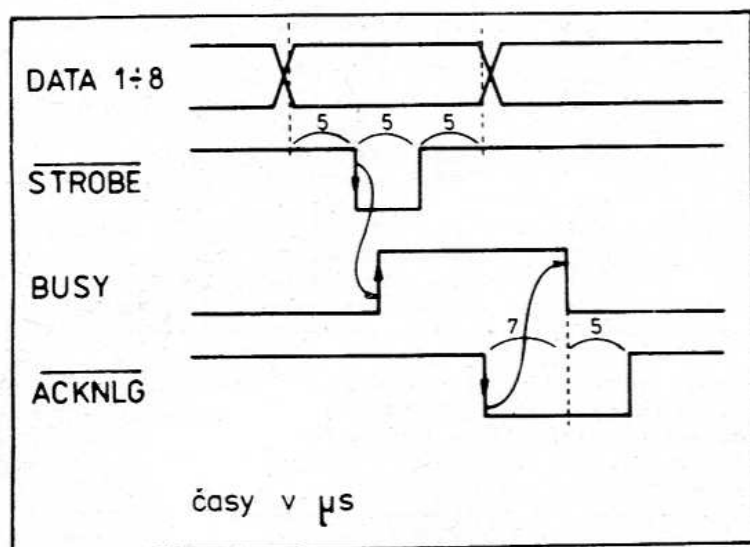
2.1. Rozhraní CENTRONICS

Rozhraní CENTRONICS je paralelní rozhraní, pracující v úrovních TTL logiky. Propojovací kabel by proto neměl být delší než 2 metry. Optimální je použití plochého kabelu se stačenými páry (twist). Použít můžeme i běžného plochého kabelu, u kterého střídáme signálový vývod se zemí. K připojení tiskáren je používáno zpravidla 36 pólového konektoru typu Amphenol 57-30360, resp. typu AMP 552274-1. Zapojení konektoru je normalizováno jen částečně. Bývá však dodrženo rozložení signálů DATA, STROBE, ACK, BUSY a GND u většiny tiskáren. Na tomto místě uvedeme funkční popis jednotlivých signálů.

signál	směr přenosu	význam
STROBE	do tiskárny	strobovací puls pro čtení dat ze sběrnice do tiskárny
DATA 1-8	do tiskárny	8 bitová sběrnice
ACKNLG	z tiskárny	potvrzení převzetí dat tiskárnou
BUSY	z tiskárny	neřprpravenost k příjmu dat ze sběrnice
PE	z tiskárny	indikace konce papíru
SELECT	z tiskárny	přprpravenost k příjmu dat ze sběrnice
GND		logická zem
AUTOFEED	do tiskárny	přepínač automatického řádkování
CHGND		ochranná zem
INIT	do tiskárny	inicializace tiskárny
ERROR	z tiskárny	indikace chyby (papír, OFF LINE)
SELECTIN	do tiskárny	volba tiskárny

U různých výrobců se setkáváme s odlišnými názvy některých signálů. Například STB místo STROBE, ACK místo ACKNLG, SLCT místo SELECT; 1-PRIME, RESET místo INIT, FAULT místo ERROR.

Průběh přenosu dat přes rozhraní CENTRONICS v handshaking cyklu, (tj. výměnou dat s potvrzením) lze ve stručnosti popsat takto: Po nastavení datové sběrnice DATA 1-8 je přiveden na sběrnici signál STROBE, tím je dán tiskárně pokyn k načtení dat do vstupních obvodů tiskárny. Tiskárna reaguje aktivací signálu BUSY, který zůstane aktivní až do ukončení zpracování dat tiskárnou. Po zpracování dat je tiskárnou aktivován signál ACKNLG, charakterizující ukončení přenosu. Následně je uvolněn signál BUSY a tiskárna je připravena k dalšímu přenosu. V jednoduchých aplikacích se obejdeme bez potvrzení převzetí dat signálem ACKNLG a přenos řídíme pouze pomocí signálů STROBE a BUSY.



Obr. 1

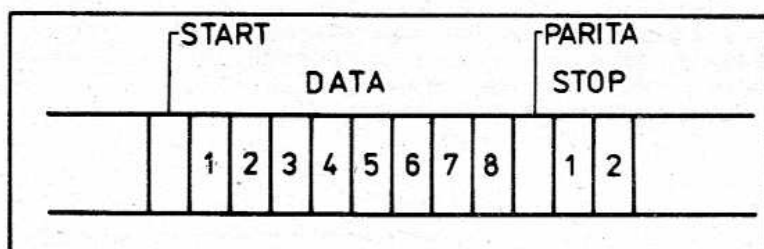
2.2. Rozhraní RS 232C, CCITT V.24

Jedná se o sériové rozhraní, pracující v úrovních +3 až +15 V pro logickou úroveň H a -3 až -15 V pro logickou úroveň L. Používá se jako univerzální rozhraní pro asynchronní přenos dat s přenosovou rychlostí 50 až 19200 Bd. K propojení se používá 25 pólového konektoru Cannon DP. 25 se standardním rozmístěním funkčních vývodů. Americký standard RS 232 se liší od evropského CCITT zejména označením signálů:

signál	vůvod	RS 232	V. 24	význam
FG, PG, CG	1	AA	101	ochranná zem
TD	2	BA	103	vysílaná data
RD	3	BB	104	přijímaná data
RTS	4	CA	105	vůzva k vysílání
CTS	5	CB	106	připravenost k přenosu
DSR	6	CC	107	připravenost dat
SG	7	AB	102	logická zem
DT	8	CF	109	detekce nosné
	11	CK	126	volba přenosové frekvence
SDT	12	SCF	122	detekce nosné zpětného kanálu
SCTS	13	SCB	121	připravenost k přenosu zpětného kanálu
STD	14	SBA	118	vysílaná data zpětného kanálu
TSET	15	DB	114	vysílací takt
SRD	16	SSB	119	přijímaná data zpětného kanálu
RSET	17	DD	115	přijímací takt
SRTS	19	SCA	120	vůzva k vysílání zpětným kanálem
DTR	20	CD	108.2	data připravena
			108.1	připojit datový kanál na vedení
SQD	21	CG	110	detekce jakosti příjmu
RI	22	CE	125	indikace volání
DTE	23	CH	111	volba modulační rychlosti
TSET	24	DA	113	přenosový takt modemu

Pro realizaci přenosu nejsou pochopitelně nutné všechny signály. Při realizaci jednoduchého handshaking cyklu vystačíme s obslužením signálů TD, RD, RTS, CTS a zemí SG. Při jednosměrném přenosu, což je i případ přenosu do tiskárny, vystačíme s jedinným signálovým párem.

Vlastní přenos je zahájen požadavkem přijímače na přenos aktivací signálu CTS. Vysílač na něj reaguje vysláním jednoho znaku, svou připravenost k přenosu indikuje aktivací signálu RTS. Přenos znaku je uveden jedním start bitem úrovně L, následuje 5 až 8 datových bitů s event. jedním paritní bitem a přenos znaku je ukončen 1; 1,5; 2 stop bity úrovně H. Není-li parita požadována, následuje za datovými bity bezprostředně ukončovací stop bit(y). Parametry přenosu (počet datových a stop bitů, parita, přenosová rychlost) jsou zpravidla volitelné a jsou závislé na druhu přenosu.



Obr. 2

2.3. Rozhraní IEC 652, IEEE 448

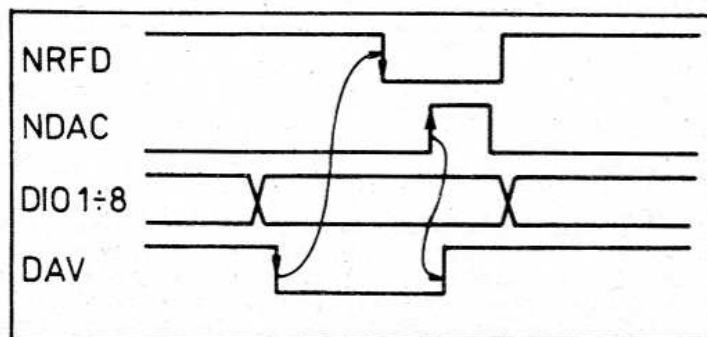
Jedná se o sériovoparalelní rozhraní určené především pro použití v měřicích sítích, umožňující propojení až 16 přístrojů. Pracuje s negativní TTL logikou, proto pro kabeláž platí stejná zásada jako u paralelního rozhraní. Na sběrnici lze pracovat s rychlostí až 250000 bytů/s. Na společné sběrnici mohou pracovat tři typy zařízení: řadič (controller), vysílač (talker) a přijímač (listener). V našem případě je řadičem mikropočítač, vysílačem např. disketová jednotka a přijímačem např. tiskárna nebo monitor. K připojování je používáno 24 pólového konektoru typu Amphenol při použití americké normy IEEE, nebo 25 (26) pólových konektorů podle doporučení norem IEC a RVHP. Rozmístění vývodů je normalizováno a je závislé na použitém konektoru, odchylky upravují příslušné národní normy (SSSR a NDR).

signál	IEC	IEEE	význam
DI01-8	1-4 14-17	1-4 13-16	8 bitová datová sběrnice
REN	5	17	nastavení dálkopisného provozu
EOI	6	5	při ATN=L konec přenosu dat
DAV	7	6	při ATN=1 identifikace přístrojů
NRFD	8	7	potvrzení platnosti dat sběrnice
NDAC	9	8	přijímač nepřipraven
IFC	10	9	data nepřevzata
SRQ	11	10	inicializace sběrnice
ATN	12	11	žádost o obsluhu řadičem
FG	13	12	ATN=0 na sběrnici jsou data
GND	18-25	18-24	ATN=1 na sběrnici jsou povely
			ochranná zem
			signální zem

Princip přenosu dat v handshaking cyklu lze stručně popsat takto:

Vysílač testuje sběrnici, zda je přijímač (e) připraven k příjmu. (NRFD=H a NDAC=L). Je-li přijímač připraven, ukládá

vysílač data na sběrnici a aktivuje signál DAV. Přijímač reaguje signálem NRFD a přijíma data, přičemž převzetí potvrzuje signálem NDAC. Vysílač na něj reaguje změnou signálu DAV a uvolněním datové sběrnice. Konverze je ukončena přechodem signálu NDAC vysílače do úrovně L a přechodem signálu NRFD přijímače do úrovně H (viz obr. 3).



Obr. 3

* * * * *

3. HARDWAROVÉ MOŽNOSTI POČÍTAČE

3.1. Paralelní konektory (Joy-ports)

Paralelní konektor je nejjednodušší I/O kanál, vhodný pro rozšiřování systému ATARI. Tyto konektory jsou vlastně přímými porty obvodu PIA 6520 (resp. 6820), což je standardní periferní adaptér mikroprocesorů řady 65xx. Tento obvod obsahuje dva nezávisle programovatelné 8 bitové I/O porty. U počítačů řady 400/800 jsou uživatelsky dostupné oba porty, zatímco u řady XL/XE je dostupný pouze jeden port. Druhý port (B) slouží pro obsluhu paměti počítače prostřednictvím obvodů MMU.

K ovládání portů slouží čtyři registry:

port A - PACTL (\$D302, 54018) a PORTA (\$D300, 54016)

port B - PBCTL (\$D303, 54019) a PORTB (\$D301, 54017)

Registry PACTL a PBCTL slouží k vlastnímu řízení portu. Určují, zda obsah registru PORTA, resp. PORTB, je datová nebo řídicí informace.

7	6	5	4	3	2	1	0	bit
0	0	1	1	1	*	0	0	obsah

* = 0 v registru PORTA resp. PORTB směr toku informace
 * = 1 v registru PORTA resp. PORTB data

Registry PORTA a PORTB jsou datové registry, jejichž význam je určen nastavením PACTL resp. PBCTL.

směr toku	7 6 5 4 3 2 1 0	bit
informace	* * * * *	bitově 1 = OUT, 0 = IN

data PORTA	7 6 5 4 3 2 1 0	bit
	4 3 2 1 4 3 2 1	obsah
	JOY1	JOY2

data PORTB (jen 400-800)	7 6 5 4 3 2 1 0	bit
	4 3 2 1 4 3 2 1	obsah
	JOY4	JOY3

Programování portu je velmi jednoduché:

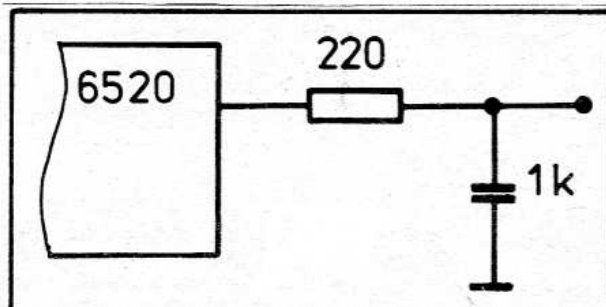
- do řídicího registru uložíme hodnotu \$30,48; tím určíme, že informace uložená do datového registru obsahuje směr toku informace portem na jeho jednotlivých bitech;
- do datového registru vložíme hodnotu odpovídající požadovanému směru přenosu na jednotlivých bitech portu;
- do řídicího registru uložíme hodnotu \$34,52; tím uvedeme port do normálního stavu pro zpracování I/O operací.

Jako jednobitový vstup můžeme využívat i trisser portu, který je obsluhován obvodem GTIA (resp. CTIA). Obsah hodnoty na trisser portu najdeme na adresách \$D010 pro JOY1, \$D011 pro JOY2, \$D012 pro JOY3 a \$D013 pro JOY4.

7 6 5 4 3 2 1 0	bit
0 0 0 0 0 0 0 *	obsah

* = obsah hodnoty trisser, 6. špička příslušného JOY

Obvod PIA pracuje v úrovních TTL logiky a lze jej přímo zatížit jednou zátěží jak pro vstup, tak pro výstup. Typické zapojení vývodů na konektor je na obr. 4 (platí i pro trisser):



Obr. 4 Zapojení vývodů portu obvodu PIA na konektor

3.2. Systémová sběrnice

Systémová sběrnice poskytuje kompletní adresové, datové a

Řídící signály vnitřního systému počítače. Umožňuje to libovolné rozšíření systému přímou implementací zařízení na vnitřní sběrnici. Aplikace vyžaduje hlubší znalost systému, zejména s ohledem na spolupráci jednotlivých prvků počítače pracujících na společné sběrnici. Propojování zařízení pak vyžaduje maximální obezřetnost. Vývod sběrnice nemá výkonové oddělení a přetížení nebo chybné zapojení může způsobit destrukci počítače. Prostřednictvím systémové sběrnice můžeme připojovat další periferní obvody (např. u nás rozšířené 8255, 8251 a 1012 apod.) a s jejich pomocí řešit nejrůznější aplikace.

3.3. Sériový port

Sériový port slouží k vzájemnému propojení všech standardních periferních zařízení ATARI, pracujících na společné sériové sběrnici. Zajišťuje příjem a vysílání dat na sériovém kanálu pomocí řídících signálů ukládaných do řídícího registru sériového portu SKCTL. Data jsou přenášena jako sériová posloupnost jednoho start bitu úrovně L, osmi datových bitů a jednoho stop bitu úrovně H. Frekvence hodinových pulsů, kterými je přenos řízen, je rovna přenosové rychlosti v Bd.

Sériový port lze využít pro následující činnosti:

a/ Sériový výstup

Slouží k výstupu dat, které procesor ukládá paralelně do osmibitového registru SEROUT. Data jsou pak naplněna do výstupního posuvného registru. Po přenesení dat do výstupního registru je generováno přerušení procesoru SEROUT Empty. Tím je procesoru umožněno naplnění registru SEROUT dalším bytem. Data v posuvném registru jsou automaticky doplněna start/stop bity a sériově vyslána na port (jako první LSBit).

Výstupní data jsou normálně vysílána v logických TTL, mohou však být vysílána v dvoutónové akustice. Akustický mód je nastaven 3. bitem řídícího registru SKCTL. V tomto případě je úroveň H reprezentována audio kanálem 1. a úroveň L audio kanálem 2. Frekvence kanálu 2 musí být přitom nižší jak frekvence kanálu 1.

Sériový výstup může být prostřednictvím 7. bitu registru SKCTL nastaven trvale na úroveň L (resp. audio kanál 2).

b/ Sériový výstup hodin

Data sériového výstupu jsou aktualizována náběžnou hranou hodinového pulsu, ten pak končí uprostřed výstupního datového bitu. Frekvence hodinových pulsů je určena audio kanálem 4., audiokanálem 2. nebo externě (podle nastavení registru SKCTL).

c/ Sériový vstup

Slouží k sériovému vstupu dat. Příjem je zahájen přijetím osmibitového datového slova s odpovídajícími start/stop bity vstupním posuvným registrem. Přijatá data jsou automaticky přenesena do osmibitového vstupního registru SERIN. Po naplnění

registru je generováno přerušení procesoru SERIN Ready. Procesor pak musí data zpracovat ještě před přijetím dalšího slova, jinak dojde k zahlcení kanálu, což je signalizováno 5. bitem stavového registru SKSTAT. V důsledku to znamená ztrátu informace, proto je nutno tento bit testovat při každém čtení registru SERIN. Rovněž tak je nutno testovat i 7. bit téhož registru, kde je detekována chyba přenosu.

d/ Přímý sériový vstup

Sériová data je možno přijímat přímo, mimo posuvný registr, čtením 4. bitu registru SKSTAT.

e/ Vstup/výstup hodinových pulsů

Při příjmu hodinových pulsů z externího zařízení slouží jako vstup. Naopak, je-li hodin počítače použito pro řízení externích zařízení, slouží jako výstup (podle stavu SKCTL). Data sériového vstupu jsou aktualizována závěrnou hranou hodinového pulsu, výstupní data hranou náběžnou.

f/ Asynchronní sériový vstup

Slouží k sériovému příjmu dat bez strobování hodinovými pulsy. Vstupní posuvný registr je vzorkován frekvencí audio kanálu 4. Ke zvýšení spolehlivosti lze použít paralelní vzorkování kanálem 3 (3. bit registru AUDCTL = H). V tomto režimu jsou kanály 3 a 4 spouštěny každým start bitem přijímaného slova, proto je bezpodmínečně nutné nastavit frekvenci kanálů 3 a 4 tak, aby se příliš nelišila od přenosové rychlosti.

Přenos sériovým portem je možný v 6 módech, které jsou nastaveny bity 4, 5 a 6 řídicího registru SKCTL.

SKCTL	OUT	OUT	IN	IN/OUT	Funkce
6 5 4	Rate	Clock	Rate	Clock	
0 0 0	ext	ext	ext	ext IN	vyšláč i přijímač řízen z externího zdroje
0 0 1	ext	ext	ch4	ext IN	vyšláč z externího zdroje, přijímač asynchronní z kanálu 4
0 1 0	ch4	ch4	ch4	ch4 OUT	vyšláč i přijímač řízen z kanálu 4
1 0 0	ch4	ch4	ext	ext IN	vyšláč řízen kanálem 4, přijímač z externího zdroje
1 1 0	ch2	ch2	ch2	ch4 OUT	vyšláč řízen kanálem 2, přijímač řízen kanálem 4
1 1 1	ch2	ch2	ch4	-	vyšláč řízen kanálem 2, přijímač asynchronní z kanálu 4

Obsazení dalších bitů registru SKCTL (\$D20F, 53775):

7 6 5 4 3 2 1 0 bit
D7 D6 D5 D4 D3 * * *

- D7 - nastavení sériového výstupu na nulu (mezeru)
D6, D5, D4 - nastavení módu sériového portu
D3 - dvoutónový audio mód

Stavový registr sériového portu SKSTAT (\$D20F, 54775):

7 6 5 4 3 2 1 0 bit
D7 D6 * D4 * * D1 1

- D7 = 0 - chyba příjmu datového paketu
D6 = 0 - zahlcení vstupního kanálu
D4 = 0 - přímé čtení vstupního kanálu
D1 = 0 - vstupní posuvný registr obsazen

Alokace dalších registrů pro obsluhu sériového portu:

- SKRES (\$D20A, 53770) - slouží k vymazání stavového registru
SKSTAT
SERIN (\$D20E, 53774) - vstupní register portu
SEROUT (\$D20D, 53773) - výstupní registr portu

Komunikační protokol sériového portu

Komunikační protokol je založen na řízení portu pouze ze strany počítače. Periferní zařízení vystupuje jako pasivní zařízení, předávající data pouze na požádání. Na sériovém portu smí probíhat pouze jediná operace. Každá chyba zachycená při přenosu způsobí chybové hlášení a ztrátu předávané informace.

Komunikační protokol zahrnuje následující prvky:

- řídící paket vysílaný počítačem,
- potvrzení převzetí řídícího paketu periferním zařízením
- datový paket z/do počítače,
- potvrzení úspěšnosti přenosu periferním zařízením

Řízení paket

Komunikační protokol umožňuje tři typy příkazů (vysílání, příjem dat a přímý příkaz), jejichž společným znakem je počet vysílaných bytů (5). Celý řídící paket je strobován počítačem pomocí signálu COMMAND. Formát paketu je:

ID
COM
AUX1
AUX2
CHSUM

- ID - určuje typ adresovaného zařízení
\$31-34, 49-52 D1-D4 disketová jednotka
\$40, 64 P1 tiskárna
\$50-53, 80-83 R1-R4 RS 232C
\$60, 96 C magnetofon

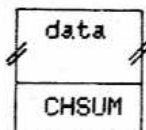
COM - určuje požadovaný příkaz
 \$52,82,R čtení
 \$57,87,W zápis
 \$53,83,S status
 \$50,80,P zápis bez kontroly
 \$21,33,! formátování
 \$22,34," formátování v ED
 \$4E,78,N čtení konfiguračního bloku v DD
 \$4F,79,O zápis konfiguračního bloku v DD

AUX1,2 - obsahuje doplňující informace pro periferní zařízení
 CHSUM - kontrolní součet (aritmetický součet prvních 4 bitů s přičtením přenosu po každém součtu)

Zaadresovaná periferie odpovídá po převzetí řídícího paketu vysláním ACK bytu (\$41,65) počítači.

Datový paket

Následuje po příjmu ACK počítačem; jeho formát je :



Datový paket je sestaven počítačem nebo řadičem periferie, která o to byla počítačem požádána. Řadič zachovává stejný formát datového paketu jako počítač. Hodnota CHSUM je aritmetickým součtem všech datových bytů v paketu, které jsou doplňovány o přenos podobně, jako u přenosu příkazového paketu.

V případě přenosu počítač - periferie reaguje periferie na přijatá data vysláním ACK bytu nebo NAK bytu (\$4E,78), je-li datový paket nepřijatelný pro volanou periferii.

Ukončení přenosu

Periferie vysílá počítači COMPLETE byte (\$43,67), je-li požadovaná operace v periférii dokončena. Nebyla-li operace zpracována odpovídajícím způsobem, vysílá periferie ERROR byte (\$45,69).

* * * * *

4. SOFTWAREOVÉ OBSLOUŽENÍ ZARÍZENÍ

Dalším krokem, který je třeba zvládnout při implementaci rozšiřujícího zařízení do systému počítače, je vytvoření nového nebo upravení rezidentního obslužného programu handleru.

Software, stojící mezi uživatelem a hardware a zajišťující obsluhu zařízení, pracuje na třech samostatných úrovních:

- CIO (Central I/O Utility),
- individuální handlery jednotlivých zařízení,
- SIO (Serial Bus I/O Utility)

CIO stojící na nejvyšší úrovni vykonává tyto funkce:

- dodání jména zařízení do tabulky zařízení (HATABS),
- ošetřování řídicího bloku IOCB,
- manipulace se záznamy,
- manipulace uživatelským bufferem.

Handlery stojící na úrovni zpracování pod CIO vykonávají tyto funkce:

- inicializace zařízení po zapnutí a systémovém RESET,
- provádění příkazů OPEN a CLOSE,
- obsluha datových přenosů,
- zajišťování speciálních funkcí,
- výkon příkazů,
- obsluha datového bufferu.

Na nejnižší úrovni stojí SIO, které je využíváno handlery pracujícími po sériové sběrnici. SIO vykonává tyto funkce:

- řízení sériové sběrnice v příslušném protokolu,
- opakování operace při neúspěšném pokusu,
- generování chybového statusu sběrnice.

Pro komunikaci na každém z přístupů do systému je použito následujících řídicích struktur:

uživatel/CIO	IOCB	I/O Control Block
	ZIOCB	Zero-Pase IOCB
handler/SIO	DCB	Device Control Block

4.1. HATABS - Device Table - tabulka zařízení

Tabulka zařízení je rezidentní část paměti RAM obsahující jednoznaková jména zařízení s příslušnou adresou tabulky vektorů handleru obsluhujícího toto zařízení prostřednictvím CIO. Tabulka obsahuje tyto rezidentní handlery (tj. takové, které jsou po inicializaci v systému k dispozici):

K - klávesnice	S - displej
E - editor	C - magnetofon
P - tiskárna	

K instalaci nového handleru je nutné tabulku doplnit o nové zařízení, poté co je tabulka inicializována. Tabulka začíná na adrese \$031A, její velikost je 38 bytů a lze do ní umístit až 12 položek - handlerů. (Poslední 2 byty v tabulce jsou nulové).

Formát tabulky:



CIO prohlíží tabulku od konce na začátek. Proto v případě vícenásobného výskytu jména zařízení v tabulce je uplatněno jméno postavené nejnižší v tabulce. K přístupu do příslušného handleru využívá CIO adresy obsažené v tabulce vektorů.

4.2. Rozhraní CIO/Handler

Tabulka vektorů každého z handlerů má následující formát:

nejnižší adr.	OPEN	vektor	2 byte
	CLOSE	vektor	2 byte
	GET BYTE	vektor	2 byte
	PUT BYTE	vektor	2 byte
	GET STAT	vektor	2 byte
	SPECIAL	vektor	2 byte
nejvyšší adr.	JMP Init Code		3 byte

Adresa 1.bytu tabulky vektorů je umístěna v tabulce zařízení (HATABS). Prvních šest položek tabulky jsou vektory obsahující adresu-1 příslušné rutiny handleru zajišťující požadovanou funkci. Sedmá položka je pak instrukcí JMP v kódu procesoru 6502 a obsahuje adresu inicializační rutiny handleru. Každý z přístupů typu uživatel/CIO je převeden na jedno nebo více volání některého z přístupů handleru definovaného v tabulce vektorů. Tabulka vektorů obsahuje adresy rutin handlerů pro předurčené funkce zajišťované CIO. Parametry nutné pro výkon jednotlivých operací jsou handleru předávány prostřednictvím registrů ZIOCB (\$0020). Obvykle je A registr využit pro uložení dat, X registr obsahuje 16 násobek identifikačního čísla bloku IOCB a Y registr obsahuje status

provedené operace. ZIOCB je pracovní kopie vlastního IOCB, se kterým CIO pracuje (při výkonu některých operací CIO může měnit parametry bloku, aniž by tím ovlivnil nastavení vlastního IOCB).

Dále popíšeme funkce jednotlivých rutin handleru tak, jak jsou uloženy v tabulce vektorů. Popis funkcí je doporučením i pro vlastní tvorbu dalších individuálních handlerů.

OPEN

Rutina je volána v návaznosti na příkaz OPEN CIO. Handler potvrdí platnost parametrů a vykoná inicializaci příslušného zařízení v souladu s dodanými parametry. Z pozice handleru jsou zajímavé následující parametry:

X registr ukazatel originálu IOCB (obvykle 16 násobek)
Y registr status (pro neimplementovanou funkci \$92)
ICDNOZ (\$0021) číslo zařízení (obvykle 1 - 4)
ICBALZ/ICBAHZ (\$0024/\$0025) adresa specifikace
 zařízení/jména souboru
ICAX1Z/ICAX2Z (\$002A/\$002B) specifické doplňující informace

Po vykonání OPEN operace je do Y registru zapsán výsledný status a požadovaný kanál je obsazen (otevřen).

CLOSE

Tato rutina je volána v návaznosti na příkaz CLOSE CIO. Handler uzavře přístup k zařízení, resp. souboru, který byl otevřen a zajistí:

- vyslání dat, které zůstaly v bufferu handleru příslušného zařízení,
- vyznačí EOF (End of File),
- aktualizuje adresář, relokační tabulky apod.

Vstupní parametry rutiny jsou stejné jako u rutiny OPEN. Po vykonání operace CLOSE vrací handler řízení CIO po uzavření požadovaného kanálu a zapsání statusu do Y registru procesoru.

GET BYTE

Rutina GET BYTE je volána v návaznosti na příkazy GET CHARACTERS nebo GET RECORD CIO. Handler dodá do A registru (střadače) procesoru jeden datový byte nebo zapíše chybový status do Y registru procesoru.

Vstupní parametry rutiny jsou totožné jako u předchozích dvou. Handler nemá být ošetřený čtení dat timeoutem, tj. ukončením operace při překročení stanoveného časového limitu, musí průběžně monitorovat stav BRKKEY (\$0011) a v případě BREAK stavu zapsat do Y registru status \$80.

Před vlastním výkonem operace kontroluje CIO, zda-li je zařízení, resp. soubor otevřen, nebo není-li otevřen pouze pro výstup. V takovém případě je operace ukončena.

PUT BYTE

Obdobným způsobem pracuje i rutina PUT BYTE v

návaznosti na příkazy PUT CHARACTERS a PUT RECORD CIO. Handler pracuje opět s registry A a Y procesoru, rovněž tak i vstupní parametry handleru jsou tytéž.

Handler vysílá datový byte přímo do specifikovaného zařízení nebo do bufferu, který přímo spravuje. Výsledný status je uložen v Y registru. Pracuje-li handler s bufferem, musí do zařízení přenést data ještě před návratem do CIO. I v této rutině kontroluje CIO, zda je zařízení/soubor otevřeno nebo není-li otevřeno pouze pro vstup.

Některé handlers pracují v interpretu std. ATARI BASIC volají rutinu PUT BYTE přímo bez přístupu prostřednictvím CIO. V tom případě ZIOBC může či nemusí mít souvislost s voláním této rutiny, protože handler musí použít výstupní úroveň IOCB k získání informací, které jsou normálně dostupné z ZIOBC. Rovněž tak ochrana přístupu k neotevřeným zařízením resp. souborům, která jsou prováděna při normálních CIO operacích, je zde pomínuta.

GET STAT

Rutina je volána v návaznosti na příkaz GET STATUS CIO. Vstupní parametry handleru jsou stejné jako u předchozích rutin. Handler dodá stavovou informaci (4 byty) z řadiče zařízení a uloží je do paměti na DVSTAT (\$02EA) až do DVSTAT+3 a rovněž dodá do Y registru procesoru výsledný status operace CIO. K volání rutiny není podstatné, je-li soubor/zařízení otevřeno.

SPECIAL

Rutiny typu SPECIAL doplňují standardní soubor dříve popsaných operací (např. RENAME, DRAW apod.). Hodnota příkazového bytu IOCB SPECIAL je větší jak \$00. Handler musí kontrolovat, je-li požadovaná operace v systému přípustná.

Vstupní parametry rutiny:

- | | |
|-----------------------------------|---------------------------------|
| - X registr | ukazatel originálu IOCB |
| - Y registr | status |
| - ICDNOZ (\$0021) | číslo zařízení |
| - ICCOMZ (\$0022) | příkazový byte |
| - ICBALXZ/ICBAHZ (\$0024, \$0025) | adresa bufferu |
| - ICBL LZ/ICBLHZ (\$0028, \$0029) | délka bufferu |
| - ICAX1Z/ICAX2Z (\$002A, \$002B) | specifické doplňující informace |

Handler provede požadovanou operaci, je-li implementována v systému a vrací se do CIO se zápisem statusu do Y registru procesoru.

Ošetření chyb

Systémem jsou ošetřeny chyby, ke kterým dochází na výstupní úrovni CIO a chyby na sériové sběrnici SIO. Zbývající chyby musí být ošetřeny přímo handlerem příslušného zařízení:

- překročení rozsahů parametrů,
- přerušení klávesou BREAK,
- nepřípustný příkaz,
- čtení po EOF.

Při ošetření chyb handlerem postupujeme dle dále uvedených pomůcek:

- pokus o jednoduchou obnovu (opakování, predikce chyby),
- pokus o interaktivní obnovu uživatelem,
- ztráta dat v malém rozsahu,
- provedení všech pokusů o zachování celistvosti souboru, zpracovávaného zařízením.

Alokace handleru

Umístění nerezidentního handleru vyžaduje vyhrazení prostoru v paměti. Prostor je nutno vybírat s ohledem na bezkonfliktní spolupráci s ostatními částmi operačního systému, včetně dalších nerezidentních handlerů.

Oblast nulté stránky RAM

V nulté stránce RAM není dostatek volného místa pro přímé umístění programu. Je-li proto nutné do této oblasti umístit nerezidentní handler, musí handler sám zajistit uložení a obnovu paměťové oblasti, kterou bude využívat. Při výběru oblasti nulté stránky je třeba postupovat obezřetně a s ohledem na následující kritéria:

- do vymezené oblasti je zakázán přístup z přerušovacích rutin (jsou-li použity),
- je zakázán veškerý přístup během modifikace paměti, tj. instalace handleru a obnova paměti.

Dále je uveden příklad použití techniky "úklidu" a obnovy oblasti nulté stránky s použitím zásobníku.

```

LDA COLORS      ;(např.)
PHA             ;uložení na zásobník
LDA COLORS+1
PHA

LDA HPOINT      ;pointer handleru
STA COLORS
LDA HPOINT+1
STA COLORS+1

xxx (COLORS),Y  ;vaše aplikace použití COLORS
PLA             ;obnova paměti
STA COLORS+1
PLA
STA COLORS

```

V uvedeném případě nesmí být volána až do obnovení proměnné COLORS Display Handler nebo Screen Editor, používající tuto proměnnou.

Oblast RAM mimo nultou stránku

Pro použití oblasti nad nultou stránkou máme možnost výběru jedné ze tří metod:

1. dynamická alokace handleru během inicializace nastavením MEMLO (\$02E7),
2. uložení handleru do RAM jako řady proměnných, v tom případě se MEMLO nastaví během ukládání handleru automaticky,
3. nahrazení rezidentního handleru jiným, s využitím oblasti RAM, kterou využíval původní handler.

Zásobník

V mnoha případech lze handlerem využívat i zásobník. Na zásobník lze uložit více jak dva tucty bytů. Stav zásobníku je však nutno kontrolovat na přetečení a nutno rezervovat vždy místo pro obsluhu přerušení.

4.3. Handler/SIO rozhraní

Zařízení komunikující na sériové sběrnici jsou obsluhována handlery sériové sběrnice pomocí SIO, které provádí veškeré přístupy na periferní zařízení v odpovídajícím protokolu sběrnice. SIO podporuje tyto funkce:

- časování a formátování dat z počítače,
- detekce chyb, pokusů a stavů,
- timeout sběrnice,
- přenos dat mezi sběrnici a volaným bufferem.

Jediným vstupem SIO Utility pro všechny operace, které vykonává, je SIOV (\$E459). Parametry jsou předávány prostřednictvím DCB (\$0300) (Device Control Block), který obsahuje:

- DDEVIC (\$0300) identifikátor typu periferního zařízení,
nastavený handlerem před voláním SIO
- DUNIT (\$0301) pořadové číslo zařízení, nastaveno handlerem
před voláním SIO, hodnota je dosazena z IODNOZ,
SIO obslouží zařízení DDEVIC s pořadovým číslem
DUNIT-1
- DCOMM (\$0302) příkazový byte, nastavený handlerem před
voláním SIO
- DSTATS (\$0303) status zařízení, proměnné s přístupem
jak pro vstup, tak pro výstup. Handler používá
pro indikaci stavu SIO po vyslání příkazového
paketu
a jeho zpracování. SIO používá DSTATS k indikaci
stavu handleru ve vztahu k požadované operaci.

Hodnota DSTATS před volání SIO:

7 6 5 4 3 2 1 0
W R * * * * *

WR 00 žádný přenos dat
 01 data přijata ze zařízení
 10 data vysílána do zařízení
 11 nepřipustná kombinace

Hodnota DSTATS po volání SIO:

7 6 5 4 3 2 1 0
 status kód

Status kódy: \$01 operace dokončena
 \$8A timeout zařízení
 \$8B zařízení neodpovídá
 \$8C chyba při čtení paketu
 \$8E zahlcení kanálu
 \$8F chyba kontrolního součtu
 \$90 chyba mimo zařízení

DBUFLO/DBUFHI (\$0304/\$0305) dvoubytový pointer bufferu pro vstupní/výstupní data, resp. status, nastaveno handlerem;
 DTIMLO (\$0306) timeout zařízení, nastavený handlerem v jednotkách 64-60 sec (např. 5 reprezentuje čas 6,4 sec.);
 DBYTLO/DBYTHI (\$0308/\$0309) délka bufferu, resp. čítač bytů, nastavený handlerem. Indikuje počet datových bytů přenášených bufferem. Parametr není přípustný, je-li WR v DSTATS nastaveno na 0.
 DAUX1/DAUX2 (\$030A/\$030B) pomocné informace, nastavené handlerem.

Ošetření chyb

SIO ošetřuje mnohé chyby vzniklé na sériové sběrnici přímo.

1. timeout sběrnice - SIO zajišťuje uniformitu timeout ACK bytu příkazových i datových paketů v jednotkách 1/60 sec, které jsou určeny hodnotou DTIMLO nastavenou handlerem (nastavení odstupe COMPLETE bytu od posledního ACK bytu);
2. chyby sběrnice - SIO detekuje a informuje UART o zahlcení kanálu a chybě v přenosech paketů, tyto chyby jsou monitorovány s každým čteným bytem;
3. chyby kontrolního součtu - SIO potvrzuje platnost kontrolního součtu přijatých datových paketů a generuje kontrolní součet pro celý přenos;
4. chybná reakce zařízení - SIO kontroluje správnost příjmu ACK a COMPLETE bytů (ACK=\$41, COMPLETE=\$43)
5. opakování operace na sběrnici - není-li napoprvé příkaz na sběrnici zpracován bezchybně opakuje SIO vysílání příkazového paketu až 14 krát, po příjmu COMPLETE kódu pak může přenášet datový paket.

5. APLIKACE

5.1. Paralelní porty (Joy-ports)

5.1.1. Paralelní rozhraní pro tiskárnu

Jako první aplikaci uvedeme tři obdobná řešení paralelního rozhraní od různých autorů. Přenos dat z počítače je řešen jako osmibitový a proto jsou uvedené aplikace určeny především pro znakový výstup. Předložená řešení jsou až na drobné detaily funkčně shodná. Všechna byla prakticky odzkoušena na různých typech tiskáren. Ponecháváme proto na úvaze uživatele, pro kterou variantu se rozhodne.

Řešení Miroslava Oškery a Slavomíra Pavlíčka (CVUT Praha)

Většina u nás dostupných tiskáren je vybavena paralelním rozhraním. To nás vedlo k pokusu připojit tiskárny tohoto typu k mikropočítači ATARI. Vyzkoušeli jsme spojení s tiskárnou ROBOTRON 6313 (NDR), vybavenou rozhraním CENTRONICS a PRT-80 (MLR), vybavenou rozhraním IRPR.

Vycházeli jsme z toho, že v počítači ATARI XL/XE máme k dispozici programovatelný obvod pro vstup/výstup (PIA), který se běžně využívá pro připojení křížového ovladače.

Spolupráce počítače ATARI s tiskárnou vybavenou paralelním rozhraním CENTRONICS probíhá v handshakingu cyklu.

Pro provoz jsou potřebné zdrojové programy, jejichž výpis je uveden dále. Programy v BASICu (první pro tisk s tiskárnou CENTRONICS, druhý s typem IRPR) lze umístit na libovolné místo v paměti. Je nutné ovšem po každém stisknutí tlačítka RESET obnovit adresu 795, na kterou se ukládá počátek tabulky podprogramů pro tisk.

Na závěr je třeba upozornit, že výrobcem udávaná zatíženost obvodu PIA je jeden vstup TTL. To znamená, že si musíme ověřit, zda tiskárna obvod příliš nepřetíží. Bylo zjištěno, že například u tiskáren typu PRT-80 některé vstupy (např. 2 a 8) měly větší proudový odběr, než mohl PIA poskytnout, zatímco většina pracovala bez problémů.

```
10 REM *****
20 REM **      Rozhraní CENTRONICS      **
30 REM ** © 1987 M. Oškera a S. Pavlíček      **
40 REM *****
50 CLS : INPUT "Kam se mam uložit",A
60 REM Tabulka zarizeni
70 DPOKE 795,A
80 DPOKE A,A+14
90 DPOKE A+2,A+69
100 DPOKE A+4,A+91
110 DPOKE A+6,A+40
120 DPOKE A+8,A+91
130 DPOKE A+10,A+91
140 POKE A+12,75
```

```

150 DPOKE A+13,A+92
160 REM Data Robotron 6313
170 FOR I=A+15 TO A+92
180 READ X
190 POKE I,X
200 NEXT I
210 REM OPEN
220 DATA 173,1,208,208,18,169,56,141,2,211,169,255,141,0
230 DATA 211,169,60,141,2,211,160,1,96,160,138,96
240 REM WRITE
250 DATA 201,155,208,2,169,10,9,128,174,17,208,208,251,141
260 DATA 0,211,41,127,141,0,211,9,128,141,0,211,160,1,96
270 REM CLOSE
280 DATA 173,17,208,208,251,169,56,141,2,211,169
290 DATA 0,141,0,211,169,60,141,2,211,160,1,96
300 REM SCHEMA ZAPOJENI
310 REM ATARI XL/XE          TISKARNA
320 REM predni joystick
330 REM - 1                  D0
340 REM - 2                  D1
350 REM - 3                  D2
360 REM - 4                  D3
370 REM zadni joystick
380 REM - 1                  D4
390 REM - 2                  D5
400 REM - 3                  D6
410 REM - 4                  STROBE
420 REM - 6                  BUSY
430 REM - 8                  ZEM, D7

```

```

0 REM *****
2 REM **      Rozhrani IRPR      *
4 REM ** © 1987 M. Oskera a S. Pavlicek **
6 REM *****
10 CLS:INPUT "Kam se mam uložit",A
20 REM Tabulka zarizeni
30 DPOKE 795,A
40 DPOKE A,A+14
50 DPOKE A+2,A+75
60 DPOKE A+4,A+90
70 DPOKE A+6,A+40
80 DPOKE A+8,A+90
90 DPOKE A+10,A+90
100 POKE A+12,76
110 DPOKE A+13,A+91
120 REM Data DCD-PRT-80
130 FOR I=A+15 TO A+91
140 READ X
150 POKE I,X
160 NEXT I
199 REM OPEN
200 DATA 173,16,208,208,18,169,56,141,2,211,169,255,141,0
202 DATA 211,169,60,141,2,211,160,1,96,160,139,96
209 REM WRITE
210 DATA 201,155,208,2,169,10,73,255,9,128,141,0,
211,41,127,141,0,211

```

```

212 DATA 9,128,164,11,240,9,174,17,208,240,247,141,
    0,211,160,1,96
219 REM CLOSE
220 DATA 169,56,141,2,211,169,0,141,0,211,169,60,141,2,211,96
230 REM SCHEMA ZAPOJENI
240 REM ATARI XL/XE          TISKARNA
250 REM predni joystick
260 REM - 1                  D0
270 REM - 2                  D1
280 REM - 3                  D2
290 REM - 4                  D3
300 REM - 6                  A0 (RESET)
310 zadni joystick
320 REM - 1                  D4
330 REM - 2                  D5
340 REM - 3                  D6
350 REM - 4                  STROBE
360 REM - 6                  BUSY
370 REM - 8                  ZEM

```

Poznámka: Autoři příspěvku děkují všem koleům, kteří se na vývoji řešení podíleli, zvláště Z. Kylarovi, který je autorem základního programu, ze kterého byly zde publikované programy odvozeny.

Řešení Miloše Bayera (Roztoky u Prahy)

Popisované rozhraní využívá joy-portů a programů zajišťujících komunikaci. Ty jsou vytvořeny v modifikacích pro magnetofon i disketovou jednotku. Nejsou delší než 150 bytů. Lze je použít s libovolnou programovací řečí, neobsazující šestou stránku.

Kazetová verze se ukládá od adresy 1792 (hex 700), disketová verze vyžaduje místo pro DOS, a proto začíná na 7424 (hex 1D00).

Programy jsou napsány v ATARI BASICu. Po opsání si pořídíte kopii (LIST"C", LIST"D"), protože po RUN už nebudete mít příležitost něco opravit. Chybové hlášení ERROR 138 se generuje v případě, že je tiskárna OFF LINE.

V okamžiku, kdy je počítači nařizeno vytisknout blok dat na tiskárnu, CIO vyše prostřednictvím akumulátoru jeden byte. Takto přebírá rutina PUTBYTE (registru A) i znaky další. PUTBYTE nejdříve kontroluje, zda je tiskárna připravena k přijetí dat. Zjistí, zda BUSY (STRIG 0) je ve stavu H (tiskárna tiskne) a určitý čas čeká na stav L. PUTBYTE také kontroluje, zda znak v registru A není EOL (155) a převádí ho na standardní povel ASCII CARRIAGE RETURN (kód 13). Tato instrukce startuje tisk znaků načtených do bufferu tiskárny a zajišťuje AUTO FEED (automatický přechod na nový řádek).

Poslat znak do joy-portů není obtížné. Bity 0 - 3 jsou k dispozici na portu 1, bity 4 - 7 na portu 2. Tiskárna nebere data jako aktuální, dokud nedostane potvrzení STROBE. Protože

máme k dispozici jen osm datových vedení, je potřeba vyřešit, kudy STROBE povede. Jedno z možných řešení spočívá v omezení počtu bitů přenášejících znak na šest a bit 7 se použije pro STROBE, což sice omezuje možnosti, ale je nejjednodušší. Protože tiskárna bit 7 při přenosu očekává, musíme ho trvale připojit na zem (pin 30). Znak je tedy stále definován osmi bity, bit 7 je stále L.

Pro zavedení a spuštění je použit autoboot. Tento způsob umožňuje před tím, než BASIC převezme kontrolu, například nahrát jinou sadu znaků, nastavit barvu pozadí atd.

Použití programů

Do disketové jednotky vložte čistou disketu (bez AUTORUN.SYS). Přepište program v BASICu a zadejte RUN. Ten se v několika sekundách zapíše na disketu jako AUTORUN.SYS. Vždy po zapnutí počítače se bude automaticky nahrávat.

Do magnetofonu XC-12 vložte čistou kazetu, navolte záznam (REC+PLAY). Přepište program a zadejte RUN. Program se asi za 20 sec zaznamená. Zavádění do počítače se provede stisknutím tlačítka START při zapnutí počítače. V okamžiku, kdy se na obrazovce objeví READY, je vše připraveno k tisku.

Rozhraní bylo vyvinuto pro tiskárnu EPSON (podle toho i název komunikačního programu), ale je použitelné pro všechny tiskárny respektující protokol CENTRONICS. Protože zatížitelnost datových vedení je zvětšena hradly AND, lze použít i tiskárny s velmi malou impedancí.

Poznámky ke stavbě rozhraní

CENTRONICS používá pro každé datové vedení zemnicí vodič. Toto uspořádání je vhodné zachovat. Při realizaci lze použít namísto 74LS08 hradlo 7408. Má větší spotřebu - asi 14 mA proti 4 mA. Lept se vejde přesně do krabičky od SORBEXU (lapač pachů do lednic). Napájení je třeba blokovat.

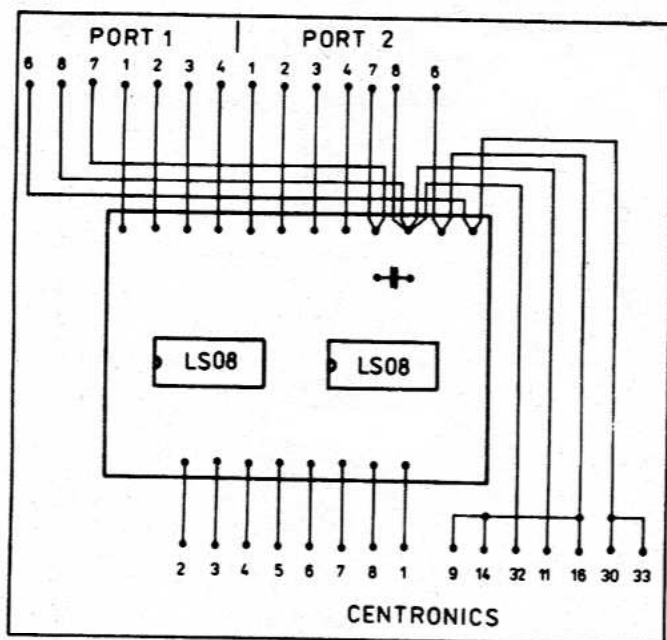
Před zapojením kontrolujte odběr!

Při pečlivé práci bude rozhraní pracovat na první zapojení.

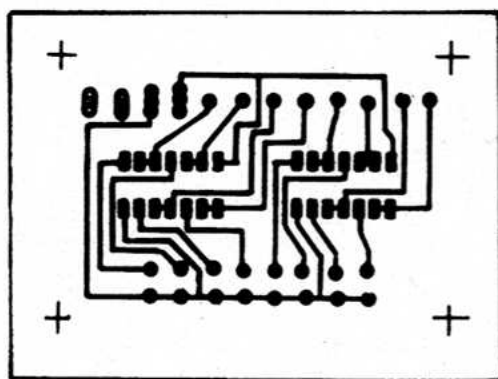
Seznam součástek

- polovodič 74LS08 - 2 ks
- konektor CENTRONICS 36 pin
- konektor D (Joystickový) 9 pin - 2 ks

Při zpracování článku bylo použito literatury <6>, <7> a <8>.



Obr. 5 Rozhraní CENTRONICS



Obr. 6 Tištěný spoj rozhraní CENTRONICS

	pin	funkce	CENTRONICS pin	funkce
JOYSTICK 0	1	BIT 0	2	DATA 1
	2	BIT 1	3	DATA 2
	3	BIT 2	4	DATA 3
	4	BIT 3	5	DATA 4
	5	POT B	-	-
	6	STRIG 0	11	BUSY
	7	+5V	-	-
	8	0 V	16	ZEM
	9	POT A	-	-
JOYSTICK 1	1	BIT 4	6	DATA 5
	2	BIT 5	7	DATA 6
	3	BIT 6	8	DATA 7
	4	BIT 7	1	STROBE
	5	POT B	-	-
	6	STRIG 1	32	FAULT
	7	+5V	-	-
	8	0 V	33	ZEM
	9	POT A	-	-
			9	DATA 8
			30	ZEM

+5V použito pro napájení 74LS08
DATA 8 spojena se zemí

Výpisy komunikačních programů:

Disketová jednotka

```

5 REM EPSON INTERFACE
10 REM vytvoření AUTORUN.SYS - zacina na radce 70
20 OPEN #1,0,"D:AUTORUN.SYS
30 X=0
40 READ D:IF D=-1 THEN END
50 PUT #1,D:GOTO 40
60 REM radka 70 zacina hlavou (typ zarizeni, pocatecni,
koncova
adresa)
70 DATA 255,255,0,29,133,29
80 DATA 165,12,141,14,29,165,13,141,15,29,76,16,29,32,44,
29,169,134,141,231
90 DATA 2,169,29,141,232,2,169,13,133,12,169,29,133,13,
169,45,141,27,3,169
100 DATA 29,141,28,3,96,56,29,74,29,130,29,97,29,130,29,
130,29,162,0,173
110 DATA 16,208,240,6,202,208,248,160,138,96,162,255,76,77,
29,162,0,169,56,141
120 DATA 2,211,142,0,211,169,60,141,2,211,169,128,141,0,
211,160,1,96,174,16
130 DATA 208,208,251,201,155,208,2,169,13,9,128,141,
0,211,32,133,29,41,127,141
140 DATA 0,211,234,9,128,141,0,211,160,1,96,160,146,
96,224,2,225,2,0,29,-1

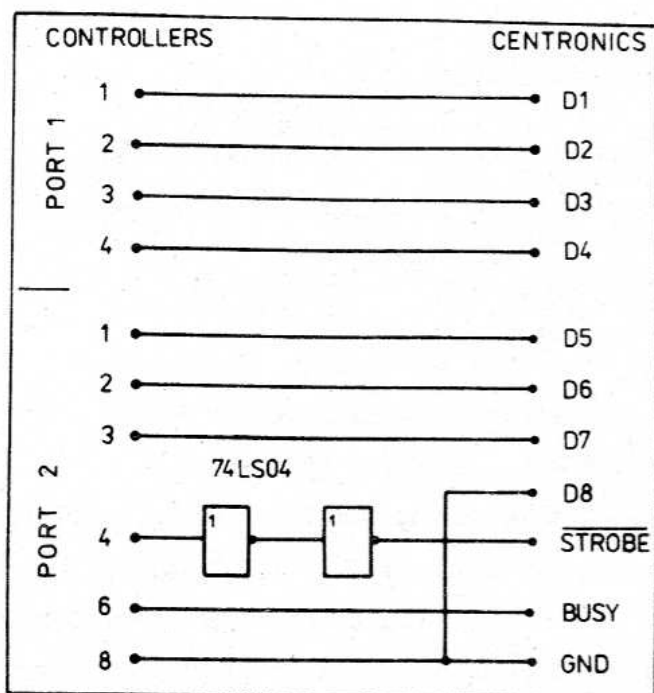
```

Magnetofon

```
10 REM EPSON INTERFACE
20 BFR=(PEEK(106)-24)*256
30 FOR X=0 TO 46:READ D:POKE 1536+X,D:NEXT X
40 REM řádky 50 a 60 obsahují autoboot
50 DATA 104,104,133,204,104,133,203,104,133,206,104,133,205,
  162,16,168,11,157,66,3
60 DATA 165,203,157,68,3,165,204,157,69,3,165,205,157,72,3,
  165,206,157,73,3
70 DATA 32,86,228,96,67,58,155
80 REM uložení komunikačního programu do
  dočasného bufferu
90 X=0:RESTORE 180
100 READ D:IF D=-1 THEN 130
110 POKE BFR+X,D:X=X+1:GOTO 100
120 REM nyní se načtená data zaznamenají na msf
130 OPEN #1,8,128,"C:"
140 X=USR(1536,BFR,PEEK(BFR+1)*128)
150 END
180 DATA 0,1,0,7,13,7,169,60,141,2,211,24,96,169,123,141,
  231,2,169,7
190 DATA 141,232,2,169,34,141,27,3,169,7,141,28,3,96,45,7,63,
  7,119,7
200 DATA 86,7,119,7,119,7,162,0,173,16,208,240,6,202,208,
  248,160,138,96,162
210 DATA 255,76,66,7,162,0,169,56,141,2,211,142,0,211,169,
  60,141,2,211,169
220 DATA 128,141,0,211,160,1,96,174,16,208,208,251,201,155,
  208,2,169,13,9,128
230 DATA 141,0,211,32,122,7,41,137,141,0,211,234,9,128,141,
  0,211,160,1,96
240 DATA 160,145,96,-1
```

Řešení Jana, Jiřího ml. a st. Pokorných (Praha)

Na obr. 7 je uvedeno propojení výstupů ovladačů a vstupu do tiskárny (je obdobou zapojení uvedeného v popisu STARTEXTERu). Přenos dat je sedmibitový. Osmý bit (číslování bitů od 1 do 8) se používá pro přenos STROBE impulsu z počítače do tiskárny. Při paralelním rozhraní se otevírá vstup do tiskárny sestupnou hranou STROBE impulsu. Pokud je sestupná hrana delší než asi 0,2 us, je třeba vložit člen, který obnoví strmost hrany (na obr.7 je to obvod 74S04).



Obr. 7 Propojení výstupů ovladačů a vstupu tiskárny

Osmý bit vstupu tiskárny je zapojen na log. 0. Je možné rovněž zapojení na log.1; v tomto případě je potřeba drobná úprava v programu HSPI pro ovládání osmého bitu tiskárny. Signál BUSY z tiskárny se přivádí na adresu \$D011 k dalšímu využití.

Uložením čísla \$38 na adresu \$D302 a \$FF na adresu \$D300 v uvedeném pořadí je naprogramován výstup z počítače. Dalším uložením \$3C na adresu \$D302 je počítač připraven k vysílání dat ukládaných do \$D300. Nastavení úrovně pro STROBE impulsy se provede uložením \$80 do \$300.

Vysílání požadovaných dat se provádí po jednotlivých bytech (DATA). Předpokládáme, že $DATA < \$80$. Používáme tři příkazy:

1. do \$D300 se uloží DATA + \$80
2. do \$D300 se uloží DATA
3. do \$D300 se uloží DATA + \$80

Celý sled příkazů je uveden v tabulce.

Nastavení výstupu

```
POKE 54018,56: REM Řídící logika
POKE 54016,255: REM Nastavení výstupu
POKE 54018,60: REM V $D300 jsou data
POKE 54016,128: REM Úroveň pro STROBE
```

Vysílání dat

```
POKE 54016,DATA+128: REM DATA < $80
POKE 54016,DATA: REM bit 8 - log.0
POKE 54016,DATA+128: REM bit 8 - log.1
```

Program HSP1 pro řízení výstupu

Program HSP1 se ukládá v RAM od \$600 do \$6D4 a umožňuje využití standardních příkazů pro tisk a to LPRINT, LIST "P:", PUT #, PRINT # (včetně OPEN #, CLOSE #, STATUS #). Je rovněž možné využít tisk přímým vstupem do CIO části operačního systému počítače. Při inicializaci programu HSP1 se přepisuje v tabulce zařízení adresa tabulky vektorů Handleru tiskárny (na adrese \$31B a \$31C) a přes novou tabulku vektorů se přechází k řízení výstupu na tiskárnu.

Při inicializaci programu HSP1 je nastaven normální výstup pro psaní textů. Při normálním výstupu je aktivní funkce EOL a není aktivní nastavování osmého bitu v tiskárně. V tomto případě je kód EOL (\$9B) vysílaný ve sledu dat přeložen na kód ukončující tisk řádky, přesunující tiskací hlavu k levému okraji a posunující papír.

Funkci EOL je možné zrušit uložením #1 na adresu PCX1=\$66, což je potřeba pro grafický výstup. Nastavení aktivace osmého bitu se provede uložením #1 na adresu PCX2 = \$CD. Při nastavené aktivaci osmého bitu vysílá program HSP1 před odesláním čísla většího než \$79 příkaz k obnově osmého bitu na tiskárně. Pro příkazy PUT # a PRINT # je vhodné používat kanály 1 až 6 včetně.

Funkce programu je vázána na přepis adresy tabulky vektorů Handleru tiskárny v tabulce zařízení. Pokud je při ukládání jiných programů do RAM počítače nebo při RESETu (vždy) tato adresa přepsána, je potřeba provést znovu inicializaci.

Vzhledem k jednoduchosti, nenáročnosti a možnosti realizace bez zvláštního vybavení běžnými prostředky může být uvedený systém paralelního výstupu široce využíván.

```

104 169 48 141 27 3 169 6 141 28 3 169 56 141 2
211 169 255 141 0 211 169 60 141 2 211 169 128 141 0
211 169 0 133 204 133 205 96 0 0 0 0 0 0
0 0 0 159 6 62 6 90 6 93 6 159 6 90 6
76 37 6 165 204 208 37 165 203 201 10 240 15 224 112
208 27 189 66 3 201 9 240 20 201 11 240 16 160 1
96 160 146 96 133 203 164 204 208 10 201 155 208 6 169
10 133 203 208 42 164 205 240 38 165 203 16 34 169 27
32 152 6 208 34 169 62 32 152 6 208 27 165 203 32
152 6 208 20 169 27 32 152 6 208 13 169 35 32 152
6 96 32 164 6 164 206 192 1 96 32 187 6 96 32
187 6 192 144 208 1 96 9 128 168 41 127 140 0 211
141 0 211 140 0 211 96 160 0 132 19 132 20 172 17
208 208 5 160 1 132 206 96 164 19 192 7 208 240 164
144 208 243
    
```

Příkazy pro práci s programem HSP1

USR (1536)

- Inicializace programu
- a) uložení adresy nové tabulky vektorů do tabulky zařízení
- b) nastavení výstupu na adrese \$D300
- c) nastavení úrovně log. 1 pro STROBE
- d) nastavení aktivní funkce EOL
- e) vypnutí aktivace osmého bitu tiskárny

POKE 204,X

- X = 0 - funkce EOL je aktivní
- X = 1 - zrušení funkce EOL

POKE 205,Y

- Y = 0 - vypnutí aktivace osmého bitu
 - Y = 1 - zapnutí aktivace osmého bitu
- * * *

Další velice dobré řešení autora ing. Václava Fajty využívající rovněž rozhraní CENTRONICS jsme uvedli v ZAK č.6/87 (s.46-49) a 1/88 (s.55-63) v článku nazvaném "CENTRONICS PRINTER INTERFACE".

* * *

Ke stejnému tématu byl v ZAK č.6/87 (s.42-45) otištěn článek "Interfejs ke spojení tiskárny CONSUL 2111-03 s počítačem ATARI 800 XL (130 XE)" autora F. Vondruše.

* * *

5.1.2. Sériové rozhraní pro tiskárnu

Řešení Karla Bílka (Praha)

Tento článek by měl napomoci při připojení tiskárny ROBOTRON k počítači ATARI XL/XE. Předem je nutno upozornit, že využití ATARI při tisku není úplné. Využití povelů tiskárny je sice maximální, ale rychlost tisku je pouze 50 zn/s. Tiskárny ROBOTRON 6311 a 6312 však mohou tisknout rychlostí 100 zn/s. Možnost tisku plnou rychlostí existuje, ale pak nelze použít zobrazení na displeji.

Základní parametry:

- připojení je uskutečněno přes paralelní port (pro křížový ovladač),
- komunikační vedení je napětově upraveno,
- operační systém ATARI je přepsán z ROM do RAM,
- je zaveden program umožňující komunikaci.

Připojení

Tiskárna ROBOTRON používá dva typy rozhraní, a to paralelní CENTRONICS a seriový V.24 (RS 232C). Naše řešení využívá seriového spojení tiskárny a počítače ATARI.

Tiskárna ROBOTRON pracuje pod třemi druhy protokolů - DTR, XON/XOFF, ETX/ACK.

a) DTR protokol/Hardware protokol

Výměna údajů se výlučně řídí používaným vedením (není vysílán ani přijímán žádný řídicí znak). Tiskárna pracuje pouze v režimu příjmu. Vedení 108 je ve stavu ZAP, tiskárna je připravena k příjmu dat po vedení 104. Pokud se do vyrovnávací paměti může přijmout již jenom 10 znaků, uvede se vedení 108 do stavu VYP. Po vyprázdnění paměti zapne tiskárna vedení 108 opět do stavu ZAP. V době, kdy je vedení 108 ve stavu VYP, nesmí být vysílána žádná data po vedení 104.

b) XON/XOFF protokol

Tiskárnou vysílaný znak XON (\$11) indikuje pohotovost k příjmu dat. Pokud je možné přijmout již jenom 10 znaků, vyšle tiskárna jeden XOFF (\$13). Tok dat se musí zastavit do okamžiku, než tiskárna vyšle opět XON a XOFF jsou vysílány po vedení 103, data přicházejí po vedení 104.

c) ETX/ACK protokol

Přenos dat je prováděn v blocích. Řídicí znak ETX (\$03) je na konci bloku dat. Po úplném příjmu a provedení (vytištění) je vysílán tiskárnou znak ACK (\$06). ACK se vysílá rovněž po zapnutí tiskárny. Je-li délka bloku dat větší než schopnost příjmu dat, je vystaveno vedení 108 (viz protokol DTR). ACK je vysíláno po vedení 103, ETX a data jsou přijímány po vedení 104.

Již na první pohled je z popsaného patrné, že přímé připojení není možné. Cesta řešení vede přes využití sériového portu. Sériový port ATARI ale vyžaduje zařízení s vlastní inteligencí. To znamená, že na jedno vedení "sériový BUS" je paralelně připojeno několik zařízení. Při komunikaci je zařízení jednotlivě osloveno a je uzavřen komunikační most. Ostatní zařízení pak na přenášené údaje nereagují. Toto nelze mezi tiskárnou ROBOTRON a počítačem ATARI programově zajistit. Lze ale odpojit všechna další zařízení a pak programově zajistit jeden z popsaných protokolů. Při nespolehlivosti sériového konektoru a nutném stálém přepínání periférií se nedá mluvit o pohodlné a produktivní práci.

Jako poslední možnost je navržení rozhraní, které by umožnilo uzavření komunikačního mostu. Toto řešení by však vyhovovalo pouze profesionálnímu pracovišti, neboť svou složitostí a pořizovacími náklady amatéra odradí.

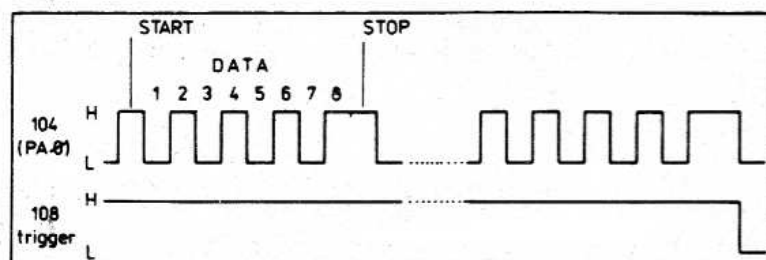
Naproti tomu paralelní port je programově velmi dobře zvládnutelný a pro amatérské nasazení tiskárny výhodný.

Tiskárna je vedením 104 připojena na vývod portu PA0 přes kompenzační obvod. Vedení 108 je opět přes kompenzační obvod připojeno na trisser 1. Na tiskárně je nastaveno:

- protokol DTR,
- rychlost přenosu 9600 baudů,
- 8 údajových bitů,
- bez zkoušky parity,
- 1 stopbit.

(Nastavení je nutno provést podle manuálu konkrétní tiskárny.)

Na obr. 8 je znázorněn přenos 2 bytů (\$55), druhý byte je poslední, který může tiskárna přijmout, a proto na vedení 108 dojde ke změně stavu.



Obr. 8 Schéma přenosu 2 bytů.

Komunikační vedení

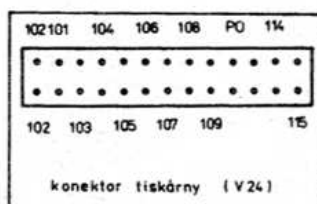
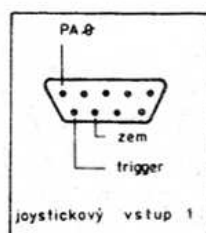
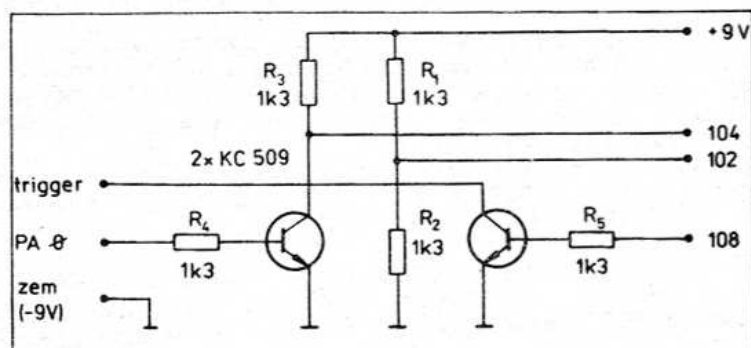
Pozor! Pouhé spojení vedení 104 - PA0 a vedení 108 - trisser není možné, protože napětové úrovně se liší. Na straně počítače je logická 0 = 0,8 V a logická 1 = 2,4 V. Tiskárna má na vedení napětí podle tabulky 1.

Tabulka 1

stav	napětová úroveň
znak	-3 V až -12 V
mezera	+3 V až +12 V
řídící vedení	
zapnuto	-3 V až -12 V
vypnuto	+3 V až +12 V

Je proto nutné mezi vedení vložit kompenzační obvod podle obr. 9. Celé zapojení je velmi jednoduché, proto jen stručný popis:

Rezistory R_1 a R_2 vytvářejí umělý střed napětí $\pm 4,5$ V, které je nutné pro činnost vedení 104. Tranzistor T1 s ochranným rezistorem báze R_4 a kolektorovým rezistorem R_3 převádí logické úrovně vstupu PA0 na logiku $\pm 4,5$ V. Tranzistor T2 pak naopak $\pm$ logiku na TTL logiku. Jako zdroj 9 V jsem použil napáječ od kapesního kalkulátoru, ale postačí i dvě ploché baterie.



Obr. 9 Schéma zapojení kompenzačního obvodu

Nejen vlastní připojení činí potíže. Také vytvoření softwaru není jednoduché. Prvním problémem je, kam s ním. Adresy \$0600 - \$9FFF mohou být obsazeny programem používajícím tiskárnu. Proto byla zvolena RAM ukrytá pod operačním systémem. Nejdříve tedy musí dojít k přepsání ROM do RAM. Poté se zavede obslužný program tiskárny do míst, kde jsou data, která se používají jen při startu počítače, takže tato část je volná pro program. V okamžiku, kdy stiskneme RESET, je paralelní port (PIA) opět nastaven do základního stavu. To znamená, že na adrese \$D301 je \$FD. Tím jsou data při restartu čtena opět z ROM. Návrat po restartu opět do RAM jsem nezajišťoval, a to z důvodu nutnosti programu umístěného tentokrát v hlavní části paměti a tím překážejícího. Ovšem používám-li pro tisk program, zajišťuji návrat přímo v něm. V Basicu se vracím méně pohodlně - POKE 54017,252.

Na tomto místě se ještě zmíním o úpravě programů a jejich nahrání. Podle hlavičky programu nejdříve najdeme část, kde program startuje. Úprava pak spočívá ve vyhledání šikovného místa, kde můžeme vyměnit instrukci za JSR do vhodného místa paměti (třeba za program). Tam nejdříve provedeme výměnou instrukci, pokud to nebyl JSR. Ten nahradíme naopak až na konci našeho přidaného programu, a to JMP na adresu, kam JSR směřoval. Pro další výklad předpokládám, že je provedena přepsaná instrukce. Prvá instrukce v našem programu je PHA, tím se nemusíme starat o to, že by v akumulátoru byla platná data. Dále následuje LDA#\$FC a STA\$D301, PLA, RTS, nebo již zmíněný JMP. To je vše.

U programů, které mají ošetřený Reset stačí tato úprava též pro nové přepnutí z ROM na RAM, neboť úvodní části programů se většinou po RESET opakuji. Je jisté, že úpravy programů vyžadují znalost strojového jazyka. V Basicu je věc jednoznačná. Zde stačí zmíněný POKE.

Upozorňuji však, že úprava programů jako je "Čapek" je věc složitá, mnohonásobně složitější, než samotné připojení tiskárny. Psaní čárek a háčků vyžaduje práci v módu jednotlivé jehly.

Naopak zavedení dalšího programu do počítače (po nahrání obslužné rutiny) je věc jednoduchá. Dáme BYE (B.), stiskneme RESET, přidržíme START nebo START + OPTION a zavedeme žádaný program

Obslužný program

Samotný obslužný program se skládá ze dvou částí. Prvá část je univerzální zaváděcí program, který přesune ROM do RAM, dále otevře kanál magnetofonu a zajistí nahrání druhé části. Tato druhá část jsou prozatím data. Před daty je umístěna nestandardní hlavička, podle které se data přesouvají v paměti. Tvar hlavičky je uveden v tabulce 2. Jak je z tabulky patrné, je možné přesouvat různý počet bloků, každý se samostatnou adresou a délkou. Vše je ukončeno přechodem do Basicu.

Tabulka 2

pořadí bytů	význam
jeden byte	počet bloků
dolní byte	adresa umístění posledního bloku (pořadí na kazetě)
horní byte	
dolní byte	počet bytů posledního bloku
horní byte	
.	následují údaje dalších bloků
.	
.	odzadu dopředu (pořadí na kazetě)
.	
dolní byte	adresa umístění prvního bloku (pořadí na kazetě)
horní byte	
dolní byte	počet bytů prvního bloku
horní byte	

Zbývá popsat samotný program. Neočekávejte, prosím, zcela vyčerpávající popis po jednotlivých instrukcích. Je to naprosto zbytečné. Pro toho, kdo se alespoň částečně orientuje v operačním systému ATARI, je věc triviální záležitostí. Pro vás ostatní pak doporučuji navštěvovat přednášky RNDr. Jiřího Boka, CSc, kde se výbornou formou seznámíte se strojovým jazykem mikroprocesoru 6502 a dozvíte se mnoho dalších zajímavých věcí.

Další výklad bude spíše obhajobou určitých částí mého programu. Chci zdůraznit, že nepřepisují nové zařízení Printer do tabulky HATABS (\$31A - \$33D), nýbrž přepisují vektory přímo na \$E430. Důvodem je tlačítko RESET. Takto nepotřebuji vždy HATABS obnovovat a tím mnohonásobně zjednodušuji připojení k jiným programům! OPEN pak nastaví port resp. bránu PA na výstup a přeíše ukazovátko interruptu VVBLKD (\$224, \$225), a to jeho horní byte. Víím, že to není nejčistější, ale šetřil jsem místem. Dnes též víím, že to nebylo zcela nutné. Místa je tam víc, než jsem očekával. Nyní bych celou záležitost svěřil rutině na \$E45C a to tak, aby rutina sériového výstupu (viz dále) nepřecházela na \$C28A, ale na skutečný vektor, který byl

v VVBLKD.

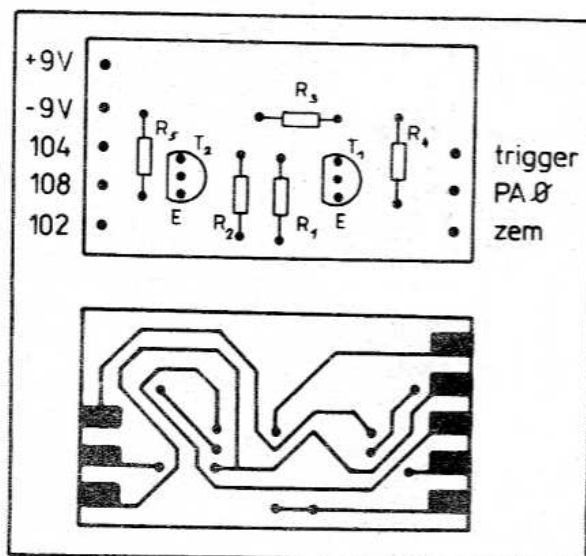
Důvodem je třeba program "DEBUGER +", který tento vektor také používá a tím dochází ke zhroucení. Není jisté, zda by se program tiskárny a jiný program vždy podělily o 20000 strojových taktů a zda by tiskárna nepřekážela v čase. Tiskárně samotné čas nevadí. Nesmí ovšem dojít k jinému přerušení během sériového výstupu. To je také důvod, proč je vysíláno 50 zn/s. Přerušení sice nemusíme použít (a tím tisknout plnou rychlostí operačního systému), ale pak je nutné zakázat zobrazení na displeji. Jinak nám VBLANK přeruší sériový výstup. Takže buď 50 zn/s a zobrazení, nebo plná rychlost a žádné zobrazení. Je zde ještě možnost vyslat během přerušení ne jeden byte, ale dva nebo i více bytů, tím se však program dost komplikuje. Tiskárna ATARI potřebuje na ukončení řádku pouze jeden znak EOL (\$9B), ROBOTRON znaky dva. Podle mého názoru je pro domácí nasazení tiskárny 50 zn/s více než dostačující.

Rychlost tiskárny je 100 zn/s, přesto v určitých pasážích nestačí a vystavuje vedení 108.

OPEN dále zjistí, zda je nastavena adresa \$2DF (počet znaků na řádek: POKE 735, počet znaků nebo OPEN #1,8, počet zn., "P:"). Jestliže ne, nastaví se automaticky 60 znaků na řádek.

PUT nejprve provádí porovnání adres \$2DE - \$2DF a akumulátoru (EOL), čímž vyhledá konec řádku. Není-li konec, vyšle obsah akumulátoru. Je-li konec, vyšle znak "další řádek" LF (\$0A) a "návrat vozíku" CR (\$0D). Samozřejmě je stále testován trigger 1, není-li plná vyrovnávací paměť tiskárny. K vysílání znaku dochází tak, že po potvrzení platnosti akumulátoru se přechází do podmíněné smyčky, kde PUT počká na přerušení. Po jeho průchodu je znak odeslán a PUT může vystoupit ze smyčky. Do Y registru uvede status operace (\$01) a vrátí řízení operačnímu systému. Samotné přerušení programu zase nejdříve zjistí platnost znaku. Je-li platný, je sériově vyslán rychlostí 9600 Bd. Dále je provedeno potvrzení vyslání a skok na \$C28A. Není-li platný znak, je proveden rovnou skok na \$C28A.

CLOSE vrací původní hodnotu na port a do VVBLKD.



Obr. 10 Plošný spoj kompenzačního členu
(určen do krabičky od klenotů)

Použitá literatura: <1>, <10>

Univerzální zaváděcí program

```

0604    A9 00      LDA    #$00
0606    AA        TAX
0607    A8        TAY
0608    EA        NOP
0609    18        CLC
060A    60        RTS
060B    43 ,C,    ???
060C    78        SEI
060D    AB ,+,    ???
060E    00        BRK
060F    8D 0E D4  STA    $D40E
0612    A9 FE      LDA    #$FE
0614    8D 01 D3  STA    $D301
0617    A9 D8      LDA    #$D8
0619    85 CC      STA    $CC
061B    84 CB      STY    $CB
061D    EE 01 D3  INC    $D301
0620    B1 CB      LDA    ($CB).Y
0622    CE 01 D3  DEC    $D301
0625    91 CB      STA    ($CB).Y
0627    C8        INY
0628    D0 F3     BNE    $061D
062A    E6 CC      INC    $CC
062C    E4 CC      CPX    $CC

```

062E	D0 ED	BNE	\$061D
0630	8A	TXA	
0631	D0 06	BNE	\$0639
0633	A2 CF	LDX	#\$CF
0635	A9 C0	LDA	#\$C0
0637	D0 E0	BNE	\$0619
0639	A9 40	LDA	#\$40
063B	8D 0E D4	STA	\$D40E
063E	58	CLI	
063F	A2 10	LDX	#\$10
0641	A9 03	LDA	#\$03
0643	9D 42 03	STA	\$342,X
0646	A9 07	LDA	#\$07
0648	9D 4A 03	STA	\$34A,X
064B	A9 80	LDA	#\$80
064D	9D 48 03	STA	\$0348,X
0650	A9 08	LDA	#\$08
0652	9D 44 03	STA	\$0344,X
0655	A9 06	LDA	#\$06
0657	9D 45 03	STA	\$345,X
065A	20 56 E4	JSR	\$E456
065D	A9 07	LDA	#\$07
065F	9D 42 03	STA	\$342,X
0662	A9 00	LDA	#\$00
0664	9D 44 03	STA	\$0344,X
0667	A9 07	LDA	#\$07
0669	9D 45 03	STA	\$0345,X
066C	A9 FF	LDA	#\$FF
066E	9D 48 03	STA	\$0348,X
0671	9D 49 03	STA	\$0349,X
0674	20 56 E4	JSR	\$E456
0677	A9 0D	LDA	#\$0D
0679	9D 42 03	STA	\$0342,X
067C	20 56 E4	JSR	\$E456
067F	A9 FE	LDA	#\$FE
0681	8D 01 D3	STA	\$D301
0684	AD 00 07	LDA	\$0700
0687	85 CB	STA	\$CB
0689	0A	ASL	A
068A	0A	ASL	A
068B	85 CC	STA	\$CC
068D	A9 07	LDA	#\$07
068F	85 01	STA	\$01
0691	A9 01	LDA	#\$01
0693	18	CLC	
0694	65 CC	ADC	\$CC
0696	90 02	BCC	\$069A
0698	E6 01	INC	\$01
069A	85 00	STA	\$00
069C	A2 04	LDX	#\$04
069E	A4 CC	LDY	\$CC
06A0	B9 01 07	LDA	\$0701,Y
06A3	95 CD	STA	\$CD,X
06A5	88	DEY	
06A6	CA	DEX	
06A7	10 F7	BPL	\$06A8
06A9	C8	INY	

06AA	84 CC	STY	\$CC
06AC	A0 00	LDY	#\$00
06AE	B1 00	LDA	(\$00).Y
06B0	91 CD	STA	(\$CD).Y
06B2	E6 00	INC	\$00
06B4	D0 02	BNE	\$5B8
06B6	E6 01	INC	\$01
06B8	E6 CD	INC	\$CD
06BA	D0 02	BNE	\$5BE
06BC	E6 CE	INC	\$CE
06BE	C6 CF	DEC	\$CF
06C0	D0 EC	BNE	\$06AE
06C2	A5 D0	LDA	\$D0
06C4	F0 04	BEQ	\$06CA
06C6	C6 D0	DEC	\$D0
06C8	D0 E4	BNE	\$06AE
06CA	C6 CB	DEC	\$CB
06CC	D0 CE	BNE	\$069C
06CE	4C 77 E4	JMP	\$E477
06D1			

Rutina tiskárny

OPEN

E49F	A5 2A	LDA	\$2A
E4A1	C9 08	CMP	#\$08
E4A3	F0 03	BEQ	\$E4A8
E4A5	A0 83	LDY	#\$83
E4A7	60	RTS	
E4A8	A5 2B	LDA	\$2B
E4AA	D0 07	BNE	\$E4B3
E4AC	AD DF 02	LDA	\$02DF
E4AF	D0 05	BNE	\$E4B6
E4B1	A9 41	LDA	#\$41
E4B3	8D DF 02	STA	\$02DF
E4B6	A9 CB	LDA	#\$CB
E4B8	8D 25 02	STA	\$225
E4BB	4C 65 CB	JMP	\$CB65
CB65	A9 0B	LDA	#\$0B
CB67	8D 02 D3	STA	\$D302
CB6A	A0 01	LDY	#\$01
CB6C	60	RTS	

CLOSE

CB6D	A9 3C	LDA	#\$3C
CB6F	8D 02 D3	STA	\$D302
CB72	A0 01	LDY	#\$01
CB74	60	RTS	

PUT

CB75	48		PHA	
CB76	A6	11	LDX	\$11
CB78	D0	09	BNE	\$CB83
CB7A	68		PLA	
CB7B	A2	3C	LDX	##3C
CB7D	8E	02 D3	STX	\$D302
CB80	A0	80	LDY	##80
CB82	60		RTS	
CB83	AE	10 D0	LDX	\$D010
CB86	4C	8D CB	JMP	\$CB8D
CB89	EA		NOP	
CB8A	4C	CC CB	JMP	\$CBCC
CB8D	EA		NOP	
CB8E	D0	E6	BNE	\$CB76
CB90	C9	98	CMP	##98
CB92	D0	05	BNE	\$CB99
CB94	A2	00	LDX	##00
CB96	8E	DE 02	STX	\$02DE
CB99	AE	DE 02	LDX	\$02DE
CB9C	D0	13	BNE	\$CBB1
CB9E	20	BC CB	JSR	\$CBBC
CBA1	AE	DF 02	LDX	\$02DF
CBA4	8E	DE 02	STX	\$02DE
CBA7	A9	0A	LDA	##0A
CBA9	20	BC CB	JSR	\$CBBC
CBAE	A9	0D	LDA	##0D
CBAE	4C	B4 CB	JMP	\$CBB4
CBB1	CE	DE 02	DEC	\$02DE
CBB4	EA		NOP	
CBB5	20	BC CB	JSR	\$CBBC
CBB8	A0	01	LDY	##01
CBBA	68		PLA	
CBBB	60		RTS	
CBBC	85	2F	STA	\$2F
CBBE	A9	01	LDA	##01
CBC0	8D	08 CC	STA	\$CC08
CBC3	AD	08 CC	LDA	\$CC08
CBC6	D0	FB	BNE	\$CBC3
CBC8	60		RTS	
CBC9	38		SEC	
CBCA	B0	0B	BCS	\$CBD7
CBCC	48		PHA	
CBCD	AD	08 CC	LDA	\$CC08
CBD0	F0	27	BEQ	\$CBF9
CBD2	A5	2F	LDA	\$2F
CBD4	A0	09	LDY	##09
CBD6	18		CLC	
CBD7	A2	01	LDX	##01
CBD9	90	02	BCC	\$CBDD
CBD8	A2	00	LDX	##00
CBD0	8E	00 D3	STX	\$D300
CBE0	A2	10	LDX	##10
CBE2	8E	07 CC	STX	\$CC07
CBE5	CE	07 CC	DEC	\$CC07

CBE8	D0 FB	BNE	\$CBE5
CBEA	C0 00	CPY	#\$00
CBEC	F0 06	BEQ	\$CBF4
CBEE	4A	LSR	A
CBEF	88	DEY	
CBF0	D0 E5	BNE	\$CBD7
CBF2	F0 D5	BEQ	\$CBC9
CBF4	A9 00	LDA	#\$00
CBF6	8D 08 CC	STA	\$CC08
CBF9	68	PLA	
CBFA	4C 8A C2	JMP	\$C28A

5.1.3. Připojení dálnopisu k počítači ATARI XL/XE

Ve Zpravodaji AK č.5/87 (s.43-47) jsme publikovali řešení ing. Stanislava Štulpy v článku "Připojení dálnopisu SIEMENS 100 k počítači ATARI 800 XL". Uvedené řešení vyžaduje tzv. dálnopisný přídatek pro TV přijímač. Zde přinášíme další dvě řešení na připojení dálnopisu k počítači ATARI XL/XE.

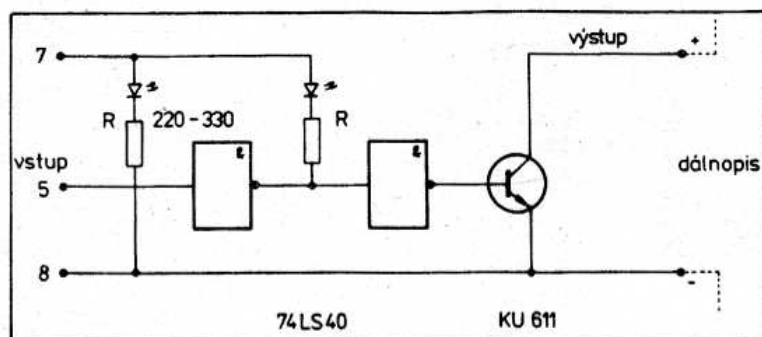
* * *

Řešení Daniela Prymucha (Hořovice)

Rozhraní dálnopisu (obr.11) je připojeno na konektor pro joystick 2 a využívá programovatelnost obvodu 6520. Obvod má sedmý bit naprogramován jako výstupní a přes dvě oddělovací hradla ovládá tranzistor KU 611, který spíná přímo dálnopisné relé.

Můžeme použít jakýkoliv tranzistor NPN s dostatečným ztrátovým výkonem (KD 601, KU 612 apod.) a přibližně jakékoliv hradlo. V popisovaném řešení bylo použito 10 SM 7440 LŠ. Hradla slouží jako oddělovací člen mezi počítačem a dálnopisem. V žádném případě nedoporučuji použití tranzistoru PNP. Jinak konstrukce nemá žádné zásluhy (při své jednoduchosti je ani mít nemůže).

Při ožiování doporučuji vyzkoušet rozhraní nejprve s počítačem. Zde se nám osvědčí obě dvě diody LED (jedna signalizuje připojení napájecího napětí a druhá znázorňuje vstupní signál), a potom jen s dálnopisem, kdy rozhraní připojíme na baterii a při přivádění signálu (+,-) musí relé spínat. Pokud vše funguje, připojíme dálnopis přes rozhraní k počítači.



Obr. 11 Rozhraní dálnopisu

Návod k připojení dálnopisu

Po odklopení vrchního krytu dálnopisu najdeme hlavní svorkovnici. Nyní najdeme vodiče, které vedou proud pro ovládání relé. Na svorkovnici jsou vedle sebe a jsou propojeny kusem vodiče. Tento vodič vyjmeme a místo něj zapojíme výstup rozhraní, kolektor na kladnější stranu (rozdílové napětí činí asi 62 V).

Pro tisk používám programy DAL.MCU (univerzální). Lze jej použít u všech programů, např. SPEED SCRIPT, PASCAL, EDIT, ASSEMBLER, TURBO, BASIC apod. Program DAL.MCS (speciální) lze použít jen pro některé programy, např. SPEED SCRIPT, DEBUG MONITOR atd.

Oba programy byly naprogramovány v assembleru, přeloženy a upraveny do systému TURBO 2000.

Po nahrání programy DAL, znovu spustí program TURBO 2001 a my najdeme program, při kterém chceme tisknout.

Během hledání programu můžeme používat tlačítko RESET. Program se vždy znovu inicializuje a spustí TURBO 2001. Pro zavedení nelze použít program TURBO 2000 vzhledem k jeho uložení v paměti - je nutno použít program TURBO 2001. Při použití programu TURBO 2000 se systém zhroutí ...

Tisk na dálnopis se provádí stejně jako tisk na tiskárnu.

Tabulka transkripce

DAL.MCU

DAL.MCS

```

0 0123456789abcdef
1
2 . 's/ s'()x+,-./
3 0123456789:: = ?
4 abcdefghijklmno
5 pqrstuvwxyz[/] -
6 abcdefghijklmno
7 pqrstuvwxyz

```

```

0123456789abcdef
. 's/ s'()x+,-./
0123456789:: = ?
abcdefghijklmno
pqrstuvwxyz[/] -
abcdefghijklmno
pqrstuvwxyz

```

Znaky od 80h do ffh jsou tištěny jako normální znaky s výjimkou znaku E01 (9bh), který je převeden na dva znaky bs a lf, které způsobí přechod na začátek řádku a posun papíru o jednu řádku.

Obslužné programy

```

dal.mcs
10 : program pro ovladani dalnopisu.
20 : dalnopis ma kod p .
30 : ovlada se stejne jako tiskarna.
40 : napr. lprint, list 'p:'
50 x=s600
60 : tabulka utilit
70 .word open -1
80 .word close -1
90 .word get -1
0100 .word put -1
0110 .word status -1
0120 .word spec -1
0130 jsr inita
0140 : utility
0150 get
0160 spec
0170 status ldy #146
0180 inita rts
0190 :
0200 hi = 128
0210 lo = 0
0220 :
0230 open
0240 vstup lda #s38
0250 ldx #128
0260 sta sd302
0270 stx sd300
0280 lda #s3c
0290 sta sd302
0300 lda #hi
0310 sta sd300
0320 ldy #0
0330 jsr tisk

```

```

0340 close ldy #1
0350 rts
0360 :
0370 put cmp #155
0380 bne moc
0390 eol lda #67
0400 sta pocet
0410 ldy #1
0420 jsr tisk
0430 ldy #2
0440 jsr tisk
0450 rts
0460 moc dec pocet
0470 bne pok1
0480 pha
0490 jsr eol
0500 pla
0510 pok1 asl a
0520 lsr a
0530 cmp #36
0540 bne dol
0550 ldy #23
0560 jsr tisk
0570 rts
0580 dol cmp #42
0590 bne krat
0600 ldy #28
0610 jsr tisk
0620 rts
0630 krat cmp #123
0640 bcc pok2
0650 mez ldy #3
0660 jsr tisk
0670 rts
0680 pok2 cmp #97
0690 bcc pok3
0700 sbc #32
0710 pok3 cmp #65
0720 bcc cis
0730 cmp #91
0740 bcc pis
0750 sbc #26
0753 cis cmp #33
0754 bcc mez
0760 sta uk11
0770 ldy #4
0780 jsr tisk
0790 lda uk11
0800 sec
0810 sbc #2
0820 tay
0830 jsr tisk
0840 ldy #0
0850 jsr tisk
0860 rts
0870 pis sec
0880 sbc #60

```

```

0890  tay
0900  jsr tisk
0910  rts
0940  : tab. kodu znaku
0950  cci .byte 254,208,196,200,246
0960  : kody pismen
0970      .byte 198,242,220,210,194,218
0980      .byte 244,232,204,214,222,228
0990      .byte 248,216,240,236,238,212
1000      .byte 202,224,206,252,230,250
1010      .byte 234,226
1020  : kody cislic a pomoc. znaku
1030      .byte 248,202,210,0,250,214
1040      .byte 202,222,228,0,226,216
1050      .byte 198,248,250,236,238,230
1060      .byte 194,212,224,234,206,204
1070      .byte 240,220,220,200,252,200
1080      .byte 242,200
1090  : 6 znaku za pismeny
1100      .byte 218,250,244,200,198,200
1110  : podprogram tisku
1120  tisk lda cci,y
1130      ldy #8
1140  lo1 lsr a
1150      bcc lo2
1160      ldx #h1
1170      stx sd300
1180      jmp delka
1190  lo2 ldx #lo
1200      stx sd300
1210      jmp delka
1220  lo3 dey
1230      bne lo1
1240      ldy #1
1250      rts
1260  : delka signalu
1270  delka sty rot
1280      ldx #19
1290  d2 ldy #248
1300  d1 dey
1310      bne d1
1320      dex
1330      bne d2
1340      ldy rot
1350      jmp lo3
1360  uk11 .byte 0
1370  pocet .byte 67
1375  rot .byte 0
1380  :
1390  : zacleneni programu do systemu
1400  :
1410      lda #s50
1420      sta 743
1430      lda #s7
1440      sta 744
1450      lda #0
1460      sta s31b

```

```

1470 lda #6
1480 sta s31c
1490 jsr vstup
1500 lda 25
1510 sta 12
1520 lda 7
1530 sta 13
1540 jmp 1166

```

dal.mcu

```

10 : program pro ovladani dalnopolisu.
20 : dalnopolis ma kod p .
30 : ovlada se stejne jako tiskarna.
40 : napr. lprint list 'p:'
50 x=s600
60 : tabulka utilit
70 .word open -1
80 .word close -1
90 .word get -1
0100 .word put -1
0110 .word status -1
0120 .word spec -1
0130 jsr inita
0140 : utility
0150 get
0160 spec
0170 status ldy #146
0180 inita rts
0190 :
0200 hi = 128
0210 lo = 0
0220 :
0230 open
0240 vstup lda #s38
0250 ldx #128
0260 sta sd302
0270 stx sd300
0280 lda #s3c
0290 sta sd302
0300 lda #h1
0310 sta sd300
0320 ldy #0
0330 jsr tisk
0340 close ldy #1
0350 rts
0360 :
0370 put cmp #155
0380 bne moc
0390 eol lda #67
0400 sta pocet
0410 ldy #1
0420 jsr tisk
0430 ldy #2
0440 jsr tisk
0450 rts
0460 moc dec pocet

```

```

0470 bne pok1
0480 pha
0490 jsr eol
0500 pla
0510 pok1 asl a
0520 lsr a
0630 cmp #123
0640 bcc pok2
0650 mez ldy #3
0660 jsr tisk
0670 rts
0680 pok2 cmp #97
0690 bcc pok3
0700 sbc #32
0710 pok3 cmp #65
0720 bcc cis
0730 cmp #91
0740 bcc pis
0750 sbc #26
0753 cis cmp #33
0754 bcc mez
0760 pha
0770 ldy #4
0780 jsr tisk
0790 pla
0800 sec
0810 sbc #2
0820 tay
0830 jsr tisk
0840 ldy #0
0850 jsr tisk
0860 rts
0870 pis sec
0880 sbc #60
0890 tay
0900 jsr tisk
0910 rts
0940 : tab. kodu znaku
0950 cci .byte 254,208,196,200,246
0960 : kody pismen
0970 .byte 198,242,220,210,194,218
0980 .byte 244,232,204,214,222,228
0990 .byte 248,216,240,236,238,212
1000 .byte 202,224,206,252,230,250
1010 .byte 234,226
1020 : kody cislic a pomoc. znaku
1030 .byte 248,202,210,214,250,214
1040 .byte 202,222,228,210,226,216
1050 .byte 198,248,250,236,238,230
1060 .byte 194,212,224,234,206,204
1070 .byte 240,220,220,200,252,200
1080 .byte 242,200
1090 : 6 znaku za pismeny
1100 .byte 218,250,244,200,198,200
1110 : podprogram tisku
1120 tisk lda cci,y
1130 ldy #8

```

```

1140 lo1 lsr a
1150 bcc lo2
1160 ldx #h1
1170 stx sd300
1180 jmp delka
1190 lo2 ldx #lo
1200 stx sd300
1210 jmp delka
1220 lo3 dey
1230 bne lo1
1240 ldy #1
1250 rts
1260 : delka signalu
1270 delka sty rot
1280 ldx #19
1290 d2 ldy #248
1300 d1 dey
1310 bne d1
1320 dex
1330 bne d2
1340 ldy rot
1350 jmp lo3
1370 pocet .byte 67
1375 rot .byte 0
1380 :
1390 : zacleneni programu do systemu
1400 :
1410 lda #s20
1420 sta 743
1430 lda #s7
1440 sta 744
1450 lda #0
1460 sta s31b
1470 lda #6
1480 sta s31c
1490 jsr vstup
1500 lda 0
1510 sta 12
1520 lda 7
1530 sta 13
1540 jmp 1166

```

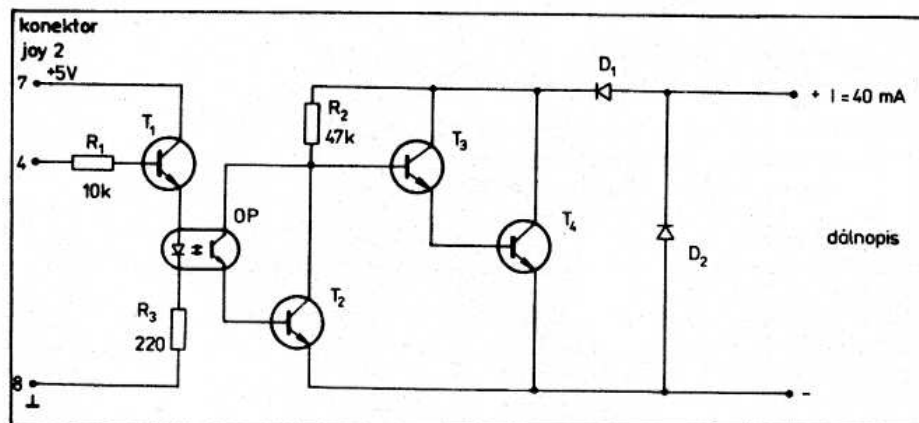
Program pro obsluhu dálnopisu je vygenerován na pásek programem v BASICu. Tento obslužný program se nahrává do počítače po stisknutí klávesy START a zapnutí počítače - PLAY na magnetofonu - RETURN. Po vypsání hlavičky a textu READY provádějí všechny instrukce používající zařízení "P:" výstup na dálnopis.

Dále je možno provést tisk kopie obrazovky (HARDCOPY) stiskem klávesy SELECT. Po dotisknutí obsahu obrazovky je nutno stisknout klávesu RESET.

Kopii obrazovky je dále možno provést napsáním příkazu "DOS". Tento příkaz se může vyskytnout i v programu BASICu. V tomto případě program po dotisknutí obrazovky pokračuje dále v činnosti.

POZOR! Na dálnopisu byly vyměněny typy u několika typových pák. Jednalo se o tyto znaky:

< > i 3 .



Obr. 12 Schéma zapojení dálkopisu

(Poznámka: autorem poskytnuté schéma upravil
ing. Jaroslav Biskup)

Seznam součástí:

T1.....	KC308C
T2.....	KC507
T3.....	KC507
T4.....	KT504
D1,D2.....	KY130/86
R1.....	10k
R2.....	47k
R3.....	220
OB.....	VK 164 12

Poznámka redakce: V dále opsaném výpisu (originál z rukopisu článku nebylo možné použít, protože byl plný rukopisných doplňků) byla snaha zachovat transkripci, v jaké tiskne dálnopis. Některé znaky dálnopisu však nemá textový editor ČAPEK 2.1., ve kterém píšeme toto číslo Zpravodaje AK. Proto je v dále uvedených výpisech použito znaků, které umí ČAPEK a tiskárna ATARI 1029. Pro vysvětlení: namísto malého čtverečku, který vytiskne dálnopis, najdete ve výpisu křížek (#), namísto svislé čárky uvozovky (") a namísto paragrafu dolar (\$).

```

1 rem dos printer atari basic verse 3
2 rem
10 graphics 2: setcolor 2,0,0: setcolor 0,2,6
20 print #6;"          dps"
30 print #6;"          printer"
40 print #6: print #6: print #6;"          prosim cekej"
50 dim a$(540)
60 for i=1 to 540
65 read a
66 sum=sum+a
70 a$(i)=chr$(a)
80 next i
90 if sum=48594 then 120
100 print #6: print #6;"          chyba dat"
110 end
120 graphics 0:?"vloz kazetu, na ktere ma"
130 ? "byt zapsan program ve forme "
140 ? "souboru boot. zmackni play"
150 ? "a record na magnetofonu"
160 ? "a stiskni return "
170 open #1:8,128,"c:" : print #1;a$;:          close #1: end
600 data 0,5,0,7,35,7
610 data 169,60,141,2,211,173
620 data 183,8,141,231,2,173
630 data 184,8,141,232,2,173
640 data 187,8,133,10,173,188
650 data 8,133,11,24,96,169
660 data 7,172,189,8,174,190
670 data 8,32,92,228,173,129
680 data 8,141,27,3,173,130
690 data 8,141,28,3,174,143
700 data 8,172,144,8,32,66
710 data 198,32,71,7,96,234
720 data 234,234,169,56,141,2
730 data 211,169,0,141,0,211
740 data 169,60,141,2,211,160
750 data 1,96,169,56,141,2
760 data 211,169,128,141,0,211
770 data 169,60,141,2,211,169
780 data 127,32,32,8,169,255
790 data 141,66,8,169,192,141
800 data 67,8,32,55,8,169
810 data 0,141,68,8,169,127
820 data 32,32,8,160,1,96

```

830 data 201,155,208,6,32,250
 840 data 7,76,247,7,41,127
 850 data 201,37,240,25,201,38
 860 data 240,21,201,97,48,6
 870 data 201,123,16,2,41,223
 880 data 201,34,48,7,201,91
 890 data 16,3,76,185,7,169
 900 data 0,72,76,193,7,56
 910 data 233,32,72,201,32,48
 920 data 18,172,68,8,240,28
 930 data 169,127,32,32,8,169
 940 data 0,141,68,8,76,226
 950 data 7,172,68,8,208,10
 960 data 169,111,32,32,8,169
 970 data 1,141,68,8,173,69
 980 data 8,201,69,208,3,32
 990 data 250,7,238,69,8,104
 1000 data 170,189,70,8,32,32
 1010 data 8,160,1,96,169,11
 1020 data 32,32,8,169,255,141
 1030 data 66,8,169,128,141,67
 1040 data 8,32,55,8,169,35
 1050 data 32,32,8,32,45,8
 1060 data 32,45,8,32,45,8
 1070 data 169,0,141,69,8,96
 1080 data 162,8,141,0,211,10
 1090 data 32,45,8,202,208,246
 1100 data 96,160,101,140,66,8
 1110 data 160,10,140,67,8,206
 1120 data 66,8,208,251,206,67
 1130 data 8,208,246,96,0,0,
 1140 data 0,0,19,19,83,19
 1150 data 91,19,19,83,123,39
 1160 data 75,71,27,99,31,95
 1170 data 55,119,103,67,43,7
 1180 data 87,115,51,15,59,107
 1190 data 23,63,47,79,19,99
 1200 data 79,59,75,67,91,47
 1210 data 23,51,107,123,39,31
 1220 data 27,15,55,119,43,83
 1230 data 7,115,63,103,95,87
 1240 data 71,131,8,91,7,70
 1250 data 7,68,7,137,7,68
 1260 data 7,68,7,145,8,42
 1270 data 32,68,80,83,32,80
 1280 data 82,73,78,84,69,82
 1290 data 32,32,40,99,41,49
 1300 data 57,56,56,32,66,117
 1310 data 115,101,107,32,32,65
 1320 data 66,46,51,32,42,32
 1330 data 155,32,9,0,0,208
 1340 data 8,191,8,169,1,133
 1350 data 9,173,31,208,41,2
 1360 data 208,3,32,208,8,76
 1370 data 98,228,32,92,7,32
 1380 data 250,7,165,212,72,165
 1390 data 213,72,165,88,133,212

```

1400 data 165,89,133,213,169,23
1410 data 141,185,8,169,39,141
1420 data 186,8,162,0,161,212
1430 data 41,127,201,96,176,2
1440 data 105,32,32,138,7,230
1450 data 212,208,2,230,213,206
1460 data 186,8,16,232,32,250
1470 data 7,206,185,8,16,217
1480 data 104,133,213,104,133,212
1490 data 32,71,7,160,1,96

```

```
=====
dos - printer pro atari basic verze 3
=====
```

```
(c)1988 busek, atariklub jaromer
```

```
=====
pocatek  org $0700      ;obsluha dalnospisu
          dfb $00,$05    ;v atari-basicu
          dfw pocatek
          lda #$3c
          sta $d302
          lda memlo      ;dolni hranice volne ram
          sta $02e7
          lda memlo+1
          sta $02e8
          lda stdos      ;vektor start dos
          sta $0a
          lda stdos+1
          sta $0b
          clc
          rts
start    lda #$07      ;preruseni uzivatele
          ldy prerus
          ldx prerus+1
          jsr $e45c
          lda vekt      ;ulozeni nove adr
          sta $031b      ;tab rutin pro p
          lda vekt+1
          sta $031c
          ldx text
          ldy text+1
          jsr $c642      ;tisk textu
konec    jsr close
basic    rts
close    lda #$38      ;nastaveni portu
          sta $d302      ;na vstupni
          lda #$00
          sta $d300
          lda #$3c
          sta $d302
          ldy #$01
          rts
open     lda #$38      ;prepnuti portu
          sta $d302      ;na vystup
          lda #$80      ;8. bit
          sta $d300
          lda #$3c

```

	sta \$d302	
	lda #\$7f	;start motoru
	jsr busyco	;vyslani a...
	lda #\$ff	
	sta timerl	
	lda #\$c0	
	sta timerh	
	jsr cekl	;ceka na otacky
	lda #\$00	;flag a...
	sta stav	;aktualni prerazeni
	lda #\$7f	;vyslani a...
	jsr busyco	
	ldy #\$01	;status
	rts	
write	cmp #\$9b	;eol ?
	bne nebyl	
	jsr crlf	
	jmp exit	
nebyl	and #\$7f	;zruseni inverse
	cmp #\$25	;jde o ?
	beq mezera	
	cmp #\$26	;jde o ?
	beq mezera	
	cmp #\$51	;poc.obl.malych pismen
	bmi mimo	
	cmp #\$7b	;koniec -"-
	bpl mimo	
	and #\$df	;zmena na velka pismena
mimo	cmp #\$22	;dolni mez tabulky
	bmi mezera	
	cmp #\$5b	;horni mez
	bpl mezera	
	jmp prekod	
mezera	lda #\$00	;kod mezery- \$20
	pha	
	jmp velka	
prekod	sec	
	sbc #\$20	;kod-\$20
	pha	;znak do stacku
	cmp #\$20	;stred tab. pocit.od 0
	bmi mala	
velka	ldy stav	;aktualni zmena
	beq endzmen	;byla velka
	lda #\$7f	;kod a...
	jsr busyco	
	lda #\$00	
	sta stav	;novy stav
	jmp endzmen	
mala	ldy stav	;aktualni zmena
	bne endzmen	;byla mala
	lda #\$6f	;kod l... jsr busyco
	jsr busyco	
	lda #\$01	
	sta stav	;novy stav
endzmen	lda citac	
	cmp #\$45	;pocet znaku v radku
	bne wr1	

```

wrl      jsr crlf          ;bylo jich moc
         inc citac
         pla               ;kod ze stacku
         tax
         lda tab,x         ;cteni dps kodu
         jsr busyco        ;tisk znaku
exit      ldy #$01         ;status
         rts
crlf      lda #$0b         ;kod cr
         jsr busyco
         lda #$ff          ;zpozdeni po navratu vozu
         sta timerl
         lda #$80
         sta timerh
         jsr cek1
         lda #$23          ;kod lf
         jsr busyco
         jsr time20
         jsr time20
         jsr time20
         lda #$00
         sta citac
         rts
busyco    ldx #$08         ;poc. bitu
col       sta #$300        ;bit na port
         asl
         jsr time20        ;ceka 20 ms
         dex
         bne col          ;vyslano vse?
         rts
time20    ldy #$65         ;konst.zpoz
         sty timerl
         ldy #$0a
         sty timerh
cek1      dec timerl       ;odcitani 1
         bne cek1
         dec timerh
         bne cek1
         rts

timerl    dfb $00          ;data a tabulka
timerh    dfb $00
stav      dfb $00          ; 01=1.../ 00=a...
citac     dfb $00          ;soucasna pozice v radku
tab       dfb $13          ;sp tabulka dps kodu
         dfb $13 ;
         dfb $53 ;
         dfb $13 ;#
         dfb $5b ;$
         dfb $13 ;
         dfb $13 ;
         dfb $53 ;
         dfb $7b ;(
         dfb $27 ;)
         dfb $4b ;
         dfb $47 ;+

```

```

dfb $1b ;,
dfb $63 ;-
dfb $1f ;.
dfb $5f ;/
dfb $37 ;0
dfb $77 ;1
dfb $67 ;2
dfb $43 ;3
dfb $2b ;4
dfb $07 ;5
dfb $57 ;6
dfb $73 ;7
dfb $33 ;8
dfb $0f ;9
dfb $3b ;:
dfb $6b ;;
dfb $17 ;(
dfb $3f ;=
dfb $2f ;)
dfb $4f ;?
dfb $13 ;
dfb $63 ;a
dfb $4f ;b
dfb $3b ;c
dfb $4b ;d
dfb $43 ;e
dfb $5b ;f
dfb $2f ;g
dfb $17 ;h
dfb $33 ;i
dfb $6b ;j
dfb $7b ;k
dfb $27 ;l
dfb $1f ;m
dfb $1b ;n
dfb $0f ;o
dfb $37 ;p
dfb $77 ;q
dfb $2b ;r
dfb $53 ;s
dfb $07 ;t
dfb $73 ;u
dfb $3f ;v
dfb $67 ;w
dfb $5f ;x
dfb $57 ;y
dfb $47 ;z
vekt dfw adr ;vektor obsluznych rutin
adr dfw open-1
dfw close-1
dfw konec+1
dfw write-1
dfw konec+1
dfw konec+1
text dfw adr1
adr1 asc "+ dps printer "
asc "(c)1988 busek ab.3+ "

```

	dfb \$9b	
memlo	dfw end+5	;prvni volna ram
soucrad	dfb \$00	;citac radku
soucs1	dfb \$00	;citac sloupce
stdos	dfw tiskobr	
prerus	dfw adr2	
adr2	lda #\$01	;boot?
	sta \$09	
	lda \$d01f	;cteni consol
	and #\$02	;select
	bne exitvbw	
	jsr tiskobr	;tisk obrazovky
exitvbw	jmp \$e462	;exitvbw
tiskobr	jsr open	
	jsr crlf	
	lda \$d4	;pracovni prostor
	pha	
	lda \$d5	
	pha	
	lda \$58	;adresa videoram
	sta \$d4	
	lda \$59	
	sta \$d5	
	lda \$17	;pocet radku sr.0
	sta soucrad	
novline	lda #\$27	;pocet sloupce
	sta soucs1	
	ldx #\$00	
ctiznak	lda (\$d4,x)	;cteni znaku z videoram
	and #\$7f	;nulovani 8. bitu
	cmp #\$60	
	bcs tiskni	
	adc #\$20	
tiskni	jsr write	;tisk znaku na dps
	inc \$d4	;dalsi znak
	bne hbyte	
	inc \$d5	
hbyte	dec soucs1	;sloupec-1
	bpl ctiznak	
	jsr crlf	;konec radku
	dec soucrad	;radek-1
	bpl novline	;dalsi radek
	pla	;end tisk, obnova adres
	sta \$d5	
	pla	
	sta \$d4	
	jsr close	
	ldy #\$01	;status
end	rts	

5.1.4. Přídavná klávesnice

(Překlad z <9> poskytl kolegové ze Západočeského kraje prostřednictvím ins. Miloše Adama)

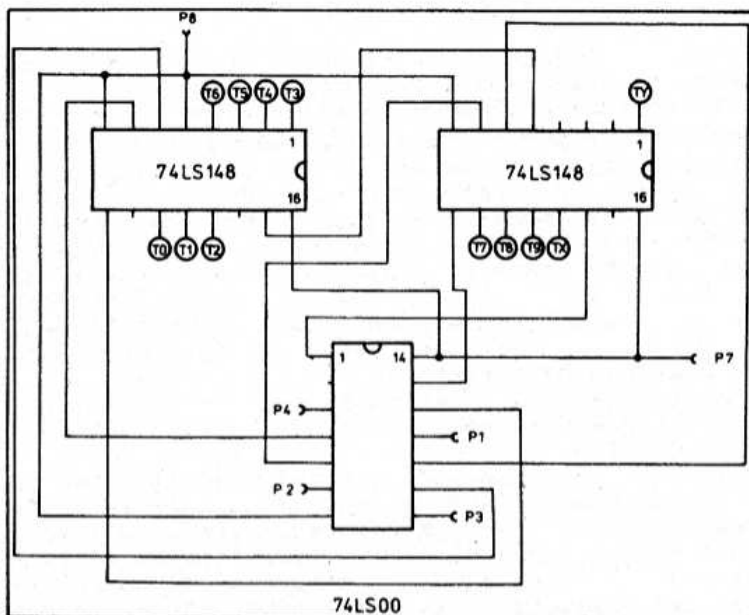
Pomocí velmi jednoduchých a levných prostředků lze připojit do joy-portu přídavnou klávesnici. Zopakujeme si, že celé tajemství připojení křížového ovladače je v obvodu PIA. Jedná se o standardní periferní obvod 6520. Obvod obsahuje mj. dva 8bitové porty použitelné jak pro vstup, tak i pro výstup dat. Využití portů je programově volitelné. Je nutné připomenout, že dřívější modely počítačů ATARI komunikovaly přes 4 joy-ports, nové modely komunikují pouze přes 2 konektory. Vlivem tohoto omezení nebylo možné druhý port PIA vyvést ven. Z 16 datových bitů tedy zůstalo jen 8. Technický zásah do počítače spočívající ve vyvedení zbývajících bitů není možný. Tento port totiž přejímá jiné úkoly, především spolupráci s MMU (paměťový organizátor).

Za těchto okolností se může zdát připojení přídavné klávesnice nereálné. Klávesnice totiž obsahuje 12 tlačítek, ale může jich mít až 16. (0-15).

Navržené zapojení (viz obr.14) sestává mimo tlačítek ze tří IO - jednoho 74LS00 a dvou 74LS148, což je dekódér priority (8 za 3). V zapojení se využívá jen jeden vstup ovladačů. Využívají se jen 4 datové linky k přenosu maximálně 16 různých stavů. Napájecí napětí pro IO je odebíráno z konektoru joy-portu, takže se vystačí bez přídavného zdroje. Výpis programu ukazuje, jak je možné testovat klávesnici. Není-li žádné tlačítko stlačené, získá se 0, jinak číslo od 1 do 11.

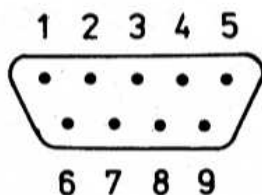
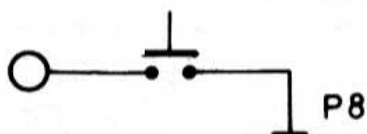
Výpis programu

```
100 REM =====
110 REM =ZUSATZTASTATUR=UEBER=
120 REM =DIE=PIA=ABFRAGEN=====
130 REM =====
140 REM =von=Wolfsang=Czerny==
150 REM =====
160 GRAPHICS 0
200 X=PEEK(632)
210 IF X=0 THEN 200
220 ? "DIE=GETIPTE=ZAHL=LAUTET=";X-1
230 GOTO 200
```



Obr. 14 Přídavná klávesnice

T0÷TY - klávesy



P1÷P8 - špičky joyportu 2

Obr. 14 Přídavná klávesnice - zapojení

5.1.5. Ovladače

Konstrukčnímu řešení křížových ovladačů, tomuto naprosto nezbytnému doplňku řady uživatelů počítačů ATARI XL/XE, jsme na stránkách ZAK věnovali dostatečnou pozornost hned v prvních číslech.

Přehledně bylo na toto téma pojednáno v článku M. Vavrdy "Jak se dají využívat ovládací vstupy u počítačů ATARI 800 XL a 130 XE/" (viz ZAK 2/87, s.14-18).

V článcích "Návod na výrobu jednoduchého ovladače (joysticku) typ RH1 a konektoru" autora R. Hány (viz ZAK 2/87, s.28-30), "Jednoduchý ovladač (joystick) pro ATARI 800 XL (130 XE), typ PR1" od P.R. (viz ZAK 3/87, s.29-30), "Výroba konektoru pro ovladač (joystick)" od autora J. Štěrbý (viz ZAK 3/87, s.28-29) jsme uvedli vše, co je potřebné vědět při amatérské stavbě ovladače včetně konkrétních návodů a schematic.

V článku "Stavíte rovněž ovladač s potenciometrem (PADDLE)?" od autora Dr. J.Boka, CSc. (viz ZAK 2/87, s.25-28) jsme zveřejnili návod dalšího zajímavého ovladače.

V tomto výčtu musíme připomenout i návod na stavbu světelného pera od autora ing. J.Biskupa (viz 4/87, s.26-29).

Jsmo toho názoru, že v tomto sborníku je potřebné se alespoň okrajově ovladači zabývat. V této úvaze nám náramně pomohl ing. Jiří Chábera, který poskytl návod na stavbu hned elektronického křížového ovladače. Tento návod si zaslouží zařazení do sborníku, mj. proto, že autor poskytl vynikajícím způsobem zpracovanou dokumentaci.

Elektronický křížový ovladač

ing. Jiří Chábera

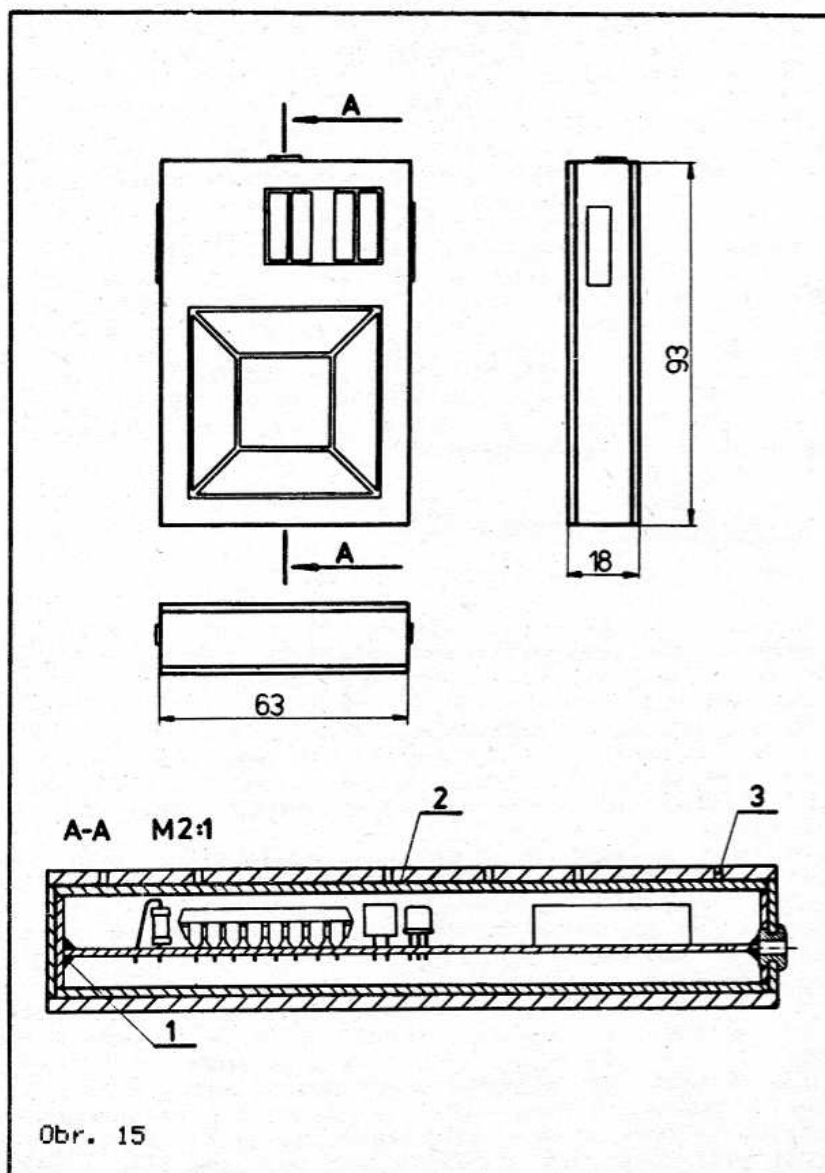
Popisovaný křížový ovladač je možné použít ke všem osmibitovým počítačům Atari, k počítačům Commodore a Sharp MZ, které jsou vybaveny standardními I/O porty.

Je určen pro nejnáročnější uživatele, a to jak z hlediska vlastního (hrubého) zacházení, kterému se zvláště u některých her nelze ubránit, tak z hlediska komfortu obsluhy a možnosti využití speciálních funkcí. Kromě dvou mikrospínačů totiž ovladač neobsahuje žádné mechanické prvky, takže je prakticky nezničitelný.

Protože ovladač v podstatě pracuje v kódu BCD, známém z techniky logických obvodů, lze vhodným uspořádáním nebo kombinací pomocných sensorových plošek zvýšit jeho užité vlastnosti. Na obr.15 je zobrazen vlastní ovladač s uspořádáním hlavních a pomocných sensorových plošek, které je podle mého názoru optimální.

Hlavní směry, které se promítají na portu počítače a nepřímou ukládají do systémové proměnné STICK(n) jako decimální hodnoty 14, 7, 13 a 11, se aktivují vodivým spojením jedné z hlavních plošek se střední čtvercovou ploškou nebo s kostrou zařízení. Směry příčné, 10, 6, 5 a 9, pak spojením dvou sousedních plošek se zemí. U klasického ovladače nelze ani při nejlepší vůli dosáhnout jiných hodnot proměnné STICK, než právě uvedených, nepočítáme-li pochopitelně klidovou polohu 15. S

popisovaným ovladačem lze spojením vhodné kombinace senzorových plošek této možnosti plně využít a získat tak šestnáctipovelový ovladač. Jehož využití si každý počítačový fanda snadno domyslí. Nestandardních hodnot lze pohodlně dosáhnout pomocnými senzorovými ploškami, které jsou umístěny nad různými hlavními směry (viz obr.15). S profesionálním software je popisovaný ovladač samozřejmě plně kompatibilní.



Novopečení majitelé počítačů, kteří zatím dávají přednost nejrozumnějším hrám (a nejen oni), jistě ocení tzv. autofire, který je navíc řízen bistabilním klopným obvodem a indikován LED diodou. Automatický trisr je nejen vhodný pro ovládání některých her, ale také podstatně zvyšuje životnost použitých mikrospínačů, jejichž cena není právě nejnižší. Funkce bude podrobně vysvětlena při popisu elektronické části.

Lepeších výsledků, než s použitím klasického pákového ovladače, lze dosáhnout pouze reflexivními pohyby prstu po ovládací ploše, přitom se však nesmí vizuálně kontrolovat jeho pozice. Dokonalé zvládnutí bezchybného ovládání senzorového ovladače je pouze otázkou času.

Konstrukce mechanické části

Předem podotýkám, že mechanická část elektronického ovladače nemá v žádném případě rozhodující vliv na jeho funkci a je věcí každého konstruktéra, jak ji realizuje. Předkládaný návod je třeba chápat jen jako doporučení nebo inspiraci.

Na obr.15 je promítnut vlastní ovladač se základními rozměry. V dolní části jsou pak podélný řez A-A ve dvojnásobném zvětšení. V horní části nárys, nad směrovou růžicí, jsou umístěny pomocné senzorové plošky a z bokorysu je patrné umístění mikrospínačů. Vzhledem k tomu, že existuje celá řada mikrospínačů různých rozměrů, nemá význam se jejich montáží podrobněji zabývat.

Nejvhodnějším materiálem pro zhotovení vrchní a spodní desky je hliníkový nebo duralový plech tloušťky 2 - 3 mm, který se dobře opracovává a úspěšně odolává korozi. Všechny ostatní části jsou zhotoveny z kuprexitu, jehož poměděné strany je využito ke spojení jednotlivých dílů pájením. V horní části ovladače je umístěna pružová průchodka pro přívodní kabel, místo které lze případně použít malý sedmipólový konektor.

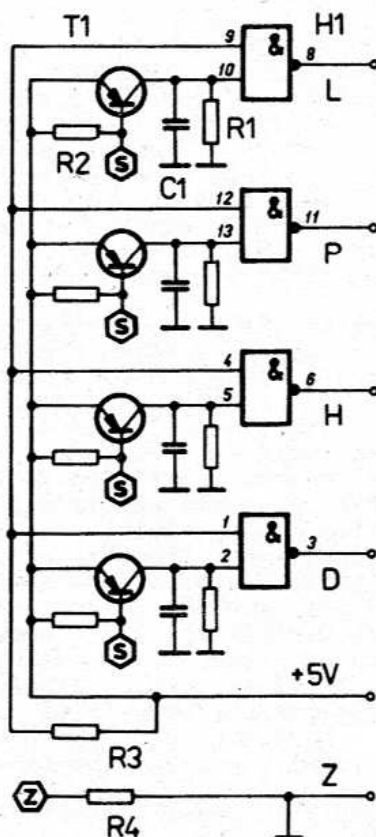
Pečlivě obroušené segmenty vyřezané z hliníkového plechu je nutné předem bezproudově poniklovat v Niklíku, který lze běžně zakoupit v prodejnách drogerie za 13 Kčs; na hliník a jeho slitiny nelze totiž bez problémů pájet. Stejným způsobem se pak připraví hliníkový rámeček, do kterého se segmenty uloží tak, aby se vzájemně nedotýkaly. Takto sestavená vrchní deska se nalepí (pozice 2 - lepeno) na destičku kuprexitu s vyvrtanými otvory pro připájení vodičů a spáry mezi segmenty se zalijí nevodivou dvousložkovou pryskyřicí (pozice 3 - zalito). Rozměry kuprexitové destičky jsou o dvojnásobek tloušťky kuprexitu menší než rozměry hliníkového rámečku, jak je patrné z řezu A-A.

Podobným způsobem se zhotoví i spodní deska, která slouží jako víko celého ovladače. Do otvorů v bočnicích se zevnitř zasune malý hliníkový hranolek podlepený o málo větší kuprexitovou destičkou, která se dotýká funkční části mikrospínače. Výška bočnic, které se spojí pájením, závisí na rozměrech použitých mikrospínačů a na schopnostech konstruktéra minimalizovat výšku osazení plošného spoje. Vlastní plošný spoj se připájí (pozice 1 - pájeno) k distančním kuprexitovým destičkám a volně zasune do vytvořené krabičky.

Ovladač je nutné nakonec po všech stranách obrousit jemným smirkovým papírem, nejlépe pod vodou. Indikační LED diodu je možné umístit na libovolném místě na horní desce. Plošný spoj je však navržen pro její umístění v levém horním rohu.

Elektronika je rozdělena na dvě základní části: obvod senzorového ovladače a obvod autofire.

Základem senzorové části (viz obr.16) je IO1 (MH7400), místo něhož lze použít všechna běžně dostupná hradla NAND, včetně Schmittových klopných obvodů (74132PC) a obvodů CMOS (MHB4011). Při náhradě IO1 je však nutné příslušně změnit hodnoty všech rezistorů (zejména při použití obvodů CMOS).



Obr. 15

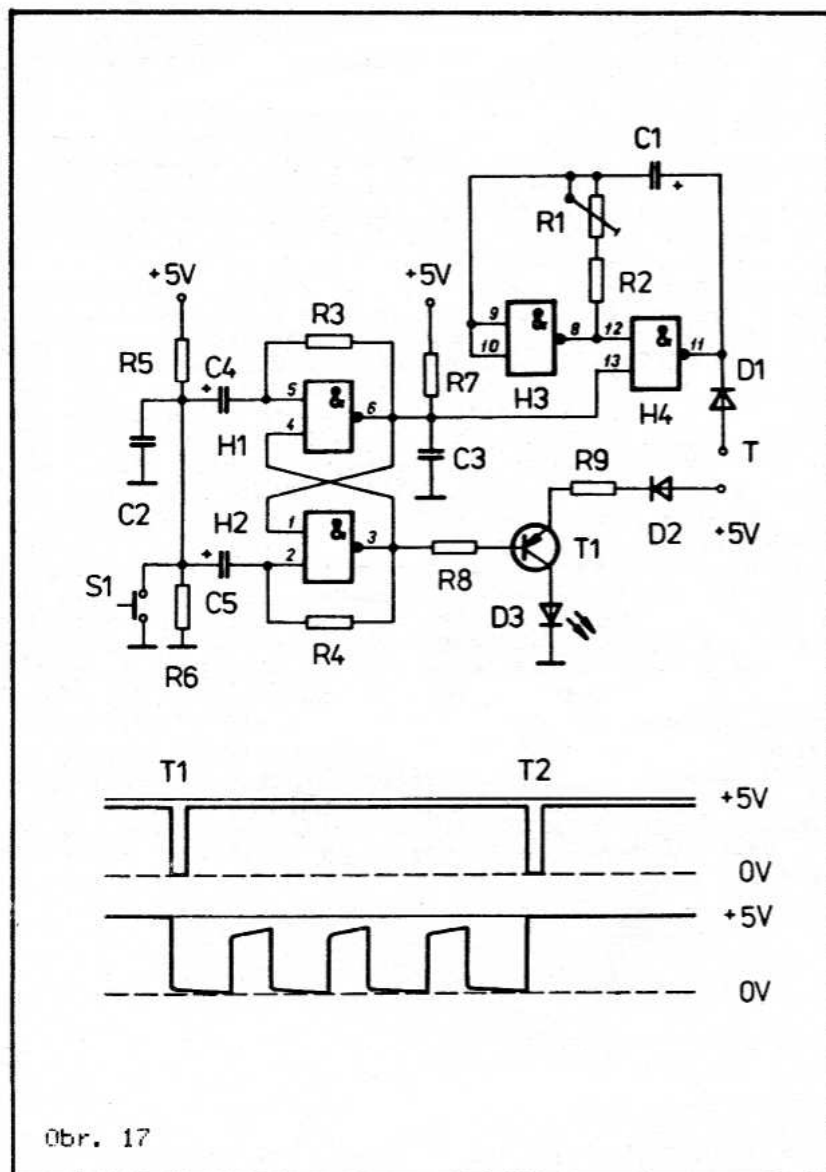
V klidovém stavu je na vstupu všech hradel los 1. Spojením senzorové plošky S se zemí Z se uvede tranzistor T1 do vodivého stavu a tím se vyvolá úbytek napětí na rezistoru R1, který odpovídá los 1 na vstupu 10 hradla H1. Protože druhý vstup hradla (9) je připojen přes rezistor R3 na napájecí napětí, hradlo se překlápí a na výstupu 8 bude los 0, která vyvolá změnu BCD hodnoty na portu počítače. Hodnotou rezistoru R2 lze v širokém rozsahu měnit citlivost senzoru (vodivost pokožky je u každého jiná), kondenzátor C1 slouží k potlačení zákmitů tranzistoru, které by jinak vyvolaly rušení obrazu. Aby nedošlo k proražení přechodu EB tranzistoru náhodným spojením kovovým předmětem, je mezi zemní senzor a kostru zařízení vložen rezistor R4, který není na desce plošných spojů. Funkce ostatních hradel je analogická.

Obvod autofire je možné rozdělit na dva samostatné funkční celky, a to na bistabilní klopný obvod (hradla H1 a H2) a na astabilní klopný obvod (hradla H3 a H4). Vzhledem k tomu, že pracovní proudy obvodů CMOS jsou na úrovni svodových proudů kondenzátorů, nelze z těchto obvodů bez problémů realizovat takto řešený bistabilní KO. Jinak platí to, co bylo řečeno o IO1.

Funkce BKO je natolik známá, že ji není třeba blíže popisovat. Kondenzátor C2 s rezistorem R6 potlačují zákmity mikrospínače a omezují vliv vnějšího rušení elektromagnetickým polem. Rezistor R7 spolu s kondenzátorem C3 zajišťují nastavení BKO při připojení napájecího napětí. Rezistor R8 v obvodu indikace omezuje proud bázi tranzistoru T1, rezistor R9 pak proud LED diodou. Protože los 1 na vstupu 3 hradla H2 je díky zpětnovazebnému rezistoru R4 napětově nižší než napětí na emitoru tranzistoru T1, je nutné zařadit do obvodu tohoto tranzistoru diodu D2; jinak zůstane tranzistor pootevřen a LED dioda bude slabě svítit.

Vlastní astabilní KO je tvořen hradly H3 a H4, rezistory R1 a R2 a kondenzátorem C1. Změnou odporu trimru R1 lze dosáhnout jemné změny kmitočtu výstupního napětí. Rezistor R2 pak zabraňuje nekontrolovanému rozkmitání AKO. Vstup 13 hradla H4 má řídící funkci; pokud je na něm los 0 je AKO blokován a na výstupu 11 hradla H4 je los 1, to znamená, že trisr není aktivován. K výstupu označeném T je paralelně připojen druhý mikrospínač, který má funkci obyčejného trisru. Dioda D1 chrání hradlo H4 proti zkratu při stlačení druhého mikrospínače, což výrazně snižuje spotřebu obvodu. Na místě kondenzátorů C1, C4 a C5 je vhodné použít malé tantalové kapky, které jsou v současné době v naší obchodní síti běžně k dostání, i když ne za zanedbatelnou cenu.

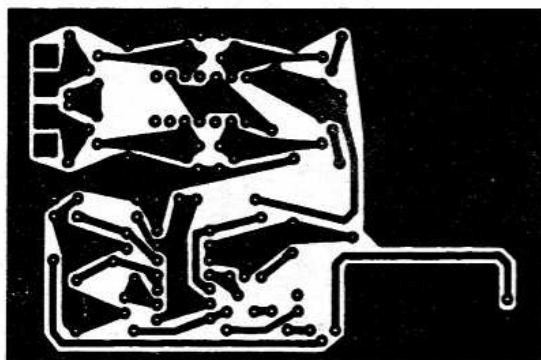
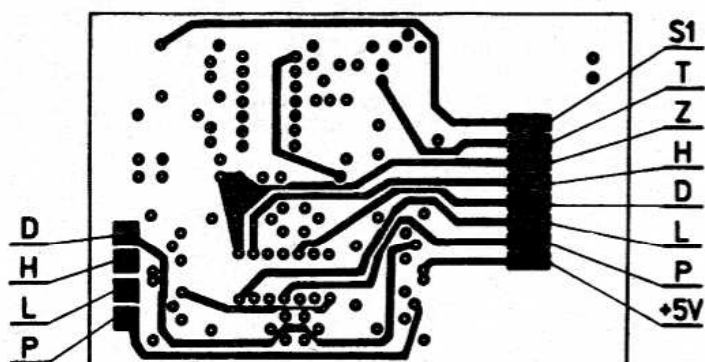
Ve spodní části obr.17 je znázorněn časový průběh napětí na spínači S1 a na výstupu 11 hradla H4. Při stisku tlačítka S1 se přepne BKO a odblokuje AKO, který se rozkmitá. Při vyšších kmitočtech má napětí pilovitý charakter, při kmitočtech vyšších se tvar napětí blíží hraně. Astabilní KO kmitá až do opětovného stisknutí mikrospínače S1. Pokud je AKO v činnosti (vstup 13 hradla H4 - log 1) svítí indikační dioda D3 (výstup 3 hradla H2 - log 0).



Plošný spoj

Na obr.18 je zobrazen oboustranný plošný spoj, horní deska ze strany součástek a dolní ze strany spojů. Měřítka uvedené v pravém horním rohu platí pro formát A4, ve formátu A5 velikost plošného spoje odpovídá skutečnosti. (Při tisku nelze tento údaj zaručit - pozn.red.)

M 1,5:1

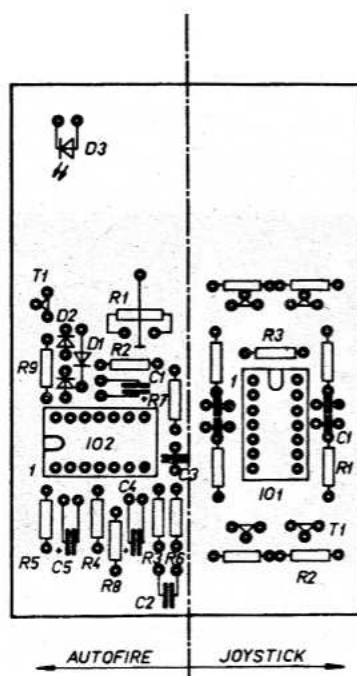


Obr. 18

V pravé části kuprexitové desky je prostor pro připevnění mikrospínačů. To lze realizovat přilepením nebo přišroubováním. Efektivnější je ale vyříznout v kuprexitu lůžka a spínače do nich pevně zasunout. Čtveřici kontaktů je nutné provrtat a vzájemně propojit horní a dolní plošný spoj. Kontakty slouží k připojení senzorových plošek. Na kontaktní pole se shora připojí mikrospínač S1 proti zemi, mikrospínač S2 proti zemi a zároveň přívodní vodič trisru (T), zemní vodič (Z), horní (H), dolní (D), levý (L) a pravý (P) směr, napájení (+5V).

Samostatné nepropojené plošky okolo vývodů (zejména na desce ze strany součástek) není samozřejmě nutné na plošném spoji vytvářet. Slouží pouze pro snazší orientaci.

Na obr. 19, o jehož měřítku platí totéž, co bylo uvedeno o obr. 18, je zobrazeno rozložení součástek na plošném spoji. Je navržen pro malé kapkové kondenzátory (C1, C4, C5) (ostatní kondenzátory jsou keramické na nízké napětí), odporový miniaturní trimr, tranzistory v plastovém pouzdře s drátovými vývody a na miniaturní rezistory. Jakékoliv jiné součástky vyžadují změny ve struktuře obrazce spoje. Na desce je počítáno se zapojením dvou diod D2 pro případ, že by jedna nestačila. Pokud ji však nepoužijeme, je nutné kontakty propojit drátovou spojkou.



M 1,5:1

Obr. 19

Oživení a nastavení obvodů

Spotřeba celého ovladače se pohybuje podle typu použitých IO v rozsahu 5 - 30 mA v klidovém stavu. Před zapojením napájecího napětí je třeba překontrolovat ohmetrem odpor mezi zemí a kontaktem +5 V. Pokud je nulový, je nutné navít a odstranit zkrat na plošném spoji. Nejlépe je oživovat zapojení s plochou baterií - při případném zkratu ušetříme napájecí obvody počítače. V klidovém stavu musí být napětí na výstupech hradel IO1 na úrovni los 1, v opačném případě je chyba v obvodu báze tranzistoru (zkrat, senzoru, malý odpor izolace, atd.). Napětí na vývodech 1, 4, 9 a 12 musí být na úrovni los 1. Pokud bude při spínání rušen obraz televizního přijímače, je možné zvýšit kapacitu kondenzátorů z 10 nF na 22 nF.

Obvody autofire je vhodné oživovat po částech. Podotýkám, že uvedené hodnoty RC zaručují bezvadnou funkci zapojení pouze ve spolupráci s IO MH7400. Pro jiné IO je třeba vypočítat nové hodnoty RC. Hlavní vliv na schopnost BKO klopit mají kapacity C4 a C5, zpětnovazební rezistory R4 a R5 a velikost zátěže na výstupech 3 a 5. Pokud ani změnou těchto hodnot nedosáhneme perfektní funkce BKO, je možné změnit tvrdost napětového děliče R5/R6, obvykle to však není nutné.

Pokud BKO klopí k naší spokojenosti a pracuje-li indikace, je možné přistoupit k nastavení oscilátoru. Pokud máme k dispozici osciloskop, je nastavení jednoduché a příjemné. V opačném případě je nutné zapojit výstup (T) k portu počítače a vytvořit jednoduchý BASIC program, který indikuje stav proměnné STICK(n). Pokud AKO vůbec neklopí, je nutné měnit velikost odporu R2.

K funkci autofire je nutné poznamenat následující. Každý program, ať jde o BASIC nebo MC, čte stav portu ve smyčce, tedy jistou frekvencí. Podle zákona schválnosti je tato frekvence řádově shodná s používanou frekvencí AKO. Výsledný efekt se projevuje v modulaci vstupního signálu nízkým kmitočtem, což lze pozorovat při nejrůznějších hrách jako střelbu dávkami. Tento efekt je možné minimalizovat nastavením trimru do optimální polohy při každé hře. Pro tento účel je třeba vyvrtat ve spodní desce ovladače otvor pro malý šroubováček. Zdatnější konstruktéři mohou místo odporového trimru použít miniaturní potenciometr.

Seznam součástek (k obr.16)

I01 - MH7400 (MH7403, MH7437, MHB4011 - viz text)
T1 - KC308 (KC309, KC307)
C1 - 10-22 nF, keramický
R1 - 1k5-2k2
R2 - M1-M58 (podle citlivosti)
R3 - 1k2
R4 - 470

Seznam součástek (k obr.17)

I02 - MH7400 (MH7437, MH7403)
T1 - KC308 (KC309, KC307)
D1 - libovolná Si
D2 - Si, I(F)>20 mA
D3 - libovolná LED
C1 - 10M, tantalová kapka
C2 - 22 pF, keramický
C3 - 120 pF, keramický
C4,C5 - 3M3, tantalová kapka
R1 - 220-330, miniaturní trimr
R2 - 100
R3,R4 - 68k-M1
R5 - 12k-15k
R6 - M12-M15
R7 - 6k8-10k
R8 - 4k7-10k
R9 - 220-330

5.1.6. Propojení počítačů

(převzato z <5>)

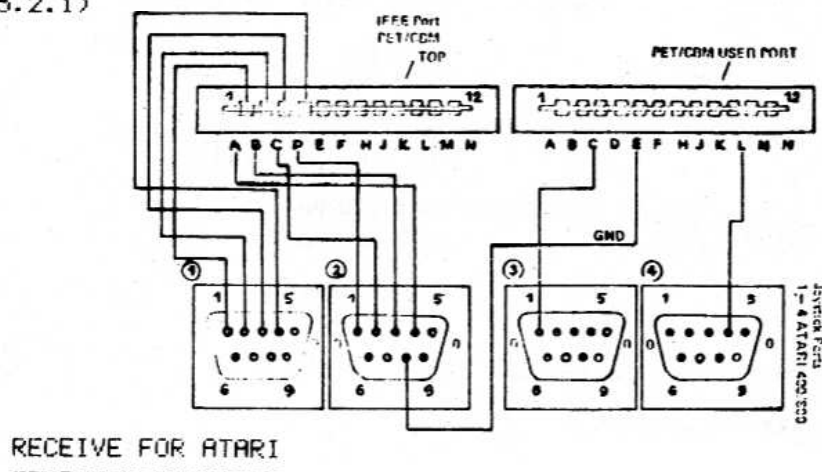
Propojení počítačů ATARI pro přenos dat na jiný typ počítače demonstruje následující aplikace, která umožňuje propojení počítače řady 400/800 s počítači řady PET/CBM fy Commodore.

Aplikace je pro svou jednoduchost založena na využití všech čtyř Joy-portů. Prvé dva (port A) Joy-porty 1 a 2 jsou využity jako datová sběrnice přístupná pro oba počítače (PET/CBM pracuje na rozhraní IEEE). Joy-porty 3 a 4 (port B) jsou použity pro předávání sběrnice a řízení směru přenosu, bit 0 pro ATARI, bit 7 pro PET/CBM.

Rutiny PUT a GET, na které se uvedený program odvolává nejsou v listinu funkční. Program slouží pouze pro ilustraci principu předávání sběrnice mezi počítači. Pro konkrétní aplikaci je nutno, kromě těchto rutin, doplnit kritéria pro ukončení přenosu.

Při použití počítačů řady XL je nutno kromě uprav handleru doplnit minimální hardware pro řízení datové sběrnice, nebo použít jednosměrný přenos dat jinou formou.

Jiný způsob propojení dvou počítačů je založen na propojení pomocí sériového portu s řízením přenosu typu Kermit (viz kap.5.2.1)



```

PORTB EQU $D301
PBCTL EQU $D303
PORTA EQU $D300
PACTL EQU $D302

```

```

PUT EQU $3309

```

```

ORG $A800

```

```

* SET BIT 0 ON PORTB
* AS OUTPUT

```

```

A800: A930          LDA #$30
A802: 8D03D3        STA PBCTL
A805: A901          LDA #/00000001
A807: 8D01D3        STA PORTB
A80A: A934          LDA #$34
A80C: 8D03D3        STA PBCTL

                *      GIVE YOUR HAND TO THE
                *      PET

A80F: A901          RFD      LDA #1
A811: 8D01D3        STA PORTB

                *      WAIT UNTIL PET TAKES
                *      YOUR HAND

A814: 2C01D3 WAITDAY BIT PORTB
A817: 30FB          BMI WAITDAY

                *      GET DATA FROM BUS
                *      PUT THEM SOMEWHERE

A819: AD00D3        LDA PORTA
A81C: 200933        JSR PUT

                *      TAKE YOUR HAND BACK

A81F: A900          LDA #0
A821: 8D01D3        STA PORTB

                *      WAIT UNTIL PETS HAND
                *      IS IN HIS POCKET

A824: 2C01D3 WAITDAYN BIT PORTB
A827: 10FB          BPL WAITDAYN

                *      START AGAIN
A829: 4C0FA8        JMP RFD

```

PHYSICAL ENDADDRESS: \$A82C

*** NO WARNINGS

```

PORTB      $D301
PORTA      $D300
PUT         $3309
WAITDAY     $A814
PBCTL      $D303
PACTL      $D302      UNUSED
RFD        $A80F
WAITDAYN    $A824

```

SEND FOR PET CBM

PORTB EQU \$E84F

PBCTL EQU \$E843
PORTA EQU \$A822

GET EQU \$FFCF USER GET BYTE
* ROUTINE

ORG \$033A,\$A800

```

        *          SET BIT 7 ON PET
        *          TO OUTPUT
033A: A980      LDA #/10000000
033C: 8D43E8     STA PBCTL

        *          GET DATA FROM USER
        *          PUT IT ON BUS

033F: 20CFFF GETDATA JSR GET
0342: 8D22A8     STA PORTA

        *          TELL ATARI DATA VALID

0345: A900 DAV    LDA #0
0347: 8D4FE8     STA PORTB

        *          WAIT UNTIL ATARI
        *          GIVES HIS HAND

034A: AD4FE8 WAITNRFD LDA PORTB
034D: 2901       AND #/00000001
034F: D0F9       BNE WAITNRFD

        *          SHAKE HANDS WITH ATARI

0351: A980 DAND   LDA #/10000000
0353: 8D4FE8     STA PORTB

        *          WAIT UNTIL ATARI RELEASE
        *          HIS HAND

0356: AD4FE8 WAITRFD LDA PORTB
0359: 2901       AND #/00000001
035B: F0F9       BEQ WAITRFD

        *          START AGAIN WITH DATA

035D: 4CF03      JMP GETDATA
```

PHYSICAL ENDADDRESS: \$A825

*** NO WARNINGS

PORTB \$E84F
PORTA \$A822
GETDATA \$033F
WAITNRFD \$034A
WAITRED \$0356

PBCTL	\$E843	
GET	\$FFCF	
DRV	\$0345	UNUSED
DANV	\$0351	UNUSED

5.2 Sériový port

5.2.1 Připojení tiskárny

ing. Zdeněk Hájek, CSc., Praha

Cílem řešení bylo vytvořit vnější rozhraní pro připojení tiskárny tak, aby to svou inteligencí zabezpečilo nutnou komunikaci počítače ATARI s periférií. Pak není nutné vkládat do počítače dodatečně tiskové rutiny a příkazy pro výstup (LPRINT, LIST atd.), budou prováděny přímo. Takto řešené rozhraní navíc umožní realizaci několika typů rozhraní volbou pevně zabudovaných podprogramů.

Předtím, než přistoupím k vlastnímu popisu řešení, bude vhodné říci několik slov ke způsobu komunikace počítače ATARI s tiskárnou, kterou má rozhraní modelovat. Spojení je asynchronní, sériové, rychlostí 19200 Bd přes konektor sériového portu. Využívány jsou tyto signály (uvedena je i špička 13 pólového konektoru):

```

3 ..... DATA IN
4 ..... GND, zem
5 ..... DATA OUT
7 ..... COMMAND

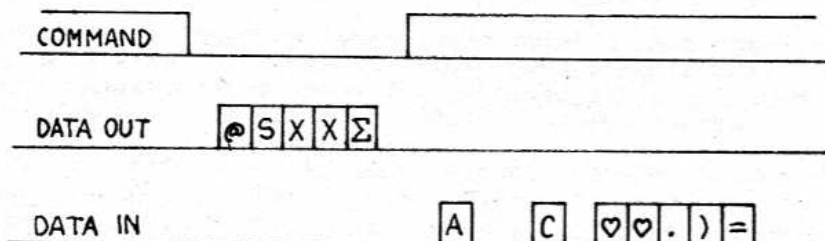
```

DATA IN, OUT jsou kanály pro vstup a výstup sériových dat, signál COMMAND slouží k rozlišení povelů od dat. Normálně je na úrovni 1, v případě vysílání povelů přechází do nuly. Všechny signály jsou na úrovni TTL. Přenos dat pracuje v komunikačním protokolu sériového portu ATARI (viz kap.3.3).

Vlastní komunikaci můžeme rozdělit do dvou částí:

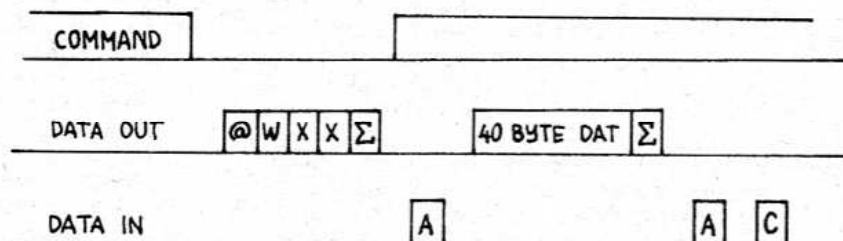
1. Dotaz počítače na status tiskárny (periférie pak odpovídá).
Dotaz na status je vyslán vždy před započatím komunikace (LPRINT atd.).
2. Vlastní přenos dat.

Identifikační kód je #40 ('@'). Příkaz dotazu na status je #53 ('S?') pro přenos dat pak #57 ('W?'). Na každý povel periférie odpovídá. Na obr.20 je časový diagram komunikace při dotazu počítače na status tiskárny. Připojení tiskárny je potvrzeno vysláním znaku #41 ('A'-ACKNOWLEDGE). Pak je vyslán znak #43 ('C'-COMPLETE) a série 4 znaků #00, #00, #14, #29 zakončených kontrolním součtem #3D.



Obr. 20 Komunikace při dotazu na STATUS

Následuje vlastní přenos dat podle časového diagramu na obr.21. Po přijetí povelu pro zápis dat se tiskárna ohlásí znakem #41 ('A'). Pak následuje vlastní přenos vloku 40 byte ukončený kontrolním součtem. Pokud je vše v pořádku, vyšle tiskárna opět znak #41 ('A'). Je-li připravena na příjem dalších dat, vyšle znak #43 ('C') a vše se opakuje včetně vyslání povelu pro zápis.



Obr. 21 Komunikace při vyslání dat

Data jsou vždy vysílána v blocích po 40 bytech. Logické ukončení dat v bloku je provedeno znakem #9B (EOL). Zaplnění zbytku bloku při počtu dat menším než 40 závisí na povelích pro úpravu tisku použitých v BASICu. Při povelu LPRINT je po příslušném počtu znaků vyslán znak #9B a zbytek do 40 vyplněn mezerami #20. Při LPRINT, následuje za zprávou 10 znaků mezery, pak ukončovací znak #9B a zbytek je zaplněn mezerami. Při povelu LPRINT; je za zprávou #9B a zbytek zaplněn ukončovacími znaky #9B. Toto řešení umožňuje tiskárně organizovat tisk podle povelů. Pokud je logická délka zprávy delší než 40 znaků, je rozložena do více bloků a znak #9B (EOL) je až na logickém konci zprávy.

Z uvedeného popisu komunikace vyplývá, že rozhraní by bylo velmi obtížné sestavit v pevné logice, a že nejlhodnější je navrhnout ho jako inteligentní s použitím mikroprocesoru (např. jednočipový MH 80 35).

Činnost mikroprocesorem řízeného rozhraní můžeme rozdělit do 5 částí:

1. dialog s počítačem (reakce na dotazy a povel),
2. přijetí a uschování bloku dat,
3. zpracování povelů pro úpravu textu (, ;),
4. transformace tabulky kódů znaků vyslaných počítačem ATARI do tabulky kódů znaků, které je tiskárna schopna zpracovat,

5. vyslání dat do tiskárny v požadovaném tvaru a komunikace s ní. Hlášení o splnění úlohy.

Činnost pod prvními třemi body je obecná a je stejná při použití libovolného typu rozhraní (sériové, paralelní atd.). Další body jsou již závislé na konkrétně připojeném zařízení. Proto je možné celkem snadno modifikovat program pro jiné periférie.

Z tohoto důvodu uvádím kromě zápisu programu i vývojové diagramy jednotlivých úseků. Pro snazší orientaci jsou v obrázcích uvedeny i počáteční adresy bloků. Bloky s postranními čarami značí podprogramy. Startovací adresa je uvedena uvnitř bloku. Hlavní program je na obr.22 a 23, podprogramy pro příjem a vyslání znaku (komunikace s ATARI) na obr.24 a 25, podprogram pro zpracování přijatého bloku dat označený jako TISK na obr.26 a podprogram pro vyslání znaku na tiskárnu na obr.27. Vlastní výpis programu je v tabulce 1.

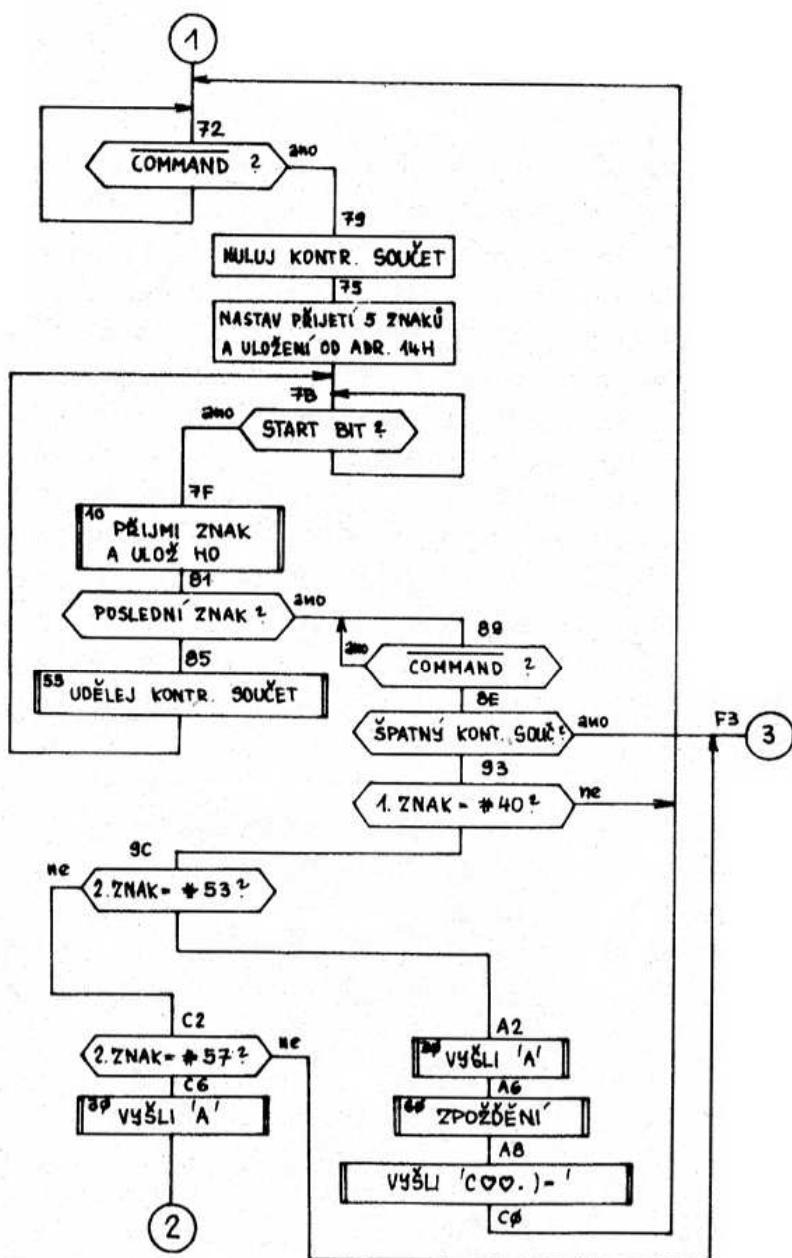
Signály DATA OUT přijímá mikroprocesor na vstupu INT, DATA IN vysílá na portu P 27, COMMAND přijímá na portu P 26 (viz obr.28). S tiskárnou komunikuje mikroprocesor přes port P 1 a vstup T0. Paralelní data vystupují na portu P 11 - P 17 (7 bitů); P 10 je výstup data platná a vstup T0 přijímá hlášení o připravenosti tiskárny na příjem dalších dat. Pak teprve může rozhraní vyslat do počítače ATARI znak kompletnosti #43 ('C').

Odchylky v programovém řešení rozhraní pro různé tiskárny nebo jiná zařízení (plotr) se promítnou do vývojového diagramu na obr.7, kde v zarámované části je adresní část programu. Nakreslený diagram platí pro tiskárnu DZM 180, která nemá grafické znaky a může zpracovávat pouze alfanumerické znaky: velkou abecedu, číslice a základní značky. Proto program převádí malá písmena na velká a inverzní znaky na normální. Grafické symboly jsou převedeny na mezery (SPACE). Ukončovací znak #9B je na diagramu označován jako RETURN, znak LF značí nový řádek.

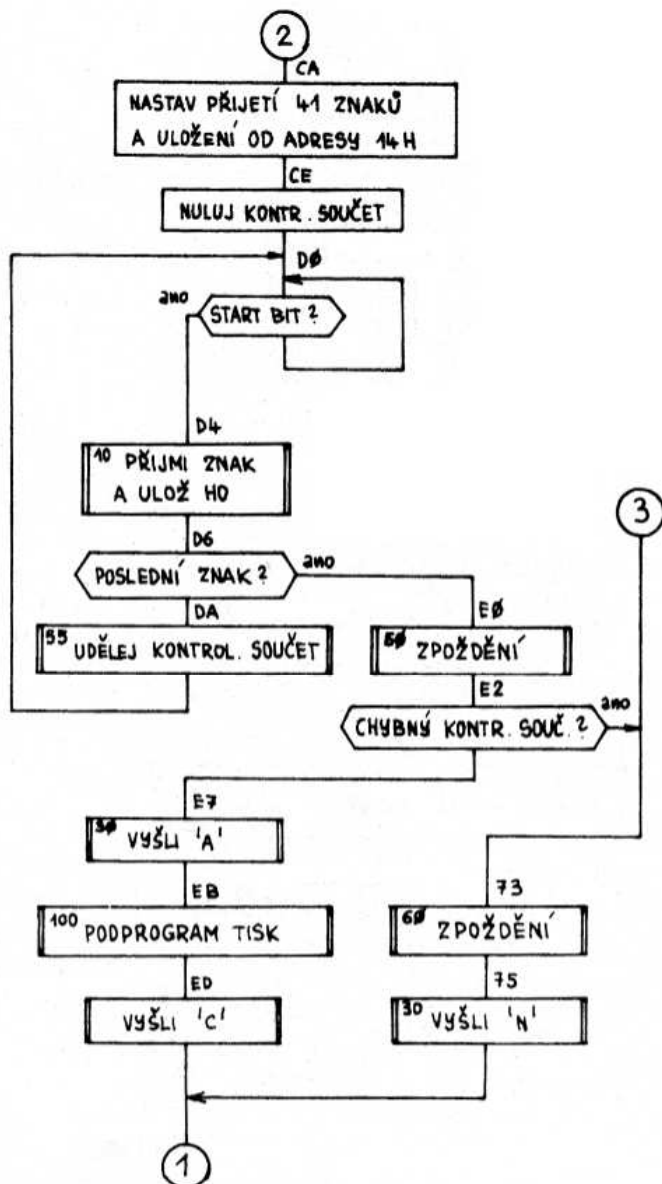
Rozhraní je navrženo s mikroprocesorem MH 80 35, program je uložen v paměti EPROM 2716. Zařízení je napájeno z tiskárny (cca 200 mA je pro ATARI mnoho). Zapojení konektoru tiskárny je patrné z obr.28. Použitou frekvenci krystalu 6 MHz je nutno dodržet, protože z ní je odvozeno časování při příjmu a vyslání sériových dat.

Tabulka 1

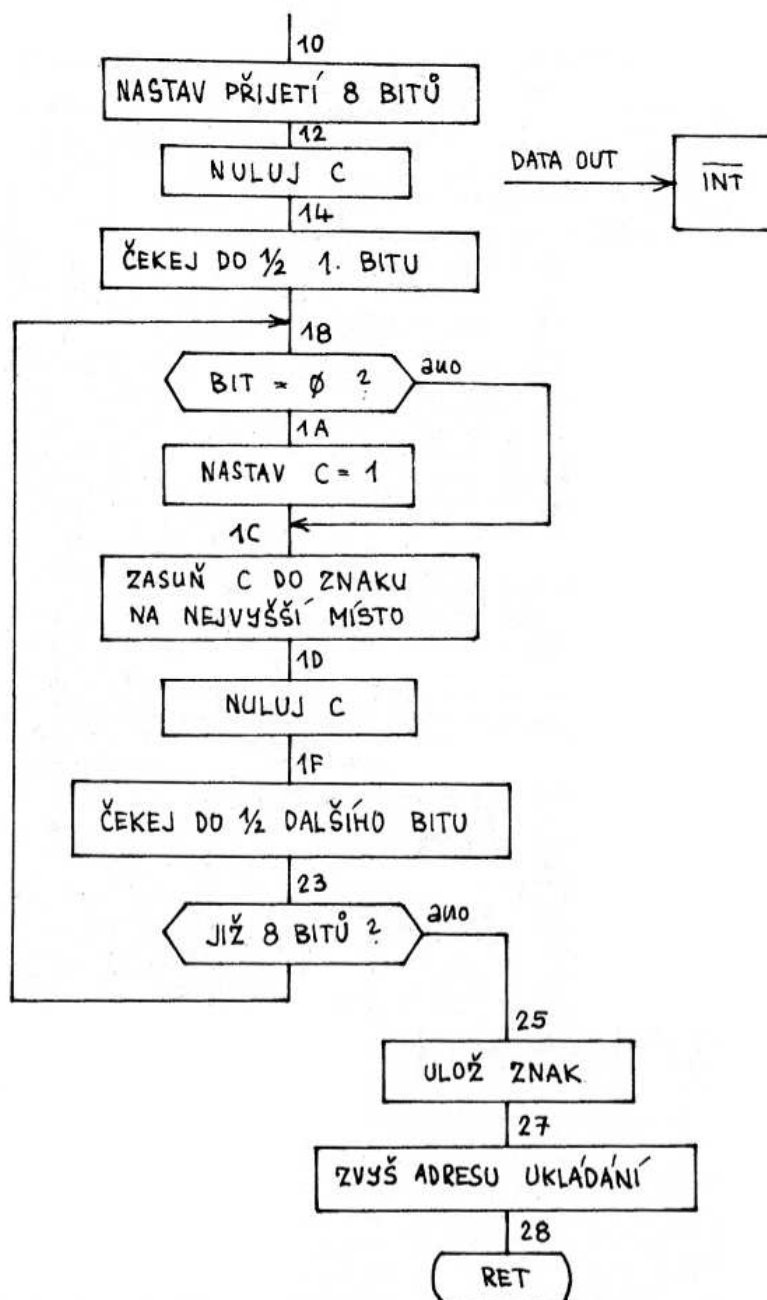
000:	04	70	00	00	00	00	00	00	00	00	00	00	00	00	00	00
010:	BA	08	97	00	B9	07	E9	16	86	29	A7	00	67	97	00	B9
020:	05	E9	21	EA	18	AC	A0	18	83	04	1C	00	00	00	00	00
030:	9A	7F	BA	08	B9	07	E9	36	FC	77	F2	4D	9A	7F	00	00
040:	B9	05	E9	42	EA	39	8A	80	B9	07	E9	4A	83	BA	80	04
050:	40	00	00	00	00	FC	6F	E6	5A	17	AF	83	00	00	00	00
060:	23	90	16	64	62	55	16	6A	04	66	65	83	00	00	00	00
070:	8A	FF	0A	D2	72	BB	05	B8	14	27	AF	86	7F	04	7B	14
080:	10	EB	85	04	89	14	55	04	7B	0A	D2	8E	04	89	C8	F0
090:	DF	96	F3	B8	14	F0	03	C0	C6	9C	04	70	18	F0	03	AD
0A0:	96	C2	BC	41	14	30	14	60	BC	43	14	30	BC	00	14	30
0B0:	BC	00	14	30	BC	14	14	30	BC	29	14	30	BC	3D	14	30
0C0:	04	70	03	FC	96	F3	BC	41	14	30	BB	29	B8	14	27	AF
0D0:	86	D4	04	D0	14	10	EB	DA	04	E0	14	55	04	D0	00	00
0E0:	14	60	C8	F0	DF	96	F3	BC	41	14	30	24	00	BC	43	14
0F0:	30	04	70	14	60	BC	4E	14	30	04	70	00	00	00	00	00
100:	BB	28	B8	14	F0	53	7F	A0	03	E5	C6	29	03	FB	F6	14
110:	23	20	24	1D	03	C0	F6	1B	F0	24	1D	03	40	34	40	EB
120:	04	04	ED	23	0A	34	40	04	ED	00	00	00	00	00	18	F0
130:	03	65	C6	21	24	23	00	00	00	00	00	00	00	00	00	00
140:	97	F7	37	39	99	FE	36	46	23	01	39	18	83	00	00	00



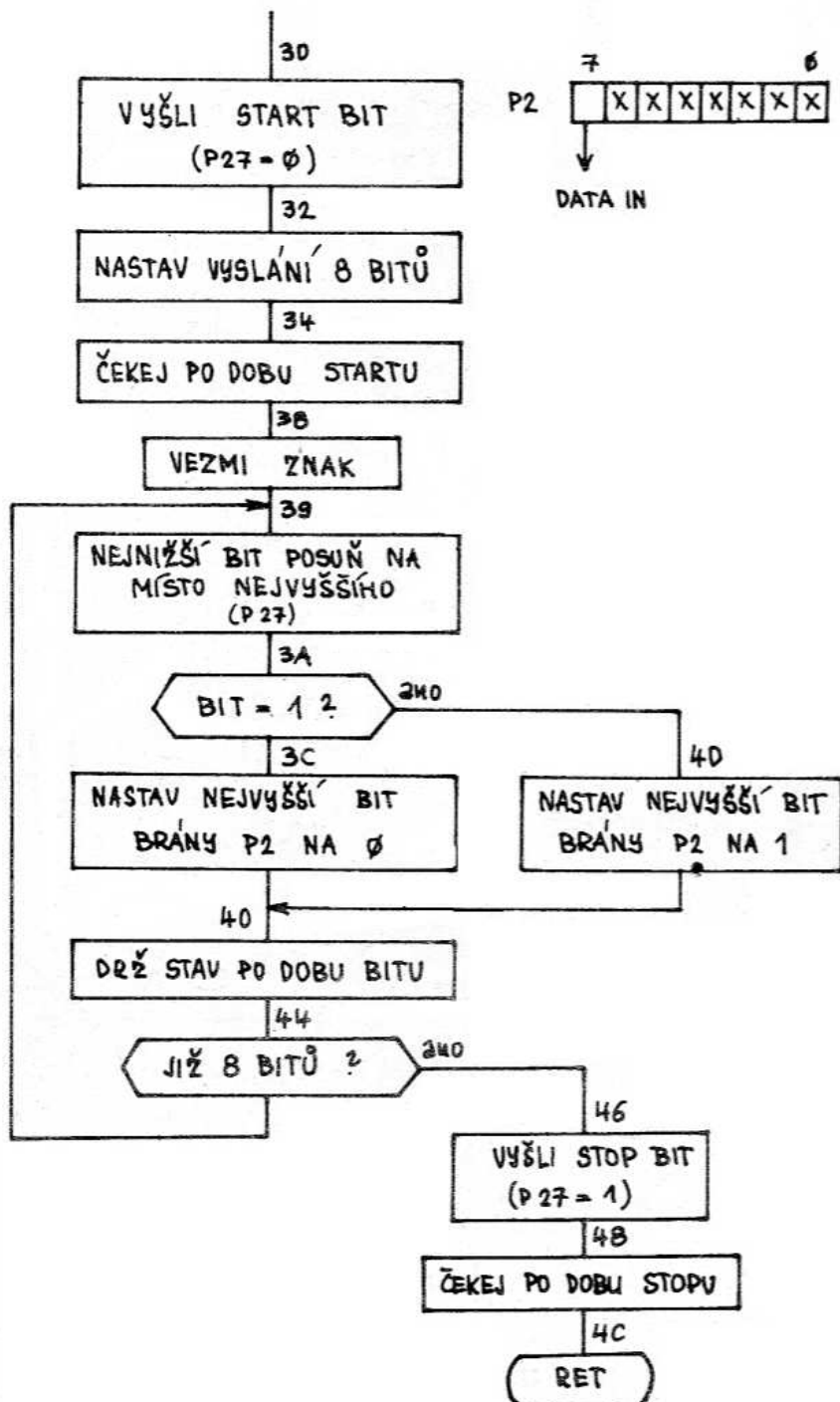
Obr. 22 Hlavní program - část 1



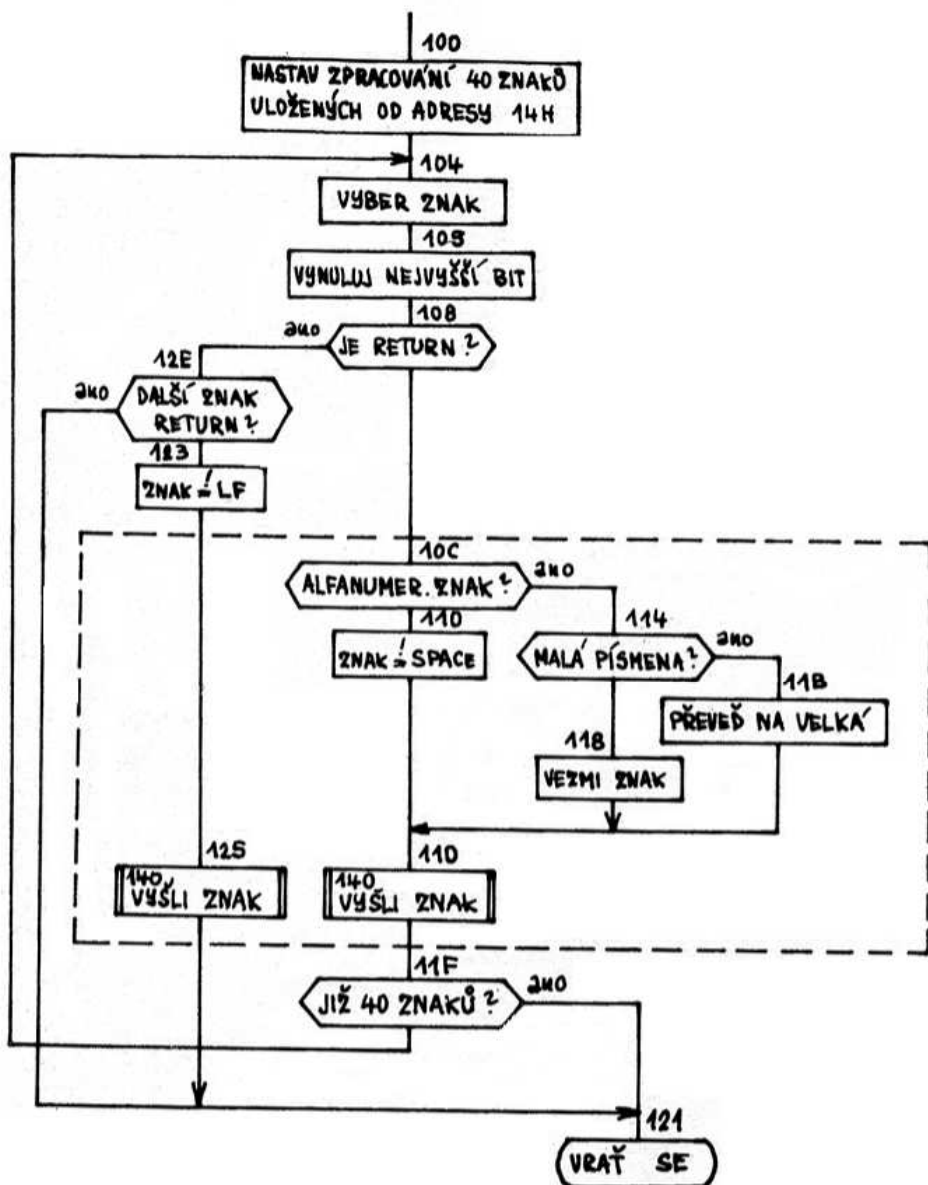
Obr. 23 Hlavní program - část 2



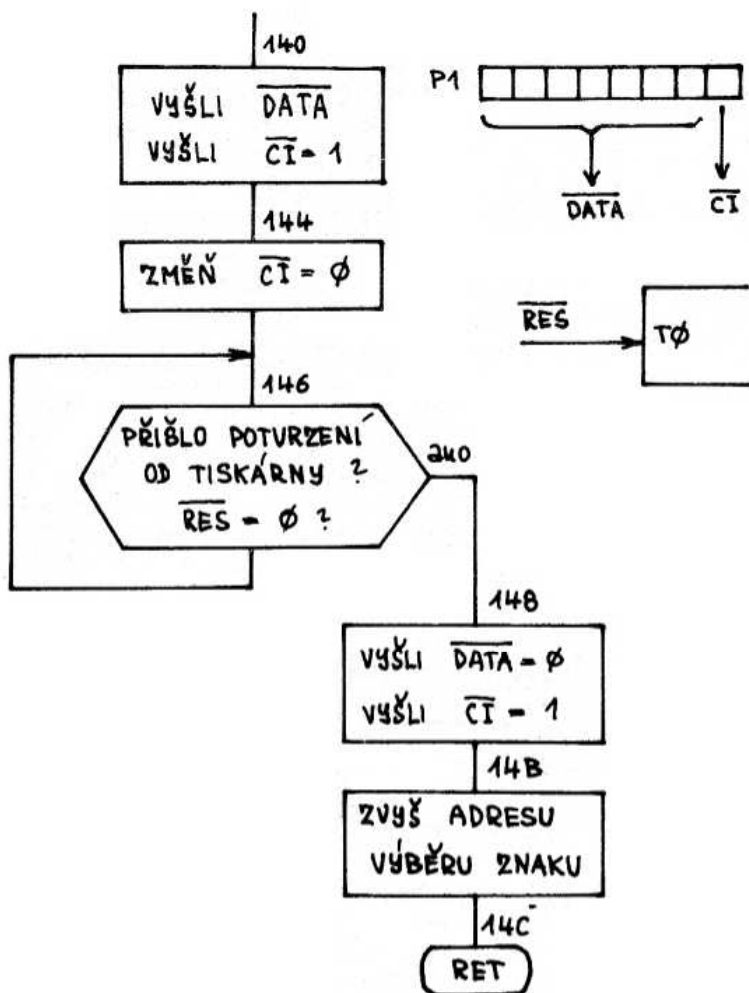
Obr. 24 Příjem a uložení znaku



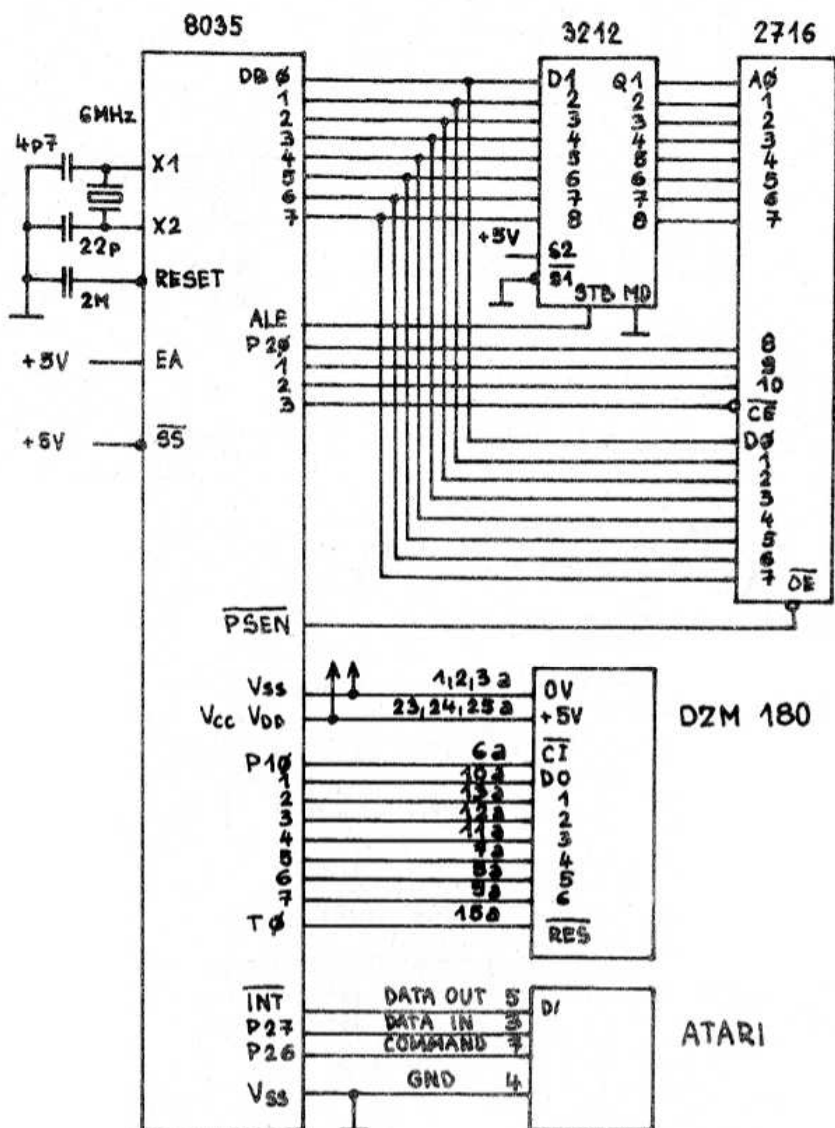
Obr. 25 Podprogram vyslání znaku do počítače



Obr. 26 Podprogram TISK



Obr. 27 Podprogram obsluhy tiskárny (vyslat znak)



Obr. 28 Zapojení rozhraní

V další části této kapitoly uvedeme řešení ing. Jaroslava Chochlíka a RNDr. Jána Sládka ze Žiliny */, využívající rovněž jednočipový mikropočítač 8035. Popsaný modul umožňuje kromě připojení libovolné tiskárny s rozhraním CENTRONICS, IRPR rovněž připojení magnetofonu se zvýšenou přenosovou rychlostí, které bude popsáno dále.

*/ Zpravodaj ATARI KLUB Olomouc, 1988, 5-6, s.129-193. Převzato se souhlasem redakce.

Poznámka: Jedná se o řešení odlišné od systému TURBO 2000, resp. SUPER TURBO autora Jiřího Richtera.

Modul pro připojení tiskárny k počítači ATARI

Komunikační modul pro připojení tiskárny k počítačům ATARI umožňuje připojení libovolné tiskárny s rozhraním podle normy CENTRONICS nebo IRPR. Výběr normy rozhraní se provede přepínačem SW5 umístěným v modulu.

Tiskárna se připojí na komunikační modul prostřednictvím konektoru "TISKARNA" (CANNON 25 pin) s rozložením signálů typickým pro tiskárny PRT, vyráběné družstvem Datacoop, MLR.

Následující tabulka obsahuje rozložení signálů na konektory "TISKARNA" podle typu rozhraní:

pin č.	CENTRONICS	IRPR
1	AUTOLF	AUTOLF
4	BUSY	AC
5	STROB	SC
6	DATA 8	D 7
7	DATA 7	D 6
8	DATA 6	D 5
9	DATA 5	D 4
10	DATA 4	D 3
11	DATA 3	D 2
12	DATA 2	D 1
13	DATA 1	D 0
17	(+5V)	(+5V)
21-25	GND	GND

Popis modulu pro připojení tiskárny

Základní úlohou modulu je převést data ze sériového portu počítače ATARI do paralelní datové sběrnice rozhraní tiskárny a poskytnout potřebné řídicí signály. Ve skutečnosti musí zabezpečit komunikaci v poměrně složitém komunikačním protokolu mezi počítačem ATARI a tzv. inteligentní periférií, což jsou pro počítače ATARI všechny standardní periférie mimo kazetového

magnetofonu.

Proto je jádrem tohoto modulu jednočipový počítač typu 8048 (8035). Dále tento modul obsahuje EPROM paměť typu 2716 (2 Kbyty), IO typu 74LS373, krystal 6MHz, konektor CANNON 25 pin ("TISKARNA"), konektor pro sériový port ATARI, sadu mikropřepínačů pro výběr funkcí a mikropřepínač pro inicializaci. Tyto součástky jsou osazené na navrhnutém plošném spoji. Napájení (jen +5V) je volitelné z počítače nebo z tiskárny; POZOR! Nikdy nesmí být zapojeny obě možnosti najednou!!! Běžně se bude využívat jen napájení z počítače, ačkoliv kvůli ochraně jeho zdroje před přetížením by bylo vhodné použít zvláštní pomocný zdroj, z něhož by bylo možné napájet i magnetofon (ATARI dataset apod.).

Tento modul je možné použít i na autonomní testování tiskárny s rozhraním CENTRONICS nebo IRPR - tedy bez dat z počítače ATARI. Potom se však musí použít napájení z tiskárny (jestliže ho tiskárna na svém vstupním konektoru poskytuje), nebo ze zvláštního zdroje +5V. V tomto režimu se na tiskárně vytiskne text stručně popisující funkci a doporučené nastavení mikropřepínačů modulu. Text se do tiskárny posílá poprvé s osmým bitem vynulovaným a podruhé s osmým bitem nastaveným do logické jedničky. Část tohoto textu je uvedena v následující tabulce:

SW TAPE	PRINTER
1 C-BOOT	0!
2 0!	0!
3 S0-S	0-CR AFT.EOL
4 S1-E A	0-LF AFT.EOL
5 S2-L T	0-CENT/C-IRPR
6 S3-. E	C-TEST
7 0!	C!
8 NC	AUTOLF

Pro tiskárnu mají význam jen informace uvedené ve sloupci PRINTER. K jednotlivým přepínačům:

- SW1 - pro tiskárnu musí být vždy jeho kontakt nespojený (0=OPEN)
SW2 - jako SW1
SW3 - jestliže necháme jeho kontakt nespojený, vysílá se do tiskárny CARRIAGE RETURN po přijetí EOL v textu
SW4 - když necháme jeho kontakt nespojený, posílá do tiskárny LINE FEED po přijetí EOL v textu
SW5 - nespojený kontakt zabezpečí činnost rozhraní CENTRONICS
- spojený kontakt (C-CLOSE) zabezpečí činnost rozhraní IRPR
SW7 - pro tiskárnu musí být vždy kontakt spojený - CLOSE!
SW8 - je použitelný pouze pro tiskárny, které mají funkci AUTO LINE FEED ovládanou přes konektor TISKARNA.

Upozornění!

I když architektura tohoto modulu je velmi úsporná a pro čtení přepínačů SW3 až SW6 se využívá datová sběrnice paralelního rozhraní tiskárny, je třeba tyto přepínače (SW3 až

SW5) po inicializaci vypnout. Přesněji - jeden z přepínačů by mohl být trvale zapnutý - a v praxi se tato možnost asi nejčastěji využije pro rozhraní IRPR (SW5 spojený), nikdy však nesmí být spojeny najednou dva a více přepínačů SW3 až SW6! Po inicializaci se přečtou přepínače a jejich obsah zůstane uchován až do následující inicializace. Potom má obsluha k dispozici asi 7 sekund na rozpojení přepínačů. Proto je i výpis textu v režimu autonomního testu o tento čas po inicializaci opožděný.

Modul se inicializuje zapnutím napájení, stlačením inicializačního tlačítka (mikropřepínače) nebo i poklesem napájení, a proto se musí spolehlivě zabezpečit napájení modulu a jeho filtrování.

Vzhledem k tomu, že pro styk s tiskárnou se používají porty vlastního mikropočítače, je třeba zacházet s tímto modulem velmi opatrně. POZOR hlavně na STATICKOU ELEKTRINU! Taktéž by se neměly používat dlouhé kabele (max. 3 metry) a tiskárna se zakončovacími rezistory s malým odporem; tyto rezistory se připájejí na +5V a jejich hodnota by neměla klesnout pod 470 hm. Např. ve velmi rozšířených tiskárnách ROBOTRON K 6310 až 6316 je třeba ošetřit všechny datové a použité řadičí signály tak, že v modulu rozhraní IFSP se původní rezistory odstraní a zapojí se nové rezistory, připojené na +5V, přičemž jejich hodnota by neměla být menší než 470 hm. Toto je třeba prověřit i pro jiné tiskárny, avšak většina moderních tiskáren může být buzena z portů mikropočítače typu 8048 přímo.

Programová slučitelnost s operačním systémem ATARI

Modul simuluje standardní tiskárnu pro počítače ATARI 800 XL, 130 XE apod. Pro uživatele je přístupný jako zařízení se symbolickým jménem "P:". Je možné používat jej ve spolupráci s BASICem i jinými programy, které umí využívat systémovou tiskárnu.

Obsluha modulu pro připojení tiskárny

Při běžném používání modulu pro připojení tiskárny se připojí kabel tiskárny do konektoru "TISKÁRNA" a kabel s konektorem pro spojení s ATARI se zasune do konektoru "PERIPHERAL" počítače ATARI. Po nastavení přepínačů na modulu je možné zapnout napájení počítače. Jestliže jsou více jak dva z přepínačů SW3 až SW4 spojené, má obsluha asi 7 sekund na jejich rozpojení. Potom je možné tiskárnu používat. Magnetofon ATARI by neměl být připojený na konektor PERIF2!

Jestliže potřebujeme změnit funkci modulu, je potřeba zastavit výpis, přepnout přepínače do požadované polohy, inicializovat modul a případně znovu rozepnout přepínače SW3 až SW6.

V režimu autonomního testování je obsluha podobná, lišit se může pouze při připojení na počítač ATARI v tom případě, že nevyužíváme jeho zdroj i na napájení modulu.

Schéma zapojení a konstrukční pokyny jsou uvedeny v závěru kapitoly 5.2.

5.2.2. Připojení páskové jednotky

Jak jsme se zmínili v předchozí kapitole, popsaný modul lze využít pro připojení magnetofonu, pracujícího se zvýšenou přenosovou rychlostí.

Při vývoji představovaného systému se zrychleným přenosem údajů mezi počítačem ATARI a magnetofonem jsme si dali za cíl dosáhnout maximální míru slučitelnosti s původním programovým vybavením a co možno nejvíce využít existujících technických prostředků počítače ATARI. Dále jsme se snažili co nejvíce odstranit rušivé efekty, známé z jiných podobných systémů, především rozsynchronizování, resp. úplná ztráta obrazu po dobu přenosu dat. Prakticky všechny cíle se nám podařilo splnit.

Představovaný systém zrychleného přenosu dat používá pro záznam metodu dvojnásobné frekvence, avšak dekodování při čtení dat je odlišné od průmyslově používaného způsobu. Je to dáno tím, že na rozdíl od průmyslového standardu, kde se úvod záznamu synchronizuje skupinou nul, je pro ATARI a jeho asynchronní způsob přenosu dat potřebné opatřit úvod záznamu spojitým signálem na úrovni logické jednotky. Každý bit dat se zaznamenává na kazetu následovně:

- logická nula je zaznamenána jednou periodou frekvence F
- logická jednička je zaznamenána dvěma periodami frekvence $2F$.

Pro přenosovou rychlost přibližně 4000 Bd má frekvence F hodnotu 3958,6 Hz a frekvence $2F$ hodnotu 7917,2 Hz.

Pro přenosovou rychlost přibližně 5300 Bd má frekvence F hodnotu 5278,1 Hz a frekvence $2F$ hodnotu 10556,2 Hz.

Pro přenosovou rychlost přibližně 8000 Bd má frekvence F hodnotu 7917,2 Hz a frekvence $2F$ hodnotu 15834,4 Hz.

Podstatou zvýšení spolehlivosti přenosu dat je změna řízení vysílání dat z ATARI prostřednictvím IO "POKEY". Originální způsob vysílání, který se snažil co nejpřesněji dosáhnout přenosovou rychlost 600 Bd, používá pro generování této rychlosti registry IO "POKEY" AUDF3 a AUDF4, spojené do jednoho 16 bitového čítače. Takto vytvořený generátor však není synchronní s generátory odvozenými z registrů AUDF1 (pro log.1) a AUDF2 (pro log.0). Aby se dosáhl synchronismus všech tří generátorů, musí se "POKEY" ovládat tak, že přenosová rychlost je dána jen 8-bitovým registrem AUDF4. Jednotlivé registry se potom programují následovně:

rychlost AUDF1 AUDF2 AUDF4 AUDCTL

1320	5	7	17	0
4000	3	7	7	0
5300	2	5	5	0
8000	1	3	3	0

Připomeňme si nyní některé další parametry tohoto záznamu. Už pro přenosovou rychlost 4000 Bd se dosahuje hustota záznamu asi 84 bitů na milimetr. Je to víc, než u nejrozšířenějšího průmyslového standardu PE 1600, kde je hustota záznamu asi 63 bit/mm. Přitom v průmyslových počítačových páskových systémech

se používají nosiče magnetických vrstev s několikanásobně větší tloušťkou a tedy i pevností, kde pásková dráha se nastavuje s přesností na tisíce milimetru, magnetické medium se před příchodem na magnetofonovou hlavu čistí na čistících břítech a aktivita vrstvy media prakticky nepřichází do styku s hnacím mechanismem.

U kazetových datových magnetofonů je však situace jiná. Například jen kolísání rychlosti dosahuje často více než 1 procento a rozptyl rychlosti na různých magnetofonech je ještě větší. Proto nejvyšší uvedené rychlosti jsou jen těžko použitelné pro běžné datasety.

Při simulaci datasetu jiným počítačem ATARI a ověřování schopnosti původního programového vybavení a technických prostředků ATARI jsme zkoušeli jejich spolehlivou činnost bez náročných úprav, až do přenosové rychlosti 10000 Bd. Při ověřování v praxi se skutečným datovým magnetofonem XC 12 jsme dosáhli spolehlivou činnost při rychlosti 4000 Bd.

Záznam dat ani při nejvyšších rychlostech nepředstavuje žádné problémy. Dekodér záznamu je teoreticky schopen pracovat až do nejvyšších uvedených rychlostí. Pro zvýšení spolehlivosti pro vyšší přenosové rychlosti by však bylo potřebné použít magnetofon s podstatně lepšími parametry. Proto jsme se rozhodli používat přenosovou rychlost 4000 Bd.

Popis potřebných úprav pro zrychlený přenos dat

Zrychlený přenos dat mezi počítačem a magnetofonem vyžaduje:

- úpravu operačního systému ATARI,
- úpravu magnetofonu,
- vytvoření dekodéru záznamových dat.

Úprava operačního systému ATARI

Na rozdíl od jiných systémů zrychleného přenosu dat nevyžaduje náš systém vytvářet nové ovladače pro kazetovou pásku. Využívá se programového vybavení, aby se co nejvíce zachovala slučitelnost s původním programovým vybavením. Mění se jen některé konstanty v původním operačním systému. Úpravy se provádějí v kopii operačního systému. Celou činnost je možné shrnout do následujících bodů:

- standardní rychlostí se do počítače ATARI zavede krátký program a spustí se. Tento program zabezpečí:
 - zkopírování operačního systému ATARI z ROM do RAM,
 - změnu parametrů operačního systému důležitých pro zrychlení přenosu dat,
 - aktivování zaváděcího programu pro zavedení programů zvýšenou rychlostí.

(Změněný operační systém zůstává v RAM i po stisknutí RESET.)

- když používáme programy uložené na kazetě, spustíme jejich zavádění stisknutím libovolné klávesy
- když chceme použít BASIC, stiskneme RESET a příkazem POKE 54017,252 uvedeme do činnosti upravený operační systém v RAM.

Celý postup je podrobně popsán v části týkající se software. Připomínáme, že by bylo ideální provést změnu operačního systému v ROM (EPROM) paměti počítače ATARI.

Úprava magnetofonu

V datovém magnetofonu, například XC 12, je potřebné vyřadit frekvenční omezení vstupního dvojtranzistorového zesilovače odstraněním kondenzátoru C6 (470 pF), případně nahradit tento kondenzátor jiným s kapacitou max. 100 pF v případě, že by se vstup vlivem velkého zesílení rozkmitával. Na vstup prvního operačního zesilovače (pin číslo 8 IO LM324) se připojí jednoduchý tranzistorový tvarovač, kterého výstup se vyvede na pin číslo 13 konektoru PERIPHERAL pomocí vodiče původně určeného pro signál AUDIO IN (červený vodič, pin číslo 11). Z výstupu tvarovače se potom buď signál AUDIO IN přes kondenzátor 22 až 100 pF, takže z televizoru uslyšíme skutečná (vytvářovaná) nosná data a ne přeslech z vedlejší stopy. V záznamové větvi je vhodné změnit odpor R42 10k na hodnotu až asi 3k3 a kondenzátor C22 4n7 na hodnotu asi 2n2. Individuálně je možné tyto hodnoty optimalizovat v konkrétním datovém magnetofonu.

podle používaného materiálu (kazety), aby nedošlo k přebuzení magnetického materiálu.

Dekoder na mikroprocesorovém subsystému

Možnost realizovat dekodeur tímto způsobem jsme zvolili po vytvoření modulu pro připojení tiskárny. Toto řešení má několik výhod. Dosáhnou se všechny stanovené parametry (a elešance řešení), přitom se využije modul, který není úplně vyžitý a není jej potřebné zvláště upravovat. Modul obsahuje EPROM typu 2710 s kapacitou 2 kB, ve které však data pro ovládání tiskárny zabírají jen první polovinu. Do druhé poloviny je možné vložit instrukce a data pro dekodeur dat z magnetofonu a jak uvidíme dále, i další užitečné informace. Režim práce se potom přepíná přepínačem SW7, který vybírá horní nebo dolní polovinu paměti a následnou (nebo úvodní) inicializaci. Data z magnetofonu (nejen tedy ze speciálního datového magnetofonu) vytvářené do logických úrovní TTL se přivedou na vstup mikropočítače 8048, totožného se vstupním signálem BUSY, kterým oznamuje tiskárna připravenost k převzetí dat z rozhraní. Komunikační signály modulu s počítačem ATARI zůstávají nezměněné a v tomto případě se z nich využívá jen signál DATA IN (pin číslo 3) a napájecí vodiče. Mění se však význam přepínačů. Jejich popis je uveden v tabulce:

SW TAPE	PRINTER
1 C-BOOT	O!
2 O!	O!
3 S0-S R	O-CR AFT.EOL
4 S1-E A	O-LF AFT.EOL
5 S2-L T	O-CENT/C-IRPR
6 S3-. E	C-TEST
7 O!	C!
8 NC	AUTOLF

Pro dekodeur mají význam jen informace uvedené ve sloupci TAPE. K jednotlivým přepínačům:

SW1 - když je zapnutý (CLOSE), po inicializaci se posílá do počítače ATARI zaváděcí program (BOOTSTRAP)

SW2 - musí být vždy rozpojený

SW3 - S0 - nejnižší bit pro výběr rychlosti přenosu dat

SW4 - S1 - bit pro výběr rychlosti přenosu dat

SW5 - S2 - bit pro výběr rychlosti přenosu dat

SW6 - S3 - nejvyšší bit pro výběr rychlosti přenosu dat

SW7 - pro režim dekodér musí být kontakt vždy rozpojený (0=OPEN)

SW8 - pro režim dekodér není využitý.

Režim automatického vysílání zaváděcího programu do počítače ATARI se aktivuje inicializací modulu (stisknutím inicializačního tlačítka, náběhem napájecího napětí), při současně spojeném přepínači SW1. Je to velmi užitečná funkce tohoto modulu, protože uživatel nemusí zaváděcí program nahrávat z magnetofonu a vyhledávat jej na pásce a následně měnit kazetu. Přenos dat (jsou to dva standardní bloky po 128 bytech a ECF blok) se uskutečňuje rychlostí asi 820 Bd. Celá činnost se podobá zavádění z datového magnetofonu, avšak místo zakládání pásky se zaváděcím programem, který nám zabezpečí zkopírování ROM operačního systému do RAM, jeho změnu a opětovný start zaváděcího programu (nyní však už rychlostí 4000 Bd) a manipulaci s datovým magnetofonem nám stačí inicializovat modul dekodéru. V magnetofonu můžeme mít připravený program zaznamenaný rychlostí 4000 Bd a i při chybě v průběhu zavádění manipulujeme pouze s úsekem pásky v kazetě, kde je zaznamenaný požadovaný program. Když se vyskytne v průběhu zavádění chyba (BOOT ERROR) a došlo i k vypadnutí zaváděcího programu, náprava se provede velmi elegantně - přípravou ATARI (jako pro kazetu, tj. RESET se současným stisknutím START+OPTION) a inicializací modulu dekodéru. Zaváděcí program pro rychlost 4000 Bd verze 2.0 a vyšší se však až do přepsání jiným zaváděcím programem, který je často součástí herních i uživatelských programů, po výskytu chyby sám restartuje.

K vlastnímu přenosu dat zaváděcího programu dochází asi 15 s po inicializaci. Toto zpoždění je nutné kvůli zachování kompatibility s neupraveným operačním systémem ATARI a kvůli poskytnutí času obsluze na aktivování zavádění (podržení stisknutých tlačítek START+OPTION a následné stisknutí klávesy, např. RETURN).

Přepínače SW3 až SW5 jsou v tabulce 2 označené symboly S0 až S3. Těmito přepínači se převádí korekce zpožděné konstanty v dekodéru pro různé přenosové rychlosti. Celkové zpoždění vzorkování v dekodéru je možné vyjádřit vzorcem:

$$T = (S_3 \cdot 2^3 + S_2 \cdot 2^2 + S_1 \cdot 2^1 + S_0 \cdot 2^0) \cdot 5 (\text{mksek}) \quad (1)$$

Symboły S0 až S3 nabývají hodnoty 1, když jsou přepínače SW3 až SW6 rozpojené (O=OPEN) a hodnoty 0, když jsou tyto přepínače spojené (C=CLOSE).

Pro přenosovou rychlost 4000 Bd je teoreticky nejvhodnější hodnota zpoždění 95 mikrosekund. Takového zpoždění dosáhneme v souladě se vzorcem (1) tak, že výraz uvnitř závorek bude rovný číslu 19, tedy přepínače musí vygenerovat číslo 10. Toho se dosáhne následující konfigurací přepínačů:

přepínač:	SW6	SW5	SW4	SW3
stav :	Open	Close	Open	Close
hodnota :	8	0	2	0

Hodnota 95 mikrosekund je teoretická a v praxi ji má možnost uživatel korigovat v poměrně širokém rozsahu.

Tímto dekodérem je možné dekodovat (demodulovat) i standardní záznam rychlostí 600 Bd. Teoretická hodnota zpoždění je 110 mikrosekund a potom musí tyto přepínače vygenerovat číslo 13 ($110/5=22$ a $22-9=13$). Tuto konfiguraci zobrazuje následující zápis:

přepínač:	SW6	SW5	SW4	SW3
stav :	Open	Open	Close	Open
hodnota :	8	4	0	1

Takováto jistá teoretická konfigurace přepínačů je vhodná i pro rychlost 1320 Bd, protože nosné frekvence jednotlivých logických úrovní jsou pro obě přenosové rychlosti stejné. Samozřejmě, různé magnetické materiály (pásky), nastavení magnetofonu a formátování může vyžadovat korekce. Tuto účinnou flexibilitu dekodéru můžeme s výhodou využít při čtení pásky, která byla nahrána na magnetofon s odlišnou magnetickou rychlostí.

Nulová hodnota z přepínačů SW3 až SW6 poskytuje zpoždění 45 mikrosekund. Tato hodnota by byla teoreticky vhodná pro dekodování záznamu rychlostí asi 10000 Bd. Data zaznamenávaní takovouto rychlostí však žádný běžný kazetový magnetofon nedokáže spolehlivě reprodukovat.

Z praktických zkušeností jsme dospěli k následujícím poznatkům. Není ztracený záznam, který se nedá reprodukovat na poprvé. Korekce pomocí přepínačů jsou velmi účinné, hlavně když se reprodukuje záznam vytvořený na jiném magnetofonu. Je však vhodné na zrychlení přenosu dat si vyhradit podle možnosti jenom jeden typ kazet co možná nejkvalitnějších. Magnetofon je nutné udržovat v maximální čistotě. Nikdy se nesmí nechávat stlačené tlačítko PLAY po skončení záznamu nebo čtení - gumová kladka může nenávratně zničit nosič. Přímé mechanické poruchy na nosiči nejsou vždy vidět pouhým okem. Na zničení nosiče stačí mechanický drop-out (výpadek) magnetické vrstvy s rozměrem řádově nižším než délka nosiče, na kerém je zaznamenaný jeden bit - asi dvanáct tisícín milimetru. Toto mějte na paměti i při práci s nižšími přenosovými rychlostmi! Vyhnete se mnohým nepříjemným překvapením.

Obsluha mikroprocesorového dekodéru

Jestliže používáme běžný magnetofon ATARI, např. XC 12,

připojíme jej přes konektor "PERIF22", který je funkční podmnoužinou konektoru "PERIPHERAL". Přitom na konektor "TISKARNA" nesmí být připojen kabel tiskárny, aby neovlivňoval úroveň signálu BUSY. Přepínače na dekodéru nastavíme do požadované polohy a provedeme inicializaci. Je-li potřebné zavést program BOOTSTRAP z modulu, provedeme jej dle výše popsaného postupu pro tuto činnost. Pak se používání dekodéru neodlišuje od používání běžného magnetofonu, dokonce ani při různých přenosových rychlostech. Jen v případě, že se nám nepodaří zavést data z magnetofonu do počítače ani na opakovaný pokus, pokusíme se zlepšit zpoždění vzorkování přestavováním přepínačů SW3 až SW6. Používáním dekodéru se nám poměrně rychle podaří najít optimální nastavení těchto přepínačů pro náš vlastní magnetofon a používaný záznamový materiál.

Jestliže používáme jiný magnetofon, je třeba jej vybavit tvarovačem a výstup tohoto tvarovače přivést na vstup BUSY konektoru "TISKARNA". Až na komfort obsluhy je práce shodná s prací magnetofonu.

Konstrukční pokyny a schéma úpravy magnetofonu jsou uvedeny v závěru kapitoly 5.2.

5.2.3. Sériová komunikační linka

Z dílny autorů ins. J. Chochlíka a RNDr. J. Sládka pochází i modul na připojení sériové komunikační linky k počítači ATARI, který umožňuje napojení jiného počítače s přenosovou rychlostí 600 až 9600 Bd za podpory programu typu KERMIT. S modulem je možné porovnat i další sériová přídavná zařízení, pracující s přenosovou rychlostí v rozsahu 600 - 9600 Bd. Modul pracuje s proudovou smyčkou 0/20 mA (odpovídá standardu IRPS).

Přenosová rychlost sériové komunikační linky je programově volitelná a běžně je možné využívat standardní rychlosti 600, 1200 ... 9600 Bd. Není problémem spektrum rychlostí rozšířit.

Modul pro připojení sériové komunikační linky pracuje na straně venkovního rozhraní s proudovou smyčkou 0/20 mA, na straně vnitřního rozhraní (připojení k ATARI) s úrovněmi LS TTL.

Z uvedeného vyplývá, že modul tvoří obousměrný převodník: proudová smyčka 20mA/LS TTL. Strana pro připojení k počítači ATARI (LS TTL) je vybavena i blokovacím vstupem. Pro blokování převodníku slouží signál MOTOR CONTROL, který zabezpečuje bezkonfliktní spolupráci převodníku a ATARI masnetofonu. V případě, kdy modul používá na zavedení programového vybavení do paměti počítače systém "SHORA DOLU" (DOWNLINE LOADING), je třeba blokování signálem MOTOR CONTROL zrušit (odpojením blokovacího vstupu od počítače). Prevodník poskytuje vstup i výstup proudové smyčky v pasivním stavu a bez regulace proudu. V případě, že se modul použije na spojení se zařízeními, které mají proudové linky pasivní, je třeba použít venkovní zdroj napětí (např. z přídavného zařízení). V případě, že ani druhá strana proudové smyčky nemá regulátor proudu, je třeba proud v smyčce omezit (např. statickým sériovým odporem). Rozložení signálů na konektoru proudové smyčky odpovídá rozložení signálů rozhraní IRPS pro počítače SMEP (je shodné např. s rozložením signálů na terminálech CM 7202, CM 7202.M1 apod.). Situaci popisuje následující tabulka:

číslo pinu	název signálu
9	přijímač +
10	přijímač -
18	vyšlač +
19	vyšlač -
5	napájení vstup
20	napájení výstup

Pro stranu proudové smyčky (tedy rozhraní IRPS) je použit konektor CANNON 25 pinov. Piny číslo 5 až 20 se využívají na odpojení napájení v čase, kdy modul není připojený na komunikační linku a z toho vyplývá, že tyto piny musí být na konektoru z venkovní strany připojené přípojkou (je to běžné pro kabely počítačů a přídavných zařízení SMEP).

Spojení počítače ATARI s jiným počítačem

Pomocí modulu pro připojení sériové komunikační linky je možné realizovat spojení počítače ATARI s jiným počítačem (vybaveným rozhraním IRPS) velmi jednoduše. Např. pro spojení s

počítačem SMEP je možné využít běžnou terminálovou linku IRPS - tedy počítač ATARI se připojí místo terminálu SMEP (např. CM7202) a to tak, že kabel IRPS z terminálu zasuneme do konektoru IRPS modulu. Podpůrné programové vybavení v počítači ATARI (např. KERMIT, Emulátor VT52 apod.) obsluhuje sériovou linku tak, aby byla dodržena některá ze standardních (případně i jiných) přenosových rychlostí a formátů přenášených znaků. V tomto případě je formát znaků následovný:

- 1 START bit
- 8 DATA bitů
bez paritního bitu
- 1 STOP bit

Připojení přídavných zařízení se sériovým rozhraním k počítači ATARI

Pro připojení přídavných zařízení se sériovým rozhraním IRPS platí to, co bylo popsáno výše. Jediným možným problémem zůstává neschopnost rozhraní přídavného zařízení pracovat v aktivním režimu. Jestliže přídavné zařízení tuto možnost nemá, je třeba použít pomocný zdroj napětí pro proudovou smyčku.

Programová podpora sériové komunikační linky

Sériová komunikační linka je podporována speciálním programovým vybavením, jehož popis je v části SW. Za zmínku stojí operační systém ATARI, který se doplní o ovladač (HANDLER) se symbolickým jménem "R:". V některých aplikačních programech např. v KERMITu je však ovladač s tímto jménem nedostupný, protože byla zvolena přímější cesta k ovládání linky. V originálním systému se jménem "R:" označuje ovladač pro linku RS 232-C. Připomeňme, že na rozdíl od rozhraní RS 232-C (též V.24, resp. S2), který pracuje na venkovní straně s napětovými úrovněmi, rozhraní IRPS používá proudovou smyčku 0/20 mA. Rozhraní IRPS s proudovou smyčkou bylo upřednostněno před napětovým z důvodu většího rozšíření rozhraní IRPS v našich podmínkách. Proto případné použití počítače ATARI jako terminálu počítače (např. emulace terminálu VT52 pro počítače SMEP) je proudová smyčka vhodnější i z důvodu většího aktivního dosahu oproti přímému spojení pomocí napětových rozhraní.

Komentář k obrazové příloze a poznámky k realizaci

Je velmi pravděpodobné, že ne každý majitel počítače ATARI využije kompletní komunikační adaptér. Proto nejdříve provedeme realizaci jednotlivých částí a potom sestavení celého kompletu.

Nejsložitější a pro většinu uživatelů i nejužitečnější částí je modul pro připojení tiskárny. Jeho realizace by neměla dělat potíže středně zdatnému amatérovi - elektronikovi. Modul je postavený na plošném spoji, který byl navržen pro uchycení do krabičky s jedním rozměrem asi 72 mm zapájkováním, ale je možné uchytnit jej i jiným způsobem, např. přišroubováním. Plošný spoj byl původně navržen jen pro připojení tiskárny.

Chceme-li modul použít i jako dekodér dat z magnetofonu, musíme provést následující změny:

a) Doplnit odpor 30k pro adresní linku A10 EPROM paměti a spojit jej s přepínačem SW7 podle schématu na obr.29, tj. uzemnit opačný konec přepínače.

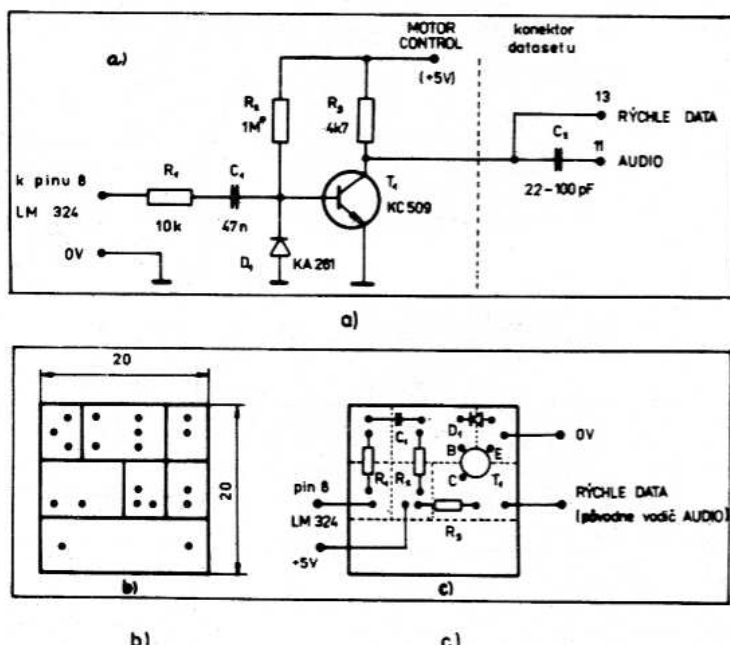
b) Připojit pin č.13 (RYCHLÁ DATA) konektoru PERIF2 na signál BUSY. Toto spojení je vhodné provést pomocí přepínače, který umožňuje připojení na vstup I/O mikropočítače (pin č.1) alternativně: buď signál BUSY z tiskárny (pin č.4 konektoru "TISKÁRNA"), nebo signál "RYCHLÁ DATA" z magnetofonu (pin č.13 konektoru "PERIF2"). Tímto je možné ponechat zapojené kabely z tiskárny i z magnetofonu současně, a výběr právě používaného zařízení provádět doplněným přepínačem. Chceme-li přepínač ušetřit, je třeba dodržet návod na používání, uvedený v části HW2 a HW3.

c) Chceme-li používat tiskárnu s rozhraním IRPR, je třeba spojit pin č.5 konektoru "TISKÁRNA" s pinem č.38 mikropočítače - podle obr. 29. Plošný spoj a firemní zařízení byly původně navrženy jen pro rozhraní CENTRONICS. Je-li zapojená drátová propojka "C" (CENTRONICS), dostane se na pin č.5 konektoru "TISKÁRNA" (STROBE) strobovací puls s délkou 1 mikrosekunda. Je-li zapojena drátová propojka "C+I" (CENTRONICS + IRPR), dostává se na pin č.5 pro rozhraní CENTRONICS puls s délkou 10 mikrosekund, a pro rozhraní IRPR puls, jehož délka závisí na potvrzení tiskárny (signálem AC). Drátové propojky ("C" a "C+I"), vyznačené v schématu na obr. 29, nejsou součástí plošného spoje a proto je třeba pro variantu C+I původní spoj přerušit (z pinu č.10 mikropočítače na pin č.5 konektoru "TISKÁRNA") a nový spoj realizovat z pinu č.38 mikropočítače na pin č.5 konektoru.

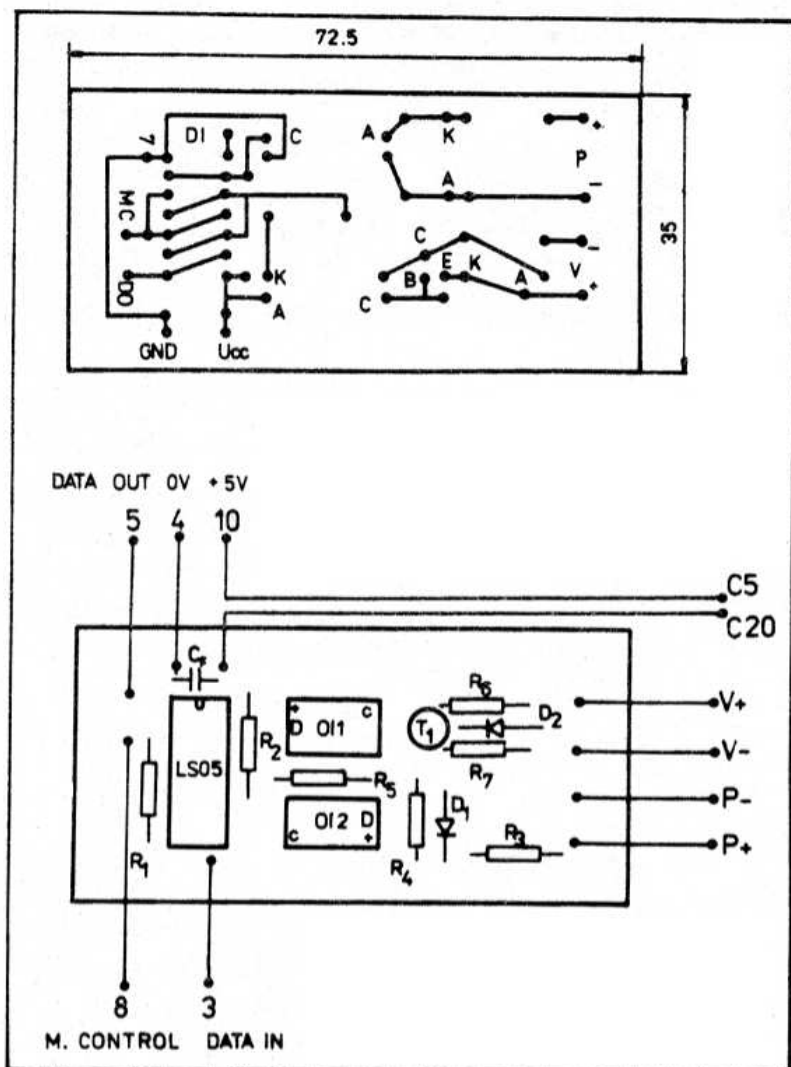
Realizace je zaručeně opakovatelná, neboť mimo přepínačů nejsou na modulu žádné nastavovací prvky.

V případě, že uživatel bude používat zrychlený přenos dat z magnetofonu do počítače, je třeba magnetofon upravit. (Není to potřebné pro rychlost 1320 Bd!) Úprava není náročná a nenaruší původní funkci magnetofonu.

Do magnetofonu se doplní tvarovač podle obr.30. Elektrické schéma je na obr.30 a), na obr.30 b) je zobrazen plošný spoj a na obr. 30 c) je zobrazeno jeho osazení součástkami. Z obrázku je patrná nenáročná a nenásilná úprava - představuje minimální náklady a v maximální míře využívá existující prostředky, přičemž zlepšuje i další funkci magnetofonu, včetně podpory pro zrychlený přenos dat - umožňuje totiž sluchovou kontrolu skutečné datové stopy, a ne přeslech z vedlejší zvukové stopy. Kritickou součástkou je odpor R_2 , jehož hodnota závisí od zesílení tranzistoru T1. Pro tranzistory s vysokým zesílením vyhovuje hodnota 1M, pro tranzistory s nižším zesílením je třeba hodnotu R_2 snížit, výjimečně až na 1M. Tvarovač tvaruje zesílený a předzpracovaný signál z magnetofonové hlavy na obdélníky se středem přibližně 1:1 pro vstupní signál konstatované frekvence, např. úvodní synchronizační signál. Destička tvarovače se umístí do magnetofonu a signál z něj se vyvede pomocí vodiče původně určeného pro signál AUDIO (červený vodič). Kondenzátor C2 se umístí do konektoru magnetofonu a jeho hodnotou můžeme nastavit odstup kontrolního příposlechu pomocí vstupu AUDIO z magnetofonu z modulační daty přijatými počítačem. Připomínáme, že modulace zvukového kanálu daty přijatými počítačem je volitelná funkce a některé programy ji záměrně programově potlačují. Pro požadovaný velký odstup je vhodná hodnota kondenzátoru C2 asi 22 pF, pro minimální odstup (a tedy pro velkou úroveň příposlechu) vyhovuje C2 rovně 100 pF. Tvarovač se napájí napětím asi +5 V ze signálu MOTOR CONTROL (zelený vodič). Tento tvarovač můžeme použít i pro zpracování signálu z komerčního magnetofonu (pozor na přenosové parametry!).



Obr. 30 Schéma zapojení tvarovače signálu



Obr. 31 a), b) Osazená deska a plošný spoj

ATARI PERIPHERAL	Modul tlačiarne	Modul sér. kom. linky	PERIF2	PERIF3
(1)	1	3	4 (4)	3 (3)
(2)	2	5	5 (5)	4 (4)
(3)	3	8	6 (6)	5 (5)
(4)	4	4	8 (8)	6 (6)
(5)	5	10	10 (10)	8 (8)
(6)	6	4	11 (11)	10 (10)
(7)	7	10	14 (14)	11 (11)
(8)	8	14 (T4)		
(9)	9			
(10)	10			
(11)	11			
(12)	12			
(13)	13			

Číselné označenie signálov

- 1 HODINY VSTUP 9 POTVRDENIE
- 2 HODINY VÝSTUP 10 +5 Voltov
- 3 DATA VSTUP 11 ZVUKOVÝ VSTUP
- 4 ZEM (0 Voltov) 12 +12 Voltov
- 5 DATA VÝSTUP 13 PRERUŠENIE
- 6 ZEM (0 Voltov) 14 RÝCHLE DATA
- 7 POKL
- 8 RIADENIE MOTORA

V zátvorkách sú uvedené čísla pínov konektorov PERIPHERAL, PERIF2 a PERIF3. (T4) je označenie pínu č.4 na konektore "TLAČIAREN" modulu pre pripojenie tlačiarne (BUSY).

Obr. 32 Spojení jednotlivých modulů komunikačního adaptéru pomocí rozšířené sériové komunikační sběrnice ATARI

Obr. 32 zobrazuje připojení jednotlivých modulů pomocí sériové sběrnice ATARI, rozšířené o signál č.14 "RYCHLÁ DATA". Konektor PERIF2 slouží na připojení upraveného magnetofonu pro režim dekódování dat z magnetofonu pomocí modulu pro připojení tiskárny. Konektor PERIF3 slouží na připojení upraveného i neupraveného magnetofonu, když se využívá interní dekodér z magnetofonu (pro rychlosti 500 až 1320 Bd). Podobně jako PERIF2 je to zjednodušená funkční podmnožina konektoru PERIPHERAL - obsahuje jen signály nutné pro činnost magnetofonu. Z uvedeného vyplývá, že všechny moduly zde popisované mohou být současně připojené na sériovou komunikační sběrnici ATARI. Kombinace tiskárna-komunikační linka musí být po dobu komunikace s tiskárnou obsluhována ručně (odpojením konektoru IRPS).

Celý komplet je možné doplnit o samostatný zdroj napětí +5V (0,5 A) a případně napětí +12V a -12V a vestavět do vhodné krabičky, např. U5.

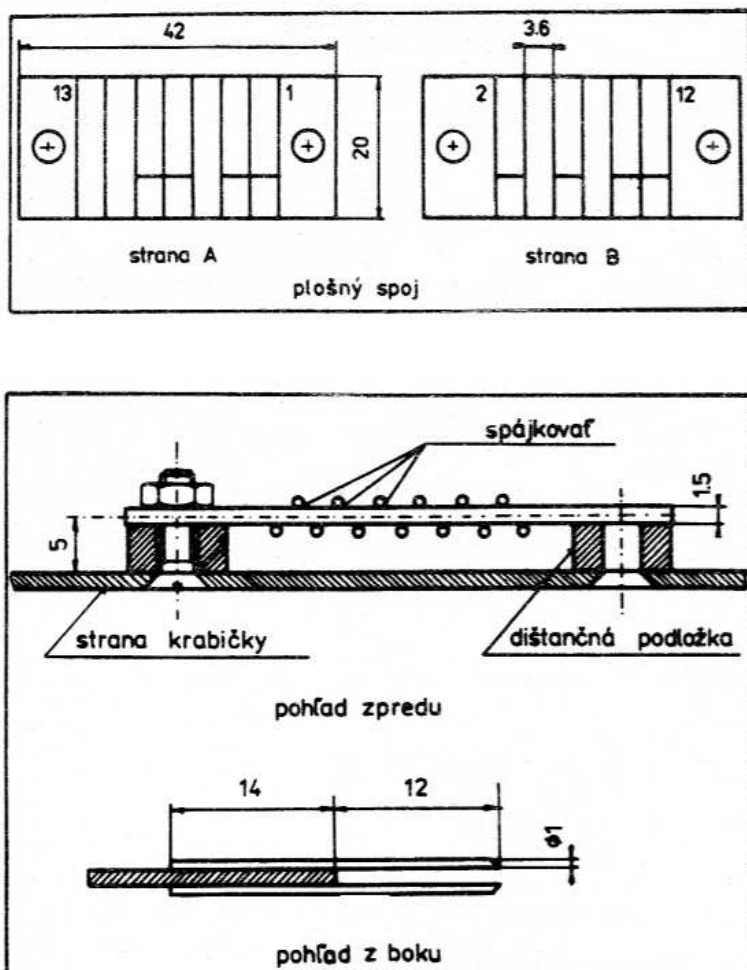
Napájení +5V je třeba zabezpečit proti poklesu při činnosti magnetofonu např. připojením blokovacích kondenzátorů s kapacitou min. 500M/6V. Z toho vyplývá, že modul se nesmí připojovat na počítač pod napětím, ale vždy jen ve vypnutém stavu.

Je možné samostatně postavit i používat jednotlivé moduly. V tomto materiálu je dost informací, které umožňují jejich samostatnou stavbu i využívání i pro méně zkušené uživatele.

Upozornění autorů: Komunikační adaptér pro počítač ATARI a příslušné programové vybavení včetně firemního pro mikropočítač 8048 apod. je předmětem ZN 12/88 (evidence ve VÚVT, k.u.o., Nerudova ul. 33, p.p.E-12, 010 01 Žilina).

Poznámka redakce: Programová podpora pro komunikační modul, jakož i podrobný popis jednotlivých programů bude k dispozici v klubové programové bance, jakmile ji autoři poskytnou a uvolní k šíření.

Poznámka k realizaci: Na plošný spoj se nejdříve připájkuje všech 13 jehlic z drátu o průměru 1 mm a délce 25 mm. Jehlice by měly vyčnívat z plošného spoje asi 12 mm. Vyčnívající jehlice je vhodné na hrotech srazit v tupém úhlu dle obrázku, případně i z vnitřní strany. Potom se připájkuje vodiče na piny konektoru č. 4, 5, 8, 10, 11 a 13, resp. 3, 4, 5, 8, 10, 11. Plošný spoj se přišroubuje stěnu krabičky, přičemž se distančními destičkami nastaví jeho vzdálenost na 5 mm podle obrázku.



Obr. 33 Náčrt konstrukce konektorů PERIF2 a PERIF3.

5.3. Systémová sběrnice

5.3.1. Programátor pamětí EPROM

Jan Pavlásek (Praha)

Přestože ATARI XL/XE má celkem dobré vybavení pro připojení periférií (obvody PIA a POKEY), může se stát, že máme ještě další potřeby, které tento paralelní a sériový port neuspokojí. Pro tyto případy je počítač vybaven dvěma konektory, na kterých jsou vyvedeny signály sběrnice, které potřebujeme na připojení dalších obvodů.

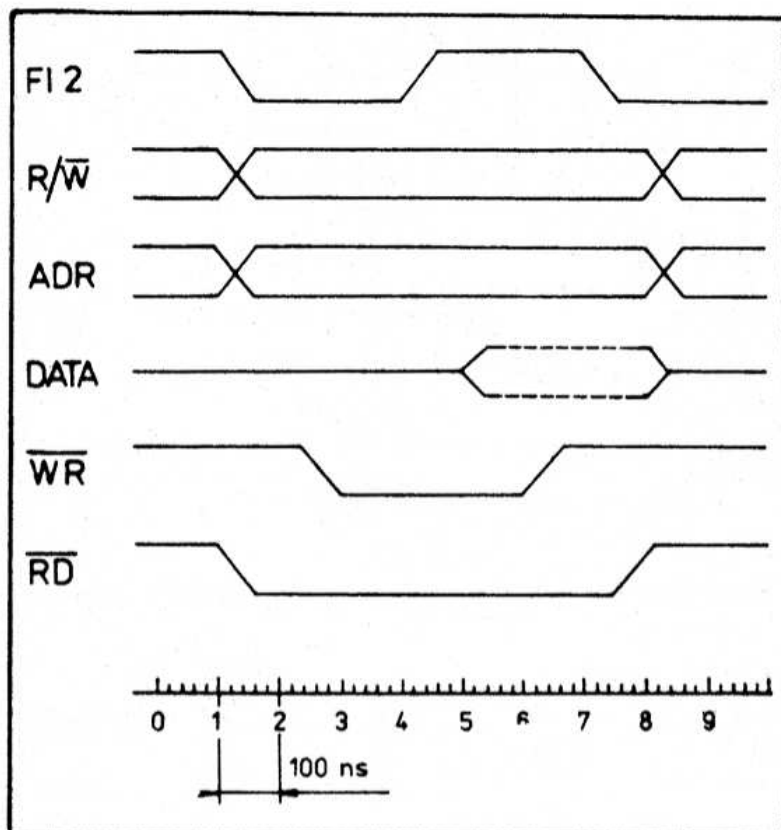
Při využívání této možnosti musíme mít na mysli určité zásady, které spočívají v podstatě na tom, že signály sběrnice musí mít v určitém čase přesně definované napětové úrovně, a že budiče těchto signálů mohou z důvodů přetížení dodávat proud jen do určité velikosti.

Mikroprocesory typu 65xx mají zatížitelnost signálů minimálně -1,6 mA při úrovni L, to znamená, že součet všech proudů z připojených obvodů by neměl tuto hodnotu přesáhnout (samozřejmě i zátěž obvodů uvnitř počítače.)

Pro informaci: obvody typu 74LS mohou mít vstupní proud maximálně -0,4 mA a obvody 82xx -0,2 mA. Ve skutečnosti má většina obvodů tento proud daleko menší, takže si často můžeme dovolit signály nijak neoddělovat, čímž si ulehčíme práci.

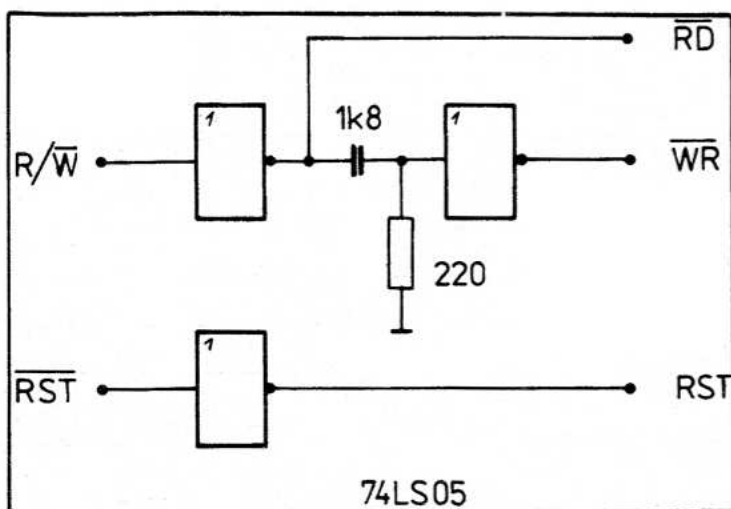
Další činitel, který se nevyplácí přehlédnout, je impedance, hlavně její kapacitní složka, která může ovlivnit vzdálenost vodičů. Také je třeba si uvědomit, že ne všechny obvody mají stejná potřebná napětí pro stavy L a H.

Signály na sběrnici mají určité vztahy, které jsou popsány na obr. 34

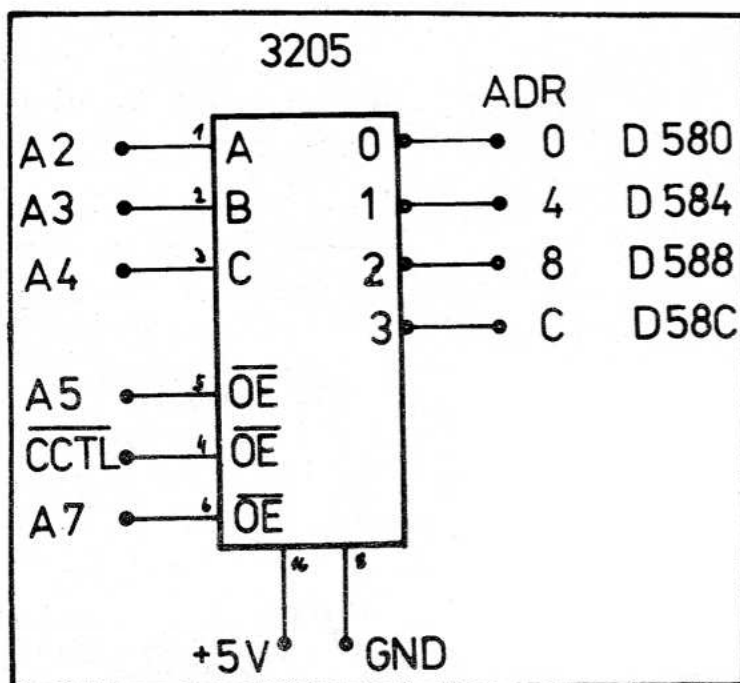


Obr. 34 Signály na sběrnici

Pro spolupráci s touto sběrnici je navrženo mnoho užitečných obvodů řady 65xx a 68xx, které však nejsou apatrně nikdy nebudou na našem trhu. (V ATARI XL/XE je použit paralelní port 6520.) V obchodech ale můžeme občas zakoupit československé nebo sovětské obvody řady 82xx, které jsou sice určeny pro spolupráci s jinou sběrnici, ale my si můžeme pro ně potřebné řídicí signály vyrobit z jiných obvodů. (Např. pomocí dekodéru na obr. 35).

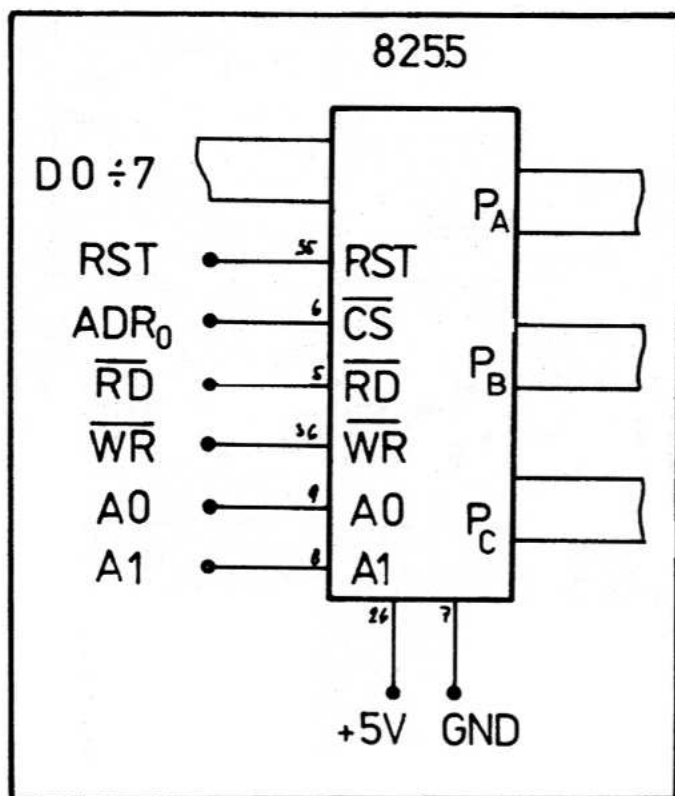


Obr. 35 Dekodér



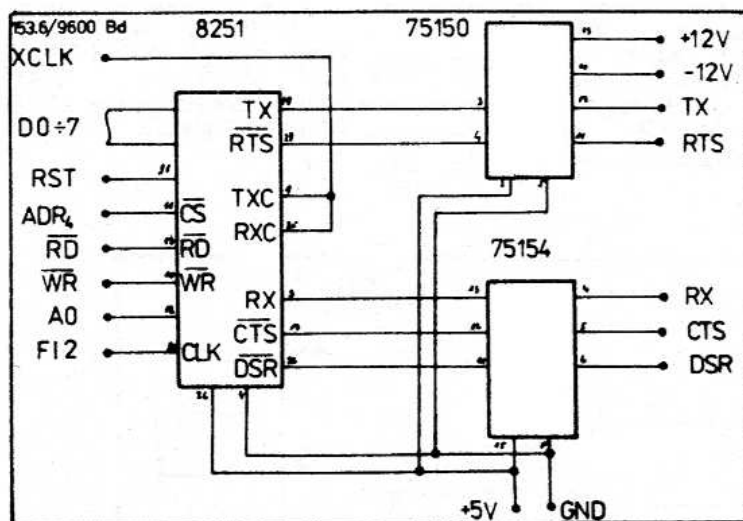
Obr. 36 Příklad dekodování signálů z adresové sběrnice

Signály pro výběr dalších obvodů se musí dekodovat z adresové sběrnice. Příklad jak, to je na obr. 36. Zde je výhodně použit signál označený v dokumentaci od počítače jako CCTL. Je to vlastně dekodovaná adresa skupiny 256 byte počínaje adresou \$D500. Adresu si můžeme dekodovat samozřejmě také na jiném volném místě. Potom nám ale přibudou další obvody pro dekodování adresových bitů A8 až A15.



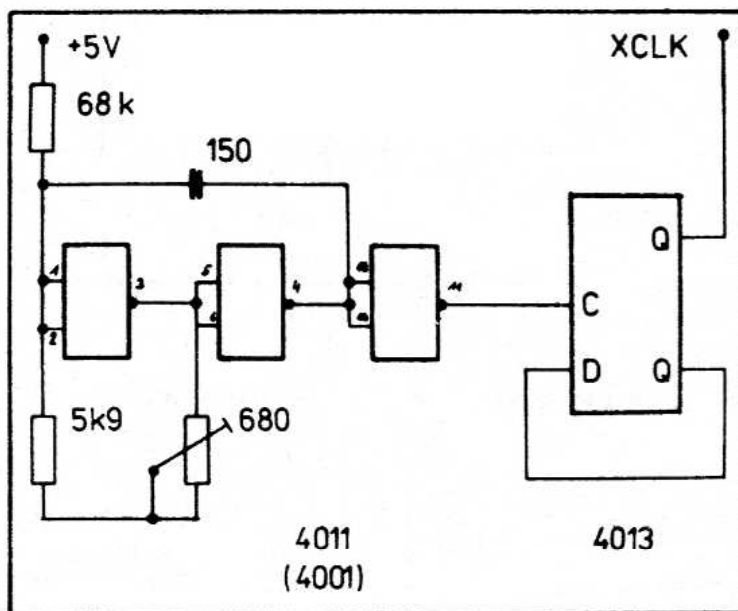
Obr. 37 Připojení paralelního portu 8255

Další dva obrázky (obr.37 a 38) ukazují připojení paralelního portu 8255 a sériového portu 8251. Zde jsou použity další, přímo propojené signály datové sběrnice, adresové bity A0 a A1 a signál Fi2. Obvody 75150 potřebují ještě další dvě napájecí napětí +12V a -12V. Pro ně musíme použít zvláštní napájecí zdroj. Ostatní obvody můžeme napájet bez problémů ze zdroje 5V pro ATARI XL/XE.



Obr. 38 Připojení sériového portu

Obvod 8251 potřebuje signál hodinového kmitočtu, který je lepší vytvořit dělením kmitočtu oscilátoru, řízeném stalem. Pokud máme dostatečné stabilní napájecí napětí 5V, odívku vyhoví i oscilátor nakreslený na obr.39.



Obr. 39 Oscilator

Na paralelní a sériový port můžeme připojovat celou řadu dalších zařízení, třeba A/D a D/A převodníky, různá ovládání, tiskárny, jiné počítače atd. Tato zařízení mohou být i velmi jednoduchá, jako třeba dále popsaný programovací přípravek na paměti 2716-27256. Je to vlastně jen trochu složitější propojení konektorů portu a paměti. Tato jednoduchost zde vyžaduje několik kompromisů. Programovací napětí a typy pamětí se zde přepojují ručně, takže při používání musíme dávat větší pozor, než kdyby vše bylo řízeno programem. Protože ne všechny programované paměti mohou být v pořádku, je dobré chránit všechny signály portu 8255 alespoň diodami zapojenými katodou na napájecí napětí 5V.

Z názorného programu napsaném v TURBO BASICu je zřejmé, jak programátor pracuje. Data se kterými pracujeme jsou na disketu zapisována ve formátu .COM, tj. mají na začátek připsáno šest bytů (\$FF, \$FF, počáteční adresu a koncovou adresu). Je to proto, aby se pro úpravy dat daly používat různé programy pro práci ve strojovém kódu, jako je třeba DEBUG.

Následující tabulka ukazuje, jak je obvod 8255 (objímka pro paměť a čtyři nebo pět spínačů) propojen. V poslední tabulce jsou popsány vývody různých pamětí EPROM a CMOS RAM.

Konektor	Port 8255	Spínač/signál, pin
EPROM	signál pin	

1	-		
2	-	C4	17
3	1	B7	25
4	2	B6	24
5	3	B5	23
6	4	B4	22
7	5	B3	21
8	6	B2	20
9	7	B1	19
10	8	B0	18
11	9	A0	4
12	10	A1	3
13	11	A2	2
14	12	GND	
15	13	A3	1
16	14	A4	40
17	15	A5	39
18	16	A6	38
19	17	A7	37
20	18	-----3/C5p10 4/GND	
21	19	C2	8
22	20	-----5/C7p19 8/VPP	
23	21	-----6/C3p15 7/VPP	
24	22	C1	6
25	23	C0	4
26	24	-----2/5V 1/C5p10	
27		C6	19
28		+5V	

NASTAVENÍ SPÍNAČŮ:

Číslo spínačů: 1234 číslo značí, že spínač je sepnut,
5678 *, že není sepnut

typ paměti:

Vpp:

2716	1*3*	5*7*	25V
2532	1*3*	5*7*	25V
2732	R 1*3*	W 1*3*	21V 12.5V
	56**	*6*8	
2564	1*3*		25V
	56**		
2764	*2*4		21V 12.5V
	56**		
27128	*2*4		21V 12.5V
	56**		
27256	*2*4		12.5V
	56**		

pin	2715	2732	2532	2764	2584	6284	6116	27128	27256
1	--	--	--	Vpp A12	Vpp CS1:	--	--	Vpp A12	Vpp A12
2	--	--	--	A7	A7	A12	--	A7	A7
3	A7	A6	A7	A6	A6	A7	A7	A6	A6
4	A6	A5	A6	A5	A5	A6	A6	A5	A5
5	A5	A4	A5	A4	A4	A5	A5	A4	A4
6	A4	A3	A4	A3	A3	A4	A4	A3	A3
7	A3	A2	A3	A2	A2	A3	A3	A2	A2
8	A2	A1	A2	A1	A1	A2	A2	A1	A1
9	A1	A0	A1	A0	A0	A1	A1	A0	A0
10	A0	D0	A0	D0	D0	A0	A0	D0	D0
11	D0	D1	D0	D1	D1	D0	D0	D1	D1
12	D1	D2	D1	D2	D2	D1	D1	D2	D2
13	D2	GND	D2	GND	GND	D2	GND	GND	GND
14	GND	D3	GND	D3	D3	GND	D3	D3	D3
15	D3	D4	D3	D4	D4	D3	D3	D4	D4
16	D4	D5	D4	D5	D5	D4	D4	D5	D5
17	D5	D6	D5	D6	D6	D5	D5	D6	D6
18	D6	D7	D6	D7	D7	D6	D6	D7	D7
19	D7	CE!	D7	CE!	CE!	D7	CE!	CE!	CE!
20	CE!/PGM	A11	A11	A11	A11	CE!	CE!	CE!	A10
21	A10	A10	A10	A10	A10	A10	A10	A10	OE!/VFF
22	OE!	OE!/VPP	PGM!/PD	OE!	PGM!/PD	OE!	OE!	OE!	OE!/VFF
23	VPP	A11	VPP	A11	A12	A11	WE!	A11	A11
24	A9	A9	A9	A9	A9	A9	A9	A9	A9
25	A8	A8	A8	A8	A8	A8	A8	A8	A8
26	+5V	+5V	+5V	NC	+5V	OE2	+5V	A13	A13
27	--	--	--	PGM!	CS2!	WR!	--	PGM!	A14
28	--	--	--	+5V	+5V	+5V	--	+5V	+5V

```

10 REM *** Programator pameti EPROM ***
20 REM
30 REM popis funkci programu:
40 REM .NUL kontroluje, maji-li vsechna data
50 REM ....hodnotu $FF tj. jestli je pamet prazdna
60 REM .VER kontroluje, jestli nactena EPROM umoznuje
70 REM .....naprogramovat data ze souboru
80 REM .DMP,DMC vypisuji data na obrazovku
90 REM .ADD pripisuje k jiz nactenym datum
100 REM ....dalsi data ze souboru
110 REM DIF porovnavá nactena data s daty ze souboru
120 REM SAV zapisuje data do souboru
130 REM LOR cte data ze souboru
140 REM R2,W2 cte,programuje EPROM 2716
150 REM R4,W4 cte,programuje EPROM 2732
160 REM R4T,W4T cte,programuje EPROM 2532
170 REM R8,W8 cte,programuje EPROM 2764
180 REM R8T,W8T cte,programuje EPROM 2564
190 REM R16,W16 cte,programuje EPROM 27128
200 REM R32,W32 cte,programuje EPROM 27256
210 DIM A$(16384):DIM NAM$(14):DIM D$(8)
220 PA=54656:PB=PA+1:PC=PA+2:CSR=PA+3
230 NUL=0:B0=1:B1=2:B2=4:B3=8:B4=16:B5=32
240 B6=64:B7=128:FF=$FF:PRP=65:OFF=$8000
250 TRAP 260
260 AA=ADR(A$):CLOSE
270 ? :? "NUL VER DMP DMC ADD DIF SAV LOR "
280 ? "R/W2 R/W4 R/W4T R/W8 R/W8T R/W16 R/W32"
290 INPUT NAM$
300 IF NAM$="SAV" THEN EXEC SAV
310 IF NAM$="LOR" THEN EXEC LOR
320 IF NAM$="DIF" THEN EXEC DIF
330 IF NAM$="ADD" THEN EXEC ADD
340 IF NAM$="VER" THEN EXEC VER
350 IF NAM$="NUL" THEN EXEC NULL
360 IF NAM$="DMP" THEN EXEC DMP
370 IF NAM$="DMC" THEN EXEC DMC
380 IF NAM$="R2" THEN EXEC R2
390 IF NAM$="R4" THEN EXEC R4
400 IF NAM$="R4T" THEN EXEC R4T
410 IF NAM$="R8" THEN EXEC R8
420 IF NAM$="R8T" THEN EXEC R8T
430 IF NAM$="R16" THEN EXEC R16

```

```

440 IF NAM$="R32" THEN EXEC R32
450 IF LENG=0 THEN ? "Schazi data !" : GOTO 260
460 IF NAM$="W2" THEN EXEC W2
470 IF NAM$="W4" THEN EXEC W4
480 IF NAM$="W4T" THEN EXEC W4T
490 IF NAM$="W8" THEN EXEC W8
500 IF NAM$="W8T" THEN EXEC W8T
510 IF NAM$="W16" THEN EXEC W16
520 IF NAM$="W32" THEN EXEC W32
530 IF NAM$(1,1)="W" THEN ? " ? " !! Odpoj programovaci napeti !! " ?
540 GOTO 260
550 REM *****
560 PROC SAVA
570   ? LENG, " BYTE"
580   INPUT "NAME OF OUTPUT FILE"; NAM$
590   OPEN #1, 8, 0, NAM$
600   BADR=0 : EADR=BADR+LENG-1
610   PUT #1, FF : PUT #1, FF
620   PUT #1, BADR MOD 256 : PUT #1, BADR DIV 256
630   PUT #1, EADR MOD 256 : PUT #1, EADR DIV 256
640   BPUT #1, AA, LENG
650   CLOSE #1
660 ENDPROC
670 REM *****
680 PROC LOR
690   INPUT "Name of input file"; NAM$
700   OPEN #1, 4, 0, NAM$
710   GET #1, P : GET #1, P : IF P <> FF THEN ? "File not ready !" : ENDPROC
720   GET #1, P : GET #1, BADR : BADR=BADR*256+P
730   GET #1, P : GET #1, EADR : EADR=EADR*256+P
740   LENG=EADR-BADR+1
750   BGET #1, AA, LENG
760   CLOSE #1
770   ? LENG, " BYTE"
780 ENDPROC
790 REM *****
800 PROC R2
810   LENG=2048
820   EXEC LNG
830   POKE CSR, B7 : B4 : K=0
840   FOR J=ZAC TO KON : POKE PC, J : B3
850     FOR I=NUL TO FF : POKE PB, I : POKE AA+K, PEEK(PA)
860     K=K+1 : NEXT I : ? " * " : NEXT J

```

```

870 POKE CSR,155
880 ? :? K;" BYTE"
890 ENDPROC
900 REM *****
910 PROC W2
920 EXEC LNG
930 POKE CSR,B7:K=0
940 POKE PC,B7
950 ? "Pripoj 25V !":GET P
960 FOR J=ZAC TO KON:POKE PC,J:B7
970     FOR I=NUL TO FF:POKE PB,I:POKE PA,PEEK(AA+K)
980         POKE PC,J:B5:B7:FOR N=0 TO PRP:NEXT N:POKE PC,J:B7
990         K=K+1:NEXT I:?"*":NEXT J
1000 ? :? K;" BYTE"
1010 ENDPROC
1020 REM *****
1030 PROC R4
1040 LENG=4096
1050 EXEC LNG
1060 POKE CSR,B7:B4:K=0
1070 FOR J=ZAC TO KON:POKE PC,J
1080     FOR I=NUL TO FF:POKE PB,I:POKE AA+K,PEEK(PA)
1090         K=K+1:NEXT I:?"*":NEXT J
1100 POKE CSR,155
1110 ? :? K;" BYTE"
1120 ENDPROC
1130 REM *****
1140 PROC R4T
1150 LENG=4096
1160 POKE CSR,B7:B4:K=0
1170 FOR P=NUL TO B5 STEP B5
1180     FOR J=NUL TO 7:POKE PC,J:P:B3
1190         FOR I=NUL TO FF:POKE PB,I:POKE AA+K,PEEK(PA)
1200             K=K+1:NEXT I:?"*":NEXT J:NEXT P
1210 POKE CSR,155
1220 ? :? K;" BYTE"
1230 ENDPROC
1240 REM *****
1250 PROC R8
1260 LENG=8192
1270 EXEC LNG
1280 POKE CSR,B7:B4:K=0

```

```

1290   FOR J=280 TO KON:POKE PC,J!B6
1300   FOR I=NUL TO FF:POKE PB,I:POKE AA+K,PEEK(PA)
1310       K=K+1:NEXT I:?"*";:NEXT J
1320   POKE CSR,155
1330   ? :? K;" BYTE"
1340 ENDPROC
1350 REM *****
1360 PROC R8T
1370   LENG=8192
1380   DATA 0,32,8,40
1390   RESTORE 1380
1400   POKE CSR,B7!B4:K=0
1410   FOR L=NUL TO 3:READ P
1420       FOR J=NUL TO 7:POKE PC,P!J
1430           FOR I=NUL TO FF:POKE PB,I:POKE AA+K,PEEK(PA)
1440               K=K+1:NEXT I:?"*";:NEXT J:NEXT L
1450       POKE CSR,155
1460       ? :? K;" BYTE"
1470   ENDPROC
1480 REM *****
1490 PROC R16
1500   LENG=16384
1510   EXEC LNG
1520   POKE CSR,B7!B4:K=0
1530   FOR J=280 TO KON:POKE PC,J!B6
1540       FOR I=NUL TO FF:POKE PB,I:POKE AA+K,PEEK(PA)
1550           K=K+1:NEXT I:?"*";:NEXT J
1560       POKE CSR,155
1570       ? :? K;" BYTE"
1580   ENDPROC
1590 REM *****
1600 PROC R32
1610   LENG=16384
1620   EXEC LNG
1630   POKE CSR,B7!B4:K=0
1640   FOR J=280 TO KON:POKE PC,J
1650       FOR I=NUL TO FF:POKE PB,I:POKE AA+K,PEEK(PA)
1660           K=K+1:NEXT I:?"*";:NEXT J
1670       POKE CSR,155
1680       ? :? K;" BYTE"
1690   ENDPROC
1700 REM *****
1710 PROC W4
1720   EXEC LNG

```

```

1730 POKE CSR,B7:K=0
1740 POKE PC,B5
1750 ? "Pripoj 25V !":GET P
1760 FOR J=ZAC TO KON:POKE PC,J!B5
1770     FOR I=NUL TO FF:POKE PB,I:POKE PA,PEEK(AA+K)
1780     POKE PC,J:FOR N=0 TO PRP:NEXT N:POKE PC,J!B5
1790     K=K+1:NEXT I: ? "*" :NEXT J
1800 ? : ? K : " BYTE"
1810 ENDPROC
1820 REM *****
1830 PROC W4T
1840 POKE CSR,B7:K=0
1850 POKE PC,B7
1860 ? "Pripoj 25V !":GET P
1870 FOR P=NUL TO B5 STEP B5
1880     FOR J=NUL TO 7:POKE PC,P!J!B7
1890     FOR I=NUL TO FF:POKE PB,I:POKE PA,PEEK(AA+K)
1900     POKE PC,P!J:FOR N=0 TO PRP:NEXT N:POKE PC,P!J!B7
1910     K=K+1:NEXT I: ? "*" :NEXT J:NEXT P
1920 ? : ? K : " BYTE"
1930 ENDPROC
1940 REM *****
1950 PROC W8
1960 EXEC LNG
1970 POKE CSR,B7:K=0
1980 POKE PC,B7!B6
1990 ? "Pripoj 21V (12.5V) !":GET P
2000 FOR J=ZAC TO KON:POKE PC,J!B7!B6
2010     FOR I=NUL TO FF:POKE PB,I:POKE PA,PEEK(AA+K)
2020     POKE PC,J!B7:FOR N=0 TO PRP:NEXT N:POKE PC,J!B7!B6
2030     K=K+1:NEXT I: ? "*" :NEXT J
2040 ? : ? K : " BYTE"
2050 ENDPROC
2060 REM *****
2070 PROC W8T
2080 RESTORE 2090
2090 DATA 0,32,8,40
2100 POKE CSR,B7:K=0
2110 POKE PC,B7
2120 ? "Pripoj 25V !":GET P
2130 FOR L=NUL TO 3:READ P
2140     FOR J=NUL TO 7:POKE PC,P!J!B7

```

```

2150      FOR I=NUL TO FF:POKE PB,I:POKE PA,PEEK(AA+K)
2160      POKE PC,P!J:FOR N=0 TO PRP:NEXT N:POKE PC,P!J:B7
2170      K=K+1:NEXT I:?"*":NEXT J:NEXT L
2180  ? :? K;" BYTE"
2190 ENDPROC
2200 REM *****
2210 PROC W16
2220  EXEC LNG
2230  POKE CSR,B7:K=0
2240  POKE PC,B7!B6
2250  ? "Pripoj 21V (12.5V) !":GET P
2260  FOR J=ZAC TO KON:POKE PC,J!B7!B6
2270      FOR I=NUL TO FF:POKE PB,I:POKE PA,PEEK(AA+K)
2280      POKE PC,J!B7:FOR N=0 TO PRP:NEXT N:POKE PC,J!B7!B6
2290      K=K+1:NEXT I:?"*":NEXT J
2300  ? :? K;" BYTE"
2310 ENDPROC
2320 REM *****
2330 PROC W32
2340  EXEC LNG
2350  POKE CSR,B7:K=0
2360  POKE PC,B7
2370  ? "Pripoj 12.5V !":GET P
2380  FOR J=ZAC TO KON:POKE PC,J!B7
2390      FOR I=NUL TO FF:POKE PB,I:POKE PA,PEEK(AA+K)
2400      POKE PC,J:FOR N=0 TO PRP:NEXT N:POKE PC,J!B7
2410      K=K+1:NEXT I:?"*":NEXT J
2420  ? :? K;" BYTE"
2430 ENDPROC
2440 REM *****
2450 PROC DMP
2460  INPUT "Adresa (HEX)";D$:A=DEC(D$):N=LENG-A
2470  FOR I=1 TO N STEP 8
2480      IF A<$0100 THEN ? "00";
2490      ? HEX$(A);" ";
2500      FOR J=1 TO 8:D=PEEK(A+AA):D$(J,J)=CHR$(D)
2510      IF (D<32) OR (D>124) THEN D$(J,J)="."
2520      A=A+1:PRINT HEX$(D);" ";:NEXT J
2530  PRINT D$:NEXT I
2540 ENDPROC
2550 REM *****
2560 PROC NULL

```

```

2570 FOR I=AA TO AA+LENG-1
2580 D=PEEK(I)
2590 IF D<>FF THEN ? "Chyba ! a ";HEX$(I-AA); " d ";HEX$(D)
2600 NEXT I
2610 ? "End of NULL":ENDPROC
2620 REM *****
2630 PROC DIF
2640 INPUT "Name Dif File ";NAM$
2650 OPEN #1,4,0,NAM$
2660 FOR I=0 TO 5:GET #1,D:NEXT I
2670 FOR I=0 TO LENG-1:GET #1,D
2680 IF D<>PEEK(AA+I)
2690 ? "dif a ";HEX$(I); " f ";HEX$(D); " m ";HEX$(PEEK(AA+I))
2700 ENDIF
2710 NEXT I: ? "End of DIF":ENDPROC
2720 REM *****
2730 PROC VER
2740 INPUT "Name Prog File ";NAM$
2750 OPEN #1,4,0,NAM$
2760 FOR I=0 TO 5:GET #1,D:NEXT I
2770 FOR I=0 TO LENG-1:GET #1,D
2780 IF D<(FF-PEEK(AA+I))
2790 ? "Chyba ! a ";HEX$(I); " f ";HEX$(D); " m
";HEX$(PEEK(AA+I))
2800 ENDIF
2810 NEXT I: ? "End of VER":ENDPROC
2820 REM *****
2830 PROC DMC
2840 INPUT "Adresa (HEX)";D$:A=DEC(D$):N=LENG-A
2850 FOR I=A+AA TO A+AA+N-1
2860 D=PEEK(I)
2870 IF A<256 THEN ? " ";
2880 ? HEX$(A); " ";HEX$(D); " ";
2890 IF D&128: ? "*"; ELSE : ? "."; ENDIF
2900 IF D&64: ? "*"; ELSE : ? "."; ENDIF
2910 IF D&32: ? "*"; ELSE : ? "."; ENDIF
2920 IF D&16: ? "*"; ELSE : ? "."; ENDIF
2930 IF D&8: ? "*"; ELSE : ? "."; ENDIF
2940 IF D&4: ? "*"; ELSE : ? "."; ENDIF
2950 IF D&2: ? "*"; ELSE : ? "."; ENDIF
2960 IF D&1: ? "*"; ELSE : ? "."; ENDIF
2970 ? :A=A+1
2980 NEXT I

```

```

2990 ENDPROC
3000 REM *****
3010 PROC ADD
3020   INPUT "NAME OF ADD FILE";NAM$
3030   OPEN #1,4,0,NAM$
3040   GET #1,P:GET #1,P
3050   IF P<>FF THEN ? "File not ready !":ENDPROC
3060   GET #1,P:GET #1,BADR1:BADR1=BADR1*256+P
3070   GET #1,P:GET #1,EADR:EADR=EADR*256+P
3080   LENG1=EADR-BADR1+1
3090   BGET #1,AA+LENG,LENG1
3100   LENG=LENG+LENG1
3110   CLOSE #1
3120   ? LENG;" BYTE"
3130 ENDPROC
3140 REM *****
3150 PROC LNG
3160   ? "Data ";LENG DIV 1024;"KB"
3170   TRAP 3200
3180   INPUT "Zacatek [kB]";ZAC
3190   INPUT "Velikost [kB]";KON:GOTO 3210
3200   ZAC=0:KON=LENG DIV 1024
3210   ZAC=ZAC*4:KON=KON*4
3220   IF LENG DIV 256<KON THEN ? "Chyba ve velikosti dat"
3230   KON=ZAC+KON-1
3240   ? "$";HEX$(ZAC*256);"..$";HEX$(KON*256+255)
3250 ENDPROC

```

Použitá a doporučená literatura

- <1> Operating System User s Manual ATARI 400/800.
- <2> Hardware manual ATARI 400/800.
- <3> Resche, J., Wietheff, A.: ATARI Profibuch.
- <4> Chasin, M.: Assembly Language Programming for the Atari Computers.
- <5> Wasner, H.C.: Hofacker Book fot your Atari Computer.
- <6> ATARI User, 4/1987.
- <7> Amatérské radio, 10/1984.
- <8> EPSON 286 manual.
- <9> Happy Computer sonder heft ATARI.
- <10> Robotron K 6311, K 6312 - manual.

Publikované zo súhlasom - vid' Prohlášení představitelů AK Praha.