

VYSOKÁ ŠKOLA STROJNÍ A TEXTILNÍ V LIBERCI

Fakulta strojní

Katedra technické kybernetiky Školní rok: 1992/93

ZADÁNÍ DIPLOMOVÉ PRÁCE

pro Rolanda HASE

obor 23-40-8 ASŘ výrobních procesů ve strojírenství

Vedoucí katedry Vám ve smyslu zákona č. 172/1990 Sb. o vysokých školách určuje tuto diplomovou práci:

Název tématu:

Aplikace řídicího modulu IMM 552

Zásady pro vypracování:

- 1) Prostudujte dokumentaci k řídicímu modulu IMM 552, seznamte se s jazyky BASIC a C vhodnými pro práci s IMM 552.
- 2) Navrhněte a realizujte vstupní a výstupní obvody, které umožní připojit k IMM 552 stejnosměrný motor, tachodynamo, IRC čidlo. Dále vybavte IMM 552 vstupy 0-10 V a výstupy 0-10 V.
- 3) Realizujte model dynamické soustavy vhodný pro použití při výuce regulace. Na modelu ověřte základní principy regulátorů (PID) s využitím IMM 552 vybaveného dle (2).

VYSOKÁ ŠKOLA STROJNÍ A TEXTILNÍ
Ústřední knihovna
LIBEREC 1, STUDENTSKÁ 8
PSC 461 17

V 158

1935

KTR/ASŘ


Rozsah grafických prací:
Rozsah průvodní zprávy: 40 stran
Seznam odborné literatury:
1) Dokumentace k IMM 552
2) Dokumentace BASIC, C

Vedoucí diplomové práce: Ing. Petr Tůma
Konzultant: Ing. J. Janeček, CSc.; Ing. B. Janeček, CSc.
Doc. Ing. O. Modrlák, CSc.

Zadání diplomové práce: 31.10.1992
Termín odevzdání diplomové práce: 28.5.1993



Vedoucí katedry


Doc. Ing. Vojtěch Konopa, CSc.


Děkan

Prof. Ing. Jaroslav Exner, CSc.

V Liberci

dne 19.10. 1992

Vysoká škola strojní a textilní v Liberci
Fakulta strojní

Roland Hase
Aplikace řídicího modulu IMM 552

Diplomová práce

1993

Vysoká škola strojní a textilní v Liberci
fakulta strojní
Obor 23-40-8

ASŘ výrobních procesů ve strojírenství
Katedra technické kybernetiky

APLIKACE ŘÍDÍCÍHO MODULU IMM552

Autor: Roland Hase
Vedoucí práce: Ing. Petr Tůma, CSc.
Konzultant: Ing. Josef Janeček, CSc.
Ing. Bedřich Janeček, CSc.
Doc. Ing. Osvald Modrlák, CSc.

UNIVERZITNÍ KNIHOVNA
TECHNICKÉ UNIVERZITY V LIBERCI



3146075514

Místopřísežně prohlašuji, že jsem diplomovou práci
vypracoval samostatně, s použitím uvedené literatury.

V Liberci dne 28.května 1993

--- *Starč* ---

Poděkování

Děkuji vedoucímu mé diplomové práce Ing. Petru Tůmovi, CSc., za odbornou pomoc, cenné rady a připomínky k diplomové práci.

Děkuji uvedeným konzultantům a Ing. J. Kloboučkovi, kteří též přispěli cennými radami k zdárnému dokončení práce.

Dále bych chtěl poděkovat zástupcům firmy Weidmüller za poskytnutou pomoc, která umožnila včasné dokončení práce.

OBSAH

1. Úvod	7
1.1 Úvodní část	7
1.2 Popis obsahu práce	8
2. Hardware	9
2.1 Napájení	9
2.2 Modul IMM552	9
2.2.1 CPU PCB 80C552	10
2.2.2 Adresový dekodér	12
2.2.3 Paměť	13
2.2.4 Hodiny reálného času	14
2.2.5 Referenční zdroj	14
2.2.6 Stavba IMM552	15
2.3 Sériové rozhraní	19
2.4 Analogový vstup	20
2.5 Číslicový vstup	21
2.6 Výkonová část zařízení	25
2.6.1 Obvod buzení	26
2.6.2 Obvod kotvy	26
3. Software	30
3.1 Programování v BASICu	31
3.1.1 MICROBASIC - stručný popis	31
3.1.2 Popis příkazů BASICu	35
3.1.3 Aritmetické, logické a relační operátory	38
3.1.4 Priorita operátorů	38
3.1.5 Speciální příkazy	38

3.1.6 Změna interpretu BASICu pro IMM552	39
3.1.7 Uchování programu v RAM při provozu na baterii	41
3.1.8 Programování autostartovací EPROM	41
3.2 Programová podpora řízení	42
3.2.1 Ovládací a zobrazovací program	43
3.2.2 Program pro regulaci a počítačnické podmínky	43
4. Závěr	44
Příkazy MICROBASICU (tabulka)	45
Literatura	46
Seznam příloh	47

1. Úvod

1.1 Úvodní část

S rozvojem a modernizací technologických procesů stoupají nároky na jejich řízení, koordinování jednotlivých pracovních činností a na řízení těchto procesů v reálném čase. Je nutné získávat co nejrychleji co největší množství informací o stavu procesů, ty vyhodnocovat a na základě řídicího algoritmu provádět příslušné akční zásahy (regulaci). Je však nutné mít možnost zásáhnout do regulačního procesu a to formou změny žádané hodnoty, nebo nastavených parametrů regulátoru.

Pro řízení běžných procesů v průmyslu (regulace vytápění objektů, řízení otáček motorů, regulace odběru elektrické energie, řízení výšky hladiny tekutiny v nádobě, atd.) není nutné nasazovat tu nejmodernější a nejvýkonnější řídicí elektroniku, která je právě k dispozici. U takovýchto aplikací se stále častěji přistupuje k použití 8 bitových jednočipových mikropočítačů, tzv. mikrokontrolerů. Tato zařízení umožňují řídit tyto procesy v reálném čase a splňují kladené požadavky, a to vše při nižších pořizovacích nákladech.

1.2 Popis obsahu práce

Hlavním předmětem předložené práce je vytvoření přizpůsobovacích obvodů (interface) pro řídicí mikropočítačový modul IMM552 (od firmy Weidmüller) dále použitelný pro řízení stejnosměrného elektromotoru s cizím buzením, opatřeného tachodynamem a čidlem IRC 120/2500 (inkrementální rotační čidlo). Dále pak ověřit chování celého zařízení pomocí software. Následně má být toto zařízení použito při výuce předmětů zabývajících se regulací.

Pro řízení je použit číslicový proporcionálně - integračně - derivační regulátor (PSD), který s danou periodou vzorkování vyhodnocuje regulační odchylku tvořenou rozdílem žádané hodnoty a skutečné hodnoty výstupní veličiny a provádí akční zásahy s cílem vynulování regulační odchylky.

Výstupní veličinou soustavy mohou být:

1. otáčky vztažené k napětí tachodynamu
2. otáčky vypočtené z počtu impulsů vystupujících z IRC čidla za určitý časový interval
3. poloha daná počtem impulsů vystupujících z IRC čidla

Tuto soustavu je tedy možno řídit třemi různými způsoby. Pro zajištění komunikace se zařízením je připojen počítač typu PC ve funkci terminálu.

2. Hardware

Celé zřízení lze rozdělit do několika částí:

1. napájení
2. modul IMM552
3. sériové rozhraní
4. analogový vstup (tachodynamo)
5. číslicový vstup (IRC čidlo)
6. výkonová část zařízení

2.1 Napájení

Napájení celého zařízení je řešeno stejnosměrným napětím 30V. Z tohoto napětí je dále odvozeno napájení řídicí elektroniky, která potřebuje stejnosměrné napětí +5V. To je provedeno integrovaným stabilizátorem MA 7805. Výstup je blokován filtračním kondenzátorem.

2.2 Modul IMM552

OEM - modul IMM552 (industrial microcomputer modul) je kompletní mikropočítačový systém na malé desce o rozměrech 60mm x 90mm.

V modulu jsou integrovány následující části:

mikroprocesor PCB 80C552-CMOS

adresový dekodér PLD (GAL)

32 kB CMOS-RAM

8 kB CMOS-EEPROM

max. 64 kB CMOS-EPROM

hodiny reálného času RTC72421A

reset systému MAX690

baterie s vypínací logikou pro RAM a RTC

zdroj referenčního napětí 2,5V AD580

2.2.1 CPU PCB 80C552

Obvod PCB 80C552 je moderní 8-bitový jednočipový mikrokontroler, vyrobený technologií CMOS, pro aplikace v řízení, plně kompatibilní s průmyslovým standardem 80C51. Architektura I/O portů dává možnost nasazení tohoto kontroleru v oblastech samočinné elektroniky, specifického řízení motorů, a řízení převodovek, ad.

Na čipu je integrováno:

8051 code-compatible CPU

8kB interní ROM

256B interní RAM

2x16-bit. timer/counter

1x16-bit. timer/counter se 4 capture registry a 3 compare registry

watchdog-timer s programovatelným časem doběhu

5 8-bit. I/O portů

8-mi kanálový 10-bit. A/D převodník ($50\mu\text{s}=1$ převod)

2-kanálový PWM-výstup 8 bitů

full-duplex UART kompatibilní se standardem PCB 80C51

I²C-bus I/O-port s master/slave funkcemi

Toto zařízení tedy funguje jako aritmetický procesor mající technické prostředky schopné dvou binárních plus-bit operací. Instrukční soubor obsahuje víc jak 100 instrukcí: 44% jednobajtových, 41% dvoubajtových a 15% třibajtových. S 12MHz krystalem trvá 58% instrukcí $1\mu\text{s}$ a 40% instrukcí $2\mu\text{s}$. Násobící a dělicí instrukce trvají $4\mu\text{s}$.

4 capture registry umožňují po příchodu patřičného signálu odečíst momentální stav timeru 2. Tím se dají provést velmi přesná měření. Dále lze s pomocí 3 compare registrů v přesně definovaném okamžiku změnit stav všech linek portu 4.

Čtvrtý timer (timer 3) lze použít jako watchdog-timer. Tuto funkci lze uvolnit signálem /EW. Po uvolnění musí být timer 3 během programovatelného časového intervalu opět navrácen zpět. Jestliže se tak nestane (např. z důvodu havárie systému), pak se vyvolá RESET systému.

Zároveň je integrován 8-mi kanálový A/D převodník, který pracuje s 10-bit. převodem. Doba převodu je 50 μ s (při taktu 12MHz).

Přímo jako hardware jsou dále realizovány 2 8-bitové PWM výstupy, které mohou být následnými integracemi a zapojeními použity jako D/A výstupy. Také jsou pomocí toho možná např. výkonová řízení s pomocí postupně připojovaných výkonových stupňů. Perioda a střída PWM signálů je programovatelná.

I²C-bus je vhodný především pro spojení různých řídicích modulů do sítí. Tento I²C-bus je však vhodný jen pro krátké vzdálenosti, např. propojení v rámci přístrojové skříně.

2.2.2 Adresový dekodér

Pro dekódování adres je zabudován programovatelný logický obvod (GAL). Jelikož jsou také signály /PSEN a /RD připojeny na vstupy PLD obvodu, je možno dekódování adres volit v širokém rozmezí. S adresami je možno dekódovat

signály A8-A15.

Standardní obsazení adres je následující:

obvod	adresy	
RAM	0000H-7FFFH	data
EPROM	0000H-7FFFH	program
EEPROM	8000H-9FFFH	data/program
RTC	FC00H-FCFFH	data
CSF	F800H-FBFFH	active low

2.2.3 Paměť

Pro data je uživateli k dispozici 32kB RAM (U5). Příslušné propojení na desce zajistí, aby se obsah paměti RAM neztratil ani při vypnutí modulu IMM552. Toto vypnutí je aktivováno od cca 4,6V provozního napětí. Pokud je nutné, může místo standardního EEPROM (U4) být osazena další 32kB RAM, což snad stačí pro většinu aplikací.

Pro program může být na IMM552 osazeno maximálně 64kB EPROM. Standardní nastavení umožňuje použití 27C512 EPROM. Ve vestavěném EEPROM (U4) mohou být uživatelem uchována příslušná data a programy. Zápis do EEPROM je možný teprve propojením /WR->/WEEP. Selektce EEPROM může být zamezena odstraněním jumperu JP3 (2-3), což je zvláště užitečné při použití BASIC interpretu, když je v EEPROM BASIC program s autostartovacími funkcemi.

2.2.4 Hodiny reálného času

Hodiny reálného času na IMM552 (U8=RTC72421) mohou být spuštěny přímo přes datovou sběrnici. Spotřeba těchto hodin je v režimu power-down jen několik μA , takže mohou běžet dlouhou dobu (několik let). Jelikož je možné také obsah paměti RAM udržet baterií, záleží přirozeně na paměti, osazení obvodu RAM a teplotě IMM552, jak dloho baterie pro napájení vydrží. Pro případy, kdyby kapacita byla malá, lze externě připojit větší.

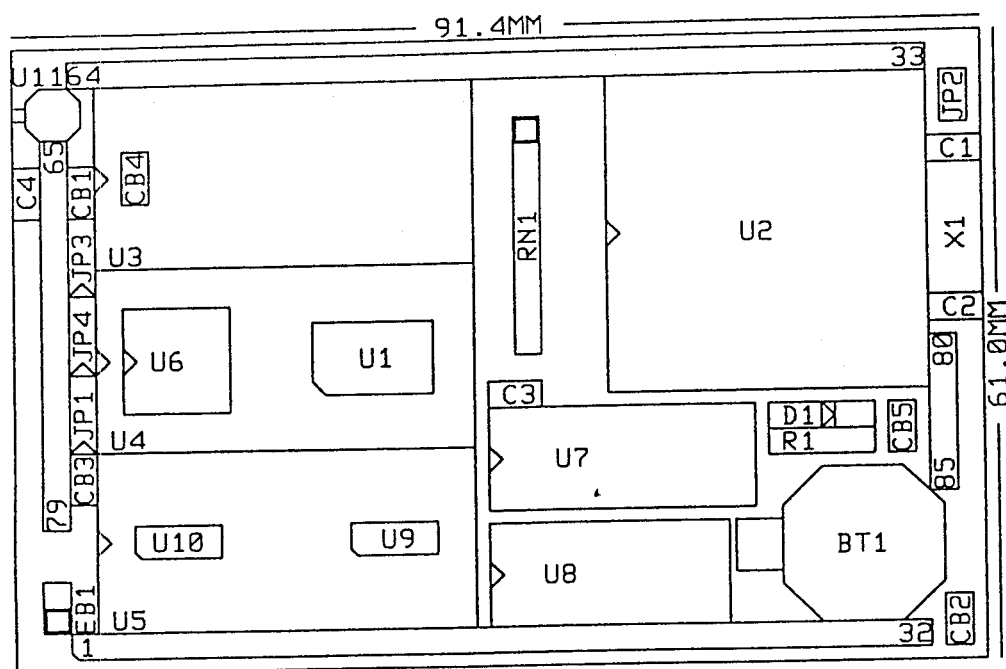
2.2.5 Referenční zdroj

Jako referenční napětí pro A/DA převodník je na IMM552 k dispozici referenční zdroj s nominální hodnotou 2,500V (U11=AD580). Pro použití, která potřebují externí referenci, nebo ta, která používají interní referenční napětí, jsou určeny vývody AVREF+ a AVREF-. S pomocí JP6, popřípadě JP5, lze AVREF- popř. AGND odděleně vyvést na samostatné části systému. Jumpery JP5 a JP6 jsou provedeny na spodní straně desky jako pájecí jumpery a propojeny tenkým plošným spojem.

2.2.6 Stavba IMM552

Většina integrovaných obvodů IMM552 mohla, díky použití 4-polohové - multilayer desky, být vestavěná pomocí patice. Také použití emulátoru reálného času je možno díky použití PLCC patice pro CPU.

Procesor je standardně taktován frekvencí 11,0592MHz. Z toho mohou být příslušnými dělicími faktory vytvořeny standardní počty Bd. Pro zabudování modulu je celá deska namontována na příslušné základní desce. Uživatel si pak vyřeší jen příslušné interfaceové problémy, a může IMM552 použít skoro jako jeden integrovaný díl. Celková montážní výška modulu je cca 21mm nad základní nosnou deskou.



Obrázek č. 1 Modul IMM552

Osazení modulu jumperů:

JP1 chip-select EEPROM/RAM U4	1-2=napájení baterií
	2-3=EEPROM
JP2 vypínání interní ROM CPU (/EA)(zablokováno)	
JP3 EEPROM U3	1-2=27C256
	2-3=27C512(standart)
JP4 UB přepínání U4	1-2=EEPROM
	2-3=napájení baterií
JP5 propojení GND a AGND (zablokováno)	
JP6 propojení VREF- s GND (zablokováno)	

Osazení desky modulu:

U1	74HCT373
U2	80C552
U3	EPROM 27C256
U4	EEPROM 28C64
U5	RAM 62C256
U6	MAX690
U7	GAL 16V8
U8	RTC 72421
U9/U10	74HC00 SOT-14
U11	AD580
RN1	SIL-odpor 8x22k 9-pólový
R1	KSR odpor 10k
BT1	lithiová baterie s držákem
C1,2	kondenzátor 22pF
C3,4,CB1..5	vícevrstvý kondenzátor 0,1μF
EG1	tantalový kondenzátor 1μF/16V
D1	dioda 1N4148
X1	krystal 11,0592MHz

Zapojení konektoru:

1	+5V	64	+5V	79	N.C.
2	N.C.	63	P5.7 ADC7	78	/WR
3	XTAL2	62	P5.6 ADC6	77	/WEEEP
4	A7	61	P5.5 ADC5	76	/WR
5	A6	60	P5.4 ADC4	75	A8
6	A5	59	P5.3 ADC3	74	A9
7	A4	58	P5.2 ADC2	73	A10
8	A3	57	P5.1 ADC1	72	A11
9	A2	56	P5.0 ADC0	71	A12
10	A1	55	STADC	70	A13
11	A0	54	/PWM0	69	A14
12	D7	53	/PWM1	68	A15
13	D6	52	/EW	67	AVREF+
14	D5	51	P4.0 CSMR0	66	AVREF-
15	D4	50	P4.1 CSMR1	65	AVSS
16	D3	49	P4.2 CSMR2		
17	D2	48	P4.3 CSMR3		
18	D1	47	P4.4 CSMR4	85	BAT
19	D0	46	P4.5 CSMR5	84	N.C.
20	ALE	45	P4.6 CMT0	83	UBAT
21	/PSEN	44	P4.7 CMT1	82	N.C.
22	P3.7 /RD	43	P1.0 CT0I	81	N.C.
23	P3.6 /WR	42	P1.1 CT1I	80	STD.P
24	RESIN	41	P1.2 CT2I		
25	RESOUT	40	P1.3 CT3I		
26	/CSRTC	39	P1.4 T2		
27	/CSF	38	P1.5 RT2		
28	P3.5 T1	37	P1.6 SCL		
29	P3.4 T0	36	P1.7 SDA		
30	P3.3 /INT0	35	P3.0 RXD		
31	P3.2 /INT1	34	P3.1 TXD		
32	GND	33	GND		

Následujícím zvláštností by se měla věnovat pozornost.

Pro programování EEPROM je nutné přemostit povolení zápisu pro EEPROM (PIN 76, 77). Také ve vypnutém stavu mohou být tyto PINy propojeny. Během zapínání a vypínání je /CE-vstup zajištěn pomocí Gatter (U9C), aby se zamezilo nechtěnné změně obsahu EEPROM. Vymazání celého obsahu EEPROM je možné pomocí následujícího řádku:

```
FOR I=8000H TO 9FFFH : XBY(I)=OFFH : PHO.I,CR, : NEXT I
```

Má-li být zrušena eventuální schopnost autostartu, pak se to provede pomocí:

```
XBY(8000H)=OFFH
```

Je-li v PROG2 nebo v PROG4 naprogramován autostart, pak je nutno při rušení postupovat takto:

jumper JP3 (2-3) odstranit

zapnout

zadáním "SPACE" IMM552 odstartovat

jumper JP3 (2-3) opět připojit (eventuálně přemostit šroubovákem)

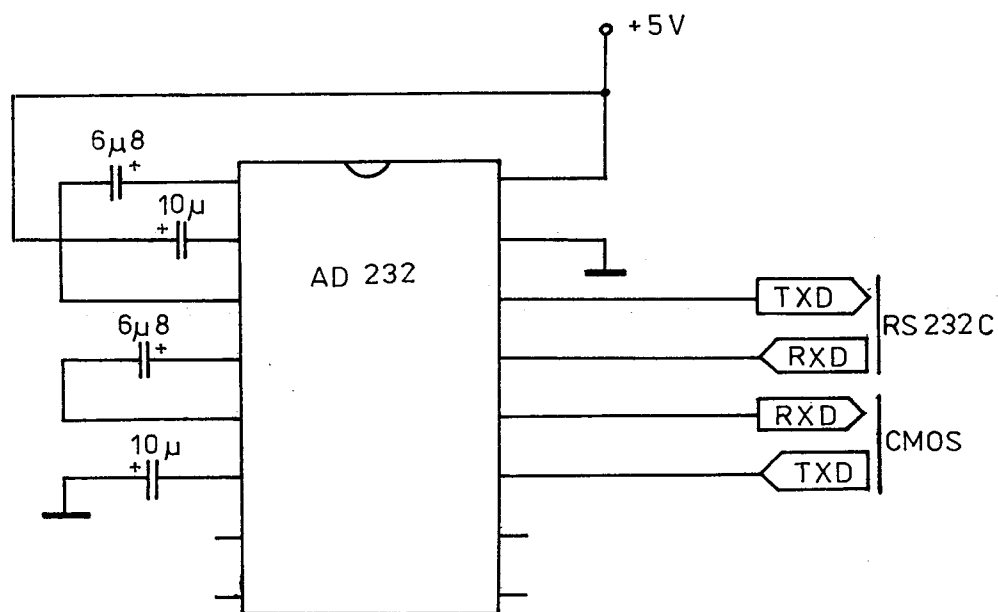
pomocí XBY(8000H)=OFFH odstranit schopnost autostartu

vypnout

jumper JP3 (2-3) opět nasadit

2.3 Sériové rozhraní

Pro komunikaci s okolním prostředím je modul IMM552 vybaven sériovým rozhraním se signály RXD a TXD ovšem v úrovních CMOS. Pro připojení modulu k počítači typu PC bylo tedy zapotřebí provést převedení napěťových úrovní CMOS na úroveň RS-232-C a naopak. Toto je provedeno pomocí obvodu AD 232 (výrobek fy. Analog Device), který při napájecím napětí +5V převede signály na požadovanou napěťovou úroveň. Obvod obsahuje násobič napětí +5V na +10V a invertor napětí +10V na -10V. Zapojení obvodu viz. obrázek.



Obrázek č.2 Schema zapojení sériové linky

2.4 Analogový vstup

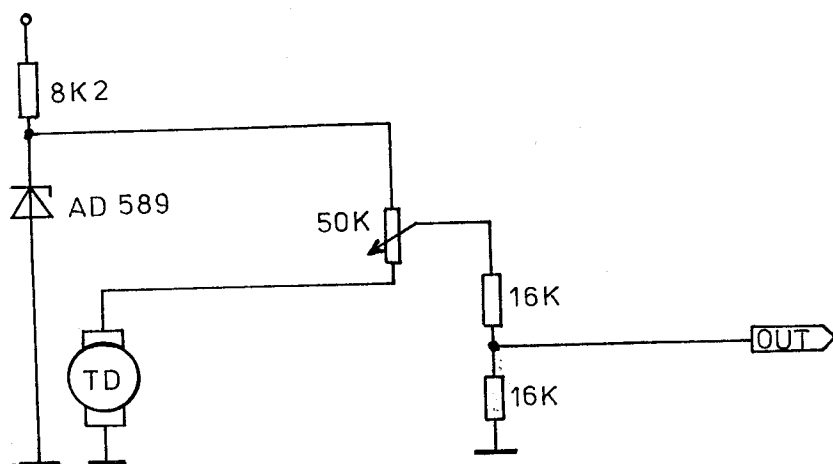
Pro připojení tachodynamu k řídicímu modulu bylo nutno vybavit modul vstupem $\pm 10V$. Tachodynamo připojené k stejnosměrnému elektromotoru dává na výstupu napětí lineárně závislé na otáčkách. Výrobce udává, že jsou to $2V/1000 \text{ min}^{-1}$. Při maximálních otáčkách, které lze dosáhnout při maximálním napětí kotvy motoru a minimálním buzení motoru, lze na výstupu změřit napětí okolo $9V$, což leží v daném rozsahu.

Jestliže se motor otáčí na jednu stranu, je na výstupu tachodynamu kladné napětí, když se točí na druhou stranu, je na výstupu tachodynamu záporné napětí. Toto bylo potřeba také vyřešit.

Modul IMM552 má integrován A/D převodník převádějící kladné napětí v rozmezí $0V$ až $2,5V$ na číslo v rozsahu 0 až 1024 (rolišení 10 bitů). Napětí z tachodynamu je nejprve upraveno odporovým děličem tak, aby při vstupní úrovni $-10V$ bylo na výstupu děliče napětí $-1,2V$. Toto napětí je spolu s referenčním napětím $+1,2V$, které je vytvořeno z napájecího napětí pomocí zenerovy diody (typ AD 589), přivedeno na odporový dělič, kde dojde k posunutí záporného napětí na úroveň kladného.

Toto napětí je definováno vztahem:

$$U_v = (U_{ref} - U_{tv}) \frac{R_2}{R_1 + R_2} + U_{tv}$$



Obrázek č.3 Schema zapojení analogového vstupu

Při U_{tv} měnícím se v rozsahu $-10V \div +10V$ je $U_v = 0V \div 1.2V$. Všechna napětí jsou uvažována proti zemi. Jako ochrana analogového vstupu je zapojena zenerova dioda D (2.7V) a pro odstranění nežádoucích záskmitů napětí U_v filtrační kondenzátor C.

2.5 Číslicový vstup

Dalším bodem zadání této práce bylo připojení inkrementálního snímače polohy na řídicí modul IMM552, aby bylo možno řídit otáčky motoru, nebo polohu (o kolik otáček (jaký úhel) se má rotor otočit), na základě signálů vystupujících z tohoto snímače.

Inkrementální rotační snímač je typu IRC 120/2500. Na jeho výstupech lze odebírat signál ve tvaru pravoúhlých pulsů v úrovni TTL. Pro řízení motoru jsem použil kontaktů

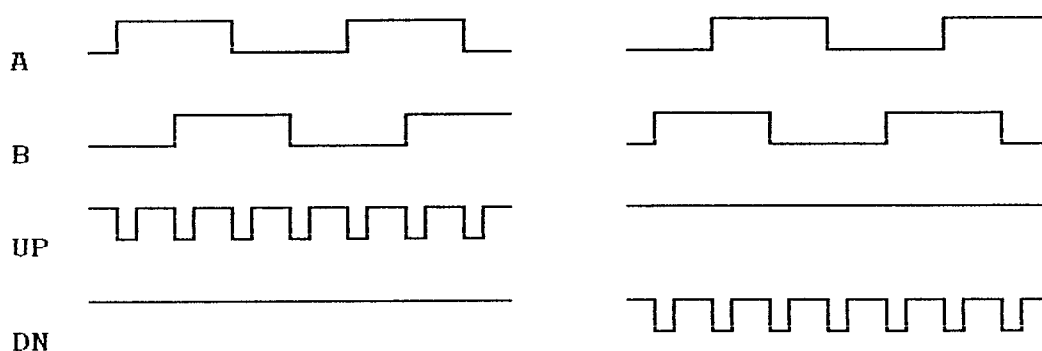
V a G připojovacího konektoru, z kterých vystupují pravouhlé pulsy fázově posunuté o 90 stupňů (elektrických). 1 otáčka odpovídá 2500 pulsů na příslušném vývodu.

Nechť na výstupu V je fáze A a na výstupu G je fáze B. Předbíhá-li fáze A fázi B, točí se motor jedním směrem a naopak opožďuje-li se fáze A za fázi B točí se druhým směrem.

Pro připojení na reverzibilní čítač (je možné čítání na obě strany) bylo nutno navrhnout vyhodnocovací obvod, který by:

- a) odstranil hazardy na impulsech,
- b) měl dva výstupy, přičemž by jeden byl při čítání v úrovni log 1 a na druhém by byly pulsy a naopak (závislost na směru otáčení) pro připojení na reverzibilní čítač.

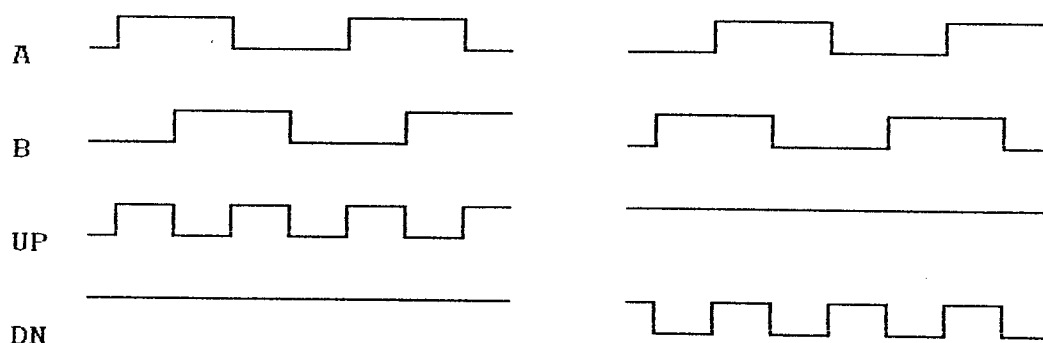
Při návrhu tohoto obvodu jsem vyšel ze zapojení uveřejněném v [6]. Skládá se z multiplexeru 74153 a paměti typu LATCH 4042 (CMOS) a vyrábí 4 impulsy na periodu vstupních signálů. Funkce je znázorněná na obrázku.



Obrázek č.4 Průběhy signálů na vstupech a výstupech zapojení dle [6].

Rozhodl jsem se však toto zapojení pozměnit z důvodů velkého počtu generovaných pulsů (4 x vstupní) a také proto, že jsem neměl k dispozici obvod 4042.

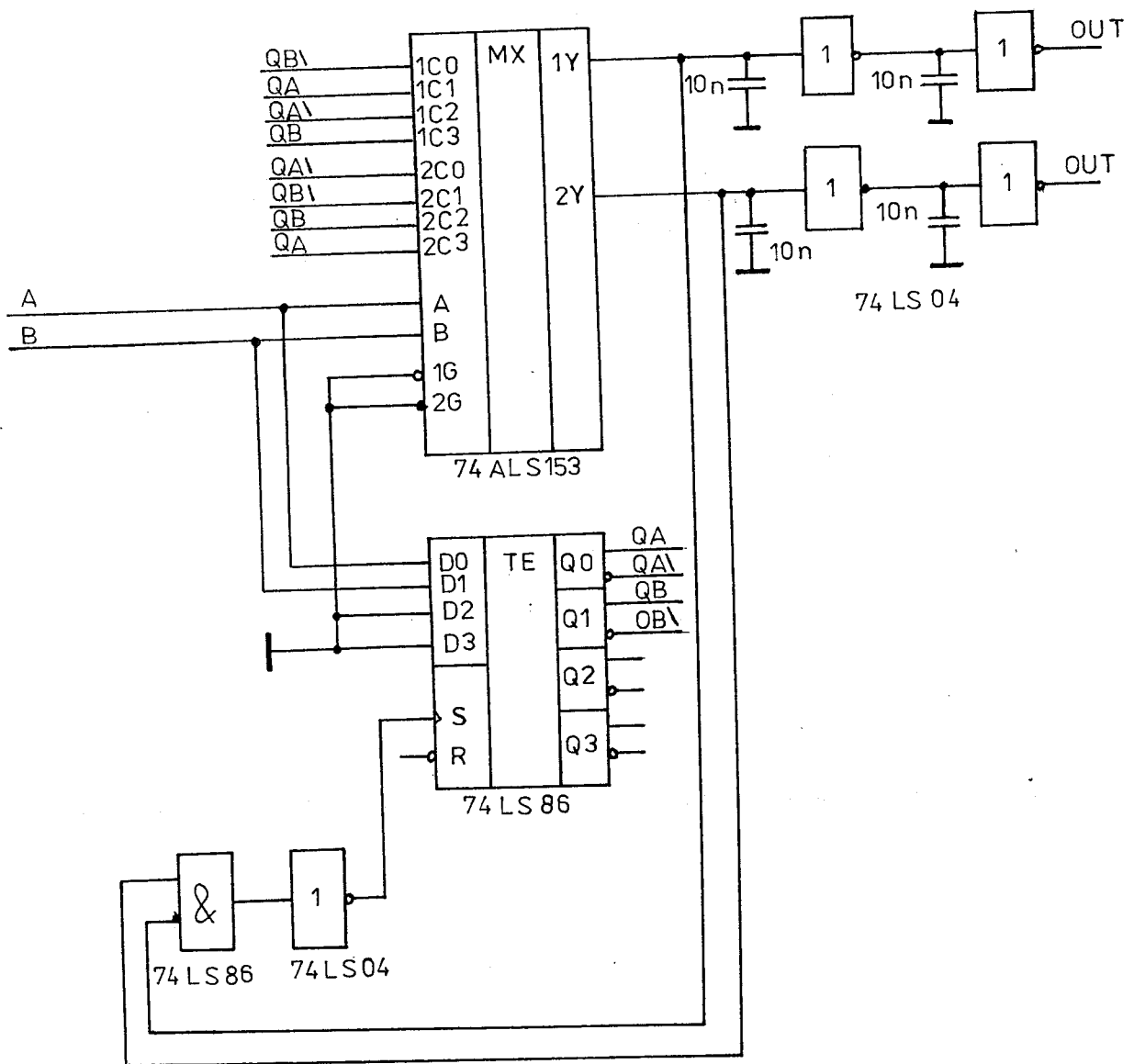
Základem mého zapojení je rovněž multiplexer 74153 a dále pak obvod 74175 (4 x klopný obvod D), 7486 (4 x 2 XOR) a jeden invertor (7404). Obvod generuje na výstupu 2 impulsy na periodu vstupních signálů. Funkce je zřejmá z obrázku.



Obrázek č.5 Průběhy signálů na vstupech a výstupech mnou navrženého zapojení

Při použití snímače s dělením 2500 na jednu otáčku obvod umožní vyhodnotit 5000 poloh (na otáčku), tj. polohování po $0,072^\circ$, což pro účel této práce stačí. Zapojení je vyobrazeno na obrázku č.6.

Jeden výstupu multiplexeru je v úrovni log 1 zatímco na druhém jsou pravoúhlé pulsy. Pulsy jsou úplně bez hazardů avšak na druhém výstupu je log 1 se zákmity k log 0 majících



Obrázek č. 6 Schema zapojení obvodu pro IRC čidlo

tvar ostrých špiček. Tento jev jsem odstranil zapojením dvou filtrů tvořených v sérii zapojeným kondenzátorem, invertorem, kondenzátorem a invertorem. Invertor má funkci tvarovače. Na výstupech jsou pak log 1 a pulsy bez hazardů.

Takto upravené výstupy jsou připojeny na hodinové vstupy revrzipbilního čítače zapojeného s ještě dalšími 3 čítači do dekády. Při výpočtu, kolik je zapotřebí bitů čítače, jsem vyšel z výsledků testování tohoto zařízení při regulaci otáček motoru na základě vzorkování měřené veličiny tachodynamu. Zde jsem se z důvodu software nemohl dostat s periodou vzorkování pod 50 ms. Po rozvaze o maximálním počtu impulsů na nejmenší periodu vzorkování a vzhledem k dynamickým vlastnostem soustavy jsem došel k závěru, že je nutné použít čítač o rozsahu 16 bitů. Pro vlastní čítání mohu využít rozsah 15 bitů, přičemž poslední bit vlastně slouží k rozlišení směru otáčení. Při maximálních otáčkách pak čítač vystačí na 0,45 s, což je tedy největší perioda vzorkování. Dolních osm bitů je připojeno na port 4 a horních osm bitů na port 1 řídicího modulu IMM552.

2.6 Výkonová část zařízení

Výkonovou část tvoří dva samostatné obvody:

- a) obvod kotvy
- b) obvod buzení

2.6.1 Obvod buzení

Jelikož toto řešení neklade požadavek na reverzaci napájení budícího vinutí, je zapojení velmi jednoduché. Je tvořeno darlingtonovou dvojicí PNP, jako ochrana proti napěťovému přetížení vlivem indukčnosti kotvy při spínání tranzistoru je zapojena rychlá křemíková dioda.

Obvod je oddělen od řídicí elektroniky optronem, aby se zamezilo průniku cizího napětí a následnému zničení integrovaných obvodů. Zapojení je na obrázku č.7.

Vše ?

2.6.2 Obvod kotvy

Jestliže je motor, který má reverzovat otáčky řízen bez reverzace napájení budícího vinutí, je nutno mít možnost reverzovat napájení kotvy. Tento požadavek velmi dobře splňují mostová zapojení. Při konstrukci této části jsem vycházel ze zapojení doporučených v literatuře [9]. Bylo však nutno provést některé úpravy, aby bylo zařízení funkční.

Nejdříve jsem vyzkoušel zapojení složené ze čtyř výkonových tranzistorů PNP a dvou budících stupňů. Mělo však tu nevýhodu, že byla zátěž zapojena mezi emitor jednoho a kolektor druhého tranzistoru. Proto jsem raději použil jiné zapojení, které se též skládá ze čtyř výkonových tranzistorů, dvou PNP a dvou NPN, a dvou budících stupňů.

Kotva motoru je zapojena v diagonále mostu na kolektorech tranzistorů. Přitom dva tranzistory v diagonále mostu zapojené proti sobě (např. T_2 , T_5) jsou buzeny z jednoho předzesilovacího stupně T_3 , viz. obrázek č.7. Podle polaroty vstupního napětí na svorkách A, \bar{A} je jeden vstupní předzesilovací stupeň vodivý. Je-li např. na svorce A kladné napětí, je vodivý tranzistor T_3 , který dále sepne tranzistory T_2 a T_5 . Tak se uzavře proudový okruh z kladného pólu napájecího napětí přes tranzistor T_2 , kotvu motoru a tranzistor T_5 . Při obrácené polaritě vstupního napětí sepne tranzistor T_4 a přes tranzistor T_1 a T_6 bude kotva motoru napájena proudem v opačném směru a kotva reverzuje otáčky. Paralelně k výkonovým tranzistorům jsou zapojeny rychlé křemíkové diody, které chrání tranzistory proti přepětí vlivem indukčnosti kotvy při přepínání.

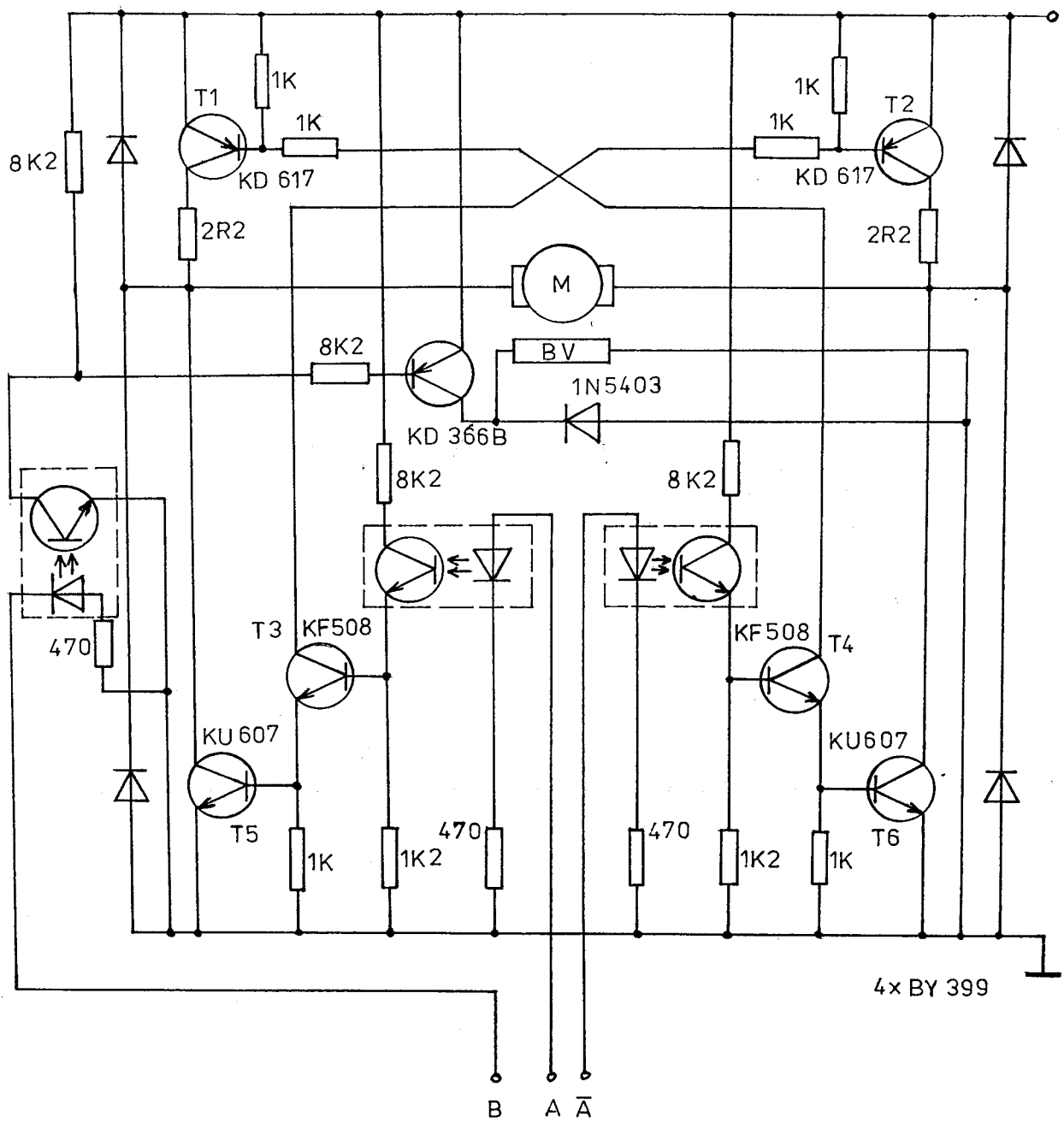
Toto zapojení dosahuje dobrých regulačních vlastností, je-li kotva napájena impulsy střídavé polarity. Činnost kotvy je potom závislá na rozdílu délky impulsů jednotlivých polarit. Motor se nepohybuje jsou-li impulsy v obou polaritách stejně dlouhé. V tomto vyváženém stavu je motor pevně držen rozdílem střídavě vyvozovaných momentů. Při rozběhu urychlují delší impulsy kotvu, přičemž zrychlující moment klesá se vzrůstající rychlostí otáčení. Kratší impulsy naopak při zpomalování působí intenzívní brzdění, které má charakter brzdění protiproudem.

Jelikož může nastat situace, že dříve než se zavře tranzistor T_2 se začne otvírat tranzistor T_6 a naopak, totéž

platí i pro tranzistory T_1 a T_5 , jsou do kolektorů tranzistorů zapojeny ochranné odpory $2R_2$, aby se zamezilo vzniku mžikových zkratů, a překročení mezních parametrů tranzistorů.

Zařízení je od řídicí části opět odděleno optronem, aby se zamezilo průniku cizího napětí do řídicí elektroniky a případnému zničení elektronických obvodů.

Na vývody A, /A a B se přivádějí impulsy z vývodů PWM0 a PWM1 modulu IMM552, které jsou upraveny (i výkonově) integrovaným obvodem pro připojení optronů.



Obrázek č.7 Schema zapojení výkonové části

3. Software

Při programování IMM552 je několik možností jaký software použít. Na modulu je v paměti ROM uložen interpret jazyka BASIC. Konkrétně je to IMM552 MICROBASIC 1.3. Tento interpret je prakticky shodný s jazykem MCS - 52 BASIC V1.1, který je dodáván při osazení procesorem 8052AH - Basic. Manuál k tomuto BASICu jsem měl k dispozici.

Dál výrobce dodává programové balíky pro ty aplikace, kde nestačí rychlost interpretu BASICu.

Jsou to:

Basic Compiler BXC51 - plně kompatibilní s interpretem 8052AH - Basic, použitelný pro všechny procesory řady 8051.

Assembler A51/MS-DOS - překladač zdrojového textu obsahující instrukce pro všechny procesory řady 8051.

BASITOLS - simulátor terminálu pro práci pod operačním systémem MS-DOS.

BASITOLS - PROFI - rozšířená verze BASITOLS pro profesinální použití.

SIM-51 (TESTE 51) - obrazovkově orientovaný interaktivní simulační program, který umožňuje simulaci a tím také testování programů napsaných pro procesor řady 8051.

Mikroprocesor 8051 je simulovaný počítačem PC.

MIC-51 - křížový překladač pro procesory typu 8051.

Pro práci s IMM552 jsem použil programovacího jazyka BASIC.
Chtěl jsem též použít jazyk C, ale musela by se odstranit
paměť ROM s jazykem BASIC.

3.1 Programování v BASICu

Po připojení napájecího napětí přejde software do AUTO
- BAUD rutiny, tedy nic se nestane, a je nutno nejdříve
stisknout klávesu SPACE a pak dojde k hlášení:

```
×IMM552 MICROBASIC 1.3×
```

```
READY
```

```
>
```

3.1.1 MICROBASIC - stručný popis

Čísla v pohyblivé řádové čárce (real)

- rozsah $\pm 1E-127$ až $\pm .999999999E+127$
- počítání probíhá s přesností 8 desínných míst

Celá čísla

- rozsah 0 (0000H) až 65535 (OFFFH)
- lze zadávat i v hexadecimálním tvaru 0F000H

Konstanty

- celý rozsah reálných čísel

Proměnné

Ve verzi 1.1 MCS BASIC-52 mohou mít délku maximálně 8 znaků. Nesmí obsahovat klíčové slovo.

Pole (Array)

V MCS BASIC-52 jsou povolena jednorozměrová pole. Index je v kulaté závorce. Maximální velikost pole je 254 prvků.

Proměnné jsou v MCS BASIC-52 statické, tj. při použití nového jména se pro tuto proměnnou vyhradí místo 8 byte, toto místo lze uvolnit jen použitím příkazu CLEAR, čímž se ale zruší všechny proměnné.

Zásobník

MCS BASIC-52 obsazuje prvních 512 byte externí paměti kde jsou implementovány dva zásobníky (CONTROL-STACK a ARGUMENT-STACK).

Řádkový editor

MCS BASIC-52 obsahuje minimální řádkový editor, proto lze jednou zadaný řádek změnit jen novým zadáním. Je však možné při psaní pomocí DELETE (7FH) smazat poslední zadaný znak nebo pomocí CONTROL-D smazat celý řádek. Při LIST nebo PRINT na konzole lze výpis zastavit pomocí CONTROL-S a pokračovat pomocí CONTROL-Q.

Příkazy

CONT - po zastavení běhu programu příkazem STOP nebo stlačením CTRL+C lze pokračovat tímto příkazem dále ve vykonávání programu, ne však byla-li provedena změna v programu, či nastala-li nějaká chyba

LIST - výpis programu na konzole

zastavení CTRL+S

pokračování CTRL+Q

ukončení CTRL+C

LIST čř - výpis programu od řádku číslo čř

LIST čř1-čř2 - výpis programu od čř1 do čř2

LIST# - výpis programu na tiskárnu (na sériovém portu)

předtím musí být udána přenosová rychlost pomocí

příkazu BAUD

LIST@ - opovídá příkazu LIST, jen s tím rozdílem, že výpis je proveden vlastním programem, program je volán od adr. 403CH. Dál musí být BIT 27H v interní paměti nastaven. Tento bit lze nastavit příkazem DBY(27H)=DBY(27H).OR.80H nebo programem v assembleru. Je-li LIST@ proveden bez nastavení bitu, následuje výstup na konzole. Při zavolání rutiny od adresy 403CH se nachází výstupní znak v akumulátoru a v R5 banky registrů 0. Rutina může změnit akumulátor a datapointr (DPTR), ale ne jiný registr banky 0.

NEW - zruší program v RAM, všechny proměnné nastaví na 0, dále se zruší všechny řetězce a BASIC - přerušeni.

Funkce hodin reálného času, reprezentace řetězců a interní STACK POINTER (adr. 3EH) se nezmění.

NULL [integer] - příkaz určí počet vyslaných NULL znaků (ASCII 00H), které se vyšlou po návratu vozíku (CR). Tím lze změnit rychlost vypisování. [Integer] může být v rozsahu 0÷255.

RUN - proměnné se vynulují, zruší se přerušeni BASICu a spustí se program od prvního řádku. Program může být v paměti RAM nebo ROM. Tímto příkazem nelze spustit program od určitého řádku. K tomu je nutno použít příkaz GOTO.

EPROM - FILE příkazy

Programy v BASICu mohou být v paměti EPROM (i víc programů).

RAM a ROM [integer]

Při provedení RAM se využívá paměť RAM pro současný program, což je normální režim.

ROM [integer] provede se výběr programu z EPROM, bez udání čísla se vybere ROM 1. Následně lze program modifikovat v RAM. Při pokusu modifikace v EPROM dojde k hlášení ERROR:PROM-MODE. Při přepnutí za provozu zůstane program v RAM zachován. Při provádění programu v EPROM lze využít celé místo v RAM pro uložení proměnných.

XFER - vybraný program se zkopíruje do RAM a nastaví se mód RAM.

PROG - příkazem se překopíruje momentálně vybraný program (RAM/EPROM) do EPROM. BASIC přitom ukáže číslo pod kterým ho uloží.

3.1.2 Popis příkazů BASICu

BAUD - nastavení přenosové rychlosti (bez použití je rychlost cca. 1Bd).

Formát: 1 startbit

8 datových bitů

2 stopbity bez parity

CALL [integer] - vykonání programu v assembleru od adresy integer, pro hodnoty 0+127 se hodnota násobí 2 a přičte 4100H (CALL 0 = 4100H, CALL 1 = 4102H).

CLEAR - všechny proměnné nastaví na nulu, zruší interupty a stacky, musí se pak provést ONEX1 nebo ONTIME. Pro ošetřování chyb ONERR [integer].

CLEAR1 - zablokuje přerušování BASICu

CLEAR2 - zablokuje registry

CLOCK1 - spustí vnitřní hodiny BASICu, proměná TIME je každých 5ms zvětšena o 0,005. Používá timer/counter 0 pro vyvolání přerušení, hodnota pro interval se počítá v proměné XTAL.

CLOCK0 - zastavuje hodiny BASICu, poslední údaj zůstává zachován.

DATA - označení dat

READ - čtení dat

RESTORE - nastavení ukazatele na první data

DIM - dimenze polí o více jak 10 prvcích

DO - UNTIL [real výraz] opakování, až je výraz TRUE

DO - WHILE [real výraz] opakování, dokud je výraz TRUE

END - konec programu

FOR TO [STEP] NEXT - smyčka, step < 0 není dovolen

GOSUB [číslo] RETURN - vyvolání podprogramu

GOTO [číslo] - skok na řádek číslo

ON [výraz] GOTO [č0,č1,...] - dle hodnoty výrazu skok na řádek č0, č1, ...

ON [výraz] GOSUB [č0,č1,...] - dle hodnoty výrazu skok na podprogram na řádku č0, č1, ...

INPUT - vstup z konzoly

ONEX1 [č] - skok na podprogram na řádku č a ukončený RETI při přerušení na INT1 CPU

ONTIME [výraz],[č] - povolení přerušení od času, TIME > výraz, pak skok na podprogram na řádku č, musí být ukončen RETI, přerušení jen po celých sekundách

PRINT nebo P. nebo ? - výstup na konzolu, je-li poslední znak čárka potlačí se CLRF.

TAB ([výraz]) - výpis na pozici danou výrazem.

SPC ([výraz]) - výpis mezer, jejichž počet se rovná výrazu.

USING (Fn) - určí výpis čísel v pohyblivé řádové čárce, n = počet míst, n = 0 následující nuly se nevypíší, n = 1,2 nemá význam (minimálně 3 místa).

USING (#.#) - výstup v pevném formátu

REM - komentář

RETI - konec podprogramu přerušení

STOP - zastavení programu

STRING [výr1], [výr2] - rezervace paměti pro řetězce, bez provedení tohoto následují po operacích s řetězcí chybová hlášení, výr1 = počet byte pro řetězec, výr2 = počet byte na jeden řetězec

3.1.3 Aritmetické, logické a relační operátory

V podstatě se shodují s ostatními programovacími jazyky.

3.1.4 Priorita operátorů

1. operace v závorkách
2. mocnění
3. negace
4. násobení a dělení
5. sčítání a odčítání
6. relační výrazy
7. log a (.AND.)
8. log nebo (.OR.)
9. log exklusiv or (.XOR.)

3.1.5 Speciální operátory

CBY([výraz]) - vrátí obsah pam. místa výraz v programové paměti

DBY([výraz]) - vrátí obsah pam. místa výraz v interní datové paměti

XBY([výraz]) - vrátí obsah pam. místa výraz v extrení datové paměti

GET - vrátí ASCII hodnotu stisknuté klávesy