TECHNICKÁ UNIVERZITA V LIBERCI
Fakulta strojní

DIPLOMOVÁ PRÁCE

Kontrolní systém rámové pily

1995

Jiří Beran

Technická univerzita v Liberci

obor 23-40-8

Automatizované systémy řízení výrobních procesů ve strojírenství

Katedra technické kybernetiky

Kontrolní systém rámové pily

UNIVERZITNÍ KNIHOVNA
TECHNICKÉ UNIVERZITY V LIBERCI
3146065795

Jméno a příjmení autora : **Jiří Beran**

Vedoucí práce : **Ing. Josef Grosman**

Rozsah práce a příloh :	**Počet stran :**	**44**
	Počet obrázků :	**14**
	Počet tabulek :	**8**
	Počet příloh :	**14**

VYSOKÁ ŠKOLA STROJNÍ A TEXTILNÍ V LIBERCI

Fakulta strojní

Katedra technické kybernetiky Školní rok: 1994/95

ZADÁNÍ DIPLOMOVÉ PRÁCE

pro Jiřího BERANA

obor 23-40-8 Automatizované systémy řízení výrobních procesů ve strojírenství

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

Název tématu:

Kontrolní systém rámové pily

Zásady pro vypracování:

1. Seznamte se s požadavky na kontrolní systém rámové pily (měřené technologické veličiny, signalizace, konstrukční požadavky)
2. Proveďte rešerši technických prostředků pro realizaci kontrolního systému (procesor, snímače, signalizace,možnosti odlaďování)
3. Navrhněte technické řešení kontrolního systému a jeho programové zabezpečení.
4. Realizujte navržené řešení

TECHNICKÁ UNIVERZITA V LIBERCI
Univerzitní knihovna
Voroněžská 1329, Liberec 1
PSČ 461 17

V 121/95 S
KTK/ASŘ
44 s, 60 s. příl.

Rozsah průvodní zprávy: cca 40 stran

Seznam odborné literatury:
[1] Provoz a údržba rámové pily
[2] Katalogy řídících systémů a součástek
[3] Články v časopisech

Vedoucí diplomové práce: Ing. Josef Grosman

Zadání diplomové práce: 31.10.1994

Termín odevzdání diplomové práce: 26.5.1995

VYSOKÁ ŠKOLA STROJNÍ A TEXTILNÍ FAKULTA STROJNÍ LIBEREC

L.S.

Ing. Petr Tůma, CSc.
pověřený vedením katedry

Prof.Ing.Jaroslav Exner, CSc.
děkan

V Liberci dne 31.10.1994

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

V Liberci dne 25. května 1995

Podpis ...Jiří Beran.....

Poděkování

Úvodem své diplomové práce bych rád poděkoval ing. Josefu Grosmanovi, vedoucímu mé diplomové práce za pomoc, kterou mi při jejím vytváření poskytl.

Seznam použitých zkratek

KS - Kontrolní systém
RP - Rámová pila
SC - Servocontroler
SM - Servomotor
PC - Personální počítač

Obsah

1. ÚVOD 7
2. POPIS RÁMOVÉ PILY 8
   2.1 Popis ovládání rámové pily 9
   2.2 Popis havarijních stavů, jejich příčiny, indikace a řešení 9
   2.3 Požadavky na kontrolní systém 11
   2.4 Konstrukční požadavky 12
3. REŠERŠE TECHNICKÝCH PROSTŘEDKŮ 13
   3.1 Procesory 13
   3.2 Paměti 14
   3.3 Snímače 16
   3.4 Zobrazovací prvky 17
   3.5 Signalizační prvky 17
4. VARIANTY ŘEŠENÍ 18
   4.1 Varianta 1. KS na základě PC 18
   4.2 Varianta 2. KS na základě procesoru 8051 19
5. POPIS REALIZACE VYBRANÉ VARIANTY 20
   5.1 Popis procesoru 8031 22
   5.2 Popis obvodu 27C256 25
   5.3 Popis obvodu 6264 25
   5.4 Popis obvodu 8255A 25
   5.5 Popis obvodu 9346A 26
   5.6 Popis výstupních bloků 27
   5.7 Popis obvodu 8253 27
   5.8 Popis snímače otáček 28
   5.9 Popis snímače teploty 28
   5.10 Popis převodníku odpor -> kmitočet 29
   5.11 Popis multiplexeru 74150 31
   5.12 Popis obvodu M5451 31
   5.13 Popis obvodu ICL 232 32
   5.14 Popis servocontroleru 32
   5.15 Popis ukazatele otáček 33
   5.16 Návrh schematu a plošného spoje 34
   5.17 Výběr vhodného vyššího programovacího jazyka 35
   5.18 Testovací program pro kontrolní systém 37
6. ZÁVĚR 42
7. SEZNAM POUŽITÉ LITERATURY 43
8. SEZNAM PŘÍLOH 44

1. Úvod

Dnes již není mnoho oblastí lidské činnosti kde by počítače nenašly uplatnění. Setkáváme se s nimi téměř na každém kroku. Počítač přestává být pouhým cílem tvůrčí lidské činnosti a stává se jejím prostředkem, nebo spíše jejím aktivním pomocníkem. Díky miniaturizaci, které se podařilo vtěsnat celý počítač do jednoho čipu se dnes můžeme s počítačem setkat i ve výrobcích běžné průmyslové spotřeby, jako jsou televizor nebo pračka.

Zatímco pro dřívější dobu byl charakteristický masový nástup počítačů různých konstrucí vzájemně většinou neslučitelných, dnes je to především snaha po standardizaci a unifikaci jednotlivých komponent. Tato snaha se příznivě promítá do cen výpočetní techniky.

Jednou z oblastí, kde použití počítače není běžnou záležitostí, je dřevozpracující prvovýroba. Použití automatizační techniky v této oblasti má své specifické rysy, které jsou dané charakteristickými vlastnostmi zpracovávané suroviny. Dřevo jako přírodní biologický materiál je velmi nepravidelné co se týká tvaru i kvalitativních hmotových parametrů.

Z uvedených skutečností vyplývá následující orientace automatizační techniky v dřevozpracujícím průmyslu:

- odstranění fyzicky namáhavé práce a její zefektivnění
- zvýšení bezpečnosti práce
- snížení ztrát energie
- maximální využití a zhodnocení dřevní suroviny

Cílem této práce bylo ukázat, že i v tak specifickém výrobním procesu může být použití počítače přínosem.

První část této práce je věnována stručnému popisu rámové pily, jejímu ovládání a popisu nejčastěji se vyskytujících havarijních situací. Je zde proveden rozbor požadavků kladených na kontrolní systém rámové pily.

V další části je zpracována stručná rešerše technických prostředků používaných v automatizační technice a při konstrukci číslicových systémů

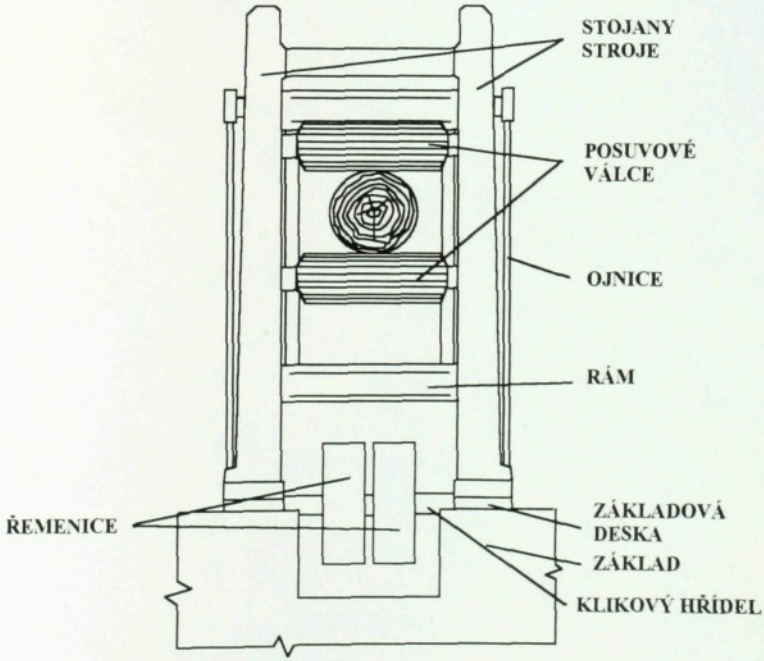
Ve čtvrté kapitole jsou zpracovány dvě varianty řešení kontrolního systému a jejich technicko ekonomické porovnání.

Pátá část je věnována popisu realizace vybrané varianty. Je zde proveden podrobný popis jednotlivých součástí ze kterých je kontrolní sysytém sestaven a popis programu, který slouží k ověření funkcí jednotlivých obvodů.

2. Popis rámové pily

Rámová pila je nejužívanějším strojem pro opracování kulatiny. Nejběžněji používaná je rámová pila svislá s pohonem spodním. Základy stroje jsou umístěny pod podlahou. Pevně se základem je spojena základová deska. Je ze šedé litiny z jednoho kusu a spojuje stojany rámovky se základem. Jsou na ní umístěny hlavní ložiska stroje - ložiska klikového hřídele.

Oba stojany jsou spojeny horní a spodní příčkou. Na stojanech jsou umístěna vodící ložiska pilového rámu. Na rámu (z obou stran a ve všech čtyřech rozích) jsou upevněna smykadla, která kloužou po kovových vedeních. Pilový rám slouží k zavěšování a napínání pilových listů a pohybuje se s nimi jako jeden celek nahoru a dolu. skládá se z horní a spodní rámové příčky a rámových sloupků. Pohyb rámu je zajištěn pomocí dvou ojnic. Na klikovém hřídeli, který je umístěn v ložiskách v základové desce, jsou nalisovány dva setrvačníky. Dále jsou na něm umístěny dvě řemenice - pevná a volná po kterých se přesouvá hnací řemen. Pevná řemenice slouží k pohonu stroje.



Obr. 2.1 Celkové uspořádání rámové pily.

Volné řemenice je použito při zastavování stroje. K pohonu je použito elektromotoru o příkonu 35 kWh. Přísun kulatiny do řezu se děje pomocí dvou dvojic posuvových válců, které jsou připevněny na obou stojanech. Válce jsou poháněny přes řetězový převod elektromotorem s plynule proměnlivým počtem otáček. Celkové uspořádání RP je na obr. 2.1.

2.2. Popis ovládání rámové pily

K ovládání RP slouží ovládací pult, zařízení pro přesun řemene a zařízení pro regulaci posuvové rychlosti.

Ovládací pult

Ovládací pult slouží k ovládání hlavního elektromotoru, k ovládání posuvového elektromotoru a je zde umístěno i tlačítko signalizační houkačky. Dále jsou zde umístěny ovládací přepínače přítlačných válců.

Zařízení pro přesun řemene

Přesouvací zařízení slouží při spouštění a zastavování stroje. Řídí se ručním kolem ze stanoviště u rámovky. Kolo je nasazeno na hřídeli, na jehož druhém konci je naklínován pastorek. Pastorek zapadá do hřebenu vysouvací tyče, která posouvá přesouvací vidlicí.

Zařízení pro regulaci posuvové rychlosti

Posuv je poháněn trojfázovým komutátorovým elektromotorem s napájeným statorem.(system Winter -Eichberg) . Otáčky elektromotoru jsou regulovány velikostí napětí na rotoru. Velikost napětí rotoru je regulována regulačním transformátorem.

2.3. Popis havarijních stavů, jejich příčiny,indikace a řešení

Havarijním stavem je myšlena taková situace, kdy hrozí bezprostřední nebezpečí poškození stroje, nebo hrozí, že se stroj dostane do takové situace, kdy bude muset být použito nestandardního postupu k jeho opětovnému uvedení do provozu.

Nejčastěji se vyskytující havarijní stavy:

1.- Přehřátí některého ložiska nebo motoru
2.- Náhlý pokles otáček za plného posuvu
3.- Nadměrné prokluzování řemene

Přehřátí některého ložiska nebo motoru

Přehřátím některého ložiska se rozumí nárůst jeho teploty nad povolenou mez. Vpraxi se udává, že teplota ložiska by neměla překročit teplotu kolem 60°C. Se zvyšující teplotou se zhoršují mazací vlastnosti oleje a mohlo by dojít k zadření ložiska. Nadměrné zahřívání ložiska může být způsobeno poruchou mazání nebo špatným seřízením jeho vůlí.

Zadření ložiska by mělo za následek jeho mechanické poškození nebo i poškození dalších součástí stroje. Takto vzniklé škody by si vyžádaly opravu velkého rozsahu a tím i několikatýdenní odstávku stroje.

V případě vodících ložisek rámu je přehřátí indikováno vznikem kouře a snížením výkonu stroje. Dochází k přídavnému nesymetrickému namáhání pilového rámu a mohlo by dojít k poškození kluzáku.

Zadírání ložiska na klikovém hřídeli je indikováno pouze snížením výkonu stroje "stroj jde těžce", což však může být způsobeno i mnoha dalšími příčinami. Obsluha proto nemusí včas rozeznat hrozící nebezpečí. Zadření tohoto ložiska by si vyžádalo generální opravu celého uložení klikového hřídele.

Náhlý pokles otáček za plného posuvu

Příčinou náhlého poklesu otáček bývají často shluky suků, před kterými obsluha nesnížila rychlost posuvu.Další příčinou může být kroucení nebo svírání klády.
Během poklesu otáček dochází ke snížení řezné rychlosti, ale rychlost posuvu zůstává stále
stejná což vede k nárůstu řezné síly. Nárůstem řezné síly se zvýší příkon hnacího elektromotoru. Nadměrný příkon elektromotoru vede k jeho odpojení od sítě. Z tohoto je zřejmé, že celá situace je velice nestabilní a bez včasného zásahu obsluhy vede k zastavení. V případě, že by došlo k úplnému zastavení vlivem řezné síly, hrozí nebezpečí mechanického poškození stroje. Došlo-li k odpojení motoru od sítě není již možno pokračovat v řezání. Obsluha musí přesunout hnací řemen na volnou řemenici a stroj zastavit. Pokud obsluha nestihla přesunout hnací řemen na volnou řemenici musí pomocí ručního natáčecího zařízení stroj několikrát protočit a řemen přesunout ručně.

Tomuto meznímu stavu je možno zabránit včasným snížením posuvové rychlosti. Pokles otáček však bývá tak rychlý, že obsluha řeší tuto situaci stlačením tlačítka POSUV STOP a následným pomalým rozjezdem.

Nadměrné prokluzování řemene

Nadměrné prokluzování hnacího řemene není tak nebezpečné jako předchozí havarijní stavy, ale vede k jeho rychlejšímu opotřebení a ke zbytečným ztrátám energie a tím i ke snížení výkonu stroje. Nadměrný prokluz signalizuje sníženou odolnost stroje proti náhlému poklesu otáček.

2.4. Požadavky na kontrolní systém

Kontrolní systém musí plnit několik základních funkcí:

- Poskytovat obsluze aktuální a pravdivé informace o stavu stroje
- Včas upozornit obsluhu na odchylky od standartních hodnot měřených veličin
- Upozornit obsluhu na možnost vzniku havarijní situace

Možnosti Kontrolního systému (dále KS) jsou dány počtem a druhem snímaných veličin a počtem akčních členů schopných zasahovat do ovládání stroje. Z popisu havarijních stavů je patrno, že KS musí být schopen měřit teplotu ložisek stroje a teplotu hnacího elektromotoru. Dále je nutné měřit počet otáček klikového hřídele a počet otáček hnacího motoru. Je třeba, aby obsluha byla informována o rychlosti posuvu, z toho vyplývá požadavek snímat počet otáček posuvového motoru. KS musí být schopen rozpoznat v jakém režimu činnosti se stroj nalézá, aby nedošlo k chybnému rozpoznání havarijní situace.

Režimy práce rámové pily:

Režim klidu - Tento režim odpovídá stavu stroje před nebo po pracovní směně. Hnací motor je v klidu, hnací řemen je na volné řemenici. Otáčky posuvového motoru nemusejí být nulové.

Režim rozeběhu - Otáčky hnacího motoru nejsou nulové, hnací řemen je na volné řemenici nebo na pevné. Otáčky posuvového motoru jsou většinou nulové.

Režim plného výkonu - Otáčky hnacího motoru nejsou nulové, jejich velikost je v povoleném intervalu. Hnací řemen je přesunut na pevné řemenici. Otáčky posuvového motoru nejsou nulové.

Režim nestandardní situace - Tento režim se od režimu plného výkonu liší pouze počtem otáček posuvového motoru. Otáčky jsou mimo rozsah používaný při plném výkonu, popřípadě jsou nulové.

Režim havarijní situace - Situace, které předchází režim plného výkonu. Havarijní situace nastává při poklesu otáček klikového hřídele pod stanovenou mez nebo při přehřátí některého z ložisek.

Režim zastavování - Režim zastavování je jednoznačně určen přesunutím hnacího řemene z pevné řemenice na volnou.

Aby mohl KS rozpoznat režimy rozeběhu a zastavování je nutné snímat polohu přesouvací vidlice.

Zvláště důležitá je vhodná volba prostředků, kterými bude KS signalizovat obsluze stav stroje. Signalizace nesmí zatěžovat obsluhu více než je nezbytně nutné, nesmí rozptylovat její pozornost. Nesmí však nastat situace kdy obsluha přehlédne důležitou informaci. Proto je vhodné doplnit vizuální signalizaci signalizací zvukovou, zvláště pro případy havarijních stavů.

Z výše uvedeného vyplývají požadavky na počet a druh snímaných veličin:

- **měření teploty**	8 vodících ložisek rámu 2 ložiska klikového hřídele 1 teplota motoru
- **měření otáček**	otáčky klikového hřídele otáčky hnacího elektromotoru otáčky posuvového elektromotoru
- **měření polohy**	poloha přesouvací vidlice - dvouhodnotový vstup

2.5. Konstrukční požadavky na KS

Provoz RP se vyznačuje vysokou prašností a hlučností. Součásti a zařízení mechanicky spojené se strojem jsou vystaveny vybracím. Z tohoto se odvíjejí i konstrukční požadavky na KS. Je nutné, aby konstrukce KS odolávala prachu a vibracím. Mechanické pohyblivé části je nutno vhodně zakrytovat nebo jinak ochránit před působením prachu.

3. Rešerše technických prostředků

Aby bylo možno zodpovědně navrhnout uspořádání kontrolního systému je nezbytně nutné se nejprve seznámit s dostupnými technickými prostředky. Pod pojmem technické prostředky se rozumí počítače, snímače, signalizační prvky, převodníky a veškerá součástková základna. V současné je nabídka na trhu velice široká.

Nejrozšířenějšími personálními počítači se v poslední době staly počítače třídy PC. Ostatní počítače (např ATARI, COMODORE a pod.) byly úplně vytlačeny. Výhodami Počítačů PC je jejich modulární koncepce a dodržování kompatibility. Díky politice otevřenosti firmy IBM vyrábí dnes PC mnoho výrobců v různých kvalitativních i finančních hladinách. PC pracují většinou s procesory 386, 486 a PENTIUM. Běžné taktovací kmitočty jsou od 40 MHz výše. Základová deska PC je opatřena rozšiřujícími konektory - sloty. To umožňuje rozšířit snadno možnosti PC pomocí přídavných desek - tzv. přídavné karty. Dostupné jsou karty univerzálních vstupů a výstupů, analogových i digitálních. Počet analogových vstupů nebo výstupů bývá až 32, u digitálních až 144. Běžně dostupné jsou karty osazené až 16 relé a opatřené opticky oddělenými vstupy. Běžné jsou také různé paměťové nebo komunikační karty. Úplnější popis přídavných karet by přesáhl rozsah celé diplomové práce. Počítače PC jsou vyráběny také v průmyslovém provedení se zvýšenou odolností vůči vlivům prostředí.

3.1. Procesory

Srdcem každého počítače je procesor. Procesor je programovatelný sekvenční automat vykonávající strojové instrukce předepsané programem. Činnost procesoru při vykonávání instrukcí je řízena řadičem. Obvody řadiče spolu s obvody aritmetické jednotky tvoří část procesoru programátorovi nepřístupnou. Část přístupnou pak tvoří registry a obvody přerušovacího systému. Procesor komunikuje s okolím pomocí adresové, datové a řídící sběrnice. Adresová sběrnice určuje paměťové místo nebo periferní obvod se kterým bude probíhat přenos dat. Data jsou přenášena po datové sběrnici. Na řídící sběrnici předává procesor signály pomocí kterých zabezpečuje komunikaci s okolím. Procesory lze dělit podle různých měřítek:

Podle délky slova: na 4 bitové, 8 bitové, 16 nebo 32 bitové. Kromě speciálních aplikací se většinou delších slov nepoužívá. Procesory pracují ve dvojkové soustavě proto jsou používané délky slov některým z násobků 2^n. Některé procesory mající vnitřní strukturu např. 16 bitovou mohou pracovat s vnější sběrnicí pouze 8 bitovou, tím je umožněno využití obvodů 8 bitových mikropočítačových sad nižších vývojových stupňů.

Podle instrukčního souboru: na procesory s úplným instrukčním souborem a s redukovaným instrukčním souborem. Procesory s úplným instrukčním souborem bývají označovány CISC, procesory s redukovaným souborem RISC. Procesory CISC mají bohatší instrukční soubor, ale zpracování jedné instrukce trvá mnohem déle než u procesorů RISCových. Toto je způsobeno složitější strukturou řadiče procesoru.

Některé procesory mají v pouzdru zaintegrovány další obvody paměti RAM, EPROM, EEPROM, ovladače pro LCD displey a pod. Pokud je procesor sám schopen vykonávat činnost počítače bývá označován jako jednočipový počítač. Jsou určeny pro aplikace, kde by byl rozsáhlý mikropočitačový systém nevhodný pro své rozměry nebo spotřebu. Přehled nejpoužívanějších procesorů je v tabulce 3.1.

Tab. 3.1. Stručný přehled nejpoužívanějších procesorů

Procesor	Délka slova	Charakteristika	Výrobce
80386, 486	32 bitů	Univerzální procesor používaný v počítačích třídy PC a v průmyslových počítačích.	INTEL
8051	8 bitů	Jednočipový mikropočítač používaný pro konstrukci regulátorů a jednoduchých řídících členů.	INTEL
Z80	8 bitů	Univerzální procesor používaný při konstrukci personálních počítačů.	ZILOG
M68000	32 bitů	Univerzální procesor používaný při konstrukci personálních nebo průmyslových počítačů.	MOTOROLA

3.2. Paměti

Další nezbytnou součástí počítačů jsou paměťové obvody. Nejpoužívanějšími hledisky pro klasifikaci pamětí jsou : způsob přístupu, možnost změny dat a princip realizace paměťové buňky.

Z hlediska způsobu přístupu k buňkám se paměti dělí na:

- paměti s libovolným přístupem RAM (Random access memory)
- paměti se sériovým nebo sekvenčním přístupem SAM (serial access memory)

Z hlediska možnosti změny dat na:

- paměti PROM jejichž obsah je definován výrobcem. U těchto pamětí nelze obsah nijak změnit.
- paměti EPROM , které lze elektricky programovat ve speciálním přípravku. Obsah paměti EPROM lze vymazat elektromagnetickým zářením o vlnové délce kratší než 400 nm. Programovat nebo mazat lze jen celou paměť najednou. Počet opakování programovacího cyklu záleží na způsobu mazání paměti a pohybuje se řádově v desítkách až do sta.
- paměti EEPROM . Tyto paměti lze elektricky programovat i mazat přímo v zařízení pomocí speciálního vstupu. Počet programovacích cyklů je je kolem jednoho miliónu. Data jsou v paměti uschována po dobu zhruba 10 let.
- paměti SRAM. Označení SRAM znamená statická paměť RAM Do paměti SRAM lze libovolně zapisovat a libovolně z ní číst.Za tuto výhodu se platí ztrátou dat při výpadku napájecího napětí.
- paměti DRAM. Označení DRAM znamená dynamicky obnovovaná paměť RAM. Základní odlišnost od paměti SRAM spočívá v tom, že data se v paměti DRAM udrží pouze určitou omezenou dobu a je nutno je periodicky obnovovat. Po dobu obnovování není paměť schopna převzít nebo dodat data o čemž informuje procesor. Výhodou DRAM pamětí je jednoduchá stavba paměťové buňky a s tím spojená možnost vyrábět paměti velkých kapacit.

Tab. 3.2. Přehled nejběžnějších paměťových obvodů

Označení	Typ	Kapacita , přístupová doba
HY 6116ALP15	SRAM	2 k x 8 , 150 ns
MK 6264N70	SRAM	8 k x 8 , 70 ns
M 41256	DRAM	256 k x 1 , 100 ns
M 27C64	CMOS UV EPROM	8 k x 8 , 150 ns
M 27C256	CMOS UV EPROM	32 k x 8 , 150 ns
M 27C512	CMOS UV EPROM	64 k x 8 , 150 ns
ST 93C46	CMOS EEPROM	64 x 16 nebo 128 x 8
ST 24C02	CMOS EEPROM	256 x 8
ST 24C04	CMOS EEPROM	512 x 8

Stejné typy obvodů bývají vyráběny různými výrobci, většinou pod stejným číselným označením pouze se lišícím začátečními písmeny.

3.3. Snímače

- Aby mohl počítač reagovat na změny vnějšího prostředí musí být opatřen snímači. Snímače lze rozdělit do několika skupin podle toho jaké veličiny snímají nebo podle toho na jakém principu pracují.

Rozdělení podle snímané veličiny je následující:

- snímače polohy
- snímače zrychlení
- snímače průtoku
- snímače tlaku
- snímače vlhkosti
- snímače teploty
- snímače otáček
- koncové spínače

Popisovat podrobně všechny typy snímačů by nemělo smysl, s ohledem na požadované snímané veličiny (viz . kapitola Požadavky na kontrolní systém) se omezíme popis snímačů teploty a otáček .

Snímače teploty

Snímače odporové - termistory

U odporového snímače se vlivem teploty mění hodnota jeho odporu. Jejich nevýhodou je nelineární charakteristika. O kvalitě termistoru rozhodují především jeho tolerance, teplotní konstanta a teplotní rozsah. V prodeji je mnoho druhů termistorů lišících se především odporem, teplotním rozsahem a pod.

Snímače termočlánkové

Principem termočlánkového snímače je termoelektrický článek na jehož kontaktech vzniká napětí úměrné teplotě. Termoelektrický článek se skládá ze dvou vodičů z různých kovů (např. Cu,NiCu; PtRh,Pt; Fe,NiCu; a pod.) na koncích spojených. Zahřejeme-li každý spoj na různou teplotu začne termočlánkem protékat proud. Pomocí termočlánků je možno měřit teploty až do 1500 °C.

Snímače otáček

Můžeme podle principu činnosti rozdělit na analogové a impulsní. Typickým představitelem analogového snímače je tachodynamo. Tachodynamo je vlastně precizně zhotovený stejnosměrný elektromotor s permanentními magnety. Tachodynamo je schopno měřit pouze rychlost otáčení ne aktuální polohu.

Impulsní (inkremantální) snímače neměří přímo rychlost otáčení, ale signalizují úhlové přírůstky. Způsoby snímání mohou být indukční, fotoelektrické nebo magnetické.

Signály ze snímačů většinou nelze přímo zpracovat počítačem proto se mezi snímač a počítač zařazují převodníky. Převodníky dělíme podle charakteru převodu na analogo/digitální, digitálně/analogové a na převodníky upravující signál na standartizovanou úroveň.

3.4. Zobrazovací prvky

Zobrazovací prvky mohou být buď aktivní nebo pasivní. Pasivní zobrazovací prvky potřebují vnější zdroj světla. Patří mezi ně zobrazovací prvky pracující na základě tekutých krystalů. Aktivní zobrazovací prvky nepotřebují vnější zdroj světla, jsou to například obrazovkové displeje, LED displeje a pod.

Dále můžeme dělit zobrazovací prvky podle typu informace, kterou jsou schopny zobrazit. Může to být informace dvouhodnotová (kontrolky, LED diody), celočíselná (LED displej, LCD displej) nebo grafické informace (grafický displej, monitor a pod).

3.5. Signalizační prvky

Signalizační prvky mohou být vizuální nebo zvukové. Podle charakteru signalizované informace mohou být informační nebo výstražné.

Mezi vizuální signalizační prvky patří například signalizační kontrolky, výstražné majáky a pod.

Úplná rešerše technických prostředků pro výpočetní a automatizační techniku není možná, a není ani cílem této diplomové práce. Tato rešerše slouží spíše k získání přehledu a pro lepší orientaci v této oblasti.

4. Varianty řešení

Způsobů, kterými lze zrealizovat kontrolní systém je celá řada. Jednou z možností je sestavit KS ze standardizovaných dílů používaných v automatizační a měřící technice. Použitím těchto dílů lze zkrátit dobu návrhu na minimum. Veškeré tvůrčí úsilí lze soustředit na návrh programového vybavení. Toto na první pohled jednoduché řešení má i svá úskalí. Navržené zařízení může být technicky nadbytěčné, což se velice nepříznivě projeví na jeho ceně.

Další možností je navrhnout a zrealizovat vlastní kontrolní systém. Toto řešení není tak pohodlné jako při realizaci pomocí standardních dílů, ale umožňuje provést "návrh na míru".

Při návrhu KS jsem zvažoval dvě varianty :

1. KS na základě počítače PC

2. Vlastní návrh na základě procesoru 8051

4.1. Varianta 1. - KS na základě PC

Kontrolní systém je tvořen počítačem PC s procesorem 386DX rozšířeným o desku A/D převodníků, desku releových výstupů, digitálních vstupů a desku komunikační.

V tabulce 4.1. je uveden přehled hlavních dílů tohoto řešení. V tomto přehledu není zahrnuto zařízení, které by informovalo obsluhu o stavu měřených veličin, např. monitor nebo displej.

Výhodou této varianty je možnost jejího dalšího snadného rozšiřování pomocí přídavných desek.

Nevýhodou je poměrně vysoká cena. Další nevýhodou jsou větší rozměry a větší spotřeba energie oproti variantě 2.

Tab. 4.1 Přehled hlavních komponent a jejich cen pro variantu 1.

Komponenta	Popis	Cena
RACK 200	Průmyslové šasi	9910,-
SSC 386DX	Procesorová deska	9470,-
AX C104	Komunikační karta	1930,-
AX 7010	Klávesnice (prachotěsná)	1250,-
AX 5232	Deska A/D převodníků	6310,-

AX 5032	Deska 16ti releových výstupů a 16 optoizolovaných vstupů	9930,-
ACL 711B	Multifunkční karta	5680,-
	CELKEM	44480,-

4.2. Varianta 2. KS na základě procesoru 8051

Kontrolní systém je tvořen jednodeskovým počítačem s procesorem 8031. K procesoru jsou připojeny obvody vstupu a výstupu tak, aby byl schopen plnit požadované funkce. Jsou to především paralelní stykový obvod 8255 pro rozšíření vstupně/výstupních bran procesoru, obvod 8253 - trojnásobný čítač pro snímání otáček a obvody pro rozšíření paměti. V tabulce 4.2. je uveden přehled hlavních dílů tohoto řešení.

Výhodami této varianty jsou menší rozměry a spotřeba oproti variantě 2. Další výhodou je menší cena zařízení.

K nevýhodám patří horší možnosti dalšího rozšiřování zařízení.

Tab. 4.2 Přehled hlavních komponent a jejich cen pro variantu 2.

Komponenta	**Popis**	**Cena**
8031	Procesor	135,-
8255	Paralelní stykový obvod	110,-
8253	3 x Čítač	100,-
27256	EPROM Paměť	130,-
6264	RAM Paměť	120,-
Desky plošných spojů		1000,- (odhad)
Ostatní součástky		1000,- (odhad)
Ostatní náklady		1000,-
	CELKEM	4000,-

Vybrat jednoznačně lepší variantu není možné. Výhody a nevýhody jednotlivých variant jsou uvedeny v jejich popisu. Při porovnání nákladů je zřejmé, že varianta 2 je několikanásobně lacinější. Proto jsem volil pro realizaci variantu 2.

5. Popis realizace vybrané varianty

Byla zrealizována varianta kontrolního systému s procesorem 8031. Blokové schema znázorňující celkové uspořádání je na straně 21. KS je umístěn na dvou jednostranných plošných spojích, které jsou pomocí distančních sloupků spojeny v jeden kompaktní celek. Na jednom pl. spoji je umístěn procesor spolu se vstupními a výstupními obvody. Druhý pl. spoj tvoří čelní panel KS. Jsou na něm umístěny signalizační diody, dva LED displeje sestavené ze sedmisegmentových číslicovek a čtyři tlačítka sloužící k přepínání zobrazované informace na spodním displeji.

Signalizační diody jsou na čelním panelu rozmístěny do obrazce symbolizujícího rám a klikový hřídel stroje. V místě odpovídajícím poloze ložiska je vždy dvojice diod. Zelená, signalizující, že teplota je v povolených mezích a červená pro signalizaci překročení mezních hodnot. Velikost teploty je zobrazována na horním displeji, na kterém je možno zobrazit číslo v rozsahu 0 - 199. Spodní displej slouží k zobrazení počtu otáček klikového hřídele, velikosti posuvové rychlosti nebo velikosti prokluzu hnacího řemene. Na tomto displeji je možno zobrazit číslo v rozsahu 0 - 399.

Jádro systému je tvořeno procesorem doplněným podpůrnými obvody. Procesor byl doplněn o paměť programu, paměť dat a obvody zajišťující styk procesoru s okolím.

Paměť programu je tvořena obvodem EPROM 27C256 o kapacitě 32 kB , paměť dat obvodem RAM 6264 o kapacitě 8 kB.

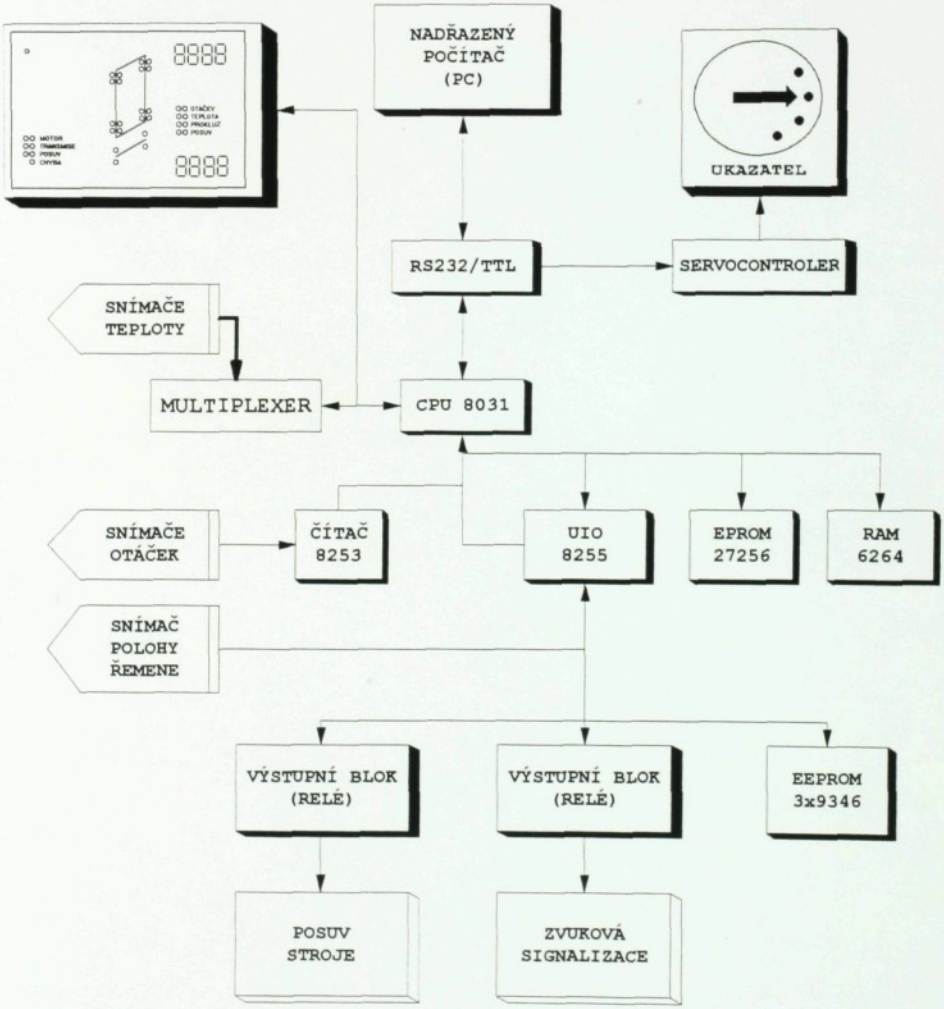
Počet vstupů a výstupů procesoru byl pomocí obvodu 8255 rozšířen o tři osmibitové paralelní porty. K tomuto obvodu je připojena paměť EEPROM , obvody výstupu a hradla typu AND pomocí kterých je přepínán signál TxD sériové linky.

Paměť EEPROM může být tvořena až třemi obvody 9346A . Tato paměť je nezávislá na napájecím napětí. Zde může sloužit např. pro úschovu nastavitelných hodnot (maximální teploty, minimálních otáček, parametry seriové linky a dalších).

Pomocí sériové linky komunikuje procesor s nadřazeným počítačem a servocontrolerem, který ovládá servomotor ukazatele otáček klikového hřídele. Na základové desce jsou dva konektory CANON 9 pro připojení obou linek. Signál RxD linky určené pro komunikaci s nadřazeným počítačem je přiveden na pin P3.0 procesoru. Signál RxD určený pro komunikaci se servocontrolerem je přiveden na pin P1.0. Signál TxD je přiveden na vstup dvou hradel AND, druhé dva vstupy hradel jsou ovládány výstupy PC6 a PC7 obvodu 8255A. Takto je možno signál TxD použít pro jednu z obou linek nebo pro obě najednou. Pro převod signálů mezi úrovněmi TTL a RS232 je použito obvodu ICL 232.

Obvody výstupu jsou dva. Každý je tvořen tranzistorovým zesilovačem ovládajícím relé. Relé jsou opatřena třemi trojicemi přepínacích kontaktů. Rozpínací kontakty jednoho z relé jsou zapojeny do série s tlačítkem POSUV STOP na ovládacím

BLOKOVÉ SCHÉMA KONTROLNÍHO SYSTÉMU



pultu RP, což umožňuje kontrolnímu systemu zastavit posuv stroje. Spínací kontakty druhého relé jsou zapojeny paralelně s tlačítkem HOUKAČKA .

Signály ze snímačů otáček jsou zpracovávány obvodem 8253. Tento obvod je tvořen třemi šestnáctibitovými čítači. Povolovací vstupy GATE obvodu jsou přes odpor 10k spojeny s napětím +5V. Signál přivedený na vstup čítače 0 je přiveden i na vstup INT0 procesoru. Otáčky jsou snímány snímačem MH1SS1, což je tzv. Hallova sonda - obvod reagující na přítomnost magnetického pole.

Oba programovatelné obvody (8255 i 8253) jsou připojeny k procesoru tak, že jejich vnitřní registry jsou adresovatelné jako paměťová místa. Výběrové signály CS* těchto obvodů jsou ovládány obvodem 74ALS138 . Adresovací vstupy tohoto obvodu jsou připojeny na adresové vodiče A12, A13, A14 procesoru. Výběrový vstup G1 je připojen na A15, vstupy G2A a G2B jsou trvale spojeny se zemí. Obvod 74ALS138 realizuje funkci adresového dekodéru.

Pro snímání teploty je použito odporového teplotního čidla KTY 10 - 6. Čidlo je připojeno na vstup převodníku odpor-kmitočet. . Signály z převodníků jsou multiplexovány obvodem 74LS150. Díky tomuto zapojení je možno měřit teplotu až šestnácti míst. Adresovací vstupy obvodu jsou připojeny na P1.4 až P1.7 procesoru. Výstup z obvodu 74LS150 je přiveden na PB.2 obvodu 8255.

5.1. Popis procesoru 8031

Jedná se o osmibitový mikroprocesor, který má přímo na čipu implementovánu paměť, čítače a další podpůrné prostředky.Obvod může vykonávat úkoly mikropočítače bez dalších podpůrných obvodů, proto je též označován jako "jednočipový mikropočítač".Mikropočítač je zapouzdřen do plastového, nebo keramického pouzdra DIL se čtyřiceti vývody.

Mikropočítač obsahuje:	Centrální procesorová jednotka	8 bitů
	Vnitřní datová paměť	00H-FFH
	Zdroj hodin - oscilátor	
	Dva 16 bitové čítače	
	Přerušovací systém	
	Sériový kanál	
	Čtyři vstupně / výstupní kanály	

Centrální procesorová jednotka je osmibitová.Její instrukční soubor obsahuje následující skupiny instrukcí: Aritmetické instrukce

Instrukce logických operací

Instrukce bitových operací
Instrukce přesunů dat
Instrukce větvení programu

Způsoby adresování jsou dva: přímé a nepřímé.

Přímo lze adresovat: 1.- Čtyři banky registrů po osmi registrech.
2.- 128 bytů vnitřní datové paměti a oblast SFR registrů
3.- bitově přístupnou oblast na adresách 20H-2FH

Nepřímo lze adresovat celou vnitřní paměť dat , vnější paměť dat a paměť programu.Vnitřní datová paměť je rozdělena na dvě části. V dolní polovině adresového prostoru 00H - FFH je 128 bytů běžné uživatelské paměti .V horní polovině je umístěna sada registrů SFR. (Speciální funkční registry).Tyto registry jsou určeny k řízení běhu programu, vestavěných periférií a přerušovacího systému.

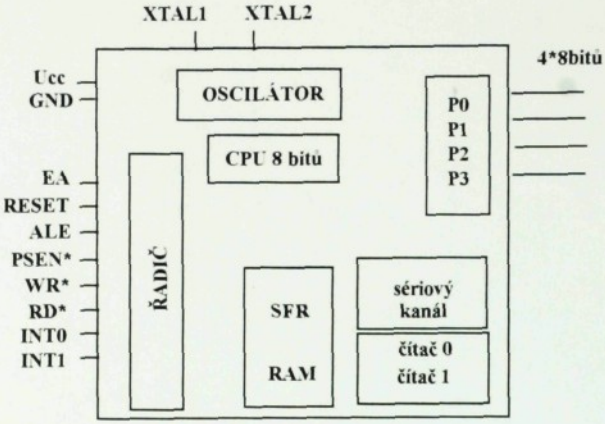
Zdroj hodin - oscilátor vyžaduje ke své činnosti pouze připojení krystalu na vývody XTAL1 a XTAL2. Běžná taktovací frekvence procesoru je 12 MHz, ale na trhu jsou dostupné i verze pro 50 MHz. Při použití krystalu 12 MHz trvá jeden instrukční cykl 1 mikrosekundu.
Vnitřní hodiny lze zabezpečit i z vnějšího zdroje. Signál z tohoto zdroje se připojuje k vývodu XTAL1.

Oba 16 bitové čítače mohou pracovat s vnitřními i vnějšími hodinami. Čítač pacující s vnitřními hodinami, odvozenými od hodin oscilátoru bývá označován jako "časovač".Pro čítač pracující s vnějšími hodinami se užívá termínu "čítač vnějších událostí". Oba čítače čítají vzestupně a lze je přednastavit na libovolnou hodnotu.Čítače lze nastavit do čtyř pracovních módů.Programově se čítače řídí pomocí dvou registrů TMOD a TCON. Čítač 1 může být použit pro generování přenosové rychlosti sériového kanálu.

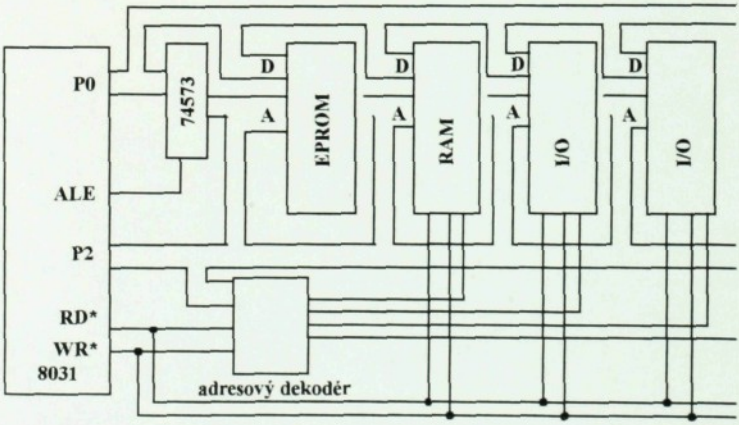
Přerušovací systém mikropočítače je dvouúrovňový. Je schopen zpracovat dvě vnější žádosti (INT0 a INT1) a dvě vnitřní žádosti způsobené přetečením čítače 0 nebo 1. Dále je schopen zpracovat žádost aktivovanou sériovým kanálem.

Sériový kanál je plně duplexní. (příjem i vysílání může probíhat současně) Sériový kanál lze pomocí bitů SM1 a SM0 nastavit do čtyř módů. V módu 1 a 3 je přenosová rychlost generována pomocí čítače 1.

Všechny čtyři vstupně / výstupní kanály jsou obousměrné. Ve směru výstupu jsou vybaveny vyrovnávací pamětí. Signály portu P0 a P2 jsou použity pro generování signálů vnější adresové a datové sběrnice. Signály portu P3 mohou mít další funkce. Při čtení je nutno nejprve do odpovídajícího bitu výstupní vyrovnávací paměti zapsat jedničku.



Obr. 5.1. Vnitřní struktura procesoru 8031



Obr. 5.2. Realizace vnějších pamětí a připojení periférií.

5.2. Popis obvodu EPROM 27C256

Obvod 27C256 je elektricky programovatelná paměť o kapacitě 32 kB. Je vyráběna unipolární technologií HMOSE. Podstatou paměťové buňky je tranzistor s izolovanou řídicí elektrodou, jejíž náboj určuje hodnotu zapamatovaného bitu. Časová konstanta úniku náboje je asi 100 let. V praxi se však počítá s udržením informace po dobu 5 až 10 let. Obsah EPROM lze vymazat elektromagnetickým zářením o vlnové délce menší než 400 nm. Výhodami tohoto obvodu jsou jednotné napájecí napětí +5V a možnost sníženého příkonu v režimu STANDBY při deselekci součástky. Vybavovací doba běžného provedení obvodu je kolem 120 ns.

5.3. Popis obvodu RAM 6264

Obvod 6264 je statická paměť typu RAM (paměť s libovolným přístupem) o kapacitě 8 kB. Tento typ paměti umožňuje zápis i čtení, ale při výpadku napájecího napětí jsou všechny informace uložené v této paměti ztraceny.

Obvod je opatřen dvěma výběrovými signály CS1* a CS2. Dále je opatřen signálem pro zápis WE* a signálem pro čtení OE*. Adresa paměťové buňky je určována pomocí adresových vodičů A0 - A12.

5.4. Popis obvodu 8255A

Víceúčelový programovatelný obvod pro vstup a výstup obsahuje 3 osmibitové brány, jejichž funkce může být naprogramována. Každá brána může být vstupní nebo výstupní. Brány A a B pracují vždy jako celek, brána C je rozdělena na dvě poloviny. Horní polovina je ve skupině s branou A, spodní polovina je ve skupině s branou B. Bity brány C lze ovládat jednotlivě.

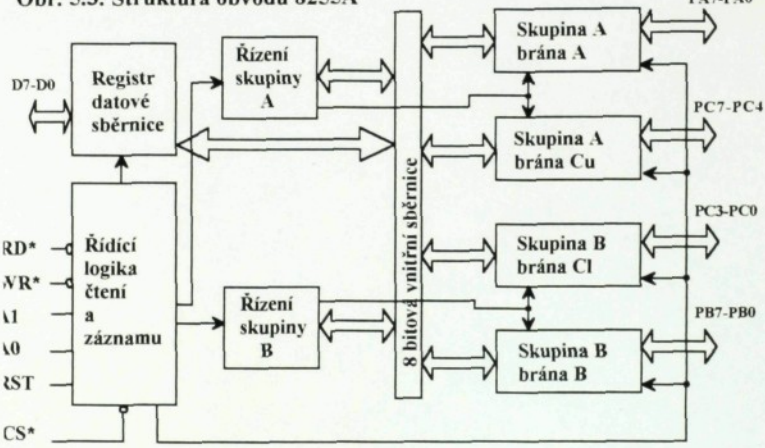
Přes vyrovnávací registr je obvod spojen s datovou sběrnicí procesoru.(vývody D7-D0) Po datové sběrnici přijímá řídící slova (CW) i výstupní data a toutéž cestou předává vstupní data. Logika čtení a záznamu řídí všechny interní a externí přesuny dat a zápis CW. Přijímá bitové signály z adresové a z řídící sběrnice a generuje příkazy pro oba bloky řízení skupin bran. Selekční signál CS* povoluje komunikaci mezi obvodem a procesorem. Bity A1 a A0
slouží k adresaci jedné ze tří bran nebo registru CWR (registru řídicího slova). Existují 3 mody činnosti volitelné programem:

Mod 0 : Každá brána může být nezávisle nastavena jako vstuní nebo výstupní.

Mod 1 : Brány A a B pracují jako vstupní nebo výstupní v korespondenčním režimu. Každá z těchto bran má přiřazeny 3 bity brány C.

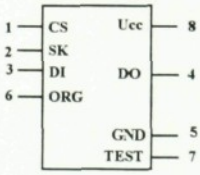
Mod 2 : Lze nastavit pouze u brány A , je to mod obousměrné sběrnice s korespondenčním režimem.

Obr. 5.3. Struktura obvodu 8255A



5.5. Popis obvodu EEPROM 9346A

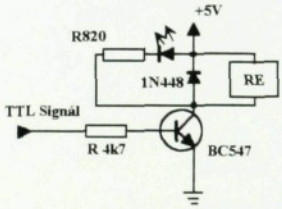
Obvod 9346A je EEPROM paměť o kapacitě 128 B. Skratka EEPROM znamená elektricky mazatelná a programovatelná pevná paměť. Někdy je též použito označení E^2PROM. Komunikace s pamětí probíhá pomocí sériového protokolu. Do paměti je vždy nejprve vyslán startbit, následuje příkaz a adresa byte. Do paměti lze zapisovat příkazem WRITE, číst z ní pomocí READ. Celou paměť je možno smazat najednou pomocí příkazu ERAL nebo celou naplnit příkazem WRAL. Před jakoukoli programovací operací je nutno vyslat do paměti příkaz EWEN. Paměť může být organizována buď 64x16 b nebo 128x8 b. Způsob organizace je určen vývodem ORG, jeho spojením s GND je zaručena organizace 128x8. Signál CS je výběrový . Pinem označeným SK vstupuje hodinový signál, pinem DI vstupují data a z pinu DO jsou data čtena. Při programování paměti je pomocí vývodu DO signalizováno zda je paměť připravena přijmout další data. Paměť je uložena v pouzdru DIL 8. Rozložení vývodů pouzdra je na obr. 5.4.



Obr. 5.4.

5.6. Popis výstupních bloků

Oba výstupní bloky jsou stejné. Každý blok je ovládán signálem úrovně TTL. Signál TTL je přiveden na bázi spínacího tranzistoru, který ovládá relé. Zde je použito relé RP 220. Spínací napětí tohoto relé je 6V. Relé je spínáno 5V , což však stačí k jeho bezpečnému sepnutí. Paralelně k relé je připojena signalizační LED dioda. Relé je indukční zátěž, proto je nutné přiřadit k němu paralelně diodu aby se zabránilo proražení tranzistoru.



Obr. 5.5. Schema výstupního bloku.

5.7. Popis obvodu 8253

Obvod 8253 obsahuje tři na sobě nezávislé programovatelné šestnáctibitové čítače s předvolbou. Čítače jsou synchronní,dekrementující, tj. čítající od předvolby směrem dolů buď binárně nebo dekadicky. Může pracovat v šesti pracovních modech. Každý z čítačů má vyveden hodinový vstup CLK, hradlovací signál GATE (funkci lze programově ovlivnit) a výstup OUT, který indikuje dosažení programového intervalu. Podobně jako obvod 8255 komunikuje 8253 s procesorem přes vyrovnávací registr, který je spojen s datovou sběrnicí procesoru. Jednotlivé mody obvodu se nastavují řídícím slovem (CW) a jsou to:

Generace přerušení: Výstup se změní na hodnotu OUT=1 po proběhnutí naprogramovaného intervalu. Vstup GATE povoluje čítání.

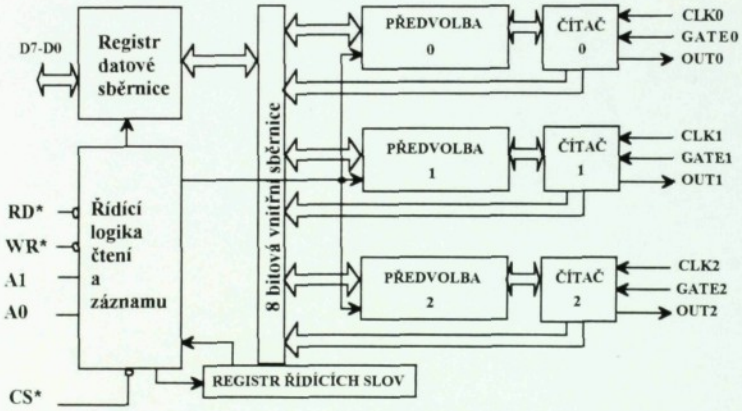
Znovuspustitelný monostabilní obvod: Čítání se spouští náběžnou hranou signálu GATE.

Programovatelný dělič frekvence: Provádí dělení vstupní frekvence číslem n, automaticky se obnoví předvolba v čítači, lze synchronizovat signálem GATE.

Programovatelný dělič frekvence se symetrickým výstupem: Tento režim je obdobný režimu 2.

Programové hradlování: Teprve po nastavení čítače začíná čítač čítat k nule. Hodinová frekvence může být hradlována vstupem GATE.

Obvodové hradlování: Čítač čeká na náběžnou hranu GATE, každá náběžná hrana způsobí počátek čítání.



Obr. 5.6. Vnitřní struktura obvodu 8253.

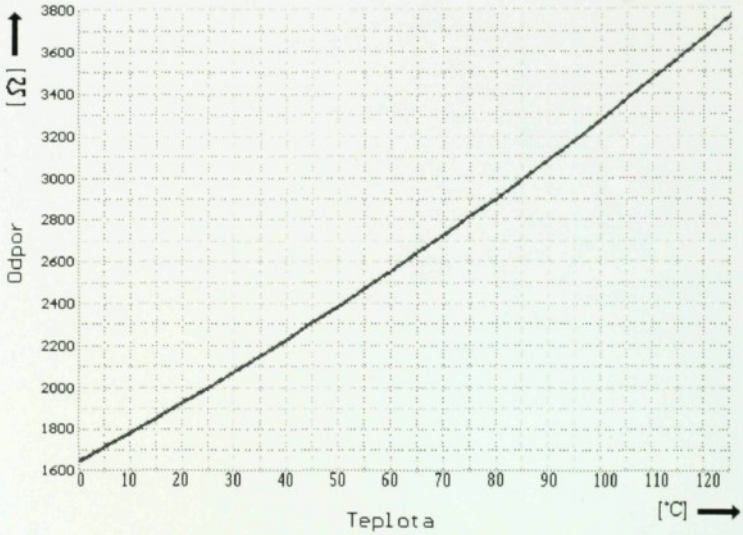
5.8. Popis snímače otáček

Jako snímač otáček je použit snímač MH1SS1. Jedná se vlastně o integrovaný obvod skládající se ze čtyř částí: Hallova generátoru, rozdílového zesilovače, Schmittova klopného obvodu a výkonového výstupního zesilovače. Hallův generátor vytváří stejnosměrné napětí, které je úměrné magnetickému poli. Toto napětí, zesílené v rozdílovém zesilovači se přivádí na klopný obvod, který zajišťuje jednoznačnost stavu na výstupu. Napěťové změny se zesílí ve výkonovém výstupním zesilovači. Obvod má jeden vývod zenmící, jeden napájecí a dva vývody výstupní. Výstupní signál je TTL kompatibilní.

5.9. Popis snímače teploty

Ke snímání teploty je použito odporové teplotní čidlo KTY10 - 6. Čidlo je umístěno v pouzdru TO92. Rozsah snímaných teplot je od -50 do +125 °C. Odpor čidla

při teplotě 25 °C je 2000 ohmů. Z následujícího grafu je patrno, že čidlo má téměř lineární charakteristiku.



Obr. 5.7. Charakteristika teplotního čidla KTY 10-6.

5.10. Popis převodníku odpor -> kmitočet

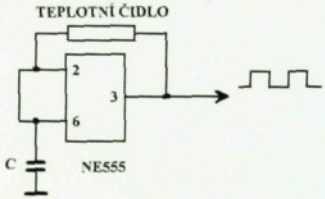
Pro převod odporu snímače na kmitočet je použito obvodu NE 555 zapojeného jako astabilní vibrátor. Rozsah povolených napájecích napětí obvodu je od 4.5V - 16V. Přesnost časování samotného IO je lepší než 1%. Obvod má záporný teplotní loeficient 0,0005 %/C. Maximální možný kmitočet oscilací je 1 MHz, v praxi se však doporučuje využít IO do kmitočtu 500 kHz.

Princip činnosti převodníku je následující: Na začátku je kondenzator vybit. Na vstupech komparatorů 2 a 6 je napěří menší než 1/3 Ucc, na výstupu 3 je Ucc. Kondenzátor se začne nabíjet přes teplotní čidlo s odporem R. Dosáhne-li napětí na kondenzátoru 2/3 Ucc překlopí se výstup 3 do 0 a kondenzátor se začne vybíjet přes R. Kondenzátor se vybije na 1/3 Ucc a výstup 3 se překlopí do Ucc. Toto se dále opakuje.

Na přesnost převodníku mají vliv přesnost obvodu NE555, přesnost teplotního čidla a přesnost kondenzátoru. Přesnost obvodu NE555 je kolem 1%. Teplotní čidlo se

vyrábí ve dvou provedeních spřesností 4% a 1%. Přesnost kondenzátorů se velice liší podle druhu a podle provedení. Nejméně vhodné je použití keramických nebo běžných elektrolytických kondenzátorů. Keramické kondenzátory velkých kapacit nemají dostatečně stabilní kapacitu a elektrolytické mají obvykle velké svodové proudy. Vhodné je použít kondenzátor slídový, polyesterový nebo tantalový. Polyesterové kondenzátory se vyrábějí s tolerancí 10 nebo 5%. Tantalové s přesností 20 %. Z uvedeného plyne vhodnost použití polyesterového kondenzátoru.

Délku periody lze vypočítat podle vzorce : $T=2*(R*C*\ln(0,5)$



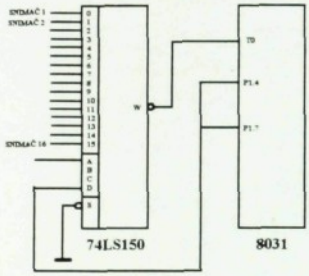
Obr. 5.8. Schema převodníku.

Tab. 5.9. Odpory čidla a teoretické délky půlperiod pro kondenzátor 4,7 uF

°C	Odpor [Ω]	Perioda [us]	°C	Odpor [Ω]	Perioda [us]
0	1 645,27	5 359,94	65	2 640,24	8 601,35
5	1 713,43	5 581,99	70	2 726,56	8 882,56
10	1 782,98	5 808,57	75	2 814,26	9 168,27
15	1 853,93	6 039,71	80	2 903,37	9 458,57
20	1 926,28	6 275,41	85	2 993,87	9 753,41
25	2 000,02	6 515,64	90	3 085,77	10 052,79
30	2 075,16	6 760,43	95	3 179,07	10 356,74
35	2 151,71	7 009,82	100	3 273,76	10 665,22
40	2 229,36	7 262,79	105	3 369,85	10 978,26
45	2 308,96	7 522,11	110	3 467,33	11 295,83
50	2 389,69	7 785,11	115	3 566,21	11 617,96
55	2 471,81	8 052,64	120	3 666,49	11 944,66
60	2 555,33	8 324,73	125	3 768,16	12 275,88

5.11. Popis multiplexu 74LS150

Obvod 74LS150 je šestnáctikanálový multiplexer pro funkci výběru dat. Na vstupy A,B,C,D se přivádí údaj adresy. Vybavovací vstup S umožňuje propojování více obvodů do jednoho funkčního celku. Aby mohlo dojít k přenosu dat musí být vstup vybavení S na nízké úrovni. Obvod má jeden inversní výstup

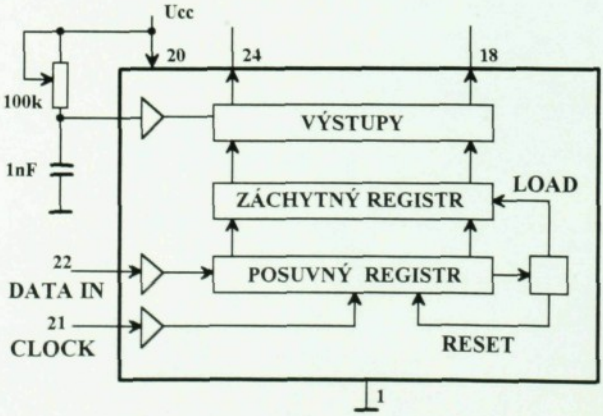


Obr. 5.10. Připojení multiplexoru k procesoru

5.12. Popis obvodu M5451

Ovladač LED displeje M5441 je zhotoven N-MOS technologií, je umístěn v pouzdru DIP 40. Obvod umožňuje ovládat 35 LED diod. Komunikace s obvodem

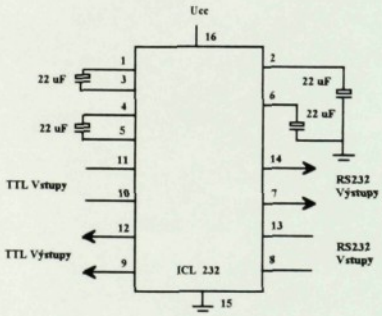
Obr. 5.11. Blokové schéma obvodu M5451



probíhá pomocí signálů DATA IN a CLOCK. Toto ho předurčuje pro spolupráci s procesorem. Počet vývodů obvodu umožňuje ovládání až 3,5 místného displeje. Displej musí být se společnou anodou. Proud procházející jednotlivými segmenty je možno ovládat velikostí proudu vstupujícího do pinu 19.

5.13. Popis obvodu ICL232

Obvod ICL 232 slouží při realizaci sériového rozhraní k oddělení signálů úrovně TTL a signálů úrovně RS232. Pomocí tohoto obvodu je možno zrealizovat dvě sériové linky (dva signály TxD a dva RxD). Tento, nebo jemu podobné obvody vyrábí mnoho výrobců pod různými označeními např. MAX232, TSC232 nebo AD232. Obvody ICL, MAX a TSC se zapojují naprosto shodně. Obvod AD232 má doporučené zapojení odlišné.



Obr. 5. 12. Doporučené zapojení obvodu ICL232

5.14. Popis SERVOCONTROLERU

Servocontrolér (dále SC) je samostatný řídící počítač na bázi procesoru 8031. Je schopen ovládat 8 servomotorů. Komunikace mezi SC a nadřazeným počítačem probíhá přes sériové rozhraní standartu RS 232C (malá varianta). Nadřazeným počítačem může být jakýkoli počítač schopný sériové komunikace standartu RS 232 C, nebo terminál schopný akceptovat tisknutelné ASCII znaky a znaky \CR a \LF. Ovládání polohy servomotorů probíhá pomocí příkazů vyslaných po sériové lince do SC. Její parametry jsou následující: rychlost - 4800 Bd, počet stop bitů 2, parita žádná, délka slova 8 bitů.

Tab. 5. 13. Přehled příkazů pro SC

Znak x nahrazuje pořadové číslo servomotoru pro který je příkaz určen

PŘÍKAZ	VÝZNAM
HELP	Příkaz vyvolá zobrazení přehledu nápovědy
Sx:ON	Příkaz zapíná ovládání SM
Sx:OFF	Vypnutí ovládání SM
Sx:[+-]pp%	Nastavení požadované polohy SM; + nebo - určuje směr, číslo pp určuje vzdálenost od středu v procentech
Sx:ddddD	Nastavení polohy SM ; číslo dddd udává (dekadicky) přímo délku řídícího pulsu v mikrosekundách.
Sx:hhhhH	Nastavení polohy SM; číslo hhhh udává (hexadecimálně) délku řídícího pulsu v mikrosekundách
H1	Vyvolání nápovědy: popis SC
H2	Vyvolání nápovědy: zapnutí/vypnutí SM
H3	Vyvolání nápovědy: aktuální nastavení
H4	Vyvolání nápovědy: nastavení polohy
H5	Vyvolání nápovědy: nastavení typu SM
H6	Vyvolání nápovědy: předdefinované typy
?STAV	Stejné jako H3
Sx:TYPt	Nastavení typu t servomotoru připojeného ke kanálu x. ($1 < t < 15$)
SV:ON	Zapnutí všech kanálů
SV:OFF	Vypnutí všech kanálů

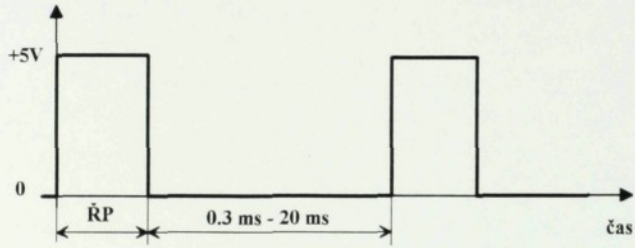
5.15. Popis ukazatele otáček

Ukazatel rychlosti otáčení klikového hřídele podává obsluze okamžitou informaci o počtu otáček stroje. Tato informace je velmi důležitá, neboť z ní obsluha usuzuje na kolik je využit výkon stroje při řezu. Byl zvolen ručkový typ ukazatele, protože nevyžaduje takové soustředění obsluhy při čtení zobrazované informace jako např. ukazatel digitální.

Ukazatel je tvořen prachotěsným pouzdrem s čelním sklem. Uvnitř je umístěn servomotor zajišťující pohyb ručky. Je zde použito proporcionálního modelářského servomotoru FUTABA FP S28 . Servomotor (SM) je ovládán servocontrolerem, který je umístěn rovněž uvnitř ukazatele. SM je připojen k servocontroleru pomocí tří vodičů,

dvou napajecích a jednoho řídícího. SM je napájen stejnosměrným napětím +5V. Řídící signál je napěťově kompatibilní s TTL logikou. Velikost výchylky servomotoru odpovídá délce kladného pulsu (ŘP). Katalogový list servomotoru je uveden v příloze 11.

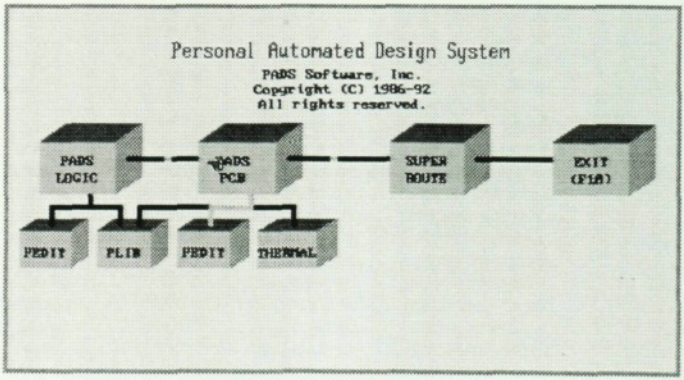
Časový diagram řídícího signálu servomotoru:



Obr. 5. 14.

5.16. Návrh schematu a plošného spoje

Pro návrh schematu a plošného spoje jsem použil návrhový system PADS firmy PADS Software Inc. . Na obr. 5.15. je vidět titulní obrazovka programu s nabídkou jednotlivých modulů. Ve volně šířené verzi programu jsou k dispozici pouze moduly pro návrh schematu a pro návrh plošného spoje. Vzhledem k tomu, že tato verze umí ovládat pouze základní paměť počítače jsou možnosti návrhu omezeny. Maximální počet spojů od pinu k pinu je 300, max. počet součástí 50.



Obr. 5.15.

Návrh schematu

Schema KS jsem navrhl pomocí programu LOGIC. Vzhledem k výše uvedeným omezením programu jsem musel návrh rozdělit na tři samostatná schemata. Na prvním schematu jsou vstupní konektory spolu s multiplexerem a výstupní bloky sestávající z tranzistorového zesilovače a z relé, dále je zde umístěn usměrňovač a stabilizovaný zdroj. Na druhém schematu je umístěn procesor spolu s ostatními programovatelnými obvody, obvody RAM, EPROM, EEPROM a obvod oddělující sériovou linku. Na třetím schematu je obvod displeje. Jsou zde oba LED displeje a obvody pro jejich ovládání. Displej má vlastní stabilizovaný zdroj.

Přestože je program LOGIC vybaven rozsáhlou knihovnou součástek byl jsem nucen dodefinovat součástky M5451, LED prvky a relé. Při definování nové součástky, která není zhotovena ve standartním pouzdru, je nutno nejprve nadefinovat nový typ pouzdra v programu PCB.

Návrh plošného spoje

Plošné spoje jsem navrhl pomocí programu PCB. Celé zapojení KS je rozděleno na dva plošné spoje. Na prvním z nich je procesor a ostatní číslicové obvody, konektory pro připojení vstupů a výstupní obvody. Tento plošný spoj je zhotoven jako jednostranný s drátovými propojkami. Druhý plošný spoj je také jednostranný. Jedna jeho strana tvoří čelní panel. Z této strany jsou vidět oba LED displeje a ostatní LED diody rozmístěné v obrazci, který simbolizuje rám a klikový hřídel stroje. Na druhé straně jsou umístěny oba obvody M5451 ovládající LED diody.

Zhotovení plošných spojů

Oba plošné spoje jsou zhotoveny fotocestou. Návrh pl. spoje je nutno vytisknout na kvalitní laserové tiskárně. Podle této předlohy se zhotoví negativní filmová kopie. Přes tuto negatvní předlohu se pomocí UV zářiče osvítí deska cuprextitu opatřená vrstvou fotocitlivého laku. V místech, kde je lak osvícen zůstane, zbytek laku je odstraněn. Tak dostaneme na desce masku pro vyleptání. Podrobnější popis zhotovování pl. spojů fotocestou by neměl smysl. Zhotovení je lépe přenechat specializované firmě.

5.17. Výběr vhodného vyššího programovacího jazyka

Pro mikropočítač 8031 je k dispozici mnoho různých překladačů. např: C, pascal, basic,forth a mnoho druhů assemblerů. V paxi se nejčastěji používá jazyka C. Pro aplikace náročné na rychlost výpočtu se používá assembleru nebo kombinace obou. Pro

jednodušší aplikace se používá BASICu. Forth se dnes již téměř nepoužívá. Jeho zvláštností je, že při zadávání matematických formulí používá polskou logiku.

Překladač jazyka C

Jazyk C původně vyvinul pro operační systém UNIX pan Dennis M. Ritche. K výhodám tohoto jazyka patří jeho procedurálnost, snadná spolupráce s jinými programovacími jazyky (assemblerem) a v neposlední řadě množství knihovních funkcí a datových typů. Nevýhodou je, že překladač generuje poměrně rozsáhlý strojový kód.

Překladač jazyka BASIC

Basic byl vyvinut J.Kemenym a T.Kurtzem v šedesátých letech v USA. K jejho výhodám patří jeho jednoduchost a fakt, že generuje poměrně krátký strojový kód. Nevýhody jazyka basic jsou jeho neprocedurálnost, nutnost číslování řádků a relativně chudší instrukční soubor.

Při výběru vhodného programovacího jazyka je nutno brát zřetel na následující hlediska :

Rychlost výsledného kódu

Velikost výsledného kódu

Přehlednost výsledného kódu (myšleno program v assembleru vytvořený překladačem)

K otestování obou překladačů bylo použito programu, který ve smyčce od 0 do 100 provádí operaci sčítání,odčítání,násobení a dělení čísel typu char. Výpis programu pro překladač C je v příloze 12. Program pro BASIC je v příloze 13. Porovnání výsledků obou překladačů je v tabulce 5.16.

Tab. 5.16. Porovnání překladačů C a BASIC

Programovací jazyk	Doba výpočtu	Velikost ASM kodu	Velikost HEX kodu
C	8 vteřin	6900 Byte	11157 Byte
BASIC	4 vteřiny	11482 Byte	5315 Byte

Z porovnání je zřejmé, že jazyk C generuje delší strojový kód než BASIC. Delšímu strojovému kódu odpovídá i delší doba výpočtu. Na řídící program KS je kladen

požadavek co nejkratší doby výpočtu. Z ekonomického hlediska je vhodnější kratší program, který umožní použít menší paměť programu a tudíž méně finančně náročnou. Bohužel překladač BASICu umožňuje použít pouze datových typů char a byte což je naprosto nedostačující. Proto jsem volil pro naprogramování testovacího programu jazyk C.

5.18. Testovací program pro KS

Testovací program slouží k ověření funkce obvodů kontrolního systému a pro ověření komunikace mezi KS a SC. Program je napsán v jazyce C, jeho výpis je v příloze 9. Procedůry z tohoto programu mohou být bez úpravy použity při sestavení konečného programu pro řízení KS. Jádro programu sloužící ke komunikaci s nadřazeným počítačem nebo terminálem bylo odladěno pomocí softwarového simulátoru EMILY 52. Ostatní části programu byly odladěny s použitím simulátoru paměti EEPROM přímo v KS.

Testování KS probíhá pomocí příkazů zadávaných po sériové lince nadřazeným počítačem. Parametry sériové linky jsou následující: rychlost - 4800 Bd, počet stop bitů 2, parita žádná, délka slova 8 bitů. Odpovědi testovacího programu na příkazy jsou předávány též po sériové lince zpět nadřazenému počítači. Přehled jednotlivých příkazů je v tabulce 5.17.

Běh programu začíná zpracováním procedury main (). Nejprve jsou inicializovány globální proměnné. Dále je inicializován čítač 1 procesoru sloužící k řízení přenosové rychlosti sériového kanálu. Nastavení a spuštění čítače je provedeno vloženými instrukcemi v assembleru. Potom následuje inicializace programovatelného obvodu 8255, nejprve je zresetován a následně je jeho CW registr naplněn řídícím slovem. Řídící slovo má hodnotu 82H, což znamená, že brány A a C jsou vstupní a brána B výstupní. Vývody C6 a C7 jsou nastaveny na úroveň H ostatní vývody bran A a C na úroveň L. Nastavením vývodů C6 a C7 je umožněno vysílání znaků po sériové lince. Nejprve je vyslána úvodní hlavička. Dále jsou dosazeny nulové hodnoty do pole proměnných udávajících stav diod na čelním panelu a vynulovány všechny tři čítače obvodu 8253. Jako poslední příkaz procedury main je volána procedura comand.

Procedura comand vypíše prompt "KS>" a čeká na zadání příkazu. Příkaz je zaznamenán v proměnné povel. Podle obsahu této proměnné jsou volány další procedury obstarávající vykonání požadovaného příkazu. Byl-li zadán neznámý příkaz je vypsána zpráva "Tento příkaz neznám!". Celá procedura je tvořena nekonečnou smyčkou, která je realizována příkazem while(1).

Tab. 5.17. Přehled příkazů testovacího programu pro KS

PŘÍKAZ	VÝZNAM
AKTUALNI SNIMAC?	Zobrazí číslo snímače, který je multiplexován na vstup procesoru 8031
NASTAV SNIMAC	Nastaví aktuální snímač
HELP	Zobrazí seznam příkazů
RELE A ON	Sepne relé A
RELE A OFF	Rozepne relé A
RELE B ON	Sepne relé B
RELE B OFF	Rozepne relé B
MER OTACKY	Začne měřit otáčky. Velikost otáček je zobrazována přes sériovou linku na terminálu
MER TEPLOTU	Začne měřit teplotu. Velikost teploty je zobrazována přes sériovou linku na terminálu
SBA	Nastavení jednoho bitu portu A na úroveň H
CBA	Nastavení jednoho bitu portu A na úroveň L
UBB	Zobrazí hodnotu na portu B
CTI BIT B	Zobrazí hodnotu jednoho bitu portu B
SBC	Nastavení jednoho bitu portu C na úroveň H
CBC	Nastavení jednoho bitu portu C na úroveň L
UKAZATEL	Nastavení polohy ukazatele na žádanou hodnotu (0 - 100)
UKAZATEL ON	Zapnutí ukazatele (servomotor "drží" polohu)
UKAZATEL OFF	Vypnutí ukazatele
DIODA ON	Nastaví příznak diody na hodnotu 1
DIODA OFF	Nastaví příznak doidy na hodnotu 0
ZOBRAZ	Provede zobrazení diod na panelu displeje podle jejich příznaku.
CTI CITAC	Přečte hodnotu zadaného čítače
NULUJ CITAC	Vynuluje zadaný čítač
EWEN	Zapíše do paměti EEPROM příka EWEN, který je nutno použít před použitím operací pro zápis do paměti.
ZAPIS DO EEPROM	Zapíše hodnotu do paměti EEPROM na zadanou adresu
CTI EEPROM	Přečte hodnotu z paměti EEPROM ze zadané adresy
TLACITKA	Zobrazí stav tlačítek na panelu displeje

Příkazy akceptované testovacím programem lze rozdělit do několika skupin:

Příkazy pro snímání vstupních hodnot

- AKTUALNI SNIMAC?

Zobrazí číslo snímače, který je "přepojen" multiplexerem na vstup P3.4

- NASTAV SNIMAC

Vypíše prompt "Zadej aktuální snímač:" a čeká na zadání dvou znaků tvořících číslo od 0 do 15. Zadání jiné hodnoty není ošetřeno.

- MER OTACKY

Periodicky vypisuje velikost otáček klikového hřídele. Na jednom řádku jsou postupně zobrazeny tři hodnoty. První hodnota udává počet cyklů kolikrát proběhla měřicí smyčka podprogramu otik() během periody signálu ze snímače otáček, druhá hodnota udává délku této periody v mikrosekundách a třetí hodnota vypočtenou velikost otáček za vteřinu. Je-li perioda signálu delší než 327680 mikrosekund je tato situace vyhodnocena jako porucha snímače.

- MER TEPLOTU

Periodicky vypisuje velikost teploty naměřené snímačem, který je nastaven jako aktuální. Na jednom řádku jsou postupně zobrazeny tři hodnoty. První hodnota udává počet cyklů kolikrát proběhla měřicí smyčka podprogramu mertik() během periody signálu ze snímače, druhá hodnota udává délku této periody v mikrosekundách a třetí hodnota vypočtenou velikost teploty ve stupních Celsia. Je-li perioda signálu delší než 81920 mikrosekund je tato situace vyhodnocena jako porucha snímače.

- UBB

Zobrazí hodnotu přečtenou z brány B obvodu 8255.

- CTI BIT B

Vypíše prompt "Zadej cislo bitu:" a čeká na zadání jednoho znaku reprezentujícího číslo bitu. Poté je zobrazena hodnota tohoto bitu.

- TLACITKA

Zobrazí stav tlačítek umístěných na čelním panelu

Příkazy pro ovládání výstupů

- **RELE A ON** Sepne relé A
- **RELE A OFF** Rozepne relé A
- **RELE B ON** Sepne relé B
- **RELE B OFF** Rozepne relé B

- **SBA** Nastaví bit brány A na hodnotu H
- **CBA** Nastaví bit brány A na hodnotu L
- **SBC** Nastaví bit brány C na hodnotu H
- **CBC** Nastaví bit brány C na hodnotu L

- DIODA ON

Vyzve k zadání čísla diody. Toto číslo je očekáváno ve tvaru dvou za sebou zadaných znaků. Zadání tohoto příkazu nemá za následek změnu stavu odpovídající diody, pouze nastaví v poli proměnných dio[] odpovídající proměnnou na hodnotu 1.

- DIODA OFF

Podobně jako u příkazu DIODA ON. Příznak je nastaven na 0.

- ZOBRAZ

Provede aktualizaci zobrazení signalizačních diod na čelním panelu podle proměnných dio[].

Příkazy pro práci s EEPROM

- EWEN

Zapíše do paměti EEPROM příkaz EWEN. Tento příkaz je nutno zapsat před použitím první operace pro zápis do paměti.

- ZAPIS DO EEPROM

Vyzve k zadání adresy na kterou má být proveden zápis a k zadání hodnoty, která má být zapsána. Adresa i hodnota je očekávána jako posloupnost tří znaků.

- CTI EEPROM

Vyzve k zadání adresy ze které má bý hodnota přečtena. Adresa je očekávána jako posloupnost tří znaků.

Příkazy pro ovládání čítačů otáček

- NULUJ CITAC

Vypíše dotaz "Který citac:?" a čeká na zadání čísla čítače ve formě jednoho znaku. Do zadaného čítače je nastavena hodnota 0. Je zvláštností obvodu 8253, že předvolba čítače je přepsána až při příchodu čítacího impulsu. Toto je nutno míti na paměti a patřičně poupravit přednastavovanou hodnotu.

- CTI CITAC

Vypíše dotaz "Který citac:?" a čeká na zadání čísla čítače ve formě jednoho znaku. Poté je zobrazena hodnota zadaného čítače. Čítače jsou dekrementující. Největší možná hodnota, kterou je možno nastavit je nula. Hodnota čítače bude proto záporná.

Příkazy pro ovládání ukazatele otáček

- UKAZATEL ON

Vyšle do servocontroleru příkaz "S1:ON".

- UKAZATEL OFF

Vyšle do servocontroleru příkaz "S1:OFF".

- UKAZATEL

Vyzve k zadání hodnoty v rozsahu 0 - 100 . Podle této hodnoty nastaví ukazatel otáček.

6. Závěr a zhodnocení realizované varianty

Cílem této práce bylo navrhnout a zrealizovat kontrolní systém pro rámovou pilu. Použití automatizační techniky v této oblasti výrobního procesu je zatím spíše vyjímkou, přesto však může znamenat značný přínos.

Díky zkvalitnění informovanosti obsluhy je možno zabránit zbytečným ztrátám energie. Použitím kontrolního systému se snižuje riziko poškození stroje při nepozornosti obsluhy. Nemalým přínosem je i zvýšení bezpečnosti práce. Obsluha již není nucena kontrolovat rukou teplotu jednotlivých ložisek stroje a nebezpečně se tak přibližovat rychle se pohybujícím součástem stroje.

Navržený kontrolní systém je schopen měřit teplotu na šestnácti místech, měřit otáčky klikového hřídele stroje, hnacího a posuvového elektromotoru. Dále je schopen ovládat posuv a signalizační houkačku. Velikost měřené teploty je signalizována na čelním panelu. Rovněž tak měřené otáčky. Pro snadnější orientaci obsluhy jsou otáčky signalizovány také pomocí ručkového ukazatele umístěného na dobře viditelném místě.

Ukazatel otáček je zrealizován jako samostatné zařízení komunikující s nadřazeným počítačem po sériovém rozhraní. Toto umožňuje jeho použití i v jiných aplikacích.

Kontrolní systém je navržen tak aby splňoval požadované cíle. S jeho dalším rozšiřováním se v budoucnu nepočítá. Přesto je možno aby byl pomocí sériového rozhraní začleněn do rozsáhlejšího řídícího systému.

7. Seznam odborné literatury

[1] R.Ille a kol. Rámová pila a technika řezání

[2] PADS Starter Kit PADS Software,Inc.,USA Překlad ing.Milan Klaus Copyright CADware s.r.o.1994

[3] GM ELECTRONIC Součástky pro elektroniku Květen 1994

[4] Katalog THOMSON

[5] Amatérské RADIO 4-94

[6] Amatérské RADIO 2-92

[7] Vladimír Šubrt Jednočipové mikropočítače INTEL 8048-8096

[8] Firemní literatura AUTOCONT

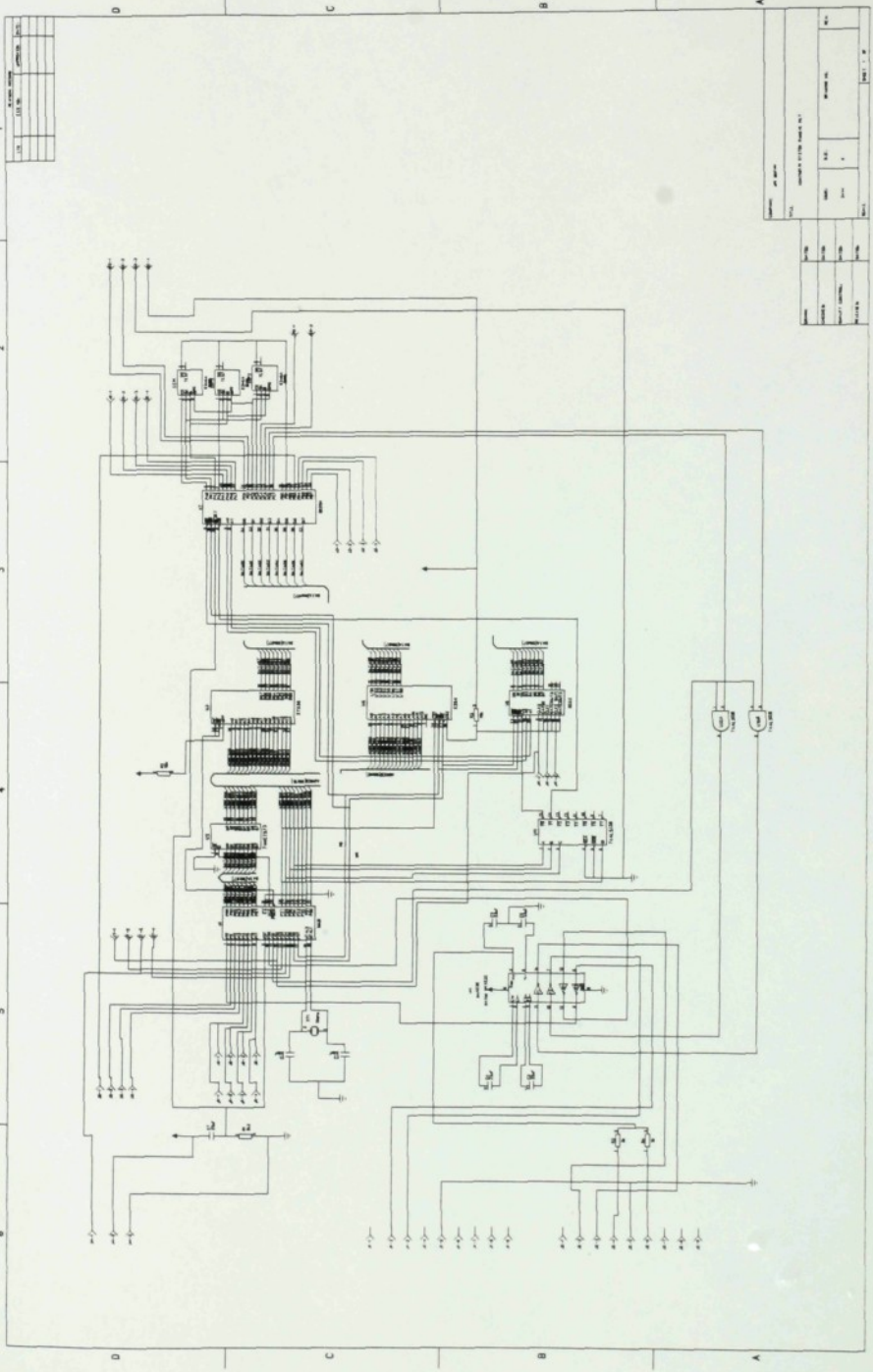
[9] Firemní literatura ERA COMPONENTS spol.s.r.o.

[10] Firemní literatura FUTABA CORPORATION OF AMERICA

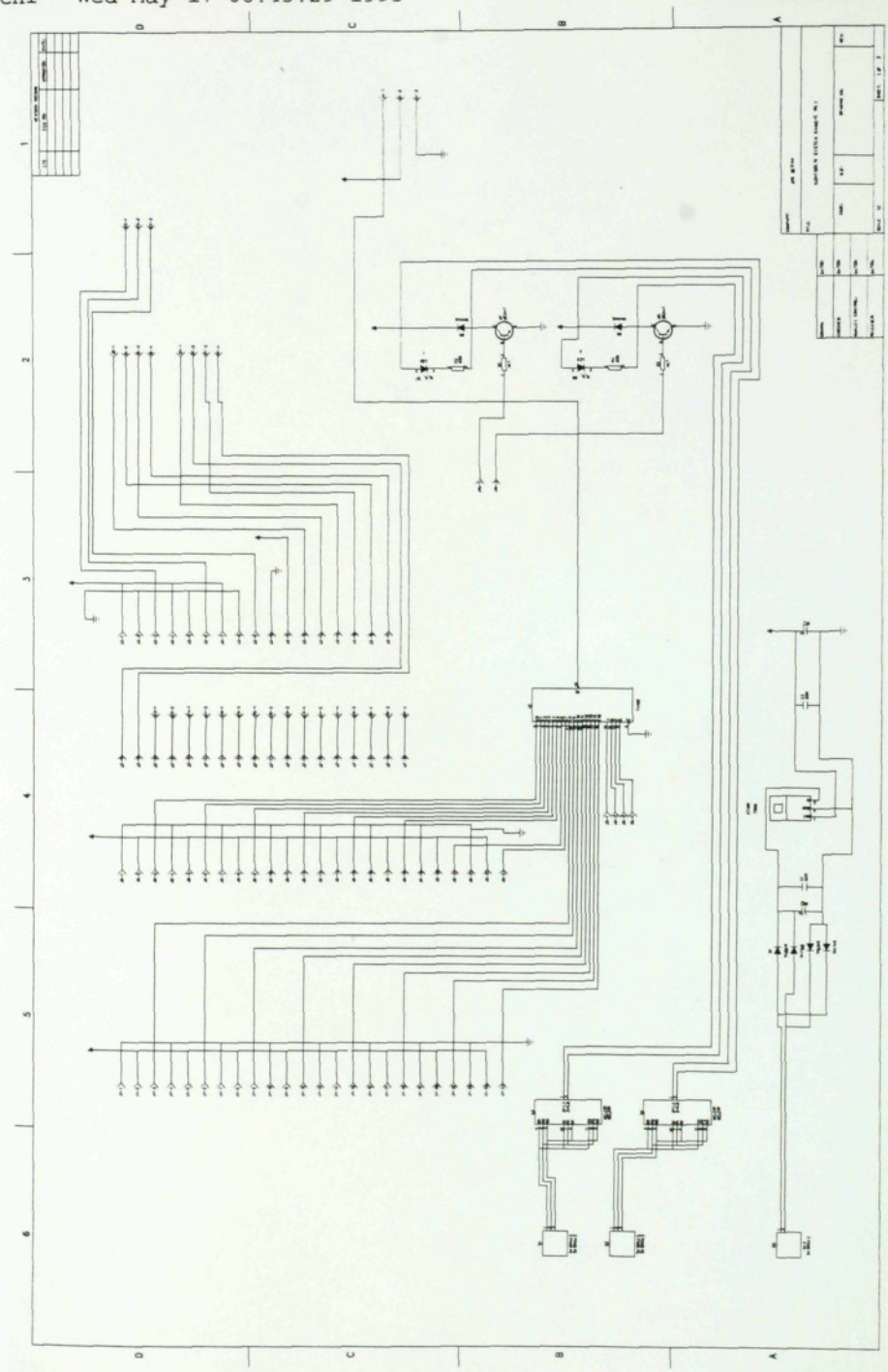
8. Seznam příloh

PŘÍLOHA 1	Schema CPU A
PŘÍLOHA 2	Schema CPU B
PŘÍLOHA 3	Schema displeje
PŘÍLOHA 4	Schema servocontroleru
PŘÍLOHA 5	Obraz plošného spoje CPU
PŘÍLOHA 6	Obraz plošného spoje displeje
PŘÍLOHA 7	Obraz čelního panelu
PŘÍLOHA 8	Obraz plošného spoje servocontroleru
PŘÍLOHA 9	Výpis testovacího programu
PŘÍLOHA 10	Výpis ovládacího programu servocontroleru
PŘÍLOHA 11	Katalogový list servomotoru FUTABA
PŘÍLOHA 12	Výpis programu c_test
PŘÍLOHA 13	Výpis programu b_test
PŘÍLOHA 14	Schéma zapojení konektorů KS

PŘÍLOHA 1

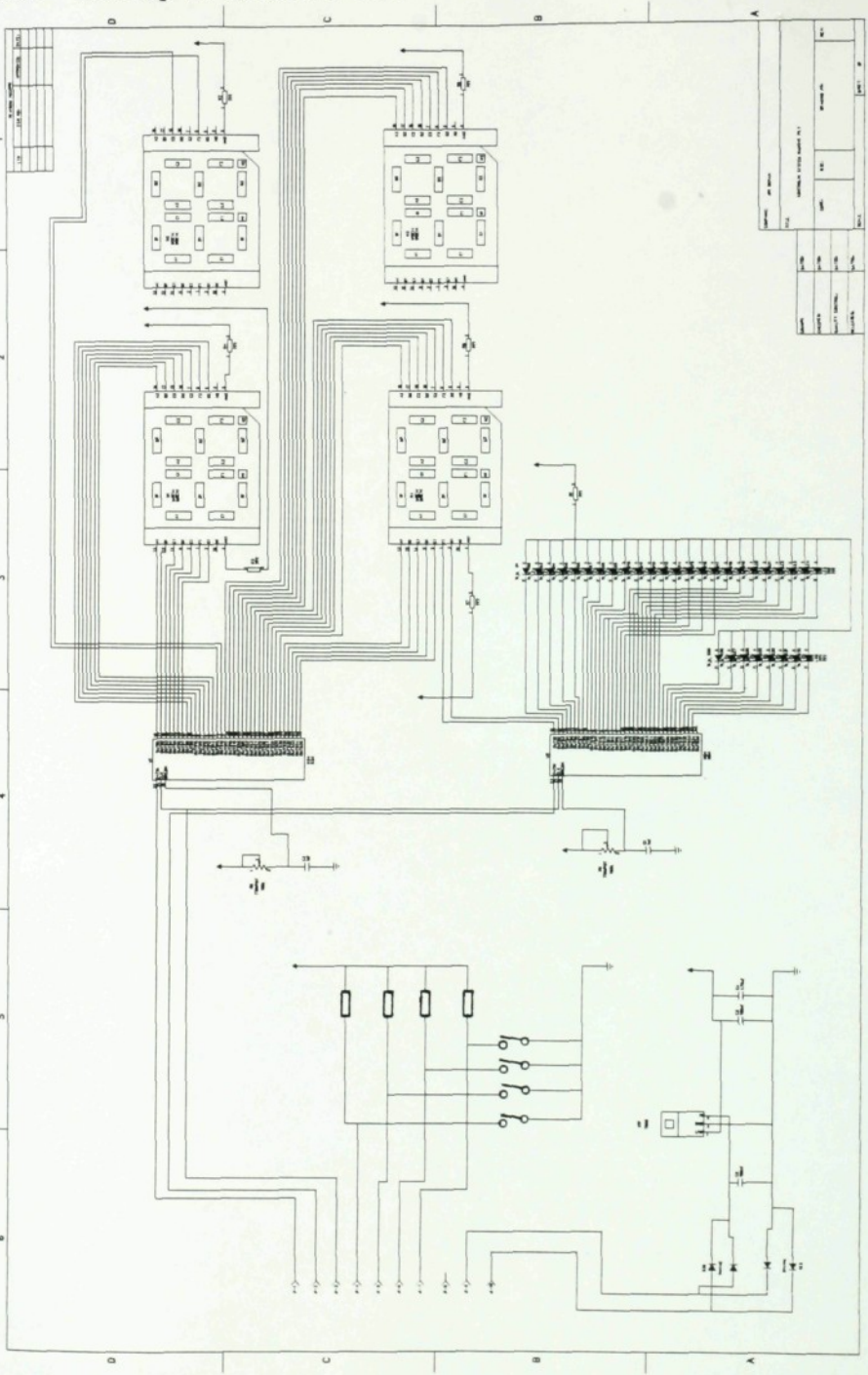


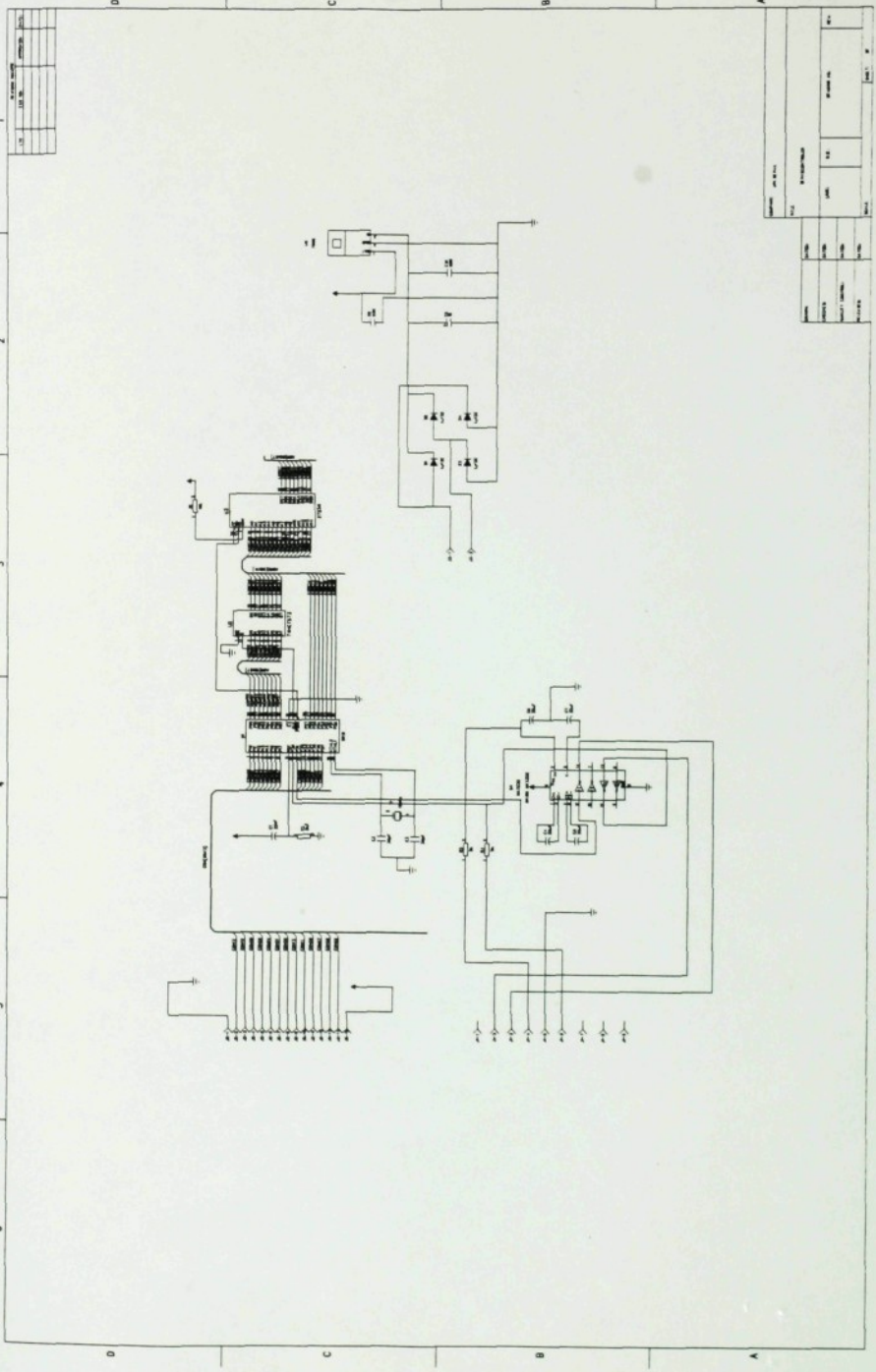
PŘÍLOHA 2



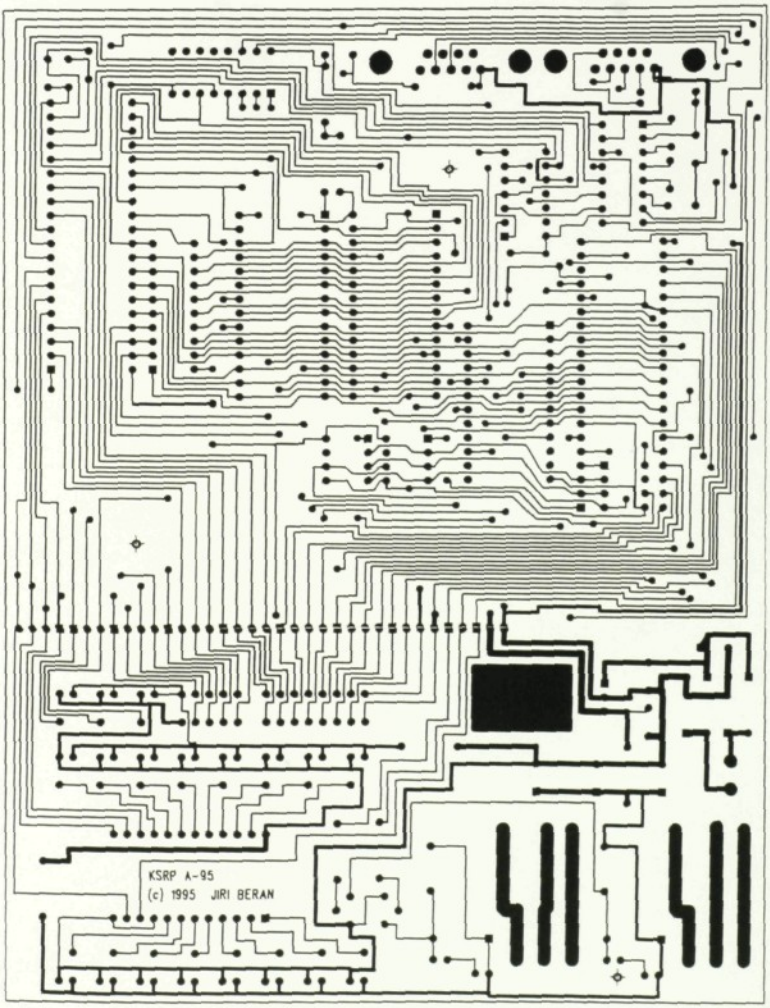
EJ.SCH1 - Wed May 17 08:45:23 1995

PŘÍLOHA 3

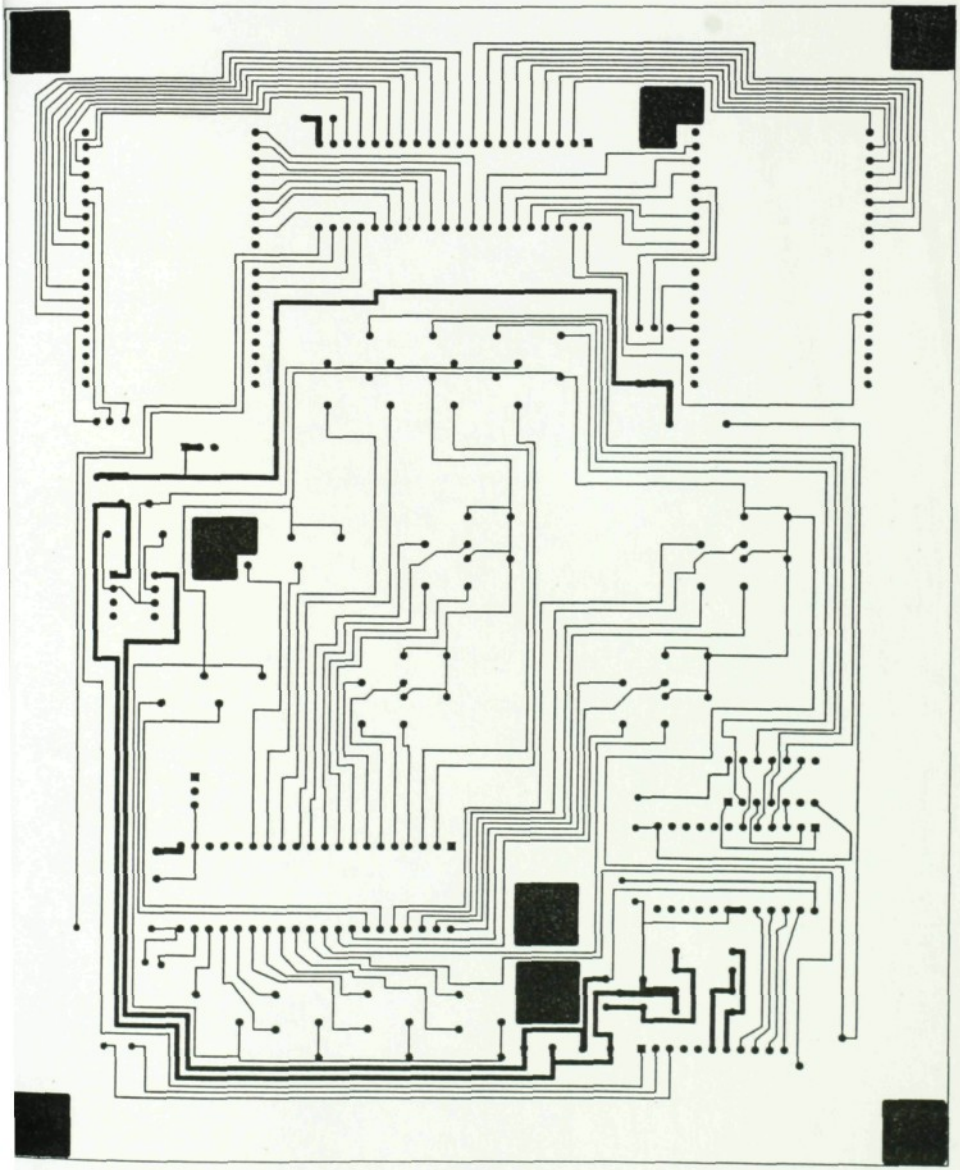


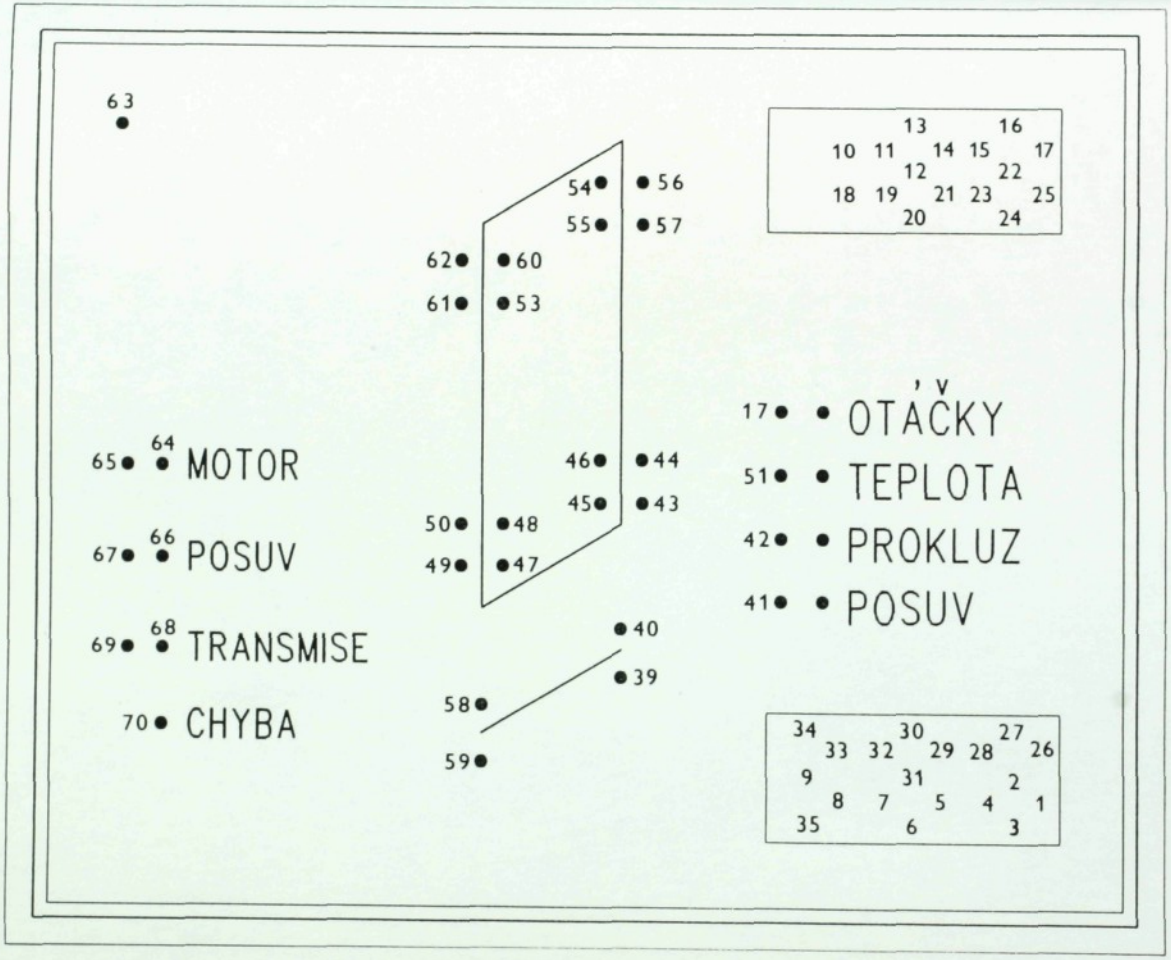


PŘÍLOHA 5



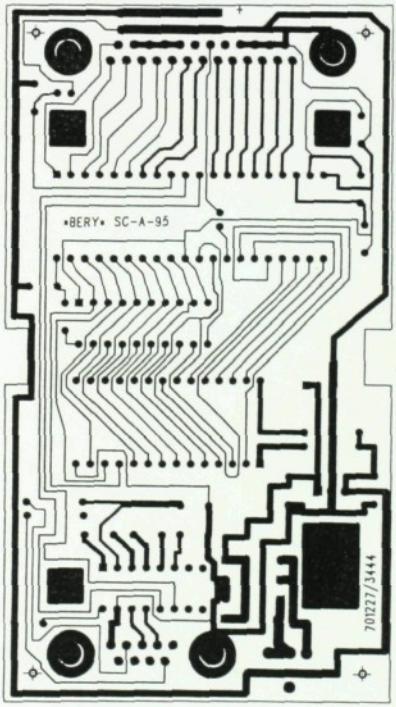
PŘÍLOHA 6




63
13 16
10 11 14 15 17
12 22
18 19 21 23 25
20 24
54 56
55 57
62 60
61 53
17 OTÁČKY
65 64 MOTOR
46 44
51 TEPLOTA
45 43
50 48
42 PROKLUZ
67 66 POSUV
49 47
41 POSUV
40
69 68 TRANSMISE
39
58
70 CHYBA
59
34 30 27
33 32 29 28 26
9 31 2
8 7 5 4 1
35 6 3

PŘÍLOHA 7

PŘÍLOHA 8



```
/*-----------------------------------------------------------------------*/
/*      TESTOVACI PROGRAM PRO KONTROLNI SYSTEM RAMOVE PILY               */
/*     ---------------------------------------------------               */
/*      SYSTEM OBSAHUJE:      8031       CPU                              */
/*                            27256      32 kB EPROM                      */
/*                            6264       8  kB RAM (NESDILENA)            */
/*                            8255       PIO    A,B,C,CW 9000H-9003H      */
/*                            8253       PIT    C1,C2,C3,CW 8000H-8003H   */
/*                            9346       EEPROM      (128x8¦64x16)        */
/*-----------------------------------------------------------------------*/
/*      PRIKAZY TESTOVACIHO PROGRAMU                                     */
/*     ----------------------------                                      */
/*      AKTUALNI SNIMAC       -  VYPISE AKTUALNI SNIMAC                  */
/*      NASTAV SNIMAC         -  NASTAVI AKTUALNI SNIMAC                 */
/*      HELP                  -  ZOBRAZI SOUPIS PRIKAZU                  */
/*      RELE A ON             -  SEPNUTI RELE A                          */
/*      RELE A OFF            -  ROZEPNUTI RELE A                        */
/*      RELE B ON             -  SEPNUTI RELE B                          */
/*      RELE B OFF            -  ROZEPNUTI RELE B                        */
/*      MER OTACKY            -  MERI OTACKY                             */
/*      MER TEPLOTU           -  MERI TEPLOTU                            */
/*      SBA                   -  NASTAV BIT PORTU A                      */
/*      CBA                   -  NULUJ BIT PORTU A                       */
/*      UBB                   -  PRECTE PORT B                           */
/*      CTI BIT B             -  PRECTE BIT PORTU B                      */
/*      SBC                   -  NASTAV BIT PORTU C                      */
/*      CBC                   -  NULUJ BIT PORTU C                       */
/*      UKAZATEL              -  NASTAVI POLOHU UKAZATELE                */
/*      UKAZATEL ON           -  ZAPNE UKAZATEL                          */
/*      UKAZATEL OFF          -  VYPNE UKAZATEL                          */
/*      DIODA ON              -  ZAPNE DIODU                             */
/*      DIODA OFF             -  VYPNE DIODU                             */
/*      NULUJ CITAC           -  NULUJE CITAC                            */
/*      CTI CITAC             -  CTE CITAC                               */
/*      EWEN                  -  ZAPISE DO EEPROM COMAND EWEN            */
/*      ZAPIS DO EEPROM       -  ZAPIS DO EEPROM                         */
/*      CTI EEPROM            -  CTE EEPROM                              */
/*      TLACITKA              -  VYPISE STAV TLACITEK                    */
/*      ZOBRAZ                -  ZOBRAZI DIODY                           */
/*-----------------------------------------------------------------------*/

extern _rdata char P1,P3;
extern _rbit char INT0;

     int      i;
     int      j;
     int      ukpoz;
     int      scdel;          /* MEZERA NA ZNAK                  */
     int      sccdel;         /* MEZERA NA COMAND                */
     int      edel;           /* PAUZA PRO  EEPROM               */
     int      pom01;
     int      pom02;
     int      pom03;
     int      eflag;
     int      minservo;
     int      mintepl;
     int      ukaf;           /* POHYB SC POUZE PO 1             */
     int      otcount;
```

```
int       tepcount;
int       tetat;
int       dio[72];
int       aksni;              /* AKTUALNI SNIMAC                    */
int       tabtep[25];         /* TABULKA PERIOD TEPLOT              */
int       vtepsi;
int       vtepli;
int       vtep2i;
int       mtepsi;
int       mtepli;
int       mtep2i;
long       ottat;
float       otacky;
float       otatf;
float       prtep;            /* PRUMERNA TEPLOTA                   */
float       prper;
float       vtepl;
float       vtep2;
float       vtepsk;
float       mtepl;
float       mtep2;
float       mtepsk;

char       zn;               /* ZNAK                               */
char       zn1;              /* ZNAK1                              */
char       eadr;
char       ehod;
char       pom;
char       outa;              /* VYSTUP NA PORTU A                 */
char       outc;              /* VYSTUP NA PORTU C                 */
char       povel[]="-------------------------------------------";

/*-----------------------------------------------------------------*/

                /*-------------------------- MERENI OTACEK ------*/
long      otik(void)
     {
     #ASM
          SETB INTO
          MOV  DPTR,#0000H
MT001:    INC  DPTR
          MOV  A,DPH
          JB   ACC.7,MTOERR
          JNB  INTO,MT001
          MOV  DPTR,#0000H        ; 2
MT002:    INC  DPTR               ; 2
          MOV  A,DPH              ; 1
          JB   ACC.7,MTOERR       ; 2
          JB   INTO,MT002         ; 2       SESTUPNA HRANA +- 5 us
          MOV  DPTR,#0000H        ; 2
MT003:    INC  DPTR               ; 2       CYKL TRVA 10 us #########
          MOV  A,DPH              ; 1                               #
          JB   ACC.7,MTOERR       ; 2                               #
          NOP                     ; 1                               #
          NOP                     ; 1                               #
          NOP                     ; 1                               #
          JNB  INTO,MT003         ; 2                         #########
          MOV  R1,DPH             ; MSB       DELKA PERIODY V DPTR
          MOV  RO,DPL             ; LSB        DPTR * 10 DELKA V us
          LJMP MTOEND
MTOERR:   MOV  R1,#OFFH           ; MSB
```

```
            MOV   R0,#0FFH          ; LSB
            MOV   R2,#00H
            MOV   R3,#00H
MTOEND:     NOP                     ; KOD PRO CHYBU SNIMACE
      #ENDASM
      }

void      merot(void)
      {
      printf("\n\rMereni otacek.\n\r");
      ottat=0;
      while( !chrrdy())
      {
      ottat=otik();
      if( ottat == 65535 )       {
                        puts("\n\rPorucha snimace.!!!!\n\r");
                        }
                  else      {
                        otatf=ottat*10;
                        otacky=1/((4*otatf)/1000000);
                        printf("\n\r%ld",ottat);
                        printf(" v us :%ld",ottat*10);
                        printf(" => %f ot./s",otacky);
                        };
      };
      }

int       mertik(void)
      {
      #ASM
            MOV   DPTR,#0000H
MT01:       INC   DPTR
            MOV   A,DPH
            JB    ACC.5,MTERR
            JNB   P3.4,MT01
            MOV   DPTR,#0000H       ; 2
MT02:       INC   DPTR              ; 2
            MOV   A,DPH             ; 1
            JB    ACC.5,MTERR       ; 2
            JB    P3.4,MT02         ; 2       SESTUPNA HRANA +- 5 us
            MOV   DPTR,#0000H       ; 2
MT03:       INC   DPTR              ; 2       CYKL TRVA 7 us ##########
            MOV   A,DPH             ; 1                              #
            JB    ACC.5,MTERR       ; 2                              #
            JNB   P3.4,MT03         ; 2                         ##########
            MOV   R1,DPH            ; MSB        PULKA PERIODY V DPTR
            MOV   R0,DPL            ; LSB         DPTR * 7 DELKA V us
            LJMP  MTEND
MTERR:      MOV   R1,#4EH           ; MSB
            MOV   R0,#20H           ; LSB
MTEND:      NOP                     ; 20000 KOD PRO CHYBU SNIMACE
      #ENDASM
      }

void      mertep(void)       /*------------- MERENI TEPLOTY -------*/
      {
      printf("\n\rMereni teploty.\n\r");
      tetat=0;
      while( !chrrdy())
      {
      tetat=mertik();
```

```
    if( tetat == 20000 )        {
                        printf("\n\rPorucha snimace.!!!!\n\r");
                        }
                else        {
                        printf("\n\r%d",tetat);
                        printf(" v us :%d",tetat*7);
                vtepsk=tetat*7;
                vtepsi=vtepsk;
                /* VYBERE NEJBLIZSI MENSI       */
                for(i=1;i<21;i++)
                        {
                        if(tabtep[i] <= vtepsi)
                                {
                                vtep1i=tabtep[i];
                                j=i+1;
                                mtep1i=i*5;
                                };
                        };
                /* VYBERE NEJBLIZSI VYSSI       */
                vtep2i=tabtep[j];
                mtep2i=j*5;
                /* INTERPOLUJE                  */
                vtep1=vtep1i;
                vtep2=vtep2i;
                mtep1=mtep1i;
                mtep2=mtep2i;
  mtepsk=((vtepsk-vtep1)*((mtep2-mtep1)/(vtep2-vtep1)))+mtep1;
                printf(" =>  %f st.",mtepsk);
                        };
            };
    }

void      setbita(a)
    char a;
    {
    pom=1;
    for(;a!=0;a--)pom=pom*2;
    a=pom;
    outa=(outa | a);
    a=outa;
    #ASM
          MOV         DPTR,#2
          LCALL       @czgetl
          extern      @czgetl
          MOV         A,R0
          MOV         DPTR,#9000H
          MOVX        @DPTR,A
    #ENDASM
    }

void      setbitc(a)
    char a;
    {
    pom=1;
    for(;a!=0;a--)pom=pom*2;
    a=pom;
    outc=(outc | a);
    a=outc;
    #ASM
          MOV         DPTR,#2
          LCALL       @czgetl
```

```
        extern      @czgetl
        MOV         A,R0
        MOV         DPTR,#9002H
        MOVX        @DPTR,A
#ENDASM
}

void    clrbita(a)
char a;
{
pom=1;
for(;a!=0;a--)pom=pom*2;
a=pom;
outa=(outa | a);
outa=(outa ^ a);
a=outa;
#ASM
        MOV         DPTR,#2
        LCALL       @czgetl
        MOV         A,R0
        MOV         DPTR,#9000H
        MOVX        @DPTR,A
#ENDASM
}

void    clrbitc(a)
char a;
{
pom=1;
for(;a!=0;a--)pom=pom*2;
a=pom;
outc=(outc | a);
outc=(outc ^ a);
a=outc;
#ASM
        MOV         DPTR,#2
        LCALL       @czgetl
        MOV         A,R0
        MOV         DPTR,#9002H
        MOVX        @DPTR,A
#ENDASM
}

void    relaon(void)
{
printf("\n\rRele A sepnuto.\n\r");
setbitc(4);
}

void    relbon(void)
{
printf("\n\rRele B sepnuto.\n\r");
setbitc(5);
}

void    relaoff(void)
{
printf("\n\rRele A rozepnuto.\n\r");
clrbitc(4);
}
```

```
void        relboff(void)
      {
      printf("\n\rRele B rozepnuto.\n\r");
      clrbitc(5);
      }

void        vybbit(void)
      {
      printf("\n\rZadej cislo bitu:");
      zn=getchar();
      zn=zn-0x30;
      }

void        sba(void)
      {
      vybbit();
      setbita(zn);
      }

void        sbc(void)
      {
      vybbit();
      setbitc(zn);
      }

void        cba(void)
      {
      vybbit();
      clrbita(zn);
      }

void        cbc(void)
      {
      vybbit();
      clrbitc(zn);
      }

char        ctib(void)
      {
      #ASM
            MOV  DPTR,#9001H
            MOVX A,@DPTR
            MOV  R0,A
      #ENDASM
      }

void        ubb(void)
      {
      printf("\n\rPrectena hodnota : %d",ctib());
      }

char        ctibbf()
      {
      zn1=ctib();
      pom=1;
      for(;zn!=0;zn--)pom=pom*2;
      pom=( zn1 & pom);
      if(pom > 0)pom=1;
      return(pom);
      }
```

```
void        ctibb()
      {
      vybbit();
      printf("\n\rHodnota bitu je : %d",ctibbf());
      }

void        delay(a)
      int a;
      {
      for(;a>0;a--)
      {
#ASM
      NOP
#ENDASM
      };
      }

void        putchd(a)
      char a;
      {
      putchar(a);
      delay(scdel);
      }

void        ukaon(void)
      {
      setbitc(7);
      putchd('S');
      putchd('1');
      putchd(':');
      putchd('O');
      putchd('N');
      putchd('\r');
      delay(100);
      clrbitc(7);
      }

void        ukaoff(void)
      {
      setbitc(7);
      putchd('S');
      putchd('1');
      putchd(':');
      putchd('O');
      putchd('F');
      putchd('F');
      putchd('\r');
      delay(100);
      clrbitc(7);
      }

void        ukazatel(a)
      int         a;
      {
      if(a>100)a=100;
      if(a<0)a=0;
      setbitc(7);
      a=(a*1.5)-68;
      if(ukaf==0)minservo=a;
      if(ukaf==1)
            {
```

```
        if(a > minservo)minservo=minservo+1;
        if(a < minservo)minservo=minservo-1;
        a=minservo;
        };
    putchd('S');
    putchd('1');
    putchd(':');
    if(a>0)putchd('+');
    if(a<0)putchd('-');
    itoa(abs(a),povel); /*      jednomistne cislo      */
    if(abs(a)<10)
        {
        pom=povel[0];
        putchd(pom);
        };              /*      dvoumistne cislo       */
    if(abs(a)<100 & abs(a)>9)
        {
        pom=povel[0];
        putchd(pom);
        pom=povel[1];
        putchd(pom);
        };
    putchd('%');
    putchar('\r');
    delay(100);
    clrbitc(7);
    }

void    ukazat(void)
    {
    printf("\n\rZadej hodnotu (0-100) :");
    gets(povel);
    ukpoz=atoi(povel);
    ukaon();
    delay(sccdel);
    ukazatel(ukpoz);
    delay(sccdel*6);
    ukaoff();
    }

void    signtp(void)
    {
    ukaf=1;
    tetat=0;
    printf("\n\rSignalizuji teplotu.\n\r");
    delay(sccdel*3);
    ukaon();
    delay(sccdel*2);
    ukazatel(0);
    delay(sccdel*2);
    prtep=0;
    pom02=mertik();
    pom02=pom02-5;
    pom01=0;
    pom03=0;
    while(!chrrdy())
        {
        tetat=mertik();
        while(tetat==20000)
                {
                printf("\n\rPorucha snimace\n\r");
```

```
                        tetat=mertik();
                        };
            tetat=tetat-pom02;
            prtep=((20*prtep)+tetat)/21;
            tetat=prtep;
            if(pom01<202)pom01++;
            if(pom01>200)
                  {
                  if(mintepl != tetat)
                                    {
                                    ukazatel(tetat);
                                    mintepl=tetat;
                                    };
                  delay(sccdel);
                  };
            };
            ukaf=0;
            delay(sccdel*2);
            ukazatel(0);
      delay(sccdel*2);
      ukaoff();
      }

void      zobraz(void)
      {
                              /* P1.2  data 1 */
                              /* P1.6  data 2 */
                              /* P1.4  clk    */
      P1.4=0;
      delay(10);
      P1.2=0;
      P1.6=0;
      delay(10);
      for(i=0;i<38;i++)
            {
            P1.4=1;
            delay(10);
            P1.4=0;
            delay(10);
            };
      P1.2=1;
      P1.6=1;
      delay(10);
      P1.4=1;
      delay(10);
      P1.4=0;
      delay(10);
      for(i=1;i<36;i++)
            {
            if(dio[i]==1)P1.2=1; else P1.2=0;
            if(dio[(35+i)]==1)P1.6=1; else P1.6=0;
            delay(10);
            P1.4=1;
            delay(10);
            P1.4=0;
            delay(10);
            };
      }
void      dioon(void)
      {
      printf("\n\rZadej cislo diody. :");
```

```
      zn=getchar();
      zn=zn-48;
      zn1=getchar();
      zn1=zn1-48;
      dio[((zn*10)+zn1)]=1;
      zobraz();
      }

void      diooff(void)
      {
      printf("\n\rZadej cislo diody. :");
      zn=getchar();
      zn=zn-48;
      zn1=getchar();
      zn1=zn1-48;
      dio[((zn*10)+zn1)]=0;
      zobraz();
      }

void      nulci(a)
      char a;
      {
      if(a==0)
      {
#ASM
      MOV  DPTR,#8003H
      MOV  A,#30H
      MOVX @DPTR,A
      MOV  DPTR,#8000H
      MOV  A,#0H
      MOVX @DPTR,A
      MOV  DPTR,#8000H
      MOV  A,#0H
      MOVX @DPTR,A
#ENDASM
      goto endnuc;
      };
      if(a==1)
      {
#ASM
      MOV  DPTR,#8003H
      MOV  A,#70H
      MOVX @DPTR,A
      MOV  DPTR,#8001H
      MOV  A,#0H
      MOVX @DPTR,A
      MOV  DPTR,#8001H
      MOV  A,#0H
      MOVX @DPTR,A
#ENDASM
      goto endnuc;
      };
      if(a==2)
      {
#ASM
      MOV  DPTR,#8003H
      MOV  A,#080H
      MOVX @DPTR,A
      MOV  DPTR,#8002H
      MOV  A,#0H
      MOVX @DPTR,A
```

```
      MOV  DPTR,#8002H
      MOV  A,#0H
      MOVX @DPTR,A
#ENDASM
      };
endnuc:;
      }

int       ctic(a)
      char a;
      {
      if(a==0)
      {
#ASM
      MOV  DPTR,#8000H
      MOVX A,@DPTR
      MOV  R0,A
      MOV  DPTR,#8000H
      MOVX A,@DPTR
      MOV  R1,A
#ENDASM
      goto endctic;
      };
      if(a==1)
      {
#ASM
      MOV  DPTR,#8001H
      MOVX A,@DPTR
      MOV  R0,A
      MOV  DPTR,#8001H
      MOVX A,@DPTR
      MOV  R1,A
#ENDASM
      goto endctic;
      };
      if(a==2)
      {
#ASM
      MOV  DPTR,#8002H
      MOVX A,@DPTR
      MOV  R0,A
      MOV  DPTR,#8002H
      MOVX A,@DPTR
      MOV  R1,A
#ENDASM
      };
endctic:;
      }

void       cticit(void)
      {
      printf("\n\rKtery citac?:");
      zn=getchar();
      zn=zn-48;
      if(zn==0)
      {
#ASM
      MOV   DPTR,#8003H
      MOV   A,#0H
      MOVX  @DPTR,A
#ENDASM
```

```
        };
        if(zn==1)
        {
#ASM
        MOV    DPTR,#8003H
        MOV    A,#40H
        MOVX   @DPTR,A
#ENDASM
        };
        if(zn==2)
        {
#ASM
        MOV    DPTR,#8003H
        MOV    A,#80H
        MOVX   @DPTR,A
#ENDASM
        };
        printf("\n\rCitac ma hodnotu: %d",ctic(zn));
        }

void        nulcit(void)
        {
        printf("\n\rKtery citac?:");
        zn=getchar();
        zn=zn-48;
        nulci(zn);
        }

void        nsak(a)
        char a;
        {
        if(a==0){P1.4=0;P1.5=0;P1.6=0;P1.7=0;};
        if(a==1){P1.4=1;P1.5=0;P1.6=0;P1.7=0;};
        if(a==2){P1.4=0;P1.5=1;P1.6=0;P1.7=0;};
        if(a==3){P1.4=1;P1.5=1;P1.6=0;P1.7=0;};
        if(a==4){P1.4=0;P1.5=0;P1.6=1;P1.7=0;};
        if(a==5){P1.4=1;P1.5=0;P1.6=1;P1.7=0;};
        if(a==6){P1.4=0;P1.5=1;P1.6=1;P1.7=0;};
        if(a==7){P1.4=1;P1.5=1;P1.6=1;P1.7=0;};
        if(a==8){P1.4=0;P1.5=0;P1.6=0;P1.7=1;};
        if(a==9){P1.4=1;P1.5=0;P1.6=0;P1.7=1;};
        if(a==10){P1.4=0;P1.5=1;P1.6=0;P1.7=1;};
        if(a==11){P1.4=1;P1.5=1;P1.6=0;P1.7=1;};
        if(a==12){P1.4=0;P1.5=0;P1.6=1;P1.7=1;};
        if(a==13){P1.4=1;P1.5=0;P1.6=1;P1.7=1;};
        if(a==14){P1.4=0;P1.5=1;P1.6=1;P1.7=1;};
        if(a==15){P1.4=1;P1.5=1;P1.6=1;P1.7=1;};
        }

void        nsaksni(void)
        {
        printf("\n\rZadej aktualni snimac:");
        zn=getchar();
        zn1=getchar();
        zn=zn-48;
        zn1=zn1-48;
        aksni=(zn*10)+zn1;
        }

void        csh(void)
        {
```

```
        setbita(3);
        }

void        csl(void)
        {
        clrbita(3);
        }

void        skh(void)
        {
        setbita(1);
        }

void        skl(void)
        {
        clrbita(1);
        }

void        dh(void)
        {
        setbita(2);
        }

void        dl(void)
        {
        clrbita(2);
        }

void        eouth(void)
        {
        csh();
        delay(edel);
        dh();
        delay(edel);
        skh();
        delay(edel);
        skl();
        delay(edel);
        }

void        eoutl(void)
        {
        csh();
        delay(edel);
        dl();
        delay(edel);
        skh();
        delay(edel);
        skl();
        delay(edel);
        }

void        clk()
        {
        skh();
        delay(edel);
        skl();
        delay(edel);
        }

void        aksd(void)
```

```
    {
    printf("\n\rAktuani snimac je : %d",aksni);
    }

void      even(void)
    {
    eouth();
    eoutl();
    eoutl();
    eouth();
    eouth();
    eoutl();
    eoutl();
    eoutl();
    eoutl();
    eoutl();
    dl();
    delay(edel);
    csl();
    delay(edel);
    clk();
    clk();
    clk();
    }

void      ezap(void)
    {
    eouth();
    eoutl();
    eouth();              /* ADRESA                          */
    pom=eadr;
    pom=(pom & 0x40);
    if(pom > 0)eouth(); else eoutl();
    pom=eadr;
    pom=(pom & 0x20);
    if(pom > 0)eouth(); else eoutl();
    pom=eadr;
    pom=(pom & 0x10);
    if(pom > 0)eouth(); else eoutl();
    pom=eadr;
    pom=(pom & 0x08);
    if(pom > 0)eouth(); else eoutl();
    pom=eadr;
    pom=(pom & 0x04);
    if(pom > 0)eouth(); else eoutl();
    pom=eadr;
    pom=(pom & 0x02);
    if(pom > 0)eouth(); else eoutl();
    pom=eadr;
    pom=(pom & 0x01);
    if(pom > 0)eouth(); else eoutl();
    pom=ehod;
    pom=(pom & 0x80);
    if(pom > 0)eouth(); else eoutl();
    pom=ehod;
    pom=(pom & 0x40);
    if(pom > 0)eouth(); else eoutl();
    pom=ehod;
    pom=(pom & 0x20);
    if(pom > 0)eouth(); else eoutl();
    pom=ehod;
```

```
        pom=(pom & 0x10);
        if(pom > 0)eouth(); else eoutl();
        pom=ehod;
        pom=(pom & 0x08);
        if(pom > 0)eouth(); else eoutl();
        pom=ehod;
        pom=(pom & 0x04);
        if(pom > 0)eouth(); else eoutl();
        pom=ehod;
        pom=(pom & 0x02);
        if(pom > 0)eouth(); else eoutl();
        pom=ehod;
        pom=(pom & 0x01);
        if(pom > 0)eouth(); else eoutl();
        dl();
        delay(edel);
        csl();
        delay(edel);
        clk();
        clk();
        clk();
        clk();
        }

void        ecti(void)
        {
        zn=1;
        eouth();
        eouth();
        eoutl();
        pom=eadr;
        pom=(pom & 0x40);
        if(pom > 0)eouth(); else eoutl();
        pom=eadr;
        pom=(pom & 0x20);
        if(pom > 0)eouth(); else eoutl();
        pom=eadr;
        pom=(pom & 0x10);
        if(pom > 0)eouth(); else eoutl();
        pom=eadr;
        pom=(pom & 0x08);
        if(pom > 0)eouth(); else eoutl();
        pom=eadr;
        pom=(pom & 0x04);
        if(pom > 0)eouth(); else eoutl();
        pom=eadr;
        pom=(pom & 0x02);
        if(pom > 0)eouth(); else eoutl();
        pom=eadr;
        pom=(pom & 0x01);
        if(pom > 0)eouth(); else eoutl();
        dl();
        delay(edel);
        skh();
        delay(edel);
        ehod=0;
        zn=1;
        if(ctibbf()==1)ehod=ehod+128;
        skl();
        delay(edel);
        skh();
```

```
    delay(edel);
    zn=1;
    if(ctibbf()==1)ehod=ehod+64;
    skl();
    delay(edel);
    skh();
    delay(edel);
    zn=1;
    if(ctibbf()==1)ehod=ehod+32;
    skl();
    delay(edel);
    skh();
    delay(edel);
    zn=1;
    if(ctibbf()==1)ehod=ehod+16;
    skl();
    delay(edel);
    skh();
    delay(edel);
    zn=1;
    if(ctibbf()==1)ehod=ehod+8;
    skl();
    delay(edel);
    skh();
    delay(edel);
    zn=1;
    if(ctibbf()==1)ehod=ehod+4;
    skl();
    delay(edel);
    skh();
    delay(edel);
    zn=1;
    if(ctibbf()==1)ehod=ehod+2;
    skl();
    delay(edel);
    skh();
    delay(edel);
    zn=1;
    if(ctibbf()==1)ehod=ehod+1;
    skl();
    delay(edel);
    csl();
    clk();
    clk();
    clk();
    }

void      ezapas(void)
    {
    printf("\n\rZadej adresu :");
    zn1=0;
    zn=getchar();
    zn=zn-48;
    zn1=zn1+(zn*100);
    zn=getchar();
    zn=zn-48;
    zn1=zn1+(zn*10);
    zn=getchar();
    zn=zn-48;
    zn1=zn1+zn;
    eadr=zn1;
```

```
    if(eflag==1)
        {
        printf("\n\rZadej hodnotu :");
        zn1=0;
        zn=getchar();
        zn=zn-48;
        zn1=zn1+(zn*100);
        zn=getchar();
        zn=zn-48;
        zn1=zn1+(zn*10);
        zn=getchar();
        zn=zn-48;
        zn1=zn1+zn;
        ehod=zn1;
    };
    }

void    ezapd(void)
    {
    ezapas();
    ezap();
    }

void    ectid(void)
    {
    eflag=0;
    ezapas();
    eflag=1;
    ecti();
    printf("\n\rHodnota je : %d",ehod);
    }

void    tlac(void)
    {
    printf("\n\rStav tlacitek je:");
    printf("\n\r                    OTACKY  : ");
    if(P1.3 == 1)printf("1"); else printf("0");
    printf("\n\r                    TEPLOTA : ");
    if(P1.1 == 1)printf("1"); else printf("0");
    printf("\n\r                    PROKLUZ : ");
    if(P1.7 == 1)printf("1"); else printf("0");
    printf("\n\r                    POSUV   : ");
    if(P1.7 == 1)printf("1"); else printf("0");
    printf("\n\r  ");
    }

void    help(void)
    {
puts("*************************************************************");
puts("*                    PREHLED PRIKAZU                        *");
puts("*                   -----------------                       *");
puts("*       AKTUALNI SNIMAC          UKAZATEL ON                *");
puts("*       NASTAV SNIMAC            UKAZATEL OFF               *");
puts("*       HELP                     SBC                        *");
puts("*       RELE A ON                CBC                        *");
puts("*       RELE A OFF               DIODA ON                   *");
puts("*       RELE B ON                DIODA OFF                  *");
puts("*       RELE B OFF               NULUJ CITAC                *");
puts("*       MER OTACKY               CTI CITAC                  *");
puts("*       MER TEPLOTU              EWEN                       *");
puts("*       SBA                      UKAZATEL                   *");
```

```
puts("*        CBA                          ZOBRAZ                          *");
puts("*        UBB                          ZAPIS DO EEPROM                 *");
puts("*        CTI BIT B                    CTI EEPROM                      *");
puts("*        TLACITKA                                                     *");
puts("*************************************************************");
     }

void comand(int coms)
     {
     while( 1 )
     {
     printf("\n\rKS>");
     gets(povel);
     if(strcmp(povel,"RELE A ON")==0){relaon();goto comend;};
     if(strcmp(povel,"RELE A OFF")==0){relaoff();goto comend;};
     if(strcmp(povel,"RELE B ON")==0){relbon();goto comend;};
     if(strcmp(povel,"RELE B OFF")==0){relboff();goto comend;};
     if(strcmp(povel,"MER OTACKY")==0){merot();goto comend;};
     if(strcmp(povel,"MER TEPLOTU")==0){mertep();goto comend;};
     if(strcmp(povel,"SBA")==0){sba();goto comend;};
     if(strcmp(povel,"CBA")==0){cba();goto comend;};
     if(strcmp(povel,"UBB")==0){ubb();goto comend;};
     if(strcmp(povel,"CTI BIT B")==0){ctibb();goto comend;};
     if(strcmp(povel,"SBC")==0){sbc();goto comend;};
     if(strcmp(povel,"CBC")==0){cbc();goto comend;};
     if(strcmp(povel,"UKAZATEL")==0){ukazat();goto comend;};
     if(strcmp(povel,"UKAZATEL ON")==0){ukaon();goto comend;};
     if(strcmp(povel,"UKAZATEL OFF")==0){ukaoff();goto comend;};
     if(strcmp(povel,"ZOBRAZ")==0){zobraz();goto comend;};
     if(strcmp(povel,"DIODA ON")==0){dioon();goto comend;};
     if(strcmp(povel,"DIODA OFF")==0){diooff();goto comend;};
     if(strcmp(povel,"NULUJ CITAC")==0){nulcit();goto comend;};
     if(strcmp(povel,"CTI CITAC")==0){cticit();goto comend;};
     if(strcmp(povel,"NASTAV SNIMAC")==0){nsaksni();goto comend;};
     if(strcmp(povel,"AKTUALNI SNIMAC?")==0){aksd();goto comend;};
     if(strcmp(povel,"EWEN")==0){even();goto comend;};
     if(strcmp(povel,"ZAPIS DO EEPROM")==0){ezapd();goto comend;};
     if(strcmp(povel,"CTI EEPROM")==0){ectid();goto comend;};
     if(strcmp(povel,"TLACITKA")==0){tlac();goto comend;};
     if(strcmp(povel,"HELP")==0){help();goto comend;};
     printf("\n\rTento prikaz neznam!\n\r");
comend:      ;
     };
     }

/*************************************************** M A I N *******/
main()
{
/*-------------------------------------------------------------*/
/*      INICIALIZACE PROMENNYCH                                */
/*-------------------------------------------------------------*/
outa=0x00;
outc=0x00;
scdel=100;
sccdel=2000;
edel=100;
eflag=1;
aksni=0;
tabtep[0]=5359;              /*  O stupnu                    */
tabtep[1]=5581;
tabtep[2]=5808;
```

```
tabtep[3]=6039;
tabtep[4]=6275;
tabtep[5]=6515;
tabtep[6]=6760;
tabtep[7]=7009;
tabtep[8]=7262;
tabtep[9]=7522;
tabtep[10]=7785;
tabtep[11]=8052;
tabtep[12]=8324;
tabtep[13]=8601;
tabtep[14]=8882;
tabtep[15]=9168;
tabtep[16]=9458;
tabtep[17]=9753;
tabtep[18]=10052;
tabtep[19]=10356;
tabtep[20]=10665;
tabtep[21]=10978;       /* 105 st.                          */
/*************************** SERIOVY KANAL NA 4800 ************/
#ASM
     CLR       TR1
     MOV       TMOD,#20H
     MOV       TH1,#-13
     MOV       TL1,#-13
     MOV       PCON,#10000000B
     CLR       SM0
     SETB       SM1
     SETB       TR1
#ENDASM
/************************** INICIALIZACE 8255 ****************/
/*      ADRESY:       CWR       9003H
           A       9000H
           B       9001H
           C       9002H
     BRANY A,C VYSTUPNI
            B VSTUPNI                                        */
P3.5=0;                          /* RESET */
delay(10);
P3.5=1;
delay(30);
P3.5=0;                          /* CWR */
#ASM
     MOV       DPTR,#9003H
     MOV       A,#82H
     MOVX       @DPTR,A
     NOP
#ENDASM

/********************** INICIALIZACE BRANY A **************/
clrbita(0);
clrbita(1);
clrbita(2);
clrbita(3);
clrbita(4);
clrbita(5);
clrbita(6);
clrbita(7);
/********************** INICALIZACE BRANY C ****************/
clrbitc(0);
clrbitc(1);
```

```
clrbitc(2);
clrbitc(3);
clrbitc(4);
clrbitc(5);
setbitc(6);
setbitc(7);
printf("\n\r\n\r\n\r");
printf("*******************************************\n\r");
printf("*       KONTROLNI SYSTEM RAMOVE PILY      *\n\r");
printf("*            TESTOVACI PROGRAM            *\n\r");
printf("*       ============================      *\n\r");
printf("*           (c) 1995  Jiri Beran          *\n\r");
printf("*******************************************\n\r");
/*********************** INICIALIZACE DIOD ***************/
for(i=0;i<71;i++)dio[i]=0;
/*********************** NASTAVENI 8253 ******************/
P3.2=1;
nulci(0);
nulci(1);
nulci(2);
comand(1);
}
```

```
;*****************************************************************
;*          Řídící program SERVOCONTROLERU                       *
;*          ------------------------------------------------     *
;*                    (c) 1995  Jiří Beran                       *
;*****************************************************************

NEZNA       EQU   7BH
POMOC       EQU   32H
POMOC2      EQU   27H
POMOC3      EQU   24H
FAQ         EQU   12H
JP1         EQU   23H
MINKAN      EQU   6FH     ;CISLO MINULE OBSLOUZENEHO KANALU
ZAPY        EQU   20H
ZAP0        EQU   00H
ZAP1        EQU   01H
ZAP2        EQU   02H
ZAP3        EQU   03H
ZAP4        EQU   04H
ZAP5        EQU   05H
ZAP6        EQU   06H
ZAP7        EQU   07H
PLUS        EQU   10H
NAHORE0     EQU   08H
NAHORE1     EQU   09H
NAHORE2     EQU   0AH
NAHORE3     EQU   0BH
NAHORE4     EQU   0CH
NAHORE5     EQU   0DH
NAHORE6     EQU   0EH
NAHORE7     EQU   0FH
AKZAP       EQU   17H
AKNAH       EQU   16H
TYP1        EQU   28H
TYP2        EQU   29H
TYP3        EQU   2AH
TYP4        EQU   2BH
TYP5        EQU   2CH
TYP6        EQU   2DH
TYP7        EQU   2EH
TYP8        EQU   2FH
B1          EQU   70H
B2          EQU   71H
B3          EQU   72H
B4          EQU   73H
MEZERH      EQU   30H
MEZERL      EQU   31H
PULSY       EQU   40H
PULS1H      EQU   40H     ;DELKA PULSU V uS
PULS1L      EQU   41H
PULS2H      EQU   42H
PULS2L      EQU   43H
PULS3H      EQU   44H
PULS3L      EQU   45H
PULS4H      EQU   46H
PULS4L      EQU   47H
PULS5H      EQU   48H
PULS5L      EQU   49H
PULS6H      EQU   4AH
PULS6L      EQU   4BH
PULS7H      EQU   4CH
PULS7L      EQU   4DH
PULS8H      EQU   4EH
PULS8L      EQU   4FH
        ORG   0000
        LJMP  ZAC
        ORG   0BH                   ;VEKTOR PRERUSENI OD CT0
        LJMP  OWL1
ZAC: NOP      ;Nastaveni ser.kanalu
        MOV   TMOD,#20H             ;Citac 0 rezim 2
        MOV   TH1,#-13              ;Hodnota citace
        MOV   TL1,#-13
        MOV   PCON,#10000000B       ;Prenos.rychl. 4800
        CLR   SM0                   ;Rez.1,povoleni prijem
        SETB  SM1
        MOV   IE,#10000000B         ;Povoleni.vsech.preru
        SETB  TR1                   ;Spusti casovac
        CLR   NAHORE7
        SETB  PT0
```

```
        SETB    ET0
        MOV     TH0,#-10
        MOV     TL0,#00
        ANL     TMOD,#0F0H
        ORL     TMOD,#01H
        SETB    TR0
        MOV     MEZERH,#HIGH(-4000)
        MOV     MEZERL,#LOW(-4000)
;=================================================================
;       P R O G R A M
;=================================================================
        MOV     R1,#HIGH (H1T)  ;vypis uvodni hlavicky
        MOV     R2,#LOW (H1T)
        LCALL   TISK
        CLR     ZAP0
        CLR     ZAP1
        CLR     ZAP2
        CLR     ZAP3
        CLR     ZAP4
        CLR     ZAP5
        CLR     ZAP6
        CLR     ZAP7
        CLR     FAQ
        MOV     PULS1H,#HIGH(-1310)
        MOV     PULS1L,#LOW(-1310)
        MOV     PULS2H,PULS1H
        MOV     PULS2L,PULS1L
        MOV     PULS3H,PULS1H
        MOV     PULS3L,PULS1L
        MOV     PULS4H,PULS1H
        MOV     PULS4L,PULS1L
        MOV     PULS5H,PULS1H
        MOV     PULS5L,PULS1L
        MOV     PULS6H,PULS1H
        MOV     PULS6L,PULS1L
        MOV     PULS7H,PULS1H
        MOV     PULS7L,PULS1L
        MOV     PULS8H,PULS1H
        MOV     PULS8L,PULS1L
        MOV     JP1,#0H
        MOV     TYP1,#1H
        MOV     TYP2,#1H
        MOV     TYP3,#1H
        MOV     TYP4,#1H
        MOV     TYP5,#1H
        MOV     TYP6,#1H
        MOV     TYP7,#1H
        MOV     TYP8,#1H
        MOV     MINKAN,#0H
        CLR     POMOC3
        LJMP    COMAND
        LJMP    ENT     ;skok na konec programu
H1T:    DW      H1D-H1T-2
        DB      13,10,'*****************************************'
        DB      13,10,'*     SERVOCONTROLER IC.95.A            *'
        DB      13,10,'*     ______________________            *'
        DB      13,10,'*     (c) 1995 Jiri Beran               *'
        DB      13,10,'*****************************************'
H1D:    NOP
H17T:   DW      H17D-H17T-2
        DB      13,10,'*****************************************'
        DB      13,10,'*     Nastaveni polohy kanalu           *'
        DB      13,10,'*     ________________________          *'
        DB      13,10,'* a) Primo hodnotou                     *'
        DB      13,10,'*  S1:1310H HEXA  (hodnota v us)        *'
        DB      13,10,'*  S1:1310D DECTMALNE                   *'
        DB      13,10,'* b) V procentech                       *'
        DB      13,10,'*  S1:+25%                              *'
        DB      13,10,'* Hodnota udavajici procenta musi       *'
        DB      13,10,'* zachovat znamenkem. 0 udava stredni   *'
        DB      13,10,'* polohu.Kladny smer je ve smeru        *'
        DB      13,10,'* hodin.                                *'
        DB      13,10,'*****************************************'
H17D:   NOP
;=================================================================
;       T I S K
;=================================================================
TISK:   MOV     DPH,R1
        MOV     DPL,R2
        MOV     A,#0
        MOVC    A,@A+DPTR
        MOV     R4,A
        INC     DPTR
```

```
        MOV     A,#0
        MOVC    A,@A+DPTR
        MOV     R3,A
        INC     DPTR
TISK4:  MOV     A,#0
        MOVC    A,@A+DPTR          :PRENESE ZNAK PRO VYTISKNUTI
        MOV     SBUF,A             :VYSLE ZNAK Z ACC DO KANALU
TISK6:  JBC     TI,TISK5           :Test zda je vysilac prazdny-nul TI
        LJMP    TISK6
TISK5:  NOP
        MOV     R1,DPH             :ULOZI POZICI KDE TISKNUL
        MOV     R2,DPL
        MOV     A,R3               :DEKREMENTACE R3R4 JAKO 16b.
        JNZ     SNIZ1
        MOV     A,R4
        JNZ     SNIZ2
        MOV     R3,#255            :PRO R3=0 R4=0
        MOV     R4,#255
        LJMP    ESNIZ
SNIZ2:  DEC     R4                 :PRO R3=0 R4!=0
        LJMP    ESNIZ
SNIZ1:  MOV     A,R4
        JNZ     SNIZ3
        DEC     R3                 :PRO R3!=0 R4=0
        MOV     R4,#255
        LJMP    ESNIZ
SNIZ3:  DEC     R4                 :PRO R3!=0 R4!=0
ESNIZ:  NOP
        MOV     A,R3               :ZJIST. ZDA R3 A R4 JSOU NULA
        JZ      TISK1
        LJMP    TISK2
TISK1:  MOV     A,R4
        JZ      TISK3
        LJMP    TISK2
TISK3:  LJMP    ETISK
TISK2:  MOV     DPH,R1
        MOV     DPL,R2
        INC     DPTR               :POSUNE POZICI ZNAKU
        LJMP    TISK4
ETISK:  RET                        :NAVRAT Z PRG. TISK
;=================================================================
;               K L A V E S Y
;=================================================================
KLAVESY:NOP                        :Ceka na stisk klavesy
        CLR     RI
        SETB    REN
CE1:    JBC     RI,PRI
        LJMP    CE1
PRI:    NOP
        MOV     A,SBUF
        MOV     R2,A
        XRL     A,#'H'
        JZ      J12
        MOV     A,R2
        XRL     A,#'h'
        JZ      J12
        LJMP    J11
J12:    LJMP    HE1
J11:    MOV     A,R2
        XRL     A,#13
        JZ      J14
        LJMP    J13
J14:    LJMP    HE1
J13:    MOV     A,R2
        XRL     A,#'J'
        JZ      J15
        LJMP    ENDKLA
J15:    LJMP    HE2
ENDKLA: LJMP    KLAVESY
;=================================================================
;               C O M A N D
;=================================================================
COMAND: NOP                        :Interpreter prikazu
        LCALL   PULQ
        LCALL   NEWLINE
        LCALL   PROMPT
        LCALL   CLRBUF
        LCALL   SCAN
        MOV     NEZNA,#0
HELP:   NOP                        :HELP
        LJMP    HELP02
HELP01: DB      '*elp   '
HELP03: DB      'HELP   '
```

```
HELP04:   DB    'H.    '
HELP05:   DB    'h.    '
HELP02:   NOP
          MOV    R1,#HIGH(HELP01)
          MOV    R2,#LOW(HELP01)
          LCALL    PORO
          JZ    HELD01
          LCALL    HELPQ
          MOV    NEZNA,#1
HELD01:   MOV    R1,#HIGH(HELP03)
          MOV    R2,#LOW(HELP03)
          LCALL    PORO
          JZ    HELD02
          LCALL    HELPQ
          MOV    NEZNA,#1
HELD02:   MOV    R1,#HIGH(HELP04)
          MOV    R2,#LOW(HELP04)
          LCALL    PORO
          JZ    HELD03
          LCALL    HELPQ
          MOV    NEZNA,#1
HELD03:   MOV    R1,#HIGH(HELP05)
          MOV    R2,#LOW(HELP05)
          LCALL    PORO
          JZ    HELD
          LCALL    HELPQ
          MOV    NEZNA,#1
HELD:     NOP            ;=========================================
STAV:     NOP            ;STAV
          LJMP    STAV02
STAV01:   DB    'STAV?  '
STAV02:   MOV    R1,#HIGH(STAV01)
          MOV    R2,#LOW(STAV01)
          LCALL    PORO
          JZ    STAD
          LCALL    STAVQ
          MOV    NEZNA,#1
STAD:     NOP            ;=========================================
ZAPNI:    NOP            ;ZAPNI
          LJMP    ZAPNI02
ZAPNI01:DB    'S*:ON  '
ZAPNI02:MOV    R1,#HIGH(ZAPNI01)
          MOV    R2,#LOW(ZAPNI01)
          LCALL    PORO
          JZ    ZAPND
          LCALL    ZAPNIQ
          MOV    NEZNA,#1
ZAPND:    NOP            ;=========================================
ZAPNV:    NOP            ;ZAPNI    VSECHNY
          MOV    R1,#HIGH(DCPR6)
          MOV    R2,#LOW(DCPR6)
          LCALL    PORO
          JZ    ZAPNVD
          LCALL    ZAPNVQ
          MOV    NEZNA,#1
ZAPNVD:   NOP            ;=========================================
VYPNV:    NOP            ;VYPNI VSECHNY
          MOV    R1,#HIGH(DCPR7)
          MOV    R2,#LOW(DCPR7)
          LCALL    PORO
          JZ    VYPNVD
          LCALL    VYPNVQ
          MOV    NEZNA,#1
VYPNVD:   NOP            ;=========================================
VYPNI:    NOP            ;VYPNI
          LJMP    VYPNI02
VYPNI01:DB    'S*:OFF  '
VYPNI02:MOV    R1,#HIGH(VYPNI01)
          MOV    R2,#LOW(VYPNI01)
          LCALL    PORO
          JZ    VYPND
          LCALL    VYPNIQ
          MOV    NEZNA,#1
VYPND:    NOP            ;=========================================
NASTAV:   NOP            ;NASTAVI HODNOTOU HEXA
          LJMP    NASTAV2
NASTAV1:DB    'S*:****H '
NASTAV2:MOV    R1,#HIGH(NASTAV1)
          MOV    R2,#LOW(NASTAV1)
          LCALL    PORO
          JZ    NASTAD
          LCALL    NASTAVQ
          MOV    NEZNA,#1
```

```
NASTAD:   NOP          ;=========================================
DASTAV:   NOP          ;NASTAVI HODNOTU DECIMALNE
          LJMP   DASTAV2
DASTAV1:DB    'S*:****D '
DASTAV2:MOV    R1,#HIGH(DASTAV1)
          MOV    R2,#LOW(DASTAV1)
          LCALL    PORO
          JZ    DASTAD
          LCALL    DASTAVQ
          MOV    NEZNA,#1
DASTAD:   NOP          ;=========================================
QASTA:    NOP          ;PRO S*:***D
          MOV    R1,#HIGH(DCPR5)
          LCALL    PORO
          JZ    QUASTAD
          LCALL    QASTAQ
          MOV    NEZNA,#1H
QUASTAD:NOP            ;=========================================
DATYP:    NOP          ;NASTAVI TYP
          LJMP   DATYP2
DATYP1:   DB    'S*:TYP* '
DATYP2:   MOV    R1,#HIGH(DATYP1)
          MOV    R2,#LOW(DATYP1)
          LCALL    PORO
          JZ    DATYPD
          LCALL    DATYPQ
          MOV    NEZNA,#1
DATYPD:   NOP          ;=========================================
PROC0:    NOP          ;PRO 0%
          MOV    R1,#HIGH(DCPR0)
          MOV    R2,#LOW(DCPR0)
          LCALL    PORO
          JZ    PROC0D
          LCALL    PROC0Q
          MOV    NEZNA,#1
PROC0D:   NOP          ;=========================================
PROC2:    NOP          ;PRO -**%
          MOV    R1,#HIGH(DCPR2)
          MOV    R2,#LOW(DCPR2)
          LCALL    PORO
          JZ    PROC2D
          LCALL    PROC2Q
          MOV    NEZNA,#1
PROC2D:   NOP          ;=========================================
PROC1:    NOP          ;PRO +**%
          MOV    R1,#HIGH(DCPR1)
          MOV    R2,#LOW(DCPR1)
          LCALL    PORO
          JZ    PROC1D
          LCALL    PROC1Q
          MOV    NEZNA,#1
PROC1D:   NOP          ;=========================================
STO1:  NOP             ;PRO +100%
          MOV    R1,#HIGH(DCPS)
          MOV    R2,#LOW(DCPS)
          LCALL  PORO
          JZ    STO1D
          LCALL  STO1Q
          MOV    NEZNA,#1
STO1D: NOP
STZ1:  NOP             ;PRO -100%
          MOV    R1,#HIGH(DCZS)
          MOV    R2,#LOW(DCZS)
          LCALL  PORO
          JZ    STZ1D
          LCALL  STZ1Q
          MOV    NEZNA,#1
STZ1D: NOP
PROC4:    NOP          ;PRO -**%
          MOV    R1,#HIGH(DCPR4)
          MOV    R2,#LOW(DCPR4)
          LCALL    PORO
          JZ    PROC4D
          LCALL    PROC4Q
          MOV    NEZNA,#1
PROC4D:   NOP          ;=========================================
PROC3:    NOP          ;PRO **%
          MOV    R1,#HIGH(DCPR3)
          MOV    R2,#LOW(DCPR3)
          LCALL    PORO
          JZ    PROC3D
          LCALL    PROC3Q
          MOV    NEZNA,#1
```

```
PROC3D:   NOP              ;=================================
          MOV   R1,#HIGH(DCH1) ;H1
          MOV   R2,#LOW(DCH1)
          LCALL  PORO
          JZ    QDCH1D
          LCALL  PDCH1Q
          MOV   NEZNA,#1
QDCH1D: NOP
          MOV   R1,#HIGH(DCH2)
          MOV   R2,#LOW(DCH2)
          LCALL  PORO
          JZ    QDCH2D
          LCALL  PDCH2Q
          MOV   NEZNA,#1
QDCH2D: NOP
          MOV   R1,#HIGH(DCH3)
          MOV   R2,#LOW(DCH3)
          LCALL  PORO
          JZ    QDCH3D
          LCALL  STAVQ
          MOV   NEZNA,#1
QDCH3D: NOP
          MOV   R1,#HIGH(DCH4)
          MOV   R2,#LOW(DCH4)
          LCALL  PORO
          JZ    QDCH4D
          LCALL  PDCH4Q
          MOV   NEZNA,#1
QDCH4D: NOP
          MOV   R1,#HIGH(DCH5)
          MOV   R2,#LOW(DCH5)
          LCALL  PORO
          JZ    QDCH5D
          LCALL  PDCH5Q
          MOV   NEZNA,#1
QDCH5D: NOP
          MOV   R1,#HIGH(DCH6)
          MOV   R2,#LOW(DCH6)
          LCALL  PORO
          JZ    QDCH6D
          LCALL  PDCH6Q
          MOV   NEZNA,#1
QDCH6D: NOP
          MOV   R1,#HIGH(DCH7)
          MOV   R2,#LOW(DCH7)
          LCALL  PORO
          JZ    QDCH7D
          LCALL  PDCH7Q
          MOV   NEZNA,#1
QDCH7D: NOP
          MOV   A,NEZNA          ;NEBYL-LI PRIKAZ VYKONAN ZAVOLA NEZNAM
          JZ    COMA01
          LJMP    ENDCOM
COMA01:   LCALL   NEZNAM
ENDCOM:   LJMP    COMAND
PDCH1Q: MOV   R1,#HIGH(KHE07T)
          MOV   R2,#LOW(KHE07T)
          LCALL  TISK
          RET
PDCH2Q: MOV   R1,#HIGH(KHE04T)
          MOV   R2,#LOW(KHE04T)
          LCALL  TISK
          RET
PDCH3Q: NOP                ; PRESMEROVANO NA STAV
PDCH4Q: MOV   R1,#HIGH(KHE05T)
          MOV   R2,#LOW(KHE05T)
          LCALL  TISK
          RET
PDCH5Q: MOV   R1,#HIGH(KHE06T)
          MOV   R2,#LOW(KHE06T)
          LCALL  TISK
          RET
PDCH6Q: MOV   R1,#HIGH(KHE03T)
          MOV   R2,#LOW(KHE03T)
          LCALL  TISK
          RET
PDCH7Q: MOV   R1,#HIGH(KHE02T)
          MOV   R2,#LOW(KHE02T)
          LCALL  TISK
          RET
;=================================================================
;          P U L Q
;=================================================================
```

```
PULQ: NOP    ;ZJISTI POCET ZAPNUTYCH SERV
        MOV    A,ZAPY
        CPL    A
        MOV    B,#0H
        JNB    ACC.0,PRPUL1
        INC    B
PRPUL1: NOP
        JNB    ACC.1,PRPUL2
        INC    B
PRPUL2: NOP
        JNB    ACC.2,PRPUL3
        INC    B
PRPUL3: NOP
        JNB    ACC.3,PRPUL4
        INC    B
PRPUL4: NOP
        JNB    ACC.4,PRPUL5
        INC    B
PRPUL5: NOP
        JNB    ACC.5,PRPUL6
        INC    B
PRPUL6: NOP
        JNB    ACC.6,PRPUL7
        INC    B
PRPUL7: NOP
        JNB    ACC.7,PRPUL8
        INC    B
PRPUL8: NOP
        MOV    R1,B
        INC    R1    ;ABY NEBYLA NULA
        MOV    A,R1
        MOV    R2,A
        MOV    R0,#HIGH(500)
        MOV    R1,#LOW(500)
        LCALL  NASOB
        MOV    A,R4
        MOV    R1,A
        MOV    A,R5
        MOV    R2,A
        LCALL  XADE
        MOV    70H,R4
        MOV    71H,R5
        LCALL  TUXAM
        MOV    MEZERH,70H
        MOV    MEZERL,71H
        RET
;------------------------------------------------------------------------
;          N E Z N A M Y  P R I K A Z
;------------------------------------------------------------------------
NEZNAM:   NOP            ;NEZNAMY PRIKAZ
        LCALL    NEWLINE
        MOV    R1,#HIGH(NEZNAT)
        MOV    R2,#LOW(NEZNAT)
        LCALL    TISK
        LCALL    UKABUF
        MOV    A,#' '
        LCALL    VYSLI
        MOV    A,#' '
        LCALL    VYSLI
        MOV    A,#'?'
        LCALL    VYSLI
        LCALL    NEWLINE
        LJMP    NEZNAD
NEZNAT:   DW    NEZNAD-NEZNAT-2
        DB    'Co znamena '
NEZNAD:   NOP
        RET
;------------------------------------------------------------------------
;          S Y N T A X
;------------------------------------------------------------------------
SYNTAX:   NOP            ;SYNTAX ERROR
        LCALL    NEWLINE
        MOV    R1,#HIGH(SYNTAT)
        MOV    R2,#LOW(SYNTAT)
        LCALL    TISK
        LCALL    UKABUF
        MOV    A,#' '
        LCALL    VYSLI
        MOV    A,#' '
        LCALL    VYSLI
        MOV    A,#'?'
        LCALL    VYSLI
        LCALL    NEWLINE
```

```
        LJMP    NEZNAD
SYNTAT:   DW    SYNTAD-SYNTAT-2
        DB    'Syntax ERROR '
SYNTAD:RET
;-----------------------------------------------------------------------
;         H E L P Q
;-----------------------------------------------------------------------
HELPQ: MOV    R1,#HIGH(KHE01T)
        MOV    R2,#LOW(KHE01T)
        LCALL  TISK
        RET
;-----------------------------------------------------------------------
;         ZAPNVQ
;-----------------------------------------------------------------------
ZAPNVQ:   MOV    ZAPY,#0FFH
        RET
;-----------------------------------------------------------------------
;         VYPNVQ
;-----------------------------------------------------------------------
VYPNVQ:   MOV    ZAPY,#0H
        RET
;-----------------------------------------------------------------------
;         PROC3Q
;-----------------------------------------------------------------------
PROC3Q:MOV    A,75H
        MOV    76H,A
        MOV    A,74H
        MOV    75H,A
        MOV    A,#'0'
        MOV    74H,A
        CLR    PLUS
        LCALL    PROC2M
        RET
STZ1Q: MOV    74H,#'9'
        MOV    75H,#'A'
        CLR    PLUS
        LCALL  PROC2M
        RET
;-----------------------------------------------------------------------
;         PROC1Q
;-----------------------------------------------------------------------
PROC1Q:MOV    A,75H
        MOV    76H,A
        MOV    75H,A
        MOV    A,#'0'
        MOV    74H,A
        LCALL    PROC2Q
        RET
;-----------------------------------------------------------------------
;         ST01Q
;-----------------------------------------------------------------------
STO1Q: MOV    74H,#'9'
        MOV    75H,#'A'
        LCALL  PROC2Q
        RET
;-----------------------------------------------------------------------
;         PROC4Q
;-----------------------------------------------------------------------
PROC4Q:   NOP              ;NASTAVI PLUS NA 0
        CLR    PLUS
        LCALL    PROC2M
        RET
;-----------------------------------------------------------------------
;         PROC2Q
;-----------------------------------------------------------------------
PROC2Q:SETB    PLUS
PROC2M:   NOP              ;DOSADI STREDNI HODNOTU PRO DANY TYP
        LCALL    KERSER          ;KTERE SERVO JE V PRIKAZU -> A
        DEC    A
        MOV    POMOC2,A
        MOV    B1,A
        MOV    B2,A
        MOV    R1,A
        MOV    R2,#8H
        LCALL  JEVETSA
        XRL    A,#2H
        JZ     NAPO1
        LCALL  SYNTAX
        LJMP   ERREND
NAPO1:    MOV    A,B1
        LCALL    KERTYP          ;JAKEHO TYPU JE ->A
        DEC    A
        MOV    B1,A
```

```
        MOV     R1,A
        MOV     R2,#0H
        MOV     A,B1
        LCALL   VYTYHO          ;VYBERE TYPOVE HODNOTY
        MOV     POMOC,B2
        MOV     70H,R1          ;STREDNI HODNOTA H
        MOV     71H,R2          ;STREDNI HODNOTA L
        MOV     72H,R3          ;TOLERANCE H
        MOV     73H,R4          ;TOLERANCE L
        MOV     R1,72H
        MOV     R2,73H
        MOV     R1,74H
        MOV     R2,75H
        MOV     60H,70H         ;STREDNI HODNOTA H
        MOV     61H,71H         ;STREDNI HODNOTA L
        MOV     62H,72H         ;TOLERANCE H
        MOV     63H,73H         ;TOLERANCE L
        MOV     64H,74H
        MOV     65H,75H
        MOV     66H,76H
        MOV     73H,#'0'
        MOV     74H,#'0'
        MOV     75H,64H
        MOV     76H,65H
        LCALL   DEXA            ;NA 70 71 CISLA HEXA PROCENT
        MOV     R1,70H
        MOV     R2,71H
        MOV     R0,62H
        MOV     R2,#64H
        LCALL   VYDEL
        MOV     R2,71H
        LCALL   NASOB           ; V R4 R5 MAME ODCHYLKU
        JB      PLUS,POKR       ;NENI LI PLUS UDELA ZAPORNOU HODNOTU
        MOV     70H,R4
        MOV     71H,R5
        LCALL   TUXAM
        MOV     R4,70H
        MOV     R5,71H
POKR:   MOV     A,61H
        ADD     A,R5
        MOV     71H,A
        MOV     A,60H
        ADDC    A,R4
        MOV     70H,A           ;NA 70 71 MAME POZADOVANOU HODNOTU
        MOV     A,61H           ;MAX TEORETICKA HODNOTA V R3 R4
        ADD     A,63H
        MOV     R4,A
        MOV     A,60H
        ADDC    A,62H
        MOV     R3,A
        MOV     A,#'@'
        MOV     A,R3
        MOV     R1,A
        MOV     A,R4
        MOV     R2,A
        MOV     A,70H
        MOV     R1,A
        MOV     A,71H
        MOV     R2,A
        MOV     A,70H
        MOV     R5,A
        MOV     A,71H
        MOV     R6,A
        LCALL   JEVETSI
        XRL     A,#2H
        JNZ     P00WE
        MOV     A,#'A'
        LCALL   NEWLINE
        MOV     R1,#HIGH(TPR11)
        MOV     R2,#LOW(TPR11)
        LCALL   TISK
        LJMP    ERREND
P00WE:;JE LI PULS > STRED- TOLERANCE GOTO ERREND
        ;SECTE STRED + TOLERANCE
        ;60 61   62 63  DO  R3 R4
        ;UDELA ZAPORNOU TOLERANCI
        MOV     A,70H   ;UKLIDI 70 71
        MOV     R3,A
        MOV     A,71H
        MOV     R4,A
        MOV     A,62H
        MOV     70H,A
        MOV     A,63H
```

```
        MOV     71H,A
        LCALL   TUXAM
        MOV     A,70H
        MOV     62H,A
        MOV     A,71H
        MOV     63H,A
        MOV     A,R3
        MOV     70H,A
        MOV     A,R4
        MOV     71H,A
        MOV     A,61H     ;MAX TEORETICKA HODNOTA V R3 R4
        ADD     A,63H
        MOV     R4,A
        MOV     A,60H
        ADDC    A,62H
        MOV     R3,A
        MOV     A,#'S'
        MOV     A,R3
        MOV     R1,A
        MOV     A,R4
        MOV     R2,A
        MOV     A,70H
        MOV     R1,A
        MOV     A,71H
        MOV     R2,A
        MOV     A,70H
        MOV     R5,A
        MOV     A,71H
        MOV     R6,A
        LCALL   JEVETSI
        XRL     A,#1H
        JNZ     P01WE
        MOV     A,#'&'
        LCALL   NEWLINE
        MOV     R1,#HIGH(TPR11)
        MOV     R2,#LOW(TPR11)
        LCALL   TISK
        LJMP    ERREND
P01WE:LCALL     TUXAM
        MOV     A,POMOC2
        MOV     B,#2H
        MUL     AB
        ADD     A,#PULS1H
        MOV     R1,A
        MOV     @R1,70H
        INC     R1
        MOV     @R1,71H
ERREND:  RET
;=================================================================
;          PROC0Q
;=================================================================
PROC0Q:  NOP            ;0 PROCENT
        NOP             ;DOSADI STREDNI HODNOTU PRO DANY TYP
        LCALL   KERSER          ;KTERE SERVO JE V PRIKAZU -> A
        DEC     A
        MOV     B1,A
        MOV     B2,A
        MOV     R1,A
        MOV     R2,#0H
        MOV     R3,A
        MOV     R2,#9H
        LCALL   JEVETSA
        XRL     A,#2H
        JZ      NAP01
        LCALL   SYNTAX
        LJMP    ER0END
NAP01:  MOV     A,B1
        LCALL   KERTYP          ;JAKEHO TYPU JE ->A
        DEC     A
        MOV     B1,A
        MOV     R1,A
        MOV     R2,#0H
        MOV     A,B1
        LCALL   VYTYHO          ;VYBERE TYPOVE HODNOTY
        MOV     POMOC,B2
        MOV     70H,R1
        MOV     71H,R2
        MOV     72H,R3
        MOV     73H,R4
        MOV     R1,72H
        MOV     R2,73H
        LCALL   TUXAM
        LCALL   NASTB           ;DOSADI STREDNI HODNOTU DO PULSU
```

```
ER0END:   RET
;=================================================================
;         VYTYHO
;=================================================================
VYTYHO:   NOP          ;VYBERE TYPOVE HODNOTY DO R1 R2 R3 R4
          MOV   DPH,#HIGH(DEFTYP)
          MOV   DPL,#LOW(DEFTYP)
          MOV   B,#4H
          MUL   AB
          MOV   R6,A
          MOVC  A,@A+DPTR
          MOV   R1,A
          INC   R6
          MOV   A,R6
          MOVC  A,@A+DPTR
          MOV   R2,A
          INC   R6
          MOV   A,R6
          MOVC  A,@A+DPTR
          MOV   R3,A
          INC   R6
          MOV   A,R6
          MOVC  A,@A+DPTR
          MOV   R4,A
          RET
;=================================================================
;         K E R T Y P
;=================================================================
KERTYP:   NOP          ;VRATI V A TYP SERVA A
          MOV   POMOC,A
          MOV   A,#TYP1
          ADD   A,POMOC
          MOV   R1,A
          MOV   A,@R1
          RET
;=================================================================
;         K E R S E R
;=================================================================
KERSER:   NOP          ;VYBERE SERVO DO A
          MOV   A,71H
          MOV   R1,A
          LCALL   TUXA
          MOV   A,R3
          RET
;=================================================================
;         Z A P N I
;=================================================================
ZAPNIQ:   NOP
ZAPNSS:   NOP
          MOV   A,#'&'
          MOV   R1,#71H          ;Do R1 * NEBO 1-8
          MOV   A,@R1
          MOV   R1,A
          XRL   A,#'1'
          JNZ   ZAPN02
          SETB  ZAP0
ZAPN02:   MOV   A,R1
          XRL   A,#'2'
          JNZ   ZAPN03
          SETB  ZAP1
ZAPN03:   MOV   A,R1
          XRL   A,#'3'
          JNZ   ZAPN04
          SETB  ZAP2
ZAPN04:   MOV   A,R1
          XRL   A,#'4'
          JNZ   ZAPN05
          SETB  ZAP3
ZAPN05:   MOV   A,R1
          XRL   A,#'5'
          JNZ   ZAPN06
          SETB  ZAP4
ZAPN06:   MOV   A,R1
          XRL   A,#'6'
          JNZ   ZAPN07
          SETB  ZAP5
ZAPN07:   MOV   A,R1
          XRL   A,#'7'
          JNZ   ZAPN08
          SETB  ZAP6
ZAPN08:   MOV   A,R1
          XRL   A,#'8'
          JNZ   ZAPN09
```

```
        SETB    ZAP7
ZAPN09: MOV     A,R1
        XRL     A,#'*'
        JNZ     ZAPN0A
        LCALL   SYNTAX
ZAPN0A: RET
;=================================================================
;          V Y P N I
;=================================================================
VYPNIQ:  NOP
VYPNSS:  MOV     A,#'&'
         MOV     R1,#71H          ;Do R1 * NEBO 1-8
         MOV     A,@R1
         MOV     R1,A
         XRL     A,#'1'
         JNZ     VYPN02
         CLR     ZAP0
VYPN02:  MOV     A,R1
         XRL     A,#'2'
         JNZ     VYPN03
         CLR     ZAP1
VYPN03:  MOV     A,R1
         XRL     A,#'3'
         JNZ     VYPN04
         CLR     ZAP2
VYPN04:  MOV     A,R1
         XRL     A,#'4'
         JNZ     VYPN05
         CLR     ZAP3
VYPN05:  MOV     A,R1
         XRL     A,#'5'
         JNZ     VYPN06
         CLR     ZAP4
VYPN06:  MOV     A,R1
         XRL     A,#'6'
         JNZ     VYPN07
         CLR     ZAP5
VYPN07:  MOV     A,R1
         XRL     A,#'7'
         JNZ     VYPN08
         CLR     ZAP6
VYPN08:  MOV     A,R1
         XRL     A,#'8'
         JNZ     VYPN09
         CLR     ZAP7
VYPN09:  MOV     A,R1
         XRL     A,#'*'
         JNZ     VYPN0A
         LCALL   SYNTAX
VYPN0A:  RET
;=================================================================
;          QASTAQ
;=================================================================
QASTAQ:MOV       A,76H
       MOV       77H,A
       MOV       A,75H
       MOV       76H,A
       MOV       A,74H
       MOV       75H,A
       MOV       A,73H
       MOV       74H,A
       MOV       A,#'0'
       MOV       73H,A
       LCALL     DASTAVQ
       RET
;=================================================================
;          D A S T A V
;=================================================================
DASTAVQ:MOV      R1,#71H
        MOV      A,@R1
        LCALL    KERSER
        DEC      A
        MOV      POMOC,A
        LCALL    DEXA        ;DEKADICKOU NA HEXA
        NOP                  ;VYPSAT HEXA HODNOTU
        MOV      R1,70H
        MOV      R2,71H
        MOV      A,POMOC
        LCALL    NASTP
        RET
;=================================================================
;          N A S T A V
;=================================================================
```

```
NASTAVQ:MOV    R1,#71H        ;UKLIDI CISLO KANALU
        MOV    A,@R1
        LCALL    KERSER
        DEC    A
        MOV    POMOC,A
        MOV    POMOC2,A
        LCALL    TUXA12
        MOV    A,POMOC2
NASTP:   NOP
KAN07:   NOP              ;KANAL *
        MOV    POMOC2,A
        LCALL    KERTYP
        DEC    A
        LCALL    VYTYHO
        MOV    A,R1
        MOV    60H,A
        MOV    A,R2
        MOV    61H,A
        MOV    A,R3
        MOV    62H,R3
        MOV    A,R4
        MOV    63H,R4
           ;MAX TEOR HODNOTA
        MOV    A,61H
        ADD    A,63H
        MOV    R4,A
        MOV    A,60H
        ADDC    A,62H
        MOV    R3,A
           ;POROVNANI
        MOV    A,70H
        MOV    R5,A
        MOV    A,71H
        MOV    R6,A
        LCALL    JEVETSI
        XRL    A,#2H
        JNZ    P00QW
        LCALL    NEWLINE
        MOV    R1,#HIGH(TPR11)
        MOV    R2,#LOW(TPR11)
        LCALL    TISK
        LJMP    ERAEND
P00QW:   NOP
        MOV    A,70H
        MOV    R3,A
        MOV    A,71H
        MOV    R4,A
        MOV    A,62H
        MOV    70H,A
        MOV    A,63H
        MOV    71H,A
        LCALL    TUXAM
        MOV    A,70H
        MOV    62H,A
        MOV    A,71H
        MOV    63H,A
        MOV    A,R3
        MOV    70H,A
        MOV    A,R4
        MOV    71H,A
        MOV    A,61H          ;MAX TEOR HODNOTA
        ADD    A,63H
        MOV    R4,A
        MOV    A,60H
        ADDC    A,62H
        MOV    R3,A
        MOV    A,70H
        MOV    R5,A
        MOV    A,71H
        MOV    R6,A
        LCALL    JEVETSI
        XRL    A,#1H
        JNZ    P01QW
        LCALL    NEWLINE
        MOV    R1,#HIGH(TPR11)
        MOV    R2,#LOW(TPR11)
        LCALL    TISK
  L     JMP    ERAEND
P01QW:
NASTC:   NOP              ;NASTAVI PULSY
        LCALL    TUXAM
NASTB:   MOV    A,POMOC
        MOV    B,#2H
```

```
        MUL   AB
        ADD   A,#PULS1H
        MOV   R1,A
        MOV   @R1,70H
        INC   R1
        MOV   @R1,71H
ERAEND:   NOP
EKAN07:   NOP
        RET
;-----------------------------------------------------------------------
;         DATYP
;-----------------------------------------------------------------------
DATYPQ:   NOP          ;NASTAVY TYP
        MOV   R1,71H
        LCALL   TUXA
        MOV   A,#27H
        ADD   A,R3
        MOV   R4,A
        MOV   R1,76H
        MOV   R3,#00FFH
        LCALL   TUXA
        ;V R3 MAME CISLO TYPU
        MOV   A,R3
        MOV   R1,A        ;NE VETSI NEX 0FH
        MOV   R2,#0FH
        LCALL   JEVETSA
        XRL   A,#1H
        JNZ   QDATYP
        LCALL   SYNTAX
        LJMP   QDBTYP
QDATYP:   NOP          ;NE 0
        MOV   A,R3
        MOV   R1,A
        MOV   R2,#8H
        LCALL   JEVETSA
        XRL   A,#8H
        JNZ   QDBTYP
        LCALL   SYNTAX
QDBTYP:   MOV   A,R4
        MOV   R1,A
        MOV   A,R3
        MOV   @R1,A
ENDDAT:   RET
;-----------------------------------------------------------------------
;         TUXAM
;-----------------------------------------------------------------------
TUXAM:   NOP          ;Z HODNOTY NA 70 71 UDELA ZAPORNOU
        MOV   R1,70H
        MOV   R2,71H
        MOV   A,R2
        CPL   A
        MOV   R2,A
        INC   R2
        MOV   A,R1
        CPL   A
        MOV   R1,A
        MOV   A,R2
        JNZ   ENDTUX
        INC   R1
ENDTUX:   MOV   70HLR1
        MOV   71HLR2
        RET
;-----------------------------------------------------------------------
;         T U X A 1 2
;-----------------------------------------------------------------------
TUXA12:   NOP          ;ZNAKY Z 73 74 75 76  DO 70 71 JAKO CISLA
        MOV   R1,73H
        LCALL   TUXA
        MOV   A,R3
        SWAP   A
        MOV   70HLA
        MOV   R1,74H
        LCALL   TUXA
        MOV   A,70H
        ADD   A,R3
        MOV   70HLA
        MOV   R1,75H
        LCALL   TUXA
        MOV   A,R3
        SWAP   A
        MOV   71HLA
        MOV   R1,76H
        LCALL   TUXA
```

```
        MOV     A,71H
        ADD     A,R3
        MOV     71H,A
        RET
;------------------------------------------------------------------------
;               T U X A
;------------------------------------------------------------------------
TUXA:   NOP             ;ZNAK Z R1 DO R3 JAKO CISLO
        MOV     A,R1
        XRL     A,#'0'
        JNZ     TUXA1
        MOV     R3,#0H
TUXA1:  MOV     A,R1
        XRL     A,#'1'
        JNZ     TUXA2
        MOV     R3,#1H
TUXA2:  MOV     A,R1
        XRL     A,#'2'
        JNZ     TUXA3
        MOV     R3,#2H
TUXA3:  MOV     A,R1
        XRL     A,#'3'
        JNZ     TUXA4
        MOV     R3,#3H
TUXA4:  MOV     A,R1
        XRL     A,#'4'
        JNZ     TUXA5
        MOV     R3,#4H
TUXA5:  MOV     A,R1
        XRL     A,#'5'
        JNZ     TUXA6
        MOV     R3,#5H
TUXA6:  MOV     A,R1
        XRL     A,#'6'
        JNZ     TUXA7
        MOV     R3,#6H
TUXA7:  MOV     A,R1
        XRL     A,#'7'
        JNZ     TUXA8
        MOV     R3,#7H
TUXA8:  MOV     A,R1
        XRL     A,#'8'
        JNZ     TUXA9
        MOV     R3,#8H
TUXA9:  MOV     A,R1
        XRL     A,#'9'
        JNZ     TUXAA
        MOV     R3,#9H
TUXAA:  MOV     A,R1
        XRL     A,#'A'
        MOV     R3,#0AH
        JNZ     TUXAB
TUXAB:  MOV     A,R1
        XRL     A,#'B'
        JNZ     TUXAC
        MOV     R3,#0BH
TUXAC:  MOV     A,R1
        XRL     A,#'C'
        JNZ     TUXAD
        MOV     R3,#0CH
TUXAD:  MOV     A,R1
        XRL     A,#'D'
        JNZ     TUXAE
        MOV     R3,#0DH
TUXAE:  MOV     A,R1
        XRL     A,#'E'
        JNZ     TUXAF
        MOV     R3,#0EH
TUXAF:  MOV     A,R1
        XRL     A,#'F'
        JNZ     TUXAEND
        MOV     R3,#0FH
TUXAEND:RET
;------------------------------------------------------------------------
;               X A D E
;------------------------------------------------------------------------
XADE:   NOP             ;HEXA V R1 R2 NA DEC R1 R2
        MOV     DPH,#0H
        MOV     DPL,#0H
        MOV     B1,#0H
        MOV     B2,#0H
        MOV     B3,#0H
        MOV     B4,#0H
```

```
XADE0:   MOV    A,DPH
         XRL    A,R1
         JNZ    XADEDA
         MOV    A,DPL
         XRL    A,R2
         JNZ    XADEDA
         LJMP   XADEEN
XADEDA:  INC    DPTR        ;DPTR++
         INC    B4
         MOV    A,B4
         XRL    A,#0AH
         JNZ    XADEES
         MOV    B4,#0H
         INC    B3
         MOV    A,B3
         XRL    A,#0AH
         JNZ    XADEES
         MOV    B3,#0H
         INC    B2
         MOV    A,B2
         XRL    A,#0AH
         JNZ    XADEES
         MOV    B2,#0H
         INC    B1
         MOV    A,B1
         XRL    A,#0AH
         JNZ    XADEES
         MOV    B1,#0H
         INC    B4
XADEES   LJMP   XADE0
XADEEN:  NOP
         MOV    A,#0H
         MOV    A,B1
         SWAP   A
         ADD    A,B2
         MOV    R1,A
         MOV    A,#0H
         MOV    A,B3
         SWAP   A
         ADD    A,B4
         MOV    R2,A
XADEEND:RET
;-----------------------------------------------------------------------
;          D E X A
;-----------------------------------------------------------------------
;DEKADICKE CISLO KTERE JE NA ADR. 73 74 75 76 PREVEDE NA HEXA NA 70 71
DEXA: MOV    R1,73H
         LCALL   TUXA
         MOV     73H,R3
         MOV     R1,74H
         LCALL   TUXA
         MOV     74H,R3
         MOV     R1,75H
         LCALL   TUXA
         MOV     75H,R3
         MOV     R1,76H
         LCALL   TUXA
         MOV     76H,R3
         MOV     R2,73H
         MOV     R0,#03H
         MOV     R1,#0E8H
         LCALL   NASOB
         MOV     70H,R4
         MOV     71H,R5
         MOV     R2,74H
         MOV     R0,#00H
         MOV     R1,#64H
         LCALL   NASOB
         CLR     C
         MOV     A,71H
         ADD     A,R5
         MOV     71H,A
         MOV     A,70H
         ADDC    A,R4
         MOV     70H,A
         MOV     R2,75H
         MOV     R0,#0H
         MOV     R1,#0AH
         LCALL   NASOB
         CLR     C
         MOV     A,71H
         ADD     A,R5
         MOV     71H,A
```

```
        MOV   A,70H
        ADDC  A,R4
        MOV   70H,A
        MOV   R5,76H
        MOV   R4,#0H
        CLR   C
        MOV   A,71H
        ADD   A,R5
        MOV   71H,A
        MOV   A,70H
        ADDC  A,R4
        MOV   70H,A
DEXAEND:RET
;-----------------------------------------------------------------------
;          VYHEHO
;-----------------------------------------------------------------------
VYHEHO:   NOP            ;VYPISE HEX Z R1 R2 NA SERKA
        MOV   A,#' '
        LCALL   VYSLI
        MOV   A,R1
        MOV   R5,A
        CLR   ACC.3
        CLR   ACC.2
        CLR   ACC.1
        CLR   ACC.0
        SWAP  A
        MOV   R1,A
        LJMP  VYHE1
VYHEZ:  DB    '0123456789ABCDEF'
VYHE1:MOV   DPH,#HIGH(VYHEZ)
        MOV   DPL,#LOW(VYHEZ)
        MOV   A,R1
        MOVC  A,@A+DPTR
        LCALL   VYSLI
        MOV   A,R5
        CLR   ACC.4
        CLR   ACC.5
        CLR   ACC.6
        CLR   ACC.7
        MOV   R1,A
        MOV   DPH,#HIGH(VYHEZ)
        MOV   DPL,#LOW(VYHEZ)
        MOVC  A,@A+DPTR
        LCALL   VYSLI
        MOV   A,R2
        CLR   ACC.3
        CLR   ACC.2
        CLR   ACC.1
        CLR   ACC.0
        SWAP  A
        MOV   DPH,#HIGH(VYHEZ)
        MOV   DPL,#LOW(VYHEZ)
        MOVC  A,@A+DPTR
        LCALL   VYSLI
        MOV   A,R2
        CLR   ACC.4
        CLR   ACC.5
        CLR   ACC.6
        CLR   ACC.7
        MOV   DPH,#HIGH(VYHEZ)
        MOV   DPL,#LOW(VYHEZ)
        MOVC  A,@A+DPTR
        LCALL   VYSLI
        RET
;-----------------------------------------------------------------------
;          N A S O B
;-----------------------------------------------------------------------
NASOB:    NOP            ;(R0 R1)*R2 -> (R3 R4 R5)
        CLR   C
        MOV   A,R1
        MOV   B,R2
        MUL   AB
        MOV   R5,A
        MOV   R4,B
        MOV   A,R0
        MOV   B,R2
        MUL   AB
        ADD   A,R4
        MOV   R4,A
        MOV   A,B
        ADDC  A,#0
        MOV   R3,A
        RET
```

```
;-----------------------------------------------------------------------
;        JEVETSI
;-----------------------------------------------------------------------
JEVETSI:NOP            : R3 R4  R5 R6
        MOV    A,R3
        MOV    R1,A
        MOV    A,R5
        MOV    R2,A
        LCALL   JEVETSA
        JNZ    JEVE01
        MOV    A,R4
        MOV    R1,A
        MOV    A,R6
        MOV    R2,A
        LCALL   JEVETSA
        JNZ    JEVE01
        MOV    A,#0H
        LJMP   JEVETD
JEVE01:  MOV    A,R3        ;A -> C
        MOV    R1,A
        MOV    A,R5
        MOV    R2,A
        LCALL   JEVETSA
        MOV    B,A
        XRL    A,#1H
        JNZ    JEVE02
        MOV    A,#1H
        LJMP   JEVETD
JEVE02:  MOV    A,B        ;A <- C
        XRL    A,#2H
        JNZ    JEVE03
        MOV    A,#2H
        LJMP   JEVETD
JEVE03:  MOV    A,R4
        MOV    R1,A
        MOV    A,R6
        MOV    R2,A
        LCALL   JEVETSA
        MOV    B,A
        XRL    A,#1H
        JNZ    JEVE04
        MOV    A,#1H
        LJMP   JEVETD
JEVE04:  MOV    A,B
        XRL    A,#2H
        JNZ    JEVE05
        MOV    A,#2H
JEVE05:  NOP
JEVETD:  RET
;-----------------------------------------------------------------------
;        JEVETSA
;-----------------------------------------------------------------------
JEVETSA:NOP            ;POROVNA R1 R2
        MOV    A,R1
        XRL    A,#0H
        JNZ    JEVEP
        MOV    A,R2
        XRL    A,#0H
        JNZ    JEVEP
        MOV    A,#0H
        LJMP   JEVEPR
JEVEP:  MOV    A,R1
        XRL    A,#0H
        JNZ    VEZN1
        MOV    A,#2H
        LJMP   JEVEPR
VEZN1:  MOV    A,R2
        XRL    A,#0H
        JNZ    VEZN
        MOV    A,#1H
        LJMP   JEVEPR
VEZN:   DEC    R1
        DEC    R2
        LJMP   JEVETSA
JEVEPR:  RET
;-----------------------------------------------------------------------
;         VYDEL
;-----------------------------------------------------------------------
VYDEL:  NOP            ;(R0 R1)/R2 -> (R0 R1)
        MOV    R3,#0
        MOV    R7,#16
POSUV:  MOV    A,R1
        ADD    A,ACC
```

```
        MOV     R1,A
        MOV     A,R0
        ADDC    A,ACC
        MOV     R0,A
        MOV     A,R3
        ADDC    A,ACC
        MOV     R3,A
        MOV     F0,C
        SUBB    A,R2
        JB      F0,NOVY
        JC      MENSI
NOVY:   MOV     R2,A
        INC     R1
MENSI:  DJNZ    R7,POSUV
        RET
;-----------------------------------------------------------------------
;               P R I R A D
;-----------------------------------------------------------------------
PRIRAD:  NOP                    ;V ACC CISLO SERVA
        MOV     R6,A
        MOV     POMOC2,A
        LCALL   NEWLINE
        MOV     R1,#HIGH(TPR1)          ;SERVO
        MOV     R2,#LOW(TPR1)
        LCALL   TISK
        MOV     DPH,#HIGH(VYHEZ)        ;1-8
        MOV     DPL,#LOW(VYHEZ)
        MOV     A,R6
        INC     A
        MOVC    A,@A+DPTR
        LCALL   VYSLI
        MOV     R1,#HIGH(TPR3)          ;' - '
        MOV     R2,#LOW(TPR3)
        LCALL   TISK
        MOV     A,#'T'                  ;TYP
        LCALL   VYSLI
        MOV     A,#'Y'
        LCALL   VYSLI
        MOV     A,#'P'
        LCALL   VYSLI
        MOV     A,#TYP1
        ADD     A,R6
        MOV     R1,A
        MOV     A,@R1
        MOV     DPH,#HIGH(VYHEZ)
        MOV     DPL,#LOW(VYHEZ)
        MOVC    A,@A+DPTR
        LCALL   VYSLI
        NOP                     ;ON NEBO OFF DLE ZAP*
        MOV     A,R6
        MOV     R2,ZAPY
        MOV     R1,A
PRDAZ:  MOV     A,R1
        JZ      PRDAL
        MOV     A,R2
        RR      A
        MOV     R2,A
        LJMP    PRDAZ
PRDAL:  MOV     A,R2
        JB      ACC.0,PRDON
        MOV     R1,#HIGH(TPR5)          ;OFF
        MOV     R2,#LOW(TPR5)
        LCALL   TISK
        LJMP    PRDOF
PRDON:  MOV     R1,#HIGH(TPR7)          ;ON
        MOV     R2,#LOW(TPR7)
        LCALL   TISK
PRDOF:  NOP                     ;POCET ss_____
        MOV     A,R6
        MOV     B,#2H
        MUL     AB
        ADD     A,#PULSY
        MOV     R1,A
        MOV     A,@R1
        MOV     R2,A
        INC     R1
        MOV     A,@R1
        MOV     R1,A
        MOV     DPH,R2
        MOV     DPL,R1
        MOV     A,DPH
        MOV     R1,A
        MOV     A,DPL
```

```
        MOV     R2,A
        MOV     70H,R1
        MOV     71H,R2
        MOV     R1,70H
        MOV     R2,71H
        LCALL   XADE
        MOV     A,R2
        MOV     R6,A
        MOV     A,R1
        MOV     R5,A
        MOV     A,R5
        SWAP    A
        ANL     A,#0FH
        MOV     DPH,#HIGH(VYHEZ)
        MOV     DPL,#LOW(VYHEZ)
        MOVC    A,@A+DPTR
        LCALL   VYSLI
        MOV     A,R5
        ANL     A,#0FH
        MOV     DPH,#HIGH(VYHEZ)
        MOV     DPL,#LOW(VYHEZ)
        MOVC    A,@A+DPTR
        LCALL   VYSLI
        MOV     A,R6
        SWAP    A
        ANL     A,#0FH
        MOV     DPH,#HIGH(VYHEZ)
        MOV     DPL,#LOW(VYHEZ)
        MOVC    A,@A+DPTR
        LCALL   VYSLI
        MOV     A,R6
        ANL     A,#0FH
        MOV     DPH,#HIGH(VYHEZ)
        MOV     DPL,#LOW(VYHEZ)
        MOVC    A,@A+DPTR
        LCALL   VYSLI
        MOV     R1,#HIGH(TPR9)
        MOV     R2,#LOW(TPR9)
        MOV     A,POMOC2
        LCALL   KERTYP
        DEC     A
        LCALL   VYTYHO          ;VYBRAL TYPOVE HODNOTY
        MOV     A,R1
        MOV     60H,A           ;STRED H
        MOV     A,R2
        MOV     61H,A           ;STRED L
        MOV     A,R3
        MOV     62H,A           ;TOLER H
        MOV     A,R4
        MOV     63H,A           ;TOLER L
        MOV     A,POMOC2        ;DO R3 R4 VYBERE PULS
        MOV     B,#2H
        MUL     AB
        ADD     A,#PULS1H
        MOV     R1,A
        MOV     A,@R1
        MOV     R3,A
        MOV     70H,A
        INC     R1
        MOV     A,@R1
        MOV     R4,A
        MOV     71H,A
        LCALL   TUXAM
        MOV     A,70H
        MOV     R3,A
        MOV     A,71H
        MOV     R4,A
        MOV     A,60H           ;PRO00%
        XRL     A,R3
        JNZ     NUNU
        MOV     A,61H
        XRL     A,R4
        JNZ     NUNU
        MOV     R2,#0H
        MOV     A,#'
        LCALL   VYSLI
        LJMP    VEN1
NUNU:   NOP
        MOV     A,60H
        MOV     R1,A
        MOV     A,61H
        MOV     R2,A
        MOV     A,70H
```

```
        MOV     R1,A
        MOV     A,71H
        MOV     R2,A
        MOV     A,60H
        MOV     R5,A
        MOV     A,61H
        MOV     R6,A
        LCALL   JEVETSI         ;JEVETSI_____
        XRL     A,#1H
        JZ      WB01
        MOV     A,#'N'
        MOV     70H,R3
        MOV     71H,R4
        LCALL   TUXAM
        MOV     R3,70H
        MOV     R4,71H
        MOV     70H,60H
        MOV     71H,61H
        MOV     A,#'-'
        LCALL   VYSLI
        LJMP    SECE
        LJMP    WB02
WB01:   NOP             ;PULS-STRED - POMOC1
        MOV     A,#'+'
        LCALL   VYSLI
        MOV     A,60H
        MOV     70H,A
        MOV     A,61H
        MOV     71H,A
        LCALL   TUXAM
SECE:   MOV     A,71H
        ADD     A,R4
        MOV     71H,A
        MOV     A,70H
        ADDC    A,R3
        MOV     70H,A           ;DO 70 71 SECET ~PULS-STRED
        MOV     A,70H
        MOV     R0,A
        MOV     A,71H
        MOV     R1,A
        MOV     R2,#100
        LCALL   NASOB
        MOV     A,R3
        MOV     64H,A
        MOV     A,R4
        MOV     65H,A
        MOV     A,R5
        MOV     66H,A
        MOV     60H,#0H
        MOV     67H,#0H
        MOV     68H,#0H
        MOV     69H,#0H
QAZZ:   LCALL   JEVELO
        XRL     A,#1H
        JNZ     QAZ#1
        LJMP    QAZ#2
QAZ#1:  INC     60H
        MOV     A,69H
        ADD     A,63H
        MOV     69H,A
        MOV     A,68H
        ADDC    A,62H
        MOV     68H,A
        MOV     A,67H
        ADDC    A,#0H
        MOV     67H,A
        LJMP    QAZZ
QAZ#2:  NOP
        MOV     A,60H
        MOV     69H,A
        MOV     A,69H
        MOV     R2,A
        MOV     R1,#0H
VEN:    NOP             ;V R2 JSOU PROCENTA V HEXA
        DEC     R2
VEN1:   MOV     R1,#0H
        LCALL   XADE
        MOV     A,R1
        XRL     A,#1H
        JNZ     UFUF
        MOV     A,#'1'
        LCALL   VYSLI
UFUF:   NOP     ;V R2 JSOU PROCENTA DEC
```

```
        MOV    A,R2
        SWAP   A
        ANL    A,#0FH
        MOV    DPH,#HIGH(VYHEZ)
        MOV    DPL,#LOW(VYHEZ)
        MOVC   A,@A+DPTR
        LCALL  VYSLI
        MOV    A,R2
        ANL    A,#0FH
        MOV    DPH,#HIGH(VYHEZ)
        MOV    DPL,#LOW(VYHEZ)
        MOVC   A,@A+DPTR
        LCALL  VYSLI
WB02: MOV    A,#'%'
        LCALL  VYSLI
        MOV    A,#' '
        LCALL  VYSLI
        MOV    A,R1
        XRL    A,#1H
        JZ     UFUF2
        MOV    A,#' '
        LCALL  VYSLI
UFUF2:  MOV    A,#'*'
        LCALL  VYSLI
        RET
;-----------------------------------------------------------------------
;          JEVELO
;-----------------------------------------------------------------------
JEVELO: NOP    ;JE VETSI LONG 67 68 69 > 64 65 66
        MOV    R1,67H
        MOV    R2,64H
        LCALL  JEVETSA
        JZ     JELD1
        LJMP   JELD
JELD1:  MOV    R3,68H
        MOV    R4,69H
        MOV    R5,65H
        MOV    R6,66H
        LCALL  JEVETSI
JELD:   RET
;-----------------------------------------------------------------------
;          S T A V
;-----------------------------------------------------------------------
STAVQ:MOV    R1,#HIGH(H5T)
        MOV    R2,#LOW(H5T)
        LCALL  TISK
        MOV    A,#0H
        LCALL  PRIRAD
        MOV    A,#1H
        LCALL  PRIRAD
        MOV    A,#2H
        LCALL  PRIRAD
        MOV    A,#3H
        LCALL  PRIRAD
        MOV    A,#4H
        LCALL  PRIRAD
        MOV    A,#5H
        LCALL  PRIRAD
        MOV    A,#6H
        LCALL  PRIRAD
        MOV    A,#7H
        LCALL  PRIRAD
        LJMP   ZABIN
        JB     ZAP8,STAVQ1
        MOV    A,#'0'
        LCALL  VYSLI
        LJMP   STAW1
STAVQ1:   MOV    A,#'1'
        LCALL  VYSLI
STAW1:    JB     ZAP1,STAVQ2
        MOV    A,#'0'
        LCALL  VYSLI
        LJMP   STAW2
STAVQ2:   MOV    A,#'1'
        LCALL  VYSLI
STAW2:    JB     ZAP2,STAVQ3
        MOV    A,#'0'
        LCALL  VYSLI
        LJMP   STAW3
STAVQ3: MOV    A,#'1'
        LCALL  VYSLI
STAW3:  JB     ZAP3,STAVQ4
        MOV    A,#'0'
```

```
          LCALL  VYSLI
          LJMP   STAW4
STAVQ4: MOV    A,#'1'
          LCALL  VYSLI
STAW4:  JB     ZAP4,STAVQ5
          MOV    A,#'0'
          LCALL  VYSLI
          LJMP   STAW5
STAVQ5: MOV    A,#'1'
          LCALL  VYSLI
STAW5:  JB     ZAP5,STAVQ6
          MOV    A,#'0'
          LCALL  VYSLI
          LJMP   STAW6
STAVQ6: MOV    A,#'1'
          LCALL  VYSLI
STAW6:  JB     ZAP6,STAVQ7
          MOV    A,#'0'
          LCALL  VYSLI
          LJMP   STAW7
STAVQ7: MOV    A,#'1'
          LCALL  VYSLI
STAW7:  JB     ZAP7,STAVQ8
          MOV    A,#'0'
          LCALL  VYSLI
          LJMP   STAW8
STAVQ8: MOV    A,#'1'
          LCALL  VYSLI
STAW8:  NOP
ZABIN:    NOP
          LCALL    NEWLINE
          MOV    R1,#HIGH(RADEKH)
          MOV    R2,#LOW(RADEKH)
          LCALL    TISK
          RET
H5T:    DW     H5D-H5T-1
          DB     13,10,'*****************************************'
          DB     13,10,'*    Aktualni nastaveni polohy:                    *'
          DB     13,10,'*    ------------------------                      *'
H5D:    NOP
;-----------------------------------------------------------------------------
;          P O R O
;-----------------------------------------------------------------------------
PORO:     NOP              ;POROVNA BUFFER S KONSTANTOU NA R1,R2
          MOV    DPH,R1
          MOV    DPL,R2
          MOV    R1,#70H          ;UKAZATEL V BUFERU
          MOV    R4,#1          ;1-SHODNE 0-ROZDILNE
          MOV    R5,#0          ;UKAZATEL OD ZACATKU BUFERU
PORO02:     MOV    A,R5
          MOVC    A,@A+DPTR
          MOV    R2,A
          XRL    A,@R1
          JZ     PORO01
          MOV    A,@R1          ;V pameti muze byt hvezdicka
          XRL    A,#'*'
          JZ     PORO01
          MOV    A,R2
          XRL    A,#'*'
          JZ     PORO01
          MOV    R4,#0
PORO01:     INC    R1
          INC    R5
          MOV    A,R1
          XRL    A,#78H
          JZ     POROEND
          LJMP   PORO02
POROEND:MOV    A,R4
          RET
;-----------------------------------------------------------------------------
;          S C A N
;-----------------------------------------------------------------------------
SCAN:     NOP
          MOV    R1,#70H          ;Adresa buferu pro klavesnici
SCAN07:     CLR    RI          ;Ceka na stisk klavesy
          SETB   REN
SCAN1:    JBC    RI,SCAN2
          LJMP   SCAN1
SCAN2:    MOV    A,SBUF
          MOV    R2,A
          XRL    A,#13
          JZ     SCAN03
          MOV    A,R1
```

```
        XRL   A,#'AH
        JZ    SCAN03
        LJMP  SCAN04
SCAN03: LJMP  ENDSCAN
SCAN04: NOP
        MOV   A,R2
        MOV   @R1,A
        MOV   SBUF,A        ;Vysle ECHO
SCAN05: JBC   TI,SCAN06
        LJMP  SCAN05
SCAN06: INC   R1
        LJMP  SCAN07
ENDSCAN:NOP
        RET
;-----------------------------------------------------------------
;        P R O M P T
;-----------------------------------------------------------------
PROMPT:   MOV   A,#'S'
        LCALL   VYSLI
        MOV   A,#'C'
        LCALL   VYSLI
        MOV   A,#':'
        LCALL   VYSLI
        MOV   A,#'>'
        LCALL   VYSLI
        RET
;-----------------------------------------------------------------
;        N E W L I N E
;-----------------------------------------------------------------
NEWLINE:MOV   A,#13
        LCALL   VYSLI
        MOV   A,#10
        LCALL   VYSLI
        RET
;-----------------------------------------------------------------
;        V Y S L I
;-----------------------------------------------------------------
VYSLI:  NOP
        MOV   SBUF,A
VYSLI1: JBC   TI,VYSLI2
        LJMP  VYSLI1
VYSLI2: RET
;-----------------------------------------------------------------
;        U K A B U F
;-----------------------------------------------------------------
UKABUF:   NOP
        MOV   A,#'>'
        LCALL   VYSLI
        MOV   R1,#70H
        MOV   R2,#'AH
UKABUF2:MOV   A,@R1
        LCALL   VYSLI
        INC   R1
        MOV   A,R1
        XRL   A,R2
        JZ    UKABUF1
        LJMP  UKABUF2
UKABUF1:MOV   A,#'<'
        LCALL   VYSLI
        RET
;-----------------------------------------------------------------
;        C L R B U F
;-----------------------------------------------------------------
CLRBUF:   MOV   A,#' '
        MOV   70HA
        MOV   71HA
        MOV   72HA
        MOV   73HA
        MOV   74HA
        MOV   75HA
        MOV   76HA
        MOV   77HA
        MOV   78HA
        MOV   79HA
        MOV   7AHA
        RET
;-----------------------------------------------------------------
;        P R E R U S E N I
;-----------------------------------------------------------------
OWLI: MOV   26HA
        MOV   A,MINKAN
        XRL   A,#8H
        JNZ   INI2
```

```
        MOV    MINKAN,#0H
INI2: CLR    TR0
        MOV    P1,#0H
KANAL0:   NOP    ;PRO KANAL0
        MOV    A,MINKAN
        XRL    A,#0H
        JNZ    KAEND0        ;NENI-LI MINKAN TENTO GO NA KONEC TOHOTO KAN
        JB     ZAP0,AOLU0
        CLR    JP1.0
        LJMP   IAEND0        ;NENI-LI ZAPNUT SHODI OUT GO NA KONEC KAN
AOLU0:  JB     NAHORE0,DOLU0
        MOV    TH0,PULS1H
        MOV    TL0,PULS1L
        SETB   NAHORE0
        SETB   JP1.0
        LJMP   ENDOWL
DOLU0:  CLR    JP1.0
        MOV    TH0,MEZERH
        MOV    TL0,MEZERL
        CLR    NAHORE0
IAEND0:   INC    MINKAN
KAEND0:   NOP
KANAL1:   NOP    ;PRO KANAL1
        MOV    A,MINKAN
        XRL    A,#1H
        JNZ    KAEND1        ;NENI-LI MINKAN TENTO GO NA KONEC TOHOTO KAN
        JB     ZAP1,AOLU1
        CLR    JP1.1
        LJMP   IAEND1        ;NENI-LI ZAPNUT SHODI OUT GO NA KONEC KAN
AOLU1:  JB     NAHORE1,DOLU1
        MOV    TH0,PULS2H
        MOV    TL0,PULS2L
        SETB   NAHORE1
        SETB   JP1.1
        LJMP   ENDOWL
DOLU1:  CLR    JP1.1
        MOV    TH0,MEZERH
        MOV    TL0,MEZERL
        CLR    NAHORE1
IAEND1:   INC    MINKAN
KAEND1: NOP
KANAL2:   NOP    ;PRO KANAL2
        MOV    A,MINKAN
        XRL    A,#2H
        JNZ    KAEND2        ;NENI-LI MINKAN TENTO GO NA KONEC TOHOTO KAN
        JB     ZAP2,AOLU2
        CLR    JP1.2
        LJMP   IAEND2        ;NENI-LI ZAPNUT SHODI OUT GO NA KONEC KAN
AOLU2:  JB     NAHORE2,DOLU2
        MOV    TH0,PULS3H
        MOV    TL0,PULS3L
        SETB   NAHORE2
        SETB   JP1.2
        LJMP   ENDOWL
DOLU2:  CLR    JP1.2
        MOV    TH0,MEZERH
        MOV    TL0,MEZERL
        CLR    NAHORE2
IAEND2:   INC    MINKAN
KAEND2: NOP
KANAL3:   NOP    ;PRO KANAL3
        MOV    A,MINKAN
        XRL    A,#3H
        JNZ    KAEND3        ;NENI-LI MINKAN TENTO GO NA KONEC TOHOTO KAN
        JB     ZAP3,AOLU3
        CLR    JP1.3
        LJMP   IAEND3        ;NENI-LI ZAPNUT SHODI OUT GO NA KONEC KAN
AOLU3:  JB     NAHORE3,DOLU3
        MOV    TH0,PULS4H
        MOV    TL0,PULS4L
        SETB   NAHORE3
        SETB   JP1.3
        LJMP   ENDOWL
DOLU3:  CLR    JP1.3
        MOV    TH0,MEZERH
        MOV    TL0,MEZERL
        CLR    NAHORE3
IAEND3:   INC    MINKAN
KAEND3: NOP
KANAL4:   NOP    ;PRO KANAL4
        MOV    A,MINKAN
        XRL    A,#4H
        JNZ    KAEND4        ;NENI-LI MINKAN TENTO GO NA KONEC TOHOTO KAN
```

```
        JB    ZAP4,AOLU4
        CLR   JP1.4
        LJMP  IAEND4        ;NENI LI ZAPNUT SHODI OUT GO NA KONEC KAN
AOLU4:  JB    NAHORE4,DOLU4
        MOV   TH0,PULS5H
        MOV   TL0,PULS5L
        SETB  NAHORE4
        SETB  JP1.4
        LJMP  ENDOWL
DOLU4:  CLR   JP1.4
        MOV   TH0,MEZERH
        MOV   TL0,MEZERL
        CLR   NAHORE4
IAEND4: INC   MINKAN
KAEND4: NOP
KANAL5: NOP   ;PRO KANAL5
        MOV   A,MINKAN
        XRL   A,#5H
        JNZ   KAEND5        ;NENI-LI MINKAN TENTO GO NA KONEC TOHOTO KAN
        JB    ZAP5,AOLU5
        CLR   JP1.5
        LJMP  IAEND5        ;NENI-LI ZAPNUT SHODI OUT GO NA KONEC KAN
AOLU5:  JB    NAHORE5,DOLU5
        MOV   TH0,PULS6H
        MOV   TL0,PULS6L
        SETB  NAHORE5
        SETB  JP1.5
        LJMP  ENDOWL
DOLU5:  CLR   JP1.5
        MOV   TH0,MEZERH
        MOV   TL0,MEZERL
        CLR   NAHORE5
IAEND5: INC   MINKAN
KAEND5: NOP
KANAL6: NOP   ;PRO KANAL6
        MOV   A,MINKAN
        XRL   A,#6H
        JNZ   KAEND6        ;NENI-LI MINKAN TENTO GO NA KONEC TOHOTO KAN
        JB    ZAP6,AOLU6
        CLR   JP1.6
        LJMP  IAEND6        ;NENI-LI ZAPNUT SHODI OUT GO NA KONEC KAN
AOLU6:  JB    NAHORE6,DOLU6
        MOV   TH0,PULS7H
        MOV   TL0,PULS7L
        SETB  NAHORE6
        SETB  JP1.6
        LJMP  ENDOWL
DOLU6:  CLR   JP1.6
        MOV   TH0,MEZERH
        MOV   TL0,MEZERL
        CLR   NAHORE6
IAEND6: INC   MINKAN
KAEND6: NOP
KANAL7: NOP   ;PRO KANAL7
        MOV   A,MINKAN
        XRL   A,#7H
        JNZ   KAEND7        ;NENI-LI MINKAN TENTO GO NA KONEC TOHOTO KAN
        JB    ZAP7,AOLU7
        CLR   JP1.7
        LJMP  IAEND7        ;NENI-LI ZAPNUT SHODI OUT GO NA KONEC KAN
AOLU7:  JB    NAHORE7,DOLU7
        MOV   TH0,PULS8H
        MOV   TL0,PULS8L
        SETB  NAHORE7
        SETB  JP1.7
        LJMP  ENDOWL
DOLU7:  CLR   JP1.7
        MOV   TH0,MEZERH
        MOV   TL0,MEZERL
        CLR   NAHORE7
IAEND7: INC   MINKAN
KAEND7: NOP
ENDOWL: MOV   P1,JP1
        SETB  TR0
        MOV   A,26H
        RETI
;-----------------------------------------------------------------------
;       T E X T Y
;-----------------------------------------------------------------------
TPR1:   DW    TPR2-TPR1-1
        DB    '* SERVO'
TPR2:   NOP
TPR3:   DW    TPR4-TPR3-1
```

```
        DB    '.'
TPR4:   NOP
TPR5:   DW    TPR6-TPR5-1
        DB    'OFF'
TPR6:   NOP
TPR7:   DW    TPR8-TPR7-1
        DB    'ON'
TPR8:   NOP
TPR9:   DW    TPR10-TPR9-1
        DB    'us'
TPR10:  NOP
TPR11:  DW    TPR12-TPR11-1
        DB    'Pozadovana hodnota je mimo povoleny rozsah !'
TPR12:  NOP
RADEKII:  DW    RADEKD-RADEKII-1
        DB    '*****************************************'
RADEKD:   NOP
KHE01T: DW    KHE01D-KHE01T-1
        DB    13,10,'*****************************************'
        DB    13,10,'*          H E L P                      *'
        DB    13,10,'*          -------                      *'
        DB    13,10,'*  H1 - Popis SERVOCONTROLERu            *'
        DB    13,10,'*  H2 - Zapnutí/vypnutí kanálu            *'
        DB    13,10,'*  H3 - Aktuální nastavení  (STAV?)       *'
        DB    13,10,'*  H4 - Nastavení polohy                  *'
        DB    13,10,'*  H5 - Nastavení typu                    *'
        DB    13,10,'*  H6 - Předdefinované typy               *'
        DB    13,10,'*  H7 - * B E R Y *                       *'
        DB    13,10,'*****************************************'
KHE01D: NOP
KHE03T: DW    KHE03D-KHE03T-1
        DB    13,10,'*****************************************'
        DB    13,10,'* Definované typy servomechanismu:        *'
        DB    13,10,'* ------------------------------         *'
        DB    13,10,'* 1 - S28   1310us +- 700us (FUTABA)      *'
        DB    13,10,'* 2 - S128  1520us +- 700us (FUTABA)      *'
        DB    13,10,'* 3 - STI   1230us +- 700us               *'
        DB    13,10,'* 4 -       1310us +- 600us               *'
        DB    13,10,'* 5 -       1310us +- 500us               *'
        DB    13,10,'* 6 -       1310us +- 400us               *'
        DB    13,10,'* 7 -       1310us +- 300us               *'
        DB    13,10,'* 8 -       1520us +- 600us               *'
        DB    13,10,'* 9 -       1520us +- 500us               *'
        DB    13,10,'* A -       1520us +- 400us               *'
        DB    13,10,'* B -       1520us +- 300us               *'
        DB    13,10,'* C -       1230us +- 600us               *'
        DB    13,10,'* D -       1230us +- 500us               *'
        DB    13,10,'* E -       1230us +- 400us               *'
        DB    13,10,'* F -       1230us +- 300us               *'
        DB    13,10,'*****************************************'
KHE03D: NOP
KHE04T: DW    KHE04D-KHE04T-1
        DB    13,10,'*****************************************'
        DB    13,10,'*    Zapnutí/Vypnutí kanálu               *'
        DB    13,10,'*    -----------------------              *'
        DB    13,10,'* Zapnutí kanálu:  S1:ON                   *'
        DB    13,10,'*                  |                      *'
        DB    13,10,'*                  Číslo kanálu           *'
        DB    13,10,'*                  |                      *'
        DB    13,10,'* Vypnutí kanálu:  S1:OFF                  *'
        DB    13,10,'* Zapnutí všech kanálu:  SV:ON             *'
        DB    13,10,'* Podobně vypnutí kanálu:  SV:OFF          *'
        DB    13,10,'*****************************************'
KHE04D: NOP
KHE05T: DW    KHE05D-KHE05T-1
        DB    13,10,'*****************************************'
        DB    13,10,'*    Nastavení polohy kanálu              *'
        DB    13,10,'*    ------------------------             *'
        DB    13,10,'* a) Přímo hodnotou                        *'
        DB    13,10,'*  S1:1310D (Hodnota v us DECIMALNÍ)       *'
        DB    13,10,'*  S2:04CEH (Hodnota v us HEXA)            *'
        DB    13,10,'* b) V procentech                          *'
        DB    13,10,'*  S1:+25% , S1:-50% , atd ...             *'
        DB    13,10,'*  Hodnota udávejte procenta mezi          *'
        DB    13,10,'*  začínat znaménkem.                      *'
        DB    13,10,'*  0 odpovídá střední hodnotě              *'
        DB    13,10,'*****************************************'
KHE05D: NOP
KHE06T: DW    KHE06D-KHE06T-1
        DB    13,10,'*****************************************'
        DB    13,10,'*    Nastavení typu kanálu                *'
        DB    13,10,'*    ---------------------                *'
        DB    13,10,'* S1:TYP1  Udává typ serva                *'
```

```
        DB    13,10,'*          (kanal 1 typ 1)                    *'
        DB    13,10,'*    Z chla typu jsou stanoveny               *'
        DB    13,10,'*    hodnoty pro stredni polohu               *'
        DB    13,10,'*    a pro min a max hodnotu.                 *'
        DB    13,10,'*    Je mozno volit 15 typu  (1-F)            *'
        DB    13,10,'*****************************************'
KHE06D: NOP
KHE07T: DW    KHE07D-KHE07T-1
        DB    13,10,'*****************************************'
        DB    13,10,'*      Popis SERVOCONTROLERU              *'
        DB    13,10,'*      ---------------------              *'
        DB    13,10,'* SC je tvoren uP 8031.SC' ovlada az        *'
        DB    13,10,'* 8 servomechanismu.S PC komunikuje        *'
        DB    13,10,'* pres seriovy port.                       *'
        DB    13,10,'*****************************************'
KHE07D: NOP
;------------------------ DEFINICE TYPU
;       STREDNI HODNOTA  TOLERANCE
DEFTYP:   DB    05H,1EH,02H,0BCH    ;1310,700    FUTABA    FP-S28
        DB    05H,0F0H,02H,0BCH    ;1520,700    FUTABA    FP-S128
        DB    04H,0CEH,02H,0BCH    ;1230,700
        DB    05H,1EH,02H,0BCH
        DB    05H,1EH,02H,0BCH
        DB    05H,01EH,02H,0BCH
        DB    05H,01EH,02H,0BCH
        DB    05H,01EH,02H,0BCH
        DB    05H,01EH,02H,0BCH
        DB    05H,01EH,02H,0BCH
        DB    05H,01EH,02H,0BCH
        DB    05H,01EH,02H,0BCH
        DB    05H,01EH,02H,0BCH
        DB    05H,01EH,02H,0BCH
        DB    05H,01EH,02H,0BCH
        DB    05H,01EH,02H,0BCH
        DB    05H,01EH,02H,0BCH
        DB    05H,01EH,02H,0BCH
        ;------------------------ DEFINICE PRIKAZU
        DCPR0:    DB    'S*:0%    '
        DCPR1:    DB    'S*:+**%  '
        DCPR2:    DB    'S*:+***% '
        DCPR3:    DB    'S*:-**%  '
        DCPR4:    DB    'S*:-***% '
        DCPR5:    DB    'S*:***D  '
        DCPR6:    DB    'SV:ON    '
        DCPR7:    DB    'SV:OFF   '
        DCH1:  DB    'H1    '
        DCH2:  DB    'H2    '
        DCH3:  DB    'H3    '
        DCH4:  DB    'H4    '
        DCH5:  DB    'H5    '
        DCH6:  DB    'H6    '
        DCH7:  DB    'H7    '
        DCPS:  DB    'S*:+100% '
        DCZS:  DB    'S*:-100% '
        BEEP:  DB    'BEEP    '
        DPOK:    DW    0000
;------------------------ K O N E C ------------------------------------------
        ENT:    END
```

Futaba

FP-S128 for J Series
FP-S28 for FN. G. L Series

Very precise neutral, small, rugged servo

These very precise neutral, small, rugged servos are designed for use with Futaba digital proportional radio control sets. The FP-S128 is for the J Series and the FP-S28 is for the FN, G and L Series. Use them for channel expansion, or as replacement parts.

[FEATURES]

- New indirect drive/completely sealed potentiometer substantially improves vibration and shock resistance, and neutral accuracy.
- Unique Futaba power-saving custom IC provides high starting torque narrow dead band, and excellent trackability.
- Fiberglass PBT (polybutylene terephthalate) servo case is mechanically strong and is invulnerable to glow fuel.
- Strong polyacetyl resin precision servo gear featuring smooth operation, accurate neutral, and minimal backlash.
- Fiberglass epoxy PC board with THRU-THE-HOLE plating improves the servo amp vibration and shock resistance.
- Thick film gold plated connector pins eliminate the problem of faulty contact, improve reliability against shock and vibration, and prevent reverse insertion.
- Special pad grommets simplify mounting of the servo, and are extremely vibration-resistant.
- Four kind of special adjustable horn are available.
- High 48.7oz-in (3.5 kg-cm) output torque is perfect for almost all models.

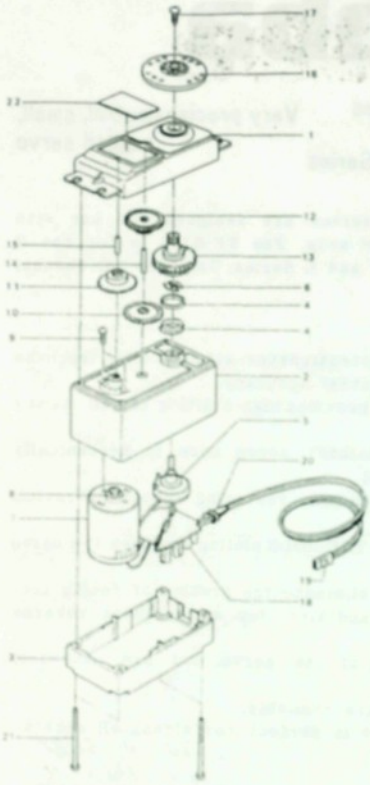
[RATINGS]

Contol system	+pulse control 1520μs neutral-S128 / 1310μs neutral-S28
Operating angle	Rotary system, one side 45° or greater (including trim)
Power supply	4.8V or 6.0V (shared with receiver)
Power consumption	6.0V, 8mA (at idle)
Output torque	48.7oz/in (3.5kg·cm)
Operating speed	0.24sec/60°
Dimensions	1.59×0.79×1.59inch (40.5×20×40.5㎜)
Weight	1.87oz (53g)

[USAGE PRECAUTIONS]

- Use the FP-S128 with a J series digital proportional set.
- Use FP-S28 with an FN, G, or L Series digital proporional set. Futaba servos can not be used with other makes of RC sets.
- Select the servo mounting position and install the rod adjuster, flexible wire, hinges, etc. so each control operates smoothly.
- When installing the servo with screws, use the screws through grommets and tighten the screws so the grommets is slightly crushed. Be especially careful when the steering angle is large. If the screws are too tight, the vibrationproofing effect will be lost.
- Spare horns are provided. Use them according to the application.

(Specifications are subject to chang without prior notice.)



№	Part name	Part №
1	Upper case	FCS-28
2	Middle case	
3	Bottom case	
4	Metal bearing	S 04134
5	Potentiometer	i 39995
6	Potentiomer drive plate	S 02753
7	Motor	S 91212
8	Motor pinion	S 02461
9	Screw 2×3	J 50002
10	1st gear	FGS-28
11	2nd gear	FGS-28
12	3rd gear	FGS-28
13	Final gear	FGS-28
14	Intermediate shaft	S 02495
15	2nd shaft	S 02494
16	Servo horn D	FSH-6W
17	Binding head tapping screw 2.6×8	FSH-41
18	Printed wirina board	AS1202
19	S128……3PB-WRB300	FPC-8M
	S 28……with 3PC cord	FPC-3M
20	w/gum bush	S 90045
21	Pan head truss screw	J 50400
22	Nameplate	S 80700

SPLINED SERVO HORNS

horn A
(FSH-6X)

horn B
(FSH-6S)

horn C
(FSH-6R)

horn D
(FSH-6W)

FUTABA CORPORATION OF AMERICA

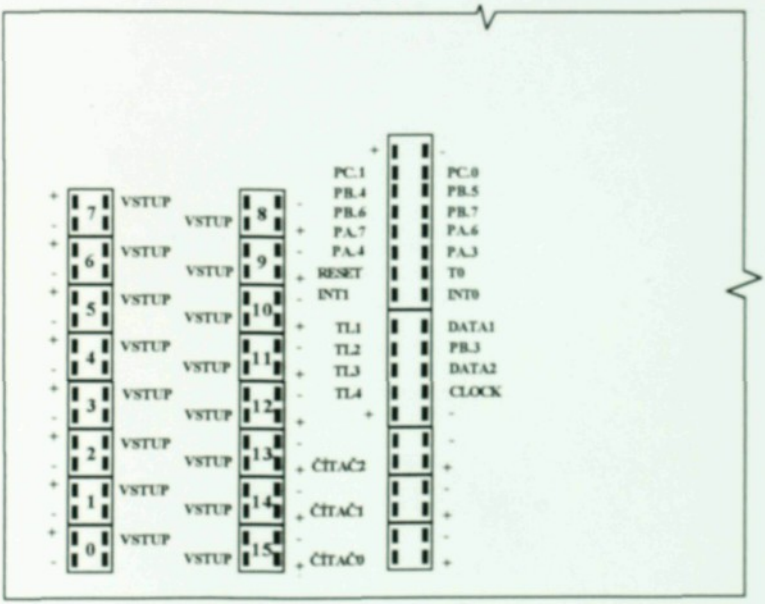
555 West Victoria Street Compton, California 90220 U. S. A.
Phone : (213) 537－9610 Telex : 23－0691227 Facsimile : 213－637－8529

FUTABA CORPORATION

Tokyo Office : DAIDO BLDG., 3－1－16, Sotokanda, Chiyoda-ku, Tokyo, Japan.
Phone : (03) 255－5881 Telex : J26532

```
char      i;
char      cislo1;
char      cislo2;
char      cislo3;
char      cislo4;
char      zn;
char      *retezec;

void      tiskc(a)
     char       a;
     {
     itoa(a,retezec);
     puts(retezec);
     }
void      tiskz(char *s)
     {while(*s)putchar(*s++);}
main()
{
puts("*******************************************");
puts("*        PROGRAM PRO TESTOVANI            *");
puts("*        PREKLADACE JAZYKA C              *");
puts("*        stiskni klavesu                  *");
puts("*******************************************");
cislo1=0;
cislo2=0;
cislo3=0;
cislo4=0;
getchar(zn);
for(i=1;i<101;i++)
     {
     cislo1=i+1;
     cislo2=i-1;
     cislo3=i*2;
     cislo4=i/2;
     tiskz("TISK PRO I=");
     tiskc(i);
     tiskz("I+1=");
     tiskc(cislo1);
     tiskz("I-1=");
     tiskc(cislo2);
     tiskz("I*2=");
     tiskc(cislo3);
     tiskz("I/2=");
     tiskc(cislo4);
     };
puts("\n----- KONEC TESTU -----");
kon:;
goto kon;
}
/*END*/
```

```
10      TITLE
20      TITLE      ******************************************
30      TITLE      *        PROGRAM PRO TESTOVANI           *
40      TITLE      *        PREKLADACE BASIC 51             *
50      TITLE      *        stiskni klavesu                 *
60      TITLE      ******************************************
70      READ      ZNAK
100      CISLO1 = 0
110      CISLO2 = 0
120      CISLO3 = 0
130      CISLO4 = 0
200      FOR      I = 1 TO 100
210      TITLE      TISK PRO I=;
220      RCISLO = I
230      GOSUB      25000
300      CISLO1 = I+1
310      CISLO2 = I-1
320      CISLO3 = I*2
330      CISLO4 = I/2
400      TITLE      I+1=;
410      RCISLO = CISLO1
420      GOSUB      25000
500      TITLE      I-1=;
510      RCISLO = CISLO2
520      GOSUB      25000
600      TITLE      I*2=;
610      RCISLO = CISLO3
620      GOSUB      25000
700      TITLE      I/2=;
710      RCISLO = CISLO4
720      GOSUB      25000
800      TITLE
1000      NEXT I
2000      TITLE      ----- KONEC TESTU ------
10000      GOTO      10000
25000   REM RUTINY   ------------------------
25002 REM            KONVERZE CISLA NA STRING
25005 REM            A JEHO VYTISK
25010 FOR RI = 0 to 2
25020 MEM[2:RI] = RCISLO MOD 10 + 48
25030 RCISLO = RCISLO / 10
25040 NEXT RI
25045 REM             VYTISK
25050 FOR RI = 0 to 2
25060 PRINT MEM[2:2-RI];
25070 NEXT RI
25075 TITLE
25080 RETURN
25100 END
```

Zapojení konektorů na základové desce.