

Středoškolská odborná činnost 2007/2008

Obor 10 – elektrotechnika, elektronika a telekomunikace

Model výtahu

Autor:
Martin Kresta
SPŠ elektrotechniky a informatiky,
Kratochvílova 7
702 00 Ostrava, 4.ročník

Konzultant práce:
Ing. Ladislav Škapa
(SPŠEI, Ostrava)

Prohlášení

Prohlašuji tímto, že jsem soutěžní práci vypracoval samostatně a v seznamu literatury jsem uvedl veškerou použitou literaturu a další informační zdroje včetně internetu.

V Ostrave dne:.....

.....
Martin Kresta

Poděkování

Úvodem bych chtěl poděkovat těmto lidem, bez jejichž ochotné pomoci by byla práce výrazně složitější:

- Ing. Ladislavu Škapovi, vedoucímu i konzultantovi mé práce, který mi poskytl několik zajímavých tipů v oblasti softwaru.
- Ing. Zdeňkovi Nálevkovi za trpělivé vysvětlení problematiky elektrických pohonů zejména PFM a užitečné rady v této oblasti.
- Petru Stehlíkovi a firmě ZAM-servis s.r.o. za poskytnutí schématu a softwaru pro výrobu ISP programátoru, který byl použit při vývoji modelu výtahu.
- Jiřímu Starému za soustružnické práce při výrobě ocelového hřídele a ložisek.

OBSAH

Prohlášení.....	2
Poděkování.....	3
1. CHARAKTERISTIKA TÉMATU	6
1.1. Zdůvodnění tématu práce.....	6
1.2. Specifikace problému	6
1.3. Použité metody.....	7
1.4. Organizace práce.....	7
2. POPIS FUNKCE	8
3. NÁVOD K OBSLUZE	9
4. KONSTRUKCE	10
4.1. Výtahová šachta.....	10
4.2. Pohonná jednotka.....	10
4.3. Kabina a protizávaží	11
4.4. Stabilita	11
5. ELEKTRONIKA.....	12
5.1. Zdroje.....	13
5.2. Elektronická pojistka	13
5.3. Napěťový supervisor.....	14
5.4. Elektrický pohon.....	14
5.5. Displej	15
5.6. Tlačítka	15
5.7. LED	15
5.8. Elektronika v kabině	15
6. SOFTWARE.....	16
6.1. Atmel.....	16
6.2. Program.....	16
6.3. MASTER	17
6.3.1. Nastavení	17
6.3.2. Test čidel	18
6.3.3. Test tlačítek	18
6.3.4. Vyhodnocení požadavku	18
6.3.5. Nástup/výstup.....	18
6.3.6. Kontrola paměti.....	19
6.3.7. Odeslání dat.....	19
6.3.8. Plynulý rozjezd.....	19
6.3.9. Plynulé zastavení.....	20
6.3.10. Subrutina on a off.....	20
6.3.11. Zobrazení.....	20
6.3.12. Subrutina ledka.....	20
6.3.13. Časování 4 sekundy.....	20
6.3.14. Subrutina savetest.....	21
6.3.15. Obsluha přerušení Timeru 0.....	21
6.3.16. Obsluha přerušení sériové linky.....	21

6.4. SLAVE.....	22
6.4.1. Testování tlačítek	22
6.4.2. Odeslání dat.....	22
6.4.3. Subrutina antikmit	22
6.4.4. Obsluha přerušení sériové linky	23
6.4.5. Subrutina dekoder	23
6.4.6. Generování tónů	23
6.4.7. Obsluha přerušení timeru 0	23
7. ZÁVĚR.....	24
8. LITERATURA.....	24
9. PŘÍLOHY	25
9.1. Blokové schéma elektroniky.....	25
9.2. Elektrotechnická schémata	26
9.2.1. MASTER.....	26
9.2.2. SLAVE	27
9.2.3. Výpis součástek.....	28
9.2.4. Zdroj	29
9.2.5. Control 1-6	30
9.2.6. Sensor 1-6.....	31
9.3. Vývojové diagramy.....	32
9.3.1. Vývojový diagram MASTER.....	32
9.3.2. Rutina sentest	33
9.3.3. Rutina zobraz.....	34
9.3.4. Rutina posli (MASTER).....	35
9.3.5. Rutina butttest (MASTER).....	36
9.3.6. Rutina lift.....	37
9.3.7. Rutina ready	38
9.3.8. Subrutina jed	39
9.3.9. Subrutina stuj.....	40
9.3.10. Subrutiny on a off.....	41
9.3.11. Subrutina ledka.....	42
9.3.12. Subrutina wait4	43
9.3.13. Subrutina savetest – 1.část	44
9.3.14. Subrutina savetest – 2.část	45
9.3.15. Obsluha přerušení sériové linky (MASTER).....	46
9.3.16. Obsluha přerušení timeru 0 (MASTER)	47
9.3.17. Rutina Butttest (SLAVE)	48
9.3.18. Subrutina antikmit	49
9.3.19. Subrutina posli (SLAVE)	50
9.3.20. Obsluha přerušení sériové linky (SLAVE)	51
9.3.21. Subrutina dekoder	52
9.3.22. Subrutiny ah a d.....	53
9.3.23. Obsluha přerušení timeru 0 (SLAVE).....	54

1. CHARAKTERISTIKA TÉMATU

1.1. Zdůvodnění tématu práce

Každoročně na naší škole probíhá SOČ. V letošním školním roce jsem se se svým tématem přihlásil i já. Zajímá mě využití automatizační techniky v praxi, zvláště pak mikroprocesorová technika. Rozhodl jsem se setrojit funkční model výtahu. Takovýto model jsem si vybral proto, že je vhodný pro řízení mikroprocesorem a umožňuje s využitím vhodných periférií demonstrovat možnosti mikroprocesoru. Přestože tato práce bude reprezentovat předmět mikroprocesorová technika, dá se říct, že jde o propojení předmětů mikroprocesorová technika, elektronika, elektrotechnická měření, programové vybavení, fyzika, matematika a další. Komplexnost práce je také jeden z důvodů proč jsem si toto téma vybral. Hlavním důvodem ale zůstává možnost vyzkoušet si v praxi programování mikroprocesoru pomocí jazyku symbolických adres a další praktické uplatnění teoretických znalostí.

V současnosti ještě funguje mnoho starších výtahů, které se modernizují tak, že se starý reléový systém nahradí novým modernějším, jehož základ tvoří mikroprocesor. To mimo jiné umožňuje jednodušší realizaci plynulého pohybu, zvyšuje se tak efektivita, celkově bezpečnost a spolehlivost systému. Toto moderní řešení je využito i pro realizaci modelu výtahu, který je tímto po stránce řízení srovnatelný s běžně používanými výtahy.

Model jako takový může sloužit jako prezentační nebo demonstrační model a díky podpoře ISP programování i jako výukový model předmětu mikroprocesorová technika.

1.2. Specifikace problému

Model řeší problém řídicího softwaru a elektroniky pro řízení osobního výtahu. Z hlediska softwaru je to především správná algoritmizace, ošetření nežádoucích elektro-mechanických jevů, realizace řízení motoru a generování tónu. Z hlediska elektroniky pak vstupní a výstupní periférie a jejich přizpůsobení pro řízení digitálními signály s nízkou proudovou zatížitelností, obvod pro spínání motoru a stejnosměrné stabilizované zdroje pro motor a pro ostatní elektroniku včetně mikroprocesoru. Pro předvedení funkčnosti tohoto systému byl zhotoven reálný model se 6-ti stanicemi přizpůsobený pro implementaci elektroniky. Celek funkčního modelu výtahu je tedy tvořen propojením konstrukční, elektronické a softwarové části.

Běžně používané osobní výtahy musejí být navíc vybaveny různými mechanickými bezpečnostními prvky. U navrhovaného modelu však vzhledem k omezeným konstrukčním možnostem nejsou tyto prvky na modelu realizovány. Model slouží pouze k demonstraci softwaru a elektroniky.

1.3. Použité metody

Při realizaci bylo použito několik zajímavých softwarových i elektronických metod. Například použití integrovaných stabilizátorů pro stejnosměrné zdroje, elektronická pojistka řízená procesorem pro odpojení napájení motoru při přetížení, napěťový supervisor pro vygenerování resetu pro mikroprocesory při poklesu napájecího napětí, řízení dvou LED pomocí jednoho pinu mikroprocesoru s použitím komplementárního zapojení tranzistorů, použití speciálního obvodu pro řízení motoru pomocí pulsně frekvenční modulace a optické závory.

Z hlediska softwaru bylo využito téměř všech možností použitého mikroprocesoru. Jsou to sériová komunikace mezi procesorem v kabině a hlavním procesorem, timer v režimu „baudrate generator“, generování tónu s proměnnou frekvencí pomocí timeru v režimu „autoreload“, softwarová realizace PFM (pulsně frekvenční modulace) pro plynulé rozjezdy a zastavení motoru, čekací smyčky a další.

1.4. Organizace práce

Práce byla započata vývojem jednotlivých funkčních částí elektroniky na nepájivém poli. Následovala výroba plošných spojů a vývoj softwaru. Pro testování bylo nutno fyzicky zhotovit výtahovou šachtu a pohonnou jednotku. Následně byla vyrobena elektronika i software pro procesor v kabině a realizována sériová komunikace. Postupně byl odlaďován software a dokončována mechanická konstrukce šachty výtahu. Zhotovením elektronické pojistky a napěťového supervisoru byl rámcově ukončen proces výroby.

První kapitola hlavní části vypovídá o funkci výtahu, reakcích na požadavky obsluhy, a o principu činnosti. Obsahuje i návod k obsluze.

Druhá kapitola popisuje mechanickou konstrukci výtahové šachty, použitý materiál a výrobní technologie.

Následující kapitola pojednává o návrhu a funkci elektroniky, odůvodnění použitých prvků a o návrhu desek plošných spojů.

Poslední kapitola obsahuje podrobný popis softwaru, algoritmy a stručný popis mikroprocesoru s jádrem 8051.

2.POPIS FUNKCE

Model je napodobeninou skutečného osobního výtahu včetně principu ovládání. Každé patro (stanice) je vybaveno jednociferným 7-mi segmentovým displejem, červenou a zelenou LED a tlačítkem. V kabině je opět displej se dvěma LED, 6 tlačítek pro výběr patra, tlačítko stop a tlačítko zvonek.

Signalizace LED:

- Svítící červená LED: zaneprázdněnost výtahu, zajištěné dveře.
- Svítící zelená LED: nástup/výstup, otevřené dveře.
- Bez signalizace LED: čeká na požadavek, zajištěné dveře.

Jestliže výtah čeká na požadavek (bez signalizace LED), ihned po zadání požadavku (přivolání výtahu) je požadavek vyhodnocen. V případě, že v době přivolání je již kabina zaneprázdněna, uloží se požadavek do paměti a je vyhodnocen ihned, jakmile je aktuální požadavek vyřešen. Avšak v případě, že v této době nastane nový požadavek obsluhy z kabiny, má tento požadavek vyšší prioritu a je vyhodnocen přednostně. Do paměti lze uložit maximálně 3 požadavky, čtvrtý požadavek je ignorován. Požadavky z paměti jsou vyhodnocovány v tom pořadí, v jakém byly uloženy.

Při pohybu kabiny svítí všechny červené LED, které signalizují zajištěné dveře a na displejích se zobrazuje aktuální pozice kabiny.

Po zastavení kabiny v příslušném patře se rozsvítí zelená a zhasne červená LED v kabině i v daném patře a zazní zvukový signál. Zároveň se rozsvítí displej v kabině, který zobrazuje její aktuální pozici. V případě, že do cca 4 sekund není zadán další požadavek, všechny LED i displej v kabině zhasnou a výtah čeká na požadavek, anebo se začne vyhodnocovat požadavek z paměti, byl-li nějaký uložen.

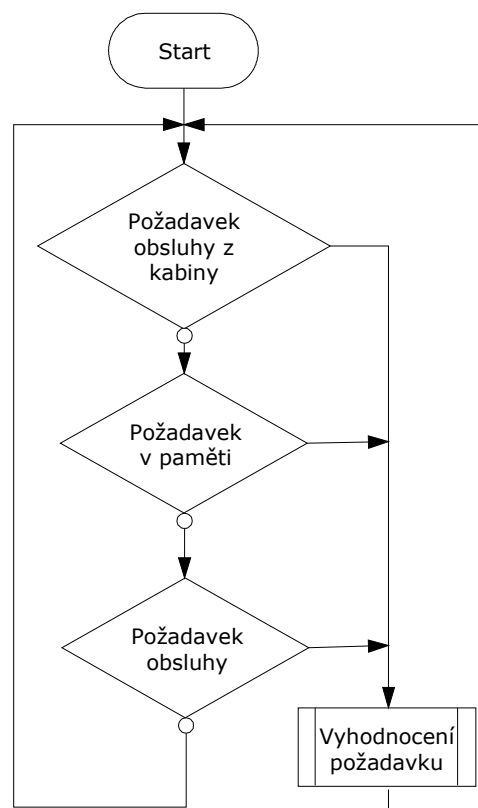
V případě požadavku obsluhy z kabiny se výtah uvede do pohybu okamžitě. Displej v kabině zůstane rozsvícený, svítí všechny červené LED. Po zastavení se celá situace opakuje.

Při stisku tlačítka STOP se kabina okamžitě zastaví a čeká na další požadavek. Zastaví-li takto kabina v kterémkoliv patře, otevřou se dveře. Při zastavení mimo patro zůstanou dveře zajištěny.

Tlačítko ZVONEK spustí generování výstražného tónu bez ohledu na to v jakém stavu či poloze se výtah nachází.

Jediný okamžik, kdy systém nereaguje na zadaný požadavek nastane jen v době zastavování a rozjezdu kabiny (cca 0,6s). To je dáno omezenými možnostmi použitého mikroprocesoru. V době generování tónu dojezdu do patra systém nereaguje na požadavky obsluhy z kabiny (cca 1s).

Pro názornost je funkce výtahu graficky znázorněna vývojovým diagramem.



4.KONSTRUKCE

Návrh konstrukce vyžadoval kompromis mezi funkčností, složitostí výroby a cenou. Základními konstrukčními materiály jsou dřevo, hliník a plexisklo. Veškeré spoje uvedených materiálů byly realizovány pomocí běžně dostupných spojovacích prvků, jako jsou vruty či šrouby, v případě dílů z plexiskla bylo použito speciální lepidlo. Samotná výroba, dělení či úpravy jednotlivých dílů představovalo použití zejména klasických obráběčských metod, jako vrtání, řezání, pilování, dále pak ohýbání, lisování apod. Díly z plexiskla sloužící k uchycení ložisek zhotovila a dodala firma Valter Špalek podle mnou zpracované výkresové dokumentace. Hřídel pro navíjení tažného lana je vyroben z oceli a podle mé výkresové dokumentace ji zhotovil Jiří Starý. Dřevěné díly pro základnu a nosný prvek pro snímače polohy byly po tvarovém opracování povrchově upraveny šedou barvou.

4.1.Výtahová šachta

Základ výtahové šachty tvoří dolní panel, horní panel a 4 L profily. Dolní i horní panel čtvercového půdorysu je zhotoven z překližky o rozměrech 140x140mm a tloušťce 12mm. Panely jsou od sebe vzdáleny 940mm. K těmto panelům jsou v každém rohu pomocí vrutů přišroubovány hliníkové L profily. Tyto profily o rozměru 15x15mm jsou 1000mm dlouhé a mají zlatou povrchovou úpravu. Vzniklá konstrukce tvoří dostatečně tuhý nosný základ pro vybavení ostatními funkčními částmi či díly. Na čelní stěně konstrukce jsou k vnitřním stranám zlatých hliníkových profilů ještě přišroubovány 2 ploché I profily stříbrné barvy stejné délky jako hliníkový profil L. Stříbrný I profil na pravé straně konstrukce slouží jako nosný prvek pro umístění ovládacích prvků. Stříbrný profil na levé straně konstrukce pouze dotváří optickou symetrii čelní stěny. Délka profilu I po pravé straně je rovnoměrně rozdělena do šesti úseků, které představují jednotlivá patra. V úrovni každého patra je do I profilu zabudován displej, tlačítko a 2 LED. Tyto prvky jsou nutné k signalizaci a ovládní kabiny výtahu a jsou zabudovány tak, že tvoří s povrchem profilu jednu rovinu. Vodící lišty, které „vedou“ kabinu při pohybu, tvoří dva T profily. Lišty jsou ukotveny mezi horní a dolní panel zapuštěním do neprůchozí díry. Umístění T profilů je podmíněno velikostí kabiny, která je umístěna uprostřed čelní stěny a lícuje s její rovinou. Dalším prvkem je smrková lišta o rozměrech 17x10mm a délce 940mm. Tvoří spojnicí mezi dolním a horním panelem, je umístěna cca v jejich středu a upevněná je podobně jako T profily. Tato lišta slouží jako nosný prvek pro optické snímače polohy. Šachta je dále vybavena protizávažím, umístěným v její zadní části. Vodící ocelové lanko o průměru 1,5mm pro závaží je napnuto a ukotveno v dolním i horním panelu. Protizávaží pohybující se po ocelovém lanku je z oceli a má hmotnost odpovídající přibližně hmotnosti kabiny. V polovině výšky šachty jsou pro zlepšení tuhosti vodících lišt a celé konstrukce vloženy výztuhy.

4.2.Pohonná jednotka

Na horním panelu je umístěna pohonná jednotka. Hřídel pro navíjení tažného lana vyrobený z oceli je svým tvarem přizpůsoben pro uložení do ložisek a pro optimální navíjení tažného lana. Z důvodu úspory hmotnosti je hřídel částečně dutý. Kluzná pouzdra 10x10mm jsou upevněna v úchytech z plexiskla tl.10mm. Tyto úchyty byly zhotoveny dle výkresové dokumentace, firmou Valter Špalek. Úchyty jsou na horním panelu umístěny a přišroubovány tak, aby hřídel byl přímo nad těžištěm kabiny. Pod hřídelem je v horním panelu vyvrtaná díra průměru 25mm pro průchod tažného lana. Motor, jako hnací jednotka kabiny, byl zakoupen jako finální výrobek včetně zabudované převodovky. Proti protáčení je zafixován mechanickou zábranou, dovolující jen mírný pohyb do stran. Přenos kroutícího momentu motoru na navíjecí hřídel je realizován danou konstrukcí obou hřídelů.

4.3.Kabina a protizávaží

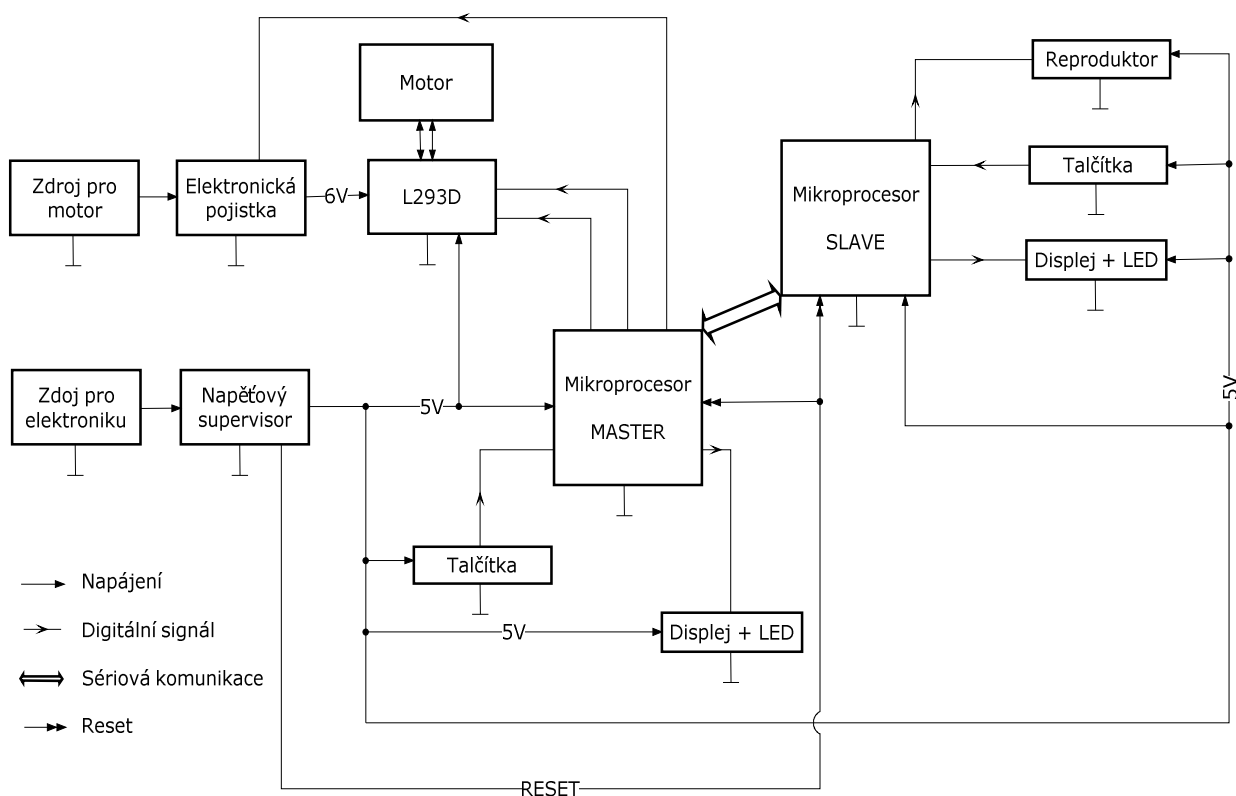
Kabina je zhotovena z kouřového plexiskla tl.2mm a má tvar kvádrů o rozměrech 70x70x90 mm. V čelní stěně jsou kruhové otvory pro tlačítka a jeden obdélníkový otvor pro displej a LED. V podlaze kabiny je několik malých děr pro lepší zvukový efekt reproduktoru. V zadní stěně kabiny je otvor pro vyvedení 5-ti žilového plochého kabelu a celá zadní stěna je odnímatelná pro montáž elektroniky. Na každé boční stěně jsou přilepeny dva vodící hliníkové L profily a na zadní straně jeden T profil pro přerušování světelných závor optických čidel. Lano je ke kabině připevněno pomocí snadno rozebíratelného šroubového spoje. Hmotnost celé kabiny i s elektronikou je cca 120g. Protizávaží je zhotoveno ze tří ocelových kvádrů o rozměrech 10x10x50mm, ve kterých jsou vyvrtány díry pro ocelové vodící lanka a pro závěs. Vzhledem ke složitosti výroby nejsou v kabině ani ve výtahové šachtě realizovány dveře. K signalizaci otevřených či zavřených dveří slouží LED.

4.4.Stabilita

Celá konstrukce je vzhledem k výšce a rozměrům podstavy poměrně nestabilní, proto je vhodné spojit dolní panel s pevnou podložkou, nebo celou konstrukci opřít o svislou oporu.

5.ELEKTRONIKA

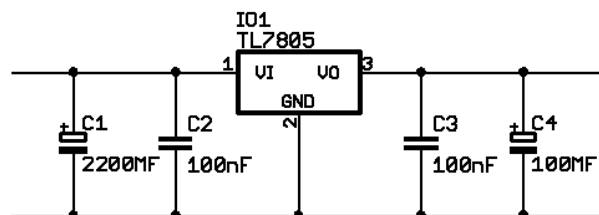
Elektronika je členěna do několika funkčních bloků, jak je patrné z blokového schématu na obr.1. Při výrobě desek plošných spojů (dále jen DPS) se kladl důraz na mechanické přizpůsobení pro montáž těchto DPS. V naprosté většině je použita SMD technologie pro menší rozměry. Na DPS s hlavním řídicím mikroprocesorem je použita dvouvrstvá deska. Všechny DPS byly vyrobeny leptáním v roztoku FeCl_3 . Obrazec spojů byl navržen pomocí freeware verze programu EAGLE, vytištěn na laserové tiskárně a na kupřetiti přenesen nažehlením. Pro propojení jednotlivých DPS byly použity vícežilové ploché kabely a samožezné konektory. V celém modelu výtahu je celkem 15 DPS. Celá elektronika byla navržena přímo pro toto konkrétní použití, pouze elektronická pojistka je převzatá a upravená.



obr.1-blokové schéma elektroniky

5.1. Zdroje

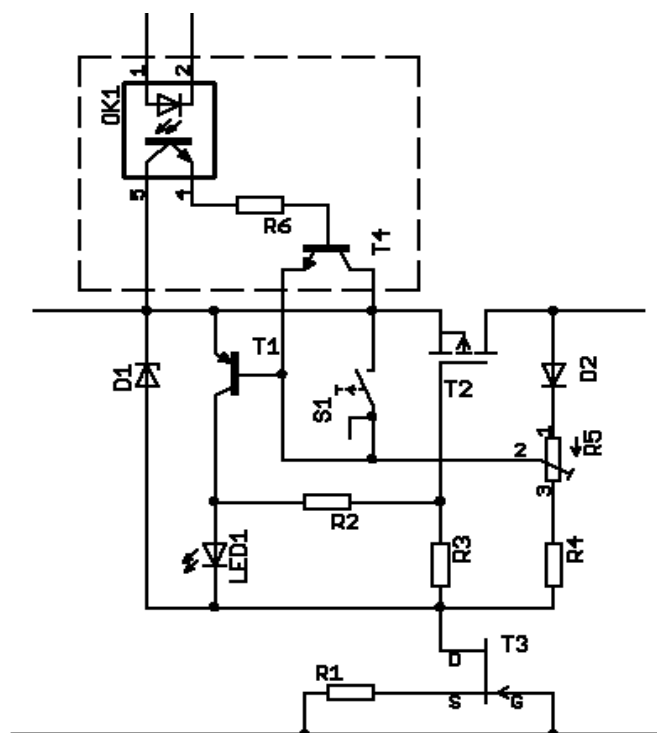
Základ zdroje pro motor tvoří klasický integrovaný třibodový stabilizátor TL7806. Tento je napájen přímo z napájecího konektoru přes diodový dvoucestný usměrňovač, paralelně zapojený vyhlazovací elektrolytický kondenzátor C1 22 μ F a filtrační keramický kondenzátor C2 100nF. Na výstupních svorkách jsou opět paralelně zapojeny kondenzátory C3 100nF a C4 100 μ F – viz obr.2. Tento zdroj má na svém výstupu stabilizované napětí 6V. Vzhledem k malé proudové zátěži tohoto zdroje (cca 50mA) je stabilizátor s malým chladičem přímo na DPS. Zdroj pro procesor a periférie je téměř shodný se zdrojem pro motor, jen je použit stabilizátor TL7805. Vzhledem k vyššímu proudovému odběru je stabilizátor vyveden z DPS a je upevněn na jeden z hliníkových L profilů, který slouží jako chladič.



obr.2 – zapojení stabilizátoru

5.2. Elektronická pojistka

Výstup zdroje pro motor je zapojen na vstup dvojbranu elektronické pojistky. Tato pojistka má za úkol odpojit napájení motoru v případě přetížení kabiny nebo mechanické poruchy. Základ pojistky tvoří MOS-FET tranzistor T2, který je zařazen v kladné větvi sériově se zátěží a tím rozpíná nebo spíná obvod – viz obr.3. Tento tranzistor rovněž plní funkci snímače proudu, úbytek napětí na T2 je úměrný protékajícímu proudu. Tento úbytek je vyhodnocován PNP tranzistorem T1 jehož napětí U_{BE} je dáno součtem napětí na tranzistoru T2, shottkyho diodě D2 (0,3V) a napětí na trimru R7. Překročí-li proud zátěží nastavenou mez, úbytek napětí na T2 vzroste natolik, že pootevře tranzistor T1, tím pádem stoupne napětí na děliči R2,R3 jež je vyveden na řídicí elektrodu T2 a způsobí tím přivření T2. Úbytek na T2 tím ještě více stoupne a nastává řetězová reakce. Nakonec je T1 v saturaci a LED1 se rozsvítí. Pojistka má klopný charakter. Pro „nahození“ pojistky slouží tlačítko S1, které je zapojeno mezi bází a emitor tranzistoru T1 a stiskem tlačítka způsobí úplné zavření tranzistoru T1 vlivem nulového napětí U_{BE} . Zenerova dioda D1 slouží jako zdroj napětí pro napájení pojistky a J-FET tranzistor T3 jako proudový zdroj. Citlivost pojistky se nastavuje trimrem R7. Při testování pojistky nastal problém při rozjezdu a zastavení motoru. Pojistka je velmi citlivá na napěťové i proudové špičky a proto nemohla být nastavena na dostatečnou citlivost. Tento problém byl

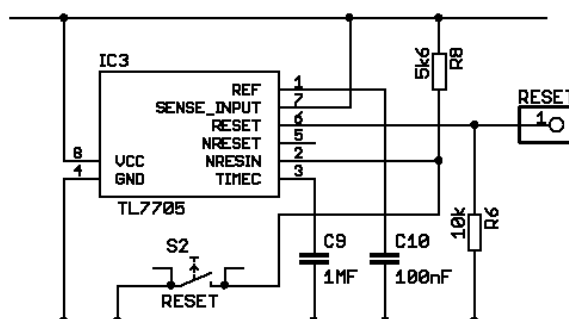


obr.3 - zapojení pojistky s vyznačenou dodatečnou částí

dodatečně vyřešen PNP tranzistorem T4 ve spínacím režimu, který je zapojen paralelně k tlačítku S1. Báze T4 je připojována přes rezistor na kladné napětí prostřednictvím optočlenu OK1. Tento optočlen je přímo spínán mikroprocesorem. V době rozjezdu a zastavení je tedy pojistka deaktivována procesorem. Význam pojistky spočívá v ochraně vinutí motoru proti spálení. V případě přetížení pojistka odpojí napájení. Po odstranění poruchy musí obsluha ručně „nahodit“ pojistku stiskem tlačítka.

5.3. Napěťový supervisor

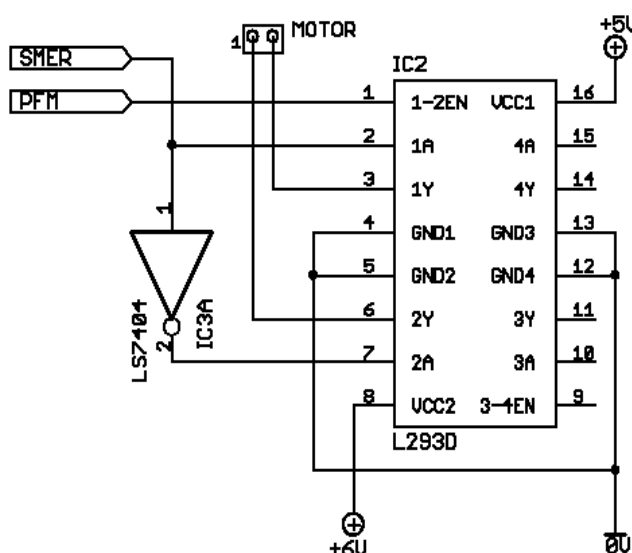
Napájecí napětí pro procesor je nutno hlídat supervisorem. Ten v případě poklesu napětí vygeneruje reset pro oba mikroprocesory. Je tvořen integrovaným obvodem TL7705 – viz obr.4. Tento obvod je přímo určen k tomuto použití a je dodáván ve více variantách pro různá napětí. Pro snadnější orientaci uvádím překlad úryvku z katalogového listu: TL77xx je integrovaný obvod hlídající napájecí napětí. Je speciálně navržen pro použití s mikroprocesory. Během provozu testuje napájecí napětí a drží výstup RESET aktivní tak dlouho, dokud napětí nedosáhne předepsané hodnoty. Přivedení log0 na vstup NRESIN¹ má stejný efekt. Pro zajištění provedení resetu má TL77xx vnitřní zpoždění, které zpozdí návrat výstupu RESET do neaktivního stavu. Toto zpoždění je dáno hodnotou kondenzátoru na vstupu TIMEC. Další externí kondenzátor musí být zapojen mezi výstup REF a zem pro redukci vlivu rychlých změn napájecího napětí.



obr.4- zapojení napěťového supervisoru

5.4. Elektrický pohon

Řízení motoru obstarává obvod L293D – viz obr.5, který je vhodný pro řízení stejnosměrných motorů. Umožňuje pomocí logických vstupů spínat indukční zátěž a zároveň pomocí logických vstupů přepólovat výstupní napětí. Obvod je napájen jednak napětím digitálních úrovní a jednak napětím, které poskytuje na výstupu. Díky svým parametrům je velice vhodný pro realizaci PFM regulace otáček motoru a umožňuje tím plynulý rozjezd a zastavení kabiny. Jedinou nevýhodou tohoto obvodu je poměrně velká klidová spotřeba. Jako pohon byl použit stejnosměrný elektromotor 5V/100mA se zabudovanou převodovkou. Otáčky bez zátěže jsou cca 110/min. Napájení motoru je však zvýšeno na 6V pro větší výkon. Proud v zátěži nepřesahuje 50mA. Proud při zablokování hřídele je cca 180mA, elektronická pojistka vypíná při proudu 80mA.

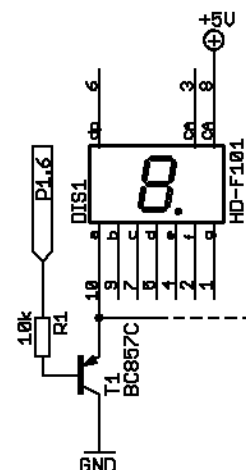


obr.5 – zapojení obvodu L293D

¹ slouží k manuálnímu resetu.

5.5. Displej

V každém patře je 7-mi segmentový LED displej, který číselně zobrazuje aktuální polohu kabiny výtahu. Segmenty jsou zapojeny do paralelních skupin tak, že jeden pin mikroprocesoru ovládá celou skupinu 6-ti segmentů. Například port P1.6 ovládá segment „a“ na všech 6-ti displejích – viz obr.6. Jedna paralelní skupina segmentů má odběr 30-40mA, proto musí být spínána přes tranzistor.



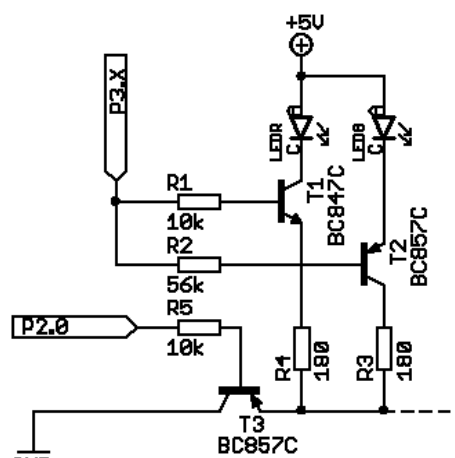
obr.6 – zapojení segmentu „a“

5.6. Tlačítka

Tlačítka slouží pro přivolávání výtahu. Šest tlačítek je připojeno na piny P2.2 – P2.7 řídicího mikroprocesoru. Stiskem tlačítka je přímo na port přivedena log1 (5V). Při uvolnění tlačítka zajišťuje log0 (0V) tzv. pull-down rezistor 5,6kΩ připojující pin na zem.

5.7. LED

V celé šachtě výtahu je celkem 12 signalizačních LED. V každém patře jedna červená a jedna zelená. Pro úsporu pinů mikroprocesoru je každá dvojice ovládána jen jedním pinem portu 3 – viz obr.7. Log 1 otvírá přes rezistor R1 NPN tranzistor T1 a rozsvítí tím červenou LED. T2 je zavřený. V případě log0 se naopak otvírá T2 a T1 se zavře. Možná vás překvapila rozdílná hodnota R1 a R2. To je dáno tím, že v případě log1 se uplatní pull-up rezistor ve vnitřní struktuře procesoru (cca 40kΩ). To znamená $R_B = R1 + R_{PULL-UP}$. Tranzistor T3 pak slouží k rozsvícení nebo zhasnutí LED a je společný pro všech 6 dvojic. Všechny 12 LED tedy využívá pouze 7 pinů procesoru.



obr.7-zapojení signalizačních LED

5.8. Elektronika v kabině

Elektronika v kabině zprostředkovává komunikaci mezi uživatelem v kabině a celým řídicím systémem výtahu. Základem je mikroprocesor, který komunikuje s hlavním procesorem prostřednictvím sériové linky. Osm tlačítek a displej jsou připojeny k pinům procesoru obdobně jako tyto periférie ve výtahové šachtě. Signalizační LED jsou připojeny přímo, každá k jednomu pinu. Reprodaktor pro generování tónu je ovládán jedním pinem přes PNP tranzistor ve spínacím režimu. Celá elektronika včetně tlačítek, displeje i LED je osazena na jedné DPS, která je přizpůsobená pro montáž do kabiny. Reprodaktor je vyveden a připevněn na podlahu kabiny.

6.SOFTWARE

Výtah představuje složité zřízení, jehož činnost je nutno elektronicky řídit. K tomuto řízení se nejlépe hodí buďto programovatelný automat PLC nebo mikroprocesor. Tyto moderní systémy dnes zcela vytlačují zastaralé reléové obvody. Umožňují bezpečnější, přesnější a komfortnější řešení a celkově zvyšují efektivitu využití výtahu.

Pro řízení modelu výtahu byl vybrán mikroprocesor AT89S52 firmy Atmel. Vzhledem k zachování co nejmenších rozměrů a snadnějšímu návrhu DPS bylo použito SMD pouzdro TQFP44. Hlavními kritérii pro výběr byla podpora ISP programování, podpora sériové komunikace a dostatečný počet digitálních vstupně/výstupních bran. Pro konkrétnější představu uvádím tabulku se základními údaji.

6.1.Atmel

Firma Atmel vznikla roku 1984 a zabývá se vývojem a výrobou mikroprocesorů. AT89S52 je klonem dobře známého CISC procesoru AT89C51, jehož základ tvoří jádro 8051 vyvinuté firmou Intel. Jak je z tabulky patrné, nová varianta má větší paměť, podporuje ISP programování a má přidán jeden 16-ti bitový časovač, který lze použít v módu „baudrate generator“ jako generátor rychlosti přenosu pro sériovou linku. Zkratka ISP – In System Programming znamená možnost sériového programování pomocí 4 pinů přímo v aplikaci. Toto programování je velmi rychlé a pro programování SMD procesorů téměř nezbytné.

	AT89C51	AT89S52
Flash paměť [kB]	4	8
USB	ne	ne
ISP	ne	ano
EEPROM	ne	ne
RAM [B]	128	256
f _{MAX} [MHz]	24	33
U _{CC} [V]	4,0-5,5	4,0-5,5
I/O pinů	32	32
UART	1x	1x
16-bitový časovač	2x	3x
Watchdog	ne	ano
SPI	ne	ne
A/D Convertor	ne	ne

6.2.Program

K tomu aby mohl být procesor použit v konkrétní aplikaci, musí do něj být nahrán program. Program je nějakým způsobem zaznamenaný postup operací, které se mají vykonat, tedy algoritmus. Ke grafickému znázornění algoritmu programu slouží takzvaný vývojový diagram. Vývojový diagram používá pro znázornění jednotlivých dílčích operací speciální symboly a šipky, které znázorňují posloupnost. Programy pro mikroprocesory mohou být psány buďto v jazyce symbolických adres (assembleru) nebo ve vyšších programovacích jazycích, např. C. Takto napsaný zdrojový kód je kompilerm přeložen do strojového kódu, HEX souboru. Tento HEX soubor se pak do mikroprocesoru nahraje prostřednictvím programátoru připojeného k PC přes sériové rozhraní COM nebo USB. Konkrétní program, který zajišťuje řízení modelu výtahu je poměrně složitou posloupností a k jeho snadnějšímu pochopení mohou pomoci vývojové diagramy a komentovaný výpis programu.

Model je řízen dvojicí výše uvedených procesorů. Hlavní řídicí procesor je označen jako MASTER a procesor v kabině jako SLAVE. Základem celého programu jsou dva registry "ap" (aktuální poloha) a "zp" (žádaná poloha), nesoucí v sobě číselnou hodnotu (1-6), která odpovídá aktuální/žádané poloze výtahu. Úkolem programu je uvádět obsah registrů do rovnováhy a tím vlastně plnit požadavek obsluhy. Program musí dodržovat určitá pravidla a zároveň komunikovat se všemi perifériemi. Výsledný program se tedy skládá z mnoha rutin a subrutin, které jsou volány z určitých míst posloupnosti. Jsou to například rutiny, které zajišťují tyto funkce: komunikace s čidly, komunikace s displejem, komunikace s tlačítky,

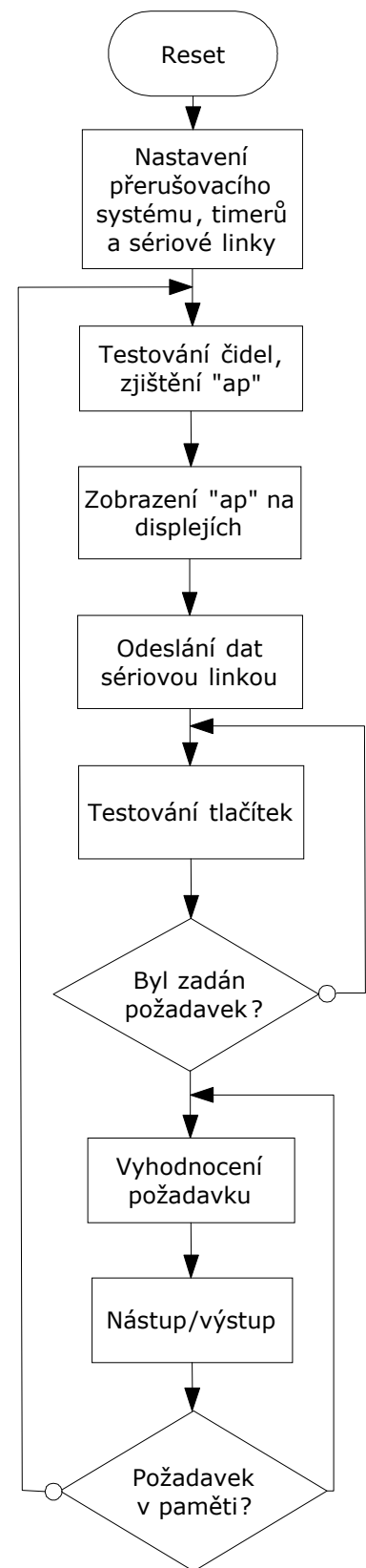
plynulý rozjezd, plynulé zastavení, generování tónu, kodér pro odesílaná data sériovým kanálem, dekodér pro data přijata sériovým kanálem, obsluha sériové linky apod. U obou mikroprocesorů je použit jako generátor hodinového signálu krystal s frekvencí 4,0MHz.

6.3.MASTER

Program v procesoru MASTER funguje následovně –viz obr.8. Po hardwarovém resetu se provedou potřebná nastavení přerušovacího systému, sériové linky a časovačů. Dále se nastaví výchozí hodnoty některých registrů a příznakových bitů. Po nastavení začíná samotný program návštěvím start. Nejprve se testují čidla pro zjištění aktuální polohy kabiny ap (rutina sentest). Následně je zjištěná pozice zobrazena na displejích ve všech patrech (rutina zobraz) a odeslána sériovou linkou do procesoru SLAVE (rutina posli). Nyní je systém připraven a čeká na 1. požadavek. V cyklu se testují všechna přivolávací tlačítka tak dlouho, dokud není některé stisknuto, nebo není jiný požadavek (rutina butttest). Výsledkem této rutiny je žádaná poloha zp. Tento požadavek se vyhodnocuje v rutině lift. Ta má za úkol dopravit kabinu výtahu do požadovaného patra a končí v okamžiku zastavení kabiny. Následuje kontrola zastavení v patře, otevření dveří a čekání na nástup/výstup (4 sekundy). To zajišťuje rutina ready. Po ukončení této rutiny se vždy kontroluje, zda-li není v paměti uložen nějaký požadavek. Pokud ano, je požadavek z paměti vyhodnocen, v opačném případě se program vrací na návštěvím start. V přerušovacím systému procesoru je povoleno přerušení od sériové linky a od timeru 0. Součástí programu jsou tedy i obsluhy těchto přerušení.

6.3.1.Nastavení

Po resetu je nutné nakonfigurovat hardwarové periférie procesoru. K tomu slouží takzvané speciální funkční registry (dále jen SFR). Tyto registry jsou 8-mi bitová paměťová místa. Některé registry jsou adresovatelné bajtově i bitově, některé pouze bajtově. Registr IE povoluje přerušení od sériové linky a od timeru 0. Přestože je použit i timer 2, jeho přerušení nesmí být povoleno. Registry PCON a SCON slouží k nastavení sériového kanálu, registr TMOD nastavuje mód timeru 0. Pro určení přenosové rychlosti sériové linky je použit timer 2 v režimu baudrategenerator. Pomocí registrů TH2,TL2,RCAP2H a RCAP2L je nastavena výchozí a obnovovací hodnota pro horní a spodní bajt 16-ti bitového timeru. Do těchto registrů je nastavena hodnota FFCHEX což odpovídá přenosové rychlosti cca 2400 B/s. Příznakové bity paměti, pohybu, otevřených dveří apod. jsou vynulovány.



obr.8-program MASTER

6.3.2. Test čidel

Testování čidel zajišťuje rutina sentest, která postupně testuje čidla v patrech 0-5. K tomu je použito větvení s pomocí instrukce jnb. Je-li kabina v patře, do registru ap se nastaví příslušná hodnota (1-6) a příznakový bit vpatre. Pokud žádné čidlo v daný okamžik nezaznamenalo kabinu, tedy všechny čidla mají na svém výstupu log1, rutina sentest, končí aniž by změnila obsah ap. Pouze bit vpatre je vynulován.

6.3.3. Test tlačítek

Testování přivolávacích tlačítek obstarává rutina buttest. Zároveň však taky kontroluje, zda není uložen požadavek v paměti, anebo nebyl-li zadán požadavek z kabiny. Testování se provádí v cyklu tak dlouho, dokud nenastane požadavek. Nejprve se testují příznaky požadavku z kabiny a požadavku v paměti (F a F1), je-li jeden z těchto příznaků nastaven, opustí se cyklus, příznak je vynulován a rutina končí (požadavek již byl vložen do registru zp). Jestliže oba tyto příznaky jsou v log0, cyklus pokračuje testováním všech tlačítek. Je-li některé tlačítko stisknuto, opustí se cyklus a do registru zp se nastaví příslušná hodnota (1-6). Po opuštění tohoto cyklu a následném ukončení rutiny buttest vždy následuje rutina lift, která představuje vyhodnocování požadavku. Proto cyklus nemůže být ukončen dříve, než je znám požadavek.

6.3.4. Vyhodnocení požadavku

Vyhodnocení požadavku je po stránce softwaru nejkomplicovanější částí. Zajišťuje jej rutina lift, která obsahuje ještě další takzvané subrutiny. Do rutiny lift vstupují registry ap a zp, a jejím cílem je uvést obsahy těchto registrů do rovnováhy. Nejprve se znovu volá rutina sentest pro ověření aktuální polohy. Dále se rozsvítí všechny červené signalizační LED. Nyní se pomocí instrukce subb odečte od obsahu registru ap obsah registru zp. Je-li výsledek nulový², rutina končí. Je-li výsledek záporný, je nastaven směr pohybu nahoru, pokud je výsledek kladný, směr pohybu je dolů. Nyní se zavolá subrutina jed pro plynulý rozjezd kabiny. Během jízdy se v cyklu testují čidla rutinou sentest a sleduje se změna aktuální polohy. Pokud je tato změna zaznamenána je nový údaj zobrazen, odeslán do kabiny a porovnává se s žádanou polohou. Dále se v cyklu kontroluje správný směr pohybu a testují se tlačítka subrutinou savetest³. Je-li zaznamenán špatný směr pohybu kabiny, pomocí rutiny otocka se provede změna směru. Rovnají-li se obsahy registru ap a zp, cyklus končí. Následuje rutina stuj pro plynulé zastavení. Nakonec se provede zobrazení aktuální pozice na displeji a tím rutina lift končí. Subrutina otocka pouze volá rutinu stuj, neguje bit smer a volá rutinu jed.

6.3.5. Nástup/výstup

Po zastavení kabiny v patře se provádí rutina ready. Nejprve se opět provede rutina sentest, kontroluje se správnost zastavení v patře a nastaví se příznakové bity libit a rebit. Příznak libit rozsvěcuje displej v kabině a příznak rebit otvírá v kabině dveře. V případě zastavení kabiny mezi patry je tedy rebit vynulován, a je nastaven další příznakový bit gobit který rozsvítí v kabině červenou LED, tedy dveře zůstanou zajištěny. Po nastavení

² Zamená, že se kabina právě nachází v patře, do kterého byla přivolána

³ Případný požadavek je uložen do paměti.

příznakových bitů se volá rutina posli pro odeslání aktuálního stavu do kabiny. Pokračuje se otevřením příslušných dveří ve výtahové šachtě. To zajišťuje rutina ledka. Dále je volána rutina wait4 pro časové zpoždění cca 4sekundy. Protože je toto zpoždění realizováno časovačem, mohou být během tohoto zpoždění testovány tlačítka. Po 4 sekundách se všechny piny ovládající LED nastaví do log1 (svítí červená LED), avšak zároveň jsou pinem lede všechny LED zhasnuty. Nastavení všech LED do červené se tedy provádí pouze proto, aby při příštím rozsvícení již nemusely být nastavovány. Následně se vynulují příznaky gobit a rebit viz.výše, a aktuální stav je odeslán. Tím rutina ready končí.

6.3.6.Kontrola paměti

Během vyhodnocování požadavku, tedy během trvání rutiny lift a ready, mohl být do paměti uložen nový požadavek, nebo během vykonávání rutiny ready mohl přijít nový požadavek obsluhy z kabiny. Proto po vyhodnocení aktuálního požadavku jsou tyto možnosti kontrolovány. Testuje se bit membit1, je-li nastaven, znamená, že v paměti je požadavek a pokračuje se na návěští loading. Pokud není nastaven skáče se na návěští start a celý cyklus se opakuje. Za návěštím loading se ještě kontroluje příznak požadavku z kabiny F , je-li tento příznak nastaven, následuje skok na start, protože požadavek obsluhy z kabiny má vyšší prioritu než požadavek v paměti. Pokud je příznak F nulový, provede se přesun hodnoty z první paměťové pozice mp1 do registru zp. Případná druhá a třetí paměťová pozice se přesunou do nižší úrovně a uvolní se tak třetí paměťová pozice mp3. Následně se nastaví příznak F1 a skáče se na start.

6.3.7.Odeslání dat

Rutina která zajišťuje odeslání dat z mikroprocesoru MASTER do mikroprocesoru SLAVE sériovou linkou je nazvána posli. Odesílá se vždy jen jeden bajt, který v sobě nese aktuální informace o pozici kabiny a o stavu výtahu. Odesílá se registr statbyte2, který se nachází v bitově adresovatelné oblasti paměti. Do dolní tetrády tohoto bajtu je vložena hodnota z registru ap a horní tetráda obsahuje tři příznakové bity. Příznak pohybu kabiny gf2 je nastavován podle příznaku gobit. Příznak otevření dveří kabiny rf2 podle příznaku rebit a příznak rozsvícení displeje v kabině lf2 podle příznaku libit. Tyto bity jsou nastaveny prostřednictvím carry a instrukce mov. Nakonec se statbyte2 přesune do registru sbuf a automaticky se zahájí přenos. Tím rutina posli končí.

6.3.8.Plynulý rozjezd

Plynulý rozjezd je realizován pomocí softwarově generovaného PFM signálu, který spíná napájení motoru. To znamená, že v cyklu je vždy generována konstantní doba log1 na výstupu, která se střídá s postupně se zkracující dobou log0. To má za následek plynulé zvyšování otáček motoru. Nakonec je doba log0 zanedbatelně krátká a motor tedy dosahuje plných otáček. Nejprve se v rutině vynuluje bit fuse, čímž se deaktivuje elektronická pojistka. Nastaví se příznak gobit a zavolá se rutina posli. Do registru zero se nastaví výchozí hodnota 200_D . Nyní začíná cyklus ve kterém se nejprve nastaví bit mot, který ovládá motor, poté se volá subrutina on. Po návratu z této subrutiny se dekrementuje registr zero, jeho hodnota se zkopíruje do pomocné proměnné pom a vynuluje se bit mot. Nakonec se volá subrutina off , která zajišťuje dobu generování nízké úrovně. Poté se celý cyklus opakuje. Cyklus se opakuje tak dlouho dokud není v registru zero nulová hodnota. Dále se aktivuje pojistka nastavením bitu fuse a rutina jed končí.

6.3.9.Plynulé zastavení

Brzdění je realizováno podobným cyklem jako rozjezd, pouze doba t_{OFF} se nezkracuje, ale prodlužuje. To je dáno tím, že v cyklu se registr zero inkrementuje z hodnoty 0 až na hodnotu 255_D. Před vstupem do cyklu se opět deaktivuje pojistka a současně se kontroluje zda již není kabina v klidu. Po ukončení cyklu se vynuluje příznak gobit a aktivuje se elektronická pojistka. Rutina stuj končí.

6.3.10.Subrutina on a off

Subrutina on slouží k časování konstantní doby t_{ON} . K tomu je použita časová smyčka s použitím instrukce djnz. Do registru puls je vložena hodnota 166_D, která je postupně dekrementována. To znamená zpoždění 1ms. Doba t_{ON} , tedy trvá 1milisekundu.

Subrutina off funguje podobně, avšak doba t_{OFF} je závislá na tom, pokolikáté se cyklus provádí, proto do této subrutiny vstupuje proměnná pom a ta je pak ve smyčce dekrementována. Pro „protažení“ je ve smyčce navíc zařazeno zpoždění 6 cyklů. Takže jedno opakování cyklu, ve kterém se jednou dekrementuje registr pom trvá 8 cyklů, což je při použitém krystalu s frekvencí 4,0MHz doba 24 μ s. Subrutina off je tedy schopna realizovat zpoždění o délce 24 μ s až 6,12ms. Tento rozsah se při testování ukázal jako nejvhodnější pro optimální průběh rychlosti rozjezdu a zastavení.

6.3.11.Zobrazení

Pro zobrazení aktuální polohy kabiny slouží rutina zobraz. Obsah registru ap je číselně zobrazen na 7-mi segmentovém displeji. V rutině se hodnota registru ap přesune do registru R0 a postupně se porovnává s hodnotami 0-6 a v případě rovnosti je do registru disp vložena hodnota která odpovídá rozsvícení příslušných segmentů.viz tab. Pro hodnoty 0;7-255 je zobrazeno „E“, které značí chybné zjištění polohy. Pro hodnoty 1-6, je zobrazena číslice 0-5, pro nulté až páté patro.

Port:	-----	P1.6	P1.5	P1.4	P1.3	P1.2	P1.1	P1.0
Segment:	dp	a	b	c	d	e	f	g
Zobrazení	0	1	0	0	0	0	0	1
	1	1	1	0	0	1	1	1
	2	1	0	0	1	0	0	0
	3	1	0	0	0	0	1	1
	4	1	1	0	0	1	1	0
	5	1	0	1	0	0	1	0
	E	1	0	1	1	0	0	0

6.3.12.Subrutina ledka

Tato subrutina je velmi primitivní. Postupně otestuje všechna čidla a v patře, ve kterém se nachází kabina, zhasne červenou a rozsvítí zelenou LED. Po otestování všech čidel rutina končí. V případě, že kabina nebyla zaznamenána v žádném patře, svítí všechny červené LED.

6.3.13.Časování 4 sekundy

Časování doby otevřených dveří je realizováno pomocí timeru 0 v módu 1. Do 16-ti bitového časovače je vložena nula, tato nula je postupně inkrementována a k přerušení dochází v okamžiku přetečení, tedy po dosažení hodnoty 65 535. Je tedy nastavena největší možná doba časování cca 200ms. Do registru time se vloží hodnota 20, povolí se přerušení od

timeru 0 a timer 0 se spustí. Dále se v cyklu opakuje volání subrutiny savetest a kontrola nenulovosti proměnné time. Proměnná time se dekrementuje v obsluze přerušení timeru 0. Cyklus končí v okamžiku kdy obsah proměnné time je roven nule, tím končí i subrutina wait4.

6.3.14.Subrutina savetest

Tato subrutina slouží k testování tlačítek v době vyhodnocování požadavku a k uložení případného dalšího požadavku do správné paměťové pozice. Nejprve se postupně testují všechna přivolávací tlačítka, není-li žádné tlačítko stisknuto rutina končí. Je-li některé tlačítko stisknuto, podobně jako v rutině buttest se provede přiřazení příslušné hodnoty do registru mp, který představuje jakousi mezipaměť. Následně se testují příznakové bity membit1, membit2 a membit3. Jsou-li všechny tyto bity nastaveny na log1, znamená to, že jsou již všechny 3 paměťové pozice obsazené, a rutina končí. Jinak je hodnota z mezipaměti uložena do nejnižší volné paměťové pozice. Před uložením se však kontroluje, zda hodnota v mezipaměti není shodná s hodnotami v registrech ap, zp nebo s nižšími paměťovými pozicemi, tedy registry mp1 a mp2. V případě, že se tato hodnota shoduje alespoň s jednou zmíněnou hodnotou, požadavek není uložen a rutina končí. Touto kontrolou je tedy zajištěno, že do paměti nemůže být uložen požadavek shodný s aktuální polohou kabiny, s žádanou polohou kabiny, ani nemůže být v paměti uložen stejný požadavek ve více paměťových pozicích. Přestože je rutina poměrně obsáhlá, nejdelší možný čas jaký může být vykonávaná je cca 105 μ s. Jelikož je perioda přerušení timeru 0 nastavena na 200ms, nehrozí, aby v rámci vykonávání této rutiny nastalo přerušení 2x a následkem toho došlo k chybě ve vykonávání programu.

6.3.15.Obsluha přerušení Timeru 0

Po přerušení od Timeru 0 ukazuje program counter na adresu 0B_{HEX} na této adrese v paměti programu je zapsána instrukce skok na návěští timing. Za tímto návěštím následuje obsluha přerušení. Nejprve se provede pomocí instrukce push záloha obsahu akumulátoru do stacku, zakáže se přerušení od sériové linky a zastaví se timer 0. Dále se provede vynulování registru časovače a dekrementuje se hodnota v registru time. Je-li po této dekrementaci hodnota již nulová, zakáže se přerušení od timeru 0, povolí se přerušení od sériové linky, a pomocí instrukce pop se provede obnovení hodnoty akumulátoru. V opačném případě se spustí timer 0, povolí se přerušení od sériové linky, a provede se obnovení hodnoty akumulátoru. Příkazem reti pak obsluha přerušení končí.

6.3.16.Obsluha přerušení sériové linky

Vektorem přerušení sériové linky je adresa 23_{HEX}. Z této adresy program counter skáče na návěští uart kde je zapsána obsluha přerušení. V obsluze se nejprve provede záloha obsahu akumulátoru a zakáže se přerušení od timeru 0. Nyní se zjišťuje, zda přerušení vzniklo vlivem ukončení vysílání, nebo příjmu. V případě ukončení vysílání se pouze vynuluje příznak přerušení ti, povolí se přerušení od timeru 0, obnoví se akumulátor a obsluha přerušení končí. V případě příjmu dat se přijatá data zkopírují z registru sbuf do registru statbyte1. Není-li kabina v daném okamžiku v pohybu, nastaví se příznak požadavku z kabiny F. Dále se dolní tetráda obsahu registru statbyte1 zkopíruje do registru zp a nastaví tím žádanou polohu. Do registru time se vloží hodnota 1, proto aby se ukončilo zpoždění a požadavek byl ihned vyhodnocen. Poté se vynuluje příznak přerušení ri a povolí se přerušení od timeru 0. Obsluha přerušení pak končí obnovou akumulátoru.

Celý program byl vytvořen v jazyku symbolických adres assembleru, ve freeware verzi programu KEIL μ vision2. Tento program obsahuje debugger pro odladění programu. Vývojové diagramy jsou v příloze.

6.4.SLAVE

Mikroprocesor SLAVE má za úkol komunikovat s obsluhou v kabině prostřednictvím signalizačních LED, displeje, reproduktoru a tlačítek. Získaná data odesílá do procesoru MASTER sériovou linkou. Po resetu se opět nejprve provede nastavení potřebných speciálních funkčních registrů a výchozích hodnot některých proměnných. V celém programu se pak v cyklu opakují jen dvě rutiny. Rutina zobraz – ta je naprosto stejná jako v programu MASTER, proto nebude znovu popisována a rutina buttest pro testování tlačítek. Dále program obsahuje obsluhu přerušení od sériové linky a timeru 0. Nejzajímavější částí programu je zřejmě generování tónů. Pro generování tónu s plynule kolísající frekvencí je použit timer 0 v módu 2, tedy 8-mi bitový časovač v režimu autoreload.

Nastavení

Po resetu jsou nastaveny SFR a některé proměnné. Nastavení sériové linky a timeru 2 je naprosto stejné jako u mikroprocesoru MASTER, aby byl zajištěn bezpečně fungující přenos dat. Pouze v registru TMOD se nastaví timer 0 do módu 2. Dále je do registru ap vložena výchozí hodnota 0, příznak libit je vynulován a příznak tibit nastaven.

6.4.1.Testování tlačítek

Testování probíhá podobně jako u procesoru MASTER. Postupně se testují všechna tlačítka. Pro odstranění zákmitů tlačítek je po stisku i po uvolnění tlačítka realizováno zpoždění subrutinou antikmit cca 5ms. Je-li stisknuto některé z tlačítek 0-5, do registru zp se vloží příslušná hodnota a pomocí subrutiny posli se odešle do řídicího mikroprocesoru. Tím rutina buttest končí. Při stisku tlačítka STOP se do registru zp se vloží hodnota z registru ap a po odeslání tím způsobí okamžité zastavení kabiny. Je-li stisknuto tlačítko ZVONEK, nastaví se do registru TH0 hodnota 255- výchozí hodnota pro timer 0. Dále se do akumulátoru a do registru reload vloží hodnota 1. Ta určuje frekvenci pulsování generovaného tónu. Nyní se spustí timer 0 a čeká se na uvolnění tlačítka. Následně je timer 0 zastaven a rutina končí.

6.4.2.Odeslání dat

Subrutina posli je velmi primitivní. Před odesláním se kontroluje zda bylo ukončeno předchozí vysílání, případně se čeká na toto ukončení. K tomu je použit pomocný příznak tibit – příznak ukončeného vysílání. Je-li tedy tento bit nastaven pokračuje se jeho vynulováním a následně se zahájí přenos zkopírováním obsahu registru zp do registru sbuf a subrutina končí.

6.4.3.Subrutina antikmit

Pro odstranění zákmitů tlačítek je realizováno zpoždění cca 5ms pomocí 3 dekrementačních nevnořených smyček. Čas 5ms se při testování ukázal jako dostatečný. Do proměnné kmit je vložena hodnota 255, která je postupně dekrementována až do hodnoty

nula. To se provede celkem 3x což při použitím krystalu 4,0MHz vytvoří zpoždění oněch 5ms.

6.4.4. Obsluha přerušení sériové linky

Vektorem přerušení sériové linky je adresa 23_{HEX}. Z této adresy program counter skáče stejně jako u procesoru MASTER na návěští uart kde je zapsána obsluha přerušení. V případě přerušení vlivem ukončeného vysílání se pouze vynuluje příznak přerušení ti, nastaví se příznak tibit a obsluha končí. V případě příjmu se zkopírují přijatá data z registru sbuf do registru statbyte1 a volá se subrutina dekodek pro vyhodnocení právě přijatých dat. Registr statbyte1 se nachází v bitově adresovatelné oblasti paměti, aby bylo možno zjišťovat stavy jednotlivých příznakových bitů. Po ukončení subrutiny dekodek se vynuluje příznak přerušení ri a obsluha přerušení končí.

6.4.5. Subrutina dekodek

V této rutině se vyhodnocují přijatá data a v rámci této rutiny je mimo jiné také generován tón otevření dveří. Nejprve se dolní tetráda přijatého bajtu, tedy dolní tetráda registru statbyte1 zkopíruje do registru ap. Dále se pracuje s příznaky obsažené v horní tetrádě přijatého bajtu. Dle příznaku gf1 se nastaví bit red, ovládající červenou LED, avšak vzhledem k tomu že LED svítí při log0, je tento bit nejprve negován. Dále se dle rf1 nastaví bit green, opět s předchozí negací. Posledním příznakem je lf1 dle jehož negace se nastaví bit libit, který ovládá napájení displeje. V případě, že při dekódování byla rozsvícena zelená LED, tedy byl vynulován bit green provede se generování tónů „a“ a „d“ pomocí subrutin ah a d a subrutina dekodek končí. Jinak rutina končí bez generování tónu.

6.4.6. Generování tónů

Pro akustickou signalizaci dojezdu kabiny do patra a otevření dveří jsou generovány tóny „a“ a „d“, každý po dobu cca 0,5sekundy. Generování je zajištěno v cyklu s podmínkou na konci. V tomto cyklu se střídavě nastavuje a nuluje bit (pin) piez, který je zapojen na reproduktor. Mezi těmito změnami je realizováno zpoždění, které určí délku půlperiody signálu a tím i výšku tónu. Zpoždění je realizováno jednou či dvěma dekrementačními smyčkami. V cyklu se také inkrementuje počítadlo, a po dosažení určité hodnoty je cyklus ukončen. Tím je dána doba trvání tónu. Konkrétní hodnoty zpoždění jsou odvozeny od frekvence tónu „a1“ – 440Hz a tónu „d1“ – 296Hz.

6.4.7. Obsluha přerušení timeru 0

Timer 0 slouží pro generování tónu zvonku. Perioda přerušení tohoto timeru udává půlperiodu výsledného tónu. Obsluha přerušení je zapsána za návěštím ton, kam je program counter přeměřován z adresy 0B_{HEX}. V obsluze se nejprve neguje bit piez, dekrementuje se akumulátor a v případě že po této dekrementaci není nulový, obsluha končí. V konkrétním případě se v akumulátoru nachází hodnota 1, takže po dekrementaci bude vždy 0. V tom případě obsluha nekončí a provede se zvýšení nebo snížení obsahu registru TH0, čímž se mění perioda přerušení timeru 0 a tím i frekvence generovaného tónu. Obsah TH0 nabývá hodnot 0-100. V každé obsluze přerušení se hodnota mění o 1. Po dosažení maximální hodnoty (100) se hodnota snižuje až do dosažení minima(0), pak se opět začne zvyšovat. Viz.vývojový diagram v příloze.

7.ZÁVĚR

Výsledkem práce je dle původního plánu funkční model osobního výtahu. Při realizaci bylo použito mnoho znalostí získaných z odborných předmětů nebo samostudiem. Model dobře ukazuje možnosti řízení mikroprocesorem s jádrem 8051. Při vývoji byla největším problémem realizace plynulých rozjezdů a zastavení. K tomu je možno využít přechodového děje RC článku s vhodnou časovou konstantou τ a vhodného napěťového sledovače. Toto řešení je sice jednoduché po stránce softwaru řídicího procesoru, avšak není možná pozdější softwarová regulace průběhu zastavování či rozjíždění. Proto bylo zvoleno modernější a efektivnější řízení pomocí PFM. V původním testovacím zapojení byl motor spínán výkonovým tranzistorem přímo z procesoru, avšak vzhledem k indukční zátěži motoru byl tento systém nestabilní. Tento problém byl vyřešen použitím obvodu L293D, který je pro toto použití určen, umožňuje řízení digitálními signály a svou vnitřní strukturou eliminuje negativní vliv proudových špiček vinutí motoru. Schéma elektronické pojistky bylo převzato z již neexistující webové stránky.

8.LITERATURA

Při vývoji elektroniky i softwaru jsem čerpal z katalogových listů použitých součástek a z webových stránek týkajících se této problematiky.

MORAWIEC, Martin. Kurz assembler, [online]

<<http://programujte.com/index.php?akce=clanek&cl=2005071301-assembler-uvod>>

MORAWIEC, Martin. Keil-práce v programu, [online]

<<http://programujte.com/index.php?akce=clanek&cl=2005121003-keil-prace-v-programu>>

HANKOVEC, David. Popis 8051, [online]

<<http://www.dhservis.cz/popis8051.htm>>

BEZSTAROSTI, Jiří. Jednochip-to je ono! [online]

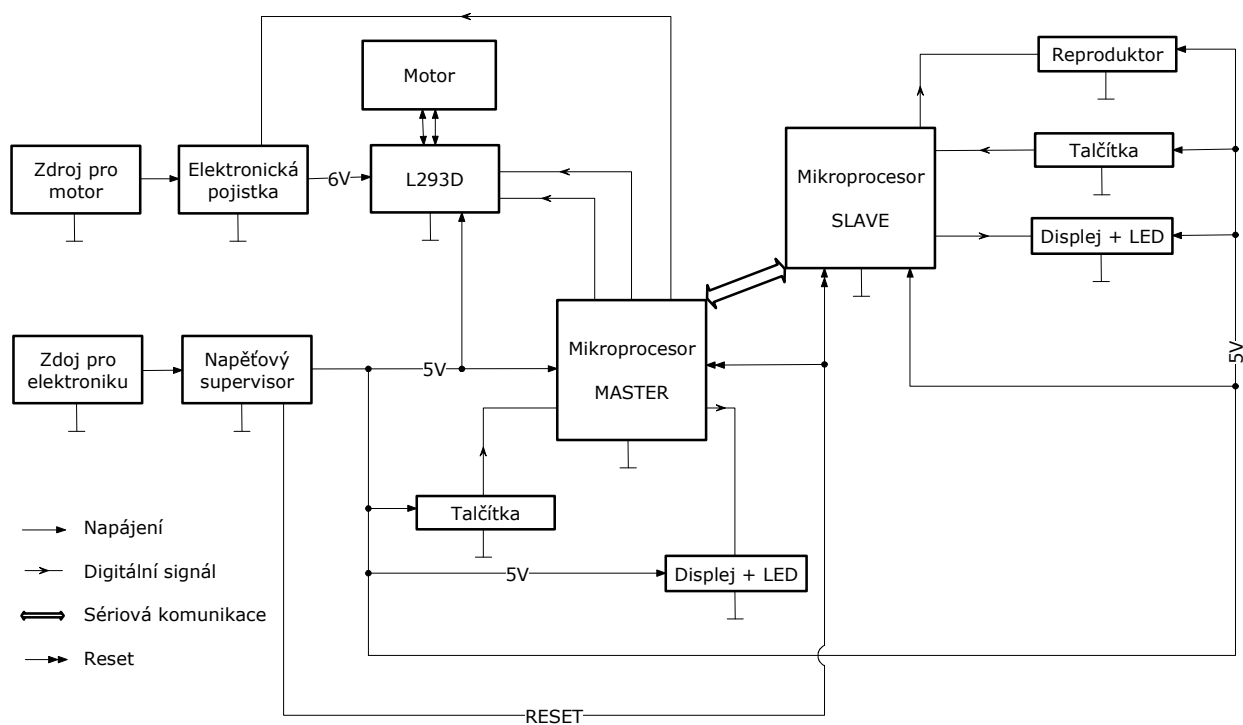
<<http://www.bezstarosti.cz/elec/jednochip/jednochip.htm>>

ATMEL. Code examples [online]

<http://www.atmel.com/dyn/products/tools.asp?family_id=604#1535>

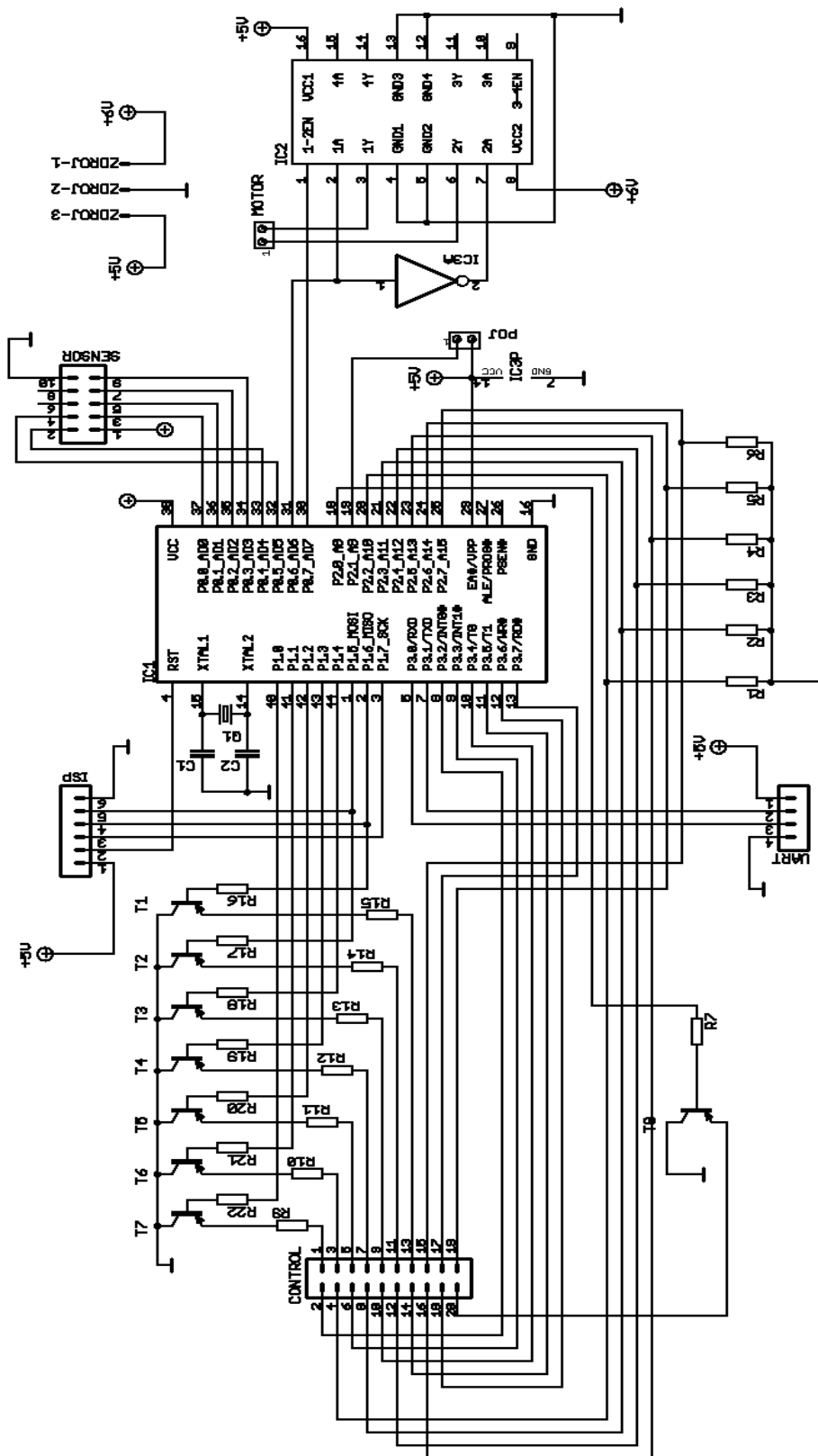
9.PŘÍLOHY

9.1.Blokové schéma elektroniky

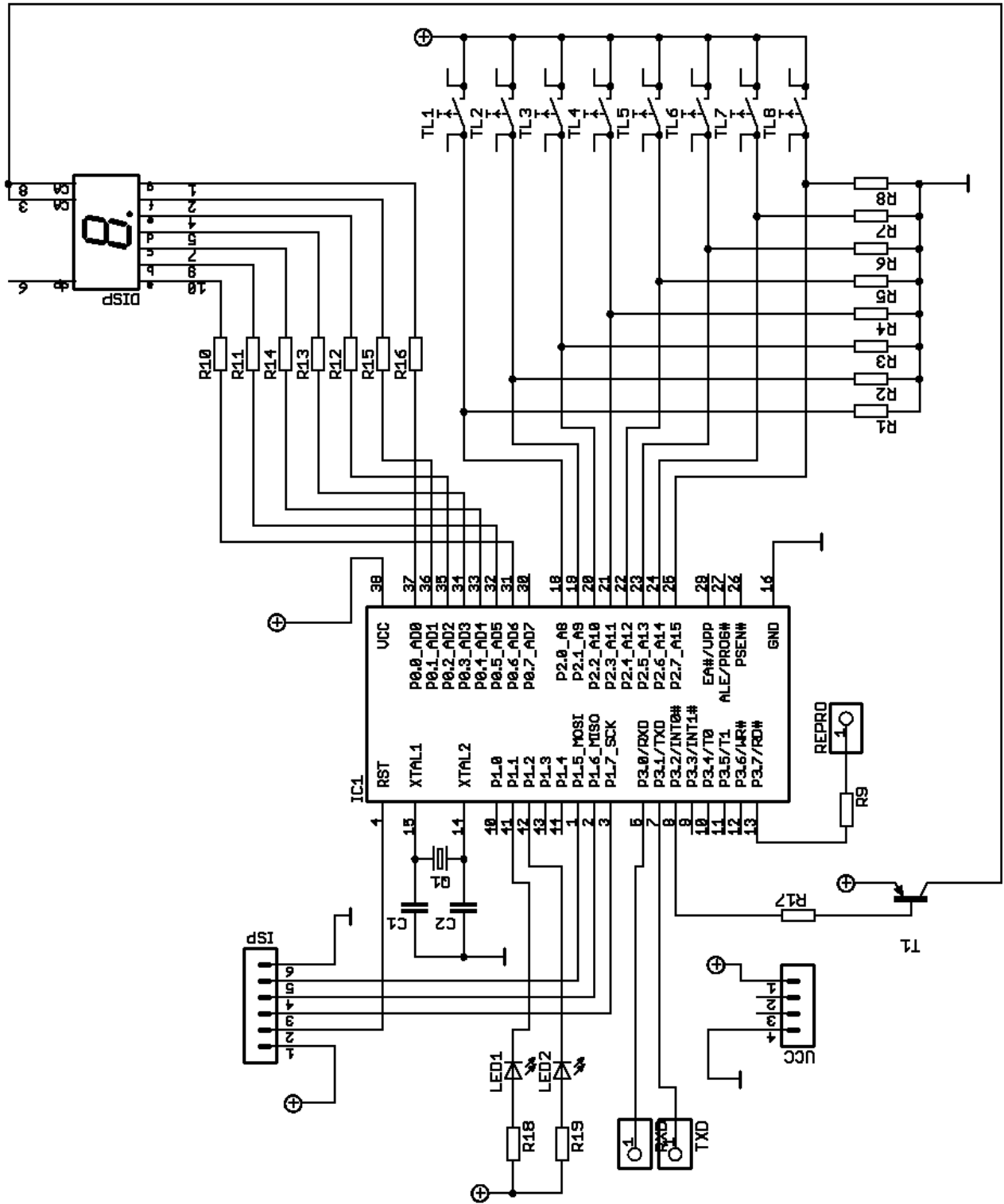


9.2. Elektrotechnická schémata

9.2.1.MASTER



9.2.2.SLAVE



9.2.3.Výpis součástek

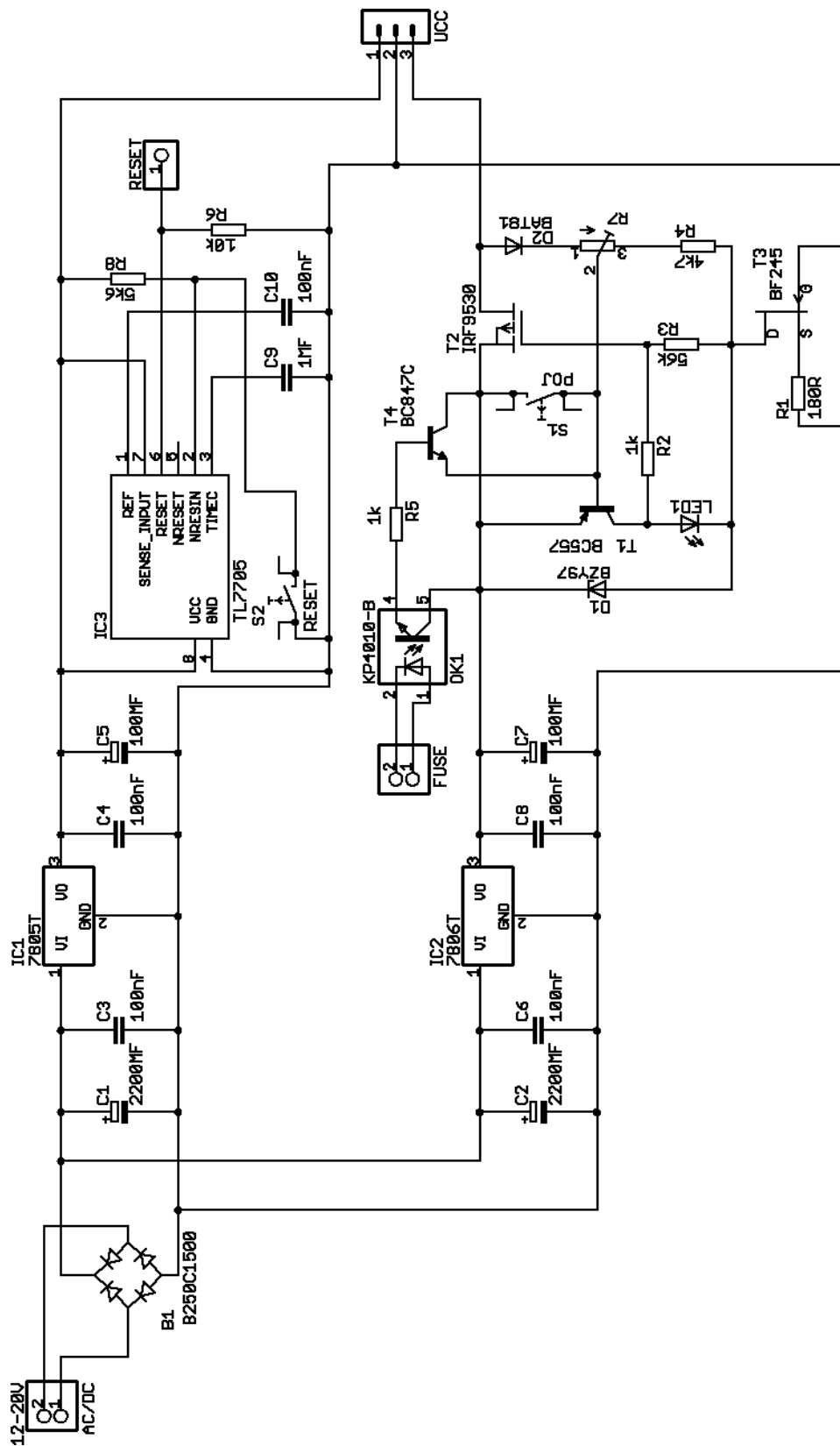
MASTER:

R1-R6.....5k6
R7,R16-R22.....10k
R9-R15.....80R
C1,C2.....33pF
T1-T8.....BC857C
Q1.....4,000MHz
IC1.....AT89S52
IC2.....L293D
IC3.....74LS04

SLAVE:

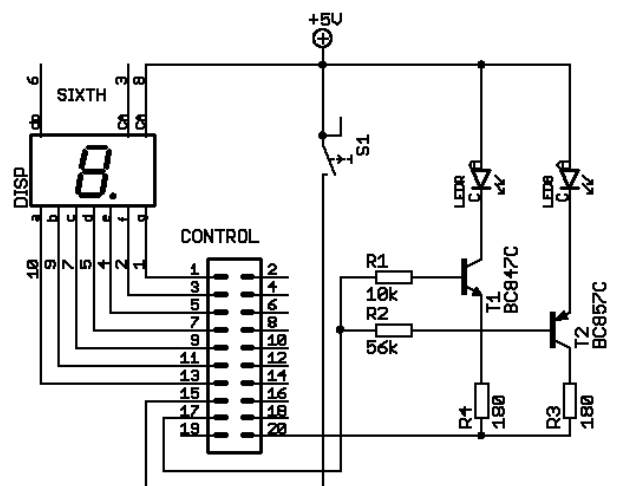
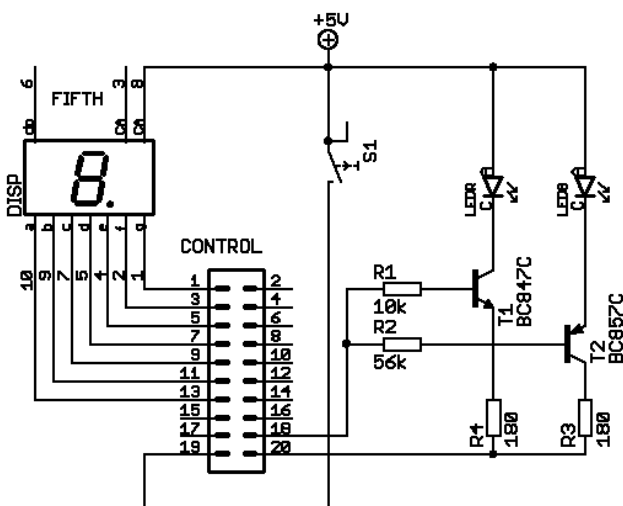
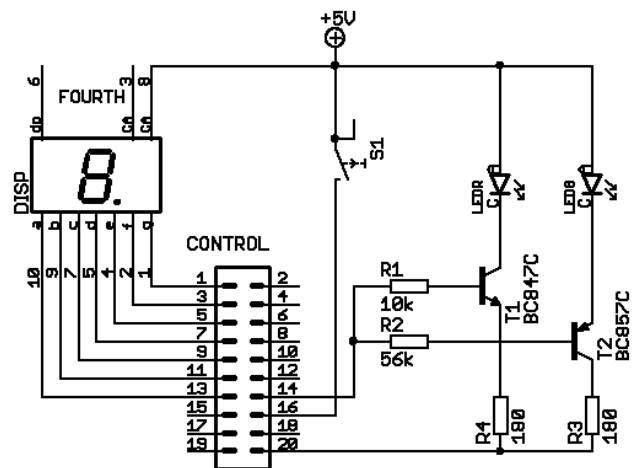
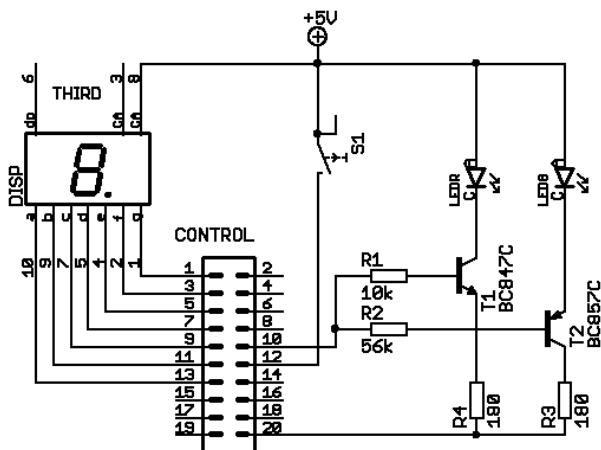
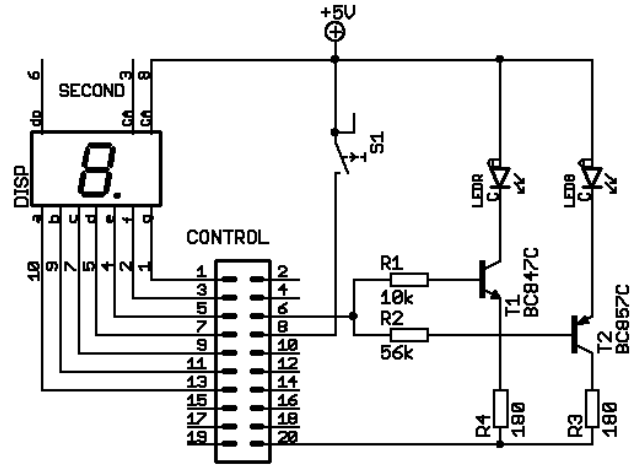
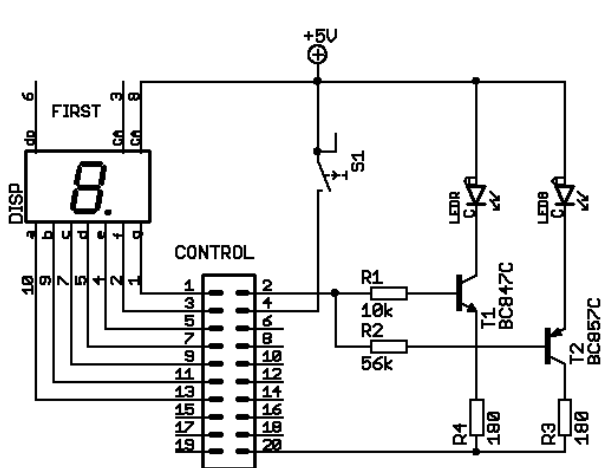
R1-R8.....5k6
R9,R18,R19.....180R
R10-R16.....510R
R17.....10k
C1,C2.....33pF
T1.....BC857C
Q1.....4,000MHz
IC1.....AT89S52
DISP.....HD-A342RD
LED1.....LED2x5mm,zelená
LED2.....LED2x5mm,červená
TL1-TL8.....tlačítko, bez aretace, modré

9.2.4.Zdroj



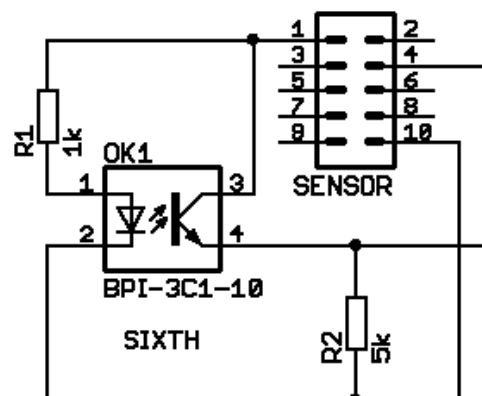
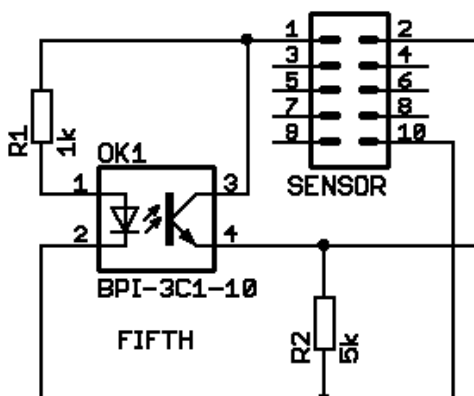
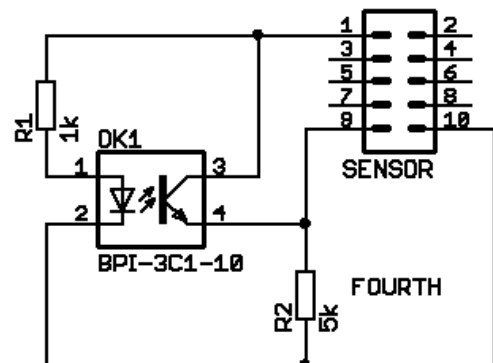
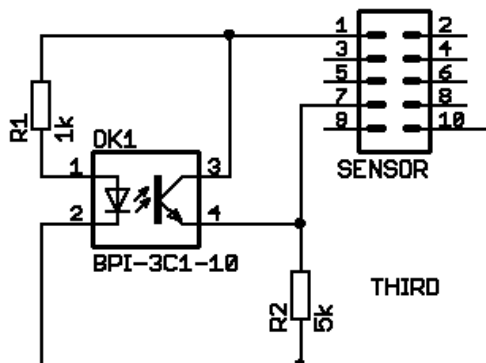
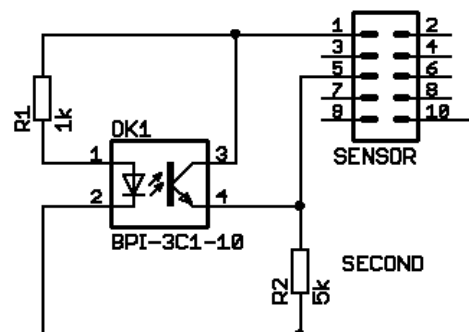
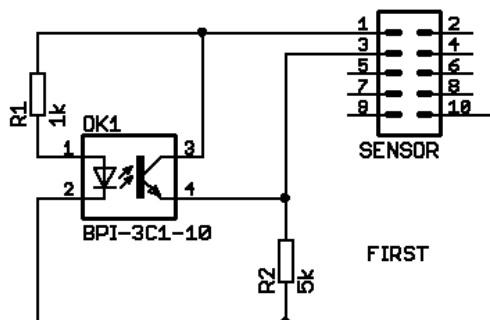
9.2.5. Control 1-6

Ovládací panely obsahující tlačítko, 2 LED a displej.



9.2.6.Sensor 1-6

DPS s optickými čidly, pro zjišťování polohy kabiny.

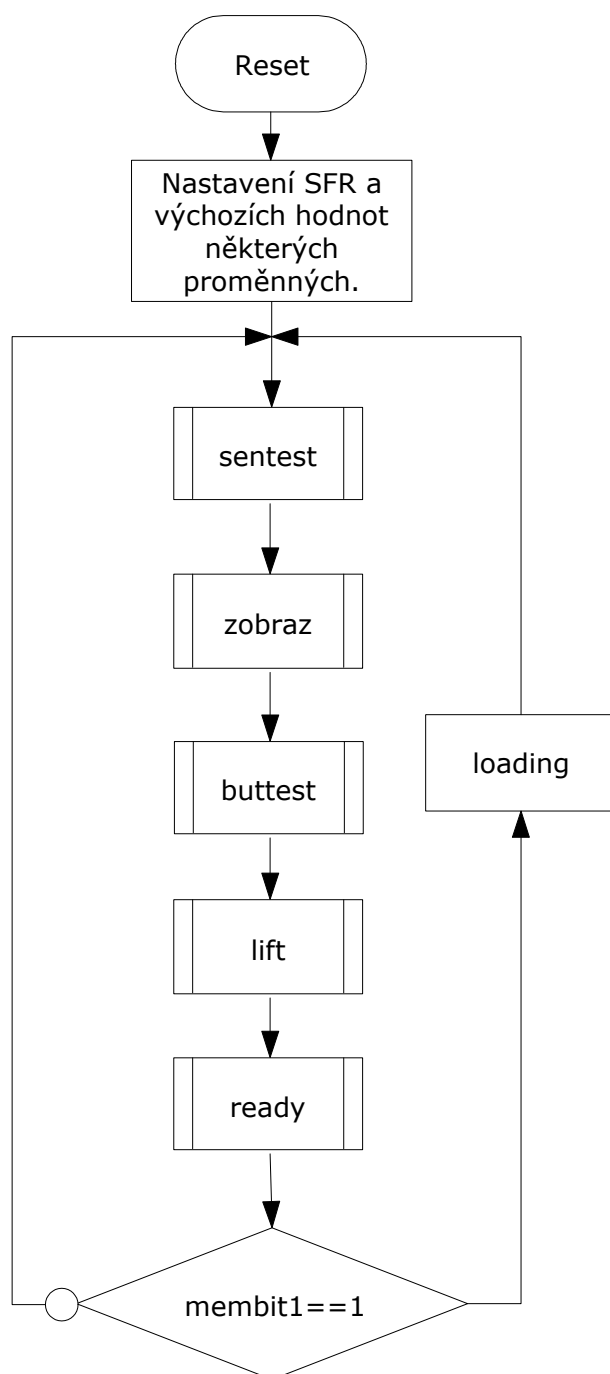


9.3. Vývojové diagramy

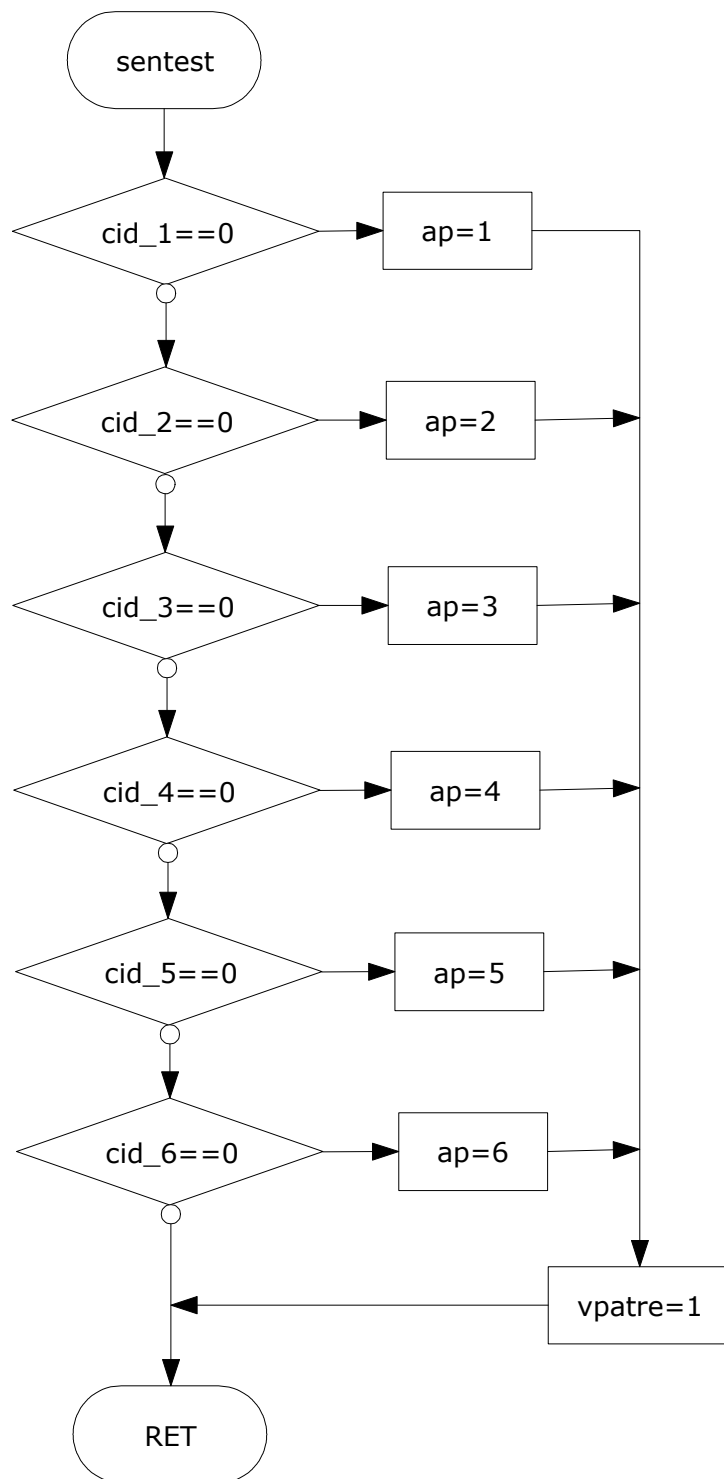
Použité symboly:

- x=y.....přiřazení
- x=\x\.....negace bitu
- x==y.....porovnání (x je rovno y)
- x!=y.....porovnání (x je různé od y)

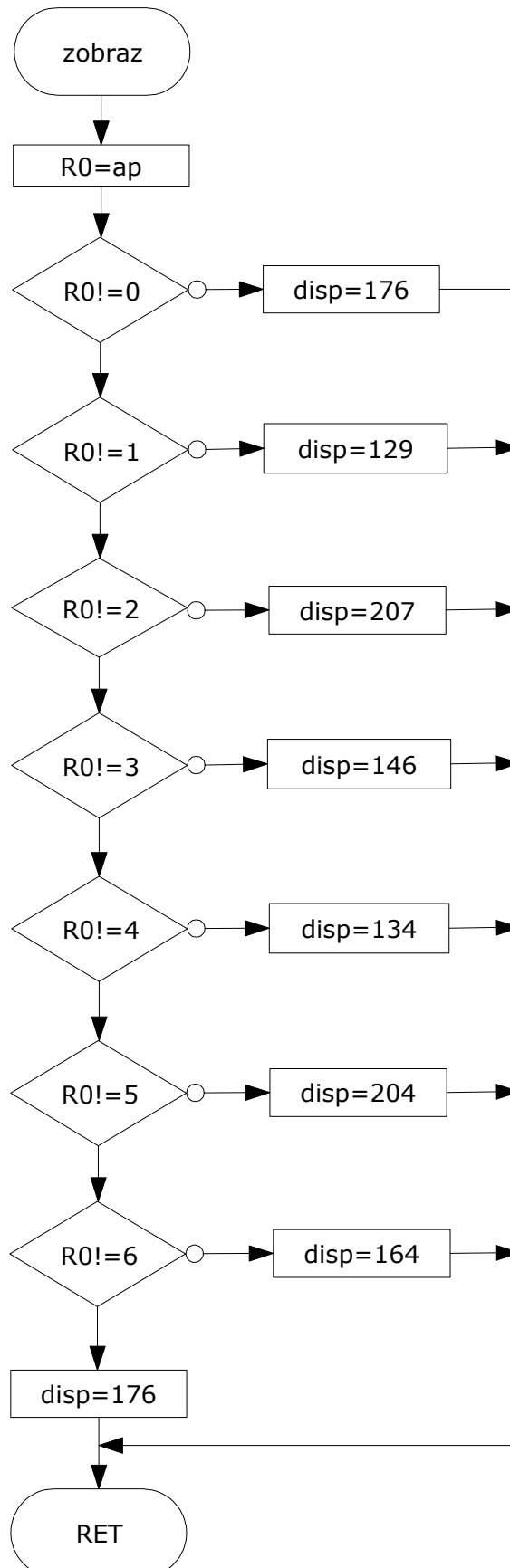
9.3.1. Vývojový diagram MASTER



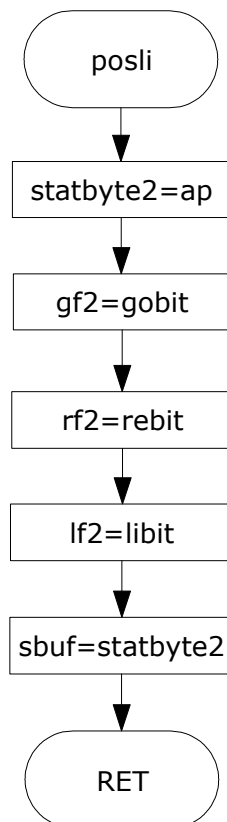
9.3.2. Rutina sentest



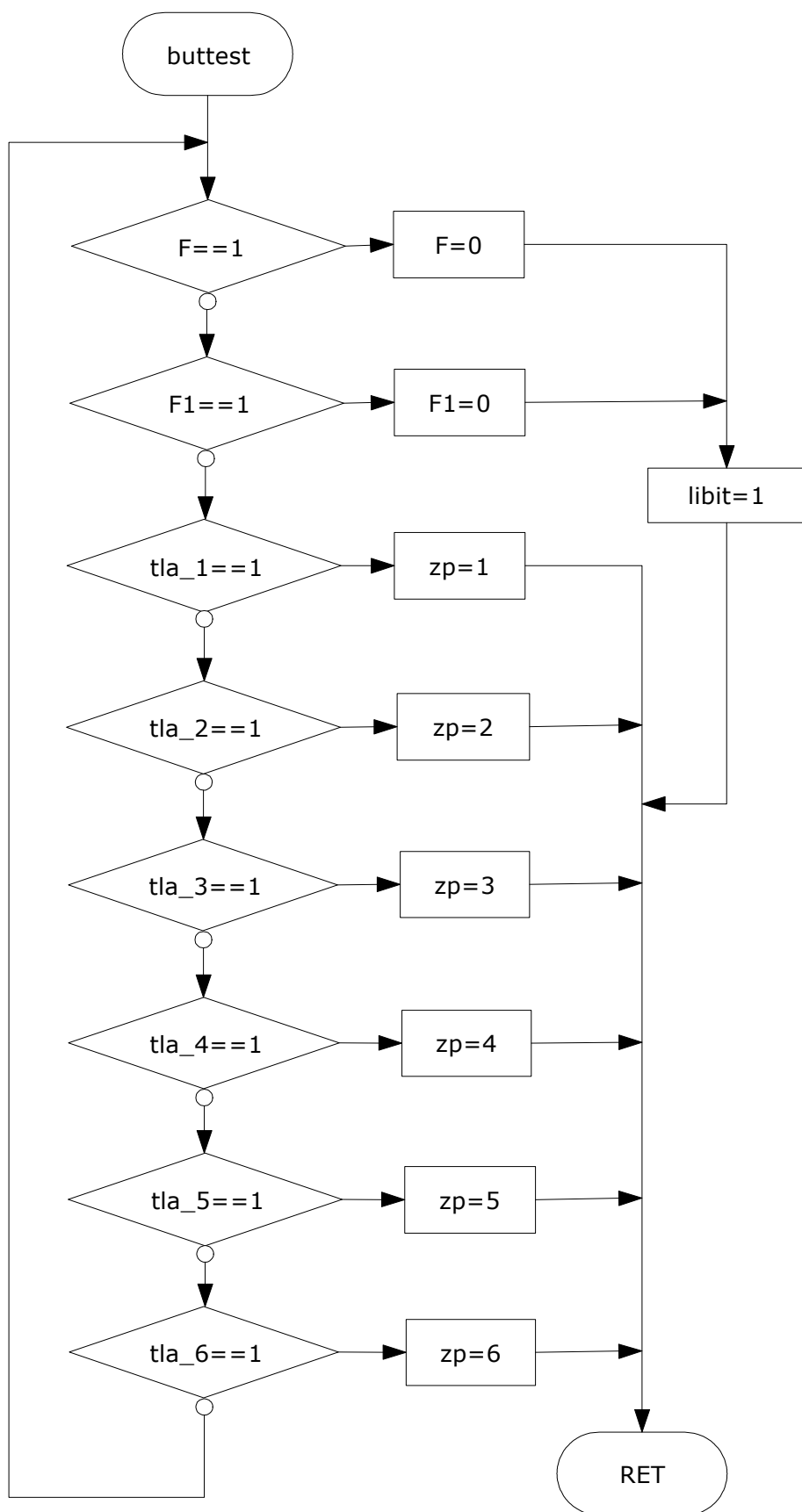
9.3.3. Rutina zobraz



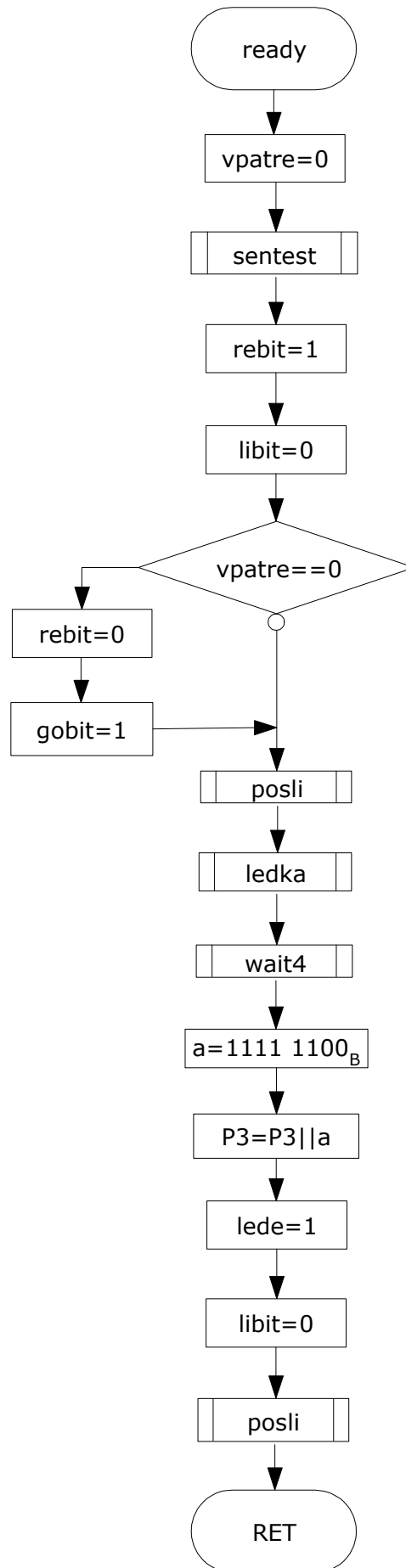
9.3.4. Rutina posli (MASTER)



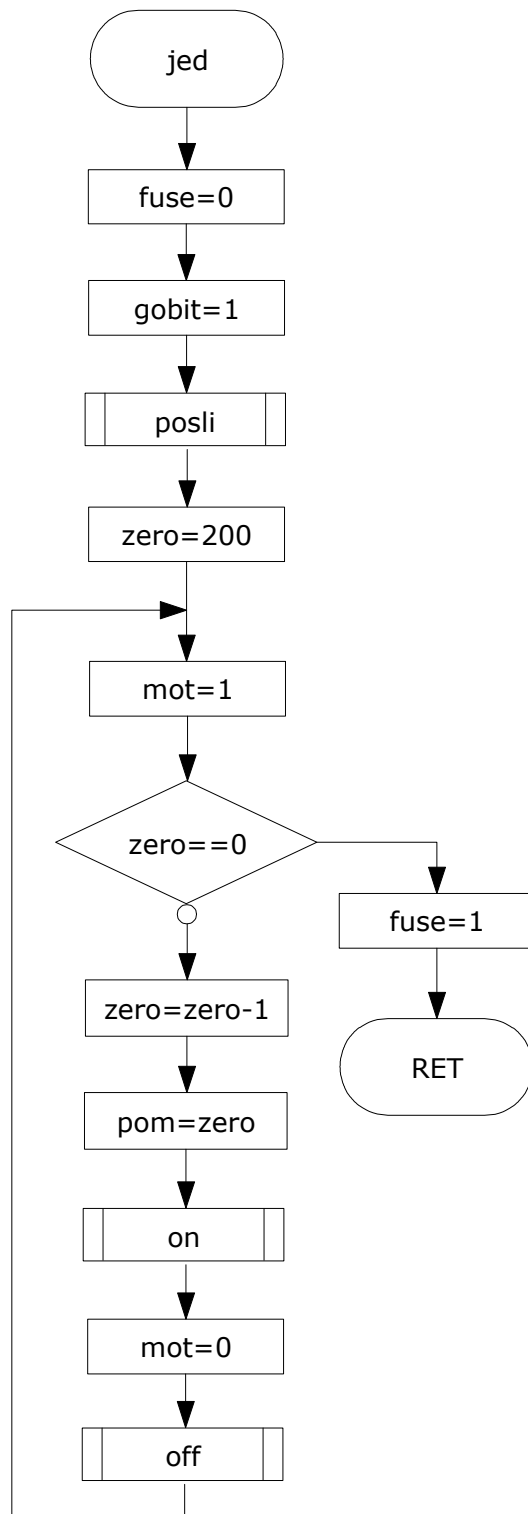
9.3.5. Rutina buttest (MASTER)



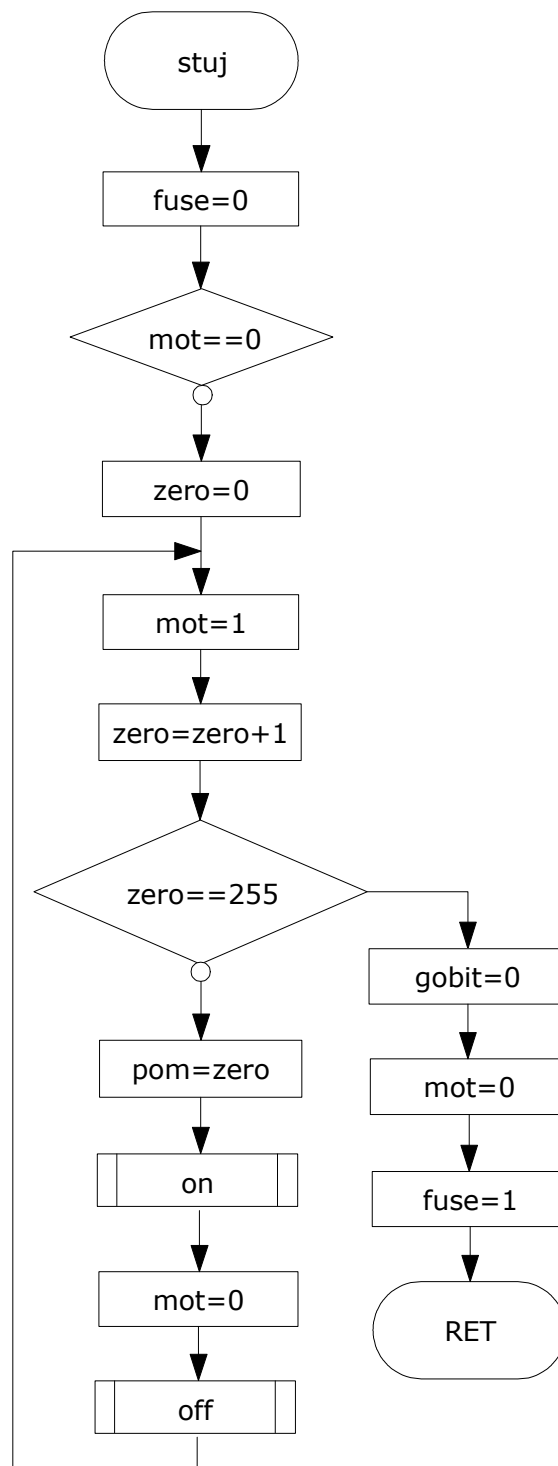
9.3.7. Rutina ready



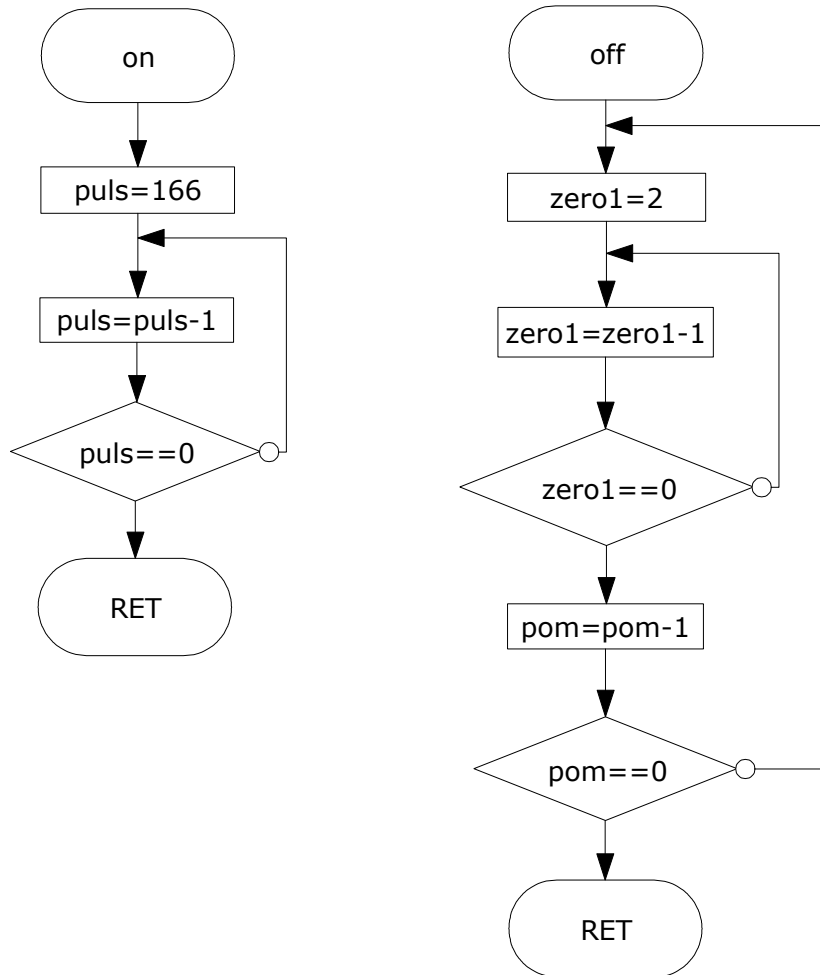
9.3.8.Subrutina jed



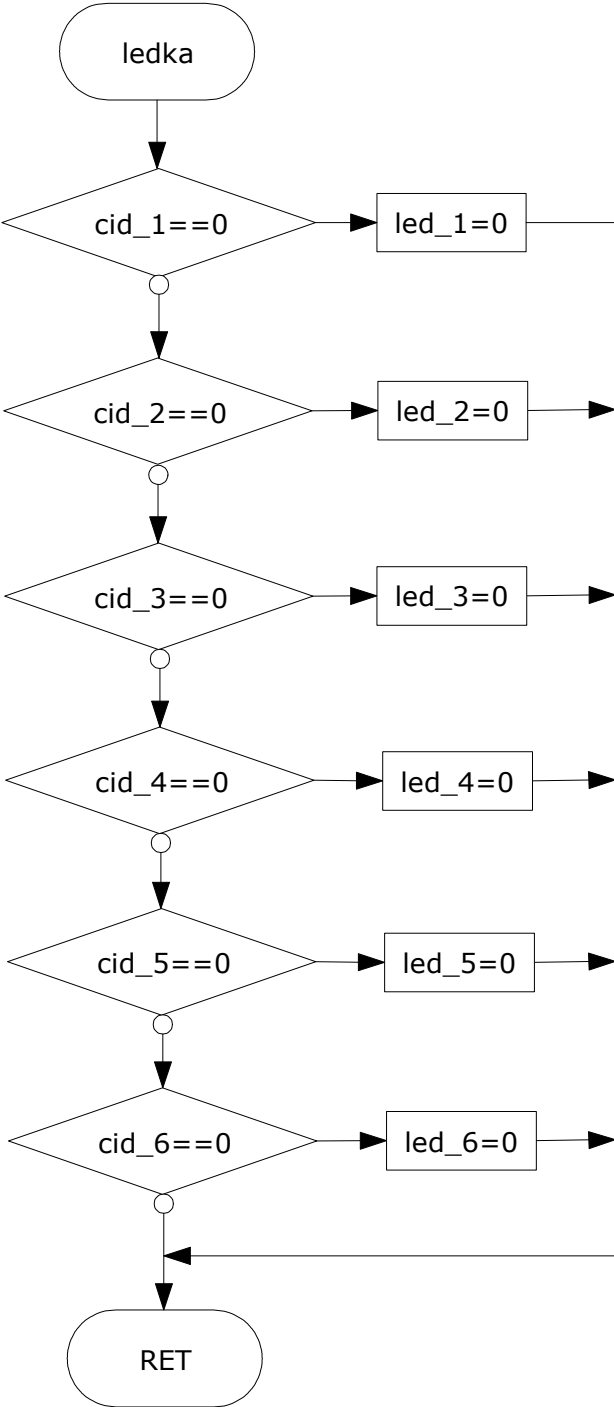
9.3.9.Subrutina stuj



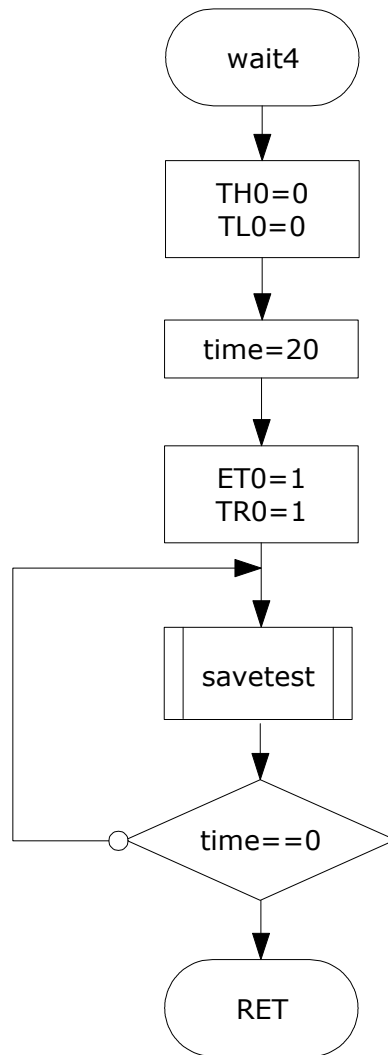
9.3.10.Subrutiny on a off



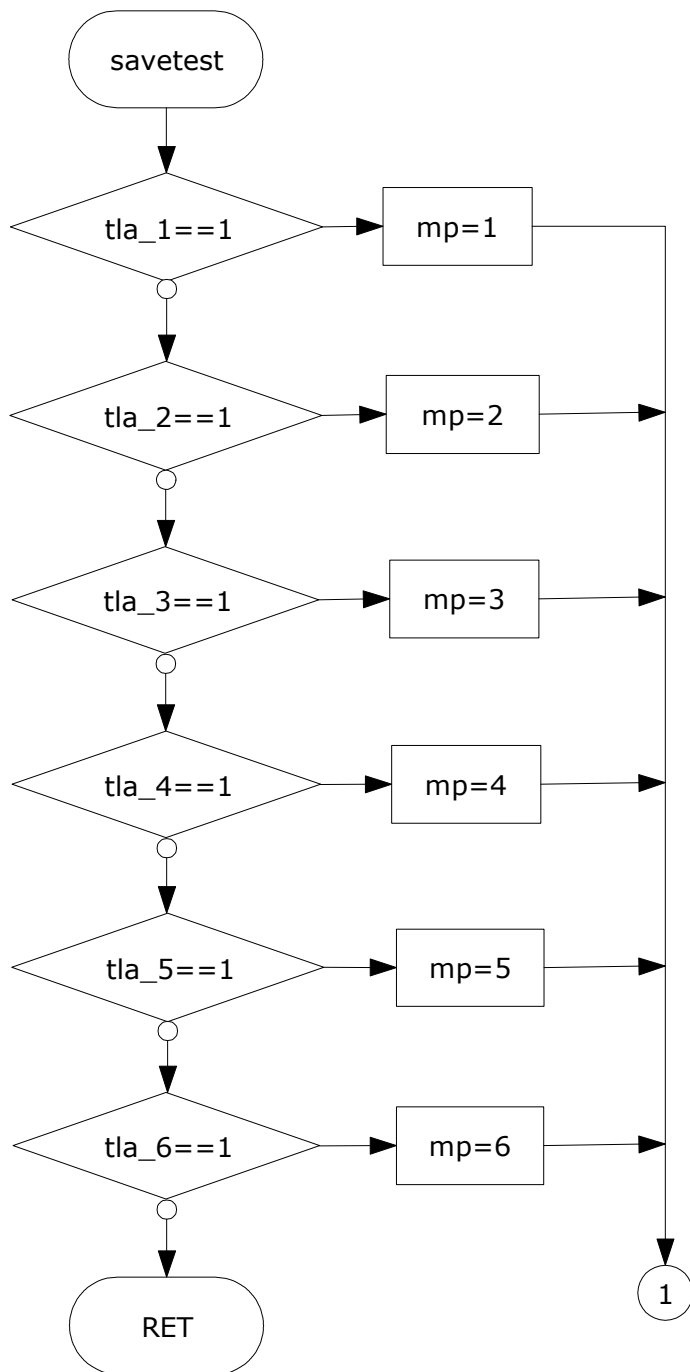
9.3.11.Subrutina ledka



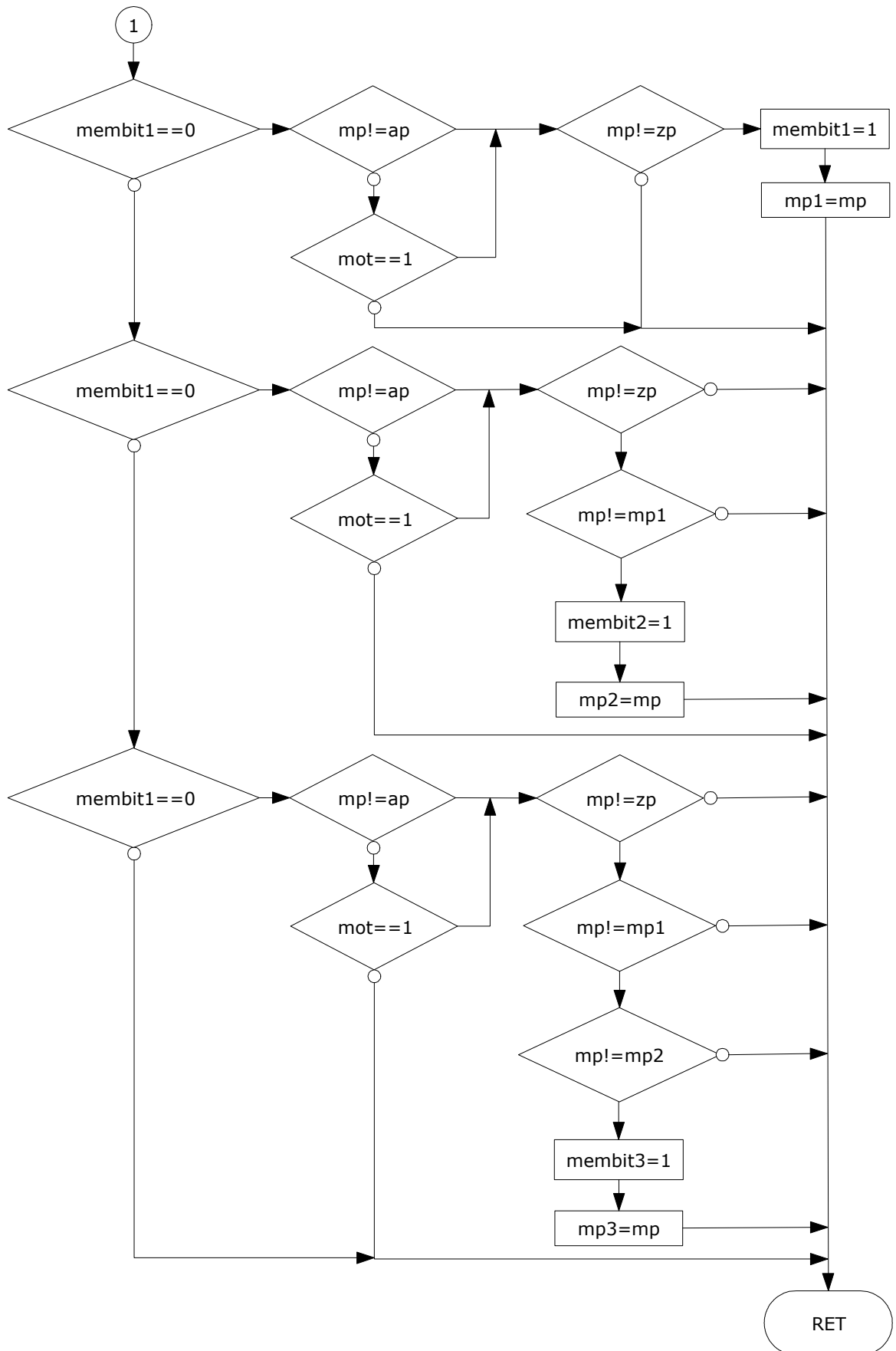
9.3.12.Subrutina wait4



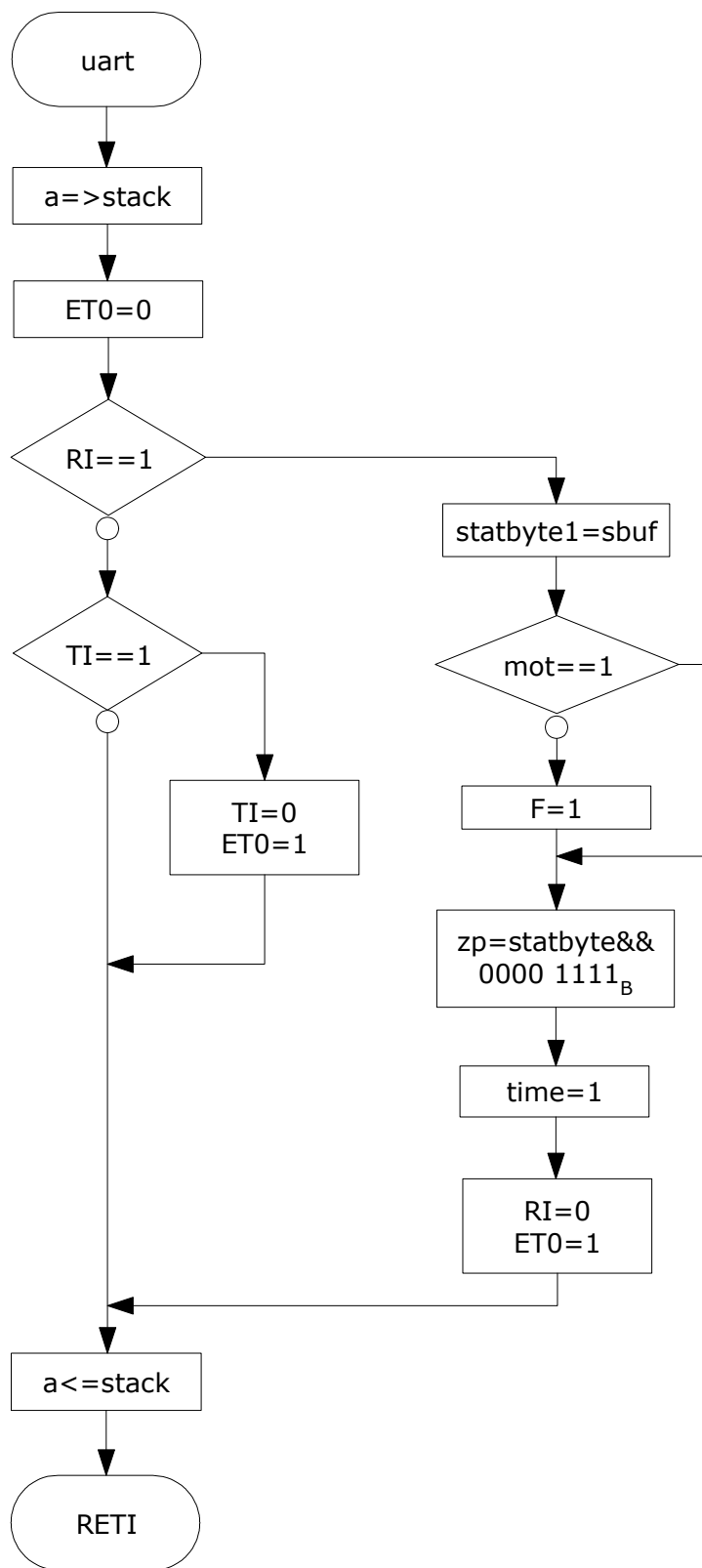
9.3.13.Subrutina savetest – 1.část



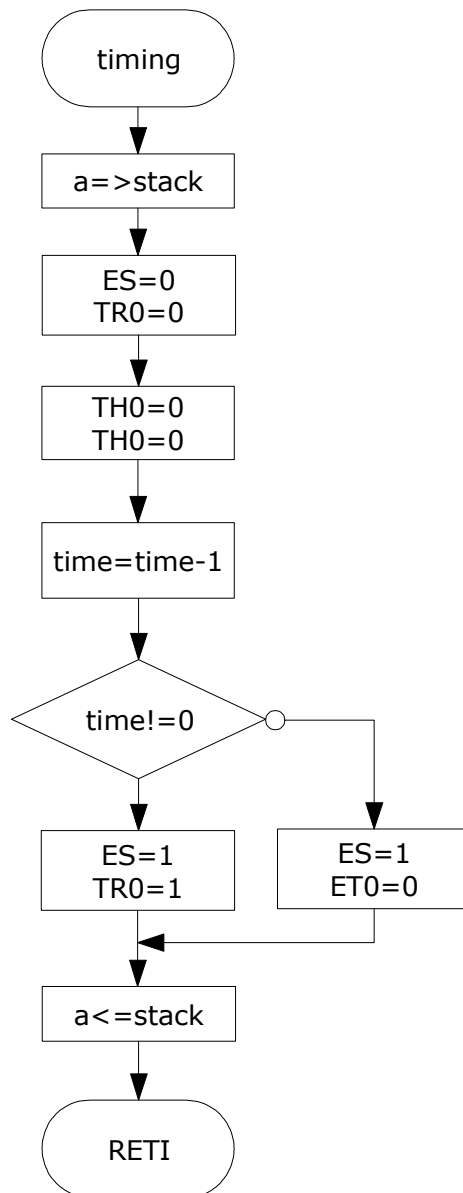
9.3.14.Subrutina savetest – 2.část



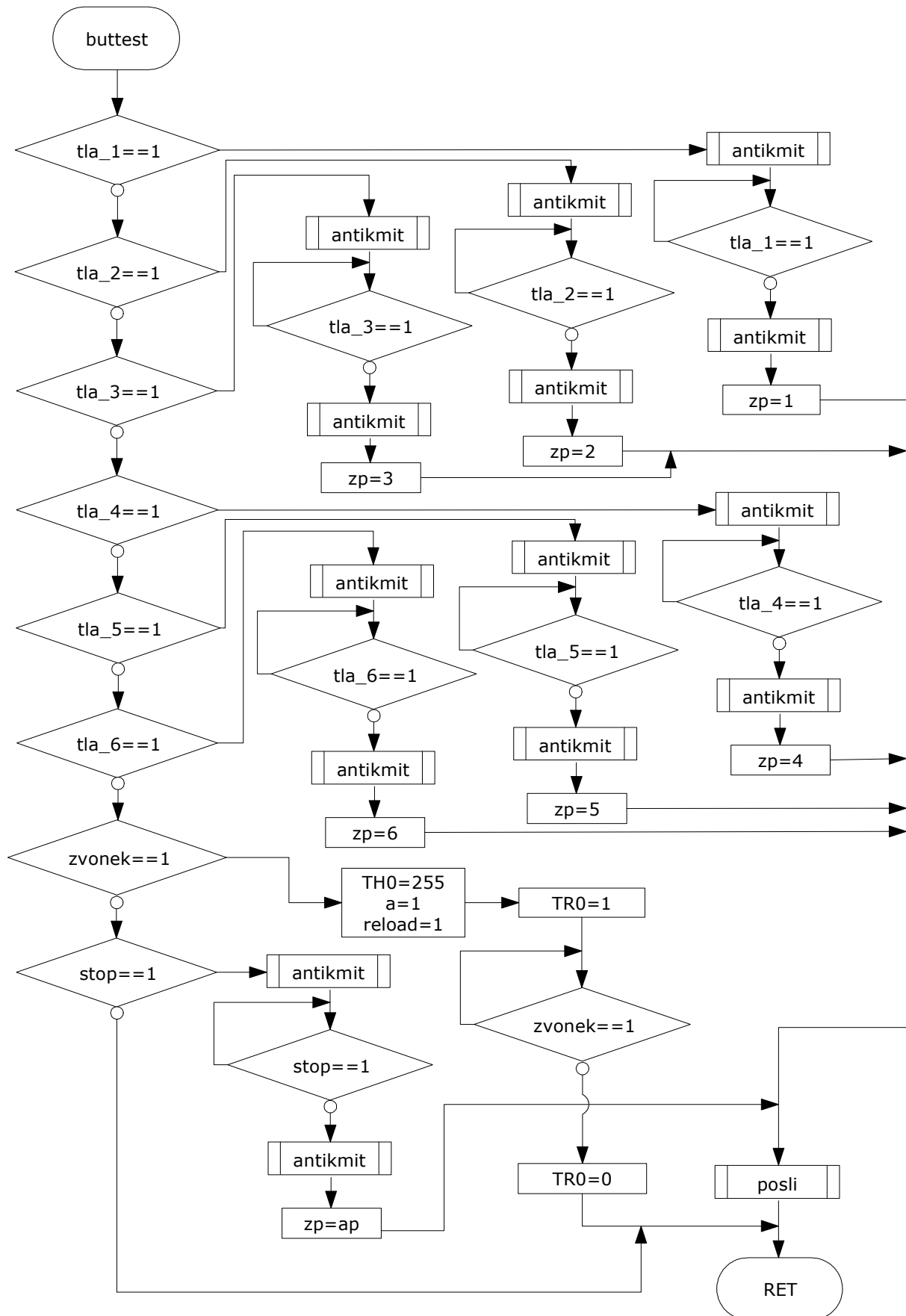
9.3.15. Obsluha přerušeni sériové linky (MASTER)



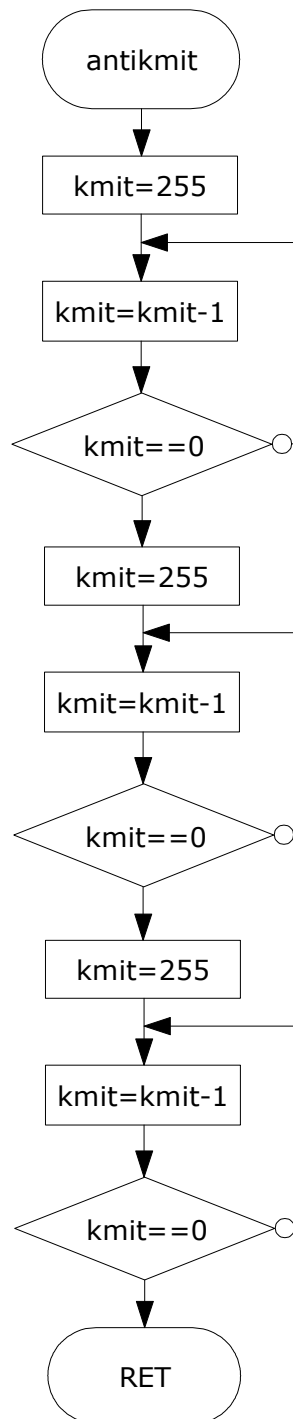
9.3.16. Obsluha přerušení timeru 0 (MASTER)



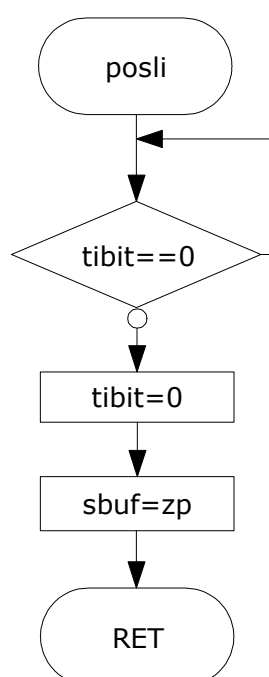
9.3.17. Rutina Buttest (SLAVE)



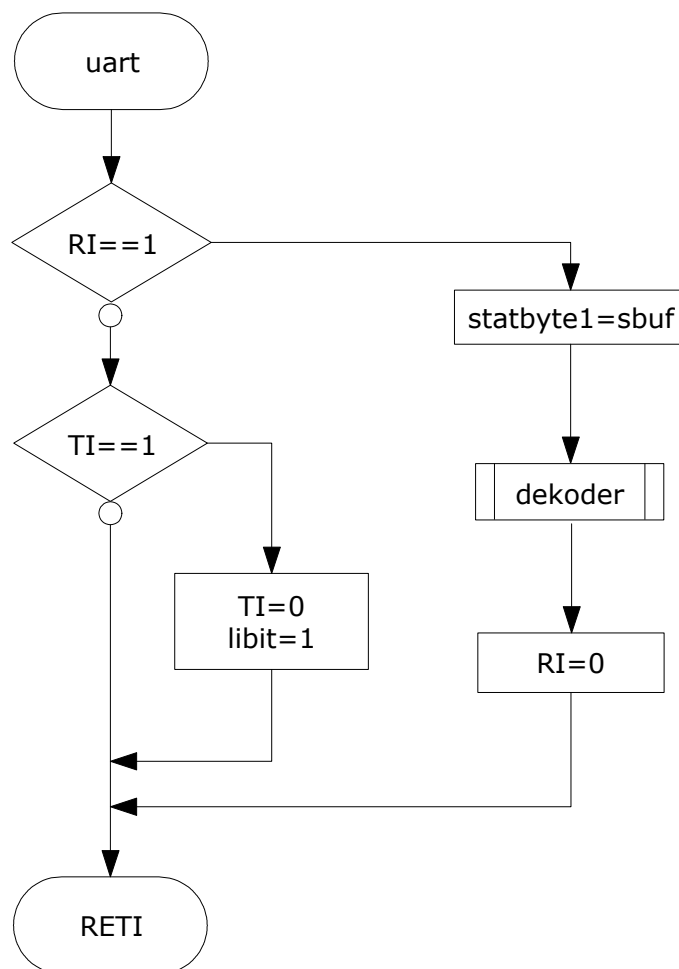
9.3.18.Subrutina antikmit



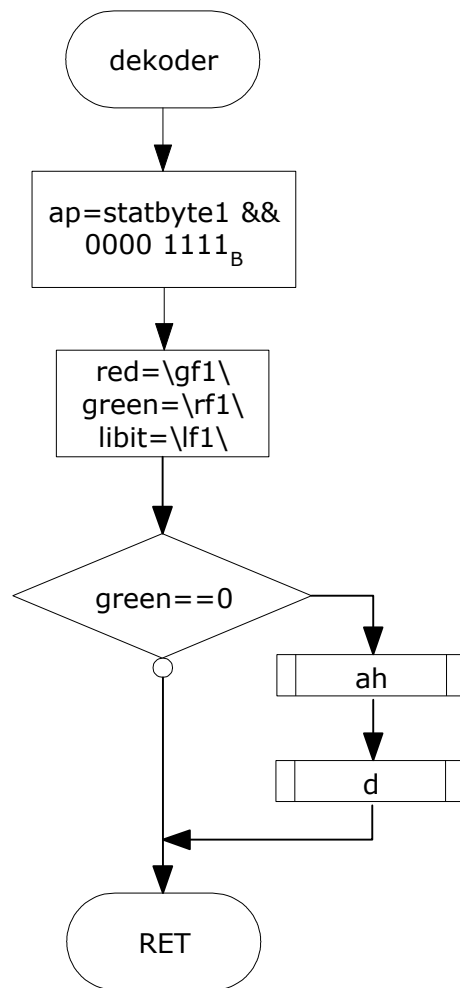
9.3.19.Subrutina posli (SLAVE)



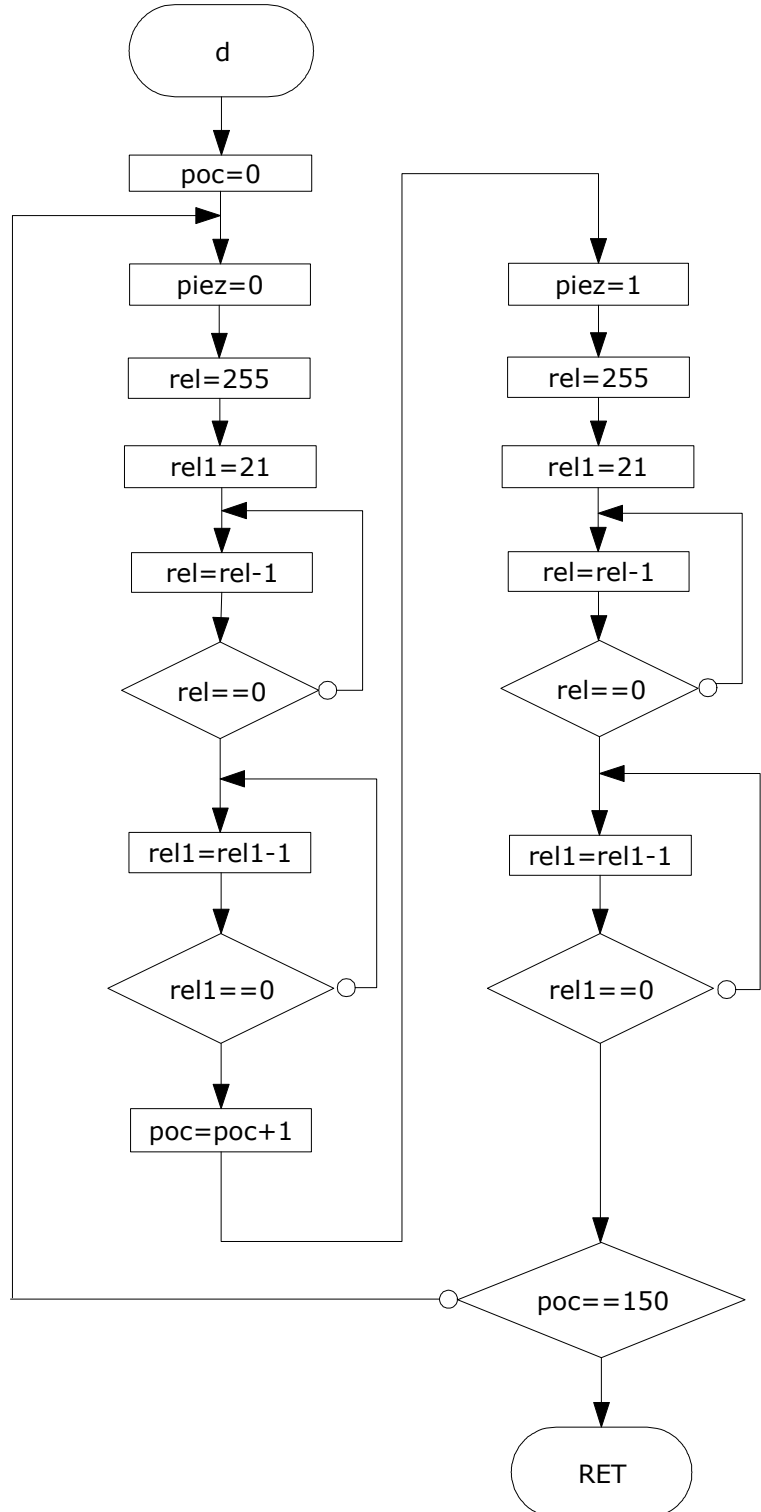
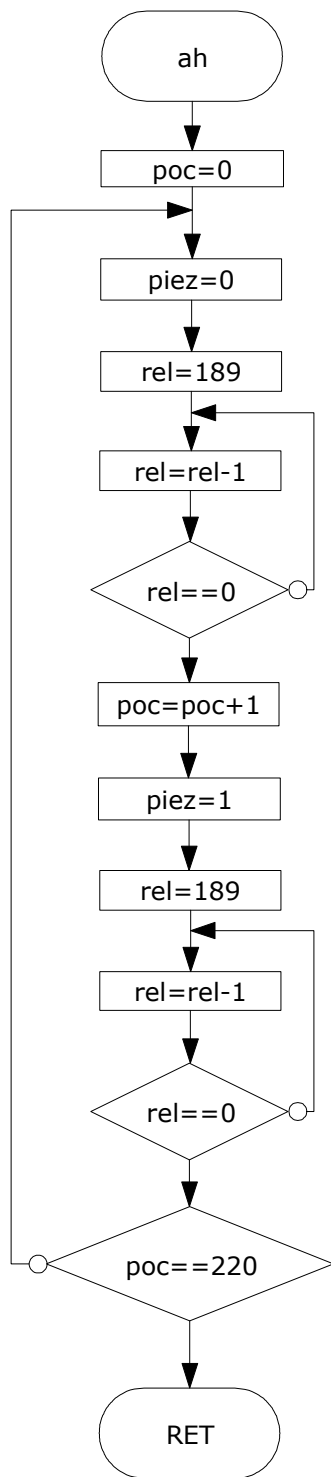
9.3.20. Obsluha přerušeni sériové linky (SLAVE)



9.3.21.Subrutina dekode



9.3.22.Subrutiny ah a d



9.3.23. Obsluha přerušení timeru 0 (SLAVE)

