Středoškolská technika 2011

Setkání a prezentace prací středoškolských studentů na ČVUT

INFORMAČNÍ LED DISPLAY

Martin Vitásek

Střední průmyslová škola elektrotechniky a informatiky
Kratochvílova, 7/1490, Ostrava - Moravská Ostrava

Středoškolská odborná činnost 2010/2011

Obor 10 – elektrotechnika, elektronika a telekomunikace

Informační LED displej

Autor:
Martin Vitásek, 2. ročník
SPŠ elektrotechniky a informatiky
Kratochvílova 7
702 00 Ostrava

Konzultant práce:
Ing. Karel Gogolka
(SPŠEI, Ostrava)

Ostrava, 2011

STŘEDNÍ PRŮMYSLOVÁ ŠKOLA ELEKTROTECHNIKY A INFORMATIKY

Prohlášení

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

V Ostravě dne:…… ……

podpis autora

OBSAH

1. ÚVOD

1.1 Zdůvodnění tématu

Každým rokem u nás na škole probíhá SOČ. Protože se zajímám o programování mikroprocesorů a jejich využití v praxi, rozhodl jsem se sestavit informační LED displej řízený mikroprocesorem a přihlásit se do SOČ. Nápad na vytvoření informačního displeje vznikl, když jsem rozebíral staré domácí kino, ze kterého jsem vykuchal pětimístný alfanumerický LED displej. Cílem práce bylo vytvořit kompletní návrh elektronického zařízení – informační LED displej – a realizovat jeho výrobu. Hlavním důvodem práce ale zůstává vyzkoušet si v praxi programování mikroprocesoru v jazyku C, seznámit se se sériovou sběrnicí SPI, a získat zkušenosti s ovládáním periferních obvodů mikroprocesoru. Mimo jiné může toto zařízení díky vyvedenému ICSP konektoru sloužit také jako výukový prostředek pro předmět mikroprocesorová technika.

1.2 Specifikace problému

Toto elektronické zařízení řeší hlavně problém řídícího softwaru a také elektroniky pro řízení LED displeje. Z hlediska softwaru se jedná o správné načasování jednotlivých operací, vytvoření efektu posunujícího se textu, ukládání dat do paměti EEPROM, překlad ASCII znaků pro displej, zajištění komunikace s řadičem displeje po sběrnici SPI a snímaní stavu tlačítek. Z hlediska elektroniky se pak jedná o stabilizovaný zdroj pro mikroprocesor a displej a hlavně o buzení LED displeje. Pro předvedení funkčnosti tohoto systému bylo celé zapojení umístěno do plastové krabičky. Celek funkčního zařízení je tedy tvořen elektronickou, konstrukční a softwarovou částí.

1.3 Použité metody

Při realizaci zařízení byl využit řadič LED displeje, který jednak zmenšil rozměry celého systému a také výrazně snížil vytížení řídícího mikroprocesoru. Také byl použit integrovaný stabilizátor napětí pro stejnosměrné zdroje a mikroprocesor od firmy Microchip. Z hlediska softwaru pak bylo využito přerušení, jak pro snímání stavu tlačítek, tak pro časování jednotlivých operací, čekací smyčky, pole hodnot v programové paměti pro překlad ASCII znaků a další.

1.4 Organizace práce

Práce začala sestavením elektroniky na nepájivém kontaktním poli a vytvořením softwaru pro základní funkci displeje. Následoval návrh plošných spojů a jejich výroba. Po vyhotovení elektroniky byl celý systém umístěn do plastové krabičky. Nakonec byly provedeny drobné úpravy vzhledu a dolaďován software pro řízení displeje.

Práce obsahuje:

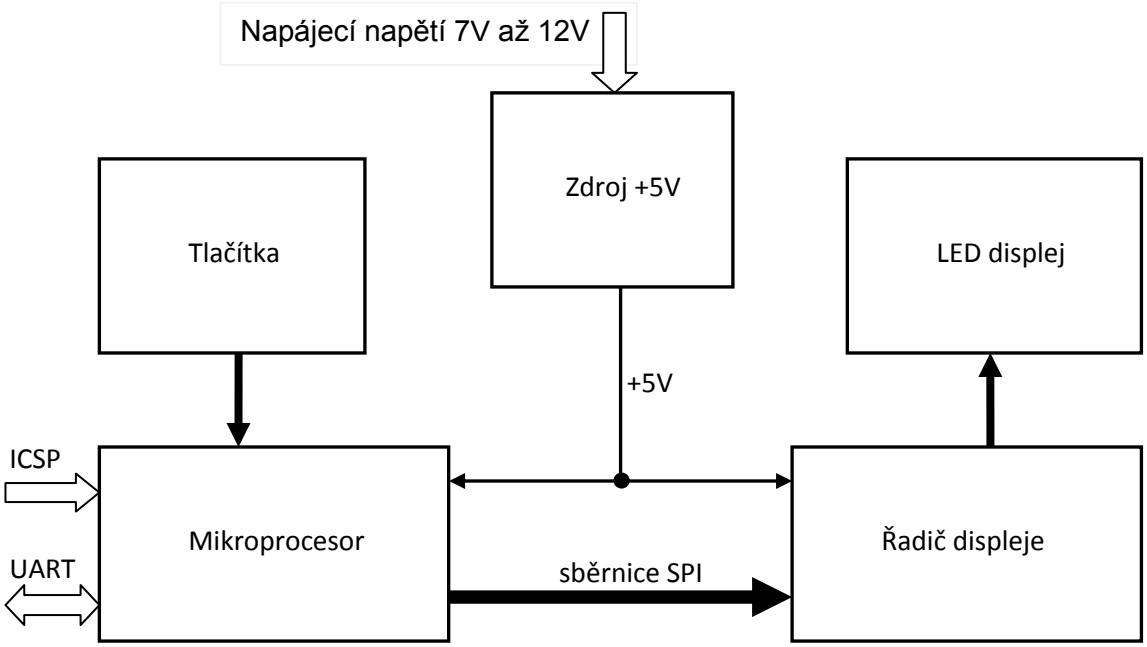
- Popis funkce systému
- Popis hardwarové části
- Popis softwarové části

2. POPIS FUNKCE

Systém je vybaven pětimístným LED displejem a dvěma tlačítky. Při zapnutí se načte z paměti text zobrazovaný před posledním vypnutím a je zobrazován na displeji. Při krátkém stisku černého tlačítka se na displeji zobrazí číslo následujícího textu a začne se zobrazovat text uložený pod tímto číslem. Při držení černého tlačítka se s určitým časovým odstupem přičítá číslo textu, který chceme zobrazit, a zároveň se zobrazuje na displeji. Jakmile tlačítko pustíme, aktivuje se právě zvolený text a začne se zobrazovat na displej. Funkce červeného tlačítka je obdobná. Při krátkém stisku červeného tlačítka se na displeji zobrazí číslo předchozího textu a začne se zobrazovat text uložený pod tímto číslem. Při držení červeného tlačítka se s určitým časovým odstupem odčítá číslo textu, který chceme zobrazit, a zároveň se zobrazuje na displeji. Jakmile tlačítko pustíme, aktivuje se právě zvolený text a začne se zobrazovat na displej.

3. ELEKTRONIKA

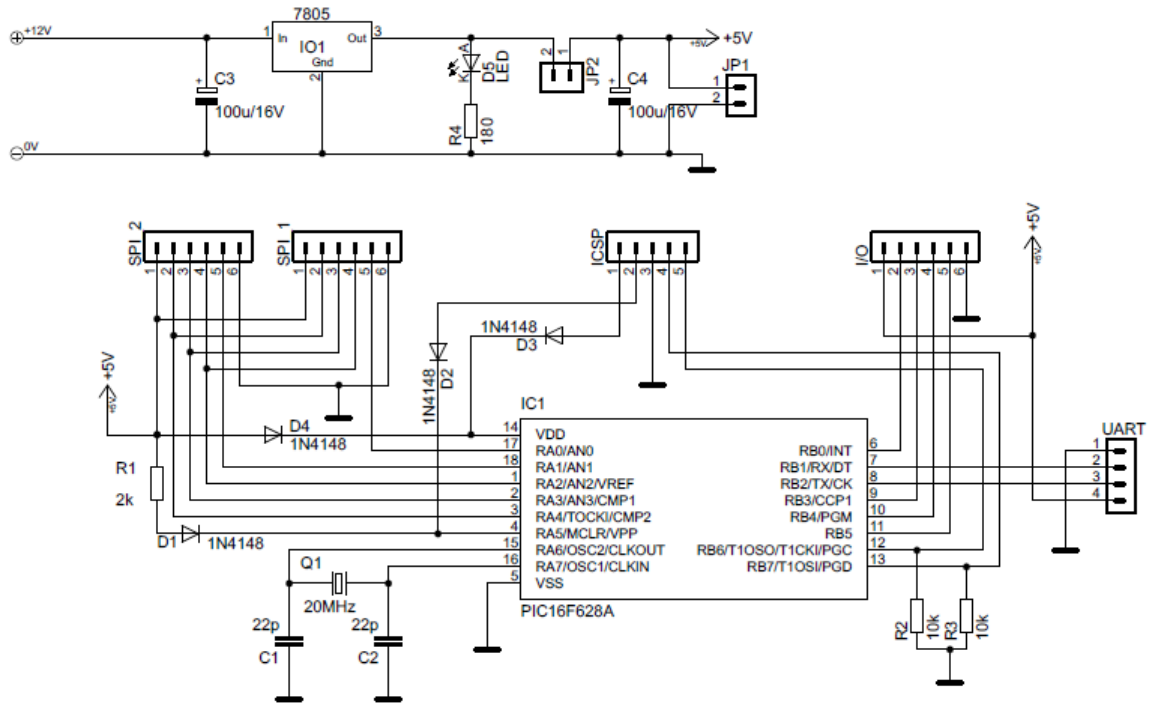
Elektronika je rozdělena do několika funkčních bloků jak můžete vidět na blokovém schématu na obr. 1. Desky plošných spojů jsou jednostranné a byly navrhnuty ve free verzi programu Eagle. Při vývoji byl kladen důraz na rozměry desek a jejich přizpůsobení pro montáž do krabičky. Všechny plošné spoje byly vyrobeny fotocestou. Pro propojení desek plošných spojů byly použity vícežílové ploché kabely. Celá elektronika se skládá ze tří desek a byla navržena tak, že má možnost snadného rozšíření.



Obr. 1- blokové schéma systému

3.1 Řídící deska

Řídící deska obsahuje mikroprocesor a zdroj. Zdroj je tvořen vyhlazovacím kondenzátorem a stabilizátorem napětí 7805. Stabilizátor je opatřen malým hliníkovým chladičem. Na výstupu stabilizátoru je také připojen kondenzátor, indikační LED dioda a propojka JP2, která umožňuje vyřadit zdroj při jiném způsobu napájení. Zdroj dodává stabilizované napětí 5V. Druhá část desky je osazena mikroprocesorem PIC16F628A, který běží na frekvenci 20MHz a řídí činnost celého systému – viz. obr. 2. Deska podporuje připojení dvou periferních obvodů komunikujících pomocí sběrnice SPI, možnost sériového programování v zapojení tzv. ICSP (in circuit serial programing), dále je možnost připojení zařízení komunikujícího přes sériový port v TTL úrovních a ještě jsou vyvedeny čtyři I/O porty na samostatný konektor. Toto řešení dovoluje snadné rozšíření systému o další moduly. Diody D1 až D4 slouží k tomu, aby se toto zařízení navzájem neovlivňovalo s připojeným programátorem. Díky tomuto zapojení je možno mikroprocesor naprogramovat i bez připojení napájecího napětí ze zdroje, protože mikroprocesor je při vypnutém systému napájen přímo z programátoru. Rezistory R2 a R3 pak udržují log. 0 při nepřipojeném programátoru na portech RB6 a RB7.



Obr. 2- schéma řídicí desky

Zapojení konektorů:

SPI:
1 – +5V
2 – DIN (data in)
3 – DOUT (data out)
4 – CLK (clock)
5 – STB (strobe)
6 – GND

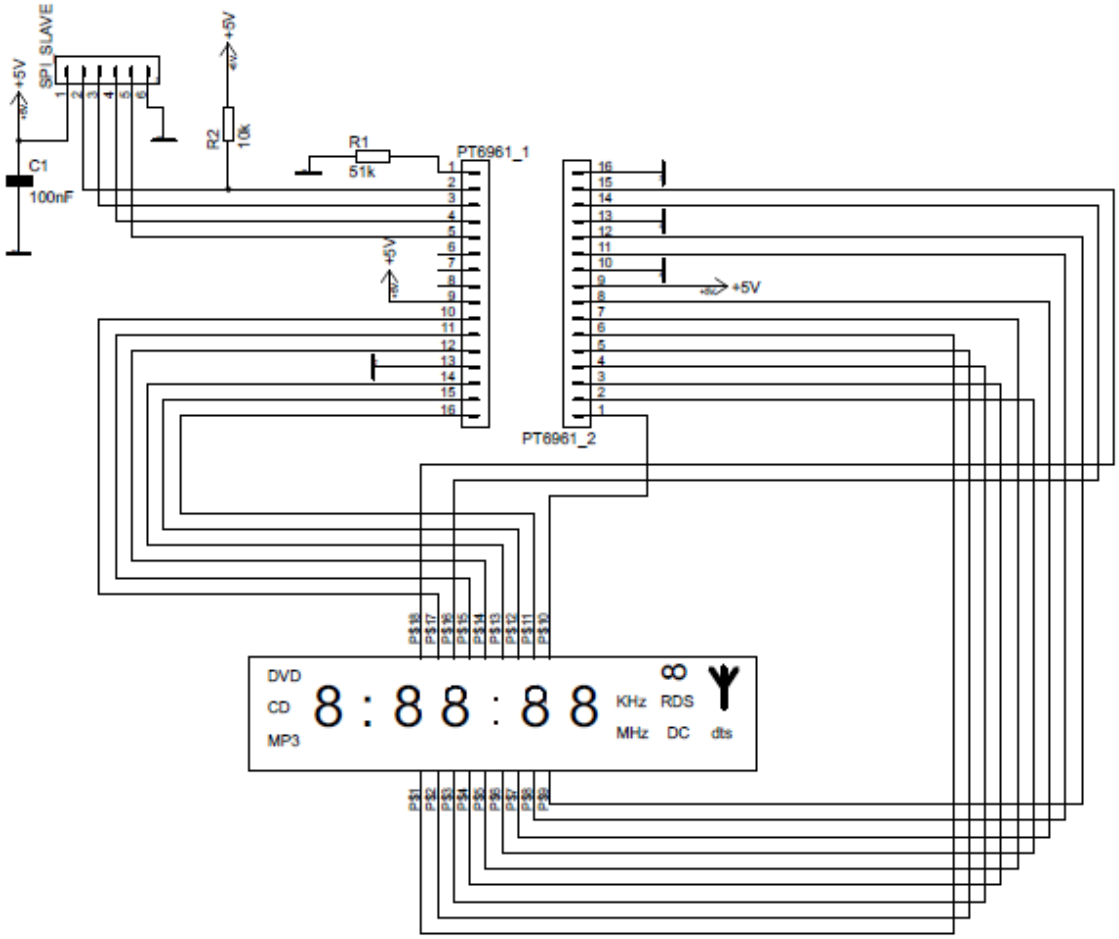
ICSP:
1 – VDD
2 – VPP
3 – GND
4 – Data
5 – Clock

I/O:
1 – +5V
2 – RB0
3 – RB3
4 – RB4
5 – RB5
6 – GND

UART:
1 – GND
2 – RX
3 – TX
4 – +5V

3.2 Deska displeje

Na desce displeje je umístěn samotný LED displej, blokovací kondenzátor a dva rezistory. Řadič displeje PT6961 je umístěn mimo desku na malém plošném spoji z důvodu omezeného prostoru krabičky a je připojen k desce vícežilovým kabelem. Schéma zapojení řadiče je podobné jako doporučené zapojení v dokumentaci integrovaného obvodu PT6961 - viz. obr. 3. Rezistor R1 určuje frekvenci vnitřního oscilátoru řadiče a rezistor R2 zajišťuje na lince DOUT (data out) log. 1 při klidovém stavu na sběrnici. Tato deska je připojena do konektoru SPI_2 na řídicí desce.

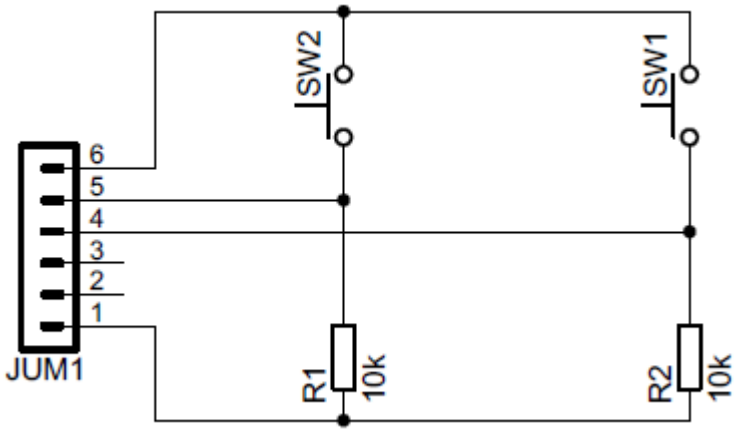


Obr. 3- schéma desky displeje

Zapojení konektoru SPI_slave:
1 – +5V
2 – DOUT (data out)
3 – DIN (data in)
4 – CLK (clock)
5 – STB (strobe)
6 – GND

3.3 Tlačítka

Tlačítka bylo nutno umístit na samostatnou destičku, aby mohla být připevněna k horní části krabičky. Na desce jsou umístěny dvě tlačítka a dále jen pull-up rezistory, které zajišťují log. 1 na výstupu při nestisknutém tlačítku viz. obr. 4. K řídicí desce jsou tlačítka připojeny do konektoru I/O.



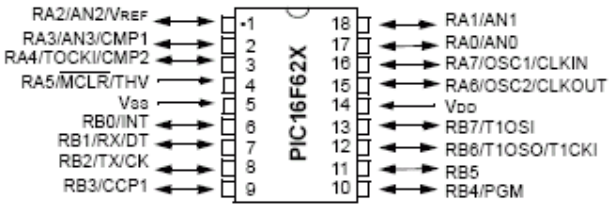
Obr. 4- schéma zapojení desky s tlačítky

Zapojení konektoru JUM1: 1 – +5V
2 – nepřipojen
3 – nepřipojen
4 – výstup tlačítko 1
5 – výstup tlačítko 2
6 – GND

3. 4 Charakteristika použitých součástek

3.4.1 Mikroprocesor PIC16F628A

PIC16F628A je osmibitový mikroprocesor patřící do rodiny PIC16FXX. Je vyroben technologii CMOS, je založen na rozšířené architektuře RISC (Reduced Instruction Set) a má oddělenou datovou a programovou paměť (Harvardská architektura). Konkrétnější informace o mikroprocesoru PIC16F628A jsou uvedeny v tab. 1.



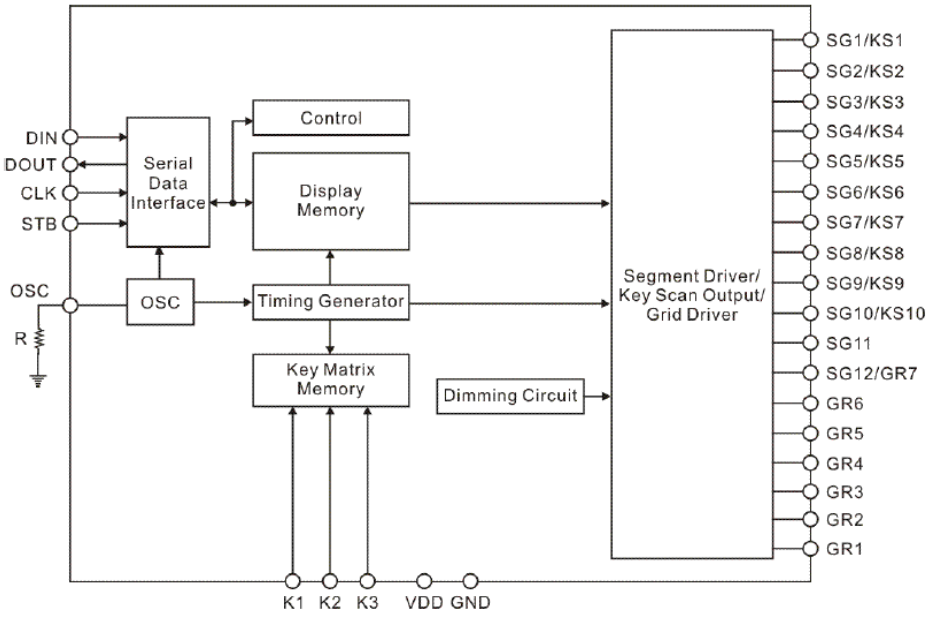
Obr. 5- Rozmístění vývodů mikroprocesoru PIC16F628A

Velikost programové paměti	2048 slov
Velikost datové paměti RAM	224B
Velikost paměti EEPROM	128B
Počet instrukcí	35
Počet I/O vývodů	16
Počet časovačů 8/16-bitových	2/1
USART	1
PWM	1
Komparátor	2
Maximální frekvence	20MHz
Napájecí napětí	3V až 5,5V
Zatížitelnost portů	20mA
Odebíraný proud	< 1uA – 3V standby 15uA – 3V, 32 kHz < 2mA – 5V, 4 MHz
Další vybavení	Power-on Reset Power-up Timer Oscilator Start-up Timer Watchdog Timer Code-protection SLEEP mode

Tab. 1- základní údaje o mikroprocesoru PIC16F628A

3.4.2 Řadič PT6961

Řadič PT6961 je periferní integrovaný obvod, který komunikuje s mikroprocesorem pomocí sběrnice SPI. Tento obvod budí LED displej v multiplexním režimu. Může také snímat stav třiceti tlačítek, ale této funkce v tomto projektu není využito. Viz. blokové schéma vnitřního zapojení na obr. 6.

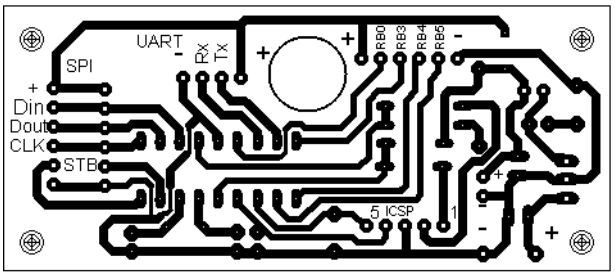


Obr. 6- vnitřní zapojení integrovaného obvodu PT6961

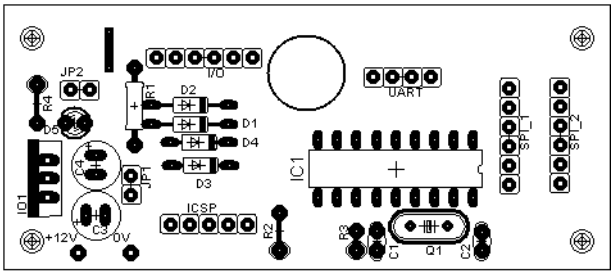
4. PŘEDLOHY DESEK PLOŠNÝCH SPOJŮ

Vytvořeno v programu Eagle 5.10.

4.1 Řídící deska

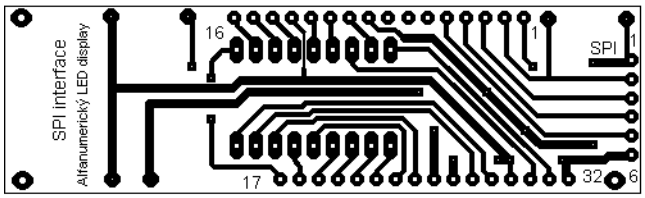


Obr. 7- předloha plošného spoje řídicí desky

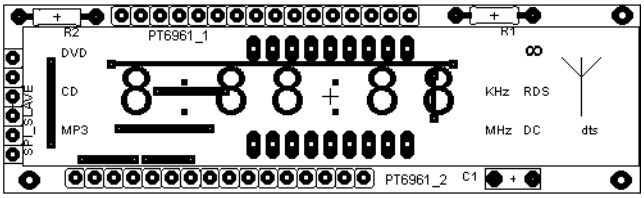


Obr. 8- osazovací plán řídicí desky

4.2 Deska displeje

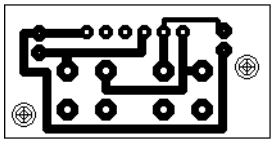


Obr. 9- předloha plošného spoje pro displej

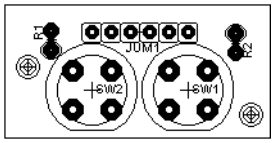


Obr. 10- osazovací plán desky displeje

4.3 Deska tlačítek



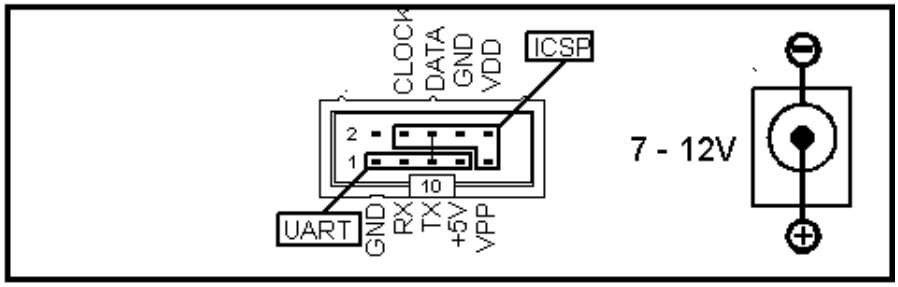
Obr. 11- předloha plošného spoje pro tlačítka



Obr. 12- osazovací plán desky tlačítek

5. KONSTRUKCE

Celá elektronika byla umístěna do plastové krabičky o rozměrech 90x65x35mm. Řídící deska je upevněna v zadní části krabičky pomocí dvou šroubů. Deska displeje je umístěna v přední části krabičky a deska s tlačítky je upevněna na horní kryt. Přední panel tvoří poloprůhledné plastové zrcadlo, za kterým je umístěn LED displej. Na zadním panelu je napájecí konektor a konektor pro připojení programátoru nebo zařízení komunikujícího přes UART- viz. obr. 13.



Obr. 13- Zadní panel

6. SOFTWARE

Software byl napsán v jazyku C ve free verzi vývojového prostředí MikroC PRO for PIC (verze 3.2). Pro naprogramování mikroprocesoru byl použit klon programátoru PICkit 2. Software je ještě ve vývoji, ale aktuální verze softwaru je plně funkční. Současná verze softwaru nepodporuje komunikaci přes jednotku UART a pro změnu textu je nutno ve zdrojovém kódu přepsat text a mikroprocesor přeprogramovat. Celý kód byl pro větší přehlednost rozdělen do čtyř souborů, které budou v následujících podkapitolách popsány.

6.1 Soubor LED_display_main.c

V tomto souboru je kód, který zajišťuje komplexní funkci systému. To je nastavení výchozích hodnot při spuštění, zajištění zobrazování textu, detekce a obsluha tlačítek a obsluha přerušení.

6.1.1 Inicializace systému

Při zapnutí systému se nastaví vstupní a výstupní porty a nastaví se výchozí hodnoty registrů. Dále se zavolá funkce pro inicializaci řadiče displeje, přečte se z paměti EEPROM číslo naposledy zobrazovaného textu před vypnutím. Následně se nastaví přerušení od časovače TMR0 a od změny na portu B, kde jsou připojena tlačítka. Poté se v registru OPTION_REG nastaví přiřazení předděliče pro časovač TMR0 a nastaví se jeho dělící poměr na hodnotu 1:2. Nakonec se zapíše hodnota do registru TMR0 a předá se řízení hlavní smyčce programu.

6.1.2 Zobrazování posunujícího se textu

Z hlavní smyčky programu je volána funkce, ve které se testuje hodnota registru pocitadlo, jehož hodnota je inkrementována v obsluze přerušení od TMR0. Pokud je hodnota tohoto registru vyšší než hodnota 25xTEXT_SPEED, kde TEXT_SPEED je prodleva mezi obnovou displeje (díky vynásobení hodnoty TEXT_SPEED dvaceti pěti se dostane čas zpoždění v milisekundách), dojde k vynulování registru pocitadlo a následnému zavolaní funkce pro obnovu dat na displeji.

6.1.3 Obsluha tlačítek

Funkce pro obsluhu tlačítek je taktéž volána z hlavní smyčky programu. Jako stisk tlačítka je považována situace, kdy registr blacktl pro černé tlačítko nebo registr redtl pro červené tlačítko je roven 1 (tyto registry jsou nastavovány v obsluze přerušení od tlačítek) a zároveň je-li na vstupu daného tlačítka log. 0. Díky tomuto řešení jsou eliminovány zákmity tlačítek. Při indikaci stisku černého tlačítka dojde k přičtení hodnoty jedna do registru cislo_textu (tento registr obsahuje číslo právě zobrazovaného textu) a zobrazí se na displeji číslo textu, který se začne zobrazovat po opuštění této funkce. Pokud je po přičtení čísla jedna do registru cislo_textu hodnota větší než POCET_TEXTU, což je počet textových řetězců, je do registru ještě před zobrazením na displeji nahrána hodnota 1. Také se do paměti EEPROM uloží kopie registru cislo_textu. Při indikaci stisku červeného tlačítka je funkce obdobná, akorát místo přičtení se z registru cislo_textu odečte jednička, a pokud je hodnota registru cislo_textu rovna nule, nahraje se do registru hodnota POCET_TEXTU.

6.1.4 Obsluha přerušení

Při přerušení mikroprocesor přeruší aktuální činnost a zjistí jeho původ. V obsluze přerušení se testují příznaky v systémovém registru INTCON. Jestliže je zjištěno přerušení od tlačítek, vynuluje se jeho příznak a otestují se vstupy, na kterých jsou tlačítka připojena. Pokud je zjištěn stisk černého tlačítka, uloží se do registru blacktl hodnota 1 v opačném případě 0. Stejně se postupuje u červeného tlačítka, ale hodnoty se ukládají do registru redtl. Pokud je zjištěno přerušení od časovače TMR0, vynuluje se jeho příznak, inkrementuje se registr pocitadlo a obnoví se hodnota v registru TMR0.

6.2 Soubor LED_display_texty.h

V tomto souboru je definována hodnota TEXT_SPEED udávající interval obnovy dat na displeji v milisekundách a hodnota POCET_TEXTU udávající počet používaných textových řetězců. Dále jsou v tomto souboru definovány textové řetězce s textem, který bude na displeji zobrazován.

6.3 Soubor eeprom_rw.h

Soubor obsahuje funkci pro čtení z paměti EEPROM a zápis do paměti EEPROM. Čtení z paměti EEPROM proběhne zápisem adresy do registru EEADR a následným přečtením dat z registru EEDATA. Zápis už je o něco složitější. Je třeba si uvědomit, že se jedná o paměť EEPROM, do které trvá zápis i několik milisekund. Proto se nejprve otestuje, zda již skončil případný předchozí zápis do paměti. Následuje přesun adresy a dat určených k zapsání do systémových registrů EEADR a EEDATA. Poté se zakáží všechna přerušení a dvěma po sobě jdoucími „klíči“ (55_{HEX} a AA_{HEX}) se odemkne zápis do paměti. Nakonec se povolí zápis a všechna přerušení. Zápis dat pak již proběhne automaticky.

6.4 Soubor LED_display_drive.h

V tomto souboru je kód zajišťující veškeré operace s displejem. To je inicializace řadiče, vytvoření efektu posunujícího se textu, načítaní dat z textového řetězce a posílaní na displej, překlad ASCII znaků atd.

6.4.1 Inicializace řadiče displeje

Před zahájením komunikace s řadičem displeje je nutno nastavit jeho výchozí hodnoty. Nejprve se do řadiče odešle bajt, který nastaví pracovní mód displeje na 7 digits, 11 segments. Následně se odešle bajt, který nastaví řadič do normálního pracovního režimu, nastaví inkrementaci adresy při zápisu do paměti a nastaví se do módu příjmu dat pro displej. Poté se odešle bajt, v němž je řadiči oznámeno, že má zapnout displej a nastavit jeho jas na maximum. Nakonec jsou do řadiče odeslána data, která zajistí zhasnutí všech segmentů displeje.

6.4.2 Funkce vytvářející efekt posunujícího textu

Celá funkce je řízena hodnotou v registru text_rotate_smerovac. Jestliže je hodnota tohoto registru rovna nule, zhasnou se všechny segmenty pro zobrazování textu a hodnota tohoto registru se nastaví na 1. Pokud je jeho hodnota rovna 1, zavolá se funkce, která vybere řetězec na základě hodnoty v registru cislo_textu, a vytáhne z vybraného textového řetězce znak nacházející se na pozici, na níž ukazuje hodnota registru index (registr index slouží k indexaci přístupu do textového řetězce). Znak je z textového řetězce zkopírován do registru character. Poté je hodnota registru character testována. Jestliže je hodnota tohoto registru rovna nule, znamená to, že textový řetězec je u konce, do registru text_rotate_smerovac je uložena hodnota 2 a do registru character je uložena mezera (ASCII kód). Pokud je hodnota registru character různá od nuly, inkrementuje se registr index. Následně je zajištěn posun všech znaků o jedno místo doleva a na místo nejvíce vpravo je vložen znak z registru character. Nakonec se zavolá funkce, ve které se přeloží ASCII znaky a přeložená data se uspořádají pro poslání do řadiče a odešlou se. Pokud je hodnota registru text_rotate_smerovac rovna 2, začnou se zprava vkládat mezery. Po vložení pěti mezer se do registru text_rotate_smerovac uloží hodnota 3, což signalizuje konec textu a vynuluje se registr index. Hodnota registru text_rotate_smerovac je automaticky vynulována po dalším zavolání této funkce. Celý cyklus se takto neustále opakuje.

6.4.3 Funkce pro přímý přístup na displej

Tato funkce je využívána pro zobrazení čísla textu při stisku tlačítka. Vstupním parametrem funkce je pět znaků v ASCII kódu. Tato funkce zajistí překlad těchto znaků a jejich odeslaní na displej.

6.4.4 Překlad znaků a seřazení dat pro displej

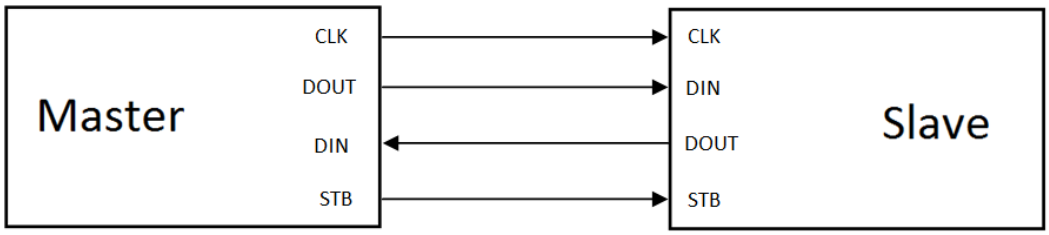
Pro zobrazení na displeji se musí přeložit znaky kódované ASCII kódem na kombinaci rozsvícených a zhasnutých segmentů. Toto se provádí algoritmem, který nejprve upraví vstupní data kódovaná v ASCII kódu a následně pomocí tabulky hodnot v programové paměti přeloží tento znak. Pokud se zjistí na vstupu znak, který na displeji nelze zobrazit, je tento znak nahrazen mezerou. Po přeložení znaků pro celý displej se data setřídí a uloží do paměti RAM, kde jsou připravena k odeslání do řadiče.

6.4.5 Odeslání dat na displej

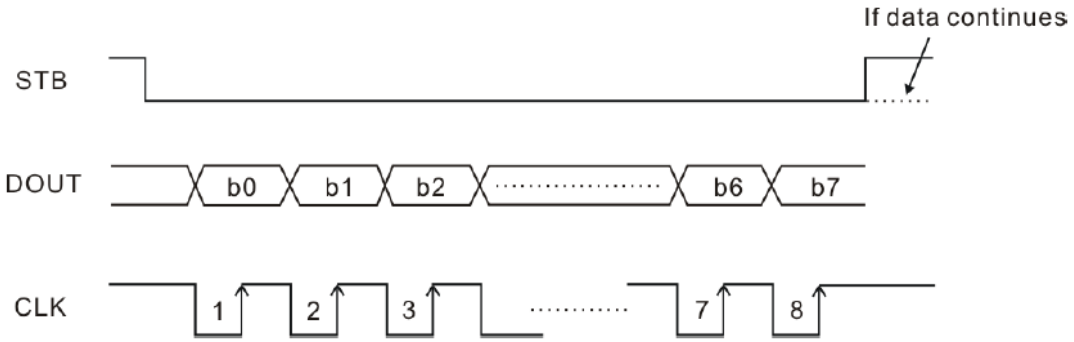
Tato funkce je velice jednoduchá. Nejprve se do řadiče pošle příkaz, aby se adresa v jeho paměti nastavila na 00_{HEX}. Následně se odešle čtrnáct bajtů dat, které jsou načteny z paměti RAM. V těchto čtrnácti bajtech je uložen stav všech segmentů displeje.

6.4.6 Softwarová sběrnice SPI

Komunikace s řadičem displeje probíhá přes sběrnici SPI (Serial Peripheral Interface). Protože mikroprocesor PIC16F628A nemá toto komunikační rozhraní implementováno hardwarově, muselo být vytvořeno softwarově. Byl vytvořen pouze software, který odesílá data do řadiče, protože není nutno z řadiče číst. Zapojení sběrnice je uvedeno na obr. 14 a časový průběh napětí při odesílání dat na jednotlivých vodičích je zobrazen na obr. 15.



Obr. 14. Zapojení sběrnice SPI



Obr. 15. Časový průběh napětí při odesílání dat do řadiče

6.4.7 Funkce pro ovládání ostatních segmentů displeje

Protože jsou na displeji doplňkové segmenty, byly vytvořeny funkce pro jejich ovládání. Současná verze softwaru ale tyto funkce nepoužívá.

7. ZÁVĚR

Výsledkem práce je podle původního plánu funkční elektronické zařízení schopné zobrazovat textové zprávy. Při realizaci jsem využil mnoho znalostí získaných hlavně samostudiem. Největším přínosem práce bylo prohloubení znalostí ve vývoji softwaru v jazyku C pro mikroprocesory a také získaní nových informací z oblasti mikroprocesorové techniky. Přestože software k tomuto systému je ještě ve vývoji, jeho aktuální verze je plně funkční. Další vývoj softwaru by se měl ubírat k doplnění komunikace přes sériový port, případně doplnění některých funkcí systému.

8. LITERATURA

Strolený, Jaroslav: Popis procesoru PIC16F627 a PIC16F628, [online], dostupné na: http://doveda.byl.cz/procesory/pic16f62x/index.htm

Novotný, Zdeněk: Škola programování PIC 6, [online], dostupné na: http://pandatron.cz/?150&skola_programovani_pic-6_dil

Novotný, Zdeněk: Škola programování PIC 12, [online], dostupné na: http://pandatron.cz/?156&skola_programovani_pic-12_dil

PRINCETON TECHNOLOGY CORP. Datasheet PT6961, [online], dostupné na: http://pdf1.alldatasheet.com/datasheet-pdf/view/391689/PTC/PT6961.html

MICROCIP. Datasheet PIC16F628A, [online], dostupné na: http://ww1.microchip.com/downloads/en/devicedoc/40044f.pdf

MIKROELEKTRONIKA. mikroC PRO for PIC, [online], dostupné na: http://www.mikroe.com/eng/products/view/7/mikroc-pro-for-pic/

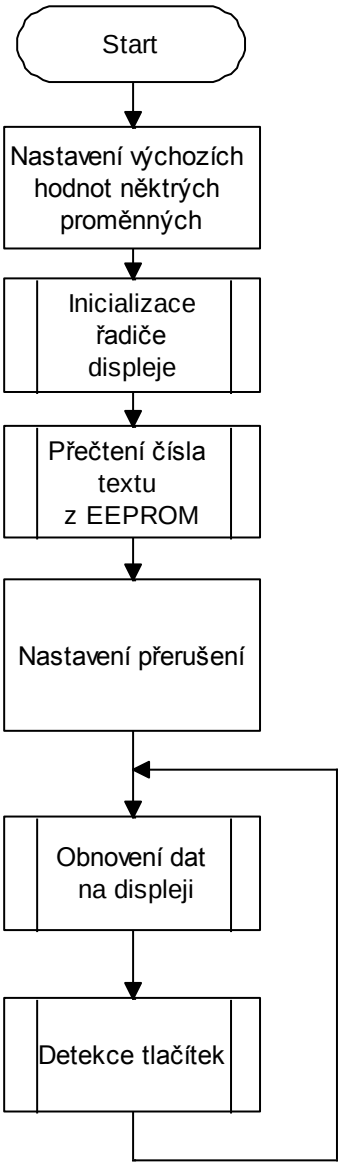
Matoušek, David: C pro mikrokontroléry Atmel AT89S52. První vydání, Praha, BEN – technická literatura, 2007. ISBN 978-80-7300-215-9.

9. PŘÍLOHY

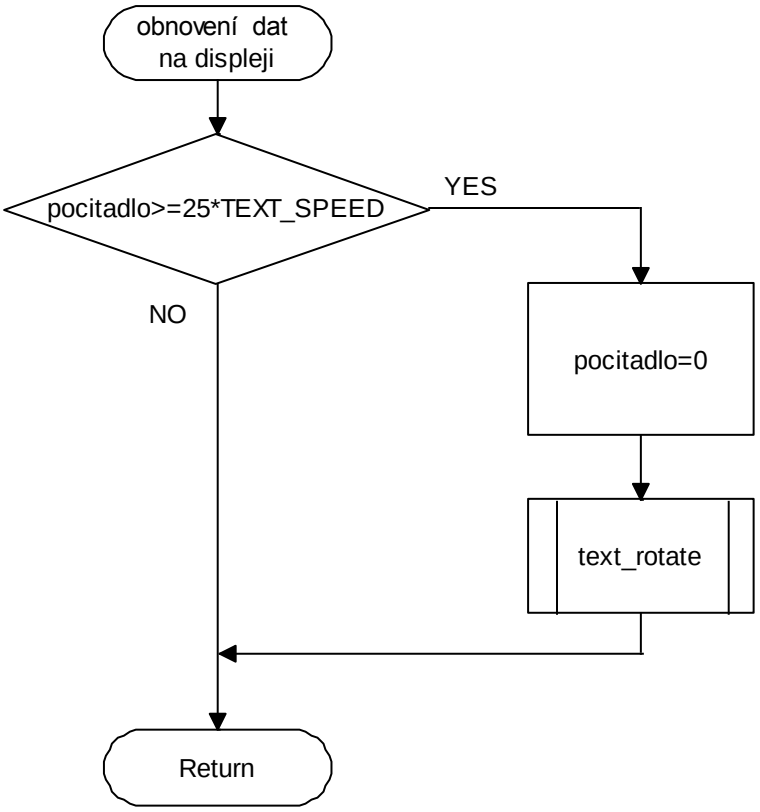
9.1 Vývojové diagramy

Použité symboly ve vývojových diagramech pro přiřazování, porovnávání atd. odpovídají syntaxi jazyka C.

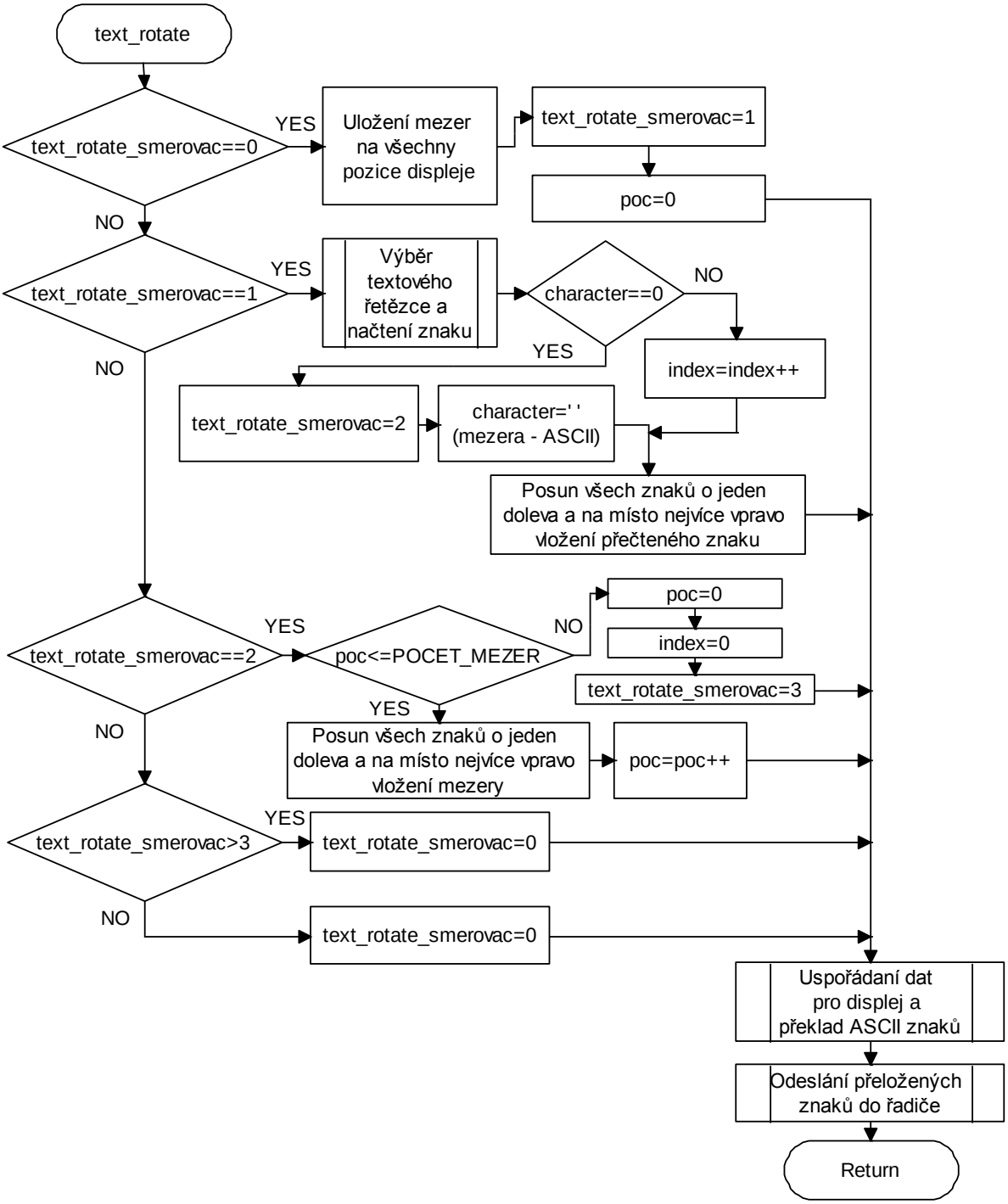
9.1.1 Hlavní funkce



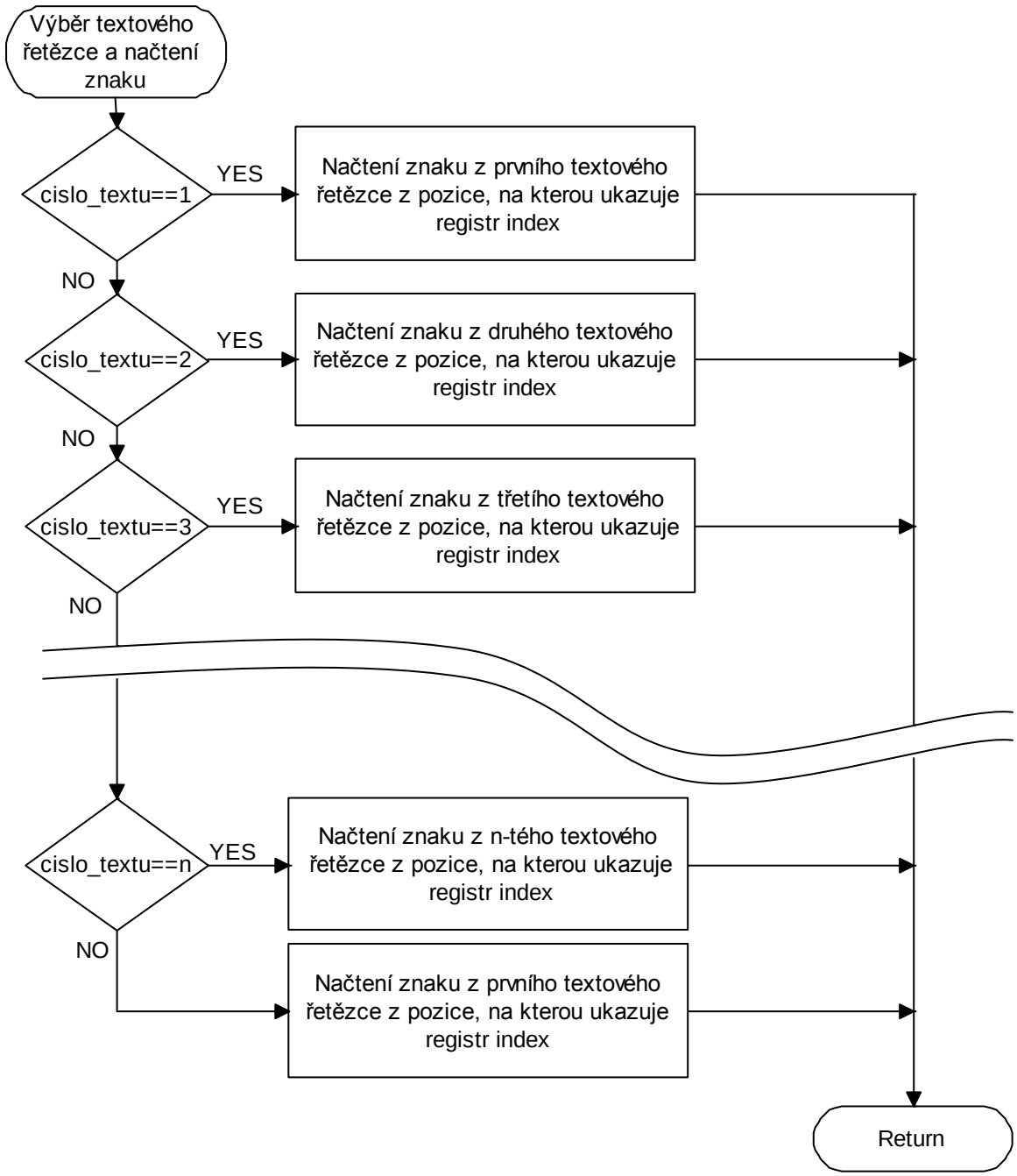
9.1.2 Zpoždění mezi obnovou dat na displeji



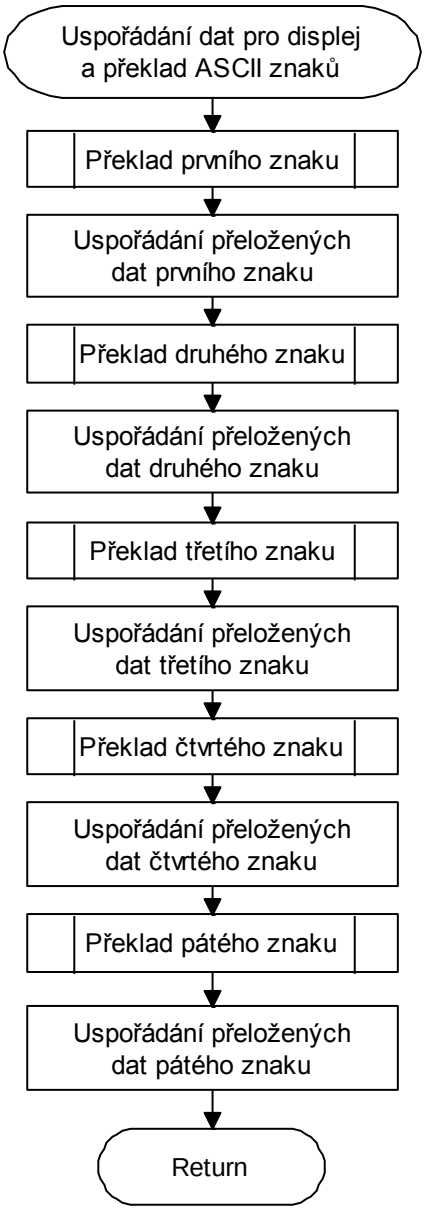
9.1.3 Vytvoření posunujícího se textu



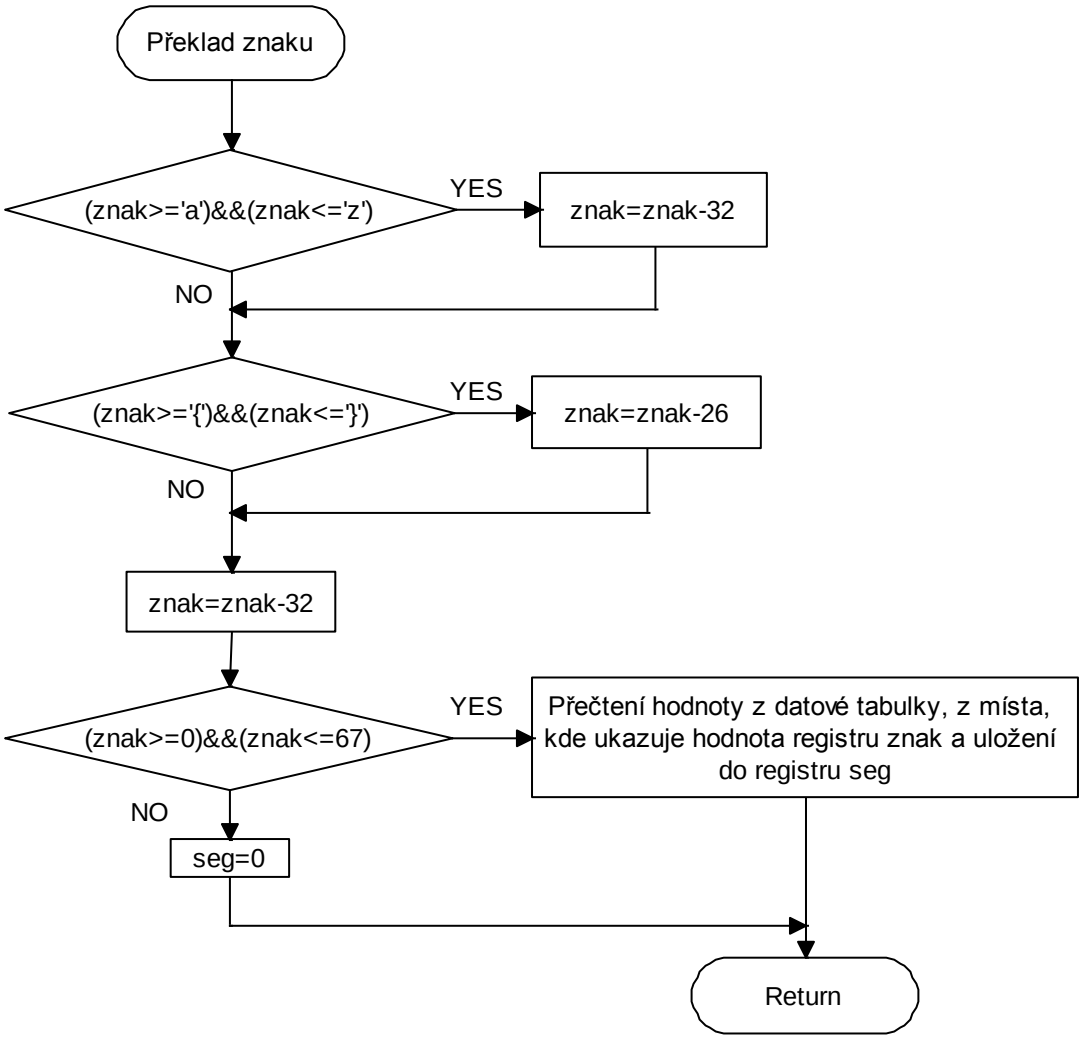
9.1.4 Výběr textového řetězce



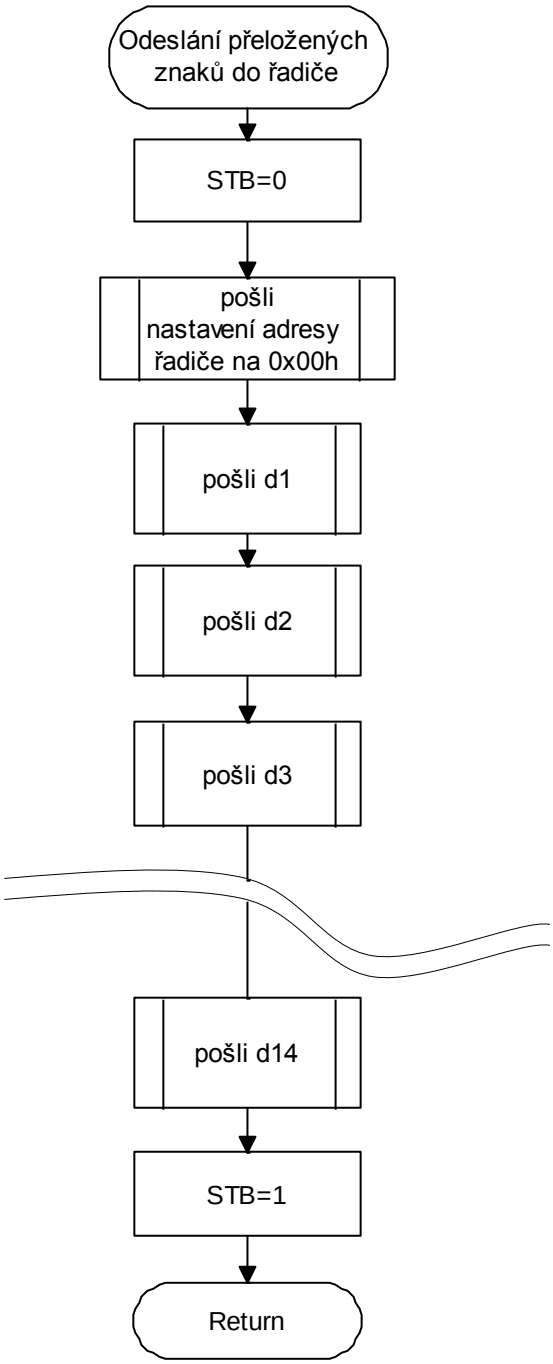
9.1.5 Uspořádání dat pro displej



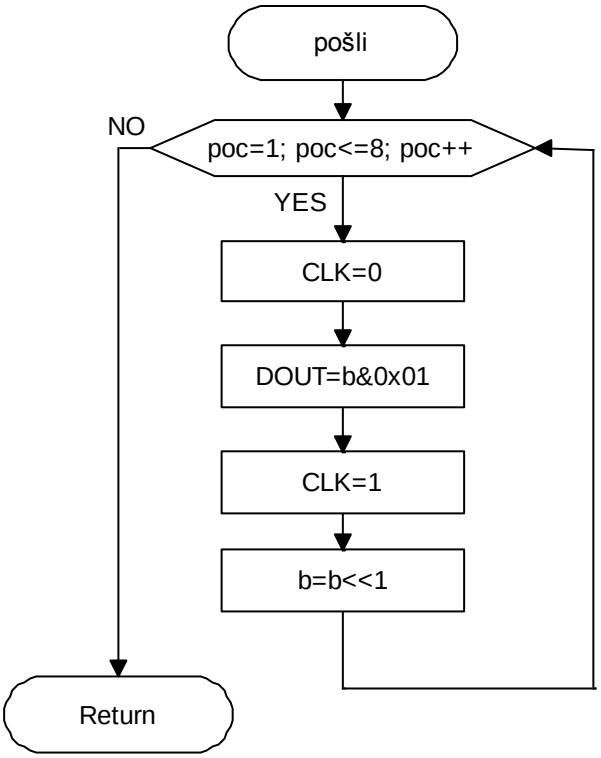
9.1.6 Algoritmus překladu znaků



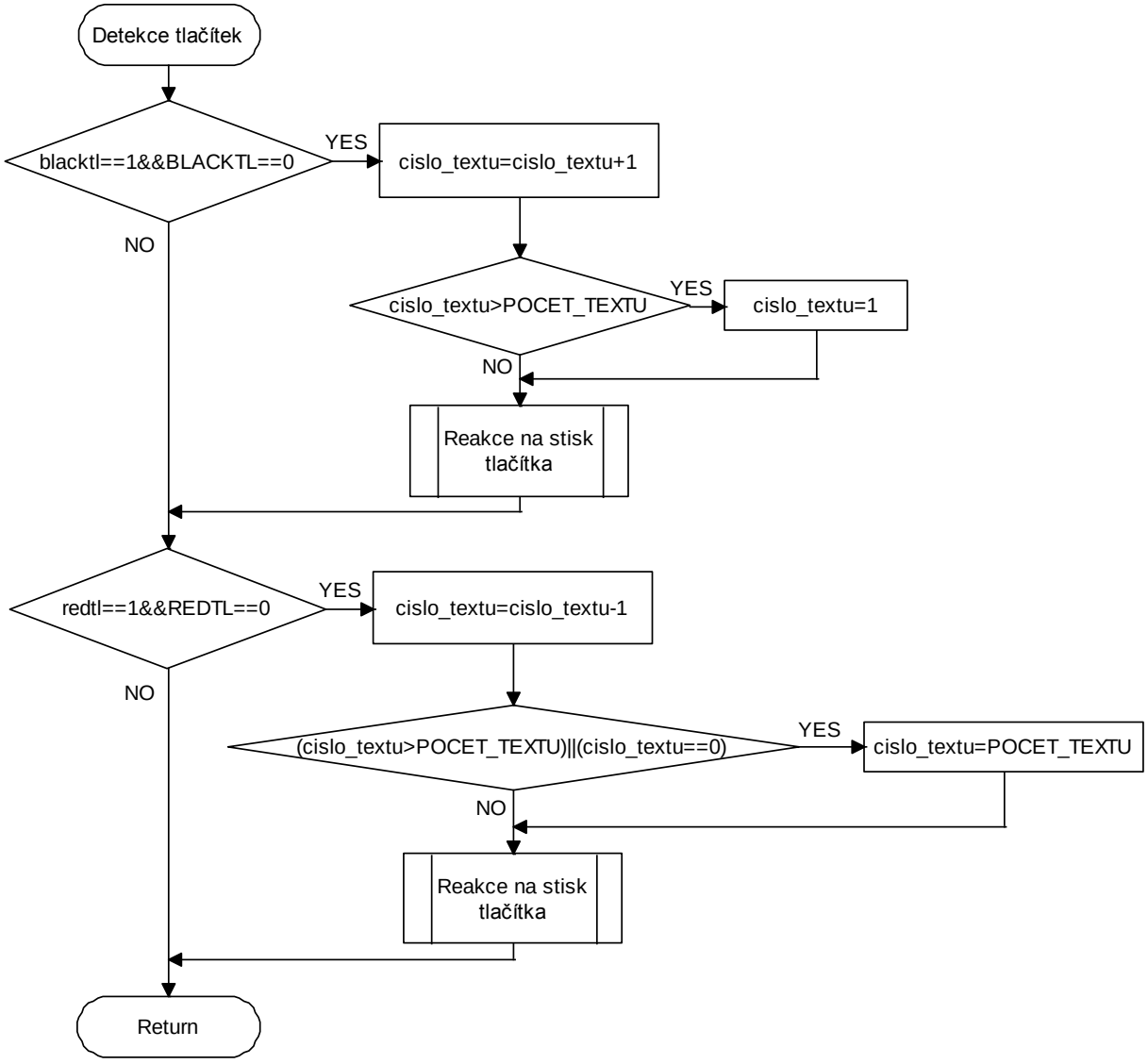
9.1.7 Odeslání dat do řadiče



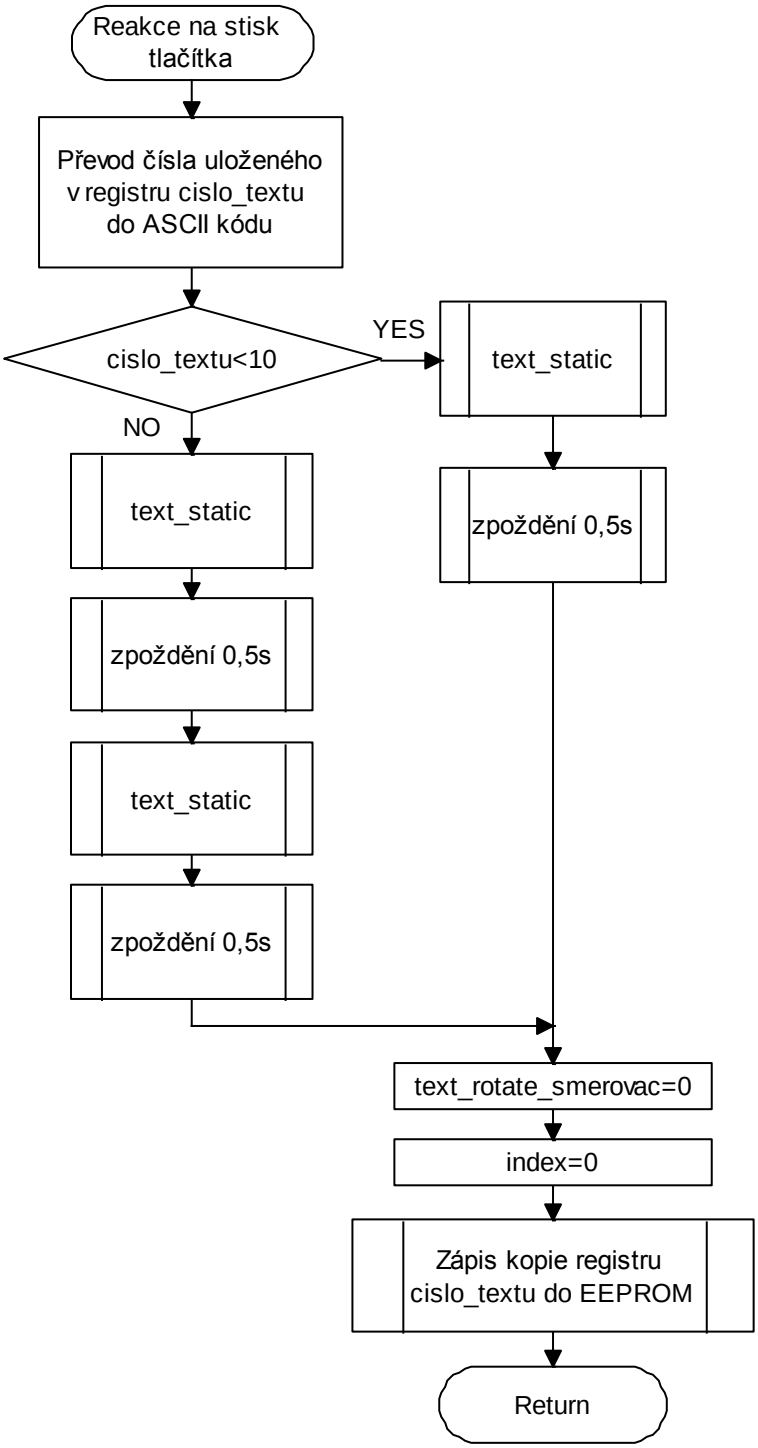
9.1.8 Softwarová SPI sběrnice – odesílání dat



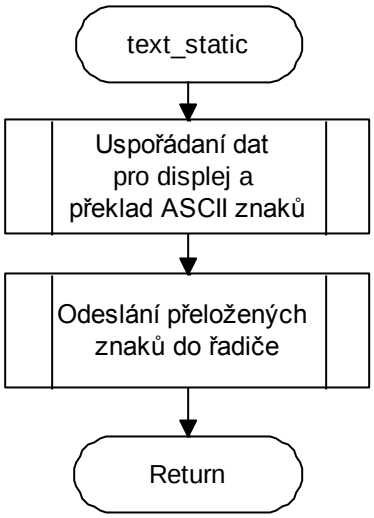
9.1.9 Detekce tlačítek



9.1.10 Reakce na stisk tlačítka



9.1.11 Funkce pro přímý přístup na displej



9.1.12 Obsluha přerušení