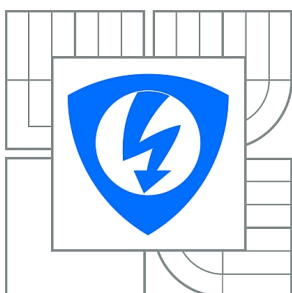




VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY



**FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH
TECHNOLOGIÍ**

ÚSTAV AUTOMATIZACE A MĚŘICÍ TECHNIKY

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION
DEPARTMENT OF CONTROL AND INSTRUMENTATION

LABORATORNÍ ÚLOHY PRO MIKROKONTROLÉR COLD FIRE MCF52233

LABORATORY ASSIGNMENTS FOR MICROCONTROLLER COLD FIRE MCF52233

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

AUTOR PRÁCE

AUTHOR

Bc. ONDŘEJ DYČKA

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. TOMÁŠ MACHO, Ph.D.

BRNO 2010



VYSOKÉ UČENÍ
TECHNICKÉ V BRNĚ

Fakulta elektrotechniky
a komunikačních technologií

Ústav automatizace a měřicí techniky

Diplomová práce

magisterský navazující studijní obor
Kybernetika, automatizace a měření

Student: Bc. Ondřej Dyčka

ID: 74925

Ročník: 2

Akademický rok: 2009/2010

NÁZEV TÉMATU:

Laboratorní úlohy pro mikrokontrolér ColdFire MCF52233

POKYNY PRO VYPRACOVÁNÍ:

1. Seznamte se s mikrokontroléry ColdFire MCF52233 firmy Freescale.
2. Navrhněte a realizujte laboratorní úlohy, které by demonstrovaly použití binárních vstupů a výstupů, přerušovacího subsystému, čítačů a časovačů, A/D převodníku, I2C a PWM. Vyžaduje-li to úloha, realizujte přizpůsobovací obvody včetně desky plošných spojů.
3. Pro každou úlohu vytvořte zadání a stručný návod.

DOPORUČENÁ LITERATURA:

[1] MCF52235 ColdFire® Integrated Microcontroller Reference Manual. Rev. 5. Denver (Colorado): FreescaleSemiconductor, Inc. 09/2007. Document Number MCF52235RM.

Termín zadání: 8.2.2010

Termín odevzdání: 24.5.2010

Vedoucí práce: Ing. Tomáš Macho, Ph.D.

prof. Ing. Pavel Jura, CSc.

Předseda oborové rady

UPOZORNĚNÍ:

Autor diplomové práce nesmí při vytváření diplomové práce porušit autorská práva třetích osob, zejména nesmí zasahovat nedovoleným způsobem do cizích autorských práv osobnostních a musí si být plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č.40/2009 Sb.

Abstrakt

Tato diplomová práce se zabývá návrhem laboratorních úloh pro mikrokontrolér MCF52233 firmy Freescale. Úlohy se soustředí na práci se základními periferiemi jako jsou čítače a časovače, obsluha AD převodníků, PWM modulace či sběrnice I2C. Dále se úlohy zabývají vyhodnocováním maticové klávesnice a komunikací s alfanumerickým displejem. K těmto úlohám je vytvořen přípravek externích periferií, který je připojen k vývojovému kitu M52233DEMO. Tento přípravek obsahuje periferie potřebné k úlohám – tlačítka, LED diody, potenciometr a I2C obvod. Jednotlivé periferie většinou sdílí své připojení k mikrokontroléru s jinou periferií. Proto přípravek obsahuje sérii přepínačů, které je potřeba nastavit pro správnou funkčnost dané periferie. Deska externích periferií také obsahuje porty pro připojení maticové klávesnice a alfanumerického displeje. Připojení displeje vyžaduje externí napájení.

Práce se zabývá realizací laboratorních úloh s využitím programového vybavení firmy Freescale. Vývojové prostředí pro programování a základní trasování je v programu CodeWarrior. Pokročilejší možnosti vyčítání informací z běžícího programu v mikrokontroléru nabízí program FreeMaster, který je využit u úlohy AD převodníků.

Každá laboratorní úloha obsahuje úvod, zadání, návod a vývojový diagram realizované úlohy. Návodů jsou sepsány tak, aby obsahovaly důležité kroky ke správnému zvládnutí zadané úlohy. Jednotlivé úlohy jsou koncipovány tak, aby postupným zvyšováním náročnosti úloh získali studenti základní programovací návyky a zvládali tak složitější úlohy.

Klíčová slova

ColdFire MCF52235, vývojový kit M52233DEMO, CodeWarrior,
FreeMaster

Abstract

This master's thesis contains design of practice exercises for microcontroller MCF52233 firm Freescale. Exercises are focused on working with basic peripherals as counters and timers, measurement with A/D converters, PWM modulation or I2C bus. Exercises are interested in evaluation of matrix keyboard and communication with alphanumeric display controller. To these exercises is made a preparation of external peripherals, which is connected with evaluation kit M52233DEMO. This preparation contains all necessary peripherals, which are used for practice exercises – buttons, LED diodes, potentiometer and I2C periphery. Individual periphery usually shares connection to microcontroller with another periphery. Therefore preparation contains switches, which are necessary to set up for correct functionality of the peripherals. The board of external peripheral also includes ports for connecting matrix keyboards and alphanumeric display. Connecting the display requires an external power supply.

Work deals with implementation of laboratory exercises using a software firm Freescale. Development environment for programming and basic debugging is in program CodeWarrior. More advanced possibility of reading information from a running program in the microcontroller offers program FreeMaster, which is used in the exercise of AD converters.

Each lab exercise contains an introduction, task, instructions and the flowchart of realized exercise. Instructions are written so that it contains important steps to properly manage assignments. Individual exercises are designed to gradually increasing complexity of exercises students obtain basic programming habits and manage the complex exercise.

Keywords

ColdFire MCF52235, evaluation kit M52233DEMO, CodeWarrior,
FreeMaster

Bibliografická citace

DYČKA, O. *Laboratorní úlohy pro mikrokontrolér ColdFire MCF52233*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2010. 67 s. Vedoucí bakalářské práce Ing. Tomáš Macho, Ph.D.

Prohlášení

„Prohlašuji, že svou diplomovou práci na téma *Laboratorní úlohy pro mikrokontrolér ColdFire MCF52233* jsem vypracoval samostatně pod vedením vedoucího diplomové práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené diplomové práce dále prohlašuji, že v souvislosti s vytvořením této diplomové práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení § 152 trestního zákona č. 140/1961 Sb.“

V Brně dne: **24. května 2010**

.....
podpis autora

Poděkování

Chtěl bych poděkovat všem lidem, bez nichž by nemohla tato práce vzniknout, především svému vedoucímu práce Ing. Tomáši Machovi, Ph.D., za vedení v průběhu realizace laboratorních úloh a za poznámky k práci. Dále bych chtěl poděkovat rodině, zejména za jejich pomoc a podporu při mém studiu.

OBSAH

OBSAH	8
SEZNAM OBRÁZKŮ A TABULEK	10
1. ÚVOD	12
2. VLASTNOSTI MIKROKONTROLÉRU COLDFIRE	14
2.1 VLASTNOSTI JÁDRA COLDFIRE V2	14
2.2 MIKROKONTROLÉR MCF52233	16
2.3 PERIFERIE MIKROKONTROLÉRU MCF52233	16
2.4 PROGRAMOVÉ NÁSTROJE	17
2.4.1 Vývojové prostředí CodeWarrior	17
2.4.2 Ladicí nástroj FreeMaster.....	19
3. NÁVRH A REALIZACE POMOCNÝCH PŘÍPRAVKŮ	22
3.1 REALIZACE	22
3.2 PŘIPOJENÍ KLÁVESNICE.....	25
3.3 PŘIPOJENÍ DISPLEJE.....	28
3.4 SCHÉMA DESKY EXTERNÍCH PERIFERÍ	32
4. LABORATORNÍ ÚLOHY	33
4.1 ÚLOHA DEMONSTRUJÍCÍ VYUŽITÍ BINÁRNÍCH VSTUPŮ A VÝSTUPŮ	33
4.1.1 Úvod.....	33
4.1.2 Zadání	33
4.1.3 Návod	33
4.1.4 Realizace	35
4.2 ÚLOHA DEMONSTRUJÍCÍ VYUŽITÍ ČASOVAČE	36
4.2.1 Úvod.....	36
4.2.2 Zadání	37
4.2.3 Návod	37
4.2.4 Realizace	38
4.3 ÚLOHA DEMONSTRUJÍCÍ VYUŽITÍ ČÍTAČE.....	39
4.3.1 Úvod.....	39
4.3.2 Zadání	39
4.3.3 Návod	39
4.3.4 Realizace	41

4.4	ÚLOHA DEMONSTRUJÍCÍ VYUŽITÍ A/D PŘEVODNÍKU	42
4.4.1	Úvod	42
4.4.2	Zadání	43
4.4.3	Návod	43
4.4.4	Realizace	45
4.5	ÚLOHA DEMONSTRUJÍCÍ VYUŽITÍ PWM MODULACE	45
4.5.1	Úvod	45
4.5.1	Zadání	46
4.5.2	Návod	47
4.5.3	Realizace	48
4.6	ÚLOHA PRO KLÁVESNICI	48
4.6.1	Zadání	48
4.6.2	Návod	49
4.6.3	Realizace	50
4.7	ÚLOHA PRO OVLÁDÁNÍ DISPLEJE.....	51
4.7.1	Zadání	51
4.7.2	Návod	51
4.7.3	Realizace	53
4.8	ÚLOHA DEMONSTRUJÍCÍ VYUŽITÍ I2C SBĚRNICE	54
4.8.1	Úvod	54
4.8.2	Zadání	55
4.8.3	Návod	55
4.8.1	Realizace	58
5.	ZÁVĚR	61
6.	SEZNAM LITERATURY.....	63
7.	SEZNAM PŘÍLOH	64

SEZNAM OBRÁZKŮ A TABULEK

Obrázky:

Obr. 1 blokový diagram mikroprocesoru MCF523X.....	15
Obr. 2 blokový diagram mikrokontroléru MCF5233X.....	15
Obr. 3 Vývojové prostředí CodeWarrior	18
Obr. 4 Zobrazení průběhu vyhodnocování trojosého akcelerometru MMA7260Q ...	19
Obr. 5 Nastavení komunikačního protokolu programu FreeMaster	20
Obr. 6 Dialogové okno pro vytvoření sledované proměnné	21
Obr. 7 Schéma maticové klávesnice	25
Obr. 8 Maticová klávesnice F-KV16KEY	26
Obr. 9 Zapojení maticové klávesnice F-KV16KEY	27
Obr. 10 Připojení maticové klávesnice	27
Obr. 11 Displej DEM16217SYH	28
Obr. 12 Blokové schéma displeje DEM16217SYH	29
Obr. 13 Připojení napětí pro řadič HD4470	30
Obr. 14 Ukázka datové komunikace s řadičem HD4470.....	31
Obr. 15 Připojení LCD displeje	31
Obr. 16 Vývojový diagram programu pro ovládání hada tlačítka	35
Obr. 17 Vývojový diagram programu světelného hada s čekací smyčkou	36
Obr. 18 Vývojový diagram úlohy časovače	38
Obr. 19 Vývojový diagram programu stopek	41
Obr. 20 Blokové schéma A/D převodníku mikrokontroléru MCF52233.....	42
Obr. 21 Vývojový diagram úlohy měření AD převodníků	45
Obr. 22 Průběh pulsně šířkové modulace	46
Obr. 23 Vývojový diagram úlohy PWM modulace	48
Obr. 24 Vývojový diagram úlohy vyčítání znaků z klávesnice	50
Obr. 25 Vývojový diagram úlohy pro ovládání displeje	53
Obr. 26 Přenos dat na sběrnici I2C	54
Obr. 27 Vývojový diagram hlavní funkce úlohy I2C sběrnice	58
Obr. 28 Vývojový diagram funkce zápisu dat na I2C sběrnici	59
Obr. 29 Vývojový diagram funkce čtení dat z I2C sběrnice	60

Tabulky:

Tab. 1 Seznam periférií na kitu M52233DEMO	22
Tab. 2 Seznam vývodů použitých na desce externích periférií	23
Tab. 3 Metoda postupující nuly	26
Tab. 4 Seznam vývodů řadiče displeje DEM16217SYH	29

1. ÚVOD

V dnešním světě se stále více setkáváme se zařízeními obsahujícími mikrokontrolér. Je to způsobeno nízkou cenou a širokým sortimentem procesorů nabízených výrobci. Zařízení tak mohou obsahovat levnou variantu mikrokontroléru pouze s potřebnými periferiemi pro dané zařízení. Proto je potřeba studenty elektrotechnických oborů vyučovat tak, aby byli schopni takoveto zařízení programovat, či kompletně navrhovat. A tak se s programováním mikrokontrolérů setkáme na vysokých i odborných středních školách. I na školách se vyvíjí rozsah znalostí a použité mikrokontroléry. První oblíbené školní mikrokontroléry byly založeny na procesoru Intel 8051. Dnešní výuka bývá založena na 16bitových procesorech. Možná se trh práce posune a na školách se bude demonstrovat použití procesorů 32bitových. Ovšem dnešní cena – okolo 120\$ – je zatím vysoká, není nezbytně nutné učit studenty programovat 32bitové procesory se širokou škálou periferií přímo na čipu. Základní programovací návyky vychází z 8bitových, potažmo 16bitových procesorů.

I tato práce obsahuje základní úlohy s běžnými periferiemi, které obsahují i 16bitové procesory. Jako pokračovací kurz pro tyto laboratorní úlohy může být využito dalších periferií, které 32bitové procesory obsahují. Dále pak programování aplikací, které vyžadují větší početní výkon těchto mikrokontrolérů (šifrovací algoritmy, přenos dat po Ethernetu, složité řízení zařízení, atd.).

Pro správné vytvoření laboratorních úloh je důležité, aby byla předkládaná látka dobře pochopitelná pro studenty. Toto je velice složité a z velké míry to závisí na zkušenostech vyučujících a možnostech výuky (počet hodin týdně, provázanost s ostatními předměty.) Laboratorní úlohy musí být koncipované tak, aby byly zvládnutelné v daném čase. Také je potřeba nastavit obtížnost jednotlivých úloh, aby to pomalejší studenty neodradilo a bystřejší studenti se při programování nenudili. Další součástí úspěšného předání znalostí studentům prostřednictvím laboratorních úloh je jistá zábavnost úloh. Z vlastní zkušenosti vím, že pokud student testuje vnitřní paměť pomocí testovací sekvence, není to tak záživné, jako když si student může hrát s displejem, tlačítky a diodami LED.

Je vidět, že vytvořit plán výuky tak, aby studenty co nejvíce zaujal a předal návyky programování, není jednoduché. O to víc by si studenti měli vážit práci vyučujících.

K úspěšnému programování úloh jsou nutné znalosti samotného procesoru. Spolu s laboratorními úlohami by studenti měli získávat obecné informace ohledně mikrokontrolérů na přednáškách. V laboratorních cvičeních by pak měli získat konkrétní informace k mikrokontroléru MCF52233, který bude součástí přípravku pro programování. Tyto informace by měli být vodítkem pro správné programování (adresy registrů, vzorce pro výpočty rychlostí čítačů/časovačů, atd.).

Cílem této práce je vytvořit koncept, který může být použit jako podklad pro vytvoření laboratorních úloh s 32bitovým mikrokontrolérem firmy Freescale řady ColdFire MCF52233. Tyto laboratorní úlohy se snaží pojmout základy při programování mikrokontrolérů. Studenti se v nich seznámí s ovládáním vstupů a výstupů a základních periferií jako jsou převodníky, komunikační sběrnice, či displeje nebo klávesnice. Jako základ je použit studentský vývojový kit M52233DEMO firmy Freescale, který obsahuje mikrokontrolér MCF52233 včetně programátoru, základních periferií a portu pro připojení dalších periferií.

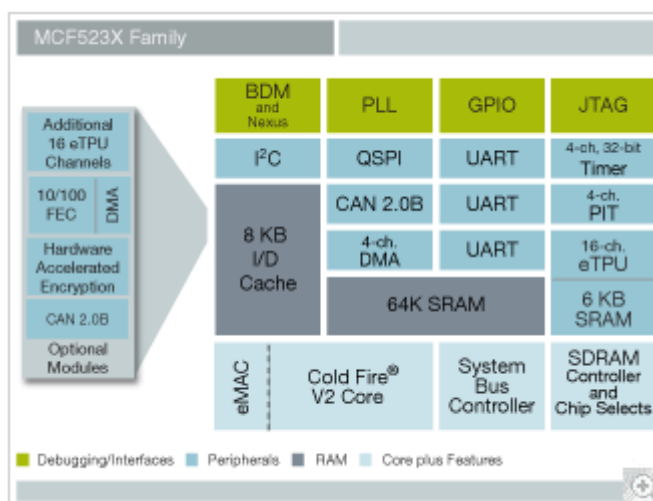
Dále se tato práce zabývá vytvořením desky externích periferií, která slouží jako rozšíření ke kitu M52233DEMO, aby poskytovala všechny potřebné periferie pro laboratorní úlohy. Deska obsahuje indikační LED diody, tlačítka, připojení pro klávesnici a displej. Dále obsahuje dvoubarevnou LED diodu pro testování pulsně šířkové modulace (PWM), potenciometr pro snímání dat A/D převodníkem a obvod pro komunikace po sběrnici I2C. Tento obvod je jednoduchý obvod reálného času (RTC) a kalendáře. Vytváří tak spolu s mikrokontrolérem MCF52233 na kitu M52233DEMO laboratorní přípravek, na kterém mohou studenti realizovat laboratorní úlohy, popřípadě testovat další možnosti mikrokontroléru MCF52233.

2. VLASTNOSTI MIKROKONTROLÉRU COLDFIRE

2.1 VLASTNOSTI JÁDRA COLDFIRE V2

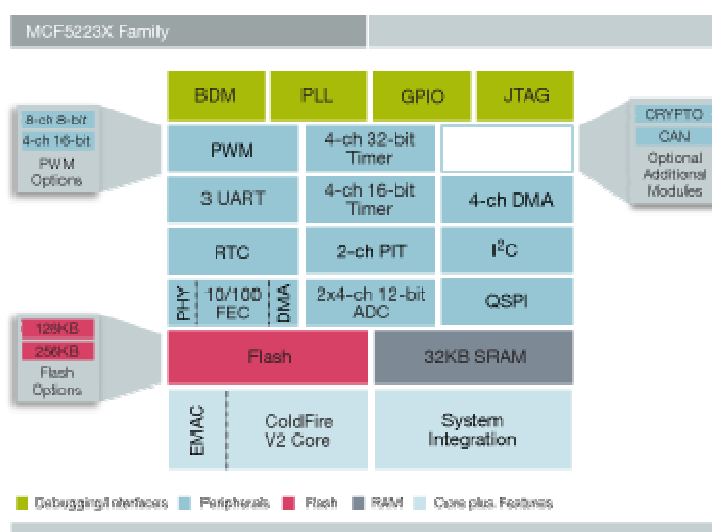
Rodina procesorů ColdFire navazuje na známou sérii procesorů M68000, která byla základem několika osobních počítačů (například Apple Lisa, Macintosh a Atari ST) i vědeckých kalkulaček firmy Texas Instrument. Tyto procesory byly oblíbené v 80. letech minulého století a jejich vývoj byl ukončen v roce 1996. Tehdejší procesory pracovaly s frekvencí okolo 10 MHz. Instrukční sada ISA_A vychází z instrukční sady M68000. Instrukční sada procesorů ColdFire ISA_A+ byla optimalizována pro současné programovací jazyky a práci s 32bitovými registry. Například obsluha přerušení u procesorů ColdFire vychází ze série M68000, a proto se tato obsluha podobá nastavením přerušení na osobních počítačích. Pro tento úkon nestačí standardní deklarace jako u standardních 16bitových mikrokontrolérů funkcí. Je nutné vytvořit funkci pro nastavení vektoru přerušení, výměny starého vektoru přerušení a podobně.

Rodina ColdFire má jádra procesoru ve čtyřech verzích V1 až V4. ColdFire V1 a V2 jsou použity pro mikrokontroléry, mezi nimi i MCF52233. Dále firma Freescale vyrábí verze mikroprocesorů (MPU) ColdFire V2 až V4, které jsou pro embedded aplikace. Na obrázku 1 je blokový diagram mikroprocesoru řady MCF523X a na obrázku 2 blokový diagram mikrokontroléru MCF5233X. Obě jádra jsou typu ColdFire V2 a je na nich patrný rozdíl mezi mikrokontrolérem a mikroprocesorem. Tyto dva pojmy často splývají, avšak mikrokontrolér je obvod sám o sobě funkční, obsahuje tedy všechny potřebné periferie pro běh programu. Kdežto mikroprocesor slouží pouze pro výpočty a neobsahuje například paměť pro uložení dat a programu, pouze obsahuje systémovou sběrnici pro připojení těchto pamětí.



Obr. 1 blokový diagram mikroprocesoru MCF523X

http://www.freescale.com/webapp/sps/site/prod_summary.jsp?code=MCF523X



Obr. 2 blokový diagram mikrokontroléru MCF5233X

http://www.freescale.com/webapp/sps/site/prod_summary.jsp?code=MCF5233X

Jádro ColdFire V2 je navrženo pro masové nasazení v elektroprůmyslu. Je tedy vyrobeno za minimální výrobní cenu a přitom dosahuje kvalitní výpočetní výkon. Jádro podporuje takt procesoru až 166 MHz, který je vyroben 0,13 μm technologií. Tento 32bitový RISC procesor podporuje délku instrukce 16, 32 nebo 48 bitů. V jádru je také obsažena podpora pro trasování a ladění programu – Background Debug Mode (BDM).

2.2 MIKROKONTROLÉR MCF52233

Mikrokontroléry řady MCF52233 jsou z rodiny ColdFire® s redukovanou instrukční sadou (RISC). Mikrokontrolér disponuje jádrem V2 ColdFire pracujícím až s frekvencí 60 MHz (2.1 MIPS na 1 MHz). Na čipu je 32 kB SRAM paměti a 256 kB Flash paměti pro data a program. Procesor pracuje s 32 bitovými adresovými a datovými registry. Rozšířená jednotka pro násobení čísel v akumulátoru (EMAC) dovoluje násobit 16bitová čísla i čísla o velikosti 32 bitů. Součástí jádra je hardwarová dělička. Mikrokontrolér obsahuje širokou škálu integrovaných rozhraní. Spolu s rozhraními je potřeba výkonně a efektivně zabezpečit jejich přesnost, popřípadě bezpečnost. Proto procesor obsahuje jednotku pro urychlení šifrování (CAU) – koprocesor, který urychluje softwarové šifrování pro přenosy realizované tímto mikrokontrolérem. Jednotka CAU také obsahuje generátor náhodných čísel.

Pro kontrolu a ladění napsaných programů má mikrokontrolér vylepšený režim pro ladění na pozadí (BDM DEBUG_B+), který obsahuje i trasování v reálném čase pro určování průběhu dynamických výpočtů. Pro testování mikrokontroléru zapájeného na desce podporuje kontrolér testovací rozhraní JTAG.

2.3 PERIFERIE MIKROKONTROLÉRU MCF52233

Mikrokontrolér je zařízení, které poskytuje nejenom výpočetní výkon, ale musí obsahovat i široké spektrum rozhraní pro připojení periférií. I mikrokontroléry MCF52233 nejsou výjimkou a obsahují několik možností připojení. Spolu s hardwarovou podporou těchto rozhraní výrobce implementoval na čip řídicí jednotky a rozšířené nastavovací registry pro jednotlivá rozhraní, čímž ulehčuje práci programátorům.

Na čipu mikrokontroléru je implementován ethernetový modul s fyzickou vrstvou ethernetu (EPHY). Pro obsluhu ethernetu mikrokontrolér obsahuje Fast Ethernet Controller (FEC) s 10/100 Mb sítí half/full duplex. To umožňuje zařízení připojit na jednu z nejrozšířenějších dálkových sítí nejen mezi počítači.

Mikrokontrolér podporuje sběrnici FlexCAN 2.0B. Dále jsou na čipu vytvořeny 3 řadiče synchronní/asynchronní sériové linky (UART). Řadič sériové linky obsahuje podporu modemových signálů request-to-send (RTS) a clear-to-send (CTS). Další interface sériového přenosu je založen na sběrnici SPI (synchronní sériová linka) a tím je QSPI. Toto rozhraní dovoluje jednotlivé pakety spřáhnout za sebe a vyslat na linku SPI.

Dalším rozšířeným rozhraním je linka I2C. Tato linka se používá pro komunikaci s jednoduchými zařízeními jako inteligentními senzory, displeji či klávesnicemi. Linka podporuje v režimu master-slave více řídících zařízení typu master.

U mikrokontroléru MCF52233 nalezneme také základní periferie běžné i u jednodušších 16bitových procesorů. Obsahuje osmi kanálový 12bitový rychlý A/D převodník. Pro pulsní řízení je k dispozici 8kanálová (8bitová) a 4kanálová (16bitová) pulsní šířková modulace (PWM). Pro generování přesných časových úseků má mikrokontrolér čtyřkanálový časovač GPT, který podporuje i funkci output compare (využitelné např. u měření délky trvání signálu na vstupu).

2.4 PROGRAMOVÉ NÁSTROJE

2.4.1 Vývojové prostředí CodeWarrior

Vývojové prostředí pro práci s mikrokontroléry Freescale se nazývá CodeWarrior. Základní rozdělení prostředí je ve dvou distribucích pro 8bitové a 16bitové procesory a pro rodinu ColdFire. Existuje několik licencí, mezi nimi i volná speciální edice, dostupná na adrese

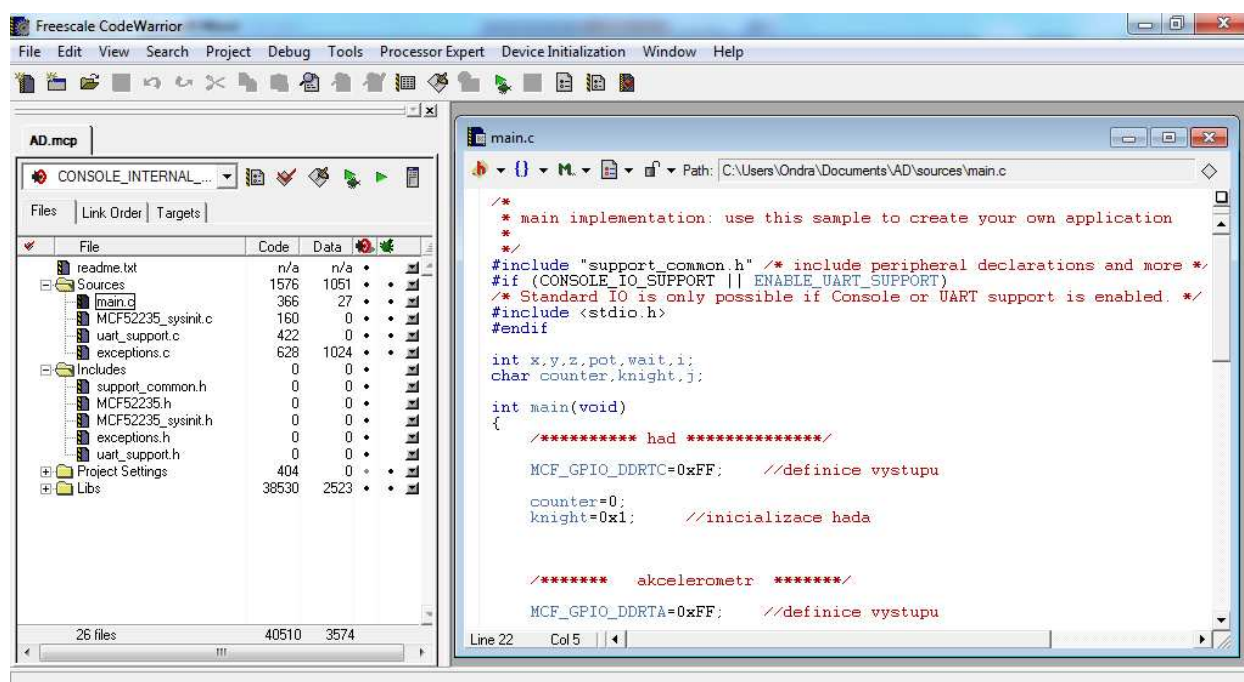
<http://www.freescale.com/webapp/sps/site/overview.jsp?nodeId=01272600610BF1>.

Omezení je pouze ve velikosti kódu (128 kB) a bez podpory jazyka C++. Tato verze pro studentské úlohy stačí.

Tento programovací nástroj využívá plné možnosti mikrokontrolérů firmy Freescale. Mezi hlavní výhody patří nástroje pro trasování procesoru v reálném čase a velké množství podpůrných knihoven. Tyto knihovny dovolují jednoduchý přístup k vnitřním registrům mikrokontrolérů a nastavování jednotlivých bitů v registru. Princip je založen na maskování jednotlivých bitů, přičemž konečný zápis je pak pro programátory dobře čitelný. Například pro nastavení bitu ALM v registru RTCISR je pomocí příkazu `MCF_RTC_RTCISR |= MCF_RTC_RTCISR_ALM`. Pomocí integrovaných knihoven je také možné nastavit hodinový kmitočet mikrokontroléru. V knihovně `m52233_sysinit.h` je nastavený kmitočet na 60 MHz příkazem `#define SYSTEM_CLOCK_KHZ 60000`.

Další výhodou vývojového prostředí je nadefinování standardního výstupu funkce `printf()`. Tento výstup je nastaven do konzolového okna při trasování. Takže je možné během programu vypisovat jednotlivé údaje do konzoly pomocí klasického formátování ve funkci `printf()`. Tento výpis je výhodné použít v jednoduchých případech, ovšem je časově náročný a brzdí tak běh samotného programu v mikrokontroléru.

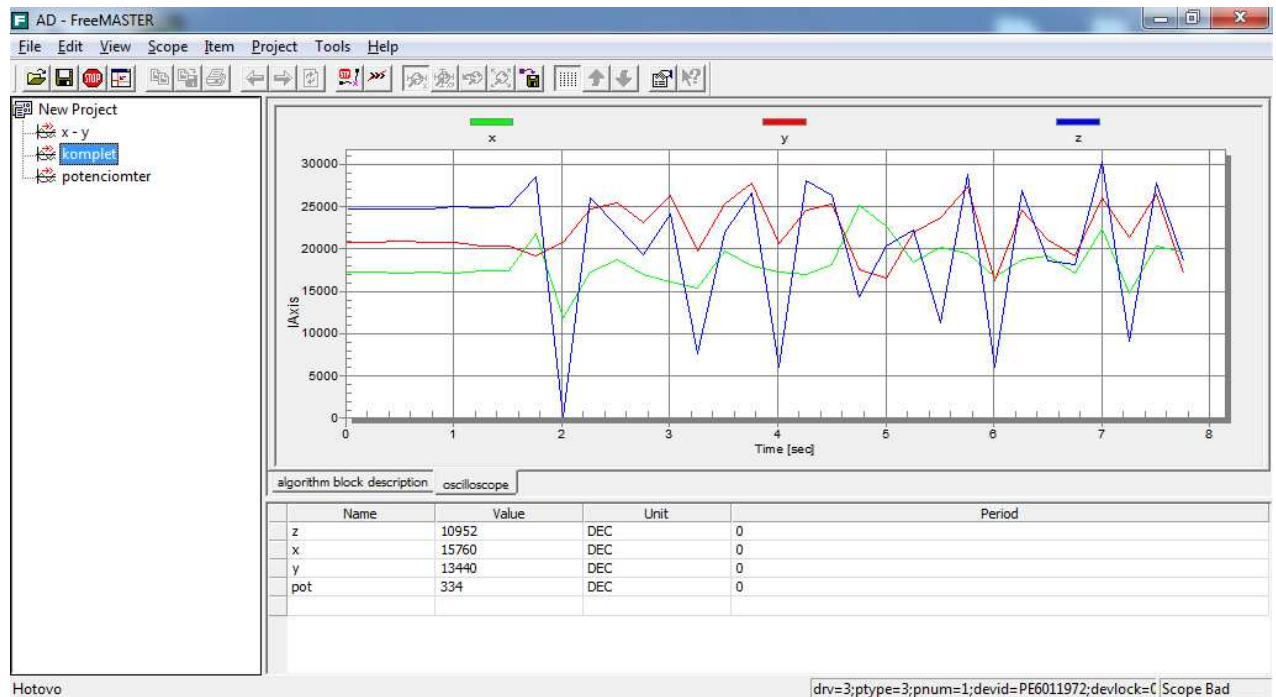
Pomocí jednoduchého průvodce lze založit nový projekt pro mikrokontrolér MCF52233 a programovací jazyk C. Na obrázku 3 je zobrazeno základní pracovní prostředí pro práci s projektem. V pravé části je prostor pro zobrazení zdrojových kódů a v levé části soupis souborů spojených s projektem. Aby bylo možné používat vlastní vytvořené knihovny, je potřeba zdrojové soubory knihovny přidat do projektu.



Obr. 3 Vývojové prostředí CodeWarrior

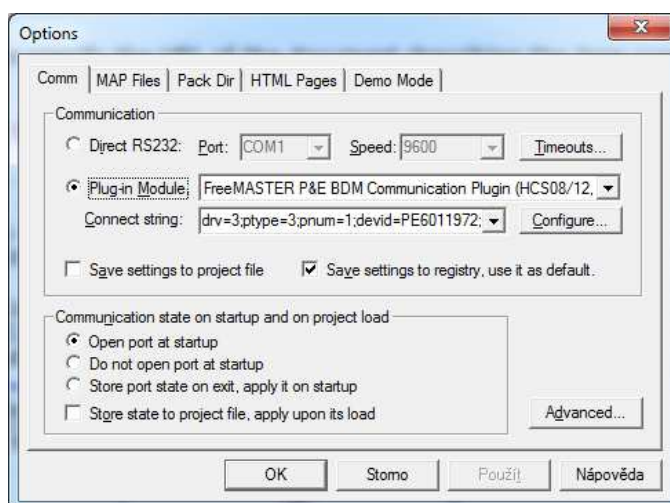
2.4.2 Ladicí nástroj FreeMASTER

Firma Freescale má dobře zpracovanou podporu pro ladění aplikací na svých mikrokontrolérech. A to jak po stránce hardware, tak i software. V základním prostředí CodeWarrior nalezneme prostředky pro trasování programů. Ovšem existuje rozšiřující program FreeMASTER, který tyto možnosti značně rozšiřuje. Je možné vyčítat aktuální data z proměnných a registrů v běžícím programu mikrokontroléru. Jako rozšíření můžeme tyto data vynášet do grafů. Tím získáváme představu o vývoji hodnot proměnných i v závislosti na čase. To je výhodné například pro ladění aplikací využívající převodníky A/D. Jednoduchým testováním můžeme vyčíst měřený vstupní signál, jeho mezní hodnoty, a tím správně využít měřicí rozsah převodníků. Na obrázku 4 je ukázka programu FreeMASTER pro zobrazování měřených dat z trojosého akcelerometru MMA7260Q.



Obr. 4 Zobrazení průběhu vyhodnocování trojosého akcelerometru MMA7260Q

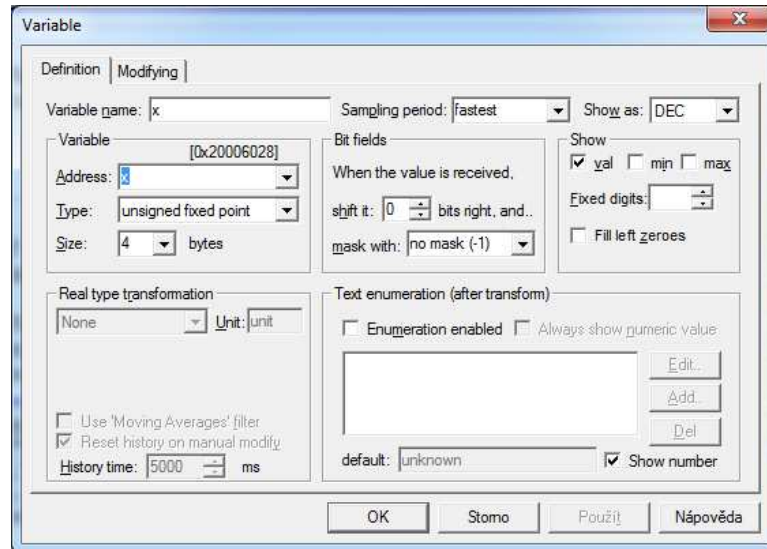
Program je volně k dispozici po bezplatné registraci na adrese http://www.freescale.com/webapp/sps/site/prod_summary.jsp?code=FREEMASTER v sekci Download (položka FMASTERSW). Nastavení komunikace je zobrazeno na obrázku 5, kde je potřeba nastavit P&E BDM komunikační protokol. Samotné propojení s projektem z prostředí CodeWarrior je v sekci nastavení projektu *Project - Options – MAP Files*, kde je potřeba načíst data projektu ze souboru `\bin\MCF52235_CONSOLE_INTERNAL_RAM.elf`.



Obr. 5 Nastavení komunikačního protokolu programu FreeMaster

Tímto propojením lze ovládat běh programu, který je nahrán v procesoru pomocí prostředí CodeWarrior. Připojit k mikrokontroléru lze pouze jeden program, proto pro použití FreeMasteru je nutné vypnout debugovací režim v CodeWarrioru.

Nyní lze přidat sledování proměnné z projektu, které můžeme v reálném čase vyčítat z mikrokontroléru. V sekci *Options – Variables – New* se objeví okno pro přidávání nových proměnných, které chceme sledovat. Nastavovací dialogové okno je zobrazeno na obrázku 6. Požadovanou proměnnou lze napsat do položky *Adress* přímo tak, jak je pojmenovaná v projektu CodeWarrior. Jako volby jednotlivých proměnných je i čas zjištění proměnné, který je vhodný zvolit na *Sampling period: fastest* – nejrychlejší možné vyčítání. Dále lze mimo jiné definovat formát proměnné, či sledování minima maxima.



Obr. 6 Dialogové okno pro vytvoření sledované proměnné

Jednotlivé sledované proměnné jsou pak zobrazeny v hlavním okně programu. Lze zobrazovat i časové průběhy jednotlivých proměnných. V položce *Item – Create Scope* lze vytvořit graf, kde v položce *Setup* můžeme nastavovat proměnné zobrazované v grafu.

3. NÁVRH A REALIZACE POMOCNÝCH PŘÍPRAVKŮ

Pro laboratorní úlohy je potřeba k mikrokontroléru MCF52233 připojit potřebné periferie – led diody, tlačítka, brány pro připojení klávesnice a displeje, obvod kalendáře na I2C a potenciometr pro snímání A/D převodníku.

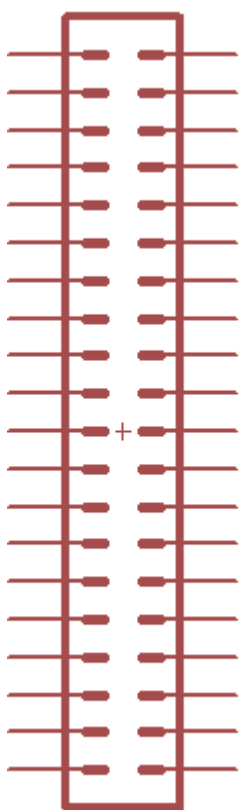
3.1 REALIZACE

Vývojový kit M52233DEMO obsahuje konektor se 40 vývody, kde jsou vyvedeny různé brány mikrokontroléru MCF52233. Na tomto vývojovém kitu se nacházejí jednoduché periferie pro základní práci. V tabulce 1 jsou uvedeny jejich vývody, které jsou připojeny na brány mikrokontroléru.

Periferie	Brána
LED0	PTC[0]
LED1	PTC[1]
LED2	PTC[2]
LED3	PTC[3]
POTENCIOMETR	PAN[0]
UTXD0	PUA[0]
URXD0	PUA[1]
URTS0	PUA[2]
UCTS0	PUA[3]
ACC_X	PAN[4]
ACC_Y	PAN[5]
ACC_Z	PAN[6]
ACC_G_SEL1	PWM1
ACC_G_SEL1	PWM2
ACC_SLEEP	PWM5

Tab. 1 Seznam periferií na kitu M52233DEMO

Všechny periferie na kitu obsahují přepínače, které umožňují jejich odpojení od bran mikrokontroléru. Protože externí periferie využívají stejné brány mikrokontroléru, obsahuje i deska přepínače pro odpojení. Je proto možné použít externí periferie na desce, či periferie připojené přímo na kitu M52233DEMO. Rozpis použitých bran mikrokontroléru je zobrazen v tabulce 2:

Popis	Vývod	Č.		Č.	Popis	Vývod
Napájení 3,3 V	Vcc	1		2	X	
Zem	GND	3	4	X		
Klávesnice COL 0	PUB [0]	5	6	X		
Klávesnice COL 1	PUB [1]	7	8	X		
Klávesnice COL 2	PUB [2]	9	10	PAN [0]	LCD_DB[0]/Pot	
Klávesnice COL 3	PUB [3]	11	12	PAN [1]	LCD_DB[1]	
PWM Red	PWM1	13	14	PAN [2]	LCD_DB[2]	
PWM Green	PWM3	15	16	PAN [3]	LCD_DB[3]	
Klávesnice ROW0	PQS [0]	17	18	PAN [4]	LCD_DB[4]	
Klávesnice ROW1	PQS [1]	19	20	PAN [5]	LCD_DB[5]	
Klávesnice ROW2	PQS [2]	21	22	PAN [6]	LCD_DB[6]	
Klávesnice ROW3	PQS [3]	23	24	PAN [7]	LCD_DB[7]	
TL0	PUA [0]	25	26	SCL	I2C_SCL	
TL1	PUA [1]	27	28	SDA	I2C_SDA	
TL2	PUA [2]	29	30	X		
TL3	PUA [3]	31	32	PTA[3]	LCD_RS	
	X	33	34	PTC [0]	LED1	
LCD_RW	PAS [3]	35	36	PTC [1]	LED2	
LCD_E	PAS [2]	37	38	PTC [2]	LED3	
	X	39	40	PTC [3]	LED4	

X – nezapojený vývod

Tab. 2 Seznam vývodů použitých na desce externích periférií

Pro jednoduchou indikaci byly použity LED diody červené barvy. Tyto diody jsou připojeny na vývody brány PTC [0:3], stejně jako diody na kitu M52233DEMO. To proto, aby indikační diody byly pohromadě s ostatními ovládacími prvky na desce s externími perifériemi. Pro přehlednost laboratorních úloh je doporučeno vypnout LED diody na kitu M52233DEMO (přepínač LED_EN) a používat k indikaci diody

na desce externích periférií. LED diody obsahují ochranný odpor, který definuje maximální proud při svitu diody. Tento proud je zvolený 12 mA, z napájecího napětí výstupů 3,3 V je odpor definován podle vztahu 1. Led diody jsou zapojeny tak, aby při vstupu logické jedničky na bráně mikrokontroléru svítily.

$$R = \frac{U}{I} = \frac{3,3V}{12mA} = 270\Omega \quad (1)$$

Tlačítka jsou připojena na bránu PUA [0:3], na které je na kitu připojena komunikační sběrnice RS232. Proto je nutné pro správnou funkci tlačítek tuto sběrnici odpojit (přepínač COM_EN). Pro případ použití sběrnice RS232 na kitu je potřeba odpojit tlačítka na desce externích periférií přepínačem TL_EN. Tlačítka jsou zapojena tak, že při vypnutém stavu je na bráně PUA logická nula a při sepnutí je přivedena logická jednička.

Maticová klávesnice je připojena na brány nazvané PUB a PQS. Brána PUB [0:3] je použita jako vstup do mikrokontroléru COL (sloupce klávesnice), brána PQS [0:3] je použita jako výstup z mikrokontroléru ROW (řádky klávesnice).

Pro připojení řadiče displeje je datová sběrnice řadiče připojena na bránu PAN [0:7]. Dále řídicí signál RW je připojen na PAS [3], potvrzovací signál E na PAS [2], a signál výběru registru RS je připojen na PTA [3].

Společně s datovou sběrnicí pro LCD displej je také na vývodu PAN [0] vstup pro A/D převodník. Proto je na desce přepínač POT_LCD_SW který přepíná využití vývodu PAN [0] pro datovou komunikaci s LCD či připojuje potenciometr na vstup AD převodníku. Na kitu M52233DEMO sice je potenciometr připojen, ale je malý a pro laboratorní účely (odolnost vůči hraní studentů, časté otáčení) je nevyužitelný. Proto doporučuji interní potenciometr vypnout přepínačem POT_EN na kitu M52233DEMO.

Zařízení připojené ke komunikační sběrnicí I2C je obvod reálného času a kalendáře PCF8583. Napájení obvodu je 3,3 V, které je přivedeno z kitu M52233DEMO. Sběrnice I2C je náchylná na změnu kapacity na datových vodičích, proto jsou na sběrnicí připojeny odpory 4,7 kΩ. Obvod je připojen na vývod mikrokontroléru I2C_SDL, který slouží jako datový vodič sběrnice I2C. Pro časový signál je využit vývod I2C_SCL.

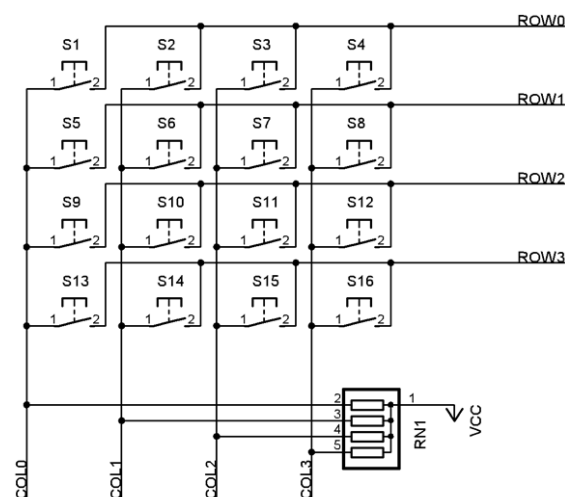
3.2 PŘIPOJENÍ KLÁVESNICE

Klávesnice slouží ke vkládání příkazů pro zařízení. Je to soubor tlačítek uspořádaných tak, aby ovládání daného zařízení bylo efektivní. Proto existuje nepřeberné množství uspořádání a provedení klávesnic. Prakticky existují dva základní principy klávesnic pro mikrokontroléry:

- klávesnice s rozhraním
- maticové klávesnice

Klávesnice s rozhraním obsahují jednoduchý mikrokontrolér, který vyhodnocuje stavy tlačítek na klávesnici a komunikuje po sběrnici. Bývají to nejčastější komunikační sběrnice I2C, či sériová linka, popřípadě SPI. Klávesnice pak může mít libovolné uspořádání tlačítek s libovolným principem snímání – tlačítková nebo dotyková klávesnice – vše řeší vnitřní mikrokontrolér v klávesnici. Programátor pak ke klávesnici přistupuje přes komunikační sběrnici jako k perifernímu zařízení, ze kterého vyčítá stavy tlačítek. Výhoda tohoto řešení je v úspoře vodičů a někdy i úspoře důležitého výpočetního výkonu mikrokontroléru. Na druhou stranu samotný princip vyhodnocování klávesnice je ukryt před studenty v klávesnici, a proto jsem zvolil maticovou klávesnici, aby bylo možné na demonstrační úloze ukázat vyhodnocování klávesnice.

Maticová klávesnice je zobrazena na obrázku 7. Je to základní zapojení pro minimální počet potřebných vstupů a výstupů mikrokontroléru.



Obr. 7 Schéma maticové klávesnice

Při takovémto zapojení připojíme řádky (row) jako výstup z mikrokontroléru a sloupce (col) na vstup mikrokontroléru. Poté vyšleme na první řádek logickou nulu na definované pozici mezi jedničkami. Jestliže se na sloupci objeví nula, znamená to, že je některá klávesa stisknutá. Tuto akci opakujeme až do otestování všech řádků, čímž zjistíme stavy všech tlačítek na klávesnici. Vyhodnocování postupující nuly je zobrazeno v tabulce 3, která je pro matici kláves 4x4 a vyžaduje proto pouze 4 kroky.

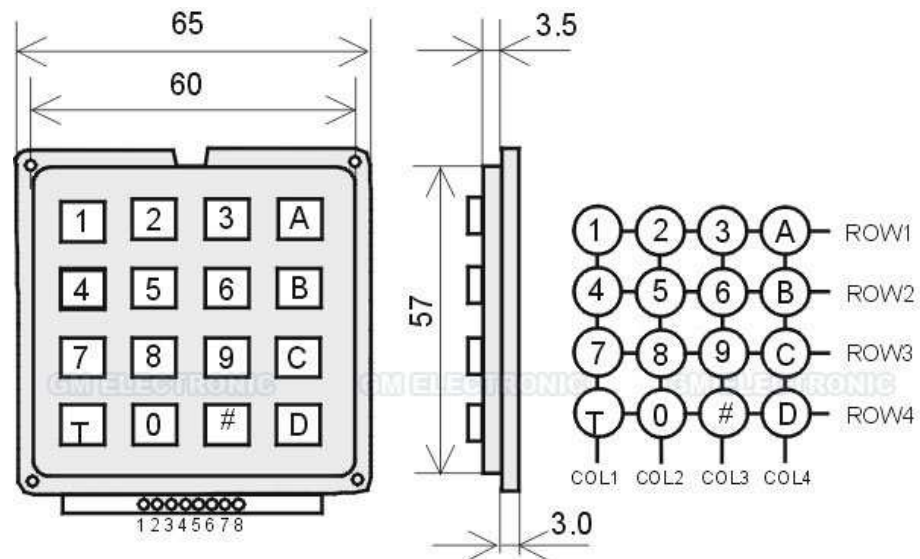
Krok	ROW0	ROW1	ROW2	ROW3	COL0	COL1	COL2	COL3	Tlačítko	Tlačítko	Tlačítko	Tlačítko
1	0	1	1	1	0	1	1	1	S1			
2	1	0	1	1	1	1	1	1				
3	1	1	0	1	1	1	0	1			S11	
4	1	1	1	0	0	1	1	0	S14			S16

Tab. 3 Metoda postupující nuly

Pro laboratorní úlohy jsem vybral klávesnici pod označením F-KV16KEY (zdroj: GME <http://www.gme.cz/cz/index.php?product=637-091>). Je to standardní klávesnice s 16 tlačítky – obrázek 8. Obsahuje spínací kontakty pro maximální napětí 24 V a proud 20 mA, maximální odpor kontaktů je 200 Ω .



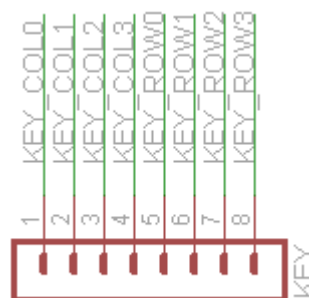
Obr. 8 Maticová klávesnice F-KV16KEY



Obr. 9 Zapojení maticové klávesnice F-KV16KEY

Na obrázku 9 je znázorněno vnitřní zapojení maticové klávesnice F-KV16KEY. Pro připojení k mikrokontroléru byly vybrány brány nazvané PUB a PQS. Brána PUB [0:3] je použita jako vstup do mikrokontroléru COL (sloupce), brána PQS [0:3] je použita jako výstup z mikrokontroléru ROW (řádky).

Připojení maticové klávesnice k mikrokontroléru je zobrazeno na obrázku 10.



Obr. 10 Připojení maticové klávesnice

3.3 PŘIPOJENÍ DISPLEJE

Tak jako klávesnice je základní zařízení pro vkládání požadavků pro zařízení, tak displej je základní zařízení pro interpretaci stavů a dějů ovládaných mikrokontrolérem.

Základním indikačním prvkem je kontrolka – její svit znamená určitý stav. Dalším krokem je uspořádání světel do zobrazení čísel – sedmissegmentovka. Následovalo masové rozšíření alfanumerických displejů, které obsahovaly sadu zobrazitelných znaků. V současné době se však setkáme nejčastěji s grafickými displeji, které vyžadují vlastní vykreslovací jádro, které se stará o zobrazení.

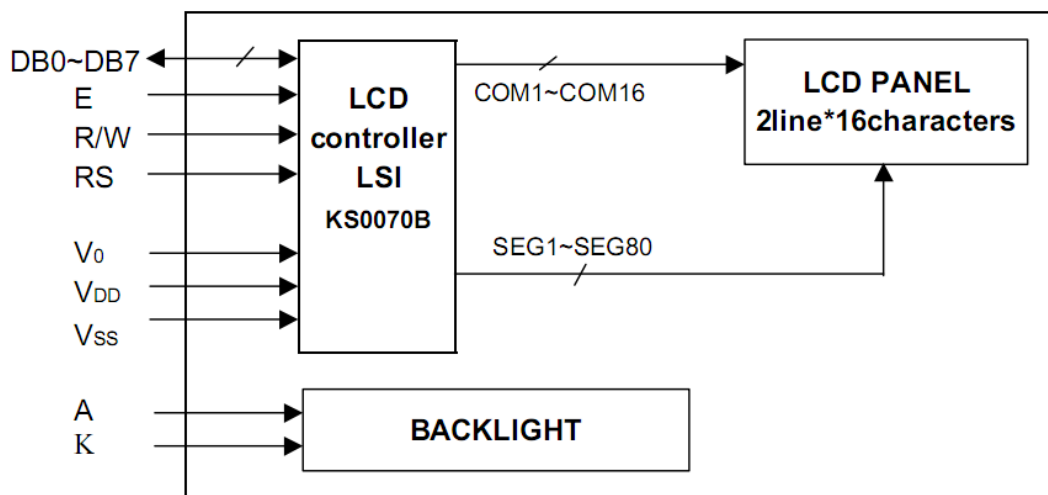
Pro laboratorní úlohy jsem vybral alfanumerický displej s řadičem. Displej tedy obsahuje sadu zobrazitelných znaků a je založen na standardu obvodu HD44780. Vybral jsem ho proto, že tento standard je velice rozšířený a můžeme se s ním setkat u základních zařízení. Vybral jsem typ DEM16217SYH (zdroj: GME <http://www.gme.cz/cz/index.php?product=513-113>), který je zobrazen na obrázku 11. Displej disponuje zobrazením 16 znaků na dva řádky. Displej nemá podsvícení displeje. S podsvícením displeje stoupne cena displeje, a proto jsem zvolil levnější variantu, která by na pracovišti s laboratorní úlohou měla být dostačující.

Výhodou je, že pokud výrobce ponechá stejné rozmístění vývodů pro připojení, můžeme displeje libovolně měnit. V případě potřeby jiného displeje je možné zaměnit vodiče na propojovacím kabelu mezi displejem a deskou s vývody bran mikrokontroléru.

Na obrázku 12 je zobrazeno blokové schéma displeje. Popis vývodů je uveden v tabulce 4.



Obr. 11 Displej DEM16217SYH



Obr. 12 Blokové schéma displeje DEM16217SYH

Číslo	Symbol	Funkce
1	Vss	Zem pro logiku displeje
2	Vdd	Napájení logiky displeje
3	Vo	Napájení zobrazení displeje
4	RS	Volba registru instrukčního (0) nebo datového (1)
5	R/W	Volba čtení (1) a zápisu (0)
6	E	Signál pro povolení vystavených dat na sběrnici
7	DB0	Obousměrná datová komunikace pro posílání a vyčítání dat z řadiče
8	DB1	
9	DB2	
10	DB3	
11	DB4	
12	DB5	
13	DB6	
14	DB7	Možnost zapojení i 4 vývodové brány
15	X	Připraveno pro podsvícení displeje
16	X	

X - nezapojen

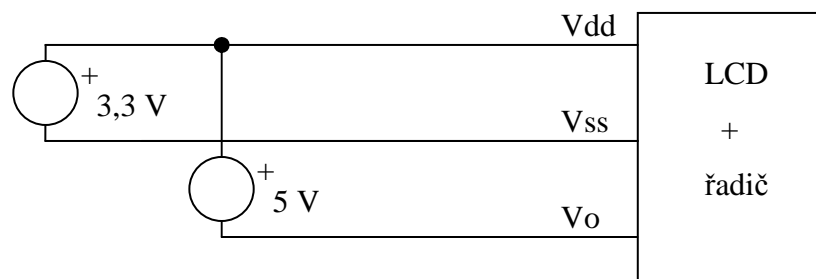
Tab. 4 Seznam vývodů řadiče displeje DEM16217SYH

Protože napájení na vývojovém kitu M52233DEMO je 3,3 V, je důležité, aby byl displej s tímto napájením kompatibilní. Výrobce to řeší tak, že odděluje napájení samotného displeje a logických obvodů řadiče displeje. Pro napájení displeje je vyžadováno napájecí napětí 5 V, řadič může být napájen napětím v rozmezí 2,7 – 5 V. Řadič je kompatibilní se standardem obvodu HD44780. Problém nastává u některých levných displejů s řadičem, který podporuje napájecí napětí pouze 5 V.

Napájení je tedy rozděleno na tři vývody:

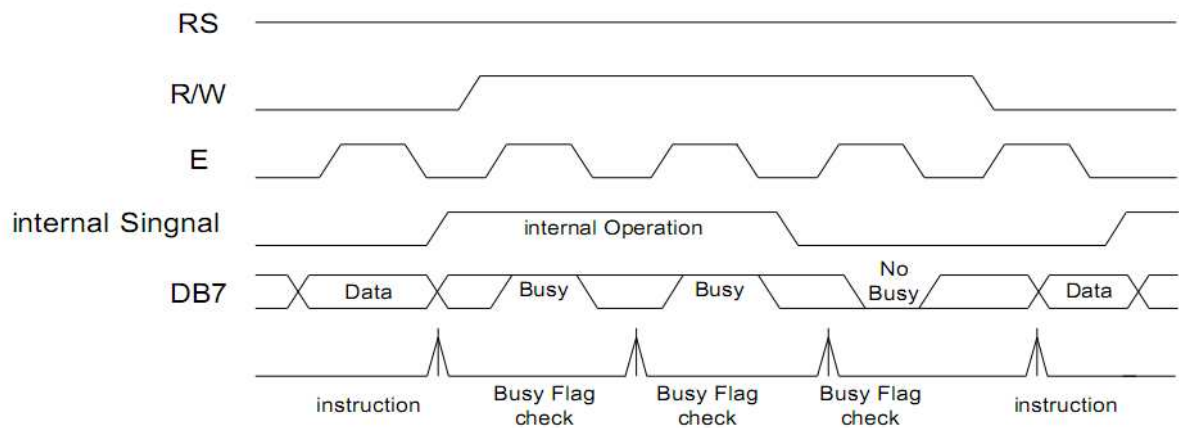
- Vss – zem (GND)
- Vdd – napájení logiky, které je dovolené v rozmezí 2,7 V – 5,5 V
- Vo – napájení displeje (zobrazovací části), typicky 5 V

Zapojení jednotlivých napětí je zobrazeno na obrázku 13. Z důvodu připojení dvou zemí se napájení 5 V i 3,3 V připojí na stejný vývod, a výsledné rozdílové napětí se pak tvoří vůči různé zemi GND. Tím lze obejít nutnost použití napěťového převodníku, který by musel mít výstupní napětí – 2,7 V (3,3 V – 5 V) na vývodu Vo proti zemi GND.



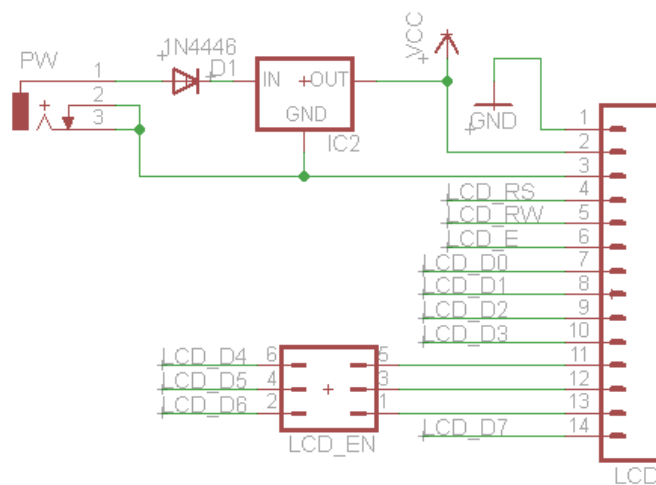
Obr. 13 Připojení napětí pro řadič HD4470

Pro připojení displeje jsem použil datovou bránu PAN [0:7], která bude sloužit jako datová brána pro řadič displeje DB. Tato brána může být v režimu vstupně/výstupním, pokud budeme chtít vyčítat informace z displeje (např. aktuální poloha kurzoru či připravenost displeje). Dále řadič potřebuje připojit tři řídicí signály pro ovládání komunikace mezi displejem a mikrokontrolérem. Pro řídicí signál RW jsem vybral vývod brány PAS [3], pro potvrzovací signál E jsem vybral vývod brány PAS [2], a pro signál výběru registru RS jsem vybral vývod brány PTA [3]. Tyto signály slouží ke kompletní komunikaci mikrokontroléru s řadičem displeje. Na obrázku 14 je ukázka komunikace mezi řadičem a mikrokontrolérem v časovém diagramu.



Obr. 14 Ukázka datové komunikace s řadičem HD4470

Praktické propojení desky s výstupy z mikrokontroléru s alfanumerickým displejem je zobrazeno na obrázku 15. Zapojení obsahuje připojení datových a řídicích signálů včetně napájení displeje. Napájení logiky je 3,3 V a je to napájecí napětí mikrokontroléru. Dále je potřeba napájet displej 5 V. Bohužel vývojový kit M52233DEMO napětí z USB 5 V převádí přímo na desce na 3,3 V, a toto napětí je referenční napájecí napětí pro mikrokontrolér i jeho vstupy a výstupy. Proto jsem pro napájení displeje použil napěťový stabilizátor 7805, na vstupu je i ochranná dioda proti přepólování. Toto externí napájení není nutné odebírat z adaptéru. Existuje možnost vyvedení napájení z počítačového zdroje, který obsahuje napájení 12 V. Další možnost je vyrobit převodník pro USB sběrnici a využívat napětí 5 V na USB.



Obr. 15 Připojení LCD displeje

3.4 SCHÉMA DESKY EXTERNÍCH PERIFERIÍ

Pomocí programu Eagle bylo vytvořeno schéma desky pro připojení externích periferií ke kitu M52233DEMO. Toto schéma je uvedeno v příloze A.

Pomocí stejného programu byla navržena deska plošných spojů, jejíž výkres je v příloze B.

4. LABORATORNÍ ÚLOHY

4.1 ÚLOHA DEMONSTRUJÍCÍ VYUŽITÍ BINÁRNÍCH VSTUPŮ A VÝSTUPŮ

4.1.1 Úvod

Mikrokontrolér obsahuje mnoho konfigurovatelných vstupů a výstupů. Základní paralelní připojení je realizováno pomocí 8bitových bran. Mikrokontrolér MCF52233 obsahuje těchto portů až 13, přičemž ne všechny brány jsou 8bitové. Porty jsou sdílené s dalšími periferiemi, a proto je konečný počet dán počtem volných vývodů. Toto číslo také omezuje velikost pouzdra, které má na kitu M52233DEMO 80 vývodů. Konkrétně tento demo kit má vyvedeny vstupy a výstupy na konektor se 40 vývody, které jsou využitelné pro následující laboratorní úlohy.

První úloha po seznámení s vývojovým prostředím pro programování mikrokontrolérů (pro Freescale procesory je to CodeWarrior) bývá právě úloha pro oživení vstupů a výstupů. Při získání neznámého mikrokontroléru je pro programátora rozchození vstupů a výstupů známka dobrého nastavení. Je potřeba nastavit zdrojový hodinový signál pro procesor a poté co běží, nastavit vstupy a výstupy. U procesorů Freescale bývá nutnost nastavovat u jednotlivých portů směr toku dat – vstupní či výstupní brána. Typická úloha pro tento test je vytvoření blikajícího hada ovládaného jednotlivými tlačítky.

4.1.2 Zadání

Vytvořte světelného hada, který bude ovládán dvěma tlačítky. Při stisku tlačítka TL0 se had posune o jednu pozici doleva, při stisku TL1 o jednu pozici doprava.

Modifikujte program tak, aby běhal cyklicky samostatně. Interval mezi sepnutím jednotlivých LED diod realizujte pomocí čekacích smyček.

4.1.3 Návod

Tato úloha je zaměřena na ovládání vstupních a výstupních bran mikrokontroléru MCF52233. Mikrokontrolér má pouhých 80 vývodů (vzhledem k počtu možných bran), a proto jeden vývod může mít až čtyři funkce. Tyto funkce se definují pomocí přepínače P_xPAR, kde x je označení brány, kterou chceme

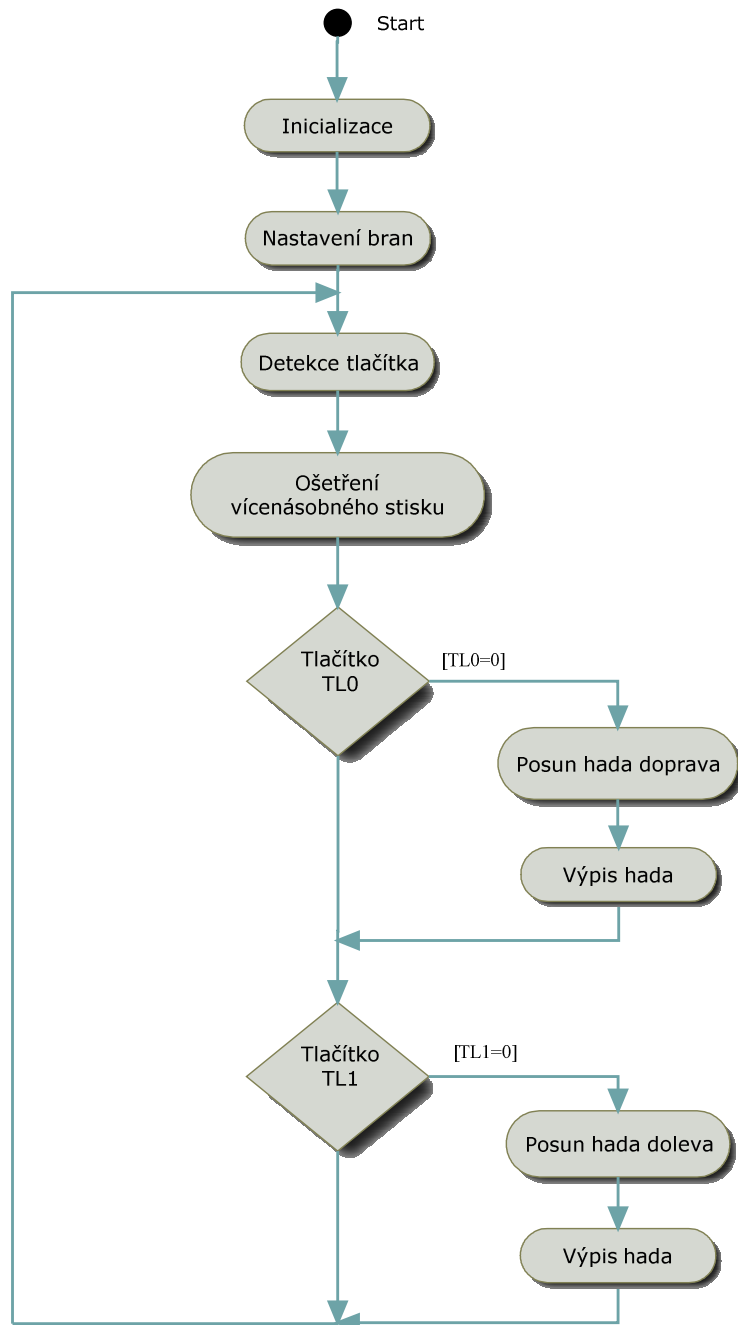
nastavit (například PANPAR). Ve výchozím stavu jsou však všechny vývody nastaveny na režim GPIO – tedy režim vstupně/výstupních bran. Knihovna pro použití bran na mikrokontroléru MCF52233 se jmenuje `MCF52233_GPIO.h`. Zde jsou uvedeny všechny obslužné registry pro řízení jednotlivých bran.

LED diody jsou připojeny k mikrokontroléru na bránu TC. Je nutné nastavit směr toku informací na vývodu brány – zda se jedná o vstup, či výstup. K tomuto účelu slouží ovládací registr DDRTC, který je potřeba nastavit do jedničky, aby vývod byl považován za výstup. Tlačítka jsou připojena na bránu UA. Aby byla tato brána považována za vstup do mikrokontroléru, je potřeba nastavit ovládací registr DDRUA na logickou 0. Tento krok, který je základní inicializací pro každý program, definuje vstupy a výstupy, se kterými už můžeme pracovat v programovacím jazyce C. Poté budeme přistupovat na tlačítka a LED diody na desce externích periférií.

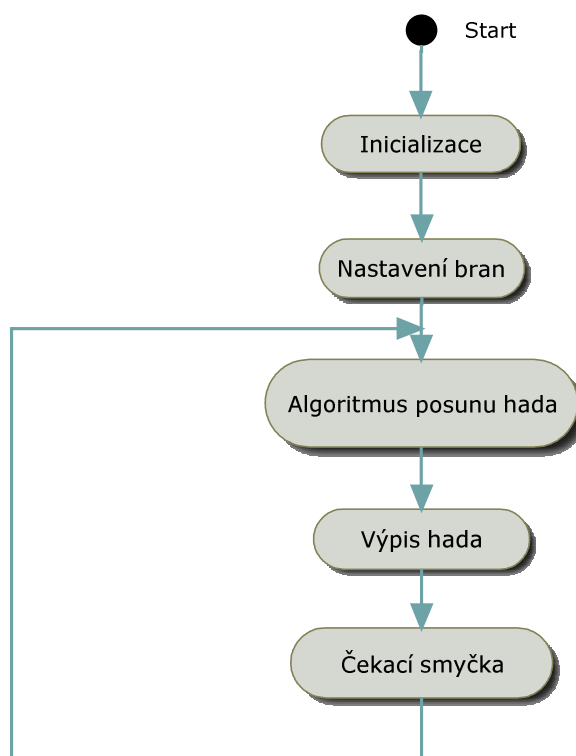
Pro vytvoření světelného hada je možné použít bitový posun v jazyce C, který je interpretován příkazem `>>` pro posun doprava a `<<` pro posun doleva. Můžeme využít jazyka C a zjišťovat sepnutí tlačítka podmínkou. S využitím knihovny `MCF52233_GPIO.h` v prostředí ColdFire jsme schopni zjišťovat stav jednoho vývodu na bráně UA. Ke zjištění stavu tohoto vývodu lze použít maskování `MCF_GPIO_PORTUA& MCF_GPIO_PORTUA_PORTUA0`, které zjistí stav tlačítka TL0. Protože mikrokontrolér je rychlý, nemůžeme vyčítání stavu tlačítka nechat běžet cyklicky – načítali bychom tisíce zmáčknutí na jeden stisk tlačítka. Proto je nutné ošetřit tento stav. Výhodné je použít zapamatování předchozího stavu tlačítka. Poté v podmínce testujeme nejen stav stisknutého tlačítka, ale také zda-li je hodnota jiná než předchozí. Tím detekujeme hranu tlačítka. Na tlačítku bez zámků jsme tímto způsobem schopni detekovat pouze jeden stisk.

Při cyklickém běhu hada je rychlost běhu procesoru tak velká, že je had okem neviditelný. Můžeme použít čekací smyčku, kdy procesor bude čekat. Je to neefektivní, protože procesor nebude zpracovávat jiné výpočty, ale tato volba je jednoduchá a slouží k základnímu testování běhu mikrokontroléru. Jako čekací smyčku můžeme použít `for` cyklus, kdy procesor bude přičítat určitou hodnotu, a po dočítání smyčky opustí. Protože čas nelze odhadnout, musíme konstantu přičítání odpovídající časovému zpoždění odhadnout. Z důvodu omezení maximální velikosti čísel je možné, že je potřeba využít vnořené cykly pro dosažení dostatečně dlouhého čekacího času.

4.1.4 Realizace



Obr. 16 Vývojový diagram programu pro ovládání hada tlačítky



Obr. 17 Vývojový diagram programu světelného hada s čekací smyčkou

4.2 ÚLOHA DEMONSTRUJÍCÍ VYUŽITÍ ČASOVAČE

4.2.1 Úvod

Čítače a časovače jsou součástí systému General Purpose Timer (GPT). Obsahují čtyři 16bitové čítače, kterým lze definovat rychlost 7bitovou děličkou vnitřního kmitočtu. Systém obsahuje volně běžící čítače s definovatelným přerušením i pulsní akumulátory. Tyto čítače mají různé módy a jsou různě modifikovatelné. Celé obsáhlé schéma se nachází v datasheetu k mikrokontroléru [1]. Režimy čítačů dovolují vytvářet časové úseky pomocí přednastavitelných registrů. Tyto registry procesor v definovaných časových intervalech inkrementuje a při dočítání do maximální hodnoty vyvolá přerušení. V kombinaci volby časového intervalu čítání a nastavení čítacího registru máme možnost nastavit široký rozsah časových smyček.

V praxi se při programování mikrokontroléru setkáme s problémem časování často. Pro připojení na sběrnici či obsluhu periférií je potřeba dodržet časové úseky definované standardem přenosu.

Spolu s touto úlohou je spojen problém ohledně obsluhy přerušení. Aby totiž byl dodržen definovaný čas časovače, procesor po dočítání přednastavené hodnoty vyvolá přerušení. A právě v této obsluze přerušení je potřeba správně vyhodnotit a obsloužit dočítání časovače.

Pro přerušovací rutinu platí několik zásad:

- 1) Na začátku zákaz přerušení dané periferie (aby nedošlo k vícenásobnému volání)
- 2) Vyhodnocení zdroje přerušení podle příznakových registrů
- 3) Někdy je vyžadováno vynulovat příznak přerušení
- 4) Po provedení rutiny přerušení následuje opětovné povolení přerušení

4.2.2 Zadání

Vytvořte pomocí časovače definovanou čekací smyčku 200 ms. Pro správné vyhodnocování časového intervalu zpracovávají časový úsek v obsluze přerušení.

Modifikujte předchozí zadání (světelného hada) a místo čekací smyčky využijte funkci časovače.

4.2.3 Návod

Úloha se zabývá obsluhou přerušení a využití časovačů. Mikrokontrolér MCF52233 obsahuje širokou škálu použitelných čítačů a každý má své výhody. Já jsem si zvolil jednoduchý 16bitový čítač PIT0. Tento čítač se inicializuje nastavením hodnoty do registru MCF_PIT0_PMR, kterou čítač cyklicky dekrementuje. Po dočítání registru do nuly se vyvolá přerušení. Čítač disponuje funkcí autoreload – tedy při dočítání je opět nastavena původní hodnota registru MCF_PIT0_PMR. Nastavení časové hodnoty pro čítač tedy spočívá ve správném nastavení registru MCF_PIT0_PMR. Ten se vypočítá podle vztahu 3:

$$t = \frac{2}{60\text{MHz}} \times 2^{\text{PRE}[3:0]} \times \text{PMR}[15:0] \quad (3)$$

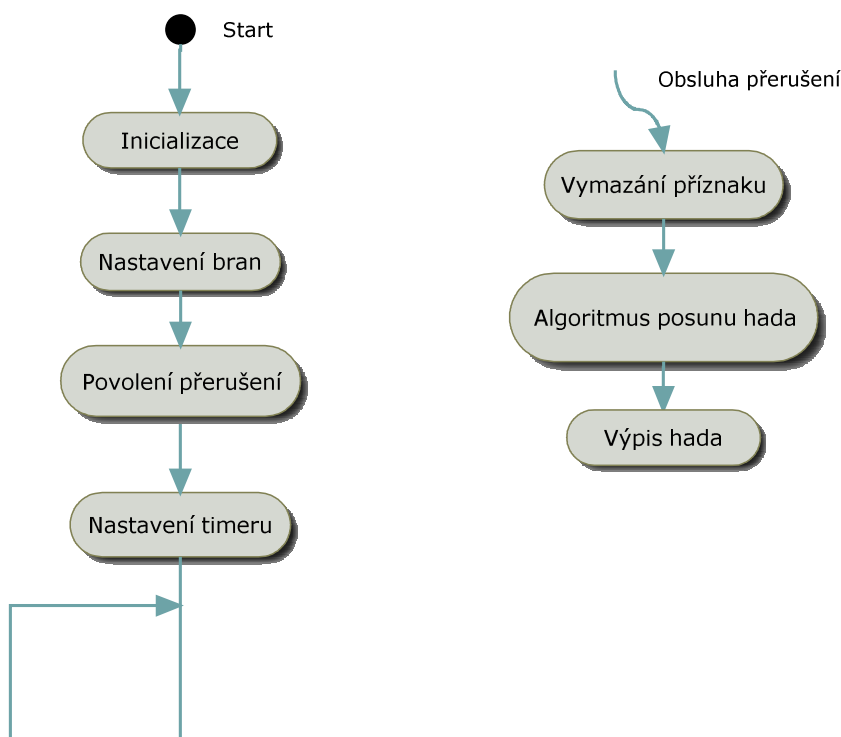
Pro $t = 500$ ms jsem zvolil $\text{PRE} = 7$, $\text{PMR} = 0xB71B$

$$t = \frac{2}{60\text{MHz}} \times 2^7 \times 46875 = 200\text{ms}$$

Další součástí úlohy je správné nastavení přerušení. Protože mikrokontrolér MCF52233 je 32bitový, je připraven pro běh operačních systémů v podobě systému μ Linux. Proto obsluha přerušení u tohoto procesoru připomíná obsluhu přerušení u počítačů. Je zde nutné zavést vektor přerušení, bezpečně jej prohodit a povolit masky jednotlivých přerušení. Tento proces vychází z obsluhy přerušení u procesoru M68000 a je obsažen ve funkcích, které jsou součástí vypracované úlohy.

Při obsluze přerušení je potřeba zadat vektor přerušení, který se dále zpracovává knihovními funkcemi. Tento vektor přerušení zjistíme z datasheetu [1], pro časovač PIT0 je to číslo 55. Inicializace přerušení je provedeno funkcí `mcf5xxx_set_handler(64 + 55, (uint32)Gpt_IsrHandler);`. Samotná obsluha přerušení je ve funkci `__declspec(interrupt) void Gpt_IsrHandler(void)`. Při vstupu do obsluhy přerušovací rutiny je důležité vymazat příznak přerušení. To můžeme provést příkazem `MCF_PIT0_PCSR |= MCF_PIT_PCSR_PIF;`. Poté následuje samotná obsluha přerušení. V této úloze to může být poslání hada na výstup, který byl vytvořen v předchozí úloze.

4.2.4 Realizace



Obr. 18 Vývojový diagram úlohy časovače

4.3 ÚLOHA DEMONSTRUJÍCÍ VYUŽITÍ ČÍTAČE

4.3.1 Úvod

Čítače jsou součástí systému GPT. Obsahují funkce output compare a input compare. Funkce input compare zachytí hodnotu volně běžícího čítače po detekci hrany signálu na vstupu mikrokontroléru. Tímto způsobem se dá například měřit délka impulsu. Opakem toho je funkce output compare, která při načítání definované hodnoty vyše definovanou úroveň na výstup mikrokontroléru. To umožňuje například generování impulsů definované časové šířky.

Mikrokontrolér MCF52233 obsahuje rozšíření čítačů v podobě systému reálných hodin (RTC). Tento systém obsahuje vestavěný generátor hodin. Ne hodin systémových (CLK), ale běžných pro měření časů s podporou sekund, minut, hodin a dokonce i dní. Systém podporuje nastavení alarmů. To umožňuje vyvolávat přerušování po definovaných časových úsecích.

4.3.2 Zadání

Pomocí systému reálných hodin (RTC) vytvořte stopky, které doběhnou nastaveného časového intervalu budou signalizovat rozsvícením LED diody. Pomocí tlačítka TL0 zvyšujete přednastavený časový interval a pomocí tlačítka TL1 jej snižujete. Potvrzení volby a restart stopek ovládejte pomocí tlačítka TL2.

4.3.3 Návod

Mikrokontrolér MCF52233 obsahuje systém reálných hodin (RTC) přímo na čipu. Tato úloha je zaměřena na ovládání systému reálných hodin pomocí tlačítek. Hned po startu mikrokontroléru běží hodiny, pokud je systém RTC povolen. Pro vytvoření stopek můžeme využít funkci alarmu, kdy systém RTC vyhodnotí dosažení času alarmu příznakem. Tento příznak lze vyhodnocovat v přerušování. Při prvotním seznámení s úlohou postačí vyhodnocování v průběhu nekonečné smyčky programu. Správné ošetření časového údaje ale vyžaduje použít obsluhu přerušování, která se pro systém RTC nachází na vektoru 63.

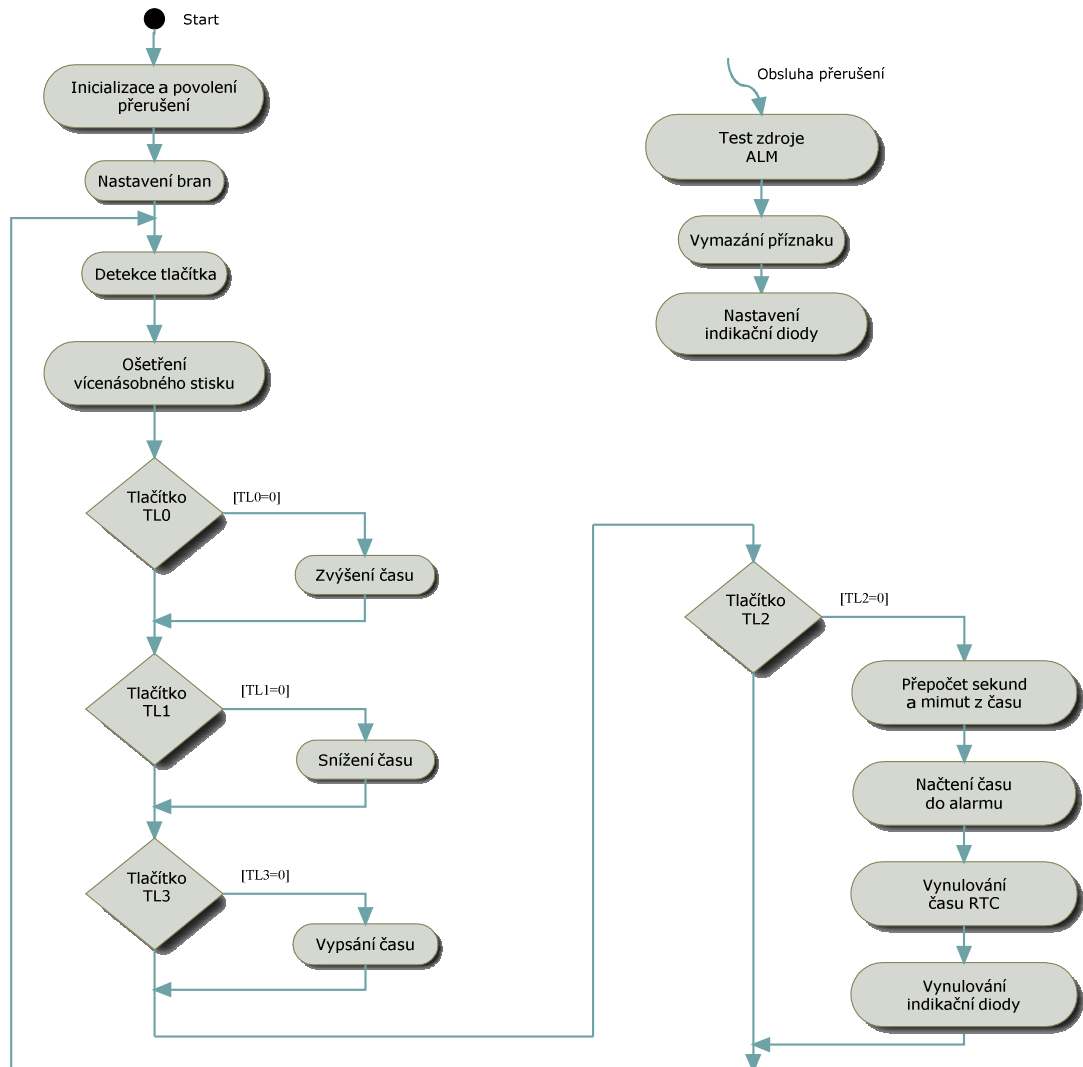
Pro detekci tlačítek použijeme ošetření zákmitů z předchozí úlohy. Jednotlivými tlačítky je potřeba nastavit požadovaný čas čítání – TL0 čas zvyšuje, TL1 čas snižuje. Nastavený čas je možné tlačítkem TL3 poslat na terminál v počítači, pomocí vestavěné funkce printf. Při stisku tlačítka TL2 dojde k odstartování stopek. V této chvíli je potřeba nastavit požadovaný čas do alarmu. Je zde také nutné

přepočíst čas do minut a sekund, což lze vypočítat pomocí celočíselného dělení a zbytku po něm. Minuty jsou pak $cas / 60$ a sekundy $cas \% 60$. Aby bylo možné využít funkci alarmu, je potřeba volně běžící hodiny v systému RTC vynulovat.

Při dočítání přednastaveného času, systém RTC nastaví příznak v registru MCF_RTC_RTCISR_ALM. Příznak je vyhodnocován v obsluze přerušení s vektorem přerušení 63. Na začátku obsluhy je nutné vyhodnotit zdroj přerušení. Tento vektor přerušení je totiž společný pro celý systém RTC a je nutné vybrat pouze událost alarmu ALM. Poté je potřeba příznak vynulovat příkazem $MCF_RTC_RTCISR \mid = MCF_RTC_RTCISR_ALM;$. Na konci obsluhovací rutiny je rozsvícení LED diody jako indikaci dočítání našich stopek.

Při opětovném stisku tlačítka TL2 dojde k načtení zvolené hodnoty cas do alarmu systému RTC. Indikační LED dioda se zhasne a probíhá opětovné stopování času.

4.3.4 Realizace



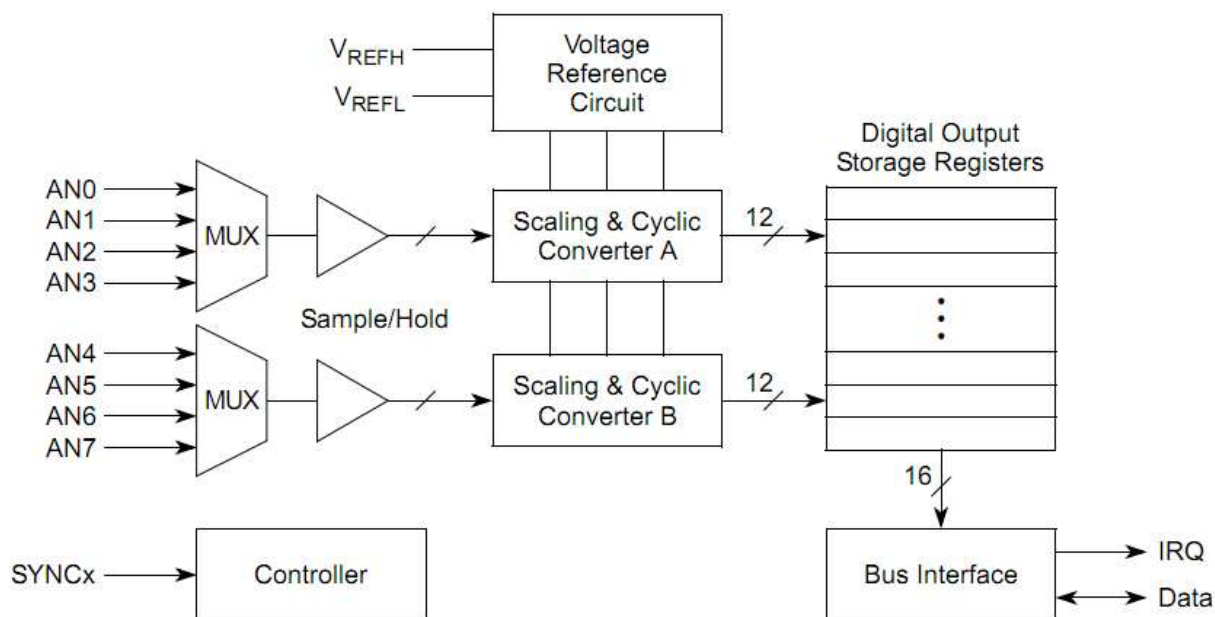
Obr. 19 Vývojový diagram programu stopek

4.4 ÚLOHA DEMONSTRUJÍCÍ VYUŽITÍ A/D PŘEVODNÍKU

4.4.1 Úvod

Pro zpracovávání analogových přenosů disponuje mikrokontrolér A/D převodníky. Převodníky slouží k převedení napětí na číslo, které odpovídá poměru měřeného napětí vůči rozsahu převodníku. Lze jimi například vyhodnocovat odpor potenciometru nebo zrychlení změřené snímačem MMA7260Q (3osý akcelerometr na desce vývojového kitu M52233DEMO).

Na obrázku 20 je blokové schéma A/D převodníku v mikrokontroléru MCF52233. Tento rychlý převodník obsahuje 8kanálový multiplex přivedené na dva nezávislé vzorkovače a tvarovače (sample and hold), které předávají hodnotu do 10bitových nebo 12bitových převodníků.



Obr. 20 Blokové schéma A/D převodníku mikrokontroléru MCF52233 [1]

Při této úloze mohou studenti poznat výhody programu FreeMaster, popsaného v kapitole [2.5.2](#). Samotný program pak obsahuje měření, které zpracuje a uloží do proměnných. Vyjádření měření a zobrazení průběhů je pak realizováno pomocí FreeMasteru. Tím lze získat představu o rozsahu měřených hodnot, jejich minimálních a maximálních hodnotách. Poté je možné provést normalizaci měřených hodnot například do rozsahu <0;1000>.

Při testování akcelerometru je nutné uvádět vývojový kit M52233DEMO do pohybu, aby bylo možné detekovat zrychlení. To je ovšem nevýhodné pro laboratorní úlohy, kdy by studenti museli hýbat připojeným kitem ve vzduchu spolu s připojovacími kabely. Proto je vhodné využít k měření zrychlení přitažlivou sílu země. Tato síla směřuje vždy kolmo k zemi o velikosti $g = 9,81 \text{ m/s}^2$. Proto můžeme na trojosém akcelerometru sledovat rozložení této síly do tří vektorů, které jsou vyhodnocovány akcelerometrem. Tudiž pak můžeme pouhým překlápěním vůči hraně vývojového kitu sledovat změnu gravitačního zrychlení.

4.4.2 Zadání

Vytvořte program pro vyhodnocování polohy potenciometru připojeného k A/D převodníku. Pro vyhodnocování použijte měřítko tak, aby rozsah výstupních hodnot byl $\langle 0;1000 \rangle$.

Dále vytvořte program pro vyhodnocování změny zrychlení na 3osém akcelerometru MMA7260Q.

4.4.3 Návod

V průmyslovém nasazení se často setkáme s měřením napětí mikrokontrolérem. Z toho důvodu je potřeba správně obsloužit A/D převodníky. Mikrokontrolér MCF52233 obsahuje 8 převodníků, které lze využít i jako 4 v diferenčním zapojení. Při měření potenciometru je potřeba na desce externích periferií přepnout přepínač připojení potenciometru a odpojení datové brány pro komunikaci s displejem (přepínač POT_LCD_SW). Pro správnou funkčnost měření potenciometru je také nutné odpojit potenciometr na kitu M52233DEMO přepínače POT_EN.

Pro měření jsem zvolil mód Once Sequential – tedy jedno měření kanál po kanále. A/D převodník má vstupy připojeny na bránu AN. Ovšem funkce vstupu do převodníků je druhotná – primární výchozí je použití jako vstupy a výstupy. Proto je potřeba přepínačem PANPAR zvolit vývody, které chceme připojit k převodníku. Tento úkon je realizován příkazem

```
MCF_GPIO_PANPAR | =  
MCF_GPIO_PANPAR_PANPAR0 |  
MCF_GPIO_PANPAR_PANPAR4 | MCF_GPIO_PANPAR_PANPAR5 |
```

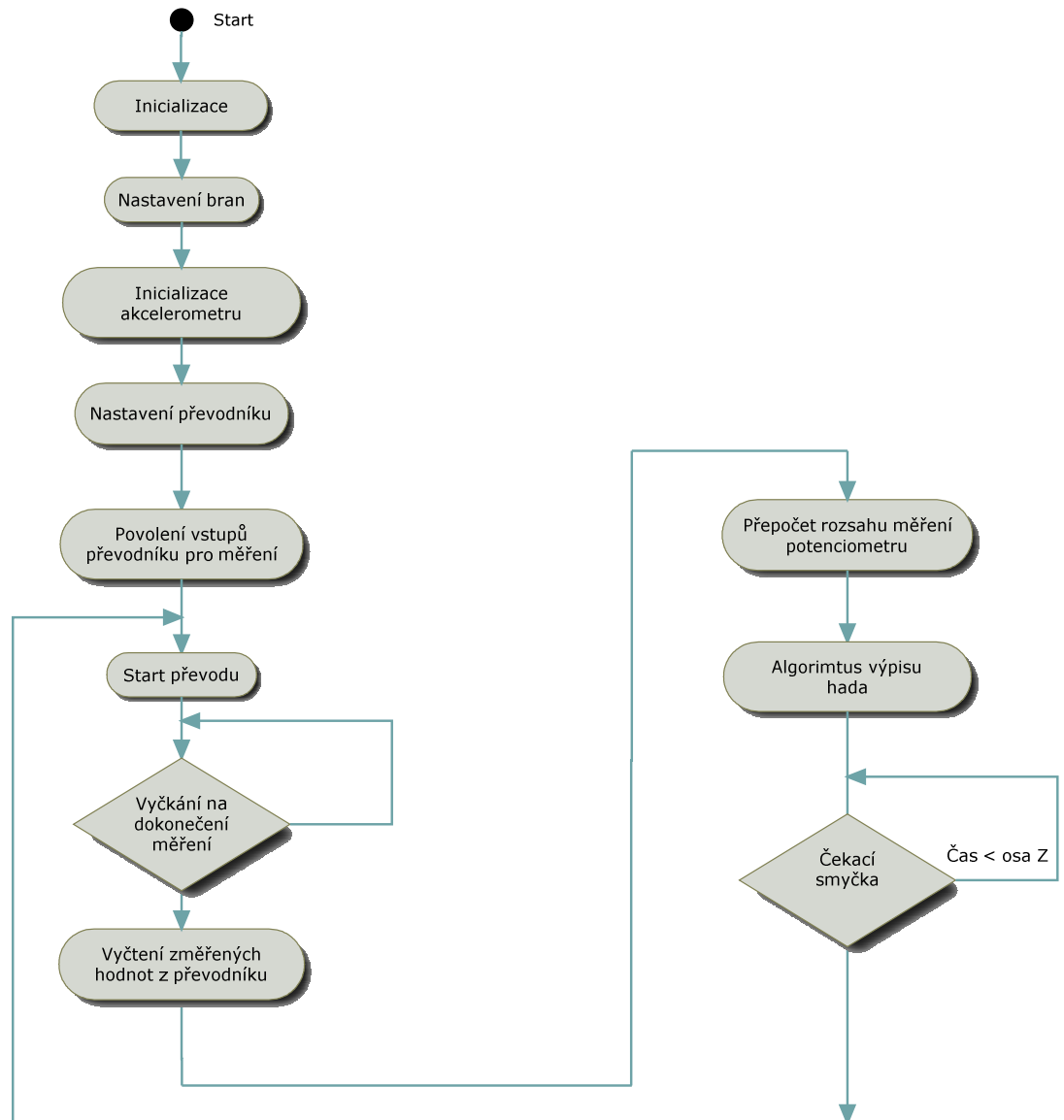
MCF_GPIO_PANPAR_PANPAR6; Tímto jsou připojeny vstupy pro snímání změn na 3osém akcelerometru MMA7260Q (vývody 4,5 a 6) a měření potenciometru (vývod 0). Pro měření je potřeba odpojit LCD displej přepínačem LCD_EN.

Samotné měření na převodníku je potřeba spustit příkazem `MCF_ADC_CTRL1 |= MCF_ADC_CTRL1_START0;`. Protože měření na převodníku vyžaduje určitý čas, je potřeba v nekonečné smyčce hlavního programu vyčkat na měření. Toto je jednodušší řešení, lze využít i obsluhu měření v přerušení. Jednoduchá čekací smyčka čeká na příznak konce měření `while (MCF_ADC_ADSTAT & (MCF_ADC_ADSTAT_CIP1)){};`. Poté můžeme ze stavových registrů převodníku vyčíst naměřené hodnoty – například příkazem `x=MCF_ADC_ADRSLT4;`

Tímto postupem jsme schopni vyčítat měřené napětí na převodníku a naměřená data můžeme dále zpracovávat v programu. Například pro měření na potenciometru můžeme otáčení normalizovat na rozsah $\langle 0;1000 \rangle$. To určíme tak, že změříme maximální a minimální hodnotu, kterou můžeme při otáčení potenciometru získat. Unifikovaná hodnota v definovaném rozsahu se pak vypočítá podle vztahu 4:

$$x = \frac{Max - Min}{1000} - Min \quad (4)$$

4.4.4 Realizace

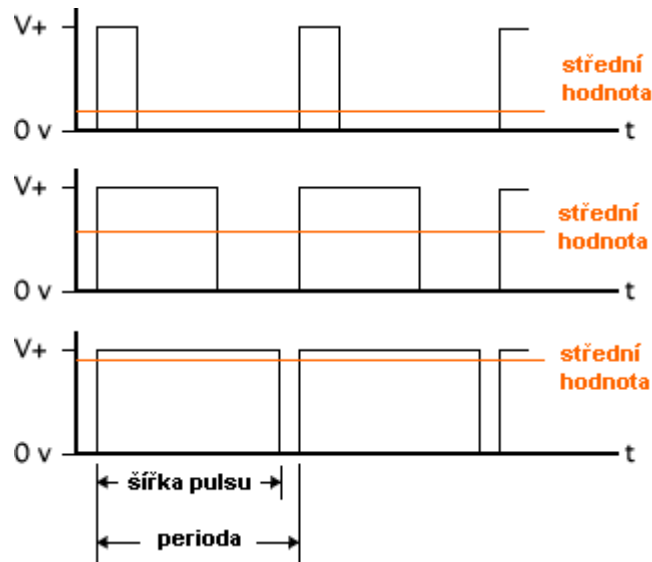


Obr. 21 Vývojový diagram úlohy měření AD převodníků

4.5 ÚLOHA DEMONSTRUJÍCÍ VYUŽITÍ PWM MODULACE

4.5.1 Úvod

Pulsně šířková modulace (PWM) se používá například pro řízení motorů. Využívá efektu, že výkon odpovídá ploše budícího signálu. Změnou šířky impulsu jsme schopni na konstantní amplitudě regulovat výkon do zátěže, který odpovídá střední hodnotě signálu. Princip změny střídy budícího signálu je zobrazen na obrázku 22.



Obr. 22 Průběh pulsně šířkové modulace

Většina mikrokontrolérů obsahuje PWM modulátory na čipu, konkrétně MCF52233 má 8kanalový a 8bitový PWM časovač. Šířku impulsu je možno pomocí registru nastavit v rozmezí 0 - 100%. Periodu impulsu lze nastavit 8bitovým časovačem.

PWM modulace se nepoužívá jen pro regulování výkonu dodávaného do zátěže. Své místo si našla také u přenosů. Například některé snímače obsahují PWM výstup, který je odolný vůči rušení. S PWM modulací se také setkáme v měřicí technice u inteligentních snímačů pro autokalibraci - nastavování sigma-delta převodníků.

4.5.1 Zadání

Vytvořte program pro ovládání barvy dvoubarevné ledky, připojené na výstup PWM mikrokontroléru.

Ovládejte barvu pomocí potenciometru s využitím předchozí úlohy (A/D převodník).

4.5.2 Návod

Princip pulsně šířkové modulace PWM je v této úloze demonstrován na dvoubarevné LED diodě. Pomocí definované šířky pulsu jsme schopni řídit výkon dodávaný do diody. Pokud budeme řídit i druhou barvu na LED diodě, jsme schopni vytvořit určité míchání barev. LED dioda je připojena na PWM1 a PWM3. Je to opět duální funkce, a proto je nutné jako u převodníku nastavit jejich funkci příkazem

```
MCF_GPIO_PTAPAR=
```

```
MCF_GPIO_PTAPAR_PTAPAR1(0x3) | MCF_GPIO_PTAPAR_PTAPAR0(0x3);
```

Základní nastavení PWM spočívá v délce impulsu a frekvenci pulsů. Tyto ovládací registry je možné nastavit

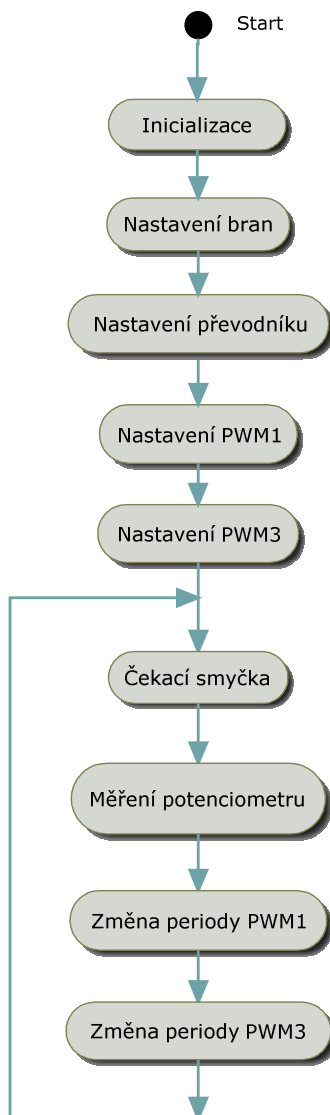
```
MCF_PWM_PWMPER1 = MCF_PWM_PWMPER_PERIOD(0x64);
```

```
MCF_PWM_PWMDTY1 = MCF_PWM_PWMDTY_DUTY(0xa);
```

 kde period je frekvence a duty je šířka pulsu.

S využitím předchozí úlohy AD převodníků změříme nastavení potenciometru. Protože výsledek měření je v rozsahu 0 až 0xFF, můžeme hodnotu potenciometru přímo předávat jako požadovanou šířku pulsu PWM. Při použití této šířky pulsu pro zelenou barvu a doplněk do celé periody pro červenou barvu, získáme plynulý přechod od zelené do červené barvy. Tento přechod je pak ovládán potenciometrem.

4.5.3 Realizace



Obr. 23 Vývojový diagram úlohy PWM modulace

4.6 ÚLOHA PRO KLÁVESNICI

4.6.1 Zadání

Napište funkci, která bude číst znaky z klávesnice. Znaky vyhodnocujte a zobrazujte na displej.

Všechna čísla na klávesnici vypisujte na displej. Při stisku klávesy „A“ zapište na displej mezeru. Při stisku klávesy „B“ přejděte na nový řádek. Stisk klávesy „C“ způsobí vymazání displeje. Klávesou „D“ smažte poslední znak.

4.6.2 Návod

Klávesnice je připojena na brány PUB a PQS. Pro vyčítání je vhodné použít metodu postupující nuly, která je zobrazena v tabulce 1. Podle předchozích nastavení je potřeba nastavit bránu PUB jako vstup do mikrokontroléru COL (sloupce klávesnice) a bránu PQS jako výstup z mikrokontroléru ROW (řádky klávesnice).

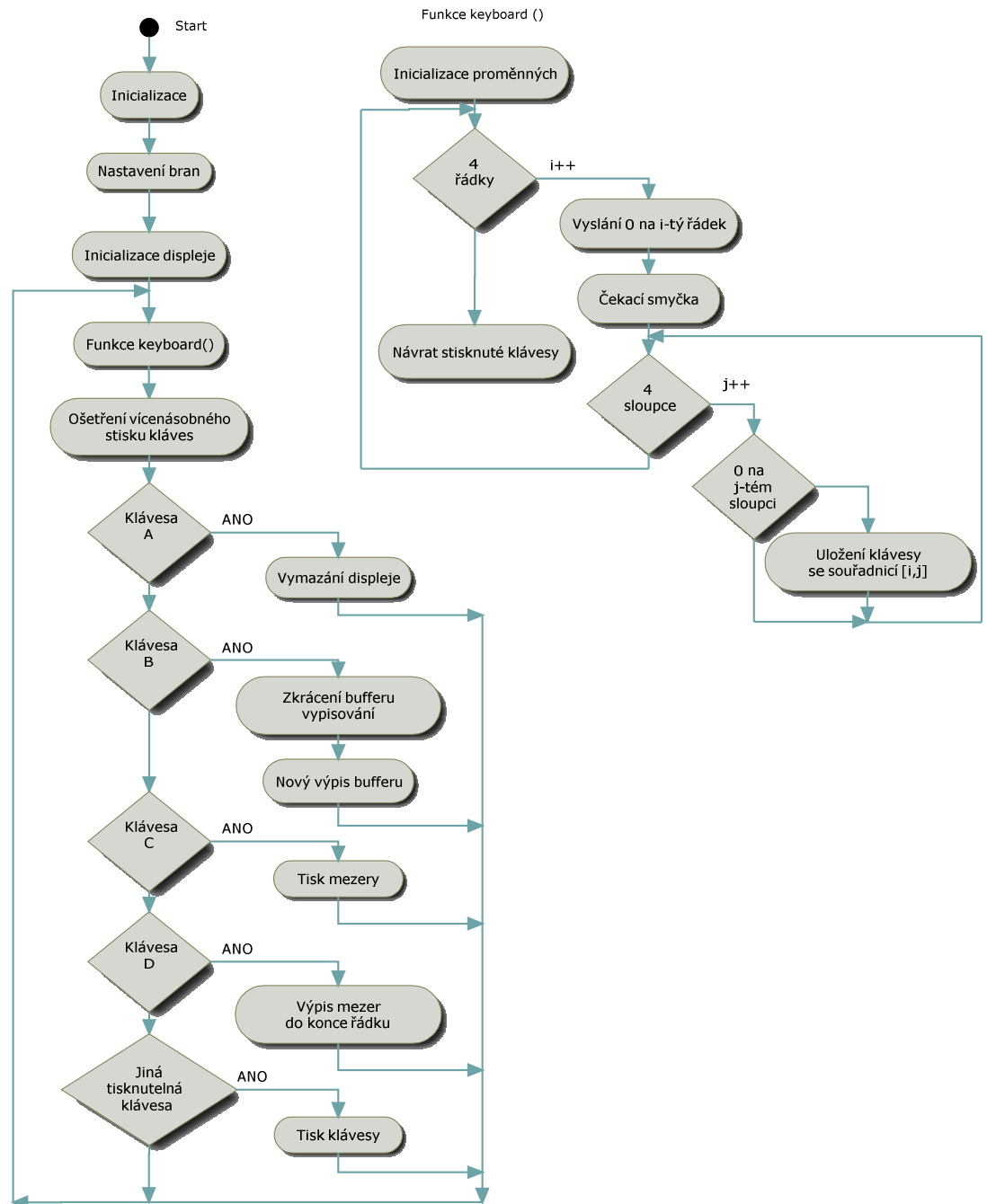
Vyčtení znaku z klávesnice je výhodné umístit do funkce, která vrátí znak tlačítka na klávesnici, nebo 'x', pokud není stisknuto žádné tlačítko. Pro klávesnici se šestnácti tlačítky je potřeba provést 16 kroků pro vyčtení všech tlačítek. První cyklus probíhá čtyřikrát a obsahuje další vnořený cyklus. První cyklus vysílá logickou nulu na výstup `MCF_GPIO_PORTQS=~(0x0|(1<i))`. Protože procesor standardně pracuje na 60 MHz, dochází při takové rychlosti k chybám vyčítání tlačítek. Z toho důvodu následuje čekací smyčka, aby byly vysílané logické nuly detekovatelné. Vnořený cyklus se také provádí čtyřikrát. Jeho účelem je zkoumat, zda na některém vstupu se nenachází nula z výstupu, propojená přes zmáčknuté tlačítko. Test je realizován `if(((MCF_GPIO_SETUB>>j)&0x01)==0)`. Pokud se tak stane, tlačítko je naadresováno aktuální pozicí obou cyklů. Například klávesnici 5 odpovídá kombinace `if(i==1&&j==1) key='5'`.

Pokud tedy cyklicky voláme funkci pro čtení klávesnice, získáváme jako odpověď aktuální zmáčknuté tlačítko. Ovšem při vyčítání v nekonečné smyčce dostáváme řadu stejných tlačítek, které jsou zmáčknuté a vyhodnocené mikrokontrolérem. Proto je potřeba reagovat pouze na změnu tlačítek, abychom získali zmáčknuté tlačítko pouze jednou, nezávislé na délce jeho stisknutí. To je možné realizovat instrukcí `if((klav!='x')&&(oldklav!=klav))`, kde `klav` je aktuální tlačítko a `oldklav` tlačítko zjištěné předchozím testem. Teprve potom můžeme předat stisknuté tlačítko dále ke zpracování.

Další část se zabývá samotným zpracováním jednotlivých kláves spojené s funkcí ovládní displeje. Klávesa *A* smaže displej zavoláním funkce `clr_scr()`. Pro funkci klávesy *B*, která má smazat znak, je potřeba vytvořit ve funkci displeje proměnnou `buffer`, která obsahuje to, co je zobrazeno na displeji. Díky tomu stačí vymazat poslední znak z proměnné `buffer`, displej vymazat a celý obsah opět vypsát na displej. Klávesa *C* vypíše na displej mezeru. Stiskem klávesy *D* se provede náročnější úkon odřádkování. Je důležité, aby během tohoto úkonu žádný znak nebyl či nepřibyl a fungoval na obou dvou řádcích. Pokud není chyba ve správném počítání aktuální pozice kurzoru na displeji a proměnné `buffer`, stačí vypsát

do konce řádku mezery. Výpis mezer je definován příkazem `while (kurzor%16!=0)`, kde kurzor je aktuální poloha kurzoru na displeji.

4.6.3 Realizace



Obr. 24 Vývojový diagram úlohy vyčítání znaků z klávesnice

4.7 ÚLOHA PRO OVLÁDÁNÍ DISPLEJE

4.7.1 Zadání

Vytvořte knihovnu pro ovládání displeje. Knihovna bude obsahovat funkce:

<code>disp_init</code>	pro inicializaci displeje
<code>disp_clr</code>	pro vymazání displeje
<code>send_char</code>	pro odeslání znaku na displej
<code>disp_gotoxy</code>	pro nastavení pozice pro výpis na displeji

4.7.2 Návod

Tato úloha zahrnuje základní komunikaci mikrokontroléru s řadičem HD4470. Propojení je realizováno přes datovou bránu AN [0:7]. Při použití displeje je nutné povolit datový vodič AN [0], který je společný s potenciometrem (POT_LCD_SW). Řídicí signál ovládání směru komunikace RW je na PAS [3], potvrzovací signál E je na PAS [2], a signál výběru registru RS je na PTA [3]. Přepínač LCD_EN odpojí od brány LCD v případě, že chceme vyhodnocovat napětí z akcelerometru MMA7260Q.

Základní stavební prvek komunikace mikrokontroléru s displejem je odesílání příkazů pro displej. Pro začátek postačí dva druhy příkazů – posílání znaků a řídicí příkazy. Jelikož jsou tyto funkce základní, jsou definovány jako funkce se vstupním parametrem vysílaného příkazu. Funkce `send_char` vysílá tisknutelné znaky na displej. Na začátku se nastaví signály $RS = 1$ a $RW = 0$. Poté se vyšle požadovaný znak jako ASCII číslo na bránu AN. Nyní je potřeba pomocí povolovací E signálu vytvořit impuls na znamení, že data jsou na bráně AN připravena a řadič displeje je může vyčíst. To se provede nastavením $E = 1$, poté následuje čekací smyčka vytvářející impuls a poté se nastaví $E = 0$. Funkce `send_com` pro vyslání ovládacího příkazu pro displej je totožná, ovšem na začátku se nastaví $RS = 0$ na znamení, že se jedná o ovládací příkaz, a ne tisknutelný znak.

Na konci obou funkcí je test připravenosti displeje, aby funkce předali řadič displeje ve stavu schopném přijímat příkazy. Tato funkce je nazvána `isbusy`. Protože mikrokontrolér vyčítá informace od řadiče displeje, je nutné nastavit bránu AN jako vstupní. Dále se nastaví pro vyčítání „busy flag“ $RS = 0$ a $RW = 1$. Funkce vytváří pulsy povolovacího signálu E a přitom zkoumá odpověď řadiče. Nejprve nastaví $E = 0$, po časové smyčce je nastaveno $E = 1$. Celý cyklus je opakován do té doby, než displej vynuluje „busy flag“. Je to provedeno příkazem

`while(MCF_GPIO_SETAN & MCF_GPIO_SETAN_SETAN7)`, protože signál „busy flag“ se nachází na posledním bitu brány AN. Tímto je zaručeno správné předávání příkazů řadiči displeje.

Aby bylo možné displej použít, je nutné jej inicializovat. Dle datasheetu displeje [3] je definována posloupnost ovládacích příkazů, kterými lze konfigurovat displej. Jako první je vyslán příkaz `send_com(0x3C)`, který inicializuje samotnou osmibitovou komunikaci s řadičem. Následuje `send_com(0xE)`, který provede start displeje a nastaví kurzor na displeji. Poté příkazem `send_com(0x1)` displej vymažeme. Jako poslední je nastavení `send_com(0x6)`, které definuje mimo jiné posun kurzoru doprava.

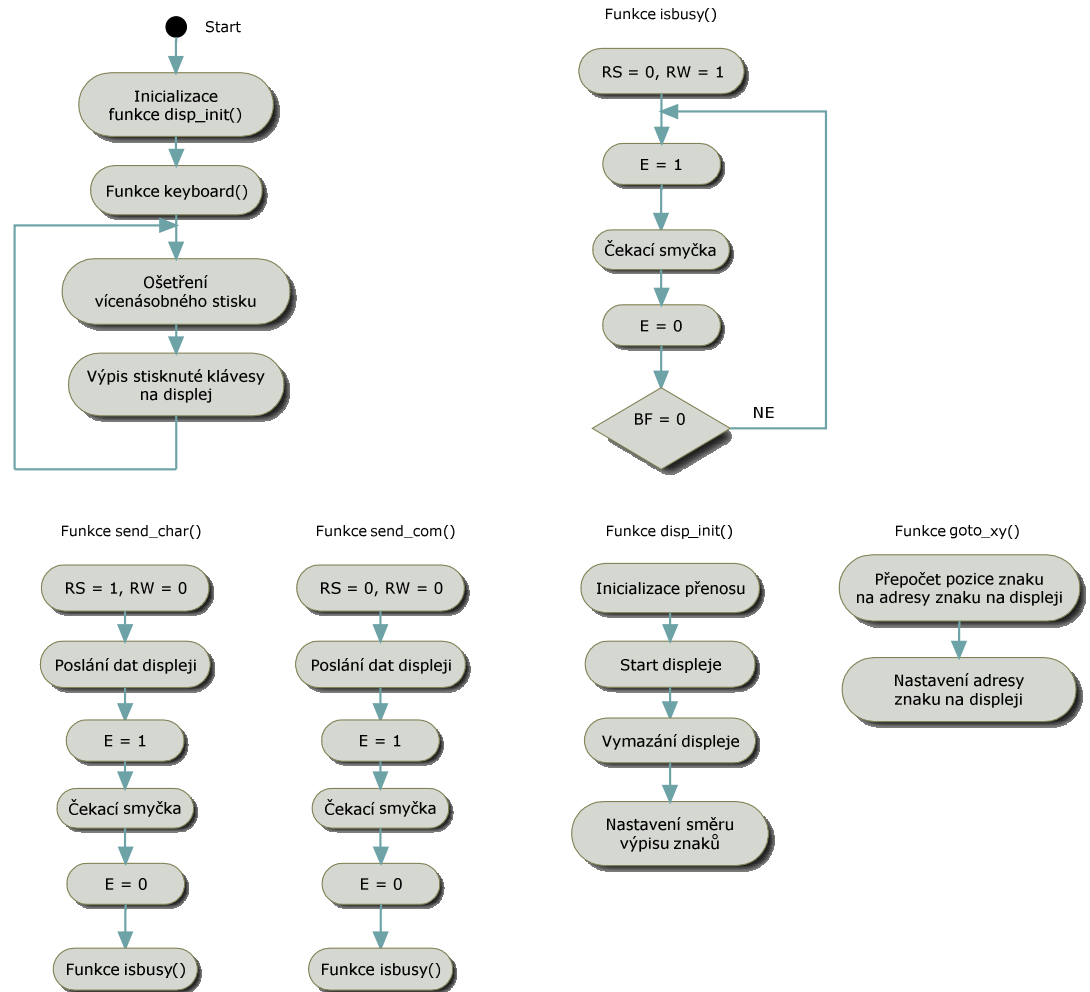
Dle zadání funkce `clr_scr` pro výmaz displeje je tedy jednoduchá, obsahuje pouze jediný ovládací příkaz pro výmaz displeje `send_com(0x1)`. Funkce pro nastavení libovolné polohy na displeji `goto_xy` je také řešena pomocí ovládacího příkazu. Aby bylo možné nastavit řádek i sloupec, je potřeba tyto volby přepočítat a vyslat je řadiči displeje. První řádek je adresován od 0, takže je možné žádnou pozici znaku `x` vyslat řadiči pomocí příkazů `if(y==0)adr=x` a `send_com(0x80 | adr)`. V případě volby druhého řádku `y = 1` se pozice na displeji odvozuje od adresy prvního znaku na druhém řádku, která je 40. Proto je druhý řádek ošetřen příkazem `if(y==1)adr=0x40+x`. Podobným postupem by se postupovalo, kdyby řádků na displeji bylo více.

Jako rozšíření výše popsané knihovny může být funkce pro výpis pole znaků, které chceme předat displeji. Je možné pak využít formátování standardního výpisu, které uložíme do pole znaků a vytiskneme na displej. K tomuto účelu slouží standardní funkce jazyka `c` `sprintf`, která může být použita následovně `sprintf(pole, "%d text", promenna)`. Je výhodné, aby funkce pro výpis pole `print_pole` řešila i řádkování textu na displeji. Proto samotná funkce vytváří pole znaků, které se nacházejí na displeji (`buffer`) a hodnotu aktuální pozice kurzoru (`kurzor`). Tyto proměnné usnadňují další práci s displejem jako mazání znaků, libovolné řádkování atd. Je-li kurzor na konci displeje `if(kurzor==16)`, provede se odřádkování příkazem `send_com(0xA8)`. Při zaplnění displeje `if(kurzor==32)` se displej vymaže a vypíše se pouze druhá půlka displeje, uložená v poli `buffer`.

Testování výpisu znaků na displej lze mimo jiné provádět výpisem přečtených tlačítek z klávesnice. Výhoda výpisu na displej oproti standardnímu

výpisu do konzole prostředí CodeWarrior je především v jeho rychlosti. Výpis do konzole je časově náročný, kdežto na displej lze vypisovat například hodnoty vnitřních proměnných programu rychleji.

4.7.3 Realizace



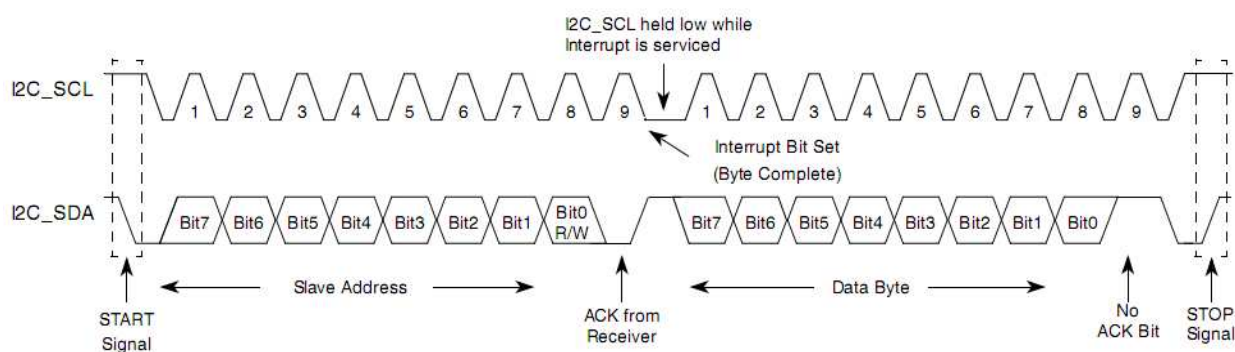
Obr. 25 Vývojový diagram úlohy pro ovládání displeje

4.8 ÚLOHA DEMONSTRUJÍCÍ VYUŽITÍ I2C SBĚRNICE

4.8.1 Úvod

Komunikační sběrnice I2C byla vyvinuta firmou Philips. V současné době je hojně rozšířená a její podporu nalezneme i u nejjednodušších 8bitových mikrokontrolérů. Používá se pro připojení inteligentních senzorů, LCD displejů, pamětí či rozšiřujících periferií (například brány či převodníky). Výhoda této sériové sběrnice je, že potřebuje pouze dva vodiče – pro data (SDA) a hodiny (SCL). Komunikační model je typu master-slave. Master řídí přenos a vytváří hodiny SCL. Jednotky slave poslouchají a zpracovávají řídicí signály vysílané jednotkou master. Master jednotky slave oslovuje podle jedinečné adresy na sběrnici. Tato adresa je 7bitová a skládá se z pevné části adresy (definované výrobcem) a volitelné části, kterou vytvoříme propojením na desce plošných spojů. Poslední bit adresy definuje čtení nebo zápis dat pro jednotku slave. Tato sériová 8bitová sběrnice má tři rychlosti. Standardní rychlost je do 100 kbps, rychlé přenosy jsou do 400 kbps. Pro některé testovací účely existuje standart HS (high speed) od firmy Philips dovolující rychlost 3,4 Mbps. Počet celkových zařízení na sběrnici I2C a délka vodičů je omezena elektrotechnickými vlastnostmi sběrnice a to do maximální celkové zátěžové kapacity 400 pF.

Přenos dat na sběrnici I2C probíhá podle standartu, který je zobrazen na obrázku 26.



Obr. 26 Přenos dat na sběrnici I2C

[1]

Výhoda sběrnice je, že zařízení nepotřebuje adresový dekodér, vše se posílá po lince. Na začátku přenosu master vyšle adresu slave zařízení. Master začíná a ukončuje přenos po sběrnici. Data jsou vysílána po bajtech (první MSB a poslední LSB) a počet bajtů ve zprávě není omezen. Po správném přijetí dat jednotka slave vyšle na konci bajtu potvrzovací 9 bit nazvaný ACK (acknowledge).

Mikrokontrolér obsahuje podporu I2C sběrnice, takže generuje a kontroluje potvrzovací signály uvnitř periferie. K této sběrnici pak přistupujeme přes ovládací registry a výměna dat je realizována registrem. Ten můžeme v přerušení plnit daty pro vysílání, nebo jeho čtením získáme přijatá data z linky.

Zařízení, které můžeme připojit na I2C sběrnici, je velice mnoho. Nakonec jsem vybral obvod PCF8583, který slouží jako systém reálných hodin (RTC) a kalendáře. Disponuje čítačem sekund, minut, hodin, dní měsíců a roků. Obsahuje také dva nezávislé alarmy. Obvod vyžaduje vlastní krystal pro vnitřní hodiny. Čítání je pak nezávislé na běhu procesoru. Mikrokontrolér přes sběrnici I2C získává informace o čase z obvodu PCF8583, nebo jej nastavuje zápisem do jeho registrů.

4.8.2 Zadání

Inicializujte obvod pro systém reálných hodin a kalendáře PCF8583, připojený na I2C sběrnici. Vytvořte ovládací proceduru, která nastaví pomocí klávesnice správný čas a datum v obvodu PCF8583.

Poté vypisujte na displej informace z obvodu PCF8583, a to aktuální čas i datum.

4.8.3 Návod

Obvod reálných hodin a kalendáře PCF8583 je připojen přes sběrnici I2C. Podporu této sběrnice obsahuje mikrokontrolér přímo na čipu, a proto ke sběrnici přistupujeme přes vnitřní registry. Sběrnice I2C je typu master/slave, a je nutné nastavit adresu zařízení, se kterým chceme komunikovat. Obvod má volbu posledního bitu adresy, který je nastaven na nulu. Proto adresa pro zápis do obvodu PCF8583 je A0 a pro četní dat je adresa A1.

Pro začátečníky se sběrnici I2C je vhodné používat nejprve jednoduchou strukturu příkazů spolu s čekacími smyčkami. Teprve po správném otestování sekvence jednotlivých příkazů je vhodné přejít k problému časování této úlohy. Jednotlivé čekací smyčky pak lze nahradit čekáním na stavové bity, které indikují správné provedení požadované instrukce na sběrnici I2C.

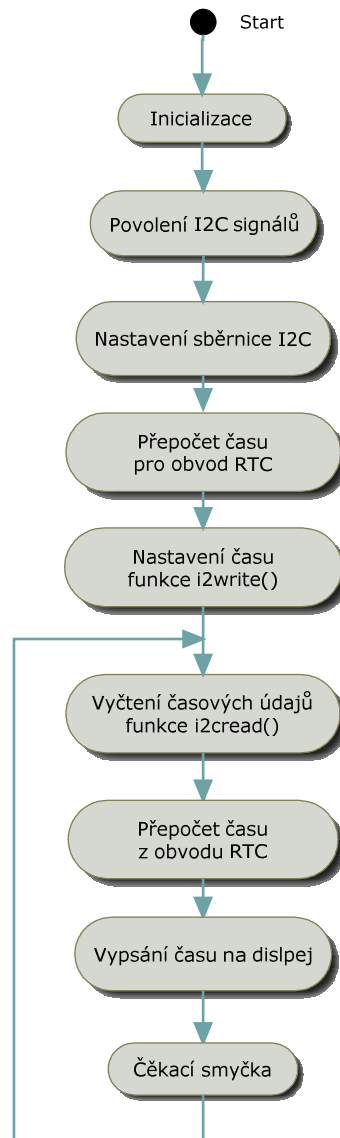
Nejdříve je nutné inicializovat sběrnici I2C. Na začátku je potřeba používané registry vymazat. Dále se podle datasheetu [1] doporučuje provést vyčtení registru `temp=MCF_I2C_I2DR`. Další důležitý krok inicializace je nastavení rychlosti sběrnice. Maximální dovolený hodinový kmitočet obvodu PCF8583 na sběrnici je 100 kHz. Při systémových hodinách 60 MHz procesoru je potřeba nastavit děličku I2C kmitočtu. Protože není vyžadována rychlost, zvolil jsem 27 MHz, tedy nastavení děličky `MCF_I2C_I2FDR = 0x3F`.

Je výhodné vytvořit funkce pro zápis a čtení dat ze sběrnice I2C. Jednodušší je začít funkcí pro zápis dat na sběrnici. Po správné inicializaci je potřeba přepnout sběrnici do vysílacího režimu příkazem `MCF_I2C_I2CR |= MCF_I2C_I2CR_MTX`. Na začátku přenosu vygenerujeme startovací sekvenci `MCF_I2C_I2CR |= MCF_I2C_I2CR_MSTA`. Po tomto příkazu je nutné vyčkat na připravenost sběrnice, jinak dojde k datové kolizi a procesor zůstane v ochranné nekonečné smyčce. Tudíž se správné časování pozná správným průchodem programu, v opačném případě procesor ukončí svoji činnost na některém z řádků kódu. Pro začátek je možné zde ponechat čekací smyčku. Ve správném ošetření času zde čekáme na příznakový bit `MCF_I2C_I2SR_IIF`, který se nastaví do 1 po správném dokončení přenosu bytu. Může dále následovat kontrola příchodu ACK, tedy potvrzení druhé strany, že data byla přijata. Příchod ACK je indikován natavením bitu `MCF_I2C_I2SR_RXAK`. Nyní vystavíme data na sběrnici. Jako první vyšleme adresu zařízení, do kterého chceme zapisovat, tedy `MCF_I2C_I2DR = 0xA0`. Opět následuje již popsané dodržení časového odstavu následující instrukce od vyslání dat. Nyní následuje vyslání adresy registru obvodu PCF8583, do kterého chceme zapisovat. Po časovém odstavu následují samotná data, která chceme zapsat na zvolenou adresu. Zde můžeme data vysílat po jednotlivých bajtech, přičemž adresa registru, kam jsou data zapsána, se automaticky inkrementuje. Tím můžeme zapsat blok dat do registrů za sebou od počáteční zvolené adresy. Na konci přenosu je důležité vyslat ukončovací podmínku příkazem `MCF_I2C_I2CR &= ~MCF_I2C_I2CR_MSTA`. Aby byl dodržen správný časový odstup pro následující práci se sběrnici, je vhodné, aby se na konci funkce provedl test připravenosti sběrnice. Jakmile je sběrnice připravena k dalšímu používání, vynuluje se tzv. „busy flag“ v registru `MCF_I2C_I2SR_IBB`. Tímto krokem je provedeno korektní obslužení sběrnice I2C pro zápis dat do zvoleného zařízení.

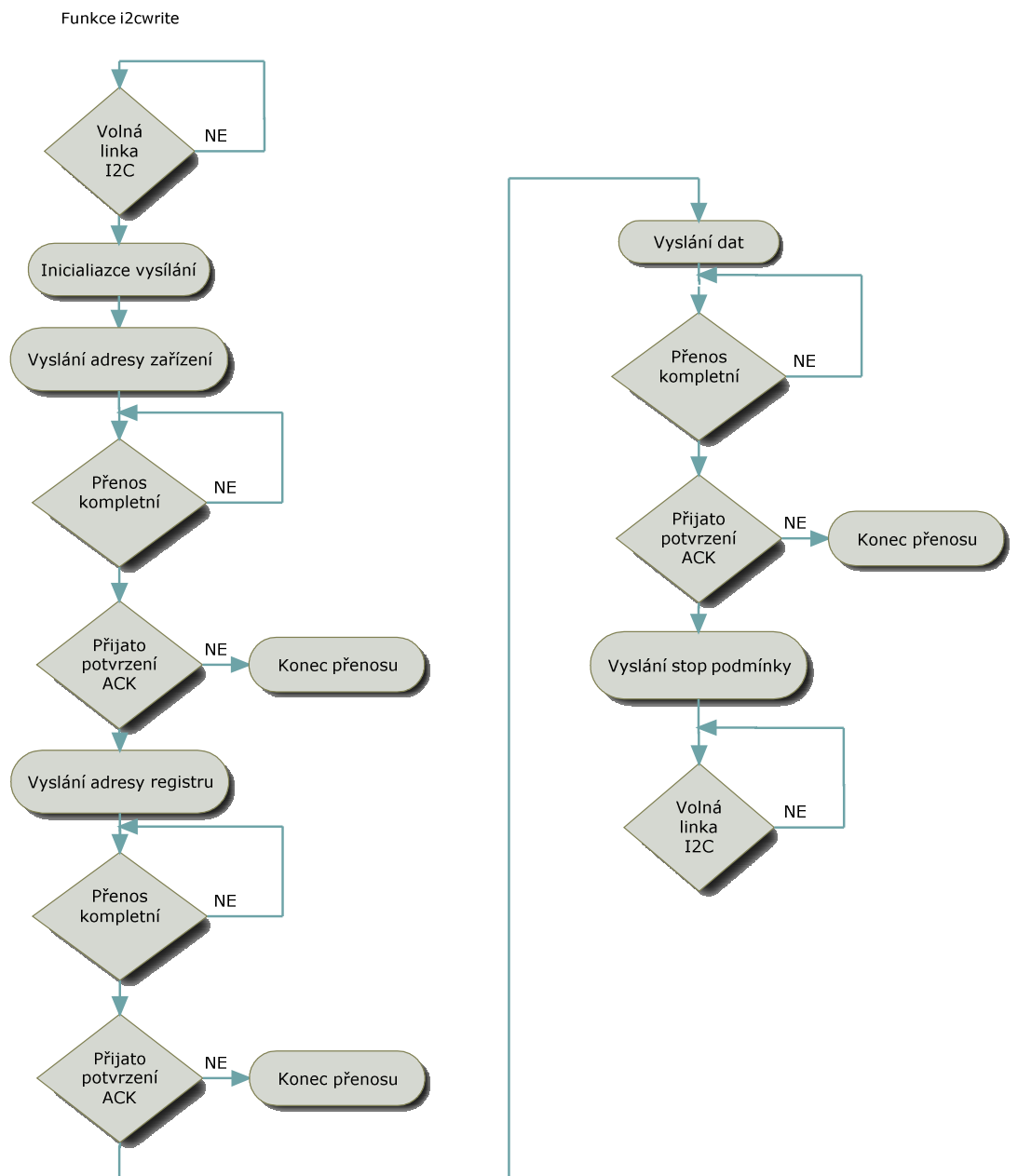
Rutina pro čtení dat je složitější, než rutina pro zápis dat, ale vychází ze stejných principů. Začátek je stejný jako pro zápis. Po nastavení vysílacího módu je na sběrnici vyslána adresa pro zápis, tedy A0. Následuje adresa registru, ze kterého chceme číst data v obvodu PCF8583. Nyní následuje opakovaný start na sběrnici I2C realizovaný příkazem `MCF_I2C_I2CR |= MCF_I2C_I2CR_RSTA`. V tuto chvíli požádáme zařízení o posláání dat. Nejprve se odešle adresa zařízení pro čtení A1. Mikrokontrolér se pomocí příkazu `MCF_I2C_I2CR &= ~MCF_I2C_I2CR_MTX` přepne do přijímacího stavu. Poté přicházejí požadovaná data od obvodu PCF8583, která se nacházejí v registru `MCF_I2C_I2DR`. Po správném přijetí datového paketu se ukončí přenos ukončovací sekvencí příkazem `MCF_I2C_I2CR &= ~MCF_I2C_I2CR_MSTA`. Pro řádné ukončení přenosu a uvolnění sběrnice je proveden opět test připravenosti „busy flag“.

Tímto úloha obsahuje důležité knihovny pro příjem a zápis dat do obvodu reálných hodin a kalendáře. Tento obvod se po připojení napájení uvede do původního stavu, kdy jsou hodiny nastaveny na 0:00:00. Jakmile je obvod připojen na napájecí napětí 3,3 V, automaticky začíná inkrementovat čas. Tento čas je uložen na adresách 0x01 pro setiny sekundy, 0x02 pro sekundy, na adrese 0x03 minuty a na 0x04 hodiny. Číslo je v registrech rozděleno na jednotky a desítky. Proto je potřeba dané číslo vyčíst příkazem `sec=(10*(sec&0xF0)>>4)+(sec&0x0F)`, kde `sec` je vyčtené nezpracované číslo z registru obvodu PCF8583. Zpracovaná přijatá data můžeme vytisknout na displej. Pokud hodiny nenastavíme, zobrazovaný čas udává dobu připojeného napětí na obvod PCF8583.

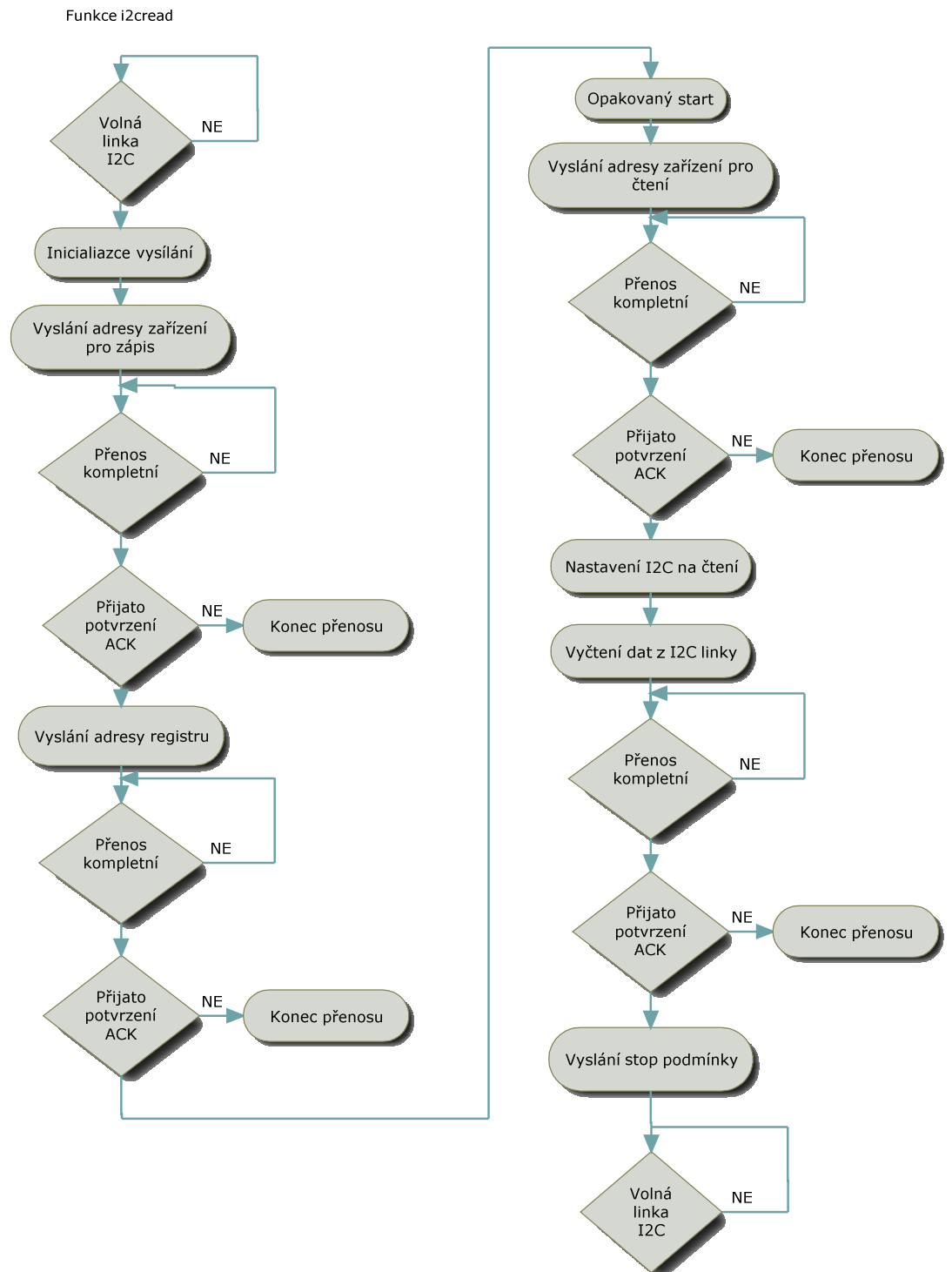
4.8.1 Realizace



Obr. 27 Vývojový diagram hlavní funkce úlohy I2C sběrnice



Obr. 28 Vývojový diagram funkce zápisu dat na I2C sběrnici



Obr. 29 Vývojový diagram funkce čtení dat z I2C sběrnice

5. ZÁVĚR

Tato práce obsahuje podklady pro sestavení laboratorních úloh s mikrokontrolérem ColdFire MCF52235. Tyto úlohy jsou koncipovány tak, aby si studenti vyzkoušeli práci se základními periferiemi mikrokontrolérů.

Aby bylo možné úlohy realizovat, vytvořil jsem přípravek, který obsahuje periferie nutné pro běh úloh. Tento přípravek využívá základní vývojový kit M52233DEMO, který obsahuje mikrokontrolér MCF52233 s programátorem, dalšími periferiemi a konektorem pro připojení externích periférií. Právě tímto konektorem je propojen přípravek pro laboratorní úlohy s mikrokontrolérem. Obsahuje indikační LED diody, tlačítka, dvoubarevnou LED diodu, potenciometr, porty pro připojení klávesnice a displeje a obvod PCF8583, který komunikuje na sběrnici I2C a vytváří obvod RTC a kalendáře. Práce se zabývá návrhem této desky externích periférií, včetně návrhu desky plošných spojů. Protože jednotlivé periferie sdílí vývody s dalšími systémy, je nutné mezi jednotlivými periferiemi přepínat a povolovat je pomocí přepínačů. Celkový seznam připojených a sdílených periférií je zobrazen v tabulce 2.

Hlavní náplní práce jsou laboratorní úlohy. Každá laboratorní úloha má svůj úvod, zadání úlohy, návod pro vypracování a vývojový diagram realizovaných úloh. V návodu jsou popsány důležité kroky, které směřují ke správnému zvládnutí úlohy. Tyto úlohy jsou vypracované ve vývojovém prostředí CodeWarrior a jsou uloženy na příloženém CD. Na něm se nachází i podklady pro návrh desky externích periférií.

První úloha se zabývá prací se vstupy a výstupy, kde studenti vytvoří světelného hada. Oživení vstupů a výstupů je u neznámého mikrokontroléru základní úloha programátora. Další úloha je zaměřena na další základní stavební prvek mikrokontrolérů, a to jsou časovače. Zde si studenti vyzkouší vytvoření přesně definované časové smyčky a obsluhu přerušování jejího příznaku. S časovači souvisí čítače, které jsou obsaženy v následujícím zadání. Studenti budou v této úloze obsluhovat vnitřní systém reálných hodin (RTC) a vytvoří na něm stopky. Další zadání se snaží studenty seznámit s principem A/D převodníku. V této úloze zjišťují stav potenciometru a vyhodnocují údaje z 3osého akcelerometru. Při zpracování dat

je v úloze využít šikovný nástroj FreeMaster, který dovoluje sledovat proměnné v běžícím programu na mikrokontroléru. Studenti si dále vyzkouší ovládat pulsně šířkovou modulaci PWM, kde budou řídit pomocí PWM barvu dvoubarevné ledky. Dále může následovat úloha, ve které studenti vytvoří funkci pro vyčítání znaků z maticové klávesnice. V praxi většina zařízení vyžaduje zobrazovat informace na displej. Komunikace s displejem je náplní předposlední úlohy. Studenti by měli vytvořit knihovnu pro ovládání displeje se čtyřmi řádky. Displej disponuje základním řadičem HD44780, který obsahuje většina základních alfanumerických displejů. Jako poslední úloha je vybrán problém ohledně obsluhy sběrnice I2C. Pro tuto úlohu byl vybrán obvod PCF8583, který obsahuje systém reálných hodin (RTC) a kalendář. Studenti budou tento obvod nastavovat přes I2C sběrnici a následně z něj vyčítat data.

Koncepce úloh vychází z mých vlastních zkušeností z předmětů o mikrokontrolérech na střední a vysoké škole. Ovšem jejich konečné sestavení a vytvoření časového harmonogramu jednotlivých úloh záleží na zkušenostech vyučujícího. Úlohy pak musí být sestaveny tak, aby tvořili jeden celek, nebyly příliš náročné ani příliš lehké, a aby probíraná látka korespondovala s přednáškami předmětu.

Na tyto navržené laboratorní úlohy může navazovat předmět s dalšími úlohami, které budou řešit hlubší problémy, které je schopen zvládat 32bitový mikrokontrolér rodiny ColdFire. Například vytvoření různých webových serverů, komunikace po síťovém protokolu Telnet, či šifrování přenosů pomocí SSH. Zpracované laboratorní úlohy pak mohou vytvářet základ pro tyto složitější úlohy. Například zpracování dat AD převodníkem, které je následně odesíláno protokolem Telnet a na vzdáleném počítači je toto měření vyhodnocováno.

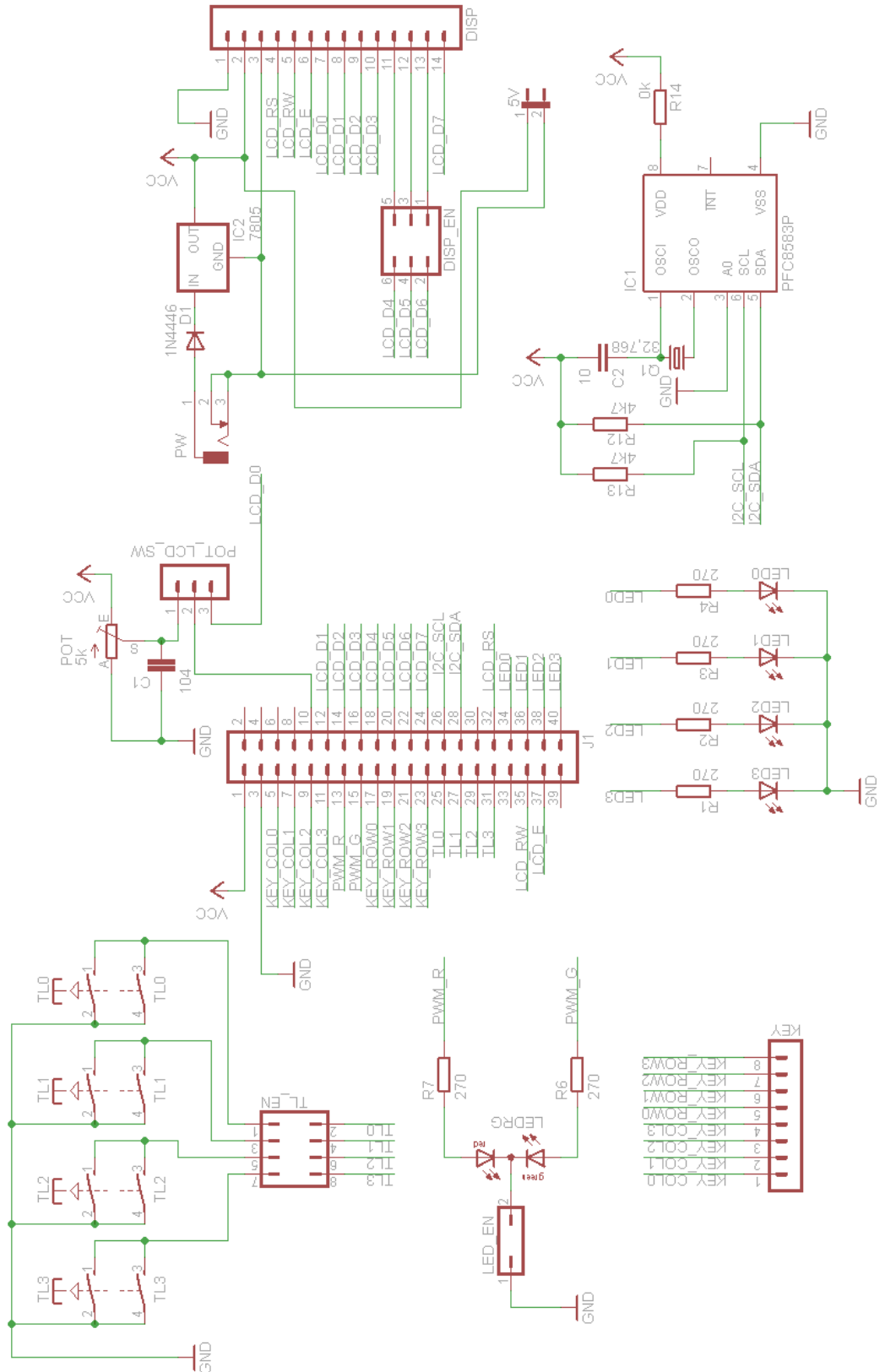
6. SEZNAM LITERATURY

- [1] *MCF52233 ColdFire® Integrated Microcontroller Reference Manual*, Freescale Semiconductor, Rev 5 09/2007
(www.freescale.com/files/32bit/doc/ref_manual/MCF52233RM.pdf)
- [2] *HD44780U (LCD-II)(Dot Matrix Liquid Crystal Display Controller/Driver)*, Hitachi
(<http://www.datasheetcatalog.org/datasheet/HitachiSemiconductor/mXvyusu.pdf>)
- [3] *LCD Module DEM16217SYH*, Display Elektronik GmbH, 2.4.2001
(<http://pdf1.alldatasheet.net/datasheet-pdf/view/118384/ETC/DEM16217SYH.html>)
- [4] *PCF8583 Clock/calendar with 240 × 8-bit RAM*, Philips Semiconductors, 28.3.1997
(http://www.datasheetcatalog.org/datasheet/philips/PCF8583_5.pdf)
- [5] *Schematic of M52233DEMO*, Freescale Semiconductor, Rev E1 04/2007
(www.freescale.com/files/32bit/hardware_tools/schematics/M52233DEMO_SCH.pdf)
- [6] *Human Fall Detection Using 3-Axis Accelerometer*, Freescale Semiconductor, Rev 2.0 12/2005
(www.freescale.com/files/sensors/doc/ref_manual/MMA7260QHFD.RM.pdf)

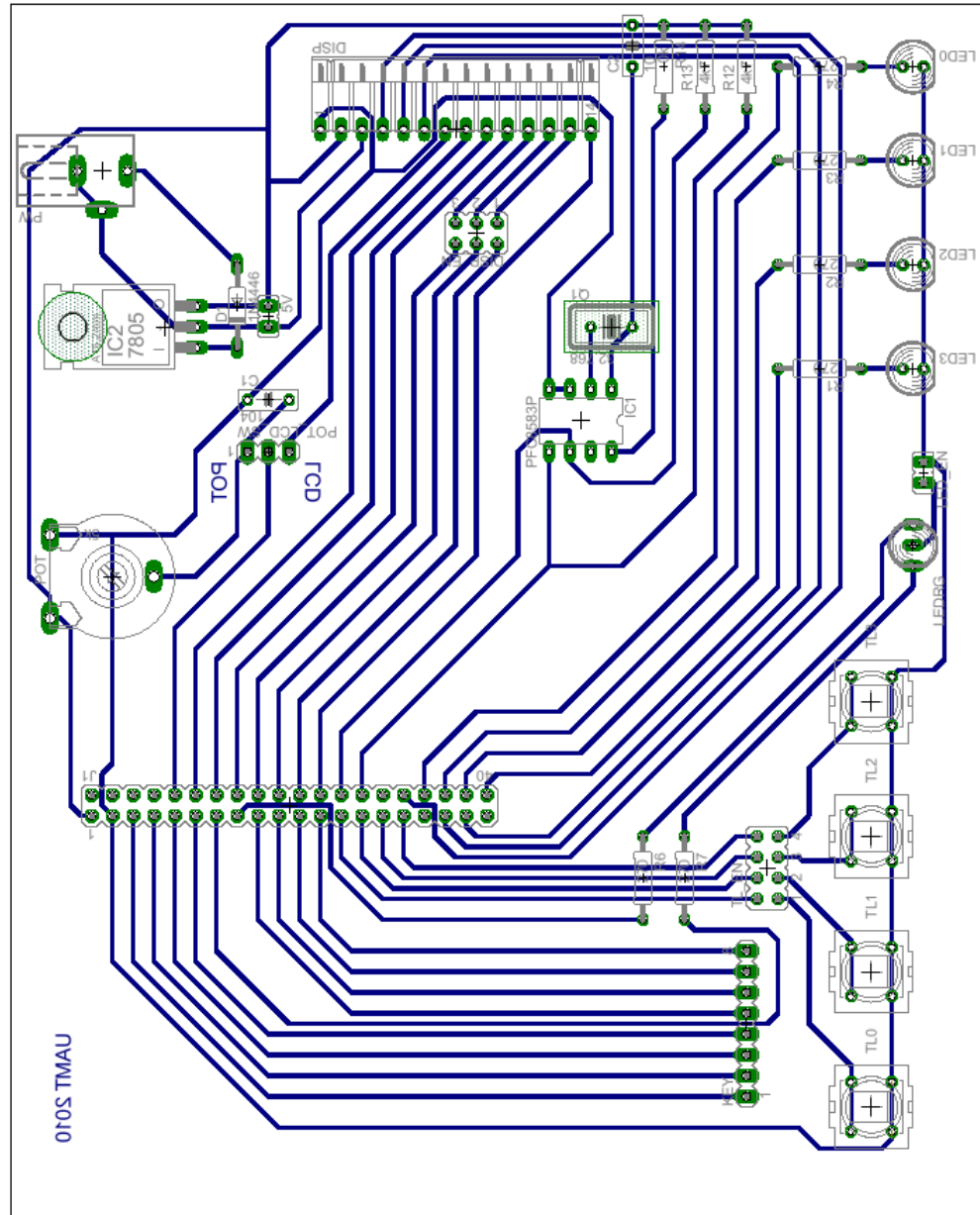
7. SEZNAM PŘÍLOH

A – Schéma zapojení desky externích periferií.....	65
B – Výkres desky plošných spojů	66
C – Obsah přiloženého CD.....	67

Příloha A – Schéma zapojení desky externích periferií



Příloha B – Výkres desky plošných spojů



Příloha C – Obsah přiloženého CD

CD:\DIP prace\

DYCKADIP.pdf - text diplomové práce

CD:\Vypracované úlohy\

- složka obsahuje projekty
vypracovaných laboratorních úloh
ve vývojovém prostředí CodeWarrior

CD:\Literatura\

MCF52233RM.pdf - literatura [1]

HD44780U.pdf - literatura [2]

DEM16217SYH.pdf - literatura [3]

PCF8583_5.pdf - literatura [4]

M52233DEMO_SCH.pdf - literatura [5]

MMA7260QHFD RM.pdf - literatura [6]

CD:\Prilohy\

schema.pdf - schéma zapojení slave zařízení

deska.pdf - návrh desky plošných spojů

\EAGLE - složka obsahuje projekt návrhu desky