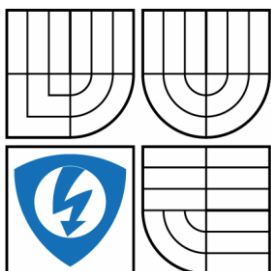


VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

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

KNIHOVNA PRO ŘÍZENÍ LCD DISPLEJE GD-5360P

LIBRARY FOR CONTROL OF LCD DISPLAY GD-5360P

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

MIROSLAV ŠTIBRANÝ

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. TOMÁŠ MACHO, Ph.D.

BRNO 2014



VYSOKÉ UČENÍ
TECHNICKÉ V BRNĚ

Fakulta elektrotechniky
a komunikačních technologií

Ústav automatizace a měřicí techniky

Bakalářská práce

bakalářský studijní obor
Automatizační a měřicí technika

Student: Miroslav Štibraný

ID: 146973

Ročník: 3

Akademický rok: 2013/2014

NÁZEV TÉMATU:

Knihovna pro řízení LCD displeje GD-5360P

POKYNY PRO VYPRACOVÁNÍ:

1. Seznamte se s LCD displejem GD-5360P, jeho připojením k mikrokontroléru Freescale MC9S08LH64 a vývojovou deskou TWR-S08LH64.
2. Přehledně zdokumentujte způsoby ovládání displeje.
3. Navrhněte a implementujte knihovnu v jazyce C, která bude umožňovat ovládání displeje. Zdokumentujte rozhraní k jednotlivým knihovním funkcím.
4. Navrhněte a implementujte alespoň dvě laboratorní úlohy demonstrující použití knihovny.

DOPORUČENÁ LITERATURA:

- [1] MC9S08LH64 Reference Manual. Rev. 5.1. Freescale Semiconductor. 2012.
[2] TWR-S08 User gude. Axiom manufacturing. 2010.

Termín zadání: 10.2.2014

Termín odevzdání: 26.5.2014

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

Konzultanti bakalářské práce:

doc. Ing. Václav Jirsík, CSc.

Předseda oborové rady

UPOZORNĚNÍ:

Autor bakalářské práce nesmí při vytváření bakalářské 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

Táto bakalárska práca sa zaoberá návrhom knižnice pre riadenie LCD displeja GD-5360P na vývojovej doske TWR-S08LH64 s mikrokontrolérom MC9S08LH64. V úvodnej časti sú uvedené základné informácie o princípe LCD displejov a o danom hardvéri (vývojová doska, mikrokontrolér a LCD displej). Podstatnou časťou práce je rozbor funkcií radiča LCD displeja integrovaného v mikrokontroléri a taktiež vysvetlenie prístupu k ovládaniu stavu jednotlivých segmentov LCD displeja GD-5360P. Hlavným výstupom bakalárskej práce je navrhnutá a následne implementovaná knižnica na ovládanie LCD displeja. Pre účely vyučovania v kurze BMIC sú vytvorené dve laboratórne úlohy, ktoré používajú LCD displej.

Kľúčové slová

LCD displej GD-5360P, vývojová doska TWR-S08LH64, mikrokontrolér MC9S08LH64, knižnica pre ovládanie LCD displeja, TOD, modul Time of Day, KBI, modul Keyboard interrupt,

Abstract

This bachelor thesis deals with design of library for control of LCD display GD-5360P on development kit TWR-S08LH64 with microcontroller MC9S08LH64. The introductory section provides basic information about principle of LCD displays and about determined hardware (development kit, microcontroller and LCD display). The significant part of thesis is analysis of all functions of LCD display driver integrated in the microcontroller and explanation of access to the status of individual segments. The main output of the bachelor thesis is to design and subsequently implement the library for control of LCD display GD-5360P. Two laboratory assignments that use LCD display are created for the purpose of teaching in course BMIC.

Keywords

LCD display GD-5360P, development kit TWR-S08LH64, microcontroller MC9S08LH64, library for control of LCD display, TOD, Time of Day module, KBI, Keyboard interrupt module

Bibliografická citácia:

ŠTIBRANÝ, M. *Knihovna pro řízení LCD displeje GD-5360P*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2014. 55 s. Vedoucí bakalářské práce Ing. Tomáš Macho, Ph.D..

Prohlášení

„Prohlašuji, že svou bakalářskou práci na téma Knihovna pro řízení LCD displeje GD-5360P jsem vypracoval samostatně pod vedením vedoucího bakalářské 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 bakalářské práce dále prohlašuji, že v souvislosti s vytvořením této bakalářské 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 části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

V Brně dne: 19. května 2014

.....
podpis autora

Pod'akovanie

Ďakujem vedúcemu bakalárskej práce Ing. Tomášovi Machovi, Ph.D. za jeho cenné rady a výborné vedenie pri vypracovávaní mojej bakalárskej práce.

V Brně dne: 19. května 2014

.....
podpis autora

Obsah

1	Úvod.....	11
2	Rozbor	12
2.1	Vývojová doska TWR-S08LH64.....	12
2.2	Mikrokontrolér MC9S08LH64	12
2.3	Základný princíp činnosti LCD displejov	13
2.4	Statický režim a režim multiplex	14
2.5	LCD displej GD-5360P.....	15
2.6	Prepojenie LCD displeja s mikrokontrolérom	16
3	Mikrokontrolér a radič LCD displeja	19
3.1	Radič pre LCD displeje.....	19
3.2	Hodinový signál pre LCD radič	19
3.2.1	Inicializácia hodinového signálu pre LCD radič.....	20
3.3	Zdroje napätia pre budenie segmentov.....	21
3.4	Registre LCD radiča.....	24
3.4.1	Inicializácia LCD radiča	24
3.4.2	Ovládanie stavu segmentov LCD displeja	30
4	Knižnica na ovládanie displeja GD-5360P.....	32
4.1	Návrh funkcií knižnice.....	32
4.2	Implementované funkcie.....	33
4.2.1	Funkcia <i>Init_clock</i>	33
4.2.2	Funkcia <i>Init_LCD</i>	33
4.2.3	Ovládanie špeciálnych segmentov	34
4.2.4	Funkcia <i>All_seg_on</i>	36
4.2.5	Funkcia <i>All_seg_off</i>	36
4.2.6	Funkcia <i>Write_char</i>	36
4.2.7	Funkcia <i>Write_spec_char</i>	36
4.2.8	Funkcia <i>Write_string</i>	37
4.2.9	Funkcia <i>Num_small</i>	37
4.2.10	Funkcia <i>Num_small_hex</i>	37
4.2.11	Funkcia <i>Num_big</i>	37
4.2.12	Funkcia <i>Num_big_hex</i>	38
4.2.13	Funkcia <i>Num_aux</i>	38
4.2.14	Funkcia <i>Num_aux_hex</i>	38

4.2.15	Funkcia <i>Clr_week_day</i>	38
4.2.16	Funkcia <i>Set_week_day</i>	38
4.2.17	Funkcia <i>Battery</i>	39
4.2.18	Funkcia <i>Time_HM</i>	39
4.2.19	Funkcia <i>Time_HMS</i>	39
4.2.20	Funkcia <i>Stopwatch</i>	39
5	Laboratórne úlohy.....	40
5.1	Laboratórna úloha: Zobrazenie hodín na LCD displeji.....	41
5.1.1	Zadanie úlohy.....	41
5.1.2	Teoretický úvod.....	41
5.1.3	Doporučený postup pri programovaní zadania.....	46
5.1.4	Vzor štruktúry programu.....	46
5.1.5	Možné chyby.....	47
5.1.6	Záver úlohy.....	47
5.2	Laboratórna úloha: Stopky.....	47
5.2.1	Zadanie úlohy.....	47
5.2.2	Teoretický úvod.....	48
5.2.3	Doporučený postup pri programovaní zadania.....	50
5.2.4	Vzor štruktúry programu.....	51
5.2.5	Záver úlohy.....	51
6	Záver.....	52

Zoznam obrázkov

Obrázok 1: Bloková schéma mikrokontroléra MC9S08LH64 [3].....	13
Obrázok 2: LCD displej GD-5360P – rozmiestnenie segmentov [5]	15
Obrázok 3: Detail sedemsegmentového znaku [5].....	16
Obrázok 4: Detail indikátoru stavu nabitia batérie [5]	16
Obrázok 5: Prepojenie mikrokontroléra MC9S08LH64 s LCD displejom GD-5360P [6].....	17
Obrázok 6: Zdroje hodinových signálov mikrokontroléra MC9S08LH64 [3]	20
Obrázok 7: Register ICSC2 – ICS Control Register 2 [3].....	21
Obrázok 8: Moduly ICS (Internal Clock Source) a XOSCVLP (External Reference Clock) [3].....	21
Obrázok 9: Bloková schéma zdrojov napájania pre radič LCD displeja [3]	23
Obrázok 10: Register LCDC0 (LCD Control Register 0) [3].....	25
Obrázok 11: Register Lcdrvc – LCD Regulated Voltage Control Register [3]	26
Obrázok 12: Register LCDSupply – LCD Voltage Supply Register [3]	26
Obrázok 13: LCDPEN0 až LCDPEN5 – LCD Pin Enable Registers 0–5 [3].....	27
Obrázok 14: Register LCDBPEN0 až LCDBPEN5 – Backplane Enable Registers 0–5 [3].....	28
Obrázok 15: LCDWF40 a LCDWF41 – LCD Waveform Registers [3]	29
Obrázok 16: Register LCDBCTL – LCD Blink Control Register [3]	29
Obrázok 17: Register LCDWF0 – LCD Waveform Register 0 [3]	30
Obrázok 18: Časová postupnosť riadenia fáz	31
Obrázok 19: Prepojenie LED diód s mikrokontrolérom [6].....	41
Obrázok 20: Register PTCDD – Port C Data Direction [3]	41
Obrázok 21: Register PTCDD – Port C Data Register [3]	42
Obrázok 22: Bloková schéma modulu TOD [3].....	42
Obrázok 23: Register TODC – TOD Control Register [3].....	43
Obrázok 24: Register TODSC – TOD Status and Control Register [3]	43
Obrázok 25: Pripojenie tlačidiel k pinom portu A [6].....	44
Obrázok 26: Register KBISC – KBI Interrupt Status and Control Register [3]	44
Obrázok 27: Register KBIES – KBI Interrupt Edge Select Register [3].....	45
Obrázok 28: Register PTAPE – Port A Pull Enable Register [3].....	45
Obrázok 29: Register KBIPE – KBI Interrupt Pin Select Register [3].....	45
Obrázok 30: Register PTAD – Port A Data Register [3].....	45
Obrázok 31: 16-bitový čítač (zjednodušená bloková schéma) [3].....	49
Obrázok 32: Register TPM1SC – TPM Status and Control Register [3]	50
Obrázok 33: Register TPM1MODH – TPM Counter Modulo Register High [3]	50
Obrázok 34: Register TPM1MODL – TPM Counter Modulo Register Low [3]	50

Zoznam tabuliek

Tabuľka 1: Vývody LCD displeja GD-5360P [3][5][6]	18
Tabuľka 2: Konfigurácia prepínačov napájania [3]	23
Tabuľka 3: Hodnoty rámcovej frekvencie [3].....	25
Tabuľka 4: Možnosti ovládania stavu segmentov.....	31
Tabuľka 5: Makrá pre ovládanie špeciálnych segmentov	34
Tabuľka 6: Možnosti nastavenia deličky hodinového signálu pre čítač [3].....	49

1 ÚVOD

Práca s mikrokontrolérmi neodlučiteľne patrí k potrebným schopnostiam budúceho absolventa Fakulty elektrotechniky a komunikačných technológií, nielen so zameraním na automatizáciu. Pre praktické vyučovanie mikroprocesorovej techniky sa v bakalárskom kurze BMIC (Mikroprocesory) využíva vývojová doska TWR-S08LH4, obsahujúca LCD displej GD-5360P. Hlavným cieľom tejto práce je vytvoriť na ovládanie spomínaného LCD displeja knižnicu, ktorej funkcie budú zamerané na potreby vyučovania v kurze BMIC.

Začiatok práce je venovaný základnému princípu LCD displejov, popisu vývojovej dosky TWR-S08LH64 s mikrokontrolérom MC9S08LH64, základným pojmom týkajúcich sa riadenia LCD displeja GD-5360P. Nasleduje popis možností radiča LCD displeja integrovaného v mikrokontroléri a jeho inicializácia spolu s prehľadom spôsobu ovládania stavu jednotlivých segmentov tohto LCD displeja. Samostatná kapitola je venovaná návrhu potrebných funkcií knižnice spolu s popisom navrhnutých a následne v jazyku C implementovaných funkcií. Posledná kapitola sa zaoberá laboratórnymi úlohami určenými pre kurz BMIC, ktoré využívajú vytvorenú knižnicu pre LCD displej.

2 ROZBOR

Na úvod je potrebné zoznámiť sa s pojmami týkajúcimi sa LCD displejov a uviesť prehľad možností zadaného hardvéru. Jedná sa o LCD displej GD-5360P a mikrokontrolér MC9S08LH64, oboje sa nachádzajú na vývojovej doske TWR-S08LH64.

2.1 Vývojová doska TWR-S08LH64

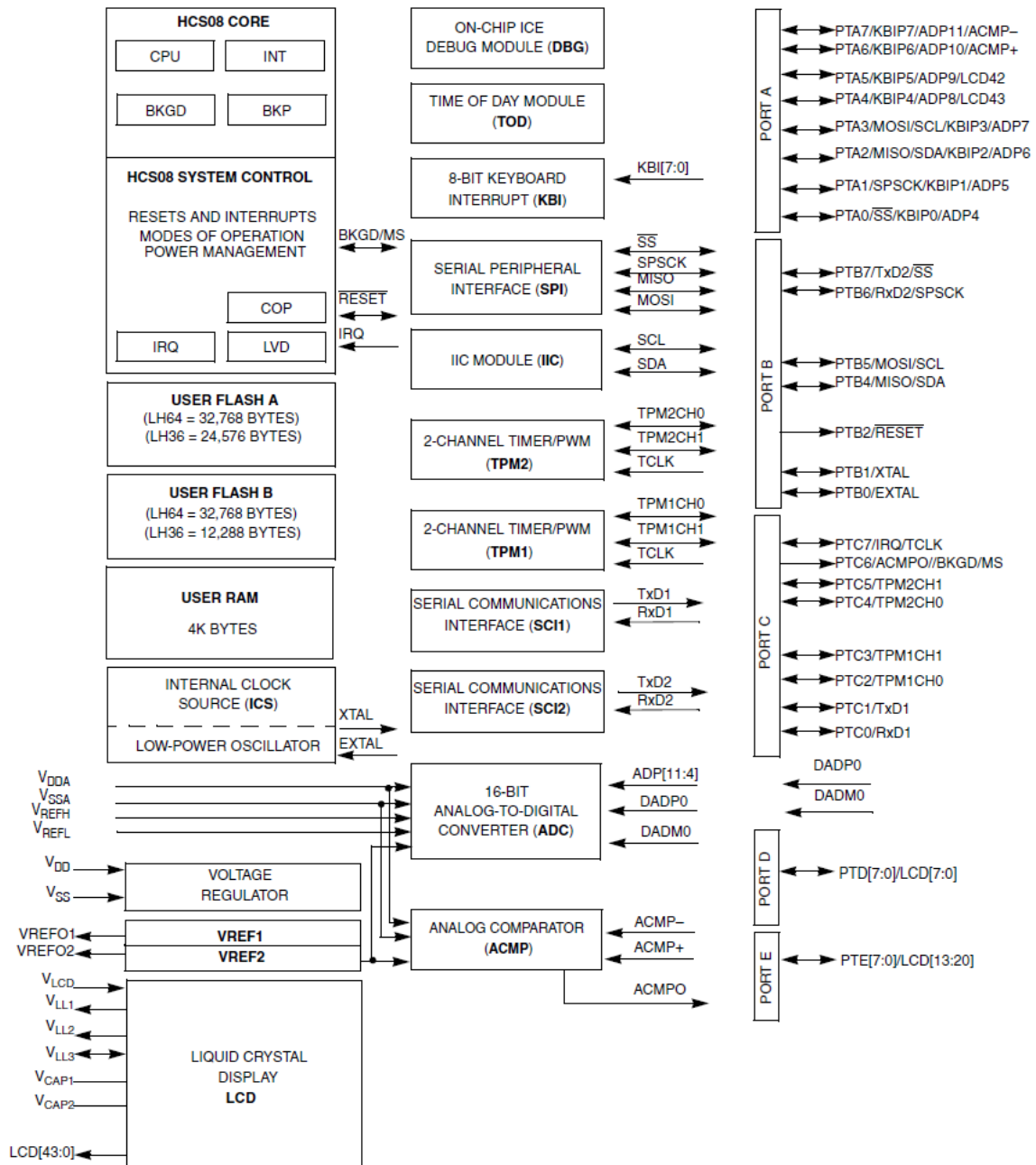
Táto doska obsahuje základné periférie vhodné pre účely výučby. Celá doska je napájaná pomocou mini-B USB konektora, cez ktorý sa zároveň mikrokontrolér MC9S08LH64 programuje. Na tejto doske je osadená 80 vývodová verzia mikrokontroléra. Na vývoj softvéru existuje vývojové prostredie priamo od výrobcu – CodeWarrior for Microcontrollers. Okrem LCD displeja GD-5360P, ktorého ovládaniu sa bude venovať väčšina tejto práce, obsahuje doska TWR-S08LH64 aj tlačidlo RESET, 4 nezávislé tlačidlá, 4 indikačné LED diódy, odporový potenciometer, akcelerometer, konektor sériovej komunikácie RS-232, piezo-bzučiak a svetelný senzor. [4]

2.2 Mikrokontrolér MC9S08LH64

Patrí do skupiny 8-bitových mikrokontrolérov typu HCS08 od firmy Freescale Semiconductor. Disponuje jadrom HCS08, pamäťou FLASH s kapacitou 64 KiB, pamäťou RAM s kapacitou 4 kB, piatimi vstupno-výstupnými portami (PORT A až PORT E) a viacerými modulmi:

- Vnútorňý zdroj hodinového signálu *ICS* (Internal Clock Source)
- Modul externého hodinového signálu *XOSCVP* (External oscillator module)
- Radič LCD displeja (Liquid Crystal Display Driver)
- Špeciálny čítač *TOD* určený pre počítanie času (Time of Day Module)
- Modul prerušenia od tlačidiel *KBI* (8-bit Keyboard Interrupt)
- Modul ladenia *DBG* (Debug Module)
- Modul komunikácie *SPI* (Serial Peripheral Interface)
- Modul komunikácie *IIC* (IIC Module)
- Čítač/PWM modulátor *TPM1*, *TPM2* (2-channel Timer/PWM)
- Modul sériovej komunikácie *SCI1*, *SCI2* (Serial Communications Interface)
- 16-bitový A/D prevodník *ADC* (16-bit Analog to Digital Converter)
- Analógový komparátor *ACMP* (Analog Comparator)
- Zdroj referenčného napätia *VREF1* (Voltage Reference Module)
- Supervízor (watchdog) *COP* (Computer Operating Properly)

Mikrokontrolér pracuje s 3,3V logikou. Bloková schéma mikrokontroléra je na obrázku 1. [3]



Obrázok 1: Bloková schéma mikrokontroléra MC9S08LH64 [3]

2.3 Základný princíp činnosti LCD displejov

Skratka LCD (angl. liquid crystal display) predstavuje displej s tekutými kryštálmi. Tekuté kryštály dokážu zmenou elektrického poľa a ďalších faktorov zmeniť svoju štruktúru. Zmena štruktúry má za následok zmenu optických vlastností – zmenu priepustnosti žiarenia. Práve táto vlastnosť tekutých kryštálov sa využíva v LCD displejoch. V kľudovom stave (bez pôsobenia elektrického poľa, resp. iných faktorov) môžu tekuté kryštály dopadajúce svetlo prepúšťať alebo pohlcovať. Hovoríme o tzv. pozitívnom alebo negatívnom zobrazení. Pri vystavení tekutých kryštálov elektrickému

poľu sa zmení ich štruktúra a s ňou sa zmení zároveň aj uhol polarizácie, pre ktorý kryštály prepúšťajú svetlo – pri pozitívnom zobrazení pôvodne svetlá oblasť stmavne, pri negatívnom nastane prípad opačný. [1]

Na budenie LCD displejov sa používa striedavé napätie. Dlhodobé budenie jednosmerným napätím totiž spôsobuje elektrochemický rozklad kvapalných kryštálov, teda ich zničenie. Tekuté kryštály nie sú elektricky vodivé, spolu s polarizačnými elektródami sa správajú ako kondenzátory a pre budič predstavujú kapacitnú záťaž. Frekvencia budenia sa volí približne 30 Hz, používanie vyšších frekvencií budiaceho napätia nielenže viac zaťažuje budič (práve kvôli kapacitnej záťaži), ale aj skracuje životnosť displeja. [1][2]

2.4 Statický režim a režim multiplex

LCD displej spravidla obsahuje väčší počet segmentov, ktoré sú vhodne elektricky prepojené. Ovládanie stavu (budenie) jednotlivých segmentov (prepúšťanie alebo pohltie dopadajúceho svetla) sa zabezpečuje priloženým elektrickým napätím. Ak sa o budenie každého segmentu stará práve jeden zdroj, jedná sa o statický režim riadenia. Pri malom počte LCD segmentov riadených v statickom režime je potrebný malý počet zdrojov napätia, čo nebýva zásadná komplikácia.

Problém nastáva pri vyššom počte segmentov (napr. grafické LCD displeje). Vtedy je potrebné kvôli dosiahnutiu malého počtu zdrojov a prepojov riadiť jednotlivé segmenty v prepínacom režime, tzv. režim multiplex. V tomto režime sa ovláda vždy len istá skupina segmentov po určitú krátku dobu. Postupne sa vystriedajú všetky skupiny segmentov. Doba vystriedania všetkých skupín definuje opakovaciu frekvenciu, nazýva sa tiež rámcová frekvencia (angl. frame frequency). Prepínanie skupín a ovládanie jednotlivých segmentov musí pracovať na dostatočnej frekvencii, aby neboli viditeľné len jednotlivé skupiny segmentov, ale všetky segmenty zároveň. Využíva sa tu nedokonalosť ľudského zraku (zotrvačnosť), pretože ľudské oko nedokáže rozlíšiť dva po sebe dostatočne rýchlo nasledujúce deje. [1]

Najčastejším elektrickým usporiadaním segmentov je matica. Napríklad 16 segmentov usporiadaných do matice 4x4 tak, že medzi štyrmi vodičmi pre každý riadok a štyrmi vodičmi pre každý stĺpec je zapojený práve jeden segment. Ovládanie týchto segmentov sa rozdelí na 4 rovnaké časové úseky, v ktorých sa naraz ovládajú segmenty na rovnakom riadku, resp. v rovnakom stĺpci.

Vygenerovanie väčšieho počtu potrebných striedavých napätí nie je jednoduchá záležitosť a preto boli vyvinuté špeciálne obvody, ktoré sa nazývajú radiče (angl. driver). Na LCD displejoch s väčším počtom segmentov (napr. grafické displeje) býva radič často krát už súčasťou displeja.

2.5 LCD displej GD-5360P

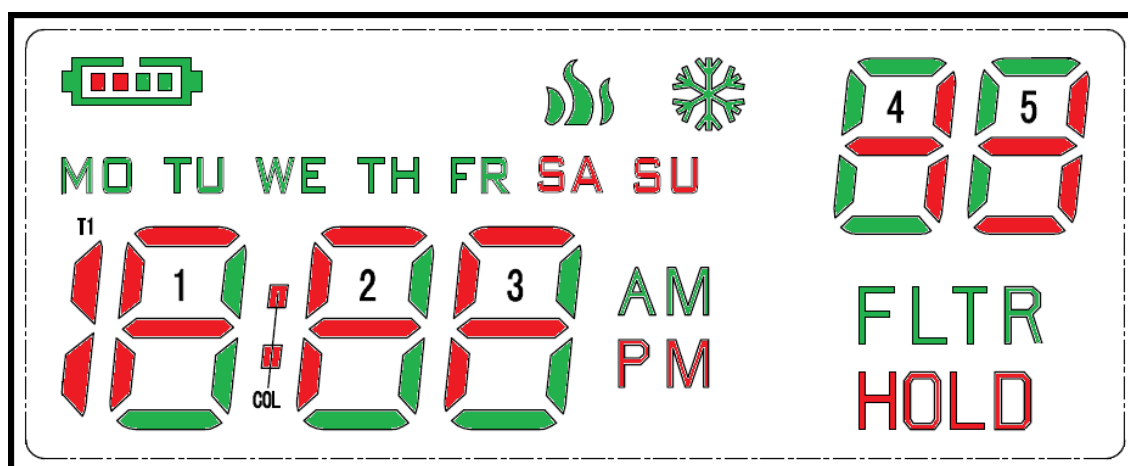
Displej GD-5360P má celkom 30 vývodov, pomocou ktorých sa riadi až 55 segmentov (päť sedemsegmentových znakov, znak číslice 1, dvojbodka, symboly pre indikáciu tepla (plameň) a chladu (snehová vločka), indikátor stavu nabitia batérie, preddefinované textové symboly *AM*, *PM*, *FLTR*, *HOLD* a anglické skratky dní v týždni (*MO*, *TU*, *WE*, *TH*, *FR*, *SA*, *SU*). Určený je hlavne na zobrazovanie časových údajov v 12-hodinovom formáte. [5]

Často sa počet vývodov uvádza v tvare 2x28, ktorý vyjadruje spôsob vnútorného zapojenia jednotlivých segmentov – 2 spoločné vývody *COM* (v špecifikácii displeja uvádzané ako *COM1* a *COM2* – každý segment je pripojený na práve jeden zo spoločných vývodov) a 28 vývodov spoločných vždy pre 2 segmenty. Všetky segmenty tak tvoria maticu 2x28 (2 vývody backplane a 28 vývodov frontplane). [5]

Údaj zo špecifikácie displeja $DUTY = \frac{1}{2}$ je vlastne prevrátená hodnota počtu spoločných vývodov *COM*. Samotná hodnota vyjadruje akú časť pracovného cyklu daného rámcovou frekvenciou je daný segment napájaný. [5]

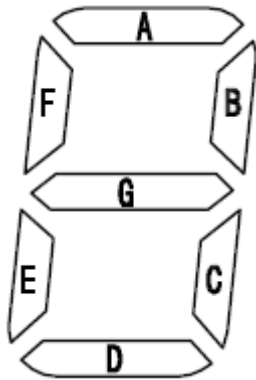
Bez privedeného napätia na segmenty nie sú zobrazené žiadne znaky, jedná sa preto o LCD displej s pozitívne zobrazovanými segmentmi, znakmi. Pre zobrazenie segmentu je potrebné dosiahnuť riadiace napätie 2,7 V. [5]

Displej GD-5360P nemá v sebe integrovaný radič, ktorý by sa staral o jeho riadenie. Na vývojovej doske TWR-S08LH64 túto funkciu plní radič integrovaný v mikrokontroléri MC9S08LH64, s ktorým je displej GD-5360P na tejto doske prepojený (obrázok 5). Takéto riešenie je energeticky úsporné, čo je dôležité hlavne pri zariadeniach napájaných z batérií a menších akumulátorov. [3]

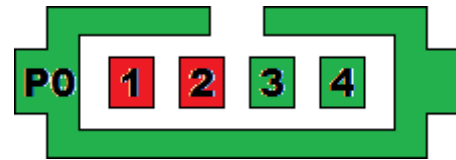


Obrázok 2: LCD displej GD-5360P – rozmiestnenie segmentov [5]

Na obrázku 2 a 4 sú zelenou farbou vyznačené všetky segmenty pripojené k backplane vývodu displeja *COM1*, červenou ku *COM2*. Pozície sedemsegmentových znakov sú očíslované na obrázku 2.



Obrázok 3: Detail sedemsegmentového znaku [5]

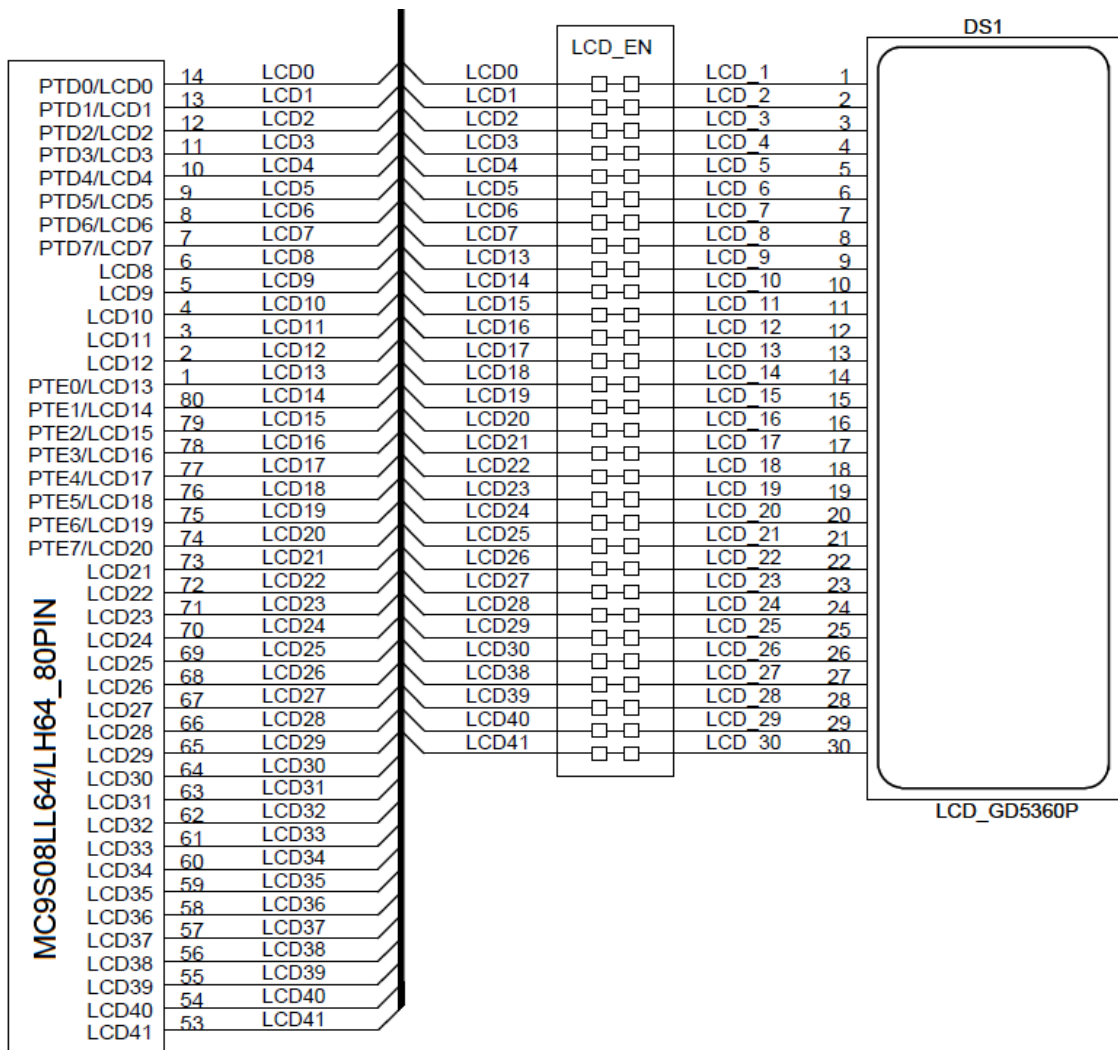


Obrázok 4: Detail indikátoru stavu nabitia batérie [5]

2.6 Prepojenie LCD displeja s mikrokontrolérom



Prepojenie LCD displeja GD-5360P s mikrokontrolérom MC9S08LH64 je zrejmé z obrázku 5, prehľad je uvedený v tabuľke 1. Pre zjednodušenie nie sú na obrázku 5 zobrazené všetky vývody mikrokontroléra. Konkrétne prepojenie je dôležité pre správne nastavenie niektorých registrov radiča LCD displeja.

V tabuľke 1 sa vyskytujú údaje v tvare napr. 1A. Z ostatných údajov na konkrétnom riadku tabuľky vyplýva, že medzi pinom LCD displeja číslo 5 (frontplane pin) a pinom LCD displeja číslo 30 (backplane pin *COM2*) je pripojený A segment sedemsegmentového znaku s pozíciou 1. Segmenty pripojené k rovnakému pinu mikrokontroléra sa oba ovládajú rovnakým registrom *LCDWF*. V tomto prípade je to register *LCDWF4*.



Obrázok 5: Prepojenie mikrokontroléra MC9S08LH64 s LCD displejom GD-5360P [6]

Tabuľka 1: Vývody LCD displeja GD-5360P [3][5][6]

Číslo vývodu LCD displeja	Backplane pin		Pin mikrokontroléra MC9S08LH64	LCDWF register
	COM1	COM2		
1	MO	T1	0	LCDWF0
2	P0	P1	1	LCDWF1
3	P3	P2	2	LCDWF2
4	P4	-----	3	LCDWF3
5	TU	1A	4	LCDWF4
6	1B	1F	5	LCDWF5
7	1C	1G	6	LCDWF6
8	1D	1E	7	LCDWF7
9	WE	COL	13	LCDWF13
10	TH	2A	14	LCDWF14
11	2B	2F	15	LCDWF15
12	2C	2G	16	LCDWF16
13	2D	2E	17	LCDWF17
14	FR	3A	18	LCDWF18
15		SA	19	LCDWF19
16		SU	20	LCDWF20
17	3B	3F	21	LCDWF21
18	3C	3G	22	LCDWF22
19	3D	3E	23	LCDWF23
20	AM	PM	24	LCDWF24
21	FLTR	HOLD	25	LCDWF25
22	4D	5D	26	LCDWF26
23	4E	4C	27	LCDWF27
24	4F	4G	28	LCDWF28
25	4A	4B	29	LCDWF29
26	5E	5C	30	LCDWF30
27	5F	5G	38	LCDWF38
28	5A	5B	39	LCDWF39
29	COM1	-----	40	LCDWF40
30	-----	COM2	41	LCDWF41

3 MIKROKONTROLÉR A RADIČ LCD DISPLEJA

3.1 Radič pre LCD displeje

V mikrokontroléri MC9S08LH64 je integrovaný radič pre LCD displeje. Tento radič dokáže pracovať v režime statickom a aj v režime multiplex, na prácu v režime multiplex je na viac optimalizovaný. [3]

V anglickej dokumentácii k mikrokontroléru MC9S08LH64 sa pre vývody pripojovaného LCD displeja používajú dva anglické výrazy: frontplane a backplane. Frontplane vývod je pripojený na vrchnú časť LCD displeja (vrchná polarizačná elektróda), backplane na zadnú časť (zadná polarizačná elektróda). Medzi elektródami sa nachádzajú tekuté kryštály. Ak sú segmenty zapojené do matice, frontplane vývody sú vývody riadkov (stĺpcov) a backplane sú vývody stĺpcov (riadkov). [3]

Integrovaný radič mikrokontroléra generuje potrebné napätia na riadenie LCD displejov, ktoré neobsahujú vlastný integrovaný radič. Použitá 80 vývodová verzia mikrokontroléra (na vývojovej doske TWR-S08LH64) disponuje 44 vývodmi použiteľnými na ovládanie LCD displejov. Sú označené nasledovne: *LCD0*, *LCD1*, ..., *LCD43*. Každý z nich môže fungovať ako frontplane alebo ako backplane vývod. Avšak maximálne 8 vývodov mikrokontroléra sa môže využiť ako vývod backplane. Doplňujúci počet vývodov k maximálnemu počtu vývodov LCD radiča (v tomto prípade k 44) sa môže použiť ako vývody vo funkcii frontplane. [3]

Radič má nastaviteľnú rámcovú frekvenciu, frekvenciu blikania segmentov a výstupné napätie na riadenie segmentov. [3]

3.2 Hodinový signál pre LCD radič

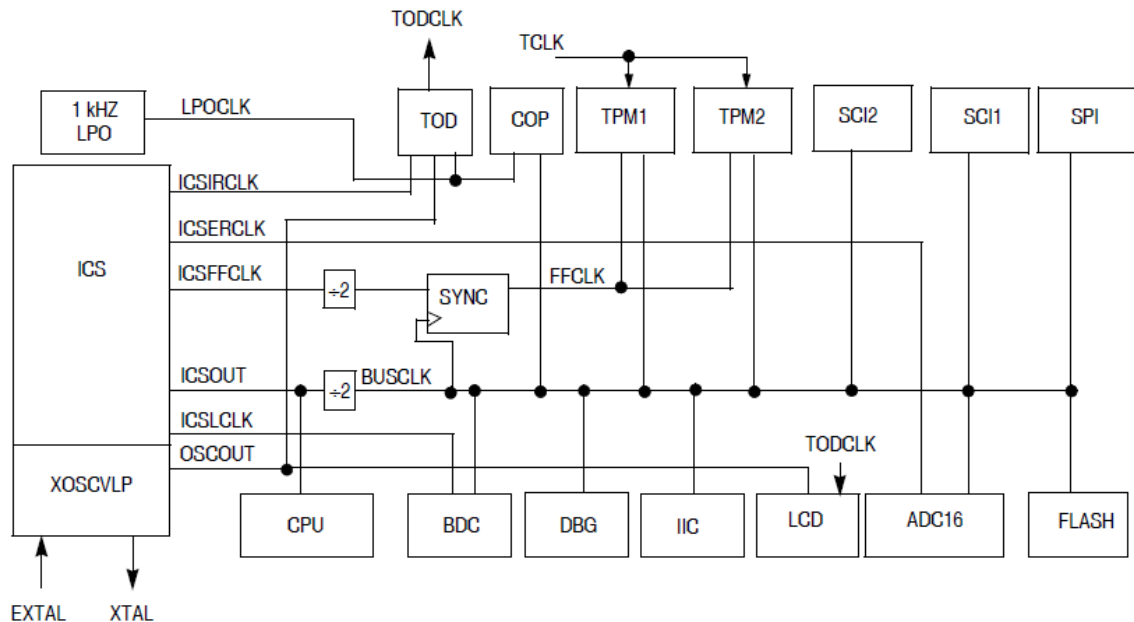
Modul LCD radiča môže byť taktovaný dvoma zdrojmi hodinového signálu: *ALTCLK* alebo *OSCOUT* (obrázok 6). Od zvoleného signálu sa odvíja rámcová frekvencia obnovovania segmentov LCD displeja, frekvencia blikania segmentov a pracovný kmitočet nábojovej pumpy. [3]

Vstup hodinového signálu *ALTCLK* do radiča je výstupom modulu *TOD* (Time of Day) – signál *TODCLK*. Modul *TOD* má na svojom vstupe deličku kmitočtu vybraného vstupného hodinového signálu, ktorým môže byť: *LPO* – 1kHz interný oscilátor, *OSCOUT* – externý hodinový signál alebo *ICSIRCLK* – vnútorný referenčný hodinový signál z modulu *ICS* (Internal Clock Source). [3]

Druhou možnosťou zdroja hodinového signálu pre LCD radič je priamo signál *OSCOUT*, ktorý je výstupom modulu *XOSCVLP* (External oscillator module). Tento modul umožňuje priviesť ľubovoľný externý hodinový signál (privedený na pin mikrokontroléru *EXTAL*) alebo hodinový signál vytvorený externým kryštálom, pripojeným medzi piny mikrokontroléru *XTAL* a *EXTAL*, na zbernicu signálu *OSCOUT*.

Vývojová doska TWR-S08LH64 obsahuje práve takto zapojený kryštál s kmitočtom 32,768 kHz a pri tomto kmitočte sa dosahuje najnižšia spotreba energie radiča. [3]

Preto najvýhodnejším spôsobom je použitie práve tohto kryštálu ako zdroj hodinového signálu pre LCD radič.



Obrázok 6: Zdroje hodinových signálov mikrokontroléra MC9S08LH64 [3]

3.2.1 Inicializácia hodinového signálu pre LCD radič

Ešte pred inicializáciou radiča LCD displeja, je nutné konfigurovať zdroj hodinových impulzov pre LCD radič. V prechádzajúcej časti textu je vysvetlená výhoda použitia externého kryštálu. Hodinový signál z tohto zdroja je potrebné nastaviť a povoliť v registri *ICSC2*, ktorý ovláda modul *ICS* a *XOSCVLP*. [3]

Bitom *EREFS* (External Reference Select) sa nastavuje typ použitého externého zdroja hodinového signálu. Pretože sa bude používať externý kryštál – oscilátor, je potrebné tento bit nastaviť na hodnotu log. 1. [3]

Bitom *RANGE* sa spresní rozsah použitej frekvencie oscilátora. Na doske TWR-S08LH64 je použitý 32,768kHz kryštál. Táto frekvencia patrí v rozsahu k najnižším, preto je treba zapísať do bitu *RANGE* hodnotu log. 0. [3]

Vývojová doska TWR-S08LH64 je optimalizovaná aj pre režim s nízkym odberom energie. Hodnota log. 0 v bite *HGO* konfiguruje oscilátor pre nízky odber. [3]

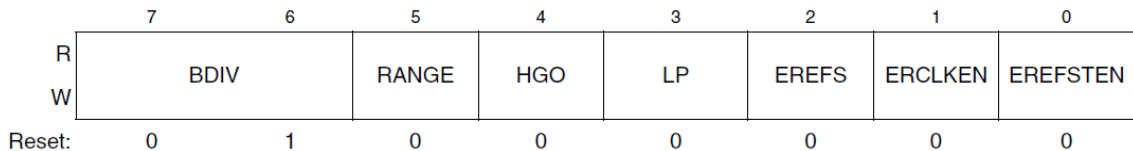
Zápisom log. 1 do bitu *ERCLKEN* (External Reference Enable) sa povoľuje činnosť modulu *XOSCVLP* (External oscillator module). [3]

Mikrokontrolér MC9S08LH64 dokáže pracovať vo viacerých módoch, pri ktorých dokáže napríklad šetriť energiu (pri napájaní z batérií) alebo deaktivovať niektoré vnútorné moduly. Bit *EREFSTEN* povoľuje externý hodinový signál aj pri operovaní v

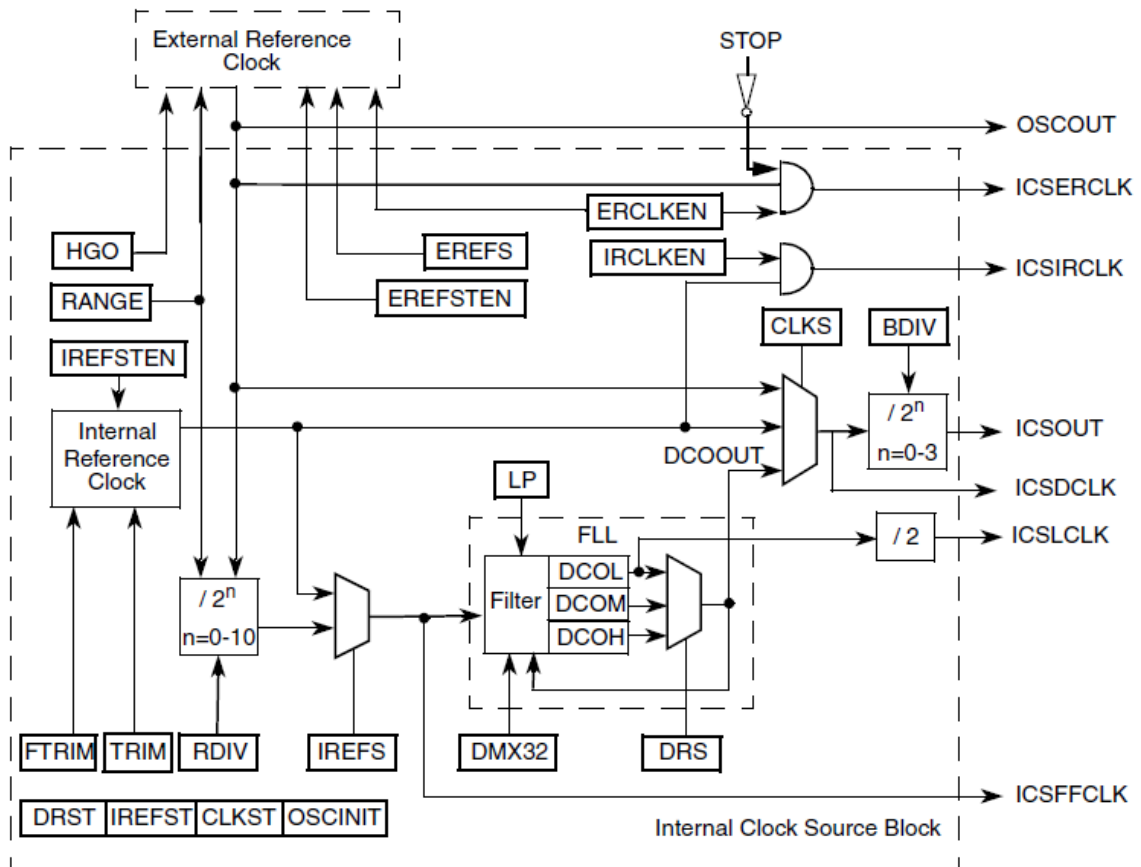
stop módoch. Toto riešenie ovládania displeja tieto módy neuvažuje a preto sa bit *EREFSTEN* nastavuje na hodnotu log. 0. [3]

Pri zápise do registra *ICSC2* je potrebné zachovať hodnoty ostatných bitov po resete, pretože ovplyvňujú modul ICS. Modul ICS je zdrojom hodinových signálov pre ďalšie moduly mikrokontroléra.

Inicializácia hodinového signálu pre radič LCD displeja bude v nasledovnom tvare: *ICSC2 = 0b01000110*;



Obrázok 7: Register ICSC2 – ICS Control Register 2 [3]



Obrázok 8: Moduly ICS (Internal Clock Source) a XOSCVLP (External Reference Clock) [3]

3.3 Zdroje napätia pre budenie segmentov

Radič LCD displeja vytvára na všetkých nastavených backplane a frontplane vývodoch také napäťové úrovne, aby sa polarizovali len požadované segmenty displeja. Pracuje pri tom s tromi úrovňami napätia – V_{LL1} , V_{LL2} a V_{LL3} , kde V_{LL1} je rovné tretine

V_{LL3} a V_{LL2} dvom tretinám V_{LL3} . Napätie V_{LL3} musí byť také napätie, aké je potrebné na vybudenie segmentu. Tieto napätia sú vzťahované k potenciálu 0V (zem vývojovej dosky TWR-S08LH64). Blokovo schéma zdrojov napájania segmentov LCD displeja je na obrázku 9. Tabuľka 2 uvádza stav prepínačov powersw v závislosti na zvolenej možnosti generovania napätí V_{LL1} až V_{LL3} – nastavením bitov VSUPPLY v registri LCDSUPPLY (viď kapitola 3.4).

Dizajn radiča umožňuje nastavenie viacerých spôsobov generovania, resp. privedenia týchto troch napätí. Dajú sa rozdeliť na riešenia bez alebo s použitím vnútornej nábojovej pumpy (angl. charge pump) ako zdroja napätí V_{LL1} až V_{LL3} . Nábojová pumpa je DC-DC menič napätia, ktorý dokáže svoje vstupné napätie znižovať a aj zväčšovať. [3]

Bez použitia nábojovej pumpy existuje jedna možnosť vytvorenia napätí V_{LL1} až V_{LL3} – pripojenie externého napätia V_{LL3} na pin mikrokontroléra MC9S08LH64 s označením VLL3. Napätia V_{LL1} a V_{LL2} sú potom odvodené od V_{LL3} deličom napätia, V_{LL1} je rovné tretine V_{LL3} a V_{LL2} dvom tretinám V_{LL3} . Nevýhodou použitého rezistorového deliča napätia je to, že takto vytvorené napätia V_{LL1} až V_{LL3} sú závislé od odoberaného prúdu, čo môže byť problém pri displejoch s väčším počtom riadených segmentov. [3]

S použitím nábojovej pumpy existuje viacero možností nastavenia napätí V_{LL1} až V_{LL3} . O generovanie a reguláciu napätia sa stará nábojová pumpa. Ako referenčné napätie pre nábojovú pumpu je možné použiť externé napätie V_{LCD} , napätie vnútorného regulovateľného zdroja V_{IREG} , alebo napájacie napätie mikrokontroléra V_{DD} . [3]

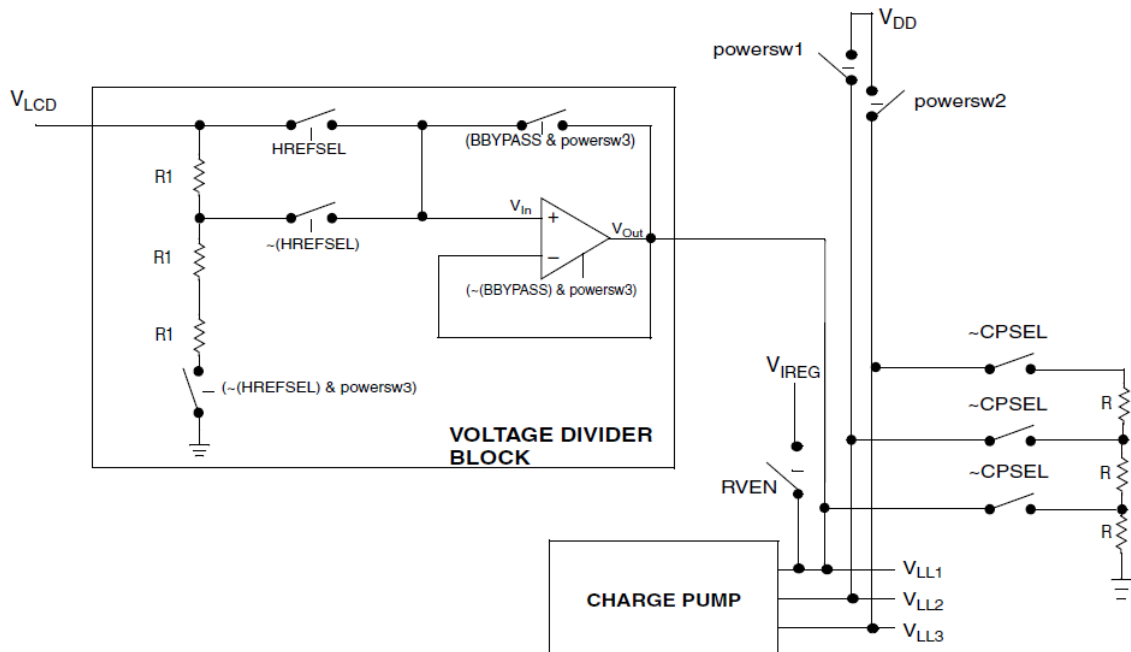
Pri použití vonkajšieho napätia V_{LCD} , ktoré je pripojené na VLCD pin mikrokontroléra, je nutné zabezpečiť, aby bolo toto napätie v rozsahu 0,9 až 1,8 V. Napätie V_{LCD} bude referenčným napätím nábojovej pumpy V_{LL1} , napätie V_{LL2} bude dvojnásobok a V_{LL3} trojnásobok V_{LL1} . Pred vstupom napätia V_{LCD} do vstupu nábojovej pumpy V_{LL1} je ešte možné pomocou vstupného deliča zmenšiť napätie V_{LCD} na hodnotu $V_{LCD} \cdot (2/3)$. Pri použití tohto deliča sa ale zmení hodnota vnútorného odporu takto upraveného zdroja napätia, čo by mohlo na ňom pri vyššom počte ovládaných segmentov displeja spôsobovať úbytky napätia. Preto je možné v obvode (správnou konfiguráciou registrov) zapojiť operačný zosilňovač zapojený ako napäťový sledovač, ktorý má na svojom výstupe rovnaké napätie ako na neinvertujúcom vstupe. Jeho vstupná impedancia je dostatočne vysoká, že nezaťažuje vstupný delič zapojený na vstupe V_{LCD} a zároveň operačný zosilňovač zmenší výstupný odpor takto upraveného signálu, čím sa dá vyhnúť vyššie spomínanému problému s úbytkami napätia. [3]

Ďalšou možnosťou je použitie vnútorného regulovateľného zdroja napätia V_{IREG} , ktorý sa dá nastaviť na 1 V alebo 1,67 V. Napätie V_{IREG} je pre nábojovú pumpu referenčným napätím V_{LL1} . Napätie V_{LL2} bude ako v predchádzajúcom prípade dvojnásobok a V_{LL3} trojnásobok napätia V_{LL1} . [3]

Napätie V_{DD} sa dá použiť ako referenčné napätie pre napätie V_{LL2} alebo V_{LL3} . Nábojová pumpa sa znova postará o rozloženie napätí V_{LL1} až V_{LL3} tak, aby V_{LL2} bolo dvojnásobok a V_{LL3} trojnásobok V_{LL1} . [3]

Na vývojovej doske TWR-S08LH64 nie sú piny mikrokontroléra VLL1, VLL2, VLL3 ani VLCD vyvedené k žiadnej zbernici ani konektoru. Preto sa pri predpoklade nezasahovania do zapojenia vývojovej dosky nedá využiť ani možnosť pripojenia referenčného napätia z pinu VLCD ani pripojenie referenčného napätia na pin VLL3.

Najlepšie riešenie je využiť nábojovú pumpu spolu s vnútorným nastaviteľným zdrojom napätia (napätie V_{IREG}). Týmto výberom je možné jemne doladiť veľkosť napätia V_{LL1} a tým v malom rozsahu ovplyvňovať kontrast LCD displeja.



Obrázok 9: Bloková schéma zdrojov napájania pre radič LCD displeja [3]

Tabuľka 2: Konfigurácia prepínačov napájania [3]

VSUPPLY [1:0]	Konfigurácia napájania	Stav prepínačov powersw		
		powersw1	powersw2	powersw3
00	Nastavenie V_{LL2} na napätie V_{DD}	1	0	0
01	Nastavenie V_{LL3} na napätie V_{DD}	0	1	0
10	Nastavenie V_{LL1} na napätie V_{LCD} (externé napätie na pine VLCD)	0	0	1
11	Nastavenie V_{LL3} z externého zdroja (napätím na pine VLL3)	0	0	0
	Nastavenie V_{LL1} na napätie V_{IREG} (vnútorný regulovateľný zdroj)			

3.4 Registre LCD radiča

Displej sa ovláda pomocou zápisu vhodných hodnôt v nasledujúcich registroch LCD radiča. Nachádzajú sa vo flash pamäti mikrokontroléra MC9S08LH64. [3]

LCD Control Register 0	<i>LCDC0</i>
LCD Control Register 1	<i>LCDC1</i>
LCD Voltage Supply Register	<i>LCDSUPPLY</i>
LCD Regulated Voltage Control Register	<i>LCDRVC</i>
LCD Blink Control Register	<i>LCDBCTL</i>
LCD Status Register	<i>LCDS</i>
LCD Pin Enable Registers 0 – 7	<i>LCDPEN0 – LCDPEN7</i>
Backplane Enable Registers 0 – 7	<i>BPEN0 – BPEN7</i>
LCD Waveform Registers	<i>LCDWF0 – LCDWF63</i>

LCD radič disponuje mnoho funkciami spolu s ich variáciami. Prehľad funkcií jednotlivých bitov vo všetkých registroch LCD radiča by bol príliš obsiahly a zložitý, preto je výhodnejšie popísať jeho funkcie na konkrétnom príklade so zadanou vývojovou doskou TWR-S08LH64 a s displejom GD-5360P.

3.4.1 Inicializácia LCD radiča

Pred tým, ako je možné ovládať stav segmentov LCD displeja je nutné vykonať inicializáciu radiča. Inicializáciou sa rozumie výber zdroja hodinového signálu, konfigurácia zdroja napätí V_{LL1} až V_{LL3} , nastavenie rámcovej frekvencie a frekvencie blikania segmentov, povolenie funkčnosti vývodov LCD radiča, nastavenie počtu a fyzického prepojenia spoločných vývodov displeja (vývody backplane).

Doporučený postup inicializácie radiča je uvedený v špecifikácii mikrokontroléra MC9S08LH64, ďalej uvádzané riešenie sa ho pridrža.

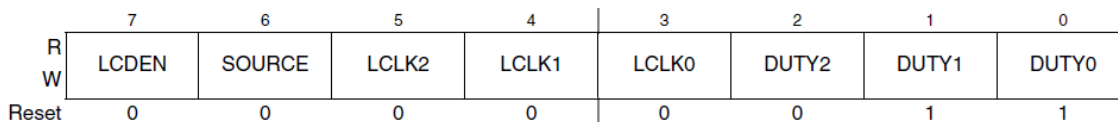
Pred konfiguráciou registra *LCDC0* sa musí v tomto registri vynulovať bit *LCDEN*. Po resete mikrokontroléra je táto podmienka splnená a nie je preto nutné tento bit zvlášť nulovať. [3]

V registri *LCDC0* je potrebné vybrať zdroj hodinového signálu pre radič. Zápis log.0 do bitu *SOURCE* nastaví ako zdroj hodinového signálu externý 32,768kHz kryštál. Vzhľadom na vnútorné zapojenie LCD displeja GD-5360P je potrebné, aby radič pracoval v multiplexovanom režime. Bity *DUTY[2:0]* nastavujú počet spoločných vývodov displeja (backplane vývodov). Displej GD-5360P má dva vývody backplane ($DUTY = \frac{1}{2}$), preto je potrebné nastaviť bity *DUTY[2:0]* na 0b001. [3][5]

Bity *LCLK[2:0]* nastavujú rámcovú frekvenciu budenia segmentov, frekvencia musí byť v povolenom rozsahu radiča od 28 do 58 Hz. Pre stĺpec v tabuľke 3, kde je Duty Cycle rovný $\frac{1}{2}$, sa vyberie platná hodnota, napr. 32 Hz, potom $LCLK[2:0] = 4$, teda binárne 0b100. Hodnoty v tabuľke 3 sú platné pre hodinový signál LCD radiča

s frekvenciou 32,768 kHz. Ak by boli segmenty displeja zobrazované prerušovane, je potrebné zvýšiť rámcovú frekvenciu znížením hodnoty v bitoch $LCLK[2:0]$. [3]

Zápis hodnoty do registra $LCDC0$ bude v tvare $LCDC0 = 0b00100001$;



Obrázok 10: Register LCDC0 (LCD Control Register 0) [3]

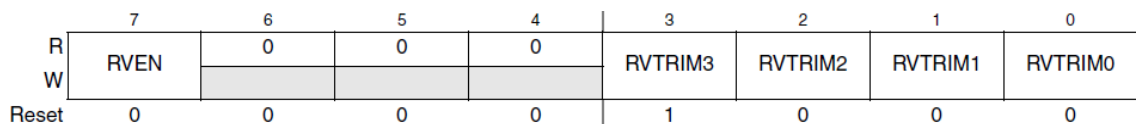
Tabuľka 3: Hodnoty rámcovej frekvencie [3]

Duty Cycle	1/1	1/2	1/3	1/4	1/5	1/6	1/7	1/8
$LCLK[2:0]$	Rámcová frekvencia [Hz]							
0	64	64	68,3	64	68,3	56,9	73,1	64
1	51,2	51,2	54,6	51,2	54,6	45,5	58,5	51,2
2	42,7	42,7	45,5	42,7	45,5	37,9	48,8	42,7
3	36,6	36,6	39	36,6	39	32,5	41,8	36,6
4	32	32	34,1	32	34,1	28,4	36,6	32
5	28,4	28,4	30,3	28,4	30,3	25,3	32,5	28,4
6	25,6	25,6	27,3	25,6	27,3	22,8	29,3	25,6
7	23,3	23,3	24,8	23,3	24,8	20,7	26,6	23,3

V oddiele 3.3 bola vysvetlená výhodnosť použitia regulovateľného napätia V_{IREG} spolu s nábojovou pumpou. Zdroj napätia V_{IREG} sa aktivuje zápisom log. 1 do bitu $RVEN$ v registri $LCDRVC$. Jeho dolné 4 bity ($RVTRIM[3:0]$) umožňujú doladiť výstupné napätie tohto zdroja až o $\pm 12\%$ od základnej hodnoty, ktorá môže byť 1 V alebo 1,67 V. Iniciálne hodnota v bitoch $RVTRIM[3:0]$ je dekadicky číslo 8 a predstavuje 100% základnej hodnoty napätia. Základná hodnota napätia sa nastavuje bitom $HREFSEL$ v registri $LCDSUPPLY$. [3]

LCD displej GD-5360P potrebuje pre zobrazenie segmentu napätie 2,7 V a preto napätie V_{LL3} musí mať minimálne takúto úroveň. Keďže je toto napätie trojnásobkom napätia V_{LL1} a napätie V_{LL1} sa dá nastaviť na 1 V alebo 1,67 V, je nutné zvoliť možnosť 1 V (zápisom log. 0 do bitu $HREFSEL$ v registri $LCDSUPPLY$) a následne doladiť bitmi $RVTRIM[3:0]$ na požadovanú hodnotu V_{LL3} . Pre dosiahnutie V_{LL3} rovné približne riadiacemu napätiu LCD displeja 2,7 V sa do bitov $RVTRIM[3:0]$ zapíše hodnota 2 (0b0010). Zníženie hodnoty $RVTRIM[3:0]$ o 6 od iniciálnej hodnoty 8 znížilo napätie V_{IREG} 6-krát o 1,5 % od základnej hodnoty napätia. Napätie V_{LL1} sa rovná napätiu V_{IREG} a bude približne 0,91 V, potom napätie V_{LL3} bude približne 2,73 V. [3][5]

Zápis do registra $LCDRVC$ bude v tvare $LCDRVC = 0b10000010$;



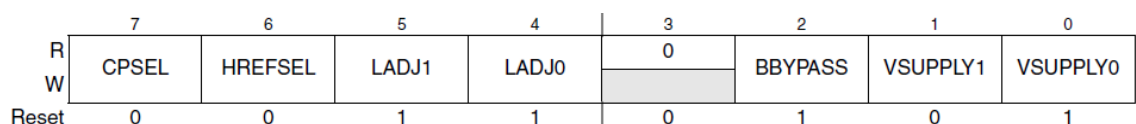
Obrázok 11: Register LCDRVC – LCD Regulated Voltage Control Register [3]

Nábojová pumpa sa povoľuje zapísaním log. 1 do bitu *CPSEL* v registri *LCDSUPPLY*. Z predchádzajúceho odseku vyplýva, že pre nastavenie základnej hodnoty napätia V_{IREG} na 1 V je nutné, aby mal bit *HREFSEL* hodnotu log. 0. Pomocou bitov *LADJ0* a *LADJ1* je možní upraviť frekvenciu hodinového signálu, ktorý riadi činnosť nábojovej pumpy. Pri potrebe riadiť a teda aj napájať rozsiahlejší LCD displej vzrastajú nároky na tvrdosť zdroja napätí V_{LL1} až V_{LL3} , v tomto prípade na nábojovú pumpu. Zvýšením frekvencie sa zväčší jej tvrdosť. Nedostatočná frekvencia pre nábojovú pumpu sa prejaví na intenzite segmentov displeja. Iniciálna hodnota oboch bitov *LADJ* je po resete log. 1, pri inicializácii sa tieto dve hodnoty budú zápisom *LADJ0* = 1 a *LADJ1* = 1 zachovávať. Záleží na subjektívnom pohľade užívateľa na intenzitu segmentov, ktorú môže v prípade potreby zmenou týchto dvoch bitov jednoducho zmeniť. [3]

Na nastavení bitu *BBYPASS* v tomto konkrétnom prípade nezáleží, pretože sa nevyužíva možnosť priviesť externé referenčné napätie V_{LCD} . Pri inicializácii sa bude zapisovať na pozíciu tohto bitu iniciálna hodnota po resete mikrokontroléra, teda *BBYPASS* = 1. [3]

Posledné dva bity v registri *LCDSUPPLY* upresňujú, ktorá možnosť riadenia napájania sa bude využívať. Pretože sa na generovanie napätia V_{LL1} využíva vnútorný nastaviteľný zdroj (napätie V_{IREG}), nastaví sa bity *VSUPPLY0* a *VSUPPLY1* na hodnotu log. 1. [3]

Zápis do tohto registra bude v tvare *LCDSUPPLY* = 0b10110111;



Obrázok 12: Register LCDSUPPLY – LCD Voltage Supply Register [3]

Nie všetky piny mikrokontroléra, ktoré sa dajú použiť ako vývody LCD radiča, sa skutočne využívajú. Je preto potrebné povoliť len tie, ktoré sa budú využívať. Na to slúžia registre *LCDPEN0* až *LCDPEN7*. Keďže mikrokontrolér MC9S08LH64 disponuje len 44 LCD vývodmi, nastaviť sa musí len 6 registrov *LCDPEN* (*LCDPEN0* až *LCDPEN5*), ostatné registre nie sú v pamäti vyhradené, neexistujú. [3]

K mikrokontroléru je pripojených všetkých 30 vývodov displeja GD-5360P. Sú pripojené k vývodom *LCD0* až *LCD7*, *LCD13* až *LCD30*, *LCD38* až *LCD41* (obrázok 5). Bity v registroch *LCDPEN0* až *LCDPEN5* sú očíslované a vždy svojou logickou hodnotou povoľujú alebo zakazujú funkciu LCD pinu mikrokontroléra. Číslo bitu *PEN* korešponduje s číslom výstupu radiča LCD displeja (bit *PEN0* povoľuje alebo

zakazuje pin s označením *LCD0*). Podľa prepojenia mikrokontroléra MC9S08LH64 a LCD displeja GD-5360P je nutné nastaviť jednotlivé bity a registre nasledovne: Bity *PEN0* až *PEN7*, *PEN13* až *PEN30* a *PEN38* až *PEN41* sa nastavujú na hodnotu log. 1 (povolenie funkčnosti vývodu radiča). Ostatné bity sa nastavujú na hodnotu log. 0, pretože po resete nemajú tieto bity definovaný iníciaľny stav a je nutné ich stav inicializovať na správnu hodnotu. [3]

Zápis do jednotlivých registrov *LCDPEN* bude vypadáť takto:

LCDPEN0 = 0b11111111; *LCDPEN3* = 0b01111111;
LCDPEN1 = 0b11100000; *LCDPEN4* = 0b11000000;
LCDPEN2 = 0b11111111; *LCDPEN5* = 0b00000011;

		7	6	5	4	3	2	1	0
<i>LCDPEN0</i>	R								
	W	PEN7	PEN6	PEN5	PEN4	PEN3	PEN2	PEN1	PEN0
	Reset	Indeterminate after reset							
<i>LCDPEN1</i>	R								
	W	PEN15	PEN14	PEN13	PEN12	PEN11	PEN10	PEN9	PEN8
	Reset	Indeterminate after reset							
<i>LCDPEN2</i>	R								
	W	PEN23	PEN22	PEN21	PEN20	PEN19	PEN18	PEN17	PEN16
	Reset	Indeterminate after reset							
<i>LCDPEN3</i>	R								
	W	PEN31	PEN30	PEN29	PEN28	PEN27	PEN26	PEN25	PEN24
	Reset	Indeterminate after reset							
<i>LCDPEN4</i>	R								
	W	PEN39	PEN38	PEN37	PEN36	PEN35	PEN34	PEN33	PEN32
	Reset	Indeterminate after reset							
<i>LCDPEN5</i>	R								
	W	PEN47	PEN46	PEN45	PEN44	PEN43	PEN42	PEN41	PEN40
	Reset	Indeterminate after reset							

Obrázok 13: *LCDPEN0* až *LCDPEN5* – LCD Pin Enable Registers 0–5 [3]

Pre generovanie správnych priebehov napätí na LCD pinoch pripojených k displeju je ešte potrebné nastaviť, ku ktorým vývodom radiča sú pripojené spoločné vývody displeja, tzv. backplane vývody (*COM1* a *COM2*). Podľa obrázku 5 alebo tabuľky 1 sú vývody s označením *COM1* a *COM2* pripojené k pinom radiča *LCD40* a *LCD41*. Nastavenie vývodu radiča do funkcie vývodu backplane sa vykoná zápisom log. 1 do bitov *BPEN40* a *BPEN41* (v registri *LCDBPEN*). Do ostatných bitov v registroch *LCDBPEN0* až *LCDBPEN5* je nutné zapísať log. 0, pretože po resete nemajú tieto registre definovaný iníciaľny stav a je nutné ich inicializovať na správnu hodnotu. [3]

Zápis do jednotlivých registrov *LCDBPEN* bude vypadáť takto:

LCDBPEN0 = 0b00000000; *LCDBPEN3* = 0b00000000;
LCDBPEN1 = 0b00000000; *LCDBPEN4* = 0b00000000;

$LCDBPEN2 = 0b00000000;$

$LCDBPEN5 = 0b00000011;$

		7	6	5	4	3	2	1	0
LCDBPEN0	R								
	W	BPEN7	BPEN6	BPEN5	BPEN4	BPEN3	BPEN2	BPEN1	BPEN0
	Reset	Indeterminate after reset							
LCDBPEN1	R								
	W	BPEN15	BPEN14	BPEN13	BPEN12	BPEN11	BPEN10	BPEN9	BPEN8
	Reset	Indeterminate after reset							
LCDBPEN2	R								
	W	BPEN23	BPEN22	BPEN21	BPEN20	BPEN19	BPEN18	BPEN17	BPEN16
	Reset	Indeterminate after reset							
LCDBPEN3	R								
	W	BPEN31	BPEN30	BPEN29	BPEN28	BPEN27	BPEN26	BPEN25	BPEN24
	Reset	Indeterminate after reset							
LCDBPEN4	R								
	W	BPEN39	BPEN38	BPEN37	BPEN36	BPEN35	BPEN34	BPEN33	BPEN32
	Reset	Indeterminate after reset							
LCDBPEN5	R								
	W	BPEN47	BPEN46	BPEN45	BPEN44	BPEN43	BPEN42	BPEN41	BPEN40
	Reset	Indeterminate after reset							

Obrázok 14: Registre LCDBPEN0 až LCDBPEN5 – Backplane Enable Registers 0–5 [3]

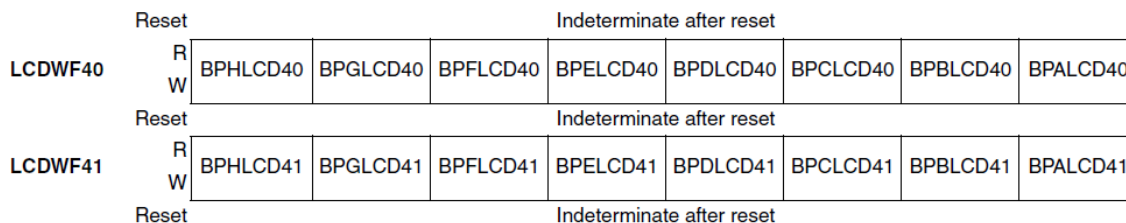
Radič displeja pracuje v režime multiplex a pretože má ovládaný displej dva spoločné vývody ($DUTY = \frac{1}{2}$), ovládajú sa počas jednej periódy rámcovej frekvencie práve dve fázy. Počas jednej periódy sa segmenty každej fázy budia vždy dvakrát, raz kladným a raz záporným napätím. Segmenty sa budia striedavým napätím preto, aby nenastala degradácia tekutých kryštálov. Priradenie, ktorý backplane vývod LCD displeja bude ovládaný v jednej z dvoch fáz, sa nastavuje zápisom log. 1 do práve jedného bitu v každom z registrov *LCDWF* ovládajúcich backplane vývody. Fázy sú pomenované písmenami *A* až *H* a v registroch *LCDWF* sa vyskytujú aj v názve jednotlivých bitov. Poradové číslo *LCDWF* registra nastavujúceho fázu opäť zodpovedá fyzickému prepojeniu displeja a mikrokontroléra. Na vývojovej doske TWR-S08LH64 má LCD displej GD-5360P prepojené backplane vývody *COM1* a *COM2* k vývodom mikrokontroléra *LCD40* a *LCD41* a ovládajú sa teda registrami *LCDWF40* a *LCDWF41*. Keďže $DUTY = \frac{1}{2}$, aktívne sú len fázy *A* a *B*. Ostatné fázy nie sú vtedy aktívne. [3][5]

Pri využití možnosti radiča rozblikať segmenty na displeji s menej ako štyrmi vývodmi backplane sa počet a poradie fáz modifikuje. V tomto konkrétnom prípade sa nebudú pravidelne striedať len normálne fázy *A* a *B*, ale pridajú sa ešte tzv. alternatívne fázy *k* fázam *A* a *B*, fázy *E* a *F*, ktoré je ale taktiež nutné povoliť, resp. priradiť vývody backplane aj k týmto alternatívnym fázam. Backplane vývod *COM1* sa priradí napríklad k fáze riadenia *A* a *E*, *COM2* k fáze riadenia *B* a *F*.

Zápis do jednotlivých registrov *LCDWF* bude vypadať takto:

$LCDWF40 = 0b00010001;$

$LCDWF41 = 0b00100010;$



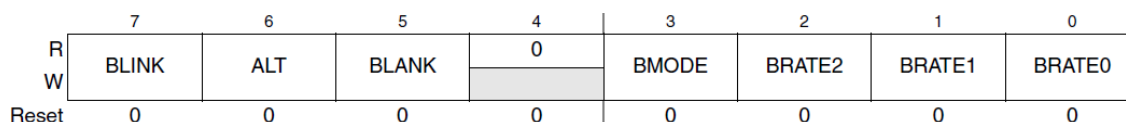
Obrázok 15: LCDWF40 a LCDWF41 – LCD Waveform Registers [3]

Radič pre LCD displej disponuje s už vyššie spomínanou možnosťou zobrazovať blikajúce segmenty. Nastavenia sa vykonávajú v registri *LCDBCTL*. Najnižšie tri bity *BRATE[2:0]* určujú frekvenciu blikania segmentov. Bity *ALT*, *BLANK* a *BMODE* nastavujú jeden z viacerých možných sekvencií blikania displeja. [3]

Vhodným zápisom dát do frontplane *LCDWF* registrov je možné vytvoriť dve rôzne zobrazenia segmentov na displeji, radič sa postará o ich striedavé zobrazovanie – vid' obrázok 18 (zobrazenie v normálnej a v alternatívnej fáze). Pre dosiahnutie trvalého zobrazenia niektorých segmentov a blikania niektorých iných segmentov je tento typ sekvencie najvýhodnejší. V registri *LCDBCTL* je preto potrebné nastaviť bit *ALT* na log. 0, bit *BLANK* na log. 0 a bit *BMODE* na log. 1. [3]

Pri zobrazení hodín na LCD displeji je efektné, aby dvojbodka oddeľujúca hodiny od minút blikala s frekvenciou 1 Hz. Pri použití externého 32,768kHz kryštálu ako zdroja hodinového signálu pre LCD radič sa pre dosiahnutie žiadanej frekvencie blikania 1 Hz zapíše do bitov *BRATE[2:0]* hodnota 0b011. Pre povolenie činnosti modulu blikania sa zapíše do bitu *BLINK* log. 1. [3]

Zápis do registra *LCDBCTL* bude v tvare $LCDBCTL = 0b10001011$;



Obrázok 16: Register *LCDBCTL* – LCD Blink Control Register [3]

Registre *LCDWF* zodpovedajúce frontplane vývodom displeja ovládajú stav jeho segmentov. Aby po inicializácii neboli na LCD displeji zobrazené žiadne segmenty, je potrebné do týchto frontplane *LCDWF* registrov zapísať hodnotu nula, pretože po resete nemajú tieto registre definovaný iníciaľny stav. Jedná sa o registre *LCDWF0* až *LCDWF7*, *LCDWF13* až *LCDWF30*, *LCDWF38* a *LCDWF39*. Obdobné platí aj pre registre backplane vývodov displeja (*LCDWF40* a *LCDWF41*). Nestačí v nich nastaviť len potrebné bity, ale aj všetky ostatné bity inicializovať na hodnotu log. 0. [3]

Posledným krokom inicializácie je spustenie činnosti radiča LCD a to zápisom hodnoty log. 1 do bitu *LCDEN* v registri *LCDC0*. [3]

Uvádzané riešenie inicializácie nenastavuje niektoré registre LCD radiča, pretože sa nevyužívajú všetky jeho možnosti. Jedná sa o menej podstatné nastavenia v registroch *LCDC1* a *LCDS*.

3.4.2 Ovládanie stavu segmentov LCD displeja

Ku každému frontplane vývodu LCD displeja GD-5360P (okrem vývodu č. 4) sú pripojené vždy dva segmenty. Stav týchto segmentov ovláda register *LCDWF*, ktorý je daný konkrétnym prepojením LCD displeja s mikrokontrolérom (viď tabuľka 1). Napríklad vývod displeja číslo 1 je pripojený k vývodu mikrokontroléra *LCD0*. Potom segmenty pripojené vývodu displeja č.1 (*MO* a *TI*) ovláda register *LCDWF0*.

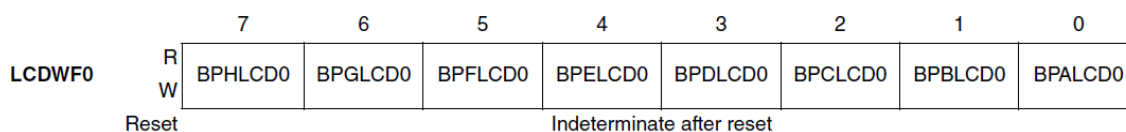
Register *LCDWF* je osembitový, každý jeho bit ovláda jeden segment – práve ten segment, ktorý je pripojený k vývodu backplane priradenému v inicializácii ku konkrétnej fáze riadenia. V inicializácii LCD radiča bol backplane vývod *COM1* priradený k normálnej fáze *A* a k alternatívnej fáze *E*, backplane vývod *COM2* k normálnej fáze *B* a k alternatívnej fáze *F*. [3]

Pre trvalé zobrazenie segmentu je potrebné zapísať hodnotu log. 1 do príslušných bitov (oboch fáz – normálnej aj alternatívnej) podľa backplane vývodu segmentu do registra *LCDWF* (určeného podľa frontplane pinu segmentu). Zhasnutie segmentu sa realizuje zápisom log. 0 do rovnakých bitov v rovnakých registroch. [3]

Pre zobrazenie blikajúceho segmentu sa zapíše hodnota log. 1 do bitu prislúchajúcemu práve jednej fáze riadenia (normálnej alebo alternatívnej). Do bitu ovládajúcemu segment v opačnej (alternatívnej alebo normálnej) fáze sa musí zapísať hodnota log. 0.

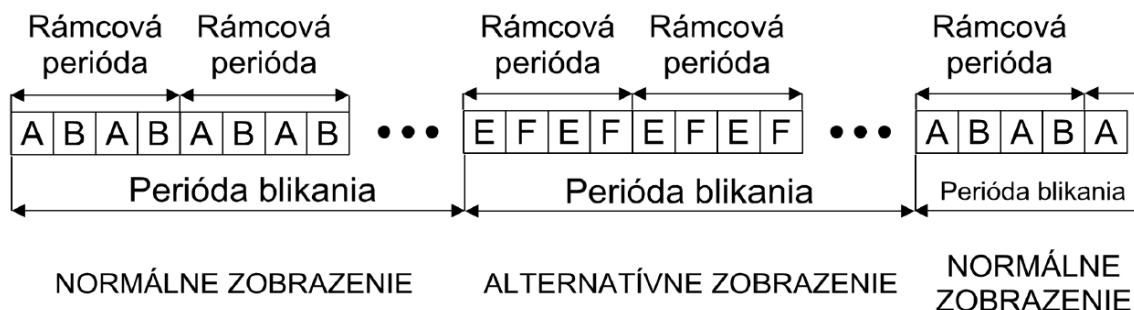
Napríklad segment *MO* je pripojený medzi frontplane pin č.1 a backplane vývod *COM1*. Pre trvalé zobrazenie tohto segmentu sa zapíše hodnota log. 1 do bitov pre fázu *A* a *E* (*BPALCD0* a *BPELCD0*) v registri *LCDWF0* (frontplane pin č.1). Segment *TI* je pripojený k rovnakému frontplane pinu, ale k inému backplane vývodu – ku *COM2*. Pre jeho trvalé zobrazenie sa zapíše hodnota log. 1 do bitov pre fázu *B* a *F* (*BPBLCD0* a *BPFLCD0*) v rovnakom registri – register *LCDWF0*.

Pre zobrazenie blikajúceho segmentu sa zapíše hodnota log. 1 len do jedného bitu (do jednej fázy), napríklad pre segment *MO* do bitu *BPALCD0*. Do bitu pre alternatívnu fázu k normálnej fáze *A*, fáze *E* (*BPELCD0*), sa zapíše naopak hodnota log. 0. Segment bude blikať s nastavenou frekvenciou 1 Hz, pretože sa pravidelne striedajú fázy *A* a *B*, s fázami *E* a *F* (obrázok 18). Iný segment, napríklad *TI* sa môže nastaviť opačne. Je pripojený ku *COM2*, preto sa do bitu pre fázu *B* (*BPBLCD0*) zapíše hodnota log. 0 a do bitu pre alternatívnu fázu *F* (*BPFLCD0*) hodnota log. 1. Segment *TI* bude blikať, avšak v protifáze oproti segmentu *MO*. Vždy teda bude zobrazený iba jeden z nich. Ďalšie kombinácie sú uvedené v tabuľke 4, kde sú uvedené niektoré možnosti ovládania stavu segmentov ovládaných pomocou registra *LCDWF0* (segmenty *MO* a *TI*).



Obrázok 17: Register *LCDWF0* – LCD Waveform Register 0 [3]

Rámcová frekvencia bola v inicializácii radiča nastavená na 32 Hz, frekvencia blikania na 1 Hz. Vzájomné striedanie fáz *A* a *B* (multiplex) nie je ľudským zrakom rozpoznateľné. Naopak, striedanie normálneho a alternatívneho zobrazenia je vďaka nižšej frekvencii už viditeľné.



Obrázok 18: Časová postupnosť riadenia fáz

Tabuľka 4: Možnosti ovládania stavu segmentov

Zobrazenie	Alternatívne		Normálne		Popis
	Fáza	<i>F</i>	<i>E</i>	<i>B</i>	
Bit v registri LCDWF0	<i>BPFLCD0</i>	<i>BPELCD0</i>	<i>BPBLCD0</i>	<i>BPALCD0</i>	
Priradenie k backplane	<i>COM2</i>	<i>COM1</i>	<i>COM2</i>	<i>COM1</i>	
	0	0	0	0	Nezobrazený žiadny segment
	1	1	1	1	Trvale zobrazené oba segmenty (<i>MO</i> aj <i>TI</i>)
	0	1	0	1	Trvale zobrazený iba <i>MO</i>
	1	0	1	0	Trvale zobrazený iba <i>TI</i>
	0	0	1	1	Blikajú oba segmenty zároveň
	1	1	0	0	
	0	0	0	1	Bliká iba <i>MO</i>
	0	1	0	0	
	0	0	1	0	Bliká iba <i>TI</i>
	1	0	0	0	
	1	0	0	1	Segmenty blikajú v protifáze

4 KNIŽNICA NA OVLÁDANIE DISPLEJA GD-5360P

LCD displeje majú vo všeobecnosti graficky prezentovať užívateľovi dáta z mikrokontroléra. Sú nápomocné aj pri práci programátora, pre rýchly výstup informácií z mikrokontroléra už počas vývoja softvéru.

4.1 Návrh funkcií knižnice

Cieľom je vytvoriť knižnicu v jazyku C, ktorá umožní ovládať zadaný LCD displej GD-5360P. Musí byť schopná inicializovať hodinový signál určený pre radič LCD displeja, inicializovať samotný radič a ovládať všetky segmenty displeja (obrázok 2), vrátane výpisu písmen a čísiel na sedemsegmentové pozície. Písmená a čísla budú v knižnici preddefinované a bude ich možno vypísať aj ako reťazec znakov. Užívateľovi bude poskytnutá možnosť si v knižnici zdefinovať svoje vlastné znaky.

Pri programovaní mikrokontroléra sa často pracuje s premennými predstavujúcimi čísla. Preto musí knižnica obsahovať funkciu pre výpis číselných premenných na LCD displej. Pre jasnosť zobrazovaného čísla bude najvhodnejšie zobrazovať čísla na prvé tri sedemsegmentové pozície, spolu s využitím segmentu *T1* (znak jednotky). Bude teda možné zobrazovať číslo v rozsahu od -99 do 1 999.

Na pozície displeja 4 a 5 bude možné taktiež zapísať číselnú premennú (v rozsahu od -9 do 99).

Mikrokontrolér MC9S08LH64 obsahuje však aj 16-bitový A/D prevodník. Pre výpis 16-bitového výsledku prevodu v človeku najbližšej (dekadickej) sústave je potrebné zobrazovať číslo od 0 do 65 535. Preto bude potrebná aj funkcia pre výpis väčších čísel ako 1 999. Využiť sa na to dá všetkých päť sedemsegmentových pozícií spolu so segmentom *T1*. Bude schopná na displej vypísať číselnú premennú v rozsahu od -9 999 do 199 999. Zobrazenie však nebude najjasnejšie a najefektnejšie ako je tomu napríklad na znakových displejoch.

Predchádzajúce tri navrhnuté možnosti zobrazovania čísel budú umožňovať aj výpis v hexadecimálnom tvare, obmedzené len na nezáporné čísla.

Obsluha rôznych ďalších segmentov je taktiež dôležitá (napríklad dvojbodka oddeľujúca pozície pre zobrazenie hodín a minút a ďalšie...). Pre všetky budú vytvorené makrá, umožňujúce ich trvalé zobrazenie, blikanie a zakázanie ich zobrazenia. Dostupná bude taktiež funkcia pre obsluhu skupiny segmentov zobrazujúcich stav nabitia batérie a pre obsluhu segmentov anglických skratiek dní v týždni.

Súčasťou zadania je vytvoriť aj demonštračné laboratórne úlohy. Pre ne budú vytvorené špeciálne funkcie, ktoré študentom urýchlia ich programovanie.

4.2 Implementované funkcie

Funkcie sú implementované v súbore *LCD_GD-5360P.c*, pre používanie knižnice vo viacerých moduloch je vytvorený hlavičkový súbor *LCD_GD-5360P.h*. Oba súbory sa nachádzajú v prílohe na CD.

Funkcie nevracajú žiadne návratové hodnoty, úspešnosť ich vykonania závisí len na overení platnosti vstupných argumentov (platnosť rozsahu). Túto kontrolu si môže užívateľ knižnice vykonať sám a na jej základe vhodne vetviť program. Naprogramovanie návratových hodnôt funkciám generuje vo vývojovom prostredí CodeWarrior pri nekontrolovanej návratovej hodnote funkcie zbytočné a hlavne máťuce hlásenia o chybách. Pretože má navrhnutá knižnica slúžiť pri vyučovaní predmetu BMIC (Mikroprocesory), návratové hodnoty funkcií by boli pri vyučovacom procese nežiaduce.

Využívať funkcie pre ovládanie segmentov displeja je možné až po inicializácii hodinového signálu pre LCD radič a inicializácii LCD radiča.

Podrobnejší popis činnosti zložitejších funkcií je pre prehľadnosť uvedený priamo v knižničných súboroch, kde sú súvisiace časti kódu sústredené na jednom mieste.

4.2.1 Funkcia *Init_clock*

void Init_clock (void);

Funkcia *Init_clock* inicializuje hodinový signál pre LCD radič – nakonfiguruje modul *ICS* (Internal Clock Source) a modul *XOSCVLP* (External oscillator module). Frekvenciu hodinového signálu *OSCOUT* (hodinový signál pre radič LCD displeja) nastaví na 32,768 kHz. Funkcia musí byť volaná pred funkciou *Init_LCD* (inicializácia radiča LCD displeja).

4.2.2 Funkcia *Init_LCD*

void Init_LCD (void);

Funkcia *Init_LCD* inicializuje LCD radič pre zadaný displej GD-5360P. Pred jej volaním je nutné nakonfigurovať zdroj hodinového signálu pre radič LCD displeja pomocou funkcie *Init_clock*. Funkcia *Init_LCD* nastavuje:

- fyzické prepojenie LCD displeja a mikrokontroléra
- počet backplane vývodov displeja (2 spoločné vývody)
- rámcovú frekvenciu 32 Hz
- výber zdroja hodinového signálu pre radič (signál *OSCOUT*)
- zdroj vnútorného napätia V_{REG} na 0,91 V
- nábojovú pumpu pre generovanie riadiacich napätí pre LCD displej
- frekvenciu blikania špeciálnych segmentov na 1 Hz

4.2.3 Ovládanie špeciálnych segmentov

Pre sprehľadnenie programu sa pre ovládanie stavu špeciálnych segmentov používajú makrá. Pre každý špeciálny segment sú v hlavičkovom súbore knižnice nadefinované vždy štyri makrá. Jedno pre trvalé zobrazenie segmentu, dve pre ovládanie blikania a jedno pre vypnutie špeciálneho segmentu. Parametre makier závisia na fyzickom prepojení LCD displeja s mikrokontrolérom. Makrá pre ovládanie špeciálnych segmentov sú uvedené v tabuľke 5. Frekvencia blikania segmentov je nastavená na 1 Hz a dá sa zmeniť v inicializácii radiča LCD displeja.

Tabuľka 5: Makrá pre ovládanie špeciálnych segmentov

Segment	Názov makra				Popis segmentu
	Zapnutie segmentu	Blikanie segmentu	Blikanie v protifáze	Vypnutie segmentu	
<i>P0</i>	<i>P0_on</i>	<i>P0_blink_on</i>	<i>P0_blink_inv_on</i>	<i>P0_off</i>	Indikácia stavu nabitia batérie
<i>P1</i>	<i>P1_on</i>	<i>P1_blink_on</i>	<i>P1_blink_inv_on</i>	<i>P1_off</i>	
<i>P2</i>	<i>P2_on</i>	<i>P2_blink_on</i>	<i>P2_blink_inv_on</i>	<i>P2_off</i>	
<i>P3</i>	<i>P3_on</i>	<i>P3_blink_on</i>	<i>P3_blink_inv_on</i>	<i>P3_off</i>	
<i>P4</i>	<i>P4_on</i>	<i>P4_blink_on</i>	<i>P4_blink_inv_on</i>	<i>P4_off</i>	
<i>MO</i>	<i>MO_on</i>	<i>MO_blink_on</i>	<i>MO_blink_inv_on</i>	<i>MO_off</i>	Anglické skratky dní v týždni
<i>TU</i>	<i>TU_on</i>	<i>TU_blink_on</i>	<i>TU_blink_inv_on</i>	<i>TU_off</i>	
<i>WE</i>	<i>WE_on</i>	<i>WE_blink_on</i>	<i>WE_blink_inv_on</i>	<i>WE_off</i>	
<i>TH</i>	<i>TH_on</i>	<i>TH_blink_on</i>	<i>TH_blink_inv_on</i>	<i>TH_off</i>	
<i>FR</i>	<i>FR_on</i>	<i>FR_blink_on</i>	<i>FR_blink_inv_on</i>	<i>FR_off</i>	
<i>SA</i>	<i>SA_on</i>	<i>SA_blink_on</i>	<i>SA_blink_inv_on</i>	<i>SA_off</i>	
<i>SU</i>	<i>SU_on</i>	<i>SU_blink_on</i>	<i>SU_blink_inv_on</i>	<i>SU_off</i>	
<i>T1</i>	<i>T1_on</i>	<i>T1_blink_on</i>	<i>T1_blink_inv_on</i>	<i>T1_off</i>	Znak číslice 1
<i>COL</i>	<i>COL_on</i>	<i>COL_blink_on</i>	<i>COL_blink_inv_on</i>	<i>COL_off</i>	Dvojbodka
<i>AM</i>	<i>AM_on</i>	<i>AM_blink_on</i>	<i>AM_blink_inv_on</i>	<i>AM_off</i>	Ante meridiem
<i>PM</i>	<i>PM_on</i>	<i>PM_blink_on</i>	<i>PM_blink_inv_on</i>	<i>PM_off</i>	Post meridiem
<i>HEAT</i>	<i>HEAT_on</i>	<i>HEAT_blink_on</i>	<i>HEAT_blink_inv_on</i>	<i>HEAT_off</i>	Symbol plameňa
<i>COLD</i>	<i>COLD_on</i>	<i>COLD_blink_on</i>	<i>COLD_blink_inv_on</i>	<i>COLD_off</i>	Symbol vločky
<i>FLTR</i>	<i>FLTR_on</i>	<i>FLTR_blink_on</i>	<i>FLTR_blink_inv_on</i>	<i>FLTR_off</i>	Preddefinovaný text FLTR
<i>HOLD</i>	<i>HOLD_on</i>	<i>HOLD_blink_on</i>	<i>HOLD_blink_inv_on</i>	<i>HOLD_off</i>	Preddefinovaný text HOLD

Na nastavovanie a nulovanie niektorých bitov v registroch LCDWF sa používajú tieto pomocné makrá:

```
#define Set_bits(num,bit)    LCDWF##num |= (( 1 << (bit) ) | ( 1 << (bit+4) ) )
#define Set_bit(num,bit)    LCDWF##num |= ( 1 << (bit) )
```

```

#define Set_bit2(num,bit)  LCDWF##num |= ( 1 << (4+bit) )
#define Clear_bits(num,bit) LCDWF##num &= ~( ( 1 << (bit) ) | ( 1 << (bit+4) ) )
#define Clear_bit(num,bit) LCDWF##num &= ~( ( 1 << (bit) ) )
#define Clear_bit2(num,bit) LCDWF##num &= ~( ( 1 << (4+bit) ) )

```

Parameter *num* je číslo LCDWF registra, ktorý ovláda určitý segment. Parameter *bit* je poradové číslo bitu v nastavovanom registri LCDWF. Každý bit registra LCDWF nastavuje aktivitu segmentu len v určitej fáze riadenia displeja.

Príklad použitia pomocných makier pre segment *P0*:

Segment *P0* je pripojený k pinu mikrokontroléra č. 1, teda jeho stav sa ovláda v registri *LCDWF1*. Z druhej strany je tento segment pripojený k backplane vývodu *COM1*. Všetky segmenty pripojené ku *COM1* sa ovládajú v *A* a *E* fáze riadenia displeja.

Zapnutie segmentu *P0*

```
#define P0_on  Set_bits(1,0)
```

Pomocou makra *Set_bits(1,0)*; sa nastaví nultý bit (fáza *A*) a štvrtý bit (fáza *E*) registra *LCDWF1*, segment bude trvalo zobrazený.

Zapnutie blikania segmentu *P0*

```
#define P0_blink_on  Clear_bit2(1,0); Set_bit(1,0)
```

Aby segment blikal, musí byť aktívny len v normálnej alebo len v alternatívnej fáze riadenia displeja. Makro *Clear_bit2(1,0)*; vynuluje štvrtý bit registra *LCDWF1*. Tento bit nastavuje aktivitu segmentu v alternatívnej fáze (vo fáze *E*). Makro *Set_bit(1,0)*; naopak nastaví nultý bit rovnakého registra. Nultý bit nastavuje aktivitu segmentu počas normálnej fázy riadenia (fáza *A*). Segment *P0* bude blikat' nastavenou frekvenciou 1 Hz.

Zapnutie blikania segmentu *P0* v opačnej fáze

```
#define P0_blink_inv_on  Clear_bit(1,0); Set_bit2(1,0)
```

Makro *Clear_bit(1,0)*; vynuluje nultý bit registra *LCDWF1*. Tento bit nastavuje aktivitu segmentu v normálnej fáze (vo fáze *A*). Makro *Set_bit2(1,0)*; naopak nastaví štvrtý bit rovnakého registra. Štvrtý bit nastavuje aktivitu segmentu počas alternatívnej fázy riadenia (fáza *E*). Segment *P0* bude znova blikat' nastavenou frekvenciou 1 Hz, avšak v opačnej fáze ako v predchádzajúcom prípade.

Vypnutie segmentu *P0*

```
#define P0_off  Clear_bits(1,0)
```

Makro *Clear_bits(1,0)*; vynuluje nultý a štvrtý bit registra *LCDWF1*, segment *P0* nebude zobrazený.

4.2.4 Funkcia *All_seg_on*

void All_seg_on (void);

Funkcia zobrazí všetkých 55 segmentov LCD displeja GD-5360P zapísaním do všetkých frontplane *LCDWF* registrov hodnotu 0x33 – zobrazenie všetkých segmentov pripojených k obojstrannému vývodu displeja (počas normálnej aj alternatívnej fázy zobrazovania).

4.2.5 Funkcia *All_seg_off*

void All_seg_off (void);

Táto funkcia zapíše do všetkých frontplane registrov *LCDWF* hodnotu nula, čo spôsobí vypnutie všetkých 55 segmentov LCD displeja GD-5360P. Používa sa aj pri inicializácii LCD radiča, kedy je potrebné inicializovať stav frontplane *LCDWF* registrov po resete mikrokontroléra, pretože vtedy nemajú definovaný iniciálny stav.

4.2.6 Funkcia *Write_char*

void Write_char (char character, unsigned char position);

Funkcia *Write_char* zobrazí znak *character* na pozícii sedemsegmentového znaku s poradovým číslom *position*. Argument *position* môže byť celé číslo v rozsahu od 1 do 5. Pozícia je naznačená na obrázku 2. Rozsah platného argumentu pre zobrazovaný znak vychádza z ASCII tabuľky (od znaku pre medzeru po znak Z), definované v poli *table[]* (súbor *LCD_GD-5360P.c*). V prípade zadania nedefinovaného znaku alebo pozície väčšej ako 5, funkcia nevykoná žiadny výpis. Funkcia avšak akceptuje aj písmená malej abecedy, ale kvôli nedostatku segmentov sa nedá rozlišovať výpis malého alebo veľkého písmena abecedy.

Pri zadaní znaku z povoleného rozsahu, ktorý sa z dôvodu nedostatku segmentov nedá na displeji zobrazit', funkcia vypíše medzeru. Jedná sa o tieto znaky:

*! # \$ % & () * + , . / : ; < = > ? @*

Funkcia dokáže zobrazit' nasledujúce znaky:

*medzera " ' - 0 1 2 3 4 5 6 7 8 9 A B C D E F
G H I J K L M N O P Q R S T U V W X Y Z*

4.2.7 Funkcia *Write_spec_char*

void Write_spec_char (unsigned char char_code, unsigned char position);

Funkcia *Write_spec_char* zobrazí špeciálny znak, nedefinovaný v poli *table_special[]* (súbor *LCD_GD-5360P.c*), na pozícii sedemsegmentového znaku

s poradovým číslom *position*. Význam argumentu *position* je rovnaký ako vo funkcii *Write_char*. Vstupný argument *char_code* je poradové číslo symbolu v poli *table_special[]* začínajúc od nuly. Na tomto mieste je nadefinovaný určitý špeciálny znak. Tieto znaky môže užívateľ upravovať, prípadne si doplniť vlastné, musí však taktiež upraviť hodnotu *spec_char_max* v hlavičkovom súbore knižnice *LCD_GD-5360P.h*, na celkový počet definovaných znakov v tomto poli. Pri definovaní vlastných špeciálnych znakov je potrebné v poli *table_special[]* doplniť sedembitové číslo, každý bit predstavuje jeden zo segmentov sedemsegmentového znaku (hodnota bitu 1 daný segment zobrazuje).

4.2.8 Funkcia *Write_string*

*void Write_string (char *str, unsigned char init_position);*

Funkcia *Write_string* vypisuje reťazec znakov *str* od začiatkovej pozície *init_position*. Pre argument *init_position* platia rovnaké pravidlá ako pre argument *position* pri funkciách *Write_char* alebo *Write_spec_char*. Pri pokuse zapísať reťazec s väčším počtom znakov ako 5 sa vypisuje samozrejme len po pozíciu 5.

4.2.9 Funkcia *Num_small*

void Num_small (int num);

Táto funkcia je určená pre zobrazenie čísla *num* v rozsahu od -99 do 1 999. Pokiaľ je číslo v povolenom rozsahu, zobrazí sa na prvé tri sedemsegmentové pozície s využitím segmentu *T1* (znak jednotky). Ak je zároveň menšie ako 100, nezobrazia sa pred ním nuly, ale medzery.

4.2.10 Funkcia *Num_small_hex*

void Num_small_hex (unsigned int num);

Táto funkcia je určená pre zobrazenie čísla *num* v hexadecimálnom tvare, v rozsahu od 0 do 0x1FFF. Pokiaľ je číslo v povolenom rozsahu, zobrazí sa na prvé tri sedemsegmentové pozície s využitím segmentu *T1* (znak jednotky). Ak je zároveň menšie ako 0x100, nezobrazia sa pred ním nuly, ale medzery.

4.2.11 Funkcia *Num_big*

void Num_big (long int num);

Táto funkcia je určená pre zobrazenie čísla *num* v rozsahu od -9 999 do 199 999. Pokiaľ je číslo v povolenom rozsahu, zobrazí sa na všetkých 5 sedemsegmentových

pozícií s využitím segmentu *TI* (znak jednotky). Ak je zároveň menšie ako 10 000, nezobrazia sa pred ním nuly, ale medzery.

4.2.12 Funkcia *Num_big_hex*

void Num_big_hex (unsigned long int num);

Táto funkcia je určená pre zobrazenie čísla *num* v hexadecimálnom tvare, v rozsahu od 0 do 0x1FFFFFF. Pokiaľ je číslo v povolenom rozsahu, zobrazí sa na všetkých 5 sedemsegmentových pozícií s využitím segmentu *TI* (znak jednotky). Ak je zároveň menšie ako 0x10000, nezobrazia sa pred ním nuly, ale medzery.

4.2.13 Funkcia *Num_aux*

void Num_aux (int num);

Táto funkcia je určená pre zobrazenie čísla *num* v rozsahu od -9 do 99. Pokiaľ je číslo v povolenom rozsahu, zobrazí sa na posledné dve sedemsegmentové pozície. Ak je zároveň menšie ako 10, nezobrazí sa pred ním nula, ale medzera.

4.2.14 Funkcia *Num_aux_hex*

void Num_aux_hex (unsigned int num);

Táto funkcia je určená pre zobrazenie čísla *num* v hexadecimálnom tvare, v rozsahu od 0 do 0xFF. Pokiaľ je číslo v povolenom rozsahu, zobrazí sa na posledné dve sedemsegmentové pozície. Ak je zároveň menšie ako 0x10, nezobrazí sa pred ním nula, ale medzera.

4.2.15 Funkcia *Clr_week_day*

void Clr_week_day (void);

Funkcia zmaže segmenty zobrazujúce anglické skratky dní v týždni.

4.2.16 Funkcia *Set_week_day*

void Set_week_day (enum days day);

Funkcia zobrazí vždy jediný segment dňa v týždni. Vstupným parametrom je položka z preddefinovaného typu enum *days*. Zadeinované sú tieto položky:

MO, TU, WE, TH, FR, SA, SU

4.2.17 Funkcia *Battery*

void Battery (unsigned char ctrl);

Funkcia obsluhuje zobrazovanie stavu nabitia batérie. Vstupným parametrom *ctrl* je číslo od 0 do 5, ovláda zobrazovanú úroveň stavu nabitia. Zadaním nuly sa nezobrazí žiadny segment tejto skupiny segmentov, zadaním čísla päť sa zobrazia všetky segmenty v tejto skupine, predstavujúce plné nabitie.

4.2.18 Funkcia *Time_HM*

void Time_HM (unsigned int hours, unsigned int minutes);

Funkcia zobrazuje na displeji hodiny v 12-hodinovom formáte. Vstupný argument *hours* je počet hodín od 0 do 23, argument *minutes* počet minút od 0 do 59. Z počtu hodín vhodne prepína segmenty *AM* a *PM*. Dvojbodka oddeľujúca hodiny od minút bliká nastavenou frekvenciou určenou pri inicializácii radiča LCD displeja v registri *LCDBCTL* (1 Hz). Pri zadaní hodnôt mimo rozsah sa výpis nevykoná.

4.2.19 Funkcia *Time_HMS*

void Time_HMS (unsigned int hours, unsigned int minutes, unsigned int seconds);

Táto funkcia je určená pre jednu z vytvorených laboratórnych úloh. Správa sa podobne ako funkcia *Time_HM*, zobrazuje navyše aj sekundy. Pri zadaní hodnôt mimo rozsah sa výpis nevykoná.

4.2.20 Funkcia *Stopwatch*

void Stopwatch (unsigned int minutes, unsigned int seconds, unsigned int ticks);

Táto funkcia je určená pre jednu z vytvorených laboratórnych úloh, zobrazuje minúty, sekundy a stotiny sekúnd. Vstupný argument *minutes* je počet minút od 0 do 19, argument *seconds* počet sekúnd od 0 do 59 a argument *ticks* počet stotín sekundy od 0 do 99. Pri zadaní hodnôt mimo rozsah sa výpis nevykoná. Dvojbodka oddeľujúca minúty od sekúnd bliká nastavenou frekvenciou určenou pri inicializácii radiča LCD displeja v registri *LCDBCTL* (1 Hz).

5 LABORATÓRNE ÚLOHY

Táto kapitola sa venuje demonštračným laboratórnym úlohám, pri ktorých sa využije vytvorená knižnica. Určené sú pre praktické vyučovanie v kurze BMIC (Mikroprocesory). V tomto kurze prebieha vyučovanie na už spomínanej vývojovej doske TWR-S08LH64. Do tohto kurzu vstupujú študenti s rôznymi znalosťami mikroprocesorovej techniky, preto budú navrhované laboratórne úlohy orientované na prácu so základnými a jednoduchými perifériami a modulmi mikrokontroléra MC9S08LH64 a samozrejme na prácu s LCD displejom. Nutná je ale znalosť základov programovacieho jazyka C (z predchádzajúceho kurzu BPC2). Laboratórne cvičenie v predmete BMIC je dotované tromi vyučovacími hodinami (150 minút), preto budú zadania koncipované na takú úroveň, aby vypracovanie trvalo približne 120 až 140 minút.

Hlavné časti návodu na laboratórnu úlohu budú zadanie, teoretický úvod, doporučený postup vypracovania a záver úlohy.

V úlohách sa využívajú aj prerušenia. V nich sa vykonávajú väčšinou všetky výpočty a výpisy na LCD displej. Toto riešenie nie je optimálne, obsluha prerušenia by mala trvať vždy čo najkratšie, vykonať len nevyhnutné operácie, nastaviť potrebné príznaky a vrátiť beh programu do hlavnej slučky, v ktorej sa vykonávajú časovo náročnejšie výpočty a výpisy. Pre vyučovanie, názornosť a prípadné hľadanie chyby v programe je však výhodnejšie použiť takúto, z iného hľadiska neoptimálnu, možnosť.

Úlohy sú koncipované na zoznámenie sa a vyskúšanie si práce s jednotlivými perifériami a modulmi samostatne a až nakoniec skĺbenie viacerých podprogramov do jedného funkčného celku, hlavne z dôvodu rýchlejšej identifikácie prípadnej chyby v programe. Vypracovanie jednej z úloh nie je podmienkou pre vypracovanie druhej, sú však zoradené podľa náročnosti.

Návod k druhej laboratórnej úlohe obsahuje niektoré zhodné časti textu z prvej úlohy. V tomto texte sa ale zbytočne neopakujú. Samostatné a kompletne zadania sa nachádzajú v elektronickej podobe na CD ako príloha.

Zadania vytvorených úloh neobsahujú vývojové diagramy. Na laboratórnej úlohe je najdôležitejšie naučiť sa, resp. precvičiť si vytváranie vlastného vývojového diagramu, ktorému jeho autor dokáže porozumieť, samozrejme, najlepšie. Študent nemusí študovať viaceré špecifikácie, potrebné údaje sú vždy uvedené v teoretickom úvode úlohy, preto má na cvičení dostatok času na vymyslenie svojho vývojového diagramu. Programovanie už v návode vypracovaného vývojového diagramu by sa nedalo považovať za dostatočnú aktivitu počas laboratórneho cvičenia. Išlo by len o neprínosný preklad algoritmu z návodu do vývojového prostredia.

Zdrojové kódy naprogramovaných laboratórných úloh sa nachádzajú v prílohe na CD.

5.1 Laboratórna úloha: Zobrazenie hodín na LCD displeji

V tejto úlohe si študent vyskúša konfiguráciu modulov TOD (Time of Day) a KBI (Keyboard Interrupt) s využitím obslužných rutín prerušení v jazyku C.

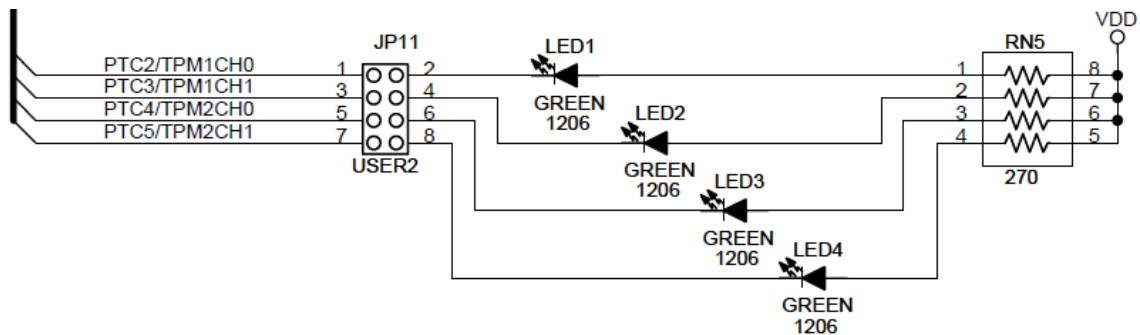
5.1.1 Zadanie úlohy

Zoznámte sa s knižnicou pre LCD displej, hlavne s funkciou *Time_HMS*, ktorá umožňuje výpis času na LCD displej. Pomocou nej a modulu TOD, ktorý poskytuje sekundové prerušenie, napíšte v programovacom jazyku C program, ktorý bude na displej vypisovať čas – hodiny, minúty a sekundy. S využitím modulu KBI doprogramujte možnosť dostaviť čas pomocou tlačidiel SW1 až SW3. Môžete využiť doporučený postup pri programovaní jednotlivých modulov spolu s kontrolou ich funkčnosti.

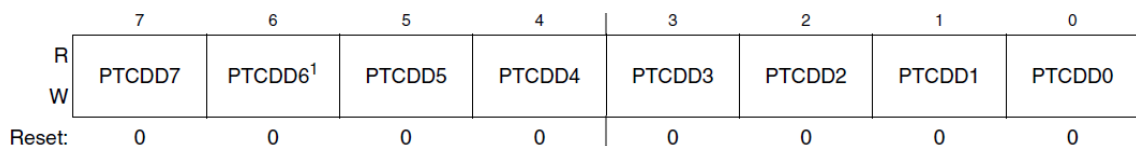
5.1.2 Teoretický úvod

5.1.2.1 Ovládanie LED diód

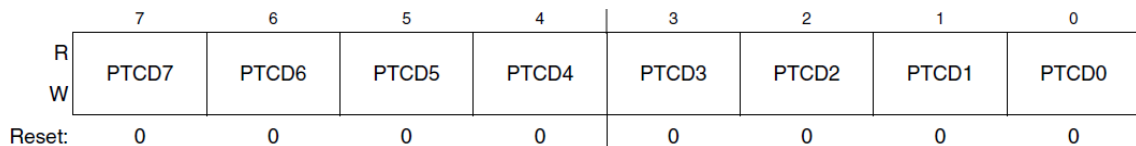
Pri testovaní programu sa pre indikáciu často používajú LED diódy. Anódami sú cez sieť obmedzovacích rezistorov pripojené na napätie VDD – úroveň log. 1. Pre ovládanie LED diód je potrebné nastaviť príslušné piny *PTCx* portu C do funkcie výstupu. Nastavenie pinu do funkcie výstupu sa prevedie zápisom log. 1 do bitov *PTCDDn* v registri *PTCDD*. Rozsvietenie/zhasnutie diód sa ovláda zápisom log. 0/log. 1 na príslušné piny *PTCx* portu C (negatívna logika). Dáta sa zapisujú do bitov *PTCDn* v registri *PTCD*.



Obrázok 19: Prepojenie LED diód s mikrokontrolérom [6]



Obrázok 20: Register PTCDD – Port C Data Direction [3]



Obrázok 21: Register PTCD – Port C Data Register [3]

5.1.2.2 Práca s LCD displejom GD-5360P

Do založeného projektu je nutné vložiť knižnicu na ovládanie LCD displeja. V hlavnom programe je nutné ju pripojiť pomocou `#include "LCD_GD-5360P.h"` a taktiež je potrebné inicializovať hodinový signál pre radič pomocou funkcie `Init_clock` a samotný radič pre LCD displej pomocou funkcie `Init_LCD`. Po tejto inicializácii je možné plne ovládať displej pomocou zadaných funkcií. Na otestovanie úspešnej inicializácie je možné použiť funkciu `All_seg_on`, ktorá zobrazí všetky segmenty LCD displeja GD-5360P.

Funkcia `Time_HMS` je určená pre túto laboratórnu úlohu, zobrazuje na LCD displeji hodiny, minúty a sekundy. Zároveň sa zobrazí blikajúca dvojbodka oddeľujúca hodiny od minút.

Predpisy funkcií:

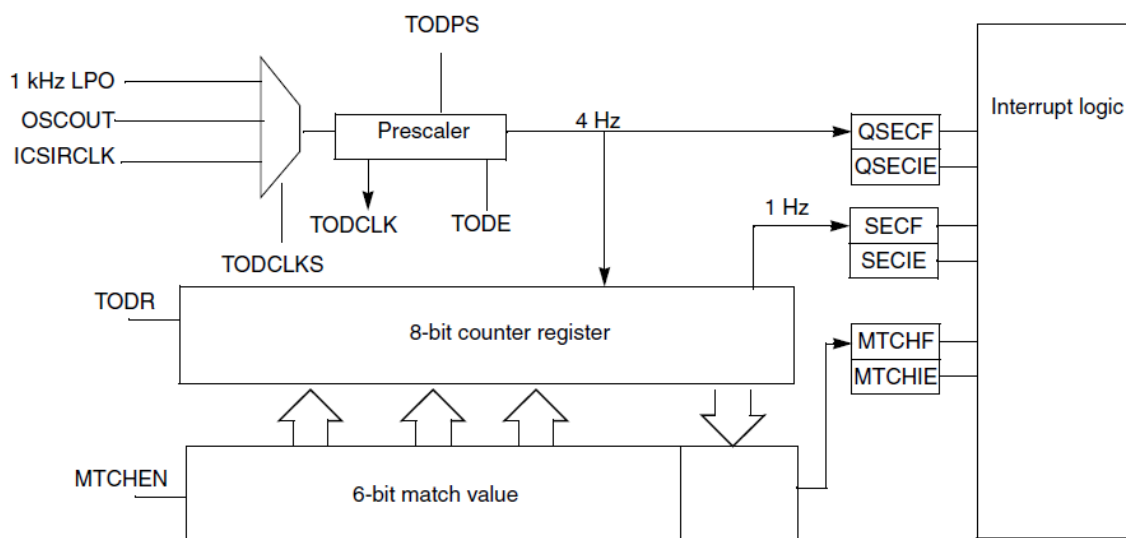
`void Init_clock (void);`

`void Init_LCD (void);`

`void All_seg_on (void);`

`void Time_HMS (unsigned int hours, unsigned int minutes, unsigned int seconds);`

5.1.2.3 Modul TOD (Time of Day)

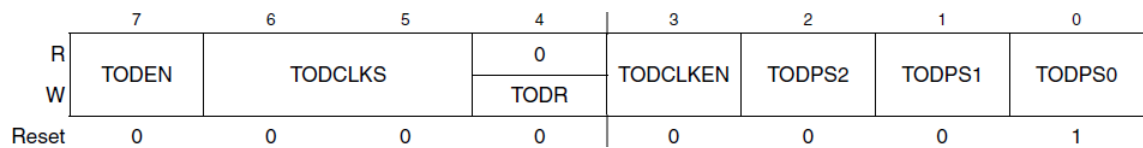


Obrázok 22: Bloková schéma modulu TOD [3]

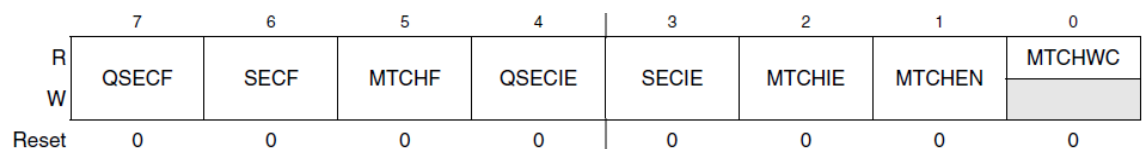
Tento modul dokáže generovať prerušenia v intervaloch 0,25 s, 1 s alebo pri zhode 8-bitového čítača TODCNT s registrom TODM (Match register). Pre túto úlohu bude popísané použitie sekundového prerušenia. Modul TOD sa ovláda viacerými registrami, pre generovanie sekundového prerušenia sú dôležité len registre *TODC* a *TODSC*.

Postup inicializácie modulu TOD

1. Zakázať činnosť modulu *TOD* vynulovaním bitu *TODEN* v registri *TODC* (po resete je už zakázaný). Hodnoty *TODCLKS* a *TODPS* sa musia nastavovať pri neaktívnom module *TOD*.
2. Ako zdroj hodinového signálu pre modul *TOD* zvoliť 1kHz signál z LPO (Low Power Oscillator) – zápisom 0b01 do bitov *TODCLKS* v registri *TODC*.
3. Nastaviť k tomuto signálu vhodnú deličku kmitočtu – zápisom 0b000 do bitov *TODPS[2:0]* v registri *TODC*.
4. Nastaviť generovanie sekundového prerušenia – zápisom log. 1 do bitu *SECIE* v registri *TODSC*.
5. Povolenie činnosti modulu *TOD* – zápisom log. 1 do bitu *TODEN* v registri *TODC*.



Obrázok 23: Register TODC – TOD Control Register [3]



Obrázok 24: Register TODSC – TOD Status and Control Register [3]

Po spustení programu sa každú sekundu vyvolá prerušenie, ktorého obsluhu je nutné naprogramovať do nasledujúceho bloku:

```

interrupt 19 ISR_TOD ( ) {
    // obsluha prerušenia modulu TOD
    // zmazanie príznaku SECF
}

```

Obsluha prerušenia musí zmazať príznak *SECF* v registri *TODSC* a tým povoliť vykonanie nasledujúceho prerušenia. Príznak sa zmaže zápisom log. 1 do bitu *SECF* v registri *TODSC*.

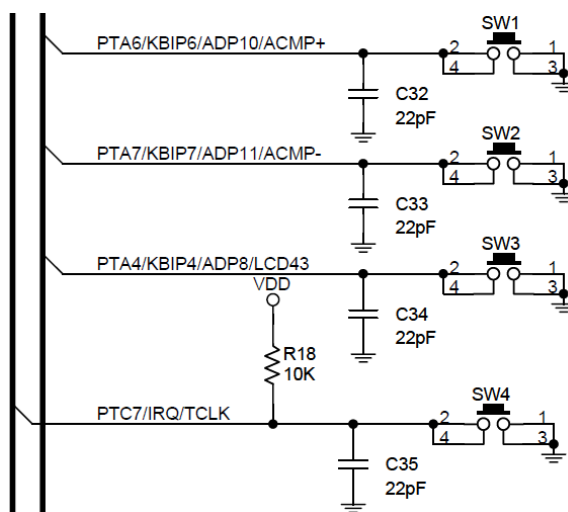
Činnosť modulu TOD sa dá kedykoľvek zakázať zmazaním bitu *TODEN* v registri *TODC*, pri opätovnom povolení jeho činnosti tento časovač začne odznova generovať sekundové prerušenie (8-bitový čítač štvrt'sekundových impulzov sa automaticky nastaví na nulu).

Presnosť sekundového prerušenia sa odvíja od presnosti použitého hodinového signálu. Vnútorňý 1kHz oscilátor nepatrí medzi presné zdroje kmitočtu. Pre orientačné meranie času je však jeho presnosť dostatočná.

5.1.2.4 Modul Keyboard Interrupt (KBI)

Tento modul dokáže vyvolať prerušenie pri stlačení jedného z tlačidiel SW1 až SW3 (práve tie sú totiž pripojené k pinom *KBIP_n*, ktoré sú kontrolované modulom KBI). Číslovanie pinov modulu *KBIP_n* je zhodné s číslovaním bitov v registroch *KBIES* a *KBIPE*, napríklad funkcie pinu *KBIP6* sa nastavujú bitom *KBEDG6* a *KBIPE6*. Obdobné platí aj pre register *PTAPE*.

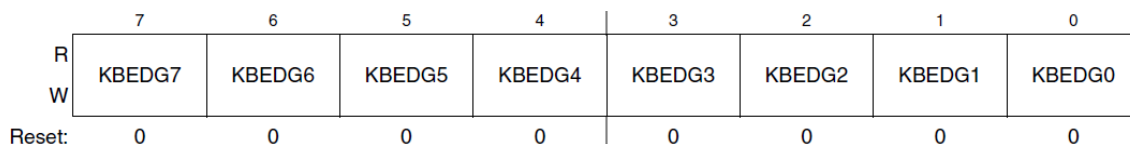
Tlačidlá pri stlačení pripájajú piny mikrokontroléra k napätiu 0V, čo predstavuje úroveň log. 0. Preto je potrebné nastaviť, aby modul KBI generoval prerušenie pri padajúcej hrane napätia na týchto pinoch. Aby tento modul zaznamenal padajúcu hranu, musí byť v kľudovom stave na pinoch úroveň log. 1 – treba nastaviť tzv. pull-up (pripojenie pinov mikrokontroléra k úrovni zodpovedajúcej log. 1 cez tzv. pull-up rezistory).



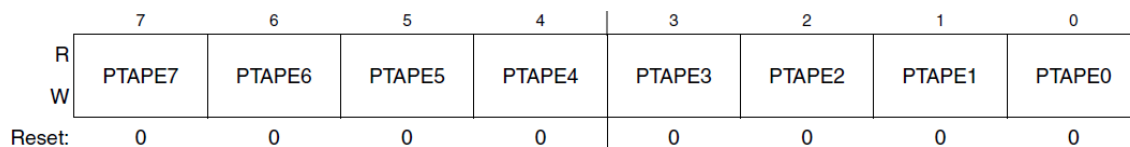
Obrázok 25: Pripojenie tlačidiel k pinom portu A [6]

	7	6	5	4	3	2	1	0
R	0	0	0	0	KBF	0	KBIE	KBIMOD
W						KBACK		
Reset:	0	0	0	0	0	0	0	0

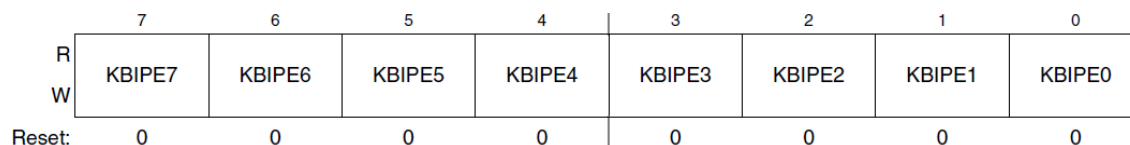
Obrázok 26: Register KBISC – KBI Interrupt Status and Control Register [3]



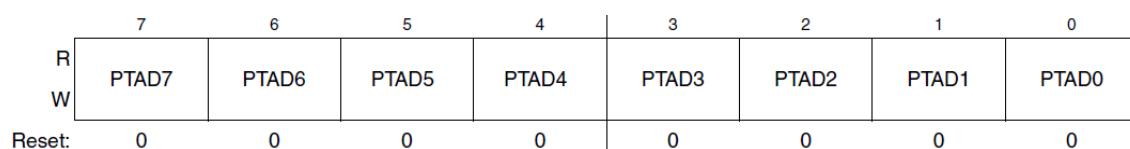
Obrázok 27: Register KBIES – KBI Interrupt Edge Select Register [3]



Obrázok 28: Register PTAPE – Port A Pull Enable Register [3]



Obrázok 29: Register KBIPE – KBI Interrupt Pin Select Register [3]



Obrázok 30: Register PTAD – Port A Data Register [3]

Postup inicializácie modulu KBI

1. Zakázať činnosť modulu KBI – vynulovať bit *KBIE* v registri *KBISC*
2. Zvoliť citlivosť pinov KBI modulu na padajúcu hranu vynulovaním potrebných bitov *KBEDG_n* v registri *KBIES* (podľa schémy pre zapojenie tlačidiel SW1 až SW3 na obrázku 25)
3. Nastaviť pull-up na pinoch pripojených k tlačidlám SW1 až SW3 v registri *PTAPE* (podľa schémy na obrázku 25) – zápis log. 1 povoľuje pull-up
4. Povoľiť prerušenie od pinov, ku ktorým sú pripojené tlačidlá – log. 1 v bitoch *KBIPE_n* v registri *KBIPE*
5. Vymazať možný nepravdivý príznak prerušenia, ktorý mohol vzniknúť počas nastavovania predchádzajúcich registrov – zapísať log. 1 do bitu *KBACK* v registri *KBISC*
6. Povoľiť činnosť modulu KBI – zapísať log. 1 do bitu *KBIE* v registri *KBISC*

Pri stlačení tlačidla sa vyvolá prerušenie, ktorého obsluhu je nutné naprogramovať do nasledujúceho bloku:

```

interrupt 16 ISR_KBI ( ) {
    // obsluha prerušenia modulu KBI
    // zmazanie príznaku KBF
}

```

Obsluha prerušenia musí zmazať príznak *KBF* v registri *KBISC* a tak umožniť obslúžiť nasledujúce prerušenie. Príznak sa zmaže zápisom log. 1 do bitu *KBACK* v registri *KBISC*.

Rozlíšenie, ktoré tlačidlo vyvolalo prerušenie sa naprogramuje v obsluhu prerušenia od tohto modulu prečítaním hodnoty na pinoch, ku ktorým sú tlačidlá pripojené. Tieto piny portu A musia byť nastavené ako vstupné – pomocou log. 0 v bitoch registra *PTADD*. Po resete sú piny portu A prednastavené do funkcie vstupu. Prerušenie sa po stlačení vyvolá takmer okamžite, tlačidlo je počas obsluhy prerušenia ešte stále stlačené, hodnota log. 0 v príslušných bitoch registra *PTAD* určuje, ktoré tlačidlo je stlačené.

5.1.3 Doporučený postup pri programovaní zadania

1. Napíšte program, v ktorom otestujte funkčnosť knižnice pre LCD displej. Môžete použiť napríklad funkciu *All_seg_on* (alebo ktorúkoľvek inú). Vyskúšajte si výpis s použitím funkcie *Time_HMS*.
2. Napíšte program, ktorý využije prerušenie generujúce modul TOD (Time of Day). Pre jednoduchosť napríklad rozblikajte niektorú z LED diód.
3. Napíšte program, ktorý využije prerušenie generujúce modul KBI (Keyboard Interrupt). Napríklad jedným tlačidlom zapnite a iným tlačidlom vypnite niektorú z LED diód.
4. Ak sú predchádzajúce programy funkčné, využite ich pri naprogramovaní hlavnej úlohy – výpis hodín na LCD displej pomocou funkcie *Time_HMS*. Počítajte jednotlivé sekundové prerušenia a vhodne vyhodnocujte počet vypisovaných hodín, minút a sekúnd. Naprogramujte prechod medzi 59 sekundami a minútami, medzi 59 minútami a hodinami a nakoniec prechod medzi 11PM a 0AM. Doprogramujte obsluhu tlačidiel vo funkcii inkrementovania hodnoty hodín, minút a sekúnd (jedno tlačidlo ovláda hodiny, druhé minúty a tretie sekundy). Zabezpečte, aby sa pri zmene času tlačidlom údaj aktualizoval čo najrýchlejšie (vypisujte ihneď pri zmene časových údajov – v oboch prerušeníach) a aby pri nastavovaní času tlačidlami nebol modul TOD aktívny.

5.1.4 Vzor štruktúry programu

```

#include <hidef.h>
#include "derivative.h"
#include "LCD_GD-5360P.h"

```

```

interrupt 19 ISR_TOD ( ) { } // obsluha prerušenia od modulu TOD
interrupt 16 ISR_KBI ( ) { } // obsluha prerušenia od modulu KBI
void main (void) {
    EnableInterrupts;
    // inicializácia LCD displeja, modulov TOD a KBI, portu mikrokontroléra
    for(;;) {
        __RESET_WATCHDOG();
    }
}

```

5.1.5 Možné chyby

Popis niektorých chýb, ktoré môžu nastať:

1. Zákmitý signálu na tlačidlách pri stlačení môžu spôsobiť viacnásobné vyvolanie prerušenia, než je žiadané. Program, ktorý by ošetril tento náhodný jav, by musel byť komplikovanejší, preto toto riešenie zákmity neuvažuje.
2. Príznak prerušenia nebol v obsluhu prerušenia zmazaný. Vtedy program v tomto prerušení „uviazne“ a watchdog pravidelne resetuje program.

5.1.6 Záver úlohy

Výsledkom tejto laboratórnej úlohy má byť program, ktorý na LCD displeji zobrazuje časové údaje: hodiny, minúty a sekundy v 12-hodinovom móde – údaje sa dajú nastaviť tlačidlami SW1 až SW3. V programe majú byť ošetrené prechody pri pretečení sekúnd, minút a hodín. Časový údaj sa po zmene tlačidlom mení ihneď (nečaká sa na ďalšie sekundové prerušenie) a pri nastavovaní času tlačidlami sa nepripočítavajú sekundy. Pokiaľ program vytvorený študentom disponuje takýmito funkciami, môže sa zadanie považovať za splnené.

5.2 Laboratórna úloha: Stopky

V tejto úlohe si študent vyskúša konfiguráciu 16-bitového čítača TPM a modul KBI (Keyboard Interrupt) s využitím obslužných rutín prerušení v jazyku C.

5.2.1 Zadanie úlohy

Pomocou 16-bitového čítača v mikrokontroléri napíšte program, ktorý bude stopovať čas a zobrazí ho na displeji pomocou funkcie *Stopwatch*, určenou pre túto úlohu. Na LCD displeji zobrazte počet ubehnutých minút, sekúnd a stotín sekundy.

Využite modul KBI a naprogramujte jedno tlačidlo pre odštartovanie a pozastavenie merania času. Druhým tlačidlom stopky zastavte a resetujte časové údaje.

5.2.2 Teoretický úvod

5.2.2.1 Práca s LCD displejom GD-5360P

Do založeného projektu je nutné vložiť knižnicu na ovládanie LCD displeja. V hlavnom programe je nutné ju pripojiť pomocou `#include "LCD_GD-5360P.h"` a taktiež je potrebné inicializovať hodinový signál pre radič pomocou funkcie `Init_clock` a samotný radič pre LCD displej pomocou funkcie `Init_LCD`. Po tejto inicializácii je možné plne ovládať displej pomocou zadefinovaných funkcií. Na otestovanie úspešnej inicializácie je možné použiť funkciu `All_seg_on`, ktorá zobrazí všetky segmenty LCD displeja GD-5360P.

Funkcia `Stopwatch` zobrazuje na LCD displeji minúty, sekundy a stotiny sekundy. Zároveň sa zobrazí blikajúca dvojbodka oddeľujúca minúty od sekúnd.

Segment `HOLD` sa dá na LCD displeji rozblikať pomocou makra `HOLD_blink_on`; alebo `HOLD_blink_inv_on`;

Predpisy funkcií:

`void Init_clock (void);`

`void Init_LCD (void);`

`void All_seg_on (void);`

`void Stopwatch (unsigned int minutes, unsigned int seconds, unsigned int ticks);`

5.2.2.2 16-bitový čítač TPM1

Mikrokontrolér MC9S08LH64 disponuje 16-bitovým čítačom zvoliteľných hodinových impulzov spolu s nastaviteľnou deličkou týchto impulzov. Pri dosiahnutí vrcholu čítača dokáže generovať prerušenie, vrchol čítača je možné nastaviť v 8-bitových registroch `TPM1MODH` a `TPM1MODL` (tvoria 16-bitový register `TPM1MOD`).

Čítač sa spúšťa výberom hodinového signálu bitmi `CLKSB:CLKSA`. Hodnota registrov `TPM1MODH` a `TPM1MODL` je po resete nastavená na log. 0. Vrcholom čítača je vtedy maximum 16-bitového registra `TPM1MOD` (hodnota 65 535). Ak je v registri `TPM1MOD` nenulová hodnota, je vrcholom čítača práve táto hodnota. Čítač sa po jej dosiahnutí resetuje a počítá odznova, zároveň sa generuje prerušenie. Zastavenie počítania impulzov neovplyvňuje aktuálnu hodnotu čítača v registroch `TPM1CNTH` a `TPM1CNTL`. Zápis do týchto registrov čítač nuluje.

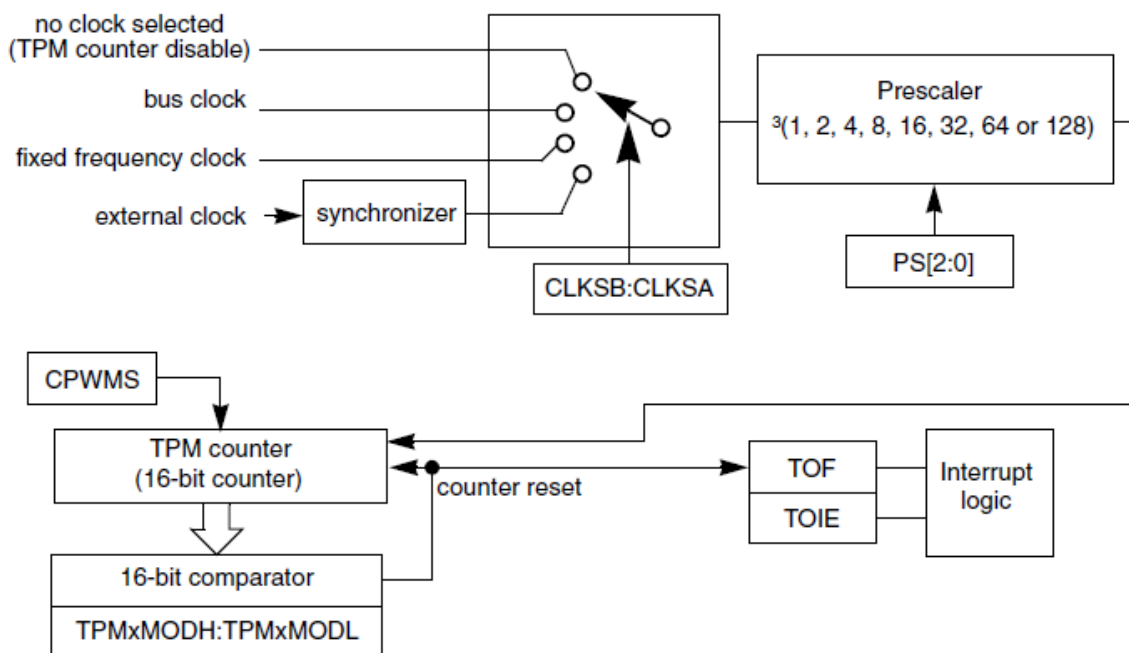
Frekvencia vstupného hodinového signálu do čítača musí byť pre presné nastavenie intervalov prerušenia známa. Najjednoduchšie je zvoliť za zdroj hodinového signálu pre čítač signál bus clock a predtým ho vhodne nastaviť. Pre nastavenie signálu

bus clock na 4,98 MHz použite postup uvedený ďalej v texte. Kvôli stručnosti nie je postup nastavenia vysvetlený podrobnejšie ako iné moduly.

Signál bus clock sa vyberá zápisom hodnoty 0b01 do bitov *CLKSB:CLKSA* v registri *TPMISC*. Zápis 0b00 do týchto bitov čítač zastaví, ale neresetuje dovtedy napočítané impulzy. Delička vstupného signálu pre čítač sa nastavuje bitmi *PS[2:0]* v registri *TPMISC* (viď tabuľka 6). Generovanie prerušenia pri dosiahnutí vrcholu čítača sa povoľuje zápisom log. 1 do bitu *TOIE* v registri *TPMISC*.

Tabuľka 6: Možnosti nastavenia deličky hodinového signálu pre čítač [3]

<i>PS[2:0]</i>	0b000	0b001	0b010	0b011	0b100	0b101	0b110	0b111
Deliaci faktor	1	2	4	8	16	32	64	128



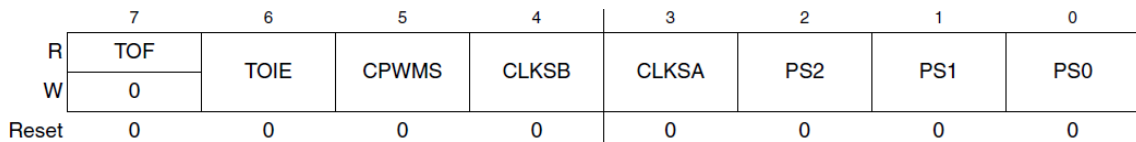
Obrázok 31: 16-bitový čítač (zjednodušená bloková schéma) [3]

Nastavenie modulu Internal Clock Source ($f_{\text{BUS CLOCK}} = 4,98 \text{ MHz}$):

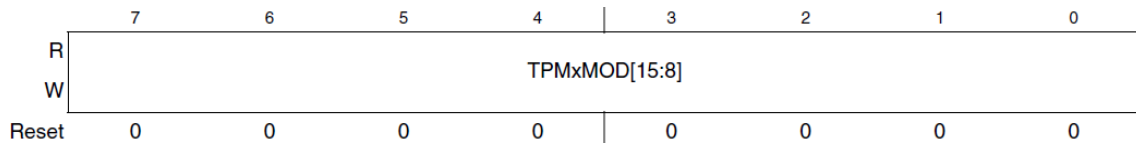
```

ICSC1_CLKS = 0b00;
ICSC1_RDIV = 0b000;
ICSC1_IREFS = 0;
ICSC2_BDIV = 0b01;
ICSC2_RANGE = 0;
ICSC2_EREFS = 1;
ICSSC_DM32 = 1;
ICSSC_IREFST = 0;

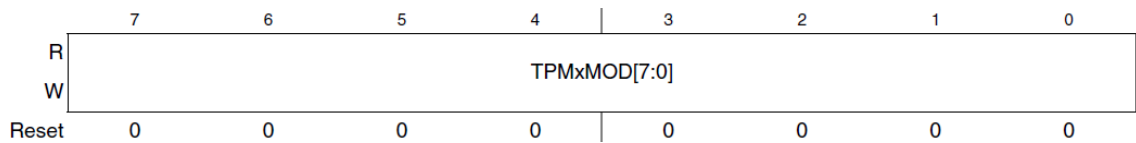
```



Obrázok 32: Register TPM1SC – TPM Status and Control Register [3]



Obrázok 33: Register TPM1MODH – TPM Counter Modulo Register High [3]



Obrázok 34: Register TPM1MODL – TPM Counter Modulo Register Low [3]

Ak je prerušenie povolené, tak sa pri dosiahnutí vrcholu čítača vyvolá prerušenie, jeho obsluhu je nutné umiestniť do nasledujúceho bloku:

```
interrupt 6 ISR_TPM1 ( ) {
    // obsluha prerušenia čítača TPM1
    // zmazanie príznaku TOF
}
```

Obsluha prerušenia musí zmazať príznak *TOF* v registri *TPM1SC* a tak umožniť obslúžiť nasledujúce prerušenie. Pre zmazanie je potrebné prečítať register *TPM1SC* a do bitu *TOF* zapísať log. 0.

5.2.3 Doporučený postup pri programovaní zadania

1. Napíšte program v ktorom otestujete funkčnosť knižnice pre LCD displej. Môžete použiť napríklad funkciu zobrazujúcu všetky segmenty LCD displeja – *All_seg_on* (alebo ktorúkoľvek inú). Vyskúšajte si výpis s použitím funkcie *Stopwatch*.
2. Napíšte program, ktorý využije prerušenie generujúce modul KBI (Keyboard Interrupt). Napríklad jedným tlačidlom zapnite a iným tlačidlom vypnite niektorú z LED diód.
3. Naprogramujte hlavnú úlohu – stopky.
Najprv inicializujte modul *ICS*. Zo známej frekvencie $f_{\text{BUS CLOCK}} = 4,98 \text{ MHz}$ určite vhodnú deličku signálu pre čítač a vhodnú hodnotu vrcholu čítača, pri ktorej bude sa bude generovať prerušenie každých $T=10 \text{ ms}$. Môžete použiť nasledujúci vzorec:

$$TPM1MOD = T * f_{BUS\ CLOCK} / deliaci\ faktor \quad [-; s, Hz, -]$$

Povoľte prerušenie pri dosiahnutí vrcholu čítača. Čítač vhodne spúšťajte a zastavujte, vyhodnocujte počet prerušení a prepočítavajte ich na stotiny sekundy, sekundy a minúty. Napíšte program tak, aby sa na LCD displeji po spustení programu zobrazil nulový stav stopiek. Jedným tlačidlom spustíte počítanie času, hodnoty normalizujete a zároveň vypisujete na LCD displej. Ak stopky bežia, opätovným stlačením tohto tlačidla stopky pozastavte. Na LCD displeji bude zobrazený aktuálny stav stopiek a bude blikať segment *HOLD*. Ďalším stlačením tohto tlačidla stopky spustíte a pokračujte v počítaní času. Iným tlačidlom stopky resetujte (zastavte počítanie času a vynulujte časové údaje). Ošetríte prechody medzi 99 stotinami sekundy a sekundami, medzi 59 sekundami a minútami a medzi 19 minútami a 0 minútami.

5.2.4 Vzor štruktúry programu

```
#include <hidef.h>
#include "derivative.h"
#include "LCD_GD-5360P.h"

interrupt 6 ISR_TPM1 ( ) { } // obsluha prerušenia od čítača
interrupt 16 ISR_KBI ( ) { } // obsluha prerušenia od modulu KBI
void main (void) {
    EnableInterrupts;
    // inicializácia displeja, modulu KBI a ICS, čítača
    for(;;) {
        __RESET_WATCHDOG();
    }
}
```

5.2.5 Záver úlohy

Výsledkom tejto laboratórnej úlohy sú stopky, po spustení programu sa zobrazujú nulové hodnoty minút sekúnd a stotín sekundy. Jedným tlačidlom sa beh stopiek zastavuje a znova spúšťa, iným tlačidlom sa stopky resetujú. V programe sú ošetrené prechody pri pretečení minút, sekúnd a stotín sekundy. Pokiaľ program vytvorený študentom disponuje takýmito funkciami, môže sa zadanie považovať za splnené.

6 ZÁVER

Bakalárska práca dokumentuje LCD displej GD-5360P, vývojovú dosku TWR-S08LH64 a mikrokontrolér MC9S08LH64. Sústreďuje o nich dôležité informácie z viacerých špecifikácií do jedného dokumentu. Tieto sú potrebné pre pochopenie ovládania konkrétneho LCD displeja.

Uvedené sú možnosti taktovania radiča rôznymi zdrojmi hodinových signálov a rôzne zdroje troch potrebných napäťových úrovní pre radič. Dôležitou časťou je výber najvhodnejšej možnosti taktovania a napájania LCD displeja. Najpodrobnejšia časť je venovaná vysvetleniu inicializácie zdroja hodinového signálu pre radič a inicializácie samotného radiča. Spôsob fyzického prepojenia LCD displeja a radiča spolu úzko súvisí a je dôležité jednak pre správnu inicializáciu radiča a aj pre ovládanie stavu jednotlivých segmentov LCD displeja.

Navrhnutá a implementovaná knižnica na ovládanie LCD displeja obsahuje funkcie, ktoré využívajú všetky zobrazovacie možnosti LCD displeja. Na displeji je možné vypisovať písmená veľkej abecedy a číslice na jednu z piatich pozícií, zapisovať reťazec znakov. Na výpis číselných premenných je k dispozícii 6 funkcií – pre výpis v troch rozsahoch v decimálnej a v hexadecimálnej sústave. Možný je taktiež výpis záporných čísiel.

Radič umožňuje pri správnom nastavení a ovládaní rozblikať žiadané segmenty displeja, o blikanie sa vtedy stará radič, blikanie nie je potrebné ošetrovať v programe ďalšími časovačmi. Každý špeciálny segment sa dá zapnúť, vypnúť a nastaviť na dva módy blikania.

Užívateľovi knižnice je poskytnutá možnosť jednoducho si dodefinovať vlastný znak a ten následne vypisovať.

Pre kurz BMIC sú navrhnuté dve laboratórne úlohy: Zobrazenie hodín na LCD displeji a Stopky. Obe využívajú knižnicu pre LCD displej. Knižnica obsahuje aj funkcie určené pre laboratórne úlohy. Každá z úloh obsahuje zadanie úlohy, teoretický úvod s popisom použitých modulov mikrokontroléra, doporučený postup pri programovaní, vzor štruktúry programu, popis možných chýb, ktoré môžu vzniknúť pri programovaní a záver úlohy. Záver úlohy slúži na kontrolu splnenia zadania konkrétnej laboratórnej úlohy. Kompletne zadania sú v prílohe na CD.

Literatúra

- [1] DOLEČEK, Jaroslav. *Moderní učebnice elektroniky*. Praha: BEN - technická literatura, 2005, 154 s. ISBN 80-730-0184-5.
- [2] MATOUŠEK, David. *Práce s inteligentními displeji LCD: [znakové a grafické displeje, přípravky a programy]*. 1 vyd. Praha: BEN - technická literatura, 2006, 222 s. ISBN 80-730-0121-7.
- [3] FREESCALE SEMICONDUCTOR: MC9S08LH64 Reference Manual, Rev. 5.1 [online]. 4/2010, updated 5/2012. [cit. 2013-10-26] Dostupné na URL: http://www.freescale.com/files/microcontrollers/doc/ref_manual/MC9S08LH64RM.pdf
- [4] AXIOM MANUFACTURING: TWR-S08 User Guide, Rev. F [online]. 9/2009 updated 4/2010. [cit. 2013-10-26] Dostupné na URL: http://www.freescale.com/files/microcontrollers/doc/user_guide/TWRS08LH64UG.pdf
- [5] FREESCALE SEMICONDUCTOR: TWR Module LCD Glass Specification, MC9S08LH64/36 8-bit LCD MCU Getting Started CD 2010 [CD]. Brno: VUT Brno, Fakulta elektrotechniky a komunikačních technologií, 2013.
- [6] FREESCALE SEMICONDUCTOR: TWR-MC9S08LL/LH AXM-0500, MC9S08LH64/36 8-bit LCD MCU Getting Started CD 2010 [CD]. Brno: VUT Brno, Fakulta elektrotechniky a komunikačních technologií

Zoznam príloh

Príloha 1. Obsah priloženého CD nosiča

Príloha 1. Obsah priloženého CD nosiča

a) Súbory knižnice pre LCD displej GD-5360P

V priečinku Knižnica LCD GD-5360P

LCD_GD-5360P.h

LCD_GD-5360P.c

b) Zadania dvoch laboratórnych úloh

V priečinku Zadania úloh

Hodiny.pdf

Stopky.pdf

c) Projekty vývojového prostredia CodeWarrior

(so vzorovo naprogramovanými laboratórnymi úlohami)

V priečinku Projekty

Priečínok Hodiny

Priečínok Stopky

Samotný zdrojový kód konkrétnej laboratórnej úlohy je v podpriečinku Sources v súbore *main.c*

d) Elektronická verzia BP

Súbor *Stibrany_Miroslav_2014_BP.pdf*