

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ
ÚSTAV BIOMEDICÍNSKÉHO INŽENÝRSTVÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION
DEPARTMENT OF BIOMEDICAL ENGINEERING

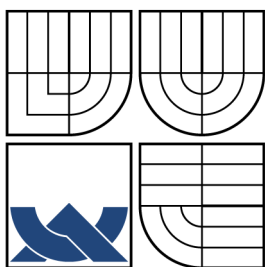
MAGNETOTERAPEUTICKÝ PŘÍSTROJ PRO LÉČBU
DĚTSKÉ MOZKOVÉ OBRNY

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

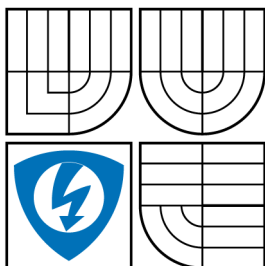
AUTOR PRÁCE
AUTHOR

PETR WALEK

BRNO 2010



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA ELEKTROTECHNIKY
A KOMUNIKAČNÍCH TECHNOLOGIÍ
ÚSTAV BIOMEDICÍNSKÉHO INŽENÝRSTVÍ



FACULTY OF ELECTRICAL ENGINEERING AND
COMMUNICATION
DEPARTMENT OF BIOMEDICAL ENGINEERING

MAGNETOTERAPEUTICKÝ PŘÍSTROJ PRO LÉČBU DĚTSKÉ MOZKOVÉ OBRNY

MAGNETOTHERAPEUTIC DEVICE FOR CHILD NEUROLOGY

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

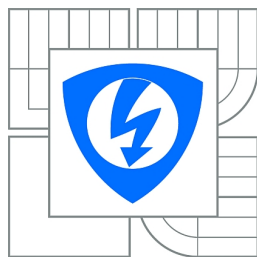
AUTOR PRÁCE
AUTHOR

PETR WALEK

VEDOUCÍ PRÁCE
SUPERVISOR

ING. JIŘÍ SEKORA

BRNO 2010



**VYSOKÉ UČENÍ
TECHNICKÉ V BRNĚ**

**Fakulta elektrotechniky
a komunikačních technologií**

Ústav biomedicínského inženýrství

Diplomová práce

magisterský navazující studijní obor
Biomedicínské a ekologické inženýrství

Student: Bc. Petr Walek

ID: 78286

Ročník: 2

Akademický rok: 2009/2010

NÁZEV TÉMATU:

Magnetoterapeutický přístroj pro léčbu dětské mozkové obrny

POKYNY PRO VYPRACOVÁNÍ:

Definujte parametry magnetického pole použitelného pro účely léčby dětské mozkové obrny. Na základě těchto parametrů navrhnete schéma zapojení řídicí a výkonové části magnetoterapeutického přístroje pro léčbu dětské mozkové obrny a sestavte programy pro použité mikrokontroléry. Při návrhu dbejte zásad pro konstrukci elektrického zařízení pro lékařské účely.

Práce musí obsahovat úplné elektrické schéma zapojení, rozpisku součástek, programy pro mikrokontroléry, výkres plošných spojů. Správná funkce obvodů bude ověřena simulací v PSpice.

DOPORUČENÁ LITERATURA:

[1] CHVOJKA, J. Magnetoterapie v teorii a praxi. Brno: Professional publishing, 2000. 99s. ISBN 80-86419-01-0

[2] HRNČÍŘ, K. O magnetoterapii [online]. Biomag. Dostupné z:
<<http://www.biomag.cz/magnetoterapie-o-magnetoterapii.php>>.

Termín zadání: 12.10.2009

Termín odevzdání: 21.5.2010

Vedoucí práce: Ing. Jiří Sekora

prof. Ing. Jiří Jan, CSc.
Předseda oborové rady

UPOZORNĚNÍ:

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

ABSTRAKT

Práce Magnetoterapeutický přístroj pro léčbu dětské mozkové obrny se ve své první části věnuje léčebnému využití pulzního magnetického pole. Shrnuje léčebné účinky, které má pole interagující s živou tkání a určuje, které z nich jsou vhodné pro léčbu dětské mozkové obrny. Dále se věnuje jednotlivým biotropním parametrům pulzního magnetického pole a rozebírá příčiny a důsledky dětské mozkové obrny. V druhé části je popsáno blokové schéma magnetoterapeutického přístroje pro léčbu dětské mozkové obrny. Při popisu blokového schématu je kladen důraz na vysvětlování principiální funkce přístroje. Ve třetí části je popsáno obvodové řešení řídicí, kontrolní a generující části přístroje a je detailně popsána funkce a zapojení jednotlivých součástek. K obvodovému řešení přístroje je vytvořena deska plošných spojů. Přístroj je řízen třemi mikrokontroléry, proto jsou v poslední části popsány vývojové diagramy řídicích programů i programy samotné.

KLÍČOVÁ SLOVA

Dětská mozková obrna, pulzní magnetické pole, biotropní parametry magnetického pole, přímá číslicová syntéza (DDS), solenoid, plochá spirálová cívka.

ABSTRACT

Project Magnetotherapeutic device for Child Neurology in his forepart paises medical use of Pulsatile Magnetic Field. Summaries a therapeutic effects of field which is interacting with biotic issue and determines which of those effects are acceptable for treatment of Infantile Paralysis. Below it describes single Bio-Parametres of Pulsatile Magnetic Field and analyses reasons and effects of Infantile Paralysis. In second part is described a block diagram of magnetotherapeutic device for treatment of Infantile Paralysis. Described is principle of function of blocks. In third part is defined circuit solving of controlling, supervisory and generating part of magnetotherapeutic device. Function and wiring of components is described in detail. There is also created a Printed Circuit Board. Device is controlled by three microcontrollers, therefore there are described flowcharts and controlling programs for microcontrollers.

KEYWORDS

Infantile Paralysis, Pulse Magnetotherapy, Bio-Parameters of Magnetic Field, Digital Direct Synthesis, Solenoid, Flat Spiral Coil.

WALEK P. *Magnetoterapeutický přístroj pro léčbu dětské mozkové obrny*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2010. 70 s., 18 s. příloh. Vedoucí diplomové práce Ing. Jiří Sekora.

PROHLÁŠENÍ

Prohlašuji, že svou diplomovou práci na téma „Magnetoterapeutický přístroj pro léčbu dětské mozkové obrny“ jsem vypracoval samostatně pod vedením vedoucího diplomové práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

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

V Brně dne

.....

(podpis autora)

Poděkování

Děkuji vedoucímu diplomové práce Ing. Jiřímu Sekorovi za odborné rady při tvorbě diplomové práce, za ochotu a pomoc při finální úpravě práce i za trpělivost a čas, který mi věnoval.

V Brně dne 21.května 2010

.....

Podpis autora

OBSAH

Úvod	10
1 Interakce pulzního magnetického pole s živou tkání	13
1.1 Účinky pulzního magnetického pole na tkáň	14
1.2 Biotropní parametry magnetického pole	15
1.2.1 Indukce magnetického pole	16
1.2.2 Maximální gradient magnetického pole	16
1.2.3 Vektor	17
1.2.4 Frekvence	17
1.2.5 Tvar impulzu	17
1.2.6 Doba expozice	18
2 Dětská mozková obrna	19
2.1 Příčiny vzniku DMO	19
2.2 Typy DMO	20
2.3 Léčba DMO pomocí PMP	20
2.4 Možné kontraindikace	21
3 Blokové schéma přístroje pro léčbu DMO	22
3.1 Popis funkce řídicí, kontrolní a generující části	22
3.1.1 Přímá číslicová syntéza	23
3.1.2 Příklad nastavení parametrů terapie a analýza signálů	24
3.2 Popis funkce silové a aplikační části	28
3.2.1 Solenoidální aplikátor	29
3.2.2 Plochý aplikátor	31
4 Schéma zapojení řídicí, kontrolní a generující části	33
4.1 Popis řídicího a kontrolního MCU a zapojení jeho periférií	33
4.1.1 Napájení MCU, hodinový signál, resetovací obvod a externí přerušení	33
4.1.2 Maticová klávesnice a LCD displej	34
4.1.3 Komunikace pomocí sériové sběrnice I ² C	35
4.1.4 Obvod reálného času a teplotní čidlo	36
4.2 Popis generujících MCU a zapojení jejich periférií	38
4.2.1 Napájení MCU, hodinový signál a resetovací obvod	38
4.2.2 Digitálně/analogový převodník	39
4.2.3 Analogový filtr	40
4.3 Deska plošných spojů	41

5	Vývojové diagramy programů pro mikrokontroléry	44
5.1	Vývojový diagram pro řídicí a kontrolní mikrokontrolér	44
5.1.1	Popis hlavního programu	44
5.1.2	Popis podprogramů a obsluh přerušení	46
5.2	Vývojový diagram pro generující mikrokontroléry	48
5.2.1	Popis hlavního programu	48
5.2.2	Popis podprogramů	50
6	Programy pro mikrokontroléry	51
6.1	Programy pro řídicí MCU	51
6.1.1	Popis hlavního programu	51
6.1.2	Popis podprogramů a obsluh přerušení	55
6.2	Programy pro generující MCU	60
6.2.1	Popis hlavního programu	60
6.2.2	Popis podprogramů a obsluh přerušení	63
6.3	Simulace částí programů	64
7	Závěr	67
	Literatura	69
	Seznam symbolů, veličin a zkratk	71
	Seznam příloh	72
A	Blokové schéma řídicí, kontrolní a generující části	I
B	Blokové schéma silové části	II
C	Schéma zapojení řídicí, kontrolní a generující části	III
D	Deska plošných spojů	IV
E	Vývojové diagramy programů	VIII
E.1	Vývojový diagram pro řídicí a kontrolní mikrokontrolér	VIII
E.2	Vývojový diagram pro generující mikrokontrolér	XII

SEZNAM OBRÁZKŮ

1	Nákres prvního magnetoterapeutického přístroje [8]	10
1.1	Ukázka indukovaného elektrického pole v kosti [8]	13
1.2	Biotropní parametry obdélníkového pulzu [7]	18
3.1	Příklady vzorků uložených v paměti mikrokontroléru	24
3.2	Průběh signálu o frekvenci 100 Hz se střídou 1:20	25
3.3	Průběh signálu o frekvenci 1 Hz se střídou 19:20	27
3.4	Filtrovaná a nefiltrovaná část impulsu se šířkou 40 ms	28
3.5	Ukázka homogenity magnetického pole uvnitř solenoidu	29
3.6	Ukázka konstrukce solenoidální cívky [5]	30
3.7	Konstrukce ploché spirálové cívky	31
4.1	Struktura paketů při komunikaci po sběrnici I ² C	36
4.2	Tvar modulové charakteristiky filtru	40
6.1	Vyhodnocování stisku tlačítka u maticové klávesnice	56
6.2	Stav simulace po vygenerování třetího vzorku signálu	64
6.3	Stav simulace po vygenerování čtvrtého vzorku signálu	65
6.4	Simulace komunikace po sběrnici I ² C	66
A.1	Blokové schéma řídicí, kontrolní a generující části přístroje	I
B.1	Blokové schéma silové části	II
C.1	Schéma zapojení přístroje generujícího PMP	III
D.1	Matrice ze strany součástek (zvětšeno v poměru 2:1)	IV
D.2	Matrice ze strany spojů (zvětšeno v poměru 2:1)	V
D.3	Osazovací plán ze strany součástek (zvětšeno v poměru 2:1)	VI
D.4	Osazovací plán ze strany spojů (zvětšeno v poměru 2:1)	VII
E.1	První část vývojového diagramu hlavního programu pro řídicí MCU	VIII
E.2	Druhá část vývojového diagramu hlavního programu pro řídicí MCU	IX
E.3	Vývojové diagramy podprogramů pro řídicí MCU	X
E.4	Vývojové diagramy obsluh přerušení pro řídicí MCU	XI
E.5	První část vývojového diagramu hlavního programu pro generující MCU	XII
E.6	Druhá část vývojového diagramu hlavního programu pro generující MCU	XIII
E.7	Vývojové diagramy podprogramů pro generující MCU	XIV

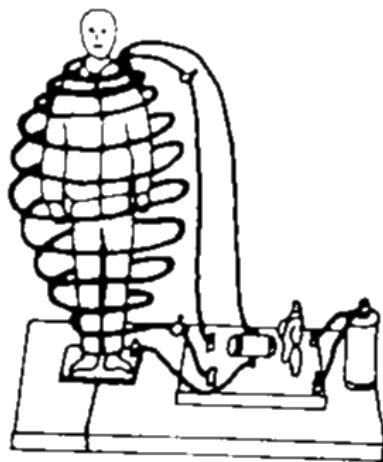
SEZNAM TABULEK

1.1	Srovnání účinků magnetických polí různých intenzit	16
4.1	Tabulka propojení externích součástek se svorkovnicemi na DPS (první část)	42
4.2	Tabulka propojení externích součástek se svorkovnicemi na DPS (druhá část)	42
6.1	Seznam příkazů pro LCD	52
6.2	Popis přednastavených programů	53
6.3	Popis funkcí generovaných Processor Expert	61

ÚVOD

Účelem této práce je prostudovat možnosti léčby dětské mozkové obrny pomocí pulzního magnetického pole. Navrhnout funkční blokové schéma přístroje generujícího pulzní magnetické pole a popsat celkovou funkci přístroje. Navrhnout obvodové řešení přístroje, vytvořit vývojové diagramy a programy pro použité mikrokontroléry a realizovat řídicí a ovládací část přístroje.

Léčebné účinky magnetického pole jsou známy už více než 2000 let. V té době byl znám pouze jediný přirozený magnet a to oxid železito-železitý, který se v antickém Řecku těžil v okolí města Magnesia. Odtud pochází název magnet. Řekové znali schopnost magnetu přitahovat některé kovové předměty a přisuzovali mu božskou moc, proto ho začali využívat k léčbě. Magnetický prášek byl podáván ústně, byl přidáván do elixíru věčného mládí atd. Největší účinek však měly obklady s magnetickým práškem přikládáné na rány a zlomeniny. Existují i zmínky o tom, že už římský císař Claudius si léčil dnu pomocí koupele s elektrickými rejnoky. První lékař, který si uvědomoval účinky magnetického pole na lidský organizmus, byl osobní lékař královny Alžběty W. Gilbert.



Obr. 1: Nákres prvního magnetoterapeutického přístroje [8]

Znalost magnetizmu se prohlubovala v 17. stol., kdy byl vyroben první umělý permanentní magnet. Dále se o rozvoj vědění o magnetizmu zasloužili v 18. a 19. stol. Oersted (dokázal souvislost mezi elektrickými a magnetickými silami), Faraday (objevil elektromagnetickou indukci) a hlavně Maxwell (objevil elektromagnetické pole

a definoval jeho vlastnosti v Maxwellových rovnicích). První magnetoterapeutický přístroj byl patentován v roce 1869 v Berlíně. Jeho nákres je na obrázku 1.

V Čechách se magnetoterapie začala využívat od 50. let 20. stol., kdy začal MUDr. Novák dělat experimenty s klecí, ve které bylo několik závitů měděného drátu, připojeného na zdroj přerušovaného stejnosměrného proudu. Dále se magnetoterapií zabýval MUDr. Hokynář, který pomocí ní léčil bércové vředy, revmatická onemocnění a onemocnění pohybového aparátu. K léčení používal statické magnetické pole s indukcí několika desítek mT. V 80. letech vyvinul MUDr. Jiří Chvojka CSc. první generátor pulzního magnetického pole a v roce 1991 otevřel první lůžkové oddělení magnetoterapie v Československu.

I když si to někteří lidé neuvědomují, všichni žijeme v trvalém působení magnetického pole. Nelze se tomu divit, člověk totiž nemá receptory, které by byly citlivé na magnetické pole (jako například sluch na akustické vlnění). Toto trvalé působení je způsobeno geomagnetickým polem Země, které lidský organizmus potřebuje k životu a jeho nedostatek mu může způsobit vážné zdravotní problémy ¹.

Magnetické pole je popisováno dvěma hlavními veličinami, a to intenzitou magnetického pole H a indukcí magnetického pole B .

Intenzita magnetického pole - značí se H a její jednotka je $A \cdot m^{-1}$. Vyjadřuje účinky pole, které je vyvolané průchodem proudu vodičem v určité vzdálenosti od vodiče. Ze vzorce (1) vyplývá, že je přímo úměrná protékajícímu proudu I a nepřímo úměrná vzdálenosti od vodiče r .

$$H = \frac{I}{2 \cdot \pi \cdot r} \quad [A \cdot m^{-1}, A, m]. \quad (1)$$

Indukce magnetického pole - značí se B a její jednotka je T (Tesla). Vyjadřuje počet magnetických indukčních čar na jednotku plochy. Na rozdíl od intenzity magnetického pole je materiálově závislá. Vypočítá se podle vzorce

$$B = \mu \cdot H \quad [T, H \cdot m^{-1}, A \cdot m^{-1}]. \quad (2)$$

Permeabilita - Tato veličina udává míru magnetizace látky působícím magnetickým polem. Udává vzájemný vztah mezi indukcí a intenzitou, což vyjadřuje rovnice (2).

Relativní permeabilita - je to bezrozměrné číslo, které udává odlišnost magnetických vlastností materiálu od magnetických vlastností vakua. Značí se μ_r . Vyjadřuje se jako poměr permeability daného materiálu k permeabilitě vakua $\mu_0 = 4 \cdot \pi \cdot 10^{-7} H \cdot m^{-1}$.

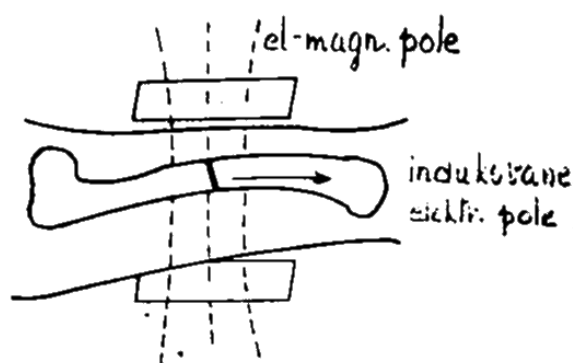
$$\mu_r = \frac{\mu}{\mu_0} \quad [- , H \cdot m^{-1}, H \cdot m^{-1}]. \quad (3)$$

¹Tato skutečnost byla dostatečně prokázána v začátcích kosmického výzkumu. První astronauti měli vážné zdravotní problémy, tzv. „vesmírná nemoc“, která byla způsobovaná absencí geomagnetického pole Země

Elektrické a magnetické pole k sobě neoddělitelně patří, spolu vytvářejí pole elektromagnetické. Magnetické pole je vlastně jen zjednodušením pole elektromagnetického (při zanedbání elektrické složky). Elektromagnetické pole je popsáno Maxwellovými rovnicemi [5]. Vzorce (1) (2) a (3) byly převzaty z [5].

1 INTERAKCE PULZNÍHO MAGNETICKÉHO POLE S ŽIVOU TKÁNÍ

V této kapitole bych se chtěl zaměřit pouze na pulzní nízkofrekvenční magnetické pole (dále jen PMP), protože se nejčastěji používá k léčbě dětské mozkové obrny. Druhá Maxwellova rovnice (zákon elektromagnetické indukce) říká, že každá změna magnetického pole vyvolá vznik vířivého elektrického pole. Protože PMP je časově proměnné a živá tkáň je vodivá, je jeho účinek neoddelitelně spjat se vznikem indukovaných proudů v exponované tkáni.



Obr. 1.1: Ukázka indukovaného elektrického pole v kosti [8]

Vektor elektrické intenzity ve tkáni je orientován po obvodu soustředných kružnic, které jsou kolmé k vektoru magnetické indukce. Intenzita indukovaného elektrického pole je přímo úměrná velikosti a rychlosti změn magnetické indukce procházející jednotlivými kružnicemi. Je třeba si uvědomit, že časový průběh indukovaného elektrického pole je závislý na časovém průběhu magnetického pole, ale není s ním totožný. Léčebné účinky indukovaných proudů jsou srovnatelné s léčebnými účinky elektroterapie. Hlavní výhodou však je, že nenastává lokální poškození buněk v místě vstupu proudu do těla, jako je tomu při kontaktní elektroterapii.

Léčebné účinky na tkáň však nemá pouze indukované elektrické pole, ale i magnetické pole samo o sobě. Magnetické pole prostupuje rovnoměrně do exponované tkáně, čili do každé buňky. Každá buňka obsahuje ionty, které jsou magneticky ovlivnitelné a způsobují aktivaci buněk.

1.1 Účinky pulzního magnetického pole na tkáň

Účinky je možno shrnout do sedmi základních skupin.

Ovlivnění látkové výměny buněk

Podle výše popsaného principu PMP rovnoměrně prostupuje buňkou a indukuje v ní proudy. Což má za následek změnu membránového napětí buňky a tím i změnu propustnosti buněčné membrány. To vede ke zvýšení látkové výměny v buňce, tedy i lepšímu okysličování, zásobování živinami a také k lepšímu odvádění škodlivin z buněk. Tyto účinky jsou základem každého ozdravného procesu.

Vasodilatační efekt

Znamená rozšíření cév, zvětšení průtoku krve a zlepšení prokrvování tkání. Vlivem PMP dochází k efluxu (vstřebávání iontů Ca^{+} po skončení svalové kontrakce) a tím i povolení napětí svaloviny cév a jejich dilataci. Dále hraje podstatnou roli aktivace vagového systému¹. Aktivací vagu dochází ke zvýšení produkce EDRF², což je faktor, který vyvolává produkci oxidu dusnatého. Zvýšením produkce oxidu dusnatého a následnou vazodilatací se sníží svalová slabost.

Magnetické pole působí i na červené krvinky, které polarizuje kladným nábojem. Polarizace působí proti „penízkovatění erytrocytů“ (jejich slepování do shluků, nebo řetězců), způsobuje jejich opětovné rozprostření do prostoru a tím zvětšuje efektivní plochu schopnou vázat kyslík. Polarizace červených krvinek ovlivňuje napětí cév, vlásečnic a tepen, ty se pak rozšíří a zvýší se průtok krve. Rovněž plasticita erytrocytů je PMP pozitivně ovlivněna. Plastičtější červené krvinky se mohou lépe vyhýbat případným překážkám v krevním řečišti a tím se zamezuje tvorbě krevních sraženin.

Toto vše má pozitivní vliv na celý kardiovaskulární systém. Vasodilatační efekt je patrný již po několika minutách terapie a trvá několik hodin. Je provázen dočasným zvýšením teploty v exponované oblasti.

Myorelaxační účinek

Znamená uvolnění svalů, snížení jejich napětí (tonu). PMP způsobuje zrychlení procesu odplavování kyselých metabolitů ze svalů. Ty způsobují bolestivé dráždění ve svalectech např. při zánětech. Tím dochází k odstranění bolesti a povolují křeče.

¹vagus — bloudivý nerv

²endothel derived relaxing factor

Zrychlení regeneračního procesu

Bylo prokázáno a potvrzeno nejen u kostí, ale také u měkkých tkání. Urychlení hojení se vysvětluje lepším prokrvováním. Při dostatečném prokrvování je ve tkáni dostatek živin a kyslíku a tím se zrychluje a zkvalitňuje hojení. Někdy se také vysvětluje nespecifikovaným podrážděním cytoplazmatické membrány, po kterém dochází k vybuzení metabolického řetězce.

Analgetický účinek

Dochází k tišení bolesti v exponované oblasti, to je způsobeno naindukovanými proudy v nervových vláknech. Proudův blokují nervová vlákna vedoucí od místa bolesti přes míchu do mozku. Tento proces společně s např. myorelaxací, zvýšeným vyplavováním endorfinů atd. způsobuje ústup bolesti. Ústup bolesti je registrován po několika terapiích a analgetický účinek trvá po několik týdnů až měsíců.

Protiedémový účinek

Protitokový účinek má přímou návaznost na zrychlení látkové výměny ve tkáni exponované PMP. Urychlení látkové výměny umožní rychlejší vstřebání edému.

Protizánětlivé působení

Zdůvodňuje se zvýšenou produkcí superoxidů, které mají za následek produkci peroxidu vodíku a zvýšení fagocytózy (schopnost pohlcení pevné částice z prostředí do buňky, v tomto případě se jedná o pohlcení škodliviny). Dále je PMP schopno zvýšit vlastní fagocytickou schopnost neutrofilů (druh bílé krvinky).

1.2 Biotropní parametry magnetického pole

Výše uvedené léčebné účinky PMP ale mohou platit pouze pro přesně definované signály, které jsou určeny svými biotropními parametry. Při nedodržení těchto parametrů se magnetické pole může stát spíše škodlivé a dostavují se patologické účinky. Proto je třeba všechny tyto parametry náležitě popsat a jejich znalost je nezbytně nutná při klinickém použití přístrojů pro magnetoterapii.

Mezi biotropní parametry signálu patří :

1.2.1 Indukce magnetického pole

Velikost indukce magnetického pole je třeba považovat za nejdůležitější biotropní parametr, který při překročení představuje značný rizikový faktor. Empiricky stanovená hodnota pro optimální léčebné účinky činí $B = 1 - 10 \text{ mT}$. Pro posouzení velikosti indukci používaných v magnetoterapii slouží výčet indukci různých druhů magnetických polí.

- biomagnetická v rozsahu indukci $10^{-14} - 10^{-8} \text{ T}$
- geomagnetická v rozsahu indukci $10^{-10} - 10^{-4} \text{ T}$
- technická v rozsahu indukci $10^{-6} - 10^2 \text{ T}$

Srovnání účinků magnetických polí různých intenzit je uvedeno v tabulce 1.1.

1.2.2 Maximální gradient magnetického pole

Gradient příslušné veličiny se definuje jako změna veličiny určená na jednotku délky ve směru největšího spádu veličiny. Tento směr je většinou totožný s osou aplikátoru (viz kapitola 3.2 na straně 28). Gradient tedy určuje nehomogenitu magnetického pole. Každé pole, které má nenulový gradient, je nehomogenní. Pro gradient magnetického pole se užívá jednotka $[\text{mT}/\text{cm}]$, ačkoliv jednotka $[\text{cm}]$ je vzhledem k velikosti některých tkání a orgánů příliš velká.

Jako maximální přípustná hodnota gradientu se udává $5 \text{ mT}/\text{cm}$. Tato normalizovaná hodnota se však pro použití v magnetoterapii jeví jako příliš přísná, většina

Tab. 1.1: Srovnání účinků magnetických polí různých intenzit

B [mT]		Biologická odpověď
hlava	tělo	
0,25 - 2,5	0,06 - 0,6	Žádný efekt
2,5 - 25	0,6 - 6	Výrazný léčebný efekt, příznivý vliv na nervový systém rychlejší hojení ran a zlomenin
25 - 250	6 - 60	Změny v dráždivosti centrálního nervového systému, stimulace excitabilních tkání, možná zdravotní poškození
250	60	Vážné nebezpečí poškození zdraví, možnost fibrilace

aplikátorů má však v některých místech gradient vyšší. Z porovnání gradientu vycházejí nejlépe velké dutinové aplikátory, ty však ozařují i orgány, které by měly být působení magnetického pole ušetřeny (pohlavní orgány, žlázy s vnitřní sekrecí).

1.2.3 Vektor

Vektor magnetického pole patří mezi vedlejší biotropní parametry. Vyjadřuje směr maximální amplitudy magnetického pole. Jak u solenoidálních aplikátorů, tak u plochých cívek je směr vektoru totožný s osou aplikátoru.

1.2.4 Frekvence

Frekvence patří mezi důležité biotropní parametry. Je definována jako počet period za jednu sekundu. Jednotkou je [Hz] (Hertz). V magnetoterapii se převážně využívá nízkofrekvenční pole v rozsahu frekvencí 1–100 Hz. Bylo zjištěno, že největší biologickou odezvu vyvolá budící signál o frekvenci alfa vlny EEG člověka, tedy 8–14 Hz.

Znát samotnou frekvenci magnetického pole nestačí, protože nám neposkytuje informaci o průběhu signálu v jedné periodě. Je tedy třeba znát tvar signálu.

1.2.5 Tvar impulzu

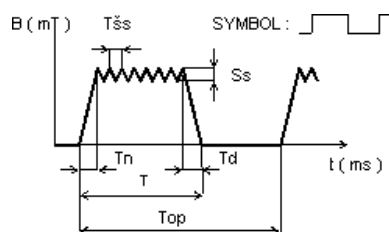
Patří mezi nejdůležitější biotropní parametry, na kterých závisí biologická odezva a vlastní terapeutický účinek. Z hlediska velikosti biologické odpovědi organismu se jako nejúčinnější tvar signálu jeví obdélníkový průběh buzený proudovými impulzy, které skokově mění svou hodnotu z maximální na nulovou hodnotu.

V terapeutické praxi se používají dva druhy pulzů :

- 1) **PS(Single Pulse)** - Jednotlivé signály, které působí přímo na buňku. Tento tvar pulzu podporuje růst cév, ty pronikají do oddělených částí kostí a tím se zlepšuje jejich růst. Používají se bifazické obdélníkové impulzy, obdélníkové impulzy s překmity do záporných hodnot, jehlové impulzy atd.
- 2) **PBs (Puls Bursts)** - Salvy impulzů příliš neovlivňují růst cév, ale značně se zvyšuje resorbce iontu Ca^{+} z tkání.

Méně časté už jsou sinusové a pulsusové průběhy (mají odříznutou zápornou půlvlnu).

U obdélníkového průběhu se sledují kromě frekvence i jiné parametry. Tyto parametry jsou popsány na obrázku 1.2



T_N	–	doba trvání náběžné hrany (s)
T_D	–	doba trvání sestupné hrany (s)
T	–	doba trvání pulzu(s)
T_{SS}	–	doba opakování šumové složky (s)
T_{OP}	–	opakovací doba impulzů (s)
$\check{S}S$	–	indukce šumové složky (mT)

Obr. 1.2: Biotropní parametry obdélníkového pulzu [7]

1.2.6 Doba expozice

Doba expozice je značně závislá na druhu onemocnění a závisí také na velikosti indukce. Při velkých hodnotách indukce se musí expoziční doby zkrátit a naopak při malých indukcích může být doba značně dlouhá. Běžné expoziční doby se pohybují řádově od několika minut do desítek hodin. Za ideální se považuje doba mezi 15 - 30 minutami. Při pokusech bylo zjištěno, že organismus je schopen se na přítomnost dlouhodobého magnetického pole adaptovat. Navíc může příliš dlouhé působení magnetického pole vyvolat nežádoucí reverzibilní změny.

Nedodržení biotropních parametrů, zejména velikosti indukce a doby expozice může, způsobit patologické změny v tkáních a vážné poškození organismu.

2 DĚTSKÁ MOZKOVÁ OBRNA

Dětská mozková obrna (dále jen DMO) není onemocnění jako takové, ale tento název zastřešuje skupinu chronických onemocnění, které se zpravidla projevují poruchami hybnosti. Pojem dětská vysvětluje věk, ve kterém se onemocnění vyskytuje, pojem mozková vyjadřuje, že onemocnění je způsobeno poruchou mozku a obrna vyjadřuje, že nemoc postihuje pohybové ústrojí. Závažnost postižení DMO se liší při různých formách nemoci a pacient od pacienta. Postižený může mít například problémy s jemnými mechanickými pracemi, jako jsou psaní a stříhání nůžkami. Hůře postižení pacienti mohou mít problémy s chůzí, udržením rovnováhy a mohou být sužováni neovladatelnými pohyby končetin a úst. DMO může být doprovázena dalšími onemocněními, jako jsou různá mentální postižení, epilepsie, hydrocefalus (patologické zvýšení množství mozkomíšního moku, což může vést k neúměrnému růstu hlavičky dítěte), poruchy citlivosti, poruchy sluchu, strabismus (v důsledku šilhání mozek ignoruje podněty z jednoho oka) a zaostávání tělesného růstu.

2.1 Příčiny vzniku DMO

DMO není jedna nemoc, ale skládá se ze skupiny onemocnění, která jsou sice příbuzná, ale mají různé příčiny. Podle příčiny vzniku se DMO rozděluje na vrozenou a získanou formu.

Získaná forma - v důsledku úrazu, dopravní nehody, týrání, zánětu mozkových blan, virové encefalitidy, nedostatku kyslíku při porodu. Krev novorozence dokáže kompenzovat dočasný nedostatek kyslíku a umožňuje dítěti přežít krátkou asfyxii (dušení), pokud asfyxie trvá příliš dlouhou dobu, dítě umírá, v lepším případě přežívá, ale většinou za cenu poškození mozku.

Vrozená forma - je přítomná již v okamžiku narození, ale projevit se může až po několika měsících nebo letech života dítěte. Výzkumem bylo zjištěno, že některé okolnosti v těhotenství nebo při porodu mohou vést k poškození motorických center mozku. Mezi tyto okolnosti můžeme zařadit zarděnky, novorozeneckou žloutenku, Rh inkompatibilitu (organizmus matky produkuje protilátky proti krvinkám plodu a v důsledku toho vzniká u plodu žloutenka), nebo cévní mozkovou příhodu.

2.2 Typy DMO

V dnešní době lékaři rozpoznávají různé typy DMO a podle charakteru pohybového postižení je řadí do čtyř kategorií:

Spastická forma - touto formou trpí asi 70 – 80 % dětí postižených DMO. Projevuje se spasticitou (zvýšeným napětím, trvalým stažením svalů). Podle oblasti postižení se rozlišuje spastická diparéze (postižení obou dolních končetin), spastická hemiparéze (postižení končetin na jedné polovině těla), spastická triparéze (postižení obou dolních a jedné horní končetiny) a spastická kvadraparéze (postižení všech končetin).

Dyskinetická forma - pro tuto formu jsou charakteristické mimovolné krouživé pohyby rukou a nohou. Někdy je postiženo svalstvo obličeje nebo jazyka.

Ataktická forma – postihuje především vnímání rovnováhy a cit pro hloubku. Pacienti mívají špatnou koordinaci pohybů, chůze je velice nestabilní, mají problémy provádět jemné práce (psaní, zapínání knoflíků).

Směšená forma – je běžné, že výše uvedené typy se kombinují, nejčastější kombinací bývá spastická forma s dyskinetickou formou.

2.3 Léčba DMO pomocí PMP

U onemocnění centrálního nervového systému a zvláště DMO se pulzní magnetické pole projevuje překvapivými kladnými výsledky tam, kde ostatní lékařské metody selhaly. Pro léčbu PMP se jeví jako nejvhodnější formy nemoci, u kterých se vyskytuje svalová atonie (ztráta napětí ve svalech), nebo formy charakterizované rigiditou (svalovou ztuhlostí). Pacienti s těmito příznaky bývají často také slabomyslní.

I když je známo, že centrální nervový systém je citlivý na PMP, je jeho léčebný účinek těžce vysvětlitelný. Zlepšení stavu pacienta je navozeno analgetickým, vasodilatačním a také výrazným myorelaxačním účinkem ve svalech. PMP bývá také aplikováno na mozek, kde hrají důležitou roli Ca^{+} ionty. Rozdíl koncentrací Ca^{+} iontů mezi prostředím uvnitř buňky a vně buňky je o 3 až 4 řády větší ve prospěch prostředí vně buňky. Poškozená buňka není schopna tento rozdíl udržet a Ca^{+} ionty do ní vstupují. Tam aktivují řadu enzymů, které zasahují do bílkovinné struktury a tím dochází k patologické reakci buňky. Pokud je však buňka ještě schopna reagovat na magnetické pole, dojde k efluxu Ca^{+} iontů z buňky, tím dojde k vyrovnání koncentrací Ca^{+} iontů uvnitř a vně buňky a tím i k ozdravnému efektu. Jelikož se velice nesnadně určuje, jaká je odezva tkáně na jednotlivé biotropní parametry PMP, jsme vesměs odkázáni na výsledky klinických pokusů.

Je vhodné podotknout, že magnetoterapie sama nestačí a souběžně s ní musí probíhat intenzivní rehabilitace např. Vojtovou metodou. Díky analgetickému účinku PMP je tato terapie podstatně snadnější a pro dítě méně bolestivá.

2.4 Možné kontraindikace

Léčba magnetoterapií musí být vyloučena, pokud pacient trpí některou z těchto nemocí: TBC, zhoubný nádor, otrava krve, ischemická choroba srdeční, psychiatrické onemocnění. Dále je magnetoterapie nevhodná pro těhotné ženy a pacienty s kardiostimulátorem.

3 BLOKOVÉ SCHÉMA PŘÍSTROJE PRO LÉČBU DMO

Blokové schéma jsem rozdělil do dvou částí a chtěl bych při jeho popisu klást důraz na vysvětlení funkce přístroje jako celku. Detailní popis fungování a zapojení jednotlivých součástí bude proveden v kapitole 4 na straně 33).

- řídicí, kontrolní a generující část (vyobrazena v příloze A na straně I příloh)
- silová a aplikační část (vyobrazena v příloze B.1 na straně II příloh)

3.1 Popis funkce řídicí, kontrolní a generující části

Přístroj obsahuje tři mikrokontroléry (dále jen MCU), dva generující a jeden řídicí. Řídicí MCU ovládá chod celého přístroje, kontroluje bezpečnost pacienta v průběhu terapie a spolu s LCD displejem a maticovou klávesnicí zajišťuje komunikaci přístroje s obsluhou. Generování signálů je zajištěno dvěma MCU, které pracují jako přímé číslicové syntezátory (blíže v kapitole 3.1.1 na straně 23). Vygenerovaný signál je převeden číslicově-analogovým převodníkem (dále jen DAC) na analogovou veličinu a filtrován dolnoproustním filtrem. Bezpečnost pacienta během terapie je zajišťována pomocí teplotního čidla, které snímá teplotu tkáně pacienta a obvodu reálného času (dále jen RTC), který měří dobu uplynulou od začátku terapie. Terapie může být kdykoliv zastavena stisknutím externího resetujícího tlačítka.

Po připojení napájecího napětí k přístroji se na LCD zapíše výzva k zadání parametrů terapie. Tyto parametry se zadávají pomocí dvanáctitlačítkové maticové klávesnice a obsluha se řídí pokyny, které jsou zapisovány na LCD. Parametry se zadávají samostatně pro oba aplikátory a obsluha si může vybrat ze tří přednastavených programů, nebo zadat parametry terapie individuálně podle svého uvážení. Pokud si zvolí druhou možnost, zadává frekvenci, střidu a indukci impulzů. Jako poslední parametr se nastavuje doba terapie (pro individuální nastavení i pro přednastavené programy).

Jakmile jsou do řídicího MCU zadány všechny požadované parametry je zahájen výpočet proměnných, které jsou potřebné pro generování signálu. Po dokončení výpočtu se proměnné odešlou po sběrnici I²C do generujících mikrokontrolérů. Ty čekají na jejich příjem a zahájí výpočet jedné periody signálu. Řídicí MCU pak nastaví teplotní čidlo, tak aby po překročení kritické teploty tkáně generovalo reset. Reset bude generovat také RTC po uplynutí doby terapie. Přístroj je připraven k zahájení terapie, je spuštěn RTC a vyslán startovací impuls ke generujícím MCU.

V průběhu terapie je řídicí MCU zacyklen v nekonečné smyčce a každou vteřinu je obvodem reálného času generováno externí přerušení. V rámci obsluhy tohoto přerušení je z teplotního čidla načtena aktuální teplota tkáně pacienta a z RTC čas, který uběhl od začátku terapie.

Po dobu generování signálu je pomocí interních čítačů generujících MCU řízena mezní frekvence dolnoproustných filtrů. V mezeře mezi impulzy signálu, kdy je signál nulový, je filtr uveden do stavu SHUTDOWN, ve kterém je jeho výstup ve stavu vysoké impedance.

3.1.1 Přímá číslicová syntéza

Generující MCU pracují na principu přímé číslicové syntézy (dále jen DDS)¹. Výhodou toho systému je velká stabilita generovaného kmitočtu, možnost generování různých průběhů, možnost přesně definovat tvar průběhu a teplotní nezávislost obvodu. Základní zapojení DDS syntezátoru se skládá ze zdroje hodinového signálu, generátoru adresy, paměti, DAC a dolnoproustného filtru. Vlastní syntezátor má dva vstupy, jsou to hodinový signál f_c a řídicí slovo FSW (Frequency Setting Word) o šířce N . Oba tyto vstupy ovlivňují frekvenci výstupního signálu. V každém taktu hodinového signálu se do generátoru adresy přičte slovo FSW. Generátor adresy tak v podstatě ukazuje do paměti, ve které jsou uloženy vzorky generovaného signálu, na index vzorku, který se bude generovat. Toto se děje až do okamžiku, než generátor adresy přeteče. Po přetečení generátoru se začne generovat nová perioda signálu.

Pro frekvenci výstupního signálu platí vzorec převzatý z [11]:

$$f_0 = \frac{FSW}{2^N} \cdot f_c \quad [Hz, -, -, Hz]. \quad (3.1)$$

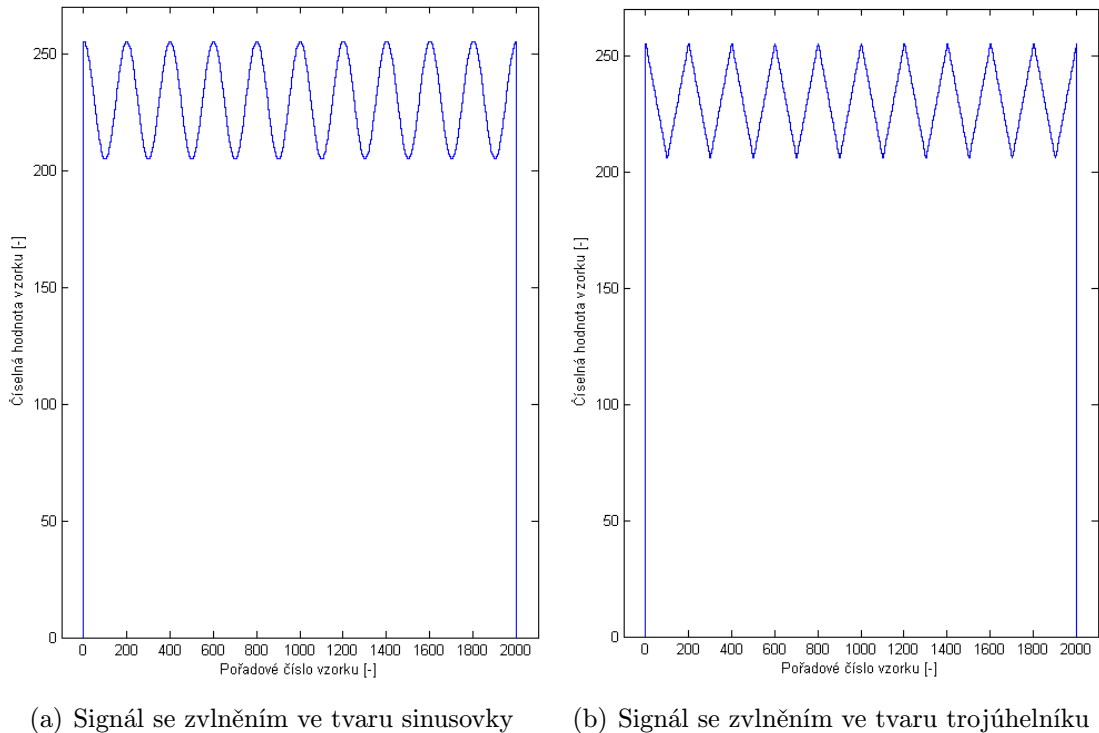
Ze vzorce (3.1) vyplývá, že čím bude vyšší hodnota vstupního slova FSW, tím více se bude výstupní frekvence blížit frekvenci hodinového signálu. Vyšších hodnot frekvencí lze ale dosáhnout pouze za cenu většího zkreslení průběhu výstupního signálu. Je to způsobeno tím, že čím bude vyšší FSW, tím více vzorků v paměti bude „přeskočeno“ a nebude se podílet na výstupním signálu. Naopak pokud bude frekvence generovaného signálu nižší, bude jedna perioda složena z více vzorků a tím bude zkreslení výstupního průběhu menší. Frekvence výstupního signálu je principiálně omezena $f_c/2^N < f_0 < f_c/2$ s rozlišením $f_c/2^N$ (vzorec platí pouze pro generování harmonického průběhu). Pro účely nízkofrekvenční pulzní magnetoterapie postačují frekvence maximálně do 100 Hz. Nízká frekvence umožňuje generovat signál, který má velké množství vzorků na periodu a tím můžeme přesně definovat průběh signálu.

¹z anglického „Digital Direct Synthesis“

3.1.2 Příklad nastavení parametrů terapie a analýza signálů

V paměti generujícího MCU, je uloženo 2000 vzorků signálu o šířce slova 8 bitů. MCU je řízen hodinovým signálem o frekvenci $f_{bus} = 10$ MHz. Vygenerování následujícího vzorku zabere MCU 100 hodinových cyklů. Frekvence generování vzorků signálu je tedy $f_c = 100$ kHz.

Na obrázku 3.1 jsou vyobrazeny dvě možné posloupnosti vzorků uložených v paměti. Jelikož přístroj umožňuje generovat signály o různých frekvencích s různou

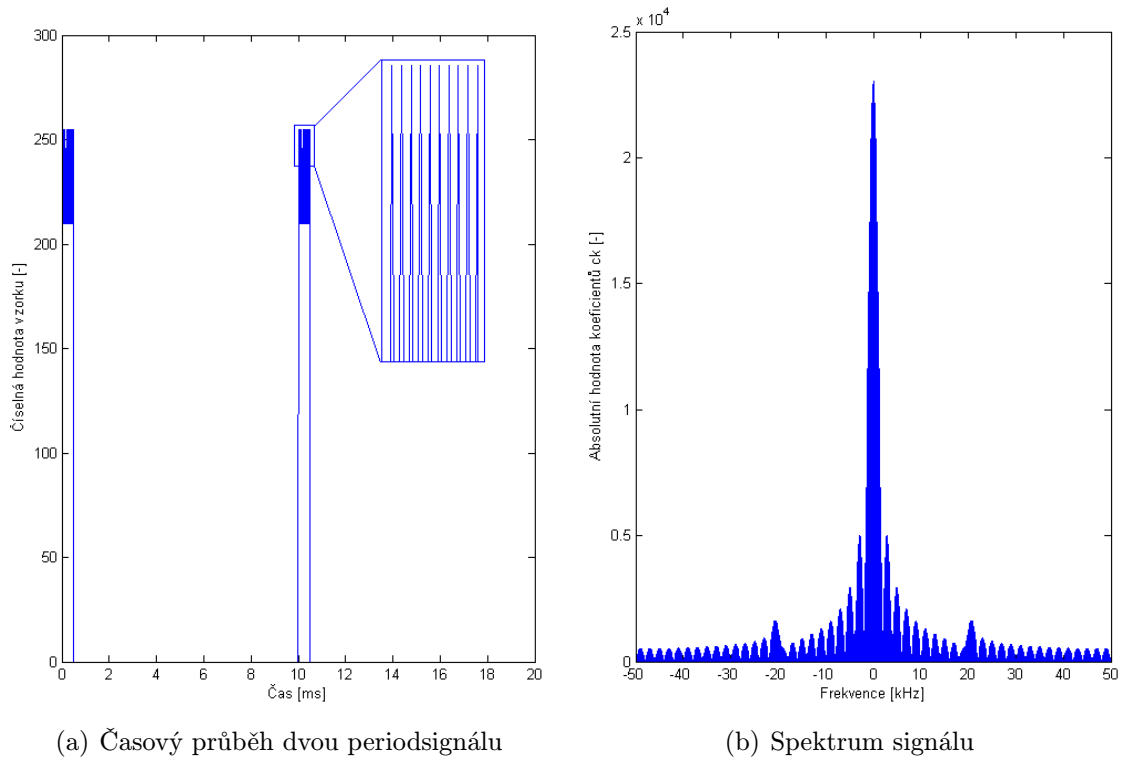


Obr. 3.1: Příklady vzorků uložených v paměti mikrokontroléru

střídou (konkrétně s frekvencí od 1 od 100 Hz a střídou od 1:20 do 19:20), budou mít průběhy různý tvar, dobu trvání, zkraslení a hlavně spektrální vlastnosti. Abychom mohli zjistit jakou mezní frekvenci musí mít dolnoproustní filtr, aby signál vyhlazoval, ale příliš nezkrasloval, musíme zjistit spektrální vlastnosti různých impulzů. V programu Matlab byla provedena simulace činnosti generujících MCU, přičemž byly použity vzorky vyobrazené na obrázku 3.1(a).

Jedná se o dva extrémní signály, první signál má nejkratší dobu trvání impulzu a bude přístrojem generován, pokud obsluha nastaví frekvenci $f_{gen} = 100$ Hz a střídu 1:20. Druhý signál má nejdelší dobu trvání impulzu a je generován, pokud obsluha nastaví frekvenci $f_{gen} = 1$ Hz a střídu 19:20.

Rozbor signálu s nejkraší dobou trvání impulsu



Obr. 3.2: Průběh signálu o frekvenci 100 Hz se střídou 1:20

Nejdříve musíme ze zadaných parametrů vypočítat frekvenční slovo FSW. Údaj o opakovací frekvenci impulsů převedeme na údaj o periodě impulsu.

$$T_{celk} = \frac{1}{f_{gen}} = \frac{1}{100} = 0,01 \text{ s} \quad (3.2)$$

Následně musíme získat časový údaj o době trvání impulsu, tedy o době po kterou se bude generovat signál. Toho dosáhneme tak, že vydělíme periodu druhou a zpětně vynásobíme první číslicí.

$$T_{fsw} = \frac{T_{celk} \cdot str1}{str} = \frac{0,01 \cdot 1}{20} = 500 \mu\text{s} \quad (3.3)$$

Pro výpočet slova FSW ovšem musíme znát „zdánlivou“ frekvenci generovaných impulsů, proto musíme proměnnou T_{fsw} invertovat. Frekvence, na kterou nastavíme DDS syntezátor je tedy 2 kHz.

$$f_{fsw} = \frac{1}{T_{fsw}} = \frac{1}{500 \cdot 10^{-6}} = 2000 \text{ Hz} \quad (3.4)$$

Nyní již můžeme vypočítat slovo FSW.

$$FSW = \frac{f_{fsw} \cdot 2^N}{f_c} = \frac{2000 \cdot 2^{16}}{100 \cdot 10^3} = 1310,72 \quad (3.5)$$

Tato hodnota FSW bude do šestnáctibitového generátoru adresy přičtena právě 50x, což znamená, že se impuls bude skládat z padesáti vzorků vyčítaných s frekvencí $f_c = 100 \text{ kHz}$. Dobu mezery mezi pulzy vyjádříme z rozdílu mezi číslicemi střídy, ten pak násobíme dobou trvání jedné dvacetiny signálu.

$$cas1 = \frac{1}{f_{gen} \cdot str} \cdot (str - str1) = \frac{1}{100 \cdot 20} \cdot (20 - 1) = 9,5 \text{ ms} \quad (3.6)$$

Na obrázku 3.2 je vyobrazen časový průběh vygenerovaného signálu a jeho amplitudové frekvenční spektrum. Na detailu zvlnění amplitudy signálu je patrné, že signál je značně zkreslen. Jde o nejrychlejší signál, který lze generovat a skládá se z malého množství vzorků, což způsobuje zkreslení.

V amplitudovém spektru signálu jsou významné složky v celém kmitočtovém rozsahu. Je to způsobeno podobností s Dirackovým impulzem, který má konstantní úroveň amplitudového spektra v celém frekvenčním rozsahu. Jakákoliv filtrace by u tohoto průběhu měla za následek spíše degradaci signálu.

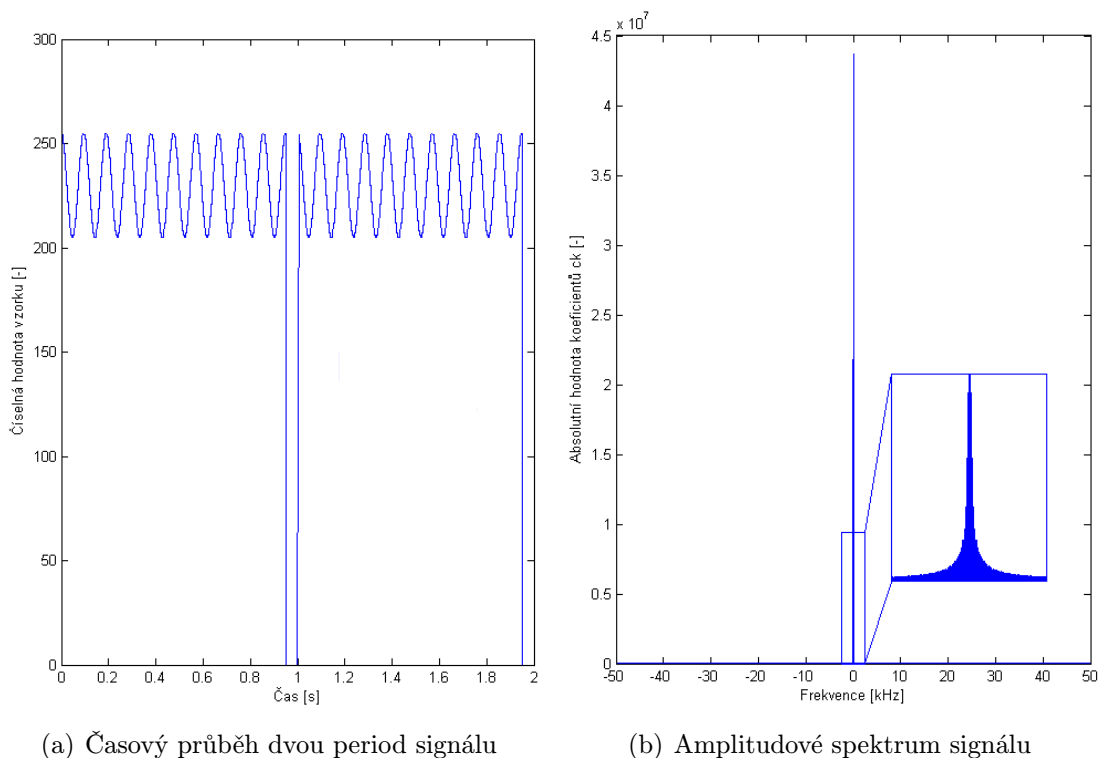
Rozbor signálu s nejdelší dobou trvání impulsu

Výpočty proměnných pro nastavení generátoru jsou shodné s předchozím příkladem, proto jsou zde uvedeny pouze jejich výsledky.

- $f_{fsw} = 1,0426 \text{ Hz}$
- $FSW = 0,6898$
- $cas1 = 0,05 \text{ s}$

Vygenerovaný signál s nejdelší délkou trvání je na obrázku 3.3. Signál se velmi přesně blíží průběhu, který jsme chtěli generovat. Signál je však značně znehodnocen kvantizačním šumem. Významné spektrální složky jsou u tohoto průběhu v těsné blízkosti stejnosměrné složky. Je to způsobeno podobností signálu se stejnosměrným signálem, který má pouze jednu spektrální čáru na nulovém kmitočtu. Při simulacích se jako nejlepší pro odstranění kvantizačního šumu jevila mezní frekvence $f_{mez} = 0,5 \text{ kHz}$.

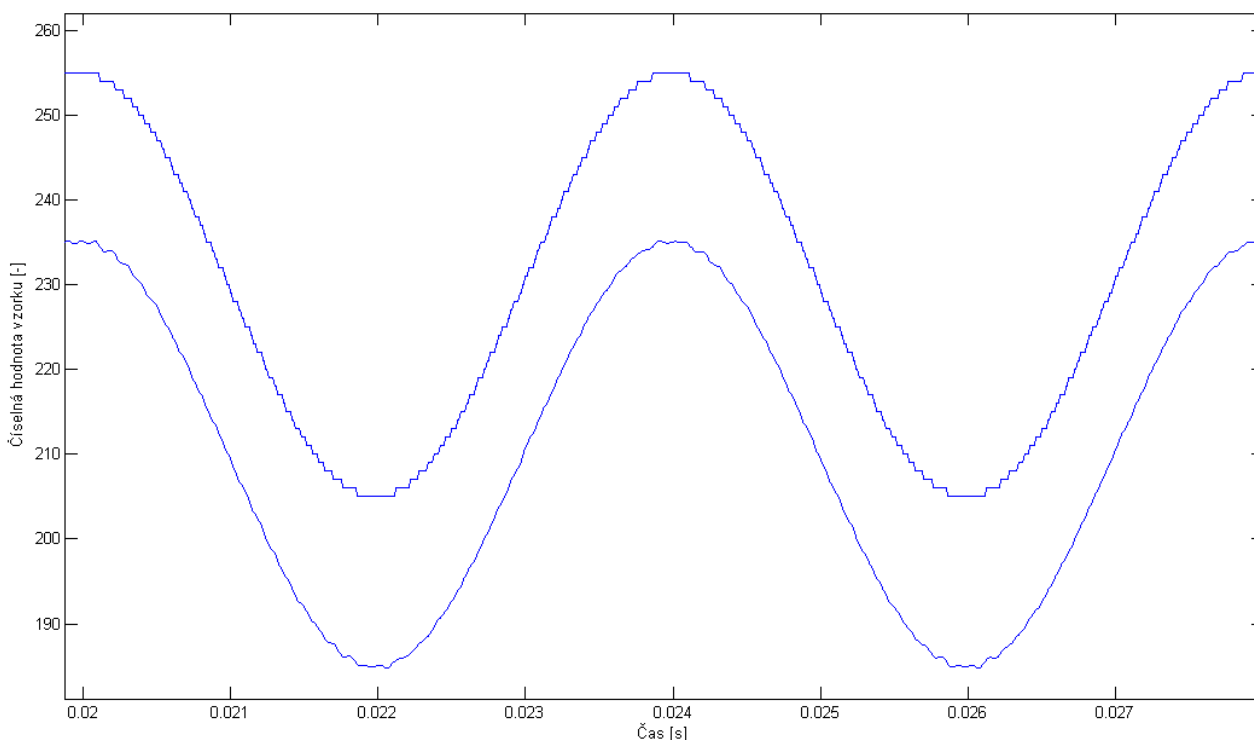
Použitý dolnoproustní filtr umožňuje měnit svou mezní frekvenci v rozsahu 1 Hz až 45 kHz. Řízení mezní frekvence filtru probíhá pomocí hodinového signálu, který je generovaný MCU. Takto můžeme nastavit individuální mezní frekvenci filtru pro



Obr. 3.3: Průběh signálu o frekvenci 1 Hz se střídou 19:20

každý generovaný impulz. Rozhodující vliv na nastavení mezní frekvence filtru, tak aby byl odstraněn kvantizační šum a signál nebyl příliš zkreslen má šířka impulzu (impulzy mohou ovšem mít stejnou šířku i při jejich různé opakovací frekvenci, vliv opakovací frekvence je však minimální). Ze šířky generovaných impulzů se tedy bude stanovovat mezní frekvence donopropustního filtru.

V programu Matlab byly odsimulovány průběhy pětatřiceti impulzů v rozmezí šířek 0,5 až 950 ms a z průběhů filtrovaných signálů byla stanovena optimální mezní frekvence filtru pro každý impulz. Výsledná závislost mezní frekvence filtru na šířce impulzu byla aproximována dvěma polynomiálními funkcemi pátého řádu, pomocí kterých bude MCU vypočítávat mezní frekvenci, na kterou nastaví dolnopropustní filtr. Na obrázku 3.1.2 jsou vyobrazeny části nefiltrovaného (nahore) a filtrovaného (dole) signálu se šířkou 40 ms. Podle odvozeného polynomu je mezní frekvence filtru, kterým byl signál filtrován 9,95 kHz. Z obrázku je patrné, že kvantizační šum byl značně potlačen.



Obr. 3.4: Filtrovaná a nefiltrovaná část impulsu se šířkou 40 ms

3.2 Popis funkce silové a aplikační části

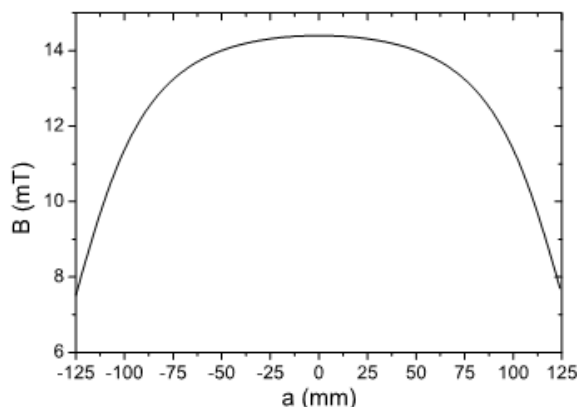
Proud dodávaný z výstupu D/A převodníku se pohybuje v řádu jednotek mA a nemá dostatečnou velikost k tomu, aby v aplikátorech vyvolal magnetické pole o potřebné indukci. Požadovaný proud je o několik řádů vyšší (viz kapitola 3.2.1), proto je potřeba signál dostatečně zesílit. Zesílení signálu je provedeno pomocí spínaného zdroje, jehož referenční napětí je řízeno signálem, který je odebírán z výstupu vyhlazovacího filtru. Je použito zapojení zdroje s regulací na primární straně impulzního transformátoru.

Síťové napětí je ve spínaném zdroji nejprve usměrněno dvoucestným usměrňovačem a vyhlazeno filtrem. Získané stejnosměrné napětí je spínáno výkonovým spínačem, čímž se rozdělí na posloupnost impulsů. Šířka těchto impulsů ovlivňuje efektivní hodnotu spínaného signálu a zároveň i výstupní výkon měniče. Spínaný signál je konečně usměrněn, filtrován a v aplikátoru přeměněn na magnetické pole. Výstupní signál je snímán a přivede se na vstup diferenčního zesilovače. Na jeho druhý vstup se přivede referenční napětí, zesilovač oba tyto průběhy odečte a na výstupu tak bude chybový signál. Referenční napětí je odebíráno z výstupu generující části a je jím v podstatě zdroj řízen. Chybový signál je na vstupu komparátoru porovnáván s napětím pilového průběhu. Při shodném napětí na obou vstupech dojde k jeho

překlopení. Komparátor v podstatě generuje pulzně modulovaný signál, jehož střída je ovlivňována chybovým napětím. Výstup komparátoru řídí spínání výkonového spínače, čímž se celá regulační smyčka uzavírá.

Oproti klasickému zdroji stabilizovaného napětí se návrh liší použitým referenčním napětím, které v tomto případě není konstantní. Zdroj musí stíhat sledovat změny referenčního napětí, aby vygenerované impulzy nebyly zkreslené. Korektnost sledování změn v referenčním signálu závisí na frekvenci pilového signálu. Čím vyšší bude frekvence pilového signálu, tím častěji se bude měnit střída (tedy i efektivní hodnota) pulzně modulovaného signálu a bude se častěji upravovat úroveň napětí na výstupu zdroje.

3.2.1 Solenoidální aplikátor



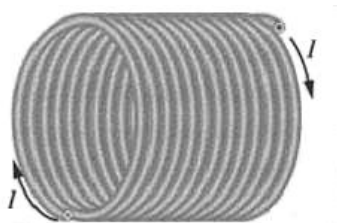
Obr. 3.5: Ukázka homogenity magnetického pole uvnitř solenoidu (měreno pro proud $I=1A$) [12]

Hlavní výhodou tohoto typu aplikátoru je homogenita indukovaného magnetického pole uvnitř aplikátoru. Solenoidální aplikátor je v podstatě odizolovaný vodič spirálovitě stočený do tvaru válce. Pokud tímto vodičem protéká proud, vytváří solenoid uzavřené magnetické pole. Tento druh aplikátorů se používá v případech, kdy potřebujeme, aby magnetické pole pronikalo hluboko do exponované tkáně. V případě léčby DMO se používá pro ozařování celého trupu. Nevýhodou těchto aplikátorů je relativně malá účinnost. K vyvolání stejné indukce, jakou mají např. ploché aplikátory budeme potřebovat větší proud protékající cívkou. Z toho plyne i větší energetická spotřeba přístroje. Homogenitu pole uvnitř solenoidu lze posoudit

ze vztahu převzatého z [12]:

$$B(a) = \frac{\mu_0 N I}{2l(r_2 - r_1)} \cdot \left[\left(\frac{l}{2} + a \right) \cdot \ln \frac{r_2 + \sqrt{r_2^2 + \left(\frac{l}{2} + a \right)^2}}{r_1 + \sqrt{r_1^2 + \left(\frac{l}{2} + a \right)^2}} + \left(\frac{l}{2} - a \right) \cdot \ln \frac{r_2 + \sqrt{r_2^2 + \left(\frac{l}{2} - a \right)^2}}{r_1 + \sqrt{r_1^2 + \left(\frac{l}{2} - a \right)^2}} \right] \quad (3.7)$$

Ze vzorce (3.7) lze vysledovat, že indukce je přímo úměrná proudu protékajícímu



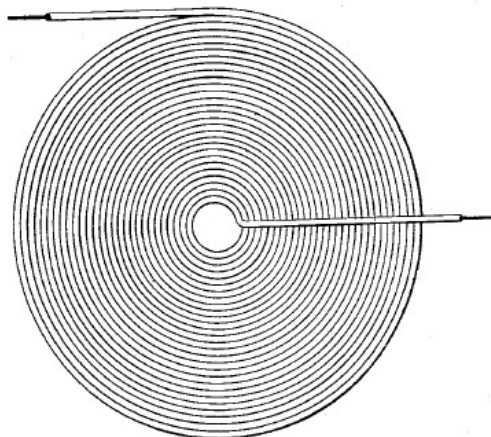
Obr. 3.6: Ukázka konstrukce solenoidální cívky [5]

cívkou I [A] a počtu závitů N . Dále je závislá na délce l [m], vnějším r_1 [m], vnitřním poloměru r_2 [m] a na vzdálenosti od středu solenoidu a [m]. V případě, že si zvolíme určité rozměry aplikátoru, konstantní proud, který do něj vstupuje a budeme propočítávat indukce v různé vzdálenosti, zjistíme, že pole v solenoidu vlastně homogenní není. Směrem k okrajům solenoidu se indukce značně zmenšuje. Toto je patrné z obrázku 3.5. Rozměry aplikátoru by měly být dostatečně velké, aby PMP pokrylo celou oblast pacientova trupu.

Mějme aplikátor o rozměrech $r_1 = 0,35$ m $r_2 = 0,36$ m a $l = 0,8$ m s 3000 závitů a chceme, aby generoval magnetické pole o indukci $B = 10$ mT. Maximální dosažitelná indukce solenoidu je v jeho středu, kdy $a = 0$ m, proto budu proud vyvolávající indukci počítat pro tuto hodnotu. Vztah (3.7) po vyjádření proudu a zjednodušení přejde v :

$$I = \frac{B \cdot 2 \cdot l \cdot (r_2 - r_1)}{\left[l \cdot \ln \frac{r_2 + \sqrt{r_2^2 + \left(\frac{l}{2} \right)^2}}{r_1 + \sqrt{r_1^2 + \left(\frac{l}{2} \right)^2}} \right] \cdot \mu_0 \cdot N} = \frac{10 \cdot 10^{-3} \cdot 2 \cdot 0,8 \cdot (0,36 - 0,35)}{\left[0,8 \cdot \ln \frac{0,36 + \sqrt{0,36^2 + \left(\frac{0,8}{2} \right)^2}}{0,35 + \sqrt{0,35^2 + \left(\frac{0,8}{2} \right)^2}} \right] \cdot 4 \cdot \pi \cdot 10^{-7} \cdot 3000} = 2,837 A \quad (3.8)$$

3.2.2 Plochý aplikátor



Obr. 3.7: Konstrukce ploché spirálové cívky

Ploché aplikátory se obecně používají v případech, kde nevyžadujeme hluboké pronikání magnetického pole do tkáně pacienta. Při stejném protékajícím proudu mají větší indukci než solenoidální aplikátory, magnetické pole je však značně nehomogenní a jeho indukce se vzdáleností od aplikátoru rychle klesá. Plochý aplikátor může být realizován plochou spirálovitou cívku. Konstrukce takovéto cívky je vyobrazena na obrázku 3.7, jedná se o vodič stáčeující se do středu kružnice. Indukce takovéto cívky se dá stanovit jako součet příspěvků od každého závitů. Výpočet se dá zjednodušit, jestliže nahradíme spirálu tenkým kruhovým závit, jehož průměr odpovídá střednímu průměru jednoho závitů spirály. Pro indukci magnetického pole jednoho kruhově symetrického závitů platí, že složka indukce jdoucí ve směru poloměru je nulová a podélná složka je vyjádřena vztahem:

$$B_0 = \frac{\mu_0 \cdot I}{2} \cdot \frac{R^2}{(R^2 + z^2)^{3/2}} \quad [T, H \cdot m^{-1}, A, m, m, m] \quad (3.9)$$

V tomto vztahu rozměr R představuje poloměr závitů kruhové cívky a z představuje vzdálenost v podélném směru (ve směru osy z , tedy do tkáně pacienta). Celkovou indukci spirálové cívky pak získáme jako součet jednotlivých složek od všech kruhových závitů.

$$B_{0celk} = \sum_{i=1}^n \frac{\mu_0 \cdot I}{2} \cdot \frac{R^2}{(R^2 + z^2)^{3/2}} \quad [T, H \cdot m^{-1}, A, m, m, m] \quad (3.10)$$

Indukce klesá se vzdáleností z s druhou mocninou, to znamená, že indukce magnetického pole ve směru ke tkáni pacienta opravdu velice rychle klesá. Tato skutečnost

i fakt, že pole dáno součtem indukcí od jednotlivých kruhových cívek způsobuje, že pole se mění jak v příčném, tak i podélném směru aplikátoru. Ploché aplikátory se při léčbě DMO pomocí PMP používají na ozařování oblasti hlavy pacientů. Vztahy (3.9) a (3.10) jsou převzaty z [1].

4 SCHÉMA ZAPOJENÍ ŘÍDICÍ, KONTROLNÍ A GENERUJÍCÍ ČÁSTI

Schéma zapojení přístroje bylo vytvořeno v návrhovém programu Eagle a je vyobrazeno v příloze C.

4.1 Popis řídicího a kontrolního MCU a zapojení jeho periférií

Základní popis mikrokontroléru

Funkci řídicího a kontrolního MCU plní mikrokontrolér MC9S08AW48 v 44 pinovém provedení v pouzdře 44QFP od firmy Freescale Semiconductor. Tento MCU patří do rodiny HCS08 a využívá stejnojmenné osmibitové jádro.

Procesor obsahuje 49152 bajtů paměti FLASH a 2048 bajtů statické paměti RAM. Umožňuje pracovat se základními perifériemi, jako jsou analogově-digitální převodník (16-ti kanálový, desetibitový) a čítač/časovač s pulzní šířkovou modulací (dvoukanálový a šestikanálový).

Komunikace mikrokontroléru s okolím je zajišťována pomocí 36 obecně použitelných vstupně-výstupních pinů a komunikačních rozhraní (SPI, SCI, I²C). Dále je MCU vybaven piny $\overline{RESET}$ a IRQ.

4.1.1 Napájení MCU, hodinový signál, resetovací obvod a externí přerušení

MCU je napájen napětím +5 V, které se přivádí mezi piny V_{DD} a V_{SS} , ty jsou označovány jako primární napájecí piny MCU. Napájecí napětí je k MCU připojeno přes dva kondenzátory. Prvním je C_{BY1} , ten má funkci blokovacího kondenzátoru (z anglického Bypass Capacitor), slouží k potlačení vysokofrekvenčního rušení v napájecím napětí a proto musí být připojen co nejbližší k napájecím pinům. C_{BY1} je keramický kondenzátor s kapacitou $0,1\mu F$, který by měl mít co nejkratší přívody, aby měl co nejmenší vlastní šum. Druhým je tzv. skupinový kondenzátor (z anglického Bulk) C_{BLK1} , je to tantalový kondenzátor s kapacitou $10\mu F$. Slouží jako zdroj energie pro současnou nabíjení několika kapacitních zátěží.

Dalšími napájecími piny jsou V_{DDAD} , V_{SSAD} , ty slouží pro napájení analogově-digitálního převodníku, a V_{REFH} , V_{REFHL} , které slouží jako referenční napětí pro analogově digitální převodník. Na všechny tyto piny je opět připojen blokovací kondenzátor C_{BY11} .

Činnost MCU je časována pomocí hodinového signálu, ten řídí časování jádra a jsou od něj odvozeny časovací signály pro všechny periférie. V zapojení je použit jako zdroj hodinového signálu krystalový rezonátor o frekvenci 20 MHz zapojený mezi piny XTAL a EXTAL, přes tyto piny se rezonátor propojí s vnitřním Piercovým oscilátorem. V zapojení je rezonátor doplněn o rezistor a dva kondenzátory. Odpor R_F by měl být uhlíkový, aby měl co nejmenší parazitní indukčnost. Jeho velikost je $5\text{ M}\Omega$ a má za úkol udržovat vstup EXTAL v lineárním rozsahu při náběhu oscilací krystalu. Kondenzátory C_1 a C_2 jsou voleny v rozmezí mezi 5 pF a 25 pF a závisí na druhu krystalového rezonátoru.

Aby mohla obsluha přístroje kdykoliv přerušit terapii, je na vstup $\overline{RESET}$ připojeno mechanické tlačítko, po jehož stisknutí se vstup $\overline{RESET}$ připojí k nulovému potenciálu. Odpor R_{UP1} a kondenzátor C_{R1} tvoří RC filtr, který má zabránit nechtěnému resetu, pokud by se na vstup $\overline{RESET}$ dostávalo rušení. Dále je k resetovacímu pinu připojen výstup $\overline{INT}$ obvodu reálného času a výstup $\overline{T_CRIT_A1}$ teplotního čidla. Piny PTA0 a PTA1 jsou spojeny s piny $\overline{RESET}$ generujících mikrokontrolérů, aby bylo možné kdykoliv ukončit terapii. Dále jsou propojeny piny PTF0 až PTF3 řídicího MCU s piny PTB6 a PTB7 generujících MCU. Po tomto spojení se přenášejí startovací a potvrzovací signály.

4.1.2 Maticová klávesnice a LCD displej

Ke komunikaci přístroje s obsluhou a pro zadávání parametrů terapie je použita maticová klávesnice KV12BLK. Jde o seskupení dvanácti tlačítek, které jsou uspořádány do matice 3×4 . Klávesnice je k MCU připojena pouze sedmi vodiči na piny PTB0 až PTB3 a PTD0 až PTD2. Čtyři vodiče (připojené k portu B) jsou přes zdvihací rezistory $R_{UP2} - R_{UP5}$ o hodnotě $4,7\text{ k}\Omega$ (z anglického pull up) spojeny s napětím $+5\text{ V}$ a je na nich (pokud není stisknuta klávesa) logická 1. Další 3 vodiče jsou připojeny k pinům přes diody $D_1 - D_3$, tyto diody chrání před zkratem při současném zmáčknutí více kláves ve stejném řádku. Tyto tři vodiče jsou připojeny k mikrokontroléru připojeny na port D. Ve schématu je připojení zjednodušeno použitím sběrnice KLAVBUS.

Další pomocnou periférií při komunikaci mezi přístrojem a obsluhou je LCD displej. V přístroji je použit LCD displej DMC20481-NY-LY. Jedná se o displej firmy Optrex Corporation, který je schopen zobrazit čtyři řádky po dvaceti znacích. Každý znak je vytvořen z matice bodů o rozloze 5×8 a celý displej je podsvícen LED diodami. Displej obsahuje interní řadič HD44780 od firmy Hitachi. Je napájen pomocí pinů V_{DD} , V_{EE} a V_{SS} , vstup V_{DD} (napájení pro logické obvody) je připojen na napětí $+5\text{ V}$. Od napětí pro napájení je odvozeno napájení pro rozsvěcování jednotlivých bodů displeje pomocí trimru R_1 , který zde funguje jako dělič napětí

pro pin V_{EE} . Pin V_{SS} je připojen na nulový potenciál. Piny LDA (anoda) a LDC (katoda) slouží k napájení podsvícení displeje. Anoda je připojena k napětí +5 V přes odpor R_1 a katoda je připojena k nulovému potenciálu.

Komunikace s MCU probíhá pomocí trojice řídicích signálů a čtyř, nebo osmi datových signálů. Čtyřbitová komunikace je pomalejší, ale používá poloviční počet vodičů a jsou k ní využity vyšší piny LCD displeje DB[7:4]. Probíhá ve dvou krocích. V prvním je mikrokontrolérem vyslán vyšší nibl osmibitových dat, nebo instrukcí. Ve druhém kroku je komunikace dokončena vysláním nižšího niblu. V přístroji je použita osmibitová komunikace, datové piny LCD displeje DB[7:0] jsou propojeny s piny PTE[7:0] na portu E pomocí sběrnice LCDBUS. Řídicí signály jsou propojeny s piny mikrokontroléru PTG0 a PTG2. Jelikož se na LCD bude pouze zapisovat, řídicí signál $R/\overline{W}$ je pevně připojen k nulovému potenciálu.

Řídicí signály mají tyto významy:

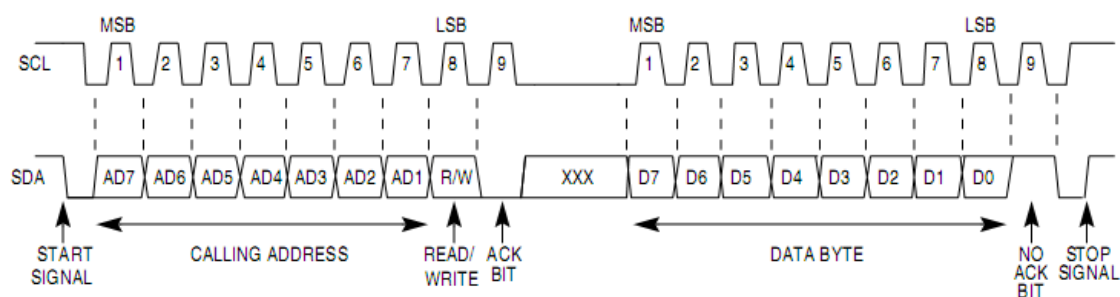
- $R/\overline{W}$ – pokud je signál v logické 0 bude se provádět zápis do LCD, pokud bude v logické 1 bude se provádět čtení dat z LCD
- RS – pokud je v logické 0, budou se přenášet instrukce, pokud bude v logické 1, budou se přenášet data
- E – povolovací signál (Enable), kladný impulz startuje komunikaci a potvrzuje platnost dat na vodičích

4.1.3 Komunikace pomocí sériové sběrnice I²C

Komunikace řídicího a kontrolního MCU s generujícími MCU, RTC i teplotním čidlem je realizována pomocí sběrnice I²C. Komunikace probíhá pomocí dvou vodičů, které jsou připojeny k pinům SDA (datová linka) a SCL (hodinová linka). Každé zařízení připojené k I²C musí mít vlastní jedinečnou sedmibitovou adresu. Ke sběrnici lze tedy připojit maximálně 128 zařízení. Oba vodiče jsou přes pull-up rezistory R_{UP6} a R_{UP7} připojeny k napájecímu napětí a jejich klidová úroveň je logická 1. Obvody komunikující po sběrnici se rozdělují do dvou skupin na Master (řídicí) a Slave (adresované). Zařízení Master zahajuje a ukončuje komunikaci a generuje hodinový signál, v přístroji plní funkci Master řídicí MCU a ostatní obvody, které komunikují po I²C fungují jako Slave.

Zařízení komunikují po sběrnici I²C s pomocí komunikačního protokolu. Komunikace je řízena hodinovým signálem, který je přenášen linkou SCL. Po dobu trvání vysoké úrovně hodinového signálu musí být datový (adresní) signál na lince SDA konstantní. Výjimku tvoří dvě speciální události start a stop, při kterých se mění úroveň na lince SDA při konstantní úrovni na lince SCL.

Obvod Master začíná komunikaci zapsáním adresy odvodu Slave na linku SDA. Adresní paket je vždy devítibitový, obsahuje sedmibitovou adresu zařízení, řídicí bit $R/\overline{W}$ (specifikuje zda chce Master přijímat, nebo vysílat data) a potvrzovací bit $\overline{ACK}$ (Acknowledge). Rozpozná-li zařízení Slave startovací bit porovnává následujících sedm adresních bitů se svou vlastní adresou a pokud se adresy shodují, zařízení Slave v průběhu devátého hodinového cyklu nuluje linku SDA, čímž potvrdí rozpoznání své adresy. Pokud je Slave zaneprázdněn, nebo se adresy neshodují, je v devátém hodinovém cyklu nečinný a Master ukončí komunikaci stop bitem. Pokud obvod Slave potvrdil rozpoznání adresy bitem $\overline{ACK}$, následuje vyslání datového paketu (buď obvodem Master, nebo Slave, podle hodnoty bitu $R/\overline{W}$). Datový paket je velmi podobný adresnímu, je opět devítibitový, obsahuje osm bitů dat a bit $\overline{ACK}$. Po odeslání adresního paketu odvod Master generuje stop bit a komunikace končí. Na obrázku 4.1 jsou vyobrazeny průběhy na linkách SCL a SDA pro jeden přenos dat. Je patrné, že zařízení Slave rozpoznalo svou adresu, ale přenos dat byl neúspěšný (bit $\overline{ACK}$ je v jedničce).



Obr. 4.1: Struktura paketů při komunikaci po sběrnici I²C, převzato z [2]

4.1.4 Obvod reálného času a teplotní čidlo

Jako obvod reálného času je v přístroji použita součástka PCF8583 firmy Philips. Jedná se o obvod schopný počítat časové intervaly v rozmezí od vteřin až po roky. Komunikovat s tímto obvodem lze po sběrnici I²C. Obsahuje funkce časovače, alarmu a možnost generování přerušení pinem s otevřeným kolektorem. Součástka obsahuje piny OSCI a OSCO, které umožňují připojení externího zdroje hodinového signálu (krystalového rezonátoru). Pro přístroj plně postačuje interní oscilátor na frekvenci 32,768 kHz, proto tyto vstupy zůstávají nezapojeny. Piny V_{DD} a V_{SS} slouží pro připojení napájecího napětí (součástka je napájena ze zdroje +5 V), napájení je blokováno

kondenzátorem C_{BY3} . Na pinu CLKOUT je implementován časovací výstup s otevřeným kolektorem, na kterém lze generovat signály s frekvencí 32,768 kHz, 1024 Hz, 32 Hz a 1 Hz. Tento pin je přímo propojen s pinem IRQ řídicího MCU. Piny SCL a SDA slouží ke komunikaci pomocí sběrnice I²C a pin s otevřeným kolektorem $\overline{INT}$ je připojen k pinu $\overline{RESET}$ řídicího MCU.

Obvod obsahuje 16 osmibitových registrů, ke kterým lze přistupovat pomocí sběrnice I²C. Každý registr má jedinečnou adresu (od 00₁₆ po 0F₁₆), specifickou funkci a význam jednotlivých bitů.

Význam jednotlivých skupin registrů:

- Adresy 00₁₆ a 01₁₆ : řídicí a stavové registry, nastavení módu a povolení přerušení
- adresy 02₁₆ až 08₁₆ : čítací registry, jejich číselná hodnota je rovna hodnotě počítané veličiny (od sekund po roky)
- adresy 09₁₆ až 0C₁₆ : registry pro nastavení alarmu, nastavení času při jehož dosažení bude generován alarm a povolení přerušení alarmu pro jednotlivé veličiny
- adresa 0D₁₆ : nastavení frekvence na výstupu CLKOUT
- adresa 0E₁₆ : povolení časovače a výběr frekvence
- adresa 0F₁₆ : hodnota časovače

Hodnoty v čítacích registrech a v registrech pro nastavení alarmu musí být uloženy v kódu BCD. Po úspěšné adresaci součástky se jako další Byte při komunikaci po I²C vysílá interní adresa registru, ke kterému chci přistupovat, a až poté se přenáší data. Výjimku tvoří situace, kdy chceme číst z několika po sobě jdoucích registrů. V tomto případě se registry nemusí pokaždé znovu adresovat, ale interní ukazatel na registry se automaticky inkrementuje.

Pro měření teploty tkáně pacienta, je použito čidlo LM92CIM firmy National Semiconductor s měřícím rozsahem -55°C až 150°C . Čidlo má dva vstupy pro napájení $+V_S$ a GND, mezi které je připojeno napájecí napětí +3,3 V, které je blokováno kondenzátorem C_{BY4} . Napětí 3,3 V je zvoleno, protože při tomto napájecím napětí má čidlo nejlepší přesnost. V oblasti tělesné teploty a vzrůstu teploty tkáně při terapii se přesnost pohybuje od $\pm 0,33^{\circ}\text{C}$ do $\pm 0,5^{\circ}\text{C}$. Senzor má dále vstupy A0 a A1, které slouží pro nastavení dvou nejnižších bitů SLAVE adresy při komunikaci po I²C sběrnici. Při připojení těchto dvou pinů k nulovému potenciálu, tedy bude adresa zařízení 1001000_b. Piny SDA a SCL jsou standardní piny I²C sběrnice. Piny INT a T_CRIT_A jsou výstupy vnitřních komparátorů. Teplota změřená

čidlem je porovnávána s hodnotami teplot, které jsou nastaveny v interních registrech čidla a po překročení jsou piny `T_CRIT_A` a `INT` nulovány. Tyto piny jsou konstruovány s otevřeným kolektorem bez vnitřního zdvihacího rezistoru. Nulování pinu `INT` je řízeno překročením maximální a minimální povolené teploty (je vlastně nastaveno „okno“ povolených teplot), v přístroji však není používáno. Naproti tomu pin `T_CRIT_A` je nulován při překročení teplotního prahu, který musí být vyšší než horní mez teplotního okna. Pin je připojen na resetovací vstup řídicího MCU.

Čidlo se skládá z třináctibitového převodníku teploty na digitální číslo, tři komparátorů, řídicí jednotky pro I²C sběrnici, šesti datových registrů (teplotní registr, nastavení módu a čtyři registry pro nastavení komparačních úrovní) a ukazatele na tyto registry. Ukazatel na registry determinuje, se kterým registrem se bude pracovat. Výběr pracovního registru se provede nastavením nejnižších tří bitů ukazatele na adresu příslušného registru.

Údaj o aktuální teplotě je uložen v šestnáctibitovém teplotním registru. Samotný údaj je dvanáctibitový se znaménkovým bitem a je uložen na horních třinácti bitech registru. MSB je znaménkový bit a zbylé tři bity jsou příznakové. Rozlišení senzoru je rovno hodnotě $LSB = 0,0625^{\circ}\text{C}$. Číselný údaj o velikosti teploty je v registru teplota zapsán ve formě dvojkového doplňku.

4.2 Popis generujících MCU a zapojení jejich periférií

V přístroji jsou použity dva shodné generující mikrokontroléry MC9S08GT16ACFB firmy Freescale, které ke své funkci používají shodné periférie a jsou řízeny shodným programem. Z tohoto důvodu bude popsán jen jeden generující MCU a jeho periférie. Mikrokontrolér je ve 44 pinovém provedení v pouzdře QFP a patří do stejné rodiny jako MCU řídicí.

4.2.1 Napájení MCU, hodinový signál a resetovací obvod

Procesor obsahuje 16384 bajtů paměti FLASH a 2048 bajtů statické paměti RAM. Podobně, jako řídicí MCU umožňuje pracovat s A/D převodníkem (osmikanálový, desetibitový) a čítačem/časovačem s pulzní šířkovou modulací (dva dvoukanálové). Komunikace mikrokontroléru s okolím je zajišťována pomocí 36 obecně použitelných vstupně-výstupních pinů a komunikačních rozhraní (SPI, SCI, I²C). Dále je MCU vybaven piny $\overline{RESET}$ a `IRQ`. Mikrokontrolér je napájen a časován stejně, jako MCU generující a jsou použity blokovací a skupinové kondenzátory shodných hodnot. Pin

$\overline{RESET}$ je ovládán řídicím MCU. Dále jsou piny PTB6 a PTB7 propojeny řídicím MCU. Po tomto propojení se přenáší startovací a potvrzovací signály.

4.2.2 Digitálně/analogový převodník

Pro převod osmibitových vzorků na analogovou veličinu je použit D/A převodník DAC0832LCN firmy National Semiconductor. Převodník je osmibitový, kompatibilní pro úroveň TTL a hodí se pro přímé (tedy paralelní) připojení k mikrokontroléru. Pracuje na principu převodníku s rezistorovou sítí R–2R s osmi přepínači, které jednotlivé rezistory připojují, buď k výstupu I_{OUT1} , nebo I_{OUT2} . Tyto přepínače jsou ovládány vstupním binárním signálem. Součástka je napájena napětím +15 V mezi piny V_{CC} a GND, napájecí napětí může být voleno v rozmezí +5 V až +15 V. Hodnota +15 V je zvolena, protože při tomto napájecím napětí má převodník optimální vlastnosti. Jako referenční zdroj je použito napětí +10 V, které je připojeno na pin V_{REF} . Ke generujícímu MCU je převodník připojen pomocí osmi paralelních datových vodičů, které propojují piny DI_0 až DI_7 převodníku a výstupní port A MCU. Činnost součástky je řízena mikrokontrolérem pomocí pěti signálů $\overline{CS}$ (Chip Select), ILE (Input Latch Enable), $\overline{WR_1}$ (Write 1), $\overline{WR_2}$ (Write 2) a $\overline{XFER}$ (Transfer Control Signal). Výstupy I_{OUT1} , I_{OUT2} a R_{FB} jsou přizpůsobeny k připojení operačního zesilovače z jehož výstupu je pak odebrán analogový signál - napětí. Proudů výstupními piny převodníků se dají vypočítat podle vzorců 4.1 a 4.2. Referenční napětí je děleno konstantou 15 k Ω , což je velikost vnitřních rezistorů R v síti R-2R. Tato hodnota je pak násobena poměrem vstupního datového slova k maximální vstupní hodnotě.

$$I_{OUT1} = \frac{V_{REF}}{15} \cdot \frac{D[0 : 7]}{256} \quad [A, V, k\Omega, -, -] \quad (4.1)$$

$$I_{OUT2} = \frac{V_{REF}}{15} \cdot \frac{255 - D[0 : 7]}{256} \quad [A, V, k\Omega, -, -, -] \quad (4.2)$$

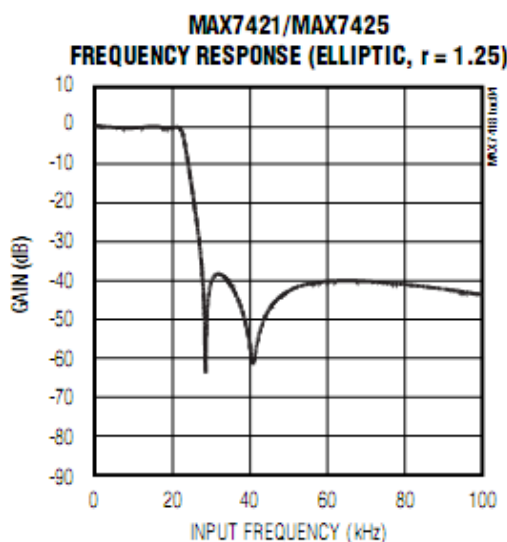
Výstupní operační zesilovač by měl mít malý vstupní klidový proud¹. Tuto podmínku splňuje operační zesilovač LF357 firmy National Semiconductor, který má vstupní klidový proud $I_{BC}=30$ pA. Vstupy zesilovače jsou zapojeny na výstupy převodníku, přičemž výstup I_{OUT2} je uzemněn. Výstup operačního zesilovače je připojen ke zpětnovazebnímu rezistoru R_{FB} , který je integrován v převodníku. Paralelně k tomuto rezistoru je připojen kondenzátor, který zrychluje odezvu na rychlé impulzy a tlumí záškuby na výstupu zesilovače. Převodník obsahuje dva interní osmibitové registry,

¹Vstupní klidový proud prochází vstupním odporem zesilovače, kde způsobuje úbytek napětí, který se po zesílení promítne na výstup operačního zesilovače. Vstupní klidový proud je způsoben nesymetrií vstupního obvodu a rovná se proudu, který se musí přivést na vstup zesilovače, aby na jeho výstupu bylo nulové napětí

přes které musí výstupní vzorky přejít, než se dostanou k vlastnímu převodníku. Tato vlastnost umožňuje držet v registrech součástky dvě různé hodnoty, což je výhodné pokud je potřeba simultánní spolupráce několika převodníků, ale vzorky do nich nejsou dodávány se stejnou frekvencí. V přístroji jsou sice dva převodníky, ale ty nepracují simultánně, proto se bude používat pouze jeden interní registr. Součástka může pracovat v několika režimech, z nichž se pro správnou funkci přístroje nejvíce hodí tzv. režim „Flow-Through“. V tomto módu je načteno vstupní datové slovo, bezprostředně projde dvojicí interních registrů a je převedeno na analogovou hodnotu. Tohoto režimu funkce dosáhneme, pokud budou vstupy $\overline{CS}$, $\overline{WR_1}$, $\overline{WR_2}$ a $\overline{XFER}$ v nízké úrovni a vstup ILE bude ve vysoké logické úrovni.

4.2.3 Analogový filtr

Funkci analogového filtru plní součástka MAX7421. Jedná se o integrovaný dolnoproпустný eliptický filtr pátého řádu se spínanými kapacitami. Povoluje měnit svou mezní frekvenci od 1 Hz do 45 kHz, je vyráběn především jako antialiasingový filtr pro vyhlazení signálu po digitálně/analogovém převodu. Modulová charakteristika filtru je zobrazena na obrázku 4.2, její zvlnění v propustném pásmu je $\pm 0,4$ dB a útlum v nepropustném je nejméně 37 dB. Charakteristika má velmi strmý přechod mezi propustným a nepropustným pásmem. Tato strmost je vyjádřena pomocí tzv. rollof faktoru, ten je charakterizován jako poměr frekvence potlačení a mezní frekvence, pro daný filtr MAX7421 činí 1,25. Součástka je napájena napětím +5 V.



Obr. 4.2: Tvar modulové charakteristiky filtru, převzato z [13]

Toto napětí se připojuje mezi piny GND a V_{DD} a je blokováno kondenzátorem C_{BY9} . Piny COM a OS slouží pro potlačení napěťového offsetu, což se v zapojení neprovádí, proto jsou piny propojeny a přes kondenzátor $C_{BY10} = 0,1 \mu F$ jsou připojeny k nulovému potenciálu. Filtr může pracovat i v tzv. Shutdown režimu, který nastane, pokud je na pinu $\overline{SHDN}$ nízká úroveň napětí. Filtr v tomto režimu odebírá z napájecího zdroje pouze minimální proud a výstup OUT je ve vysoké impedanci. Tento režim se hodí pro dobu, kdy nebude generován signál (do aplikátorů musí téct nulový proud) a filtr tak může potlačit driftový proud, který teče z DAC, pokud je na jeho vstupech hodnota 00₁₆. Piny IN slouží pro přivedení signálu a OUT jako výstup filtrovaného signálu. Mezní frekvence filtru je nastavitelná a dá se řídit pomocí frekvence obdélníkového signálu f_{CLK} připojeného k pinu CLK. Mezní frekvence se vypočítá podle vzorce (4.3).

$$f_m = \frac{f_{CLK}}{100} \quad [Hz, Hz] \quad (4.3)$$

Další možností, jak řídit mezní frekvenci, je použitím interního oscilátoru. Po připojení kondenzátoru k pinu CLK se aktivuje interní oscilátor, který generuje signál f_{CLK} . Jelikož frekvence signálu f_{CLK} je podle vzorce (4.4) závislá na velikosti kondenzátoru, dá se změnou kapacity měnit mezní frekvence filtru.

$$f_{CLK} = \frac{87000}{C_{OSC}} \quad [kHz, pF] \quad (4.4)$$

Pin CLK je připojen k pinu PTD3/TMP2CLK/TMP2CH0 generujícího MCU, mezní frekvence se tedy bude řídit hodinovým signálem.

4.3 Deska plošných spojů

Návrh desky plošných spojů (dále jen DPS) byl vytvořen v programu Eagle a je vyobrazen v příloze D na straně IV. Výsledná DPS je oboustranná. Znamená to, že spoje jsou vedeny jak na vrstvě součástek (vrstva „TOP“), tak na vrstvě spojů (vrstva „BOTTOM“). Spoje na obou stranách vrstvy jsou propojeny pomocí proků. Výsledný návrh se skládá ze dvou osazovacích plánů (ve vrstvě TOP a BOTTOM) a dvou matic (TOP a BOTTOM). Osazovací plány i matrice jsou pro větší přehlednost zobrazeny v měřítku 2:1.

Rozmístění součástek na DPS je provedeno tak, aby bylo dosaženo co nejkratší délky obou vodičů I²C sběrnice. Proto jsou součástky komunikující po této sběrnici umístěny blízko sebe. Dále jsou skupinové a blokové kondenzátory umístěny co nejbližší napájecím pinům. Krystalové rezonátory jsou umístěny co nejbližší pinům XTAL a EXTAL, stejně jako RC filtr k pinu $\overline{RESET}$. Všechna tato opatření

Tab. 4.1: Tabulka propojení externích součástek se svorkovnicemi na DPS (první část)

Součástka	Svorkovnice	Součástka	Svorkovnice
DMC20481–NY–LY	AK500/16	LM92CIM	AK500/8
DB0	1	GND	1
DB1	2	A0	2
DB2	3	A1	3
DB3	4	T_CRIT_A1	4
DB5	5	+VS	5
DB4	6	INT	6
DB6	7	SCL	7
DB7	8	SDA	8
E	9	-	9
R/W	10	-	10
RS	11	-	11
VDD	12	-	12
VSS	13	-	13
LDC	14	-	14
LDA	15	-	15
VSS	16	-	16

jsou provedena z důvodu zvýšené odolnosti proti rušení. Obě vrstvy jsou pokryty souvislou zemnicí vrstvou, která vyplňuje plochu nevyužitou pro signálové spoje.

Tab. 4.2: Tabulka propojení externích součástek se svorkovnicemi na DPS (druhá část)

Součástka	Svorkovnice	Součástka	Svorkovnice
STDHT34.1	AK500/7	tlačítko S1	AK500/2
RADEK1	1	GND	1
RADEK2	2	$\overline{RESET}$	2
RADEK3	3	-	3
RADEK4	4	-	4
SLOUPEC1	5	-	5
SLOUPEC2	6	-	6
SLOUPEC3	7	-	7

Konstrukce přístroje nedovoluje, aby na DPS byly umístěny všechny součástky. Maticová klávesnice, LCD a resetovací tlačítko musí být na ovládacím panelu přístroje. Teplotní čidlo, měřící teplotu tkáně pacienta, musí být vně přístroje. Proto jsou tyto součástky na DPS zastoupeny svorkovnicemi, ke kterým jsou pak pomocí vodičů připojeny. V tabulkách 4.1 a 4.2 jsou přiřazeny piny svorkovnic k příslušným pinům součástek.

5 VÝVOJOVÉ DIAGRAMY PROGRAMŮ PRO MIKROKONTROLÉRY

Vývojový diagram je symbolický jazyk, který se běžně používá pro přehledné vyjádření a popis algoritmů. Tento jazyk má přesně definované symboly, jejichž význam je stanoven českou státní normou ČSN ISO 5807. V následujících dvou kapitolách budou popsány dva vývojové digramy. Prvním je vývojový diagram programu pro řídicí a kontrolní mikrokontrolér (grafické vyjádření je v příloze E.1 na straně VIII). Druhým je vývojový diagram programu pro generující mikrokontrolér, jehož grafické vyjádření je v příloze E.2 na straně XII). Program pro generující mikrokontrolér lze, pouze s nepatrnými změnami, použít pro oba generující mikrokontroléry, proto je algoritmus popsán pouze jednou. Vývojové diagramy a jejich popis slouží k vysvětlení základního principu činnosti obou mikrokontrolérů. Detailně bude jejich funkce rozebrána a popsána v kapitole 6. Pro větší přehlednost a usnadnění tvoření programu jsou používány podprogramy. Tyto podprogramy jsou většinou krátké a jednoduché, proto je jejich vyjádření vývojovým diagramem (až na výjimky) zbytečné.

5.1 Vývojový diagram pro řídicí a kontrolní mikrokontrolér

5.1.1 Popis hlavního programu

Bezprostředně po připojení napájecího napětí k řídicímu MCU jsou vygenerovány signály (nízké úrovně na dvou pinech), které resetují oba generující MCU. Toto opatření je zavedeno z důvodu rychlého ukončení terapie při překročení mezních hodnot rizikových parametrů, které přístroj v průběhu terapie hlídá. Při překročení těchto mezních hodnot je řídicí MCU restován a bezprostředně jsou restovány i oba generující MCU, tím se terapie zastaví a celý přístroj se vrací do známého počátečního stavu.

Poté následuje cyklus, ve kterém obsluha zadá parametry terapie. Tento cyklus proběhne dvakrát, jednou pro solenoidální a jednou pro plochý aplikátor. Na LCD se zapíše výzva, aby obsluha vybrala program pro daný aplikátor a stiskla libovolnou klávesu. Zavolá se podprogram Sken, který v nekonečné smyčce skenuje maticovou klávesnici a vrací hodnotu stisknuté klávesy. V tomto případě jde pouze o potvrzení přečtení zprávy na LCD, proto se výstupní hodnota podprogramu nikam neukládá. Dále se na LCD zapíše výzva k výběru z tří přednastavených programů, které jsou očíslovány od jedné do trojky, nebo možnost nastavení individuálních parametrů terapie, které se spustí stisknutím čtyřky. Podprogramem Sken se do

proměnné Tlačítko načte hodnota stisknuté klávesy a podle této hodnoty se program dále větví. Pokud obsluha zvolila z jednoho z přednastavených programů, jsou do příslušných proměnných zkopírovány parametry jednotlivých terapií. Tyto parametry (řídící slovo FSW, čas trvání jednoho impulzu a proměnná, kterou se bude dělit hodnota výstupní slova pro řízení indukce) se později budou odesílat do příslušného generujícího MCU. Stiskne-li obsluha tlačítko, které neodpovídá žádnému programu ani individuálnímu nastavení, zapíše se na LCD hlášení o chybě a proces výběru terapie se opakuje.

Pokud je zvolena možnost nastavení individuálních parametrů terapie, program se podle vývojového diagramu zacyklí ve třech smyčkách, ve kterých se bude po každé nastavovat jiný parametr terapie. Tyto smyčky jsou ve vývojovém diagramu pro zjednodušení a přehlednost, ve skutečném programu budou nahrazeny třemi podobnými sekvencemi kódu. Na LCD se zapíše výzva pro zadání příslušného parametru a zavolá se podprogram Sken, který načte hodnotu stisknuté klávesy. Tato hodnota se vynásobí příslušnou vahou (např. pokud je zadávána první číslice frekvence, je hodnota klávesy násobena číslicí sto a pokud je zadávána poslední číslice, hodnota klávesy se přičítá přímo). Tento postup se opakuje tolikrát, z kolika číslic se skládá maximální hodnota zadávaného parametru (např. pro frekvenci třikrát a pro střidu dvakrát). Dále se zkontroluje, zda zadaná hodnota parametru je v rozsahu hodnot, které jsou pro přístroj přijatelné. Pokud není, na LCD se zapíše chybové hlášení, hodnota parametru se vynuluje a celý proces zadávání jednoho parametru se opakuje.

V případě, že zadání všech parametrů proběhlo v pořádku, program pokračuje výpočtem parametrů, které jsou potřebné pro činnost generujících MCU. Po úspěšném zadání všech parametrů, nebo výběru přednastavených programů pro oba aplikátory, program pokračuje dále nastavením doby terapie. Nastavení doby je pro oba aplikátory stejné, proto probíhá až po nastavení parametrů, které jsou pro oba aplikátory různé. Nastavování doby terapie probíhá v podstatě stejně, jako nastavování parametrů terapie, které bylo popsáno v předchozím odstavci.

Nastavování parametrů terapie je tedy ukončeno a následuje odesílání zadaných parametrů terapie generujícím MCU. Komunikace se realizuje pomocí sběrnice I²C. Zadané parametry odesílá funkce I2C_write_32bit, parametry jsou ve 32bitovém formátu float, proto se zadané hodnoty musí rozdělit na čtyři osmibitové proměnné, které se posílají samostatně. Po dokončení odesílání se podprogramem char_2_BCD převede dekadická hodnota doby trvání terapie na hodnotu v kódu BCD. Tento převod je nezbytný, protože do registrů čítače reálného času se musí hodnoty ukládat právě v kódu BCD. Následující funkce RTC_init nastaví čítač reálného času po sběrnici I²C tak, aby generoval resetovací signál po uplynutí doby terapie. Dalším nastavením je generování signálu o frekvenci 1 Hz na pinu CLKOUT, které zajistí,

aby bylo v řídicím MCU každou vteřinu generováno externí přerušení. Obvod reálného času je tedy nastaven, ale prozatím nefunguje, protože ještě nemá nadefinovaný zdroj hodinového signálu. Funkce `LM92_init` po sběrnici I²C nastaví teplotní čidlo tak, aby generovalo resetovací signál po překročení prahové hodnoty teploty tkáně pacienta. Všechny zmíněné podprogramy, které zprostředkovávají komunikaci po sběrnici I²C, mají stejný základ a liší se pouze ve specifikách komunikace s jednotlivými součástkami. Tento základní podprogram komunikace po I²C bude popsán v kapitole 5.1.2.

Po dokončení komunikace s perifériemi a nastavování parametrů se čeká na potvrzení dokončení výpočtů od obou generujících MCU. Toto potvrzení znamená, že generující MCU jsou připraveny zahájit terapii. Po doručení potvrzení od generujících MCU se na LCD zapíše „probíhá terapie“ a připraví se zápis proměnných, které se budou každou sekundu aktualizovat. Následně se spustí počítání času uplynulého od začátku terapie. Toho se docílí nadefinováním hodinového signálu pro RTC, které se provede nastavením příslušného bitu ve vnitřním registru RTC. Odstartování terapie se dokončí vysláním startovacích impulzů pro generující MCU a ty bezprostředně poté začnou generovat impulzy výstupního signálu. Nakonec se globálně povolí přerušení a MCU se zacyklí v nekonečné smyčce. Z této nekonečné smyčky se dostane při obsluze externího přerušení, které přichází každou vteřinu od RTC.

5.1.2 Popis podprogramů a obsluh přerušení

Podprogram `Sken` slouží pro skenování maticové klávesnice a vyhodnocení, která klávesa byla zmáčknuta. Funkce nemá žádné vstupy a jejím výstupem je číselná hodnota stisknuté klávesy. Po zavolání funkce se na výstupní piny MCU, ke kterým je klávesnice připojena, vyšle skenovací posloupnost pro daný sloupec a přečte se hodnota na pinech, které jsou definovány jako vstupní. Na základě hodnot na těchto vstupních pinech se rozhodne, zda byla stisknuta klávesa. Pokud žádná klávesa v daném sloupci stisknuta nebyla, vyšle se skenovací posloupnost pro následující sloupec. Takto se opakovaně skenují všechny sloupce dokud není detekován stisk klávesy. Po detekci stisku se zavolá funkce `Antizakmit`, která vrátí hodnotu `true`, pokud byla klávesa skutečně stisknuta a hodnotu `false`, pokud klávesa stisknuta nebyla. Pokud funkce `Antizakmit` vrátí `false`, vrátí se program ke smyčce, ve které se skenují řádky klávesnice. Pokud funkce vrátila `true`, rozhodne se podle přijaté posloupnosti bitů, která klávesa v daném řádku byla zmáčknuta, a funkce vrátí její číselnou hodnotu. Pokud přijatá posloupnost neodpovídá žádné stisknuté klávese, znamená to, že obsluha stiskla v daném sloupci dvě klávesy. V takovémto případě se na LCD zapíše varování, ať obsluha mačká pouze jednu klávesu. Nastaví se chybová proměnná a

funkce se ukončí.

Funkce Sken obsahuje volání funkci Antizakmit. Tato funkce má za úkol odstranit zákmity při stisku tlačítka a zkontrolovat, zda nejde o falešnou detekci stisku tlačítka. Po zavolání funkce se načte posloupnost bitů na výstupu klávesnice a zavolá se zpoždění běhu programu o 25 ms. Po tomto zpoždění již odezněly zákmity tlačítka, posloupnost bitů se načte znovu a porovná se s původní posloupností. Pokud jsou obě posloupnosti shodné, tlačítko bylo skutečně zmáčknuto a funkce vrátí true. Pokud se posloupnosti liší, jedná se o falešnou detekci a funkce vrátí false.

Interní čítač reálného času ¹ každou vteřinu generuje přerušení v jehož obsluze se inkrementuje chybové počítadlo. Pokud chybové počítadlo dosáhlo hodnoty tři, program se zacyklí v nekonečné smyčce a čeká se na resetování celého MCU pomocí obvodu warchdog. Pokud je chybové počítadlo menší než tři, resetuje se obvod watchdog a obsluha přerušení končí. Proměnná chybové počítadlo se nuluje také při obsluze externího přerušení od RTC. Pokud externí obvod reálného času tři sekundy po sobě negeneruje externí přerušení, je MCU takto resetován. Tento algoritmus funguje jako ochrana před zaseknutím, nebo jinou poruchou funkce externího RTC.

Externí přerušení je generováno každou sekundu externím obvodem reálného času. Na začátku obsluhy se resetuje watchdog a vymaže se obsah chybového počítadla, které bylo zmíněno v předchozím odstavci. Zakáže se interní čítač reálného času, aby negeneroval přerušení a tím se také resetuje. Dále se po sběrnici I²C načtou data a převedou se na údaj o aktuální teplotě. Tento údaj se rozdělí na celou a desetinnou část, které se budou zobrazovat samostatně. Poté se z externího RTC načtou údaje o době uplynulé od začátku terapie. Z počítačích registrů se po sběrnici I²C pošlou údaje o minutách a vteřinách. Tyto údaje se pak převedou funkcí BCD_2.char na dekadické číslo. Funkce, které načítají hodnoty po sběrnici I²C, jsou opět modifikací obecné funkce, která byla popsána výše. Dále se pro oba aplikátory vypočte aktuální terapeutická dávka, kterou pacient obdržel. Aktuální teplota, doba uplynulá od počátku terapie a aplikovaná terapeutická dávka se zobrazí na LCD. Opět se resetuje watchdog a obsluha externího přerušení končí.

¹Je to čítač, který je implementován na čipu MCU. Nejedná se tedy o externí čítač reálného času PCF8583, který byl v textu označován jako RTC

5.2 Vývojový diagram pro generující mikrokontroléry

5.2.1 Popis hlavního programu

Okamžitě po startu generujícího MCU se zavolá funkce `I2C_init`, která inicializuje periférii I²C v MCU. Po inicializaci program čeká, až řídicí MCU zapíše na datovou linku sběrnice adresu generujícího MCU. Poté proběhne příjem první proměnné, která je potřebná pro nastavení generátoru. Příjem proměnných probíhá třikrát v řadě, přičemž je známo pořadí v jakém řídicí MCU proměnné vysílá. Není tedy problém v identifikaci, která proměnná byla vyslána. Jelikož je proměnná čas vyslána v sekundách, převede se na stovky mikrosekund a převedena z formátu float na formát integer. Proměnná `cas` je vstupem funkce `Cpu_Delay100US`, která tyto úpravy vyžaduje. Z přijaté proměnné `FSW` je vypočítána šířka impulsu, která je pak vstupním parametrem funkce `mezni_frekvence`. Tato funkce pro zadanou šířku impulsu vrací mezní frekvenci, na kterou se nastaví mezní frekvence filtru. Funkce vrací mezní frekvenci v kilohertzech, ta se převede na údaj o periodě, který se převede na vyjádření v milisekundách.

Signál musí být generován s frekvencí 100 kHz, frekvence interní sběrnice MCU je 10 MHz. Jednotlivé vzorky tedy musí být vygenerovány během sta hodinových cyklů jádra MCU. Takovéto rychlosti není při přímém generování možné dosáhnout. Proto je nutné vzorky pro generování předem připravit. Tato příprava probíhá v cyklu, který je ve vývojovém diagramu zvýrazněn čárkovanou čarou. Do proměnné `generator_adresy` se překopíruje hodnota `FSW`, která byla přijata od řídicího MCU. Následně se zkontroluje, jestli generátor adresy přetekl. Pokud nepřetekl, vypočítá se z proměnné `generator_adresy` index vzorku, jehož hodnota ukazuje na pořadové číslo vzorku signálu, která se má generovat. `Index_vzorku` vyjde jako reálné číslo, které se musí převést na celočíselnou hodnotu. Tento převod se realizuje pomocí funkce `ceil1`, která jakékoliv reálné číslo převede na nejbližší vyšší celé číslo. Poté se zkontroluje, zda vygenerovaný `index_vzorku` je shodný jako `index` vygenerovaný v předchozím kroku. Pokud jsou indexy shodné, v poli `opakovani` se inkrementuje hodnota na pozici, kterou udává proměnná `counter`. Pole `opakovani` tedy bude obsahovat počet opakovaného vyčítání jednoho vzorku. Znovu se do generátoru adresy přičte `FSW` a cyklus se uzavírá kontrolou, zda přetekl generátor adresy. Pokud je vygenerovaný `index_vzorku` ve dvou po sobě jdoucích krocích různý, znamená to, že byl vygenerován nový vzorek. Inkrementuje se proměnná `counter` a do pole `indexy` na pozici, na kterou ukazuje `counter` (o jedničku větší, než v předcházejícím kroku), se zapíše hodnota nově vygenerovaného indexu. Cyklus se pak znovu uzavře přičtením `FSW` do generátoru adresy a rozhodováním, zda generátor přetekl. Pokud

generátor přetekl, znamená to, že byla vygenerována jedna perioda zvlnění signálu a příprava pro generování končí. V poli indexy jsou tedy uloženy vygenerované indexy, na kterých se nachází požadované vzorky a v poli opakovani je informace, kolikrát za sebou se bude příslušný vzorek vyčítat. Dále se v poli indexy nahradí vygenerované indexy konkrétními hodnotami vzorků a vypočítá se celkový počet různých vzorků (bez opakujičích se), které se budou generovat během doby trvání jednoho impulsu. V jednom impulsu je deset period zvlnění, čili se vynásobí proměnná counter deseti.

Pomocí funkce `Frekvence_filtr_SetPeriodReal` se nastaví čítač/časovač 2 tak, aby na výstupním pinu MCU generoval signál s dříve vypočtenou periodou. Tímto signálem je řízena mezní frekvence filtru. Následně je odesláno potvrzení o dokončení přípravy signálu řídicímu MCU a program čeká na příchod startovacího signálu. Po jeho příchodu se program zacyklí v nekonečné smyčce, ve které jsou na výstupní port MCU generovány impulzy výstupního signálu. Tato nekonečná smyčka je ve vývojovém diagramu vyznačena čerchovanou čarou.

V nekonečné smyčce probíhá cyklus, ve kterém je vygenerován jeden impuls. Bude tedy probíhat právě tolikrát, kolik je počet různých vzorků v jednom impulsu. Každým průchodem cyklem se inkrementuje proměnná `j`, která tedy může nabývat hodnot od jedné do celkového počtu různých vzorků v rámci jednoho impulsu. Vzorky signálu jsou předpřipravené pouze pro jednu periodu zvlnění signálu. Proto se při generování dělí proměnná `j` modulo počet různých vzorků za jednu periodu zvlnění. Tím se zaručí, že bude dokola generována stále jedna perioda zvlnění. Výsledek po modulárním dělení se zapíše do proměnné vzorek a v podstatě jde o pořadové číslo požadovaného vzorku v proměnné indexy. Tento vzorek je pak vygenerován na port A. Následně se rozhodne, zda v poli opakování na pořadovém čísle, které je rovné právě vygenerovanému vzorku, je nula. Pokud v poli opakování není nula, bude se právě vygenerovaný vzorek několikrát opakovat. Opakování vzorků je ekvivalentní stavu, kdy MCU na výstupním portu drží hodnotu vzorku a průběh programu je zpožděn o dobu, která by byla potřebná pro vygenerování požadovaného počtu vzorků. Toto zpoždění se realizuje pomocí čítače/časovače1. Vypočítá se hodnota modulo registru, na tuto hodnotu se časovač nastaví a poté se čeká na přetečení čítače. Po přetečení se inkrementuje proměnná `j` a celý cyklus vyčítání se uzavírá. Pokud je v proměnné opakovani nula, zpozdí se z důvodu správného časování devíti příkazy `nop`, inkrementuje se proměnná `j` a opět se celý cyklus vyčítání uzavírá. Jakmile je proměnná `j` rovna počtu různých vzorků v jednom impulsu znamená to, že byl vygenerován jeden celý impuls. Na výstupní port A se zapíše nula a zavolá se funkce `Cpu_Delay100US`. Tato funkce má jako vstupní parametr dobu trvání mezery mezi impulzy ve stovkách mikrosekund a o tuto dobu zpozdí vykonávání programu. Po uplynutí zpoždění byl vygenerován jeden impuls, mezera mezi impulzy a uzavřením nekonečné smyčky začíná generování dalšího impulsu.

5.2.2 Popis podprogramů

Funkce `mezni_frekvence` vypočítává mezní frekvenci, na kterou je potřeba nastavit analogový filtr. Výpočet je proveden pomocí dvou polynomiálních funkcí pátého řádu, jejichž koeficienty byly získány ze simulací průběhů v programu Matlab. První polynom určuje mezní frekvenci pro impulzy se šířkou 0,5 až 50 ms, druhý polynom pak pro impulzy se šířkou 50 až 950 ms. Vstupní argument funkce se překopíruje do proměnné `nasobek`. Funkce je rozdělena na dvě větve, ve kterých se počítá funkční hodnota polynomu pro vstupní argument. Kterou větví se program vydá, se rozhoduje podle šířky impulzu na základě kritéria popsaného výše. Obě větve podprogramu jsou shodné, proto zde popíšeme pouze jednu. Proměnná frekvence je inicializována tak, aby obsahovala nulu a je do ní přičten poslední koeficient polynomu, tedy koeficient u nulté mocniny. Následně proběhne čtyřikrát cyklus, ve kterém se vždy koeficient polynomu vynásobí umocněným vstupním argumentem funkce, přičemž se každým průchodem cyklem mocnina argumentu zvětší o jedničku. Po ukončení cyklu se zkontroluje, zda vypočtená mezní frekvence nepřekročila 45 kHz, což je maximální mezní frekvence, na kterou lze nastavit filtr. Pokud je tato frekvence překročena, je nastavena nová mezní frekvence na 45 kHz.

6 PROGRAMY PRO MIKROKONTROLÉRY

Kompletní kód programů je umístěn na přiloženém CD v adresáři Programy. V následujících kapitolách je pouze vysvětlena stavba a funkce programů. Zdrojové kódy byly vytvořeny v programu CodeWarrior firmy Freescale v programovacím jazyku C. V programech chybí inicializace jader procesorů a inicializace některých používaných periférií. Tyto inicializace byly vytvořeny pomocí utilit Processor Expert (generující MCU) a Device Initialization (řídící MCU). Pro řídící MCU nebyla použita utilita Processor Expert z důvodu kolize s knihovnou pro LCD.

V programech jsou použity direktivy `#include` a `#define`. Direktiva `#include` slouží k vložení jiného zdrojového textu do projektu a `#define` slouží k definování makra. Rozdíl mezi podprogramem a makrem spočívá ve způsobu překladač do JSA. Zatímco na funkci se po zavolání pouze skáče a musí se ukládat návratová adresa, makro se pokaždé celé přeloží přímo do hlavního programu.

6.1 Programy pro řídící MCU

6.1.1 Popis hlavního programu

Program začíná vložením souborů `hidef.h` a `derivative.h`, které slouží k definování typu MCU, definování vektorů přerušení, symbolické pojmenování registrů atd. Dále je vložena knihovna pro práci s LCD `lcd.h`. Knihovna pro komunikaci s LCD je získána z oficiálních internetových stránek firmy Freescale. Je vytvořena pro displeje, které používají řadič HD44780 od firmy Hitachi a funguje pro mikrokontroléry, které používají jádra HC08 a HCS08. Knihovna se musí pro konkrétní použití upravit. Musí se definovat typ mikrokontroleru i rodina do které patří, určit jestli se bude používat osmivodičová, nebo čtyřvodičová komunikace a na které piny MCU jsou připojeny řídící a datové vodiče. Moderní mikrokontroléry mají vysokou taktovací frekvenci v řádu megahertzů, ovšem vykonání instrukce řadičem LCD trvá určitou dobu (například vymazání displeje trvá 1,64 ms). Pokud by tedy šlo více instrukcí k LCD za sebou, řadič by je nestihl provádět. Proto je potřeba nastavit čítač/časovač tak, aby generoval přerušení každých 100 μ s a nastavit kolikrát musí časovač přetéct, aby byl dosažen čas jedné sekundy. O časování komunikace s LCD se už pak postará sama knihovna. LCD se pak ovládá příkazy z tabulky 6.1.

Následuje deklarace a definice potřebných proměnných, deklarace inicializační funkce jádra `MCU_init` a definice hlavní funkce `main`, na jejímž začátku se volá funkce `MCU_init` a `LCD_init`.

Tab. 6.1: Seznam příkazů pro LCD

Jméno funkce	vstupní argumenty	popis funkce
LCD_Init	void	provede inicializaci LCD
LCD_Clear	void	vymaže obsah LCD
LCD_Print	unsigned interger, délka	vypíše na LCD řetězec znaků dané délky
LCD_Cursor	unsigned integer	nastaví kurzor na vstupní adresu

Při překročení povolených parametrů terapie, je řídicí MCU resetován, a aby byla terapie skutečně zastavena jsou bezprostředně po resetu řídicího mikrokontroléru resetovány oba generující MCU. Reset je vykonán vysláním logických nul na piny 0 a 1 portu A. Okamžitě po zahájení jakéhokoliv resetu, interního i externího, je pin $\overline{RESET}$ mikrokontrolérem nulován na 34 hodinových cyklů. Poté je pin uvolněn a po dalších 38 hodinových cyklech je znovu zjištěn jeho stav. Pokud je i po tomto pin v nule, mikrokontrolér pozná, že se jednalo o externí reset. Z tohoto důvodu se do výkonu programu zavádí zpoždění, které je o málo delší, než 72 hodinových cyklů. Po tomto zpoždění jsou generující mikrokontroléry resetovány a piny PTA0 a PTA1 jsou opět nastaveny do vysoké úrovně.

Dalším krokem ve výkonu programu je zadávání parametrů terapie. Vlastní zadávání probíhá ve dvou cyklech. V prvním, ve kterém se nastavují parametry pro solenoidální aplikátor, má proměnná opakování hodnotu 1 a vzhledem k tomu se algoritmus nastavování parametrů upravuje. Při druhém průchodu cyklem se nastavují parametry pro plochý aplikátor a proměnná opakování má hodnotu 2. V prvním kroku cyklu se proměnná kontrola nastaví na jedničku. Následuje cyklus while, který bude probíhat, dokud bude proměnná kontrola rovná jedné. Jako první příkaz cyklu se tato proměnná nuluje. Pokud celý cyklus nastavování parametrů a výběru programu terapie proběhne v pořádku, bude proměnná kontrola na konci stále nulová a cyklus proběhne pouze jednou. Pokud ovšem obsluha udělá chybu, (např. stisk dvou kláves najednou, parametr terapie přesahuje povolený rozsah, je zvoleno špatné tlačítko), na LCD se zapíše hlášení o chybě, proměnná kontrola se nastaví na jedničku a celý cyklus probíhá znovu.

LCD se vymaže, kurzor se nastaví do levého horního rohu, tedy na pozici 00_H . Pro přemístění kurzoru na první znak druhého řádku se vyšle adresa 40_H , pro třetí 80_H a pro čtvrtý $c0_H$. Po vypsání příslušné výzvy se zavolá podprogram sken, funkce sken sice vrací hodnotu stisknuté klávesy, ale ta se při tomto způsobu volání nikam neuloží. Tímto se v podstatě čeká na stisk libovolné klávesy. Pokud stisk proběhl v pořádku, proměnná kontrola je rovna nule a cyklus se ukončí.

V následujícím kroku se na LCD zapíše výběr ze tří zvolených programů, nebo

individuální nastavení parametrů terapie. Kontrola správnosti stisku i skenování funguje stejně, jako v předchozím případě. Hodnota, kterou vrátí funkce sken, se uloží do proměnné tlačitko. Tato proměnná pak funguje, jako vstup funkce switch. V této funkci se rozhodne, který program byl vybrán, nastaví se jeho parametry pro příslušný aplikátor(cas, FSW, del, frekvence, davka) a příkazem break se ukončí. Nastavení správných proměnných pro příslušný aplikátor probíhá opět pomocí proměnných opakovaní. Popis parametrů jednotlivých přednastavených programů je v tabulce 6.2.

Tab. 6.2: Popis přednastavených programů

Číslo programu	1	2	3
Frekvence [Hz]	100	1	50
Střída [-/-]	1:20	19:20	10:20
Indukce [mT]	2	10	20

Vypočítá se také základní hodnota odhadu terapeutické dávky, kterou pacient obdrží během terapie. Tento odhad se stanoví podle vzorce 6.1. Odhad aplikované dávky tedy závisí na strmosti impulzů ¹, špičkové hodnotě magnetické indukce, době trvání terapie, a opakovací frekvenci impulzů. Hodnota odhadu dávky vypočtená v tomto kroku nezahrnuje vliv doby trvání terapie. Je to pouze předpřipravená hodnota, která se bude v průběhu terapie korigovat.

$$Davka = \frac{dB}{dt} \cdot B_{max} \cdot t \cdot f \quad \left[\left(\frac{T}{s} \right)^2 \cdot hod, \frac{T}{S}, T, hod, Hz \right]. \quad (6.1)$$

Pokud se obsluha rozhodla pro individuální nastavení parametrů terapie, zapíše se na LCD zpráva, aby zadala frekvenci generovaných impulzů. Poté se kurzor displeje posune na pozici 48_H, tedy na druhý řádek na pozici za dvojtečku. Zadávání frekvence začíná od stovek, proto se hodnota načtené klávesy násobí číslem 100 a přičte do proměnné frekvence. Hodnota stisknuté klávesy se zapíše na displej na pozici stovek. Stejně se načítají a vážují i zadané jednotky a desítky. Následně se podmínkou if zkontroluje, zda je zadaná frekvence ve správném rozsahu, tedy 1 - 100 Hz. Pokud není, nastaví se proměnná kontrola na jedničku, vynuluje se proměnná frekvence, na LCD se zapíše chyba, zavolá se zpoždovací funkce zpozdeni_2s() a celý cyklus nastavování frekvence začíná znovu. Podobným algoritmem proběhne i nastavení střídy a indukce impulzů. Ze zadaných parametrů terapie se pro oba

¹Ve výpočtu je zohledněna strmost s jakou jsou impulzy generovány MCU. Nedá se však předpokládat, že výkonový zesilovač bude mít dostatečné frekvenční vlastnosti a bude hrany zpomalovat. Výpočet by se tedy musel upravit podle reálných možností přístroje

aplikátory musí vypočítat proměnné, které budou potřebné pro generování impulzů. Tyto výpočty se provedou aplikací vzorců 3.2 až 3.6.

Pokud obsluha zmáčkne klávesu, která neodpovídá žádnému programu, ani individuálnímu nastavení parametrů, ukáže přepínač switch na položku default. Poté se na LCD zapíše chybové hlášení, nastaví se proměnná kontrola a celý cyklus zadávání parametrů začíná znovu. Po úspěšném zadání parametrů pro oba aplikátory zbývá zadat dobu trvání terapie. Tato doba je společná pro oba aplikátory, proto je mimo smyčku, ve které probíhalo předchozí nastavování. Zadávání doby terapie je v podstatě shodné s předchozím nastavováním parametrů.

Následně se funkcí `I2C.init()` inicializuje komunikační periférie MCU a pomocí funkcí `I2C.write_32bit()` se generujícím MCU pošlou dvaatřicetibitové, reálné hodnoty které jsou nezbytné pro generování impulzů. Tyto parametry jsou řídicí slovo (FSW), doby trvání mezery mezi impulzy (cas) a hodnota, kterou se bude dělit výstupní hodnota každého vzorku (del). Doba trvání terapie se převede na kód BCD pomocí funkce `char_2_BCD()` a MSB výsledného čísla se nastaví na jedničku. Nastavení tohoto bitu znamená v externím obvodu reálného času povolení generování alarmu po uplynutí určitého počtu minut. Výsledná hodnota je pak vstupem funkce `RTC.init()`, která inicializuje externí RTC. Následuje volání funkce `LM92.init()`, která inicializuje teplotní čidlo a má jako vstupní parametr proměnnou `T_crit`. `T_crit` vyjadřuje kritickou hodnotu teploty tkáně pacienta, při jejímž překročení bude teplotní čidlo generovat alarm. Tato teplota byla stanovena na 43°C a pro uložení do registru v teplotním čidle musí být podělena hodnotou 0,0625, což je jeho teplotní rozlišovací schopnost.

Program se následně zacyklí do dvou smyček `while`, ve kterých se čeká na potvrzení dokončení výpočtů od generujících MCU. Toto potvrzení je představováno vysokými úrovněmi na pinech PTF1 a PTF4. Je povoleno externí přerušení, vymaže se displej a zapíše se na něj veličiny které se budou každou vteřinu terapie aktualizovat. Těmito veličinami jsou doba trvání, teplota tkáně pacienta a aplikovaná terapeutická dávka. Dále se pomocí I²C sběrnice definuje v externím RTC zdroj hodinového signálu, čímž se vlastně tento čítač spustí. Algoritmus komunikace po I²C bude detailně popsán v kapitole 6.1.2. Spuštění terapie se dokončí vysláním startovacích impulzů pro generující MCU (v podobě vysokých úrovní na pinech PTF5 a PTF0) a povolením všech přerušení.

Po odstartování terapie se program zacyklí v nekonečné smyčce, ve které se v každém průchodu nastavuje interní čítač reálného času tak, aby generoval přerušení po uplynutí 1,024 s. Fakt, že je čítač povolován v každém průchodu cyklem, nemá na jeho funkci žádný vliv. Před uzavřením smyčky je nutné resetovat ochranný čítač watchdog funkcí `_RESET_WATCHDOG()`. Tato funkce je do kódu vkládána i na místa, kde by mohlo jeho přetečení watchdog způsobovat nechtěný reset systému.

6.1.2 Popis podprogramů a obsluh přerušení

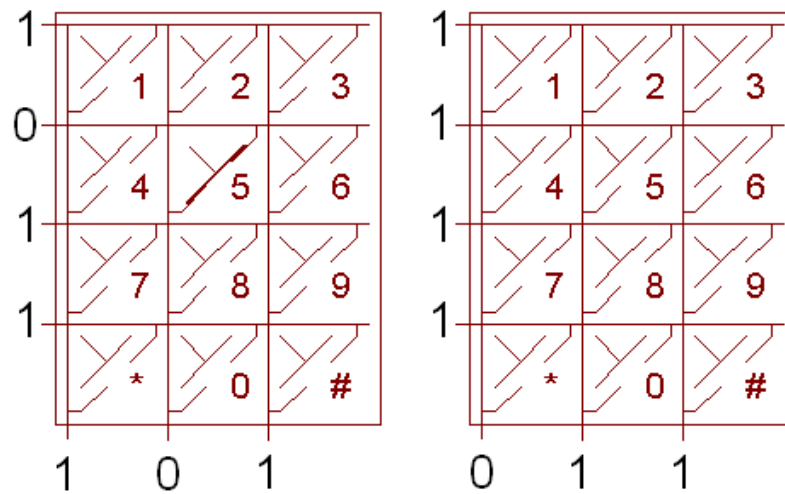
Funkce `BCD_2_char()` zajišťuje převod z číselného vyjádření ve formátu BCD na dekadické číslo. Vstupní proměnná bitově posune o čtyři pozice vlevo a uloží se do dočasné proměnné. Tím se v podstatě na nižší nibl zapsal vyšší nibl vstupní proměnné a následně se vynásobí konstantou 10. Momentálně je tedy v této proměnné hodnota „desítek“. K této hodnotě se pak připočte vstupní hodnota s maskovaným vyšším niblem („jednotky“). Funkce nakonec vrací tuto hodnotu.

Opačný postup používá funkce `byte char_2_BCD()`. Vstupní dekadické číslo se vydělí konstantou deset a uloží se do dočasné celočíselné proměnné `temp`. Při vydělení desíti a uložení do celočíselné proměnné se desetinná část výsledku nezaokrouhluje, ale usekává. Proto po vydělení je v proměnné `temp` pouze první číslice dekadické vstupní hodnoty. Tato první číslice se poté bitovým posunem o čtyři místa posune na horní nibl. Vstupní proměnná se pak vydělí modulo deset, čímž se ze vstupního čísla vyselektuje druhá číslice. Ta se přičte k dolnímu niblu proměnné `temp` a funkce tuto proměnnou vrátí.

Pro překopírování čísla z formátu `float` a rozdělení na čtyři proměnné typu `char` slouží do funkce `sluc_float`. K bitovému překopírování hodnoty z formátu s plovoucí řádovou čárkou do celočíselného formátu je použita unie `float_2_long`. Vstupním argumentem funkce je hodnota ve formátu s plovoucí řádovou čárkou. Následně se musí přiřadit jméno unii. Toto jméno bude součástí jména proměnných v unii. Unie je pojmenována `fp32`, následně proměnná typu `float` bude pojmenována `fp32.fp` a proměnná typu `long` je pojmenována `fp32.uint`. Vstupní proměnná je pak vložena do unie a následně je z unie vyčtena bitově stejná hodnota typu `long`. Tuto proměnnou již lze bitově posouvat a tudíž může být po osmi bitech rozdělena do globálního pole `unst_chr[]`.

Podprogram `zpozdeni_2s()` zajišťuje zpoždění výkonu programu přibližně o 2 s. Funkce nemá žádný vstup a jako první instrukce se vykoná reset ochranného čítače. V cyklu `for` se pak osmdesátkrát zopakuje následující sekvence příkazů. Resetuje se ochranný čítač, zavolá se funkce `Cpu_Delay100US()` se vstupním argumentem 250 a znovu se resetuje watchdog. Funkce `Cpu_Delay100US()` zprostředkuje zpoždění výkonu programu o zadaný počet mikrosekund. Jedná se o funkci, která je generovaná utilitou `Processor Expert`. Zde však `Processor Expert` využíván není, proto je kód funkce zkopírován a funkce je definována stejně jako ostatní.

Podprogram `sken` provádí tzv. skenování maticové klávesnice. Piny MCU, které jsou připojeny k diodám (`PTD0` až `PTD2`) jsou definovány jako výstupní a piny připojené k pull up rezistorům jako vstupní (`PTB0` až `PTB3`). Směr komunikace jednotlivých pinů je nastaven voláním funkce `MCU_init`. Na výstupní piny jsou cyklicky vysílány vektory hodnot `[0,1,1]` až `[1,1,0]`. V jednom okamžiku je tedy na jednom



(a) Klávesnice při stisku klávesy 5 (b) Klávesnice bez stisku klávesy

Obr. 6.1: Vyhodnocování stisku tlačítka u maticové klávesnice

sloupce nulové napětí. Pokud je v tomto sloupci stisknuto tlačítko, logická nula se dostane na vstupní pin MCU, který je připojen k danému řádku. Na obrázku 6.1(a) je vyobrazena situace při stisku klávesy 5 a stav, kdy není stisknuta žádná klávesa je na obrázku 6.1(b). Podprogram sken se skládá ze tří podobných částí, ve kterých se skenuje pokaždé jiný sloupec, proto zde bude vysvětlen algoritmus pouze pro první sloupec. Na dolní nibl potru D se zapíše binární vektor $[0,1,1,0]$, což znamená 06_H . Podmínkou if se rozhodne, zda se na vstupní port dostala nula, což nastane pokud na portu B není hodnota $0f_H$. V další podmínce se rozhoduje, jakou hodnotu vrátí funkce antizakmit. Pokud funkce vrátí logickou jedničku, je stisk klávesy potvrzen, a pokračuje se v rozpoznávání, která klávesa byla stisknuta. Pokud jedna z podmínek if nebude splněna, přejde se na skenování dalšího sloupce. Rozpoznávání se provede pomocí přepínače switch, ten má vstupní proměnnou kod_klavesy. Podle hodnoty proměnné kod_klavesy (pozice nuly na vstupních pinech portu B) se přidělí kódu klávesy její hodnota, funkce sken tuto hodnotu vrátí a ukončí se. Pokud příkaz switch skočí na řádek default, znamená to, že bylo stisknuto více kláves najednou. V tomto případě se proměnná kontrola nastaví na 1, na LCD se zapíše varování, ať obsluha mačká vždy pouze jedno tlačítko. Výkon programu se zpozdí přibližně o 2 sekundy a příkazem return se funkce sken ukončí, aniž by vrátila jakoukoliv hodnotu.

Při stisku tlačítka dochází k zákmitům kontaktů, což může vyvolávat problémy

při detekci stisku klávesy. Tento problém je vyřešen pomocí funkce `antizakmit`, která se volá z funkce `sken`. Funkce `antizakmit` také řeší případné falešné detekce stisku. Do proměnné `dolni_nibl1` se pomocí násobení s maskou zapíše pouze dolní nibl portu B. Totéž se provede zpoždění výkonu programu o 50 ms a výsledek se zapíše do proměnné `dolni_nibl2`. Pokud jsou obě hodnoty shodné, byla stisknuta klávesa, její hodnota se zapíše do proměnné `kod_klavesy` a funkce vrátí hodnotu 1. Pokud se obě proměnné liší, funkce vrátí nulu.

Pomocí funkce `I2C_init` se inicializovala periférie pro komunikaci po sběrnici I²C. V této funkci se nejprve nastaví rychlost komunikace (baud rate) a doba zpoždění mezi sestupnou hranou signálu SCL a změnou na datové lince SDA (SDA hold time). Nastavení těchto veličin probíhá v registru `IIC1F`, který je rozdělen do dvou částí `MULT` (2 bity) a `ICR` (6 bitů). Pro komunikaci požadujeme přenosovou rychlost 250 kbps. Zvolíme si hodnotu `MULT = 1` a pro tuto hodnotu se podle [2] zapíše do části `MULT` registru `IIC1F` 0b00. Podle vzorce 6.2 převzatého z [2] se vypočítá hodnota SCL divider, která se podle tabulky z [2] převede na číslo, které se zapíše do části `ICR` registru `IIC1F`.

$$SCL_divider = \frac{bus_speed}{MULT \cdot baud_rate} = \frac{10000000}{1 \cdot 250000} = 40 \quad [-, Hz, -, bps] \quad (6.2)$$

Hodnota SCL divider vyšla 40 a podle výše zmíněné tabulky se do části `ICR` registru `IIC1F` zapíše hodnota 0x0b a hodnota SDA hold value je 9. Podle vzorce 6.3 převzatého z [2] se vypočítá čas SDA hold time.

$$SDA_hold_time = \frac{SDA_hold_value}{bus_speed} = \frac{9}{10000000} = 0,9 \quad [\mu s, -, Hz,] \quad (6.3)$$

Hodnota SDA hold time musí být přibližně shodná s polovinou periody komunikační rychlosti I²C. Hodnota 0,9 μs tuto podmínku splňuje. Následně se povolí periférie I²C a prozatím se nastaví mód SLAVE.

Funkce `I2C_write_RTC()` zde může sloužit jako ukázková funkce, na které bude vysvětlen princip vysílání dat po sběrnici I²C. Ostatní funkce, které vysílají data po lince I²C, jsou velmi podobné, proto budou při jejich popisu uvedeny pouze odlišnosti a specifiky těchto funkcí. Vstupními atributy funkce jsou adresa registru v obvodu reálného času a data, která se do tohoto registru mají zapsat. Obě vstupní proměnné jsou osmibitové. Vlastní komunikace probíhá v nekonečné smyčce, která končí příkazem `break` tehdy, pokud komunikace proběhla korektně. Na začátku cyklu se provedou nastavení v registru `IIC1C` (kontrolní registr). Periférie se převede z režimu SLAVE do režimu MASTER, čímž se na sběrnici generuje signál START a nastavením příslušného bitu se nastaví vysílací mód. Do datového registru se zapíše adresa zařízení, se kterým se bude komunikovat. V cyklu `while` se čeká na odeslání adresy, které je signalizováno nastavením příznaku TCF (transfer complete flag) v I²C stavovém registru. Následuje zpoždění výkonu programu pomocí

cyklu for. Toto zpoždění slouží k tomu, aby zařízení SLAVE stačilo generovat signál acknowledge (případně ne acknowledge). Přijetí acknowledge se projeví nastavením příznaku RXAK (receive acknowledge). Tento příznak se testuje v následující podmínce if, pokud SLAVE generuje acknowledge zapíše se do datového registru hodnota vstupní proměnné, která reprezentuje adresu cílového registru v RTC. Poté se opakuje čekání na odeslání adresy, zpoždění a kontrola příchodu signálu acknowledge. Totéž se pak opakuje pro odeslání dat, které se budou zapisovat do adresovaného registru. Pokud při každém odeslání dat přešel signál acknowledge, proběhl přenos v pořádku, přepnutím do SLAVE módu generuje MCU signál STOP, realizuje se zpoždění, resetuje se ochranný čítač a funkce končí. Pokud při odesílání jakéhokoliv slova SLAVE negeneruje acknowledge, cyklus se neukončí, ale probíhá celý znovu.

Funkce I2C_write_LM92() slouží pro zápis hodnoty do registru v teplotním čidle LM92. Vstupními atributy jsou osmibitová adresa registru a šestnáctibitová data, která mají být do registru zapsána. Při zapisování do teplotních registrů LM92 se nejdříve posílá významnější Byte vstupního čísla. Údaj o teplotě má 13b, přičemž tři nejnižší bity nejsou v registru využívány. Datový registr I²C je osmibitový a při vkládání šestnáctibitové proměnné do osmibitového registru se překopírují pouze nižší byte šestnáctibitového čísla. Proto se v prvním kroku vysílání dat musí vstupní atribut bitově posunout o pět pozic vlevo. Při posílání spodního Byte vstupního atributu se tento atribut posune o tři bity vpravo. Tímto posunem se na tři nejméně významné bity zapíše nula, což ovšem nevadí, protože tyto bity nejsou v interním registru LM92 využity.

RTC_init() slouží k inicializaci teplotního čidla. V obvodu reálného času se adresa registru, do kterého se bude zapisovat, automaticky inkrementuje po každém zápisu, nebo čtení. První dva registry jsou konfigurační. V prvním se nastaví bit STOP, čímž se RTC zastaví. Ve druhém registru se povolí alarm při přetečení čítače a současně se povolí generování nízké úrovně na výstupním pinu, pokud přetekl čítač. Pomocí výše popsané funkce I2C_write_RTC() se nastaví alarmový registr pro minuty na požadovanou dobu terapie a v registru s adresou 0x0d se nastaví generování signálu s periodou 1 s na pinu CLKOUT. Po ukončení této funkce je RTC připraven k použití, ale prozatím má zastaven zdroj hodinového signálu.

LM92_init() je funkce, která po sběrnici I²C nastaví teplotní čidlo, tak aby správně fungovalo během terapie. Vstupem funkce je číselná hodnota kritické teploty, která byla podělena konstantou 0,0625. Tímto dělením se dosáhne vyjádření, které je potřebné pro uložení čísla do teplotního registru v LM92. V prvním kroku funkce se vypočítá horní mez pro komparátor. Teplotní komparátor se v přístroji nepoužívá, má ale nastavenou horní a dolní mez na 64° C a 10° C. Nastavení dolní meze je v pořádku, ale horní mez musí být nižší, než kritická teplota. Kritická teplota je nastavena na 43° C a proto se horní mez komparátoru nastaví na teplotu

o 20° C nižší, než je kritická teplota. Na rozdíl od RTC se u LM92 vnitřní ukazatel na registry po každém čtení, nebo zápisu neinkrementuje. Je tedy vždy nutné posílat po adrese součástky i adresu registru, se kterým chceme komunikovat. V prvním kroku inicializace se nastaví konfigurační registr. V tomto registru se povolí pouze bit Fault Queue, ostatní bity zůstanou v původním nastavení. Povoláním tohoto bitu se zajistí, aby nedocházelo ke zbytečnému resetu systému například při chybném změření teploty. Aby se na výstupu T_CRIT_A objevila nízká úroveň musí být čtyři po sobě následující změřené teploty vyšší, než je kritická teplota. Následuje zápis do teplotních registrů. Pomocí funkce I2C_write_LM92 se odešle kritická teplota, teplota horní meze komparátoru a hystereze všech komparačních úrovní. Hystereze slouží k určení teploty při které se komparátor znovu překlopí do vysoké úrovně po překročení meze (tato teplota se rovná rozdílu mezi komparační úrovní a hysterezí). Aby bylo snadnější načítání aktuální teploty tkáně z LM92, nastaví se ukazatel na registry na teplotní registr. Při vyčítání tak odpadne nutnost pokaždé posílat adresu interního registru.

Obsluha přerušení od interního čítače reálného času `_interrupt void isrVrti()` je zavolána pouze, pokud po čase delším než jedna vteřina nepřišlo přerušení od externího RTC. V obsluze se inkrementuje proměnná `chybove_pocitadlo` a pokud je tato proměnná rovná třem, zacyklí se výkon programu v nekonečné smyčce, ve které se čeká na přetečení ochranného čítače watchdog a tím i na reset celého systému. Pokud je `chybove_pocitadlo` menší než tři, resetuje se watchdog, a obsluha končí.

Obsluha externího přerušení `_interrupt void isrVirq ()`, které je generováno každou jednu sekundu obvodem reálného času, začíná resetováním watchdog a proměnné `chybove_pocitadlo`. Zastaví a vynuluje interní čítač reálného času a z teplotního čidla se načte aktuální teplota tkáně pacienta. Při vysílání adresy LM92 pracuje periférie I²C v režimu transmit. Po odeslání adresy se v konfiguračním registru přepne periférie do režimu receive a povolí se generování signálu acknowledge. Program čeká na dokončení přenosu dat, které se posléze přepírují do proměnné `RD_data`. LM92 posílá jako první vyšší bajt teplotního registru, proto se proměnná `RD_data` bitově posune o osm bitů vlevo. Po vygenerování acknowledge čidlo bezprostředně posílá nižší bajt dat. Po dokončení příjmu nižšího bajtu se tyto data přičtou k proměnné `RD_data`, zakáže se generování acknowledge a generuje se signál STOP. Jak bylo uvedeno výše, poslední tři bity v teplotním registru LM92 jsou rezervovány, proto se přijatá data musí bitově posunout o tři pozice vpravo. Výsledný údaj o teplotě dostaneme vynásobením rozlišovací schopností teplotního čidla, tedy hodnotou 0,0625. Následně se bude z RTC načítat doba uplynulá od začátku terapie. V prvním kroku komunikace se nastaví interní ukazatel na registry na registr, který odpočítává sekundy. Při této komunikaci MCU do LM92 zapisuje, a proto je periférie I²C ve vysílacím režimu. Při další komunikaci s RTC už MCU pracuje v režimu příjmu,

načte se hodnota sekund a následně minut. Obě načtené hodnoty se převedou na dekadické číslo funkcí `BCD_2_char` a zapíše se na předpřipravenou pozici na LCD. Dále se tyto hodnoty vyjádří v hodinách (minuty se dělí šedesáti a sekundy číslicí 3600), výsledky se převedou na datový typ `float` a přičtou se do proměnné `time`. V této proměnné je tedy celkový čas uplynutý od počátku terapie vyjádřený v hodinách. Z načteného údaje o teplotě se převodem na datový typ `char` vyselektuje celá část, která se od tohoto údaje odečte. V proměnné `teplota` je tedy pouze desetinná část údaje o teplotě. Tato desetinná část se vynásobí koeficientem 100, převede se na datový typ `char` a uloží se do proměnné `teplota_desetinna`. Touto operací se omezila přesnost měření teploty na dvě desetinná místa. Celá i desetinná část změřené teploty se zobrazí na příslušných pozicích na LCD. Jako poslední sledovaná veličina se zobrazuje aplikovaná terapeutická dávka. Ta se vypočítá vynásobením proměnné `time` s předpřipravenou proměnnou `davka1(2)` a převedením na datový typ `integer`. Na konci obsluhy externího přerušení se resetuje ochranný čítač a smaže se příznak tohoto přerušení.

6.2 Programy pro generující MCU

6.2.1 Popis hlavního programu

Program začíná vložením externích hlavičkových souborů, které jsou vytvořeny utilitou `Processor Expert`. Dále jsou definována makra `lowbyte` a `highbyte`. Makro `lowbyte` slouží k zapsání spodního Byte šestnáctibitové proměnné do osmibitové proměnné. Jedná se o prosté převedení datových typů na typ `unsigned char`². Makro `highbyte` uloží do výsledné osmibitové proměnné druhý Byte vstupní proměnné. Vstupní hodnota se nejprve převede na typ `unsigned integer` (čili na 16b), poté se bitově posune o osm pozic doprava a následně se proměnná převede na typ `unsigned char`.

Dále následuje deklarace a definice proměnných, které budou potřebné pro běh programu. Pole `signal` se vyskytuje pouze jako deklarace, jedná se však pouze zkrácený zápis, správně by proměnná měla být definována jako pole 201 vzorků generovaného signálu. První vzorek je záměrně nulový³ a následuje 200 vzorků, které tvoří jednu periodu zvlnění impulsu. V programu je použito několik funkcí, které jsou vygenerovány utilitou `Processor Expert`. Jména použitých funkcí a stručný popis jejich funkce je uveden v tabulce 6.3 a v následujícím textu už nebudou zmiňovány.

²Převod na daný datový typ se provede zapsáním typu do závorek před proměnnou

³Vzorek s indexem nula se nikdy generovat nebude

Tab. 6.3: Popis funkcí generovaných Processor Expert

Jméno funkce	Vstupní atributy	Popis funkce
PE_low_level_init	void	Inicializace jádra MCU
Cpu_Delay100US	integer	Zpozdí průběh programu o daný počet stovek mikrosekund
Frekvence_filtr_SetPeriodReal()	float	Nastaví periodu signálu na výstupním pinu časovače
Frekvence_filtr_Enable()	void	Povolení generování signálu na výstupním pinu časovače
Potvrzeni_out_PutVal()	bool	Zapíše logickou jednobitovou hodnotu na výstupní pin

Následuje přijetí a zpracování parametrů terapie po sběrnici I²C. Přijaté parametry se uloží do proměnných FSW, cas a del. Přijatá proměnná cas vyjadřuje časovou mezeru mezi impulzy v sekundách. Aby mohla být vstupem funkce Cpu_Delay100US, musí být vyjádřena ve stovkách mikrosekund a převedena na celé číslo, což se provede vydělením konstantou 0.0001 a převedením na datový typ integer. Výsledek je pak uložen do proměnné zpozdeni. Dále se ze slova FSW vypočítá šířka generovaných impulzů. Výpočet se provede dosazením do vzorce 3.1, inverzí výsledku a jeho převodem na milisekundy. Vypočítaná šířka impulzů je pak vstupem funkce mezni_frekvence(), která pro zadanou šířku vypočítá mezní frekvenci analogového filtru. Signál, kterým se mezní frekvence řídí, musí mít, podle vzorce 4.3, stokrát vyšší frekvenci, než je požadovaná mezní frekvence a musí být vyjádřen v sekundách. Toho se dosáhne vynásobením hodnotou 100 000 a následnou inverzí hodnoty.

Do proměnné generator_adresy se zapíše hodnota slova FSW a následuje cyklus while, během kterého se vyplní pole indexy a pole opakovani. Cyklus se bude opakovat, dokud proměnná generator_adresy nepřekročí desetinu své velikosti (indexy se budou vytvářet pouze pro první periodu, pokud by se vytvářely pro celý impuls probíhal by cyklus do přetečení generátoru adresy). Proměnná velikost_generatoru je typu unsigned long, proto se výsledek dělení číslem 10 musí převést na typ float, aby nedošlo k nechtěnému zaokrouhlení. Vypočítá se index_vzorku, což je pořadové číslo vzorku v poli signal. Vypočítá se podíl velikosti generátoru (65 536) a počtu vzorků, které se generují (2000). Tímto podílem se dělí obsah proměnné generator_adresy a vyjde index_vzorku v podobě racionálního čísla. Tento výpočet se dá chápat tak, že proměnná generator_adresy se rozdělí na ekvidistantní části, jejichž počet odpovídá počtu generovaných vzorků a každé této části je přiřazen určitý index. Index vzorku

musí být přirozené číslo (indexuje se jím pole vzorků), proto se upraví voláním podprogramu `ceil1`. Jestliže se vygenerovaný `index_vzorku` rovná `indexu`, který je zapsán v poli `indexy` na pozici `counter` (předchozí vygenerovaný vzorek), inkrementuje se hodnota pole `opakovani` na pozici, která se rovná proměnné `counter`. Pokud podmínka neplatí, inkrementuje se proměnná `counter`, a do proměnné `indexy` na pozici, která je udána inkrementovanou proměnnou `counter`, se zapíše index generovaného vzorku. Poté se do proměnné `generator_adresy` přičte konstanta `FSW` a cyklus se opakuje.

V následujícím `for` cyklu se nahradí `indexy` vzorků v poli `indexy` konkrétními hodnotami vzorků z pole `signal`. Nulový index se vyplní posledním vygenerovaným vzorkem. Je to kvůli modulárnímu dělení při vyčítání vzorků. Do proměnné `modulo` se uloží proměnná `counter`, ta teď ukazuje počet různých vzorků vygenerovaných během jedné periody. Následně se vypočítá celkový počet různých vzorků, které budou vygenerovány během trvání jednoho impulsu. Pomocí příslušných funkcí se nastaví a povolí generování časovače, který bude generovat pravoúhlý signál zadané periody na pinu `PTD3`. Generující MCU je nyní připraven ke generování, proto odešle potvrzovací signál řídicímu MCU zavoláním funkce `Potvrzeni_out_PutVal(1)`. Pomocí funkce `DAC_ILE_SetVal()` se na pin `PTB2` zapíše vysoká úroveň a tím se DAC nastaví do režimu „Flow - Through“. Následně se program zacyklí, dokud nepřijde od řídicího MCU startovací signál, který je reprezentován vysokou úrovní na pinu `PTB6`.

Po příjmu startovacího signálu od řídicího MCU se program zacyklí v nekonečné smyčce a na portu A generuje osmibitové vzorky. Z této smyčky se program dostane pouze po resetu od řídicího MCU. Jeden impuls se vygeneruje v cyklu `for`, který probíhá dokud nebudou vygenerovány všechny různé vzorky za dobu trvání impulsu. Vzorky které máme k dispozici tvoří pouze jednu periodu zvlnění signálu, proto se musí vygenerovat desetkrát. Proměnná `j` se mění od jedné až do celkového počtu různých indexů, proto se pro vygenerování vzorku musí modulárně dělit počtem různých vzorků za jednu periodu, tedy proměnnou `modulo`. Takto je zajištěno, že se bude jedna perioda zvlnění generovat cyklicky. Problém je ovšem s generováním posledního vzorku, jehož index je roven počtu různých vzorků za jednu periodu zvlnění, modulární podíl je tedy roven nule a poslední vzorek by se vlastně nikdy nevygeneroval. Proto byla hodnota vzorku na posledním indexu zapsána na pozici s indexem nula, jak už bylo zmíněno výše. Tato skutečnost nevádí při prvním průchodu programu cyklem, protože cyklus začíná od indexu 1. Hodnota vzorku se vyčte z pole `indexy` a pošle se na port A. Jestliže v proměnné `opakování` na pozici právě vygenerovaného vzorku není nula, znamená to, že se vzorek musí znovu vygenerovat tolikrát, kolikrát ukazuje pole `opakování` na dané pozici. V programu je opětovné generování vyřešeno držením hodnoty vzorku na portu A po daný časový

interval. K realizaci tohoto zpoždění je použit Č/Č1. Frekvence generování vzorků je 100 KHz, což odpovídá periodě $t = 10 \mu s$. Předdělička N pro Č/Č1 je rovna jedné (ponecháno defaultní nastavení PS0 až PS2 na nulové úrovni), frekvence hodinového f_0 signálu MCU je 10 MHz. Ze vzorce 6.4 vychází hodnota modulo registrového páru 100.

$$TPM2MODH : TPM2MODL = \frac{f_0}{N} \cdot t \quad [-, Hz, -, s] \quad (6.4)$$

Tato hodnota se musí vynásobit počtem opakování vzorku a musí se od ní odečíst počet taktů, které budou potřebné pro vygenerování dalšího vzorku od přetečení Č/Č1 (v případě tohoto programu je to 45). Vynuluje se počítací registr, příznak přetečení Č/Č1 a program se zacyklí dokud příznak přetečení nebude jednička. Pokud se vzorek neopakuje a generuje se pouze jednou, proběhne zpoždění pomocí devíti instrukcí nop. Instrukce nop v podstatě nevykoná nic, pouze zajistí zpoždění výkonu programu o jeden hodinový cyklus. Je to instrukce jazyka symbolických adres, čili pro použití v jazyku C musí být uvedena slovem asm. Takto se zajistí, že rozdíl mezi generovanými vzorky bude vždy přesně 100 (nebo násobek sta) hodinových cyklů, což při frekvenci hodinového signálu MCU 10 MHz odpovídá frekvenci generování jednotlivých vzorků 100 KHz. Po vygenerování různého vzorku se inkrementuje proměnná j.

Po vygenerování celého impulzu, se na port A a na pin TPB2 pošle logická nula, čímž se vynuluje generovaný signál a filtr uvede do tzv. Shutdown módu, kdy jeho výstup je ve stavu vysoké impedance. Toto je stav, kdy se signál negeneruje. Následně se průběh programu pozdrží na dobu mezery mezi impulzy, která je uložená v proměnné zpozdeni.

Poté se celá nekonečná smyčka opakuje a začne se generovat nový impulz.

6.2.2 Popis podprogramů a obsluh přerušení

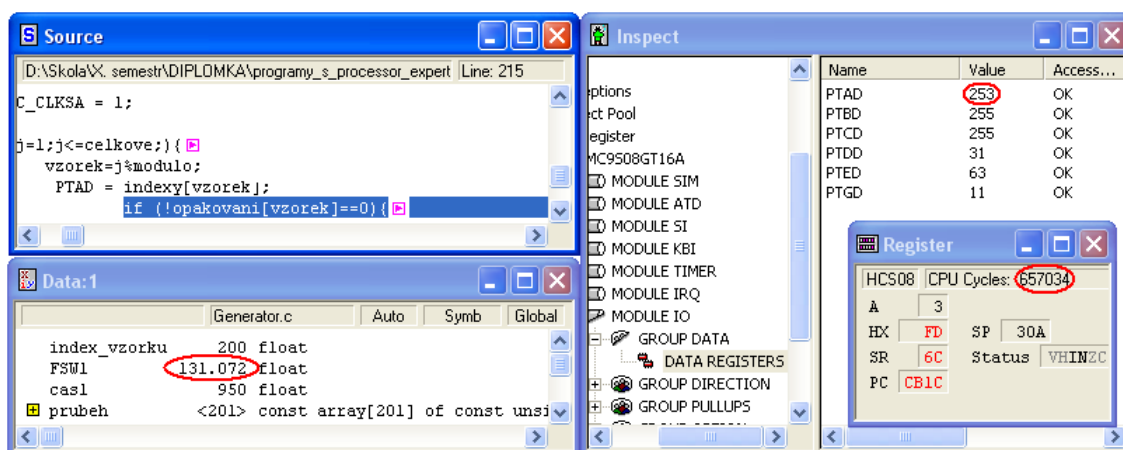
Funkce ceil1 má vstup v podobě racionálního čísla a výstup v podobě přirozeného čísla, jehož celá část je o jedničku větší, pokud je desetinná část nenulová. Po zavolání funkce se vypočítá celá část vstupního racionálního čísla převedením na datový typ unsigned integer. Při tomto převedení datových typů se usekává desetinná část vstupního čísla. Proměnná desetinná.cast se vypočítá odečtením celé části od vstupního čísla. Podmínkou if sr rozhodne, jestli je desetinná část nulová, pokud ano (vstupní číslo bylo celé), program vrátí vstupní hodnotu, pokud ne, k celé části se přičte jednička a tuto hodnotu funkce vrátí.

Pro překopírování čísla z formátu long do formátu float slouží unie long_2_float a funkce sluc_float. Vstupním atributem funkce je číslo ve formátu long. Unii long_2_float se následně přiřadí jméno lng32. V unii je proměnná typu long pojmenována jako

uint. Pokud chceme do unie vložit proměnnou typu long, musíme zapsat jméno unie a za tečku jméno proměnné. Stejně musíme postupovat, pokud z unie budeme chtít překopírovat hodnotu ve formátu float. Po ukončení podprogramu je tedy ve výstupní proměnné typu float bitově překopírovaná vstupní proměnná typu long.

Vstupem funkce `mezni_frekvence` je hodnota šířky generovaných impulzů, vyjádřená v milisekundách, ve formátu float. Vstupní atribut se překopíruje do proměnné `nasobek` a rozhodne se, zda je vstupní proměnná z rozsahu 0,5 až 50 ms. Pokud ano, do proměnné frekvence se překopíruje poslední prvek v poli `koeficienty1`, což je koeficient při nulté mocnině interpolačního polynomu. Následuje cyklus `for`, ve kterém se vypočítá celá funkční hodnota polynomu. K proměnné frekvence se vždy přičte násobek koeficientu polynomu s příslušnou mocninou vstupní proměnné. Na konci cyklu se vždy zvyšuje mocnina vstupní proměnné. Pokud polynomem vypočtená mezní frekvence převyší 45 kHz, je podmínkou `if` omezena na tuto hodnotu. Stejný cyklus výpočtu funkční hodnoty polynomu případně probíhá i pro druhý rozsah vstupní proměnné a funkce pak vrací hodnotu mezní frekvence, na kterou se musí nastavit filtr.

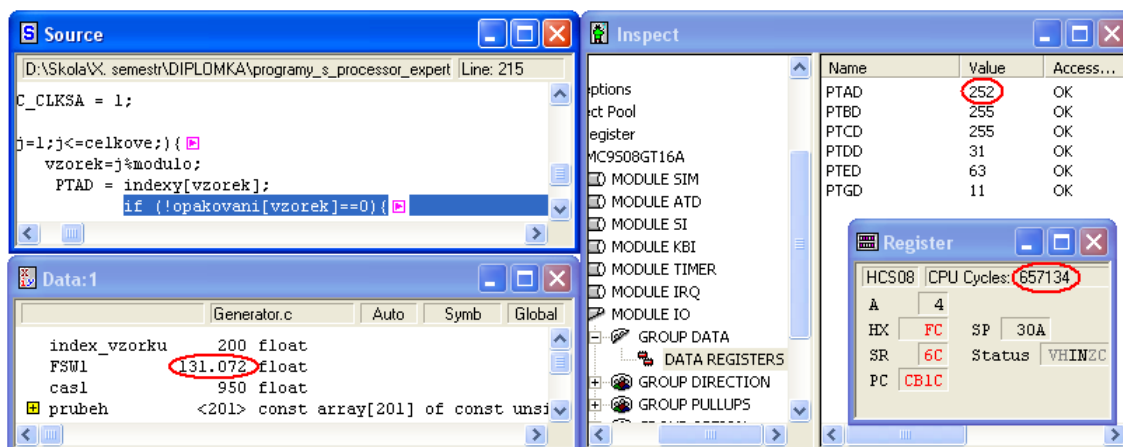
6.3 Simulace částí programů



Obr. 6.2: Stav simulace po vygenerování třetího vzorku signálu

Správnost funkce programů pro mikrokontroléry byla simulována v prostředí True-Time Simulator & Real-Time Debugger. Prostředí je součástí programu Code Warrior a umožňuje prohlížet a měnit obsah registrů, paměti, globálních i lokálních proměnných. Dále umožňuje krokovat průběh programu, prohlížet zdrojový kód přeložený do jazyka symbolických adres nebo simulovat vstupy periférií. Jako ukázk

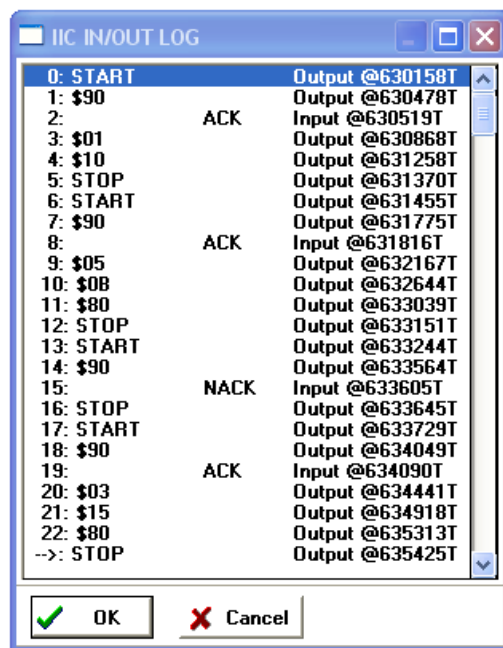
správné funkce programů jsou vybrány dvě simulace, jedna pro generující MCU a jedna pro řídicí MCU.



Obr. 6.3: Stav simulace po vygenerování čtvrtého vzorku signálu

Generující MCU musí na port A generovat vzorky přesně s frekvencí 100 kHz. Přesnost dodržení frekvence generování je důležitá u krátkých impulzů, kde by mohly nepřesnosti způsobovat poměrně velké chyby. Aby bylo možné posoudit přesnost frekvence generování v simulátoru, musí se sousední vzorky signálu vzájemně lišit. Tuto podmínku splňuje například signál o frekvenci 10 Hz a stříde 1:20. Pro tyto vstupní parametry vychází hodnota slova FSW = 131.072, která je vyznačena na obrázcích 6.2 a 6.3 v části Data:1. V části Inspect jsou zobrazeny hodnoty na portech MCU v desítkové soustavě. Hodnota na portu A je zvýrazněna. V poslední části Register je zvýrazněn počet hodinových cyklů MCU od začátku programu. Obrázky 6.2 a 6.3 zachycují průběh simulace bezprostředně po vygenerování třetího (253) a čtvrtého (252) vzorku signálu. V okně Register se odečtením počtu cyklů MCU zjistí, že vygenerování jednoho vzorku proběhlo během 100 hodinových cyklů MCU. Frekvence interní sběrnice MCU je 10 MHz a po vynásobení počtem hodinových cyklů vychází požadovaná frekvence generování vzorků 100 kHz.

True-Time Simulator & Real-Time Debbuger umožňuje simulovat komunikaci po sběrnici I²C, zadávat vstupní data a potvrzovací signály a sledovat výstupní data. Na obrázku 6.4 je simulace komunikace mezi řídicím MCU a teplotním čidlem LM92CIM. Komunikaci začíná MCU událostí START a následuje vyslání SLAVE adresy teplotního čidla. Do čidla se bude zapisovat, proto je poslední bit roven nule a celá adresa je 0x90. Zařízení SLAVE rozpoznalo svou adresu a generuje signál ACK. Dále dochází k odeslání dvou datových slov. Prvním se nastaví ukazatel na registry na hodnotu 0x01 (konfigurační registr) a do tohoto registru se zapíše hodnota 0x10.



Obr. 6.4: Simulace komunikace po sběrnici I²C

Po odeslání dat se komunikace ukončí signálem STOP. Následuje další zápis dat, tentokrát do registru s adresou 0x05. Zápis proběhl korektně a MASTER generuje signál STOP. Při odeslání dalšího datového slova zařízení SLAVE nerozpoznalo svou adresu a na datové lince zůstala vysoká úroveň, což MASTER vyhodnotí jako signál NACK. Po přijetí NACK MASTER generuje signál STOP a pokouší se navázat komunikaci znovu signálem START. Tentokrát SLAVE generuje ACK a komunikace proběhne v pořádku.

7 ZÁVĚR

Ve své diplomové práci jsem se zaměřil na možnosti použití pulzního magnetického pole při léčbě dětské mozkové obrny. Pulzní magnetické pole je schopno interagovat s živou tkání a pokud jsou dodrženy jeho biotropní parametry, má výrazné léčebné účinky. Dále jsem navrhl blokové schéma přístroje pro pulzní magnetoterapii. Blokové schéma tvoří základní koncept přístroje a vysvětlil jsem na něm princip celé jeho funkce. V rámci popisu blokového schématu jsem zejména vysvětlil princip fungování přímé číslicové syntézy, provedl jsem simulaci generovaných impulzů a popsal druhy používaných aplikátorů. Následně jsem navrhl obvodové řešení přístroje, detailně popsal jednotlivé použité součástky i jejich funkci v obvodu. Podle navrženého schématu zapojení jsem sestavil desku plošných spojů. Jelikož součástí systému jsou i tři mikrokontroléry, sestavil jsem pro ně vývojové diagramy řídicích programů a podle nich pak napsal i samotné programy pro mikrokontroléry. Funkčnost napsaných programů jsem ověřoval v simulačním programu.

Při použití vhodných aplikátorů umožňuje navržený přístroj generovat pulzní magnetické pole požadovaných biotropních parametrů. Přístroj umožňuje obsluhu vybrat ze tří přednastavených programů terapie nebo zvolit parametry terapie individuálně. Právě možnost individuálního nastavení parametrů (frekvence, střída, doby trvání terapie a indukce magnetického pole) dovoluje použít pro každého pacienta specifickou terapii, která mu bude co nejvíce prospěšná. Bezpečnost pacienta v průběhu terapie je kontrolována řídicím mikrokontrolérem, který vyhodnocuje rizikové parametry. Mezi rizikové parametry terapie, které jsou kontrolované přístrojem, patří teplota tkáně pacienta a doba trvání terapie. Rizikové parametry jsou v průběhu terapie v pravidelných intervalech (spolu s aplikovanou terapeutickou dávkou) zobrazovány na LCD a po překročení prahových hodnot parametrů je terapie okamžitě zastavena.

Způsob generování vzorků signálu je inspirován teorií přímé číslicové syntézy. Výhoda tohoto přístupu je v tom, že je možné generovat téměř libovolný průběh, je možné měnit frekvenci signálu v širokém rozsahu a signál má (při dostatečném množství vzorků na periodu) malé zkreslení. Jedním z nejdůležitějších parametrů impulzu je strmost jeho náběžné a sestupné hrany. Aby byla strmost co největší, musí být frekvence generování jednotlivých vzorků co nejvyšší. Taktovací frekvence jádra MCU je 10 MHz a požadujeme-li frekvenci generování vzorků 100 kHz, musí se jeden vzorek vypočítat a vygenerovat během 100 hodinových cyklů MCU. Pokud uvážíme výpočetní náročnost počítání s čísly s plovoucí řádovou čárkou, je kontinuální generování neproveditelné. Vzorky pro požadovaný signál jsou tedy před začátkem terapie připraveny tak, aby MCU během terapie pouze posílal vypočítané vzorky na výstupní port, a požadovaná frekvence generování vzorků bude dodržena.

LITERATURA

- [1] BIDMAN, P. BARTUŠEK, K. *Mapování indukce magnetického pole v okolí malých cívkových aplikátorů metodou magnetické rezonance*. [online]. Dostupný z <<http://www.elektrorevue.cz/cz/clanky/>>.
- [2] FREESCALE. *Katalogový list součástky MC9S08AW48*. [online]. Dostupný z <http://www.freescale.com/files/microcontrollers/doc/data_sheet/MC9S08AW60.pdf>.
- [3] FREESCALE. *Katalogový list součástky MC9S08GT16A*. [online]. Dostupný z <http://www.freescale.com/files/microcontrollers/doc/data_sheet/MC9S08GT16A.pdf>.
- [4] FRÝZA, T. *Přednášky předmětu Mikropočítačová technika*. [online]. Dostupný z <<https://www.vutbr.cz/elearning/>>.
- [5] HALLIDAY, D. *Fyzika: Část 3, Elektřina a magnetismus*. Brno: VUTIUM, 2000. 310s. ISBN: 80-214-1868-0
- [6] HRBÁČEK, J. *Komunikace mikrokontroléru s okolím. Díl 1*. Praha: BEN, 1999. 159s. ISBN: 80-86056-42-2
- [7] HRNČÍŘ, K. *O magnetoterapii* [online]. Biomag. Dostupný z <<http://www.biomag.cz/magnetoterapie/>>.
- [8] CHVOJKA, J. *Magnetoterapie v teorii a praxi*. Brno: Professional publishing, 2000. 99s. ISBN 80-86419-01-0
- [9] CHVOJKA, J. *Magnetoterapie, její přednosti a úskalí*. Nový Bydřov: okresní ústav národního zdravotnictví, 1987. 94s.
- [10] CHVOJKA, J. *Možnosti pulzního magnetického pole v dětské neurologii*. [online]. Dostupný z <<http://www.dobromysl.cz/scripts/>>.
- [11] KASAL, M. *Frekvenční syntéza v komunikačních systémech experimentální družice*. Brno: VUTIUM, 2005. 35s. ISBN: 80-214-2982-8
- [12] MATEMATICKO-FYZIKÁLNÍ FAKULTA UK. *Měření stočení polarizační roviny*. [online]. Dostupný z <<http://physics.mff.cuni.cz/vyuka/>>.
- [13] MAXIM. *Katalogový list součástky MAX7421*. [online]. Dostupný z <<http://datasheets.maxim-ic.com/en/ds/MAX7418-MAX7425.pdf>>.

- [14] NATIONAL SEMICONDUCTOR. *Katalogový list součástky DAC0832LCN*. [online]. Dostupný z <<http://www.national.com/profile/snip.cgi/DAC0832>>.
- [15] NATIONAL SEMICONDUCTOR. *Katalogový list součástky LF357*. [online]. Dostupný z <<http://www.national.com/profile/snip.cgi/openDS=LF357>>.
- [16] NATIONAL SEMICONDUCTOR. *Katalogový list součástky LM92CIM*. [online]. Dostupný z <<http://www.national.com/ds/LM/LM92.pdf>>.
- [17] OPTREX. *Katalogový list součástky DMC-20481-NY-LY*. [online]. Dostupný z <<http://www.optrex.co.jp/us/product/catalog/pdf.dmc/20481aze.pdf>>.
- [18] PHILIPS. *Katalogový list součástky PCF8563*. [online]. Dostupný z <<http://www.datasheetcatalog.org/datasheet/philips/PCF8563-02.pdf>>.
- [19] ROZMAN, J. *Elektronické přístroje v lékařství*. Praha:Academia, 2006. 406s. ISBN: 80-200-1308-03
- [20] ŠTORK, M. *Přímá digitální syntéza a její aplikace*. Sdělovací technika, 12/2000. s. 26-28. ISSN: 0036-9942
- [21] THERAPY SYSTEMS. *Magnetoterapie*. [online]. Dostupný z <<http://www.therapy.cz/>>.
- [22] WALEK,P. *Využití pulzního magnetického pole pro léčbu DMO: bakalářská práce*. Brno: FEKT VUT v Brně, 2008. 72s., 5 příl.
- [23] ŽIVNÝ, B. *Dětská mozková obrna*. [online]. Dostupný z <<http://www.mujiweb.cz/zdravi/dmob/>>.

SEZNAM SYMBOLŮ, VELIČIN A ZKRATEK

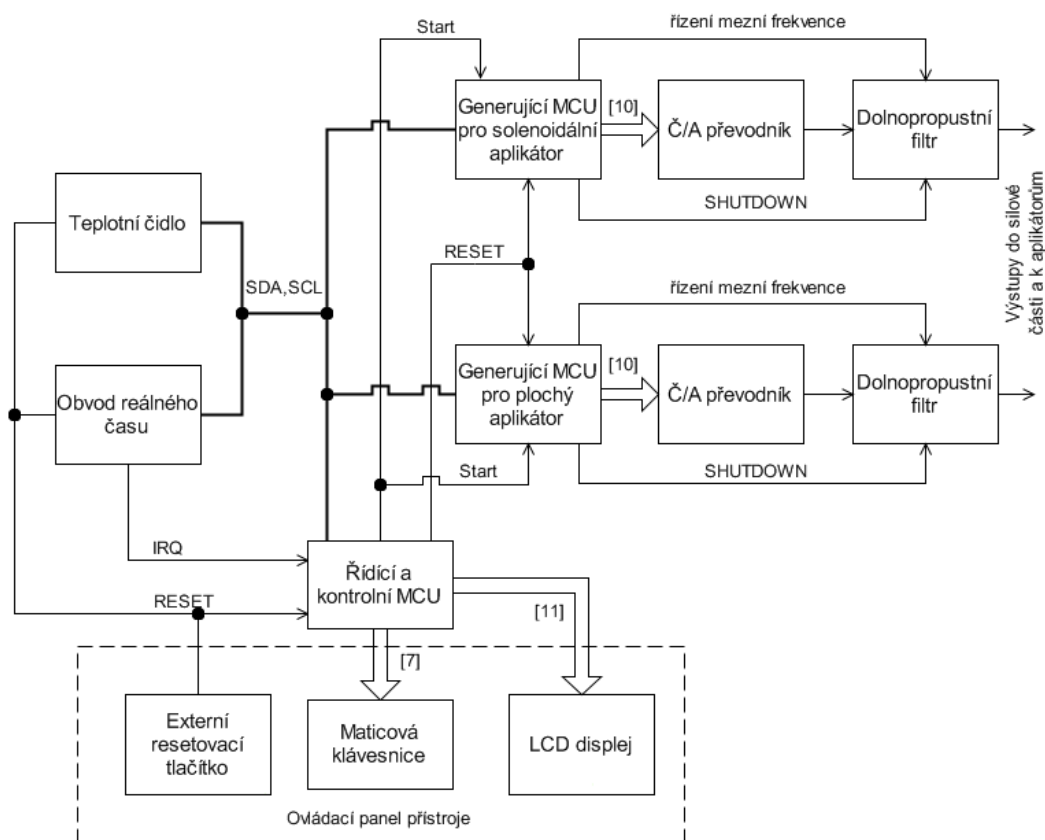
DMO	Dětská mozková obrna
PMP	Pulzní magnetické pole
H	Intenzita magnetického pole
B	Indukce magnetického pole
μ	Permeabilita
μ_r	Relativní permeabilita
EDRF	Faktor vyvolávající zvýšení produkce oxidu dusnatého v těle – Endothel Derived Relaxing Factor
Ca ⁺	Iont vápníku
DAC	Digitálně analogový převodník
LCD	Technologie zobrazování na principu tekutých krystalů – Liquid Crystal Display
f_c	Frekvence hodinového signálu
FSW	Frekvenční šídící slovo – Frequency Setting Word
MCU	Mikrokontrolér – Microcomputer Unit
FLASH	Paměť nezávislá na napájecím napětí s krátkou dobou přístupu
RAM	Paměť s náhodným přístupem – Random-Access Memory
I ² C	Sběrnice pro sériovou komunikaci
RTC	Obvod reálného času – Real Time Counter
BCD	Kód pro dvojkovou reprezentaci desítkových čísel – Binary Code Decial
DC	Střída – Duty Cycle

SEZNAM PŘÍLOH

A	Blokové schéma řídicí, kontrolní a generující části	I
B	Blokové schéma silové části	II
C	Schéma zapojení řídicí, kontrolní a generující části	III
D	Deska plošných spojů	IV
E	Vývojové diagramy programů	VIII
E.1	Vývojový diagram pro řídicí a kontrolní mikrokontrolér	VIII
E.2	Vývojový diagram pro generující mikrokontrolér	XII

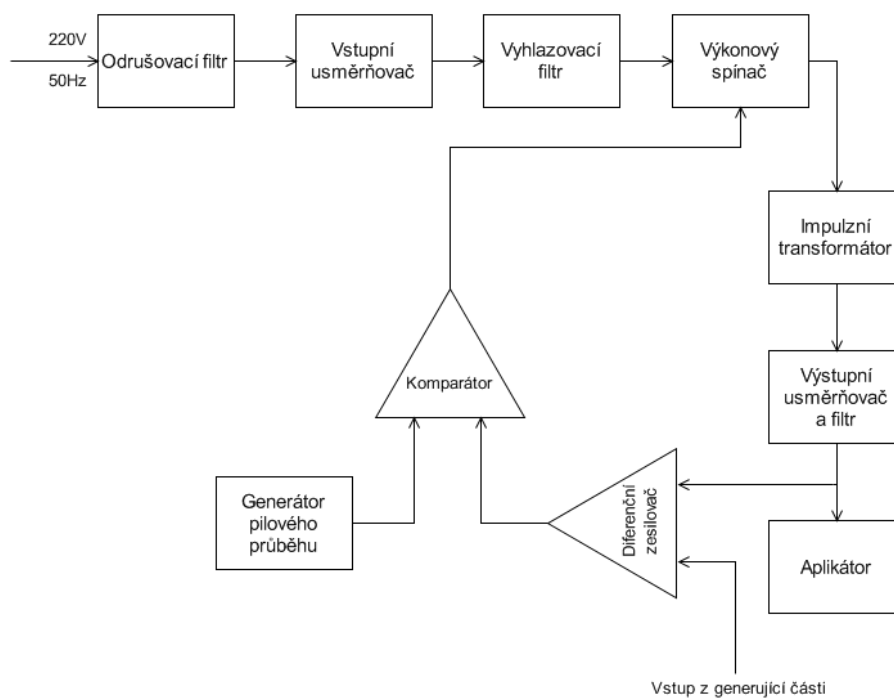
PŘÍLOHY

A BLOKOVÉ SCHÉMA ŘÍDICÍ, KONTROLNÍ A GENERUJÍCÍ ČÁSTI



Obr. A.1: Blokové schéma řídicí, kontrolní a generující části přístroje

B BLOKOVÉ SCHÉMA SILOVÉ ČÁSTI

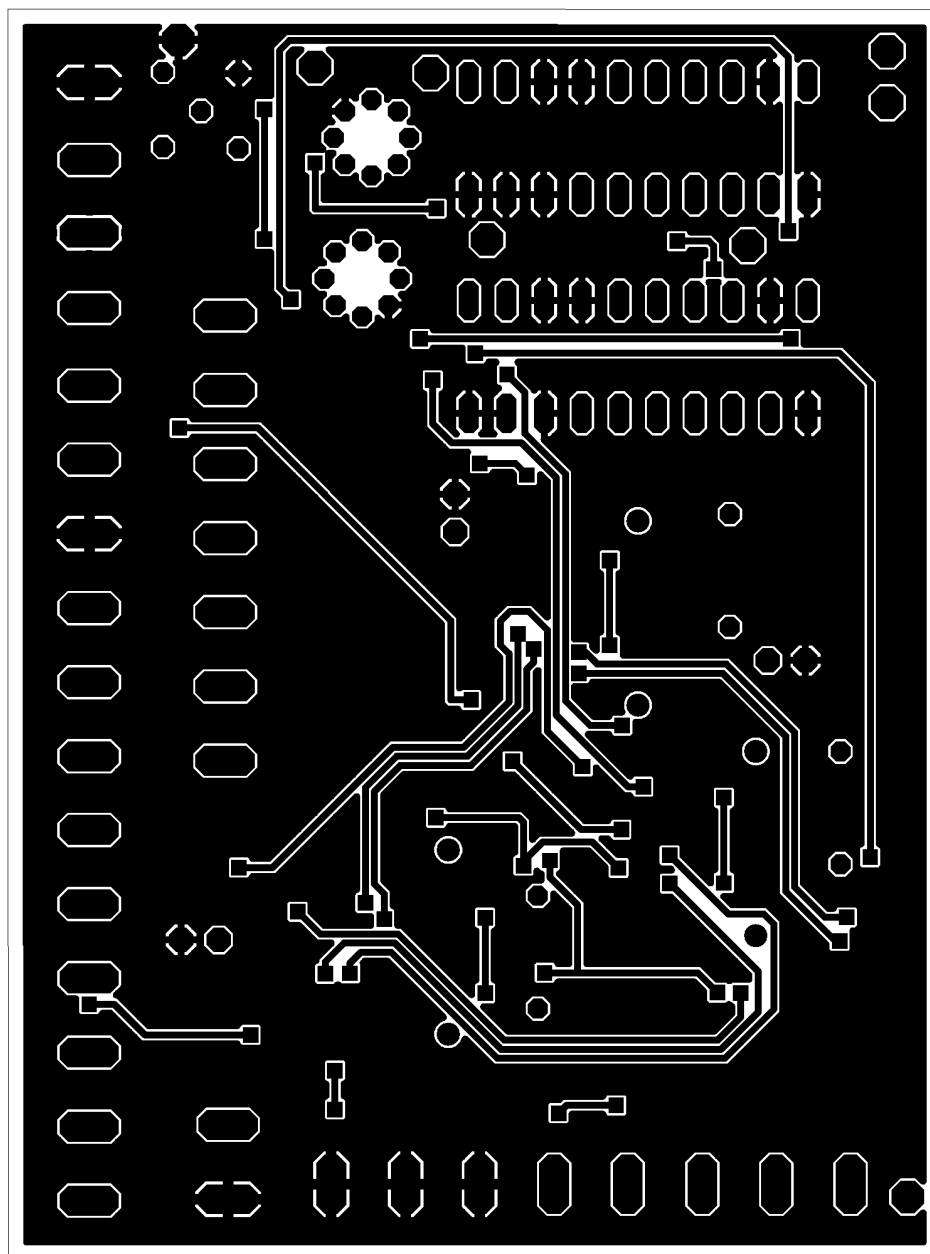


Obr. B.1: Blokové schéma silové části

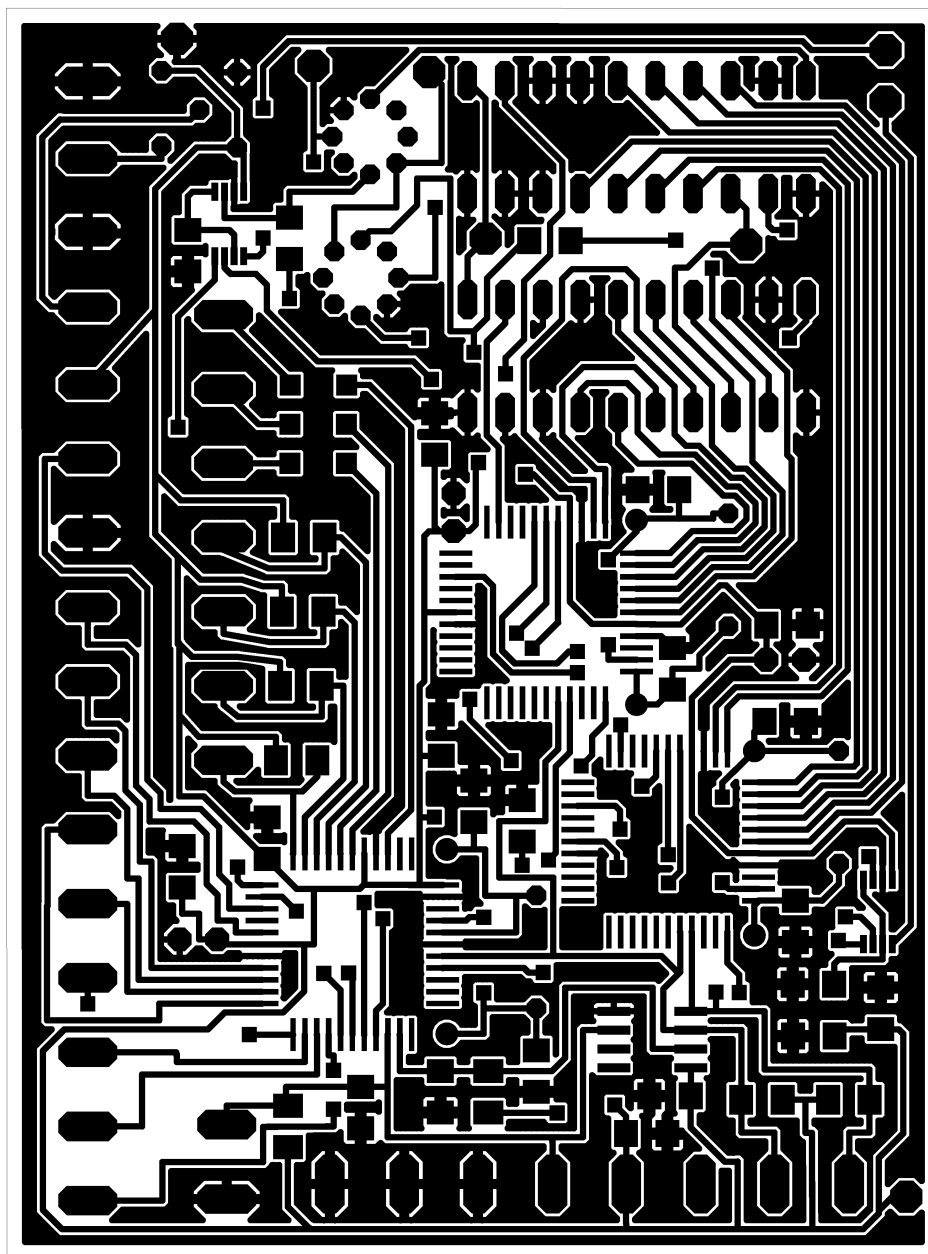
C SCHÉMA ZAPOJENÍ ŘÍDICÍ, KONTROLNÍ A GENERUJÍCÍ ČÁSTI

Obr. C.1: Schéma zapojení přístroje generujícího PMP (bez výkonové části)

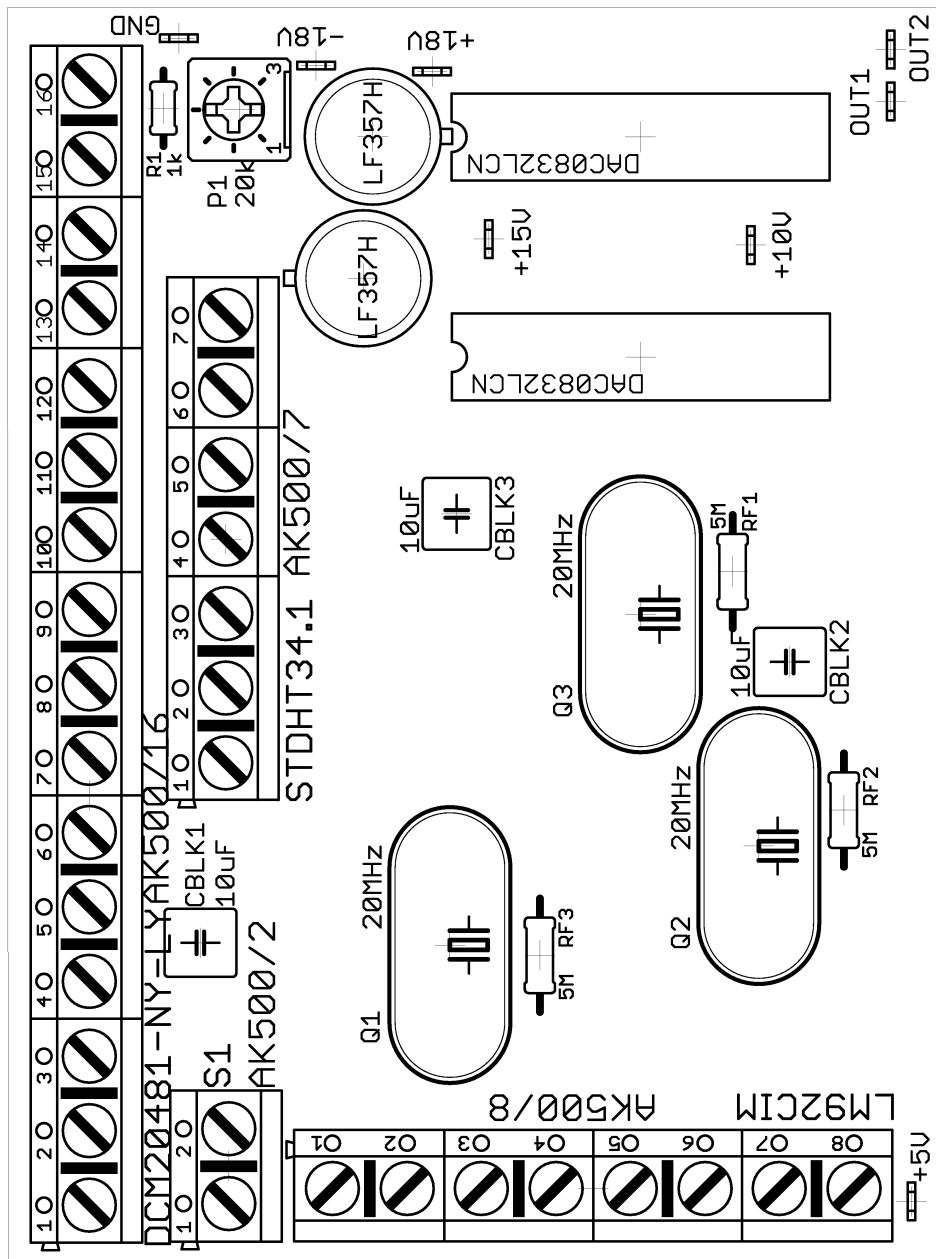
D DESKA PLOŠNÝCH SPOJŮ



Obr. D.1: Matrice ze strany součástek (zvětšeno v poměru 2:1)

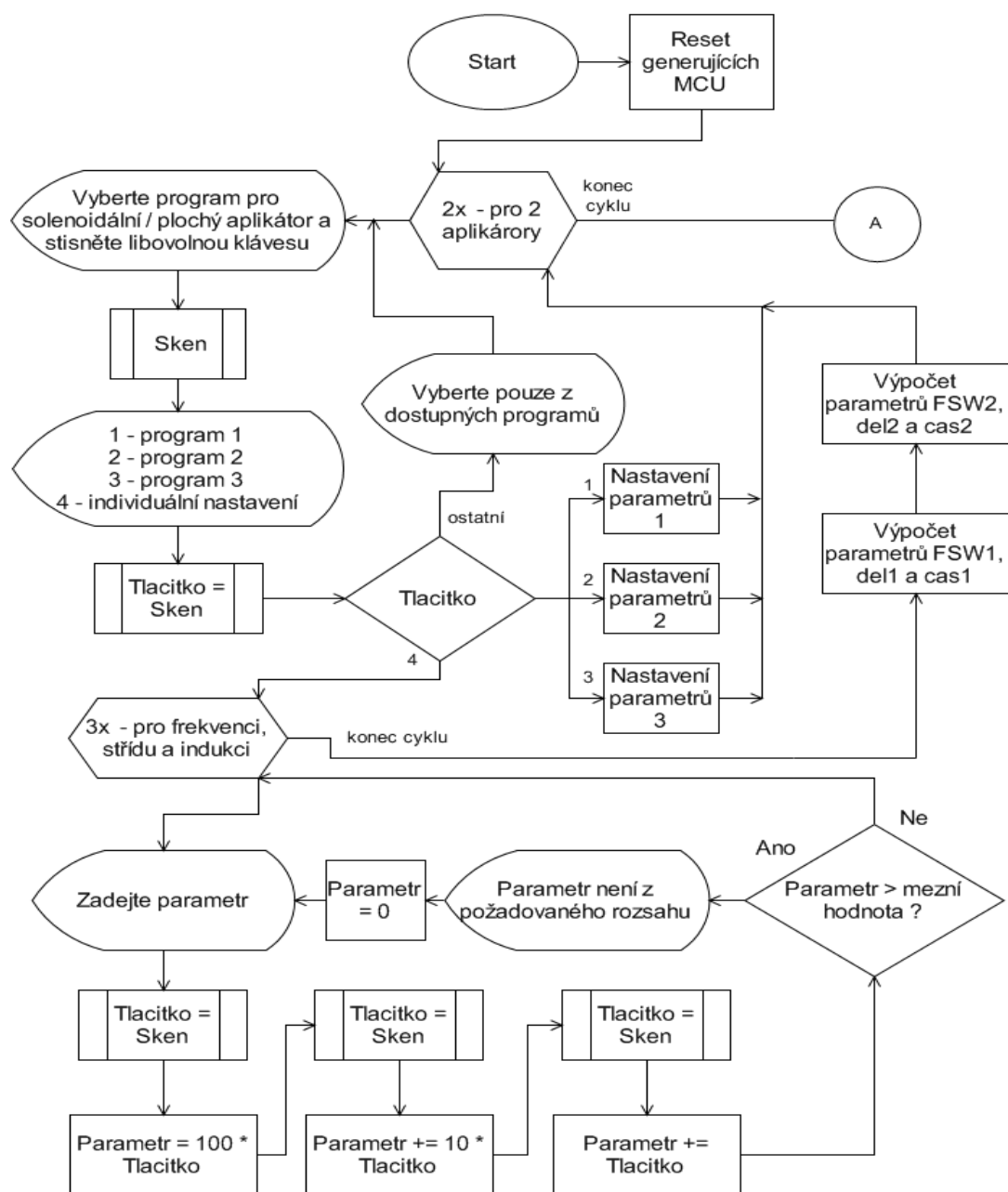


Obr. D.2: Matrice ze strany spojů (zvětšeno v poměru 2:1)

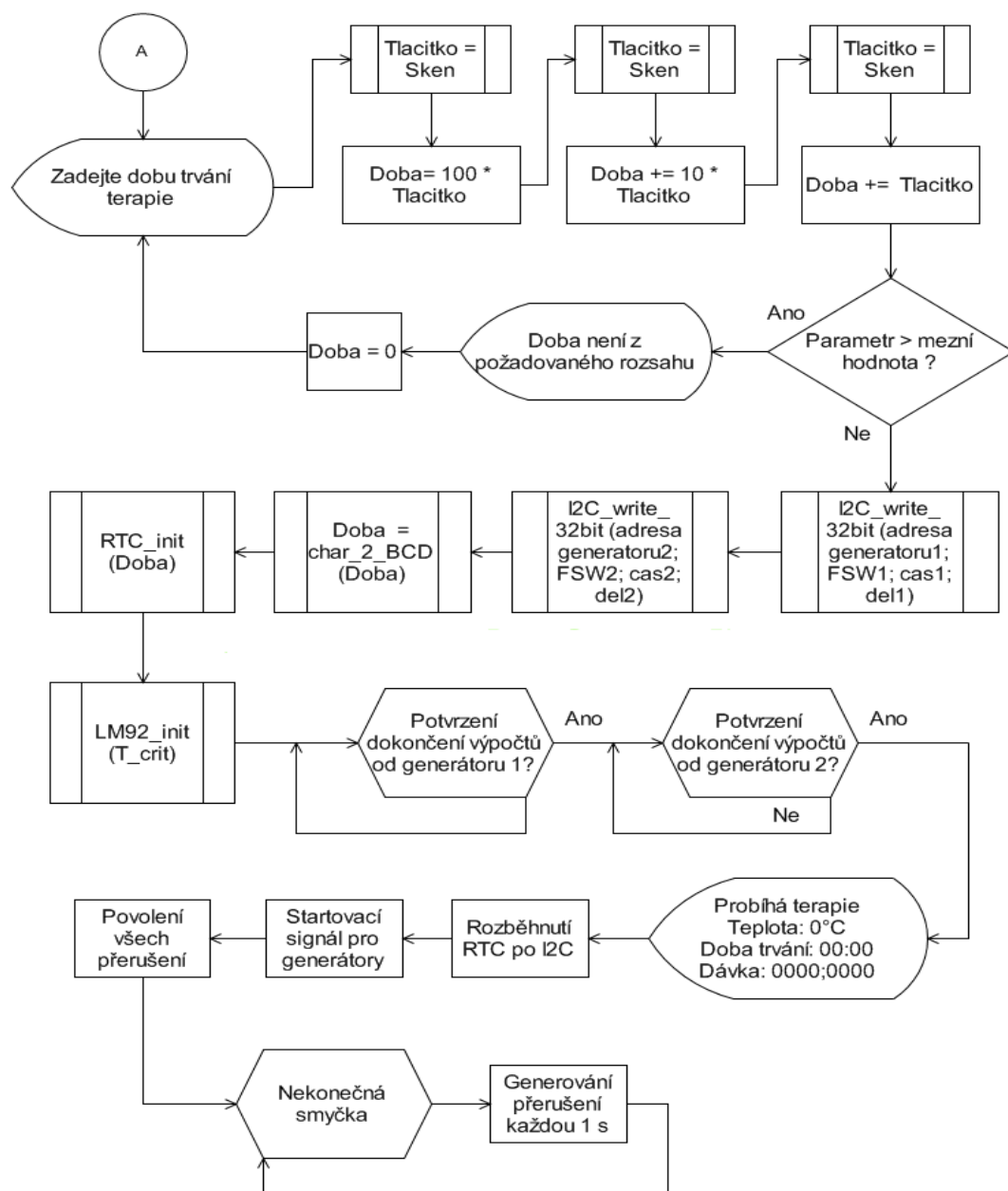


E VÝVOJOVÉ DIAGRAMY PROGRAMŮ

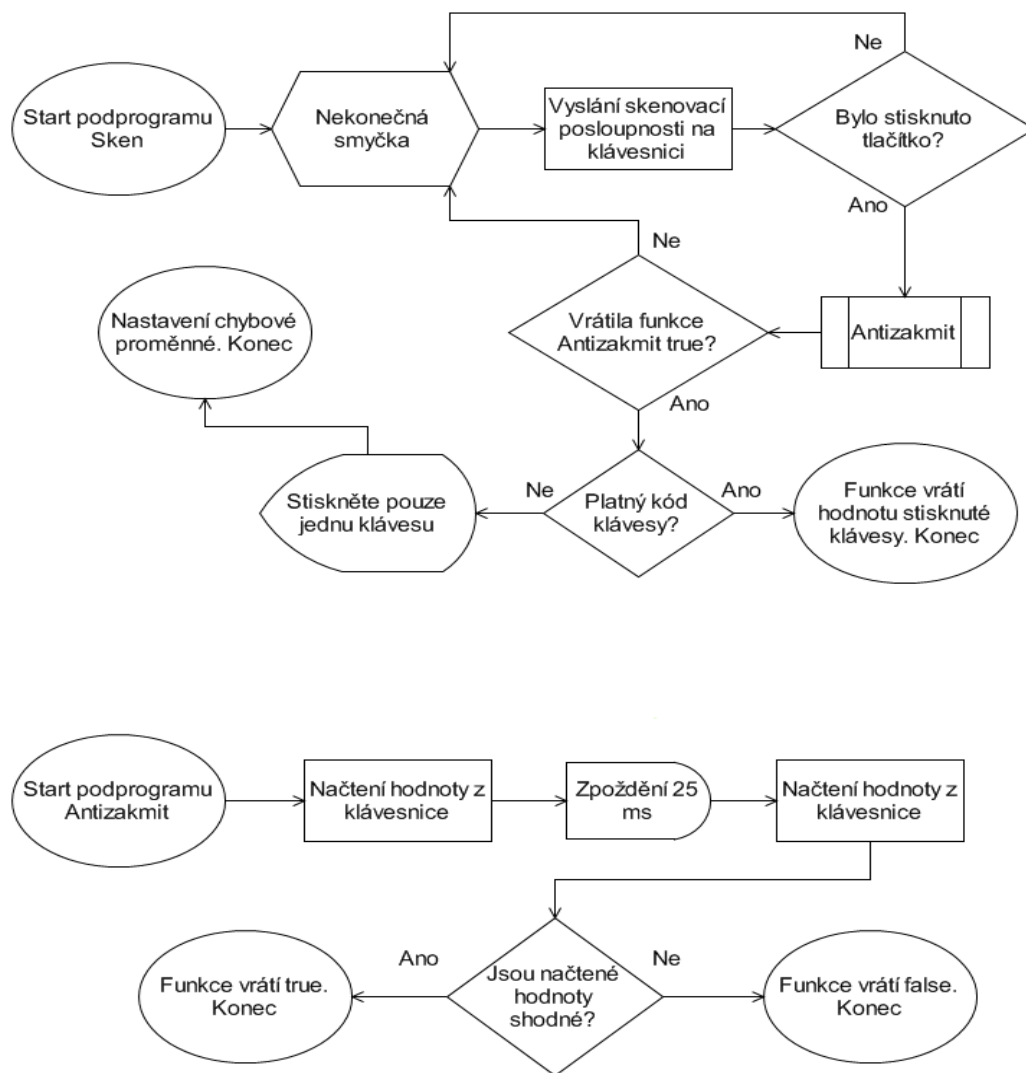
E.1 Vývojový diagram pro řídicí a kontrolní mikrokontrolér



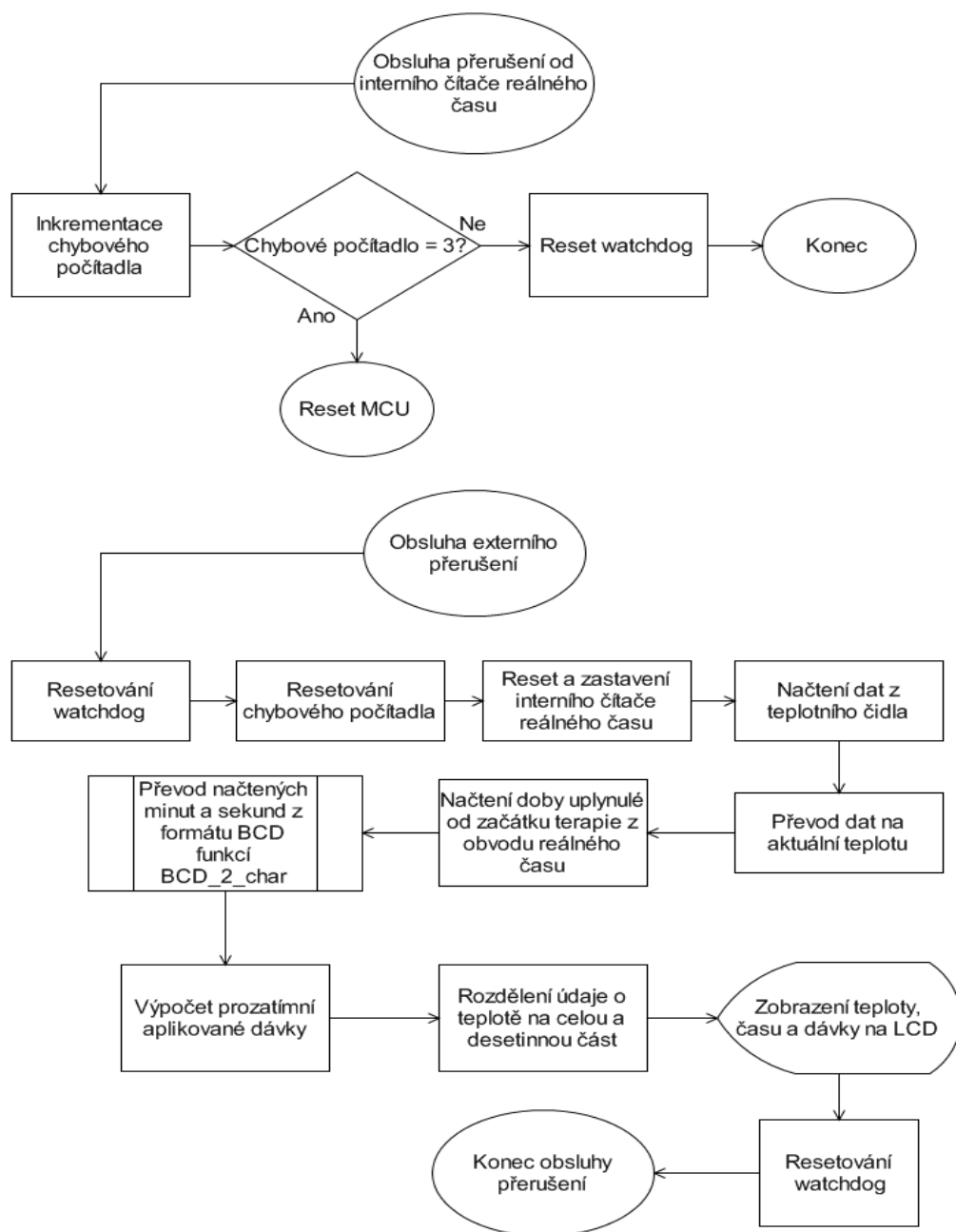
Obr. E.1: První část vývojového diagramu hlavního programu



Obr. E.2: Druhá část vývojového diagramu hlavního programu

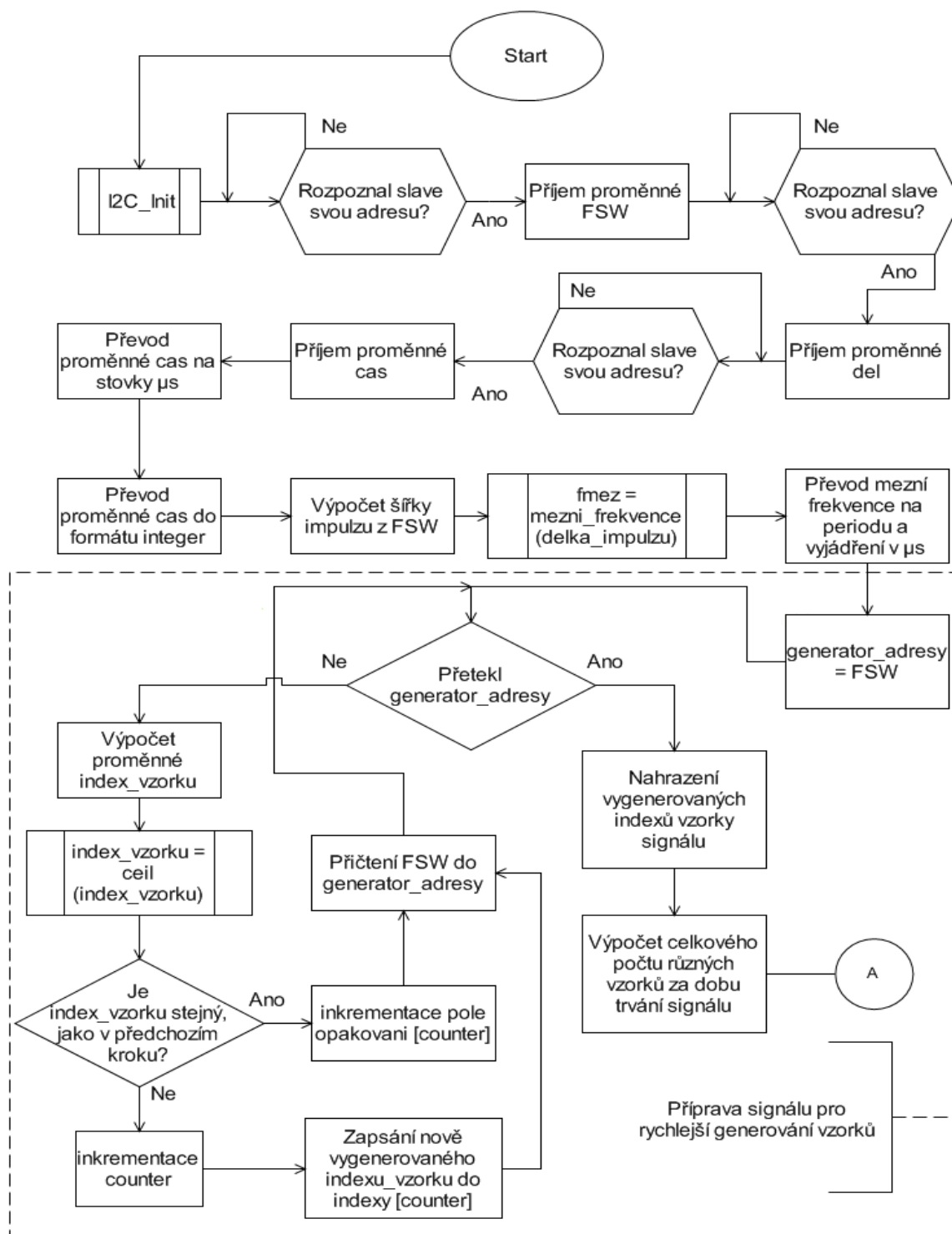


Obr. E.3: Vývojové diagramy podprogramů pro řídicí MCU

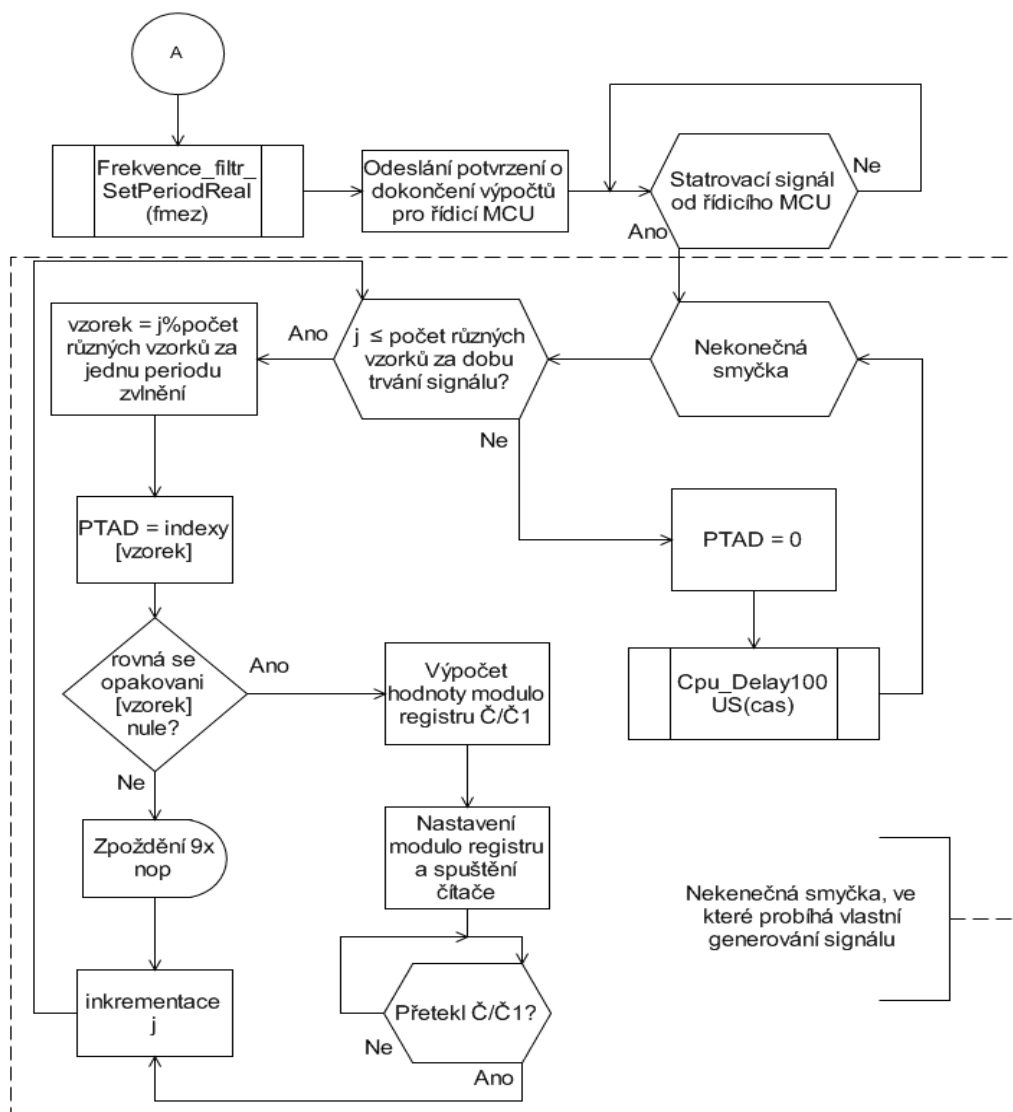


Obr. E.4: Vývojové diagramy obsluh přerušení pro řídicí MCU

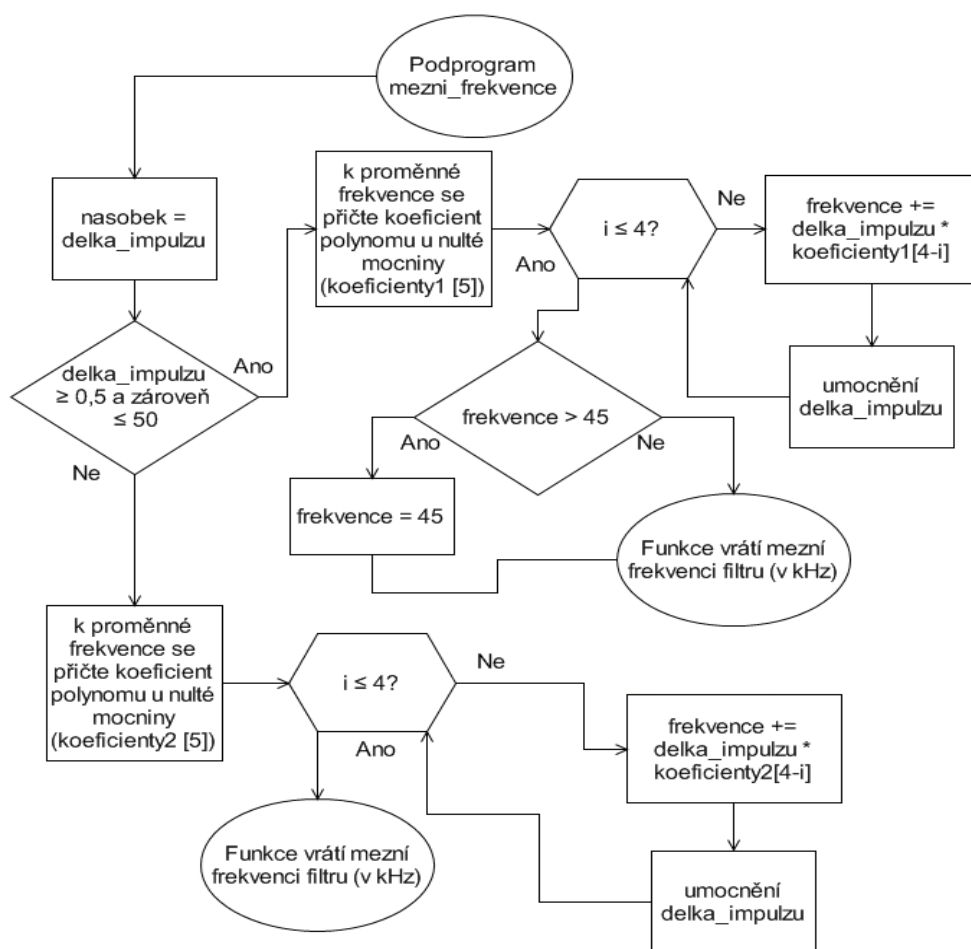
E.2 Vývojový diagram pro generující MCU



Obr. E.5: První část vývojového diagramu hlavního programu



Obr. E.6: Druhá část vývojového diagramu hlavního programu



Obr. E.7: Vývojové diagramy podprogramů pro generující MCU