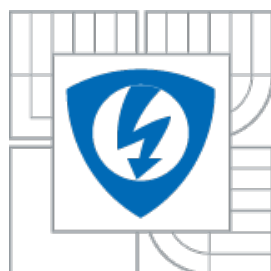


**VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ**  
BRNO UNIVERSITY OF TECHNOLOGY



**FAKULTA ELEKTROTECHNIKY A  
KOMUNIKAČNÍCH TECHNOLOGIÍ  
ÚSTAV AUTOMATIZACE A MĚŘICÍ TECHNIKY**

**FACULTY OF ELECTRICAL ENGINEERING AND  
COMMUNICATION  
DEPARTMENT OF CONTROL AND INSTRUMENTATION**

# **AUTOMATIZACE STOLKU LÉKÁŘSKÉ DENTÁLNÍ SOUPRAVY VIZIO**

VIZIO – DENTAL SYSTEM AUTOMATION

**DIPLOMOVÁ PRÁCE**  
MASTER'S THESIS

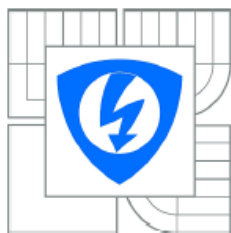
**AUTOR PRÁCE**  
AUTHOR

**Bc. MICHAL GÁBRIŠ**

**VEDOUCÍ PRÁCE**  
SUPERVISOR

**doc. Ing. ZDENĚK BRADÁČ, Ph.D.**

BRNO 2014



VYSOKÉ UČENÍ  
TECHNICKÉ V BRNĚ

Fakulta elektrotechniky  
a komunikačních technologií

Ústav automatizace a měřicí techniky

# Diplomová práce

magisterský navazující studijní obor  
Kybernetika, automatizace a měření

**Student:** Bc. Michal Gábriš  
**Ročník:** 2

**ID:** 119929  
**Akademický rok:** 2013/2014

## NÁZEV TÉMATU:

**Automatizace stolku lékařské dentální soupravy VIZIO**

## POKYNY PRO VYPRACOVÁNÍ:

Cílem práce je návrh a realizace rozšíření schopností lékařské dentální soupravy VIZIO.

1. Zpracujte literární rešerši obdobných systémů.
2. Navrhněte konstrukci, integraci a elektrotechnické řešení kapacitní klávesnice do systému. Dále navrhněte princip řízení a regulace otáček integrovaného bezkartáčového motoru.
3. Popište návrh a navrhněte elektrické schéma.
4. Navrhněte elektroniku, realizujte DPS, osadte a oživte.
5. Napište programové vybavení a integrujte se stávající elektrickou výbavou stávající dentální soupravy VIZIO.
6. Ověřte, otestujte a demonstруйте funkčnost. Vyhodnoťte výsledky.

## DOPORUČENÁ LITERATURA:

ZEZULKA, F. Prostředky průmyslové automatizace. VUTIUM. VUTIUM. Brno: VUTIUM, 2004. 176 s.  
ISBN: 80-214-2610- 1.

Dle pokynů vedoucího práce.

**Termín zadání:** 10.2.2014

**Termín odevzdání:** 19.5.2014

**Vedoucí práce:** doc. Ing. Zdeněk Bradáč, Ph.D.

**Konzultanti diplomové práce:**

**doc. Ing. Václav Jirsík, CSc.**

*Předseda oborové rady*

## UPOZORNĚNÍ:

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

## **Abstrakt**

Tato diplomová práce se zabývá návrhem a realizací rozšíření schopností lékařské dentální soupravy Vizio. Konkrétně jde o elektrotechnické řešení a programové vybavení kapacitní klávesnice a o návrh principu řízení a regulace otáček integrovaného bezkartáčového motoru. Práce obsahuje jak popis použitého hardware, tak i softwarového řešení, které bylo implementováno v jazyce assembler. Toto zařízení je již komerčně využíváno.

## **Klíčová slova**

vestavěný systém, mikrokontrolér, Microchip, PIC, měření, kapacita, dentální souprava, lékař, medicína, assembler, motor, regulace

## **Abstract**

This Diploma thesis deals with creation and realisation of extensions for medical dental unit Vizio. Specifically, it deals with electrotechnical solution and software for capacitive keyboard and with proposal of principle of control and regulation of speed for integrated brushless motor. The Diploma thesis contains both the description of used hardware components, and software solution implemented in assembly language. This device is already in commercial use.

## **Keywords**

embedded system, microcontroller, Microchip, PIC, measurement, capacity, dental unit, doctor, medicine, assembler, motor, adjustment

## **Bibliografická citace**

GÁBRIŠ, M. *Automatizace stolku lékařské dentální soupravy Vizio*. Brno: Vysoké Učení Technické v Brně. Fakulta elektrotechniky a komunikačních technologií, 2014. 89 s. Vedoucí diplomové práce doc. Ing. Zdeněk Bradáč, Ph.D.

## **Prohlášení**

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně pod vedením doc. Ing. Zdeňka Bradáče, Ph.D.

Další informace mi poskytl zadavatel práce, Ing. Jozef Chamraz, CSc. ze společnosti INOVATICA, s.r.o.

Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

.....  
Michal Gábriš  
18.5.2014

## **Poděkování**

Tímto bych chtěl poděkovat především dvěma lidem, a to vedoucímu práce, panu doc. Ing. Zdeňku Bradáčovi, Ph.D., za starostlivý a odborný přístup k práci, a zadavateli práce, panu Ing. Jozefu Chamrazovi, Csc., za přípravu zadání, které je přímo z praxe a za pomocnou ruku počas celé tvorby diplomové práce. Dále bych chtěl poděkovat všem, kteří mě podporovali a přáli mi štěstí – své rodině a přátelům.

# Obsah

1 Úvod.....	10
1.1 Popis jednotlivých kapitol.....	10
2 Literárna rešerš – prehľad trhu v oblasti dentálnych súprav.....	12
2.1 DIPLOMAT DENTAL s.r.o. Piešťany – rada DIPLOMAT [1].....	12
2.1.1 Diplomat Adept DA 370 .....	12
2.1.2 Diplomat Consul DC350.....	14
2.1.3 Diplomat Lux DL320.....	15
2.2 CHIRANA Medical, a.s. Stará Turá – rada Smile [2].....	16
2.3 Chiradent, s.r.o. Piešťany – rada VIZIO [3].....	19
2.3.1 VIZIO E.....	19
2.3.2 VIZIO L.....	20
2.4 Elettromeccanica Castellini, Bologna – rada Puma ELI 5 [4].....	22
2.4.1 PUMA ELI 5.....	22
2.4.2 PUMA ELI 5 AMBIDEXTROUS.....	22
2.5 Elettromeccanica Castellini, Bologna – rada Skema 5 [4].....	22
2.6 Elettromeccanica Castellini, Bologna – rada Skema 6 [4].....	23
2.7 Elettromeccanica Castellini, Bologna – rada Skema 8 [4].....	23
3 Konceptcia klávesnice a definícia požiadaviek.....	25
3.1 Požiadavky od zadávateľa.....	25
3.2 Dostupné riešenia.....	25
3.3 Konceptcia klávesnice.....	27
4 Elektrické riešenie klávesnice.....	30
4.1 Kapacitné senzory [6].....	30
4.1.1 Bezkontaktné kapacitné senzory.....	30
4.2 Výber vhodného riešenia (obvodu).....	33
4.3 Modul kapacitného snímania obvodu PIC16F722 [5].....	33
4.3.1 Analógový multiplexor.....	34
4.3.2 Oscilátor kapacitného snímania.....	35
4.3.3 Zdroje časovania.....	35
4.3.4 Fixná časová základňa.....	35
4.4 Obvody podsvietenia.....	36

4.4.1	LED dióda LED 1206 BLUE 45/130° [9].....	36
4.4.2	LED dióda L-KP-2012QBC-D [10].....	36
4.4.3	Spínací regulátor MC34063A [11].....	36
4.5	Riadenie LED diód pre podsvietenie.....	38
4.6	Zapojenie pinov mikrokontroléra.....	38
4.7	In-circuit sériové programovanie (ICSP) [5].....	40
4.7.1	ICSP pomocou programátora Elnec SmartProg2.....	41
4.8	Sériová komunikácia pomocou RS-232 [13].....	44
5	Riešenie programu klávesnice.....	46
5.1	Voľba vývojového prostredia .....	46
5.2	Výber programovacieho jazyka.....	46
5.3	Počiatočná konfigurácia mikrokontroléra.....	46
5.3.1	Konfigurovateľné obvody.....	46
5.3.2	Výber oscilátora.....	47
5.3.3	Watchdog Timer.....	47
5.3.4	Power-up Timer.....	47
5.3.5	Brown-out Reset.....	47
5.4	Počiatočné nastavenia.....	47
5.4.1	Vstupy a výstupy.....	47
5.4.2	Adresovateľný univerzálny synchronný/asynchronný prijímač/vysielač.....	48
5.4.3	Prerušenie.....	48
5.5	Rozlíšenie mikrokontrolérov.....	48
5.6	Algoritmus pre vyhodnotenie kapacitného snímania.....	49
5.7	Telo prerušenia.....	50
5.7.1	Funkcia ReadREG.....	51
5.7.2	Funkcia PresetREG.....	53
5.8	Hlavný program.....	56
5.8.1	Funkcia READ_CAP_SENSORS.....	57
5.8.2	Cyklus U4_LOOP.....	57
5.8.3	Cyklus U3_LOOP.....	58
6	Riešenie programu na riadenie bezuhlíkového motora.....	60
6.1	Zapojenie.....	60
6.2	Štruktúra prenášaných dát.....	60



6.3 Riešenie programu.....	64
6.3.1 Funkcia Motor_STOP.....	64
6.3.2 Funkcia Motor_START_FW.....	65
6.3.3 Funkcia Motor_START_FW_LEDON.....	65
6.3.4 Funkcia Motor_START_REVERSE.....	65
6.3.5 Funkcia Motor_START_REVERSE_LEDON.....	66
6.3.6 Funkcia Motor_START_AutoReverse.....	66
6.3.7 Funkcia Motor_START_AutoReverse_LEDON.....	66
6.3.8 Funkcia Motor_START_AutoReverseForward.....	66
6.3.9 Funkcia Motor_START_AutoRevrseForwardLEDON.....	66
6.3.10 Funkcia MotorB_COMMAND_Exe.....	67
7 Záver.....	70
7.1 Klávesnica s proximity senzormi.....	70
7.2 Bezuhlíkový motor.....	71
8 Zoznam príloh.....	74

# 1 Úvod

Táto diplomová práca vznikla v prostredí Fakulty elektrotechniky a komunikačných technológií na Vysokom učení technickém v Brne. Práca mi bola zadaná firmou Inovatica, s.r.o., so sídlom v Malackách, Slovenská Republika, ktorá sa špecializuje na moderné riešenia elektroniky a to hlavne zapuzdrených systémov (Embedded Systems). Hlavnou požiadavkou zadávateľa bolo navrhnuť a realizovať rozšírenie schopností lekárskej dentálnej súpravy Vizio, a to konkrétne:

- navrhnuť konštrukciu, integráciu a elektrotechnické riešenie kapacitnej klávesnice do systému
- navrhnuť princíp riadenia a regulácie otáčok integrovaného bezuhlíkového motora

Vzhľadom k tomu, že hlavnými určujúcimi parametrami výsledného produktu boli výsledky marketingu, požiadavky vyplývajúce z bezpečnostných predpisov a dizajnu pre cieľovú používateľskú oblasť, bolo nutné aj vývoj softwaru a hardwaru týmto skutočnostiam podriaďiť. Zadávateľ kládol veľký dôraz na výslednú kvalitu a spoľahlivosť produktu.

## 1.1 Popis jednotlivých kapitol

Uvedieme si stručný popis o čom pojednávajú jednotlivé kapitoly práce.

**Kapitola 2: Literárna rešerš** – v 2. kapitole budeme písať o situácii na trhu, na ktorý chceme s našim výrobkom vstúpiť, teda konkrétne tu budeme porovnávať dentálne súpravy predávané firmou s ktorou spolupracujeme a konkurenčnými firmami na Slovensku, ale aj zahraničným výrobcom.

**Kapitola 3: Koncepcia klávesnice a definícia požiadaviek** – v tejto kapitole sa podrobnejšie pozrieme na konkrétne požiadavky dané od zadávateľa práce, a pozrieme sa na konkrétne riešenia, ktoré možno využiť a na dizajn klávesnice.

**Kapitola 4: Elektrické riešenie klávesnice** – v tejto kapitole si rozoberieme princíp kapacitného snímania, vyberieme vhodný obvod pre naše riešenie, ktorý má modul kapacitného snímania, a detailne si pozrieme tento modul. Následne sa pozrieme aj na ostatné komponenty elektrického riešenia.

**Kapitola 5: Riešenie programu klávesnice** – v tejto kapitole sa podrobne pozrieme na vývoj softwaru pre klávesnicu.

**Kapitola 6: Riešenie programu na riadenie bezuhlíkového motora** – táto kapitola sa zaoberá riadením bezuhlíkového motora, aké prostriedky k tomu využijeme a rozoberieme si detailne program.

**Záver** – v poslednej kapitole zhrnieme význam práce a zamyslíme sa nad možnými rozšíreniami do budúcnosti.

**Prílohy** – V priložených dokumentoch sa nachádzajú informácie o hardwarovom riešení produktu – elektrické schémy a dosky plošných spojov, komunikácia s firmou NSK Japonsko, a popis komunikácie medzi prvkami dentálnej súpravy.

## 2 Literárna rešerš – prehľad trhu v oblasti dentálnych súprav

### 2.1 DIPLOMAT DENTAL s.r.o. Piešťany – rada DIPLOMAT [1]



Obrázok 2.1: DIPLOMAT ADEPT DA370 [1]

#### 2.1.1 Diplomat Adept DA 370

Táto súprava ponúka skutočne exkluzívnu starostlivosť – či sa jedná o preventívne vyšetrenie alebo o náročný stomatologický zákrok. Jej technická úroveň, ergonómia a progresívny dizajn sú zárukou tej najvyššej kvality.

Táto stomatologická súprava je riešená ako stacionárna súprava s pľuvadlovým blokom nesúcim polohovateľné kreslo.

Medzi jej výhody patria prepracovaný softvér, programovateľná elektronika, systém hygieny a dekontaminácie, či možnosť osadenia LED svietidla.

Na špeciálnu požiadavku je možné k súprave domontovať RTG aparát alebo operačný mikroskop.

Kreslová časť súpravy zabezpečuje dostatok voľného miesta pre nohy lekára a umožňuje i tzv. tretí pohyb pre lepšie nastavenie polohy pacienta čo zvyšuje ergonómiu práce lekára ako aj komfort pacienta.

Technické parametre:	
Napájacie sieťové napätie	230 V ± 10%
Frekvencia	50 Hz ± 2%
Max. príkon pri 50 Hz	1900 VA ±10%
Vstupný tlak vzduchu	0,45 až 0,8 MPa
Vstupný tlak vody	0,3 až 0,6 MPa
Hmotnosť súpravy (bez kresla)	35 kg
Typ ochrany pred úrazom el. prúdom	Prístroj triedy ochrany I
Stupeň ochrany pred úrazom el. prúdom	Príložené časti typu B
Teplota vody pre pohár	25 - 45° C
Max. zaťažiteľnosť odkladacieho stolíka	3 kg
Max. zaťažiteľnosť tray stolíka	1,5 kg
Režim prevádzky	Trvalý s prerušovaným zaťažením – zodpovedajúci obvyklej stomatologickej praxi
Okolie:	
- rozsah teploty okolia	od +10°C do +40°C
- rozsah relatívnej vlhkosti	od 30% do 75%
- rozsah atmosferického tlaku	od 700 hPa do 1 060 hPa

Tabulka 2.1: Technické parametre súpravy Diplomat Adept DA 370 [1]

## 2.1.2 Diplomat Consul DC350



Obrázok 2.3: DIPLOMAT CONSUL DC350 [1]

Vyznačuje sa maximálnou variabilitou nástrojov podľa požiadaviek lekárov. Tento typ stomatologickej súpravy môže byť vybavený okrem odsávacieho systému ďalšími až deviatimi nástrojmi, šiestimi na stolíku lekára a tromi na bloku asistenta. Obsahuje aj pľuvadlový blok s otočnou, odnímateľnou a sterilizovateľnou misou, ktorú je možné ovládať i elektronicky. Pľuvadlový blok umožňuje zabudovanie odsávacieho systému Cattani, Metasys alebo Dürr s automatickou dezinfekciou celého odsávacieho systému a hadíc i dekontamináciu vodných ciest.

Šesť nástrojový stolík lekára s vrchným vedením hadíc umožňuje osadenie až piatich rotačných svetelných nástrojov, z čoho môžu byť štyri mikromotory. Stomatologická súprava DC350 umožňuje pripojenie až dvoch bezuhlíkových, sterilizovateľných mikromotorov od švajčiarskej firmy BienAir, alebo japonského výrobcu NSK. Na komfortné ovládanie všetkých funkcií stomatologickej súpravy a kresla slúži dotykový displej. Nastavenie požadovanej pracovnej polohy stolíka lekára umožňuje pantografické rameno s elektropneumatickou brzdou.

Otočný stolík asistenta a odkladací tray stolík je uchytený na viackĺbovom, elektropneumaticky alebo mechanicky brzdenom ramene. Stolík sestry môže byť okrem malej a veľkej odsávačky vybavený až tromi nástrojmi, napríklad intraorálnou kamerou, polymerizačnou lampou či striekačkou asistenta. Ovládanie jednotlivých funkcií zo strany sestry je riešené fóliovou klávesnicou alebo dotykovým displejom. Stolík asistenta je možné doplniť nerezovou táckou či odsliňovačom na špeciálnu objednávku. Ovládanie kresla je možné zo strany asistenta, lekára i pomocou nožného ovládača.

Technické parametre:	
Napájacie sieťové napätie	230 V ± 10%
Frekvencia	50 Hz ± 2%
Max. príkon pri 50 Hz	450VA ± 10%
Vstupný tlak vzduchu	0,45 až 0,8 MPa
Vstupný tlak vody	0,3 až 0,6 MPa
Hmotnosť súpravy (bez kresla)	80kg + max 35 kg
Typ ochrany pred úrazom el. prúdom	Prístroj triedy ochrany I
Stupeň ochrany pred úrazom el. prúdom	Príložné časti typu B
Teplota vody pre pohár	25 - 45° C
Max. zaťažiteľnosť odkladacieho stolíka	3 kg
Max. zaťažiteľnosť tray stolíka	1,5 kg
Režim prevádzky	Trvalý s prerušovaným zaťažením – zodpovedajúci obvyklej stomatologickej praxi
Okolie:	
- rozsah teploty okolia	od +10°C do +40°C
- rozsah relatívnej vlhkosti	od 30% do 75%
- rozsah atmosferického tlaku	od 700 hPa do 1 060 hPa

Tabulka 2.2: Technické parametre súpravy Diplomat Consul DC 350 [1]

### 2.1.3 Diplomat Lux DL320

Súprava s dolným vedením hadíc ponúka vysoký štandard práce pre stomatológov, ktorí preferujú prácu s prístupom smerom k boku pacienta. Ponúka: dotykový LCD displej na strane lekára i sestry s intuitívnym ovládaním ponúkaných funkcií, elektro-pneumaticky brzdené ramená, elektronicky otočnú pľuvadlovú misu, ovládanie až 6 nástrojov na stolíku lekára a 5 nástrojov na strane asistenta, či energoblok umiestnený priamo v kresle. Ďalej prepracovaný software a kvalitnú elektroniku s možnosťou ovládania dvoch bezuhlíkových a sterilizovateľných MX/NLX motorov, systém hygieny a dekontaminácie, či možnosť osadenia LED svietidla vlastnej výroby s možnosťou plynulej regulácie v rozsahu 8 000-30 000 lx.



Obrázok 2.4: DIPLOMAT LUX DL320 [1]

Technické parametre: sú rovnaké ako v prípade typu Consul DC350.

## 2.2 CHIRANA Medical, a.s. Stará Turá – rada Smile [2]

Táto rada dentálnych súprav má viaceré funkcie rovnaké pre každý typ, to sú:

- Ovládanie funkcií súpravy cez fóliovú klávesnicu stolíka lekára
- Regulácia rýchlosti mikromotora už od 60 ot/min pri nadštandardnom krútiacom momente
- Elektronická funkcia cyklického reverzovania mikromotora (Giromatic)
- Nastavenie výkonu nástrojov cez fóliovú klávesnicu, alebo proporcionálne nožným ovládačom
- Vyfukovanie cez nástroj (Chip blowing) spúšťané nožným ovládačom (bez časového obmedzenia)
- Automatické vyfukovanie cez nástroj (Chip blowing) po skončení práce s nástrojom
- Otočná porcelánová pľuvadlová misa (smerom k pacientovi)
- Ovládanie kresla z dvoch miest: z klávesnice stolíka lekára, nožným ovládačom
- Prívody energií pod predným krytom kresla
- Široká škála farebných prevedení
- Možnosť kompletácie kreslom SK1.01





Obrázok 2.5: Dentálna súprava Smile Charm Z [2]



Obrázok 2.6: Dentálna súprava Smile Elegant [2]

Stredné typy súprav Smile obsahujú navyše tieto funkcie:

- Svetelná a akustická indikácia potreby ošetrenia nástroja (mazanie)
- Odnímateľný a ľahko ovládateľný systém horného vedenia hadíc, plne flexibilný, bez spätného ťahu hadíc
- Fixovanie polohy stolíka lekára automatickou brzdou
- Možnosť kompletácie i kreslom SK1.08
- Ovládanie kresla z troch miest: z klávesnice stolíka lekára, z klávesnice stolíka sestry i nožným ovládačom

Najvyššie typy Smile Elegant a Smile Charm navyše obsahujú:

- Všetky funkcie nezávisle nastaviteľné pre dvoch lekárov
- Možnosť kontroly času do mazania
- Nastavenie času oplachu misy a plnenia pohára a tiež nastavenie automatického oplachu po naplnení pohára

## 2.3 Chiradent, s.r.o. Piešťany – rada VIZIO [3]

### 2.3.1 VIZIO E

Súprava je určená pre všetky typy stomatologických výkonov, pre náročných používateľov, ktorí ocenia individualitu, komfort, ergonómiu, funkcionality ako aj nové riešenia stolíka lekára s veľkokapacitnou sklenenou dotykovou klávesnicou. Súprava VIZIO poskytuje komfort v práci a zdôrazní individualitu každého lekára.



*Obrázok 2.7: Dentálna súprava Chiradent VIZIO E [3]*

**Technické parametre:**

Napájacie sieťové napätie	230 V ± 10%
Menovitá frekvencia	50 Hz ± 5%
Max. príkon pri 50 Hz	350 VA
Vstupný tlak vzduchu	0,27 až 0,6 MPa
Vstupný tlak vody	0,45 až 0,6 MPa
Hmotnosť súpravy	145 kg ± 5 kg
Typ ochrany pred úrazom el. prúdom	prístroj triedy I.
Stupeň ochrany pred úrazom el. prúdom	prístroj typu B
Teplota vody pre pohár (kotlík na objednávku)	36 ± 5 °C
Minimálna poloha sedadla kresla	380 ± 20 mm
Maximálna poloha sedadla kresla	860 ± 20 mm
Okolie:	podľa EN 60 601-1
Rozsah teploty okolia	od +10 °C do +40 °C
Rozsah relatívnej vlhkosti	od 30 % do 75 %

Tabulka 2.3: Technické parametre súpravy Chiradent Vizio E [3]

### 2.3.2 VIZIO L

Súprava má navyše oproti typu VIZIO E odkladaciu skrinku s osvetlením, servisnú časť umiestnenú v stĺpe súpravy ako aj svetový nadčasový dizajn, ktorý nemá vo svete stomatológie alternatívu. Zákazník si môže vybrať z 8 odtieňov metalických farieb a takisto z 3 motívov tapiet v 10 farebných odtieňoch. Svoju individualitu môže zákazník zdôrazniť výberom tapety s vlastnými motívmi, ktoré zabezpečujeme na základe jeho požiadavky.



*Obrázok 2.8: Dentálna súprava VIZIO L [3]*

**Technické parametre:** sú rovnaké ako u súpravy VIZIO E.

## 2.4 Elettromeccanica Castellini, Bologna – rada Puma ELI 5

[4]

### 2.4.1 PUMA ELI 5

Súprava je navrhnutá pre 5 nástrojov, má samonastavovacie operadlo a veľmi jednoducho sa nastavuje do najpohodlnejších a ideálnych polôh podľa toho aká liečba sa vykonáva. Páky nástrojov a dĺžka hadíc pomáhajú znížiť fyzické napätie pôsobiace na lekárovo zápästie pri práci.



Obrázok 2.10: Dentálna súprava Puma Eli 5 [4]



Obrázok 2.9: Dentálna súprava Puma Eli 5 Ambidextrous[4]

### 2.4.2 PUMA ELI 5 AMBIDEXTROUS

Dostupná ako všestranná jednotka, ELI 5 môže byť rýchlo prepnutá z ľavej strany do pravej pomocou niekoľko jednoduchých krokov, ktoré nepotrebujú pomoc technického personálu. Táto jednotka je dostupná aj ako model s nástrojmi „whip arm“, ktoré maximalizujú užívateľov pracovný uhol.

## 2.5 Elettromeccanica Castellini, Bologna – rada Skema 5 [4]

Hlavnou vlastnosťou tejto súpravy je jej flexibilita, adaptabilita na hocijaký pracovný scenár. Vďaka konceptom ergonomickej prispôsobiteľnosti lekárskeho personálu bude mať optimálne pracovné podmienky v každom štádiu pacientovej liečby.

Navyše k rysom navrhnutým na zlepšenie pracovnej ekonomiky lekára a sestričky, Skema 5 ponúka vylepšené zariadenia hygieny, dizajn aby sa denné čistiace práce robili jednoduchšie a vysoký rozsah možností vyvinutých vo výskumných laboratóriách firmy Castellini.

Skema 5 adoptuje mnoho technologicky pokročilých nástrojov, ktoré sú prelomové a exkluzívne, navrhnuté na zabezpečenie výkonu a funkcií, ktoré umožňujú lekárovi dosiahnuť bez obáv každú klinickú potrebu.



*Obrázok 2.11: Dentálna súprava Skema 5 [4]*



*Obrázok 2.12: Dentálna súprava Skema 5 [4]*

## **2.6 Elettromeccanica Castellini, Bologna – rada Skema 6 [4]**

Skema 6 prispieva k potrebám sektoru dentálnej chirurgie s technologickými inováciami, ergonomickou flexibilitou a praktickým dizajnom.

Dizajnový koncept projektu Skema 6 je výsledkom výskumu namiereného na zabezpečenie čo najväčšej slobody pohybu pre lekárskeho personál počas procedúr. Skema 6 má prostriedky na garantovanie stálej bezpečnosti prevádzky. Okrem konceptu pasívnej ochrany pomocou množiny jednotiek a zariadení navrhnutých na minimalizovanie rizika zmien hygienických podmienok, vnútri aj na povrchu dentálnej jednotky má aktívne zariadenia na zabezpečenie neohroziteľnosti so zvyšujúcimi sa rizikovými faktormi.

Všetky nástroje sú integrované do elektronického systému dentálnej jednotky a poskytujú výnimočný výkon vďaka použitiu prelomových, užívateľsky prívetivých technologických riešení. Turbíny, bezuhlíkové mikromotory, skalery, plne odstrániteľné kanyly, ktoré môžu byť sterilizované v autokláve a exkluzívny ultrazvukový ručný nástroj na chirurgiu.

Základné rysy typu Skema 6 sú – optimalizácia priestoru, ergonomická flexibilita, komfort pri práci a dotykové ovládanie.

## **2.7 Elettromeccanica Castellini, Bologna – rada Skema 8 [4]**

Skema 8 je kompletne liečebné centrum firmy Castellini. Vyznačuje sa integrovanými odbornými nástrojmi, exkluzívnymi technológiami a kreslom pacienta, ktoré je unikátne.



Obrázok 2.14: Dentálna súprava Skema 8 [4]



Obrázok 2.13: Funkcia klzania súpravy Skema 8 [4]

Špeciálne vlastnosti:

- Implantor LED – bezuhlíkový mikromotor schopný rozsiahlych hodnôt krútiaceho momentu
- Sistema LAEC – maximálna efektivita a klinická bezpečnosť v ortodontických liečbach
- Autosteril – dezinfekcia obehu body postrekovača
- Surgison 2 – ultrazvukový ručný nástroj pre chirurgické výkony

Ďalšou funkciou súpravy je klzanie. Táto funkcia pozostáva zo synchronizovaného pohybu pacientovho kresla: kým operadlo sa znižuje, sedadlo sa kĺže dopredu a zvyšuje sa. Toto má 4 kľúčové výhody: udržiava konštantné pomery vzdialeností medzi činnou zónou a dvomi dentálnymi jednotkami; dovoľuje pacientovi zmeniť pozíciu bez nutnosti posunutia svetla, takže žiarovka zostáva namierená na ústnu dutinu; je oveľa lepšie využiteľný priestor okolo nakloneného operadla oproti využitiu kresla bez možnosti klzania; zvyšuje komfort pacienta.

Full Touch je veľká dotyková obrazovka (5.7"); vyrobená z tvrdeného skla, môže byť kompletne dezinfikovaná, ponúka veľkú odolnosť pred škrabancami a hlavne poskytuje užívateľsky prívetivé riadenie všetkých funkcií dentálnej súpravy. Ovládací panel Full Touch dovoľuje lekárovi nastaviť a uložiť pracovné parametre každého nástroja.



## 3 Konceptia klávesnice a definícia požiadaviek

### 3.1 Požiadavky od zadávateľa

Tieto požiadavky boli definované od zadávateľa:

- 15 kapacitných senzorov
- komunikácia klávesnice so stolíkom dentálnej súpravy – príloha A
- klávesnica musí byť čierna sklenená plocha s modrými podsvietenými ikonami, ktoré pracujú ako tlačidlá, zadávateľ dal aj rozmery
- klávesnica musí byť rekonfigurovateľná (svietia len ikony pre tie senzory, ktoré sú aktívne)
- senzory musia mať tvar ikony, ktorá znázorňuje príslušnú dentálnu operáciu (senzory musia byť transparentné – priesvitné)
- kapacitné senzory musia byť odolné voči pomalým zmenám kapacity (zmena povrchovej vlhkosti, prach, ...)

Z požiadaviek vyplýva, že treba zrealizovať:

- Senzor priblíženia s povoľovaním a zakazovaním jeho činnosti
- Podsvietenie senzora tak, aby na stolíku lekára bola vidieť jedine aktívna ikona
- Indikáciu aktivizácie senzora
- Komunikáciu s riadiacim mikrokontrolérom v stolíku lekára (klávesnica ovláda činnosť celej súpravy cez uvedený mikrokontrolér)
- Komunikáciu s LCD displejom

Z analýzy možných technických riešení vyplynulo použitie kapacitných senzorov.

### 3.2 Dostupné riešenia

**Microchip PIC16F72X** – 28/40/44 pinové mikrokontroléry, ktoré obsahujú modul pre kapacitné snímanie

Niektoré vlastnosti:

- typy 722/723/726 obsahujú 8 pinov pre kapacitné snímanie, a typy 724/727 obsahujú 16 pinov pre kapacitné snímanie
- obsahuje výkonnú CPU typu RISC
- 8 bitový A/D prevodník až so 14 kanálmi
- interný oscilátor 16 MHz alebo 500 kHz
- komunikačné rozhrania SPI, I2C, ASYNC
- In circuit programovanie

Dostupnosť: Premier Farnell plc

Cena: 722/723/726 pre 100+ kusov: od 1,22 GBP po 1,39 GBP za kus

724/727 pre 100+ kusov: od 1,60 GBP po 1,61 GBP za kus

**Analog Devices AD7150** – 2 kanálový prevodník kapacity pre snímanie priblíženia s ultra nízkym napätím

Vlastnosti:

1. 2 nezávislé vstupné kanály pre kapacitu
2. 2-vodičové sériové rozhranie kompatibilné s I2C
3. nutnosť použiť ďalší mikrokontrolér

Dostupnosť: Analog Devices, Inc.

Cena: pre 100+ kusov: 1,62 USD za kus

**Microchip PIC10F20X** – 6 pinové, 8 bitové mikrokontroléry

Vlastnosti:

1. obsahuje výkonnú CPU typu RISC
2. interný oscilátor 4 MHz
3. In circuit programovanie
4. PIC10F204 a 206 obsahujú komparátor, ktorý sa dá použiť na kapacitné snímanie jedného tlačidla

Dostupnosť: Premier Farnell plc

Cena: 204/206 pre 100+ kusov: od 0,40 GBP po 0,44 GBP za kus

**STMicroelectronics STMPE321** – 3 kanálový kontrolér pre kapacitné snímanie

Vlastnosti:

- komunikačné rozhranie I2C
- 3 piny so všeobecným použitím
- 3 kapacitné vstupy

Dostupnosť: Mercateo AG

Cena: pre 100+ kusov: 17,18 CZK za kus

**STMicroelectronics STMPE1208S** - 12 kanálový kontrolér pre kapacitné snímanie

Vlastnosti:

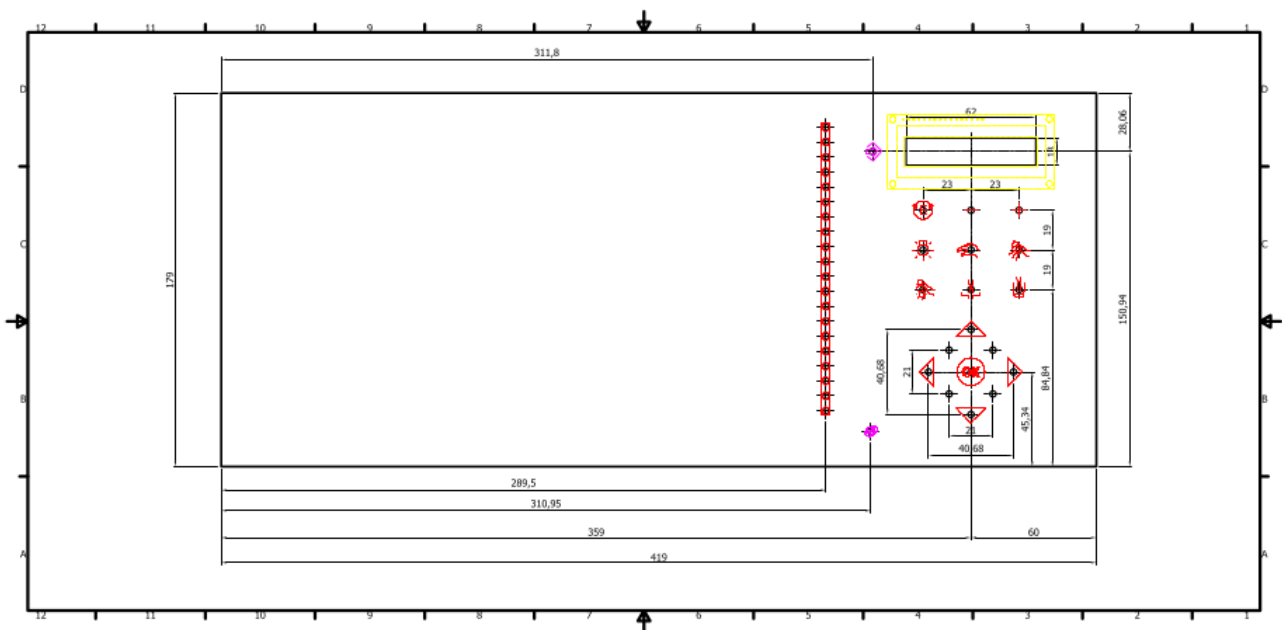
- komunikačné rozhranie I2C
- 12 pinov so všeobecným použitím
- 12 kapacitných vstupov

Dostupnosť: Premier Farnell plc

Cena: pre 100+ kusov: 1,21 GBP za kus

### 3.3 Konceptia klávesnice

Dizajnér dentálnej súpravy vyžaduje grafickú podobu klávesnice s precíznou realizáciou podľa obrázka 3.1 a obrázka 3.2. Súprava dostala ocenenie „cena za dizajn“. Na obrázku 3.3 je vysvetlenie funkcií jednotlivých tlačidiel.



Obrázok 3.1: Konceptia klávesnice s rozmermi



*Obrázok 3.2: Dizajn klávesnice*

		MENU
		plus
		mínus
		reverz otáčiek mikromotora
		vypínač operačného svietidla
		vypínač osvetlenia nástroja
		ručný mód, režim ovládania mikromotora
		prepínač chladenia nástroja
		oplach misy
		plnenie pohára
		OK tlačítko potvrdenia *program kresla
		kreslo hore *program kresla
		kreslo dole *program kresla
		operadlo chrbta dozadu *program kresla
	operadlo chrbta dopredu *program kresla	
	zamknutie klávesnice	
	<p>Podsvietenie symbolov sa mení v závislosti od aktuálnej práce zo súpravou. Vždy sú rozsvietené iba aktívne tlačidlá. Stlačenie tlačidla je indikované zvýšeným jasom ikony.</p>	

Obrázok 3.3: Vysvetlenie funkcií jednotlivých tlačidiel

## 4 Elektrické riešenie klávesnice

### 4.1 Kapacitné senzory [6]

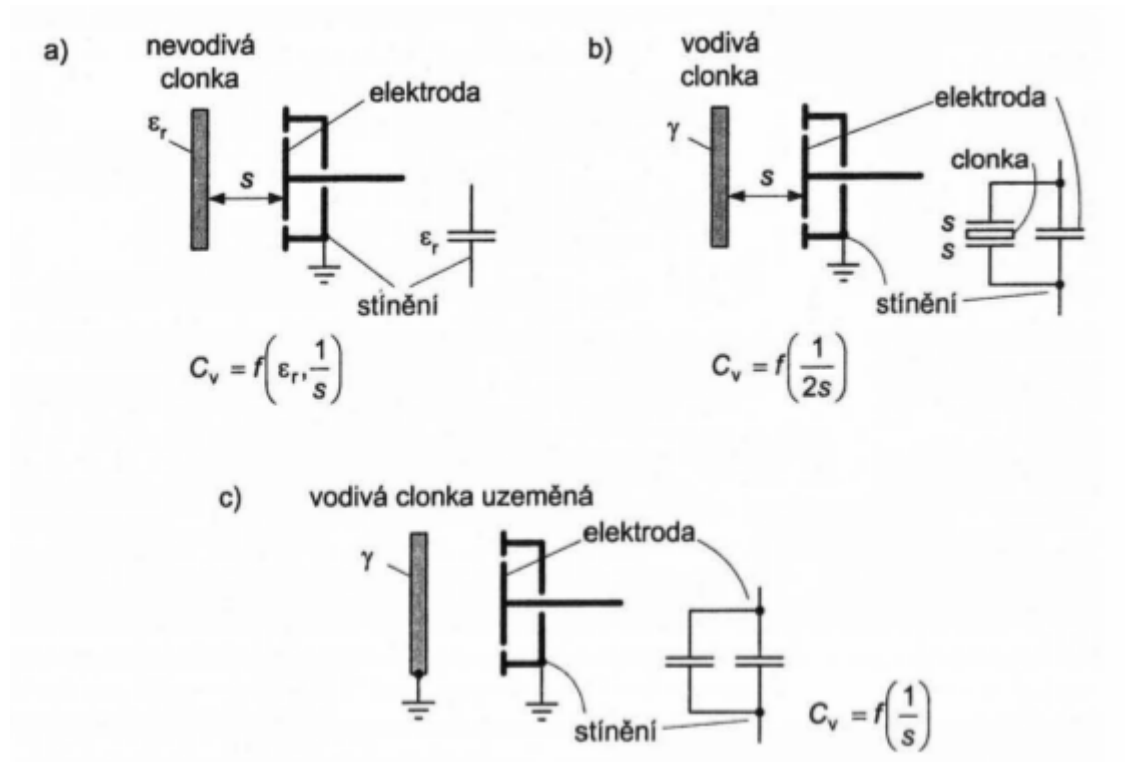
Sú vhodné pre meranie veličín ovplyvňujúcich kapacitu kondenzátoru, tj. geometriu elektród (plochu  $S$  a vzdialenosť  $d$ ) a permitivitu  $\varepsilon$  priestoru, v ktorom sa uzatvára elektrické pole kondenzátoru. U kontaktných kapacitných sensorov je meraný objekt spojený s pohyblivou elektródou, ktorá je súčasťou senzoru. Bezkontaktné kapacitné senzory detekujú prítomnosť objektov z deformácie elektrického poľa.

#### 4.1.1 Bezkontaktné kapacitné senzory

##### Technické riešenie

Tieto senzory detekujú priblíženie vodivých i nevodivých predmetov. Označujú sa ako senzory priblíženia (proximity sensor) alebo polohové spínače (proximity switch). Obvykle obsahujú kruhovú elektródu vnútri valcového tieniaceho puzdra (obr. 4.1). Priblížením clonky (meraného objektu) sa mení kapacita vnútornej elektródy voči krytu. Môžu nastať tieto prípady:

- nevodivá clonka: kapacita sa mení len zmenou permitivity. Spínacia vzdialenosť je malá;
- vodivá neuzemnená clonka: paralelne k základnej kapacite pribudne sériová kombinácia dvoch kapacít (stredná elektróda-clonka, clonka-tienenie). Zmena kapacity je väčšia;
- vodivá uzemnená clonka: paralelne k základnej kapacite pribudne ďalšia kapacita. Zmena kapacity je najväčšia.

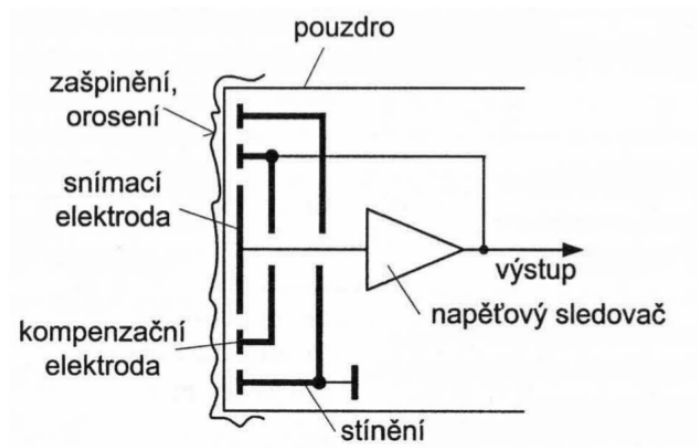


Obrázok 4.1: Kapacitný bezkontaktný senzor priblíženia [6]

Kompenzácia znečistenia alebo orosenia čelnej plochy sa dosahuje:

- S použitím aktívneho tienenia (obr. 4.2). Prídavná tieniaca (kompenzačná) elektróda je umiestnená medzi strednou elektródou a puzdro. Napätovým sledovačom je držaná na potenciále strednej elektródy. Stredná elektróda je tak obklopená krúžkom rovnakého potenciálu, nemôžu teda z nej tiecť po povrchových nečistotách žiadne kapacitné prúdy. Kapacitné prúdy samozrejme tečú medzi aktívnym tienením a puzdrom, to je ale mimo merací obvod.
- S použitím kompenzačných algoritmov, ktoré reagujú len na rýchle zmeny kapacity

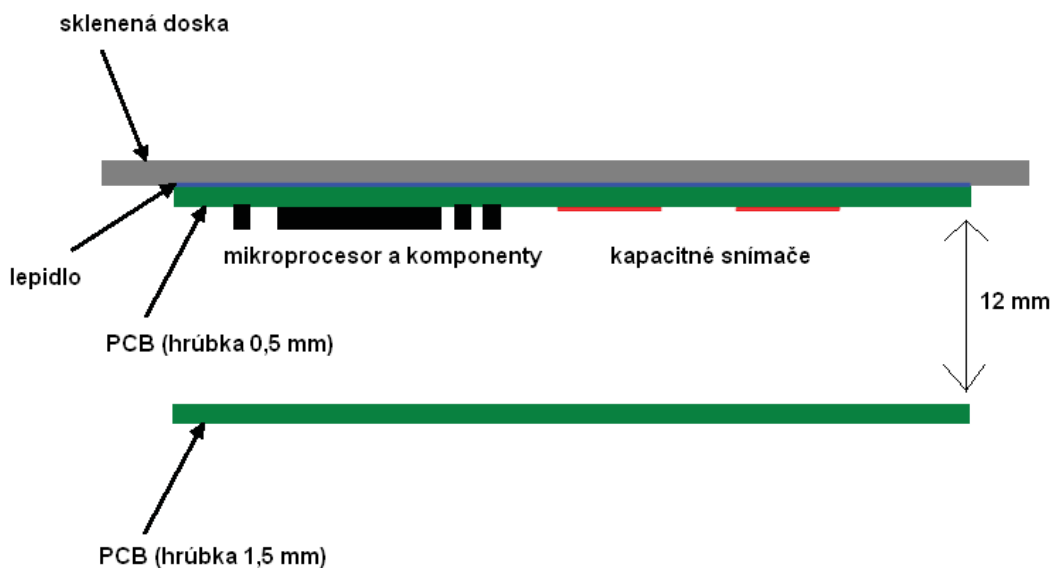
V uvedenom riešení použijeme kompenzáciu s využitím kompenzačných algoritmov.



Obrázok 4.2: Aktívne tienenie [6]

### Technologické riešenie [7]

Ide o 2-vrstvovú konštrukciu dosky plošných spojov. V tejto konštrukcii je mikrokontrolér a ostatné komponenty umiestnené na spodnej strane dosky plošných spojov. Sensorové elektródy sú umiestnené na vrchnej strane.



Obrázok 4.3: Technológia riešenia s použitím kapacitných senzorov



## 4.2 Výber vhodného riešenia (obvodu)

Obvod sme vybrali na základe:

- ceny
- realizovateľnosti
- technologickej náročnosti

Zvolili sme riešenie s použitím obvodu: **PIC16F722** v puzdre **SOIC**.

Tento obvod má prijateľnú cenu, výroba elektroniky nebude technologicky a cenovo náročná kvôli rozmerom pinov a vzdialeností medzi pinmi a vďaka tomu, že použijeme dva obvody bude jednoduchšie aj pripojenie pinov mikrokontroléru pre kapacitné snímanie a snímačov.

Zvolený obvod má 8 kapacitných senzorov, keďže sa požaduje 15 kapacitných senzorov, budeme potrebovať dva rovnaké obvody. Softvér bude identický pre oba obvody z dôvodu jednoduchosti výroby. Obvod má výkonnú CPU a komunikačné rozhrania.

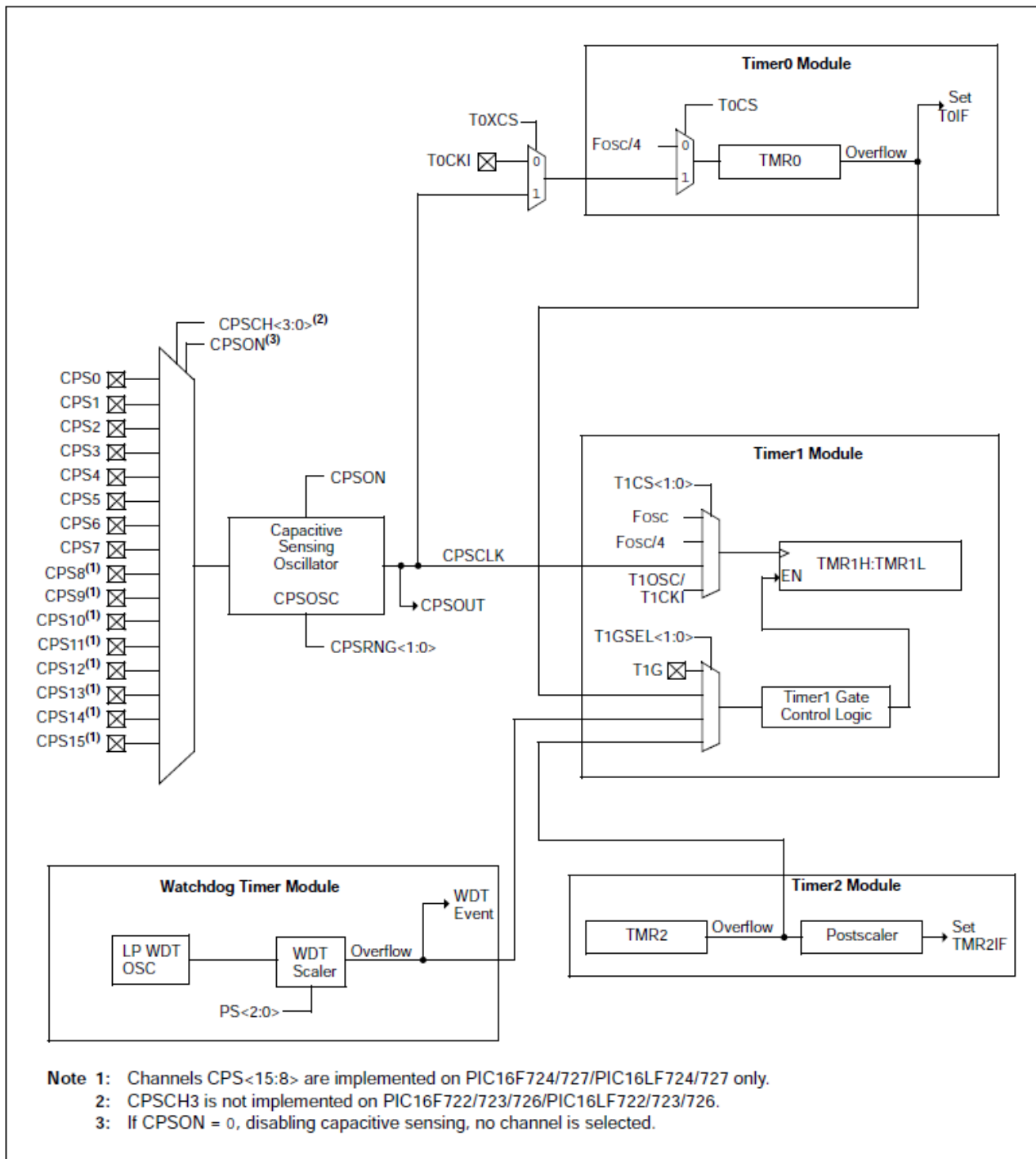
## 4.3 Modul kapacitného snímania obvodu PIC16F722 [5]

V tejto kapitole budeme popisovať modul kapacitného snímania obvodu PIC16F722.

Modul kapacitného snímania umožňuje interakciu s koncovým užívateľom bez mechanického rozhrania. V typickej aplikácii, je pripojený na kapacitnú plochu na doske plošných spojov (PCB), ktorá je elektricky izolovaná od koncového užívateľa. Ak koncový užívateľ položí prst na sklenený panel v mieste ikony, pridá sa kapacitná záťaž a spôsobí zmenu frekvencie v module kapacitného snímania. Modul vyžaduje softvér a aspoň jeden časovač na určenie zmeny vo frekvencii.

Hlavné možnosti tohto modulu:

- analógový multiplexor na monitorovanie viacerých vstupov
- oscilátor kapacitného snímania
- viacero časovačov
- softvérové riadenie
- prevádzka počas režimu Sleep



Obrázok 4.4: Bloková schéma modulu kapacitného snímania

### 4.3.1 Analógový multiplexor

Modul kapacitného snímania môže monitorovať až 16 vstupov. Tieto vstupy sú definované ako CPS<15:0>. Na určenie či nastala zmena frekvencie musí užívateľ:

- vybrať príslušný CPS pin nastavením CPSCH<3:0> bitov v registri CPSCON1
- nastaviť príslušný bit ANSEL
- nastaviť príslušný bit TRIS
- spustiť softvérový algoritmus

Výber pinu CPSx kým je modul povolený spôsobí, že oscilátor kapacitného snímania bude na pine CPSx. Chyba pri nastavení príslušného ANSEL a TRIS bitu môže spôsobiť zastavenie oscilátora a môže viesť k chybnému určeniu frekvencie.

### 4.3.2 Oscilátor kapacitného snímania

Oscilátor kapacitného snímania pozostáva z konštantného zdroju prúdu a konštantnej prúdovej výpuste pre vyrobenie trojuholníkového priebehu. Bit CPSOUT v registri CPSCON0 ukazuje stav oscilátora kapacitného snímania, či sa prúd znižuje, alebo zvyšuje. Oscilátor je navrhnutý na budenie kapacitnej záťaže a aj aby bol zdrojom hodín pre Timer0 alebo Timer1. Oscilátor má 3 rôzne nastavenia prúdu definované v CPSRNG<1:0> registra CPSCON0. Rôzne nastavenia prúdu pre oscilátor slúžia pre:

- maximalizácia počtu periód časovača na fixnú časovú základňu
- maximalizácia rozdielu v počte periód časovača počas zmeny frekvencie

### 4.3.3 Zdroje časovania

Na meranie zmeny frekvencie oscilátora pre kapacitné snímanie je vyžadovaná fixná časová základňa. Pre periódu fixnej časovej základne je oscilátor používaný ako zdroj hodín pre Timer0 alebo Timer1. Frekvencia oscilátora je rovná počtu periód časovača delených periódou fixnej časovej základne.

### 4.3.4 Fixná časová základňa

Na meranie frekvencie oscilátora pre kapacitné snímanie je nutná fixná časová základňa. Každý zdroj časovania, alebo softvérová slučka môže byť použitá na stanovenie fixnej časovej základne. To, ktorú metódu použiť je na koncovom používateľovi.

#### Timer0

Pre výber časovača Timer0 ako zdroj časovania pre modul kapacitného snímania je potrebné:

- nastaviť bit T0XCS registru CPSCON0
- vynulovať bit T0CS registru OPTION

Po výbere časovača Timer0 bude oscilátor kapacitného snímania zdrojom hodín pre Timer0.

#### Timer1

Pre výber časovača Timer1 ako zdroj časovania pre modul kapacitného snímania musíme nastaviť TMR1CS<1:0> registra T1CON na „11“. Po výbere časovača Timer1 bude oscilátor kapacitného snímania zdrojom hodín pre Timer1. Pretože modul časovača Timer1 má gate control, vytváranie časovej základne pre meranie frekvencie môže byť zjednodušené použitím:

- príznaku pretečenia časovača Timer0
- príznaku pretečenia časovača Timer2
- príznaku pretečenia WDT

Je doporučené, aby jeden z týchto príznakov spolu s módom prepínania na bráne Timer1, sa použil na vytvorenie fixnej časovej základne vyžadovanej časťou softvéru pre modul kapacitného snímania.

## 4.4 Obvody podsvietenia

Na podsvietenie klávesnice používame dva typy LED diód - LED 1206 BLUE 45/130° na osvetlenie tlačidiel a L-KP-2012QBC-D na osvetlenie deliacej čiary na klávesnici. Na napájanie osvetlovacích LED diód používame dva spínacie regulátory MC34063A.

### 4.4.1 LED dióda LED 1206 BLUE 45/130° [9]

Technické parametre:

- puzdro – 1206
- farba svetla – modrá
- vyžarovací uhol – 130°
- $U_f$  (napätie v priepustnom smere) – 3,3 V
- $I_f$  (prúd v priepustnom smere) – 20 mA
- $I_v$  (svietivosť) – 45 mcd
- dominantná vlnová dĺžka – 470 nm
- špičková vlnová dĺžka – 468 nm

### 4.4.2 LED dióda L-KP-2012QBC-D [10]

Ide o podobnú LED diódu ako v predchádzajúcom bode, ale táto je v prevedení SMD.

Technické parametre:

- puzdro – 0805
- farba svetla – modrá
- vyžarovací uhol – 120°
- $U_f$  (napätie v priepustnom smere) – 3,3 V
- $I_f$  (prúd v priepustnom smere) – 20 mA
- $I_v$  (svietivosť) – 100 mcd
- dominantná vlnová dĺžka – 470 nm

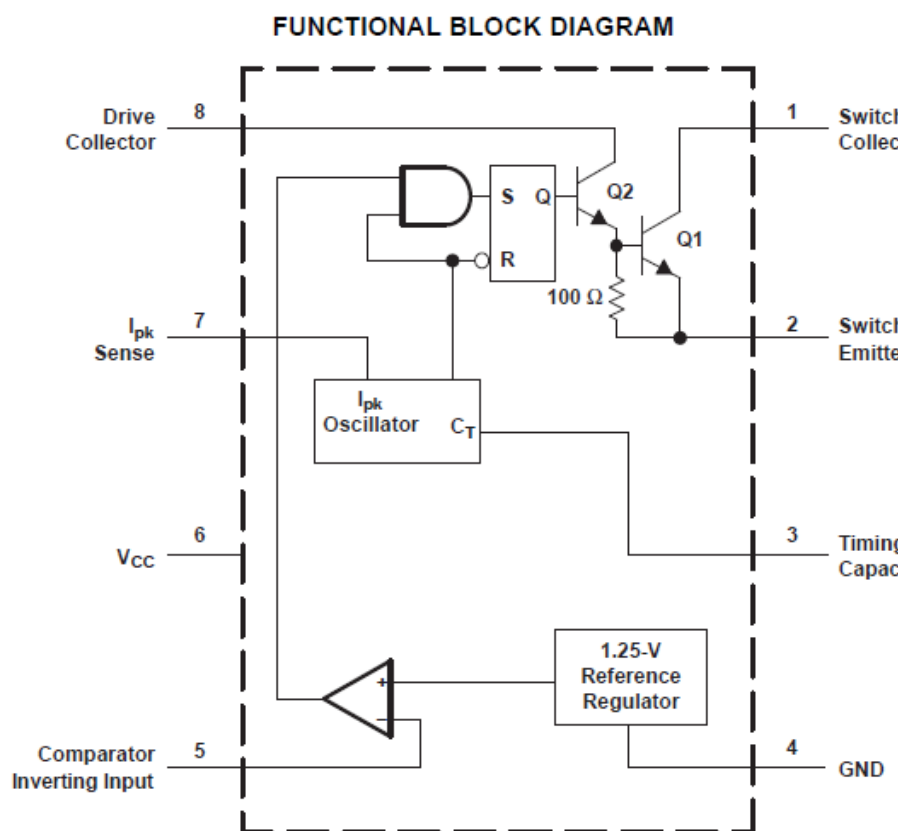
### 4.4.3 Spínací regulátor MC34063A [11]

MC34063A je jednoducho použiteľný integrovaný obvod, ktorý obsahuje všetky primárne obvody potrebné na postavenie jednoduchého DC-DC konvertoru. Toto zariadenie primárne pozostáva z internej referencie s teplotnou kompenzáciou, komparátoru, oscilátoru, PWM kontroléru s aktívnym

obmedzením prúdu, budiča a vysoko prúdového výstupného prepínača. Toto zariadenie teda vyžaduje minimum externých komponentov na tvorbu konvertorov v topológiach so zvýšením prúdu, znížením prúdu a invertujúcich.

Vlastnosti:

- rozsah vstupného napätia – od 3 V do 40 V
- prúd výstupného prepínača – do 1,5 A
- nastaviteľné výstupné napätie
- frekvencia oscilátoru – do 100 kHz
- presnosť internej referencie – 2 %
- skratové obmedzenie prúdu
- nízky pohotovostný prúd



Obrázok 4.5: Blokový diagram prepínacieho regulátoru MC34063A [11]

## 4.5 Riadenie LED diód pre podsvietenie

Riadenie je pomocou mikrokontroléra PIC16F722, ktorý zároveň realizuje kapacitné senzory a ku ktorého pinom sú jednotlivé LED diódy pripojené.

LED diódy sú napájané dvomi spínacími regulátormi MC34063A (z dôvodu výkonu), ktoré budú mať výstupné napätie 3,3 V a prúd 20 mA, aby sme splnili požiadavky LED diód. Prídavné komponenty a ich hodnoty sme zistili pomocou programu MC34063A design tool, ktorý je dostupný na adrese [12].

## 4.6 Zapojenie pinov mikrokontroléra

Mikrokontrolér PIC16F722 má k dispozícii 25 vstupno/výstupných pinov pre všeobecné použitie. Podľa toho, ktoré periférie sú povolené, niektoré piny nemusia byť dostupné ako piny pre všeobecné použitie.

Piny pre všeobecné použitie sú rozdelené do troch portov po 8 pinov. Piny z portu A majú názov RA0-RA7, piny z portu B sú RB0-RB7 a piny z portu C sú RC0-RC7.

Por. číslo	Použitý ako	Vstup/výstup	Popis zapojenia
1	VPP	Výstup	Naprogramované napätie pre In-circuit programovanie
2	RA0	Výstup	Ovládanie svietiacej diódy D25
3	RA1	Výstup	Ovládanie svietiacej diódy D22
4	RA2	Výstup	Ovládanie svietiacej diódy D21
5	RA3	Výstup	Ovládanie svietiacej diódy D20
6	CPS6	Vstup	Vstup kapacitného senzora 7
7	-	-	Nezapojený
8	VSS	Vstup	Uzemnenie
9	RA7	Výstup	Ovládanie svietiacej diódy D27
10	RA6	Výstup	Ovládanie svietiacej diódy D26
11	RC0	Výstup	Ovládanie svietiacej diódy D24
12	RC1	Výstup	Ovládanie svietiacej diódy D23
13, 14, 15, 16	-	-	Nezapojené
17	TX	Výstup	Vysielač AUSART
18	RX	Vstup	Prijímač AUSART
19	VSS	Vstup	Uzemnenie
20	VDD	Vstup	Napájacie napätie
21	CPS0	Vstup	Vstup kapacitného senzora 1
22	CPS1	Vstup	Vstup kapacitného senzora 2
23	CPS2	Vstup	Vstup kapacitného senzora 3
24	CPS3	Vstup	Vstup kapacitného senzora 4
25	CPS4	Vstup	Vstup kapacitného senzora 5
26	CPS5	Vstup	Vstup kapacitného senzora 6
27	ICSPCLK	Vstup	Vstup hodinového signálu pri In-circuit programovaní
28	ICSPDAT	Vstup	Vstup sériových dát pri In-circuit programovaní

Tabuľka 4.1: Zapojenie pinov mikrokontroléra číslo 1

Por. číslo	Použitý ako	Vstup/výstup	Popis zapojenia
1	VPP	Výstup	Naprogramované napätie pre In-circuit programovanie
2	RA0	Výstup	Ovládanie svietiacej diódy D28
3	RA1	Výstup	Ovládanie svietiacej diódy D29
4	RA2	Výstup	Ovládanie svietiacej diódy D30
5	RA3	Výstup	Ovládanie svietiacej diódy D33
6	CPS6	Vstup	Vstup kapacitného senzora 14
7	CPS7	Vstup	Vstup kapacitného senzora 15
8	VSS	Vstup	Uzemnenie
9	RA7	Výstup	Ovládanie svietiacej diódy D31
10	RA6	Výstup	Ovládanie svietiacej diódy D32
11	RC0	Výstup	Ovládanie svietiacej diódy D34
12	RC1	Výstup	Ovládanie svietiacej diódy D35
13	RC2	Výstup	Výstup asynchrónneho prenosu dát
14	RC3	Vstup	Vstup asynchrónneho prenosu dát
15, 16	-	-	Nezapojené
17	TX	Výstup	Vysielač AUSART
18	RX	Vstup	Prijímač AUSART
19	VSS	Vstup	Uzemnenie
20	VDD	Vstup	Napájacie napätie
21	CPS0	Vstup	Vstup kapacitného senzora 8
22	CPS1	Vstup	Vstup kapacitného senzora 9
23	CPS2	Vstup	Vstup kapacitného senzora 10
24	CPS3	Vstup	Vstup kapacitného senzora 11
25	CPS4	Vstup	Vstup kapacitného senzora 12
26	CPS5	Vstup	Vstup kapacitného senzora 13
27	ICSPCLK	Vstup	Vstup hodinového signálu pri In-circuit programovaní
28	ICSPDAT	Vstup	Vstup sériových dát pri In-circuit programovaní

Tabulka 4.2: Zapojenie pinov mikrokontroléra číslo 2

## 4.7 In-circuit sériové programovanie (ICSP) [5]

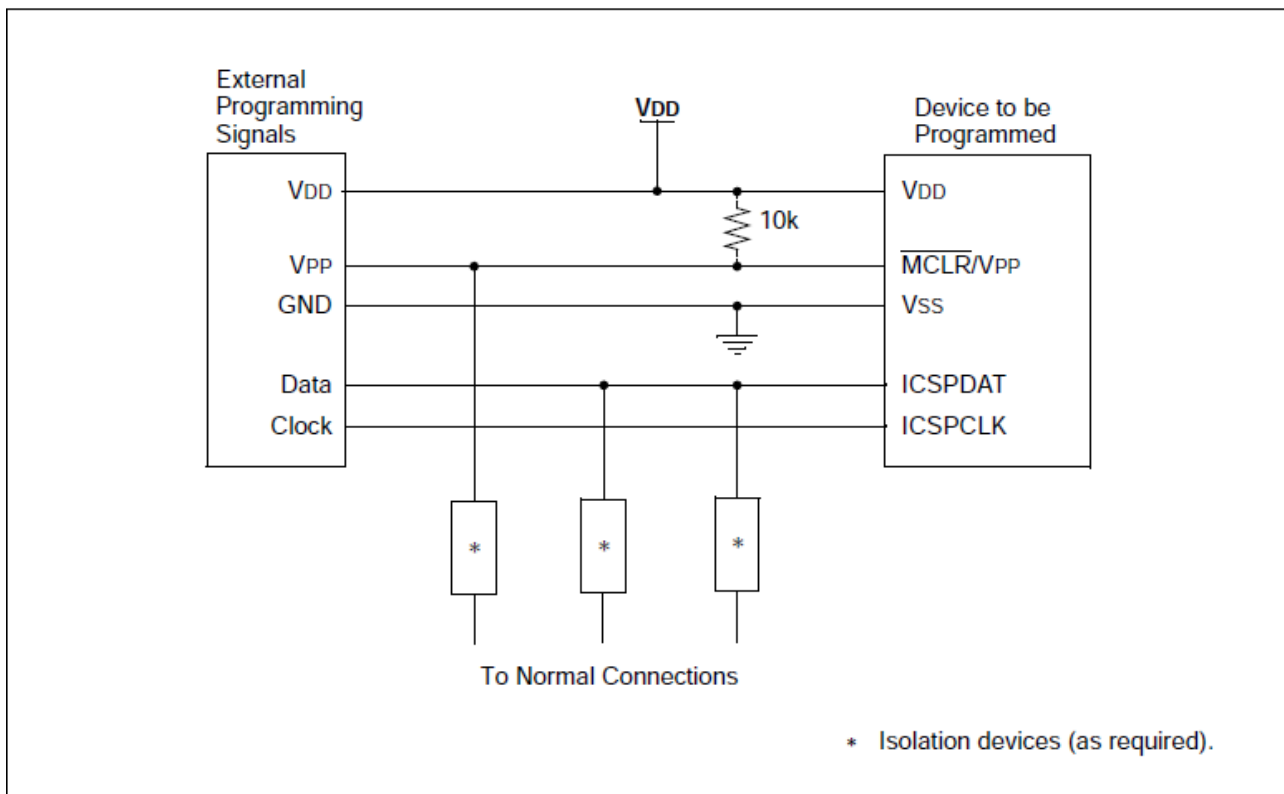
ICSP programovanie umožňuje zákazníkovi vyrábať dosky s obvodmi s nenaprogramovanými zariadeniami. Programovanie môže byť vykonané až po zostavení dosky a umožňuje aby bolo zariadenie naprogramované s najnovším firmwarom alebo vlastným firmwarom. K ICSP programovaniu je potrebných 5 pinov:

- ICSPCLK
- ICSPDAT



- $\overline{\text{MCLR/VPP}}$
- VDD
- VSS

Zariadenie je uvedené do módu programovanie/verifikácia držaním pinov ICSPCLK a ICSPDAT na nule a potom zvýšením napätia na MCLR/VPP z 0 V na VPP. V móde programovanie/verifikácia sú programová pamäť, používateľské ID a konfiguračné slová naprogramované sériovou komunikáciou. Pin ICSPDAT je obojsmerný vstupno-výstupný pin použitý na presun sériových dát a pin ICSPCLK je vstup hodinového signálu.



Obrázok 4.6: Typické pripojenie pre ICSP programovanie [5]

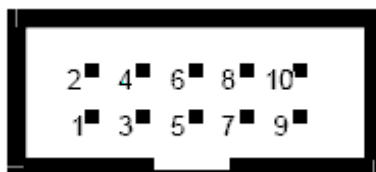
## 4.7.1 ICSP pomocou programátora Elnec SmartProg2

### ISP konektor programátora Elnec SmartProg2 [8]

- 10-pinový konektor s ochranou proti prepólovaniu
- 5 TTL pindrivery, poskytujúce H, L, CLK, pull-up, pull-down; úroveň H nastaviteľná od 1.8V do 5V k práci so všetkými (vrátane nízkonapäťovými) zariadeniami.
- 1x VCCP napätie (rozsah 2V..7V/100mA) a 1x VPP napätie (rozsah 2V..25V/50mA)
- programovacie napätie (VCCP) so schopnosťou korigovať napájacie napätie cieľového obvodu smerom hore aj dole a detekciou napätia

- programátor nie je určený k napájaniu cieľovej aplikácie pomocou VCCP pinu

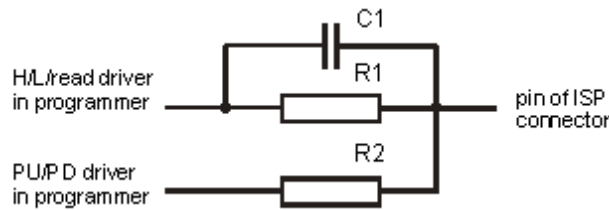
### Popis pinov ISP konektora



*Obrázok 4.7: ISP konektor  
programátora Elnec  
SmartProg2 [8]*

- 1 - VDD power supply to programmed device
- 2 - Nepripájať!
- 3 - Nepripájať!
- 4 - Nepripájať!
- 5 - Nepripájať!
- 6 - Serial Data In/Out (RB7)
- 7 - GND (VSS)
- 8 - Serial Data Clock (RB6)
- 9 - GND (VSS)
- 10 - MCLR/VPP to programmed device





Obrázok 4.9: Spôsob zapojenia logických signálov konektora ISP programátora SmartProg [8]

Hodnoty komponentov z obrázka C1 1nF, R1 1k3, R2 22k

Zapojenie a hodnoty sú tu uvedené na pochopenie a na správny výpočet hodnôt rezistorov oddeľujúcich obvod programovaný cez ISP od cieľového systému.

## 4.8 Sériová komunikácia pomocou RS-232 [13]

Rozhranie RS-232 je jedným z najstarších sériových rozhraní používaných v súčasnej dobe. Toto rozhranie je nesymetrické, o logickej úrovni signálu teda rozhoduje napätie signálového vodiča voči zemnému potenciálu (GND). Aby sa predišlo rôznym zemným potenciálom (rôznej referenčnej úrovne) medzi vysielateľom a prijímačom, je jeden z vodičov rozhrania vyhradený pre prepojenie lokálnych zemí.

Rozhranie RS-232 je dvojbodové, slúži teda pre prepojenie dvoch staníc navzájom. Logické úrovne sú určené rozdielom potenciálov medzi signálovým vodičom a zemou.

Logická „0“ - napätie na signálovom vodiči vzhľadom na zem  $+3\text{ V} < U_0 < +15\text{ V}$

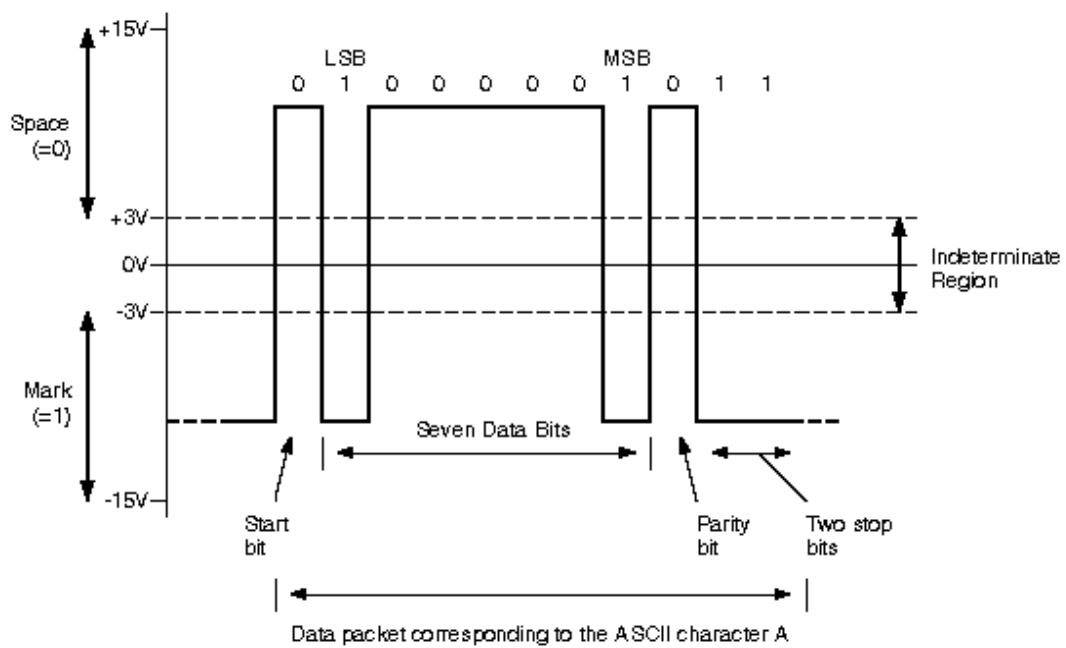
Logická „1“ - napätie na signálovom vodiči vzhľadom na zem  $-15\text{ V} < U_1 < -3\text{ V}$

Napätie naprázdno nesmie prekročiť  $\pm 25\text{ V}$ .

Kódovanie bitov je typu NRZ a použitý je arytmičný prenos dát. Prenosové rýchlosti boli pôvodne definované do 20 kb/s, v súčasnosti sa používajú nasledujúce prenosové rýchlosti (v b/s):

50, 75, 110, 150, 300, 600, **1200, 2400, 4800, 9600, 19200, 38400, 57600, 115200**, 230400, 460800, 921600, pričom tučne sú označené bežne používané komunikačné rýchlosti. Maximálna odporúčaná dĺžka vedenia je 15 m pri rýchlosti do 20 kb/s.

Na obrázku 4.10 je zobrazený prenos jedného bytu dát pomocou RS-232. Z obrázka vidíme, že kludová úroveň pri RS-232 je logická „1“, nasleduje štart bit v logickej „0“, ďalej 7 dátových bitov (môže byť 7, 8 alebo 9 dátových bitov, najčastejšie sa používa 8 dátových bitov). Nasleduje nepovinný paritný bit a jeden alebo dva stop bity, ktoré majú úroveň logickej „1“.



Obrázok 4.10: Asynchrónny prenos jedného bytu dát pomocou RS232 [14]

## 5 Riešenie programu klávesnice

V tejto kapitole budeme písať o riešení programu pre mikrokontrolér, od výberu programovacieho jazyka a vývojového prostredia, cez počítačnú konfiguráciu mikrokontroléra po hlavnú časť programu.

### 5.1 Voľba vývojového prostredia

Zvolili sme prostredie Microchip MPLAB IDE – toto vývojové prostredie je k dispozícii bezplatne, je veľmi jednoduché a prehľadné a pracuje pod operačným systémom Microsoft Windows. Zahŕňa voľne dostupné softwarové komponenty pre vývoj softwaru, emuláciu hardwaru a ladenie. Toto vývojové prostredie podporuje jazyky C a Assembler. Prostredie MPLAB IDE používame vo verzii 8.92, pretože táto verzia podporuje nami vybraný mikrokontrolér.

### 5.2 Výber programovacieho jazyka

Zvolili sme programovací jazyk assembler, pretože umožní efektívnejšie využiť malý pamäťový priestor mikrokontroléra a využívať bezplatný simulátor činnosti programu.

### 5.3 Počítačná konfigurácia mikrokontroléra

Súčasťou mikrokontroléra sú určité špeciálne obvody, ktoré riešia potreby aplikácii pracujúcich v reálnom čase. Týmto sa odlišujú mikrokontroléry od bežných procesorov. Informácie v tejto kapitole sú čerpané z [5].

#### 5.3.1 Konfigurovateľné obvody

Mikrokontrolér má k dispozícii viacero konfigurovateľných obvodov, z nich budeme využívať v našej aplikácii nasledovné:

- výber oscilátora
- obvod Watchdog Timer
- obvod Power-up Timer
- ochrana kódu
- Brown-out Reset

V riešenej aplikácii potrebujeme nakonfigurovať tieto bity:

- `_INTRC_OSC_NOCLKOUT` – použitie interného oscilátora
- `_WDT_ON` – zapnutie obvodu Watchdog Timer
- `_PWRTE_ON` – zapnutie obvodu Power-up Timer
- `_CP_ON` – nastavenie ochrany pamäte programu
- `_BOREN_OFF` – zakázanie Brown-out Reset

Konfiguračné bity sú v registroch CONFIG1 a CONFIG2 a programujú sa pomocou príkazu „\_\_CONFIG“.

### 5.3.2 Výber oscilátora

Mikrokontroléry rodiny PIC16F72X majú široký výber oscilátorov, na výber je 8 rôznych oscilačných módov. V prípade nášho zadania môžeme vybrať interný oscilátor ako zdroj hodinového signálu. Je to z toho dôvodu, že toto riešenie nevyžaduje žiaden pin mikrokontroléra. Interný oscilátor štandardne po resete zariadenia pracuje na frekvencii 8 MHz. Táto frekvencia vyhovuje pre potreby našej aplikácie.

### 5.3.3 Watchdog Timer

Watchdog timer je voľne bežiaci oscilátor integrovaný na čipe mikrokontroléra, ktorý nepotrebuje k svojmu chodu žiadne externé komponenty. Hlavnou úlohou tohto obvodu je zabrániť chybovým stavom chodu mikrokontroléra (napr. zacyklenie) tým, že sa musí pravidelne nulovať. Ak sa toto nestane, Watchdog generuje RESET zariadenia.

Watchdog timer má tieto vlastnosti:

- zdieľa 8 bitový prescaler s časovačom Timer0
- perióda obvodu je od 17 ms po 2.2 s
- časovú základňu si vytvára z integrovaného 31 kHz oscilátora

### 5.3.4 Power-up Timer

Obvod Power-up Timer zabezpečuje fixné oneskorenie dĺžky 64 ms po resete zariadenia alebo po Brown-out Reset. Pracuje s interným oscilátorom obvodu Watchdog Timer. Toto oneskorenie umožňuje stabilizáciu napájacieho napätia VDD pred štartom programu. Power-up Timer by mal byť vždy povolený keď je povolený Brown-out Reset.

### 5.3.5 Brown-out Reset

Ak  $V_{DD}$  klesne pod  $V_{BOR}$  na čas dlhší ako  $T_{BOR}$ , Brown-out zresetuje zariadenie. Ak  $V_{DD}$  klesne pod  $V_{BOR}$  kým beží Power-up Timer, čip pôjde naspäť do stavu Brown-out Reset a Power-up Timer bude znovu nainicializovaný. Keď  $V_{DD}$  stúpne nad  $V_{BOR}$ , Power-up Timer vykoná 64 ms reset.

## 5.4 Počiatočné nastavenia

### 5.4.1 Vstupy a výstupy

V trojstavových registroch TRISA, TRISB a TRISC pre porty mikrokontroléra PORTA, PORTB a PORTC, je nutné nastaviť, ktoré piny mikrokontroléra pre všeobecné použitie (GPIO) budú použité

ako vstupné a ktoré ako výstupné. Zápisom hodnoty jedna na príslušný bit určíme pin ako vstup, zápisom nuly bude pin výstupný. Popis vstupných a výstupných pinov je v kapitole Semestrálny projekt 1.

## **5.4.2 Adresovateľný univerzálny synchrónny/asynchrónny prijímač/vysielač**

Ide o sériové vstupno/výstupné komunikačné rozhranie. AUSART, tiež známy ako sériové komunikačné rozhranie SCI, môže byť nakonfigurované ako full-duplex asynchrónny systém alebo ako half-duplex synchrónny systém.

Modul AUSART má nasledovné vlastnosti:

- full-duplex asynchrónne prijímanie a vysielanie
- dvoj-znakový vstupný buffer
- jedno-znakový výstupný buffer
- programovateľná 8 alebo 9 bitová dĺžka znaku
- detekcia adresy v 9 bitovom móde
- detekcia chyby pretečenia vstupného bufferu
- detekcia chyby prijatého znaku
- half-duplex synchrónny master
- half-duplex synchrónny slave
- prevádzka v režime Sleep

Tento obvod budeme využívať na komunikáciu klávesnice so stolíkom zubnej súpravy, a zároveň aj pre komunikáciu medzi mikrokontrolérmi klávesnice navzájom.

Funkciu obvodu povolíme nastavením bitu SPEN registru RCSTA na jednotku. Vysielanie povolíme nastavením bitu TXEN v registru TXSTA. V tom istom registri nastavíme asynchrónny mód prenosu nulovaním bitu SYNC a nastavíme Baud rate na 9600 nastavením bitu BRGH v tom istom registri a nastavením hodnoty registra SPBRG podľa tabuľky v datasheete k mikrokontroléru. Prijímanie povolíme nastavením bitu CREN registru RCSTA na jednotku, a predtým odstránime prípadné pretečenie bufferu nulovaním bitu CREN.

## **5.4.3 Prerušenie**

Pre potreby našej aplikácie budeme využívať prerušenie pri prijatí znaku cez adresovateľný univerzálny synchrónny/asynchrónny prijímač/vysielač AUSART.

Globálne povolíme prerušenia zariadenia nastavením bitu PEIE registru INTCON na jednotku. Prerušenie z prijímača AUSART povolíme nastavením bitu RCIE registru PIE1.

## **5.5 Rozlíšenie mikrokontrolérov**

Keďže program je rovnaký pre oba použité mikrokontroléry, musíme v programe rozlíšiť, ktorá časť programu je pre mikrokontrolér označený na schéme (príloha B.4) U3 a ktorá pre mikrokontrolér s označením U4.



Rozlišujeme to tak, že všetky povely majú v prvom byte adresu mikrokontroléra, a to adresa 1 pre mikrokontrolér U4 a adresa 2 pre mikrokontrolér U3. Program je potom vetvený podľa adresy mikrokontroléra.

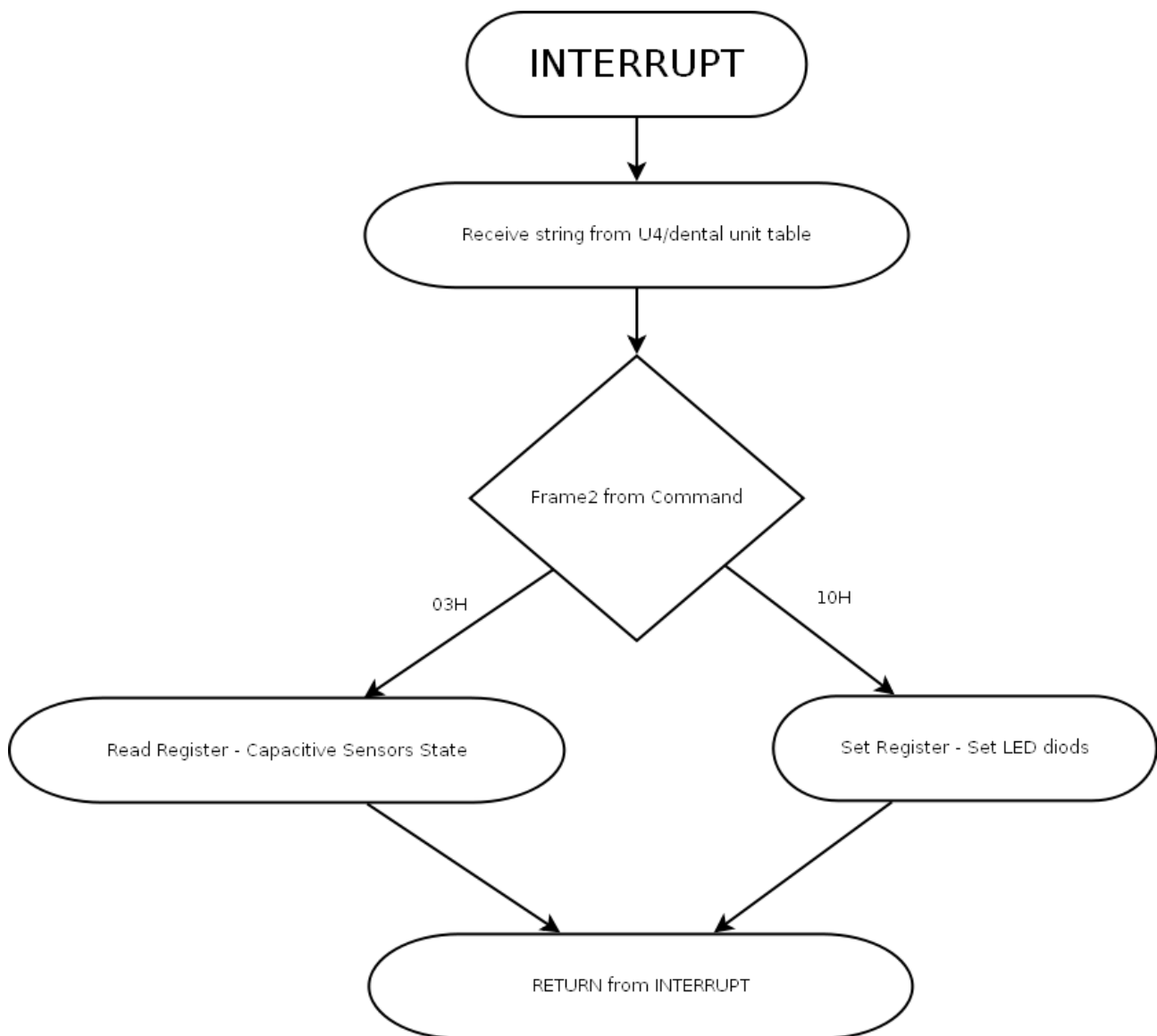
## 5.6 Algoritmus pre vyhodnotenie kapacitného snímania

Pri určení optimálneho algoritmu pre vyhodnotenie toho, ktorá klávesa je stlačená, sme si pomohli aplikačnou poznámkou [15]. My sme použili metódu Explicit Trip z aplikačnej poznámky, ktorá zisťuje stlačenie klávesy tak, že pre každý senzor počíta neustále priemernú hodnotu a keď sa dotkneme klávesy, hodnota sa skokom zmení a podľa veľkosti tejto zmeny môžeme určiť, že klávesa bola stlačená.

Oproti algoritmu z aplikačnej poznámky sme však vykonali niekoľko zmien:

- v aplikačnej poznámke počítajú priemernú hodnotu vždy, keď daná klávesa nie je identifikovaná ako stlačená a počítajú ju zo 16 hodnôt, my ju počítame pre jednoduchosť iba zo 4 hodnôt, ale počítame ju nie stále, ale práve vtedy, keď nám príde zo stolíka súpravy povel pre nastavenie svietenia LED diód, a počítame ju len pre senzory, ktoré sú aktuálne aktívne. Toto riešenie má výhodu v tom, že je odolné voči pomalým zmenám, napríklad, keď držíme prst na senzore, a pomaly s ním pohybujeme mimo senzor. Týmto sa zmení priemerná hodnota senzoru a senzor môže reagovať nesprávne. Avšak máme tu ten problém, že ak sa menia okolité podmienky, napríklad na senzor zasvieti slnko, alebo sa poleje vodou, priemerná hodnota sa nespočíta nová, ale zmení sa okamžitá hodnota senzora, a ten môže reagovať nesprávne. Preto je nutné toto riešenie ešte optimalizovať.
- v aplikačnej poznámke je hodnota výchylky oproti priemernej hodnote, pri ktorej sa javí senzor ako stlačený, rôzna pre každý kapacitný senzor, u nás pre jednoduchosť sme nechali túto hodnotu rovnakú pre každý senzor. Toto je treba ešte experimentálne overiť, lebo je možné, že ak stanovíme hodnotu pre každý senzor inú, tak zvýšime spoľahlivosť riešenia.
- v aplikačnej poznámke je do vzorca zarátaná aj hystéza (hodnota 64) a priemerná hodnota sa počíta nie pri každom prechode cyklom, ale iba raz za 3 prechody. My pre jednoduchosť toto nevyužívame.

## 5.7 Telo prerušenia



Obrázok 5.1: Vývojový diagram tela prerušenia

V prerušení musíme obslúžiť prijatie komunikačného reťazca bytov od stolíka zubnej súpravy. Komunikácia medzi stolíkom a klávesnicou je podrobne popísaná v prílohe A tejto práce.

Telo prerušenia začína zakázaním globálneho prerušenia nulovaním bitu GIE registru INTCON. Následne si uložíme W register a STATUS register do premenných, aby sme ich mohli obnoviť po ukončení obslužnej rutiny prerušenia. Keďže v okamihu prerušenia už máme prijatú prvú hodnotu, túto uložíme z registra RCREG do premennej *Frame1* a povolíme prijímanie nového bytu.

Následne zavoláme funkciu *RECEIVE\_String*, ktorá slúži na prijatie reťazca hexadecimálnych 8 bitových hodnôt. Počet prijatých hodnôt počítame pomocou premennej *counter*

a na uloženie znakov do premenných využívame nepriame adresovanie pomocou virtuálneho registra INDF a s tým, že ukazateľ FSR sa inicializuje na adresu prvého bytu premennej *Frame2* na uloženie prijatých dát a po prenose jednej hodnoty sa inkrementuje o 1 na nasledujúcu premennú.

Samotné prijímanie jednej hexadecimálnej hodnoty prebieha vo funkcii *receive\_STRING*. Na začiatku si inicializujeme premenné *TempV* a *DlyCnt*, ktoré budú slúžiť ako aktívne čakanie na prijatý byte. Potom povolíme prijatie bytu s kontrolou pretečenia nulovaním a následným nastavením bitu CREN registru RCSTA a skontrolujeme bit RCIF registru PIR1. Ak je RCIF na jednotke, znamená to, že byte bol prijatý a skopírujeme ho z registra RCREG do W registra. Ak je RCIF na nule, pokračujeme v cykle a čakáme na prijatý znak. Ak žiadny znak nepríde, nastavíme príznak ZERO registra STATUS na nulu, ak znak príde, nastavíme ZERO na jednotku.

Po návrate z funkcie *receive\_STRING* skontrolujeme príznak ZERO, podľa neho zistíme, či už boli prijaté všetky hodnoty, alebo pokračujeme v prijímaní. Ak boli už prijaté všetky hodnoty, pokračujeme na návessie *Silent*. Následne vypočítame CRC sumu zo všetkých prijatých bytov, a to tak, že pre prvý byte sa počíta CRC zavolaním funkcie *GenCRC* a pre nasledujúce byty sa CRC suma počíta už z predtým vypočítanej sumy a to zavolaním funkcie *GenCRC\_*. (použitý algoritmus CRC kalkulácie je uvedený v prílohe A) Ak CRC suma nesedí, teda ak výsledky v premenných *CRCaLO* a *CRCaHI* nie sú rovné nule, vraciame sa z prerušenia a nič nevykonáme. Ak naopak CRC sumy sú rovnaké, testujeme druhý prijatý byte, v premennej *Frame2*. Ak má tento byte hodnotu 03H, prijatý povel je čítanie registrov, a zavoláme funkciu *ReadREG*. Ak má hodnotu 10H, prijali sme povel na zápis registrov a zavoláme funkciu *PresetREG*. Následne sa vrátíme z prerušenia.

## 5.7.1 Funkcia ReadREG

Funkcia *ReadREG* má za úlohu prečítať dáta o tom, ktorý kapacitný senzor klávesnice je stlačený, teda v stave, keď sme priblížení s prstom, a tieto informácie odoslať naspäť stolíku dentálnej súpravy.

Na začiatku funkcie rozhodneme podľa prvého prijatého bytu, či ide o mikrokontrolér U3 alebo mikrokontrolér U4.

- Ak je správa prijatá mikrokontrolérom U4 (v programe označenom A1), je potrebné, aby sme poslali mikrokontroléru U3 (v programe označený A2) túto správu tiež. Preto odošleme ten istý povel, len s adresou 2. Na posielanie dát využijeme funkciu *TrnsByt\_9\_A1*.
- Po odoslaní dát čakáme na odpoveď od mikrokontroléra U3. Prijímanie dát zabezpečuje funkcia *RECEIVE\_96St*.
- Nasleduje kontrola skúšobnej sumy CRC, ak je suma nesprávna, vraciame sa z prerušenia. Ak je suma v poriadku, voláme funkciu *RESULT\_Data* a pomocou nej odošleme odpoveď do stolíka dentálnej súpravy.
- Ak je správa prijatá mikrokontrolérom U3 (v programe označenom A2), odošleme odpoveď rovnakú ako je prijatá správa, len so zmenenými bytami, ktoré obsahujú informácie o stave kapacitných senzorov (ST1, ST2 v prílohe A) mikrokontroléru U4 pomocou funkcií *TransB*.

## Funkcia TrnsByt\_9\_A1

Keďže piny, s ktorými je možné odosielať a prijímať dáta pomocou rozhrania AUSART, sú už použité na komunikáciu mikrokontroléra U4 so stolíkom dentálnej súpravy, na komunikáciu s mikrokontrolérom U3 využijeme piny pre všeobecné použitie RC2 a RC3 (zapojenie pinov mikrokontroléra je v Semestrálnej práci 1). Na týchto pinoch nie je žiadna hardwarová podpora prenosu dát a tak je nutné na programovať komunikačný protokol RS232 bezpomocne.

Funkcia má vstupné dáta vo W registry, tie uložíme do premennej *XmtRe*, následne odošleme štart bit s logickou úrovňou 0 na pine RC2. Keďže chceme komunikovať prenosovou rýchlosťou 9600 Baud (na túto hodnotu sme nastavili aj komunikačnú rýchlosť rozhrania AUSART), vložíme medzi každý odoslaný bit oneskorenie 104,2  $\mu$ s (9600 Baud = 9600 bit/s, to znamená, že 1 bit každých 104,2  $\mu$ s). Odosielanie dátových bitov prebieha rotáciou premennej *XmtRe* cez príznak CARRY (bit C registra STATUS), ak je CARRY na nule, pin RC2 nastavíme na nulu, ak je príznak na jednotke, nastavíme aj výstupný pin na jednotku. Nakoniec odošleme aj stop bity s logickou úrovňou 1.

## Funkcia RECEIVE\_96St

Táto funkcia slúži na prijatie reťazca dát pomocou asynchrónneho prenosu RS232 bez hardwarovej podpory. Na začiatku zavoláme funkciu *receive\_9*, ktorá prijme jeden byte dát. Skontrolujeme príznak ZERO aby sme zistili, či bol prenos úspešný. Ak bol úspešný, pokračujeme. Prvý prijatý byte uložíme do premennej *Frame1*. Pomocou nepriameho adresovania s registrami INDF a FSR ukladáme prijaté dáta do premennej *Frame2*, ktorá má veľkosť 12 bytov, takže sa tam zmestí celý prijatý reťazec hodnôt. Na prijatie každého bytu použijeme funkciu *receive\_9*. Ak už funkcia *receive\_9* neprijala žiadny byte, pokračujeme na návessie *Silent96* a generujeme CRC sumu pre prijatý reťazec dát. Na generovanie CRC využijeme už spomenuté funkcie *GenCRC* a *GenCRC\_*.

## Funkcia receive\_9

Funkcia slúži na prijatie jedného bytu dát pomocou asynchrónneho prenosu RS232 bez hardwarovej podpory.

Na začiatku čakáme v slučke na platný štart bit prenosu s logickou úrovňou 0. Po detekovaní štart bitu čakáme po dobu 1,5 bitu tak, aby sme nasledovné bity detekovali vždy v prostriedku. Testujeme hodnotu na pine RC3, ak je na pine RC3 logická úroveň 1, prijali sme bit s logickou úrovňou 1 a uložíme ho ako najviac významný bit premennej *RcvReg*. V ďalšom cykle rotujeme doprava premennú *RcvReg*, a v prípade, že na pine RC3 bola logická úroveň 0, na pozíciu najviac významného bitu premennej *RcvReg* sa dostane príznak CARRY, ktorý bol predtým nulovaný. Keďže chceme komunikovať prenosovou rýchlosťou 9600 Baud, vložíme medzi každý prijatý bit oneskorenie 104,2  $\mu$ s. Na konci testujeme, či je prijatý validný stop bit. Ak bol stop bit prijatý, úspešné ukončenie prijímania indikujeme nulovaním príznaku ZERO. Ak prenos neprebehol v poriadku, príznak ZERO sa nastaví na jednotku.

## **Funkcia RESULT\_Data**

Táto funkcia odosiela odpoveď z mikrokontroléra U4 do stolíka dentálnej súpravy. Táto odpoveď má tvar, ktorý je popísaný v prílohe A. Prvý odoslaný byte je adresa mikrokontroléra, keďže odosielame z U4, adresa bude 1. Premenné *ST1* a *ST2* obsahujú informácie o tom, ktorý kapacitný senzor je aktuálne stlačený, teda užívateľ s ním interaguje. Viac o týchto premenných a ako do nich nastavujeme hodnoty bude povedané v kapitole 1.8.3., funkcia *SENS\_ACTIVE* a *SENS\_NO\_ACTIVE*. Posielanie dát vykonávajú funkcie *CRC\_TRNS* pre odoslanie prvého bytu, *CRC\_TRN\_* pre odoslanie zvyšných bytov reťazca, a *TransB* pre odoslanie skúšobnej sumy CRC.

## **Funkcie CRC\_TRNS a CRC\_TRN\_**

Tieto funkcie majú vstupnú hodnotu vo W registry, ktorú si ukladajú do premennej *Msg*. Následne volajú funkciu *TransB* na odoslanie jedného bytu dát a napokon volajú funkciu pre generovanie CRC. Rozdiel medzi funkciami je v tom, že funkcia *CRC\_TRNS* volá pre generovanie CRC sumy funkciu *GenCRC* (počíta CRC pre prvý byte) a funkcia *CRC\_TRN\_* volá funkciu *GenCRC\_* (počíta CRC sumu pre nasledujúce byty).

## **Funkcia TransB**

Táto funkcia slúži na odoslanie bytu dát uloženého vo W registry z mikrokontroléra U4 do stolíka dentálnej súpravy, alebo z mikrokontroléra U3 do mikrokontroléra U4 pomocou rozhrania AUSART.

Na začiatku povolíme odosielanie dát nastavením bitu TXEN registru TXSTA. Následne vložíme dáta na odoslanie do registru TXREG a tým zahájime prenos. Čakáme v slučke dovtedy, kým nebude mať bit TXIF registru PIR1 hodnotu logickej 1, čo znamená, že prenos bol ukončený a TXREG je prázdny.

## **5.7.2 Funkcia PresetREG**

Táto funkcia má za úlohu nastaviť LED diódy osvetľujúce kapacitné senzory a indikujúce tlačidlá, ktoré sú aktívne v danom momente. LED diódy sa nastavujú povelom s kódom 10H, čo znamená zápis (podľa prílohy A) a informácie o tom, ktoré senzory budú aktívne a ktoré nie, sa nachádzajú v bytoch EN1 a EN2. Po úspešnom nastavení pošleme z mikrokontroléra správu o tom, že nastavenie prebehlo v poriadku (ACKNOWLEDGE).

Na začiatku nastavíme premenné *PORTA\_U4*, *PORTC\_U4*, *PORTA\_U3* a *PORTC\_U3*, ktoré budeme používať na nastavenie LED diód, na hodnotu FFH, čo znamená, že všetky LED diódy budú zhasnuté.

Následne rozhodneme podľa adresy mikrokontroléra, ak je adresa 1, jedná sa o mikrokontrolér U4 a voláme funkciu *SetLedU4*. Ak je adresa 2, jedná sa o mikrokontrolér U3 a zavoláme funkciu *SetLedU3*.

## **Funkcia SetLedU4**

V tejto funkcii prebieha aj počítanie priemernej hodnoty kapacitných senzorov pre mikrokontrolér U4 (keďže sme sa rozhodli, že priemerná hodnota kapacitných senzorov sa bude počítat' pri každom

nastavovaní LED diód) a zároveň prebieha nastavovanie LED diód senzorov podľa povelu zo stolíka dentálnej súpravy opäť pre U4.

Na výpočet priemernej hodnoty kapacitných senzorov zavoláme funkcie *PRIEMER\_SENSOR\_0* – *PRIEMER\_SENSOR\_7*, teda pre každý senzor je napísaná zvláštna funkcia. Viac o kapacitných senzoroch sa dočítate v Semestrálnej práci 1, a o nami použitom algoritme na zistenie, či je klávesa stlačená alebo nie, sa dozviete v kapitole 1.6.

Teraz si pripravíme do premenných *PORTA\_U4*, *PORTC\_U4* hodnoty tak, že podľa toho, aké hodnoty EN1 a EN2 nám prišli od stolíka dentálnej súpravy, podľa toho vynulujeme príslušné bity v daných premenných. Premenné boli nainicializované na hodnotu FFH, čo znamená, že nesvieti žiadna LED dióda, a teda my vynulujeme tie bity, kde príslušné LED diódy majú svietiť. Keďže nastavujeme LED diódy pre mikrokontrolér U4, tieto môžeme hneď nastaviť a to zavolaním funkcií *A\_ONE\_LED\_ON* a *C\_ONE\_LED\_ON*, ktoré zasvietia vybrané LED diódy tým, že nastavia príslušné piny ako výstupné nastavením registrov TRISA a TRISC. Hodnoty registrov TRISA a TRISC si uložíme do premenných *TRIS\_PORTA\_Save* a *TRIS\_PORTC\_Save*. Následne odošleme tento istý povel aj mikrokontroléru U3, a to pomocou funkcie *PresetA1*. Odpoveď od mikrokontroléru U3 prijmeme už známou funkciou *RECEIVE\_96St* a skontrolujeme skúšobnú sumu CRC tejto odpovede. Keďže funkcia *RECEIVE\_96St* nám zmenila číslo mikrokontroléra na 2, musíme toto číslo zmeniť naspäť na 1 a odošleme ACKNOWLEDGE pomocou funkcie *ROC* s adresou 1.

### **Funkcie *PRIEMER\_SENSOR\_0* – *PRIEMER\_SENSOR\_7***

V týchto funkciách najskôr nastavíme hodnotu registru CPSCON1 podľa čísla senzoru 0 – 7. CPSCON1 je riadiaci register kapacitného snímania, a hodnota v ňom určuje, na ktorý senzor sa prepne multiplexor. Následne zavoláme funkciu *MEASURE\_PRIEMER*. Táto nám výslednú hodnotu uloží do dvojice premenných *ACCbHI\_SAVE* a *ACCbLO\_SAVE*, tieto hodnoty si uložíme do premenných *PriemerX\_H* a *PriemerX\_L*, kde X je číslo kapacitného senzora od 0 po 7.

### **Funkcia *MEASURE\_PRIEMER***

Keďže priemer budeme počítat' zo 4 nameraných hodnôt, uložíme si hodnotu 4 do premennej *VARIABLE\_16*. Potom si vynulujeme premenné do ktorých budeme ukladať súčet, a to je dvojica premenných *ACCbHI* a *ACCbLO* a znulujeme príznak pretečenia časovačov Timer1 a Timer2, ktoré budeme používať pri stanovení hodnoty senzoru. Príznaky pretečenia vynulujeme zapísaním hodnoty 0 do bitu *TMR1IF* a *TMR2IF* registru *PIR1*. Inicializáciu ukončíme vynulovaním hodnôt časovačov Timer1 a Timer2 zapísaním hodnoty 0 do registrov *TMR1L*, *TMR1H* a *TMR2*. Oba časovače boli pritom nakonfigurované a spustené v hlavnom programe, o ktorom sa viac dozvieme v kapitole 1.8, funkcia *READ\_CAP\_SENSORS*.

Následne čakáme, kým je časovač Timer2 pretečený tak, že testujeme v slučke bit *TMR2IF*. V tomto okamihu si uložíme hodnotu časovača Timer1 do premenných *ACCaHI*, *ACCaHI\_SAVE*, *ACCaLO* a *ACCaLO\_SAVE*. Vynulujeme príznak pretečenia oboch časovačov a hodnotu časovača Timer1 nastavíme na nulu. Teraz zavoláme funkciu *add16*, čo je 16 bitový súčet dvoch 16 bitových čísel, kde vstupné operandy sú uložené v premenných *ACCaLO*, *ACCaHI*, *ACCbLO* a *ACCbHI*, a

výsledok operácie sčítania je uložený do dvojice premenných *ACCbLO* a *ACCbHI*. Ak ešte nemáme namerané všetky 4 hodnoty, zostávame v slučke a znovu čakáme na pretečenie časovača *Timer2*.

Ak už máme zmerané 4 hodnoty a ich súčet uložený v premenných *ACCbLO* a *ACCbHI*, nasleduje delenie hodnotou 4 a to rotáciou premenných o 2 bity vpravo. Následne uložíme vypočítaný priemer do premenných *ACCaHI*, *ACCaHI\_SAVE*, *ACCaLO* a *ACCaLO\_SAVE*.

### **Funkcia PresetA1**

Touto funkciou odošleme z mikrokontroléra U4 do mikrokontroléra U3 povel na nastavenie príslušných LED diód, ide teda o povel „zápis konfigurácie“ z dokumentu v prílohe A. Pôjde vlastne o ten istý povel, ktorý dostal mikrokontrolér U4 od stolíka dentálnej súpravy, s tým rozdielom, že adresu nastavíme na 2 a vypočítame novú skúšobnú sumu CRC. Na odosielanie dát používame funkcie *CRC\_TRNS\_A1*, *CRC\_TRN\_A1* a už známu funkciu *TrnsByt\_9\_A1*. Funkcie *CRC\_TRNS\_A1*, *CRC\_TRN\_A1* využívajú k prenosu dát tiež funkciu *TrnsByt\_9\_A1*. Jediný rozdiel medzi nimi je, že u prvej menovanej sa počíta nová CRC suma funkciou *GenCRC*, zatiaľčo u druhej sa počíta CRC suma už z predtým vypočítanej sumy a to funkciou *GenCRC\_*.

### **Funkcia ROC**

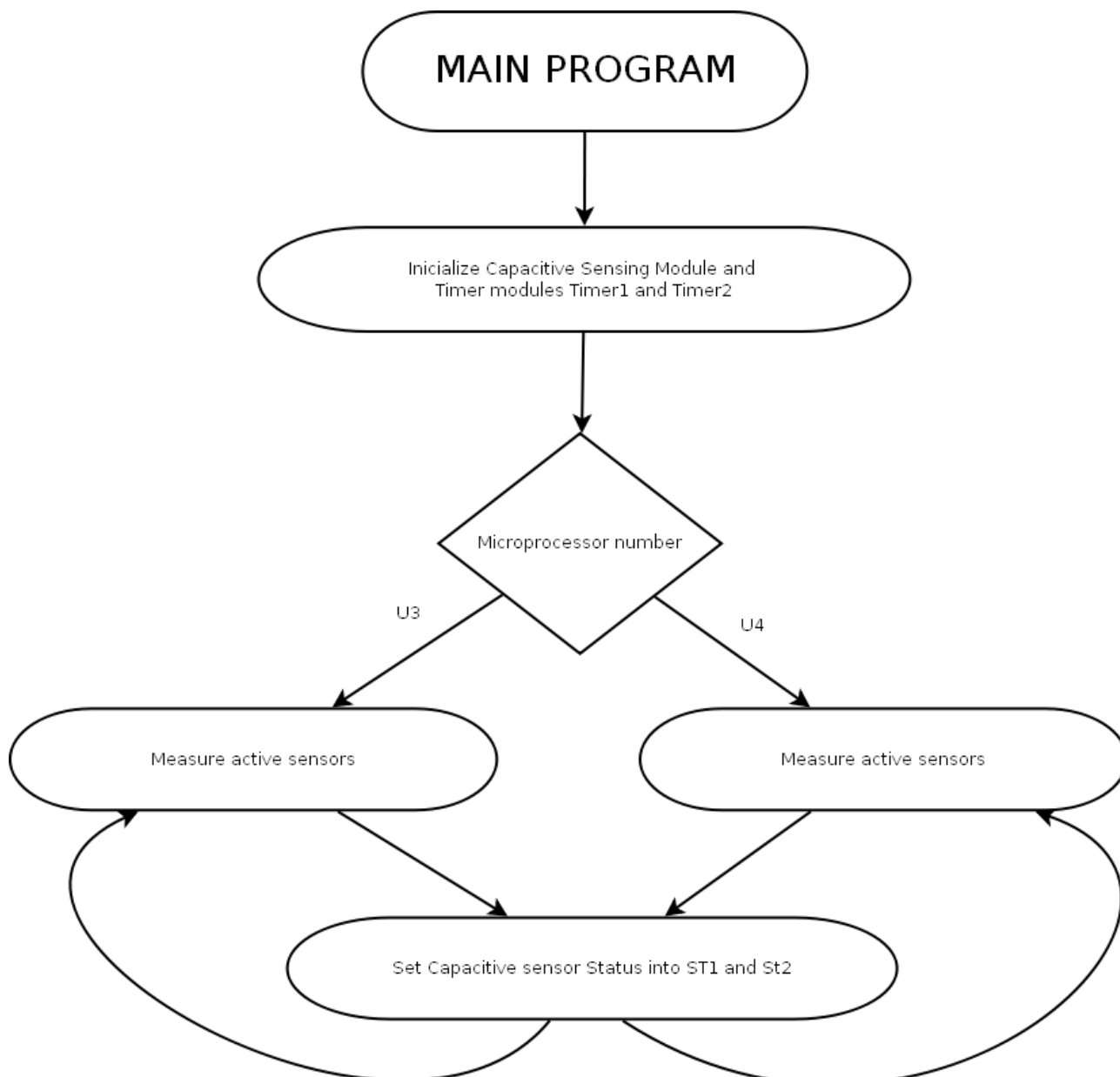
Funkcia odošle ACKNOWLEDGE podľa prílohy A a to buď z mikrokontroléra U3 do mikrokontroléra U4, alebo z mikrokontroléra U4 do stolíka dentálnej súpravy. Na prenos dát používa funkcie *CRC\_TRNS*, *CRC\_TRN\_* a *TransB*.

### **Funkcia SetLedU3**

Funkcia nastavuje svietenie LED diód, ktoré ovláda mikrokontrolér U3.

Do premenných *PORTA\_U3*, *PORTC\_U3* si pripravíme hodnoty tak, že podľa toho, aké hodnoty EN1 a EN2 nám prišli od stolíka dentálnej súpravy, podľa toho vynulujeme príslušné bity v daných premenných. Premenné boli nainicializované na hodnotu FFH, čo znamená, že nesvieti žiadna LED dióda, a teda my vynulujeme tie bity, kde príslušné LED diódy majú svietiť. Keďže nastavujeme LED diódy pre mikrokontrolér U3, tieto môžeme hneď nastaviť a to zavolaním funkcií *A\_ONE\_LED\_ON* a *C\_ONE\_LED\_ON*, ktoré zasvietia vybrané LED diódy tým, že nastavia príslušné piny ako výstupné nastavením registrov TRISA a TRISC. Hodnoty registrov TRISA a TRISC si uložíme do premenných *TRIS\_PORTA\_Save* a *TRIS\_PORTC\_Save*. Nakoniec odošleme ACKNOWLEDGE pomocou funkcie *ROC* s adresou 2.

## 5.8 Hlavný program



Obrázok 5.2: Vývojový diagram hlavného programu

Po počiatkových nastaveniach uvedených v kapitole 1.4 budeme v hlavnom programe postupne prechádzať všetky kapacitné senzory a ukladať si ich stav.

Na začiatku nastavíme číslo mikrokontroléra na 1 (do premennej *Frame1*), teda tvárime sa, že sa jedná o mikrokontrolér U4. Mikrokontrolér U3 zistí svoju adresu až vtedy, keď mu prídu dáta z mikroprocesoru U4 s adresou 2. Počkáme 1 sekundu kvôli stabilizácii napätia a zavoláme funkciu *READ\_CAP\_SENSORS*, ktorá nastaví modul kapacitného snímania a aj inicializuje časovače Timer1 a Timer2, ktoré budeme využívať k snímaniu. Následne nastavíme všetky LED diódy ako



vypnuté už známou funkciou *SetLedU4*. Táto funkcia nastaví LED diódy aj ovládané mikrokontrolérom U3, keďže zatiaľ si mikrokontrolér U3 tiež myslí že je U4. Súčasťou funkcie *SetLedU4* je aj funkcia *PresetA1*, ktorá odošle dáta do mikrokontroléra U3, takže mikrokontrolér U3 od tohto okamihu zistí svoju adresu. Teraz podľa adresy mikrokontroléra (v premennej *Frame1*) určíme, do ktorej slučky má program skočiť. Ak má mikrokontrolér adresu 1 (jedná sa o mikrokontrolér U4), skočíme na návěstie *U4\_LOOP*, ak má adresu 2 (jedná sa teda o mikrokontrolér U3), skočíme na návěstie *U3\_LOOP*.

### 5.8.1 Funkcia **READ\_CAP\_SENSORS**

V tejto funkcii nainicializujeme modul kapacitného snímania a časovače Timer1 a Timer2, ktoré budeme ku snímaniu používať. V tejto kapitole sme si pomáhali aj aplikačnou poznámkou Microchip [16], kde je detailne popísaná inicializácia modulu pre kapacitné snímanie.

Zapísaním hodnoty 30H do registrov ANSELA a ANSELB nastavíme všetky piny v portoch A a B u ktorých je to možné ako digitálne. Činnosť modulu kapacitného snímania povolíme zapísaním hodnoty 1 do bitu CPSON registra CPSCON0. V tom istom registri nastavíme vysoký rozsah oscilátora nastavením bitov CPSRNG0 a CPSRNG1 na jednotky. Zapísaním nuly do registru CPSCON1 vyberieme kanál 0 pre kapacitné snímanie. Časovač Timer1 inicializujeme zapísaním do registru T1CON. Zdroj hodín nastavíme ako oscilátor kapacitného snímania nastavením bitov TMR1CS0 a TMR1CS1. Časovač spustíme nastavením bitu TMR1ON a predeličku nastavíme na hodnotu 1:2 zapísaním hodnoty 01 do bitov T1CKPS<1:0>. Časovač Timer2, ktorý bude slúžiť ako časová základňa inicializujeme zápisom do registra T2CON. Časovač spustíme nastavením bitu TMR2ON, podeličku nastavíme zápisom do bitov TOUTPS<3:0> na 1:8 a predeličku nastavíme zápisom do T2CKPS<1:0> na 1:16. Limit časovača Timer2 nastavíme na najvyššiu možnú hodnotu FFH zápisom tejto hodnoty do registra PR2. A napokon znulujeme príznak pretečenia (resp. príznak dosiahnutia hodnoty registru PR2) časovača Timer2.

### 5.8.2 Cyklus **U4\_LOOP**

Tento cyklus prebieha dokola v mikrokontroléri U4 a jeho úlohou je prechádzať všetky kapacitné senzory a zisťovať a ukladať si ich stav.

Podľa prijatých bytov EN1 a EN2 od stolíka dentálnej súpravy zisťujeme, či je senzor aktívny (t.j. či svieti jeho LED dióda) a len aktívne senzory kontrolujeme. Ak senzor nie je aktívny, preskočíme rovno na ďalší. Z premenných *PriemerX\_H* a *PriemerX\_L*, kde X je číslo senzora od 0 po 7, ktorých hodnotu sme vypočítali vo funkciách *PRIEMER\_SENSOR\_0* – *PRIEMER\_SENSOR\_7*, prečítame hodnoty a uložíme do premenných *ACCbHI\_SAVE* a *ACCbLO\_SAVE*. Číslo senzora uložíme do premennej *SENSOR\_No* a následne voláme funkciu *MEASURE\_SENSOR*. Po prejdení všetkých kapacitných senzorov sa vraciame naspäť na návěstie *U4\_LOOP*.

### 5.8.3 Cyklus U3\_LOOP

Ten istý cyklus ako U4\_LOOP len s tým rozdielom, že prebieha v mikrokontroléri U3 a prechádzajú sa kapacitné senzory mikrokontroléru U3.

#### Funkcia MEASURE\_SENSOR

Zapísaním čísla senzoru do registru *CPSCON1* nastavíme kanál pre kapacitné snímanie na daný senzor. Potom voláme funkciu *MEASURE\_ONE\_POINT* na odmeranie hodnoty kapacitného senzoru a funkciu *CALCULATE\_DIFERENCE* na určenie výchylky hodnoty oproti priemernej hodnoty. Ak je výchylka vyššia ako stanovená hodnota, senzor sa javí ako stlačený a toto sa indikuje nastavením hodnoty premennej *Messag* na hodnotu rôznu od 0. Podľa hodnoty tejto premennej potom voláme buď funkciu *SENS\_ACTIVE* alebo *SENS\_NO\_ACTIVE*. Ak je senzor stlačený, zostávame v slučke a testujeme klávesu znovu. Celé meranie opakujeme 2x pre zvýšenie spoľahlivosti.

#### Funkcia MEASURE\_ONE\_POINT

Táto funkcia má za úlohu odmerať hodnotu daného kapacitného senzoru, teda v podstate hodnotu z časovača Timer1 za ktorú pretečie časovač Timer2, teda časová základňa.

Najskôr znulujeme príznak pretečenia časovačov Timer1 a Timer2, ktoré budeme používať pri stanovení hodnoty senzoru. Príznaky pretečenia vynulujeme zapísaním hodnoty 0 do bitu *TMR1IF* a *TMR2IF* registru *PIR1*. Inicializáciu ukončíme vynulovaním hodnôt časovačov Timer1 a Timer2 zapísaním hodnoty 0 do registrov *TMR1L*, *TMR1H* a *TMR2*.

Následne čakáme, kým je časovač Timer2 pretečený tak, že testujeme v slučke bit *TMR2IF*. V tomto okamihu si uložíme hodnotu časovača Timer1 do premenných *ACCaHI*, *ACCaHI\_SAVE*, *ACCaLO* a *ACCaLO\_SAVE*. Vynulujeme príznak pretečenia oboch časovačov a hodnotu časovača Timer1 nastavíme na nulu. Kvôli vyššej spoľahlivosti opakujeme celú procedúru 2x.

#### Funkcia CALCULATE\_DIFFERENCE

Funkcia určí výchylku hodnoty kapacitného senzora oproti priemernej hodnote. Ak je výchylka väčšia ako pevne daná hranica, senzor sa považuje za stlačený.

Z funkcie ktorá bolala volaná pred touto – *MEASURE\_ONE\_POINT* máme uchovanú aktuálnu hodnotu senzoru v premenných *ACCaHI* a *ACCaLO*. Priemerná hodnota pre senzor sa počítala vo funkcii *MEASURE\_PRIEMER* a je uložená v premenných *ACCbHI\_SAVE* a *ACCbLO\_SAVE*. Túto hodnotu skopírujeme do premenných *ACCbHI* a *ACCbLO*, pretože s nimi pracuje funkcia *sub16*, ktorá vykonáva 16 bitové odčítanie 2 čísel tak, že zobere 16 bitovú hodnotu z premenných *ACCbHI* a *ACCbLO* a odčíta od nej 16 bitovú hodnotu z premenných *ACCaHI* a *ACCaLO* a výsledok uloží opäť do *ACCbHI* a *ACCbLO*. Takto zistíme výchylku oproti priemernej hodnote. Ak je výchylka záporné číslo, spravíme komplement oboch *ACCbHI* a *ACCbLO*. Pokusnou metódou sme stanovili hranicu výchylky na FFH, a teda na stanovenie, či je klávesa stlačená alebo nie nám stačí vyhodnotiť, či je premenná *ACCbHI* rovná nule, alebo nie. Ak je rovná nule, klávesa nebola stlačená, ak je hodnota väčšia, klávesa bola stlačená.

### **Funkcia SENS\_ACTIVE**

Ak je niektorá klávesa stlačená, najskôr zistíme či sa jedná o procesor U3 alebo U4 podľa čísla procesoru v premennej *Frame1*. Podľa toho voláme buď funkcie *GetST1U3* a *GetST2U3* alebo *GetST1U4* a *GetST2U4*. Každá z týchto funkcií je vlastne dekóder, ktorý na základe čísla senzoru vráti hodnotu premennej *ST1*, resp. *ST2* tak, aby bolo indikované stlačenie správnej klávesy podľa dokumentu v prílohe A. Implementovali sme to pomocou pričítania čísla senzoru k programovému čítaču. Následne uložíme hodnoty do premenných *ST1* a *ST2*.

### **Funkcia SENS\_NO\_ACTIVE**

Žiadna klávesa nie je stlačená, preto nastavíme premenné *ST1* a *ST2* na nulu.

# 6 Riešenie programu na riadenie bezuhlíkového motora

Informácie v tejto kapitole sú čerpané z prílohy E.

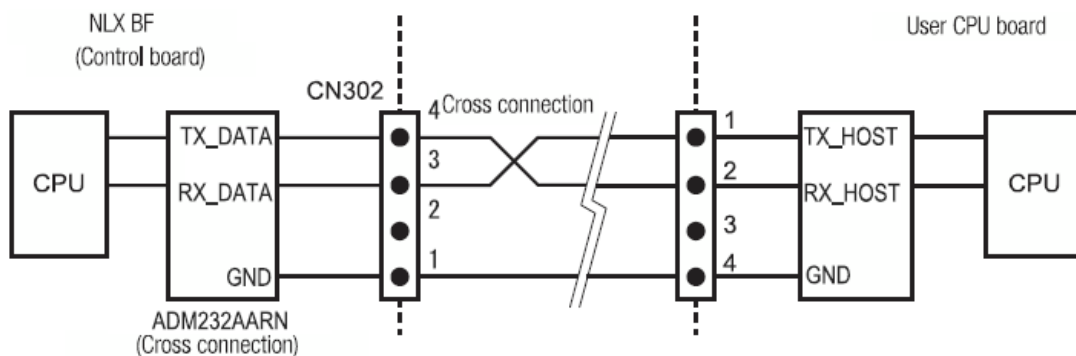
Motor sa ovláda pomocou dosky NLX BF tak, že z PC sa odosielajú dáta po sériovej linke RS232C.

Komunikačný protokol:

- Baud Rate – 19200 bps
- Počet dátových bitov – 8
- Parita – žiadna
- Počet stop bitov – 1
- Metóda prenosu – ASCII kódy
- Synchronizácia – asynchrónna komunikácia štart-stop
- Komunikačná metóda – polovičný duplex, 3 vodičový, polling

## 6.1 Zapojenie

Ako vidíme na obrázku 6.1, zapojenie je pomocou 3 vodičov, kde jeden je použitý pre vysielané dáta, jeden pre prijímané dáta a tretí je zem.



Obrázok 6.1: Zapojenie medzi PC a obvodom NLX BF [ príloha E]

## 6.2 Štruktúra prenášaných dát

Dáta, ktoré odosielať z PC do riadiaceho obvodu NLX BF majú štruktúru podľa obrázka 6.2. Dáta sa odosielajú ako ASCII znaky.

- STX – štart bit (1 byte)

Indikuje začiatok komunikačného rámca.

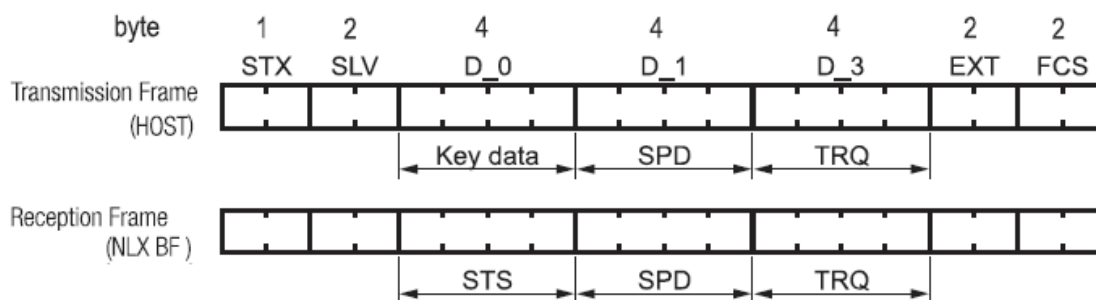
Hodnota: '@' - 0x40 v ASCII

- SLV – adresa zariadenia (2 byty)

Indikuje adresu zariadenia. Adresa modulu NLX BF je predvolená ako 05.

Hodnota: “00” – 0x3030 v ASCII

“05” – 0x3035 v ASCII



Obrázok 6.2: Štruktúra prenášaných dát pomocou RS-232C [ príloha E]

- Key data, STS – výber funkcií motora (4 byty)

Jednotlivé funkcie motora sú popísané na obrázku 6.3. Každá funkcia môže prebiehať v móde úrovňovom (level – L) alebo okamžitom (momentary – M). Úrovňový mód znamená to, že nezáleží v akom stave sa motor nachádza, ovládame ho zapísaním 1, kde sa vykoná požadovaná operácia, napr. pri povelu číslo 1 – ON, sa pri zápise 1 na daný bit zapne motor, a pri zápise 0 na ten istý bit sa motor vypne. Okamžitý mód znamená to, že záleží v akom stave sa motor nachádza, a na daný bit sa zapisuje vždy 1 a podľa toho v akom stave sa motor nachádza sa zmení jeho funkcia, napr. pri povelu číslo 1 – ON, sa pri zápise 1 na daný bit v prípade, že motor je zapnutý, tak sa motor vypne, a v prípade, že motor je vypnutý sa týmto povelom motor zapne.

Položka „BIT assignment“ v tabuľke znamená to, ktorý bit v riadiacich bytoch reprezentuje danú funkciu, napr. povel číslo 1 – ON má priradený bit 0x0001, čo znamená, že najmenej významný bit celého riadiaceho povelu bude ovládať zapnutie/vypnutie motora. Z toho vyplýva, že jedným povelom je možné ovládať viac funkcií motora naraz, resp. všetky funkcie motora sa dajú ovládať naraz jedným riadiacim povelom.

Z obrázku 6.3 vyplývajú pre nás tieto relevantné operácie:

- Motor ON / Motor OFF
- Reverse / Forward
- High Speed Mode (1000 – 40000 ot/min) / Low Speed Mode (100 – 1000 ot/min)
- Autoreverse ON/OFF
- Autoreverse + Forward ON/OFF
- LED ON/OFF

- SPD – otáčky motora (4 byty)

Určuje hodnotu otáčok motora v jednotke otáčky/min. Otáčky sa určujú zapísaním ASCII hodnoty hexadecimálnej reprezentácie decimálnej hodnoty otáčok/min, teda napr. 40000 ot/min = 0x9C40.

No.	Symbol	BIT assignment	Direction	Level	Operation
1	ON	0x0001	R/T	M	1: Motor on/off-signal by continuous communication
				L	1: Motor on 0: Motor off
2	DIR	0x0002	R/T	M	1: Forward/Reverse turn signal by continuous communication
				L	1: Reverse turn 0: Forward turn
3	L_DOWN	0x0004	R/T	M	1: Decrease LED voltage step by step (LED: bright → dark)
				L	—
4	L_UP	0x0008	R/T	M	1: Increase LED voltage step by step (LED: dark → bright)
				L	—
5	MODE	0x0010	R/T	M	1: High/Low speed mode signal by continuous communication
				L	1: High speed mode 0: Low speed mode
6	ATR	0x0020	R/T	M	1: Autoreverse valid/invalid signal by continuous communication
				L	1: Autoreverse is valid. 0: Autoreverse is invalid.
7	ATRF	0x0040	R/T	M	1: "Autoreverse + Forward" valid/invalid signal by continuous communication
				L	1: "Auto-reverse + Forward" is valid. 0: "Autoreverse + Forward" is invalid.
8	—	—	—	—	—
9	—	—	—	—	—
10	MB	0x0200	R/T	M	1:A or B side motor select signal by continuous communication
				L	1: B side motor is selected. 0: A side motor is selected.
11	LED	0x0400	R/T	L	LED on/off select signal by continuous communication 1: LED off 0: LED on
12	ERR	0x0800	T	L	1:Error occurs (*1) 0:No error occurs
	LEVEL		R	M	—
				L	1: Level operation 0: Momentary operation
13	—	—	—	—	—
14	—	—	—	—	—
15	LAMP_EN	0x4000	R	L	1:LED on(During a motor stop) ON+LED+LAMP EN=>"1"
16	SIGNAL	0x8000	R/T	M	—
				L	1: Single communication (*2) 0: Continuous communication (*3)

Obrázok 6.3: Popis funkcií motora [ príloha E]

- TRQ – krútiaci moment motora (4 byty)

Určuje hodnotu krútiaceho momentu motora prevedeného na elektrický prúd. Prúd môže mať hodnotu od 0,5 A do 3,5 A, čo reprezentujú hodnoty od 5 do 35. Táto hodnota je prevedená do hexadecimálnej reprezentácie a jednotlivé číslice sú konvertované do ASCII kódu, napr. 35 (3,5 A) = 0x0023.

Pre výpočet hodnoty elektrického prúdu z hodnoty krútiaceho momentu sa využíva rovnica:

$$Y = 11,1 X + 1,7, \quad (1)$$

kde X = krútiaci moment [Ncm]

Y = elektrický prúd [A]

Maximálny krútiaci moment: 35 (3,5 A) = 11,1 \* 3,0 [Ncm] + 1,7

Minimálny krútiaci moment: 5 (0,5 A) = 11,1 \* 0,3 [Ncm] + 1,7

- EXT – stop bit (1 byte)

Indikuje koniec komunikačného reťazca.

Hodnota: "\*" - 0x2A v ASCII

- FCS – skúšobná suma reťazca (2 byty)

Vypočítaný XOR z reťazca od prvého bytu '@' až po posledný '\*' postupne po bytoch. Na prijímacej strane, príjme reťazec a overí, či priložená FCS je správna. Ak sa zistí, že FCS nie je správna, dáta sa nepoužijú a budú považované za chybné.

- Detekcia chyby

Ak sa objaví chyba, 1 je nastavená do ERR bitu rámca STS a kód chyby je zobrazený v rámci SPD.

Chybové kódy sú zobrazené na obrázku 6.4.

Napr. 0x0200 = E9 (kód chyby: chyba spustenia motora)

Nakoľko výrobca neuvádza význam chybových kódov, zvolili sme nasledovný postup. V prípade, že sa objaví chybový kód, zopakujeme povel. V prípade, že sa chybový kód zopakuje, chybové hlásenie napíšeme na displej a ďalej situáciu rieši servisný pracovník.

SPD Data	ERR Code
0x0001	E0
0x0002	E1
0x0004	E2
0x0008	E3
0x0010	E4
0x0020	E5
0x0040	E6
0x0080	E7
0x0100	E8
0x0200	E9
0x0400	EA
0x0800	EB
0x1000	EC
0x2000	ED
0x4000	EE
0x8000	EF

Obrázok 6.4: Chybové kódy [ príloha E]

## 6.3 Riešenie programu

V tejto časti sa pozrieme detailne do jednotlivých funkcií programu. Program je riešený ako knižnica funkcií pre ovládanie bezuhlíkového motora. Pri riešení programu sme využívali informácie z datasheetu pre procesor, pod ktorým beží program, ktorý bude využívať naše funkcie, a to PIC18F252 [19]. Zoznam funkcií je vytvorený z relevantných operácií z kapitoly 1.2. Pre riadenie bezuhlíkového motora sme vytvorili tieto funkcie:

- Motor\_STOP – vyšle povel, ktorý zastaví motor
- Motor\_START\_FW – zastaví motor a spustí ho v režime forward
- Motor\_START\_FW\_LEDON – to isté ako funkcia č.2, s pridaním svietenia LED
- Motor\_START\_REVERSE – zastaví motor a spustí ho v režime reverse
- Motor\_START\_REVERSE\_LEDON – to isté ako funkcia č.4, s pridaním svietenia LED
- Motor\_START\_AutoReverse – zastaví motor a spustí ho v režime forward s funkciou autoreverse
- Motor\_START\_AutoReverse\_LEDON – to isté ako funkcia č.6, s pridaním svietenia LED
- Motor\_START\_AutoReverseForward – zastaví motor a spustí ho v režime forward s funkciou autoreverse+forward
- Motor\_START\_AutoReverseForward\_LEDON – to isté ako funkcia č.8, s pridaním svietenia LED



### 6.3.1 Funkcia *Motor\_STOP*

Táto funkcia odošle z riadiaceho mikroprocesora povel na zastavenie motora a takisto skontroluje odpoveď modulu NLX BF.

Podľa popisu funkcií motora na obrázku 6.3 vytvoríme povel na zastavenie motora. Tento povel uložíme do premenných *MotorB\_COMMAND\_1-4*. Využijeme funkciu „Motor off“ nastavením prvého bitu *MotorB\_COMMAND\_1* na ASCII hodnotu '0'. Pri všetkých poveloch používame úrovňový (level) mód, teda nastavíme *MotorB\_COMMAND\_3* na hodnotu '8'. Ostatné bity povelu budú hodnota '0'. Rýchlosť motora, ktorá má veľkosť 2 byty, si pripravíme do premenných *Motor\_Speed\_L* a *Motor\_Speed\_H*. Následne zavoláme funkciu *MotorB\_COMMAND\_Exec*, ktorá odošle riadiaci povel a prijme odpoveď modulu NLX BF.

### 6.3.2 Funkcia *Motor\_START\_FW*

Táto funkcia odošle povel na spustenie motora v režime forward s danou rýchlosťou otáčok motora a takisto skontroluje odpoveď modulu NLX BF a prípadne nastaví chybové hlásenie.

Postupujeme podľa informácií z odpovedi na otázky firmy NLX [príloha D]. Takže pre spustenie motora je potrebné odoslať tieto 3 povel:

1. Motor STOP – funkcia *Motor\_STOP*
2. Motor ON – povel na spustenie motora ale s hodnotou otáčok motora 0
3. Motor START – povel na spustenie motora s nastavenou hodnotou otáčok motora

Najskôr zavoláme funkciu *Motor\_STOP*, ktorá odošle povel na zastavenie motora. Následne zavoláme funkciu *Test\_SPEED*, ktorou otestujeme, či hodnota otáčok motora je menšia ako 1000 ot/min, potom spustíme motor v móde Low Speed, inak ho spustíme v móde High Speed. Do premennej *MotorB\_COMMAND\_2* uložíme ASCII hodnotu '0' ak chceme nastaviť Low Speed mód, a hodnotu '1' pre High Speed mód. Do premennej *MotorB\_COMMAND\_1* nastavíme funkciu Motor on – hodnota '1' a do *MotorB\_COMMAND\_3* nastavíme hodnotu '8' – úrovňový mód.

Následne pokračujeme návštvou *Speed\_CONT*, tu nastavíme do premennej *MotorB\_COMMAND\_4* hodnotu '0', uložíme si hodnoty otáčok motora, ktoré máme uložené v premenných *Motor\_SPEED\_L* a *Motor\_SPEED\_H* a nastavíme do nich hodnoty 0. Následne zavoláme funkciu *MotorB\_COMMAND\_Exec*, ktorá odošle riadiaci povel Motor ON a prijme odpoveď modulu NLX BF. Potom obnovíme hodnoty otáčok motora z pomocných premenných do *Motor\_SPEED\_L* a *Motor\_SPEED\_H* a opäť zavoláme *MotorB\_COMMAND\_Exec*, ktorá odošle riadiaci povel Motor START.

### 6.3.3 Funkcia *Motor\_START\_FW\_LEDON*

Táto funkcia sa líši od funkcie *Motor\_START\_FW* len v tom, že sa pri spustení motora zapne aj LED dióda. Túto funkciu nastavíme tým, že do premennej *MotorB\_COMMAND\_3* nastavíme hodnotu 'C', ktorá vznikne sčítaním hodnôt '8' (úrovňový mód) a '4' (LED ON).

### **6.3.4 Funkcia Motor\_START\_REVERSE**

Táto funkcia sa líši od funkcie *Motor\_START\_FW* len tým, že motor spúšťame nie v režime forward, ale v režime reverse (teda sa krúti opačným smerom). Túto funkciu nastavíme tým, že do premennej *MotorB\_COMMAND\_1* nastavíme hodnotu '3', ktorá vznikne sčítaním hodnôt '1' (Motor on) a '2' (Reverse turn).

### **6.3.5 Funkcia Motor\_START\_REVERSE\_LEDON**

Táto funkcia sa líši od funkcie *Motor\_START\_REVERSE* len v tom, že sa pri spustení motora zapne aj LED dióda. Túto funkciu nastavíme tým, že do premennej *MotorB\_COMMAND\_3* nastavíme hodnotu 'C', ktorá vznikne sčítaním hodnôt '8' (úrovňový mód) a '4' (LED ON).

### **6.3.6 Funkcia Motor\_START\_AutoReverse**

Táto funkcia sa líši od funkcie *Motor\_START\_FW* len tým, že u motora povolíme režim autoreverse (pri určitom tlaku lekára na nástroj sa zvýšené zaťaženie motora zdetekuje a motor sa automaticky začne točiť opačným smerom). Túto funkciu nastavíme tým, že do premennej *MotorB\_COMMAND\_2* nastavíme hodnotu '2' alebo '3' v závislosti na režime (High speed alebo Low speed), ktorá vznikne sčítaním hodnôt '0' alebo '1' (Low/High speed mód) a '2' (Autoreverse povolený).

### **6.3.7 Funkcia Motor\_START\_AutoReverse\_LEDON**

Táto funkcia sa líši od funkcie *Motor\_START\_AutoReverse* len v tom, že sa pri spustení motora zapne aj LED dióda. Túto funkciu nastavíme tým, že do premennej *MotorB\_COMMAND\_3* nastavíme hodnotu 'C', ktorá vznikne sčítaním hodnôt '8' (úrovňový mód) a '4' (LED ON).

### **6.3.8 Funkcia Motor\_START\_AutoReverseForward**

Táto funkcia sa líši od funkcie *Motor\_START\_FW* len tým, že u motora povolíme režim autoreverse+forward (pri určitom tlaku lekára na nástroj sa zvýšené zaťaženie motora zdetekuje a motor sa automaticky začne točiť opačným smerom, a následne pri tlaku opäť opačným smerom). Túto funkciu nastavíme tým, že do premennej *MotorB\_COMMAND\_2* nastavíme hodnotu '4' alebo '5' v závislosti na režime (High speed alebo Low speed), ktorá vznikne sčítaním hodnôt '0' alebo '1' (Low/High speed mód) a '4' (Autoreverse+forward povolený).

### **6.3.9 Funkcia Motor\_START\_AutoReverseForwardLEDON**

Táto funkcia sa líši od funkcie *Motor\_START\_AutoReverseForward* len v tom, že sa pri spustení motora zapne aj LED dióda. Túto funkciu nastavíme tým, že do premennej *MotorB\_COMMAND\_3* nastavíme hodnotu 'C', ktorá vznikne sčítaním hodnôt '8' (úrovňový mód) a '4' (LED ON).

## Funkcia Test\_SPEED

Funkcia, ktorá otestuje hodnotu otáčok motora a podľa nej určí, či spustíme motor v režime Low speed alebo High speed. Ak je hodnota vyššia ako 1000 ot/min, režim je High speed, ak je menšia, režim je Low speed..

Najskôr prečítame hodnotu otáčok zavolaním funkcie *READ\_Speed*. Následne testujeme vrchný byte hodnoty otáčok. Ak je táto hodnota v premennej *Motor\_Speed\_H* väčšia ako 3 (1000 ot/min je po prevedení na hexadecimálnu hodnotu 0x3E8), vieme, že otáčky sú v režime High speed a toto indikujeme nastavením premennej *Speed\_HS* na hodnotu 1. V prípade, že hodnota v *Motor\_SPEED\_H* je menšia ako 0x3, otáčky sú v režime Low speed a indikujeme to nastavením premennej *Speed\_HS* na 0. Ak je hodnota v danej premennej rovná 3, musíme testovať aj dolný byte *Motor\_SPEED\_L*. Ak je táto hodnota väčšia alebo rovná 0xE8, nastavíme *Speed\_HS* na hodnotu 1, v opačnom prípade na hodnotu 0.

### 6.3.10 Funkcia MotorB\_COMMAND\_Exec

Táto funkcia odošle kompletný povelový reťazec bytov s definovanou štruktúrou podľa kapitoly 6.2. z riadiaceho mikroprocesora do modulu NLX BF.

Štruktúra povelového reťazca *MotorB\_COMMAND\_Exec*

- štart byte STX - hodnota '@'
- adresa zariadenia - hodnota "05"
- funkcia motora – funkcia *MotorB\_COMMAND*
- otáčky motora – funkcia *READ\_Speed*
- krútiaci moment motora – funkcia *READ\_TorqueB\_PARAMETERS*
- stop byte EXT - hodnota '\*'
- skúšobná suma FCS

Na začiatku znulujeme premennú *FCS\_CheckSum*, do ktorej budeme počítat' skúšobnú sumu FCS. Ďalej odošleme štart bit STX (hodnota '@'). Odosielanie bytov zabezpečuje funkcia *Trns\_19200*. Hneď po prvom odoslanom byte vypočítame FCS, a budeme ju počítat' po každom odoslanom byte v premennej *FCS\_CheckSum*. Následne odošleme adresu zariadenia – SLV, ktorá má štandardne hodnotu "05" pre tento modul.

Ďalej odošleme riadiace dáta pomocou funkcie *MotorB\_COMMAND*.

Podľa [príloha D] zariadenie neakceptuje „multiple mode“ povely. Z týchto dôvodov sme zadefinovali špeciálne povely pre operácie 1-9. Tieto 4 byty máme pripravené v premenných *MotorB\_COMMAND\_1-4*.

Ďalej budeme posielat' hodnotu otáčok motora SPD. Otáčky motora v jednotkách otáčky/min sú uložené v dvojici premenných *Motor\_SPEED\_H* a *Motor\_SPEED\_L*. Hodnotu otáčok musíme skonvertovať do ASCII a tak odoslat'. Toto vykonávajú funkcie *CONVERT\_HN\_to\_ASCII* a *CONVERT\_LN\_to\_ASCII*, kde prvá funkcia prevádza horné 4 bity, a druhá funkcia spodné alebo menej významné 4 bity.

Po odoslaní rámca s hodnotou otáčok motora, nasleduje krútiaci moment motora – rámec TRQ. Keďže prvé 2 hexadecimálne číslice hodnoty krútiaceho momentu, vyjadreného v hodnotách prúdu

5 – 35, sú vždy nulové, odošleme dvakrát ASCII hodnotu znaku '0'. Hodnotu krútiaceho momentu prečítame z pamäte EEPROM pomocou funkcie *READ\_TorqueB\_PARAMETERS*, na základe čísla nástroja, ktorý je vybraný. Výsledok, ktorý máme uložený v premennej *RcvReg*, musíme previesť z hodnoty krútiaceho momentu v Ncm, ktorú zadáva lekár, na hodnotu elektrického prúdu v A. Tento prevod vykonáva funkcia *Torque\_to\_Current*. Výsledok uložíme do premennej *ACCbLO* a následne skonvertujeme už známymi funkciami *CONVERT\_HN\_to\_ASCII* a *CONVERT\_LN\_to\_ASCII*. Hodnoty započítame do výpočtu skúšobnej sumy FCS a odošleme. Následne odošleme stop bit – EXT, čo je ASCII hodnota znaku '\*' a skúšobnú sumu FCS, ktorú skonvertujeme do ASCII funkciami *CONVERT\_HN\_to\_ASCII* a *CONVERT\_LN\_to\_ASCII*.

Po odoslaní nasleduje prijatie odpovedi. Byty odpovedi prijíma funkcia *Rcv\_19200*. Touto funkciou postupne prijímame celú odpoveď so štruktúrou na obrázku 6.2. Bezchybné hlásenie je v podstate zhodné s povelovým reťazcom, ktorý sme odoslali zariadeniu NLX BF. Ak prenos jedného z bytov nebol úspešný (byte je poškodený, alebo nenastáva žiadna komunikácia), skočíme na začiatok funkcie a odošleme celý komunikačný reťazec znovu. Po každom prijatom byte počítame skúšobnú sumu FCS. Postupne prijímame štart bit, adresu zariadenia, STS, kde je uložené, či nastala chyba zariadenia. Ak chyba nenastala, pokračujeme prijímaním hodnôt otáčok a krútiaceho momentu, a napokon stop bitu a na konci skontrolujeme skúšobnú sumu FCS, či naša vypočítaná suma je rovnaká ako prijatá suma. Ak suma nesedí, opäť odošleme celý komunikačný reťazec znovu.

V prípade, že nastala chyba zariadenia, prijímame reťazec SPD, kde je uložený kód chyby podľa obrázku 6.4 a tento chybový kód uložíme do premennej *ERROR\_CODE*, ktorá má 4 byty, a dentálna súprava ho vypíše na displej ako servisné hlásenie.

### **Funkcia *Trns\_19200***

Táto funkcia odosiela dáta (nastavuje hodnotu na výstupnom pine), ktoré sú uložené vo W registry.

Najskôr odošleme štart bit s hodnotou 0 a túto hodnotu necháme na výstupe po dobu 52,1  $\mu$ s, čo znamená prenosovú rýchlosť 19200 Baud. Následne odošleme všetkých 8 dátových bitov a 1 stop bit, ktoré majú hodnotu 1.

### **Funkcia *READ\_Speed***

Táto funkcia prečíta hodnotu otáčok motora, ktorú chceme nastaviť, z pamäti EEPROM, a túto hodnotu uloží do dvojice premenných *Motor\_SPEED\_H* a *Motor\_SPEED\_L*.

Funkcia testuje, v akom režime sa nachádza bezuhlíkový motor. Motor môže byť v 3 režimoch – manual forward, manual reverse a stop on max, a pre každý režim je v pamäti EEPROM uložená iná hodnota otáčok. Aktuálny režim je uložený v premennej *MANUAL* a podľa nej prečítame z pamäti EEPROM hodnotu otáčok v danom režime.

### **Funkcia *CONVERT\_HN\_to\_ASCII* (*CONVERT\_LN\_to\_ASCII*)**

Tieto funkcie konvertujú 8 bitové číslo na ASCII znaky. Funkcia *CONVERT\_HN\_to\_ASCII* konvertuje vrchné 4 bity čísla, a funkcia *CONVERT\_LN\_to\_ASCII* konvertuje spodné 4 bity čísla.

Pri zavolaní funkcie *CONVERT\_HN\_to\_ASCII* najskôr vymeníme vrchné 4 bity so spodnými 4 bitmi a zavoláme funkciu *CONVERT\_LN\_to\_ASCII*.

Funkcia *CONVERT\_LN\_to\_ASCII* začína tým, že vymaskujeme spodné 4 bity hodnoty, ktorú konvertujeme. Potom odpočítame od tejto hodnoty hodnotu 10, a tým zistíme, či je číslo dvojciferné. Ak je výsledok menší ako 0, teda číslo je menšie ako 10, spravíme bitovú operáciu or s ASCII hodnotou znaku '0' a tým dostaneme ASCII hodnotu danej číslice. Naopak, ak je výsledok väčší ako 0, teda číslo je väčšie alebo rovné 10, pričítame k nemu hodnotu 7, a tým dostaneme výsledok väčší ako 15, vymaskujeme spodné 4 bity výsledku a pričítame k ASCII hodnote znaku '@', tým dostaneme ASCII hodnoty znakov 'A' až 'F'.

### **Funkcia Torgue\_to\_Current**

Funkcia prevádza hodnotu krútiaceho momentu v Ncm\*10 na hodnotu elektrického prúdu v A\*10 podľa vzorca (1), ktorý sme si prispôbili tak, aby sa v ňom nevyskytovali žiadne čísla s desatinnou čiarkou, náš vzorec vyzerá nasledovne:

$$100 * Y = 111 * X + 170 \quad (2)$$

kde Y = elektrický prúd v A\*10

X = krútiaci moment v Ncm\*10

Vstupnú hodnotu v premennej *RcvReg* vynásobíme číslom 111. K výsledku následne pripočítame hodnotu 170 funkciou na sčítanie dvoch 16 bitových čísel *add16b*. Teraz potrebujeme vydeliť výsledok číslom 100 podľa vzorca (2). Nato využijeme funkciu *DIV16*, ktorá vydolí 16 bitové číslo N bitovým deliteľom, v našom prípade bude N=8. Táto funkcia je prevzaná zo stránok matematických metód pre mikrokontroléry PIC [17]. Po vydelení skontrolujeme zostatok po delení, ak je zostatok väčší ako 50, pripočítame k výsledku 1 a tým výsledok zaokrúhlime na najbližšie celé číslo. Výsledok uložíme opäť do premennej *RcvReg*.

### **Funkcia add16b**

Funkcia sčíta dve 16 bitové čísla uložené v *ACCaHI:ACCaLO* a *ACCbHI:ACCbLO* a výsledok uloží do *ACCbHI:ACCbLO*.

Najskôr vykonáme klasické sčítanie spodných bytov *ACCaLO+ACCbLO*. Potom skontrolujeme, či bol nastavený Carry flag, teda výsledok pretiekol cez 8 bitov. V prípade, že je Carry nastavené, k vrchnému bytu výsledku *ACCbHI* pridáme 1. Následne sčítame vrchné byty *ACCaHI+ACCbHI*.

### **Funkcia Rcv\_19200**

Funkcia na prijatie jedného bytu od obvodu NLX BF.

Komunikačný pin je označený *ASYN\_9600\_In*. Na začiatku kontrolujeme prijatie štart bitu s hodnotou logická 0. Ak sme štart bit v určitom časovom okamžiku nedetekovali, nastavíme Zero flag v registry STATUS a tým indikujeme, že žiadna komunikácia nie je aktívna. Ak sme štart bit prijali, počkáme časový interval 1,5 bitu kvôli tomu, aby sme detekovali dátové bity v strede bitu. Následne prijímame 8 dátových bitov do premennej *RcvReg*. Napokon testujeme validný stop bit s hodnotou 1. Ak je stop bit validný, prenos je úspešný, a signalizujeme to nastavením Zero flagu na 0. Naopak, ak prenos nie je úspešný, Zero flag nastavíme na 1.

# 7 Záver

## 7.1 Klávesnica s proximity senzormi

V tejto práci, ktorá je prvou časťou z dvoch častí mojej diplomovej práce, som sa venoval konštrukcii, integrácii a elektrotechnickému riešeniu kapacitnej klávesnice lekárskej dentálnej súpravy VIZIO.

V prvej kapitole som uviedol schémy integrácie kapacitnej klávesnice do lekárskej dentálnej súpravy VIZIO, ktoré mi boli poskytnuté zadávateľom práce, firmou Inovatica, s.r.o. so sídlom v Malackách. Táto spoločnosť spolupracuje s firmou Chiradent, s.r.o. so sídlom v Piešťanoch na vývoji dentálnej súpravy VIZIO.

V druhej kapitole som sa venoval prieskumu trhu v oblasti dentálnych súprav, a to hlavne priamej konkurencii – súpravám vyrábaných na Slovensku, od spoločností Diplomat Dental, s.r.o. a Chiradent, s.r.o. so sídlom v Piešťanoch a spoločnosti Chirana Medical a.s. Stará Turá. Nakoľko všetky uvedené firmy sa stretávajú na svetových trhoch, (domáci trh je zanedbateľný) je aj vplyv tejto priamej konkurencie treba posudzovať v medzinárodnom meradle. Z týchto dôvodov som pre porovnanie uviedol aj súpravy jedného z popredných zahraničných výrobcov dentálnych súprav, spoločnosť Elettromeccanica Castellini so sídlom v talianskej Bologni. Porovnávanie s poprednými zahraničnými výrobcami je potrebné preto, že 95 percent produkcie hore uvedených firiem ide na svetový trh.

V tretej kapitole som skúmal požiadavky na klávesnicu súpravy VIZIO od zadávateľa a na základe týchto požiadavok som v štvrtej kapitole navrhoval jej elektrické riešenie, ktoré pozostávalo z výberu vhodného riešenia kapacitných senzorov a obvodom podsvietenia. Následne som navrhol schému elektrického riešenia a plošné spoje.

V prílohe uvádzam návrh dosky plošných spojov.

Ďalej som sa venoval softwarovému riešeniu kapacitnej klávesnice lekárskej dentálnej súpravy VIZIO. Popisované riešenie je riadne otestované a funkčné.

Pri programovaní a testovaní sme tiež zistili nedostatky v hardwarovom riešení. Tieto nedostatky súviseli s tým, že neplatilo, že jeden mikrokontrolér (t.j. U3 alebo U4) ovládal aj senzor aj LED diódu rovnakej klávesy (je to vidieť na schéme v prílohe B). Funkcie, ktorých sa toto týkalo sú: „plnenie pohára“ a „šípka vpravo“ (senzor na mikrokontroléri U3, LED na U4) a „hlavné svetlo“ (senzor na mikrokontroléri U4, LED na U3). V koncovom riešení sme tieto nedostatky odstránili.

V prílohe uvádzam dokument s detailným popisom komunikácie medzi mikroprocesormi U3 a U4 a medzi mikroprocesorom U4 a stolíkom súpravy, ďalej schému zapojenia mikroprocesorov s kapacitnými senzormi a LED diódami a návrh dosky plošných spojov.

Riešenie hardwaru bolo vykonané za predpokladu použitia programovania mikroprocesorov metódou In-System programming. Uvedeným princípom bola zrealizovaná rozhodujúca časť vývojových prác. Pri zavedení súpravy do výroby a jej exportovaní k cieľovým zákazníkom sa však ukázalo, že tento prístup má nedostatky logistického charakteru, ktoré spočívajú v rozdielnych pracovných metódach a rôznych požiadavkách lekárov v rôznych častiach sveta. Tieto ich rozdielne

požiadavky je potrebné riešiť v reálnom čase aktualizáciou ich softwaru. Keď prihliadneme na odborné a prístrojové vybavenie servisných pracovníkov, ktorí sú autorizovaní vykonávať uvedené aktualizácie, tak najjednoduchším a najspoľahlivejším riešením je výmena mikroprocesora v päťici. Z týchto dôvodov koncové riešenie upustilo od tohto typu programovania a SMD mikroprocesorov. Mikroprocesory v puzdre DIP sú osadené v päťiciach.

## 7.2 Bezuhlíkový motor

Popisované funkcie boli odladené ako komunikácia riadiacej elektroniky stolíka lekára s modulom NLX BF. V priebehu vývoja sa nám vyskytli mnohé nejasnosti o činnosti zariadenia, ktoré sme riešili otázkami na výrobcu, NSK Japonsko, vid' príloha D. Napriek tejto komunikácii sa domnievame, že všetky odpovede neboli úplne kompetentné, napr. multiple command mode. Z týchto dôvodov nepovažujeme všetky prezentované riešenia za optimálne, pretože podľa nás niektoré povely by sa dali napísať do jedného ako multiple command mode. Túto stratégiu sme nezvolili preto, lebo sme nemali údaje ako by tento prístup ovplyvnil spoľahlivosť funkcie dentálnej súpravy. Z dôvodov spoľahlivosti a istoty sme pre komunikáciu s NLX BF zvolili level mode, ktorý na rozdiel od momentary módu umožňuje zopakovať povel. Zubná súprava VIZIO je v súčasnej dobe vo výrobe a prakticky 100 % sa exportuje. Z uvedeného vyplýva, že vývojoví pracovníci sa sústreďujú na vytváranie modifikácií a nových funkcií vyplývajúcich z požiadaviek koncových zákazníkov. Po ukončení tejto etapy sa pristúpi k optimalizácii procedúr vyriešených v tejto práci.

# Literatura

- [1] DIPLOMAT DENTAL s.r.o.: *Stomatologické súpravy* [online].  
<http://www.diplomat-dental.com/produkty/stomatologicke-supravy/>, 2013 [cit. 2013-03-03].
- [2] CHIRANA Medical: *Stomatologické súpravy* [online].  
[http://www.chirana-eshop.com/webmed/index.php?option=com\\_content&view=article&id=89&Itemid=89&lang=sk](http://www.chirana-eshop.com/webmed/index.php?option=com_content&view=article&id=89&Itemid=89&lang=sk), 2013 [cit. 2013-03-03].
- [3] CHIRADENT Mediprogress Group: *Stomatologické súpravy* [online].  
<http://www.chiradent-sk.sk/index.php/sk/dentalne-zariadenie/2012-06-10-10-26-01/2012-06-10-10-23-23>, 2013 [cit. 2013-03-03].
- [4] Castellini: *Dental units - Castellini* [online].  
<http://www.castellini.com/en/products/dental-units>, 2013 [cit. 2013-03-20].
- [5] Microchip Technology: *PIC16F72X/PIC16LF72X, Data Sheet* [online].  
<http://www.alldatasheet.com/datasheet-pdf/pdf/348802/MICROCHIP/PIC16F72X.html>, 2009-03-26 [cit. 2013-03-23].
- [6] RIPKA, Pavel, ĎAĎO, Stanislav, KREIDL, Marcel, NOVÁK, Jiří. *Senzory a převodníky*. 1. vyd. Praha : ČVUT, 2005, s. 26-31. ISBN 80-01-03123-3.
- [7] STMicroelectronics: *S-Touch™ PCB and layout guidelines* [online].  
<http://notes-application.abcelectronique.com/005/5-10445.pdf>, 2008 [cit. 2013-04-08].
- [8] Elneec: *SmartProg2 – univerzálny programátor* [online].  
<http://www.elnec.com/sk/products/universal-programmers/smartprog2/>, 2012-10-31 [cit. 2013-04-23].
- [9] GM electronic: *LED 1206 BLUE 45/130°* [online].  
<http://www.gme.cz/led-smd/led-1206-blue-45-130-p960-036/#popis>, 2013 [cit. 2013-04-25].
- [10] GM electronic: *L-KP-2012QBC-D* [online].  
<http://www.gme.cz/led-smd/l-kp-2012qbc-d-p960-192/#dokumentace>, 2013 [cit. 2013-04-25].
- [11] Texas Instruments: *MC33063A, MC34063A 1.5-A Peak Boost/Buck/Inverting Switching Regulator (Rev. M) - mc340, Data Sheet* [online].  
<http://www.ti.com/lit/ds/symlink/mc34063a.pdf>, 2011-01-30 [cit. 2013-04-25].
- [12] NOMAD: *MC34063A design tool* [online].  
<http://www.nomad.ee/micros/mc34063a/index.shtml>, 2013 [cit. 2013-04-25].
- [13] FIEDLER, P., CACH P., ZEZULKA, F.: *Průmyslové komunikační sítě a sítě v automobilovém průmyslu* [online]. Fiedler S.I. s.r.o., Brno: 2001 [cit. 2013-05-06]. Dostupný z: [https://www.vutbr.cz/www\\_base/priloha.php?dpid=42352](https://www.vutbr.cz/www_base/priloha.php?dpid=42352).
- [14] RS232 Tutorial on Data Interface and cables [online]. <http://www.arcelect.com/rs232.htm>, 2013 [cit. 2013-05-10].
- [15] Microchip Technology: *How To Use The Capacitive Sensing Module (CSM), Application Note* [online].



<http://www.alldatasheet.com/datasheet-pdf/pdf/348802/MICROCHIP/PIC16F72X.html>, 2010-01-05 [cit. 2014-01-06].

[16] Microchip Technology: *Software Handling for Capacitive Sensing, Application Note* [online].

<http://ww1.microchip.com/downloads/en/AppNotes/01103a.pdf>, 2007-06-25 [cit. 2014-01-06].

[17] ZHER, L.: *Divide 16 bit int by 8 bit int to 8 bit int and remainder* [online].

<http://www.piclist.com/techref/microchip/math/div/16by8lz.htm>, 2010-02-24 [cit. 2014-05-06].

[18] ZEŽULKA, F. *Prostředky průmyslové automatizace*. Brno: VUTIUM, 2004. 176 s.

ISBN: 80-214-2610- 1.

[19] Microchip Technology: *PIC18FXX2, Data Sheet* [online].

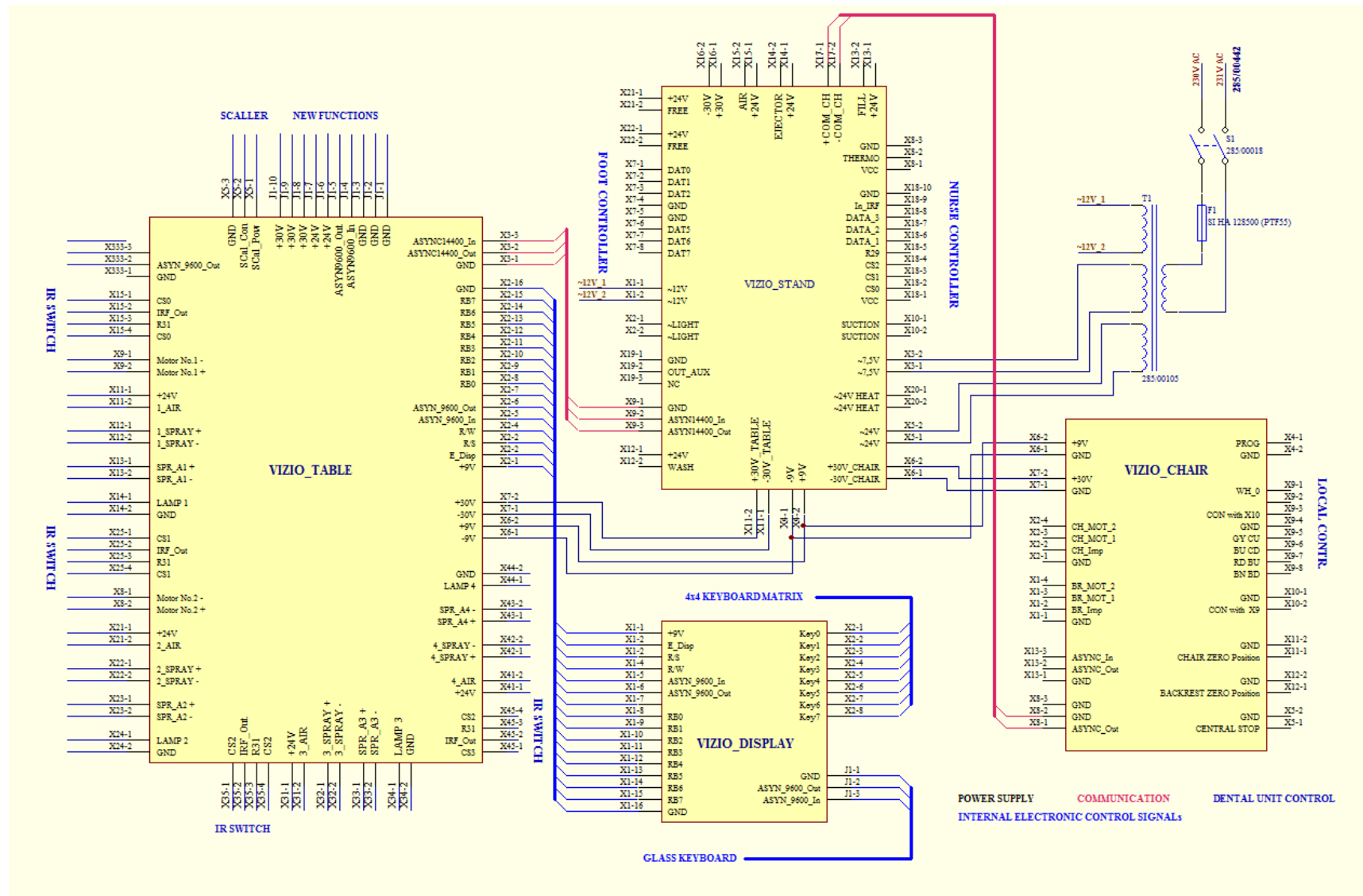
<http://ww1.microchip.com/downloads/en/DeviceDoc/39564c.pdf>, 2006-08-29 [cit. 2014-03-23].

# 8 Zoznam príloh

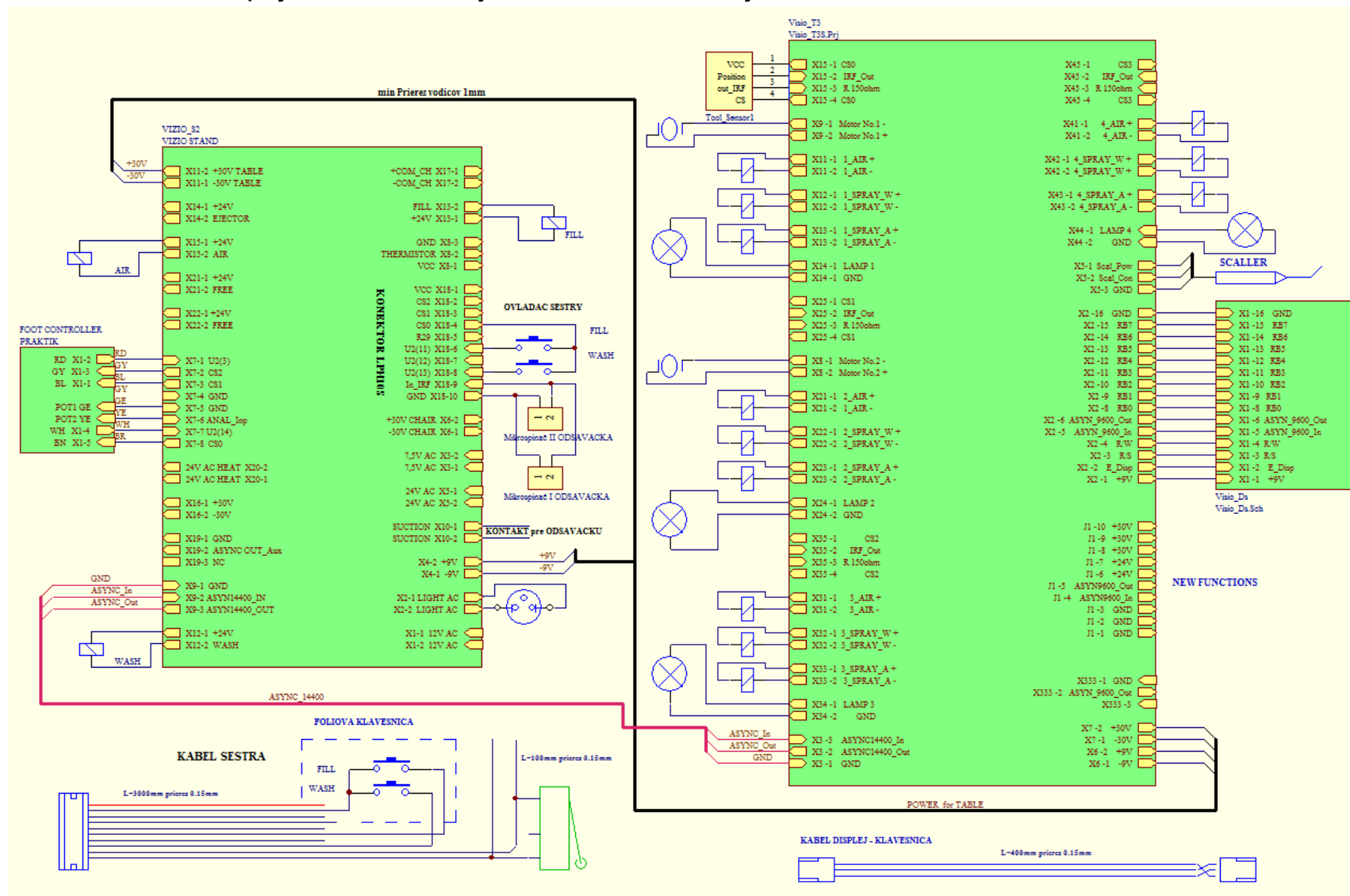
A	Popis komunikácie s klávesnicou - Communication with Keyboard that Use Capacitive Sensors.doc	
B	Elektrická schéma.....	67
B.1	Bloková schéma zapojenia elektroniky lekárskej dentálnej súpravy.....	68
B.2	Bloková schéma zapojenia elektroniky stolíka lekára a stojana.....	69
B.3	Bloková schéma zapojenia elektroniky kresla a stojana.....	70
B.4	Schéma navrhovaného elektrického riešenia – zapojenie mikrokontrolérov.....	71
B.5	Schéma navrhovaného elektrického riešenia – asynchrónny prenos.....	72
B.6	Schéma navrhovaného elektrického riešenia – prepojenie klávesnice s displejom.....	73
B.7	Schéma navrhovaného elektrického riešenia – obvody podsvietenia.....	74
C	Doska plošných spojov.....	75
C.1	Podsvietenie.....	75
C.2	Podsvietenie – osadenie súčiastok spredu.....	76
C.3	Podsvietenie – osadenie súčiastok zo zadu.....	77
C.4	Kapacitné senzory – spredu.....	78
C.5	Kapacitné senzory – zo zadu.....	79
C.6	Kapacitné senzory – osadenie súčiastok spredu.....	80
C.7	Kapacitné senzory – osadenie súčiastok zo zadu.....	81
D	Otázky na NSK Japonsko a odpovede – Reply_to_the_question_NLX_BF.pdf	
E	Manuál na ovládanie bezuhlíkového motora NLX BF od NSK Japonsko – Manual_NLX_BF.pdf	
F	CD so zdrojovými kódmi, diplomovou prácou a prílohami v elektronickej verzii	

## B Elektrická schéma

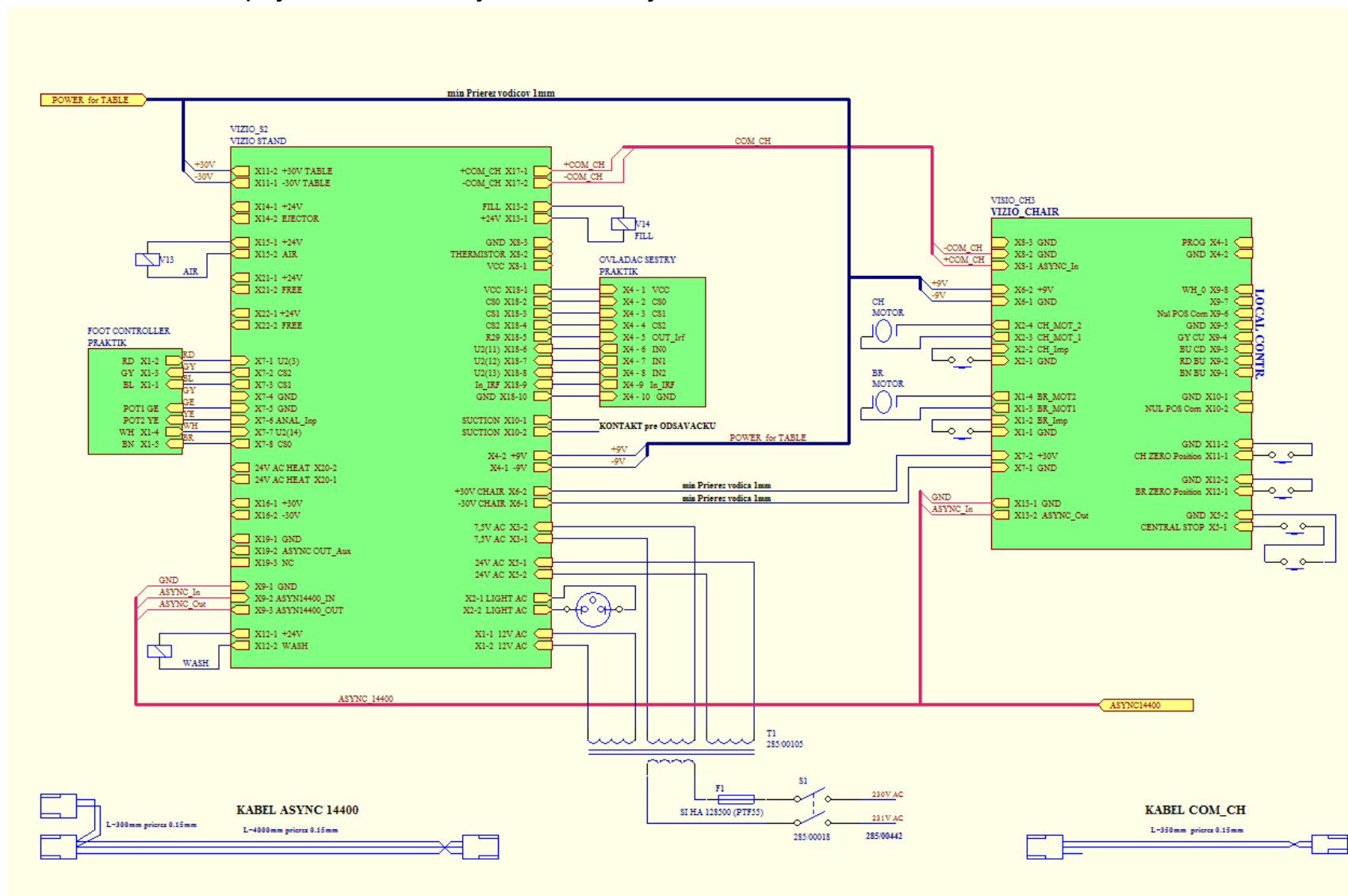
## B.1 Bloková schéma zapojenia elektroniky lekárskej dentálnej súpravy



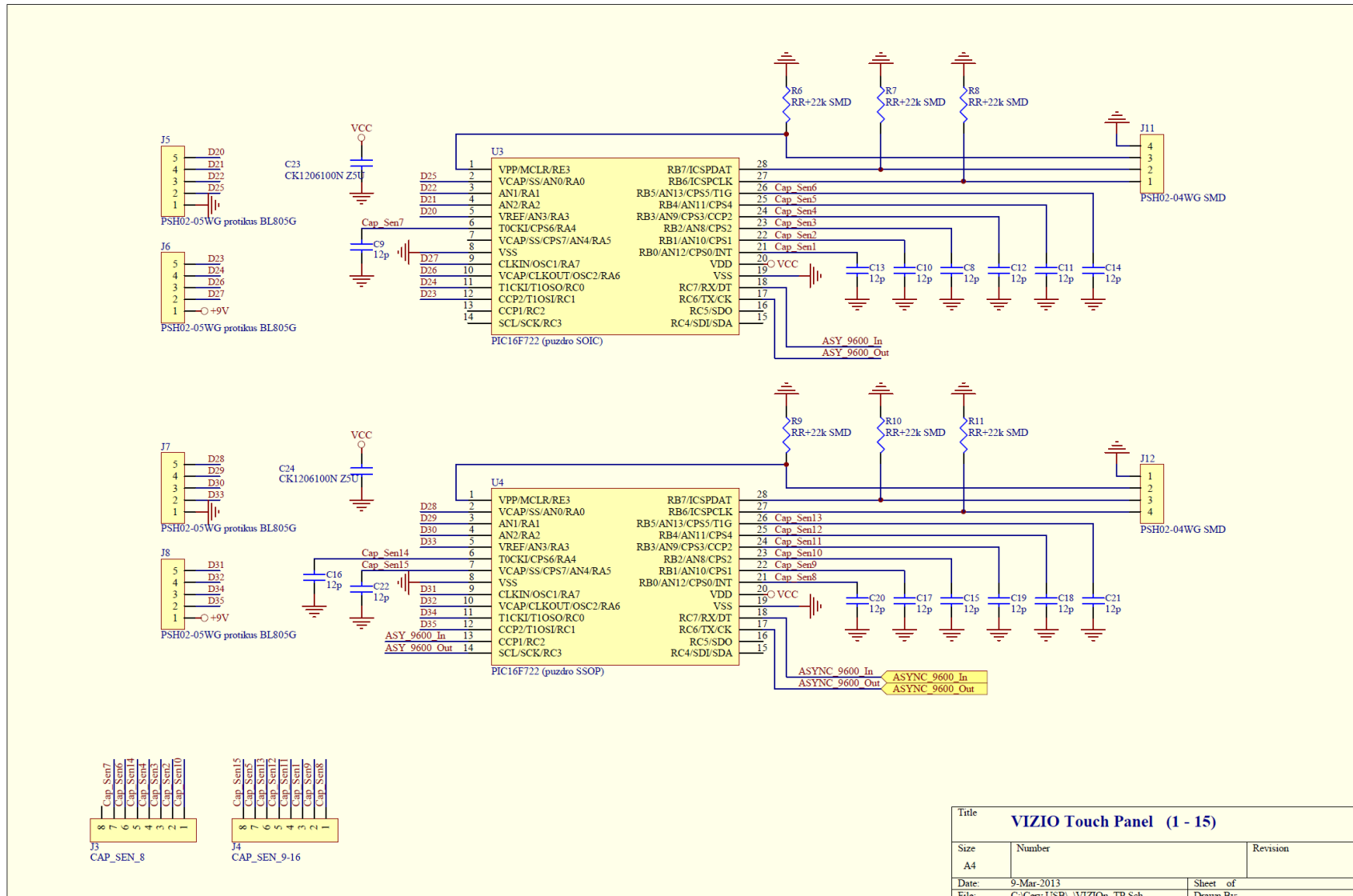
## B.2 Bloková schéma zapojenia elektroniky stolíka lekára a stojana



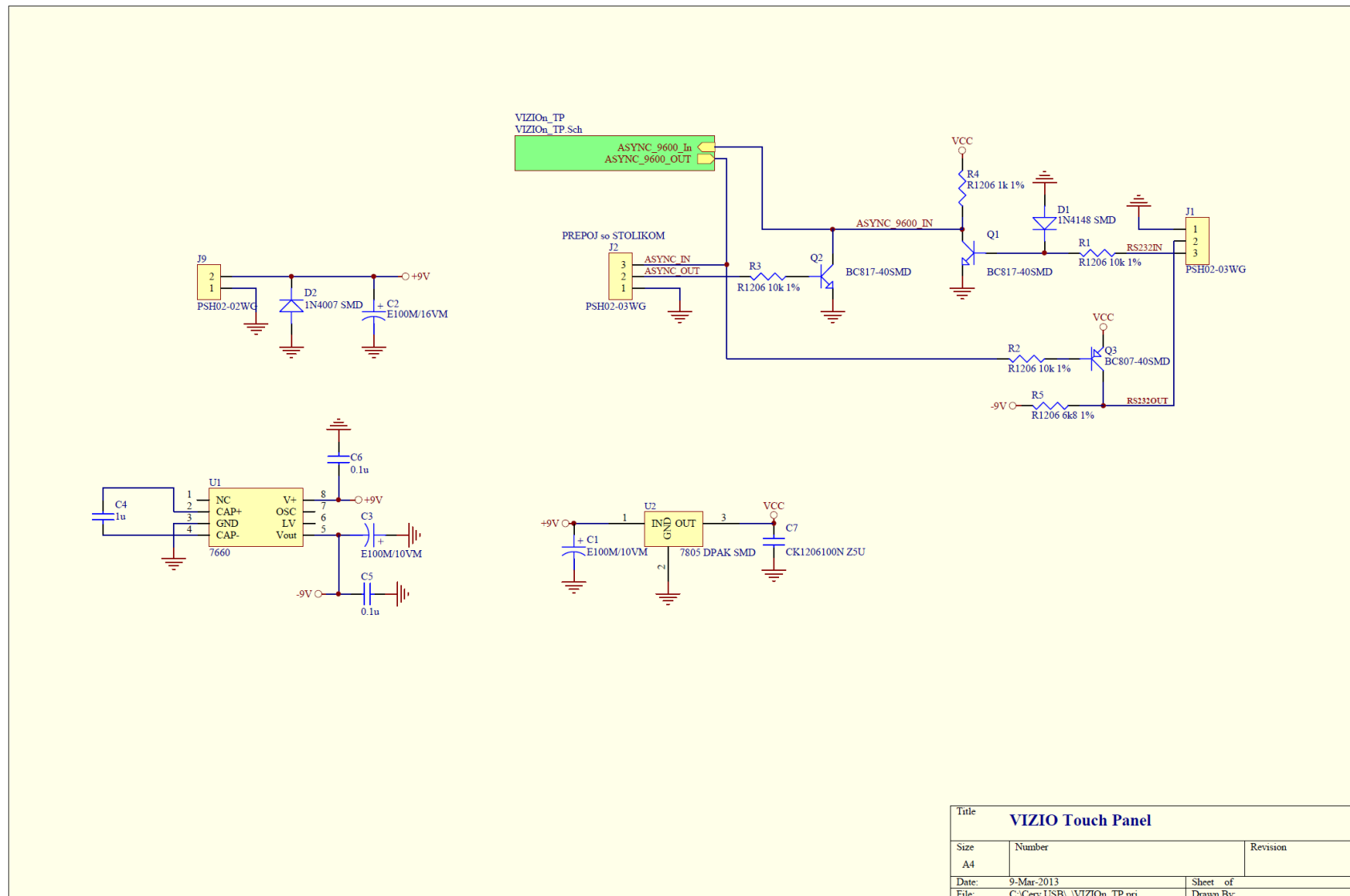
### B.3 Bloková schéma zapojenia elektroniky kresla a stojana



## B.4 Schéma navrhovaného elektrického riešenia – zapojenie mikrokontrolérov

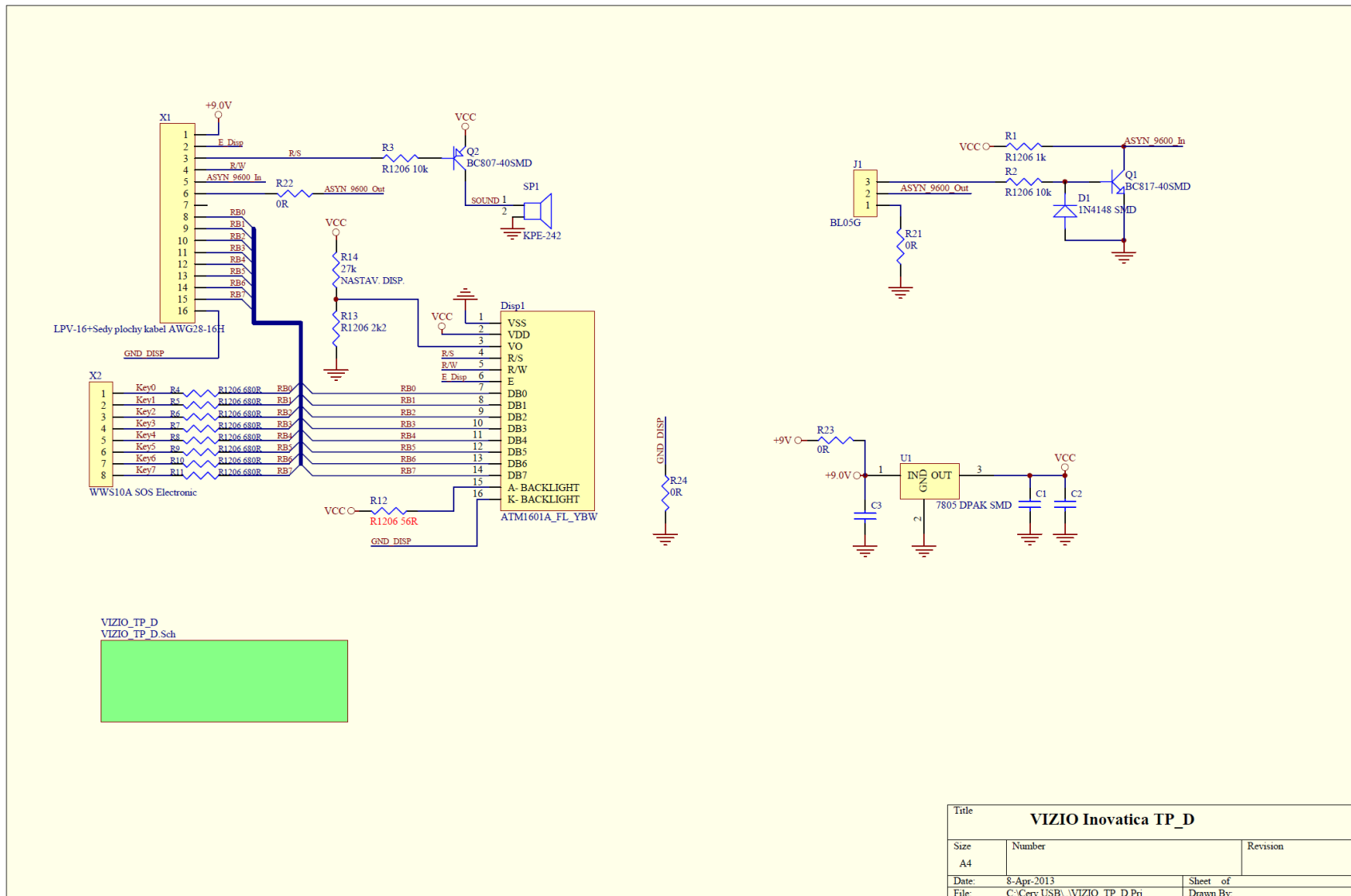


## B.5 Schéma navrhovaného elektrického riešenia – asynchrónny prenos

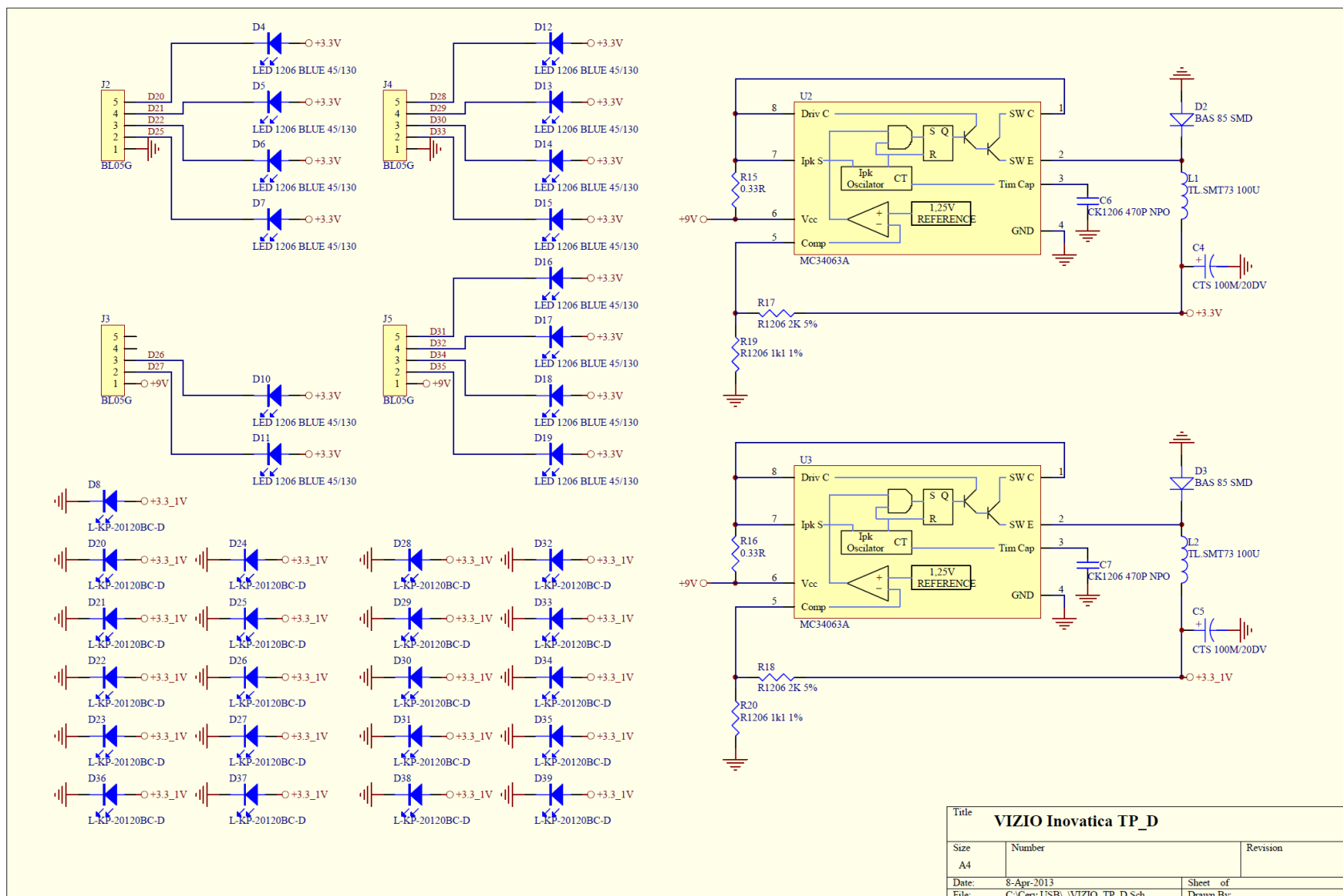




## B.6 Schéma navrhovaného elektrického riešenia – prepojenie klávesnice s displejom



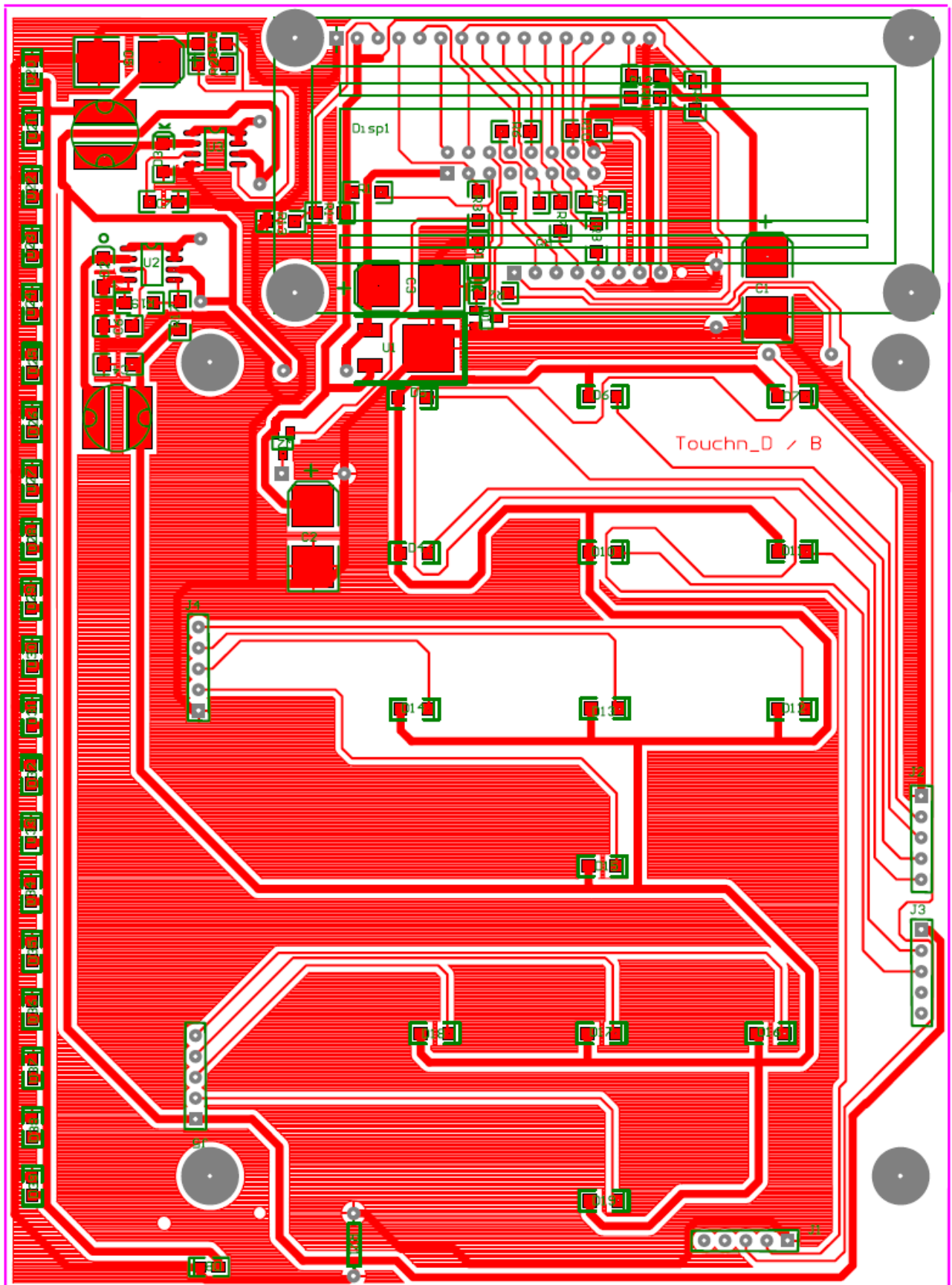
## B.7 Schéma navrhovaného elektrického riešenia – obvody podsvietenia



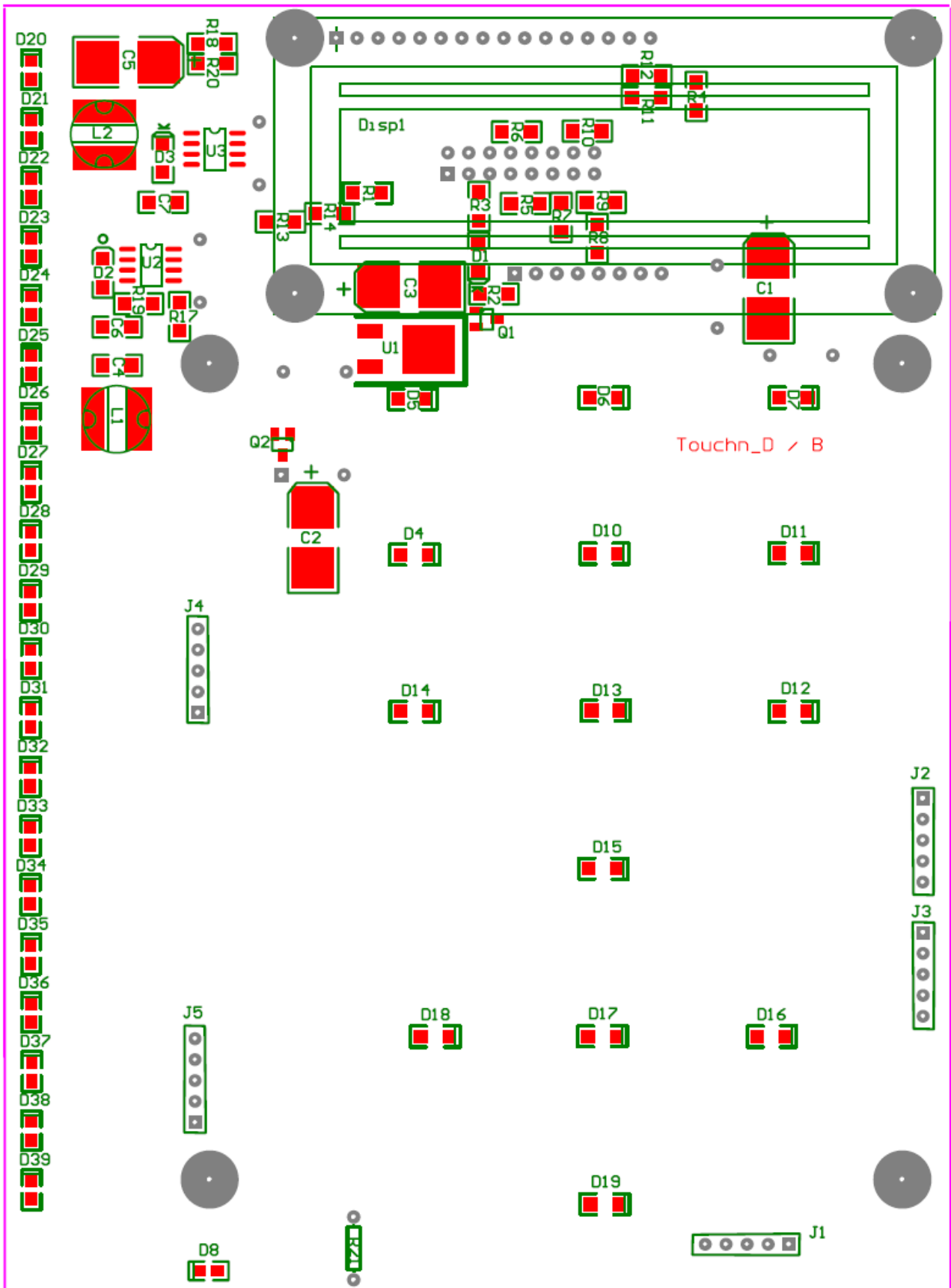
Title		
<b>VIZIO Inovatica TP_D</b>		
Size	Number	Revision
A4		
Date:	8-Apr-2013	Sheet of
File:	C:\Cerv USB\ \VIZIO TP_D.Sch	Drawn By:

## C Doska plošných spojov

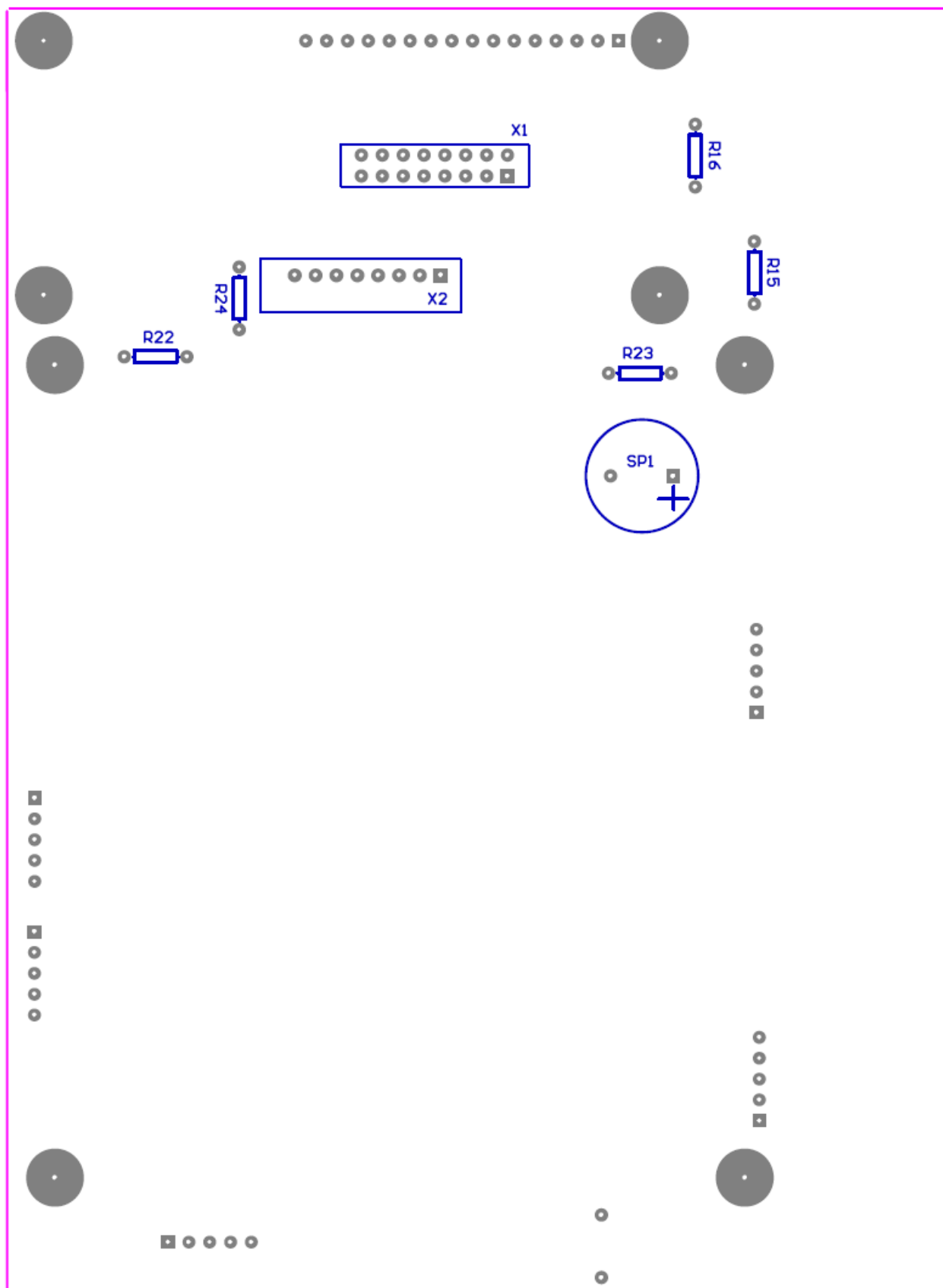
### C.1 Podsvietenie



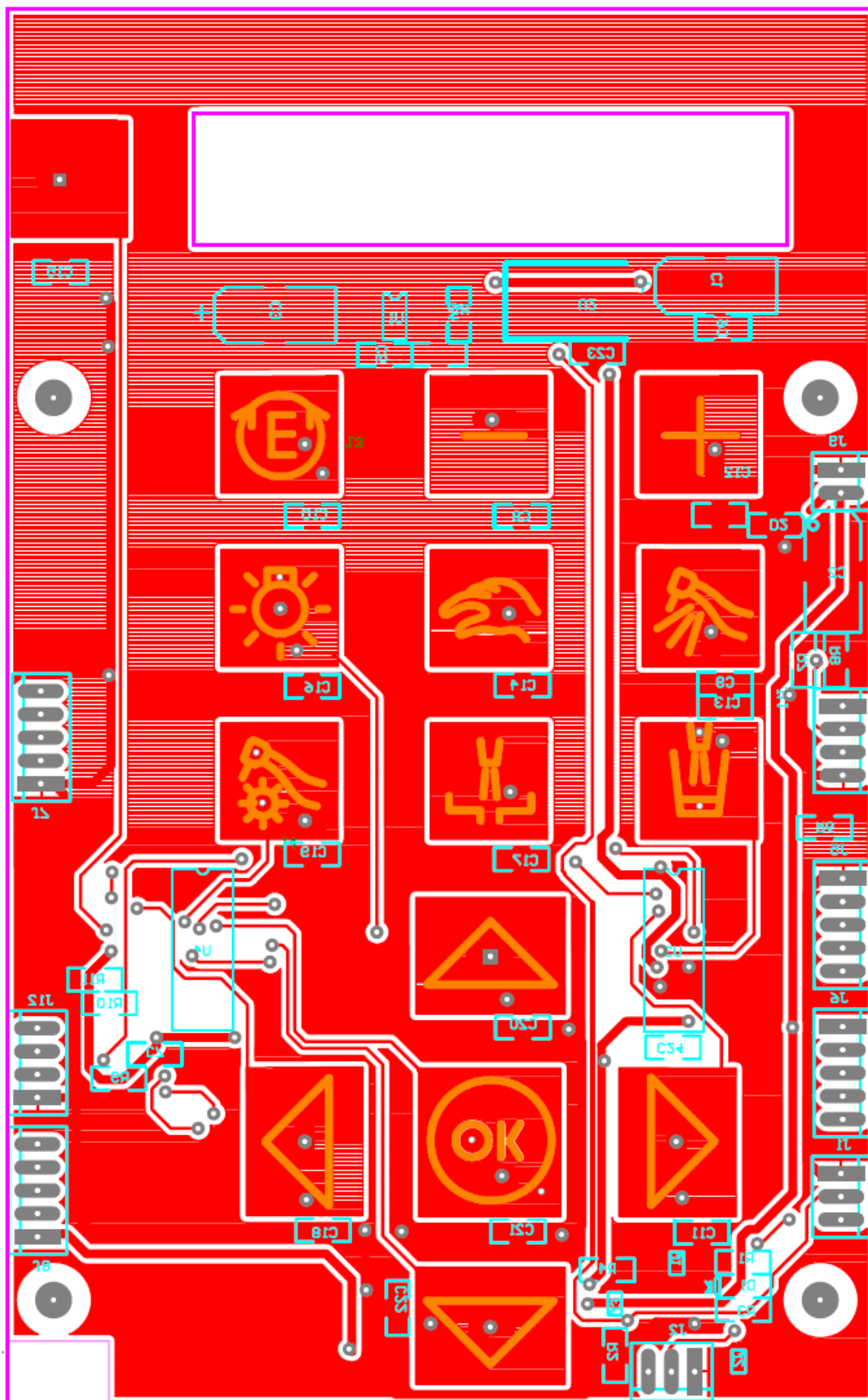
## C.2 Podsvietenie – osadenie súčiastok spredu



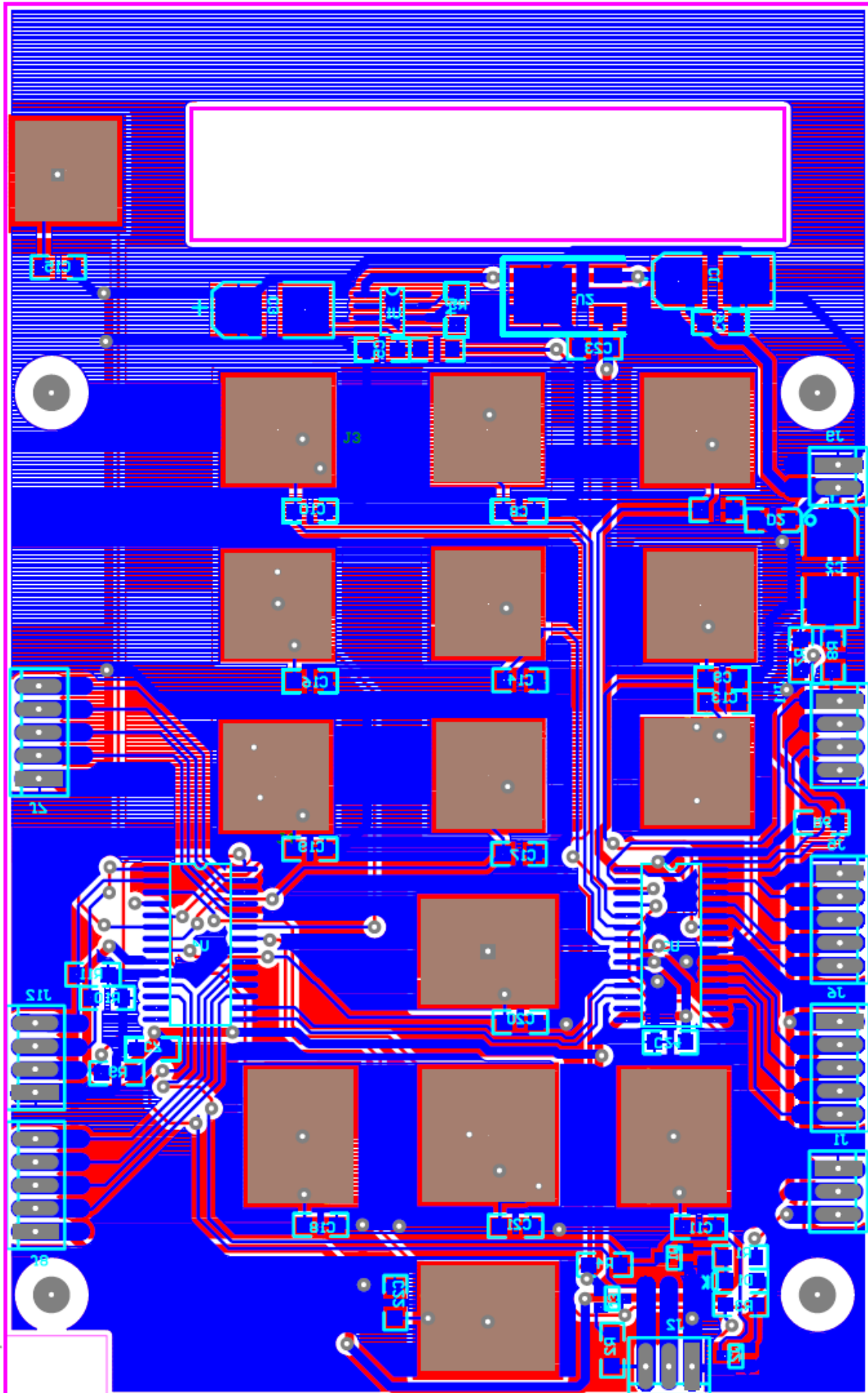
### C.3 Podsvietenie – osadenie súčiastok zo zadu



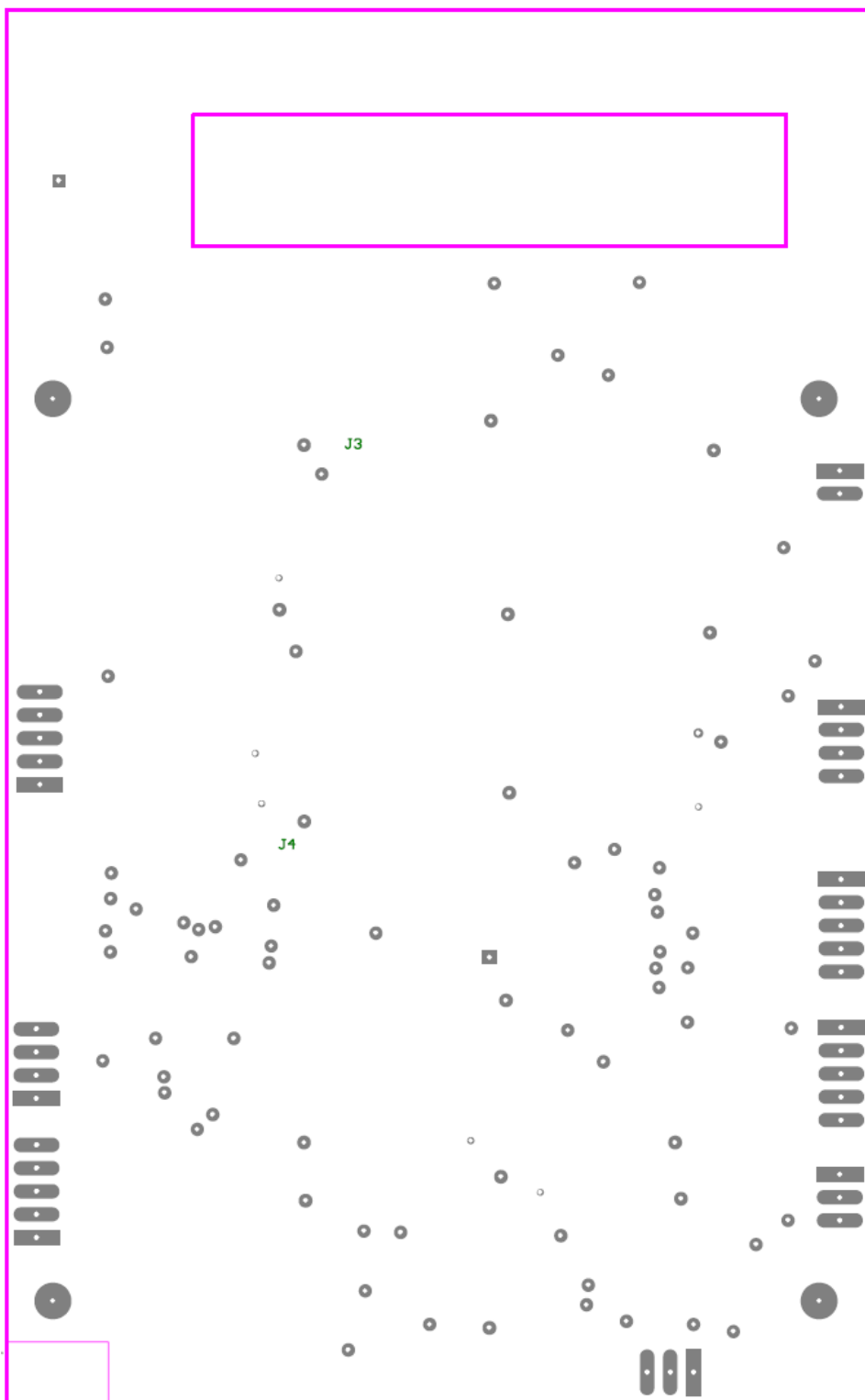
## C.4 Kapacitné senzory – spredu



## C.5 Kapacitné senzory – zo zadu



## C.6 Kapacitné senzory – osadenie súčiastok spredu





## C.7 Kapacitné senzory – osadenie súčiastok zo zadu

