



**ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE**

---

**Fakulta elektrotechnická  
Katedra radioelektroniky**

**Ovládání nemocničního lůžka pomocí mobilní aplikace**

**Controlling of hospital bed using mobile application**

Diplomová práce

Studijní program: Komunikace, multimédia a elektronika  
Obor: Multimediální technika

Vedoucí práce: Ing. Bc. Marek Neruda, Ph.D.

**Zdeněk Šubčík**

---

**Praha 2016**



České vysoké učení technické v Praze  
Fakulta elektrotechnická

katedra radioelektroniky

## ZADÁNÍ DIPLOMOVÉ PRÁCE

Student: **Zdeněk Šubčík**

Studijní program: Komunikace, multimédia a elektronika  
Obor: Multimediální technika

Název tématu: **Ovládání nemocničního lůžka pomocí mobilní aplikace**

Pokyny pro vypracování:

Navrhňte a zrealizujte mobilní aplikaci pod OS Android pro ovládání nemocničního lůžka společnosti PROMA REHA, s.r.o. Analyzujte zapojení stávajícího sesterského panelu. Navrhňte a zrealizujte DPS, která se jako modul připojí k sesterskému panelu. Naprogramujte uživatelskou aplikaci pod OS Android se stejnými funkcemi, které nabízí sesterský panel. Zrealizovaný systém otestujte.

Seznam odborné literatury:

[1] Android developers. Android [online]. 2015 [cit. 2015-09-13]. Dostupné z: <http://developer.android.com>

[2] ZÁHLAVA, Vít, Návrh a konstrukce desek plošných spojů : principy a pravidla praktického návrhu, 1. vydání, Praha: Ben, 2010.

[3] HUANG, Albert, RUDOLPH, Larry, Bluetooth for programmers, Massachusetts Institute of Technology, Cambridge (2005)

Vedoucí: Ing. Marek Neruda, Ph.D.

Platnost zadání: do konce letního semestru 2016/2017



doc. Mgr. Petr Páta, Ph.D.  
vedoucí katedry

prof. Ing. Pavel Ripka, CSc.  
děkan

V Praze dne 19. 2. 2016



## Čestné prohlášení

Prohlašuji, že jsem předloženou práci vypracoval samostatně a že jsem uvedl veškeré použité informační zdroje v souladu s Metodickým pokynem o dodržování etických principů při přípravě vysokoškolských závěrečných prací.

V Praze dne 27.5.2016

.....  
Zdeněk Šubčík

## Poděkování

Děkuji vedoucímu práce Ing. Bc. Marku Nerudovi, Ph.D. za pomoc a vstřícný přístup při tvorbě diplomové práce.

Dále bych chtěl poděkovat rodině a přítelkyni za trpělivost a pomoc během psaní této práce i po celou dobu mého studia.

## **Anotace**

Cílem této diplomové práce bylo navrhnout a realizovat bezdrátové ovládání nemocničního lůžka společnosti PROMA REHA s.r.o. K ovládání lůžka byla vytvořena mobilní aplikace Lůžko pro OS Android, která nabízí stejné funkce jako sestřerský panel. Aplikace komunikuje pomocí Bluetooth s mikrokontrolérem řídicím modul LuzkoBoard, který je připojen k řídicí jednotce sestřerského panelu. Modul emuluje stisk tlačítek a vyčítá stav LED diod na sestřerském panelu. Systém byl úspěšně otestován.

**Klíčová slova:** Android, Bluetooth, nemocniční lůžko

## **Summary**

The intention of this diploma thesis was to design and implement wireless control of smart bed, developed by PROMA REHA s.r.o. A mobile application for Android platform was developed to control smart bed. It offers the same functionality as the nurse control panel. Application communicates with LuzkoBoard module via Bluetooth, the module is connected to a control unit of the nurse panel. It emulates button press as well as it reads states of LED diodes placed on the panel. The whole system was successfully tested.

**Index terms:** Android, Bluetooth, smart bed

## Seznam obrázků

1	Oficiální logo OS Android (převzato z [4]) . . . . .	3
2	Popis jednotlivých částí lůžka (převzato z [8]) . . . . .	9
3	Nožní ovladače, rozložení tlačítek (převzato z [8]) . . . . .	11
4	Obě varianty integrovaného ovladače (převzato a upraveno z [8]) . . . . .	12
5	Rozmístění ovládacích tlačítek na ručním patientském panelu (převzato z [8])	13
6	Rozmístění ovládacích prvků na sesterském panelu . . . . .	13
7	Rozmístění stavových LED diod na sesterském panelu . . . . .	15
8	Zapojení ovládacích tlačítek na sesterském panelu (převzato z [1]) . . . . .	15
9	Řídicí jednotka sesterského panelu s označenými konektory . . . . .	17
10	Schéma zapojení pro snímání maticové klávesnice (převzato a upraveno z [10])	19
11	Časový průběh signálů s jedním stisknutým tlačítkem (převzato z [10]) . . . . .	20
12	Realizace modulu LuzkoBoard a mobilní aplikace Lůžko . . . . .	21
13	Funkční diagram analogového multiplexeru 8:1, převzato z [12] . . . . .	25
14	Schéma zapojení napájecího zdroje 3.3V . . . . .	26
15	Schéma zapojení analogových multiplexerů . . . . .	27
16	Připojení modulu LuzkoBoard k sesterskému panelu a jeho řídicí jednotce . . .	28
17	Zobrazení Menu tlačítka v <i>layoutu</i> aplikace Lůžko . . . . .	30
18	Tlačítka a symboly reflektující stav LED diod. . . . .	31
19	Diagram zobrazující navazování BT komunikace, převzato a volně přeloženo z [21] . . . . .	32
20	Dialog pro změnu Bluetooth MAC adresy modulu . . . . .	33
21	Dialog s informacemi o aplikaci . . . . .	33
22	Schéma komunikace mezi aplikací Lůžko a mikrokontrolérem . . . . .	35
23	Rámcové schéma vyhodnocení přijaté zprávy na straně mikrokontroléru . . . . .	38
24	Rámcové schéma vyhodnocení přijaté zprávy na straně mobilní aplikace . . . . .	39
25	Schéma zapojení LPC JTAG pro flashování a debugování programu . . . . .	41
26	Schéma generátoru hodin mikrokontroléru STM32L053 [22], zvýrazněny jsou použité hodiny . . . . .	42
27	Schéma programu v nekonečné smyčce pro odeslání stavů LED diod . . . . .	45

## Seznam tabulek

1	Verze OS Android s vybranými parametry a novými funkcemi [6] . . . . .	4
2	Legenda k popisu lůžka (převzato z [8]) . . . . .	10
3	Legenda k sesterskému panelu . . . . .	14
4	Připojení stavových LED diod k sesterskému panelu . . . . .	14
5	Logika tlačítek sesterského panelu (převzato a upraveno z [1]) . . . . .	16
6	Vybrané parametry Bluetooth modulu RN42 [11] . . . . .	24
7	Funkční tabulka analogových multiplexerů 74HCT4051, převzato z [12] . . . . .	25
8	Parametry výroby DPS technologií Pool Service u firmy Pragoboard s.r.o. [14]	27
9	Seznam testovacích zařízení . . . . .	29
10	Nastavení sběrnice UART na straně mikrokontroléru . . . . .	36
11	Formát zprávy ve směru aplikace Lůžko → mikrokontrolér . . . . .	36
12	Formát zprávy ve směru mikrokontrolér → aplikace Lůžko . . . . .	37
13	Nastavení použitých pinů, piny jsou seřazeny podle osazení na pouzdře LQFP48	43



# Obsah

<b>1 Úvod</b>	<b>1</b>
<b>2 Operační systém Android</b>	<b>3</b>
2.1 Historie OS Android . . . . .	3
2.2 Rozšíření aplikacemi třetích stran . . . . .	6
2.3 Alternativní ROM . . . . .	6
2.4 Komunikace pomocí Bluetooth RFCOMM . . . . .	7
<b>3 Nemocniční lůžko společnosti PROMA REHA s.r.o.</b>	<b>9</b>
3.1 Možnosti nastavení polohy . . . . .	10
3.2 Ovládací panely . . . . .	11
3.3 Sesterský panel . . . . .	13
3.3.1 Zapojení, logika tlačítek . . . . .	14
3.3.2 Možnosti připojení k panelu . . . . .	16
<b>4 Maticová klávesnice</b>	<b>19</b>
4.1 Snímání maticové klávesnice (stisk jediného tlačítka) . . . . .	19
<b>5 Řešení zadání</b>	<b>21</b>
<b>6 Návrh modulu LuzkoBoard</b>	<b>23</b>
6.1 Použité součástky . . . . .	23
6.2 Funkční části zapojení . . . . .	26
6.3 Výroba DPS . . . . .	28
6.4 Připojení modulu LuzkoBoard do sesterského panelu . . . . .	28
<b>7 Mobilní aplikace Lůžko</b>	<b>29</b>
7.1 Kompatibilita . . . . .	29
7.2 Inicializace a start aplikace . . . . .	30
7.3 Běh aplikace . . . . .	31
7.4 Komunikace s Bluetooth modulem RN42 . . . . .	32
7.5 Možnosti aplikace . . . . .	33
7.6 Podpora dalších jazyků . . . . .	33
<b>8 Komunikační protokol</b>	<b>35</b>
8.1 Parametry pro nastavení sběrnice UART . . . . .	35
8.2 Komunikační rámce . . . . .	36
8.3 Postup vyhodnocení přijaté zprávy . . . . .	37
<b>9 Řídicí program v mikrokontroléru STM32L053</b>	<b>41</b>
9.1 Naprogramování a debugování . . . . .	41
9.2 Inicializace . . . . .	41
9.3 Hlavní smyčka programu . . . . .	44
9.4 Přerušování SysTick . . . . .	45
9.5 Přerušování UART_Receive . . . . .	45

<b>10 Závěr</b>	<b>47</b>
<b>Reference</b>	<b>48</b>
<b>Seznam použitých zkratek</b>	<b>50</b>
<b>Přílohy</b>	<b>51</b>

# 1 Úvod

Mobilní telefony se staly běžnou součástí každodenního života. Z jednoduchého nástroje pro komunikaci se postupně vyvinuly do dnešní podoby tzv. chytrých telefonů, ovládaných operačním systémem. Samotné přístroje nabízejí velké množství funkcí, například komunikace přes Bluetooth a NFC nebo vysokorychlostní připojení k internetu pomocí Wi-Fi nebo mobilních sítí. V současné době je nejrozšířenějším operačním systémem pro chytré telefony a tablety Android od společnosti Google, Inc, umožňující rozšíření a aplikace třetích stran. Mobilní zařízení s OS Android nabízí díky svým funkcím široké možnosti využití. Lze ho využít jako obyčejný telefon, přehrávač multimédií, herní konzoli atd.

Cílem této práce bylo navrhnout a realizovat řešení bezdrátového ovládání nemocničního lůžka společnosti PROMA REHA s.r.o. ve stejném rozsahu, jaký nabízí sesterský panel lůžka. Vzhledem k rozšíření, dostupnosti a nízké ceně koncových zařízení se nabízí využít k ovládání lůžka mobilní telefon nebo tablet s operačním systémem Android. Bezdrátové ovládání nemocničního lůžka může výrazně zvýšit komfort lidí se sníženou schopností pohybu - mohou lůžko polohovat přímo ze svého mobilního telefonu. Toto řešení může také usnadnit práci ošetřujícího personálu, který může lůžko takto ovládat i ze vzdálenosti několika metrů.

Toto nemocniční lůžko může sloužit i pro domácí ošetřování osob s dlouhotrvajícími zdravotními potížemi. Zde má ovládání pomocí mobilního telefonu nebo tabletu větší význam, než v nemocnici, kde je neustále nablízku ošetřující personál. Všechny možnosti polohování lůžka totiž nabízí pouze sesterský panel, který bývá zavěšen na hrazení postele u nohou pacienta. Toto umístění je dáno polohou vstupního konektoru a délkou přívodního kabelu. Jeho přemístění na dosah ležícího pacienta je tedy limitováno délkou přívodního kabelu, navíc je pro manipulaci s panelem zapotřebí obou rukou. Mobilní telefon ale bývá umístěn v dosahu pacienta a lze ho pohodlně ovládat jednou rukou.

Jako základ práce jsem využil bakalářskou práci Filipa Andrásyho [1], ve které analyzoval zapojení v sesterském panelu a realizoval ovládání stehenního dílu lůžka pomocí IOIO Boardu. Z této práce jsem převzal tabulku zapojení tlačítek na sesterském panelu (s úpravou zapojení dvou tlačítek, která neodpovídala tabulce) a obrázek zobrazující toto zapojení. Dále jsem použil layout aplikace, dodaný firmou PROMA REHA s.r.o., ve kterém jsem upravil vzhled tlačítek pro uzamykání funkcí a přidal jsem tlačítko pro zobrazení kontextové nabídky.

Pro ovládání lůžka jsem místo IOIO Boardu navrhl a realizoval vlastní modul, který se nasadí na konektory řídicí jednotky sesterského panelu, aniž by omezil funkce samotného panelu. Napájí se ze stejného zdroje jako řídicí jednotka, nepotřebuje tedy vlastní zdroj energie. Dále jsem vyvinul mobilní aplikaci, která komunikuje s tímto modulem pomocí emulované sériové linky přes Bluetooth a nabízí všechny funkce sesterského panelu včetně zobrazení stavových LED diod.



## 2 Operační systém Android

V současné době (květen 2016) je Android nejrozšířenějším operačním systémem pro mobilní telefony a tablety, s podílem na trhu přes 60% [2]. Jeho největšími konkurenty jsou iOS, vyvíjený Apple, Inc., a Windows Phone, vyvíjený firmou Microsoft Corporation.

Android vyvíjí Open Handset Alliance (OHA) [3], jejímž cílem je vývoj veřejně dostupných a použitelných standardů (*open standards*) pro mobilní zařízení. OHA byla založena v roce 2007 a dnes již sdružuje více než 80 firem, zabývajících se výrobou, vývojem, provozováním nebo prodejem mobilních zařízení. Seznam jejich členů lze rozdělit do následujících kategorií:

- Mobilní operátoři (např. T-mobile, Telefonica)
- Výrobci koncových zařízení (Acer, Dell, Huawei, Samsung, ...)
- Výrobci polovodičových součástek (ARM, Intel, ...)
- Softwarové společnosti (Google Inc., eBay Inc., NXP Software, ...)
- Obchodní společnosti (Wipro Technologies, Teleca AB, ...)



Obr. 1: Oficiální logo OS Android (převzato z [4])

OS Android nabízí širokou podporu možností komunikace a zabudovaných senzorů. Díky tomu je možné zařízení využívat i jinak, než jen jako mobilní telefon. Pro rozšíření jeho využití je k dispozici např. internetový obchod Google Play, nabízející obrovské množství rozšiřujících aplikací a multimediálního obsahu.

Jak již bylo zmíněno, Android nabízí vedle funkcí mobilního telefonu (v Evropě podpora mobilních sítí GSM, UMTS, LTE a LTE Advanced v závislosti na zařízení) i podporu velkého množství dalších možností komunikace (Wi-Fi, Bluetooth, NFC, ...). Součástí systému je i podpora satelitní navigace (GPS, GLONASS). Mimo to Android podporuje velké množství vnitřních senzorů, jako je akcelerometr, gyroskop, snímače teploty a vlhkosti, snímače intenzity osvětlení nebo barometr.

### 2.1 Historie OS Android

Vývoj Androidu začal v roce 2003 ve společnosti Android Inc. Její spoluzakladatel Andy Rubin později v rozhovoru pro Business Week uvedl, že prvotní myšlenka byla vytvořit takový systém, který si bude moct uživatel přizpůsobit co nejvíce podle svých představ. Po dvou letech se ale firma dostala do finančních potíží a i proto byla odkoupena společností Google Inc., která tímto získala základ pro vývoj vlastního operačního systému pro mobilní zařízení. Původní koncept předpokládal ovládání pomocí vysouvací QWERTY klávesnice,

bez dotykového displeje. Když ale v roce 2007 představila firma Apple, Inc. svůj iPhone s ovládáním výhradně pomocí dotykového displeje, byl koncept zcela přepracován.

V říjnu 2008 byla představena první verze Androida, nainstalovaná na telefon HTC dream. Telefon stále měl vysouvací QWERTY klávesnici, ale přibyla možnost ovládání pomocí kapacitního dotykového displeje. Tomu byl také přizpůsoben design systému. Ovládací prvky musely být dostatečně velké, aby je bylo možné od sebe spolehlivě rozlišit při dotyku prstu, který je oproti kurzoru myši nebo stylusu značně nepřesný. Na druhou stranu toto řešení výrazně zvyšuje uživatelský komfort, není nutné s sebou pro ovládání telefonu nosit stylus a telefon lze ovládat jednou rukou.

Základem pro využití celého potenciálu mobilního zařízení jsou kvalitní a rozmanité aplikace, od kvalitního ovládání fotoaparátu přes internetový prohlížeč až po hry. V době uvedení na trh už musela Open Handset Alliance spolu se samotným telefonem nabízet i dostatečné množství kvalitních aplikací, pro rozmanitost nejlépe od různých vývojářů. Proto byla vypsaná veřejná soutěž, ve které byly odměněny nejlepší a nejužitečnější aplikace pro Android. Díky tomuto kroku měl Android k dispozici kromě kvalitních aplikací i početnou vývojářskou základnu. Různé soutěže jsou stále vypisovány, možnosti využití OS Android ještě stále nejsou zcela pokryty - jako příklad může posloužit výzva k vytvoření aplikace, využívající službu Google Fit<sup>1</sup> [5]. Navíc je nutné stávající aplikace udržovat v souladu s požadavky každé nové verze systému.

Android používá pro rozlišení verzí jednak úroveň API<sup>2</sup>, dále číslo verze a nakonec kódové označení. Kódové označení vzniklo až při uvedení třetí verze API, údajně proto, že uživatelům se snadněji zapamatují názvy verzí v této podobě. Kódové označení každé verze začíná na příslušné písmeno v abecedě a nese název zákusku (Cupcake, Gingerbread, KitKat, ...). Jedno označení sdružuje větší počet verzí a mění se, pokud s novou verzí přijdou výraznější změny systému.

Přehled verzí od vzniku po současnost (květen 2016) nabízí tabulka 1. Pro rozlišení verzí bylo zvoleno kódové označení, protože se, jak už bylo zmíněno, mění s těmi verzemi, které přinášejí výraznější změny nebo nové funkce.

Tabulka 1: Verze OS Android s vybranými parametry a novými funkcemi [6]

Kódové označení	číslo verze	úroveň API	vydání první verze
<i>bez označení</i>	<b>1.0</b>	<b>1</b>	<b>září 2008</b>
<ul style="list-style-type: none"> <li>- první oficiální verze, obchod Android Market</li> <li>- integrované služby Google (Gmail, Google Maps, ...)</li> <li>- podpora Wi-Fi, Bluetooth. fotoaparát</li> </ul>			
<i>bez označení</i>	<b>1.1</b>	<b>2</b>	<b>duben 2009</b>
<ul style="list-style-type: none"> <li>- drobné úpravy, odstranění známých chyb</li> </ul>			
<b>Cupcake</b>	<b>1.5</b>	<b>3</b>	<b>srpen 2009</b>
<ul style="list-style-type: none"> <li>- přidána podpora Widgetů na domovské obrazovce</li> <li>- automatické otáčení obrazovky podle senzoru</li> <li>- podpora klávesnic třetích stran, uživatelský slovník</li> </ul>			
<b>Donut</b>	<b>1.6</b>	<b>4</b>	<b>září 2009</b>
<ul style="list-style-type: none"> <li>- hlasové vyhledávání</li> <li>- podpora rozlišení displeje až WVGA</li> </ul>			

<sup>1</sup>Google Fit je platforma pro podporu aplikací sledujících zdravotní funkce uživatele

<sup>2</sup>Application Programming Interface, rozhraní pro programování aplikací

<b>Kódové označení</b>	<b>číslo verze</b>	<b>úroveň API</b>	<b>vydání první verze</b>
<b>Eclair</b>	<b>2.0 - 2.1</b>	<b>5 - 7</b>	<b>říjen 2009</b>
<ul style="list-style-type: none"> <li>- podpora Bluetooth 2.1</li> <li>- podpora HTML5</li> <li>- vylepšení funkcí fotoaparátu</li> <li>- synchronizace e-mailů a kontaktů, podpora více uživatelských účtů současně</li> </ul>			
<b>Froyo</b>	<b>2.2 - 2.2.3</b>	<b>8</b>	<b>květen 2010</b>
<ul style="list-style-type: none"> <li>- Optimalizace systému, zrychlení odezvy, podpora displejů s rozlišením až 720p</li> <li>- podpora Wi-Fi hotspot a USB tethering</li> <li>- podpora Adobe Flash</li> <li>- možnost přesunutí aplikací na externí úložiště</li> </ul>			
<b>Gingerbread</b>	<b>2.3 - 2.3.7</b>	<b>9 - 10</b>	<b>prosinec 2010</b>
<ul style="list-style-type: none"> <li>- podpora internetového volání SIP VoIP</li> <li>- podpora Near Field Communication (NFC)</li> <li>- vylepšení sledování procesů na pozadí pro vyšší výdrž baterie</li> </ul>			
<b>Honeycomb</b>	<b>3.0 - 3.2.6</b>	<b>11 - 13</b>	<b>únor 2011</b>
<ul style="list-style-type: none"> <li>- optimalizace pro tablety, změna uživatelského prostředí</li> <li>- podpora Fragmentů<sup>1</sup> v layoutu aplikace pro lepší využití velikosti displeje</li> <li>- přidána Action Bar - lišta s nástroji a funkčními tlačítky aplikace</li> <li>- změněn multitasking a jeho vyvolání</li> <li>- podpora hardwarové akcelerace, možnost připojení USB On-The-Go</li> </ul>			
<b>Ice Cream Sandwich</b>	<b>4.0 - 4.0.4</b>	<b>14 - 15</b>	<b>říjen 2011</b>
<ul style="list-style-type: none"> <li>- nové uživatelské rozhraní Holo, softwarová ovládací tlačítka</li> <li>- společná verze systému pro telefony a tablety</li> <li>- vyšší stupeň integrace senzorů do systému</li> </ul>			
<b>Jelly Bean</b>	<b>4.1 - 4.3.1</b>	<b>16 - 18</b>	<b>červen 2012</b>
<ul style="list-style-type: none"> <li>- zrychlení uživatelské odezvy s možností vývojářského ladění</li> <li>- rychlý přístup k nastavení pomocí notifikační lišty</li> <li>- podpora USB audio, dekodování AAC 5.1</li> <li>- podpora Bluetooth low energy</li> <li>- přenos dat pomocí Android Beam</li> <li>- podpora displejů s rozlišením až 4K</li> </ul>			
<b>KitKat</b>	<b>4.4 - 4.4.4</b>	<b>19 - 20</b>	<b>říjen 2013</b>
<ul style="list-style-type: none"> <li>- oficiálně končí podpora hardwarových tlačítek Back, Home a Menu</li> <li>- vylepšení vývojářských funkcí</li> <li>- podpora platformy Android Wear pro chytré hodinky</li> <li>- přechod na jádro Chromium pro zobrazování webového obsahu</li> </ul>			
<b>Lollipop</b>	<b>5.0 - 5.1.1</b>	<b>21 - 22</b>	<b>listopad 2014</b>
<ul style="list-style-type: none"> <li>- podpora 64-bitových procesorů</li> <li>- nové uživatelské rozhraní Material Design</li> <li>- snížení spotřeby a vylepšení úsporného módu (projekt Volta)</li> </ul>			
<b>Marshmallow</b>	<b>6.0 - 6.0.1*</b>	<b>23*</b>	<b>říjen 2015</b>
<ul style="list-style-type: none"> <li>- podpora USB typu C</li> <li>- zabudovaná podpora pro čtení otisků prstů</li> <li>- změny v řízení oprávnění aplikací</li> <li>- experimentální zobrazení více oken současně</li> </ul>			

<sup>1</sup> Fragmentsy slouží ke zobrazení více aktivních částí aplikace současně

\* stav v květnu 2016

## 2.2 Rozšíření aplikacemi třetích stran

Android nabízí podporu pro vývoj aplikací nezávislými vývojáři a dává jim k dispozici SDK<sup>3</sup> pro vývoj a testování. S každou novou verzí systému (úroveň API) je vydáván i nový SDK, spolu s poznámkami k nové verzi a ukázkami kódu. Návody a vzorové aplikace lze najít na internetových stránkách <https://developer.android.com>, jedná se o oficiální stránku pro podporu vývoje aplikací třetích stran. Dále je zdarma k dispozici vývojové prostředí Android Studio nebo ADT<sup>4</sup> plugin pro Eclipse.

Při vývoji nových aplikací je stále třeba dbát na zpětnou kompatibilitu. Podíl zastoupení starších verzí Androidu sice postupně klesá, ale stále je standardem vývoj aplikací kompatibilních se staršími verzemi až po API 10 (Android 2.3.3 Gingerbread). K zajištění podpory starších verzí je vývojářům k dispozici *Android Support Library*, nicméně některé funkce novějších API nemají pro starší verze alternativu.

K distribuci aplikací mezi uživatele je určen internetový obchod Google Play. Pro stahování aplikací zde stačí mít založený účet, pro nahrávání vlastních aplikací je zapotřebí vývojářský účet. Jeho založení je sice zpoplatněno, ale nabízí za to další možnosti testování aplikací pomocí vývojářské konzole. Ta umožňuje optimalizaci spotřeby a výkonu a testování nových verzí aplikace dříve, než je uveřejní. Před zveřejněním ještě musí aplikace projít kontrolou, zda neobsahuje škodlivý software.

Nahrané aplikace je možné zpoplatnit nebo nabídnout ke stažení zdarma.

## 2.3 Alternativní ROM

Android je *open-source* - to znamená, že jeho zdrojový kód je dostupný pod veřejnou licenci, konkrétně *Apache Software Licence 2.0*, a tyto zdrojové kódy lze legálně upravovat. Toho využívá např. společnost Xiaomi Tech, která vyvíjí vlastní verze ROM s názvem **MIUI**. Spolu s ní patří mezi nejrozšířenější alternativní verze Androidu **CyanogenMod**, vyvíjený CyanogenMod Open-Source Community.

Vývoji alternativních ROM výrazně pomáhá fakt, že do OS Android je integrováno velké množství předinstalovaných aplikací. Ty lze sice od verze 4.0 zakázat (neobjeví se mezi nainstalovanými aplikacemi a nelze je spustit), ale zůstávají v zařízení nainstalované a zabírají tak místo na RAM. Zařízení s čistou instalací Androidu 4.4 včetně předinstalovaných aplikací má zabraných přibližně 600 až 700 MB RAM. To je problém především u levnějších a starších modelů s menší operační pamětí - např. Android 4.4 vyžaduje pro plynulý běh systému minimálně 1 GB RAM, při větším množství nainstalovaných aplikací je doporučeno alespoň 2 GB.

Řešením je pak instalace alternativního OS, např. CyanogenMod. Ten se při běžném provozu téměř neliší od oficiální verze Androidu, ale umožňuje větší kontrolu nad nainstalovanými aplikacemi.

CyanogenMod podporuje i aplikace třetích stran, vyvíjené pro OS Android. Není třeba žádných úprav pro kompatibilitu s tímto systémem. Mobilní aplikace *Lůžko*, vyvinutá jako součást této diplomové práce, byla testována i na zařízení s tímto systémem, konkrétně verze 12.1, která vychází z Androidu 5.1.1. Jediné problémy se objevily při nastavení lokalizace (jazyka), aplikaci se nepodařilo přepnout do české lokalizace, veškeré ostatní funkce a nastavení fungovaly bez problému. Aplikace byla následně otestována i na Androidu 5.0.1 se stejným výsledkem, takže se zjevně jedná o změnu nastavení lokalizace na Androidu 5.0 a vyšším, nikoliv o nekompatibilitu CyanogenMod.

<sup>3</sup>Software Development Kit, sada knihoven pro tvorbu softwaru pro např. konkrétní operační systém

<sup>4</sup>Android Development Tools, plugin pro vývoj aplikací pro OS Android



## 2.4 Komunikace pomocí Bluetooth RFCOMM

Android nabízí podporu protokolu Bluetooth RFCOMM[7] (*Radio Frequency Communication*) od verze 2.0 Eclair (API 5). Jedná se o protokol využívající vrstvu L2CAP (*The Logical Link Control and Adaption Protocol*), která tvoří transportní vrstvu Bluetooth komunikace a zajišťuje výměnu paketů. Pomocí protokolu RFCOMM lze emulovat komunikaci po sériové lince RS232. Tento protokol umožňuje v závislosti na implementaci podporu pro až 60 připojených zařízení současně.

V aplikaci Lůžko je tento protokol využit pro komunikaci s Bluetooth modulem RN42. Není využita možnost připojení pomocí Bluetooth Low Energy, protože tu nabízí až Android 4.3 Jelly Bean (API 18) - aplikace Lůžko byla vyvinuta pro kompatibilitu i se staršími verzemi OS Android.



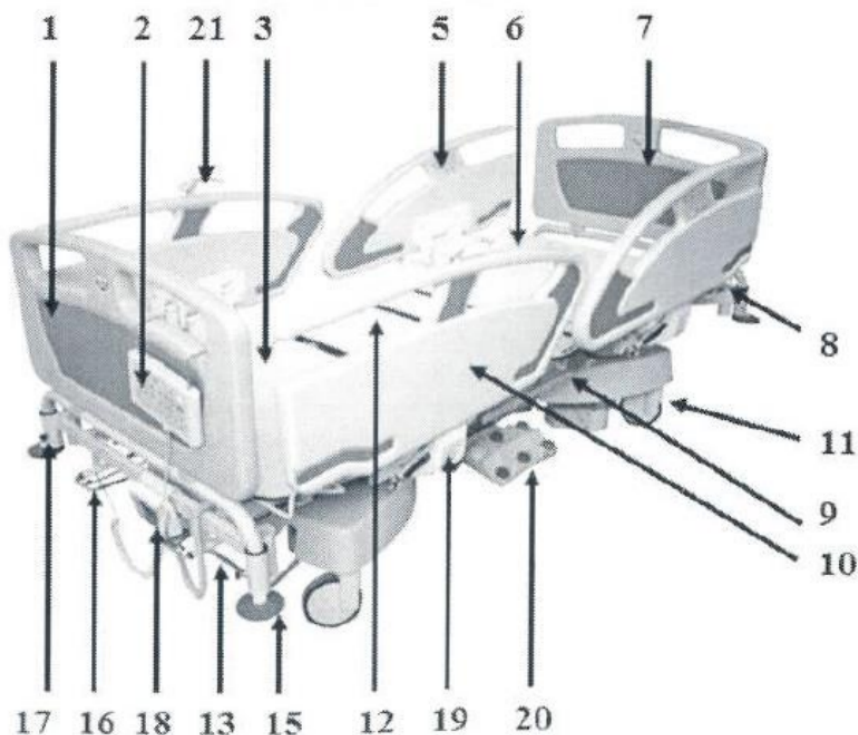
### 3 Nemocniční lůžko společnosti PROMA REHA s.r.o.

Polohovatelné nemocniční lůžko má obrovský význam pro zvýšení životního komfortu osob, které jsou ze zdravotních důvodů po delší časový úsek nuceny setrvat na pečovatelském lůžku. Umožňuje jim totiž nastavení polohy lůžka bez nutnosti asistence ošetřovatele. Takové lůžko ale může usnadnit i práci ošetřovatelům. Oproti běžnému nemocničnímu lůžku je zde k dispozici více možností nastavení polohy včetně takových, které usnadní některé životně důležité úkony - např. poloha pro resuscitaci.

Nemocniční lůžko společnosti PROMA REHA s.r.o. [8] můžeme díky široké škále nastavení polohy a možnosti ovládání několika různými panely zařadit mezi "inteligentní" lůžka. Mezi jeho funkce patří samostatný zdvih částí ložné plochy, zdvih celého lůžka nebo laterální náklon, který umožňuje pacientovi ležet podle potřeby na levém nebo pravém boku při nižším zatížení příslušné poloviny těla. Tuto funkci ocení především lidé s poraněním, které jim neumožňuje ležet na jednom boku, ale i pacienti, kteří musí na lůžku setrvat delší dobu a hrozí jim proleženiny.

Jednotlivé části lůžka se vždy polohují stisknutím a podržením příslušného tlačítka po dobu nezbytně nutnou k napolohování do požadované pozice. Uvolněním tlačítka dojde k automatické aretaci v právě nastavené poloze. Při dosažení krajní polohy lze další pohyb uskutečnit pouze opačným směrem.

K základnímu vybavení patří hrazda s hrazdičkou, která snese zatížení až 75 kg a umožňuje pacientovi vypomocet si při sedání rukama, držák na lahve s kyslíkem, teleskopický výsuvný infuzní stojan a univerzální držáky na příslušenství v každém rohu lůžka. Mezi volitelnou výbavu patří například stolek na přístroje s aretací nebo trakční systém, který je určen pro montáž na čelo lůžka.



Obr. 2: Popis jednotlivých částí lůžka (převzato z [8])

Tabulka 2: Legenda k popisu lůžka (převzato z [8])

1	Nožní čelo
2	Sesterský panel
3	Lýtkový díl
4	Polohovací hřeben
5	Integrovaný ovladač sklopné postranice
6	Zádový díl
7	Hlavové čelo
8	Univerzální držák příslušenství
9	CPR páčka
10	Sklopná postranice
11	Směrové aretační kolečko
12	Stehenní díl
13	Ovládání centrální brzdy
14	Aretace prodloužení ložné plochy
15	Nárazové kolečko
16	Výsuvná polička na lůžkoviny
17	Aretace zajištění čela
18	Panel vázícího systému
19	Ovládání zámku sklopných postranic
20	Nožní ovladač
21	Integrované madlo

### 3.1 Možnosti nastavení polohy

Nemocniční lůžko společnosti PROMA REHA s.r.o. nabízí široké možnosti nastavení polohy. Lze samostatně polohovat zádový díl až do polohy vsedě a stehenní díl pro podepření dolních končetin. Dále lze nastavit celkovou výšku lůžka a laterální náklon.

Toto lůžko také nabízí polohy, usnadňující práci personálu:

- **Trendelenburgova poloha:** Jedná se o polohu, kdy jsou pánev a nohy pacienta nad úrovní hlavy. Tato poloha se používá např. při šokovém stavu, kdy je vyžadováno větší prokrvení hlavy, nebo při některých operacích. Náklon se pohybuje v rozmezí 15 – 30° [9].
- **Obrácená Trendelenburgova poloha (antitrendelenburgova poloha):** Stejně nastavení lůžka jako u Trendelenburgovy polohy, ale s opačným náklonem, tedy takovým, aby hlava byla výš než pánev a dolní končetiny. Tato poloha se používá např. pro lepší prokrvení dolních končetin při onemocnění tepen.
- **Autokontura:** polohování zádového a stehenního dílu současně
- **Kardiacké křeslo:** poloha vsedě se zapřenými dolními končetinami. Podporuje přirozenou činnost plic, pomáhá při drenážích nebo odvodňování, především ale usnadňuje srdeční činnost pacienta.
- **Vyšetřovací poloha:** Poloha vleže, rovně na zádech. Umožňuje snadnější vyšetření pacienta, celé jeho tělo leží v rovině.
- **Resuscitační poloha (CRP):** Poloha na zádech se zakloněnou hlavou, v této poloze má pacient dobře průchozí dýchací trubici. Jedná se o polohu vhodnou pro nepřímou srdeční masáž.

- **Pozice pro snadné vstávání z lůžka:** Poloha, kde je zvednutý zádový díl a stehenní, bederní a lýtkové díly v rovině. Pacient je v poloze v sedě s narovnanými dolními končetinami, z této pozice stačí spustit nohy přes levý nebo pravý okraj lůžka. Tím se pacient ocitne v sedě s chodidly opřeny o zem a je pro něj snadnější bez asistence vstát.

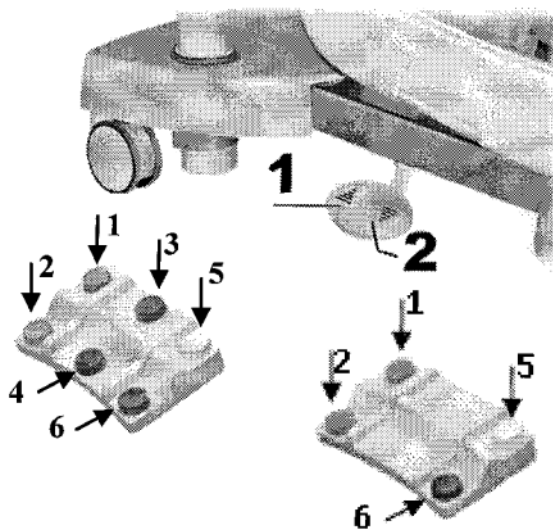
### 3.2 Ovládací panely

Lůžko nabízí možnost ovládání hned z několika samostatných panelů. Každý z nich nabízí jiné možnosti polohování a mají vzájemně určenou prioritu - obecně platí, že ze sesterského panelu je možné uzamknout určité funkce ostatních panelů. Důvodem pro toto opatření může být například situace, kdy pacient nespolupracuje a snaží se změnit polohu lůžka proti pokynům personálu - sesterský panel pak umožňuje uzamknout ovládání lůžka z ostatních panelů po dobu 30-ti sekund, během kterých může personál pacienta uklidnit.

Nejvíce funkcí nabízí **sesterský panel**. Protože byl v této práci využit pro realizaci bezdrátového ovládání lůžka, bude jeho popis podrobnější než u ostatních panelů a je mu věnována samostatná kapitola 3.3. Další ovládací panely jsou nožní ovladač, integrované ovladače a ruční patientský panel.

- **Nožní ovladače zdvihu ložné plochy:**

Součástí lůžka mohou být až dva nožní ovladače osazené 6, 4 nebo 2 tlačítkovými nožními spínači. Tlačítka reagují na stisk a podržení, během kterého je nastavována příslušná část lůžka do požadované polohy, po uvolnění tlačítka dojde k automatické aretaci právě nastavené části.

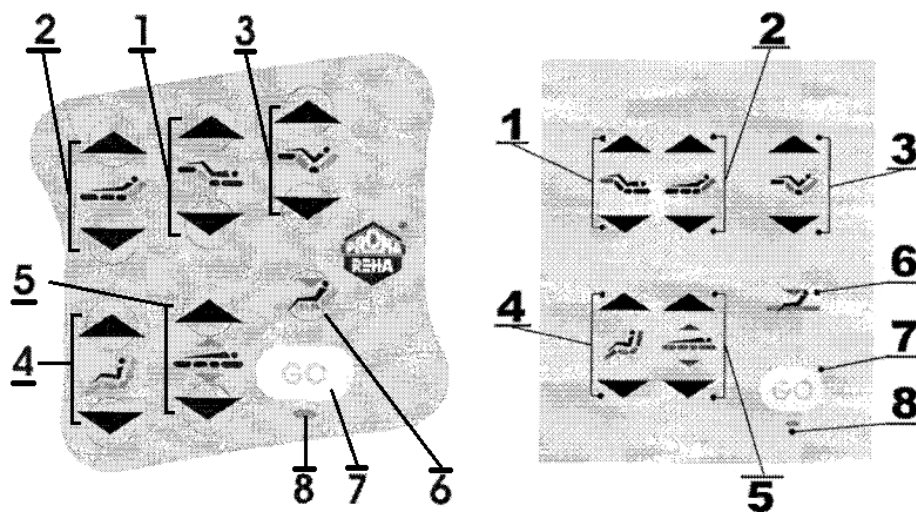


Obr. 3: Nožní ovladače, rozložení tlačítek (převzato z [8])

Funkce tlačítek (jsou-li osazena) je následující:

- 1, 2: Zdvih celé ložní plochy (nahoru / dolů)
  - 3, 4: Laterální náklon (doleva / doprava)
  - 5: Nastavení do vyšetřovací pozice
  - 6: Nastavení do resuscitační pozice (CPR)
- **Integrované ovladače:**  
Integrované ruční ovladače mohou být umístěny na vnitřní i vnější stranu zábrany

umístěné u zádového dílu. Ovladače mohou mít tvar obdélníku nebo kosočtverce, liší se však kromě tvaru obrysu už jen rozmístěním tlačítek. Funkce tlačítek je u obou typů stejná.



Obr. 4: Obě varianty integrovaného ovladače (převzato a upraveno z [8])

- 1: Polohování stehenního dílu
- 2: Polohování zádového dílu
- 3: Autokontura (společné ovládání stehenního a zádového dílu)
- 4: Nastavení do kardiackého křesla
- 5: Zdvih celé ložní plochy
- 6: Poloha pro snadné vstávání z lůžka
- 7: Aktivační tlačítko
- 8: Indikační LED dioda

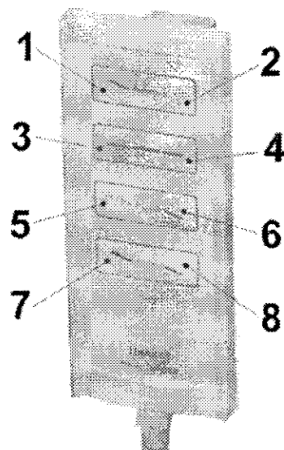
Pro zvýšení bezpečnosti musí být před polohováním lůžka nejprve stisknuto aktivační tlačítko (7). Po jeho stisknutí je možné lůžko polohovat tlačítky 1 až 6, po 10-ti sekundách nečinnosti jsou ale funkce těchto tlačítek opět uzamknuty. K indikaci, zda je panel připraven polohovat lůžko, slouží LED dioda pod aktivačním tlačítkem (8). Pokud je právě aktivní sesterský panel, je polohování integrovanými ovladači automaticky uzamčeno. Ze sesterského panelu je také možné trvale uzamknout funkce tlačítek 1 až 5 v případě, že ošetřující personál považuje některou z takto nastavitelných poloh za nevhodnou vzhledem ke zdravotnímu stavu pacienta.

- **Ruční patientský ovladač:**

Stejně jako v předchozím případě je i tento ovladač podřízen sesterskému panelu, zde je možné uzamknout všechna funkční tlačítka. Narozdíl od integrovaných ovladačů zde před polohováním lůžka není nutné panel aktivovat.

Levý sloupec tlačítek polohuje směrem vzhůru, pravý sloupec směrem k zemi.

- 1, 2: Polohování zádového dílu
- 3, 4: Zdvih celé ložní plochy
- 5, 6: Polohování stehenního dílu
- 7, 8: Autokontura



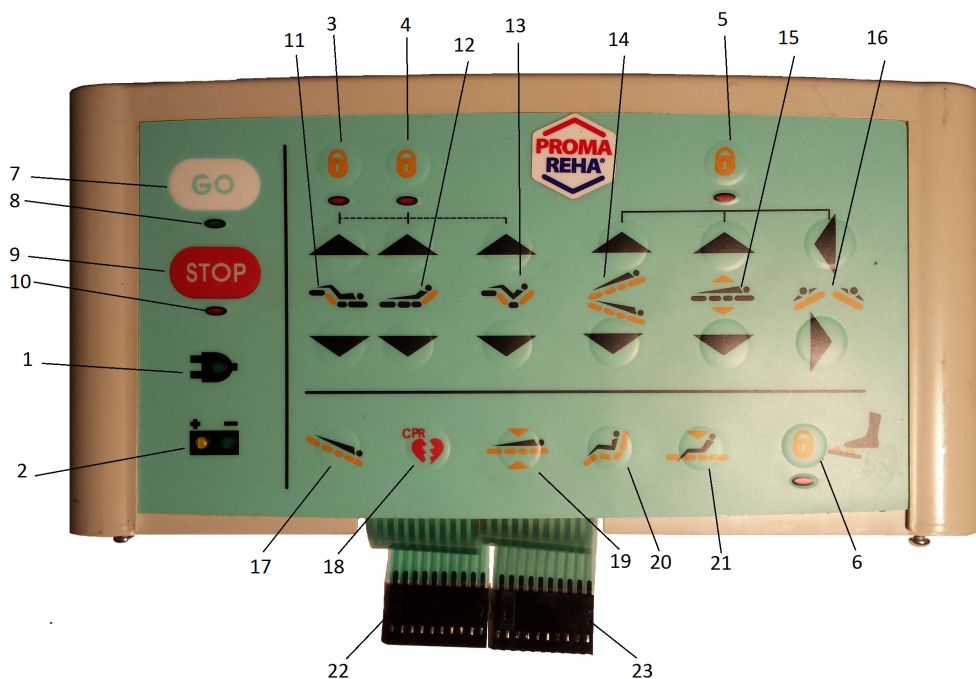
Obr. 5: Rozmístění ovládacích tlačítek na ručním patientském panelu (převzato z [8])

### 3.3 Sesterský panel

Ovladačem s největšími možnostmi nastavení polohy je sesterský panel. Pro zvýšení bezpečnosti pacienta je z něj také možné dočasně nebo trvale uzamknout většinu funkcí ostatních ovladačů.

Dočasné uzamknutí se spustí tlačítkem Stop a uzamkne všechny funkce všech ostatních panelů po dobu 30-ti sekund. Trvalé uzamčení funkcí lze nastavit stisknutím tlačítka se symbolem zámku patřícího k příslušnému nastavení polohy. Uzamčení symbolizuje stavová LED dioda pod příslušným zamykacím tlačítkem. Rozložení a význam LED diod na sesterském panelu je popsán v kapitole 3.3.1

Stejně jako u integrovaným ovladačů je pro aktivaci funkcí panelu nutné napřed stisknout aktivací tlačítko Go. Aktivace je indikována stavovou LED diodou pod tlačítkem Go, funkce se automaticky uzamknou po 15-ti sekundách nečinnosti.



Obr. 6: Rozmístění ovládacích prvků na sesterském panelu

Tabulka 3: Legenda k sesterskému panelu

č.	Popis funkce
1	Indikátor napájení lůžka
2	Dvojice diod signalizujících stav akumulátoru lůžka: - Svítící zelená dioda značí nabitý záložní zdroj - Přeblikávání zelené a oranžové diody značí vybíjení - Svítící oranžová dioda značí nízký stav záložního zdroje
3	Zámek polohování stehenního dílu
4	Zámek polohování zádového dílu
5	Zámek nastavení náklonu lůžka a zdvihu ložné plochy
6	Zámek nožního ovladače
7	Aktivační tlačítko Go
8	Indikátor aktivace funkcí sesterského panelu
9	Tlačítko Stop pro dočasné uzamčení funkcí ostatních panelů (deaktivovaný režim)
10	Indikátor deaktivovaného režimu
11	Polohování stehenního dílu
12	Polohování zádového dílu
13	Polohování stehenního a zádového dílu současně (autokontura)
14	Nastavení polohy Trendeleburg nebo Antitrendelenburg
15	Nastavení zdvihu ložné plochy
16	Nastavení laterálního náklonu (doleva nebo doprava)
17	Nastavení polohy Trendelenburg
18	Nastavení resuscitační polohy (CPR)
19	Nastavení vyšetřovací polohy
20	Nastavení polohy Kardiacké křeslo
21	Nastavení polohy pro snadné vstávání z lůžka
22	Konektor klávesnice
23	Konektor stavových LED diod

### 3.3.1 Zapojení, logika tlačítek

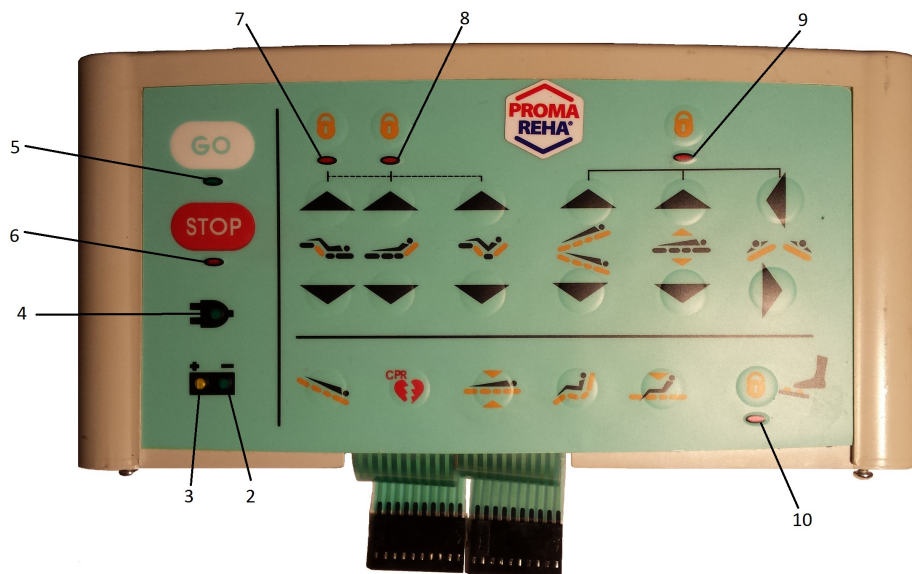
Aby bylo možné emulovat funkce sesterského panelu, musela napřed proběhnout analýza zapojení jeho tlačítek a stavových LED diod.

Zapojení LED diod je jednoduché. Protože je na panelu osazeno celkem 9 LED a připojeny jsou pomocí 10-pinového konektoru, stačí zjistit, na kterém pinu je která dioda. Desátý pin je připojen na zem. Zapojení stavových LED je rozepsáno v tab. 4 a znázorněno na Obr. 7. Číslování LED diod odpovídá pořadí pinů na vstupním konektoru.

Tabulka 4: Připojení stavových LED diod k sesterskému panelu

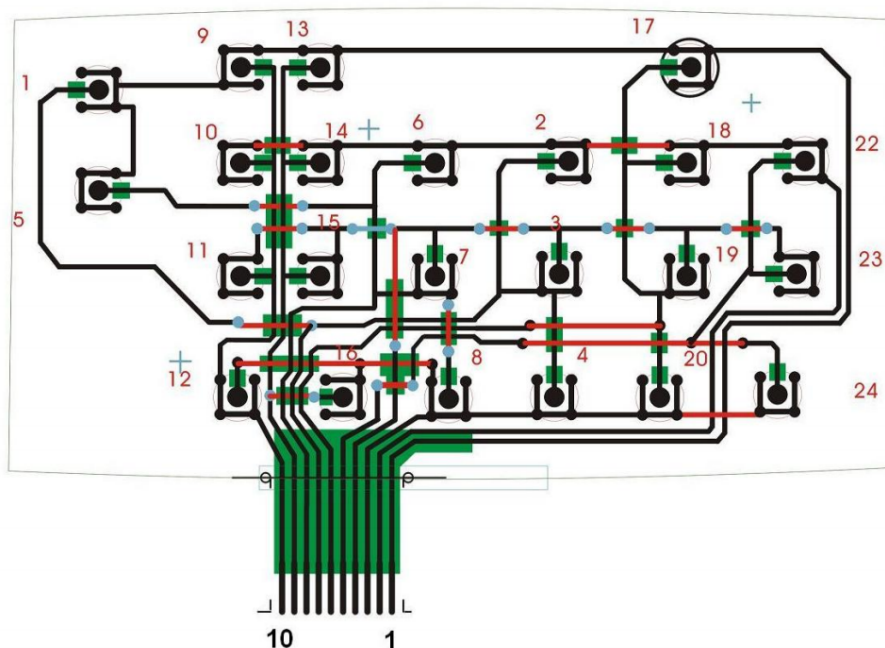
pin č.	stavová LED dioda indikující:
1	- (pin připojen na zem)
2	nabitý záložní zdroj
3	vybitý záložní zdroj
4	napájení lůžka ze sítě 230 V
5	aktivní stav
6	deaktivovaný stav
7	uzamčení polohování stehenního dílu
8	uzamčení polohování zádového dílu a autokontury
9	uzamčení polohování zdvihu a náklonu lůžka
10	uzamčení funkcí nožního panelu





Obr. 7: Rozmístění stavových LED diod na sesterském panelu

Protože je klávesnice připojena 10-pinovým konektorem a obsahuje 23 tlačítek, je zřejmé, že tlačítka nemohou být přímo připojena na samostatný pin. Jednoduchým řešením tohoto problému je zapojení tlačítek do matice a vyčítání jejich stavů jako u maticové klávesnice. Že zde toto řešení bylo použito dokládá tabulka zapojení tlačítek (tab. 5). 23 tlačítek lze zapojit do matice 5x5 nebo 6x4, zde je použita matice 6x4 (sloupce tvoří piny 1 až 6, řádky tvoří piny 7 až 10). Tabulka byla vytvořena tak, že se při stisku tlačítka hledaly ohmmetrem zkratované piny, ty jsou pak označeny jako 1, ostatní piny jsou označeny jako 0.



Obr. 8: Zapojení ovládacích tlačítek na sesterském panelu (převzato z [1])

Tabulka 5: Logika tlačítek sesterského panelu (převzato a upraveno z [1])

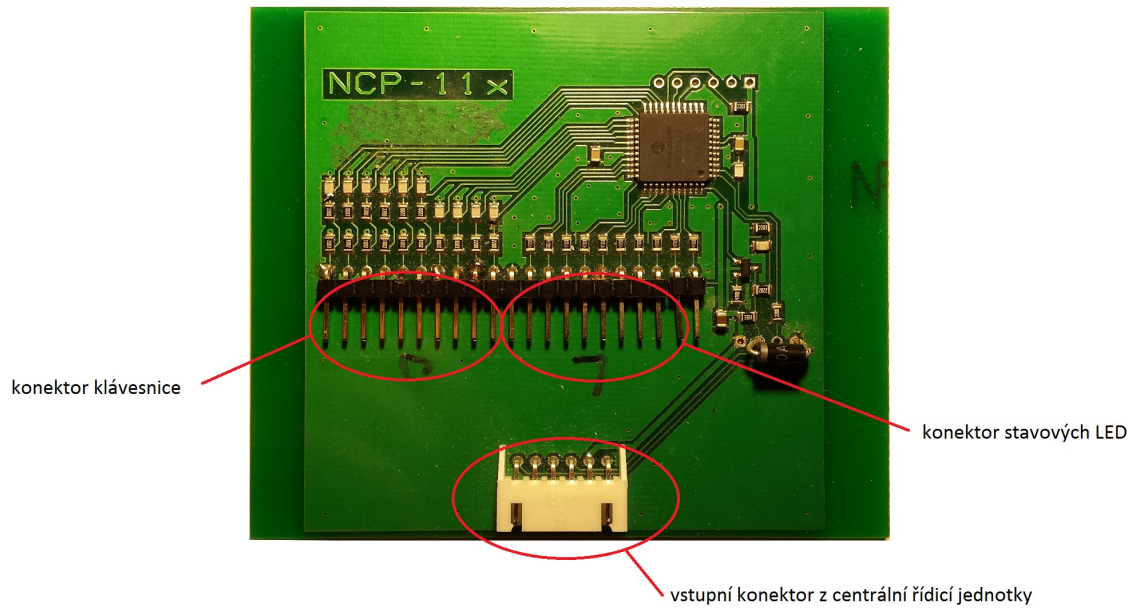
tlačítko	pin 1	pin	pin 3	pin 4	pin 5	pin 6	pin 7	pin 8	pin 9	pin10
1	1	0	0	0	0	0	1	0	0	0
2	0	1	0	0	0	0	1	0	0	0
3	0	0	0	1	0	0	1	0	0	0
4	0	0	1	0	0	0	1	0	0	0
5	1	0	0	0	0	0	0	1	0	0
6	0	1	0	0	0	0	0	1	0	0
7	0	0	0	1	0	0	0	1	0	0
8	0	0	1	0	0	0	0	1	0	0
9	1	0	0	0	0	0	0	0	0	1
10	0	1	0	0	0	0	0	0	0	1
11	0	0	0	1	0	0	0	0	0	1
12	0	0	1	0	0	0	0	0	0	1
13	1	0	0	0	0	0	0	0	1	0
14	0	1	0	0	0	0	0	0	1	0
15	0	0	0	1	0	0	0	0	1	0
16	0	0	1	0	0	0	0	0	1	0
17	1	0	0	0	0	1	0	0	0	0
18	0	1	0	0	0	1	0	0	0	0
19	0	0	0	1	0	1	0	0	0	0
20	0	0	1	0	0	1	0	0	0	0
22	0	1	0	0	1	0	0	0	0	0
23	0	0	0	1	1	0	0	0	0	0
24	0	0	1	0	1	0	0	0	0	0

### 3.3.2 Možnosti připojení k panelu

Sesterský panel je k centrální jednotce lůžka připojen pomocí konektoru se šesti vodiči, z nichž jsou využity pouze čtyři: napájení (+5V, zem) a dva datové vodiče. Konektor je konstruován tak, aby ho nebylo možné bez použití nepřiměřené síly zapojit nesprávně.

Konektory pro připojení klávesnice a stavových LED diod jsou osazeny vedle sebe ještě s jedním nezapojeným výstupem, dohromady je výstup realizován řadou 21-ti pinů na pinové liště. To může být matoucí především kvůli správnému připojení obou konektorů - fyzicky je možné je připojit i vzhůru nohama, ale toto zapojení bude nefunkční. Kvůli jednomu nadbytečnému výstupu může také dojít k posunutí jednoho nebo obou konektorů, takové zapojení opět nebude správně fungovat. Navíc na 21-pinové liště není označené, kam který 10-pinový konektor klávesnice a LED diod patří.

Řídicí jednotka sesterského panelu s popisem vstupních a výstupních konektorů je zobrazena na Obr. 9. Připojení modulu LuzkoBoard k této jednotce popisuje kapitola 6.4.

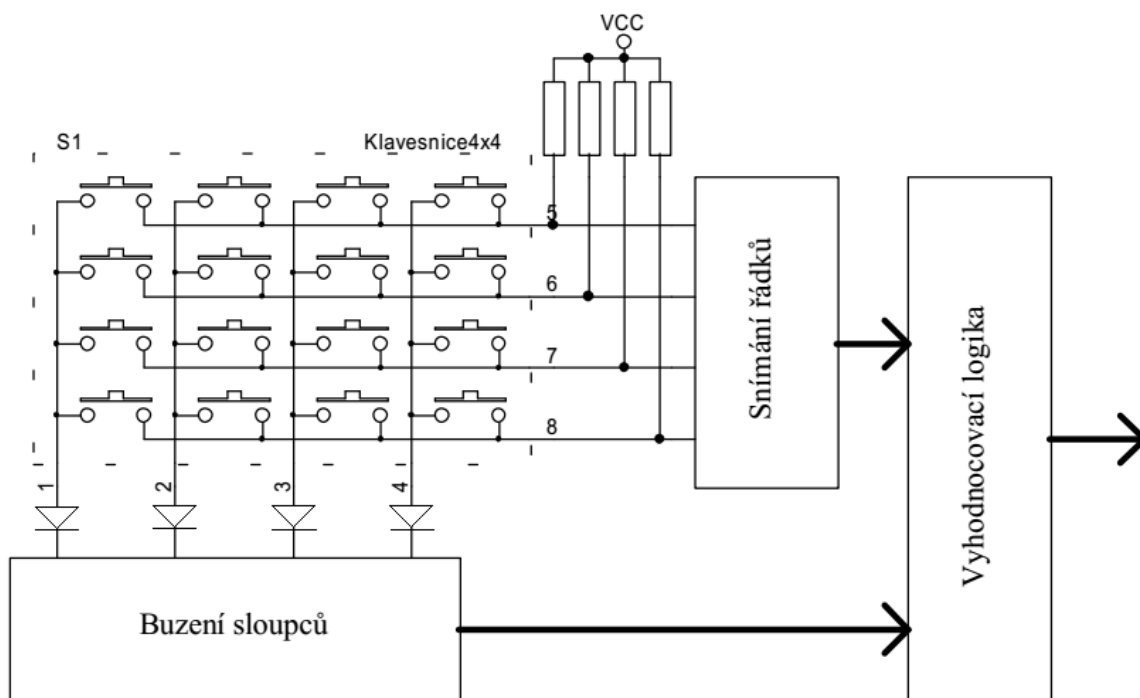


Obr. 9: Řídicí jednotka sesterského panelu s označenými konektory



## 4 Maticová klávesnice

Maticové zapojení umožňuje snímání většího počtu spínačů než kolik je k dispozici vstupů. Například 16 tlačítek lze snímat v matici 4x4 a je k tomu zapotřebí 8 vodičů (4 pro snímání řádků a 4 pro buzení sloupců). Schéma takového zapojení je na Obr. 10. V obvodu pro buzení sloupců jsou diody, které chrání obvod před zkratem v případě, že bude současně stisknuto více tlačítek na stejném řádku. Pak totiž dojde ke zkratu všech sloupcových vodičů, ke kterým jsou právě stisknutá tlačítka připojena. [10]



Obr. 10: Schéma zapojení pro snímání maticové klávesnice (převzato a upraveno z [10])

### 4.1 Snímání maticové klávesnice (stisk jediného tlačítka)

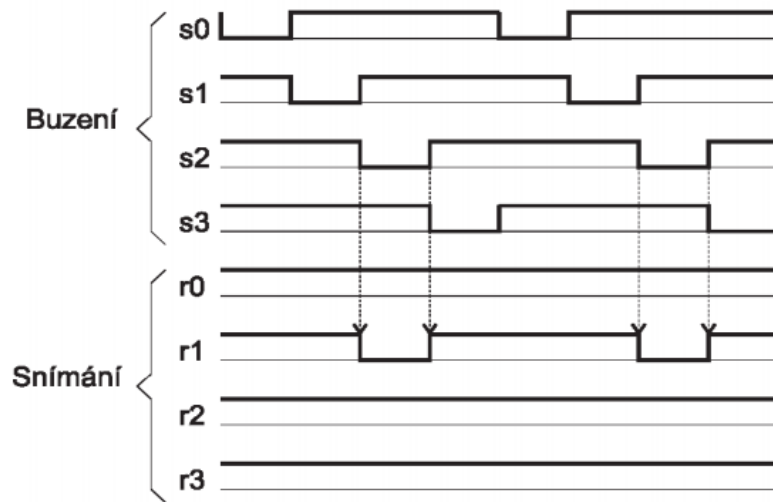
Ve výchozím stavu jsou všechny snímací vodiče (pro každý řádek) udržovány v klidovém stavu (log. 1) pomocí rezistorů vyvedených na napájecí napětí. Pomocí obvodu pro buzení sloupců je pak na ně postupně v definovaných časových intervalech přiváděna log. 0, na každý sloupec zvlášť. Pokud je některá z kláves stisknuta, budicí signál sloupce se objeví na příslušném řádkovém vodiči a je vyhodnocen obvodem pro snímání řádků.

Souřadnice tlačítka (X,Y) se pak vyhodnocují podle toho, na kterém řádkovém vodiči je přijat puls (souřadnice Y) a ve kterém časovém úseku je tento puls přijat (souřadnice X). Na Obr. 11 je zobrazen časový průběh signálu na jednotlivých vodičích se stisknutým tlačítkem na řádku  $r1$  a sloupci  $s2$ .

Snímání více stisknutých tlačítek zároveň je o trochu složitější. Dvě tlačítka současně lze vyhodnotit stejně, jako v případě jediného, ale problém může nastat při stisku tří nebo více tlačítek současně. Pokud totiž tato tlačítka tvoří v matici pravoúhlý trojúhelník, bude současně jako stisknuté vyhodnoceno i tlačítko na souřadnici, doplňující v matici tento trojúhelník na obdélník. Tento jev je způsoben tím, že stisknutá tlačítka vzájemně propojí všechny své řádkové a sloupcové vodiče. Při tomto zapojení a způsobu vyhodnocení nelze zjistit, zda je

nebo není stisknuté i čtvrté tlačítko na posledním vrcholu obdélníku. Maticová klávesnice se proto nehodí pro vyhodnocování více stisknutých tlačítek současně.

Realizace úlohy stisk více tlačítek současně neumožňuje, tento problém tedy není nutné řešit.



Obr. 11: Časový průběh signálů s jedním stisknutým tlačítkem (převzato z [10])

## 5 Řešení zadání

K ovládání lůžka byla vytvořena aplikace Lůžko pro OS Android. Tato aplikace komunikuje pomocí rozhraní Bluetooth s modulem RN42, ten dále komunikuje s řídicím mikrokontrolérem STM32L053 pomocí rozhraní UART. BT modul je stejně jako mikrokontrolér osazen na DPS vloženém do sesterského panelu.

Mikrokontrolér přijímá příkazy z aplikace a podle nich nastavuje příslušné výstupy pomocí dvou analogových multiplexerů 74HCT4051, čímž imituje stisk tlačítka na sesterském panelu. Dále mikrokontrolér sleduje stavové LED diody a odesílá jejich stav zpět aplikaci, která ve svém layoutu animuje rozsvícení/zhasnutí LED diody na sesterském panelu.

Celkový odběr modulu LuzkoBoard, vloženého do sesterského panelu, je přibližně 30 mA. Jednotlivým částem realizovaného řešení jsou věnovány kapitoly 6 až 9.



*Obr. 12: Realizace modulu LuzkoBoard a mobilní aplikace Lůžko*





## 6 Návrh modulu LuzkoBoard

DPS je navržen tak, aby ho bylo možné zapojit mezi řídicí jednotku v sesterském panelu a její výstupy, ke kterým je připojen konektor klávesnice a stavových LED diod. Požadavkem bylo, aby se DPS vešel do prostoru uvnitř sesterského panelu a aby pro jeho připojení nebylo nutné do zde uložené řídicí jednotky jakkoliv zasahovat.

Řešením tedy bylo navrhnout modul LuzkoBoard tak, aby ho bylo možné nasadit na výstupní konektory řídicí jednotky, ke kterým je v originálním zapojení připojena klávesnice a stavové LED diody. Výstup klávesnice a vstupy pro LED diody na panelu jsou pak zapojeny do výstupů modulu, tan tak tvoří jakýsi "mezikus", aniž by omezil funkce samotného panelu. Zapojení do sesterského panelu popisuje kapitola 6.4.

Dále bylo třeba vyřešit napájení modulu. Řídicí jednotka panelu nemá žádný výstup, kde by bylo možné vyvést napájení bez zásahu do jejího HW - připájení napájecího konektoru by odporovalo zadání, protože by došlo k fyzickému zásahu do řídicí jednotky panelu. Jediným řešením v souladu se zadáním je tedy přivést napájení ze vstupního konektoru sesterského panelu. Sem jsou přivedeny z centrální řídicí jednotky lůžka tyto vodiče:

- GND
- +5V
- Data –
- Data +

Odsud tedy lze vyvést napájení, pokud bude do konektoru vložena redukce. Bohužel, nepodařilo se zajistit originální konektor použitý pro připojení panelu, proto bylo nutné improvizovat. Rozteč mezi vývody má naštěstí standardní rozměr 0.1" (přibližně 2.54 mm), což je rozměr běžně používaný například u dutinkových lišt, proto bylo možné použít standardní dutinkovou lištu na jedné straně a pinovou lištu na druhé straně vstupního konektoru. Obě části redukce jsou vzájemně připájené a navíc je z nich vyvedeno napájení (tedy vodiče +5V a GND) pro modul.

Samozřejmě, lepším řešením by bylo vyčítat ze vstupního konektoru i data (z datových vodičů), ale protože nebyla k dispozici specifikace komunikačního protokolu mezi centrální jednotkou lůžka a řídicí jednotkou v sesterském panelu, bylo by velice obtížné takovéto řešení realizovat. Vyžadovalo by zachytávání komunikace mezi sesterským panelem a řídicí jednotkou a její následné vyhodnocování, což by bylo obtížné. Navíc nelze zaručit, že zde není ochrana před připojením cizího panelu např. ověřením výrobního čísla, což by pak vedlo ke komplikacím při připojení modulu k jinému lůžku, než na kterém byl modul testován.

### 6.1 Použité součástky

#### Bluetooth modul RN42

Pro realizaci komunikace mezi modulem LuzkoBoard a řídicí aplikací jsem se rozhodl využít rozhraní Bluetooth, konkrétně je zde použit Bluetooth modul RN42. Hlavním důvodem pro použití tohoto modulu bylo, že s jeho ovládáním mám již zkušenosti z bakalářské práce, dále pak jednoduchost jeho zapojení a ovládání. Modul RN42 slouží na DPS k přeposílání komunikace mezi sériovou linkou UART (BT modul ↔ mikrokontrolér) a Bluetooth komunikací (aplikace Lůžko ↔ BT modul). Realizace a popis komunikace jsou popsány v kapitole 8. Následuje tabulka vybraných parametrů modulu RN42:

Tabulka 6: Vybrané parametry Bluetooth modulu RN42 [11]

Parametr	Hodnota
Pracovní napětí (VDD)	3.0 ~ 3.6 V
Pracovní teplota	-40 ~ 85 deg C
Citlivost při 0.1% BER <sup>(1)</sup>	-80 dBm
Vysílací výkon	4 dBm
Vstupní hodnota pro log.0	-0.4 ~ 0.8 V
Vstupní hodnota pro log.1	(0.7 * VDD) ~ (VDD + 0.4 V)
Výstupní hodnota pro log.0	0.2 V
Výstupní hodnota pro log.1	(VDD - 0.2) V

<sup>(1)</sup>Bit Error Ratio, poměr počtu chybně přijatých bitů ku celkovému počtu přijatých bitů

### Mikrokontrolér STM32L053

Pro řízení modulu LuzkoBoard jsem vybral mikrokontrolér ARM STM32L053 s jádrem ARM Cortex-M0+ v pouzdře LQFP48. Splňuje všechny následující požadavky:

- Dostatečný výpočetní výkon
- Komunikace pomocí rozhraní UART
- Alespoň 14 I/O:
  - 5 výstupů pro ovládání analogových multiplexerů (emulace maticové klávesnice)
  - 9 vstupů pro vyčítání stavových LED
- Vlastní zdroj hodin
- Nízká cena

Výhodou použití mikrokontroléru z rodiny STM32 je zdarma dostupné profesionální vývojové prostředí včetně debugování, a také snadná dostupnost k velkému množství dokumentace, návodů a vzorových projektů.

### Analogový multiplexer 74HCT4051

Pro emulaci maticové klávesnice jsou využity dva analogové multiplexery typu 4051 s využitím nízko-napěťové logiky TTL. Jejich funkcí je spínání analogových vstupů/výstupů s jedním běžným (*common*) vstupem/výstupem na základě signálů na digitálních vstupech. Typ HCT je použit z toho důvodu, že digitální vstupy analogových multiplexerů typu 4051(D) mají rozhodovací úroveň log.1 nastavené pro 5V řízení, ale mikrokontrolér a BT modul pracují s napájením 3.3V. Typ HCT je realizován pomocí TTL logiky a proto je rozhodovací úroveň pro log.1 snížena na přibližně 3V.

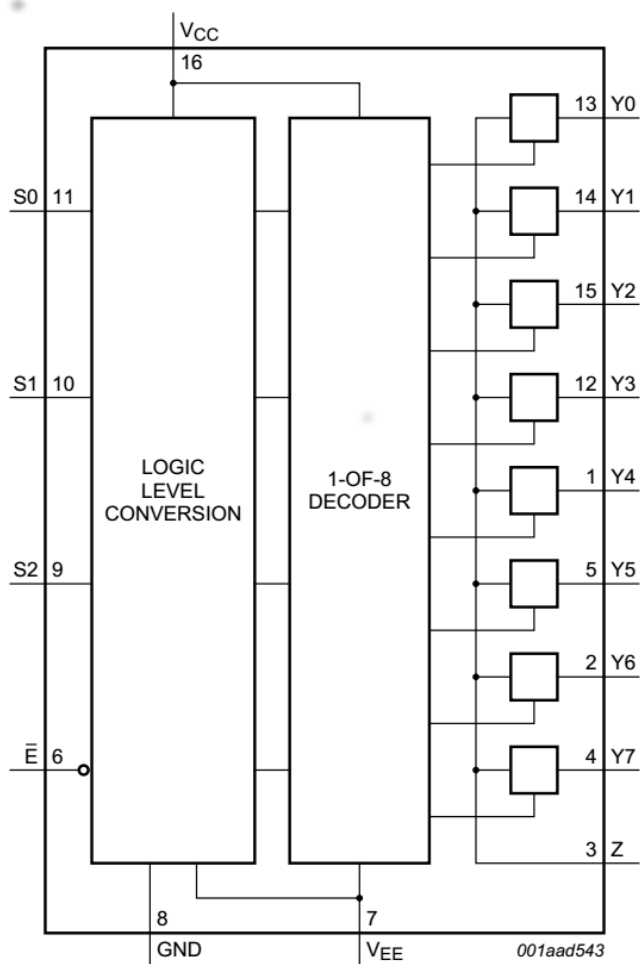
Jedná se o multiplexery typu 8:1, což znamená, že 1 běžný (*common*) vstup/výstup (**Z**) je spojen s jedním z 8 analogových vstupů/výstupů (**Y0~Y7**). Ostatní analogové vstupy/výstupy zůstávají ve stavu vysoké impedance. Volba analogového vstupu je dána signály na 3 digitálních vstupech (**S0~S2**), viz tabulka 7<sup>5</sup>.

<sup>5</sup> **0** = log.0  
**1** = log.1  
**X** = nemá význam

Tabulka 7: Funkční tabulka analogových multiplexerů 74HCT4051, převzato z [12]

Vstupy				Analogový vstup/výstup
$\bar{E}$	S2	S1	S0	
0	0	0	0	Y0
0	0	0	1	Y1
0	0	1	0	Y2
0	0	1	1	Y3
0	1	0	0	Y4
0	1	0	1	Y5
0	1	1	0	Y6
0	1	1	1	Y7
1	X	X	X	-

Jak lze vyčíst z funkční tabulky, propojení je nejprve nutné povolit nastavením vstupu *enable input* ( $\bar{E}$ ) do log.0. Pokud je vstup ( $\bar{E}$ ) v log.1, pak budou všechny analogové vstupy/výstupy ( $Y0 \sim Y7$ ) nastaveny do stavu vysoké impedance bez ohledu na stav digitálních vstupů ( $S0 \sim S2$ ). Funkce analogového multiplexeru je zřejmá z Obr. 13, na kterém je znázorněn funkční diagram.



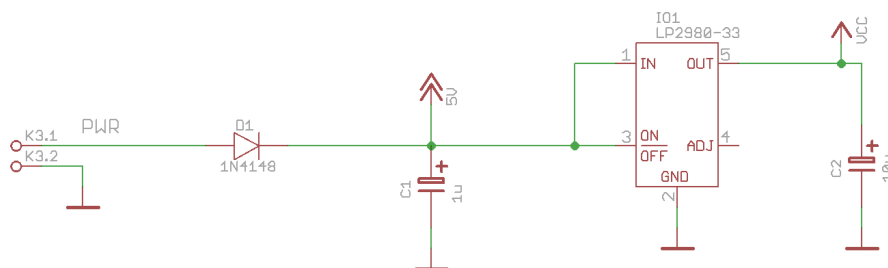
Obr. 13: Funkční diagram analogového multiplexeru 8:1, převzato z [12]

## 6.2 Funkční části zapojení

### Napájecí zdroj 3.3 V

Na DPS je ze vstupního konektoru sesterského panelu přivedeno napájení 5 V, ale Bluetooth modul RN42 a řídicí mikrokontrolér pracují s napájecím napětím 3.3 V.

Zapojení je realizováno pomocí lineárního LDO<sup>6</sup> stabilizátoru napětí LP2980, viz Obr. 14, hodnoty kondenzátorů **C1** a **C2** byly zvoleny podle doporučení v datasheetu [13]. Lineární LDO stabilizátor mohl být díky malému rozdílu vstupního a výstupního napětí (5 V a 3.3 V), výhodnou použítého stabilizátoru je nízký úbytek napětí.



Obr. 14: Schéma zapojení napájecího zdroje 3.3 V

### Zapojení modulu RN42

Modul je podle doporučení v datasheetu osazen na DPS tak, aby jeho anténa přesahovala její obrys - tak je dosaženo vyššího zisku antény.

Jak již bylo zmíněno v kapitole 6.1, BT modul slouží k přeposílání komunikace z rozhraní Bluetooth na rozhraní UART. Zapojení je velice jednoduché, modul je napájený 3.3 V, vstup **UART\_RX** a výstup **UART\_TX** jsou připojeny k odpovídajícímu vstupu a výstupu mikrokontroléru. Dále už jsou k modulu připojeny jen dvě stavové LED diody.

Zelená LED, připojená k výstupu **GPIO5**, je standardní stavová LED dioda modulu [11]. Výstup **GPIO5** totiž v základním nastavení modulu slouží jako status a umí signalizovat tři různé stavy modulu:

- LED dioda bliká s intervalem 1s, pokud je modul napájený a ve standardním režimu
- LED dioda bliká s intervalem 0.1s, pokud je modul v příkazovém režimu (*command mode*)
- LED dioda svítí, pokud je modul ve standardním režimu a připojený k jinému zařízení pomocí Bluetooth

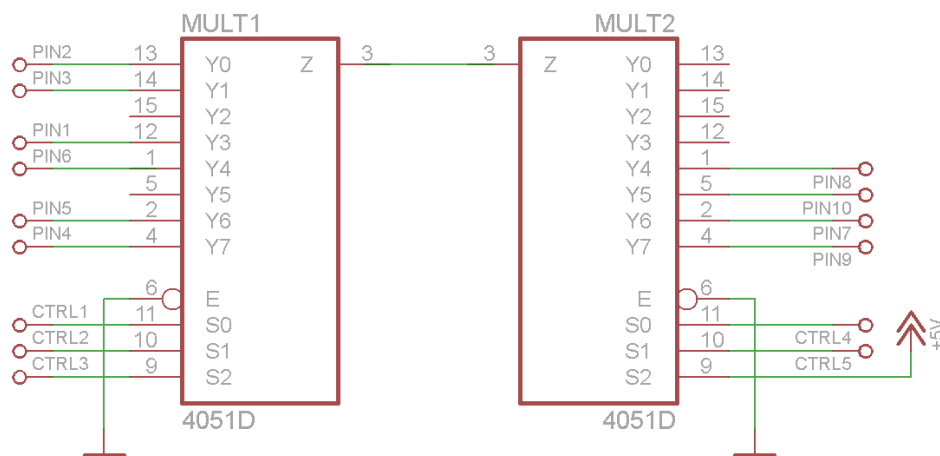
Červená dioda, připojená k výstupu **GPIO3**, slouží k signalizaci chyby v komunikaci, tedy pokud je navázáno spojení mezi modulem a řídicí aplikací, ale mikrokontrolér nevysílá žádná data.

### Emulace maticové klávesnice

Na DPS jsou použity dva multiplexery tak, že mají vzájemně propojený *common* vstup/výstup (**Z**). První multiplexer tak vybírá řádek a druhý slopec maticové klávesnice.

<sup>6</sup>Low-dropout, umožňuje konverzi stejnosměrného napětí s relativně malým rozdílem napájecího a výstupního napětí

Pomocí nastavení digitálních vstupů (**S0~S2**) obou multiplexerů je pak možné propojit jeden analogový vstup/výstup (**Y0~Y7**) na prvním multiplexeru s analogovým vstupem/výstupem na druhém multiplexeru a tak napodobit stisknutí klávesy na sesterském panelu lůžka. Jak lze vidět na schématu zapojení (Obr. 15), jeden z digitálních vstupů druhého multiplexeru je trvale připojený na napájecí napětí, tedy je v log.1. Je to proto, že tento multiplexer přepíná jen mezi čtyřmi analogovými vstupy/výstupy, proto k jeho řízení stačí dva digitální vstupy (viz tab. 7). **CTRL1~CTRL5** jsou výstupy z mikroprocesoru a **PIN1~PIN10** jsou jednotlivé vstupy z 10-pinového konektoru klávesnice.



Obr. 15: Schéma zapojení analogových multiplexerů

Pokud není stisknuté žádné tlačítko, nesmí být žádná dvojice analogových vstupů/výstupů multiplexerů propojena. Toho lze docílit tím, že na prvním multiplexeru nastavíme analogový vstup/výstup **Y2** nebo **Y5**. Takto musí být první multiplexer nastaven po inicializaci řídicího programu v mikrokontroléru a také v případě výpadku komunikace s aplikací Lůžko.

### Vyčítání stavových LED

Vyčítání stavů LED diod sesterského panelu je realizováno tak, že jednotlivé piny jsou přivedeny do vstupů mikroprocesoru, jeden pin na jeden vstup. Před připojením ke vstupu mikroprocesoru je osazen ještě ochranný rezistor s hodnotou 10kΩ, který slouží jako ochrana před přepětím (vstupy mají vlastní ochranné diody).

Tabulka 8: Parametry výroby DPS technologií Pool Service u firmy Pragoboard s.r.o. [14]

Parametr	Hodnota
Materiál <sup>(1)</sup>	FR4, 1.5mm, 18µm
Nepájivá maska (zelená)	1-2x
Servisní potisk (bílý)	1-2x
Pokovení	bezolovnatý HAL nebo chemické zlacení
Frézování vnějšího obrysu	obdélníkové, fréza 2.4mm
Velikost spoje/mezery	>= 150µm
Průměr vrtání	>= 0.3mm

<sup>(1)</sup>Platí pro 2-vrstvé DPS

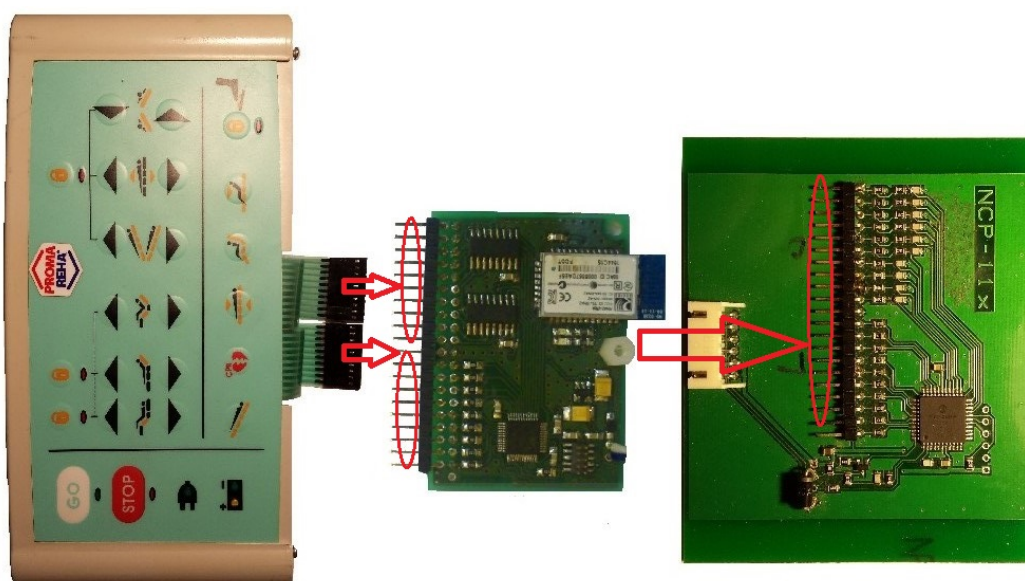
### 6.3 Výroba DPS

Výroba DPS byla realizována firmou Progoboard s.r.o. v programu Pool service. Tento program je připraven pro jednorázovou výrobu řádově jednotek kusů DPS o celkové ploše do 1 dm<sup>2</sup> za nižší cenu než při standardním postupu. Tento program má ale kromě celkové plochy DPS i další omezení, pro snížení ceny není možné měnit parametry výroby popsané v tabulce 8.

### 6.4 Připojení modulu LuzkoBoard do sesterského panelu

Vložení modulu do sesterského panelu probíhá v několika krocích. Kroky 2 a 3 jsou znázorněny na Obr. 16

1. Odpojit dvojici 10-pinových konektorů klávesnice a stavových LED od řídicí jednotky.
2. Nasunout 20-pinový konektor na řídicí jednotce v sesterském panelu do dutinkové lišty na DPS.
3. Připojit 10-pinovou dutinkovou lištu klávesnice a 10-pinovou dutinkovou lištu stavových LED diod do 20-pinového konektoru na DPS. Zde je nutné dbát na správnou orientaci obou konektorů.
4. Připojit vstupní konektor z centrální řídicí jednotky lůžka do napájecího konektoru DPS.
5. Nasunout napájecí konektor DPS (6-pinová dutinková lišta) spojený se vstupním konektorem z centrální řídicí jednotky lůžka do vstupního konektoru řídicí jednotky panelu.
6. Nyní lze připojit sesterský panel k lůžku.



Obr. 16: Připojení modulu LuzkoBoard k sesterskému panelu a jeho řídicí jednotce

## 7 Mobilní aplikace Lůžko

Aplikace Lůžko je navržena především pro snadné ovládání. Jednoduchost je samozřejmě základní požadavek, aplikace je určena i pro osoby se zhoršenou motorikou. Dalším požadavkem je spolehlivost. Ta je zvýšena přímým připojením k Bluetooth modulu bez vyhledávání všech zařízení v dosahu (více v kapitole 7.4) a především použitím vhodného a dostatečně jednoduchého komunikačního protokolu (viz kapitola 8).

### 7.1 Kompatibilita

Aplikaci Lůžko lze nainstalovat a spustit na libovolném zařízení s OS Android verze 2.3.3 (API 9) nebo vyšší, které má dotykový displej a Bluetooth adaptér. Aplikace využívá i vibrace pro uživatelskou zpětnou vazbu, ale vibrační jednotku na zařízení nutně nevyžaduje.

Podpora starších verzí OS Android již dnes nemá praktický význam, podíl aktivních zařízení s verzí starší než Android 2.3.3 je zanedbatelný [15].

Pro ověření kompatibility s různými verzemi OS Android byla aplikace Lůžko testována byla na následujících zařízeních:

Tabulka 9: Seznam testovacích zařízení

Název	Verze OS	Displej	Rozlišení	Vibrace
Samsung Galaxy S3 Mini (GT-I8191)	5.1.1*	4"	800x480 px	ano
Sony Ericsson Neo V (MT11i)	2.3.4	3.7"	854x480 px	ano
Sencor Element 9.7V3	4.1.0	9.65"	1024x768 px	ne
Alcatel OneTouchStar (6010D)	4.1.1	4"	800x480 px	ano
Samsung Galaxy S4 (GT-I9505)	5.0.1	5"	1920x1080 px	ano

\* alternativní ROM: CyanogenMod 12.1, který vychází z Adnroidu 5.1.1

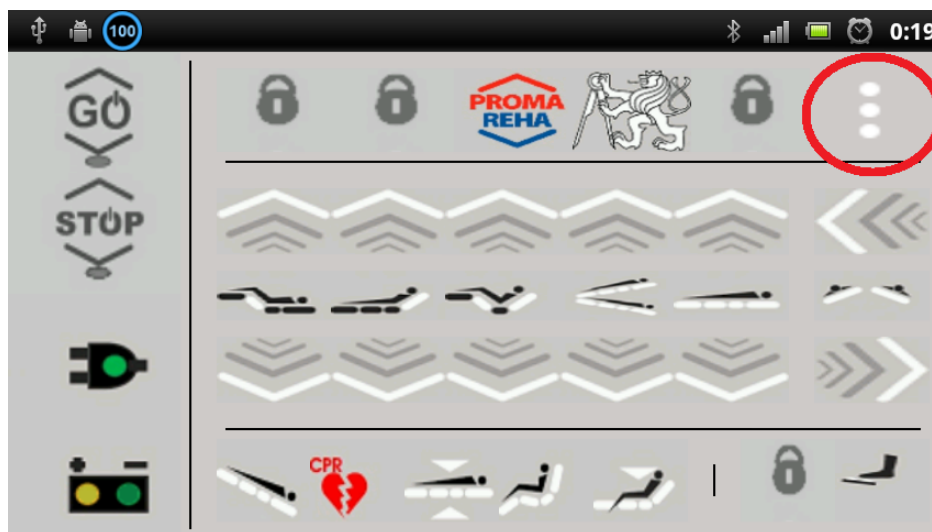
Kvůli pohodlnějšímu ovládání a pro zachování vzhledu podobnému sesterskému panelu je aplikace vždy orientována na šířku displeje (*landscape mode*), zobrazení na výšku (*portrait mode*) je v nastavení aplikace zakázáno.

Aplikace nabízí podporu českého nebo anglického jazyka s možností výběru. Standardem se dnes pro větší pohodlí uživatelů stává, že aplikace při svém startu zjistí výchozí nastavení jazyka operačního systému a sama si pak nastaví buď tento jazyk, nebo výchozí (zpravidla angličtina) v případě, že nastavený jazyk operačního systému sama nepodporuje. Aplikace Lůžko má ve výchozím nastavení angličtinu a volbu jiného jazyka (čeština) ponechává na uživateli.

Při ovládání lůžka mobilním telefonem je žádoucí maximálně využít omezenou velikost displeje. Proto je porušeno doporučení [16] pro vývoj aplikací pro OS Android a v aplikaci Lůžko je potlačeno zobrazení *Action Bar*<sup>7</sup>. Takto lze sice využít i tu část displeje, kterou by běžně *Action Bar* zabírala, ale nevýhodou tohoto řešení je, že standardně je v *Action Bar* zobrazeno tlačítko Menu. Může se tedy stát, že nebude zajištěn alternativní způsob zobrazení tlačítka pro vyvolání možností aplikace - podpora alternativního zobrazení Menu tlačítka se různí s verzemi OS Android, v některých verzích dokonce chybí úplně. Bylo tedy třeba obejít standardní postup a vyvolat menu pomocí tlačítka umístěného přímo v právě zobrazovaném *layoutu* aplikace. Toto tlačítko se nachází v pravém horním rohu (viz obrázek 17) a vyvolá spuštění *Activity*, která se stará o zobrazení nastavení aplikace. Toto řešení sice

<sup>7</sup>Action bar je nástrojová lišta přidávaná ve verzi 4.0. Slouží k zobrazení funkčních tlačítek, včetně tlačítka Menu

neodpovídá standardu, ale na druhou stranu funguje napříč všemi podporovanými verzemi OS Android a tlačítko Menu bude v aplikaci Lůžko zobrazeno na stejném místě v libovolném zařízení s OS Android verze 2.3.3 a vyšší.



Obr. 17: Zobrazení Menu tlačítka v layoutu aplikace Lůžko

## 7.2 Inicializace a start aplikace

Před prvním spuštěním aplikace musí proběhnout spárování použitého zařízení s Bluetooth modulem RN42.

Možný postup pro spárování:

1. Bluetooth modul RN42 musí být napájený a spuštěný v běžném módu (výchozí stav po zapnutí)
2. Bluetooth modul RN42 musí být v dosahu Bluetooth mobilního zařízení (řádově jednotky metrů při běžných vysílacích výkonech [18])
3. Funkce Bluetooth na mobilním zařízení musí být zapnuta
4. Viditelnost pro ostatní Bluetooth zařízení v dosahu musí být povolena (*discovery mode*)
5. Nyní je třeba zvolit modul RN42 ze seznamu viditelných zařízení a potvrdit párování
6. V závislosti na verzi OS Android se může objevit hláška, zda na zařízení, se kterým probíhá párování, je nastaven stejný kód, který se zobrazuje uživateli. Kdyby uživatel omylem zvolil nesprávné zařízení ze seznamu viditelných zařízení, tímto způsobem lze chybu odhalit. Modul RN42 má z výroby přednastavený kód **1234**, který je sice možné později změnit, ale ne dříve, než dojde ke spárování. Pro úspěšné spárování zařízení nevadí, že se kódy neshodují, ale v takovém případě je důrazně doporučeno znovu zkontrolovat, že probíhá párování se správným zařízením.

Po prvním spuštění se bude aplikace snažit připojit k Bluetooth modulu s Bluetooth MAC adresou **00:06:66:7D:AE:5F**, což je výchozí hodnota v nastavení aplikace. Hodnotu Bluetooth MAC adresy modulu je možné později změnit v nastavení aplikace.

Bluetooth MAC adresu aktivního modulu lze v některých verzích OS Android zjistit přímo v seznamu viditelných zařízení, případně lze pro zjištění Bluetooth MAC adresy použít



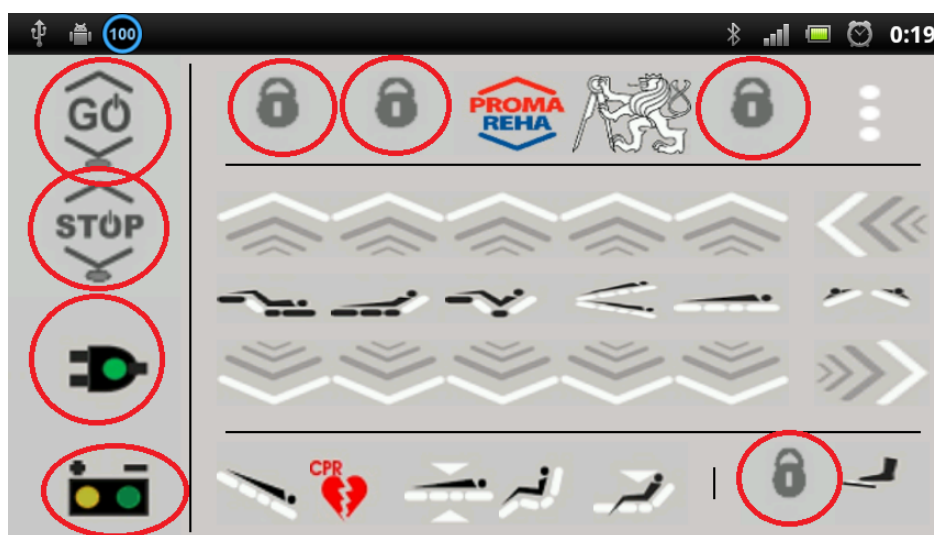
např. aplikaci *Bluetooth Address Finder*, která je zdarma dostupná v internetovém obchodě GooglePlay [20] Po úspěšném spárování mobilního zařízení s modulem a nastavení příslušné MAC adresy dojde po startu aplikace k automatickému připojení a navázání komunikace. Při příštím spuštění aplikace již bude MAC adresa modulu uložena v paměti aplikace a k navázání komunikace dojde automaticky. Bluetooth MAC adresu modulu je možné kdykoliv později v nastavení aplikace změnit.

Pokud nedojde k úspěšnému navázání komunikace, aplikace vypíše varovnou hlášku, ale i přesto se spustí. Je to z toho důvodu, že teprve z běžící aplikace má uživatel přístup do menu, kde může změnit Bluetooth MAC adresu modulu. Nicméně, aplikace je nyní spuštěna bez funkčního připojení k modulu, a proto je po každém stisknutí libovolného tlačítka znovu zobrazena varovná hláška, že nedošlo k úspěšnému navázání komunikace.

### 7.3 Běh aplikace

Po spuštění aplikace a úspěšném navázání komunikace s Bluetooth modulem je uživateli zobrazen *layout*, který je navržen tak, aby vzhledem co nejvíce připomínal sesterský panel. Jsou na něm stejně rozložena tlačítka se stejnou funkcí, jsou na něm animovány stavové LED diody ze sesterského panelu.

Stav stavových LED diod je vyčítán z příslušných vstupů na DPS v sesterském panelu a po sérové lince odeslán do Bluetooth modulu RN42, odkud je dále předán aplikaci Lůžko. Ta pak reaguje na změny stavů jednotlivých diod. Na obrázku 18 jsou zvýrazněna tlačítka a symboly, které reagují na změny stavových LED. Jsou to tlačítka **Stop** a **Go**, která indikují, zda je ovládání lůžka aktivní, dále čtyři tlačítka se symbolem zámku, která uzamykají příslušnou část ovládání lůžka, a nakonec symboly **Napájení** a **Baterie**, které indikují zda je lůžko napájeno a stav baterie.



Obr. 18: Tlačítka a symboly reflektující stav LED diod.

Při stisku některého z tlačítek je, tentokrát v opačném směru komunikace, v intervalu 100ms odeslán příkaz do mikroprocesoru, který nastaví příslušné výstupy a pomocí dvou analogových multiplexerů 8:1 propojí dva ze vstupů, čímž emuluje stisk tlačítka na sesterském panelu (emulace maticové klávesnice detailněji popsána v kapitole 4).

Pokud není požadované nastavení lůžka povoleno (indikace stavovou LED u symbolu zámku příslušejícímu k danému tlačítku), aplikace přesto emuluje stisknutí tlačítka.

Napodobuje tím chování sesterského panelu, kde uživatel může stisknout i tlačítko, které je uzamčeno. Důvodem je, že aplikace nedostává jinou zpětnou vazbu z řídicí jednotky lůžka, než jen informační stav LED diod. Proto aplikace ponechává vyhodnocení stisknutí tlačítka na řídicí jednotce lůžka.

Pro kvalitnější zpětnou vazbu k uživateli je stisk tlačítka doprovázen krátkým zavibrováním (200 ms při stisknutí a 100 ms při uvolnění). Dále je tlačítko během stisknutí animováno, pozadí tlačítka změní barvu na oranžovou, a po uvolnění tlačítka se pozadí během 2 s postupně změní zpět na výchozí barvu.

## 7.4 Komunikace s Bluetooth modulem RN42

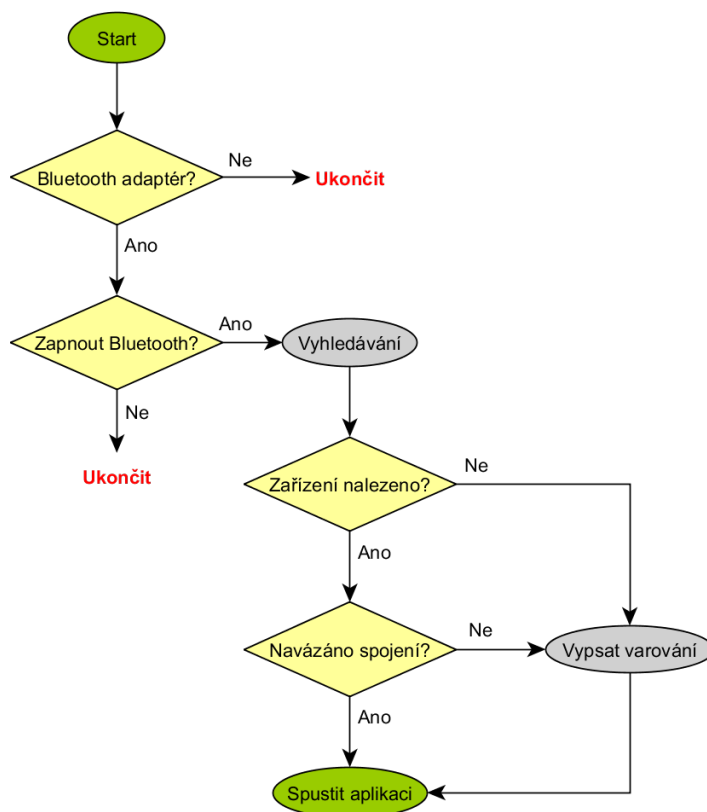
Komunikace s modulem se řídí komunikačním protokolem, viz kapitola 8.

Doporučený postup [17] pro navázání spojení pomocí Bluetooth je vyhledat všechna zařízení v dosahu, zobrazit jejich seznam a nechat uživatele, aby vybral příslušné zařízení ze seznamu. Tento postup je ale pro aplikaci Lůžko nevhodný hned ze dvou důvodů:

- Vyhledávání zařízení trvá řádově okolo 10 sekund [19].
- Je nutná interakce uživatele při výběru správného zařízení.

Navíc může uživatel nechtěně zvolit jiné zařízení ze seznamu všech viditelných zařízení v dosahu, což by zapříčinilo dočasnou nefunkčnost aplikace.

Z těchto důvodů se aplikace připojuje přímo a pouze ke známé Bluetooth MAC adrese modulu. Velikou výhodou tohoto řešení je rychlejší navázání spojení (při testování většinou do 5 s) a také jednoduchost, protože k navázání spojení může po spuštění aplikace takto dojít automaticky. Postup navazování komunikace je popsán na obrázku 19.

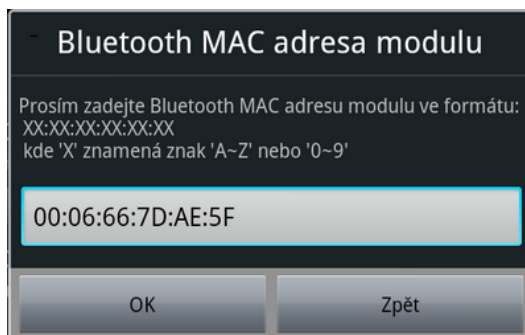


Obr. 19: Diagram zobrazující navazování BT komunikace, převzato a volně přeloženo z [21]

## 7.5 Možnosti aplikace

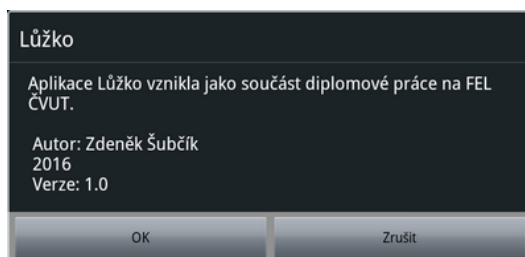
Aplikace Lůžko nabízí v Menu čtyři možnosti:

- **Jazyk:** Aplikace nabízí možnost výběru mezi anglickým a českým jazykem. Výchozím jazykem je angličtina, možnost podpory dalších jazyků je popsána v kapitole 7.6
- **Nastavení Bluetooth MAC adresy modulu:** Možnost nastavení tohoto parametru byla do aplikace přidána pro případ, že bude realizováno více DPS, na nichž budou mít jednotlivé Bluetooth moduly rozdílné MAC adresy. Takto je možné přímo v aplikaci změnit modul a není nutný zásah do zdrojových kódů.



Obr. 20: Dialog pro změnu Bluetooth MAC adresy modulu

- **Zobrazení nápovědy:** Po stisknutí zobrazí dialog s postupem pro ovládání lůžka.
- **Informace o aplikaci:** Po stisknutí zobrazí dialog s informacemi o aplikaci (verze, autor, ...).



Obr. 21: Dialog s informacemi o aplikaci

## 7.6 Podpora dalších jazyků

Aplikace podporuje zobrazení dvou jazyků, angličtiny a češtiny. Protože už je v kódu nastavení pro změnu lokalizace hotové, je možné přidat podporu dalších jazyků pomocí několika jednoduchých úprav ve zdrojovém kódu aplikace:

1. Vytvořit složku `project_dir/Luzko/res/values-xx/`, kde `project_dir` je složka, ve které je uložen projekt. Písmena `xx` značí dvoupísmenný kód jazyka, např. pro francouzštinu je tento kód `fr`.
2. Ve složce `values-xx` vytvořit soubor `string.xml`, do něj zkopírovat obsah ze souboru `'project_dir/Luzko/res/values/strings.xml'` a přeložit veškeré texty v tomto souboru do požadovaného jazyka.

- stejným způsobem do složky *values-xx* přidat soubor **arrays.xml**, v něm navíc přidat do pole *'set\_language'* název jazyka a do pole *'set\_language\_values'* číselnou hodnotu pořadí, ve kterém se jazyk bude zobrazovat v Menu
- stejným způsobem jako v bodě 3 upravit soubor **arrays.xml** ve všech složkách *values-xx* a ve složce *values* (všechny požadované složky jsou uloženy v *project\_dir/Luzko/res/*)
- V následujících souborech

*project\_dir/Luzko/src/com/example/luzko/StartupActivity.java*

*project\_dir/Luzko/src/com/example/luzko/SettingsActivity.java*

vyhledat metodu **setLanguage()** a v ní přidat do této podmínky:

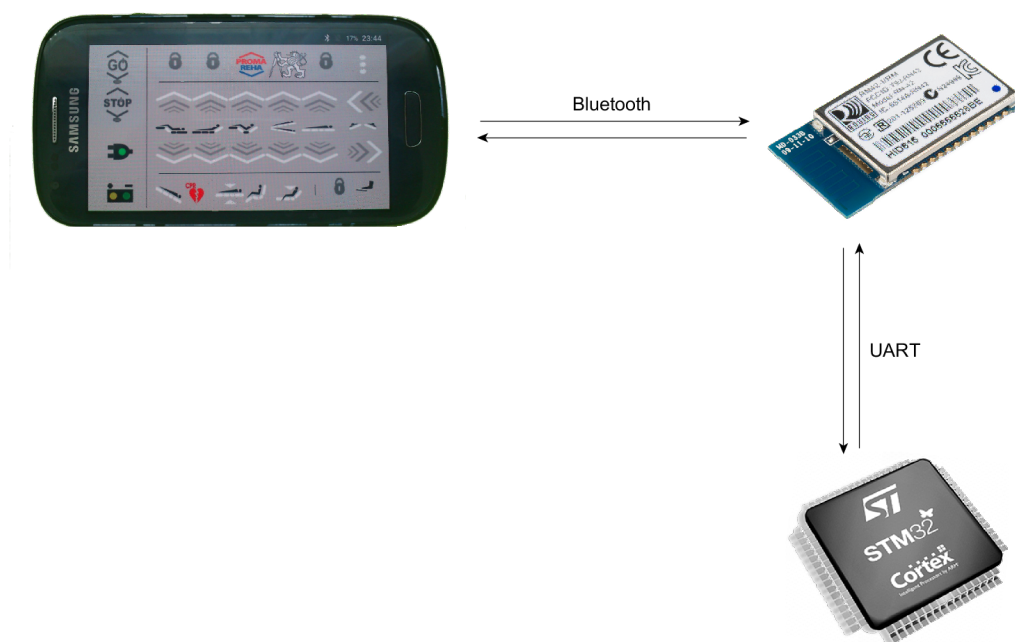
```
switch(SettingsActivity.getLanguage(getApplicationContext()))
{
    case 1: m_locale = Locale.ENGLISH;
           break;
    case 2: m_locale = new Locale("cs-CS");
           break;
    default:
           m_locale = Locale.ENGLISH;
}
```

následující dva řádky kódu, kde **X** značí pořadí jazyka v menu, viz bod 3., a **yy-YY** značí čtyřpísmenný kód lokalizace (například pro francouzskou lokalizaci bude tento kód **fr-FR**):

```
case X: m_locale = new Locale("yy-YY");
       break;
```

## 8 Komunikační protokol

Komunikační protokol je založen na zprávách o délce 1 bajt (8 bitů). Aktivními účastníky komunikace jsou mikrokontrolér osazený na modulu LuzkoBoard v sesterském panelu a řídicí aplikace Lůžko běžící pod OS Android. Komunikace probíhá obousměrně. Pasivním účastníkem je Bluetooth modul RN42, který přeposílá zprávy v jednom směru ze sběrnice UART pomocí Bluetooth do řídicí aplikace, nebo naopak zprávy přijaté přes Bluetooth přeposílá na sběrnici UART do mikrokontroléru. Schéma komunikace je znázorněno na Obr. 22.



Obr. 22: Schéma komunikace mezi aplikací Lůžko a mikrokontrolérem

### 8.1 Parametry pro nastavení sběrnice UART

Pro funkční komunikaci mezi Bluetooth modulem a mikrokontrolérem je nutné správně nastavit parametry pro přenos pomocí UART. Tabulka 10 popisuje nastavení sběrnice na straně mikrokontroléru, na straně Bluetooth modulu musí být nastaveny stejné parametry s výjimkou módů UART RX a UART TX - tato nastavení se týkají pouze mikrokontroléru, viz kapitola 9.2. Pokud se by se parametry pro přenos na obou stranách neshodovaly, budou zprávy špatně vyhodnocovány a komunikace mezi aplikací Lůžko a mikrokontrolérem nebude fungovat.

Pro ověření, že byla přijata nepoškozená zpráva ve správném formátu, jsou do rámce zprávy vloženy dva kontrolní bity. První z nich doplňuje paritu zprávy, druhý je negací paritního bitu. Tím je zajištěno, že aplikace Lůžko ani řídicí program v mikrokontroléru nebudou reagovat na poškozenou zprávu nebo na zprávu odeslanou při nesprávném nastavení sběrnice UART. Vyhodnocení příchozích zpráv popisuje kapitola 8.3

Tabulka 10: Nastavení sběrnice UART na straně mikrokontroléru

Parametr	Hodnota
Přenosová rychlost (baudrate)	9600 baud/s
Parita	ne
Start bit	1
Stop bit	1
Délka datového symbolu	8 bitů
Oversampling	16 bitů
UART TX mód	standardní
UART RX mód	interrupt

## 8.2 Komunikační rámce

Řídící aplikace s mikrokontrolérem komunikují pomocí 8-bitových zpráv, jejichž význam se liší podle směru komunikace.

### Aplikace Lůžko → mikrokontrolér

Ve směru z řídicí aplikace do mikrokontroléru jsou vysílány zprávy pouze v případě, že došlo ke stisknutí nebo uvolnění některého z emulovaných tlačítek sesterského panelu. Po stisku tlačítka se po dobu, kdy zůstane stisknuté, vysílají zprávy v intervalu 100ms, (více viz kapitola 9.3). Formát a význam zpráv vysílaných z aplikace Lůžko do mikrokontroléru popisuje tabulka 11.

Tabulka 11: Formát zprávy ve směru aplikace Lůžko → mikrokontrolér

Bit	7	6	5	4	3	2	1	0
Význam	Set	MessageID					Parity	$\overline{\text{Parity}}$

kde jednotlivé části zprávy mají následující význam:

<b>Set</b>	HIGH při stisknutí tlačítka, LOW při uvolnění tlačítka nebo chybové zprávě
<b>MessageID</b>	5-bitové ID tlačítka, 0 při chybové zprávě
<b>Parity</b>	Paritní bit, musí platit $(\sum_{n=1}^7 bit_n) \bmod(2) = 0$
$\overline{\text{Parity}}$	Negovaný paritní bit, zajišťuje, že zpráva <b>0x00</b> nebo <b>0xFF</b> bude správně vyhodnocena jako chybná

### Mikrokontrolér → aplikace Lůžko

Ve směru z mikrokontroléru do aplikace Lůžko jsou vysílány zprávy neustále v pravidelném intervalu 100ms. Zprávy obsahují stav stavových LED diod, přičemž se střídají zprávy s první a druhou částí stavů LED diod, emulované LED diody se tedy v layoutu aplikace Lůžko aktualizují v intervalu 200ms. Důvod, proč je zpráva rozdělena do dvou částí, je omezená délka pole s informacemi o stavu LED diod na 5 bitů. Každý z těchto 5 bitů nese informaci o stavu jedné LED diody, těch je celkem 9. Proto je celá zpráva rozdělena do dvou částí, první

část pro diody 1~5 a druhá část pro diody 6~9. Formát zprávy a význam jednotlivých částí popisuje tabulka 12.

Tabulka 12: Formát zprávy ve směru mikrokontrolér → aplikace Lůžko

Bit	7	6	5	4	3	2	1	0
Význam	0	LED5	LED4	LED3	LED2	LED1	Parity	$\overline{\text{Parity}}$
	1	0	LED9	LED8	LED7	LED6		

kde jednotlivé části zprávy mají následující význam:

<b>LED<sub>n</sub></b>	bit nese informaci o stavu příslušné LED
<b>Parity</b>	Paritní bit, musí platit $(\sum_{n=1}^7 bit_n) \bmod(2) = 0$
$\overline{\text{Parity}}$	Negovaný paritní bit, zajišťuje, že zpráva <b>0x00</b> nebo <b>0xFF</b> bude správně vyhodnocena jako chybná

### 8.3 Postup vyhodnocení přijaté zprávy

#### Aplikace Lůžko → mikrokontrolér

Přijátá zpráva je v řídicím programu mikrokontroléru vyhodnocena podle schématu na Obr. 23. Pokud je přijata neplatná nebo poškozená zpráva, je zahozena a multiplexery jsou nastaveny do výchozího stavu, tedy do takového, kdy není sepnuté žádné tlačítko. Neplatná nebo poškozená zpráva je detekována podle dvou paritních bitů. Je-li zpráva správně přijata, musí vždy platit následující dvě podmínky:

1. Součet všech bitů zprávy kromě LSB ( $bit_0$ ) musí být dělitelný dvěma:

$$(\sum_{n=1}^7 bit_n) \bmod(2) = 0$$

2.  $bit_0$  a  $bit_1$  nesmějí mít stejnou hodnotu,  $bit_0$  je negací  $bit_1$ :

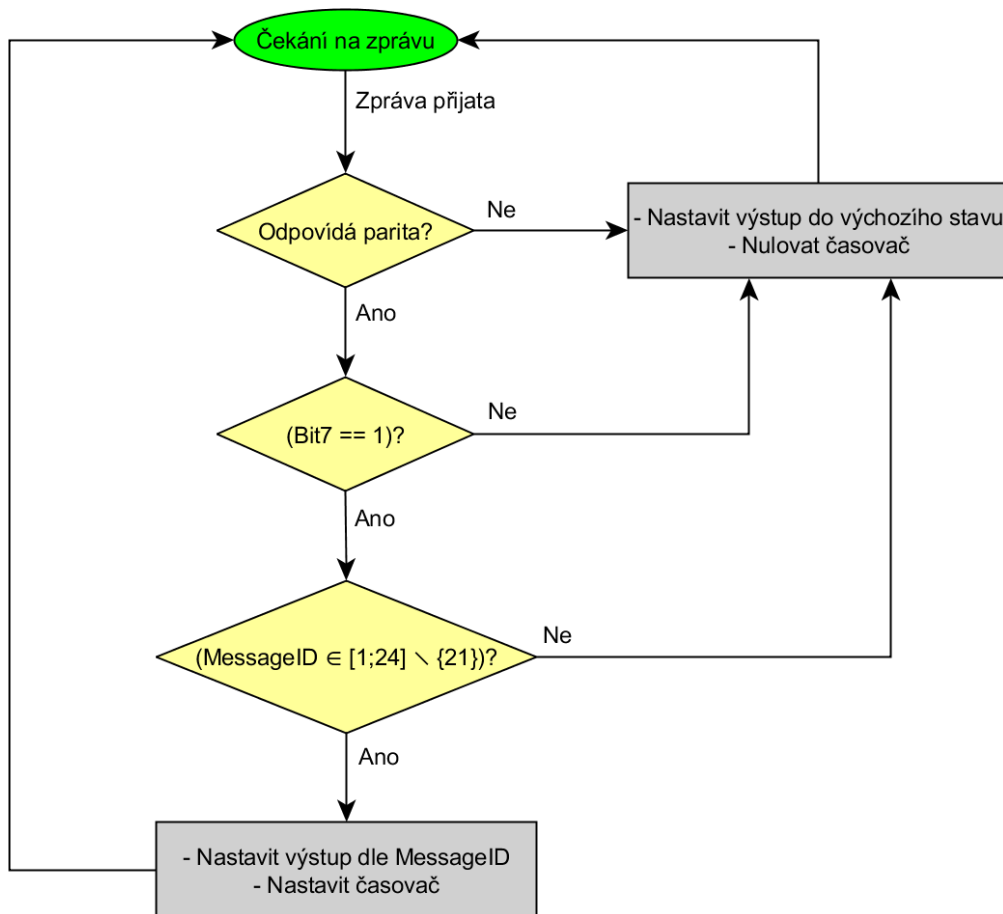
$$bit_0 = \overline{bit_1}$$

Pokud je zpráva správně přijata, ověřuje se hodnota MSB ( $bit_7$ ). Pokud je MSB = 0, pak se může jednat o zprávu, že bylo uvolněno tlačítko, nebo jde o chybovou zprávu. V obou případech bude reakce na straně mikrokontroléru stejná, rozpojí se multiplexery.

Dále se vyhodnocuje část zprávy **MessageID**, kde je uložen 5-bitový identifikátor tlačítka. Tlačítka mají platné identifikátory  $ID \in [1; 24] \setminus \{21\}$ . Neplatný identifikátor opět signalizuje neplatnou zprávu.

Pokud je zpráva správně přijata a má platný identifikátor, řídicí program v mikrokontroléru nastaví požadované výstupy a tím pomocí analogových multiplexerů emuluje stisk tlačítka.

Pro úplnost je tu uvedeno i nastavení časovače. Ten má nastavený ochranný interval, který zajišťuje, že při výpadku komunikace během stisku tlačítka nezůstanou analogové multiplexery vzájemně propojené, tedy že nebudou emulovat stisknuté tlačítko. Tato ochrana je součástí řídicího programu v mikrokontroléru, kterému se věnuje kapitola 9.



Obr. 23: Rámcové schéma vyhodnocení přijaté zprávy na straně mikrokontroléru

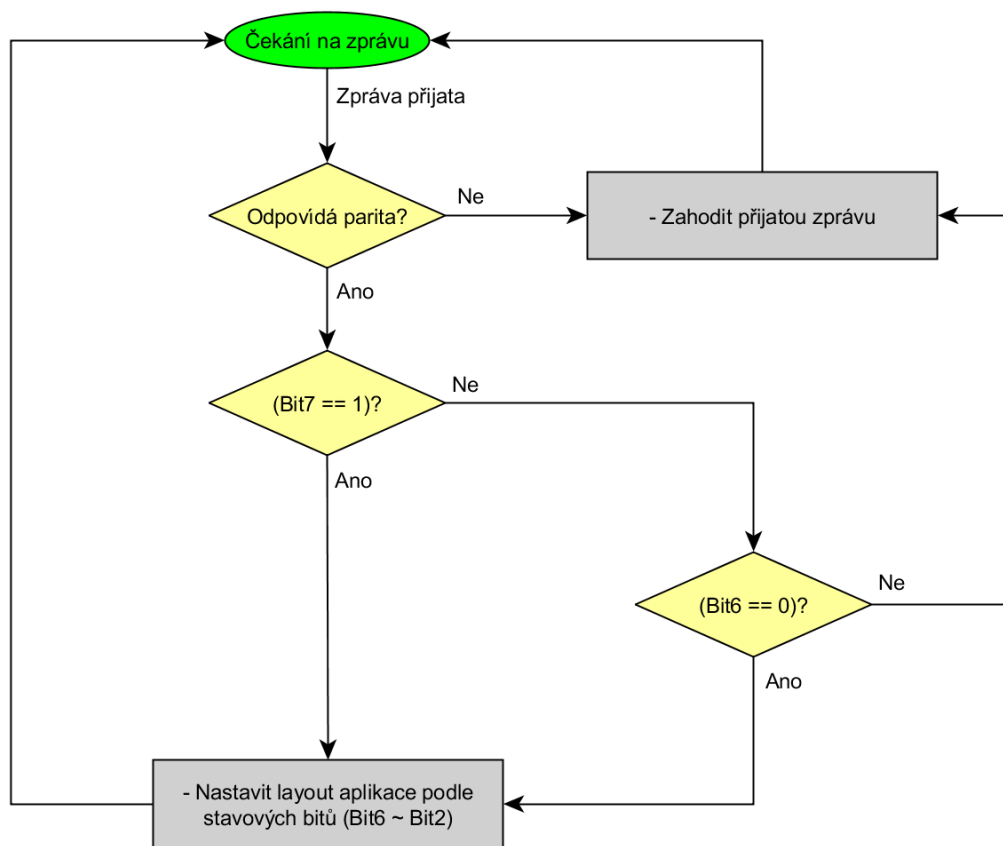
### Mikrokontrolér → aplikace Lůžko

Přijatá zpráva na straně mobilní aplikace Lůžko se vyhodnocuje obdobně jako na straně mikrokontroléru. Opět se nejprve ověřuje, zda jsou správně nastaveny oba paritní bity, a pokud nejsou, zpráva se zahazuje.

Dalším krokem je ověření hodnoty MSB (**bit<sub>7</sub>**), v tomto případě se ale jedná o identifikátor, zda jde o první nebo druhou část stavů LED diod. První část zprávy využívá všech 5 stavových bitů zprávy, proto je vždy vyhodnocena jako správně přijatá. Druhá část zprávy využívá jen 4 stavové bity, protože celkem se odesílá stav 9 stavových LED diod - 5 v první části a 4 v části druhé. Proto se u druhé části zprávy vyhodnocuje, zda je nejvyšší stavový bit (**bit<sub>6</sub>**) ve stavu LOW, a pokud ne, je zpráva vyhodnocena jako neplatná. Dále se už jen nastavují příslušná tlačítka a symboly v *layoutu* aplikace, viz kapitola 7.3.

Proces vyhodnocení přijaté zprávy v mobilní aplikaci Lůžko je znázorněn na Obr. 24





Obr. 24: Rámcové schéma vyhodnocení přijaté zprávy na straně mobilní aplikace

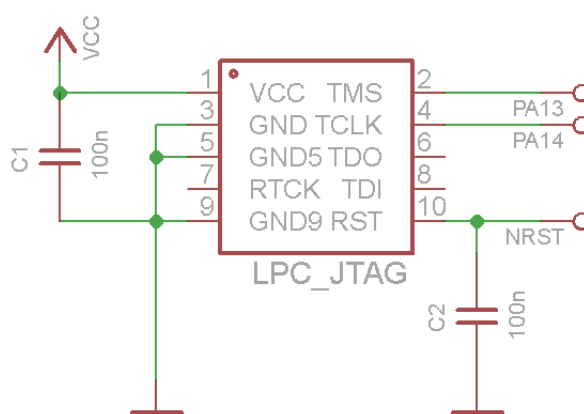


## 9 Řídicí program v mikrokontroléru STM32L053

Řídicí program v mikrokontroléru se skládá z několika částí, jež jsou podrobněji rozebrány v podkapitolách 9.2 až 9.5. Nejprve je nutno nastavit veškeré periferie mikrokontroléru. Základem je blok hodin RCC, dále vstupy a výstupy mikrokontroléru. Poté se nastaví výchozí stav všech výstupů a parametry UART. Po inicializaci periférií se může spustit hlavní část programu, tu tvoří nekonečná smyčka, ve které se zpracovávají jednotlivé události. Poslední částí je obsluha přerušení.

### 9.1 Naprogramování a debugování

Pro naprogramování mikrokontroléru a následné debugování programu je na modulu LuzkoBoard osazen LPC JTAG, schéma zapojení je na Obr. 25.



Obr. 25: Schéma zapojení LPC JTAG pro flashování a debugování programu

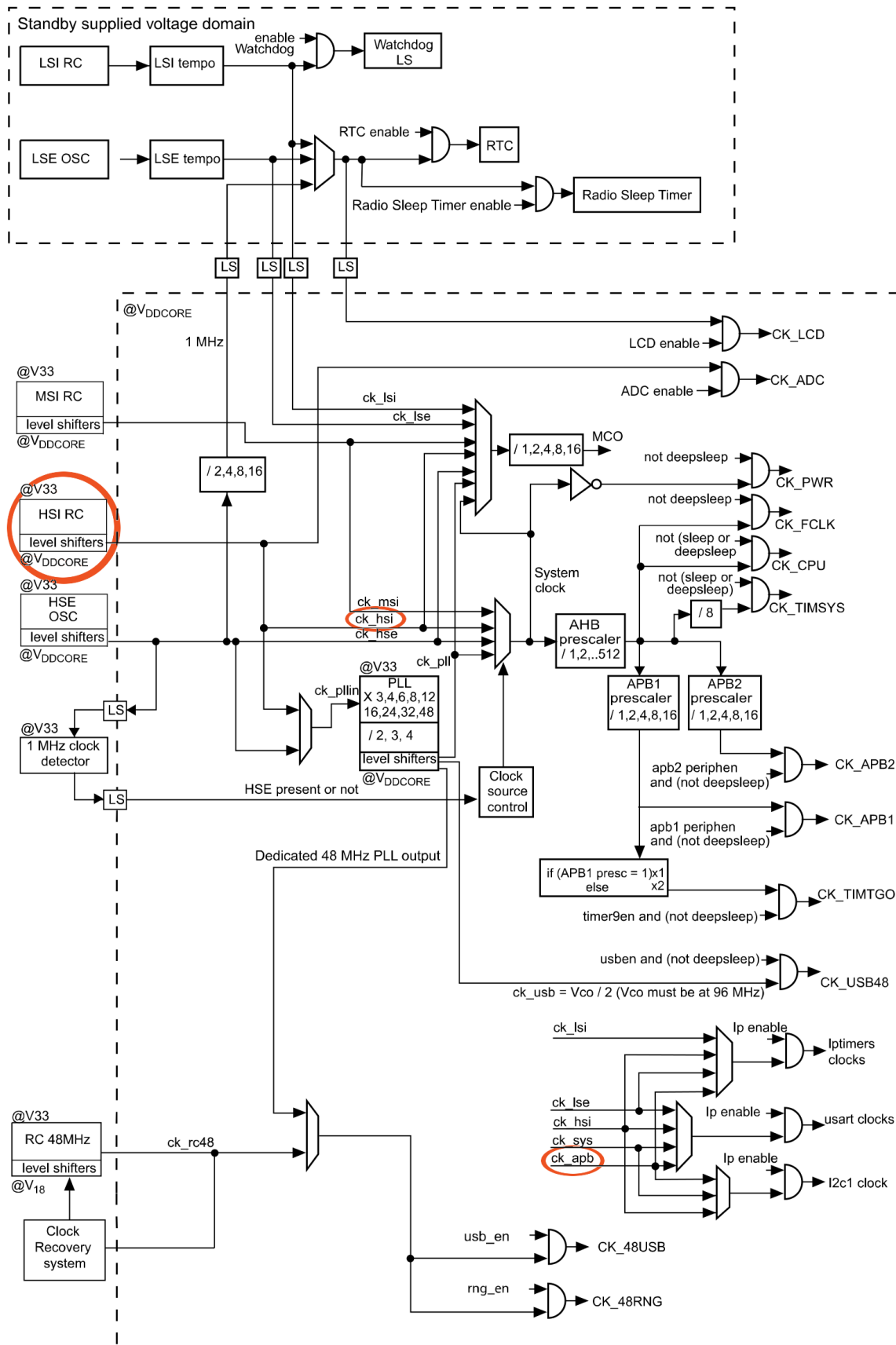
### 9.2 Inicializace

Před spuštěním samotného řídicího algoritmu je nutné nastavit všechny využité periferie a funkce mikrokontroléru, nadefinovat funkci jednotlivých pinů a nastavit správné časování.

#### Nastavení časování

Prvním krokem je nastavení časování. Použitý mikrokontrolér obsahuje několik vnitřních oscilátorů s frekvencí až 48MHz a s dostatečnou přesností, aby je bylo možno použít i pro časování sériové linky. Díky tomu nebylo nutné připojovat externí oscilátor ani krystal. Přehled vnitřních zdrojů časování a jejich rozvedení do jednotlivých periférií mikrokontroléru zobrazuje schéma na Obr. 26. Před nastavením jakékoliv periferie se jí musí napřed zapnout zdroj hodin.

V aplikaci byl použit jako zdroj hodin interní oscilátor HSI s frekvencí 16MHz. Tato frekvence je použita v celém mikrokontroléru, veškeré předděličky jsou nastaveny tak, aby dělily frekvenci hodnotou 1 a neměly tak žádný vliv.



Obr. 26: Schéma generátoru hodin mikrokontroléru STM32L053 [22], zvýrazněny jsou použité hodiny

## Nastavení pinů a jejich funkcí

Mikrokontrolér umožňuje nastavit většinu pinů jako běžný GPIO, nebo může mít pin alternativní funkci. GPIO může být nastaven jako vstup nebo výstup. Pro vstup je možno zapnout pull-up nebo pull-down rezistor zajišťující klidový stav. Některé vstupy umožňují zapnout hysterezi (Schmittův obvod). Výstupy mohou být konfigurovány jako open-collector nebo push-pull. Příkladem alternativní funkce je např. vstup AD převodníku, komunikace USART, výstup na LCD displej nebo vstup externího oscilátoru. V tomto projektu je využita alternativní funkce pinů **PA2** a **PA3**, ty jsou využity pro komunikaci pomocí sběrnice UART, jejich funkce je tedy **USART2\_RX** a **USART2\_TX**. Úplný přehled nastavení jednotlivých pinů nabízí tabulka 13. Hodnota funkce "-" znamená, že pin nemá přiřazenou funkci (zůstává nastaven jako vstup).

Tabulka 13: Nastavení použitých pinů, piny jsou seřazeny podle osazení na pouzdře LQFP48

Pin	Funkce	Pin	Funkce
NRST*	RESET	PB2	GPIO_Input
VSS*	GND	PB10	GPIO_Input
VDD*	+3.3V	PB11	GPIO_Input
PA2	USART2_TX	PB12	GPIO_Input
PA3	USART2_RX	PB13	GPIO_Input
PA4	GPIO_Output	PB14	GPIO_Input
PA5	GPIO_Output	PB15	GPIO_Input
PA6	GPIO_Output	PA8	GPIO_Input
PA7	GPIO_Output	PA13	SYS_SWDIO
PB0	GPIO_Output	PA14	SYS_SWCLK
PB1	GPIO_Input		

\* pin neumožňuje nastavení alternativní funkce

## Nastavení UART

Pro správnou funkci UART je potřeba nastavit vstupní a výstupní pin (zde **PA2** a **PA3**), zdroj hodin (zde hodiny APB 16MHz) a příslušné registry podle požadovaných parametrů (parametry viz kapitola 8.1). Zde jsou možné dva různé přístupy: ruční nastavení registrů, nebo využití knihoven HAL. V projektu je zvolena druhá možnost, protože nabízí jednodušší obsluhu UART. Funkce použitá pro nastavení této periferie pak vypadá následovně:

```
ART_HandleTypeDef huart2;
void MX_USART2_UART_Init(void)
{
    huart2.Instance = USART2;
    huart2.Init.BaudRate = 9600; // set baudrate
    huart2.Init.WordLength = UART_WORDLENGTH_8B; // length of message word -> 8 bit
    huart2.Init.StopBits = UART_STOPBITS_1;
    huart2.Init.Parity = UART_PARITY_NONE;
    huart2.Init.Mode = UART_MODE_TX_RX; // enable Transmit nad Receive
    huart2.Init.HwFlowCtl = UART_HWCONTROL_NONE;
    huart2.Init.OverSampling = UART_OVERSAMPLING_16;
    HAL_UART_Init(&huart2); // set params to MCU registers
    __HAL_UART_FLUSH_DRREGISTER(&huart2); // clear the buffer to prevent overrun
}
```

## Nastavení výchozího stavu výstupů a proměnných

Nyní už je vše potřebné pro běh programu nastaveno a zbývá jen nastavit veškeré výstupy a proměnné do výchozího stavu. To je nutné především proto, aby nemohlo dojít např. k emulaci stisknutého tlačítka nebo ke špatnému nastavení časovače ještě před spuštěním nekonečné smyčky. Také ještě nejsou povolena přerušení (*interrupt*). Program pracuje s přerušením SysTick, které se volá pravidelně v intervalu 1ms a využívá se pro cyklicky vykonávané úkony, a dále s přerušením pro příjem dat pomocí sériové linky. V programu se tedy provedou následující příkazy:

```
/* enable interrupt for UART Receive */
HAL_UART_Receive_DMA(&huart2, rx_init, 1);

/* enable SysTick interrupt each 1ms */0
HAL_SYSTICK_Config(HAL_RCC_GetHCLKFreq()/1000);
HAL_SYSTICK_CLKSourceConfig(SYSTICK_CLKSOURCE_HCLK);

/* initial state for variables */
TickCounter = 0;           // reset SysTick counter
SwitchCounter = 0;        // reset counter for multiplexers
uint8_t ledToggle = 0;    // to set what part of LED states will be transmitted
uint8_t ledMessage = 0;   // to store status LED states
switch_init();            // no pair of analogue I/O shall be linked together
```

### 9.3 Hlavní smyčka programu

Po inicializaci se v programu volá hlavní smyčka (*main loop*). Jedná se o část programu, která běží stále dokola. Pokud přijde přerušení, zpracuje se příslušná rutina a program se vrátí zpět do hlavní smyčky.

V hlavní smyčce se volá pouze funkce pro vyčtení stavů LED diod a jejich následné odeslání aplikací Lůžko pomocí sériové linky. Aby se zprávy odesílaly v pravidelných a předem nastavených intervalech (zde 100ms), využívá se pro časování čítač **SysTickCounter**, který je součástí přerušení SysTick a inkrementuje se v intervalu 1ms. Pak stačí jednoduchá podmínka, aby se funkce pro odeslání stavů LED volala až tehdy, kdy je splněna podmínka

```
(SysTickCounter >= 100)
```

Následně se čítač resetuje.

Pro větší přehlednost kódu je konstanta **100** nahrazena definovanou konstantou **TX\_TIMER**:

```
#define TX_TIMER          100                // timer for UART_TX
```

Takto definované konstanty lze při úpravě nebo údržbě kódu snadno měnit a není nutné je vyhledávat až na místech, kde jsou použity. Největším přínosem tohoto přístupu je fakt, že takováto konstanta může být referencována na více místech kódu a je zajištěno, že její hodnota bude v případě potřeby změněna na všech příslušných místech zároveň.

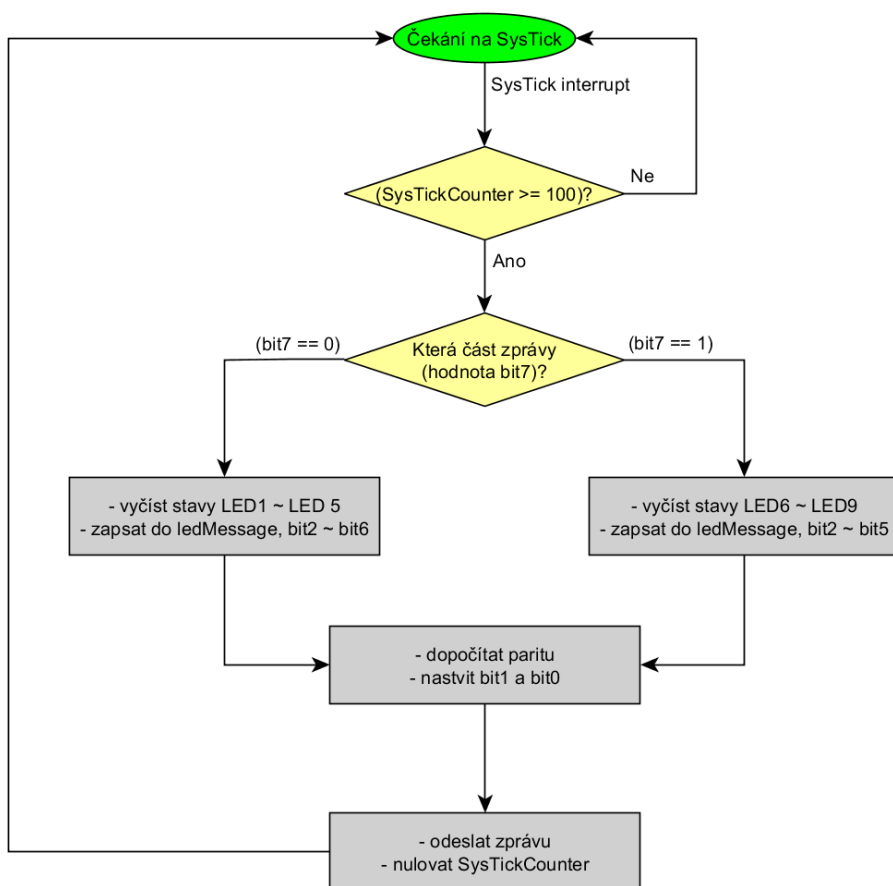
Postup pro odeslání zprávy se stavy LED diod je znázorněn na diagramu na Obr. 27. Před odesláním zprávy je ještě nutné dopočítat hodnotu paritního a negovaného paritního bitu. To je realizováno následujícím způsobem - postupně se pomocí bitového posunu a masky vyčtou hodnoty z bit<sub>2</sub> až bit<sub>7</sub> tak, aby je bylo možné posčítat. Nad jejich součtem se zavolá operace modulo 2, zde realizovaná bitovou operací AND s maskou 0x01, a do paritního bitu se zapíše výsledná hodnota. Do negovaného paritního bitu se pak může zapsat negovaná

hodnota paritního bitu, to je zde realizováno výpočtem parity a následnou bitovou operací XOR s hodnotou 0x01. Takto vypadá výpočet parity v kódu:

```

/* uint8_t ledMessage stores LED_states in bit2-bit6 and message_part_selection in bit7 */
int parity = 0;
parity += (ledMessage >> 2) & 0x01u; // get value of bit_2
parity += (ledMessage >> 3) & 0x01u;
parity += (ledMessage >> 4) & 0x01u;
parity += (ledMessage >> 5) & 0x01u;
parity += (ledMessage >> 6) & 0x01u;
parity += (ledMessage >> 7) & 0x01u;
ledMessage &= 0xFC; // reset parity bits (bit_1 and bit_0)
ledMessage |= (parity & 0x01u)<<1; // set value of parity to bit_1
ledMessage |= (parity & 0x01u)^0x01u; // set negation of parity to bit_0

```



Obr. 27: Schéma programu v nekonečné smyčce pro odeslání stavů LED diod

## 9.4 Přerušení SysTick

Hlavní funkcí tohoto přerušení je, že generuje 1ms čítač SysTickCounter. Dále je v něm implementován ochranný interval pro rozpojení multiplexerů v případě, že během stisku tlačítka v aplikaci dojde k přerušení komunikace.

## 9.5 Přerušení UART\_Receive

Toto přerušení zpracovává příchozí zprávy po sériové lince a je voláno vždy, když je přijat alespoň 1 bajt dat (a protože je délka datového slova nastavena na 8 bitů, je toto přerušení

voláno vždy při příjmu dat). Příjem dat v přerušení je zvolen proto, aby byly příchozí zprávy vyhodnoceny neprodleně po jejich přijetí.

Po přijetí celé zprávy je volána funkce `HAL_UART_RxCpltCallback()`, ve které je zpracována příchozí zpráva. Vyhodnocení příchozí zprávy je popsáno v kapitole 8.3. Při přijetí neplatné nebo poškozené zprávy jsou vždy rozpojeny multiplexery a nulován ochranný čítač. Při přijetí zprávy s odpovídajícími paritními bity a nejvyšším bitem ve stavu HIGH jsou nastaveny multiplexery tak, aby emulovaly stisk příslušného tlačítka podle přijatého MessageID a vždy je znovu nastaven čítač pro ochranný interval. Délka ochranného intervalu je nastavena takto:

$$\text{Ochranny\_interval} = 2x(\text{interval\_zpravy}) + 10\%.$$

Aplikace při emulaci stisknutého tlačítka pravidelně odesílá zprávy, že je tlačítko stisknuté, a to v intervalu 100ms. To znamená, že při tomto intervalu zpráv je ochranný interval nastaven na 220ms, a pokud do té doby nepříjde buď zpráva, že je tlačítko stále stisknuté, nebo že bylo rozpojeno, budou multiplexery imitující stisk tlačítka automaticky rozpojeny.



## 10 Závěr

Cílem této práce byla realizace bezdrátového ovládání nemocničního lůžka společnosti PROMA REHA s.r.o., které nabídne stejné funkce jako sesterský panel.

Pro ovládání bylo vytvořena aplikace Lůžko pro OS Android, která nabízí stejné funkce, jako sesterský panel, včetně zobrazení stavových LED diod. Výjimkou jsou dvě speciální funkce panelu (vypnutí napájení a reset řízení lůžka), které vyžadují současný stisk dvou tlačítek. Aplikace dále nabízí možnost výběru mezi českým a anglickým jazykem, nápovědu pro ovládání lůžka a umožňuje nastavení Bluetooth MAC adresy modulu RN42, osazeného na modulu LuzkoBoard. Je tak možné připojit se s aplikací i k jinému modulu, než se kterým byla testována, bez nutnosti úpravy zdrojového kódu.

Modul LuzkoBoard tvoří druhou část realizace bezdrátového ovládání. Tento modul emuluje stisk tlačítek na základě dat z řídicí aplikace a vyčítá stavy LED diod, aniž by omezil funkce samotného panelu. Je přímo napojen na řídicí jednotku a je navržen tak, aby se vešel do dutiny uvnitř sesterského panelu. Nevyžaduje samostatný zdroj napájení, napájí se stejně jako řídicí jednotka. Navíc nevyžaduje pro svou instalaci do panelu žádný zásah do hardwaru řídicí jednotky, lze ho připojit jako modul.

Komunikace mezi mobilní aplikací Lůžko a modulem LuzkoBoard je řešena pomocí Bluetooth modulu, který komunikuje s aplikací přes emulovanou sériovou linku a přeposílá data řídicímu mikrokontroléru pomocí rozhraní USART. Emulace klávesnice s maticovým zapojením je realizována pomocí dvou analogových multiplexerů.

Přínosem této práce je možnost ovládání nemocničního lůžka přímo pomocí tabletu nebo mobilního telefonu, který má dnes téměř každý. Lze tak usnadnit polohování ložné plochy především lidem, kteří jsou na lůžko upoutáni po delší dobu. Aplikace Lůžko se snadno ovládá a pro zjednodušení obsahuje i nápovědu k ovládání lůžka. Navíc nabízí uživatelskou odezvu v podobě zabarvení tlačítka při stisknutí a držení a také vibrační odezvu při stisku a při uvolnění tlačítka.

## Reference

- [1] ANDRÁŠSY, Filip. *Mobilní aplikace pro ovládání inteligentního lůžka*. Praha: ČVUT 2015. Bakalářská práce, ČVUT, Fakulta elektrotechnická, Katedra radioelektroniky.
- [2] IDC: *Smartphone OS Market Share, 2015 Q2* [online], vid. 23.5.2016. Dostupné z: <http://www.idc.com/prodserv/smartphone-os-market-share.jsp>
- [3] Open Handset Alliance - *Alliance members* [online], vid. 24.5.2016. Dostupné z: [http://www.openhandsetalliance.com/oha\\_members.html](http://www.openhandsetalliance.com/oha_members.html)
- [4] Geek.com - *Google explains why Acer was asked to skip Aliyun phone launch* [online], vid. 23.5.2016. Dostupné z: <http://www.geek.com/mobile/google-explains-why-acer-was-asked-to-skip-aliyun-phone-launch-1516351/>
- [5] Google Developers - *Developer Challenge: Google Fit* [online], vid. 24.5.2016. Dostupné z: <https://developers.google.com/fit/challenge/>
- [6] SocialCompare - *Android versions comparison* [online], poslední revize 20.5.2016, vid. 24.5.2016. Dostupné z: <http://socialcompare.com/en/comparison/android-versions-comparison>
- [7] Bluetooth developer portal - *RFCOMM* [online], vid. 24.5.2016. Dostupné z: <https://developer.bluetooth.org/TechnologyOverview/Pages/RFCOMM.aspx>
- [8] PROMA REHA s.r.o. - *Návod k použití pro lůžka následujících výrobních řad: Trend 2C, Trend ICU*, ver. 12-1, 27.6.2012.
- [9] BRIDGES, N., JARQUIN-VALDIVIA, A. *Use of the Trendelenburg Position as the Resuscitation Position: To T or Not to T.*, [vid. 17.5.2016]. American Journal of Critical Care. [online] Dostupné z: <http://ajcc.aacnjournals.org/content/14/5/364.full>
- [10] Praha: ČVUT - *Vyhodnocení maticové klávesnice* [online], vid. 21.5.2016. Dostupné z: [https://moodle.dce.fel.cvut.cz/file.php/29/cviceni/Maticova\\_klavesnice.pdf](https://moodle.dce.fel.cvut.cz/file.php/29/cviceni/Maticova_klavesnice.pdf)
- [11] Microchip - *RN42 Data Sheet* [online], vid. 25.1.2016. Dostupné z: [ww1.microchip.com/downloads/en/DeviceDoc/50002328A.pdf](http://ww1.microchip.com/downloads/en/DeviceDoc/50002328A.pdf)
- [12] TME Electronic Components - *74HCT4051 - IC:digital* [online], vid. 10.2.2016. Dostupné z: <http://www.tme.eu/gb/details/74hct4051/decoders-multiplexers-switches/>
- [13] Texas Instruments - *LP2980-ADJ* [online], vid. 27.1.2016. Dostupné z: <http://www.ti.com/product/LP2980-ADJ/technicaldocuments>
- [14] Pragoboard s.r.o. - *POOL servis - princip výroby* [online], vid. 29.1.2016. Dostupné z: <http://www.pragoboard.cz/node/76>
- [15] Android Developers - *Dashboards* [online], poslední revize 2.5.2016, vid. 20.5.2016. Dostupné z: <http://developer.android.com/about/dashboards/index.html>
- [16] Android Developers - *Adding the App Bar* [online], vid. 26.4.2016. Dostupné z: <https://developer.android.com/training/appbar/index.html>
- [17] Android Developers - *Bluetooth* [online], vid. 23.4.2016. Dostupné z: <http://developer.android.com/guide/topics/connectivity/bluetooth.html>

- [18] KLINE, Paul, KUMAR, C Bala, THOMPSON, Timothy. *Bluetooth Application Programming with the Java APIs*. 1st. ed. Morgan Kaufmann, 2008. ISBN 978-0-12-374342-8
- [19] PETERSON, Brian, BALDWIN, Rusty, KHAROUFEH, Jeffery. *Bluetooth Inquiry Time Characterization and Selection* [online], vid. 25.5.2016. Dostupné z: <http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=1661527>
- [20] Aplikace pro Android ve službě GooglePlay - *Bluetooth Address Finder* [online], vid. 13.2.2016. Dostupné z: <https://play.google.com/store/apps/details?id=com.ccpcreations.android.bluetoothmacfinder>
- [21] Blog T - *Bluetooth Data Transfer with Android* [online], vid. 24.5.2016. Dostupné z: <https://tsilian.wordpress.com/2012/11/06/bluetooth-data-transfer-with-android/>
- [22] ST Microelectronic - *STML053C8 Product Datasheet* [online], vid. 20.4.2016. Dostupné z: [http://www.st.com/content/st\\_com/en/products/microcontrollers/stm32-32-bit-arm-cortex-mcus/stm32l0-series/stm32l0x3/stm32l053c8.html](http://www.st.com/content/st_com/en/products/microcontrollers/stm32-32-bit-arm-cortex-mcus/stm32l0-series/stm32l0x3/stm32l053c8.html)

## Seznam použitých zkratk

<b>AAC</b>	Advanced Audio Coding
<b>ADT</b>	Android Development Tools
<b>API</b>	Application Programming Interface
<b>BER</b>	Bit Error Ratio
<b>BT</b>	Bluetooth
<b>CPR</b>	Cardiopulmonary resuscitation
<b>DPS</b>	Deska plošných spojů
<b>GLONASS</b>	Globalnaja navigacionnaja sputnikovaja sistěma [ru]
<b>GPS</b>	Global Positioning System
<b>GSM</b>	Groupe Spécial Mobile [fr] (2G)
<b>HW</b>	Hardware
<b>LDO</b>	Low-dropout regulator
<b>LED</b>	Light-Emitting Diode
<b>LTE</b>	Long Term Evolution (4G)
<b>MAC</b>	Media Access Control
<b>NFC</b>	Near Field Communication
<b>OS</b>	Operating System
<b>OHA</b>	Open Handset Alliance
<b>RAM</b>	Random Access Memory
<b>RFCOMM</b>	Radio Frequency Communication
<b>ROM</b>	Read-Only Memory
<b>SDK</b>	Software Development Kit
<b>SIP VoIP</b>	Session Initialion Protocol, Voice over Internet Protocol
<b>UMTS</b>	Universal Mobile Telecommunications System (3G)
<b>USART</b>	Universal Synchronous / Asynchronous Receiver and Transmitter
<b>USB</b>	Universal Serial Bus
<b>TTL</b>	Transistor-Transistor Logic

# Přílohy

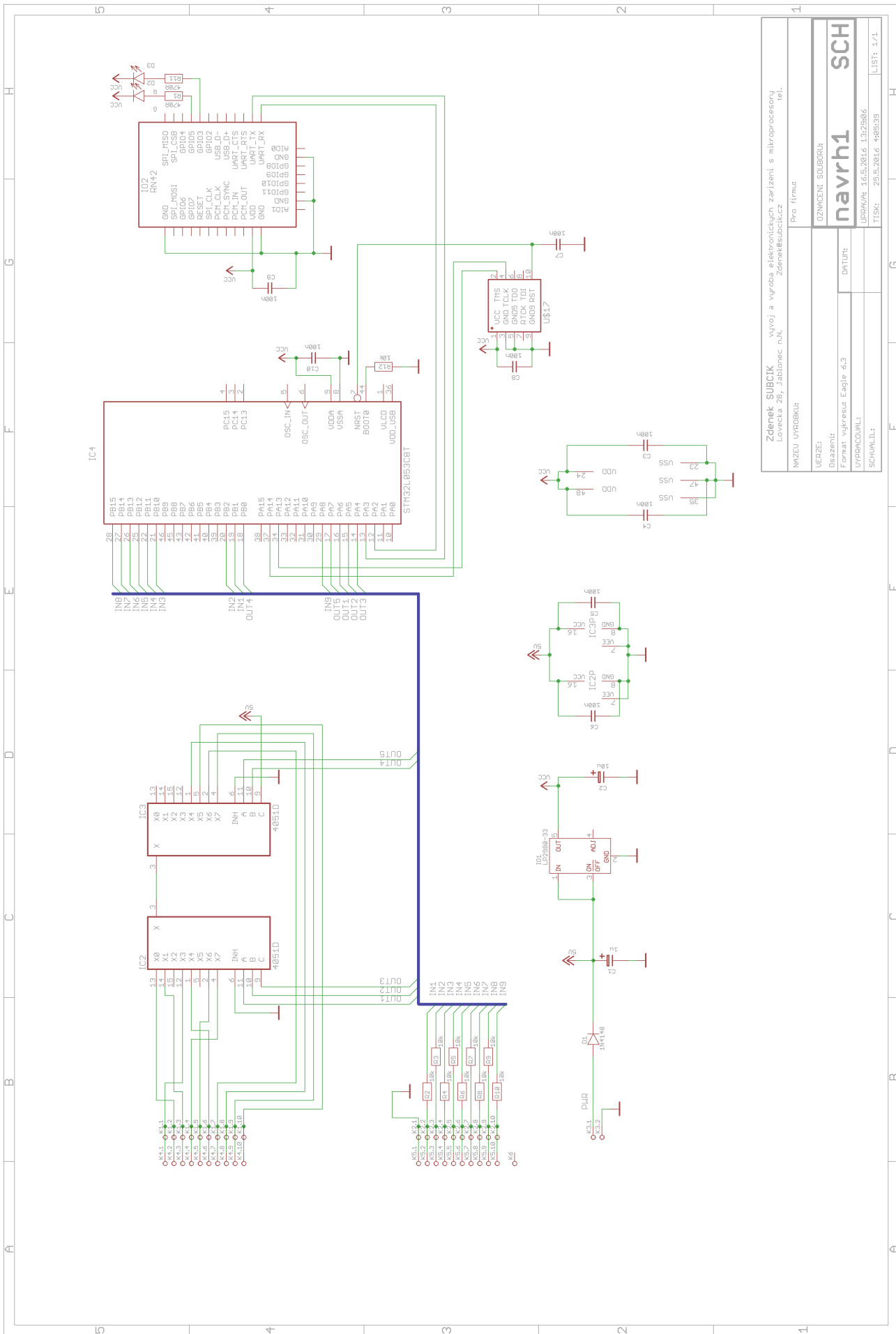
## Seznam příloh:

**I:** Schéma zapojení modulu LuzkoBoard

**II:** DPS modulu LuzkoBoard

**III:** Maska modulu LuzkoBoard

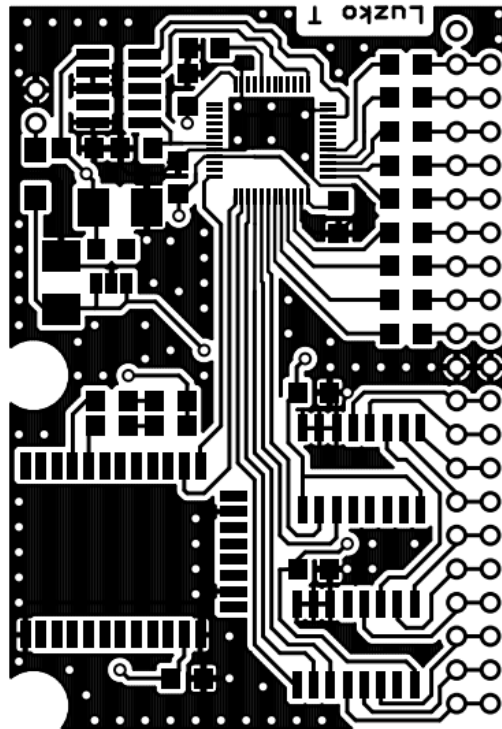
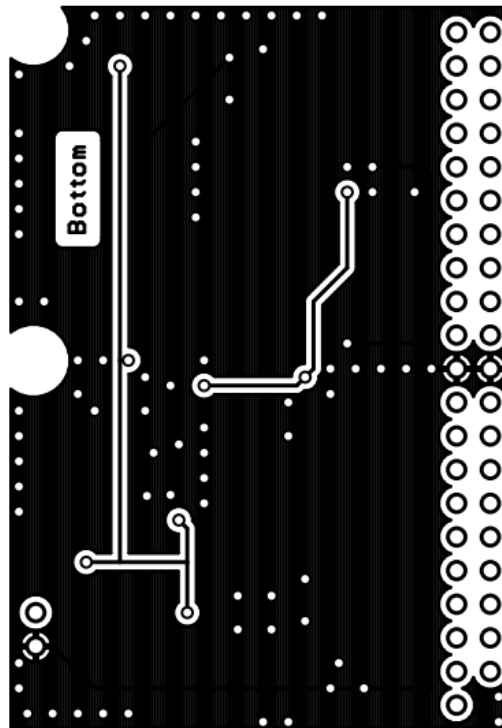
**IV:** Hodnotové a poziční osazení modulu LuzkoBoard



Zdenek SUBCIK    vývoj a výroba elektronických zařízení s mikroprocesory Lovelce 28, Jablonec n.N.    Zdenek@subcik.cz    tel.	
Pro firmu:	
NAZEV VYROBKU:	OZNAČENÍ SOUBORU:
VERZE:	<b>navrh1</b>
Osazení:	Formát výkresu: Eagle 6.3
LYPACOVÁL:	DATE:
SCHVÁLIL:	UPRÁVIL: 16.5.2016 13:28:06
	TISK: 25.5.2016 4:05:59
	LIST: 1/1

Typ vykresu:

DPS (s.spoju - s.souc.)

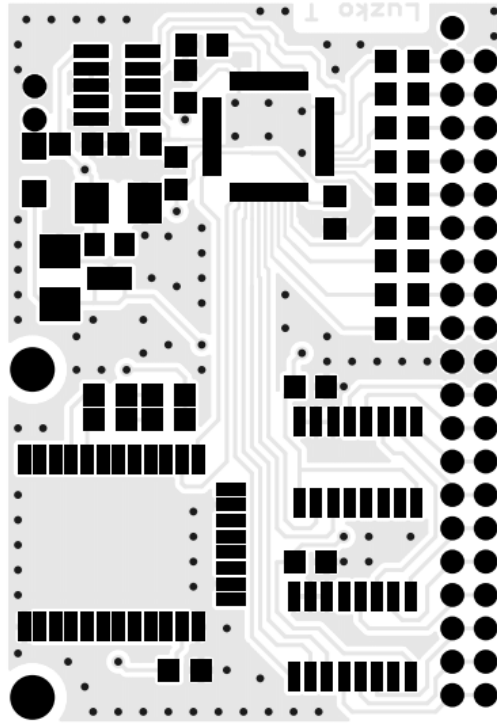
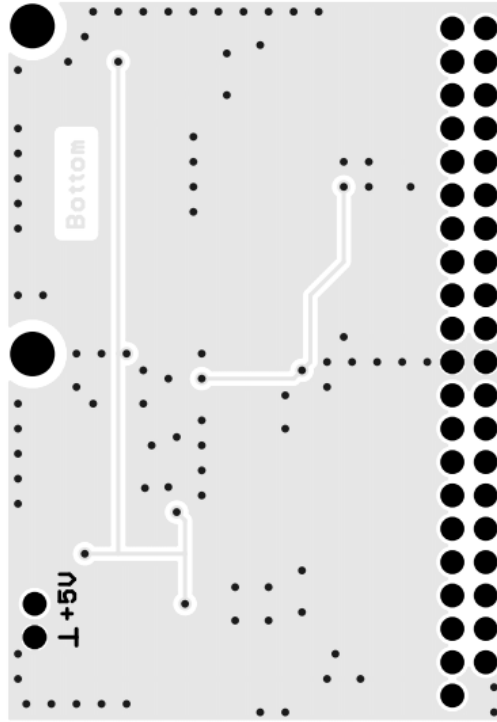


ROZMERY:
NEKOTOVANE TOLERANCE:
TL. MATERIAL / Cu:
KVALITA MAT.:
POURCHOVA UPRAVA SPOJU:
OBRYSOVE OPRACOVANI:
DESKY DODAVAT:

Firma	Urceno pro firmu:
NAZEV VYROBKU:	OZNACENI DOKUMENTU:
VERZE:	<b>navrh1 BRD</b>
Format vykresu: Eagle 6.6	DATE:
UYPRACOVAL:	UPRAVA: 1.2.2016 23:12:42
SCHVALIL:	TISK: 1.2.2016 23:12:46
	MERITOK: 2:1

Typ vykresu:

Maska (s.spoju - s.souc.)



ROZMERY:
NEKOTOVANE TOLERANCE:
TL, MATERIAL / Cu:
KVALITA MAT.:
POURCHOVA UPRAVA SPOJU:
OBYSOUE OPRACOVANI:
DESKY DODAVAT:

Firma

NAZEV VYROBKU:

Urceno pro firmu

VERZE:

OZNAČENÍ DOKUMENTU:

Format vykresu: Eagle 6.6

DATUM:

**navrh1 BRD**

UYPRACOVAL: UPRAVA: 1.2.2016 23:12:42

MERITKO:

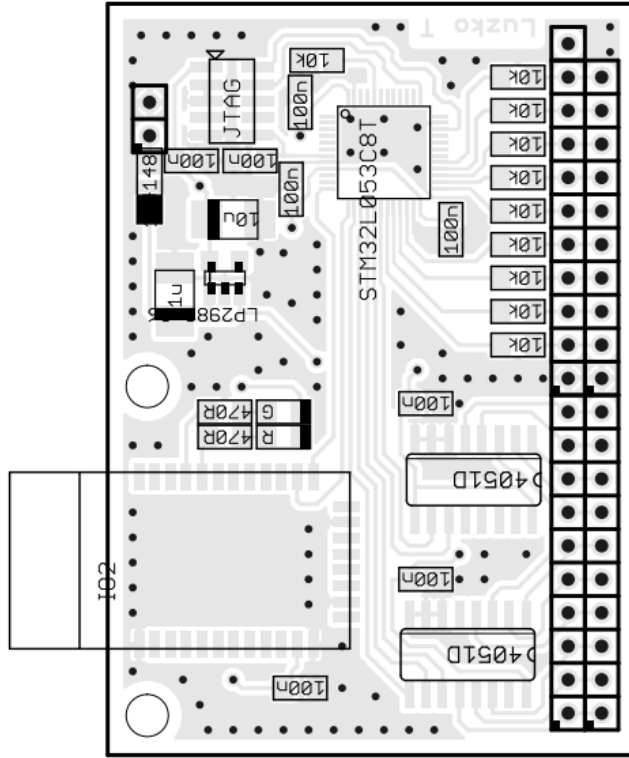
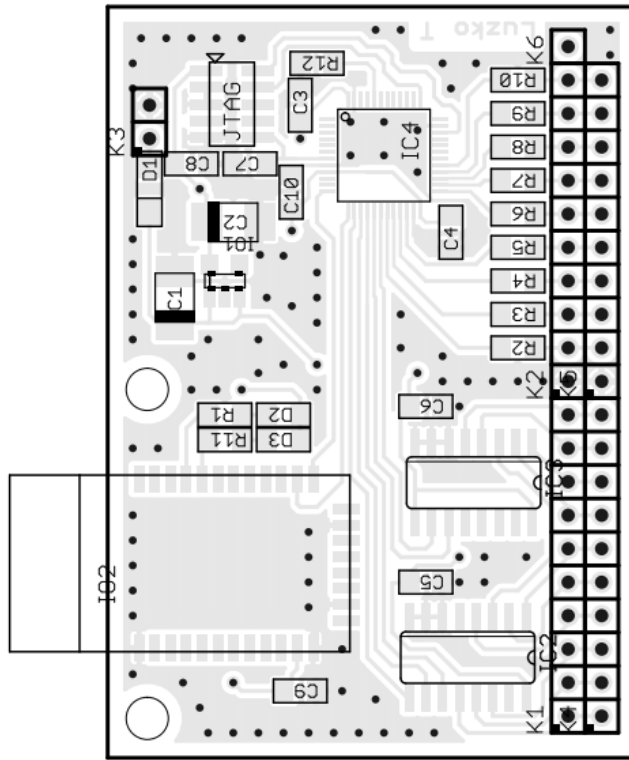
SCHVALIL: TISK: 1.2.2016 23:12:46

2:1



Typ vykresu:

Pozicni - hodnotove osazeni (s.souc.)



ROZMERY:	
NEKOTOVANE TOLERANCE:	
TL. MATERIAL / Cut	
KVALITA MAT.:	
POURCHOVA UPRAVA SPOJU:	
OBRYSOVE OPRACOVANI:	
DESKY DODAVAT:	

Firma		Urceno pro firmu:	
NAZEV VYROBKU:		OZNACENI DOKUMENTU:	
VERZE:		DATEM:	
Format vykresu: Eagle 6.6		UPRAVNI: 1.2.2016 23:12:42	
UVRACOVAN:		MERITKO:	
SCHVALIL:		TISK: 1.2.2016 23:12:46	
		2:1	
		<b>navrh1 BRD</b>	