

České vysoké učení technické v Praze  
Fakulta elektrotechnická  
Katedra měření



Bakalářská práce

# Přenos biologických signálů pomocí bezdrátové technologie Bluetooth

Jiří Kubeš

Vedoucí práce: Ing. Jakub Parák

Studijní program: Kybernetika a robotika

Obor: Senzory a přístrojová technika

21. května 2014





## ZADÁNÍ BAKALÁŘSKÉ PRÁCE

Student: **Jiří Kubeš**

Studijní program: **Kybernetika a robotika**  
Obor: **Senzory a přístrojová technika**

Název tématu česky: **Přenos biologických signálů pomocí bezdrátové Bluetooth technologie**

Název tématu anglicky: **Transfer of Biological Signals via Wireless Bluetooth Technology**

### Pokyny pro vypracování:

1. Seznamte se s bezdrátovou technologií Bluetooth, mikroprocesorovou architekturou ARM Cortex a moduly vývojového telemonitoringového systému.
2. Vytvořte programovou knihovnu pro ovládání Bluetooth modulu pomocí mikroprocesoru s možností připojení k různým systémovým platformám a parametrům nastavení.
3. Vytvořte aplikaci pro jednoduché zpracování přenesených biologických signálů.
4. Realizujte přenos testovacích dat a spolehlivost připojení zařízení k různým systémovým platformám.

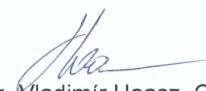
### Seznam odborné literatury:

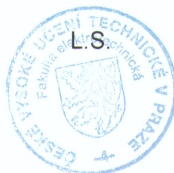
- [1] Joseph Yiu: The Definitive Guide to the ARM Cortex-M3, Oxford, 2007.
- [2] Havlík, J., Lhotská, L., Parák, J., et al.: A Modular System for Rapid Development of Telemedical Devices. Journal of Universal Computer Science (2013) 19(9): pp. 1242-1256.
- [3] Bluetooth IEEE Standard 802.15.1
- [3] Katalogové listy jednotlivých komponent a součástek.

Vedoucí bakalářské práce: Ing. Jakub Parák (K 13131)

Datum zadání bakalářské práce: 27. listopadu 2013

Platnost zadání do<sup>1</sup>: 30. ledna 2015

  
Prof. Ing. Vladimír Haasz, CSc.  
vedoucí katedry



  
Prof. Ing. Pavel Řípka, CSc.  
děkan

V Praze dne 27. 11. 2013

<sup>1</sup> Platnost zadání je omezena na dobu dvou následujících semestrů.

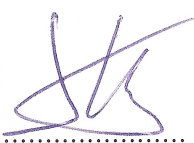


## Prohlášení

Prohlašuji, že jsem předloženou práci vypracoval(a) samostatně a že jsem uvedl(a) veškeré použité informační zdroje v souladu s Metodickým pokynem o etické přípravě vysokoškolských závěrečných prací.

Beru na vědomí, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorského zákona, ve znění pozdějších předpisů, zejména skutečnost, že České vysoké učení technické v Praze má právo na uzavření licenční smlouvy o užití této práce jako školního díla podle § 60 odst. 1 autorského zákona.

V Praze dne: 22.5.2014



Podpis



## Poděkování

Tímto bych chtěl poděkovat svému vedoucímu práce Jakobovi Parákovi za spolehlivé vedení, přínosné připomínky a maximální vstřícnost při organizování konzultací na dálku.

Dále bych rád vyjádřil vděčnost všem lidem, kteří si najdou čas, aby ostatním poradili s jejich problémy na internetových diskuzích. Díky jejich ochotě se značně zkrátilo řešení mnoha problémů, na něž jsem při práci narazil.





## **Abstrakt**

Tato práce popisuje vytvoření knihovny pro mikrokontrolery STM32X, která umožní těmto procesorům jednoduchou komunikaci s jinými zařízeními pomocí Bluetooth modulu a měření biologických signálů. Tento modul bude možno použitím jedné z těchto knihoven lehce nakonfigurovat a spojit se s dalšími zařízeními. Druhá knihovna zajistí měření až čtyř biologických signálů a jejich vyslání pomocí Bluetooth modulu. Další část této práce se zabývá vytvoření programu pro PC, který naměřené a přijaté hodnoty zpracuje, zobrazí na monitoru a uloží na pevný disk.

## **Abstract**

This thesis describes creating library for microcontrollers STM32 which allows these controllers to communicate easily with other devices using Bluetooth module and to measure biological signals. By using one of these libraries, Bluetooth module can be initialized and connected to another Bluetooth device. Second library provides measurement up to four biological signals and their transmission via Bluetooth. Next part of the thesis is implementing PC application which processes, displays and saves received data.

# Obsah

<b>I</b>	<b>Úvod</b>	<b>1</b>
<b>II</b>	<b>Teoretický úvod</b>	<b>2</b>
<b>1</b>	<b>Technologie Bluetooth</b>	<b>2</b>
1.1	Fyzická vrstva . . . . .	2
1.1.1	Vysílač . . . . .	2
1.1.2	Přijímač . . . . .	2
1.1.3	Modulace . . . . .	2
1.2	Verze technologie Bluetooth . . . . .	3
1.2.1	Verze 1.x . . . . .	3
1.2.2	Verze 2.x . . . . .	3
1.2.3	Verze 3.0 . . . . .	3
1.2.4	Verze 4.0 . . . . .	3
1.3	Základní protokoly technologie . . . . .	3
1.4	Zabezpečení . . . . .	4
1.5	Princip párování . . . . .	4
<b>2</b>	<b>Procesory ARM</b>	<b>4</b>
2.1	Vnitřní architektura . . . . .	5
2.1.1	Registry . . . . .	5
2.1.2	Instrukční sada . . . . .	5
2.1.3	ARMv7 . . . . .	6
2.1.4	Cortex – M3 . . . . .	6
2.2	Výrobci . . . . .	6
<b>3</b>	<b>Konečné stavové automaty</b>	<b>6</b>
3.1	Mooreův automat . . . . .	7
3.2	Mealeův automat . . . . .	7
3.3	Booleova algebra . . . . .	7
<b>III</b>	<b>Současný stav</b>	<b>9</b>
<b>4</b>	<b>Obecné využití bluetooth</b>	<b>9</b>

<b>5</b>	<b>Využití bluetooth v medicíně</b>	<b>9</b>
5.1	Bluetooth EKG . . . . .	10
<b>6</b>	<b>Obecné využití procesorů ARM</b>	<b>10</b>
<b>7</b>	<b>Využití procesorů ARM v medicíně</b>	<b>11</b>
7.1	Využití ARM pro EKG . . . . .	11
7.2	Umělé vidění pro slepé pacienty . . . . .	11
7.3	Detekce pádu . . . . .	12
<b>IV</b>	<b>Cíl projektu</b>	<b>13</b>
<b>V</b>	<b>Hardware</b>	<b>14</b>
<b>8</b>	<b>Bluetooth modul KC-Wirefree</b>	<b>14</b>
8.1	Syntaxe AT příkazů . . . . .	14
8.2	Používané AT příkazy . . . . .	15
<b>9</b>	<b>STM32</b>	<b>16</b>
9.1	A/D převodník . . . . .	16
9.2	DMA . . . . .	16
9.3	USART . . . . .	17
9.4	Čítače/časovače . . . . .	17
9.5	Paměť . . . . .	18
9.5.1	Vestavěná FLASH paměť . . . . .	18
<b>10</b>	<b>Vývojový kit</b>	<b>18</b>
10.1	STM32VLDISCOVERY . . . . .	18
10.2	STM32LDiscovery . . . . .	18
<b>11</b>	<b>EVO-Primer</b>	<b>19</b>
<b>12</b>	<b>Další použité moduly</b>	<b>20</b>
<b>13</b>	<b>Invertující spínaný zdroj</b>	<b>21</b>
<b>VI</b>	<b>Software</b>	<b>23</b>

<b>14 Konkrétní implementace stavového automatu</b>	<b>23</b>
14.1 Stavový automat pro inicializaci . . . . .	25
14.2 Stavový automat pro měření a komunikaci . . . . .	26
<b>15 Aplikace na straně příjmu</b>	<b>28</b>
15.1 Popis uživatelského rozhraní . . . . .	28
15.2 Popis funkce . . . . .	29
15.3 Zpracování signálů . . . . .	31
<b>VII Ověření funkčnosti</b>	<b>32</b>
<b>16 Ověření funkčnosti aplikace</b>	<b>32</b>
<b>17 Test filtrů</b>	<b>32</b>
17.1 Filtry EKG signálu . . . . .	32
17.2 Filtry PPG signálu . . . . .	34
17.3 Ověření algoritmu pro měření tepu . . . . .	34
<b>18 Komunikace s mobilním zařízením na platformě Android</b>	<b>35</b>
<b>VIII Závěr</b>	<b>37</b>
<b>Dodatek A: Obsah příloženého DVD</b>	<b>45</b>



# Část I

## Úvod

V dnešní době se začínají rozšiřovat zařízení monitorující vitální funkce lidského těla. Nejčastěji nacházejí využití u sportovců a v lékařství, kde se využívají jednak pro výzkum, jednak pro monitorování zdravotního stavu pacientů.

Existuje mnoho zařízení, která jsou schopna měřit biologické signály. Ve většině případů se ovšem jedná o drahé profesionální přístroje se spoustou funkcí a možností, nebo o levné jednoúčelové přístroje, jejichž výstup se nedá použít pro budoucí analýzu dat. Současným trendem je návrh levných, jednoduchých zařízení, která poskytnou co nejvíce funkcí, ale hlavně umožní data později analyzovat. Vzhledem k požadavku na co nejnižší cenu se většinou podrobná analýza dat provádí na univerzálních přístrojích typu PC ve specializovaném programu.

Trendem moderních přístrojů je nahrazování kabelů bezdrátovými rozhraními. Bezdrátová spojení disponují větší flexibilitou a mobilitou, mnohdy i výhodnější cenou. Jedním z nejpoužívanějších bezdrátových rozhraní současnosti je technologie Bluetooth, která nachází uplatnění v mnoha oborech lidské činnosti. Největší zastoupení má ve spotřební elektronice, poslední dobou však začíná rozšiřovat své uplatnění i do dalších oborů.

Práce je rozdělena na dvě části. První část se zabývá návrhem knihoven pro měření a odesílání signálů. Vzhledem k tomu, že se jedná o knihovnu, je žádoucí, aby fungovala na více typech mikrokontrolérů.

Druhá část práce je vytvoření jednoduchého programu na PC, který bude přijaté signály přijímat a zpracovávat.

Výsledkem této práce bude knihovna pro mikrokontroléry STM32, která měří biologické signály a bezdrátově je pomocí Bluetooth vysílá do PC. Pro PC bude napsán program, který bude implementovat algoritmy jednak pro komunikaci po virtuální sériové lince, jednak pro zpracování různých druhů signálů. Dále bude naměřený signál zobrazovat v grafu a ukládat pro pozdější analýzu. Z přijatých signálů program vypočte a zobrazí tepovou frekvenci.

## Část II

# Teoretický úvod

## 1 Technologie Bluetooth

Bluetooth je bezdrátová technologie pro přenos dat na krátké vzdálenosti. Pro přenos využívá krátkých vln o frekvenci přibližně 2440 MHz[13]. Tato technologie byla vyvinuta firmou Ericsson jako bezdrátová náhrada za kabelové rozhraní RS-232. Bluetooth je standardizován normou IEEE 802.15.1. Nejnovější verze BT je 4.0 z června roku 2010.

### 1.1 Fyzická vrstva

Bluetooth operuje v 2,4 GHz ISM pásmu, které není licencované. Přesné pásmo, které využívá je 2400 až 2483.5 MHz[13]. RF kanály jsou od sebe vzdáleny 1 MHz.

#### 1.1.1 Vysílač

Požadavky na výkonové úrovně antén jsou dány výkonovými skupinami[13], které jsou uvedeny v následující tabulce.

Výkon. třída	Maximální výkon	Minimální výkon	Dosah [m]
1	100 mW (20 dBm)	1 mW	100
2	2,5 mW (4 dBm)	0,25 mW	10
3	1 mW (0 dBm)	N/A	1

Tabulka 1: Třídy vysílačů Bluetooth

#### 1.1.2 Přijímač

Referenční citlivost přijímače je -70 dBm[13]. Skutečná citlivost je určena maximální chybovostí 0,1 % pro přenos na bitové úrovni. Skutečná citlivost by měla být nižší nebo rovna referenční. Jednotlivé kanály se navzájem ovlivňují, takže jsou pro ně definovány úrovně rušení pro různě vzdálené kanály. Kanály nesplňující tuto úroveň se nazývají spurious channels a může jich být maximálně 5.

#### 1.1.3 Modulace

Data se modulují pomocí GFSK, tedy Gaussovské frekvenční modulace[13]. Modulační index je mezi 0,28 a 0,35. Logickou jedničku je reprezentuje kladný přírůstek frekvence, logickou nulu záporný.



## 1.2 Verze technologie Bluetooth

Postupem času technologie BT procházela vývojem a přicházely nové verze. Ty jsou momentálně celkem 4.

### 1.2.1 Verze 1.x

Schváleno jako standard 802.15.1. Postupně se opravovaly chyby a zvyšovali rychlosti až na 721 kbps. Standardizováno rozhraní mezi radičem a hostitelem. Zaveden indikátor síly signálu či podpora pro nešifrované kanály. Jednotlivé verze byly zpětně kompatibilní.

### 1.2.2 Verze 2.x

Hlavní novinkou je bezpečné jednoduché párování SSP. Obsahuje rozšířené informace o zařízení EIR, které pomáhá lepšímu vyhledávání a tím i šetření energie. Díky technologii EDR je maximální rychlost 2 Mbps. Zpětně kompatibilní s v1.2.

### 1.2.3 Verze 3.0

Pomocí technologie HS (High-Speed) se zvýšila maximální teoretická rychlost až na 24 Mbps[3]. Toto zásadní zrychlení má na svědomí technologie PAL, která umožňuje použít jinou fyzickou vrstvu, například WiFi. S větší přenosovou rychlostí přichází i větší spotřeba energie. Ale díky větším prodlevám mezi vysíláním, kdy se modul uvede do režimu spánku, je spotřeba ve výsledku mnohem nižší.

### 1.2.4 Verze 4.0

V této verzi není snaha o zvyšování přenosové rychlosti, ale o optimalizování přenosu menších bloků dat, které nejsou příliš časté[4]. V předchozích verzích bylo velmi energeticky náročné udržovat spojení i při nulovém vytížení. Toto řeší právě nová verze, které se říká Smart. Zařízení ve verzi 4.0 se dělí na Smart Ready a Smart. Smart Ready jsou zařízení typu notebook, chytrý telefon, atd. Smart zařízení jsou periferie, které se připojují ke Smart Ready zařízením. Smart technologie zajišťuje, že po připojení Smart zařízení ke Smart Ready mohou tato zařízení zůstat spojená hodiny i dny a to s minimální spotřebou. U minulých verzí bylo energeticky výhodnější zařízení odpojit.

## 1.3 Základní protokoly technologie

**Bluetooth HCI** – Komunikace mezi hostem a radičem. Existuje více norem pro různé fyzické transportní vrstvy, např. USB nebo USART.

**L2CAP** – Multiplexní spojení mezi dvěma zařízeními pomocí různých protokolů vyšší úrovně. Poskytuje segmentaci a on-air pakety. V základním režimu poskytuje nastavitelné MTU až do výše 64 kB. Spolehlivost je dána opakovaným přenosem jedním kanálem a CRC kontrolou.

**LMP** – Řídí rádiové spojení mezi dvěma zařízeními. Je realizován na řadiči.

**RSCOMM** – Radiofrekvenční komunikace nahrazující kabelový protokol – vytváří virtuální sériový datový tok (emuluje RS-232).

**SDP** – Umožňuje zařízení zjistit, jaké služby poskytují ostatní zařízení.

## 1.4 Zabezpečení

Zabezpečení se provádí jak na aplikační, tak na linkové vrstvě. Metody pro šifrování i autentifikaci jsou v každém zařízení implementovány stejně. Na linkové vrstvě se využívá čtyř různých entit: adresa zařízení (48 bitů), privátní klíč ověřovací (128 bitů), privátní klíč šifrovací (8-128 bitů) a náhodné číslo (128 bitů). Privátní klíče jsou odvozeny při inicializaci a nejsou nikdy smazány. Šifrovací klíč je při autorizaci odvozen od ověřovacího. Narozdíl od ověřovacího klíče, šifrovací může mít různou délku – jednak z různých požadavků na šifrování, jednak se jedná o rezervu do budoucna. Náhodné číslo je generováno pseudonáhodným algoritmem a periodicky se mění. Pro zabezpečení se dále používá PIN. Některá zařízení jsou distribuována s pevným PIN. Jedná se o zařízení bez rozhraní potřebných k zadání PIN (např. handsfree). V ostatních zařízeních PIN zadávají uživatelé. PIN může být buď číselný nebo alfanumerický s kódováním (UTF-8). Všechny tyto uvedené klíče se používají ke generování linkového klíče, který sdílí dvě nebo více zařízení v síti. Linkový klíč může být čtyř typů (každý se hodí pro jinou aplikaci) a pro každý typ se k jeho odvození využívá jiných klíčů.

## 1.5 Princip párování

Vzhledem k tomu, že při každém připojení by bylo nepraktické zadávat znovu PIN, existuje párování. Na začátku procesu párování se vytvoří tajný klíč (PIN) společný pro obě zařízení, který se v obou zařízeních uloží. Při novém připojení je PIN ověřen. Jakmile je proces ověření dokončen, může se zahájit přenos dat. PIN může být z jednoho zařízení smazán a ve druhém stále existovat – PIN ve druhém zařízení pak není vázán na žádné zařízení a druhé zařízení se s tímto PIN nemůže připojit k prvnímu.

## 2 Procesory ARM

První ARM (Advanced RISC Machine) procesor byl vyroben v roce 1984. Už v té době měl 32 bitovou šířku slova. Firma ARM Limited, která procesor začala vyrábět, časem od

výroby ustoupila a začala prodávat licenci na výrobu jejich procesorů ostatním firmám. Od té doby se soustředí pouze na vývoj. Každý výrobce pak může jednotlivé verze modifikovat tím, že přidává různé periferie. Od roku 2011 existují ARM procesory se 64 bitovou architekturou.

## 2.1 Vnitřní architektura

Architektura ARM[21] procesorů je navržena tak, aby při zachování velkého výkonu byla co nejmenší spotřeba i velikost. ARM architektura má následující vlastnosti:

- Velký jednotný soubor registrů
- Load/Store architektura, ve které se data zpracovávají pouze v registrech, ne přímo v paměti
- Jednotná délka instrukcí
- Úplná kontrola ALU a shifteru
- Automatická inkrementace či dekrementace adres pro optimalizaci programových smyček
- Podmínečné vykonávání všech instrukcí

### 2.1.1 Registry

ARM procesory mají 31 univerzálních registrů. Vždy je vidět pouze 16 z nich, ostatní se využívají ke zrychlení vyhodnocení vyjímek. Banka 16 hlavních registrů je využívána neprivilegovaným kódem (nazývají se Uživatelské registry). Dva registry (14 a 15) z hlavní banky mají speciální funkce: linkový registr (Link Register) ukazující na další instrukci po závorce, a programový ukazatel (Program Counter), který lze využít jako ukazatel na instrukci následující dvě instrukce po právě vykonávané instrukci. Registr 13 je obvykle používán jako ukazatel zásobníku (Stack Pointer).

### 2.1.2 Instrukční sada

Instrukce v procesorech ARM se dají rozdělit na šest skupin. Většina datových instrukcí umožňuje nastavit příznakové bity (Negative, Zero, Carry a Overflow) podle výsledku operace. Téměř všechny instrukce mají 4 bitové podmínkové pole, které určuje, zda se podmínka vykoná či nikoliv. Jedna hodnota určuje, že se instrukce vykonává vždy, jedna hodnota se používá pro těch několik málo instrukcí, jenž podmíněčné vykonávání neumožňují. Zbylých 14 určuje podmíněčné vykonání instrukce.

### 2.1.3 ARMv7

ARMv7 – M [22] využívá Thumb-2 technologie uvedené v ARMv6T2, což přineslo zvýšení ARM a Thumb instrukčních sad. Tato technologie přinesla výrazné zlepšení pro nasazení ARM procesorů v systémech reálného času.

Klíčové vlastnosti jsou následující:

- Vysoce deterministické operace – cacheless, minimální zpoždění přerušení
- Určené pro vestavěné systémy – málo vstupně výstupních pinů, nové možnosti nastavení pinů
- Umožňuje velké nasazení v průmyslu
- Ladění a softwarové profilování pro událostmi řízené systémy

### 2.1.4 Cortex – M3

Procesory s jádrem Cortex – M3 [20] jsou navrhovány jako procesory s nízkou spotřebou, malým počtem vstupně/výstupních bran a velmi malou latencí přerušení. Cortex – M3 zahrnuje procesorové jádro, Nested Vector Interrupt Controller (NVIC) pro zajištění nízkých latencí přerušení, několik sběrnicových rozhraní, Memory Protection Unit (MPU) a řešení pro ladění programu.

## 2.2 Výrobci

- Apple – A series
- nVidia – Tegra
- STMicroelectronics – STM32Fx, STM32L1, STM32T, STM32W
- Samsung - Enyos
- Texas Instruments – Sitara, Hercules, OMAP, Tiva, Stellaris
- A mnoho mnoho dalších...

## 3 Konečné stavové automaty

Jedná se o matematický model řešení rozličných problémů. FSM má konečný počet stavů a v jednom konkrétním čase se může nacházet pouze v jednom stavu. Mezi stavy se přechází pomocí přechodů spouštěných pomocí spouštěcí podmínky. Principu FSM se využívá v několika různých oborech. Mimo elektrotechniku najdou uplatnění například umělá inteligence, biologie, komunikace nebo aplikace neuronových sítí.

### 3.1 Mooreův automat

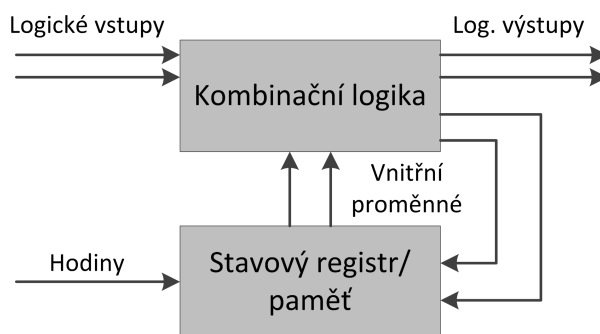
Výstup záleží pouze na současném stavu. Jeho hlavní výhodou je jednoduchost.

### 3.2 Mealeův automat

Výstup závisí jak na vstupech, tak na současném stavu. Je složitější, ale vede k redukci počtu stavů.

### 3.3 Booleova algebra

FSM lze také chápat jako sekvenční logické obvody, tedy logické obvody s pamětí. Zjednodušené schéma vypadá následovně.



Obrázek 1: Stavový automat chápaný jako logický sekvenční obvod

A právě pro návrh kombinačních logických obvodů je nezbytná znalost Booleovy algebry.

Booleova algebra [25] (podle George Boolea) je taková algebra, která popisuje množinové a logické operace. Pokud budeme brát  $\neg$  jako logickou negaci,  $\vee$  jako logický součet (sloučení) a  $\wedge$  jako logický součin (průnik), pak v Booleově algebře platí následující axiomy:

$$x \vee y = y \vee x \quad (1)$$

$$x \wedge y = y \wedge x \quad (2)$$

$$x \wedge 1 = x \quad (3)$$

$$x \vee 0 = x \quad (4)$$

$$x \vee (y \wedge z) = (x \wedge z) \vee (x \vee y) \quad (5)$$

$$x \wedge (y \vee z) = (x \wedge y) \vee (x \wedge z) \quad (6)$$

$$x \vee (\neg x) = 1 \quad (7)$$

$$x \wedge (\neg x) = 0 \quad (8)$$

Dále se dá odvodit, že platí následující pravidla:

$$x \vee 1 = 1 \quad (9)$$

$$x \wedge 0 = 0 \quad (10)$$

$$x \wedge x = x \quad (11)$$

$$x \vee x = x \quad (12)$$

PPG<sub>-</sub>

$$x \wedge (x \vee y) = x; \quad x \vee (x \wedge y) = x \quad (13)$$

$$\neg(\neg x) = x \quad (14)$$

$$\neg x \wedge \neg y = \neg(x \vee y) \quad (15)$$

$$\neg x \vee \neg y = \neg(x \wedge y) \quad (16)$$

## Část III

# Současný stav

V této kapitole se pokusím přiblížit to, jak se v současné době využívá technologie BT a mikropočítačů ARM se zaměřením na zdravotní techniku.

## 4 Obecné využití bluetooth

Technologie BT má dnes velmi široké zastoupení [8] díky své jednoduchosti. Nejčastěji se s ní lze setkat ve spotřební elektronice, kde se používá například pro headsety, bezdrátové audio systémy či bezdrátové propojení myší, klávesnici a gamepadů s PC a konzolemi. V počítačové technice se dá dále využít pro připojení tiskárny, grafických tabletů či různých dálkových ovládaní.

Dalším důležitým odvětvím, které využívá Bluetooth je automobilový průmysl. Pomocí Bluetooth se dá propojit automobil s mobilním telefonem, což přináší mnoho možností. Umožňuje to například přijímat hovory palubním počítačem, využívat mobilní internet pro GPS navigaci či přehrávat hudbu z mobilního telefonu.

Bluetooth se využívá i v inteligentních budovách. Používá se ke komunikaci senzorů s řídicími jednotkami a komunikaci řídicích jednotek mezi sebou. S použitím BT se tak dají jednoduše automatizovat domy, například otevírání garážových vrat, větrání, klimatizace, hlídání novorozenců či šetření energiemi.

Kromě vyjmenovaných oblastí se Bluetooth uplatní i v mnoha dalších odvětvích. Jedním z nich je i lékařství, o čemž pojednává následující kapitola.

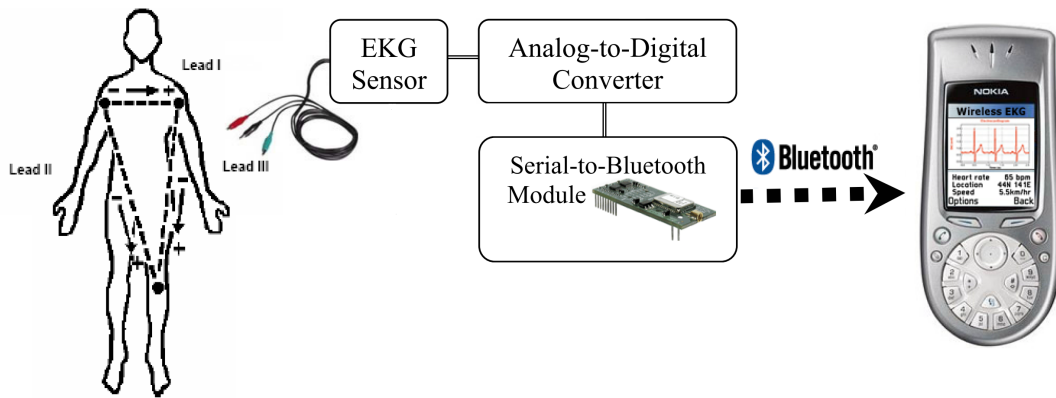
## 5 Využití bluetooth v medicíně

Bluetooth technologii v medicíně využívá přes 40 milionů zařízení[8].

Jedním ze zařízení, které využívají BT, jsou stetoskopy. Lékař jednak poslouchá srdce jako u klasického stetoskopu, ale zároveň se data posílají do PC či mobilního telefonu, kde se zobrazí a mohou lékaři pomoci s přesnější analýzou.

Dále se dá použít v měřicích hladiny cukru. Měřič průběžně měří hladinu cukru, kterou posílá do jiného zařízení (např. mobilní telefon). Toto zařízení pak hodnoty zaznamenává a umožní lékaři zpětnou analýzu.

Veškerá data, která naměří lékařské přístroje s BT, se dají posílat na zařízení disponující softwarem pro zpracování těchto dat. Může se jednat o speciální jednoúčelové aplikace naprogramované přímo pro dané účely nebo skripty například v Matlabu.



Obrázek 2: EKG s využitím Bluetooth

## 5.1 Bluetooth EKG

Jako příklad konkrétního medicínského zařízení, které využívá Bluetooth pro přenos dat, může sloužit projekt[26] svým zaměřením podobný tomuto. Autoři se pokusili vytvořit zařízení měřící pacientovo EKG a to pak posílat do mobilního telefonu. V mobilním telefonu se data budou ukládat nebo budou pomocí GPRS sítě dále odesílána přímo specialistům. Vzhledem k relativně nízké propustnosti GPRS sítě (řádově desítky kb/s) se EKG signál vzorkuje frekvencí 125 Hz. A/D převodníku má rozlišení 14 bitů, packet s jedním vzorkem má bitů 24. Cesta signálu je znázorněna na obr. 2.

## 6 Obecné využití procesorů ARM

Šíře využití procesorů a mikrokontrolérů s jádrem ARM je obrovská. Nachází uplatnění téměř v každém oboru, ve kterém je zapotřebí nějakého kontroléru. Díky zaměření na vysoký výkon a nízkou spotřebu se hojně využívají ve vestavěných systémech a bateriově napájených aplikacích.

Existuje velké množství procesorů s ARM jádrem a každý se hodí pro různé aplikace. Například procesory s důrazem na vysoký výkon a bezpečnost se hodí pro automobily, průmysl či lékařská zařízení. ARM procesory určené do mobilních telefonů se zase kromě vysokého výkonu zaměřují i na spotřebu energie. Jiné procesory se zaměřují na podporu systémů reálného času. Tyto procesory najdou využití v lékařství, komunikacích a bezdrátových sítích, audio a videotechnice, automobilech či průmyslu.

Jednou z mála oblastí, kam ARM procesory ještě výrazně nepronikly, je výroba procesorů pro osobní počítače, respektive notebooky. Existuje několik málo ultrabooků s ARM procesorem, jde ovšem pouze o velmi malou část.



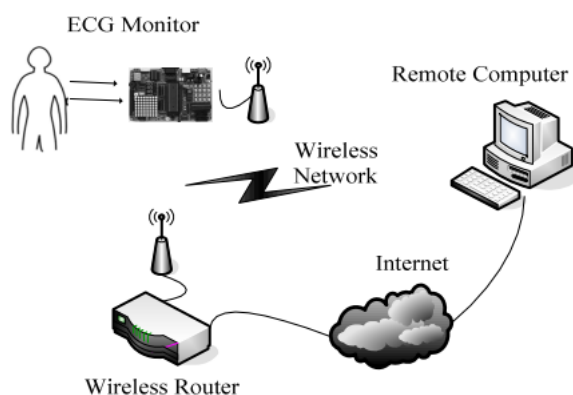
## 7 Využití procesorů ARM v medicíně

Díky obrovskému rozšíření mají nutně ARM procesory zastoupení i v medicíně. V posledních letech se stále více výzkumných pracovníků soustředí na různá monitorovací zařízení, pro která jsou díky své malé spotřebě procesory ARM přímo předurčeny.

### 7.1 Využití ARM pro EKG

V současné době je jednou z typických aplikací ARM procesorů EKG. Příkladem takové aplikace může být práce Jing Lianga a Yinqin Wua[28]. Ti ve svém zařízení využili dvoujádrový procesor OMAP, v němž se nachází jádra ARM a DSP. Jádro DSP má na starosti real-time část jako například analýza EKG signálu. O vše ostatní (například přenos dat přes WiFi, konfigurace) se stará jádro ARM.

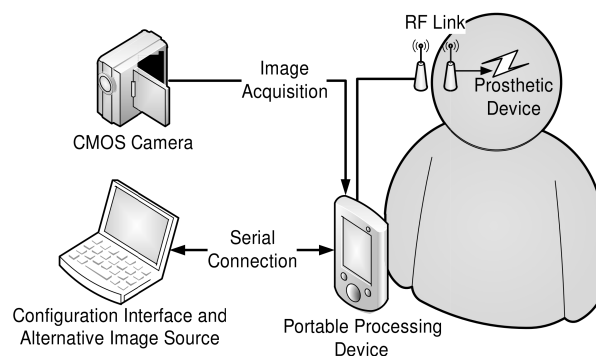
Princip zařízení je znázorněn na obr. 3. EKG je snímáno senzory z lidského těla a poté digitalizováno pomocí A/D převodníku. Po konverzi se provede ještě potřebná manipulace s daty potřebná pro vyslání. Pro kontrolu systému je využit operační systém Embedded Linux pro jeho kvality v plánování procesů či dobré podpoře řadičů zařízení. Vzdálený klient dostává data prostřednictvím internetu. Z přijatých dat se pak ve speciálním softwaru rekonstruuje průběh EKG a analyzují případné problémy.



Obrázek 3: Architektura monitorovacího systému EKG[28]

### 7.2 Umělé vidění pro slepé pacienty

Jednou z mnoha dalších aplikací ARM procesorů je zpracování obrazu pro systémy nahrazující oči slepým pacientům[?]. Na obr. 4 je konkrétní aplikace[?]. CMOS kamera snímá obraz do procesní jednotky. Zde se obraz uloží a bezdrátově vyšle do zařízení s 98 elektrodami implantovanými v suprachoroidální části oka. Procesní jednotka pak může být nakonfigurována z externího přístroje pomocí sériového rozhraní. Pro výrobu tohoto zařízení byla nejvíce limitující rychlost přenosu dat. Zařízení by mělo být schopné snímat



Obrázek 4: Implementace umělého vidění[?]

okolí tak, aby byl obraz plynulý, z čehož plyne minimálně 20 snímků za vteřinu - jedná se tedy o velký datový tok.

Mozkem celého zařízení je Mini2440. Jedná se o mini počítač osazený procesorem S3C2440 ARM9 od firmy Samsung. Hlavní důvody pro výběr Mini2440 byly rychlost procesoru (400 MHz), malá velikost a hlavně port pro CMOS kamery přímo na čipu. Díky tomuto portu může sběr obrazových dat probíhat mnohem větší rychlostí než s klasickým USB rozhraním.

Jako kamera byla zvolena kamera Omnicision OV9650 z důvodu hardwarové kompatibility, dostupných řadičů a možností snímat 60 snímků za vteřinu v rozlišení 320x240 pixelů.

### 7.3 Detekce pádu

V poslední době naše populace značně stárne. S rostoucím věkem se motorické schopnosti postupně zhoršují, z čehož vyplývá zvýšený počet vážných úrazů starších lidí způsobených pádem. V této souvislosti nabývají na důležitosti zařízení schopná pád detekovat a přivolat pomoc.

Touto problematikou se zabývá tento zajímavý projekt[27]. Autoři schovali veškerou elektroniku do pulloveru (svetru). Samozřejmostí pak je, že svetr lze klasicky vyprat a elektronika zůstane nepoškozena. Omyvatelnost pulloveru je zajištěna tím, že je elektronika umístěna ve voděodolném pouzdru Polar Electro Oy. V ostatních ohledech funguje zařízení podobně jako většina ostatních zabývajících se detekcí pádu. Jeden akcelerometr se připevňuje na hrud', druhý pak na paži. Oba akcelerometry jsou tříosé. Signál z akcelerometrů je vzorkován frekvencí 20 Hz, filtrován Butterworth filtrem a poté je odečteno gravitační zrychlení. Informace o pohybu se schoromažďují bezdrátově a ukládány na paměťové kartě k pozdější analýze.

## Část IV

# Cíl projektu

Hlavní téma této práce je navržení a naprogramování knihovny implementující algoritmy pro měření biologických signálů a jejich bezdrátové vysílání do PC či mobilních zařízení (smartphone, tablet). Cílem je realizovat takový způsob, který by přinesl velkou přesnost a spolehlivost společně se solidní rychlostí přenosu. Jako fyzická vrstva slouží technologie Bluetooth. Knihovna by měla umožňovat komunikaci se zařízeními na různých platformách (např. osobní počítače s operačními systémy Windows XP, Windows 7 či Linux nebo mobilní telefony a tablety s operačním systémem Android). Zařízení používající tuto knihovnu by mělo být schopné se samo inicializovat a spojit s vybraným zařízením s minimálním uživatelským zásahem. Velký důraz je kladen na univerzálnost, tedy aby veškeré principy komunikace fungovaly na všech platformách stejně.

Pro bezpečné splnění Nyquistova kritéria rekonstrukce signálů byla zvolena vzorkovací frekvence 1 kHz. Ta je dostačující pro signály typu EKG či PPG, které jsou v této práci použity jako demonstrativní. Přenosová rychlost musí být taková, aby se všechna naměřená data stihla přenést a zároveň taková, aby nedocházelo k chybám při přenosu. Z toho důvodu se jeví optimální přenosová rychlost kolem 1 Mb/s.

Dalším cílem této práce je vytvořit software pro PC, jenž bude zobrazovat naměřená data v reálném čase. Data budou zobrazována v grafu, kde bude vidět přesný průběh naměřeného signálu EKG i PPG. Dále bude z dat vypočten tep, a to jak ze signálu EKG, tak PPG. Přijatá data bude program ukládat v rozumném formátu na disk pro pozdější analýzu.

## Část V

# Hardware

Tato kapitola obsahuje hlavní HW prvky a jejich popis.

## 8 Bluetooth modul KC-Wirefree

Tento modul je vhodný jako náhrada kabelové sériové linky (RS-232) v aplikacích, které potřebují propojení bod-bod. Modul se ovládá pomocí jednoduchých AT příkazů[17, 18]. S modulem se komunikuje pomocí rozhraní UART, a to jak 3 vodičově, tak 5 vodičově (kromě vodičů GND, TX a RX jsou zde i CTS a RTS). Mezi jeho hlavní výhody patří řízení spotřeby energie. Podle dokumentace zvládá komunikovat rychlostí až 3 MB/s.

### 8.1 Syntaxe AT příkazů

AT příkaz ve verzi 2.4 začíná prefixem *AT+ZV*. Ve verzi 3.0 se pro AT příkaz na začátku místo *AT+ZV* píše pouze *AT*. Odpovědi začínají *AT-ZV*, resp. *->*. Za tímto prefixem následuje mezerou oddělený AT příkaz, resp. odpověď. Poté opět mezerami oddělené parametry. AT příkaz může mít parametrů více, ale nemusí mít žádný.

Ukázka podoby AT příkazu Bond ve verzi 2.4:

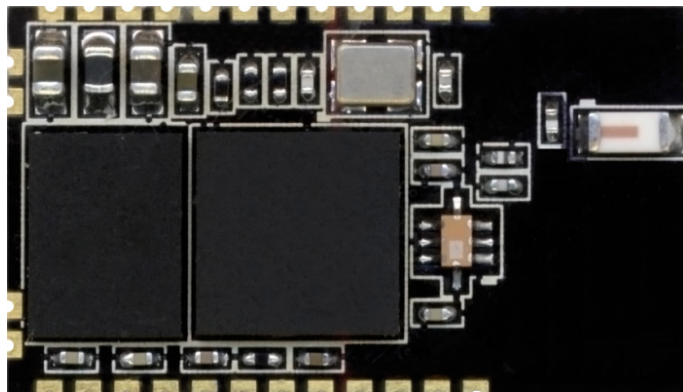
```
AT+ZV Bond [bdaddr] [pin].
```

*AT+ZV* je prefix; *Bond* název příkazu; *bdaddr* je první parametr, konkrétně adresa párovaného zařízení; *pin* je druhý parametr, konkrétně PIN nutný pro spárování.

Ukázka stejného příkazu pro verzi 3.0 (příkaz má jiné jméno):

```
AT Pair <btaddr>.
```

*AT* je prefix; *Pair* název příkazu; *btaddr* je parametr, adresa párovaného zařízení.



Obrázek 5: Modul KC-Wirefree

## 8.2 Používané AT příkazy

Tato kapitola obsahuje seznam používaných AT příkazů s krátkým popisem.

Příkazy jsou uvedeny pro v2.4, která se aktuálně používá. Ve verzi 3.0 zůstala většina příkazů zachována, některé mají změněné jméno nebo mají upravené parametry. Některé z příkazů byly v nové verzi rozděleny.

**AT+ZV Bond** – používá se ke spárování s určitým zařízením. Tento příkaz nevytvoří spojení, pouze uloží nezbytné informace do své párovací tabulky. Ve verzi 2.4 má dva parametry: adresa párovaného zařízení a PIN. Ve verzi 3.0 se jmenuje Pair a má parametr pouze jeden a to adresu, PIN se používá defaultní pro dané zařízení.

**AT+ZV Bypass** – nastaví kcSerial do bypass módu, pokud je právě spojený s jiným zařízením.

**AT+ZV ChangeBaud** - mění přenosovou rychlost sériové linky. Změna se provede až po odeslání odpovědi. Má jeden parametr, a to přenosovou rychlost v baud/s. V verzi 2.4 jsou možné hodnoty pevně dané, ve verzi 3.0 libovolné.

**AT+ZV EraseBondTable** – smaže informace o všech spárovaných zařízeních. Ve verzi 3.0 PairingDelete.

**AT+ZV LocalName** – změní jméno BT modulu. Má jeden parametr, a to budoucí jméno. Ve verzi 3.0 se jmenuje Name.

**AT+ZV Reset** – resetuje modul.

**AT+ZV Security** – povoluje použití zabezpečení. To je v defaultním nastavené zakázané. V obou používaných verzích má jeden parametr. Ve verzi 2.4 má hodnoty E (enabled) nebo D (disabled), kdežto ve verzi 3.0 si lze vybrat z úrovní zabezpečení. Hodnoty odpovídající jednotlivým úrovním jsou 0=žádné, 1=nevynucené, 2=Standardní, 3=pouze autentifikované.

**AT+ZV SPPConnect** – využívá se pro připojení k určitému zařízení. Po úspěšném připojení modul automaticky přechází do Bypass módu. Ve verzi 2.4 má jeden povinný parametr, a to adresu vzdáleného zařízení. Ve verzi 3.0 se příkaz jmenuje Connect a má dva nepovinné parametry. Prvním z nich je opět adresa, která se nemusí uvádět. V tomto případě se využije adresa posledního připojeného zařízení. Druhým parametrem je RfComm kanál nebo UUID. Pokud nebude vyplněn, využije se poslední hodnota z posledního připojení.

**AT+ZV UpdateInquiryScan** – upravuje viditelnost pro ostatní zařízení a délku vyhledávání. Ve verzi 3.0 je rozdělen na příkaz Discoverable a InquiryScan. Má tři parametry (první parametr přešel k příkazu Discoverable, zbylé dva k InquiryScan). První parametr určuje viditelnost modulu. Další zadává délku scanu od 18 do 4096 (defaultně 18). Poslední udává čas mezi scany ve stejném rozmezí (defaultně 2048).

**AT+ZV UpdatePageScan** – určuje, zda se může zařízení připojit k jinému. Parametry má stejné jako předchozí příkaz, pouze první neudává, zda je zařízení viditelné, ale připojitelné. Stejně jako minulý AT příkaz se v novější verzi rozdělil na dva příkazy a to **Connectable** a **ConnectScan**. **AT+ZV Version** – vrací informace o použitém FW.

## 9 STM32

Následující kapitola obsahuje krátký popis použitého procesoru a použitých periférií[24, 23]. Vzhledem k tomu, že knihovna je psána pro dva různé procesory, jsou u jednotlivých periférií uváděny informace společné pro oba procesory. Pokud se některé zásadní věci liší, jsou rozdíly uvedeny.

### 9.1 A/D převodník

V obou mikropočítačích je 12 bitový převodník s postupnou aproximací[24, 23]. Rodina STM32F100xx může měřit 18 multiplexovaných zdrojů signálu (16 externích, 2 interní). Rodina STM32Lxx má 42 multiplexovaných kanálů, může měřit 40 externích zdrojů signálu a 2 interní. Převod může být prováděn ve 4 různých módech - jednorázový odběr, kontinuální, přerušovaný či scan mód. Převody se vždy provádí v maximální rychlosti pro zajištění maximální vzorkovací rychlosti. Automatická kontrola spotřeby zajišťuje velké energetické úspory díky zapnutí AD převodníku pouze na čas nezbytný pro převod.

Lze využít funkce analogového watchdogu, který hlídá, zda se napětí nedostalo za uživatelem zvolené meze.

AD převodník nabízí tři různá rozlišení - 8, 10 a 12 bitů. Lze volit mezi zarovnáním výsledku doleva nebo doprava. Každý kanál má nastavitelný vzorkovací čas nezávislý na ostatních kanálech. Mezi jednotlivá měření jde zařadit automatické programovatelné hardwarové zpoždění. Měření lze spouštět pomocí externí spouště.

Může generovat přerušení několik druhů přerušení (End of Conversion, End of Injected conversion, Analog Watchdog Event). Převodník může sám sebe zkalibrovat.

### 9.2 DMA

Používá se pro vysokorychlostní přenos dat mezi dvěma místy, kterými může být paměť nebo nějaká periferie. Hlavní výhodou DMA je to, že přenos dat probíhá automaticky bez jakéhokoliv zásahu CPU, což ho volňuje pro jiné operace. Oba procesory mají dva DMA kontroléry, které mají dohromady 12 kanálů. Každému z kanálů může být nastavena softwarová priorita o čtyřech různých úrovních, kromě toho mají ještě hardwarovou prioritu.

Přenos se realizuje po systémové sběrnici, který DMA kontrolér sdílí s CPU. Díky tomu může dojít k zablokování přístupu CPU ke sběrnici po několik sběrniceových cyklů. Samotný přenos probíhá tak, že po nastalé události vyše periferie žádost pro DMA kontrolér obsluhující tyto žádosti v závislosti na nastavených prioritách. Jakmile DMA kontrolér přistoupí k periférii (paměti), vyše potvrzení. Jakmile periferie dostane potvrzení, uvolní žádost. Po uvolnění žádosti DMA kontrolér uvolní potvrzení. Pokud má periferie více žádosti, může iniciovat další přenos.

### 9.3 USART

Poskytuje klasickou plně duplexní komunikaci (synchronní i asynchronní) s kódováním NRZ[24, 23]. Kromě full-duplex kanálu podporuje i jednosměrnou komunikaci a poloduplexní jednovodičovou komunikaci. Podporuje také LIN rozhraní nebo SmartCard Protocol. Toto rozhraní lze využít pro multiprocessorovou komunikaci. Pro zvýšení přenosových rychlostí je možné využít DMA s multibufferovým nastavením.

Rychlost přenosu lze nastavit až do 4,5 Mb/s, délku slova na 8 nebo 9 bitů, 1 nebo 2 stop bity. Má oddělené povolovací bity pro vysílač přijímač, lze tedy používat pouze jeden a druhý mít vypnutý.

Pro USART existuje deset možných zdrojů přerušení. Může detekovat změnu na vodiči CTS, mezeru na LIN, prázdný vysílací registr, dokončení vysílání, plný přijímací registr, klidový stav na vodiči a 4 chybové stavy.

USART detekuje 4 různé druhy chyb – předběhnutí, detekci šumu, chybu rámce, chybnou paritu. Všechny tyto chyby mohou generovat přerušení.

Pro vzorkování signálu na vstupu RX se využívá metody Oversampling. Je tomu proto, aby bylo jednodušší oddělit užitečný signál od šumu. Oversampling[5] je metoda, při níž se signál vzorkuje frekvencí mnohonásobně vyšší, než je Nyquistova frekvence.

Pokud je USART vysílač zakázán, pak se pin TX nachází ve stavu odpovídajícím jeho vstupně výstupnímu nastavení. Pokud je povolen a nevysílá, pak se nechází ve stavu logické jedničky (vysoká úroveň). Pokud USART operuje v jednovodičovém módu, pak je tento pin používán i pro příjem dat.

### 9.4 Čítače/časovače

Tento mikroprocesor má celkem 17 čítačů/časovačů, které se dělí na několik skupin podle toho, co umí a k čemu jsou určeny. Všechny čítače jsou 16 bitové. Některé z nich mají jednotky Input capture, Output compare nebo One-pulse mode output. Dá se pomocí nich lehce generovat PWM.

V tomto projektu se používá pouze TIM6, který spolu s TIM7 patří do skupiny Basic timers, jenž (jak název napovídá) umějí pouze základní funkce. Perioda čítání se dá měnit

programovatelnou děličkou, která může dělit jakýmkoliv číslem v rozmezí 1-65535. Čítač je samo obnovovací, to znamená, že po každém přetečení se nemusí manuálně nastavovat hodnota čítače, ale automaticky se nahraje hodnota z pre-load registru. Na rozdíl od ostatních skupin čítačů/časovačů generuje pouze jednu událost, a to přetečení. Tato skupina lze ještě využít jako synchronizační obvod pro D/A převodník.

TIM6 a TIM7 jsou naprosto nezávislé, nesdílejí žádné registry.

## 9.5 Paměť

V obou rodinách jsou paměť programu, datová paměť, registry apod. organizovány do stejného lineárního 4 GB prostoru. Adresovatelná paměť se dělí do 8 bloků po 512 MB. Bajty jsou v paměti ukládány ve formátu little endian[9]. To znamená, že na nejmenší adrese slova se ukládá bajt s nejnižší váhou, na nejvyšší adrese pak ten s nejvyšší váhou.

### 9.5.1 Vestavěná FLASH paměť

Maximální hustota je 512 kB. Paměť se dělí na hlavní blok a informační blok, jejichž velikost záleží na typu procesoru. FLASH rozhraní obsahuje ochranu proti čtení i zápisu. Hlavní blok může sloužit k emulaci EEPROM paměti. K tomu se využívá (u používaných procesorů) rozsah paměti 0x08080000 až 0x08082FFF.

## 10 Vývojový kit

Pro vývoj knihoven byly použity dva vývojové kity, STM32VLDISCOVERY[31] a STM32LDiscovery [32]. Oba kity mohou být napájené jak z USB, tak z externího zdroje 5 V nebo 3,3 V. Z kitů lze napájet aplikaci 5 V nebo 3 V. Na obou kitech jsou dvě uživatelské LED (modrá a zelená), uživatelské tlačítko a resetovací tlačítko. Procesory na obou deskách jsou zaměřené na nízkou spotřebu. Jinak se oba kity liší. Oba kity jsou na obr. 6.

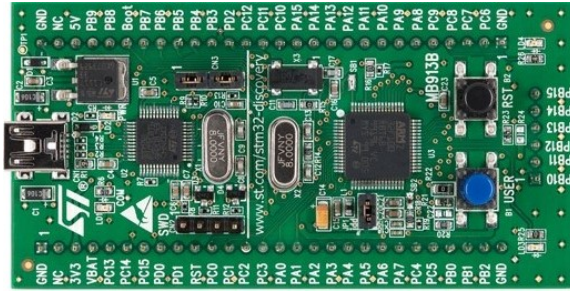
### 10.1 STM32VLDISCOVERY

Tento kit[31] je osazený mikrokontrolérem STM32F100RB se 128 kB Flash pamětí a 8 kB RAM paměti. Na desce je ST-Link debugger, který lze využít i pro ladění programu externích mikrokontrolerů.

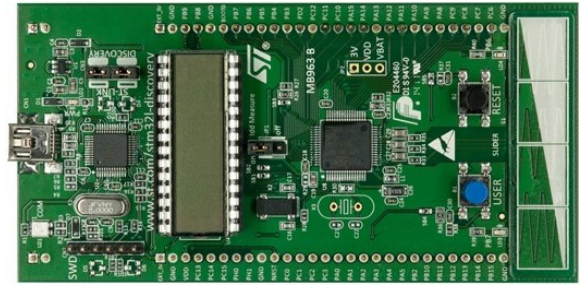
### 10.2 STM32LDiscovery

Na kitu[32] je mikrokontrolér STM32L152RBT6 se 128 kB Flash memory a 16 kB RAM. Tento procesor má v porovnání s předchozím propracovanější řízení spotřeby, což při





(a) STM32VLDISCOVERY kit



(b) STM32LDISCOVERY kit

Obrázek 6: Vývojové kity

správném využívání ještě více snižuje spotřebu. Stejně jako STM32VLDISCOVERY obsahuje ST-Link debugger, na tomto kitu je v nové verzi. Oproti STM32VLDISCOVERY je na kitu navíc LCD v pouzdru DIP28, jeden lineární dotykový senzor a čtyři dotyková tlačítka.

## 11 EVO-Primer

Po dokončení bude knihovna využívána v zařízení založeném na STMicroelectronics Evo-Primer Platform[14], konkrétně se jedná o třetí generaci této platformy. Jde o nástroj určený pro rapid-prototyping. Tato platforma může být osazena osmi nebo dvaatřiceti bitovými procesory od firmy STMicroelectronics. Tato generace zvyšuje flexibilitu a snižuje nároky na návrh tím, že poskytuje základní sadu levných vyměnitelných mikrokontrolérových desek.

Primer obsahuje následující vybavení:

- Vestavěné debugovací a programovací USB rozhraní pro PC
- Ergonomický design
- Barevný dotykový TFT displej
- 4 menu tlačítka
- Dvouosý joystick s tlačítkem
- Hardwarový audio kodek + jack konektor
- Třiosý MEMs akcelerometr
- Li-Ion baterie s nabíjecím obvodem
- Konektor pro Micro SD karty
- Konektor pro rozšiřující obvody s rozhraními USART, SPI, I2C a vstupně výstupními piny



Obrázek 7: STMicroelectronics EvoPrimer[12]

Osazením různými mikrokontroléry lze docílit různých dalších vlastností a vybavení. Například pro použití STM32F105/107 lze využít Mini-USB konektoru s On-the-Go[6] funkcionalitou, s mikrokontroléry STM32F407/417 přichází možnost využívání 1,3 megapixelového obrazového senzoru, tříosý gyroskop, 16 MB paměti typu SRAM či STA529A audio kodek. S některými mikrokontroléry lze využít senzor teploty, tlaku, rozhraní pro segmentové displeje, solární panely a Li-Ion baterie. V portfoliu nabízených mikrokontrolérů se nacházejí i ultralow power čipy, které vynikají svou velmi nízkou spotřebou.

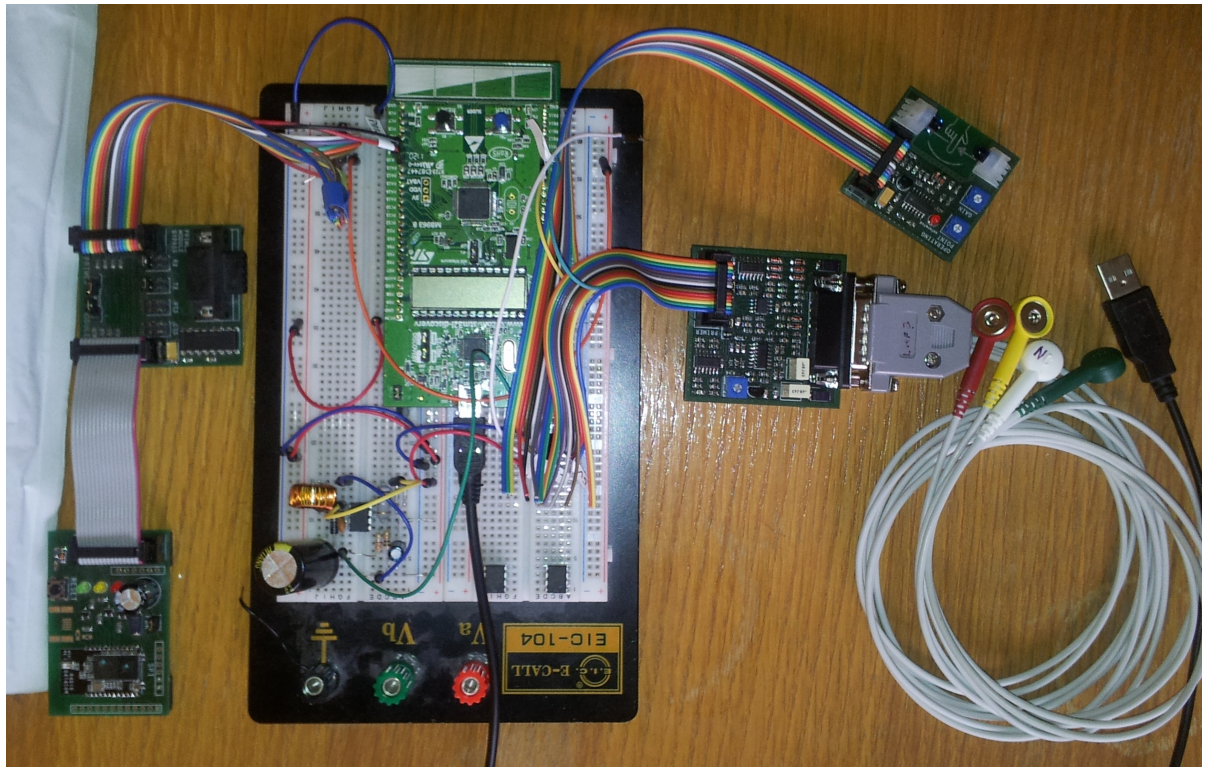
## 12 Další použité moduly

Kromě Bluetooth modulu, který se používá ke komunikaci, jsou potřeba ještě další moduly, a to pro sběr biologických dat. Pro demonstraci se používají moduly pro měření EKG a PPG[34].

EKG modul je navržen po vzoru profesionálních EKG senzorů. EKG seignál měří z elektrod IIa III. Tvoří ho diskrétné součástky, obsahuje filtr potlačující šum. Navíc má trimr, kterým lze nastavit posun signálu.

PPG modul je navržen tak, aby měřil signál podle standartního Nellcor senzoru. Modul obsahuje dva trimry pro nastavení posunu a zesílení.

Vývojový kit s moduly a spínaným zdrojem jsou na obr. 8.



Obrázek 8: Připojení modulů k testovacímu kitu

### 13 Invertující spínaný zdroj

Vzhledem k tomu, že EKG modul obsahuje symetricky napájené operační zesilovače, bylo potřeba vytvořit invertující zdroj napětí z 5 V na -5 V. Díky velké stabilitě napětí, jednoduchosti a nízké ceně byl jako řídicí obvod vybrán MC34063. Na schématu (obr. Subfigure 9a) jsou vidět další součástky, které byly na stavbu zdroje potřeba. Jejich hodnoty byly vypočteny podle návodu[19] od firmy Texas Instruments.

Nejprve je potřeba spočítat poměr času vypnuté a zapnutí spínače podle vzorce:

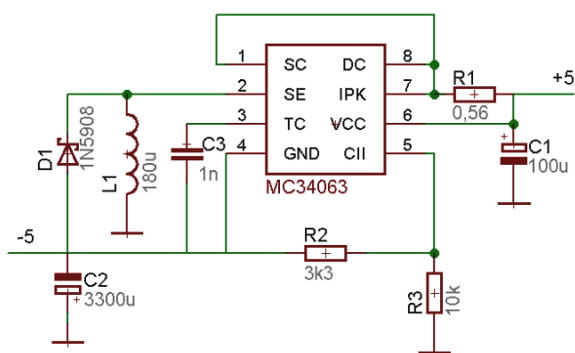
$$\frac{t_{on}}{t_{off}} = \frac{|U_{out}| + U_f}{U_{in} - U_{sat}} = \frac{5 + 0,75}{5 - 1,3} = 1,55 \quad (17)$$

$U_{out}$  a  $U_{in}$  jsou výstupní, resp. vstupní napětí,  $U_f$  je propustné napětí diody a  $U_{sat}$  je saturační napětí spínacího tranzistoru. Z výsledku pak plyne střída 61%. Minimální indukčnost cívky pro práci ve spojitém režimu se vypočte ze vztahu:

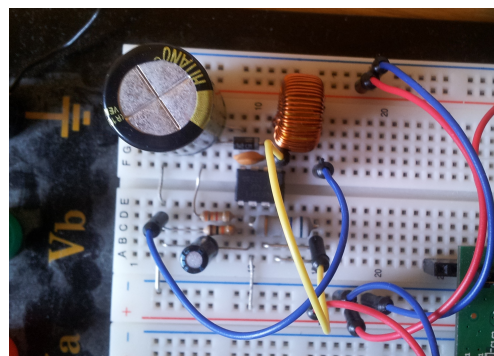
$$L_{min} = \frac{U_{in} - U_{sat}}{I_{peak}} t_{onmax} = \frac{U_{in} - U_{sat}}{2I_{out} \left( \frac{t_{on}}{t_{off}} + 1 \right)} \cdot \frac{s}{f} = \frac{5 - 1,3}{2 \cdot 0,1 \cdot (1,55 + 1)} 2,44 \cdot 10^{-5} = 180 \mu H \quad (18)$$

$I_{peak}$  je špičkový proud protékající cívkou,  $I_{out}$  je maximální stálý výstupní proud,  $s$  je střída a  $f$  je pracovní frekvence vnitřního oscilátoru. Výstupní proud byl zvolen 100 mA (dostačující hodnota pro napájení EKG modulu) a pracovní frekvence nastavena kondenzátorem 1000 pF na hodnotu 25 kHz.

Parametry výstupního kondenzátoru se dají spočítat pomocí požadovaného zvlnění výstupního napětí. U tohoto kondenzátoru není tak důležitá kapacita, jako parametr ESR (ekvivalentní sériový odpor), který by měl být co nejmenší, řádově setiny  $\Omega$ . Kondenzátory s takto nízkým ESR jsou v ČR těžko k sehnání a pokud ano, pak za cenu, která je několikanásobná v porovnání se zbytkem zdroje. Vysoká hodnota ESR se dá částečně kompenzovat zvýšením kapacity výstupního kondenzátoru (nejedná se o ideální řešení), případně vhodným zapojením tranzistoru.



(a) Schéma zapojení



(b) Fyzická realizace na nepájivém poli

Obrázek 9: Invertující spínaný zdroj s obvodem MC34063

## Část VI

# Software

Hlavním úkolem této práce je implementování komunikace mezi měřícím zařízením a zařízením zpracovávajícím tyto data.

Hlavní důraz je kladen na software pro mikropočítač. Pro jeho implementaci se využívá principu stavového automatu (viz. 3). Zařízení by mělo být schopné s minimálním uživatelským zásahem konfigurovat BT tranciever a připojit se k PC (mobilu/tabletu). Po navázání spojení je potřeba, aby zařízení vyčkalo, dokud nedostane signál k začátku měření. Dále se požaduje, aby měřilo data z více kanálů s rychlostí minimálně 1 kSa/s pro každý kanál (jedná se o vzorkovací rychlost na výstupu zařízení). Po neočekávaném ukončení spojení se nesmí v mikropočítači objevit chyba.

Na software na straně příjmu nebyly kladeny žádné zvláštní požadavky. Pouze měl být schopný zobrazovat přijaté hodnoty v grafu a zároveň je ukládat do souboru. Hodnoty by měly být ukládány v jednoduchém formátu vhodném pro pozdější zpracování.

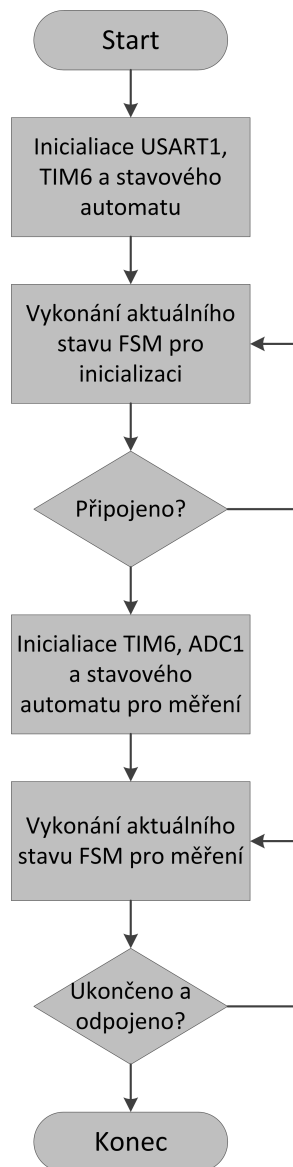
## 14 Konkrétní implementace stavového automatu

V této aplikaci jsou 2 různé implementace stavového automatu. Jedna slouží k inicializaci Bluetooth modulu a připojení k PC. V okamžiku připojení tento stavový automat ukončuje činnost a nastupuje druhý, který má na starost měření, odesílání dat a komunikaci s PC. Zařazení automatů za sebe je znázorněno na obr.10.

Implementace těchto stavových automatů je založena na práci bc. Jiřího Kubiase a ing. Michala Sojky, Ph.D. z katedry řídicí techniky. Veškerá data, která jsou potřeba pro činnost stavového automatu, se nacházejí v datové struktuře. Stavů reprezentuje funkce se dvěma parametry – ukazatelem na zmiňovanou strukturu a obsluhovanou událost. Informace o současném a minulém stavu je uložena ve struktuře pomocí ukazatele na funkci.

Běh stavového automatu řídí funkce, která zavolá funkci reprezentující současný stav a vykoná její hlavní událost. Pokud se během onoho jednoho průchodu stavem změní současný stav, pak řídicí funkce vykoná ukončovací událost současného stavu a vstupní událost nového stavu. Z toho plyne, že vstupní a ukončovací událost se vykoná vždy jednou, zatímco hlavní událost se vykonává do té doby, dokud nenastane tzv. podmínka přechodu.

Každý průchod stavem by měl být co nejrychlejší, z čehož plyne snaha o minimální využívání čekacích smyček – čekání je realizováno více průchody stavem, než jeho delším



Obrázek 10: Běh programu v mikrokontroléru

---

### Algoritmus 1 Řídící funkce stavového automatu

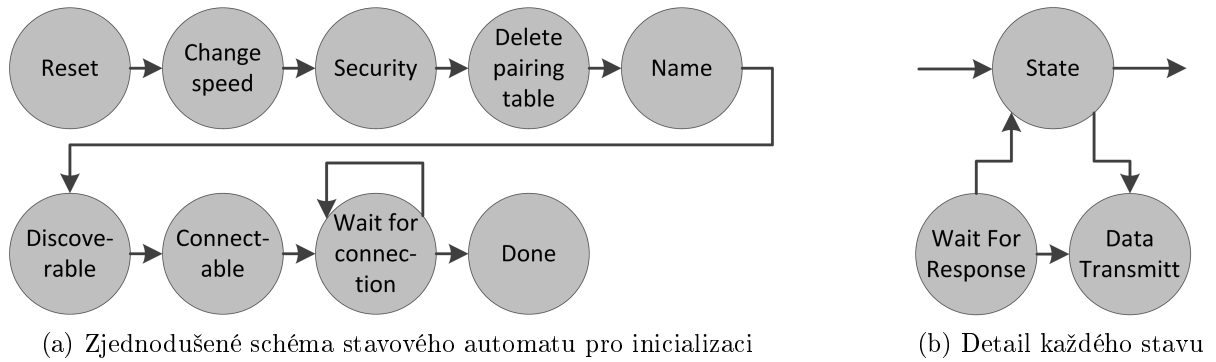
---

```

void initBluetooth(struct fsm *fsm) {
    while (fsm->currState != done) {
        fsm->lastState = fsm->currState;
        fsm->currState(fsm, DO);
        if (fsm->lastState != fsm->currState) {
            fsm->lastState(fsm, EXIT);
            fsm->currState(fsm, ENTRY);
        }
    }
}

```

---



Obrázek 11: Stavový automat pro inicializaci

vykonáváním. Například pro čekání na příjem dat ze sériové linky by se nevyužil příkaz `while(USART_GetFlag(USART1, USART_Flag_RXNE) == RESET)`, ale příkazu `if(USART_GetFlag(USART1, USART_Flag_RXNE) != RESET)` s tím, že pokud by tato podmínka nebyla splněna, stav se ukončí a při novém spuštění se podmínka kontroluje znovu.

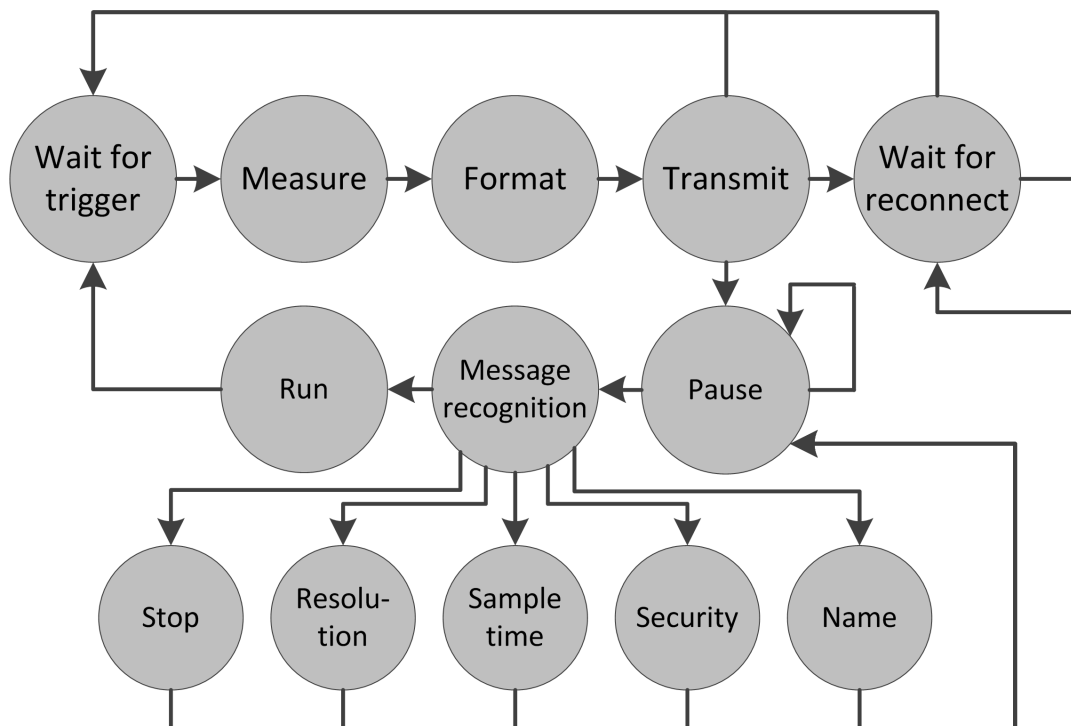
## 14.1 Stavový automat pro inicializaci

Jak už bylo zmíněno výše, Bluetooth modul se ovládá pomocí AT příkazů. Každý AT příkaz má vyhrazen jeden stav, respektive metodu. Zjednodušené schéma tohoto stavového automatu je na obr. 11a. Pro přehlednost do schématu nejsou zařazeny stavy `dataTransmit` a `waitForResponse`. Žádný stav totiž přímo neobsahuje mechanismy pro vysílání a přijímání zpráv, pro tyto účely existují zmiňované dva stavy. Na obr. 11b je vidět, jak funguje jejich volání.

Při obsluze vstupní (entry) události se zkopíruje text AT příkazu do řetězce ve struktuře s daty stavového automatu. Pokud AT příkaz obsahuje nějaký volitelný parametr (například volba zabezpečení přenosu), je tento parametr nejdříve načten z FLASH paměti a poté zakomponován do zprávy. Původně byla pro zakomponování parametrů využívána funkce `sprintf`, ale po zjištění, že používaný překladač GNU při použití této knihovny generuje až několikanásobně větší kód než např. překladač KEIL, bylo formátování pomocí `sprintf` zrušeno a nahrazeno knihovnou vytvořenou speciálně pro tento účel. Rozdíl ve velikosti přeloženého kódu je způsoben tím, že GNU nebyl (na rozdíl např. od zmiňovaného překladače KEIL) navrhován primárně pro vestavěné systémy a jeho `printf` knihovna je mnohem komplexnější.

V hlavní události se obvykle přepne do vysílacího stavu, který je následován stavem přijímacím. Po přijetí odpovědi se přepne do jiného stavu.

Obsluha ukončovací (exit) události má na starosti kontrolu chyb při příjmu a jejich řešení. Pokud se jedná pouze o špatně přijatý řetězec, je opakováno vyslání. Dalšími chybami může být např. nevykonání daného AT příkazu nebo navrácení špatného parametru.



Obrázek 12: Stavový automat pro měření

Tyto chyby mají pak individuální řešení v závislosti na stavu.

Pro tento stavový automat je připraveno několik stavů, jejichž posloupnost lze měnit v závislosti na požadovaném způsobu připojení. Prozatím se posloupnost musí měnit přímo v kódu, do budoucna je v plánu udělat postup připojování adaptabilní – aby se například pokusil připojit k poslední stanici, se kterou komunikoval a poté až vyčkal na připojení od jiných stanic.

## 14.2 Stavový automat pro měření a komunikaci

Tento stavový automat započne svou činnost v okamžiku, kdy se modul připojí k jinému Bluetooth zařízení a přejde do Bypass módu. Tento stavový automat obsahuje několik cyklicky se opakujících stavů. Kromě těchto pravidelně se opakujících stavů implementuje další, které mají za úkol vyhodnotit a odpovědět na AT příkazy zaslané z PC. Schéma stavového automatu je na obr. 12

Cyklicky opakujícími stavy jsou: čekání na spoušť, měření, formátování, přenos. V tomto pořadí jsou tyto stavy prováděny. Stav čekání na spoušť může být jako jediný přerušovaný druhou skupinou příkazů, tedy těch, které mají za úkol odpovídat na příkazy z PC.

Čekání na spoušť je velmi jednoduchý stav, který čeká, dokud nemá časovač 6 určitou hodnotu. Poté přejde do stavu měření. V souladu s požadavkem na co nejkratší průchod není čekání řešeno smyčkou, ale neustálým vykonáváním stavu.

Stav měření provede měření vybraného kanálu. Měření může být více z důvodu zvýšení přesnosti průměrováním. Pro počet měření na jeden vzorek může být zvolena pouze



mocnina čísla 2, protože je pak jednodušší a rychlejší dělení. Po naměření potřebného počtu vzorků a spočtení průměru přechází automat do stavu formátování.

Ve stavu formátování se naměřená data naformátují do řetězce, který se v následujícím stavu odesílá. Data jsou formátována v pevném formátu tak, aby extrahování informace z řetězce bylo co nejjednodušší. Formát je následující:

XXXX;XXXX;XXXX;XXXX,

kde XXXX je 12 bitové číslo zapsané v desítkové soustavě. Každé ze 4 čísel oddělených středníkem reprezentuje jeden kanál. První se používá pro PPG, druhý pro EKG, zbylé dva zatím nejsou využité.

Přenos zajišťuje vyslání dat do PC. Po vyslání dat se opět stavový automat vrací do stavu, kdy čeká na spoušť. Odesílá se celkem 19 znaků plus LF, tedy dohromady 20 znaků. USART je nastavený na 1 stop bit a žádnou paritu, tedy jeden znak má celkem 10 baudů. Vysílací čas jedné zprávy se tedy rovná

$$T_v = nN \frac{1}{f_v} \quad (19)$$

kde  $n$  je počet baudů jednoho znaku,  $N$  je počet znaků a  $f_v$  přenosová rychlost. Vzhledem k tomu, že  $T_v$  musí být nižší, než 1 ms (vzorkovací perioda), platí pro minimální přenosovou rychlost

$$f_{vmin} = nN \frac{1}{T_v} = 10 \cdot 20 \cdot \frac{1}{10^{-3}} = 2 \cdot 10^5 \text{ baud} \quad (20)$$

Defaultní přenosová rychlost Bluetooth modulu (115,2 kb/s) tedy není dostačující pro přenos dat. Je proto tedy nutné použít vyšší rychlost - byla zvolena rychlost 460 800 baud/s, protože se jedná o maximální rychlost, při které modul ve verzi 2.4 zvládne komunikovat bez chyb.

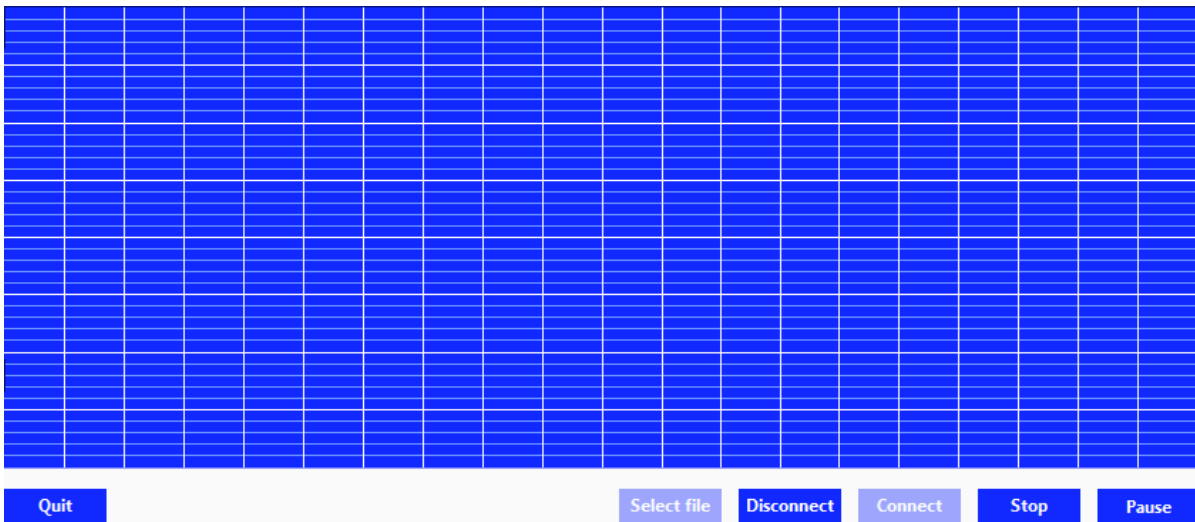
Vzhledem k malému prostoru pro přijímání zpráv a vysílání odpovědí, je možné přijímat AT příkazy pouze tehdy, bylo-li předtím měření pozastaveno příkazem *Pause*. Příkaz *Run* se poté měření opět spustí.

Pokud je přijata zpráva *Pause*, automat přejde do stavu, kdy čeká na další AT příkazy. Po přijetí příkazu se příkaz identifikuje a vykoná. Poté je zaslána odpověď. V případě, že byl přijat řetězec, který neodpovídá žádnému příkazu, je zasláno chybové hlášení.

Ve stavech vyhodnocujících AT příkazy se z řetězce dostane parametr, pokud nějaký obsahuje. Pak se s ohledem na parametr provede rutina daného příkazu. Příkazy jsou následující:

**CMD Run** – spouští měření.

**CMD Pause** – pozastavuje měření.



Obrázek 13: Hlavní panel aplikace v LabView

**CMD Stop** – ukončuje měření a odpojuje se od virtuální sériové linky.

**CMD Resolution** – pokud neobsahuje žádný parametr, pouze vrací hodnotu současného rozlišení. Pokud obsahuje číselný parametr s hodnotami rozlišení, které odpovídají rozlišení, jenž je kontrolér schopen poskytnout, pak dojde k přepsání informace o používaném rozlišení ve FLASH paměti a odeslán potvrzující řetězec. Pokud obsahuje jakýkoliv jiný parametr, je odesláno chybové hlášení.

**CMD Name** – změní jméno zařízení, které je uloženo v paměti. Změna se projeví až po restartu zařízení.

**CMD SampleTime** – funkce je obdobná příkazu Resolution s tím rozdílem, že vrací vzorkovací periodu v milisekundách.

**CMD Security** – nastavuje úroveň zabezpečení. Opět funkce obdobná s příkazem Resolution.

## 15 Aplikace na straně příjmu

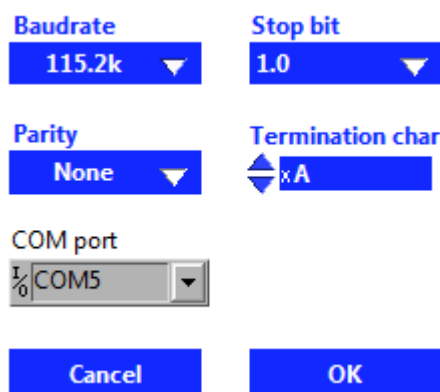
Vývojové prostředí LabView bylo zvoleno z několika důvodů: dostupnost licence na nejnovější vydání ve školní síti; zpracované knihovny, včetně těch pro sériovou linku; LabView je určeno pro vývoj aplikací pro sběr dat, kontrolu a řízení přístrojů či průmyslové aplikace.

### 15.1 Popis uživatelského rozhraní

Na obr. 13 je vidět hlavní panel aplikace. Největším a nejdůležitějším prvkem je graf, který zobrazuje naměřený průběh. Graf pouze zobrazuje hodnoty, kliknutím na něj nelze provádět změny či vyvolat dialogové okno.

V pravé dolní části se nachází 4 tlačítka. Tlačítko *Run* spouští měření tím, že odešle modulu příkaz *Run*. Po spuštění měření se změní funkce tlačítka - po kliknutí se vyše příkaz *Pause*, který měření pozastaví, ale nevypne. Od toho je tlačítko *Stop*, které vysílá stejnojmenný příkaz.

Další dvě tlačítka slouží k otevření či zavření komunikačního kanálu. Po stisku tlačítka *Disconnect* se nejen uzavře komunikační kanál, ale povolí se tlačítko *Connect* a zakáže se tlačítko *Disconnect*. Stiskem tlačítka *Connect* se spustí dialog sloužící k otevření komunikačního kanálu. Ten je zobrazen na obr. 14. Nastavuje se zde komunikační rychlost, parita, počet stop bitů a ukončovací znak zprávy. Poslední položkou je výběr sériového portu. Tlačítkem *OK* se připojí k vybranému sériovému portu a zavře se okno. Dále se v hlavním okně znemožní opětovný klik na tlačítko *Connect* a naopak umožní se stisk tlačítka *Disconnect*. Tlačítkem *Cancel* se zavře okno bez jakékoliv další akce. Pokud bylo stisknuto tlačítko *OK* a připojení k sériové lince bylo bez bych, následuje vyslání AT příkazů do zařízení pro zjištění používaného rozlišení A/D převodníku a vzorkovací periody. Až se napevno rozhodne, jaké bude mít sériová linka parametry, toto dialogové okno se nebude využít a po kliknutí na tlačítko *Connect* se aplikace připojí k sériovému portu.



Obrázek 14: Dialog sloužící k připojení k sériové lince

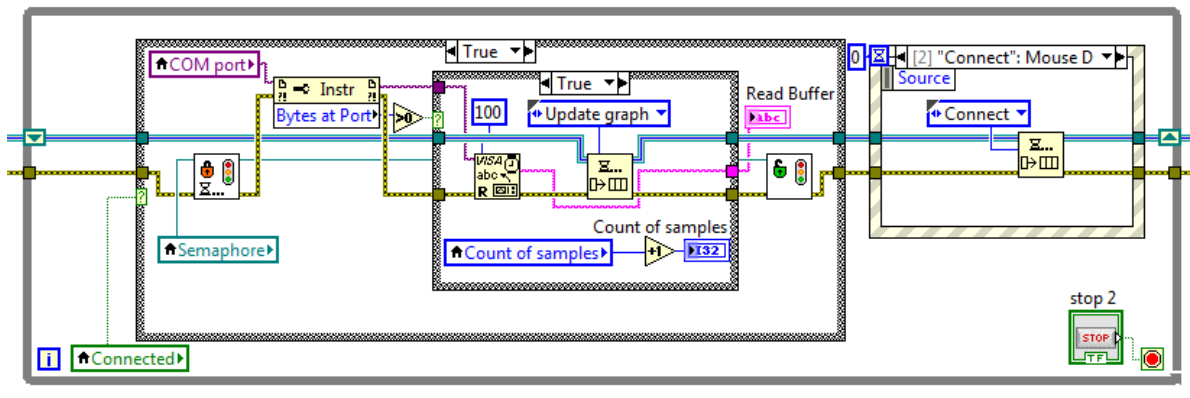
Tlačítko *Open file* zobrazí dialog pro vybrání nebo vytvoření souboru, do kterého se budou ukládat naměřené hodnoty. Po otevření či vytvoření souboru запиše na konec (v případě vytvoření souboru na začátek) řetězec se současným datem a časem.

Ukončovací tlačítko *Quit* před ukončením běhu programu odpojí sériovou linku a uloží a zavře soubor.

## 15.2 Popis funkce

Vzhledem k tomu, že tato aplikace je určena pro sběr dat v reálném čase, je potřeba zajistit, aby žádná data nebyla ztracena například důsledkem obsluhování prvků uživatelského rozhraní. Jedním z možných řešení tohoto problému je rozdělit aplikaci na více

## Algoritmus 2 Hlavní vlákno (producent)



vláken. V tomto konkrétním případě stačí dvě - jedno pro sběr dat a druhé pro obsluhu uživatelského rozhraní.

LabVIEW je implicitně více vláknové. To znamená, že pokud v programu budou dvě větve, které nesdílí žádné proměnné, jejich vykonávání proběhne paralelně, tedy jako dvou vláken. Tohoto principu lze u některých jednodušších aplikací úspěšně využívat, ovšem v této aplikaci bylo potřeba sdílet některé proměnné a prostředky, a proto je postavená na principu Producent/Konzument.

Producentem, a tedy hlavním vláknem, je v této aplikaci vlákno měřící, které sbírá data ze sériové linky. Zdrojový kód hlavního vlákna je vidět v algoritmu/obrázku 2. Vlákno je řešeno jako smyčka `while`, která je ukončena stiskem tlačítka *STOP*. Do vlákna vstupují dvě proměnné - fronta (modrá linka) a speciální cluster (hnědá linka), který obsahuje informaci o chybách. U fronty jsou vidět na vstupu a výstupu značky Shift registeru, které zajišťují, že se při každém novém průchodu použije fronta v takovém stavu, který měla na konci minulého průchodu.

Pokud je PC připojený k Bluetooth modulu ( $ConnecProdukemted == 1$ ), pak se provede kód v této podmínkové struktuře, který je vidět v algoritmu2. V opačném případě se neprovede nic. Pokud podmínka platí, program zkontroluje semafor *Semaphore*, který je volný v případě, že vedlejší vlákno nepřistupuje k sériové lince. V okamžiku, kdy je volný, ho toto vlákno uzamkne a k sériové lince může přistupovat pouze toto vlákno. Po uzamčení program vyčte počet bytů ve vstupním bufferu. Pokud není prázdný, je jeho obsah zapsán do řetězce Read Buffer a do fronty událostí je přidána událost pro překreslení grafu. Paralelně (protože tyto procesy nesídlí žádné proměnné) k tomuto se o jedničku inkrementuje počítadlo vzorků. Pokud je buffer prázdný, neprovede se nic. Poté je semafor opět uvolněn. Nakonec je zkontrolováno, zda některý prvek uživatelského rozhraní nevygeneroval událost. Pokud ano, je do fronty přiřazena příslušná událost.

Vedlejší vlákno má na starosti obsluhu událostí, které hlavní vlákno zařadí do fronty. Jedná se o události *Stop*, *Run/Pause*, *Connect*, *Disconnect*, *Update graph*, *Open file* a *Quit*. Všechny události kromě *Update graph* obsluhují stisk tlačítek, jejichž činnost je popsána v kapitole Popis uživatelského rozhraní. Událost *Update graph* nastane při příjmu dat

---

**Algoritmus 3** Ukázka dat uložených v souboru

---

```
Záznam z 10.5.2014 21:42:32
0062;0743;0000;0000
0063;0805;0000;0000
0065;0735;0000;0000
0067;0694;0000;0000
0065;0794;0000;0000
0065;0820;0000;0000
0061;0725;0000;0000
0062;0726;0000;0000
0064;0822;0000;0000
0065;0801;0000;0000
0063;0739;0000;0000
```

---

ze sériové linky. Z přijmutého řetězce jsou extrahovány hodnoty jednotlivých kanálů a přepočteny na napětí. Tyto hodnoty jsou před zobrazením ještě filtrovány, na pevný disk jsou však ukládány hrubá data, tedy ještě před filtrací.

### 15.3 Zpracování signálů

Zpracování signálu se provádí v rámci události *Update graph*. Z přijatého řetězce jsou extrahovány hodnoty pro jednotlivé kanály. Po přepočtení na napětí se signály filtrují pomocí Butterworth filtru použitým jako pásmová propust. Filtr EKG propouští frekvence v rozmezí 7 až 17 Hz, filtr PPG v rozmezí 0,3 a 3 Hz.

Tepová frekvence se z EKG zjišťuje pomocí měření časového intervalu mezi tzv. R-špičkami. Ty jsou detekovány pomocí prahové hodnoty, která je odvozena ze střední hodnoty signálu. Signál musí prahovou hodnotu přesáhnout alespoň na 5 ms. Je to doba dostatečně krátká na to, aby jsi jak PPG signál, tak EKG signál byl schopný zaručeně překročit, a zároveň doba dostatečně dlouhá na to, aby ji nepřekročil žádný šum, který pronikne přes filtr. Za detekci špiček je zařazen VI, který detekuje pouze náběžné hrany vstupního signálu. Ty určují časové intervaly mezi začátky špiček, které se měří.

Pokud byl zvolen soubor pro ukládání dat, přijaté signály se do něj ukládají. Signály se ukládají nefiltrované, protože pro další zpracování mohou být zajímavá jiná frekvenční pásma než ta propuštěná filtrem. Data jsou ukládána ve formátu, v jakém byla přijata, viz 14.2. Ukázka dat uložených v souboru je vidět v alg. 3.

## Část VII

# Ověření funkčnosti

## 16 Ověření funkčnosti aplikace

Aplikace byla převážně testována na notebooku ASUS N61VG16 s 64 bitovým dvoujádrovým procesorem Intel T6600 pracujícím na kmitočtu 2,2 GHz a operační paměti 4 GB.

Všechna tlačítka plní požadovanou funkci, vzájemné zakazování jednotlivých tlačítek funguje podle předpokladů. Dlouhá odezva tlačítka *OK* v připojovacím dialogu je způsobena časem potřebným pro spojení PC a Bluetooth modulu a inicializačním dialogem mezi nimi. Enumerační tlačítko pro výběr sériového portu není ve stejném designu jako zbytek aplikace, protože je to přímo součást VISA knihoven, do kterých se nelze dostat.

Ukládání dat do souboru funguje spolehlivě, při zápisu nedochází k žádným chybám. Otevírání či vytváření souboru probíhá bez chyb. Pro zavírání souboru bylo přidáno ošetření pro případ, kdy se uživatel pokouší zavřít soubor, i když žádný není otevřený.

Při testování různých filtrů bylo zjištěno, že v závislosti na další činnosti PC je ztracena (nepřijata) část dat. Objem ztracených dat byl v řádu jednotek až několika desítek procent. Při velkém vytížení procesoru i operační paměti přesahovali ztráty dat i 25%. Na výkonějším PC (lepší procesor a rychlejší RAM paměť) se ztráty snížily, nedosáhly ovšem nuly. Použitím procesoru s více jádry (nebo s funkcí Hyperthreading) by se ztráty měli snížit na nulu. K zajištění nulových ztrát by vedlo i snížení vzorkovací rychlosti, ale požadavkem bylo vytvoření knihovny, která umožní mikroprocesoru vzorkovat rychlostí až 1kHz. Pro praktické užití se ovšem běžně využívá i vzorkovacích frekvencí kolem 250 Hz.

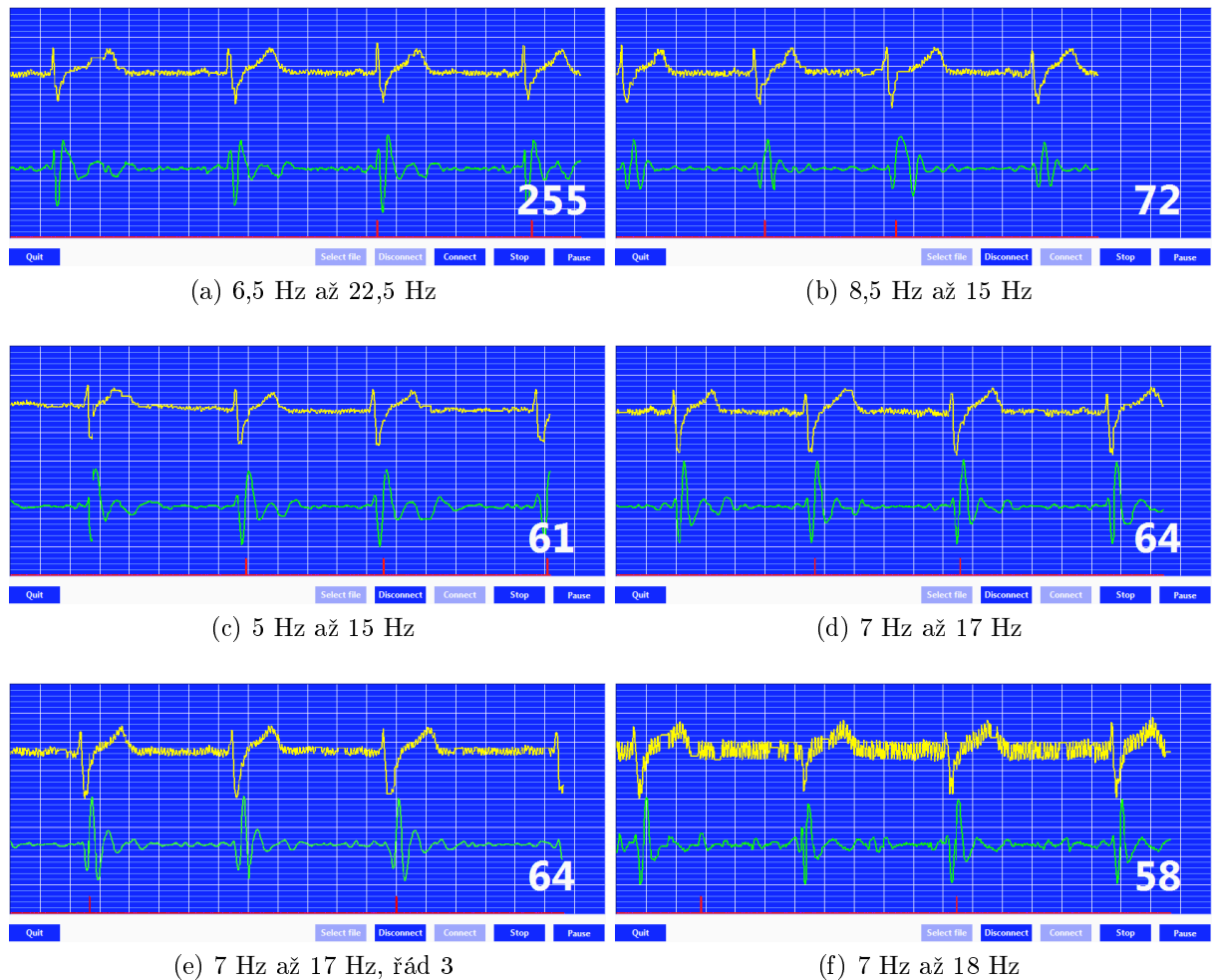
Byla otestována poloviční vzorkovací rychlost, tedy 500 Hz, při které byly ztráty minimální, v řádu desetin procent.

## 17 Test filtrů

V této části je porovnání funkčnosti několika filtrů pro každý typ signálu.

### 17.1 Filtry EKG signálu

V EKG signálů jsou nejdominantnější tzv. QRS celky[33], ze kterých se dá zjistit velké množství informací. EKG signál bývá velmi často zarušen z několika různých zdrojů najednou, například síťovým napájením, změnou rozhraní na přechodu elektrody a kůže,



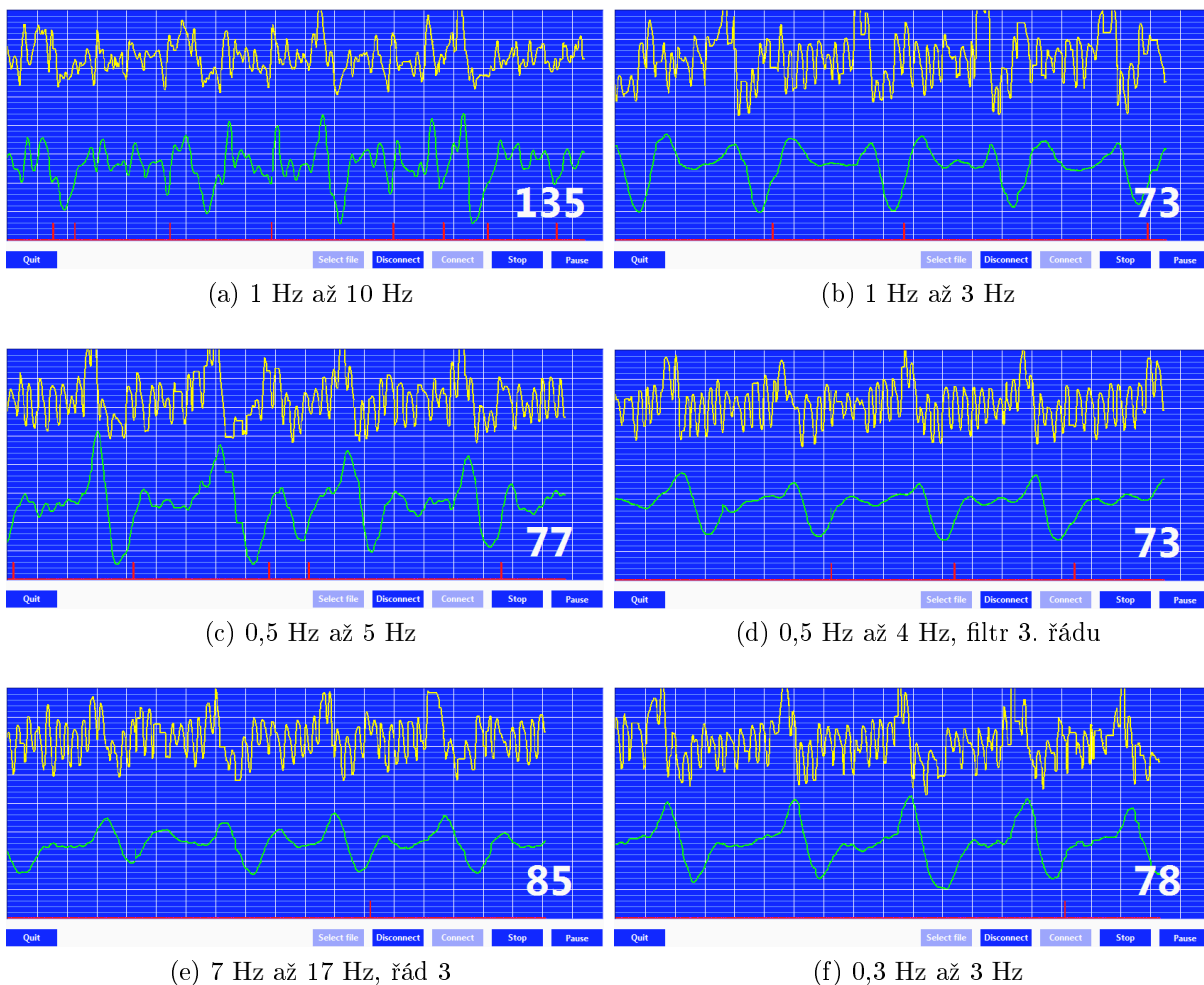
Obrázek 15: Různé typy filtrů EKG signálu

pohyb svalů apod. Tyto rušivé elementy je potřeba ze signálu dostat. Vzhledem k tomu, že většina energie QRS bývá v pásmu kolem 10-15 Hz[33](některé zdroje uvádějí pásma 5 - 12 Hz, některé například 8 - 40 Hz), signál bude filtrován pásmovou propustí, konkrétně Butterworthovým filtrem. Na obrázcích 15 jsou vidět průběhy EKG signálu před a po filtraci s ponechanými různými pásmy. Pokud není uvedeno jinak, jedná se o Butterworthovy filtry 2. řádu.

Všechny filtry zobrazené na obr. 15 dokáží z přijatého zarušeného signálu dostat QRS celky. Nejlépe se jeví filtry od 7 Hz do 17 Hz. Filtr druhého řádu rychleji reaguje na změny než filtr třetího řádu, naopak filtr třetího řádu má hladší průběh.

Na obrázku 15a je nejhorší z šesti zobrazených filtrů. Na výstupu filtru se objevují v každém QRS dvě velké kladné špičky způsobující špatnou detekci špiček a tím i výpočet tepu. Obě kladné špičky algoritmus detekuje a krátký čas mezi nimi způsobí, že výsledný tep je nesmyslně vysoký. Proměnná obsahující výsledný tep je reprezentována jako 8 bitové číslo, což omezuje maximální hodnotu tepu na 255.

Obr. 15f zobrazuje, co se stane se signálem, pokud některá z elektrod má horší kontakt (alespoň nějaký, ne žádný).



Obrázek 16: Různé typy filtrů PPG signálu

## 17.2 Filtry PPG signálu

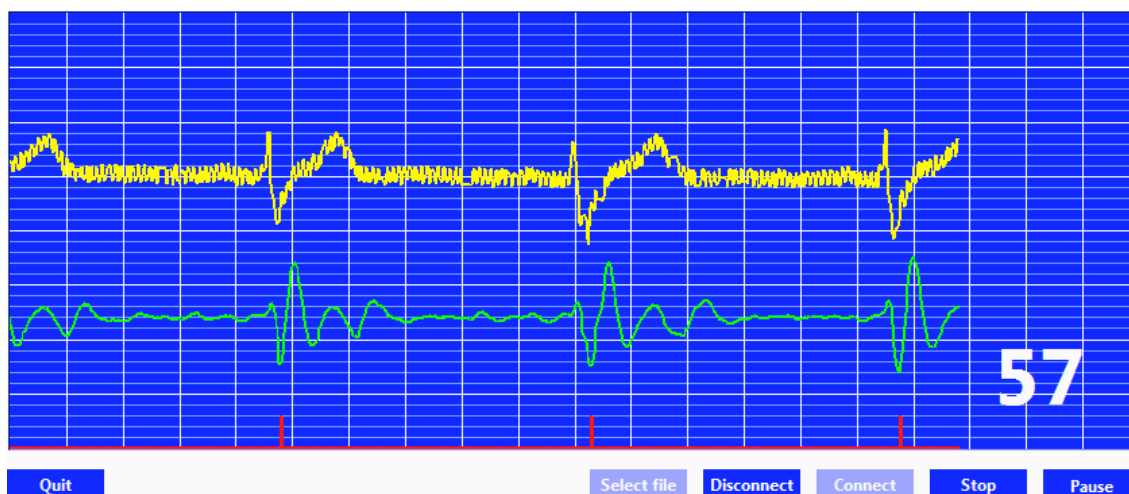
PPG signál je stejně jako EKG rušen a musí být filtrován. U modulu měřícím PPG signál je kromě klasického rušení sítovou frekvencí důležitý další faktor, a to konstrukce modulu, respektive umístění lineární optické závory. Jednak je modul otevřený, do snímací diody tedy svítí i denní světlo (resp. osvětlení). To se dá z větší části vyřešit zakrytím modulu jiným předmětem. Druhým problémem je rozptyl světla při odrazu od prstu, případně při vycházení světla z prstu. Pokud se jedná o uzavřené PPG, většina světla projde do prstu a neodrazí se od něj pryč, případně se může ještě odrazit zpět od PPG do prstu.

U PPG je zajímavé pásmo v řádu jednotek Hz[30]. Proto se při experimentálním hledání ideálního filtru experimentovalo se spodními frekvencemi kolem 0,5 Hz a horními kolem 5 Hz.

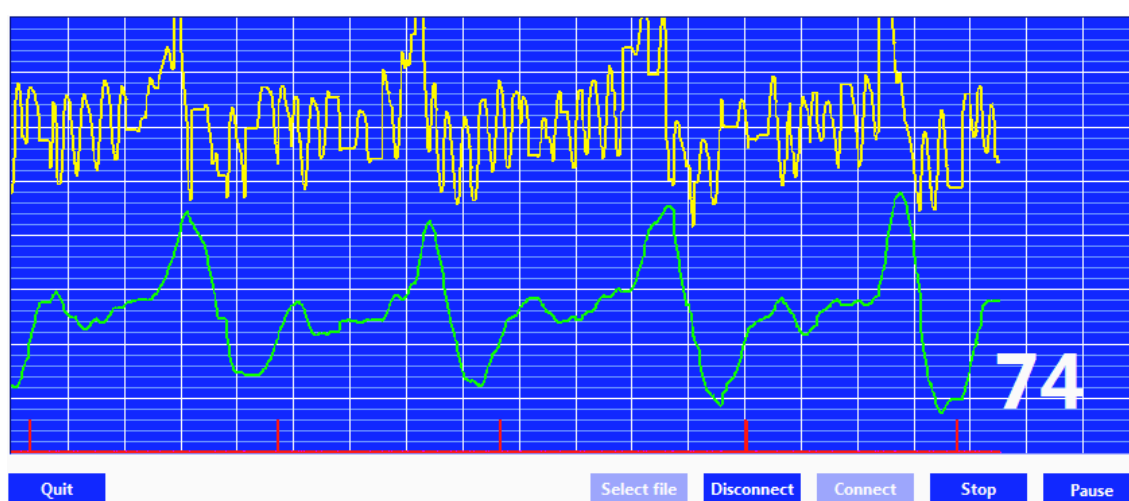
## 17.3 Ověření algoritmu pro měření tepu

Na obrázku 17 jsou vidět jak filtrované, tak nefiltrované signály EKG a PPG. Na obou obrázcích je dole v grafové oblasti červený průběh, který ukazuje časové okamžiky, kdy





(a) Detekce špiček a výpočet tepu ze signálu EKG



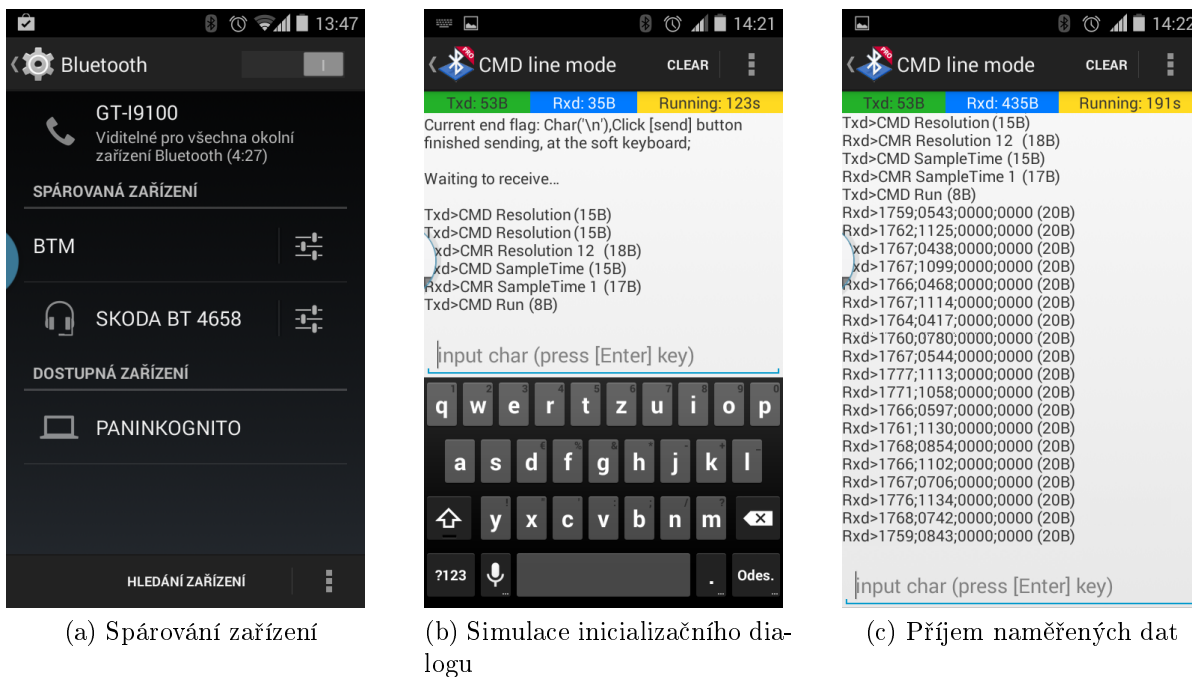
(b) Detekce špiček a výpočet tepu ze signálu PPG

Obrázek 17: Detekce špiček a výpočet tepu

byla detekována špička signálu. Při správně navrženém filtru a vhodně zvolené prahové hodnotě algoritmus spolehlivě detekuje špičky v signálech a měří tep. Na grafech nemusí být vždy příznak detekce hrany vidět, protože v grafová oblast má přibližně 800 pixelů na šířku, zatímco v grafu se zobrazuje přes 1000 bodů na jeden kanál, takže grafická karfa některé špičky vynechává.

## 18 Komunikace s mobilním zařízením na platformě Android

Pro tyto zařízení nebyl vytvořen žádný program, ovšem zařízení využívající vytvořenou knihovnu musí být schopná se k nim připojit. Pro ověření schopnosti bez problému komunikovat byl použit program *Bluetooth spp tools pro* (obr. 18b a 18c), který je zdarma dostupný na Google Play. Tato aplikace je přímo určená k testování Bluetooth komunikace.



Obrázek 18: Ukázka komunikace se zařízením na platformě Android

Připojení a komunikace byla testována na mobilním telefonu Samsung i9100 s operačním systémem ParanoidAndroid 4.2-BETA3 (ekvivalent Androidu 4.4.2 KitKat) a jádrem DorimanxV10.43 a Sony Xperia ST26i s operačním systémem Android 4.1.2 JellyBean se sériovým jádrem verze 3.4.0. U obou zařízení proběhlo připojení i komunikace úspěšně. Lze tedy tvrdit, že navržený postup připojování funguje pro Android ve verzi 4.0 a vyšší. Pro verze androidu 1.x či 2.x nebylo připojení testováno. Pro zajímavost bylo připojení testováno na mobilním telefonu Nokia Lumia 520 s operačním systémem Windows Phone 8 (verze 8.0.10517.150). Připojení bylo úspěšné, komunikace testovaná nebyla z důvodu nedostupnosti terminálu pro Bluetooth.

Pro připojení je použita stejná posloupnost příkazů jako pro připojení k PC s Windows. Po spárování (obr. 18a) je pouze potřeba otevřít vytvořený kanál. Po otevření kanálu (a přepnutí modulu do Bypass módu), přichází na řadu inicializační dialog, který je simulovaný na obr. 18b. Po odeslání příkazu *Run* se spustí měření a přijímají se data 18c v definovaném formátu (viz 14.2). Senzory byly při vysílání dat připojené k A/D převodníkům, ovšem EKG ani PPG senzor nemonitoroval žádného pacienta.

## Část VIII

# Závěr

Cílem této práce bylo vytvoření knihoven pro mikrokontroléry STM32, které budou obsluhovat komunikaci po Bluetooth a vzorkování signálů na vstupu interního A/D převodníku.

V hardwarové části byly vytvořeny konektory pro jednotlivé moduly a navržen zdroj -5 V kvůli potřebě symetrického napájecí operačních zesilovačů v EKG modulu. Byl zvolen spínaný zdroj s univerzálním obvodem MC34063. Zapojený bylo realizováno na nepájivém poli.

Softwarová část se dělí na dvě části. V první části byly navrženy dva stavové automaty pro mikrokontroléry STM32. První z nich zajišťuje inicializace Bluetooth modulu a spojení se zařízením, jemuž má data zasílat. Po úspěšném připojení je první stavový automat ukončí činnost a spustí se druhý. Hlavní činností druhého stavového automatu je kontinuální měření až čtyř vstupních signálů, formátování výsledků do zprávy a vysílání této zprávy. Kromě této základní činnosti obsluhuje počáteční inicializační dialog, kdy druhému zařízení zasílá informace o rozlišení a vzorkovací periodě, a zajišťuje vyhodnocení a odpovědi na AT příkazy, které zasílá druhé zařízení.

V rámci druhé části softwarového návrhu byla v LabVIEW navržena a vytvořena aplikace pro příjem a zobrazení naměřených dat. Aplikace má jednoduché GUI, které umožňuje připojení a odpojení od měřicího zařízení, zastavení, pozastavení či znovu spuštění měření a vybrání souboru, do kterého se ukládají data. Aplikace navíc před zobrazením signály filtruje pásmovou propustí. Navržené jsou filtry pro EKG a PPG.

Testovalo se připojení k několika různým platformám. Z mobilních platform to byla zařízení s operačním systémem Android 4.x a Windows Phone 8, z PC platform pak zařízení s operačními systémy Windows 7, Windows Vista a Windows XP. Ke všem těmto platformám se modul připojí a zařízení s nimi dokáže komunikovat.

Během měření se testovala hlavně spolehlivost přenosu. Při maximální testované vzorkovací frekvenci 1 kHz pracovalo zařízení spolehlivě. Při testování aplikace bylo zjištěno, že při vzorkovacích frekvencích vyšších než cca 650 Hz nezvládá aplikace včas vyčíst zásobník s přijatými daty a dochází ke ztrátám dat. Dále se také testovaly filtry, u nichž se zjišťovalo, jak dokáží potlačit nepotřebné části signálů.



## Seznam obrázků

1	Stavový automat chápaný jako logický sekvenční obvod . . . . .	7
2	EKG s využitím Bluetooth . . . . .	10
3	Architektura monitorovacího systému EKG[28] . . . . .	11
4	Implementace umělého vidění[?] . . . . .	12
5	Modul KC-Wirefree . . . . .	14
6	Vývojové kity . . . . .	19
7	STMicroelectronics EvoPrimer[12] . . . . .	20
8	Připojení modulů k testovacímu kitu . . . . .	21
9	Invertující spínaný zdroj s obvodem MC34063 . . . . .	22
10	Běh programu v mikrokontroléru . . . . .	24
11	Stavový automat pro inicializaci . . . . .	25
12	Stavový automat pro měření . . . . .	26
13	Hlavní panel aplikace v LabView . . . . .	28
14	Dialog sloužící k připojení k sériové lince . . . . .	29
15	Různé typy filtrů EKG signálu . . . . .	33
16	Různé typy filtrů PPG signálu . . . . .	34
17	Detekce špiček a výpočet tepu . . . . .	35
18	Ukázka komunikace se zařízením na platformě Android . . . . .	36

## Seznam použitých zkratk

A/D – Analog/Digital (např. převodník)	LMP – Link Management Protocol
ALU – Aritmetic and Logical Unit	MCU - Memory Protection Unit
ARM – Advaced RISC Machine	NVIC -Nested Vector Interrupt Controller
BT – Bluetooth	PAL - Protocol Adaptation Layer
CMOS - Complementary Metal Oxide Semiconductor	PIN – Personal Identification Number
D/A – Digital/Analog (např. převodník)	RF – Radio Frequeny
EDR - Enhanced Data-Rate	RISC - Redused Instruction Set for Controllers
EKG - elektrokardiogram	RSCOMM - Radio Frequency Communications
FSM – Finite State Machine (stavový automat)	SDP - Service Discovery Protocol
HCI – Host Controller Interface	USB - Universal Serial Bus
HS - High Speed	UUID – Universal unique identifier
FW – firmware	USART – Universal Synchronous and Asynchronous Reveicer and Transmitter
ISM – Industrial, scientific and medical	VI - Virtual Instrument
Kbps – kilobit per second	$\mu$ P – mikroprocesor
L2CAP - Logical Link Control and Adaptation Protocol	

## Reference

- [1] Yiu, J.: The Definitive Guide to the ARM Cortex-M3, Oxford, 2007
- [2] Havlík, J., Lhotská, L., Parák, J., et al: A Modular System for Rapid Development of Telemedical Devices, *Journal of Universal Computer Science* (2013) 19(9): pp. 1242-1256
- [3] Agarwal, N.: New features and advantages of Bluetooth 3.0+HS [online]. *The Windows Club*, 2011 [vid. 14\_04\_05]. Dostupné z: <http://www.thewindowsclub.com/new-features-and-advantages-of-bluetooth-3-0hsdormant>
- [4] Butler, K.: What is Bluetooth 4.0? [online]. *LAPTOP magazine*, 2012 [vid. 14\_04\_05]. Dostupné z: <http://blog.laptopmag.com/just-what-is-bluetooth-4-0-anyway>
- [5] Weisstein, E. W.: Oversampling [online]. *Wolfram Mathworld*. 2012 [vid. 14\_04\_24]. Dostupné z: <http://mathworld.wolfram.com/Oversampling.html>
- [6] Vojáček, A.: Co se skrývá pod komunikací označenou jako USB OTG? [online]. *HW.cz - Vše o ekeltronice a programování*. (2012) [vid. 14\_05\_01]. Dostupné z: <http://www.hw.cz/teorie-a-praxe/dokumentace/co-se-skryva-pod-komunikaci-oznacenu-jako-usb-otg.html>
- [7] Orry, L.: C Compilers for ARM: Benchmark [online]. *RAISONANCE, MCU applications development solutions*. (2006) [vid. 14\_04\_29]. Dostupné z: [http://www.raisonance.com/arm\\_benchmark.html](http://www.raisonance.com/arm_benchmark.html)
- [8] *Using Bluetooth Products Bluetooth technology website* [online]. University of Maryland (2013) [vid. 13\_12\_05]. Dostupné z: <http://www.bluetooth.com/Pages/using-bluetooth-products.aspx>
- [9] *Big and Little Endian* [online]. Bluetooth SIG. (2013) [vid. 14\_04\_10]. Dostupné z: <http://www.cs.umd.edu/class/sum2003/cmsc311/Notes/Data/endian.html>
- [10] STM32VLDISCOVERY Discovery kit for STM32F100 Value Line - with STM32F100RB MCU. *STMicroelectronics* [online]. ©2014 STMicroelectronics. [vid. 14\_05\_12]. Dostupné z: [http://www.st.com/st-web-ui/static/active/en/fragment/product\\_related/rpn\\_information/board\\_photo/board\\_stm32vl\\_discovery.jpg](http://www.st.com/st-web-ui/static/active/en/fragment/product_related/rpn_information/board_photo/board_stm32vl_discovery.jpg)
- [11] STM32L-DISCOVERY Discovery kit for STM32L151/152 line - with STM32L152RB MCU. *STMicroelectronics* [online]. ©2014 STMicroelectronics. [vid. 14\_05\_12]. Dostupné z: <http://www.st.com/st-web>

ui/static/active/en/fragment/product\_related/rpn\_information/board\_photo/stm32l-discovery\_board.jpg

- [12] STMicroelectronics: STM32 Primer Starter Kit for easy evaluation and development with STM32 MCUs. *Design engineer community* [online]. Design engineer community, 2011. [vid. 14\_05\_12]. Dostupné z: <http://www.element14.com/community/docs/DOC-40452/1/stmicroelectronics-stm32-primer-starter-kit-for-easy-evaluation-and-development-with-stm32-mcus>
- [13] 802.15.1. Wireless medium access control (MAC) and physical layer (PHY) specifications for wireless personal area networks (WPANs). 2. vyd. New York, USA: IEEE, 14. června 2005
- [14] *STMicroelectronics EVOPrimer platform* [online]. Mouser Electronics, Inc. (2014) [vid. 14\_03\_15]. Dostupné z: <http://cz.mouser.com/new/stmicroelectronics/stmicroevoprimer/>
- [15] STMicroelectronics. *STM32 value line Discovery*. září 2010, rev. 1
- [16] STMicroelectronics. *STM32L1 discovery kits: STM32L-DISCOVERY and 32L152CDISCOVERY*. duben 2013, rev. 3
- [17] KC Wirefree. *User Guide kcSerial v2.4*. Tempe, Arizona, USA, únor 2010.
- [18] KC Wirefree. *kcSerial 3.0 Firmware User Guide*. Tempe, Arizona, USA, květen 2011.
- [19] Texas Instruments. *Application of the MC34063 Switching Regulator*. Dallas, Texas, září 2005.
- [20] ARM Limited. *Cortex-M3 Technical Reference Manual*. červenec 2010.
- [21] ARM Limited. *ARM Architecture Reference Manual*. říjen 2005.
- [22] ARM Limited. *ARM v7-M Architecture Reference Manual*. únor 2010.
- [23] STMicroelectronics. *RM0038 Reference Manual*. duben 2013.
- [24] STMicroelectronics. *RM0041 Reference Manual*. červenec 2011.
- [25] Losenický, M.. *Sekvenční logické obvody asynchronní*. 2010.
- [26] Proulx, J., Clifford, R., Sorensen, S., Lee, D. J., Archibald, J.: Development and Evaluation of a Bluetooth EKG Monitoring Sensor. *Proceedings of the 19th IEEE Symposium, Washington, DC, USA, 2006* [online]. Washington, 2006. [vid. 14\_02\_15]. Dostupné z: <http://80.ieeexplore.ieee.org/dialog.cvut.cz/stamp/stamp.jsp?tp=&arnumber=1647621>



- [27] Niazmand, K., Jehle, C., D'Angelo, L. T., Lueth, T. C.: A New Washable Low-Cost Garment for Everyday Fall Detection. *32nd Annual International Conference of the IEEE EMBS, Buenos Aires, Argentina, 31. srpna - 4. září 2010* [online]. Buenos Aires, 2011. [vid. 14\_04\_04]. Dostupné z: <http://80.ieeexplore.ieee.org/dialog.cvut.cz/stamp/stamp.jsp?tp=&arnumber=5627298>
- [28] Liang, J., Wu, Y.: Wireless ECG Monitoring System Based on OMAP. *2009 International Conference on Computational Science and Engineering, 29 - 31 srpna 2009* [online]. 2009. [vid. 14\_02\_15]. Dostupné z: <http://80.ieeexplore.ieee.org/dialog.cvut.cz/stampYinqin/stamp.jsp?tp=&arnumber=5283496>
- [29] Matteucci, P. B., Byrnes-Preston, P., Spencer, C. C. Y., Lovell, N., Suaning, G.: ARM-Based Visual Processing System for Prosthetic Vision. *33rd Annual International Conference of the IEEE EMBS Boston, Massachusetts USA, 30. srpna - 3. září, 2011* [online]. Boston, 2011. [vid. 14\_02\_15]. Dostupné z: <http://80.ieeexplore.ieee.org/dialog.cvut.cz/stamp/stamp.jsp?tp=&arnumber=6090974>
- [30] Lee, Y., Kwon, O., Shin, H. S., Jo, J. Lee, Y.: Noise reduction of PPG signals using a particle filter for robust emotion recognition. *2011 IEEE International Conference on Consumer Electronics - Berlin (ICCE-Berlin), 5. - 9. září, 2011* [online]. Berlín, 2011. [vid. 14\_02\_15]. Dostupné z: <http://80.ieeexplore.ieee.org/dialog.cvut.cz/stamp/stamp.jsp?tp=&arnumber=6031807>
- [31] STMicroelectronics. *STM32 value line Discovery*. září 2010, rev. 1
- [32] STMicroelectronics. *STM32L1 discovery kits: STM32L-DISCOVERY and 32L152CDISCOVERY*. duben 2013, rev. 3
- [33] Elgendi, M., Jonkman, M., DeBoer, F.: Frequency Bands Effects on QRS Detection [online]. *School of Enigneering and Information Technology, Charles Darwin University, 2008* [vid. 14\_05\_05]. Dostupné z: [http://www.researchgate.net/profile/Mohamed\\_Elgendi/publication/221334234\\_Frequency\\_Bands\\_Effects\\_on\\_QRS\\_Detection/file/e0b4951b69be5320f6.pdf](http://www.researchgate.net/profile/Mohamed_Elgendi/publication/221334234_Frequency_Bands_Effects_on_QRS_Detection/file/e0b4951b69be5320f6.pdf)
- [34] Parák, J., Dvořák, J., Pokorný, M., Havlík, J.: Modular Development Telemonitoring System [online]. *School of Enigneering and Information Technology, Charles Darwin University, 2008* [vid. 14\_05\_05]. Dostupné z: <http://80.ieeexplore.ieee.org/dialog.cvut.cz/stamp/stamp.jsp?tp=&arnumber=6256343>
- [35] Kubias, J., Sojka, M.: *eb\_jaws11*, Praha, 2012 [přístup 12\_09\_05]



## **Dodatek A: Obsah přiloženého DVD**

Popis obsahuje jednotlivých složek:

**BDT** - vytvořené knihovny pro mikrokontroléry STM32

**BP** - PDF s bakalářskou prací

**DS** - katalogové listy jednotlivých součástek a komponent

**LabVIEW** - aplikace pro PC napsaná v LabVIEW

**Mat** - materiály použité k vypracování teoretického rozboru a k čerpání doplňujících informací

**Sig** - naměřené signály