

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ
ÚSTAV TELEKOMUNIKACÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION
DEPARTMENT OF TELECOMMUNICATIONS

MĚŘIČ VARIABILITY TEPOVÉ FREKVENCE

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE
AUTHOR

RADEK STARZYCZNY

BRNO 2013



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY



**FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH
TECHNOLOGIÍ**

ÚSTAV TELEKOMUNIKACÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION
DEPARTMENT OF TELECOMMUNICATIONS

MĚŘIČ VARIABILITY TEPOVÉ FREKVENCE

METER OF PULSE FREQUENCY VARIABILITY

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

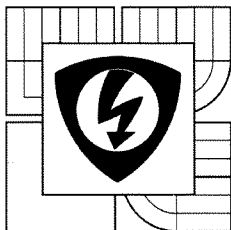
RADEK STARZYCZNY

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. IVO STRAŠIL

BRNO 2013



VYSOKÉ UČENÍ
TECHNICKÉ V BRNĚ

Fakulta elektrotechniky
a komunikačních technologií

Ústav telekomunikací

Bakalářská práce

bakalářský studijní obor
Teleinformatika

Student: Radek Starzyczny

Ročník: 3

ID: 125642

Akademický rok: 2012/13

NÁZEV TÉMATU:

Měřič variability tepové frekvence

POKYNY PRO VYPRACOVÁNÍ:

Cílem práce je vyvinout zařízení, které bude přijímat signál z hrudního pásu sporttesteru a zobrazovat na displeji tepovou frekvenci a některé vypočtené parametry její variability (HRV - alespoň parametry SD1, SD2, SD1/SD2). Zařízení bude realizováno na vývojovém kitu pro mikrokontroléry Cortex.

DOPORUČENÁ LITERATURA:

- [1] ACHARYA, U. R., et al. Heart rate variability: a review. Medical and Biological Engineering and Computing [online]. 2006, 44, [cit. 2011-10-02]. Dostupný z WWW: <<http://www.phys.mcw.edu/documents/HeartRateVariabilityAnIrregularitywecanlivewitReview.pdf>>.
- [2] KITLAS, A, et al. Nonlinear dynamics methods in analysis of heart rate variability. Annales Academiae Medicae Bialostocensis [online]. 2005, 50, [cit. 2011-10-02]. Dostupný z WWW: <http://www.advms.pl/roczniki_2005_supl_2/volumes/16_Kitlas.pdf>.

Termín zadání: 11.2.2013

Termín odevzdání: 5.6.2013

Vedoucí práce: Ing. Ivo Stražil

Konzultanti bakalářské práce:

prof. Ing. Kamil Vrba, CSc.

předseda oborové rady



UPOZORNĚNÍ:

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

ABSTRAKT

Cílem práce bylo vytvořit modul umožňující příjem a následné zpracování HRV (Heart Rate Variability) signál z hrudních pásů sporttesteru. Modul je koncepčně řešen jako rozšiřující karta pro vývojový kit STM32-Primer2. Pro příjem signálů ze sportovních pásů je použito čidlo firmy Polar RMCM-01. Přístroj umožňuje výpočet a zobrazení srdeční frekvence a některé další parametry její variability.

KLÍČOVÁ SLOVA

Variabilita Srdečního tepu, Srdeční tep, RMCM-01, Polar, Primer2, CircleOS

ABSTRACT

The aim of this thesis was to design and construct an autonomous module for receiving and processing HRV (Heart Rate Variability). The device is made as an extension board for development kit STM32. For receiving HR signal is used sensor RMCM-01. The device allow calculate and display the heart rate and its variability and some more parameters from collected data.

KEYWORDS

Heart Rate Variability, Heart Rate, RMCM-01, Polar, Primer2, CircleOS

PROHLÁŠENÍ

Prohlašuji, že svou bakalářskou práci na téma „Měřič variability tepové frekvence“ jsem vypracoval samostatně pod vedením vedoucího bakalářské práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené bakalářské práce dále prohlašuji, že v souvislosti s vytvořením této bakalářské práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a/nebo majetkových a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon), ve znění pozdějších předpisů, včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

Brno

.....

(podpis autora)

PODĚKOVÁNÍ

Rád bych poděkoval vedoucímu bakalářské práce panu Ing. Ivo Strašilovi za odborné vedení, konzultace, trpělivost a podnětné návrhy k práci.

Brno

.....

(podpis autora)



Faculty of Electrical Engineering
and Communication
Brno University of Technology
Purkynova 118, CZ-61200 Brno
Czech Republic
<http://www.six.feec.vutbr.cz>

PODĚKOVÁNÍ

Výzkum popsany v této bakalářské práci byl realizován v laboratořích podpořených z projektu SIX; registrační číslo CZ.1.05/2.1.00/03.0072, operační program Výzkum a vývoj pro inovace.

Brno

.....

(podpis autora)



EVROPSKÁ UNIE
EVROPSKÝ FOND PRO REGIONÁLNÍ ROZVOJ
INVESTICE DO VAŠÍ BUDOUCNOSTI



OBSAH

1	Úvod	11
2	Teoretický rozbor problematiky	12
2.1	Tvar EKG	12
2.1.1	RR interval	12
2.1.2	Tepové frekvence	13
2.1.3	Variabilita srdeční frekvence	13
2.2	Grafické znázornění HRV	14
2.3	Čidlo RMCM-01	16
2.4	Vývojový kit STM32-Primer2	17
2.4.1	Baterie a napájení	18
2.4.2	Operační systém CircleOS	18
2.4.3	Rozšiřující rozhraní	19
2.5	Vývojové prostředí Raisonance Ride7	21
2.6	Software pro zpracování naměřených dat	22
3	Realizace	23
3.1	Rozšiřující modul	23
3.1.1	Testovací schéma modulu	23
3.1.2	Finální schéma modulu	24
3.1.3	Deska plošných spojů	25
3.2	Použité periferie a jejich programování	26
3.2.1	Seznam a popis vlastních funkcí užitých v programu	26
3.2.2	Použité funkce z CircleOS	27
3.2.3	LED	28
3.2.4	SD/MMC karta	29
3.2.5	Čítač/Časovač	29
3.2.6	NVIC	31
3.2.7	Externí přerušení	31
3.2.8	LCD Displej	33
3.2.9	Vykreslování obrázků programem	33
3.3	Program	35
3.3.1	Chování programu	35
3.3.2	Zobrazované hodnoty	36
3.3.3	Výpočty programu	37
3.3.4	Ovládací a indikační prvky programu	38
3.3.5	Srovnání měřených dat	39

4 Závěr	40
Literatura	41
Seznam symbolů, veličin a zkratk	43
Seznam příloh	45
A Kompatibilita zařízení firmy Polar	46
B Schéma vnitřní struktury STM32F103x	47
C Zdroje hodinového kmitočtu	48
D Výstup programu KubiosHRV	49
E Obsah CD	50

SEZNAM OBRÁZKŮ

2.1	Základní tvar EKG křivky	12
2.2	Znázornění RR intervalu	13
2.3	Časový záznam délky RR intervalů	14
2.4	Příklad Poincarého mapy	15
2.5	Vyobrazení bezdrátového čidla RMCM-01	16
2.6	POLAR T31	16
2.7	Popis vývojového kitu STM32-Prime2	17
2.8	Ukázka rozšiřující karty	19
2.9	Pozice vstupně-výstupní brány 1 a 2 na DPS vývojového kitu	20
2.10	Hlavní okno vývojového prostředí	21
2.11	Program KubiosHRV	22
3.1	Princip funkce navrhovaného zařízení	23
3.2	Testovací schéma	23
3.3	Schéma rozšiřujícího modulu	24
3.4	Deska plošných spojů, pohled TOP	25
3.5	Deska plošných spojů, pohled BOTTOM	25
3.6	Rozmístění jednotlivých prvků	25
3.7	Rozšiřující modul - realizace	26
3.8	Schéma připojení LED k mikroprocesoru	28
3.9	Schéma připojení LCD k STM32	33
3.10	Program image2primer	34
3.11	Princip programu	35
3.12	Spuštěný program - naměřená data	36
3.13	Indikační prvky	38
3.14	Ovládací prvky	38

SEZNAM TABULEK

2.1	Vlastnosti přijímače RMCM-01	16
2.2	Priorita a přehled funkcí při obsluze přerušení	19
2.3	Seznam bran pro rozšiřující konektor	20
2.4	Formát datového souboru	22
3.1	Seznam použitých součástek pro testovací obvod	24
3.2	Seznam použitých součástek	24
3.3	Popis TIM_ClockDivision	30
3.4	Popis TIM_CounterMode	30
3.5	Výčet linek	32
3.6	Popis módů časovače	32
3.7	Tabulka pro EXT_Trigger	32
3.8	Funkce ovládacích prvků	38
3.9	Funkce indikačních prvků	38
3.10	Srovnání vypočtených dat	39
A.1	Seznam kompatibilních zařízení firmy polar	46

1 ÚVOD

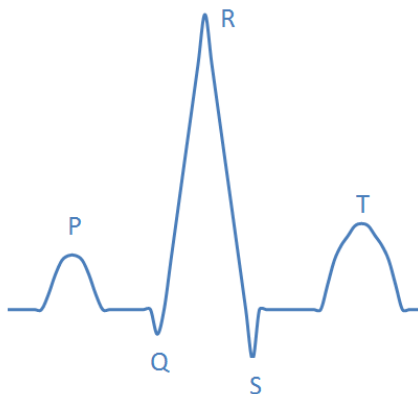
Cílem práce je vytvořit modul umožňující příjem a následné zpracování HRV (Heart Rate Variability) signálu z hrudních pásů sporttesteru. Modul má být koncepčně řešen jako rozšiřující karta pro vývojový kit STM32-Primer2. Pro příjem signálů z hrudních pásů sporttesteru bude použito čidlo firmy Polar RMCM-01. Ze zpracovaného signálu bude program vypočítávat a zobrazovat na displeji tepovou frekvenci a některé vypočtené parametry její variability. Naměřené hodnoty bude možno ukládat na integrovanou SD kartu, pro možnost pozdějšího zpracování.

První část práce je věnována teoretickému rozboru dané problematiky, konkrétně rozboru elektrické aktivity srdce a její grafické znázornění – elektrokardiogram.

Druhá část práce se věnuje realizaci rozšiřující karty, popisem vývojového kitu a realizovaného programu pro zpracování signálů z bezdrátového čidla.

2 TEORETICKÝ ROZBOR PROBLEMATIKY

2.1 Tvar EKG



Obr. 2.1: Základní tvar EKG křivky

Nejrozšířenějším a neinvazivním způsobem měření srdečního tepu je EKG (elektrokardiogram). Tato metoda detekuje nepatrné elektrické změny na kůži způsobené kontrakcemi svalů. Kontrakce myokardu jsou oproti jiným kontrakcím svalů charakteristického tvaru které lze rozdělit do několika fází [2, 3]:

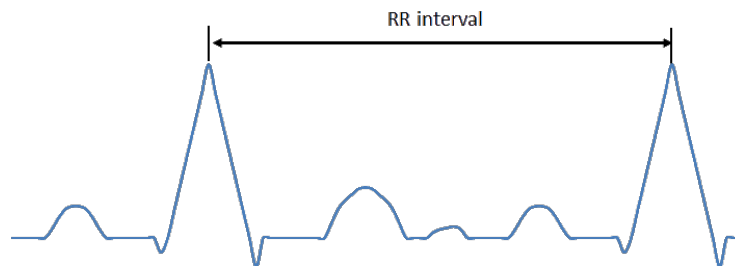
- První fáze má tvar vlny, často označovaná jako P vlna, je způsobena depolarizací (snižování membránového potenciálu) předsíní, předcházející kontrakci.
- Následuje trojice kmitů nazývaných komplex QRS, způsobena depolarizací komor.
- Poslední vlnu označovanou T způsobuje repolarizace (zvyšování membránového potenciálu) komor myokardu, naznačující návrat do klidového stavu.

Signál myokardu se šíří do celého těla bez větších útlumů a lze zaznamenat prakticky na kterémkoliv místě.

2.1.1 RR interval

Délka RR intervalu je řídicí hodnotou v popisu srdeční činnosti. Pro účely analýzy HRV (Heart Rate Variability, viz 2.1.3) jsou zajímavé intervaly RR před kterými jsou vlny P, abnormální intervaly jsou vynechány. Hodnotu intervalu získáme odečtením času poslední R vlny od předcházející.

Za předpokladu znalosti RR intervalu je možné určit tepovou frekvenci a také jiné charakteristické parametry variability srdečního tepu.



Obr. 2.2: Znázornění RR intervalu

2.1.2 Tepové frekvence

Tepová frekvence HR (Heart Rate) udává počet tepů (stahů myokardu) za jednu minutu. Tepová frekvence se vypočítá ze vzorce:

$$HR = \frac{60}{RR \text{ interval}} [\text{tep/minuta}] \quad (2.1)$$

HR se liší podle intenzity tělesné zátěže a kondice jedince. Klidová HR se pohybuje v rozmezí 65-90 tepů za minutu, u trénovaných jedinců klesá až k 50 tepům za minutu [10].

Maximální tepová frekvence (MHR - Maximum Heart Rate) je hodnota, která odpovídá maximální zátěži jakou je schopný organismus snášet a krátkodobě i udržet. Dle [1] MHR závisí na věku jedince. Rovnice pro výpočet MHR:

$$MHR = 217 - (0,85 * \text{věk}) [\text{tep/minuta}] \quad (2.2)$$

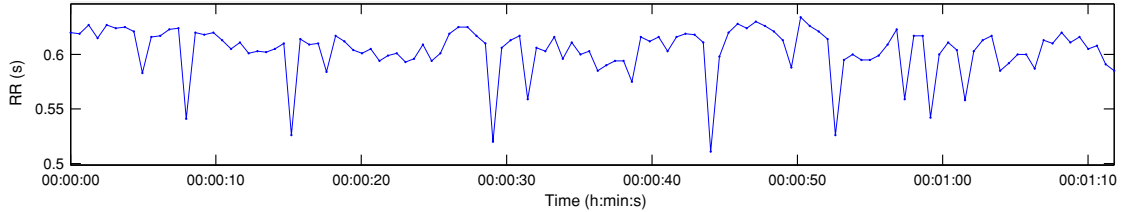
2.1.3 Variabilita srdeční frekvence

Pod pojmem variabilita srdeční frekvence HRV (Heart Rate Variability) rozumíme odchylky v intervalech mezi jednotlivými kontrakcemi myokardu (RR intervaly) a také okamžitou tepovou frekvenci HR (Heart Rate). HRV reprezentuje stav autonomního nervového systému řídící srdeční činnost. Analýza HRV má široké uplatnění v moderní medicíně - lze díky ní získat informace o kondici pacienta po infarktu myokardu, spánkových fázích, dokonce se v poslední době pomocí HRV zkouší léčit deprese[13]. HRV ale také ovlivňují vnější faktory - příkladem je počasí, hluk, psychická či fyzická zátěž. Proto je důležité při prezentaci získaných dat zvážit také vnější faktory.

Variabilita srdeční frekvence se zpravidla vynáší jako závislost tepové frekvence HR nebo časových intervalů mezi každými dvěma jednotlivými kontrakcemi myokardu (RR intervaly) v závislosti na čase.

Vzhledem k neustálé proměnlivosti RR intervalu, je vhodné počítat HR jako průměr několika posledních naměřených hodnot.

2.2 Grafické znázornění HRV



Obr. 2.3: Časový záznam délky RR intervalů

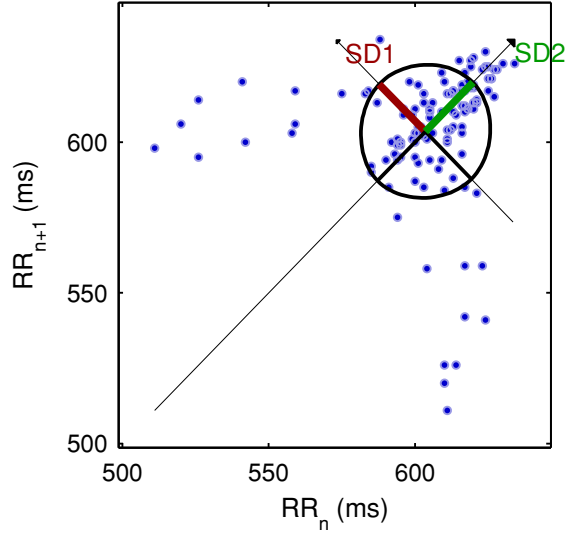
Na obrázku 2.3 je časový záznam délky trvání RR intervalů¹. Na ose x je vyneseno čas měření, na ose y délka RR intervalů v milisekundách. Vyobrazené impulsy jsou do grafu zaneseny při příchodu následující R vlny. Z tohoto zobrazení ovšem není zcela patrná diference délky RR intervalů, impulsy jsou vynášeny v závislosti na čase a tak je porovnání impulsu pouhým okem složité. Kvůli nepřehlednosti takto zobrazených hodnot se začaly používat jiné metody zobrazení - Poincarého mapy.

Poincarého mapy

Poincarého mapy zobrazují fluktuace hodnot $RR_{(i+1)}$ proti RR_i . Mapa vznikne nanesením délky RR intervalu s indexem i na osu x a na osu y interval s indexem $i+1$. Tvar takto vyobrazených dat připomíná elipsu, jejichž tvar popisuje vývoj délky RR intervalu v čase. V případě podlouhlé elipsy lze konstatovat, že byl zaznamenán velký rozdíl mezi minimální a maximální délkou trvání RR intervalů. Pokud se elipsa svým tvarem blíží spíše kružnici, rozdíl je minimální. Z Poincarého mapy lze vyčíst také další parametr, kterým je hustota bodů, respektive jejich vzdálenost. Tento parametr může vypovídat o určitých abnormalitách srdečního rytmu², nebo o chybách měření. Na obrázku 2.4 jsou znázorněny směrodatné odchylky SD1 a SD2 ve směrech kolmých os x_1 a x_2 , kde osa x_2 odpovídá identickým hodnotám $RR_{(n+1)} = RR_n$.

¹Data naměřená realizovaným přípravkem, zpracovaná programem KubiosHRV

²Čím blíže jsou jednotlivé body, tím stabilnější tepová frekvence a naopak



Obr. 2.4: Příklad Poincarého mapy

Hodnota SD1 popisuje krátkodobou variabilitu způsobenou zejména respirační sinusovou arytmií, hodnota SD2 udává variabilitu dlouhodobou. Poincarého mapy se využívají např. při studiu spánkových stádií.

Pro výpočet hodnot SD1 a SD2 je potřeba definovat vektory \vec{x} a \vec{y} :

$$\begin{aligned}\vec{x} &\equiv (x_1, x_2, \dots, x_n) \equiv (RR_1, RR_2, \dots, RR_n) \\ \vec{y} &\equiv (y_1, y_2, \dots, y_{n+1}) \equiv (RR_2, RR_3, \dots, RR_{n+1})\end{aligned}\quad (2.3)$$

Střed elipsy lze popsat jako

$$\begin{aligned}\sum_{i=1}^n x_c &= \sum_{i=1}^n x_i, & \sum_{i=1}^n y_c &= \sum_{i=1}^n y_i \\ x_c &= \bar{x}_i, & y_c &= \bar{y}_i\end{aligned}\quad (2.4)$$

Vzorec pro výpočet SD1²:

$$\begin{aligned}SD1^2 &= \text{Var} \left(\frac{1}{\sqrt{2}} RR_n - \frac{1}{\sqrt{2}} RR_{n+1} \right) \\ &= \text{Var} \left(\frac{1}{\sqrt{2}} x - \frac{1}{\sqrt{2}} y \right) \\ &= \frac{1}{n} \sum_{i=1}^n \frac{(x_i - \bar{x} - y_i + \bar{y})^2}{2} \\ &= \frac{1}{n} \sum_{i=1}^n \frac{((x_i - x_c) - (y_i - y_c))^2}{2}\end{aligned}\quad (2.5)$$

Vzorec pro výpočet SD2²:

$$SD2^2 = \frac{1}{n} \sum_{i=1}^n \frac{((x_i - x_c) + (y_i - y_c))^2}{2}\quad (2.6)$$

2.3 Čidlo RMCM-01



Obr. 2.5: Vyobrazení bezdrátového čidla RMCM-01

Bezdrátový přijímač srdečního tepu RMCM-01 zpracovává signál z hrudních pásů sporttesteru. Dokáže detekovat kompatibilní zařízení v okolí. Seznam kompatibilních zařízení firmy POLAR lze nalézt v příloze A.1. Příkladem kompatibilního hrudního pásu je POLAR T31 vyobrazený na obrázku 2.6.



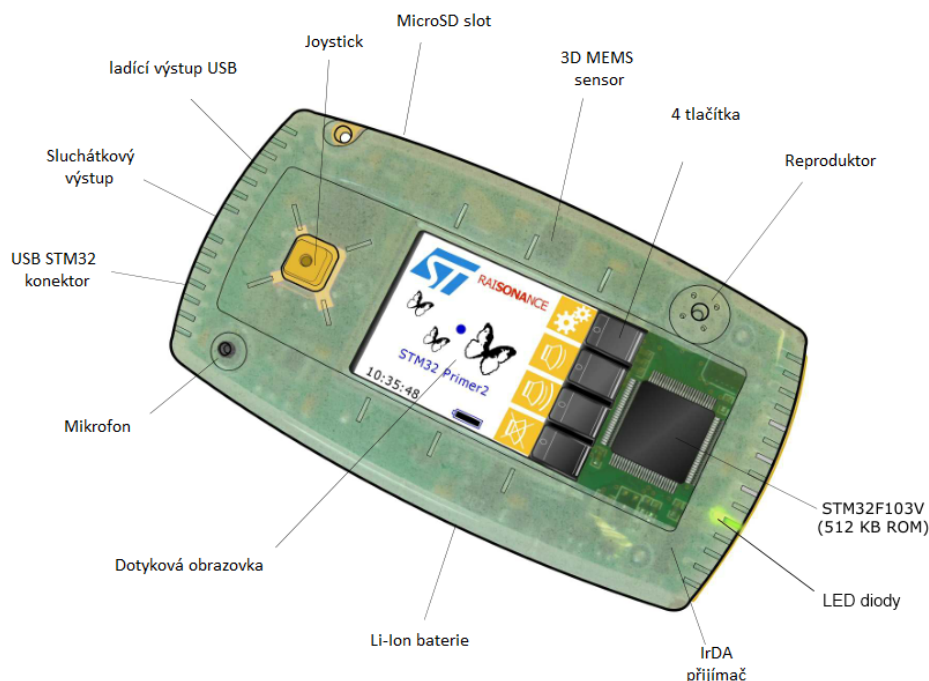
Obr. 2.6: POLAR T31

Po připevnění hrudního pásu na tělo začne pás vysílat signál odpovídající stahům myokardu. Výstupní brána FPLS čidla RMCM-01 generuje signál s napětím 3 V o délce 6 ms při detekci jakéhokoliv přijímaného signálu z kompatibilních zařízení. Pokud je v dosahu zařízení umožňující měření HRV, přijímač tento signál zachytí a transformuje do výstupu pojmenovaného HR. Na této výstupní bráně jsou očekávány po zpracování signálu 1 ms impulzy s napětím 3 V odpovídající RR intervalům (stahům myokardu). V případě, že žádné z kompatibilních vysílacích zařízení není v dosahu, žádný výstupní signál není generován [7]. Tabulka 2.1 popisuje jednotlivě vstupně-výstupní brány přijímače.

Parametr	Minimum	Maximum	Typicky	Jednotky
Napájecí napětí	2,5	3,4	3,0	V
Dosah	80	100	92	cm
Šířka pulzu HR			1	ms
Šířka pulzu FPLS			6	ms
Provozní teplota	0	60	20	°C

Tab. 2.1: Vlastnosti přijímače RMCM-01

2.4 Vývojový kit STM32-Primer2



Obr. 2.7: Popis vývojového kitu STM32-Prime2

Vývojový kit STM32 Primer2 bude sloužit jako zobrazovací a záznamová jednotka signálu z čidla RMCM01. Jedná se o vývojový prostředek pro platformu ARM Cortex M3. Hardware připomíná svými rozměry i konstrukcí mobilní telefon. Obsahuje 32bitový mikrokontrolér STM32F103 od firmy ST Microelectronics. V zařízení je integrován hardwarový programátor/debugger¹ RLINK, který usnadňuje práci s mikrokontrolérem.

Primer2 má předinstalován vlastní operační systém Circle0S.

Vlastnosti STM32 Primer2:

- Mikrokontrolér STM32F103VET6
 - jádro ARM Cortex M3, 512Kb Flash ROM, 64Kb SRAM
 - maximální vnitřní takt procesoru 72MHz
 - výpočetní výkon 90DMIPS, 1,25DMIPS/MHz (Dhrystone 2.1)
 - integrováný snímač teploty
- ADC (Analog-to-Digital Converter) (12bit),
- DAC (Digital-to-Analog Converter) (12bit), PWM (Pulse Width Modulation)

¹Debugger je součástí vývojového prostředí, sloužící k nalezení chyb v programech, především k označení místa v programu, kde došlo k programátorské chybě

- vstupně-výstupní piny GPIO, využitelné pro přerušení
- CAN, USART, SPI, I2C, IrDA, LIN
- dotykový grafický LCD displej (24bit, 128x160 pixelů)
- 4 směrový joystick se středovým tlačítkem, 4 tlačítka
- mikrofon a reproduktor pro záznam a reprodukci zvuku, sluchátkový výstup
- microSD slot
- 3-osý akcelerometr, který lze použít pro ovládání GUI operačního systému
- dva mini USB konektory, jeden pro programování a ladění, druhý pro uživatelské aplikace

2.4.1 Baterie a napájení

STM32-Primer2 obsahuje 400 mAh Li-Ion dobíjecí baterii. Standardně je napájení a případné dobíjení baterie zajištěno po propojení USB konektoru s PC. Pokud není USB propojeno, napájení zařízení zajišťuje interní baterie. Výrobce uvádí, že při plně nabitě baterii lze STM32-Primer2 využívat 6 hodin [6].

2.4.2 Operační systém CircleOS

Pro vývojový kit STM32-Primer 2 je dostupný operační systém vyvinutý firmou Raisonance. Jádro OS představuje funkce *main()*, která má za úkol prvotní nastavení periférií a systémových přerušení¹ vývojového kitu. O správu rutin přerušení se stará funkce *SysTickHandler*, která je volána každou milisekundu²[9]. Priorita těchto funkcí je v tabulce 2.2.

CircleOS poskytuje API (Application Programming Interface) usnadňující práci s perifériemi jako jsou LCD displej, LED, bzučák, tlačítka, MEMS (Micro-Electro-Mechanical Systems), Audio a další. Celkový přehled dostupných funkcí CircleOS API je ve formě webových stránek na adrese http://www.stm32circle.com/circleos_doc/globals.html

Pro práci s GPIO (General Purpose Input/Output) bránami na rozšiřujícím rozhraní, stejně jako obsluhu přerušení nebo správu čítačů/časovačů, neposkytuje CircleOS žádné rozhraní. Je potřeba připojit hlavičkové soubory pro rodinu mikrokontrolérů STM32F10x. Jedná se o STM32 Standard Peripheral Library, dostupné z [14].

V CircleOS je možné načíst více nezávislých aplikací. Každá z aplikací má plný přístup k CPU a v RAM může využívat blok o velikosti 16 KB (adresní rozsah od 0x2000000 do 0x20003FFF) [9].

¹Rutina přerušení je část programu, která se stará o obsluhu událostí vzniklých z přerušení

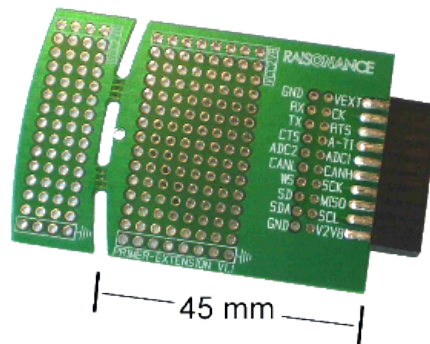
²Lze měnit pomocí funkce *SysTick_Configuration()*

Priorita	Název	Popis
1	MEMS_Handler	Obsluha cidel polohy
2	LED_Handler	Obsluha LED
3	BUTTON_Handler	Správa obslužných tlačítek
4	BUZZER_Handler	Správa bzucáku
5	MENU_Handler	Obsluha událostí z menu
7	LCD_Handler	Obsluha pro LCD display
9	RTC_DisplayTime	Hodiny presného času
10	AUDIO_Handler	Správa zvukového zařízení
11	TOUCHSCR_Handler	Správa dotykové vrstvy
12	TOOLBAR_Handler	Obsluha panelu nástroju
13	POWER_Handler	Obsluha napájení

Tab. 2.2: Priorita a přehled funkcí při obsluze přerušení

CircleOS lze získat po bezplatné registraci na webových stránkách firmy STM32: <http://www.stm32circle.com/projects/download.php?id=101&file=Circle-OS-4.32.zip>

2.4.3 Rozšiřující rozhraní

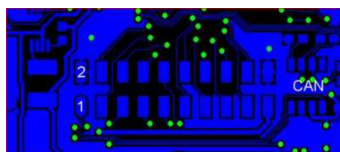


Obr. 2.8: Ukázka rozšiřující karty

Rozšiřující rozhraní umožňuje připojení vlastního zařízení k vývojovému kitu Primer2. Přestože je Primer2 vybaven různými senzory (MEMS, interní teplotní senzor,..), existují aplikace, jako je například tato práce, kdy senzory již integrované nestačí. Proto je možné propojit vlastní obvod pomocí standardizovaného HE14 pravoúhlého konektoru. Rozšiřující rozhraní je přístupné po odklopení zadního oranžového krytu. Na obr 2.8 je ukázka rozměrů karty. Přesnou specifikaci lze nalézt v *Primer2 Extension Mechanical* [5].

Číslo	Název	STM32	Možné funkce
1	VCC2V8	V _{cc}	Napájení 2,8V, cca 100mA
2	GND	GND	Zem
3	CX_I2CSCL	PB.06	GPIO, I2C1_SCL, TIM4_CH1, USART1_TX
4	CX_I2CSDA	PB.07	GPIO, I2C1_SDA, TIM4_CH2, USART1_RX
5	AUDIO_SPI_MISO	PB.14	GPIO, SPI2_MISO, USART3_RTS, TIM_CH2N
6	AUDIO_I2S2_SD	PB.15	GPIO, SPI2_MOSI, TIM1_CH3N
7	AUDIO_I2S2_SCK	PB.13	GPIO, SPI2_SCK, USART3_CTS, TIM1_CH1N, SPI2 je také propojeno na výstup z mikrofonu
8	AUDIO_I2S2_WS	PB.12	GPIO, SPI2_NSS, USART3_CK, TIM1_BKIN
9	CX_CANH		
10	CX_CANL		
11	CX_ADC1	PC.04	GPIO, ADC12_IN14
12	CX_ADC2	PC.05	GPIO, ADC12_IN15
13	CX_ADC_TIM	PB.00	GPIO, ADC12_IN8, TIM3_CH3, TIM1_CH2N
14	CX_USART_CTS	PA.00	PA0/WKUP/ADC, GPIO, WKUP, USART2_CTS, ADC12_IN0, TIM2_CH1_ETR
15	CX_USART_RTS	PA.01	GPIO, USART2_RTS, ADC12_IN1, TIM2_CH2
16	CX_USART_TX	PA.02	GPIO, USART2_TX, ADC12_IN2, TIM2_CH3
17	CX_USART_CK	PA.04	GPIO, SPI1_NSS, USART2_CK, ADC12_IN4
18	CX_USART_RX	PA.03	GPIO, USART2_RX, ADC12_IN3, TIM2_CH4
19	VCC_EXT		Jako externí zdroj
20	GND	GND	Zem

Tab. 2.3: Seznam bran pro rozšiřující konektor



Obr. 2.9: Pozice vstupně-výstupní brány 1 a 2 na DPS vývojového kitu

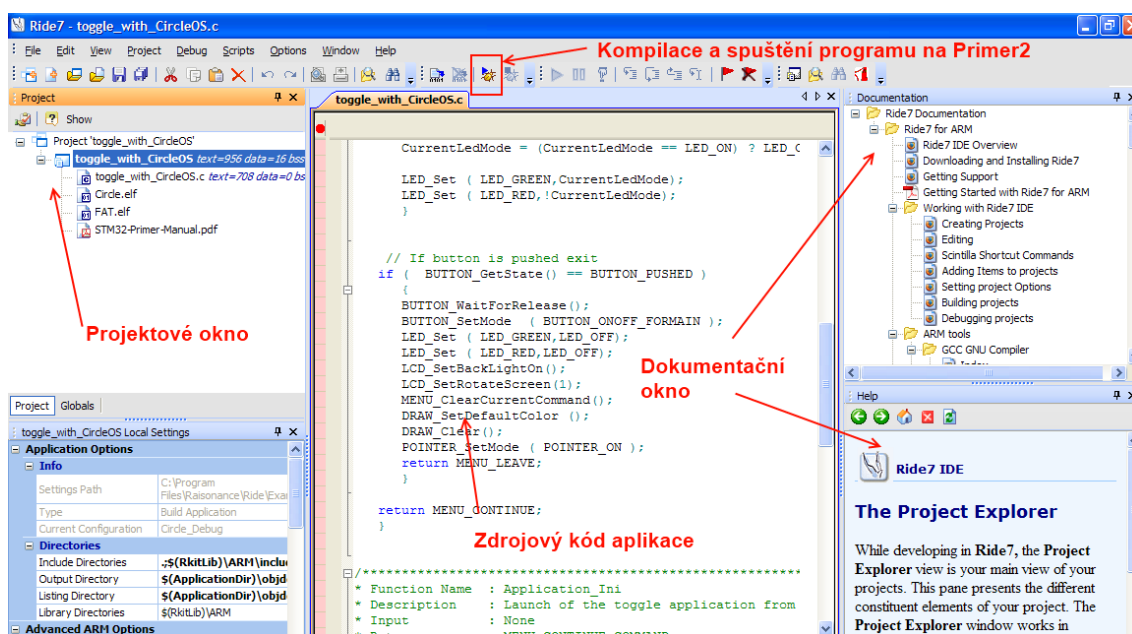
V tabulce 2.3 je seznam názvů a pozic jednotlivých vstupně-výstupních bran. Pro správnou funkci navrhovaného zařízení je nutné použít minimálně jednu, pro dodatečnou indikaci detekovaného zařízení pak další vstupně-výstupní bránu. Rozhraní disponuje napájecím napájením 2.8 V.

2.5 Vývojové prostředí Raisonance Ride7

Plnou podporu pro vývoj aplikace pro Primer2 má vývojové prostředí firmy Raisonance RIDE7. Vývojové prostředí je určeno pouze pro MS Windows. Na adrese <http://www.mcu-raisonance.com/microcontrollers-link-free-download.html> lze nalézt aktuální instalační soubory vývojového prostředí Ride7 a kompilátoru RKit-ARM. Software vyžaduje registraci na stránkách výrobce, bez ní je funkčnost programu omezena na 30 dní.

U Primer2 bývá problém v kompatibilitě verze Ride7, RKit-ARM a samotného Circle OS, proto je vhodné při prvním setkání s Primer2 vše aktualizovat nebo instalovat poslední vydané verze. Předejde se tím možným komplikacím s nekompatibilními verzemi jednotlivých komponent.

Aplikace bude napsána v programovacím jazyce ANSI C. Pro překlad je využit překladač gcc ARM s licencí GNU GPL.



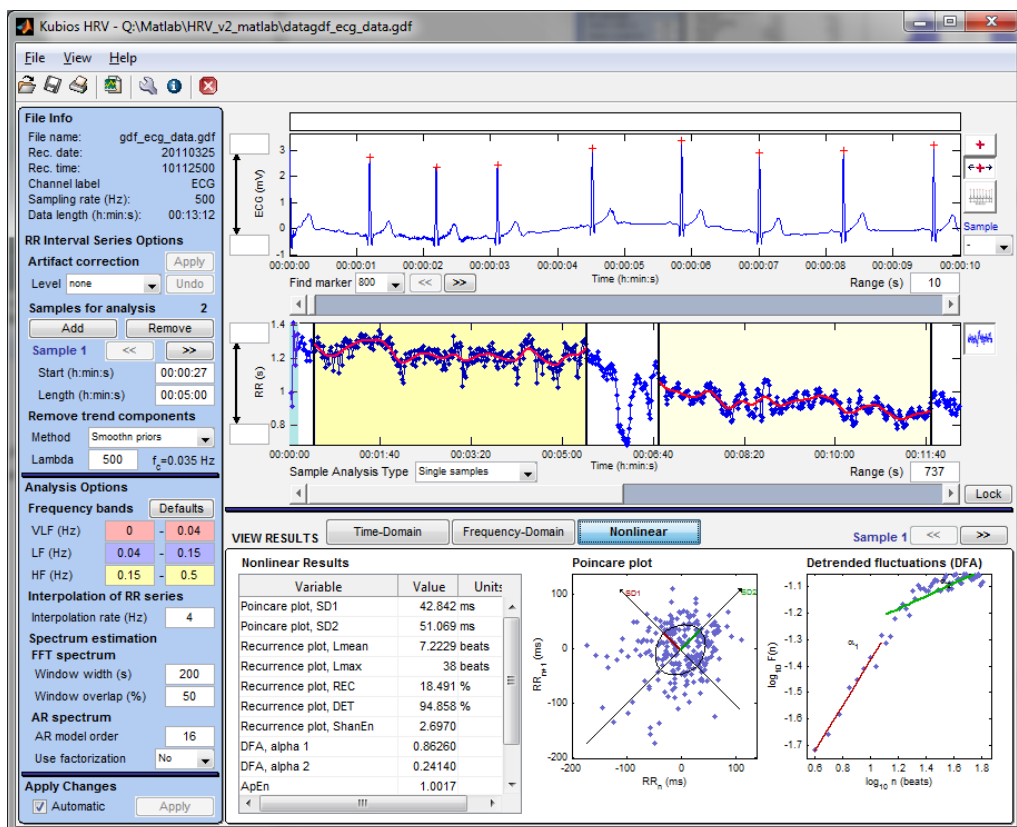
Obr. 2.10: Hlavní okno vývojového prostředí

2.6 Software pro zpracování naměřených dat

Pro zpracování naměřených dat na PC lze využít software KubiosHRV, který lze stáhnout na <http://kubios.uef.fi/>. Program lze provozovat na platformách Windows i Linux. Uživatelská příručka je k nalezení na http://kubios.uef.fi/media/Kubios_HRV_2.1_Users_Guide.pdf. Program očekává jako vstupní data textový soubor (ASCII) s hodnotami RR intervalů nebo záznam EKG. Práce počítá s měřením RR vln signálu, proto bude pro ukládání dat použit formát znázorněný v tabulce 2.4. Pro program to znamená, že se bude na SD kartu zapisovat časový rozdíl mezi jednotlivými impulzy z čidla RMCM01 v milisekundách.

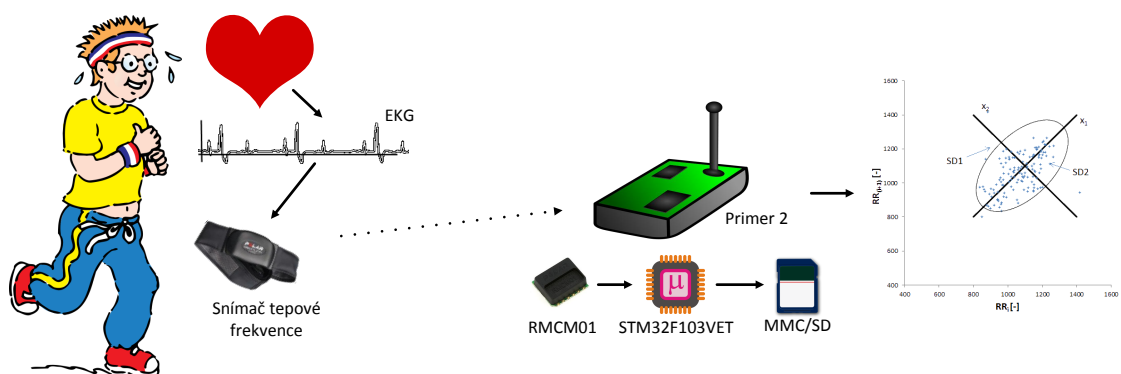
RR ₁	0.590
RR ₂	0.670
RR ₃	0.670
...	..
RR _n	0.888

Tab. 2.4: Formát datového souboru



Obr. 2.11: Program KubiosHRV

3 REALIZACE



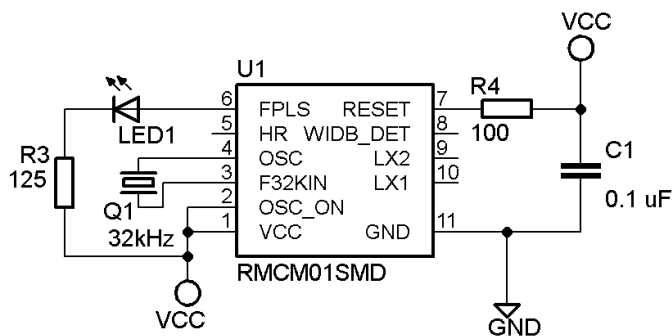
Obr. 3.1: Princip funkce navrhovaného zařízení

Princip funkcionality navrhovaného zařízení lze popsat obrázkem 3.1. Navrhované zařízení má být schopno reagovat na signál z hrudního pásu sporttesteru a dle naměřených dat zobrazovat hodnoty jednotlivých parametrů HRV.

3.1 Rozšiřující modul

3.1.1 Testovací schéma modulu

Před zapájením bezdrátového čidla na finální desku plošných spojů je vhodné otestovat jeho funkčnost připojením LED diody na výstup FPLS s rezistorem a připojit obvod k napájecímu napětí 3 V. Takové testovací zapojení je vyobrazeno na obrázku 3.2. LED dioda začne blikat v okamžiku detekce hrudního pásu sporttesteru¹.



Obr. 3.2: Testovací schéma

¹Tuto funkcionalitu později zajišťuje červená LED na vývojovém kitu

Výpočet předřadného odporu R_3 k diodě LED1:

$$R_3 = \frac{U_{cc} - U_{LED}}{I_F} * 1000 = \frac{3 - 0.5}{20} * 1000 = 125[\Omega] \quad (3.1)$$

kde

U_{cc} je napájecí napětí

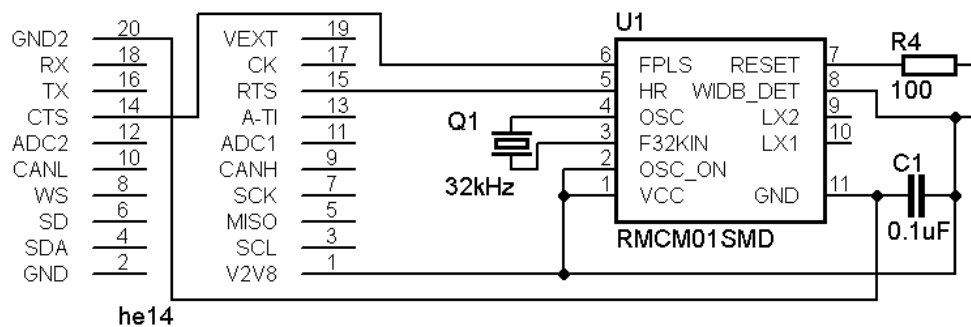
U_{LED} je úbytek napětí na LED

I_F je požadovaný proud LED

Název	Hodnota	Pouzdro	Popis
C1	0.1 μ F	5x11	Kondenzátor
Q1	0,032768 MHz	DT38	Krystal
R1	100 Ω	0207	Rezistor
R2	125 Ω	0207	Rezistor
U1	RMCM01SMD	RMCM01SMD	Čidlo RMCM-01

Tab. 3.1: Seznam použitých součástek pro testovací obvod

3.1.2 Finální schéma modulu



Obr. 3.3: Schéma rozšiřujícího modulu

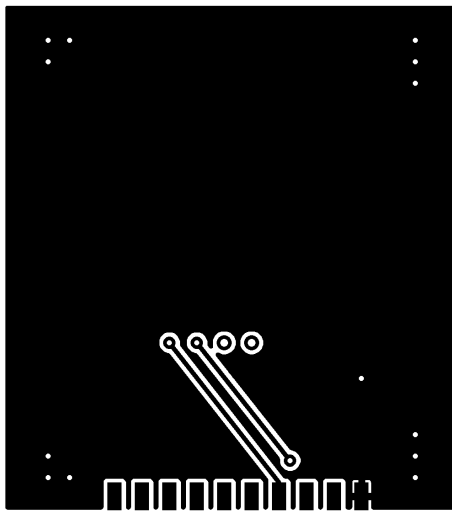
Navrhované zapojení vychází z doporučení firmy Polar [7] ke kterému byl přidán blokovací kondenzátor C1 pro odrušení případných negativních vlivů z napájecí větve (kompenzace chvilkových poklesů napájení).

Název	Hodnota	Pouzdro	Popis
C1	0.1 μ F	SMD 0805	Kondenzátor
Q1	0,032768 MHz	DT38	Krystal
R1	100 Ω	SMD 0805	Rezistor
SSW-110-X-D	SSW-110-X-D	SSW-110-X-D	Konektor
U1	RMCM01SMD	RMCM01SMD	Čidlo RMCM-01

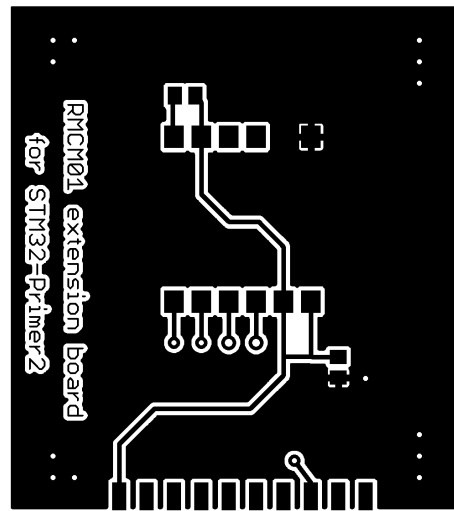
Tab. 3.2: Seznam použitých součástek

3.1.3 Deska plošných spojů

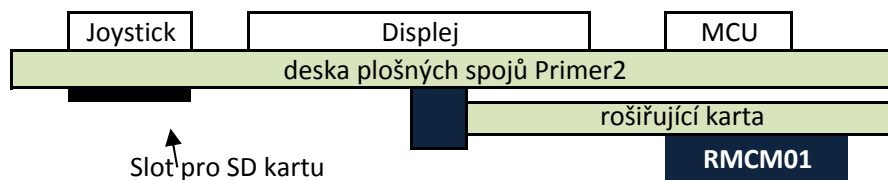
K návrhu DPS byl použit program firmy CadSoft Eagle PCB Design Software. Jedná se o dvouvrstvý návrh. Vzhledem ke konstrukci STM32 Primer2 (obrázek 3.6) je nutné všechny součástky umístit na stranu BOTTOM. Při návrhu byl kladen důraz na co nejmenší vzdálenost mezi součástkami. Stranu TOP téměř celou pokrývá zemnicí vrstva která je přenášena pomocí prokovaných otvorů také na volnou plochu strany BOTTOM kvůli odfiltrování případných elektromagnetických rušení. Finální verze rozšiřující karty je vyobrazena na obrázku 3.7.



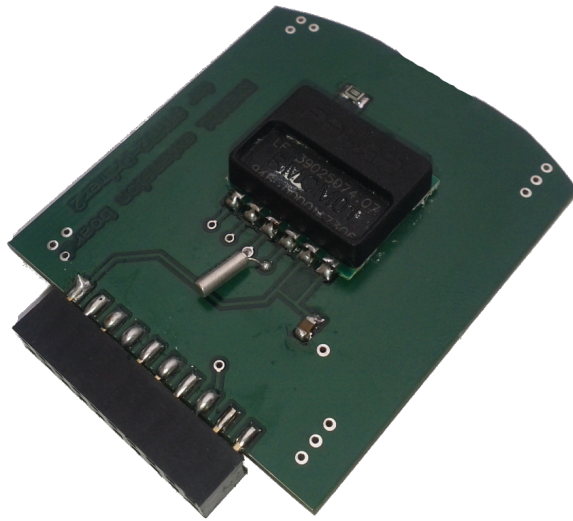
Obr. 3.4: Deska plošných spojů, pohled TOP



Obr. 3.5: Deska plošných spojů, pohled BOTTOM



Obr. 3.6: Rozmístění jednotlivých prvků



Obr. 3.7: Rozšiřující modul - realizace

3.2 Použité periferie a jejich programování

3.2.1 Seznam a popis vlastních funkcí užitých v programu

```
void GPIO_Inicializace(void);
```

Funkce pro nastavení vstupně-výstupních bran. Tato funkce nejprve povolí taktování na sběrnici `RCC_APB2Periph_GPIOA`, poté nastaví bránu `PA.00` (FPLS) a `PA.01` (HR) jako vstupní s rychlostí 50 MHz^2 a také `PB.08` jako alternativní funkci pro možnost ovládání podsvětlení LCD. Nakonec povolí taktování na sběrnici `RCC_APB2Periph_AFIO` a napojí linky přerušení z `GPIOA` pro `PA.00` a `PA.01`.

```
void TIM3_Inicializace(void);
```

Funkce pro nastavení čítače/časovače `TIM3`. Vzestupný mód čítání, perioda `65535`, dělička `36000-1`.

```
void NVIC_Inicializace(void);
```

Funkce pro nastavení `NVIC`. Pro všechny tři zdroje přerušení (časovač/čítač, `PA.00`, `PA.01`) přemapuje adresu, na které lze nalézt obslužnou rutinu pro zpracování těchto přerušení.

²Rychlost udává maximální možnou rychlost, se kterou budou data vzorkována z výstupního registru na výstupní port, nebo ze vstupního portu do vstupního registru

```
tHandler preruseni_fpls(void);
```

Funkce volaná při obsluze přerušení z kanálu EXTIO_IRQn. Jedná se o PA.00 - detekce vysílacího zařízení v okolí RMCM01.

```
tHandler preruseni_tim3(void);
```

Funkce volaná při přetečení čítače/časovače TIM3. Pokud dojde k přetečení TIM3, zastaví se zápis dat na SD kartu.

```
tHandler preruseni_hr(void);
```

Funkce volaná při obsluze přerušení z kanálu EXTII1_IRQn. Pokud přerušení přišlo od HR_LINE, uloží se obsah čítače/časovače TIM3 a jeho obsah se vynuluje. Funkce zajišťuje měření RR intervalů.

```
void EXTI_Inicializace(void);
```

Funkce pro nastavení typ přerušení.

```
void start_record();
```

Funkce nastaví proměnné tak, aby byl možný zápis na SD kartu za předpokladu, že je vložená do slotu.

```
void stop_record();
```

Funkce nastaví proměnné tak, aby zápis na SD kartu nebyl umožněn.

3.2.2 Použité funkce z CircleOS

Přehled použitých funkcí z CircleOS. Seznam obsahuje funkce volané z hlavního programu. Tyto funkce mohou vyžadovat přístup k jiným neuvedeným funkcím CircleOS.

```
void LCD_Init(void);
void LCD_FillRect(u16 x, u16 y, u16 width, u16 height, u16 color);
void LCD_DrawRect(u16 x, u16 y, u16 width, u16 height, u16 color);
void LCD_SetOffset (ENUM_Offset Offset);
void UTIL_int2str(u8* ptr, int_t X, len_t digit, bool fillwithzero);
void UTIL_SetPll (enum eSpeed speed);
void DRAW_DisplayString(u16 x, u16 y, const u8* ptr, len_t len);
void DRAW_SetImage (const color_t * ptr, coord_t x, coord_t y, coord_t width,
    coord_t height);
void DRAW_DisplayVbat (coord_t x, coord_t y);
void DRAW_DisplayTime (coord_t x, coord_t y);
```

```

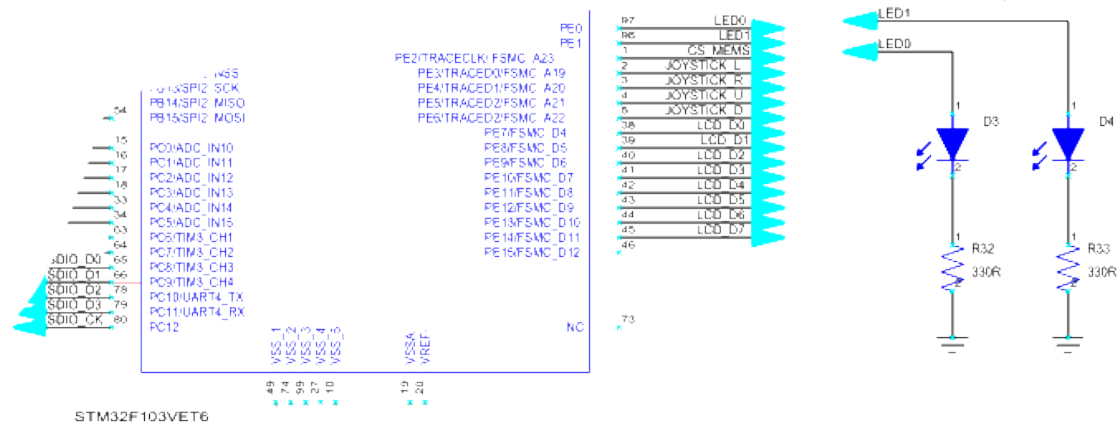
void FS_Seek (PFILEINFO fileinfo, u32 offset);
u32 FS_OpenFile (PVOLINFO volinfo, u8 *path, u8 mode, PFILEINFO fileinfo)
u32 FS_Unmount (enum STORAGE_device device);
u32 FS_Mount (enum STORAGE_device device);
u32 FS_GetVolumeInfo (u8 unit, u32 startsector, PVOLINFO volinfo);
u32 FS_WriteFile (PFILEINFO fileinfo, u8 *buffer, u32 *successcount, u32 len);

```

Dokumentace funkcí CircleOS je k dispozici na http://www.stm32circle.com/circleos_doc/.

3.2.3 LED

Program používá červenou LED diodu k indikaci signálu z čidla RMCM01. Bezdrátové čidlo generuje na výstupní bráně FPLS 6 ms impuls o velikosti 3 V při detekci vysílaného signálu z hrudního pásu sporttesteru. Takto detekovaný signál program interpretuje aktivováním/deaktivováním výstupu PE.01 (vstupně-výstupní brána 97 na mikroprocesoru STM32F103VET6), na který je připojena LED dioda s červenou barvou.



Obr. 3.8: Schéma připojení LED k mikroprocesoru

```

tHandler preruseni_fpls(void){
    if(EXTI_GetITStatus(FPLS_LINE) != RESET){
        StateLED_FPLS = (StateLED_FPLS == LED_ON) ? LED_OFF : LED_ON;
        LED_Set( LED_RED, StateLED_FPLS );
        EXTI_ClearITPendingBit (FPLS_LINE);
    }
}

```

Kód 3.1: Funkce volaná přerušením od PA.01

3.2.4 SD/MMC karta

Zápis na SD kartu je umožněn pouze v případě, že je SD/MMC karta vložena do slotu (popis kitu na obrázku 2.7) a správně detekována pomocí CircleOS funkce:

```
FS_Mount(MMCSD_SDIO)
```

Návratová hodnota je typu `u32`. Jedná se o počáteční sektor MBR (Master Boot Record) připojeného oddílu. Zjištění připojené SD/MMC karty se provádí pouze jednou a to při spuštění programu následujícím kódem:

```
StartMBR=FS_Mount(MMCSD_SDIO);
if (StartMBR != 0xFFFFFFFF){
    if (!FS_GetVolumeInfo(0, StartMBR, &volume_info)){
        sdcard_ok = 1;
    }
}
```

Zápis dat na SD kartu zajišťuje následující blok kódu, kde proměná `sdcard_ok` značí dostupnou SD kartu a `record` povolené zaznamenávání. K zapisovanému řetězci jsou ručně přidány netisknutelné znaky `0x0d` (návrát kurzoru) a `0x0a` (nový řádek) aby výsledný textový soubor splňoval konvence ukončení řádku z MS Windows.

```
if(sdcard_ok==1 && record==1){
    j=0;
    UTIL_int2str(display_buffer, RR_buf[puls-1], 7, 1);
    FS_OpenFile(&volume_info, Filename, FS_WRITE, &file_info);
    FS_Seek(&file_info, 1000000);
    display_buffer[8]=0x0d;
    display_buffer[9]=0x0a;
    FS_WriteFile(&file_info, (u8 *)display_buffer, &j, 10);
}
```

3.2.5 Čítač/Časovač

Jádro Cortex M3 obsahuje 8 časovačů, které jsou v literatuře označovány jako TIM+číslo časovače. Každý časovač se skládá z 16-bitového čítače s automatickou obnovou hodnoty³ a programovatelnou děličkou hodinové frekvence. Existují 3 druhy časovačů/čítačů:

- pokročilé: TIM1, TIM8
- obecné: TIM2, TIM3, TIM4, TIM5
- základní: TIM6, TIM7

Pokročilé a obecné čítače/časovače mohou být použity pro různé účely, například měření délky vstupního signálu (input capture) nebo tvorba výstupního signálu (Output Compare, PWM, atp). Délku pulzu a amplitudu lze pomocí děličky časovače nastavovat v rozmezí několika mikrosekund až po milisekundy.

³funkce auto-reload

Základní časovače mohou být použity pro generování časových událostí stejně jako ostatní časovače, ale jsou využity především k DA převodu - DAC (Digital-to-Analog Converter). Základní časovače jsou vnitřně napojeny k DAC a díky tomu jsou vhodné k jeho řízení. Struktura použitá k nastavování čítače/časovače mimo TIM6 a TIM7:

```
typedef struct {
    u16 TIM_Period;
    u16 TIM_Prescaler;
    u16 TIM_ClockDivision;
    u16 TIM_CounterMode;
    u8 TIM_RepetitionCounter;
} TIM_TimeBaseInitTypeDef;
```

Kód 3.2: Konfigurační struktura časovače/čítače

TIM_Period definuje periodu. Uvedená hodnota se obnovuje při každém přetečení. Platné hodnoty jsou v rozsahu 0x0000 až 0xFFFF.

TIM_Prescaler definuje hodnotu, kterou se má dělit hodinový kmitočet časovače/čítače. Platné hodnoty jsou v rozsahu 0x0000 až 0xFFFF.

TIM_ClockDivision definuje hodinový kmitočet pro čítač/časovač. Platné hodnoty jsou uvedené v následující tabulce 3.3

TIM_CounterMode definuje typ čítání. Platné hodnoty jsou uvedené v tabulce 3.4:

TIM_RepetitionCounter definuje čítač opakování. Pro PWM mód (N+1) odpovídá:

Hodnota	Popis
TIM_CKD_DIV1	$T_{DTS} = T_{ck_tim}$
TIM_CKD_DIV2	$T_{DTS} = 2 \times T_{ck_tim}$
TIM_CKD_DIV4	$T_{DTS} = 4 \times T_{ck_tim}$

Tab. 3.3: Popis TIM_ClockDivision

Hodnota	Popis
TIM_Counter_Up	Čítač inkrementální.
TIM_Counter_Down	Čítač dekrementální.
TIM_Counter_CenterAligned1	tohle není ani v ref.
TIM_Counter_CenterAligned2	je tam nějaká kravina
TIM_Counter_CenterAligned3	bez vysvětlení.

Tab. 3.4: Popis TIM_CounterMode

3.2.6 NVIC

NVIC (Nested Vectored Interrupt Controller) je řadič s vektory přerušení a podporou vnořených přerušení. Umožňuje nastavení až 256ti úrovní priorit pro 240 externích přerušení. Prioritu přerušení lze měnit dynamicky. U procesoru Cortex-M3 je doba reakce na přerušení fixní. Procesor má zabudovanou podporu systémových výjimek, které jsou identifikovány čísly 1 až 15 a dále podporu externích přerušení, které implementuje výrobce mikrokontroléru.

Obsluha rutin může být přerušena pouze výjimkou, která má vyšší prioritu. Pokud mají dvě výjimky nastavenou stejnou úroveň priority, přednost při vykonávání má funkce přerušení s nižším číslem výjimky. Když dojde k požadavku na obsluhu výjimky a ta nemůže být právě obsloužena, protože je například prováděna obsluha přerušení s vyšší prioritou, zůstane výjimka v nevyřízeném stavu, dokud nemůže být obsluha výjimky zahájena.

Struktura použitá k nastavování NVIC

```
typedef struct{
    uint8_t NVIC_IRQChannel;
    uint8_t NVIC_IRQChannelPreemptionPriority;
    uint8_t NVIC_IRQChannelSubPriority;
    FunctionalState NVIC_IRQChannelCmd;
} NVIC_InitTypeDef;
```

Kód 3.3: Konfigurační struktura NVIC

`NVIC_IRQChannelPreemptionPriority` konfigurace priority IRQ kanálu.

`NVIC_IRQChannelSubPriority` konfigurace subpriority IRQ kanálu.

`NVIC_IRQChannelCmd` povolení IRQ kanálu. Možné hodnoty jsou `ENABLE` nebo `DISABLE`.

3.2.7 Externí přerušení

Externí přerušení je jedno z možných přerušení řízených NVIC. Kterákoliv GPIO vstupně-výstupní brána může být nakonfigurována tak, aby vyvolala externí přerušení. Externí přerušení jsou řazena pro GPIO stejného pořadí. Událost na `PA.00` tedy vyvolá přerušení `EXTI0`. Pro vstupně-výstupní brány s číslem vyšším než 4 je použito skupinové přerušení (`EXTI9_5_IRQHandler` a `EXTI15_10_IRQHandler`), proto je v jejich obsluze nutné kontrolovat, od které linky přerušení přišlo. Tuto kontrolu zajistí následující porovnání:

```
if (EXTI_GetITStatus(LINE) != RESET)
```

kde `LINE` je označení linky přerušení, které nás zajímá.

Struktura pro nastavení:

```
typedef struct {
    u32 EXTI_Line;
    EXTIMode_TypeDef EXTI_Mode;
    EXTIrigger_TypeDef EXTI_Trigger;
    FunctionalState EXTI_LineCmd;
} EXTI_InitTypeDef;
```

EXTI_Line nastaví linku pro externí přerušení. Platné hodnoty v tabulce 3.5.

Linka	Popis
EXTI_Line0	Linka pro externí přerušení 0
EXTI_Line1	Linka pro externí přerušení 1
...	...
EXTI_Line18	Linka pro externí přerušení 18

Tab. 3.5: Výčet linek

EXTI_Mode nastavení typu reakce pro povolené linky. Možné hodnoty jsou v tabulce 3.6.

EXTI_Mode	Popis
EXTI_Mode_Event	Reakce na událost
EXTI_Mode_Interrupt	Reakce na přerušení

Tab. 3.6: Popis módů časovače

EXTI_Trigger nastavuje spouštěč . Platné hodnoty v tabulce 3.7.

EXTI_Trigger	Popis
EXTI_Trigger_Falling	Reakce na spádovou hranu
EXTI_Trigger_Rising	Reakce na náběžnou hranu
EXTI_Trigger_Rising_Falling	Reakce na náběžnou i spádovou hranu

Tab. 3.7: Tabulka pro EXT_Trigger

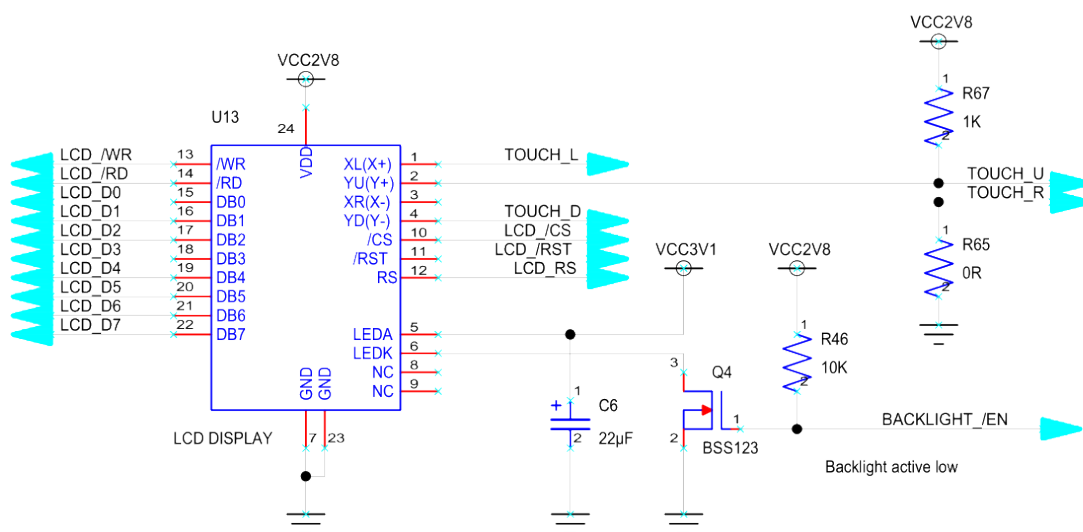
EXTI_LineCmd povolení linky externího přerušení. Možné hodnoty jsou ENABLE nebo DISABLE.

Protože kombinace Ride7 (7.42.12.035), RKitARM for Ride7 (1.48.13.0029), CircleOS (4.32) při obsluze přerušení vedla k nekonečnému zacyklení v rutině EXTI2_IRQHandler (a to i v případě, že EXTI2_IRQHandler nebyl konfigurován), bylo nutné přemapovat funkce pro obsluhu přerušení, a tím dát linkeru jasně najevo, kde v paměti se obsluha přerušení nachází. Přemapování obslužné rutiny pro EXTI0_IRQn lze provést pomocí následující funkce:

```
UTIL_SetIrqHandler((EXTI0_IRQn << 2) + 0x40, preruseni_fpls);
```

3.2.8 LCD Displej

Pro komunikaci s mikrokontrolérem slouží datová sběrnice o šířce osmi bitů a 4vodičová řídicí sběrnice. Samostatně je veden signál pro reset a pro podsvětlení displeje. Protože se o řízení LCD stará CirceOS, není komunikace mezi mikrokontrolérem a LCD blíže popsána. Do programu bylo přidáno ovládání podsvětlení kvůli snížení spotřeby. Vstupně-výstupní brána LEDK je označena BACKLIGHT_/EN a v programu namapována na PB.08. Díky tomu lze v programu volat funkci `LCD_SetBackLightOn()` a `LCD_SetBackLightOff()`.



Obr. 3.9: Schéma připojení LCD k STM32

3.2.9 Vykreslování obrázků programem

Aby bylo možné programem vykreslovat obrázky, je nutné je nejdříve převést do formátu hlavičkového souboru, který lze následně vykreslit na LCD displej pomocí funkce poskytované CircleOS. Program `image2primer` dokáže převést obrázky (jako `bmp`, `jpg`, `gif`, `png`) do tohoto formátu. Takto vytvořený hlavičkový soubor obsahuje informace o každém pixelu zpracovávaného obrázku. Pro každý pixel obrázku je zapotřebí dva bajty, maximální velikost obrázku je proto kolem 50x50 pixelů.

Vytvoření hlavičkového souboru z obrázku

- spustit aplikaci `image2primer.exe`
- načíst obrázek pro převod tlačítkem "Load image"
- zvolit barvu pozadí (standardně černá)
- tlačítkem "Convert" zahájit konverzi, zvolit místo pro uložení souboru (*.h)

Použití obrázku v programu

Do sekce s veřejnými proměnnými je nutné vložit deklaraci:

```
static const u16 image[] = {  
    #include "obrazek.png.h"  
};
```

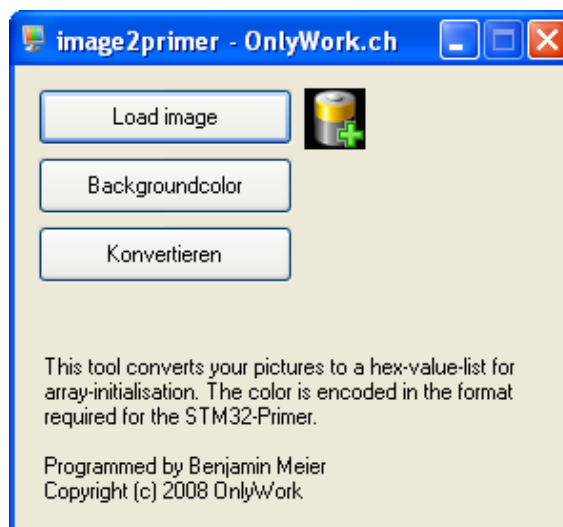
Kód 3.4: Deklarace obrázku

Kdekoliv v programu lze následně volat funkci, která vykreslí obrázek:

```
DRAW_SetImage(image, coord_t x, coord_t y, coord_t width, coord_t height);
```

Kód 3.5: Funkce pro vykreslení obrázku

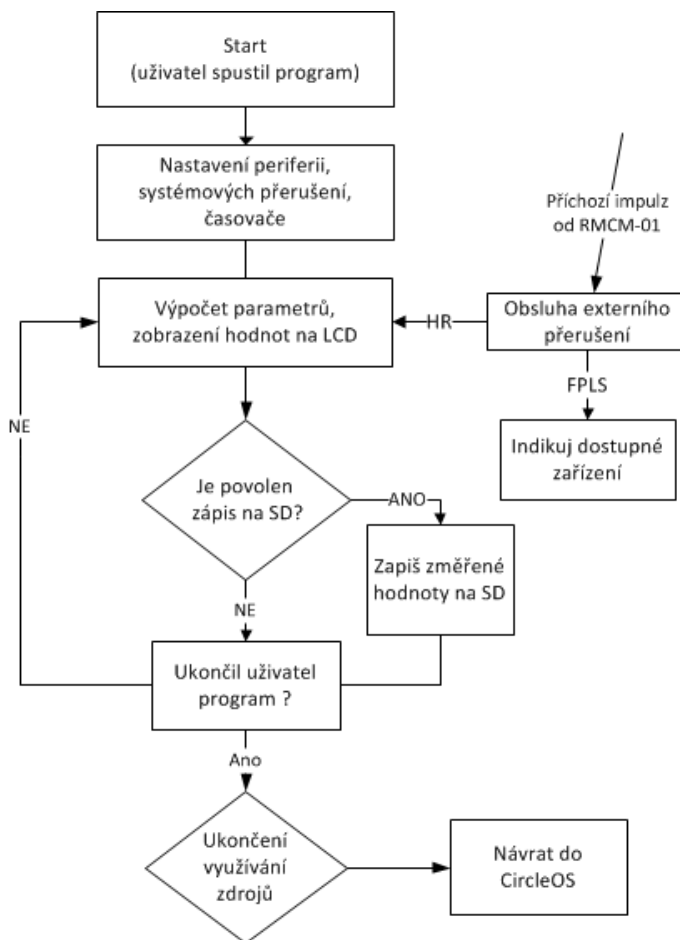
Program vyžaduje nainstalovaný .NET-Framework verze minimálně 2.0. Program lze stáhnout na stránkách <http://www.stm32circle.com/projects/project.php?id=52>.



Obr. 3.10: Program image2primer

3.3 Program

Samotný program pro zpracování a ukládání měřených dat z bezdrátového přijímače lze jednoduše popsat diagramem na obrázku 3.11.



Obr. 3.11: Princip programu

3.3.1 Chování programu

Pokud uživatel spustí program z hlavní nabídky, CircleOS zavolá funkci:

```
enum MENU_code Application_Ini (void)
```

ve které je definováno jaký HW aplikace využije. Nastavují se zde vstupně-výstupní brány, přerušeni a také se hlídá minimální verze CircleOS, pod kterou lze program provozovat. Tuto funkci volá CircleOS pouze jednou, a to právě při spuštění aplikace.

Následně je cyklicky volána funkce:

```
enum MENU_code Application_Handler (void)
```

ve které se zpracovávají naměřené hodnoty. Každým průchodem této funkce se kontroluje počet naměřených hodnot, možnost zápisu na SD kartu (případně zápis) a také se vypisují vypočtené hodnoty na LCD display.

Zastavit cyklické volání funkce `Application_Handler` a tím také ukončit celý program lze pomocí `MENU_Quit()`.

O zpracování příchozího externího přerušení se starají funkce

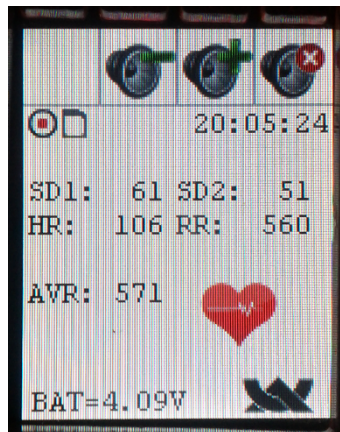
```
tHandler preruseni_fpls(void);
```

a

```
tHandler preruseni_hr(void);
```

3.3.2 Zobrazované hodnoty

Program nemá žádné uživatelské nastavení. Po spuštění začne zobrazovat na LCD display vypočítané hodnoty. Na obrázku 3.12 je vidět spuštěný program.



Obr. 3.12: Spuštěný program - naměřená data

SD1

Popis: Délka vedlejší poloosy obálky dat

Výpočet:

$$SD1 = \sqrt{\frac{1}{n} \sum_{i=1}^n \frac{((x_i - x_c) - (y_i - y_c))^2}{2}} \quad (3.2)$$

SD2

Popis: Délka hlavní poloosy obálky dat

Výpočet:

$$SD2 = \sqrt{\frac{1}{n} \sum_{i=1}^n \frac{((x_i - x_c) + (y_i - y_c))^2}{2}} \quad (3.3)$$

HR

Popis: Tepová frekvence

Výpočet:

$$HR = \frac{60000}{\frac{1}{n} \sum_{i=1}^n RR_n} \quad (3.4)$$

AVR

Popis: Průměrná délka RR intervalu v milisekundách

Výpočet:

$$AVR = \frac{1}{n} \sum_{i=1}^n RR_n \quad (3.5)$$

RR

Popis: Poslední zaznamenaný RR interval

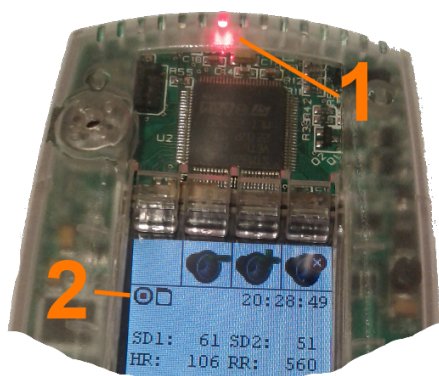
3.3.3 Výpočty programu

Následující blok programu ukazuje, jak program prochází pole RR intervalů pojmenované `RR_buf` a vypočítává hodnoty k zobrazení.

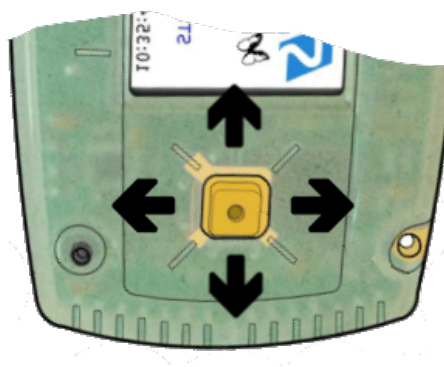
```
for ( j=1 ; j<puls ; j++) {
    HR += RR_buf[j];
}
RR_avrg = HR / ( puls-1);
HR = 60000 / (HR/puls-1);
for ( j=1 ; j<puls ; j++) {
    SD += (((RR_buf[j] - RR_avrg) - (RR_buf[j+1] - RR_avrg)) * ((RR_buf[j]
        ]-RR_avrg) - (RR_buf[j+1]-RR_avrg) )) /2;
}
SD1 = sqrtf( (SD / (puls-1)) );
for ( j=1 ; j<puls ; j++) {
    SD += (((RR_buf[j] - RR_avrg) + (RR_buf[j+1] - RR_avrg)) * ((RR_buf[j]
        ]-RR_avrg) + (RR_buf[j+1]-RR_avrg) )) /2;
}
SD2 = sqrtf( (SD / (puls-1)) );
```

3.3.4 Ovládací a indikační prvky programu

Program se ovládá pomocí joysticku umístěným pod LCD displej. Rozhraní nedisponuje žádným menu pro nastavení. Po spuštění programu jsou proměnné nastaveny tak, aby nebyl prováděn zápis na SD kartu. Tuto funkci je možné povolit pouze za předpokladu, že je SD karta dostupná. Zápis se povoluje stiskem joysticku doleva. Prvky pro informování uživatele jsou umístěny do horního levého rohu. Na LCD displej je vypisován i aktuální čas (pravý horní roh) a stav baterie.



Obr. 3.13: Indikační prvky



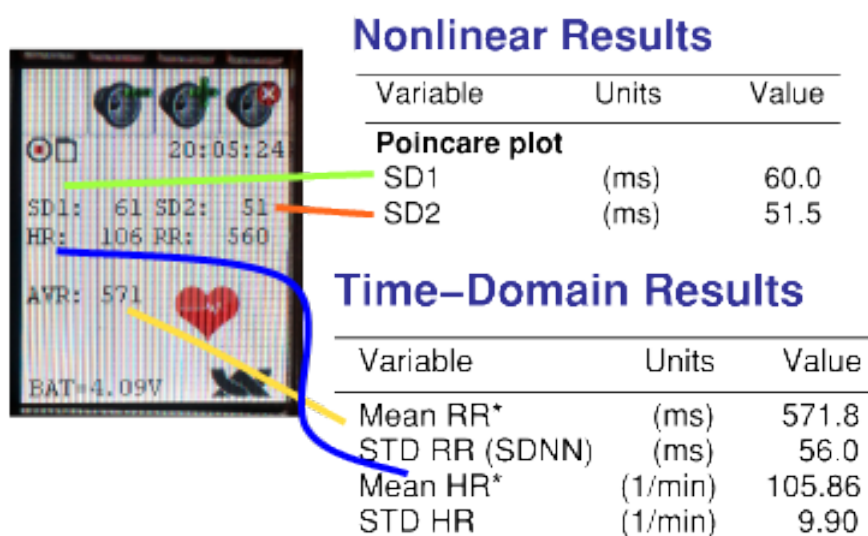
Obr. 3.14: Ovládací prvky

Směr	Funkce	Popis
Nahoru	LCD_SetBackLightOn()	Zapne podsvícení displeje
Dolů	LCD_SetBackLightOff()	Vypne podsvícení displeje
Doleva	start_record()	Začne ukládat měřená data na kartu
Doprava	stop_record()	Vypne zaznamenávání dat na kartu
Stisk středu	release_resource()	Ukončení programu

Tab. 3.8: Funkce ovládacích prvků

Označení	Prvek	Funkce
1	LED dioda	Indikace vysílajícího zařízení
2		Indikace spuštěného zápisu na SD kartu
2		Indikace dostupné SD karty

Tab. 3.9: Funkce indikačních prvků



3.3.5 Srovnání měřených dat

Obrázek 3.3.5 je výsledkem spojení výstupu z programu KubiosHRV (kompletní výstup v příloze D) a fotografií spuštěného programu na vývojovém kitu Primer2 (obrázek 3.12). Obrázek slouží pro názorné srovnání hodnot vypočtených programem a nezávislým software. Pro přehlednost byly data přepsána do tabulky 3.10. Odchytky jsou způsobené zaokrouhlováním při vypočítání dat v programu.

Proměnná	Program	KubiosHRV
SD1	61 ms	60 ms
SD2	51 ms	51,5 ms
HR	106 ms	105,86 ms
∅HR	571 ms	571,8 ms

Tab. 3.10: Srovnání vypočtených dat

4 ZÁVĚR

Cílem této bakalářské práce bylo sestrojít zařízení umožňující příjem a následné zpracování HRV signálu z hrudních pásů sporttesteru. Pro příjem jsem použil čidlo firmy Polar RMCM-01. Čidlo RMCM-01 bylo schopno pracovat v rozsahu napájecího napětí od 2,5 V do 2,8 V, které poskytuje vývojový kit Primer2. Jako zdroj signálu jsem použil hrudní pás dodávaný k eliptickému trenažeru inSPORTline Atlanta.

Návrh desky plošných spojů se v praxi osvědčil. Při programování a testování aplikace se ukázalo, že případné elektromagnetické rušení nemá vliv na funkčnost zařízení.

Program pro mikrokontrolér STM32F103VET6 jsem napsal pomocí vývojového prostředí Ride7 v jazyce ANSI C. Součástí vývojového prostředí jsou prostředky pro ladění aplikace, které mi umožnily náhled do obsahu paměti mikrokontroléru. Při programování jsem narazil na nefunkční vykonávání rutin externího přerušení. Vzhledem ke kombinaci více prvků jako například STM32 knihovny a nový CircleOS, zabralo nalezení chyby a vyřešení tohoto problému nejvíce času.

Mikrokontrolér se programuje z PC přes USB rozhraní. Pro komunikaci s LCD displejem a SD kartou jsem využil funkce převzaté z operačního systému CircleOS.

Realizované zařízení má oproti sporttesteru, který jsem měl v průběhu psaní bakalářské práce k dispozici, možnost měřená data zaznamenávat a zobrazovat vedle parametru HR také parametry SD1 a SD2. Naměřené hodnoty jsem použil jako vstupní data pro software KubiosHRV, díky kterému jsem ověřil správnost hodnot zobrazovaných realizovaným programem.

Zařízení by bylo možno rozšířit o graf zobrazující záznam délky RR intervalu v závislosti na čase pro názornější reprezentaci dat, nebo menu pro nastavení základních parametrů uživatele a porovnání již proběhlých měření.

LITERATURA

- [1] MILLER et al *Predicting max HR*. Časopis Medicine & Science in Sports & Exercise, 25(9), strana 1077-1081 (1993)
- [2] GANNONG, William F. *Přehled lékařské fyziologie*. 20. vydání. Praha: Galén, 2005. 890s. ISBN 80-7262-311-7.
- [3] TROJAN, Stanislav. *Lékařská fyziologie*. 4. vydání. Praha : Grada Publishing, a.s., 2003. 772 s. ISBN 80-247-0512-5.
- [4] GAMELIN, F. X., BERTHOIN S., BOSQUET L. *Validity of the Polar S810 Heart Rate Monitor to Measure R-R Intervals at Rest*, Ročník 38, číslo 5, pp. 887–893, 2006. [citováno 10.11.2012]. Dostupné z URL: <<http://sportexperts.org/pdf/PDF%2017.pdf>>
- [5] *Primer2 Extension Mechanical* [online]. [cit.14.11.2012]. Dostupné po registraci z URL: <<http://www.stm32circle.com/resources/stm32primer2.php#resources>>.
- [6] *STM32-Primer2 - User manual* [online]. poslední aktualizace 28.5.2010 [cit. 8.11.2012]. Dostupné po registraci z URL: <<http://www.stm32circle.com/resources/download.php?STM32-Primer2-Manual.pdf>>.
- [7] Polar Electro Europe BV *RMCM-01 Heart Rate Receiver Component* [online]. poslední aktualizace 11.11.2004 [cit. 17.2.2005]. Dostupné z URL: <<http://produceconsumerobot.com/heartfeltapparel/content/RMCM01.pdf>>.
- [8] *CircleOS v4.32 Documentation* [online]. 2012, poslední aktualizace 11.11.2004 [cit. 17.2.2005]. Dostupné z URL: <http://www.stm32circle.com/circleos_doc/index.html>
- [9] *CircleOS V4Conception document* [online]. 2012, poslední aktualizace 26.1.2012 [cit. 10.12.2012]. Dostupné z URL: <http://www.stm32circle.com/resources/download.php?CircleOS_Conception.pdf>
- [10] *Resting Heart Rate Table* [online]. [cit. 1.11.2012]. Dostupně z URL: <<http://www.topendsports.com/testing/heart-rate-resting-chart.htm>>.
- [11] *CircleOS* [online]. 2012, poslední aktualizace 26.1.2012 [cit. 10.12.2012]. Dostupné z URL: <<http://www.stm32circle.com/projects/download.php?id=101&file=Circle-OS-4.32.zip>>

- [12] STMicroelectronics, *UM0427 ARM®-based 32-bit MCU STM32F101xx and STM32F103xx firmware library* [online]. [cit. 28. 4. 2013]. Dostupné z URL: <https://my.st.com/public/STe2ecomunities/mcu/Lists/cortex_mx_stm32/Attachments/25258/UM0427%20USER%20MANUAL.pdf>
- [13] KARAVIDAS, M. *HRV Biofeedback Léčby Deprese* [online]. Dostupné z URL: <http://www.cdzjesenik.cz/0-HRV_Biofeedback_Deprese.pdf>
- [14] *STM32 Legacy Firmware Libraries* <https://my.st.com/public/STe2ecomunities/mcu/Lists/cortex_mx_stm32/Flat.aspx?RootFolder=%2Fpublic%2FSTe2ecomunities%2Fmcu%2FLists%2Fcortex_mx_stm32%2FSTM32%20Legacy%20Firmware%20Libraries>
- [15] *KubiosHRV Users Guide* [online]. Dostupné z URL: <http://kubios.uef.fi/media/Kubios_HRV_2.1_Users_Guide.pdf>
- [16] *Definitive Guide To The ARM Cortex M3* [online] Dostupné z URL: <http://tinymicros.com/mediawiki/images/7/75/Definitive_Guide_To_The_ARM_Cortex_M3.pdf>

SEZNAM SYMBOLŮ, VELIČIN A ZKRATEK

ADC Analog-to-Digital Converter

AHB Advanced High-performance Bus

ANSI C American National Standards Institute for the C programming language

APB Advanced Peripheral Bus

API Application Programming Interface

CPU Central Processing Unit

DAC Digital-to-Analog Converter

DPS Deska plošných spojů

EKG elektrokardiogram

GPIO General Purpose Input/Output

gcc GNU Compiler Collection

GNU GNU's Not Unix

GPL General Public License

GUI Graphical User Interface

HR Heart Rate

HRV Heart Rate Variability

IO Integrovaný obvod

LCD Liquid Crystal Display

MBR Master Boot Record

MEMS Micro-Electro-Mechanical Systems

MMC MultiMediaCard

MHR Maximum Heart Rate

NVIC Nested Vectored Interrupt Controller

PWM Pulse Width Modulation

RAM Random-Access Memory

RCC Reset and Clock Control

ROM Read-Only Memory

SMD Surface Mount Device

SMT Surface Mount Technology

SRAM Static Random Access Memory

SD Secure Digital

LED Light-Emitting Diode

SEZNAM PŘÍLOH

A Kompatibilita zařízení firmy Polar	46
B Schéma vnitřní struktury STM32F103x	47
C Zdroje hodinového kmitočtu	48
D Výstup programu KubiosHRV	49
E Obsah CD	50

A KOMPATIBILITA ZAŘÍZENÍ FIRMY POLAR

Zařízení Frekvence signálu	Polar T31 transmitter 5 kHz	Polar T31 coded transmitter 5kHz	Polar WearLink transmitter 5kHz	Polar WearLink Hybrid 5kHz + 2.4GHz	Polar WearLink transmitter W.I.N.D 2.4 GHz
Polar AW200 **	-	-	-	-	-
Polar AkompatibilníN300	z	kompatibilní	kompatibilní	kompatibilní	-
Polar AkompatibilníN500	z	kompatibilní	kompatibilní	kompatibilní	-
Polar AkompatibilníN700	z	kompatibilní	kompatibilní	kompatibilní	-
CS100, CS200, CS300, CS400	-	kompatibilní	kompatibilní	kompatibilní	-
CS500, CS600, CS600kompatibilní	-	-	-	kompatibilní	kompatibilní
F1/F2/F3	kompatibilní	*	*	*	-
F4 (non-coded)	kompatibilní	*	*	*	-
F4	z	kompatibilní	kompatibilní	kompatibilní	-
F5	kompatibilní	*	*	*	-
F6, F7, F11, F55	z	kompatibilní	kompatibilní	kompatibilní	-
F92ti	z	kompatibilní	kompatibilní	kompatibilní	-
FA20 **	-	-	-	-	-
FS1/FS2/FS3	kompatibilní	*	*	*	-
FS2c/FS3c	z	kompatibilní	kompatibilní	kompatibilní	-
FT1, FT2	z	kompatibilní	kompatibilní	kompatibilní	-
FT4, FT7	-	kompatibilní	kompatibilní	kompatibilní	-
FT40, FT60, FT80	-	kompatibilní	kompatibilní	kompatibilní	-
RCkompatibilní5	-	kompatibilní	kompatibilní	kompatibilní	kompatibilní
RS100	z	kompatibilní	kompatibilní	kompatibilní	-
RS200, RS300kompatibilní	-	kompatibilní	kompatibilní	kompatibilní	-
RS400	-	kompatibilní	kompatibilní	kompatibilní	-
RS800, RS800Ckompatibilní	-	-	-	kompatibilní	kompatibilní
S625kompatibilní, S725kompatibilní	-	-	kompatibilní	kompatibilní	-

Tab. A.1: Seznam kompatibilních zařízení firmy polar

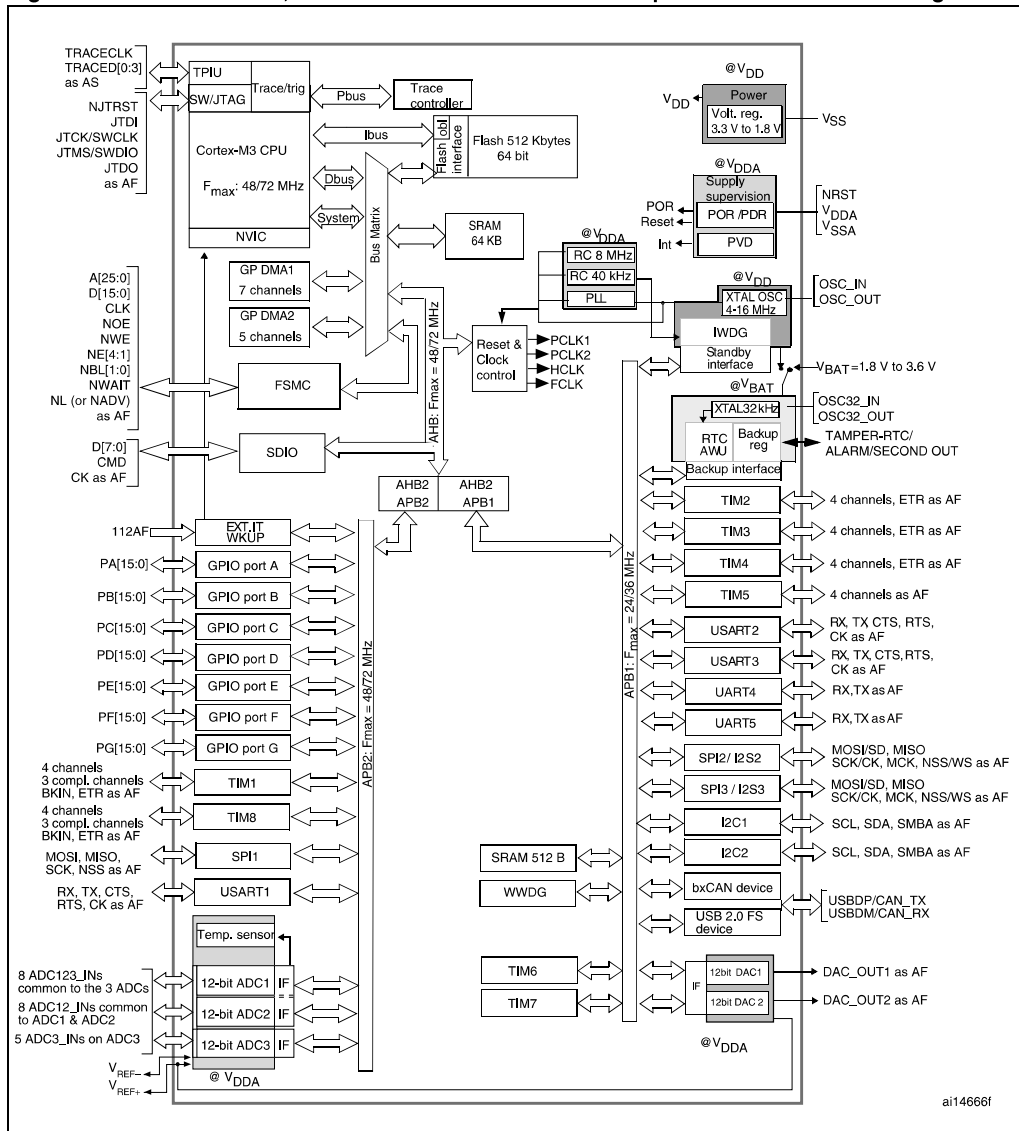
* - Toto zařízení používá nekódovaný přenos. Lze použít pro měření HR, ale bez výhod Polar OwnCode®.

z - Toto zařízení používá kódovaný přenos. Lze použít nekódovaný signál ale Polar OwnCode® funkce nejsou podporovány.

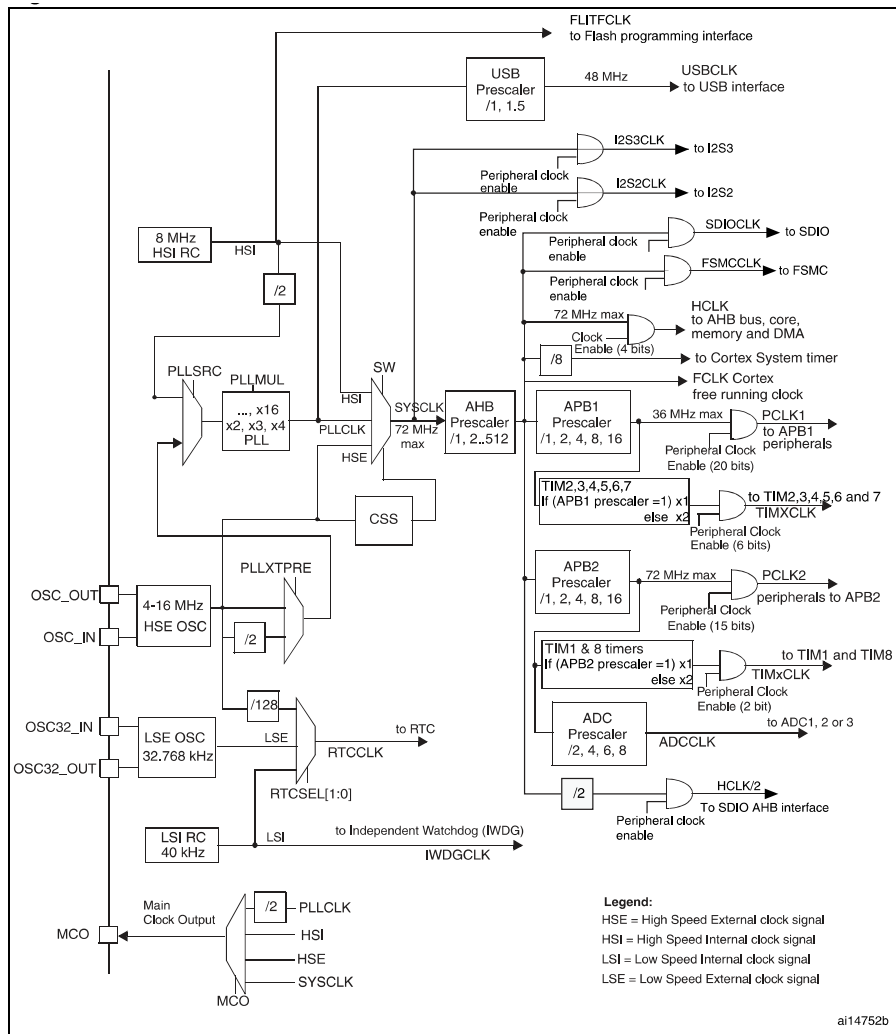
** - Toto zařízení nedovoluje monitorovat HRV.

B SCHÉMA VNITŘNÍ STRUKTURY STM32F103X

Figure 1. STM32F103xC, STM32F103xD and STM32F103xE performance line block diagram



C ZDROJE HODINOVÉHO KMITOČTU



D VÝSTUP PROGRAMU KUBIOSHRV

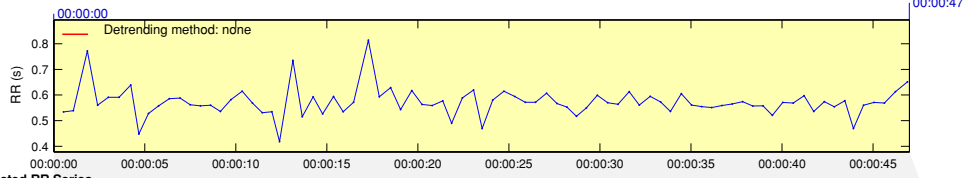
HRV Analysis Results

20_01_3.TXT - xx/xx/xx - xx:xx:xx

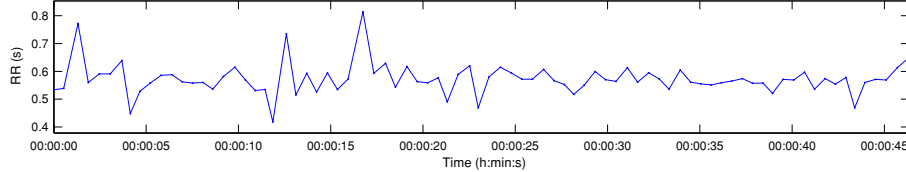
Page 1/1

RR Interval Time Series

Results for a single sample



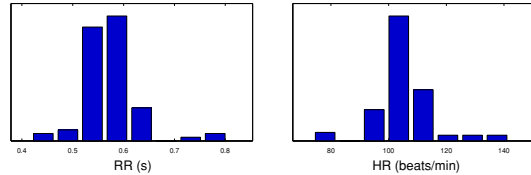
Selected RR Series



Time-Domain Results

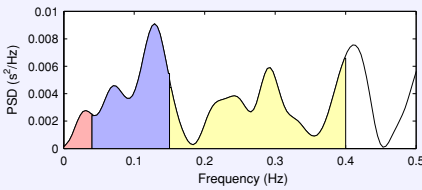
Variable	Units	Value
Mean RR*	(ms)	571.8
STD RR (SDNN)	(ms)	56.0
Mean HR*	(1/min)	105.86
STD HR	(1/min)	9.90
RMSSD	(ms)	84.4
NN50	(count)	25
pNN50	(%)	30.9
RR triangular index		5.857
TINN	(ms)	275.0

Distributions*



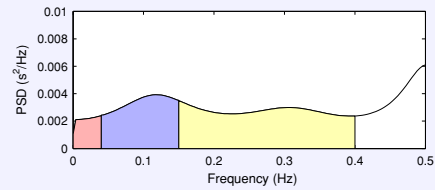
Frequency-Domain Results

FFT spectrum (Welch's periodogram: 256 s window with 50% overlap)



Frequency Band	Peak (Hz)	Power (ms²)	Power (%)	Power (n.u.)
VLF (0-0.04 Hz)	0.0313	70	5.1	
LF (0.04-0.15 Hz)	0.1289	578	41.6	43.9
HF (0.15-0.4 Hz)	0.3984	727	52.3	55.1
Total		1389		
LF/HF		0.796		

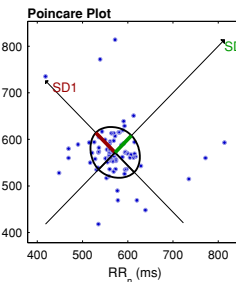
AR Spectrum (AR model order = 16, not factorized)



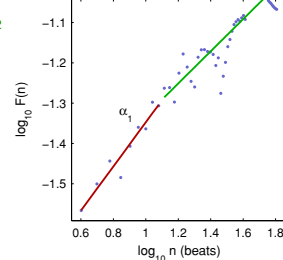
Frequency Band	Peak (Hz)	Power (ms²)	Power (%)	Power (n.u.)
VLF (0-0.04 Hz)	0.0391	86	7.5	
LF (0.04-0.15 Hz)	0.1172	373	32.5	35.1
HF (0.15-0.4 Hz)	0.1523	684	59.6	64.4
Total		1148		
LF/HF		0.545		

Nonlinear Results

Variable	Units	Value
Poincare plot		
SD1	(ms)	60.0
SD2	(ms)	51.5
Recurrence plot		
Mean line length (Lmean)	(beats)	11.27
Max line length (Lmax)	(beats)	32
Recurrence rate (REC)	(%)	39.07
Determinism (DET)	(%)	97.41
Shannon Entropy (ShanEn)		2.809
Other		
Approximate entropy (ApEn)		0.733
Sample entropy (SampEn)		1.537
Detrended fluctuations (DFA): α_1		0.548
Detrended fluctuations (DFA): α_2		0.399
Correlation dimension (D2)		2.048



Detrended fluctuations (DFA)



*Results are calculated from the non-detrended selected RR series.

E OBSAH CD

- `./CircleOS:` Složka obsahuje CircleOS
- `./Eagle:` Složka obsahuje DPS a schéma pro rozšiřující kartu
- `./image2primer:` Složka obsahuje program image2primer
- `./KubiosHRV:` Složka obsahuje instalační soubory KubiosHRV
- `./Program:` Složka obsahuje zdrojové kódy programu
- `./Raisonance:` Složka obsahuje instalační soubory vývojového prostředí
- `./Stm32:` Složka obsahuje knihovny firmy STM32