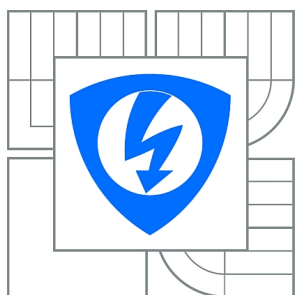


**VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ**

BRNO UNIVERSITY OF TECHNOLOGY



**FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH  
TECHNOLOGIÍ**

**ÚSTAV TELEKOMUNIKACÍ**

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION  
DEPARTMENT OF TELECOMMUNICATIONS

## **MĚŘIČ VARIABILITY TEPOVÉ FREKVENCE**

HEART RATE VARIABILITY MEASUREMENT DEVICE

**BAKALÁŘSKÁ PRÁCE**

BACHELOR'S THESIS

**AUTOR PRÁCE**

AUTHOR

**IVO SÜSS**

**VEDOUCÍ PRÁCE**

SUPERVISOR

**Ing. IVO STRAŠIL**

BRNO 2014

## **ABSTRAKT**

Cílem bakalářské práce bylo vyvinout zařízení, které bude přijímat signál ze sporttesterého pásu a následně bude tento signál zpracovávat. Měřič je postaven na vývojovém kitu STM32 Primer2. Pro příjem signálů ze sporttesterových pásů používá čidlo RMCM-01 firmy Polar. Měřič umožňuje měření parametrů srdeční variability, záznam na paměťovou kartu a práci s naměřenými daty z paměťové karty.

## **KLÍČOVÁ SLOVA**

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

## **ABSTRACT**

The aim of this thesis was to develop a device that will receive the signal from sporttester belt and then that signal will be processed. The meter is based on development kit STM32 Primer 2. To receive signals from sporttester belts uses sensor RMCM-01 Polar. The meter allows measurement of parameters of heart variability, recording on a memory card and work with the measured data from the memory card.

## **KEYWORDS**

Heart rate variability, Heart rate, RMCM-01, Polar, Primer2, CircleOS

## PROHLÁŠENÍ

Prohlašuji, že svou bakalářskou práci na téma Měříč 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 diplomové práce panu Ing. Ivovi 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

Úvod	10
<b>1 Teoretický rozbor</b>	<b>11</b>
1.1 EKG	11
1.1.1 Vlna P	11
1.1.2 Interval PQ(PR)	11
1.1.3 QRS Komplex	12
1.1.4 Úsek ST	12
1.1.5 Vlna T	13
1.1.6 Interval QT(RT)	13
1.1.7 Vlna U	13
1.2 HRV (Heart Rate Variability) – variabilita tepové frekvence	13
1.2.1 Autonomní nervový systém	13
1.2.2 HRV a krevní tlak	14
1.2.3 HRV a infarkt myokardu	14
1.2.4 HRV a nervový systém	14
1.2.5 HRV a srdeční arytmie	15
1.2.6 HRV a diabetes	15
1.2.7 HRV a selhání ledvin	15
1.2.8 HRV a pohlaví, věk	15
1.2.9 HRV a léky	15
1.2.10 HRV a kouření	16
1.2.11 HRV a alkohol	16
1.2.12 HRV a spánek	16
1.2.13 Závěr	16
1.3 Metody měření	16
1.3.1 RR interval	16
1.3.2 Tepová frekvence	17
1.3.3 Časová analýza	17
1.3.4 Geometrická metoda	18
1.3.5 Poincarého mapa	19
1.4 Vývojové prvky	20
1.4.1 Vývojový kit STM32 Primer 2	20
1.4.2 Čidlo a sporttesterový pás	21
1.4.3 CircleOS	23
1.4.4 Vývojové prostředí Raisonance Ride7	24
1.4.5 Software pro zpracování naměřených dat KubiosHRV	25

<b>2</b>	<b>Realizace</b>	<b>26</b>
2.1	Seznam vlastních funkcí . . . . .	27
2.2	Seznam funkcí ze CircleOS . . . . .	29
2.3	SD karta . . . . .	29
2.4	LED . . . . .	30
2.5	Časovač TIM . . . . .	30
2.6	NVIC . . . . .	32
2.7	EXTI . . . . .	32
2.8	Vykreslování obrázků . . . . .	34
2.9	Popis programu a jeho ovládání . . . . .	35
2.9.1	Spuštění . . . . .	35
2.9.2	Měření . . . . .	35
2.9.3	Stará data . . . . .	36
2.9.4	Konec . . . . .	39
2.9.5	Popis indikace a ovládání . . . . .	39
2.10	Srovnání naměřených dat . . . . .	40
<b>3</b>	<b>Závěr</b>	<b>42</b>
	<b>Literatura</b>	<b>43</b>
	<b>Seznam symbolů, veličin a zkratk</b>	<b>44</b>
	<b>Seznam příloh</b>	<b>45</b>
A	Vnitřní schéma STM32 Primer2	46
B	Výstupní list z programu Kubios HRV pro můj měřič	47
C	Výstupní list z programu Kubios HRV pro EKG měřič	48
D	Obsah CD	49

# SEZNAM OBRÁZKŮ

1.1	Znázornění tvaru EKG . . . . .	11
1.2	Znázornění junkčního bodu . . . . .	12
1.3	RR interval . . . . .	17
1.4	Geometrické znázornění RR intervalů . . . . .	19
1.5	Ukázka Poincarého mapy . . . . .	19
1.6	Ukázka a popis vývojového kitu STM32 Primer 2 . . . . .	20
1.7	Čidlo RMCM-01 firmy Polar . . . . .	21
1.8	Ukázka kompatibilního pásu Polar T31 . . . . .	22
1.9	Vývojové prostředí Raisonance Ride7 . . . . .	24
1.10	Hlavní okno programu KubiosHRV . . . . .	25
2.1	Popis zobrazených hodnot . . . . .	26
2.2	Program Image2primer.exe . . . . .	34
2.3	Hlavní menu . . . . .	35
2.4	Měření . . . . .	35
2.5	Obsah paměťové karty . . . . .	36
2.6	Možnosti zpracování souboru . . . . .	37
2.7	Vykreslená Poincarého mapa . . . . .	37
2.8	Výpis naměřených dat ze souboru . . . . .	38
2.9	Přehrání dat ze souboru . . . . .	38
2.10	Konec přehrávání dat ze souboru . . . . .	39
2.11	Popis indikačních prvků . . . . .	39
2.12	Výpis naměřených dat ze souboru . . . . .	40



# SEZNAM TABULEK

1.1	Tabulka vlastností čidla RMCM-01 . . . . .	22
1.2	Tabulka priorit obslužných přerušení a jejich popis . . . . .	23
1.3	Ukázka formátu zápisu dat . . . . .	25
2.1	Tabulka hodnot hodinového kmitočtu časovače TIM . . . . .	31
2.2	Tabulka možností čítání časovače TIM . . . . .	31
2.3	Seznam linek externího přerušení . . . . .	33
2.4	Typy událostí externího přerušení . . . . .	33
2.5	Možnosti spouště externího přerušení . . . . .	33
2.6	Popis indikačních prvků . . . . .	39
2.7	Tabulka srovnání naměřených dat . . . . .	41

# ÚVOD

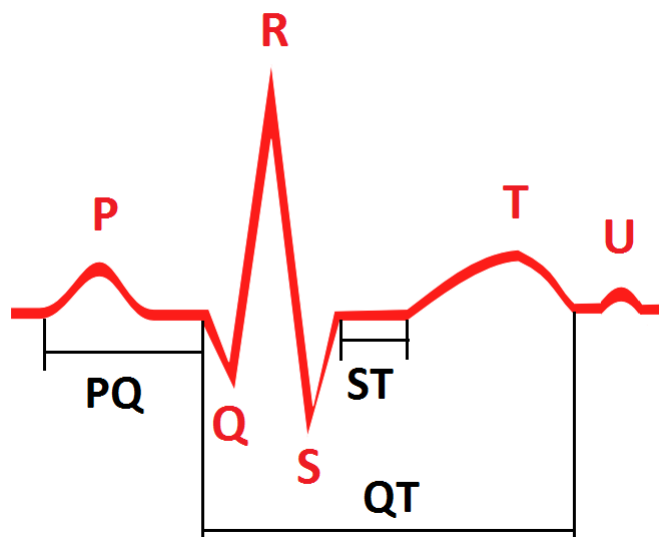
Cílem práce je naprogramovat program pro zpracování HRV (Heart Rate Variability) signálu z hrudních pásů sporttesteru. Program bude vyvíjen na vývojovém kitu STM32 Primer 2. Pro příjem signálů z hrudních pásů sporttesteru bude použito čidlo RMCM-01 firmy Polar. Program bude vypočítávat tepovou frekvenci a některé parametry její variability, které bude ukazovat na displayi. Bude možné naměřené hodnoty ukládat na paměťovou kartu pro další zpracovávání na počítači v programu KubiosHRV.

V první části práce se věnuji teoretickému rozboru EKG signálu a variabilitě tepové frekvence, popisu vývojového kitu STM32 Primer 2 včetně jeho vlastního operačního systému CircleOS, vývojovému prostředí Ride7 a programu KubiosHRV.

V druhé části se věnuji popisu způsobu měření HRV parametrů a popisuji mnou vytvořený program a jeho ovládání. Na konci ověřuji správnost měření.

# 1 TEORETICKÝ ROZBOR

## 1.1 EKG



Obr. 1.1: Znázornění tvaru EKG

Na EKG (obr. 1.1) rozeznáváme následující vlny, kmity a intervaly:

vlny: P, T, U

kmity: Q, R, S

interval PQ (PR), úsek ST, interval QT

### 1.1.1 Vlna P

Ukazuje depolarizaci srdečních síní. Většinou je to pozitivní kulovitá vlna, jejíž trvání nepřesáhne 0,1 sekundy a její výška je maximálně 0,25 mV. Vlna P nejčastěji chybí u fibrilaci síní.

### 1.1.2 Interval PQ(PR)

Je to interval od začátku vlny P k začátku kmity Q nebo R. Je to doba za kterou se dostane vzruch ze síní, převodní soustavou až do komorové svaloviny. Interval PQ(PR) většinou trvá 0,12-0,20 sekund.

### 1.1.3 QRS Komplex

QRS komplex trvá 0,06-0,1 sekund

#### Kmit Q

První negativní kmit, předcházející kmit R. Znázorňuje depolarizaci septa a papilárních svalů. Normální kmit Q trvá maximálně 0,03 sekund. Nepřesahuje výchylky R kmitu a je hluboký 0,3 mV.

#### Kmit R

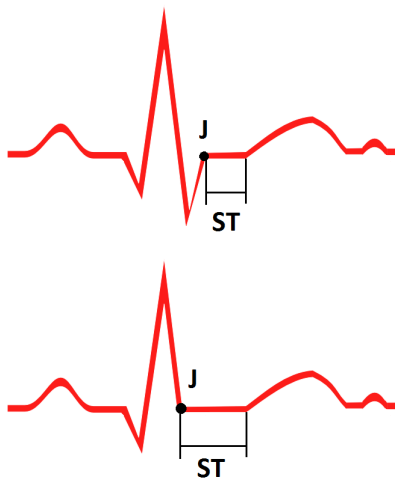
Je vždy pozitivní, pokud je více pozitivních kmitů označují se přidaným apostrofem R' R". Velikost kmitu R je při detekci elektrodou v končetinách do 1 mV, na hrudníku do 2,5 mV. U mladistvých do 0,7 mV a 3,5 mV.

#### Kmit S

Je negativní a přichází po kmitu R.

### 1.1.4 Úsek ST

Je to interval od konce QRS komplexu do začátku vlny T. V tomto intervalu se komory repolarizují. Nachází se zde junkční bod J (obr. 1.2), v místě konce QRS komplexu a začátku úseku ST. Normálně se junkční bod nachází na stejné linii jako konec kmitu Q.



Obr. 1.2: Znázornění junkčního bodu

### 1.1.5 Vlna T

Je také jako úsek ST, projevem repolarizace komor. Normální vlna T je asymetrická, pozvolna stoupá a na vrcholu rychleji klesne, trvá 0,2 sekundy a velikost má 0,2-0,8 mV.

### 1.1.6 Interval QT(RT)

Představuje elektrickou systolu, stažení srdeční svaloviny. Je to čas od začátku kmitu Q, nebo R pokud Q není, do konce vlny T. Interval QT velmi ovlivňuje tepová frekvence, proto se užívá QTc, což je hodnota QT s korekcí na frekvenci c. Trvání QTc je 0,34-0,42 sekundy, u žen je delší než u mužů a s rostoucím věkem se prodlužuje.

### 1.1.7 Vlna U

Je vidět jen někdy, vždy je menší než vlna T a vysoká je maximálně 0,05-0,1 mV. Není jasné co tuto vlnu způsobuje.

## 1.2 HRV (Heart Rate Variability) – variabilita tepové frekvence

Variabilita tepové frekvence slouží k diagnostice stavu autonomního nervového systému. Pomocí ní lze: zjistit stav pacienta po infarktu myokardu, predikovat diabetes, monitorovat účinnost podávaných léků, monitorovat přijetí či odpírání implantovaného orgánu, zjistit zda pacient kouří či je alkoholik, zkoumat fáze spánku. Protože variabilita tepové frekvence reflektuje stav nervového systému má na ní tedy vliv i okolní vjemy jako hluk či počasí.

### 1.2.1 Autonomní nervový systém

Skládá se ze sympatické a parasympatické části. Sympatika jsou napřílad reakce na stres, cvičení a srdeční choroby. Sympatické aktivity zvyšují rychlost tepu. Parasympatické aktivity snižují rychlost tepu a jsou to například projevy vnitřních orgánů, traumata, alergické reakce a vdechování dráždivých látek. Parasympatika poskytují regulační rovnováhu ve fyziologických autonomních funkcích. [1] Souhrn stimulací sympatik a parasympatik upravují srdeční frekvenci (interval RR) na různých frekvencích. Sympatická aktivita upravuje srdeční frekvenci v rozsahu 0,04-0,15 Hz. Parasympatická aktivita mění srdeční frekvenci od 0,15 do 0,4 Hz.

### 1.2.2 HRV a krevní tlak

EKG důkaz hypertrofie levé srdeční komory zvyšuje riziko onemocnění a smrti. Variabilita tepové frekvence je velmi snížena u pacientů s hypertrofií levé komory srdeční sekundující hypertenzi nebo onemocnění aortové chlopně. Aktivita srdečních vagových nervů je ovlivněna arteriálním baroreflexem. Amplituda respirační sinusové arytmie, zjištěné z HRV, koreluje s citlivostí baroreflexu, která se snižuje u pacientů s hypertenzí a cukrovkou. Toto snižování citlivosti baroreflexu koreluje s hypertrofií levé srdeční komory. Byl hledán vztah mezi výkyvy krevního tlaku a srdeční frekvence, jak ve frekvenční oblasti tak pomocí diferenčních rovnic, ale výsledky nebyly uspokojivé a nebylo z nich možno původní signál plně obnovit. Nicméně European Society of Hypertension, oddělení kardiovaskulární variability a baroreflexie, vytvořilo databázi EUROBAVAR pro testování a porovnávání metod. Jeden z navrhovaných vztahů, křížená korelace baroreflexové citlivosti na spontánním krevním tlaku a HRV, používající data z EUROBAVAR, lze považovat za metodu pro experimentální i klinické použití, protože výsledky korelují v blízkosti průměrných dat z databáze EUROBAVAR.[1]

### 1.2.3 HRV a infarkt myokardu

Větší sympatetická aktivita a snížená parasympatetická srdeční kontrola byla objevena u pacientů s akutním infarktem myokardu. Sympatetická aktivita snižuje fibrilační práh a předurčuje komorovou fibrilaci. Vagová aktivita zvyšuje práh na ochranu proti maligní komorové tachyarytmii. Stupeň respirační sinusové arytmie ukazuje lineární vztah s parasympatetickou srdeční kontrolou. Díky tomu lze HRV použít k prognóze u pacientů po infarktu myokardu, protože bylo zjištěno, že HRV klesá po posledním infarktu myokardu.[1]

### 1.2.4 HRV a nervový systém

Poruchy centrálního a periferního nervového systému, mají vliv na HRV. Všechny běžné cyklické změny v srdečním tepu jsou sníženy při závažném poškození mozku nebo při depresi. Význam analýzy HRV u psychiatrických poruch, vyplývá ze skutečnosti, že je možné snadno detekovat sympato-vagové rovnováhy (relativní cholinergní a adrenergní modulace HRV), pokud v nich existuje patologie. Zde jsou ale konfliktní zprávy, protože je prokázáno, že u fyzicky zdravých dospělých osob s depresí je HRV stejná jako u fyzicky zdravých dospělých osob bez deprese.[1]

### 1.2.5 HRV a srdeční arytmie

Kardiovaskulární systém je považován za nelineární. Byly pokusy [1] :

- O nelineární analýzu EKG a HRV, které prokázali, že koordinovaná mechanická činnost v srdci v průběhu ventrikulární fibrilace může být vidět v EKG pomocí vlnkové transformace.
- Bylo použito nelineární dynamické modelování v EKG k detekci arytmie.
- Nelineárními technikami a umělou inteligencí byly HRV signály klasifikovány do různých skupin.
- Pomocí autoregresivního modelování byly srdeční arytmie klasifikovány do šesti tříd.

### 1.2.6 HRV a diabetes

Diabetes může způsobit vážné autonomní dysfunkce. Na základě studií bylo zjištěno, že autonomní aktivita byla snížena u pacientů s diabetem dříve, než byly zřejmé klinické příznaky neuropatie.[1]

### 1.2.7 HRV a selhání ledvin

Vápník negativně koreluje s průměrnou dobou RR intervalu a normalizovanému výkonu vysokých frekvencí po hemodialýze. Model baroreflexní kontroly krevního tlaku navrženého na základě zpoždění diferenciální rovnice byl použit k predikci adaptace krátkodobé kardiovaskulární kontroly u chronických selhání ledvin. To ukazovalo, že pacienti s chronickým selháním ledvin měli průměrný výkon v pásmu nízkých frekvencí vyšší a v pásmu vysokých frekvencí nižší než zdravé subjekty.[1]

### 1.2.8 HRV a pohlaví, věk

U novorozenců, menší variaci srdečního tepu mají kluci než holky. S věkem HRV klesá a variace je větší u žen než mužů. Tento pokles začíná už v dětství. Děti mají vysokou sympatetickou aktivitu, která se rychle snižuje mezi pátým až desátým rokem života. Ukázalo se, že ve srovnání s muži, mají ženy nižší riziko k onemocnění ischemickou srdeční chorobou. [1]

### 1.2.9 HRV a léky

HRV se dá použít i pro zkoumání účinnosti léků. Na druhou stranu se musí při interpretaci HRV zvážit vliv léků které pacient užívá. Pomocí HRV se dá kvantifikovat účinnost léků na autonomní nervový systém. [1]

### 1.2.10 HRV a kouření

Studie prokázaly, že kuřáci mají zvýšené sympatetické aktivity a sníženou vagovou aktivitu. Jedním z mechanismů, kterými kouření ztěžuje kardiovaskulární funkce, je jeho vliv na řízení autonomního nerovnovážného systému. [1]

### 1.2.11 HRV a alkohol

HRV se snižuje požitím alkoholu. Studie prokázala vagovou neuropatii u mužů závislých na alkoholu. EKG indexy vagové aktivity měly výrazně nižší indexy srdeční vagové nervové činnosti u alkoholických subjektů než u zdravých dobrovolníků. [1]

### 1.2.12 HRV a spánek

HRV se mění v závislosti na fázi spánku. [1] Lze tedy s jeho pomocí hledat příčiny spánkových problémů.

### 1.2.13 Závěr

Jak můžeme vidět, variabilita tepové frekvence má široké využití a velký potenciál do budoucna. Protože odráží fyzický i psychický stav člověka, jejím použitím se dá vytvořit celkový zdravotní obraz osoby. Spolu s obrazem místa v kterém daná osoba bydlí či pracuje a s jejími návyky, lze najít „kámen úrazu“ jejích problémů. Popřípadě lze predikovat možné budoucí problémy, pokud se vezme její zdravotní obraz a namodeluje se prostředí místa kam se bude stěhovat nebo prostředí budoucího zaměstnání.

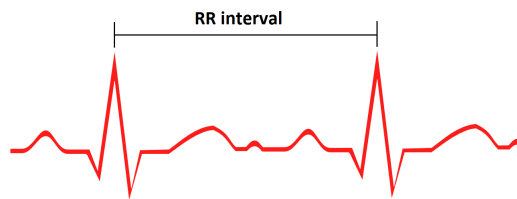
## 1.3 Metody měření

Způsobů měření variability tepové frekvence je hodně, budu se zabývat pouze časovou analýzou, geometrickou metodou s Poincarého mapami, protože je výpočetně nejjednodušší vzhledem k použitému vývojovému kitu. Další metody jsou frekvenční analýza, kde se používá Furierova transformace a vlnková transformace. Nelineární metody používající korelaci, Lyapunovův exponent, aproximační entropii, fraktální dimenze, Hurstův exponent.

### 1.3.1 RR interval

Je to základní prvek při výpočtech HRV a tepové frekvence při časové analýze. Počítány jsou RR intervaly z R kmitů před kterými je vlna P. Hodnotu RR intervalu získáme, odečteme-li čas posledního R kmitu od předcházejícího (Obr. 1.3).





Obr. 1.3: RR interval

### 1.3.2 Tepová frekvence

Tepová frekvence HR (Heart rate) udává počet stahů myokardu za jednu minutu.

$$HR = \frac{60}{RR\ interval} \quad [tep/min] \quad (1.1)$$

Tepová frekvence se mění dle fyzického zatížení a celkové kondice organismu. Maximální tepová frekvence udává jak velkou tepovou frekvenci dokáže organismus snést bez jakýchkoliv problémů. Velikost maximální tepové frekvence závisí na věku a kondici jednotlivce. Vzorců pro výpočet je více [8].

$$HR_{max} = 220 - věk \quad [tep/min] \quad (1.2)$$

$$HR_{max} = 206,3 - (0,711 * věk) \quad [tep/min] \quad (1.3)$$

$$HR_{max} = 217 - (0,85 * věk) \quad [tep/min] \quad (1.4)$$

$$HR_{max} = 206,9 - (0,67 * věk) \quad [tep/min] \quad (1.5)$$

$$HR_{max} = 202 - (0,55 * věk\ muže) \quad [tep/min] \quad (1.6)$$

$$HR_{max} = 216 - (1,09 * věk\ ženy) \quad [tep/min] \quad (1.7)$$

Nejpřesnějším způsobem zjištění maximální tepové frekvence je laboratorní test, kde kromě zjištění maximální tepové frekvence se měří i plno jiných parametrů např. spotřeba kyslíku. Test se provádí formou běhu na běžecím pásu (pro běžce) nebo cykloergometrií (pro cyklisty). [4];

### 1.3.3 Časová analýza

Základní prvek při časové analýze variability tepové frekvence je RR interval, z kterého se vypočítávají všechny zbylé parametry HRV a tepová frekvence.

Parametry[6][7]:

$\overline{RR}$  - Průměrná hodnota RR intervalu

$$\overline{RR} = \frac{1}{N} \sum_{i=1}^N RR_i \quad [ms] \quad (1.8)$$

SDNN – Standardní odchylka RR intervalu

$$SDNN = \sqrt{\frac{1}{N-1} \sum_{i=1}^N (RR_i - \overline{RR})^2} \quad [ms] \quad (1.9)$$

SDSD – Směrodatná odchylka mezi sousedními RR intervaly

$$SDSD = \sqrt{\frac{1}{N-1} \sum_{i=1}^{N-1} (|RR_i - RR_{i+1}| - \overline{RRdif})^2} \quad [ms] \quad (1.10)$$

$$\overline{RRdif} = \frac{1}{N-1} \sum_{i=1}^{N-1} (RR_i - RR_{i+1}) \quad (1.11)$$

RMSSD – Střední kvadratická odchylka

$$RMSSD = \sqrt{\frac{1}{N-1} \sum_{i=1}^{N-1} (RR_{i+1} - RR_i)^2} \quad [ms] \quad (1.12)$$

NN50 - Počet následujících rozdílů intervalů RR, které se od sebe liší o víc jak 50ms.

$$NN50 = \sum_{i=1}^N (|RR_{i+1} - RR_i| > 50ms) \quad [-] \quad (1.13)$$

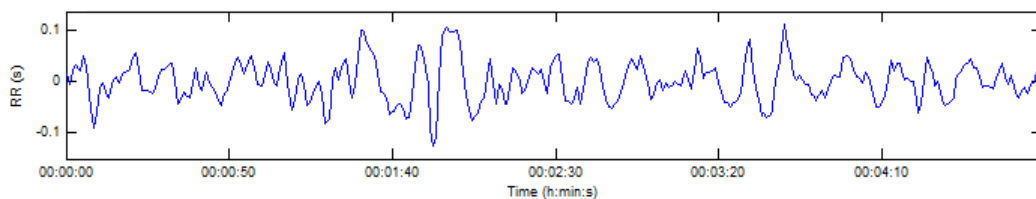
pNN50% - Počet následujících rozdílů intervalů RR, které se od sebe liší o víc jak 50 ms, vyjádřeno jako procento z celkového počtu všech analyzovaných EKG cyklů.

$$pNN50\% = \frac{NN50}{N} 100 \quad [\%] \quad (1.14)$$

Parametry SDNN, SDSD, RMSSD, pNN50% mají vysokou hodnotu u případů pre-ventikulární kontrakce, nemoci chorého sinu a fibrilaci síní.

### 1.3.4 Geometrická metoda

Geometrická metoda zobrazuje naměřené RR intervaly v podobě grafu, kde na ose X je délka záznamu RR intervalů a na ose Y je délka jednotlivého RR intervalu (obr. 1.4). Z obrázku není dobře patrná časová změna RR intervalů, proto se k lepšímu zobrazení naměřených dat používá Poincarého mapa.

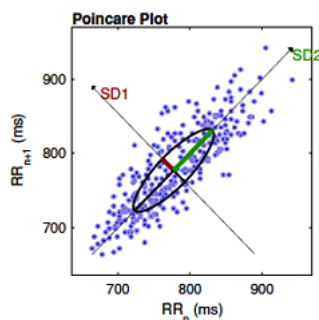


Obr. 1.4: Geometrické znázornění RR intervalů

### 1.3.5 Poincarého mapa

Znázorňuje časovou změnu RR intervalu, kde každý RR interval je vyneseno do grafu jako funkce předchozího RR intervalu. Poskytuje souhrnné informace o srdci stejně jako detailní EKG graf. Standardní odchylky jsou SD1 a SD2. SD1 znázorňuje rychlou variabilitu způsobenou respirační sinusovou arytmií. SD2 popisuje dlouhodobou variabilitu  $R - R_i$ . SD1/SD2 je poměr rychlé variability ku dlouhodobé. Tento poměr je větší v případě preventikulární kontrakce, fibrilaci síní, syndromu chorého sinu a komorové fibrilaci, díky větší variabilitě RR intervalu. A naopak je menší než průměr v případě kompletního srdečního bloku a ischemické/dilatační kardiomyopatie.[1]

Data zobrazená ve výsledné mapě představují elipsu  $RR_i = RR_{i+1}$ . Z jejího tvaru lze poznat délky RR intervalů. Pokud se elipsa blíží kruhu, RR intervaly jsou hodně podobné. Čím protáhlejší má elipsa tvar, tím jsou RR intervaly delší.



Obr. 1.5: Ukázka Poincarého mapy

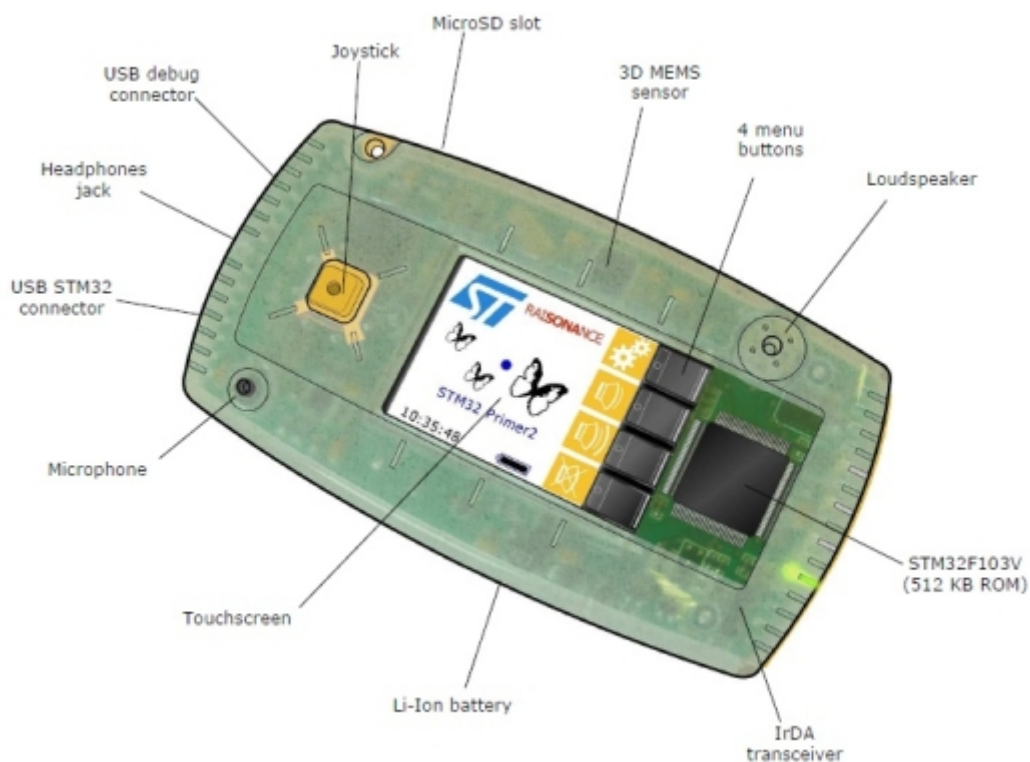
Vzorce SD1 a SD2 [6]:

$$SD1 = \sqrt{\frac{1}{2}SDSD^2} \quad (1.15)$$

$$SD2 = \sqrt{2SDNN^2 - \frac{1}{2}SDSD^2} \quad (1.16)$$

## 1.4 Vývojové prvky

### 1.4.1 Vývojový kit STM32 Primer 2



Obr. 1.6: Ukázka a popis vývojového kitu STM32 Primer 2

K realizaci této práce bude použit vývojový kit STM32 Primer 2, na platformě ARM Cortex M3, zobrazený na obrázku 1.6. Na něj se bude zaznamenávat a zpracovávat přijatý signál z hrudního pásu. Bude také sloužit k zobrazování aktuálně měřených dat a vyhodnocování již naměřených a uložených dat. Rozměry připomíná mobilní telefon či krabičku na mýdlo. Na přední straně je dotykový TFT display s rozlišením 128\*160 pixelů a s 16 777 216 barvami, čtyři ovládací tlačítka, 5-ti směrný joystick, mikrofon a reproduktor, indikační LED diody. Na spodní části je 3,5mm Jack konektor pro připojení sluchátek a dva mini USB porty (jeden pro nabíjení/programování a druhý pro připojení externích uživatelských komponent). Na vrchu je IrDA (infraport) rozhraní a vyústění pro DPS rozšiřujících karet/modulů. Po odejmutí zadního krytu nalezneme slot pro microSD paměťové karty, integrovanou baterii s jumperem pro její připojení a piny GPIO rozhraní pro připojení rozšiřujících karet/modulů. Tento kit také obsahuje 3osý akcelerometr, který se dá použít k ovládání GUI vlastního operačního systému CircleOS.

## Vlastnosti STM32 Primer2:

- Mikrokontrolér STM32103VE firmy STMicroelectronics  
ARM 32-bit Cortex™-M3 CPU, 72 MHz, 90 DMips s 1.25 DMips/MHz
- 512KB Flash ROM, 64KB SRAM
- ADC, DAC, PWM
- Komunikační periférie: USB 2.0, CAN, USART, SPI, I2C, LIN
- Dotykový TFT display 128\*160px, 24b
- 5-směrný joystick
- 4 ovládací tlačítka
- Mikrofon a reproduktor
- Slot na microSD paměťové karty
- Mini USB pro uživatelské periférie
- 3,5mm Jack konektor
- Akcelerometr - LIS3LV02DL od firmy STMicroelectronics
- IrDA (infraport) i ke komunikaci mezi dvěma zařízeními STM32 Primer2
- 20ti pinové GPIO rozhraní
- Li-Ion baterie s kapacitou 400mAh. Buď je zařízení napájeno přes USB a baterie se dobíjí pokud je připojena, nebo je zařízení napájeno z baterie pokud není připojeno k USB. Plně nabitá baterie vydrží 6 hodin [11].
- Vlastní operační systém CircleOS

### 1.4.2 Čidlo a sporttesterový pás



Obr. 1.7: Čidlo RMCM-01 firmy Polar

Aby bylo možné bezdrátově snímat data z hrudního pásu sporttesteru (obr. 1.8) je potřeba použít vlastní rozhraní, protože STM32 Primer 2 neobsahuje bezdrátový přijímač pro příjem dat z hrudních pásů sporttesteru, je tento nedostatek vyřešen jako rozšiřující modul do GPIO rozhraní s čidlem RMCM-01 firmy Polar, na obrázku 1.7. Čidlo obsahuje dva výstupy PFSL a HR. Pokud čidlo RMCM-01 zachytí vysílaný signál z některého z kompatibilních sporttesterů, vygeneruje na výstupu PFLS signál o velikosti 3V a délce 6ms, čímž signalizuje řídicí jednotce, v našem

případě kitu Primer 2, že je v okolí kompatibilní zařízení a je možno přijímat a zpracovávat jeho signál. Následně signál vyslaný ze sporttesteru, je přijmut čidlem RMCM-01 a transformován na jeho výstup HR, který generuje impulzy o velikosti 3V a délce 1ms, odpovídající kmitu R, které se dále v řídicí jednotce zpracovávají ve formě měření intervalu RR. [9]



Obr. 1.8: Ukázka kompatibilního pásu Polar T31

Parametr	Minimální	Tipický	Maximální	Jednotka
Dosah	80	92	100	cm
Napájecí napětí	2,5	3	3,4	V
Operační teplota	0	20	60	°C
Šířka impulzu HR		1		ms
Šířka impulzu FPLS		6		ms

Tab. 1.1: Tabulka vlastností čidla RMCM-01

### 1.4.3 CircleOS

CircleOS je jednoduchý operační systém, vyvinutý francouzskou firmou Raisonance, speciálně pro vývojový kit Primer 1 a Primer 2. Aktuální vydání CircleOS lze zdarma po registraci stáhnout z webu: <http://www.stm32circle.com/projects/circleos.php>. CircleOS je množina knihoven (API - Application Programming Interface), sloužící k ovládání hardwareových periférií vývojového kitu. Např. LCD, tlačítek, LED diod, Audia, irDA, a jiných. Přehled všech API systému CircleOS je na adrese: [http://www.stm32circle.com/circleos\\_doc/globals.html](http://www.stm32circle.com/circleos_doc/globals.html). Jádro OS tvoří funkce main(), která provádí prvotní nastavení periférií a systémových přerušení<sup>1</sup> vývojového kitu STM32 Primer 2. Rutiny přerušení spravuje přerušení nazvané SysTickHandler, které je voláno každou milisekundu. Pořadí priority rutin přerušení je v tabulce 1.2. Čas volání přerušení SysTickHandler lze nastavit ve funkci SysTick\_Configuration(). V CircleOS je možno načíst až 42 samostatných aplikací. Každá z aplikací má přístup k CPU a RAM. Z paměti RAM může využít 8kBytů (Rozsah adres 0x2000 0000 – 0x2000 1FFF) [2].

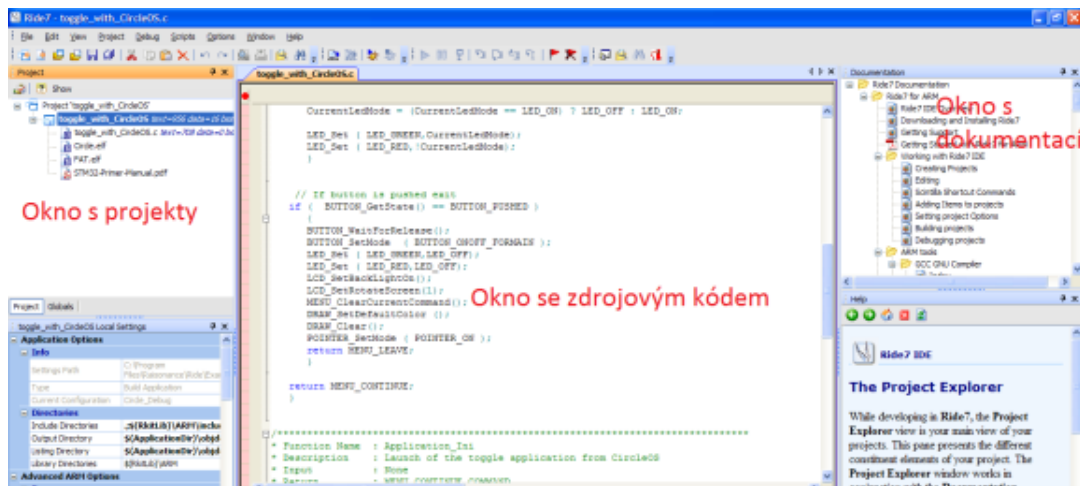
Priorita	Název	Popis
1	MEMS_Handler	Obsluha akcelerometru
2	LED_Handler	Obsluha LED diody
3	BUTTON_Handler	Obsluha tlačítek
4	BUZZER_Handler	Obsluha reproduktoru
5	MENU_Handler	Obsluha událostí z menu
6	POINTER_Handler	Obsluha ukazatelů do paměti
7	LCD_Handler	Obsluha LCD displeje
8	DRAW_Handler	Obsluha vykreslování obrázků
9	RTC_DisplayTime	Přesný čas
10	AUDIO_Handler	Obsluha zvukového zařízení
11	TOUCHSCR_Handler	Obsluha dotykové vrstvy displeje
12	TOOLBAR_Handler	Obsluha panelu nástrojů
13	POWER_Handler	Obsluha napájení

Tab. 1.2: Tabulka priorit obslužných přerušení a jejich popis

<sup>1</sup>Rutina přerušení je část kódu, která obsluhuje událost vzniklou vlivem přerušení.

## 1.4.4 Vývojové prostředí Raisonance Ride7

Pro programování vývojového kitu STM32 Primer 2 se používá vývojové prostředí Ride7 firmy Raisonance. Instalační soubory Ride7 a debuggeru Rkit-ARM jsou pouze pro windows, aktuální verze jsou dostupné na adrese [http://support-raisonance.com/extranet/tools/index.php?param=tool\\_id\\_7](http://support-raisonance.com/extranet/tools/index.php?param=tool_id_7). Po nainstalování je nutná registrace online, bez ní je funkčnost omezena na dobu 30 dnů. [10] Program se bude vyvíjet v jazyce ANSI C.



Obr. 1.9: Vývojové prostředí Raisonance Ride7

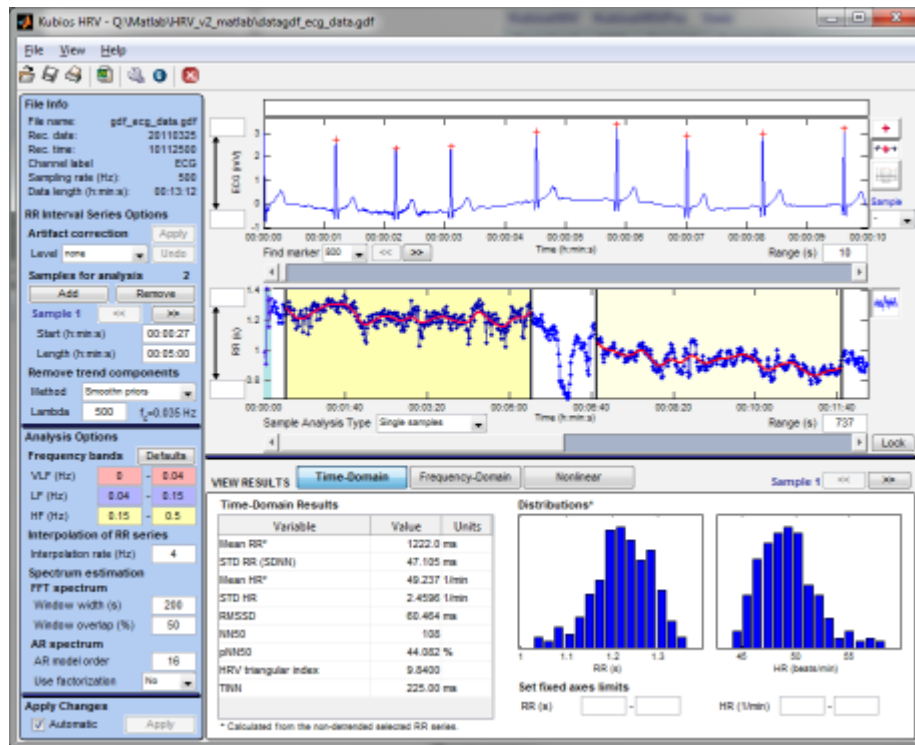


### 1.4.5 Software pro zpracování naměřených dat KubiosHRV

Na počítači se můžou naměřená data zpracovat programem KubiosHRV (obr. 1.10). Tento program lze stáhnout z <http://kubios.uef.fi/>. Uživatelská příručka je dostupná z adresy [http://kubios.uef.fi/media/Kubios\\_HRV\\_2.1\\_Users\\_Guide.pdf](http://kubios.uef.fi/media/Kubios_HRV_2.1_Users_Guide.pdf). Program lze nainstalovat jak na Windows tak i na Linux. Jako vstupní data lze použít záznam EKG nebo záznam RR intervalů v textovém souboru(ASCII) [6]. Měření v rámci této práce se bude provádět pomocí RR intervalů, takže vstupní data do tohoto programu budou časové hodnoty RR intervalů v textovém souboru. Tudíž program v STM32 Primer 2 bude ukládat na paměťovou kartu, do textového souboru, časové rozdíly příchodích R impulzů z čidla RMCM-01 v milisekundách (tab. 1.3).

RR1	0,123
RR2	0,145
RR3	0,167
...	...
RR <sub>n</sub>	0,189

Tab. 1.3: Ukázka formátu zápisu dat



Obr. 1.10: Hlavní okno programu KubiosHRV

## 2 REALIZACE

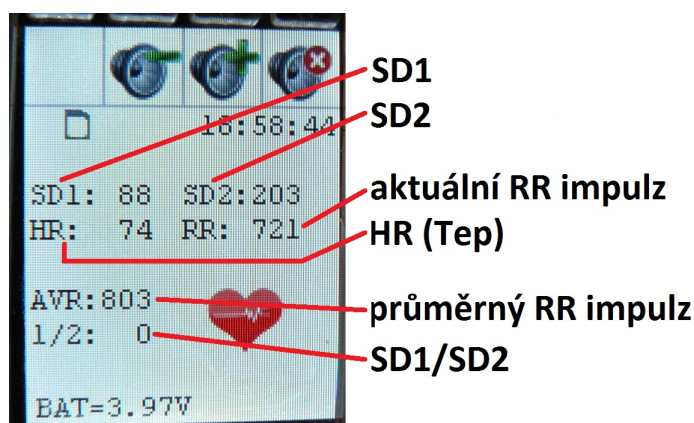
Měřič přijímá ze sporttesterového hrudního pásu HR signál. Délka pulzu se filtruje. Při každém přijetí pulzu se otestuje, zda je menší než 2000ms a větší než 350ms. Pokud ano, tak se zapíše do pole čas, uběhlý od přijetí minulého pulzu, časovač se vynuluje a počítá znovu. Velikost pole je 512 míst. Následující kód zobrazuje průchod polem a výpočet parametrů SD1, SD2, HR, SD1/SD2. Na obrázku 2.1, je popis měřícího okna s vypočtenými a zobrazenými hodnotami.

```
for (j = 1; j <= puls; j++)
{RR = RR + RR_buf[j];}
RR_a = RR / puls;
HR = 60000 / (RR / puls);

for (j = 1; j <= (puls-1); j++)
{
  if(RR_buf[j] > RR_buf[j+1])
  {RRii = (RR_buf[j] - RR_buf[j+1]);}
  else
  {RRii = (RR_buf[j+1] - RR_buf[j]);}
  RRdifn = RRdifn + (RR_buf[j] - RR_buf[j+1]);
  RRdif = RRdifn / (puls-1);
  SD = SD + ((RRii - RRdif)*(RRii - RRdif));
}
SDSD = sqrtf( (SD / (puls-1)) );
SD1 = (sqrtf( (SDSD*SDSD) / 2) / 2);

for (j = 1; j <= puls; j++)
{SDD = SDD + ((RR_buf[j]-RR_a)*(RR_buf[j]-RR_a));}
SDNN = sqrtf( (SDD / (puls-1)) );
SD2 = (sqrtf( (2 * (SDNN * SDNN)) - ((SDSD * SDSD) / 2)) / 2);

SDpul = SD1 / SD2;
```



Obr. 2.1: Popis zobrazených hodnot

## 2.1 Seznam vlastních funkcí

```
void Konec(void);
```

Funkce ukončí program a navrátí uživatele do CircleOS.

```
void Zpet_do_menu(void);
```

Funkce odpojí kartu a navrátí uživatele do hlavní nabídky programu.

```
void Zpet_karta(void);
```

Funkce vrací uživatele do nabídky práce se souborem.

```
void Info_o_souboru(void);
```

Funkce zobrazí informace o souboru, jeho název a počet záznamů které obsahuje. Je použito v nabídce práce se souborem.

```
void Zpracovani_souboru(void);
```

Funkce zobrazí informaci uživateli, že zpracovává soubor x s y záznamama. Je použito v každé ini funkci, při práci se souborem (Poincarého mapa, výpis dat, přehrání).

```
void Zpracovat(void);
```

Funkce je spouštěna při výběru souboru z paměťové karty, pouze vybraný soubor otevře a vykreslí nabídku s možnostmi jeho zpracování.

```
void Joystick(void);
```

Funkce "Zpět" při stisku joysticku směrem doleva. Je použita všude kde není možnost vrácení v nabídce.

```
void Vypocet(void);
```

Funkce vypočítává parametry SD1, SD2, Tep(HR) a průměrnou délku RR intervalu.

```
void Elipsa(void);
```

Funkce elipsa z vložených informací vykreslí, a otočí o 45°, elipsu do Poincarého mapy.

```
void Update_list(void);
```

Funkce k vykreslení a pohybování se v seznamu souborů na paměťové kartě.

```
void Null_prom(void);
```

Funkce nuluje všechny proměné použité ve funkci void Vypocet(void).

```
void Vypis_parametru(void);
```

Funkce vypíše parametry z funkce void Mereni(void);

```
void start_record(void);
```

Pokud je vložena paměťová karta, funkce vykreslí obrázek nahrávání, dle času nastaví jméno souboru, otevře jej a přepne proměnou `record` na hodnotu jedna, čímž povoluje zápis.

```
void stop_record(void);
```

Funkce vymaže obrázek nahrávání, uzavře soubor a přepne proměnou `record` na hodnotu nula, čímž zápis zakazuje.

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

Funkce pro nastavení časovače TIM3 na vzestupné počítání s periodou 65535.

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

Funkce pro nastavení adres přerušení (časovač TIM3 a PA.01), na kterých program najde rutinu pro zpracování těchto přerušení.

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

Funkce pro nastavení vstupně-výstupních bran rozhraní GPIO. Nastaví bránu PA.01 jako vstupní bránu k přijímání HR signálu z čidla. A bránu PB.08 jako funkci k řízení podsvícení displaye. Nakonec propojí bránu PA.01 s přerušením HR.

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

Funkce pro nastavení externího přerušení a jeho povolení.

```
void EXTI_DeInicializace(void);
```

Funkce pro zakázání externího přerušení.

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

Funkce volaná při přetečení časovače TIM3. Při nepřijímání HR signálu ze sport-testerového pásu po delší dobu, časovač přeteče přes svou maximální hodnotu (dobu) a signalizuje tak nečinnost měření a ukončuje zápis dat na SD kartu.

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

Funkce volaná při vzniku přerušení od HR signálu. Problikne červenou LED diodu k signalizaci přijmutí HR signálu, uloží hodnotu časovače TIM3 a vynuluje ho. Tato funkce měří RR interval.

## 2.2 Seznam funkcí ze CircleOS

```
void DRAW_DisplayString (u16 x, u16 y, const u8* ptr, len_t len);
void DRAW_DisplayVbat (coord_t x, coord_t y);
void DRAW_DisplayTime (coord_t x, coord_t y);
void DRAW_SetImage (const color_t * ptr, coord_t x, coord_t y, coord_t width, coord_t height);
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);
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);
```

Popis jednotlivých funkcí je na: [http://www.stm32circle.com/circleos\\_doc/](http://www.stm32circle.com/circleos_doc/)

## 2.3 SD karta

Práce s SD kartou jde pouze tehdy, pokud je karta vložena do slotu a řádně připojena a detekována. Připojení se provádí funkcí ze CircleOS:

```
FS_Mount(MMCSD_SDIO);
```

Návratová hodnota udává počáteční sektor MBR (Master Boot Sector) připojeného oddílu karty. Níže uvedeným kódem se provádí detekce připojené karty.

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

Otevření souboru pro čtení se provádí příkazem:

```
FS_OpenFile( &volume_info, SouborNaZpracovani, FS_READ, &file_info );
```

Otevření souboru pro zápis se provádí příkazem:

```
FS_OpenFile( &volume_info, SouborNaZpracovani, FS_WRITE, &file_info );
```

Zápis na SD kartu se provádí sérií příkazů uvedených níže. Zjistí se velikost souboru, poté se nastaví kurzor na konec souboru a následně se provede zápis obsahu proměnné `display_buffer` do které byly vloženy znaky `0x0d` (návrat kurzoru) a `0x0a` (nový řádek).

```

size = FS_Size( &file_info );
FS_Seek(&file_info, size);
display_buffer[8]=0x0d;
display_buffer[9]=0x0a;
FS_WriteFile(&file_info, (u8 *)display_buffer, &j, 10);

```

Čtení z SD karty se provádí obdobně. Příkazem `FS_Seek(&file_info, k);` se pomocí proměnné `k` nastaví kurzor na danou souřadnici (`k=0` je začátek souboru) a příkazem

```

FS_ReadFile(&file_info, Precteny_obsah, &q, 8);

```

se provede čtení následujících znaků. Počet přečtených znaků se určuje podle posledního parametru. V tomto případě 8, tedy když `k=0` tak se přečte prvních 8 znaků. Funkce pro zavření souboru:

```

FS_Close( &file_info );

```

Funkce pro odpojení paměťové karty:

```

FS_Unmount(MMCSD_SDIO);

```

## 2.4 LED

Zařízení indikuje červenou LED diodou příchozí HR impuls z čidla RMCM01 pomocí níže uvedeného kódu.

```

tHandler preruseni_hr(void)
{
if(EXTI_GetITStatus(HR_LINE) != RESET)
{
StateLED_HR = ( StateLED_HR == LED_ON ) ? LED_OFF : LED_ON;
LED_Set( LED_RED, StateLED_HR );
EXTI_ClearITPendingBit(HR_LINE);
}
}

```

## 2.5 Časovač TIM

Mikroprocesor obsahuje 8 časovačů označených jako TIM+číslo. Každý časovač se skládá ze 16-bitového čítače s možností automatické obnovy hodnoty <sup>1</sup> a programovatelnou děličkou hodinového kmitočtu. Mikroprocesor obsahuje 3 druhy časovačů:

- Základní – TIM6, TIM7
- Univerzální – TIM3, TIM3, TIM4, TIM5
- Pokročilé – TIM1, TIM8

---

<sup>1</sup>funkce auto-reload

**Pokročilé** a **univerzální** časovače se mohou použít k měření délky pulsu vstupního signálu nebo vytváření výstupní křivky (PWM, výstupní porovnávání, ...). Délku pulsu a periodu křivky lze nastavit, použitím děličky hodinového kmitočtu, na hodnoty od pár mikrosekund po několik milisekund.

**Základní** časovače mohou být použity k časovým operacím stejně jako pokročilé a univerzální časovače, ale mohou být také použity k DA (Digital-to-Analog) převodu. Základní časovače jsou vnitřně spojeny s DA převodníkem a umožňují ho řídit.

Popis struktury pro všechny časovače kromě TIM6 a TIM7:

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

**TIM\_Period** – Nastavení periody časovače. Tato hodnota se obnoví při přetečení časovače. Hodnota musí být v rozmezí 0x0000 – 0xFFFF.

**TIM\_Prescaler** – Nastavení děličky hodinového kmitočtu časovače. Hodnota musí být v rozmezí 0x0000 – 0xFFFF.

**TIM\_ClockDivision** – Nastavení hodinového kmitočtu pro časovač. Tab.2.1

Hodnota	Popis
TIM_CKD_DIV1	$T_{DTS} = T_{ck\_tim}$
TIM_CKD_DIV2	$T_{DTS} = 2 * T_{ck\_tim}$
TIM_CKD_DIV3	$T_{DTS} = 4 * T_{ck\_tim}$

Tab. 2.1: Tabulka hodnot hodinového kmitočtu časovače TIM

**TIM\_CounterMode** – Nastavení typu čítání časovače. Tab.2.2

Hodnota	Popis
TIM_Counter_Up	Inkrementace
TIM_Counter_Down	Dekrementace
TIM_Counter_CenterAligned1	-
TIM_Counter_CenterAligned2	-
TIM_Counter_CenterAligned3	-

Tab. 2.2: Tabulka možností čítání časovače TIM

**TIM\_RepetitionCounter** – Nastavení hodnoty opakování čítače. Vždy když odpočet

dosáhne nuly a přeteče, odčítání se restartuje od hodnoty N. U PWM v módu N+1 to odpovídá:

- PWM periodě v módu okrajového zarovnání (edge-aligned).
- PWM půl periodě v módu centrálního zarovnání (center-aligned).

Hodnota musí být v rozmezí 0x00 - 0xFF. Tento parametr slouží pouze časovačům TIM1 a TIM8.

## 2.6 NVIC

Nested vectored interrupt controller - slouží k povolování/zakazování IRQ přerušení.

- Podporuje až 81 přerušení (záleží na typu STM32 zařízení).
- Lze jím nastavit prioritu přerušení v rozmezí 0-15. Nejvyšší priorita je značena hodnotou 0, nejnižší 15.
- Prioritu přerušení lze měnit dynamicky za chodu.
- Lze přerušení uskupovat do skupin a podskupin.

Popis struktury:

```
typedef struct
{
    u8 NVIC_IRQChannel;
    u8 NVIC_IRQChannelPreemptionPriority;
    u8 NVIC_IRQChannelSubPriority;
    FunctionalState NVIC_IRQChannelCmd;
} NVIC_InitTypeDef;
```

NVIC\_IRQChannel – Výběr kanálu IRQ.

NVIC\_IRQChannelPreemptionPriority – Nastavení priority kanálu.

NVIC\_IRQChannelSubPriority – Nastavení subpriority kanálu.

NVIC\_IRQChannelCmd – Nastavení povolení kanálu, pomocí hodnot ENABLE (povolení) nebo DISABLE (zakázání).

## 2.7 EXTI

External interrupt/event controller se skládá z 19 detekčních linek, které mohou vyvolat požadavek na událost nebo přerušení. Každá linka může být nezávisle nakonfigurována na typ přerušení (puls nebo nepřetržitý signál) a typ spouště (náběžná hrana, sestupná hrana nebo obě).

Popis struktury:

```
typedef struct
{
    u32 EXTI_Line;
    EXTIMode_TypeDef EXTI_Mode;
    EXTIrigger_TypeDef EXTI_Trigger;
```



```
FunctionalState EXTI_LineCmd;
} EXTI_InitTypeDef;
```

EXTI\_Line – Nastavení linky externího přerušení. Tab.2.3

Hodnota	Popis
EXTI_Line0	Linka 0 externího přerušení
EXTI_Line1	Linka 1 externího přerušení
...	...
EXTI_Line18	Linka 18 externího přerušení

Tab. 2.3: Seznam linek externího přerušení

EXTI\_Mode – Nastavení typu reakce linky externího přerušení. Tab.2.4

Hodnota	Popis
EXTI_Mode_Event	Linka konfigurována na událost
EXTI_Mode_Interrupt	Linka konfigurována na přerušení

Tab. 2.4: Typy událostí externího přerušení

EXTI\_Trigger – Nastavení typu spouště. Tab.2.5

Hodnota	Popis
EXTI_Trigger_Falling	Reakce na sestupnou hranu
EXTI_Trigger_Rising	Reakce na vzestupnou hranu
EXTI_Trigger_Rising_Falling	Reakce na vzestupnou i sestupnou hranu

Tab. 2.5: Možnosti spouště externího přerušení

EXTI\_LineCmd - Nastavení povolení externího přerušení, pomocí hodnot **ENABLE** (povolení) nebo **DISABLE** (zakázání).

## 2.8 Vykreslování obrázků

Pokud chceme na display vykreslovat obrázky (např. bmp, jpg, gif, png,...) musí se převést do hlavičkového souboru (\*.h) pomocí Image2primer.exe (obr. 2.2) programu. Pouze převedené obrázky lze vykreslit na display funkcí ze CircleOS. Hlavičkový soubor obsahuje hodnotu, o velikosti 2 bajty, pro každý pixel. Maximální velikost obrázku je 50x50 pixelů. Ke spuštění programu je potřeba Microsoft .NET-Framework 2.0 nebo vyšší. Program je možno stáhnout na: <http://www.stm32circle.com/projects/project.php?id=52>

Vytvoření hlavičkového souboru se provede následně:

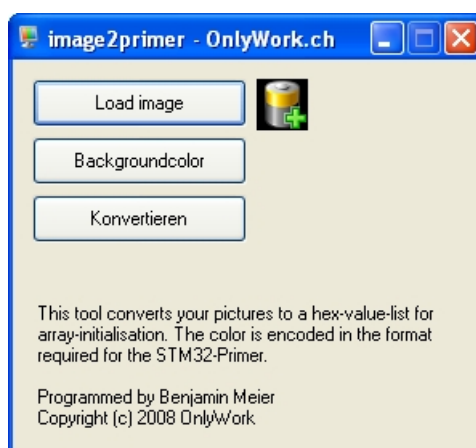
- Spustí se image2primer.exe
- Klikne se na "Load image" a vybere se obrázek, který chceme převést.
- Pokud chceme můžeme zvolit barvu pozadí obrázku.
- Klikne se na "Convert" a zadá se cesta kam se má uložit převedený obrázek (\*.h).

Vykreslení obrázku se provádí deklarací veřejné proměné ve tvaru:

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

Kdekoliv v programu je pak obrázek možno vykreslit pomocí funkce:

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



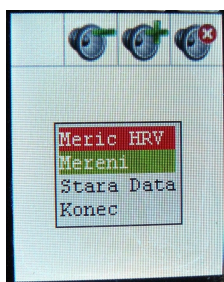
Obr. 2.2: Program Image2primer.exe

## 2.9 Popis programu a jeho ovládání

### 2.9.1 Spuštění

Při spuštění programu se volá funkce `enum MENU_code Application_Ini(void)`, která zjistí zda je daná verze CircleOS dostačující pro běh programu, nastaví rychlost volání aplikace a velikost displaye. Následně se cyklicky volá funkce `enum MENU_code Application_Handler(void)`, která vykreslí Hlavní nabídku (obr. 2.3) s možnostmi: Měření, Stará data a Konec.

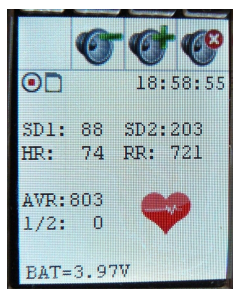
V nabídce se pohybuje pomocí joysticku směrem nahoru/dolů a vybraná nabídka se potvrdí stiskem joysticku.



Obr. 2.3: Hlavní menu

### 2.9.2 Měření

Při výběru Měření se zavolá funkce `enum MENU_code Mereni_Ini(void)`, kde se nastavují vstupně-výstupní brány, časovač TIM3 a testuje se přítomnost paměťové karty a nastavuje povolení k zápisu na ni. Následně se cyklicky volá funkce `enum MENU_code Mereni_Handler(void)`, kde se volá funkce `void Vypocet(void)`, ve které se zpracovávají naměřené hodnoty. Funkcí `void Vypis_parametru(void)` se vypisují vypočítané parametry na display (obr. 2.4). Kontroluje se zda je povolen zápis na kartu a popřípadě je proveden zápis.



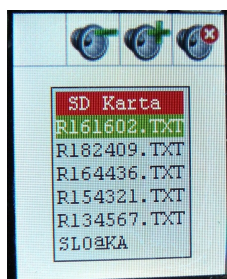
Obr. 2.4: Měření

V další části funkce se sleduje interakce uživatele s joystickem, v případě stisku joysticku nahoru/dolů se volají funkce sloužící k zapnutí nebo vypnutí podsvícení displeje (`LCD_SetBackLightOn()`; / `LCD_SetBackLightOff()`;). V případě stisku joysticku doleva či doprava se volají funkce `void start_record(void)` resp. `void stop_record(void)`, sloužící k povolení/zakázání možnosti záznamu RR intervalů na kartu. Při stisku joysticku se vše zakáže a ukončí, a uživatel je navrácen do Hlavní nabídky.

### 2.9.3 Stará data

Při výběru Stará data se se spustí funkce `enum MENU_code StaraData_Ini(void)`, která vymaže obsah displeje, nastaví rychlost volání aplikace, otestuje zda je přítomná paměťová karta, nastaví cestu do hlavního adresáře a vypíše seznam souborů, které se nachází na paměťové kartě. Následně se cyklicky spouští funkce `enum MENU_code StaraData_Handler(void)`, kde se testuje zda byl dříve otevřen nějaký soubor a pokud byl, tak se zavře. Dále se zde ovládá seznam souborů a složek na paměťové kartě (obr. 2.5), pokud byl vybrán soubor s příponou `*.txt` tak se spustí funkce `void Zpracovat(void)`.

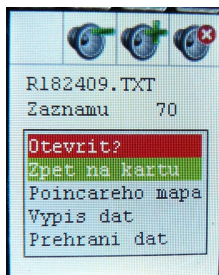
Ovládání seznamu se provádí joystickem nahoru/dolů a vybírá se stiskem joysticku. Joystickem doleva se provede funkce `void Zpet_do_menu(void)`, kde se provede odpojení paměťové karty a vykreslí se Hlavní nabídka.



Obr. 2.5: Obsah paměťové karty

Při vybrání souboru se provede funkce `void Zpracovat(void)`, kde se soubor otevře a následně zobrazí nabídku s možnostmi práce se souborem (obr. 2.6). V záhlaví se zobrazí název otevřeného souboru a počet záznamů které obsahuje.

V nabídce se pohybuje opět pomocí joysticku nahoru/dolů a potvrzuje se stiskem. Nabídka obsahuje návrat na paměťovou kartu kde se může vybrat jiný soubor, vykreslení Poincarého mapy, výpis dat nebo přehrání záznamu.



Obr. 2.6: Možnosti zpracování souboru

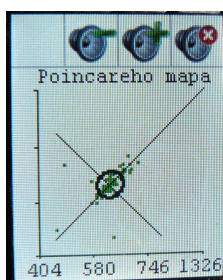
### Zpět na kartu

Nabídka Zpět na kartu navrátí uživatele zpět do Stará data, kde se daný soubor zavře a vykreslí se opět obsah paměťové karty.

### Poincarého mapa

V této nabídce se spustí funkce `enum MENU_code Poincare_Ini(void)`, kde se nejprve projede celý soubor řádek po řádku a zjistí se nejnižší a nejvyšší zaznamenaný RR impulz. Dále se projede znovu soubor řádek po řádku, vypočítá (pomocí funkce `void Vypocet(void)`) a zaznamená se maximální hodnota parametrů SD1 a SD2 a celkový součet parametrů SD1, SD2 a RR impulzů snížených o minimální RR impulz, z důvodu optimalzovaného nastavení os v grafu. Následně se přepočítá rozptyl hodnoty parametrů na 1 pixel a vypočítají se průměrné hodnoty parametrů k vykreslení elipsy. V dalším kroku se vykreslí osy, body RR impulzů, elipsa, přímky SD1 a SD2, a nakonec se popíše osy <sup>2</sup> (obr. 2.7).

Pak se cyklicky spouští funkce `enum MENU_code Poincare_Handler(void)`, ve které se volá funkce `void Jojstik(void)`, kde se při stisku joystickem doleva spustí funkce `void Zpet_karta (void)` a uživatel se navrátí zpět do nabídky souboru.



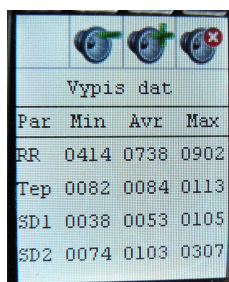
Obr. 2.7: Vykreslená Poincarého mapa

<sup>2</sup>Hodnoty u značek na ose y jsou rovny hodnotám u značek na ose x.

## Výpis dat

U výpisu dat se volá funkce `enum MENU_code Vypis_Ini(void)`, ve které se projede celý soubor řádek po řádku a pomocí volání funkce `void Vypocet(void)` se vypočítají a zaznamenají minimální, průměrné a maximální hodnoty parametrů SD1, SD2, Tepu a RR intervalu, které se následně vykreslí do tabulky (obr. 2.8).

Následně se cyklicky spouští funkce `enum MENU_code Vypis_Handler(void)`, ve které se volá funkce `void Jojstik(void)`, kde se při stisku joystickem doleva spustí funkce `void Zpet_karta(void)` a uživatel se navrátí zpět do nabídky souboru.



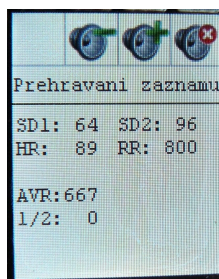
The screenshot shows a display with a title bar containing three joystick icons. Below the title bar, the text "Vypis dat" is displayed. A table follows with the following data:

Par	Min	Avr	Max
RR	0414	0738	0902
Tep	0082	0084	0113
SD1	0038	0053	0105
SD2	0074	0103	0307

Obr. 2.8: Výpis naměřených dat ze souboru

## Přehrání dat

Při výběru Přehrání dat se spustí funkce `enum MENU_code Prehrani_Ini(void)`, kde se nastaví rychlost volání aplikace, vynulují se proměnné a vymaže se obsah displaye. Pak se cyklicky volá funkce `enum MENU_code Prehrani_Handler(void)`, kde se čte soubor řádek po řádku a vypočítávají se parametry SD1, SD2, Tep, RR interval a průměrný RR interval, které se následně zobrazují na displayi (obr. 2.9).



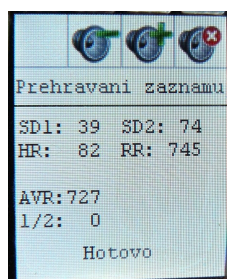
The screenshot shows a display with a title bar containing three joystick icons. Below the title bar, the text "Prehravani zaznamu" is displayed. The following data is shown:

SD1: 64	SD2: 96
HR: 89	RR: 800
AVR: 667	
1/2: 0	

Obr. 2.9: Přehrání dat ze souboru

Výpočet a zobrazení je v podmínce `if(z!=size)`, ve které se také nachází volání funkce `void Jojstik(void)`, k zastavení přehrávání a navrácení do nabídky souboru kdykoliv během přehrávání. Po projetí celého souboru se podmínka naplní a vykreslí se nápis "Hotovo" (obr. 2.10) značící ukončení přehrávání z důvodů konce

záznamu a následně je volána funkce `void Jojstik(void)` k možnosti ukončení a návratu do nabídky souboru.



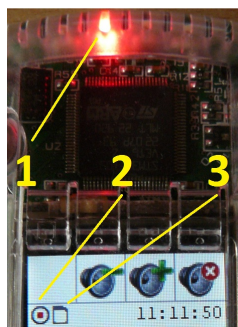
Obr. 2.10: Konec přehrávání dat ze souboru

### 2.9.4 Konec

Při výběru Konec se spustí funkce `void Konec(void)` a program se ukončí a uživatel je navrácen do CircleOS.

### 2.9.5 Popis indikace a ovládání

K ovládání slouží joystick, jeho funkce jsou popsány v předchozím popisu fungování jednotlivých sekcí programu. V případě Měření měřič indikuje tři stavy, viz. tab. 2.6



Obr. 2.11: Popis indikačních prvků

Prvek	Indikace	Popis
1	LED dioda	Bliknutím indikuje přijatý HR impulz z pásu
2	REC	Povolený zápis na kartu
3	Karta	Paměťová SD karta je vložena a správně detekována

Tab. 2.6: Popis indikačních prvků

## 2.10 Srovnání naměřených dat

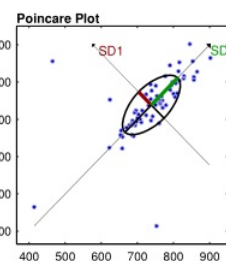
Pro zjištění správnosti měření je použit jak výpočet programu KubioHRV s daty z měřiče, tak i záznam ze specializovaného měřiče EKG s elektrodami. Na obrázku 2.12 je srovnání výřezů výsledků včetně Poincarého mapy. Celé výstupy z programu KubiosHRV jsou v příloze B a C.

### Data z mého měřiče v programu KubiosHRV Time-Domain Results

Variable	Units	Value
Mean RR*	(ms)	748.0
STD RR (SDNN)	(ms)	63.3
Mean HR*	(1/min)	80.78
STD HR	(1/min)	6.04

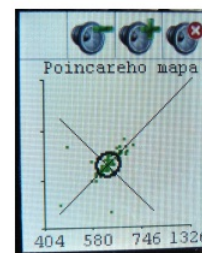
### Nonlinear Results

Variable	Units	Value
<b>Poincare plot</b>		
SD1	(ms)	49.8
SD2	(ms)	103.1



### Data zobrazená v měřiči

Par	Min	Avr	Max
RR	0414	0738	0902
Tep	0082	0084	0113
SD1	0038	0053	0105
SD2	0074	0103	0307



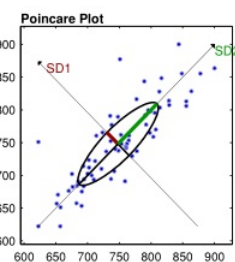
### Data z EKG měřiče v programu KubiosHRV Time-Domain Results

Variable	Units	Value
Mean RR*	(ms)	738.5
STD RR (SDNN)	(ms)	80.4
Mean HR*	(1/min)	82.45
STD HR	(1/min)	11.60

### Nonlinear Results

Variable	Units	Value
<b>Poincare plot</b>		
SD1	(ms)	24.8
SD2	(ms)	86.7

### Recurrence plot



Obr. 2.12: Výpis naměřených dat ze souboru



Pro přehlednost v tabulce 2.7 jsou data k porovnání. Jak vidíme naměřená data mezi sebou korelují, tudíž měřič funguje. Výrazný rozdíl v parametrech SD1 a SD2 u EKG měření je způsoben kvalitnějším měřením tepu. Celkový rozdíl hodnot je způsoben zaokrouhlováním při výpočtu. Poincarého mapy také mezi sebou korelují, ale srovnáním byla objevena chyba v popisu os, o které se zmiňuji více v závěru.

Měření	Prům. RR	HR	SD1	SD2
Kubios	748	80.78	49,8	103,1
Měřič	738	84	53	103
EKG	738,5	82,45	24,8	86,7

Tab. 2.7: Tabulka srovnání naměřených dat

### 3 ZÁVĚR

Cílem bakalářské práce bylo vyvinout zařízení, které bude přijímat signál z hrudního pásu sporttesteru a následně bude tento signál zpracovávat.

V úvodní části práce jsem popsal jednotlivé části EKG křivky, dále jsem popsal k čemu variabilita tepové frekvence slouží a jak se vypočítají její vybrané parametry. V další kapitole jsem se věnoval popisu vývojového kitu STM32 Primer 2 a softwaru KubiosHRV na zpracování naměřených dat. Dále jsem popisoval vytvořený program, jednotlivé jeho funkce z kterých je naprogramován. V předposlední kapitole jsem popsal práci s programem a jeho ovládání. Na závěr jsem shrnul výsledky měření.

Pro příjem HR signálu jsem použil čidlo RMCM01 firmy Polar, které bylo již ve vývojovém kitu STM32 Primer2. Pro vysílání jsem použil sporttesterový pás POLAR T31. Program jsem programoval, ve vývojovém prostředí Raisonance Ride7, jazykem ANSI C. Vývojový kit se programuje přes USB.

Naměřená data jsem použil jako vstupní data pro program KubiosHRV na PC, kde jsem ověřil správnost měření. Pro kontrolu jsem měl i naměřená data ze specializovaného EKG měřiče, které jsem také použil jako vstupní data pro program KubiosHRV.

Při ověřování správnosti měření jsem zjistil chybu v popisu os při vykreslování poincarého mapy na mém měřiči. Display je bohužel malý a proto je potřeba přepočítat rozptyl hodnoty na 1 pixel, aby se graf vešel na display a správně se vykreslil. V tomto přepočtu je ale chyba, protože se přepočítává zvláště RR interval a zvláště se nastavují hodnoty k osám. Tuto chybu by bylo možné odstranit lepší algoritmizací přepočtu rozptylu a v závislosti na ní nastavit hodnoty k osám.

Měřič by bylo možno rozšířit o možnost smazání souboru z paměťové karty, nastavení rychlosti přehrávání dat s možností pozastavit přehrávání a indikací doby přehrávání.

## LITERATURA

- [1] ACHARYA, U. R., et al. *Heart rate variability: a review. Medical and Biological Engineering and Computing* [online]. 2006, 44, [cit. 1. 11. 2013]. Dostupné z URL: <<http://knowledgetranslation.ca/sysrev/articles/project21/Ref%20ID%208067-20090628231006.pdf>>.
- [2] *CircleOS V4 Conception document* [online]. 2012, poslední aktualizace 25. 11. 2013 [cit. 21. 12. 2013]. Dostupné z URL: <[http://www.stm32circle.com/resources/download.php?CircleOS\\_Conception.pdf](http://www.stm32circle.com/resources/download.php?CircleOS_Conception.pdf)>.
- [3] HAMAN, P. *Výukový web EKG* [online]. [cit. 8. 11. 2013]. Dostupné z URL: <<http://ekg.kvalitne.cz/popis5.htm>>.
- [4] *Jak vypočítat maximální tepovou frekvenci* [online]. [cit. 1. 1. 2014]. Dostupné z URL: <<http://sporttester.info/2012/jak-vypocitat-maximalni-tepovou-frekvenci/>>.
- [5] KITLAS, A, et al. *Nonlinear dynamics methods in analysis of heart rate variability. Annales Academiae Medicae Bialostocensis* [online]. 2005, 50, [cit. 17. 11. 2013]. Dostupné z URL: <[http://www.advms.pl/roczniki\\_2005\\_supl\\_2/volumes/16\\_Kitlas.pdf](http://www.advms.pl/roczniki_2005_supl_2/volumes/16_Kitlas.pdf)>.
- [6] *KubiosHRV Users Guide* [online]. [cit. 26. 12. 2013] Dostupné z URL: <[http://kubios.uef.fi/media/Kubios\\_HRV\\_2.1\\_Users\\_Guide.pdf](http://kubios.uef.fi/media/Kubios_HRV_2.1_Users_Guide.pdf)>.
- [7] MAŠEK, O. *Heart Rate Variability Analysis* [online]. 2009, 66, [cit. 26. 12. 2013]. Dostupné z URL: <<https://cyber.felk.cvut.cz/research/theses/papers/49.pdf>>.
- [8] *Maximum Heart Rate* [online]. [cit. 1. 1. 2014]. Dostupné z URL: <<http://www.brianmac.co.uk/maxhr.htm>>.
- [9] *Polar Electro Europe BV RMCM-01 Heart Rate Receiver Component* [online]. poslední aktualizace 11. 11. 2004 [cit. 20. 12. 2013]. Dostupné z URL: <<https://www.sparkfun.com/datasheets/Wireless/General/RMCM01.pdf>>.
- [10] *Ride7 ARM Getting Started* [online]. 2012 [cit. 26. 12. 2013]. Dostupné z URL: <[http://support-raisonance.com/extranet/includes/download\\_doc.php?doc\\_id=12](http://support-raisonance.com/extranet/includes/download_doc.php?doc_id=12)>.
- [11] *STM32-Primer2 - User manual* [online]. poslední aktualizace 28. 5. 2010 [cit. 1. 12. 2012]. Dostupné po registraci z URL: <<http://www.stm32circle.com/resources/download.php?STM32-Primer2-Manual.pdf>>.

# SEZNAM SYMBOLŮ, VELIČIN A ZKRATEK

ADC Analog Digital Converter

API Application Programming Interface

ANSI C American National Standards Institute C

ASCII American Standard Code for Information Interchange

CPU Central Processing Unit

DAC Digital Analog Converter

DPS Deska Plošných Spojů

EKG ElektroKardioGram

GPIO General-Purpose Input/Output

GUI Graphic User Interface

HRV Heart Rate Variability

IrDA Infrared Data Association

LCD Liquid Crystal Display

LED Light-Emitting Diode

Li-Ion Lithium-Iont

MBR Master Boot Sector

OS Operační Systém

PWM Puls Width Modulation

RAM Random Access Memory

ROM Read Only Memory

SD Security Digital

SRAM Static Random Access Memory

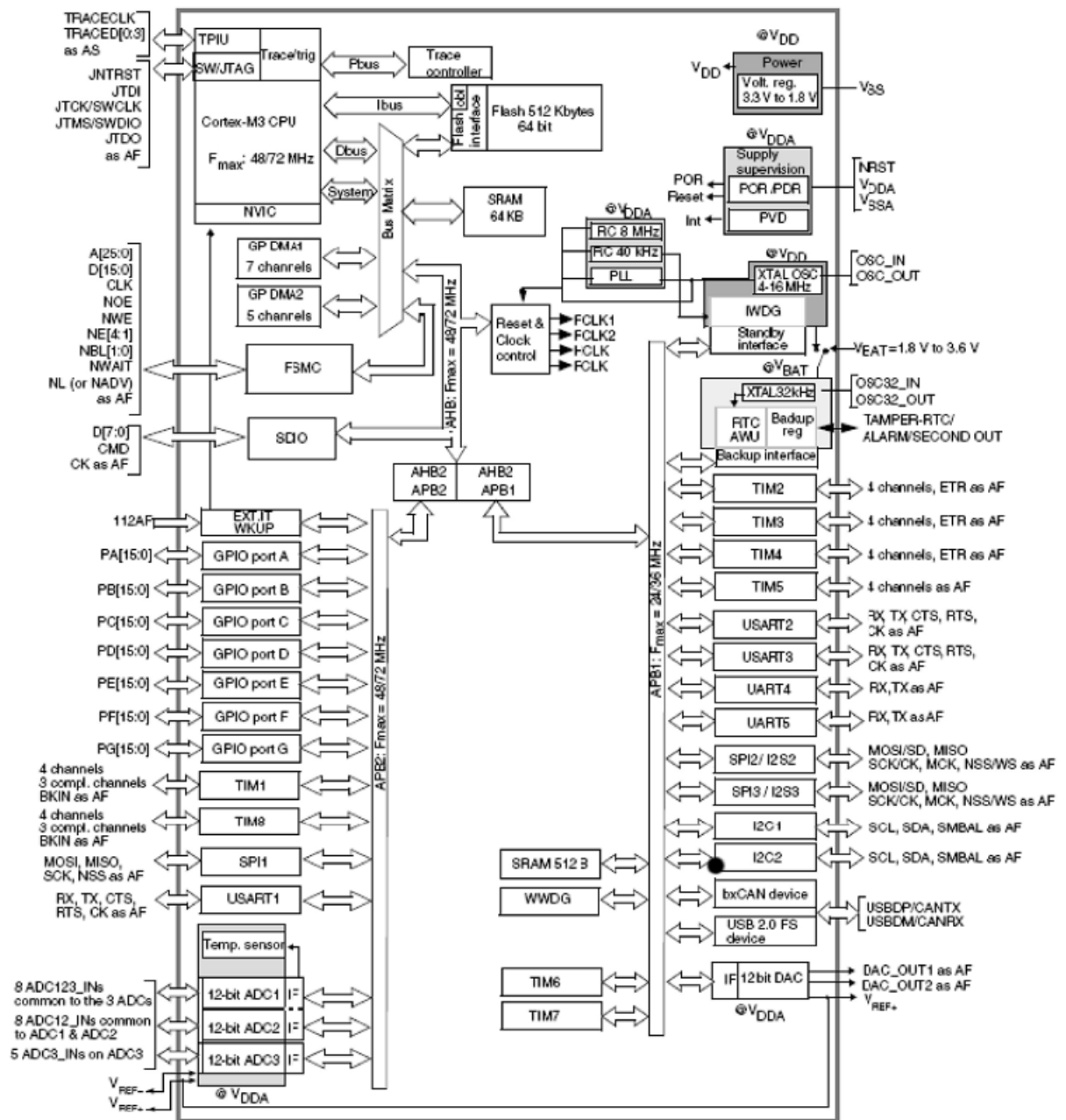
TFT Thin Film Transistor

USB Universal Serial Bus

## SEZNAM PŘÍLOH

A	Vnitřní schéma STM32 Primer2	46
B	Výstupní list z programu Kubios HRV pro můj měřič	47
C	Výstupní list z programu Kubios HRV pro EKG měřič	48
D	Obsah CD	49

# A VNITŘNÍ SCHÉMA STM32 PRIMER2



# B VÝSTUPNÍ LIST Z PROGRAMU KUBIOS HRV PRO MŮJ MĚŘIČ

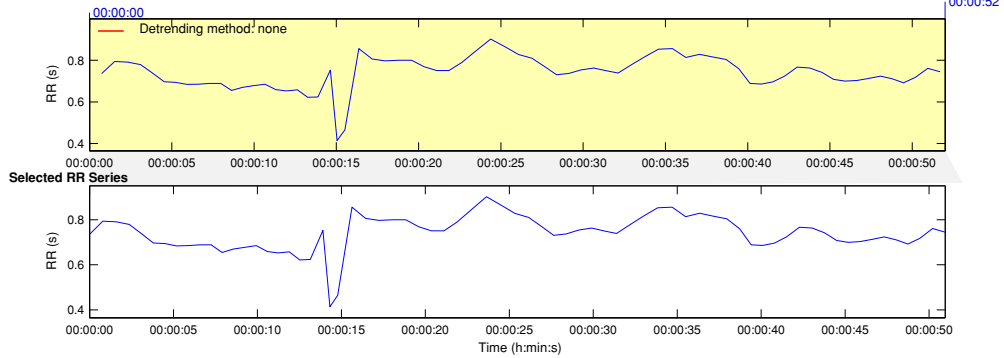
## HRV Analysis Results

R182409.TXT – xx/xx/xx – xx:xx:xx

Page 1/1

RR Interval Time Series

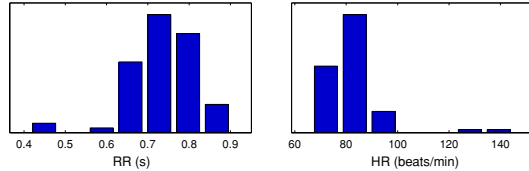
Results for a single sample



### Time-Domain Results

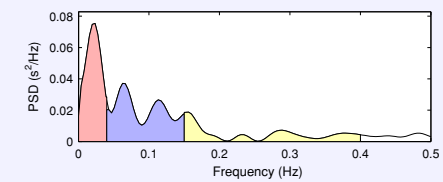
Variable	Units	Value
Mean RR*	(ms)	738.5
STD RR (SDNN)	(ms)	80.4
Mean HR*	(1/min)	82.45
STD HR	(1/min)	11.60
RMSSD	(ms)	70.0
NN50	(count)	8
pNNS0	(%)	11.6
RR triangular index		10.000
TINN	(ms)	425.0

Distributions\*



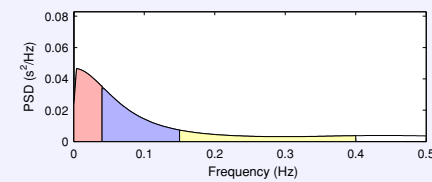
### Frequency-Domain Results

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



Frequency Band	Peak (Hz)	Power (ms <sup>2</sup> )	Power (%)	Power (n.u.)
VLF (0–0.04 Hz)	0.0234	2146	37.8	
LF (0.04–0.15 Hz)	0.0625	2337	41.2	66.2
HF (0.15–0.4 Hz)	0.1563	1184	20.9	33.5
Total		5676		
LF/HF		1.973		

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

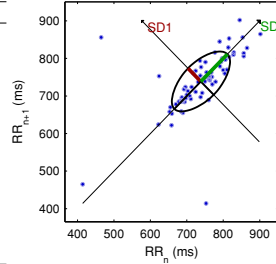


Frequency Band	Peak (Hz)	Power (ms <sup>2</sup> )	Power (%)	Power (n.u.)
VLF (0–0.04 Hz)	0.0039	1651	36.3	
LF (0.04–0.15 Hz)	0.0430	1921	42.3	66.4
HF (0.15–0.4 Hz)	0.1523	964	21.2	33.3
Total		4543		
LF/HF		1.991		

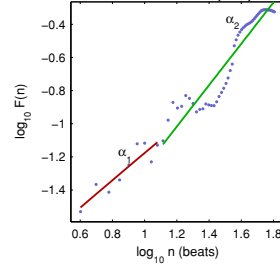
### Nonlinear Results

Variable	Units	Value
<b>Poincare plot</b>		
SD1	(ms)	49.8
SD2	(ms)	103.1
<b>Recurrence plot</b>		
Mean line length (Lmean)	(beats)	7.76
Max line length (Lmax)	(beats)	37
Recurrence rate (REC)	(%)	32.25
Determinism (DET)	(%)	97.00
Shannon Entropy (ShanEn)		2.589
<b>Other</b>		
Approximate entropy (ApEn)		0.553
Sample entropy (SampEn)		0.964
Detrended fluctuations (DFA): $\alpha_1$		0.825
Detrended fluctuations (DFA): $\alpha_2$		1.249
Correlation dimension (D2)		2.089

Poincare Plot



Detrended fluctuations (DFA)



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

# C VÝSTUPNÍ LIST Z PROGRAMU KUBIOS HRV PRO EKG MĚŘIČ

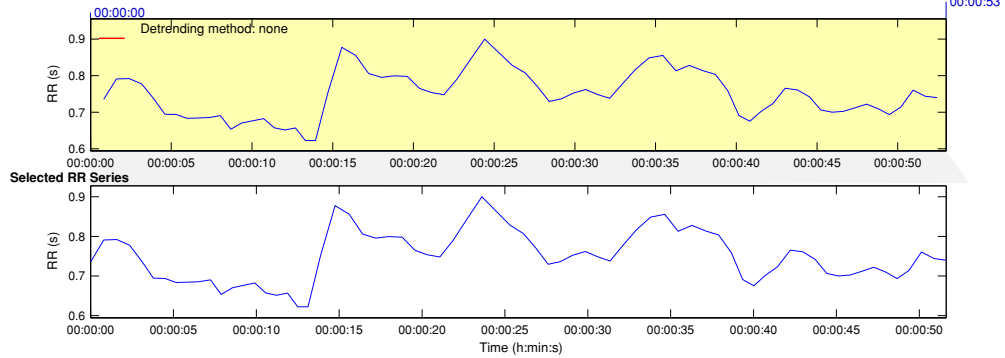
## HRV Analysis Results

data500Hz.txt - xx/xx/xx - xx:xx:xx

Page 1/1

RR Interval Time Series

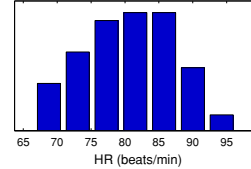
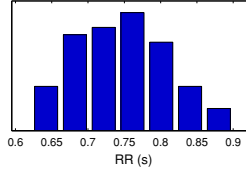
Results for a single sample



### Time-Domain Results

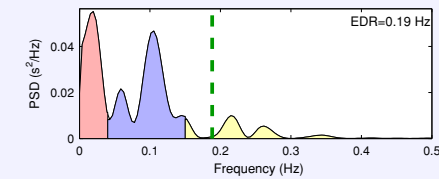
Variable	Units	Value
Mean RR*	(ms)	748.0
STD RR (SDNN)	(ms)	63.3
Mean HR*	(1/min)	80.78
STD HR	(1/min)	6.81
RMSSD	(ms)	34.7
NN50	(count)	6
pNN50	(%)	8.7
RR triangular index		14.000
TINN	(ms)	260.0

Distributions\*



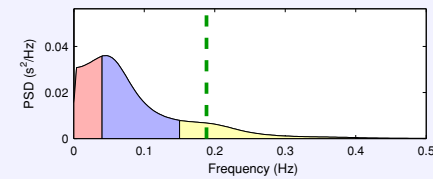
### Frequency-Domain Results

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



Frequency Band	Peak (Hz)	Power (ms <sup>2</sup> )	Power (%)	Power (n.u.)
VLF (0-0.04 Hz)	0.0195	1595	36.4	
LF (0.04-0.15 Hz)	0.1055	2222	50.8	79.9
HF (0.15-0.4 Hz)	0.2148	559	12.8	20.1
Total		4375		
LF/HF		3.978		

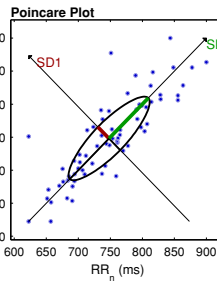
AR Spectrum (AR model order = 16, not factorized)



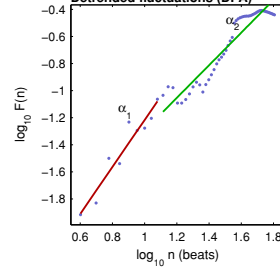
Frequency Band	Peak (Hz)	Power (ms <sup>2</sup> )	Power (%)	Power (n.u.)
VLF (0-0.04 Hz)	0.0391	1281	30.4	
LF (0.04-0.15 Hz)	0.0469	2208	52.3	75.1
HF (0.15-0.4 Hz)	0.1523	729	17.3	24.8
Total		4219		
LF/HF		3.027		

### Nonlinear Results

Variable	Units	Value
<b>Poincare plot</b>		
SD1	(ms)	24.8
SD2	(ms)	86.7
<b>Recurrence plot</b>		
Mean line length (Lmean)	(beats)	7.26
Max line length (Lmax)	(beats)	60
Recurrence rate (REC)	(%)	24.78
Determinism (DET)	(%)	97.61
Shannon Entropy (ShanEn)		2.445
<b>Other</b>		
Approximate entropy (ApEn)		0.513
Sample entropy (SampEn)		1.055
Detrended fluctuations (DFA): $\alpha_1$		1.737
Detrended fluctuations (DFA): $\alpha_2$		1.204
Correlation dimension (D2)		2.238



Detrended fluctuations (DFA)



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



## D OBSAH CD

- ./Image2primer - Složka obsahuje program Image2prime.
- ./Kubios HRV - Složka obsahuje instalační soubory programu KubiosHRV.
- ./Program - Složka obsahuje zdrojové kódy programu.
- ./Ride7 - Složka obsahuje instalační soubory vývojového prostředí.
- ./ARM - Složka obsahuje instalační soubory debuggeru k mikroprocesoru vývojového kitu.