

Západočeská univerzita v Plzni  
Fakulta aplikovaných věd  
Katedra informatiky a výpočetní techniky

## **Bakalářská práce**

# **System pro bezdrátové měření tepové frekvence**

# Prohlášení

Prohlašuji, že jsem bakalářskou práci vypracoval samostatně a výhradně s použitím citovaných pramenů.

V Plzni dne 11. května 2012

Ondřej Pillár

# Poděkování

Na tomto místě bych rád poděkoval vedoucímu práce Ing. Jiřímu Ledvinovi, CSc. za metodické vedení, Ing. Karlu Dudáčkovi, Ing. Karlu Novotnému a v neposlední řadě své rodině za vytvoření vhodných podmínek.

# **Abstract**

## **Hearth rate monitoring system**

This bachelor thesis deal with the development of battery system for heart rate gauging. In this work are stated and described a few basic gauging methods. Next there is described wireless net ZigBEE and evolutional kit eZ430-RF2480. Practical part deal with plan of functional gauging prototype.

## **Keywords**

ZigBee, MSP430, CC2480, heart rate, ez430-RF2480, oxymetr, measurement, wireless network, Texas Instruments, ZASA, ZACCEL

# Obsah

<b>1</b>	<b>Úvod</b>	<b>1</b>
<b>2</b>	<b>Měření tepové frekvence</b>	<b>2</b>
2.1	Fotoelektrická metoda . . . . .	2
2.2	Impedanční metoda . . . . .	4
2.3	Akustická metoda . . . . .	5
<b>3</b>	<b>ZigBee</b>	<b>6</b>
3.1	OSI model ZigBee . . . . .	6
3.2	Topologie sítě . . . . .	8
<b>4</b>	<b>Popis hardwarového vybavení</b>	<b>10</b>
4.1	Vývojový kit eZ430-RF2480 . . . . .	10
4.2	Procesor MSP430F2274 . . . . .	12
4.3	Procesor CC2480 . . . . .	13
4.3.1	Komunikace mezi procesory . . . . .	14
4.3.2	Aplikační rozhraní . . . . .	17
4.3.3	Simple API . . . . .	17
4.4	Snímač tepové frekvence PulseSensor . . . . .	19
<b>5</b>	<b>Realizace bezdrátového měření tepové frekvence</b>	<b>20</b>
5.1	Blokové schéma měřícího systému . . . . .	20
5.2	Přenos zpráv . . . . .	21
5.2.1	Aplikační protokol . . . . .	21
5.3	Aplikace . . . . .	22
5.3.1	Koncové zařízení . . . . .	23
5.3.2	Koordinátor . . . . .	30
<b>6</b>	<b>Programové vybavení pro vizualizaci naměřených dat</b>	<b>32</b>
<b>7</b>	<b>Porovnání finálního produktu s komerčním prostředkem</b>	<b>33</b>

---

<b>8 Závěr</b>	<b>35</b>
<b>A Přílohy</b>	<b>41</b>
A.1 Uživatelská příručka programu WorkOut . . . . .	41
A.2 Popis tříd programu WorkOut . . . . .	42

# 1 Úvod

Bakalářská práce se zabývá návrhem systému pro bezdrátové měření tepové frekvence. K tomu je potřeba analyzovat dostupné bezdrátové technologie, zejména pak komunikační technologii ZigBee, a její realizaci firmou Texas Instruments, které patřila a stále patří k průkopníkům při vývoji toho standardu. K tomu účelu je k dispozici vývojový kit eZ430-RF2480 s nímž je potřeba se detailně seznámit. Nedílnou součástí je samozřejmě zorientovat se v metodách měření tepové frekvence. V praktické části pak popsat proces vývoje celého měřicího systému a v závěru jej porovnat s komerčními prostředky.

Tep a tím pádem i tepová frekvence patří mezi základní pozorovatelné tělesné signály, které se dají na lidském těle sledovat. Tato práce je v mnohých ohledech koncipována jako první krok při snaze o návrh systému, který by v budoucnu mohl sloužit jako měřicí systém pro sledování tepové frekvence při sportovních aktivitách, zejména pak fitness sportech jako je např. Spinning, Schwin, Heat. K tomu je nutné se teoreticky a samozřejmě i prakticky seznámit se všemi výše jmenovanými částmi, navrhnout jistý prototyp systému, který bude možné v následujících projektech dále vyvíjet.

## 2 Měření tepové frekvence

Tepová frekvence je jedna ze základních informací, podle kterých můžeme sledovat činnost srdce a často slouží jako ukazatel při posuzování stavu organismu v klidu i při zátěži. Tepová frekvence typicky udává počet srdečních tepů za určitou jednotku času. Změnami tepové frekvence lze snadno sledovat i případné výjimečné stavy srdečního tepu, jako je například arytmie – nepravidelnost v činnosti srdce, abnormální bradykardii – snížení tepové frekvence pod určitou hladinu, či abnormální tachykardii – zvýšení tepové frekvence nad určitou hladinu. Zvláště při sledování stavu organismu při sportovních aktivitách, udává průběžné sledování tepové frekvence cenné informace o chování sportovcova organismu [Šeda(2010)].

Pro stanovení tepové frekvence musíme jednoznačně detekovat činnost srdce, tu lze získat měřením projevů, které vznikají při jeho činnosti. Mezi tyto veličiny patří:

- elektrický signál
- akustický signál
- změny tlaku krve v oběhovém systému
- změny impedance tkáně související se změnou množství krve v daném úseku tkáně
- změny rychlosti proudění krve

Pro měření tepové frekvence při sportovních aktivitách není kladen velký důraz na přesnou znalost okamžité hodnoty. Taková hodnota by byla velmi kolísavá a údaje by byly špatně čitelné. Je proto lepší sledovat tzv. klouzavý průměr – průměr naměřených hodnot za posledních několik tepů. V následujících řádcích budou proto rozebrány metody měření z pohledu využitelnosti v praxi, náročnosti realizace a použitelnost přesnosti měření.

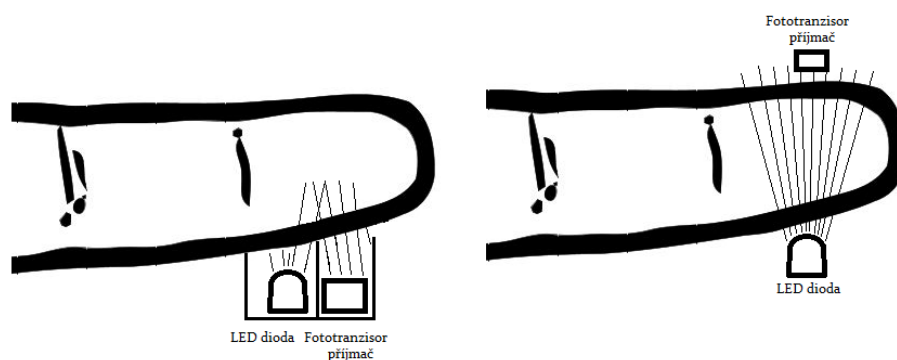
### 2.1 Fotoelektrická metoda

Fotoelektrická pletysmografie registruje změny světelné propustnosti tkáně díky změnám tlaku krve při srdeční činnosti (viz obrázek2.1). Využívají se



dvě metody:

- průsvitové
- reflexivní



Obrázek 2.1: Reflexivní a průsvitová fotoelektrická metoda

Uvedená metoda se využívá ke sledování srdeční frekvence také vyhodnocování objemových změn. Fotoelektrický snímač nesmí být citlivý na nasycení krve kyslíkem, které způsobuje změnu zabarvení krve. Systém je velmi často používán v komerčních aplikacích při orientačním měření v tzn. Oxymetrech (obrázek 2.2).

Konstrukce snímače je velmi jednoduchá a naměřená data se dají zrekonstruovat bez nutnosti hlubší lékařské teorie. Proto bude této metody použito i ve výsledné realizaci měřícího systému. Princip měřícího zařízení je detailně rozebrán v kapitole 4.4. Mezi nedostatky patří velmi velká náchylnost na rušení signálu. Tkáň by měla být pokud možno ve stálém dotyku se senzorem, neměla by se moc pohybovat. Oxymetry k tomuto účelu používají jakési skřipce, které vymezují polohu a zajišťují konstantní měřící prostředí.



Obrázek 2.2: Komerční oxymetr firmy Masimo [Mas(2011)]

## 2.2 Impedanční metoda

Impedanční kardiografie (IKG) využívá měření změn hrudní impedance v průběhu srdečního cyklu. Impedanční senzory jsou umístěny na hrudníku. Celková impedance hrudníku je nepřímo úměrná obsahu tekutiny. Změny impedance tkáně lze dobře zjistit například zapojením měřáku do můstku a měřit jeho vyvážení [Fak(2010)].

Na tomto principu funguje i hrudní pás z vodivé gumy, který je využíván v komerčních produktech tzv. SportTestech. Princip spočívá v měření proměnlivé impedance tkáně. Při měřeních, které byly prováděny na pásu se ale metoda ukázala jako značně komplikovaná a náročná na filtraci naměřených dat, což by přesáhlo rozsah této bakalářské práce. Jako další komplikace se ukázal nedostatek informací a podkladů k této metodě, protože výrobci hrudních pásů tyto materiály samozřejmě nikde nezveřejňují.

## 2.3 Akustická metoda

V podstatě nejjednodušší a nejstarší z možných měřících metod, spolu se změnou tlaku v soustavě. Z trupu pacienta se mikrofonem snímá signál a převádí jej na signál elektrický. Jakkoli je tento princip jednoduchý, je velmi citlivý na parazitní šum a následné získávání dat z takového signálu může být poměrně složité [Šeda(2010)].

Mezi další metody pro měření tepu patří:

- Detektor R vlny
- Metoda objemové změny tkáně

## 3 ZigBee

ZigBee je bezdrátová komunikační technologie, standardizovaná od roku 2004, která je vystavěná na normě IEEE 802.15.4. Je určena zejména pro spojení nízkovýkonových zařízení na malé vzdálenosti (cca 70 metrů). Primární nasazení je tudíž v průmyslových aplikacích a v senzorických sítích.

Tato technologie patří do skupiny bezdrátových sítí PAN (Personal Area Networks), podobně jako další velmi rozšířený standard Bluetooth. Ten má uplatnění zejména ve spotřební elektronice, ale v průmyslu je celá řada aplikací, ve kterých není jeho nasazení vhodné. Z těchto důvodů byla roku 2002 založena ZigBee Alliance více než 60 firem např. Texas Instruments, Honeywell, Motorola, Philips, Samsung, Omron atd. které se podílely a stále podílí na vývoji a rozvoji tohoto standardu [J. Koton(2006)]. Mezi základní vlastnosti patří:

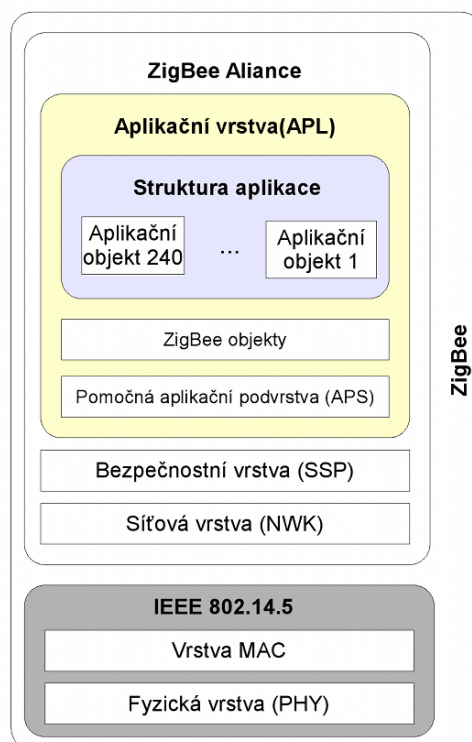
- spolehlivost
- příznivá cena
- velmi malá spotřeba energie

### 3.1 OSI model ZigBee

Jak již bylo zmíněno v úvodním odstavci, je technologie založena na IEEE normě 802.15.4. Konkrétně to znamená, že ZigBee přejímá první dvě vrstvy (Fyzickou a MAC). Referenční OSI model popisuje obrázek 3.1, ze kterého je patrné, že ZigBee v podstatě rozšiřuje normu IEEE 802.15.4 o další tři vrstvy. Vrstvu síťovou (NTW), bezpečnostní (SSP) a vrstvu aplikační (APL). Z obrázku je dále patrné, že APL se skládá z pomocné aplikační podvrstvy, ZigBee objektů a aplikačních objektů.

Pomocná aplikační podvrstva je zodpovědná za párování zařízení podle poskytovaných služeb a požadavků. To je realizováno pomocí tzv. párovací (binding) tabulky. ZigBee objekt definuje roli jednotlivých zařízení v rámci sítě (koordinátor, směrovač, koncové zařízení). Dále zajišťuje vyhledávání nových zařízení a jimi poskytovaných služeb. V neposlední řadě zodpovídá za zabezpečení (volí jeho způsob, jako např. veřejné klíče, symetrické klíče).

Uživatelské aplikační objekty implementují konkrétní požadavky aplikace dle definovaného ZigBee profilu. ZigBee profil zastrešuje definice možných zařízení, formátů a typů zpráv. Každý profil je určen unikátním 16 bitovým identifikátorem podle specifikace ZigBee Alliance [zig()]. Dohromady jsou tyto vrstvy nazývány ZigBee Stack (česky ZigBee Zásobník).



Obrázek 3.1: Referenční model ZigBee

Frekvenční pásma, ty jsou popsány v normě IEEE 802.15.4 a ZigBee je nijak nemění, stejně tak jako princip adresování v síti. To je prováděno buď nezkrácenou 64 bitovou adresou, nebo zkrácenou 16 bitovou adresou. Norma také obecně dělí typy zařízení mezi tzv ZPF (**Z**ařízení s **P**lnou **F**unkčností) a ZOF (**Z**ařízení s **O**mezenou **F**unkčností). ZPF je typický kořenový prvek sítě, který musí být stále aktivní. Naproti tomu ZOF je koncový prvek sítě.

Norma dále definuje 3 prvky topologie sítě (Koordinátor, PAN Koordinátor, Device). ZigBee tuto koncepci upravuje. Typy zařízení jsou detailně popsány v následující kapitole.

## 3.2 Topologie sítě

ZigBee umožňuje sestavit 3 typy sítí (viz obrázek 3.2), než budou blíže popsány je potřeba ještě doplnit princip adresování a přehled prvků, které mohou tvořit ZigBee síť.

Každá nově vytvořená síť je jednoznačně identifikována 16 bitovým PAN ID pro případ že by v jednom místě bylo současně provozováno více ZigBee sítí (obecně více sítí standardu IEEE 802.15.4). Nezkrácená adresa je vždy pevně určena výrobcem zařízení, zkrácená adresa může být zařízení přidělována koordinátorem sítě s tou výjimkou, že koordinátor má vždy adresu 0.

Prvky ZigBee sítí jsou:

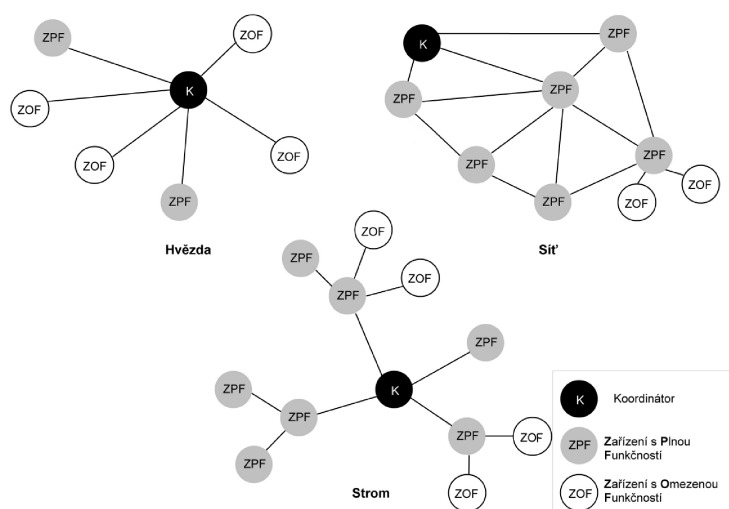
- **Koordinátor** (Coordinator)- Nejdůležitější člen každé sítě. Koordinátor tvoří kořen každé sítě a může fungovat jako most do další sítě. V každé jedné síti může být maximálně jeden koordinátor a je to právě ta jednotka, která založila síť. Koordinátor přiděluje nové síti PAN ID a určuje komunikační kanál. Po vytvoření sítě a připojení dalších jednotek by již mohl být koordinátor vyrazen. Z popisu je tedy zřejmé, že zařízení musí být typu ZPF (viz odstavec 3.1).
- **Směrovač** (Router)- Tento člen může ve značné míře přebírat funkcionality koordinátoru a stávat se tedy jakýmsi rodičem pro další jednotky, které se k němu připojí. I směrovač musí být typu ZPF.
- **Koncové zařízení** (End Device)- Koncový list celé topologie. Koncové zařízení je nejlevnější a nejjednodušší ze všech uvedených prvků. Zařízení je určeno pouze k přijetí resp. odeslání dat. Mezi těmito stavy může být zařízení ve stavu nízké spotřeby a tím se výrazně prodlužuje délka provozu zařízení na jedno nabití.

Nyní již k popisu jednotlivých typů topologií (viz obrázek 3.2):

- **Hvězda** - Koordinátor je centrálním prvkem sítě a přes něj komunikují všechny koncové jednotky. V tuto chvíli se koordinátor nepostradatelnou součástí sítě.
- **Strom** - Tato možnost využívá k práci směrovač. Ten přebírá v funkci centrálního prvku ke kterému jsou připojeny koncové jednotky, ty tedy

komunikují s koordinátorem nepřímo. To umožňuje větší rozsah celé sítě. Koordinátor stále zůstává centrálním prvkem celé soustavy přes který jde veškerá komunikace.

- **Sít'** - Nejsložitější možnost. Kombinuje oba dva výše zmíněné typy a umožňuje komunikaci mezi dvěma rovnocennými prvky.



Obrázek 3.2: Topologie sítě ZigBee

## 4 Popis hardwarového vybavení

### 4.1 Vývojový kit eZ430-RF2480

eZ430-RF2480 je demonstrační kit firmy Texas Instruments. Celý přípravek poskytuje všechny nezbytné hardwarové a softwarové součásti pro vývoj ZigBee sítě.

Vývojový kit obsahuje následující hardwarové části (viz obrázek4.2) :

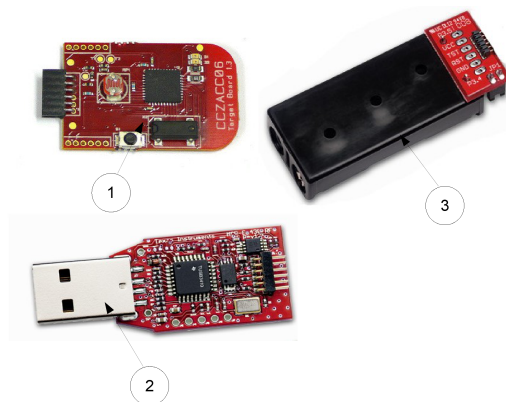
- 3x eZ430-RF2480 koncové zařízení (1)
- 1x MSP430 USB FET programátor (2)
- 2x AAA eZ430 bateriový adaptér (3)

Koncové zařízení obsahuje 2 procesory, jimž jsou věnovány samostatné kapitoly, snímač okolního osvětlení, SMD anténu, červenou a zelenou LED diodu, tlačítko a periferní obvody. Dále je k dispozici 2x6 pinů po stranách desky, na které jsou vyvedeny některé významné piny z uživatelského procesoru (viz obrázek 4.1) a 6 pinů pro připojení USB programátoru. Zejména oněch 12 pinů, sdružených po 6 do dvou portů je z pohledu uživatele zajímavých. Např. se na nich dá sledovat komunikace mezi procesory (piny P1.6, P2.6), jsou k dispozici kanály AD převodníku (P2.4, P3.1, P3.2, P3.3, P3.4, P3.5), atd. [Tex()].



PORT	PIN	POPIS
2	1	GND
	2	VDD_EXT
	3	P4.2 / TB2
	4	P4.3 / TB0 / A12 / OA00
	5	RESET_N
	6	SRDY
3	1	P4.4 / TB1 / A13 / OA10
	2	P4.5 / TB2 / A14 / OA13
	3	P2.2 / TA0 / A2 / OA01
	4	P2.3 / TA1 / A3 / VREF- / VeREF- / OA11 / OA1O
	5	P2.4 / TA2 / A4 / VREF+ / OA10
	6	MRDY

Obrázek 4.1: Rozložení vývodů procesoru na desku



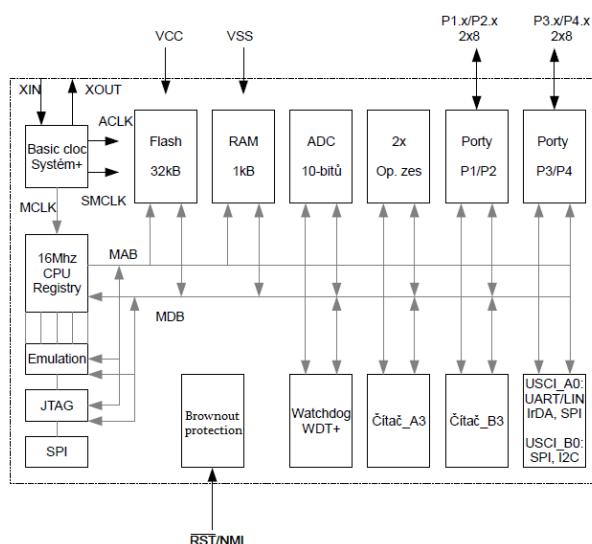
Obrázek 4.2: Vývojový kit eZ430-RF2480

## 4.2 Procesor MSP430F2274

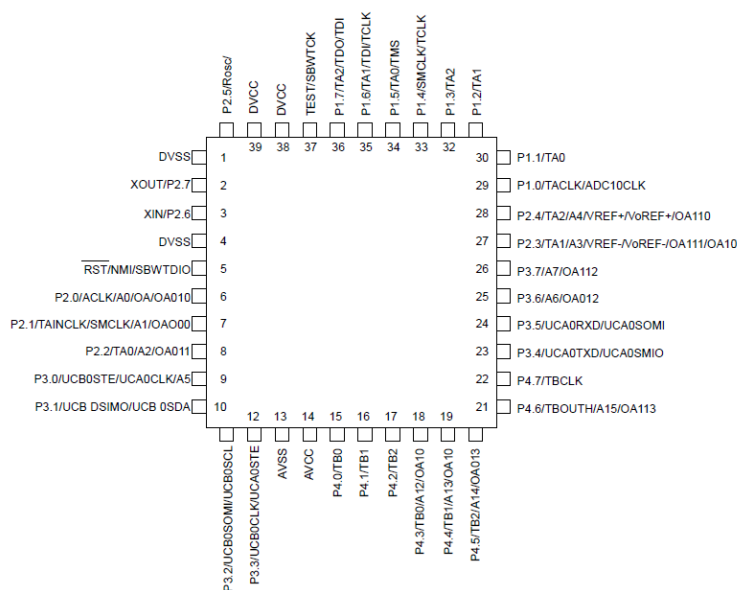
Procesor MSP430F2274 od firmy Texas Instruments patří do rodiny procesorů MSP430, což jsou ultra-nízko napěťové mikroprocesory určené pro širokou škálu využití. Jedná se o 16 bitový RISC procesor založený na von Neumannově architektuře, se sdíleným místem pro instrukce i data, čas jednoho instrukčního cyklu je 62,5ns. Kódování slov je systémem little-endian [Tex(2011b)]. Mezi základní vlastnosti procesoru patří:

- 6 pracovních režimů - 1 aktivní - odběr  $270\mu\text{A}$ , 5 režimů nízké spotřeby - standby  $0.7\mu\text{A}$ , vypnutý  $0.1\mu\text{A}$
- 32KB Flash
- 1KB RAM
- 10 bitový, 200K vzorků/s, AD převodník
- 2 operační zesilovače
- 2 16 bitové čítače/časovače, "Watchdog" časovač

Všechny komponenty procesoru ukazuje obrázek 4.3



Obrázek 4.3: Popis komponent procesoru MSP430



Obrázek 4.4: Rozložení pinů procesoru MSP430

### 4.3 Procesor CC2480

CC2480, někdy také označován jako CCZACC06, je první síťový procesor od firmy Texas Instruments implementující technologii ZigBee 2006 a vývojový model Z-Accel. Z-Accel je koncept, který značně usnadňuje práci při návrhu a vývoji nových bezdrátových produktů, jakožto i zakomponování bezdrátových rozhraní do již existujících produktů. Z-Accel jednotky, jako je procesor CC2480, kombinují všechny nezbytné části pro vytvoření bezdrátové sítě v jednom pouzdře. Uživatelský procesor potom komunikuje s jednotkou Z-Accel pomocí rozhraní SPI (viz obrázek 4.5) a ke komunikaci využívá předem definovanou API. Dá se proto říct, že Z-Accel přesně a čistě odděluje vlastní systém a bezdrátové komponenty [Tex(2006)].

Mezi nejdůležitější vlastnosti procesoru patří:

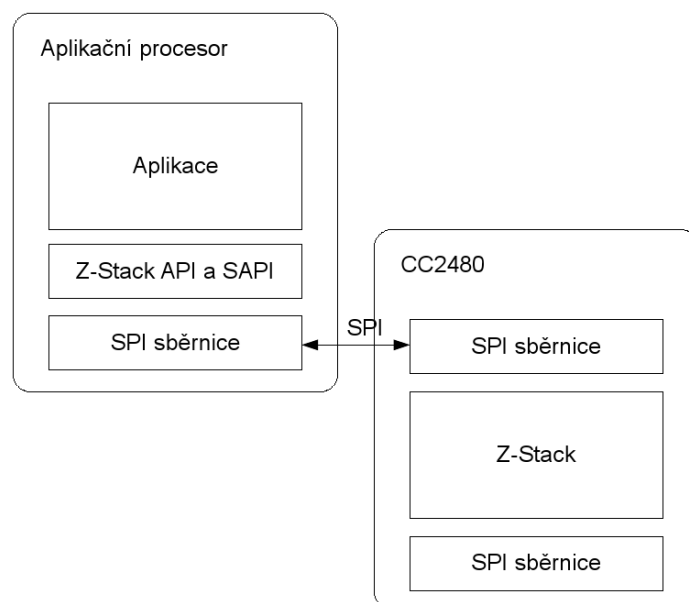
- Napájecí napětí 2.0V - 3.6V
- Malý proudový odběr RX: 27mA, TX: 27mA
- 4 operační módy

- Měřák napětí baterie, teplotní senzor
- 7-12 bitový, 2 kanálový AD převodník

Co se týče vlastností procesoru, není již třeba ho dále představovat, ty nejdůležitější byly zmíněny. Pro uživatele tohoto čipu to ani není nijak nutné. Je totiž mnohem důležitější znalost komunikačního rozhraní (API), kterým se procesor ovládá.

### 4.3.1 Komunikace mezi procesory

Jak již bylo zmíněno výše, základní filozofií celého konceptu Z-Accel spočívá ve striktní oddělení bezdrátového a aplikačního procesoru. Tyto procesory spolu komunikují po společné sběrnici SPI. Seriová linka nabízí vysokou přenosovou rychlost, teoreticky až 70 MHz a je určena k přenosům na kratší vzdálenost stylem Master - Slave [SPI()]. Aplikační procesor (MSP430F2274) komunikuje v roli master, síťový procesor (CC2480) je slave. Pro vzájemnou



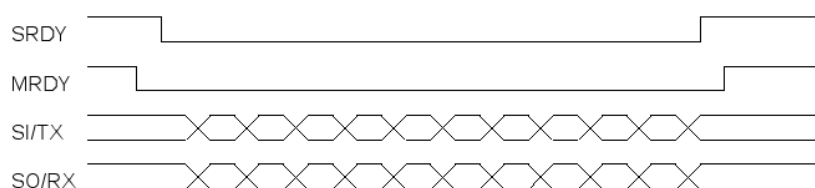
Obrázek 4.5: Koncept Z-Accel

komunikaci využívají procesory následující signály:

- **MDRY** - Master ready. Generuje master ve chvíli, kdy má k dispozici data pro CC2480. Je aktivní v nízké úrovni.
- **SRDY** - Slave ready. Signál je nastaven zařízením slave ve chvíli, kdy je připraveno ke komunikaci, tzn. buď odesílat, nebo přijímat data.
- **SI/TX** - Slave input / Master output data.
- **SO/RX** - Master output / Slave input data.
- **RESET** - Slave reset. Resetuje síťový procesor. Aktivní v nízké úrovni.

Při vzájemné komunikaci existují tři typické scénáře, které popisují obrázky 4.6, 4.8, 4.7

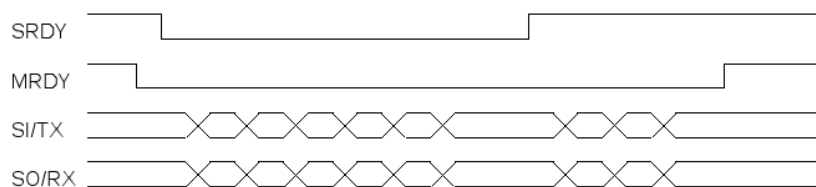
- **AREQ** - Aplikační procesor odesílá příkaz síťovému procesoru (viz obrázek 4.6).
- **SREQ** - Velmi podobná situace s tím rozdílem, že po odeslání dat čeká aplikační procesor na obdržení SRSP rámce. (viz obrázek 4.7).
- **POLL** - Síťový procesor žádá o zahájení komunikace (viz obrázek 4.8).



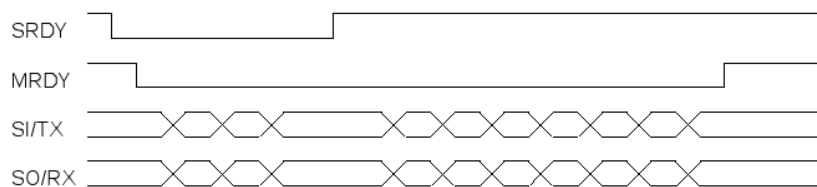
Obrázek 4.6: Příkaz AREQ

*Pozn.: Detailní popis komunikace je k dispozici v [Tex(2006)] str 15*

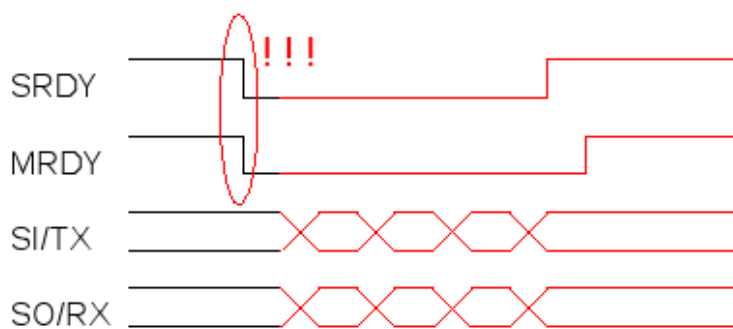
Při bližším prozkoumání celé situace ovšem vyvstává poměrně důležitý moment. V jednu chvíli by o komunikaci mohly požádat obě dvě jednotky (master i slave). Situaci na lince názorně popisuje obrázek 4.9. Podle pravidel



Obrázek 4.7: Příkaz SREQ



Obrázek 4.8: Příkaz POLL



Obrázek 4.9: Konflikt při komunikaci

komunikace nastaví zařízení, které chce komunikovat, svůj řídicí signál do nízké úrovně a očekává reakci od druhého procesoru. Pokud by tedy například master chtěl odeslat příkaz AREQ, snížil by MRDY do 0 a čekal by na nízkou úroveň signálu SRDY. V tu samou chvíli by ovšem chtěl komunikovat i síťový procesor. Analogicky by tedy shodil SRDY a očekával by signál MRDY v 0. A nyní nastává zmiňovaný problém. Obě dvě zařízení mylně interpretují tuto situaci jako stav, ve kterém mohou zahájit komunikaci myslí si, že opačné zařízení je připraveno přijímat data. Na tuto situaci ovšem síťový procesor CC2480 není připraven a přestává reagovat na další příkazy. Jediná možnost je síťový procesor CC2480 resetovat signálem RESET. Tento jeho stav není nikde dokumentován a může působit značné komplikace.

### 4.3.2 Aplikační rozhraní

Příkazy, které nabízí síťový procesor lze obecně rozdělit do čtyř následujících typů rozhraní [Tex(2006)]:

- **Rozhraní SYS** - jedná se o velmi "nízko úrovněový přístup" k hardwaru a softwaru procesoru CC2480. Umožňuje přístup např k AD převodníku, IO pinům, a dalším komponentám.
- **Konfigurační rozhraní** - pro konfiguraci různých proměnných procesoru CC2480.
- **AF, ZDO rozhraní** - nejrobustnější rozhraní pro zpřístupnění všech funkcí síťového procesoru a pro vytvoření jakékoli ZigBee sítě.
- **Simple API** - zjednodušené rozhraní pro základní jednoduchou konfiguraci ZigBee sítí, jednotek a přenos zpráv.

Ačkoli rozhraní AF a ZDO nabízí zdaleka nejširší přístup k síťovému procesoru a ZigBee síti, velká většina (je udáváno zhruba 85%) všech aplikací si pohodlně vystačí pouze se zjednodušeným Simple API rozhraním. Je proto dobré rozebrat příkazy, které toto rozhraní nabízí podrobněji.

### 4.3.3 Simple API

Rozhraní obsahuje 10 příkazů z nichž 1 slouží pro reset CC2480, 2 pro jeho konfiguraci a zbylých 7 je určeno přímo ke správě ZigBee sítě :

- **ZB\_SYS\_RESET\_REQ** - příkaz pro resetování síťového procesoru. Je vhodné procesor resetovat po některých konfiguracích.
- **ZB\_APP\_REGISTER\_REQUEST** - určuje parametry se kterými se bude tato jednotka připojovat do sítě.
- **ZB\_START\_REQUEST** - startuje činnost ZigBee Zásobníku v síťovém procesoru. CC2480 načte konfigurační hodnoty z paměti. Po ukončení tohoto procesu je zařízení schopno posílat, přijímat a směřovat provoz v síti (záleží samozřejmě na konfiguraci).
- **ZB\_PERMIT\_JOINING\_REQUEST** - slouží k určování pravidel připojování se nových zařízení do již existující sítě.
- **ZB\_BIND\_DEVICE** - umožňuje zaslat zprávu i do zařízení, jehož adresa není známá (resp známe pouze jeho MAC adresu). Pokud není ani MAC adresa známá, je možno využít také možnosti tzv "bindování". K tomu ale cílové zařízení musí být nakonfigurováno (viz následující bod).
- **ZB\_ALLOW\_BIND** - zadáním toho příkazu je zařízení nastaveno do režimu "bindování" což umožňuje, že mu jsou zasílány i neadresované zprávy.
- **ZB\_SEND\_DATA\_REQUEST** - Příkaz zahájí přenos dat do jiného zařízení. Měl by být použit pouze ve chvíli, kdy již proběhla registrace aplikačního procesoru (ZB\_APP\_REQUEST) a zařízení úspěšně vytvořilo nebo se připojilo do sítě.
- **ZB\_GET\_DEVICE\_INFO** - na základě předaného parametru vrací informaci o zařízení (viz 4.3.3).

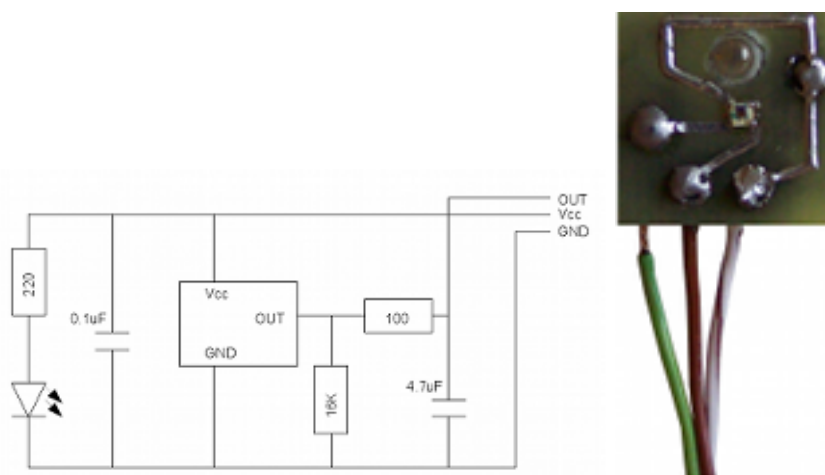
Parametr	Typ příkazu
0	stav zařízení
1	MAC adresa zařízení
2	síťová adresa zařízení
3	síťová adresa rodičovského zařízení
4	MAC adresa rodičovského zařízení
5	kanál sítě
6	PAN ID sítě
7	rozšířené PAN ID sítě



- **ZB\_FIND\_DEVICE\_REQUEST** - Získání síťové adresy libovolného zařízení v síti za předpokladu znalosti MAC adresy.
- **ZB\_WRITE\_CONFIGURATION, ZB\_READ\_CONFIGURATION** - zápis / čtení řídicích registrů síťového procesoru.

## 4.4 Snímač tepové frekvence PulseSensor

Projekt PulseSensor je volně dostupný snímač tepové frekvence za jehož vývojem stojí Joel Murphy a Yury Gitman. Jedná se o snímač založený na principu fotoelektrické metody (viz 2.1). Jako snímač prosvětlení je použit fotočlen APDS-9008. Celý přípravek je možné zakoupit přímo u výrobce [Joel Murphy()]. Pro účel této práce byl přípravek vyroben na Západočeské univerzitě. Schéma zapojení a výslednou realizaci ukazuje obrázek 4.4.



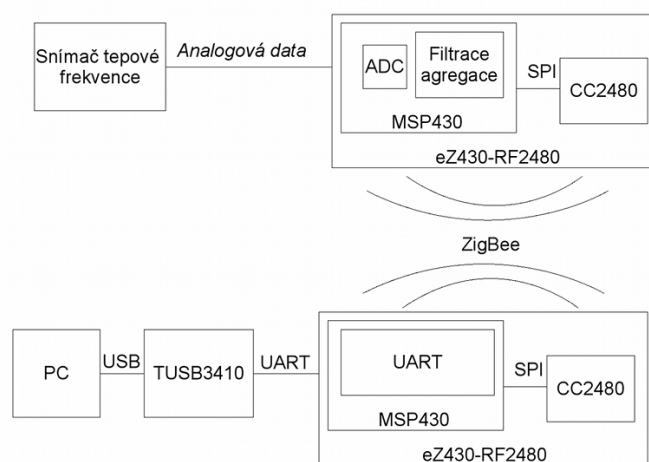
Obrázek 4.10: Schéma zapojení a výsledná realizace

# 5 Realizace bezdrátového měření tepové frekvence

Nyní, když byl určen potřebný teoretický základ ke všem částem projektu je nasnadě popsat všechny komponenty měřicího systému a jejich vzájemnou komunikaci. K tomu účelu se dobře hodí blokový diagram.

## 5.1 Blokové schéma měřicího systému

Celý měřicí systém lze dekomponovat na jednotlivé dílčí části a ty následně řešit samostatně. Měřicí senzor generuje analogová data, která je potřeba nějakým způsobem vzorkovat a zpracovávat. Je tudíž nutné použití AD převodníku. Koncové zařízení musí data zpracovat, upravit a bezdrátově přenést ke koordinátoru, ten informaci přijme a odešle do PC. V osobním počítači jsou data přijata a graficky zobrazena. Celou situaci přehledně popisuje obrázek 5.1.



Obrázek 5.1: Blokové schéma

Bezdrátový přenos zpráv technologií ZigBee je společný pro obě dvě strany systému, proto bude rozebrán ve společné kapitole 5.2 *Přenos zpráv*, v kapitole 5.3 *Aplikace* je popsáno nastavení a funkce obou jednotek .

## 5.2 Přenos zpráv

Před tím, než bude uveden princip přenosu zpráv je nutné zmínit několik skutečností, které vedly k výběru toho typu přenosu. Systém měření tepové frekvence je systém informační, nic nereguluje ani nijak neřídí. Není tudíž kladen tak veliký důraz na zaručené přenesení zprávy ani na rychlost přenosu. Koncové zařízení by také mělo být energeticky co možné nejúspornější a při vysílání/přijímání má jednotka poměrně veliký proudový odběr (řádově desítky mA). Snaha je tudíž tyto časy minimalizovat.

Tyto skutečnosti vedly k volbě nejjednoduššího možného typu přenosu, tj. s náhodným přístupem ke komunikačnímu kanálu a bez potvrzení přenesené zprávy.

### 5.2.1 Aplikační protokol

Protože systém počítá s případným použitím více koncových zařízení, mohlo by při komunikaci s centrálním prvkem docházet k případným kolizím, nelze ani vyloučit situaci, kdy není zpráva úspěšně doručena z důvodů nejrůznějších rušení atd. Bylo by proto dobré uvažovat o možnosti jestli by přece jen nešly případné kolize při přenosu nějakým způsobem eliminovat či detekovat a následně opravit.

Nejprve rozvažme, jaká data budou z koncového zařízení směrem ke koordinátoru přenášena. Vzhledem k tomu, že veškerá informace, kterou snímač přenáší se omezuje pouze na průměrný čas mezi dvěma pulzy za jednotku času (1s), stačí tedy přenášet pouze počet sekund a milisekund. K tomu stačí jeden byte pro každé číslo. Dle normy IEEE 802.15.4 je přenášená informace (payload) v každém packetu 0-104 bytů (viz kapitola 3.1) [J. Koton(2006)], je zde značný prostor k přidání ještě nějakých informací do přenášené zprávy. Pokud navíc nebude přesazeno oněch 104 bytů, odešle se pouze 1 zpráva, čímž se tedy nijak neprodlouží doba komunikace a tudíž ani energetické náklady.

Jaké informace tedy přidávat do přenášené zprávy, které by umožňovaly detekci, či opravu chyby? Zcela nepochybně je to číslo zprávy, případně jiná forma časové značky. To umožní koordinátoru kontrolovat kontinuitu komunikace. Tím by byla vyřešena detekce chyby. Pokud bude navíc do zprávy přidaná informace o předchozích několika měřeních, bude koordinátor při detekci chyby přenosu schopen zpětně zrekonstruovat data, které při předchozích vysíláních nebyla přenesena. Aplikační protokol popisuje obrázek 5.2. Na základě této rozvahy byl vytvořen model komunikace, který je velmi

Počet bytů	1	1	1	1	1	1	1	1	1	1	1
Typ zprávy	ID zprávy	sec <sub>i</sub>	msec <sub>i</sub>	sec <sub>t-1</sub>	msec <sub>t+1</sub>	sec <sub>t+2</sub>	msec <sub>t+2</sub>	sec <sub>t+3</sub>	msec <sub>t+3</sub>	sec <sub>t+4</sub>	msec <sub>t+4</sub>

Obrázek 5.2: Aplikační protokol

podobný přenosu s tzv. klouzavým okénkem [Ing. Jiří Ledvina(2012)]. Jeho princip znázorňuje obrázek 5.3.



Obrázek 5.3: Přenos s klouzajícím okénkem

## 5.3 Aplikace

Úvodem je nutné zdůraznit, že některé části zdrojového kódu, zejména ty, které se týkají inicializace rozhraní (SPI, UART), vytvoření ZigBee sítě a další konfigurační funkce jsou vytvořeny na základě volně dostupných projektů výrobce. Jedná se zejména o aplikaci ZASA (ZigBee Accelerator Sample Application), která tvoří v podstatě referenční projekt pro vývoj ZigBee



čítač, který měří dobu do následujícího úderu. Průměrná doba mezi dvěma pulzy je poté odeslána na koordinátor. Pojd'me si nyní celou situaci rozebrat poněkud podrobněji.

## Připojení do ZigBee sítě

Nyní přichází chvíle, kdy vývojář pracující s technologií Z-Accel nejvíce ocení celý tento koncept a zvláště pak zjednodušené aplikační rozhraní SAPI. Celý proces připojení koncového zařízení do sítě je totiž omezen na volání funkcí, které již fyzicky ovládají SPI linku a implementují konkrétní příkazy. Firma Texas Instruments navíc dává tyto funkce k dispozici jako součást aplikace ZASA. Připojení prvku do sítě a jeho konfigurace je tudíž maximálně usnadněno. Vždyť to je také jeden základních požadavků celé technologie Z-Accel!

Pro ilustraci jednoduchosti konfigurace PAN\_ID, čísla kanálu a typu zařízení je uveden zdrojový kód. Kompletní zdrojový kód je k dispozici v příloze.

```
/* typ zarizeni */
device = ZG_DEVICE_TYPE_ENDDEVICE;
zb_WriteConfiguration(ZCD_NV_LOGICAL_TYPE, 1, &tmp);
/* PAN ID site */
panID16 = 0xFFFF;
zb_WriteConfiguration(ZCD_NV_PANID, 2, &val16);
/* kanal c. 16 */
channel32 = 0x00010000;
zb_WriteConfiguration(ZCD_NV_CHANLIST, 4, &val32);
/* reset zarizeni */
zb_SystemReset();
```

Po úspěšné konfiguraci zařízení a připojení prvku do sítě generuje síťový procesor potvrzovací zprávu `START_CONFIRM` na jejím základě je tento stav indikován rozsvícením červené LED diody, zařízení je přepnuto do módu nízké spotřeby. V něm čeká zařízení na stisk tlačítka po němž započne měření.

## Snímání analogového signálu, AD převodník

V kapitole 4.4 byl popsán použitý senzor tepové frekvence. Aby bylo možné jeho výstup zpracovávat je nejprve potřeba jej převést do digitální podoby. K tomuto účelu je drtivá většina procesoru vybavena AD převodníkem, jinak je tomu i v případě MSP430 (viz kapitola 4.2). Následující řádky budou věnovány seznámení se s jeho řídicími registry a jejich nastavením.

## AD převodník

Úkolem této periferie (jak naznačuje název) je převést analogový signál na 10 bitovou digitální reprezentaci. Pro svou funkci využívá dvě programově volitelné napěťové úrovně ( $V_{r+}$  a  $V_{r-}$ ) s jejichž definuje horní a dolní hladinu vzorkovaného napětí. Převedená hodnota může být uložena jak v přímém kódu tak v dvojkovém kódu v registru ADC10MEM. Vzorkovaná digitální data ( $N_{ADC}$ ) můžeme jednoduše odvodit podle vztahu:

$$N_{ADC} = 1023 \cdot \frac{V_{IN} - V_{R-}}{V_{R+} - V_{R-}} \quad (5.1)$$

Pro generování vzorkovacího hodinového signálu může být využito 4 zdrojů: SMCLK, MCLK, ACLK a interní oscilátor ADC10OSC. K přivedení vzorkovaného signálu může být využito 9 externích a 4 interních analogových kanálů. Kanály které nejsou vybrány jsou izolovány od ADC kvůli eliminaci šumu. Jako řídicí registry procesoru slouží ADC10CTL0 a ADC10CTL1.

*Pozn.: To je samozřejmě jen zlomek možností, ale dostačují pro jednorázové navzorkování dat. Další nastavení AD převodníku popisuje datasheet procesoru MSP430F2274 oddíl 20, strana 25 [Tex(2011b)]*

V kapitole 4.1 je popsána testovací deska a piny procesoru, které jsou na ni vyvedeny. Po nahlédnutí do tabulky vyplývá, že na desce je k dispozici 6 kanálů AD převodníku.

Vzorkovaný signál je připojen k pinu P2.2, očekávané napětí je v rozmezí 0V-2V. Konfigurace AD převodníku bude tudíž následující:

```
/*16 taktů pro vzorkování, ref 2,5V, přerušení ON, zapni jádro ADC*/  
ADC10CTL0 = ADC10SHT_2 + REF2_5V + REFON + ADC10ON + ADC10IE;  
/*A2 jako vstupní kanál*/  
ADC10CTL1 = INCH_2;  
/*analogový vstup kanálu A2 povolen*/  
ADC10AE0 = 0x04;
```

Každé vzorkování AD převodníku poté vyvolá přerušení které je obsluženo konkrétní rutinou ve které dojde k aktualizaci hodnoty vzorkovaného signálu, nastaví se příznak konkrétního přerušení a AD převodník se vypne.

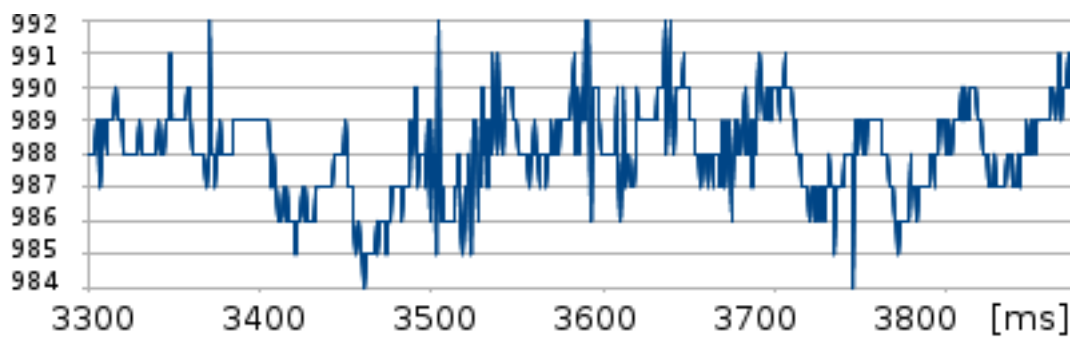
```
#pragma vector=ADC10_VECTOR  
__interrupt void isrAdc()  
{  
    halAdcVal = ADC10MEM;  
    eventFlag |= HALEVT_ADC;  
    ADC10CTL0 = 0;  
    __low_power_mode_off_on_exit();  
}
```

## Filtrování signálu

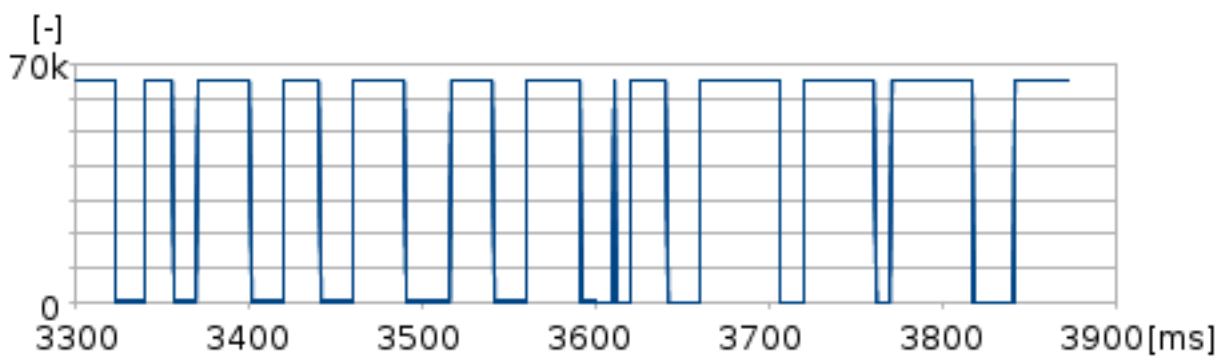
Nyní je již k dispozici digitální reprezentace analogového signálu. Bohužel jako většina analogově měřených dat, jsou i tyto hodnoty značně ovlivněny rušením. Je proto potřeba tato data filtrovat a normalizovat, aby bylo možné z nich získat informaci o aktuálním stavu srdce. Analogový signál je vzorkován a filtrován každou 1ms.

Následující úsek zdrojového kódu popisuje algoritmus filtrace a normalizace signálu, obrázek 5.5 ukazuje signál po jeho navzorkování AD převodníkem, na obrázku 5.6 je vidět vyfiltrovaný signál.





Obrázek 5.5: Vzorkovaný signál



Obrázek 5.6: Filtrovaný signál

```

Signal = halAdcVal;

/*Normalizování signálu do 0*/
readings += Signal; // počet vzorků pro normalizování
sampleCounter++; // proměnná využita jako čítač vzorků

if ((sampleCounter %300) == 0){
offset = readings / 300; // průměrování
readings = 0;
}
NSignal = Signal - offset; // v tomto místě probíhá normalizace

Lxv[1]=Lxv[0];
Lxv[0]=Lxv[1] + (NSignal-Lxv[1])/2; // Filtrování 100 Hz

FSignal = Lxv[0] * Scale; // Zesílení

if (FSignal >= Peak && Pulse == false){ // Srdeční uder způsobí, že ADC vzorek roste.
Peak = FSignal; // Hledání momentu vrcholu křivky
peakTime = sampleCounter; // Pokud je to vrchol, nastal nyní
}
/*je čas zjistit, jestli křivka nedosáhla vrcholu*/
// Kvůli úspoře času není nutné počítat každou 1ms, stačí po 20ms.
if ((sampleCounter %20) == 0){
// pokud signál několikrát po sobě klesá, byl zazanamenám puls
if (FSignal < 0 && Pulse == false){
Pulse = true; //dokud je puls, bude true
lastHRV = HRV;
ledBlink(BOTHLDIOD);
HRV = peakTime - lastPeakTime; // měření času mezi 2 pulzy
lastPeakTime = peakTime;
updateData(HRV);
}
if (FSignal > 0 && Pulse == true){ // když signál roste, můžeme vymazat proměnné
Pulse = false; // puls už není
Peak = 0; //
}
}
}

```

*Pozn.: Algoritmus filtrace je volně dostupný na stránkách výrobce snímače PulseSensor [Joel Murphy()]*

## Agregace dat

Ve chvíli, kdy algoritmus detekuje srdeční činnost, spočte čas od minulého pulsu, uloží jej do lokální proměnné `time` a aktualizuje globální proměnnou `timeToBeat` podle vzorce:

$$timeToBeat_t = 0,7 \cdot timeToBeat + 0,3 \cdot timeToBeat_{t-1} \quad (5.2)$$

Je tedy vidět, že při aktualizaci proměnné `timeToBeat` není pouze stará

hodnota nahrazena novou, ale je zde stará hodnota ještě zohledněna. Důvodem je fakt, že tepová frekvence se v čase mění poměrně spojitě a tudíž není žádoucí, aby hodnota výrazně kolísala. Například při měření byl vypočten čas 50ms, při následujícím měření se tento čas změnil na 55ms. Při výpočtu by tedy tepová frekvence kolísala:

$$\begin{aligned} \left[\frac{tep}{min}\right] \quad Tf &= \frac{100}{timeToBeat} \cdot 60 & (5.3) \\ \frac{100}{50} \cdot 60 &\doteq 120 \left[\frac{tep}{min}\right] \\ \frac{100}{55} \cdot 60 &\doteq 109 \left[\frac{tep}{min}\right] \end{aligned}$$

Rozdíl 5ms je přitom velmi malý a může být způsoben buď skutečnou změnou tepové frekvence, ale stejně tak nepřesností měřící metody či chybou při filtrování. Použijeme-li popsanou metodu postupného zpřesňování výsledku, vypočtené časy mezi jedním pulzem se změny na 50ms, resp 53.5ms. Výsledky budou následující:

$$\frac{100}{50} \cdot 60 \doteq 120 \left[\frac{tep}{min}\right] \quad (5.4)$$

$$\frac{100}{53.5} \cdot 60 \doteq 112 \left[\frac{tep}{min}\right] \quad (5.5)$$

Je také dobré vzít v úvahu, že při snímání může dojít k nepřesnostem. Např. prst může být oddělen od snímače, snímací algoritmus nedetekuje tep atd. V tom případě by samozřejmě doba mezi detekcí pulzu značně narostla (až na několika násobek skutečné hodnoty). Je proto dobré s touto situací počítat a hodnoty, které přesáhnou čtyřnásobek času mezi signály ignorovat a neměnit původní hodnotu.

Nyní si zrekapitulujme situaci. Koncové zařízení je nakonfigurováno, připojeno do fungující ZigBee sítě a každou milisekundu je vyvoláno přerušení časovače na jehož základě jsou vzorkována a filtrována data. Je tedy k dispozici vše potřebné, aby jednotka mohla každou 1s vyslat zprávu (viz obrázek 5.2) směrem ke koordinátoru. Všechna data jsou průběžně uložena v poli `buffer` které reprezentuje naposledy odeslanou zprávu. Ve chvíli kdy má dojít k dalšímu odeslání je inkrementováno `ID_zpravy` a zbylé položky v poli jsou posunuty o 2 místa vpravo, čímž dojde k realizaci plovoucího okénka, nejstarší hodnoty budou smazány a na první místo zařazeny aktuální hodnoty.

## 5.3.2 Koordinátor

### Vytvoření ZigBee sítě

Proces vytvoření nové sítě je podobný jako při připojení zařízení do sítě. Je využito rozhraní SAPI a funkcí dostupných z aplikace ZASA. Pro názornost je uveden jeden krok při konfiguraci. Kompletní zdrojový kód je uveden v příloze.

```
/* typ zarizeni */
device = ZG_DEVICE_TYPE_COORDINATOR;
zb_WriteConfiguration(ZCD_NV_LOGICAL_TYPE, 1, &tmp);
/* PAN ID site */
panID16 = 0xFFFF;
zb_WriteConfiguration(ZCD_NV_PANID, 2, &val16);
/* channel c. 16 */
val32 = 0x00010000;
zb_WriteConfiguration(ZCD_NV_CHANLIST, 4, &val32);
zb_SystemReset();
```

Každý úspěšně provedený konfigurační krok potvrzuje síťový procesor adekvátním `CONFIRM` příkazem. V případě úspěšného připojení do sítě je to `START_CNF`, pokud úspěšně odešle zařízení zprávu, generuje `SEND_DATA_CNF` atd.

### Příjem zpráv

Nyní když je síť vytvořena čeká zařízení na příjem zprávy v nekonečné smyčce. Při příjmu zprávy je vyvoláno přerušení. To generuje signál `SRDY`. V jeho obslužné rutině je detekováno o jaký typ SPI příkazu se jedná. Pokud jde

o indikaci přijatých dat RCV\_DATA\_IND zpracuje koordinátor data a v požadovaném formátu je odešle přes UART do PC.

```
/* Přerušeni vyvolala zpráva o přijetí dat */
if (rpcFlags == RCV_DATA_IND) {
    receive();
}
void receive() {
    /* Vstup do funkce, která odešle data přes UART do PC */
    printMessage(zaccelDataBuf);
}
```

## Odesílání dat přes UART

Aby bylo možné komunikovat přes rozhraní UART, musí být rozhraní inicializováno. K tomu slouží registry UCACTL0, UCACTL1, UCAMCTL, UCA0BR0, UCA0BR1 a jsou nastaveny následovně:

```
UCA0CTL1 = UCSWRST;
/* Softwarový reset */
UCA0CTL1 |= UCSSEL_3;
/* Formát rámce: bez parity; 1 stop-bit; 8 bitů / byte. */
UCA0CTL0 = 0;
/* Nastaví zdroj časování & rychlost přenosu. */
UCA0BR0 = 0x34;
UCA0BR1 = 0x00;
UCA0MCTL = UCOS16 | UCBRF0;
/* Nastavení portů P3.4, .5 - UART0 TXD, RXD */
P3SEL |= 0x30;
UCA0CTL1 &= ~UCSWRST;
/* povolení přerušeni */
IE2 |= UCA0RXIE;
```

S těmito parametry tedy probíhá přenos přes UART do USB programátoru a následně přes USB do PC. Formát přenášené zprávy po UART je identický jako zpráva přenášená od koncového zařízení ke koordinátoru (viz obrázek 5.2).

## 6 Programové vybavení pro vizualizaci naměřených dat

Pro vizualizaci naměřených hodnot byl vytvořen program WorkOut v programovacím jazyce Java v prostředí NetBeans. Program může sloužit jako nástroj pro průběžné zobrazování aktuální tepové frekvence (viz obrázek ??), případně umožňuje zobrazit historii měření formou grafu (viz obrázek ??).

Uživatelský manuál je k dispozici v příloze , případně se v příloze nachází stručný popis všech nezbytných tříd programu. Podrobná dokumentace je vygenerována programem JavaDoc a je k dispozici na přiloženém CD.

## 7 Porovnání finálního produktu s komerčním prostředkem

Pro otestování vyvinutého měřícího systému byl zapůjčen komerční Sport-Tester Sigma PC 15 od firmy Sigma Sport (viz 7.1). Za jehož zapůjčení děkuji své přítelkyni. Testování probíhalo následovně. s Výsledky měření ukazuje



Obrázek 7.1: Komerční sport tester Sigma PC 15

následující tabulka. Vzhledem k tomu, že není k dispozici žádná zaručeně přesná metoda, je použit jako referenční komerční produkt a chyba je počítána v poměru k jeho výsledkům

Sigma PC 15[tep/min]	Přípravek	Chyba
30	29 - 31	<1%
60	55-67	8%
90	80-102	10%
120	105-130	±12%
150	145-153	10%
180+	značně kolísá	>20%

Z měření je patrné, že při ustálené hodnotě chyba při měření vyšší se vzrůstající frekvencí srdečního úderu.

Při rychle se měnící tepové frekvenci se ukázala nepřesnost metody a zejména nepřesnost při filtrování a průměrování. V tomto směru by zcela nepochybně bylo potřeba provést rozsáhlejší testování s různými nastaveními filtrovacích konstant a zvážit další metody agregace dat.



## 8 Závěr

V teoretické části práce je vysvětlena tepová frekvence, jsou rozebrány některé významné děje či poruchy rytmu, které mohou nastat během srdečního cyklu. Dále jsou uvedeny základní metody snímání tepové frekvence včetně jejich kladů a záporů, případně principů realizace. Na základě této rozvahy je vybrána metoda, která bude použita při vývoji systému snímání tepové frekvence.

Úkolem druhé poloviny teoretické části je popsat síť ZigBee, její OSI model, prvky sítě a typy topologií. Dále se blíže seznámit s přípravkem eZ430-RF2480, zejména s oběma jeho procesory, modelem komunikace a komunikační API.

V rozsáhlejší praktické části je krok po kroku rozebrán proces vývoje měřícího systému. Základním krokem je návrh blokového schématu a následně popsání každé z jeho částí. Následující kapitoly popisují každou jednu část, od sestavení fungující bezdrátové sítě, vytvoření programu pro členy sítě až po vizualizaci naměřených dat. Poslední částí práce je porovnání vytvořeného přípravku s komerčním produktem.

Při testování bylo docíleno následujících výsledků: Při klidné tepové frekvenci se rozdíl mezi komerčním SportTesterem a vytvořeným přípravkem pohybuje kolem  $\pm 10\%$ , při prudkých změnách tepové frekvence a vysoké frekvenci (180+ tep/min) dochází k nepřesnosti způsobené průměrováním hodnot a nepřesností snímací metody zhruba  $\pm 30\%$ .

V současné pracovní verzi se snímač nedá příliš použít v reálném provozu, chybí zde optimalizace spotřeby u koncového zařízení, jednotky by bylo třeba opatřit vhodným krytem, je třeba doladit průběžné připojování a odpojování dalších jednotek za běhu měření a nebylo provedeno testování provozu při více připojených koncových zařízeních.

Vizualizační program je připraven pro širší využití. Je zde implementováno vytváření nových uživatelů, správa jejich databáze a možnost vytvoření SQL databáze. S těmito údaji by v budoucnu bylo možné do systému přidat informaci o spálených kaloriích při cvičení, sledovat dlouhodobou aktivitu cvičících atd. To však přesahuje rámec bakalářské práce.

# Literatura

- [SPI()] *What is SPI*. Dostupné z: <http://www.fpga4fun.com/SPI1.html>.
- [zig()]
- [Fak(2010)] *E-Learningové prostředí pro integraci výuky klinických předmětů na LF a FZV UP Olomouc*. Fakulta zdravotnických věd Univerzity Palackého v Olomouci, 2010. Dostupné z: <http://pfyziol1fup.upol.cz/castwiki2/?p=4338>.
- [Ing. Jiří Ledvina(2012)] ING. JIRÍ LEDVINA, C. Přednášky k předmětu Úvod do počítačových sítí. 2012.
- [J. Koton(2006)] J. KOTON, V. K. P. v. *Standard nízkorychlostní bezdrátové komunikace ZigBee*. České vysoké učení technické, 2006. Dostupné z: <http://access.feld.cvut.cz/view.php?cisloclanku=2006032001>.
- [Joel Murphy()] JOEL MURPHY, Y. G. *Pulse Sensor*. Dostupné z: [www.pulsesensor.com](http://www.pulsesensor.com).
- [Mas(2011)] *Pulsní oxymetr Masimo radical 7*. Masimo, 2011. Dostupné z: <http://www.hoyer.cz/produkty/intenzivni-pece/pulsni-oxymetry/masimo-radical-7/>.
- [Tex()] *eZ430-RF2480 User's Guide (Rev. A)*. Texas Instruments. Dostupné z: <http://www.ti.com/lit/ug/swru151a/swru151a.pdf>.
- [Tex(2006)] *CC2480 Developer's Guide*. Texas Instrumets, 2006. Dostupné z: <http://focus.ti.com/lit/an/swra176/swra176.pdf>.
- [Tex(2011a)] *MSP430F2x7x code examples (Rev. E)*. Texas Instrumets, 2011a. Dostupné z: <http://www.ti.com/litv/zip/slac123d>.

- 
- [Tex(2011b)] *MSP430x2xx Family User's Guide (Rev. E)*. Texas Instrumets, 2011b. Dostupné z: <http://focus.ti.com/lit/ug/slau144e/slau144e.pdf>.
- [Šeda(2010)] SEDA, J. Měřič tepové frekvence, 2010.

# Seznam obrázků

2.1	Reflexivní a průsvitová fotoelektrická metoda . . . . .	3
2.2	Komerční oxymetr firmy Masimo [Mas(2011)] . . . . .	4
3.1	Referenční model ZigBee . . . . .	7
3.2	Topologie sítě ZigBee . . . . .	9
4.1	Rozložení vývodů procesoru na desku . . . . .	11
4.2	Vývojový kit eZ430-RF2480 . . . . .	11
4.3	Popis komponent procesoru MSP430 . . . . .	12
4.4	Rozložení pinů procesoru MSP430 . . . . .	13
4.5	Koncept Z-Accel . . . . .	14
4.6	Příkaz AREQ . . . . .	15
4.7	Příkaz SREQ . . . . .	16
4.8	Příkaz POLL . . . . .	16
4.9	Konflikt při komunikaci . . . . .	16
4.10	Schéma zapojení a výsledná realizace . . . . .	19
5.1	Blokové schéma . . . . .	20

---

5.2	Aplikační protokol . . . . .	22
5.3	Přenos s klouzajícím okénkem . . . . .	22
5.4	Stavový diagram aplikace . . . . .	23
5.5	Vzorkovaný signál . . . . .	27
5.6	Filtrovaný signál . . . . .	27
7.1	Komerční sport tester Sigma PC 15 . . . . .	33
A.1	Výběr COM portu . . . . .	41
A.2	Zobrazení aktuální tepové frekvence . . . . .	41
A.3	Zobrazení tepové frekvence formou grafu . . . . .	42

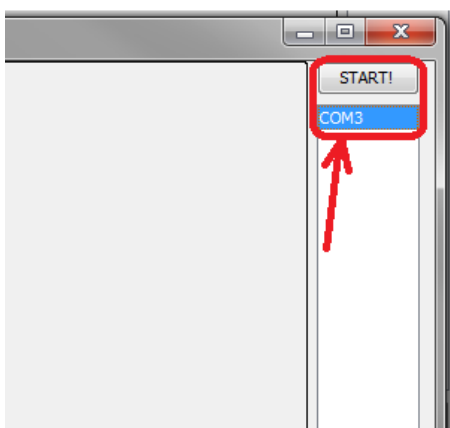
## Přehled použitých zkratek

IKG	Impedanční kardiografie
IEEE	The Institute of Electrical and Electronics Engineers
MAC	Media Access Control
NTW	Network layer
SSP	Security Services Provider
APL	Application Layer
USB	Universal Serial Bus
SPI	Serial Peripheral Interface
API	Application Programming Interface
PC	Personal Computer
UART	Universal Asynchronous Receiver and Transmitter
ZASA	ZigBee Accelerator Sample Application
LED	Light Emitting Diode
AREQ	Asynchronous Request
SRDY	Synchronous Request
ZDO	ZigBee Device Objects
PAN	Personal Area Network
SMCLK	Sub-System Master Clock
MCLK	Master Clock
ACLK	Auxiliary Clock

# A Přílohy

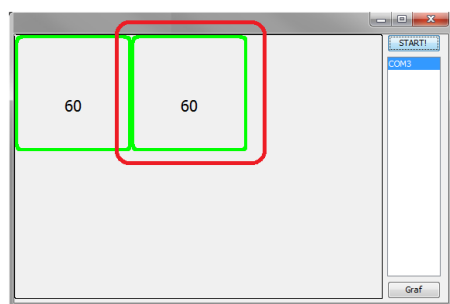
## A.1 Uživatelská příručka programu WorkOut

Po spuštění aplikace je třeba vybrat jeden z dostupných portů, na kterém je připojen USB FET programátor (viz A.1). Po vybrání jednoho z portů



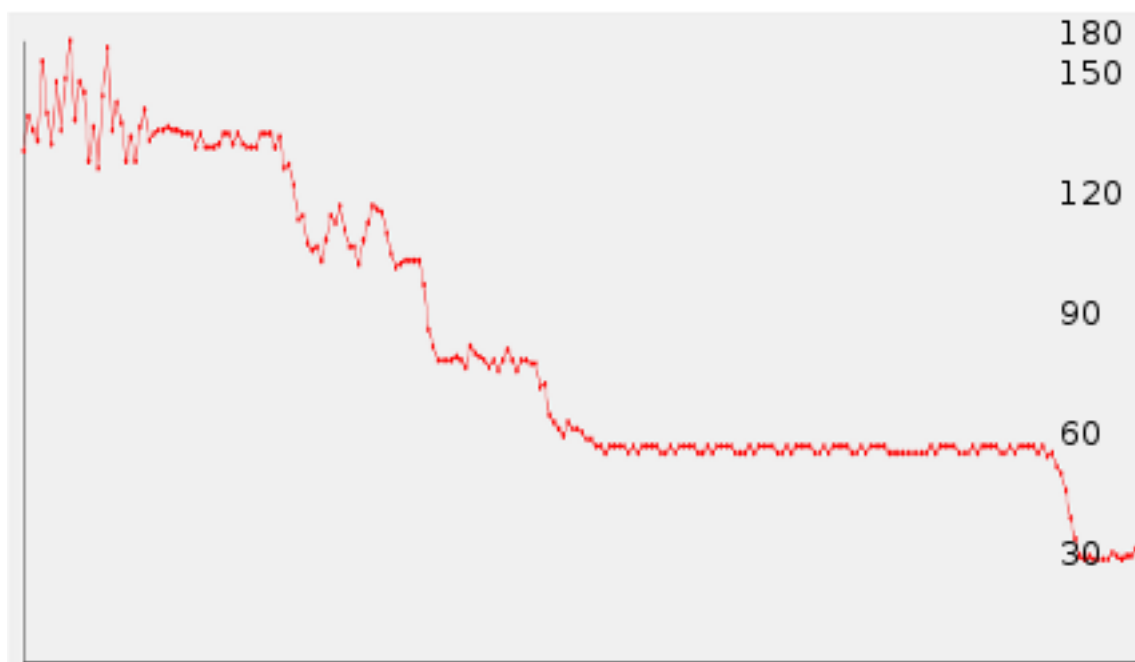
Obrázek A.1: Výběr COM portu

lze již program spustit tlačítkem **START**. Program je nyní nakonfigurován na zobrazování aktuální tepové frekvence dvou jednotek. Stisknutím tlačítka



Obrázek A.2: Zobrazení aktuální tepové frekvence

**Graf** se zobrazí historie měření tepové frekvence.



Obrázek A.3: Zobrazení tepové frekvence formou grafu

## A.2 Popis tříd programu WorkOut

Program se skládá z 4 hlavních tříd a několika dalších pomocných knihoven pro realizaci grafického uživatelského rozhraní.

- **PortTester.java** - třída inicializující komunikaci se sériovou linkou. Umožňuje nastavit parametry přenosu a vytvoří seznam dostupných COM portů.
- **PortReader.java** - vlákno které realizuje čtení hodnot ze sériové linky. A zpracovává přijatá data.
- **Message.java** - datová třída reprezentující poslední obdrženou zprávu.
- **User.java** - datová třída reprezentující jednoho uživatele, jeho váhu, výšku, věk, jméno. Využívá kolekce ArrayList pro uchování všech doručených zpráv.
- **BeepThread.java** - vlákno které jednou za stanovený čas změní barvu rámečku.



- **OneUserForm.java** - grafická třída která zobrazuje rámeček v němž je aktuální tepová frekvence.
- **WorkOutView.java** - hlavní třída uživatelského rozhraní, zobrazuje formulář, na něm okno každého uživatele. Umožňuje přepnout zobrazení na vizualizaci pomocí grafu.
- **HeartRateMonitor.java** - zobrazuje graf tepové frekvence, používá k tomu historii zpráv.