

**ZÁPADOČESKÁ UNIVERZITA V PLZNI
FAKULTA ELEKTROTECHNICKÁ**

KATEDRA APLIKOVANÉ ELEKTRONIKY A TELEKOMUNIKACÍ

DIPLOMOVÁ PRÁCE

Inteligentní podprsenka

ZÁPADOČESKÁ UNIVERZITA V PLZNI

Fakulta elektrotechnická

Akademický rok: 2020/2021

ZADÁNÍ DIPLOMOVÉ PRÁCE

(projektu, uměleckého díla, uměleckého výkonu)

Jméno a příjmení: **Bc. Jan MORÁVKA**
Osobní číslo: **E17N0049P**
Studijní program: **N2612 Elektrotechnika a informatika**
Studijní obor: **Telekomunikační a multimediální systémy**
Téma práce: **Inteligentní podprsenka**
Zadávající katedra: **Katedra elektroniky a informačních technologií**

Zásady pro vypracování

Ověření možností nositelné elektroniky a smart textilií na inteligentní podprsence.

1. Prostudujte a popište technologie smart textilií a nositelné elektroniky pro použití v inteligentní podprsence.
2. Zvolte vhodné komunikační subsystémy inteligentní podprsenky z hlediska možnosti realizace, odběru, ceny a údržby.
3. Navrhněte elektronický komunikační subsystém inteligentní podprsenky.
4. Podle možností tento realizujte subsystémy.
5. Zhodnoťte dosažené výsledky.

Rozsah diplomové práce: **40 – 60 stran**
Rozsah grafických prací: **podle doporučení vedoucího**
Forma zpracování diplomové práce: **tištěná/elektronická**

Seznam doporučené literatury:

Žalud, V.: Radiokomunikační systémy pro internet věcí. In Radiokomunikace 2017 Pardubice ISBN 978-80-87942-33-8

Vedoucí diplomové práce: **Ing. Richard Linhart, Ph.D.**
Katedra elektroniky a informačních technologií

Datum zadání diplomové práce: **9. října 2020**
Termín odevzdání diplomové práce: **27. května 2021**



Prof. Ing. Zdeněk Peroutka, Ph.D.
děkan

L.S.



Doc. Ing. Jiří Hammerbauer, Ph.D.
vedoucí katedry

V Plzni dne 9. října 2020

Abstrakt

Předložená práce je zaměřena na inteligentní textilie, nositelnou elektroniku a technologie s tím spojené. Ve stručnosti se také zabývá představením internetu věcí. Práce v úvodní části obsahuje rešerše stávajících technologií, bezdrátových komunikačních standardů a protokolů, a řeší, které z nich jsou pro danou aplikaci nejvhodnější z hlediska energetické náročnosti, kvality a rychlosti přenosu dat a z hlediska spolehlivosti. V druhé části se práce zabývá návrhem a realizací prototypu komunikačního systému včetně obslužného software zařízení, které spadá do kategorie nositelné elektroniky.

Pro testování navrženého systému byla zvolena vývojová deska LoPy4 firmy Pycom společně se senzorovým modulem Pysense. Následné testování zahrnuje sběr dat ze senzorů (zrychlení, napětí baterie, EKG, srdeční frekvence, tělesná teplota), měření parametrů datového přenosu a měření energetické náročnosti zařízení s využitím různých podporovaných technologií (Wi-Fi, LoRa a Sigfox). Posledním bodem práce je zpracování a vizualizace dat v cloudovém uložišti a jeho službách. Navrhovaný systém může nalézt uplatnění v aplikacích jako je například nositelná elektronika, zdravotnictví nebo měření enviromentálních jevů.

Klíčová slova

Smart textilie, inteligentní textilie, textilní senzory, Sigfox, LoRa, internet věcí, Pycom, LoPy4, Pysense, Python, MicroPython, Pymakr, Pybytes, Wia

Abstract

Submitted thesis focuses on study in the field of smart textile, wearable electronics and affiliated technologies. It also introduces the Internet of Things briefly. At first, the thesis contains literature search of nowadays used technologies, wireless communication standards and protocols, and deals with which one of them is best suited for the presented application in terms of power consumption, quality and speed of data transmission and reliability. In the second section, the thesis deals with the design and construction of a prototype of communication system and its software for a device that falls into the area of wearable electronics.

For the testing purposes of the proposed system, development board LoPy4 together with Pysense sensor module from Pycom company were chosen. Following testing includes collecting the data from the sensors (acceleration, battery voltage, EKG, heart beat rate, temperature), measuring parameters of data transmission and measuring of power consumption while using various supported transmission technologies (Wi-Fi, LoRa and Sigfox). Lastly, the thesis deals with the processing and visualization of the data in cloud services. The proposed system can find usage in applications such as wearable electronics, health care and also measurement of environmental phenomena.

Key words

Smart textile, intelligent textile, textile sensors, Sigfox, LoRa, Internet of Things, Pycom, LoPy4, Pysense, Python, MicroPython, Pymakr, Pybytes, Wia

Prohlášení

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně, s použitím odborné literatury a pramenů uvedených v seznamu, který je součástí této diplomové práce.

Dále prohlašuji, že veškerý software, použitý při řešení této diplomové práce, je legální.

.....

podpis

V Plzni dne 27.5.2021

Bc. Jan Morávka

Obsah

OBSAH	7
ÚVOD	9
SEZNAM SYMBOLŮ A ZKRATEK	11
1 SMART TEXTILIE	14
1.1 KLASIFIKACE SMART TEXTILÍ.....	15
1.1.1 Pasivní smart textilie.....	15
1.1.2 Aktivní smart textilie.....	15
1.1.3 Ultra smart textilie.....	15
1.2 MATERIÁLY A TECHNOLOGIE PRO VÝROBU SMART TEXTILÍ.....	16
1.3 TEXTILNÍ SENZORY.....	17
1.4 ELEKTRODY.....	20
1.5 NOSITELNÁ ANTÉNA.....	21
1.6 DESKY PLOŠNÝCH SPOJŮ.....	22
1.7 PROPOJENÍ CHYTRÉ TEXTILIE S NOSITELNOU ELEKTRONIKOU.....	25
1.8 DOSTUPNÉ SMART TEXTILNÍ SYSTÉMY NA TRHU.....	26
2 INTERNET VĚCÍ	28
2.1 VYUŽITÍ IOT.....	29
2.1.1 Další uplatnění.....	29
2.2 ARCHITEKTURA IOT.....	30
2.3 DATOVÁ KOMUNIKACE.....	30
2.3.1 Komunikační standardy.....	31
3 TECHNOLOGIE BEZDRÁTOVÉ KOMUNIKACE	32
3.1 SIGFOX.....	33
3.2 LORA.....	36
3.3 WI-FI.....	39
3.4 BLUETOOTH.....	40
3.5 NARROW BAND LTE.....	40
3.6 VYUŽITÍ TELEKOMUNIKAČNÍCH TECHNOLOGIÍ PRO NOSITELNOU ELEKTRONIKU.....	41
4 NÁVRH SYSTÉMU, VÝVOJ SOFTWARE A REALIZACE PROTOTYPU	44
4.1 DATOVÁ NÁROČNOST A ÚSPORA OBJEMU PŘENÁŠENÝCH DAT.....	45
4.2 ZVOLENÉ TELEKOMUNIKAČNÍ TECHNOLOGIE.....	48
4.3 HARDWARE.....	49
4.4 SOFTWAREVÉ PROSTŘEDKY.....	54
4.4.1 Program obsluhy senzorů a řízení bezdrátové komunikace.....	56
4.4.2 Sigfox backend.....	56
4.4.3 LoRa backend.....	58
4.5 PLATFORMY IOT.....	61
4.5.1 Pybytes.....	61
4.5.2 Wia.....	64
5 OVĚŘENÍ PARAMETRŮ SYSTÉMU	75
5.1 HODNOTY SNR A RSSI.....	75
5.2 SPOTŘEBA ELEKTRICKÉ ENERGIE.....	77
ZÁVĚR	83
SEZNAM LITERATURY A INFORMAČNÍCH ZDROJŮ	86

SEZNAM OBRÁZKŮ	93
SEZNAM TABULEK	94
PŘÍLOHY	1
PŘÍLOHA A – DOPROVODNÉ OBRÁZKY K AKTUALIZACI FIRMWARU DESKY PYSENSE.....	1
PŘÍLOHA B – PROGRAM OBSLUHY SENZORŮ A ŘÍZENÍ BEZDRÁTOVÉ KOMUNIKACE PRO PYBYTES	3
PŘÍLOHA C – PROGRAM OBSLUHY SENZORŮ A ŘÍZENÍ BEZDRÁTOVÉ KOMUNIKACE	5
PŘÍLOHA D – ÚPRAVA KNIHOVNY LTR329ALS01.PY	12

Úvod

Předkládaná práce je zaměřena na technologie inteligentních textilií, nositelné elektroniky, bezdrátových komunikací a ve stručnosti také na internet věcí (také označovaný jako IoT, z anglického Internet of Things). Všechny tyto vědní oblasti jsou v současnosti velmi často diskutovány ve světě komunikačních a informačních technologií. Propojení věcí, čidel a senzorů a jejich integrace do textilií a nositelné elektroniky je již poměrně stará myšlenka, nicméně až v posledních letech zažívá svůj skutečný průlom. Podle odborníků má vývoj v těchto oblastech budoucnost a potenciál transformovat náš svět do něčeho, co si někteří z nás ani nedovedou představit. Internet věcí, smart textilie a nositelná elektronika jsou velmi zajímavé oblasti s mnoha novými příležitostmi pro firmy i jednotlivce, a proto se o ně zajímá čím dál více lidí. Problematika těchto oblastí a jejich vzájemného propojení je velmi rozsáhlá a kvůli tomu se zdá být obtížné ji uchopit v celé její šíři.

První, rešeršní část práce, slouží k seznámení se současným stavem a problematikou v oblasti smart textilií, jejich klasifikací, technologiemi a materiály pro jejich výrobu a možnostmi jejich využití v nositelné elektronice.

Cílem teoretické části práce je ve stručnosti uvést problematiku internetu věcí a jeho různých odvětví. Dále se tato část práce zabývá bezdrátovými telekomunikačními technologiemi, které jsou pro nositelnou elektroniku a IoT využívány.

Cílem praktické části práce je zvolit a navrhnout elektronický komunikační systém pro následnou realizaci prototypu zařízení, které spadá do kategorie nositelné elektroniky. V úvodu této části je popsán návrh, jak by mohl vypadat kompletní systém pro takové zařízení včetně požadavků na něj. Dále je představen použitý hardware, který byl zvolen tak, aby disponoval širokou škálou funkcí včetně konektivity k bezdrátovým sítím Wi-Fi, Bluetooth a LPWAN.

Dalším cílem je vývoj vlastního obslužného programu, který bude schopen odesílat data získaná ze senzorů a další užitečné informace prostřednictvím zvolených bezdrátových sítí. S tím souvisí i úprava zaznamenaných dat za účelem úspory přenášeného objemu dat.

Součástí této práce je také následná vizualizace a zpracování dat ve zvolených cloudových službách.

V poslední části se práce zabývá ověřením parametrů realizovaného bezdrátového přenosu, spotřebou elektrické energie zařízení pro různé bezdrátové sítě a zhodnocením dosažených výsledků.

Seznam symbolů a zkratek

API.....	Rozhraní pro programování aplikací (Application Programming Interface)
B (kB, MB, GB)..	bajt, jednotka velikosti dat (Byte, kilobyte, megabyte, gigabyte)
BLE.....	Bluetooth Low Energy
CE	Označení prohlášení o shodě výrobku s nařízeními Evropské unie
CIoT.....	Spotřebitelský internet věcí (Consumer Internet of Things)
CRC	Kontrolní součet umožňující detekci chyb (Cyclic Redundancy Check)
DBPSK	Diferenční binární klíčování fázovým posuvem (Differential Binary Phase Shift Keying)
DPS.....	Deska plošných spojů
DSSS.....	Technika modulace signálu s rozprostřeným spektrem (Direct Sequence Spread Spectrum)
EIoT	Podnikový internet věcí (Enterprise Internet of Things)
EKG	Elektrokardiogram
EMA	Evropská agentura pro léčivé přípravky (European Medicines Agency)
EMG	Elektromyogram
EUI	Unikátní identifikátor zařízení (Extended Unique Identifier)
FCC.....	Označení prohlášení o shodě výrobku s nařízeními ve Spojených státech amerických (Federal Communications Commission)
FDA	Úřad pro kontrolu potravin a léčiv Spojených států amerických (Food and Drug Administration)
FHSS.....	Metoda modulace signálu s rozprostřeným spektrem technikou skoků ve frekvenci (Frequency Hopping Spread Spectrum)
FFC	Ohebné plochá kabely (Flex Flat Cable)
FPC.....	Ohebné desky plošných spojů (Flex Printed Circuit)
HTTP	Hypertextový přenosový protokol (Hyper Text Transfer Protocol)
Hz (kHz, MHz, GHz)..	Hertz, jednotka kmitočtu (kilohertz, megahertz, gigahertz)
I2C	Označení typu sériové sběrnice (Inter-Integrated Circuit)
I2S.....	Označení typu sériové sběrnice určené pro přenos zvukových dat (Inter-IC Sound)
IIoT	Průmyslový internet věcí (Industrial Internet of Things)
IoT	Internet věcí (Internet of Things)

LAN	Lokální počítačová síť (Local Area Network)
LPWAN	Nízkopříkonové síť velkého dosahu (Low Power WAN)
LTE	Označení technologie vysokorychlostního přenosu dat v mobilních sítích (Long Term Evolution)
M2M	Typ komunikace mezi stroji (Machine to Machine)
M2P	Typ komunikace mezi strojem a člověkem (Machine to People)
MAC	Unikátní identifikátor síťového zařízení (Media Access Control)
MEMS	Mikro-Elektro-Mechanické Systémy
MQTT	Síťový protokol pro přenos zpráv (Message Queuing Telemetry Transport)
NB/UNB	Úzkopásmová / Ultra úzkopásmová technologie přenosu dat (Narrow Band / Ultra Narrow Band)
NFC	Technologie rádiové bezdrátové komunikace mezi elektronickými zařízeními na velmi krátkou vzdálenost (Near Field Communication)
NVRAM	Napěťově nezávislá paměť s libovolným přístupem (Non-Volatile RAM)
OTAA	Typ autentizace zařízení v síti LoRa (Over-The-Air-Activation)
P2P	Typ komunikace mezi lidmi (People to People)
QRS	Označení části průběhu křivky EKG
RAM	Paměť s libovolným přístupem (Random Access Memory)
RFID	Identifikace pomocí rádiové frekvence (Radio Frequency IDentification)
RGB	Barevný model červená-zelená-modrá (Red-Green-Blue)
RSSI	Indikátor úrovně přijímaného signálu (Received Signal Strength Indicator)
SF	Činitel rozprostření (Spread Factor)
SIM	Účastnická identifikační karta (Subscriber Identity Module)
SNR	Poměr signálu k šumu (Signal to Noise Ratio)
SPI	Sériové periferní rozhraní (Serial Peripheral Interface)
TCP/IP	Sadu komunikačních protokolů pro počítačové síť (Transmission Control Protocol/Internet Protocol)
TTN	Otevřená, decentralizovaná infrastruktura pro internet věcí (The Things Network)
UART	Počítačová sběrnice sloužící k asynchronnímu sériovému přenosu dat (Universal Asynchronous Receiver-Transmitter)
URL	Řetězec znaků, který slouží k přesné specifikaci umístění zdrojů

- informací na internetu. (Uniform Resource Locator)
- USB Univerzální sériová sběrnice pro připojení periférií k počítači (Universal Serial Bus)
- WAN..... Rozlehlá síť (Wide Area Network)
- WLAN Bezdrátová lokální počítačová síť (Wireless Local Area Network)
- WPAN Bezdrátová personální síť (Wireless Personal Area Network)
- WWAN..... Bezdrátová WAN (Wireless Wide Area Network)
- ZIF Typ konektoru pro elektroniku (Zero Insertion Force)

1 Smart textilie

Smart textilie, v překladu inteligentní či chytré textilie, jsou velice širokou oblastí produktů, které rozšiřují funkčnost a užitečnost běžných tkanin. Tyto textilie obsahují textilní produkty, tedy nitě, vlákna nebo příze, vytvářející tkanou, netkanou nebo pletenou strukturu. Tato struktura spolu s elektronickými obvody, které mohou být zakomponovány v textiliích nebo jsou externě připojitelné, může komunikovat s okolním prostředím nebo uživatelem textilie. Samotné inteligentní textilie se také popisují jako materiály, jež mohou snímat podmínky a podněty okolního prostředí, například mechanické, teplotní, chemické, elektrické či magnetické zdroje, a reagovat na ně. Ať už se jedná o samotné smart textilie nebo i ty s integrovanými elektronickými prvky, jde o textilie s přidanou hodnotou, které díky svým vlastnostem poskytují možnost využití například ve zdravotnictví, v ochranných oblecích, v automobilovém průmyslu či při sportovních aktivitách. Při integraci elektronických prvků do textilií je nutné brát ohled na to, aby bylo jejich nošení pohodlné, neomezovalo pohyb, příliš nezvyšovalo hmotnost, aby bylo možno oblek prát apod. [1][2][4]

Funkce inteligentních textilií lze rozdělit do 5 základních skupin, kterými jsou snímání, zpracovávání dat, ovládání, komunikace a napájení. Největší rozvoj nastal v oblasti snímání, čímž se výrazně zvýšila možnost využití této technologie. Pomocí chytrých textilií s integrovanými elektronickými prvky je možné měřit teplotu těla, EKG či EMG, akustické veličiny srdce či plic, tok krve, biologické a chemické parametry, dýchání, krevní tlak, radiaci, množství potu a mechanické nebo elektrické parametry kůže. Jednou z největších výhod takových textilií je schopnost stálého monitorování těchto parametrů za pomoci zařízení, která jsou schopna dle naměřených hodnot sestavit profil uživatele. Tento profil je následně možné porovnávat s aktuálními naměřenými hodnotami, které lze později v případě jejich odchýlení od standardu vyhodnotit jako nebezpečné mnohem rychleji, než je běžně možné. [1]

Původní impuls pro významný rozvoj v této oblasti vzešel z armádních a vesmírných výzkumů, protože v těchto oblastech jsou lidé permanentně vystaveni vysokému ohrožení života a nutnost ochrany člověka a komunikace s ním v těchto velmi náročných podmínkách byla prioritou. Výzkum inteligentních materiálů a textilií a jeho výsledky byly později přeneseny i do běžného života. Vývoj v oblasti zdravotnictví byl nastartován tlakem

populace na kvalitu zdravotní péče, která zahrnuje monitorování, diagnostiku, terapii a také odborný přístup při zotavování pacienta. [1][3][5]

1.1 Klasifikace smart textilií

Inteligentní textilní systém byl definován jako textile obsahující akční členy, senzory a jednotky ke zpracování dat. Smart textile byly postupem času vylepšovány od první generace (pasivní smart textile) přes aktivní smart textile až do současného stavu, kdy jsou využívány a vyvíjeny tzv. ultra smart textile. [3][4][6]

1.1.1 Pasivní smart textile

Pasivní smart textile, také označované jako funkční textile, poskytují oproti běžným textiliím různé doplňkové funkce. Pracují v pasivním režimu, což znamená, že aktivně nereagují na změny okolního prostředí (například proměnná vlhkost, teplota, intenzita slunečního záření a další) a ani nijak nemění svoji funkčnost či vlastnosti. Tyto funkce mohou být například antimikrobiální, proti zápachu, tepelně izolační, protibalistické, žáruvzdorné, proti UV záření apod. Zdroje se neshodují, zda pasivní smart textile označovat za inteligentní nebo ne, jelikož samy nejsou schopné aktivně reagovat na podmínky prostředí. [1][3][7]

1.1.2 Aktivní smart textile

Druhou generací jsou textile již obsahující senzory a akční členy. Tyto smart textile aktivně přizpůsobují své vlastnosti v závislosti na okolních podmínkách. Aktivní inteligentní textile mohou využívat například materiály s fázovou změnou, tvarovou pamětí nebo barviva měnící svou barvu v závislosti na okolní teplotě. Díky takovým materiálům mohou například akumulovat či regulovat teplotu, být paropropustné (hydrofilní / neporézní), voděodolné nebo mohou pohlcovat páru. [1][3][7]

1.1.3 Ultra smart textile

Poslední generací jsou ultra smart textile, které dokáží samostatně vnímat a reagovat na okolní podmínky nebo podněty a také mají schopnost těmto podmínkám autonomně přizpůsobovat svou funkčnost, resp. adaptovat se. Tyto textile obsahují nejen senzory a akční členy, ale také elektronickou řídicí jednotku. Jednotka vyhodnocuje vnější podmínky a na jejich základě přizpůsobuje funkčnost inteligentní textile. Zároveň může zajišťovat

komunikaci s uživatelem. Výroba těchto textilií spojuje klasický textilní průmysl s dalšími obory, jako jsou technologie senzorů a akčních členů, materiálové vědy, telekomunikace, chemický a elektrotechnický průmysl apod. [1][3][7]

1.2 Materiály a technologie pro výrobu smart textilií

Funkční textilní materiály tvoří základní součást smart textilních systémů. Stávající technologie, které jsou současně vyvíjeny za specifickým účelem smart textilní aplikace, zahrnují materiály s fázovou změnou, termochromní materiály, materiály s tvarovou pamětí, polymery, optická vlákna, fotovoltaické články, mikro-elektronické mechanické systémy MEMS a další. Pro výrobu smart textilií je požadováno, aby tyto materiály byly lehké, pružné, mechanicky odolné, levné a snadno zpracovatelné. [1][3][7][8]

Například membrány paropropustných a zároveň voděodolných textilií mohou být realizovány několika způsoby. Nejčastěji se jedná o tenký (~0,01 mm) mikroporézní film z expandovaného polytetrafluorethylenu, polyurethanu nebo polyesteru., který je laminován na povrch textilie. Mikroskopické póry membrány jsou dostatečně velké na to, aby jimi mohly procházet přibližně 700krát menší molekuly vody ve formě vodní páry a zároveň jsou přibližně 20.000krát menší než kapka vody. Tyto membrány mají navíc na povrch nanášeny vrstvy s malým povrchovým napětím (oproti vodě), což způsobí, že se kapky vody k membráně nepřilnou, ale sklouznou po ní dolů. Další typy voděodolných membrán využívají materiály, které po zvlhčení nabobtnají, a zabrání tak průniku vody. [9][10][12]

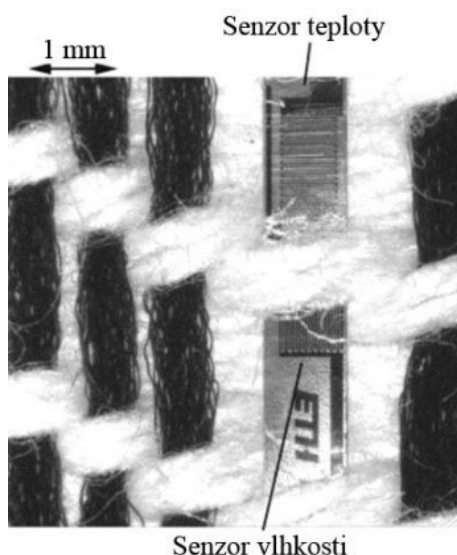
Materiály s tvarovou pamětí (slitiny kovů nikel-titan, zlato-kadmium, měď-hliník-nikel a další, nebo polymery obsahující směsi styrenu, butadienu, polyethylen tereftalátu, polyurethanu apod.) mohou být využity k výrobě oblečení schopného měnit své tepelně izolační vlastnosti. Například je-li tento materiál integrovaný mezi dvěma vrstvami textilie vystaven nízkým teplotám, zdeformuje se a zvětší vzduchovou mezeru mezi těmito vrstvami čímž zvýší tepelnou izolaci celého souvrství. Naopak po následovném vystavení vyšším teplotám se materiál vrací do svého původního tvaru a vzduchová mezera se zmenšuje. Dalším příkladem může být triko, které za zvýšených teplot dokáže samo ohrnout rukávy nebo zvýšit svoji prodyšnost. [9][11]

K výrobě senzorů či pouze jako propojovací substrát se používají vodivé kompozitní nebo čistě kovové nitě a vlákna nebo vodivé pasty. Výhoda použití kompozitních vláken je v přiblížení vlastností výrobku k vlastnostem běžných textilií, jako je pružnost a ohebnost. Vodivé nitě a vlákna se vyrábí kombinací elektricky vodivého materiálu a textilního vlákna. Mohou to být například vlákna plněná uhlíkem nebo kovovými částicemi, vlákna potažená vodivými polymery nebo vlákna spředená s tenkou kovovou nití. Tato různá vodivá vlákna mohou být snadno integrována přímo do textilní struktury pomocí běžných technologií výroby textilu, například vyšívání, tkaní či pletení. V případě vodivých past se využívá sítotisku, který je levný, jednoduchý, dostupný a nemusí respektovat strukturu tkaniny. Tyto pasty musí obsahovat vhodný elektricky vodivý materiál, jako je například stříbro, měď a zlato. Sítotiskem může být dosaženo rozlišení až 200 μm . Nevýhodou sítotisku však je, že vytvrzené pasty mají špatnou mechanickou odolnost a v náročných podmínkách či při praní může dojít k přerušení tištěného obvodu. Další možností je tzv. rozprašování, které zajistí lepší mechanickou odolnost. Technologií rozprašování může být dosaženo rozlišení až 100 μm . Provádí se ve vakuové komoře, ve které je terč ze zlata rozprašován plazmou argonu. Ke tkanině je v této komoře přiložena stínící maska se vzorem obvodu, přičemž část rozprášených atomů zlata prochází skrz vzor v masce a dopadá na tkaninu, na které se takto vytvářejí elektrické obvody. Ve srovnání se sítotiskem je tento proces poměrně drahý a časově náročný. [1][2][3][6][7][8][13][14]

Aby bylo možné elektronické zařízení nosit na lidském těle integrované v textilií, musí být elektronické součástky a propojení na textilních substrátech spolehlivé, odolné, flexibilní a nesmí být pro svého nositele zbytečně omezující nebo nepohodlné. [3][6][13]

1.3 Textilní senzory

Většina senzorů používaných k monitorování životních funkcí a pohybu uživatele musí být v těsném či přímém kontaktu s tělem. Z tohoto důvodu musí být oděvy pružné a přiléhavé, přičemž je kladen důraz na pohodlí. Jedním z možných způsobů provedení textilních senzorů je všíti miniaturních flexibilních substrátů, na kterých je nanesen vodivý polymer například pro měření vlhkosti nebo vodivá plocha či meandr měřící teplotu. [1][3][14][15][16]



Obr. 1.1: Ukázka senzoru vlhkosti a teploty na tenkém polyimidovém pásku vetkaném do textilie (převzato z [16] a upraveno)

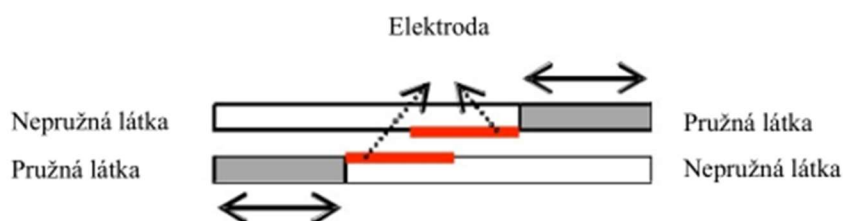
Textilní senzory je možné dělit na pasivní a aktivní. Aktivní snímače převádí vnitřní vstupní energii na rozdíl elektrického potenciálu přímo, zatímco pasivní senzory potřebují externí napájecí zdroj. Tabulka 1.1 poskytuje přehled o senzorech používaných pro monitorování životních funkcí lidského těla. [3][6]

Tabulka 1.1: Přehled typů a snímané veličiny senzorů (převzato z [6] a upraveno)

Snímaná veličina	Původ signálu	Snímací zařízení	Typy produktů
Elektrokardiogram	Elektrická aktivita srdce	Bioelektrody	Tkané, pletené, vyšíváné, tištěné, naprašované nebo kapacitní elektrody
Elektromyogram	Elektrická aktivita svalů	Bioelektrody	Tkané, pletené, vyšíváné, tištěné, naprašované nebo kapacitní elektrody
Dýchání	Expanze a kontrakce hrudníku vlivem dýchání	Piezoelektrické, piezorezistivní senzory, měření bioimpedance	Piezoelektrická, piezorezistivní vodivá vlákna, elektrody
Okysličení krve	Absorpce světla hemoglobinem v krvi	Optické senzory	Světelné zdroje a detektory, optická vlákna
Pohyb	Kinematika těla	Piezorezistivní a optické senzory, akcelerometr	Vlákna vodivých polymerů, optická vlákna, akcelerometr
Teplota	Teplota těla	Termočlánek, termistor	Vlákna vodivých polymerů, kovová vlákna, elektronické součástky

- **Kapacitní senzory**

Kapacitní senzory jsou využívány pro aplikace, ve kterých je snímána například svalová aktivita, pohyb a držení těla nebo působení tlaku. Senzor je založen na principu kapacitoru. Skládá se z polymerního dielektrika a textilních elektrod. Princip měření spočívá ve vzájemném pohybu elektrod v jejich rovině, což vede ke změně aktivní oblasti elektrod, a tak i ke změně kapacity. K tomu například dochází, když pacient dýchá a jeho hrudník se rozšiřuje a stahuje. Viz obr. 1.2. [3][16]



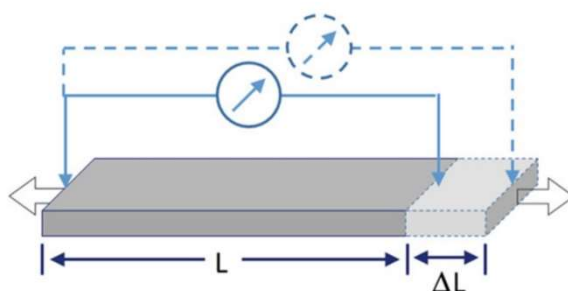
Obr. 1.2: Struktura netkaného textilního kapacitního dýchacího senzoru (převzato z [16] a upraveno)

- **Piezoelektrické senzory**

Princip těchto senzorů spočívá v mechanickém namáhání piezoelektrického materiálu, což vede ke vzniku gradientu elektrického potenciálu uvnitř materiálu. Tyto senzory mohou produkovat elektrické signály i při velmi malém namáhání a reagují na téměř jakýkoliv typ deformace. Piezoelektrické senzory se používají ve formě tenkého filmu mezi dvěma elektrodami a lze je relativně snadno integrovat do textilií. Lze je využít například jako tlakové snímače v autosedačkách nebo v hrudním pásu snímajícím srdeční tep a dýchací cyklus. [2][3][6]

- **Piezorezistivní senzory**

Princip těchto senzorů je založen na změně délky (ΔL) kovového vlákna vlivem mechanického namáhání a s ní spojenou změnou elektrického odporu tohoto vlákna. Viz obr. 1.3. [3][6]



Obr. 1.3: Princip piezorezistivních senzorů (převzato z [6])

Piezorezistivní textilní senzory mohou například měřit teplotu a pnutí anebo monitorovat životní funkce pomocí různých textilních struktur všitých například v rukavicích, kalhotách, botách, pásech, trikách a podprsenkách. [3][6]

- **Optické senzory**

Princip optických senzorů spočívá v měření změny intenzity, fáze, polarizace nebo vlnové délky světla. Tyto senzory mají uplatnění v aplikacích pro snímání teploty, tlaku, deformace nebo přítomnosti plynu. Jsou malé a lehké. Optické snímače ve formě flexibilních polymerních vláken lze snadno integrovat do textilií a jsou vhodné i z důvodu odolnosti vůči nadměrnému mechanickému namáhání. [3][6]

1.4 Elektrody

Monitorování elektrického potenciálu biologického původu vyžaduje stabilní signál s nízkým podílem šumu. Toho lze dosáhnout zajištěním dobrého elektrického kontaktu mezi pokožkou lidského těla a měřicím zařízením. K tomu slouží právě elektrody, které zde fungují jako rozhraní. Elektrody mohou být polarizovatelné (také tzv. suché) nebo nepolarizovatelné (také tzv. mokré). [3][17]

- **Nepolarizovatelné elektrody**

Jedná se o často používané gelové elektrody. Díky gelu vzniká vodivá cesta mezi elektrodou a pokožkou pacienta, kterou volně protéká elektrický proud. Elektroda se v podstatě chová jako rezistor. Gel také zajišťuje dobrou přilnavost elektrod, takže je možné je snadno upevnit na pokožku. To napomáhá prevenci artefaktů způsobených pohybem elektrody na pokožce. Výhodou nepolarizovatelných elektrod je, že ukazují velmi čisté a spolehlivé signály (např. EKG). Nevýhodou je nemožnost dlouhodobého pozorování, protože gel vysychá, a tím

se zhoršuje přenos signálu z těla na elektrodu. Gel také může způsobit podráždění pokožky. Proto nejsou vhodné pro dlouhodobé monitorování. [3][17]

- **Polarizovatelné elektrody**

Při použití polarizovatelných elektrod žádný elektrický proud neprotéká. Namísto toho v důsledku změny iontové koncentrace na rozhraní elektrody s pokožkou vzniká posuvný proud. Tyto elektrody se chovají jako kapacitory. Pracují na základě kapacitní vazby mezi vodivým materiálem a pokožkou. Ke své funkci nepotřebují žádný gel, postačuje vlhkost a pot na pokožce. Nejjednodušší polarizovatelná elektroda je kovový disk s tenkou izolační vrstvou. Jednou z pokročilejších elektrod je vodivá textilie, jejíž výhodou je podobnost vlastností s obyčejnými tkaninami, jako například pružnost a dobrá tvarovatelnost. Použití polarizovaných elektrod je omezené z důvodu jejich citlivosti na pohybové artefakty. Těm lze částečně zabránit vhodným uložením elektrod v textilií, která by v tomto případě udržovala těsný kontakt mezi elektrodou a pokožkou. Na rozdíl od gelových elektrod jsou textilní elektrody díky svým vlastnostem vhodnou volbou pro dlouhodobé pozorování nebo pro zařízení pro monitorování pacientů v domácí zdravotní péči. [3][17]

1.5 Nositelná anténa

Pokrok v oblasti výroby vodivých textilií umožnil významný vývoj i v oblasti nositelných antén. Anténa umožňuje přenos dat ze zařízení umístěných uvnitř oděvu na zařízení mimo oděv, a proto je bývá nezbytnou součástí systému. Pro využití v inteligentních textiliích musí být nositelná anténa tenká, lehká, nenáročná na údržbu, mechanicky odolná a snadno integrovatelná do textilie i do radiofrekvenčních (RF) obvodů. [2][3][18]

Pro blízkou komunikaci v rámci inteligentních textilií se nejčastěji používají tkané nebo vyšíváné vícesmyčkové antény. Tyto antény musejí být navrženy tak, aby odolávaly rušení způsobenému přítomností lidského těla a aby návrh kompenzoval změny parametrů antény vlivem změny její geometrie v závislosti na pohybu člověka. Mezi další anténní systémy patří skládané dipólové anténní soustavy, pokovené tkaninové flíčkové a spirálové antény a mikropásková anténní pole. [2][3][18]

1.6 Desky plošných spojů

Desky plošných spojů (DPS) jsou základní propojovací strukturou jednotlivých součástek téměř všech elektronických zařízení, včetně těch určených pro nositelnou elektroniku. Jejich základem je elektroizolační a zároveň nosný substrát, na kterém je nanесena vrstva mědi. Fotolitografickým procesem se z původně jednolitě plochy mědi utváří komplexní propojovací struktury. Na takto vytvořené desky se mohou nanášet další elektroizolační a vodivé vrstvy, a vytvářet tak vícevrstvé desky plošných spojů. Například základní desky v počítačích nebo mobilních telefonech mají běžně 8-10 vrstev. Zatímco pro elektrická propojení se používá výhradně měď, jako nosný substrát se používá mnoho různých materiálů, které dávají deskám plošných spojů jejich elektroizolační, mechanické a další vlastnosti. [60][61][62][63][64][65][66][67][68][69]

Pro nositelnou elektroniku a její integraci do smart textilií jsou rozhodujícími vlastnostmi DPS jejich velikost, tvar, ohebnost, mechanická a chemická odolnost, hmotnost, odolnost vůči vlhkosti a teplotním změnám, náročnost na výrobu a cena. Z hlediska ohebnosti se DPS dělí na pevné, ohebné a na hybridní (kombinace pevných a ohebných). Kromě flexibility je výhodou ohebných DPS také jejich menší velikost a hmotnost a dielektrické vlastnosti. Často bývají odolnější vůči vlhkosti, chemikáliím a teplotním změnám. Například nejběžnější substrát pro pevné DPS, FR4, má teplotu skelného přechodu (T_g) okolo 140 °C (kvalitnější až 170 °C) a relativní permitivitu 4 až 4,5. Oproti tomu ohebné DPS mají teplotu skelného přechodu až 260 °C a relativní permitivitu 3 až 3,7. Jako substrát pro ohebné DPS se nejčastěji používají polyimidy, polyestery nebo teflon. Konkrétně to může být např. Iteq IT-180, Nelco 4000-13 nebo Isola P95. Poloměr ohybu flexibilních DPS závisí na typu desky a typu ohýbání. [60][61][62][63][64][65][66][67][68][69]

Tabulka 1.2: Možnosti ohýbání flexibilních DPS

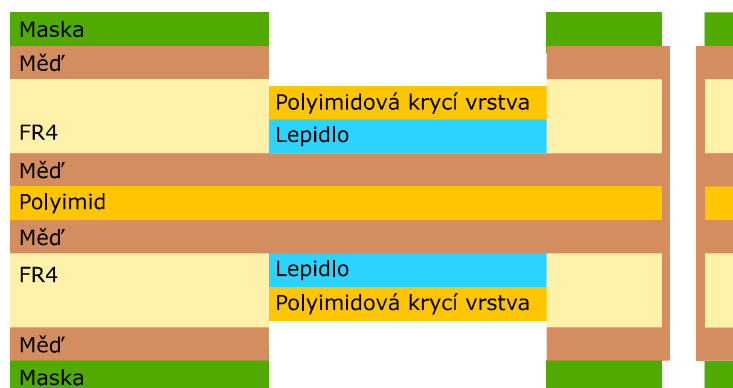
	Dynamický ohyb	Semi-dynamický ohyb	Statický ohyb
Četnost ohýbání	časté	max. 20x	pouze k instalaci
Počet vrstev	doporučeno max. 2	doporučeno max. 4	1–10
Minimální poloměr ohybu [v násobcích výšky DPS]	100–150x	20–50x	10x (1–2vrstvé DPS) 20x (vícevrstvé DPS)

Tabulka 1.3: Srovnání polyimidových a polyesterových substrátů flexibilních DPS

Vlastnost substrátu	Polyimid	Polyester
Tloušťka substrátu	12,5–125 μm	50–125 μm
Pevnost v tahu	170 MPa	140–240 MPa
Prodloužení při porušení	70 %	60–165 %
Provozní teplota	-125 až +240 °C	-60 až 105 °C
Teplotní roztažnost	20 ppm / stupeň Celsia	27 ppm / stupeň Celsia
Permitivita	3,4	3

I přes své výhodné vlastnosti nejsou ohebné DPS použitelné všude. Použití ohebných či pevných DPS závisí především na účelu, rozměrech, vlastnostech a použití navrhovaného zařízení. Ve většině případů je nejvíce žádaná kombinace obou typů desek. V takovém případě je pak nutné zabývat se propojením ohebné a pevné DPS. [60][61][62][63][64][65][66][67][68][69]

Nejspolehlivějším propojením mezi ohebnou a neohebnou DPS je výroba hybridní DPS. V tomto případě se v podstatě jedná o jednu DPS, která má pevnou a flexibilní část, viz následující obrázek. [60][61][62][63][64][65][66][67][68] [69]



Obr. 1.4: Čtyřvrstvá hybridní deska plošných spojů (převzato ze [66] a upraveno)

Výhoda spojení ohebných a neohebných DPS pomocí výroby hybridních typů je, že odpadá nutnost použití různých konektorů a pájených nebo lepených spojů. Kromě již zmíněné spolehlivosti jsou tak hybridní DPS jednodušší a rychlejší na finální výrobu. Zároveň lze prostor, který by zabíral konektor, využít jinak, anebo může být celá deska menší. [60][61][62][63][64][65][66][67][68][69]

V případě, kdy je flexibilní DPS využívána k připojení výměnných modulů k základní desce nebo design zařízení neumožňuje využít hybridní DPS, je nutné využít konektory. K těmto účelům slouží FPC (Flat Printed Circuit), FFC (Flat Flex Cable) a ZIF (Zero Insertion Force) konektory určené právě pro ohebné plošné spoje, viz následující obrázek. [60][61][62][63][64][65][66][67][68][69] [70]

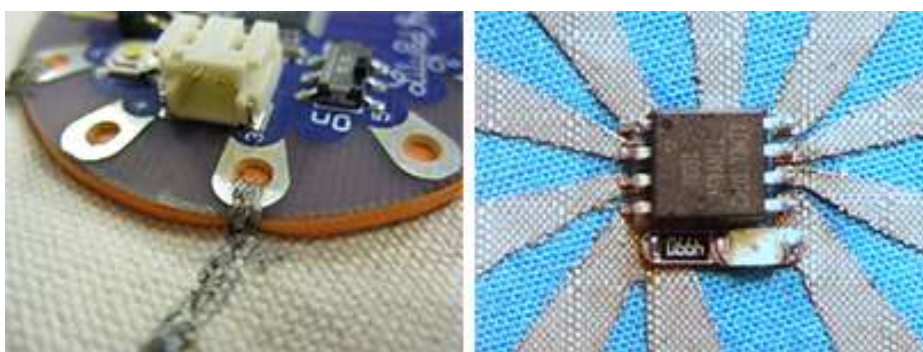


Obr. 1.5: Ukázka konektorů pro flexibilní plošné spoje (převzato ze [70])

Tyto konektory se nejčastěji vyrábí z nylonu a mohou mít velice malé rozteče kontaktů (0,3 mm), kterých může být až 80. Dále mohou být nízké, i pod 1 mm, a mohou operovat ve velkém teplotním rozsahu (-40 až 105 °C, někdy dokonce až 150 °C). Většinou jsou tyto konektory určené maximálně na 50 V a 0,5 A, výkonnější verze i na 125 V a 1 A. Velkou nevýhodou těchto velmi malých konektorů je jejich malá odolnost. Většina výrobců uvádí jako maximální počet cyklů spojení – rozpojení pouze 20. [70]

1.7 Propojení chytré textilie s nositelnou elektronikou

Propojení chytré textilie, resp. vsíťých senzorů, s elektronickým řídicím systémem může být permanentní nebo rozpojitelné. Permanentním spojením může být například přišitý či vpletený kovový vodič, jiný úsek vodivých textilních přízí nebo přímo DPS. Další možností permanentního spojení může být přilepení vhodným vodivým lepidlem nebo pájení. Při pájení je třeba brát ohled na použitou textilií a její teplotní odolnost vzhledem k vysokým teplotám pájení. Nevýhodou permanentních spojení je nutnost již při návrhu zařízení počítat s tím, že se textilie budou čistit a prát. Elektronika tudíž musí být navržena tak, aby také odolala pracím a čistícím cyklům. Takové zařízení tím bude výrazně dražší a náročnější na výrobu.



Obr. 1.6: Příklady permanentních spojení – přišití (vlevo) a pájení (vpravo) (převzato z [87][88])

Rozpojitelná spojení umožňují před praním chytré textilie elektronický řídicí systém z oblečení sejmout, což odstraňuje nevýhody permanentních spojení. Tato spojení jsou realizována různými cvočky, magnetky, patentky nebo druky, do kterých se nasouvají hroty. Další výhodou těchto propojek je, že se často vyrábí z niklu, mědi nebo mosazi, a díky tomu je možné je pájet k dalším součástem systému. Nikl je hůře pájitelný než mosaz či měď, ale zato je výrazně odolnější vůči korozi.

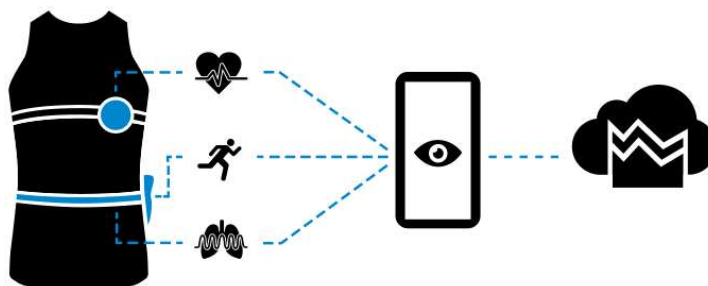


Obr. 1.7: Příklady rozpojitelných spojení – druky (vlevo) a patentky (vpravo) (převzato z [86])

1.8 Dostupné smart textilní systémy na trhu

- **Hexoskin**

Smart textilní triko kanadské společnosti Hexoskin obsahuje několik flexibilních senzorů. Tyto senzory zaznamenávají dechovou a srdeční aktivitu pomocí 1svodového EKG a břišního a hrudního pásu všitých do trika. Dále triko obsahuje tříosý akcelerometr pro sledování pohybů uživatele. Na konci roku 2018 uvedla společnost na trh vylepšenou verzi s názvem Astroskin. Tato verze poskytuje vylepšené 3svodové EKG a navíc ještě senzory pro měření teploty těla, tlaku a okysličení krve. Obě tyto verze umožňují sledování měřených hodnot v reálném čase pomocí technologie Bluetooth v aplikaci v mobilním telefonu, a to po dobu až 48 hodin. Zároveň společnost Hexoskin vyvinula vlastní online platformu. Cena těchto produktů se pohybuje od 6 000 až po 13 000 Kč. [71]



Obr. 1.8: Hexoskin a jeho online platforma (převzato ze [71] a upraveno)

- **Emglare**

Česká společnost Emglare, nyní sídlící v USA, nabízí smart textilní sportovní trika a podprsenky, ale také klasickou podprsenku. Všechny tyto produkty jsou schopné monitorovat EKG a mají integrovanou baterii. Baterie poskytuje energii až pro 16 hodin nepřetržitého záznamu a je dobíjena bezdrátově. Data jsou přenášena do mobilního telefonu pomocí Bluetooth. Jakýkoliv z těchto produktů je k dostání za 5 500 Kč. [80]



Obr. 1.9: Produkty Emglare (převzato ze [80] a upraveno)

- **HeartIn**

HeartIn je společnost založená v roce 2014 v Miami, která v roce 2017 uvedla svůj produkt HeartIn Fit ve dvou verzích – smart textilní triko a sportovní podprsenka. Oba tyto produkty mají všité dvě elektrody schopné snímat EKG. Naměřená data jsou následně přenášena do aplikace v mobilním telefonu pomocí technologie Bluetooth. Udávaná výdrž baterie je 24 hodin a cena produktu je 6 600 Kč. [75][76][77]

- **Master Caution**

Za zmínku stojí také Master Caution od izraelské společnosti HealthWatch. Toto smart textilní triko je výjimečné tím, že je pravděpodobně jediné na trhu, které umožňuje monitorovat kompletní 12svodové EKG. Dále ještě obsahuje senzory teploty, dýchání a držení těla. Všechna tato data jsou přenášena pomocí technologie Wi-Fi nebo mobilní sítě 4G do Master Caution cloudu, odkud jsou následně dostupná jak pro uživatele, tak pro praktického lékaře či kardiologa. [78][79]

- **Nexkin**

Francouzská společnost Chronolife představila na konci roku 2019 smart textilní triko podobné Hexoskinu. Pomocí 10 senzorů umožňuje kontinuální sledování různých fyziologických parametrů – EKG, břišního a hrudního dýchání, tělesné teploty a tělesné aktivity. Tato data jsou následně pomocí Bluetooth přenesena do aplikace v mobilním telefonu, kde jsou vyhodnocena. Přehled fyziologických dat je také k nahlédnutí ve webovém rozhraní. Udávaná výdrž baterie je 24 hodin. V roce 2020 obdržel Nexkin certifikace CE, FCC a EMA, avšak certifikace FDA zatím nebyla dokončena. Produkt by měl být komerčně dostupný během roku 2021. [72][73][74]

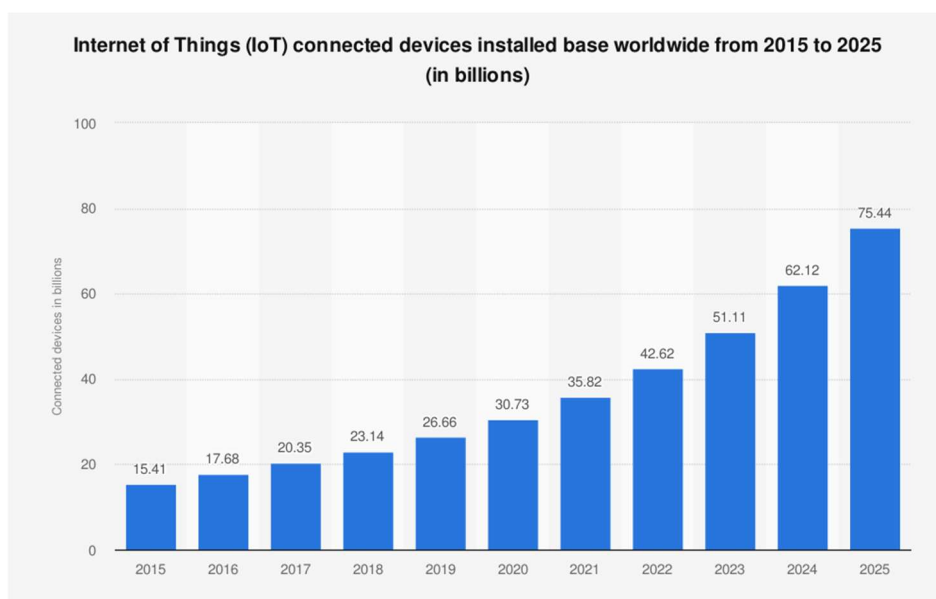


Obr. 1.10: Zleva: HeartIn [75], Master Caution [78] a Nexkin [72]

2 Internet věcí

Pojem internet věcí byl poprvé použit v roce 1999 ve stejnojmenné prezentaci Kevina Ashtona. Ten ve své prezentaci popsal systém, ve kterém byly senzory neboli věci připojeny k internetu, a dále to, jak je sdílení dat z těchto senzorů přínosné pro zvyšování vnímání okolního světa. Jako příklad použil RFID tagy, které se v obchodech používají za účelem počítání a sledování pohybu zboží bez lidského zásahu. [26][32]][33]

Dle společnosti CISCO je vznik internetu věcí datován mezi roky 2008 až 2009, kdy počet připojených zařízení k internetu překročil počet lidí světové populace. Mezi prvními zařízeními byly k internetu připojovány počítače, později chytré telefony apod. Předpokládá se, že v roce 2025 bude k internetu připojeno více jak 75 miliard zařízení (viz obr. 2.1), které vygenerují téměř 80 zettabajtů dat. [19][21][33]



Obr. 2.1: Vývoj a předpověď počtu připojených zařízení v letech 2015-2025 (převzato z [26])

Cílem IoT je propojit co nejvíce zařízení, systémů a služeb za účelem získání co nejvíce dat. Tato data lze pak převést na informace, které je možné aplikovat. Díky těmto informacím mohou vznikat systémy, které budou moci na základě takto nasbíraných dat rozhodovat a provádět různé činnosti zcela autonomně. Z toho vyplývá, že čím více dat budou zařízení schopna produkovat, tím více jich bude k dispozici pro analýzu, a tím více informací bude možné využít v každodenním životě nebo průmyslu. [22][32][33]

2.1 Využití IoT

Internet věcí se dělí do třech hlavních kategorií dle účelu využití dat. První kategorií je spotřebitelský internet věcí (Consumer IoT, CIoT), jehož smyslem je zjednodušit každodenní život člověka. Patří sem inteligentní domácnosti, nositelná elektronika a chytrá zařízení. Využití chytrých zařízení ve spotřebitelské sféře má prakticky neomezené možnosti a záleží pouze na subjektivním přání jejich uživatelů. Jako příklad lze uvést automatické nebo dálkové ovládání regulace topení a spotřebičů, detekce otevřených oken a dveří, detekce unikající vody, inteligentní lednička nebo chytré hodinky a náramky. [23][24][33]

Druhou kategorií je průmyslový internet věcí (Industrial IoT, IIoT), jehož cílem je zlepšit efektivitu využívání zdrojů. Řadí se sem například průmyslová automatizace, kde se díky vzájemnému sdílení informací dosahuje optimálního výkonu a minimálních prostojů. Dále dopravní průmysl, kde se díky sdílení dopravních informací zvyšuje plynulost dopravy a bezpečnost na silnicích. V energetickém průmyslu je cílem snižování spotřeby energií. V neposlední řadě je využíván ve zdravotnictví, kde umožňuje monitorovat zdravotní stav pacientů v nemocnicích a i těch, kteří se léčí doma. To přispívá ke zjednodušení poskytování zdravotní péče. Umožnění automatického sběru a analýzy dat o zdravotním stavu pacientů může odhalit počáteční fáze onemocnění, a napomoci tak včasnému zásahu lékařů. [25][26][27][28][33]

Třetí kategorií je podnikový internet věcí (Enterprise IoT, EIoT). Ten spočívá ve sběru a analýze dat, která jsou nezbytná pro správnou funkci tzv. Business Intelligence. Jedná se například o znalosti, dovednosti, technologie, aplikace, kvalitu, rizika a bezpečnostní otázky, modely navyšování zisků, rozšíření automatizace a zefektivnění podnikových procesů, podporu podnikového řízení, plánování, rozhodování, prediktivní analýzy, sdílení informací napříč firmou apod. [22][29][30][33]

2.1.1 Další uplatnění

Další uplatnění systémů internetu věcí se zaměřuje například na města a jejich efektivitu, životní prostředí a kontrolu jeho kvality, bezpečnost, ochranu osob a majetku, obchod, logistiku, zemědělství, chov zvířat. Konkrétní aplikace či zařízení mohou být součástí více systémů s různým zaměřením. Systémy se zaměřením na zlepšení kvality života ve městech slouží k měření škodlivých záření, úrovně radiace, hluku a koncentrace škodlivých plynů

a pevných částic. Například i naše město Plzeň se zapojilo celosvětové iniciativy Smart Cities, v jejímž rámci již spustilo několik desítek projektů z oblasti dopravy, informačních technologií, životního prostředí, vzdělávání i bezpečnosti. Podobná zařízení jako ve městech dále také umožňují detekovat lesní požáry nebo zemětřesení. Mezi společné cíle všech systémů internetu věcí patří především zvýšení efektivity využívání energií a zdrojů a zlepšení životní úrovně společnosti. [31]

2.2 Architektura IoT

U technologií IoT bohužel neexistuje jedna univerzální architektura. Různí výzkumníci vytvářeli odlišné architektury tak, aby vyhovovaly jejich konkrétním požadavkům. V základu je pro fungování IoT systému nutné, aby obsahoval řadu senzorů, komunikační síť a výpočetní část. Na tom staví i tzv. třívrstvá architektura. Tato architektura je tou nejzákladnější a byla představena během prvních výzkumů v oblasti IoT. Obsahuje tři vrstvy: fyzickou, síťovou a aplikační. [32][34]

Aplikační vrstva

Síťová vrstva

Fyzická vrstva

Obr. 2.2: Třívrstvá architektura IoT, (převzato z [34] a upraveno)

Fyzická vrstva je tvořena senzory pro sběr dat a poznatků o svém okolí. Síťová vrstva zajišťuje spojení s ostatními chytrými zařízeními, síťovými prvky a servery. Aplikační vrstva pak zodpovídá za dostupnost služeb uživateli a také definuje různé oblasti nasazení IoT, například chytré domy, města a zdravotnictví. [32][34]

2.3 Datová komunikace

Jedná se o samotný přenos digitálního signálu prostřednictvím přenosového média. Ve většině případů se jako přenosové médium pro systémy IoT využívá elektromagnetické vlnění. Komunikační rozhraní se dělí na vysílač a přijímač. V současné době se rozlišují tři typy komunikace. První z nich, Machine to Machine (M2M), je typ komunikace mezi přístroji, která umožňuje vzájemné sdílení dat bez nutnosti přítomnosti člověka. Tato technologie je využívána zejména v průmyslové automatizaci, logistice a také v oblasti dopravy. Machine to People (M2P) je typ komunikace mezi přístrojem a člověkem. Person

to Person (P2P) je přímá komunikace mezi lidmi. Možné je využití telefonování, SMS zpráv, elektronické pošty nebo sociálních sítí. [20][32][35]

V březnu 2015 vydala Komise Internetové Architektury (IAB) dokument popisující architekturu síťového propojení objektů. Tento dokument definuje čtyři způsoby propojení mezi prvky IoT systému, které je možné navzájem kombinovat a aplikovat současně na jednom produktu. Každý z těchto modelů má svá specifika použití. Device-to-device je typ přímé komunikace mezi dvěma a více zařízeními. Tento komunikační model bývá běžně uplatňován pro automatizaci domácností, kde není kladen důraz na vysoké přenosové rychlosti. Device-to-cloud je typ komunikace, kde se zařízení připojuje přímo k aplikačním službám konkrétní cloudové služby, která řídí komunikaci a výměnu dat mezi různými systémy. Cloudová služba dále umožňuje tato zařízení vzdáleně ovládat a aktualizovat je. Device-to-gateway je typ komunikace využívaný v případě požadavku spojení několika zařízení s podporou komunikačních protokolů jako Bluetooth či ZigBee s cloudovou službou. Brána v tomto případě slouží jako prostředek k propojení těchto sítí. Back-end data-sharing je typ komunikace rozšiřující device-to-cloud o sdílení dat se systémy třetích stran. Uživatelé tak mohou analyzovat data z několika různých zařízení, která jsou uložena v různých cloudových službách. [20][32][36][37]

2.3.1 Komunikační standardy

Komunikační standardy definují, co přenášená data reprezentují a jak se s nimi má nakládat. Komunikace podle standardů zajišťuje vzájemnou kompatibilitu v komunikaci mezi zařízeními různých výrobců. Při výběru vhodného standardizovaného komunikačního protokolu je třeba brát v úvahu mnoho faktorů, například množství a účel přenášených dat, způsob komunikace, zabezpečení přenosu dat a zabránění neoprávněnému přístupu k datům. [20][32]

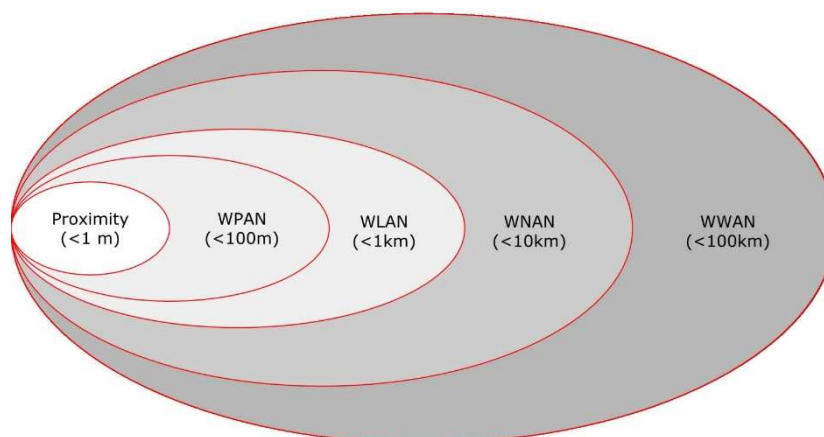
V dnešní době je jeden z nejpoužívanějších protokolů pro přenos dat mezi klientem a serverem Hyper Text Transfer Protocol (HTTP). Jedná se o jednoduchý protokol, který přenáší data určené URL adresou. Umožňuje efektivní využití síťových prostředků a také paralelní výměnu dat v jednom spojení. Dalším často používaným protokolem je Message Queuing Telemetry Transport (MQTT). Jedná se o jednoduchý, lehce implementovatelný a otevřený protokol pro zasílání zpráv mezi klienty přes tzv. brokera (centrální řídicí prvek). [20][32][38][39][40]

3 Technologie bezdrátové komunikace

Bezdrátová komunikace pro chytrá zařízení a nositelnou elektroniku probíhá na různých kmitočtech od stovek MHz po jednotky GHz. Používané jsou různé druhy komunikačních systémů. Ty lze rozdělit podle výkonu na nízkopříkonové a standardní. Mezi nízkopříkonové bezdrátové sítě, označované jako LPWAN, patří Sigfox, LoRa, ZigBee, Bluetooth LE a další. Mezi standardní bezdrátové technologie se řadí sítě Wi-Fi, Bluetooth a mobilní sítě. [20][33][41][42]

Tabulka 3.1: Rozdělení bezdrátových technologií dle dosahu [41]

Proximity	WPAN	WLAN	WNAN	WWAN
NFC	Bluetooth LE	802.11a/b/g/n/ac	Wi-SUN	2G/3G/4G
RFID	ZigBee	802.11af	ZigBee-NAN	LTE
	Thead	802.11ah		5G
	Z-Wave	802.11p		Sigfox
	WirelessHART			LoRa
	ISA100.11a			Telensa
				PTC



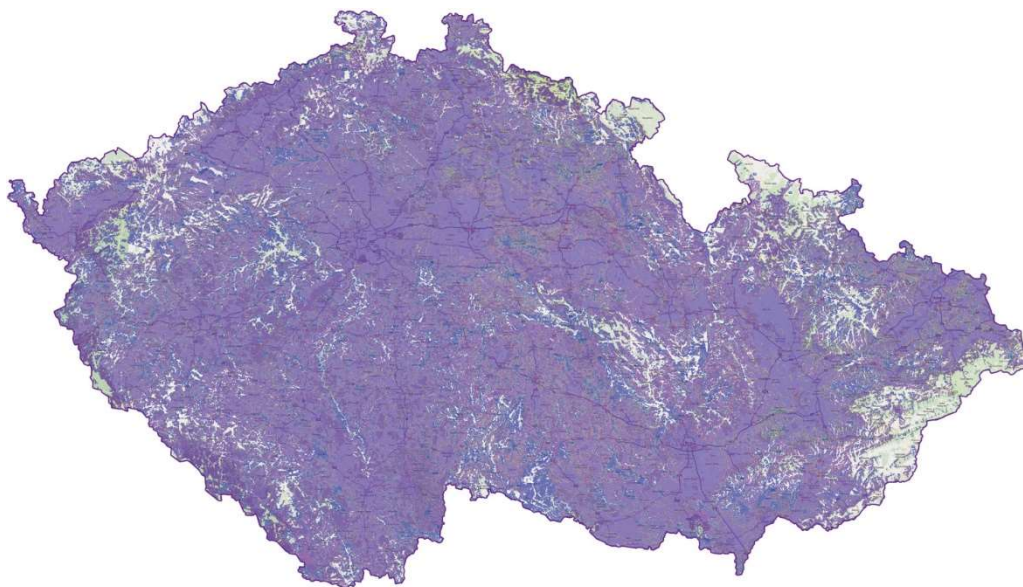
Obr. 3.1: Přehled dosahu bezdrátových technologií (inspirováno [41])

Nízkopříkonové sítě jsou navrženy za účelem minimalizace spotřeby elektrické energie. Přenosové rychlosti se zde pohybují maximálně ve stovkách kilobitů za sekundu. Výhoda těchto sítí je v jejich velkém dosahu. Ten se pohybuje v řádu jednotek kilometrů v zastavěné oblasti až po desítky kilometrů v případě přímé viditelnosti. Díky své nízké spotřebě elektrické energie jsou zařízení využívající tyto sítě schopna dlouhodobého provozu na baterie. Další výhodou je možnost připojení velkého počtu koncových zařízení k jedné základnové stanici. [20][33][41][42]

Opak k nízkopříkonovým sítím tvoří standardní bezdrátové technologie, jako jsou Wi-Fi a mobilní sítě. Přenosové rychlosti těchto technologií dosahují jednotek až stovek Mb/s. Dosah těchto sítí v husté městské zástavbě dosahuje desítek až stovek metrů a spotřeba elektrické energie bývá výrazně vyšší. Tyto technologie jsou méně odolné vůči rušení a hůře pracují při velkých počtech koncových zařízení připojících se k jedné základnové stanici. [20][33][41][42]

3.1 Sigfox

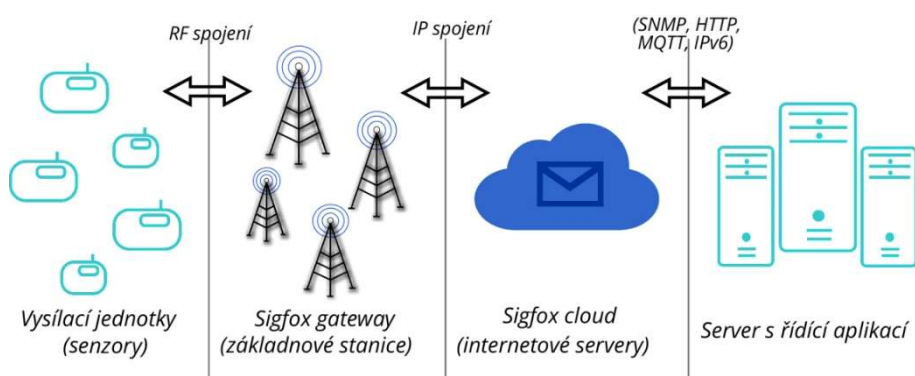
Sigfox je bezdrátový komunikační systém vyvíjený za účelem energeticky nenáročného přenosu malých objemů dat. Potenciál nachází u aplikací, kde dochází k občasnému zasílání naměřených hodnot nebo hlášení alarmových stavů. Dosah sítě v zastavěných oblastech je přibližně 2 km a mimo zastavěné oblasti dosahuje až 40 km. Prvním a zatím jediným veřejným operátorem pro technologii Sigfox v České republice je společnost SimpleCell Networks, a. s. Síť je budována ve spolupráci s mobilním operátorem T-Mobile, který poskytuje svoji infrastrukturu pro umístění základnových stanic. Obr. 3.2 ukazuje mapu pokrytí České republiky v říjnu 2020, kdy bylo pokryto 95 % území a 98 % populace. [20][33][43][44]



Obr. 3.2: Pokrytí sítě Sigfox v ČR (převzato z [43])

- **Architektura sítě**

Síť Sigfox má topologii hvězdy a je budována na buňkovém principu. Každá z těchto buněk obsahuje základnovou stanici. Základnové stanice obdržené zprávy následně přeposílají pomocí internetové komunikace TCP/IP na servery Sigfox cloudu. Jedna zpráva může být přijata více základnovými stanicemi. Tyto duplicitní zprávy se v cloudu odstraní na základě sekvenčního čísla zprávy. V cloudu jsou zprávy dále přetříděny a odeslány do zákaznických zařízení nebo zvolených cloudových služeb. [20][33][43][44]



Obr. 3.3: Architektura sítě Sigfox (převzato z [44])

Uživatelé se ke svým datům mohou dostat přímo v Sigfox cloudu, který je spravován provozovatelem sítě, nebo pomocí softwarového rozhraní API (Application Programming Interface) ze svých systémů či aplikací. Všechna zařízení s podporou technologie Sigfox jsou vybavena certifikátem a mají svůj vlastní unikátní identifikátor. Díky identifikátorům není nutné používat nastavitelné adresy (jako např. adresy IP), SIM karty apod. Identifikátor lze přirovnat k MAC adrese, kterou má každé zařízení unikátní. [20][33][43][44]

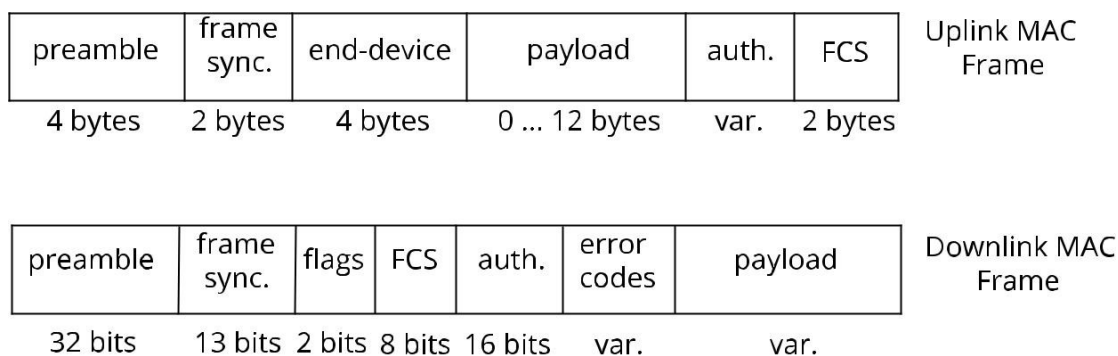
- **Přenos a modulace dat**

V Evropě využívá technologie Sigfox bezlicenční frekvenční pásmo 862-876 MHz a v USA pásmo 902-928 MHz. Maximální povolený zařízením vyzářený výkon je v Evropě 14 dBm (25 mW) a v USA 22 dBm (158 mW). Ke komunikaci se využívá úzkopásmového přenosu signálu (Ultra Narrow Band, UNB). Rychlost přenosu dat je 100 nebo 600 bit/s (závisí na regionu, v ČR 100 bit/s). Zprávy v době přenosu zabírají šířku pásma v závislosti na této rychlosti. Vysílaná zpráva je modulována pomocí DBPSK (Differential Binary Phase Shift Keying), která vyžaduje pro přenosovou rychlost 1 bit/s pouze 1 Hz šířky pásma. Při přenosové rychlosti 100 bit/s v síti Sigfox bude tedy využita šířka pásma 100 Hz. Nízké přenosové rychlosti a úzké přenosové pásmo DBPSK modulace umožňuje efektivní

využití přenosového spektra a přijímač může demodulovat i signály blízko hladině šumu. Citlivost přijímačů v síti Sigfox je regulována na základě přenosové rychlosti. Při přenosové rychlosti 100 bit/s se citlivost přijímače nastavuje na -142 dBm a při rychlosti 600 bit/s na -134 dBm. Tyto hodnoty odpovídají i limitům RSSI. Limit SNR je 8 dB. [20][33][43][44]

- **Přenosový protokol**

Přenosový protokol pro Sigfox byl vytvořen pro přenos malých objemů dat. Na obr. 3.4 je zobrazena struktura přenosového rámce. Ten je tvořen 26 bajty, z nichž jsou bajty 10 až 22 volitelné a jsou použity pro uživatelská data (tzv. payload). Zpětný kanál pro odesílání zpráv do zařízení je omezený na maximálně 8 bajtů uživatelských dat. [20][33][43][44]



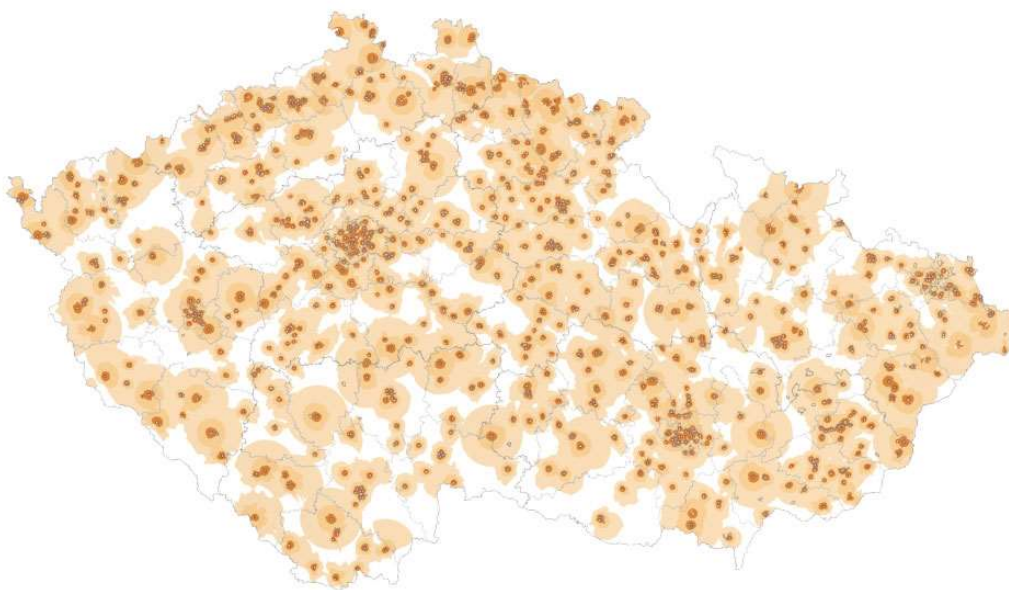
Obr. 3.4: Přenosový protokol (převzato z [45])

Síť Sigfox má relativně malou přenosovou rychlost (100 bit/s), 26 bajtů dat přenosového rámce se tak přenáší přes 2 s. Odesílané zprávy s 12 bajty dat mohou v praxi představovat např. záznam až 96 dvoustavových signálů, 6 hodnot v datovém formátu integer nebo 3 hodnoty ve formátu float. Zpětné zprávy s 8 bajty dat postačují pro jednoduché příkazy (např. zapni / vypni), změny rozsahů senzorů nebo pro změnu periody odesílání zpráv. [20][33][43][44]

Společnost SimpleCell Networks, a. s., nabízí 4 úrovně předplatného, které jsou definovány denním limitem zpráv na jedno zařízení. Denní limit pro zprávy ve směru od zařízení je 140, data je tedy možné odesílat přibližně každých 10 min. Ve směru k zařízení je pak limit 4 zprávy. Ve zbývajícím čase zařízení nekomunikuje, a tím šetří elektrickou energii. Cena předplatného je také ovlivněna počtem připojených zařízení. [43]

3.2 LoRa

Jedná se o další bezdrátový komunikační systém speciálně vyvinutý pro IoT aplikace. Jeho název je odvozen z anglického „long range“ (dlouhý dosah). Má velmi nízkou spotřebou elektrické energie (LPWAN), nízký datový tok a dosah až několik desítek kilometrů. Typicky se využívá pro měření, řízení a vzdálenou signalizaci. Sítě LoRa pracují na stejných kmitočtech a se stejným maximálním povoleným vysílacím výkonem jako síť Sigfox. Na obr. 3.5 je znázorněno pokrytí České republiky sítí LoRa v říjnu 2020. [33][46]

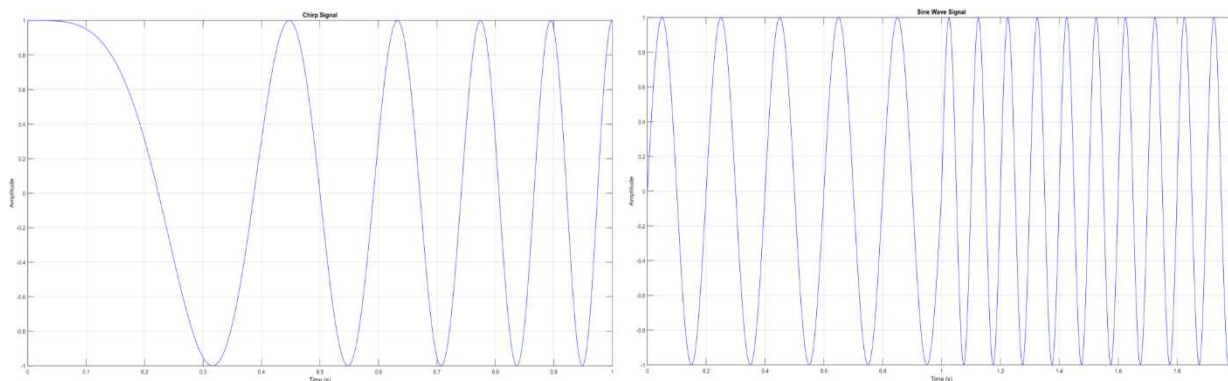


Obr. 3.5: Pokrytí sítí LoRa (převzato z [47])

Architektura sítě LoRa je v uspořádání do hvězdy. Jednotlivé uzly mohou být připojeny do veřejných nebo privátních sítí. Jednotlivé buňky sítě mohou být velikosti několika málo kilometrů v zastavěné oblasti, nebo až desítky kilometrů ve volném prostředí. [20][33][48]

- **Modulace**

Pro komunikaci v síti LoRa se využívá rozmítaná (chirp) modulace, kterou patentovala firma Semtech. Chirp je poměrně nový druh modulace. Signál je rozprostřen po celém vysílacím kanálu podobně jako u DSSS a FHSS. Rozdílem je způsob změny kmitočtu – ostatní modulace provádí skokovou změnu, zatímco chirp provádí změnu lineární. Díky tomu snižuje nároky na stabilitu zdroje kmitočtu, zvyšuje odolnost vůči Dopplerovu jevu a jeden kanál může využívat několik vysílačů současně. Tato lineární frekvenční modulace používá binární klíčování. Signál s rostoucí frekvencí reprezentuje symbol 1 a signál s klesající frekvencí symbol -1. [20][33][48][49]



Obr. 3.6: Ukázka signálu po modulaci rozmítaným signálem (vlevo) a po modulaci skokovou změnou kmitočtu (vpravo) (převzato z [33])

Technologie LoRa umožňuje volit různé přenosové rychlosti a vysílací výkon, přičemž pro každou přenosovou rychlost je pevně stanovená šířka pásma a číselník rozprostření signálu. Šířka pásma může být nastavena v rozmezí 7,8–500 kHz, typicky však 125, 250 nebo 500 kHz. Číselník rozprostření (SF, Spread Factor) udává, kolik bitů je zakódováno v každém symbolu. Volbou těchto parametrů lze ovlivňovat dosah zařízení, výdrž baterií a zároveň limity RSSI a SNR signálu. [20][33]

Tabulka 3.2: Třídy datového toku v síti LoRa (převzato z [48] a upraveno)

Třída datového toku	Konfigurace	Datový tok [b/s]	Limit RSSI [dB]	Limit SNR [dB]
0	LoRa: SF12 / 125 kHz	250	-137	-20
1	LoRa: SF11 / 125 kHz	440	-135	-17,5
2	LoRa: SF10 / 125 kHz	980	-132	-15
3	LoRa: SF9 / 125 kHz	1 760	-130	-12,5
4	LoRa: SF8 / 125 kHz	3 125	-127	-10
5	LoRa: SF7 / 125 kHz	5 470	-125	-7,5
6	LoRa: SF7 / 250 kHz	11 000	-121	-7,5
7	FSK: 50 kb/s	50 000	-	-

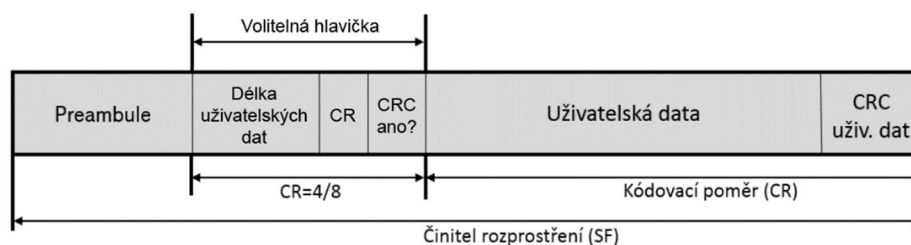
- **Zprávy**

Každá zpráva obsahuje záhlaví, které obsahuje EUI číslo. EUI je unikátní identifikátor každého zařízení, je možné jej připodobnit k MAC adrese u ethernetových zařízení. Všechny zprávy zároveň obsahují své sekvenční číslo. Všechny základnové stanice, které jsou v dosahu čidla, zprávu přijmou a přepošlou ji na server, kde se na základě sekvenčního čísla zprávy odstraňují duplicity. Na základě tohoto čísla je také možné vyhodnotit, zda nedošlo ke ztrátě jedné či více zpráv. [33]

V přenosovém rámci zprávy jsou dále obsaženy provozní informace ohledně RF vrstvy. Díky obousměrné komunikaci lze podle těchto informací monitorovat stav zařízení, měnit

spread factor a výsílací výkon. Při hustší infrastruktuře základnových stanic regulace výkonu ovlivňuje dosah signálu, čímž eliminuje rušení a zároveň pozitivně ovlivňuje energetickou náročnost. [20][33]

Maximální délka zprávy v síti LoRa je až 255 bajtů. Takto velkou zprávu lze vysílat pouze pro spread factor 7. Pro vyšší hodnoty spread factoru se velikost zprávy snižuje v souvislosti s dobou vysílání jedné zprávy. Se zkracující se zprávou se také zmenšuje prostor pro uživatelská data (payload). Při délce zprávy 255 bajtů je pro uživatelská data k dispozici 240 bajtů. Při nastavení spread factoru 12 se prostor pro uživatelská data zmenší na 51 bajtů. Struktura přenosového rámce je znázorněna na následujícím obrázku. [20][33]



Obr. 3.7: Struktura LoRaWAN rámce (převzato z [58] a upraveno)

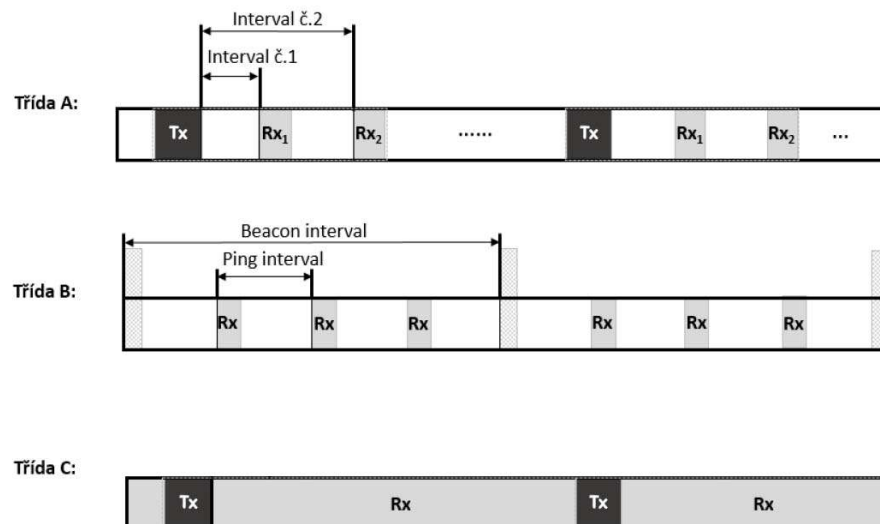
Rámec začíná preambulí o obvyklé délce 12,25 symbolů. Následuje volitelná fyzická hlavička (20 bitů), která obsahuje informace o délce uživatelských dat, kódovacím poměru a zda má být použito CRC uživatelských dat. Tato hlavička je součástí rámce pouze v tzv. explicitním módu. V implicitním módu je z rámce vyřazena, avšak v tomto případě musí být tyto informace nastaveny fixně ve vysílacím i přijímacím zařízení. Nakonec jsou součástí rámce uživatelská data (1–255 B) a volitelných 16 bitů pro CRC uživatelských dat. [20][33]

- **Energetická náročnost**

Výhodou technologie LoRa je její nízká energetická náročnost. Ta je zajištěna kombinací několika faktorů, například nízkým výsílacím výkonem a možností jeho regulace během provozu nebo periodou, s jakou zařízení komunikuje. Tím je zajištěna i možnost dlouhodobé funkce při bateriovém napájení. [20][33][42]

Síť LoRa může být využívána pro širokou škálu aplikací, které mohou mít různé nároky na způsob komunikace a úsporu energie. Proto byly definovány 3 různé třídy zařízení. Zařízení třídy A má nejmenší spotřebu energie a je tedy vhodná pro případ, kdy není k dispozici trvalý přívod napájení. Komunikace je vždy zahájena ze strany koncového

zařízení na základě definované události nebo na základě definovaného časového intervalu. Po odeslání zprávy má zařízení definovány 2 intervaly trvající 2 s, během kterých je připraveno přijmout zprávu. Na základě přijatých zpráv jsou vyvolány konkrétní akce (např. zapnuto/vypnuto nebo změna periody odesílání zpráv). Zařízení třídy B dokáže přijímat zprávy s využitím předem definovaných časových oken. Rozdíl mezi třídami A a B je, že zařízení třídy B v pravidelném plánovaném intervalu aktivují svou radiovou část a očekávají příchozí zprávu, avšak k odeslání zprávy nemusí dojít. Tato časová okna jsou synchronizována pomocí krátkých synchronizačních rámců tzv. beacon frames. V důsledku toho mají zařízení třídy B vyšší spotřebu elektrické energie. Zařízení třídy C je vhodná pro aplikace, kde není nutné energetickou náročnost řešit. Přijímací část je u těchto zařízení trvale aktivována. Příchozí zpráva tak může být ze sítě přijata v libovolném okamžiku. Příjem je přerušeno pouze, pokud zařízení vysílá. [20][33][42]



Obr. 3.8: Aktivita koncových zařízení LoRa

3.3 Wi-Fi

Wi-Fi je jedna z nejrychlejších bezdrátových technologií. Ve standardu IEEE 802.11ac je možné přenášet data rychlostí přes 1 Gb/s. Ve většině případů komunikují zařízení Wi-Fi ve frekvenčních pásmech v okolí 2,4 GHz a 5 GHz. Nevýhodou technologie Wi-Fi je její krátký dosah. Ten se při přímé viditelnosti pohybuje do maximálně několika set metrů, avšak v zástavbě a zarušeném prostoru (např. sousedními Wi-Fi sítěmi) klesá až na jednotky metrů. Důvodem pro kratší dosah je použití vyšších frekvencí, které hůře prostupují překážkami a zároveň nejsou příliš odolné vůči rušení. Wi-Fi sítě jsou používány v případech, kdy je

nutné přenášet velké množství dat. Pro vyšší elektrickou spotřebu jsou tato zařízení ve většině případů napájena ze sítě. [42][50][51]

3.4 Bluetooth

Technologie Bluetooth od verze 4.0 (vydána v roce 2010) umožňuje tři různé režimy provozu zařízení. První dvě varianty jsou určeny pro aplikace s vyšší požadovanou datovou propustností, stabilitou a dosahem. Třetí režim, Bluetooth LE (BLE, z anglického „Low Energy“), je oproti předchozím energeticky úspornější. Bluetooth pracuje, podobně jako Wi-Fi, na frekvencích v okolí 2,4 GHz. Klasické Bluetooth má k dispozici 79 kanálů o šířce 1 MHz, zatímco BLE má 40 kanálů o šířce 2 MHz. Nevýhodou technologie Bluetooth je její krátký dosah maximálně několik málo desítek metrů, v domácnosti spíše jednotky metrů. Nejnovější verze Bluetooth 5.0 přináší navýšení přenosových rychlostí (až 2 Mb/s), navýšení dosahu a nižší energetickou náročnost. [32][42][52][53][54]

3.5 Narrow Band LTE

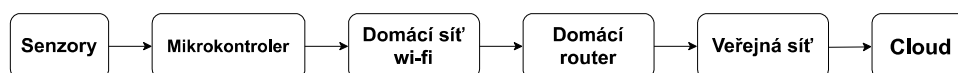
Narrow Band LTE (NB-LTE) patří k novým bezdrátovým úzkopásmovým technologiím LPWAN s nízkou energetickou náročností a širokým pokrytím. Jedná se o technologii primárně určenou pro využití v IoT aplikacích. V České republice tuto technologii poskytuje Vodafone a nově také T-Mobile. Na rozdíl od technologií LoRa a Sigfox pracuje Narrow Band v licencovaných frekvenčních pásmech LTE a využívá stávající základnové stanice mobilních operátorů. Využití stávající infrastruktury umožňuje rychlé a ekonomické nasazení této technologie. Narrow Band LTE se dělí do dvou kategorií: LTE Cat M1 a LTE Cat NB1. Kategorie LTE Cat M1 umožňuje přenosové rychlosti až 1 Mb/s a může pracovat v plném i polovičním duplexním režimu. Šířka pásma je u této kategorie 1,08 MHz a signál je umístěn uvnitř spektra LTE kanálu. Kategorie LTE Cat NB1 je orientována na nízkou spotřebu energie a umožňuje komunikaci pouze v polovičním duplexu. Maximální přenosové rychlosti jsou zde 250 Kb/s. Šířka pásma je 180 kHz, tudíž může být signál umístěn uvnitř spektra LTE kanálu nebo v ochranných pásmech spektra LTE. Zároveň lze vysílat i mimo spektrum LTE. [20][33][42][55][56][57]

3.6 Využití telekomunikačních technologií pro nositelnou elektroniku

Chytrá nositelná zařízení, jako například různé fitness náramky apod., využívají bezdrátových telekomunikačních technologií k přenosu informací a dat do uživatelem zvoleného cílového uložiště (např. mobilní telefon, osobní počítač či cloudové uložiště). Různá zařízení využívají různé technologie v závislosti na svých požadavcích a možnostech. V následujících odstavcích bude představeno několik vybraných možností, jak dostat data z nositelné elektroniky na cloudové uložiště a jaké vlastnosti či oblasti použití tyto způsoby mají.

Prvním z možných způsobů je využití sítě Wi-Fi. Síť Wi-Fi je velmi dobře známá a doma ji provozuje téměř každý. Zároveň je zdarma dostupná na mnoha veřejných místech, například v barech a restauracích. Oproti ostatním bezdrátovým komunikačním technologiím umožňuje velmi vysoké přenosové rychlosti, s čímž je ovšem spojena vyšší energetická náročnost. Dosah této sítě je omezený, v praxi na několik málo desítek metrů od přístupového bodu (routeru). Ve velkých objektech, např. školách nebo firemních budovách, se často Wi-Fi síť implementuje v tzv. mesh konfiguraci. To znamená, že více přístupových bodů (routerů) vytváří jednu větší Wi-Fi síť společně. Uživatel nositelné elektroniky využívající síť Wi-Fi je tak omezen na pohyb v relativně malé oblasti, resp. jedné budově. Využití Wi-Fi sítě je tedy zaměřeno spíše na domácí, pracovní a do jisté míry i nemocniční prostředí. Spolehlivost tohoto řešení ovlivňuje především spolehlivost poskytovatele internetového připojení a jak rychle je v případě poruchy schopen obnovit dostupnost svých služeb. Dalším faktorem mohou být také výpadky dodávky elektrické energie.

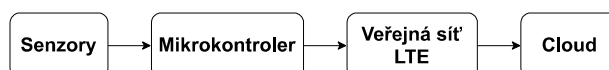
Kompletní cesta dat od senzorů až po cloud vypadá následovně:



Dalším ze způsobů přenosu dat je využití mobilních sítí. Výhodou mobilních sítí je jejich dostupnost a přenosová rychlost. Dle informací Českého telekomunikačního úřadu pokrývají všichni tři hlavní mobilní operátoři (Vodafone, T-Mobile a O2) mobilním internetem 4. generace (LTE) více jak 99,7 % obyvatel a více jak 95 % území České republiky. Nevýhodou mobilních sítí je jejich vysoká energetická náročnost a vyšší pořizovací cena

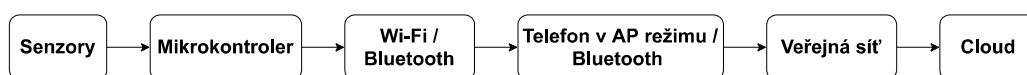
zařízení schopného se k mobilní síti připojit. Proti těmto nevýhodám se staví technologie Narrow Band LTE vycházející z klasické LTE technologie. Tato technologie míří na chytrá zařízení a využití v internetu věcí. Její energetická náročnost je výrazně menší a dosah pokrytí vyšší. To je vykoupeno přenosovou rychlostí, která dosahuje maximálně 1 Mb/s, avšak i to je pro přenos dat z inteligentních zařízení v naprosté většině případů dostačující. Vodafone jako jediný uvádí pokrytí České republiky technologií NB-LTE, a to 100 %. Díky své dostupnosti tak nachází technologie LTE uplatnění v široké škále zařízení.

Cesta dat od senzorů až po cloud v tomto případě vypadá takto:



Další dva způsoby do sebe spojují způsoby předchozí. Oba využívají mobilní telefon jako prostředníka mezi chytrým zařízením a internetem. V prvním případě je mobilní telefon nastaven jako přístupový bod a komunikuje s inteligentním zařízením prostřednictvím sítě Wi-Fi. Takto získaná data následně přeposílá do internetu skrze své mobilní (nejčastěji LTE) připojení. Druhý způsob je velmi podobný, ale ke komunikaci mezi chytrým zařízením a mobilním telefonem se využívá technologie Bluetooth. Ačkoli obě technologie pracují většinou ve stejném frekvenčním pásmu okolo 2,4 GHz, Bluetooth má nižší nároky než Wi-Fi. Bluetooth je energeticky výrazně úspornější, dosahuje nižších přenosových rychlostí a má kratší dosah. Zatímco v síti Wi-Fi obslouží jeden přístupový bod obvykle jednu celou domácnost, v síti Bluetooth je to spíše jedna místnost. I přes nevýhodu svého dosahu je Bluetooth využíváno ve většině chytrých fitness náramků, bezdrátových sluchátek a dalších podobných chytrých doplňků pro sportovní či rekreační účely. Ve všech těchto případech totiž není za potřebí přenášet větší objemy dat a zároveň se při všech těchto aktivitách počítá s tím, že má uživatel svůj mobilní telefon přímo u sebe v kapse či batohu. Uživatel si tak vytváří vlastní osobní síť a díky tomu není omezen pohybem v relativně malé oblasti jako u domácí Wi-Fi sítě. Zároveň může naměřená data sledovat v reálném čase v aplikaci na telefonu.

Následující blokové schéma zobrazuje cestu dat mezi senzory a cloudem v případě využití mobilního telefonu jako přístupového bodu.



Poslední dvě vybrané možnosti jsou sítě vyvíjené za účelem velkých dosahů a velmi nízké energetické náročnosti. Síť LoRa a Sigfox mohou dosahovat až několika desítek kilometrů od své základnové stanice, ale jejich přenosové rychlosti se pohybují spíše ve stovkách bitů či jednotkách kilobitů za sekundu. I takto pomalé datové přenosy jsou v některých případech dostačující, avšak tyto sítě neposkytují možnost kontinuální komunikace. Data jsou odesílána v jednotlivých krátkých zprávách, ve kterých je pro uživatelská data velmi málo prostoru. V síti Sigfox je to 12 bajtů a v síti LoRa od 55 do 240 bajtů. Zároveň jsou obě tyto sítě provozovány v bezlicenčním frekvenčním pásmu v okolí 868 MHz a vztahují se na ně různá legislativní opatření. Tím nejvýznamnějším je nařízení omezující aktivní dobu zařízení v síti na 1 % času z daného časového úseku. Je-li tento časový úsek např. jedna hodina, může být radiová část zařízení aktivní pouze 36 sekund. Vzhledem k těmto vlastnostem a omezením jsou sítě Sigfox a LoRa použitelné spíše pro hlášení nouzových stavů nebo pro sledování velmi pomalu měnících se veličin, pro sledování životních funkcí jsou nevhodné.

Cesta dat od senzorů po cloudové uložení vypadá v tomto případě takto:

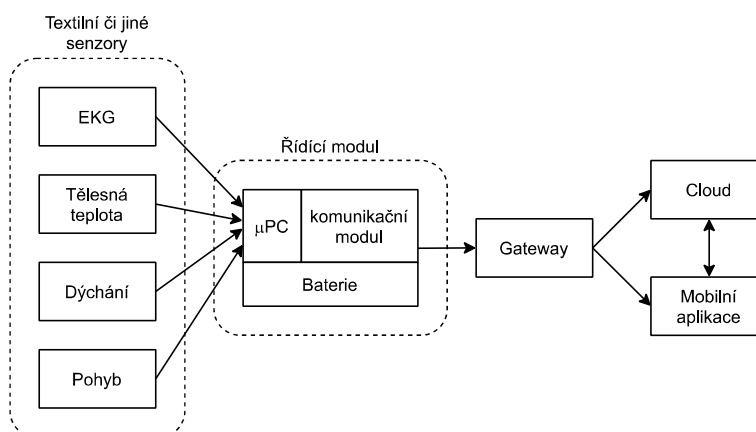


4 Návrh systému, vývoj softwaru a realizace prototypu

Vlastní práce se zabývá návrhem a realizací vhodného prototypu systému pro inteligentní podprsenku. Ať už se jedná o podprsenku pro ženy, nebo tílko či upnuté triko pro muže, je oblast hrudníku tou nejvhodnější pro umístění senzorů snímajících životně důležité funkce, jako je srdeční aktivita a dýchání. Nejvhodnější je především pro to, že je tato poloha co možná nejbližší k srdci a zároveň obepíná celý hrudník. Umístěním senzoru pohybu v této oblasti je navíc možné získávat informace o držení těla a pohybu člověka – například zda stojí či leží, běží či jde, anebo detekovat nárazy a pády.

Inteligentní podprsenka má za úkol monitorovat životní funkce člověka. K tomu jsou zapotřebí především senzory dýchání a srdeční aktivity. Další důležitou informací o zdravotním stavu člověka je jeho tělesná teplota, k čemuž je nutný senzor teploty. Pro monitorování pohybu a případnou detekci pádu lze použít tříosý akcelerometr. Data z těchto senzorů budou snímána pomocí mikrokontroleru, který je bude následně bezdrátově přenášet na cloudové uložení či do mobilní aplikace. Zvolený hardware a následný vývoj řídicího programu a obsluha cloudových služeb je popsána v následujících kapitolách.

Vzhledem k tomu, že jde o nositelnou elektroniku, neměly by tyto senzory nijak znepříjemňovat její používání. Výsledný výrobek by měl být komfortní a uživatele nijak neomezovat. Dosažení tohoto cíle by bylo s využitím konvenčních senzorů prakticky nemožné, proto by měly být implementovány textilní elektrody a senzory pro monitorování daných veličin. Blokové schéma návrhu systému inteligentní podprsenky je znázorněno na následujícím obrázku.



Obr. 4.1: Blokové schéma systému pro nositelnou elektroniku

4.1 Datová náročnost a úspora objemu přenášených dat

Tato kapitola se bude zabývat tím, jaké množství dat bude zapotřebí k přenosu daných veličin.

Pomalou měnící se veličiny není nutné snímat a přenášet velmi často. To se týká například napětí baterie, které stačí aktualizovat jednou za minutu. Zvolený hardware (viz. dále) implicitně pracuje s čísly ve formátu s plovoucí desetinnou čárkou (float), jehož velikost je 4 bajty. Vzhledem k tomu, že důležitou součástí informace o stavu baterie jsou i desetiny či setiny voltu, je formát float vyhovující. S využitím sítě Wi-Fi pro přenos jsou 4 bajty dat relativně zanedbatelné, jelikož Wi-Fi sítě samotné jsou dimenzovány a rychlé přenosy velkých objemů. Problém však nastává v případě, kdy budeme chtít využít sítě Sigfox nebo LoRa, které mají značně omezenou délku jedné zprávy. Vhodným algoritmem upravíme formát dat tak, abychom zmenšili potřebný objem dat pro přenášenou informaci.

Napětí baterie budeme v cloudu požadovat s přesností na dvě desetinná místa. Abychom nemuseli přenášet formát s desetinnou čárkou (float), roznásobíme naměřenou hodnotu 100krát a zaokrouhlíme na celá čísla. Zároveň víme, že při plném nabití baterie se její napětí bude pohybovat v okolí 4,2 voltů a že pokud klesne pod 3,3 voltů, zařízení se vypne anebo vůbec nespustí. Sledovaný rozsah napětí je po roznásobení 100krát a zaokrouhlení roven hodnotám od 330 do 420. Čísla v tomto rozsahu je již možné přenášet v 2bajtovém formátu short (do 2 bajtů formátu short se vejdu hodnoty od 0 do 65535). My si ale ještě pomůžeme tím, že budeme naměřenou hodnotu odesílat pouze jako odchylku od pevně dané známe hodnoty (v tomto případě od hodnoty 3,0 V). Pokud tedy od již získaného rozsahu 330 až 420 odečteme 300, získáme nový rozsah 30 až 120. Tento nový rozsah se již pohodlně vejde do rozsahu hodnot 1bajtového formátu char (0-255), a takto jej budeme přenášet skrze sítě Sigfox a LoRa. Skutečnou hodnotu napětí baterie v cloudu následně získáme aplikací opačného postupu – k doručené hodnotě přičteme 300 a výsledek vydělíme 100krát.

Další relativně pomalu měnící se veličinou je tělesná teplota. Ta se za normálního zdravotního stavu člověka pohybuje okolo 37 °C. V klidovém stavu, například během spánku, může klesnout až ke 36 °C, zatímco v zátěži může vystoupat až k 39 °C. Naopak při onemocnění může tělesná teplota vystoupat na 42 až 43 °C. Tyto změny však nejsou náhlé, tělesná teplota během zátěže může stoupat přibližně o 1 °C za hodinu. Z toho plyne, že pro

snímání tělesné teploty také postačí interval jedné minuty. Pro tělesnou teplotu jsou opět důležité i desetiny stupně Celsia, k čemuž je taktéž vhodný 4bajtový formát float. Pro úsporu dat můžeme využít obdobný způsob jako u napětí baterie. Omezíme-li se sledovaným rozsahem teplot na běžné tělesné teploty, a nikoliv celý rozsah teplotního senzoru, získáme po roznásobení 10krát rozsah hodnot ~ 360 - 430 . Danou hodnotu opět můžeme přenášet jako odchylku od známé hodnoty a tudíž se i s hodnotou tělesné teploty po úpravě vejde do rozsahu jednoho bajtu.

Hodnotu frekvence dýchání nelze odečítat jako okamžitou hodnotu tak, jak tomu je u tělesné teploty nebo napětí baterie. V tomto případě je nutné nasbírat data za daný časový úsek a následně z těchto dat vypočítat odpovídající frekvenci dýchání. Například je-li frekvence dýchání snímána piezorezistivním senzorem, je pak pohyb hrudníku způsobený dýcháním převáděn na změny elektrického odporu senzoru. V časovém průběhu změn odporu pak lze detekovat špičky a spočítat kolik se jich vyskytuje v daném časovém úseku. Tento výpočet je možné provést v řídicím modulu zařízení a na cloud či do telefonu přenášet pouze výslednou hodnotu. Běžná klidová dechová frekvence zdravého jedince se pohybuje mezi 8 až 16 nádechy za minutu, v zátěži může dosahovat až 60 nádechů za minutu. Pro frekvenci dýchání postačí celočíselná hodnota. V tomto případě stačí pro přenos informace 1 bajt s periodou přenosu 2 sekundy.

EKG je rychle se měnící veličina, která poskytuje lékařům rozsáhlé informace o funkci srdce pacienta. Jedná se o poměrně komplexní křivku, jejíž časový průběh se dělí do několika intervalů. Lékaři posuzují funkci srdce pacienta na základě tvaru a změn křivky v porovnání s křivkou EKG zdravého člověka. Posuzují se různé odchylky od normálu či různé variace srdečního rytmu. K tomu, aby křivka EKG obsahovala dostatečné množství informací potřebných ke správné diagnóze, je nutné odečítat veličiny z elektrod s určitou vzorkovací frekvencí. Většina nemocničních zařízení vzorkuje EKG signály s frekvencí 1 000 Hz a vyšší. Z různých studií (např. [81][82][83]) vyplývá, že i signál vzorkovaný s frekvencí 500 či 250 Hz obsahuje stále velké množství užitečných informací pro analýzu variací srdečního rytmu. Signál vzorkovaný s frekvencí 100 Hz stále obsahuje užitečné informace, a dokonce i ze signálu vzorkovaného s frekvencí 50 Hz je zřejmý QRS komplex křivky EKG. S takto nízkou vzorkovací frekvencí již není možné analyzovat drobné variace srdečního rytmu, avšak například pro výpočet průměrného počtu úderů srdce za jednu minutu postačí. Při vzorkování 16bitovým převodníkem pak získáme 2bajtová data na každý

vzorek signálu. S vzorkovací frekvencí 250 Hz by pak byla datová náročnost 500 B/s. I zde je možné uspořít. Úrovně napětí signálu EKG se pohybují v rozmezí přibližně -0,3 až 1,7 mV. Uvážíme-li nějakou rezervu, můžeme počítat například s rozsahem -1 až 3 mV. Abychom nemuseli řešit přenos záporných čísel, přičteme k naměřené hodnotě 1 mV, čímž se dostaneme na rozsah 0 až 4 mV. Zatímco pro napětí baterie jsou jednoty mV zanedbatelné, v případě měření EKG se signál běžně zaznamenává s přesností na jednotky μV . Po roznásobení 1000krát dostaneme číselný rozsah 0 až 4000. Dále číslo z tohoto rozsahu vydělíme 16krát a zaokrouhlíme na celá čísla. Tím se dostaneme na rozsah čísel 0 až 250 a díky tomu se opět vejde do jednoho bajtu dat. Touto úpravou zanášíme do informace chybu, která po zpětném výpočtu naměřené hodnoty napětí může dosahovat až 8 μV , avšak tato chyba není příliš významná a rekonstruovaný průběh signálu je možné nadále spolehlivě analyzovat. S touto úpravou se tedy datová náročnost sníží na 1 bajt na vzorek, čemuž odpovídá 250 B/s při vzorkovací frekvenci 250 Hz. Další možné zásadní snížení datové náročnosti je, pokud nás bude zajímat pouze frekvence úderů srdce a nepotřebujeme znát detailní průběh křivky EKG. Frekvenci tlukotu srdce můžeme z navzorkovaného signálu EKG vypočítat a následně ji přenášet jako celočíselnou hodnotu. Klidová frekvence srdečního tepu je u zdravých lidí přibližně 60-100 úderů za minutu. V zátěži nebo při srdečních onemocněních se pak tato hodnota může vyšplhat až na hodnoty přesahující 200 úderů za minutu. Pro tyto hodnoty opět stačí číselný rozsah jednoho bajtu a aktualizovat ji můžeme každé 2 sekundy, z čehož pak plyne datová náročnost 1 B/s.

Stanovení optimální vzorkovací frekvence akcelerometru není zcela jednoduchou záležitostí, jelikož je závislá na typu pohybu. V případě, že se jedinec nepohybuje, ale nehybně stojí, sedí či leží, postačuje vzorkovací frekvence v řádu jednotek vzorků za sekundu. Mají-li být data akcelerometru použita pro rekognici fyzické aktivity člověka při sportu nebo například při detekci pádu, pak se může optimální vzorkovací frekvence vyšplhat na desítky až několik set vzorků za sekundu. V mobilních telefonech či chytrých hodinkách, fitness náramcích apod. se běžně používá vzorkovací frekvence 100 Hz. Některé studie (např. [84][85]) uvádí jako postačující frekvenci i 50 Hz a některé představují pro optimální řešení měnit vzorkovací frekvenci v závislosti na fyzické aktivitě uživatele, k čemuž využívají různé algoritmy vyhodnocení. Akcelerometr zvoleného hardware dokáže dle dokumentace zaznamenat zrychlení v nastavitelných rozsazích ± 2 g, ± 4 g nebo ± 8 g. V případě nastavení na nejvyšší rozsah a při požadavku přesnosti na dvě desetinná místa bude úprava pro úsporu dat vypadat následovně – roznásobení 100krát, zaokrouhlení pro

získání celočíselné hodnoty a přičtení 800 pro zbavení se záporných čísel. Takto se dostaneme na hodnoty v rozmezí 0 až 1600 a tudíž se vejde do 2 bajtů dat na jeden vzorek jedné osy. Pokud uvážíme vzorkovací frekvenci 100 Hz dostáváme datovou náročnost 200 B/s na osu. Celkově tedy 600 B/s na tři osy.

Datová náročnost pro uvedené veličiny je tedy jednoznačně nejvíce závislá na vzorkovací frekvenci senzoru EKG a akcelerometru. Oproti tomu datová náročnost pro přenos informace o frekvenci dýchání, tělesné teplotě a stavu baterie je vzhledem k datové náročnosti EKG a akcelerometru téměř zanedbatelná. Maximální celková datová náročnost pro uvedených pět sledovaných veličin je tedy $\sim 0,85$ kB/s.

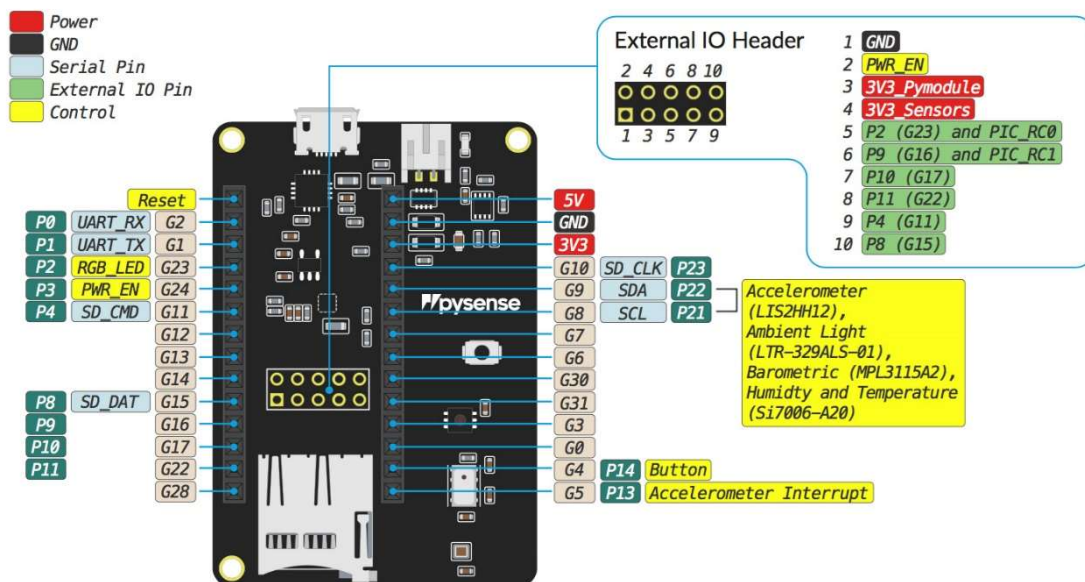
4.2 Zvolené telekomunikační technologie

Námětem této práce je vývoj komunikačního subsystému pro nositelnou elektroniku, který bude schopen číst data ze senzorů a předávat je k dalšímu zpracování. Z průzkumu trhu vyplývá, že většina takových zařízení využívá k přenosu dat technologie Bluetooth, v menší míře pak Wi-Fi. Tato zařízení dále využívají mobilních telefonů a jejich aplikací ke zpracování a vizualizaci naměřených dat. Zároveň je také využívají jako brány (gateway) do veřejných sítí a pro sdílení dat v cloudových uložiscích a službách. Výhodou těchto řešení je možnost kontinuálního či velmi častého přenosu i relativně velkých objemů dat mezi monitorovacím a vyhodnocovacím zařízením, což je pro sledování životních funkcí člověka zvláště žádoucí. Oproti tomu sítě jako LoRa či Sigfox jsou využívány spíše pro méně náročné aplikace s menším objemem dat a v případech, kdy je zapotřebí sledované veličiny kontrolovat pouze několikrát za den. Jejich použití v nositelné elektronice by se dalo obhájit jako záloha v případě nutnosti hlášení kritických stavů při ztrátě spojení s mobilním telefonem. Vzhledem k tomuto je v této práci přednostně využíváno technologií Wi-Fi.

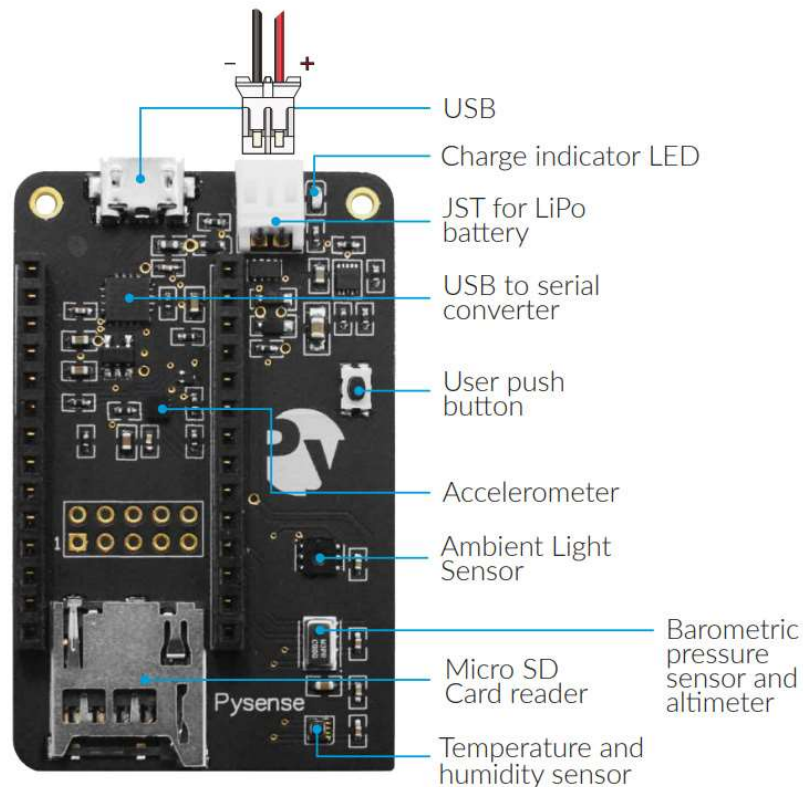
4.3 Hardware

V této kapitole je popsán použitý hardware. Konkrétně se jedná o sadu modulu LoPy4 s vývojovou deskou Pysense od firmy Pycom. Pycom je relativně nový a neznámý výrobce, a proto nebylo jednoduché některé informace získat. K tomu nepřispěla ani často ne zcela dostatečně dostupná dokumentace. Zaměstnanci zákaznické podpory a lidé z různých veřejných fór však byli většinou ochotni a nápomocni.

Vývojová deska Pysense, neboli sensorový modul, poskytuje řadu enviromentálních senzorů a dalších užitečných funkcí. První integrovaný obvod umožňuje měření teploty s rozlišením 12 bitů v kombinaci s měřením barometrického tlaku s rozlišením 20 bitů, další integrovaný obvod kombinuje senzor vlhkosti s druhým senzorem teploty, oba s rozlišením 12 bitů. Třetím obvodem je tříosý 16bitový akcelerometr a posledním je obvod kombinující dva 16bitové senzory osvětlení, každý s jinou charakteristikou citlivosti v závislosti na vlnové délce dopadajícího světla. Dále lze využít USB rozhraní s přístupem přes sériový port, nabíječky lithiových baterií a kompatibility s paměťovými kartami microSD. Velmi důležitou funkcí desky Pysense je také režim hlubokého spánku se spotřebou přibližně 8 μ A. Deska má rozměry 55 x 35 x 10 mm, dokáže operovat v rozsahu teplot do -40 do +85 $^{\circ}$ C a váží 11 gramů.

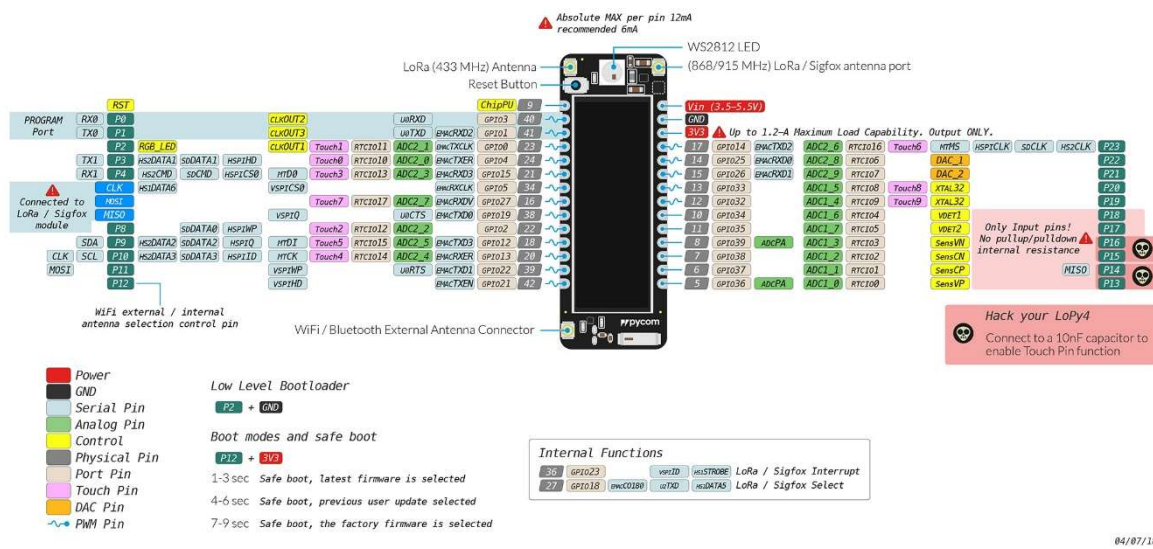


Obr. 4.2: Pinout diagram desky Pysense (převzato z [59])



Obr. 4.3: Umístění jednotlivých součástí desky Pysense (převzato z [59])

Modul LoPy4 poskytuje podporu pro síť Wi-Fi, Bluetooth, Sigfox a LoRa. Obsahuje výkonný dvoujádrový mikroprocesor ESP32 od firmy Espressif, který je taktovaný až na 240 MHz. Tento mikroprocesor je doplněn 4 MB RAM a 8 MB externí flash paměti. Dále obsahuje UART, SPI, I2C, I2S rozhraní a osm 12bitových analogově digitálních převodníků. Modul má implementovaný systém MicroPython. Díky tomu je tento modul vhodný pro vývoj aplikací v době testování a četných úprav kódu programu. MicroPython si lze představit jako obdobu operačního systému pro klasické počítače, ovládá všechny hardwarové součásti (přídavné moduly, senzory, vstupy a výstupy) a navíc rozumí jazyku Python. Upravené části kódu nebo příkazy můžeme vkládat do příkazového řádku konzole a jsou vykonány okamžitě. Tím odpadá nutnost opakované kompilace a nahrávání upraveného programu do paměti tak, jako je to u jiných mikrokontrolerů (např. známé Arduino).



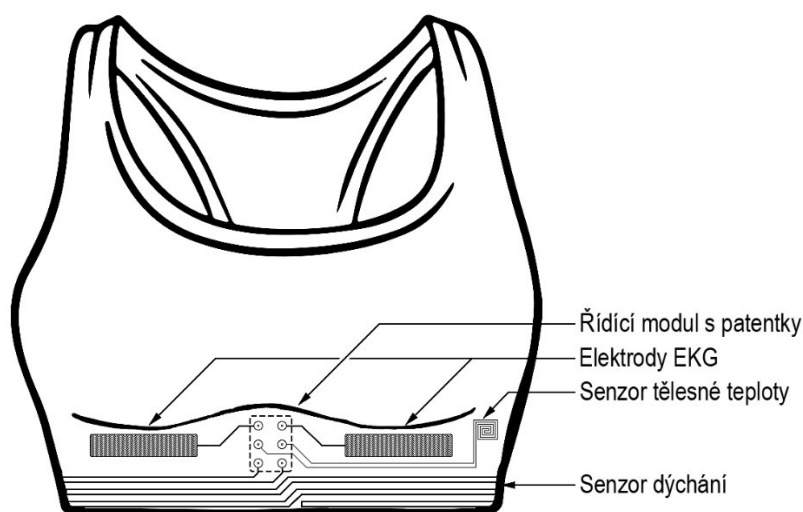
Obr. 4.4: Pinout diagram modulu LoPy4 (převzato z [59])

Při využívání sítě Wi-Fi používá zařízení automaticky svou vlastní anténu, která je integrována na desce modulu LoPy4. Modul také disponuje konektorem pro připojení externí Wi-Fi antény. Pak je možné prostřednictvím příkazu v kódu volit, kterou z Wi-Fi antén bude zařízení využívat. V případě záměru využití sítí Sigfox nebo LoRa je však nutné připojit externí anténu, jelikož modul nemá pro tyto sítě anténu integrovanou. V případě používání zařízení bez připojené antény riskujeme jeho nevratné poškození.

- **Textilní senzory**

Jak již bylo uvedeno, pro monitorování tělesné teploty, dechové frekvence a signálu EKG je vhodné ve smart textilních systémech použít textilní senzory. Elektrody EKG jsou tvořeny jednoduchými souvislými vodivými plochami, přičemž signál EKG je tvořen rozdílem potenciálů mezi dvěma elektrodami. Pro zajištění kvalitního přenosu signálu musí být tyto elektrody velmi dobře elektricky vodivé a s dobrým těsným kontaktem s pokožkou. Plocha elektrody může být tvořena hustým prošitím textilie vodivou nití. Senzor dýchání je tvořen obvinutím hrudníku vodivou nití, která má naopak velkou rezistivitu. Pracuje na piezorezistivním principu podobně jako tenzometry, čili při natažení vodivé nitě v důsledku roztažení hrudníku se elektrický odpor této nitě zvýší a naopak. Změny elektrického odporu tak přímo kopírují dýchání jedince, přičemž větší rezistivita (v řádech stovek či tisíců Ohmů) nitě je žádoucí, jelikož se lépe měří než velmi malé hodnoty (např. desetiny Ohmů). Senzor teploty je tvořen vodivou nití, jejíž elektrický odpor je závislý na

teplotě nitě, resp. teplotě těla. Tělesná teplota tak přímo ovlivňuje elektrický odpor senzoru, který je snímán podobně jako u senzoru dýchání.



Obr. 4.5: Umístění senzorů a řídicího modulu na podprsence

Schopnost senzorů poskytnout kvalitní signály závisí na jejich umístění. Elektrody EKG jsou umístěny co nejbližší k srdci. Zároveň i v místě, kde pružná textilie zajistí těsný kontakt s pokožkou a kde napomůže redukci pohybových artefaktů. Senzor frekvence dýchání je umístěn v oblasti hrudníku v místě, které je vlivem dýchání nejvíce rozpínáno a smršťováno. To zapříčiní větší rozdíly mezi maximálními a minimálními hodnotami odporu senzoru, a tudíž i přesnější a jednodušší analýzu. Senzor teploty je také umístěn v místě, kde je zajištěn těsný kontakt s pokožkou. Zároveň je v místě, kde se člověk v porovnání s oblastí prsou či zad méně potí, což by mohlo nepříznivě ovlivnit naměřené hodnoty teploty.

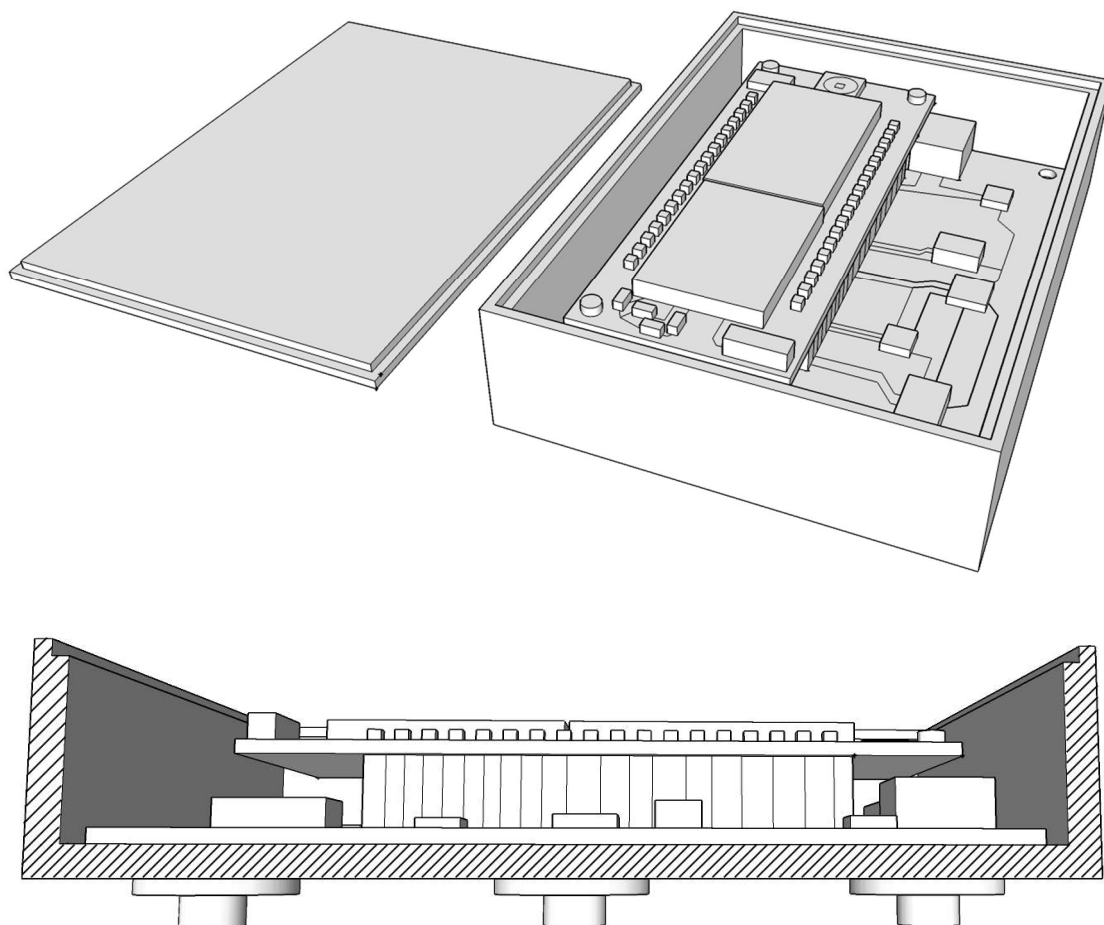
- **Propojení textilních senzorů a řídicího modulu**

Vhodnou volbou pro propojení textilních senzorů s řídicím modulem se jeví klasické patentky. Ty umožňují nejen elektrické, ale i mechanické spojení řídicího modulu s textilií. Výhodou tohoto řešení je možnost řídicí modul z textilie jednoduše sejmout v případě nutnosti čištění či praní a zároveň umožňuje využití jednoho řídicího modulu pro více chytrých textilií.



Obr. 4.6: Klasická patentka na textil

Výhodou je i materiál, ze kterého se patentky běžně vyrábějí. Na první pohled by se mohlo zdát, že se jedná o nerezovou ocel, avšak je to klasický lisovaný železný plech, který je poniklován. To ale v tomto případě není na škodu, jelikož díky poniklování lze patentku pájet. Jedna část patentky tedy může být přišita vodivými nitěmi k textilií, zatímco druhá část může být připájena k vodiči a přilepena ke dnu plastové krabičky, ve které je umístěn řídicí modul, a připájeným drátkem vedeným skrze otvor ve dnu se zajistí přenos signálu ze sensorů až k řídicímu modulu.



Obr. 4.7: Návrh prototypu řídicího modulu s krabičkou a patentky

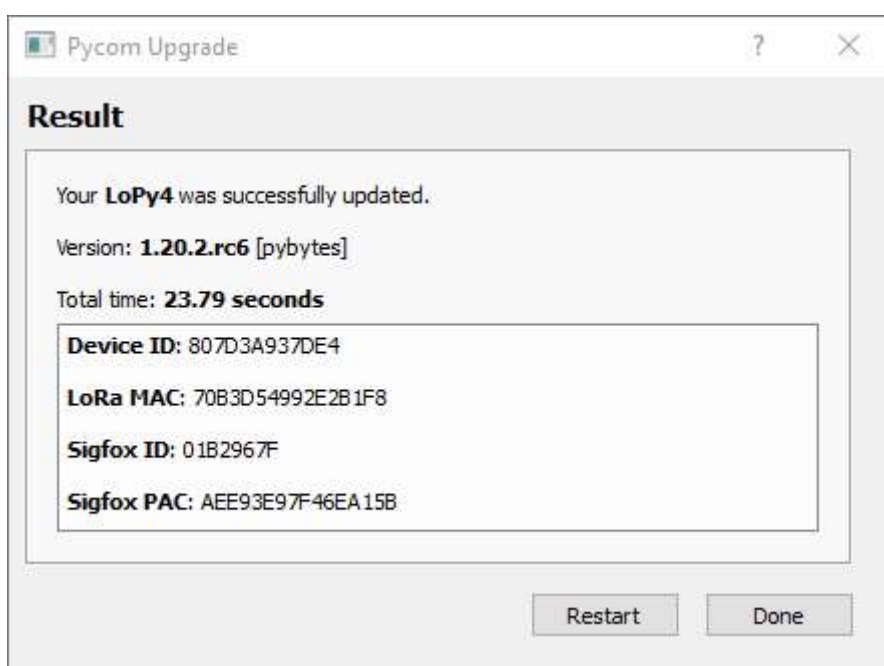
4.4 Softwarové prostředky

V této kapitole je popsán software potřebný k ovládní modulu a vytvoření aplikací. Pro vývoj aplikací byla použita aplikace Visual Studio Code (dále jen VSC), což je ve své podstatě jednoduchý textový editor vytvořený k účelu programování. Aby se z tohoto textového editoru stalo plnohodnotné vývojové prostředí, je nutné doinstalovat plugin Pymacr pro práci ve skriptovacím jazyku Python. Aplikace VSC je volně dostupná ke stažení pro operační systémy Windows, macOS a Linux. Přídavné pluginy jsou taktéž zdarma. Přidání pluginu do aplikace se provádí uvnitř programu kliknutím na záložku Extensions, vyhledáním požadovaného pluginu a jeho instalací.

Ještě před samotným používáním modulů Pysense a LoPy4 bylo nutné provést aktualizaci jejich firmware. Tyto aktualizace jsou poněkud složitější a v dostupné oficiální dokumentaci jsou popsány nedostatečně. Následující postupy jsou již ověřené a fungují spolehlivě.

Jako první se musí provést aktualizace firmware desky Pysense. K tomu budeme potřebovat programy Zadig, DFU-util (oba ke stažení na docs.pycom.io/pytrackpysense/installation/firmware/) a soubor firmwaru desky (aktuálně nejnovější je `pysense_0.0.8.dfu`). Otevřeme program Zadig, v záložce Options vybereme „List all devices“ a k instalaci vybereme "libusbk (v3.0.7.0)". Poté stiskneme a držíme tlačítko na desce Pysense, a následně ji, zatím samostatně bez vsunutého modulu LoPy4, připojíme k počítači. V seznamu zařízení se objeví "Unknown Device #1", který vybereme a klikneme na Install Driver (*příloha A, obr. 1*). Teprve poté můžeme tlačítko na desce Pysense uvolnit. Po krátké prodlevě se zobrazí zpráva o úspěšné aktualizaci. Pak desku Pysense odpojíme. Dále rozbalíme stažený zip soubor DFU-util.zip a do stejné složky si zkopírujeme i soubor firmwaru desky `pysense_0.0.8.dfu`. Poté z této složky otevřeme příkazový řádek a připravíme si příkaz "`dfu-util-static.exe -D pysense_0.0.8.dfu`". Dále opět stiskneme a držíme tlačítko na Pysense desce, připojíme ji k USB a spustíme příkaz v příkazovém řádku. O úspěšné aktualizaci firmware nás ujistí výpis záznamu aktualizace v okně příkazového řádku (*příloha A, obr. 2*). Nakonec je možné v programu Zadig zkontrolovat úspěšnost aktualizace tak, že vybereme ze seznamu zařízení Pysense a v kolonce Driver by mělo být „usbser“. Zároveň se nám ve Správci zařízení počítače již nezobrazí Unknown device #1 ale Sériové zařízení USB (*příloha A, obr. 3*).

Aktualizace modulu LoPy4 je již podstatně jednodušší. K té budeme potřebovat aktualizací program dostupný na <https://pycom.io/downloads/>. Modul LoPy4 zasuneme to desky Pysense tak, aby RGB led dioda byla nad konektorem USB, a následně celý komplet připojíme k počítači. V počítači nainstalujeme a spustíme aktualizací program, který nás provede aktualizací. V rámci aktualizace nás program vyzývá k registraci Pybytes. Tento krok je možné přeskočit, jelikož zařízení lze registrovat později příkazem skrze konzoli programu VSC anebo opětovným spuštěním tohoto aktualizací programu. V posledním okně se dovídáme důležité kódy potřebné k registraci do sítě Sigfox (konkrétně Sigfox ID a Sigfox PAC). Tyto kódy si poznamenejme.



Obr. 4.8: Poslední okno aktualizací programu modulu LoPy4

Posledním krokem před možností práce s moduly je prvotní nastavení Pymakr pluginu v programu VSC. Na spodní liště okna VSC klikneme All commands a vybereme Pymakr > Extra > List Serial Ports. Tento příkaz nám vypíše dostupné sériové porty a automaticky detekuje port, ke kterému máme připojené naše zařízení. Tento port si zkopírujeme, v All commands vybereme Pymakr > Global Settings, a zkopírovaný port vložíme do kolonky „adress“. Poté Soubor nastavení uložíme a zavřeme. Klikneme znovu na All commands a vybereme Pymakr > Connect. V konzoli se objeví tři šipky (>>>) indikující úspěšné připojení.

4.4.1 Program obsluhy senzorů a řízení bezdrátové komunikace

Tento program je nahrán v paměti mikrokontroleru a řídí veškerou jeho činnost. Program byl napsán v jazyce Python s využitím knihoven vestavěných v MicroPythonu a knihoven od firmy Pycom. Část programu zabývající se získáním dat ze senzorů je pro všechny vytvořené aplikace stejná. Část programu řídící bezdrátovou komunikaci je pak pro každou použitou síť jiná. Kompletní vlastní program modulu je přiložen k práci v příloze B.

4.4.2 Sigfox backend

Jelikož platforma Sigfoxu není open-source, je třeba zařízení registrovat. Využívání sítě Sigfox není zdarma. V České republice ji provozuje společnost SimpleCell Networks, a.s. V tomto případě ale není zakoupení předplatného pro přístup do sítě nutné. Společnosti Pycom a Sigfox spolu spolupracují a se zakoupením modulu firmy Pycom získáme i roční předplatné. K registraci zařízení do sítě tedy postačí vyplnění formuláře na stránce buy.sigfox.com/activate.

Zde vybereme Českou republiku a klikneme „Next“. Dále vyplníme klíče Sigfox ID a Sigfox PAC získané pomocí aktualizčního programu. V dalších oknech vytvoříme vlastní účet a potvrdíme jej konfirmačním emailem. Poté se již můžeme přihlásit do Sigfox backendu na stránce backend.sigfox.com.

Jako první musíme v Sigfox backendu nastavit parametr „Keep-alive“. Jeho výchozí hodnota je nastavena na 0 a to znamená vypnutí komunikace. Přejdeme tedy do záložky „Device Type“ a klikneme na „DevKit 1 (PYCOM)“. Z rozbaleného menu vybereme „Edit“ a na následující stránce nastavíme Keep-alive parametr na 30.

První vytvořenou aplikací je poslání zprávy „Hello Sigfox“ na Sigfox backend. V novém okně programu VSC vytvoříme novou složku a v ní soubor pojmenovaný `main.py`, který následně otevřeme pro editaci a vložíme do něj následující kód:


```

from network import WLAN
import socket

# iniciace Sigfoxu pro region RCZ1 (Evropa)
sigfox = Sigfox(mode = Sigfox.SIGFOX, rcz = Sigfox.RCZ1)



# vytvoření Sigfox socketu a jeho nastavení do blokujícího stavu
s = socket.socket(socket.AF_SIGFOX, socket.SOCK_RAW)
s.setblocking(True)

# nastavení socketu pouze pro odesílání
s.setsockopt(socket.SOL_SIGFOX, socket.SO_RX, False)

# odeslání zprávy
s.send('Hello Sigfox')




```

Program nahrajeme na desku kliknutím na tlačítko „Upload“ na spodní liště. Po nahrání programu se deska automaticky restartuje a program se spustí. K ověření doručení zprávy využijeme Sigfox backend, kde by se v záložce „Device“ měla změnit barva „Communication status“ na zelenou.

Communication status	Device type	Group	Id	Last seen	Name	Token state
	LoPy4	University of West Bohemia	1B2967F	2020-06-14 17:52:08	PYCOM_DevKit_1-device	

Obr. 4.9: Stav spojení zařízení se Sigfox backendem

Obsah přijaté zprávy zobrazíme kliknutím na ID zařízení a výběrem „Messages“ v levém menu. Zde uvidíme seznam všech přijatých zpráv včetně času přijetí a vlastního sekvenčního čísla zprávy.

Time	Seq Num	Data / Decoding	LQI	Callbacks	Location
2020-05-11 20:19:32	22	48656c6c6f205369676666f78			

Obr. 4.10: Ukázka přijaté zprávy

V kolonce data je uvedeno „48656c6c6f205369676666f78“. To je způsobeno tím, že Sigfox automaticky dekoduje zprávy jako hexadecimální a v tomto formátu s nimi dále pracuje. Správnost zprávy lze ověřit v jakémkoliv webovém konvertoru.

Hex to String (Hex to Text) ☆

Enter the hexadecimal text to decode

Convert

The decoded string:

Obr. 4.11: Ověření správnosti zprávy převedením zpět na text

V dalším kroku vytvoříme jednoduchý callback. Funkce callback umožňuje přijatá data přeposílat do jiných cloudových služeb nebo je posílat přímo na zvolenou emailovou adresu. V záložce „Device Type“ klikneme na jméno zařízení a v levém menu vybereme „Callbacks“. Vytvoříme nový „Custom Callback“, „Type“ nastavíme na DATA a UPLINK, „Channel“ zvolíme EMAIL, do pole Recipient vyplníme zvolenou emailovou adresu a do polí Subject a Message napíšeme předmět a text emailové zprávy. S následující přijatou zprávou se pak automaticky odešle email na zadanou adresu se zvoleným obsahem. Emailová zpráva může obsahovat i samotná data, časové razítko, sekvenční číslo zprávy a další.

4.4.3 LoRa backend

Síť LoRaWAN spravuje a rozvíjí v celosvětovém měřítku organizace LoRa Alliance. Členem této organizace je také společnost The Things Network (dále jen TTN), která poskytuje služby LoRa ve více než 150 zemích světa. Z map pokrytí dostupných na webových stránkách TTN je zřejmá koncentrace právě v Evropě, kde je v provozu přes 7 000 přístupových bran.

Pro využití sítě TTN je nejprve nutné se registrovat na jejich webových stránkách (thethingsnetwork.org) a vytvořit zde tzv. aplikaci. Po registraci a přihlášení klikneme na svůj profil a z rozbaleného menu vybereme záložku „Console“. V otevřeném okně vybereme „Applications“ a vytvoříme novou. „Application ID“ můžeme zvolit libovolně. Po vytvoření aplikace je nutné přidat naše zařízení. V právě vytvořené aplikaci přejdeme

do záložky „Devices“ a zde jej registrujeme. Pole „Device ID“ můžeme opět vyplnit libovolně, avšak „Device EUI“ je unikátní kód pro každé zařízení. Tento kód získáme spuštěním následujících příkazů na našem LoPy4 modulu:

```
from network import WLAN
import ubinascii
lora = LoRa()
print("DevEUI: %s" % (ubinascii.hexlify(lora.mac()).decode('ascii')))
```

Poskytnutý kód zkopírujeme a zařízení registrujeme. V „Device overview“ jsou pro nás důležité především kódy „Application EUI“ a „App Key“, které budeme potřebovat vložit do programu pro LoPy4 modul. První aplikace s využitím sítě LoRa bude opět posílání jednoduché zprávy „Hello TTN“. V okně VSC vytvoříme novou složku s novým main.py souborem, do kterého vložíme následující kód, který poté nahrajeme na desku:

```
from network import LoRa
import socket

# jednotlivé unikátní kódy pro naše zařízení a TTN aplikaci
dev_eui = ubinascii.unhexlify('70B3D54992E2B1F8')
app_eui = ubinascii.unhexlify('70B3D57ED002EC71')
app_key = ubinascii.unhexlify('B983A18BF614E9C270D789137FC2298A')

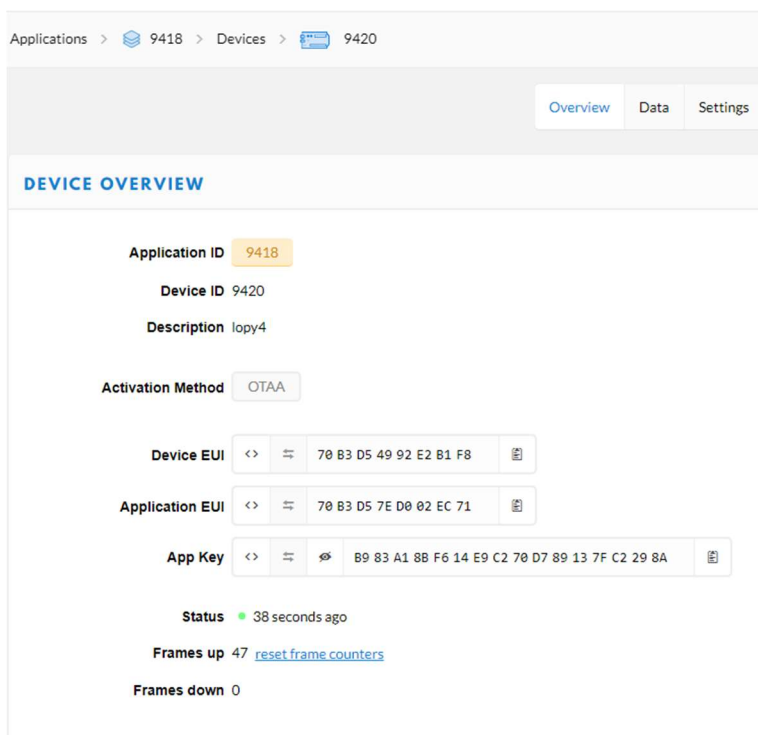
# iniciace sítě LoRa pro evropský region
lora = LoRa(mode=LoRa.LORAWAN, region=LoRa.EU868)
lora.join(activation=LoRa.OTAA, auth=(dev_eui, app_eui, app_key), timeout=0,
dr=5)

# čekání než se zařízení přihlásí do sítě
while not lora.has_joined():
    time.sleep(2.5)

# vytvoření LoRa socketu
s = socket.socket(socket.AF_LORA, socket.SOCK_RAW)
# nastavení datové rychlosti přenosu
s.setsockopt(socket.SOL_LORA, socket.SO_DR, 5)
# uvedení socketu do blokujícího stavu
s.setblocking(True)
# odeslání zprávy
s.send('Hello Sigfox')
# uvedení socketu do neblokujícího stavu
s.setblocking(False)
```

Příchozí žádost o přihlášení do sítě a příchozí zprávy můžeme sledovat v TTN backendu v záložce „Data“. Pokud se zde žádné záznamy neobjeví, znamená to, že žádná z bran sítě

LoRa zprávu nepřijala. To může být způsobeno tím, že není žádná v dosahu, ale také tím, že brána přijala zprávu poškozenou. Pokud se tak stane, můžeme si pomoci změnou hodnoty parametru „dr“ v příkazu `lora.join(...)` a v příkazu nastavení datové rychlosti přenosu. Tento parametr udává použitý Spread Factor. Hodnota `dr=5` odpovídá SF 7, tedy nejvyšším datovým rychlostem, ale také nejnižšímu dosahu a odolnosti proti rušení. Snižováním parametru „dr“ k nule zvyšujeme SF k 12 a zvyšujeme tak i pravděpodobnost, že bude zpráva doručena do sítě TTN.



Obr. 4.12: Okno „Device Overwiev“ v síti TTN

Po úspěšném přihlášení a přenosu dat se v „Device overwiev“ změní barva statusu na zelenou. V záložce „Data“ se nám pak zobrazí přijatá zpráva, kterou rozklikneme. Zpráva obsahuje tzv. metadata, která poskytují informace o datovém přenosu (čas příjmu, frekvence, použitý SF, kódový poměr a seznam bran které zprávu přijaly včetně přijatého RSSI a SNR). V případě příjmu jedné zprávy více branami se duplicitní zprávy automaticky odstraňují na úrovni serveru na základě sekvenčního čísla zprávy. Dále zpráva obsahuje přenášená data, tzv. payload, která jsou stejně jako v případě Sigfoxu formátována do hexadecimálního kódu. Zpráva „Hello TTN“ je tedy přijata jako „41686F6A2054544E“ a její správnost opět ověříme jakýmkoliv dostupným konvertorem.

4.5 Platformy IoT

Tato kapitola je věnována dvěma zvoleným platformám IoT využitým v rámci vytvořených aplikací. Jedná se o popis jejich funkcí a možností a popis jejich používání.

4.5.1 Pybytes

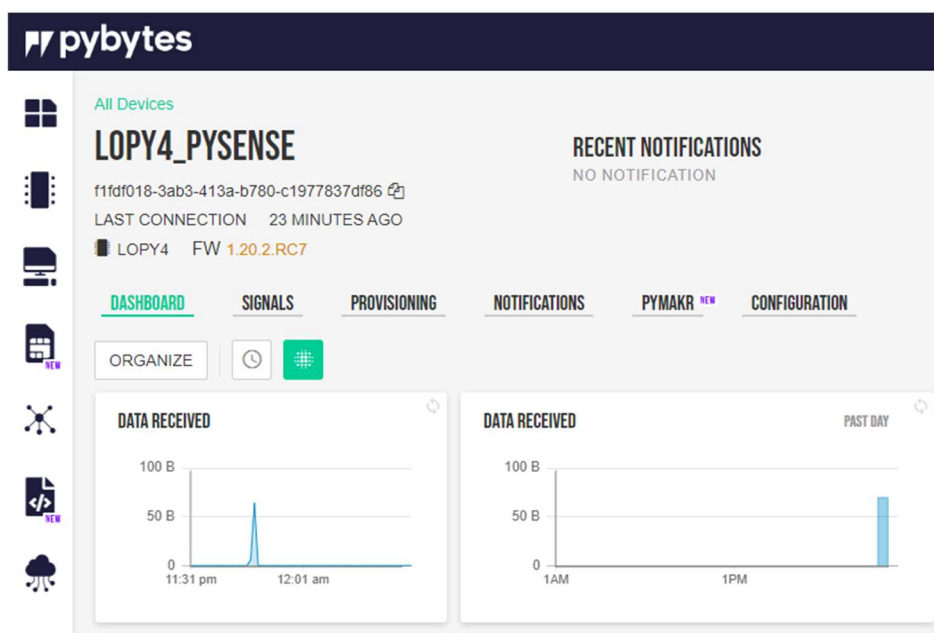
První zvolenou platformou je platforma s názvem Pybytes. Tato platforma byla zvolena především proto, že ji spravuje a rozvíjí stejná společnost, která vyvíjí i použité moduly, tedy společnost Pycom. Díky tomu poskytuje platforma značnou integritu s jednotlivými vývojovými moduly. Za zmínku stojí možnost aktualizace firmware zařízení na dálku, tzv. aktualizace Over-the-Air (OTA). Další užitečná funkce platformy Pybytes je integrovaný webový plugin Pymakr, který funguje stejně tak jako na PC, a integrace konzole, skrze kterou je možné odesílat a nechat vykonávat příkazy v reálném čase. Je tedy možné zakládat nové složky a projekty, které lze následně na dálku nahrát do zařízení.

Prvním krokem k využívání Pybytes je založení účtu na webové stránce `pybytes.pycom.io`. Po přihlášení je nutné přidat zařízení. Vlevo na vertikální liště přejdeme do záložky „Devices“ a dále klikneme na „Add Device via USB“. Poté vybereme ze seznamu náš modul (LoPy4) a vybereme, které síť chceme využívat (můžeme zvolit všechny). Dále vyplníme jméno a heslo k Wi-Fi síti, ke které se má modul LoPy4 připojit. Poté vyplníme název našeho zařízení a jako LoRa Server zvolíme TTN. „LoRa activation type“ ponecháme na LoRa OTAA (Over-the-Air-Authentization). Po vytvoření zařízení budeme přesměrováni do záložky „Provisioning“, kde získáme aktivační token, kterým aktivujeme funkce Pybytes v modulu LoPy4. Aktivaci můžeme provést dvěma způsoby, a to pomocí aktualizacího programu (krok, který jsme přeskočili při aktualizaci firmware desky v kapitole 3.2), anebo pomocí aktivačního příkazu vloženého do konzole. Samotná aktivace trvá přibližně 15 sekund.

V dalším kroku již naprogramujeme desku LoPy4 ke sběru dat ze senzorů a jejich odeslání do Pybytes cloudu. Pro tuto aplikaci je již nutné importovat knihovny jednotlivých senzorů a další knihovny, které nejsou součástí systému. Ve složce se souborem `main.py` vytvoříme další složku s názvem „lib“, do které všechny potřebné knihovny nakopírujeme. Při dalším nahrání programu do desky LoPy4 se s programem automaticky nahrají i knihovny. Všechny potřebné knihovny jsou volně dostupné ke stažení na GitHub profilu společnosti Pycom

(github.com/pycom/pycom-libraries). Testovací řídicí program pro Pybytes je součástí přílohy B a obsahuje zkušební přenos dat ze všech dostupných senzorů modulu. Vzhledem k již zmíněné vysoké integraci Pybytes v systému desky LoPy4 není nutné manuálně naprogramovat připojení a odeslání dat do sítí Wi-Fi, LoRa nebo Sigfox. Bohužel ale ani není možné v programu zvolit preferovanou síť. Preference sítí jsou uloženy v konfiguračním souboru `pybytes_config.json` ve složce projektu. Tento soubor je možné upravit, avšak změny se projeví až po restartování desky. Druhou možností změny preference sítí je změna pořadí v cloudu Pybytes (Device>Configuration>Networks), následné vygenerování nového aktivačního tokenu v záložce Provisioning a opětovná aktivace Pybytes v modulu LoPy4. Příkaz kódu pro odeslání dat do sítí Wi-Fi, Lora a Sigfox je pak vždy stejný.

Bohužel Pybytes neumožňují odeslat všechna data v jednom balíčku, a proto je nutné opakované odesílání pro každou naměřenou veličinu zvlášť. Odeslaná data, v tomto případě tzv. signály, se zobrazují v Pybytes cloudu v záložce Device>Signals v tabulce „Undefined Signals“. Na této stránce je také možné jednotlivé signály definovat a přiřadit jim příslušné jednotky. Vpravo klikneme na „Define New Signal“ a vyplníme. Vizualizaci dat na dashboardu cloudu Pybytes nastavíme kliknutím na nově definovaný signál a dále kliknutím na „Create New Display“. Zde vybereme, jakým způsobem chceme veličinu vizualizovat (tabulka, graf) a doplníme nastavení. Políčko „Pull size“ udává počet hodnot/bodů zobrazených v tabulce/grafu. Dále zaškrtneme políčko „Display on Dashboard“ a klikneme „Create“. Obdobně vytvoříme vizualizace pro další libovolné signály. Poté již můžeme sledovat všechny naměřené hodnoty v záložce „Dashboard“. Velikosti a umístění jednotlivých grafů/tabulek je možné libovolně upravovat pomocí tlačítka „Organize“.



Obr. 4.13: Dashboard cloudové služby Pybytes

Jak již bylo zmíněno, v případě, že chceme využít síť Sigfox, je nutné manuálně změnit preferenci sítě. Navíc k tomu je nutné nastavit cloudu Pybytes Sigfox API přístup. V Sigfox backendu přejdeme do záložky „Group“ a dále do námi již dříve vytvořené skupiny. V levém postranním menu klikneme na „API Access“ a vytvoříme nový. Do názvu vyplníme „Pybytes“ a přiřadíme profil „LIMITED ADMIN“. Zpět v cloudu Pybytes v záložce „Settings“ přejdeme na nastavení sítě Sigfox, kde vyplníme vygenerované klíče „API login“ a „API password“, a připojíme se. Pokud vše proběhne úspěšně, zobrazí se v záložce „Provisioning“ v poli Sigfox registrace dvě zelená znamení. První znamená úspěšné spojení cloudu Pybytes s backendem Sigfox a druhé znamená úspěšné automatické vytvoření callbacku v Sigfox backendu, který přeposílá přijatá data do cloudu Pybytes. Data jsou poté přijímána cloudem Pybytes, přiřazena k jednotlivým signálům a automaticky dekodována z hexadecimálního formátu.

Bohužel je využití sítě Sigfox pro uvedenou aplikaci nevhodné. Z důvodu nutnosti odesílání naměřených veličin v samostatných zprávách bychom vyčerpali denní limit 140 zpráv během tří hodin (uvažujeme-li odesílání dat každých 10 minut). Zároveň bychom v tomto případě porušovali nařízení vlády České republiky a Evropské unie, které stanovuje maximální možnou délku aktivní doby radiové části zařízení na 1 % času z jedné hodiny. Jinými slovy, v jedné hodině může být radiová část zařízení Sigfox aktivní pouze 36 sekund. Poslední nevýhodou je omezená délka zprávy. Sigfox má omezení na 12 bajtů, avšak režie Pybytes z těchto 12 bajtů využije další 4 bajty (2 pro identifikaci zařízení, 1 pro číslo signálu

a 1, který udává typ dat – celočíselná hodnota, hodnota s plovoucí řádovou čárkou nebo text). Pro odesílání jedné naměřené hodnoty v jedné zprávě by bylo 8 bajtů dostačující. Avšak pokud bychom chtěli posílat textové zprávy, znamenalo by to omezení na pouhých 8 znaků abecedy. Tyto problémy částečně řeší využití sítě LoRa.

Před používáním sítě LoRa je nutné znovu zopakovat aktivační proceduru. Systém Pybytes využívá pro LoRa infrastrukturu společnosti TTN. Natavení pro síť LoRa je ovšem jednodušší, opět díky integraci Pybytes v systému modulu LoPy4. Po aktivaci modulu a odeslání zprávy do sítě LoRa se v backendu automaticky vytvoří nový profil a callback pro přeposlání dat do cloudu Pybytes. Uživatel vůbec nemá k profilu TTN přístup, vše je automatizované. Stejně jako v síti Sigfox platí i v síti LoRa nařízení omezující dobu aktivity radiové části zařízení. Výhodou sítě LoRa je v tomto případě pouze větší možná délka jedné zprávy, a to až 240 bajtů. Ovšem i to je pro navzorkovaný časový průběh signálu EKG a všech tří os akcelerometru málo. Proto jsou pro síť LoRa a Sigfox odesílány pouze data o frekvenci srdečního tepu vypočítaného ze signálu EKG a celková velikost zrychlení vypočítaná z dat akcelerometru.

4.5.2 Wia

Cloudová služba Wia je charakteristická rozmanitou podporou zařízení a programovacích jazyků. Důležitou součástí služby je také možnost práce s daty pomocí java scriptů a vyvolání různých akcí na základě přijatých dat.

Nejprve je nutné registrovat se na stránkách `dashboard.wia.io`. Po přihlášení vytvoříme nový „Space“. Dále v záložce „Devices“ přidáme nové zařízení. Zvolíme zařízení Pycom a následně modul LoPy4. V záložce „Configuration“ právě vytvořeného zařízení si zkopírujeme tzv. Device Secret Key, který vložíme do programu desky LoPy4. Pomocí tohoto klíče je služba Wia schopná rozřazovat všechna příchozí data jednotlivým uživatelům.

Obslužný program je pro všechny následující aplikace shodný a je součástí přílohy C. Program obsahuje funkci sběru dat ze senzorů, funkce pro výpočet srdečního tepu a frekvence dýchání, funkci pro nezbytnou úpravu dat pro přenos do sítí Sigfox a LoRa a jednotlivé funkce pro připojení a odeslání dat do všech sítí. Výběr požadované sítě provedeme nastavením parametru `useNetwork`. Pro síť Wi-Fi je tento parametr roven 1, pro

sít' LoRa je roven 2, pro Sigfox 3, a pokud chceme odesílat data do všech sítí zároveň, je roven 4. Prozatím budeme využívat pouze sít' Wi-Fi a nastavíme tedy parametr `useNetwork = 1`. Program uložíme a nahrajeme na desku.

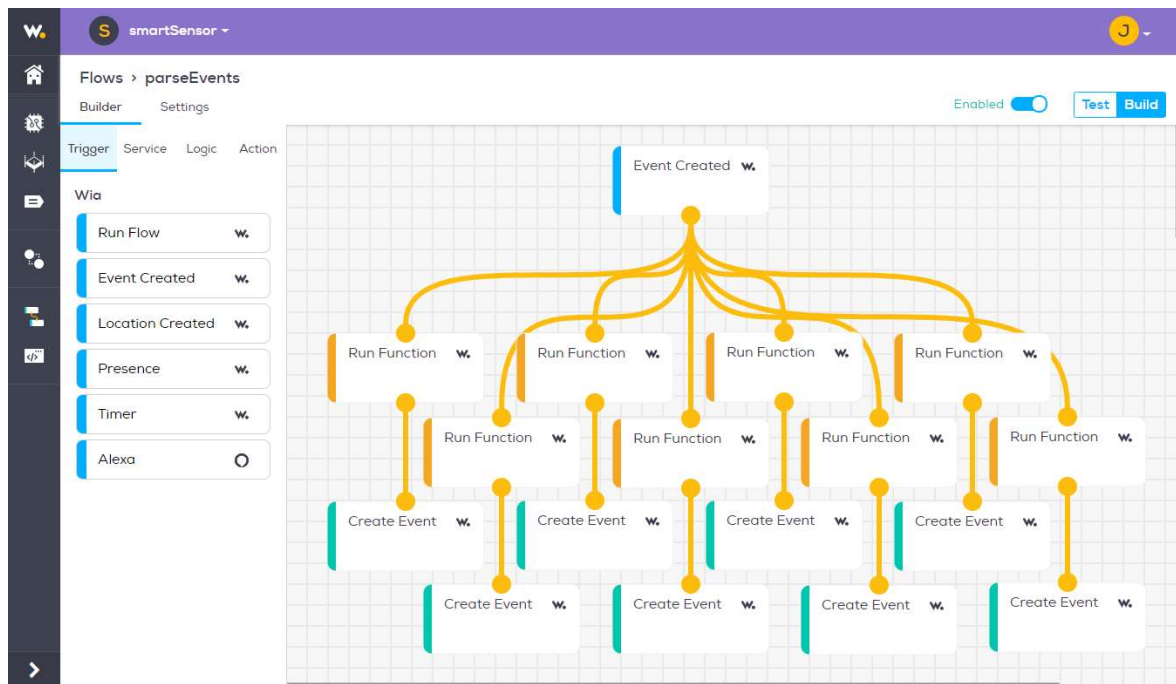
Příchozí data do cloudu Wia jsou dostupná v záložce „Device>Events“. Každý z těchto „Eventů“ obsahuje unikátní identifikační kód, svůj název, data a časové razítko. Vizualizaci dat nastavíme v záložce „Overwiev“ pod tlačítkem „Add Widget“. V okně widgetu nastavíme název, typ vizualizace (graf, číslo, text, ...) a doplníme další informace včetně názvu eventu, ze kterého má vizualizace brát data. Bohužel Wia předpokládá v jednom eventu jednu hodnotu. Pokud event obsahuje hodnot více, neumožňuje Wia vytáhnout z daného eventu pouze jednu z hodnot pro použití ve vizualizaci. Obdobný problém nastává i s vizualizací navzorkovaných časových průběhů signálu EKG a hodnot všech tří os akcelerometru. Přenos vzorků samotných (klidně i několika desetitisíců najednou), včetně jejich časových značek pro časovou osu, proběhne naprosto v pořádku.

Data je možné posílat na Wiu po jedné hodnotě pokaždé s vlastním názvem eventu. Tento přístup je sice ze strany Wia cloudu jednodušší, avšak ze strany modulu LoPy4 se jedná o navýšení aktivní doby z 24 s na 34 s, a tím i navýšení spotřeby elektrické energie na téměř dvojnásobek. Řešením této situace je vytvoření jednoduché aplikace ve Wia cloudu, která rozdělí data obsažená v jednom eventu do svých jednotlivých eventů.

V levém postranním menu přejdeme do „Flow Studio“, kde vytvoříme nový „flow“. Tyto „flows“ umožňují práci s daty a na jejich základě vyvolání příslušných akcí. Záložka „Triggers“ obsahuje bloky, resp. podmínky, kdy se má flow spustit. Jako trigger vybereme blok „Event Created“, do kterého vložíme název eventu tak, jak jej máme definovaný v řídicím programu desky. Dále zvolíme, pro jaké zařízení ze seznamu má flow fungovat. Následně přejdeme do záložky „Logic“, ve které zvolíme blok „Run Function“. Tento blok umožňuje pracovat s daty pomocí javascriptů. Vložíme a uložíme následující skript:

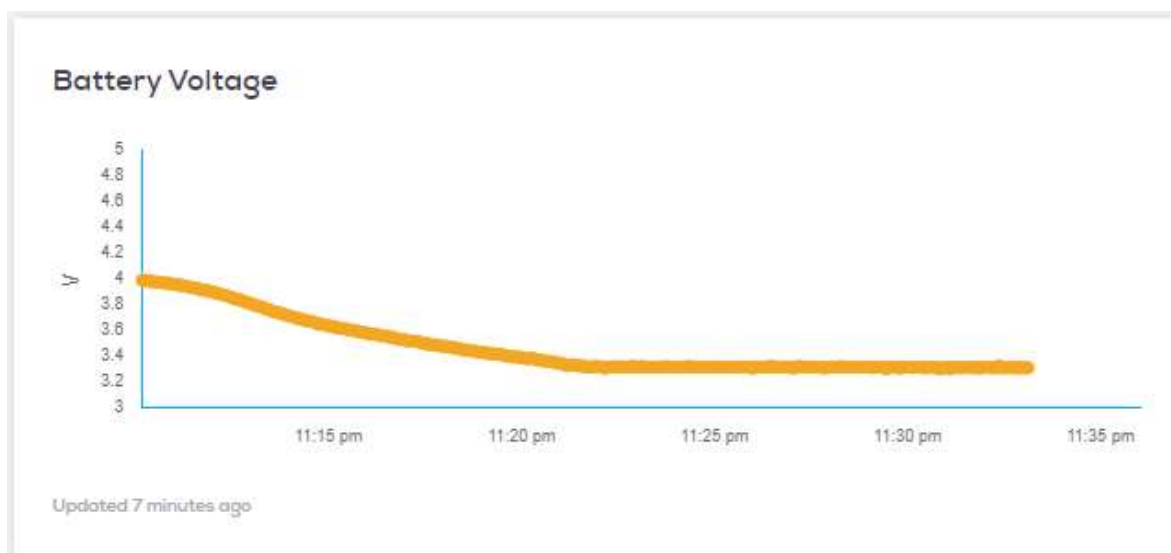
```
var x = input.body.data.temp;  
output.body.data = x;
```

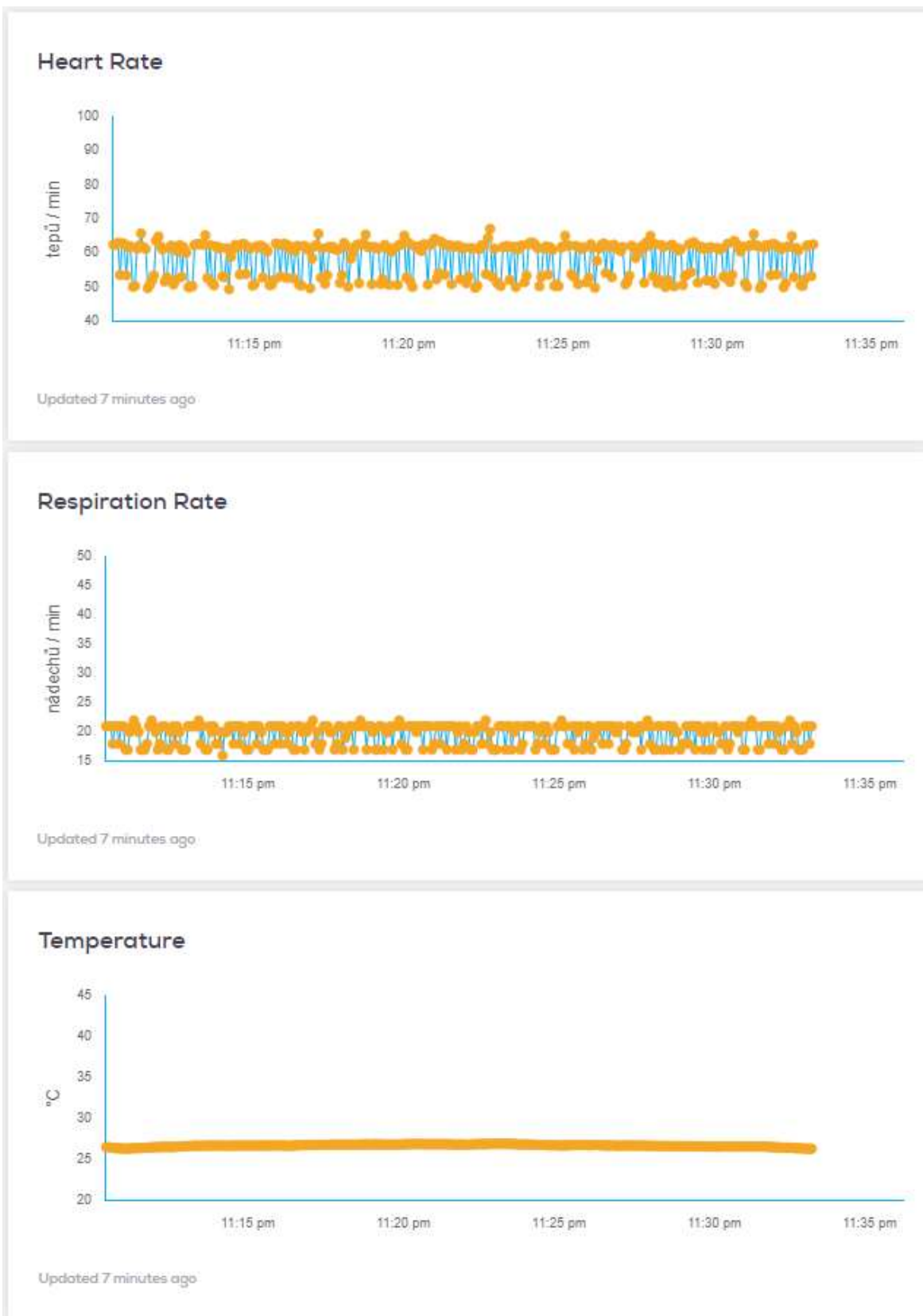
Následně ze záložky „Action“ vybereme blok „Create Event“ do kterého doplníme název nového eventu. Nakonec je nutné bloky mezi sebou propojit. Obdobně vytvoříme bloky „Run Function“ a „Create Event“ pro všechny data (viz následující obrázek).

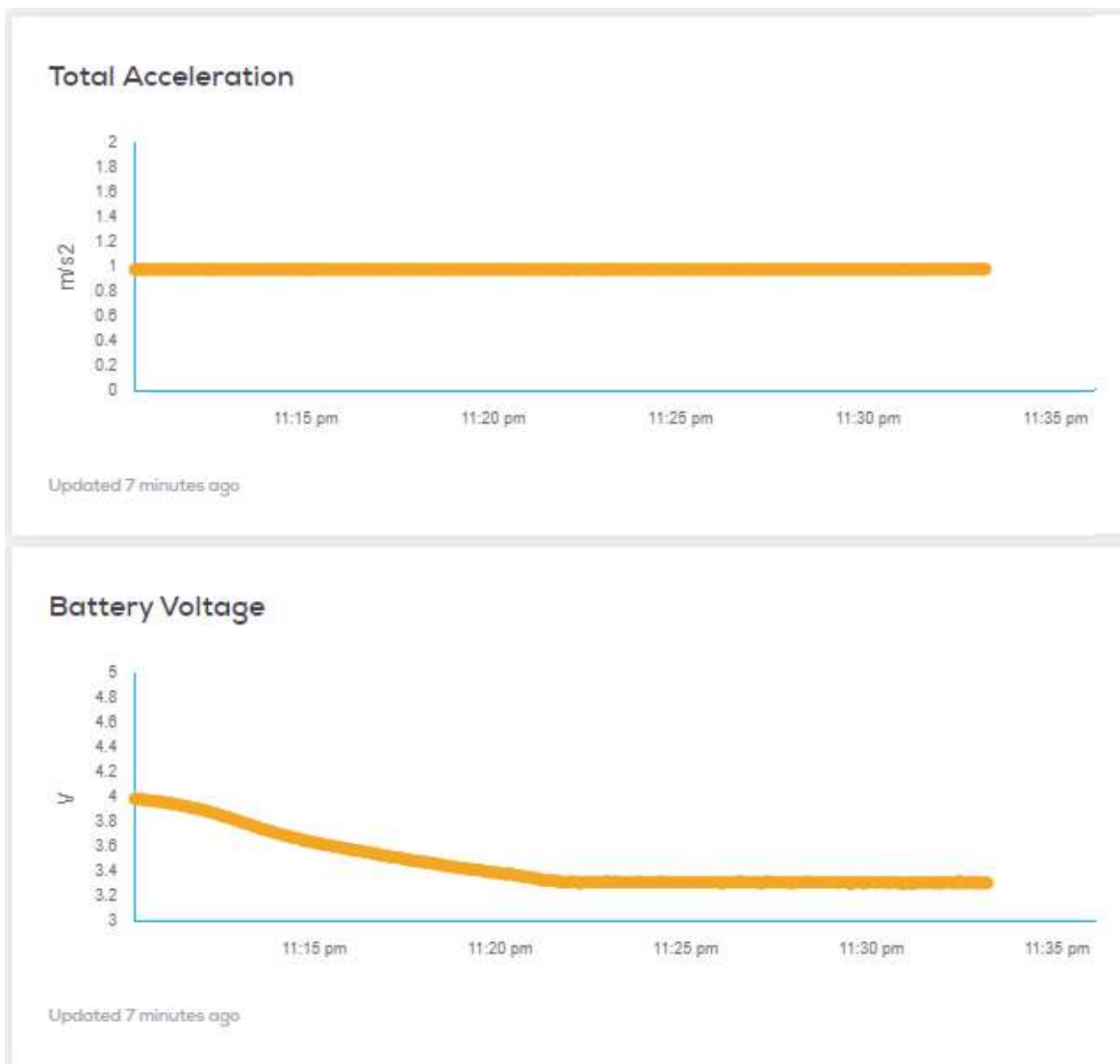


Obr. 4.14: Rozdělení dat pomocí flow

Po rozdělení naměřených dat do jednotlivých eventů je již možné tato data jednoduše vizualizovat. Vizualizovaná data je možné zobrazit v záložce „Overwiew“ daného zařízení nebo na dashboardu celého „Space“. Na následujících grafech je pěkně vidět vybíjení baterie a připojení zařízení k nabíječce. Měření napětí není zcela přesné, protože není měřeno na terminálech baterie, ale na vstupu LDO napěťového regulátoru mikrokontroleru. Proto měří při nabíjení okolo 4,7 V, i když ve skutečnosti napětí baterie při nabíjení nepřekročí 4,2 V.

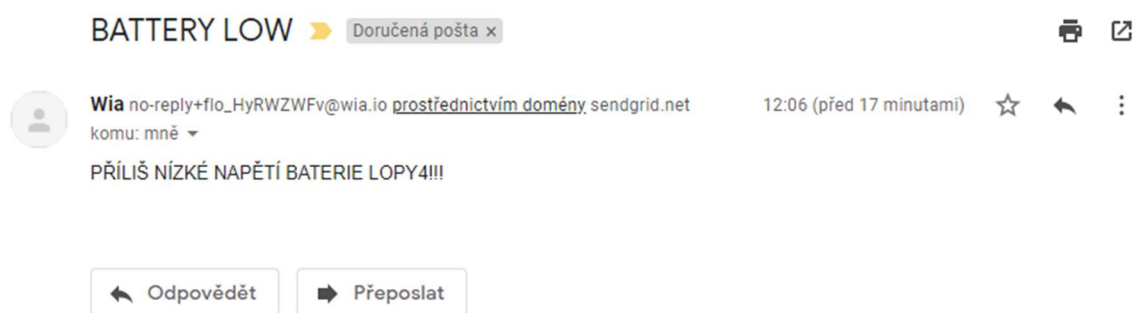






Obr. 4.15: Vizualizace naměřených hodnot

Jako další vytvoříme aplikaci, která nás bude upozorňovat na nízký stav baterie prostřednictvím emailu. V záložce „Flow Studio“ vytvoříme nový flow. Jako trigger zvolíme opět „Event Created“. Ze záložky Logic vybereme nejprve „Update State“, kde vyplníme název (Key) a hodnotu (Value). Do pole „Value“ vložíme následující kód: `${input.body.data.batt}`. Dále přidáme blok „Check State“ kde vyplníme podmínku (v tomto případě „Less than“ a hodnotu napětí baterie, pod kterou se vyvolá akce. Akci s názvem „Send Email“ vybereme v záložce „Action“ a vyplníme emailovou adresu, předmět a text.



Obr. 4.16: Upozornění na nízký stav baterie prostřednictvím emailu

V další části práce se budeme zabývat integrací služeb Sigfox do služby Wia. Nejprve upravíme v řídicím programu parametr `useNetwork` na hodnotu 3. V levém postranním menu ve Wia přejdeme na záložku „Integrations“ a zde vybereme Sigfox. Wia po nás nyní požaduje klíče API login a API password. Wia vyžaduje jiný profil přístupu než Pybytes. Proto musíme vytvořit nové klíče s rozdílem jiného jména a profilem „DEVICES_MESSAGES [R]“. API klíče zkopírujeme do Wia a integraci vytvoříme. Wia nám poté vygeneruje tzv. „inbound key“ ve tvaru `ik_xxx`, který budeme potřebovat pro vytvoření callbacku v Sigfox backendu (viz následující obrázek).

Callbacks

Type

Channel

Custom payload config

URL syntax: `http://host/path?id={device}&time={time}&key1={var1}&key2={var2}...`
 Available variables: `device, time, data, seqNumber, deviceTypeid`
 Custom variables:

Url pattern

Use HTTP Method

Send SNI (Server Name Indication) for SSL/TLS connections

Headers

header value

Content type

Body

```
{
  "integrations":{
    "sigfox":{
      "id":"{device}"
    }
  },
  "name":"sigfoxDataUplink",
  "data":{
    "sigfoxData":"{data}",
    "time":"{time}",
    "duplicate":"{duplicate}",
    "snr":"{snr}",
    "station":"{station}",
    "avgSnr":"{avgSnr}",
    "lat":"{lat}",
    "lng":"{lng}",
    "rssi":"{rssi}",
    "seqNumber":"{seqNumber}"
  }
}
```

Obr. 4.17: Nastavení callbacku pro službu Wia

Po doručení zprávy na backend Sigfoxu a úspěšném přeposlání dat do cloudu Wia se ve Wia v záložce „Devices“ automaticky vytvoří nové zařízení, ve kterém v záložce „Events“ můžeme zobrazovat příchozí data.

Nezbytným krokem byla úprava a kódování naměřených dat. V případě využití sítě Wi-Fi nebylo nutné se příliš zabývat množstvím odesílaných dat, avšak s omezením délky zprávy v síti Sigfox na 12 bajtů bylo nutno odesílané množství dat nějakým způsobem řešit (viz. kapitola 4.1). Po úpravě dat bylo ještě nutné převést data z decimálního zápisu na

hexadecimální. Celý tento postup úpravy a převodu dat je zapsán v kódu následovně (ukázka pro úpravu a převod hodnoty napětí baterie):

```
battPack = ustruct.pack('b', round(sensorData['batt'] * 100) - 300)
```

Všechny takto upravené a zakódované hodnoty převedeme z decimálního zápisu na hexadecimální a poskládáme je za sebe do jedné hexadecimální proměnné pomocí funkce `packData()`.

Po doručení dat na cloud Wia je nejprve nutné rozdělit doručený hexadecimální kód na jednotlivé bajty, resp. dvojice bajtů, které bude následně nutné převést na decimální hodnoty. Toho dosáhneme opět pomocí aplikace ve „Flow Studiu“. Triggerem zde bude opět event s příchozími daty a akcí bude opět vytvoření nového dekodovaného eventu. Mezi tyto bloky opět vložíme blok „Run Function“ a následujícím javascriptem:

```
var hexBatt = input.body.data.slice(0,2);  
var decBatt = parseInt(hexBatt,16);  
output.body.data = (decBatt + 300)/100;
```

První řádka javascriptu vytáhne z celého hexadecimálního stringu zadané bajty dat. Jelikož je s daty pracováno ve formátu Little Endian (což znamená, že první bajt je bajt s nejmenším významem), museli bychom, pro hodnoty přenášené ve dvou bajtech, nejprve načíst druhý a pak až první bajt. Druhá řádka převádí hexadecimální zápis na decimální a poslední řádka zajišťuje přepočet/dekodování na reálnou naměřenou hodnotu.

Obdobným způsobem vytvoříme scripty na dekodování všech veličin. Nutné při tom je znát pořadí, v jakém naměřené veličiny skládáme za sebe do jedné hexadecimální proměnné v programu modulu LoPy4. Následné vytvoření a nastavení vizualizace dat je již shodné s předchozím projektem, ve kterém jsme využívali síť Wi-Fi.

Další část práce se zabývá stejnou aplikací, avšak s využitím sítě LoRa. Podobně jako u projektu se sítí Sigfox je nejprve nutné změnit v řídicím programu modulu LoPy4 parametr `useNetwork` na hodnotu 2. Integraci v cloudu Wia provedeme v záložce „Integrations“, kde vybereme integraci The Things Network. Po vytvoření integrace nám Wia poskytne tzv. „Inbound Key“, který budeme potřebovat pro vytvoření callbacku v TTN backendu.

V backendu TTN v naší již dříve vytvořené aplikaci (kapitola 4.4.3) přejdeme na záložku „Integrations“ a z nabídky vybereme „HTTP Integration“. Na následující stránce vyplníme všechny potřebné údaje (viz následující obrázek) a integraci vytvoříme.

The screenshot shows the 'ADD INTEGRATION' page in the TTN Console. The page title is 'HTTP Integration (v2.6.0)' by 'The Things Industries B.V.'. The description states: 'Sends uplink data to an endpoint and receives downlink data over HTTP.' There is a link to 'documentation'.

The form fields are:

- Process ID:** The unique identifier of the new integration process. Value: 9419.
- Access Key:** The access key used for downlink. Value: default key (with buttons for 'devices' and 'messages').
- URL:** The URL of the endpoint. Value: https://integrations.wia.io/api/ttn/uplink.
- Method:** The HTTP method to use. Value: POST.
- Authorization:** The value of the Authorization header. Value: Bearer ik_m3O6NQ2lhBRn5T7IEBqX4xGIJVFwE.

Obr. 4.18: Integrace TTN do cloudu Wia

Pole „Process ID“ můžeme zvolit libovolně, „Access Key“ ponecháme default, za URL zvolíme `https://integrations.wia.io/api/ttn/uplink`, „Method“ zvolíme POST a do pole „Authorization“ vyplníme „Bearer + vygenerovaný Inbound Key z Wii“.

Podobně jako při integraci se Sigfoxem se následující doručená zpráva do sítě TTN přepoše do cloudu Wia, kde se opět automaticky vytvoří nové zařízení v záložce „Devices“. Event

s příchozími daty se jmenuje ttnUplink a obsahuje veškerá naše data včetně metadat dané zprávy. Metadata obsahují například informace o použitém SF, na jaké frekvenci byla zpráva vysílána a s jakým SNR a RSSI byla zpráva přijata.

Nevýhodou oproti integraci Sigfox je fakt, že integrace TTN nepreposílá přijatá data ve stejném formátu, ve kterém je přijme – v hexadecimálním, ale kóduje je do formátu Base64. Konverzi z Base64 do formátu Hex je možné provést pomocí javascriptů jazyka Java, ovšem tuto konverzi se nepodařilo ve službě Wia zprovoznit. Naštěstí umožňuje backend TTN s přijatými daty pracovat a upravit je ještě před přeposláním do služby Wia. V aplikaci na webu TTN přejdeme do záložky „Payload Formats“, kde vytvoříme jednoduchou funkci na převod dat z hexadecimálního zápisu na decimální. Díky tomuto formátování tak na Wiu dorazí data nejen ve formátu Base64, ale také jako seznam decimálních hodnot podobně jako tomu bylo při využití sítě Wi-Fi.

Dekódování decimálních hodnot na reálné naměřené hodnoty můžeme provést jak v TTN backendu, tak pomocí „flows“ v cloudu Wia. Pro odlišnost s předchozím projektem jsme data dekodovali přímo v backendu TTN.

The screenshot shows the TTN Console interface for configuring payload formats. The breadcrumb trail is 'Applications > 9418 > Payload Formats'. The main navigation tabs are 'Overview', 'Devices', 'Payload Formats', 'Integrations', 'Data', and 'Settings'. The 'PAYLOAD FORMATS' section is active, showing a dropdown menu set to 'Custom'. Below the dropdown are tabs for 'decoder', 'converter', 'validator', and 'encoder'. A 'remove decoder' link is visible on the right. The code editor contains the following JavaScript function:

```
1 function Decoder(bytes, port) {
2   var batt = bytes[0];
3   var temp = bytes[1];
4   var heartRate = bytes[2];
5   var breathRate = bytes[3];
6   var accelAvg = (bytes[5] << 8) | bytes[4];
7
8   return {
9     batt: (batt+250)/100,
10    temp: (temp+250)/10,
11    heartRate: heartRate,
12    breathRate: breathRate,
13    accelAvg: accelAvg/100,
14  };
15 }
```

A status message at the bottom right of the code editor says 'decoder has no changes'.

Obr. 4.19: Formátování a dekódování dat v backendu TTN

Posledním krokem před možností vizualizovat data v cloudu Wia je nutnost přijatá data rozdělit do svých jednotlivých eventů. To provedeme opět pomocí nové aplikace ve „Flow Studiu“, jejíž naprogramování je obdobné jako u předchozích projektů. Stejně tak je obdobné i vytvoření vizualizací dat.

5 Ověření parametrů systému

Cílem měření bylo ověření vlastnosti připojení a odeslání dat do sítí Wi-Fi, LoRa a Sigfox, a to z hlediska kvality signálu, spotřeby elektrické energie a také z hlediska spolehlivosti.

5.1 Hodnoty SNR a RSSI

Parametr RSSI je zkratka pro Received Signal Strength Indicator a reprezentuje celkovou přijatou energii na přijímači. Parametr SNR je zkratka pro Signal to Noise Ratio a reprezentuje poměr výkonové úrovně užitečného signálu k úrovni šumu.

Hodnoty SNR a RSSI pro síť Sigfox získáme z přehledu v Sigfox backendu nebo z metadat eventu ve Wia. Backend TTN zprávy ani jejich metadata neukládá, a tudíž jsou hodnoty SNR a RSSI dostupné pouze z metadat eventu v cloudu Wia. Hodnoty v následující tabulce jsou zprůměrovány ze 100 vzorků odesílaných po 15 minutách.

Tabulka 5.1: Naměřené parametry SNR a RSSI

		Chodba cihlového domu	Byt – místnost s okny	Garáž	Volné prostředí mimo město
Sigfox	SNR [dB]	21,05	21,48	19,89	12,78
	RSSI [dBm]	-98,67	-92,36	-116,47	-131,61
	Ztráta zpráv [%]	0	0	0	2,5
LoRa	SNR [dB]	-6,92	-5,66	--	--
	RSSI [dBm]	-117,86	-115,37	--	--
	Ztráta zpráv [%]	36,8	31,8	100	100

Nutno podotknout, že všechna měření pro síť LoRa probíhala s nastaveným Spread Factorem na nejvyšší možný, tedy 12. S nastaveným Spread Factorem na 11 v chodbě panelového domu nebyla zpráva přijata nikdy, a v mém bytě v místnosti s okny dorazila zpráva pouze ve výjimečných případech (přibližně 1 ze 100). Na spolehlivost doručení zprávy má zajisté také vliv počasí. Za lehkého deště klesaly hodnoty SNR k hraničním -20 dB a hodnoty RSSI k -120 dB. Naopak za hezkého počasí dorazila zpráva výjimečně i k druhé vzdálenější braně.

Vzdálenost zařízení k nejbližším LoRa branám při měření v bytě, na chodbě cihlového domu a ve zděné garáži byla přibližně 2 km, 2,5 km, 3 km, 3,5 km a 4,5 km. Bohužel ale není možné zjistit, kterou konkrétní branou byly zprávy přijaty. Metadata zprávy LoRa sice

obsahují identifikační čísla bran, které zprávu přijmou, ale jejich lokaci ne. Stejně tak není možné získat lokaci z portálu TTN a společnost samotná tuto informaci neposkytuje.

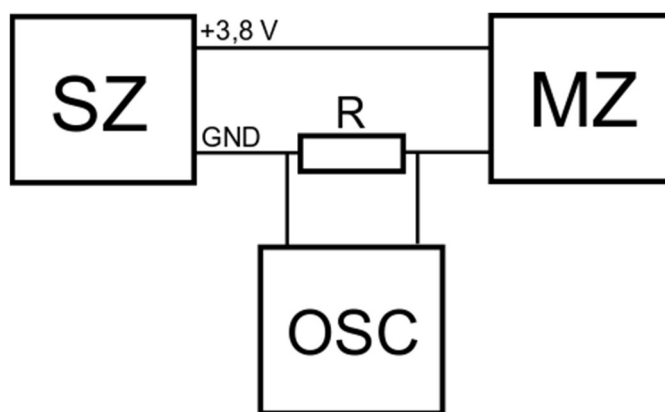
Naměřené hodnoty v tabulce jsou průměrné hodnoty naměřené během celého jednoho dne, avšak zajímavostí bylo sledovat změnu krátkodobého průměru v závislosti na denní době. Například při měření v garáži bylo mezi půlnocí a sedmou ráno naměřené průměrné RSSI -115 dBm a mezi sedmou ráno a druhou hodinou odpoledne -118 dBm.

Další měření probíhalo během cestování po Plzni a jejím blízkém okolí. Hodnoty SNR a RSSI jsou v tomto případě nepoužitelné, protože je neustále přibližujeme a vzdalujeme od jednotlivých základnových stanic. Důležitá je ale procentuální ztrátovost zpráv. Během cestování autem po městě při průměrné rychlosti 40 km/h bylo ztraceno v síti Sigfox i síti LoRa 20 % zpráv. Při chůzi po městě klesla ztráta zpráv Sigfoxu na 0 %, zatímco v síti LoRa zůstává výsledek nezměněn. Během cestování v blízkém okolí města Plzně v autě průměrnou rychlostí 80 km/h bylo ztraceno v síti Sigfox i síti LoRa přes 90 % zpráv. Vliv na ztrátovost zpráv nemá pouze zvýšená rychlost vozidla, ale také fakt, že vozidlo samo částečně stíní elektromagnetické vlnění. Při měření za jízdy bylo zařízení umístěno na palubní desce pod čelním sklem. Při nulové rychlosti a vysílacím zařízení umístěné mimo vozidlo klesla ztráta zpráv v síti Sigfox na necelá 2 %, avšak v síti LoRa pouze na 46 %.

Získat hodnoty SNR a RSSI pro síť Wi-Fi je v podstatě nemožné. Zprávy odesílané jejím prostřednictvím neobsahují žádná metadata a běžné Wi-Fi routery tyto hodnoty ani neuvádějí. Můj Wi-Fi router je o něco chytřejší a dokáže zobrazovat hodnotu RSSI připojených zařízení. Při měření s vysílacím zařízení ve stejné místnosti s routerem byla hodnota RSSI -32 dBm, při měření v místnosti za dvěma stěnami -69 dBm a při měření na ulici před domem se již zařízení k routeru vůbec nepřipojilo. Na síti Wi-Fi jsou kladeny zcela jiné nároky na přenosové rychlosti a dosah. Tyto hodnoty tudíž nelze porovnávat s hodnotami RSSI sítí LPWAN, kde lze za dobrý signál považovat i signál s hodnotou RSSI -110 dBm.

5.2 Spotřeba elektrické energie

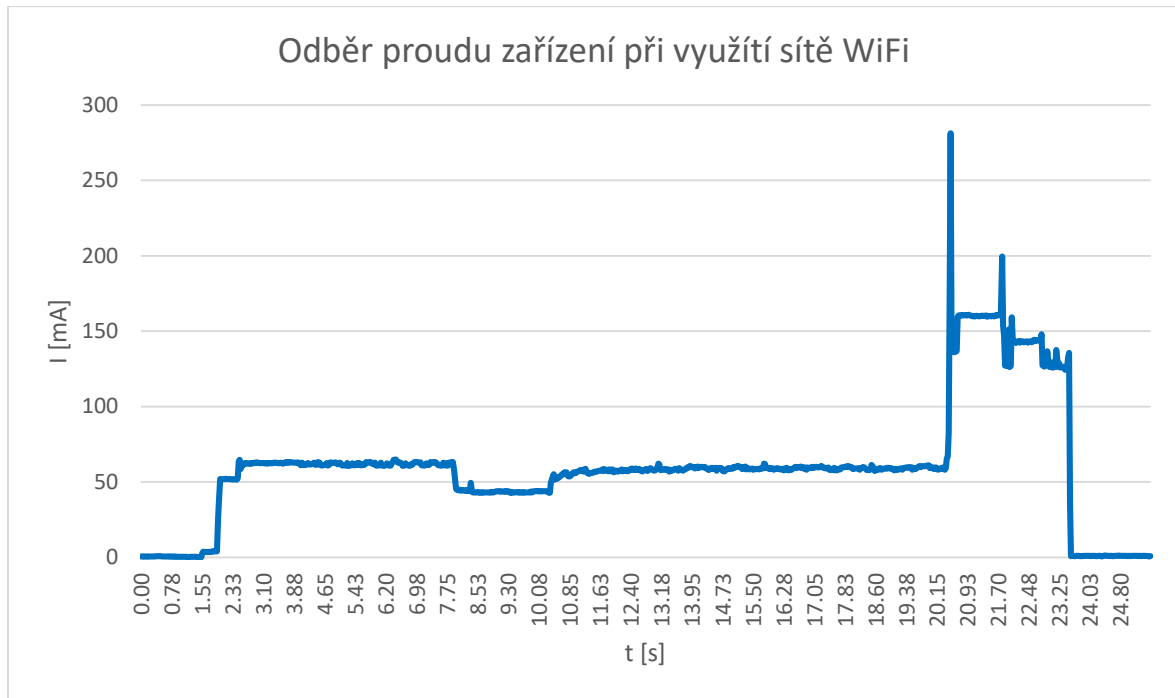
Spotřeba elektrické energie zařízení byla měřena pomocí osciloskopu Agilent 54621A, který měřil úbytek napětí na rezistoru (se známou hodnotou) vloženého do obvodu mezi měřené zařízení a stabilizovaný zdroj. Stabilizovaný zdroj je v tomto případě galvanicky oddělen od sítě. S osciloskopem byla použita sonda Agilent 10074C. Katalogová hodnota rezistoru je $0,47 \Omega$ s tolerancí 5%. Naměřená hodnota rezistoru je $0,469 \Omega$, což je vzhledem k aplikaci zanedbatelná odchylka.



Obr. 5.1: Schéma zapojení měření

Z naměřeného úbytku napětí na rezistoru a z jeho známé hodnoty pak pomocí Ohmova zákona dopočítáme odebíraný proud. Průměrný odebíraný proud v době aktivity zařízení byl vypočítán součtem absolutních hodnot vzorků odebíraného proudu vydělený jejich počtem.

Měření odebíraného elektrického proudu probíhalo v režimu, kdy se zařízení spustí, přibližně 10 sekund načítá data, která následně odešle do zvolené sítě, a nakonec přechází do režimu hlubokého spánku. Pouze při použití sítě Wi-Fi bylo testováno i kontinuální měření a odesílání dat. Důvodem je omezení aktivní doby, a tudíž i omezení možného maximálního počtu odeslaných zpráv za den v sítích Sigfox a LoRa.



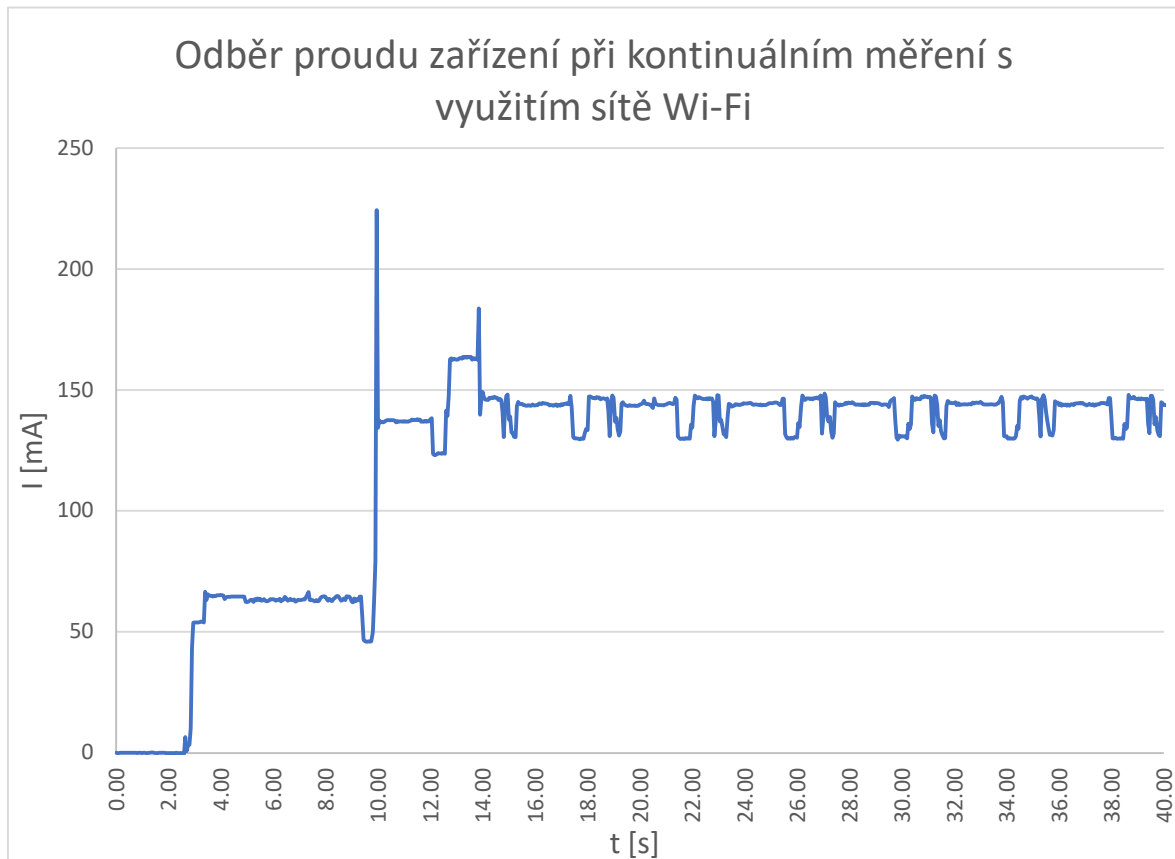
Obr. 5.2: Průběh odebíraného proudu zařízením s využitím sítě Wi-Fi

Průměrný odebíraný proud zařízením od okamžiku jeho probuzení, přes vykonání naprogramovaného kódu, až po okamžik přechodu do hlubokého spánku byl 68,8 mA. Celý tento časový úsek trval 22 sekund. Na počátku nejprve startuje systém MicroPython, který trvá přibližně 6 sekund. Bohužel tato startovací fáze probíhá po každém probuzení zařízení z režimu spánku. Během této doby je odebírána průměrný proud 56,5 mA, poté již následuje vykonávání kódu. Nejprve jde o získání všech dat ze senzorů, které trvá přibližně 12,5 sekundy při průměrném odebíraném proudu 51 mA. Poté následuje inicializace radiové části zařízení a odeslání dat.

V grafu je zřejmý vysoký špičkový odběr proudu ve chvíli, kdy se spouští a inicializuje Wi-Fi radiová část. V této špičce dosahuje odebíraný proud hodnoty až 280 mA. Druhá, nižší proudová špička dosahuje hodnoty 200 mA a jedná se o moment připojení k Wi-Fi routeru. Poté následuje samotné odeslání dat, které trvá přibližně 0,8 sekundy a při kterém je odebírána průměrný proud 140 mA.

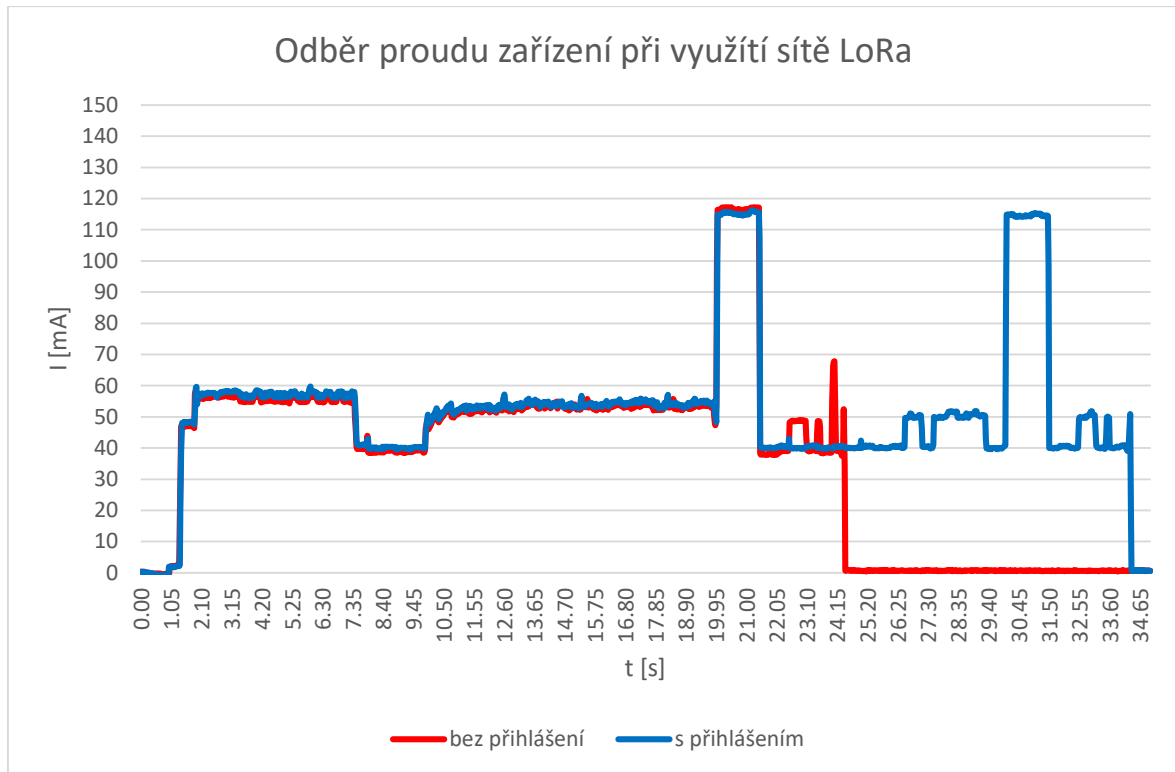
Nakonec přejde zařízení do hlubokého spánku na předem definovanou dobu. Úbytek proudu na měřícím rezistoru o hodnotě $0,469 \Omega$ byl v režimu hlubokého spánku tak malý, že jej nebylo možné změřit. Proto se odebíraný proud v režimu hlubokého spánku měřil s rezistorem o velikosti $5 \text{ k}\Omega$. Úbytek napětí na tomto rezistoru byl 47 mV, a z toho

vypočítaný odebíraný proud ve spánku je roven $9,4 \mu\text{A}$. Pro takto malou hodnotu proudu by byl úbytek napětí na rezistoru o hodnotě $0,469 \Omega$ přibližně $4,4 \mu\text{V}$, což už je pro osciloskop s maximálním rozlišením 1 mV/dílek neměřitelná hodnota.



Obr. 5.3: Průběh odebíraného proudu zařízením při kontinuálním měření s využitím sítě Wi-Fi

V režimu kontinuálního měření přechází zařízení po náběhu systému rovnou k iniciaci a připojení k síti Wi-Fi. Poté se již cyklicky opakuje přibližně 2sekundové měření a odeslání dat. Během této doby odebírá zařízení nepřetržitě průměrně 142 mA .

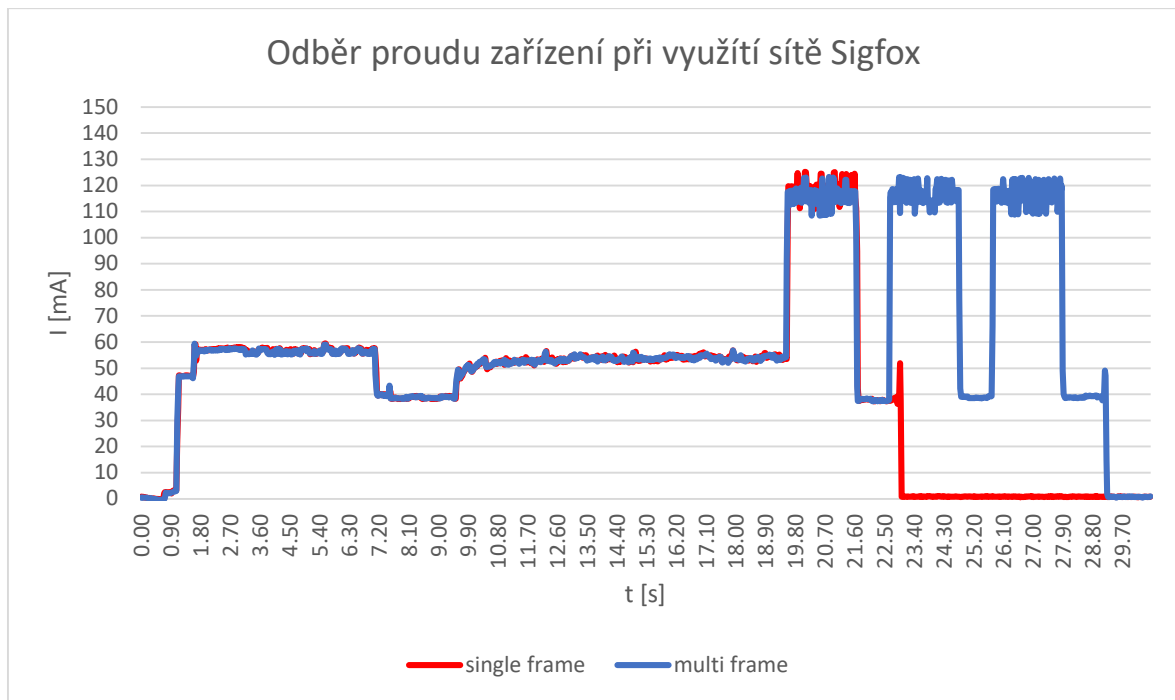


Obr. 5.4: Průběh odebíraného proudu zařízením s využitím sítě LoRa

Pro síť LoRa bylo nutné naměřit dva různé průběhy. Modrý je průběh „s přihlášením“ (viz obr. 5.4) a jedná se o přihlášení zařízení k síti TTN. Během tohoto přihlášení jsou vygenerovány tzv. session klíče, které jsou používány k validaci zprávy v síti a k šifrování a dešifrování uživatelských dat. Tyto klíče ovšem není nutné generovat před každým odesláním dat do sítě. Je možné je uložit do NVRAM paměti zařízení, takže nejsou při přechodu do hlubokého spánku vymazány. Po probuzení se tyto klíče z NVRAM paměti opět načtou a je možné odesílat data bez nutnosti nového přihlášení do sítě (červený průběh).

Odběr proudu při startu systému a získávání dat ze senzorů je shodný pro všechny sítě. Průběhy odběru proudu se liší pouze v inicializaci příslušné radiové části a odeslání dat. Při odeslání dat do sítě LoRa odebírá zařízení průměrný proud 116 mA a odeslání jedné zprávy trvá 1,5 sekundy (SF=12).

Průměrný odebíraný proud v aktivní době zařízení a „s přihlášením“ byl 54,4 mA a tento interval trval 33,3 s. Průměrný odebíraný proud v aktivní době zařízení a „bez přihlášení“ byl 53,8 mA a tento interval trval 23,4 s.



Obr. 5.5: Průběh odebíraného proudu zařízením s využitím sítě Sigfox

Pro Sigfox byly taktéž naměřeny dva průběhy (viz obr. 5.5). Dle specifikací chování zařízení v síti Sigfox je totiž možné nechat odeslat jednu zprávu pouze jednou (červený průběh) nebo třikrát, pokaždé na jiné frekvenci (modrý průběh). Toto chování je možné nastavit. Trojnásobným odesláním zprávy se zajišťuje redundance a zvyšuje pravděpodobnost, že zpráva dorazí na server. Ne vždy je to ale nutné, a pokud to podmínky dovolí, je z hlediska výdrže baterie vhodné tuto proceduru vypnout.

Odeslání jedné zprávy do sítě Sigfox trvá 2,12 s. Během této doby odebírá zařízení průměrný proud 116,5 mA. Poté následuje sekundová pauza. Průměrný odebíraný proud v aktivní době zařízení je v případě trojnásobného odeslání zprávy 64,1 mA a tento interval trvá 28,5 sekund. V případě jednonásobného odeslání zprávy je průměrný odebíraný proud 56,9 mA a doba po kterou je zařízení aktivní trvá 22,2 sekund.

Následující tabulka 5.2 udává odhadovanou dobu provozu zařízení při napájení 2500 mAh Li-Pol baterií s periodou měření 10 minut a při kontinuálním měření. Kontinuální měření by bylo pro nositelná zařízení s uplatněním ve zdravotnictví ideální, avšak využití sítí Sigfox a LoRa toto nedovoluje z důvodu nařízení Evropské unie o omezení aktivní vysílací doby.

Tabulka 5.2: Odhad doby provozu zařízení

	Perioda měření [min]	Odhad doby provozu [dnů]
Wi-Fi	10	42
	Kontinuální měření	0,75
LoRa (bez přihlašování)	10	51
Sigfox (trojnásobné odeslání)	10	35,6
Sigfox (jednonásobné odeslání)	10	51

Při použití baterií s menší kapacitou a tím pádem i menší fyzickou velikostí (díky čemuž by byly vhodnější pro použití v nositelné elektronice) by pak odhadovaná doba provozu klesala spíše k jednotkám dnů a v případě kontinuálního provozu sítě Wi-Fi na jednotky hodin.

Odhad doby provozu na baterie je do jisté míry orientační. Roli hraje například teplota, při které se baterie vybíjí, opotřebení baterie apod. Odhad také nekalkuluje s faktem, že při vybíjení baterie neklesá její napětí lineárně.

Závěr

Cílem této diplomové práce bylo v teoretické části seznámení se s problematikou v oblasti technologií smart textilií a jejich využití. V úvodu této kapitoly je popsáno, co se pod pojmem smart textilie skrývá, jak probíhal jejich vývoj, a také základní klasifikace typů smart textilií. V dalších částech této kapitoly jsou popsány různé materiály pro výrobu smart textilií a různé technologie jejich výroby. V poslední části kapitoly jsou uvedeny požadavky na textilní aplikace a teorie funkce různých typů textilních senzorů a elektrod.

Ve druhé kapitole je ve stručnosti řešena problematika internetu věcí. V úvodu této kapitoly je popsáno, co se pod pojmem „internet věcí“ skrývá, vývoj v průběhu let a predikce do budoucna. Další části kapitoly popisují využití Internetu věcí v různých oblastech infrastruktury od průmyslového řízení a zdravotnictví, přes logistiku a obchod, až po zemědělství a usnadňování každodenního života lidí. Dále je ve stručnosti popsána architektura a typy datové komunikace zařízení v IoT. Třetí kapitola se zabývá technologiemi bezdrátové komunikace a jejich porovnáním. Detailněji jsou zde popsány technologie LPWAN sítí Sigfox a LoRa, které se vyznačují nízkou spotřebou a schopností přenášet data až na vzdálenosti několika desítek kilometrů.

Následující kapitola je těžištěm vlastní práce, jejímž cílem byl návrh, vývoj software a realizace prototypu zařízení spadajícího do kategorie nositelné elektroniky. V úvodu kapitoly je popsán hardware použitý k realizaci. Jedná se o vývojovou sadu od firmy Pycom, konkrétně modul LoPy4 se sensorovou deskou Pysense. Tato sada disponuje širokou škálou funkcí včetně konektivity k síti Wi-Fi a LPWAN sítím Sigfox a LoRa. Právě prověření a porovnání parametrů připojení k LPWAN sítím je jedním z dosažených cílů práce. Dalším dosaženým cílem je vyvinutý obslužný program modulu, který je schopen odesílat a v případě potřeby upravovat data získaná ze senzorů do zvolených sítí. Skutečné textilní senzory však realizovány nebyly a data, která by tyto senzory generovaly byla pouze simulována.

Součástí této realizace je také následná vizualizace a zpracování dat ve zvolených cloudových službách. Podrobný popis práce s vývojovou sadou LoPy4 + Pysense a práce s cloudovými platformami je součástí této kapitoly. Kompletní vlastní obslužný program je součástí přílohy C.

Závěr práce se věnuje měření a ověření parametrů realizovaného prototypu. Z naměřených hodnot průměrného odebraného proudu zařízením při využití různých bezdrátových sítí je zřejmá podobnost mezi sítěmi Sigfox a LoRa za podmínek, že se zařízení nepřihlašuje do sítě LoRa před každým odesláním zprávy a že zařízení odesílá do sítě Sigfox jednu zprávu pouze jednou. V takovém případě je odhadovaná výdrž provozu zařízení na baterii s kapacitou 2500 mAh a s četností odesílání dat každých 10 minut okolo 51 dnů. Pokud by bylo z hlediska spolehlivosti doručení zprávy do sítě Sigfox nutné odesílání jedné zprávy třikrát, vzroste spotřeba zařízení o přibližně 50 %. Spotřeba elektrické energie při využití sítě Wi-Fi je oproti sítím Sigfox a LoRa přibližně o 20 % vyšší. Ovšem pokud bude zařízení měřit a odesílat data prostřednictvím sítě Wi-Fi kontinuálně, bude také odebírat mnohokrát více energie. Oproti odeslání dat s periodou 10 minut jde o navýšení spotřeby elektrické energie o přibližně 600 %. V režimu hlubokého spánku se zařízení chová vždy stejně, nezávisle na využívané síti, a hodnota odebraného proudu v této době je pouhých 9,4 μ A.

S přihlédnutím k výsledkům měření kvality signálu a spolehlivosti doručení zpráv na server je zřejmé, že v rámci domácnosti a dosahu sítě je nejspolehlivějším řešením síť Wi-Fi, i přes její vyšší energetickou náročnost. Zároveň není problém do sítě Wi-Fi odesílat data i kontinuálně, a to s téměř libovolným objemem odesílaných dat. Výhodou dnešní doby je, že většina z nás vlastní chytré telefony s připojením k mobilním sítím, které jsou schopné toto připojení sdílet s dalšími zařízeními prostřednictvím vlastního přístupového bodu sítě Wi-Fi. Díky tomu můžeme využít síť Wi-Fi téměř kdekoli a spolehlivost doručení zpráv na server se tak přenáší spíše na spolehlivost dostupnosti mobilní sítě v daném místě. Avšak v případě nedostupnosti sítě Wi-Fi je z hlediska spolehlivosti vítězem síť Sigfox. Obzvláště pak v městské zástavbě (konkrétně v bytě cihlového domu), kde byla pro síť Sigfox naměřena nulová průměrná ztrátovost zpráv, zatímco v síti LoRa (za stejných podmínek) okolo 35 %. To je do jisté míry způsobeno tím, že společnost Sigfox spolupracuje s mobilními operátory a využívá jejich infrastruktury, zatímco síť LoRa má v Plzni a blízkém okolí méně než 15 základnových stanic. Oproti teoretickému dosahu sítě LoRa se v praxi podařilo navázat spojení pouze na vzdálenosti okolo 5 km, a to i mimo městskou zástavbu. Dosah sítě Sigfox nebylo možné určit, jelikož počty a polohy základnových stanic nejsou známy.

O tom, která z bezdrátových sítí je nejlepší nelze jednoznačně obecně rozhodnout. Finální verdikt bude záviset především na využití daného zařízení a na požadavcích jeho uživatele.

Za podmínek, že bude uživatel chtít mít obecný přehled o svojí fyzické aktivitě například během sportu nebo turistiky, a že nemá zapotřebí při těchto aktivitách mít u sebe mobilní telefon, jsou sítě Sigfox či LoRa vyhovující. Naopak pokud bude chtít uživatel informace o svém stavu častěji než jednou za 10 minut anebo bude-li chtít i další podrobnější a tím pádem i obsáhlejší data, budou již sítě Sigfox a LoRa z důvodů svých omezení nedostačující a uživatelé budou muset využít síť Wi-Fi. Stejně tak tomu bude i v případech, kdy by mělo být zařízení využíváno ve zdravotnictví za účelem sledování zdravotního stavu pacientů. V síti Wi-Fi je možné zvolit interval odesílání dat téměř libovolně krátký a zároveň nejsme omezeni maximálním objemem přenášených dat. Prostřednictvím sítě Wi-Fi je možné přenášet tisíce vzorků časových průběhů křivek EKG či zrychlení akcelerometru bez problému. Naopak nevýhodou Wi-Fi je její vyšší energetická náročnost a nutnost udržení zařízení v dosahu přístupového bodu (Wi-Fi routeru nebo mobilního telefonu).

Jako další pokračování těchto prací, ale již nad rámec zadání diplomové práce, by bylo možno například řešit otázky výroby textilních senzorů a jejich implementace do textilií, odesílání alarmových hlášení například záchranné službě nebo implementace lokalizace zařízení pomocí systému GPS a další.

Seznam literatury a informačních zdrojů

- [1] L. Van Langenhove. „Smart Textiles for Medicine and Healthcare, Materials, systems and applications“. Cambridge, England: Woodhead. 2007. ISBN 978-184-5690-274.
- [2] Stoppa M, Chiolerio A. „Wearable Electronics and Smart Textiles: A Critical Review“, *Sensors*, roč. 14, č. 7, s. 1–36, Červenec 2014. [Online]. [Cit. 7.1.2020]. Dostupné z: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4168435/>
- [3] Moučková K. Technologie a využití smart textilií pro monitorování životních funkcí. Bakalářská práce, ZČU Plzeň, 2017
- [4] TAO, Xiaoming. Smart fibres, fabrics and clothing: materials, systems and applications. Cambridge, England: Woodhead Pub., 2001, xvii, 316 p. ISBN 1855735466.
- [5] H. D. Henock, „Literature over view of Smart textiles“. University of Boras Swedish School of Textiles, čer-2010.
- [6] T. Xiaoming, Ed., Handbook of Smart Textiles, roč. 2015. Springer Science+Business Media Singapore.
- [7] SARIF ULLAH PATWARY, Md Syduzzaman. Smart Textiles and Nano-Technology: A General Overview. *Journal of Textile Science & Engineering*, 2015. ISSN 21658064. [Cit. 18.1.2020]. Dostupné z: https://www.researchgate.net/publication/279847961_Smart_Textiles_and_Nano-Technology_A_General_Overview
- [8] MESTROVIC, Michael A, Richard J. N. HELMER, Louis KYRATZIS a Dinesh KUMAR. Preliminary study of dry knitted fabric electrodes for physiological monitoring. 2007 3rd International Conference on Intelligent Sensors, Sensor Networks and Information [online]. IEEE, 2007, 601-606 DOI: 10.1109/ISSNIP.2007.4496911. ISBN 978-1-4244-1501-4. [Cit. 18.1.2020]. Dostupné z: <https://ieeexplore.ieee.org/document/4496911>
- [9] Classification and Application of Smart Fabric Textiles. *Testex*. [online] Nov 25, 2020. [Cit. 20.3.2021]. Dostupné z: <https://www.testertextile.com/classification-and-application-of-smart-fabric-textiles/>
- [10] K. J. Cunningham. „How Waterproof/Breathable Fabrics Work“. MyOpenCountry. [online] Aug 27, 2018. [Cit. 20.3.2021]. Dostupné z: <https://www.myopencountry.com/how-waterproof-breathable-fabrics-work/>
- [11] S. Thakur. „Shape Memory Polymers for Smart Textile Applications“. [online] Sep 20, 2017. [Cit. 20.3.2021]. Dostupné z: <https://www.intechopen.com/books/textiles-for-advanced-applications/shape-memory-polymers-for-smart-textile-applications>
- [12] The GORE-TEX Membrane: What it is, how it works and why you need it. Gore-Tex brand. [online] Dec 03, 2019. [Cit. 20.3.2021]. Dostupné z:

- <https://www.gore-tex.com/blog/the-gore-tex-membrane-what-it-is-how-it-works-and-why-you-need-it>
- [13] Y. Kim, H. Kim, a H.-J. Yoo, „Electrical Characterization of Screen-Printed Circuits on the Fabric". IEEE Transactions on Advanced Packaging. 2010, 33(1), s. 196–205. ISSN 1521-3323.[online]. [Cit. 19.1.2020]. Dostupné z: <https://ieeexplore.ieee.org/document/5345718>
- [14] M. Pacelli, G. Loriga, N. Taccini, a R. Paradiso, „Sensing Fabrics for Monitoring Physiological and Biomechanical Variables: E-textile solutions", in 2006 3rd IEEE/EMBS International Summer School on Medical Devices and Biosensors, 2006, s. 1–4.
- [15] MATTANA, Giorgio, Thomas KINKELDEI, David LEUENBERGER, Caglar ATAMAN, Jinyu J. RUAN, Francisco MOLINA-LOPEZ, Andres Vasquez QUINTERO, Giovanni NISATO, Gerhard TROSTER, Danick BRIAND a Nico F. DE ROOIJ. Woven Temperature and Humidity Sensors on Flexible Plastic Substrates for E-Textile Applications. IEEE Sensors Journal. 2013, vol. 13, issue 10, s. 3901-3909. DOI: 10.1109/JSEN.2013.2257167. [Cit. 19.1.2020]. Dostupné z: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6494581>
- [16] T.-H. Kang et al., „Sensors on Textile Substrates for Home-Based Healthcare Monitoring", in 1st Transdisciplinary Conference on Distributed Diagnosis and Home Healthcare, 2006. D2H2., 2006, s. 5–7.
- [17] B. Taji, S. Shirmohammadi, S. Member, V. Groza, a I. Batkin, „Impact of Skin–Electrode Interface on Electrocardiogram Measurements Using Conductive Textile Electrodes", IEEE Trans. Instrum. Meas., roč. 2014, č. 63, čer. 2014.
- [18] Jung-Sim Roh, Yong-Seung Chi, a Tae Jin Kang, „Wearable textile antennas", Int. J. Fash. Des. Technol. Educ., Listopad 2010.[online]. [Cit. 23.1.2020]. Dostupné z: <https://www.tandfonline.com/doi/pdf/10.1080/17543266.2010.521194>
- [19] EVANS, Dave. The Internet of Things how the next evolution of the Internet is changing everything. Cisco Systems, Inc. [online]. [Cit. 30.1.2020]. Dostupné z: https://www.cisco.com/c/dam/en_us/about/ac79/docs/innov/IoT_IBSG_0411FINA L.pdf
- [20] Bubílek A. Využití technologie IoT. Diplomová práce, ZČU Plzeň, 2018.
- [21] Obrázek 2.1: Vývoj a předpověď počtu připojených zařízení v letech 2015-2025 [online]. [Cit. 7.1.2020]. Dostupné z: <https://www.statista.com/statistics/471264/iot-number-of-connected-devices-worldwide/>
- [22] Internet věcí [online]. [Cit. 30.1.2020]. Dostupné z: <http://i2ot.eu/internet-of-things/>
- [23] Consumer Internet of Things (CIoT) – what is it and how does it evolve? [online]. [Cit. 30.1.2020]. Dostupné z: <https://www.i-scoop.eu/internet-of-things-guide/what-is-consumer-internet-of-things-ciot/>

- [24] Představte si budoucnost v propojeném světě internetu Věcí. IDC IOT Forum 2015. [online]. [Cit. 30.1.2020]. Dostupné z: <https://feedit.cz/2015/10/27/idc-iot-forum-2015-predstavte-si-budoucnost-v-propojenem-svete-internetu-veci/>
- [25] PAINE, Tony. Průmyslový Internet věcí a komunikace na okraji.
- [26] MAŘÍK, Vladimír. Průmysl 4.0 a jeho dopady v energetice, dopravě a stavebnictví. 2016.
- [27] Jak internet věcí změní zdravotnictví [online]. [Cit. 31.1.2020]. Dostupné z: https://www.technickytydenik.cz/rubriky/archiv/jak-internet-veci-zmeni-zdravotnictvi_31859.html
- [28] SLAVÍK, Jakub. Co to je a jak funguje inteligentní město – smart city.
- [29] Enterprise Internet of Things (Enterprise IoT) [online]. [Cit. 31.1.2020]. Dostupné z: <http://www.enterox.com/IoT/articles/enterprise-internet-of-things.htm>
- [30] Enterprise System and Internet of Things (IoT) [online]. [Cit. 31.1.2020]. Dostupné z: <http://www.enterox.com/IoT/articles/enterprise-system-iot.htm>
- [31] Libelium. 50 Sensor Applications for a Smarter World. Libelium. [Online] [Cit. 31.1.2020]. Dostupné z: http://www.libelium.com/resources/top_50_iot_sensor_applications_ranking/
- [32] ČERMÁK, Petr a Tomáš Kramný. TECHNOLOGIE IOT [online]. [Cit. 2.2.2020]. Dostupné z: <https://docplayer.cz/87865380-Technologie-iot-s-t-u-d-iu-m.html>
- [33] Řehořík V. Internet věcí na platformě Arduino. Diplomová práce, ZČU Plzeň, 2018.
- [34] Pallavi Sethi and Smruti R. Sarangi. Internet of Things Architectures Protocols and App. [online]. [Cit. 2.2.2020]. Dostupné z: https://www.researchgate.net/publication/312957467_Internet_of_Things_Architectures_Protocols_and_Applications
- [35] Special Issue of the International Journal on Network Management (IJNM) on Management of the Internet of Things and Big Data. University of Nicosia [online]. [Cit. 2.2.2020]. Dostupné z: http://www.cs.unic.ac.cy/cmavrom/IJNM_SI_Internet_of_Things_and_Big_Data.pdf
- [36] ARKKO J., THALER D. and MCPHERSON D. Internet architecture board (IAB). 2011.[online]. [Cit. 2.2.2020]. ISSN: 2070-1721. Dostupné z: <https://tools.ietf.org/html/rfc7452>
- [37] Internet of Things Communications Models [Online] [Cit. 12.2.2020]. Dostupné z: <https://www.supinfo.com/articles/single/4235-internet-of-things>

- [38] Hypertext Transfer Protocol. World Wide Web Consortium . [Online] [Cit. 12.2.2020]. Dostupné z: <https://www.w3.org/Protocols/rfc2616/rfc2616.html>
- [39] MQTT: Enabling the Internet of Things. IBM . [Online]. [Cit. 12.2.2020]. Dostupné z: <https://community.ibm.com/community/user/imwuc/viewdocument/mqtt-enabling-the-internet-of-thin?CommunityKey=183ec850-4947-49c8-9a2e-8e7c7fc46c64&tab=librarydocuments>
- [40] IoT MQTT prakticky v automatizaci. Automatizace.HW . [Online]. [Cit. 12.2.2020]. Dostupné z: <https://automatizace.hw.cz/iot-mqtt-prakticky-v-automatizaci-1dil-uvod.html>
- [41] Vojáček, Antonín. Základní úvod do oblasti internetu věcí (IoT). Automatizace hw. [Online] 16. září 2016. [Cit. 15.2.2020]. Dostupné z: <http://automatizace.hw.cz/zakladni-uvod-do-oblasti-internetu-veci-iot.html>
- [42] Balek J. Internet věcí na platformě STM32. Diplomová práce, ZČU Plzeň, 2017.
- [43] SimpleCell [online]. [Cit. 17.2.2020]. Dostupné z: <https://simplecell.eu/>
- [44] Sigfox [online]. [Cit. 17.2.2020]. Dostupné z: <https://www.sigfox.com/>
- [45] SigFox Protocol Stack | LTN | SigFox Protocol Layers. [online]. [Cit. 17.2.2020]. Dostupné z: <https://www.rfwireless-world.com/Tutorials/Sigfox-protocol-stack.html>
- [46] Český telekomunikační úřad. Využívání vymezených radiových kmitočtů. Český telekomunikační úřad. [Online] [Cit. 19.2.2020]. Dostupné z: <http://www.ctu.cz/vyuzivani-vymezenych-radiovykh-kmitoctu>
- [47] Služby CRA IoT [online]. [Cit. 19.2.2020]. Dostupné z: <https://www.cra.cz/sluzby-iot>
- [48] LoRa Alliance. LoRaWan 1.0.1 Specification. zendesk. [Online] listopad 2015. [Cit. 19.2.2020]. Dostupné z: <https://enablingsupport.zendesk.com/hc/en-us/articles/205928502-LoRaWan-1-0-1-specification>
- [49] Semtech. LoRaWAN. Semtech. [Online] červen 2016. [Cit. 19.2.2020]. Dostupné z: <https://www.semtech.com/lora>
- [50] Cisco. 802.11ac: The Fifth Generation of Wi-Fi Technical White Paper. Cisco. [Online] 27. březen 2014. [Cit. 23.2.2020]. Dostupné z: http://www.cisco.com/c/en/us/products/collateral/wireless/aironet-3600-series/white_paper_c11-713103.html
- [51] Baoyong Chi, Lixue Kuang, Haikun Jia. A 60-GHz wireless transceiver with dual-mode power amplifier for IEEE 802.11ad in 65nm CMOS. IEEE. [Online] 21. červenec 2016. [Cit. 23.2.2020]. Dostupné z: <https://www.semanticscholar.org/paper/A-60-GHz-wireless-transceiver-with-dual-mode-power-Chi-Kuang/281b5462419d1261c13e0654219e5bc7618ec0ba>

- [52] Shinill Chang, Hyunchol Shin. Design considerations for bluetooth low energy CMOS RF transceivers for IoT. IEEE. [Online] 20. říjen 2016. [Cit. 24.2.2020]. Dostupné z: <http://ieeexplore.ieee.org/document/7601307/>. ISBN 978-1-4673-8801-6.
- [53] Sims, Gary. The truth about Bluetooth 5 – Gary explains. Android authority. [Online] 14. duben 2017. [Cit. 24.2.2020]. Dostupné z: <https://www.androidauthority.com/bluetooth-5-speed-range-762369/>
- [54] Bluetooth Sig. Bluetooth core 5 specification. Bluetooth. [Online] 6. prosinec 2016. [Cit. 24.2.2020]. Dostupné z: <https://www.bluetooth.com/specifications/adopted-specifications>.
- [55] SCHLIENZ, J., RADDINO, D. Narrowband Internet of Things (Rohde&Schwarz Whitepaper). [online]. [Cit. 25.2.2020]. Dostupné z: https://cdn.rohde-schwarz.com/pws/dl_downloads/dl_application/application_notes/1ma266/1MA266_0e_NB_IoT.pdf
- [56] Nokia. LTE evolution for IoT connectivity (Nokia Whitepaper). [online]. [Cit. 25.2.2020]. Dostupné z: https://halberdbastion.com/sites/default/files/2017-06/Nokia_LTE_Evolution_for_IoT_Connectivity_White_Paper.pdf
- [57] Landström S., Bergström J., Westerberg E., Hammarwall D. NB-IoT: a sustainable technology for connecting billions of devices. [online]. [Cit. 25.2.2020]. Dostupné z: <https://www.ericsson.com/49ead5/assets/local/reports-papers/ericsson-technology-review/docs/2016/etr-narrowband-iot.pdf>
- [58] SEMTECH. SX1276/77/78/79 - 137 MHz to 1020 MHz Low Power Long Range Transceiver. [Online] [Cit. 20.3.2020]. Dostupné z: <https://www.mouser.com/datasheet/2/761/sx1276-1278113.pdf>
- [59] Obrázky pinoutů modulů Pycom. IoT [online]. [Cit. 20.3.2020]. Dostupné z: <https://docs.pycom.io/datasheets/>
- [60] Wearable PCB design guidelines. Proto-Electronics [Online] [Cit. 12.2.2021]. Dostupné z: <https://www.proto-electronics.com/blog/wearable-pcb-design-guidelines>
- [61] PCBs for wearables and ubiquitous computation. Rushpcb [online]. [Cit. 12.2.2021]. Dostupné z: <https://rushpcb.com/pcbs-for-wearables-and-ubiquitous-computing/>
- [62] Wearable device PCB design. Andwin Circuits [online]. [Cit. 12.2.2021]. Dostupné z: <https://www.andwinpcb.com/wearable-device-pcb-design-consider-three-important-factors/>
- [63] Fundamentals guide wearable PCB designs. Embedded. [online] September 28, 2015. [Cit. 12.2.2021]. Dostupné z: <https://www.embedded.com/fundamentals-guide-wearable-pcb-designs/>

- [64] The Many Aspects of Wearables and Printed Circuit Boards (PCBs). Advanced Circuits [online] September 1, 2016. [Cit. 12.2.2021]. Dostupné z: <https://www.4pcb.com/blog/the-many-aspects-of-wearables-and-printed-circuit-boards-pcbs/>
- [65] Printed Circuit Board – Game Changer Of The Future. Crimp Circuits. [online] Jan 03, 2020. [Cit. 12.2.2021]. Dostupné z: <https://www.crimpcircuits.com/blog/printed-circuit-board-game-changer-of-the-future/>
- [66] A. Bahl. „Why Are Flex PCBs Used in Medical Devices and Wearables?“. Sierra Circuits. [online] Dec 14, 2020. [Cit. 12.2.2021]. Dostupné z: <https://www.protoexpress.com/blog/flex-pcb-used-medical-devices-wearables/>
- [67] A. Bahl. „How to Choose PCB Materials and Laminates for Fabrication“. Sierra Circuits. [online] Jun 16, 2020. [Cit. 12.2.2021]. Dostupné z: <https://www.protoexpress.com/blog/comparing-the-manufacturability-of-pcb-laminates/>
- [68] Flex & Rigid-Flex PCB. Sierra Circuits. [online]. [Cit. 12.2.2021]. Dostupné z: <https://www.protoexpress.com/pcb/flex-pcb-manufacturing-capabilities/>
- [69] What is Flexible PCB. PCB Way. [online]. [Cit. 12.2.2021]. Dostupné z: https://www.pcbway.com/pcb_prototype/What_is_Flexible_PCB.html
- [70] Easy-On FFC FPC Connectors. Molex. [online]. [Cit. 12.2.2021]. Dostupné z: https://www.molex.com/molex/products/family/easyon_ffc_fpc_connectors
- [71] Hexoskin Health Sensors & AI. Hexoskin. [online]. [Cit. 27.2.2021]. Dostupné z: <https://www.hexoskin.com/>
- [72] Chronolife Real Time Intelligence for Healthcare. Chronolife. [online]. [Cit. 27.2.2021]. Dostupné z: <https://www.chronolife.net/>
- [73] B. Brown. „Tell-All T-Shirt Monitors Your Health“. HealthTech Insider. [online] Jan 01, 2020. [Cit. 27.2.2021]. Dostupné z: <https://healthtechinsider.com/2020/01/01/tell-all-t-shirt-monitors-your-health/>
- [74] Nexkin Multi-Parameter Body Monitoring Shirt Unveiled. MedGadget. [online] Dec 17, 2019. [Cit. 27.2.2021]. Dostupné z: <https://www.medgadget.com/2019/12/nexkin-multi-parameter-body-monitoring-shirt-unveiled.html>
- [75] Wearable ECG monitor with an automated stress/fatigue level determination and AI-based arrhythmia diagnostics. HeartIn. [online]. [Cit. 28.2.2021]. Dostupné z: <https://heartin.net/>
- [76] B. Brown. „ECG T-shirt Delivers Continuous Monitoring“. HealthTech Insider. [online] Jan 29, 2019. [Cit. 28.2.2021]. Dostupné z: <https://healthtechinsider.com/2019/01/29/ecg-t-shirt-delivers-continuous-monitoring/>

- [77] C. Yerbic. „HeartIn Wearable Preventative Arrhythmia Care“. *Cardiology 2.0*. [online] Feb 2020. [Cit. 28.2.2021]. Dostupné z: <https://magazine.cardiology2.com/feb2020/heartin>
- [78] Master Caution®. HealthWatch. [online]. [Cit. 1.3.2021]. Dostupné z: <https://healthwatchtech.com/>
- [79] B. Brown. „Interwoven Garment with 12–Lead ECG and More“. *HealthTech Insider*. [online] Feb 14, 2019. [Cit. 1.3.2021]. Dostupné z: <https://healthtechinsider.com/2019/02/14/interwoven-garment-with-12-lead-ecg-and-more-video/>
- [80] Clothes with built-in ECG and heart rate measurement. *Emlgare*. [online]. [Cit. 1.3.2021]. Dostupné z: <https://emlgare.com/>
- [81] S. Mahdiani, V. Jeyhani, M. Peltokangas and A. Vehkaoja, "Is 50 Hz high enough ECG sampling frequency for accurate HRV analysis?," 2015 37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), Milan, Italy, 2015, pp. 5948-5951, [online]. [Cit. 6.4.2021]. doi: 10.1109/EMBC.2015.7319746.
- [82] Kwon, Ohhwan et al. "Electrocardiogram Sampling Frequency Range Acceptable for Heart Rate Variability Analysis." *Healthcare informatics research* vol. 24,3 (2018): 198-206. [online]. [Cit. 6.4.2021]. doi:10.4258/hir.2018.24.3.198
- [83] Mathias Baumert, Martin Schmidt, Sebastian Zaunseder, Alberto Porta, "Effects of ECG sampling rate on QT interval variability measurement." *Biomedical Signal Processing and Control*, Volume 25, 2016, Pages 159-164, ISSN 1746-8094, [online]. [Cit. 6.4.2021]. <https://doi.org/10.1016/j.bspc.2015.11.011>
- [84] Aftab Khan, Nils Hammerla, Sebastian Mellor, ThomasPlötz, "Optimising sampling rates for accelerometer-based human activity recognition." *Pattern Recognition Letters*, Volume 73, 2016, Pages 33-40, ISSN 0167-8655, [online]. [Cit. 6.4.2021]. <https://doi.org/10.1016/j.patrec.2016.01.001>
- [85] Sigcha L, Pavón I, Arezes P, Costa N, De Arcas G, López JM. Occupational Risk Prevention through Smartwatches: Precision and Uncertainty Effects of the Built-In Accelerometer. *Sensors*. 2018; 18(11):3805. [online]. [Cit. 6.4.2021]. <https://doi.org/10.3390/s18113805>
- [86] LEVORA, Viktor. Elektrické propojovací struktury v chytrých textiliích a jejich spolehlivost. B.m.,2015. Západočeská univerzita v Plzni.
- [87] Obrázky příkladů permanentních spojení. [online]. [Cit. 12.3.2021]. Dostupné z: <https://www.kobakant.at/DIY/?p=1272>
- [88] Obrázky příkladů permanentních spojení. [online]. [Cit. 12.3.2021]. Dostupné z: <http://highlowtech.org/wiki/pmwiki.php?n=Main.FabricPCBSolder>

Seznam obrázků

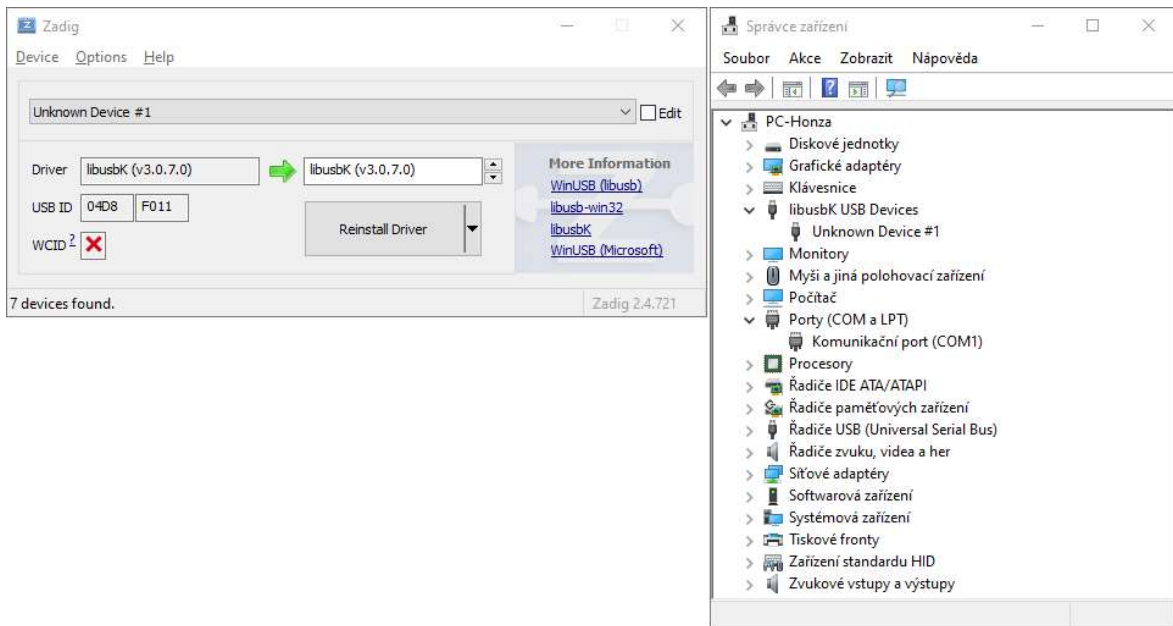
OBR. 1.1: UKÁZKA SENZORU VLHKOSTI A TEPLoty NA TENKÉM POLYIMIDOVÉM PÁSKU VETKANÉM DO TEXTILIE (PŘEVZATO Z [16] A UPRAVENO)	18
OBR. 1.2: STRUKTURA NETKANÉHO TEXTILNÍHO KAPACITNÍHO DÝCHACÍHO SENZORU (PŘEVZATO Z [16] A UPRAVENO).....	19
OBR. 1.3: PRINCIP PIEZOREZISTIVNÍCH SENZORŮ (PŘEVZATO Z [6])	20
OBR. 1.4: ČTYŘVRSTVÁ HYBRIDNÍ DESKA PLOŠNÝCH SPOJŮ (PŘEVZATO ZE [66] A UPRAVENO).....	23
OBR. 1.5: UKÁZKA KONEKTORŮ PRO FLEXIBILNÍ PLOŠNÉ SPOJE (PŘEVZATO ZE [70]).....	24
OBR. 1.6: PŘÍKLADY PERMANENTNÍCH SPOJENÍ – PŘÍŠITÍ (VLEVO) A PÁJENÍ (VPRAVO) (PŘEVZATO Z [87][88])	25
OBR. 1.7: PŘÍKLADY ROZPOJITELNÝCH SPOJENÍ – DRUKY (VLEVO) A PATENTKY (VPRAVO) (PŘEVZATO Z [86])	25
OBR. 1.8: HEXOSKIN A JEHO ONLINE PLATFORMA (PŘEVZATO ZE [71] A UPRAVENO).....	26
OBR. 1.9: PRODUKTY EMGLARE (PŘEVZATO ZE [80] A UPRAVENO)	26
OBR. 1.10: ZLEVA: HEARTIN [75], MASTER CAUTION [78] A NEXKIN [72].....	27
OBR. 2.1: VÝVOJ A PŘEDPOVĚĎ POČTU PŘIPOJENÝCH ZAŘÍZENÍ V LETECH 2015-2025 (PŘEVZATO Z [26]).....	28
OBR. 2.2: TŘÍVRSTVÁ ARCHITEKTURA IoT, (PŘEVZATO Z [34] A UPRAVENO)	30
OBR. 3.1: PŘEHLED DOSAHU BEZDRÁTOVÝCH TECHNOLOGIÍ (INSPIROVÁNO [41])	32
OBR. 3.2: POKRYTÍ SÍTÍ SIGFOX V ČR (PŘEVZATO Z [43])	33
OBR. 3.3: ARCHITEKTURA SÍTĚ SIGFOX (PŘEVZATO Z [44]).....	34
OBR. 3.4: PŘENOSOVÝ PROTOKOL (PŘEVZATO Z [45]).....	35
OBR. 3.5: POKRYTÍ SÍTĚ LoRa (PŘEVZATO Z [47]).....	36
OBR. 3.6: UKÁZKA SIGNÁLU PO MODULACI ROZMÍTANÝM SIGNÁLEM (VLEVO) A PO MODULACI SKOKOVOU ZMĚNOU KMITOČTU (VPRAVO) (PŘEVZATO Z [33]).....	37
OBR. 3.7: STRUKTURA LoRaWAN RÁMCE (PŘEVZATO Z [58] A UPRAVENO)	38
OBR. 3.8: AKTIVITA KONCOVÝCH ZAŘÍZENÍ LoRa.....	39
OBR. 4.1: BLOKOVÉ SCHÉMA SYSTÉMU PRO NOSITELNOU ELEKTRONIKU	44
OBR. 4.2: PINOUT DIAGRAM DESKY PYSENSE (PŘEVZATO Z [59]).....	49
OBR. 4.3: UMÍSTĚNÍ JEDNOTLIVÝCH SOUČÁSTÍ DESKY PYSENSE (PŘEVZATO Z [59]).....	50
OBR. 4.4: PINOUT DIAGRAM MODULU LoPy4 (PŘEVZATO Z [59])	51
OBR. 4.5: UMÍSTĚNÍ SENZORŮ A ŘÍDÍČÍHO MODULU NA PODPRSENCE.....	52
OBR. 4.6: KLASICKÁ PATENTKA NA TEXTIL.....	53
OBR. 4.7: NÁVRH PROTOTYPU ŘÍDÍČÍHO MODULU S KRABÍČKOU A PATENTKY	53
OBR. 4.8: POSLEDNÍ OKNO AKTUALIZAČNÍHO PROGRAMU MODULU LoPy4.....	55
OBR. 4.9: STAV SPOJENÍ ZAŘÍZENÍ SE SIGFOX BACKENDEM.....	57
OBR. 4.10: UKÁZKA PŘIJATÉ ZPRÁVY	57
OBR. 4.11: OVĚŘENÍ SPRÁVNOSTI ZPRÁVY PŘEVEDENÍM ZPĚT NA TEXT	58
OBR. 4.12: OKNO „DEVICE OVERVIEW“ V SÍTĚ TTN	60
OBR. 4.13: DASHBOARD CLOUDOVÉ SLUŽBY PYBYTES	63
OBR. 4.14: ROZDĚLENÍ DAT POMOCÍ FLOW.....	66
OBR. 4.15: VIZUALIZACE NAMĚŘENÝCH HODNOT	68
OBR. 4.16: UPOZORNĚNÍ NA NÍZKÝ STAV BATERIE PROSTŘEDNICTVÍM EMAILU	69
OBR. 4.17: NASTAVENÍ CALLBACKU PRO SLUŽBU Wia	70
OBR. 4.18: INTEGRACE TTN DO CLOUDU Wia	72
OBR. 4.19: FORMÁTOVÁNÍ A DEKÓDOVÁNÍ DAT V BACKENDU TTN.....	74
OBR. 5.1: SCHÉMA ZAPOJENÍ MĚŘENÍ	77
OBR. 5.2: PRŮBĚH ODEBÍRANÉHO PROUDU ZAŘÍZENÍM S VYUŽITÍM SÍTĚ Wi-Fi.....	78
OBR. 5.3: PRŮBĚH ODEBÍRANÉHO PROUDU ZAŘÍZENÍM PŘI KONTINUÁLNÍM MĚŘENÍ S VYUŽITÍM SÍTĚ Wi-Fi..	79
OBR. 5.4: PRŮBĚH ODEBÍRANÉHO PROUDU ZAŘÍZENÍM S VYUŽITÍM SÍTĚ LoRa	80
OBR. 5.5: PRŮBĚH ODEBÍRANÉHO PROUDU ZAŘÍZENÍM S VYUŽITÍM SÍTĚ SIGFOX	81

Seznam tabulek

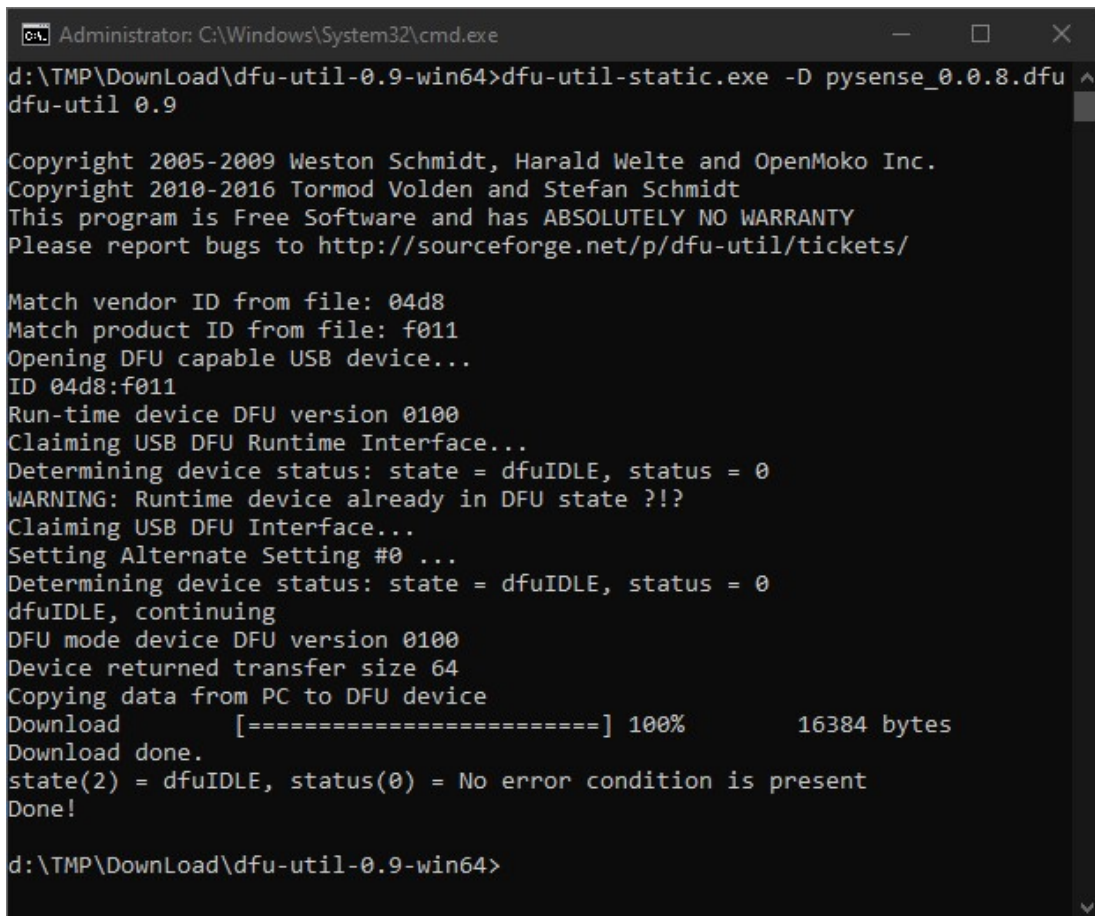
TABULKA 1.1: PŘEHLED TYPŮ A SNÍMANÉ VELIČINY SENZORŮ (PŘEVZATO Z [6] A UPRAVENO).....	18
TABULKA 1.2: MOŽNOSTI OHÝBÁNÍ FLEXIBILNÍCH DPS	22
TABULKA 1.3: SROVNÁNÍ POLYIMIDOVÝCH A POLYESTEROVÝCH SUBSTRÁTŮ FLEXIBILNÍCH DPS.....	23
TABULKA 3.1: ROZDĚLENÍ BEZDRÁTOVÝCH TECHNOLOGIÍ DLE DOSAHU [41].....	32
TABULKA 3.2: TŘÍDY DATOVÉHO TOKU V SÍTI LORA (PŘEVZATO Z [48] A UPRAVENO).....	37
TABULKA 5.1: NAMĚŘENÉ PARAMETRY SNR A RSSI	75
TABULKA 5.2: ODHAD DOBY PROVOZU ZAŘÍZENÍ	82

Přílohy

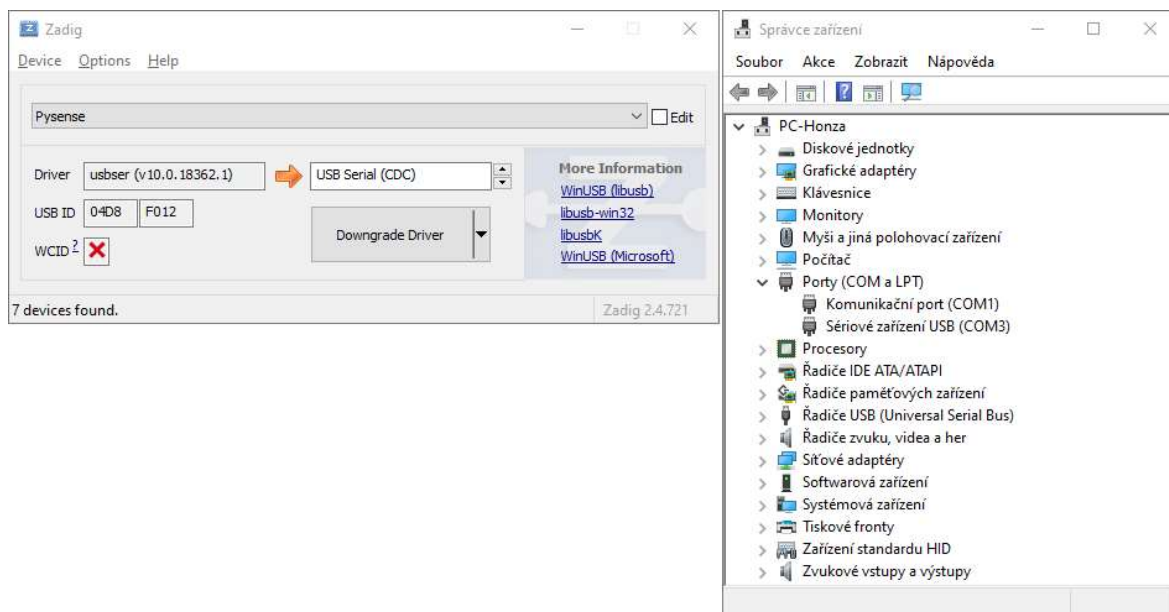
Příloha A – Doprovodné obrázky k aktualizaci firmwaru desky Pysense



Okna programů Zadig a Správce zařízení před aktualizací firmwaru



Okno příkazové řádky s instalací firmwaru



Okna programů Zadig a Správce zařízení po aktualizaci firmwaru

Příloha B – Program obsluhy senzorů a řízení bezdrátové komunikace pro Pybytes

```
import math
from pysense import Pysense
from LIS2HH12 import LIS2HH12
from SI7006A20 import SI7006A20
from LTR329ALS01 import LTR329ALS01
from MPL3115A2 import MPL3115A2, ALTITUDE, PRESSURE

py = Pysense()
li = LIS2HH12(py)
lt = LTR329ALS01(py)
si = SI7006A20(py)
mp = MPL3115A2(py, mode=PRESSURE)

# nastavení intervalu spánku
sleepTime = 600

# funkce pro získání dat ze senzorů
def getSensorData():

    batt = py.read_battery_voltage()
    press = mp.pressure()/1000 # dělíme 1000x pro převod na kPa
    alt = 44330.77*(1-pow((press/101.326),0.1902632)) # výpočet
                                                    nadmořské výšky z naměřeného tlaku
    temp = (mp.temperature() + si.temperature())/2 # výpočet průměru
                                                    teplot z dvou senzorů teploty

    relHum = si.humidity()
    dewPoint = si.dew_point()
    (accelX, accelY, accelZ) = li.acceleration()
    # výpočet celkové velikosti působícího zrychlení
    accel = math.sqrt(pow(accelX,2) + pow(accelY,2) + pow(accelZ,2))
    light = round(lt.lux())

    data = {'batt': batt, 'temp': temp, 'press': press, 'alt': alt, 'relHum': relHum, 'dewPoint': dewPoint, 'accel': accel, 'light': light, 'heartRate': heartRate}

    return data
```

```
def sendData(data):
    pybytes.send_signal(1, sensorData['light'])
    pybytes.send_signal(2, sensorData['accel'])
    pybytes.send_signal(3, sensorData['batt'])
    pybytes.send_signal(4, sensorData['alt'])
    pybytes.send_signal(5, sensorData['temp'])
    pybytes.send_signal(6, sensorData['press'])
    pybytes.send_signal(7, sensorData['relHum'])
    pybytes.send_signal(8, sensorData['dewPoint'])

def goToSleep():
    time.sleep(0.2)
    py.setup_sleep(sleepTime)
    py.go_to_sleep()

sensorData = getSensorData() # načtení dat ze senzorů
sendData(sensorData) # odeslání dat na Pybytes

goToSleep() # přechod do režimu spánku
```

Příloha C – Program obsluhy senzorů a řízení bezdrátové komunikace

```
from network import Sigfox
from network import LoRa
from network import WLAN
import socket
import machine
import ubinascii
import time
import pycom
import micropython
import math
import ustruct
import urequests as requests
from pysense import Pysense
from LIS2HH12 import LIS2HH12
from SI7006A20 import SI7006A20
from LTR329ALS01 import LTR329ALS01
from MPL3115A2 import MPL3115A2, ALTITUDE, PRESSURE

pycom.heartbeat(False) # vypnutí integrované RGB led

py = Pysense()
li = LIS2HH12(py)
lt = LTR329ALS01(py)
si = SI7006A20(py)
mp = MPL3115A2(py, mode=PRESSURE)
# inicializace AD převodníku na pinu 16 s útlumem 11dB
adcObject = machine.ADC()
adc = adcObject.channel(pin = 'P16', attn = adcObject.ATTN_11DB)

#volba sítě: Wi-Fi(=1), LoRa(=2), Sigfox(=3), all(=4)
network = 1

# definice interval doby spánku
sleepTime = 1200

# definice vzorkovacích frekvencí pro akcelerometr a sensory EKG a dýchání
ecgSampleFrequency = 250
accelSampleFrequency = 100
respSampleFrequency = 10

# OTAA aktivační parametry pro TTN
dev_eui = ubinascii.unhexlify('70B3D54992E2B1F8')
app_eui = ubinascii.unhexlify('70B3D57ED002EC71')
app_key = ubinascii.unhexlify('B983A18BF614E9C270D789137FC2298A')

# definice názvu a hesla Wi-Fi sítě
```

```
WIFI_SSID = 'ASUS_5Z'
WIFI_KEY = '*****'

# unikátní klíč a odkaz pro odesílání dat na Wiu pomocí Wi-Fi
DEVICE_SECRET_KEY = 'd_sk_Xe97qN9k88c7QsUUqobQ3DtA'
url = "https://api.wia.io/v1/events"
headers = { "Authorization": "Bearer " + DEVICE_SECRET_KEY, "Content-
Type": "application/json" }

def goToSleep():
    time.sleep(0.2)
    py.setup_sleep(sleepTime)
    py.go_to_sleep()

def loadEcgSample(): # načtení ekg vzorků ze souboru do proměnné
    f = open('ecg_sample_1min_' + str(ecgSampleFrequency) + 'hz.txt','r')
    sample = f.read() #uloží obsah souboru do proměnné jako jeden dlouhý
        textový řetězec
    f.close()
    sample = sample.replace(sample[0],'') #odstraní ByteOrderMask
    sample = sample.split(",") #rozdělení textového řetězce na mnoho malých
        s použitím ',' jako separátoru
    sample = [float(g) for g in sample] #převod textu na čísla
    return sample

ecg_sample = loadEcgSample()
i = 0
measTime = 2
SAMPLE_LEN = len(ecg_sample)
def measure(): # funkce pro měření a načítání hodnot ze senzorů
    global i
    global ecg_sample
    global SAMPLE_LEN
    ecgSampleValues = []
    ecgSampleTimes = []
    accelXSampleValues = []
    accelYSampleValues = []
    accelZSampleValues = []
    accelSampleTimes = []
    respSampleValues = []
    respSampleTimes = []
    stopTime = time.ticks_ms() + measTime*1000
    lastEcgSample = time.ticks_ms()
    lastAccelSample = time.ticks_ms()
    lastRespSample = time.ticks_ms()
    while time.ticks_ms() < stopTime:
        if time.ticks_ms() > (lastEcgSample + 350/ecgSampleFrequency):
            ecgSampleValues.append(ecg_sample[i])
            ecgSampleTimes.append(time.ticks_ms())
```

```

        lastEcgSample = time.ticks_ms()
        i=i+1
        if i == SAMPLE_LEN:
            i=0
        if time.ticks_ms() > (lastAccelSample + 600/accelSampleFrequency):
            (X,Y,Z)=li.acceleration()
            accelXSampleValues.append(X)
            accelYSampleValues.append(Y)
            accelZSampleValues.append(Z)
            accelSampleTimes.append(time.ticks_ms())
            lastAccelSample = time.ticks_ms()
        if time.ticks_ms() > (lastRespSample + 1000/respSampleFrequency):
            respSampleValues.append(adc())
            respSampleTimes.append(time.ticks_ms())
            lastRespSample = time.ticks_ms()
    return ecgSampleValues,ecgSampleTimes,accelXSampleValues,accelYSampleVal
ues,accelZSampleValues,accelSampleTimes,respSampleValues,respSampleTimes

```

```

RPM_TERSHOLD_WINDOW = 10*respSampleFrequency
rpm_History_Values = []
rpm_History_Times = []
def calculateRpm(rpmValues, rpmTimes): # funkce pro výpočet nádehů za minutu

```

```

    global rpm_History_Values
    global rpm_History_Times
    global RPM_TERSHOLD_WINDOW
    rpm_History_Values = rpm_History_Values + rpmValues
    rpm_History_Values = rpm_History_Values[-RPM_TERSHOLD_WINDOW:]
    rpm_History_Times = rpm_History_Times + rpmTimes
    rpm_History_Times = rpm_History_Times[-RPM_TERSHOLD_WINDOW:]
    tresholdMin,tresholdMax =min(rpm_History_Values),max(rpm_History_Values)
    tresholdHigh = tresholdMin + (tresholdMax-tresholdMin)*(0.75)
    tresholdLow = tresholdMin + (tresholdMax-tresholdMin)*(0.5)

```

```

    breaths = []
    breathFlag = False
    vzorek = 0

```

```

    while vzorek < len(rpm_History_Values):
        if rpm_History_Values[vzorek] > tresholdHigh and breathFlag == False:
            breathFlag = True
            breaths.append(rpm_History_Times[vzorek])
        if rpm_History_Values[vzorek] < tresholdLow and breathFlag == True:
            breathFlag = False
        vzorek=vzorek+1

```

```

if (tresholdHigh-tresholdLow) > 0.3:
    rpm = 60*(breaths[-1] - breaths[0])/(1000*len(breaths))
    return rpm
else:
    return 0

ECG_TERSHOLD_WINDOW = 10*ecgSampleFrequency
ecg_History_Values = []
ecg_History_Times = []
def calculateBpm(ecgValues, ecgTimes): # funkce pro výpočet úderů srdce

    global ecg_History_Values
    global ecg_History_Times
    global ECG_TERSHOLD_WINDOW
    ecg_History_Values = ecg_History_Values + ecgValues
    ecg_History_Values = ecg_History_Values[-ECG_TERSHOLD_WINDOW:]
    ecg_History_Times = ecg_History_Times + ecgTimes
    ecg_History_Times = ecg_History_Times[-ECG_TERSHOLD_WINDOW:]
    tresholdMin, tresholdMax = min(ecg_History_Values), max(ecg_History_Values)
    tresholdHigh = tresholdMin + (tresholdMax-tresholdMin)*(0.75)
    tresholdLow = tresholdMin + (tresholdMax-tresholdMin)*(0.5)

    beats = []
    beatFlag = False
    vzorek = 0

    while vzorek < len(ecg_History_Values):
        if ecg_History_Values[vzorek] > tresholdHigh and beatFlag == False:
            beatFlag = True
            beats.append(ecg_History_Times[vzorek])
        if ecg_History_Values[vzorek] < tresholdLow and beatFlag == True:
            beatFlag = False
            vzorek=vzorek+1

    if (tresholdHigh-tresholdLow) > 0.3:
        bpm = 60*(beats[-1] - beats[0])/(1000*len(beats))
        return bpm
    else:
        return 0

# získání dalších dat a uložení všech naměřených hodnot do seznamu
def getSensorData(rpm,bpm,eV,eT,aX,aY,aZ,aT):

    batt = py.read_battery_voltage()
    ecg = eT,eV
    accel = aT,aX,aY,aZ
    temp = (mp.temperature() + si.temperature())/2 # výpočet teploty z dvou
                                                    integrovaných senzorů

```

```

    accelAvg = math.sqrt(pow(sum(aX)/len(aX),2) + pow(sum(aY)/len(aY),2) +
pow(sum(aZ)/len(aZ),2)) # výpočet velikosti vektoru celkového gravitačního
zrychlení působící na zařízení

    data = {'batt': batt, 'temp': temp, 'ecg': ecg, 'accel': accel, 'accelAvg
': accelAvg, 'heartRate': bpm, 'breathRate': rpm}

    return data

# pack/encode data for Sigfox in order to save space (getSensorData() retur
ns json dictionary with float values, each 4 bytes in size)
def packData(sensorData):

    battPack = ustruct.pack('b', round(sensorData['batt']*100)-250) # 1 B
    tempPack = ustruct.pack('b', round(sensorData['temp']*10)-250) # 1 B
    heartRatePack = ustruct.pack('b', round(sensorData['heartRate'])) # 1 B
    breathRatePack = ustruct.pack('b', round(sensorData['breathRate'])) # 1 B
    accelAvgPack = ustruct.pack('h', round(sensorData['accelAvg']*100)) # 2 B

    dataPacket = battPack + tempPack + heartRatePack + accelAvgPack + breath
RatePack

    return dataPacket # 5 float hodnot (20 B) zmenšeno na 6 bajtů

def sendToSigfox():
    # iniciace Sigfoxu pro RCZ1 region (Evropa)
    sigfox = Sigfox(mode = Sigfox.SIGFOX, rcz = Sigfox.RCZ1)
    # vytvoření Sigfox socketu
    s = socket.socket(socket.AF_SIGFOX, socket.SOCK_RAW)
    # uvedení socketu do blokujícího stavu
    s.setblocking(True)
    # nastavení pouze pro uplink dat
    s.setsockopt(socket.SOL_SIGFOX, socket.SO_RX, False)
    # nastavení socket jako jeden rámeček
    s.setsockopt(socket.SOL_SIGFOX, socket.SO_TX_REPEAT, 0)
    # odeslání dat
    s.send(dataToSend)

def sendToLora():
    # Inicializace LoRa v LORAWAN módu.
    lora = LoRa(mode=LoRa.LORAWAN, region=LoRa.EU868)
    lora.nvram_restore() # načtení původních TTN klíčů

    if not lora.has_joined(): # join a network using OTAA
        lora.join(activation=LoRa.OTAA, auth=(dev_eui, app_eui, app_key),
timeout=0, dr=0) # dr=0 represents SF=12
    else:
        pass

```

```
iterace = 0
# čekání na připojení k síti
while not lora.has_joined():
    time.sleep(2.5)
    iterace = iterace + 1
    if iterace == 15:
        #print('Could not join, going to sleep...')
        goToSleep()

# vytvoření LoRa socketu
ss = socket.socket(socket.AF_LORA, socket.SOCK_RAW)
# nastavení SF a přenosových rychlostí
ss.setsockopt(socket.SOL_LORA, socket.SO_DR, 0)
# uvedení socket do blokujícího stavu
ss.setblocking(True)
# odslání dat
ss.send(dataToSend)
# uvedení socket do neblokujícího stavu
ss.setblocking(False)

lora.nvram_save() # uložení TTN klíčů do nvram paměti

# inicializace Wi-Fi do STA módu
wlan = WLAN(mode=WLAN.STA)
def connectToWifi(): # funkce pro připojení Wi-Fi síti
    try:
        wlan.connect(WIFI_SSID, auth=(WLAN.WPA2, WIFI_KEY), timeout=5000)
        while not wlan.isconnected():
            machine.idle()
    except Exception:
        goToSleep()
        pass

# odeslání dat na Wiu
def post_event(json):
    try:
        if json is not None:
            req = requests.post(url=url, headers=headers, json=json)
            if req.status_code is not 200:
                machine.reset()
            else:
                #print('posting', json, 'to Wia')
                pass
            return req.json()
        else:
            pass
    except:
        pass
```



```
def sendToWifi(sensorData): # kontrola zdali je již Wi-Fi připojená
    if wlan.isconnected():
        post_event({"name": 'sensorData', "data": sensorData})
    else:
        connectToWifi()
        post_event({"name": 'sensorData', "data": sensorData})

while True:
    (eV,eT,aX,aY,aZ,aT,rV,rT)=measure()
    bpm = calculateBpm(eV,eT)
    rpm = calculateRpm(rV,rT)
    sensorData = getSensorData(rpm,bpm,eV,eT,aX,aY,aZ,aT)
    dataToSendToLPWAN = packData(sensorData)

    if network == 1:
        sendToWifi(sensorData)
    if network == 2:
        sendToLora(dataToSendToLPWAN)
    elif network == 3:
        sendToSigfox(dataToSendToLPWAN)
    elif network == 4:
        sendToLora(dataToSendToLPWAN)
        time.sleep(0.5)
        sendToSigfox(dataToSendToLPWAN)
        time.sleep(0.5)
        sendToWifi(sensorData)

#goToSleep()
```

Příloha D – Úprava knihovny LTR329ALS01.py

Senzor osvětlení desky Pysense integruje dvě nezávislá čidla, každé s jinou charakteristikou citlivosti v závislosti na vlnové délce. Výslednou celkovou hodnotu osvětlení ale nemůžeme kvůli různým charakteristikám citlivosti vypočítat jednoduchým součtem nebo zprůměrováním. Konkrétní vzorce pro výpočet celkového osvětlení nejsou součástí knihovny ani veřejně dostupné dokumentace k senzoru. Výrobce pouze zmiňuje, že vzorce jsou součástí přílohy dokumentace, avšak tato příloha chybí. Naštěstí byl výrobce ochoten zaslat přílohu se vzorci emailem. Knihovnu LTR329ALS01.py otevřeme pro úpravu v textovém editoru a na její konec vložíme následující funkci:

```
def lux(self):
    light_level = self.light()
    if light_level[0]+light_level[1] > 0:
        ratio = light_level[1]/(light_level[0]+light_level[1])
        if ratio < 0.45:
            return (1.7743 * light_level[0] + 1.1059 * light_level[1]) /
self.ALS_GAIN_VALUES[self.gain] / self.ALS_INT_VALUES[self.integration]
        elif ratio < 0.64 and ratio >= 0.45:
            return (4.2785 * light_level[0] - 1.9548 * light_level[1]) /
self.ALS_GAIN_VALUES[self.gain] / self.ALS_INT_VALUES[self.integration]
        elif ratio < 0.85 and ratio >= 0.64:
            return (0.5926 * light_level[0] + 0.1185 * light_level[1]) /
self.ALS_GAIN_VALUES[self.gain] / self.ALS_INT_VALUES[self.integration]
        else:
            return 0
    else:
        return 0
```

Tuto funkci je pak možné volat z řídicího programu modulu.